NASA-CR-171696
19840004489

Proceedings
of the

NASA

Symposium
on
MATHEMATICAL PATTERN RECOGNITION
& IMAGE ANALYSIS

June 1-3, 1983
Johnson Space Center
Houston, Texas

All Missing Pages

Missing in Original Document

FINAL REPORT

"MATHEMATICAL PATTERN RECOGNITION

AND IMAGE ANALYSIS PROGRAM"

Contract NAS 9-16664

July 16, 1982 - July 15, 1983

Prepared for:

Earth Resources Research Division
NASA/Johnson Space Center
Houston, Texas 77058

by

L. F. Guseman, Jr.
Principal Investigator and MPRIA Program Coordinator
Department of Mathematics
Texas A&M University
College Station, Texas 77843

TABLE OF CONTENTS

PREFACE

This volume comprises the Proceedings of the Symposium on Mathematical Pattern Recognition and Image Analysis (MPRIA) held June 1-3, 1983, at the NASA/Johnson Space Center, Houston, Texas.

The Symposium was initiated with a brief Program Overview presented by Drs. Howard G. Hogg, NASA Headquarters, and R. P. Heydorn, NASA/JSC.

The first paper appearing in the Proceedings was prepared by Professor Robert M. Haralick in support of his excellent invited keynote address. The remaining eighteen papers of the Proceedings present the results of various research efforts initiated during FY 1982 as part of NASA's Remote Sensing Research Program. Five of the papers present results from the four research efforts carried out by the following NASA principal investigators:

R. P. Heydorn - NASA/Johnson Space Center

David D. Dow - National Space Technology Laboratories

Manouher Naraghi - Jet Propulsion Laboratory

Daniel N. Held - Jet Propulsion Laboratory

The remaining thirteen papers present results from the eleven research efforts initiated July 16, 1982, under Contract NAS 9-16664 and carried out by the following principal investigators:

H. P. Decell, Jr./B. C. Peters, Jr. - University of Houston

Carl Morris - University of Texas at Austin

L. Schumaker/L. F. Guseman, Jr. - Texas A&M University

K. S. Shanmugan - University of Kansas

E. Parzen/W. B. Smith - Texas A&M University

A. H. Strahler - Hunter College

Waldo Tobler - University of California, Santa Barbara

E. M. Mikhail - Purdue University

Grahame Smith - SRI International

L. Kanal - LNK Corporation

L. S. Davis/A. Rosenfeld - University of Maryland

In an attempt to group presentations of a similar nature, the Symposium was divided into three MATH/STAT SESSIONS and two PATTERN RECOGNITION sessions. This grouping also reflects the topical contents of the MPRIA Technical Workshops on MATH/STAT and PATTERN RECOGNITION held January 27-28, 1983 and February 3-4, 1983, respectively.

The papers appear in the Proceedings in the order in which they were presented at the Symposium. An agenda and a list of attendees who registered for the Symposium are included in the Appendix.

L. F. Guseman, Jr.
Principal Investigator and
MPRIA Program Coordinator
Contract NAS 9-16664

KEYNOTE ADDRESS

RELATIVE ELEVATION DETERMINATION FROM LANDSAT IMAGERY

R.M. Haralick, S. Wang

Dept. of Electrical Engineering and Computer Science

Virginia Polytechnic Institute and State University

ABSTRACT - In LANDSAT imagery, spectral and spatial information can be used to detect the drainage network as well as the relative elevation model in mountainous terrain. To do this, the mixed information of material reflectance and topographic modulation in the original LANDSAT imagery must be first separated. From the material reflectance information, big visible rivers can be detected. From the topographic modulation information, ridges and valleys can be detected and assigned relative elevations. Finally, a relative elevation model can be generated by interpolating values for non-ridge and non-valley pixels.

1. Introduction

It is a common task for a photointerpreter to examine the spatial pattern on an aerial image and by appropriate interpretation be able to tell the elevation of one area relative to another and be able to infer the stream network and the drainage network even though some of the streams may be below the resolution of the sensor. There is a wealth of information in spatial patterns on aerial imagery but most computer data processing of remotely sensed imagery, being limited to pixel spectral characteristics, does not make use of it.

In this paper, we describe a procedure by which a relative elevation model can be infered from a LANDSAT scene of mountainous and hilly terrain. The processing has a number of distinctly different steps. First to appropriately prepare the imagery for processing we must destripe it and perform haze removal. Destriping can be done by the Horn and Woodham [1979] technique. Haze removal can be done by the Switzer, Kowalik and Lyon [1981] technique. These two steps constitute the preprocessing and are not discussed in this paper.

To a first order effect, after preprocessing the cause of the intensity value at any pixel is due to the combined effect of the angle at which the sun illuminates the ground patch corresponding to the pixel and the reflectance of the

surface material on the ground patch. To make sense of the spatial pattern first requires separating these two effects. For this purpose we modify the Eliason, Soderblom and Chavez [1981] technique to create two main images from the LANDSAT imagery. The first image is a reflectance image and the second image is a topographic modulation image which has information related to surface slope and sun illumination. The details of this technique are given in Section 2.

As discussed in Section 3, the reflectance image can be used by the Alfoldi and Munday [1978] procedure for identification of all areas of water. The topographic modulation image can be used to identify the ridges and the valleys. This is discussed in Section 4. With the valleys identified, each valley pixel may be assigned a relative elevation which increases as the valley path from the pixel to the river it empties in increases. Ridges must be assigned elevations higher than their neighboring valleys and each ridge pixel can be assigned a relative elevation which decreases on the ridge path from the pixel to the saddle point where the ridge crosses a valley. The ridge valley elevation assignment procedure is discussed in Section 5. Once ridges and valleys have been located and assigned relative elevations, a complete elevation model can be generated by interpolating values for non-ridge and non-valley pixels. The interpolation procedures are discussed in Section 6.

Since the launch of the first Earth Resources and Technology Satellite (ERTS, later renamed LANDSAT) in July 1972, much work in remote sensing has been done by using pattern analysis and picture processing techniques for image classification, restoration and enhancement. Few people have tried the scene analysis or artificial intelligence approach to describe the image in terms of the properties of objects or regions in the image and the relationships between them. Ehrich [1977] found global lineaments by partitioning the image into windows and applying long, straight linear filters at different orientations in each window to extract local evidence. Dynamic programming [Montanari, 1971; Martelli, 1972] was then used to form complete global lineaments. VanderBrug [1976] tested various detectors to get linear features in satellite imagery. This was only at the local level. Later VanderBrug [1977a] used relaxation to reduce noise in the output. Finally VanderBrug [1977b] defined a merit function that can be used to select pairs of segments to be merged so that local line detector responses can be linked together into a global representation of the curves. His work is closely related to the Shirai [1973] technique which employed sequential line following to find edges in scenes containing polyhedra. Li and Fu [1976] used tree grammars to locate highways and rivers from LANDSAT pictures. The above investigations deal with the extraction of all the linear features from an image, but they do not deal

with the interpretation of these linear features. In the following investigations, knowledge about the desired features are considered crucial in such analyses.

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

Similarly, knowledge is important in our problem which requires analysis both at the local and global levels. Local level analysis will be discussed in Section 2 to 4; global level analysis will be discussed in Section 5 to 6.

2. Illumination model

The brightness and darkness in each band of LANDSAT images come from two main sources. First, they can be due to material properties. For example, in the spectral region (.8 - 1.1 μm) of band 7, water bodies absorb infrared radiation, so they appear as clearly delineated dark bodies; living vegetation reflects strongly in this portion of the infrared, so areas of living green vegetation appear as bright regions. Second, they may be due to topography and sun illumination angle effects. The mountain side facing to the sun appears as a bright region; the mountain side facing away from the sun may appear as a shadow or dark region. Unfortunately, the LANDSAT data values are some combination of these two effects. Eliason, Soderblom, and Chavez [1981] address this problem by defining an illumination model involving material reflectance and topographic modulation images. In the following, we will introduce a modified Lambertian model in which the information of diffuse light and shadows is also included.

For a pixel (x,y) which receives sunlight, the original LANDSAT image G measuring the amount of reflected light at band b is

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

where r is the surface reflectance, I is illumination flux, θ is the angle between sun incidence direction and surface normal, D is diffuse light, and H is the haze due to atmospheric scattering. On the other hand, for a pixel (x,y) in shadow, G is simply

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

After the haze H(b) is removed by the Switzer, Kowalick and Lyon [1981] technique, for pixels receiving sunlight, the ratio image of bands b_1 and b_2 is

$$\frac{G'(x,y,b_1)}{G'(x,y,b_2)} = \frac{G(x,y,b_1) - H(b_1)}{G(x,y,b_2) - H(b_2)}$$

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

$$= \alpha \frac{r(x,y,b_1)}{r(x,y,b_2)}$$

if we assume $I(b_1) = \alpha I(b_2)$ and $D(b_1) = \alpha D(b_2)$.

Similarly, for pixels in shadows,

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

In either case, the ratio is independent of $\cos\theta$. Thus, by clustering using different ratio images as features, the pixels grouped in one cluster should belong to the same ma-

terial reflectance group. The result is called a reflectance cluster image.

A window of 4-bands LANDSAT scene is shown in Figure 1. The image was taken in April 1976 over areas in Nicholas County, West Virginia and neighboring counties. The ratio images of 5/4, 6/5, 7/6 are shown in Figure 2, and the reflectance cluster image is shown in Figure 3.

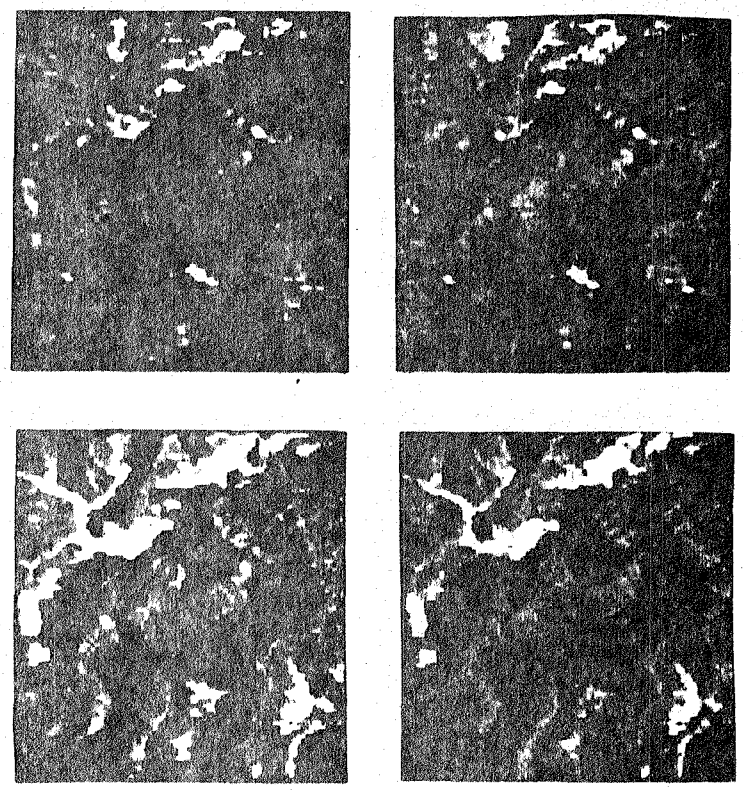


Figure 1 - 4- bands LANDSAT scene in W. Va.

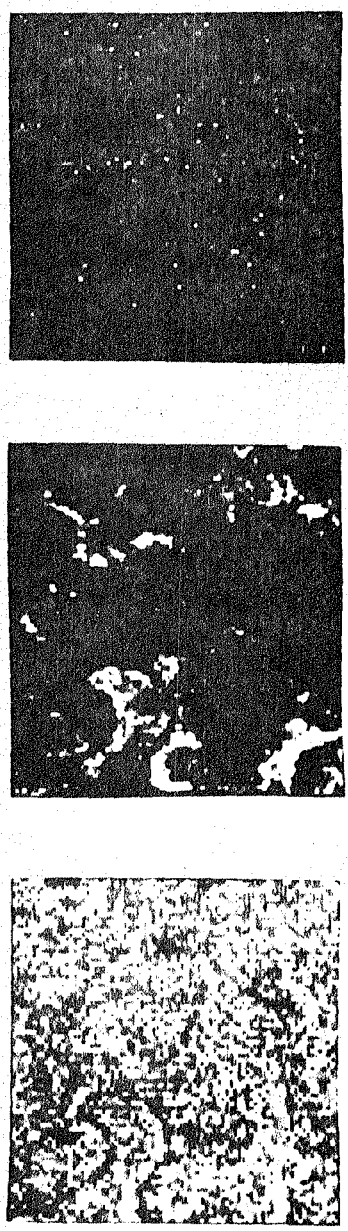


Figure 2 - Ratio images of 5/4, 6/5 and 7/6

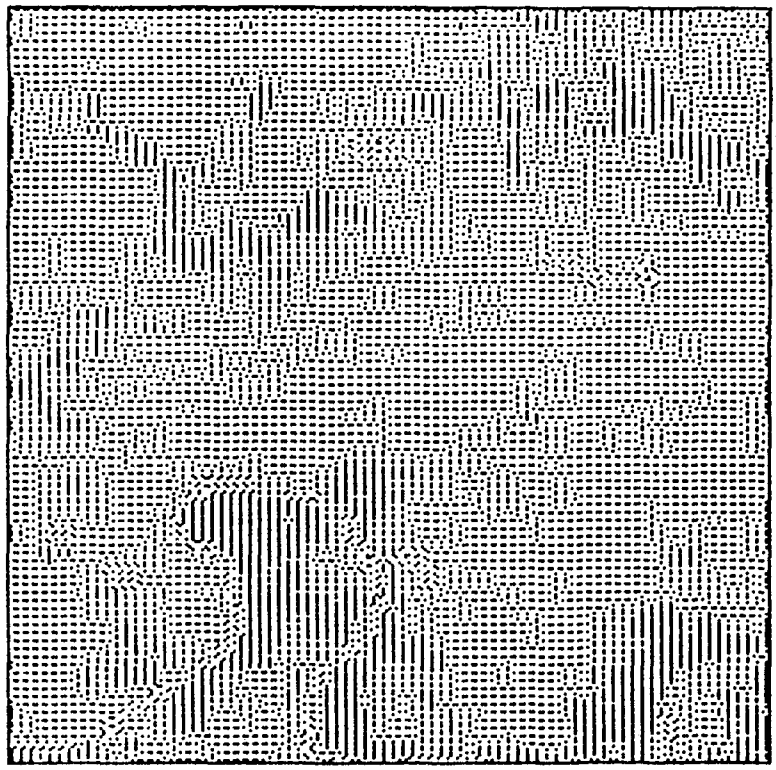


Figure 3 - Reflectance cluster image.

The reflectance cluster image is a function

$$R_c: \quad X \times Y \rightarrow \{1,2, ---, N_c\}$$

where X is the set of row coordinates, Y is the set of column coordinates, and N_c is the total number of clusters. Each reflectance cluster cl is a subset of pixels defined by

$$C(c1) = \{(x,y) \mid R_c(x,y) = c1\},\ 1 \leq c1 \leq N_c$$

The pixels in each C(c1) do not have identical gray tone intensities in the dehazed G' image. This is due to the fact that some pixels are directly lit and others are in shadow. By performing a second clustering on G' within each C(c1), we can split each C(c1) into a bright sub-cluster $C_0(c1)$ consisting of directly lit pixels and a dark sub-cluster $C_1(c1)$ consisting of pixels in shadow. A binary shadow image Sw can be defined by

$$Sw: X \times Y \rightarrow \{0,1\},$$

$$Sw(x,y) = \begin{cases} 0 \text{ if } (x,y) \in C_0(R_c(x,y)) \\ 1 \text{ if } (x,y) \in C_1(R_c(x,y)). \end{cases}$$

This is shown in Figure 4.

Figure 4 - Binary shadow image.

After the lit and shadowed pixels are identified, we extract a diffuse light image D_f which contains in each pixel (x,y) the value r(x,y,b)D(b), a reflectance image R which contains in each pixel (x,y) the value r(x,y,b)I(b), and a topographic modulation image T_p which contains in each pixel (x,y) the value cosθ(x,y). Thus, for directly lit pixels

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

and for shadowed pixels

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

Since shadowed pixels contain the information of diffuse light only, the mean dehazed G' value of pixels in $C_1(cl)$ can be used to represent the reflected diffuse light information for cluster cl. The diffuse lit image D_f is defined by

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

where $cl = R_c(x,y)$. If the reflectance cluster image were perfect, we would have

Assumption 1 : r(x,y,b) is a constant $\bar{r}(cl,b)$ for all (x,y) in C(cl) with $cl = R_c(x,y)$.

In this case,

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

$$= \bar{r}(cl,b)\ D(b)$$

Since directly lit pixels contain the information of diffuse light as well as direct sun illumination, the mean $G' - D_f$ value of pixels in $C_0(c1)$ can be used to represent the reflected sun illumination information for cluster c1. If pixel (x,y) is in reflectance cluster c1, that is, if $R_c(x,y) = c1$, then the reflectance image R can be defined by

$$R(x,y,b) = \sum_{(u,v)\ \varepsilon\ C_0(c1)} \frac{G'(u,v,b) - D_f(u,v,b)}{\#\ C_0(c1)}$$

$$= \bar{r}(c1,b)\ I(b) \sum_{C_0(c1)} \frac{\cos\theta(u,v)}{\#\ C_0(c1)}$$

$$= \bar{r}(c1,b)\ I(b)\ X_c(c1)$$

where X_c is the spatial average of $\cos\theta$ for pixels in $C_0(c1)$. It is meaningful to look at R image only if we make the following assumption.

Assumption 2: $X_c(c1)$ takes the same value X_c for all reflectance clusters.

Finally, from equation (*),

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

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

which contains the information about the cosine of the angle between the surface normal and the illumination direction. The D_f, R, and T_p images for Figure 1 are shown in Figure 5,6, and 7.

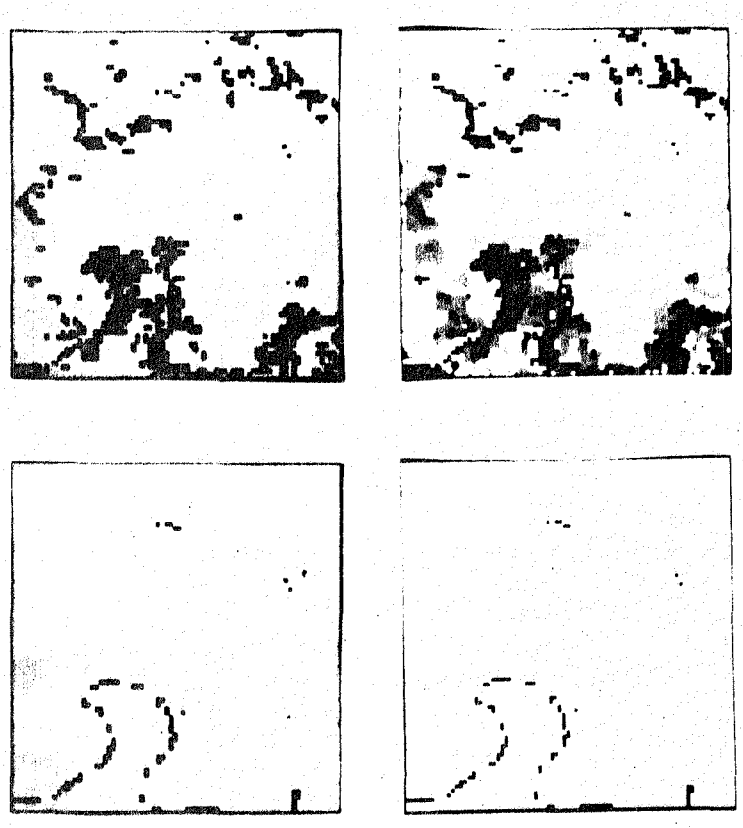


Figure 5 - Diffuse light image.

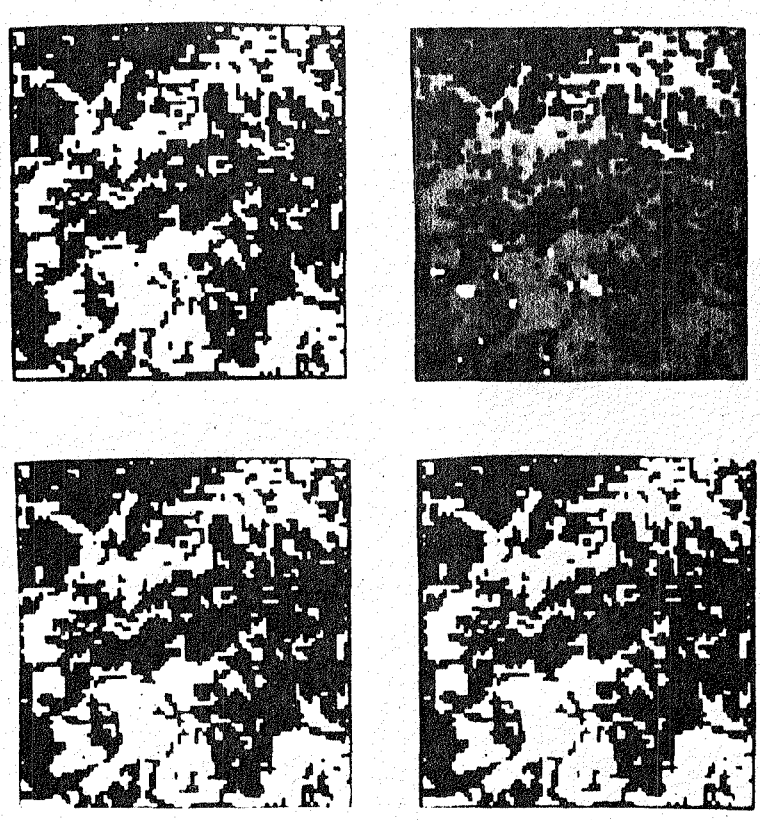


Figure 6 - Reflectance image.

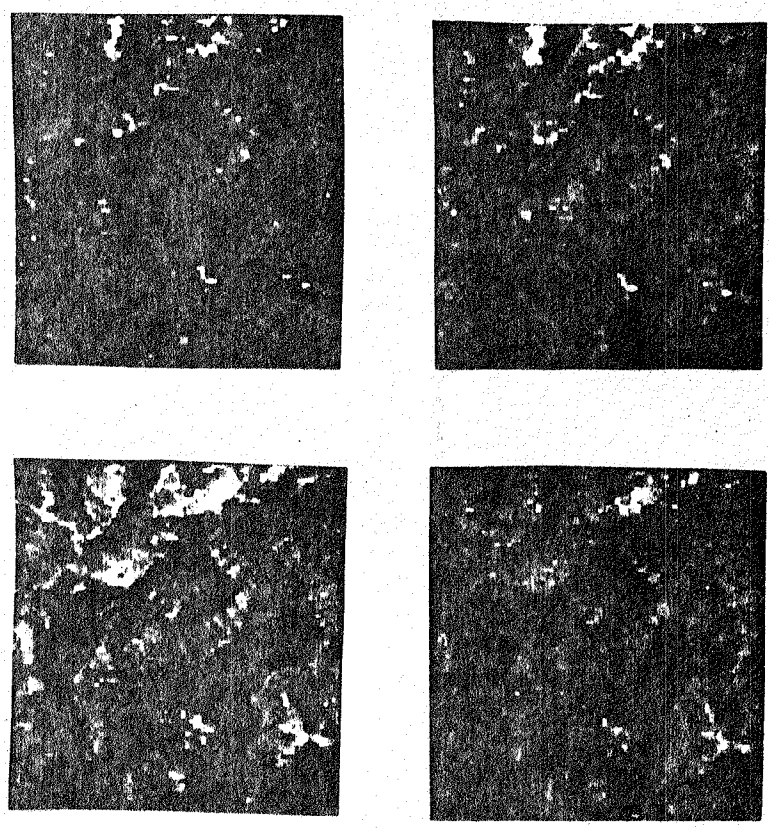


Figure 7 - Topographic modulation image.

3. Detection of Visible Rivers

Visible river detection can play an important part in generating an elevation model since elevations away from the river must increase. Visible rivers can be detected using the material reflectance image created by the technique discussed in the last section. In the spectral region (.8 - 1.1 μm) of band 7, water bodies absorb infrared radiation, so visible rivers appear as dark curves, and lakes appear as dark regions. In the material reflectance image of band 7, these dark features become more clear because shadows are removed. However, not all dark features are water bodies; the real water bodies can be identified by the following process [Alfoldi and Munday, 1978].

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

(2) In this coordinate system, Munday [1974] has determined a curve (Figure 8) which is the locus of the positions of chromaticity values of water bodies. If, for some pixels, the x, y values calculated in 1 are close to this curve, then those pixels can be identified as portions of water bodies.

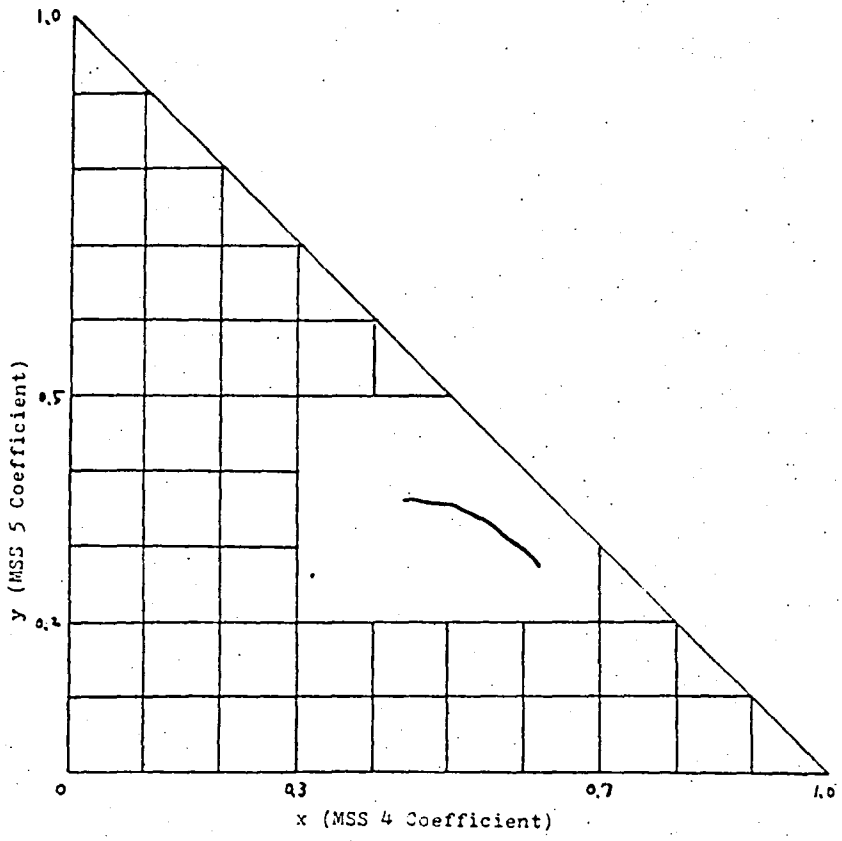


Figure 8 - Chromaticity plot.

4. Ridge-Valley Detection

In this section, we describe how to extract shadowed and bright areas, create linear features on the borders between these areas, and then classify these linear features into ridge and valley segments. In the next two sections, we discuss how to generate a relative elevation model.

From the shadow image of Figure 4, we can get the connected components of bright and shadowed regions. Because valleys and ridges exist on the borders between these regions, the perimeters of these bright and shadowed regions are segmented into border segments according to their left regions, right regions, and orientations. A border segment is a maximally long sequence of connected pixels which are on the border between two given regions. Because the detection of ridges and valleys is highly orientation-dependent and the sun illumination comes from east in Figure 1, each border segment is further broken into several pieces according to orientation: all the east-west parts can be separated from the north-south parts. The result is shown in Figure 9.

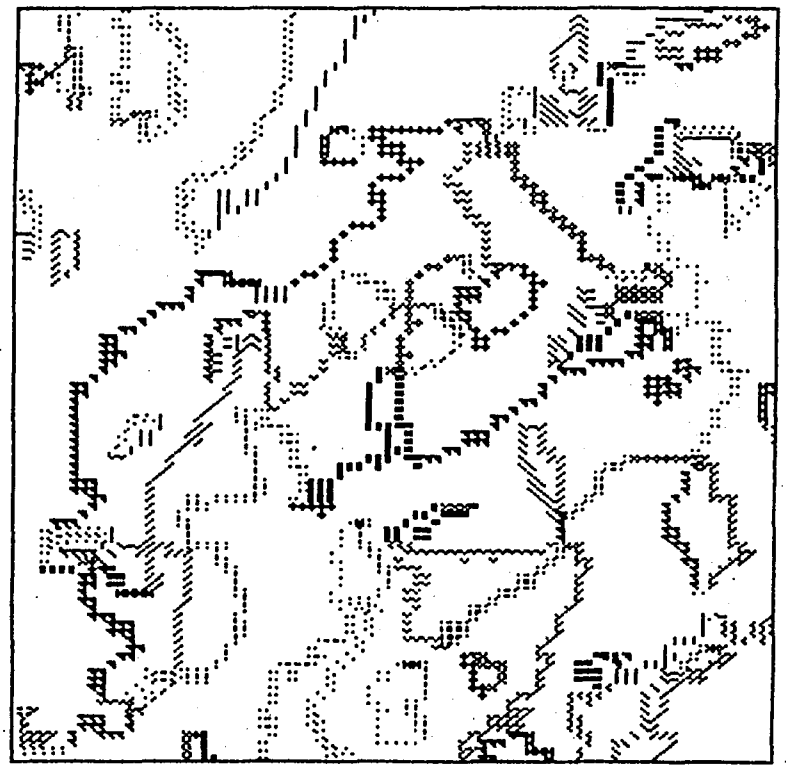


Figure 9 - Border Segments

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

East-west region boundaries are classified according to the labeling of neighboring north-south boundaries as well as their orientation relative to the east-west boundaries. As shown in Figure 10, east-west boundaries have the same labeling of the north-south boundary which makes the angle between them larger. The results of ridge-valley finding are shown in Figure 11. Assignment of relative elevation to ridge and valley is discussed in the next section.

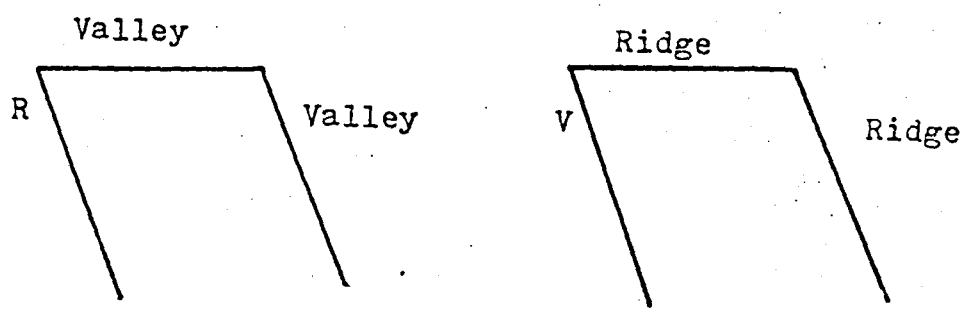


Figure 10 - Classifying east-west border segments.

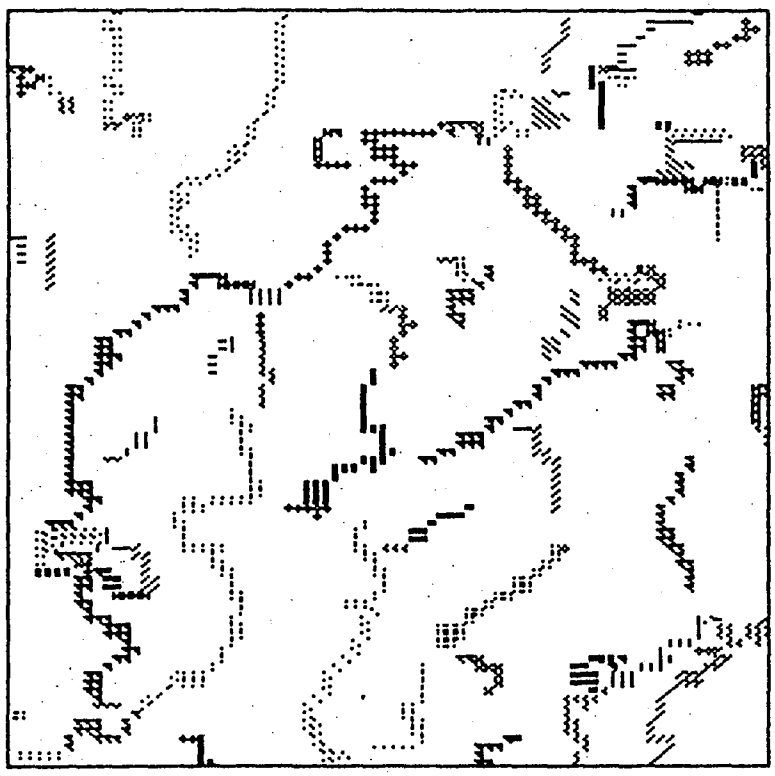


Figure 11 - a. Valley map consisting of the border segments which are identified as valleys.

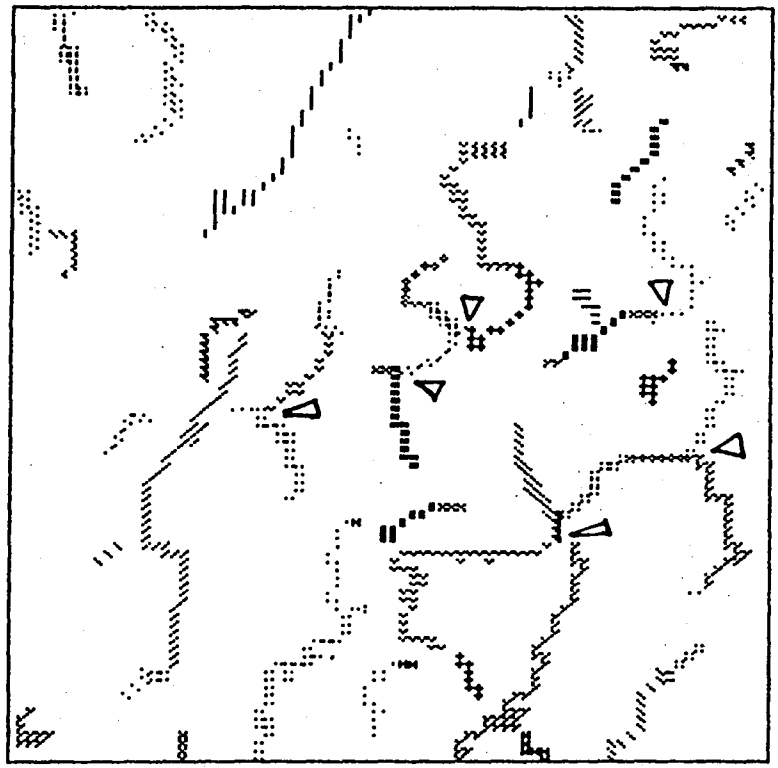


Figure 11 - b. Ridge map consisting of the border segments which are identified as ridges.

5. Relative Elevations of Ridge and Valley Segments

In this section, we describe how to estimate the relative elevations of the ridges and valleys. First we will describe a model which can do the elevation assignment job, then we will give the equations of elevation assignment.

Assuming that we have a stream network in a mountainous area, and we know where the biggest rivers are, we can trace the network, starting from the biggest rivers, to find the flow directions of all the stream segments because water always flows from higher locations to lower locations. In other words, if the valley segments detected in the last section formed a network, then starting from the visible rivers detected in Section 3, we can trace the network and assign relative elevations to all the segments. Unfortunately, the observed valley segments do not form a network; there are many gaps. As shown in Figure 12, if it is dark on the right and bright on the left of stream Vb, then Vg cannot be detected due to the shadow on the right of Vb, and a gap exists between Vb and a smaller stream Vs.

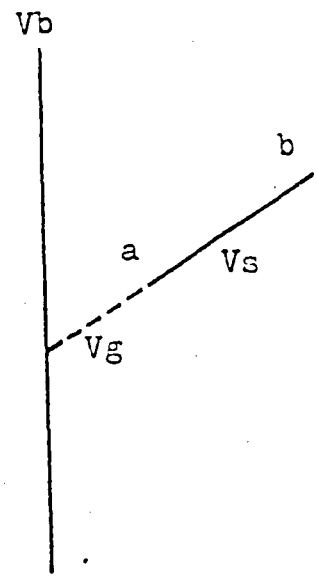


Figure 12 - The gap between a smaller and a larger stream.

The knowledge that the cross-sections of valleys are V-shaped can be used to bridge the gaps. If one looks at topographic maps, the elevation contours of valleys such as in Figure 13 can be frequently found. Thus, if one draws a line ab perpendicular to the valley Va, the elevations are

increasing from point o to point a , and also from point o to point b. However, if a ridge point is encountered during the process, the increasing has to stop because the elevation starts to decrease. Thus the route of growth is directed both by the valleys and by the ridges, in other words, by global information.

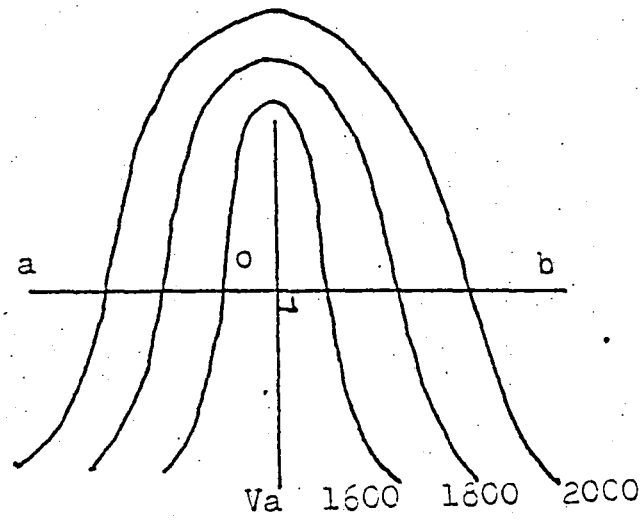


Figure 13 - The elevation pattern of valleys and its relation to elevation growing

Applying this idea to Figure 12 and assuming that growing propagates away from valley segment Vb, the end a of valley segment Vs will be touched first by this growing, and it is

deduced that end b of Vs must be higher than end a. This is the basic idea for determining the higher-lower ends of all the valley segments. The elevations of all the points in one segment can be calculated if we know its slope. On the other hand, ridges get elevations when the growing stops at them. Now, we will give the simple equations of elevation assignment.

Our elevation growing model simply assumes that elevation increases monotonically from valleys to ridges or along valley segments from rivers to the saddles where a valley crosses a ridge. It can be used for assigning initial relative elevations to each pixel. Because no attempt is made to realistically account for the topographic shape of the hillsides from the valley to the ridge, the initial relative elevations will be more accurate for the ridge or valley labeled pixels than the non-ridge and non-valley labeled pixels. Section 6 discusses a more realistic procedure for hillside elevation estimation using the ridge valley elevations calculated in this section.

There are two ways a pixel can get assigned an elevation depending on whether the pixel belongs to a valley segment or whether the pixel does not belong to a valley segment. Let U be the set of valley segments. Two slopes are associated with each valley segment Vs in U: Sv(Vs) and Sp(Vs).

Sv(Vs) is the slope along Vs itself. Sp(Vs) is the slope of lines outside of Vs and perpendicular to Vs.

The elevation growing model constructs the elevation function El: Zr X Zc -> Ip, where Zr is the set of row coordinates, Zc is the set of column coordinates, and Ip is the set of zero and positive integers. If p is a pixel belonging to a valley segment Vs and pl is the lower end pixel identified as in Figure 12, then

$$El(p) = El(pl) + Sv(Vs) * Dist(p, pl)$$

where Dist is the Euclidean distance between two pixels.

If p does not belong to any valley segment, and its elevation is originated from pixel pr of valley segment Vs, then

$$El(p) = El(pr) + Sp(Vs) * Dist(p, pr).$$

In a small area, one can assume the elevations of visible rivers are lowest. Assigning some initial elevation values to the pixels of the valley segments classified as visible rivers, the elevations of all the other pixels in the image window can be related to the initial elevations of visible river segments by repeatedly using the above two equations. The relative heights of valley segments created by elevation growing model are indicated by arrows in Figure 14, and the ground truth is shown in Figure 15.

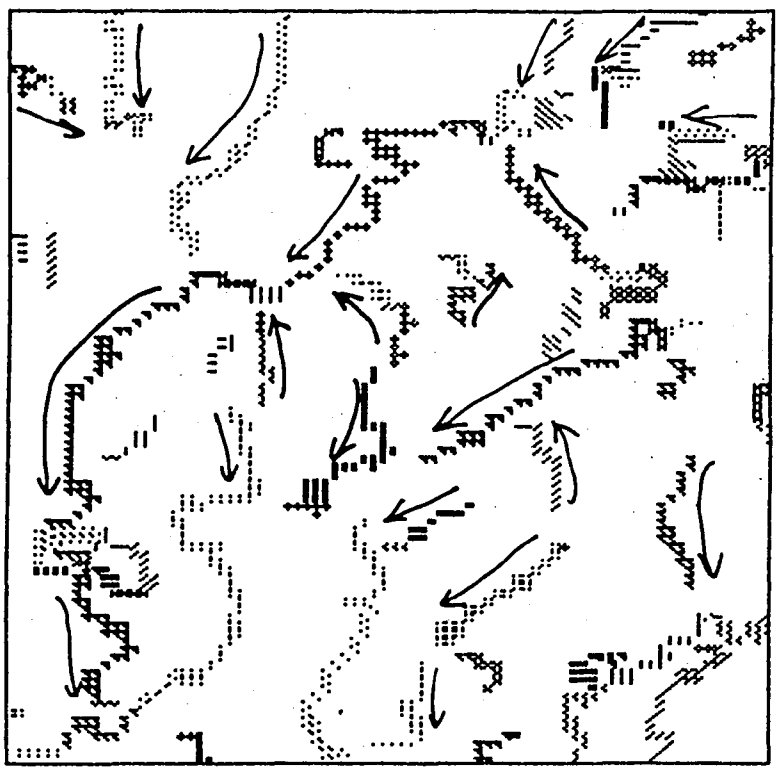


Figure 14 - Relative elevations of valley segments. The arrow are from high ends to low ends.

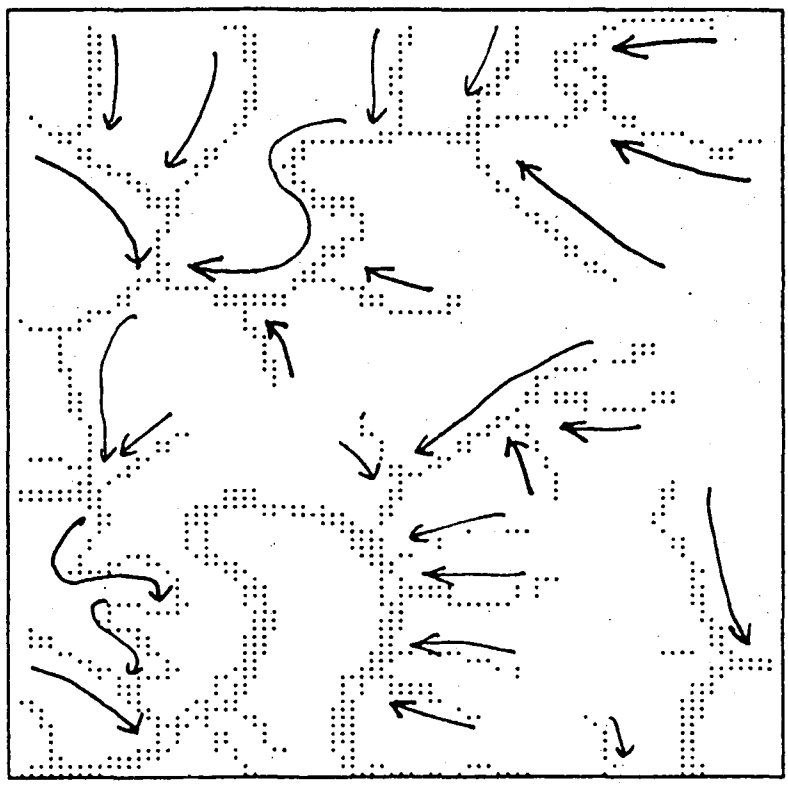


Figure 15 - Stream map created from ground truth.

5.1 Identification of Peak Junctions

When several valleys and ridges point toward a junction, very often this juction is a peak (peak at junction). The peak itself is formed by the junction of several ridges that radiate outward from the peak. (The idealized situation represented in Figure 16 shows four symetrically oriented ridges; in our area, real peaks are often formed by junctions of two or three ridges.) Ridges of course are separated by valleys, so the higher tips of valley segments tend to point toward peaks. The ridge segments intersect to form a peak, whereas valley segments tend to point towards peaks, without actually joining. In this subsection, we discuss the criteria which can be used to identify peak junctions.

Because ridge segments are the major features of peaks, we make the constraint that the number of ridge segments at a junction is larger than the number of valley segments. For many situations, it seems reasonable to relate the heights of peaks to the lengths of ridges that form the peaks. For our class of topographic forms (for example), it is unlikely that very high peaks can be formed by the intersection of very short ridges. As a result, to exclude very low peaks and false peaks from consideration, we impose a rather arbitrary constraint upon definitions of peaks. Currently, we define a peak junction as a junction composed of

four border segments, with the number of its ridge segments larger than the number of valley segments, and the length of its longest ridge segment longer than 800 meters. The peaks thus located in Figure 1 are marked as triangles in Figure 11.b. The correspondence between this result and the topographical map is suprisingly good.

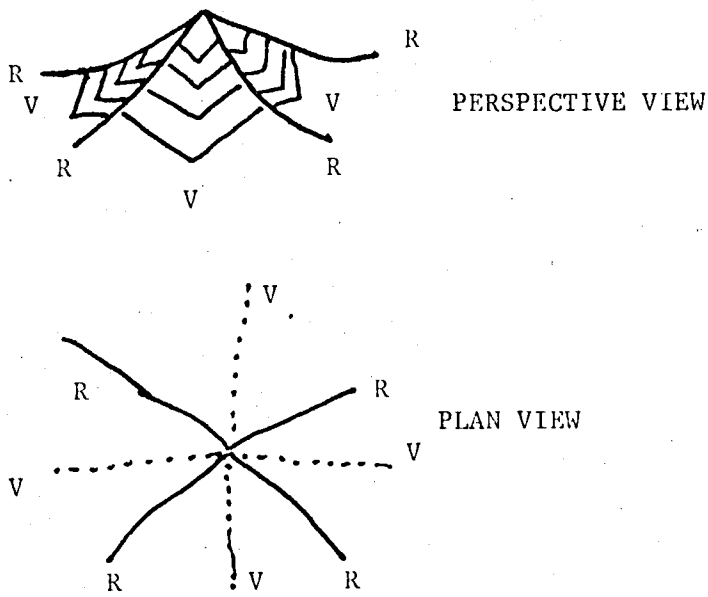


Figure 16 - Idealized relationships between peaks, valleys, ridges.

6. Interpolating Between Ridges and Valleys

In the last section all pixels were assigned elevations, but because realistic shape of the hillsides from valleys to ridges were not taken into account, only the relative elevations of the ridges and valleys are held to be accurate. In this section we describe a few interpolation procedures which permit more realistic elevation assignment to non-valley and non-ridge pixels.

The first interpolating surface has the given elevation values at ridges and valleys and has a 3 X 3 digital Laplacian of zero at all non-ridge and non-valley pixels. This will be referred to as the Laplacian surface. The system of linear equations which this constraint gives rise to can be written as

$$A\,x = b.$$

The vector x is the solution and represents the values to be assigned to each 'variable' (non-ridge non-valley) pixel in the elevation model. The A matrix is defined by applying the digital Laplacian mask operator (Figure 17) to each variable pixel. A mask operator is applied to a pixel by placing the mask over the image so that the central (large positive) mask value is directly over the pixel whose value is to be computed. The pixel value is changed to make the sum of the mask values times the corresponding image values under them equal to zero. For the Laplacian surface only,

Neumann boundary conditions are enforced along the outside rows and columns of the elevation model image. That is, the outer-most row or column is repeated so that the mask operator can be applied to the outside pixels. There is one row in A for each variable pixel in the elevation model and one coefficient value in that row for each variable. A is a sparse matrix since no variable is constrained by more than four other variables (due to the definition of the digital Laplacian mask operator). The b vector is the right hand side of each of the linear equations in the system. The constants on the left hand side of each equation (that result from applying the Laplacian operator to a variable pixel that has a known pixel 4-adjacent to it) are carried to the right hand side and appear in b. For equations representing variable pixels not 4-adjacent to known pixels, the corresponding b element is zero.

$$\begin{matrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{matrix}$$

Figure 17 - A digital Laplacian mask

The second interpolating surface has the given boundary values and minimizes the quadratic variation of the result-

ing surface [Grimson, 1981]. The boundary conditions with which the surface must agree are depth values along the zero-crossings. If the surface elevation function is E and subscripts denote partial differentiation, then the final surface E minimizes

$$\iint (E^2_{xx} + 2\ E^2_{xy} + E^2_{yy})\ d_x\ d_y$$

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

$$Q\ x = b.$$

The x and b vectors have the same meaning as in the Laplacian case and are constructed similarly. The Q matrix is likewise similar to the A matrix of the Laplacian. Instead of using Neumann boundary conditions at the edge of the image, the quadratic variation surface is defined by using special masks to fit the rows and columns near the outside edges. The six masks (Figure 18) are rotated as necessary and applied to the only appropriate variable pixels of the elevation image to define Q. Mask two is applied to corner pixels, mask three is applied to pixels in the outside row or column that are adjacent to a corner pixel, mask four is applied to other pixels in the outside rows and columns,

mask five is applied to pixels in the next-to-the outside row and columns that are 8-adjacent to corner pixels, mask six is applied to other pixels in the next to the outside rows and columns, and mask 1 is applied to all other variable pixels in the image.

$$\begin{array}{ccccc} & & 2 & & \\ & 4 & -16 & 4 & \\ 2 & -16 & 40 & -16 & 2 \\ & 4 & -16 & 4 & \\ & & 2 & & \end{array}$$

(1)

$$\begin{array}{ccc} & 2 & \\ -8 & & \\ 8 & -8 & 2 \end{array}$$

(2)

$$\begin{array}{ccccc} & & 2 & & \\ & 4 & -12 & 4 & \\ -8 & & 20 & -12 & 2 \end{array}$$

(3)

$$\begin{array}{ccccc} & & 2 & & \\ & 4 & -12 & 4 & \\ 2 & -12 & 22 & -12 & 2 \end{array}$$

(4)

$$\begin{array}{ccccc} & & 2 & & \\ & 4 & -16 & 4 & \\ -12 & & 36 & -16 & 2 \\ & 4 & -12 & 4 & \end{array}$$

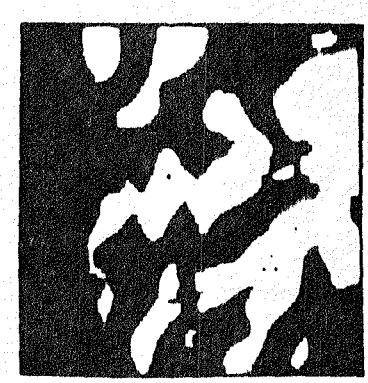
(5)

$$\begin{array}{ccccc} & & 2 & & \\ & 4 & -16 & 4 & \\ 2 & -16 & 38 & -16 & 2 \\ & 4 & -12 & 4 & \end{array}$$

(6)

Figure 18 - Six masks for the quadratic variation method.

The third kind of interpolation surfaces can be created without using any mask. For each non-boundary pixel, we can first find its distances to the nearest valley pixels and nearest ridge pixels. From these distances and the elevations at these nearest valley pixel and nearest ridge pixel, either a linear, cubic, or fifth order fit interpolation can be used to calculate the elevation of this non-boundary pixel. If cubic fit is used, the first order derivative is zero at ridge and valley pixels. If fifth order fit is used, both the first and second order derivatives are zero at ridge and valley pixels. The resulting images with higher brightness indicating higher elevation and the corresponding surface plots are shown in Figure 19. The image and surface plot of the elevations read from digital terrain tape [NCIC, 1980] for this area are shown in Figure 20. The reconstructed LANDSAT images by using diffuse light image (Figure 5), reflectance image (Figure 6), elevation model (Figure 19a), and an artificial sun at specified azimuth and elevation angles are shown in Figure 21. They are reasonable reconstructions.



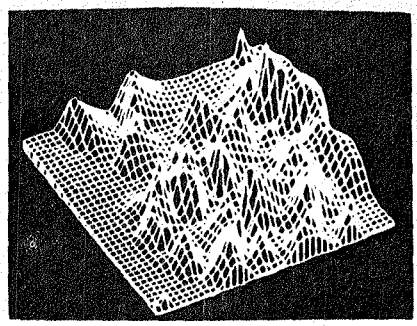
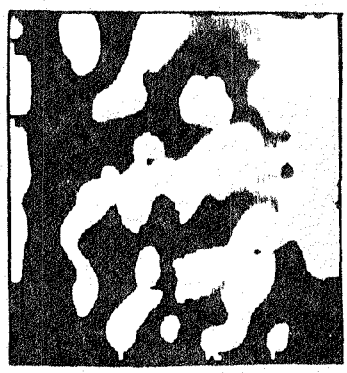


Figure 19a. Elevation Model by Method 1, Laplacian Mask



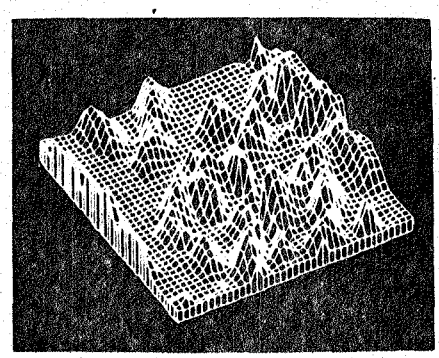
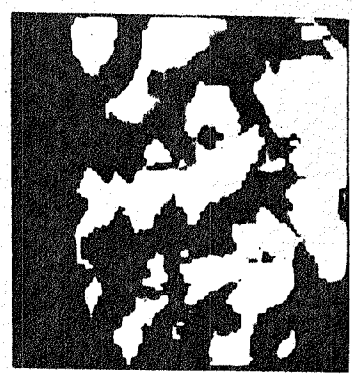


Figure 19b. Elevation Model by Method 2, Quadratic variation



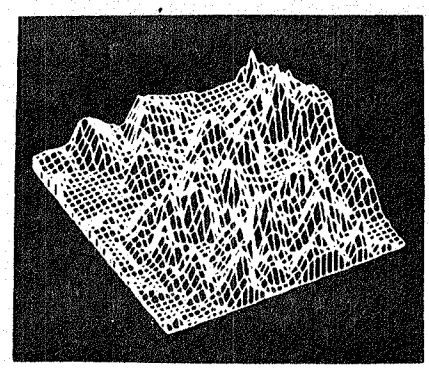


Figure 19c. Elevation Model by Method 3, Linear fit

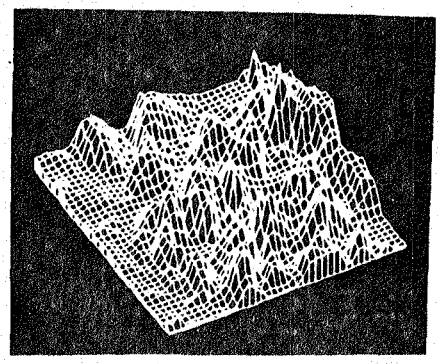
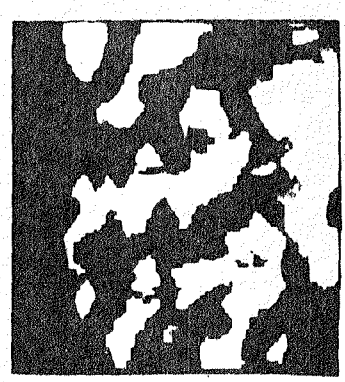


Figure 19d. Elevation Model by Method 3, Cubic fit



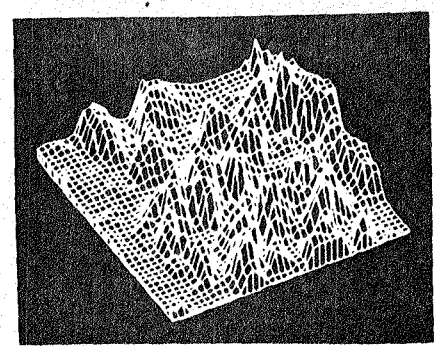


Figure 19e. Elevation Model by Method 3, Fifth order fit

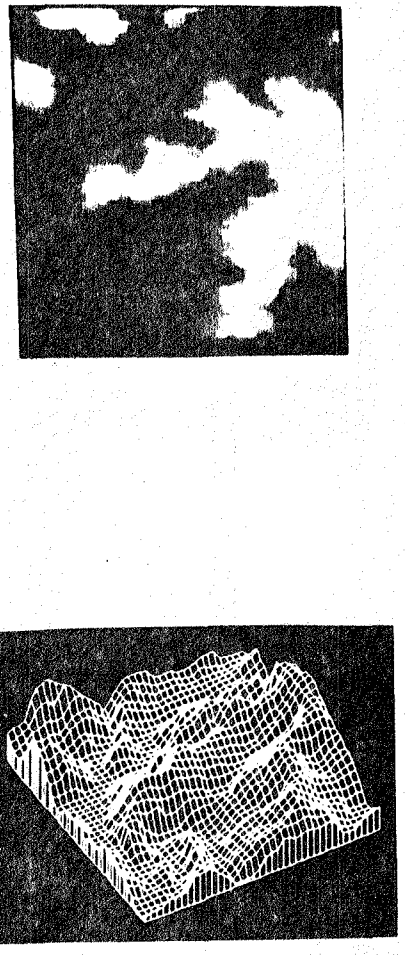


Figure 20 – Elevation model from digital terrain tape.

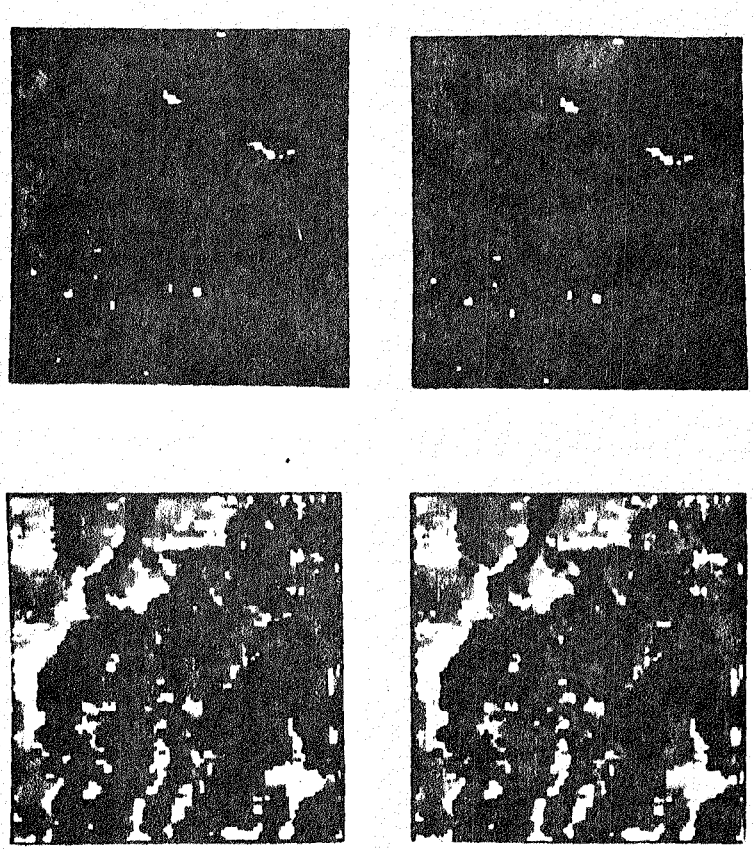


Figure 21 - Reconstructed LANDSAT imagery

7. Conclusion

In order to reconstruct 3D spatial information from LANDSAT imagery, we need to identify shadowed and directly lit pixels as well as local slope information. A model involving reflectance, topography, diffuse light, and haze has been discussed and a technique for computing this information has been given. The shadow reflectance, and elevation images look quite good by comparing with the topographic map of the same area and our understanding of the vegetation surface cover.

Once the shadow image and local slope information is determined, ridge and valley segments are detected and then an elevation growing model is used to assign relative elevations to them. Interpolation generates surface elevation at all locations from the known values at ridge and valley segements.

Acknowledgement

The authors want to thank J. Campbell, R.W. Ehrich, and L.G. Shapiro for their helpful suggestions and discussions.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Papers Presented by

Investigators From the

Remote Sensing Research Program

MPRIA

ESTIMATING LOCATION PARAMETERS IN A MIXTURE MODEL

Richard P. Heydorn and Rehka Basu
NASA Johnson Space Center
Houston, Texas

ABSTRACT

This paper considers mixture models of the form

$$h = \sum_{j=1}^{M} \lambda_j f_{\theta_j}^{j}$$

where θ_j is a translation parameter. An approach is discussed which makes use of a Caratheodory theorem on the trigonometric moment problem to determine M and θ_j, j=1,2,...,M. This theorem is also applied to show that translates of many common distributions lead to identifiable mixtures.

INTRODUCTION

Let $F = \{f_\xi;\ \xi \in \mathbb{R}^N\}$ be a family of probability density functions and let G be a distribution function on $\mathbb{R}^N$ where $\mathbb{R}^N$ is the set of real vectors of dimension N. For the given G we define a mixture density h as

$$h = \int f_\xi dG(\xi) \tag{1}$$

Since all the members of F are used in this definition, it makes sense to say that according to equation (1) F defines a mapping, say $\tilde{F}$, from the set of all G-distributions, say G, to the set of all induced h-densities, say H. If $\tilde{F}: G \to H$ is one-to-one and onto then we say that H is identifiable. This formulation is essentially due to Teicher [1]. Thus, identifiability implies that, for a given mixture density h, a knowledge of the family F will allow us to uniquely determine G. This has practical implications for estimating the proportion of a material class on the ground using remotely sensed observations of that material. To illustrate the point, we offer the following example.

Suppose we are given spectral measurements, x, of points (pixels) on the ground which have been obtained from a satellite-multispectral scanner system. We imagine that these x's are observations on some random variable X distributed according to density h. Suppose that through experimentation we have found that any given material class on the ground gives rise to measurements that are normally distributed and that in a given region the mixture model that applies is:

$$h(x) = \sum_{j=1}^{M} \lambda_j \frac{1}{\sqrt{2\pi\sigma^2}} e_j^{-(1/2)(x-\mu_j)^2/(\sigma^2)} \tag{2}$$

With reference to equation (1) we see that in this example G assigns a point probability λ_j to the points (μ_j, σ), $j=1,2,\ldots,M$. This is an example of a finite mixture model. Since the M material classes are associated with the parameters (μ_j, σ), $j=1,2,\ldots,M$, λ_j can be considered as the a-priori probability of observing the j-th class or λ_j is the proportion of the j-th class present in the given region. The primary aim is to determine the λ_j-values but to do that one has to estimate M, μ_j, σ, $j=1,2,\ldots,M$. Studies within the AgRISTARS program suggest that a multivariate version of the model given in equation (2) fits reasonably well to agricultural data, as well as to data from natural vegetative classes, c.f., Lennington et al. [2] . In those studies maximum likelihood estimation methods were used to estimate the λ_j's, the means, and the covariances. The number of classes, M, was determined by applying a heuristically derived algorithm.

In this paper we consider a finite mixture model of the form

$$h = \sum_{j=1}^{M} \lambda_j \, f^j_{\Theta_j} \tag{3}$$

where Θ_j is a location parameter and $f^j_{\Theta_j}$ may depend upon other parameters (this is the reason for using the superscript "j") in addition to Θ_j. In the simplest case we have the pure translation family, $F_f = \{f_\Theta : \Theta \in \mathbb{R}\}$ where each member is a translate of some given f. The model in equation (2) is a specific example.

Our approach will make use of a theorem of Caratheodory on a trigonometric moment problem as discussed in Grenander and Szegö [3]. Of particular interest will be the constructive proof (due to Szegö)

which provides a means for computing M and Θ_j, j=1,2,...,M in equation (3). We begin by discussing the pure translation case. For that case it is possible to compute the proportions λ_j in addition to M and Θ_j for j=1,2,...,M. Since in the more general case each f_θ^j can depend upon more than just a location parameter our methods do not lead to values for the λ_j's. However, for certain families of densities knowing M and each Θ_j may simplify the estimation of these other parameters (e.g., see Redner [4]).

THE PURE TRANSLATION CASE

For $f_{\Theta_j} \in F_f$ let

$$h = \sum_{j=1}^{M} \lambda_j f_{\Theta_j} \tag{4}$$

Since f is a density with a characteristic function F, the characteristic function of h is [note: in this paper ω is in radians]

$$H(\omega) = \sum_{j=1}^{M} \lambda_j F(\omega) e^{i\omega\Theta_j}$$

For any ω that is not a zero of F,

$$H(\omega)/F(\omega) = \sum_{j=1}^{M} \lambda_j e^{i\omega\Theta_j} \tag{5}$$

The following theorem due to Caratheodory applies to the form given by equation (5).

THEOREM 1

Let $c_1, c_2, \ldots, c_n$ be complex constants where $c_\nu \neq 0$ for some ν. There exists an integer M, $1 \leq M \leq n$ and constants λ_j, $e^{i\Theta_j}$ such that each λ_j is real and positive and $\Theta_k \neq \Theta_j$, $k \neq j$ and

$$c_\nu = \sum_{j=1}^{M} \lambda_j e^{i\nu\Theta_j} \quad , \qquad \nu = 1,2,\ldots,n$$

where M, λ_j, and Θ_j are unique.

For a proof see Grenander and Szegö [3] pages 56 to 61.

COROLLARY 1

F_f leads to an identifiable mixture.

PROOF:

Since $H(\omega)/F(\omega) = \sum_{j=1}^{M} \lambda_j e^{i\omega\Theta_j}$, this representation must be unique by Theorem 1.

This corollary, which is an immediate consequence of the Caratheodory theorem, was also proved in a different manner by Yakowitz and Spragins [5]. We now consider the determination of M and Θ_j, $j=1,2,\ldots,M$ by methods developed by Szegö [3].

Since F is (uniformly) continuous and F(0)=1, there exists an interval about $\omega=0$ for which the magnitude of F is positive. Let $(-b,b)$ be the largest such interval and for $k=0,1,2\ldots,n$ let $\omega_k = k\beta$ where

$$\beta = \min\left(\frac{2\pi}{(n+1)\max|\Theta_j|} \ , \ \frac{b}{n+1}\right)$$

For these choices of ω_k, let $C_k = H(\omega_k)/F(\omega_k)$, $C_{-k} = \overline{(H(\omega_k)/F(\omega_k))}$ and consider the Hermitian matrix

$$\Phi = \begin{pmatrix} 1 & C_1 & C_2 & \cdots & C_n \\ C_{-1} & 1 & C_1 & \cdots & C_{n-1} \\ \vdots & & & & \\ C_{-n} & \cdots & \cdots & \cdots & 1 \end{pmatrix}$$

From (5) with $\omega_k = k\beta$.

$$\Phi = \sum_{j=1}^{M} \lambda_j \begin{pmatrix} 1 \\ e^{-i\beta\Theta_j} \\ \vdots \\ e^{-in\beta\Theta_j} \end{pmatrix} \quad (1, e^{i\beta\Theta_j}, \ldots, e^{in\beta\Theta_j})$$

Thus Φ is a linear combination of M rank one matrices, and since the λ_j are unique the rank of Φ must be M. The Toeplitz form $\bar{v}' \Phi v$ is

$$\bar{v}'\Phi v = \sum_{j=1}^{M} \lambda_j \left| \sum_{k=0}^{M} v_k (e^{i\beta\Theta_j})^k \right|^2$$

Since n>M, there must be at least one zero eigenvalue of Φ. Let v be the corresponding eigenvector, i.e., $\bar{v}'\Phi v=0$. Since $\lambda_j>0$ for j=1,2,...,M the complex polynomial

$$P(z) = \sum_{k=0}^{n} v_k z^k$$

where $z = e^{i\beta\Theta}$, must have roots at $z_j = e^{i\beta\Theta_j}$.

We see therefore that the rank of Φ determines the number of distinct translates and the roots of P(z) are the distinct translations. The

proportions, $\lambda_1, \lambda_2, \ldots, \lambda_n$ can be determined by substituting specific x-values in equation (4) and solving the resulting system of linear equations.

THE GENERAL CASE

We now consider the general form given by equation (3). For this case we choose families of the form

$$F = \{ f_\Theta^\alpha : \Theta \epsilon \mathbb{P}, \alpha \epsilon \mathbb{R}^N \}$$

where $\mathbb{P}$ is the set of rational numbers and $\mathbb{R}^N$ is the set of N-dimensional real vectors. We will show that at least for certain cases, e.g., when f_Θ^α is an exponential, double exponential, gamma, or beta this family leads to unique determination of Θ from a mixture.

Since F is not generated by one function as was F_f, we cannot proceed exactly as we did in the previous section. Our approach for this case will exploit the limiting behavior of $F_0^\alpha(\omega)$ as ω gets large where F_0^α is the characteristic function of f_0^α.

THEOREM 2

Let $h = \sum_{j=1}^{M} \lambda_j f_{\Theta_j}^j$, $f_{\Theta_j}^j \epsilon F$ and let the characteristic function of f_0^j be of the form

$$F_0^j(\omega) = \frac{a_0^j + a_1^j(i\omega) + \ldots + a_p^j(i\omega)^p}{b_0^j + b_1^j(i\omega) + \ldots + b_q^j(i\omega)^q} + o\left(\frac{1}{\omega^{q-p+1}}\right)$$

with $q>p$, $\frac{a_p^j}{b_q^j} > 0$ for each j or $\frac{a_p^j}{b_q^j} < 0$ for each j.

Let $\omega_{kn} = k\beta + 2\pi n$, $\beta = \dfrac{2\pi}{(K+1)\max|\Theta_j|}$, $k=0,1,\ldots,K$, $K>M$, $n=1,2,\ldots$.

a) If there exists a vector v so that for $\Theta_1, \Theta_2, \ldots, \Theta_M$

i) $\sum_{k=1}^{K} v_k e^{i\beta k\Theta_j} = 0$

then

$$\lim_{n\to\infty} \Big| \sum_{k=1}^{K} \sum_{\ell=1}^{K} \overline{v}_k v_\ell (i\omega_{\ell-k,n})^{q-p} H(\omega_{\ell-k,n}) \Big| = 0$$

b) If there exists a vector v so that

ii) $$\lim_{N\to\infty} \sup_{n>N} \Big| \sum_{k=1}^{K} \sum_{\ell=1}^{K} \overline{v}_k v_\ell (i\omega_{\ell-k,n})^{q-p} H(\omega_{\ell-k,n}) \Big| = 0$$

then i) holds for $\Theta_1, \Theta_2, \ldots, \Theta_M$.

Before we prove this theorem, consider the example density functions given in Table 1. We see that the exponential and gamma densities each fit the forms given by Theorem 2. In the case of the gamma density γ can be any positive number but n must be known. In the case of the beta density, $\beta_{m,n}$, whose characteristic function is $B_{m,n}$ notice that $B_{m,n}(\omega) = B_{n,m}(-\omega)e^{i\omega}$. That is, when m and n are reversed the characteristic function can be gotten from the original characteristic function by changing ω to $-\omega$ and multiplying by $e^{i\omega}$. Thus, for example, if we have $\beta_{n,1}(x) = \big((n+2)!/n!\big)x^n(1-x)$ then the leading term in the characteristic function contains $e^{i\omega}$. To make Theorem 2 apply in this case we must multiply $H(\omega)$ by $e^{-i\omega}$ or replace Θ_j by Θ_j+1 for $j=1,2,\ldots,M$. In the case of $\beta_{1,n}(x) = \big((n+2)!/n!\big)x^1(1-x)^n$ Theorem 2 applies directly.

TABLE 1: Characteristic Functions of Some Common Distributions

Function	Characteristic Function
Exponential ($b>0$)	
$f_0(x) = \begin{cases} b\, e^{-bx}, & x>0 \\ 0, & x<0 \end{cases}$	$F_0(\omega) = \dfrac{b}{b-i\omega}$
Double Exponential ($a>0$, $b>0$)	
$f_0(x) = \begin{cases} \dfrac{ab}{a+b} e^{ax}, & x<0 \\ \dfrac{ab}{a+b} e^{-bx}, & x>0 \end{cases}$	$F_0(\omega) = \dfrac{ab}{(a+i\omega)(b-i\omega)}$
Gamma ($\gamma>0$)	
$f_0(x) = \begin{cases} \dfrac{1}{n!\gamma^{n+1}} x^n e^{-x/\gamma}, & x>0 \\ 0, & x<0 \end{cases}$	$F_0(\omega) = \dfrac{1}{(1-i\omega\gamma)^{n+1}}$
Beta ($n>2$)	
$f_0(x) = \begin{cases} \dfrac{(n+2)!}{n!} x^n(1-x), & 0<x<1 \\ 0, & x<0,\ x>1 \end{cases}$	$F_0(\omega) = \dfrac{(n+2)!}{n!}\left[\dfrac{e^{i\omega}}{(i\omega)^2} - \dfrac{2n}{(i\omega)^3} e^{i\omega} + o\left(\dfrac{1}{(\omega)^4}\right)\right]$
Beta ($n>3$)	
$f_0(x) = \begin{cases} \dfrac{(n+3)!}{n!2} x^n(1-x)^2, & 0<x<1 \\ 0, & x<0,\ x>1 \end{cases}$	$F_0(\omega) = \dfrac{(n+3)!}{n!2}\left[\dfrac{2e^{i\omega}}{(i\omega)^3} - \dfrac{6n}{(i\omega)^4} + e^{i\omega} + o\left(\dfrac{1}{(\omega)^5}\right)\right]$

In addition, we could consider a convolution of a given density, f, with members of Table 1, i.e., the family F'' whose members are of the form $f*f_\Theta^\alpha$, $f_\Theta^\alpha \in F$ and f_0^α a member of Table 1 (where "*" denotes convolution). To make this theorem apply we need to modify the theorem slightly by considering $H(\omega)/F(\omega)$ in place of $H(\omega)$ where F is the characteristic function of f.

PROOF OF THEOREM 2:

$$H(\omega) = \sum_{j=1}^{M} \lambda_j F_0^j(\omega) e^{i\omega\Theta_j}$$

and

$$(i\omega)^{q-p} F_0^j(\omega) = \frac{\dfrac{a_0^j}{(i\omega)^p} + \dfrac{a_1^j}{(i\omega)^{p-1}} + \ldots + a_p^j}{\dfrac{b_0^j}{(i\omega)^q} + \dfrac{b_1^j}{(i\omega)^{q-1}} + \ldots + b_q^j} + o\left(\frac{1}{\omega}\right)$$

Thus

$$Q(\omega) \triangleq (i\omega)^{q-p} H(\omega) - \eta(\omega) = \sum_{j=1}^{M} \lambda_j \frac{a_p^j}{b_q^j} e^{i\omega\Theta_j}$$

where

$$\eta(\omega) = \sum_{j=1}^{M} \lambda_j \left((i\omega)^{q-p} F_0^j(\omega) - \frac{a_p^j}{b_q^j}\right) e^{i\omega\Theta_j}$$

Now

$$Q(\omega_{kn}) = \sum_{j=1}^{M} \lambda_j \frac{a_p^j}{b_q^j} e^{i2\pi n\Theta_j} e^{ik\beta\Theta_j} \tag{6}$$

Thus given a vector v that satisfies i)

(7)
$$\left|\sum_{k=1}^{K}\sum_{\ell=1}^{K}\bar{v}_k v_\ell Q(\omega_{\ell-k,n})\right| = \left|\sum_{j=1}^{M}\lambda_j \frac{a_p^j}{b_q^j} e^{i2\pi n\Theta_j}\right|\left|\sum_{k=1}^{K} v_k e^{ik\beta\Theta_j}\right|^2 \Big|$$

and since $\lim_{n\to\infty}|\eta(\omega_{kn})|=0$ and $\sum_{k=1}^{K} v_k e^{ik\beta\Theta_j} = 0$, the assertion in a) holds.

Next consider b). We assume $\frac{a_p^j}{b_q^j} > 0$ for each j; otherwise, if $\frac{a_p^j}{b_q^j} < 0$ for each j we need only multiply equation (6) by -1 before we begin our argument. Given $\varepsilon > 0$ suppose that in equation (7) we can find a vector v that satisfies ii) but

$$\left|\sum_{k=1}^{K} v_k e^{ik\beta\Theta_j}\right| > \varepsilon$$

Since the Θ_j are rational, they are of the form $\Theta_j = \frac{n_{1j}}{n_{2j}}$ where n_{1j}, n_{2j} are integers. Thus, we can choose a subsequence of (n) of the form $(n') = ((\prod_{j=1}^{M} n_{2j})\ell)$, where $\ell=1,2,\ldots\ldots$, so that $2\Pi n'\Theta_j$ is of the form $\pm 2\Pi t$ where t is an integer. Hence $e^{i2\Pi n'\Theta_j} = 1$ for all j which means

$$\left|\sum_{j=1}^{M}\lambda_j \frac{a_p^j}{b_q^j} e^{i2\Pi n'\Theta_j}\right|\left|\sum_{k} v_k e^{ik\beta\Theta_j}\right|^2 > \varepsilon^2 \sum_{j=1}^{M}\lambda_j \frac{a_p^j}{b_q^j} > 0 .$$

But this implies (noting that over (n') all limits exist)

$$0 = \lim_{N\to\infty} \sup_{n>N} \Big| \sum_{k=1}^{K} \sum_{\ell=1}^{K} \bar{v}_k v_\ell Q(\omega_{\ell-k,n}) \Big|$$

$$= \lim_{n'\to\infty} \Big| \sum_{k=1}^{K} \sum_{\ell=1}^{K} \bar{v}_k v_\ell Q(\omega_{\ell-k,n'}) = \varepsilon^2 \sum_{j=1}^{M} \lambda_j \frac{a_p^j}{b_q^j} > 0$$

which is a contraction, and this completes the proof.

Theorem 2 says that any vector vector, v, for which

$$\Big| \sum_k \sum_\ell \bar{v}_k v_\ell (i\omega_{\ell-k,n})^{q-p} H(\omega_{\ell-k,n}) \Big| \stackrel{\Delta}{=} \gamma_n$$

goes to zero as n gets large must be a vector that makes the Toeplitz form $|\sum_k v_k (e^{i\beta\Theta_j})^k|^2$ zero. Hence the theory for finding M and Θ_j, j=1,2,...,M discussed in the previous section applies here for large ω provided we decide on M by looking at eigenvalues γ_n whose magnitudes are small and also account for the fact that $\{H(\omega_{\ell-k,n})\}$ is not necessarily Hermitian, but "approximately so."

We now show that the densities given in Table 1 lead to identifiable mixtures. Recall that in a finite mixture identifiability implies that the representation given by equation (3) is unique. By the methods we have discussed so far, Theorem 1 will only guarantee that M and Θ_j for j=1,2,...,M are unique. The problem here is that the λ_j values and some of the nontranslation parameters appear as products in the limiting form of the characteristic function. Thus, to guarantee identifiability we must consider a subfamily of F in which the nontranslation parameters in the density $f^j_{\Theta_j}$ are in one-to-one correspondence with the translation parameters.

Let F' be a slight modification of the family F. Namely, let

$$F' = \{f_\theta^\alpha : \theta \epsilon P,\ \alpha \epsilon R^p,\ \theta = \theta' \text{ implies } f_\theta^\alpha = f_{\theta'}^{\alpha'}\}$$

Thus F' is the subfamily of F for which no two members can have the same translation value, θ, but still be unequal.

COROLLARY 2

Let $f_o^\alpha \epsilon F$ have a characteristic function as given by Theorem 2.

a) F leads to a unique determination of M and the translation parameters.

b) F' leads to an identifiable mixture.

PROOF:

We assume $\frac{a_p^j}{b_q^j} > 0$, as we did in Theorem 2; otherwise, we consider $-H(\omega)$ in place of $H(\omega)$. Let $h = \sum_{j=1}^{M} \lambda_j f_{\theta_j}^j$. Then, as in equation (6)

$$Q(\omega_{kn'}) = \sum_{j=1}^{M} \lambda_j \frac{a_p^j}{b_q^j} e^{i2\Pi n'\theta_j} e^{ik\beta\theta_j}$$

with $\theta_j = \frac{n_{1j}}{n_{2j}}$, $n' = (\prod_{j=1}^{M} n_{2j})\ell$, $\ell = 1,2,\ldots$. Thus following the proof of Theorem 2)

$$\lim_{n' \to \infty} (i\omega_{k,n'})^{q-p} H(\omega_{kn'}) = \sum_{j=1}^{M} \lambda_j \frac{a_p^j}{b_q^j} e^{ik\beta\theta_j},\ \lambda_j \frac{a_p^j}{b_q^j} > 0.$$

And since the right side of this expression satisfies Theorem 1, the representation is unique. This proves assertion a).

In particular $\lambda_j \frac{a_p^j}{b_q^j}$, $j=1,2,\ldots,M$ are unique. But since the Θ_j, $j=1,2,\ldots,M$ are also unique, it follows from the definition of F' that a_p^j, b_q^j, $j=1,2,\ldots,M$ are unique and therefore the λ_j, $j=1,2,\ldots M$ are unique. This completes the proof.

In the case of the betas $\beta_{n,1}$ and $\beta_{n,2}$ given in Table 1, we need only consider $h(x+1) = \sum_{j=1}^{M} \lambda_j f^j(x+1+\Theta_j)$ in place of the above form for h. In the case of $\beta_{1,n}$ and $\beta_{2,n}$, we need not translate h.

NUMERICAL EXAMPLES

In order to explore the numberical behavior of these methods, simulation studies were conducted. Some examples of the simultation results are presented. The characteristic function of a mixture of normals, with equal variances, or a mixture whose component densities were exponential, or double exponential, or gamma or beta (as in Table 1) was used. In each case the mixture contained three densities.

Table 2 shows the case where the three densities are beta densities. The two end distributions ($\Theta=1$ and $\Theta=2$) are held fixed and the center distribution ($1<\Theta<2$) is considered for several values of the translation parameter. The results show that when two of the Θ's are close together, the error in the determination of their values is larger than for the case where they are far apart, as would be expected. In each case the ϕ matrix had three large eigenvalues (i.e., substantially larger than zero) so that it was an easy matter to say that, for numerical purposes, the rank of ϕ should be 3.

TABLE 2: Determining Location Parameters for a Mixture of Betas

$$h(x) = \sum_{j=1}^{3} \lambda_j f^j(x-\Theta_j)$$

$$f^j(x-\Theta_j) = \begin{cases} \dfrac{(n_j+3)!}{n_j!2}(x-\Theta_j)^2(1-x+\Theta_j)^n, & \Theta_j<x<\Theta_j+1 \\ 0, \quad x<\Theta_j, \quad x>\Theta_j+1 \end{cases}$$

λ	β	n	True Θ	True Θ -Estimate
1/3,1/3,1/3	1	8,4,4	1,1.01,2	.18852, .00826, .00002
			1,1.05,2	-.00091,-.00560,-.00003
			1,1.2,2	.00002,-.00001, .00006
			1,1.4,2	-.00001,-.00029, .00006
			1,1.6,2	-.00001,-.00027,-.00009
			1,1.8,2	-.00001,-.00104,-.00068

In Tables 3-6, the same basic experiment was repeated for the normal (equal variances) and the exponential, double exponential, and gamma. In these cases values of Θ close to 1 were studied. Except in the case of amixture of normals with the means .01 apart, in each case the rank of ϕ was judged to be 3. The same difficulties with determining Θ-values occurred as were noticed with the beta mixtures.

Theorem 2 uses a scale factor $\beta = \frac{2\Pi}{(K+1)\max|\Theta_j|}$. The precise value of β is not important to this theorem. We need only choose a scale factor so that $e^{ik\beta\Theta}$, $k=0,1,\ldots,K$ does not repeat. In Tables 3-6 we explored the use of $\beta=1$ and $\beta=1.5$ and we noted that there can be considerable differences in the determination of the Θ-values. In a real case, the choice of β would also presumably influence the accuracy of the answers, however, at this time we have not studied its effect enough to comment on possible appropriate values.

CONCLUDING REMARKS

To apply these methods one must know F in advance in order to determine the appropriate operator, e.g. $(i\omega)^{q-p}$, to apply to the characteristics function of the mixture, H. There is, however, some leadway. For example, we see from Table 1 that one could have a mixture of double exponentials, of gammas (n=1), or of betas of the form $\frac{(n+2)!}{n!}(x-\Theta)(1-x+\Theta)^n$, $0<x<1$, and still determine M and Θ-values by using the operator $(i\omega)^2$. Thus, some inexact knowledge of the underlying mixture model can be tolerated. Since, we have not as yet explored the estimation problems associated with these methods we

TABLE 3: Determining Location Parameters for a Mixture of Normals

$$h(x) = \sum_{j=1}^{3} \lambda_j \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-\Theta_j)^2}$$

λ	β	True Θ	True Θ -Estimate
.5,.3,.2	1.0	1,1.01,2	-.00031,not found*,.00011
		1,1.03,2	-.00038,-.00008,0
		1,1.05,2	.00233, .00641,0
		1,1.1,2	-.00012,-.00026,-.00001
.5,.3,.2	1.5	1,1.01,2	-.00372,not found*,.00011
		1,1.03,2	.00059, .00146, .00001
		1,1.05,2	.00005, .00004,0
		1,1.1,2	-.00007,-.00001,0

*The computer program could not distinguish between Θ=1 and Θ=1.01.

TABLE 4: Determining Location Parameters for a Mixture of Exponentials

$$h(x) = \sum_{j=1}^{3} \lambda_j f^j(x-\Theta_j)$$

$$f^j(x-\Theta_j) = \begin{cases} b\, e^{-b(x-\Theta_j)}, & x \geq \theta_j \\ 0, & x < \Theta_j \end{cases}$$

λ	β	b	True Θ	True Θ -Estimate
.5,.3,.2	1	1,5,2	1,1.01,2	.19612, .00249, .00003
			1,1.03,2	.00266, .00366, .00002
			1,1.05,2	.00415, .00014, .00001
			1,1.1,2	.00192, .00076, .00001
.5,.3,.2	1.5	1,5,2	1,1.01,2	-.00614,-.01134, 0
			1,1.03,2	-.00002,-.00005, 0
			1,1.05,2	-.00018,-.00011, 0
			1,1.1,2	-.00004,-.00002, 0

TABLE 5: Determining Location Parameters for a Mixture of Double Exponentials

$$h(x) = \sum_{j=1}^{3} \lambda_j f^j(x-\Theta_j)$$

$$f^j(x-\Theta_j) = \begin{cases} \frac{ab}{a+b} e^{a(x-\Theta_j)}, & x<\Theta_j \\ \frac{ab}{a+b} e^{-b(x-\Theta_j)}, & x>\Theta_j \end{cases}$$

λ	β	a	b	True Θ	True Θ -Estimate
.5,.3,.2	1	1	2	1,1.01,2	.12330, .00627, .00003
				1,1.03,2	.00110, .00338, 0
				1,1.05,2	.00213, .00831, .00004
				1,1.1,2	.00004, .00003, 0
.5,.3,.2	1.5	1	2	1,1.01,2	.00297, .00600, 0
				1,1.03,2	.00050, .00076, 0
				1,1.05,2	.00024, .00043, 0
				1,1.1,2	.00008, .00012, 0

TABLE 6: Determining Location Parameters for a Mixture of Gammas

$$h(x) = \sum_{j=1}^{3} \lambda_j f^j(x-\Theta_j)$$

$$f^j(x-\Theta_j) = \begin{cases} \dfrac{1}{n!\gamma^{n+1}} (x-\Theta_j)^n e^{-(x-\Theta_j)/\gamma}, & x>\Theta_j \\ 0, & x\leq\Theta_j \end{cases}$$

λ	β	γ	n	True Θ	True Θ -Estimate
.5,.3,.2	1	4	3	1,1.01,2	2.72623, .00628, .00011
				1,1.03,2	-.01094, .01407,-.00001
				1,1.05,2	-.01472, .02170, 0
				1,1.1,2	.00022, .00086,-.00001
.5,.3,.2	1.5	4	3	1,1.01,2	.00650, .00618, .00001
				1,1.03,2	.00042, .00235, 0
				1,1.05,2	.00007, .00015, 0
				1,1.1,2	.00001, .00007, 0

cannot comment on whether or not such an inexact knowledge of the mixture will translate over to more general lack-of-fit problems when real data is encountered.

In this paper we have only considered the univariate case. However, at least in the case of mixtures of normals, it would appear that the multivariate extension is straightforward provided one is clever about choosing the sampling values of ω. In future work we hope to consider multivariate extensions.

ACKNOWLEDGMENTS

The authors wish to thank Virginia Martin for providing the numerical examples that are given in this paper.

REFERENCES

[1] Teicher, H. Identifiability of Finite Mixtures. Annals of Math Stat. 34 (1963) 1265-1269.

[2] Lennington, R.K., C.T. Sorensen, R.P. Heydorn. Can Crop Types be Resolved Using Mixture Distribution Components. Proceedings of 1982 Machine Processing of Remotely Sensed Data Symposium.

[3] Grenander, U., G. Szegö. Toeplitz Forms and Their Applications (Berkeley, University of California Press, 1958).

[4] Redner, R.A. Maximum Likelihood Estimation for Mixture Models. NASA Johnson Space Center, NASA-CR-160967 (Sept. 1980).

[5] Yakowitz, S.S., J.D. Spragins. On the Identifiability of Finite Mixtures. Annals of Math Stat. 39 (1968) 209-214.

MULTIVARIABLE DENSITY ESTIMATION AND REMOTE SENSING

David W. Scott
Rice University

ABSTRACT

In this paper we describe our current efforts to develop methods and computer algorithms to effectively represent multivariate data commonly encountered in remote sensing applications. This may involve scatter diagrams but we are emphasizing multivariate representations of nonparametric probability density estimates. The density function provides a useful graphical tool for looking at data and a useful theoretical tool for classification. We call our approach a thunderstorm data analysis.

1. Graphical Tools in Data Analysis

A recent theme in multivariable data analysis as advocated by, for example, John and Paul Tukey [13] emphasizes graphical techniques for looking for multidimensional structure in data. The bivariate scatter diagram has been a very useful tool in this approach. For data in more than two dimensions, careful selection of bivariate projections can reveal structure in higher dimensions; see, for example, a description of the projection pursuit algorithm [3]. Alternately glyphs may be drawn instead of dots in a bivariate scattergram and data values not displayed are represented by features in the glyph, such as length, angle, etc. Computer graphics workstations have recently made trivariate scatter diagrams feasible. A true three-dimensional effect may be had by either continuous rotation of the scatter diagram or by a variety of stereographic techniques using red/green or polarized glasses. Holograms and rapidly vibrating mirrors also can proved 3-D effects. For data with more than three variables, side-by-side scatter diagrams of subsets of variables with visual links (such as coloring the same point in the different diagrams) allow a representation of the data.

Scatter diagrams do have limitations in data analysis. The most important problems relate to sample size. For moderately large samples ($n > 500$) data replication (or overstriking on the graphical medium) begins to occur frequently. This problem has been referred to as the problem of "too much ink" [12]. In one example of a fairly large 3-D scatter diagram with $n = 22,932$ on a 512 by 512 graphics terminal, only 4,000 pixels were observable [5]. With continuous rotation many more points are viewable but current computer technology limits real-time

rotations to about one thousand points. Secondly, clusters of points that are close together are difficult to detect in scatter diagrams. In other words scatter diagrams provide only modest indications of the density of points in a given region. Thirdly, our impression of data from the same underlying density function is highly dependent on the sample size. This makes comparisons of scatter diagrams with different sample sizes nontrivial. The eye naturally leaves the center of the data and focuses on outliers and apparent structure (lines) in outlying regions. Such features may or may not be of great importance depending on the objectives of the data analysis. In a recent example of a bivariate scatter diagram of 412,776 points, a frequency polygon analysis revealed that over 97% of the points fell inside the 1% contour (that is, points where $\hat{f}(x,y) = 1\%$ of $\hat{f}(\text{mode})$) which occupied less than $\frac{1}{15}$-th of the display area [6]. Almost half of the pixels in the display area were illuminated. On a 256 by 256 display, many points were replicated over 300 times and one more than 1000 times.

We also advocate using scatter diagrams for looking at data. However since we are interested in discovering structure such as modes and high density regions, we have found that the density function is a more useful tool when taking a preliminary look at data in several dimensions. The density function does not change with sample size, although the quality of estimation changes. In a sense the scatter diagram points to the density function, as Jim Thompson has described it. In the next sections we describe our current work based on multivariate nonparametric density estimation.

2. Computational and Representational Problems in Multivariate Density Estimation

Nonparametric density estimation methods for multivariate data are often simple extension of well-studied univariate versions. The multivariate histogram is a computationally efficient estimator but suffers from empty bin problems and bin edge effects. Statistically more efficient and smoother multivariate estimators may be obtained by kernel or nearest neighbor methods; see Tapia and Thompson [10]. Efficient algorithms for the latter have been developed but little is known about nearest neighbor global properties beyond some pointwise results. Some empirical evidence indicates nearest neighbor estimates tend to peak at modes and some optimal binning studies seem to draw the same conclusion [11]. Some special attention and techniques are needed in the tails since the raw estimate does not have a finite integral.

Thus we believe at this time the fixed multivariate kernel estimator of Cacoullos [2] is a useful technique for data in 2-4 dimensions. Unfortunately computational requirements grow rapidly in higher dimensions if one desires to evaluate the estimate of a representative multivariate mesh. The estimator also requires the entire raw data in order to compute the pointwise estimates. Some research has focused on one and two dimensional numerical approximations to kernel estimates in order to achieve computational efficiency [9]. However few results are currently available for more variables.

Another approach is to construct a frequency polygon estimator (formed by connecting with straight lines the mid-bin values of a histo-

gram). This estimator has the same order of statistical efficiency as the kernel estimator and also the computational efficiency of the histogram. However bin edge effects still can be a problem for small samples and in higher dimensions. Thus we have recently proposed a new density estimator based on a frequency polygon of the averaged shifted histogram (ASH) estimator [7]. The ASH is simply the pointwise average of m histograms with common equally spaced bins of width h but different bin origins $t_0 + \frac{i}{m}$, $i = 0,\ldots,m-1$. Thus the ASH looks like a histogram with bin width h/m. As $m \to \infty$ the ASH is identical to the statistically efficient triangular kernel estimate. Values of m between 3 and 10 are sufficient for most purposes. Multivariate versions are easily constructed by shifting and averaging in all co-ordinate directions.

Representational difficulties have been addressed for three and four variable density estimates (function surfaces in four and five dimensions, respectively) by displaying appropriate contour plots. For trivariate data a contour of $\hat{f}(x,y,z)$ will be a set of points

$$S_c = \{ (x,y,z) \in \mathbb{R}^3 : \hat{f}(x,y,z) = c \} .$$

The set S_c will be a surface in $\mathbb{R}^3$ (or more than one surface if the density is multimodal at this level). On a graphics terminal we have chosen to represent S_c by intersecting it with a series of equally spaced planes orthogonal to the x-axis, say, and then drawing the contours defined by these intersections. The resulting "wire" diagrams give a strong 3 dimensional impression. If color is available, several contour levels may be simultaneously displayed by using a different color for each level. We refer to our picture as a thunderstorm data representation.

It is helpful to imagine what this representation looks like for trivariate Gaussian data. For the independent variable case, S_c is simply a sphere so that a color display would show several concentric spheres with the mode located at the center. This is roughly illustrated in Figure 1. If the data are correlated we will see ellipsoids rather than spheres.

To represent the density estimate of four variables, $\hat{f}(x,y,z,t)$, we look at the sets

$$S_{t,c} = \{ (x,y,z) \in \mathbb{R}^3 : \hat{f}(x,y,z,t) = c \} .$$

Here we have arbitrarily chosen one variable and placed it in a reference frame which may conveniently be thought of as a "time" axis. By looking at a time-lapse sequence of representations of $S_{t,c}$ we obtain a useful view of the data which highlights important features such as modes, outliers, symmetry, skewness, and covariance structure. This sequence is similar to a time-lapse movie of a thunderstorm from its original formation to peak of storm to its eventual end.

Again it is useful to construct this representation for quadravariate Gaussian data. For a fixed contour level c, as t moves through the relevant interval of support (t_{min}, t_{max}), $S_{t,c}$ will be a sequence of initially expanding spheres (ellipsoids) which continue to grow until the mode is reached and then contracting and finally vanishing when $S_{t,c}$ becomes the null set.

We have recently experimented with these representations using Landsat remote sensing reflectance intensity data sets in $\mathbb{R}^3$ (n = 23,000) and with a particle physics data set in $\mathbb{R}_4$ (n = 500); see

Scott [5] and Scott and Thompson [8]. A 16mm color film was used to record the time-lapse thunderstorm representation of the particle physics data set. These data have been analyzed by Friedman and Tukey [3] and by Tukey and Tukey [13] using exploratory data and scatter diagram techniques. Our representations seem to be successful in uncovering important data features and structure and seem to require less training in the four dimensional case than required for four dimensional rotating scatter diagram methods.

3. Graphical and Model-Based Discrimination and Classification

We shall assume that our data samples are labelled so that supervised clustering and discrimination are feasible. As a preliminary step, side-by-side scatter diagrams may be displayed to get a rough feeling for the separability of cluster classes. This may also be accomplished by displaying side-by-side density contour plots for the cluster classes. For large training samples the latter is more useful (see the comparison of a scatter diagram and contour plot for 412,776 points mentioned in section 1). The scatter diagram might indicate no separation at all.

When the preliminary density estimates have been refined by optimal data-based choices of smoothing parameters, classification may be accomplished using a Bayesian classifier. Evaluation of the averaged shifted histogram for each class involves only a bin location operation (subtraction and division) and then a table lookup for each training class (hash function, perhaps). This is a computationally efficient operation

although large memory requirements are necessary in several dimensions. We plan to implement this strategy and report on our results shortly.

Examples

We shall consider the scatter diagram approach discussed in section 3 as a preliminary step towards producing a nonparametric classifier. The data are trivariate and come from a model applied to individual pixels (1.1 acre) using temporally measured Landsat data. Approximately biweekly 4-channel remote sensing relectance intensity data were converted into a single "greenness" time series by looking at a certain linear combination of the 4-channel data. The time series was fitted by Badhwar's [1] growth model which looks somewhat like a bell-shaped curve. For each pixel three parameters from Badhwar's model were extracted: x, the time of peak greenness; y, the ripening or reproduction period; and z, the peak greenness level. Each measurement was recorded on a discrete scale from 0 to 249. The data are processed in a segment which is 5 by 6 nautical miles and contains 22,932 (117 by 196) pixels. Ground truth was obtained by sending observers to the fields.

In Figure 2 we show a view of the 3-D scatter diagram for segment 1380 in Minnesota, 1978. Notice the orientation of the axes (located at the true origin) in this projected and rotated view. The projected x-axis is defined by the vector (-.71,.71,0) and the y-axis is defined by (-.58,-.58,.58). This scatter diagram is a mixture of "pure" and "mixed" pixels. In Figure 3 we show a scatter diagram of 3,947 pure pixels of corn from segment 1380. Figure 4 depicts the 5,162 pure pixels of soybeans. A quick impression of the separability of corn and

soybeans is possible from these graphs, but again recall that a large fraction of the data are hidden, making the discrimination judgment very difficult.

Small grains present a difficult problem for a classifier. In Figure 5 we view segment 1899 in North Dakota, 1977, using the same projection plane as before. The two segments look quite different in this representation. Figure 6 represents 1,756 pure pixels of sugar beets. Figure 7 represents 3,355 pure pixels of spring wheat. Finally, Figure 8 shows 4,362 pure pixels of barley. These classes present a challenge for any discrimination procedure.

5. Conclusion

We have attempted to illustrate how nonparametric density methods may be brought to bear directly on multivariate remote sensing problems. Multivariate parametric models based on mixture models [4] have many advantages, both conceptually and in production mode. The fitting problems in the parametric case are usually quite difficult. We hope to investigate how nonparametric models may provide guidance to the fitting and verification of such parametric models. This would be a direct use of the exploratory capabilities of the nonparametric models.

References

[1] Badhwar, G.G., Carnes, J.G. and Austin, W.W., Use of Landsat-Derived Temporal Profiles for Corn-Soybean Feature Extraction and Classification, Remote Sensing of Environment 12 (1982) 57-59.

[2] Cacoullos, T., Estimation of a Multivariate Density, Annals of the Institute of Statistical Mathematics 18 (1966) 179-189.

[3] Friedman, J.H. and J.W. Tukey, A Projection Pursuit Algorithm for Exploratory Data Analysis, IEEE Trans. Comp. C-23 (1974) 881-890.

[4] Heydorn, R.P., and Basu, R., Estimating Proportions of Materials Using Mixture Models, in Guseman, L.F. (ed.), Proceedings of the NASA/MPRIA Workshop: Math/Stat (January 1983).

[5] Scott, D.W., Nonparametric Probability Density Estimation for Data Analysis in Several Dimensions, to appear Proc. 27th Conference on the Design of Experiments in Army Res Dev and Testing (1983).

[6] Scott, D.W., Frequency Polygons: Theory and Application, submitted (1983).

[7] Scott, D.W., Average Shifted Histograms, working paper (1983).

[8] Scott, D.W., and Thompson, J.R., Probability Density Estimation in Higher Dimensions, To Appear In: Proceedings of the 15th Symposium on the Interface of Computer Science and Statistics (1983).

[9] Silverman, B.W., Density Estimation for Univariate and Bivariate Data, in Barnett, V. (ed.), Interpreting Multivariate Data, (John Wiley & Sons, New York, 1981).

[10] Tapia, R.A., and Thompson, J.R., Nonparametric Probability Density Estimation, (Johns Hopkins Univ. Press, Baltimore, 1978).

[11] Terrell, G.R., and Scott, D.W., Variable Window Density Estimates, to be presented ASA meeting, Toronto 1983.

[12] Tufte, E.R., The Visual Display of Quantitative Information, (Graphics Press, Cheshire, CT, 1983).

[13] Tukey, P.A. and J.W. Tukey, Graphical Display of Data Sets in 3 or More Dimensions, in Barnett, V. (ed.), Interpreting Multivariate Data, (John Wiley & Sons, New York, 1981).

Acknowledgment: This research supported by NASA/Lockheed under grant PO-0200100079.

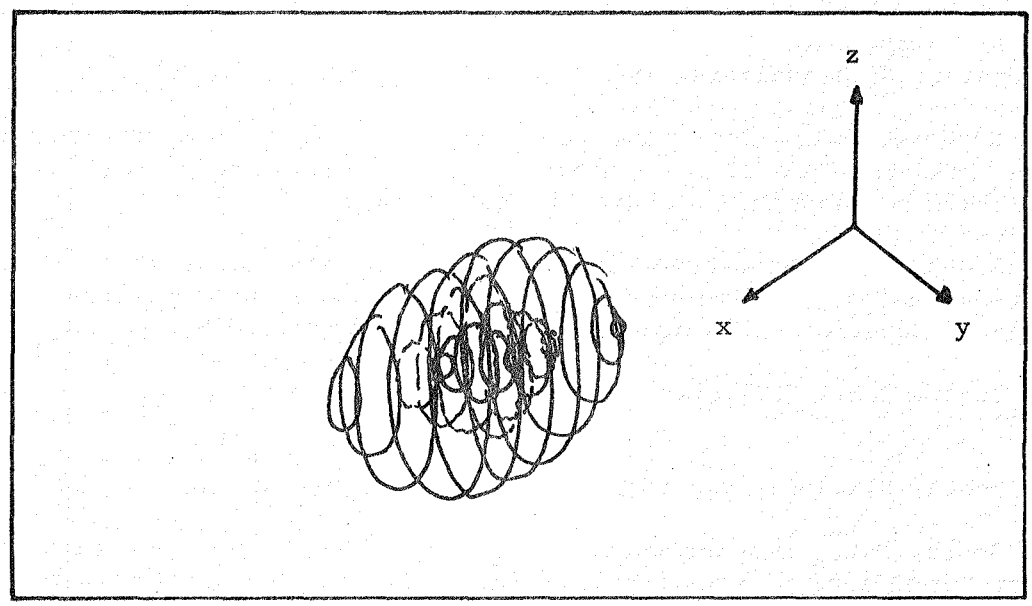


Figure 1. Representation of Contours of Three Dimensional Density Estimate if Gaussian

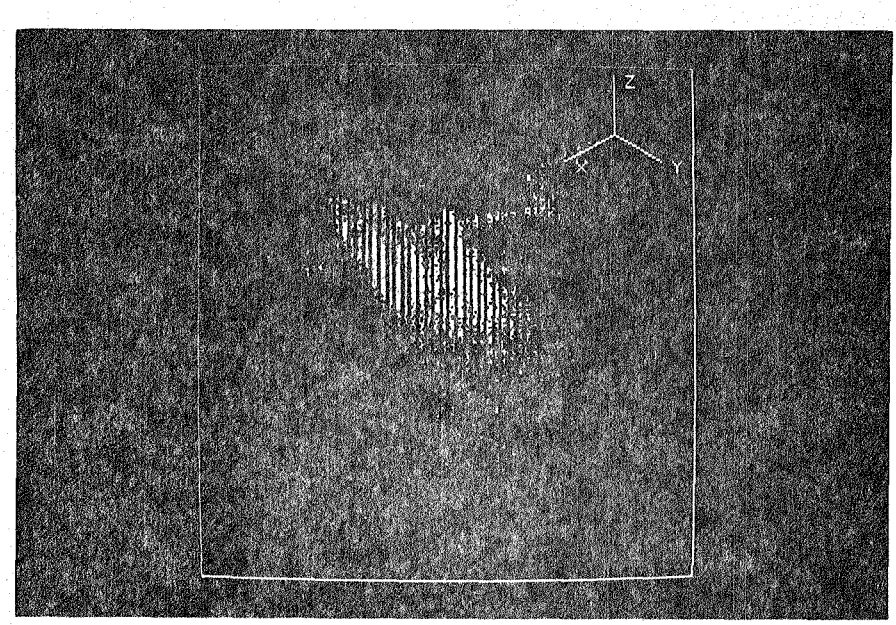


Figure 2. Projected and Rotated Three Dimensional Scatter Diagram of Segment 1380 (1978).

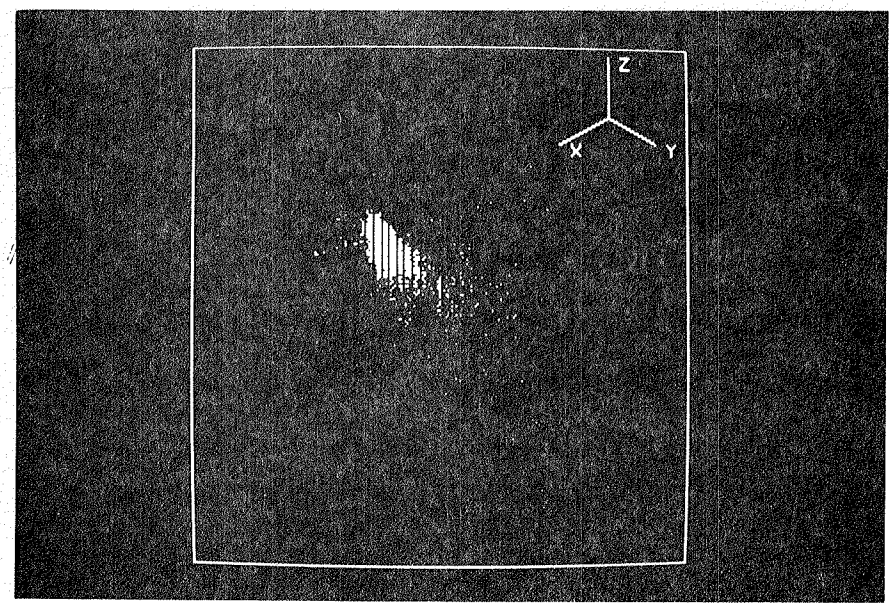


Figure 3. Pure corn pixels in segment 1380 (n=3,947).

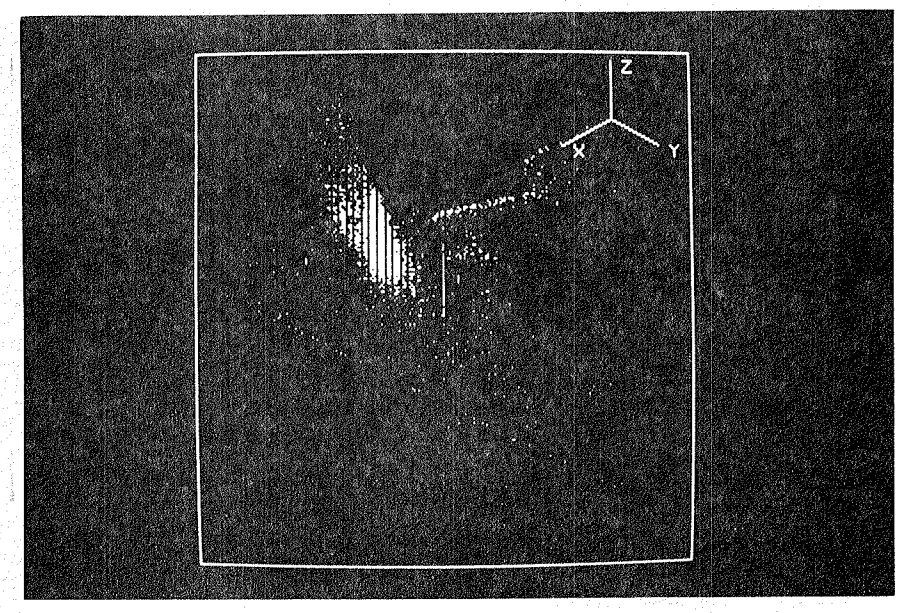


Figure 4. Pure soybean pixels in segment 1380 (n=5,162).

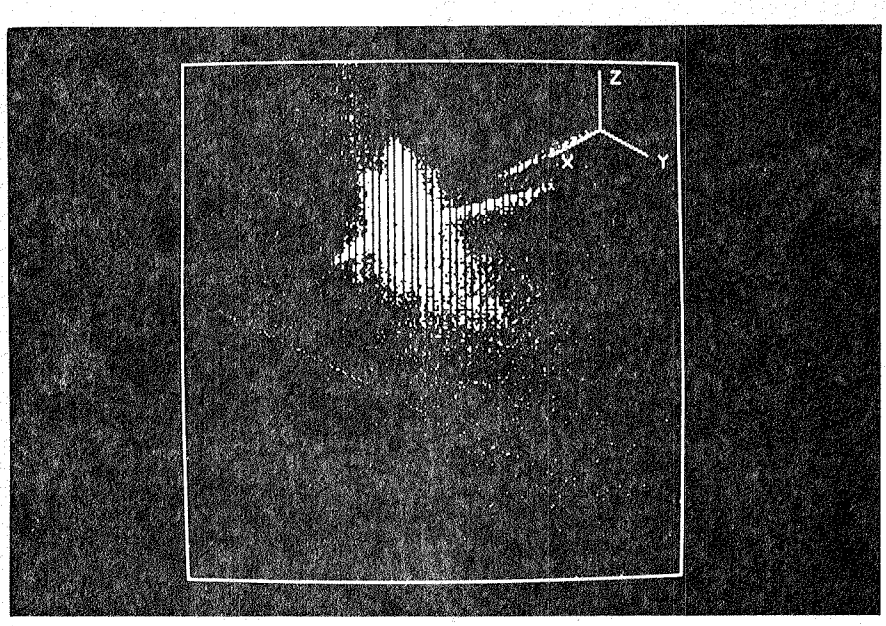


Figure 5. Scatter diagram of segment 1899 (1977) (n=22,932).

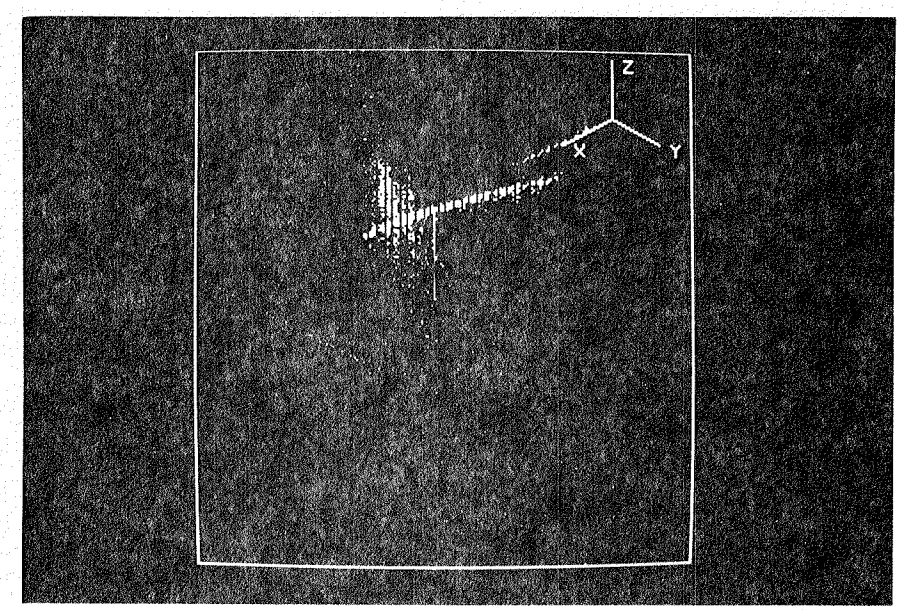


Figure 6. Sugar beet pixels in segment 1899 (n=1,756).

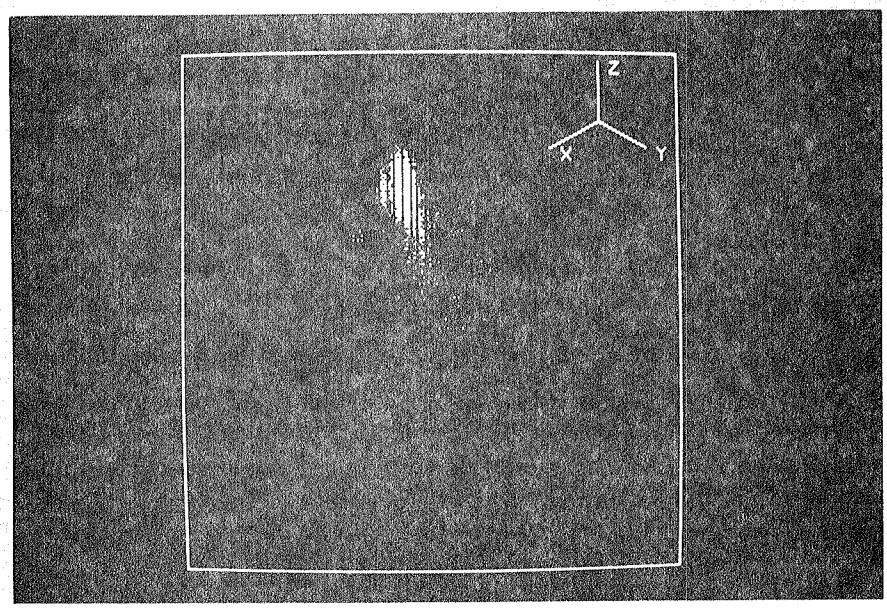


Figure 7. Spring wheat pixels in segment 1899 (n=3,355).

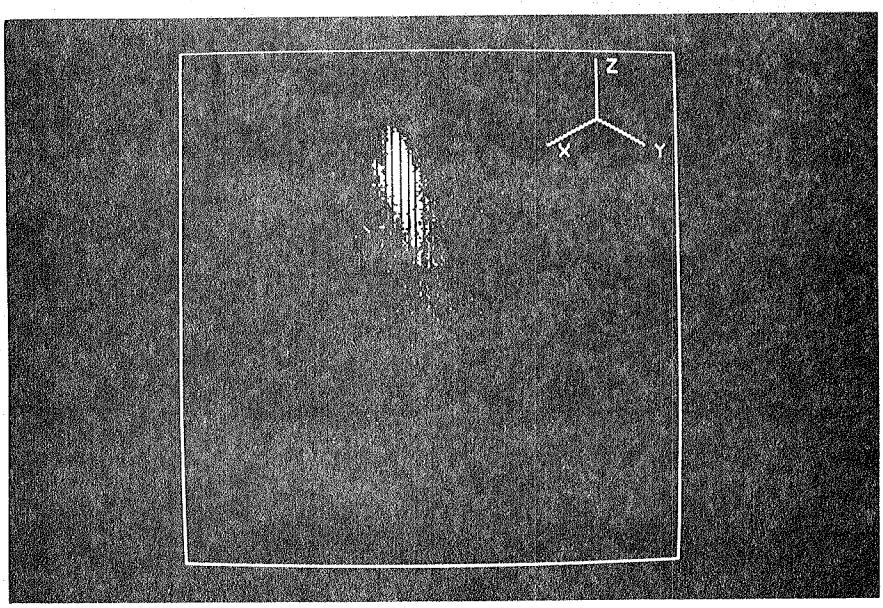


Figure 8. Barley pixels in segment 1899 (n=4,362).

Page intentionally left blank

AUTOREGRESSIVE MODELS FOR USE IN SCENE SEGMENTATION

M. Naraghi
Jet Propulsion Laboratory
California Institute of Technology

ABSTRACT

A scene segmentation approach is presented which is based on generating autoregressive field models for each scene component (class) from its a priori spatial statistics. A methodology is also described for using these models in achieving optimal segmentation of a scene. The derivations are presented for the case of single band imagery, however, the method is believed to be extendable to multispectral data.

1. Introduction

A subject of central importance in image pattern recognition and analysis has been scene segmentation and classification of scene components. In addressing this subject, a number of different methodologies and approaches have been proposed and implemented. These range from simple thresholding concepts to methods that define a scene component by a set of texture measures and achieve segmentation using such measures [1].

This research, being reported in this paper, is concerned with the development of techniques for segmentation when the scene components (referred to as classes) are or can be described statistically. Specifically, the concepts and procedures that are developed apply to the cases where the scene components are members of a two-dimensional and stationary Gaussian process. Though, the final goal of this activity is to have segmentation techniques for multispectral data, this report covers the approach for a single band image. The extension of the derived methods for application to multispectral data are currently under investigation.

Statistical description of scene components has been established as a viable approach in pattern recognition and image analysis [1]-[4]. In the following, the approach taken is that of first describing each class by an autoregressive model using the a priori statistics of that class and then employing these models in achieving segmentation. After the general notation is established in Part 2, the modeling technique is derived in Part 3. In Part 4 the segmentation technique which uses the derived models is presented and discussed.

2. Preliminaries and Notations

For a single band image let there be M classes $w_1, \ldots, w_M$, where the intensities of pixels in each class are a sample function of a two-dimensional (2-D) Gaussian and stationary random process with known a priori means $\mu, \ldots, \mu_M$ and autocorrelations $R_1(\tau_1,\tau_2), \ldots, R_M(\tau_1,\tau_2)$. So for the k^{th} class, the a priori mean μ_k and the autocorrelation $R_k(\tau_1,\tau_2)$ are defined by

$$\mu_k = E\, I^k(i,j) \tag{2.1}$$

$$R_k(\tau_1,\tau_2) = E[I^k(m,n)-\mu_k][I^k(i,j)-\mu_k]$$

where $\tau_1 = |m-i|$, $\tau_2 = |n-j|$, $I^k(i,j)$ denotes the intensity value at pixel location (i,j) in the k^{th} class and E is the expectation operator.

In the subsequent sections, autoregressive models of various orders will be defined and used. Figure 1 defines what is meant by specifying various autoregressive model orders on a two-dimensional grid. Thus a first order model for location (i,j) contains the pixels{(i-1,j),(i,j-1), (i-1,j-1)} and a second order model contains the pixels {(i-1,j), (i,j-1), (i-1,j-1), (i-2,j), (i, j-2), (i-2,j-1), (i-1,j-2), (i-2,j-2)} and so on. Index i represents the line (row) indicator and j is the sample (column) indicator on a 2-D grid.

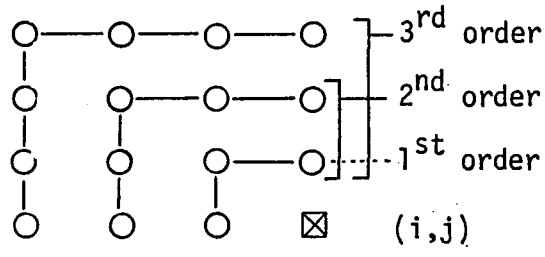


Fig. 1

3. Autoregressive Modeling Procedure

Autoregressive models have been analyzed and used in the area of image processing and analysis for some time [1]-[3], [5]-[6]. In general, for a zero mean Gaussian random process $x(i,j)$, these models are of the form [7]

$$x(i,j) = \sum_{(p,q)\in D}\sum \alpha_{pq}\, x(i-p,j-q) + U(i,j) \tag{3.1}$$

where,

$$D = \{(p,q):-M \le p \le M,\ -N \le q \le N,\ (p,q) \neq (0,0)\} \tag{3.2}$$

and $U(i,j)$ are a set of independent Gaussian random variables, where

$$E\,U(i,j) = 0$$

$$E\,U(i,j)\,U(k,\ell) = \begin{cases} \sigma^2 & \text{if } i=k \ \&\ j=\ell \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

σ^2 and α_{pq} are constants if $x(i,j)$ is stationary and they are a function of (i,j) if $x(i,j)$ is nonstationary.

A causal form of the model in (3.1) is the subject of interest in this paper. In this causal form (3.1) is written as

$$x(i,j) = \sum_{\substack{p=0 \\ p+q \neq 0}}^{P}\ \sum_{q=0}^{P} \alpha_{pq}\, x(i-p,j-q) + U(i,j) \tag{3.4}$$

where, again with stationarily, α_{pq} are constants and $U(i,j)$ are a set of identically distributed random variables satisfying (3.3). Here P is the order of the autoregressive model corresponding to the definition of the model order given in Figure 1. An example of such a causal model is

the first order model

$$(3.5)\quad x(i,j) = \alpha_{01}\, x(i,j-1) + \alpha_{10}\, x(i-1,j) + \alpha_{11}\, x(i-1,j-1) + U(i,j)$$

which has a two-dimensional separable correlation function [6] of the form

$$(3.6)\qquad R(\tau_1,\tau_2) = \sigma_s^{\,2}\, \mathrm{EXP}\{-\beta_1|\tau_1| - \beta_2|\tau_2|\}$$

The thrust of modeling in segmenting a scene is to transform the information provided a priori about each class (namely the correlation) into an autoregressive model and use these models in subsequent development of segmentation methods. Clearly the choice of autoregressive forms is arbitrary and there is no claim made here that all classes can be modeled by such forms. However, the causality restriction that has been imposed (and will be adhered to throughout this paper) is necessitated by the particular modeling procedure described in 3.1 and the properties of the derived models which are descussed in 3.2.

3.1. The Autoregressive Modeling Technique

In the following a procedure is developed for deriving the model from the given a priori correlation. Since this process is done for each class, then the class indicator (superscript k) is omitted from all arguments in the ensuing discussion.

For a given 2-D and stationary correlation function $R(\tau_1,\tau_2)$, let us assume a model order P. First we will develop a technique for defining the model for a given P and then we'll show how the "best" order P is chosen. For a given order P, the model is

$$x(i,j) = \sum_{\substack{k=0 \\ k+\ell \neq 0}}^{P} \sum_{\ell=0}^{P} \alpha_{k\ell}\, x(i-k,j-\ell) + U(i,j) \tag{3.7}$$

This model is completely defined if the values of all the constants $\alpha_{k\ell}$ and the variance of the zero mean white noise process U(i,j) are known. Thus, for a given order P, there are $(P+1)^2$ unknowns to be computed where $(P+1)^2-1$ of these are the unknowns $\alpha_{k\ell}$ and one unknown is σ^2 where

$$\sigma^2 = E\, U^2(i,j) \tag{3.8}$$

The criterion adopted here for computing these unknown parameters is that of minimum variance of U(i,j). Thus $\alpha_{k\ell}$ are found such that $E\, U^2(i,j)$ is minimized and σ^2 is taken to be that minimum value. From (3.7)

$$E\, U^2(i,j) = E\Big[x(i,j) - \sum_{\substack{k=0 \\ k+\ell \neq 0}}^{P} \sum_{\ell=0}^{P} \alpha_{k\ell}\, x(i-k,j-\ell)\Big]^2 \tag{3.9}$$

Differentiating (3.9) with respect to $\alpha_{k\ell}$'s and setting it equal to zero results in the $(P+1)^2-1$ equations

$$(3.10) \qquad E[x(i,j) - \sum_{\substack{k=0 \\ k+\ell \neq 0}}^{P} \sum_{\ell=0}^{P} \alpha_{k\ell}\, x(i-k,j-\ell)]\, x(m,n) = 0$$

$$m = i, i-1, \ldots., i-P$$

$$n = j, j-1, \ldots., j-P$$

$$(m,n) \neq (i,j)$$

Carrying the expectation operator through in (3.10) and rearranging the terms results in a system of linear equations of the form

$$(3.11) \qquad A\,\underline{\alpha} = b$$

where elements of the vector $\underline{\alpha}$ are the coefficients $\alpha_{k\ell}$ and the elements of the matrix A and the vector b are values of the correlation function $R(\tau_1,\tau_2)$.

Having solved for the coefficients $\alpha_{k\ell}$ in (3.11), it remains to determine the quantity σ^2 in order to have the model defined. Expanding the quadratic form in (3.9), σ^2 can be written as

$$(3.12) \qquad \sigma^2 = E\,U^2(i,j) = E\left[x(i,j) - \sum_{\substack{k=0 \\ k+\ell \neq 0}}^{P} \sum_{\ell=0}^{P} \alpha_{k\ell}\, x(i-k,j-\ell)\right] x(i,j)$$

$$-E\left[x(i,j) - \sum_{\substack{k=0 \\ k+\ell \neq 0}}^{P} \sum_{\ell=0}^{P} \alpha_{k\ell}\, x(i-k,j-\ell)\right]\left[\sum_{\substack{k=0 \\ k+\ell \neq 0}}^{P} \sum_{\ell=0}^{P} \alpha_{k\ell}\, x(i-k,j-\ell)\right]$$

But from the relations in (3.10)

$$E\left[x(i,j) - \sum_{\substack{k=0 \\ k+\ell\neq 0}}^{P}\sum_{\ell=0}^{P} \alpha_{k\ell}\, x(i-k,j-\ell)\right]\left[\sum_{\substack{k=0 \\ k+\ell\neq 0}}^{P}\sum_{\ell=0}^{P} \alpha_{k\ell}\, x(i-k,j-\ell)\right] = 0$$

Thus

$$\sigma^2 = E[x(i,j)]^2 - \sum_{\substack{k=0 \\ k+\ell\neq 0}}^{P}\sum_{\ell=0}^{P} \alpha_{k\ell}\, E\, x(i,j)\, x(i-k,j-\ell) \tag{3.13}$$

$$= R(0,0) - \sum_{\substack{k=0 \\ k+\ell\neq 0}}^{P}\sum_{\ell=0}^{P} \alpha_{k\ell}\, R(k,\ell)$$

To have completely defined the modeling process, it remains to show how the model's order P is chosen. Before stating the process that allows one to choose the optimal order, let us review what is the objective of the modeling endeavour and what is meant by optimal. As stated before, the objective is that of generating an autoregressive model whose second moment characteristics (the correlation function) approximates the given a priori correlation function $R(\tau_1,\tau_2)$ as closely as one wishes. However, the criterion chosen for defining the model has been minimization of the white noise variance. Besides the intuitive appeal of this criterion, it will be shown in the next section that this criterion also satisfies the stated objective above. Hence finding the best order is achieved by generating models of various orders and choosing the one whose white noise has minimum variance. In general, then

successively higher order models are assumed and their parameters $\alpha_{k\ell}^{P}$ and σ_P^2, P = 1, 2, ..., are computed. Optimal choice of P is made according to one or more of the following:

1. σ_P^2 does not change with increasing P i.e., $\sigma_{P+1}^2 = \sigma_P^2$. This is the case where the underlying process has an exact auto-regressive model of order P as will be shown in Section 3.2.
2. Only few values of the a priori correlation function $R(\tau_1,\tau_2)$ are specified which limits how high the order P that can be chosen.
3. Rate of decrease of σ_P^2 as P increases. This is the case where the underlying process does not lend itself to a small order regression model in which case an approximate model is chosen on the basis of trade-off between the decrease in σ_P^2 and additional segmentation cost and complexity due to the increase in the number of model coefficients. As an example of σ^2 varies as in Figure 2 as a function of P, then the value $\hat{P}$ could be taken as the best order.

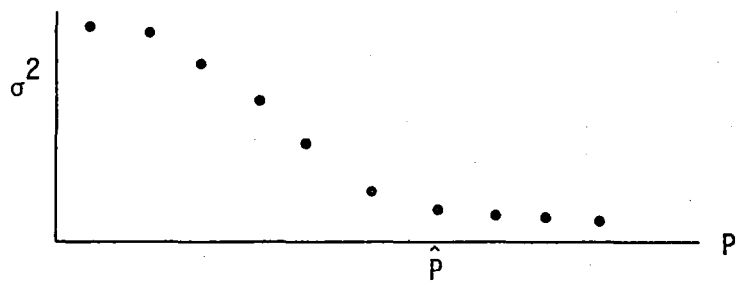


Figure 2.

3.2. Properties of the Modeling Technique

The properties are:

1. If the underlying 2-D process satisfies a finite order autoregressive model, this procedure will find that model. The proof of this property is given in Appendix A.
2. When an approximate model of order P is chosen, the correlation generated by this model matches the a priori correlation at, at least, $(P+1)^2$ points. The proof of this property is given in Appendix B.
3. In deriving the model, only numerical values of the correlation $R(\tau_1,\tau_2)$ are needed and no analytic form is required. Therefore in practice, $R(\tau_1,\tau_2)$ can be obtained numerically using training areas.
4. Though, beyond the scope of present considerations, this method is believed to be applicable when stationary constraint is removed and nonstationary processes are to be modeled.
5. For a given correlation function, the described procedure will always generate a model. This model, however, may be unstable hence unacceptable for our purposes since it cannot represent a homogeneous process. Under these circumstances, then, tests must be performed to insure stability [10].

4. Scene Segmentation

Having found an either exact or approximate autoregressive model for each class, the following describes how these models are used in achieving optimal segmentation. The optimality criterion is derived in Appendix C and it is evident that this criterion is somewhat different than the familiar classification criterion. This is to be expected since the segmentation process, by nature, not only is a classification process but is a partitioning process as well.

Development of a general segmentation method that satisfies all the intrinsic conditions of the optimality criterion of (C-8) is currently under investigation. In the next part, however, a segmentation method is presented which divides the image into blocks (a group of pixels) and classifies each block according to the optimality principle.

4.1. Segmentation Procedure

Let the models associated with the M classes w_1, ..., w_M be of orders P_1, ..., P_M, respectively and let

$$P = \max\{P_1, \ldots, P_M\} \tag{4.1}$$

The segmentation of the image is achieved by dividing the entire image blocks of $(P+1)x(P+1)$ in size and classifying the individual blocks starting at the upper left hand corner and in the row by row fashion. Let B_{ij} designate the block in row i and column j. Within each block let the intensities of the image be $y(k,\ell)$, $k=1, \ldots, P$ and $\ell = 1, \ldots, P$ and finally let the pixels in B_{ij} be rearranged in the vector y_{ij} as follows:

$$y_{ij} = \{y(1,1), y(1,2), \ldots, y(1,P), y(2,1), \ldots, y(P,P)\} \tag{4.2}$$
$$\{y(1,1), \ldots, y(P,P)\} \in B_{ij}$$

A given block B_{ij} is considered to be a starting block if the three blocks $B_{i-1,j}$, $B_{i,j-1}$ and $B_{i-1,j-1}$ either do not exist (i.e., B_{ij} is on the uppermost or the left hand most part of the image) or these blocks do exist but they all have not been classified into the same class (i.e. if $B_{i,j-1} \in w_2$ and $B_{i-1,j} \in w_4$, for example). With this definition, then, the segmentation process will be totally defined by describing how a starting and a non-starting block are classified.

Assuming equal a priori probability of occurence of each class, w_1, ..., w_M,

(4.3) $$P(w_1) = P(w_2) = ... = P(w_M)$$

a starting block B_{ij} is classified to the class w_k if

(4.4) $$P(y_{ij}|w_k) \geq P(y_{ij}|w_\ell) \quad \ell = 1, ..., M$$

Since

(4.5)

$$P(y_{ij}|w_\ell) = \frac{1}{(2\pi)^{N/2}|\phi_\ell|^{1/2}} EXP[-1/2(y_{ij} - \underline{\mu}_\ell)^T \phi^{-1}(y_{ij} - \underline{\mu}_\ell)]$$

where

$N = (P+1)^2$, $\underline{\mu}_\ell$ is an N x 1 vector whose elements are the mean value of class w_ℓ and ϕ_ℓ is the covariance matrix of the vector y_{ij} as defined in (4.2). Note that for each class w_ℓ, the matrix ϕ_ℓ is determined from the a priori class statistics in (2.1), hence $|\phi_\ell|$ and ϕ_ℓ^{-1} are computed only once for each class.

Substituting (4.5) in (4.4) and taking natural logarithm and simplifying both sides yields the following rule for classifying a starting block B_{ij}:

(4.6) $B_{ij} \in w_k$ if

$$|\phi_k| + (y_{ij} - \underline{\mu}_k)^T \phi_k^{-1}(y_{ij} - \underline{\mu}_k)$$

$$\leq |\phi_\ell| + (y_{ij} - \underline{\mu}_\ell)^T \phi_\ell^{-1}(y_{ij} - \underline{\mu}_\ell)$$

for all $\ell = 1, \ldots, M$

Now if B_{ij} is not a starting block this means that $B_{i-1,j}$, $B_{i,j-1}$ and $B_{i-1,j-1}$ have all been already classified into the same class, say w_k. The block B_{ij} is also classified in the class w_k if

$$P(y_{ij}|y_{i,j-1}, y_{i-1,j}, y_{i-1,j-1}, w_k) \geq P(y_{ij}|w_\ell) \tag{4.7}$$

for all $\ell = 1, \ldots, M$

$\ell \neq k$

Otherwise, B_{ij} is classified in class w_n where

$$P(y_{ij}|w_n) \geq P(y_{ij}|w_\ell) \tag{4.8}$$

for all $\ell = 1, \ldots, M$

$n, \ell \neq k$

In other words, if (4.7) is not satisfied, then B_{ij} is determined not to belong to w_k and is treated as a astarting block for any other class except w_k.

The right hand side of (4.7) and both sides of (4.8) are evaluated using (4.5). The left hand side of (4.7), however, is to be evaluated using the autoregressive model of the class w_k. Let the zero mean model of this class be

$$x(i,j) = \sum_{m=0}^{P_k} \sum_{n=0}^{P_k} \alpha_{mn}^{k} \, x(i-m,j-n) + U(i,j)$$

which indicates that each element, of the vector y_{ij} in (4.2) satisfies

$$(4.9) \qquad y(r,q) - \mu_k = \sum_{\substack{m=0 \\ m+n\neq 0}}^{P_k} \sum_{n=0}^{P_k} \alpha_{mn}^{k} \left[y(r-m,q-n) - \mu_k \right] + U(r,q)$$

where r and q now refer to the actual location on the two-dimensional grid in the image. For each element of y_{ij} corresponding to location (r,q) on the image let

$$(4.10) \qquad \hat{y}(r,q) = \sum_{\substack{m=0 \\ m+n\neq 0}}^{P_k} \sum_{n=0}^{P_k} \alpha_{mn}^{k} \left[y(r-m,q-n) - \mu_k \right]$$

Substituting (4.10) in (4.9) results in

$$(4.11) \qquad y(r,q) - \mu_k - \hat{y}(r,q) = U(r,q)$$

But since U(r,q) are a set of independent variables, the left hand side of (4.7) is equivalent to

$$(4.12) \qquad P = p(y_{ij} | y_{i,j-1}, y_{i-1,j}, y_{i-1,j-1}, w_k) =$$

$$p(U(r_1,q,)) \; p(U(r_2,q_2)) \; \ldots .$$

where again (r_ℓ, q_ℓ) is the location of the ℓ^{th} element of y_{ij} on the image. Substituting (4.11) in (4.12) yields

$$(4.13) \qquad P = \frac{1}{(2\pi)^{N/2} \sigma_k^{N}} \; EXP\left[- \frac{1}{2\sigma_k^{2}} \sum_{\ell=1}^{N} \left(y(r_\ell, q_\ell) - \hat{y}(r_\ell, q_\ell) - \mu_k \right)^2 \right]$$

where

$$\sigma_k^2 = E\, U^2(r,q)$$

$$N = (P+1)^2$$

As before, for the sake of comparison in (4.7) the quantity

$$(4.14) \qquad P' = N \ln(\sigma_k^2) + \frac{1}{\sigma_k^2} \sum_{\ell=1}^{N} [y(r_\ell, q_\ell) - \hat{y}(r_\ell, q_\ell) - \mu_k]^2$$

is used in the actual implementation.

4.2. Optimality of the Segmentation Procedure

In order to discuss the optimal characteristics of the procedure of 4.1 it must be pointed out that the procedure as presented takes a group of pixels (a block) and classifies them (it) into a given class. Hence on the pixel by pixel basis, the procedure cannot be optimal since a class boundary can be such that it goes through a given block while the procedure, as it stands now, will classify all the pixels in that block into a particular class. However, ignoring the misclassification of the pixels around the boundaries and viewing the image in a block form, the question remains as to whether the blocks are classified optimally or not.

At this stage, however, instead of considering the overall optimality of the procedure let us consider implications of the optimality rule when a non-starting block is processed and classified. The reason for this limited analysis, at this time, is the author's belief that it is this part of the process that shed's the most light in the development of future optimal segmentation techniques. So let B_{ij} be an arbitrary non-starting block and let us assume that the segmentation achieved up to B_{ij} has been optimal. Let B be the set of all the blocks previous to B_{ij} (in the operational scheme of the last section) that has already been optimally segmented. For the sake of notational ease, and without loss of generality, let us further assume that B is classifiedinto a particular class w_p. So

$$p(B|w_p) \geq p(B_1|w_a) \ldots p(B_\ell|w_h) \tag{4.15}$$

for all a, b=1, ..., M and all subsets B_m of B. Now if the procedure classified B_{ij} into w_p as well, then from (4.7)

$$(4.16) \qquad p(B_{ij}|B_{i,j-1}, B_{i-1,j}, B_{i-1,j-1}, w_p) \geq p(B_{ij}|w_k)$$

$$\text{for all } k \neq p$$

But due to the Markov property of the process in class w_p

$$(4.17) \qquad p(B_{ij}|B_{i,j-1}, B_{i-1,j}, B_{i-1,j-1}, w_p) = p(B_{ij}|B,w_p)$$

Substituting (4.17) in (4.16) and multiplying both sides by (4.15) results in

$$(4.18) \quad p(B_{ij}|B,w_p)p(B|w_p) \geq p(B_1|w_{\bar{a}}) \ .... \ p(B_\ell|w_b)p(B_{ij}|w_k)$$

But

$$(4.19) \qquad p(B_{ij}|B,w_p)p(B|w_p) = p(B_{ij},B|w_p)$$

hence (4.18) becomes

$$(4.20) \qquad p(B,B_{ij}|w_p) \geq p(B_1|w_a) \ .... \ p(B_\ell|w_b)p(B_{ij}|w_k)$$

for all a, b and $k \neq p$ and all subsets B_m. Thus (4.20) shows that when (4.7) is satisfied then the segmentation remains optimal.

Similarly it can be shown that if (4.7) is not satisfied, the segmentation will remain optimal.

Appendix A

Let the zero mean 2-D Gaussian process $x(\cdot,\cdot)$ satisfy a p^{th} order autoregressive model of the form

$$x(i,j) = \sum_{\substack{k=0 \\ k+\ell=0}}^{P} \sum_{\ell=0}^{P} \alpha_{k\ell}\, x(i-k,j-\ell) + U(i,j) \quad 0 \cdot \tag{A.1}$$

$$E\, U(i,j) = 0$$

$$E\, U^2(i,j) = \sigma^2$$

then $x(\cdot,\cdot)$ is a Markov process having the property

$$p[x(i,j)|x(i,j-1), \ldots., x(i-P,j-P), \ldots., x(i-P-m,j-P-m)] = \tag{A.2}$$

$$p[x(i,j)|x(i,j-1), \ldots., x(i-P,j-P)]$$

for any $m \geq 0$. From (A.2), then we have

$$E\, x(i,j)|x(i,j-1), \ldots. x(i-P,j-P), \ldots. x(i-P-m,j-P-m) = \tag{A.3}$$

$$E\, x(i,j)|x(i,j-1), \ldots., x(i-P, j-P)$$

But from (A.1)

$$E\, x(i,j)|x(i,j-1), \ldots., x(i-P,j-P) = \sum_{\substack{k=0 \\ k+\ell\neq 0}}^{P} \sum_{\ell=0}^{P} \alpha_{k\ell}\, x(i-k,j-\ell) \tag{A.4}$$

Now suppose for an order P+m the modeling procedure of Section 3 finds the model:

$$x(i,j) = \sum_{\substack{k=0 \\ k+\ell \neq 0}}^{P+m} \sum_{\ell=0}^{P+m} \beta_{k\ell}\, x(i-k,j-\ell) + U'(i,j) \tag{A.5}$$

$$E\, U'(i,j) = 0$$

$$E\, U'^{2}(i,j) = \sigma'^{2}$$

However the minimum variance criterion of (3.9) necessitates that

$$E\, x(i,j) \mid x(i,j-1), \ldots, x(i-P,j-P), \ldots, x(i-P-m,j-P-m) \tag{A.6}$$

$$= \sum_{\substack{k=0 \\ k+\ell \neq 0}}^{P+m} \sum_{\ell=0}^{P+m} \beta_{k\ell}\, x(i-k,j-\ell)$$

Finally comparison of (A.4) and(A.6) with condition (A.3) necessitates that coefficients $\beta_{k\ell}$ have values:

$$\beta_{k\ell} = \begin{cases} \alpha_{k\ell} & \text{for } k \leq P,\ \ell \leq P \\ 0 & \text{otherwise} \end{cases} \tag{A.7}$$

Substitution of (A.7) in (3.13) will result in

$$\sigma'^{2} = \sigma^{2} \tag{A.8}$$

hence proving the lemma that if the underlying stationary and Gaussian 2-D process can be modeled by a finite order autoregressive model, then the modeling procedure of Section 3.1 will result in that model.

Appendix B

Let the P^{th} order model obtained from the modeling procedure be

$$x(i,j) = \sum_{\substack{k=0 \\ k+\ell \neq 0}}^{P} \sum_{\ell=0}^{P} \alpha_{k\ell}\, x(i-k,j-\ell) + U(i,j) \tag{B.1}$$

$$E\, U(i,j) = 0$$

$$E\, U^2(i,j) = \sigma^2$$

Let vectors Z and W be defined as

$$W^T = (R_{01}\ R_{02}\ \cdots\ R_{10}\ R_{12}\ \cdots\ R_{PP}\ R_{00})^T \tag{B.2}$$

$$Z^T = (\alpha_{01}\ \alpha_{02}\ \cdots\ \alpha_{10}\ \alpha_{12}\ \cdots\ \alpha_{PP}\ \sigma^2)^T$$

Thus the first $(P+1)^2-1$ elements of Z are the same as the elements of the vector $\underline{\alpha}$ in (3.11). This allows us to combine (3.11) and (3.13) and state that the model parameters are found by solving a $(P+1)^2$ system of linear equations of the form

$$A_1\, Z = Q_1 \tag{B.3}$$

where A_1 now is a $(P+1)^2 \times (P+1)^2$ matrix and vectors Z and Q_1 are $(P+1)^2 \times 1$ size vectors. But the elements of A_1 and Q_1 are elements of the vector W, hence the set of equations in (B.3) is also a linear set of equations in R_{01}, R_{02}, ..., R_{10}, R_{12}, ..., R_{PP} and R_{00}. Thus (B.3) can be rearranged to an equavlent form

$$A_2\, W = Q_2 \tag{B.4}$$

where now the elements of A_2 and Q_2 are the various elements of the vector Z or namely the model parameters.

Now suppose the first $(P+1)^2$ correlations that are generated by the model in (B.1) are C_{01}, C_{02}, ... etc. and let

$$\hat{W}^T = (C_{01}\ C_{02}\ \cdots\ C_{10}\ C_{12}\ \cdots\ C_{PP}\ C_{00})^T \tag{B.5}$$

Since $x(\cdot,\cdot)$ is zero mean and stationary, the correlation values C_{01}, C_{02}, ... etc. must satisfy (3.10) and (3.13). This system of linear equations has the form

$$A_2 \hat{W} = Q_2 \tag{B.6}$$

Finally comparison of (B.4) and (B.6) yields

$$W = \hat{W}$$

and thus the proof of the stated property.

Appendix C

Optimal Segmentation Criterion

For the sake of notational simplicity, the following discusson and derivations are presented in a one-dimensional setting. However, each step and the result hold true for two-dimensional signals as well.

In an M class environment $W_1, \ldots, W_M$ let

$$x = \{x_1, x_2, \ldots, x_N\} \tag{C.1}$$

be a set of observed data. The segmentation problem, then, is the process of partitioning x into disjoint subsets $x_1, \ldots, x_M$ and assigning each subset to one of the classes $W_1, \ldots, W_M$ (one or more of the subsets can be empty). In accordance with Baye's criteria of optimality, namely minimization of average loss, the average loss $\mathscr{L}$ incurred by partitioning x into two subsets x_1 and x_2 and assigning x_1 to class w_k and x_2 to class w_ℓ is

$$\mathscr{L} = L\{(w_k, w_\ell), (x_1,x_2)\} \tag{C.2}$$

$$= \sum_{i=1}^{M} \sum_{j=1}^{M} C[(w_k, w_\ell)|(w_i,w_j)]\, p[(w_i,w_j)|(x_1,x_2)]$$

where $C[(w_k,w_\ell)|(w_i,w_j)]$ is the cost associated with assigning x_1, x_2 to the classes w_k, w_ℓ while in fact they belong to classes w_i, w_j, respectively. Assuming a symmetric cost function for C of the form

$$C[(w_k,w_\ell)|(w_i,w_j)] = 1 - \delta(k-i,\ell-j) \tag{C.3}$$

where

$$\delta(k-i,\ell-j) = \begin{cases} 1 & \text{if } k=i \text{ and } \ell=j \\ 0 & \text{otherwise} \end{cases}$$

and substituting (C.3) in (C.2) results in

(C.4)
$$\mathscr{L} = \sum_{i=1}^{M}\sum_{j=1}^{M} P[(w_i,w_j)|(x_1,x_2)] -$$

$$\sum_{i=1}^{M}\sum_{j=1}^{M} \delta(k-i,j-\ell)\ P[w_i,w_j)|(x_1,x_2)]$$

$$= 1 - P[(w_k,w_\ell)|(x_\cdot,x_2)]$$

$$= 1 - \frac{P[(x_1,x_2)|(w_k,w_\ell)]\ P(w_k,w_\ell)}{P(x_1,x_2)}$$

But by definition

(C.5)
$$P(x_1,x_2) = P(x)$$

$$P[(x_1,x_2)|(w_k,w_\ell)] = \begin{cases} P(x_1|w_k)p(x_2|w_\ell) & \text{if } w_k \neq w_\ell \\ P(x_1,x_2|w_k) & \text{if } w_k = w_\ell \end{cases}$$

and assuming independent class occurences

(C.6)
$$P(w_k,w_\ell) = \begin{cases} P(w_k)P(w_\ell) & \text{if } w_k \neq w_\ell \\ P(w_k) & \text{if } w_k = w_\ell \end{cases}$$

So for a given partition x_1, x_2 of x, the classification $x_1 \in w_k$ and $w_2 \in w_\ell$ is optimal if

(C.7)
$$P[(x_1,x_2)|(w_k,w_\ell)]\, P(w_k, w_\ell) \geq$$
$$P[(x_1,x_2)|(w_i,w_j)]\, P(w_i,w_j)$$
$$\text{for all } i, j = 1, \ldots., M$$

where the densities on both sides satisfy (C.5) and (C.6).

The discussion, so far, has been based on what the optimal rule will be if one is given a two segment partition x_1 and x_2 of the set x. However, (C.4) holds true for all possible two segment partitions of x denoted by (x_1', x_2'), ..., (x_1^P, x_2^P) where P is the total number of possible of such partitions. Hence a particular two-segment segmentation of x (partitioning and classification) of the form $x_1^q \in w_k$ and $x_2^q \in w_\ell$ is optimal if

(C.8)
$$P[(x_1^q,x_2^q)|(w_k,w_\ell)]\, P(w_k,w_\ell) \geq$$
$$P[(x_1^m,x_2^m)|(w_i,w_j)]\, P(w_i,w_j)$$
$$\text{for all } i, j = 1, \ldots., M$$
$$\text{and } m = 1, \ldots., P$$

where, again, P is the total number of possible two-segment partitions on x. Finally, (C.2) through (C.8) can be expanded to include three or four or in general s-segment partitions on x.

References

[1] R.M. Haralick, "Statistical and Structural Approaches to Texture," Proceedings of the IEEE, Vol. 67, No. 5, p. 786-804, May 1979.

[2] J.W. Modestino, R.W. Fries, A.L. Vickers, "Texture Discrimination Based Upon an Assumed Stochastic Texture Model," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-3, No. 5, p. 557-580, Sept. 1981.

[3] W.K. Pratt, O.D. Faugeras, A. Gagalomicz, "Application of Stochastic Texture Field Models to Image Processing," Proceedings of the IEEE, Vol. 69, No. 5, p. 542-551, May 1981.

[4] O.D. Faugeras, W.K. Pratt, "Decorrelation Methods of Texture Feature Extraction," IEEE Trans. on Pattern Recognition and Machine Intelligence, Vol. PAMI-2, No. 4, p. 323-332, July 1980.

[5] A. Rosenfeld, Image Modeling, Academic Press, 1981.

[6] A. Habibi, "Two-Dimensional Baysian Estimate of Images," Proceedings of the IEEE, Vol. 60, No. 7, p. 878-883, July 1972.

[7] K. Deguchi, I. Morishita, Two-Dimensional Auto-Regressive Model For the Representation of Random Image Fields, Proceedings of the 6th IEEE International Conference on Pattern Recognition, Munich, Germany, Oct. 19-22, 1982.

[8] D.T. Jostheim, "Autoregressive Modeling and Spectral Analysis of Array Data in the Plane," IEEE Transactions on Geoscience and Remote Sensing, Vol. GE-19, No. 1, p. 15-28, Jan. 1981.

[9] H. Akaike, "Fitting Autoregressive Models for Prediction," Ann. Math. Statistics, Vol. 21, p. 243-247, 1969.

[10] J.L. Shanks, S. Treitel, J.H. Justice, "Stability and Synthesis of Two-Dimensional Recursive Filters," IEEE Trans. on Audio and Electroacoustics, Vol. AU-20, No. 2, p. 115-128, June 1972.

Page intentionally left blank

MIXTURE MODELS FOR DEPENDENT OBSERVATIONS

by

Charles Peters

Department of Mathematics

University of Houston

ABSTRACT

This paper concerns parametric mixture models appropriate for data presented in homogeneous blocks of varying sizes from several unidentified source populations. For most applications, the data elements within each block are dependent. Models are proposed for multivariate normal data incorporating two types of dependence, exchangeability of elements within blocks, and a Markov structure for blocks. The consequences of assuming exchangeability, when in fact the Markov structure holds, are explored. Computational problems for each model are considered, and results of a simple test of the exchangeability hypothesis for LANDSAT data are presented.

Introduction

The mixture density estimation problem considered in this paper may be described as follows. A sample of N independent observations $\Theta_1,\ldots,\Theta_N$ is given, each observation Θ_i consisting of a positive integer n_i (block size) and a $p \times n_i$ matrix

$$X_i = (X_{i1}|\ldots|X_{in_i})$$

whose columns $X_{ij} \in \mathbb{R}^p$ are the basic experimental measurements. Each observation Θ_i comes from one of k populations $\Pi_1,\ldots,\Pi_k$, where k is known but the population of origin of each observation is unknown. Let $q_\ell > 0$ denote the probability that an observation comes from Π_ℓ.

Although the data blocks X_i are independent, the basic measurements X_{ij} within each block are possibly dependent. For applications in remote sensing of agricultural resources, the parameters of primary interest are q_ℓ and $E[n_i|\Pi\ell]$, the mean block size for the ℓth population, where each block is a set of multispectral measurements from a single agricultural field belonging to a single crop class Π_ℓ. The product $q_\ell E[n_i|\Pi_\ell]$ is related to the acreage in the sampling region covered by the class Π_ℓ. The procedures suggested herein are automatic procedures capable of handling large sample sizes N as well as large dimensionality p, with human intervention restricted mainly to a posterior description of classes. It should be possible to modify these procedures, along the lines indicated by Walker [11], to provide for the inclusion of a relatively small number of labelled samples, whose class origins are known, and perhaps to improve upon the estimates of the parameters derived from the labelled samples at

a relatively small additional cost.

Let the observations be generically denoted by $\Theta = (n, X)$ and let $f(n, x \mid \Pi_\ell)$ be the density function of Θ, given that Θ comes from Π_ℓ. Let $f(x \mid n, \Pi_\ell)$ be the density function of X, given n and given that Θ comes from Π_ℓ, and let $f(n \mid \Pi_\ell)$ be the density of n given population Π_ℓ. The mixture density for Θ is

$$f(n, x) = \sum_{\ell=1}^{k} q_\ell f(n, x \mid \Pi_\ell) \tag{1.1}$$

$$= \sum_{\ell=1}^{k} q_\ell f(n \mid \Pi_\ell) f(x \mid n, \Pi_\ell).$$

and the log likelihood for the sample is

$$L = \sum_{i=1}^{N} \log \sum_{\ell=1}^{k} q_\ell f(n_i \mid \Pi_\ell) f(x_i \mid n_i, \Pi_\ell). \tag{1.2}$$

We shall assume particular parametric forms for $f(n \mid \Pi_\ell)$ and $f(x \mid n, \Pi_\ell)$ which are simple enough that they are estimable from (1.2). In particular, we shall consider multivariate normal forms for $f(x \mid n, \Pi_\ell)$ which incorporate either exchangeability of observations within blocks or a first order autoregressive covariance structure. The consequences of the exchangeability hypothesis are presented in some detail, and the possibility of approximating the autoregressive form by exchangeability is considered. Finally, we present the results of a simple test of exchangeability for LANDSAT data.

Two Covariance Hypotheses

Throughout the remainder of this paper it will be assumed that $f(x \mid n, \Pi_\ell)$ is a pxn-variate normal density function. To simplify notation, let $Y = (Y_1 | \ldots | Y_n)$ be a random $p \times n$ matrix having density $f(x \mid n, \Pi_\ell)$. We assume that the column process $Y_1, \ldots, Y_n$ of Y is stationary with unknown mean $\mu_{n\ell}$ and covariance function $\Gamma_{n\ell}(h) = \mathrm{cov}(Y_j, Y_{j+h})$. Next to independence, the simplest assumption about $\Gamma_{n\ell}(h)$ is the exchangeability hypothesis that Y and YW have the same distribution for each $n \times n$ permutation matrix W (to denote this we write $Y \overset{d}{=} YW$). In terms of $\Gamma_{n\ell}$, the exchangeability hypothesis can be formally expressed as

$$E : \Gamma_{n\ell}(h) = \begin{cases} \Sigma_{n\ell} & \text{if } h \neq 0 \\ \psi_{n\ell} + \Sigma_{n\ell} & \text{if } h = 0 \end{cases}$$

for some (unspecified) symmetric $p \times p$ matrices ψ_n and $\Sigma_{n\ell}$ satisfying the conditions that $\psi_{n\ell}$ and $\psi_{n\ell} + n\Sigma_{n\ell}$ are positive definite.

Experiments in image texture generation [9] and studies of spatial correlation in LANDSAT images [4] suggest that the correlation of data elements as a function of spatial separation might be modeled as an autoregressive process of low order. Accordingly, as an alternative to (E), we are led to consider the hypothesis (M) that $\Gamma_{n\ell}(h)$ has a first order autoregressive, or Markov, structure.

$$M : \Gamma_{n\ell}(h) = \Omega_{n\ell}^{\frac{1}{2}} A_{n\ell}^{|h|} \Omega_{n\ell}^{\frac{1}{2}},$$

for some unspecified positive definite $p \times p$ matrix $\Omega_{n\ell}$ and symmetric $p \times n$ matrix A with spectral radius less than one.

The theorems stated below exhibit some consequences of the exchangeability hypothesis which are of importance in computation and in testing the hypothesis. J_n denotes the vector $(1, 1, \ldots, 1)^T_{1 \times n}$, while I_n denotes the $n \times n$ identity matrix. Λ_n' denotes the group of $n \times n$ orthogonal matrices W such that $WJ_n = J_n$.

Theorem 1: If Y is a normally distributed $p \times n$ matrix whose distribution satisfies (E) then $YW \overset{d}{=} Y$ for each member of Λ_n'. If P is an $n \times (n-1)$ matrix satisfying $P^TP = I_{n-1}$ and $P^TJ_n = 0$, then $Z = YP$ has columns $Z_1, \ldots, Z_{n-1}$ which are independently distributed as $N_p(0, \psi_{n\ell})$. The statistics $\bar{Y} = \frac{1}{n} \sum_{i-1}^{n} Y_i$ and $S = \sum_{i=1}^{n} (Y_i - \bar{Y})(Y_i - \bar{Y})$ are independent, $\bar{Y}$ is normal $N_p(\mu_{n\ell}, \Sigma_{n\ell} + \frac{1}{n}\psi_{n\ell})$, and S has the Wishart distribution $W_p(n-1, \psi_{n\ell})$.

As a corollary of Theorem 1, if $n > p + 2$ and (E) is true, then the distribution of

$$F = \frac{n - p - 2}{p} Z_1^T \left(\sum_{j=2}^{n-1} Z_i Z_j^T \right) Z_1$$

is central $F_{p, n-p-2}$. This observation is used as a simple test of (E) described in a later section. It is interesting to note that the distribution of F does not depend essentially on the normality of Y. Using results of A.P. Dawid [5] it can be shown that if Y is any random $p \times n$ matrix such that $YW \overset{d}{=} Y$ for each $W \in \Lambda_n'$, and $\sum_{j=2}^{n-1} Z_i Z_j^T$ is

$\Gamma(h) = \Omega^{\frac{1}{2}} A |h| \Omega^{\frac{1}{2}}$. Let $\hat{f}(y)$ be a normal density satisfying (E) with column mean $\hat{\mu}$ and covariance function

$$\hat{\Gamma}(h) = \begin{cases} \hat{\Sigma} & h \neq 0 \\ \hat{\Sigma} + \hat{\alpha} & h = 0 \end{cases},$$

The degree to which $\hat{f}$ approximates f is measured by the relative entropy

$$H(\hat{f}, f) = \int_{\mathbb{R}^{pn}} f(y) \log \frac{f(y)}{\hat{f}(y)} \, dy \quad .$$

The relationship between this criterion and the L_1 distance, which might be considered more meaningful, is not very clear. The sharpest relationship we have been able to find is given in the next theorem. A corollary of the theorem is that if $H(\hat{f}_j, f) \to 0$ then $\int_{\mathbb{R}^{pn}} |\hat{f}_j - f| \to 0$, a result proved by Geman [8].

Theorem 3: Let $\hat{f}$ and f be arbitrary density functions on $\mathbb{R}^m$. For each $\epsilon > 0$,

$$\frac{1}{2} \int_{\mathbb{R}^m} |\hat{f}(y) - f(y)| \, dy \leq \epsilon + \frac{\epsilon}{\epsilon - \log(1 + \epsilon)} H(\hat{f}, f) \, .$$

It is straightforward to show that if expectations are taken with respect to the true density f, then

(3.1) $$E(\overline{Y}) = \mu,$$

$$\operatorname{cov}(\overline{Y}) = \frac{1}{n} \Omega^{\frac{1}{2}} B \, \Omega^{\frac{1}{2}} ,$$

almost surely positive definite, where Z is defined in Theorem 1, then F has the $F_{p,\ n-p-2}$ distribution. Therefore the test based on F is a distribution free test for the invariance of the distribution of Y under right multiplication by elements of Λ_n'.

By writing out the density of Y under (E) it is easy to see that $(\bar{Y},\ S)$ is sufficient for the family of all normal distributions satisfying exchangeability. Under very mild restrictions the sufficiency of $(\bar{Y},\ S)$ implies (E). Thus, unless (E) holds for all source populations Π_ℓ, some loss of estimation accuracy in the parameters of primary interest (q_ℓ and $E[n_i \mid \Pi_\ell]$) in the mixture model is to be expected when the data within blocks is condensed to block means and scatters.

Theorem 2: Let F be a family of normal distributions of a $p \times n$ matrix Y and suppose that some member of F satisfies (E). If $(\bar{Y},\ S)$ is sufficient for F, then (E) holds for each member of F.

Approximating the Markov Structure by Exchangeability

Even if the Markov assumption is more appropriate for applications, the computations involved in estimating the mixture parameters are very much simpler if exchangeability is assumed. In this section we will show that approximating the Markov form by exchangeability leads to certain conclusions about the dependence on n of the covariance parameters $\psi_{n\ell}$ and $\Sigma_{n\ell}$ of (E).

Let $f(y)$ be the normal density of a $p \times n$ matrix Y whose columns satisfy the Markov assumption with mean μ and convariance function

and $E(S) = n\Omega - \Omega^{\frac{1}{2}} B\, \Omega^{\frac{1}{2}}$,

where $B = (I - A)^{-1}(I + A) - \frac{2}{n}(I - A)^{-2}A(I - A^n)$.

The log-likelihood for the density $\hat{f}$ is

$$\log \hat{f}(y) = -\frac{n-1}{2}\log|\hat{\psi}| - \frac{1}{2}\log|\hat{\psi} + n\hat{\Sigma}|$$

$$- \frac{1}{2}\operatorname{tr}\hat{\psi}^{-1}S - \frac{n}{2}\operatorname{tr}(\hat{\psi} + n\hat{\Sigma})^{-1}(\bar{Y} - \hat{\mu})(\bar{Y} - \hat{\mu})^T$$

The parameters which maximize the expectation, with respect to f, of $\log \hat{f}(y)$ are

$$\hat{\mu} = E(\bar{Y})$$

$$\hat{\psi} = \frac{1}{n-1}E(S)$$

$$\hat{\Sigma} = \operatorname{cov}(\bar{Y}) - \frac{1}{n(n-1)}E(S).$$

Combining these equations with equations (3.1), and replacing $\hat{\Sigma}$ by the new parameter $\hat{R} = \hat{\psi} + n\hat{\Sigma} = n \operatorname{cov}(\bar{Y})$ we have

Theorem 4: $H(\hat{f}, f)$ is minimized when

$$\hat{\mu} = \mu$$

$$\hat{\psi} = \frac{n}{n-1}\Omega - \frac{1}{n-1}\Omega^{\frac{1}{2}} B\, \Omega^{\frac{1}{2}}$$

$$\hat{R} = \Omega^{\frac{1}{2}} B\, \Omega^{\frac{1}{2}} ,$$

where $B = (I - A)^{-1}(I + A) - \frac{2}{n} A (I - A)^{-2}(I - A^n)$.

Although it is not obvious, these parameters satisfy the required constraints; that is, $\hat{\psi}$ and $\hat{R}$ are positive definite. As $n \to \infty$, $\hat{R}$ and $\hat{\psi}$ tend to constants. This implies that $\hat{\Sigma}$ is $O(\frac{1}{n})$ for large n. We will make use of this observation in the next section.

The maximum value of $E[\log \hat{f}(Y)]$ is

$$- \frac{n - 1}{2} \log|\hat{\psi}| - \frac{1}{2} \log|\hat{R}| - \frac{np}{2} ,$$

where $\hat{\psi}$ and $\hat{R}$ are given in Theorem 4.

For large values of n this is approximately

$$- \frac{n}{2} \log|\Omega| - \frac{1}{2} \log|(I - A)^{-1}(I + A)| - \frac{np}{2} .$$

Since

$$E[\log f(Y)] = - \frac{n}{2} \log|\Omega| - \frac{n - 1}{2} \log|I - A^2| - \frac{np}{2} .$$

we have the following expression, for large values of n, for the minimum entropy:

$$H(\hat{f}, f) \approx - \frac{n}{2} \log|I - A^2| .$$

Estimating the Mixture Parameters

The most successful method for estimating the parameters in a mixture of distributions from a single exponential family is maximum likelihood [10]. When the component distributions of the mixture are parametrized

in the right way, the EM procedure has a very natural and easily implemented formulation [10], [6]. For density functions $f(x \mid n, \Pi_\ell)$ corresponding to the Markov assumption the likelihood equations for the mixture parameters are extremely complicated, and there is no obvious alternative to using a standard optimization procedure to maximize the likelihood function. There are difficulties involved in obtaining exact maximum likelihood estimates with a sample sequence from a single autoregressive series (see [7, p.329] and [1]), and it is reasonable to think that these problems will be compounded in the mixture setting proposed, resulting in multiple solutions, slow convergence, etc. In general, the situation when $f(x \mid n, \Pi_\ell)$ satisfies the exchangeability condition is not much better; however, the special case wherein $\Sigma_{n\ell} = \frac{1}{n}\Sigma_\ell$ and $\psi_{n\ell} = \psi_\ell$, and Σ_ℓ and ψ_ℓ are independent of n, is amenable to solution by the EM procedure. For large values of n these assumptions are consistent with the remarks at the end of the last section, if the Markov assumption holds with parameters independent of n.

Let each $f(x \mid n, \Pi_\ell)$ have the form (E) with mean $\mu_{n\ell} = \mu_\ell$ and covariance parameters $\psi_{n\ell} = \psi_\ell$, $\Sigma_{n\ell} = \frac{1}{n}\Sigma_\ell$. Define $R_\ell = \psi_\ell + \Sigma_\ell$. Then $\frac{1}{n} R_\ell$ is the covariance matrix of the column-mean $\overline{X}$ of an observed block of measurements given that the observation comes from Π_ℓ and given the block size n. Suppose the density $f(n \mid \Pi_\ell)$ is from an exponential family

$$f(n \mid \Pi_\ell) = C(\lambda_\ell)h(n)e^{F(\lambda_\ell)t(n)} \qquad n = 1, 2,...$$

where the parameter λ_ℓ is the expected value of $t(n)$ under $f(n \mid \Pi_\ell)$,

[3]. From (1.1) and (1.2) the derivative of the log-likelihood with respect to λ_ℓ is

$$(4.1) \qquad \frac{\partial L}{\partial \lambda_\ell} = \sum_{i=1}^{N} \frac{q_\ell f(n_i, X_i|\Pi_\ell)}{f(n_i, X_i)} \left[\frac{C'(\lambda_\ell)}{C(\lambda_\ell)} + F'(\lambda_\ell)t(n_i)\right] .$$

By differentiating the equation

$$\sum_n C(\lambda_\ell)h(n)e^{F(\lambda_\ell)t(n)} = 1$$

with respect to λ_ℓ, one sees that

$$\frac{C'(\lambda_\ell)}{C(\lambda_\ell)} = - F'(\lambda_\ell)\lambda_\ell$$

(see [3]). Hence $\frac{\partial L}{\partial \lambda_\ell} = 0$ if and only if

$$(4.2) \qquad \lambda_\ell = \sum_{i=1}^{N} \frac{f(n_i, X_i|\Pi_\ell)}{f(n_i, X_i)} t(n_i) \Big/ \sum_{i=1}^{N} \frac{f(n_i, X_i|\Pi_\ell)}{f(n_i, X_i)} .$$

Similarly, by considering $\frac{\partial L}{\partial q_\ell}$, one sees that for a maximum of L we must have

$$(4.3) \qquad q_\ell = \frac{1}{N} \sum_{i=1}^{N} \frac{q_\ell f(n_i, X_i|\Pi_\ell)}{f(n_i, X_i|\Pi_\ell)} .$$

Now let $\bar{X}_i$ and S_i be the mean and scatter of the columns of X_i. Then

$$\frac{\partial}{\partial \mu_\ell} f(n_i, X_i|\Pi_\ell) = f(n_i, X_i|\Pi_\ell)\left[-\frac{n_i}{2} R_\ell^{-1}(\bar{X}_i - \mu_\ell)\right]$$

$$\frac{\partial}{\partial \psi_\ell} f(n_i, X_i|\Pi_\ell) = f(n_i, X_i|\Pi_\ell)\left[-\frac{n_i - 1}{2} \psi_\ell^{-1} + \frac{1}{2}\psi_\ell^{-1} S_i \psi_\ell^{-1}\right]$$

$$\frac{\partial}{\partial R_\ell} f(n_i, X_i|\Pi_\ell) = f(n_i, X_i|\Pi_\ell)\left[-\frac{1}{2} R_\ell^{-1} + \frac{n}{2} R_\ell^{-1}(\bar{X}_i - \mu_\ell)(\bar{X}_i - \mu_\ell)^T R_\ell^{-1}\right] .$$

From these equations it follows that the derivatives of L with respect to μ_ℓ, ψ_ℓ and R_ℓ all vanish when

$$(4.4) \quad \mu_\ell = \sum_{i=1}^{N} n_i \frac{f(n_i, X_i|\Pi_\ell)}{f(n_i, X_i)} \bar{X}_i \Big/ \sum_{i=1}^{N} n_i \frac{f(n_i, X_i|\Pi_\ell)}{f(n_i, X_i)} ,$$

$$(4.5) \quad \psi_\ell = \sum_{i=1}^{N} \frac{f(n_i, X_i|\Pi_\ell)}{f(n_i, X_i)} S_i \Big/ \sum_{i=1}^{N} (n_i - 1)\frac{f(n_i, X_i|\Pi_\ell)}{f(n_i, X_i)} ,$$

$$(4.6) \quad R_\ell = \sum_{i=1}^{N} \frac{f(n_i, X_i|\Pi_\ell)}{f(n_i, X_i)} n_i(\bar{X}_i - \mu_\ell)(\bar{X}_i - \mu_\ell)^T \Big/ \sum_{i=1}^{n} \frac{f(n_i, X_i|\Pi_\ell)}{f(n_i, X_i)} .$$

The iterative procedure suggested by equations (4.2)-(4.6), namely, evaluating the right hand sides with the estimates $\lambda_\ell^{(j)}$, $q_\ell^{(j)}$, $\mu_\ell^{(j)}$, $\psi_\ell^{(j)}$, $R_\ell^{(j)}$ at the j<u>th</u> step, to obtain the estimates $q_\ell^{(j+1)}$, $\mu_\ell^{(j+1)}$, $\psi_\ell^{(j+1)}$, $R^{(j+1)}$, at the (j+1)<u>st</u> step, can be shown to be a slightly modified EM procedure (see [10], and [6]).

Testing the Exchangeability Hypothesis

Standard testing procedures for the two covariance hypotheses considered would require large block sizes n_i and a large sample of observations segregated as to block size and type. The remarks at the end of the second section concerning the distribution of the statistic F under the hypothesis (E) suggest a test which is much easier to implement. For the ith block of measurements X_i, let $Z_i = (Z_{i1}|\ldots|Z_{i,\,n_i-1}) = X_iP_i$, where P_i is a $n_i \times (n_i - 1)$ matrix satisfying the conditions given in Theorem 1. Let

$$F_i = \frac{n_i - p - 2}{p} Z_{i1}^T \left(\sum_{j=2}^{n_i - 1} Z_{ij}Z_{ij}^T \right)^{-1} Z_{i1}$$

If (E) holds for all classes then each F_i is distributed as $F_{p,\,n_i-p-2}$. Thus the number of observed blocks for which F_i falls in some given quantile range of its distribution can be tabulated and compared to its expected value. Table 1 shows these comparisons for 216 quasi-fields of LANDSAT agricultural data from LACIE segment 1645 and 57 quasi-fields from LACIE segment 1633. The quasi-fields are those found by an automatic image segmentation program (AMOEBA) and may not be representative of real agricultural fields. The given χ^2 goodness of fit statistics are significant at levels between 10% and 20%. The hypothesis (E) appears to be rather weakly disconfirmed for this data.

TABLE 1 - Disbribution of F-Ratios

Segment 1645 - 216 Fields

Percentiles	0 - 5%	5 - 10%	10 - 90%	90 - 95%	95 - 100%
Number	18	14	163	9	12
Frequency	8.2%	6.5%	75.5%	4.2%	5.6%

$\chi^2 = 6.72$

Segment 1633 - 57 Fields

Percentiles	0 - 5%	5 - 10%	10 - 90%	90 - 95%	95 - 100%
Number	6	1	44	4	2
Frequency	10.5%	1.3%	77.7%	7.0%	3.5%

$\chi^2 = 5.45$

Appendix

Proofs of the Theorems

Proof of Theorem 1: The covariance of Y can be written as $\psi_{n\ell} \otimes I_n + \Sigma_{n\ell} \otimes J_n J_n^T$, where $\otimes$ denotes the kronecker product. For $W \in \Lambda_n'$, $YW = I_p \otimes W^T(Y)$ has covariance $(I_p \otimes W^T)(\psi_{n\ell} \otimes I_n + \Sigma_{n\ell} \otimes J_n J_n^T)(I_p \otimes W)$ $= \psi_{n\ell} \otimes I_n + \Sigma_{n\ell} \otimes J_n J_n^T$. The mean of YW is $\mu_{n\ell} J_n^T W = \mu_{n\ell} J_n^T$. Therefore, $YW \overset{d}{=} Y$. By a similar argument, if $P^T J_n = 0$, $P^T P = I_{n-1}$ and $Z = YP$, then $E(Z) = 0$ and $\operatorname{cov}(Z) = (I_p \otimes P^T)(\psi_{n\ell} \otimes I_n + \Sigma_{n\ell} \otimes J_n J_n^T)(I_p \otimes P) =$ $\psi_{n\ell} \otimes I_{n-1}$. Therefore the columns of Z are independently distributed as $N_p(0, \psi_{n\ell})$. To prove the last assertion let

$$Q = (n^{-1} J_n \mid P)_{n \times n}$$

where P has the same properties as above. In block form, the covariance of $YQ = (\overline{Y} \mid Z)$ is

$$\left(\begin{array}{c|c} \frac{1}{n}\psi_{n\ell} + \Sigma_{n\ell} & 0 \\ \hline 0 & \psi_{n\ell} \otimes I_{n-1} \end{array} \right)$$

Therefore, $\overline{Y}$ and Z are independent and $\overline{Y} \sim N_p(\mu_n , \frac{1}{n}\psi_{n\ell} + \Sigma_{n\ell})$. Moreover, $S = ZZ^T$ and by the first part of the theorem $S \sim W_p(n-1, \psi_{n\ell})$.

Proof of Theorem 2: Let f_0 be a density function in F satisfying the hypothesis (E). Define

$$h_f(y) = f(y) \,/\, f_0(y)$$

for $f \in F$. By a version of the Neyman-Fisher theorem (Theorem 6.1 of [2]), if $(\overline{Y}, S)$ is sufficient,

$$h_f(y) = g_f(\overline{y}, S)$$

almost everywhere, where g_f is a Borel measureable function on the space of $(\overline{Y}, S)$. For a given $f \in F$ and $W \in \Lambda_n'$, the set

$$U = \{y \mid h_f(y) \neq h_f(yW)\}$$

is an open set contained in $B_1 \cup B_2$, where

$$B_1 = \{y \mid h_f(y) \neq g_f(\overline{y}, S)\} ,$$

and

$$B_2 = B_1W^T = \{y \mid h_f(yW) \neq g_f(\overline{y}, S)\} .$$

By Theorem 1, the pr. measure λ_0 corresponding to f_0 is invariant under Λ_n'. Since $\lambda_0(B_1) = 0$ if follows that $\lambda_0(B_2) = 0$ also, and hence, $\lambda_0(U) = 0$. Therefore U is empty and h_f is an invariant function. This implies that each $f \in F$ is invariant under Λ_n' and must satisfy (E).

Proof of Theorem 3: The function

$$g(\epsilon) = \frac{\epsilon}{\epsilon - \log(1 + \epsilon)}$$

is positive and strictly decreasing on $(0, \infty)$. Thus, if $\frac{\hat{f}}{f} - 1 \geq \epsilon$ we have

$$\frac{\hat{f}}{f} - 1 \le g(\epsilon)\left[\frac{\hat{f}}{f} - 1 - \log\frac{\hat{f}}{f}\right].$$

Therefore,

$$\frac{1}{2}\int_{\mathbb{R}^m} |\hat{f} - f| = \int_{\hat{f}>f} (\hat{f} - f)$$

$$= \int_{0 < \frac{\hat{f}}{f} - 1 \le \epsilon} (\frac{\hat{f}}{f} - 1)f + \int_{\frac{\hat{f}}{f} - 1 > \epsilon} (\frac{\hat{f}}{f} - 1)f$$

$$\le \epsilon + g(\epsilon) \int_{\mathbb{R}^m} [\frac{\hat{f}}{f} - 1 - \log\frac{\hat{f}}{f}]\, f$$

$$= \epsilon + g(\epsilon) \int_{\mathbb{R}^m} f \log(\frac{f}{\hat{f}})$$

$$= \epsilon + g(\epsilon)H(\hat{f}, f) .$$

References

1. Anderson, T.W. and Mentz, R.P., On the structure of the likelihood function for autoregressive and moving average models, J. Time Ser. Anal. 1 (1980) 83-94.

2. Bahadur, R.R., Sufficiency and Statistical decision functions, Ann. Math. Statist. 25 (1954) 423-463.

3. Barndorff-Nielsen, O., Information and Exponential Families in Statistical Theory (John Wiley and Sons, New York, 1978.)

4. Campbell, J.B., Spatial correlation effects upon accuracy of supervised classification of land cover, Photogrammetric Eng. and Rem. Sens. 47 (1981) 335-363.

5. Dawid, A.P., Spherical matrix distributions and a multivariate model, J. Roy. Statist. Soc., Ser. B, 39 (1977) 254-261.

6. Dempster, A.P. et al, Maximum likelihood from incomplete data via the EM algorithm, J. Roy. Statist. Soc. B, 39 (1977) 1-38.

7. Fuller, W.A., Introduction to Statistical Time Series (John Wiley and Sons, New York, 1976.)

8. Geman, S., Sieves for nonparametric estimation of densities and regressions, Reports in Pattern Analysis No. 99 (1981), Div. of Appl. Math., Brown University.

9. McCormick, B., Time series models for texture synthesis, Int. J. Comput. Inform. Sci. 3 (1974) 329-343.

10. Redner, R. and Walker, H.F. Mixture densities, maximum likelihood, and the EM algorithm, Proceedings of the NASA Workshop on Density Estimation and Function Smoothing. Texas A&M University, March 1982.

11. Walker, H.F., The numerical evaluation of the mle for the parameters for a mixture of normal distributions from a partially identified sample. Report 54, Department of Mathematics, University of Houston, 1976.

Page intentionally left blank

AN EMPIRICAL BAYES APPROACH TO SPATIAL ANALYSIS

C. N. Morris
and
H. Kostal

University of Texas, Austin

ABSTRACT

Multi-channel Landsat data are collected in several passes over agricultural areas during the growing season. This paper considers how empirical Bayes modeling can be used to develop crop identification and discrimination techniques that account for spatial correlation in such data. Our approach models the unobservable parameters and the data separately, hoping to take advantage of the fact that the bulk of spatial correlation lies in the parameter process. The problem is then framed in terms of estimating posterior probabilities of crop types for each spatial area. Some empirical Bayes spatial estimation methods developed earlier for this project are used to estimate the logits of these probabilities.

1. Introduction

Multi-channel satellite image data, collected by Landsat, are recorded as a multivariate (four dimensional, for four channels) time series (multiple passovers - five to seven times, spanning a several month long growing season) in two spatial dimensions. These data are part of the "fundamental research data base" described in an appendix to Guseman (1983), each file covering a 30 square nautical mile agricultural site divided into 22,932 pixels (picture segments, which are the measurement units). Also available for each site is "ground truth", being discrete (categorical) parameters indicating crop or ground cover type. Continuous parameters might, additionally, need to be estimated, but only discrete parameters are considered in this paper.

Figure 1 illustrates the set-up, centering on pixel i. There, Y_i might most generally be the 20 = 4 x 5 dimensional vector consisting of responses for four channels and five acquisition times. Here we often will assume that this dimension is reduced, perhaps by using Badhwar transformations (Badhwar, 1982) or a linear summary of the data. Thus Y_i may be univariate or multivariate. Pixel i has coordinates $x_i = (x_{i1}, x_{i2})'$, and ground truth parameter θ_i. These parameters label crop types, which, of course, are highly correlated with those in nearby pixels due to spatial continuity of crop types.

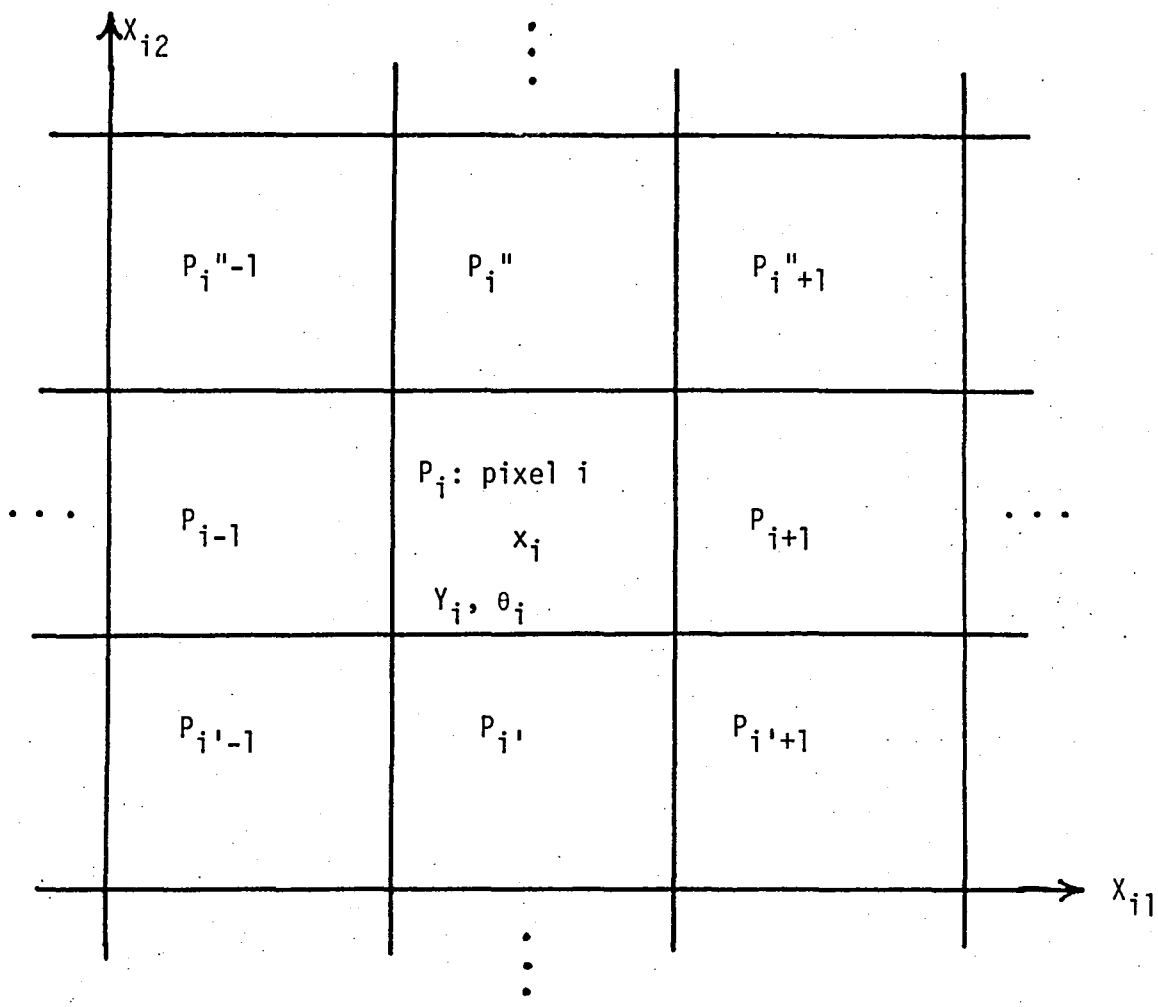


FIGURE 1. Areal problem organized into pixels or pixel-groups, centered at pixel i. Pixel coordinates are $x_i = (x_{i1}, x_{i2})$. Responses are Y_i. True parameters θ_i may be continuous or discrete labels.

Our goal is to estimate the probabilities of each crop type for each pixel, using the data $\{Y_i\}$, incorporating the spatial information. That is, we must determine for each i = 1, ..., n (n = number of pixels), the probabilities

(1.1) $$P(\theta_i = m|\text{data}),\ m = 1, 2, \ldots, M = \text{no. crop types.}$$

Having the classification probabilities (1.1) permits construction of a "probability map" of crop types. This formulation also handles "split pixels" naturally, interpreting probabilities as fractions of each crop type in a pixel.

One can use the probability map to answer many questions. The fraction of a crop type may be obtained for any specified region by summing probabilities for the relevant pixels. Field boundaries may be determined as occurring when classification probabilities change abruptly. Thus we concentrate on the classification probabilities, by pixel. Of course, the spatial methods developed here are applicable to groups of pixels, as well as to pixels themselves, and the best grouping size must be considered. For simplicity of exposition, however, the remainder of the discussion will be framed in terms of pixels.

Formula (1.1) suggests a Bayesian type of calculation. We shall consider Bayesian and empirical Bayesian (EB) approaches to this problem.

2. Empirical Bayes Modeling

Empirical Bayes models, Morris (1983b), involve two stochastic processes: one for the parameters θ, and one for the data Y. In general, we assume that

(2.1) $Y \equiv \{Y_i\}$ has density $f(y|\theta)$ if the true values are $\theta = \{\theta_i\}$

(2.2) $\theta = \{\theta_i\}$ has density $\pi(\theta)$, with $\pi \in \Pi$, a class of "priors."

We call this an empirical Bayes (EB) statistical problem. It is a parametric empirical Bayes (PEB) problem if $\Pi = \{\pi_\alpha$ on $\Theta \equiv$ parameter set: $\alpha \in G\}$, G a parameter set describing the prior.

The marginal distribution

$$h(y|\alpha) = \int_\Theta f(y|\theta)\pi(\theta|\alpha)d\theta \tag{2.3}$$

provides a basis for estimating $\alpha \in G$, and for estimating Bayes rules, e.g., for estimating the Bayes estimator

$$\hat{\theta}_\alpha = E[\theta|Y,\alpha]. \tag{2.4}$$

In Landsat applications, however, the parameters θ will correspond to crop labels, and thus it is more meaningful to replace (2.4) by (2.5) and estimate the posterior probabilities:

(2.5) $$p_Y(\alpha) = P(\theta = m|Y,\alpha).$$

Note that because α (index of the stochastic process determining θ) is unknown, (2.5) is a quantity requiring estimation.

Empirical Bayes theory assumes that the prior distributions (2.2) exist, but $\pi \in \Pi$ is not known (Π will be highly restricted relative to all priors on θ, however). This differs from the Bayes approach in that the data are used to estimate the prior. Methods that result from this approach, however, also often have good frequency operating characteristics, e.g., James-Stein (1962), Efron-Morris (1973, 1975). Spatial applications suggest that the prior densities π incorporate dependence between the parameters θ_i.

The most developed examples of (2.1), (2.2) include $Y_i|\theta_i \overset{ind}{\sim} N(\theta_i,V)$, V known, and $\theta_i|\alpha \overset{ind}{\sim} N(z_i'\beta,A)$, $\alpha = (\beta, A)$, $A \geq 0$, $\beta \in R^p$, z_i a known vector. In spatial applications, z_i will depend on x_i. The estimate of the mean

(2.6) $$\hat{\theta}_i = (1-\hat{B})Y_i + \hat{B}\,(z_i'\hat{\beta})$$

with $\hat{B}$ and $\hat{\beta}$ estimated from the marginal distribution of Y is an empirical Bayes version of Stein's estimator, which has been proved superior to the estimator Y_i. Generalizations and other applications of this theory are reviewed in Morris (1983b) and Efron-Morris (1975).

Empirical Bayes applications to spatial problems have been particularly plentiful. Examples cited in (Morris, 1983b) include:

(a) Revenue Sharing. Fay and Herriot (1979) show that estimates of per capita census income in "small areas" can be improved by combining data from neighboring areas.

(b) Insurance. The insurance industry uses "credibility" (empirical, Bayes) methods to determine to what extent risks in neighboring territories should be used to estimate risks in a particular territory.

(c) Fire Alarms. Carter and Rolph (1974) develop empirical Bayes estimates for spatial data (alarm box locations) to determine better estimates of false alarm rates.

(d) Epidemiology. Efron and Morris (1975) show that empirical Bayes estimates of toxoplamosis prevalence improve substantially upon area-specific estimates in El Salvador.

(e) Forestry. Burk and Ek (1980) improve sample estimates of forestry volume for specific areas by developing empirical Bayes estimates that use information from neighboring areas.

In these cases empirical Bayes methods were demonstrated to work better than standard methods in the most convincing way: by showing that had they been used with real data, that better predictions, and decisions, would have resulted. The demonstrated success of these spatial empirical Bayes applications encourages interest in developing and extending such methodology for remotely sensed image spatial data. However, this latter application is substantially more complex than its predecessors, and therefore substantial additional development will be required.

3. An Approach to Estimating Spatial Probabilities

The empirical Bayes framework models parameters and observations as being realizations of separate stochastic processes. This section considers these processes in more depth, in the context of Landsat data.

3.1. The parameter process. The bulk of Landsat spatial information is captured in the parameter process, i.e. in the distribution of crop labels. Statistical procedures that incorporate this information will perform better than those that ignore it. In practice, the parameter process is unobservable. However, "ground truth" data are available from Landsat experiments, and may be used to construct discriminant prodedures.

The ground truth discrete parameter process is very complicated, involving the distribution of areal segments and the crop types within them. Work on this project by M. Naraghi (on random fields), by H. J. Newton (on spatially homogeneous processes), and by H. P. Decell, Jr. and C. Peters (on special covariance structures), is reported in (Guseman, 1983).

These papers provide approaches to modeling the covariance, or autoregressive structure required for spatial parameter processes. However, we additionally require those to be discrete categorical processes, thereby introducing further modifications.

The simplest labeling processes are those that involve only two labels, "zero-one processes". At various initial stages in this

research, it is desirable to consider simplified models, binary processes being one possible choice. Autologistic models provide another method for modeling zero-one data (Ripley, 1981).

The empirical Bayes approach permits unknown parameters to exist in the prior distribution, requiring that their values be estimated from data available in the actual application. Thus, one needn't completely specify the parameter process.

3.2. The data process. Data $\{y_i\}$ are provided for each pixel, with distributions dependent on the parameter values. Spatial information in this process is important only if it affects the conditional distribution of $\{y_i\}$ given $\{\theta_i\}$. Spatial correlation induced in the $\{y_i\}$ values via the $\{\theta_i\}$ correlations alone is most easily ignored, and therefore is a desirable simplification, if the data permits.

If the spatial aspects of the y values (permitted to be continuous) can be justifiably ignored, then we may use data for which ground truth is available to estimate the density function of the intensity measurements associated with crop type m:

$$f_m(y),\ m = 1, 2, \ldots, M. \tag{3.1}$$

These density functions might be adequately estimated as sample proportions in certain cases, but more effective choices are likely to result from density smoothing procedures, for example, as discussed by for Landsat data by D. W. Scott (for multi-dimensional data) and E. Parzen (univariate and multivariate density quantile estimators),

both in (Guseman, 1983). Also see Wahba (1981).

Now consider the following implemention of these ideas. We will use y_i, the data in pixel i alone, to estimate

(3.2) $$p_i \equiv P(\theta_i = 1|y)$$

among the possible labels m = 1, 2, ..., M. Note that, in this approach, y_i may be multivariate. Any time-aspects of Landsat data are ignored, for now. Let $\pi_1, \ldots, \pi_M$ be the prior probabilities of the M crop types. Then we may calculate (3.2), using Bayes' rule,

(3.3) $$p_i = \frac{\pi_1 f_1(y_i)}{\sum_1^M \pi_j f_j(y_i)} .$$

This may be viewed, without essential loss of generality, as a two-label parameter process, by collapsing the last M-1 labels into one "null" label:

(3.4) $$\pi_o = \pi_2 + \ldots + \pi_M,$$

and

(3.5) $$f_o(y) = \sum_2^M \pi_j f_j(y)/\pi_o .$$

Letting

(3.6) $$z_i \equiv \log(\frac{p_i}{1-p_i}),\ \tilde{z}_i = \log(\frac{\pi_1}{\pi_o})$$

we have, equivalent to (3.3),

$$z_i = \tilde{z}_i + \log\left[\frac{f_i(y_i)}{f_o(y_i)}\right] \quad . \tag{3.7}$$

Thus, the familiar logarithm of likelihood ratio estimates the log-odds (logit) of (3.2).

R. Heydorn and R. Basu (on mixture models) in (Guseman, 1983) adopt a formulation similar to the preceding. They show how to estimate M and the $\pi_1, \ldots, \pi_M$ values by considering the $f_j(y)$ to be normal distributions, and hence, taking (3.5) to be a mixture of normal distributions.

Even if the Heydorn-Basu distributional assumptions must be dropped in favor of more complicated (non-normal, multivariate, etc.) likelihood functions, (3.7) is an easily comprehended function and an optimal data summary. Thus, (3.7) deserves much study in the light of real Landsat data.

We have thus far ignored the time dimension. The values assumed for the Y_i may incorporate this via Badhwar profile features, computed from the "greenness" time series. Alternatively, the likelihood ratio criterion here may indicate other time-summaries, induced by allowing the y_i to be the matrix of time and band dependent values.

4. A Simple Discriminant Example

The simple example here uses the univariate logarithm of likelihood ratio data z_i, (4.2) below, as appropriate data summaries. We then improve them, considered as logit estimates, by incorporating other z_j values from neighboring pixels. In the case of homoskedastic (equal variances and covariances for the groups -- an assumption not in good agreement with Landsat data) normal distributions and M=2, the z_i are simply Fisher's discriminant functions. They are thus normally distributed and are candidates for continuous parameter empirical Bayes estimation, as described for the $\{y_i\}$ values of section 2.

For independent homoskedastic normal measurements

$$y_i \sim N(\mu_m, \sigma^2), \tag{4.1}$$

where m is one of two labels, 0 or 1, depending on which label applies in pixel i, it is easy to show that

$$z_i = \tilde{z}_i + \delta\left[\frac{y_i - \bar{\mu}}{\sigma}\right] \tag{4.2}$$

with $\tilde{z}_i = \log(\pi_1/\pi_0)$, $\bar{\mu} = (\mu_1 + \mu_0)/2$, $\sigma^2 = \mathrm{Var}(y_i)$, $\delta = (\mu_1 - \mu_0)/\sigma$. Given (4.2), we estimate p_i as

$$\hat{p}_i = \exp(z_i)/[1 + \exp(z_i)]. \tag{4.3}$$

Table 1 lists the $\hat{p}_i$ as the probability of soybean ($\theta = 1$) versus an unassigned category ($\theta=0$), taking $\pi_0=\pi_1=.5$. Here $\delta = 1.5$, $\bar{\mu} = 52$ and $\sigma = 6$ are estimated from a small amount of Band 3, Acquisition 4 data from one transect. (This example is kept quite simple in order to illustrate the concepts most clearly.)

Table 1

Thirteen pixels, in one west-east transect, first six unassigned, last six soybean, middle pixel split. y_i = Band 3 value from Acquisition 4 (July 1978). $\hat{p}_i$ = probability of soybean using y_i only. p_i^* is based on three point smoothing of the y_i values (y_i^*). Average p_i^* error slightly improves on average of $\hat{p}_i$ for estimating true θ_i (average errors are .22 and .24). $\bar{p}_i$ estimates use strong spatial information involving prior knowledge of groups of six pixels, with average error .03. See text.

Pixel i	True θ	y_i	$\hat{p}_i$	y_i^*	p_i^*	$\bar{p}_i$
1	0	38	.03	39.3	.04	.00001
2	0	42	.08	41.3	.06	.00001
3	0	44	.12	44.3	.13	.00001
4	0	47	.22	46.7	.21	.00001
5	0	49	.32	47.7	.25	.00001
6	0	47	.22	46.0	.18	.00001
7	.5	42	.08	44.3	.13	.08
8	1	44	.12	46.3	.20	.9997
9	1	53	.56	52.0	.50	.9997
10	1	59	.85	58.7	.84	.9997
11	1	64	.95	61.3	.91	.9997
12	1	61	.90	63.0	.94	.9997
13	1	64	.95	63.0	.94	.9997
Average $\lvert\theta_i - p_i\rvert$:			.24		.22	.03

Stein-type estimators, described later, would shrink the logit values z_i toward a smoothed version of the z_i. Here we smooth by using a three point moving average z_i^* involving the z - values in the preceding and next pixel along the transect as recorded in Table 1. We would ordinarily expect to use neighboring pixels to the north and south too, but did not do so in this simple example involving just one transect. The probabilities p_i^* are in average slightly closer to the true θ_i than are the $\hat{p}_i$. The amount of shrinkage toward z_i^* is estimated to be full (B = 1), in this example, and thus p_i^* is also the Stein, or empirical Bayes, estimator. However, the shrinking factor used, in Morris (1983b), and discussed here in Section 5, assumes the y_i, given the θ_i, to be independent. In these data, the y_i appear to be spatially correlated, and, if so, shrinking factors accounting for this must be developed.

The $\hat{p}_i$ in Table 1 can be improved enormously if one has more spatial information. Suppose, for example, that we know that the last six pixels are the same: either <u>all</u> are soybean, or <u>none</u> are soybean. The z_i values are then should be summed over the six pixels before computing the estimate of the soybean probability. That probability, called $\bar{p}_i$ in Table 1, is .9997 for each of the last six pixels. Compare this with $\hat{p}_i$ = .12 for i = 8! We also get $\bar{p}_i$ = .00001 as the soybean probability in the first six pixels ("unassigned"). The only non-negligable error is the $\bar{p}_7$ = .08 value for the middle (split) pixel.

Weaker forms of spatial information than that just discussed can, and should, be used. For example, suppose it were known that the 13 pixels in Table 1 begin with pixels in the unassigned category, and switch to soybeans after a random pixel position "I". Then, assuming equally likely probabilities for I = 1, ..., 12 a priori, and independent y_i values, the posterior probabilities of I are proportional to the likelihood

$$L(i) = \prod_{i+1}^{13} \frac{f_i(y_i)}{f_o(y_i)}, \quad i = 1, \ldots, 12. \tag{4.4}$$

Formula (4.4) provides probabilistic basis for estimating the change point (areal boundary), and the probabilities. Of course, more complicated models must be considered in realistic situations.

Other forms of logistic regression and discriminant analysis have been proposed to deal with spatial correlation , see, for example, (Switzer, 1980).

5. Shrinkage Estimation Using Affinity Matrices

We developed the notion of "affinity matrices" in an earlier report (Morris, 1983a). These n x n matrices, n = number of pixels, indicate the spatial affinity of pixels. An affinity matrix A is a stochastic matrix, the rows of A being probability vectors: $Ae = e$, $e \equiv (1, \ldots, 1)'$ being the vector of units. Generally A will be a sparse matrix, only a few neighboring pixels being chosen to help estimate any particular one. Estimates z^* like those in (5.1) below are similar to moving average estimates.

The log-odds z_i for pixel i are based on the raw data y_i for that pixel. Stein-type shrinkage estimators, used in conjunction with affinity matrices, and applied to the z_i values, can improve the logit estimate z_i by shrinking z_i to a smoothed value z_i^* computed as an average of responses over neighboring values. That is, letting

$$z^* = Az, \quad z = (z_1, \ldots, z_n)', \tag{5.1}$$

A an affinity matrix, then z^* is a vector of spatially smoothed log-odds estimates. We need to choose between z_i and z_i^*, however. A Stein-type shrinkage rule allows the data to determine the degree to which z^* should be used in preference to z, by employing a shrinking factor B in

$$\hat{z}_i = (1 - B)z_i + Bz_i^*, \tag{5.2}$$

with B calculated as

$$B = \frac{(k-r-2)V}{\Sigma_i (z_i - z_i^*)^2} \quad (5.3)$$

The value r in (5.3) is chosen to account for the use of A, being the trace of A if A is symmetric (Morris, 1983a), and V is the common, known variance of the z_i, being δ^2 in the formulation of (4.2). Minimax results with respect to squared error loss for some estimators of this type are given in (Morris, 1983a).

For spatial data, which are only locally homoegenous, an estimator with a localized shrinkage factor can be expected to improve upon estimators like those of (5.2), (5.3), which use a single, global, shrinkage factor. When the shrinkage factor is calculated separately for each pixel, (5.2) becomes

$$\hat{z}_i = (1 - B_i)z_i + B_i z_i^*. \quad (5.4)$$

If $A \equiv (a_{ij})$, then a choice of B_i is, from (Kostal, 1983),

$$B_i = \frac{d_i V}{\Sigma_j a_{ij}^2 (z_j - z_i^*)^2} \quad (5.5)$$

Here d_i is a suitably chosen positive constant depending on A, allowing the shrinkage in pixel i to be determined by the z_j values for pixels to which the affinity matrix assigns nonzero weight.

6. Empirical Bayes for Time Series Analyses

Thus far, this paper ignores the time-series characteristics of the data, but Landsat data includes a time series $\{y_{it}\}$ for each pixel i (typically, 5 times). For simplicity, we shall first consider the time series $\{y_t\}$ for a given pixel.

A Bayesian structure for time series analysis is given by Harrison and Stevens (1976). Their DLM (dynamic linear model) consists of an observation distribution

$$(6.1) \qquad y_t|F_t, \theta_t, V_t \sim N(F_t\theta_t, V_t),$$

with independent error terms. The parameter distribution, also with independent error terms, is specified as

$$(6.2) \qquad \theta_t|G,\theta_{t-1}, W_t \sim N(G\theta_{t-1}, W_t).$$

The series is initialized by specifying

$$(6.3) \qquad \theta_o|m_o,C_o \sim N(m_o,C_o).$$

The posterior distribution of u_t given $y^t \equiv (y_1', \ldots, y_t')'$ is

$$(6.4) \qquad \theta_t|y^t \sim N(m_t, C_t),$$

where m_t and C_t, given recursively by the Kalman filter, are the posterior mean vector and covariance matrix. The posterior mean m_t provides an estimate of θ_t. These moments cannot be calculated unless all the process parameters are known. If there are unknown

process parameters, such as m_o and C_o (the prior moments), they often can be estimated using the marginal distribution of y^t. These estimates then are used to estimate the posterior mean m_t and thus θ_t.

When several time series $\{y_{it}\}$ follow (6.1) - (6.2) independently with different initializing distributions

$$(6.5) \qquad \theta_{oi}|m_{oi}, C_{oi} \sim N(m_{oi}, C_{oi}),$$

empirical Bayes methods lead to estimates of m_{oi} with smaller mean squared error than those obtained from the marginal distributions of y^t for pixel i alone.

The parameters F_t, V_t, G, W_t, m_o and C_o in (6.1) - (6.3) will depend on the crop type in the pixel. Let K_m denote the model which obtains when the pixel contains crop type m (m = 1, ..., M). The prior probability of each model is the prior probability of each crop type $\pi = (\pi_1, \ldots, \pi_M)'$. Thus the response density (3.1), $f_m(y^t)$, is the marginal density of y^t for pixel i under model K_m. This density would be used to obtain the logit z_i, as in (3.6), Thus incorporating the time-aspect of the Landsat data into the probabilistic structure of Section 3.

References

[1] Badhwar, G. D., J. G. Carnes, and W. W. Austin, "Use of Landsat-Derived Temporal Profiles for Corn-Soybean Feature Extraction and Classification", Remote Sensing of Environment. 12 (1982) 57-79.

[2] Burk, T. E. and A. R. Ek, "Application of Empirical Bayes/James Stein Procedures to Simultaneous Estimation Problems in Forestry Inventory", Minnesota Agricultural Experiment Station Scientific Journal Series, University of Minnesota. Paper No. 11 (1980) 466.

[3] Carter, G. and Rolph, J., "Empirical Bayes Methods Applied to Estimating Fire Alarm Probabilities", J. Am. Stat. Assn. 69 (1974) 880-885.

[4] Efron, B. and Morris, C., "Stein's Estimation Rule and its Competitors--an Empirical Bayes Approach", J. Am. Stat. Assn. 68 (1973) 117-130.

[5] Efron, B. and Morris, C., "Data Analysis Using Stein's Estimator and its Generalization", J. of Am. Stat. Assn. 70 (1975) 311-319.

[6] Fay, R. E. III, and Herriot, R. A., "Estimates of Income for Small Places: An Application of James-Stein Procedures to Census Data," J. Am. Stat. Assn. 74 (1979) 269-277.

[7] Guseman, L. F., ed., Proceedings of the NASA/MPRIA Workshop: Math/Stat. Texas A&M (January 1983).

[8] Harrison, P. J. and C. F. Stevens, "Bayesian Forecasting", Journal of Royal Statistical Society, B48 (1976) 205-247.

[9] James. W. and Stein, C., "Estimation with Quadratic Loss", Proc. 4th Berkeley Symp. on Math. Statist. and Prob., (1961) 361-379. Berkeley: U. of California Press.

[10] Kostal, H., "Localized Shrinkage Factors and Minimax Results", in Proceedings of the NASA/MPRIA Workshop, L. Guseman, ed., Texas A&M (1983) 109-114.

[11] Morris, C., "A Minimax Approach to Spatial Estimation Using Affinity Matrices", in Proceedings of the NASA/MPRIA Workshop, L. Guseman, ed., Texas A&M (1983a) 101-108.

[12] Morris, C., "Parametric Empirical Bayes Inference: Theory and Applications", J. Am. Stat. Assn. 78 (1983b) 47-65, with discussion.

[13] Ripley, B. D., Spatial Statistics, New York: Wiley Press (1981).

[14] Switzer, P., "Extensions of Linear Discriminant Analysis for Statistical Classification of Remotely Sensed Satellite Imagery", Mathematical Geology, 12 (1980) 367-376.

[15] Wahba, G., "Data-based Optimal Smoothing of Orthogonal Series Density Estimates", Annals of Statistics. 9 (1981) 146-156.

Page intentionally left blank

SPLINE CLASSIFICATION METHODS

by

L. F. Guseman, Jr. and L. L. Schumaker
Center for Approximation Theory
Department of Mathematics
Texas A&M University
College Station, Texas 77843

Abstract

This paper is concerned with the use of spline functions in the development of classification algorithms. In particular, a method is formulated for producing spline approximations to univariate density functions when each density function is described by a histogram of measurements. The resulting approximations are then incorporated into a Bayesian classification procedure for which the probability of misclassification can be readily computed. Some preliminary numerical results are presented to illustrate the method.

§1. Introduction.

This paper is concerned with the use of spline functions as a tool in statistical pattern classification algorithms. In particular, we show how splines can be used to estimate the conditional density functions for the classes of interest and to find the associated classification regions. Moreover, we also show how to compute the probability of misclassification associated with the algorithm.

The paper is divided into 6 sections. In Section 2 we discuss the general Bayes classification procedure. In Section 3 we present a method for estimating densities based on polynomial splines. The problems of computing the related classification regions and the probability of misclassification are treated in Sections 4 and 5, respectively. We close the paper with a discussion of examples and future research.

§2. The Bayes Classification Procedure.

Let π_1 and π_2 be distinct classes of interest with known a-priori probabilities α_1 and α_2, respectively. Let $X : \pi_1 \cup \pi_2 \to R$ be a random variable, where $X(w) = x$ is the measurement in R taken from an element w of $\pi_1 \cup \pi_2$. Suppose that the measurements of elements from each of π_1 and π_2 are characterized by density functions p_1 and p_2, respectively. Then the Bayes optimal classifier is defined as follows:

Assign an element w to π_i if its measurement $x = X(w)$ belongs to R_i, $i = 1,2$, where R_1 and R_2 are the Bayes Decision Regions defined by

$$R_1 = \{x \in R: \alpha_1 p_1(x) \geq \alpha_2 p_2(x)\} \tag{2.1}$$
$$R_2 = R \sim R_1.$$

The numerical implementation of this classification procedure requires the determination of the sets R_1 and R_2, which in turn amounts to finding the roots of the equation $\alpha_1 p_1(x) - \alpha_2 p_2(x) = 0$.

Associated with this classification scheme, we define the probability of misclassification (cf. [1,2]) by

$$G = 1 - \int_R \max[\alpha_1 p_1(x), \alpha_2 p_2(x)]dx \tag{2.2}$$
$$= \alpha_1 \int_{R_2} p_1(x)dx + \alpha_2 \int_{R_1} p_2(x)dx .$$

In general, the evaluation of G is a difficult numerical problem, even when p_1 and p_2 are known density functions. One case where G can be computed exactly (along with the Bayes decision regions R_1 and R_2) is the case where p_1 and p_2 are known or estimated univariate normal density functions (cf. [12] and [13]).

In most practical problems, the densities p_1 and p_2 will not be known, and an essential first step in performing Bayesian classification is to compute reasonable estimates of these densities. This is a classical problem in statistical analysis. One of the standard

nonparametric approaches to this problem is to approximate p_1 and p_2 by fitting histograms constructed from measurements taken from elements of the corresponding classes. We discuss this fitting problem in the following section.

§3. Estimating Densities Using Splines.

In this section we discuss the problem of fitting a spline function to a histogram. We begin with some notation. Suppose that $t_1 < t_2 < \ldots < t_{N+1}$ and $h_1,\ldots,h_N$ are given real numbers. These numbers describe a histogram function $h: R \to R$, defined by

$$(3.1)\quad h(x) = \begin{cases} h_i & , \quad \text{if } x \in [t_i,t_{i+1}) \ , \quad 1 \leq i \leq N \\ 0 & , \quad \text{otherwise.} \end{cases}$$

The values t_i, $1 \leq i \leq N+1$, describe the edges of the bins of the histogram, while the values h_i, $1 \leq i \leq N$, describe the height of each bin (cf. Figure 1).

Several techniques have been developed for approximating histograms using spline functions. In what appears to be the first paper on the subject, Bedau [3] constructs the natural spline s which interpolates the histogram in the sense that $s(x_i) = h_i$, $i = 1,\ldots,N$, where $x_i = (t_i + t_{i+1})/2$ are the centers of the bins. Later Boneva, Kendall, & Stefanov [7] and Schoenberg [17] analyzed the problem of finding a spline s (the integral of a natural spline) which fits the histogram in the sense that

$$\int_{t_i}^{t_{i+1}} s(t)dt = h_i(t_{i+1} - t_i) \quad , \quad i = 1,\ldots,N.$$

This condition assures that the area under the spline between each pair of points t_i and t_{i+1} exactly matches the area in the corresponding bin of the histogram. These authors referred to their approximations as histosplines. Schoenberg [17] also considered fitting histograms using smoothing natural splines (and referred to the resulting fits as splinograms). But as observed later by the above authors and others (cf. [8]), a major drawback of methods based on natural splines is the tendency of the fitting spline to dip below the axis near the ends of its support set.

Another approach to fitting a histogram h(x) using splines is to attempt to construct an approximating s(x) as a linear combination of B-splines. To discuss B-splines, we now introduce further notation. Suppose that $y_1 < y_2 < \ldots < y_{n+m}$ is a set of real numbers. Then associated with these points there is a set of B-splines $B_1(x),\ldots,B_n(x)$ with the properties:

$B_i(x)$ is a piecewise polynomial of order m with join points (knots) located at the points $y_i,\ldots,y_{i+m}$;

$B_i(x)$ has m-2 continuous derivatives on R;

$B_i(x)$ is positive on (y_i,y_{i+m}) and vanishes elsewhere;

$B_i(x)$ can be computed efficiently and accurately.

An example of quadratic B-splines (m=3) defined for equally-spaced knots is presented in Figure 2.

B-splines possess a variety of other important properties which make them ideal for approximation purposes, (cf. the books [9,18]). In particular, linear combinations of the form

$$(3.2) \qquad s(x) = \sum_{j=1}^{n} c_j B_j(x)$$

are easy to manipulate on a digital computer. The use of B-spline series of this form also has the advantage that s has support on the interval $[y_1, y_{n+m}]$, and if we choose all the coefficients to satisfy the constraint

$$(3.3) \qquad c_j \geq 0 \quad , j = 1,\ldots,n,$$

then s will also be a nonnegative function.

The first author to use B-spline series as in (3.2) to fit densities appears to be Marsaglia [15]. His approach was to find coefficients $c_1,\ldots,c_n$ to maximize $c_1 + \ldots + c_n$ subject to (3.3) and the constraint that $s(x) \leq p(x)$, all $x \in R$. This can be recast as a linear programming program. Although Masaglia obtained reasonably good results with this technique for smooth functions p, when applied to histogram functions h it tends to produce a spline s which lies substantially under the histogram.

Another approach to constructing a spline s of the form (3.2) fitting a histogram h as in (3.1) is to choose $c_1,\ldots,c_n$ to minimize in some sense the vector $e = [e_1,\ldots,e_N]$ with $e_i = s(x_i) - h_i$, and, as before, $x_i = (t_i + t_{i+1})/2$, $i = 1,\ldots,N$.

Bennett [4,5] considered the cases where the quantity to be minimized is either the ℓ_1 or ℓ_∞ norm of the vector e. Both of these problems (subject to the constraint (3.3)) can be cast as linear programming problems.

Since we are working with histograms as approximations to a density function, it seems to us that it is important to match areas (cf. the above discussion of the methods of splinograms and histosplines). Thus we propose the following alternative to the above spline methods: Find $c_1,\ldots,c_n$ satisfying the constraint (3.3) such that the resulting spline minimizes the expression

$$\|s - h\|_1 = \int_{-\infty}^{\infty} |s(x) - h(x)|\,dx \quad . \tag{3.4}$$

This problem can be recast as:

$$\text{minimize} \quad \varepsilon_1 + \varepsilon_2 + \ldots + \varepsilon_N \tag{3.5}$$

over $c_j \geq 0$, $1 \leq j \leq n$ and $\varepsilon_i \geq 0$, $1 \leq i \leq N$, subject to the constraints

$$-\varepsilon_i \leq \sum_{j=1}^{n} c_j I_{ij} - A_i \leq \varepsilon_i \quad , \quad i = 1,\ldots,N \tag{3.6}$$

where $A_i = h_i(t_{i+1}-t_i)$ is the area of the i-th bin, and

$$I_{ij} = \int_{t_i}^{t_{i+1}} B_j(t)\,dt \quad , \quad i = 1,\ldots,N \text{ and } j = 1,\ldots,n. \tag{3.7}$$

Problem (3.5) is easily translated into a standard linear programming program which can be solved using readily available packages. The numbers I_{ij} in (3.7) can be computed easily by well-known B-spline algorithms

(cf. p. 200 of [18]). The application of this method to a practical problem requires the selection of the order m of the spline as well as the number and location of the knots. In general, we recommend that m be taken to be 2,3 or 4 which leads to linear, quadratic, and cubic splines, respectively.

The selection of the knots is a more difficult problem. So far our numerical tests have been conducted with visual selection of the knots. Our experience suggests that it is reasonable to select the first and last knots at $t_1 - w$ and $t_{N+1} + w$, where w is the average bin width. A reasonable choice for the remaining knots is to place one at the center of each bin for odd orders, and at the bin edges for even orders. If additional knots are desired, they should be added in regions where the histogram has rapid changes in height. It is even possible to insert multiple knots (where a given knot location is selected two or more times). Multiple knots reduce the smoothness of the spline while adding to its flexibility. For a given order m, it is clear that the difference between the spline s and the histogram h measured in the L_1-norm decreases as we add more and more knots.

§4. Finding the Bayes Decision Regions.

Suppose now that we are attempting to build a Bayes classifier corresponding to two classes as in Section 1, and that we have approximations s_1 and s_2 to the corresponding densities p_1 and p_2. We now address the problem of finding the Bayes decision regions

$$\hat{R}_1 = \{x \in R : \alpha_1 s_1(x) \geq \alpha_2 s_2(x)\} \quad \text{and} \quad \hat{R}_2 = R \sim \hat{R}_1 .$$

As noted in Section 1, this problem is equivalent to finding the zeros of the function

$$(4.1) \qquad r(x) = \alpha_1 s_1(x) - \alpha_2 s_2(x) \quad .$$

If s_1 and s_2 are both splines of the same order m based on the same set of knots, then r is also a spline of the same type, and our problem is reduced to locating its zeros. In general, however, we may choose s_1 and s_2 to be splines of different orders (say m_1 and m_2) and based on different knot sequences Δ_1 and Δ_2 . In this case the following observation is important.

THEOREM: If s_i are splines of order m_i corresponding to knot sequences Δ_i, i = 1,2, then the function r defined in (4.1) is a spline of order $m = \max(m_1,m_2)$ with knots $\Delta = \Delta_1 \cup \Delta_2$.

Proof: It is clear that both s_1 and s_2 are piecewise polynomials of order m between the knots of Δ, and it follows that r is also. The fact that r has m-2 continuous derivatives on R is easily checked. □

In order to translate this theorem into a useful algorithm for finding the zeros of r, it is desirable to rewrite both s_1 and s_2 as B-spline expansions in terms of B-splines of order m defined on the knot sequence Δ. Fortunately, there are stable algorithms for converting a B-spline expansion of given degree with given knots to an equivalent

B-spline expansion of another degree with a larger set of knots, (cf. [6,11]). There is no need to examine these algorithms in detail here; we have programmed them for our classification package.

After writing s_1 and s_2 as linear combinations of a common set of B-splines, the problem of finding the zeros of the function r defined in (4.1) reduces to the problem of finding the zeros of a given B-spline expansion. This problem can be attacked by converting the B-spline expansion to a piecewise polynomial representation and then finding the zeros of each polynomial piece. However, more robust and efficient methods for finding zeros of splines are being developed (cf.[14]).

§5. Computing the Probability of Misclassification.

Suppose again that s_1 and s_2 are spline approximations to the densities p_1 and p_2, and suppose that we have found the associated Bayes decision regions $\hat{R}_1$ and $\hat{R}_2$. Then it is clear that an approximation to the probability of misclassification G associated with the densities p_1 and p_2 is given by the expression

$$(5.1) \qquad \hat{G} = \alpha_1 \int_{\hat{R}_1} s_1(x)dx + \alpha_2 \int_{\hat{R}_2} s_2(x)dx.$$

Since both s_1 and s_2 are B-spline series and the sets $\hat{R}_1$ and $\hat{R}_2$ are unions of intervals, to compute $\hat{G}$ we need to be able to integrate a given B-spline series over any given finite interval [a,b]. But there exist standard, highly efficient and accurate algorithms for just this purpose (cf. p. 200 of [18]). We have implemented such a package and (up to roundoff) it produces the values of $\hat{G}$ exactly.

§6. Discussion.

The spline classification algorithm outlined in this paper has been implemented as a FORTRAN package. The package consists of a set of subroutines which performs density fits for given histograms, finds the classification regions, and computes the associated probability of misclassification. In addition, the package includes various subroutines for evaluating, integrating, graphing, and finding the zeros of B-spline series. A FORTRAN implementation of an algorithm of Ravindran [16] is used to solve the linear programming problem (3.5) - (3.7).

Some preliminary fits to the histogram given in Figure 1 were made using quadratic and cubic B-splines. In Figures 3 and 4 we present the fits obtained using quadratic B-splines with different interior knot selections and multiple knots at the endpoints. Figures 5 and 6 present the fits obtained using cubic B-splines with interior knots at the bin centers and multiple knots at different left endpoints. An additional knot was inserted (at 0.0) for the fit presented in Figure 7.

Using the results of the quadratic B-spline fit (Figure 4) to the original histogram and its translate (by 4 units) we determined the Bayes decision regions $\hat{R}_1$ and $\hat{R}_2$ and, assuming equal a priori probabilities, computed the resulting value of $\hat{G}$. These results appear in Figure 8.

In this paper we have concentrated on the classification problem for two classes. It is clear that most of what we have said carries over to the case of three or more classes. In particular, the histograms for each class can be fit with splines in the same way as described here. To find the classification regions now will require pairwise comparison of the spline fits to the densities. The probability of misclassification can then be found as before.

This paper has dealt only with univariate classification. We intend to apply similar techniques to the multivariate case. In particular, we intend to fit multivariate histogram functions using either tensor-product splines or multivariate B-splines defined on triangulations. In either case we expect to be able to accurately find the classification regions and to compute the probability of misclassification.

REFERENCES

1. Anderson, T. W., An Introduction to Multivariate Statistical Analysis, Wiley, New York, 1958.

2. Andrews, H. C., Introduction to Mathematical Techniques in Pattern Recognition, Wiley-Interscience, New York, 1972.

3. Bedau, K. D., Darstellung und Fortschreibung von Einkommenschichtungen unter Verwenderng von Spline-Funktionen, Vierteljahrshefte zur Wirtschaftsforschung (1969), 406-425.

4. Bennett, J. W., Estimation of multivariate probability density functions using B-splines, Ph.D. dissertation, Rice Univ., 1974.

5. Bennett, J. W., de Figueiredo, R. J. P. and Thompson, J. R., Classification by means of B-spline potential functions with application to remote sensing, Rice Report, 1974.

6. Bohm, W., Inserting new knots into B-spline curves, Comput. Aided Design 12 (1980), 199-201.

7. Boneva, L., Kendall, D., and Stefanov, I., Spline transformations: Three new diagnostic aids for the statistical data-analyst, J. Royal Stat. Soc. B, 33 (1971), 1-77.

8. de Boor, C., Appendix to "Splines and histograms" by I. J. Schoenberg, in Spline Functions and Approximation Theory, A. Meir and A. Sharma, eds., ISNM 21, Birkhauser, Basel, 1973, 329-358.

9. de Boor, C., A Practical Guide to Splines, Springer-Verlag, New York, 1978.

10. de Boor, C., and Schumaker, L. L., On calculating with B-splines, II. Integration, in Numerische Methoden der Approximations Theorie, Band 3, ISNM Vol. 30, Birkhauser, 1976, 123-146.

11. Cohen, E., Lyche, T., and Riesenfeld, R., Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics, Comp. Graphics and Image Processing 14 (1980), 87-111.

12. Guseman, L. F., Jr., Peters, C., and Walker, H. F., On minimizing the probability of misclassification for linear feature selection, The Annals of Statist. 3 (1975), 661-668.

13. Guseman, L. F., Jr., and Walker, H. F., On minimizing the probability of misclassification for linear feature selection: A computational procedure, in The Search for Oil, D. B. Owens, ed., Marcel Dekker, New York, 1975, 61-81.

14. Lyche, T. and Schumaker, L. L., Finding the zeros of spline functions, manuscript.

15. Marsaglia, G., One sided approximations by linear combinations of functions, in Approximation Theory, A. Talbot, ed. Academic Press, New York, 1970, 233-242.

16. Ravindran, A., A computer routine for quadratic and linear programming problems, Comm. ACM, 15 (1972), 818-820.

17. Schoenberg, I. J., Splines and histograms, in Spline functions and Approximation Theory, A. Meir and A. Sharma, eds., ISNM 21, Birhauser, Basel, 1972, 277-327.

18. Schumaker, L. L., Spline Functions: Basic Theory, Wiley-Interscience, New York, 1981.

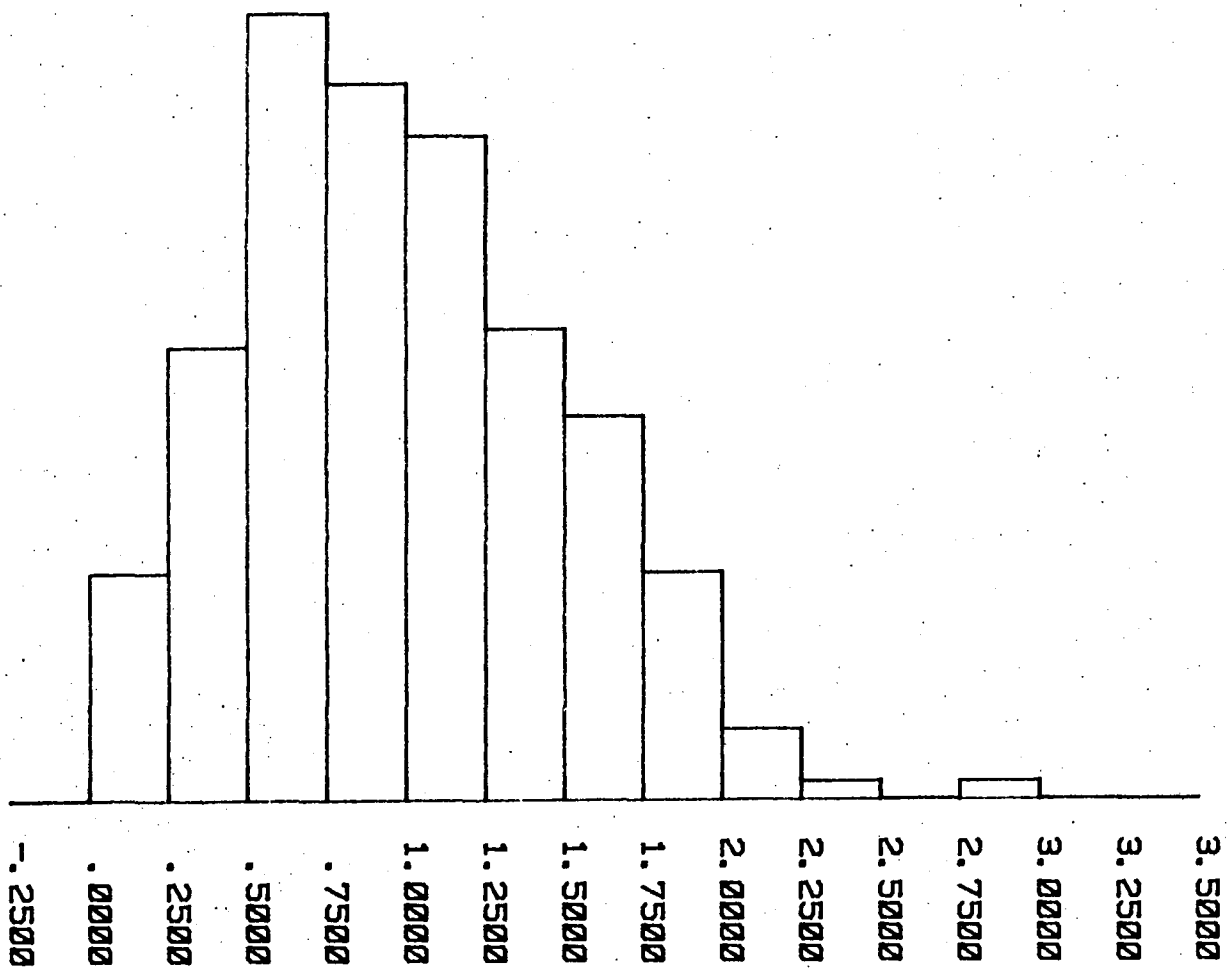


Figure 1: Original Histogram

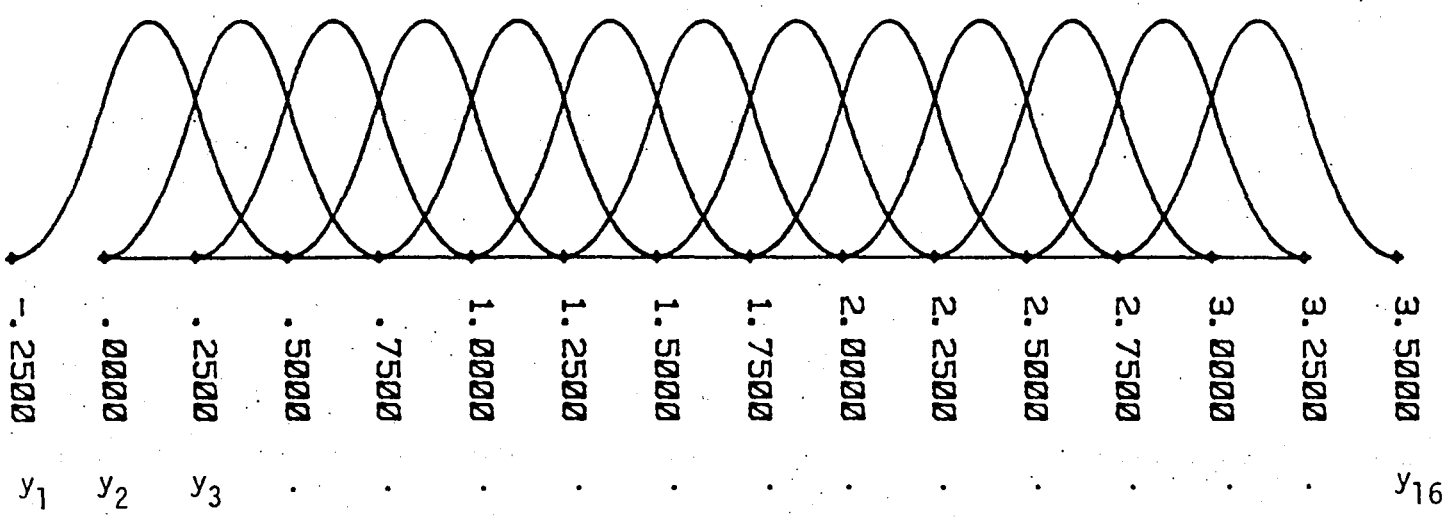


Figure 2: Quadratic B-splines (m=3)

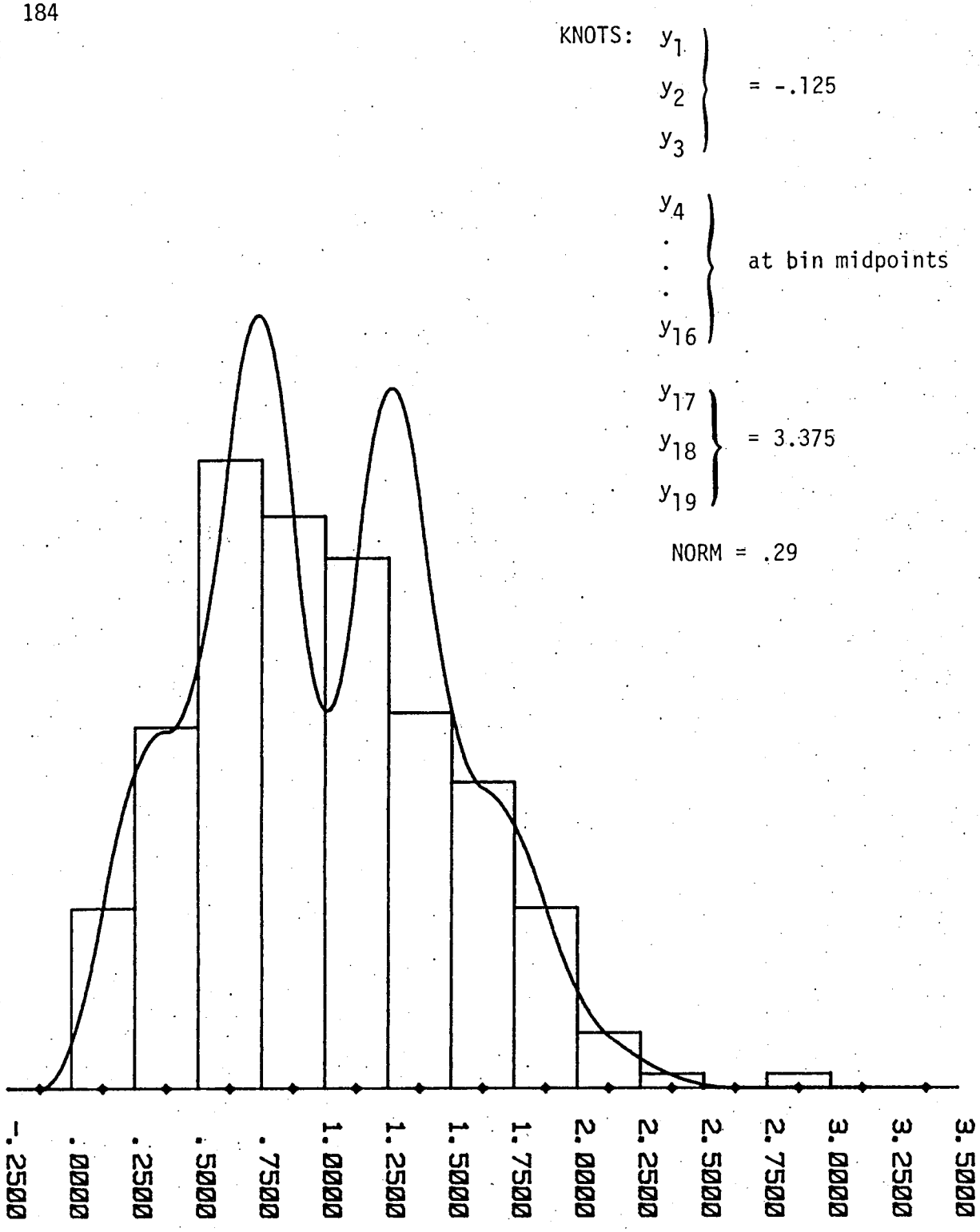


Figure 3: Quadratic B-Spline Fit-Knots at Bin Centers

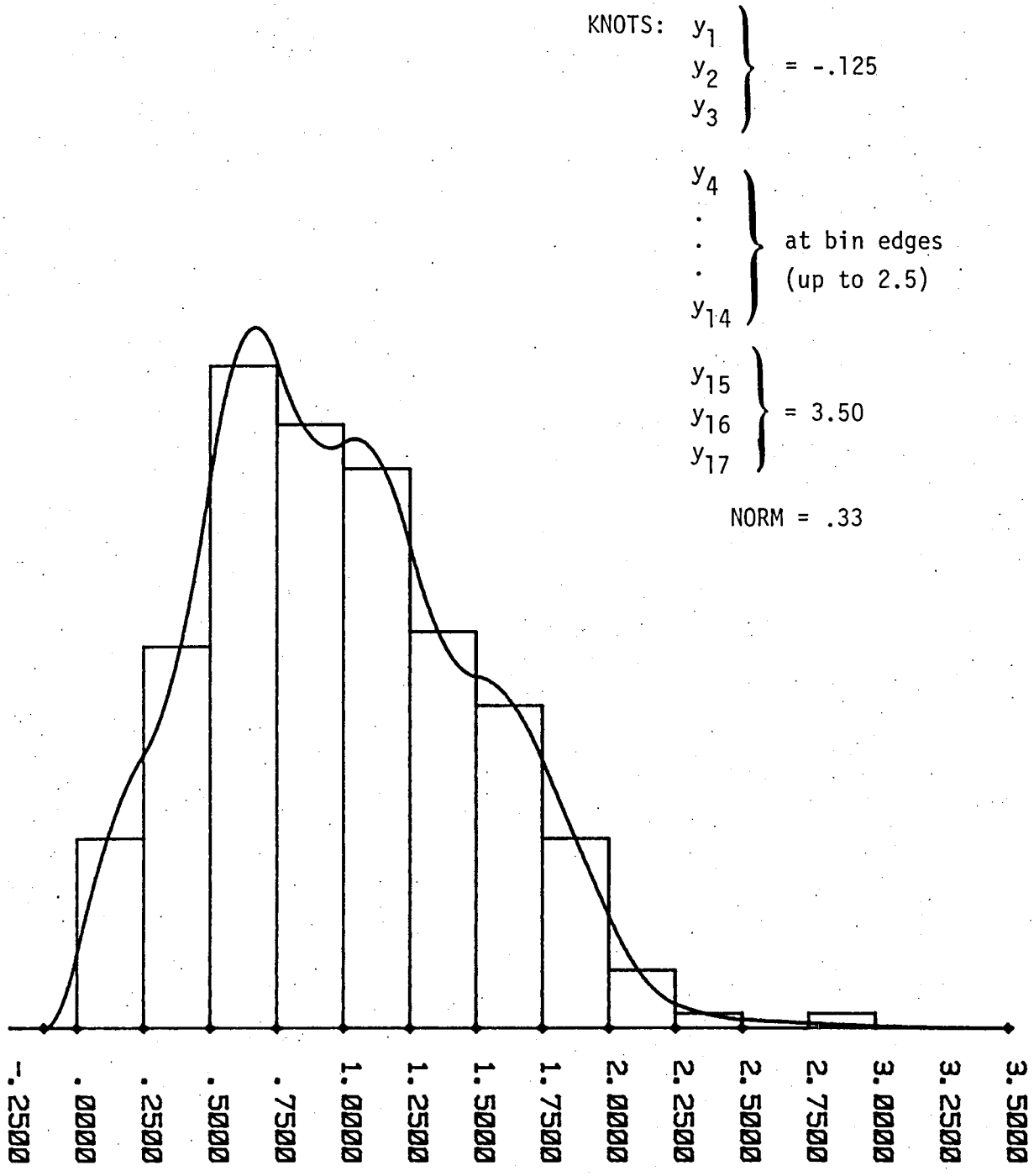


Figure 4: Quadratic B-Spline Fit-Knots at Bin Edges

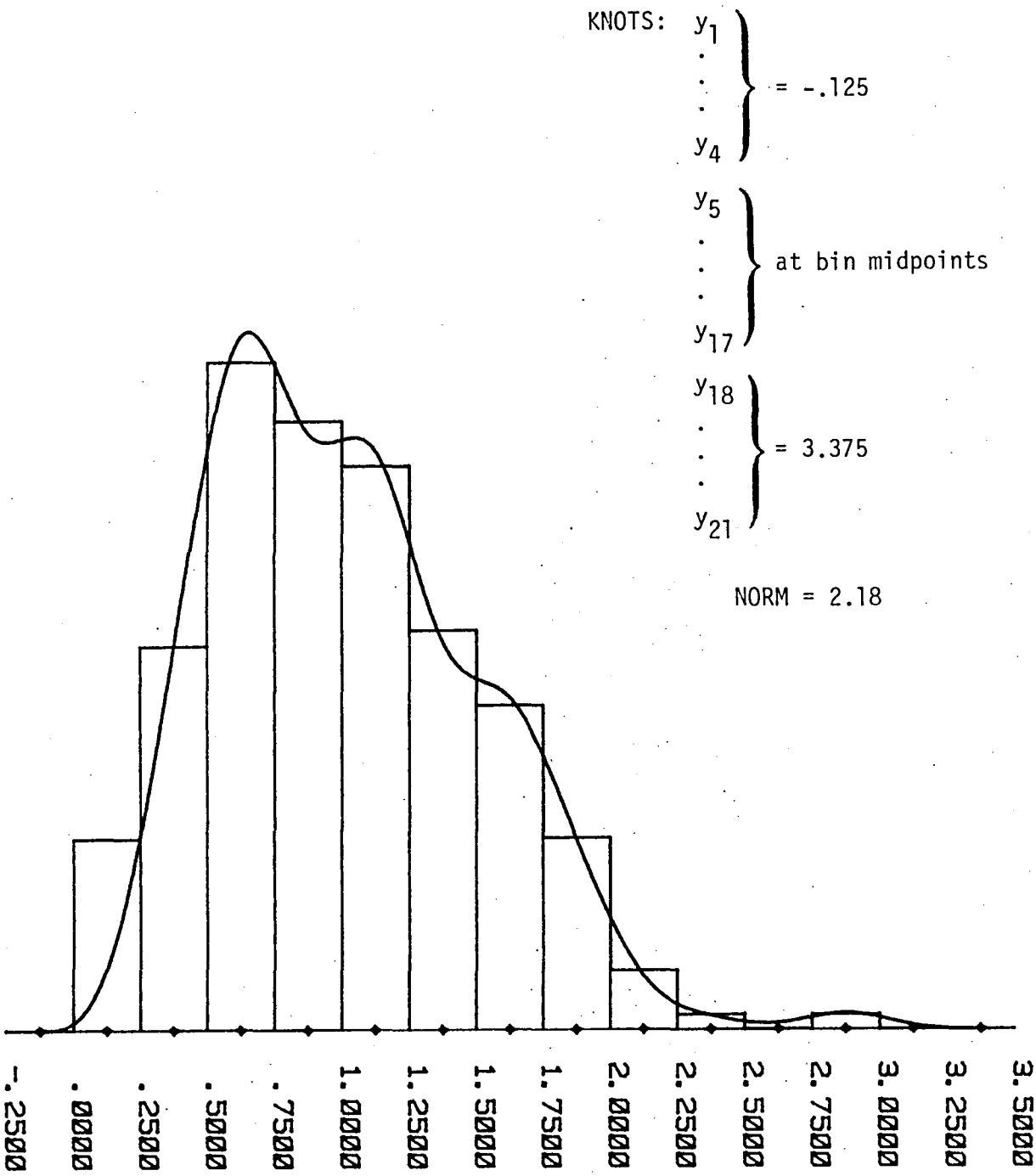


Figure 5: Cubic B-spline Fit Over [-.125, 3.375]

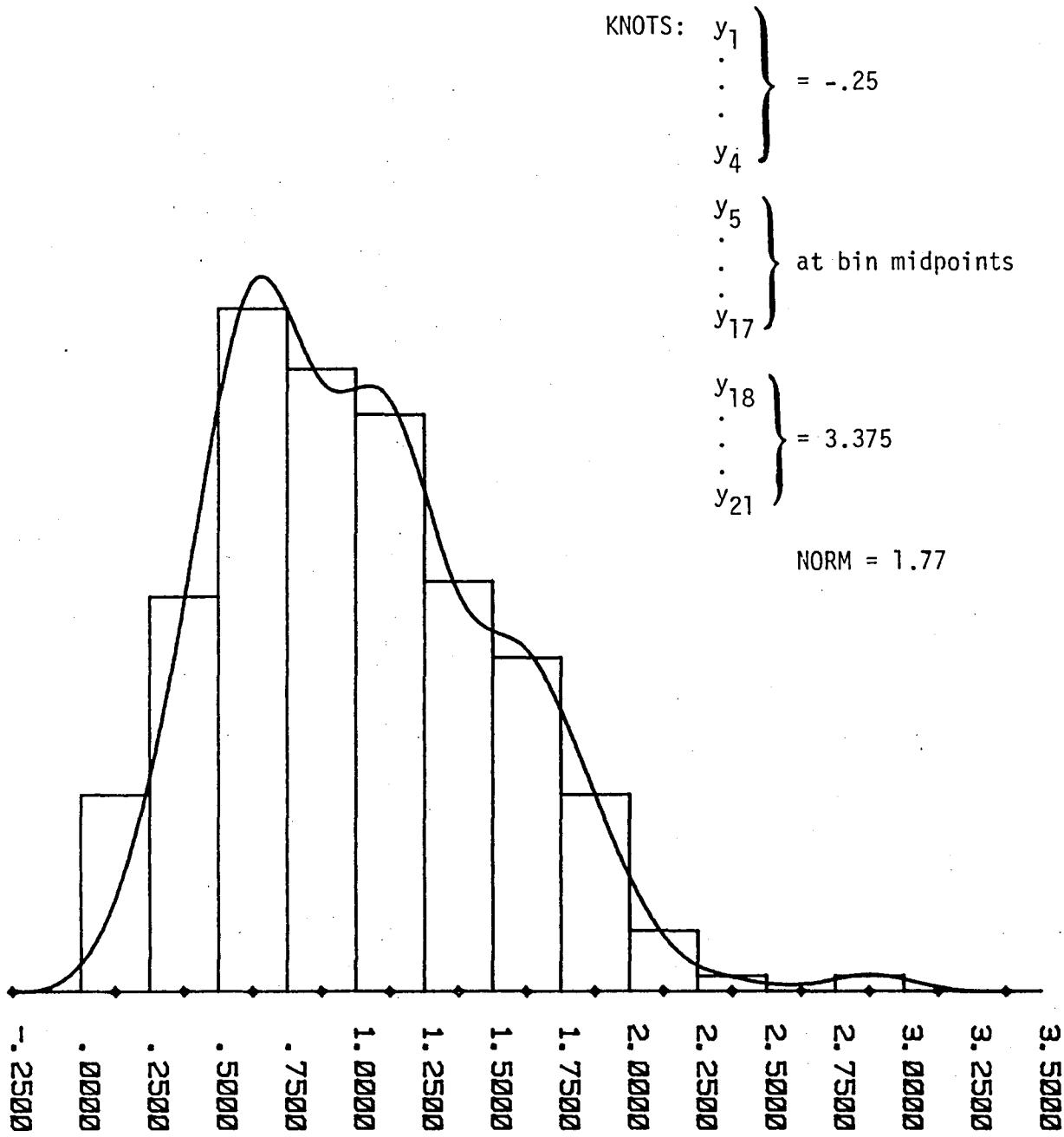


Figure 6: Cubic B-spline Fit Over [-.25, 3.375]

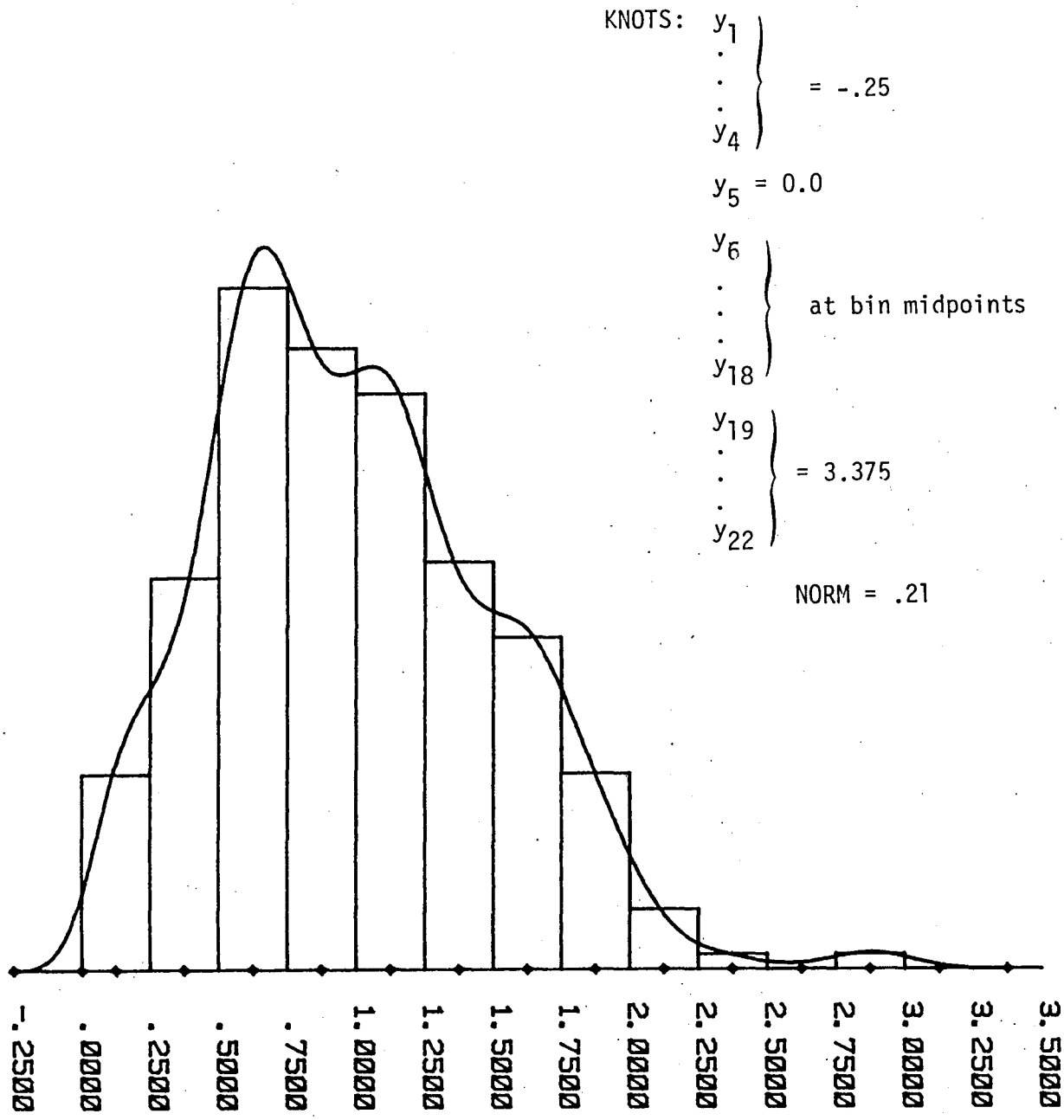


Figure 7: Cubic B-spline Fit Over [-.25, 3.375]
With Additional Knot At 0.0

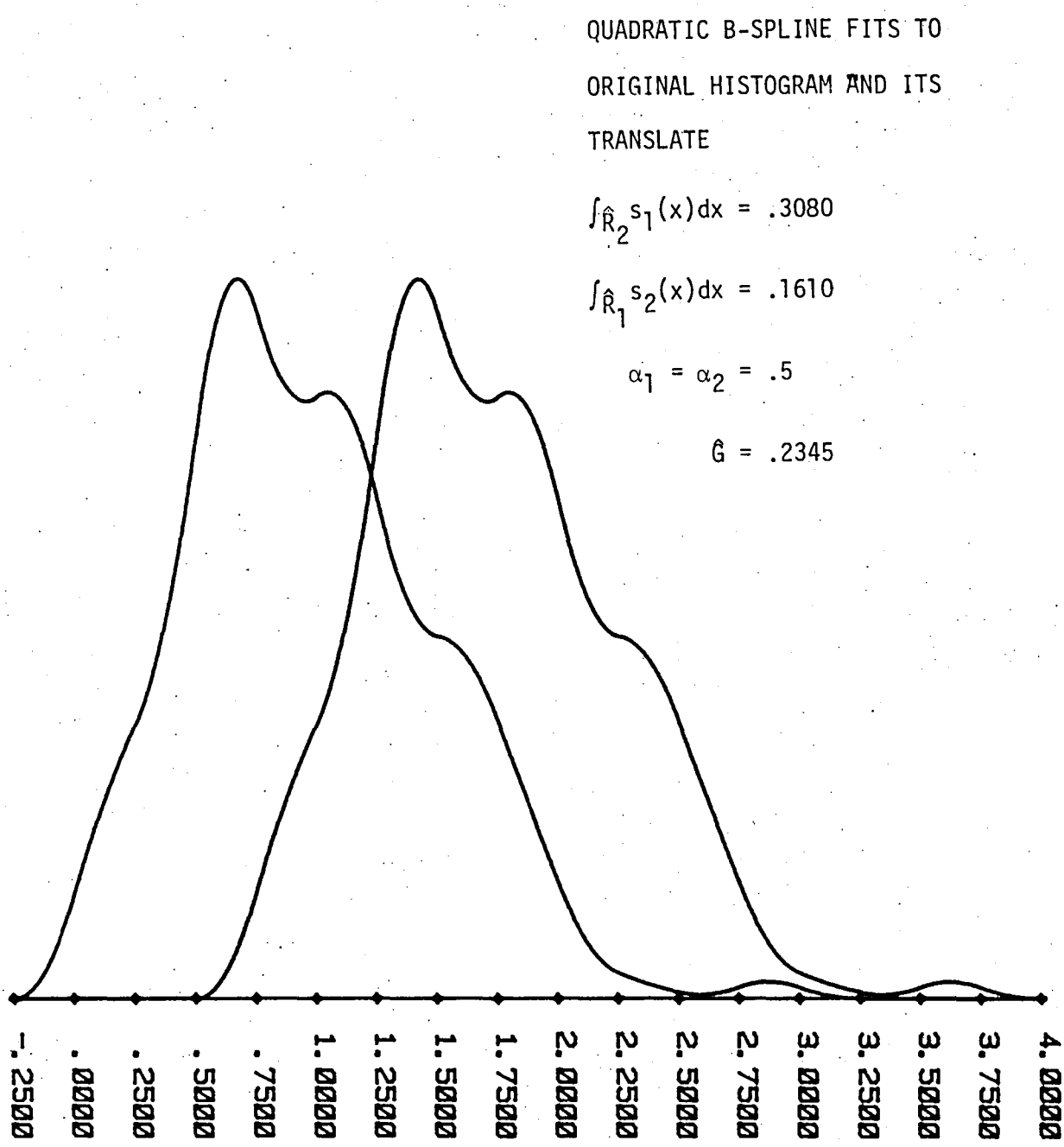


Figure 8: PMC Computation

Page intentionally left blank

QUANTILE DATA ANALYSIS OF IMAGE DATA

Emanuel Parzen
Department of Statistics
Texas A&M University
College Station, Texas 77843

ABSTRACT

Quantile data analysis and functional statistical inference methods are introduced and applied to provide representations of spectral data which may lead to simple statistical discriminators effective for the estimation of ground truth from satelite spectral measurements.

To estimate the ground truth of a pixel, we propose to estimate the probability of each possible ground truth, given observed (estimated) quantile-theoretic statistical characteristics of the multi-spectral image data corresponding to the pixel and its neighboring pixels. This paper describes a research strategy for determining which statistical characteristics discriminate best.

Results are reported of quantile data analysis of an extensive collection of training files of image data.

1. Introduction

To conduct research in image analysis, one must define its data, ends, and means.

The data consists of files. An image file consists of measurements taken on a specified date at a specified 5 x 6 nautical mile site on the earth's surface. A site is divided into a rectangular grid of (more than 20,000) surface elements [approximately 1 acre] called pixels. On each pixel, spectral measurements are made by satelite on four (and perhaps seven) channels [of the electromagnetic energy spectrum]. Each spectral measurement is an integer from 0 to 256.

The ends [goals] of image analysis is to estimate ground truth within the pixel; labels for ground truth include alfalfa, corn, soybeans, sugar beets, spring wheat, spring oats, grass, pasture, trees, water.

A file is called a training file if a ground truth record is available; each pixel is divided into six sub-pixels and ground truth is recorded for each sub-pixel.

The means of image analysis are currently under investigation by many investigators. A probability approach considers ground truth as a parameter [denoted θ]. A formal Bayesian statistical solution to the estimation of ground truth from data is to calculate $p(\theta|\text{data})$, the posterior probability distribution of θ [the ground truth parameter] given the data. A formal maximum likelihood solution to the estimation of ground truth from data consists of two steps: (1) calculate the

likelihood function of θ, which equals $p(\text{data}|\theta)$, the conditional distribution of the data given that it is observed from a pixel with ground truth θ, and (2) use optimization algorithms to determine $\hat{\theta}$, the parameter value which maximizes likelihood. The foregoing formal statistical procedures are often described as being theoretically "optimal." But they may not be "good" in practice in the sense of correctly identifying ground truth with high probability.

To obtain high probability of discrimination, we recommend (1) measuring suitable characteristics of probability models of the data, (2) treating the measured characteristics as new data, and estimating the likelihood function p(measured characteristics of data$|\theta$), and (3) determining characteristics whose distributions for different values of θ are as wide apart as possible [the likelihood function is not flat and its optimum is easily determined].

This paper investigates the use of quantile data analysis to obtained measured characteristics of image data which have good power of discrimination between different values of ground truth. Only univariate analysis methods are used on channel 2 and channel 3 spectral observations. Future research will be concerned with bivariate analysis of the joint distribution of channel 2 and channel 3 measurements. Our approach to quantile data analysis strongly recommends that bivariate analysis be built on a foundation of univariate analysis. Therefore the univariate analysis techniques developed in this paper will not be rendered obsolete by the bivariate techniques to be developed in future research.

2. Outline of Quantile-Data Analysis of a Pixel

Let us describe a proposed method of statistical data analysis based on characteristics of the sample quantile functions of batches of measurements. Given a pixel whose ground truth we would like to estimate, let (t_1,t_2) be its coordinates which represent its position within the rectangular grid of pixels into which the scene has been divided.

Define $A_\nu(t_1,t_2)$, the ν-neighborhood of a pixel, to be the set of pixels with coordinates $(t_1 + j_1,\ t_2 + j_2)$, where $j_1,j_2=0,\pm1,\ldots,\pm\nu$. For example $A_1(t_1,t_2)$ contains 9 pixels, $A_2(t_1,t_2)$ contains 25 pixels, $A_3(t_1,t_2)$ contains 49 pixels.

For k=2 and 3, the channel k measurements of the pixels in $A_\nu(t_1, t_2)$ are collected to form a <u>data batch</u> whose sample quantile function $\tilde{Q}(u)$ is formed. The "measured data characteristics" we associate with a pixel are various characteristics of the sample quantile function of a batch of measurements formed from the pixels surrounding a given pixel. The remainder of this section reviews quantile data analysis and defines the summary statistics that it suggests.

The probability law of a random variable X is usually described by its distribution function $F(x)=Pr[X\leq x]$, $-<\infty x<\infty$, and probability density function $f(x)=F'(x)$. The quantile approach uses [see, for example, Parzen (1983)]

$$(1)\quad Q(u) = F^{-1}(u) = \inf\{x:F(x)\geq u\}\ ,$$

$$(2)\quad q(u) = Q'(u)\ ,$$

(3) $fQ(u) = f(Q(u)) = \{q(u)\}^{-1}$, and

(4) $J(u) = -(fQ)'(u)$.

A quick measure of location is the median $Q(0.5)$. A quick index of scale is the interquartile range $Q(0.75) - Q(0.25)$, formed for the quartiles $Q(0.25)$ and $Q(0.75)$.

Quick measures of distributional shape are provided by values (as u tends to 0 and 1) of the informative quantile function [recently introduced by Parzen].

$$IQ(u) = \frac{Q(u) - Q(0.5)}{2\{Q(0.75) - Q(0.25)\}} \quad , \quad 0 \leq u \leq 1.$$

We cannot emphasize how powerful the IQ function appears to be in practice as a tool for the diagnosis of distributional shapes.

The IQ function is independent of location and scale parameters. It is approximately equivalent to normalizing a quantile function to have the properties $Q(0.5) = 0$, $Q'(0.5) = 1$. The IQ graph of the function provides us at a glance with a vague estimate of tail behavior as defined by tail exponents.

A fundamental description of the tail behavior of distributions is provided by the left tail exponent α_0 and the right tail exponent α_1 defined as follows:

$$fQ(u) = u^{\alpha_0} L_0(u) \text{ as } u \to 0$$

$$fQ(u) = (1-u)^{\alpha_1} L_1(u) \text{ as } u \to 1$$

where $L_0(u)$ and $L_1(u)$ are slowly varying functions.

A function L(u) is slowly varying as $u \to 0$ if, for every $y > 0$,

$$\lim_{u \to 0} \frac{L(yu)}{L(u)} = 1.$$

Tail behavior is defined in terms of a tail exponent as follows:

$\alpha<1$: short tail

$\alpha=1$: medium tail

$\alpha>1$: long tail

Medium tail ($\alpha=1$) distributions are further classified by the value of

$$h_0 = \lim_{u \to 0} \frac{f(u)}{u} \quad , \quad h_1 = \lim_{u \to 1} \frac{f(u)}{1-u} \quad ;$$

the letter h is suggested by the notion of hazard function. We define

$h = 0$: medium-long tail

$0 < h < \infty$: medium-medium tail

$h = \infty$: medium-short tail .

Extensive calculations of informative quantile functions indicate that the value IQ_0 of IQ(u) for u near 0 is a quick indicator of left tail behavior:

$-0.5 \leq IQ_0 < 0$: short left tail,

$-1.0 \leq IQ_0 < -0.5$: medium-short left tail,

$IQ_0 < -1.0$: medium-medium to long left tail.

Similarly the value IQ_1 of IQ(u) for u near 1 is a quick indicator of right tail behavior:

$0 < IQ_1 \leq 0.5$: short right tail,

$0.5 < IQ_1 < 1.0$: medium-short left tail,

$1.0 < IQ_1$: medium-medium to long right tail .

An important family of distributions is the Weibull with shape parameter β . Its quantile function $Q(u)$ is of the form

$$Q(u) = \mu + \sigma Q_o(u)$$

where

$$Q_o(u) = \frac{1}{\beta} \{\log (1-u)^{-1}\}^{\beta} \quad .$$

Its density-quantile

$$f_o Q_o (u) = (1-u) \{\log (1-u)^{-1}\}^{1-\beta} \quad .$$

Its right tail exponent is $\alpha = 1.$, and its left tail exponent is $\alpha_o = 1-\beta$. Insight into the interpretation of informative quantile functions is obtained by computing them for Weibull distributions.

Given data, we distinguish three types of estimators of population parameters, which we call: (1) fully non-parametric, (2) fully parametric, and (3) functional-parametric. Fully non-parametric estimators assume no model, and provide quick estimators. Fully parametric estimators assume a model known up to a finite number of parameters which must be estimated. Functional-parametric estimators are based on methods of functional statistical inference.

A fully non-parametric estimator $\tilde{Q}(u)$ of $Q(j)$, given a sample of n distinct values $X_{1;n} < X_{2;n} < \ldots < X_{n;n}$, is defined by (for $j=1,\ldots,n$)

$$\tilde{Q}(u) = X_{j;n} \quad , \quad \frac{j-1}{n} < u \leq \frac{j}{n} \quad .$$

For a large sample, or for grouped values, we form a histogram before

computing $\tilde{Q}(u)$ by linear interpolation at an equi-spaced grid of values kh, k=1,2,...,[1/h] where usually h = 0.01.

3. Example and Interpretation of a Quantile Data Analysis

To illustrate the uses of measured data characteristics provided by quantile data analysis, let us consider the analysis of a training file which contains both image data and ground truth data. We search through the ground truth file to see what codes appear more than a few times. The codes found to be present corresponded to the ground truth values listed in Table A. For a ground truth value j, we created a data batch consisting of all the channel 2 values observed in a pixel one of whose sub-pixels had a ground truth equal to the value j. We created a similar data batch of channel 3 observations. Table A lists the sample sizes of the number of observations in these data batches and the medians and interquartile ranges of the channel 2 and channel 3 observations. One immediately sees a pattern which might provide a discrimination statistic Δ to be used in determining ground truth. One might be able to readily distinguish the category "grass, pasture, trees" from "corn, soybeans, sugar beets, spring wheat, spring oats" by the values of

$$\Delta_1 = \text{median (channel 3)} - \text{median (channel 2)}$$

$$\Delta_2 = \log \frac{\text{IQ range (channel 3)}}{\text{IQ range (channel 2)}} \quad .$$

The values of these statistics are given in Table A. Note that $\Delta_1 > 2$ for grass, pasture, and trees, and $\Delta_1 < 2$ for crops. Of the crops,

alfalfa is closest in statistical characteristics to grass, pasture, and trees; this conclusion is reached also in Table B.

Table A lists statistics based on comparisons of location and scale estimators; Table B lists discriminators which are based on shape and tail characteristics. We consider the following four characteristics as statistics which might discriminate between (ground truth) distributions:

$$\Delta_3 = \text{MEAN IQ} = \frac{\text{MEAN - MEDIAN}}{2 \times \text{INTERQUARTILE RANGE}}$$

$$\Delta_4 = \text{STD IQ} = \frac{\text{STANDARD DEVIATION}}{2 \times \text{INTERQUARTILE RANGE}}$$

$$\Delta_5 = IQ_0 = IQ(u) \text{ for } u \text{ near } 0$$

$$\Delta_6 = IQ_1 = IQ(u) \text{ for } u \text{ near } 1 \quad .$$

The values of these statistics in this example indicate that trees, pasture, and grass have spectral observations with distributions closer to normal, while crops have spectral observations with distributions further from normal.

It should be strongly emphasized that these empirical patterns found in one file are not intended to be general algorithms applicable to all files. They are presented only as an illustration of the kinds of facts about image data which quantile data analysis proposes to discover through extensive computation on training files.

TABLE A

	Sample Size	Median Channel 2	Median Channel 3	Δ_1 Median (3) -Median(2)
Alfalfa	377	19	20	1
Corn	8,755	15	14	-1
Soybeans	11,000	15	13	-2
Sugar Beets	793	14	12	-2
Spring Wheat	2,296	18	16	-2
Spring Oats	558	18	16	-2
Grass	174	23	26	3
Pasture	-248	-21	28	7
Trees	95	20	24	4

	IQ Range Channel 2	IQ Range Channel 3	Δ_1 Log IQ(3) -Log IQ(2)
Alfalfa	9	16.75	.62
Corn	5	6.5	.26
Soybeans	5	8	.47
Sugar Beets	4	4.5	.12
Spring Wheat	6	9	.41
Spring Oats	8	11	.32
Grass	8	12.5	.45
Pasture	5	13	.96
Trees	6	11	.61

TABLE B

	Mean IQ Channel 2	Mean IQ Channel 3	STD IQ Channel 2	STD IQ Channel 3
Alfalfa	-.08	-.07	.32	.27
Trees	-.08	-.01	.38	.35
Pasture	-.04	-.06	.41	.32
Grass	-.01	.02	.36	.34
Spring Wheat	.07	.11	.38	.41
Spring Oats	.09	.12	.36	.35
Sugar Beets	.14	.06	.42	.49
Corn	.17	.10	.44	.51
Soybeans	.17	.13	.46	.41

	IQ_0 Channel 2	IQ_0 Channel 3	IQ_1 Channel 2	IQ_1 Channel 3
Alfalfa	-.34	-.32	1.05	.68
Trees	-.75	-.72	1.0	.68
Pasture	-1.0	-.76	1.1	.65
Grass	-.75	-.72	.81	.68
Spring Wheat	-.58	-.44	2.08	1.66
Spring Oats	-.43	-.36	1.18	1.0
Sugar Beets	-.37	-.44	2.25	2.55
Corn	-.40	-.46	2.8	3.0
Soybeans	-.40	-.31	2.7	1.93

Note: STD IQ = .37 for normal. Above line characteristics close to normal. Below line characteristics far from normal.

4. Quantile Data Analysis of Statistical Characteristics Estimated from Pixel Nieghborhoods

A program of fundamental research on the quantile data analysis approach to picture segmentation poses many detailed problems for research. This section gives an example of one sample quantile data analysis. (1) Consider all pixels in a file whose ground truth is a specified crop (spring wheat is considered here). (2) For each such pixel form a 5 by 5 neighborhood of pixels (with the specified pixel at the center). (3) For each neighborhood form a data batch of spectral observations (channels 2 and 3 are considered here). (4) For each data batch, form its sample quantile function and estimate typical univariate quantile theoretic statistical characteristics: median, IQR (interquartile range), mean IQ (mean of informative quantile function), STDIQ (standard deviation of IQ function), IQ(.01), IQ(.99), average log spacings (which is a fully non-parametric estimator of entropy of the IQ function), and log σ_0 [where σ_0 is the score deviation, defined as the sum of products of the spacings of the IQ function and a specified density-quantile function $f_0Q_0(u)$]. The specified density quantile functions that we use are the logistic distribution

$$f_0Q_0(u) = u(1-u) \quad ,$$

and the Weibull distribution with quantile shape parameter β [we choose $\beta = 0.7$]

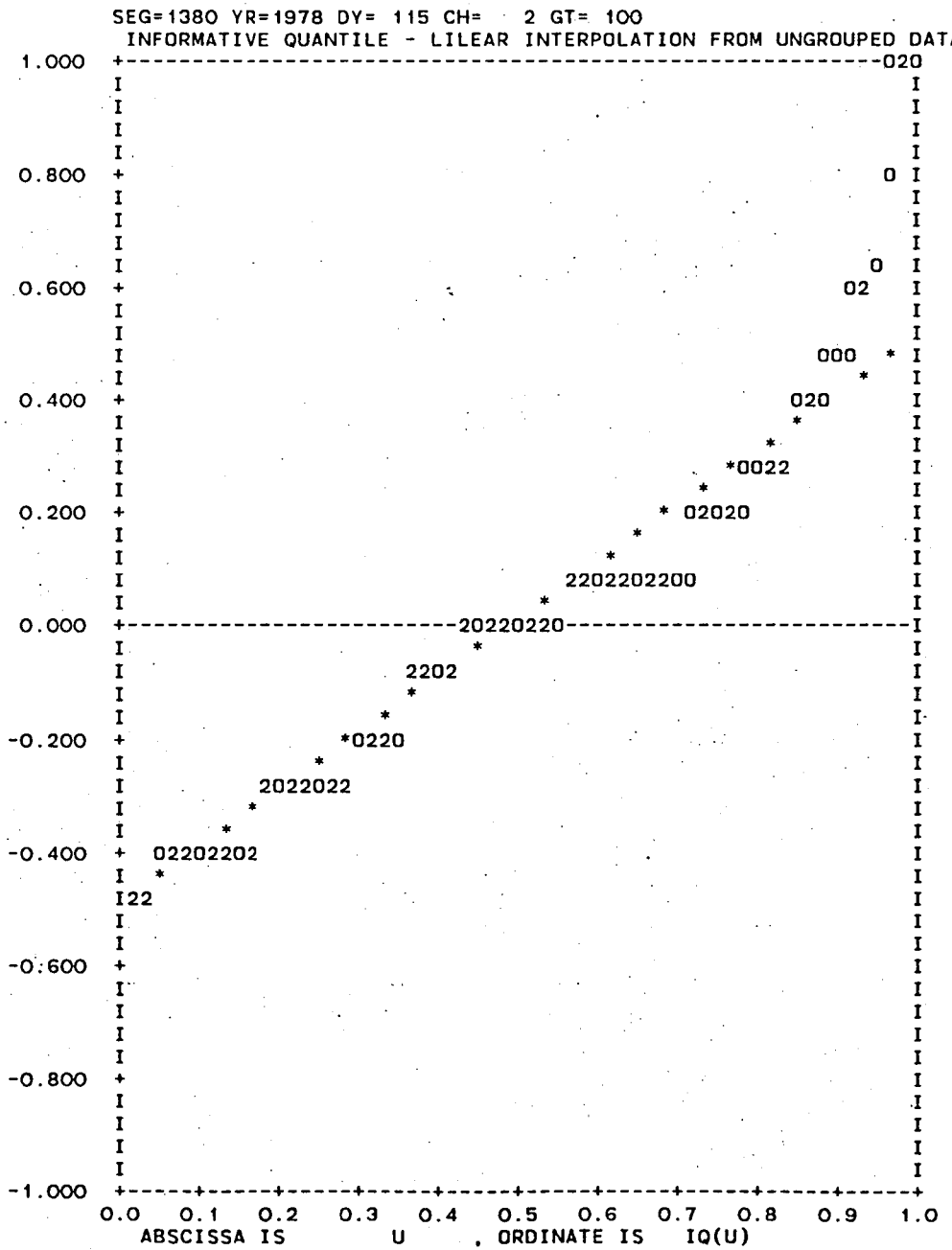
$$f_0Q_0(u) = (1-u)\ \{-\log\ (1-u)\}^{1-\beta}$$

Step (5) is to form, for each statistical characteristic, a data batch of the several thousand estimates of that characteristic corresponding to the pixels in the training file with the specified ground truth [here, spring wheat (code 100)]. Step (6) is to do a one-sample quantile data analysis of this data batch. These analyses are presented in detail for the following estimators: median channel 2, mean IQ channel 2, median channel 3, mean IQ channel 3.

The following table lists some basic summary measures for a one-sample statistical analysis of a data batch of statistical characteristics of Spring Wheat pixel neighborhoods:

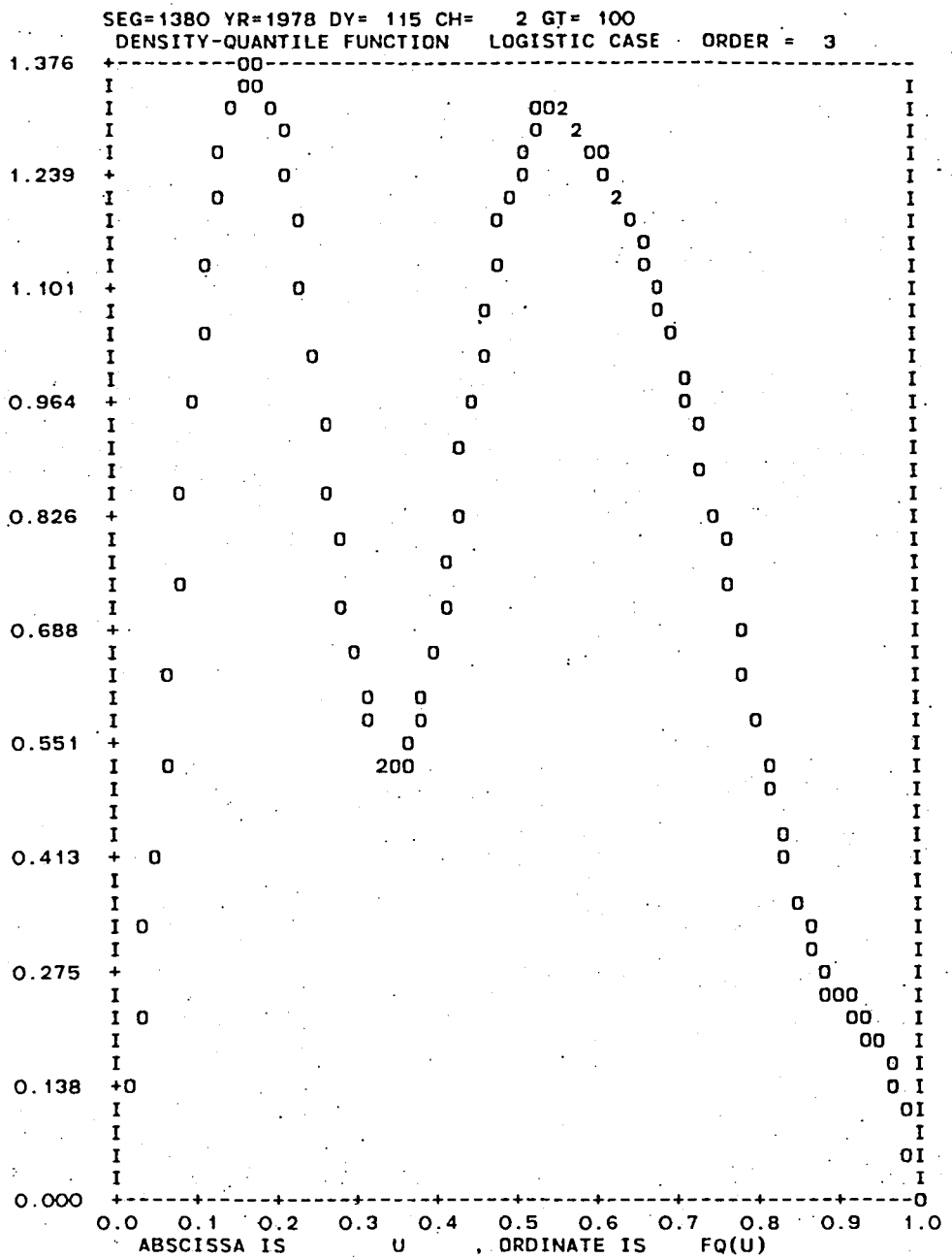
	Median Channel 2	Median Channel 3	Mean IQ Channel 2	Mean IQ Channel 3
Median	18	16	.02	.04
IQR	5	7.75	.14	.13
Mean IQ	.04	.09	.03	.01
Std IQ	.41	.36	.44	.49
Av. Log Spacings	-.68	-.59	.43	.43
σ_0 Weibull	.67	.61	.78	.80
σ_0 Logistic	.34	.20	.22	.22

We give for these variables: (1) printer plots of the informative quantile functions; (2) estimated density quantile functions, computed by the method of autoregressive density estimation; using the logistic and Weibull distributions as bases; and (3) diagnostic distribution functions (to be compared with the uniform) that help us decide which autoregressive order to accept as providing a "parsimonious" estimator.



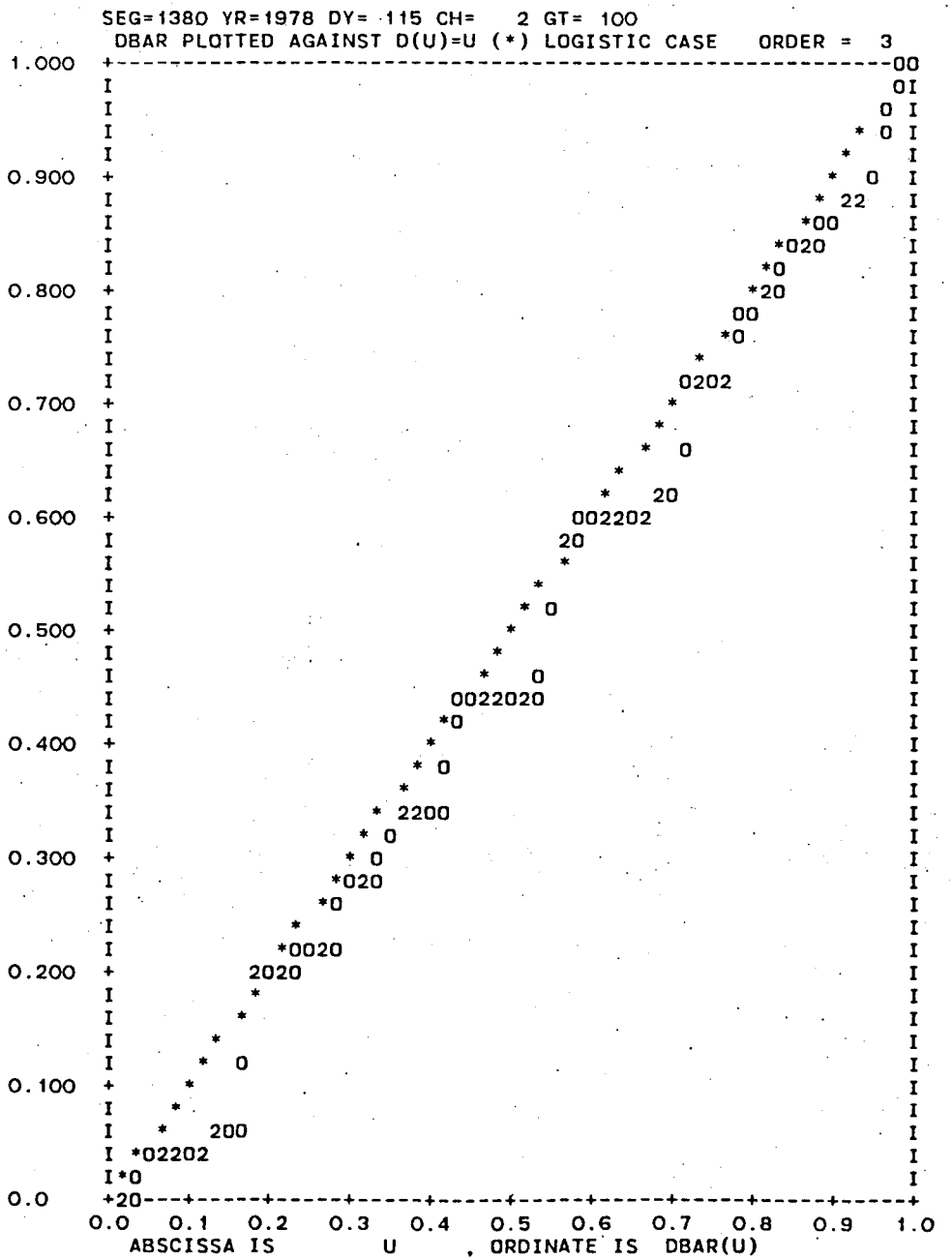
Spring Wheat Pixel Neighborhood Channel 2

Medians: IQ Plot indicates not normal but possibly Weibull. To test Weibull, we do not currently estimate shape parameter β, but choose $\beta = 0.7$.

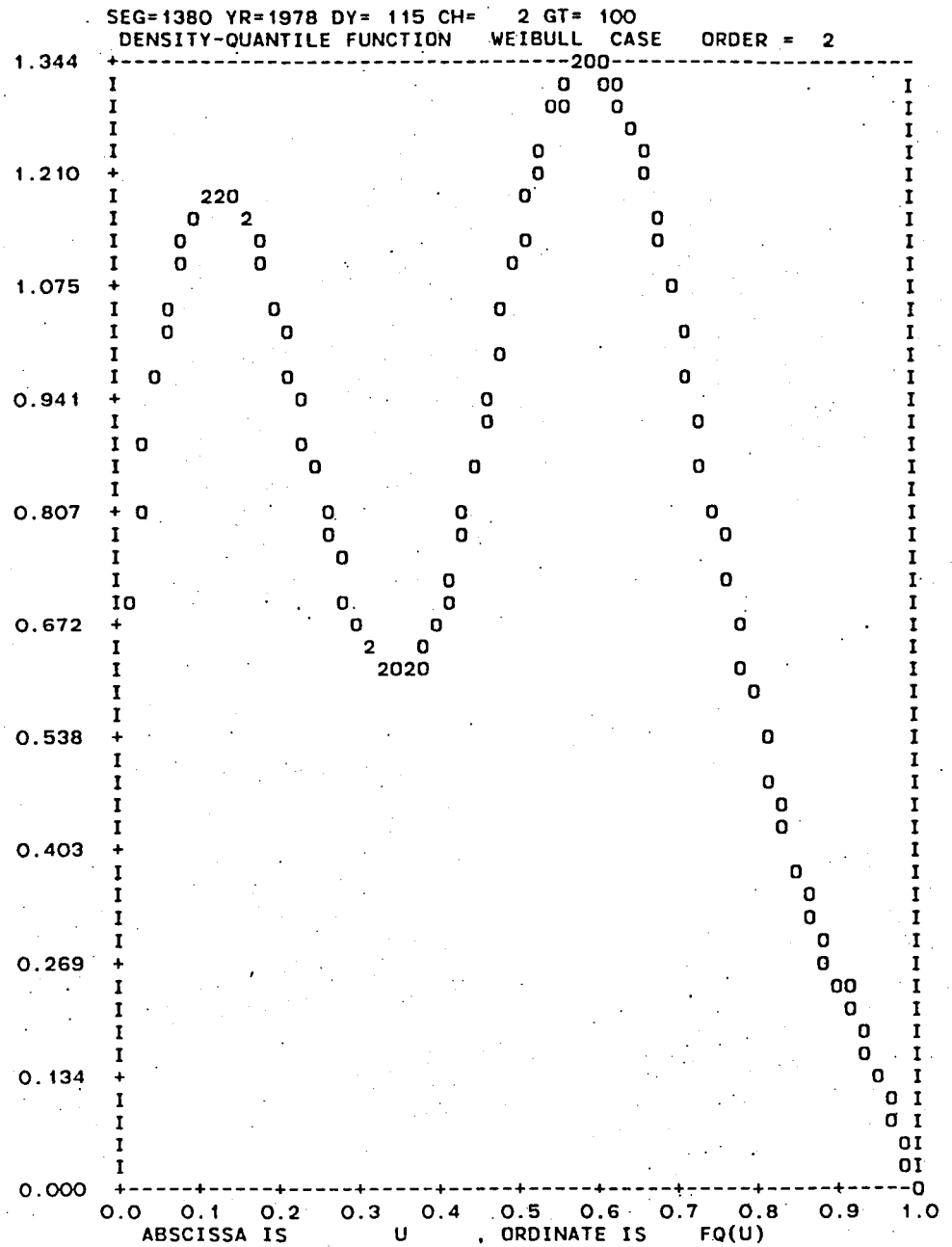


Spring Wheat Pixel Neighborhood Channel 2

Medians: Autoregressive density quantile estimator (with logistic base and order 3) indicates bimodal density.

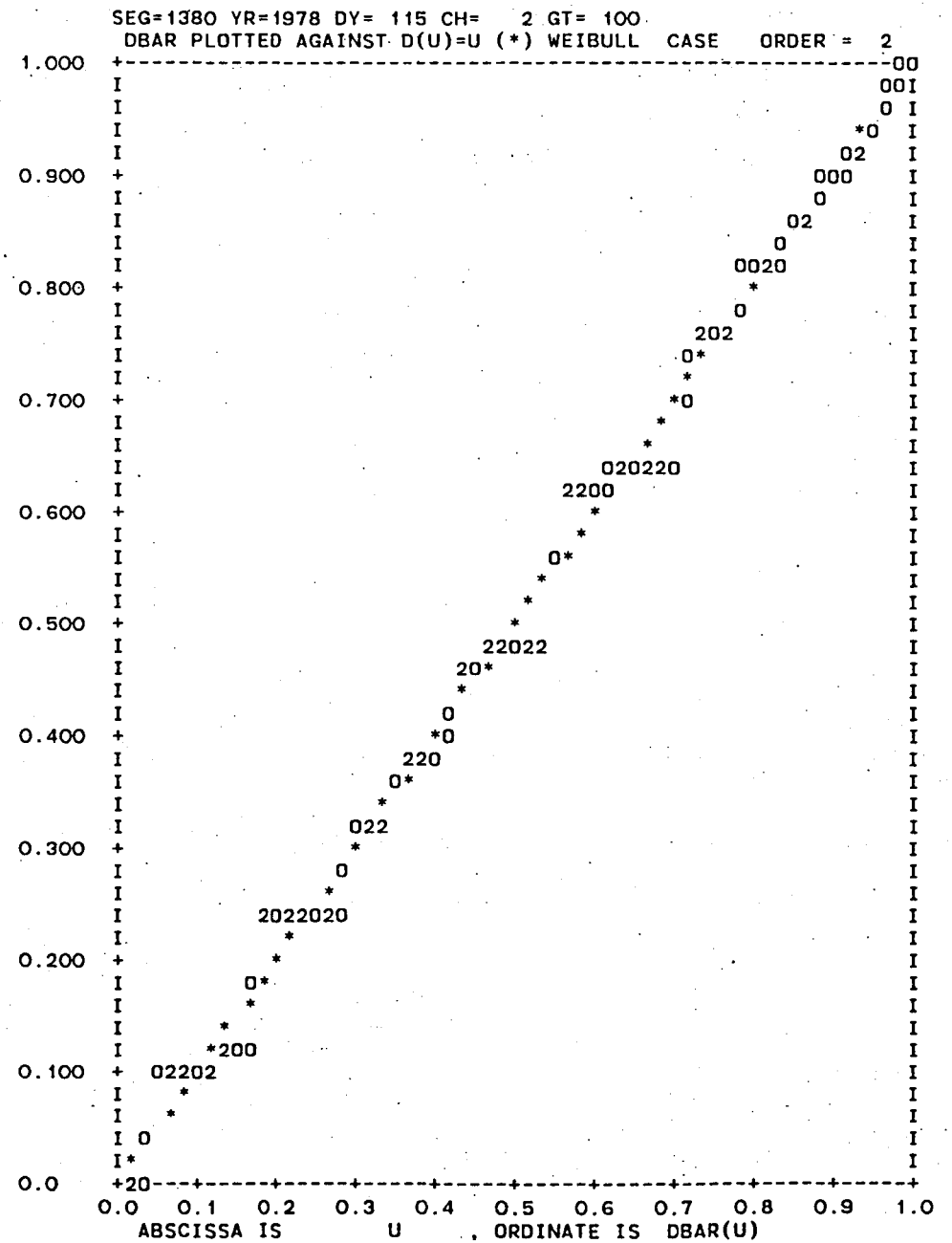


Spring Wheat Pixel Neighborhood Channel 2 Medians. Diagnostic of fit of AR density-quantile estimator (with logistic base and order 3).



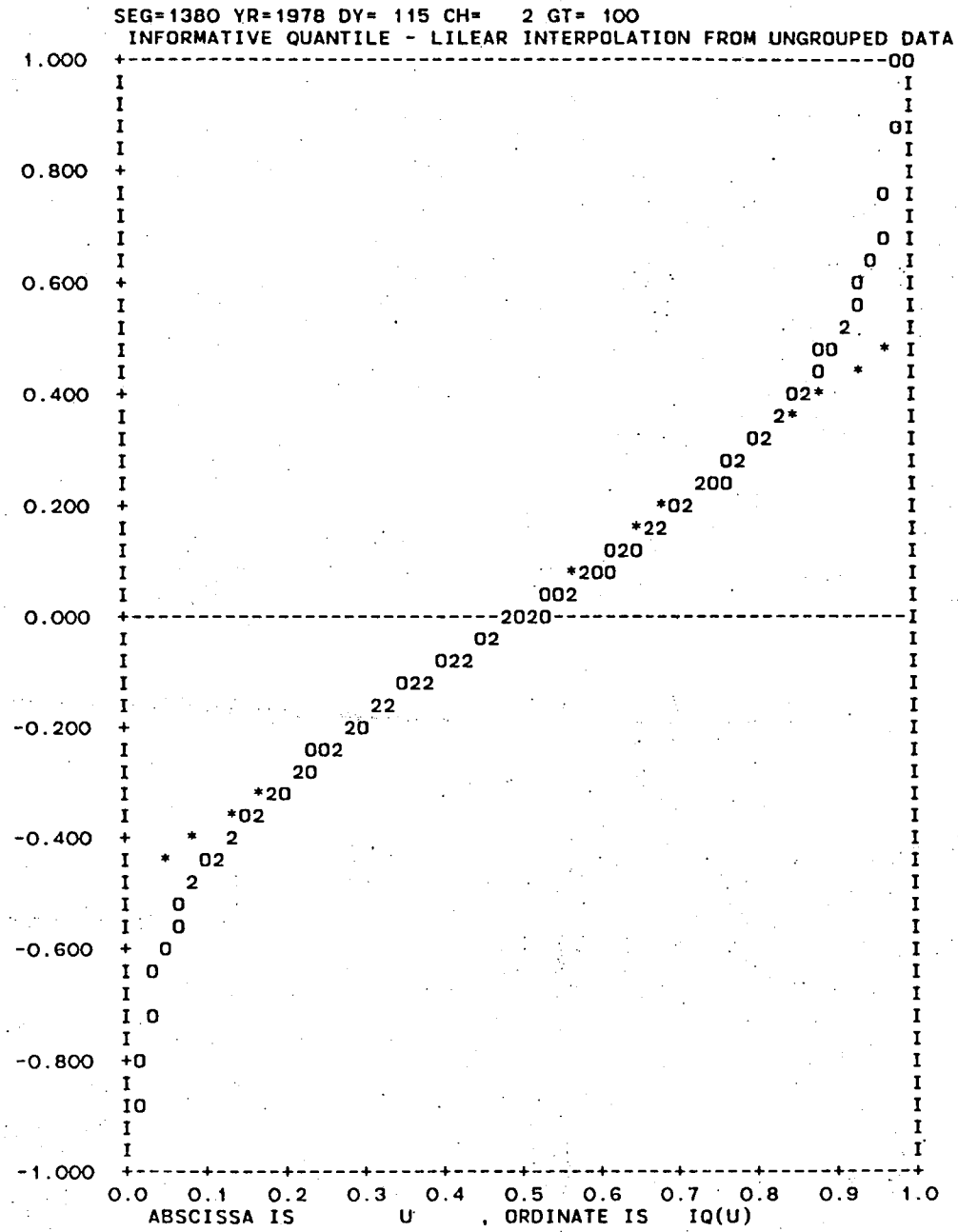
Spring Wheat Pixel Neighborhood Channel 2

Medians: Autoregressive density quantile analysis (with Weibull shape parameter .7 base and order 2) indicates bimodal density.



Spring Wheat Pixel Neighborhoods Channel 2

Medians: Diagnostic of fit of AR density quantile estimator (with Weibull shape parameter .7 base and order 2).



Spring Wheat Pixel Neighborhood Channel 2

Mean IQ: IQ plot indicates almost perfect normality.

AUTOREGRESSIVE PARAMETRIC SELECT ANALYSIS

Mean IQ Logistic Base

SQUARED MODULUS OF FOURIER COEFFICIENTS

```
PHI2( 1) =  0.00099923                                                  *
PHI2( 2) =  0.00039880     *
PHI2( 3) =  0.00049510           *
PHI2( 4) =  0.00095499                                          *
PHI2( 5) =  0.00067322                        *
```

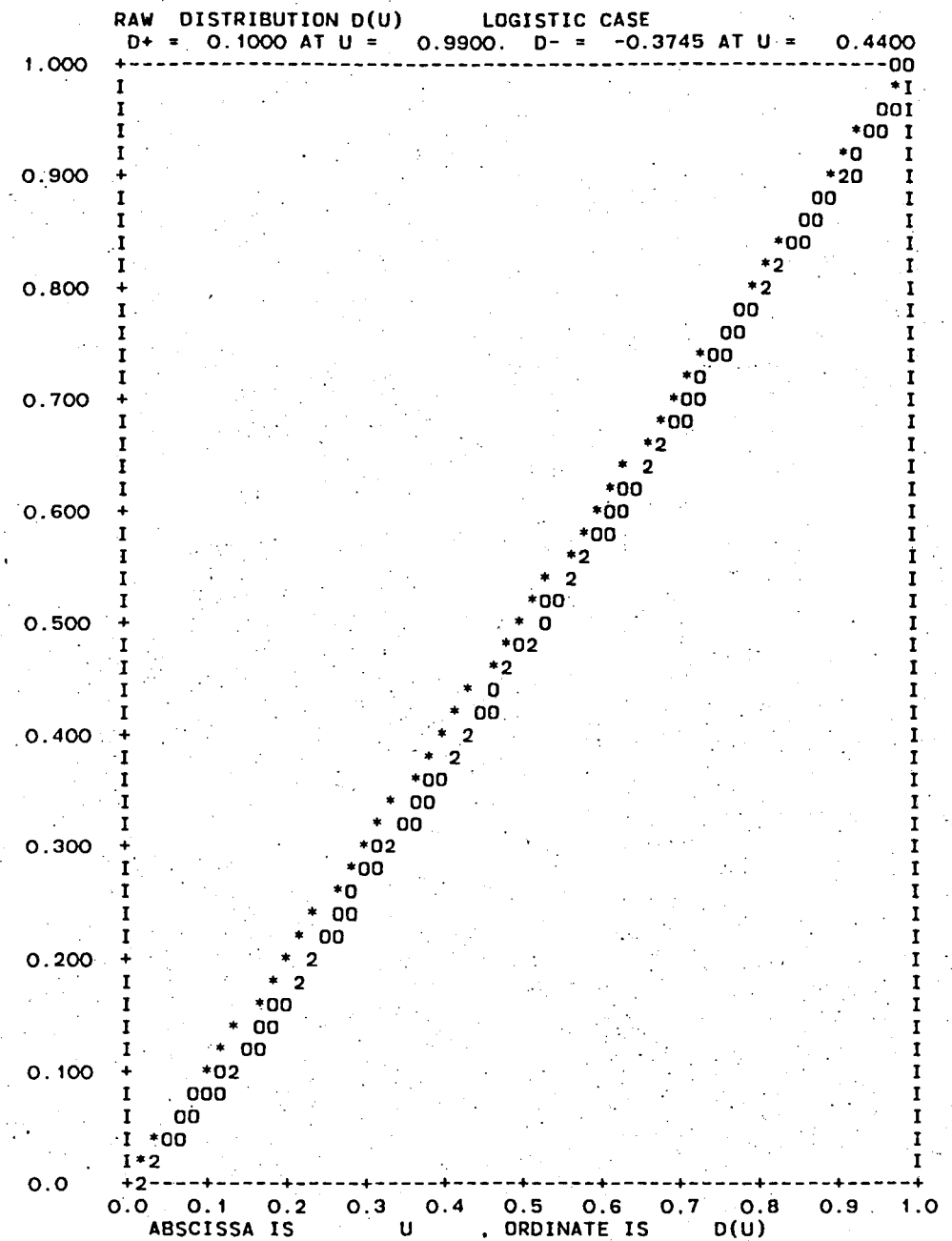
AUTOREGRESSIVE PARAMETRIC SELECT ANALYSIS

Mean IQ Weibull Case

SQUARED MODULUS OF FOURIER COEFFICIENTS

```
PHI2( 1) =  0.07646000                                                  *
PHI2( 2) =  0.03759194                        *
PHI2( 3) =  0.02402839              *
PHI2( 4) =  0.02089854           *
PHI2( 5) =  0.01483078     *
```

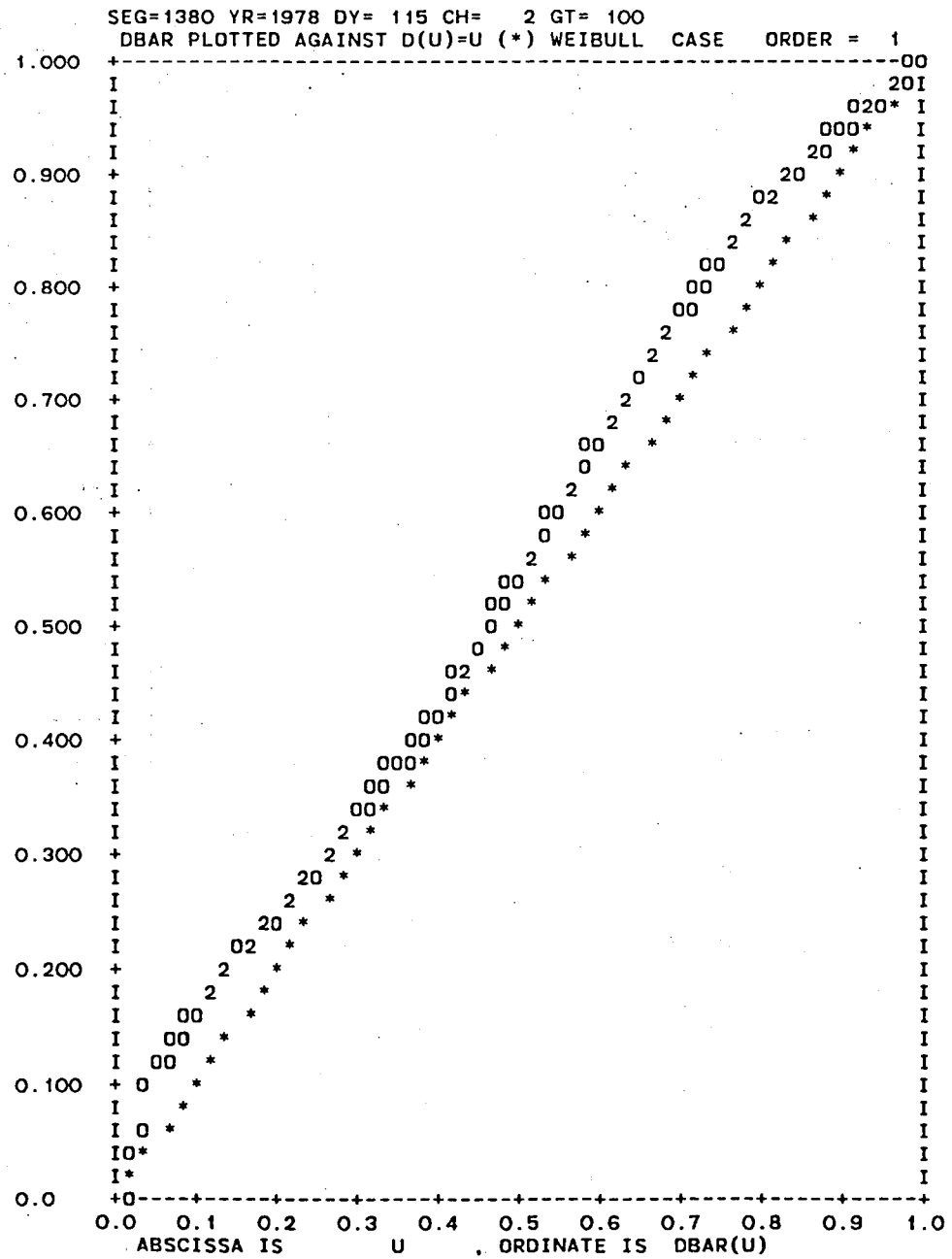
Spring Wheat Pixel Neighborhood Channel 2 Mean IQ: Pseudo-correlations square nodulus (phi 2) accept logistic distribution, reject Weibull distribution fit.



Spring Wheat Pixel Neighborhood Channel 2

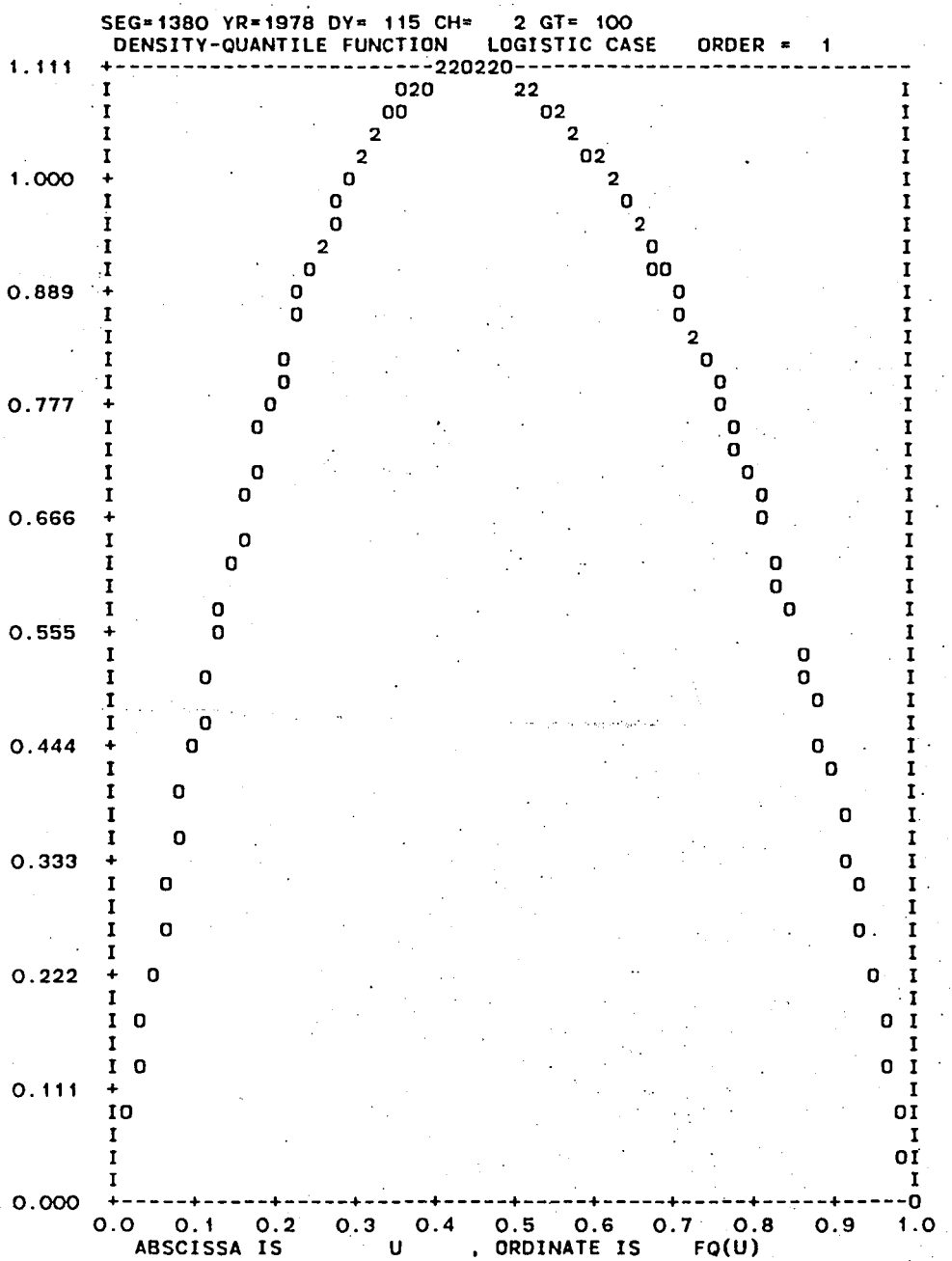
Mean IQ: Cumulative weighted spacings

$\tilde{D}(u)$ plot indicates accept fit of logistic distribution.



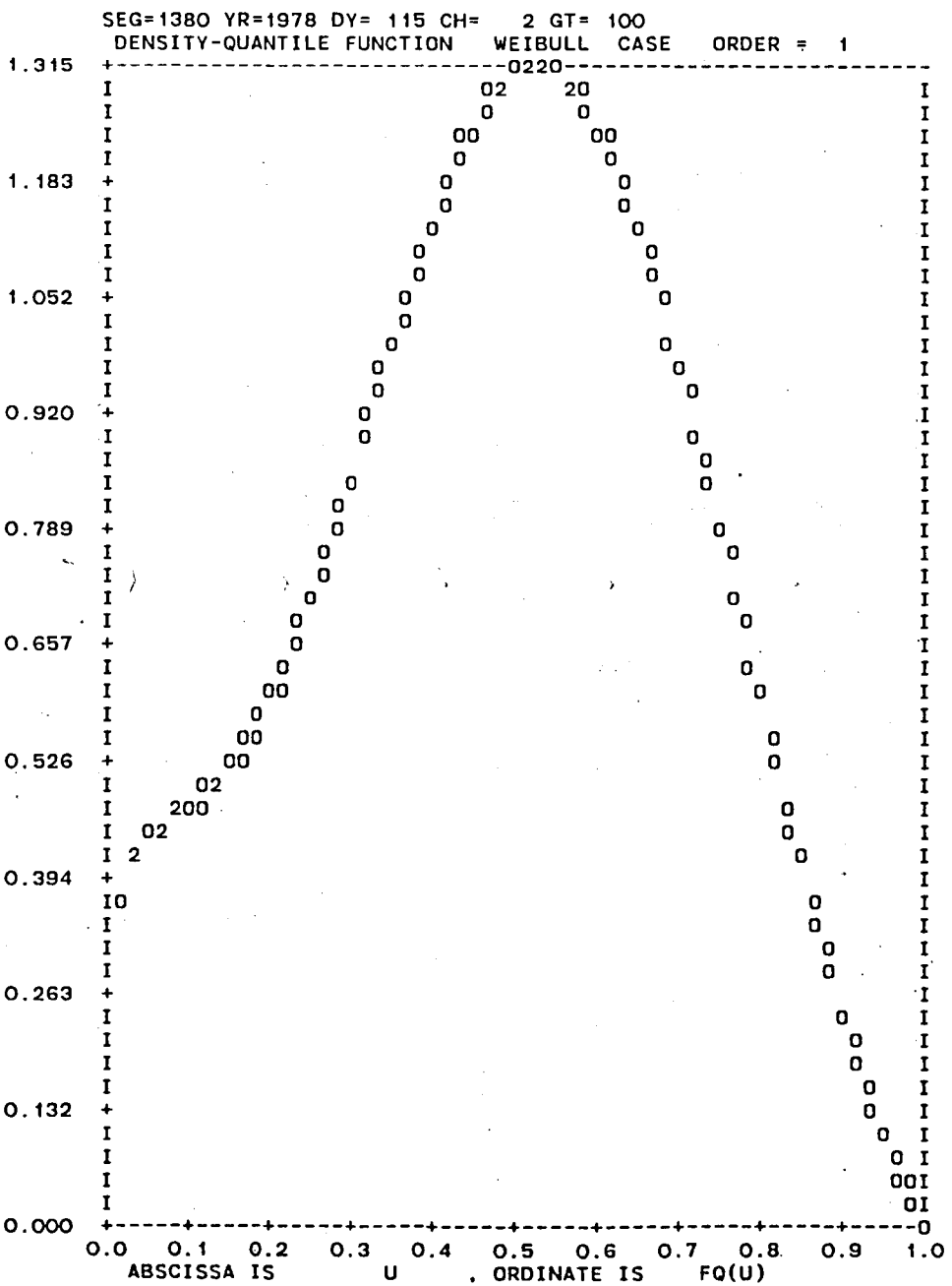
Spring Wheat Pixel Neighborhood Channel 2

Mean IQ: Cumulative weighted spacings D(u) plot indicates reject Weibull distribution fit.



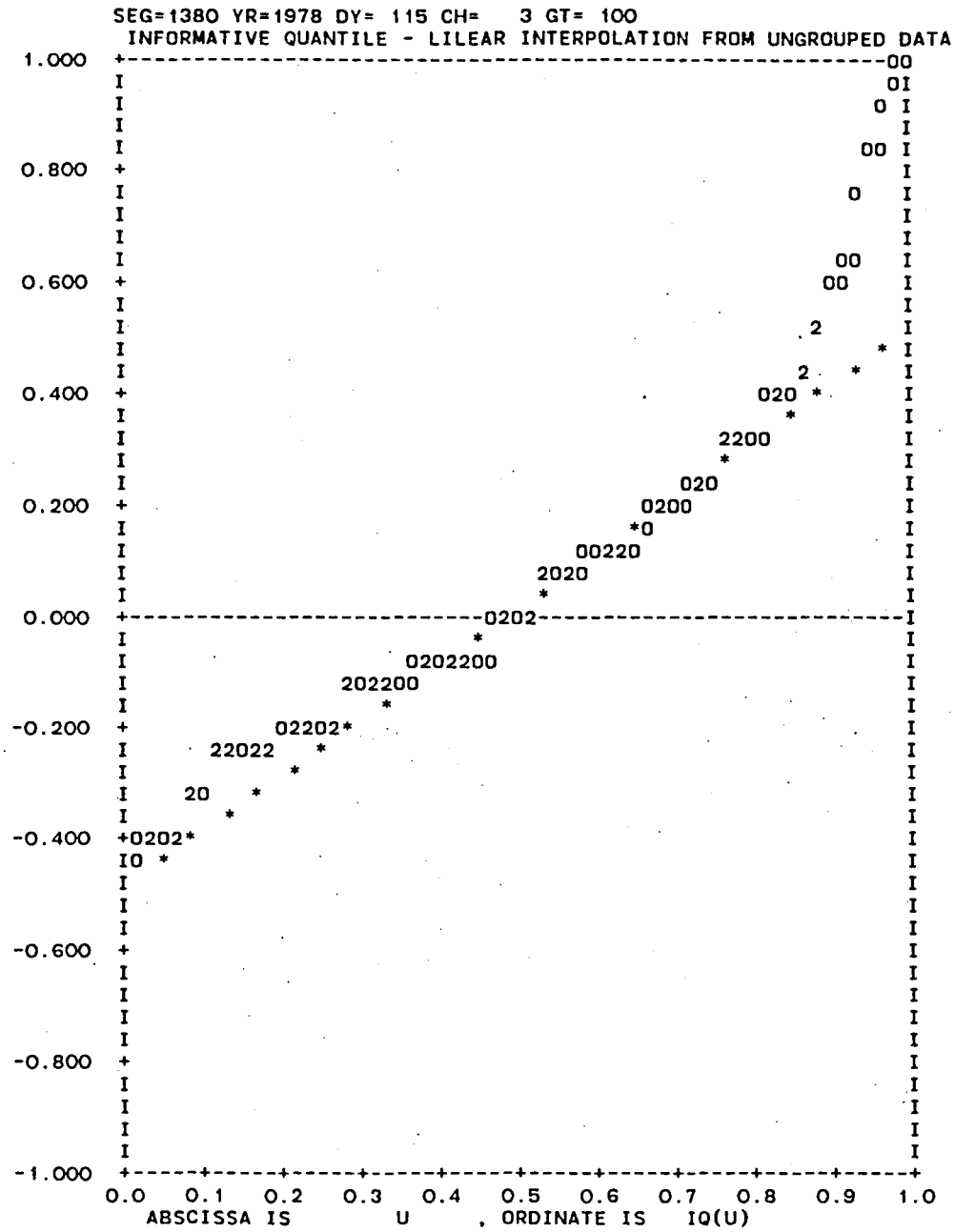
Spring Wheat Pixel Neighborhood Channel 2

Mean IQ: Autoregressive density-quantile estimator (with logistic base and order 1) indicates normal-like density.



Spring Wheat Pixel Neighborhood Channel 2

Mean IQ: Autoregressive density-quantile estimator (with Weibull base and order 1) indicates density which is symmetric and unimodal but less normal-like.



Spring Wheat Pixel Neighborhood Channel 3

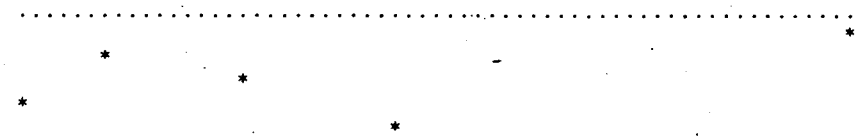
Medians: IQ plot indicates not normal but possibly Weibull.

AUTOREGRESSIVE PARAMETRIC SELECT ANALYSIS

Median Channel 3 Logistic

SQUARED MODULUS OF FOURIER COEFFICIENTS

PHI2(1) = 0.03838847
PHI2(2) = 0.00517420
PHI2(3) = 0.01129055
PHI2(4) = 0.00169468
PHI2(5) = 0.01800444

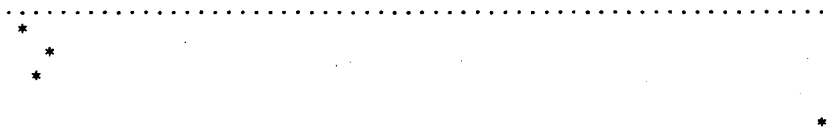


AUTOREGRESSIVE PARAMETRIC SELECT ANALYSIS

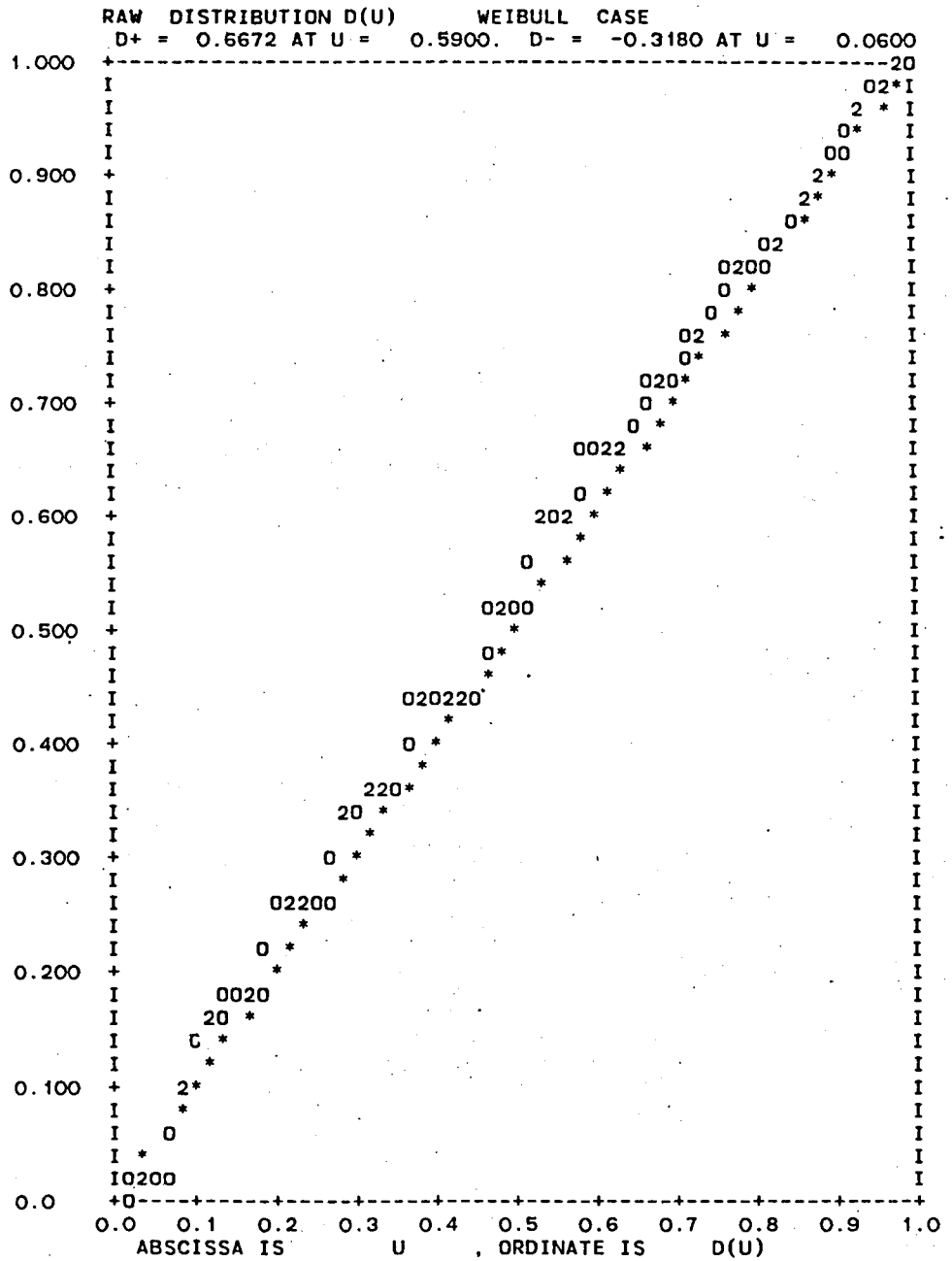
Median Channel 3 Weibull

SQUARED MODULUS OF FOURIER COEFFICIENTS

PHI2(1) = 0.00177358
PHI2(2) = 0.00257716
PHI2(3) = 0.00229294
PHI2(4) = 0.00134046
PHI2(5) = 0.01826305

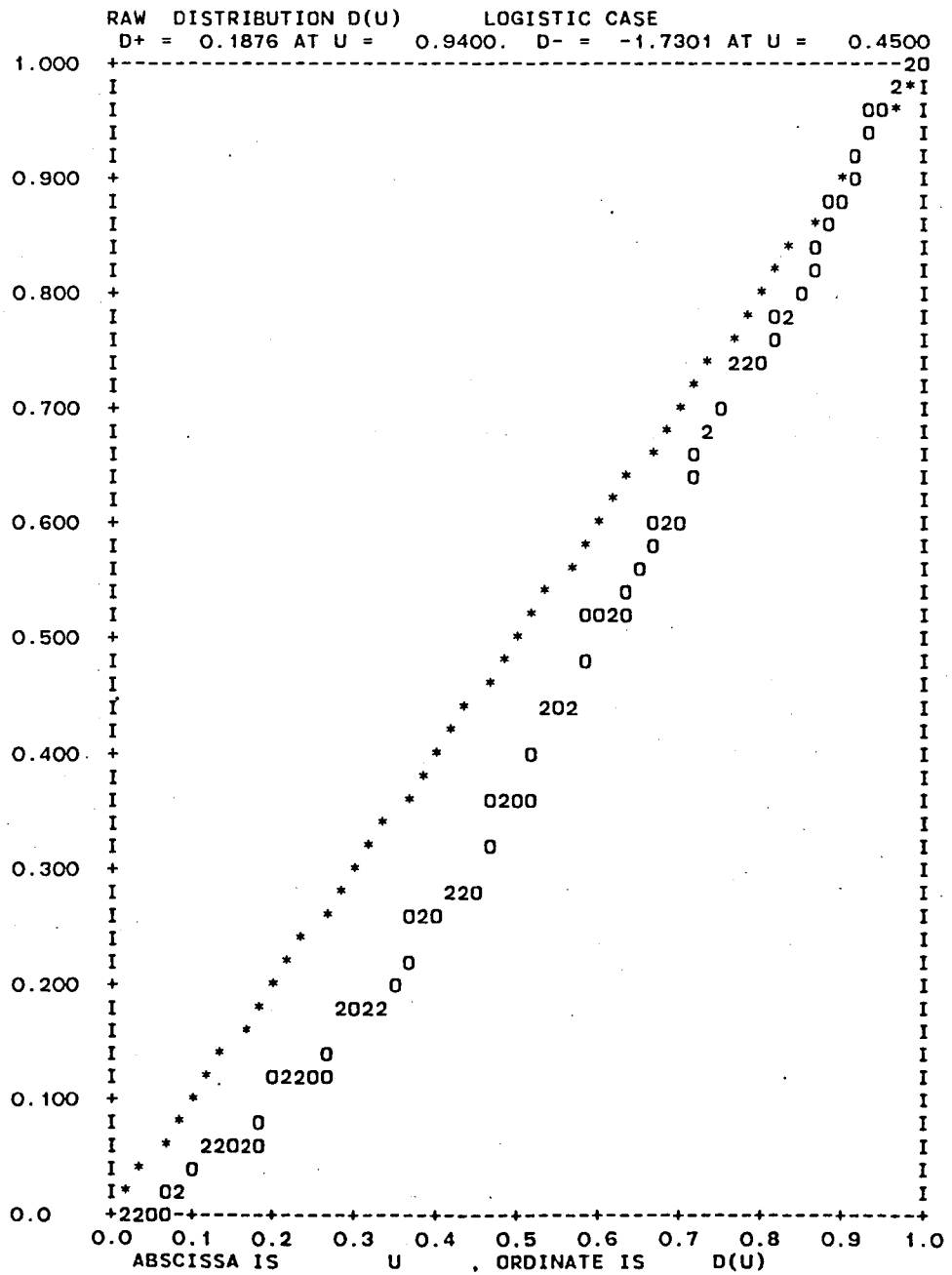


Spring Wheat Pixel Neighborhood Channel 3 Medians: Pseudo-correlations square modulus (phi 2) accept Weibull distribution, reject logistic distribution fit.



Spring Wheat Pixel Neighborhood Channel 3

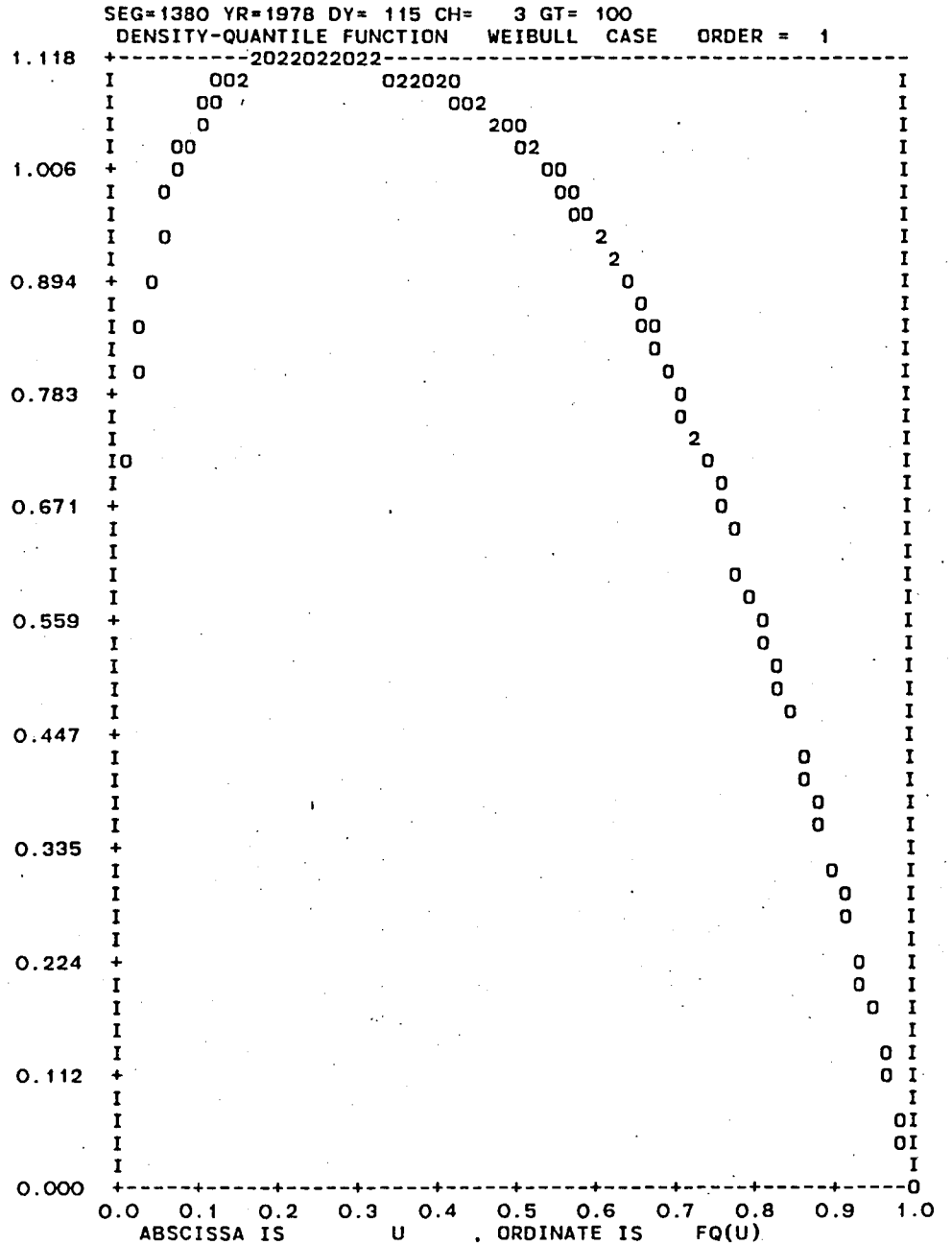
Medians: Cumulative weighted spacings $\tilde{D}(u)$ plot indicates accept fit of Weibull distribution (shape parameter β =0.7).



Spring Wheat Pixel Neighborhood Channel 3

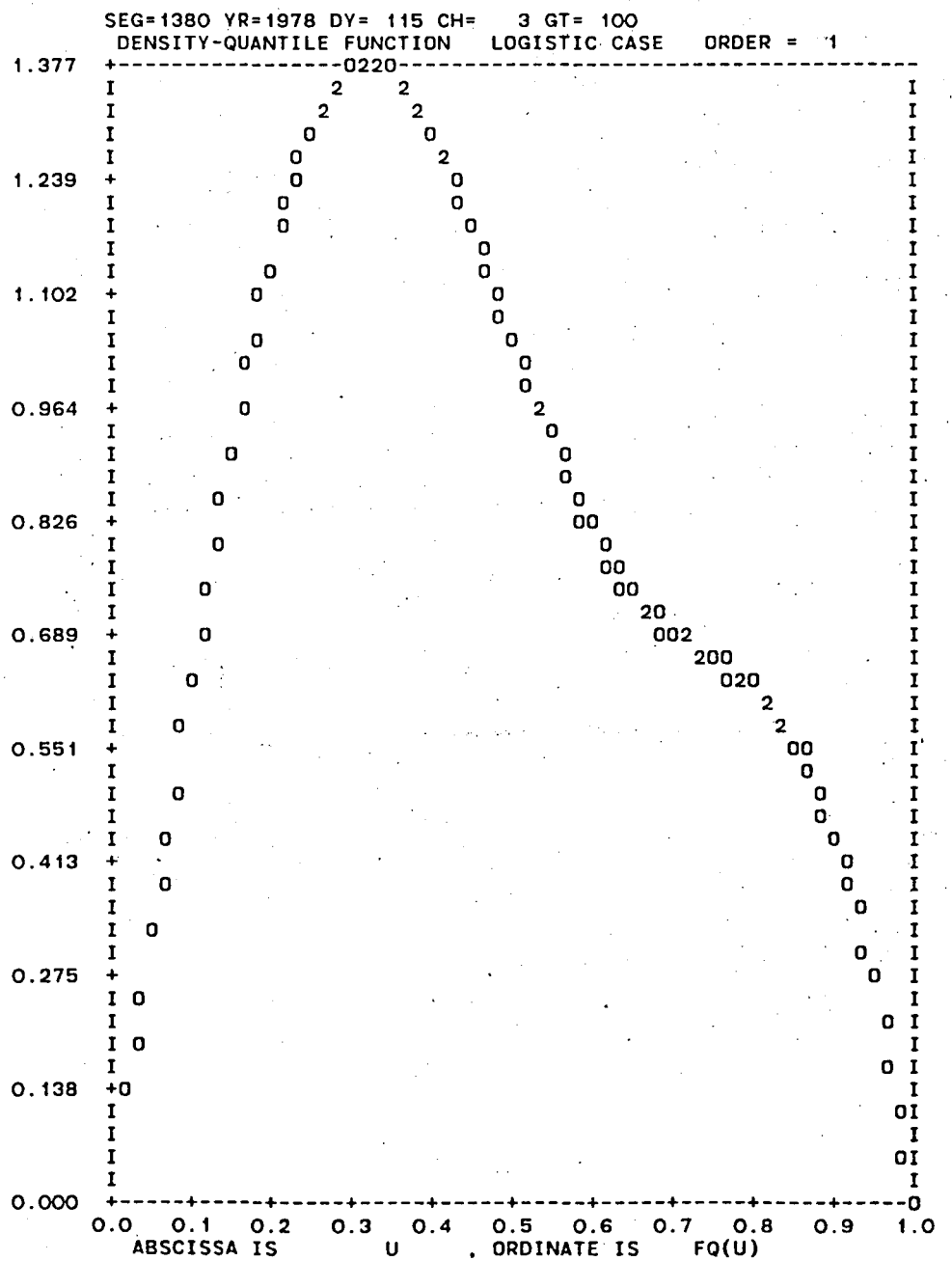
Medians: Cumulative weighted spacings

$\tilde{D}(u)$ plot indicates reject fit of logistic distribution.



Spring Wheat Pixel Neighborhood Channel 3

Medians: Autoregressive density-quantile estimator (with Weibull base and order 1)



Spring Wheat Pixel Neighborhood Channel 3

Medians: Autoregressive density-quantile estimator (with logistic base and order 1).

SUMMARY OF AR PARAMETRIC SELECT ANALYSIS FOR LOGISTIC CASE

Median Channel 3

ORDER	RES_VAR	LOG(RES_VAR)	AIC	CAT
1	0.96161	-0.03914	-0.01914	-1.01922
2	0.94997	-0.05133	-0.01133	-1.01100
3	0.93683	-0.06525	-0.00525	-1.00444
4	0.92880	-0.07387	0.00613	-0.99229
5	0.91700	-0.08665	0.01335	-0.98433

OPTIMAL ORDER BY CAT CRITERION IS 1 MAXIMUM ORDER CHECKED IS 5

OPTIMAL ORDER BY AIC CRITERION IS 1 MAXIMUM ORDER CHECKED IS 5

SUMMARY OF AR PARAMETRIC SELECT ANALYSIS FOR WEIBULL CASE

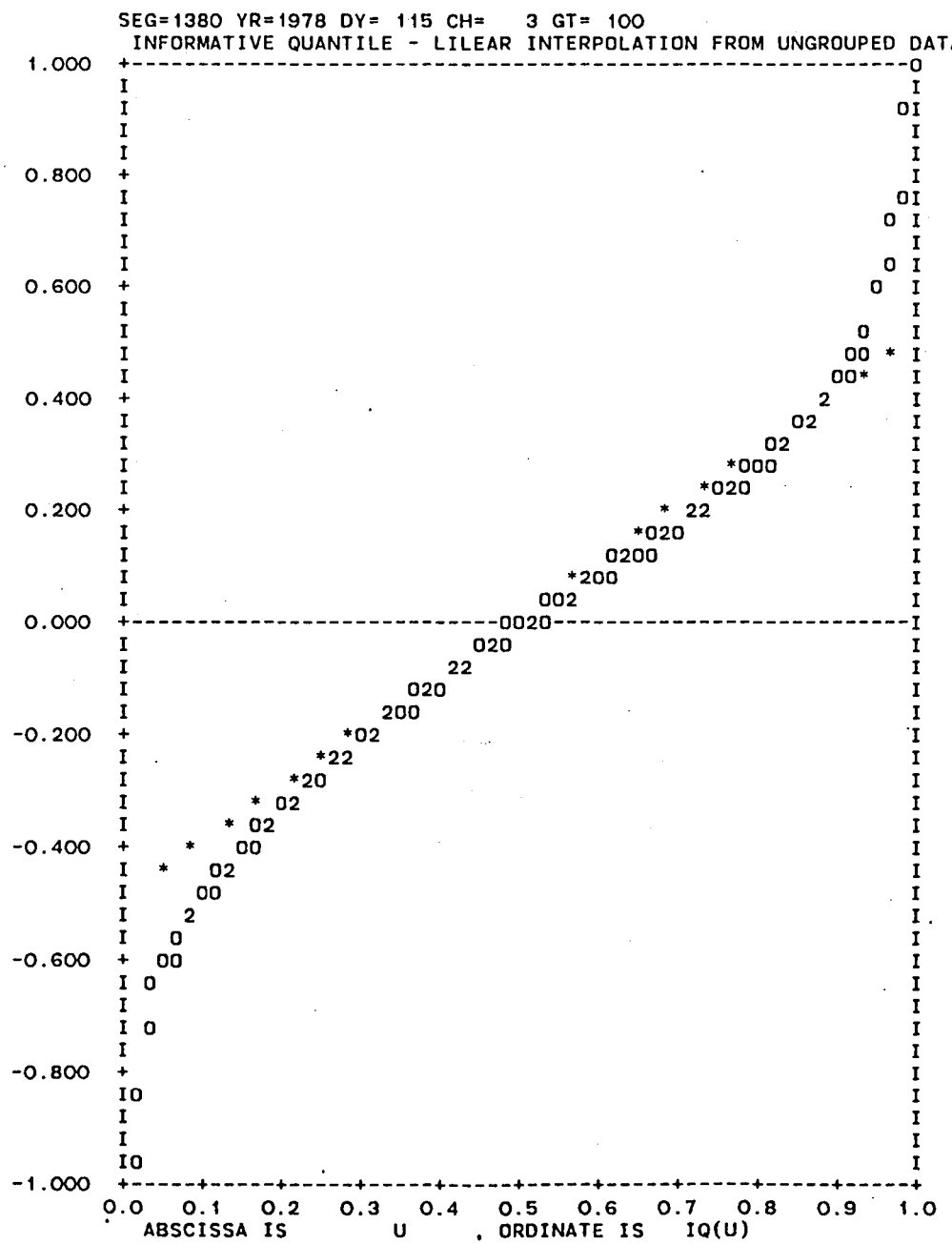
Median Channel 3

ORDER	RES_VAR	LOG(RES_VAR)	AIC	CAT
1	0.99823	-0.00177	0.01823	-0.98184
2	0.99556	-0.00445	0.03555	-0.96461
3	0.99366	-0.00636	0.05364	-0.94667
4	0.99218	-0.00785	0.07215	-0.92837
5	0.97480	-0.02552	0.07448	-0.92561

OPTIMAL ORDER BY CAT CRITERION IS 0 MAXIMUM ORDER CHECKED IS 5

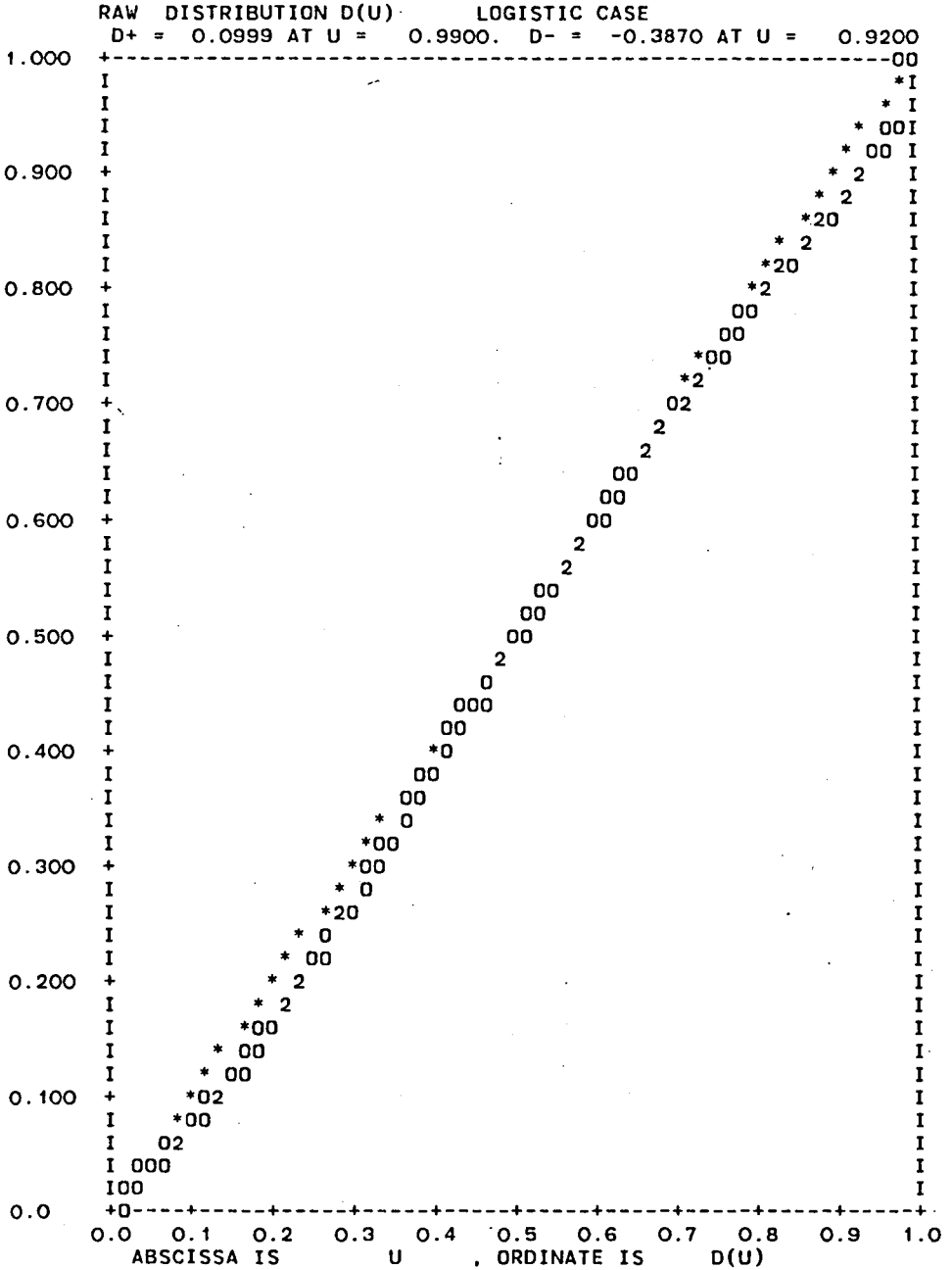
OPTIMAL ORDER BY AIC CRITERION IS 0 MAXIMUM ORDER CHECKED IS 5

Spring Wheat Pixel Neighborhood Channel 3 Medians: AIC AR order determining analysis.



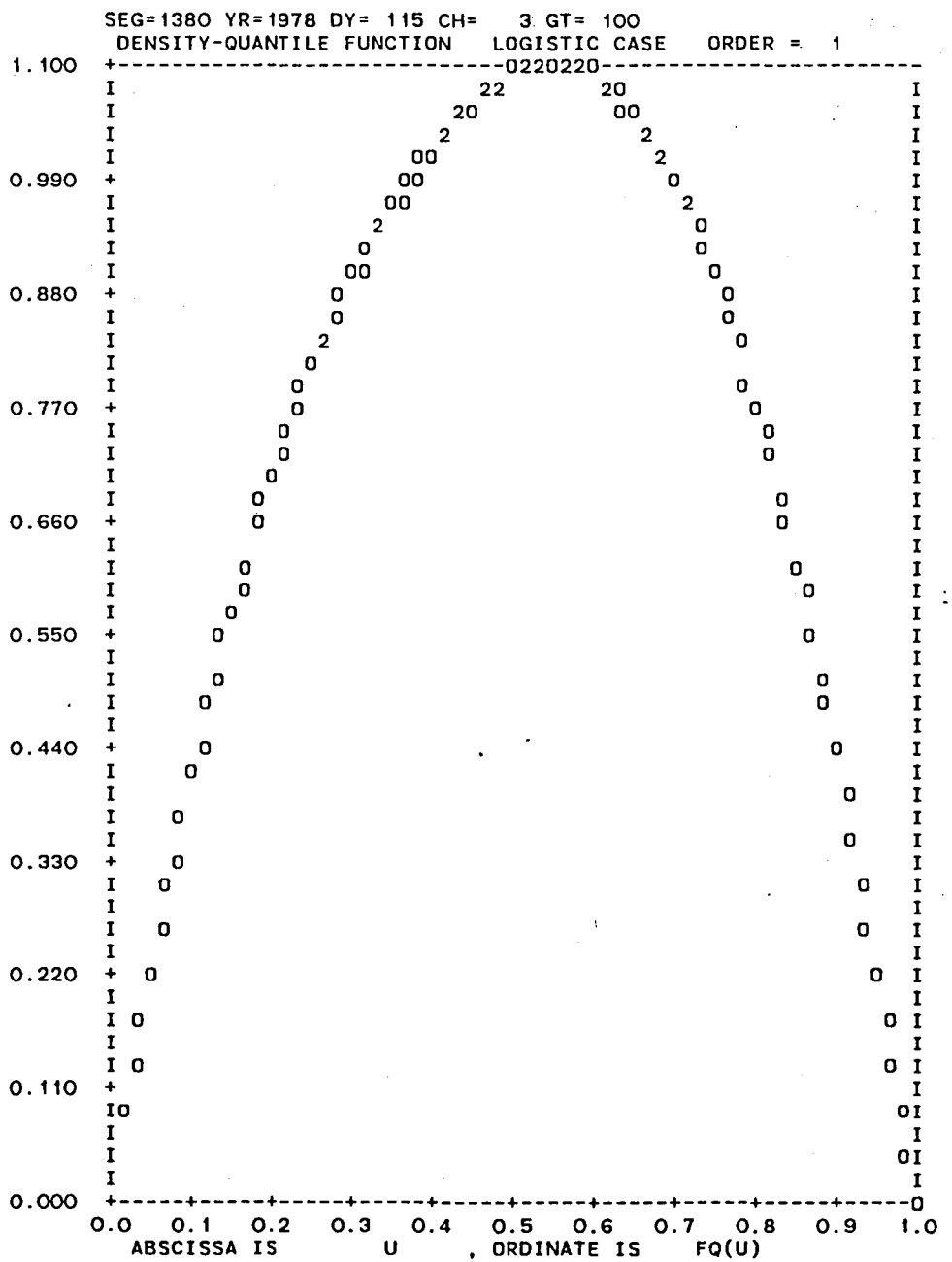
Spring Wheat Pixel Neighborhood Channel 3

Mean IQ: IQ plot indicates normality.



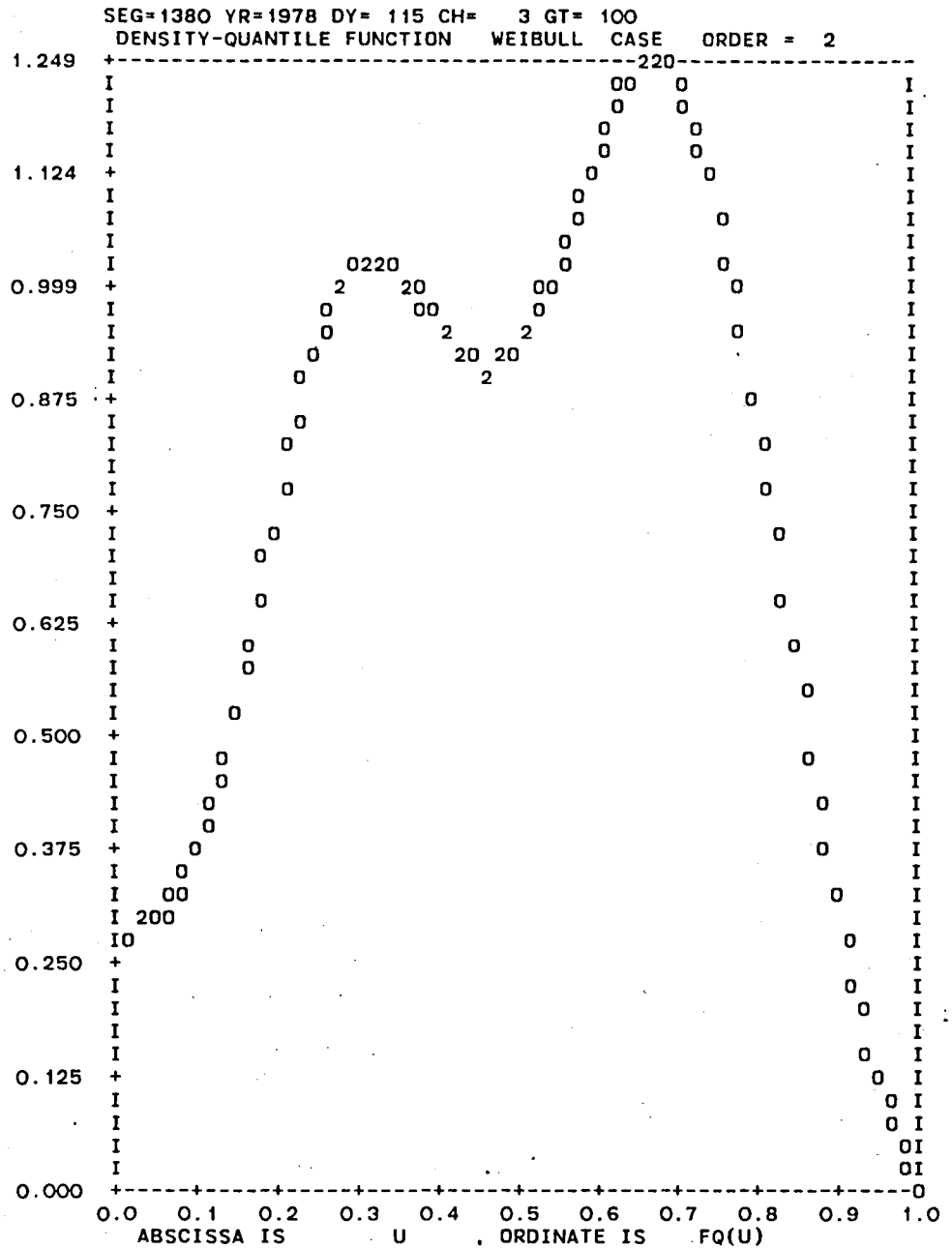
Spring Wheat Pixel Neighborhood Channel 3

Mean IQ: Cumulative weighted spacings D(u) plot indicates accept fit of logistic distribution.



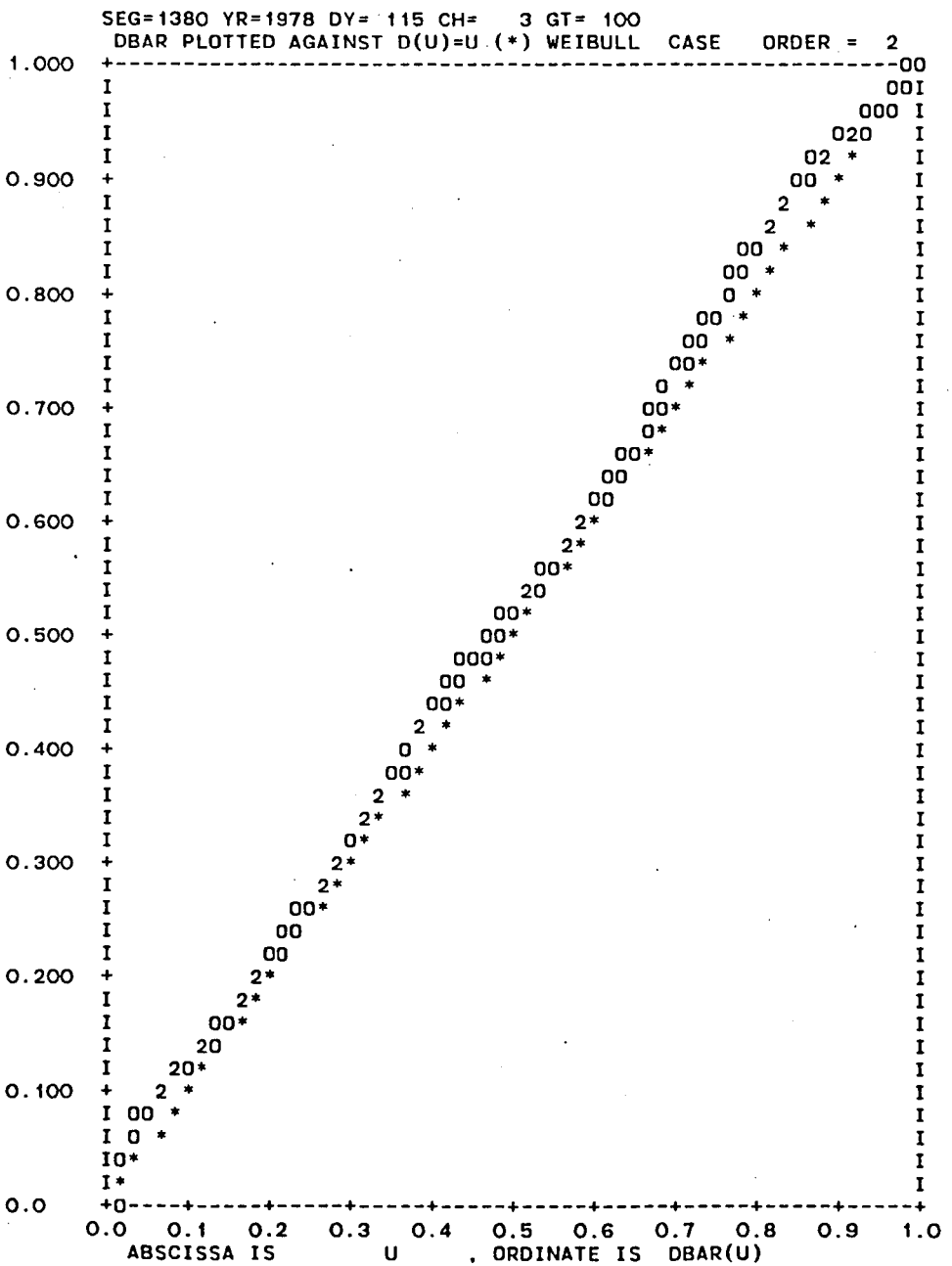
Spring Wheat Pixel Neighborhood Channel 3

Mean IQ: Autoregressive density-quantile estimator (with logistic base and order 1) indicates normal-like density.



Spring Wheat Pixel Neighborhood Channel 3

Mean IQ: Autoregressive density-quantile estimator (with Weibull base and order 2) indicates a density not in accord with logistic analysis, thus casting doubt on current reliability of AR order determining techniques.



Spring Wheat Pixel Neighborhood Channel 3

Mean IQ: Diagnostic of fit of AR density-quantile estimator (with Weibull base and order 2) indicates that it "overfits" and might generate spurious modes in the density.

SUMMARY OF AR PARAMETRIC SELECT ANALYSIS FOR WEIBULL CASE

Mean IQ Channel 3

ORDER	RES_VAR	LOG(RES_VAR)	AIC	CAT
1	0.90931	-0.09507	-0.07507	-1.07785
2	0.88209	-0.12546	-0.08546	-1.08899
3	0.86845	-0.14104	-0.08104	-1.08376
4	0.86084	-0.14984	-0.06984	-1.07086
5	0.85631	-0.15512	-0.05512	-1.05400

OPTIMAL ORDER BY CAT CRITERION IS 2 MAXIMUM ORDER CHECKED IS 5

OPTIMAL ORDER BY AIC CRITERION IS 2 MAXIMUM ORDER CHECKED IS 5

SUMMARY OF AR PARAMETRIC SELECT ANALYSIS FOR LOGISTIC CASE

Mean IQ Channel 3

ORDER	RES_VAR	LOG(RES_VAR)	AIC	CAT
1	0.99882	-0.00118	0.01882	-0.98126
2	0.99655	-0.00346	0.03654	-0.96365
3	0.99392	-0.00610	0.05390	-0.94643
4	0.99086	-0.00918	0.07082	-0.92966
5	0.98756	-0.01251	0.08749	-0.91315

OPTIMAL ORDER BY CAT CRITERION IS 0 MAXIMUM ORDER CHECKED IS 5

OPTIMAL ORDER BY AIC CRITERION IS 0 MAXIMUM ORDER CHECKED IS 5

Spring Wheat Pixel Neighborhood Channel 3 Mean IQ: AIC AR order determining analysis.

Appendix: Quantile and FUN.STAT Data Analysis

This appendix presents some of the new characterizations of probability laws which are being developed under the names of quantile data analysis, and functional statistical inference analysis. Estimators of these characteristics are currently available for one sample and two samples, univariate and bivariate [Parzen (1979), (1983), Woodfield (1982)].

These methods seem to have much potential to contribute to the solution of the problem of digital image representation: the determination and modeling of basic characteristics or features of the digital image which can be incorporated into the process of identifying classes and attributes in a scene. They provide new approaches to determining scene probability density functions and class conditional density functions of digital image data in order to understand spectral characteristics and extract desired information. They can provide data representations which reduce the dimensions of multivariate image data while preserving information pertaining to scene classes and attributes.

A. One Sample: Univariate

Let X be continuous random variable of which we observe a random sample. To estimate distribution function $F_X(x) = Pr[X \leq x]$ and probability density $f(x) = F'(x)$, we estimate: quantile function $Q_X(u) = F_X^{-1}(u)$; quantile density $q_X(u) = Q_X'(u)$; density quantile $fQ_X(u) = f_X(Q_X(u))$. A quantile data analysis of the random sample

1. Forms sample distribution function $\tilde{F}_X(x)$, sample quantile function $\tilde{Q}_X(u)$, sample quantile density $\tilde{q}(u)$ at a grid of values of u in $0<u<1$.

2. Plots sample version of informative quantile function

$$IQ(u) = \frac{Q(u)-Q(0.5)}{2\{Q(0.75) - Q(0.25)\}}$$

whose values as u tends to 0 and 1 indicates the tail exponents of the probability law of X.

3. Determines standard distribution functions $F_0(x)$ to test

$$H_0: F(x) = F_0(\frac{X-\mu}{\sigma}) \text{ or } Q(u) = \mu + \sigma Q_0(u)$$

for location and scale parameters μ and σ to be estimated. A test of H_0 which does not require estimation of μ and σ can be based on [Parzen (1979)]

$$\tilde{d}(u) = f_0Q_0(u)\ \tilde{q}(u) \div \tilde{\sigma}_0 \quad ,$$

$$\tilde{\sigma}_0 = \int_0^1 f_0Q_0(t)\ \tilde{q}(t)\ dt$$

which estimate respectively

$$d(u) = f_0Q_0(u)\ q(u) \div \sigma_0 \quad ,$$

$$\sigma_0 = \int_0^1 f_0Q_0(t)\ q(t)\ dt.$$

4. Form successive autoregressive estimators

$$\hat{d}_m(u) = \hat{K}_m \left| 1 + \hat{\alpha}_m(1)\ e^{2\pi iu} + \ldots + \hat{\alpha}_m(m)\ e^{2\pi ium} \right|^{-2}$$

whose negentropy

$$\hat{H}_m = \int_0^1 - \log \hat{d}_m(u)\ du = - \log \hat{K}_m$$

is used to determine optimal orders $\hat{m}$. Note that $\hat{H}_m$ estimates the entropy difference

$$\Delta = \{\log \sigma_o - \int_0^1 \log f_o Q_o(u)\} - \{- \int_0^1 \log fQ(u)\, du\}$$

5. Estimate $fQ(u)$ by

$$\hat{fQ}_m(u) = f_o Q_o(u) \div \tilde{\sigma}_o \hat{d}_m(u)$$

where m is chosen equal to an optimal order $\hat{m}$.

B. Two Sample: Univariate

Let X and Y be continuous random variables with random samples $X_1,\ldots,X_m$ and $Y_1,\ldots,Y_n$ respectively, and with respective distribution functions $F(x) = Pr[X \leq x]$, $G(x) = Pr[Y \leq x]$. The pooled sample $X_1,\ldots,X_m$, $Y_1,\ldots,Y_n$ can be regarded as a random sample from the distribution function

$$H(x) = \lambda F(x) + (1-\lambda) G(x), \qquad \lambda = \frac{m}{m+n} \quad .$$

To test the hypotheses of equality of distributions, H_o: $F(x) = G(x) = H(x)$, it is customary in non-parametric statistics to introduce

$$D_X(u) = F H^{-1}(u), \qquad D_Y(u) = G H^{-1}(u)$$

with densities [equivalent to likelihood ratios]

$$d_X(u) = \frac{f H^{-1}(u)}{f H^{-1}(u)} \quad , \qquad d_Y(u) = \frac{g H^{-1}(u)}{h H^{-1}(u)} \quad .$$

Note that $h H^{-1}(u) = \lambda f H^{-1}(u) + (1- \lambda) g H^{-1}(u)$; therefore

$$d_X(u) = \left\{ \lambda + (1-\lambda) \frac{g H^{-1}(u)}{f H^{-1}(u)} \right\}^{-1} \quad .$$

Parzen (1983) shows that all conventional two-sample nonparametric test procedures are functionals of the following raw estimator of $D_X(u)$:

$$\tilde{D}_X(u) = \{\tilde{H}\ \tilde{F}_X^{-1}\}^{-1}(u)$$

from which one can form "pseudo-correlations" $\tilde{\rho}(v)$ and linear rank statistics $\Delta(J)$ with score function $J(u)$,

$$\tilde{\rho}(v) = \int_0^1 e^{2\pi i u v}\, d\,\tilde{D}_X(u)\ , \quad \Delta(J) = \int_0^1 J(u)\, d\tilde{D}_X(u)\ ,$$

and autoregressive estimators $\hat{d}_{X,m}(u)$ of $d_X(u)$.

When one observes several variables $X^{(1)}, X^{(2)},\ldots,X^{(j)},\cdots$ one estimates functionals of $D_j(u) = F_{X(j)}(H^{-1}(u))$ or $D_{jk}(u) = F_{X(j)}(F^{-1}_{X(k)}(u))$.

C. One Sample: Bivariate

Let (X_1, X_2) be jointly continuous random variables with distribution function $F_{X_1,X_2}(x_1,x_2) = \Pr[X_1 \le x\ ,\ X_2 \le x_2]$ and density $f_{X_1,X_2}(x_1,x_2)$. The joint density quantile function is defined by

$$fQ_{X_1,X_2}(u_1,u_2) = f_{X_1,X_2}(Q_{X_1}(u_1),\ Q_{X_2}(u_2))\ .$$

To estimate fQ we define

$$D_{X_1,X_2}(u_1,u_2) = F_{X_1,X_2}(Q_{X_1}(u_1),\ Q_{X_2}(u_2))$$

which is the distribution function of $U_1 = F_{X_1}(X_1)$, $U_2 = F_{X_2}(X_2)$; it has density

$$d_{X_1,X_2}(u_1,u_2) = \frac{\partial^2}{\partial u_1 \partial u_2} D(u_1,u_2)$$

satisfying

$$fQ_{X_1,X_2}(u_1,u_2) = fQ_{X_1}(u_1)\, fQ_{X_2}(u_2)\, d_{X_1,X_2}(u_1,u_2) \quad .$$

To estimate d_{X_1,X_2} from a random sample $(X_1^{(j)}, X_2^{(j)})$, $j=1,\ldots,n$, form

$$\tilde{D}_{X_1,X_2} = \tilde{F}_{X_1,X_2}(\tilde{Q}_{X_1}(u_1), \tilde{Q}_{X_2}(u_2))$$

and a raw estimator $\hat{d}_{X_1,X_2}(u_1,u_2)$. We smooth $\log \tilde{d}_{X_1,X_2}(u_1,u_2)$ by a smooth estimator $\log \hat{d}_{X_1,X_2}(u_1,u_2)$ minimizing a criterion <u>similar</u> to

$$\sum_{j=1}^{n} \left| \log d[U_1^{(j)},U_2^{(j)}] - \log d_m[U_1^{(j)}, U_2^{(j)}] \right|^2$$

where $\log d_m(u_1,u_2)$ has the parametric representation

$$\log d_m(u_1,u_2) = \sum_{\nu_1,\nu_2} \theta_{\nu_1,\nu_2} \exp i\,(u_1\nu_1 + u_2\nu_2) - \psi(\theta_{\nu_1,\nu_2}) \; ;$$

where the summation is over $\nu_1,\nu_2 = 0, \pm 1,\ldots,\pm m$, and $\psi(\theta_{\nu_1,\nu_2})$ is an integrating factor to make $d_m(u_1,u_2)$ a probability density. The foregoing estimators have been implemented in T. J. Woodfield [1982]. The problem of choosing a best value of the order m is approached by evaluating the entropy of d_m.

D. Two Samples: Bivariate

Let (X_1,X_2) and (Y_1,Y_2) be random vectors with respective distribution functions $F(x_1,x_2)$ and $G(y_1,y_2)$, and respective random samples

$$(X_1,{}^{(j)}X_2{}^{(j)}),\ j=1,\ldots,m \quad \text{and} \quad (Y_1{}^{(k)},Y_2{}^{(k)}),\ k=1,2,\ldots,n.$$

Let $H(x_1,x_2)$ denote the distribution function of the pooled random sample, with marginal distribution functions $H_1(x_1)$ and $H_2(x_2)$. Define

$$D_1(u_1,u_2) = F(H_1^{-1}(u_1),\ H_2^{-1}(u_2)) \quad ,$$

$$D_2(u_1,u_2) = F(H_1^{-1}(u_1),\ H_2^{-1}(u_2)) \quad .$$

From $D_1(u_1,u_2)$ and $D_2(u_1,u_2)$ one can form raw estimators $d_1(u_1,u_2)$ and $d_2(u_1,u_2)$ of the densities

$$d_1(u_1,u_2) = \frac{f(H_1^{-1}(u_1),\ H_2^{-1}(u_2))}{h_1H_1^{-1}(u_1)\ h_2H_2^{-1}(u_2)} \quad ,$$

$$d_2(u_1,u_2) = \frac{g(H_1^{-1}(u_1),\ H_2^{-1}(u_2))}{h_1H_1^{-1}(u_1)\ h_2H_2^{-1}(u)} \quad .$$

Therefore

$$\log d_1(u_1,u_2) - \log d_2(u_1,u_2)$$

$$= \log f(H_1^{-1}(u_1),\ H_2^{-1}(u_2)) - \log g(H_1^{-1}(u_1),\ H_2^{-1}(u_2)) \quad .$$

The likelihood ratio $f(x_1,x_2)/g(x_1,x_2)$ can be effectively estimated by estimating $\log d_1(u_1,u_2) - \log d_2(u_1,u_2)$. It seems most natural to

estimate [using exponential model representations]

$$\log d_1(u_1,u_2) - \log d_{11}(u_1) - \log d_{12}(u_2)$$

where $d_{11}(u_1)$ and $d_{12}(u_2)$ are the marginal densities of $d_1(u_1,u_2)$ which are estimated separately by methods of two samples: univariate.

The final output are contour plots of the classification statistic

$$L(x_1,x_2) = \log f(x_1,x_2) - \log g(x_1,x_2) \quad .$$

A point (x_1,x_2) is classified in population 1 or 2 by whether $L(x_1,x_2)$ exceeds a threshold which depends on the prior probabilities and loss function.

Appendix: Exploratory Quantile Data Analysis of Training Files

The basic tool for determining statistical characteristics that are good discriminators is to determine (for each file, ground truth, and channel) a data batch of measurements in the specified channel on all pixels with the specified ground truth. The statistical characteristics of these data batches are summarized (as on the attached pages) and studied to determine patterns which can discriminate between different ground truths. The file numbers are those used in the Fundamental Research Data Base [see Guseman (1983)].

OBS	FILE_NO	CHANNEL	GR_TRUTH	SMPL_SIZ	Q25	Q50	Q75	IQR	MEAN_IQ	STD_IQ	IQ01	IQ99	LOG_SPC	LOGW_SPC	LOGF_QOO	SIGMA_O	LOGS_IGO	LGSOWS
1	1	2	90	377	15.000	19	24.0000	9.0000	0.08564	0.322475	-0.3456	1.05556	-1.3536	-1.5800	-0.90686	0.50636	-0.68050	.
2	1	2	92	8755	14.000	15	19.0000	5.0000	0.17327	0.442017	-0.4000	2.80000	0.1066	-0.5821	-0.90812	0.80297	-0.21943	.
3	1	2	97	10573	14.000	15	19.0000	5.0000	0.17437	0.463493	-0.4000	2.70000	-1.3959	-1.8841	-0.90812	0.65714	-0.41986	.
4	1	2	98	793	13.000	14	17.0000	4.0000	0.14415	0.416479	-0.3750	2.25000	-1.1693	-1.5917	-0.90686	0.61600	-0.48451	.
5	1	2	100	2296	15.000	18	21.0000	6.0000	0.07333	0.448731	-0.5833	2.08333	0.0933	-0.6163	-0.90686	0.82098	-0.19726	.
6	1	2	104	558	15.000	18	23.0000	8.0000	0.09492	0.382529	-0.4375	1.18750	-2.0936	-2.4581	-0.90812	0.58059	-0.54370	.
7	1	2	111	174	19.000	23	27.0000	8.0000	-0.01331	0.326849	-0.7500	0.81250	-5.1738	-5.5843	-0.90812	0.60793	-0.49770	.
8	1	2	113	248	19.000	21	24.0000	5.0000	-0.00419	0.407626	-1.0000	1.10000	-1.5537	-2.3137	-0.90812	0.86235	-0.14809	.
9	1	2	114	95	18.000	20	24.0000	6.0000	0.08244	0.381941	-0.7500	1.00000	-1.8184	-2.3698	-0.90812	0.69993	-0.35677	.
10	1	2	164	2980	15.000	19	22.0000	7.0000	0.00592	0.373707	-0.5714	1.50000	-0.0626	-0.6454	-0.90812	0.72225	-0.32539	.
11	1	2	242	1326	15.000	19	21.0000	6.0000	-0.00299	0.367563	-0.6667	1.58333	-0.0152	-0.6651	-0.90812	0.77239	-0.25827	.
12	1	3	90	377	14.000	20	30.7500	16.7500	0.07147	0.274792	-0.3284	0.68657	-1.0524	-1.1649	-0.90686	0.45186	-0.79437	.
13	1	3	92	8755	11.500	14	18.0000	6.5000	0.09520	0.506777	-0.4615	3.00000	0.1017	-0.5798	-0.90812	0.79720	-0.22665	.
14	1	3	97	10573	10.000	13	18.0000	8.0000	0.13187	0.407418	-0.3125	1.93750	-0.1479	-0.6297	-0.90812	0.65293	-0.42629	.
15	1	3	98	793	9.500	12	14.0000	4.5000	0.06506	0.492558	-0.4444	2.55556	-0.6937	-1.3238	-0.90812	0.75730	-0.27800	.
16	1	3	100	2296	12.500	16	21.5000	9.0000	0.11257	0.407630	-0.4444	1.66667	-0.5910	-1.0575	-0.90812	0.64299	-0.44162	.
17	1	3	104	558	13.000	16	24.0000	11.0000	0.11668	0.345959	-0.3636	1.00000	-0.8114	-1.1307	-0.90812	0.55493	-0.58891	.
18	1	3	111	174	20.500	26	33.0000	12.5000	0.02218	0.339094	-0.7200	0.68000	-3.7113	-4.1667	-0.90812	0.63589	-0.45273	.
19	1	3	113	248	20.000	28	33.0000	13.0000	-0.05665	0.319116	-0.7692	0.65385	-1.3456	-1.8080	-0.90812	0.64040	-0.44567	.
20	1	3	114	95	18.000	24	29.0000	11.0000	-0.01359	0.347879	-0.7273	0.68182	-4.1139	-4.6001	-0.90812	0.65577	-0.42194	.
21	1	3	164	2980	13.000	18	24.0000	11.0000	0.05420	0.367705	-0.4545	1.40909	-0.5273	-0.9115	-0.90812	0.59221	-0.52390	.
22	1	3	242	1326	14.500	20	26.0000	11.5000	0.03849	0.350681	-0.5217	1.43478	-2.1322	-2.4893	-0.90812	0.57637	-0.55101	.
23	6	2	19	84	26.000	27	29.0000	3.0000	0.10974	0.422269	-0.8333	1.33333	-7.7867	-8.4229	-0.90686	0.76287	-0.27066	.
24	6	2	20	84	24.025	27	28.9375	4.9125	-0.02021	0.313054	-0.5089	1.01781	-7.0616	-7.3181	-0.90686	0.52187	-0.65034	.
25	6	2	21	68	26.780	27	29.0000	2.2200	0.30070	0.676187	-1.1261	2.25224	-6.9264	-7.9460	-0.90812	1.11795	0.11149	.
26	6	2	22	138	26.000	28	30.0000	4.0000	0.05138	0.400085	-0.7500	1.12500	-7.2153	-7.9296	-0.90812	0.82389	-0.19372	.
27	6	2	24	75	29.000	30	33.0000	4.0000	0.08499	0.314726	-1.0000	0.75000	-7.9774	-8.4502	-0.90812	0.64713	-0.43521	.
28	6	2	25	98	31.880	33	36.0000	4.1200	0.02669	0.411211	-1.3349	0.84951	-7.2732	-8.1135	-0.90812	0.93442	-0.06783	.
29	6	2	26	59	28.700	29	32.0000	3.3000	0.08760	0.436342	-1.0606	1.06059	-6.8373	-7.6235	-0.90812	0.88522	-0.12192	.
30	6	2	27	66	29.000	29	31.0000	2.0000	0.17336	0.514212	-1.7500	1.50000	-7.3150	-8.3163	-0.90812	1.09776	0.09327	.
31	6	2	29	90	26.920	29	30.0000	3.0800	-0.14964	0.638682	-1.1364	1.13636	-5.8349	-7.1081	-0.90812	1.44067	0.36511	.
32	6	2	30	147	30.000	33	36.0000	6.0000	-0.00780	0.289214	-0.9167	0.66667	-7.0916	-7.5682	-0.90686	0.65036	-0.43023	.
33	6	2	80	262	29.000	33	35.0000	6.0000	-0.10390	0.357624	-0.9167	0.58333	-1.7481	-2.5181	-0.90812	0.87100	-0.13811	.
34	6	2	90	110	27.000	29	31.0000	4.0000	0.04559	0.376118	-0.8750	1.12500	-7.0515	-7.5858	-0.90812	0.68816	-0.37374	.
35	6	2	92	70	30.000	32	33.0000	3.0000	-0.07507	0.374930	-1.6667	0.50000	-8.1278	-8.8591	-0.90812	0.83789	-0.17687	.
36	6	2	94	719	29.000	31	33.0000	4.0000	0.03415	0.538090	-1.1250	1.62500	-4.4185	-5.5421	-0.90812	1.24040	0.21543	.
37	6	2	95	802	29.000	30	33.0000	4.0000	0.09802	0.471782	-1.0000	1.50000	-0.4534	-1.3925	-0.90812	1.03154	0.03105	.
38	6	2	100	7449	29.000	31	33.0000	4.0000	0.00000	0.435172	-1.1250	1.37500	0.2642	-0.7829	-0.90812	1.14906	0.13894	.
39	6	2	101	667	29.000	32	34.0000	5.0000	-0.04075	0.413778	-1.0000	1.40000	-0.3265	-1.1803	-0.90812	0.94716	-0.05429	.
40	6	2	103	166	29.000	33	35.0000	6.0000	-0.09408	0.388473	-0.9167	0.75000	-6.3937	-7.0892	-0.90812	0.80842	-0.21268	.
41	6	2	104	286	29.000	30	33.0000	4.0000	0.07518	0.449739	-1.0000	1.25000	-0.6947	-1.7583	-0.90812	1.16820	0.15547	.
42	6	2	111	3033	27.500	30	33.0000	5.5000	-0.05079	0.472771	-0.7273	1.09091	-0.2712	-1.3387	-0.90812	1.17272	0.15933	.
43	6	2	112	52	29.000	30	32.0000	3.0000	-0.01842	0.545337	-1.3333	1.33333	-6.3356	-7.4244	-0.90812	1.19797	0.18063	.
44	6	2	164	3344	28.000	30	33.0000	5.0000	0.04901	0.460380	-0.8000	2.10000	-0.3333	-1.2640	-0.90812	1.02287	0.02262	.
45	6	2	240	581	15.000	20	29.0000	14.0000	0.06107	0.257827	0.0714	0.78571	-1.3245	-1.3068	-0.90812	0.39618	-0.92590	.
46	6	2	242	1724	25.000	29	33.0000	8.0000	-0.02228	0.402009	-0.4375	0.81250	0.0246	-0.8394	-0.90812	0.95689	-0.04407	.
47	6	2	250	430	29.000	32	33.0000	4.0000	-0.10988	0.482274	-1.2500	1.87500	-0.7696	-1.6981	-0.90812	1.02054	0.02033	.
48	6	3	19	84	25.000	26	29.0000	4.0000	0.15562	0.428386	-0.5000	1.12500	-7.5269	-8.0800	-0.90686	0.70208	-0.35371	.
49	6	3	20	84	23.000	26	29.0000	6.0000	0.09133	0.417561	-0.3333	1.25000	-6.1938	-6.6268	-0.90812	0.62182	-0.47511	.

OBS	FILE_NO	CHANNEL	GR_TRUTH	SMPL_SIZ	Q_25	Q_50	Q_75	I_QR	MEAN_IQ	STD_IQ	IQ_01	IQ_99	LOG_SPC	LOG_WSPC	LOG_FOQO	SIGMA_O	LOG_SIGO	LG_SO_WS
50	6	3	21	68	28.000	28.0000	30.0000	2.0000	0.27126	0.704507	-1.5000	2.00000	-6.2044	-7.4344	-0.90686	1.38154	0.32320	.
51	6	3	22	138	26.000	29.0000	31.0000	5.0000	-0.06978	0.379523	-0.7000	0.70000	-2.1660	-2.9263	-0.90812	0.86258	-0.14783	.
52	6	3	24	75	29.000	32.0000	33.0000	4.0000	-0.10535	0.368432	-1.2500	0.62501	-7.1877	-7.8403	-0.90812	0.77452	-0.25551	.
53	6	3	25	98	32.880	34.0000	36.0000	3.1200	0.02724	0.504708	-1.9231	0.96153	-1.4831	-2.5190	-0.90812	1.13631	0.12779	.
54	6	3	26	59	32.000	34.0000	39.4499	7.4500	0.06483	0.372958	-0.8054	0.87249	-4.6571	-5.2885	-0.90686	0.75918	-0.27551	.
55	6	3	27	66	28.000	29.0000	31.5849	3.5849	0.10743	0.357394	-0.9763	0.83684	-7.1108	-7.7215	-0.90812	0.74273	-0.29742	.
56	6	3	29	90	25.375	29.0000	33.0000	7.6250	0.00003	0.375881	-0.4590	0.78688	-1.6667	-2.3720	-0.90812	0.81641	-0.20284	.
57	6	3	30	147	32.000	34.0000	36.9999	4.9999	-0.01149	0.378298	-1.2000	0.70001	-5.5537	-6.3597	-0.90812	0.90293	-0.10211	.
58	6	3	80	262	31.000	35.4999	38.0000	7.0000	-0.08160	0.412245	-0.9643	0.75001	-1.5098	-2.4394	-0.90812	1.02177	0.02154	.
59	6	3	90	110	35.000	37.0000	39.1799	4.1799	-0.00605	0.457698	-1.7943	1.07657	-6.1105	-7.1068	-0.90812	1.09219	0.08818	.
60	6	3	92	70	33.000	34.5000	36.0000	3.0000	0.01198	0.453114	-2.0833	1.41667	-7.3690	-8.3061	-0.90812	1.02944	0.02901	.
61	6	3	94	719	32.000	35.0000	37.0000	5.0000	-0.07525	0.529481	-1.3000	1.20000	-4.9297	-6.1265	-0.90812	1.33465	0.28867	.
62	6	3	95	802	29.000	32.0000	36.5000	7.5000	0.04627	0.390512	-0.6667	1.13333	-0.4756	-1.0958	-0.90812	0.74982	-0.28793	.
63	6	3	100	7449	29.000	33.0000	36.0000	7.0000	-0.01981	0.343015	-0.7857	1.42857	-0.0980	-0.7663	-0.90812	0.78671	-0.23990	.
64	6	3	101	667	31.000	34.0000	37.0000	6.0000	0.00871	0.441873	-1.0000	1.16667	-0.9997	-1.8437	-0.90812	0.93786	-0.06415	.
65	6	3	103	166	29.000	33.0000	36.0000	7.0000	-0.04188	0.368805	-0.7857	0.71429	-2.2483	-2.9266	-0.90812	0.79469	-0.22981	.
66	6	3	104	286	29.500	32.0000	35.0000	5.5000	-0.02102	0.403830	-0.9091	0.90909	-6.8991	-7.7758	-0.90812	0.96907	-0.03142	.
67	6	3	111	3033	30.000	34.0000	37.0000	7.0000	-0.08816	0.522955	-0.8571	1.28571	0.2503	-0.9520	-0.90812	1.34200	0.29416	.
68	6	3	112	52	36.000	40.0000	44.0000	8.0000	-0.04710	0.376295	-1.1250	0.43750	-1.3283	-2.1256	-0.90686	0.89624	-0.10955	.
69	6	3	164	3344	29.000	33.0000	36.0000	7.0000	-0.02656	0.429949	-0.7857	1.85714	0.0516	-0.8871	-0.90812	1.03101	0.03054	.
70	6	3	240	581	10.000	18.0000	29.7498	19.7498	0.05451	0.268453	0.1013	0.70887	-1.4929	-1.4949	-0.90812	0.40409	-0.90613	.
71	6	3	242	1724	25.500	32.0000	36.0000	10.5000	-0.08133	0.404946	-0.4762	0.90476	-3.2287	-4.0342	-0.90812	0.90245	-0.10264	.
72	6	3	250	430	34.000	37.0000	41.5000	7.5000	0.03987	0.398162	-1.0000	1.06667	-1.6581	-2.4186	-0.90812	0.86273	-0.14766	.
73	11	2	2	97	38.000	40.0000	42.0000	4.0000	0.09039	0.419874	-0.6250	1.75000	-7.3856	-7.9428	-0.90686	0.70493	-0.34966	.
74	11	2	3	59	40.000	42.0000	44.0000	4.0000	-0.00626	0.363281	-0.8750	1.00000	-7.0617	-7.6737	-0.90686	0.74464	-0.29486	.
75	11	2	11	112	38.000	40.0000	40.9375	2.9375	-0.04870	0.314599	-0.8511	0.68086	-8.3158	-8.8274	-0.90686	0.67351	-0.39526	.
76	11	2	14	68	38.000	40.0000	42.0000	4.0000	0.01268	0.309138	-0.6250	0.62500	-8.1677	-8.6187	-0.90812	0.63311	-0.45711	.
77	11	2	15	154	37.000	38.0000	39.0000	2.0000	-0.05372	0.460384	-0.7500	1.00000	-8.3292	-9.3228	-0.90812	1.08931	0.08555	.
78	11	2	16	176	35.000	37.0000	40.0000	5.0000	-0.00641	0.352234	-0.2000	0.80000	-1.4149	-2.0477	-0.90812	0.75936	-0.27529	.
79	11	2	20	89	34.000	35.0000	38.0000	4.0000	0.10804	0.419158	0.0000	1.25000	-6.8548	-7.4064	-0.90812	0.70017	-0.35644	.
80	11	2	21	61	34.000	35.0000	35.0000	1.0000	-0.16516	0.970326	0.0000	2.49996	-6.7873	-8.3991	-0.90812	2.02124	0.70371	.
81	11	2	22	187	35.000	36.0000	38.0000	3.0000	0.12541	0.518729	-0.1667	3.00000	-1.4564	-2.1781	-0.90812	0.82991	-0.18644	.
82	11	2	23	276	40.000	41.0000	42.0000	2.0000	-0.02924	0.511546	-1.5000	1.50000	-7.8764	-8.9025	-0.90812	1.12527	0.11802	.
83	11	2	27	184	35.000	37.0000	38.0000	3.0000	-0.10586	0.363504	-0.3333	1.00000	-1.1591	-1.8038	-0.90812	0.76842	-0.26342	.
84	11	2	80	327	35.000	37.0000	38.0000	3.0000	-0.03545	0.467447	-0.3333	1.33333	-1.8450	-2.7436	-0.90812	0.99053	-0.00952	.
85	11	2	90	137	32.000	35.0000	38.0000	6.0000	-0.03985	0.376386	0.0000	0.58333	-6.5088	-7.3084	-0.90812	0.89714	-0.10855	.
86	11	2	99	1268	38.000	40.0000	42.0000	4.0000	0.00589	0.548328	-0.6250	2.62500	0.1869	-0.8899	-0.90812	1.18372	0.16866	.
87	11	2	100	1258	32.000	33.0000	35.0000	3.0000	0.12269	0.614127	0.3333	2.00000	-1.1158	-2.4396	-0.90812	1.51549	0.41574	.
88	11	2	101	908	36.000	38.0000	41.0000	5.0000	0.02581	0.477375	-0.3000	1.30000	-2.6301	-3.7291	-0.90812	1.21036	0.19092	.
89	11	2	104	145	22.000	30.0000	35.0000	13.0000	-0.00894	0.271717	0.1923	0.65385	-6.4074	-6.4934	-0.90812	0.43950	-0.82211	.
90	11	2	105	1215	30.500	33.0000	36.0000	5.5000	0.12760	0.647085	0.1818	5.36363	-0.5111	-1.1246	-0.90812	0.74478	-0.29467	.
91	11	2	107	12998	30.000	32.0000	36.5000	6.5000	0.19345	0.565437	0.2308	4.69231	-0.0432	-0.6174	-0.90812	0.71613	-0.33389	.
92	11	2	175	804	36.000	38.0000	40.0000	4.0000	0.01380	0.362123	-0.3750	1.12500	-0.4216	-1.1079	-0.90812	0.80097	-0.22193	.
93	11	2	176	75	40.000	40.0000	42.0000	2.0000	0.17919	0.557002	-1.2500	1.75000	-6.9107	-7.9306	-0.90686	1.11974	0.11310	.
94	11	2	179	248	40.000	41.0000	42.0000	2.0000	0.00876	0.672536	-1.5000	1.50000	-1.5916	-2.8273	-0.90812	1.38751	0.32751	.
95	11	2	240	813	35.000	38.0000	50.0000	15.0000	0.15651	0.383497	-0.1000	1.46667	-0.3297	-0.7312	-0.90812	0.60252	-0.50664	.
96	11	3	2	97	44.000	46.0000	48.3599	4.3599	-0.05873	0.461914	-1.1468	0.80276	-6.7082	-7.6067	-0.90686	0.99164	-0.00839	.
97	11	3	3	59	41.700	44.0000	46.0000	4.3000	0.00828	0.362519	-0.9302	0.81395	-7.4606	-8.0711	-0.90812	0.74262	-0.29757	.
98	11	3	11	112	41.000	43.0000	44.8750	3.8750	-0.04080	0.373627	-0.9032	0.90324	-7.7439	-8.3910	-0.90812	0.77026	-0.26103	.

OBS	FILE_NO	CHANNEL	GR_TRUTH	SMPL_SIZ	Q_25	Q_50	Q_75	I_Q_R	MEAN_IQ
99	11	3	14	68	41.0000	44.5000	48.7499	7.7499	0.00934
100	11	3	15	154	41.0000	42.0000	44.0000	3.0000	-0.01168
101	11	3	16	176	35.0000	37.0000	39.0000	4.0000	0.04833
102	11	3	20	89	35.0000	37.0000	39.0000	4.0000	0.08762
103	11	3	21	61	36.0000	37.0000	39.0000	3.0000	0.10095
104	11	3	22	187	36.0000	38.0000	39.0000	3.0000	0.10243
105	11	3	23	276	41.0000	42.0000	44.0000	3.0000	0.07719
106	11	3	27	184	37.0000	38.0000	39.0000	2.0000	-0.05223
107	11	3	80	327	38.0000	39.0000	41.0000	3.0000	0.09934
108	11	3	90	137	38.0000	41.0000	44.0000	6.0000	-0.08112
109	11	3	99	1268	42.0000	45.0000	47.0000	5.0000	0.00174
110	11	3	100	1258	34.0000	35.0000	37.0000	3.0000	0.11312
111	11	3	101	908	37.0000	39.0000	44.0000	7.0000	0.07129
112	11	3	104	145	22.0000	29.0000	37.0000	15.0000	0.02661
113	11	3	105	1215	38.0000	39.0000	44.0000	6.0000	0.19637
114	11	3	107	12998	35.0000	37.0000	41.0000	6.0000	0.14851
115	11	3	175	804	38.0000	39.0000	42.0000	4.0000	0.15476
116	11	3	176	75	39.0000	42.0000	44.5699	5.5699	0.01077
117	11	3	179	248	41.0000	44.0000	45.0000	4.0000	-0.13473
118	11	3	240	813	37.0000	41.0000	49.8197	12.8197	0.12388
119	2	2	90	377	18.0000	22.0000	26.7500	8.7500	0.01643
120	2	2	92	8755	17.0000	19.0000	21.0000	4.0000	0.08750
121	2	2	97	10573	18.0000	20.0000	24.0000	6.0000	0.10750
122	2	2	98	793	19.0000	20.0000	23.0000	4.0000	0.21854
123	2	2	100	2296	15.0000	18.0000	21.0000	6.0000	0.07270
124	2	2	104	558	14.0000	15.0000	18.0000	4.0000	0.15617

OBS	STD_IQ	IQ_01	IQ_99	LOG_SPC	LOG_WSPC	LOG_FOQO	SIGMA_O	LOG_SIGO	LG_SO_WS
99	0.235690	-0.5484	0.35484	-8.2709	-8.3490	-0.90812	0.43605	-0.83000	.
100	0.388799	-1.0000	0.83333	-8.6831	-9.3572	-0.90686	0.79239	-0.23270	.
101	0.485654	-0.1250	1.25000	-1.5900	-2.5285	-0.90812	1.03079	0.03032	.
102	0.389348	-0.1250	1.25000	-7.2055	-7.6596	-0.90812	0.63511	-0.45395	.
103	0.412752	-0.1667	1.16667	-7.0332	-7.6502	-0.90812	0.74746	-0.29108	.
104	0.758390	-0.3333	4.50000	-6.4167	-7.3487	-0.90812	1.02421	0.02393	.
105	0.435878	-0.8333	1.33333	-8.1175	-8.8666	-0.90686	0.85400	-0.15782	.
106	0.529777	-0.5000	1.75000	-7.8936	-8.9375	-0.90812	1.14551	0.13585	.
107	0.438788	-0.5000	1.66667	-2.0614	-2.8461	-0.90812	0.88397	-0.12333	.
108	0.449681	-0.4167	0.50000	-2.1215	-3.1952	-0.90812	1.18011	0.16561	.
109	0.453383	-0.9000	2.00000	0.1700	-0.7820	-0.90812	1.04488	0.04391	.
110	0.683917	0.1667	2.00000	-1.0996	-2.6160	-0.90812	1.83730	0.60830	.
111	0.377377	-0.2143	0.92857	-1.1503	-2.0485	-0.90812	0.99014	-0.00991	.
112	0.278329	0.2333	0.63333	-1.4291	-1.5725	-0.90812	0.46546	-0.76473	.
113	0.587574	-0.2500	4.83333	0.0025	-0.7074	-0.90812	0.82017	-0.19824	.
114	0.566747	-0.0833	4.58333	0.0080	-0.6664	-0.90812	0.79156	-0.23375	.
115	0.427600	-0.3750	1.75000	0.0988	-0.7925	-0.90812	0.98332	-0.01682	.
116	0.286119	-0.5386	0.89768	-7.8258	-8.0574	-0.90686	0.50897	-0.67536	.
117	0.441134	-1.0000	0.87500	-7.2207	-7.9961	-0.90812	0.87576	-0.13266	.
118	0.415452	-0.1950	1.67711	-0.6044	-1.1408	-0.90812	0.68958	-0.37167	.
119	0.343387	-0.5269	1.20000	-0.9547	-1.3491	.	0.59900	-0.51249	0.83658
120	0.572547	-0.6250	3.37500	0.3307	-0.6199	.	1.04465	0.04368	0.66354
121	0.442594	-0.5000	2.33333	0.0775	-0.6228	.	0.81336	-0.20658	0.41624
122	0.712776	-0.7500	4.25000	-0.3916	-1.2954	.	0.99698	-0.00303	1.29241
123	0.457327	-0.5025	2.33333	-0.1113	-0.7367	.	0.75462	-0.28154	0.45512
124	0.513247	-0.5000	3.12500	-1.7848	-2.2800	.	0.66252	-0.41170	1.86825

OBS	FILE_NO	CHANNEL	GR_TRUTH	SMPL_SIZ	Q_25	Q_50	Q_75	IQR	MEAN_IQ	STD_IQ	IQ_01	IQ_99	LOG_SPC	LOG_WSPC	LOG_FOQO	SIGMA_O	LOG_SIGO	LG_SO_WS
125	2	2	111	174	15.0000	17.0000	19.0000	4.0000	0.029684	0.42386	-0.5313	1.8750	-6.3085	-6.7216	.	0.610	-0.49383	6.22773
126	2	2	113	248	19.0000	20.0000	23.0000	4.0000	0.096609	0.51549	-0.7500	2.2500	-2.1118	-2.8814	.	0.872	-0.13723	2.74415
127	2	2	114	95	14.0000	15.0000	19.7199	5.7199	0.135383	0.38424	-0.4371	1.5734	-1.9182	-2.2945	.	0.588	-0.53050	1.76402
128	2	2	164	2980	18.0000	20.0000	23.7500	5.7500	0.092174	0.52215	-0.6957	3.0435	0.2152	-0.6131	.	0.924	-0.07856	0.53450
129	2	2	242	1326	17.2500	20.0000	24.0000	6.7500	0.062222	0.39022	-0.5926	1.4815	0.0099	-0.6459	.	0.778	-0.25107	0.39487
130	2	3	90	377	33.0000	43.0000	53.7500	20.7500	0.059363	0.39547	-0.5354	1.2048	-0.3070	-0.7531	.	0.631	-0.46080	0.29229
131	2	3	92	8755	24.0000	29.5000	35.7500	11.7500	0.095745	0.49171	-0.5319	2.7021	0.0642	-0.5198	.	0.724	-0.32282	0.19700
132	2	3	97	10573	24.0000	30.0000	36.7500	12.7500	0.082013	0.45035	-0.5490	2.2745	-0.7583	-1.2533	.	0.662	-0.41184	0.84146
133	2	3	98	793	27.0000	31.0000	36.0000	9.0000	0.087718	0.47173	-0.6700	1.8889	-1.2208	-1.8513	.	0.759	-0.27636	1.57490
134	2	3	100	2296	40.0000	50.5000	59.0000	19.0000	-0.012375	0.36361	-0.7771	0.9868	-0.0037	-0.6066	.	0.738	-0.30399	0.30257
135	2	3	104	558	45.0000	55.0448	65.7500	20.7500	-0.035356	0.35698	-0.8445	0.7218	-0.5098	-1.1162	.	0.741	-0.30040	0.81584
136	2	3	111	174	45.8125	50.1250	56.0000	10.1875	0.024783	0.48401	-1.1227	1.2699	-2.9237	-3.8349	.	1.004	0.00429	3.83915
137	2	3	113	248	38.2500	47.0000	52.0000	13.7500	-0.054036	0.36826	-0.9640	0.9091	-0.7254	-1.4139	.	0.804	-0.21832	1.19561
138	2	3	114	95	43.2400	49.0000	56.0000	12.7600	0.002850	0.45506	-1.0580	1.2931	-2.3053	-3.1337	.	0.925	-0.07845	3.05528
139	2	3	164	2980	26.0148	35.0000	47.0000	20.9852	0.062615	0.32554	-0.4289	1.3104	-0.7642	-1.0185	.	0.521	-0.65263	0.36583
140	2	3	242	1326	36.0000	46.0000	54.0000	18.0000	-0.024542	0.33170	-0.7222	1.0000	-0.3208	-0.7943	.	0.648	-0.43342	0.36083
141	3	2	90	377	12.0000	14.0000	16.0000	4.0000	0.019298	0.39258	-0.6250	1.6250	-1.4727	-1.9344	.	0.641	-0.44521	1.48915
142	3	2	92	8755	11.0000	11.0000	13.0000	2.0000	0.350000	0.93136	-0.5000	7.7500	0.7991	-0.4952	.	1.473	0.38740	0.88257
143	3	2	97	10573	11.0000	13.0000	15.0000	4.0000	0.065000	0.47157	-0.5000	3.5000	0.1337	-0.5785	.	0.823	-0.19460	0.38393
144	3	2	98	793	13.0000	14.0000	15.0000	2.0000	0.096596	0.57447	-0.7500	3.7500	0.0860	-0.9220	.	1.106	0.10108	1.02305
145	3	2	100	2296	12.0000	14.0000	16.0000	4.0000	0.043784	0.42661	-0.5000	3.1250	0.1147	-0.6083	.	0.832	-0.18377	0.42457
146	3	2	104	558	12.0000	14.0000	15.8124	3.8124	0.022792	0.37163	-0.5246	2.0984	-1.2114	-1.6041	.	0.598	-0.51412	1.08997
147	3	2	111	174	11.0000	13.0000	14.8125	3.8125	0.026883	0.28774	-0.3934	0.7869	-1.0595	-1.3541	.	0.542	-0.61223	0.74191
148	3	2	113	248	13.0000	14.0000	16.0000	3.0000	0.081615	0.40880	-0.6667	1.0000	-0.2959	-1.1161	.	0.917	-0.08662	1.02951
149	3	2	114	95	11.0000	12.4800	14.0000	3.0000	0.044670	0.32296	-0.5800	0.7533	-2.2721	-2.6252	.	0.575	-0.55372	2.07150
150	3	2	164	2980	11.0000	13.0000	15.0000	4.0000	0.096250	0.50341	-0.5000	3.6250	0.1843	-0.5748	.	0.863	-0.14767	0.42717
151	3	2	242	1326	12.0000	14.0000	16.0000	4.0000	0.071112	0.41081	-0.5000	1.6250	-1.4259	-1.9329	.	0.670	-0.39989	1.53298
152	3	3	90	377	39.0000	43.0000	48.0000	9.0000	0.047341	0.39573	-0.7222	1.0556	-1.7096	-2.2544	.	0.696	-0.36207	1.89231
153	3	3	92	8755	39.2500	43.5000	47.0000	7.7500	-0.009678	0.37780	-1.0000	1.5161	-0.0750	-0.8784	.	0.902	-0.10349	0.77487
154	3	3	97	10573	39.0000	44.0000	48.0000	9.0000	0.009338	0.40226	-0.9444	1.6667	-0.8617	-1.5676	.	0.818	-0.20099	1.36656
155	3	3	98	793	45.0000	50.0000	53.0000	8.0000	-0.096465	0.43597	-1.2500	0.8750	-0.5014	-1.4274	.	1.019	0.01915	1.44652
156	3	3	100	2296	37.0000	41.0000	45.7500	8.7500	0.030151	0.36095	-0.7429	1.3143	-0.9597	-1.5350	.	0.718	-0.33160	1.20336
157	3	3	104	558	37.0000	40.0000	43.0000	6.0000	0.020173	0.46660	-0.8333	1.7500	-6.3428	-7.0865	.	0.849	-0.16320	6.92327
158	3	3	111	174	37.0000	43.0000	48.7499	11.7499	0.018509	0.31700	-0.5213	0.7660	-5.1939	-5.5167	.	0.558	-0.58408	4.93257
159	3	3	113	248	39.0000	43.0000	46.1799	7.1799	0.040447	0.43400	-0.7319	1.5321	-5.8781	-6.5004	.	0.752	-0.28456	6.21582
160	3	3	114	95	39.0000	41.0000	44.0000	5.0000	0.012112	0.38422	-1.0000	0.9000	-6.6068	-7.2926	.	0.802	-0.22108	7.07153
161	3	3	164	2980	37.0000	42.0000	46.0000	9.0000	-0.015330	0.37170	-0.8889	1.3889	-0.5355	-1.1915	.	0.778	-0.25088	0.94057
162	3	3	242	1326	40.2500	43.0000	47.0000	6.7500	0.051269	0.42644	-0.9430	2.0741	-0.8711	-1.6134	.	0.848	-0.16458	1.44881
163	4	2	90	377	13.0000	15.0000	19.0000	6.0000	0.088114	0.36715	-0.4167	1.0833	-2.0987	-2.4372	.	0.566	-0.56843	1.86873
164	4	2	92	8755	11.0000	12.0000	13.0000	2.0000	0.265000	1.07239	-0.5000	8.5000	0.8443	-0.4832	.	1.523	0.42060	0.90379
165	4	2	97	10573	11.0000	12.0000	14.0000	3.0000	0.250000	0.67607	-0.3333	4.6667	0.5014	-0.4751	.	1.072	0.06970	0.54485
166	4	2	98	793	15.0000	15.5000	18.0000	3.0000	0.130000	0.46486	-0.7500	2.5833	0.4238	-0.6269	.	1.155	0.14385	0.77074
167	4	2	100	2296	15.0000	19.0000	21.7500	6.7500	-0.022091	0.35933	-0.5926	1.2593	-0.0866	-0.6918	.	0.740	-0.30161	0.39021
168	4	2	104	558	15.0000	20.0000	24.0000	9.0000	0.038804	0.37734	-0.5000	1.3333	-0.7724	-1.2062	.	0.623	-0.47307	0.73314
169	4	2	111	174	14.0000	15.0000	18.0000	4.0000	0.113748	0.40674	-0.5000	1.5000	-0.3056	-0.9627	.	0.779	-0.24976	0.71297
170	4	2	113	248	16.0000	19.0000	22.0000	6.0000	0.078022	0.48477	-0.6667	1.4167	-0.9834	-1.6811	.	0.811	-0.20925	1.47180
171	4	2	114	95	12.0000	14.0000	17.0000	5.0000	0.128838	0.45227	-0.4000	1.6000	-1.4567	-1.8520	.	0.600	-0.51155	1.34042
172	4	2	164	2980	12.0000	13.0000	15.0000	3.0000	0.236666	0.74993	-0.5000	4.1667	0.5905	-0.5389	.	1.249	0.22254	0.76144
173	4	2	242	1326	13.0000	15.0000	19.0000	6.0000	0.120547	0.39356	-0.4167	1.7500	-1.3534	-1.7234	.	0.585	-0.53691	1.18648

OBS	FILE NO	CHAN EL	GR TRUTH	SMPL SIZ	Q25	Q50	Q75	IQR	MEAN IQ	STD IQ	IQ O1	IQ 99	LOG SPC	LOG WSPC	LOG FOQO	SIGMA O	LOG SIG O	LOG S IW S
174	4	3	90	377	47.0000	51.0000	57.0000	10.0000	0.11187	0.54211	-0.9720	1.8000	-1.0773	-1.9210	.	0.939	-0.06316	1.85783
175	4	3	92	8755	49.0000	54.0000	59.0000	10.0000	0.00794	0.40551	-1.2000	1.7500	0.0179	-0.8324	.	0.945	-0.05657	0.77584
176	4	3	97	10573	0.0000	0.0000	0.0000	0.0000	0.54000	5.40000	0.0000	54.0000	5.6519	-0.2058	.	141.298	4.95087	5.15666
177	4	3	98	793	61.0000	71.0000	77.7500	16.7500	-0.04775	0.33029	-0.8060	0.7463	-0.9628	-1.4863	.	0.682	-0.38332	1.10298
178	4	3	100	2296	43.0000	49.0000	56.0000	13.0000	0.03077	0.39067	-0.8077	1.5385	0.0063	-0.6597	.	0.786	-0.24087	0.41878
179	4	3	104	558	40.0000	47.0000	53.0000	13.0000	0.01238	0.39469	-0.6538	1.8077	-0.3133	-0.8201	.	0.670	-0.40002	0.42011
180	4	3	111	174	46.2500	55.0000	62.0000	15.7500	0.00690	0.32112	-0.6349	0.7302	-3.7986	-4.1619	.	0.581	-0.54366	3.61820
181	4	3	113	248	44.0000	49.0000	53.0000	9.0000	-0.00196	0.37798	-0.8333	1.0000	-1.3375	-1.9590	.	0.752	-0.28540	1.67357
182	4	3	114	95	47.0000	51.0000	55.0000	8.0000	-0.03403	0.50460	-2.1875	1.0625	-0.9529	-2.2132	.	1.424	0.35340	2.56656
183	4	3	164	2980	47.0000	53.0000	59.0000	12.0000	0.01909	0.42414	-1.0913	1.5833	-1.2025	-1.9647	.	0.865	-0.14471	1.81998
184	4	3	242	1326	49.2500	54.0000	59.0000	9.7500	0.05192	0.44531	-1.0769	1.5897	-0.5454	-1.3403	.	0.894	-0.11194	1.22837
185	5	2	90	377	14.0000	16.0000	24.0000	10.0000	0.16075	0.37199	-0.2500	1.3500	-1.4246	-1.6591	.	0.510	-0.67239	0.98671
186	5	2	92	8755	12.2500	14.0000	15.0000	2.7500	0.23235	0.89847	-0.5455	6.0000	-0.4299	-1.3389	.	1.002	0.00215	1.34102
187	5	2	97	10573	12.0000	12.5000	15.0000	3.0000	0.30333	0.84649	-0.4167	5.5833	0.5960	-0.4769	.	1.181	0.16602	0.64293
188	5	2	98	793	15.0000	16.0000	18.0000	3.0000	0.19833	0.65293	-0.8333	3.6667	0.5786	-0.5688	.	1.272	0.24050	0.80929
189	5	2	100	2296	20.0000	26.0000	32.0000	12.0000	0.02053	0.36533	-0.5833	1.1667	-0.0841	-0.6151	.	0.687	-0.37587	0.23926
190	5	2	104	558	20.8124	29.5000	37.0000	16.1876	-0.00049	0.32287	-0.5405	0.7568	-1.3346	-1.6888	.	0.575	-0.55274	1.13603
191	5	2	111	174	15.2500	18.7500	21.8125	6.5625	0.03371	0.35199	-0.4571	1.6953	-5.8105	-6.0374	.	0.507	-0.67999	5.35745
192	5	2	113	248	18.0000	21.0000	23.7500	5.7500	-0.00767	0.33992	-0.7826	0.9565	-1.2237	-1.7680	.	0.696	-0.36247	1.40558
193	5	2	114	95	14.0000	17.0000	20.0000	6.0000	0.06910	0.47639	-0.5833	1.8333	-1.3445	-1.8854	.	0.694	-0.36594	1.51947
194	5	2	164	2980	13.0000	15.0000	19.0000	6.0000	0.16106	0.53115	-0.3333	3.0000	-0.1000	-0.7010	.	0.736	-0.30590	0.39508
195	5	2	242	1326	14.0000	18.0000	22.0000	8.0000	0.06258	0.39451	-0.4375	1.6250	-1.9339	-2.3006	.	0.583	-0.54022	1.76037
196	5	3	90	377	49.0000	55.3899	62.0000	13.0000	0.03732	0.33783	-0.4804	1.1388	-4.9206	-5.2071	.	0.538	-0.62028	4.58686
197	5	3	92	8755	50.0000	54.0000	58.0000	8.0000	0.06437	0.47052	-1.1250	2.1875	0.1278	-0.7405	.	0.962	-0.03859	0.70192
198	5	3	97	10573	60.2500	68.0000	74.7500	14.5000	-0.01621	0.33677	-0.7586	1.0000	-0.1971	-0.7710	.	0.717	-0.33296	0.43803
199	5	3	98	793	65.0000	74.0000	77.8748	12.8748	-0.10873	0.35418	-1.0486	0.5825	-0.6155	-1.3129	.	0.811	-0.20954	1.10331
200	5	3	100	2296	49.0000	53.0000	58.0000	9.0000	0.05001	0.46307	-1.0000	1.9444	0.1266	-0.7220	.	0.943	-0.05830	0.66370
201	5	3	104	558	47.8124	53.0000	57.7500	9.9376	0.01209	0.42672	-0.8050	1.4591	-4.8370	-5.4721	.	0.762	-0.27180	5.20027
202	5	3	111	174	45.1250	54.0000	61.7499	16.6249	-0.00587	0.28341	-0.5113	0.6015	-4.4886	-4.7421	.	0.520	-0.65337	4.08873
203	5	3	113	248	47.0000	50.0000	55.0000	8.0000	0.09465	0.47385	-1.5331	1.5000	-1.3288	-2.3599	.	1.132	0.12421	2.48407
204	5	3	114	95	44.0000	49.0000	54.0000	10.0000	0.01776	0.45430	-1.1000	1.3000	-3.4513	-4.2417	.	0.890	-0.11649	4.12522
205	5	3	164	2980	50.0000	55.0000	62.0000	12.0000	0.05976	0.39738	-0.9167	1.4167	-0.5358	-1.2137	.	0.795	-0.22891	0.98481
206	5	3	242	1326	50.0000	54.5000	61.7500	11.7500	0.08167	0.40925	-0.7447	1.4681	-0.7698	-1.3693	.	0.735	-0.30730	1.06203

References

Guseman, L. F. (1983) Proceedings of the NASA/MPRIA Workshop: Math/Stat. Appendix: Fundamental Research Data Base.

Parzen, E. (1979) Nonparametric Statistical Data Modeling. Journal of the American Statistical Association, 74, 105-131.

Parzen, E. (1983) FUN.STAT Quantile Approach to Two Sample Statistical Data Analysis. The Canadian Journal of Statistics.

Woodfield, Terry Joe (1982) Statistical Modeling of Bivariate Data. Ph.D. Thesis Texas A&M Department of Statistics.

DISCRIMINATION RELATIVE TO MEASURES OF NON-NORMALITY

W. B. Smith
Texas A&M University

and

E. P. Shine
Texas A&M University

ABSTRACT

The objective of this research is to investigate the robustness of discriminant functions to non-normality. This study will assess the performance of procedures relative to measures of the difference between the actual distribution of the observations and the usual assumption of normal densities. For example, the two population, mixed distributions problem with equal costs of misclassification will be considered. The parameters will be estimated by maximum likelihood and recently proposed robust methods.

the LDF and the quadratic discriminant function (QDF) when covariance matrices are unequal. She restricted the covariance matrices to differ by only a scalar multiple, $\Sigma_2 = \sigma^2\Sigma_1$, $\sigma^2 > 0$.

Adopting a canonical form from Dunn and Holloway [5] the densities were transformed to $N(\underline{0}, I)$ in Π_1 and $N(\underline{\mu}, \sigma^2 I)$ in Π_2. The LDF did satisfactorily in a moderate range of σ^2 (near one) and improved as the distance between populations increased. Marks and Dunn [13] also investigated the performance of discriminant functions when covariance matrices differed. They considered a more general model which has canonical form $N(\underline{0}, I)$ in Π_1 and $N(\underline{\mu}, \Lambda)$, in Π_2, where Λ = diag $(\lambda, \ldots, \lambda, 1, \ldots, 1)$. Using Monte Carlo methods the sample LDF outperformed the sample QDF only in a small range of λ^2 near one.

Several studies have been performed to investigate the LDF under non-normality. Lachenbruch, Sneeringer, and Revo [11] used three types of non-linear transformations discussed in Johnson [10] to study the LDF under non-normal conditions. They performed a Monte Carlo experiment to simulate sampling from non-normal populations and compared the misclassification probabilities to those expected when both populations are normal. The sample LDF exhibited substantial differences from expectations when sampling from normal populations, and the overall misclassification probabilities increased for some of the transformations. Moreover, the authors found the misclassification probabilities for one population to be larger than expected while the other population was smaller than expected. They did note that the range of the variables affected the performance of the sample LDF; a bounded variable produced less effect than an unbounded variable. Their study was restricted to

independently distributed variables.

Ashikaga ([2], [3]) has studied the LDF using the mixed-normal distribution. The model is

$$f_1(\underline{X}) = (1 - \alpha^{(1)})N(\underline{\mu}_1^{(1)}, \Sigma_1) + \alpha^{(1)} N(\underline{\mu}_1^{(1)} + \underline{\theta}, \sigma^2\Sigma_1)$$

and

$$f_2(\underline{X}) = (1 - \alpha^{(2)})N(\mu_1^{(2)}, \Sigma_1) + \alpha^{(2)} N(\underline{\mu}_1^{(2)} + \underline{\theta}, \sigma^2\Sigma_1), \sigma^2 > 0.$$

In canonical form these models reduce to

$$g_1(\underline{X}) = (1 - \alpha^{(1)})N(\underline{0}, I) + \alpha^{(1)}N(C'\underline{\theta}, \sigma^2 I)$$

and

$$g_2(\underline{X}) = (1-\alpha^{(2)})N(C'(\underline{\mu}_1^{(2)}-\underline{\mu}_1^{(1)}),I)+\alpha^{(2)}N(C'(\underline{\mu}_1^{(2)}-\underline{\mu}_1^{(1)}) + C'\underline{\theta}, \sigma^2 I),$$

where $\Sigma_1 = CC'$. In choosing mixture proportions ($\alpha^{(1)}$ and $\alpha^{(2)}$), Ashikaga considered both populations to have mixed-normal distributions ($\alpha^{(1)} = \alpha^{(2)}$) and the case where one population had an assumed normal distribution while the other had a mixed-normal distribution ($\alpha^{(1)} = 0$, $\alpha^{(2)} \neq 0$).

The distinctive feature of Ashikaga's study was the introduction of two measures of non-normality which illuminate relationships between the robustness of the LDF and the extent of non-normality. The first measure was based on a multivariate skewness statistic of Malkovitch and Afifi [12]. A second measure derived by the author was an overall measure of non-normality, it being the sample size necessary to test a simple hypothesis that an observation is from a normal population versus the alternative that it was from a mixed normal population. When $\alpha^{(1)} = \alpha^{(2)}$ with identical distributions, the LDF did well if Π_1 and Π_2 had sufficient separation, (say, $\Delta^2 > 4$, where Δ^2 is Mahalanobis distance).

In the case of one population having a normal distribution and the other a mixed-normal distribution, the LDF performed poorly for all values of Δ^2.

Randles and others [14] constructed two discriminant functions to be robust to changes in the population model. The first is a generalization of Fisher's [6] approach to derive the linear discriminant function by maximizing the separation of the groups,

$$\underline{\beta}'(\bar{\underline{x}}^{(1)} - \bar{\underline{x}}^{(2)})/\sqrt{\underline{\beta}'S\underline{\beta}}.$$

If $\underline{m} = (\bar{\underline{x}}^{(1)} + \bar{\underline{x}}^{(2)})/2$, then Randles found the vector $\underline{\beta}_o$ which maximizes

$$\frac{1}{n_1}\sum_{i=1}^{n_1}\tau([\underline{\beta}'(\underline{x}_i - \underline{m})]/\sqrt{\underline{\beta}'S\underline{\beta}}) + \frac{1}{n_2}\sum_{i=1}^{n_2}\tau([\underline{\beta}'(\underline{m} - \underline{x}_i)]/\sqrt{\underline{\beta}'S\underline{\beta}}),$$

where τ is a nondecreasing and nonconstant, odd function. The function τ is selected to reduce the influence of observations far away from the center, $\underline{m}$.

The second procedure is to substitute Huber-type estimates for the parameters in the linear discriminant function. This method replaces each mean and covariance matrix with robust estimators

$$\bar{\underline{x}}_*^j = \sum_{i=1}^{n_j} w_i\underline{x}_i / \sum_{i=1}^{n_j} w_i$$

and

$$S_*^j = \sum_{i=1}^{n_j} w_i^2(\underline{x}_i - \bar{\underline{x}}^i)(\underline{x}_i - \underline{x}^i)' / \sum_{i=1}^{n_j} w_i^2, \quad j=1, 2,$$

respectively. The weights are

$$w_i = 2/d_i, \text{ if } d_i > 2,$$
$$= 1, \text{ if } d_i \leq 2$$

and the distance d_i is defined by

$$d_i = [(\underline{x}_i - \bar{\underline{x}})'(S^j)^{-1}(\underline{x}_i - \bar{\underline{x}})]^{\frac{1}{2}}.$$

Randles has found five iterations are sufficient to reduce the effect of outliers by computing new distances and weights at each stage using the robust estimates of the previous stage.

2. Bayes' Classification and Mixture Distributions

An observation is classified into one of q populations (denoted by $\Pi_1, \Pi_2, \ldots, \Pi_q$) on the basis of a discrimination rule and a p-vector observation, $\underline{x} = (x_1, x_2, \ldots, x_p)'$. Assume that the populations have equal costs of misclassification, but possibly different prior probabilities (denoted by $\pi_1, \pi_2, \ldots, \pi_q$). Also, assume that the distribution of $\underline{X}$ is a composition or mixture of m component distributions. Thus, if $f_j(\underline{x})$ represents the p.d.f. for Π_j, then

$$(1) \qquad f_j(\underline{x}) = \sum_{i=1}^{m} \alpha_i^{(j)} g_i^{(j)}(\underline{x}),$$

where $g_i^{(j)}(\underline{x})$ is the i^{th} component p.d.f. and $\alpha_i^{(j)}$ is the i^{th} component mixing proportion for population j; i=1, ..., m and j=1, ..., q. Equation (1) allows for a richer and more flexible class of p.d.f.'s than used in previous studies.

In general, a classification rule should depend upon whether or not the source component of an observation can be identified and this information incorporated. For example, let the q populations represent forest/terrain types. Then multispectral scanner measurements on each population could be represented as a mixture of components (equation (1)). Additionally, if in sampling a sub-pixel could be pure and identified as having a observation from a particular component (say,

1. Literature Review

Recall, that Fisher [6] proposed the linear discriminant function (LDF) to classify an individual into one of two populations, Π_1 and Π_2. Let S be the pooled covariance matrix and $\underline{d} = \bar{\underline{x}}^{(1)} - \bar{\underline{x}}^{(2)}$ be the difference between means from samples drawn from the two populations. Then the sample LDF is

$$\ell(\underline{x}) = (\underline{x} - \tfrac{1}{2}(\bar{\underline{x}}^{(1)} + \bar{\underline{x}}^{(2)}))' S^{-1}\underline{d}.$$

While derivation was independent of any distributional assumption, it required that the populations have the same covariance matrix.

Welch [16] obtained the Bayes' classification rule which minimizes the average probability of misclassification when prior probabilities that an individual was selected from Π_1 or Π_2 are known. He established that the LDF was optimal (in the Bayes sense) if the observations in both populations are normally distributed with the same covariance matrix. Later, Wald [15] generalized this procedure to include costs of misclassification and also replaced any unknown parameters by their maximum likelihood estimates. Hoel and Peterson [8] extended these results to include more than two populations.

However, in practice the assumptions under which the linear discriminant function is Bayes are seldom satisfied. Nonetheless, the linear discriminant function with parameters estimated from training samples (sample LDF) is widely used and serves as a benchmark by which other procedures are judged.

A number of studies have considered the behavior of the LDF when assumptions under which it is optimal are violated. Gilbert [7] compared

slash pine), then this information should be used both in estimation and classification. For other applications, see Chang and Afifi [4].

It is more likely however, that such additional information is unavailable. Hosmer and Dick [9] present a fisheries example to illustrate this situation. The case in which the observation's component is not known will be the basic model for this study, but the known component case will also be considered.

2. Component Identity Known

Suppose $\underline{x}$ is known to come from component a and define an indicator vector $\underline{y} = (y_1, y_2, \ldots, y_m)'$, such that

$$y_k = \begin{cases} 1, & k = a \\ 0, & k \neq a \end{cases} .$$

In this case $\underline{y}$ follows a multinomial distribution with parameters n=1 and $\alpha_1^{(1)}, \alpha_2^{(1)}, \ldots, \alpha_m^{(1)}$ in Π_j. Let the conditional distribution of $\underline{X}$ given $\underline{y}$ be $g_a^{(j)}(\underline{x})$ for component a in Π_j. Then the joint distributions are

$$f(\underline{x}, \underline{y}) = g_a^{(j)}(\underline{x}) \prod_{i=1}^{m} \alpha_a^{(j)}, \tag{2}$$

in Π_j j=1, ..., q. For $y_a = 1$ the Bayes' classification rule is:

Assign observation $\underline{x}$ to Π_k if

$$\pi_k \alpha_a^{(k)} g_a^{(k)}(\underline{x}) \geq \pi_j \alpha_a^{(j)} g_a^{(j)}(\underline{x}) \tag{3}$$

for all $j \neq k$. If (3) is satisfied for more than one population Π_k, then assign the observation to population min(k).

Then $d_a(x) = \pi_j \alpha_a^{(j)} g_a^{(j)}(\underline{x})$ is the discriminant score for observations from Π_j given the $y_a = 1$. In this case probability of correct classification is

$$(4) \qquad P_{Corr(a)} = \sum_{k=1}^{q} \frac{\pi_k \alpha_a^{(k)}}{\sum_{\ell=1}^{q} \pi_\ell \alpha_a^{(\ell)}} \int_{D_{k|a}} g_a^{(k)}(\underline{x}) d\underline{x},$$

where $D = \bigcup_{k=1}^{q} D_{k|a}$, $D_{k|a} \cap D_{k'|a} = \phi$, $k \neq k'$, is a partition of the sample space of $\underline{x}$ determined by the Bayes rule. And the total probability of correct classification is

$$(5) \qquad \begin{aligned} P_{Corr} &= \sum_{i=1}^{m} \left(\sum_{\ell=1}^{q} \pi_\ell \alpha_i^{(\ell)} \right) P_{Corr(i)} \\ &= \sum_{k=1}^{q} \sum_{i=1}^{m} \pi_k \alpha_i^{(k)} \int_{D_{k|i}} g_i^{(k)}(\underline{x}) d\underline{x} \ . \end{aligned}$$

A special case of the above result is given by Chang and Afifi (1974) by considering the two population case when the conditional distribution of $\underline{X}$ given $\underline{Y} = \underline{y}$ is multivariate normal (see Table 1).

TABLE 1

Chang and Afifi's [4] Model for Barbituate Overdosers

	Population 1 (Survivors)	Population 2 (Died)
Prior Probability	π_1	π_2
Component 1 (Short-acting Drug)	$\alpha_1^{(1)} = 1-\theta_1$ $g_1^{(1)} = N(\underline{x}; \underline{\mu}_1, \Sigma)$	$\alpha_1^{(2)} = 1-\theta_2$ $g_1^{(2)} = N(\underline{x}; \underline{\mu}_2, \Sigma)$
Component 2 (Long-lasting Drug)	$\alpha_2^{(1)} = \theta_1$ $g_2^{(1)} = N(\underline{x}; \underline{\mu}_1 + \Delta, \Sigma+\Gamma)$	$\alpha_2^{(2)} = \theta_2$ $g_2^{(2)} = N(\underline{x}; \underline{\mu}_2 + \Delta, \Sigma+\Gamma)$

The Bayes' classification procedure results in a "double" LDF rule:

If $\underline{x}$ belongs to component 1, then assign $\underline{x}$ to Π_1 if

$$-\tfrac{1}{2}(\underline{\mu}_1 + \underline{\mu}_2)'\Sigma^{-1}(\underline{\mu}_1 - \underline{\mu}_2) + \underline{x}'\Sigma^{-1}(\underline{\mu}_1 - \underline{\mu}_2) + \log(\theta_1/\theta_2) \geq \log(\pi_2/\pi_1).$$

Likewise, if $\underline{x}$ belongs to component 2,

then assign $\underline{x}$ to Π_1 if

$$-\tfrac{1}{2}(\underline{\mu}_1 + \mu_2)'(\Sigma+\Gamma)^{-1}(\underline{\mu}_1 - \underline{\mu}_2) + \underline{x}'(\Sigma+\Gamma)^{-1}(\underline{\mu}_1 - \underline{\mu}_2) + \log[(1-\theta_1)/(1-\theta_2)] > \log(\pi_2/\pi_1) \tag{6}$$

If $\underline{x}$ is not assigned to Π_1, then assign $\underline{x}$ to Π_2.

2. Component Identity Unknown

When the component indicator vector $\underline{Y}$ is unknown the only data available is $\underline{X}$. The component may be unknown because the pixel may be mixed, or the data could be retrospective or too costly to obtain. The class component densities are given in equation (1), and Bayes rule is:

Assign $\underline{x}$ to Π_k if

$$\pi_k \sum_{i=1}^{m} \alpha_i^{(k)} g_i^{(k)}(\underline{x}) \geq \pi_j \sum_{i=1}^{m} \alpha_i^{(j)} g_i^{(j)}(\underline{x}) \tag{7}$$

for $j=1, \ldots, 1$, and k is the smallest index for which the inequalities hold.

The probability of correctly classifying $\underline{x}$ is

$$P_{Corr(D)} = \sum_{k=1}^{q} \pi_k \sum_{i=1}^{m} \alpha_i^{(k)} \int_{D_k} g_i^{(k)}(\underline{x})d\underline{x} \tag{8}$$

where $D = \bigcup_{k=1}^{q} D_k$ is a Bayes' rule partition of the sample space.

Risk Characterization

Let $R_i^{(k)}(\underline{x})$ be the risk that an observation $(\underline{x})$ is from the i^{th} component, given that $\underline{x} \in \Pi_k$; i=1, 2, ..., m; k=1, 2, ..., q. Thus,

$$R_i^{(k)}(\underline{x}) = P\{\underline{X} \in \text{Component } i \mid \underline{X} = \underline{x} \in \Pi_k\}$$

$$= P\{Y_i = 1 \mid \underline{X} = \underline{x} \in \Pi_k\}$$

$$(9) \qquad = \frac{\alpha_i^{(k)} \cdot g_i^{(k)}(\underline{x})}{\sum_{\ell=1}^{m} \alpha_\ell^{(k)} \cdot g_\ell^{(k)}(\underline{x})}$$

Now relate the Bayes' classification rule when the component of $\underline{x}$ is unknown to the m possible Bayes' classification rules when the component of $\underline{x}$ is known. Define

$$(10) \qquad \Lambda_{j,k} = \frac{\pi_k \sum_{i=1}^{m} \alpha_i^{(k)} \cdot f_i^{(k)}(\underline{x})}{\pi_j \sum_{i=1}^{m} \alpha_i^{(j)} \cdot f_i^{(j)}(\underline{x})}$$

to be the weighted likelihood ratio for Π_j and Π_k under model (2). Similarly, define

$$(11) \qquad \lambda_{j,k}^{(i)} = \pi_k \alpha_i^{(k)} g_i^{(k)}(\underline{x}) / \pi_j \alpha_i^{(j)} g_i^{(j)}(\underline{x}).$$

Theorem 1: If $R_i^{(k)}(\underline{x}) > 0$ for i=1, ..., m then $\Lambda_{j,k}(\underline{x}) > 1$ if and only if

$$\sum_{i=1}^{m} R_i^{(i)}(\underline{x}) \, [\lambda_{j,k}^{(i)}(\underline{x})]^{-1} \leq 1.$$

Proof: From (10) $\Lambda_{j,k}(\underline{x}) > 1$ implies that

$$\sum_{i=1}^{m} (\pi_k \alpha_i^{(k)} g_i^{(k)}(\underline{x}) - \pi_j \alpha_i^{(j)} g_i^{(j)}(\underline{x})) \geq 0.$$

Algebraic manipulations yield the result.

For a particular value of $\underline{x}$ and $R_i^{(k)}(\underline{x}) > 0$ for i=1, 2, ..., m. note that the Bayes' classification rule when the component is unidentified is a convex combination of the m alternative rules in the component of Π_k. Now compare the Bayes' rules for component known model with model with component unknown. First, define the indicator function as follows:

$$(12) \qquad I(\underline{x}) = \begin{cases} 1, \text{ if } \underline{x} \in \text{Component } \ell \\ 0, \text{ if } \underline{x} \notin \text{Component } \ell \end{cases}$$

Bayes' rule: Component of $\underline{x}$ known:

Assign $\underline{x}$ to Π_1, if

$$\sum_{i=1}^{m} I_i(\underline{x})\lambda_{j,k}^{(i)}(\underline{x})^{-1} \leq 1$$

for j, k = 1, 2, ..., q. If this inequality holds for more than one value of k, then assign $\underline{x}$ to the population with smallest k.

Bayes' rule: Component of $\underline{x}$ known and $R_i^{(k)}(\underline{x}) \geq 0$ for i=1, 2, ..., m:

Assign $\underline{x}$ to Π_k if

$$\sum_{i=1}^{m} R_i^{(k)}(\underline{x})\lambda_{j,k}^{(i)}(\underline{x}) \leq 1$$

for j, k = 1, 2, ..., q. If this inequality holds for more than one value of k, then assign $\underline{x}$ to the population with smallest k.

Example 1: Let population j have an m-component distribution where i^{th} component has p-variate normal distribution with

$Q_{i,\ell}(\underline{x})$ is a QDF and

$L^{(j,k)}(\underline{x})$ is the LDF.

Theorem 2:

If $R_i^{(k)}(\underline{x}) = 0$ for $i \in I^\circ \subset \{1, 2, \ldots, m\}$, then $\Lambda_{j,k}(\underline{x}) \geq 1$ if and only if

$$[\Lambda_{j,k}(\underline{x})]^{-1} \sum_{i\in I^\circ} R_i^{(j)}(\underline{x}) + \sum_{i\notin I^\circ} R_i^{(k)}(\underline{x}) [\lambda_{j,k}^{(i)}]^{-1} \leq 1$$

Proof:

$$\Lambda_{j,k}(\underline{x}) = \frac{\pi_k \sum_{i=1}^{m} \alpha_i^{(k)} \cdot g_i^{(k)}(\underline{x})}{\pi_j \sum_{i=1}^{m} \alpha_i^{(j)} \cdot g_i^{(j)}(\underline{x})} \geq 1$$

$$\sum_{i=1}^{m} \left(\frac{\alpha_i^{(k)} g_i^{(k)}(\underline{x})}{\sum_{\ell=1}^{m} \alpha_\ell^{(k)} g_\ell^{(k)}(\underline{x})} - \frac{\pi_j \alpha_i^{(j)} \cdot g_i^{(j)}(\underline{x})}{\pi_k \cdot \sum_{\ell=1}^{m} \alpha_\ell^{(k)} \cdot g_\ell^{(k)}(\underline{x})} \right) \geq 0$$

$$- \sum_{i\in I^\circ} \frac{\pi_j \alpha_i^{(j)} \cdot g_i^{(j)}(\underline{x})}{\pi_k \sum_{\ell=1}^{m} \alpha_\ell^{(k)} \cdot g_\ell^{(k)}(\underline{x})} + \sum_{i\notin I^\circ} R_i^{(k)}(\underline{x}) [1 - \lambda_{j,k}^{-1}] \geq 0$$

$$\Lambda_{j,k}(\underline{x})^{-1} \sum_{i\in I^\circ} R_i^{(j)}(\underline{x}) + \sum_{i\notin I^\circ} R_i^{(k)}(\underline{x}) \lambda_{j,k}^{-1}(\underline{x}) \leq 1$$

Corollary:

If $\lambda_{j,k}^{(i)} \geq 1$ for $i = 1, 2, \ldots, m$, then assign the observation (whose component is unknown) to Π_k.

Proof:

This is a direct result of (11) and (16). Thus, all observations which would be classified into Π_k regardless of their component of origin, will be assigned to Π_k when the component is unknown. Those observations

$$\text{mean} = \underline{\mu}_i + \underline{\theta}_i^{(j)}$$

$$\text{variance} = \sigma_i^2\Sigma$$

and mixture parameters $\alpha_i^{(j)}$, j=1, ..., q. Also, define $\phi_{i,\ell}^{(k)}$, the weighted likelihood function, to be

$$(15) \qquad \phi_{i,\ell}^{(k)} = \alpha_\ell^{(k)} g_\ell^{(k)}(\underline{x}) / \alpha_i^{(k)} g_i^{(k)}(\underline{x}) = R_\ell^{(k)}(\underline{x}) / R_i^{(k)}(\underline{x}).$$

Then, $\phi_{i,\ell}^{(k)}$ represents the weighted likelihood ratio between the densities of an observation $\underline{x}$ from the i^{th} component of Π_k to the density from the ℓ^{th} component of Π_k. In this case, the Bayes' rule is as follows:

Classify $\underline{x}$ into Π_k if

$$\frac{\pi_k \cdot \sum_{i=1}^{m} \alpha_i N(\underline{\mu}_i + \underline{\theta}_i^{(k)}, \sigma_i^2\Sigma)}{\pi_j \cdot \sum_{i=1}^{m} \alpha_i N(\underline{\mu}_i + \underline{\theta}_i^{(j)}, \sigma_i^2\Sigma)} \geq 1 \; ; \quad j=1, 2, \ldots, p.$$

or

$$\sum_{i=1}^{m} R_i^{(k)}(\underline{x}) \, [\lambda_{j,k}^{(i)}]^{-1} = \sum_{i=1}^{m} [1 + \sum_{\substack{\ell=1 \\ \ell \neq i}}^{m} \phi_{i,\ell}^{(k)}]^{-1} \, [\lambda_{j,k}^{(i)}]^{-1}$$

$$= \sum_{i=1}^{m} \left[\frac{\alpha_i \, N(\underline{\mu}_i + \underline{\theta}_i^{(k)}, \sigma_i^2\Sigma)}{\sum_{\ell=1}^{m} \alpha_\ell \cdot N(\underline{\mu}_i + \underline{\theta}_\ell^{(k)}, \sigma_i^2\Sigma)} \right] \cdot \left[\frac{\pi_k \alpha_k^{(i)} N(\underline{\mu}_i + \underline{\theta}_i^{(k)}, \sigma_i^2\Sigma)}{\pi_j \alpha_j^{(i)} \cdot N(\underline{\mu}_i + \underline{\theta}_1^{(j)} \sigma_i^2\Sigma)} \right]^{-1}$$

$$= \sum_{i=1}^{m} [1 + \sum_{\substack{\ell=1 \\ \ell \neq i}}^{m} \exp\{Q_{i,\ell}(\underline{x})\}]^{-1} \cdot [\exp\{L^{(j,k)}(\underline{x})\}] \leq 1,$$

where

classified into Π_k by the Bayes' rule when the component is identified but with at least one $\lambda_{j,k}^{(i)} < 1$, are not necessarily assigned to Π_k with the component information unknown.

3. Simulation Studies

A simulation experiment was conducted to investigate the robustness of the LDF with plug-in estimates under moderate non-normality. Section 3.1 describes the simulation experiment. The Bayes' rule and sample LDF errors are described using measures of non-normality in Section 3.2, while the difference in their classification performances is studied in Section 3.3. Lastly, the performances of the sample LDF using maximum likelihood estimates and Huber-type estimates are compared in Section 3.4.

3.1 The Simulation Experiment

The simulation experiment to investigate the robustness of the sample LDF to non-normality is based on the two-component mixed-normal distribution. The classification model studied was the canonical form of the distribution with proportional component covariance matrices. The result, due to Ashikaga [3] is

$$f_1(\underline{x}) = (1 - \alpha)\, N(\underline{0}, I) + \alpha N(\underline{\delta}, \sigma^2 I) \text{ in } \Pi_1 \text{ and}$$

$$f_2(\underline{x}) = (1 - \alpha)\, N(\underline{\mu}, I) + \alpha N(\underline{\mu}+\underline{\delta}, \sigma^2 I) \text{ in } \Pi_2,$$

where $0 < \alpha < 1$, $\sigma^2 > 1$, $\underline{\mu} = (\mu_1, 0, \ldots, 0)'$, and $\underline{\delta} = (\delta_1, \delta_2, 0, \ldots, 0)'$.

Table 2 lists the parametric configurations which were studied.

TABLE 2

Parameter Values Studied

$$\alpha = 0, .1, .2, .3, .3, .5, .6, .7, .8, .9$$

$$||\theta|| = \left(\sum_{i=1}^{p} \theta_i^2\right)^{\frac{1}{2}} = 0, 1, 2, 3$$

$$\sigma^2 = 1, 4, 9, 16$$

$$\Delta^2 = 1, 4$$

The LDF was studied when the parameters were replaced by maximum likelihood estimates as in Anderson [1] and by Huber-type estimates as in Randles and others [14].

The robustness criterion was the difference in misclassification errors between the LDF with plug-in parameter estimates and the Bayes' rule with parameters known. The LDF misclassification errors were computed from 100 repetitions of the following scheme:

(1) Draw training samples of size n_1 from Π_1 and n_2 from Π_2 and compute the LDF.

(2) Draw an index sample of size 50 from Π_1 and size 50 from Π_2. Classify the index samples and compute the average misclassification probability.

The Bayes' rule errors were also computed using Monte Carlo procedures due to the difficulty of the numerical computation.

The misclassification errors were indexed by Mahalanobis distance between populations Δ^2 and measures of non-normality. Two measures introduced by Malakovich and Afifi [12] and studied by Ashikaga [3] were multivariate skewness

$$\beta_1^* = \max_{\underline{c}} \{\beta_1(\underline{c}'\underline{x})\}$$

and multivariate kurtosis

$$\beta_2^* = \max_{\underline{c}} \{[\beta_2(\underline{c}'\underline{x}) - 3]^2\} ,$$

where β_1 and β_2 are the univariate skewness and kurtosis measures.

3.2 Probabilities of Misclassification vs. β_1^* and β_2^*

Prior to looking at various plots of differences in misclassification errors between the Bayes' rule and sample LDF classifiers, it is helpful to consider the relationship between the actual level of misclassification error and indictors of non-normality.

The overall Bayes' misclassification errors are plotted against the skewness measure β_1^* in Figure 3.1 for $\Delta^2 = 1$ and Figure 3.2 for $\Delta^2 = 4$. For the particular mixed-normal pdf's under study the largest errors occured when the pdf was symmetrical. The maximum errors decrease as the skewness β_1^* rises to moderate values (3 to 4). Representative graphs of the overall misclassification error for the class of LDF's with plug-in parameter estimates are given in Figures 3.3 and 3.4 for $\Delta^2 = 1$ and $\Delta^2 = 4$, respectively. Here the LDF was estimated by Huber-type estimators as in Randles and others [14]. The training samples had 25 observations from each population. These graphs are similar to the plots of the Bayes' error, except that the maximum errors were approximately two percent larger than the Bayes' errors at $\Delta^2 = 1$, but only one percent larger at $\Delta^2 = 4$. While the graphs for the Bayes' errors and the sample LDF are similar for the largest errors at various levels of

two classifiers, but for training samples of size 25 from each population. No differences are noted from the situation with smaller training samples.

Figures 3.13 through 3.16 plot the differences between the sample LDF and Bayes' classification errors when the parameters of the LDF were replaced by Huber-type estimators. Once more there is a drop in the difference between the errors from approximately 9 percent at $\Delta^2 = 1$ to less than 2.5 percent at $\Delta^2 = 4$. As with the maximum likelihood estimated LDF, no relationship was shown between skewness and the difference in overall errors.

The differences between the sample LDF with maximum likelihood estimators and the Bayes' rule errors vs β_2^* are plotted in Figures 3.17 through 3.20. There appears to be a decrease in the largest differences for higher values of kurtosis when $\Delta^2 = 1$. These differences are much smaller when $\Delta^2 = 4$. It has been previously shown by Ashikaga [3] that the LDF is the Bayes' rule for scale-contaminated mixed-normal models. Thus, for this sub-class of mixed-normal models, the effects of kurtosis on the differences between the sample LDF and the Bayes' classifier errors present themselves solely through the plug-in parameter estimators. For training sample sizes of 25 from each population those models with only scale-contamination exhibited under two percent difference between the errors of the two procedures at $\Delta^2 = 1$. For the entire group of mixed-normal models studied, the difference between these two errors ranged up to 9 percent when kurtosis was 0.

3.4 Sample LDF Misclassification Errors vs. β_1^* and β_2^*

Figures 3.21 and 3.22 plot the differences between overall misclassification errors when the parameters of the LDF are replaced by Huber-type estimators and maximum likelihood estimators versus β_1^*. Here we have training samples of size 25 from each population. We see that the largest differences between these two plug-in schemes decrease from approximately two percent for $\Delta^2 = 1$ to 1.25 percent at $\Delta^2 = 4$. For $\Delta^2 = 1$ the largest differences in the two errors seem to shrink as β_1^* increases but is based on relatively few pdf's with moderate skewness. Similar results are obtained for β_2^* in figures 3.23 and 3.24. In the sub-class of scale-contaminated distributions, the difference in these two error rates was under 0.5 percent for $\Delta^2 = 1$ and 0.3 percent for $\Delta^2 = 4$.

skewness, these plots do not reveal relationships between the Bayes' errors and sample LDF errors for particular distributions. We will need to examine the actual differences between the two classification schemes for particular distributions in order to study the robustness of the sample LDF. The overall misclassification errors were also plotted against the kurtosis coefficient β_2^*. Figures 3.5 and 3.6 graph the Bayes' error against β_2^* and Figures 3.7 and 3.8 the sample LDF with Huber-type estimates against β_2^*. The largest errors occur when β_2^* is near zero and decrease quickly for β_2^* greater than five.

A drawback of Malakovich and Afifi's [12] multivariate kurtosis measure is that the linear combination of $\underline{x}$ with univariate kurtosis most different from 3, the value of β_2^* for a univariate normal distribution, can correspond to either a flat or peaked distribution. Reinspecting Figures 3.5 and 3.6, the points with largest misclassification errors (circled) correspond to platykurtic or normal pdfs.

3.3 Differences between the Sample LDF and Bayes' Errors for β_1^* and β_2^*.

Figures 3.9 and 3.10 plot the differences between the errors for the sample LDF and the Bayes' classifiers (P(Sample LDF) - P(Bayes)). Here the LDF was estimated by maximum likelihood from training samples of size 15 from each population. The maximum differences between misclassification errors were approximately 9 percent for $\Delta^2 = 1$ and dropped to less than 2.5 percent for $\Delta^2 = 4$. Neither graph indicated any relationship between the skewness coefficient and the difference in errors. Figures 3.11 and 3.12 also plot the differences between errors for these

Figure 3.1

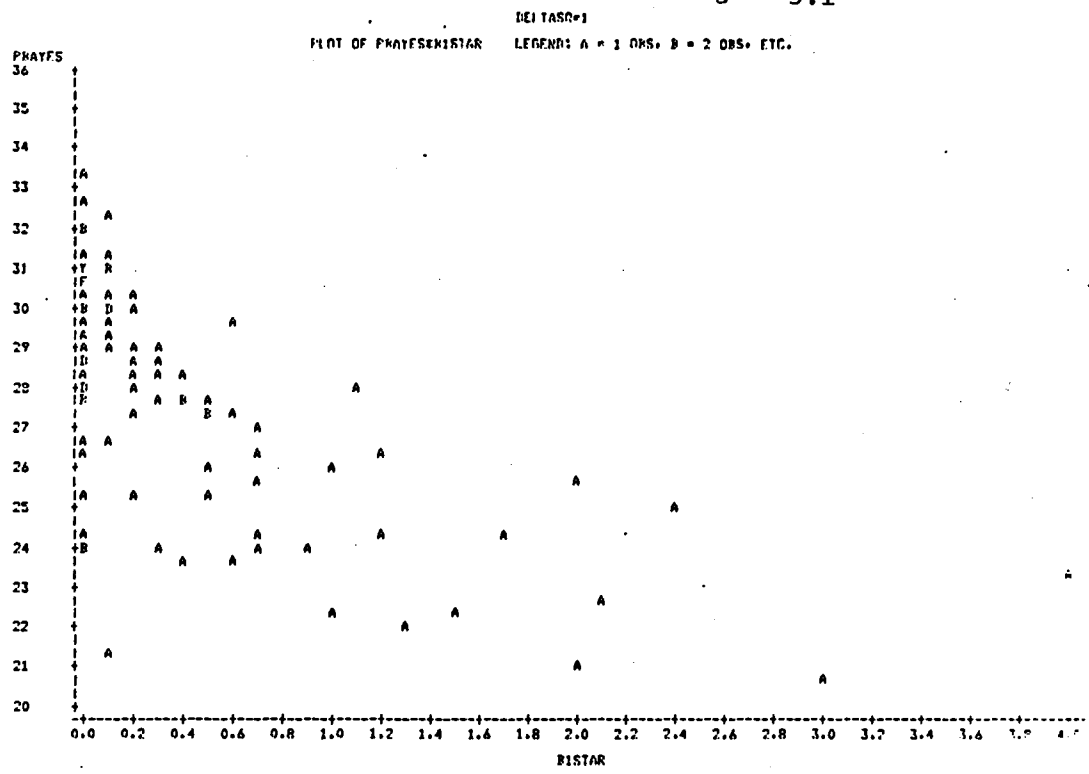


Figure 3.2

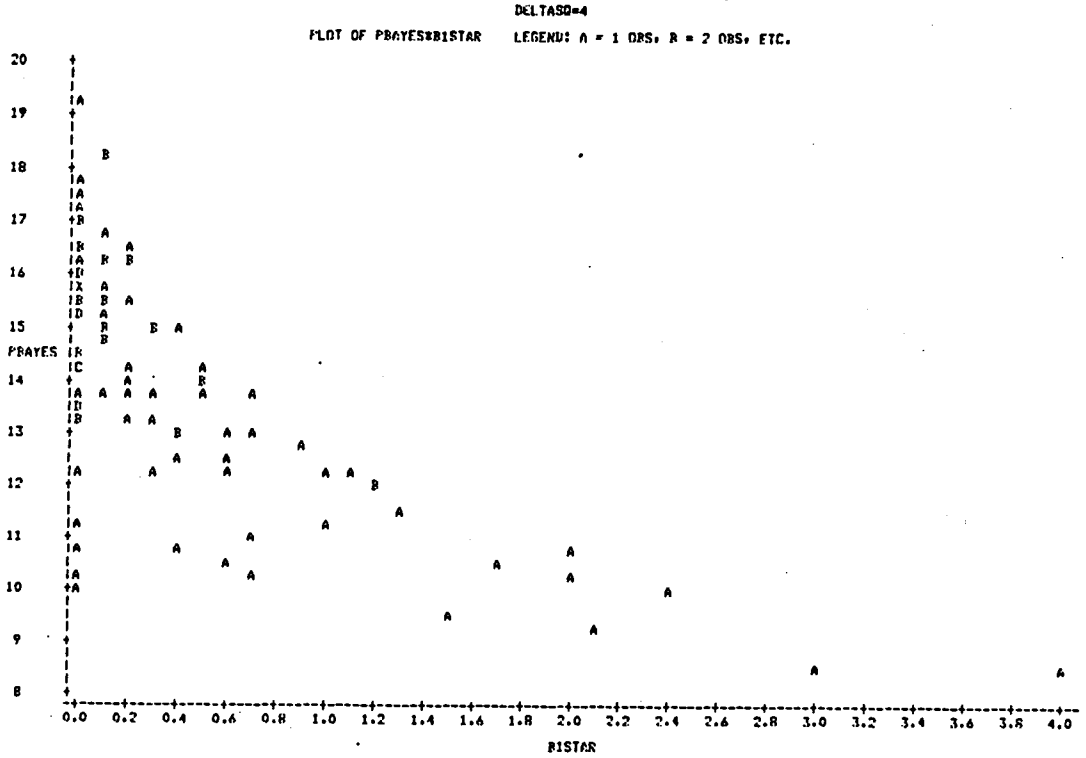


Figure 3.3

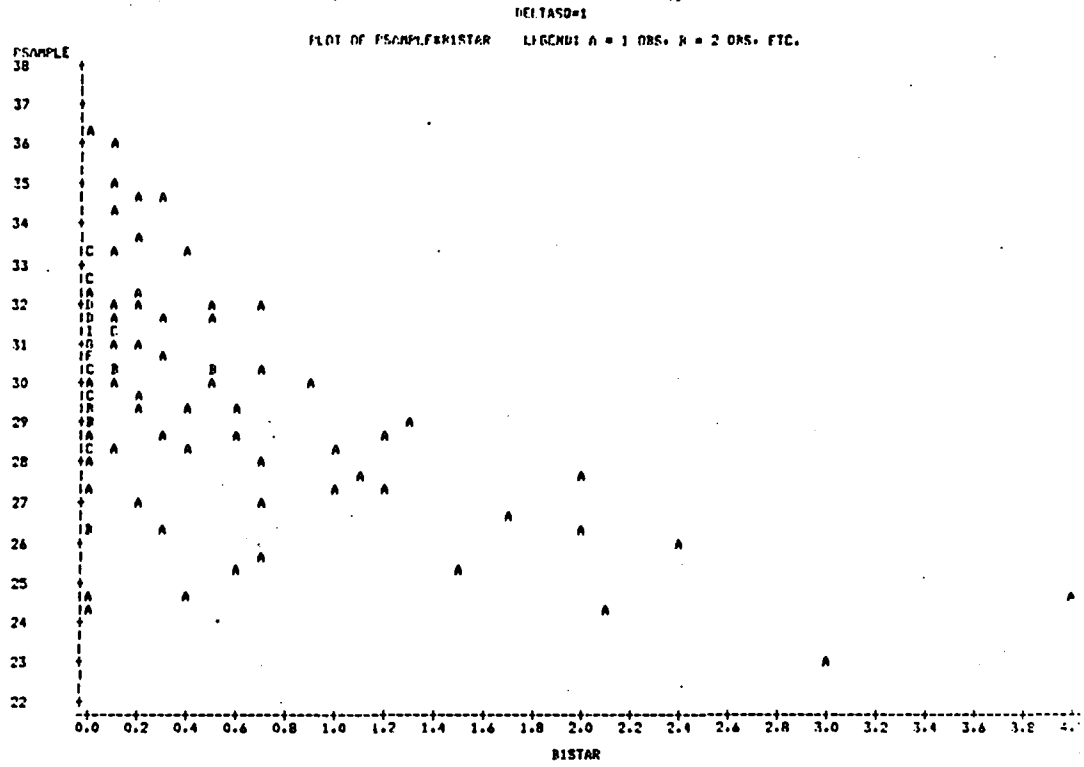


Figure 3.4

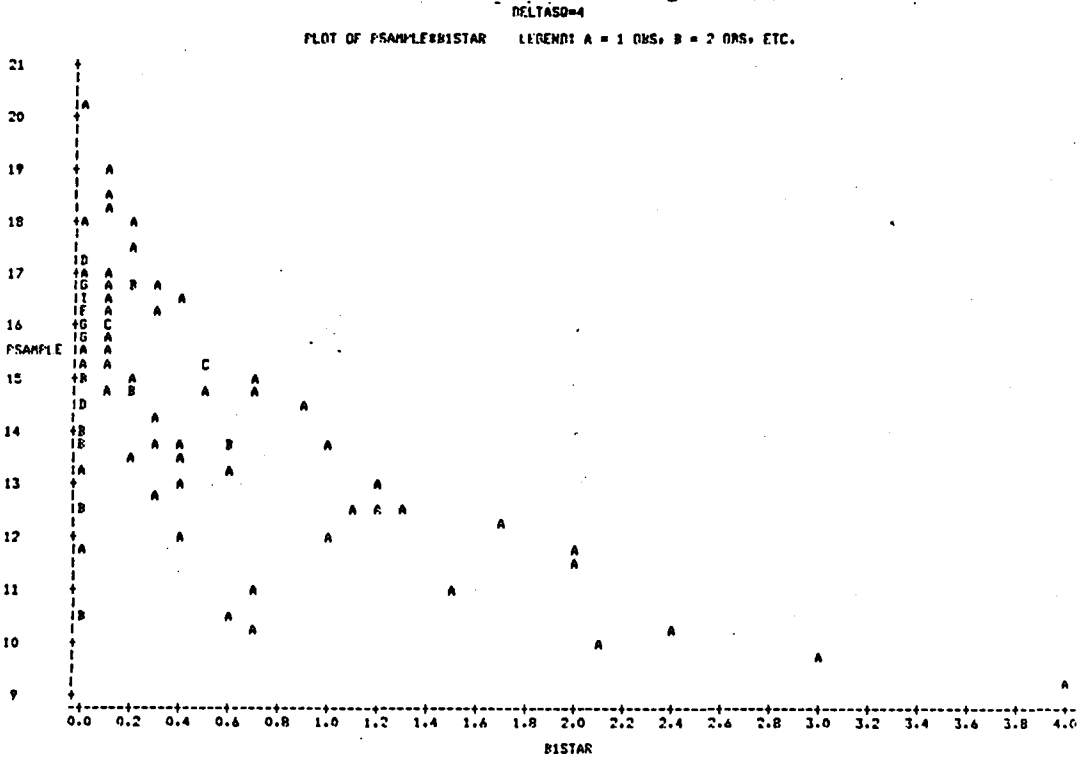


Figure 3.5

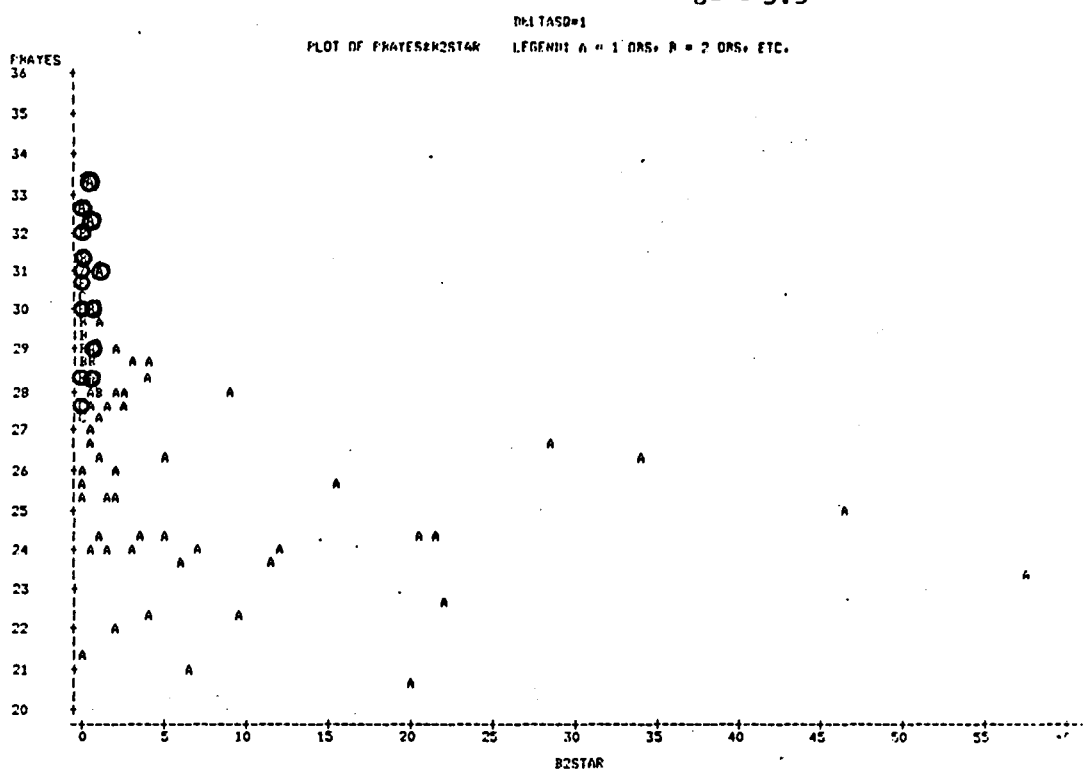


Figure 3.6

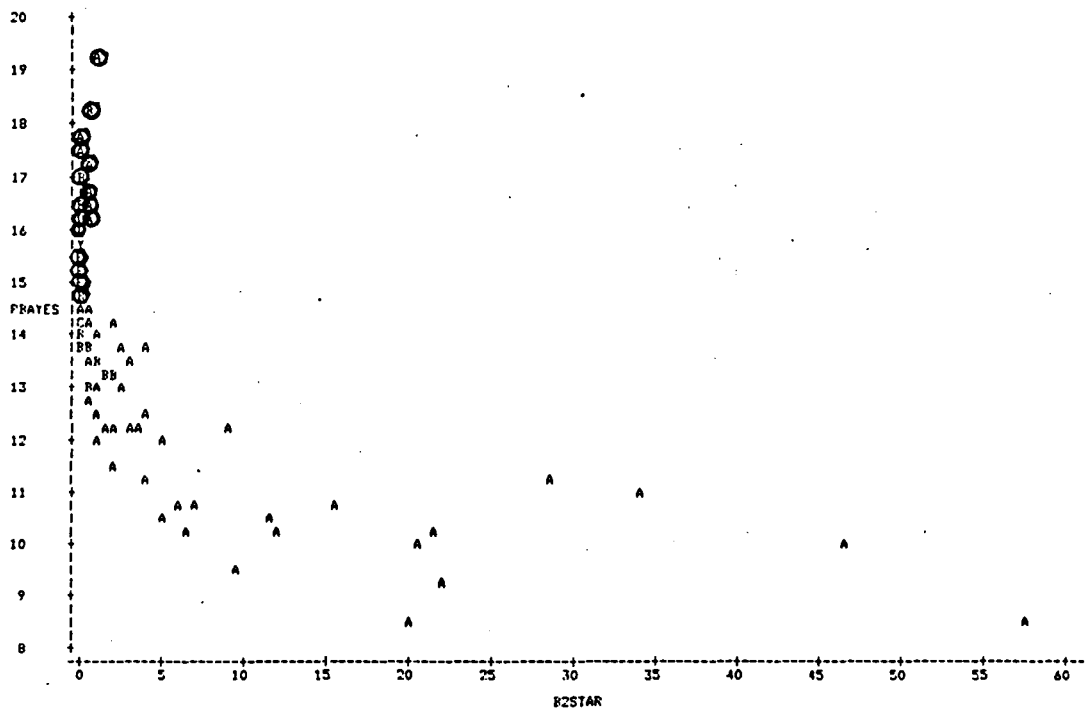


Figure 3.7

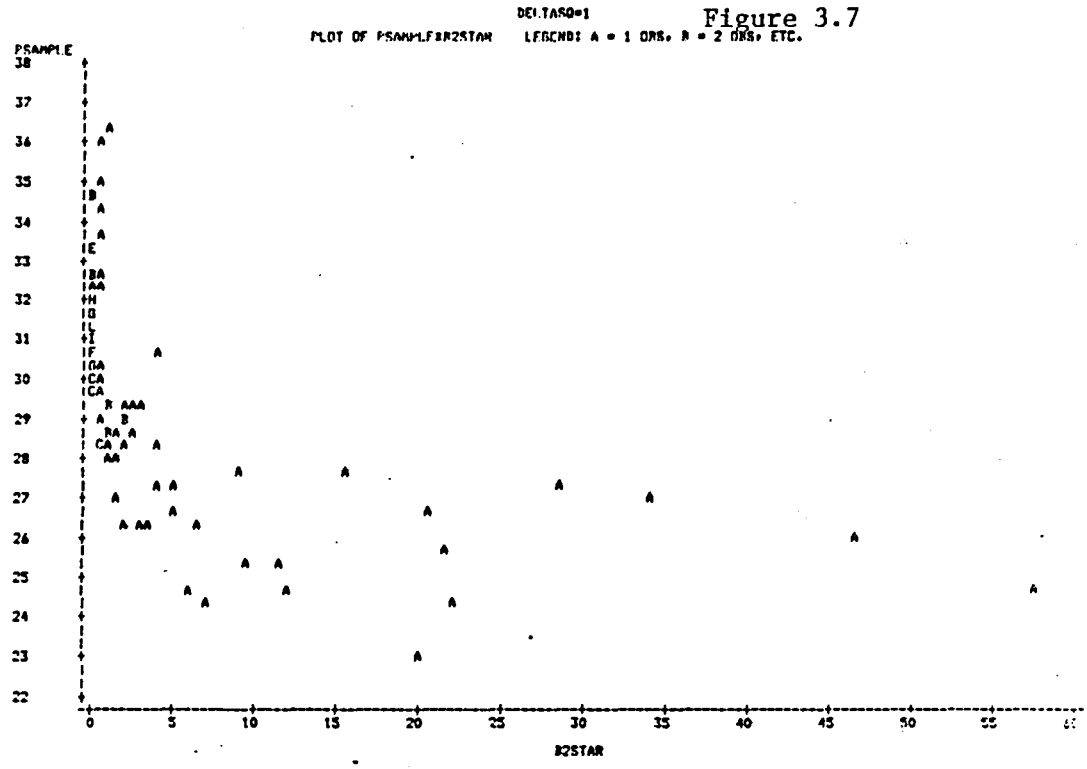


Figure 3.8

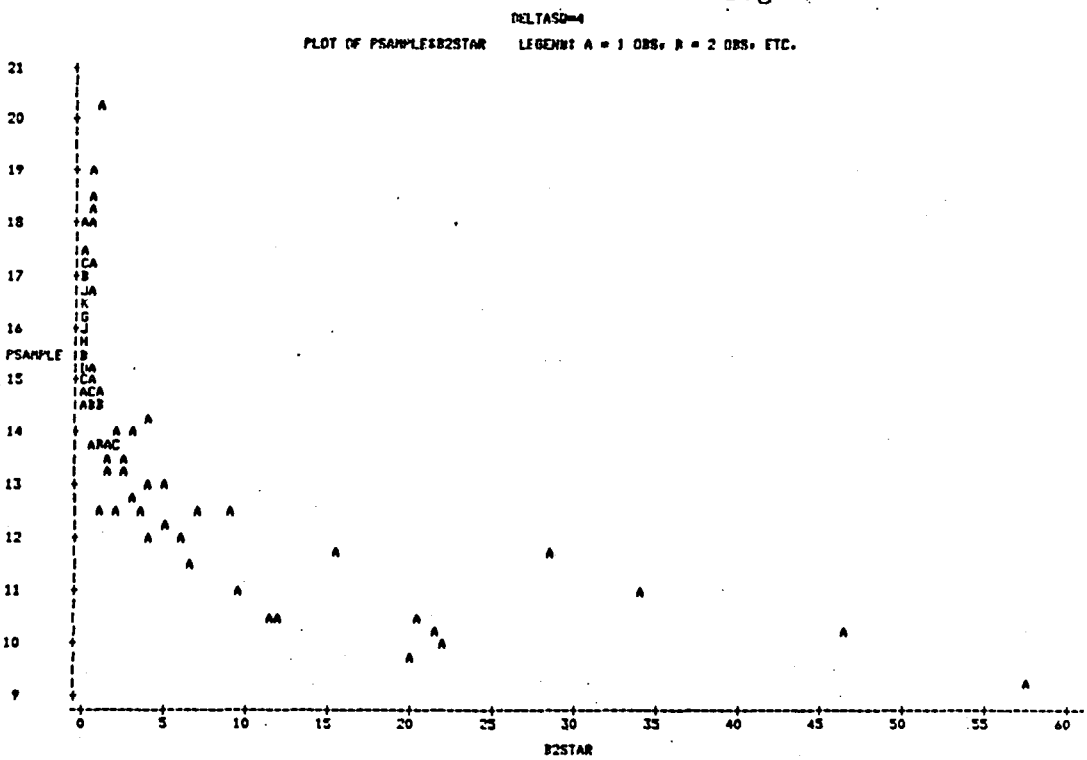


Figure 3.9

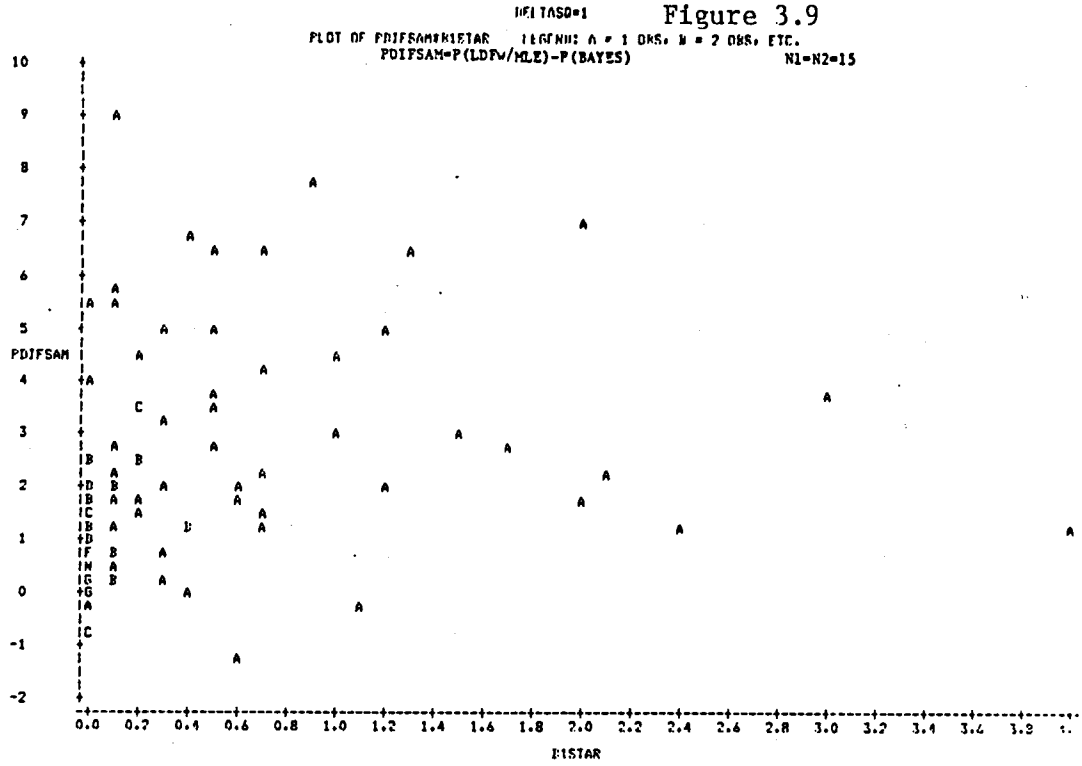
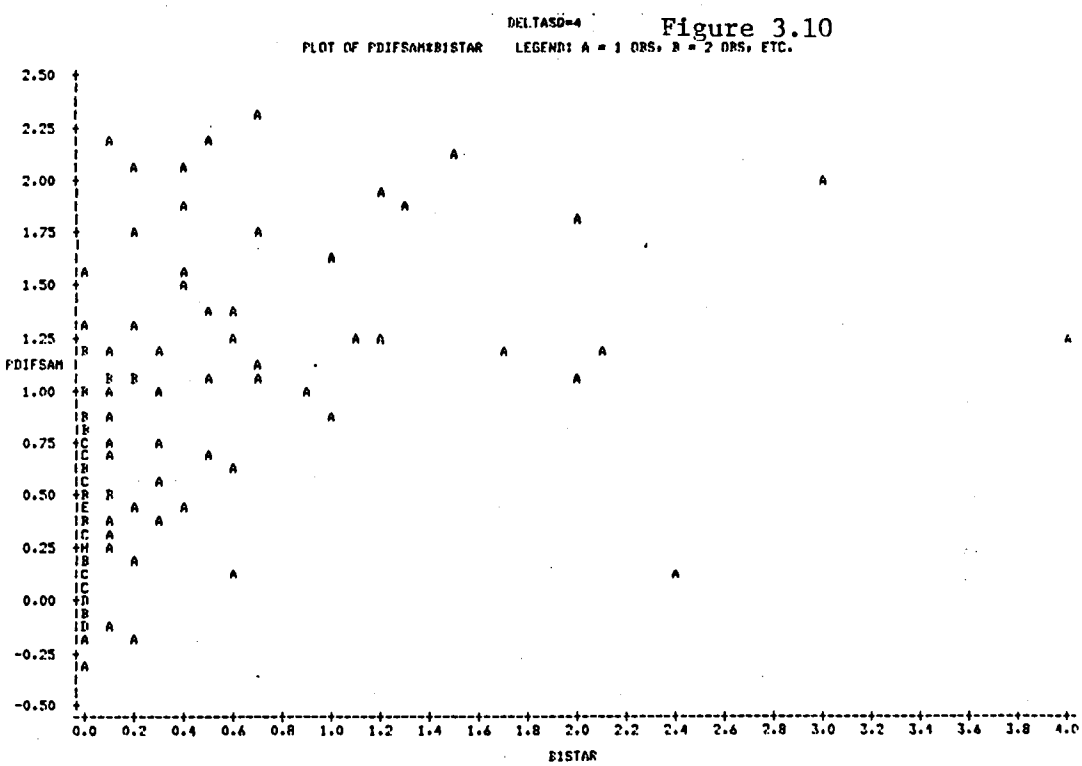


Figure 3.10



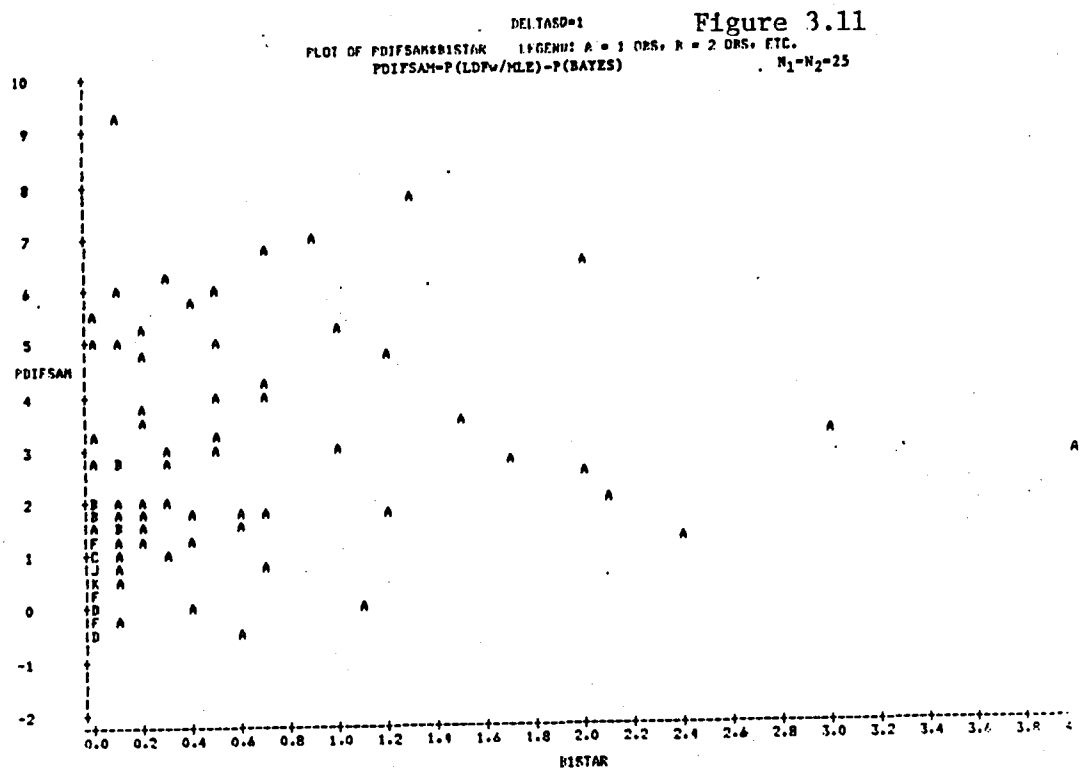


Figure 3.11

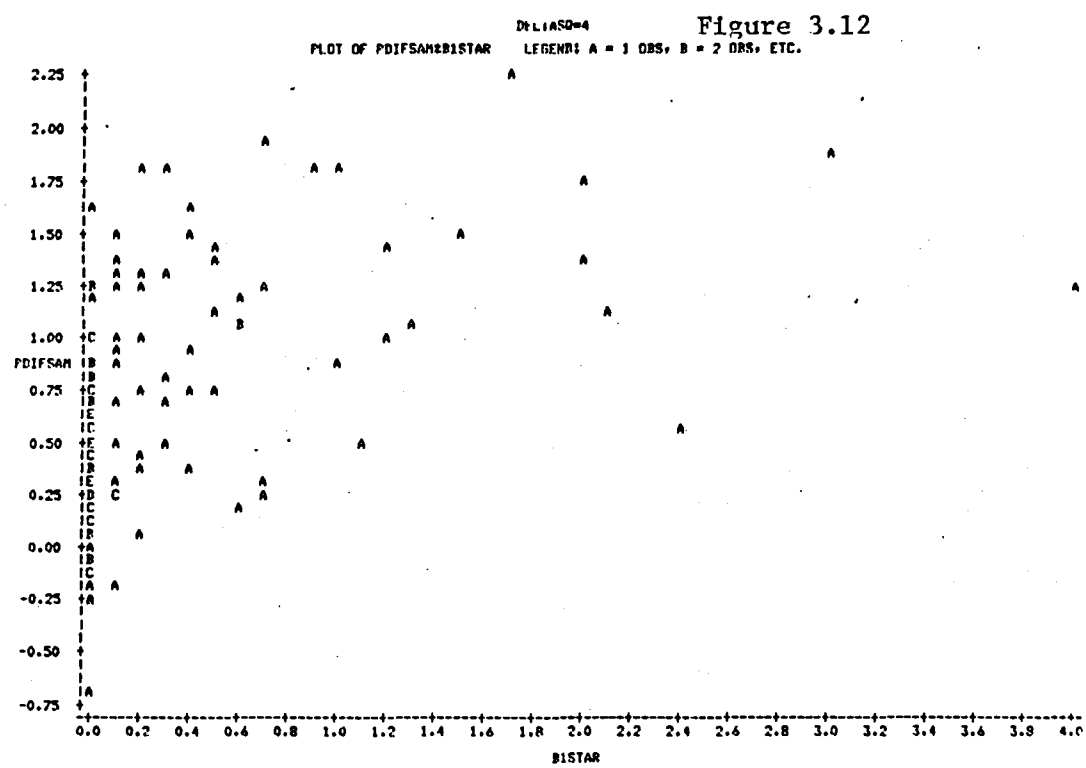


Figure 3.12

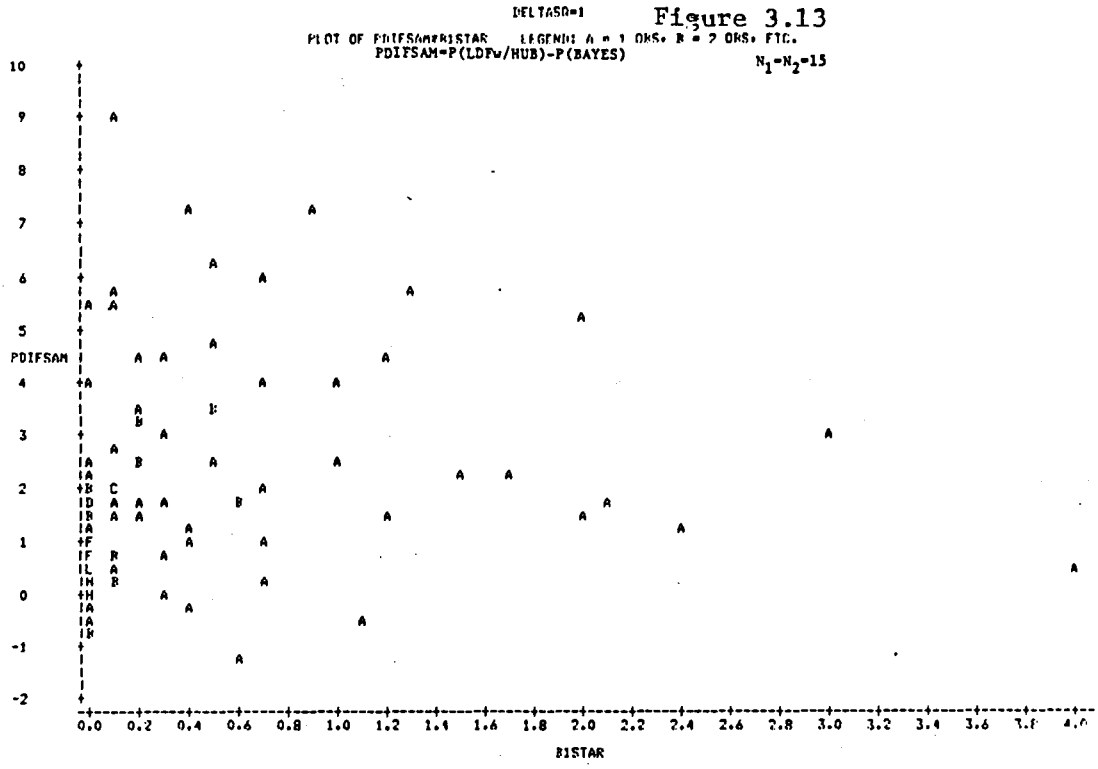


Figure 3.13

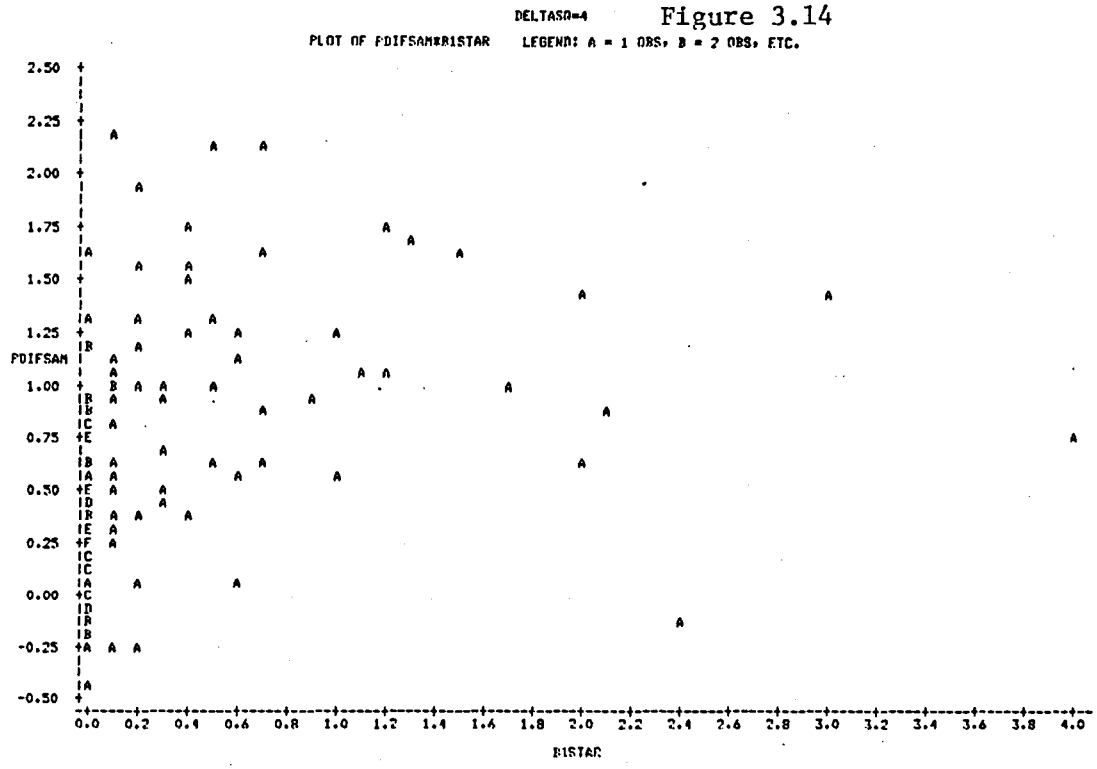


Figure 3.14

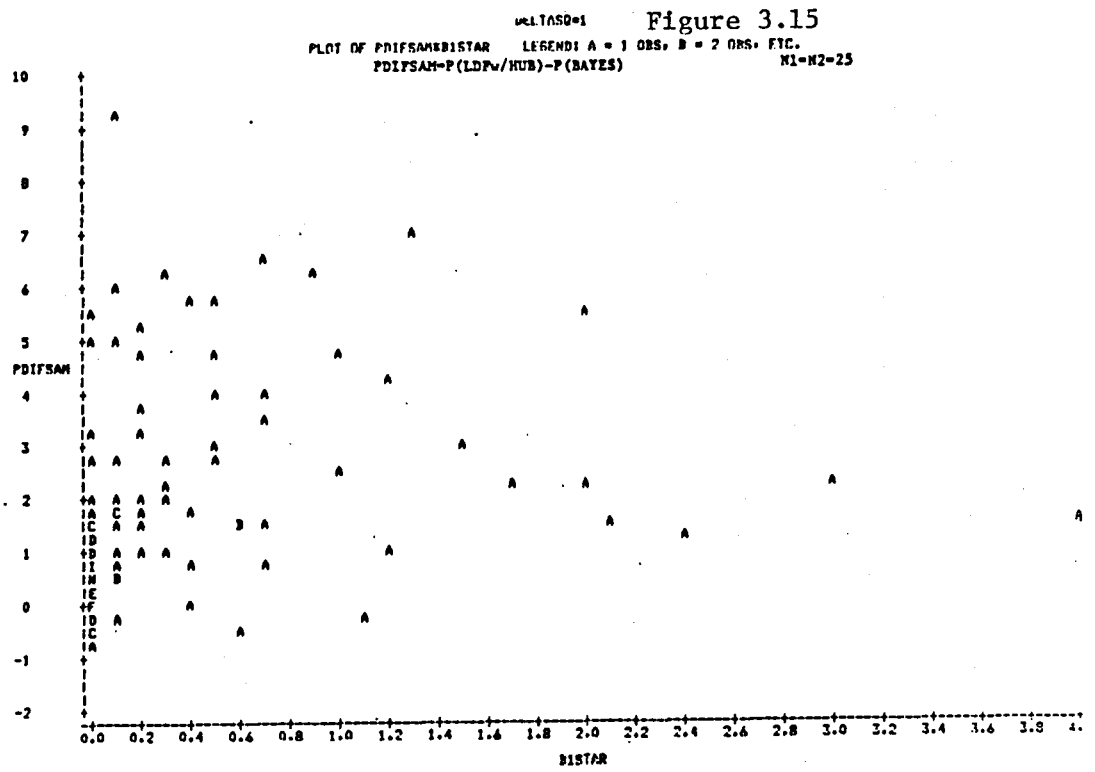


Figure 3.15

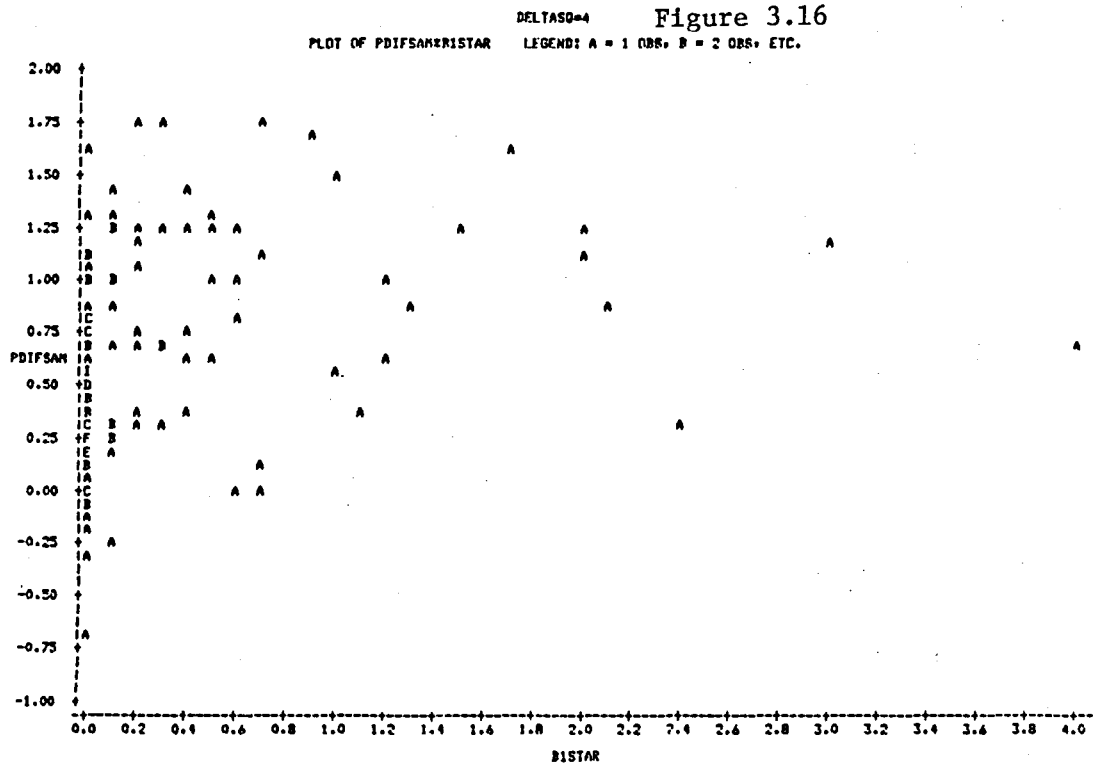


Figure 3.16

Figure 3.17

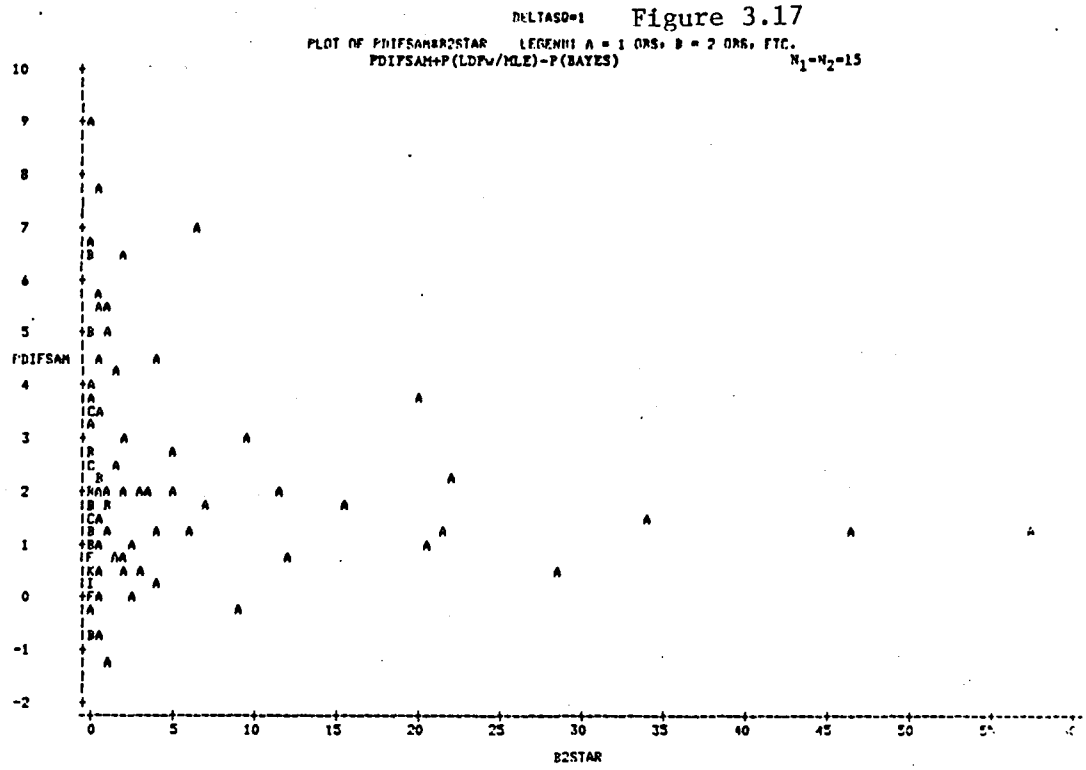


Figure 3.18

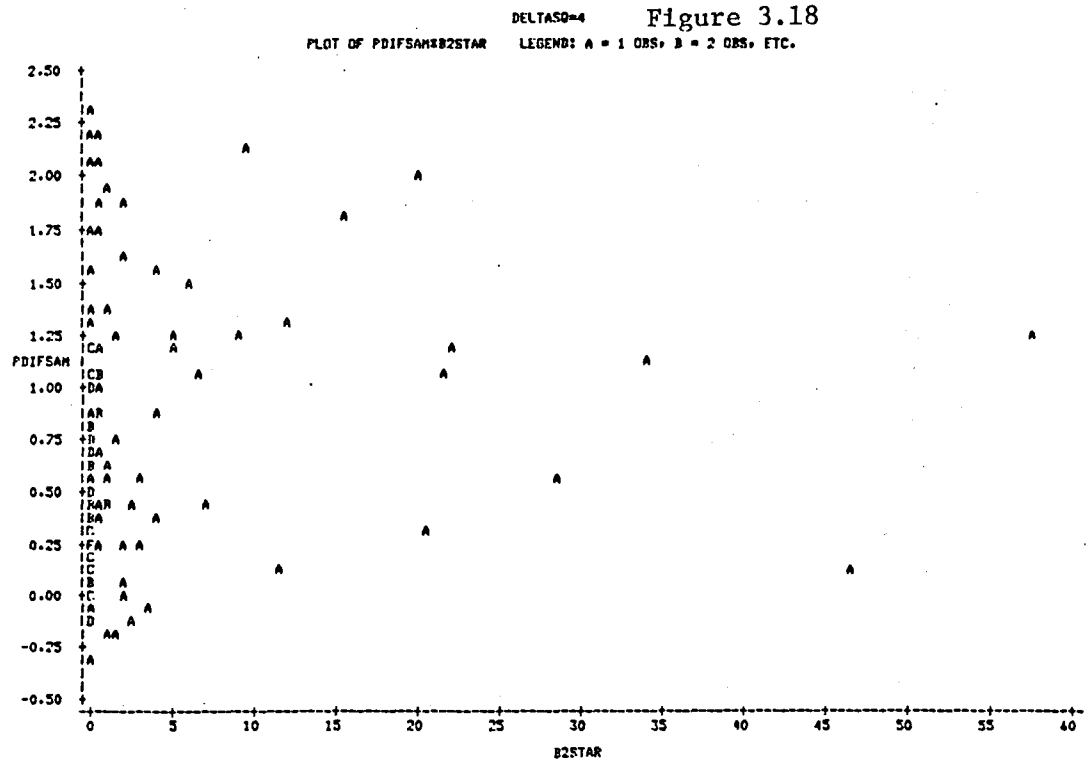


Figure 3.19

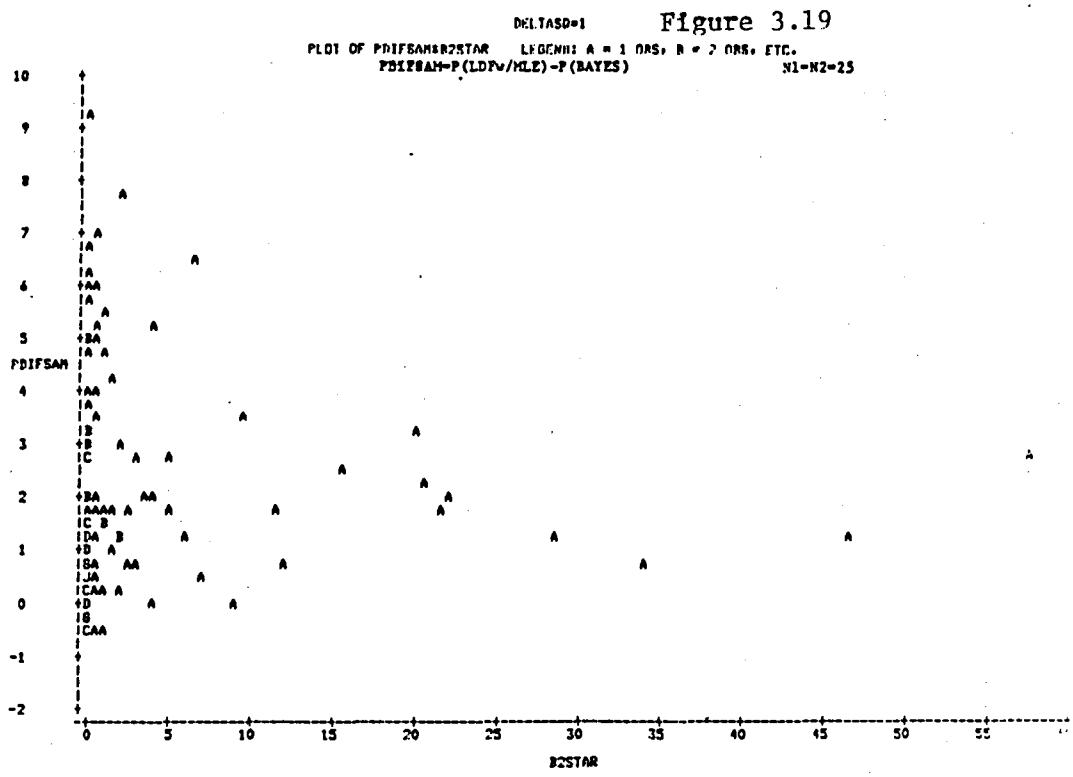
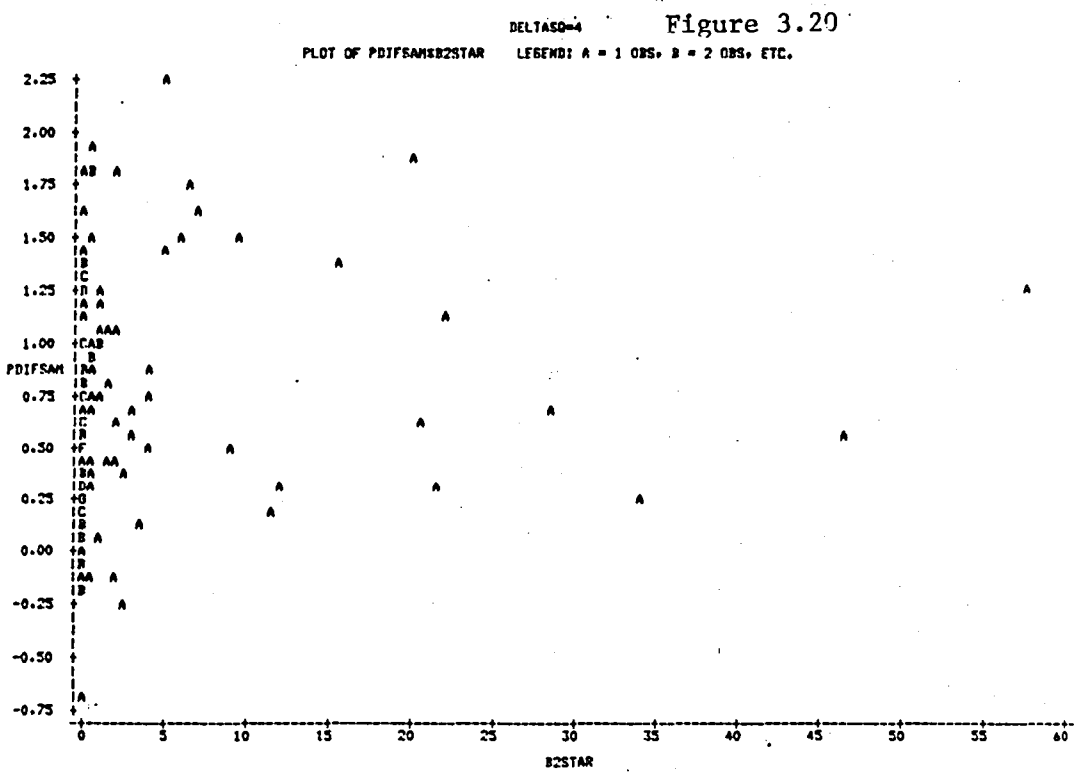


Figure 3.20



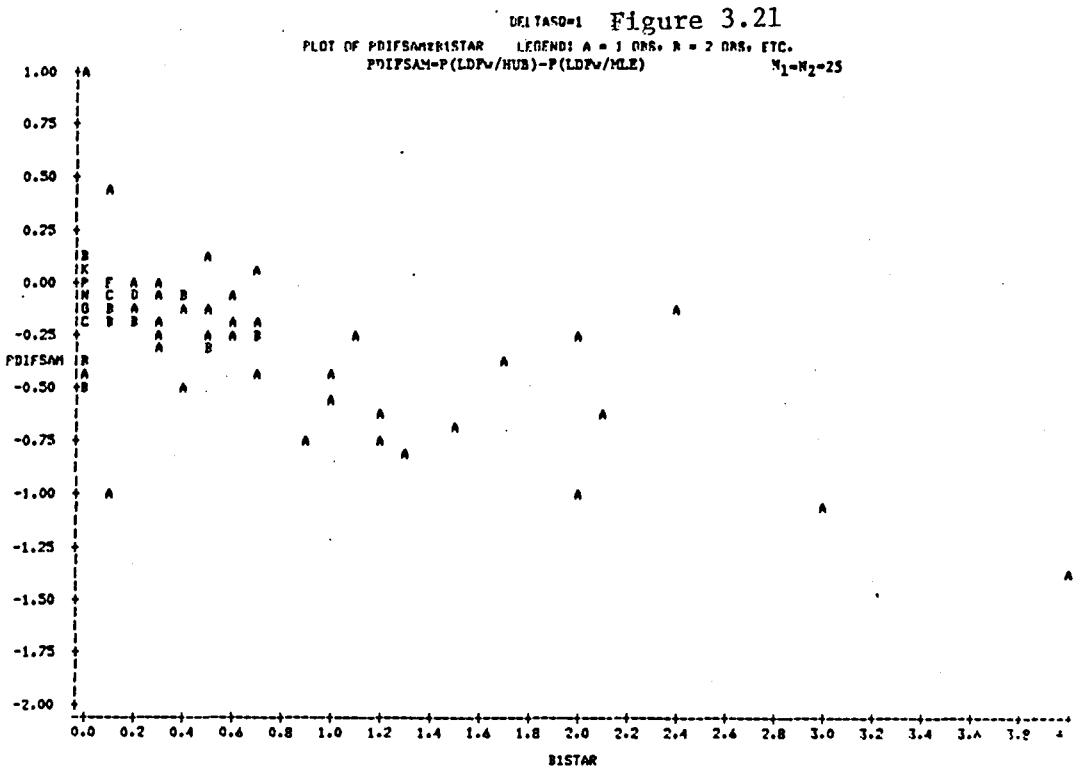


Figure 3.21

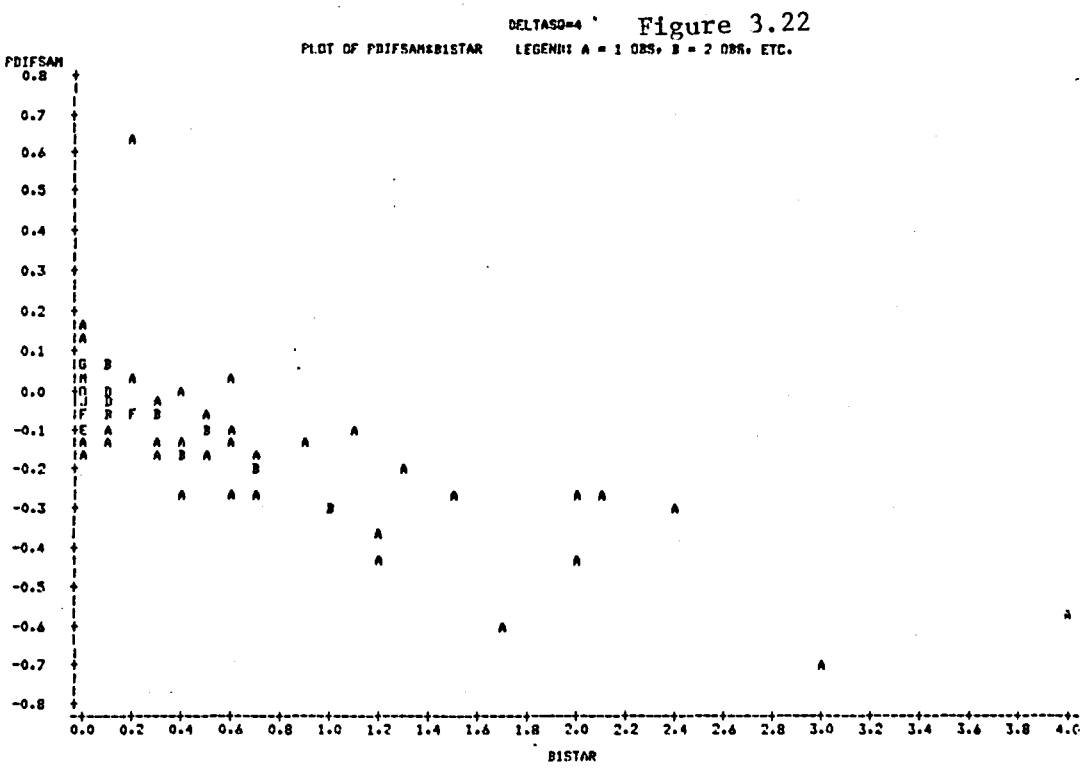


Figure 3.22

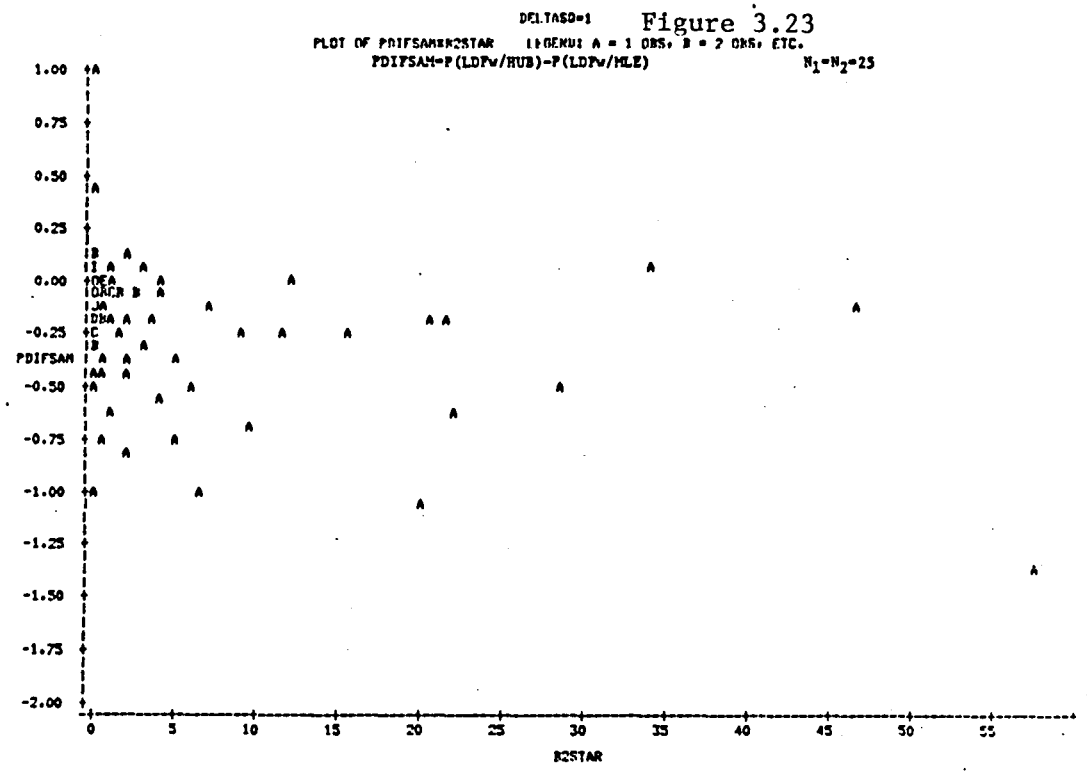


Figure 3.23

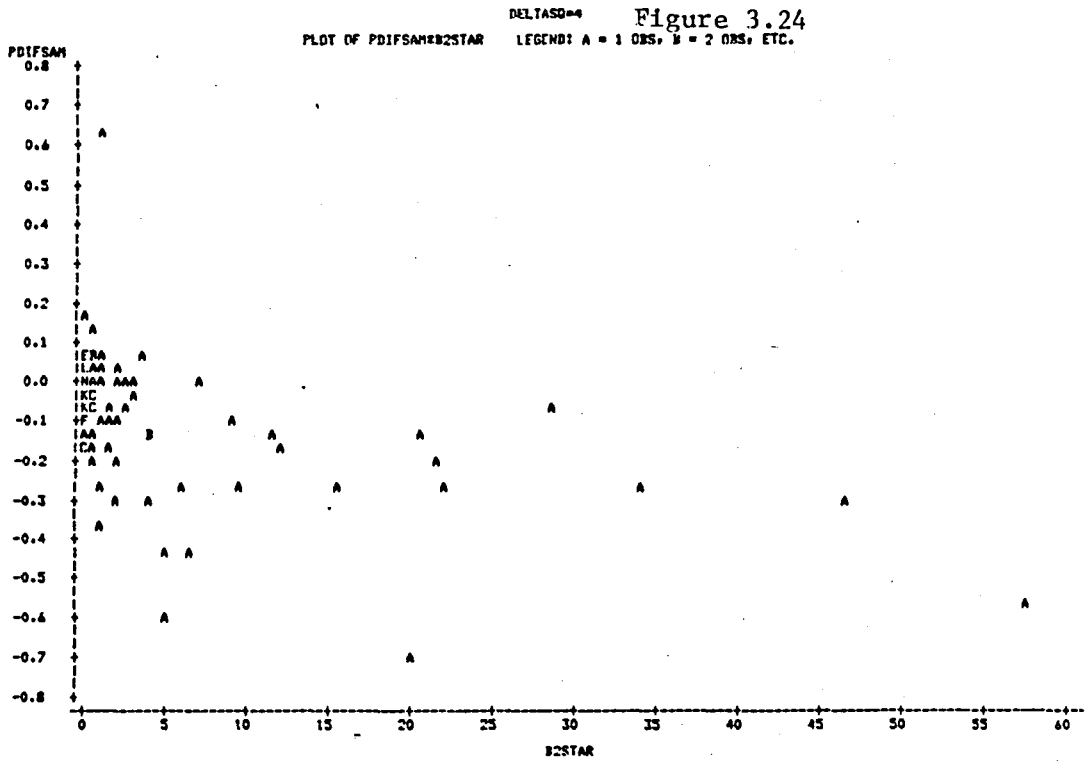


Figure 3.24

References

[1] Anderson, T. W., Introduction to Multivariate Statistical Analysis, (John Wiley and Sons, New York, 1958).

[2] Ashikaga, T. and Chang, P. C., Robustness of Fisher's Linear Discriminant Function Under Two-Component Mixed Normal Models, Jrnl. of Amer. Stat. Assn. 64 (1981) 676-680.

[3] Ashikaga, T., Robustness of the Linear Discriminant Function for a Mixed Normal Model, Ph.D. dissertation, University of California, Los Angeles (1973).

[4] Chang, P. C. and Afifi, A. A., Classification Based on Dichotomous and Continuous Variables, Jrnl. of Amer. Stat. Assn. 69 (1974) 336-339.

[5] Dunn, O. J. and Holloway, L. N., The Robustness of Hotelling's T^2, Jrnl. of Amer. Stat. Assn. 64 (1967) 1399-1412.

[6] Fisher, R. A.,The Use of Multiple Measurements in Taxonomic Problems, Ann. of Eugenics 7 (1939) 179-188.

[7] Gilbert, E. S.,The Effect of Unequal Variance-Covariance Matrices on Fisher's LDF, Biometrics 25 (1969) 505-516.

[8] Hoel, P. G. and Peterson, R. P., A Solution to the Problem of Optimal Classification, Ann. of Math. Stat. 20 (1949) 433-438.

[9] Hosmer, Jr. D. W. and Dick, N. P., Information and Mixtures of Two Normal Distributions, Jrnl. of Stat. Comp. and Simul. 6 (1977) 137-148.

[10] Johnson, N. L., Systems of Frequency Curves Generated by Methods of Translation, Biometrika 36 (1949) 149-176.

[11] Lachenbruch, P. A., Sneeringer, C., and Revo, L. T., Robustness of the Linear and Quadratic Discriminant Functions to Certain Types of Non-Normality, Comm. in Stat. 1 (1973) 39-56.

[12] Malkovich, J. R., and Afifi, A. A., On Tests of Multivariate Normality, Jrnl. of Amer. Stat. Assn. 68 (1973) 176-179.

[13] Marks, S. and Dunn, O. J., Discriminant Functions When Covariance Matrices are Unequal, Jrnl. of Amer. Stat. Assn. 69 (1974) 555-559.

[14] Randles, R. H., Broffitt, J. D., Ramberg, J. S., and Hogg, R. V., Generalized Linear and Quadratic Discriminant Functions Using Robust Estimates, Jrnl. of Amer. Stat. Assn. 73 (1978) 564-568.

[15] Wald, A., On a Statistical Problem Arising in the Classification on an Individual into Two Groups, Ann. of Math. Stat. 15 (1974) 145-163.

[16] Welch, B. L., Note on Discriminant Functions, Biometrika 31 (1939) 218-220.

REPEATED-MEASURES ANALYSIS OF IMAGE DATA

H. J. Newton
Texas A&M University

ABSTRACT

It is suggested that using a modified analysis of variance procedure on data sampled systematically from a rectangular array of image data can provide a measure of homogeneity of means over that array in single directions and how variation in perpendicular directions interact. The modification of analysis of variance required to account for spatial correlation is described theoretically and numerically on simulated data.

1. Introduction

Incorporating spatial correlation into the analysis of multivariate image data observed in the plane leads to massive data management and computational problems. In this paper we describe an initial attempt to answer problems in the plane by sampling the data in parallel transects so that one need only consider correlation in one direction. Thus given a (K x T) array of d-dimensional observations, divide the K rows into g groups and select n_i rows for the i^{th} group so that rows within a group are essentially uncorrelated. Then the correlation within rows can be modeled using ordinary time series techniques and can be incorporated in an analysis of variance procedure in analogy with that for long repeated measures designs.

Let $\underset{\sim}{y}_{ik} = \left(\underset{\sim}{y}_{ijk}^T, \ldots, \underset{\sim}{y}_{iTk}^T\right)^T$ be a (Td x 1) random vector representing the T d-dimensional vectors for the kth observation in the ith group of observations, k=1,...,n_i, i=1,...,g. Assume

$$(1) \qquad \underset{\sim}{y}_{ijk} = \underset{\sim}{u}_{ij} + \underset{\sim}{\eta}_{ijk}$$

where the $\underset{\sim}{\eta}$'s are zero mean random vectors which are uncorrelated for different i's and/or k's but $\underset{\sim}{\eta}_{ijk}$'s having the same j are correlated. Thus in (196 x 117) 4-dimensional image data one might let g be between 3 and 5 and the n_i's be 4 or 5. In this paper then we visualize analyzing the means of small number of groups of time series (here the "time" index j represents position within a row, i.e the

East-West location of an observation).

In Section 2 we consider the univariate case, i.e using data on only one channel or some function of four channels at each location. Then in Section 3 we discuss possible extensions to the general d-dimensional case.

2. Univariate Long Repeated Measures Analysis

When d is one, equation (1) appears to be describing a two-factor analysis of variance model with the factors being group number and time (i.e column index). Such data is often called repeated measures data since, because of the correlation, one can think of y_{ik} as containing repeated measurements on the same experimental unit.

There are three basic hypotheses one is interested in testing; 1) equality of group means averaged over time (denoted H_G), 2) equality of time means averaged over group (H_T), and 3) no interaction between group and time means (denoted H_{GT}), i.e the graphs of the group means over time are "parallel". In analyzing image data we visualize using the test of H_G to measure homogeneity in the North-South direction, H_T to measure homogeneity East-West, and H_{GT} to measure whether variability in the North-South direction is constant over East-West location. Also, arrays at varying locations can be fairly quickly classified using such a procedure. In Table 1 we list the statistics used to test these hypotheses and their null distributions in the case of no correlation within rows. We then describe how these tests can be modified to account for correlation.

Table 1. The Usual Two Factor ANOVA

Hypothesis	Test Statistic	Null Distribution
H_G	$F_G = \frac{SS_{G/(g-1)}}{SSE_{1/(N-g)}} = \frac{MS_G}{MSE_1}$	$F_{g-1,\ N-g}$
H_T	$F_T = \frac{SS_{T/(T-1)}}{SSE_{2/(T-1)(N-g)}} = \frac{MS_T}{MSE_2}$	$F_{T-1,(T-1)(N-g)}$
H_{GT}	$F_{GT} = \frac{SS_{GT/(T-1)(g-1)}}{SSE_{2/(T-1)(N-g)}} = \frac{MS_{GT}}{MSE_2}$	$F_{(T-1)(g-1),(T-1)(N-g)}$

$$N = \sum_{i=1}^{g} n_i, \text{ e.g. } \bar{y}_{.j.} = \sum_{i=1}^{g} \sum_{k=1}^{n_i} y_{ijk/N}$$

$$SS_G = \sum_{i=1}^{g} n_i \, (\bar{y}_{i..} - \bar{y}...)^2, \quad SSE_1 = \sum_{i=1}^{g} \sum_{k=1}^{n_i} (\bar{y}_{i.k} - \bar{y}_{i..})^2 \ ,$$

$$SS_T = \sum_{j=1}^{T} N(\bar{y}_{.j.} - \bar{y}...)^2, \quad SSE_2 = \sum_{i=1}^{g} \sum_{j=1}^{T} \sum_{k=1}^{n_i} (y_{ijk} - \bar{y}_{ij.} - \bar{y}_{i.k} + \bar{y}_{i..})^2$$

$$SS_{GT} = \sum_{i=1}^{g} \sum_{j=1}^{T} n_i \, (\bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + \bar{y}...)^2$$

To incorporate correlation into the analysis, we let Σ_{ik} be the (T x T) covariance matrix of $\underset{\sim}{y}_{ik}$. In this paper, we shall assume that $\Sigma_{ik} = \Sigma$ for all i and k. Thus we are assuming that the $\underset{\sim}{y}_{ik}$ are independent $N_T(\underset{\sim}{u}_i, \Sigma)$ random variables where $\underset{\sim}{u}_i^T = (u_{i1}, \ldots, u_{iT})$. The following theorem indicates how the analysis can be modified when

$\ddagger \neq \sigma^2 I_T$.

Theorem 1 (Geisser and Greenhouse [2])

a) The null distribution of F_G is unaffected by correlation.

b) The null distributions of F_T and F_{GT} are approximately

$$F_T \sim F_{\varepsilon(T-1),\varepsilon(T-1)(N-g)} \quad , \quad F_{GT} \sim F_{\varepsilon(T-1)(g-1),\varepsilon(T-1)(N-g)}$$

where the degrees of freedom reduction factor ε is given by

$$\varepsilon = \frac{[tr(A\ddagger)]^2}{(T-1)tr(A\ddagger A\ddagger)} \quad ,$$

where $A = I_T - \frac{1}{T} 1_T 1_T^T$ and 1_T is a T-vector of ones.

c) A lower bound for ε is $\varepsilon \geq \frac{1}{T-1}$ and thus conservative $(1-\alpha)$ level tests for H_T and H_{GT} are to compare F_T and F_{GT} to $F_{\alpha,1,N-g}$ and $F_{\alpha,g-1,N-g}$ respectively.

Note that ε can be written as

$$\varepsilon = \frac{\left[\sum_{i=1}^{T-1} \lambda_i(A\ddagger)\right]^2}{(T-1)\sum_{i=1}^{T-1} \lambda_i^2(A\ddagger)} \tag{2}$$

where $\lambda_1(A\ddagger) \geq \ldots \geq \lambda_{T-1}(A\ddagger)$ are the T-1 nonzero eigenvalues of the rank T-1 matrix $A\ddagger$. Thus from (2) it is easy to see that $\varepsilon = 1$ (and using the F tests with no degrees of freedom reduction for correlation are correct) if and only if all the eigenvalues of $A\ddagger$ are the same.

The results above are for a general, symmetric, positive definite matrix $\ddagger$. It seems clear in the image data problem that it is

reasonable to assume that $\sharp$ is Toeplitz, i.e

$$\sharp = \text{Toepl}\ (\sigma(0), \sigma(1),\ldots,\sigma(T-1)),$$

i.e the (j,k)th element of $\sharp$ is a number $\sigma(|j-k|)$. Thus we are assuming that for each i,k, $n_{i1k},\ldots,n_{iTk}$ is a sample realization from a covariance stationary time series having autocovariance function $\sigma(\cdot)$.

Two questions naturally arise: 1) Is there a higher lower bound for ε than $1/(T-1)$ when $\sharp$ is Toeplitz, 2) Can one use an estimator of ε in the test rather than routinely performing the conservative test?

Epsilon for Toeplitz Matrices

We let $\sharp_T = \text{Toepl}\ (\sigma(0),\ldots,\sigma(T-1))$, $A_T = I_T - \frac{1}{T}\underset{\sim}{1}_T\underset{\sim}{1}_T^T$, and also index ε with a T, i.e

$$\varepsilon_T = \frac{[\text{tr}(A_T\sharp_T)]^2}{(T-1)\text{tr}(A_T\sharp_T A_T\sharp_T)} .$$

While there appears to be no easily written lower bound for ε_T in terms of series length T, experience with a large number of possible autocovariance sequences indicates that ε_T rarely falls more than one or two percent below its limit as $T\to\infty$. This limit is given in the following theorem.

Theorem 2 (Spector and Newton [7])

If the covariance sequence $\{\sigma(v), v=0,\pm1,\ldots\}$ is absolutely summable then

$$\lim_{T\to\infty} \varepsilon_T = \frac{\sigma^2(0)}{\sum_{v=-\infty}^{\infty} \sigma^2(v)} = \frac{1}{\sum_{v=-\infty}^{\infty} \rho^2(v)}$$

where $\rho(v) = \sigma(v)/\sigma(0)$ is the autocorrelation function corresponding to $\sigma(\cdot)$. Further, if $f(\omega)$, $\omega\varepsilon[-\pi,\pi]$, is the spectral density function corresponding to $\sigma(\cdot)$ then

$$\lim_{T\to\infty} \varepsilon_T = \frac{\left[\int_{-\pi}^{\pi} f(\omega)d\omega\right]^2}{2\pi \int_{-\pi}^{\pi} f^2(\omega)d\omega} \quad .$$

We note that these quantities and their estimation have arisen elsewhere in time series analysis (see Parzen [5], p. 984 for example).

Suppose $\sigma(\cdot)$ is the autocovariance sequence of a covariance stationary autoregressive process of order p with coefficients $\underset{\sim}{\alpha} = (\alpha_1,\ldots,\alpha_p)^T$ and residual variance σ^2 (denoted AR(p,$\underset{\sim}{\alpha}$,σ^2)), i.e

$$\sum_{j=0}^{p} \alpha_j \sigma(j-v) = \delta_v \sigma^2 \quad , \quad v \geqslant 0 \quad ,$$

where $\alpha_0 = 1$ and δ_v is the Kronecker delta. Then for p=1 and p=2 we have the following corollary.

Corollary 1

If $\sigma(\cdot)$ corresponds to an AR(1) or AR(2) process we have

$$\lim_{T\to\infty} \varepsilon_T = \begin{cases} \dfrac{1-\alpha_1^2}{1+\alpha_1^2}, & p=1 \\ \dfrac{(1-\alpha_2^2)[(1+\alpha_2)^2-\alpha_1^2]}{\alpha_1^2(1-4\alpha_2+\alpha_2^2)+(1+\alpha_2^2)(1+\alpha_2)^2}, & p=2 \end{cases}.$$

In Figure 1 we graph the limiting values of ε_T for p=2 for values of α_1 and α_2 that make the process stationary, i.e values for which the zeros of $1 + \alpha_1 z + \alpha_2 z^2$ are outside the unit circle.

For example if T = 101 and $\alpha_1 = -1$, $\alpha_2 = .4$ then 1/(T-1) is .01 while effectively a lower bound for ε_T is .28. Thus if one had good estimators of α_1 and α_2 a much less conservative test of H_T and/or H_{GT} could be determined.

Using an Estimator of ε

We consider five estimators of ε.

Each consists of forming an estimator of $\ddagger$ from the N residual time series $\underset{\sim}{e}_{ik} = (e_{i1k},\ldots,e_{iTk})^T$ where

$$e_{ijk} = y_{ijk} - \bar{y}_{ij.} - \bar{y}_{i.k} + \bar{y}_{i..},$$

and then substituting this estimator of $\ddagger$ into (2) to estimate ε.

1) $\hat{\varepsilon}$ - Ignoring the Toeplitz form of $\ddagger$, one can estimate $\ddagger$ as one would in ordinary multivariate analysis, i.e

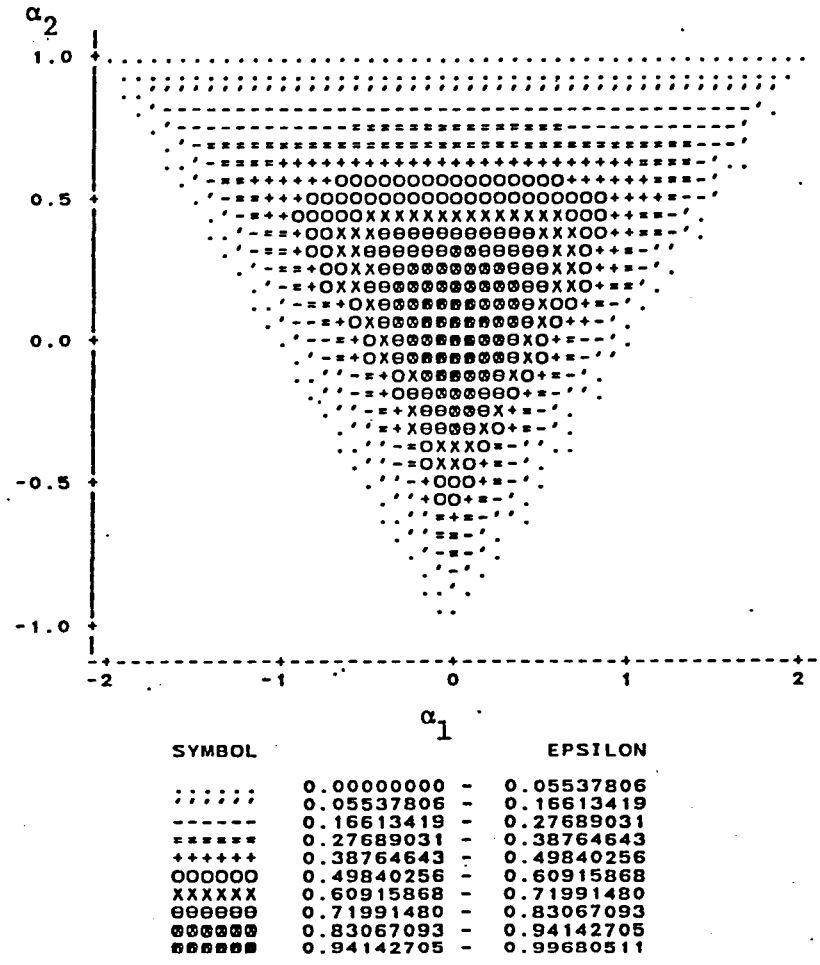


Figure 1. Limit of Epsilon for an AR(2) Process

$$\hat{\ddagger} = \frac{1}{N} \sum_{i=1}^{g} \sum_{k=1}^{n_i} (e_{ik} - \bar{e})\,(e_{ik} - \bar{e})^T$$

where $\bar{e} = \left(\frac{1}{N} \sum_{i=1}^{q} \sum_{k=1}^{n_i} e_{ijk}\right)_{Tx1}$. This is the traditional estimator (Huynh and Feldt [3]) used for general $\ddagger$.

2) $\hat{\varepsilon}^{(np)}$ - Nonparametric (i.e not assuming an AR model) Pooled estimators of $\hat{\sigma}(0),\ldots,\hat{\sigma}(T-1)$ of $\sigma(0),\ldots,\sigma(T-1)$ can be calculated and

then $\hat{\ddagger}^{(np)}$ = Toepl $(\hat{\sigma}(0),\ldots,\hat{\sigma}(T-1))$.

3) $\hat{\varepsilon}^{(p)}$ - Parametric (i.e assuming an AR model) pooled estimators $\tilde{\sigma}(0),\ldots,\tilde{\sigma}(T^*)$ of $\sigma(0),\ldots,\sigma(T^*)$ can be calculated and $\hat{\ddagger}^{(p)}$ = Toepl $(\tilde{\sigma}(0),\ldots,\tilde{\sigma}(T^*))$. The integer $T^* \geq T-1$.

4) $\hat{\varepsilon}^{(\infty,np)}$ - Nonparametric limit of epsilon estimator

$$\hat{\varepsilon}(\infty,np) = \frac{\hat{\sigma}^2(0)}{\sum_{|v|\leq T-1} \hat{\sigma}^2(v)}$$

5) $\hat{\varepsilon}^{(\infty,p)}$ - Parametric limit of epsilon estimator

$$\hat{\varepsilon}(\infty,p) = \frac{\tilde{\sigma}^2(0)}{\sum_{|v|\leq T^*} \tilde{\sigma}^2(v)}$$

To compare the performance of these estimators in terms of the size of the test of H_T and H_{GT} we generated 100 sets of nine zero mean time series of length 50 from each of twenty AR processes. These processes were chosen to present a wide range of time series types. In each set the nine series were randomly divided into three groups of three. Thus T=50, g=3, and $n_1=n_2=n_3=3$. For each data set, the five estimators of ε were calculated and for a given estimator ε^* the p-value of the test determined (assuming $F_T \sim F_{\varepsilon^*(T-1),\varepsilon^*(T-1)(N-g)}$ and $F_{GT} \sim F_{\varepsilon^*(T-1)(g-1),\varepsilon^*(T-1)(N-g)}$). Now if the test using ε^* has the correct size then the 100 p-values for each of the twenty AR models should appear to be a random sample of size 100 from a uniform

distribution on the interval zero to one (Lehmann [4], p. 150). In Table 2 we list the results of testing the p-values for uniformity using the Cramér-von Mises statistic as well as descriptive statistics for five estimators. From this table we note:

1) The traditional estimator $\hat{\varepsilon}$ is woefully inadequate for the types of data we're considering.

2) Using $\hat{\varepsilon}^{(\infty,np)}$ leads to a poor test.

3) Any of $\hat{\varepsilon}^{(np)}$, $\hat{\varepsilon}^{(p)}$, and $\hat{\varepsilon}^{(\infty,p)}$ lead to tests having good size.

Studying the power of the tests of H_T and H_{GT} numerically is of course very difficult as there are so many possible alternatives. To get some idea of the power, we generated 100 sets of 6 series of length 50 (allocated to 2 groups of 3 series) for each of the 20 AR models (these are the η_{ijk}'s) and then formed

$$y_{ijk} = \mu_{ij} + \eta_{ijk}$$

where

$$\mu_{ij} = \begin{cases} 0 & , \quad i=1,2, \quad j \neq 10 \\ \lambda & , \quad i=1, \quad j=10 \end{cases}$$

for $\lambda = 0, 2, 4, 6, 8, 10$. Thus the means are all zero except the 10th observation in group one is λ. In Figure 2 we give a typical empirical power curve again showing that the tests using $\hat{\varepsilon}^{(p)}$ and $\hat{\varepsilon}^{(np)}$ are competitive with the test using the true ε.

Table 2. Results of Using Five ε Estimators for 20 AR Processes

Order	Coeffs		ε_{50}		ε_∞	$\hat{\bar{\varepsilon}}$		$\hat{\bar{\varepsilon}}(np)$		$\hat{\bar{\varepsilon}}(p)$		$\hat{\bar{\varepsilon}}(\infty,np)$		$\hat{\bar{\varepsilon}}(\infty,p)$	
						$s^2_{\hat{\varepsilon}}$		$s^2_{\hat{\varepsilon}}(np)$		$s^2_{\hat{\varepsilon}}(p)$		$s^2_{\hat{\varepsilon}}(\infty,np)$		$s^2_{\hat{\varepsilon}}(\infty,p)$	
			CVMT[a]	CVMGT[b]		CVMT	CVMGT	CVMT	CVMGT	CVMT	CVMGT	CVMT	CVMGT	CVMT	CVMGT
1	-.8		.2403		.2195	.0827		.2246		.3192		.1289		.3074	
						$.7\times10^{-4}$		.0021		.0028		.0016		.0031	
			.077	.076		1.45	.975	.106	.097	.063	.175	.708	.424	.054	.158
1	-.5		.6127		.6000	.1018		.4923		.6584		.2786		.6497	
						$.3\times10^{-4}$		.0041		.0045		.0060		.0048	
			.881	.031		3.13	2.09	.935	.043	.910	.043	1.39	.446	.908	.040
1	.5		.6114		.6000	.1009		.4870		.6151		.2978		.6019	
						$.3\times10^{-4}$		.0037		.0038		.0058		.0040	
			.263	.197		2.58	1.98	.323	.162	.245	.183	.764	.420	.247	.176
1	.8		.2331		.2195	.0800		.1971		.2553		.1283		.2415	
						$.8\times10^{-4}$		.0021		.0029		.0014		.0029	
			.056	.090		1.21	.681	.098	.033	.075	.102	.431	.198	.067	.008
2	-.971	.464	.4688		.4710	.0962		.3676		.4658		.2227		.4633	
						$.4\times10^{-4}$		.0011		.0014		.0023		.0017	
			.157	.186		2.22	1.96	.226	.242	.171	.192	.640	.673	.173	.192
2	.019	.746	.3085		.2845	.0874		.2653		.3477		.1735		.3237	
						$.8\times10^{-4}$		.0021		.0027		.0017		.0028	
			.143	.131		1.73	1.76	.220	.250	.140	.101	.624	.657	.153	.131
2	1.746	.868	.1436		.1233	.0665		.1280		.1796		.0889		.1618	
						$.7\times10^{-4}$		.0004		.0005		.0003		.0006	
			.111	.082		.606	.641	.088	.106	.206	.107	.301	.320	.135	.080

Table 2 (Continued)

Order	Coeffs	ε_{50}	ε_∞	$\bar{\hat{\varepsilon}}$	$\bar{\hat{\varepsilon}}(np)$	$\bar{\hat{\varepsilon}}(p)$	$\bar{\hat{\varepsilon}}(\infty,np)$	$\bar{\hat{\varepsilon}}(\infty,p)$
2	-1.84 .861	.0724	.0808	$.0482_{-4}$	.0729	.1111	.0435	.1164
				$.7\times10^{-4}$	.0002	.0002	.0001	.0002
		.060 .153		.461 .560	.072 .177	.125 .106	.705 .719	.159 .121
3	-.690 -.771	.1827	.1862	$.0723_{-4}$	.1576	.2216	.0968	.2219
	.612			$.7\times10^{-4}$	.0006	.0008	.0007	.0008
		.354 .099		.896 .519	.311 .050	.428 .207	.534 .230	.431 .209
3	1.174 .252	.2616	.2498	$.0815_{-4}$	.2112	.2722	.1394	.2596
	-.121			$.6\times10^{-4}$	.0007	.0006	.0010	.0006
		.703 .078		1.85 1.45	.726 .201	.714 .076	1.05 .640	.705 .093
3	-1.404 1.188	.4409	.4383	$.0949_{-4}$	.3495	.4502	.2131	.4458
	-.474			$.5\times10^{-4}$	.0014	.0011	.0026	.0014
		.118 .078		2.08 1.40	.188 .056	.124 .081	.661 .310	.125 .078
3	-1.227 .0426	.0646	.0340	$.0455_{-4}$	.0676	.1031	.0409	.0779
	.5106			$.9\times10^{-4}$	.0002	.0002	.0001	.0003
		.226 .087		.821 .178	.233 .084	.025 .355	1.13 .288	.119 .135
4	-.250 .7287	.4836	.4602	$.0965_{-4}$	.3890	.5005	.2414	.4771
	.0126 .2951			$.5\times10^{-4}$	.0029	.0032	.0031	.0034
		.037 .058		2.39 1.93	.128 .120	.030 .050	.610 .544	.041 .054
4	-2.304 1.972	.1123	.1079	.0586	.1023	.1405	.0671	.1398
	-.7915 .1724			.0001	.0003	.0004	.0002	.0006
		.133 .301		869 .432	.198 .265	.096 .455	.677 .332	.089 .440

Table 2 (Continued)

Order	Coeffs	ε_{50}	ε_∞	$\hat{\bar{\varepsilon}}$	$\hat{\bar{\varepsilon}}(np)$	$\hat{\bar{\varepsilon}}(p)$	$\hat{\bar{\varepsilon}}(\ ,np)$	$\hat{\bar{\varepsilon}}(\ ,p)$
4	1.672 1.800	.3632	.3438	.0907	.2948	.3790	.1899	.3579
	.9796 .3439			$.5\times10^{-4}$	.0013	.0015	.0016	.0016
		.114 .491	1.71 2.06		.137 .571	.132 .510	.443 .940	.123 .518
4	.0152 -1.039	.4014	.3929	.0938	.3285	.4151	.2040	.4029
	.1036 .4777			$.4\times10^{-4}$	.0016	.0016	.0023	.0018
		.111 .279	1.73 2.21		.086 .398	.048 .257	.460 .974	.049 .271
5	-.602 .7191	.1670	.1425	.0731	.1587	.2105	.1084	.1865
	.2561 .0322			.0001	.0011	.0014	.0008	.0015
	.1136	.059 .137	.988 .525		.094 .119	.048 .253	.415 .203	.047 .174
5	.3898 -1.135	.1944	.1115	.0761	.1888	.2767	.1119	.2130
	-.5186 .7926			.0001	.0024	.0036	.0012	.0041
	.4913	.062 .095	1.297 1.13		.129 .124	.111 .041	.683 .632	.088 .065
5	-2.119 1.788	.0566	.0226	.0420	.0588	.0923	.0324	.0624
	-.2257 -.8062			$.5\times10^{-4}$	$.9\times10^{-4}$	.0001	$.5\times10^{-4}$	.0002
	.5863	.125 .223	.391 .617		.127 .233	.355 .058	.930 1.10	.128 .195
5	-1.848 1.983	.2769	.2053	.0873	.2741	.4055	.1498	.3857
	-1.476 .6256			.0001	.0031	.0024	.0026	.0036
	-.1861	.088 .071	1.52 1.29		.095 .086	.217 .165	.634 .571	.178 .135
Unif. Tests Rejected (α = .05)		2 1	19 18		2 1	2 0	15 10	2 1

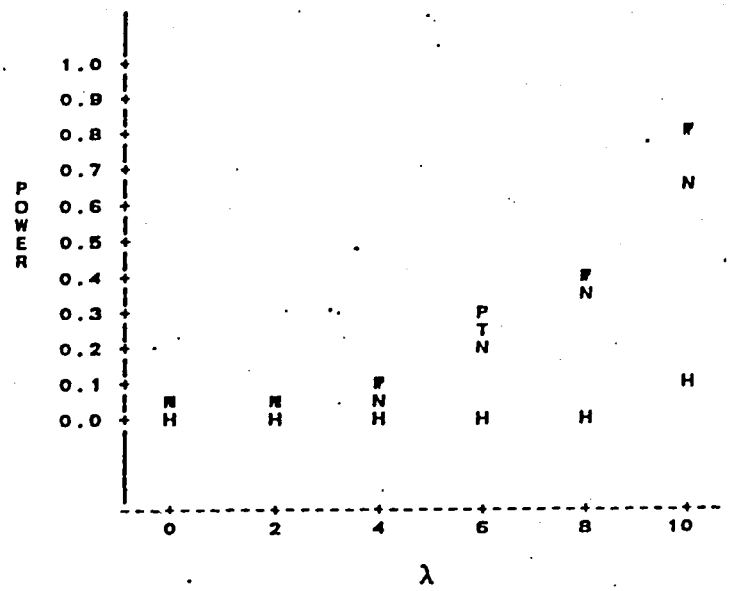
[a]CVMT: Cramer-Von Mises Statistic for Testing Uniformity of p-values for H_T (5% critical value = .461)

[b]CVMGT: Same as CVMT except for H_{GT}

Model: AR2(0.01895, 0.74637)

$H = \hat{\varepsilon}$ $N = \hat{\varepsilon}^{(np)}$ $P = \hat{\varepsilon}^{(p)}$ $T = \varepsilon$

Main Effect Test



Interaction Test

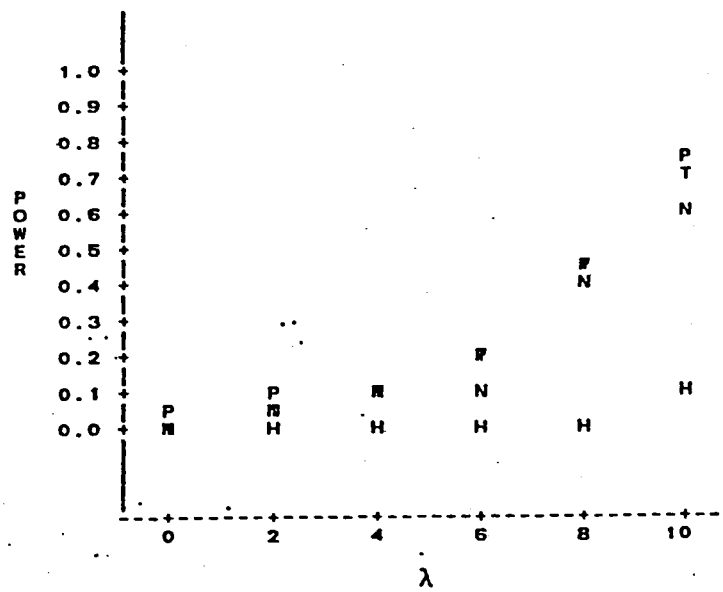


Figure 2. Empirical Power Curves of Tests of H_T and H_{GT}.

3. Extension to Multivariate Analysis of Variance

The extension of the method of Section 2 to the case where y_{ijk} is a vector rather than a scalar is not obvious. We are currently investigating the effect of having correlation across the levels of one factor in a two factor multivariate analysis of variance (MANOVA) as this is how the correction factor ε was first discovered in the univariate case (see Box [1]). A promising area of investigation is to note that the distribution of a statistic that is a transformation of Wilk's lambda can be well approximated by a random variable having an F distribution (see Rao [6], p. 556).

References

[1] Box, G. E. P., Some theorems on quadratic forms applied in the study of analysis of variance problems, II. Effects of inequality of variance and of correlation between errors in the two-way classification, Annals of Mathematical Statistics. 25 (1954) 484-498.

[2] Geisser, S. and Greenhouse, S. W., An extension of Box's results on the use of the F-distribution in multivariate analysis, Annals of Mathematical Statistics. 29 (1958) 885-891.

[3] Huynh, H. and Feldt, L. S., Estimation of the Box correction for degrees of freedom from sample data in randomized block and split-plot designs, Journal of Educational Statistics, 1 (1976) 69-82.

[4] Lehmann, E. L., Testing Statistical hypotheses (Wiley, New York, 1959).

[5] Parzen, E., An approach to time series analysis, Annals of Mathematical Statistics. 32 (1961) 951-989.

[6] Rao, C. R., Linear statistical inference and its applications (Wiley, New York, 1973).

[7] Spector, P. and Newton, H. J., Box's degrees of freedom correction factor for time-repeated measures designs, Technical Report, Department of Statistics, Texas A&M University (June, 1983).

SAR SPECKLE NOISE REDUCTION USING WIENER FILTER

Tae H. Joo

and

Daniel N. Held

JET PROPULSION LABORATORY

Pasadena, CA

ABSTRACT

Synthetic aperture radar images are degraded by speckle. In this paper, we present a multiplicative speckle noise model for SAR images. Using this model, we derive a Wiener filter by minimizing the mean-squared error using the known speckle statistics. Implementation of the Wiener filter is discussed and experimental results are presented. We conclude with a discussion of possible improvements to this method.

Introduction

Synthetic aperture radar (SAR) is a coherent imaging system [1]. SAR imagery suffers from speckle noise degradation. The speckle noise results from coherent illumination of a rough surface and its characteristics are well known [3]. In the following, we derive the Wiener filter based on a multiplicative noise model, discuss the filter implementation, and show the experimental results of the filtering. We conclude the paper by outlining planned future work.

Wiener Filter

The speckle noise intensity is described by an exponential probability density with identical mean and standard deviation. In order to use the speckle statistics in reducing the noise in SAR intensity images, we propose the following signal processing model.

$$y(n\ m) = s(n\ m)\ d(n\ m) \tag{1}$$

where $y(n\ m)$ = SAR intensity image

$s(n\ m)$ = scene

$d(n\ m)$ = speckle noise

The probability density function of the speckle noise, $d(n\ m)$, is

$$pdf(D) = \begin{cases} e^{-D} & D \geq 0 \\ 0 & D < 0 \end{cases}$$

with mean = 1 and variance = 1. The mean and the standard deviation of y(n m) are equal to the scene, s(n m).

Using (1), we design a Wiener filter to estimate s(n m) given y(n m). The Wiener filter is the optimal linear filter in the sense that it minimizes the expected value of the mean squared error between the true and the estimated signals [4]. The estimate of the scene is denoted $\hat{s}(n\ m)$ and is determined by filtering y(n m) such that $\hat{s}(n\ m) = h(n\ m) * y(n\ m)$ where h(n m) denotes the Wiener filter and * indicates convolution. In frequency domain,

$$\hat{S}(\omega_1\ \omega_2) = H(\omega_1\ \omega_2)\ \ Y(\omega_1\ \omega_2)$$

where capital letters denote the Fourier transformed functions. We minimize error = $E((s(n\ m) - \hat{s}(n\ m))^2)$ where E(.) denotes the expected value. Using the orthogonality principle, which states that the best linear estimate is obtained if the error between the desired and estimated is uncorrelated with the observations, we have

$$R_{ys}(n\ m) = h(n\ m) * Ryy(n\ m) \qquad (2)$$

where $R_{ys}(n\ m) = E(y(1\ k)\ s(1-n\ k-m))$

$R_{yy}(n\ m) = E(y(1\ k)\ y(1-n\ k-m))$

and stationarity is assumed.

Equation (2) is the Wiener-Hopf equation for this problem.

From the model (1) and assuming the scene and the noise are uncorrelated, (2) becomes, in the frequency domain,

$$H(\omega_1\ \omega_2) = \frac{P_s(\omega_1\ \omega_2)}{P_y(\omega_1\ \omega_2)} = \frac{P_s(\omega_1\ \omega_2)}{P_s(\omega_1\ \omega_2) * P_d(\omega_1\ \omega_2)} \qquad (3)$$

where

$P_s(\omega_1\ \omega_2)$ is the power density spectrum of s(n m)

$P_d(\omega_1\ \omega_2)$ is the power density spectrum of d(n m)

and * denotes convolution.

The Wiener filter, given by (3), requires the knowledge of the power density spectra (PDS's) of both the noise and the scene. We now discuss a method of determining the power spectra and implementing of the filter.

Implementation

As derived by Goodman [3], the autocorrelation function of the speckle noise is the sum of a constant term and a function which is dependent on the scattering area. We assume that the scattering area is such that the PDS of the noise is a bandlimited white spectrum with an impulse at DC corresponding to a constant offset in the correlation domain. Using the fact that half of the noise power is contained in the DC component and half at other frequencies [2], we have

$$P_d(\omega_1\ \omega_2) = \delta(\omega_1\ \omega_2) + \frac{1}{W_1 W_2} \qquad (4)$$

where $\delta(\omega_1\ \omega_2)$ = two-dimensional impulse function

and $$|P_d(\omega_1\ \omega_2)| = 0 \quad \text{for } |\omega_1| > w_1/2 \quad \text{and } |\omega_2| > w_2/2.$$

Using (4) in (3), the Wiener filter becomes

$$H(\omega_1\ \omega_2) = \frac{P_s(\omega_1\ \omega_2)}{P_s(\omega_1\ \omega_2) + \frac{1}{W_1 W_2}\int_{-w_1/2}^{w_1/2}\int_{-w_2/2}^{w_2/2} P_s(\omega_1\ \omega_2)\, d\omega_1 d\omega_2}$$

and using

$$\int_{-w_1/2}^{w_1/2}\int_{-w_2/2}^{w_2/2} P_y(\omega_1\ \omega_2)\, d\omega_1 d\omega_2 = 2\int_{-w_1/2}^{w_1/2}\int_{-w_2/2}^{w_2/2} P_s(\omega_1\ \omega_2)\, d\omega_1\, d\omega_2$$

we finally have

$$H(\omega_1\ \omega_2) = \frac{P_y(\omega_1\ \omega_2) - \frac{1}{W_1 W_2}\int_{-w_1/2}^{w_1/2}\int_{-w_2/2}^{w_2/2} P_y(\omega_1\ \omega_2)\, d\omega_1 d\omega_2}{P_y(\omega_1\ \omega_2)} \qquad (5)$$

Equation (5) describes the Wiener filter to be implemented. Note that only the PDS of the image is required.

We estimate the PDS of y(n m) by averaging its periodograms. If the underlying process is white Gaussian, the variance of the averaged-periodogram estimator is reduced by $1/\sqrt{N}$ if N peridograms are averaged [6]. In this work, we average four periodograms to estimate $P_y(\omega_1\ \omega_2)$ and determine $H(\omega_1\ \omega_2)$. Because the filter is of infinite duration, it must be truncated. In practice, most of the energy is concentrated near the origin thus truncation does not cause much difficulty.

In practice, the Wiener filter of (5) is approximated using the discrete Fourier transforms. Using the averaged-periodogram

estimate of PDS of y(n m), denoted by $P_y(k_1k_2)$, (5) is replaced by

$$H(k_1k_2) = \frac{P_y(k_1k_2) - \frac{1}{2N^2}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}P_y(k_1k_2)}{P_y(k_1k_2)} \qquad (6)$$

where N is the discrete Fourier transform length.

Figure 1 shows the algorithm based on (6).

Experimental Results

We have applied the signal processing algorithm as described in Fig. 1 on a SEASAT SAR image of an agricultural field (SEASAT orbit number 1355). Figure 2(a) is the given intensity image and Fig. 2(b) is the Wiener filtered image. These images indicate that filtering reduces the speckle noise significantly. Figure 3(a) is the Wiener filter in the frequency domain which has the low-pass characteristic since the data is basically a low-pass signal as shown in 3(b). Figure 4 shows the slices of the impulse response of the truncated filter which indicates that most of the energy is indeed concentrated near the origin.

We define the "equivalent number of looks" (ENL) of the image by ENL = mean/standard deviation. The ENL of an area with uniform reflectivity is equal to 1 because of the exponential probability density function of speckle noise. For the filtered image of Fig. 2(b), the computed ENL is approximately 2.2. Figure 5 shows the 2-look image obtained by incoherent averaging of the image [7]. By comparing Figures 2(b) and 5, we conclude that, qualitatively, the speckle noise of the filtered image is, as expected, reduced to that of the 2-look image.

Conclusions

In this paper, we have derived the Wiener filter for multipicative speckle noise model using a priori knowledge of the noise PDS. An algorithm for implementation of the Wiener filter was discussed. The results of Wiener filtering were given and compared to the 2-look image. The Wiener filtering significantly reduced the speckle noise.

We conclude the paper by outlining three extensions of the work which are to be investigated. First, segmentation of the image will be examined. In the derivation of the Wiener filter, we assumed that the scene was stationary. In general, the scene is not stationary and by segmenting the image into smaller pieces, we can improve the "stationarity" of the scene. Second, other PDS estimators will be examined. In the implementation, we used the averaged periodogram to estimate the PDS's. PDS estimators such as MLM or MEM [5], which have better resolution, might be employed to improve the estimate. Third, an alternative signal processing model which includes the system response function will be examined. By using (1), we assume that the system response function of the imaging system is an impulse. By using an alternate signal processing model which includes the imaging system response function, we can remove the effect of the imperfect imaging system.

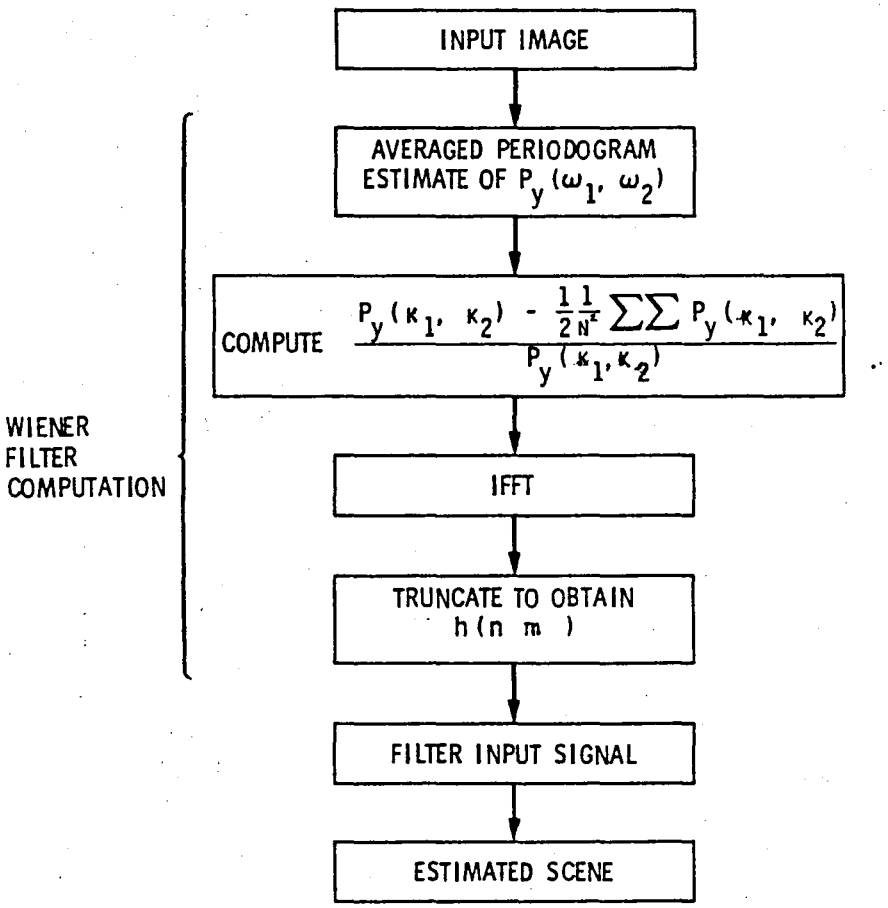


Fig. 1 Wiener filtering algorithm

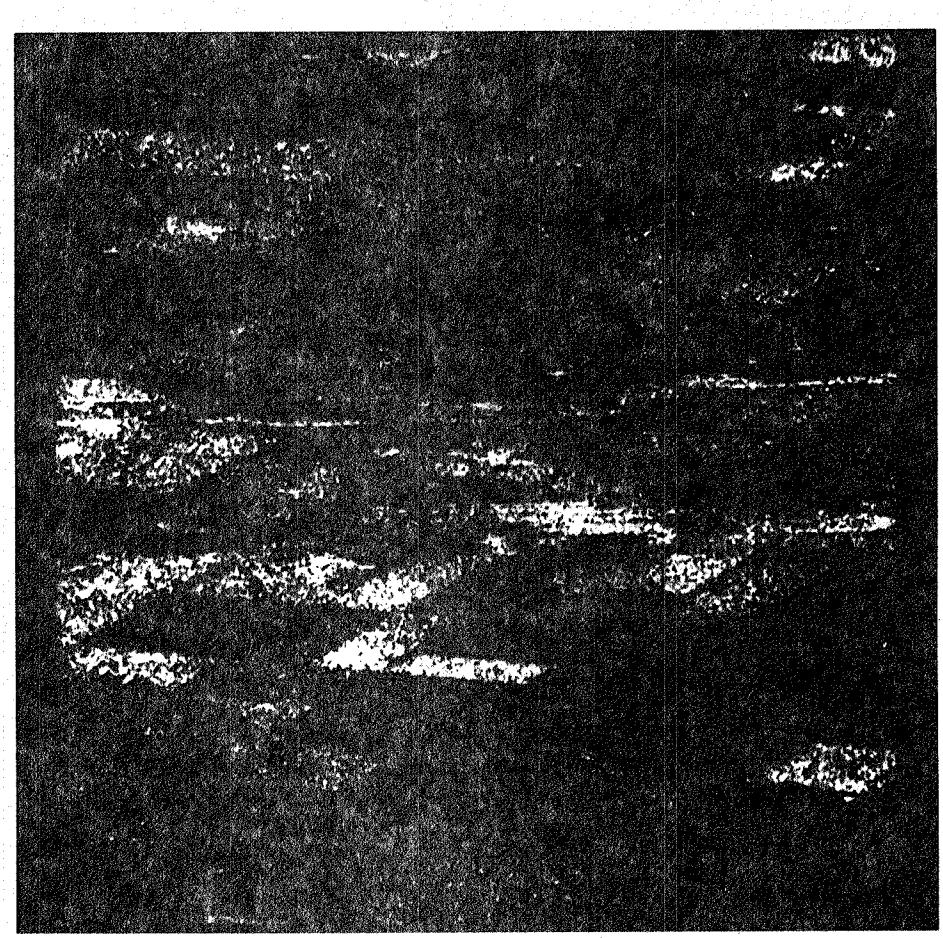


Fig. 2(a) Original noisy image

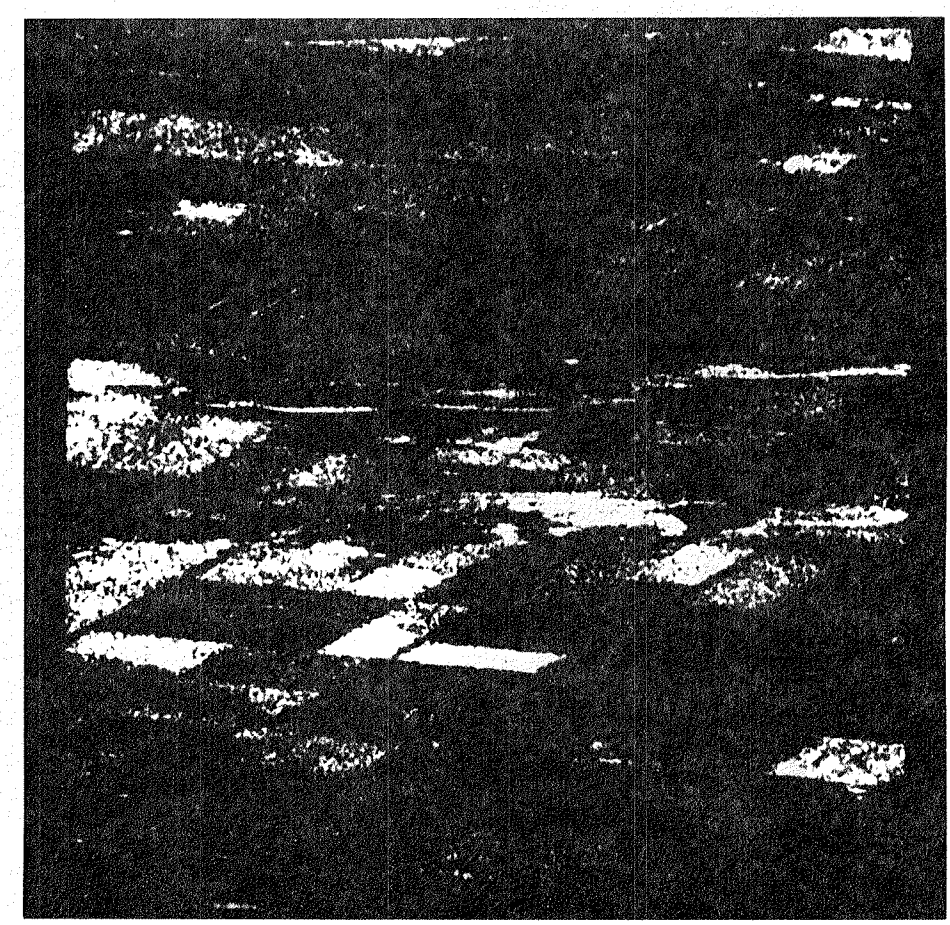


Fig. 2(b) Filtered image

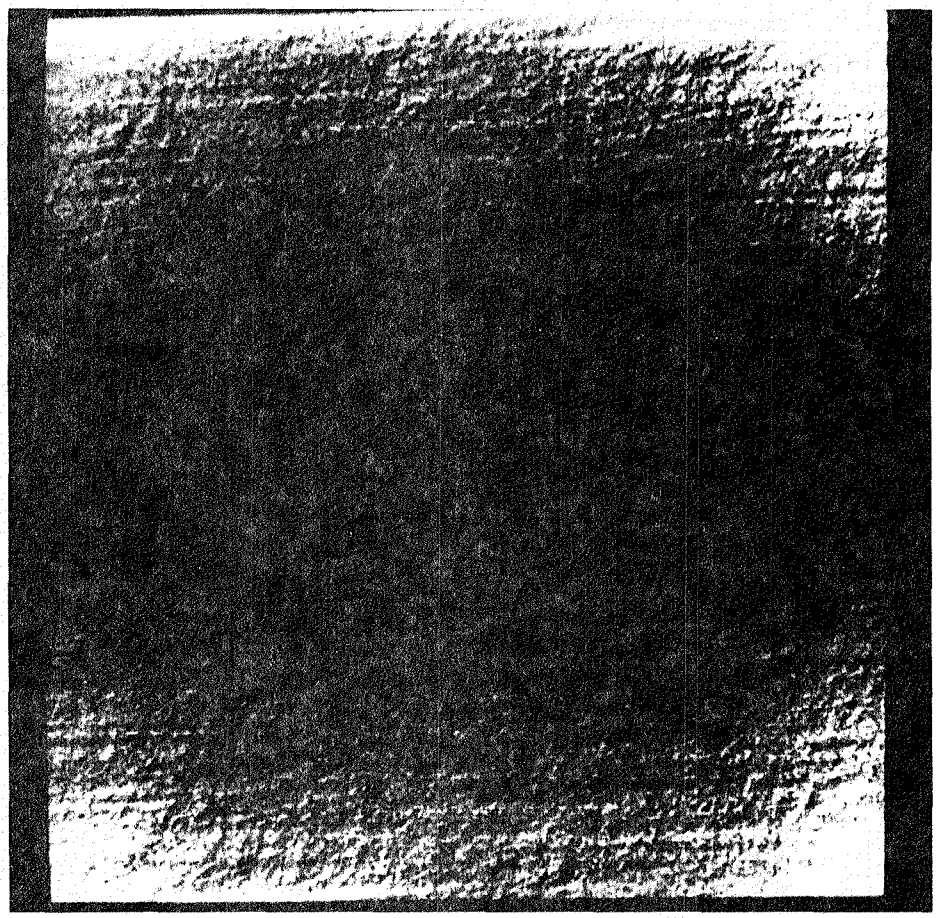


Fig. 3(a) Wiener filter in frequency domain

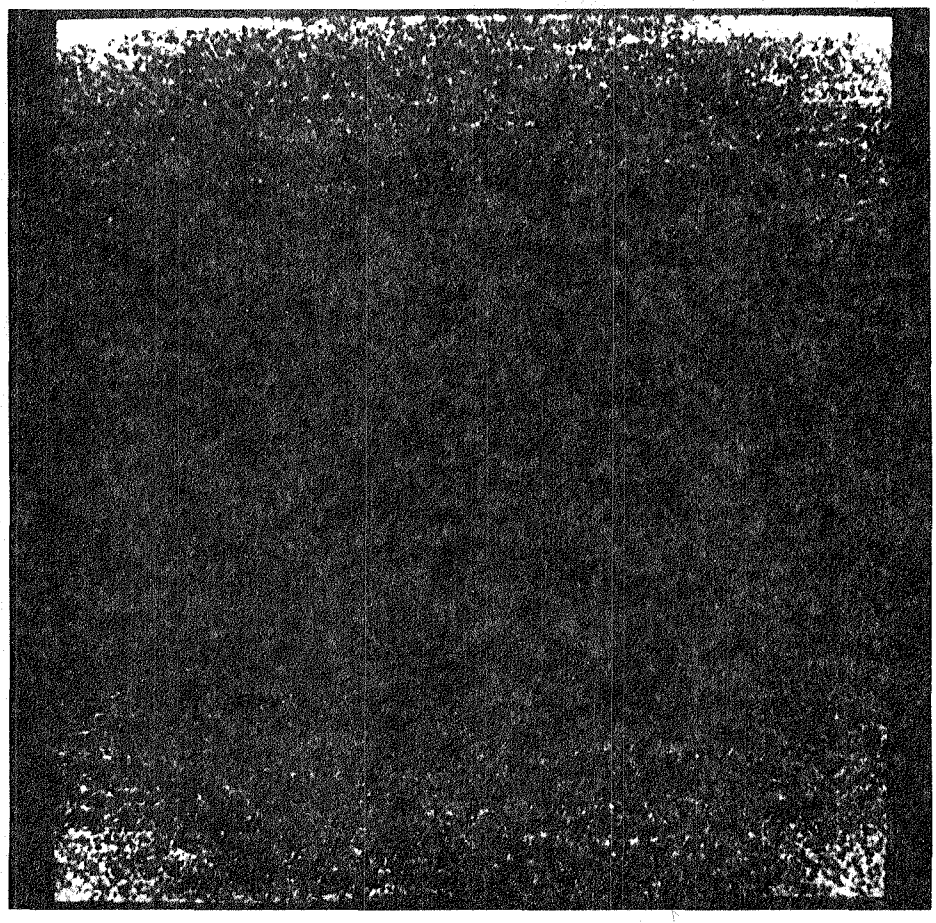


Fig. 3(b) Fourier transform of original image

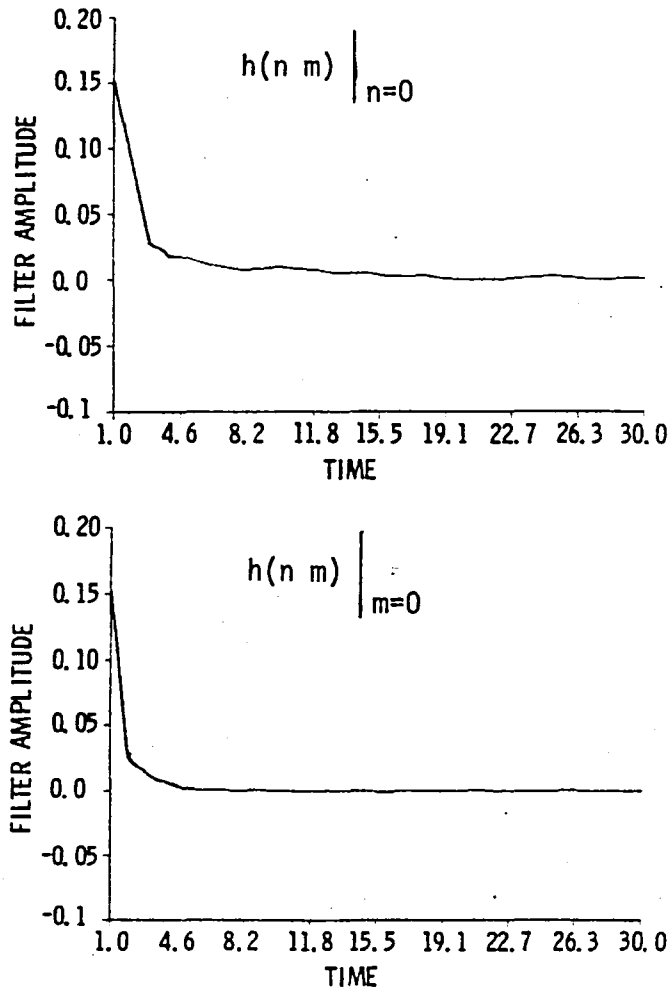


Fig. 4 Slices of the impulse response of the Wiener filter for Fig. 2(a).

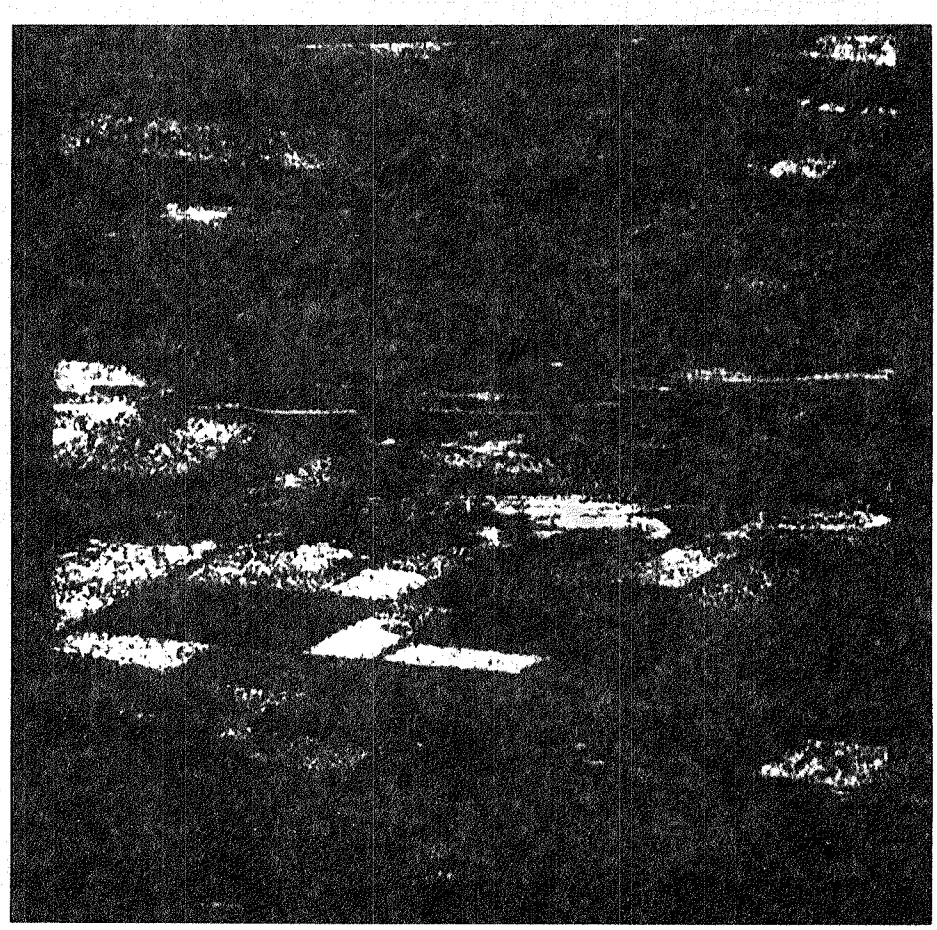


Fig. 5 2-look image

References

[1] Elachi, C. , Bicknell, T., Jordan, R. L., and Wu, C., Spaceborne Synthetic-Aperature Imaging Radars: Applications Techniques and Technology, Proc. IEEE 70 (1982) 1174-1209.

[2] Goldfinger, A. D., Estimation of Spectra from Speckled Image, IEEE Transaction on Aerospace and Electronic Systems. AES-18 (1982) 675-681.

[3] Goodman, J.W., Statistical Properties of Laser Speckle Pattern, in Daintz, J. C. (ed.), Laser Speckle (Springer-Verlay, NY, 1975).

[4] Kondo, K., Ichioka, Y., and Suzuki, T., Image Restoration by Wiener Filtering in the Presence of Signal-dependent Noise, Applied Optics 16 (1977) 2254-2258.

[5] McClellan, J. H., Multi-dimensional Spectral Estimation, Proc. IEEE 70 (1982) 1029-1039.

[6] Oppenheim, A. V., and Schafer, R. W., Digital Signal Processing (Prentiss-Hall Inc., Englewood Cliffs, N. J., 1975).

[7] Porcello, L. J., Massey, N. G., Innes, R. B., and Marks, J. M., Speckle Reduction in Synthetic-Aperature Radars, J. Opt. Soc. Am. 66(1976) 1305-1311.

IMAGE MATCHING USING GENERALIZED HOUGH TRANSFORMS

Larry S. Davis
Fu-pei Hu
Vincent Hwang
Les Kitchen

U. M. College Park

ABSTRACT

This report describes an image matching system specifically designed to match dissimilar images. A set of blobs and ribbons is first extracted from each image, and then generalized Hough transform techniques are used to match these sets and compute the transformation that best registers the image. An example of the application of the approach to one pair of remotely sensed images is presented.

This report describes progress to date on our research into the problem of matching "dissimilar" images. The dissimilarity may be due to significant changes in the scene being imaged or to the utilization of somewhat different sensors to image the same scene. In either event, one cannot expect to be able to match, or register, such images using conventional image registration techniques based on either direct intensity cross correlation or even on somewhat more sophisticated feature (e.g., edge) correlation techniques. Instead, we suggest that the images to be matched be subjected to a rather complex analysis in order to construct descriptions of the images in terms of relatively high level pieces (in the examples shown in this paper the pieces are blobs and ribbons). These pieces can, in principle, be interpreted in the context of a model for the classes of entities that are likely to appear in the images, and it is the resulting symbolic descriptions which are matched to register the images. This interpretation step is not discussed in this paper, but is a topic currently under investigation in our laboratory. Related work on symbolic image matching appears in Price [5].

Blob and Ribbon Detection

In an image, blobs and ribbons extracted usually correspond to interesting objects . For example, in aerial imagery, blobs extracted may correspond to houses and ribbons may correspond to roads. What follows is a description of algorithms for blob and ribbon extraction.

Blob Extraction:

A blob is a compact homogeneous region. In order to extract blobs, we first segment the image into homogeneous regions. Then we compute the properties of each homogeneous region and extract those regions which satisfy the blob criteria.

To segment the image into homogeneous regions, we first convolve the image with a Laplacian operator. The places where the convolved result changes sign correspond to the locations of intensity changes in the original image [3]. If we assume the intensity of the regions to be extracted is lighter than the intensity of the background, the regions to be extracted are those regions with positive value in the convolved image.

The scale of the Laplacian operator determines the scale of the positive regions in the convolved image. If we know the scale of the blobs we want to extract, we can select a Laplacian operator with the appropriate scale.

In our method, the Laplacian operator is a difference of averages between two square windows; the Laplacian's scale is specified by the sizes of the two windows used. Uniform weight is assigned to every point in the mask.

After the positive regions in the convolved image are extracted, we need to compute their properties. Assume the size of a region is A and its perimeter is P. The compactness of the region is defined as:

$$\text{compactness} = \frac{P*P}{A}$$

In the experiment presented in the next section, 18 is used as the upper bound on the compactness of regions. All regions with compactness smaller than 18 are considered to be compact. The value 18 is obtained by computing the compactness measurement for a rectangle whose length is twice as long as its width.

All the regions which satisfy the compactness criterion are blobs. However, since we apply a large scale Laplacian operator to the image, there may be some artifacts in the convolved image. For example, two separated compact regions in the image may be merged into a connected positive region in the convolved image. The merged region is usually not compact. To recover from such artifacts, we apply an 8-connected shrinking operation to the convolved image. This may break some regions into several smaller regions. All the newly generated regions which satisfy the compactness criterion are also blobs.

Ribbon Extraction:

A ribbon is an elongated homogeneous region. As discussed above, we can extract homogeneous regions by an edge detection operation. We need to decompose these regions into subregions which are elongated and whose width along the skeleton of the region is some constant. In the following, the term "ribbon" refers to a constant width ribbon with some minimal length.

In our method, we first apply a topology preserving 8-connected thinning operation [6] to the convolved image. This operation produces the skeleton map of the regions in the convolved image. We want to decompose the skeleton into line segments such that all points on

the same line segment have nearly the same distance to their nearest background points.

A branch point is a point on the skeleton map which is adjacent to at least three different skeleton points. After we compute the skeleton of a region, we delete all branch points on the skeleton. For each connected (8-connected) line segment in the resulting skeleton map, we computed the ideal width for it by histogramming the widths along the skeleton and choosing the most frequently encountered width.

The ideal width of a skeleton line segment is used to determine whether a point on the skeleton line segment is part of the skeleton of some ribbon. Suppose the ideal width of a skeleton line segment is w. A point P on the line segment is on the skeleton of some ribbon (i.e., is a ribbon point) iff:

$w-e < \text{width at } P < w+e$

Long, connected sets of ribbon points constitute ribbons. In the experiment described in the next section, only blobs are used to compute the registration; we are currently extending our registration system to include ribbons.

Image Matching

Once a description of the ribbons and blobs in two images has been computed, these descriptions can be used to match the two images using Generalized Hough Transforms (GHTs). The GHT is a generalization of the classical Hough transform algorithms which were used to detect simple shapes such as lines, circles and ellipses in images (Ballard [1], Yam and Davis [9] describe the generalizations).

The GHT can be simply illustrated by considering the problem of matching point patterns under simple transformations. Let P=[{p1,..., pn} be one set of points in the plane (P might correspond to the locations of features in one of two images that we are attempting to register) and let Q={q1,...,qm} be the second of the two point patterns (Q might be the locations of features in a small window of the second image). The problem is to determine if Q matches well against a subset of P with respect to a given set of point transformations (such as translations and rotations). One straightforward way of determining how well Q matches P is to apply the transformations, one at a time, to Q and, for each transformation, count how many points from Q are mapped onto points in P. In practice, there are only a finite number of transformations because of the bounded size of the images from which P and Q are extracted, and the limited precision to which we represent the positions of the points in P and Q. We should point out that simple binary correlation algorithms for matching under translation work exactly in this way since they slide an image containing Q over all positions in the image containing P. If T is the number of possible transformations, then this straightforward algorithm requires time proportional to Tmn.

This turn out, however, not to be the computationally most efficient way to match Q and P. If we are able to commit extra storage, then we can dramatically cut down on the amount of computation. The storage required is proportional to the number of possible transformations (although later we will briefly discuss methods which often reduce the amount of storage required). One needs to construct an

array of accumulators, with one accumulator for each of the possible transformations. After the GHT algorithm operates, the value stored in any of these accumlators will be the number of points in Q mapped onto (or, more accurately, tolerably close to) some point in P by the transformation represented by that accumulator. Consider the special case now where T contains only translations. Let HT be the array of accumulators. Then the GHT algorithm is:

```
For each point q = (xq,yq) in Q
    For each point p = (xp,yp) in P
        Let dx = xp-xq
        Let dy = yp-yq
        HT(dx,dy) = HT(dx,dy) + 1
```

In this simple case, the comparison of a point in P with a point in Q results in incrementing only a single accumulator in the array HT. This is because, of course, only two points are needed to completely determine the transformation. More generally, however, comparing a single point in P with a single point in Q will not specify a unique transformation, but will rather specify a family of transformations corresponding to some subspace of the space of transformations represented by the array HT. One can ordinarily cut down on the size of this subspace by comparing, e.g., pairs of points from P against pairs of points from Q. However, unless one can introduce some heuristics to limit the number of such pairs (or, more generally, triples, quadruples, etc.) such an approach quickly becomes computationally unfeasible.

Consider next the slightly more complicated situation where T consists of not only translations, but image plane rotations as well. Now, the array HT is a three dimensional array, the third dimension needed to represent the rotation parameter. The GHT algorithm in this case is:

```
For each q = (xq,yq) in Q
    For each p = (xp,yp) in P
        For r = 0, 2π, by dr
            xq' = xq cosr
            yq' = yq sinr
            dx  = xq'-xp
            dy  = yq'-yp
            HT(dx,dy,r) = HT(dx,dy,r) + 1
```

Here, we first apply a rotation to point q and then determine the unique translation that will map the rotated version of q onto p. Notice that it would not have been appropriate to have fixed, e.g., dx and then attempted to determine a dy and r which would map q onto p since for most dx no such dy and r would exist. We should also point out that the values of r, dx and dy computed by the above algorithms have to be subjected to some truncation so that they can be associated with an accumulator in HT.

The above algorithm can be easily adapted to matching pairs of blob patterns. We associate a position (e.g., the centroid) with each blob, and then the remaining attributes of the blob (e.g., size, orientation, compactness) can be used both to limit the pairs of blobs

which are considered as possible matches, and to bound the possible transformations which can relate the blobs. For example, one can combine the compactness and orientations of two blobs to limit the number of rotations which need be considered when the two blobs are compared. Intuitively, if both blobs are very compact (i.e., nearly round) then one cannot place too much confidence in the estimate of orientation of the blob so that perhaps all possible rotations must be considered. On the other hand, if both blobs are relatively elongated, then one might only consider a small set of rotation angles centered around the orientational difference of the axes of the two blobs.

We now turn to the problem of representing the space of transformations. The most straightforward representation is to construct an n-dimensional array, one dimension for each parameter in the set of transformations. While this is reasonable for low dimensional transformations (such as translations), it is not a reasonable approach for higher dimensional transformations. We can identify at least three alternative approaches to direct representation of the higher dimensional array.

1) Multiresolution - initially, use a very coarsely quantized high dimensional array (for example, for rotations and translations we might initially quantize the translation parameters to every 10-20 pixels and the rotation parameter to every 10-20 degrees). This will make the size of the higher dimensional array manageable. Compute the GHT using this coarse representation, and find the most likely transformation(s). Using the

same storage, compute a second GHT, but with the range of the parameters now restricted by the coarse match. This approach was used by Stockman [8].

2) Projections - Compute various projections of the high dimensional array, and search for consistent and highly likely transformations in the projections. For example, if the set of transformations includes translations (dx, dy) and image plane rotations (r), then we can compute the (dx,r) and (dy,r) projections of the three-dimensional (dx,dy,r) parameter space, and choose the peaks from (dx, r) and (dy,r) that agree on the rotation. This is the approach used in the experiments presented in the next section.

3) Adaptive quantization - Several data structure have been proposed which essentially provide a form of adaptive quantization for representing data distributions in high dimensional sapces. These data structures are based on a recursive decomposition of the space into pieces; by attempting to equalize the probability that a data point falls into any element of the decomposition, parts of the space that have higher density of data points are represented at higher resolution. Examples of such data structures are Sloan [7] and O'Rourke [4]. In the former, the decomposition is regular (i.e., subspaces are split in "half" at each stage of the decomposition), while

the latter constructs an irregular decomposition.

It is also possible to utilize the GHT algorithm to match images based on descriptions of the ribbons that appear in the images. In Davis [2], we described a GHT algorithm for matching patterns of geometric entities, such as straight line and circular arc segments. This algorithm can be easily adapted to the case where the segments have additional properties, such as the width property that is associated with ribbons.

Experimental Results

We have applied the GHT matching algorithm to blob representations of several image pairs; in this section we will present the results of one such experiment. Figures 1-2 contain two images from a pair of aerial photographs of a suburban area. Figure 2a contains just that part of the second photograph that we will match against Figure 1a. Figures 1b and 2b show the blobs detected by the algorithm described in Section 2, and Tables 1-2 contain descriptions of the blobs (position, orientation of principal axis, size, and compactness) extracted from the two images.

The GHT algorithm assumed that the matching transformation consisted of an image plane translation and rotation, so that the Hough transform is a three-dimensional space. We adopted the strategy of computing only projections of the three-dimensional space, and chose the (dx,r) and (dy,r) projections. The projected Hough transforms are displayed in Figure 3. The registration accuracy is correct to one pixel in translation and 2° in rotation.

References

[1] Ballard, D., Generalizing the Hough transform to detect arbitrary shapes, in Haralick, R. and Simon, J.C. (eds.), Digital Image Processing (Proceedings of a NATO Advanced Study Institute, Bonas, France, June 23-July 4, 1980, D. Reidel, Dordrecht, Holland, 1981), 111-122.

[2] Davis, L.S., Hierarchical generalized Hough transforms and line-segment based generalized Hough transforms, Pattern Recognition 15 (1982) 277-285.

[3] Marr, D. and Hildreth, E., Theory of edge detection, Proc. R. Soc. London Ser. B 207 (1980) 187-217.

[4] O'Rourke, J., Dynamically quantized spaces for focusing the Hough transform, IJCAI 81, Vancouver B.C., Canada, August 24-28, 1981, 734-736.

[5] Price, K.E., Symbolic matching of image and scene models, Image Understanding Workshop, Palo Alto, CA, September 15-16, 1982, 299-308.

[6] Rosenfeld, A. and Kak, A., Digital Picture Processing (Academic Press, New York, 1976) 362-366.

[7] Sloan, K.R., Jr., Dynamically quantized pyramids, IJCAI 81, Vancouver B.C., Canada, August 24-28, 1981, 75-80.

[8] Stockman, G., Kopstein, S., and Benett, Matching images to models for registration and object detection via clustering, IEEE-PAMI 4 (1982) 229-241.

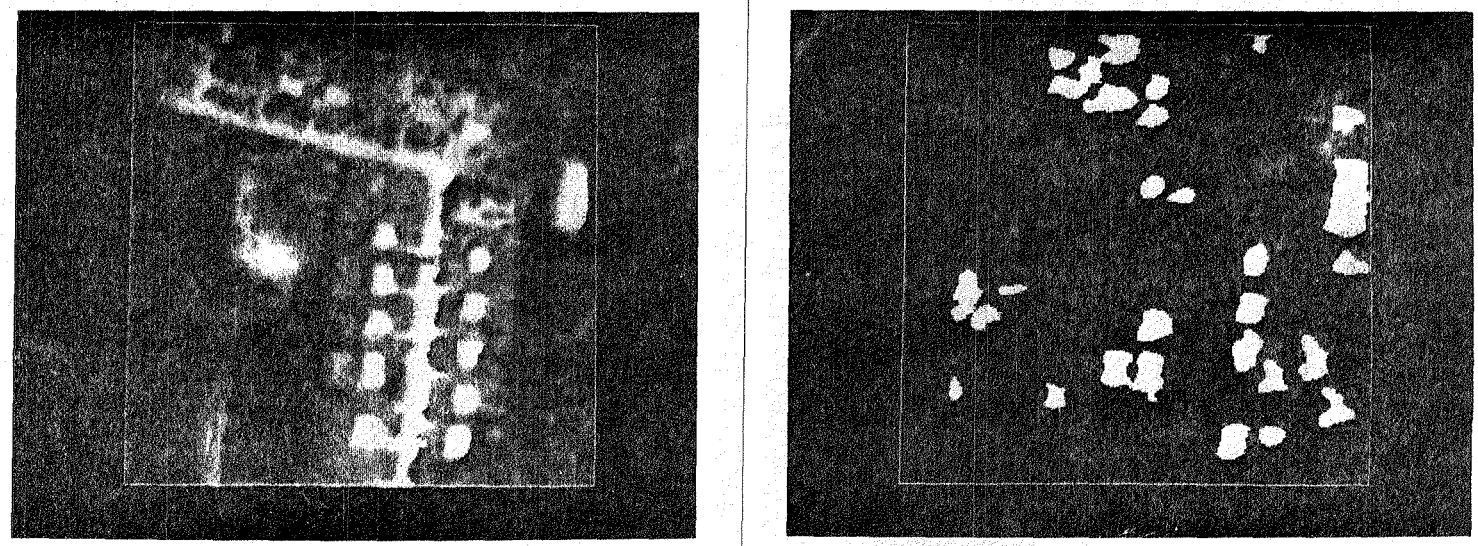
[9] Yam, S. and Davis, L.S., Image registration using generalized Hough transforms, PRIP 81, Dallas, TX, August 2-4, 1981, 526-533.

x	y	size	angle	x	y	size	angle	x	y	size	angle
12	193	13	5	15	117	46	7	21	86	21	13
28	102	22	169	37	90	31	38	37	137	20	169
44	112	48	144	53	135	23	155	53	238	27	150
91	136	20	166	96	151	15	144	98	239	127	177
132	191	31	172	134	240	27	32	149	60	10	141
152	35	47	174	158	188	36	170	163	46	24	158
168	137	40	166	181	186	40	171	185	221	41	1
191	117	42	179	195	135	51	179	196	200	28	176
202	31	12	0	207	85	20	2	212	232	39	2
227	200	20	179	230	179	40	171				

Table 1. (Number of blobs = 29)

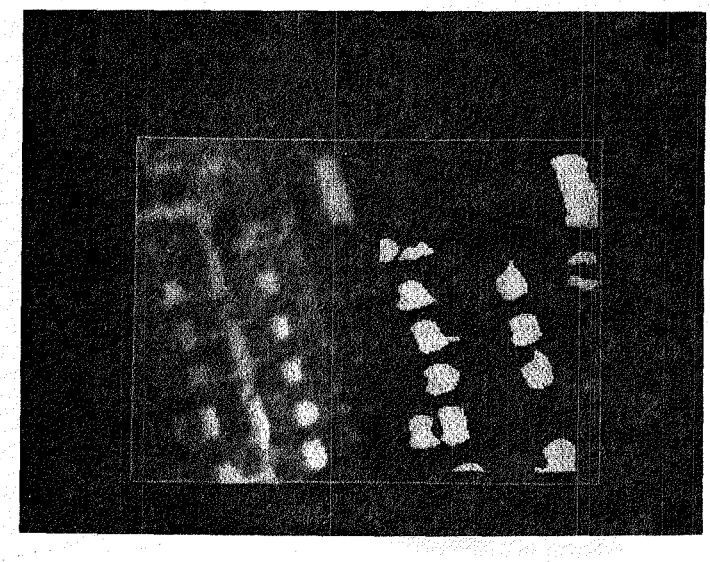
x	y	size	angle	x	y	size	angle	x	y	size	angle
28	110	113	7	60	11	17	178	61	25	17	147
64	114	13	142	78	76	37	178	78	114	14	34
86	24	37	2	103	83	37	178	108	33	42	21
126	90	39	177	131	39	39	173	156	46	45	5
160	29	36	3	173	102	46	169	179	54	10	135

Table 2. (Number of blobs = 15)



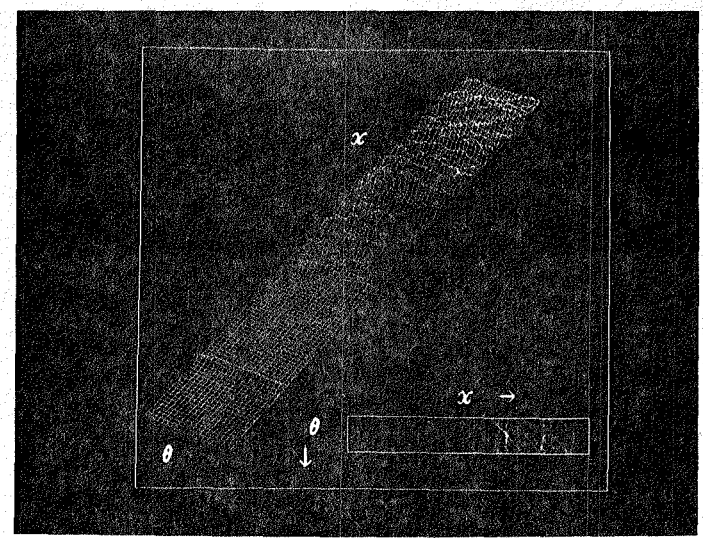
(a) (b)

Figure 1. Frame 1 (a) and extracted blobs (b).

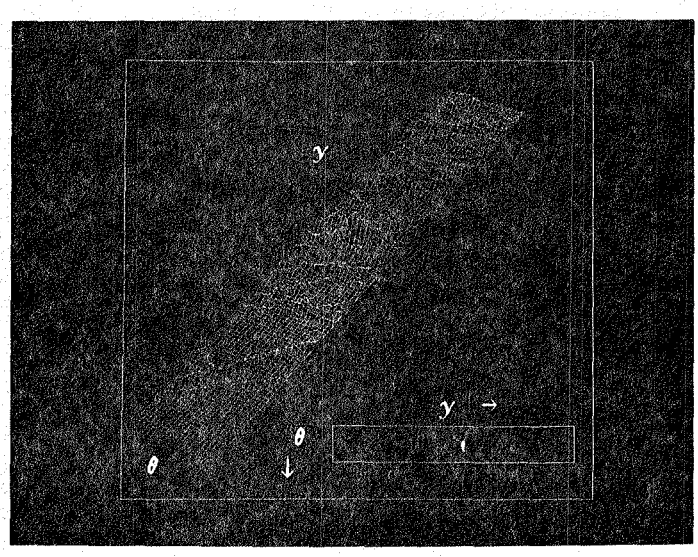


(a) (b)

Figure 2. Section of frame 2 (a) and extracted blobs (b)



(a)



(b)

Figure 3. Projected Hough transforms for (0,x)(a) and (0,y)(b).

ANALYSIS OF SUBPIXEL REGISTRATION ACCURACY

David Lavine
L.N.K. Corporation

Laveen N. Kanal
L.N.K. Corporation

Carlos A. Berenstein
University of Maryland
L.N.K. Corporation

Eric Slud
University of Maryland
L.N.K. Corporation

Charles Herman
L.N.K. Corporation

TABLE OF CONTENTS

ABSTRACT

Geometric and probabilistic models for subpixel accuracy are developed. The geometric models bound the error in offset estimation using the pixels in an observed digital straight line. One probabilistic model bounds the estimate of error offset for continuous images. The other model bounds the error for discrete images given that one is in the correct pixel.

Acknowledgement

We wish to extend great thanks to Mr. Ming Hong Li of the Beijing Institute of Aeronautics and Astronautics, Peoples Republic of China, for many helpful discussions in this work. In particular, he provided an analysis of the infinite digital line situation, independently of the work of [Do-Sm]. He also developed the algorithm for handling digital lines with points missing.

NOTATION

$\lfloor x \rfloor$ - greatest integer $\leq x$

$\lceil x \rceil$ - least integer $\geq x$

(m,n) - greatest common divisor of m and n

$L(a,b)$ - line joining points a and b

$\phi(n)$ - Euler totient function - the number of positive integers less than or equal to n which are relatively prime to n

$\mu(n)$ is the Moebius function defined as follows:

$$\mu(1) = 1;$$

if $n > 1$, let $n = p_1^{a_1},\ldots,p_K^{a_K}$ be the prime decomposition of n. Then

$$\mu(n) = (-1)^K \text{ if } a_1 = a_2 = \ldots = a_K = 1$$

$$\mu(n) = 0 \text{ otherwise}$$

Section 1.0 Introduction

The problem of aligning a sensed image to a reference image to less than a pixel accuracy has received considerable attention in recent years, but no theoretical basis for these efforts has been established. This report describes our work in the development of models for the analysis of subpixel accuracy. We have pursued several independent avenues of research in this initial study. These analyses will be combined in the coming year to provide a more complete analysis of the problem.

Two complementary approaches to the subpixel registration problem were undertaken in this study. The first approach has a deterministic geometric orientation, while the second is primarily statistical. In the first approach, we assume an approximate registration of a sensed image containing a linear feature to a reference image is available. Using the location of the observed pixels and the information that the corresponding reference feature is straight, we derive bounds on the accuracy to which the reference and sensed image can be matched. These error bounds are related to the properties of the feature, such as its length and angle. These relationships can then be used to establish criteria for the selection of good reference images. In our most restrictive model we find that subpixel accuracy is readily achievable. As we examine less restrictive models in the continuation of this work we hope to achieve more realistic bounds.

This report focusses on modeling of the subpixel registration to obtain bounds on registration accuracy and to develop model based methods.

Consequently, we generally refer to previous subpixel algorithms, only when they are relevant to the modeling and analysis problems. A previous survey of subpixel methods [Ka] ultimately led to the present study.

The current study consists of three main segments. First, we studied the registration accuracy which could be achieved by matching geometric figures, such as straight lines, between images. This work, described in Section 2, assumes the geometric figure has been extracted from the sensed image, and is known to lie in the reference image. The essence of the approach is that a slight shift in a real world edge can cause a substantial change in the digitization of that edge. We propose three progressively, more realistic models. The first model was analyzed and it was shown that a high degree of subpixel accuracy can be attained under the assumptions of the model. Future work will deal with the less restrictive forms of this model.

The second segment of our study develops bounds on subpixel registration accuracy using statistical bounds. Two cases are considered, matching of continuous images and matching of discrete images. In the continuous case we derive bounds on registration accuracy, while in the discrete case we derive bounds on subpixel accuracy given that we are on the current pixel.

The third part of our study dealt with the problem of maximum likelihood based estimation of the registration offset. Since the first two phases of the work assumed pixel registration was available, we felt it necessary to examine the credibility of this assumption. A maximum likelihood procedure was developed for estimating the location of a corner such as a field boundary in an image. Interpolation of the

correlation function did not prove to be useful in synthetic imagery from this model, though this work is in a preliminary stage.

Maximum likelihood, correlation and least squares are all used in image matching. Confusion as to the interrelationships between these methods pervades the literature. We have included a section describing work in which we establish conditions under which these methods are equivalent.

We have developed both geometric and stochastic models for subpixel accuracy. Under restrictive model assumptions, the geometric method leads to bounds on subpixel accuracy. The statistical modeling has lead to error bounds which will be examined in experimentation in the continuation of this work. There will also be a fusion of parts of the geometric and stochastic modeling. We think this initial work has provided useful models and opened up many paths for continued exploration into progressively more realistic models for subpixel accuracy.

Section 2.0 Geometric Accuracy

Matching edges in sensed and reference images can be used for registration. The degree to which the position of a real-world edge, such as a field boundary, can be located in imagery depends heavily upon ones knowledge of the scene and the sensors. Edge detectors can be used to locate reasonable candidates for edge points and then an edge can be more precisely fit using these points. Alternatively, an estimate of subpixel edge location can be formed directly from the grey levels [Hy -Ba] Hybrid approaches may also be adapted. In this section, we study the accuracy attainable using the first of these approaches, which we call the geometric accuracy approach.

Before launching into a description of our models for geometric accuracy, it is useful to consider those aspects of the registration problem we wish to capture in our models. The heart of our approach is to estimate the position of an image edge to subpixel accuracy and use this information to define a translation between the sensed and the reference image. In the ideal case, the grey levels on each side of the edge are constant off the edge pixels and the edge pixel grey levels are a simple weighted average of these two grey levels. If all grey levels are possible and the edge pixels are all known then the position of the edge can be exactly determined. Such a situation is clearly unrealistic but it serves as a starting point for approximation.

Most current methods for attaining subpixel accuracy employ some type of interpolation of the correlation fucntion. If such a method is to achieve subpixel accuracy, the digital correlation function must be able to achieve pixel accuracy. In our work, we assume pixel accuracy is avail-

able either through correlation or other methods. Thus, in the simple case of a one-dimensional shift any real world point can be determined to lie within a 3x1 pixel strip. Our results can be improved drastically if we assume we know, from registration, we are in the correct pixel, but this is a highly unrealistic assumption.

The analysis described in this section pertains to the problem of one-dimensional translations. This is not particularly restrictive since the two-dimensional problem can be easily decomposed into one-dimensional shift estimates. In the line location estimation problem, we are trying to locate a real world line $y = mx + b$ in the image. A shift $(\Delta x, \Delta y)$ between real world and image coordinates yields a line $y = m(x - \Delta x) + b + \Delta y$ in the image. This may be written as $y = mx + b + (\Delta y - m\Delta x)$ which is the original line shifted only in the y direction and by an amount $\Delta y - m\Delta x$. Our 1-d estimation procedures enable us to estimate $\Delta y - m\Delta x$. Given two lines, we can solve (possibly using least squares) for Δx and Δy separately. From this point on, we will confine ourselves to 1-d shifts.

The models described in this chapter assume a set of pixels labelled edge pixels are provided by an edge detection procedure. Three cases are considered. First, the set of edge pixels are exactly the digital edge corresponding to a line in the real world. This model is unduly restrictive since an edge which comes very near a pixel boundary can show up in the next pixel due to noise. Second, we consider a model in which the set of edge pixels given is a subset of the digital edge corresponding to the real edge. This approach is more realistic since it enables us to discard some pixels whose classification as edge pixels in in doubt.

Finally, we give a model in which some pixels lying on the digitization of the real edge are given and some incorrect pixels are given.

For the first model, in which a complete digital edge is available, a tight upper bound for the registration error as a function of the line parameters is given. Probabilistic error estimates are underway but we have not completed these calculations. For the second model, in which some pixels may be missing from the digital edge, we give a procedure which can, given any subset of a digital line, produce a tight upper bound on the registration error and the expected error. As the number of subsets of digital lines is large, a complete description of the error as a function of subset parameters is not readily available. We are currently working on analytical results to eliminate this problem. The third model has not yet been explored.

The three geometric models can be extended to include additional information such as gradient values. For the present, we decided that the additional complexities added by this information would make analysis extremely difficult. By first developing the simpler geometric models, we obtain a standard for subpixel accuracy which can provide a firm basis for such extensions. The reliability of digital edges extracted from real imagery is not considered in this report, though it is clearly important in applying the geometric accuracy results. Future work using the Landsat data base will be directed toward establishing the reliability of edge pixels.

Section 2.1 Digital Straight Line Segment Parameter Estimation

Estimation of the location parameters of a real world edge giving rise to an image edge is discussed in this section. The ideas discussed are a summary of those parts of [Do-Sm] which are useful for subpixel registration. Their basic result is a determination of all lines whose digitization is a specified chain code. In later sections, this set of lines will be used to derive error bounds on registration accuracy.

Several line digitization procedures are commonly used in graphics and image processing. Given a line segment in the upper right hand quadrant of the plane, with slope and y-intercept both between 0 and 1, we define its digitization as follows. To each intersection (a,b) between the line and a line y = a, a an integer, we associate the pixel with lower left hand corner $(a,\lfloor b \rfloor)$. (see figure 2.1). The chain code (see Fig. 2.1) of the sequence of pixels with lower left hand coordinates $(0,b_0)$, $(1,b_1)$, ..., (N,b_N) is the sequence $0_1,\ldots,0_N$ where

$$c_i = \begin{cases} 0 \text{ if } \lfloor b_i \rfloor = \lfloor b_{i-1} \rfloor \\ 1 \text{ otherwise} \end{cases}$$

The restrictions on the slope and y-intercept of the lines under consideration are made for simplicity of presentation. By symmetry the results can be extended to remove these conditions.

To determine the lines with specified chain code, it is useful to have a parameterization of the set of all chain codes of digital line segments resulting from digitizing the class of lines specified above. In [Do-Sm] the following parameterization is given. A digital line segment chain code $(C_1,\ldots C_N)$ is given by a quadruple of integers (N,p,q,s). N is the

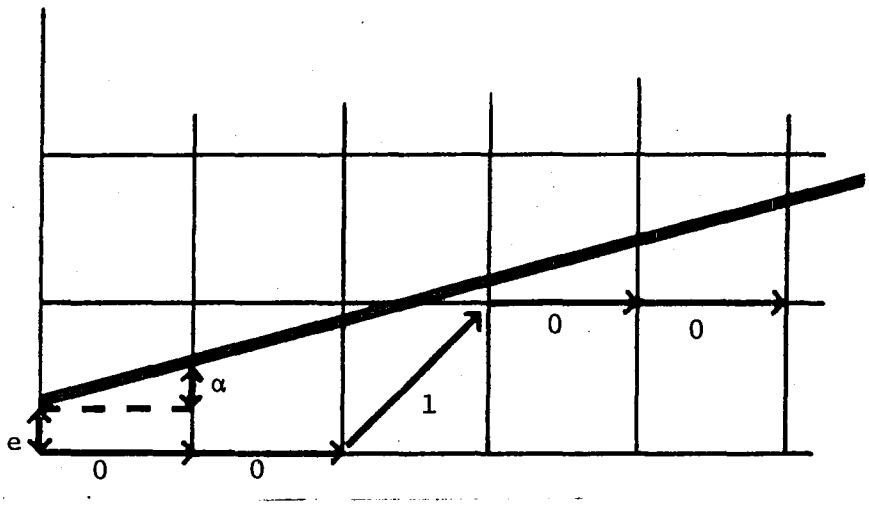


Figure 2.1 Chain code of a digital line. The digitization of the dark diagonal line has pixels with lower lefthand vertices (0,0), (1,0), (2,0), (3.1), (4.1), (5,1). The resulting chain code indicated by the arrows is 00100

length of the chain code, i.e., the number of 0's and 1's. Next, q is defined to be the smallest integer such that there exists an extension $C_{N+1},C_{N+2},\ldots$, with $C_1,C_2,C_3,\ldots$ periodic with smallest period q. Define p to be the numbers of ones in a period. The fourth parameter, s, provides a normalization of the chain code for one period. Geometrically, s may be interpreted as follows. Any chain code corresponds to a line segment with rational slope. Among all such segments, select the slope p/q with (p,q) = 1 which has the minimum q. This q is the period. The standard chain code corresponding to the first period of this chain code is the chain code of the digitization of the first q pixels of the line through the origin, y= (p/q)x. The ith element C_i, of the chain code is given by

$$C_i = \lfloor i\frac{p}{q} \rfloor - \lfloor (i-1)\frac{p}{q} \rfloor, \quad i = 1,2,\ldots a$$

The parameter s, of a code string of length N, is defined by the condition that the standard code string of p/q starts at the (s + 1)th element of the original chain code. Given the parameters N,q,p,s of a codestring, the ith element of the original codestring can be obtained by

$$C_i - \lfloor (i-s)\frac{p}{q} \rfloor - \lfloor (i-s-1)\frac{p}{q} \rfloor, \quad i = 1,2,\ldots,N$$

The parameters satisfy the constraints $0 \leq p \leq q \leq N$ and $0 \leq s \leq q-1$. A point which will be particularly important for the registration problem is that there are other constraints on the parameters other than the above inequalities. These additional constraints, described in Section 2.4 appear to be rather complicated. Our interest in these matters stems from the need to enumerate the digital lines satisfying various conditions. If not for these messy constraints, the enumeration problems would often be straightforward. Without these additional constraints for fixed N, we would obtain all digital line segments of length N by independently

varying s,p,q subject to the constraints $0 \leq p \leq q \leq N$ and $0 \leq s \leq q - 1$.

We now give an example of the computation of the parameters for a chain code.

EXAMPLE. chain code 10010100

N = 8: there are 8 digits in the code

q = 5: the above code is part of the infinite code
. . . 100101001010010 . . .

p = 2: the number of 1's in the period 10010 is 2

s = 1: the standard codestring of 2/5 is 00101. The standard codestring starts at the 2nd element of the chain code. Hence s = 1.

The primary result of [Do-Sm] is a description of the set of all lines whose digitization over the x-interval [0,N] is a set of pixels specified by a chain code. This result is of great importance for our registration accuracy results since it provides a hold on the errors which may arise by approximating the true edge by a feasible edge. The set of lines is described by a quadrilateral in the (e,α)-plane where e is the y-intercept of a line and α is the slope. The proof of the following formula has not yet appeared [Do] so we shall only present the results, which is all we need for the current work. Define functions F and L by:

$$F(s) = s - \lfloor s/q \rfloor q$$

and

$$L(s) = s + \lfloor (N - s)/q \rfloor q$$

and let ℓ be defined by the equation :

$$1 + \lfloor \ell \tfrac{p}{q} \rfloor - \ell \tfrac{p}{q} = 1/q \text{ and } 0 < \ell < q.$$

The set of feasible lines is a quadrilateral in (e,α)-space with vertices A, B, C, D given by:

$$A = (\lfloor F(s)\tfrac{p}{q} \rfloor - F(s)\tfrac{p+}{q+}, \tfrac{p+}{q+})$$

$$B = (\lfloor F(s)\tfrac{p}{q} \rfloor - F(s)\tfrac{p}{q},\ \tfrac{p}{q})$$

$$C = (1 + \lceil F(s + \ell)\tfrac{p}{q} \rceil - F(s + \ell)\tfrac{p}{q},\ \tfrac{p}{q})$$

$$D = (1 + \lceil F(s + \ell)\tfrac{p}{q} \rceil - F(s + \ell)\tfrac{p-}{q-},\ \tfrac{p-}{q-})$$

where

$$q+ = L(s + \ell) - F(s),\ p+ = (pq_+ + 1)/q$$

$$q- = L(s) - F(s + \ell),\ p- = (pq_- - 1)/q$$

The above expressions for the vertices of the feasible quadrilateral will be discussed in greater detail in later sections. A generalization of the above result to subsets of a digital line will be presented, though the manner in which it can be reduced to the above formula is unclear.

Section 2.2 Feasible Region Shape

The description of the set of all lines whose digitization is a specified chain code of a straight line segment will now be used to obtain a worst-case bound on the subpixel accuracy with which we can locate a point in the image. We will show that given a period q chain code of the digitization of a straight line segment, there exists a real number x such that the total spread on y-values at the point x of all line segments with the given chain code is 1/q (see Fig. 2.2). Thus by selecting the midpoint of this set of (x,y)'s we have estimated the position of a point on the line to within an error of 1/(2q). This provides our error bound. In Section 2.5, we will examine the distribution of 1/(2q) corresponding to a probability distribution on lines.

To see the correctness of the 1/q spread, we first observe that lines B and C of the feasible region (Sec.2.1) are parallel each with slope p/q. We show that their vertical separation is 1/q. These lines may be thought of as providing a channel where we can find x values where the spread is 1/q. Next, the relationship between the location of the feasible region vertices in (e,α)-space and the location of points on possible real line segments with the appropriate digitization is established. This will yield a polyhedral region in (x,y)-space which is the union of all feasible lines. Finally, we show that there exists a real number x such that the extent of the feasible region over x is determined only by the lines B and C, hence is of width 1/q.

The proof that B and C are 1/q units apart vertically is now given. In the case of the infinite digital line, the calculation that the spread is 1/q everywhere is straightforward. By passing to the finite case, we

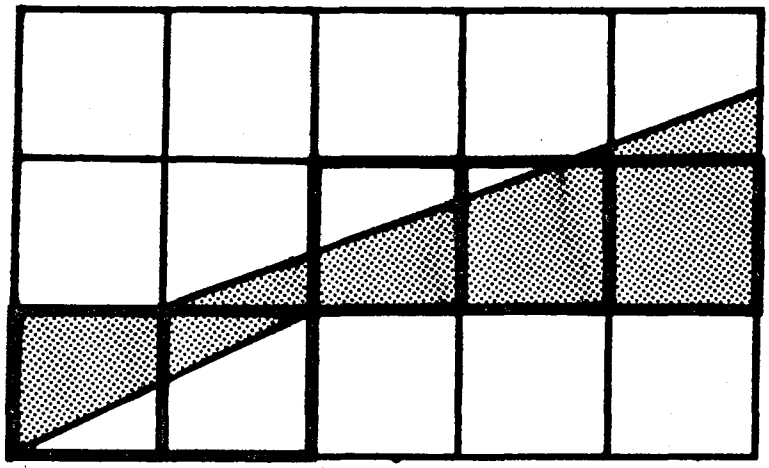


Figure 2.2 Feasible region for a digital line. The digital line consisting of those pixels with darkened boundaries has the shaded area as its feasible region.

introduce boundary effects which cause the spread to be greater near the ends of the chain code, but the following proposition shows that at least one point of the 1/q width channel is preserved.

PROP. 2.1 Using the notation of Sec. 2.1, Let B and C be the vertices of the feasible region for a chain code with parameters (N,q,p,s) corresponding to a straight line segment. Then the difference of the y intercepts of the lines corresponding to C and B is 1/q.

PROOF. Let W denote the difference in the y-intercepts. Then W is given by

$$W = 1 + \lfloor F(s + \ell)\frac{p}{q} \rfloor - F(s + \ell)\frac{p}{q} - \lceil F(s)\frac{p}{q} \rceil + F(s)\frac{p}{q}$$

By definition,

$$F(s + \ell) = s + \ell - \lfloor (s + \ell)/q \rfloor q$$

Since $0 \leq s \leq q - 1$ and $0 < \ell < q$, we have $0 < s + \ell < 2q$

Thus $\lfloor (s + \ell)/q \rfloor = 0$ of $s + \ell < q$

1 if $s + \ell \geq q$

We examine these two cases separately.

Case (1): $s + \ell < q$

$$F(s + \ell) = s + \ell$$

Thus: $W = 1 + \lfloor (s + \ell)p/q \rfloor - (s + \ell)p/q - \lceil sp/q \rceil + sp/q$

(As an aside, we note that if $s = 0$, i.e., we normalize the position of the chain code, then $W = 1/q$ follows immediately from the definition of ℓ.) To simplify the expression for W, we recall the definition of ℓ

$$1 + \lfloor \ell p/q \rfloor - \ell p/q = 1/q$$

$$\ell p/q = 1 + \lfloor \ell p/q \rfloor - 1/q$$

$$(s + \ell)p/q = sp/q + \ell p/q$$

$$= sp/q + 1 + \lfloor \ell p/q \rfloor - 1/q$$

$$\lfloor (s + \ell)p/q \rfloor = 1 + \lfloor \ell p/q \rfloor + \lfloor sp/q - 1/q \rfloor$$

$$\lfloor (s + \ell)p/q \rfloor - (s + \ell)p/q = \lfloor sp/q - 1/q \rfloor - sp/q + 1/q$$

Hence $W = 1 + \lfloor sp/q - 1/q \rfloor - sp/q + 1/q - \lceil sp/q \rceil + sp/q$

$$W = 1 + \lfloor sp/q - 1/q \rfloor - \lceil sp/q \rceil + 1/q$$

To complete our evaluation of W, we consider two subcases.

Subcase (1): sp/q is not an integer, In this case, $\lceil sp/q \rceil = \lfloor sp/q \rfloor + 1$. Thus substituting into W, we have

$$W = 1 + \lfloor sp/q - 1/q \rfloor + 1/q - \lfloor sp/q \rfloor - 1$$

$$= \lfloor sp/q - 1/q \rfloor - \lfloor sp/q \rfloor + 1/q$$

If $sp < q$, then $(sp - 1)/q < 1$, $sp/q < 1$, so we get $W = 1/q$

The situation where sp/q is an integer is considered in Subcase (2), so we may assume $sp > q$, sp/q is not an integer. Hence, there exists an integer $1 \leq r \leq q$, and an integer $k > 0$ such that

$$sp = kq + r$$

$$sp/q = k + r/q$$

Thus $\lfloor sp/q \rfloor = k$

$$sp/q - 1/q = k + (r - 1)/q$$

Since $r - 1 < q$, we see that

$$\lfloor sp/q - 1/q \rfloor = k$$

Thus $\lfloor sp/q - 1/q \rfloor - \lfloor sp/q \rfloor = 0$

Hence $W = 1/q$

Subcase (2): sp/q is an integer

We have $\lfloor sp/q \rfloor = \lceil sp/q \rceil$. Then

$$W = 1 + \lfloor sp/q - 1/q \rfloor + 1/q - \lfloor sp/q \rfloor$$

Since sp/q is integer, $\lfloor sp/q - 1/q \rfloor = sp/q - 1 = \lfloor sp/q \rfloor - 1$

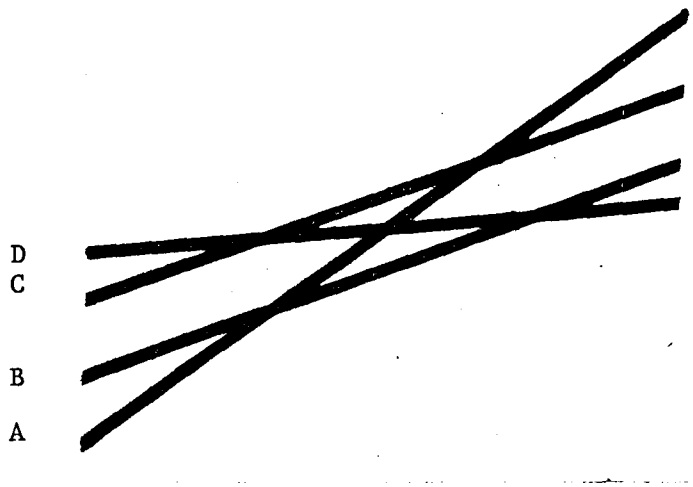


Figure 2.3 Intersections for the feasible region. The four boundary lines A, B, C, and D of a feasible region are shown. The intersection of A and D always lies between the parallel lines B and C. These lines in the x,y space correspond to the vertices A,B,C,D of the feasible quadrilateral in the (e,α) parameter space.

Thus $W = 1/q$

Case (2): $s + \ell \geq q$

Using $\lfloor (s + \ell)/q \rfloor = 1$, and $F(s + \ell) = s + \ell - q$, we get

$$W = 1 + \lfloor (s + \ell - q)p/q \rfloor - (s + \ell - q)p/q - \lceil sp/q \rceil + sp/q$$

$$\lfloor (s + \ell - q)p/q \rfloor = \lfloor (s + \ell)p/q \rfloor - p$$

$$(s + \ell - q)p/q = (s + \ell)p/q - p$$

Thus $W = 1 + \lfloor (s + \ell)p/q \rfloor - (s + \ell)p/q - \lceil sp/q \rceil + sp/q$

At this point, the arguments of Case (1) can be applied and we get $W = 1/q$.

We have established that lines B and C are separated by a vertical distance $1/q$. Next we show that, given an x value and the four lines A, B, C, D evaluated at x, the part of the feasible region lying over x is the convex hull of the four values.

PROP. 2.2 Let L be a digital line of length N with vertices A, B, C, D for the corresponding feasible region. Let A, B, C, D correspond to the equation $y = m_i x + b_i$, $i = 1,..,4$. For any $x_0 \in [0,N]$, set $M = \max\{m_i x_0 + b \mid i = 1,...,4\}$ and $P = \min\{m_i x_0 + b_i \mid i = 1,...4\}$. Then a point (x_0,y) lies on a line segment with digitization L if and only if $P \leq y \leq M$.

PROOF. Let x_0 $[0,N]$ and let $y = mx + b$ be the line corresponding to any point in the quadrilateral given by A, B, C, D. Then, since the quadrilateral is the convex hull of the set A, B, C, D, there exists real numbers t_1,t_2,t_3,t_4 such that the following conditions hold:

1) $0 \leq t_i \leq 1$ for $i = 1,...,4$

2) $\sum_{i=1}^{4} t_i = 1$

3) $\sum_{i=1}^{4} (m_i x + b_i)t_i = mx + b$ for each i

Thus $mx_0 + b = \sum_{i=1}^{4} (m_i x_0 + b_i)t_i$

$\leq M \sum_{i=1}^{4} t_i$

$= M$

Similarly we have $mx_0 + b \geq P$. Thus any feasible point (x_0,y) satisfies $P \leq y \leq M$. Now let $y_0 \varepsilon [P,M]$. If $y_0 = m_i x + b_i$ for some i then y obviously lies on a feasible line. If y_0 is not one of these four values then there exists i,j such that $m_i x_0 + b_i \leq y_0 \leq m_j x_0 + b_j$. Hence there exist t_i, t_j such that $t_i + t_j = 1$ and $0 \leq t_i, t_j \leq 1$. Setting the other two t's to zero we have a quadruple $t_1, \ldots, t_4$ such that $y_0 = \sum_{i=1}^{4} (m_i x_0 + b_i)t_i$. Thus (x_0,y_0)lies on the feasible line given by

$$y = (t_i m_i + t_j m_j)x + (t_i b_i + t_j b_j).$$

The next step in finding a point x_0 at which the feasible region has height 1/q is to determine the way in which the lines A and D intersect the parallel lines B and C. We will show there is an interval $[a,b] \subset [0,N]$ such that lines A and D lie between lines B and C over the interval [a,b]. To do this we establish the following facts (see Fig. 2.3):

Let $I(\cdot,\cdot)$ denote the x-coordinate of the intersection of the two arguments,

1) The y-intercept of A is less than or equal to the y-intercept of D

2) The y-intercept of C is less than or equal to the y-intercept of D

3) $I(D,C) \leq I(A,C)$

4) $I(A,B) \leq I(D,B)$

5) $I(D,C) \leq N,\ I(A,B) \leq N$

From the diagram, we can see that selecting $a = \max(I(A,B),I(D,C))$ and $b = \min(I(A,C),I(B,D))$, the feasible region has height $1/q$ on the interval $[a,b]$.

LEMMA 2.3 The y-intercept of A is less than or equal to the y-intercept of B.

PROOF: Denoting the y-intercepts by Y_A and Y_B we have

$$Y_B - Y_A = \lceil F(s)p/q \rceil - F(s)p/q - \lceil F(s)p/q \rceil + F(s)p_+/q_+$$
$$= F(s)(p_+/q_+ - p/q)$$

Since $F(s) = s \geq 0$, we are done if we show $p_+/q_+ - p/q > 0$.

By the definition of p_+, q_+,

$$p_+/q_+ - p/q = (pq_+ + 1)/(qq_+) - p/q$$
$$= p/q + 1/(qq_+) - p/q$$
$$= 1/(qq_+)$$

It suffices to show $q_+ > 0$. By definition,

$$q_+ = L(s + \ell) - F(s)$$
$$= s + \ell + \lfloor (N - (s + \ell)/q \rfloor q - s$$
$$= \ell + \lfloor N - (s + \ell)/q \rfloor q$$

Since $\ell > 0$, we have $q_+ > 0$.

LEMMA 2.4 The y-intercept of D is greater than or equal to the y-intercept of C.

PROOF: Denoting the y-intercepts by Y_C and Y_D we have, using the same type of arguments in the previous lemma

$$Y_D - Y_C = F(s + \ell)(p/q - p_-/q_-)$$
$$F(s + \ell) = s + \ell - \left\lfloor \frac{s + \ell}{q} \right\rfloor q$$
$$= \begin{cases} s + \ell \text{ if } s + \ell < q \\ s + \ell - q \text{ if } s + \ell \geq q \end{cases}$$

In either case $F(s + \ell) \geq 0$

$$p/q - p_-/q_- = 1/(qq_-)$$

We are done if we can show $q_- \geq 0$. If $q_- < 0$ then $p/q < p_-/q_-$. This implies the slope and y-intercept of D are greater than the slope and y-intercept of C. Hence, over the interval [0,N], the line C lies entirely below the line D and entirely above the line B. Thus there is a whole neighborhood around the point C in (e,α) space which lies in the feasible region contradicting the fact that C is on the boundary of the feasible region. We conclude that $q_- > 0$. Notice $q_- = 0$ is precluded by the form of the slope for D. Since $q_- > 0$, we see that $Y_D - Y_C > 0$.

LEMMA 2.5 $I(D,C) \leq I(A,C)$

PROOF: Given lines $y = m_1x + b_1$ and $y = m_2x + b_2$, their intersection occurs at $x = (b_1 - b_2)/(m_2 - m_1)$.

$$I(A,C) = \frac{1 + \lfloor F(s + \ell)p/q \rfloor - F(s + \ell)p/q - \lceil F(s)p/q \rceil + F(s)p_+/q_+}{p_+/q_+ - p/q}$$

We consider two cases:

Case (1): $s + \ell < q$

In this case $F(s + \ell) = s + \ell$. Recalling $p_+/q_+ - p/q = 1/(qq_+)$ we have

$$I(A,C) = qq_+(1 + \lfloor (s + \ell)p/q \rfloor - (s + \ell)p/q - \lceil sp/q \rceil + sp_+/q_+)$$

Subcase (1): sp/q not an integer

$$\lceil sp/q \rceil = \lfloor sp/q \rfloor + 1$$

$$I(A,C) = qq_+(1 + \lfloor (s + \ell)p/q \rfloor - (s + \ell)p/q + sp_+/q_+ - \lceil sp/q \rceil)$$

By the proof of Lemma 2.1 we have

$$I(A,C) = qq_+(\lfloor sp/q - \tfrac{1}{q} \rfloor - sp/q + 1/q + sp/q + sp/q_+ - \lfloor sp/q \rfloor)$$

$$= qq_+(1/q + s/qq_+) \quad \text{by the proof of Lemma}$$

$$= q_+ + s$$

Subcase (2): sp/q an integer

Once again, using the proof of Lemma 2.1 we obtain

$$I(A,C) = qq_+(1 + \lfloor (s + \ell)p/q \rfloor - (s + \ell)p/q - sp/q + sp/q + s/(qq_+))$$

$$= qq_+(1 + \lfloor (s + \ell)p/q \rfloor - (s + \ell)p/q + s/(qq_+))$$

$$= qq_+(1 + \lfloor \ell p/q \rfloor - \ell p/q + s/(qq_+))$$

$$= qq_+(1/q + s/(qq_+))$$

$$= q_+ + s$$

We now compute $I(D,C)$

$$I(D,C) = \frac{F(s + \ell)(p_-/q_- - p/q)}{p_-/q_- - p/q}$$

$$= F(s + \ell)$$

$$= s + \ell - \lfloor (s + \ell)/q \rfloor q$$

$$= s + \ell \quad \text{since } s + \ell < q \text{ in Case (1)}$$

$$I(A,D) = q_+ + s$$

$$= (s + \ell) + \lfloor (N - (s + \ell))/q \rfloor q + s$$

$$\geq s + \ell$$

$$= I(D,C)$$

Case (2): $s + \ell > q$

In this case $F(s + \ell) = s + \ell - \lfloor \frac{(s + \ell)}{q} \rfloor q$

$$I(A,C) = qq_+(1 + \lfloor (s + \ell)p/q \rfloor - \lfloor (s + \ell)/q \rfloor p - (s + \ell)p/q$$

$$+ \lfloor (s + \ell)/q \rfloor p - \lceil F(s)p/q \rceil + F(s)p_+/q_+)$$

After cancelling the terms $\neq \lfloor (s + \ell)/q \rfloor p$, we are reduced to Case

(1) and we obtain $I(A,C) = q_+ + s$. As in Case (1)

$$I(D,C) = F(s + \ell)$$
$$= s + \ell - \lfloor (s + \ell)/q \rfloor q$$
$$\geq s + \ell$$

From the proof of Case (1), we had $I(A,D) \geq s + \ell$

Thus $I(A,D) = I(D,C)$

The proof that $I(A,B) \leq I(D,B)$ follows the lines of the above proofs and is omitted. For possible application in later work we give the intersections

$$I(A,B) = s$$
$$I(D,B) = s + \ell + q_- \text{ if } s + \ell < q$$
$$= s + \ell + q_- - q \text{ if } s + \ell \geq q$$

The intersections of A and D with B and C have been computed explicitly and we can see that

$$I(D,C) \leq N$$

and $$I(A,B) \leq N.$$

Thus by our earlier remarks we are guaranteed of the existence of a real number $0 \leq x \leq N$ such that the feasible region over x has height $1/q$.

From the results of this section we may conclude that given a digital line with period q in the sensed image such that the underlying real edge has slope between zero and one, then we can determine the vertical offset between sensed and reference images to an accuracy of $1/2q$ pixels.

Section 2.3 Infinite Digital Lines

The feasible region for infinite digital lines is easily computed using the results of Section 2.2. This analysis is divided into two parts. For any infinite digital line of period q, we show the channel consists of two parallel lines, which are a vertical distance 1/q apart. Thus since the channel extends over the whole x-axis, there is no flaring at the end as in the finite case. If the infinite digital line is aperiodic, then we show the channel extends over the whole x-axis, but consists of a single line. Thus the maximum error is 1/2q of the digital line if the digital line has period q and zero if the digital line is aperiodic. The aperiodic infinite digital lines are precisely those infinite digital lines which are the digitizations of lines with irrational slope. Since the irrationals are a set of measure one in the unit interval, using the uniform probability measure, we see that the error is zero with probability one for infinite digital lines.

Before considering the periodic and aperiodic lines separately, we note that any two infinite lines with the same digitization are parallel. Let $y = mx + b$ and $y = nx + c$ be two lines. Then the difference, $h(x)$, in the y values of these lines at x is given by $h(x) = (m-n)x + (b-c)$. If m and n are not equal then there exists a $K>0$ such that $|h(x)|>1$ for all x such that $|x|>K$. Thus the two lines cannot have the same digitization.

We now consider the case of infinite digital lines of period q. By the feasible region description in Section 2.1, the lines corresponding to the vertices, A,B,C, and D of the feasible region in (e,α) space have

slopes p_-/q_-, p/q, p_+/q_+. Fixing p,q, and s and letting N go to infinite, we see the above result on the slopes of infinite lines having the same digitization imply p_-/q_- and p_+/q_+ must approach p/q. Inserting these limits into the formulas for the vertices A and D, we see that, in the limit A=B and C=D. We have shown in Section 2.2 that B and C are a vertical distance 1/q apart. This establishes the result for the infinite periodic digital line.

The infinite aperiodic line requries a different approach. We first cite a version of a classical result [Wa] on lines with irrational slope. Let $f(x) = mx + b$ be a line with m irrational. Then the set $\{mx + b - \lfloor mx + b \rfloor : x \text{ is an integer}\}$ is dense in the unit interval. It has already been shown that two lines with the same digitization have the same slopes and can only vary in their y-intercepts. Let $\varepsilon>0$ be given. Then the digitization, L, of the line $y = mx + b$ (m irrational) is aperiodic so there exists integers K_1 and K_2 such that $mK_1 + b - \lfloor mK_1 + b \rfloor < t$ and $mK_2 + b - \lfloor mK_2 + b \rfloor > 1 - \varepsilon$. Thus decreasing b by more than ε would change the digitization at K_1 and increasing b by more than ε would change the digitization at K_2. Thus for any $\varepsilon>0$, we cannot change b by more than ε without changing the digitization. Hence b is fixed. Since m is also fixed, the channel is the single line $y=mx + b$

Section 2.4 Invariant Line Measure

A probabilistic analysis of geometric accuracy requires a probability distribution in the fundamental objects, the lines. It is tempting to place a uniform distribution on the coefficients of the lines represented in some parametric form. Unfortunately, there is no canonical parametrization and the measure will not be uniform with respect to other parametrizations. A customary escape from this quandry is to impose some parametrization independent conditions which single out a probability measure. In geometric probability problems, one generally assumes the measure is invariant under translation and rotation of the geometric figures, in our case the lines. This uniquely determines a coordinate system, the (ρ,θ) polar coordinates of a line, in which the distribution is uniform with respect to the parameters. To avoid the problem of taking a uniform distribution on an unbounded set, we restrict the parameters to lie in a bounded set. The measure of this set is to be normalized to one. The above measure provides a probability measure on lines whose digitizations belong to any specified set of digital lines. This induces a probability measure on digital lines which can be used to perform a probabilistic analysis of geometric accuracy.

Section 2.5 Digital Line-Probabilistic Analysis

A worst case bound on registration accuracy using a digital edge was developed in Section 2.2. More realistic error information can be obtained using probability. In this section we consider the question of obtaining probabilistic information on the registration error assuming the real world edge giving rise to the digital edge is generated by a natural distribution on edges. We have procedures for estimating these probabilities, but due to the considerable computational cost involved in evaluating these in special cases, we prefer to first seek analytical simplifications.

Many probabilistic questions pertinent to the geometric accuracy question can be formulated. Several of the most basic are

1) Given a maximum allowed registration error, what is the probability that the actual error will not exceed this?
2) What is the expected value and the variance of the registration error?
3) Given a maximum allowed registration error and a maximum allowed probability of error find the largest region of lines (in some sense) such that lines coming from this region will result in an acceptable size error an acceptable percentage of the time?

We now turn to an analysis of the first question. We wish to determine, for any acceptable error level in the estimated offset between sensed and reference image, what is the probability that a random edge will result in a digitization which permits estimation to less than that error level. Though a formula for these probabilities as a function of digital line length is not available, a procedure for calculating these probabilities

for any given line length, N, is described and results for the case N = 10 are presented . In addition we present asymptotic upper bounds on the error.

The basic approach to computing the error probabilities is quite simple. A probability density function is given on the set, A, of all lines with slope between 0 and 1, going through the pixel with lower left vertex (0,0). Since a line has only one chain code, the sets of lines with different chain codes gives a partition of the set A. Hence the density on lines induces a density on chain codes. For a chain code with period q, the maximum error is 1/2q as was shown in Section 2.2. Thus for any specified error h, we must calculate the probability of the following set, B, of line chain codes.

$$B = \{(N,q,p,s): \ 1/2q < h\}$$

The set of all linear chain codes of length N can be enumerated. For each chain code in B, the corresponding feasible quadrilateral can be calculated as in Section 2.1. The density function on lines can then be integrated over the quadrilateral and the sum of these integrals over all members in B computed. This sum yields the desired probability. A program to perform these computations is under development.

The problem of enumerating linear chain codes was discussed in [Ro-We] where an algorithm for generating the set of linear chain codes was presented. We have not found any estimates in the literature on the number of chain codes joining two points. Since we are initially dealing with very short lines (e.g. length 10) we have taken a naive but rapidly implementable approach to the problem of line generation. First, generate a set of real lines whose digitizations are guaranteed to include all digital lines of specified

length N = 10 and slope between 0 and 1. Next digitize these lines and finally remove duplicates.

The set of all lines of the form $y = (p/q)x + m/q$, $0 \leq p \leq q$, $(p,q) = 1$, $m = 0, \ldots, q - 1$, $q = 1, \ldots, N$ together with the line $y = x$ gives rise to all digital lines of length N with slope between zero and one and going through the pixel with lower-left hand coordinate (0,0). This follows from the result proved in Section 2.2 that a digital line segment with parameters (N,p,q,s) can be extended to an infinite digital line which is a digitization of a line of slope p/q. Thus the digital lines (N,p,q,s) will be generated, if we generate all lines, $y = (p/q)x + r$, r real, $0 \leq r < 1$. As r increases from zero to one the chain code of the line can change only when the line passes through a lattice point. Let (v,w) be any lattice point through which a line of the form $y = (p/q)x + r$ passes. Then the height of the line changes by an amount rp/q as x goes from zero to r. Since the line goes through (v,w), the height at x = 0 must be $w = rp/q$. Rewriting this as $(wq - vp)/q$ and noting that the height must be between zero and one and that $wq - vp$ is an integer we see that $r = m/q$ where $0 \leq m < q$.

An upper bound on the number of chain codes of line q with specified starting pixel and slope between zero and one can be obtained using the fact that all lines of the form $y = (p/q)x + m/q$, $(p,q) = 1$, $0 \leq p \leq q$, $0 \leq m \leq q - 1$ give rise to all chain codes. Using the number-theoretic function, $\emptyset(q)$, given by

$\emptyset(q)$ = number of integers $\leq q$ and relatively prime to q,

we now derive an upper bound, $L^*(N)$, on the number of digital lines as a function of the length of the chain code. It is easily seen that

$$L^*(N) = 1 + \sum_{q=1}^{N} q\phi(q),$$

which is obtained by counting the number of lines $y = p/qx + m/q$ described above. Unexpectedly, this is not the same as the number of distinct digital lines. This is due to the fact that, when q is sufficiently close to N, a line of the form $y = (p/q)x + m/q$ can give rise to a line of period less than q. In fact, for each $q > N/2$, there are lines of the form $y = (p/q)x + m/q$ which give rise to a digital line of period strictly less than q. For example, consider the line

$$y = (1/3)x + 1/3$$

This has a chain code of length 3 given by 010 which has period 2 while $y = (1/3)x$ and $y = (1/3)x + 2/3$ have chain codes 001 and 100 respectively, each of period three. More generally, for chain codes of length N, of the m possible chain codes arising from lines of the form $y = (1/N)x + m/N$, only the case $m = 0$ and $m = N - 1$ have period N. To see this we note that the chain codes of the two cases are

m	chain code
0	00.....01
N-1	100.....0

Any other value of m shifts the one so the chain code has 0's on both ends. Any chain code with the same digit at both ends automatically has period less than N. Using the same principle, given any $q \geq N/2 + 1$, there exists lines of the form $(1/q)x + m/q$ which have chain codes of period less than q. The total number of lines of this form which have period q is $N - q + 2$. The situation is considerably more complicated when $p \neq 1$. We can show

using the above principle, that the function $L_*(N)$ is a lower bound on the number of lines, where

$$L_*(N) = 1 + \sum_{q=1}^{\lfloor N/2 \rfloor + 1} q\phi(q) + \sum_{q=\lfloor N/2 \rfloor + 2}^{N} (N - q + 1)\phi(q)$$

For N = 10, the true number of digital lines, L(10) is 136, $L_*(10) = 102$ and $L^*(10) = 218$

We have derived an upper bound and a lower bound for the number of lines. Using L* and L_*, we can develop asymptotic upper and lower bounds respectively for the expectation of the maximum registration error per chain code.

We now show that L_* is actually a lower bound on the number of lines.

PROP. 2.6 $L_*(N)$ is a lower bound for the number of digital lines of length N with $0 \leq p/q \leq 1$

PROOF: Clearly the period of a chain code is bigger or equal to that of a subchain code. Consequently, given a real line which is digitized over a segment of length N, the period doesn't diminish when we extend the interval. Thus the period of $y = (p/q)x$ is q when $(p,q) = 1$ and $q \leq N$. Recall that all digital lines (N,q,p,s) (even those of period < q) are generated by digitizing lines of the form $y = (p/q)x + m/q$ where $0 \leq p \leq q$, $(p,q) = 1$, $0 \leq m < q$. Changing m produces a permutation of the chain code within the first q elements, the second q elements, etc. The line (N,q,p,s) has the standard chain code starting at the (s + 1)st place. Hence, as long as $s + q \leq N$, one gets a full standard chain code as a subchain of the original chain code. Thus the original code has period at least q and thus exactly q. Consequently one gets for

a given N, q and for each p there are at least min(q,N-q+1) chain codes of period q. This follows from the observation that as long as $s + q \leq N$, a standard chain code is present in the chain code.

When $2q - 1 \leq N$ then all possible s satisfy $s + q \leq N$ so shifting (N,q,p,0) by modifying s, all lines have period q. For $2q - 1 \leq N$, one obtains the count

$$1 + \sum_{q=1}^{2q \leq N+1} q\phi(q)$$

for these lines. This gives the first two terms in the definition of L_*. When $2q > N + 1$ we get at least $N - q + 1$ lines of period q. This contributes

$$\sum_{q \geq (N/2)+1}^{N} (N - q + 1)\phi(q)$$

lines which is the last term in L_*.

We know that L_* is not a sharp lower bound. On the other hand, when $p = 1$ and $2q \geq N + 1$ then a sharp lower bound is $N - q + 2$. Furthermore, the fact that q_- is positive provides a necessary constraint on the possible s for a given N, q, and p:

$$s + \ell < q \text{ implies } N - s \geq q$$

where ℓ is given in Section 2.1 and satisfies

$$0 < \ell < q \quad \ell p = -1 (\mathrm{mod} q).$$

We don't know whether or not this last condition is sufficient to determine the possible s.

A form of expected error will now be defined. Let C(N,q) denote the number of digital lines of length N and period q, and let L(N) denote the

total number of lines of length N given by

$$L(N) = \sum_{q=1}^{N} C(N,q).$$

Define $S(N) = 1 + \sum_{q=2}^{N} C(N,q)/q$. The expected value of the channel width (i.e., twice the maximum error in our estimation), E(N), is given by

$$E(N) = S(N)/L(N)$$

This expected value is with respect to a distribution on digital lines on which all lines are equally likely rather than using the invariant probability measure on real lines. The invariant one is difficult to handle in evaluating the probability of the set of feasible lines corresponding to a chain code. Preliminary computations indicate that all digital lines have similar probabilities with respect to the invariant measure. For fixed N, the exact probabilities using the invariant measure can be done exactly. To get a rough estimate of the probabilities, the invariant measure can be replaced by uniform measure.

The expected maximum estimation error can be computed asymptotically.

PROP. 2.7 Up to an error term $O(\log N/N^2)$, the following holds:

$$\frac{5}{4N} \leq E(N) \leq \frac{7}{3N}$$

PROOF: We now compute asymptotic formulas for $L^*(N)$ and $L_*(N)$. Recall that if μ is the Moebius function [Ha-Wr] then

$$\phi(q) = q \sum_{d|q} \mu(d)/d$$

From [HW] we also obtain

$$\Phi(N) \equiv \sum_{j=1}^{N} \phi(j) = \frac{3N^2}{\pi^2} + O(N\log N)$$

For any N,

$$L^*(N) = 1 + \sum_{q=1}^{N} q\phi(q) = 1 + \sum_{q=1}^{N} q^2 \sum_{d|q} \mu(d)/d$$

We now write $q = dd'$ and substitute in the last term:

$$L^*(N) = 1 + \sum_{dd' \leq N} d^2 (d')^2 \mu(d)/d$$

$$= 1 + \sum_{d=1}^{N} d\mu(d) \sum_{d' \leq N/d} (d')^2$$

The term $\sum_{d' \leq N/d} (d')^2 = 1/3(N/d)^3 + O(N^2/d^2)$. Inserting this in $L^*(N)$, we obtain

$$L^*(N) = \frac{1}{3}N^3 \sum_{d=1}^{N} \mu(d)/d^2 + O(N^2 \log N).$$

Note we have used

$$\left| \sum_{d=1}^{N} d\mu(d) N^2/d^2 \right| \leq N^2 \sum_{d=1}^{N} 1/d = O(N^2 \log N)$$

But $\sum_{1}^{\infty} \mu(d)/d^2 = 6/\pi^2$ [HW].

Hence $\sum_{1}^{N} \mu(d)/d^2 = 6/\pi^2 + O(1/N)$. Substituting this into $L^*(N)$, we get

$$L^*(N) = 2N^3/\pi^2 + O(N^2 \log N).$$

We now get an asymptotic formula for $L_*(N)$

$$L_*(N) = 1 + \sum_{1}^{(N/2)+1} q\phi(q) + \sum_{(N/2)+2}^{N} (N + 1 - q)\phi(q)$$

Using the forumla for $L^*(N/2)$, we obtain

$$L_*(N) = \frac{N^3}{4\pi^2} + O(N^2 \log N) + (N + 1) \sum_{(N/2)+2}^{N} \phi(q) - \sum_{(N/2)+2}^{N} q\phi(q)$$

But $\sum_{(N/2)+2}^{N} \phi(q) = \Phi(N) - \Phi(\frac{N}{2} + 1)$

$$= \frac{3N^2}{\pi^2} - \frac{3N^2}{4\pi^2} + O(N\log N)$$

$$= \frac{9N^2}{4\pi^2} + O(N\log N)$$

Using $\sum_{(N/2)+2}^{N} q\,\phi(q) = L^*(N) - L^*(N/2+1)$

$$= 2N^3/\pi^2 - (2/\pi^2)(N/2)^3 + O(N^2\log N)$$

$$= 7N^3/4\pi^2 + O(N^2\log N)$$

Finally, we get $L_*(N) = 3N^3/4\pi^2 + O(N^2\log N)$

We now proceed to give an upper bound for E(N). Notice that

$$\Phi(N) = \sum_{1}^{N} \frac{1}{q}\phi(q) \geq S(N) + (L^*(N) - L(N))/N$$

and $\Phi(N)/L^*(N) = 3/2N + O(\log N/N^2) \simeq 3/2N$

From now on we neglect errors of the form $O(\log N/N^2)$. From these observations we have

$$S(N)/L^*(N) + (L^*(N) - L(N))/(NL^*(N)) < 3/2N$$

$$(L^*(N) - L(N))/(NL^*(N)) = 1/N - (1/N)L(N)/L^*(N)$$

On the other hand, be definition

$$S(N) = E(N)L(N)$$

We conclude

$$(E(N) - 1/N)L(N)/L^*(N) + 1/N \lesssim 3/2N$$

We now estimate, from below, the term $L(N)/L^*(N)$.

$$L(N)/L^*(N) \geq L_*(N)/L^*(N) \simeq 3/8$$

Thus $(E(N) - 1/N)3/8 + 1/N \lesssim 3/(2N)$

$$(E(N) - 1/N)3/8 \lesssim 1/(2N)$$

$$E(N) \lesssim 4/3N + 1/N$$

$$= 7/3N$$

We now have an asymptotic upper bound

$$E(N) \leq 7/3N + O(\log N/N^2)$$

It is clear that $E(N) \geq 1/N$. Let $S^*(N) = \Phi(N)$. Then we have: $NS^*(N/2) > L^*(N/2)$ (since in $S^*(N/2)$ we divide by q, $q \leq N/2$) and if $a > b$ we have $\frac{a+x}{b+x} \downarrow$ (taking derivatives)

$$\text{hence } E(N) = \frac{S^*(N/2) + \sum_{N/2+1}^{N} 1/q\ C(q,N)}{L^*(N/2) + L(N) - L^*(N/2)}$$

$$\geq \frac{S^*(N/2) + 1/N(L(N) - L^*(N/2))}{L^*(N/2) + L(N) - L^*(N/2)}$$

$$= \frac{1}{N} \frac{NS^*(N/2) + L(N) - L^*(N/2)}{L^*(N/2) + L(N) - L^*(N/2)}$$

$$\geq \frac{1}{N} \frac{NS^*(N/2) + L^*(N) - L^*(N/2)}{L^*(N/2) + L^*(N) - L^*(N/2)}$$

$$= \frac{S^*(N/2) + 1/N(L^*(N) - L^*(N/2))}{L^*(N)}$$

$$\simeq \frac{(3/\pi^2)(N^2/4) + 1/N((2N^3/\pi^2) - (2/\pi^2)(1/8)N^3)}{2N^3/\pi^2}$$

$$= 5/4N$$

So we get $5/4N \leq E(N) + O(\log N/N^2)$.

Several limitations on the utility of the calculations should be emphasized. Of the two limitations to be described, one tends to make the error estimate low while the second makes it high. The extent to which these factors may influence our estimates has not yet been determined, however, we are currently working on extensions of our methods to provide more realistic estimates. The strongest assumption lowering accuracy in the use of the above methods is that the edge pixels on the digital edge are known exactly. A weakening of this assumption is discussed in Section 2.6. On

the other hand, our calculations have provided bounds and the expected value for the maximum error per digital line. We expect the average error to be much less.

We now turn to a discussion of the finite sample behavior of the error. A closed form expression for the statistics of the error as a function of code length is difficult to derive. In order to get some feeling for the error we computed the maximum error 1/2q associated with each digital line of slope q. This error represents the maximum error in estimating the registration offset, given the digital line of slope q. The errors were calculated for all digital lines of length ten with the usual slope and origin conditions. There are 136 digital lines of length ten. Table 2.1 provides a summary of our results. The first entry, ERROR, in each row is a registration error and the second entry represents the probability that the maximum error is less than ERROR. This number is obtained as follows. Given a value ERROR, we compute the total number of digital lines for which $1/2q < \text{ERROR}$. This number is then divided by 136, the total number of digital lines to determine the percentage of digital lines with $1/2q < \text{ERROR}$. Thus we see from the table that the registration error exceed 0.25 pixels less than in 2% of the digital lines of length ten. Similarly, the error exceeds one tenth of a pixel in less than 14% of the digital lines.

The information in Table 2.1 provides exact probabilities (except for rounding error) for the digital lines of length ten. Given any longer digital line, it contains a subsegment of length ten, so these results provide worst case bounds on the maximum error for longer

lines. It should be noted that the assumption that the digital lines are equally probable is not as plausible as the assumption that the probability measure on real lines is rotation and translation invariant. This calculation will be performed in the follow-on work, but we do not expect the results to differ greatly. We also note that the worst possible error 1/2q was assumed for each digital line. The expected error over all real lines giving rise to the digital line will be much smaller.

We conclude that a very high level of subpixel accuracy is attainable in the restricted model discussed in this section. Furthermore, the calculated variation in error with line slope provides a good criteria for selecting features for registration. Future work will determine the extent to which this accuracy diminishes as we examine looser models.

ERROR	PROBABILITY (MAX ERROR) > ERROR
0.5000	0.0000
0.2500	0.0147
0.1666	0.0294
0.1250	0.0735
0.1000	0.1323
0.0833	0.2794
0.0714	0.3676
0.0625	0.6323
0.0555	0.7794
0.5000	0.9412
0.000	1.0000

Given an entry, a, in the first column, the corresponding entry in the second column is the percentage of digital lines of length ten whose maximum registration error exceeds a.

Line length = 10

Table 2.1 Error Probabilities for digital lines without points missing.

Section 2.6 Digital Lines - Points Missing

The determination of the exact set of pixels lying on the digitization of a real-world edge is not feasible, due to noise and geometric distortions. In this section, we relax the condition that a digital line segment be available to the weaker condition that only a subset of a digital line be detected. This situation is likely to arise if we try to fit a real line to suspected edge pixels and select those edge pixels for which the difference in areas between the two parts of the pixel separated by the line is not great. These pixels are more likely to be correct edge pixels. As we are unlikely to be able to guarantee the correctness of our pixels, this approach is restrictive. We think, however, that this work will provide a basis for the analysis of the more complex case in which incorrect pixels are present. This section describes methods for the analysis of the registration accuracy attainable by estimating the position of a real line using a subset of a digital line. Computer programs to estimate this accuracy are currently under development.

The description of the feasible line region for a subset of a digital line does not appear to be easily described in terms of parameters characterizing the subset. A simple observation leads to a method for calculating this feasible region in any particular case. We note that a line with slope between zero and one traversing a pixel must cross the main diagonal of the pixel (see figure 2.4). Given a subset of a digital line, the set of feasible lines is exactly the set of lines crossing the main diagonal of each pixel in the subset. Let $S=\{S_1,\ldots,S_n\}$ be a subset of a digital line of slope between zero and

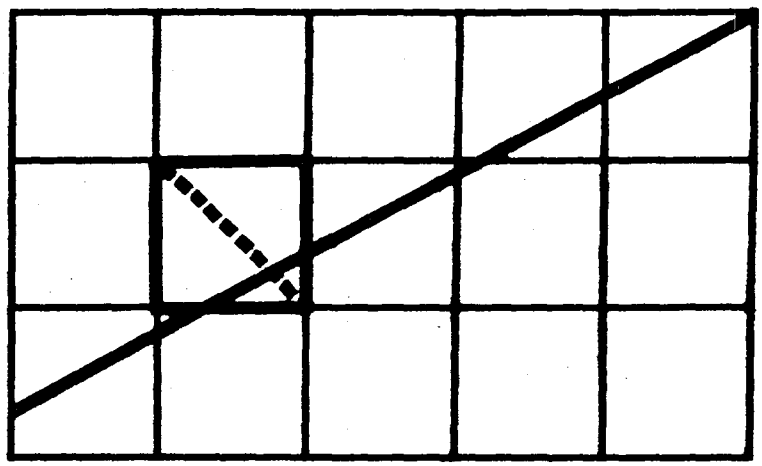


Figure 2.4 Intersection of a line with the main diagonal of a pixel. This intersection is used to derive constraints on the feasible set.

one and let s_1 and s_n be the leftmost and rightmost pixels respectively. Let the lower lefthand vertices of s_1 and s_n be (x_1,y_1) and (x_n,y_n). Then it can be shown that any line whose digitization contains these two pixels is a convex combination of the lines $L((x_1+1,y_1),(x_n,y_n))$, $L((x_1+1,y_1),(x_n,y_n+1))$, $L((x_1,y_1+1),(x_n,y_n))$, and $L((x_1,y_1+1),(x_n,y_n+1))$. Thus the feasible region is a quadrilateral in y-intercept, slope space.

Each additional pixel which our feasible lines are constrained to pass through restricts us to a subset of the feasible quadrilateral, namely the subset consisting of all lines passing through the main diagonal of the intermediate pixel. Let L be a line passing through s_1,s_n and an intermediate pixel s_i. Assume L is in the interior of the feasible quadrilateral for s_1 and s_n. Then any sufficiently small change in the slope and y-intercept of L will keep it in the feasible region. If L does not enter s_i at a vertex of s_i, then a sufficiently small change in its slope and y-intercept will not change the fact that s_i is in its digitization. If L does enter s_i from the left through a vertex, then any increase on the y-intercept if it enters at the top and decrease if it enters at the bottom will change the digitization of L to exclude s_i. Thus the boundary of the feasible region for lines going through s_1,s_i, and s_n is obtained from the feasible region for s_i and s_n by cutting the region by those curves corresponding to all lines passing through the lower and upper lefthand vertices of s_i. These curves are actually straight lines. Let (x_i,y_i) denote the lower lefthand corner of s_i. Then any line through (x_i,y_i) satisfies $y_1=mx_i+b$ or equivalently $m=(y_i-b)/x_i$. Thus the set of all lines passing through (x_i,y_i) is given

by the set $\{(b, y_i/x_i - b/x_i) \mid b \in R\}$. This set is just a line in y-intercept, slope space. This argument may be extended inductively to obtain the feasible region for any subset of a digital edge by adding one pixel at a time. Each feasible region is obtained from the previous one by intersecting it with the region contained between the two parallel lines indicated above.

The computation of the feasible region can be performed rapidly by testing each vertex going sequential around the feasible polygon to determine whether it lies between or outside the next pair of lines. This procedure tells us between which pairs of vertices the parallel lines intersect the polygon. Thus only four intersections need be computed for each extension.

Given a feasible polygon, it is possible to compute a y-value at which the width of the feasible region in x-y space is minimized. This is analagous to the channel of thickness 1/q discussed in section 2.3. As in the case of the feasible region in x-y space for a digital line segment, the feasible region in x-y space for a subset of a digital line segment is obtained by drawing the lines corresponding to the vertices of the feasible polygon. For each x-value the feasible region extends from the lowest point on these lines to the highest point over the specified x value. The minimum width can be shown to be achieved at a point where two of the lines cross. Thus to compute the minimum width, evaluate the width at each intersection of lines. For n lines, we have $n(n-1)/2$ intersections so for moderate size subsets, say 8-10 points this computation is quite fast.

We are now able to assign to each subset of a digital line, a point at which its feasible region width is minimized. Using the midpoint of this strip as an estimate for a point on the line, we can now give an upper bound on the registration error for any given subsets.

The above procedure can be used to provide error bounds for any given subset, but we would like to get some general measure of the success of the method. One approach would be to compute the maximum error for every subset of every digital line through the pixel at the origin and with slope between zero and one and specified length, say ten pixels. If we generate each digital line and take all its subsets we generate approximately 136,000 sets, though they need not all be distinct since lines can share subsets. By computing the error for each of these subsets, it is possible to determine the expected maximum error for subsets of a given size. It would also be possible to determine those approximate slopes of digital lines which are best in that the expected maximum error is minimized.

We plan to carry out the registration accuracy studies described above. These results will then be used to assess the quality of edge detection needed to assure subpixel registration accuracy. We then would like to use additional information such as gradients to provide further accuracy.

Section 2.7 Digital Line - Points Missing, Points Added

For most images it is impossible to guarantee that any set deemed to be a subset of the digitization of a real line is correct. In Section 2.6 we discussed the accuracy attainable when a subset of a digital line is available. The modeling of the further accuracy resulting from the presence of incorrect pixels appears to be quite complex. Our initial plans for study of this problem will involve the addition of varying numbers of incorrect points to a small number of subsets of digital lines to determine the resulting error. The planning of this work is in an early stage.

One aspect of the incorrect points problem deserves mention. The knowledge that the digital edge comes from a straight edge provides a powerful constraint on the feasible lines. Given a set of pixels, it is possible to determine for each digital line, how many pixels it has in common with the observed pixel set. If we know the approximate beginning and ending of the line segment, the number of digital lines passing through a substantial percentage of the observed pixels will be small. The feasible region for the digital line maximizing the number of pixels hit is a reasonable candidate for the correct digital line. If more than one line maximizes this quantity the feasible region can be extended to the union of the feasible regions of these digital lines. We intend to pursue this approach in our later work.

Section 3.0 Subpixel Translation-Registration of Stationary Random Fields

Consider the problem of registering (i.e., finding an appropriate overlay by relative translation of) a sensed planar image with respect to a larger reference image supposed to contain it. In typical remote-sensing applications, both the sensed and reference images will be given, at the same resolution, as arrays of gray-level values, one value for each pixel. Both images will typically be noisy, due to minor changes in weather or ground features; to sensor characteristics; to preprocessing and detrending; and possibly also to nonlinear filtering of gray-level images, for example by edge-enhancers and thresholding.

The primary model assumptions for our discussion of this problem are:

(a) there exist underlying continuous sensed and reference images $Z_S(\underline{x})$ and $Z_R(\underline{x})$ before discretization into pixels, where $\underline{x} = (x_1, x_2)$ are planar coordinates, such that $Z_R(\cdot)$ and $Z_S(\cdot)$ are jointly strictly stationary random fields (i.e., have translation-invariant statistics) with rapidly decaying dependence between the fields $(Z_R(\underline{x} + \underline{y}), Z_S(\underline{x} + \underline{y}))$ and $(Z_R(\underline{y}), Z_S(\underline{y}))$ as a function of $|\underline{x}| = (x_1^2 + x_2^2)^{1/2}$ (see [De] for precise conditions and definitions: Z_R and Z_S must be ϕ-mixing with $\sum_{r=1}^{\infty} r\phi^{1/2}(r) < \infty$);

(b) there exists an unknown translation-parameter $\underline{\theta} = (\theta_1, \theta_2)$, a known pixel width h, and a known kernel-function $K(\cdot,\cdot)$ such that the observed sensed and reference gray-level arrays are

$$X_S(j,k) = h^{-2}\int_0^h\int_0^h K(s,t)Z_S(jh + \theta_1 + s,\ kh + \theta_2 + t)dsdt$$

$$X_R(j,k) = h^{-2}\int_0^h\int_0^h K(s,t)Z_R(jh + s,\ kh + t)dsdt$$

The fields Z_R and Z_S are of course assumed to be highly correlated

images representing the same ground truth, and for identifiability of location it is quite important that the correlation between $Z_R(\underline{x})$ and $Z_S(\underline{x} + \underline{y})$ be small except for $\underline{y}$ close to 0. The parameter $\underline{\theta}$ is then identifiable in principle from large images $(Z_R(\underline{x}))_{|x_1|,|x_2| \leq Mh}$ and $(Z_S(\underline{y} + \underline{\theta}))_{|y_1|,|y_2| \leq Lh}$. To see whether and to what extent $\underline{\theta}$ remains identifiable from data $\{X_R(j,k): |j|,|k| \leq M\}$ and $\{X_S(j,k): |j|,|k| \leq L\}$ is precisely our problem. Note that the kernel function K models the linear transformation of a pixel image to a gray level. For simplicity (although all our results can be extended to general known K), and in apparent agreement with previous researchers, we assume in what follows that $K(s,t) \equiv 1$.

Our model assumptions are in some respects similar to, but substantially generalize, those of [Mo - Sm] (who were, however, interested also in the effects of affine distortion). In addition to (a), [Mo - Sm] assumed that $Z_R(\cdot)$ and $Z_S(\cdot + \underline{\theta})$ are directly observable and jointly Gaussian. This restrictive assumption is not necessary for an understanding of the asymptotic distribution theory, for large sensed images, of the maximum-correlation estimator $\underline{\theta}^*$ for $\underline{\theta}$ (see below). Moreover, [Mo - Sm] do not take into account the transformation of Z_R, Z_S which renders only X_R, X_S directly observable Thus their analysis, which we extend and improve in Section 3.1 of this report, is relevant only to the problem of consistent estimation of $\underline{\theta}$ in the sense of "correct local registration". We consider in Section 3.2 theoretical approaches based on model (a)-(b) above to the evaluation of sub-pixel accuracy of estimation. A summary of our findings, together with proposals for further empirical and Monte Carlo studies, concludes this section.

Section 3.1 Neighborhood-Consistency of Maximum-Correlation Estimation

The reason that we do not need to assume Gaussian distributions for gray-levels is simply that the fixed-offset "correlation" statistic for $Z_R(\cdot)$, $Z_S(\cdot\cdot + \underline{\theta})$ given by

$$(*) \qquad C(\underline{t}) = (2T)^{-2}\int_{-T}^{T}\int_{-T}^{T} Z_R(\underline{x} + \underline{t})Z_S(\underline{x} + \underline{\theta})d\underline{x}, \quad T \equiv Lh,$$

is asymptotically weakly convergent as a random process in $\underline{t}$ as $L \to \infty$ to a Gaussian random field, under the precise condition of [De] on decay of dependence mentioned in (a). If $Z_R(\cdot)$ and $Z_S(\cdot + \underline{\theta})$ are directly observable, then a natural statistic to estimate $\underline{\theta}$ is

$$\underline{\theta}^* \equiv \text{maximizer of } C(\cdot) \text{ on } [-T,T] \times [-T,T].$$

The most easily interpreted figures of merit for this (and any other) estimator are of the form

$$Q(\tau) \equiv P\{|\underline{\theta}^* - \underline{\Theta}| \leq \tau\}$$

or

$$Q_{T_0}(\tau) \equiv P\{\sup_{\underline{x}:|\underline{x}-\underline{\theta}|\leq\tau} C(\underline{x}) = \sup_{\underline{x}:||\underline{x}||_1\leq T_0} C(\underline{x})\}$$

where $||\underline{x}||_1 = \max(x_1,x_2)$ and T_0 is a fixed size of window inside which may assume $\underline{\theta}$ lies. We note that since [Mo - Sm] did not treat $C(\cdot)$ as a random field, they did not propose to evaluate quantities $Q_{T_0}(\tau)$ but rather to compare the asymptotically (in T) normal single-offset correlations $C(\underline{t})$ with either specified or "sidelobe" thresholds. Evaluation of $Q_{T_0}(\tau)$ is clearly a problem about random processes - not simply finite-dimensional distributions - for which we now formulate an asymptotic solution, assuming (a).

Let $D(\underline{t})$ denote the expectation $EC(\underline{t})$. Joint stationarity of $Z_R(\cdot)$ and $Z_S(\cdot + \underline{\theta})$ implies

$$D(\underline{t}) = EC(\underline{t}) = E\{Z_R(\underline{t})Z_S(\underline{\theta})\}$$

which would be consistently estimated when T is large by the expression $C(\underline{t})$ in (*). (In other words, [De]'s conditions imply a law of large numbers

for $C(\underline{t})$ for each $\underline{t}$). The stationary covariance function

$$V(\underline{x} - \underline{y}) \equiv \mathrm{Cov}(C(\underline{x}), C(\underline{y})) \sim T^{-2}\sigma(\underline{x} - \underline{y}) \quad \text{as } T \to \infty$$

(which defines the asymptotic covariance $\sigma(\cdot)$) can likewise be consistently estimated by a fourfold integral expression (cf. [Mo - Sm], where some simplifications occur if Z_R and Z_S are jointly Gaussian). The following Lemma and corollary apply in particular to estimate the probability $Q^*_{T_0}(\tau) \equiv P(\sup\{C^*_T(\underline{t}) : ||\underline{t}||_1 \leq T_0, \ ||\underline{t} - \underline{\theta}|| \geq \tau\})$ where $C^*_T(\cdot)$ is the Gaussian random field with the same mean and covariance as $C(\cdot)$ for fixed T. For more general conditions of applicability, see Appendix.

LEMMA 3.1 Let $Y(\underline{t})$ be a real-valued separable random field on $[-T_0, T_0]^d$ and S be the complement in $[-T_0, T_0]^d$ of a convex set such that

$$\sup\{||\underline{t}||_1 : t \in S^c\} \leq T_0 - n^{-2},$$

where n is a fixed integer ≥ 3. Suppose also that for $\underline{s}, \underline{t} \in S$, for fixed $\Gamma > 0$ and a non-decreasing continuous function $\psi(\cdot)$, that

(†) $\quad |Y(\underline{t})|/\Gamma$ and $|Y(\underline{s}) - Y(\underline{t})|/(\psi(||\underline{t} - \underline{s}||_1))$

are each stochastically smaller than the absolute value of a standard normal random variable where $\int_1^\infty \psi(\exp(-x^2))dx < \infty$. Then for any $x \geq (4d\log n)^{1/2}$,

$$P\{\sup_{\underline{t} \in S} |Y(\underline{t})| \geq x(\Gamma + \frac{2}{\sqrt{2}} \int_1^\infty \psi(n^{-u^2})du)\} \leq C_d(n) \int_x^\infty e^{-u^2/2}du$$

where

$$C_d(n) = \sqrt{2/\pi}\left(1 + \frac{d}{2d+1}(\sqrt{2} - 1)^{-1} \frac{4d\log n}{4d\log n - 1}\right) \cdot (\lceil 2T_0 n^2 \rceil)^d$$

and $\lceil \ \rceil$ denotes "roof function".

The proof, which we omit, is a direct adaptation of the method of [Ma], in which proper attention was not drawn to the very weak use made of the Gaussian assumption: the assumption (†) above is of course satisfied if $Y(\cdot)$ is Gaussian with $\Gamma = \sup_{\underline{t} \in S} [E(Y^2(\underline{t}))]^{1/2}$ and

$\sup_{\underline{s},\underline{t}\in S=||\underline{s}-\underline{t}||_1\le u} E[Y(\underline{s}) - Y(\underline{t})]^2 \le \psi^2(u)$. The method of [Ma] does permit some further relaxation of (†) at the cost of more complicated estimates.

COROLLARY 3.2 Assume (a), (*), and fix $\tau > 0$. Let T_0 be such that $||\underline{\theta}||_1 \le T_0 - n^{-2}$ for fixed $n \ge 3$. Define for fixed T

$$H_\tau \equiv \inf\{D(\underline{\theta}) - D(\underline{t}): \ ||\underline{t} - \underline{\theta}|| \ge \tau, \ ||\underline{t}||_1 \le T_0\}$$

Assume also that (†) of the Lemma holds with $Y(\underline{t}) \equiv C(\underline{t}) - C(\underline{\theta}) - D(\underline{t}) + D(\underline{\theta})$, $d = 2$, $S = \{\underline{t} : ||\underline{t}||_1 \le T_0, \ ||\underline{t} - \underline{\theta}|| \ge \tau\}$, and $\psi(u) \le a\, u^b$ with $b > 0$. Then

$$P(\sup\{C(\underline{t}) : ||\underline{t}||_1 \le T_0, \ ||\underline{t} - \underline{\theta}|| \ge \tau\} \ge C(\underline{\theta})) \le C_2(n)\int_{x_*}^{\infty} e^{-u^2/2}du,$$

whenever

$$x_* \equiv H_\tau/(\frac{2a}{\sqrt{2}-1}\int_1^\infty \exp[-bu^2\log n]du + \Gamma) \text{ is } \ge (8\log n)^{1/2}.$$

The Corollary follows immediately from the Lemma using $Y(\cdot)$ defined above since

$$P(\sup\{C(\underline{t}): ||\underline{t}||_1 \le T_0, \ ||\underline{t} - \underline{\theta}|| \ge \tau\} \ge C(\underline{\theta})) \le P(\sup_{\underline{s}\in S}|Y(\underline{s})| \ge H_\tau).$$

To make the conclusion of the Corollary more specific, we note that if $C(\cdot)$ were Gaussian then Γ can be taken $(2V(0))^{1/2} \simeq (2\sigma(0))^{1/2}/T$ while $\psi^2(u)$ can be taken $= 2(V(0) - V(u))$; if $V(\cdot)$ can be assumed differentiable at 0 (or, more conservatively, to allow covariances such as $\exp(-|u|)$, Holder continuous of exponent 1/2), then $b \ge 1/4$ and a will be of order T^{-1}. Choosing $n = \lfloor T^2 \rfloor$, and assuming the hypotheses of the Corollary, we find

$$\text{(A)} \qquad 1 - Q_{T_0}(\tau) \le 8\, T_0^2 T^4 \frac{\exp[-(H_\tau/(\Gamma+\varepsilon))^2/2]}{H_\tau/(\Gamma+\varepsilon)}, \quad \varepsilon = O(T^{-1-b}).$$

This bound, which should be quite good for sensed images of practical sizes, suggests as figure of merit for local accuracy of registration the ratios $H_\tau^2/V(\underline{0})$. These ratios can, for instance, be estimated accurately from a large reference image alone if the noise field $Z_S(\cdot + \underline{\theta}) - Z_R(\cdot)$ is independent of $Z_R(\cdot)$ with known covariance structure. It remains as a

subject for numerical experimentation, with real and simulated images of various sizes, to test both the validity and stringency of the bound (A).

Results related to (A), with bounds giving exponential decrease with T of probabilities of misregistration, have been obtained for a somewhat different model in unpublished research of C. Herman. Herman considers a model in which pixel gray-levels are independent and regionally identically distributed for a finite (small) number of geometrically identifiable homogeneous regions. Thus his work, while more special in its model of noise, does allow for some nonhomogeneity over the sensed and reference images. This suggests (and we propose in Section 4) that the empirical testing of (A) should cover nonstationary images as well. See Appendix 1 for some modifications of the Corollary in this direction.

Section 3.2 Interpolation Using Pixel-Discretized Images

We turn now to the question of estimating $\underline{\theta}$ to sub-pixel accuracy based on data $X_R(\cdot)$, $X_S(\cdot)$. That is, given observations $\{X_R(j,k), X_S(j,k)\}_{|j|,|k|<L}$, and supposing it is known that $jh \leq \theta_1 < (j^* + 1)h$, $k^*h \leq \theta_2 < (k^* + 1)h$, we want to know which characteristics of the random fields Z_R and Z_S control the possibility of a finer estimation of $\underline{\theta}$. For simplicity and definiteness, we assume in this Section, in addition to (a), (b) above with $K \equiv 1$:

(c) $Z_N(\cdot) \equiv Z_S(\cdot + \underline{\theta}) - Z_R(\cdot)$ is independent of $Z_R(\cdot)$.

Under our assumptions, the "correlation-statistics" at offset $\underline{x} \equiv (\alpha h, \beta h)$

$$(**) \qquad C(\alpha h,\beta h) \equiv \sum_{j=-L}^{L} \sum_{k=-L}^{L} X_S(j,k)X_R(j + \alpha, k + \beta)$$

have expectation (assuming the means of Z_R, Z_S have been centered to $\underline{0}$)

$$D(\underline{x}) = D(\alpha h,\beta h) \equiv \int_{R^2} e^{\left(i\underline{\lambda} \cdot (\underline{\theta} - \underline{x})\right)} \left|\frac{e^{i\lambda_1 h} - 1}{\lambda_1 h}\right|^2 \left|\frac{e^{i\lambda_2 h} - 1}{\lambda_2 h}\right|^2 G(d\underline{\lambda})$$

where $G(\cdot)$ is defined by

$$\mathrm{Cov}(Z_R(\underline{0})Z_R(\underline{x})) = \int_{R^2} e^{\left(i\underline{\lambda} \cdot \underline{x}\right)} G(d\underline{\lambda}).$$

Now as $T = Lh$ gets large, the covariances among all $C(\underline{x})$ variables go to 0, and the statistical aspect of finding the $\underline{\theta}^*$ which maximizes $C(\cdot)$ disappears: all that remains is the interpolation problem of finding a numerically estimated maximizer $\underline{\theta}$ for $D(\cdot)$ on $[j^*h, j^*h + h]x[k^*h, k^*h + h]$. In fact, since $D(\cdot)$ is "observable" (through $C(\underline{x})$) only at lattice-points $(\alpha h, \beta h)$ where α and β are integers, it is clear that without some assumptions on the functional form of $D(\cdot)$ or some prior knowledge about approximate constancy of curvature of $D(\cdot)$ on $[j^*h, j^*h + h]x[k^*h, k^*h + h]$, no precise

subpixel estimation is possible.

In order to derive an index of how precisely one can hope to estimate $\underline{\theta}$ from $\{D(\alpha h, \beta h): \alpha,\beta \text{ integers}\}^-$ -- and it is clear that for finite T, the observability of C rather than D can only degrade the accuracy of prediction -- we expand the Fourier-Stieltjes representation of $D(\underline{x})$ in a Taylor series about $\underline{\theta}$. For this we require the following assumption, which would follow from but can be slightly weaker than mean-square differentiability of $Z_R(\cdot)$:

$$\text{(d)} \qquad \xi_h \equiv \int\int \frac{|(e^{i\lambda_1 h} - 1)(e^{i\lambda_2 h} - 1)|^2}{h^4 \lambda_1^2 \lambda_1^2} \; ||\underline{\lambda}||^4 G(d\underline{\lambda}) < \infty$$

Under assumptions (a) - (d), we can write by the Mean Value Theorem

$$\text{(B)} \quad D(\underline{\theta}) - D(\underline{x}) = \iint \frac{|(e^{i\lambda_1 h} - 1)(e^{i\lambda_2 h} - 1)|^2}{2h^4 \lambda_1^2 \lambda_2^2} [\underline{\lambda} \cdot (\underline{x} - \underline{\theta})]^2 G(d\underline{\lambda})$$

$$+ \frac{\gamma}{24} \xi_h ||\underline{x}' - \underline{\theta}||^4$$

where $\underline{x}'$ lies on the line between $\underline{x}$ and $\underline{\theta}$, and $|\gamma| \leq 1$. We now suppose that $\underline{x}$, and thus also $\underline{x}'$, lies in the pixel $P = [j*h, j*h + h] \times [k*h, k*h + h]$ containing $\underline{\theta}$, and remark that if, in addition to (d), $Z_R(\cdot)$ is twice mean-square differentiable then ξ_h is uniformly bounded in h.

From (B) it follows that the maximizer of $D(\underline{x})$, known to lie in the pixel P containing $\underline{\theta}$, is at most a distance κh from $\underline{\theta}$, where

$$\kappa = (\tfrac{1}{12} \xi_h / a_1)^{1/2} h$$

and a_1 = smallest characteristic value of the quadratic form

$$q(\underline{y}) = h^{-4}\iint |(e^{i\lambda_1 h} - 1)(e^{i\lambda_2 h} - 1)|^2 \lambda_1^{-2}\lambda_2^{-2}(\underline{\lambda} \cdot \underline{y})^2 G(d\underline{\lambda}).$$

Moreover, asymptotically for small h, κ gives the approximate best-possible fraction-of-pixel accuracy attainable if $D(\cdot)$ is maximized via a local quadratic approximation. That is, knowledge of ξ_h and $q(\cdot)$ alone would allow no better bound on accuracy of numerical maximization of $D(\cdot)$.

Thus in the limit of infinite $T = Lh$ the parameter κ is an easily interpretable figure of merit for subpixel estimation of $\underline{\theta}$ based on locally biquadratic surfaces fit to $D(\cdot)$ (or equivalently, to $C(\cdot)$). It remains to suggest approximate computational procedures, for use with real or simulated images $X_R(\cdot)$, $X_S(\cdot)$, to estimate $\underline{\theta}$ and κ. In fact, for very small h (that is, rapid correlation decay for processes X_R, X_S over distance scale h, at least if $Z_R(\cdot)$ is twice mean-square differentiable) ξ_h should be closely approximated by

$$E[(\nabla_{1h} + \nabla_{2h})^2 Z_R(\underline{0})]^2 \simeq E[(\nabla_1 + \nabla_2)^2 X_R(\underline{0})]^2 \equiv \xi_h$$

and similarly the quadratic form $q(\underline{y})$ is approximately by

$$\tilde{q}(\underline{y}) \equiv E[(y_1\nabla_1 + y_2\nabla_2)X_R(\underline{0})]^2$$

where for any function $f(\underline{x})$ on R^2, $\nabla_{1h}f(\underline{x}) \equiv f(\underline{x}) - f(x_1 - h, x_2)$ and $\nabla_{2h}f(\underline{x}) \equiv f(\underline{x}) - f(x_1, x_2 - h)$, and for function $g(j,k)$,

$$\nabla_1 g(j,k) = g(j,k) - g(j-1,k), \quad \nabla_2 g(j,k) = g(j,k) - g(j,k-1).$$

Now stationarity and ergodicity of $X_R(\cdot)$ implies that consistent estimators (as $L \to \infty$) are given by

$$\hat{\xi}_h \equiv (2L)^{-2} \sum_{j=-L}^{L} \sum_{k=-L}^{L} [(\nabla_1 + \nabla_2)^2 X_R(j,k)]^2$$

$$\hat{q}(\underline{y}) \equiv (2L)^{-2} \sum_{j=-L}^{L} \sum_{k=-L}^{L} [(y_1\nabla_1 + y_2\nabla_2)X_R(j,k)]^2.$$

Letting $\hat{a}_1$ denote the smallest characteristic value of the quadratic form $\hat{q}(\cdot)$, we can define the estimated figure-of-merit for subpixel accuracy

$$\hat{\kappa} = h(\tfrac{1}{12}\hat{\xi}_h/\hat{a}_1)^{1/2}$$

As with the figure-of-merit defined in Section 3.1, we must still perform numerical experiments with real and simulated pixel-discretized images to test both the correctness and informativeness of the subpixel-accuracy bound κ.

If the estimated accuracy $\hat{\kappa}$ is $< .5$ and at the same time the estimated bounds for $1 - Q_T(2h)$ from Section 3.1 are extremely tight (say $<.001$), there still remains the problem of constructing an interpolation-estimator for the maximizer $\underline{\theta}$ of $D(\cdot)$based on the noisy values $C(\alpha h,\beta h)$. The best developed methodology for estimating (interpolated) values $D(\underline{x})$ linearly from observations $\{C(\alpha h,\beta h)\}_{\alpha,\beta}$, called "kriging" (see [Du] and [Ri], Section 4.4) suffers from one glaring defect in this context, namely: it requires that the covariances for $C(\cdot)$ be known (or estimated) at all points $\underline{x}$, not simply at lattice points $(\alpha h,\beta h)$. If for experimental purposes (as in [Mo - Sm]) we assume a special parametric form for the covariance functions of $Z_R(\cdot)$ and $Z_S(\cdot)$, then a parameter-estimation step followed by kriging-interpolation and maximization (using the kriging equations given by [Du] and [Ri]) will give a usable procedure for subpixel registration. This has not yet been tried.

Section 3.3 Summary and Proposed Numerical Experiments

We collect in this brief final Section our main theoretical results, and the corresponding numerical tests they suggest on real and simulated image-data.

(i) For large continuous sensed images with conditionally Gaussian noise given the reference image (see Appendix), formula (A) in Section 2 bounds the probability of misregistration by more than distance τ. Numerical work with pixel-discretized real and simulated images is needed to test the validity and usefulness of the bound.

(ii) When translation-registration to the nearest pixel has already been accomplished, and all imagery can be assumed spatially homogeneous with rapidly decaying correlations on the pixel distance-scale h, the estimator $\hat{\kappa}$ from Section 3.2 approximately limits the subpixel accuracy possible if the sensed and reference images were infinitely large with noise- and reference-images stochastically independent. Again, numerical experimentation will empirically determine whether these assumptions and figures-of-merit are valid or useful.

(iii) The kriging-interpolation and maximization of $C(\cdot)$ should certainly be tried, as sketched at the end of Section 3.2, using simple parametric forms for the covariances of Z_R and Z_S.

(iv) Finally, if the experiments in (i) - (iii) prove successful, theoretical and empirical extensions of this work, to the case of registration with respect to affine distortion considered by [Mo - Sm], seem both desirable and possible.

APPENDIX. Probability Bound on Local Registration Error for Conditionally Gaussian Sensed Images.

In this appendix we state and prove a Theorem giving the most important case (including that of [Mo-Sm] in nonstationary settings) in which the hypotheses of the Lemma and Corollary of Section 3.1 can be proved.

THEOREM A.1. Suppose that $Z_R(\cdot)$ and $Z_S(\cdot + \_)$, with $\_ \; 1 \; \_ \; T_0 - n^{-2}$, are (nonhomogeneous) real-valued separable random fields on R^2 for which $Z_N(\cdot) \quad Z_S(\cdot + \_) - Z_R(\cdot)$ is conditionally given $Z_R(\cdot)$ a Gaussian random field, and for which the covariance function $R(\underline{s},\underline{t})$ of $Z_R(\cdot)$ is continuous and satisfies for b,c 0

(c) $\sup \{R(\underline{s},\underline{s}) + R(\underline{t},\underline{t}) - 2R(\underline{s},\underline{t}): ||\underline{s} - \underline{t}|| \leq u\} \leq c \cdot u^b (-\log u)^{-1/2}$,

$0 < u \leq 1$

Let: $M_N(\underline{t}) \equiv E[Z_N(\underline{t}) | \{Z_R(\cdot)\}]$

$P_N(\underline{s},\underline{t}) \equiv E[Z_N(\underline{s})Z_N(\underline{t}) - M_N(\underline{s})M_N(\underline{t}) | \{Z_R(\cdot)\}]$

$\overline{D}(\underline{t}) \equiv E[C(\underline{t}) | \{Z_R(\cdot)\{]$ where $C(\cdot)$ is as in (*) for fixed T,

$\overline{V}(\underline{t}) \equiv E[(C(\underline{t}) - D(\underline{t}))^2 | \{Z_R(\cdot)\}]$

$\overline{\Gamma}^2 \equiv \sup\{V(\underline{t}): ||\underline{t}||_1 \leq T_0$

$$\Psi^2(u) \equiv 2 \cdot \sup_{||\underline{s}||_1, ||\underline{t}||_1 \leq T_0, \; ||\underline{s} - \underline{t}||_1 \leq u} (2T)^{-4} \int_{[-T,T]^4} [Z_R(\underline{x} + \underline{s}) - Z_R(\underline{x} + \underline{t})] \cdot$$

$$[Z_R(\underline{y} + \underline{s}) - Z_R(\underline{y} + \underline{t})] P_N(\underline{x} + \underline{\theta}, \underline{y} + \underline{\theta}) d\underline{x} d\underline{y}$$

$$H_\tau \equiv \inf\{\overline{D}(\underline{\theta}) - \overline{D}(\underline{t}): ||\underline{t} - \underline{\theta}|| \geq \tau, \; ||\underline{t}|| \leq T_0\}.$$

Then whenever $\overline{x} \equiv H_\tau / (\frac{2A}{2 - 1 \; 1} \int_{\infty} n^{-bu^2} du + \overline{\Gamma}) \geq (8 \log n)^{1/2}$, where

$$A \equiv \sup\{\Psi(u)/u^b: \quad 0 < u \leq 1\}$$

$$P\{\sup\{C(\underline{t}):\ ||\underline{t}||_1 \le T_0,\ ||\underline{t} - \underline{\theta}|| \ge \tau\} \ge C(\underline{\theta})|\{Z_R(\cdot)\}$$

$$\le C_2(n) \int_{\overline{x}}^{\infty} e^{-u^2/2} du$$

PROOF: First we remark that the condition (c) implies [Ad Theorem 3.3.3 and Lemma 3.4.1] that $Z_R(\cdot)$ is uniformly Holder continuous with exponent b on $[-T - T_0, T + T_0]^2$, hence uniformly bounded, and the random variable A has finite varaince. The Lemma of Section 3.1 now applies (with moments and probabilities all taken conditionally given $Z_R(\cdot)$) to the conditionally Guassian random field $Y(\underline{t}) \equiv C(\underline{t}) - C(\underline{\theta}) - \overline{D}(\underline{t}) + \overline{D}(\underline{\theta})$.

Here as in the Lemma, the conditionally Gaussian assumption could be replaced by an assumption (†).

We observe also, as in Section 3.1, that P_N and M_N are either known or can be assumed to have a given form, then all other quantities in the Lemma are defined in terms of a given (realization of the) reference image $Z_R(\cdot)$.

Section 4.0 Maximum Likelihood Corner Detection

The reliable detection of edges, angles, and other geometric configurations in sensed imagery is a key factor in many algorithms designed to achieve subpixel registration accuracy. In particular, locating image points to the correct pixel in a reference image, allows a decoupling of the pixel and subpixel registration problems. In this chapter, we describe a maximum likelihood estimation procedure for matching a sensor image corner with a reference image corner. This work is related to work of Novak [No] on the estimation of curve matching between a sensed and reference image.

Novak proposes a solution to the problem of finding a particular edge shape in a picture (which is usually called the sensor image): the edge is embedded in a binary template and, using an edge ratio statistic, the template is matched to the sensor image.

An edge, Figure 4.2, is defined to be a curve separating two homogeneous regions of differing grey levels. A template consists of an edge along with a narrow band of pixels on both sides of the edge. It is assumed that the pixels in the sensor image are statistically independent, each being distributed exponentially. The pixels which lie on the dark side of the curve have mean δ, and those on the light side have mean λ.

The edge ratio statistic is $z=g_1/g_0$, where g_1 is the sum of the grey levels of the pixels under the dark region of the template, and g_0 is the sum under the light region. The statistic is evaluated at all points of the sensor image, and that point at which z is a maximum is selected as the match point.

The selection by Novak of the edge ratio statistic appears to be based on its "good" performance on several test cases, and on the fact that its distribution can be reasonably approximated by the F distribution. In this section Novak's model will be altered so as to admit of a closed form MLE.

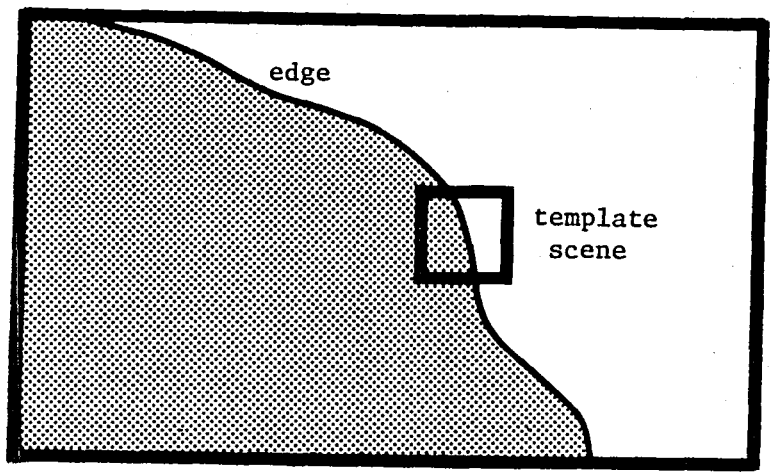


Figure 4.1 Region boundary and template.

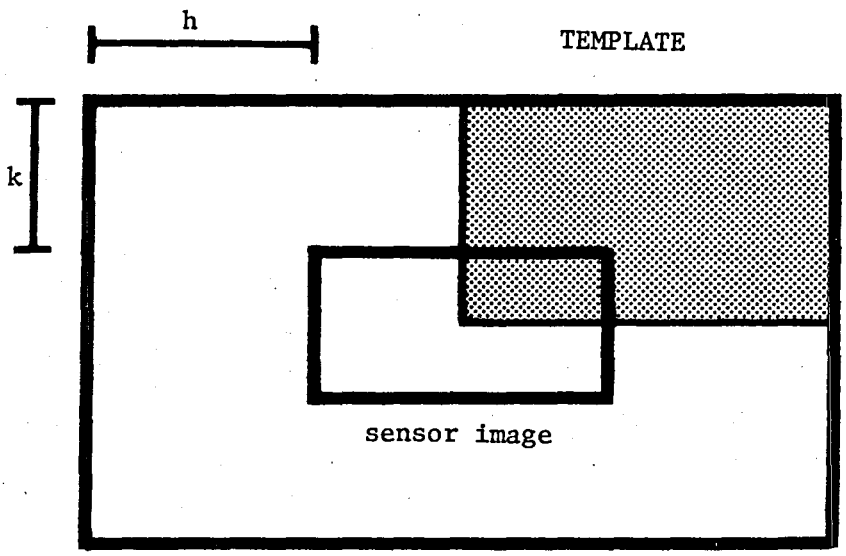


Figure 4.2 Corner image and window for corner detection analysis.

Section 4.1 The Model

In the modified model the template is assumed much larger than the sensor image, and the problem takes the following form. Imagine that the template represents a binary photograph (noiseless) of a scene in which an edge separates a dark from a light region, as in figure 4.2. A noisy photograph (the sensor image) is taken of a part of this scene - in general containing a segment of the edge. Thus given the template and the sensor image, we seek the correct overlay point.

Formally we view the template T as the lattice of nm points

$$\begin{matrix} t_{11} & t_{12} & \cdots & t_{1m} \\ t_{21} & t_{22} & \cdots & t_{2m} \\ \vdots & \vdots & & \vdots \\ t_{n1} & t_{n2} & \cdots & t_{nm}, \end{matrix}$$

in which each t_{hk} is either 0 or 1. Thus T is partitioned into the sets R_0 and R_1, where

$$R_\alpha=\{(h,k): t_{hk}=\alpha,\ 1\le h\le n,\ 1\le k\le m\},\ \alpha=0,1.$$

The sensor image S is the lattice of independent random variables

$$\begin{matrix} S_{11} & S_{12} & \cdots & S_{1p} \\ S_{21} & S_{22} & \cdots & S_{2p} \\ \vdots & \vdots & & \vdots \\ S_{q1} & S_{q2} & \cdots & S_{qp} \end{matrix}$$

in which the value S_{ij} represents the grey level of the (i,j)th pixel in the sensor image.

The distribution of each S_{ij} depends on where S is overlaid on T. If S_{11} is placed over t_{hk}, $1 \le h \le n-q$, $1 \le k \le m-p$, then the conditional density of S_{ij} $(1 \le i \le q,\ 1 \le j \le p)$ is

$$f_{ij}(s:h,k) = f(s,\theta_0)I_0[i,j] + f(s,\theta_1)I_1[i,j],$$

where $\{f(s,\theta)\}$ is a family of densities indexed by the parameter θ, and

$$I_\alpha[i,j] = \begin{cases} 1, \text{ if } (i+h-1, j+k-1) \in R_\alpha \\ 0, \text{ otherwise,} \end{cases}$$

$\alpha = 0,1$. The dependence of I_α on (h,k) has been suppressed for notational convenience.

Section 4.2 Results

Let $\underline{s}$ represent the b by m matrix $[s_{ij}]$, where s_{ij} is the observed value of S_{ij}. Then the joint conditional density of S, given that S_{11} is placed over $t_{h,k}$, is

$$f_S(\underline{s}:h,k) = \prod_{i=1}^{q} \prod_{j=1}^{p} f_{ij}(s_{ij}:h,k)$$

$$= \prod_{i,j} \{f(s_{ij},\theta_0)I_0[i,j] + f(s_{ij},\theta_1)I_1[i,j]\}.$$

In view of the discussion at the beginning of the section, we will let $f_S(\underline{s}:h,k)$ be the conditional density of the sensor image given that the correct overlay point is (h,k), i.e., S_{11} is placed over t_{hk}. The likelihood function, which we take to be the logarithm of $f_S(\underline{s}:h,k)$, is

$$L(\underline{s},h,k) = \sum_{i,j} I_0[i,j]\log f(s_{ij},\theta_0) + \sum_{i,j} I_1[i,j]\log f(s_{ij},\theta_1).$$

Thus the maximum likelihood estimate of the correct overlay point is a point (a,b) for which

$$L(\underline{s},a,b) = \max\{L(s,h,k): 1\le h\le n,\ 1\le k\le m\}.$$

Novak assumes each S_{ij} is exponentially distributed, and we shall do the same. Letting $f(s,\theta) = \exp\{-s/\theta\}/\theta$ ($\theta>0$, $s\ge 0$), $\theta_0=m_0$, and $\theta_1=m_1$, we get

$$L(\underline{s},h,k) = -\{\log m_0 \sum_{i,j} I_0[i,j] + (1/m_0) \sum_{i,j} I_0[i,j]s_{ij}$$

$$+ \log m_1 \sum_{i,j} I_1[i,j] + (1/m_1) \sum_{i,j} I_1[i,j]s_{ij}\}$$

$$= -\{n_0 \log m_0 + g_0/m_0 + n_1 \log m_1 + g_1/m_1\},$$

where $n_\alpha = \sum_{i,j} I_\alpha[i,j]$, and $g_\alpha = \sum_{i,j} I_\alpha[i,j]s_{ij}$, $\alpha=0,1$.

There are two nuisance parameters in the likelihood function, m_0 and

m_1. To eliminate them, we replace them with their MLE's conditioned on their correct overlay point being (h,k). Setting partial derivatives to zero,

$$\partial L/\partial m_\alpha = -n_\alpha/m_\alpha + g_\alpha/m_\alpha^2 = 0, \text{ or } \hat{m}_\alpha = g_\alpha/n_\alpha \ (\alpha=0,1).$$

Replacing m_α by $\hat{m}_\alpha$,

$$L(\underline{s},h,k) = -\{n_0 \log(g_0/n_0) + n_1 \log(g_1/n_1) + n_0 + n_1\}.$$

Note that $n_0+n_1 = \sum_{i,j} I_0[i,j] + \sum_{i,j} I_1[i,j] = qp$, a constant. Hence maximizing $L(\underline{s},h,k)$ is equivalent to minimizing

$$n_0 \log(g_0/n_0) + n_1 \log(g_1/n_1) = \log\{(g_0/n_0)^{n_0}(g_1/n_1)^{n_1}\},$$

which in turn is equivalent to minimizing

$$(g_0/n_0)^{n_0}(g_1/n_1)^{n_1} = \hat{m}_0^{n_0}\hat{m}_1^{n_1}.$$

Section 4.3 Conclusions

The statistic arrived at in this paper is different from Novak's edge ratio statistic. It would be of interest to compare their performance on Novak's problem, even though this would entail altering the MLE statistic. An analysis of the asymptotic behavior of the MLE statistic, as well as a comparison of the MLE of the overlay point for a variety of distributions, would shed light on the practicality of our approach to the problem. The elimination of the nuisance parameters in the maximum likelihood estimate requires justification. We hope to soon complete our analysis of the effect of replacing these parameters by their conditioned MLE's. Our results indicate that the convergence to the true parameter values is exponential, thus providing a high level of confidence in the estimates.

Section 4.4 Interpolation Experiments

The maximum likelihood estimation procedure for detecting corner location on a pixel level, suggests the possibility of extending this analysis to give a maximum likelihood subpixel estimate for corner or intersection detection. We intend to examine this possibility in the second phase of our work. As a prelude to this work, we performed experiments to determine the subpixel accuracy attainable using interpolation of the correlation function with the synthetic corner images. The results of those experiments will be compared with the maximum likelihood estimates obtained in future work.

The generated imagery consisted of a dark rectangle (as in Sec. 4.1) forming the upper right hand quadrant of the image. The rectangle was shifted in the x and y directions by uniform random shifts of less than a pixel. The rectanble was then rotated by 0°, 22.5°, and 45° to give three types of reference images. Grey-levels in the dark and light regions were generated from Gaussian distributions with different means. Gaussian noise was then added to the entire image. A 20x20 reference image and a 15x15 sensor image were used. The sensor image was correlated against the reference image to get correlation points in a 5x5 neighborhood of the center pixel. A biquadratic polynomial was then fit to this neighborhood and the peak of the polynomial was located. For each rectangle angle 0°, 22.5° , 45°, one hundred offsets were generated and the offset was estimated using the above procedure. The mean and variance of the error were computed. The mean and variance of the error, assuming the center of the pixel was the estimate, were also computed. The results are given in Table 4.1 Note that in each

case the interpolation gave a larger mean error than that obtained by selecting the center of the pixel.

The results of this limited experimentation indicate that even at low noise levels, the interpolation procedure provides low accuracy on the model imagery. During the follow-on work, we wish to extend the maximum likelihood estimates to the subpixel case and compare with these experimental results. We then wish to extend these results to edge images obtained from these synthetic images. The failure of interpolation in the experiments should not be viewed as a condemnation of the methods, for much of the application of these methods is on edge-enhanced imagery.

Section 5.0 A Comparison of Correlation, LSE and MLE for Image Matching

The most common methods of image matching are least squares estimation, maximum likelihood estimation, and correlation. Authors in the field (e.g., [Ho-Ba] and [Pr]) often claim that for their applications, two or more of these methods can be assumed equivalent. The exact conditions under which these equivalences hold are seldom presented. This paper is written to fill this lacuna.

For the purpose of conciseness, all definitions contained in this paper are presented forthwith. They are taken from [Ka-Ta] and [Ro-Ka].

A function, call it R, from the xy-plane to the real line is a discrete random field if at each lattice point (i,j) of the plane, R(i,j) is a random variable defined on the probability space (Ω,F,P). Thus at each (i,j), R(i,j) is a function from Ω to the real line. This can be made explicit by denoting R as a function of three variables $R(i,j,\omega)$, where $\omega \in \Omega$. At each (i,j) the expectation of R(i,j) is

$$E[R(i,j)] = \int_{\Omega} R(i,j,\omega)\, dP(\omega)$$

Since no confusion can arise from deleting ω, from now on we denote R as a function of two variables only.

The discrete random field R is homogeneous (or wide-sense stationary) if

(i) $E[R(i,j)] = \mu < \infty$, where μ is independent of (i,j)

and

(ii) for all integers $i_1, i_2, j_1, j_2, \alpha$, and β,

$$E[R(i_1,j_1)R(i_2,j_2)] = E[R(i_1+\alpha,\ j_1+\beta)\ R(i_2+\alpha,\ j_2+\beta)] < \infty.$$

It follows from (ii) that there is a function r, which depends only on α and β, such that

$$r(\alpha,\beta) = E[R(i+\alpha,\ j+\beta)\ R(i,j)], \qquad (1)$$

for all (i,j).

Let B be a bounded region of the xy-plane and n the number of lattice points in B, and suppose B grows to eventually encompass the entire plane. The homogeneous random field R is called correlation ergodic if, for every integer pair (α,β),

$$r(\alpha,\beta) = \lim_{n\to\infty} \frac{1}{n} \sum_{(i,j)\in B} R(i,j)R(i+\alpha,j+\beta).$$

The convergence is in probability. This means that the product moment of R(i,j) and $R(i+\alpha,j+\beta)$ (often called the auto-correlation), can be approximated by taking the average, shown on the right side of the equation, over a sufficiently large bounded region.

The restriction of a discrete random field T to a bounded region B is called an image. Usually the region of restriction, B, need not be explicitly indicated, so to increase the readability of equeations, the same symbol, say R, will be used to denote an image and the discrete random field from which the image is derived. Hence B_R refers to the region of restric- of the image R. The value that the random variable $R(i,j),(i,j)\in B_R$, assumes is called the grey level of the image R at the point (i,j). Before continuing, we point out that throughout this paper, the variables i,j,α, and β can assume integer values only. Lastly, the notation $|B_R|$ refers to the number of lattice points in the region B_R.

Suppose R and S are images with $|B_S| << |B_R|$ - we call R the reference image and S the sensor image. The least squares estimate (LSE) of the match point between S and R is any point (α,β) at which

$$L(\alpha,\beta) \equiv \sum_{(i,j)\in B_S} [S(i,j) - R(i+\alpha,j+\beta)]^2 \qquad (2)$$

is a minimum. The correlation estimate of the match point is any point (α,β) at which

$$C(\alpha,\beta) \equiv \frac{\sum_{(i,j)\in B_S} S(i,j)R(i+\alpha,j+\beta)}{\left[\sum_{(i,j)\in B_S} S^2(i,j) \sum_{(i,j)\in B_S} R^2(i+\alpha,j+\beta)\right]^{1/2}} \tag{3}$$

is a maximum.

In practice, it is desirable to render the correlation estimate independent of a uniform shift in the grey level of either S or R. Hence correlation is applied to S' and R',

$$S'(i,j) \equiv S(i,j) - \frac{1}{n_S} \sum_{(i,j)\in B_S} S(i,j)$$

and,

$$R'(i,j) \equiv R(i,j) - \frac{1}{n_R} \sum_{(i,j)\in B_R} R(i,j),$$

where n_S is the number of pixels (i.e., lattice points) in B_S and n_R is the number in B_R. This transformation is presumed throughout the remainder of this paper.

Section 5.1 Correlation and LSE

A sufficient condition that (3) and (4) give rise to the same match point is that

$$\sum_{(i,j)\in B_S} R^2(i+\alpha,j+\beta) \tag{4}$$

be constant in (α,β).

[Ho-Ba] and [Pr] claim that if (4) varies slowly as the sensor image S mover over R, then (4) can be assumed essentially constant and ignored. The vagueness of the condition 'varying slowly' can be replaced with the rigor of the following definition.

We say the discrete random field R is almost constant if for each $\varepsilon > 0$ there exists some integer M such that for every $n > M$ and for every bounded region B with n lattice points

$$P\left\{\left|r(0,0) - \frac{1}{n} \sum_{(i,j)\in B} R^2(i,j)\right| > \varepsilon\right\} < \varepsilon. \qquad (5)$$

Note that (5) requires that (4) be a convergent (in P) sequence for all (α,β), and that the rate of convergence be uniformly (in (α,β)) bounded from below.

If R is an uncorrelated and identically distributed discrete random field with finite fourth moments - i.e., for all (i,j) $E[R^4(i,j)] < \infty$ - then R is homogeneous, and by (1) $E[R^2(i,j)] = r(0,0)$, a constant. By a variant of the Law of Large Numbers (see [Ch]), R is almost constant.

It is worthwhile pointing out that a homogeneous random field R is not necessarily almost constant, and in fact even if R is correlation ergodic it need not be almost constant.

Suppose R is an almost constant discrete random field and S is any image. Clearly the LSE and correlation estimate of the match point of R and S need not be the same. However, LSE and correlation are equivalent in the sense of the following theorem. A few definitions first. $\vec{\chi}_c$ (a two-dimensional vector) is a point at which the correlation formula (3) attains its maximum, and $\vec{\chi}_s$ is a point at which the least squares formula (2) attains its minimum.

Theorem 5.1

If there is an $\varepsilon > 0$ such that

$$P\left\{C(\vec{\chi}_c) > \left(\frac{4r + \varepsilon}{4r - \varepsilon}\right)^{1/2} C(\vec{\chi}) \right\} > 1 - \frac{1}{4}\varepsilon \qquad (6)$$

for all $\vec{\chi} \neq \vec{\chi}_c$ (note $r \equiv r(0,0)$)

and

$$P\left\{L(\vec{\chi}_s) < L(\vec{\chi}) - \frac{1}{2} n \varepsilon \right\} > 1 - \frac{1}{4}\varepsilon \qquad (7)$$

for all $\vec{\chi} \neq \vec{\chi}_s$,

and if $n \equiv |B_s|$ is sufficiently large to satisfy (5) with ε replaced by $\varepsilon/4$, then

$$P(\vec{\chi}_c = \vec{\chi}_s) > 1 - \varepsilon.$$

Proof

Let $\vec{\chi}_o$ be a point at which

$$U(\alpha,\beta) \equiv \sum_{(i,j)\in B_s} S(i,j)R(i+\alpha, j+\beta)$$

attains its maximum, and define the following sets:

$$\Psi \equiv \left|r - \frac{1}{n} \sum_{(i,j)\in B_s} R^2(i,j)\right| < \frac{1}{4}\varepsilon \quad ,$$

$$\Lambda_a \equiv (\vec{\chi}_o \neq \vec{\chi}_a), \quad a = s,c,$$

$$\Phi_s \equiv (L(\vec{\chi}_s) < L(\vec{\chi}_o) - \tfrac{1}{2} n\varepsilon),$$

$$\Phi_c \equiv C(\vec{\chi}_c) > \left(\frac{4r+\varepsilon}{4r-\varepsilon}\right)^{1/2} C(\vec{\chi}_o) \quad ,$$

and $\Gamma_a \equiv \Lambda_a \cap \Phi_a \cap \Psi, \quad a = s,c.$

On the set Γ_s,

$$\begin{aligned} L(\vec{\chi}_o) &< n(r - \tfrac{1}{4}\varepsilon) - 2U(\vec{\chi}_o) + \tfrac{1}{2} n\varepsilon \\ &\leq n(r - \tfrac{1}{4}\varepsilon) - 2U(\vec{\chi}_s) + \tfrac{1}{2} n\varepsilon \\ &< L(\vec{\chi}_s) + \tfrac{1}{2} n\varepsilon < L(\vec{\chi}_o). \end{aligned}$$

It follows from this contradiction that $P(\Gamma_s) = 0$, hence $P(\Lambda_s) < \frac{1}{2}\varepsilon$.

On the set Γ_c,

$$C(\vec{\chi}_o) > \frac{U(\vec{\chi}_o)}{[n(r + \frac{1}{4}\varepsilon)]^{1/2}} > \frac{U(\vec{\chi}_c)}{[n(r - \frac{1}{4}\varepsilon)]^{1/2}} \cdot \left(\frac{4r-\varepsilon}{4r+\varepsilon}\right)^{1/2}$$

$$> C(\vec{\chi}_c)\left(\frac{4r-\varepsilon}{4r+\varepsilon}\right)^{1/2} > C(\vec{\chi}_o).$$

Thus $P(\Gamma_c) = 0$, implying $P(\Lambda_c) < \frac{1}{2}\varepsilon$.

Combining the above results we get:

$$\begin{aligned} P(\vec{\chi}_s = \vec{\chi}_c) &\geq P(\vec{\chi}_s = \vec{\chi}_o = \vec{\chi}_c) \\ &= P\{(\vec{\chi}_s = \vec{\chi}_o) \cap (\vec{\chi}_o = \vec{\chi}_c)\} \\ &> 1 - P(\Lambda_c) + 1 - P(\Lambda_s) - 1 \\ &> 1 - \varepsilon. \end{aligned}$$

This theorem imposes rather strong conditions on the points $\vec{\chi}_c$ and $\vec{\chi}_s$, however with monor modifications to the proof, these conditions can be weakened. It is sufficient that the two probabilistic inequalities (6) and (7) be true, respectively, on neighborhoods of $\vec{\chi}_c$ and $\vec{\chi}_s$.

Note that no restrictions have been placed on S. Hence the equivalence of LSE and correlation hold in the case

$$S(i,j) = R(i+\alpha, j+\beta) + N(i,j), \quad (i,j)\varepsilon B_s$$

where the noise N(i,j) are iid, and R satisfies the conditions of the theorem.

We turn our attention to a reference image R containing two homogeneous regions with means μ and ν. If the sensor image S is offset by (α,β), some of the sensor pixels will overlay region I of R and the remainder of the sensor pixels will overlay region II. The following shorthand notations will be used, in which, from context, it is understood the offset is (α,β):

$$\Sigma_I \equiv \sum_{\{(i,j):\ (i+\alpha,\ j+\beta)\ \varepsilon \text{ region I}\}}$$

and

$$\Sigma_{II} \equiv \sum_{\{(i,j):\ (i+\alpha,\ j+\beta)\ \varepsilon \text{ region II}\}}$$

In this case the LSE is given by the minimum of

$$L(\alpha,\beta) = \Sigma_I R^2(i+\alpha, j+\beta) + \Sigma_{II} R^2(i+\alpha, j+\beta)$$

$$+ \sum_{(i,j)\varepsilon B_s} S^2(i,j) - 2\Sigma_I S(i,j) R(i+\alpha, j+\beta)$$

$$- 2\Sigma_{II} S(i,j) R(i+\alpha, j+\beta),$$

and the correlation estimator by the maximum of

$$C(\alpha,\beta) = \frac{\Sigma_I\, S(i,j)R(i+\alpha,\, j+\beta) + \Sigma_{II}\, S(i,j)R(i+\alpha,\, j+\beta)}{\left\{\sum_{(i,j)\in B_s} S^2(i,j)[\Sigma_I R^2(i+\alpha,\, j+\beta) + \Sigma_{II} R^2(i+\alpha,\, j+\beta)]\right\}^{1/2}}$$

As before we require that

$$D(\alpha,\beta) = \Sigma_I R^2(i+\alpha,\, j+\beta) + \Sigma_{II} R^2(i+\alpha,\, j+\beta)$$

be approximately constant with respect to (α,β) in order to ensure that LSE and correlation give rise to the same match point. This condition is satisfied in the following circumstances.

If S is large, then shifting the offset by a few pixels will not drastically alter $D(\alpha,\beta)$, since the set of pixels included in $D(\alpha,\beta)$ remains essentially unchanged. But the cross-product term (the numerator in the correlation function) will change, because all product terms are different.

Thus in a neighborhood of say, the LSE match point, LSE and correlation will result in the same solution.

Suppose R is restricted to being a binary image, whereas S remains a grey level image. If R consists of two contiguous regions, then matching S and R is euqivalent to finding an edge, of known shape and size, in S. This edge separates two homogeneous regions with different mean grey levels.

We can assign to the pixels of R a conditional estimate of the means in each region. If S is offset by (α,β), then the conditional estimate of μ and ν are

$$\mu_{\alpha\beta} \equiv \frac{1}{n(I)}\, \Sigma_I\, S(i,j)$$

and

$$\nu_{\alpha\beta} \equiv \frac{1}{n(II)}\, \Sigma_{II}\, S(i,j)$$

Here n(I) and n(II) are the number of pixels of the sensor image which overlay, respectively, regions I and II of R at offset (α,β). The dependency of n(I) on (α,β) has been suppressed, although it is implicitly understood. Thus, at offset (α,β), the reference pixels in region I are assigned the

value $\mu_{\alpha\beta}$, and those in region II $\nu_{\alpha\beta}$.

The correlation at offset (α,β) is given by

$$C(\alpha,\beta) = \frac{[n(I)\mu_{\alpha\beta}^2 + n(II)\nu_{\alpha\beta}^2]^{1/2}}{[\sum_{(i,j)\in B_s} S^2(i,j)]^{1/2}},$$

and the LSE is the minimum of

$$L(\alpha,\beta) = \sum_{(i,j)\in B_s} S^2(i,j) - n(I)\mu_{\alpha\beta}^2 - n(II)\nu_{\alpha\beta}^2.$$

In this instance, then, correlation and LSE are equivalent.

Section 5.2 LSE and MLE

An advantage of least squares estimation and correlation estimation is that they are distribution independent, whereas maximum likelihood estimation is highly distribution dependent. In order to use MLE, more stringent requirements must be imposed on the underlying model, often rendering it less realistic.

Suppose S and R are related as follows,

$$S(i,j) = R(i + \alpha, j + \beta) + N(i,j)$$

where (α,β) is the offset we seek, and the $N(i,j)$ are iid Gaussian with mean 0 and variance σ^2. The log likelihood function of S is

$$-\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{(i,j)\in B_s}[S(i,j) - R(i + \alpha, j + \beta)]^2.$$

This expression attains a maximum when

$$\sum_{(i,j)\in B_s}[S(i,j) - R(i + \alpha, j + \beta)]^2$$

attains a minimum. This is, of course, the LSE. Note that R is not a random field, and because at the correct offset (α_0,β_0)

$$E[S(i,j)] = R(i + \alpha, j + \beta),$$

S is, in general, not a homogeneous random field.

If R is a binary image, as described at the end of Section 5.1, then MLE is equivalent to LSE which is equivalent to correlation.

Section 6.0 Conclusions and Future Work

We have developed statistical and geometric models for subpixel accuracy. Using a restrictive geometric model, we were able to derive bounds on subpixel accuracy. These bounds are useful for both error prediction and for selection of features for registration. Under the assumptions of our model, a high level of subpixel accuracy is possible. We are currently extending these results to more realistic models.

Several bounds on registration accuracy were derived under the assumption of statistical models for the images and noise. Two cases were considered. First, the reference and sensed images were assumed to be continuous and bounds on the offset error were derived. In the second model, it was assumed that the image was digitized and that a registration to the correct pixel is available. In addition, a consistent maximum likelihood estimator was developed for corner detection under a stochastic model for such features. Finally, conditions were established under which maximum likelihood, correlation and least squares methods for image matching are equivalent.

The extension and testing of our geometric modeling methods will be a key part of our continuing work. The level of subpixel accuracy attainable under our restricted model was sufficiently high to warrant detailed investigation of less restrictive models. For the case of the digitization of a real line, we will complete the probabilistic analysis using the invariant measure on lines for several lengths of digital lines. This will give more realistic information on subpixel accuracy. The subpixel accuracy attainable will be shown to be even better than

our present results indicate since we have chosen a worst case bound. We will also examine the case of a digital angle formed by two digital lines intersecting at a specified angle. Once again, this situation, which models road intersections, can only improve the subpixel accuracy.

The case of digital lines with points missing will first be investigated experimentally. Using the methods outlined in Section 2.6, we can compute bounds on the offset estimation error. The bounds derived in this manner will be adequate to describe this more general model, but we will attempt to model this situation to aid us in the still more general models. Our most general model in which points are missing and extraneous points are added to the digitalline will be investigated next. The exact form of this study will depend upon the previous results.

We will experiment with LANDSAT and simulated data to estimate the accuracy to which we can detect the pixels on a digital line. Using these observations we will develop each accuracy model to be used in evaluating the set of all digital lines to determine procedures for selection of good registration features, e.g., which line slopes are best.

Experimentation will be necessary to determine the usefulness of the statistical bounds developed in Section 3. We review briefly the proposed work in this direction.

(i) For large continuous sensed images with conditionally Gaussian noise given the reference image (see Appendix), formula (A) in Section 3 bounds the probability of misregistration by more than distance τ. Numerical work with pixel-discretized real and simulated images is needed to test the validity and usefulness of the bound.

(ii) When translation-registration to the nearest pixel has already been accomplished, and all imagery can be assumed spatially homogeneous with rapidly decaying correlations on the pixel distance-scale h, the estimator $\hat{\kappa}$ from Section 3 approximately limits the subpixel accuracy possible if the sensed and reference images were infinitely large with noise- and reference-images stochastically independent. Again, numerical experimentation will empirically determine whether these assumptions and fig ures-of-meritare valid or useful.

(iii) The kriging-interpolation and maximization of $C(\cdot)$ should certainly be tried, as sketched at the end of Section 3, using simple parametric forms for the covariances of Z_R and Z_S.

(iv) Finally, if the experiments in (i) - (iii) prove successful, theoretical and empirical extensions of this work, to the case of registration with respect to affine distortion conditions of [Mo-Sm], seem both desirable and possible.

The corner detector, used to locate a highly reliable match point for registration will be studied. This study will consist of analytical modeling for subpixel accuracy as well as experimental studies.

References

[Ad] Adler, R., The Geometry of Random Fields, (Wiley, New York, 1981).

[Ch] Chung, K.L., A Course in Probability Theory (Academic Press, New York, n1974)

[De] Deo, C., A functional central limit theorem for stationary random fields, Ann. Prob. 3 (1975) 708-715.

[Do] Dorst, L., On digitized straight line segments, Internal. report, Delft, 1982.

[Do-Sm] Dorst, L. and Smeulders, A.W.M., The estimation of parameters of digital straight line segments, Proceedings 6th International Conference on Pattern Recognition (Munich 1982) 601-603.

[Du] Duda, R., Kriging with derivative information, Fairchild Industries, Tech. Report 1982.

[Du-Ha] Duda, R. and Hart, P.E., Pattern Classification and Scene Analysis, (Wiley Interscience, New York, 1973).

[Ha-Wr] Hardy, G.H. and Wright, E.M., An Introduction to the Theory of Numbers, (Oxford at the Clarendon Press, 1971).

[Ho-Ba] Horn, B.K.P. and Backman, B.L., Using synthetic images to register real images with surface models, Comm. A.C.M. 21 (Nov. 1978) 914-924.

[Hy-Da] Hyde, P. and Davis, L., Subpixel Edge Estimation, University of Maryland, Tech. Report 1164, 1982.

[Ka] Kanal, L., Research related to automated multisensor image registration, N.A.S.A. Final Report, NAG-154, 1982.

[Ka-Ta] Karlin, S. and Taylor, H.M., A first course in stochastic processes, (Academic Press, New York, 1975).

[Ma] Marcus, M., A bound for the distribution of the maximum of continuous Gaussian processes, Ann. Math. Statist. 41 305-309.

[Mo-Sm] Mostafavi, H. and Smith, F., Image correlation with geometric distortion. Part I: Acquisition performance. Part II: Effect on local accuracy, IEEE Trans. Aerospace Elec. Sys. AES-14, 487-493; 494-500.

[Pr] Pratt, William K., Digital Image Processing, (Wiley-Interscience, New York, 1978).

[Ri] Ripley, B., Spatial Statistics (Wiley, New York, 1981).

[Ro-Ka] Rosenfeld, A. and Kak, A.C., Digital Picture Processing, (Academic Press, 1976).

[Ro-We] Rothstein, J. and Weiman, C., Parallel and sequential specification of a context sensitive language for straight lines on grids, Computer Graphics and Image Processing 5 (1976) 106-124.

[Wa] Warner, F., Differentiable Manifolds and Lie Group, (Scott-Foresman, Illinois, 1982).

SIMULATION ASPECTS IN THE STUDY OF RECTIFICATION OF SATELLITE SCANNER DATA

Edward M. Mikhail
and
Fidel C. Paderes, Jr.

Purdue University

ABSTRACT

Complete sensor/platform modelling is derived and used for the generation of synthetic data and for rectification studies of satellite scanner data. All satellite position and sensor attitude parameters are recovered. Rectification accuracy improves marginally when using more than 25 control points, and is highly sensitive to errors in image point identification.

1. INTRODUCTION

1.1 General

Remote sensing imagery produced by various sensors, such as frame cameras, scanners, etc., may be considered as a transformation of the object space, e.g. ground surface, into the image space which may be a plane, a cylindrical surface, etc. Scanner imagery, with which this paper is concerned, is the result of transforming the three-dimensional ground surface into equivalent cylindrical surface, which when developed becomes a two-dimensional image space.

Rectification is essentially the process of defining the inverse transformation which will allow us to recover the ground surface from corresponding imagery. We can fully recover the ground surface from imageries only if we have multiple coverage of the same ground area from different acquisition locations. Since the inverse transformation is from a two-dimensional surface (the imagery) into a three-dimensional surface (the ground), rectification using single coverage imagery requires that one of the three-dimensions of the ground space, usually the elevation, be known or assumed known a-priori.

Another process, which is very similar to rectification, is registration. In rectification, we determine the ground position of points in a given imagery, while in registration, we locate these points on other imageries covering the same area. The effectiveness of registration depends on how close to each other are the acquisition points of the different imageries. Because rectification and registration are very similar, methods suited for one can be applied to the other, with slight modifications.

As is well known, an imagery consists of picture elements called pixels. If the position of the exposure station, i.e. the platform (satellite) position, and the direction of the vector from the exposure station to the pixel are known, its ground position can be derived. In general, every pixel is imaged at a different time, hence a given pixel has a unique exposure station and a unique vector direction. If the satellite position corresponding to all pixels and if all pixel directions are known to the required accuracy, the problem of rectifying an image is solved. Unfortunately, either because of cost, because it is not technically possible, or both, the position of the satellite, or the ephemeris, and the direction of pixel vectors are not available with the required accuracy.

An alternative procedure for rectifying imagery, is through the use of ground control points. These are points the positions of which are known both in the imagery and on the ground. A mathematical model exists which relates the position of a point on the imagery, the corresponding satellite position, pixel vector direction and ground position. Suppose there are points with known positions both in the imagery and on the ground (control points); then presumably, using the mathematical model, we can solve for the satellite position and the pixel vector direction. This is only possible if the satellite position and pixel vector directions are expressed in parametric form since each pixel has a unique direction and a unique corresponding satellite position. A pixel vector direction can be broken down into two components, namely, the attitude or orientation of the sensor coordinate system and the direction of the pixel with respect to the sensor

coordinate system which is usually observed. Then only the sensor attitude need be modelled in parametric form.

This approach to rectification has three main elements: (1) the type of mathematical model used, (2) the method of adjustment used, and (3) the manner in which a-priori information is exploited.

1.2 Mathematical Models Used for Rectification

The two main types of models are the implicit and explicit models. The implicit model relates the point on the imagery to the corresponding point on the ground using parameters that have no direct physical significance, i.e., satellite position and sensor attitude cannot be derived from these parameters. These types of models are more commonly known as interpolative or surface fitting models. The explicit model, on the other hand, relates the point on the imagery to the point on the ground using parameters that have real physical meaning. These parameters include either the satellite position and sensor attitude themselves, or other parameters which are related to them. The group of explicit models are commonly known as parametric models. Each of the two types of models is discussed separately.

1.2.1 Interpolative or Surface Fitting Models

The most commonly used model of the interpolative type is the polynomial function. This includes similarity, affine and higher order polynomials. Normally, the ground is first projected into a mapping plane. If necessary, the image is also projected into an equivalent plane. The general form of the polynomial function is as follows:

$$(1.1a) \quad X = a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2 + a_6x^2y + a_7xy^2 + \ldots.$$

$$(1.1b) \quad Y = b_0 + b_1x + b_2y + b_3xy + b_4x^2 + b_5y^2 + b_6x^2y + b_7xy^2 + \ldots.$$

where X,Y are the map coordinates, x,y are the image coordinates (or pixel locations) and a_0, b_0, a_1, b_1, a_2, b_2, ... are the mapping parameters. Polynomials are global in the sense that only one set of parameters is used for the whole image frame.

If the density of the control points is high, global functions might not be appropriate. Then the frame might be divided into segments and a different polynomial function applied to each segment. If conditions of continuity are inforced at the boundary of the different segments, the approach becomes known as the method of splines.

A totally different approach applicable also if the control point density is high, is the method of moving averages. In this method a different polynomial is used for every point to be interpolated. Each polynomial is centered on the point of interest. The degree of each polynomial might be low and the effective area might be small but still this method is computationally expensive.

After rectifying an image, the residuals or differences between computed and observed coordinates of control points, can be calculated. Again if the density of the control points is high, it may be desirable to perform additional processing to reduce the magnitude of the residuals. The method of linear least squares prediction is best suited for this type of processing. The method assumes that the residuals belong to a random field.

1.2.2 Parametric Models

Parametric models follow closely the geometric and physical processes which produced the imagery. Because of this, parametric modelling can be logically divided into sensor modelling and platform modelling. Parametric models also depend on the assumed figure of the earth surface.

Sensor models reflect the type of sensor used. They are independent of the platform (satellite) used and the type of surface being imaged (e.g. earth). The results of sensor modelling are either corrected sensor vector directions corresponding to each pixel, or pixel positions projected on a plane. For scanner type sensor, projection of pixel positions on a plane corrects for the panoramic effect. Other corrections applicable are due to non-linearity of scanning, unequal number of pixels per scan, and the effect of scan line corrector (for Thematic Mapper Only). Sensor modelling is sometimes known as internal modelling.

The platform model describes the behavior of the satellite which is the platform for imaging. Platform modelling primarily consists of two parts: sensor attitude modelling and satellite position and orbit modelling. Attitude models can be polynomials, harmonic series or auto-regressive models. The independent parameter for attitude models is usually time. Satellite position and orbit models can be grouped into three general types. The first group defines the satellite position in terms of the satellite position vector, and the satellite orbit is defined in terms of both the satellite position and velocity vectors. Both vectors vary with time. The second group defines the satellite orbit in terms of the five orbital parameters as defined in orbit

mechanics. In this case, these orbital parameters vary with time. The satellite position is defined in the orbital plane as a function of time. The third group is similar to the second in the sense that the satellite orbit is also defined in terms of orbital parameters and that the satellite position in the orbital plane is also defined as a function of time. The main difference is that in the last group, the orbital parameters are independent of time, i.e., they are constant for a given frame. As a consequence, the shape of the orbit has to be defined. The shape of the orbit can be assumed to be a straight line, a circle or an ellipse. As a further consequence for assuming the orbital parameters constant, the deviation of the actual satellite position from its computed position using the orbital parameters has to be modelled. Satellite position deviation models can be polynomials, harmonic series or auto-regression models similar to the attitude models. Again these models are functions of time.

The last element in parametric modelling pertains to the assumed shape of the earth. The shape of the earth is important because no computation can be done on its surface unless its shape is known. For purposes of rectification, the surface of the earth can be a map projection plane, a sphere or an ellipsoid.

1.2.3 Other Model Considerations

Given a selected model with redundant data, an adjustment method is applied. There are two types of adjustment currently in use: the least squares method and Kalman filter approach. The former is a batch type of adjustment. All observations are adjusted in one pass and the parameter estimates are then computed. Inherent in this method is the

assumption that the model is fixed. The second approach is inherently sequential in nature. Observations are incorporated into the adjustment in small groups. The precision of the parameter estimates increases up to a certain limit as the number of observations incorporated into the adjustment increases. The model used in this adjustment is considered random, hence it gets adjusted together with the observations.

During rectification adjustment using ground control points, the sensor attitude and satellite position parameters are unknown. In reality, some or all of these parameters may be measured but to a precision which is inadequate for rectification. These measurements, and others that are related to them, constitute a-priori information. Instead of using these measurements as initial approximations for the corresponding parameters, they are used as a-priori estimates with proper a-priori covariance matrices. In this manner, they are allowed to vary in the adjustment. The amount of variation is commensurate with the a-priori variances and covariances.

1.3 Review of Literature

The earliest and easiest approach to rectification of satellite scanner data, is the use of polynomial models. Many authors have reported that the resulting accuracy is comparable to other methods (Forrest [10], Trinder [20], Bähr [3], Dowman [7]). Because of its reported accuracy and ease of use, polynomials are still presently the most commonly used rectification method.

The earliest parametric model applied to satellite scanner data assumes that the orbit is a straight line and that the earth surface is projected onto a mapping plane (Kratky [12], Konecny [11], Dowman

[7]). In effect, the treatment of satellite scanner data is the same as that of aircraft scanner data. Parameters describing the variations in attitude and elevations were recovered.

The next improvement in parametric modelling is due to Caron and Simon. They defined the satellite orbit and position in terms of satellite position and velocity vectors (Caron and Simon [6], Puccinelli [16]). They also did away with the use of map projection during the adjustment process. They assumed instead that the earth is a sphere and performed computations on its surface (Caron and Simon [6], Bähr [4], Sawada [18]). The parameters recovered during the adjustment were the same as those in the previous method. They are further credited with the use of Kalman filter to solve for the parameters in the adjustment.

Bahr was the first to define the satellite position in terms of orbital parameters that are functions of time (Bähr [4]). He recommended that only parameters describing the attitude and elevation variations should be recovered.

Next the orbit was defined in terms of constant orbital parameters. This assumption requires that the shape of the orbit be defined and that the deviation of the actual satellite from its predicted position be modelled in terms of time. The shape of the orbit had been modelled as a circle (Forrest [9], Levine [13], Synder [19]) and as an ellipse (Bähr [4], Sawada [18]). Only Levine so far has incorporated in his model all three components of satellite position deviation (Levine [13]). Like the others, however, he also recommended that only the parameters defining the variations in attitude and elevation be recovered.

Regarding the shape of the earth a few authors have recommended that an ellipsoid of revolution be used (Pucinelli [16], Forrest [9], Levine [13], Synder [19]). Because of the complex nature of the resulting formulas, no exact closed form have been derived so far. Computations on the surface of the earth's ellipsoid involving elevations as recommended by the above authors require approximations and/or iterations.

1.4 Preview of the Investigation

The parametric model derived for this investigation is suitable for Landsat MSS type imagery. With slight modification of its sensor dependent parameters, this model is also applicable to TM type imagery. It is sufficiently general as to encompass various specific cases published by other researchers. In addition, it extends the modelling of the satellite position to include all of its three components, while others have limited consideration to only one component, its elevation. With this general model, we are able to both generate synthetic data and study rectification. This model is also used to study the effect on ground position of target points due to both individual as well as combined errors in the various parameters.

The major factors affecting rectification accuracy are: (1) the type of model used, (2) density of ground control, (3) accuracy of ground control, and (4) the accuracy of the derived image coordinates or directions. Using synthetic data produced using the derived model we studied the effects of these different factors. The different cases of the model used are: (a) polynomial model, (b) model with straight line orbit and earth surface projected on a plane, (c) model with

circular orbit and spherical earth, (d) model with circular orbit and ellipsoidal earth, and (e) model with elliptical orbit and ellipsoidal earth. The last three models fully accounted for the satellite position deviation (three components) and the sensor attitude (three elements).

2. MATHEMATICAL MODELLING

2.1 Principles of Parametric Modelling

Figure 1 shows the geometry involved in the relationship between image and object spaces, where:

- X Y Z — is the ground coordinate system;
- x' y' z' — is the transformed sensor coordinate system parallel to the ground coordinate system;
- S — is the satellite position defined by the vector $[X_S \; Y_S \; Z_S]^T$;
- p — is the pixel position defined by $[x'_s \; y'_s \; z'_s]^T$;
- G — is the pixel ground position defined by the vector $[X_G \; Y_G \; Z_G]^T$;
- h — is the elevation of G, and
- N — is the radius of the prime vertical corresponding to G.

Since the two coordinate systems are parallel, then

$$(2.1) \qquad \begin{bmatrix} x'_p \\ y'_p \\ z'_p \end{bmatrix} = \lambda \begin{bmatrix} X_G - X_S \\ Y_G - Y_S \\ Z_G - Z_S \end{bmatrix}$$

where λ is a scale factor.

Let the original sensor coordinate system be x y z. This coordinate system is not necessarily parallel to the ground system. Let M^T be the transformation which rotates x y z into x' y' z'. Applying this transformation to the original pixel coordinates results in

$$(2.2) \qquad \begin{bmatrix} x'_p \\ y'_p \\ z'_p \end{bmatrix} = M^T \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix}$$

Substituting equation (2.2) into equation (2.1) produces the following

$$(2.3) \qquad \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \lambda M \begin{bmatrix} X_G - X_s \\ Y_G - Y_s \\ Z_G - Y_s \end{bmatrix}$$

This equation is called the collinearity equation.

The process of deriving the pixel position vector $[x_p\ y_p\ z_p]^t$ in the image space from pixel row and column numbers is called sensor modelling. The process of defining the satellite position vector $[X_s\ Y_s\ Z_s]^t$ in terms of orbital parameters, time and satellite position deviation parameters is called orbit modelling. Orbit modelling plus the process of defining M in terms of the orbital parameters, time, satellite position deviation parameters and sensor attitude parameters is called platform modelling.

Before we proceed, we will first list without proof formulas from related fields which we will need later in our derivations.

2.2 Formulas from Related Fields

Orbital mechanics provides us with the necessary formulas for establishing the position of satellite in orbit. The following

formulas assumes that the earth is a sphere of uniform mass.

(2.4a) $$E - \varepsilon_s \sin E = \sqrt{GM_e/A_s^3}\, t_s$$

(2.4b) $$\cos \nu = (\cos E - \varepsilon_s)/(1 - \varepsilon_s \cos E)$$

(2.4c) $$\sin \nu = \sqrt{1 - \varepsilon_s^2} \sin E/(1 - \varepsilon_s \cos E)$$

(2.4d) $$R = A_s (1 - \varepsilon_s \cos E)$$

(2.5) $$t_s = \sqrt{A_s^3/GM_e} \{ 2 \tan^{-1} [\sqrt{(1 - \varepsilon_s)/(1 + \varepsilon_s)} \tan (\nu/2)] - \varepsilon_s \sqrt{1 - \varepsilon_s^2} \sin (\nu/2)/[1 + \varepsilon_s \cos (\nu/2)] \}$$

(2.6) $$\tau = 2\pi \sqrt{A_s^3/GM_e}$$

See Figure 2 for aid in defining the terms:

A_s in the semimajor axis of the satellite orbit,

ε_s is the eccentricity of the satellite orbit,

R is the distance of the satellite from the earth's center,

ν is the true anomaly defined as the angle as viewed from the center of the earth between the satellite and the point on the satellite orbit nearest the earth (perigee),

t_s is time where t_s is zero at perigee,

τ is the period of the satellite orbit,

E is the eccentric anomaly,

G is the gravitational constant, and

M_e is the mass of the earth.

In Figure 2, O is the center of the orbit; P is the perigee; S is the satellite; F_1 and F_2 are the focii of the elliptical satellite orbit; F_1 coincides with the center of the earth; R is distance of

the satellite from the earth's center; A_s is the semi-major axis of the satellite orbit; and ν is the true anomaly.

Given t_s, the parameters A_s and ε_s, and the constants G and M_e, the polar coordinates R and ν of the satellite position can be solved for using equations (2.4a) to (2.4d). Equation (2.4a) has to be solved iteratively for E, the eccentric anomaly. Conversely given ν and the same set of constants, t_s can be solved for using equation (2.5).

The next field where other required formulas are available is geometric geodesy. The following formulas are useful for computing on the surface of the earth. The major assumption here is that the earth is an ellipsoid of revolution.

$$N = A_e / \sqrt{1 - \varepsilon_e^2 \sin^2 \phi} \tag{2.7a}$$

$$\delta N = e_e^2 N \tag{2.7b}$$

$$R_{avg} = A_e \sqrt{1 - \varepsilon_e^2} / (1 - \varepsilon_e^2 \sin^2 \phi) \tag{2.8}$$

$$X = (N + h) \cos \phi \cos \lambda \tag{2.9a}$$

$$Y = (N + h) \cos \phi \sin \lambda \tag{2.9b}$$

$$Z = (N + h - \delta N) \sin \phi \tag{2.9c}$$

Figures 3a and 3b will help clarify the following terms:

A_e is the semi-major axis of the ellipsoid,

ε_e is the eccentricity of the ellipsoid,

ϕ is the geodetic latitude,

λ is the geodetic longitude,

h is the elevation of a point,

N is the radius of the prime vertical,

δN is that part of N below the equator for points in the

northern hemisphere and above the equator for points in the southern hemisphere

R_{avg} is the average radius of curvature of a point on the earth's surface, and

$[X\ Y\ Z]^t$ is the vector defining the position of a point.

Map projections is the last area where necessary formulas can be found. Although other types of projection may be applicable, only one type, namely, the oblique Mercator projection, was arbitrarily chosen. The main assumption here is that the earth is a sphere.

$$(2.10a)\quad U = -R \tan^{-1}\left[\frac{\sin(\lambda - \lambda_p)\cos\phi}{\cos\phi\cos\phi_p\cos(\lambda - \lambda_p) - \sin\phi_p\sin\phi}\right]$$

$$(2.10b)\quad V = -\tfrac{1}{2} R \log\left[\frac{1 + \sin\phi\sin\phi_p + \cos\phi\cos\phi_p\cos(\lambda - \lambda_p)}{1 - \sin\phi\sin\phi_p - \cos\phi\cos\phi_p\cos(\lambda - \lambda_p)}\right]$$

Figures 4a and 4b are included for clarification of the following symbols:

ϕ_p and λ_p are the latitude and the longitude respectively of the projection pole P; the projection pole is the point of intersection with the sphere of a line normal to the central circle and passing through the earth's center;

ϕ and λ are the latitude and longitude, respectively, of the point to be projected;

U and V are the resulting map coordinates after projection; and

R is the radius of the best fitting tangent sphere to the earth surface at the point of interest.

2.3 Sensor Modelling

The main purpose of sensor modelling is to recover the true direction of the pixel vector at the moment of pixel imaging with respect to the sensor coordinate system. The sensor coordinate system is arbitrary, but for sensors of the scanner type, the following is a convenient coordinate system (see Figure 5). The origin coincides with the perspective center of the sensor optical system; the z-axis bisects the scanning angle and is positive away from the object; the y-axis is parallel to, and positive in, the scanning direction and it is also perpendicular to the z-axis; the x-axis completes a right handed coordinate system. In Figure 5, O is the origin, 2 α is the scan angle, and the x-, y-, and z-axes are as shown. Every scan has its own unique coordinate system. The pixel direction can be expressed either as a unit vector or as a pair of coordinates in a plane perpendicular to the z-axis. In the latter case, the z-coordinate of a pixel is always constant. We will use the latter in our derivations.

Sensor models are derived for both the multispectral scanner (MSS) and the thematic mapper (TM). Essentially, from the point of view of sensor modelling, the MSS and the TM are the same, except for the fact that the TM uses a scan line corrector to compensate for the motion of the satellite during scanning. This is necessary because unlike the MSS which uses only the forward scan for imaging, the TM uses both the forward and the reverse scan. For both the MSS and the TM, every frame of imagery consists of a number of scans, every scan consists of a number of lines and every line consists of a number of samples or pixels.

The position of a point in an imagery is defined by its row (r) and column (c) numbers, which are not necessarily integers. The column number c has to be corrected for the deviation of the number of samples in one scan from the nominal, which is known as the line length correction, and for the non-linearity of scanning. The line length correction is applied by simply multiplying c by a constant factor resulting in

$$(2.11) \qquad c' = \left[\frac{N_s' - 1}{N_s - 1}\right] c$$

where N_s is the observed number of samples in one scan, N_s' is the nominal number of samples in one scan, and c' is the column number with line length correction applied. The formula assumes that the scanning is linear in time or equivalently, that the velocity of scanning is constant. To correct for the non-linearity in scanning, the deviation of c' from the nominal is modelled as a polynomial series resulting in

$$(2.12) \qquad \Delta c' = a_0 + a_1 c' + a_2 c'^2 + a_3 c'^3 + a_4 c'^4 + \ldots.$$

where c' is defined in equation (2.11), $\Delta c'$ is the deviation of c' from its correct value and a_0, a_1, a_2, a_3, a_4, ... are the coefficients of the polynomial series measured during sensor calibration. The final column number corresponding to a point is as follows:

$$(2.13) \qquad c'' = c' + \Delta c'$$

where c" is the column number with both the line-length and scanning non-linearity correction applied.

For the MSS, the row number r of a point needs no correction. For the TM, the row number is compensated for the effect of the scan line

corrector. The scan line corrector is an image motion compensation device which attempts to cancel the relative motion between the satellite and the ground during image acquisition in every scan. In the TM, if no image motion compensation is applied, the ground coverage of the forward and the reverse scan will not be parallel. The compensation for the row number in the forward scan has the following form:

$$(2.14a) \quad \Delta r_F = \frac{S_P}{2} - \left[\frac{S_P}{N'_S - 1}\right][c'' - 1]$$

For the reverse scan, the compensation is just the opposite for that of the forward scan, hence,

$$(2.14b) \quad \Delta r_R = -\frac{S_P}{2} + \left[\frac{S_P}{N'_S - 1}\right][c'' - 1],$$

where:

- Δr_F and Δr_R are the compensations for the row number in the forward and the reverse scans respectively;
- c'' is defined in equation (2.13);
- N'_S is the nominal number of samples in one scan;
- S_P is the distance travelled, in pixels, of the satellite ground track in one scan. For aid in visualizing the effect of the scan line corrector see Figures 6a, 6b, and 6c.

The corrected row number for both the TM forward and reverse scan is given by

$$r' = r + \Delta r$$

where r' is the corrected row number and Δr is either Δr_F or Δr_R as

defined in equation (2.14). The problem of determining whether a point was imaged during the forward or reverse scan, by the TM will be discussed presently.

We first assume that an image frame consists only of whole scans. Then the scan number to which a point belongs is

$$(2.15) \qquad i_s = \left\lfloor \frac{r - 0.5}{N_L} \right\rfloor + 1$$

where $\lfloor \quad \rfloor$ means the largest integer not exceeding the value inside, r is the uncorrected row number, and N_L is the number of lines in one scan. If the first scan is forward, then all odd scans are forward scans and all even scans are reverse scans and vice versa. The corrected line number of a point, once its scan number is known, is

$$(2.16a) \qquad \ell = r' - (i_s - 1)\, N_L$$

and its corrected sample number is equal to the corrected column number, that is

$$(2.16b) \qquad s = c''$$

where ℓ and s are the corrected line and sample numbers of a point, respectively; and r' and c" are the corrected row and column numbers, respectively.

The direction of a pixel vector with respect to sensor system can now be expressed in terms of ℓ and s. In Figure 7, α is proportional to s, that is,

$$\alpha = \frac{\psi}{N_s' - 1}\,(s - 1) - \frac{\psi}{2},$$

and β is proportional to ℓ, that is,

$$\beta = \frac{\theta}{N_L' - 1} (\ell - 1) - \frac{\theta}{2} ,$$

where ψ and θ are the total sensor angular coverages across and along satellite track, respectively. Also in Figure 7, p is the point on the plane of the imagery (this plane is really part of a cylinder); p' is its projection on a plane perpendicular to z; c is the principal distance of the sensor optical system; and x_p', y_p' are the coordinates of point p' on the plane perpendicular to z.

From Figure 7, the following relations are written

(2.17a) $y_p' = c \tan \alpha,$

(2.17b) $x_p' = \frac{c}{\cos \alpha} \tan \beta = c \sec \alpha \tan \beta,$

and

(2.17c) $z_p' = - c.$

These expressions for the coordinates of the pixel position projected on a plane is the objective of sensor modelling.

2.4 Platform Modelling

In platform modelling, first an expression for the position of the satellite in the ground coordinate system is derived. Then, a transformation is defined which makes the ground coordinate system parallel to the sensor coordinate system. Once these are done, the satellite collinearity equation (equation (2.3)) is then readily derived.

The position of the satellite in terms of the ground coordinate system can be defined in at least three ways. The first expresses the satellite position in terms of its position vector $\vec{R}$. This approach requires that the satellite orbit, needed for defining the sensor attitude, be expressed in terms of $\vec{R}$ and the velocity vector $\vec{V}$. The

weakness of this approach is that we must express six variables as unknown functions of time, three for the components of $\vec{R}$ and three for the components of $\vec{V}$, resulting in models with very weak geometry. The usual solution for this shortcoming is to assume that $\vec{R}$ and $\vec{V}$ are known a-priori.

The second approach assumes that the parameters defining the satellite orbit are themselves functions of time. In this case, we must also express six variables as unknown functions of time. As in the first approach, the resulting model geometry is also very weak. One common solution for this problem in this case is to assume some of the parameters as fixed or known a-priori.

The third approach assumes that the parameters defining the satellite orbit are independent of time. Once the orbit is defined using nominal parameters, the nominal position of the satellite in the orbit plane, specifically the instantaneous R and the true anomaly, ν, can be defined using equations (2.4) and (2.5), if the orbit is assumed to be elliptical. If the orbit is assumed to be circular, the satellite position can be defined using equation (2.6) where A_s is made equal to the radius of the circular orbit. This approach requires that the three components of the small deviation of the actual satellite position from the predicted position using nominal orbital parameters be modelled as functions of time. Compared to the previous two approaches which required that six parameters be expressed as functions of time, the last approach results in a much stronger geometry. Therefore, this last approach is used in the derivations of the selected model.

The three components of the deviation of the satellite from its nominal position, are defined as follows: ΔR is the component parallel

to and in the same direction as the position vector $\vec{R}$ of the satellite; ΔG is in the plane of the nominal orbit, perpendicular to ΔR and positive in the direction of satellite motion; and ΔP is perpendicular to the orbital plane. The set ΔG, ΔP, ΔR forms a right handed coordinate system. Since these components are small, they can be modelled quite well by the following polynomial series:

$$\text{(2.18a)} \quad \Delta G = G_0 + G_1 (t - t_F) + G_2 (t - t_F)^2 + \ldots.$$

$$\text{(2.18b)} \quad \Delta P = P_0 + P_1 (t - t_F) + P_2 (t - t_F)^2 + \ldots.$$

$$\text{(2.18c)} \quad \Delta R = R_0 + R_1 (t - t_F) + R_2 (t - t_F)^2 + \ldots.$$

where $G_0, G_1, G_2, \ldots., P_0, P_1, P_2, \ldots., R_0, R_1, R_2, \ldots.$ are coefficients of the corresponding polynomial terms; t is time, t_F is the time at the center of the frame; and t is zero at the ascending node.

The ground coordinate system used is the geocentric system where the origin is the center of the earth, the X-axis passes through Greenwich meridian at the equator, the Z-axis is parallel to the rotational axis of the earth and the Y-axis completes the right handed coordinate system. This coordinate system rotates with the earth. We define our inertial coordinate system to coincide with the ground coordinate system when the satellite is at the ascending node, that is, when the satellite crosses the plane of the earth's equator while travelling from south to north. The only difference between the ground coordinate and the inertial coordinate systems is that while the former rotates with the earth, the latter maintains a constant angle with the projection of the earth-sun line on the earth's equatorial plane. This convention regarding the inertial coordinate system results in a plane orbit in this

coordinate system for sun-synchronous satellites such as Landsat. In Figure 8, X Y Z is the ground coordinate system, and X^i Y^i Z^i is the inertial coordinate system. In the same figure:

A	is the ascending node;
P	is the perigee (the point in the satellite orbit nearest the earth);
S	is the satellite;
Ω	is the longitude of A with respect to the inertial coordinate system;
i	is the inclination of the satellite orbit;
ω	is the argument of the perigee;
ν	is the true anomaly;
R	is the radial distance of the satellite from center;
ω_e	is the angular velocity of the earth;
t	is the time (t = 0 when the satellite is at the ascending node); and
ΔG, ΔP, ΔR	are the deviations of the satellite from its nominal position.

To define the satellite position in the ground coordinate system we have to perform a series of rotations on the ground coordinate system. The first such rotation is around the Z axis which brings the ground coordinate system into the inertial coordinate system resulting in

$$(2.19) \quad \begin{bmatrix} X^i \\ Y^i \\ Z^i \end{bmatrix} = \begin{bmatrix} \cos(-\omega_e t) & \sin(-\omega_e t) & 0 \\ -\sin(-\omega_e t) & \cos(-\omega_e t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

The second rotation is around the Z^i-axis to make the X^i-axis coincide with the line of apsides (passes through A) which results in

$$(2.20)\quad \begin{bmatrix} X^1 \\ Y^1 \\ Z^1 \end{bmatrix} = \begin{bmatrix} \cos(\Omega) & \sin(\Omega) & 0 \\ -\sin(\Omega) & \cos(\Omega) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X^i \\ Y^i \\ Z^i \end{bmatrix}$$

Substituting equation (2.19) into equation (2.20), we get

$$(2.21)\quad \begin{bmatrix} X^1 \\ Y^1 \\ Z^1 \end{bmatrix} = M_1 \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos(\Omega - \omega_e t) & \sin(\Omega - \omega_e t) & 0 \\ -\sin(\Omega - \omega_e t) & \cos(\Omega - \omega_e t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

The third rotation is around the X^1-axis by the angle $(\pi/2 + i)$, see Figure 9, or

$$(2.22)\quad \begin{bmatrix} X^2 \\ Y^2 \\ Z^2 \end{bmatrix} = M_2 \begin{bmatrix} X^1 \\ Y^1 \\ Z^1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\pi/2 + i) & \sin(\pi/2 + i) \\ 0 & -\sin(\pi/2 + i) & \cos(\pi/2 + i) \end{bmatrix} \times \begin{bmatrix} X^1 \\ Y^1 \\ Z^1 \end{bmatrix}$$

The X^2- and the Z^2-axes lie on the orbit plane while the Y^2-axis is perpendicular to it. The next rotation is around the Y^2-axis such that

the Z^2-axis passes through the satellite position that is corrected for the radial (ΔR) and orbital (ΔG) deviations. The resulting equations are

$$(2.23a)\quad \begin{bmatrix} X^3 \\ Y^3 \\ Z^3 \end{bmatrix} = M_3 \begin{bmatrix} X^2 \\ Y^2 \\ Z^2 \end{bmatrix} = \begin{bmatrix} \cos(\pi/2 + \omega + \nu + \theta_G) & 0 & -\sin(\pi/2 + \omega + \nu + \theta_G) \\ 0 & 1 & 0 \\ \sin(\pi/2 + \omega + \nu + \theta_G) & 0 & \cos(\pi/2 + \omega + \nu + \theta_G) \end{bmatrix} \begin{bmatrix} X^2 \\ Y^2 \\ Z^2 \end{bmatrix}$$

This can be seen more clearly in Figure 10 which shows the orbital plane only; ω and ν were defined previously; R_G is the magnitude of the vector sum of $\vec{R}$, $\Delta\vec{R}$, and $\Delta\vec{G}$. The angle θ_G which corrects for the deviation of the satellite along the radial (ΔR) and orbital (ΔG) direction is defined as follows:

$$(2.23b)\quad \theta_G = \tan^{-1}\left(\frac{\Delta G}{R + \Delta R}\right)$$

The last rotation needed to define the satellite position in the ground coordinate system corrects for the deviation of the satellite position perpendicular to the satellite orbit (ΔP). In Figure 11, S' is the actual satellite position, then

$$(2.24a)\quad R_G = \sqrt{(R + \Delta R)^2 + \Delta G^2}$$

$$(2.24b)\quad R' = \sqrt{(R + \Delta R)^2 + \Delta G^2 + \Delta P^2}$$

and

$$(2.24c)\quad \theta_P = \tan^{-1}\left(\frac{\Delta P}{R_G}\right)$$

Rotating around the X^3-axis by $-\theta_p$ brings the $X^3\ Y^3\ Z^3$ coordinate system into the $X^S\ Y^S\ Z^S$ coordinate system. The set of equations resulting from this rotation is

$$(2.24d)\quad \begin{bmatrix} X^S \\ Y^S \\ Z^S \end{bmatrix} = M_4 \begin{bmatrix} X^3 \\ Y^3 \\ Z^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-\theta_p) & \sin(-\theta_p) \\ 0 & -\sin(-\theta_p) & \cos(-\theta_p) \end{bmatrix} \begin{bmatrix} X^3 \\ Y^3 \\ Z^3 \end{bmatrix}$$

The $X^S\ Y^S\ Z^S$ is the satellite coordinate system. The origin of the system is still the center of the earth, the Z^S-axis passes through the actual satellite position, the X^S-axis is parallel to the nominal satellite orbit and positive in the direction of satellite motion and the Y^S-axis, which is not necessarily perpendicular to the nominal satellite orbit, completes the right handed system.

Collecting equations (2.21), (2.22), (2.23a), and (2.24d) together we get,

$$(2.25)\quad \begin{bmatrix} X^S \\ Y^S \\ Z^S \end{bmatrix} = M_4\ M_3\ M_2\ M_1 \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = M_s \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Since M_1, M_2, M_3, and M_4 are all orthogonal matrices, M_s is also orthogonal. It can be seen in Figure 11 that the vector which defines the actual satellite position in the $X^S\ Y^S\ Z^S$ coordinate system is $[0\ \ 0\ \ R']^t$ where R' is defined in equation (2.24b). Therefore the position of the satellite in the ground coordinate system is

$$(2.26)\quad \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} = M_s^T \begin{bmatrix} 0 \\ 0 \\ R' \end{bmatrix}$$

Once the satellite position in terms of the ground coordinate system is defined, the next step in platform modelling is to define the transformation M, which makes the ground coordinate system parallel to the sensor coordinate system. Since the transformation M_s, which brings the ground coordinate system into the satellite coordinate system is already defined (see equation 2.25), we only have to derive the transformation which brings the satellite coordinate system into the sensor coordinate system. This latter transformation consists of a series of rotations which correct for the fact that the vertical does not pass through the center of the earth and which properly account for the attitude of the scanner coordinate system.

In Figure 12, the relative orientation between the satellite coordinate system X^S Y^S Z^S and the ground coordinate system X Y Z is shown. In the same figure,

S'	is the ground track of the satellite S;
R'	is the distance of the satellite from the center of the earth;
θ_s	is the latitude of the satellite;
N_s	is the radius of the prime vertical;
δN_s	is that part of the prime vertical below the equator for points in the northern hemisphere and above the equator for points in the southern hemisphere;
$\delta Z_s = \delta N_s \times \sin \theta_s$	is the projection of δN_s on the Z-axis; and
h_s	is the elevation of the satellite.

The prime vertical N_s and the elevation h_s form a straight line which represents the vertical that passes through the satellite. It can be seen that the vertical does not pass through the center of the earth.

It is necessary to compensate for the non-coincidence of the vertical with center of the earth because the vertical is the nominal direction of the z-axis of the sensor coordinate system as previously defined. This compensation can be done by making the Z^S-axis parallel to the vertical or equivalently by making the Z^S-axis pass through a point whose position is defined by the sum of the vectors $\vec{R}'$ and $[0 \quad 0 \quad \delta Z_s]^t$. The vector $[0 \quad 0 \quad \delta Z_s]^t$ is a function of the satellite latitude θ_s which in turn is related to the satellite coordinates X_s, Y_s, Z_s via equation (2.9). This can be seen more clearly in Figure 13 which is a simplified version of Figure 12.

To define the angular rotations necessary for making the Z^S-axis parallel to the vertical, we first have to transform the vector $[0 \quad 0 \quad \delta Z_s]^t$ into the satellite coordinate system X^S Y^S Z^S. The result of the transformation using equation (2.25) is

$$(2.27) \qquad \begin{bmatrix} \delta X_s^S \\ \delta Y_s^S \\ \delta Z_s^S \end{bmatrix} = M_s \begin{bmatrix} 0 \\ 0 \\ \delta Z_s \end{bmatrix}$$

The elements in equation (2.27) are also shown in Figure 13.

The first rotation to make the Z^S-axis parallel to the vertical is around the Y^S-axis by the angle θ_x (see Figure 14) which results in

$$(2.28a)\quad \begin{bmatrix} X^5 \\ Y^5 \\ Z^5 \end{bmatrix} = M_5 \begin{bmatrix} X^S \\ Y^S \\ Z^S \end{bmatrix} = \begin{bmatrix} \cos\theta_x & 0 & -\sin\theta_x \\ 0 & 1 & 0 \\ \sin\theta_x & 0 & \cos\theta_x \end{bmatrix} \begin{bmatrix} X^S \\ Y^S \\ Z^S \end{bmatrix}$$

where

$$(2.28b)\quad \theta_x = \tan^{-1}\left[\frac{\delta X_s^S}{R' + \delta Z_s^S}\right]$$

R' is the radius of the satellite defined in equation (2.24b), and δX_s^S, δZ_s^S are defined in equation (2.27).

The second rotation is around the X^5-axis by the angle θ_y (see Figure 15) such that

$$(2.29a)\quad \begin{bmatrix} X^6 \\ Y^6 \\ Z^6 \end{bmatrix} = M_6 \begin{bmatrix} X^5 \\ Y^5 \\ Z^5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-\theta_y) & \sin(-\theta_y) \\ 0 & -\sin(-\theta_y) & \cos(-\theta_y) \end{bmatrix} \begin{bmatrix} X^5 \\ Y^5 \\ Z^5 \end{bmatrix}$$

where

$$(2.29b)\quad R'' = \sqrt{(R' + \delta Z_s^S)^2 + (\delta X_s^S)^2}\ ,$$

$$(2.29c)\quad \theta_y = \tan^{-1}\left(\frac{\delta Y_s^S}{R''}\right)\ ,$$

R', δX_s^S, δZ_s^S are the same as in equation (2.28) and δY_s^S is defined in equation (2.27).

After making the Z^S-axis of the satellite coordinate system X^S Y^S Z^S parallel to the vertical, we then have to account for the attitude of the sensor coordinate system during pixel imaging. This is done through a series of sequential rotations to correct for the roll ω, the pitch ϕ, and the yaw κ, applied in that order. The first rotation is that due to the roll ω, resulting in

$$(2.30)\quad \begin{bmatrix} X^{\omega} \\ Y^{\omega} \\ Z^{\omega} \end{bmatrix} = M_{\omega} \begin{bmatrix} X^{6} \\ Y^{6} \\ Z^{6} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & \sin\omega \\ 0 & -\sin\omega & \cos\omega \end{bmatrix} \begin{bmatrix} X^{6} \\ Y^{6} \\ Z^{6} \end{bmatrix}$$

The next rotation is to compensate for the pitch ϕ, such that

$$(2.31)\quad \begin{bmatrix} X^{\phi} \\ Y^{\phi} \\ Z^{\phi} \end{bmatrix} = M_{\phi} \begin{bmatrix} X^{\omega} \\ Y^{\omega} \\ Z^{\omega} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} X^{\omega} \\ Y^{\omega} \\ Z^{\omega} \end{bmatrix}$$

The last rotation which accounts for the yaw κ, produces the following set of equations:

$$(2.32)\quad \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = M_{\kappa} \begin{bmatrix} X^{\phi} \\ Y^{\phi} \\ Z^{\phi} \end{bmatrix} = \begin{bmatrix} \cos\kappa & \sin\kappa & 0 \\ -\sin\kappa & \cos\kappa & 0 \\ & 0 & 1 \end{bmatrix} \begin{bmatrix} X^{\phi} \\ Y^{\phi} \\ Z^{\phi} \end{bmatrix}$$

Since each pixel has its own unique attitude, we have to parameterize its components ω, ϕ, κ in terms of time in a similar manner to what was previously done to the components of the deviation of the satellite position. We also selected in this case polynomials, resulting in:

$$(2.33a)\quad \omega = \omega_0 + \omega_1 (t - t_F) + \omega_2 (t - t_F)^2 + \omega_3 (t - t_F)^3 + \ldots.$$

$$(2.33b)\quad \phi = \phi_0 + \phi_1 (t - t_F) + \phi_2 (t - t_F)^2 + \phi_3 (t - t_F)^3 + \ldots.$$

$$(2.33c)\quad \kappa = \kappa_0 + \kappa_1 (t - t_F) + \kappa_2 (t - t_F)^2 + \kappa_3 (t - t_F)^3 + \ldots.$$

where t is time which is zero at the satellite ascending node and t_F is the time of imaging of the frame center.

Combining equations (2.28a), (2.29a), (2.30), (2.31), and (2.32) results in

$$(2.34)\quad \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = M_\kappa\, M_\phi\, M_\omega\, M_6\, M_5 \begin{bmatrix} X^s \\ Y^s \\ Z^s \end{bmatrix} = M_a \begin{bmatrix} X^s \\ Y^s \\ Z^s \end{bmatrix}$$

All the matrices involved in equation (2.34) are orthogonal. Substituting equation (2.25) into equation (2.34) gives

$$(2.35)\quad \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = M_a\, M_s \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = M \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

The coordinate system X' Y' Z' with origin at the center of the earth is parallel to the sensor coordinate system x y z. The derivation of M and the previous derivation of the satellite position vector $[X_s\ Y_s\ Z_s]^t$ completes platform modelling.

2.5 Combined Sensor/Platform Model and Applications

The sensor and the platform models were derived independently of each other. A convenient method of relating them is to express at least some quantities involved in the platform model as functions of position of points in the imagery. Since pixel imaging is done sequentially with respect to time, it follows that pixel positions are also functions of time. We may then reverse the relationship and express time as a function of pixel positions. Furthermore, since some of the parameters in the platform model are functions of time, these parameters are also functions of pixel position. Thus, we are able to relate the platform model to the sensor model.

A convenient expression for time in terms of the pixel position for the MSS and for the odd scans of the TM is,

$$(2.36a) \quad t = t_F + 2\,(i_s - 1)\,\frac{\Delta t_c}{2} + (c - 1)\,\frac{\Delta t_s}{N_p' - 1} - \frac{N_s}{2}\,\Delta t_c$$

For the even scans of the TM, the corresponding expression is,

$$(2.36b) \quad t = t_F + [2\,(i_s - 1) + 1]\,\frac{\Delta t_c}{2} + (c - 1)\,\frac{\Delta t_s}{N_p' - 1} - \frac{N_s}{2}\,\Delta t_c$$

Terms in both equations are defined as follows:

t	is the elapsed time which is zero at the satellite ascending node;
t_F	is the time of imaging of the pixel center (approximate);
i_s	is the scan line number to which a pixel belongs;
c	is the uncorrected pixel column number;
N_p'	is the nominal number of pixels in one scan;
N_p	is the actual number of pixels in one scan;
N_s	is the number of scans in one frame;
Δt_c	is the sensor cycling time; and
Δt_s	is the one active scanning interval of the sensor.

If the odd scan for TM is the reverse scan ($N_p - c + 1$) should be substituted for c in equation (2.36a) and if the even scan is the reverse scan ($N_p - c + 1$) should be substituted for c in equation (2.36b). The main assumption in equation (2.36) is that all pixels in one column for a given scan are sampled simultaneously.

The combined sensor and platform model is expressed by the satellite collinearity equation given in Section 2.1, or

$$(2.1) \qquad \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \lambda M \begin{bmatrix} X_G - X_s \\ Y_G - Y_s \\ Z_G - Z_s \end{bmatrix}$$

sensor modelling defined the vector $[x_p \; y_p \; z_p]^t$ in terms of the pixel image row and column numbers. Platform modelling defined the satellite position vector $[X_s \; Y_s \; Z_s]^t$ and the orthogonal matrix M in terms of the orbit parameters, satellite position deviation parameters, attitude parameters, and time. Then equation (2.36) related the sensor and platform models by defining time in terms of image pixel position.

The satellite collinearity equation can be used for producing simulated data useful for studying rectification. For this application equation (2.1) is inverted to the form

$$(2.37) \qquad \begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix} = 1/\lambda \; M^t \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} + \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix}$$

Using equation (2.37), the ground position X_G, Y_G, Z_G of a pixel can be solved for given the following: the pixel row and column number in the image; the satellite orbit parameters Ω, i, ω, A_s, and ε_s; the parameters defining the satellite position deviation components ΔG, ΔP, ΔR; the parameters defining the sensor attitude components ω, ϕ, κ; the parameters defining time (t_F, Δt_c, Δt_s); the sensor constants N_p, N_ℓ, N_s, c and the scanning non-linearity correction constants; the earth related constants A_e, ε_e, ω_e, G, M_e; and the elevation h of the point. This procedure will, in effect, give us pixels whose ground positions are perfectly known.

For rectification, the original form of the satellite collinearity equation (equation 2.1) is used. The vector $[x_p\ y_p\ z_p]^t$ is first computed using the sensor calibration constants and the pixel image row and column numbers; this vector is considered as the observation in the subsequent adjustment procedure applied. Then the right hand side of equation (2.1) is linearized in terms of the parameters defining satellite deviation components, the parameters defining time, and the ground coordinates. The ground coordinates are considered either as constants or as observations. The orbit parameters are estimated using a-priori information and assumed constant because effects of errors in their a-priori estimates are compensated for by the parameters defining the satellite position deviation. Using control points with known image and ground position, the unknown parameters are solved for in an adjustment procedure. Any a-priori information regarding the unknown parameters can be incorporated into the adjustment using the proper variance-covariance matrices.

3. ACCURACY STUDIES USING SYNTHETIC DATA

3.1 Effect of Parameter Perturbations

Essentially, all rectification methods require that we have knowledge of the values of the parameters of the model being utilized. These parameters can be estimated using ground control points or they can be independently observed or both. Once these parameters are known, the ground position of pixels can be readily computed. Rectification accuracy, therefore, is directly affected by the accuracy of the parameter values.

One application of equation (2.37), which is the form of the satellite collinearity equation suited for simulation, is for computing the effect on pixel ground position of perturbations on the nominal values of the parameters. The effect on pixel ground positions of perturbation applied to a single parameter can be seen in Table I. In this table, the tabulated values are the individual perturbations; the resulting root mean square displacements in pixel ground position resulting from each individual perturbation is shown in the heading. It can be seen that within the range of values of interest, the resulting displacement varies linearly with the applied perturbations for all the parameters listed.

Also listed in Table I are the present accuracies of some independently observed parameters for the MSS and the TM together with the ground displacements (in brackets) produced by their standard deviations. It can be seen that for the MSS, inaccuracies in the observed values of roll (ω) and pitch (ϕ) produced the largest ground displacement followed by errors in the satellite position deviation parameter along the orbital plane (G) and in the sensor cycling time (Δt_c).

Table II shows the ground displacements when all the parameters are perturbed simultaneously. A set of perturbations corresponds to a column in Table I and is represented in the left column of Table II by the ground displacement produced by the individual parameters. Note that each perturbation in the set produces identical ground displacements when applied individually. The resulting ground displacements due to the combined perturbations are tabulated in the right column.

3.2 Comparison of Different Mathematical Models

One factor which affects the accuracy of rectification is the type of model used. By its very nature, the geometry of the satellite imagery is very weak. Because of this, even the best models presently existing do not allow for the recovery of parameters defining the satellite position deviation components and the attitude elements at the same time. The model we proposed in Section 2 is capable of recovering all of these parameters at the same time with one exception; instead of the satellite position deviation component along the orbit, we recover the time of imaging of the frame center. Both of these parameters cause the frame to be displaced along the orbit and for small deviations, one can satisfactorily take the place of the other.

We used five models in our test. They are: (1) the full model in Section 2 which assumes that the earth is an ellipsoid of revolution and that the orbit of the satellite is an ellipse; (2) the same model in number (1) except that the orbit of the satellite is assumed a circle instead of an ellipse; (3) the same model in number (2) with the additional assumption that the earth is a sphere; (4) the model used for aircraft scanner data which assumes that the orbit is a straight line and requires that the earth be projected on a mapping plane; and (5) the polynomial interpolative model. Two cases are run for each model.

The results for two cases are shown in Table III. Case I assumes that there is no error in identifying the control points on both the image and on the ground, and that there is no error in the derived or measured point position in both the image and the ground. There are 156 control and 156 check points that are both well distributed. Case R

assumes: that there is no error in identifying control points on the ground; that the standard deviation of the measured ground position of control points in each of the axes is 15 m resulting in 26 m standard deviation when combined (21 m in plan); that the error in identifying points in the image is uniformly distributed from -0.5 to 0.5 pixel with the resulting standard deviation of 0.28 pixel in both across and along scan direction, and that the errors in the derived position of points in the image due to sensor instabilities not including identification errors are .01 and .5 pixel in the across and along scan directions respectively. The total error in position of points across and along scan are .29 pixel (23 m) and .58 pixel (34 m) respectively; the combined error is 41 m.

Since the data for Case I are perfect, the resulting standard deviation in both the control and check points can be considered as systematic errors caused by inadequate model. Table III shows that only the last two models are inadequate in describing the geometry of the imagery. Case R, however, shows that if the errors in both the image and ground position of points are not appreciably smaller than the systematic error introduced by the model, there is really no advantage in using more sophisticated ones.

3.3 Effect of Different Control Densities

Another factor which affects the accuracy of rectification is the number or density of control. This experiment simply involves the varying of the number of control points in the two cases (I and R) studied. The model used in both cases is Model (1) in Section 3.2. The assumptions regarding the accuracy of derived or measured position of points on both

the image and the ground in Section 3.2 for Case I and Case R apply in this section as well.

The results are shown in Table IV. For Case I where the position of points in both the image and the ground are perfect, whenever the number of equations (2 per control point) exceeds the number of unknown parameters (19 in this case) rectification is almost perfect. Case R shows that any increase in the density of control points after a certain number is reached (approximately 25 points in this case) results only in a marginal increase in rectification accuracy.

3.4 Effect of Different Control Point Ground Position Accuracy

The next factor that significantly affects rectification accuracy is the accuracy of the measured ground position of control points. We assume that there is no identification error of control points on the ground; only measurement errors of ground position. Again, two cases are involved, Cases I and R. Both cases use Model (1) in Section 3.2 for rectification. Case I has 156 control points while Case R has only 25. Again, the assumptions for Cases I and R in Section 3.2 regarding the position of points in the image apply in this case.

Table V shows the effect of varying the accuracy of control points ground position for both cases. In Case I where image position is perfect, roughly 80% of the error in the ground position of control points is compensated for by the rectification process. In Case R decreasing the standard deviation of control point ground position below that of the corresponding standard deviation in the image will not increase rectification accuracy.

3.5 Effect of Derived Image Position Accuracy

During the imaging process, the direction of the ray which produced the image of a given point is defined in the sensor coordinate system. The accuracy with which we can reconstruct this direction in the sensor coordinate system depends on the accuracy of the identification of point in the image and the geometric stability of the sensor.

Table VI shows the effect of image position errors on rectification accuracy. Again, two cases are presented. Both cases use Model (1) in Section 3.2 as the rectification model. Case I has 156 points and Case R has 25 points. The assumptions regarding the accuracy of ground position of control points in Section 3.2 apply here as well.

It can be seen from Table VI that only a very small percentage of errors in the image position is compensated for by the rectification process. This is true for both Cases I and R.

4. CONCLUSIONS AND RECOMMENDATIONS

1. It is possible to recover all parameters defining satellite position deviation and sensor attitude using appropriate models.
2. Uncertainties in the roll (ω) and the pitch (ϕ) of the sensor contribute the greatest errors in system corrected images followed by uncertainties in the satellite position along the orbit and the sensor cycling time.
3. Polynomial models and those that assume that the orbit is a straight line and that require the projection of the earth's surface on a mapping plane cannot produce rectification accuracies better than half a pixel.

4. Marginal increase in rectification accuracy results by increasing the number of control points above 25.
5. A large percentage of errors in ground position of control points is compensated for by the rectification process.
6. A very small percentage of error in image position is compensated for by the rectification process.
7. Sub-pixel rectification is possible only if points on the image can be identified to sub-pixel accuracies.
8. Improving the identification accuracy of points on the image is worth further investigation since rectification accuracy is highly sensitive to this error.
9. With the sensor/platform model now available, several other registration/rectification problems can be researched. These include: (1) investigation of image correspondence; (2) study of different control types, such as points, areas, relative control, and use of geometric constraints; and (3) analysis of the optimum registration/rectification sequence.
10. Other fundamental research areas within the general problem of registration/rectification of remote sensing data include: (a) accuracy measures; (b) reduction (photogrammetric) of multiple spatial coverage with the same and different sensors; and (c) efficient means of rectification of sensor data to digital terrain models.

5. REFERENCES

[1] Alspaugh, D. W., Lecture Notes in Orbit Mechanics, Purdue University, West Lafayette, Indiana, 1980.

[2] Bähr, H. P., "Interpolation and Filtering of ERTS Imagery", ISP Commission III Symposium, Stuttgart, September, 1974.

[3] Bähr, H. P., "Geometrical Analysis and Rectification of Landsat MSS Imagery: Comparison of Different Methods", Nachr. Kart m. Vermess Wes., No. 36, Institut fur Angewandte Geodasie, Frankfurt, 1978, pp. 25 - 46.

[4] Bähr, H. P., "Geometrical Models for Satellite Scanner Imagery", ISP Commission III Congress, Helsinki, July, 1976.

[5] Bernstein, R., "Digital Image Processing of Earth Observation Sensor Data", IBM J. Research Development, January, 1976, pp. 40 - 57.

[6] Caron, R. H. and Simon, K. W., "Attitude Time-Series Estimator for Rectification of Spaceborne Imagery", J. Spacecraft, Volume 12, Number 1, January, 1975, pp. 27 - 32.

[7] Dowman, I. J. and Mohamed, M. A., "Photogrammetric Applications of Landsat MSS Imagery", International J. Remote Sensing, Volume 2, Number 2, 1981, pp. 105 - 113.

[8] Ewing, C. E. and Mitchell, M. M., Introduction to Geodesy, New York: American Elsevier, 1969.

[9] Forrest, R. B., "Simulation of Orbital Image-Sensor Geometry", Photogrammetric Engineering and Remote Sensing, Volume 47, Number 8, August, 1981, pp. 1187 - 1193.

[10] Forrest, R. B., "Geometric Correction of ERTS-1 MSS Images", ISP Commission III Symposium, Stuttgart, September, 1974.

[11] Konecny, G., "Mathematical Models and Procedures for the Geometric Restitution of Remote Sensing Imagery", ISP Commission III Congress, Helsinki, July, 1976.

[12] Kratky, V., "Photogrammetric Solution for Precision Processing of ERTS Images", ISP Commission II Congress, Ottawa, July, 1972.

[13] Levine, I., "Computational Aspects of Geometric Correction Data Generation in the Landsat-D Imagery Processing", Sixth Annual Flight Mechanics/Estimation Theory Symposium, NASA/GSFC, October, 1981.

[14] Levine, I., "The MSS Control Point Location Error Filter for Landsat-D", Sixth Annual Flight Mechanics/Estimation Theory Symposium, NASA/GSFC, October, 1981.

[15] Mikhail, E. M. and McGlone, J. C., "Current Status of Metric Reduction of (Passive) Scanner Data", ISP Commission III Congress, Hamburg, July, 1980, pp. 504 - 514.

[16] Puccinelli, E. F., "Ground Location of Satellite Scanner Data", Photogrammetric Engineering and Remote Sensing, Volume 42, Number 4, April 1976, pp. 537 - 543.

[17] Rifman, S. S., Monuki, A. T., and Shortwell, C. P., "Multi-Sensor Landsat MSS Registration", 13th International Symposium on Remote Sensing of Environment, Ann Arbor, Michigan, April, 1979.

[18] Sawada, N. et. al., "An Analytic Correction Method for Satellite MSS Geometric Distortions", Photogrammetric Engineering and Remote Sensing, Volume 47, Number 8, August, 1981, pp. 1195 - 1203.

[19] Synder, J. P., "Geometry of a Mapping Satellite", Photogrammetric Engineering and Remote Sensing, Volume 48, Number 10, October, 1982, pp. 1593 - 1602.

[20] Trinder, J. C. and Nasca, S. U., "Tests on the Mapping Application of Landsat Imagery", ISP Commission III Congress, Helsinki, July, 1976.

[21] Richardus, P. and Adler, R. K., Map Projections for Geodesists, Cartographers and Geographers, Amsterdam: North-Holland Publishing Company, 1972.

TABLE 1 EFFECT OF PERTURBATION ON A SINGLE PARAMETER ON GROUND POSITION

	AMOUNT OF INDIVIDUAL PARAMETER PERTURBATIONS				PRESENT* ACCURACY (1σ)	
PAR \ RMS (m)	0.80	8.00	80.0	800.0	MSS	TM
TIME PARAMETERS						
T_f (m sec)	.120	1.20	12.0	120.	120.0 (80.0 m)	
ΔT_c (m sec)	.001	.010	.100	1.00		.400 (320. m)
ΔT_s (m sec)	.205	2.05	205.	205.		.003 (neg.)
ORBIT PARAMETERS						
Ω (deg x 10^{-3})	.00716	.0716	.716	7.16		
I (deg x 10^{-3})	.562	5.62	56.2	562.	45.0 (64.0 m)	45.0 (64.0 m)
W (deg x 10^{-3})	3.04	30.4	304.	3040.		
A_s (m)	.195	1.95	19.5	195.		
ϵ_s (x 10^{-6})	1.65	16.5	165.	1650.		

* PRESENT RMS MEASUREMENT ACCURACY OF EACH PARAMETER AS REPORTED IN LITERATURE

(continued next page)

TABLE I EFFECT OF PERTURBATION ON A SINGLE PARAMETER ON GROUND POSITION

(continued)

+ FOR SATELLITE POSITION PERTURBATION PARAMETERS

	AMOUNT OF INDIVIDUAL PARAMETER PERTURBATIONS				PRESENT ACCURACY (1σ)
PAR \ RMS (m)	0.80	8.00	80.0	800.	
G_0 (m)	.900	9.00	90.0	900.	500. (444. m)
G_1 (m/sec)	.100	1.00	10.0	100.	
G_2 (m/sec^2)	.0085	.085	.850	8.50	
P_0 (m)	.900	9.00	90.0	900.	100. (89.0 m)
P_1 (m/sec)	.100	1.00	10.0	100.	
P_2 (m/sec^2)	.0085	.085	.850	8.50	
R_0 (m)	12.5	125.	1250.	12500.	35. (2.24 m)
R_1 (m/sec)	1.40	14.0	140.	1400.	
R_2 (m/sec^2)	.115	1.15	11.5	115.	

(continued next page)

TABLE I EFFECT OF PERTURBATION ON A SINGLE PARAMETER ON GROUND POSITION

(continued)

+ FOR SENSOR ATTITUDE PARAMETERS

	AMOUNT OF INDIVIDUAL PARAMETER PERTURBATION				PRESENT ACCURACY (1σ)	
PAR \ RMS (m)	0.80	8.00	80.0	800.	MSS	TM
ω_0 (deg x 10^{-3})	.0504	.504	5.04	50.4	100. (1590. m)	10.0 (159. m)
ω_1 (deg/sec x 10^{-3})	.00555	.055	.555	5.55	10.0 (1440. m)	.001 (.144 m)
ω_2 (deg/sec^2 x 10^{-6})	.458	4.58	45.8	458.		
ω_3 (deg/sec^3 x 10^{-6})	.0355	.355	3.55	35.5		
ϕ_0 (deg x 10^{-3})	.0504	.504	5.04	50.4	100. (1590. m)	10.0 (159. m)
ϕ_1 (deg/sec x 10^{-3})	.00561	.0561	.561	5.61	10.0 (1430. m)	.001 (.143 m)
ϕ_2 (deg/sec^2 x 10^{-6})	.458	4.58	45.8	458.		
ϕ_3 (deg/sec^3 x 10^{-6}	.0355	.355	3.55	35.5		
κ_0 (deg x 10^{-3})	.802	8.02	80.2	802.	100. (100. m)	10.0 (10.0 m)
κ_1 (deg/sec x 10^{-3})	.0859	.859	8.59	85.9	10.0 (93. m)	.001 (.009 m)
κ_2 (deg/sec^2 x 10^{-6})	7.16	71.6	716.	7160.		
κ_3 (deg/sec^3 x 10^{-6})	.561	5.61	56.1	561.		

TABLE II EFFECT OF COMBINED PERTURBATIONS IN ALL PARAMETERS ON GROUND POSITION

RMS POSITION CHANGE DUE TO INDIVIDUAL PERTURBATION (m)	RMS POSITION CHANGE DUE TO COMBINED PERTURBATIONS (m)
0.80	5.28
8.00	52.9
80.0	536.
800.	6,370.

TABLE III COMPARISON OF DIFFERENT MATHEMATICAL MODELS

MODEL/CASES	CASES			
	I (RMS M)		R (RMS M)	
	CONTROL POINT	CHECK POINT	CONTROL POINT	CHECK POINT
(1)	<1	<1	36	50
(2)	<1	<1	38	48
(3)	2	2	38	48
(4)	36	31	45	43
(5)	38	38	60	57

Case I: 156 Control Points
156 Check Points
$\sigma_{control} = 0$
$\sigma_{pixel} = 0$

Case R: 25 Control Points
156 Check Points
$\sigma_{control}$: $\sigma_x = \sigma_y = \sigma_z = 15$ m
$\sigma_{plan} = 21$ m
$\sigma_{total} = 26$ m
σ_{pixel}: $\sigma_x = .29$ pixel (23 m)
$\sigma_y = .58$ pixel (34 m)
$\sigma_{total} = 41$ m

TABLE IV EFFECT OF DIFFERENT CONTROL DENSITIES

Number of Control Points/Cases	CASE I (RMS M)		CASE R (RMS M)	
	CONTROL POINT	CHECK POINT	CONTROL POINT	CHECK POINT
156	<1	<1	40	40
25	<1	<1	36	50
16	<1	<1	26	72
12	<1	<1	24	64
9	<1	74	24	81
4	461	302	26	283

Case I: Model: Ellipsoidal Earth, Elliptical Orbit

$\sigma_{control}$: 0

σ_{pixel}: 0

156 Check Points

Case R: Model: Ellipsoidal Earth, Elliptical Orbit

$\sigma_{control}$: $\sigma_x = \sigma_y = \sigma_z = 15$ m

$\sigma_{total} = 26$ m

σ_{pixel}: $\sigma_x = .29$ pixel (23 m)

$\sigma_y = .58$ pixel (34 m)

$\sigma_{total} = 41$ m

156 Check Points

TABLE V EFFECT OF DIFFERENT CONTROL POINT GROUND POSITION ACCURACY

CONTROL ACCURACY		CASES			
		I (RMS M)		R (RMS M)	
$\sigma_x = \sigma_y = \sigma_z$ (m)	σ_{total} (m)	CONTROL POINT	CHECK POINT	CONTROL POINT	CHECK POINT
0	0	<1	<1	34	47
5	9	2	1	35	48
15	26	5	4	36	50
25	43	8	7	41	52
50	87	15	13	58	62
75	130	22	19	83	80
100	173	30	26	107	98
150	260	45	39	154	135
200	346	60	51	199	172

Case I: Model: Ellipsoidal Earth, Elliptical Orbit
156 Control Points
156 Check Points
σ_{pixel}: 0

Case R: Model: Ellipsoidal Earth, Elliptical Orbit
25 Control Points
156 Check Points
σ_{pixel}: σ_x = .29 pixel (23 m)
σ_y = .58 pixel (34 m)
σ_{total} = 41 m

TABLE VI EFFECT OF DERIVED IMAGE POSITION ACCURACY

Image Position Accuracy (PIXEL)		CASES			
		I (RMS M)		R (RMS M)	
σ_{ROW}	σ_{COL}	CONTROL POINT	CHECK POINT	CONTROL POINT	CHECK POINT
0	0	<1	<1	15	11
.29 (23 m)	.31 (18 m)	30	31	30	43
.29 (23 m)	.58 (34 m)	40	40	36	50
.29 (23 m)	.76 (44 m)	48	48	41	56
.30 (24 m)	1.04 (60 m)	60	61	47	70
.31 (25 m)	1.53 (89 m)	84	85	64	96
.33 (26 m)	2.02 (117 m)	109	109	79	122
.35 (28 m)	5.01 (291 m)	261	262	179	289

Case I: Model: Ellipsoidal Earth, Elliptical Orbit

156 Control Points

156 Check Points

$\sigma_{control}$: 0

Case R: Model: Ellipsoidal Earth, Elliptical Orbit

25 Control Points

156 Check Points

$\sigma_{control}$: $\sigma_x = \sigma_y = \sigma_z = 15$ m

$\sigma_{total} = 26$ m

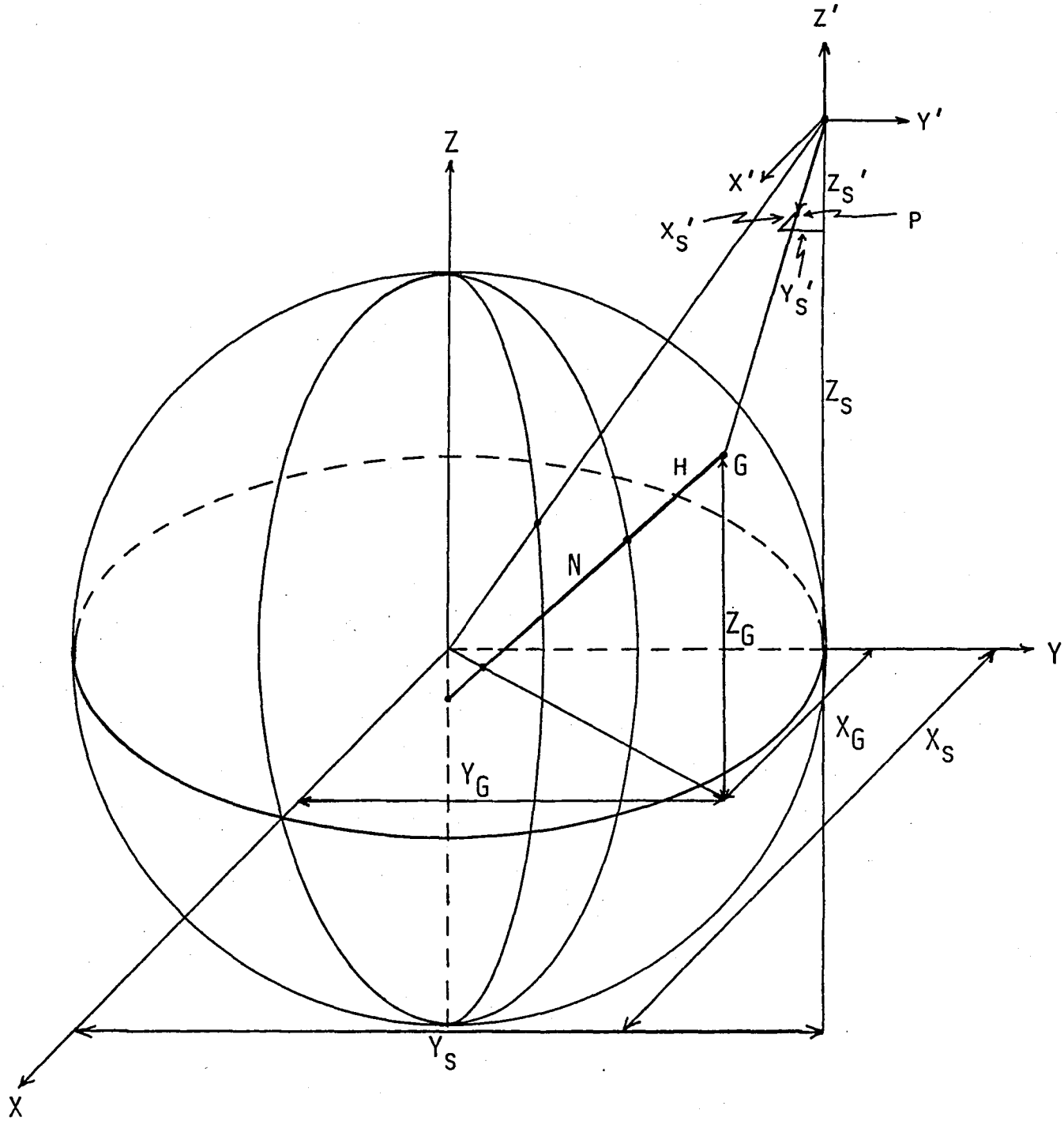


FIGURE 1

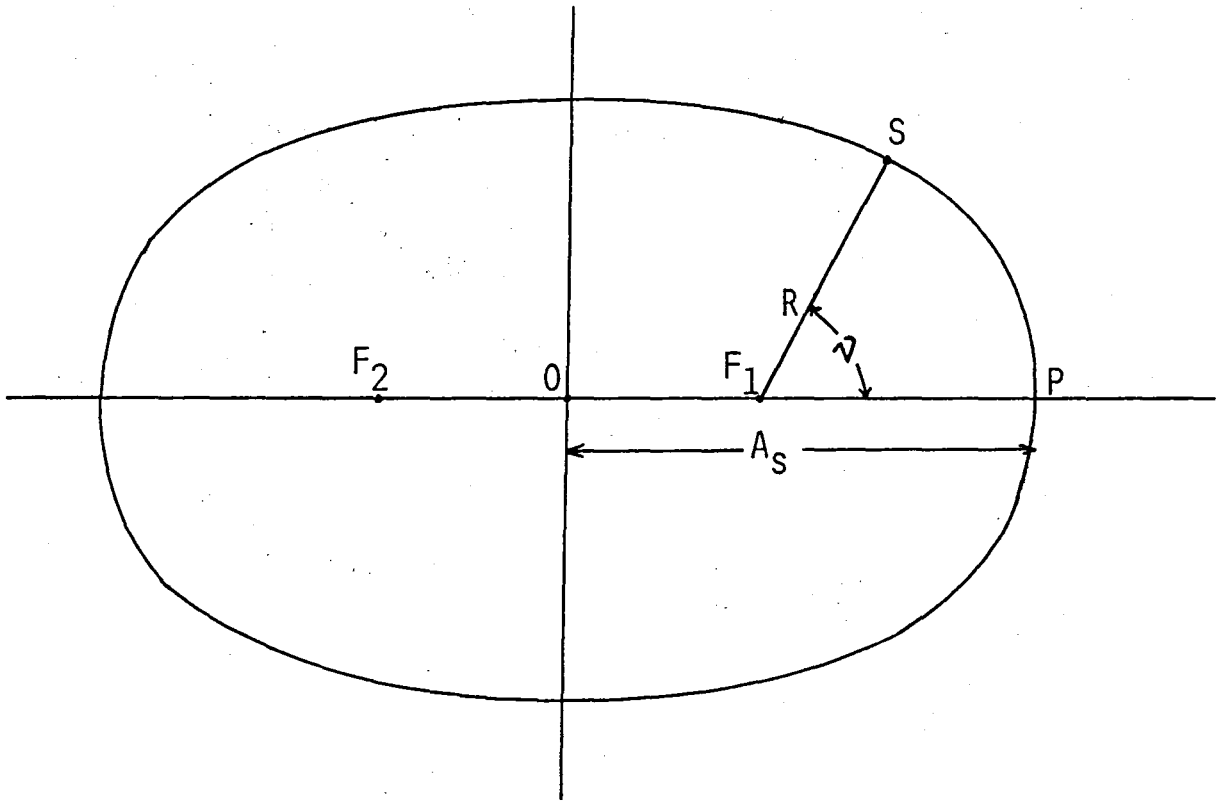


FIGURE 2 SATELLITE ORBITAL PLANE

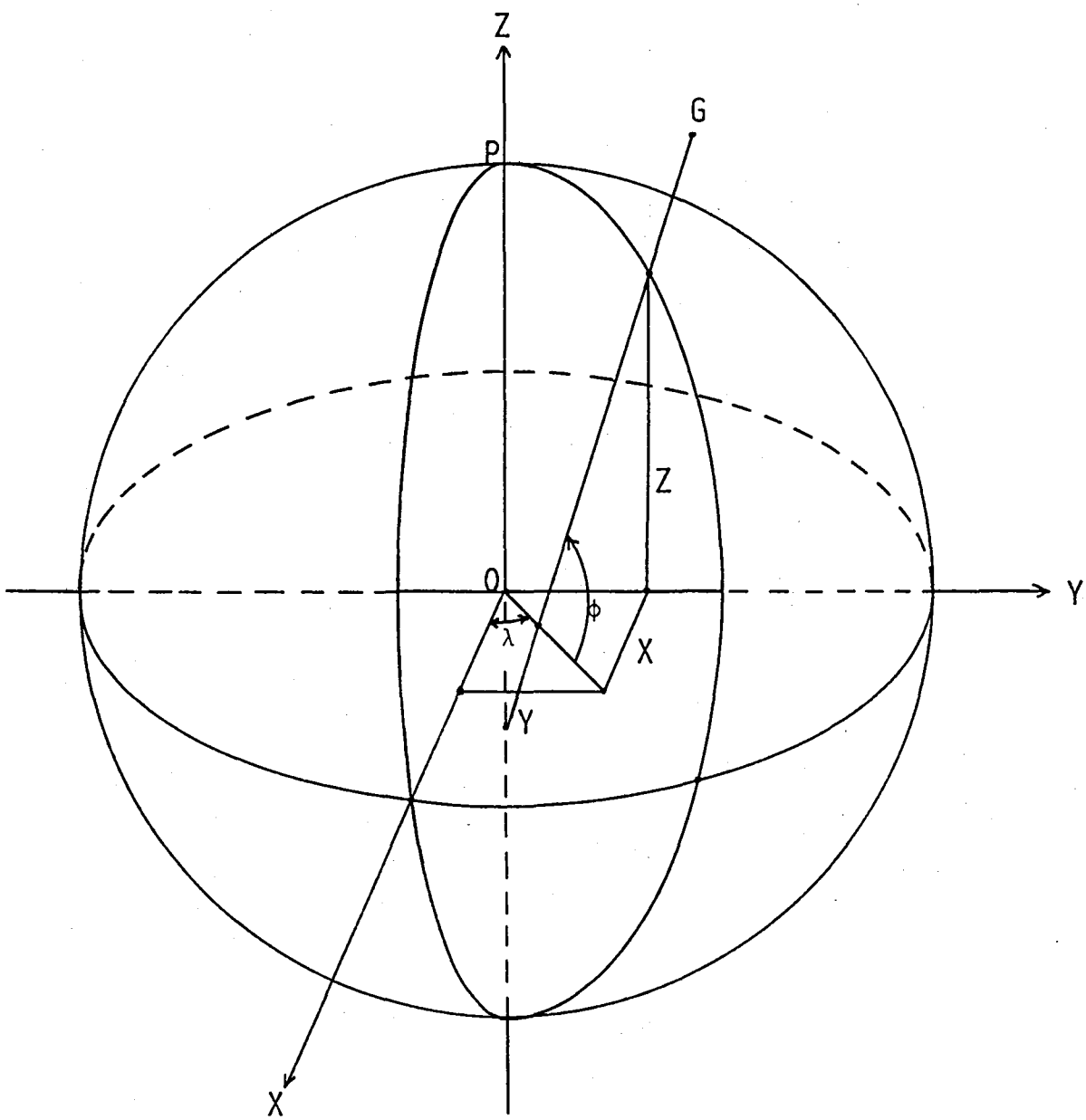


FIGURE 3A ELLIPSOID OF REVOLUTION

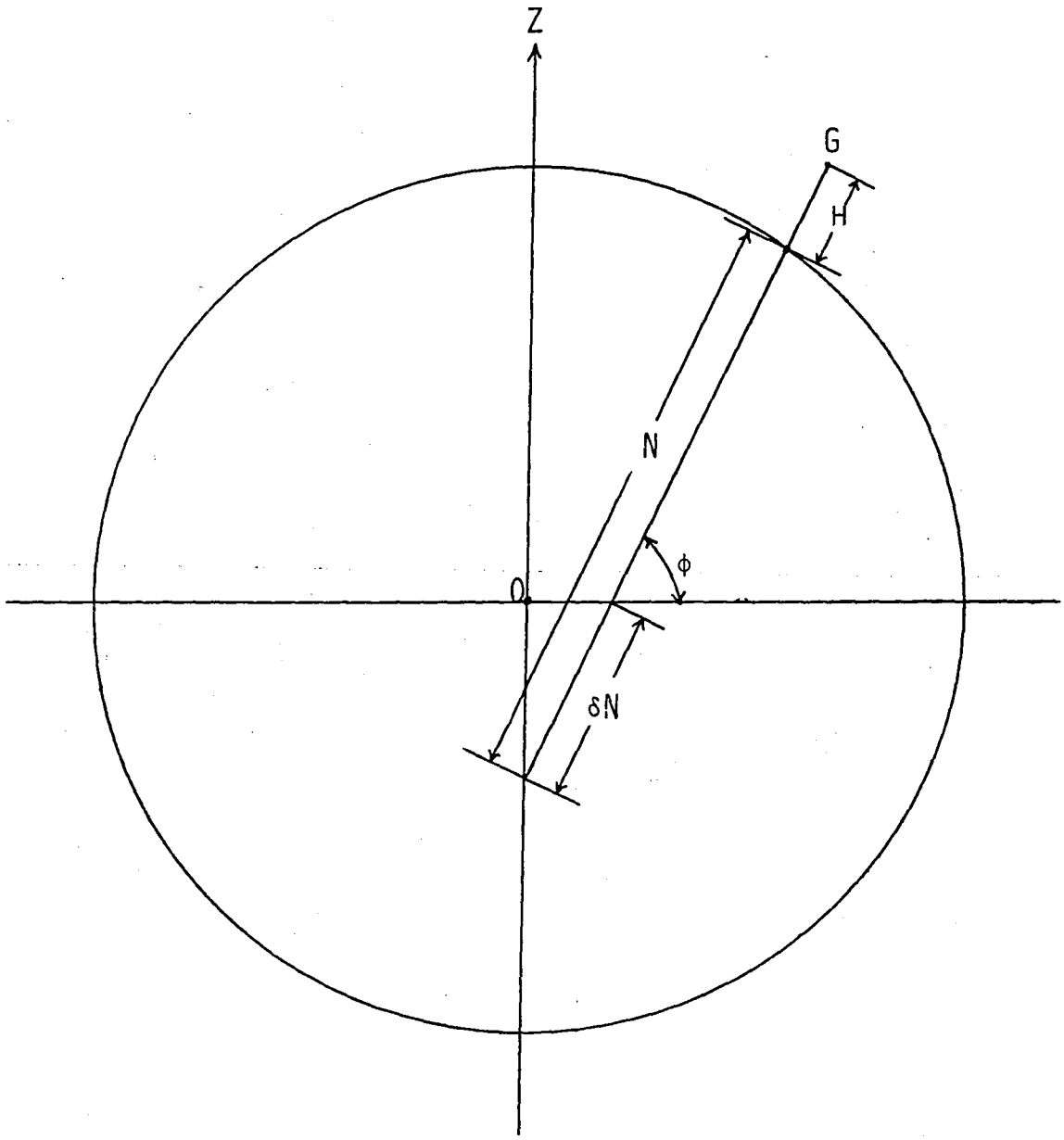


FIGURE 3B PLANE OF THE MERIDIAN

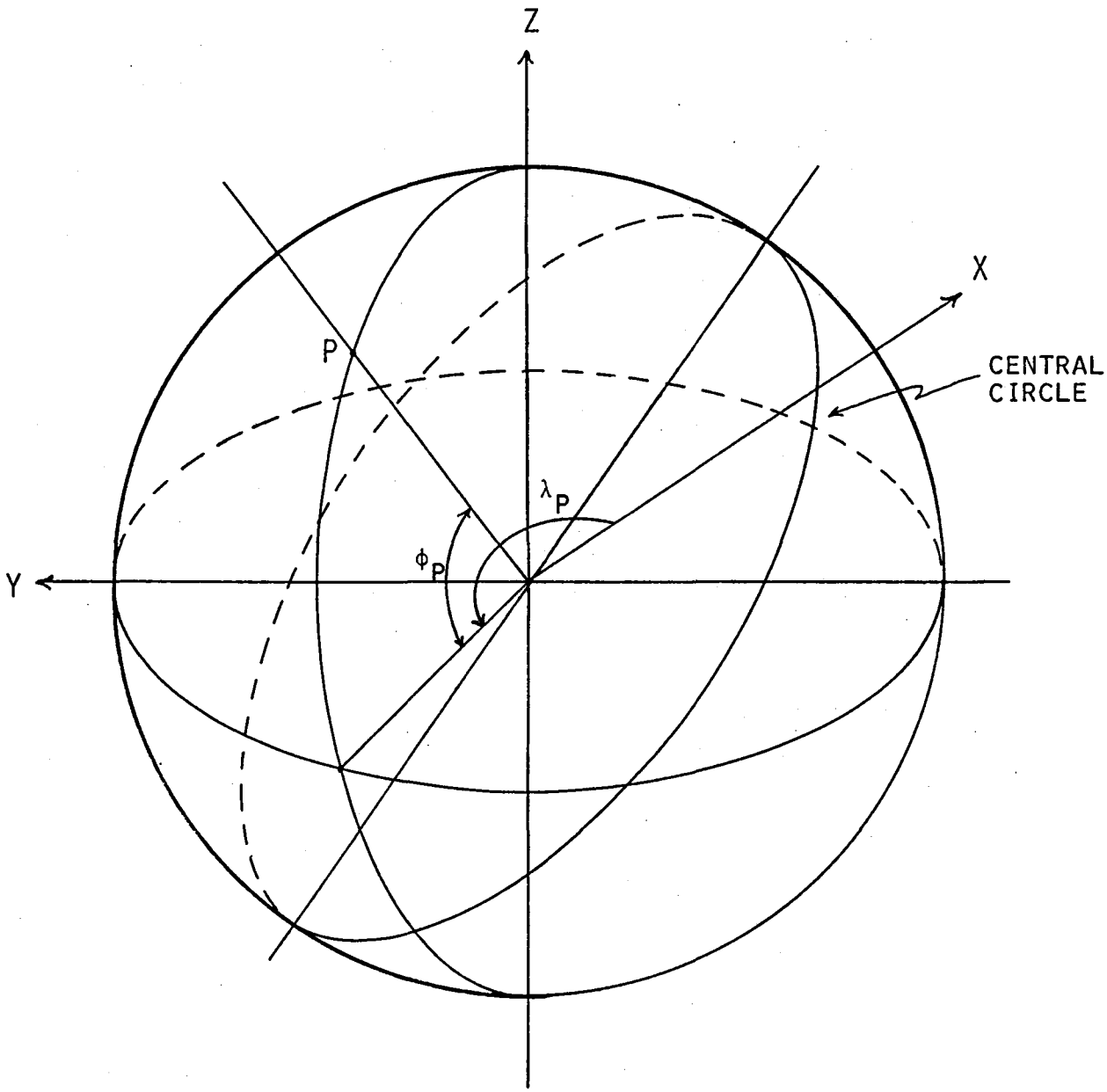


FIGURE 4A

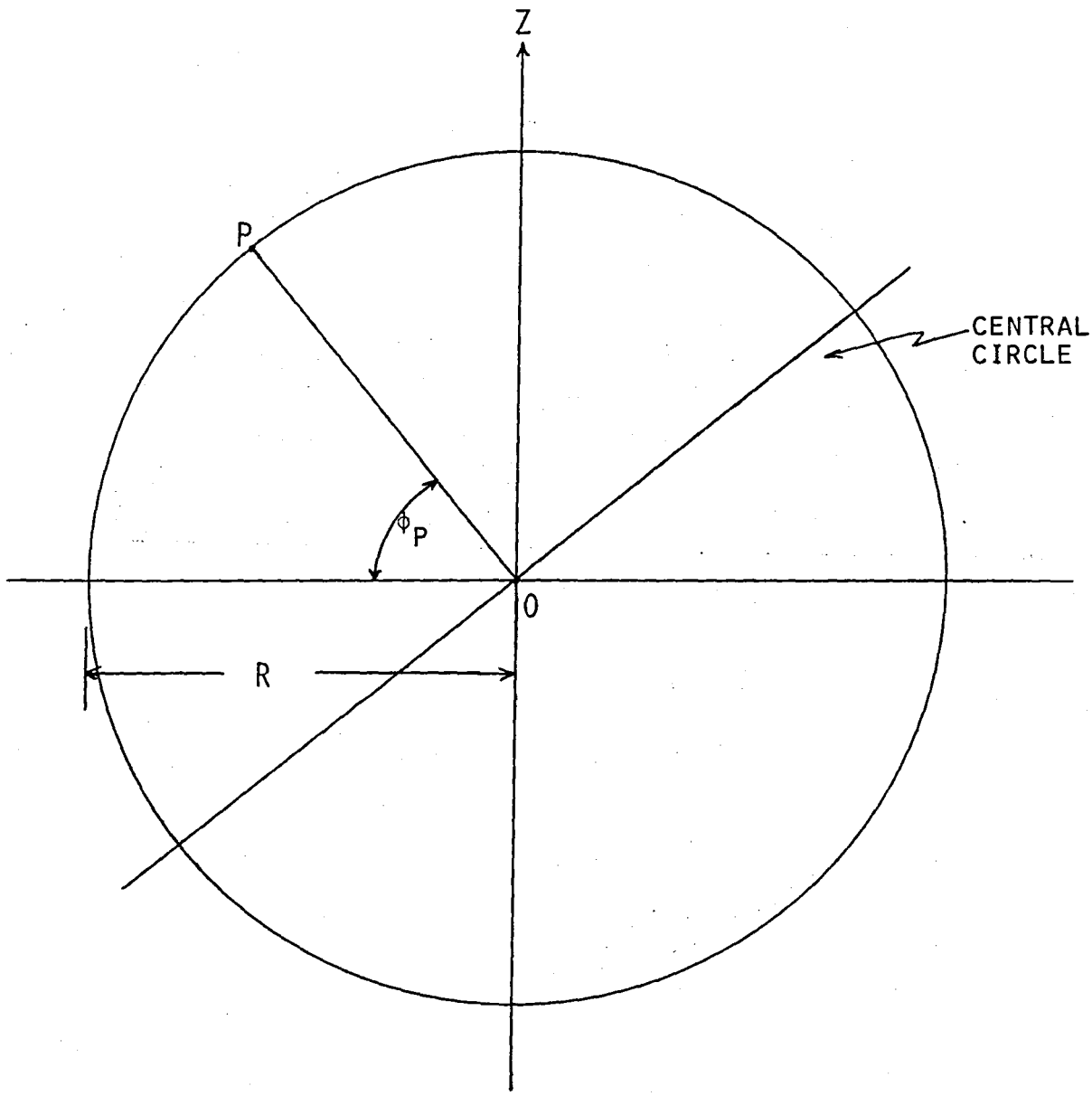


FIGURE 4B

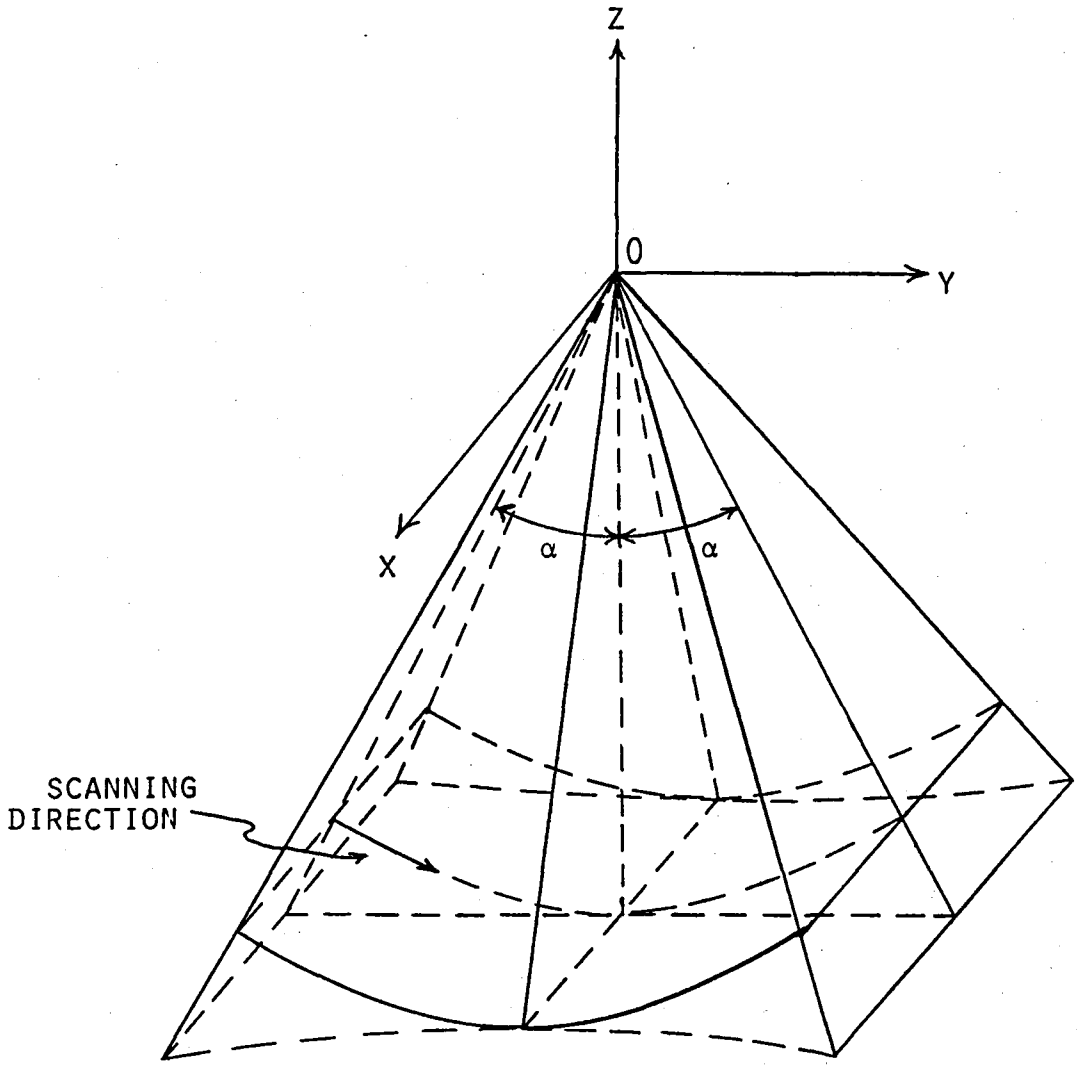


FIGURE 5 SCANNER COORDINATE SYSTEM

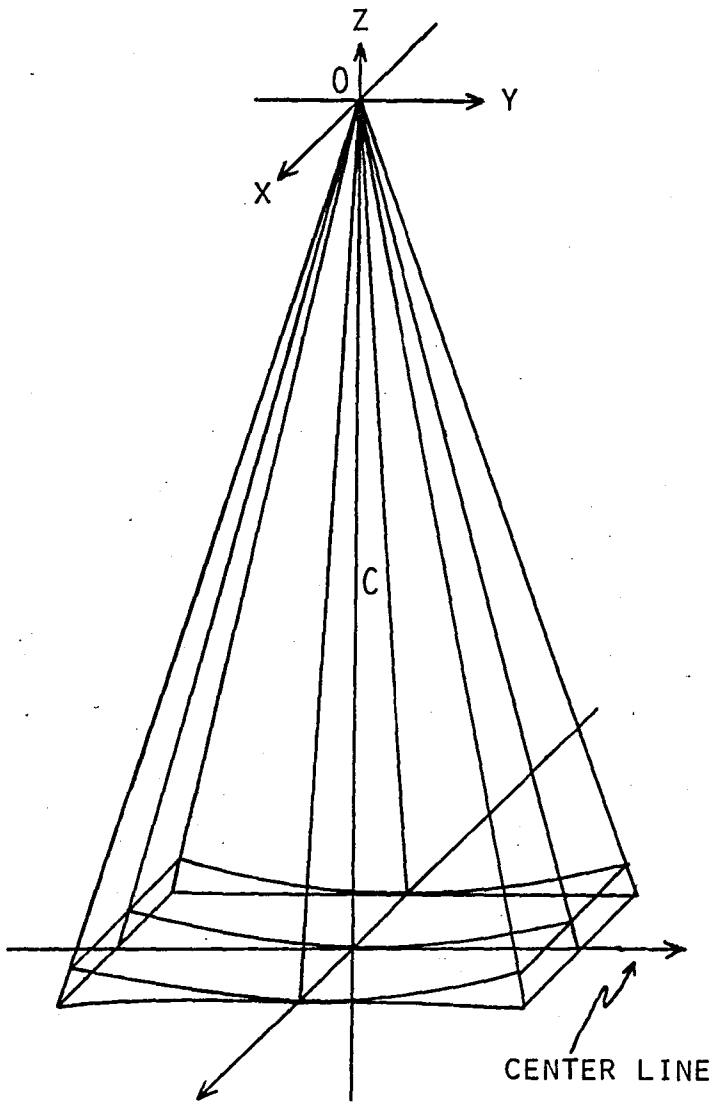


FIGURE 6A ONE MSS SCAN

(NO SCAN LINE CORRECTION)

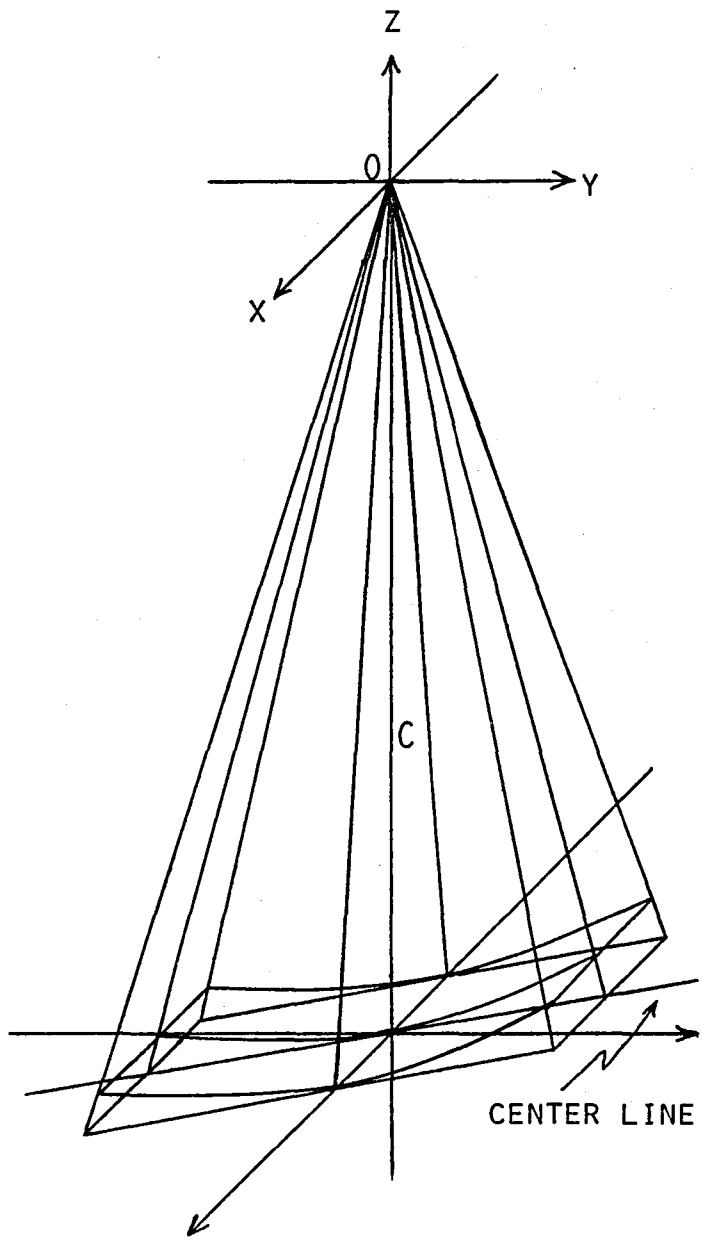


FIGURE 6B FORWARD TM SCAN

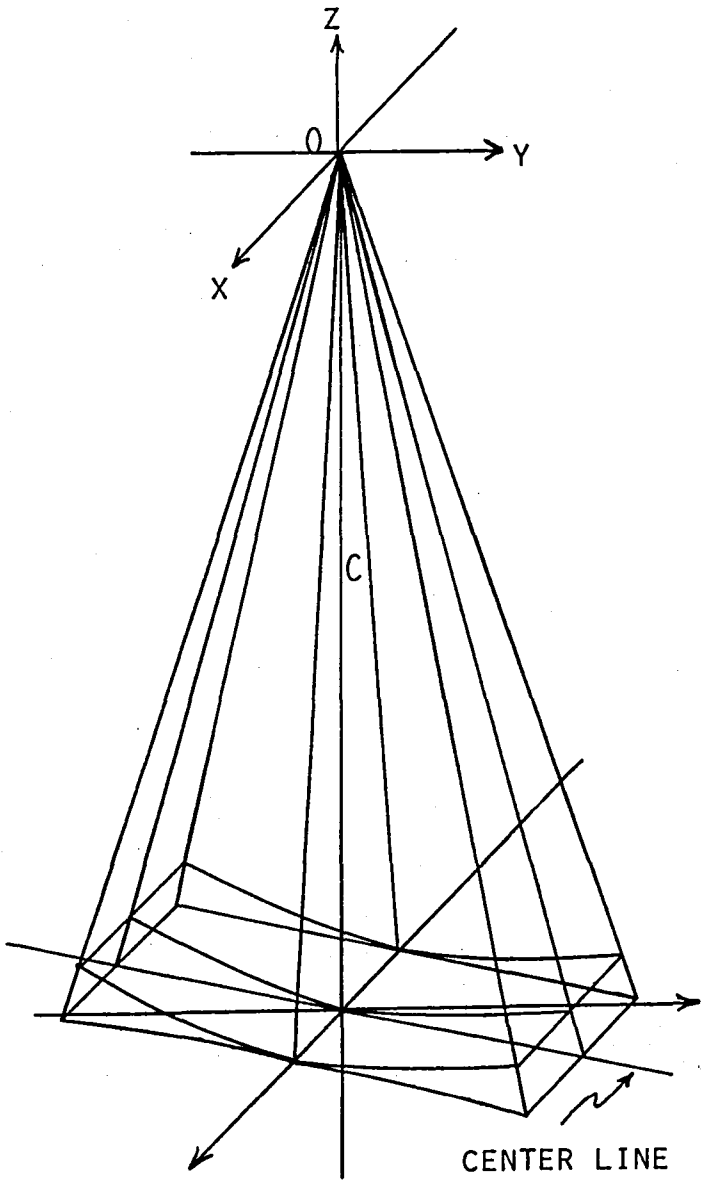


FIGURE 6c REVERSE TM SCAN

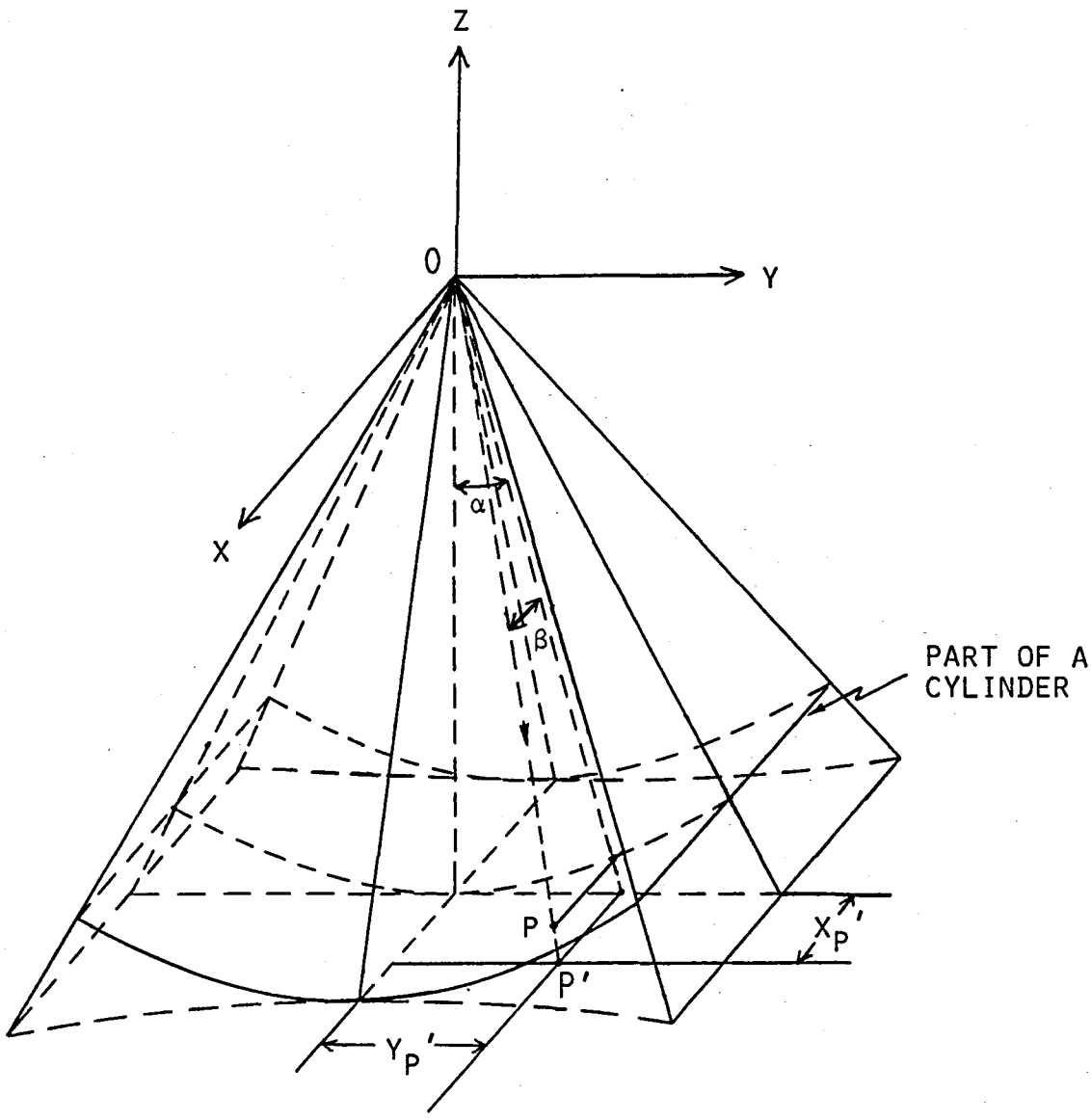


FIGURE 7 SCANNER GEOMETRY

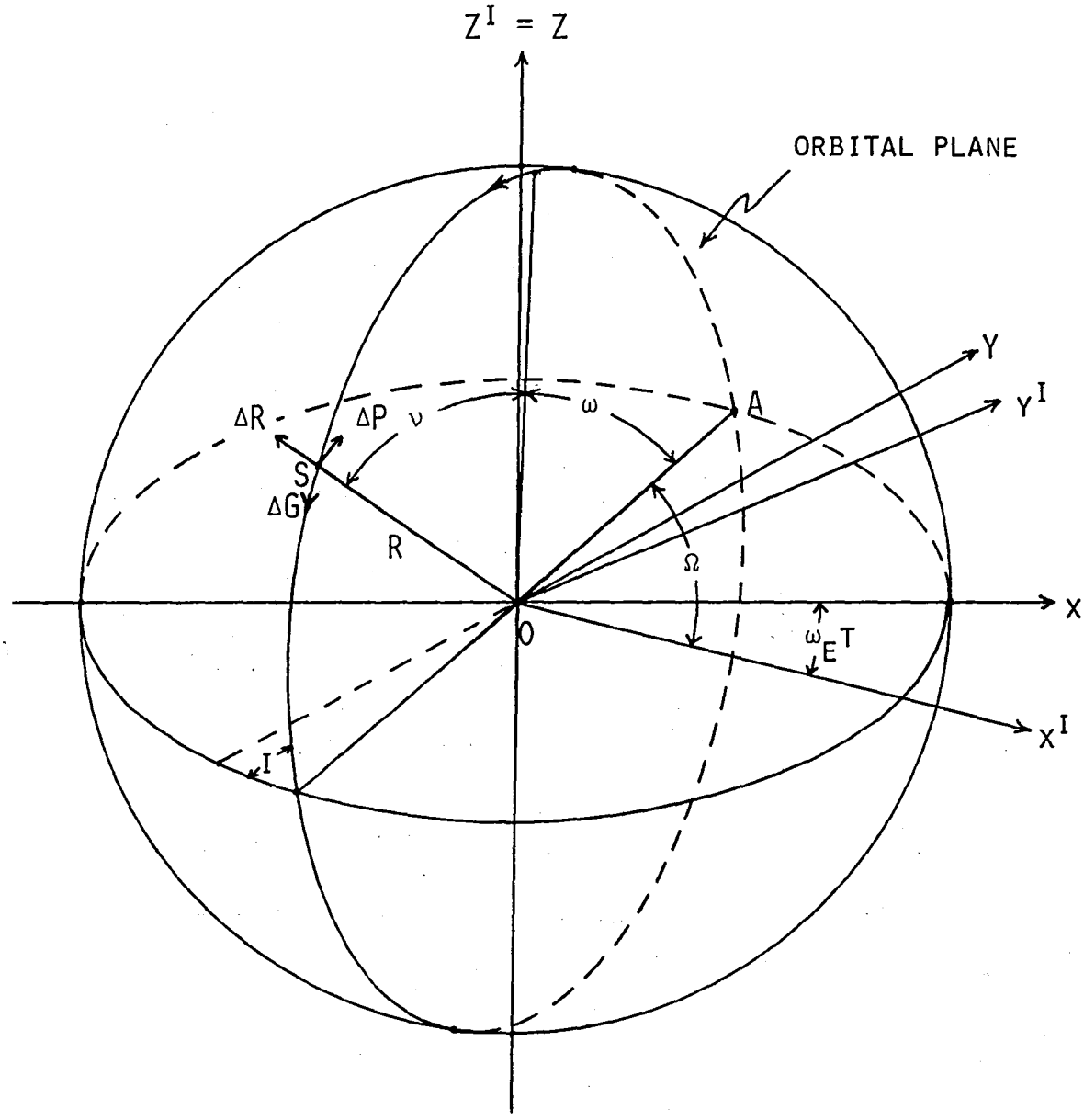


FIGURE 8 GEOMETRY OF THE SATELLITE ORBIT

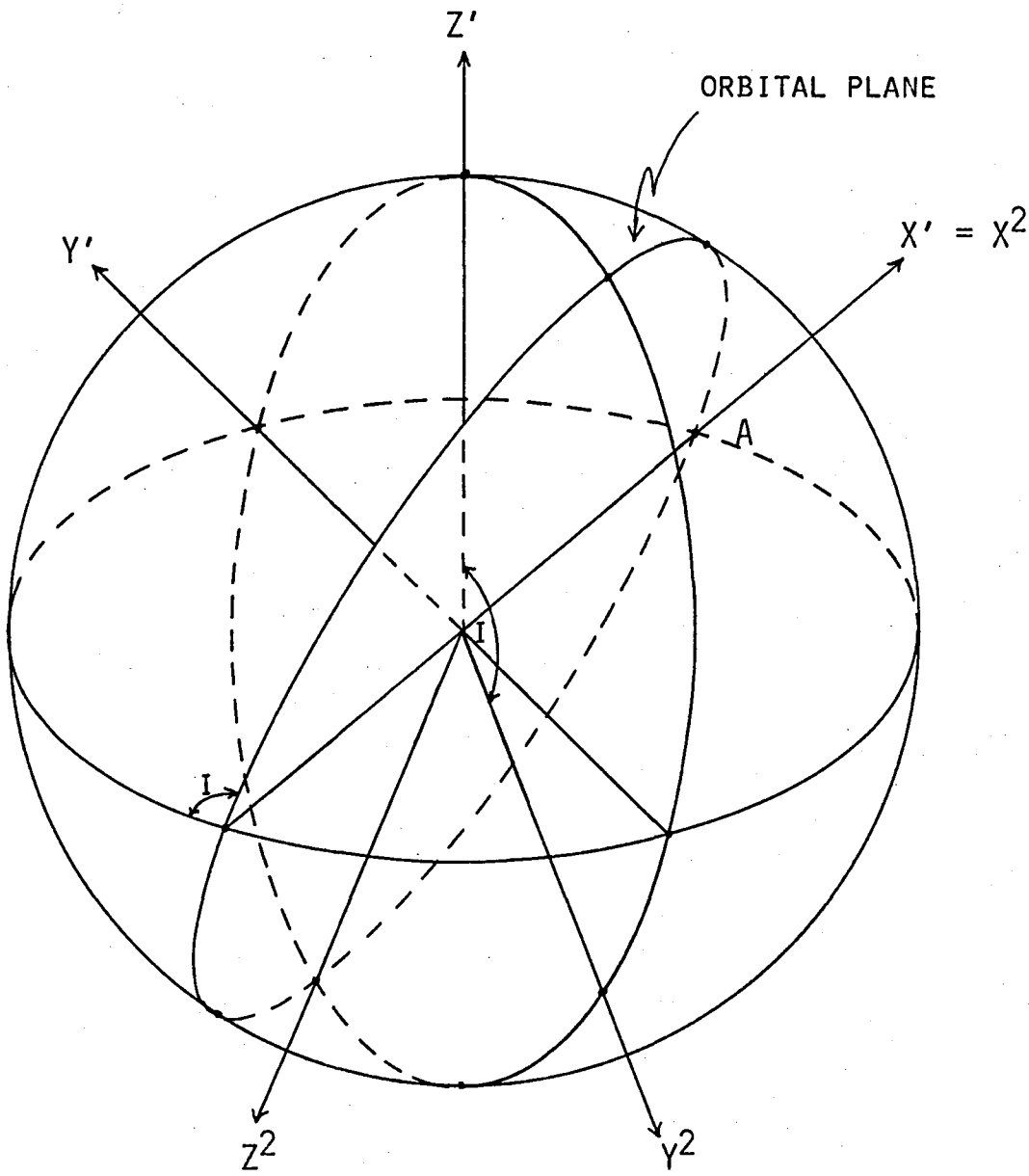


FIGURE 9

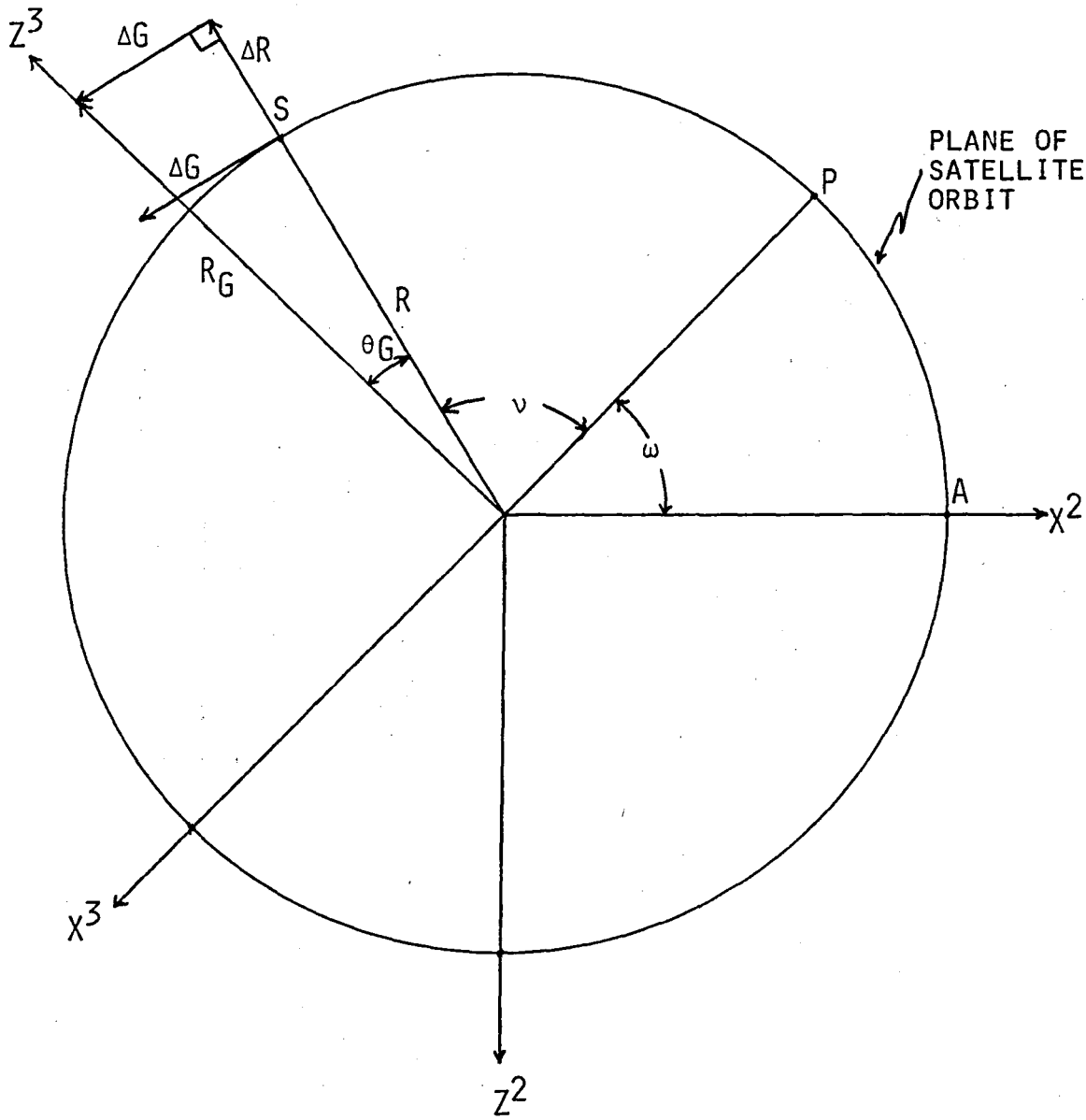


FIGURE 10

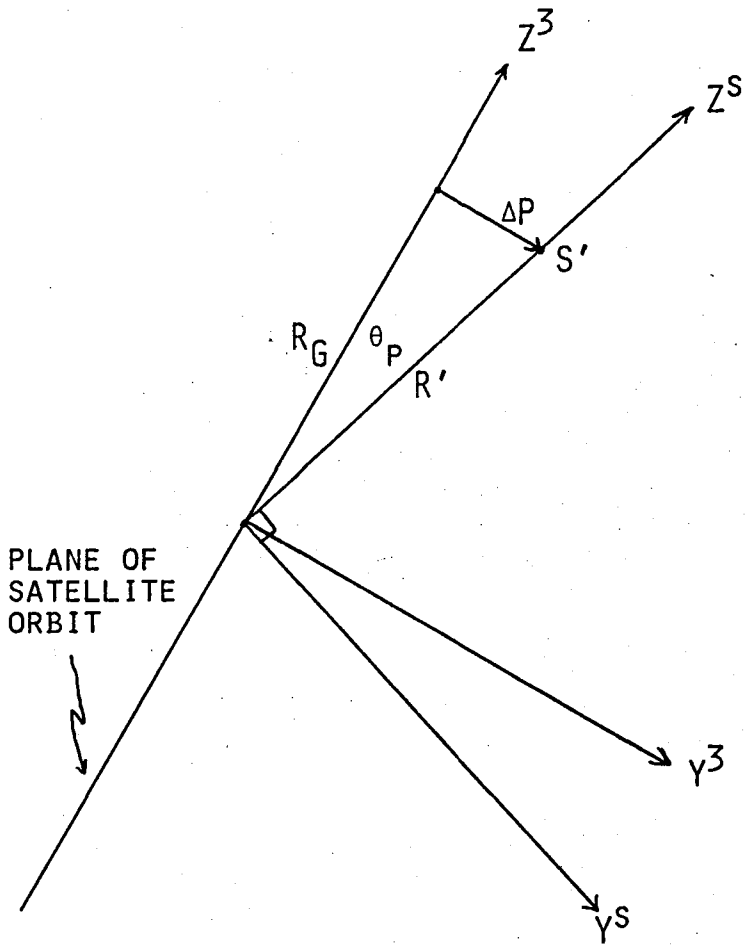


FIGURE 11

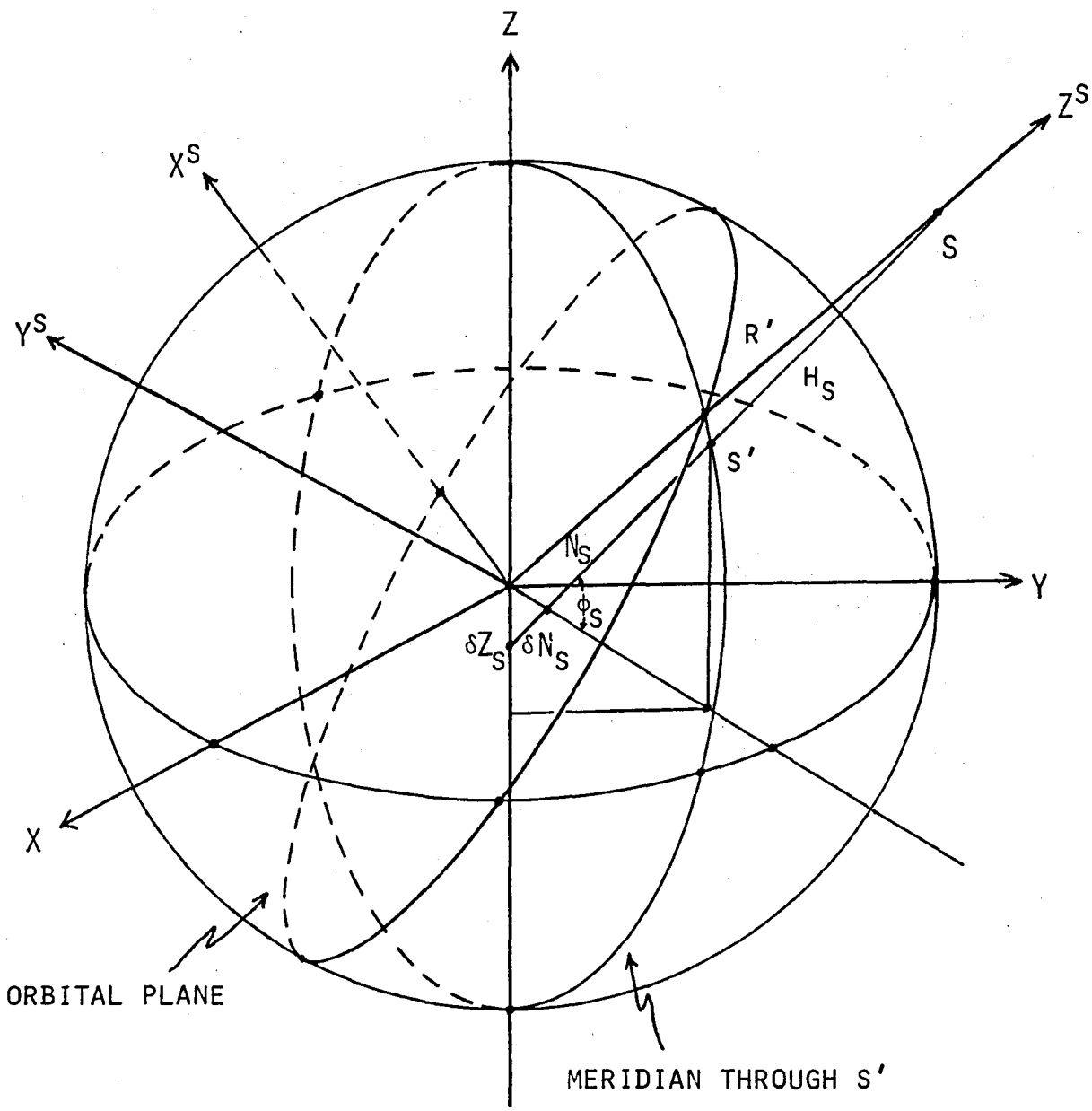

Z
Z^S
X^S
S
Y^S
R'
H_S
S'
N_S
Y
ϕ_S
δZ_S
δN_S
X
ORBITAL PLANE
MERIDIAN THROUGH S'

FIGURE 12

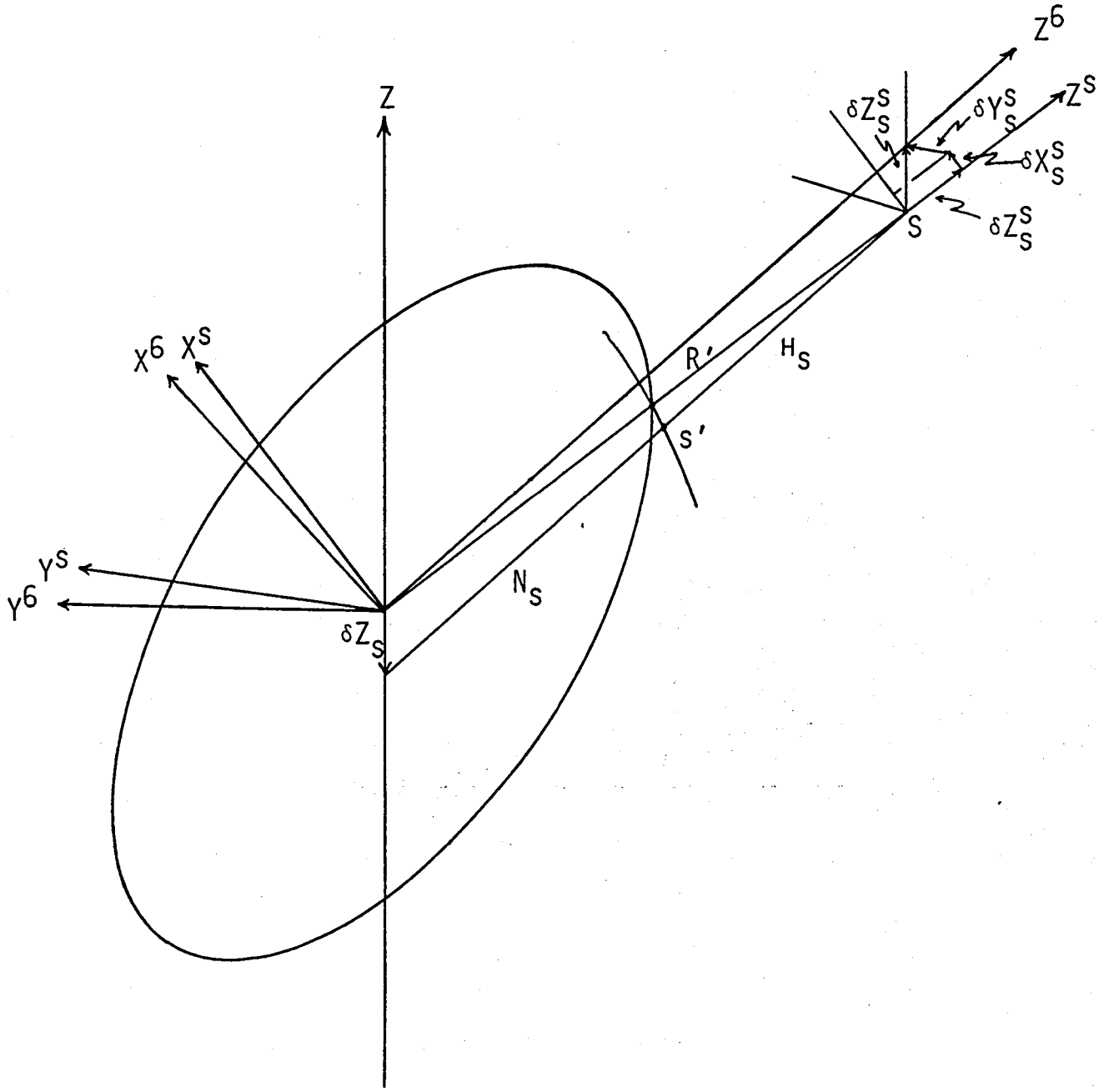

Z
Z^G
Z^S
δZ^S_S
δY^S_S
δX^S_S
S
δZ^S_S
X^G X^S
R'
H_S
S'
Y^S
Y^G
N_S
δZ_S

FIGURE 13

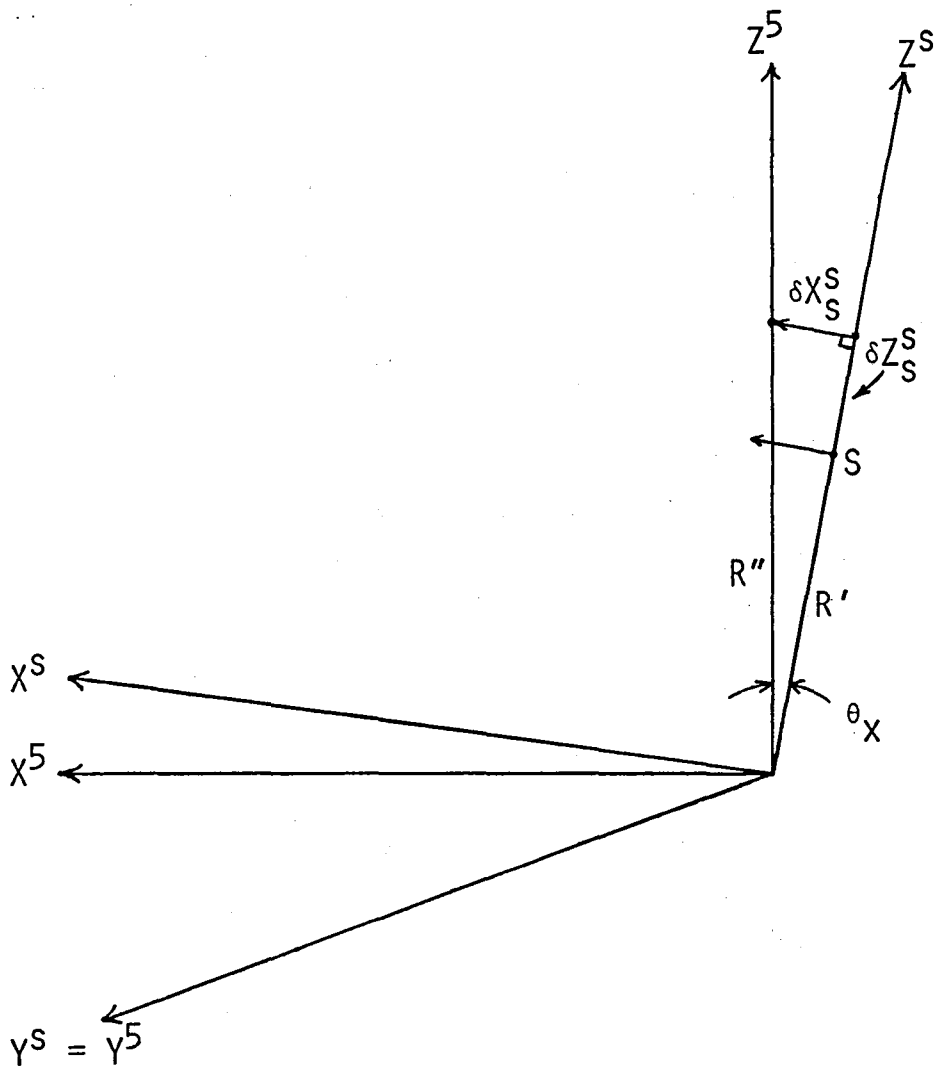


FIGURE 14

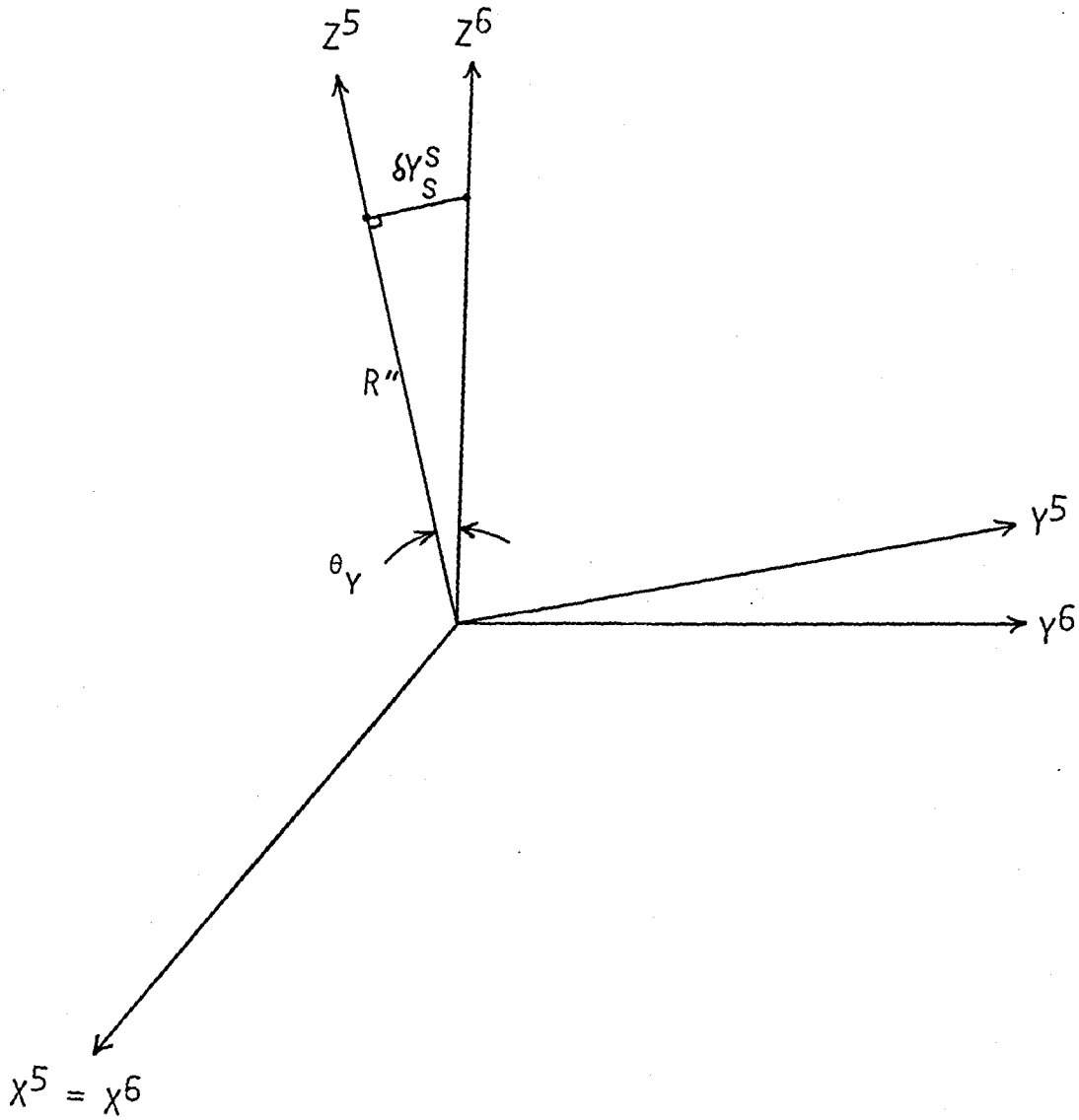


FIGURE 15

Page intentionally left blank

PROGRESS IN THE SCENE-TO-MAP REGISTRATION INVESTIGATION

D.D. DOW
NASA/National Space Technology Laboratories
Earth Resources Laboratory

ABSTRACT

This investigation focuses on the geometric accuracy of the scene-to-map registration process for P-format Landsat MSS data for scenes from Kansas and Louisiana/Mississippi. Large scale row and column bias values and row and column standard deviation values were measured for the P-format data sets indicating a poor georegistration accuracy for these geometrically corrected Landsat MSS scenes. Experimental work is underway with A-format Landsat MSS scenes from the same locations to examine the influence of the number of ground control points and the spatial distribution of ground control points on geometric registration accuracy. An early conclusion from this work is that the root mean square approach for assessing how well the ground control points fit the mapping equations measures a different aspect of geo-registration accuracy than does the approach of evaluating the bias (offset) and standard deviation using independently chosen ground reference points.

INTRODUCTION

The scene-to-map registration process is a crucial step in the preprocessing of Landsat Multispectral Scanner (MSS) and Thematic Mapper (TM) data. Georeferenced Landsat MSS products approach the national map accuracy standards for the 1:250,000 scale (USGS, 1979a). This has resulted in the utilization of the Landsat data to develop map products, to serve as a component of a multisource data base, and in change detection of land cover categories through a comparison of post-classification products developed at two different points in time. The registration and rectification of Landsat data is accompanied by geometric offsets resulting from the remapping techniques employed and radiometric distortions resulting from the resampling functions used. This study focuses on the factors influencing geometric fidelity. The factors to be examined include the spatial distribution of the ground control points utilized and the number of ground control points employed. The influence of resampling functions on geometric errors should be less than half a pixel and would only become an important factor for georeferencing Landsat products at a sub-pixel level of accuracy.

Landsat computer compatible tapes (CCT) are available in the A-format which has been radiometrically corrected and in the P-format which includes radiometric and geometric corrections. The A-format Landsat MSS data is processed through the Master Data Processor (MDP) at Goddard Space Flight Center to remove the gap problem inherent in MSS data, without resampling the data. The P-format Landsat MSS data comes in a geometrically converted form which in the standard product

employs a Hotine Oblique Mercator (HOM) projection as a map base and cubic convolution resampling. The users of Landsat data for geographic information systems face the problem that the base map projection for their work most often utilizes the Universal Transverse Mercator (UTM) system, while the base for A- and P-format MSS data is the HOM system and for the Landsat 4 TM data is the Space Oblique Mercator (SOM) system. The EROS Data Center Digital Image Processing System (EDIPS) in Sioux Falls, South Dakota has developed software to convert from one of the above map projection systems to another. The UTM system imparts a scale distortion of 1 part in 1,000 (1:1,000) compared to the 1:10,000 distortion associated with the SOM and HOM projections (USGS, 1980, a, b, and c.).

The number of ground control points (GCPs) used to geometrically correct P-format Landsat MSS tapes is listed in the CCT header record as the quality assessment number. The quality assessment number is the truncated integer of the expression (N + 7)/8, where N is the number of control points used. If no GCPs were utilized, then the P-format CCT is referred to as system corrected. Currently, all of the Landsat 4 TM tapes are system corrected to produce P-format products. For Landsat MSS products that have been system corrected, the georegistration accuracy will be within 60 pixels 99 percent of the time. When 25 to 50 GCPs are used in a Landsat scene, the georegistration accuracy will be within 1 pixel more than 99 percent of the time. The georegistration accuracy is 10 pixels for 8 to 24 GCPs and 20 pixels for 1 to 7 GCPs (Nelson and Grebowsky, 1982). A recent study by Graham and Luebbe (1981) showed that the quality

assessment number is not necessarily a good indicator of registration accuracy.

Investigations of scene-to-map registration accuracy can be divided into theoretical and empirical studies. Some examples of each type of investigation will be discussed in the following sections, beginning with the theoretical approach. Sawada et al. (1981) developed an analytical model utilizing satellite orbit/attitude information from the Scene Image Annotation Tape (SIAT) plus data on characteristics of the MSS scanning mechanism to correct geometrical distortions to within one pixel accuracy utilizing 3 GCPs to estimate nonlinear scan mirror corrections and 20 GCPs for error estimation. A second approach is to fit MSS images to ground control by means of different mathematical models and to analyze the residuals for each mathematical model as a means of determining which model will produce the greatest geometric accuracy given a specified configuration of GCPs (Wong, 1975; Steiner and Kirby, 1977; Dowman and Mohamed, 1981). Wong (1975) achieved the best results with a 20 term polynomial employing 25 to 30 ground control points with a reported limiting geometric accuracy of $\pm$55 meters. Dowman and Mohamed (1981) achieved a root mean square (rms) error of 83 meters using no GCPs, while the rms error was approximately 60 meters when 20 GCPs were used.

The empirical approach to the scene-to-map registration accuracy assessment involves selecting a second set of independently chosen ground reference points (GRPs) and comparing their location on the map with that in the georeferenced Landsat MSS product. A system corrected P-format product accuracy assessment reported standard error

in both directions of 160 meters which was reduced to 50 meters after the application of a linear least-square analysis correction procedure (USGS, 1979a). A second study of P-format data which employed GCPs from a 1:24,000 scale topographic map reported rms errors of 218 meters in the east-west direction and 880 meters in a north-south direction (Colwell et al., 1980). The first 12 lines of Table 1 presents the results of a recent investigation that examined 12 different Landsat MSS scenes in the P-format and compared the location accuracy of the tick marks in the Landsat scene by using independently chosen GRPs (Graham and Luebbe, 1981). The row offset (bias) over 12 Landsat scenes varied from -414.8 to 15.8, while the column offset (bias) varied from -0.8 to 9.5. In this case the results are given in multiples of the size of one georegistered pixel (57 meters). All of these studies suggest a need for a systematic investigation of the problems with P-format MSS data that causes distortions in the scene-to-map registration process.

METHODS

The Landsat MSS frames to be used in this study were acquired over southeastern Louisiana and coastal Mississippi (path: 23; row: 39 of the the worldwide reference system) and over eastern Kansas and western Missouri (path: 29; row: 33). The Kansas data was collected on 11/9/81 and had a quality assessment number of 2, while the Louisiana data was gathered on 11/21/81 and had a quality assessment number of 3. Both Landsat MSS scenes had 10 percent cloud cover. The Louisiana Landsat scene includes open water (Lake Pontchartrain) areas and wetlands adjacent to the metropolitan New Orleans area in which it

is difficult to choose GCPs and GRPs. The Kansas Landsat scene was more amendable to choosing evenly spaced GCPs and GRPs.

The points to be utilized for GCPs and GRPs were chosen on 1:24,000 scale, 7.5 minute quadrangle sheets produced by the U.S. Geological Survey (USGS). Where possible, three ground control or reference points were located on each 7.5 minute quadrangle sheet and the same points were identified on the Landsat scene of A-format MSS tapes. The ground points map coordinates were recorded in the UTM system as northings and eastings, while the Landsat coordinates were recorded as rows and elements. For the Louisiana P-format Landsat MSS scene 192 ground points were chosen, while 359 ground points were used for the A-format data. For the Kansas P-format Landsat MSS scene 145 ground points were chosen and 356 ground points were picked for the A-format CCT. More points were utilized for the A-format data, since the points had to be used for GCPs to carry out the georegistration procedure and GRPs to independently check the accuracy of the georegistration procedure. The types of features used as ground points included manmade (road intersections) and natural (river intersections) categories. Steiner and Kirby (1977) discuss the accuracy with which ground points can be chosen both on maps and in Landsat scenes. Since there is excellent registration between bands in the MSS (Colvocoresses and McEwen, 1973), it is not necessary to make corrections in ground point locations on the Landsat scene when different MSS bands have been utilized in detecting the ground features.

The approach used to measure the accuracy of the GCPs as a set was

to compare them to a linear polynomial model of the form:

(1) $SL = A_1 + A_2 E + A_3 N + e$

(2) $CE = B_1 + B_2 E + B_3 N + e$

where "SL" represents the scan line coordinate, "CE" represent the corrected elements, "E" represents the UTM easting, "N" represents the UTM northing, "A_1" to "A_3" and "B_1" to "B_3" are constants, and "e" represents the residual error. The root mean square (rms) determination quantifies how far the measured GCP coordinates differ from the GCP coordinates computed from the linear polynomial model. That is:

(3) $RMS = \sqrt{\sum \left[SL \text{ measured} - (A_1 + A_2 E + A_3 N) \right]^2 / dF}$

(4) $RMS = \sqrt{\sum \left[CE \text{ measured} - (B_1 + B_2 E + B_3 N) \right]^2 / dF}$

where the terms are defined as before and dF equal the degrees of freedom.

When the residual error was large for a given GCP, this suggested the possibility that the ground point coordinates may have been misread from either the map or the Landsat image. A check was made of the coordinates and corrections were made where necessary. If the point coordinates appeared to be accurate and the point had a large residual error, the point was kept. The rms value is a measure of how well the set of GCPs employed fit the mapping equations (linear polynomial model).

To evaluate the georegistration accuracy of P-format Landsat MSS data, an independently chosen set of ground reference points (GRP) was selected. The procedure of Graham and Luebbe (1981) was used to quantify the georegistration accuracy in terms of RBIAS (row offset), CBIAS (column offset), RSD (row standard deviation) and CSD (column

standard deviations). High georegistration accuracy would be characterized by sub-pixel bias and standard deviation values. The equations for computing bias and standard deviation are:

$$(5)\quad RBIAS = \frac{\sum_{i=1}^{NP} (ROW1_i - ROW2_i)}{NP}$$

$$(6)\quad RSD = \sqrt{\frac{\sum_{i=1}^{NP} (ROW1_i - ROW2_i - RBIAS)^2}{NP - 1}}$$

where NP is the number of GRPs chosen, ROW1 is the Landsat row determined using the EROS software, and ROW2 is the Landsat row read from the Landsat imagery. For the A-format Landsat MSS tapes, ROW1 is the Landsat row determined using the mapping equations which are computed from the GCPs. The ELAS module TRAN which contains the EROS subroutine PIXGEO converts UTM coordinates to Landsat row and column (elements) coordinates. The error introduced by the module TRAN is less then $\pm$ 1/2 Landsat pixel (Graham and Luebbe, 1981). The operation of the module TRAN was checked by comparing the apparent and actual location of the tick marks on the P-format Landsat MSS tape.

One of the objectives of this study is to determine how the spatial distribution of GCPs influences the resulting accuracy of the georegistration process. To characterize the spatial distribution of points, the approach of measuring the distance from a point to its nearest neighbor, irrespective of direction was employed (Clark and Evans, 1954). The module CSPA was developed to compute the parameter "R" which compares the mean observed nearest neighbor distance to the mean nearest neighbor distance if the population was distributed at

random. The "R" values can range from 0 (maximum aggregation or clumping of points) to 2.15 (maximum spacing or a regular distribution of points). In this analysis "R" values between 0.7 and 1.3 were taken to indicate a random distribution of points, values below 0.7 indicated a clustered distribution and a value above 1.3 indicated a regular distribution of points. Figure 1 shows the spatial distribution of GCPs.

RESULTS AND DISCUSSION

The georegistration accuracy assessment of the P-format Landsat MSS tapes is given on the last two lines in Table 1. Both the Kansas and the Louisiana/ Mississippi P-format MSS data show high RBIAS and RSD values and fairly high CBIAS and CSD values. The other values in Table 1 are the results of Graham and Luebbe (1981) using the same accuracy assessment methodology. Data sets 5 and 6 of Graham and Luebbe (1981) which had high RBIAS values, attributed the error to inaccuracies in the tick mark registration information on the CCT. The fact that for the 1981 data for Kansas and Louisiana/Mississippi had both high BIAS and SD values, suggests that some other factor is responsible for the very poor georegistration accuracy. A visual examination of the Kansas P-format data for 1981 revealed that the section boundaries which should have been squared on the Landsat image were instead rectangular and that roads that ran north-south on the map run northwest-southeast on the Landsat image. This information suggests that the 1981 P-format data for Kansas and Louisiana/Mississippi is distorted in other ways besides a simple north-south translation.

The P-format ground points for the 1981 data for Kansas and Louisiana/Mississippi were divided into 8 to 32 randomly chosen GCPs with the rest of the ground points used as GRPs. The GCPs were run through the ELAS georegistration module BMGC and the BIAS and SD were computed as explained in equations (5) and (6). The results of this analysis is given in Table 2 where it can be seen that the ELAS georegistration procedures (Graham et al., 1980) operated on P-format data gave sub-pixel geometric accuracy. Since this procedure involves resampling the data twice, it presumably introduces radiometric distortions into the data. The RBIAS results for Louisiana and the CBIAS results for Kansas suggest a trend of decreasing BIAS values through the use of increasing numbers of GCPs. No firm conclusions can be drawn in this regard, since the study was done without any replicates. Table 3 presents a similar type of study using A-format MSS data without any replicates. For a given number of GCPs the A-format data appears to have lower BIAS and SD values than does the P-format data. The important conclusion is that both the A-format and P-format data provide sub-pixel georegistration accuracy when as few as 8 GCPs are used on a whole Landsat scene. This study chose GCPs in groups of eight, so that when 16 GCPs were used in one run and 24 GCPs were utilized in the next run, the two sets of data shared 16, randomly chosen GCPs in common. This procedure was followed to reduce the variation in the different data sets.

The next phase of the study was to examine the influence of the spatial distribution of GCPs on the accuracy with which it is possible to georegister A-format MSS data utilizing the ELAS scene-to-map

registration software (Graham et al., 1980). The results of the initial phase of this investigation is presented in Table 4. This analysis involved 20% of a Landsat scene which utilized 8 GCPs to develop the mapping equations and the rest of the ground points to act as GRPs in order to quantify the georegistration accuracy. None of the numbers in Table 4 are statistically different at the P = 0.10 level of significance for the 5 replicates measured for the Kansas and Louisiana data. The general trend is for the BIAS values for rows and columns to increase in magnitude as one goes from a random to a regular to a clustered distribution. There is no clear general trend apparent for SD results. The CSPA module with its numerical criteria was used as described in the methods to distinguish whether the distribution of 8 GCPs followed a random, regular or clustered pattern.

The next phase of the study utilizing 20% of a Landsat scene examined the question of the relative importance of the number of GCPs versus the spatial distribution of the GCPs. Since the number of ground points in 20% of a Landsat scene varied from 28 to 40, it was decided to combine the Kansas and Louisiana data sets for this analysis. The results are presented in Table 5. The general trend is for the clustered distribution of points to have greater geometric inaccuracy (both BIAS and SD) than the random distribution of points, both for the case of 8 (statistically significant CBIAS results) and 16 GCPs. In going from 8 GCPs with a random distribution to 16 GCPs, the random distribution exhibits greater georegistration accuracy for both BIAS and SD than does the clustered spatial distribution of

points. This is another area in need of additional work, but the preliminary analysis suggests that georegistration accuracy is more sensitive to the number of GCPs used than it is to the spatial distribution of GCPs.

A final question of interest is the relationship between the RMS method of assessing georegistration accuracy and the method of Graham and Luebbe (1981) that uses an independent set of GRPs to compute BIAS and SD values. Table 6 presents a correlation analysis to answer this question. The "N" is the number of observations, the "M" is the slope and the "b" is the intercept of the regression equation, and "r" represents the correlation coefficient which varies between 1 and -1. The fact that the correlations are not statistically different at the 5 percent level of significance suggests that the RMS value and BIAS and SD measurements are quantifying different concepts. One would expect this result from theory, but many Landsat practitioners falsely utilize the RMS value as a measurement of how accurate the scene-to-map registration process is. The georegistration accuracy needs to be measured independently and the procedure of Graham and Luebbe (1981) is one approach.

REFERENCES

1) Clark, P.J. and F.C. Evans. 1954. Distance to nearest neighbor as a measure of spatial relationships in populations. Ecology 35: 445-453.

2) Colvocoresses, A.P. and R.B. McEwen. 1973. EROS Cartographic progress. Photogramm. Eng. 39: 1303-1309.

3) Colwell, J., G. David, and F. Thompson. 1980. Detection and measurement of changes in the production and quality of renewable resources. Environ. Research Inst. Michigan, Report No. 145300-4-F: 73-85.

4) Dowman, I.J. and M.A. Mohamed. 1981. Photogrammetric applications of Landsat MSS imagery. Int. J. Remote Sensing 2: 105-113.

5) Graham, M.H. and R. Luebbe. 1981. An evaluation of MSS P-format data registration. NSTL/ERL-197; AgRISTARS Report No. DC-Y1-04069.

6) Graham, M.H., R.W. Pearson, B.R. Seyfarth, B.G. Junkin and M.T. Kalcic. 1980. ELAS: Earth Resources Laboratory Applicatons Software. NSTL/ERL Report No. 183: 447 p.

7) Nelson, R. and G. Grebowsky. 1982. Evaluation of temporal registration of Landsat scenes. Int. J. Remote Sensing 3:45-50.

8) Sawada, N., M. Kidode, H. Shinoda, H. Asada, M. Iwanaga, S. Watanabe, K-I Mori, and M. Akiyama. 1981. An analytic correction method for satellite MSS geometric distortions. Photogrammetric Engineering and Remote Sensing 47: 1195-1203.

9) Steiner, D. and M.E. Kirby. 1977. Geometrical referencing of Landsat images by affine transformation and overlaying of map data. Photogrammetria 33:41-75.

10) Wong, K.W. 1975. Geometric and cartographic accuracy of ERTS-1 imagery. Photogramm. Engng. Remote Sensing 41: 621-635.

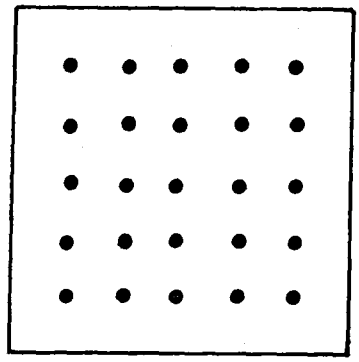
11) U.S. Geological Survey. 1980a. Map projections for Landsat data. U.S. Department of Interior; EROS Data Center; Sioux Falls, S.D; Landsat Data Users Notes No. 10:1-3.

12) U.S. Geological Survey. 1980b. The Hotine Oblique Mercator projection. U.S. Department of Interior; EROS Data Center; Sioux Falls, S.D.; Landsat Data Users Notes No. 11:4-5.

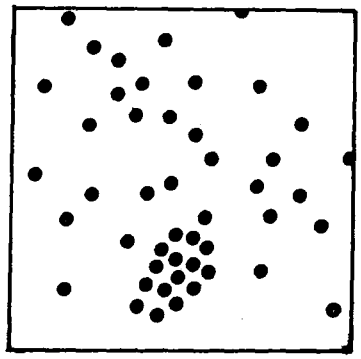
13) U.S. Geological Survey. 1980c. The Space Oblique Mercator projection. U.S. Department of Interior; EROS Data Center; Sioux Falls, S.D.; Landsat Data User Note 12:2-3.

14) U.S. Geological Survey. 1979a. EDIPS image accuracy tests. U.S. Department of Interior; EROS Data Center; Sioux Falls, S.D.; Landsat Data User Note 9:4.

A. REGULAR DISTRIBUTION OF POINTS



B. RANDOM DISTRIBUTION OF POINTS



C. CLUSTERED DISTRIBUTION OF POINTS

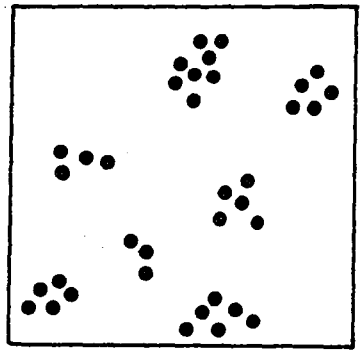


Figure 1. MAJOR CATEGORIES OF GROUND CONTROL POINT DISTRIBUTION

Table 1. P-Format Georegistration Accuracy Assessment.

DATA SET	LANDSAT MISSION NO	ASSESSMENT NUMBER	DATE GEN BY MDP	RBIAS	RSD	CBIAS	CSD
1	2	5	7/23/79	0.5	1.1	-0.3	1.3
2	2	4	7/29/79	0.9	2.4	0.1	1.1
3	2	1	8/30/79	0.2	1.3	-0.2	1.0
4	3	2	4/23/80	15.8	3.9	0.6	1.7
5	2	1	5/18/80	-414.8	5.3	9.2	0.9
6 (KS)	2	0	5/18/80	-407.4	4.2	9.5	1.0
7	2	3	5/12/79	0.7	1.1	1.4	1.0
8	3	2	6/04/79	1.3	1.1	-0.8	1.2
9	3	2	9/15/80	0.3	1.1	-0.8	1.2
10	3	3	2/15/80	-3.6	1.7	3.2	1.6
11	2	2	8/05/79	2.1	1.5	0.2	2.5
12	2	4	5/28/80	10.5	2.3	9.0	1.3
LA/MS	2	3	11/21/81	-219.4	220.8	-95.6	48.8
KS	2	2	11/09/81	251.8	226.7	100.3	40.9

Table 2. P-Format Study of Whole Landsat Scene

Location - GCPs Used	RMS	RBIAS	RSD	CBIAS	CSD	GRP
La. - 8	90	-0.57	0.06	0.49	0.11	184
La. - 16	83	-0.36	0.06	0.63	0.11	176
La. - 24	86	-0.26	0.06	0.60	0.11	168
La. - 32	97	-0.09	0.07	0.44	0.11	160
KS - 8	66	0.10	0.06	-0.79	0.12	145
KS - 16	92	0.04	0.06	-0.33	0.13	137
KS - 24	101	0.11	0.07	-0.18	0.13	129
KS - 32	96	0.09	0.07	-0.15	0.14	121

Table 3. A-Format Study of Whole Landsat Scene

Location - GCPs Used	RMS	RBIAS	RSD	CBIAS	CSD	GRP
La. - 8	65	-0.13	0.06	0.01	0.05	351
La. - 16	73	0.06	0.05	-0.09	0.05	343
La. - 24	76	0.21	0.05	-0.16	0.05	335
La. - 32	71	0.17	0.06	-0.14	0.05	327
KS - 8	45	-0.02	0.05	-0.20	0.05	348
KS - 16	41	0.01	0.05	-0.09	0.05	340
KS - 24	46	0.06	0.05	-0.10	0.05	332
KS - 32	51	0.03	0.05	-0.07	0.05	324

Table 4. Influence of Spatial Distribution of Ground Control Points on Georegistration Accuracy

Location and Type	RMS[1]	RBIAS[2]	RSD[2]	CBIAS[2]	CSD[2]
KS - Random	37.8	0.29	0.43	0.35	0.29
KS - Regular	49.6	0.32	0.39	0.36	0.25
KS - Clustered	46.4	0.88	0.49	0.49	0.49
LA - Random	48.0	0.15	0.24	0.29	0.20
LA - Regular	38.6	0.64	0.27	0.75	0.36
LA - Clustered	46.8	0.80	0.34	0.83	0.30

[1]meters

[2]pixels

NOTE: Based on 20% of a Landsat Scene of A-Format data, 8 GCPs, and 5 Replicates; none of the above numbers are statistically different at the 10% level of significance.

Table 5. Influence of the Spatial Distribution and Number of Ground Control Points on Georegistration Accuracy

Number and Type	RMS[1]	RBIAS[2]	RSD[2]	CBIAS[2]	CSD[2]
8 - Random	52.6	0.39	0.24	0.17*	0.27
8 - Clustered	44.7	0.85	0.41	0.78*	0.42
16 - Random	55.0	0.28	0.27	0.20	0.25
16 - Clustered	49.0	0.38	0.32	0.39	0.37

[1] meters

[2] pixels

* Statistically different at 10% level of significance

NOTE: Based on 20% of a Landsat Scene of A-Format Data and 7 Replicates (Louisiana and Kansas).

Table 6. Correlation Analysis: RMS vs. Absolute Value of BIAS and SD

Parameter	N	M	b	r	Statistical Significance P = 0.05
RBIAS	19	0.0047	0.0572	0.434	N.S.
RSD	19	-0.0020	0.3695	-0.288	N.S.
CBIAS	19	0.0041	-0.0968	0.406	N.S.
CSD	19	-0.0016	0.3231	-0.224	N.S.

NOTE: Based on a random distribution of points and 8 GCPs.

RELATING SPATIAL PATTERNS IN IMAGE DATA TO SCENE CHARACTERISTICS

Alan H. Strahler
Curtis E. Woodcock

Department of Geology and Geography
Hunter College

ABSTRACT

In remote sensing, the primary goal is accurate scene inference, in which characteristics of the scene are inferred from the image data. More effective inference of scene characteristics can be accomplished through the use of techniques that use explicit models of spatial pattern. Spatial patterns in image data are functionally related to the size and spacing of elements in the scene and to the spatial resolution of the image data. At resolutions where variance is high, scene inference techniques should rely heavily on data from the spatial domain. As variance decreases, effective scene inference will increasingly rely on spectral data.

INTRODUCTION

Central to the field of remote sensing is the problem of scene inference, in which the characteristics of the scene are inferred from the image data. Past attempts at scene inference have been dominated by spectral pattern recognition. However, remotely sensed measurements are typically arrayed in a systematic fashion corresponding to the areas on the ground from which the measurements were made. Thus, spatial data are also available for use in scene inference.

This paper presents the results of the analysis of spatial patterns in image data by two methods for three environments. The results enhance our understanding of the relationship between spatial pattern in image data and the characteristics of the ground scene. However, these results should be viewed as intermediate in nature, because they are only one step in the larger process of developing improved methods of using spatial data in scene inference. To understand the role spatial data plays in scene inference, a conceptual model of the remote problem is necessary.

This paper serves as the final report for the first year of NASA Contract 9-16664, Subcontract L200080, which is part of the NASA Fundamental Research Program on Mathematical Pattern Recognition and Image Analysis. In addition, this paper was presented at the 17th International Symposium on Remote Sensing of the Environment in Ann Arbor, Michigan in May 1983.

A Remote Sensing Model

A remote sensor can be defined as a device which measures the intensity of electromagnetic radiation. Associated with a sensor is a resolution cell (or pixel), defined as the size and shape of the areas in the field of view over which the electromagnetic signal strength is integrated. The response time of the sensor is the time over which the received signal is integrated. Also associated with a sensor is a response function describing the integration over wavelengths in the electromagnetic spectrum, and a point spread function defining the integration over the field of view of the sensor. A measurement is the output of a sensor response to the above integrations. A scene is defined as the spatial and temporal distribution of matter and energy fluxes from which the sensor can draw measurements. An image is a collection of measurements from a sensor that are arrayed in a systematic fashion. In the context of this paper, spatial patterns refer to the spatial arrangement of measurements in an image.

The measurements produced by a sensor can be seen as a function of the spatial and temporal distribution of energy and matter in the scene, the characteristics of the sensor, and the scattering and absorption that occurs in the atmosphere between the scene and the sensor. A remote sensing model, then, consists of three components: a scene model that specifies the form and nature of the energy and matter within the scene and their spatial and temporal order; an atmospheric model that describes

the interaction between the atmosphere and the energy emitted by the scene; and a sensor model that describes the behavior of the sensor in responding to the energy fluxes incident upon it and in producing the measurements that constitute the image.

In general, the remote sensing problem can be presented as inferring the order in the properties and distributions of matter and energy in the scene from the set of measurements comprising the image. Whether explicit or not, scene inference always inplies the application of a remote sensing model. in that assumptions must always be made concerning the ground scene, atmosphere, and sensor. The problem of scene inference, then, becomes a problem of model inversion in which the order in the scene is reconstructed from the image and remote sensing model.

The characterization of spatial patterns in image data is intended to provide an improved understanding of scene models. However, an important implication of this work concerns the relation between the size of the elements in the scene and the size of the resolution cells in the image. This fundamental property of the sensor system has important implications in the characterization of spatial pattern in image data and the inversion of the remote sensing model for scene inference.

Scene Components

In specifying scene models, it is necessary to define the entities or objects in the scene that are to be considered. These entities are actually an abstraction of a class of real objects in the scene, and

and they will be referred to as elements. In this context, elements are regarded as having uniform properties or parameters. These properties may be fundamental and invariant, or they may be stochastic in nature -- i.e., characterized by distributions. The elements in a scene can vary widely according to the interests of the interpreter. Several examples of scene elements are; leaf, branch, plant, crop row, tree, field, stand; lawn, house, car, street, garden, housing development; airplane, building, runway, truck, airport. In addition to these elements, which are essentially discrete entities, a particular type of element, the background, should be recognized. The background is usually assumed to be spatially continuous with uniform properties and parameters and is typically obscured partially by other elements in the scene. Soil, rock, snow, and vegetative understory are examples of background elements. For the purpose of this paper, geographic distributions refer to the spatial arrangements of elements in a scene.

Current Use of Spatial Scene Models in Scene Inference

In all attempts at scene inference, assumptions must be made about the scene, sensor, and atmospheric models. For scene models, these assumptions can be either defaulted to nonspatial forms, or include implicit or explicit models of the geographic distribution of elements in the scene. Most remote sensing models default to nonspatial forms in which individual measurements are processed independently of their location in the image and the characteristics of their neighbors.

Conventional supervised and unsupervised techniques both default to such nonspatial forms. Another group of remote sensing models with nonspatial scene models are the proportion estimation, or mixture models. Most of these models estimate the mixture of elements within individual pixels {11, 5, 1}, but the CLASSY algorithm {10}, estimates proportions of unknown elements for the entire image.

Some remote sensing models, such as BLOB {7}, ECHO {8}, and AMOEBA {2}, implicitly assume isotropic high spatial autocorrelation in the scene model. In these approaches, empirically derived constraints are used to enhance the likelihood that adjacent pixels are classified the same. These approaches are most effective in agricultural areas, where the assumption of high spatial autocorrelation is valid. However, to date there has been no attempt to determine the validity of this simple spatial model for other environments except through application of the model and evaluation of the results.

Haralick's sloped facet model {4}, explicitly states the nature of the spatial pattern in the image data. This model allows for linear deviation in brightness values with distance, hense the sloped nature of the facets. Again, there has been no attempt to determine the validity of that model for various combinations of scene elements and resolution cell sizes. Another remote sensing model with an explicit spatial model is the invertible coniferous forest canopy reflectance model of Strahler and Li {13}. The model requires the assumption of multiple trees per resolution cell for inversion. A Neyman Type A model of the spatial

distribution of trees is the explicit spatial model used in the inversion process.

One group of remote sensing models use measures of image texture as the basis of scene inference. Haralick {3}, provides an excellent review of the various approaches used in remote sensing as well as other applications which use image processing. In general, these texture-based approaches have implicit spatial models, and in some ways are similar to unsupervised classification. In both approaches, groups of pixels derived from the image data (on the basis of either spatial or spectral patterns) are related a posteriori to the elements in the ground scene. In these approaches, no attempt is made to understand the geographic processes in the scene that created the spatial patterns in the image data. In this respect, all work relying on image texture has been empirical.

METHODS

Whenever remotely sensed data consist of images, an important new information component is added to the measurement output by the sensor -- its spatial position. Since the position of the measurement in the image is usually a quantifiable function of the position in the scene of the resolution cell from which it is derived, each measurement can be associated with a ground location and be positioned relative to other measurements. From a statistical viewpoint, the sensor's response then becomes a regionalized variable -- a random variable whose position

in time or space is known. (Due to sensor imperfections, individual measurements may not in reality be entirely independent of their neighbors. However, from the theoretical viewpoint presented here, each measurement is considered an independent observation.)

Assume that $Y(\underline{x})$ is a regionalized random variable associated with location $\underline{x}$. As an example, a digital image can be regarded as a single realization of the variables $Y(\underline{x}_i)$, where the set of $\underline{x}_i$, i=1,...,n. correspond to the n resolution cells in the image. If the $Y(\underline{x}_i)$ are uncorrelated, then the image will consist of random noise. If, however, the $Y(\underline{x}_i)$, are in some way related, then the data will exhibit spatial structure. Perhaps the weakest assumption one can made about this structure is what Matheron {6}, refers to as the "intrinsic" hypothesis -- that the increments $Y(\underline{x}_i + \underline{h})-Y(\underline{x}_i)$ associated with a small distance $\underline{h}$ are weakly stationary. Under this assumption, the first moment of the increment, its expected value, is constant or at least only slowly varying with spatial position $\underline{x}$; and the second moment is also invariant with spatial position.

The second moment,

$$2\gamma(\underline{h}) = E \{Y(\underline{x}_i + \underline{h}) - Y(\underline{x}_i)\}^2,$$

is referred to as the variogram; $\gamma(\underline{h})$ becomes the semivariogram {6}.

Just as the variance characterizes the distribution of a nonspatial random variable. Geostatisticians have used the variogram as a primary tool to measure the zone of influence of each $Y(\underline{x}_i)$ on the next, indicate intermeshed structures, reveal anisotrophy, and detect spatial

discontinuities {6}. The one dimensional case is presented for simplicity, but this approach is easily generalized to the multi-dimensional case by considering h to be a vector.

A VICAR (Video Image Communication and Retrieval System) program VRIOGRM was written to calculate a two-dimensional variogram for image data. Ideally, a variogram should be computed using each pixel as a center or target point, against which all other pixels in the image are compared. Since remotely sensed images tend to be large, this approach is computationally unrealistic, and constraints need to be imposed. One constraint concerns the distance h over which the variogram is to be measured. This distance can be thought of as a "window size" when using image data and needs to be larger than the zone of influence and large enough for any periodicities in the data to be revealed. Since VRIOGRM produces a square variogram, $(2h + 1)^2$ pixels are compared with any center point in the image.

The second constraint concerns the selection of points in the image to be used as centers of windows. In VRIOGRM, the number of pixels in the image to be used as a center point in the calculation of the variogram is specified as a parameter. The actual locations to be used in the image are determined randomly. When the locations in the image used as center points is a sample of the entire image, it should be noted that the resulting variogram must be considered an estimate of the true variogram. The variograms shown in this paper are displayed as contour plots of bivariate histograms.

A second method used to measure spatial pattern in image data was that of graphs of local variance as a function of spatial resolution. Calculation of these graphs is accomplished by measuring local variance in the image data, degrading the imagery to successively coarser resolutions, and then measuring local variance at each new resolution. The graphs provide insight into the size and nature of elements in the scene, and can be used to help define the elements that should be used in scene inference. At a time when remotely sensed data is becoming available at continually decreasing spatial resolutions, these graphs should prove invaluable in helping understand how spatial patterns will vary for given environments as a function of spatial resolution.

For this work, local variance is measured for any image as the mean value of a texture image created by the VICAR program PIXSTAT. In this program, the standard deviation of a 3 x 3 moving window of pixels is computed, scaled, and placed in the location of the center pixel. Thus, for each window a value is produced that indicates the local tonal variance, and the mean value for the entire image serves as a reasonable measure of the overall local variance.

The algorithm that has been used to degrade the imagery to successively coarser resolutions, simply averages resolution cells to be combined into a single larger resolution cell. This approach implies an idealized square wave response on the part of the sensor and is limited to degradation at integer multiples. Although point spread functions obviously differ significantly from an idealized square wave response, the point at issue

here is the scene model, not the sensor model. Adopting such a simple sensor model avoids needless complexity at this stage of the research.

The imagery used for the analysis of spatial pattern was digitally scanned from color aerial transparencies using a microdensitometer, thus allowing the analysis of spatial pattern at finer resolutions than are available from conventional spaceborne sensors. Three images were scanned at different resolutions: a forest scene in South Dakota where individual pixels are 0.75m on a side; a forest scene in Colorado with pixels 1.5m on a side; and an agricultural scene with pixels 0.15m on a side.

RESULTS

South Dakota Forest Image

Figure 2 shows the graph of local variance as a function of spatial resolution for the South Dakota forest image. Local variance is low at the resolution that the photo was scanned, or 0.75m (Figure 1A). At this resolution, if a pixel falls on a tree, its immediate neighbors are also likely to be on the tree, since many pixels comprise individual trees. In this situation, the pixels in a 3 x 3 window are likely to have similar DNs and the local variance will be low. Similarly, if a pixel lies on the background, its neighbors are also likely to be on the background, and local variance will again be low. Naturally, some pixels will fall along the borders of the trees or background, and as a result will have high local variance, but the mean local variance for the image

will still be low.

As the size of individual resolution cells increase, the number of pixels comprising an individual tree decreases, and the likelihood that surrounding pixels will be similar decreases (Figure1B). In this situation, local variance increases. This trend continues until a peak in local variance is observed at approximately the size of individual tree crowns, or 6m. At this resolution (Figure 1C), the pattern becomes very mottled as individual pixels tend to be alternatively either on a tree or on the background, and the local variance is very high. As the resolution increases past this peak, local variance decreases. This decrease is associated with individual pixels being increasingly characterized by a mixture of both trees and background. As this mixing of elements occurs, all pixels being to look similar and the local variance continues to decrease (Figure 1D - 1G).

There is considerable structure in the contour plot of the variogram of the South Dakota forest image (Figure 3). The strength of the relationship between a given pixel and its surroundings tend to decrease with distance until it reaches a plateau at about the eighth contour line. At this distance, the relationship between pixels is essentially as if they were selected at random. Ideally, this portion of the contour plot should be flat, but it appears to have local peaks and valleys. This effect may be attributed to the fact that the contour plot is derived from an estimated variogram. With increased sampling, this mottled appearance may be reduced or even disappear.

Another notable feature of the variogram is its anisotropy, which is directly attributable to the shadowing related to the direction of illumination (Figure 1A). The variogram is markedly elongated along an axis approximately diagonal from the upper right corner to the lower left corner. Since shadows look more like the trees than the background, the shadow of a tree tends to reduce the variance measured in the direction of the shadow.

Colorado Forest Image

A picture of the area in Colorado digitally scanned from an aerial transparency for analysis of spatial pattern is shown in Figure 4. The photo was scanned at a ground resolution corresponding to 1.5m on a side. The graph of local variance as a function of spatial resolution (Figure 5) has the same basic structure as was observed for the South Dakota forest image. The local variance begins relatively low, as individual trees are multipixel elements, peaks at approximately the size of an individual tree, and then decreases as resolution size increases. Interestingly, local variance peaks at approximately 9.0m in this image (as opposed to 6.0m in the South Dakota forest image); this effect is attributable to the larger tree crown diameters found in the Colorado frame.

The structure of the variogram for the Colorado forest image (Figure 6) is again similar to the variogram of the South Dakota forest image. Variance is observed to increase with distance until it eventually reaches a plateau. The zone of influence, or distance from the center to the

plateau, is larger in the Colorado forest image, as would be expected due to the larger trees in the area. This difference in variograms is not obvious because the abscissa records the number of resolution cells rather than a direct measure of distance. Since the Colorado forest image data has resolution cells twice the size (on a side) as the South Dakota Forest image, its zone of influence is larger than it appears on the graph. As noted with the South Dakota forest image, anisotropy in the variogram is directly attributable to the direction of illumination.

While the results of the Colorado forest image data are quite similar to those for the South Dakota forest image, they serve the useful purpose of substantiating the interpretation of the results from these methods of the analysis of spatial pattern. Due to the highly experimental nature of these methods, it is reassuring to find their results consistently attributable to the characteristics of the two different scenes. Another factor that may be important for future analysis is that the Colorado forest image contains considerable variability in canopy density. It will be interesting to see how the variogram of this area changes when computed only in areas with certain densities of trees are included. These tests may allow for an improved understanding of the sensitivity of variograms to changes in scene characteristics.

Agricultural Image

A picture of the agricultural area digitally scanned from an aerial transparency for analysis of spatial pattern is shown in Figure 7. The original resolution of the digital data is 0.15m on a side, and was scanned at such a fine resolution in an attempt to analyze spatial structure within fields. Traditionally, the remote sensing community has viewed agricultural fields as homogeneous elements, largely due to the spatial resolution of the available data. However, as spatial resolution decreases on future sensors, more spatial structure within agricultural fields will be resolvable.

The graph of local variance as a function of spatial resolution for the agricultural image does not show the same structure as the graphs for the forest images, in that there is no initial low local variance (Figure 9). It was initially hypothesized that at very fine spatial resolutions, agricultural images would exhibit a similar pattern in local variance as was found in the forest images. In an agricultural setting, individual plants or crop rows would be multipixel elements, and local variance would be low. At the resolution approximately the width of the crop rows, the local variance would peak, and begin its familiar decline. However, Figure 9 shows that local variance simply decreases as a function of spatial resolution in the image data.

One reason that the initial low local variance did not occur is that the spatial resolution of the data was not fine enough to detect the

homogeneity of the crop row as an element in the scene. The distance between crop rows is approximately 5 resolution cells at the resolution of 0.15m that the data was originally scanned. In those five pixels are included the well illuminated portion of a crop row, the shaded side of the crop row, and the space between the rows. As a result, very few 3 x 3 windows in the image will have low variance. If resolution were considerably reduced, variance within both the shaded and well illuminated portions of a single crop row would be low. However, for this affect to be observed, a spatial resolution on the order of 5 cm would be required for this image. Another factor that may be contributing to the lack of initial low variance is that the crop is in a mature stage, and the crop rows have grown close together. Thus, there is not a well developed background signal between rows, against which the crop rows would be highly constrasting.

Variograms were computed for two of the agricultural fields in the image and the entire agricultural image as a whole. These variograms exhibit considerable structure related to the orientation and spacing of the rows. Figure 8A shows the variogram of the field in the upper left portion of the agricultural image (Figure 7). From the variogram it is easy to determine both the direction of the rows, and their spacing. The crop rows are oriented horizontally in this portion of the image, as can be seen by the low variance associated with horizontal movement in the image. Variance changes sharply with movement across the rows, with variance increasing up to one half of the distance between rows.

From that point, variance decreases, until a minimum is reached at the distance between rows. This cycle of high variance at the half width and low variance at even multiples of the distance between rows is repeated all the way to the edges of the variogram, and would continue if the variogram had been calculated for a larger window size. The distance between rows can be determined by counting the number of pixels between the ridges or valleys in the variogram.

A physical explanation of the periodicity in the variogram is as follows. Regardless of where the starting point is relative to a crop row, if you move in the direction perpendicular to the rows the distance of one crop row, you are likely to be in the same position relative to a crop row. In this situation, since the pixels are positioned similar DNs and the resulting variance will be low. Conversely, if you move one half the distance between crop rows, the new location will be very different relative to a crop tow, and thus the difference in DNs of the pixels and the resulting variance will be large.

For the field in the lower left portion of the image, the variogram (Figure 8B) exhibits similar structure as the previous variogram except the row direction is rotated 90 degrees. The same pattern of ridges and valleys occurs at the same spacing between rows. The pattern in the variogram for the entire agricultural image (Figure 8C) is easier to understand after looking at the variograms for the individual fields. The variogram for the entire image simply superimposes the variograms

from fields with rows in perpendicular directions.

DISCUSSION AND CONCLUSIONS

The results of this study indicate that measures of spatial pattern in image data can be related to the characteristics of the elements in the scene. The results of this analysis of spatial pattern should be viewed as a first step in understanding the relationship between scene models and spatial patterns in image data, and the eventual use of spatial data in scene inference. However, based on the results presented, some generalizations about the use of spatial data in scene inference can be made.

The graphs of local variance as a function of spatial resolution give an indication of spatial resolutions where the use of spatial data will be important, as a function of the elements in the scene. At spatial resolutions where local variance is low the information in the spatial domain is low, and scene inference based solely on spectral data may be appropriate. However, at spatial resolutions where local variance is high, the use of spatial data becomes more important, as the use of only spectral data is likely to yield poor results. These graphs also demonstrate that local variance changes as a function of the scene characteristics for a given spatial resolution. For example, at spatial resolutions in the 20-30m range (where data from new sensor systems will soon be available), the forest images begin to exhibit higher spatial variability. However,

in the agricultural image, the local variance is still quite low at those resolutions. These results indicate that the use of spatial data in scene inference will be more important in forested scenes than agricultural scenes when using data from these new sensors.

The graphs of local variance as a function of spatial resolution can be useful in helping to define the elements in a scene and the appropriate remote sensing model to be used in scene inference. The graphs for conifer forests presented above may help explain the results of previous studies designed to test the influence of spatial resolution on forest classification accuracy. Sadowski and Sarno {12} and Latty and Hoffer {9} both found that classification accuracies decreased in forested areas as the size of the resolution cells decreased. These decreasing accuracies are almost certainly an artifact of the definition of the elements in the scene and the remote sensing model used.

Starting with large resolution cells, the elements in the scene are defined as forest stands, or areas large enough to be characterized by numerous trees. The classification of forest stands is based on descriptions that generalize the characteristics of trees in stands, and can be thought of as forest types. In this situation, an element (or forest stand) is a mixture of a variety of smaller objects. A simple conceptual model of this mixture is a combination of trees and a homogeneous background. With large resolution cells, individual pixels also will be characterized by a mixture of trees and background, and will generally be representative of the larger forest stand type. However, as the

resolution cell size decreases, individual pixels will be decreasingly characterized by mixtures of trees and backgrounds. Eventually, resolution cells tend to be either in the location of a tree or on the background. At this point, the elements should switch from forest stand types to individual trees. However, in these studies the elements, or targets of classification, remained forest types throughout the study. The decreases in accuracy associated with shrinking cell size may be attributable to the increasing inappropriateness of the remote sensing model used for scene inference. From the point of view of the classifier, pixels are eventually differentiated into tree and background classes, all in areas originally designated as forest types. In one sense, at small resolution sizes the accuracy of the classification could surpass the ability of the techniques available to evaluate it. This situation suggests a restructuring of the question of what accuracy means as spatial resolutions change.

In conclusion, the two methods of measuring spatial patterns in image data reveal useful and different information concerning the characteristics of the elements in the scene. Variograms illustrated the anisotropy in the data attributable to the direction of illumination, found periodicities in the data, and measured the zone of influence of pixels on their surroundings. Variograms are a method of measuring spatial patterns in image data that may be useful in future scene inference techniques that rely more on data from the spatial domain. The second method, graphing local variance as a function of spatial resolution, is most useful

because it readily displays the interaction between scene elements and spatial resolution. Through the use of these graphs, more informed decisions can be made concerning the nature of the scene inference techniques to be used, given the spatial resolution of the data available and the nature of the scene.

References

{1} Adams, J.D., Smith, M., and J.R. Adams (1982). Use of Laboratory Spectra for Determining Vegetation Assemblages in Landsat Images. Int. Symp. on Remote Sensing of Environment, Second Thematic Conference, Remote Sensing for Geologic Exploration, in press.

{2} Bryant, J. (1979). On the Clustering of Multidimensional Pictorial Data. Pattern Recognition, 11:115-125.

{3} Haralick, R.M. (1979). Statistical and Structural Approaches to Texture. Proc. of the IEEE, 67(5):786-804.

{4} Haralick, R.M. (1980). Edge and Region Analysis for Digital Image. Computer Graphics and Image Processing, 12:60-73.

{5} Horowitz, H.M., Lewis, J.T., and A.P. Pentland (1975). Estimating the Proportions of Objects from Multispectral Scanner Data, Environmental Research Institute of Michigan, Final Report N. 109600-13-f, 117p.

{6} Huijbregts, C.J. (1975). Regionalized Variables and Quantitative Analysis of Spatial Data. In J.C. Davis and M.J. McCullash (Eds) Display and Analysis of Spatial Data, Wiley, London, 38-53.

{7} Kauth, R.J., Pentland, A.P., and G.S. Thomas (1977) BLOB an Unsupervised Clustering Approach to Spatial Preprocessing of MSS Imagery, 11th Int. Symp. on Remote Sensing of Environment, 2:1309-1317.

{8} Landgrebe, D.A. (1980). The Development of a Spectral-Spatial Classifier for Earth Observational Data. Pattern Recognition, 12:165-175.

{9} Latty, R.S., and R.M. Hoffer (1981). Computer-Based Classification Accuracy Due to the Spatial Resolution Using Per-Point vs Per-Field Classification Techniques. Machine Processing of Remotely Sensed Data Symposium, 384-392.

{10} Lennington, R.K., and M.E. Rassbach (1978). CLASSY - An Adaptive Maximum Likelihood Clustering Algorithm. Proc. of the 9th Annual Meeting of the Classification Society (North American Branch), Clemson.

{11} Marsh, S.E., Switzer, P., Kowalik, W.S., and R.J.P. Lyon (1980). Resolving the Percentage of Component Terrains within Single Resolution Elements. Photogrammetric Engineering and Remote Sensing, 46(8):1079-1086.

{12} Sadowski, F., and J. Sarno (1976). Forest Classification Accuracy as Influenced by Multispectral Scanner Spatial Resolution. NASA-CR-ERIM 1090600-71-f, 130p.

{13} Strahler, A.H., and X. Li, (1981). An Invertible Forest Canopy Reflectance Model. 15th Int. Symp. on Remote Sensing of Environment.

Figure 1A. South Dakota Forest Image at the original resolution at which it was scanned. Each pixel is 0.75m on a side.

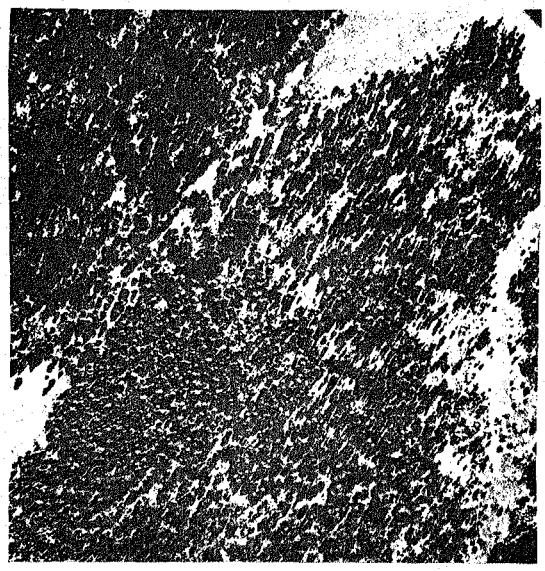


Figure 1B. South Dakota Forest Image after degradation. Each pixel in this image is 3.0m on a side and contains 16 of the original pixels.

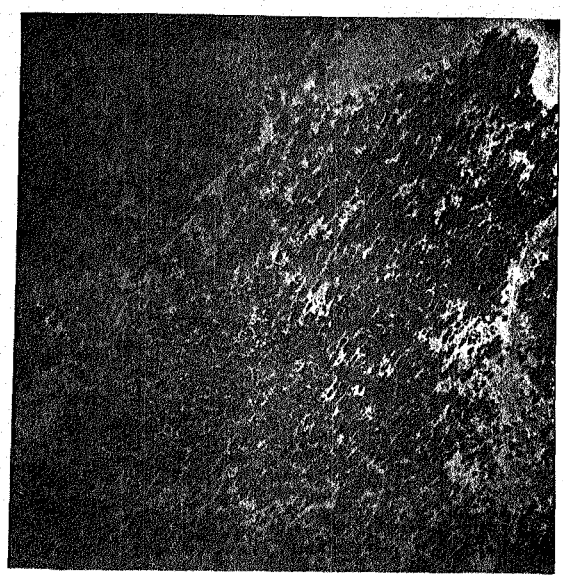


Figure 1C. South Dakota Forest Image after degradation. Each pixel in this image is 6.0m on a side.

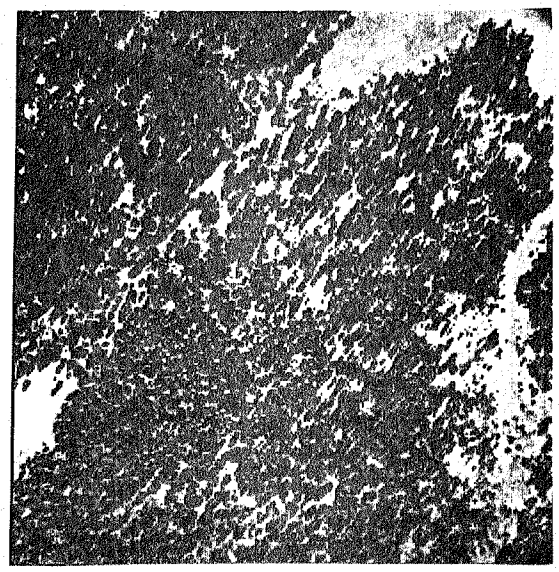


Figure 1D. South Dakota Forest Image after degradation. Each pixel in this image is 9.0m on a side.

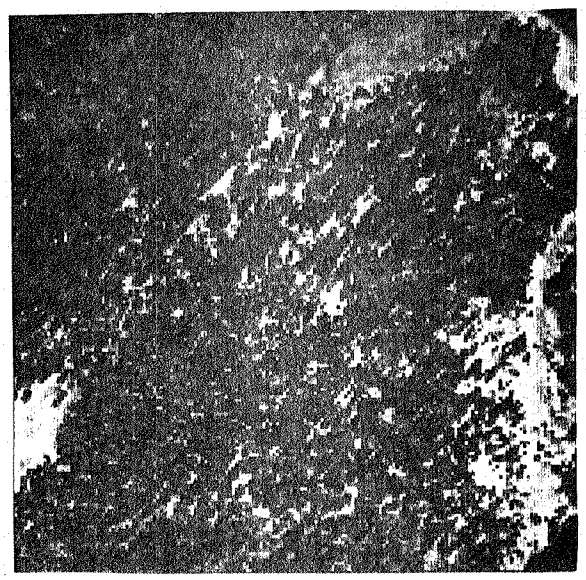


Figure 1E. South Dakota Forest Image after degradation. Each pixel in this image is 12.0m on a side.

Figure 1F. South Dakota Forest Image after degradation. Each pixel in this image is 18.0m on a side.

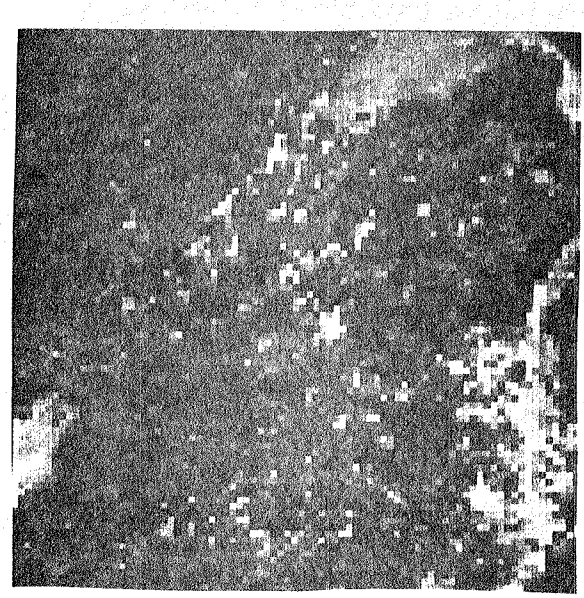


Figure 1G. South Dakota Forest Image after degradation. Each pixel in this image is 24.0m on a side.

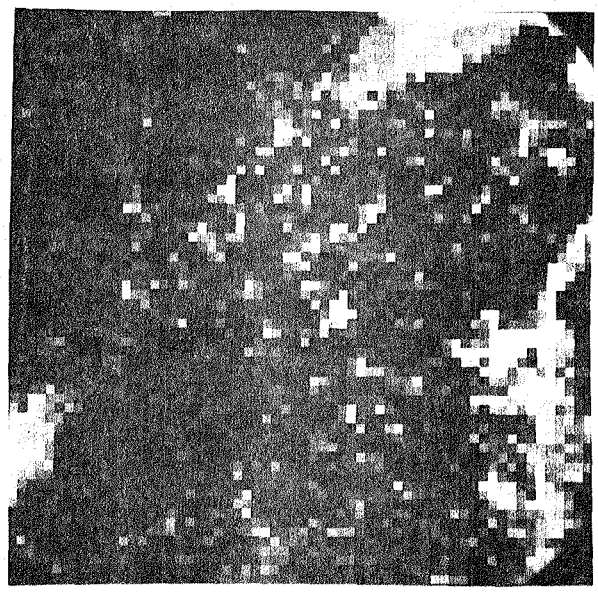


Figure 2. Graph of local variance as a function of spatial resolution for the South Dakota Forest Image data.

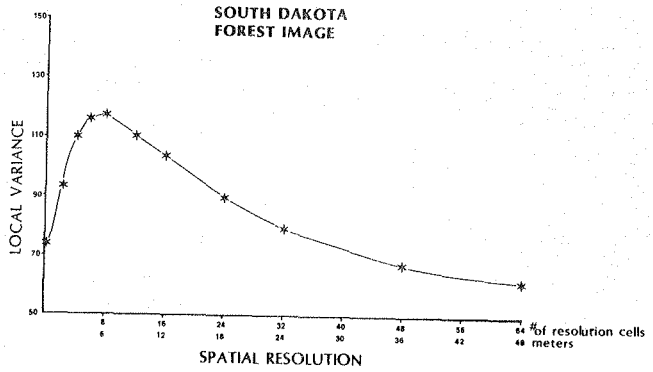


Figure 3. Contour plot of the Two-Dimensional Variogram of the South Dakota Forest Image.

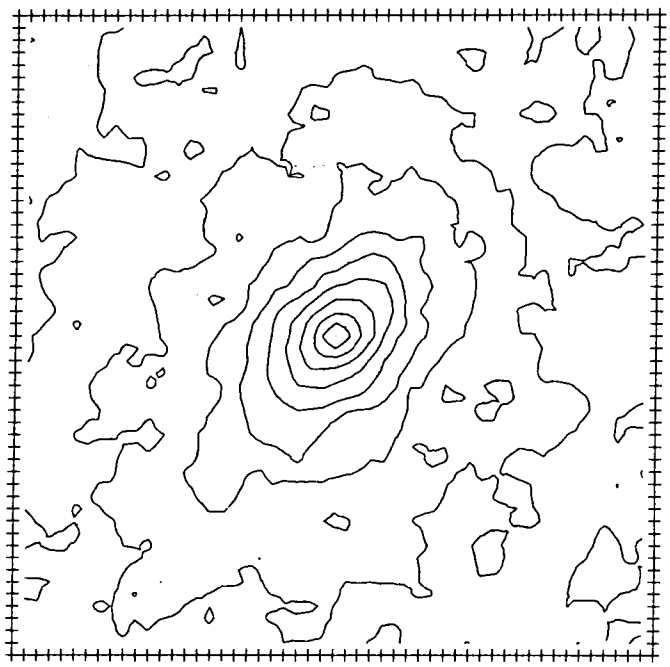


Figure 4. Photograph of the Colorado Forest scene that was digitally scanned for analysis of spatial pattern.

Figure 5. Graph of local variance as a function of spatial resolution for the Colorado Forest Image data.

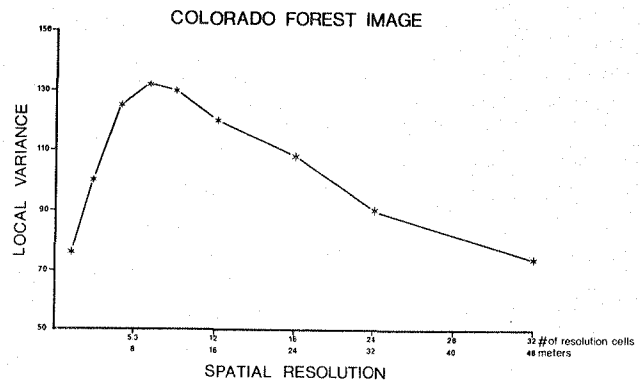


Figure 6. Contour plot of the Two-Dimensional Variogram of the Colorado Forest Image.

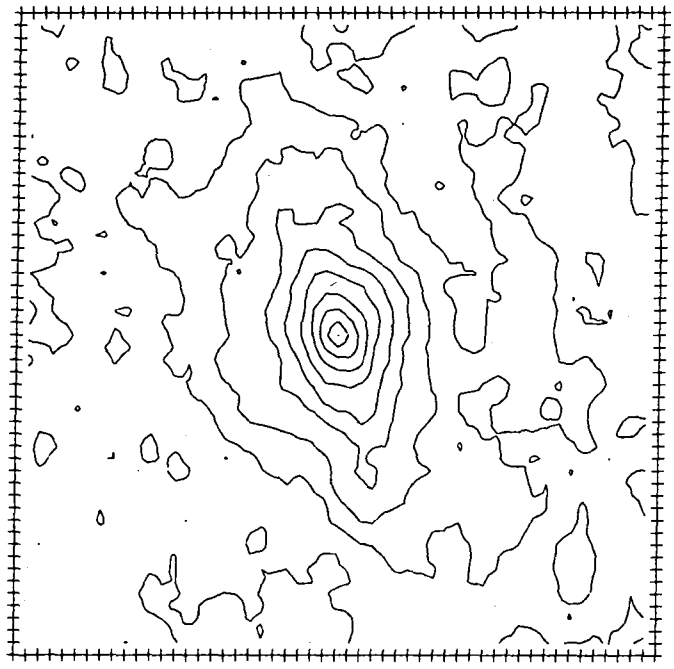


Figure 7. Photograph of the Agricultural scene that was digitally scanned for analysis of spatial pattern.

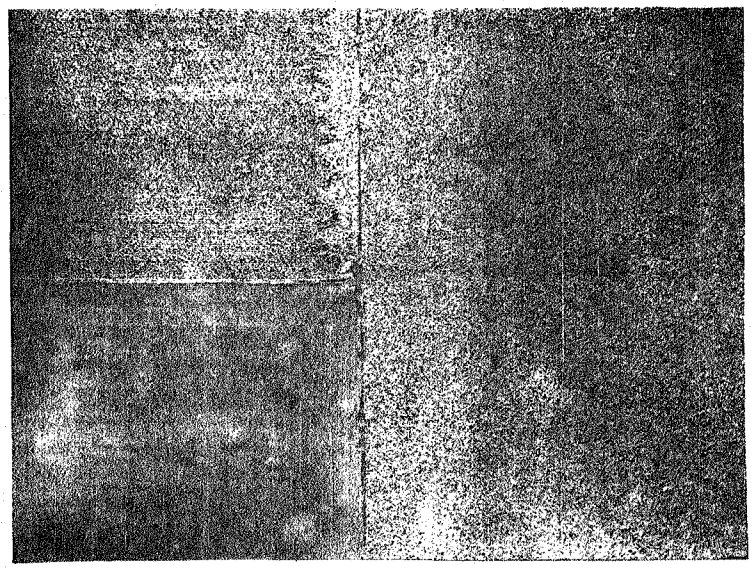


Figure 8A. Contour plot of the Two-Dimensional Variogram of the field in the upper left portion of the Agricultural Image.

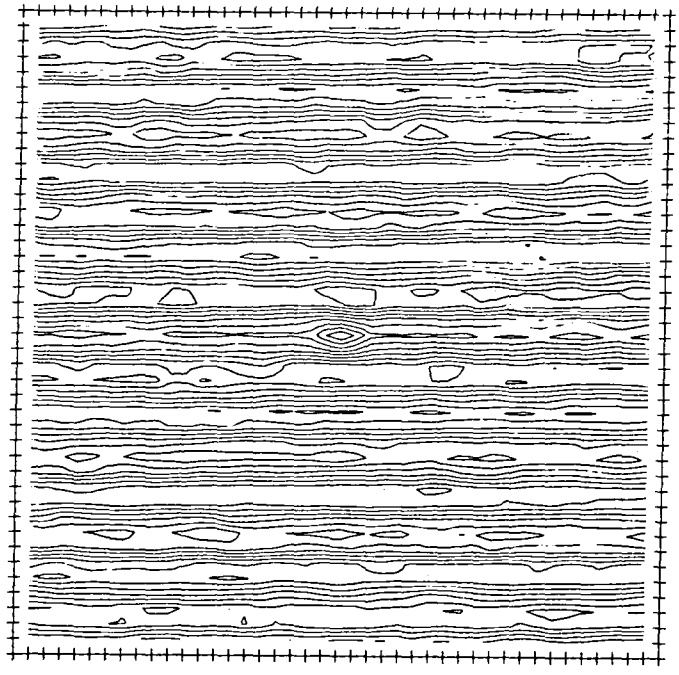


Figure 8B. Contour plot of the Two-Dimensional Variogram of the field in the lower left portion of the Agricultural Image.

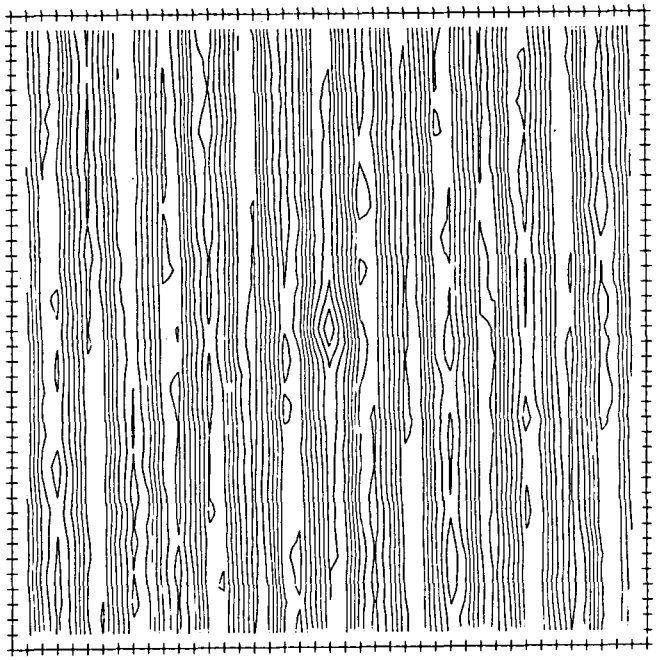


Figure 8C. Contour plot of the Two-Dimensional Variogram of the entire Argicultural Image.

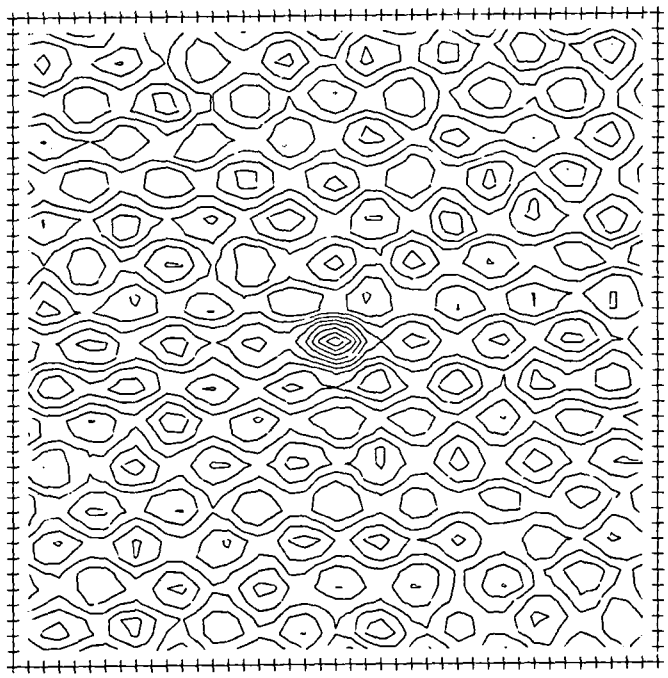
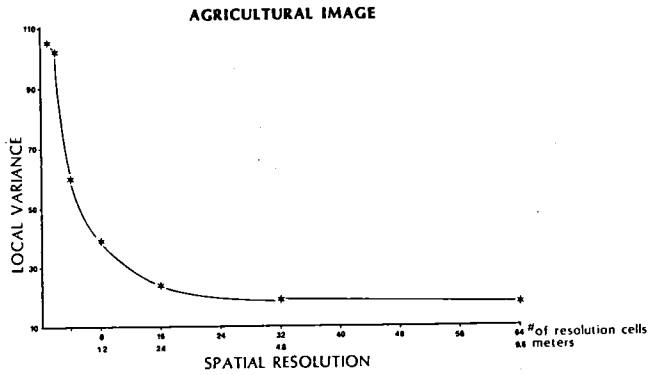


Figure 9. Graph of local variance as a function of spatial resolution for the Agricultural Image.



Page intentionally left blank

SHAPE FROM SHADING: AN ASSESSMENT

Grahame B. Smith

Artificial Intelligence Center, SRI International
Menlo Park, California 94025

Abstract

We review previous efforts to recover surface shape from image irradiance in order to assses what can and cannot be accomplished. We consider the informational requirements and restrictions of these approaches. In dealing with the question of what surface parameters can be recovered locally from image shading, we show that, at most, shading determines relative surface curvature, i.e, the ratio of surface curvature measured in orthogonal image directions. The relationship between relative surface curvature and the second derivatives of image irradiance is independent of other scene parameters, but insufficient to determine surface shape. This result places in perspective the difficulty encountered in previous attempts to recover surface orientation from image shading.

1. Introduction

The determination of land cover from aerial imagery is a task thatphoto interpreters accomplish by using both the image data and their knowledge of the structure of the world. The image data encodes the complex process whereby light is reflected from a surface. The surface shape, the surface albedo, the position of the lighting sources, and the functional form of the reflectance properties of the material are elements of this encoding. The human visual system interprets image data as a 3-D model of the scene, distinguishes among different surface materials, and ascertains the position of the lighting sources. It is difficult to believe that a machine vision system can achieve, say, surface material differentiation without simultaneously being able to recover the surface shape and the other parameters that are needed to explain the detected image intensity. Of course, it may be possible to use special sensors and multiple information sources to make it unnecessary to reconstruct a complete 3-D model of the scene, but it would be surprising if such specialization could retain sufficient generality to be useful over a range of remote sensing tasks, e.g., in both renewable and nonrenewable resources.

The machine vision approach of simultaneously recovering all the parameters necessary to account for image intensity is expressed in the notion of intrinsic images [1] (or the $2\frac{1}{2}$-D sketch [10]). These intrinsic images can be thought of as overlays, each specifying the value of one parameter that goes into the formula for calculating the image intensity. The images are not independent; if one is to be varied, the

The research reported herein was supported by the Defense Advanced Research Projects Agency under Contract MDA903-83-C-0027 and by the National Aeronautics and Space Administration under Contract NASA 9-16664. These contracts are monitored by the U.S. Army Engineer Topographic Laboratory and by the Texas A&M Research Foundation for the Lyndon B. Johnson Space Center.

others must be also — so that the predicted image intensity remains invariant (and equal to the observed value). The notional division of an image into particular intrinsic images would be of little merit unless one believed that estimates of each intrinsic image could be obtained by models that were largely independent of the other intrinsic images. While models have been proposed to recover various intrinsic images, there have been considerable efforts made to recover the scene's 3-D shape,[1] in particular the surface orientation at each image point. These 'shape-from-...' models embody a structure that would allow shape to be recovered principally from a single measure, e.g., texture, contour, or shading. While 'shape-from-...' models are not seen as complete solutions to shape reconstruction, there is an implicit expectation in their title that shape estimates can be calculated from their respective measures. Here we review the work we and others have done towards the goal of recovering surface shape from image shading. Is it attainable — or is it myth?

The importance of shape recovery is clear; if the shape is known, surface albedo, and the other parameters that determine image intensity are obtainable. Land cover differentation is dependent on knowing the [relative] surface albedo, rather than image parameters, such as intensity. If we cannot recover shape, the intrinsic image approach offers little as a model for perception. Shading is only one source of shape information. Edge information is of great importance, but there is little occlusion in aerial images. The ability to recover shape from shading seems more critical in the case of aerial imagery than for most other types of imagery.

We first review three research efforts: those of Horn and his colleagues

[1] We use the expression surface shape to denote both the intrinsic properties of the surface, e.g., cylindrical, and the orientation of the surface in space. Elsewhere, shape is sometimes used to denote only the intrinsic properties of the surface, not its orientation in space.

[6,7,9,14,15,16], Pentland [11], and our own [12,13], — to determine what can and cannot be accomplished, and to consider the informational requirements and limitations of these approaches. We discuss the dilemma of local computation versus global constraint propagation and seek to ascertain what can be computed locally, and how information can be propagated across an image. Finally, we seem to be left with the conclusion that shading, when viewed as a single source of shape information, is an insufficient source for the recovery of surface shape. Shape cannot be obtained from shading alone. However, we are able to characterize the scene information that shading provides.

An alternative approach to recovering shape from shading is model based. Can we determine which model, from a set of models, best describes the image data? This approach is dependent on discovering a small set of easily distinguishable models that adequately describe the surfaces encountered. Industrial inspection, rather than remote sensing of the environment, appears better suited to a model based procedure. In this assessment we do not consider this related, but essentially different approach.

2. Approaches to Shape from Shading

2.1 Horn and Colleagues

A study by Horn [6,7,8] of the relationship among image irradiance,[2] surface shape, surface albedo, and illumination conditions led to formulation of the image

[2]Image irradiance is the light flux per unit area falling on the image, i.e., incident flux density.

irradiance equation, which states that image irradiance is proportional to scene radiance.[3] This is expressed by the equation

$$I = R \quad ,$$

where I is the image irradiance as a function of the image coordinates, and R is the scene radiance as a function of the scene parameters. Of course, this equation relates the image irradiance at an position in the image to the scene radiance at its corresponding scene position. Implicit in this equation is an assumption of orthographic projection. However, such an assumption, to avoid complexity in the mathematical formulation, is a minor restriction and does not detract from the generality of the model.

Image irradiance is a function of the image coordinates x and y, but scene radiance is a function of the illumination strength, its position, the surface albedo, and the surface orientation. For the formulations reviewed here, we find that a number of assumptions are made so that scene radiance can be considered a function of the surface orientation variables only; constant values are used for the illumination strength, its position, and for the surface albedo. That is, shape-from-shading is analyzed for the simplified case of a constant light source and constant surface albedo. The restriction to a constant light source is not only a good approximation of the situation we experience daily (and an excellent approximation for a photograph), but also corresponds to the difficulty confronting the human visual system when this constancy is not met, e.g., under strobe lighting. The assumption of constant albedo is harder to justify, since nature obviously exhibits

[3]Scene radiance is the light flux per unit projected area per unit solid angle emitted from the scene, i.e., emitted flux density per unit solid angle.

variable albedo. Still, when we consider the manner in which facial make-up is used to alter the perceived shape of the face, it may well be that continuous changes in albedo are processed by the human visual system as if they were constant. Notwithstanding the justification for constant albedo, it is unlikely that shape-from-shading can be solved for the case of variable albedo if it cannot be solved for constant albedo. Such a restriction is in effect a case analysis to determine if shading provides sufficient shape information in a less-than-general model.

In the formulations under review, various parameterizations of surface orientation have been used. The two we specify are (i) surface gradients, i.e., the partial derivatives of depth, z, with respect to the scene (and image) coordinates x and y, and (ii) components of the surface normal, i.e., l and m, the x and y components of the surface normal. Using the notation, $p = \frac{\partial z}{\partial x}$, and $q = \frac{\partial z}{\partial y}$, we note the equivalence of the parameterizations

$$p = \frac{-l}{\sqrt{1-l^2-m^2}} \quad , and \quad q = \frac{-m}{\sqrt{1-l^2-m^2}} \quad .$$

The image irradiance equation is usually expressed as

$$I(x,y) = R(p,q) \quad , or \quad I(x,y) = R(l,m) \quad ,$$

and we shall use both forms to express the relationship between image irradiance and scene radiance for the case of constant illumination and constant albedo. As $p = \frac{\partial z}{\partial x}$, and $q = \frac{\partial z}{\partial y}$, we see that the image irradiance equation is a first-order partial differential equation and, if I and R are known, we could (at least in principle) solve the differential equation and recover the depth, z.

To have an explicit form for R, we must have a model for the type of reflection occurring at the scene surfaces. In the work reviewed here the surface is assumed

to be a perfectly uniform diffuse reflector, i.e., the scene radiance is isotropic.[4] While this model is invalid as a description of specular reflection, scene radiance in the natural world, (except for specific situations, such as water surfaces), may be approximated by such a description. The expression for scene radiance in this case is [12]

$$R(l,m) = al + bm + c\sqrt{1 - l^2 - m^2}$$

or, equivalently,

$$R(p,q) = \frac{(-ap - bq + c)}{\sqrt{1 + p^2 + q^2}} \quad ,$$

where a,b, and c are constants expressing illumination strength, its position, and the surface albedo.

The approach taken by Horn and his colleagues [6,7,9,14,15,16] is to solve the first-order partial differential equation,

$$I(x,y) = \frac{(-ap - bq + c)}{\sqrt{1 + p^2 + q^2}} \quad ,$$

assuming that a,b, and c are known — i.e., the surface albedo, and the illumination strength, and its position. While this need to know scene parameters may seem over-restrictive, such information may come from other components of a vision system. The need to know the illumination position does not seem to be a major drawback of this approach, but the requirement that the scene albedo be known is troublesome. If the conceptual model of intrinsic images is to be followed, the inability to decouple surface orientation from surface albedo would seem fundamental. Regardless of this

[4]This situation is also called Lambertian reflectance, after Lambert, who proposed a point reflection model (in which the reflected flux per unit surface area per unit solid angle varied as the cosine of the angle between the surface normal and the viewing direction) to account for the observation that matt surfaces looked equally bright from any viewing position.

difficulty, the question of whether shape can be recovered in a limited domain is basic to the investigation of vision.

Two approaches to solving the image irradiance equation are direct integration [6,7], and iterative/relaxation techniques [9,14,15,16]. The direct integration approach has been carried out generally in those circumstances in which $I(x, y)$ and its derivatives can be determined for all x and y, i.e., for a spatially unquantized, continuous-tone image. The method used is the standard technique of characteristic strips for solving a first-order hyperbolic partial differential equation [6,7]. Starting with a point at which the surface orientation is known, integration moves along a curve in the image. This curve is dictated by the image. Adjacent curves generally are not 'parallel', which makes it difficult to get complete coverage of the image. Interpolation between these curves — or strips, as they are usually called — to find initial values to commence an intervening strip integration, involves complex procedures. As far as digital images are concerned, direct integration would be hard to organize, even if we were first to model the intensities to obtain a continuous form for $I(x, y)$.

As is the case with most partial differential equations, it should be noted that the image irradiance equation has many solutions [4]. The boundary conditions (in the above method the initial values for a strip) are vital in selecting the solution that describes the surface in the image. Should the image irradiance equation be 'underconstrained' in the sense that, for a given $I(x, y)$, it admits solutions that encompass a wide range of surface types with similar boundary values, we might then expect numerical error to defeat attempts at numerical integration. In such cases, errors 'mix in' other solutions that can eventually dominate the recovered

solution, even though they may be excluded by the boundary conditions. The method of direct integration has been demonstrated on simple images [6]. These examples required only a small number of integration steps. Numerical instability has also been reported [4].

The other approach used to solve the image irradiance equation is relaxation. Relaxation procedures avoid numerical instability, but face the problem of convergence. However, they do have the advantage of being directly applicable to digital images, i.e., spatially quantized, discrete-tone images. The relaxation (or iterative) approach views the image irradiance equation not as a differential equation, but as an algebraic constraint. For pixel (i, j),

$$I_{i,j} = R(p_{i,j}, q_{i,j}) \quad ,$$

where $I_{i,j}$ is the image irradiance for the (i, j)th pixel, and $p_{i,j}$,and $q_{i,j}$ specify the surface orientation of the surface patch that is imaged at pixel (i, j). As an algebraic constraint, the image irradiance equation relates image irradiance to the two surface orientation variables, $p_{i,j}$,and $q_{i,j}$. In viewing the image irradiance equation as a algebraic constraint, we lose the interrelationship of $p_{i,j}$, $q_{i,j}$, and their neighboring values, a relationship inherent in the differential equation. To compensate for this loss, an additional constraint must be introduced that relates $p_{i,j}$,and $q_{i,j}$ to their neighboring values. Such a relationship is essential for a relaxation procedure. The relationship usually introduced attempts to capture the notion of surface smoothness [2,9, 12,14]. The particular form of the smoothness constraint may, for example, require that $p_{i,j}$,and $q_{i,j}$ be equal to the mean values of neighboring p's and q's. For any trial values for $p_{i,j}$,and $q_{i,j}$, the constraint

imposed by the image irradiance equation and the constraint resulting from surface smoothness will not be completely satisfied. The residual equation formed from each constraint specifies how well that constraint is satisfied. If $\xi_{i,j}$ is the sum of the [absolute values of the] residuals from both the image irradiance constraint and the surface smoothness constraint for the (i,j)th pixel, then, for trial values of p and q for every image pixel, the total residual error is

$$\xi = \sum_{i,j \in image} \xi_{i,j} \quad .$$

The allocation of surface orientations to all pixels should minimize this total error — that is,

$$\frac{\partial \xi}{\partial p_{i,j}} = 0 \qquad \forall\, i,j \in image \quad ,$$

$$\frac{\partial \xi}{\partial q_{i,j}} = 0 \qquad \forall\, i,j \in image \quad .$$

From these equations we obtain an iterative scheme for updating the values of p and q so that they are compatible with their neighboring values, as well as with the image irradiance equation [9,12,14]. If such a scheme is convergent, we have a procedure for obtaining shape from shading.

It should be noted that the relaxation schemes, that use the foregoing approach are possible only because the smoothness constraint relates the values at one pixel to those of its neighboring pixels. The boundary conditions needed for selecting a particular solution from the solution set of the iterative scheme are propagated by the smoothness constraint, not the image irradiance equation. Compared with the direct-integration approach, information propagation in the relaxation scheme uses a different mechanism. We must remember this when we assess results.

Success with these methods has generally been limited to small images, (usually

fewer than 30 x 30 pixels), of nearly spherical or saddle surfaces [3,9,12,14]. For an effective relaxation scheme, the initial solution should have no effect on the surface recovered. This unfortunately is not the case [12]. Boundary conditions are not propagated more than a few pixels by the smoothness constraints [12,14]. Surface recovery from large images, (bigger than 30 x 30 pixels), is ineffectual for this reason. As a consequence of the fact that smoothness is used as information propagator, assumptions (albeit weak ones) have been made about surface shape. Shading as a constraint, and smoothness as a surface type, appears insufficient to provide a basis for an effective shape-from-shading algorithm.

2.2 Pentland

The approaches to solving the shape-from-shading task that we have discussed so far have all been based on constraint propagation. Direct integration is a spatially serial solution to the propagation problem, while relaxation attempts to achieve this propagation with a temporally serial solution; in other words, relaxation employs local processing, but it must iterate until enough cycles have passed to allow information to propagate spatially. Purely local computation of scene parameters, on the other hand, is not a propagation method. While this kind of computation can use neighboring data — and not just of the nearest neighbors — it must provide an instant solution. It cannot iterate and therefore it does not provide a temporally serial solution. Such an approach to scene parameter computation avoids the numerical instability of direct integration methods, as well as the convergence and propagation problems of relaxation, but it cannot use spatially distant scene information. A local computation can use global information, such as the position

of the light source, but it cannot use scene details, such as the position of a distant edge. Of course, the reason for carrying out purely local computation stems from the hypothesis that such scene detail is not involved in the computation at this level in the visual system. Can shading provide sufficient local information to allow recovery of surface shape by purely local computation? This is the question addressed by Pentland [11].

The inadequacy of local image measurements for specifying surface orientation can be understood by counting the variables needed to specify various image measurements. Let us consider the case of a uniformly diffuse reflecting surface. Image irradiance (1 measurement) is a function of surface orientation (2 parameters), the product of surface albedo and illumination strength (1 parameter), and the position of the light source (2 parameters). The gradients of image irradiance (2 measurements) are functions of the same variables as image irradiance and, additionally, are functions of surface curvature (3 parameters). The second derivatives of image irradiance (3 measurements) are functions of all the variables mentioned above, plus the rates of change of curvature (4 parameters). Because higher image-irradiance derivatives introduce more surface shape derivatives, we have more parameters than measurements. It should be noted that a knowledge of global quantities, such as the illumination position and the product of surface albedo and illumination strength, is not sufficient to allow the surface orientation to be computed locally. If we make assumptions about the relationship among some of the above parameters, we can produce a system of equations from which surface orientation can be calculated.

Pentland investigates the case in which an image patch of a uniformly diffuse

reflecting surface can be considered identical to a point on an illuminated sphere whose reflection is also uniformly diffuse [11]. He calculates the orientation of the surface patch on the sphere that has the same appearance as the surface patch in the image. Not all image patches can be represented by points on an illuminated sphere. Spheres whose reflection is uniformly diffuse have the property

$$\frac{I_{xx}}{I_{yy}} \geq 0 \quad ,$$

where subscripts denote partial differentation with respect to those subscripts. There are surfaces, e.g., a sinusoidal surface, for which $\frac{I_{xx}}{I_{yy}}$ can be negative. The procedure for estimating surface orientation that is based on the assumption that surfaces can be approximated by locally spherical patches is applicable only to parts of an image. Notwithstanding these restrictions, an important aspect of the assumption of local sphericity is that the surface orientation is calculated by using the second derivatives of image irradiance only, i.e.,

$$\frac{1-m^2}{lm} = \frac{I_{xx}}{I_{xy}} \quad ,$$
$$\frac{1-l^2}{lm} = \frac{I_{yy}}{I_{xy}} \quad .$$

These equations are derived by differentiating the image irradiance equation and noting that, for a sphere, $l_x = \frac{1}{r}$, $l_y = 0$, $m_x = 0$, and $m_y = \frac{1}{r}$, where r is the sphere's radius.

In this model, surface orientation is directly dependent on neither image irradiance nor on the first derivatives of image irradiance. It may be estimated even in images that exhibit linear changes in irradiance induced by artifacts, and in images that exhibit constant illumination levels induced by atmospheric effects, such as

backscatter. More importantly, the formulas are independent of the illumination parameters and the surface albedo. In exchange for acceptance of a restrictive assumption with respect to surface type, one acquires not only a means of calculating surface orientation, but a procedure that needs no information other than image measurements — a procedure, in effect, that is matched to the notion of intrinsic images.

Even in those areas of an image to which this approximation can be applied, the assumption that a surface can be approximated by a patch with the same curvature in any direction needs experimental verification. The world is obviously not composed of such surfaces, but it is the difference between the estimated and the actual surface orientation that is more important than the error made in approximating the surface by a spherical patch. Application of the above formula yields qualitative agreement between the estimated and actual shape in synthetic images and in natural images of simple objects [11], (for which $\frac{I_{xx}}{I_{yy}}$ is generally positive). Shape estimates from synthetic images of ellipsoidal surfaces are 'flatter' than the actual shapes. It should be noted that shape estimates, which are integrated surface orientations, often appear 'better' than what might be expected on the basis of the surface orientation error. An algorithm based on approximating a surface patch by a spherical one seems better suited for computing the qualitative shape of a surface than the orientation of surface elements. Such an algorithm is applicable only to thoses image patches that are consistent with the interpretation of such patches as points on a sphere. The conditions necessary for enabling this kind of interpretation have not been fully characterized. Alternative models, that are applicable when an image patch is inconsistent with an interpretation that it is a point on a sphere, are

currently unknown.

In principle, because image irradiance is not differentiable at boundaries, we cannot apply the above method there. However, unlike propagation methods require our knowing boundary positions in order to stop computation, the local-computation approach may accomplish this simply by indicating (through its failure at a boundary) where the boundary is.

Pentland's approach hinges on the local-sphericity assumption. In restricted circumstances he is able to estimate surface orientation directly from the second derivatives of the image irradiance. What other, perhaps less specific, assumptions can be made that allow shape to be estimated locally? Before attempting to answer this, we review the shape-from-shading formulation we have previously proposed [12,13], — first, to assess its performance, then to provide the requisite analytical tools for answering questions about local computation.

2.3 Smith

The approach taken by Horn and his colleagues provides a formulation of the shape-from-shading task that requires knowledge of scene parameters, but places no restriction on the surface shape. Calculation of surface orientation is not a local process, and, if surface orientation is to be recovered, knowledge of boundary conditions is necessary. Pentland, on the other hand, restricts the surface shape but requires no scene parameters, no boundary conditions, and derives surface orientation by purely local computation. Is there an intermediate position? Is there a formulation that neither restricts the surface shape nor requires knowledge of scene parameters? Of course, local computation seems desirable — but is it

worth the concomitant cost of surface type restriction or the requirement that scene parameters be known a priori? The formulation previously described by us, takes such an intermediate position.

For a uniformly diffuse reflecting surface, surface orientation is related to image irradiance by the second-order partial differential equations [12]

$$\alpha\theta l_{xx} + \beta\theta m_{xx} - \alpha\gamma l_{xy} - \beta\gamma m_{xy} = \chi\theta I_{xx} - \chi\gamma I_{xy} \quad ,$$
$$\alpha\theta l_{yy} + \beta\theta m_{yy} - \alpha\delta l_{xy} - \beta\delta m_{xy} = \chi\theta I_{yy} - \chi\delta I_{xy} \quad ,$$

where

$$\alpha = I_x m_y - I_y m_x \quad ,$$
$$\beta = I_y l_x - I_x l_y \quad ,$$
$$\gamma = {l_x}^2(1-m^2) + {m_x}^2(1-l^2) + 2l_x m_x lm \quad ,$$
$$\delta = {l_y}^2(1-m^2) + {m_y}^2(1-l^2) + 2l_y m_y lm \quad ,$$
$$\theta = l_x l_y(1-m^2) + m_x m_y(1-l^2) + (l_x m_y + l_y m_x)lm \quad ,$$
$$\chi = l_x m_y - l_y m_x \quad .$$

These equations are derived from the image irradiance equation. The assumption of uniformly diffuse reflection relates some of the scene parameters, thereby allowing elimination of parameters that specify surface albedo and illumination conditions.

The assumption that surface reflection is uniformly diffuse is an assumption about the physics of image formation. While it does not describe the reflectance properties of all surface, it is a reasonable approximation to most surfaces that are encountered in the natural world. For any formulation of the relationship between shading and shape, some assumptions are necessary. Those describing properties found in nature are more palatable than restrictions for which little a priori evidence is available.

A desirable aspect of this formulation is that surface orientation is not related to image irradiance, but only to its derivatives. The existance of constant illumination

levels, from atmospheric scattering or fogging of photographic images, does not impede the potential for shape recovery. Linear changes in intensity, however, must affect the shape of any recovered surface. A more important aspect of this formulation is its independence of surface albedo. Again we reiterate that, if the notion of intrinsic images is to be useful we must find models that decouple surface shape from surface reflectance. The fact that knowledge of the illumination conditions is not required, is certainly an important aspect, but less so than the formulation's independence of surface albedo.

The penalty for not making assumptions about surface type and for not presupposing any knowledge of scene parameters, such as illumination conditions and surface albedo, is the introduction of higher-order derivatives of surface orientation in the formulation, as well as the inability to calculate surface orientation by purely local computation. Boundary conditions are necessary. To formulate a model that relates surface orientation to image irradiance is one thing; to solve it for that orientation is another.

The second-order partial differential equations (given above) relating surface orientation and image irradiance are satisfied by solutions to the first-order partial differential equation $\chi = 0$. This is undesirable, as solutions of $\chi = 0$ satisfy the surface-orientation-image-irradiance equations independently of the image measurements, I_x,I_y,I_{xx}, I_{yy}, and I_{xy}. The equation $\chi = 0$ characterizes the developable surfaces, e.g., a cylinder or cone (see Appendix B); derivation of the above surface-orientation-image-irradiance equations is impossible when the surface is developable, i.e., singularly curved. The surface-orientation-image-irradiance equations are appropiate only when the surface is doubly curved. For singularly curved surfaces,

the appropiate equations relating surface orientation and image irradiance are

$$I_x{}^2(m_y(1-l^2)+l_ylm) = I_y{}^2(l_x(1-m^2)+m_xlm) \quad ,$$

$$I_xm_y - I_ym_x = 0 \quad ,$$
$$I_xl_y - I_yl_x = 0 \quad .$$

(These equations are derived independently of any reflection function, i.e., they apply to all types of reflection, not just uniformly diffuse reflection. See Appendix C.)

If the surface-orientation-image-irradiance equations were solved by analytic procedures, the problems posed by the $\chi = 0$ solutions would vanish, as such solutions would be ruled out by boundary conditions. However, the presence of such solutions heralds difficulties for numerical methods, as the inevitable numerical errors will mix these solutions into the recovered surface orientations. Two approaches to solving the surface-orientation-image-irradiance equations have been reported [13]. These approaches are direct integration, which is implemented by finite-difference formulas, and relaxation. Both require additional information in the form of boundary conditions. Both fail to recover surface orientation. Direct integration correctly recovers the surface orientation in the vincinity of the boundary conditions, but is ineffective elsewhere. The reasons for failure of each method are of interest; direct integration fails because numerical instability makes the spatially serial method of solution impractical; relaxation fails because nonconvergence makes the temporally serial method of solution infeasible. These direct reasons for failure mask a deeper problem. The model is 'underconstrained' from the standpoint that the equations are insensitive to surface orientation. They are more sensitive to other surface parameters, such as surface curvature [13]. Underconstraint of the model

can account for lack of convergence of relaxation methods, but the numerical problems in direct integration highlight the difficulty of spatial information propagation by a mechanism that is under the control of higher-order derivatives.

The surface-orientation-image-irradiance equations alone do not form the basis for an algorithm to recover surface orientation; they do provide a tool, however, for examining the constraint shading imposes on shape. We shall subsequently use them for that purpose.

3. Local Computation Versus Global-Constraint Propagation

What can we learn from these various approaches to shape-from-shading? Direct integration of a differential model is an inadequate computational tool. Horn and his colleagues, using a low-order partial differential equation, show that some propagation of information is possible — but numerical instability poses a difficulty even for a first-order equation. This limited success with direct integration is unlikely to be upgradable to a solution procedure for natural scenes. Since higher-order formulations are plagued with numerical instability they do not offer any prospect of success.

A restricting factor in a differential model is the need for knowledge of boundary conditions. This seems to be a major limitation of such methods. These methods apply to continuous surface patches only and require a priori knowledge of solution values at some points within every region. This means that we must find regional boundaries — perhaps ascertain their type and estimate values of surface orientation

at some points within each region before we can attempt to recover shape. Is this, in effect, placing the cart before the horse?

Models of the relationship between image measurements and scene variables that are formulated as low-order differential equations offer no relief from the necessity of knowing scene parameters. While information about illumination conditions may be obtainable from other sources within the image, or maybe calculated in parallel with shape, it is difficult to envisage a situation in which the surface albedo could be calculated before the surface shape. Albedo would seem less constrained than shape. The author's higher-order differential equations show that derivatives of image irradiance can be used to remove these parameters.

While the relaxation schemes used to solve the image irradiance equation are not quite viable, their drawbacks may be attributed to the weakness of the surface shape constraint, namely smoothness, rather than an inherent deficiency of relaxation as a technique. For the higher-order surface-orientation-image-irradiance equations, insensitiveness to surface orientation does not allow assessment of the strength of surface continuity (the constraint used in the attempts to solve these equations by relaxation). The results reported from these relaxation procedures can be attributed to other aspects of the models they embody, rather than to any deficiency of the relaxation technique itself. Relaxation seems viable as a method that can satisfy global constraints without being dominated by numerical error. However, surface shape assumptions, that are more restrictive than those used in the work reviewed, appear necessary if information is to be propagated effectively over reasonable image distances. Relaxation schemes that implement low-order differential models seem practicable; schemes implementing higher-order differential

models are too sensitive to noise.

In comparison with information propagation techniques, local computation of surface orientation, as reported by Pentland, requires strong restrictions on surface shape — and even these are not adequate to characterize all cases. However, local computation, particularly when it is based on a model involving derivatives of image irradiance only, does provide a means for recovering surface orientation without any knowledge of boundary conditions, without a priori regional segmentation (it may even help in this endeavor), and without knowing the scene parameters, especially albedo. Unfortunately, we shall not get a solution to surface orientation that is quantitively correct because the surface restriction is too great. Local computation offers the computational features we want, but the penalty to be paid — severe surface shape restriction — is far too great.

What, then, seems practical? A relaxation scheme that is more constrained by surface type than those that have been examined? A scheme that implements a low-order model of information propagation? A scheme that does a lot of purely local computation? A scheme that can use boundary conditions wherever they are, but without being overly dependent on them? Of course, all this is one conjecture. There may well be a group of models that provide purely local computation, along with a means of determining when each model is applicable. Higher-order differential models, however, or low-order differential models that require too much a priori scene knowledge do not appear practicable. For any realistic model it seems inevitable that local processing must play an important role. Consequently, what can we compute locally from the shading data? This is the question we shall now address.

4. Analysis of Local Computation

The relationship between surface orientation and image irradiance for a uniformly diffuse reflecting surface that is doubly curved is given by the surface-orientation-image-irradiance equations of Section 2.3. Parameter counting reveals that local image measurements are insufficient to specify surface orientation for the general case, but shape constraints can overcome these degrees of freedom. Since we wish to calculate surface shape locally, we consider the case in which we can assume a constant curvature over the small surface patch from which we draw information for the local calculation. Of course the curvature varies with direction; we only assume that we can ignore any change in curvature over the surface patch. Of course, this assumption is not valid in general; we are restricting our attention to this case to simplify the analysis. If we cannot determine what shape information is available in this restricted case, we are not likely to understand the general case. For this case, when we ignore curvature change, $l_{xx} = l_{yy} = l_{xy} = m_{xx} = m_{yy} = m_{xy} = 0$, and from the surface-orientation-image-irradiance equations we derive the expressions

$$\frac{I_{xx}}{I_{xy}} = \frac{{l_x}^2(1-m^2) + {m_x}^2(1-l^2) + 2l_x m_x lm}{l_x l_y(1-m^2) + m_x m_y(1-l^2) + (l_x m_y + l_y m_x)lm} \quad ,$$

$$\frac{I_{yy}}{I_{xy}} = \frac{{l_y}^2(1-m^2) + {m_y}^2(1-l^2) + 2l_y m_y lm}{l_x l_y(1-m^2) + m_x m_y(1-l^2) + (l_x m_y + l_y m_x)lm} \quad .$$

Notice that these relationships are only between surface shape and the second derivatives of the image irradiance. It is the assumption of constant curvature, not the more restrictive sphericity assumption (used by Pentland to recover surface orientation from the second derivatives of image irradiance), that is necessary to relate shape and just the second derivatives of the image irradiance. Image

measurements are generally dependent on scene parameters other than those encoding shape. The first and second derivatives of image irradiance depend on the lighting position and the surface albedo, but the ratios of second derivatives are independent of all scene parameters except surface shape.

Can we use the above expressions to calculate surface orientation? We have previously [13] pointed to the insensitivity of surface-orientation-image-irradiance equations to surface orientation. The above expressions are also insensitive to surface orientation. We see this in the following considerations. Algebraic manipulation of the above expressions yields

$$\frac{I_{xx}}{I_{yy}} = \frac{{l_x}^2 + {m_x}^2 - (l_x m - l m_x)^2}{{l_y}^2 + {m_y}^2 - (l_y m - l m_y)^2} \quad .$$

Suppose that over an image patch we know values of l and m that satisfy the above expression. Consider now this expression for $\frac{I_{xx}}{I_{yy}}$ when

$$l' = w_1 l \quad and \quad m' = w_2 m \quad ,$$

at each point of the image patch. Using finite-difference formulas to calculate the derivatives of the surface normal, we obtain

$$\begin{aligned} \frac{I'_{xx}}{I'_{yy}} &= \frac{{l'_x}^2 + {m'_x}^2 - (l'_x m' - l' m'_x)^2}{{l'_y}^2 + {m'_y}^2 - (l'_y m' - l' m'_y)^2} \quad , \\ &= \frac{{w_1}^2 {l_x}^2 + {w_2}^2 {m_x}^2 - {w_1}^2 {w_2}^2 (l_x m - l m_x)^2}{{w_1}^2 {l_y}^2 + {w_2}^2 {m_y}^2 - {w_1}^2 {w_2}^2 (l_y m - l m_y)^2} \quad . \end{aligned}$$

Note that, as the magnitude of w_1 or w_2 is varied, the numerator and denominator of $\frac{I'_{xx}}{I'_{yy}}$ vary in like manner; both either increase or decrease; $\frac{I'_{xx}}{I'_{yy}}$ remains approximately equal to $\frac{I_{xx}}{I_{yy}}$. The ratios of the second derivatives of image irradiance are not

sensitive to surface orientation. We cannot get further shape information from other image measurements, as the first and second derivatives of image irradiance are dependent on the surface albedo and the lighting conditions, and the image irradiance is dependent on surface albedo, lighting conditions, and the level of constant illumination from such sources as atmospheric scatter and the dark current of the sensor. Surface orientation can be computed locally only when very restrictive assumptions about surface shape are made. Without such restrictions there is not enough information in the shading to decouple surface orientation effects from those of albedo and illumination.

If shading is insufficient to allow surface orientation to be recovered, what then does the shading specify? Does it specify curvature? Can we compute it locally? Consider the above expressions for $\frac{I_{xx}}{I_{xy}}$, and $\frac{I_{yy}}{I_{xy}}$. Suppose that we know the correct values for l and m at an image point and we want to calculate l_x,l_y,m_x, and m_y. If l_x,l_y,m_x, and m_y is a solution, then so is wl_x,wl_y,wm_x, and wm_y, where w is any constant. Curvature cannot be computed locally (without further shape assumptions). The ratios of second derivatives of image irradiance contain shape information, yet are insensitive to surface orientation and do not allow computation of the curvature. What information about the surface do they encode?

To answer this question, we first rewrite the expressions for $\frac{I_{xx}}{I_{xy}}$ and $\frac{I_{yy}}{I_{xy}}$ in vector dot product form:

$$\frac{I_{xx}}{I_{xy}} = \frac{[\frac{\partial}{\partial x}(l, m, \sqrt{1-l^2-m^2})].[\frac{\partial}{\partial x}(l, m, \sqrt{1-l^2-m^2})]}{[\frac{\partial}{\partial x}(l, m, \sqrt{1-l^2-m^2})].[\frac{\partial}{\partial y}(l, m, \sqrt{1-l^2-m^2})]} ,$$

$$\frac{I_{yy}}{I_{xy}} = \frac{[\frac{\partial}{\partial y}(l, m, \sqrt{1-l^2-m^2})].[\frac{\partial}{\partial y}(l, m, \sqrt{1-l^2-m^2})]}{[\frac{\partial}{\partial x}(l, m, \sqrt{1-l^2-m^2})].[\frac{\partial}{\partial y}(l, m, \sqrt{1-l^2-m^2})]} .$$

Using the notation $\mathbf{N} = (l, m, \sqrt{1-l^2-m^2})$, for the unit surface normal, we obtain

$$\frac{I_{xx}}{I_{xy}} = \frac{\mathbf{N}_x.\mathbf{N}_x}{\mathbf{N}_x.\mathbf{N}_y} \quad ,$$
$$\frac{I_{yy}}{I_{xy}} = \frac{\mathbf{N}_y.\mathbf{N}_y}{\mathbf{N}_x.\mathbf{N}_y} \quad ,$$

where $\mathbf{N}_x = \frac{\partial \mathbf{N}}{\partial x}$ and $\mathbf{N}_y = \frac{\partial \mathbf{N}}{\partial y}$.

For the case studied — when curvature changes are ignored — the ratios of the second derivatives of image irradiance measure the *relative squared curvature* of the surface. In other words, the ratios measure the relative change of the surface normal as we move along orthogonal image directions. However, relative curvature calculated locally at each image point constitutes insufficient information to allow surface shape reconstruction in the absence of further information about surface parameters. From shading information alone shape is an unattainable goal.

If we can find surface shapes, however, for which knowledge of relative curvature implies stronger information about the surface, e.g., surface orientation as in the case of a sphere, and if these surface shapes are reasonable approximations of the surfaces found in nature, then we may be able to recover stronger shape information locally. Locally there is not enough information to calculate surface shape without further knowledge, or without additional assumptions about surface shape. Pentland's work shows that an assumption of sphericity is strong enough to allow surface orientation to be calculated locally. Is this ability to calculate surface orientation specifically related to sphericity — or is it a feature that is generally true when we restrict the surface shape to cases in which the number of free parameters is no more than that for a spherical surface? In the foregoing discussion we have assumed that the surface is doubly curved. We shall now consider the images of singularly curved surfaces.

Just as we did for doubly curved surfaces, we assume that the derivatives of surface curvature can be ignored when we consider local computation of surface parameters. Differentiating the image irradiance equation, we obtain the same expression as before for the doubly curved surface, namely,

$$\frac{I_{xx}}{I_{yy}} = \frac{{l_x}^2 + {m_x}^2 - (l_x m - l m_x)^2}{{l_y}^2 + {m_y}^2 - (l_y m - l m_y)^2} \quad .$$

For a singularly curved surface ($l_x m_y - l_y m_x = 0$) when surface curvature is locally constant, the second derivatives of image irradiance are not independent, $I_{xx} I_{yy} = {I_{xy}}^2$. Consequently, we can derive only one expression relating shape and the second derivatives of image irradiance, rather than the two expressions we derived for doubly curved surfaces. As before, it follows that

$$\frac{I_{xx}}{I_{yy}} = \frac{\mathbf{N}_x.\mathbf{N}_x}{\mathbf{N}_y.\mathbf{N}_y} \quad .$$

At first, it might appear that there is more shape information in the first derivatives of image irradiance for

$$I_x m_y - I_y m_x = 0 \quad ,$$
$$I_x l_y - I_y l_x = 0 \quad .$$

But this is not the case, as the first and second derivatives of image irradiance are not independent. For singularly curved surfaces, when we ignore curvature change, $\frac{I_{xx}}{I_{yy}} = (\frac{I_x}{I_y})^2$.

For the singularly curved and doubly curved surfaces studied, local shading specifies the relative curvature of the surface along orthogonal image coordinates, which is the most we can hope to recover by local computation. In general, we cannot ignore curvature change over a patch. In this case, the information available

locally in the image combines data on relative curvature and curvature change. In the restricted case in which the the surface is assumed to be spherical the surface orientation can be calculated. However, this appears to be a very special situation based on the sphericity assumption rather than on a restriction in the number of parameters needed to specify the surface. Since surfaces in general are not locally spherical, one is forced to conclude that shading alone cannot enable prediction of surface shape by purely local computation.

5. Conclusions

The recovery of a scene's surface shape from its image is fundamental to the vision process. Our purpose in processing an image is the recovery of scene properties, not those of the image per se. In remote sensing it is these scene properties that we wish to measure, but, to extract them, we have to understand how these scene properties are manifested in the image data. A conceptual model of the relationship between scene and image parameters is provided by intrinsic images. Each intrinsic image specifies, for each point in the image, the value of one of the scene parameters that contribute to the measured image intensity. Vision models try to recover these parameters as best they can, whereupon a type of relaxation process adjusts their values so that they constitute a consistent interpretation of the scene's structure. Which parameters are specified by separate intrinsic images and which are composite is unknown, but it is essential that they be estimable without the need to know the values of the other intrinsic images. Shape-from-shading proposes a source of information, namely shading, from which shape information is

to be recovered — but what shape information does it actually encode?

Local shading specifies no more than the relative curvature of the scene's surface along orthogonal image directions. In general, even the recovery of relative curvature is complicated by change in the curvature of the surface. However, surface shape variables are related to image measurements in a fashion that is not dependent on knowing the other scene parameters. Shading provides direct shape information, but this is not enough for reconstruction of the surface shape. Further relationships between shape variables and image properties must be established before shape recovery is possible.

The various approaches reviewed have attempted to recover surface orientation from shading. To do so they have added extra information, such as known boundary conditions or constraints upon surface shape. The performance of these various models allows us to draw the following conclusions:

- Direct integration of differential models of scene properties requires much a priori information and has to contend with major computational problems.
- Local computation must play a major role in the recovery of scene parameters, but the models used have been overly restrictive in an effort to recover particular information.
- A relaxation mechanism, based on a strong low-order differential model, seems a viable means of propagating spatial information and constraints.

Shading provides a basis for an intrinsic image, specifying relative surface curvature and curvature change, but this intrinsic image alone is insufficient for surface shape recovery. Other models incorporating other image measurements are needed to complement shading. Such models should utilize the advantages of local

computation.

References

1. Barrow, H.G. and Tenenbaum, J.M., Recovering intrinsic scene characteristics from images, in: Hanson, A. and Riseman, E. (Eds.), *Computer Vision Systems*, Academic Press, New York, 1978 3-26.
2. Brooks, M.J., Surface normals from closed paths, *Proceedings of the Sixth IJCAI*, Tokyo, Japan, 1979 98-101.
3. Brown, C.M., Ballard, D.H. and Kimball, O.A., Constraint interaction in shape-from-shading algorithms, *Proceedings : Image Understanding Workhop*, Palo Alto, 1982 79-89.
4. Bruss, A.R., The image irradiance equation: its solution and application. AI-TR-623, Artificial Intelligence Laboratory, Massachusetts Institute of Technology (1981).
5. Courant, R. and Hilbert, D., *Methods of Mathematical Physics*, Volume 2, Interscience Publishers, New York, 1962.
6. Horn, B.K.P., Obtaining shape from shading information, in: Winston, P.H. (Ed.), *Psychology of Computer Vision*, McGraw-Hill, New York, 1975 115-155.
7. Horn, B.K.P., Understanding image intensities, *Artificial Intelligence* **8** (1977) 201-231.
8. Horn, B.K.P. and Sjoberg R.W., Calculating the reflectance map, A.I. Memo 498, Artificial Intelligence Laboratory, Massachusetts Institute of Technology (1978).
9. Ikeuchi, K. and Horn, B.K.P., Numerical shape from shading and occluding boundaries, *Artificial Intelligence* **17** (1981) 141-184.
10. Marr, D., Representing visual information, in: Hanson, A. and Riseman, E. (Eds.), *Computer Vision Systems*, Academic Press, New York, 1978 61-80.
11. Pentland, A.P., The visual inference of shape: computation from local features, Ph.D. Thesis, Department of Psychology, Massachusetts Institute of Technology (1982).
12. Smith, G.B., From image irradiance to surface orientation, (submitted for publication). Available as Technical Note 273, Artificial Intelligence Center, SRI International (1982).
13. Smith, G.B., The relationship between image irradiance and surface orientation, *Proceedings Computer Vision and Pattern Recognition Conference*, Washington, D.C. 1983.

14. Strat, T.M., A numerical method for shape from shading from a single image, S.M. Thesis, Department of Electrical Engineering and Computer Science,Massachusetts Institute of Technology (1979).
15. Woodham, R.J., A cooperative algorithm for determining surface orientation from a single view, *Proceedings of the Fifth IJCAI*, Cambridge, Massachusetts, 1977 635-641.
16. Woodham, R.J., Relating properties of surface curvature to image intensity, *Proceedings of the Sixth IJCAI*, Tokyo, Japan, 1979 971-977.

Appendix A

If a surface is twice differentiable, then

$$l_y(1-m^2)+m_y lm = m_x(1-l^2)+l_x lm \quad .$$

We call this the surface continuity equation, even though surface continuity is less demanding than the requirement that the surface be twice differentiable.
Proof: For a continuous twice-differential surface,

$$z_{xy} = z_{yx} \quad .$$

But $p = z_x$ and $q = z_y$, so

$$p_y = q_x \quad .$$

However,

$$p = \frac{-l}{\sqrt{1-l^2-m^2}} \quad ,$$

$$q = \frac{-m}{\sqrt{1-l^2-m^2}} \quad .$$

Hence,

$$p_y = -\frac{l_y(1-m^2)+m_y lm}{(1-l^2-m^2)^{\frac{3}{2}}} \quad ,$$

$$q_x = -\frac{m_x(1-l^2)+l_x lm}{(1-l^2-m^2)^{\frac{3}{2}}} \quad .$$

Then, substituting in $p_y = q_x$ yields

$$l_y(1-m^2)+m_y lm = m_x(1-l^2)+l_x lm \quad .$$

Appendix B

Developable surfaces are characterized by the differential equation

$$l_x m_y - l_y m_x = 0 \quad .$$

Proof: With the exception of a cylinder whose axis is parallel to the z axis, the differential equation defining all developable surfaces is [5]

$$z_{xx}z_{yy} - {z_{xy}}^2 = 0 \quad .$$

As the surface is twice differentiable, then $z_{xy} = z_{yx}$ so

$$z_{xx}z_{yy} - z_{xy}z_{yx} = 0 \quad .$$

As $p = z_x$ and $q = z_y$ then

$$p_x q_y - p_y q_x = 0 \quad .$$

But

$$p = \frac{-l}{\sqrt{1-l^2-m^2}} \quad ,$$
$$q = \frac{-m}{\sqrt{1-l^2-m^2}} \quad .$$

Hence,

$$p_x = -\frac{l_x(1-m^2)+m_x lm}{(1-l^2-m^2)^{\frac{3}{2}}} \quad ,$$
$$p_y = -\frac{l_y(1-m^2)+m_y lm}{(1-l^2-m^2)^{\frac{3}{2}}} \quad ,$$
$$q_x = -\frac{m_x(1-l^2)+l_x lm}{(1-l^2-m^2)^{\frac{3}{2}}} \quad ,$$
$$q_y = -\frac{m_y(1-l^2)+l_y lm}{(1-l^2-m^2)^{\frac{3}{2}}} \quad .$$

Substituting in $p_x q_y - p_y q_x = 0$ gives

$$l_x m_y - l_y m_x = 0 \quad .$$

Appendix C

The relationships between surface orientation and image irradiance for a developable surface are

$$I_x{}^2(m_y(1-l^2)+l_y lm) = I_y{}^2(l_x(1-m^2)+m_x lm) \quad ,$$
$$I_x m_y - I_y m_x = 0 \quad ,$$
$$I_x l_y - I_y l_x = 0 \quad .$$

Proof: Differentiating the image irradiance equation, $I(x,y) = R(l,m)$, we obtain

$$I_x = R_l l_x + R_m m_x \quad ,$$
$$I_y = R_l l_y + R_m m_y \quad .$$

Now

$$\begin{aligned} I_x{}^2(m_y(1-l^2)+l_ylm) = \; & R_l{}^2l_x(l_xm_y(1-l^2)+l_xl_ylm) \\ & +R_m{}^2m_x(m_xm_y(1-l^2)+l_ym_xlm) \\ & +2R_lR_ml_x(m_xm_y(1-l^2)+l_ym_xlm) \quad , \\ I_y{}^2(l_x(1-m^2)+m_xlm) = \; & R_l{}^2l_y(l_xl_y(1-m^2)+l_ym_xlm) \\ & +R_m{}^2m_y(l_xm_y(1-m^2)+m_xm_ylm) \\ & +2R_lR_ml_y(l_xm_y(1-m^2)+m_xm_ylm) \quad . \end{aligned}$$

But, for a developable surface $l_xm_y = l_ym_x$, (see Appendix B); hence

$$\begin{aligned} I_x{}^2(m_y(1-l^2)+l_ylm) = \; & R_l{}^2l_x(l_ym_x(1-l^2)+l_xl_ylm) \\ & +R_m{}^2m_x(m_xm_y(1-l^2)+l_xm_ylm) \\ & +2R_lR_ml_x(m_xm_y(1-l^2)+l_xm_ylm) \quad , \\ I_y{}^2(l_x(1-m^2)+m_xlm) = \; & R_l{}^2l_y(l_xl_y(1-m^2)+l_xm_ylm) \\ & +R_m{}^2m_y(l_ym_x(1-m^2)+m_xm_ylm) \\ & +2R_lR_ml_y(l_ym_x(1-m^2)+m_xm_ylm) \quad . \end{aligned}$$

Therefore,

$$\begin{aligned} I_x{}^2(m_y(1-l^2)+l_ylm) &= (R_l{}^2l_xl_y + R_m{}^2m_xm_y + 2R_lR_ml_xm_y)(m_x(1-l^2)+l_xlm) \quad , \\ I_y{}^2(l_x(1-m^2)+m_xlm) &= (R_l{}^2l_xl_y + R_m{}^2m_xm_y + 2R_lR_ml_ym_x)(l_y(1-m^2)+m_ylm) \quad . \end{aligned}$$

However, the surface continuity equation, (see Appendix A), is

$$l_y(1-m^2)+m_ylm = m_x(1-l^2)+l_xlm \quad .$$

We have the required result, i.e., that the relationship between surface orientation and image irradiance for a developable surface is

$$I_x{}^2(m_y(1-l^2)+l_ylm) = I_y{}^2(l_x(1-m^2)+m_xlm) \quad .$$

In terms of p and q, the equivalent form is

$$I_x{}^2q_y - I_y{}^2p_x = 0 \quad .$$

In terms of depth z, the equivalent form is

$$I_x{}^2z_{yy} - I_y{}^2z_{xx} = 0 \quad .$$

Note that, in addition,

$$\begin{aligned} I_xm_y - I_ym_x &= R_l(l_xm_y - l_ym_x) \quad , \\ I_xl_y - I_yl_x &= R_m(l_ym_x - l_xm_y) \quad . \end{aligned}$$

Hence, for a developable surface $l_xm_y - l_ym_x = 0$, we obtain the required results

$$\begin{aligned} I_xm_y - I_ym_x &= 0 \quad , \\ I_xl_y - I_yl_x &= 0 \quad . \end{aligned}$$

THE INFLUENCE OF SENSOR AND FLIGHT PARAMETERS ON TEXTURE IN RADAR IMAGES

V.S. Frost, K.S. Shanmugan, and J.C. Holtzman

Telecommunications And Information Sciences Laboratory
University of Kansas Center for Research, Inc.
Lawrence, Kansas 66045

ABSTRACT

Texture is known to be important in the analysis of radar images for geologic applications. It has previously been shown that texture features derived from the grey-level co-occurrence matrix (GLCM) can be used to separate large scale texture in radar images. Here the influence of sensor parameters, specifically the spatial and radiometric resolution and flight parameters, i.e., the orientation of the surface structure relative to the sensor, on the ability to classify texture based on the GLCM features is investigated. It was found that changing these sensor and flight parameters greatly affects the usefulness of the GLCM for classifying texture on radar images.

I. Introduction

Spectral, textural temporal and contextual features are four important pattern elements used in human interpretation of image data in general and SAR data in particular. Spectral features describe the average band-to-band tonal variations in a multi-band image set, whereas textural features describe the spatial distribution of tonal values within a band. Contextual features contain information about the relative arrangement of image segments belonging to different categories, and temporal features describe changes in image attributes as a function of time. However, when small image areas within, say, a synthetic aperture radar (SAR) image are independently processed on a computer for automated analyses only the tonal and textural features are usually available in making decisions.

In much of the automated procedures for processing radar image data from small areas, such as in crop-classification studies, only the average tonal values are used for developing a classification algorithm. Textural features are generally ignored on the basis that the poor resolution of radar imagery does not provide meaningful textural information for such applications since the areal extent of the target is usually small. However, there are many other applications such as the identification of large scale geological formations, land-use patterns, etc., where the resolution is more than adequate to provide

textural information. Indeed, in these applications texture is probably the most important image feature. It was previously shown [1] that texture features derived from the grey-level co-occurrence matrix (GLCM) can be used to discriminate texture in radar images. We describe in this paper the influence of sensor and flight parameters on our ability to quantitatively classify textures in radar images using the GLCM. The effect of spatial and radiometric resolution on texture classification was studied in one experiment. It was found that the classification was very sensitive to these sensor parameters, only the image with the best spatial and radiometric resolutions was quantitatively useful. Another experiment was conducted to determine how different flight paths, i.e., looking at the same terrain from different angles with the same sensor and incidence angle, changed the texture classification. Optical imaging systems rely on the sun to illuminate the scene and thus the sun angle becomes a factor; however, mission profiles for these sensors are usually designed to minimize this effect. For example, the LANDSAT series of sensors uses a high sun angle. On the other hand, imaging radars provide their own illumination and it is not clear what effect observing the same geologic structure from different angles will have on the automated analysis.

In the following section the texture features used here to separate different surface structures are briefly

described. The sensitivity of these texture features to changes in radiometric and spatial resolution is discussed next. Radar image simulation is then used to evaluate the sensitivity of GLCM texture features to changes in the orientation of the surface structure and the radar. The results of the two studies described in this paper indicate that the usefulness of textural features in automated analysis of radar images is sensitive to changes in the spatial and radiometric resolution of the system as well as the target/sensor geometry.

II. The Texture Features

The textural feature extraction algorithm employed here has been widely used [2-5] for analyzing a variety of photographic images. The procedure is based on the assumption that the texture information in an image block 'I' is contained in the overall or 'average' spatial relationship which the grey tones in the image 'I' have to one another. This relationship can be characterized by a set of grey-level co-occurrence (GLC) matrices. We describe a procedure for computing a set of GLC matrices for a given image block and define a set of numerical textural descriptors (features) that can be extracted from the GLC matrices. These textural features can be used for automated analysis and classification of blocks of radar imagery. Image texture may be viewed as a global pattern arising from

a deterministic or random repetition of local subpatterns or primitives. The structure resulting from this repetition could be very useful for discriminating between the contents of the image of a complex scene. A number of approaches have been suggested for extracting features that will discriminate between different textures [2-6]. Of these approaches, it has been found that textural features derived from grey-level co-occurrence matrices (GLCM) are most useful for analyzing the contents of a variety of imagery in remote sensing, biomedical and other applications [7-11]. The GLCM approach to texture analysis is based on the conjecture that the texture information in an image is contained in the overall or average spatial relationship between the grey tones of the image.

The second-order grey-level co-occurrence matrix of an image is defined as follows. Let $f(x,y)$ be a rectangular digital picture defined over the domain $x\epsilon[0,n_x)$, $y\epsilon[0,n_y)$, $x,y\epsilon I$. Let n_g be the number of grey levels in f. The unnormalized, second-order GLC matrix is a square matrix $\bar{P}$ of dimension n_g. The (i,j)-th entry in $\bar{P}$, denoted by $\bar{P}_{ij}$, is a function of the image tonal values and a displacement vector $\bar{d} = (d_1,d_2)$. The entries $\bar{P}_{ij}$ are unnormalized counts of how many times two neighboring resolution cells which are spatially separated by $\bar{d}$ occur on the image, one with grey tone i and the other with grey tone j. That is,

$$\bar{P}_{ij} = \#\left\{\left((m_1,n_1),\ (m_2,n_2)\right) \mid f(m_1,n_1) = i,\right.$$
$$\left. f(m_2,n_2) = j,\ \text{and}\ (m_2,n_2) - (m_1,n_1) = \bar{d}\right\}, \quad (1)$$

where # denotes the number of elements in the set, the indices m_1, m_2 and n_1, n_2 take on integer values in the intervals $[0,n_x)$, $[0,n_y)$. The normalized GLC matrix $\bar{P}$ with entries p_{ij} is obtained from $\bar{P}$ by dividing each entry in P by the total number of paired occurrences. The definition of second-order GLC matrices can be extended to include third- and higher-order GLC matrices. While higher-order GLC matrices may be important in some applications, much of the recent work in texture analysis has been based on second-order GLC matrices.

The second-order GLC matrices are computed for various values of the displacement vector $\bar{d}$, and features derived from the GLC matrices are used for classifying the contents of an image.

Some of the commonly used textural features derived from the GLC matrix are:

1) Uniformity (sum of squares):

$$\sum_{ij} p_{ij}^2 \quad (2a)$$

2) Contrast:

$$\sum_{i} \sum_{j} (i-j)^2 p_{ij} \quad (2b)$$

3) Correlation:

$$\sum_i \sum_j \frac{(i-\mu_x)(j-\mu_y)\ p_{ij}}{\sigma_x\ \sigma_y} \tag{2c}$$

4) Entropy:

$$\sum_i \sum_j p_{ij} \log p_{ij} \tag{2d}$$

5) Inverse Difference Moment:

$$\sum_{\substack{i \\ i \neq j}} \sum_j (p_{ij})^{\rho} / |i-j|^{v} \tag{2e}$$

6) Maximum Probability:

$$\max_{i,j} p_{ij} \tag{2f}$$

For a variety of imagery (aerial, micrographic and x-ray) the relationship between these textural features, their values and what they represent in terms of visual perception of texture are reasonably well understood. Using features of the form given above, Haralick and Shanmugan [5-7] were able to classify a variety of images with over 85% classification accuracy. These features have also been used to separate texture in radar images [1].

III. Texture Analysis Of SAR Images With Different Spatial And Radiometric Resolution

Numerical descriptions of texture (specifically those derived from the grey-level co-occurrence matrix (GLCM) as in Section II) have been shown to separate some simple geological features [1]. To efficiently design a spaceborne

SAR for geologic exploration it is of interest to determine how the ability to separate geological features using the GLCM desired features varies with important system parameters, e.g., spatial and radiometric resolution.

A limited set of radar images with different spatial and radiometric resolutions were obtained (primarily from the Jet Propulsion Laboratory [12]). These images were generated by appropriate processing of the Seasat-A SAR video signal, and were of a geologically interesting area in Tennessee (Figure 1). The specific areas that were studied are outlined in white. The combinations of spatial and radiometric resolution contained in this data set were (25 m, 4 looks), (50 m, 4 looks), (100 m, 4 looks), (50 m, 2 looks), and (50 m, 1 look). Within the Tennessee test area, five distinct textures were identified (see Table 1 for a description of the geology and topography) and five to seven samples of each texture obtained (see Figure 1). A sample of a texture is an image (in this case 3.4 km x 3.4 km in size) containing only one texture type. Thus for each set of sensor parameters 30 texture samples were obtained, a total of 150 texture samples (images) were used in this study. For each texture sample a GLCM was calculated and texture features found. Specifically, uniformity, contrast, correlation, entropy, inverse difference moment, and maximum probability were the texture features used here. Following [1] the GLCM were calculated for distances of 1, 2 and 4 at

angles of 0°, 45°, 90°, and 135°. The above texture features were calculated for each distance and angle. In addition, the average over all angles for each texture feature was calculated. Thus each texture sample is described by a set of 30 numbers (6 texture features, 4 angles, and the average for each feature).

Scatter diagrams of the numerical values for one pair of texture features are shown in Figure 2. These plots are for distance 4 and result from averaging all four GLCM angles. All five textures can be separated using the correlation and maximum probability (Figure 2) features only for the system with a 25 m spatial resolution and with four independent samples averaged. As either the radiometric resolution is degraded (decreased number of independent samples or looks averaged) or the spatial resolution is degraded the ability to separate these textures is also degraded. This same result was found for other combinations of texture features [13]. In all cases only the images with 25 m, 4 looks could be quantitatively used to separate these textures using the GLCM.

This experiment reinforces the conclusions of our previous work [1]: automatically derived texture features can be used to discriminate texture in radar images of rough terrain. Additionally, this study shows that the ability to use the GLCM to classify texture is strongly dependent upon both the sensor's spatial and radiometric resolution. Even

though the data set used for this study was very limited these results do indicate that the usefulness of textural features for radar image analysis is sensitive to the spatial and radiometric resolutions of the sensor. This should be expected because it is well known that for manual analysis the interpretability of radar images is sensitive to the radiometric and spatial resolutions [14-17]. Thus, this study demonstrated that this sensitivity also exists for automatic analysis.

IV. A Study Of The Effect Of Look Direction On Texture In SAR Images

For an automatic texture analysis system for radar to be successful, a set of texture features must be found which are invariant to the flight path of the sensor. This invariance is clearly needed because the orientation of the terrain features relative to the sensor's flight path is not known a priori. For the geologic analysis of radar imagery where terrain elevation plays a dominant role the imaging geometry of radar would seem to be a dominant factor. Also the question of invariance is important in the search for 'optimum' sensor configurations. For example, it might be possible to classify certain terrain features at one sensor orientation but not at another. However, because the orientation of the sensor to the terrain features of interest will never be known a priori an optimum sensor configuration might not exist.

The purpose of this section is to describe the results of an experiment which was aimed at determining the sensitivity of GLCM texture features, shown to be valuable as a discriminate, to the sensor flight direction, i.e., the target/sensor orientation. It was found (given the limitations of the experiment) that the texture features considered here could be classified for one or two target/sensor orientations but not for all the three orientations considered here.

To isolate the effect of sensor look direction it was necessary to use radar simulation [18] to create a set of images with controlled terrain and sensor parameters. Further it was possible using the simulation approach to remove (i.e., not include) the effect of speckle [18]. Therefore, this study focused on how shadow, layover, and range compression changed the image manifestations of complex terrain structure as the look direction of the sensor was varied.

In radar image simulation (for a complete description see [18]), the terrain to be analyzed is represented as a two-dimensional integer array referred to as a data base. This array is stored on a file containing fixed-length records. These correspond directly to rows in the array which contain a fixed number of words (columns). This relationship is shown in Figure 3.

The three data bases used in this study were generated

from data received from the U.S. Geological Survey in the form of three digital elevation models. These were received containing elevation values which correspond directly to a 1:24000 (1 inch = 2000 feet) topographical map sampled at 30 meter intervals in both the x and y directions. Let x define columns in our data base and y to refer to rows (see Figure 3). In these data x and y both represent 30 meters on the ground. Thus each elevation value was considered to be valid for an area of 30x30 square meters.

The third dimension of the data base, h, represents the elevation of each cell above a given reference elevation. Each increment in elevation corresponds to Δh, which describes a scaling factor for determining the quantization of the actual elevation. In the digital elevation models used in this study, the value for Δh was equal to one meter. This led to a convenient one-to-one relationship for the elevations.

The relationship among the values for Δx, Δy and Δh describes the degree to which the elevation changes over an area on the ground. Since only the relative structures of the terrain are of interest in this study, this relationship may be altered as needed. After first removing the reference elevation constant the data was scaled by 0.25. This allowed the Δx and Δy values to represent 7.5 meters, while the value for Δh remained equal to 1 meter.

The simulation of synthetic aperture radar imagery is

made possible through the application of a computer program developed at the University of Kansas Remote Sensing Laboratory [18]. This algorithm will simulate the effects of a spaceborne SAR with a look direction parallel to the rows of the data base array. Since the simulation program always processes the data row-by-row, the only way to achieve a different look direction is to modify, i.e., rotate, the data base. Keeping this in mind, the unmodified data base is defined to be at a look direction angle of 0°. For this study simulated radar imagery was to be generated for the same areas with look directions of 0°, 45°, and 90°. This required that the data bases be correctly oriented before the simulation was performed. For this, computer programs were applied to rotate the original data in order to simulate different look angles. Nine data bases were thus available for simulation (3 terrain models at 3 look directions). These nine data bases were then processed using the simulation program. The radar parameters used for the simulation were similar to those of the Seasat-A SAR. The altitude of the sensor was considered to be roughly 800 kilometers, and the angle of incidence between the sensor and the first cell of the data base was given to be 20 degrees. For the purposes of this study, it was assumed that all of the terrain data was of one scattering category. The scattering coefficient as a function of incidence angle is shown in Figure 4.

Using these parameters along with the assumed value of 7.5 meters for both along-track and across-track resolutions, radar images were simulated, producing the desired set of controlled images. However, these images are now rotated relative to each other. To eliminate the rotational dependence of the GLCM the simulations were converted to one coordinate system.

Visually the effect of changing the flight path is dramatic. Figure 5a-c contains the simulated radar images for one of the digital terrain models. In Figure 5a the sensor's look direction is from right to left. This is our reference direction and is referred to as the 0° look angle. The simulation of a 45° look angle (i.e., from the upper left to the lower right) is shown in Figure 5b and the 90° simulation (i.e., from top to bottom) is shown in Figure 5c. Similarly, Figures 6a-c and 7a-c contain the image simulations for two other digital terrain models. Close analysis of these images reveals many features which are totally obscured by shadow at one look angle but not at the others as was shown in [17]. Also, the spatial structure changes as the look angle is varied from 0° to 45° to 90°.

Beginning with the 0° look direction three distinct spatial structures, textures, were identified.

TEXTURE 1 contained low relief with some small hills and ridges, maximum relief is about 100-300 feet.

TEXTURE 2 contained elongated ridges and mountains usually separated by steep gradient streams, maximum relief is about 500-700 feet.

TEXTURE 3 contained long, narrow valleys with steep slopes and depths of about 300-400 feet. Valley streams have medium to low gradients.

From each texture, 3 or 4 samples (subimages) were obtained. The same subimages were then sampled from the 45° and 90° look direction simulations. A total of 33 subimages provided the input for this experiment (11 subimages for each look direction). These subimages are shown in Figure 8a-c. The specific research questions addressed by this experiment were (1) can these three textures be classified using GLCM features at any of the three look directions, and (2) can these three textures be classified using GLCM features independent of the look direction, i.e., are the texture features derived from the same spatial structure independent of the look direction of the sensor, thus, can the textures be classified using all three orientations simultaneously.

For each of the 33 subimages described above a GLCM and the resulting texture features were calculated for distances of 4, 6, and 10 at 0°, 45°, 90°, and 135° (these angles will be referred to as GLC angles as opposed to the look direction angle). It was found [13] that distances 4 and 10

showed basically the same trend as 6 so only the distance 6 results will be discussed. Also, it was found that averaging the texture features over the GLC angle as was done previously [1] destroyed our ability to separate textures one and two. This is expected from their spatial structure. Thus only results from individual GLC angles will be presented. The GLCM texture features were analyzed pair-wise as was also done previously [1].

Analysis of the data qualitatively showed that all three textures could be classified at one or two target/sensor orientations but not at all three simultaneously. For example, Figure 9a-c contains the scattergrams for the maximum probability and contrast texture features at GLC distance 6 and GLC angle of 0°. At a look direction of 0° (Figure 9a) none of the three textures can be separated, while at 45° (Figure 9b) all three textures can be classified. Analysis of other texture pairs shows the same trend, i.e., the textures considered here can be classified for one or two sensor look directions but not at all three [13]. If the texture samples for each terrain structure from all three look directions are combined it becomes obvious that the textures considered here cannot be classified independent of look direction (see Figure 10a-e).

The purpose of this analysis was to determine the sensitivity that GLCM texture features show to changes in

the orientation of the surface structure relative to the sensor. Radar image simulation was used to generate a suitable set of images with the effects of the sensor flight path isolated. Within the limitations of this experiment, i.e., three different terrain structures, and three flight directions, it was shown that (1) the GLCM texture features can be used to classify the terrain structures at one or two flight directions but not at all three, and (2) the GLCM texture features cannot be used to classify these terrain structures independent of the flight path. The search of the optimum set of sensor parameters for geologic applications is thus complicated. That is, the results of this study indicate that the optimum sensor for classifying (using either manual or automatic techniques) surface structure is dependent upon the orientation of the structure to the flight path of the sensor. Because of the monostatistic nature of radar imaging the same surface structure imaged at two different flight angles can (and often do) appear totally dissimilar. A set of sensor parameters optimized to detect these structures at one flight angle might be totally different if the flight angle were changed.

V. Conclusions And Recommendations

Texture is an important characteristic of radar images of rough terrain. It was shown that the GLCM derived

texture features can be used to classify texture. In this paper we have demonstrated that GLCM derived texture features are sensitive to both sensor and flight parameters. In fact, we lose our ability to classify texture by these features if either the radiometric or spatial resolution is degraded. We also found that these texture features are sensitive to the sensor flight path. We could classify the surface structure for one or two target/sensor orientations but not for all three considered simultaneously. That is, GLCM texture features cannot be used to classify texture independent of the flight path.

While general conclusions on the sensitivity of textural features to system and flight parameters can be made from the results of this study, there is a need to further refine these conclusions, specifically it is recommended that the sensitivity shown here be quantitatively studied. Quantitative results are needed to help guide system design and flight planning. Two approaches to obtaining quantitative results should be pursued in parallel. First, an analytic study of the relationships among surface, sensor and flight parameters and the GLCM is needed. Second, more radar images should be analyzed. With more data the qualitative discussion of the effects of spatial and radiometric resolution can be extended to a quantitative analysis, for example plots of the 'variance' of each cluster as a function of resolution

could then be studied. The ultimate goal of such an analysis would be an expression for the sensitivity of each texture feature as a function of resolution. This study also dealt with only radical changes in the flight direction over a fixed site. Further analysis is now needed to determine the effect of small angle changes, e.g., on the order of 5°. Also this study only considered one angle of incidence. It would be interesting to determine if there exist some incidence angle for which we could classify surface structure independent of the flight angle.

Acknowledgments

The authors are indebted to the individuals who have contributed to this study, Dr. Josephine Stiles and Mr. Scott Gardner. Also, thanks are expressed to Dr. Dan Held of the Jet Propulsion Laboratory for supplying the SEASAT-A SAR data.

References

[1] Shanmugan, K.S. et al., Textural features for radar image analysis, IEEE Trans. on Geoscience and Remote Sensing, GE-19 (1981) 155-156.

[2] Rosenfeld, A. and Troy, E.B., Visual Texture Analysis, Computer Science Center, University of Maryland (1970).

[3] Haralick, R.M. and Anderson, D., Texture Tone Study, RSL TR 182-2, Remote Sensing Laboratory, University of Kansas Center for Research, Inc. (1971).

[4] Sutton, R.N. and Hall, E.L., Texture measures for automatic classification of pulmonary disease, IEEE Trans. on Computers, C-21 (1972) 667-676.

[5] Haralick, R.M., Shanmugan, K. and Dinstein, I., Textural features for image classification, IEEE Trans. on Systems, Man, and Cybernetics, SMC-3 (1973) 610-621.

[6] Galloway, M.M., Texture analysis using gray level run lengths, Computer Graphics and Image Processing, 4 (1975) 172-199.

[7] Shanmugan, K. and Haralick, R.M., Computer classification of reservoir sandstones, IEEE Trans. on Geoscience Electronics, GE-11 (1973) 171-177.

[8] Haralick, R.M. and Shanmugan, K., Combined spectral and spatial processing of ERTS imagery, Remote Sensing, 3 (1974) 3-13.

[9] Kruger, R.P., Thomson, W.B. and Turney, F.A., Computer diagnosis of pneumoconiosis, IEEE Trans. on Systems, Man, and Cybernetics, SMC-4 (1974) 40-49.

[10] Weszka, J.S., Dyer, C.R. and Rosenfeld, A., A comparative study of texture measures for terrain classification, IEEE Trans. on Systems, Man, and Cybernetics, SMC-6 (1976) 269-285.

[11] Connors, R.W. and Harlow, C.A., A theoretical comparison of texture algorithms, IEEE Trans. on Pattern Analysis and Machine Intelligence, PAMI-2 (1980) 204-223.

[12] Ford, J.P., Resolution versus speckle relative to geologic interpretability of spaceborne radar images: a survey of user preference, IEEE Trans. on Geoscience and Remote Sensing, GE-20 (1982) 434-444.

[13] Frost, V.S. et al., Topics in Radar Image Processing, RSL TR 453-4, Remote Sensing Laboratory, University of Kansas Center for Research, Inc. (1982).

[14] Moore, R.K., Tradeoff between picture element dimensions and noncoherent averaging in side-looking airborne radar, IEEE Trans. on Aerospace and Electronic Systems, AES-15 (1979) 697-708.

[15] DiCaprio, G.R. and Wasielewski, J.E., Radar image processing and interpreter performance, Photogrammetric Engineering and Remote Sensing, 52 (1976) 1043-1048.

[16] Porcello, L.J., Speckle reduction in synthetic-aperture radars, J. Opt. Soc. Am., 66 (1976) 1305-1311.

[17] Dellwig, L.F., Hanson, B.C., Hardy, N.E., Holtzman, J.C., Hulen, P.L., McCauley, J.R. and Moore, R.K., A Demonstration and Evaluation of the Utilization of Side Looking Airborne Radar for Military Terrain Analysis, RSL TR 288-1, Remote Sensing Laboratory, University of Kansas Center for Research, Inc. (1975).

[18] Holtzman, J.C. et al., Radar image simulation, IEEE Trans. on Geoscience and Remote Sensing, GE-16 (1978) 296-303.

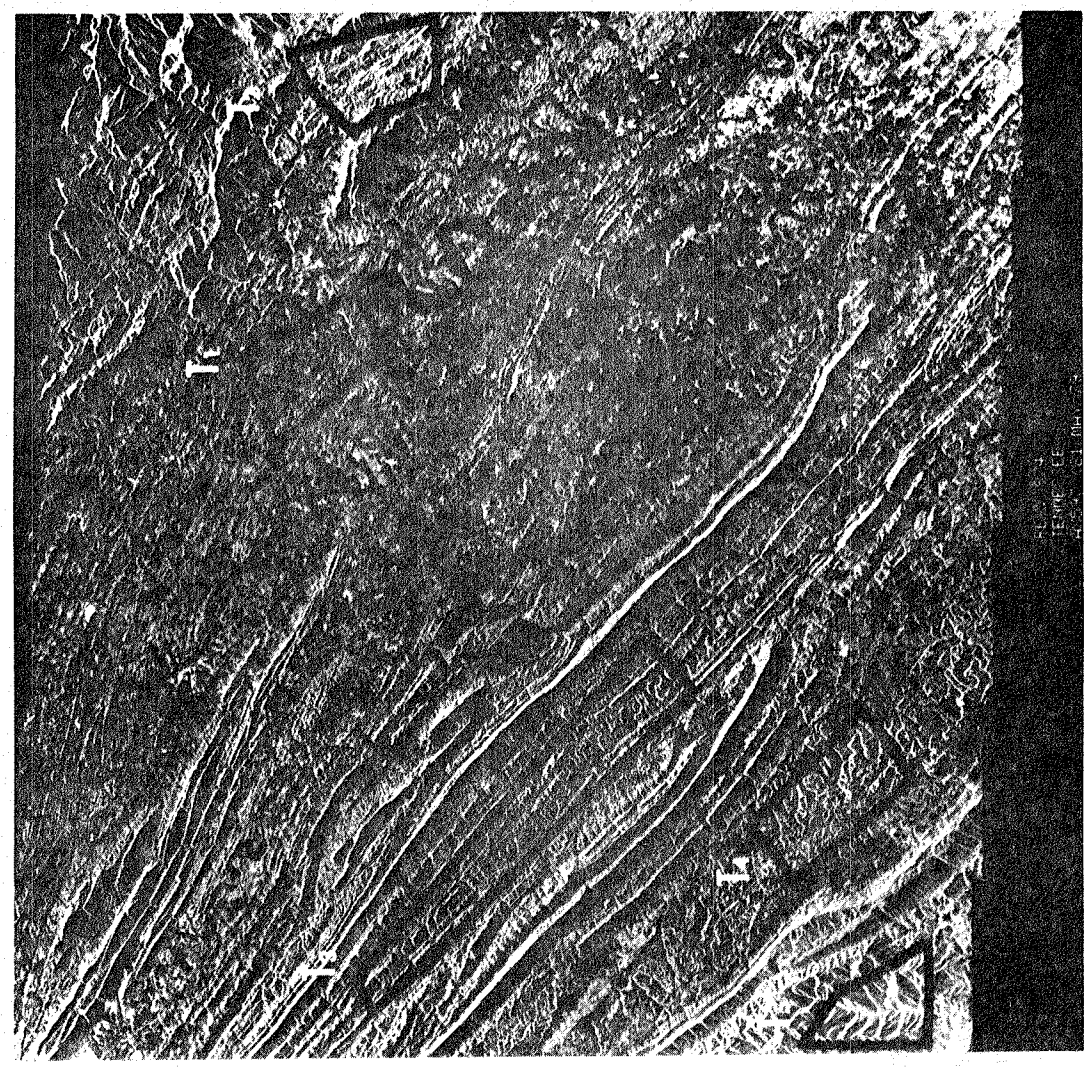


Figure 1. SEASAT-A SAR Image.

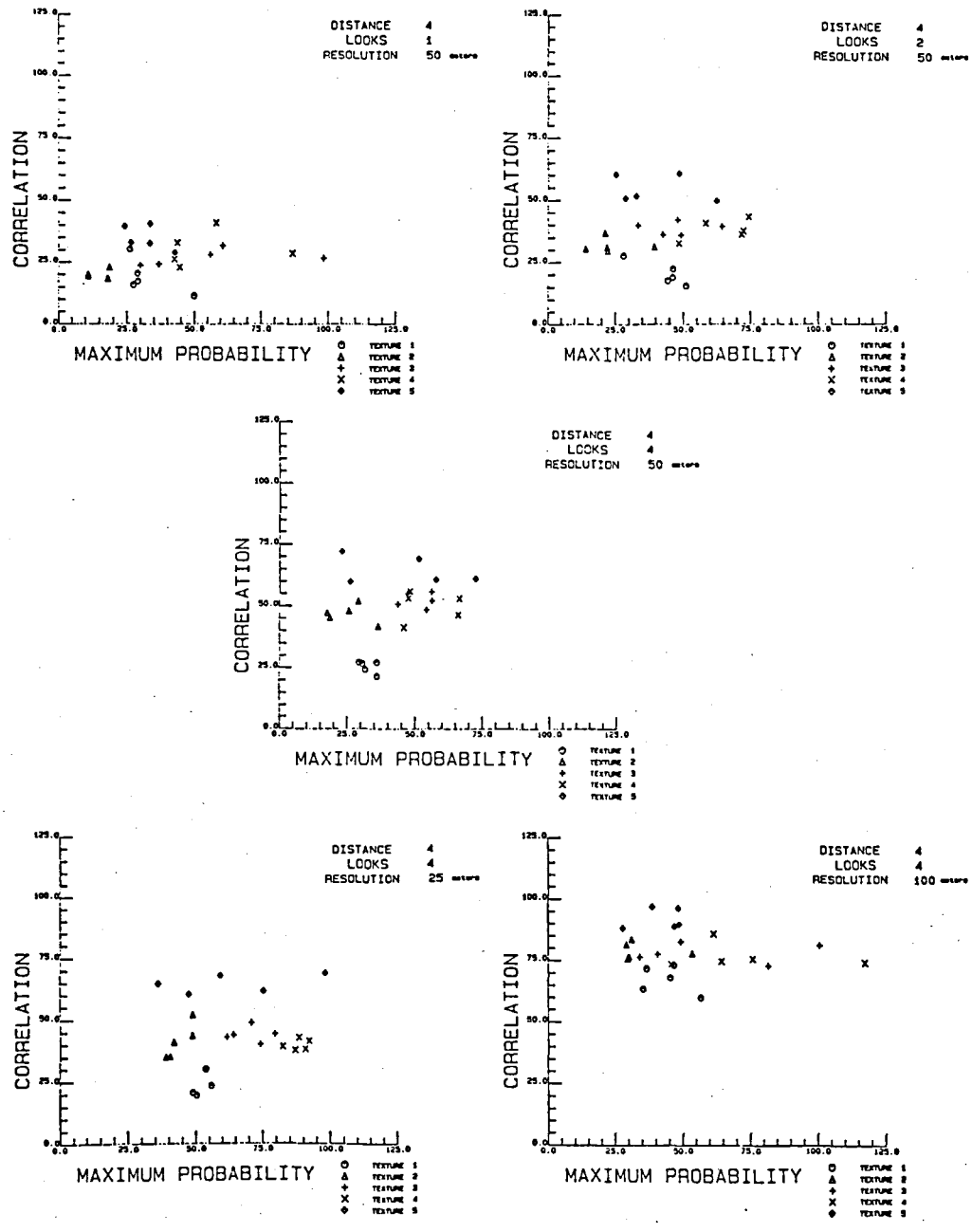


Figure 2. Scatter Diagrams for Texture Feature Pairs as a Function of Spatial and Radiometric Resolution (Average Over All GLC Angles). Maximum Probability/Correlation.

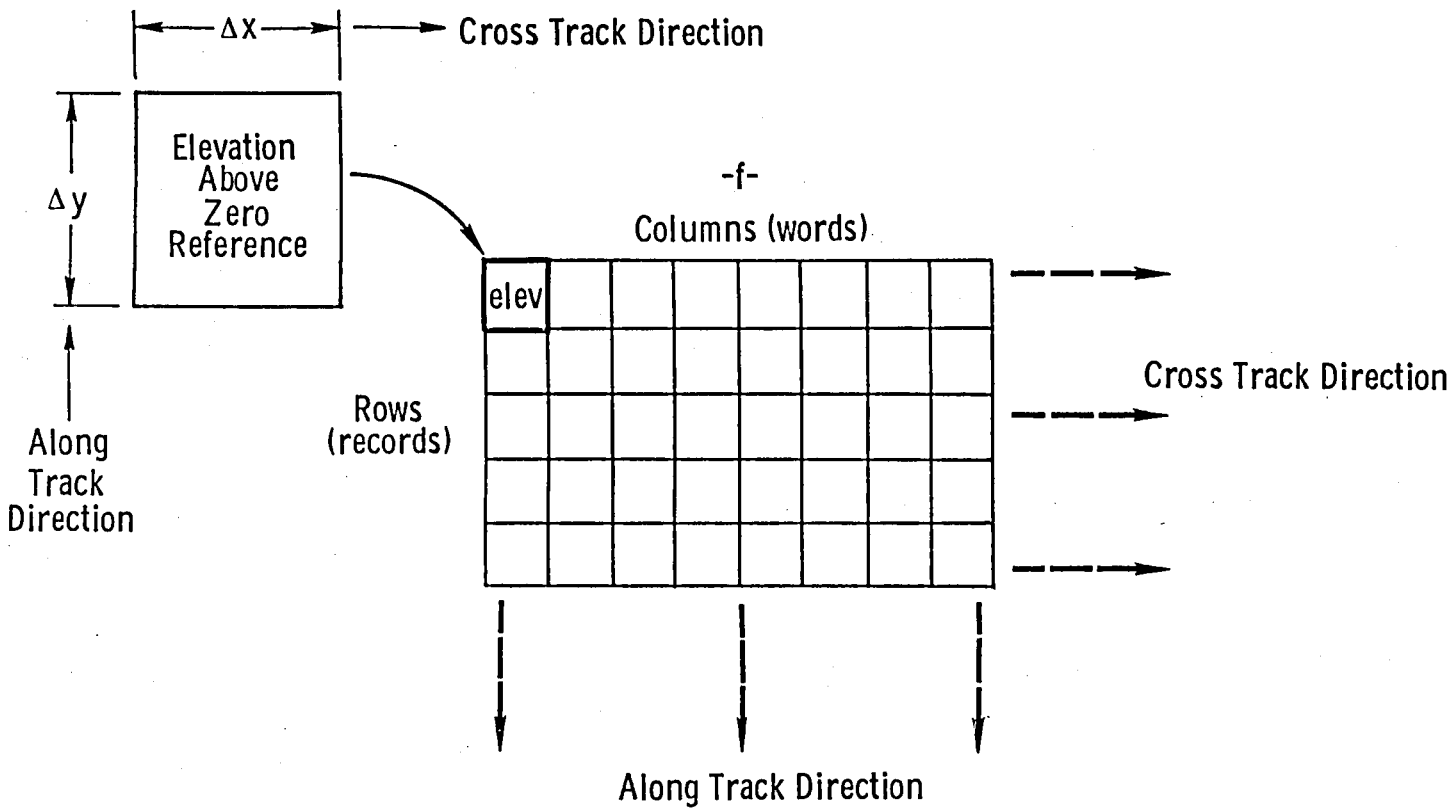


Figure 3. Data Base Geometry for Radar Image Simulation.

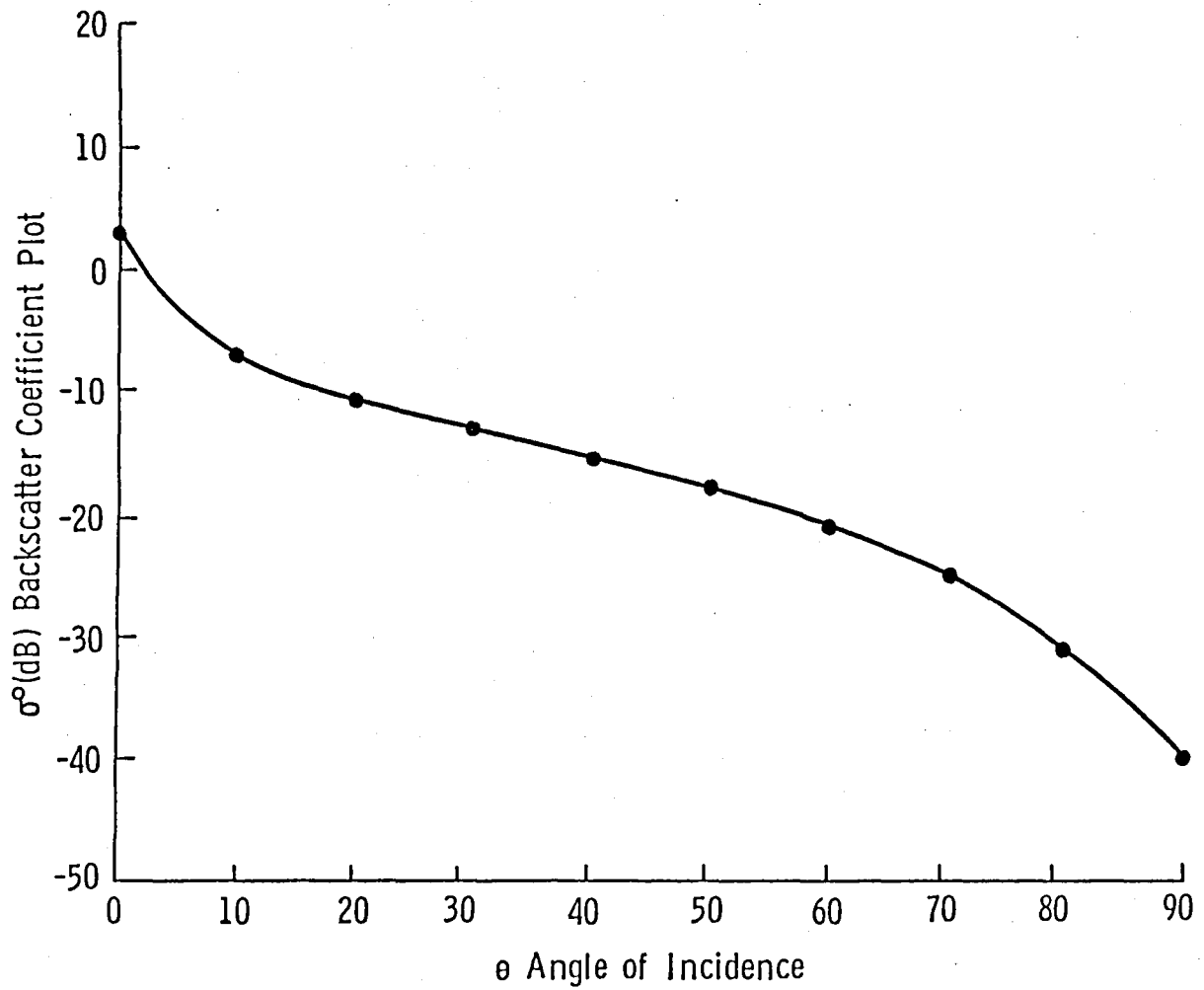


Figure 4. Backscattering Function Used in Radar Image Simulations.

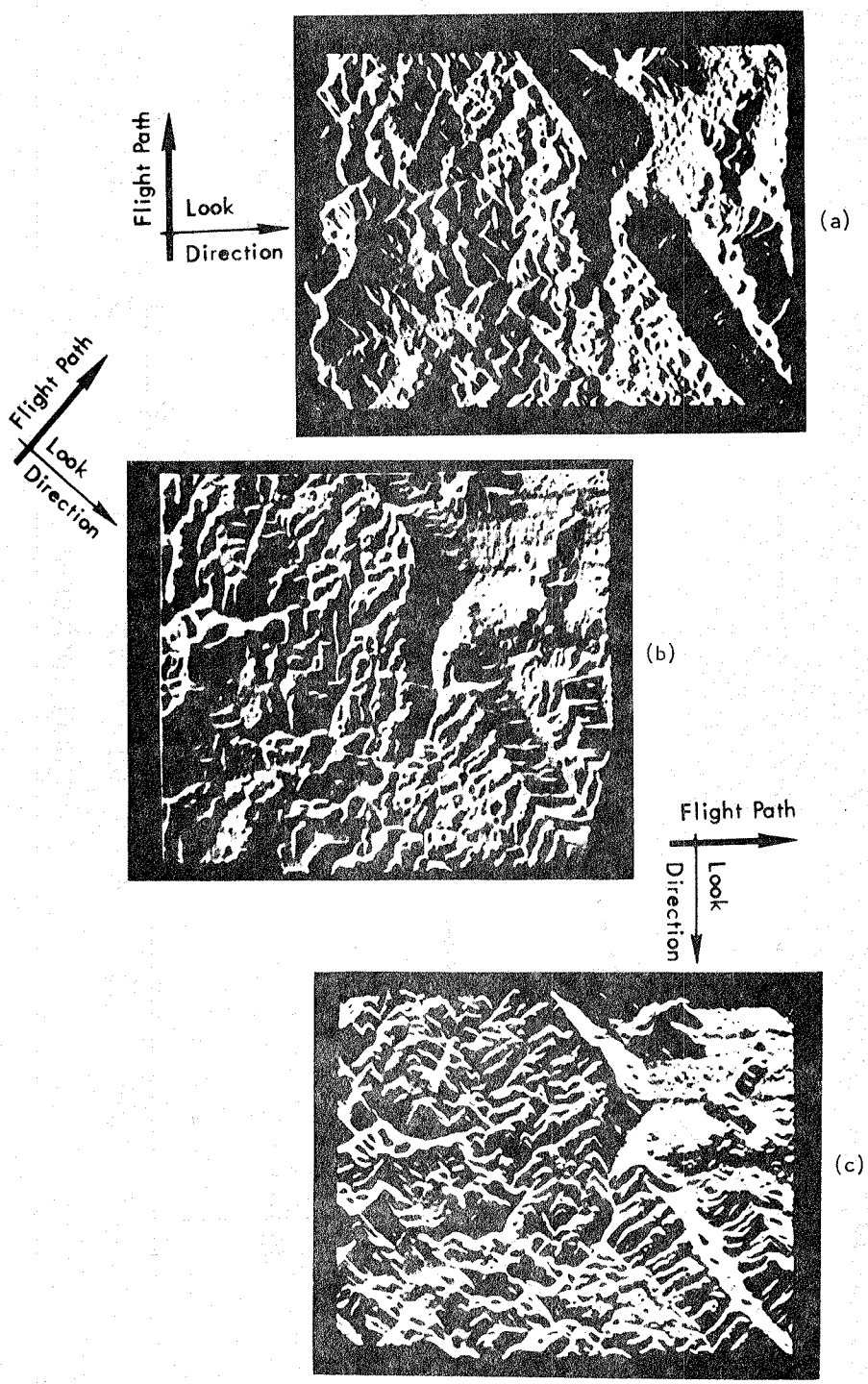


Figure 5. Radar Image Simulations for Three Flight Paths for Site 1.

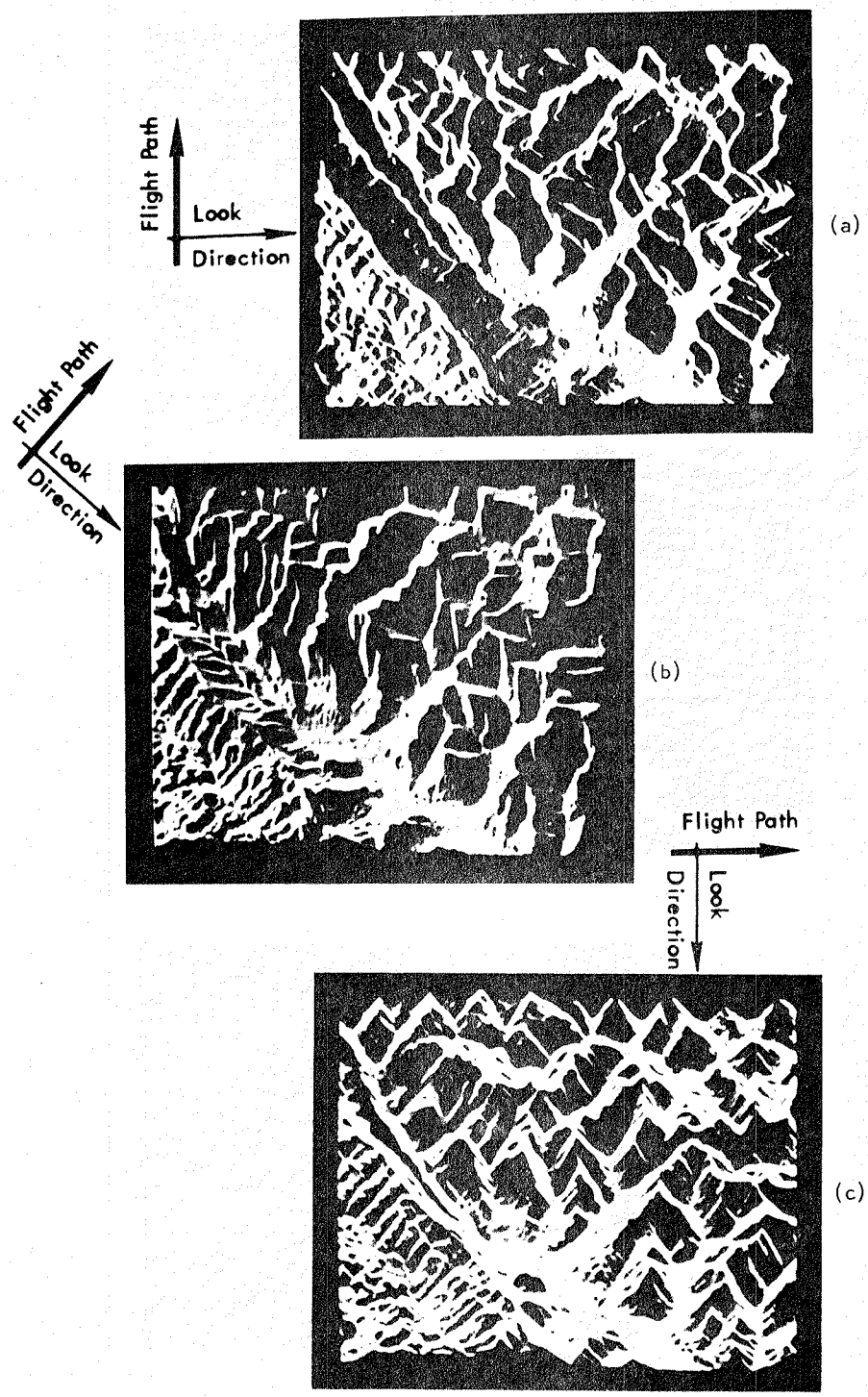


Figure 6. Radar Image Simulations for Three Flight Paths for Site 2.

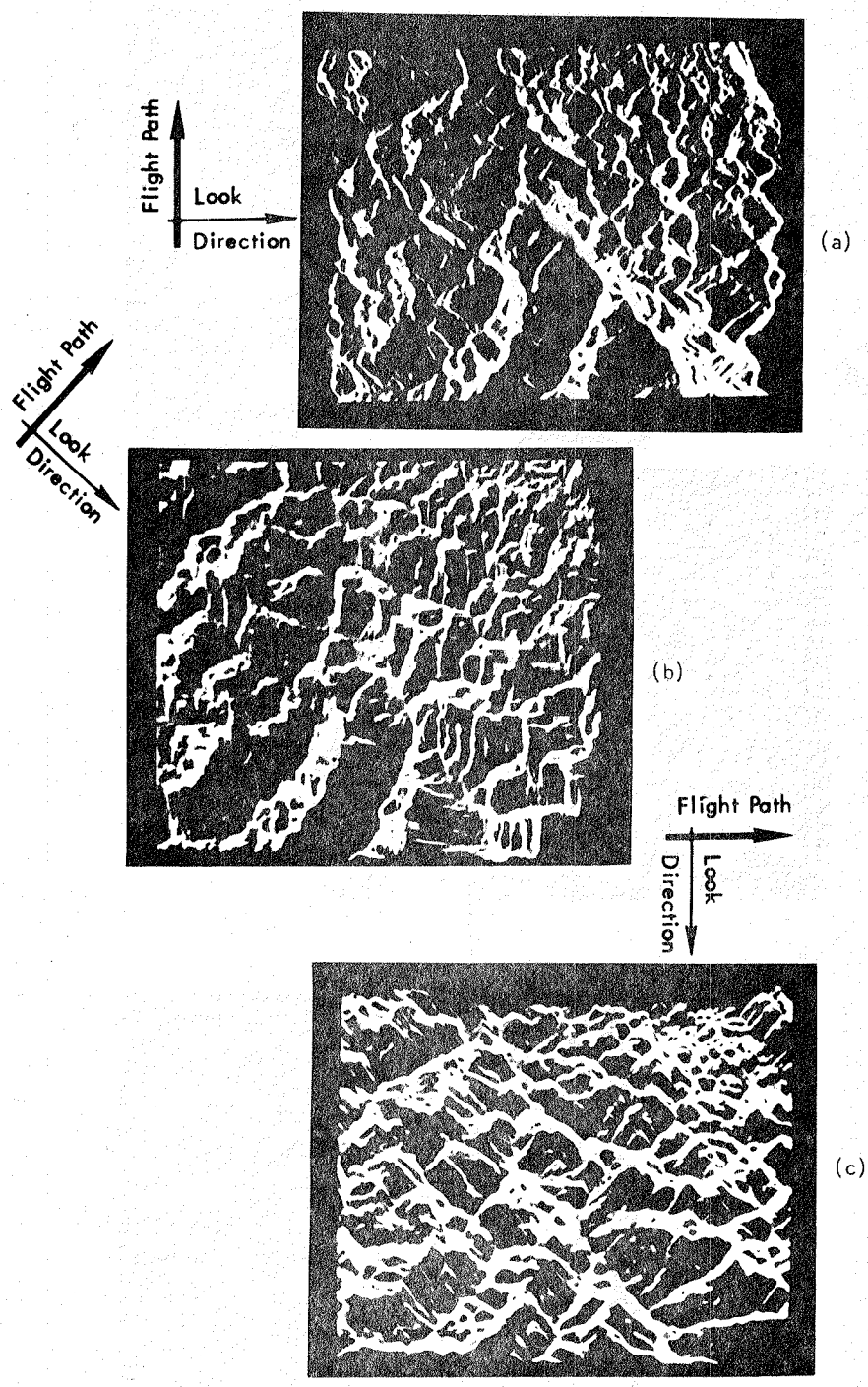


Figure 7. Radar Image Simulations for Three Flight Paths for Site 3.

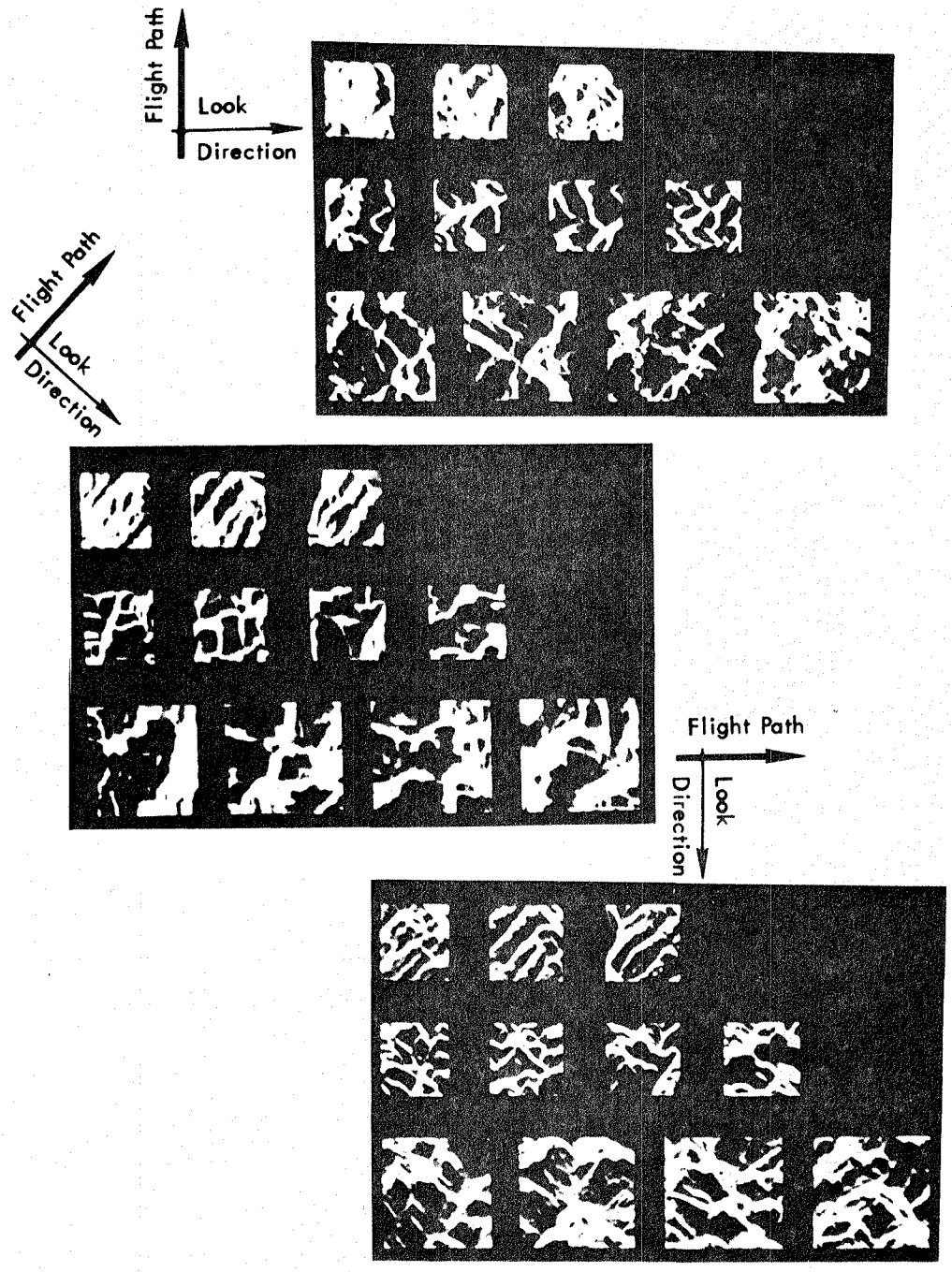
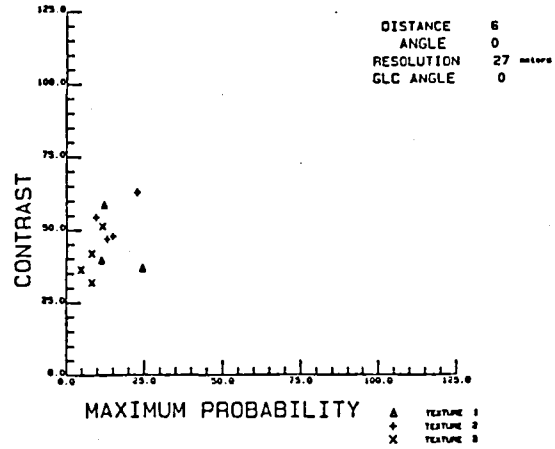
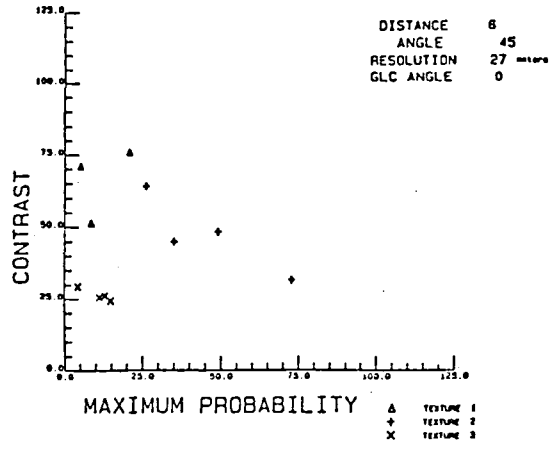


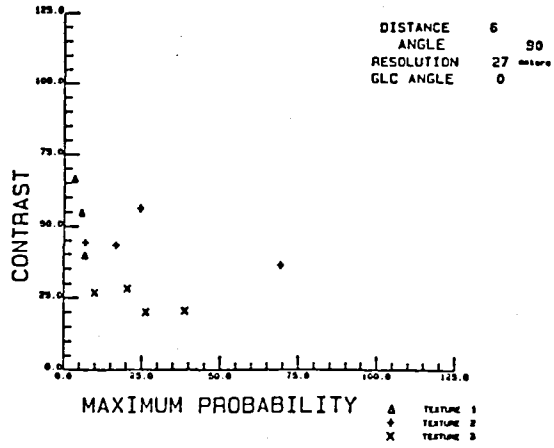
Figure 8. Texture Samples from the Radar Image Simulations.



(a) 0°



(b) 45°



(c) 90°

Figure 9. Scatter Diagrams for Texture Feature Pairs, Contrast/Maximum Probability, at a 0° GLCM Angle as a Function of Flight Path.

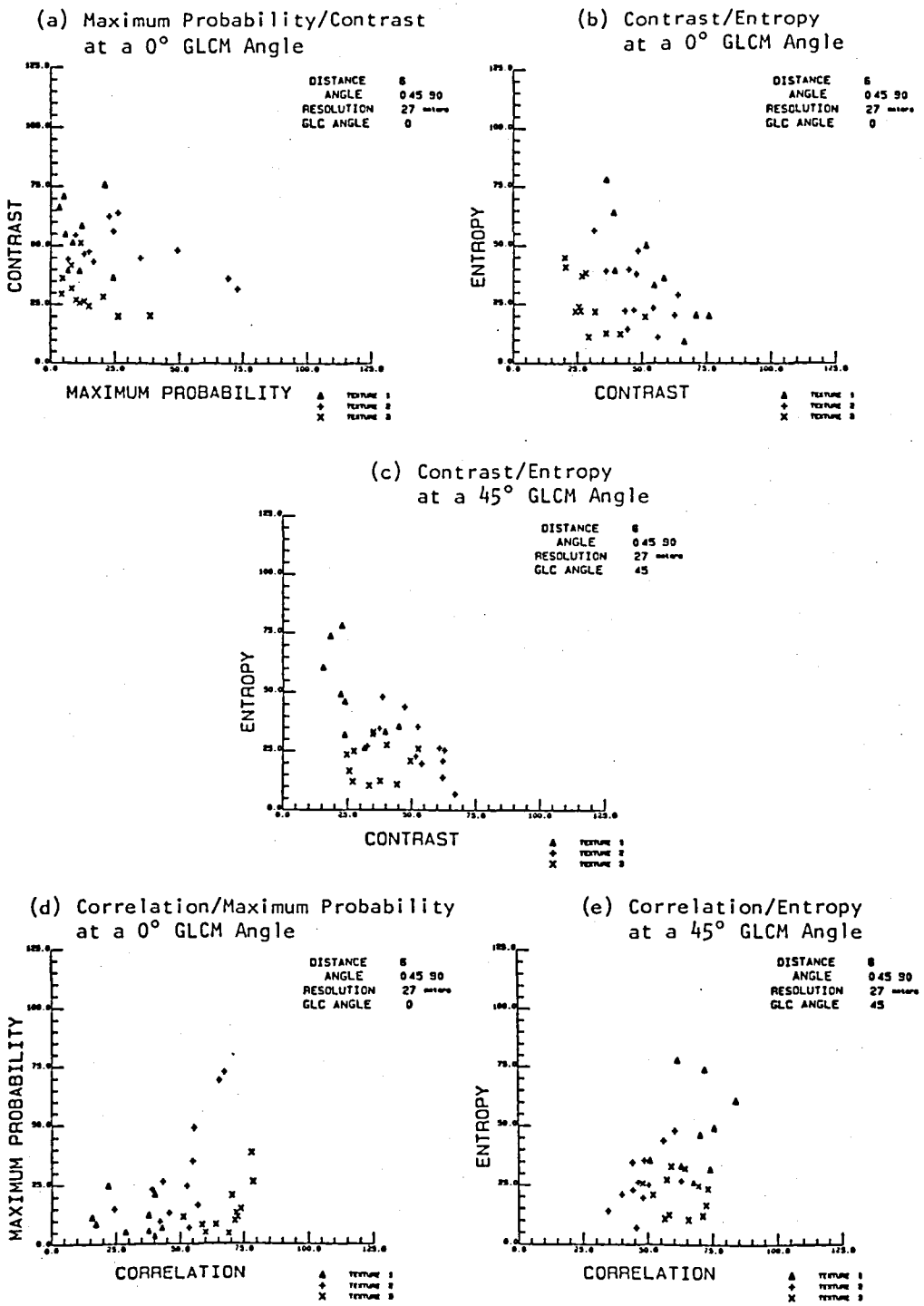


Figure 10. Scatter Diagrams for Texture Feature Pairs Combining All Flight Paths.

	GEOLOGY	TOPOGRAPHY
T_5	Rocks of the Lower Pennsylvanian Consisting of Alternating Beds of Sandstone and Shale with a Few Beds of Coal.	Mountains and Ridges with Steep Slopes and a Maximum Relief of of About 1500 Feet.
T_4	Rocks of the Lower to Middle Ordovician Consisting Primarily of Dolomite and Cherty Dolomite with Some Beds of Limestone, Shale, and Sandstone.	Rolling Hills and Several Ridges with a Maximum Relief of 200-300 Feet.
T_1	See T_4	Area of Overall Low Relief but with Many Small Hills that are Separated by Several Creeks and Streams.
T_3	Central Region (See T_4) Flanked on Either Side by Rocks of the Upper Part of the Middle Cambrian in Beds of Dolomite, Limestone, and Slate.	Rolling Hills and Elongated Ridges Separated by a Trellis Drainage Pattern and Having a Maximum Relief of About 500 Feet.
T_2	Rocks of the Upper Pre-Cambrian Consisting Primarily of Metasediments.	Mountains and Hills with Steep Slopes and a Maximum Relief of About 1000 Feet.

Table 1. Geology and Topography of the Tennessee Test Area

APPENDIX

Page intentionally left blank

AGENDA

Wednesday, June 1:

8:30 - 9:30	Program Overview Howard G. Hogg, Acting Branch Chief, Earth Resources Branch, Earth Science and Applications Division, NASA Headquarters, Washington D. C. Dr. R. P. Heydorn, Science Manager, Fundamental Research Program: MPRIA, NASA/Johnson Space Center, Houston, Texas
9:30 - 10:30	Keynote Address - Robert M. Haralick, Professor of Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia "Computer Vision For Remotely Sensed Data"
	Math/Stat: First Session
10:30 - 11:15	R. P. Heydorn and Rehka Basu, NASA/JSC "Estimating Location Parameters in a Mixture Model"
11:15 - 12:00	David Scott, Rice University "Multivariable Density Estimation and Remote Sensing"
12:00 - 1:30	Lunch
	Math/Stat: Second Session
1:30 - 2:15	Manouher Naraghi, Jet Propulsion Lab "Random Field Models for Use in Scene Segmentation"
2:15 - 3:00	Charles Peters and H. P. Decell, Jr., University of Houston "Mixture Models for Dependent Observations"
3:00 - 3:30	Break
3:30 - 4:15	Carl Morris and Hubert Kostal, University of Texas at Austin "An Empirical Bayes Approach to Spatial Estimation"
4:15 - 5:00	L. F. Guseman, Jr., and L. Schumaker, TAMU "Spline Classification Methods"
6:00 - 10:00	Social Hour and Banquet, Gilruth Center

Thursday, June 2:

8:30 - 8:45	Announcements
	Math/Stat: Third Session
8:45 - 10:00	Emanuel Parzen, TAMU "Quantile Data Analysis of Image Data"

10:00 - 10:15	Break
10:15 - 10:45	W. B. Smith and Eugene Shine, TAMU "Discrimination Relative to Measures of Non-Normality"
10:45 - 11:15	H. J. Newton, TAMU "Repeated Measures Analysis of Image Data"
11:15 - 12:00	Tae H. Joo and Daniel N. Held, Jet Propulsion Lab "SAR Speckle Noise Reduction Using Wiener Filter"
12:00 - 1:30	Lunch

Pattern Recognition: First Session

1:30 - 2:15	Larry S. Davis, Fu-pei Hu, Les Kitchen and Vincent Hwang, University of Maryland "Image Matching Using Hough Transforms"
2:15 - 3:00	Laveen N. Kanal, LNK Corporation "Subpixel Registration Accuracy and Modelling"
3:00 - 3:30	Break
3:30 - 4:15	E. M. Mikhail and F. C. Paderes, Jr., Purdue University "Simulation Aspects for the Study of Rectification of Satellite Scanner Data"
4:15 - 5:00	David Dow, National Space Technology Labs "Progress in the Scene-to-Map Registration Investigation"

Friday, June 3:

8:30 - 8:45	Announcements

Pattern Recognition: Second Session

8:45 - 9:30	Alan Strahler, Waldo Tobler and Curtis Woodcock, Hunter College "Relating Spatial Patterns in Image Data to Scene Characteristics"
9:30 - 10:15	Grahame Smith, SRI International "Shape from Shading: An Assessment"
10:15 - 10:30	Break
10:30 - 11:15	K. S. Shanmugan, University of Kansas "Power Spectral Density of Markov Texture Fields"
11:15 - 12:00	Discussion Session

LIST OF ATTENDEES

Paul E. Anuta, Purdue University

Kenneth Baker, NASA/JSC
T. C. Baker, Lockheed EMSCO
Rehka Basu, NASA/JSC
Gary Breaux, Texas A&M University
Kristine Butera, National Space Technology Laboratories

Don H. Card, NASA/Ames Research Center
R. B. Cate, Lockheed EMSCO
Raj S. Chhikara, Lockheed EMSCO
Carolyn A. Clark, Lockheed EMSCO

L. S. Davis, University of Maryland
H. P. Decell, Jr., University of Houston
David D. Dow, National Space Technology Laboratories

Jon Erickson, NASA/JSC

Al Feiveson, NASA/JSC
Mary C. Ferguson, NASA/JSC

L. F. Guseman, Jr., Texas A&M University

Forrest Hall, NASA/JSC
Robert Haralick, Virginia Polytechnic Inst. & State University
R. P. Heydorn, NASA/JSC
Howard Hogg, NASA Headquarters
Glen Houston, NASA/JSC

Tae H. Joo, Jet Propulsion Laboratory
David L. B. Jupp, Hunter College

Laveen N. Kanal, LNK Corporation
Hubert Kostal, University of Texas at Austin
Phyllis Krauss, Texas A&M University

Richard Latty, NASA Goddard/University of Maryland
Kent Lennington, Lockheed EMSCO
Gerard Livingston, Lockheed EMSCO
James Lundgren, Lockheed EMSCO

R. B. MacDonald, NASA/JSC
Jane Malin, Lockheed EMSCO
Anne Marie McAndrew, NASA/JSC
Naresh Mehta, Lockheed EMSCO
Michael Merickel, Lockheed EMSCO
E. M. Mikhail, Purdue University
Robert Mohler, Lockheed EMSCO
Carl Morris, University of Texas at Austin

Manouher Naraghi, Jet Propulsion Laboratory
H. J. Newton, Texas A&M University

Pat Odell, University of Texas at Dallas

Fidel C. Paderes, Jr., Purdue University
Emanuel Parzen, Texas A&M University
B. C. Peters, Jr., University of Houston
Brooks Pollock, Lockheed EMSCO

D. B. Ramey, Lockheed EMSCO
William E. Rice, NASA/JSC

Donna Scholtz, EROS Data Center
David Scott, Rice University
K. S. Shanmugan, University of Kansas
Sylvia Shen, Lockheed EMSCO
Grahame Smith, SRI International
H. G. Smith, Lockheed EMSCO
W. B. Smith, Texas A&M University
Charles Sorensen, Lockheed EMSCO
Mickey Steib, NASA/JSC
Alan H. Strahler, Hunter College

Shelby Tilford, NASA Headquarters
Waldo R. Tobler, University of California at Santa Barbara
M. H. Trenchard, Lockheed EMSCO
M. C. Trichel, NASA/JSC

Steven Wharton, NASA/GSFC
Curtis Woodcock, Hunter College

End of Document