CR-171819

(E85-10056 NASA-CR-171819) PROCEEDINGS OF
THE SECOND ANNUAL SYMPOSIUM ON MATHEMATICAL
PATTERN RECOGNITION AND IMAGE ANALYSIS
PROGRAM Final Report, 16 Jul. 1983 - 15
Jul. 1984 (Texas A&M Univ.) 595 p

G3/43

N85-16251
THRU
N85-16267
Unclas
00056

c. 1

Proceedings
# of the
Second Annual

E85-10056

NASA

Symposium
on
Mathematical Pattern Recognition
& Image Analysis

Original photography may be purchased
from EROS Data Center
Sioux Falls, SD 57198

June 6-8, 1984
Johnson Space Center
Houston, Texas

FINAL REPORT

"MATHEMATICAL PATTERN RECOGNITION AND IMAGE ANALYSIS PROGRAM"

Contract NAS 9-16664

July 16, 1983 - July 15, 1984

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 - L. F. Guseman, Jr. iii

Papers Presented by Principal Investigators:

Estimating Location Parameters in a Mixture
R. P. Heydorn and M. V. Martin 1

Nonparametric Analysis of Minnesota Spruce and Aspen Tree Data and Landsat Data
David W. Scott and Rodney Jee 27

Texture Classification Using Autoregressive Filtering
Wayne M. Lawton and Meemong Lee. 51

Bayesian Estimation of Normal Mixture Parameters
Charles Peters 101

Spatial Estimation From Remotely Sensed Data Via Empirical Bayes Models
Joe R. Hill, David V. Hinkley, Hubert Kostal, and Carl N. Morris 115

Multivariate Spline Methods in Surface Fitting
L. F. Guseman, Jr. and L. L. Schumaker 137

Autoregressive Spectral Estimation for Two Dimensional Time Series
H. Joseph Newton and William B. Smith. 165

Exploring the Use of Structural Models to Improve Remote-Sensing Agricultural Estimates
Richard F. Gunst and Mani Y. Lakshminarayanan. 175

Calibration or Inverse Regression: Which is Appropriate for Crop Surveys Using Landsat Data?
R. S. Chhikara and A. G. Houston 205

Evidence Accumulation for Spatial Reasoning
Takashi Matsuyama, Vincent Shang-Shoug Hwang, and Larry S. Davis 245

Power Spectral Density of Markov Texture Fields
K. Sam Shanmugan and J. C. Holtzman. 303

Rectification of Single and Multiple Frames of Satellite Scanner Imagery Using Points and Edges as Control
Fidel C. Paderes, Jr., Edward M. Mikhail, and Wolfgang Förstner 309

The Influence of the Number of Grounc Control Points on the Scene-to-Map Registration Accuracy
David D. Dow 401

Image Variance and Spatial Structure in Remotely Sensed Scenes
Curtis E. Woodcock and Alan H. Strahler. 427

Image-to-Image Correspondence: Linear-Structure Matching
Grahame B. Smith and Helen C. Wolf 467

Analysis of Subpixel Registration
Carlos A. Berenstein, Laveen N. Kanal, David Lavine, Eric C. Olson, and Eric Slud 489

PREFACE

This volume comprises the Proceedings of the Symposium on Mathematical Pattern Recognition and Image Analysis (MPRIA) held June 6-8, 1984, at the NASA/Johnson Space Center, Houston, Texas.

The Symposium was initiated with a brief Program Overview presented by Drs. M. Kristine Butera, NASA Headquarters, and R. P. Heydorn, NASA/JSC.

The sixteen papers of the Proceedings reflect the results of various research efforts initiated during FY 1983 as part of NASA's Remote Sensing Research Program. Six of the papers prsent results from the four research efforts carried out by the following NASA principal investigators:

R. P. Heydorn - NASA/Johnson Space Center

A. G. Houston - NASA/Johnson Space Center

David D. Dow - National Space Technology Laboratories

Meemong Lee - Jet Propulsion Laboratory

The remaining papers present second-year results from ten of 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

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.

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

D1

ESTIMATING LOCATION PARAMETERS IN A MIXTURE

R. P. Heydorn, NASA Johnson Space Center
M. V. Martin, Lockheed Engineering and Management Services Company, Inc.

ABSTRACT

This paper considers the problem of estimating the parameters in a finite mixture of the form $h(x) = \sum_{J=1}^{M} \lambda_j f(x - \zeta_j)$ where ζ_j , $j = 1, 2, . . . , M$ are location parameters. The approach is based on an integral equation formulation of the form $h_t (x) = \int_a^b f(x-y) g_t(y)dy$ where h_t is a smoothed version of h and g_t is a prior function that tends to be concentrated on the translation values. A solution for g_t that uses the method of regularization and one based on a posterior operator approach is considered. Numerical simulations are presented to bring out some of the estimation and numerical problems of these approaches.

INTRODUCTION

We begin with a formulation of the mixture problem as essentially given by Teicher [1]. Let $F = \{f_\zeta : \zeta \varepsilon \mathbb{R}^N\}$ 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 the mixture density h as

$$(1) \qquad h = \int f_\zeta \, dG(\zeta)$$

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 H is identifiable.

In our case we will be interested in the so called finite mixture. For the finite mixture the measure induced by G assigns positive probability to only a finite number of ζ-values. Accordingly, the finite mixture can be written as

$$(2) \qquad h = \sum_{J=1}^{M} \lambda_j f_{\zeta_j}$$

where $0 \le \lambda_j \le 1$ and $\sum \lambda_j = 1$. As we will discuss shortly, this representation of mixtures appears to be most useful in remote sensing applications.

As described by equation (2), the finite mixture model is a parametric model and therefore to specify the model one must estimate or otherwise determine the parameters M, λ_j, ζ_j for j = 1, 2, . . ., M. When H is identifiable these parameters are uniquely determined and therefore one should be able to estimate them from just the random observations which have density h. Note that since the λ_j's can be considered as the prior probabilities and are being estimated from "the data" (i.e., the observations that have density, h), this is a form of the Empirical Bayes Problem as discussed by Robbins [2].

In this paper we will be concerned with the case where F is a translation family; that is, $f_{\zeta_j}(x) = f(x - \zeta_j)$. Since the family F is now defined on the translates of a given function we will denote it by F_f. Yakowitz et al. [3] and Heydorn et al. [4] have shown that any translation family leads to an identifiable mixture.

We have chosen to restrict ourselves to the translation family for two reasons. First of all remotely sensed measurements of radiance values from a given class of materials on the Earth can often be reasonably well represented by some translation family. And, therefore, even though we consider this work as being just in the early stages, some applications appear to be possible. The other reason is simply that by studying the translation case, we believe that considerable insight into the more general problem can be derived.

Our previous work (cf. reference [4]) addressed the finite mixture model for the case where the mixture density h is known. In that case a somewhat more general version of the translation family was treated in the sense that certain (nuisance) parameters whose values were unknown were allowed. The approach was based on a theorem of Caratheodory and made use of a constructive proof of that theorem due to Szegö (as described in Grenander et al. [5]). When h is numerically well determined we found that M and the translation parameters ζ_j, j = 1, 2, . ., M could be computed in many cases. When h is not known and must be estimated from a moderately small number of random observations, then the variance in estimates of M and ζ_j, j = 1, 2, . ., M is very large. Thus for the small to moderately large sample size cases, we have chosen to take a fresh look at the problem.

In this paper we will consider an "integral equation" formulation of the mixture problem and discuss two approaches for obtaining a numerical solution. One approach is based on the regularization method of Tikhonov et al. [6]. Wahba [7] discusses this method in connection with density estimation problems; Rice et al., [16] in connection with estimating derivatives and deconvolution of densities; and Medgyessy [8] in connection with mixture problems. The other approach is based on the formulation of a posterior operator that, in a certain sense, avoids the inversion problem implicit in the regularization method.

Applications to Remote Sensing

The finite mixture model of equation (2) provides us with a representation for remotely sensed measurements that can be applied toward the solution of several application problems. We mention three such problems.

One application of remotely sensed data deals with inventories of selected materials on the surface of the Earth (for example, the determination of the acres of wheat, the acres of conifer trees, or the square miles of water). MacDonald et al. [9], for example, discussed the use of remotely sensed data for inventorying wheat. A popular approach for this application is simply to classify each measurement observation into one of M given material classes and count the classifications to determine the proportion of the area surveyed that belongs to a given class. This method only works well, in general, where classification errors are small. Large classification error can lead to biases in the inventory. If in the mixture model of equation (2), a given parameter ζ_j can be uniquely associated with a given material class, then λ_j is the proportion of that material when this association can be made. In other words, given the knowledge that each material class on the ground can be represented by a member of some known family F, that leads to identifiable mixtures, one should be able to unbiasedly determine the proportion of each material class. We point out, however, that the mixture model does not directly give us a way of assigning a material class name to each λ_j-value. This has been called the "labeling problem." One application of the finite mixture model to crop inventories and an approach to the labeling problem is discussed by Lennington et al. [10].

Another application of remotely sensed data is concerned with the determination of certain properties of materials on the Earth's surface. Goel et al. [11], for example considers the problem of solving for the variables in the Suits [12] vegetation canopy model from the light reflected (more specifically reflectance) from the canopy at several view angles. These variables include the leaf transmittance, the leaf reflectance, the soil reflectance, the projections on the horizontal and vertical planes of the average leaf area per unit volume. If we let x_k, k = 1, 2, . ., K represent reflectances and y_k, k = 1, 2, . ., K the canopy variables, then Goel addresses the problem of solving for the canopy variables given the equations $x_k = q_k\ (y_1, y_2, . ., y_k)$, k = 1, 2, . ., K. If x is the vector of reflectance variables, y the vector of canopy variables, and T is a 1-1 transformation determined by the above equations then x = T(y). Under this transformation the probability that the canopy variable values lie in a given set, say A, is given for the kth species to be $\int_{T(A)} f_{\zeta_k}(x)\ dx$ where f_{ζ_k} is determined from the mixture model. In this approach the transformation, T, is never inverted. Inversion is generally a difficult numerical operation for the Suits model, for example.

The final application we have in mind is classification. The classification function, ϕ, is a function that assigns each measurement x to one, and only one, of M possible classes. If each parameter ζ_j, j = 1, 2, . ., M in the mixture model in equation (2) is uniquely related to one of these classes then the Bayes classification function becomes

$$\phi(x) = k \text{ iff } \lambda_k f_{\zeta_k}(x) \geq \max_j \lambda_j f_{\zeta_j}(x)$$

If one is searching for members of a particular class, say, k, then a map related to a class map could be obtained by observing $\lambda_k f_{\zeta_k}(x)/\sum_{J=1}^{M} \lambda_j f_{\zeta_j}(x)$ which is the posterior probability that x is an observation from class k.

An Integral Equation Formulation of the Mixture

The formulation of the finite mixture given by equation (2) treats the quantities of interest M, λ_j, ζ_j, j = 1, 2, . ., M as parameters. While in many cases the estimation of the parameters λ_j, ζ_j, j = 1, 2, . ., M can be done quite well, the estimation of M appears to be a more difficult problem, as may be suggested by the fact that fewer papers address this problem. Part of the difficulty seems to be that the parametric formulation tends to isolate each parameter so that a separate estimator is needed for each parameter. The formulation given by equation (1) gets around this difficulty somewhat by "bundling up" these estimation problems into one problem, viz, the estimation of a function G.

If G was an absolutely continuous distribution function, not a step (Heavyside) function as in the case of a finite mixture, then G would define a density function g and we could write

$$h(x) = \int_{\mathbb{R}^N} f(x-y)g(y)dy$$

where $f \varepsilon \mathcal{F}_f$. This gives an integral equation representation for the mixture. For a finite mixture, however, g is a delta function, or more correctly a singular generalized function (as discussed, e.g., by Gel'fond et al. [13]) and so the integral equation representation is not correct. However, when $f \varepsilon \mathcal{F}_f$ we can consider a "smoothed" version of the finite mixture and thereby use the integral equation representation. We now discuss this approach in the scalor setting (i.e., $\mathbb{R}^N = \mathbb{R}$).

Following the definition of a generalized function as given in Gel'fond et al. [13], let (g,ϕ) denote a linear functional defined by g where $\phi \varepsilon D$. Here D is a family of functions, each member of which has bounded support and is continuously differentiable. For the finite mixture case we have $(g, \phi) = \sum_{j=1}^{M} \lambda_j \phi(\zeta_j)$. Since our mixture is a convolution between the kernel function f and g, we have (denoting convolution by "*")

$$(f*g, \phi) = (\sum_{j=1}^{M} \lambda_j f_{\zeta_j}, \phi) = (h,\phi)$$

where $f_{\zeta_j}(x) = f(x-\zeta_j)$. And smoothing with a function $t \varepsilon L_1$,

$$(t*h,\phi) = (t*(f*g),\phi) = (f*(t*g),\phi)$$

Since $(t*g,\phi)$ is a regular generalized function we have letting $h_t = t*h$ and $g_t = t*g$

$$h_t(x) = \int_a^b f(x-y)\, g_t\, (y)dy \tag{3}$$

where $[a,b]$ contains the domain of g_t.

We can choose the support of t to be small so that this integral equation formulation (equation (3)) can be a good approximation to the finite mixture. For example if we choose

$$t(x) = \frac{15}{16}\sqrt{n}\,(1- nx^2)^2,\ |\ \sqrt{n}\, x\ |\ \le 1 \tag{4}$$

and 0 otherwise, then $(t*h)\ (x) \to h(x)\ (n \to \infty)$ if x is any continuity point of h (cf. Bochner [14]). And since generalized function spaces are complete and H is assumed to be identifiable, $\lim\limits_{n \to \infty} (t*h,\phi) = (f*g,\phi)$.

Equation (3) expressed in terms of the operator $\mathbb{F}$ becomes $h_t = \mathbb{F}g_t$. Whenever f is continuous on $[a,b]$ (and therefore bounded), $\mathbb{F}$ is a compact linear operator. This means (cf. Kolmogorov et al. [15]) that $\mathbb{F}^{-1}$ cannot be bounded on an infinite dimension space and hence $\mathbb{F}^{-1}$ would not be a continuous linear operator. Thus, when we attempt to estimate h_t by $\hat{h}_t = h_t + \varepsilon_t$ where ε_t is an error function, $\mathbb{F}^{-1}\hat{h}_t$ could be grossly different from g_t (as measured by the supremum norm).

One approach for solving this problem is to use the regularization method of Tikhonov et al. [6]. In this approach one defines the functional

$$S_\alpha = ||\,\mathbb{F}g_t - h_t||_2 + \alpha\ ||\left(\sum_{J=1}^{n+1}\left(g_t^{(j)}\right)^2\right)^{\frac{1}{2}}||_2 \tag{5}$$

where g_t is assumed to be differentiable of order n. Here α is called the regularization parameter. The resulting solutions say $g_{t,\alpha}$ which are obtained by minimizing S_α are approximations to g_t.

If we replace h_t by some approximation $\hat{h}_t$ then as our approximations to h_t become successively better we can successively decrease α so that $g_{t,\alpha}$ approaches (in some sense) g_t. Wahba [7], for example, considers such an estimation problem and uses the method of cross validation to pick α-values.

Rather than considering the operator $\mathbb{F}$ whose inverse is discontinuous, it is possible in some cases to derive another linear compact operator from the kernel f, say $\mathbb{P}$ so that $\mathbb{P}h = g_t$. We now consider this approach for solving for g_t.

Assuming for a moment that we can take the Fourier transform in equation (3), and letting " ~ " denote the corresponding transformed function we have (for $\omega = 2\pi\nu$)

$$\tilde{h}_t(\omega) = \tilde{f}(\omega)\, \tilde{g}_t(\omega)$$

and,

$$\frac{\tilde{t}(\omega)}{\tilde{f}(\omega)} \cdot \tilde{h}(\omega) = \tilde{g}_t(\omega)$$

If $\tilde{t}(\omega)/\tilde{f}(\omega)$ has an inverse Fourier transform, say t_f then

$$\int_{a'}^{b'} t_f(x-y)h(y)dy = g_t(x)$$

and if further t_f is continuous on the bounded interval [a', b'], then the linear operator $\mathbb{P}$ defined by the kernel t_f is a compact operator as was $\mathbb{F}$. We refer to $\mathbb{P}$ as the posterior operator since it operates on h to produce a prior function.

If we consider, for example, the function t of equation (4), then since $(t*g,\phi) = (g,t*\phi)$ and $(t*\phi)\to\phi(x)$ $(n\to\infty)$ we would have $\lim(t_f * h\ ,\phi) = (g,\phi)$. Thus, by properly choosing t we can obtain a good approximation to g.

An example of a case where $\tilde{t}(\omega)/\tilde{f}(\omega)$ has an inverse Fourier transform comes from the gamma family of densities where $f(x) = \frac{1}{n!\gamma^{n+1}} x^n e^{-x/\gamma}$ for $x \geq 0$ and is 0 otherwise. Later we will consider some numerical examples of mixtures of gammas. The Fourier transform in this case is $(1 + i\omega\gamma)^{n+1}\tilde{t}(\omega\gamma)$, and hence, if t has n+1 derivatives in some interval [a,b] and $t^{(n+1)}(a) = t^{(n+1)}(b) = 0$ then the kernel function t_f is of the form $t_f(x) = (1 + \binom{n+1}{1}\gamma D + \binom{n+1}{2}\gamma^2 D^2 + \gamma^{n+1}D^{n+1})t(x)$ where D is the standard derivative operator. Notice that the above gamma function does not have an (n+1)st derivative at $x = 0$. The role of the function t in this case is to smooth the gamma function so that the (n+1)st derivative exists.

Numerical Solutions

To understand numerical estimation problems associated with both the regularization and the posterior operator methods discussed above, we conducted simulation studies using translations of beta and gamma distributions. One of the reasons for choosing these families is that they have positive support and are skewed. Both of these properties are also found in typical densities of remotely sensed measurements.

For the regularization approach we chose mixtures of translates of beta densities. Our mixture density, h, in this case was

$$h(x) = \sum_{J=1}^{M} \lambda_j\, f(x - \zeta_j)$$

where

$$f(x) = \begin{cases} 12\ (1-x)^2 x, & 0 \leq x \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

In equation (3) h_t, which is an approximation to h, is of the form

$$h_t(x) = E\ (t(x - X))$$

where X is a random variable distributed according to the mixture h. Given a set of iid random variables $X_1, X_2, \ldots, X_n$ each distributed as X, we can estimate h_t as

$$\hat{h}_t(x) = \frac{1}{n} \sum_{J=1}^{M} t(x - X_j)$$

For our simulation we chose t to be a third order B-spline, i.e.,

$$t\ (x) = \begin{cases} \frac{1}{2}(\frac{x}{\Delta})^2, \ 0 \le x \le \Delta \\ \frac{(-2(\frac{x}{\Delta})^2 + 6\ \frac{x}{\Delta} - 3)}{2}, \ \Delta \le x \le 2\Delta \\ \frac{1}{2}(3 - \frac{x}{\Delta})^2, \ 2\Delta \le x \le 3\Delta \end{cases}$$

Moreover, we used the following B-spline approximation to g_t, viz,

$$\hat{g}_t(x) = \sum_{\ell=0}^{L+1} c_\ell t(x - \ell\Delta) \tag{6}$$

Finally, we used a modification of the regularization formulation of equation (5); and that is, we considered only the second derivative of g_t in the constraint rather than all of its derivatives. This form of the regularization problem is considered by Rice et al. [16]. In that paper bias and variance expressions for the solution function (which is also approximated by B-splines) are derived.

To obtain $\hat{g}_t$ we have to solve for the c-coefficients in equation (6). Letting these solution coefficients be $\hat{c}_\ell$, $\ell = 0, 1, 2, \ldots, L+1$ we have from equation (5) with the above approximations,

$$\hat{\underline{c}} = (A'A + N\alpha B)^{-1}A'\underline{h} \tag{7}$$

where

$$A = \{a(j,\ell)\}$$

$$a(j,\ell) = \int_a^b f(x_j - y)\, t(y - \ell\Delta)\, dy$$

$$B = \frac{1}{\Delta^3}\begin{pmatrix} 4 & -4 & 1 & 0 & 0 & 0 & . & . & . & 0 \\ -4 & 4 & -4 & 1 & 0 & 0 & . & . & . & 0 \\ 1 & -4 & 4 & -4 & 1 & 0 & . & . & . & 0 \\ 0 & 1 & -4 & 4 & -4 & 0 & . & . & . & 0 \\ . & & & & & & & & & \\ . & & & & & & & & & \\ 0 & . & . & . & . & . & . & 1 & -4 & 4 \end{pmatrix}$$

$\underline{h}'$ = $(h(x_1), h(x_2), \ldots, h(x_n))$

$\hat{\underline{c}}'$ = $(\hat{c}_0, \hat{c}_1, \ldots, \hat{c}_{L+1})$

n = number of random observations from the mixture

N = number of points at which g is estimated (this is the range of the "j" index)

L+2 = number of B-splines (this is also the range of index "ℓ")

Δ = interval spacing of the B-splines

It is seen from equation (7) that $\hat{\underline{c}}$ is a ridge - regression-like estimator which minimizes the variance of each $\hat{c}$, by adding bias. The size of the bias is influenced by the regularization parameter α.

Figures 1, 2, and 3 are examples where a mixture of two beta densities, with equal mixing, are considered. In these figures the spacing between the B-splines was .05. We see from figure 1 that when $\zeta_2 - \zeta_1 = .05$ then the betas are too close to distinguish the fact that two components are present. When $\zeta_2 - \zeta_1 = .1$ as in figure 2 we begin to see two modes to the prior function plus some oscillatory behavior; when $\zeta_2 - \zeta_1 = .15$ as in figure 3 the two modes are very distinct. The major peaks of the graph of the prior function, g_t appear to be located to within .02 of the translation values (ζ_1 and ζ_2) in figures 2 and 3.

For the posterior operator approach we chose mixtures of translates of gamma distributions. In this case

$$\hat{g}_t \ (x_j) = \frac{1}{n} \sum_{l=1}^{n} t_f(x_j - X_l)$$

where now the random variables X_l, $l = 1, 2, \ldots, n$ each have density h and the components of h are of the form

$$f(x) = \begin{cases} \frac{1}{\gamma^2} xe^{-x/\gamma}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

The posterior kernel, t_f, is of the form

$$t_f(x) = t_4(x) + \frac{2\gamma}{\Delta}(t_3(x) - t_3(x - \Delta))$$

$$+ \frac{\gamma^2}{\Delta^2} (t_2(x) - 2t_2(x - \Delta) + t_2(x - 2\Delta))$$

where the t_k, $k = 2,3,4$ are 2nd, 3rd, and 4th order B-splines respectively, (t_3 is the same as the function t considered in the regularization method. The expression for t_4 and t_2 can be found in Schumaker [17]).

Figures 4 to 7 are example graphs for mixtures of two gamma densities equally mixed. In these cases the spacing between the B-splines was .1 rather than .05 as in the previous figures. Figure 4 shows that $\zeta_2 - \zeta_1 = .1$ is too small to distinguish the existence of two components to the mixture. At $\zeta_2 - \zeta_1 = .15$ we begin to see two components in figure 5. At $\zeta_2 - \zeta_1 = .2$ figure 6 shows two distinct components, and finally at $\zeta_2 - \zeta_1 = .35$, figure 7, the g_t function has two very distinct peaks. The graphs after figure 4 also show that the peaks in g_t occur at essentially the translation values.

Figures 8 and 9 are examples where the components are unequally mixed. In figure 9 we see that for $\lambda_1 = .1$ and $\lambda_2 = .9$ the first peak can get confused with the oscillatory behavior of the g_t functions.

Finally figure 10 shows a case where the number of random observations from the mixture was only 100. In the previous graphs 1000 observations were used. This suggests, perhaps, that much fewer that 1000 observations could have been used for these examples.

CONCLUDING REMARKS

Our purpose in these studies is to explore some of the estimation and numerical problems associated with solving for the prior function in a finite mixture. We chose to begin by considering mixtures of translates partially because convolving the mixture with some smooth function is the same as smoothing the prior function. This fact leads to an integral equation representation of the finite mixture. The solution of the resulting integral equation by either the regularization method or the posterior operator method leads to a graph of the prior function in which the number of components in the mixture can often be easily determined (at least visually) and the translation values can be reasonably well approximated. However, by smoothing the prior function, it now appears to be difficult to estimate the mixing proportions, $\lambda_j, j = 1,2,. . M$. When, for example, maximum likelihood methods are used to estimate these parameters, for a given M, the λ_j values are often easily estimated.

There are of course a number of numerical problems associated with these numerical approaches. The fact that the spline solution for the prior function tends to oscillate and can go negative is disturbing. We hope to examine some of these problems in future studies.

References

1. Teicher, H., Identifiability of finite mixtures. Annals of Math Stat. 34 (1963) 1265-1269.

2. Robbins, H., An empirical Bayes approach to statistics. Third Berkeley Symposium, 157-163.

3. Yakowitz, S.S., Spragins J.P., On the identifiability of finite mixtures. Annals of Math. Stat. 39 (1968) 209-214.

4. Heydorn R.P., Basu, R., Estimating location parameters in a mixture model. Proc. of the NASA Symposium on Mathematical Pattern Recognition and Image Analysis. (June 1983) 55-76.

5. Grenander, U., Szego G., Toeplitz forms and their applications. (Berkeley, University of California Press, 1958).

6. Tikhonov, A.N., Arsenin, V.Y., Solutions of ill-posed problems. (V.H. Winston and Sons, 1977).

7. Wahba, G., Practical approximate solutions to linear operator equations when the data are noisy. Siam J. Numerical Analysis, Vol. 14, No. 4, (Sept. 1977) 651-667.

8. Medgyessy, P., Decompositions and superpositions of density functions and discrete Distributions. (A. Halsted Press, 1977).

9. MacDonald, R.B., Hall, F.G., Global crop forecasting. Science 16, Vol. 208, May 1980, 670-679.

10. Lennington, R.V., Sorensen, C.T., Heydorn, R.P., A mixture model approach for estimating crop areas from landsat data. Remote Sensing of Environment, Vol. 14 (1984) 197-207.

11. Goel, N.S., Strebel, D.E., Thompson, R.L., Inversion of vegetation canopy reflectance models for estimating agronomic variables, II. Use of angle transforms and error analysis as illustrated by Suit's model. Remote Sensing of Environment, Vol. 14 (1984) 77-111.

12. Suits, G.H., The calculation of the directional reflectance of vegetative canopy. Remote Sensing of Environment, Vol. 2 (1972) 117-125.

13. Gel'fond, I.M., Shilov, G.B., Generalized functions. (Academic Press, Vol. 1, 1964).

14. Bochner, S. Harmonic analysis and the theory of probability. (University of California Press, 1960).

15. Kolmogorov, A.N., Fonin, S.V., Elements of the theory of functions and functional analysis. (Graylock Press, Vol. 1, 1957).

16. Rice, J., Rosenblatt, M., Smoothing splines: regression, derivatives, and deconvolution. Proc. of the NASA Workshop on Density Estimation and Function Smoothing, Texas A&M University, Mar. 11-13, 1982, 102-141.

17. Schumaker, L.L., Spline functions: basic theory. (John Wiley & Sons, 1981).

MIXTURE OF BETAS AND RESULTANT PRIOR FUNCTION

MIXTURE=* PRIOR=+

$\zeta_1 = .20$ $\zeta_2 = .25$

$\lambda_1 = .50$ $\lambda_2 = .50$

NUMBER OF POINTS PER DISTRIBUTION = 1000

α = 1x10**-6

Δ = .05

REGULARIZATION APPROACH

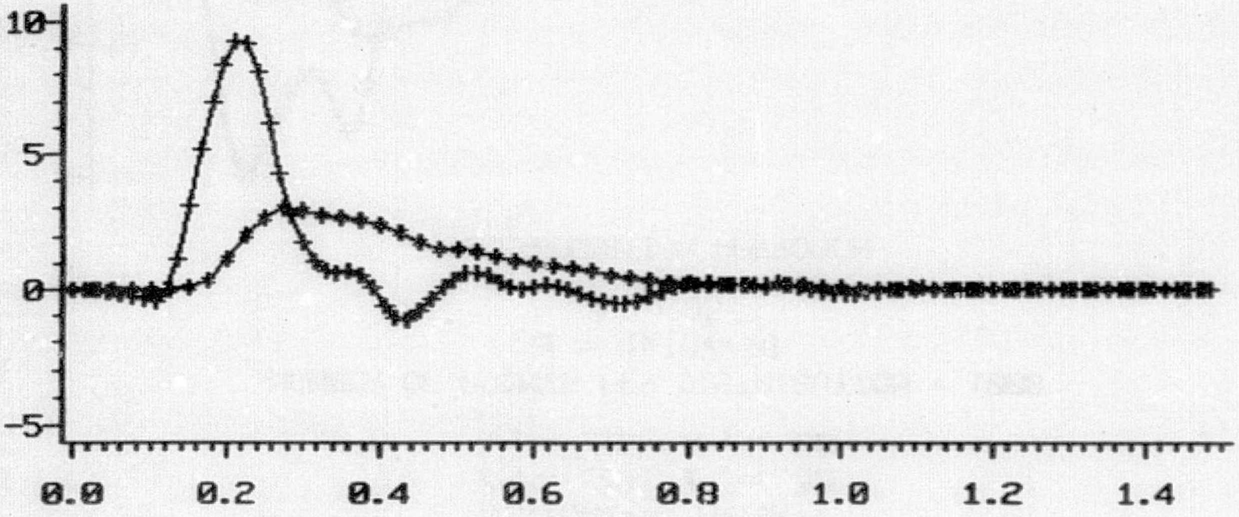


FIGURE 1

MIXTURE OF BETAS AND RESULTANT PRIOR FUNCTION

MIXTURE=* PRIOR=+

$\xi_1 = .20$ $\xi_2 = .30$

$\lambda_1 = .50$ $\lambda_2 = .50$

NUMBER OF POINTS PER DISTRIBUTION = 1000

α = 1x10**-6

Δ = .05

REGULARIZATION APPROACH

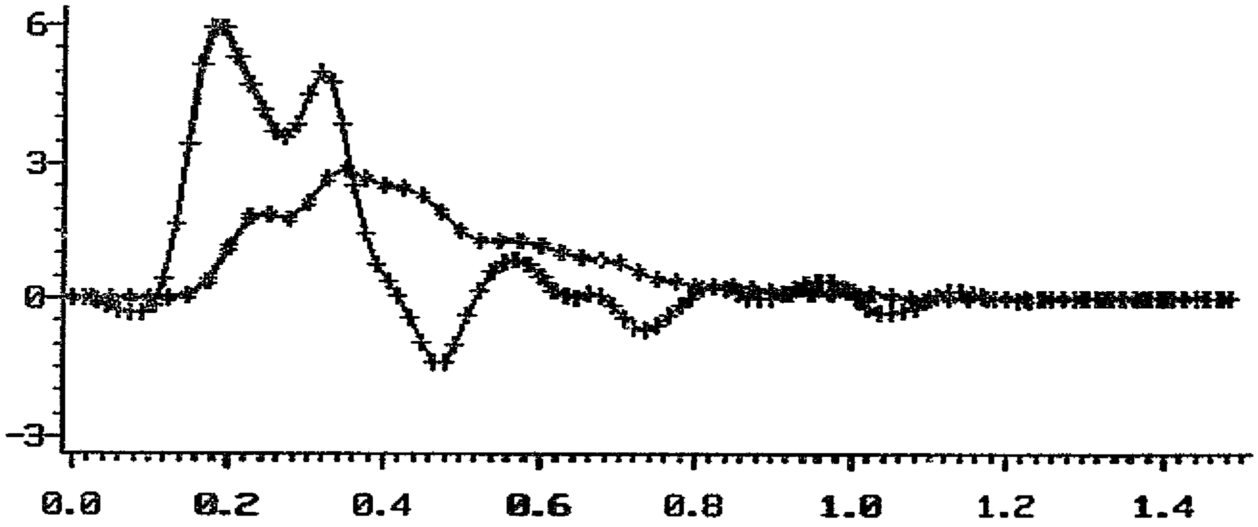


FIGURE 2

MIXTURE OF BETAS AND RESULTANT PRIOR FUNCTION

MIXTURE=* PRIOR=+

$\xi_1 = .20$ $\xi_2 = .35$

$\lambda_1 = .50$ $\lambda_2 = .50$

NUMBER OF POINTS PER DISTRIBUTION = 1000

$\alpha = 1\text{x}10{**}{-6}$

A = .05

REGULARIZATION APPROACH

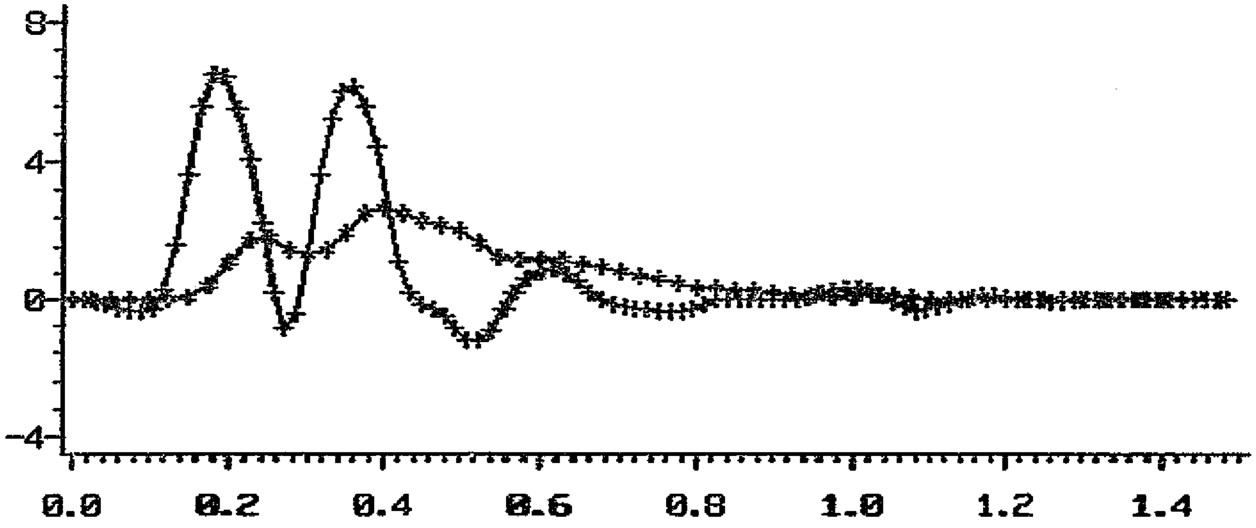


FIGURE 3

MIXTURE OF GAMMAS AND RESULTANT PRIOR FUNCTION

MIXTURE=* PRIOR=+

$\zeta_1 = .20 \quad \zeta_2 = .30$

$\lambda_1 = .50 \quad \lambda_2 = .50$

NUMBER OF POINTS PER DISTRIBUTION = 1000

$\gamma = 0.1$

$\Delta = 0.1$

POSTERIOR OPERATOR APPROACH

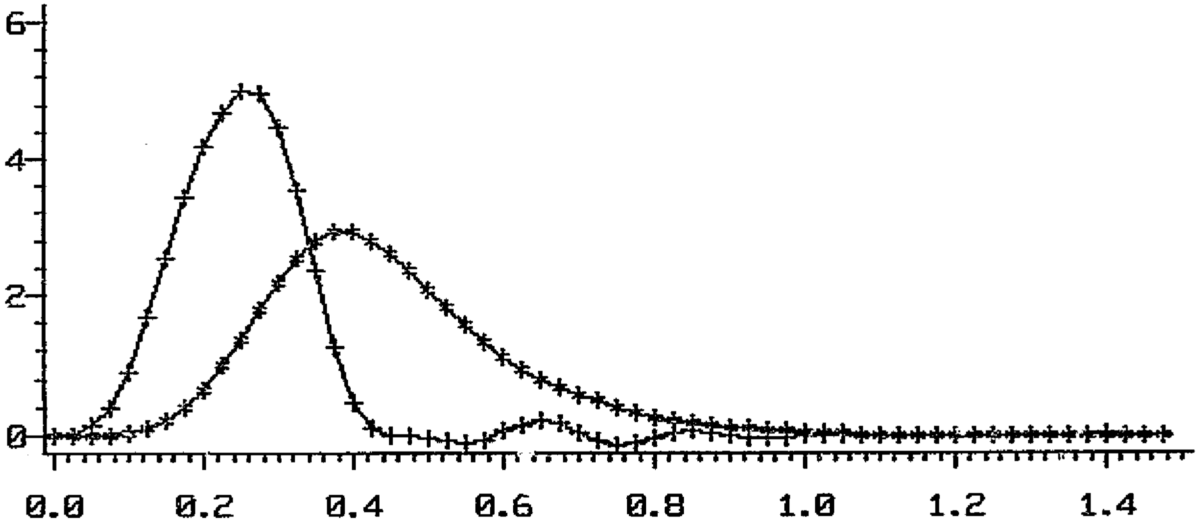


FIGURE 4

MIXTURE OF GAMMAS AND RESULTANT PRIOR FUNCTION

MIXTURE=* PRIOR=+

$\zeta_1 = .20 \quad \zeta_2 = .35$

$\lambda_1 = .50 \quad \lambda_2 = .50$

NUMBER OF POINTS PER DISTRIBUTION = 1000

$\gamma = 0.1$

$\Delta = 0.1$

POSTERIOR OPERATOR APPROACH

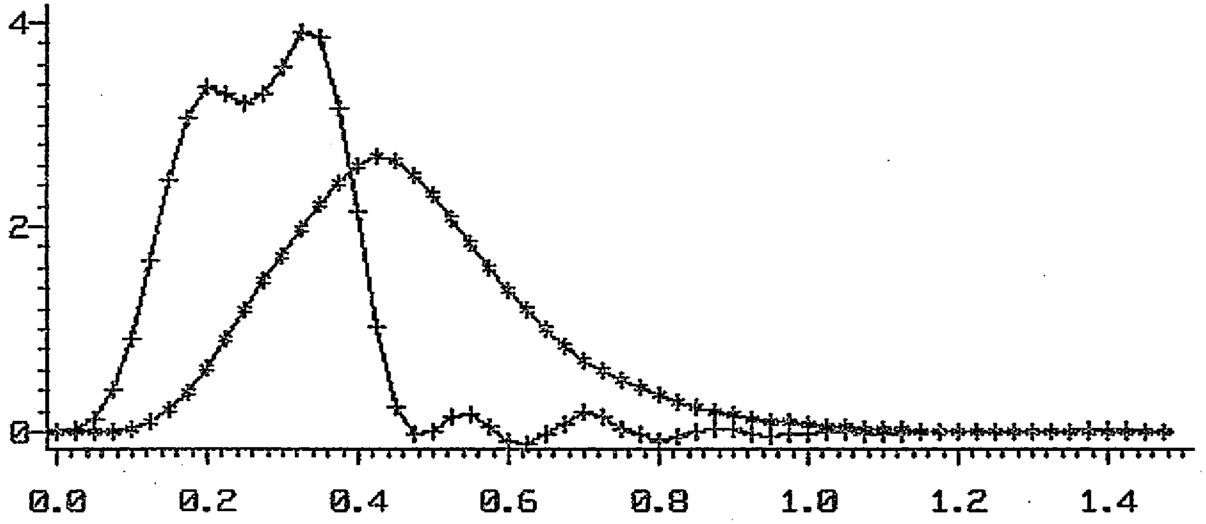


FIGURE 5

MIXTURE OF GAMMAS AND RESULTANT PRIOR FUNCTION

MIXTURE=* PRIOR=+

$\zeta_1 = .20$ $\zeta_2 = .40$

$\lambda_1 = .50$ $\lambda_2 = .50$

NUMBER OF POINTS PER DISTRIBUTION = 1000

$\gamma = 0.1$

$\Delta = 0.1$

POSTERIOR OPERATOR APPROACH

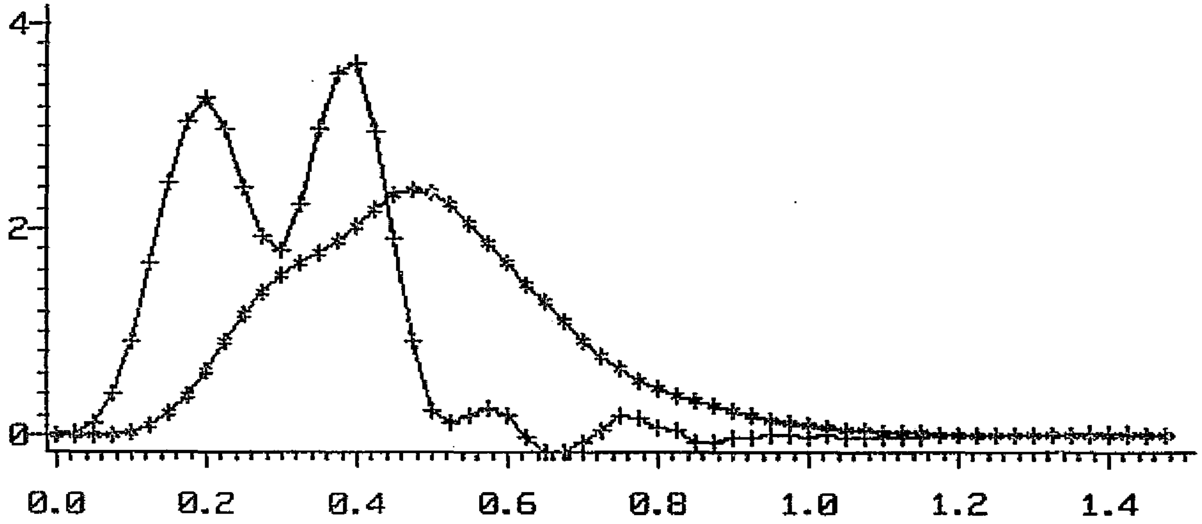


FIGURE 6

MIXTURE OF GAMMAS AND RESULTANT PRIOR FUNCTION

MIXTURE=* PRIOR=+

$\zeta_1 = .20 \quad \zeta_2 = .45$

$\lambda_1 = .50 \quad \lambda_2 = .50$

NUMBER OF POINTS PER DISTRIBUTION = 1000

$\gamma = 0.1$

$\Delta = 0.1$

POSTERIOR OPERATOR APPROACH

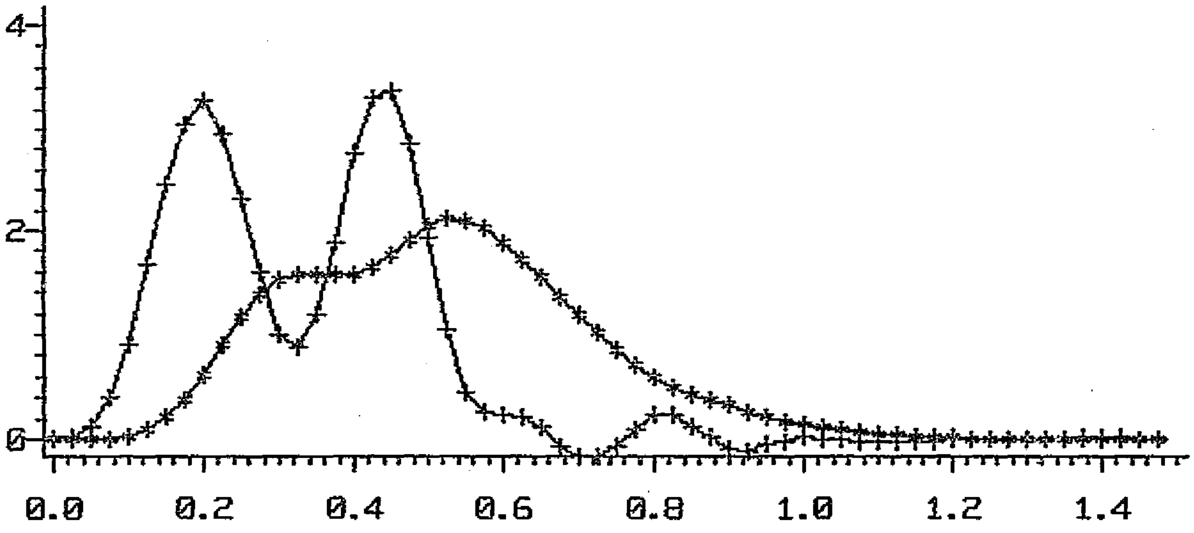


FIGURE 7

MIXTURE OF GAMMAS AND RESULTANT PRIOR FUNCTION

MIXTURE=* PRIOR=+

$\xi_1 = .20$ $\xi_2 = .55$

$\lambda_1 = .25$ $\lambda_2 = .75$

NUMBER OF POINTS PER DISTRIBUTION = 1000

$\gamma = 0.1$

$\Delta = 0.1$

POSTERIOR OPERATOR APPROACH

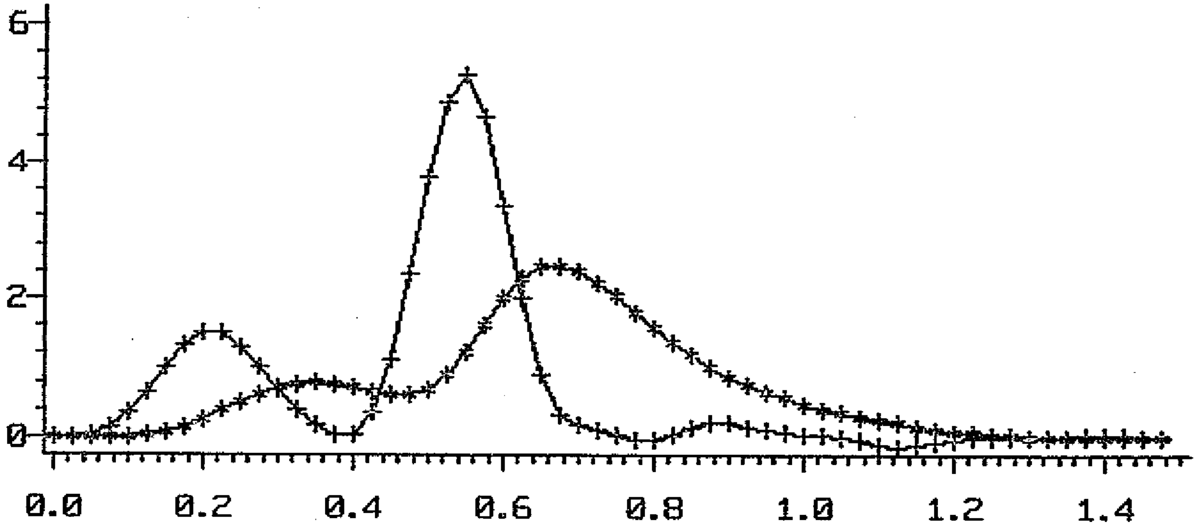


FIGURE 8

MIXTURE OF GAMMAS AND RESULTANT PRIOR FUNCTION

MIXTURE=* PRIOR=+

$\zeta_1 = .20$ $\zeta_2 = .55$

$\lambda_1 = .10$ $\lambda_2 = .90$

NUMBER OF POINTS PER DISTRIBUTION = 1000

$\gamma = 0.1$

$\Delta = 0.1$

POSTERIOR OPERATOR APPROACH

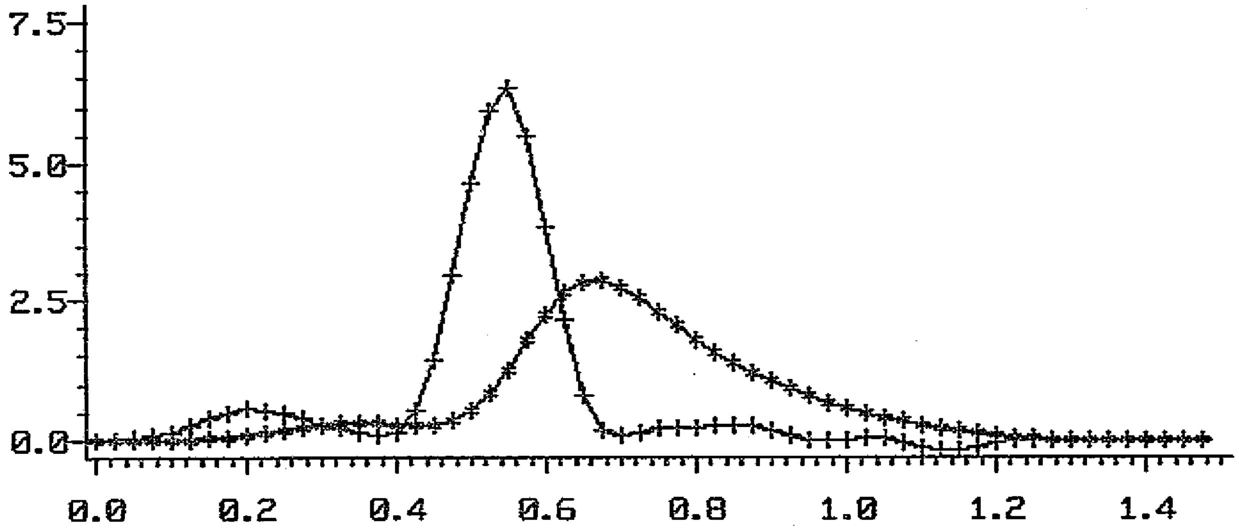


FIGURE 9

MIXTURE OF GAMMAS AND RESULTANT PRIOR FUNCTION

MIXTURE=* PRIOR=+

$\xi_1 = .20$ $\xi_2 = .55$

$\lambda_1 = .50$ $\lambda_2 = .50$

NUMBER OF POINTS PER DISTRIBUTION = 100

$\gamma = 0.1$

$\Delta = 0.1$

POSTERIOR OPERATOR APPROACH

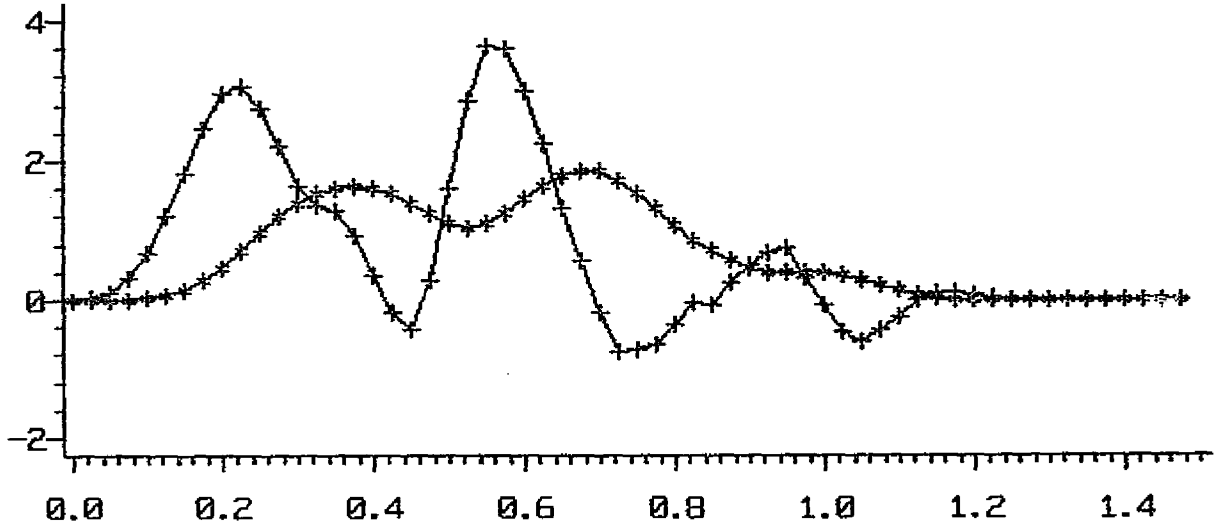


FIGURE 10

27 D2

NONPARAMETRIC ANALYSIS OF MINNESOTA SPRUCE AND ASPEN TREE DATA AND LANDSAT DATA

David W. Scott
and
Rodney Jee

Rice University

ABSTRACT

In this paper we describe our work in bringing nonparametric methods to bear on data-intensive problems faced by NASA. The theoretical development of efficient multivariate density estimators and the novel use of color graphics workstations are reviewed. The use of nonparametric density estimates for data representation and for Bayesian classification are described and illustrated. Our progress in building a data analysis system in a workstation environment is reviewed and preliminary runs presented.

1. Introduction

Our research program has focused on developing strategies based on nonparametric density estimation to aid in the varied large data analysis tasks faced by NASA. Our efforts have been twofold: first, to build these statistical tools into a highly interactive color computer graphics environment to experiment with multivariate data analysis; second, to vigorously pursue theoretical research in multivariate density estimation that will directly aid the practical application of our statistical tools.

In the first area, we have developed software based on new density estimation algorithms particularly well-suited for interactive computing. This software has been tested on simulated and real data sets with two, three, and four variables. The psychological impact of data analysis performed in this manner has been favorable and we continue our efforts to greatly facilitate use of our tools for new data sets. In the second area, we are examining those theoretical issues involved in our application of multivariate density estimation. We have developed a new density estimator called the averaged shifted histogram that is many times faster than the well-known kernel estimator. The statistical properties of this estimator are contained in Scott [7]. The calibration of density estimators is addressed in a joint paper with Terrell [11]. The result of this work should be an exportable product that can be used by researchers with varying levels of expertise in statistics.

In this paper we illustrate several applications of our methods and the workstation environment. First, our multivariate density function graphical representation is a powerful aid to constructing and understanding Gaussian mixture models for data in three and four dimensions. Second, the power of the average shifted histogram as the basis of a Bayesian classifier is illustrated on bootstrapped trivariate data. The performance is compared to that of a maximum likelihood Gaussian classifier. Third, we examine the practical significance of relatively modest departures of data from Gaussian assumptions. Our results indicate that some useful gains may be realized for such data. Fourth, we demonstrate a prototype of a data analysis system based on the Silicon Graphics Iris workstation. This system allows rapid comparison of data in up to four dimensions. We believe that significant gains will be realized as we break the two-dimensional barrier and begin to work directly with three and four dimensional data. We have attempted to create algorithms that make it possible to analyze such data with modest computational requirements, attempting to realize real-time representation and analysis.

2. Review

2.1 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]. 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.

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.

2.2 Computational and Representational Problems

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 methods; see Tapia and Thompson [10]. Thus we believe 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 over 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 which is formed by connecting with straight lines the mid-bin values of a histogram. This estimator has the same order of statistical efficiency as the kernel estimator and also the computational efficiency of the histogram. The frequency polygon of the ordinary histogram works well for two dimensional data; 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 a generalized histogram estimator called the averaged shifted histogram (ASH) [7]. The ASH is simply the pointwise average of m histograms with common equally spaced bins of width h but different bin origins $t_0 + \frac{i}{m}h$, $i = 0,\ldots,m-1$. Thus the ASH looks like a histogram with bin width h/m. As $m \to \infty$ the ASH is identical to the triangle 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 generalized contour plots. For trivariate data, a particular 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.

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 display would show several concentric spheres with the mode located at the center. If the variables 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.

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.

2.2 Graphical and Model-Based Discrimination and Classification

We shall assume that our data samples are labeled 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 (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. Examples are given next.

3. Examples

3.1 Three-Dimensional Analysis Using Badhwar Data

We shall consider the scatter diagram approach 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. Five acquisitions of 4-channel remote sensing reflectance 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 resembles 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.

The first group of examples and algorithms will be illustrated using 1977 data from segment 1663, which is located in North Dakota. The segment is primarily in agricultural use and contains large fields of sunflower, soybeans, sugar beets, spring wheat, barley, flax and oats. We wish to observe that our style of analysis is possible in a wide range of remote sensing and multivariate applications. These data have been furnished and are not completely unfamiliar to researchers.

To simplify the presentation, we chose to analyze pure pixels from three crops: sunflower, spring wheat, and barley. Sunflower is fairly easy to distinguish from other crops, but spring wheat and barley are less easy to distinguish. We have also chosen to analyze the three estimated Badhwar parameters for these pixels. This segment contained 3694, 3811, and 892 pure pixels, respectively. In Figures 1a, 2a, and 3a, scatter diagrams of the pure pixels are shown for each crop. The digital nature of the data is apparent from the display. Unfortunately in this paper we cannot use the color cue to give the value of the third orthogonal dimension. The axes are oriented as though we were looking at the data from infinity along the vector (1,1,1). The Badhwar estimation procedure produces clearly poor values for a small fraction of the pixels. Pixels with Badhwar parameters falling outside certain ranges for each crop were deleted from the analysis. The final numbers of pixels considered were 3505, 3782, and 873, respectively. The mean vectors and covariance matrices for these data were obtained using the usual maximum likelihood estimates. The contours of the fitted trivariate Gaussian density are shown in Figures 1b, 2b, and 3b, respectively. The contours are nested ellipsoids and are drawn at the levels 10%, 35%, and 70% of the respective modal value. The graphing area is the same for all views and crops. The same level contours of the estimated average shifted histograms (3 shifts in each dimension) are shown in Figures 1c, 2c, and 3c, respectively. Remembering that the average shifted histogram faithfully reproduces features in the data, we see some interesting but not dramatically non-Gaussian features for the spring wheat and barley data. What is not as clear is that the covariance matrix for the sunflower data has been affected by outliers (in spite of the deletions above) and the Gaussian mode is 0.00014 while the ASH mode is 0.00061. This illustrates the robust nature of the nonparametric density estimator. For the other two crops, the respective modes differed by less than 25%.

If we had more sophisticated displays, we could more clearly demonstrate the separation of the sunflower density from the others. In fact the separation is so large that even the overinflated Gaussian covariance estimate does not result in overlap with the other densities. In general, we should not expect to be so fortunate. There is significant overlap of the spring wheat and barley contours, the barley having a somewhat larger z-mean (peak greenness) than spring wheat. By the way, the sunflower density is forward-left in this picture, having a larger x-mean (later time of peak greenness). These features are more easily seen in the workstation environment described below.

We may ask what a mixture density would look like in this representation for a field with roughly equal numbers of pure pixels of the three crops. Using the estimated densities shown above, we formed the mixture density by adding the three densities and dividing by three. The contours (levels 1% and 10%) of the mixture density are shown in Figure 4. The sunflower contribution appears only in the 1% contour and not the 10% contour because of the inflated covariance matrix estimate. The usefulness of graphics such as these in programs that fit even more component densities seems clear and should be of great help in understanding these procedures.

The next step in our data analysis is to use the estimated densities to perform a pixel-by-pixel classification. The actual ground truth layout makes it difficult to present or understand our results, so we decided to use some modern statistical techniques to create a realistic field of size 60 pixels by 60 pixels using the bootstrap technique. The ground truth was chosen as shown in Figure 5 (unfortunately, the colors are not distinguishable with B&W film, but the error pictures indicate the pattern selected). The actual Badhwar parameters for each pixel were selected from the real database by selecting a pure pixel at random and assigning it to the "new" field. This bootstrapped field has roughly equal numbers of pixels for the three crops, about 1200 pixels each. The estimated densities were the same as in the previous discussion, obtained from the full outlier-deleted data sets.

The results of the classification are shown in Figures 5 and 6. In Figure 6, the left column indicates how pixels were classified (ASH on top; Gaussian below). To the right of these classification maps are the error maps; black pixels indicate correctly classified pixels while colored pixels indicate the crop misclassified. The misclassification table is shown in Figure 5. Both density estimators classified nearly 100% of the sunflower pixels correctly, as expected. However with the small grain pixels, the averaged shifted histogram performed much better than the Gaussian estimates, in spite of the rather modest non-Gaussian features. For spring wheat, the ASH correctly classified 76.5% while the Gaussian only 65.8%. For barley, the results were 73.8% and 76.7%, respectively, the Gaussian density performing slightly better. In an error reconciliation map, it is clear that both procedures misclassified many of the same pixels, but the nonlinear boundary of the ASH procedure resulted in superior performance.

The good performance of the ASH carries over to subsequent classification smoothing algorithms. For example, if we use a simple majority filter to smooth the previous pixel-by-pixel classification, we find the results shown in Figures 7 and 8. With this smoothing, 100% of sunflower pixels were correctly classified. For spring wheat, the results were 93.7% and 80.3%, respectively. For barley, the results were 89.9% and 93.7%, respectively. Clearly the Gaussian classifier has improved its score on barley pixels at the cost of many misclassified spring wheat pixels. This is the result of the non-Gaussian features in the data. We do not expect a substantial amount of bias in these estimates, but we have not pursued this point.

3.2 Workstations and Four-Dimensional Tree Example

The development of a workstation for the analysis of multivariate data is a project now receiving special attention due to the recent acquisition of a sophisticated computer graphics system. The Integrated Raster Imaging System (IRIS) by Silicon Graphics, Inc. provides our research at Rice with real time graphics capabilities. With the IRIS, instantaneous scalings, rotations, and translations of colored graphical images can be performed. The addition of real time graphics capability, which was lacking with our previous graphics hardware, not only enhances our analysis of 3 dimensional data but also provides for the effective representation of 4 dimensional data.

Our representation of 4 dimensional data is accomplished by using the fourth component as a "time" component, as described in section 2.2. For each interval in "time" we have a 3 dimensional data set (comprised of the first three components) which can be separately viewed as in the previous example. By displaying the smoothed sets of 3 dimensional data in sequence we effectively represent a data set of dimension 4.

Although the workstation for multivariate data analysis is currently in a prototypical stage of development, we present here its application on a data set provided by NASA. The raw data are 7 channel spectral measurements of aspen and spruce forests. This 7 dimensional data set is reduced in dimensionality from 7 to 4 by selecting the major 4 principal components axes as subspace projection axes. After the reduction in dimension from 7 to 4 and then a rescaling of data for purposes of graphical display, we calculate a sequence of contour shells for visual inspection.

In Figures 9-13 we illustrate how the workstation may be used in the comparison of data sets. On the left hand side of the photographs are the contour shells representation of the data from the aspen forests only. Opposite on the right hand side are the contour shells for the spruce forests only. Most apparent from the pictures are the observations that (1) the locations of the modes of the two data sets are well separated and (2) the shapes of the contours are different -- meaning the covariance structures are different. A more subtle feature of the contour shells is that they are not precisely elliptical, but skewed, indicating that the data are not quite Gaussian.

4. Conclusions

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.

Workstations are an exciting development for statisticians and data analysts. Our figures and the particular data sets chosen for analysis give one just a glimmer of what type of analysis will be possible on the workstation. Further work along these lines will focus on the nontrivial problem of how to reduce high dimensional data sets to dimension no greater that 4 for analysis on the workstation. The optimal reduction of dimensionality of data is the object of those working in the area labeled projection pursuit. We plan to incorporate these techniques into our software.

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, Proc. 28th Conference on the Design of Experiments in Army Res Dev and Testing (1983), 387-397.

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

[7] Scott, D.W., Average Shifted Histograms: Effective Nonparametric Estimators in Several Dimensions, submitted (1984).

[8] Scott, D.W., and Thompson, J.R., Probability Density Estimation in Higher Dimensions, Proceedings of the 15th Symposium on the Interface of Computer Science and Statistics, J.E. Gentle, Ed., North-Holland, Amsterdam, (1983), 173-179.

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

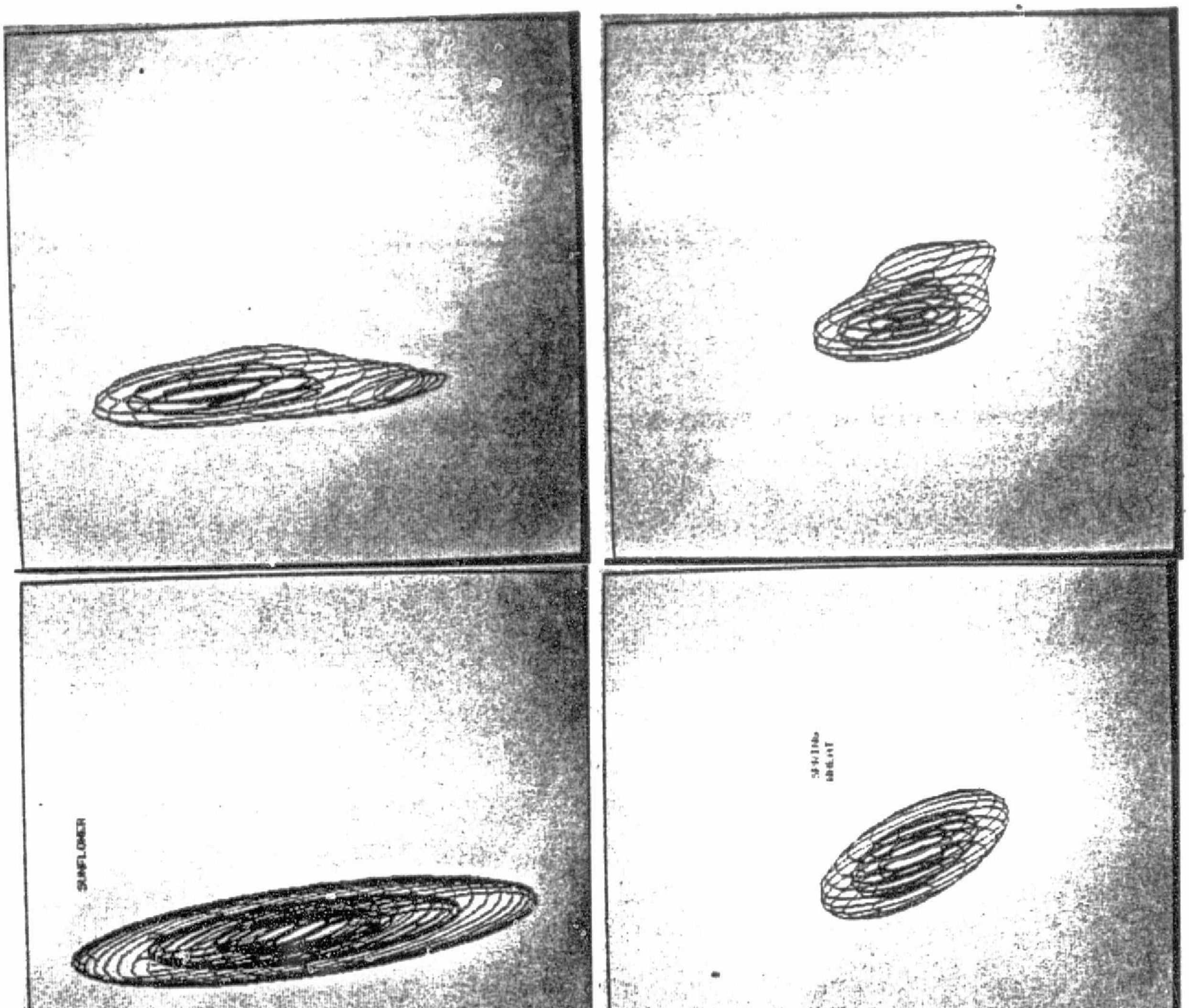
[10] Tapia, R.A., and Thompson, J.R., Nonparametric Probability Density Estimation, (Johns Hopkins Univ. Press, Baltimore, 1978).

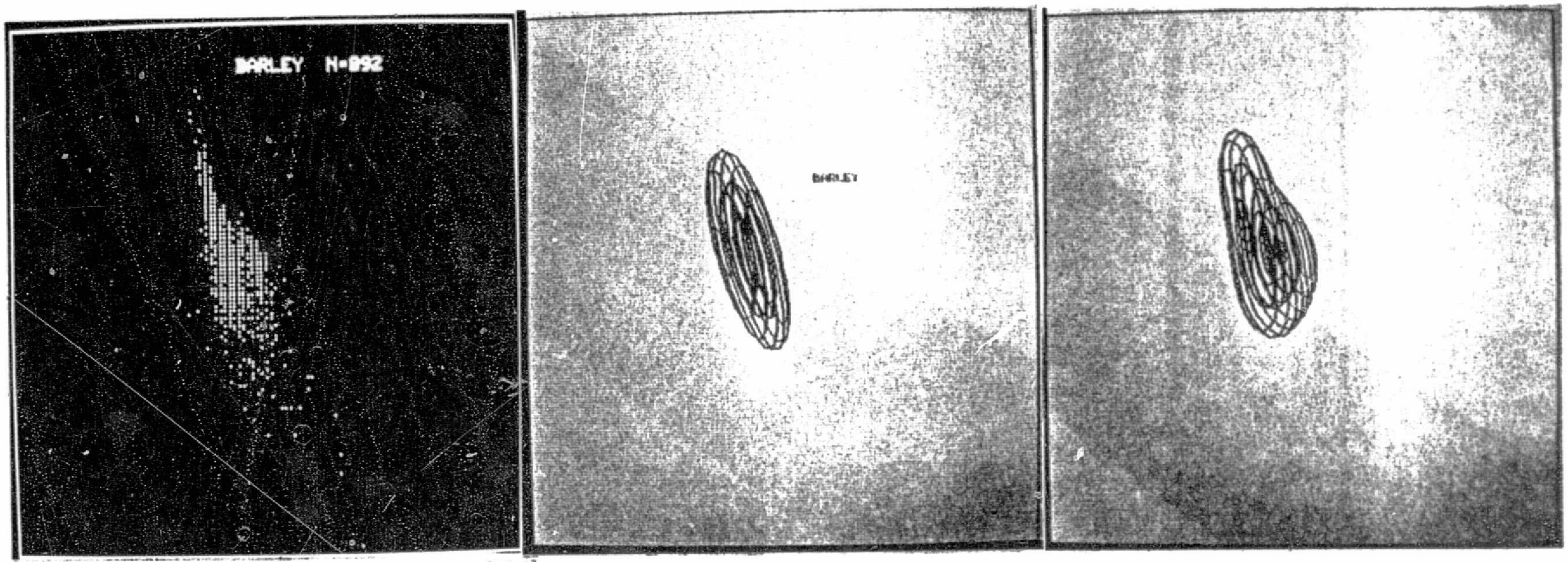
[11] Terrell, G.R., and Scott, D.W., Oversmoothed Nonparametric Density Estimates, submitted.

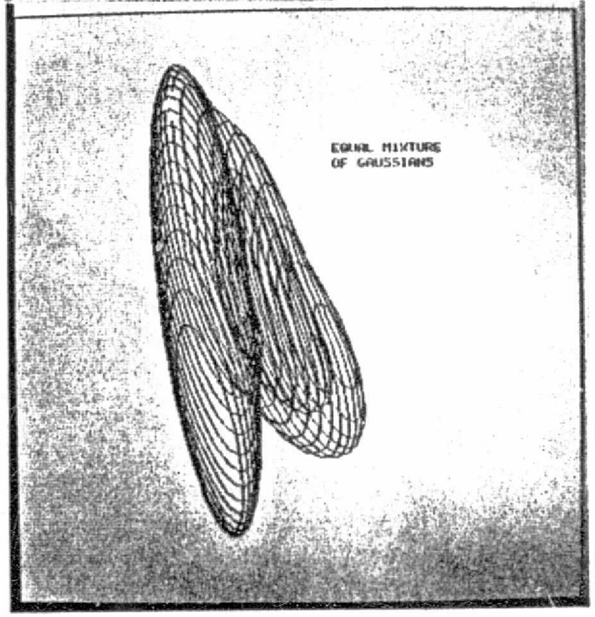
[12] Tufte, E.R., The Visual Display of Quantitative Information, (Graphics Press, Cheshire, CT, 1983).

[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).

ORIGINAL PAGE IS
OF POOR QUALITY







Previous Page:

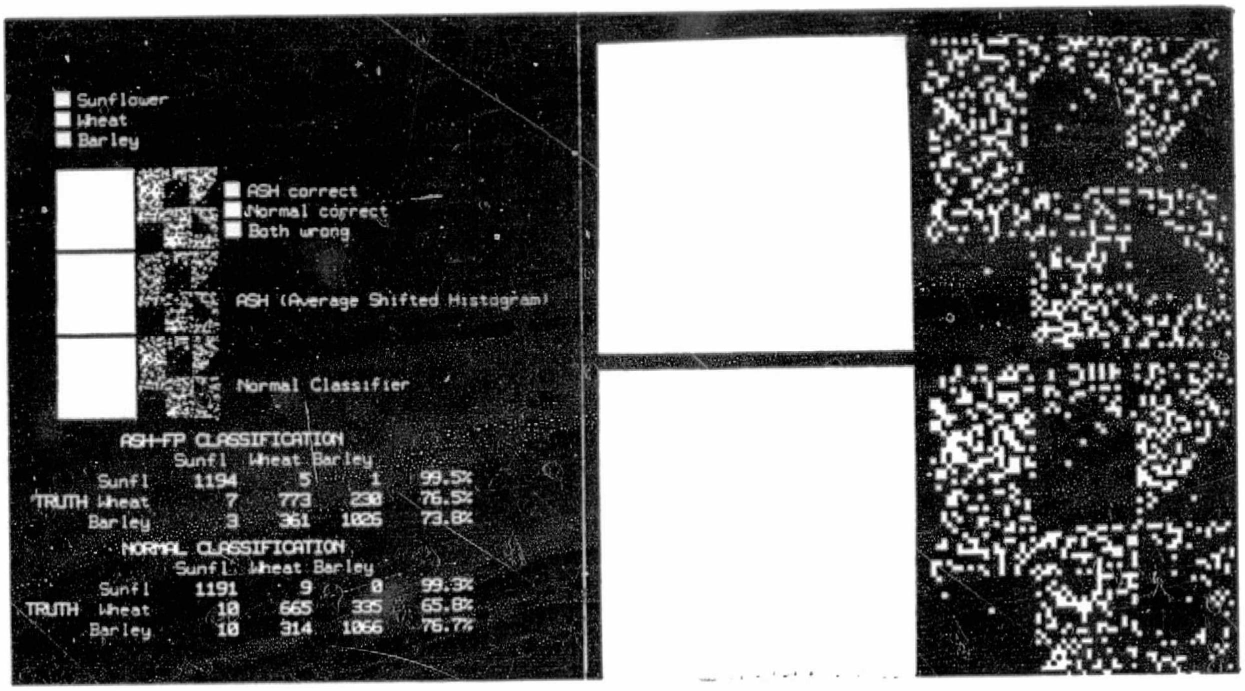
Top Row: Figures la,b,c: Sunflower pixels Scatter diagram, Normal and ASH density contours.

Second Row: Figures 2a,b,c: Same as Fig. 1 for spring wheat pixels.

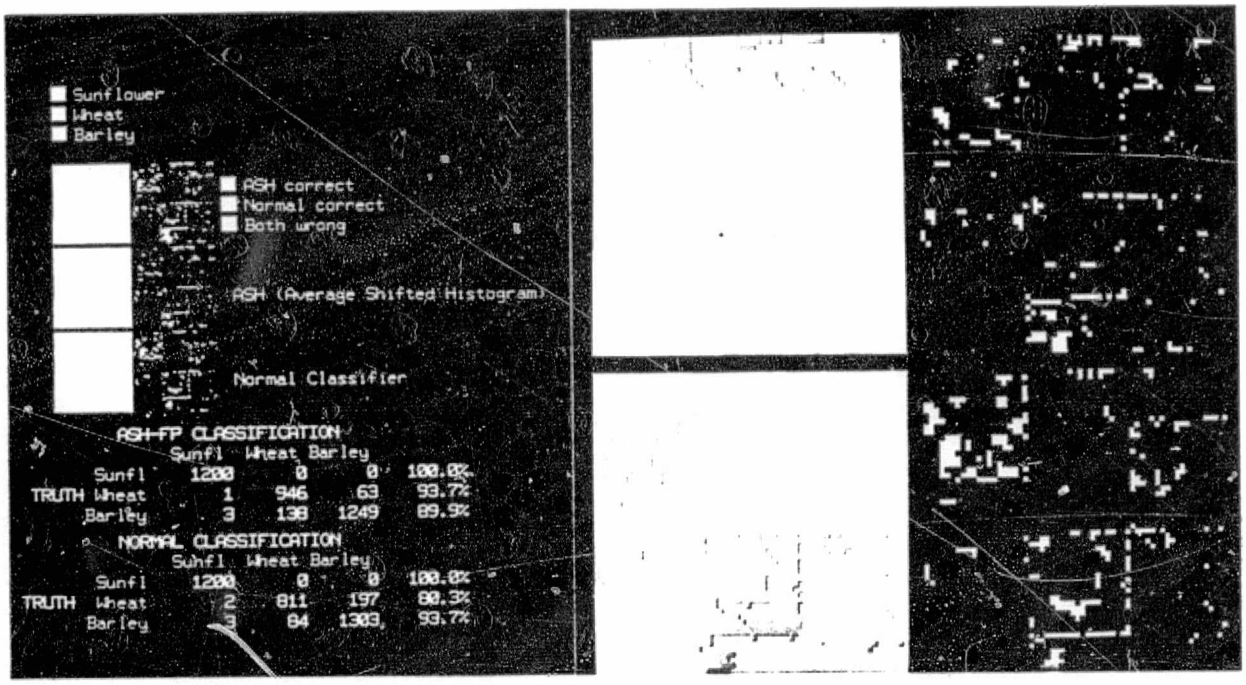
This Page:

Top Row: Figures 3a,b,c: Same for barley pixels.

Second Row: Figure 4: Contours of an equal mixing of the fitted Gaussian densities of the three crops.



```
        ASH-FP CLASSIFICATION
              Sunfl  Wheat Barley
       Sunfl   1194      5      1   99.5%
 TRUTH Wheat      7    773    230   76.5%
       Barley     3    361   1026   73.8%
        NORMAL CLASSIFICATION
              Sunfl  Wheat Barley
       Sunfl   1191      9      0   99.3%
TRUTH  Wheat     10    665    335   65.8%
       Barley    10    314   1066   76.7%
```

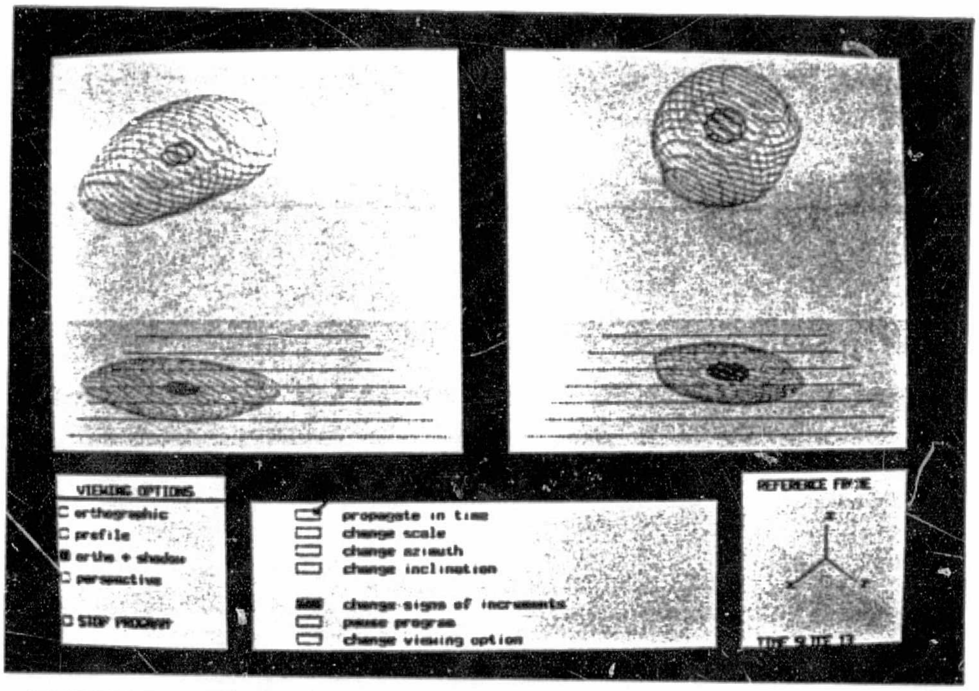


```
        ASH-FP CLASSIFICATION
              Sunfl  Wheat Barley
       Sunfl   1200      0      0  100.0%
 TRUTH Wheat      1    946     63   93.7%
       Barley     3    138   1249   89.9%
        NORMAL CLASSIFICATION
              Sunfl  Wheat Barley
       Sunfl   1200      0      0  100.0%
TRUTH  Wheat      2    811    197   80.3%
       Barley     3     84   1303   93.7%
```

Top Row: Figures 5 and 6: Pixel-by-pixel classification of bootstrapped scene. Figure 6 is a blowup of the second and third rows in Figure 5. See text.

Second Row: Figures 7 and 8: Result of majority smoothing filter applied to the previous scene. See text.

ORIGINAL PAGE IS
OF POOR QUALITY



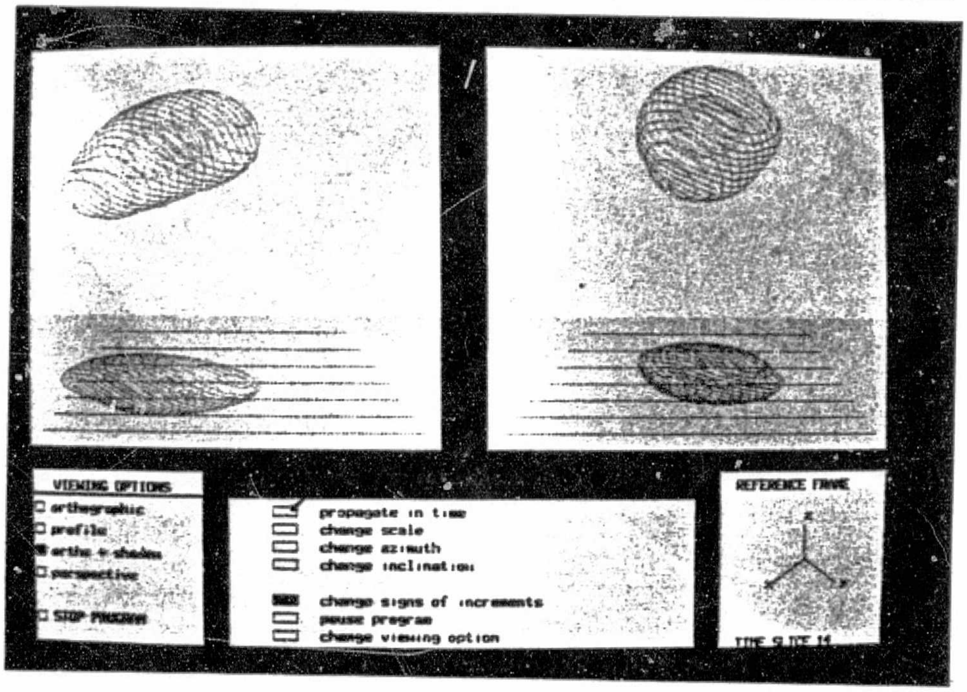
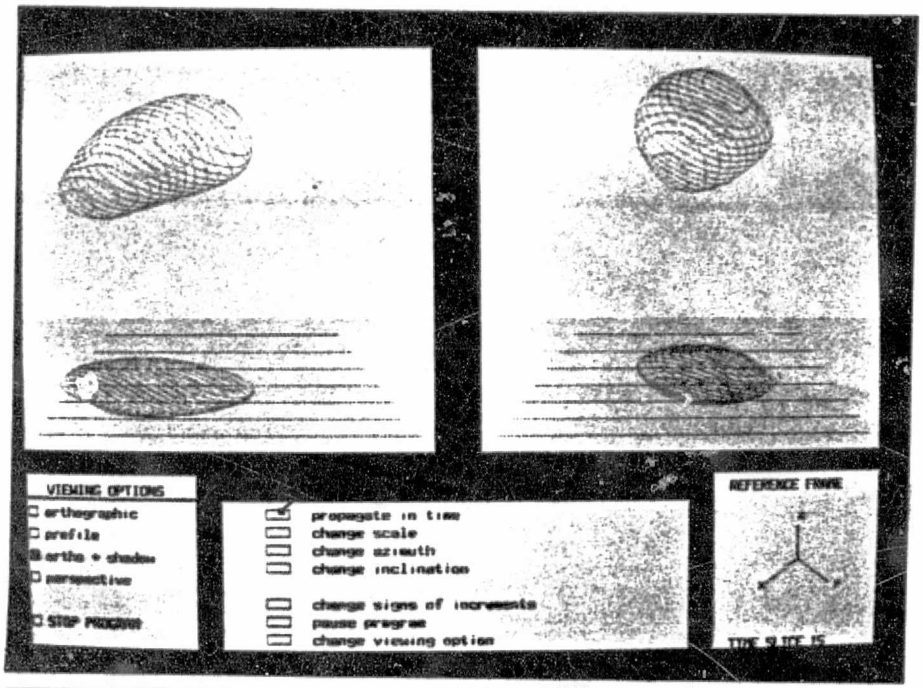


Figure 9 (top): 4-D Data represented by a sequence of contour shells. Aspen data (left) versus spruce data (right). Time slice 13 means x_4=-.2.

Figure 10 (bottom): Continuation of sequence from Figure 9 with x_4=-.1.



VIEWING OPTIONS

□ orthographic
□ profile
■ ortho + shadow
□ perspective

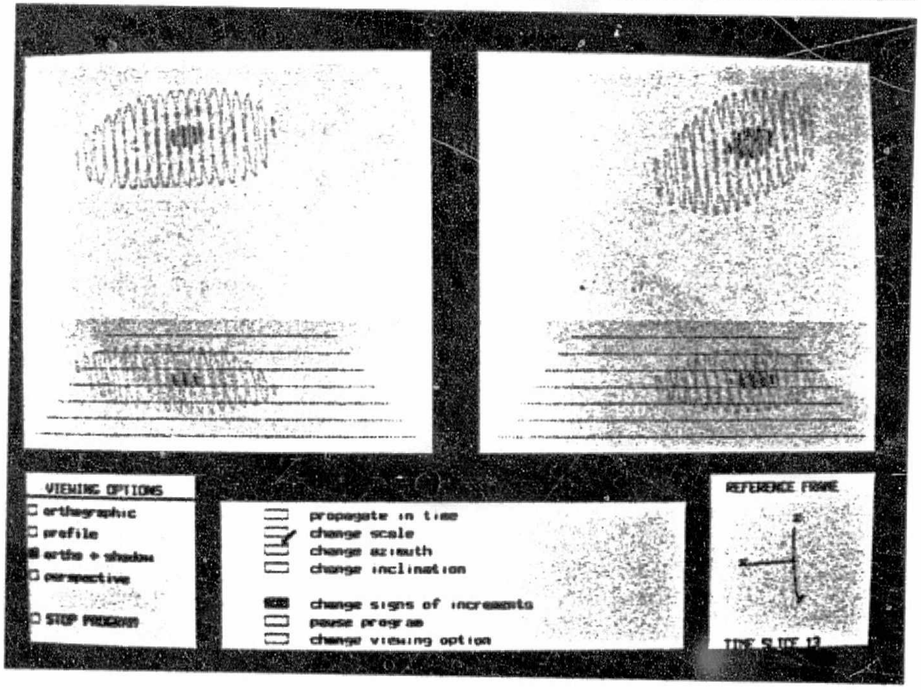
□ STOP PROGRAM

☑ propagate in time
□ change scale
□ change azimuth
□ change inclination

□ change signs of increments
□ pause program
□ change viewing option

REFERENCE FRAME

TIME SLICE 15



VIEWING OPTIONS

□ orthographic
□ profile
■ ortho + shadow
□ perspective

□ STOP PROGRAM

□ propagate in time
☑ change scale
□ change azimuth
□ change inclination

■ change signs of increments
□ pause program
□ change viewing option

REFERENCE FRAME

TIME SLICE 12

Figure 11 (top): Continuation of sequence from Figure 10 with $x_4=0$.

Figure 12 (bottom): Same contour shells as in Figure 9 with different rotation.

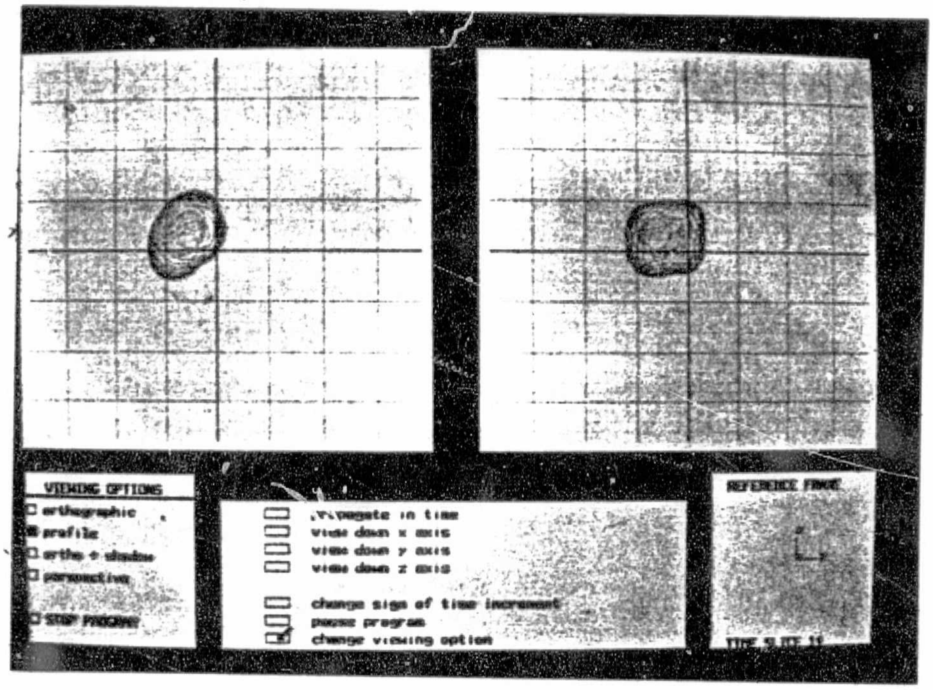


Figure 13: Same contours shells as in Figure 10 viewed down the x_1 axis with grid overlaid.

D3

TEXTURE CLASSIFICATION USING AUTOREGRESSIVE FILTERING

Wayne M. Lawton
and
Meemong Lee

Jet Propulsion Laboratory

TEXTURE CLASSIFICATION USING AUTOREGRESSIVE FILTERING

Wayne M. Lawton
and
Meemong Lee

Jet Propulsion Laboratory

ABSTRACT

A general theory of image texture models is proposed and its applicability to the problem of scene segmentation using texture classification is discussed. A new algorithm, based on half-plane autoregressive filtering, which optimally utilizes second order statistics to discriminate between texture classes represented by arbitrary wide sense stationary random fields is described. Empirical results of applying this algorithm to natural and synthesized scenes are presented and future research is outlined.

INTRODUCTION

The purpose of this paper is to present preliminary theoretical and experimental results of our investigation into approaches to automatic scene segmentation based on texture analysis. It is imperative to examine fundamental methodology.

As perceived through our senses of touch and vision, texture is a property of surfaces and images easy to assimilate yet difficult to articulate. Consistent with this view are the following remarks: "Despite its importance and ubiquity in image data, a formal approach or precise definition of texture does not exist." - Robert Haralick, ref. [10].
"Texture is an elusive notion which mathematicians and scientists tend to avoid because they cannot grasp it. Engineers and artists cannot avoid it, but mostly fail to handle it to their satisfaction." - Benoit Mandelbrct, p. 310, ref. [18].

In consideration of this view, it may be argued that, for the purpose of developing computer algorithms to perform scene segmentation based on texture analysis, empirical descriptions of texture are more appropriate than formal mathematical models. While acknowledging the utility of specific ad hoc techniques developed using tnis approach, we question the ability of this approach to provide a framework for developing techniques comparable in performance and flexibility with the human visual system. Discussing the limitations of the empirical approach of the modern behavioral school of psychology, in ref. [5] Noam Chomsky remarks:

"To go further, I believe that the inability of modern psychology to come to grips with the problems of human intelligence is in part, at least, a result of its unwillingness to undertake the study of abstract structures and mechanisms of mind....." "Had the physical sciences limited themselves by similar methodological strictures, we would still be in the era of Babylonian astronomy."

It is our opinion that formal mathematical models of texture are necessary fr · developing algorithms and for understanding their performance.

Our methodology employs three processes. First, we assert a theory for a class of image textures. This theory is a metaphor, based on our experience with the physical phenomena of image texture, which draws a comparison with a mathematical model as described by a set of axioms. This experience includes an inspection of myriad images, rich in textural detail, arising in histology, ref [16], the classic photography of Phil Brodatz, ref. [2], [3], [4], Landsat Imagery, ref. [8], and many optical, infrared and microwave images at JPL's Image Processing Laboratory. The mathematical model describes images which appear to decompose into distinct connected regions such that each region is perceived as homogeneous and distinct from the regions contiguous to it. The quality of homogeneity occurs when a region within an image has a constant brightness, a repetitive pattern or a random pattern which appears the same throughout the region. The model consists of associating with each region a stationary, ergodic random field that accounts for the texture class of the region. Contiguous regions are required to 'belong to' different texture classes though more than one region may belong to the same texture class. The image brightness values over each connected region correspond to a realization of the random field (a choice of sample values for each of the random variables corresponding to the lattice points within the region) over the region. Stationarity accounts for the homogeneous property of the image over each region and ergodicity is equivalent to the assumption that the parameters that describe the random fields (the joint probability density functions) can be estimated from the image brightness levels. Textbook accounts of random fields are found in ref. [1], [24], and [25]. The concept of stationarity we utilize is rather general. For the purpose of developing the scene segmentation algorithm discussed in this paper, which utilizes only second order statistics, we assume stationarity in the wide sense (that the means and covariances exist and are translation invariant). We have also developed a nonlinear extension of the algorithm that assume stationarity in the strict sense (that the joint probability distributions are translation invariant). Furthermore, we have extended the concept of stationarity so our model subsumes both the fractal model discussed in ref. [17], [18], and [20] and the structural model discussed in [21].

The second process employed by our methodology is a mathematical elaboration of the model. This consists of an exploration of logical consequences (theorems) of the axioms describing the model that have a significant predictive value. These predictions allow formulation of experiments to test the theory (determine whether the models proposed by the theory are consistent with observations) and they provide design criteria and design architecture for algorithms to solve the problem of scene segmentation based on texture classification.

Because the subject of random fields is in its infancy, it was necessary to discover new mathematical results as well as examine known results related to our model. The concept of representing image textures by wide sense stationary random fields and the subsequent application of autoregressive techniques to image processing problems is not new, see ref. [15, p. 238-243], [22], [9], [19], and [23]. However, assumptions are usually made restricting the class of random fields to which the techniques are applicable. These include the following assumptions:

Assumption 1. The random field has a joint Gaussian distribution.

Asseumption 2. The random field is described by a quarter plane (causal) finite autoregressive model.

Our mathematical investigations resulted in a precise characterization of the restrictions imposed by assumptions 1 and 2 above and a method for extending autoregressive techniques to arbitrary wide sense stationary random fields. Also, we generalized the autoregressive technique to 1) an autoregressive technique applicable to wide sense stationary random fields with values in a vector space (to include multispectral imagery), 2) a non-linear filtering technique, based on conditional probabilities, applicable to arbitrary strict sense stationary random fields with values in an arbitrary set (the set might consist of 'local primitives' as defined by the structural approach, see ref. [10], [21]). Each of the techniques above can be proven to be optimal in the sense of Bayes. A complete mathematical treatment of these results is beyond the scope of this paper and will be presented in a final research report.

The third process consists of designing experiments and making empirical observations. These include the computer implementation of an algorithm for scene segmentation based on texture classification, utilizing autoregressive filtering, which is described in Chapter 2, and the application of this algorithm to natural and synthetic images, which is described in Chapter 3. Chapter 4 discusses future experimental research.

2. SCENE SEGMENTATION AlGORITHM

The purpose of this chapter is to describe a new algorithm, based on an autoregressive filtering technique, for scene segmentation using texture classification. The scene model assumptions, which are quite general, are the following.

Assumption 1. The scene or image data consists of a real valued function X defined on a finite subset D of a two dimensional lattice L. The function X represented the image brightness level.

Assumption 2. The set D decomposes into a finite union of disjoint connected subsets, called regions. To each region corresponds a wide sense stationary real valued random field on L, whose second order statistical parameters represent the texture class for that region, such that the restriction of X to each region arises as a specific realization of the random field corresponding to that region over that region.

Assumption 3. The second order statistics (consisting of the mean value and the autocovariance function) of any two random fields corresponding to contiguous regions are not identical.

It is important to observe that we do not impose any special properties for the second order statistics. Nor do we assume any specific form for the joint probability density functions of the random field - in particular we do not assume they are Gaussian, in any sense, or even stationary. Furthermore, it follows from our assumptions that the performance of any algorithm for scene segmentation based on these assumptions will be limited by its ability to discriminate between texture classes based only on estimates of second order statistical parameters. To support our contention that this does not impose a practical limitation, we make the following observations. First, there is substantial experimental evidence indicating that human visual texture discrimination utilized only second order statistics, see ref. [13], [14]. Second, Fourier methods of texture analysis, which utilize second order statistical information, have proved useful in number of applications, see ref. [6], [7]. These two observations suggest that second order statistics yield a reliable basis for discriminating between statistically distinct regions in natural imagery. Third, it follows from a theorem we derived that if an algorithm utilizing second order statistics yields a high confidence of correct classification using Bayes' formula with Gaussian parameters, then the level of confidence is also high if the actual distribution is non-Gaussian. Fourth, the restriction to second order statistics reduces computational complexity immensely. Fifth, usually the actual distribution of the random field is not usually known (an exception occurs for synthetic radar images of clutter backgrounds which have a ChiSquare distribution with the two degrees of freedom).

The algorithm consists of the following steps.

Step 1. Select a scene X, natural or synthesized. We have developed a computer routine to generate realizations of jointly Gaussian distributed random fields whose spectrums can be arbitrarily specified by a finite sum of functions, each constant over a rectangle. It uses the FFT algorithm to perform an approximation to Norbert Wiener's stochastic integral representation of Gaussian random processes.

Step 2. Select training areas and estimate the means and autocovariances for each of the N texture classes.

Step 3. Choose a finite subset S which is a subset of the half-plane $\{(m,n)$ such that $n>o\} \vee \{(m,o)$ such that $m \geq o\}$. For each k=1...,N, calculate a function $F_k:S \rightarrow R$, $F_k(o,o)=1$ and the variance of F_k*R_k is minimized where $R_k$denotes the random field for K-th class.

Step 4. For each K=1,...,N, convolutional filter the original scene to form $X_k=F_k*X$.

Step 5. For each K=1,...,N, calculate $Y_k=\log\sqrt{2\pi}V_k + (X_k-M_k)^2/2V_k$ where $V_k$and $M_k$is the variance and mean of $X_k$as calculated using parameters from steps 2 and 3.

Step 6. Choose a box size $B > o$ and apply an B x B box filter (average filter) to Y_k to obtain Z_k.

Step 7. Classify pixel $(a,b) \in L$ into the j-th class if $Z_j(a,b) \leq Z_k(a,b)$ for all $k \neq j$.

A flow chart depicting this algorithm is given in the attached algorithm flow chart.

Based on the theory of half-plane prediction developed in ref. [11], [12] this algorithm has the following properties:

Property 1. As the set S gets 'large' and the box size gets large, the performance approaches that for an optimal Bayesian classifier if each texture class is described by a jointly Gaussian field.

Property 2. If the performance of the algorithm is high, then the level of confidence is at least as high as predicted for a jointly Gaussian process.

Property 3. The algorithm has minimal computational complexity among the class of all algorithms satisfying either property above.

3. Algorithm Evaluation

The autoregressive filtering algorithm was evaluated using various synthesized images and some natural scenes. Two types of image synthesis methods, convolution method and direct spectrum synthesis method were implemented. The algorithm was applied to the synthesized images with fixed parameter set up of training area (64 pixel by 64 pixel), autoregression window (24 neighbor pixels) and smoothing window (5 pixel by 5 pixel) for evaluation.

The convolution method synthesized an image as a function of neighbor pixels as described in equation 1.

$$(1)\ B(i,j) = c1A(i+1,j+1) + c2A(i+1,j+2) + c3A(i+1,j+2) + c3A(i+2,j+2)$$

where A is the uniformly distributed random noise image array and c1,c2, c3,c4 are arbitrarily assigned constants.

The Figure 3.1 shows the segmentation result of a uniformly distributed random noise image and a synthesized image using the direct convolution method. The segmentation result was found to be 85% correct. Figures 3.2 and 3.3 illustrates histogram distribution characteristics of the two images.

More synthesized images using the convolution method were tested with various c1,c2,c3, and c4 values. The figures from 3.4 and 3.6 illustrate algorithm robustness against image similarities. When the convolution constants of two synthesized images became closer to each other (i.e., images more similar to each other), the segmentation results became slightly worse.

The direct spectrum synthesis method was developed based on the fact that if the spectrum of an image generated by convolving an arbitrary function with a Gaussian random noise is known, the image can be synthesized directly from the spectrum. The equation 2 shows the detail relation.

$$(2)\quad \begin{aligned} B &= F * G \\ &= \text{APP. [Fourier } (\text{SQRT}(S))\ G\ ] \end{aligned}$$

where S is the user defined spectrum
$\text{Spec}(B) = S$
$F = \text{Fourier SQRT}(S)$
G is the Gaussian random noise.

Therefore, various spectral characteristic images can be synthesized by applying different spectrums. For implementation convenience, the spectrum was generated as several rectangular shapes of spectrum coefficients. Figures 3.7 and 3.8 illustrate the synthesized images from the given spectrums.

Figure 3.9 is the segmentation result of the two synthesized images. The result shows perfect segmentation of two textures. The figure 3.10 shows the high frequency noise effect on texture classification. Figures from 3.11 to 3.14 show segmentation results applied to various types of spectrum images with smoothing window size of 5 by 5 and 10 by 10. Enlarged smoothing window showed a little improvement.

It was found that the segmentation performance gets poorer when the spectrum distribution became broader. This is due to the relation between the spectrum energy concentration characteristics and the autoregression window size. The detail relation is not yet defined and requires further study.

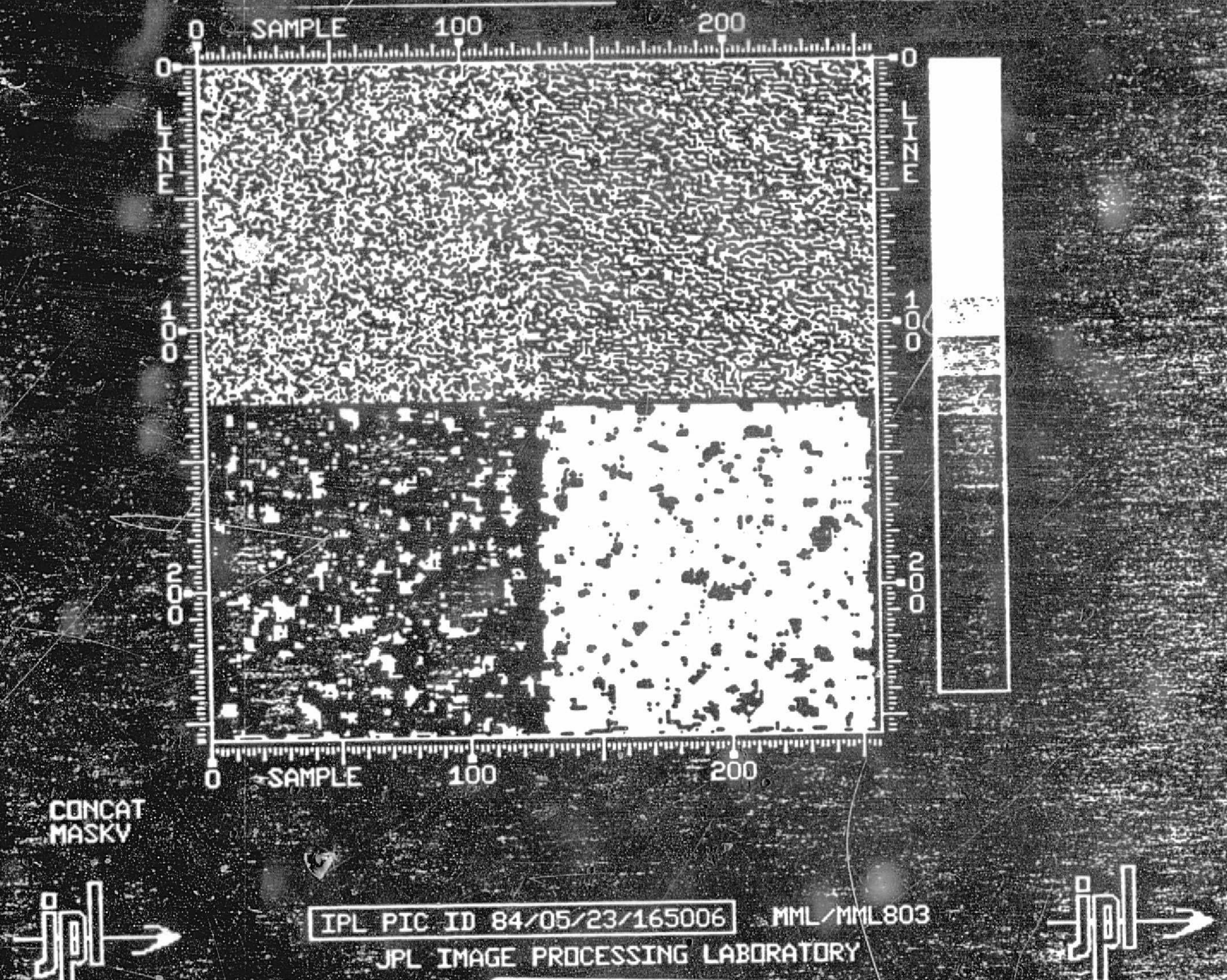

0
SAMPLE
100
200
0
LINE
100
200
0
LINE
100
200
0
SAMPLE
100
200
CONCAT
MASKV
IPL PIC ID 84/05/23/165006
MML/MML803
JPL IMAGE PROCESSING LABORATORY

Figure 3.1

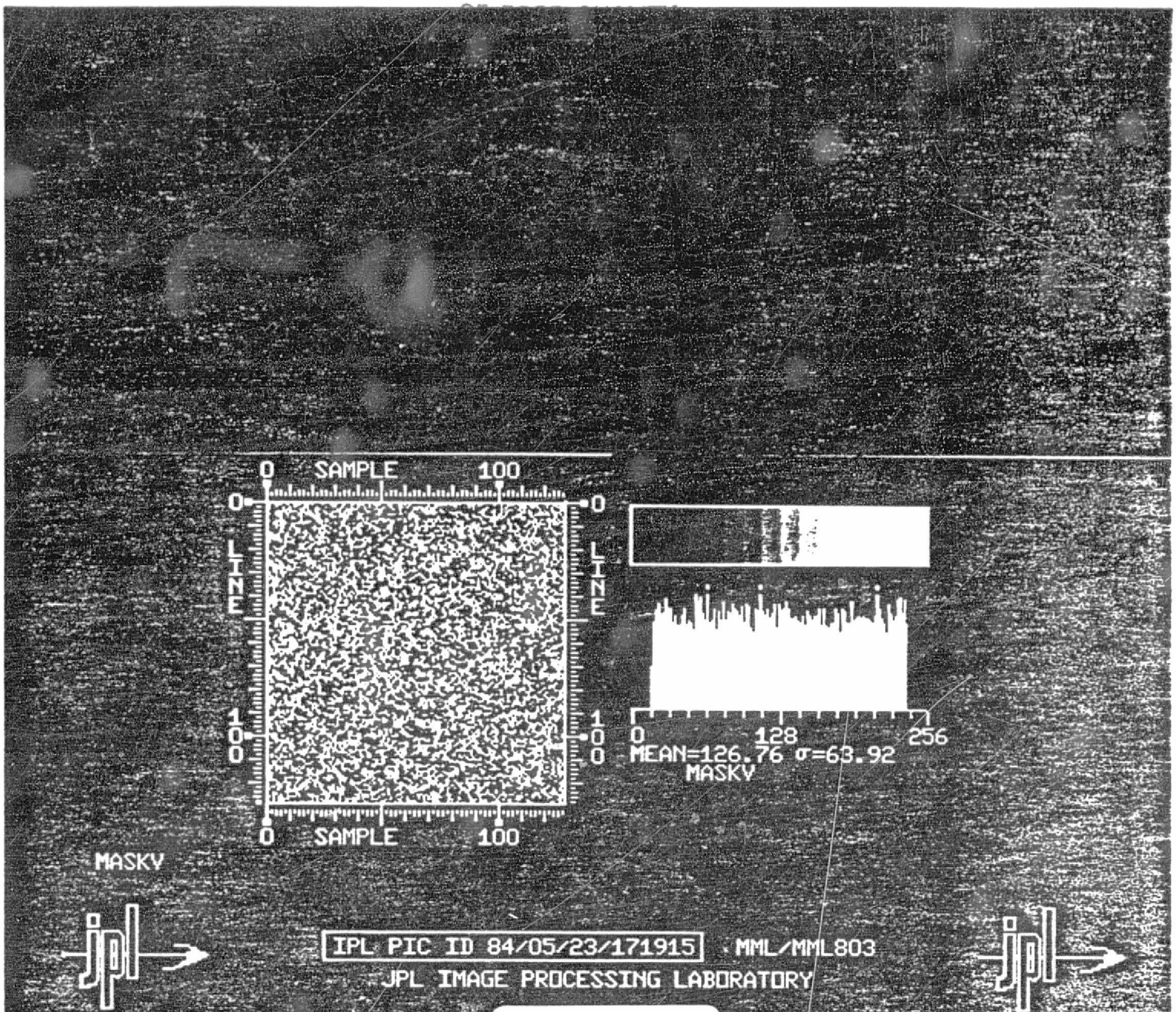

0
SAMPLE
100
0
LINE
100
0
LINE
100
0
128
256
MEAN=126.76 σ=63.92
MASKY
0
SAMPLE
100
MASKY
jpl
IPL PIC ID 84/05/23/171915
MML/MML803
JPL IMAGE PROCESSING LABORATORY
jpl

Figure 3.2

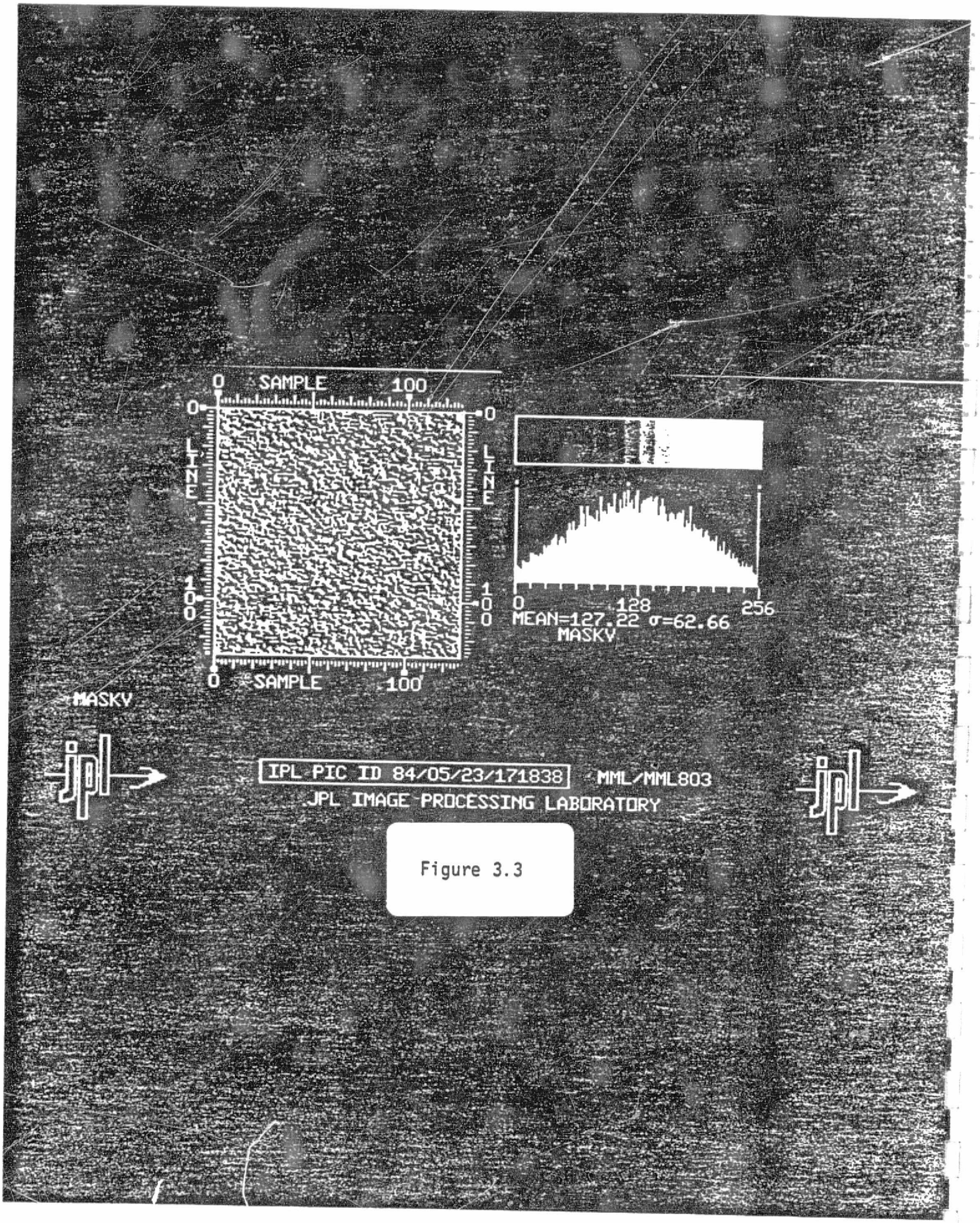

0
SAMPLE
100
0
LINE
LINE
0
100
100
0
SAMPLE
100
0
128
256
MEAN=127.22 σ=62.66
MASKY
MASKY
jpl
IPL PIC ID 84/05/23/171838
MML/MML803
JPL IMAGE PROCESSING LABORATORY
jpl

Figure 3.3

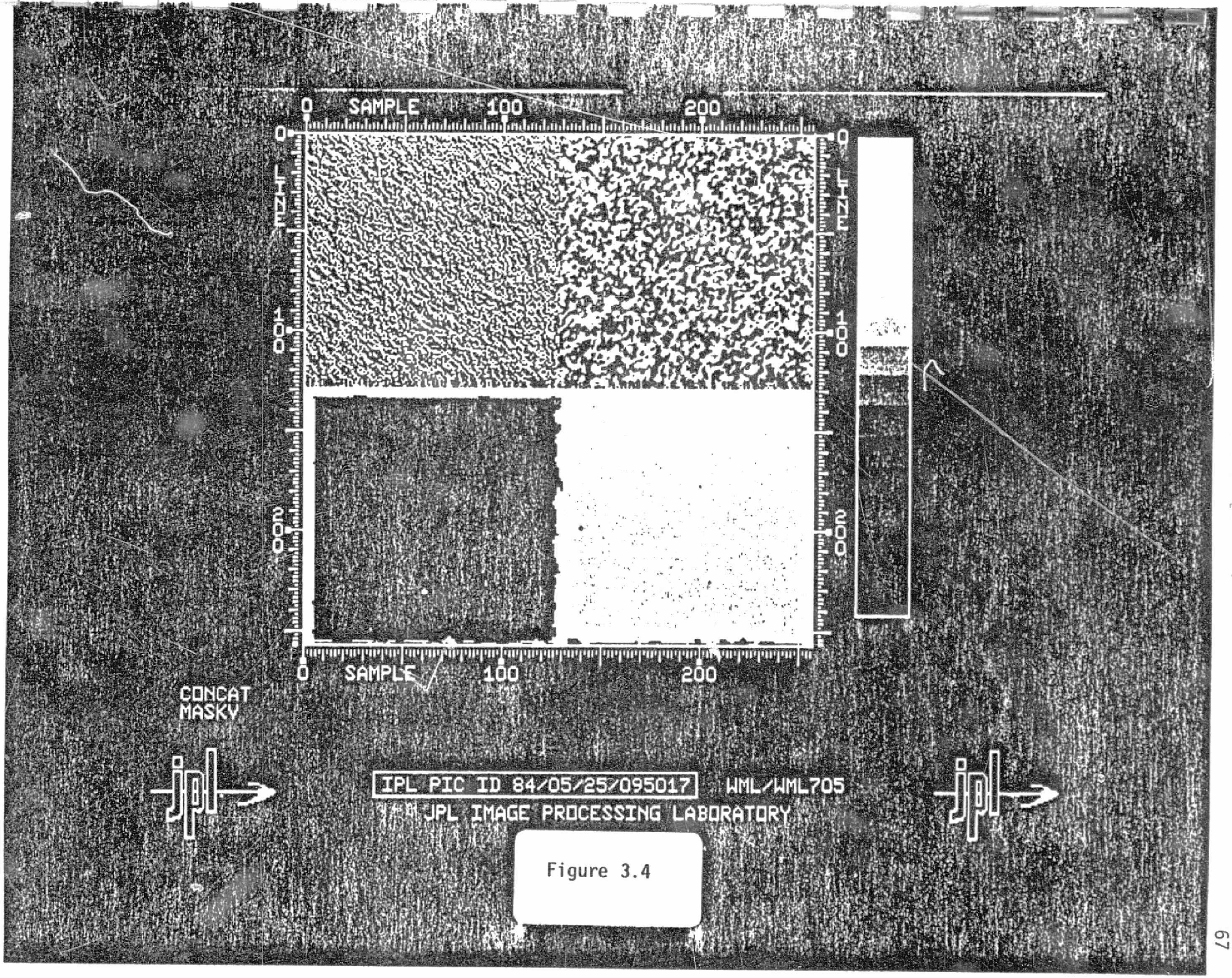

SAMPLE
0
100
200
LINE
0
100
200
SAMPLE
0
100
200
CONCAT
MASKY
IPL PIC ID 84/05/25/095017
WML/WML705
JPL IMAGE PROCESSING LABORATORY

Figure 3.4

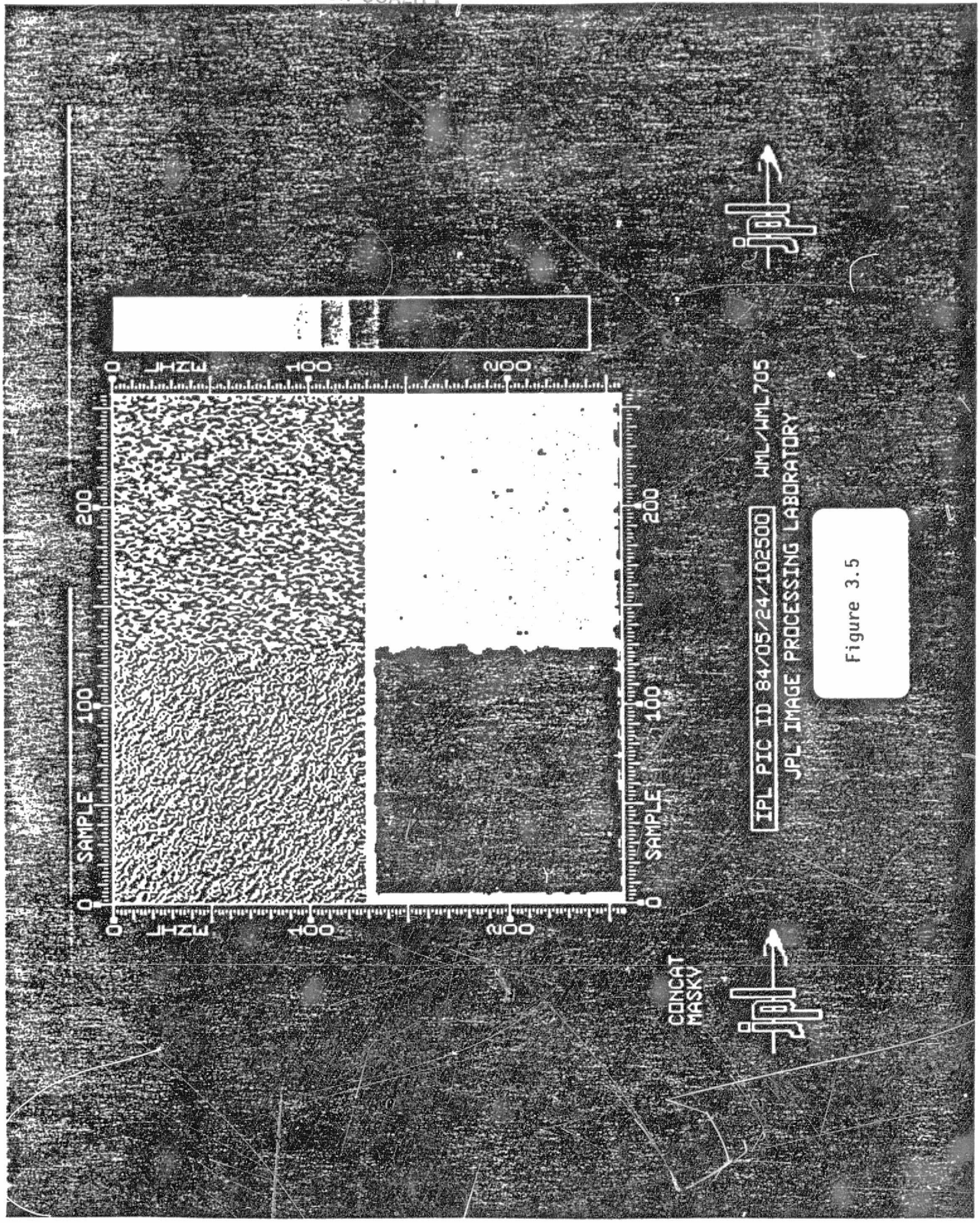

SAMPLE
0
100
200
LINE
0
100
200
CONCAT
MASKV
IPL PIC ID 84/05/24/102500 WML/WML705
JPL IMAGE PROCESSING LABORATORY
jpl

Figure 3.5

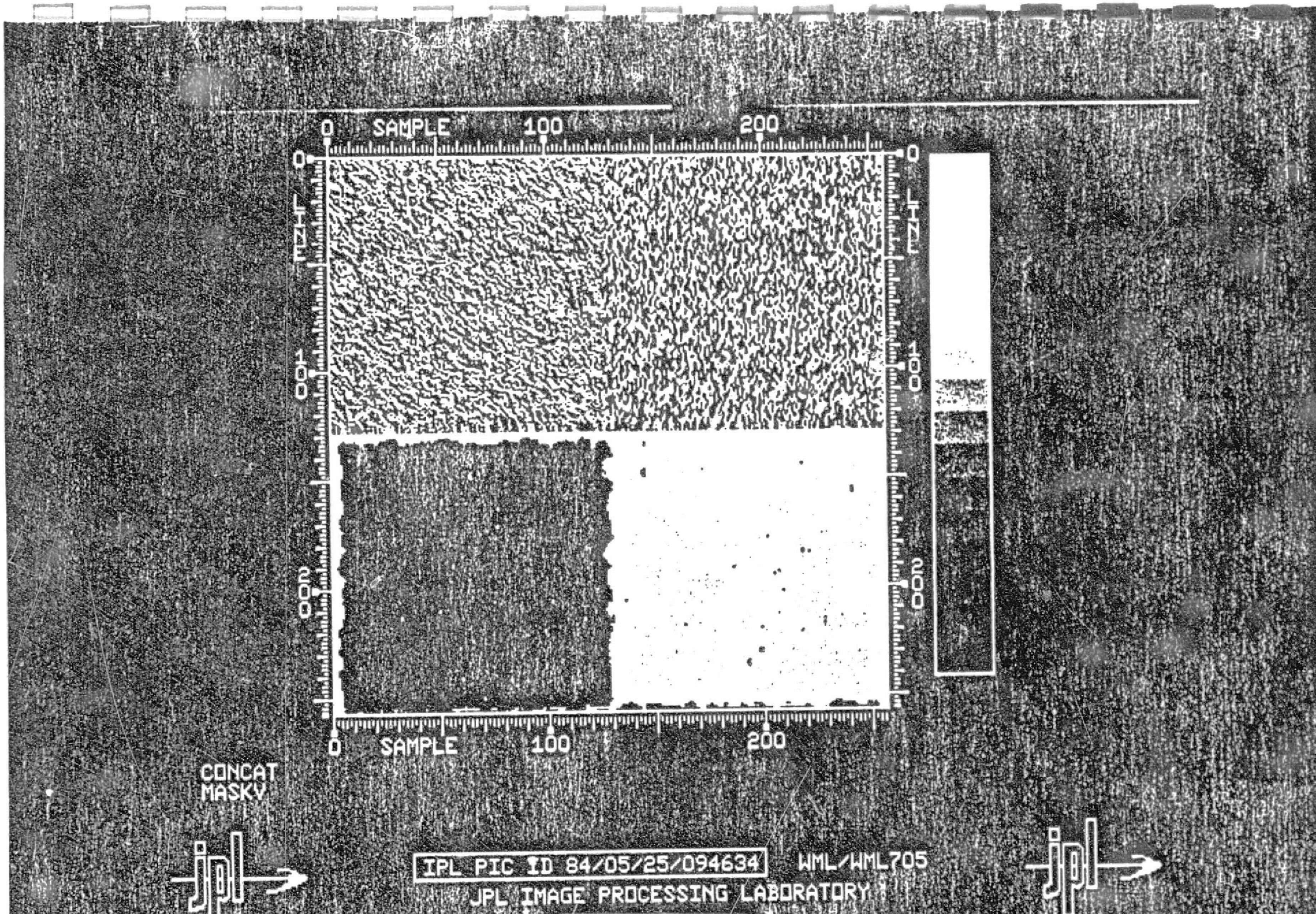

0
SAMPLE
100
200
LINE
100
200
0
SAMPLE
100
200
CONCAT
MASKV
IPL PIC ID 84/05/25/094634
WML/WML705
JPL IMAGE PROCESSING LABORATORY

Figure 3.6

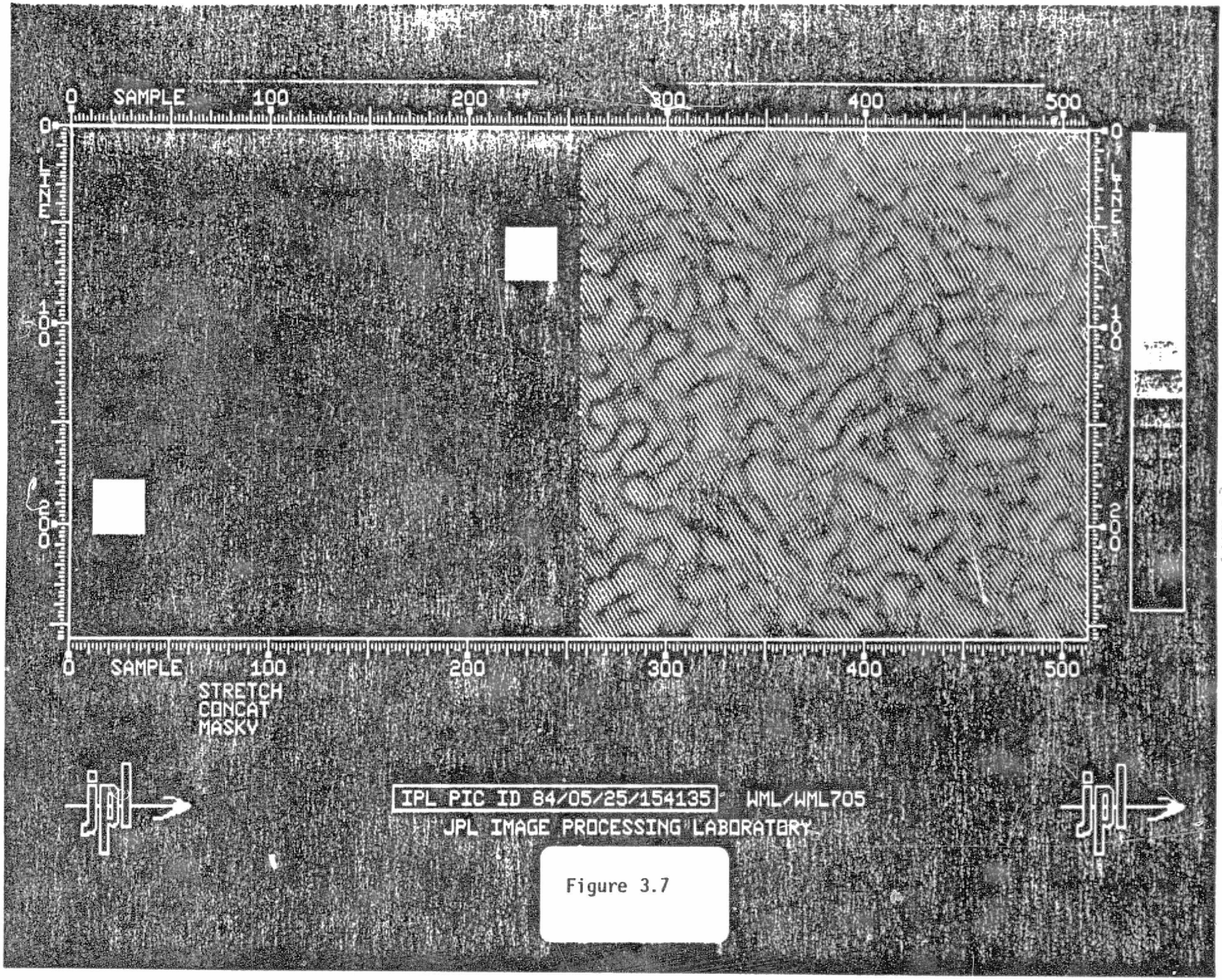

0
SAMPLE
100
200
300
400
500
0
LINE
100
200
STRETCH
CONCAT
MASKY
IPL PIC ID 84/05/25/154135
WML/WML705
JPL IMAGE PROCESSING LABORATORY
jpl

Figure 3.7

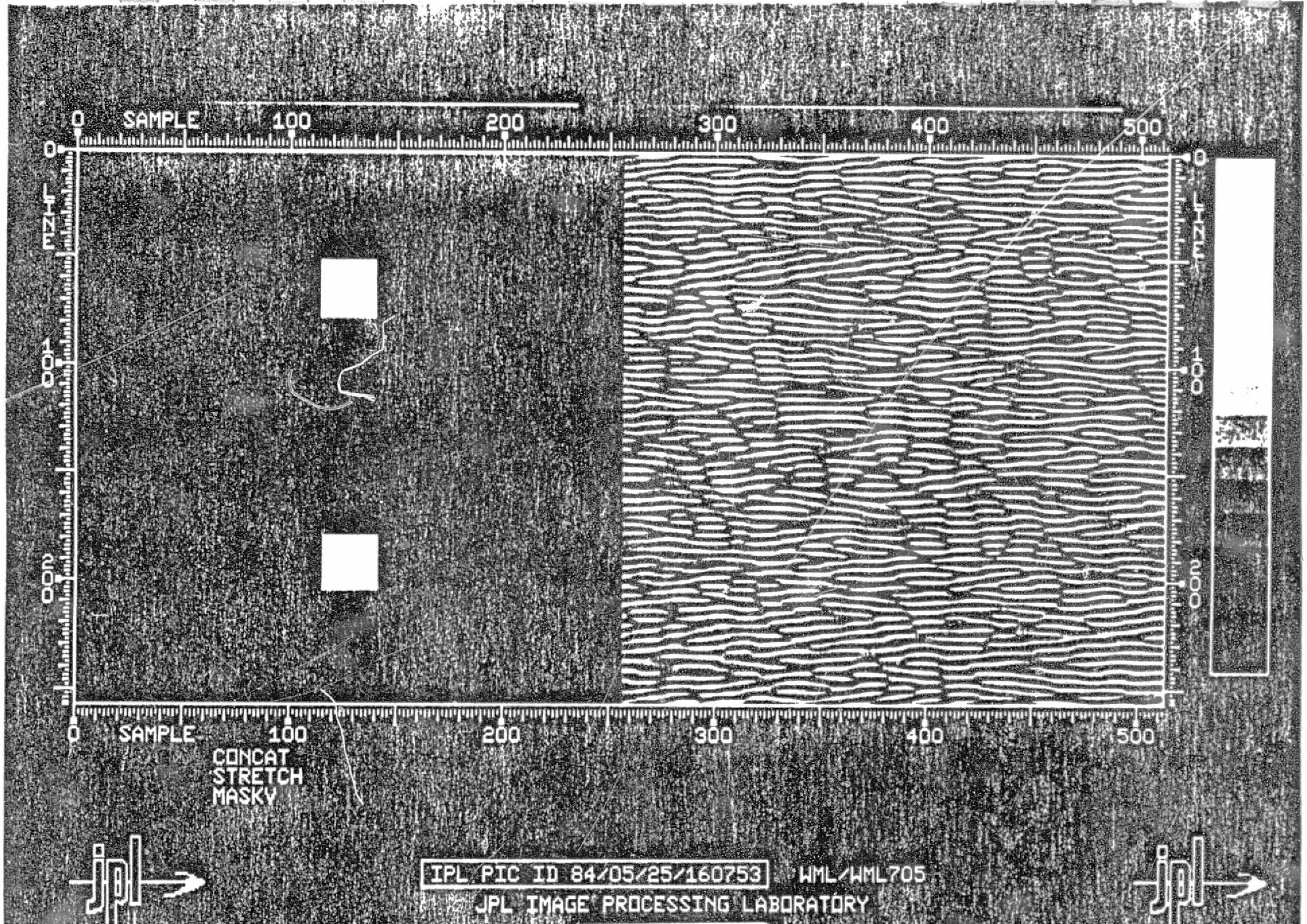

0
SAMPLE
100
200
300
400
500
LINE
0
100
200
CONCAT
STRETCH
MASKY
IPL PIC ID 84/05/25/160753
WML/WML705
JPL IMAGE PROCESSING LABORATORY

Figure 3.8

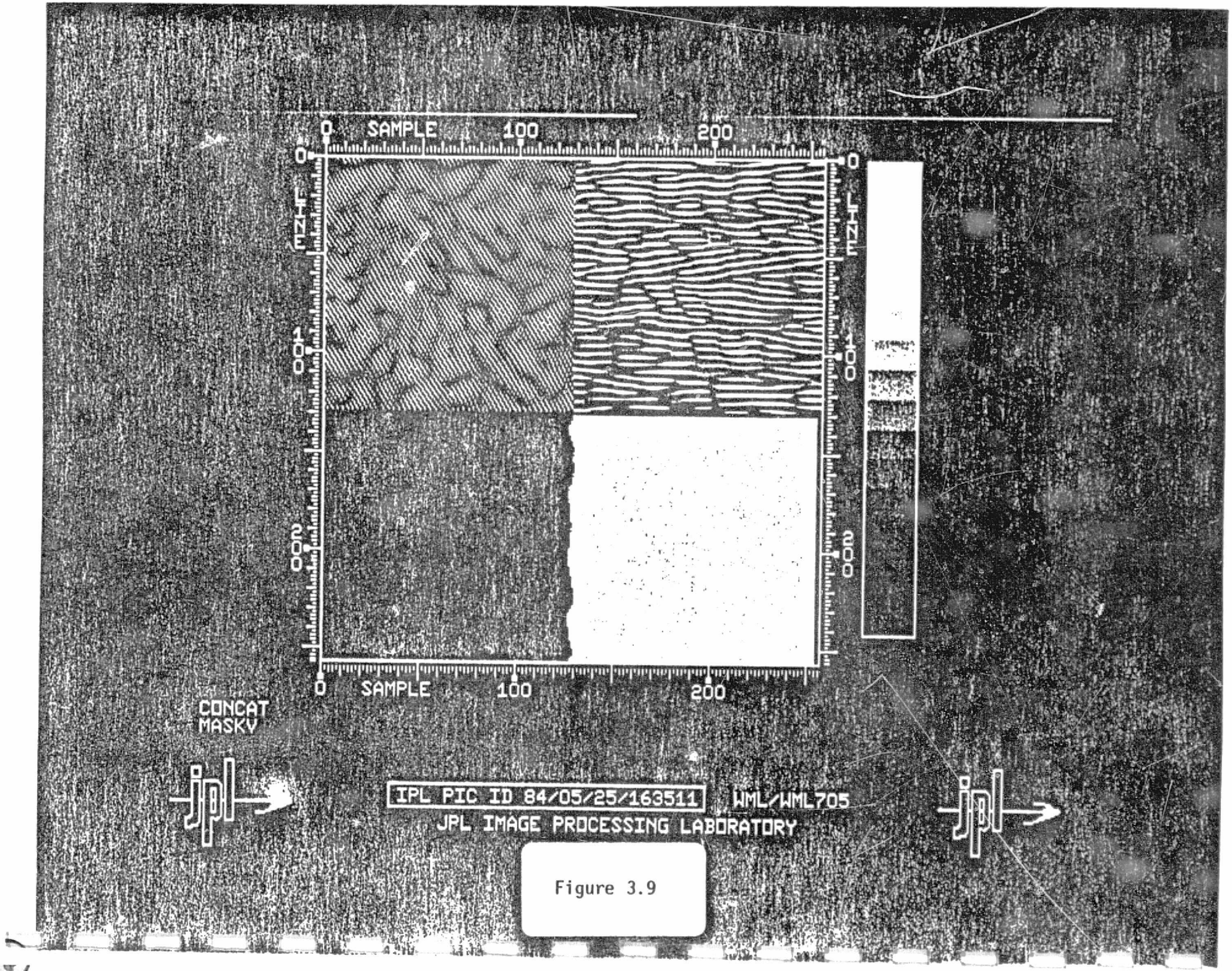

0
SAMPLE
100
200
0
LINE
100
200
0
LINE
100
200
0
SAMPLE
100
200
CONCAT
MASKV
IPL PIC ID 84/05/25/163511
WML/WML705
JPL IMAGE PROCESSING LABORATORY

Figure 3.9

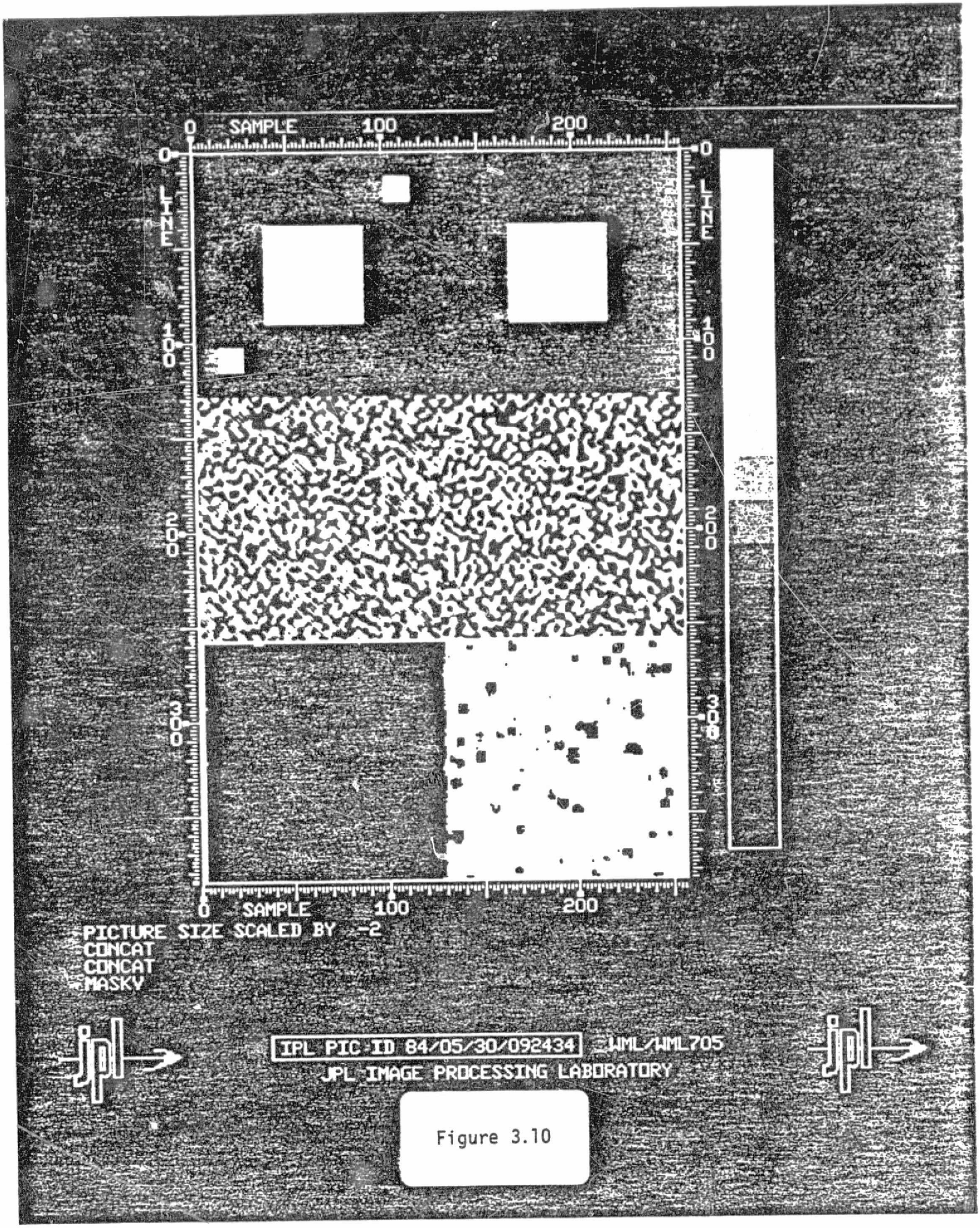

0
SAMPLE
100
200
0
LINE
100
200
300
0
LINE
100
200
300
0
SAMPLE
100
200
PICTURE SIZE SCALED BY -2
CONCAT
CONCAT
MASKY
jpl
IPL PIC ID 84/05/30/092434
WML/WML705
JPL IMAGE PROCESSING LABORATORY
jpl

Figure 3.10

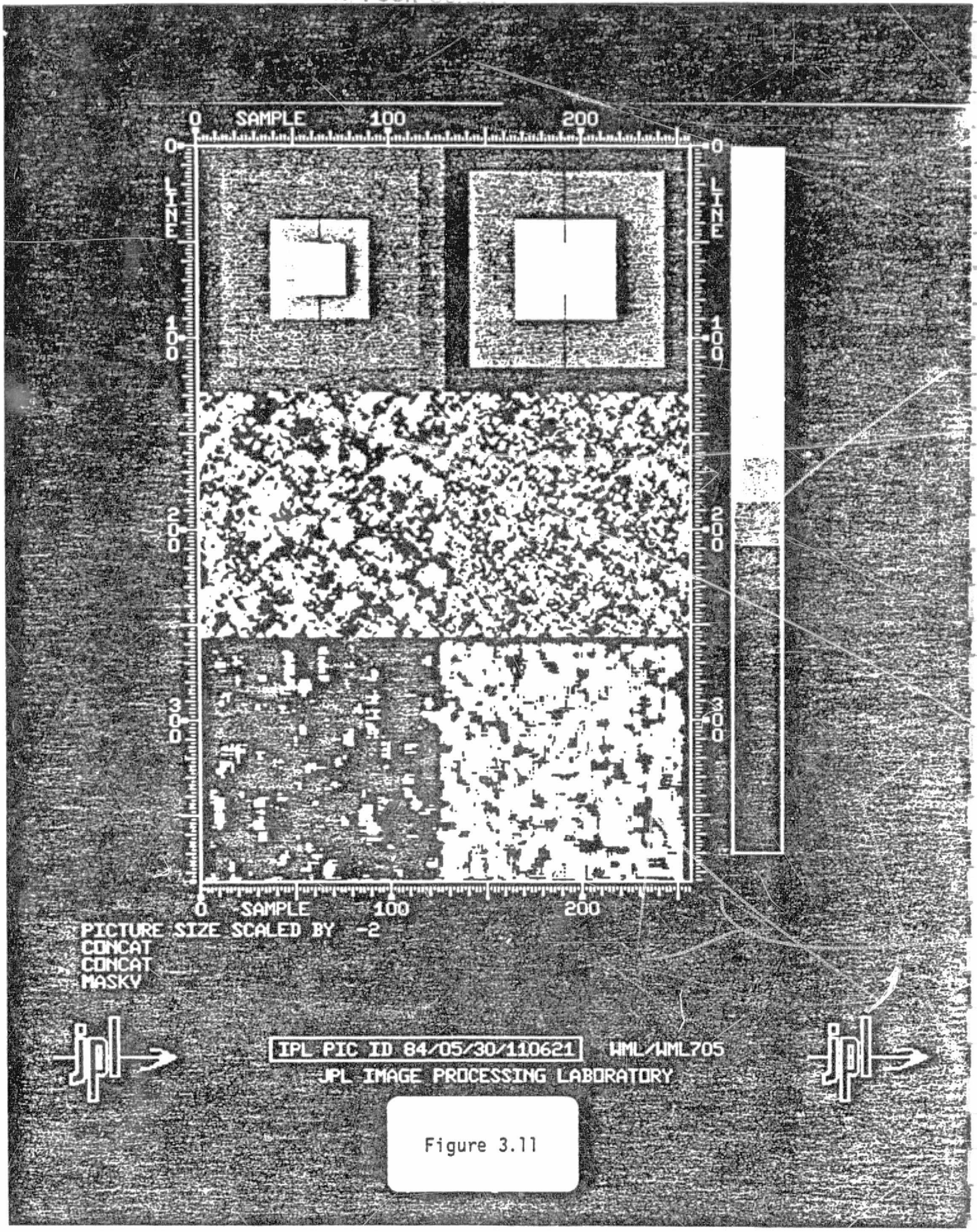

0
SAMPLE
100
200
LINE
0
100
200
300
0
SAMPLE
100
200
PICTURE SIZE SCALED BY -2
CONCAT
CONCAT
MASKY
jpl
IPL PIC ID 84/05/30/110621
WML/WML705
JPL IMAGE PROCESSING LABORATORY
jpl

Figure 3.11

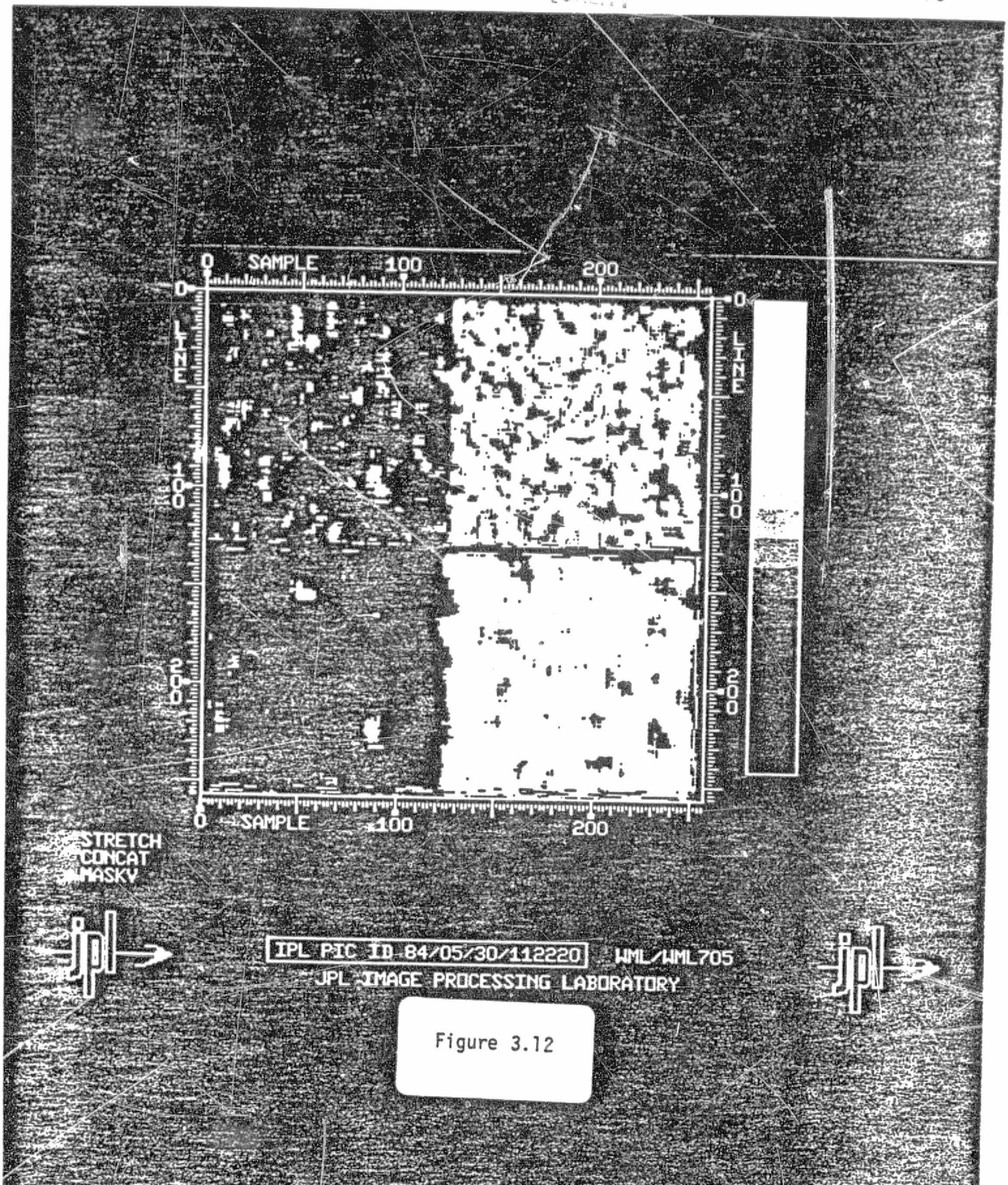

0
SAMPLE
100
200
LINE
0
100
200
0
SAMPLE
100
200
STRETCH
CONCAT
MASKV
IPL PIC ID 84/05/30/112220
WML/WML705
JPL IMAGE PROCESSING LABORATORY

Figure 3.12

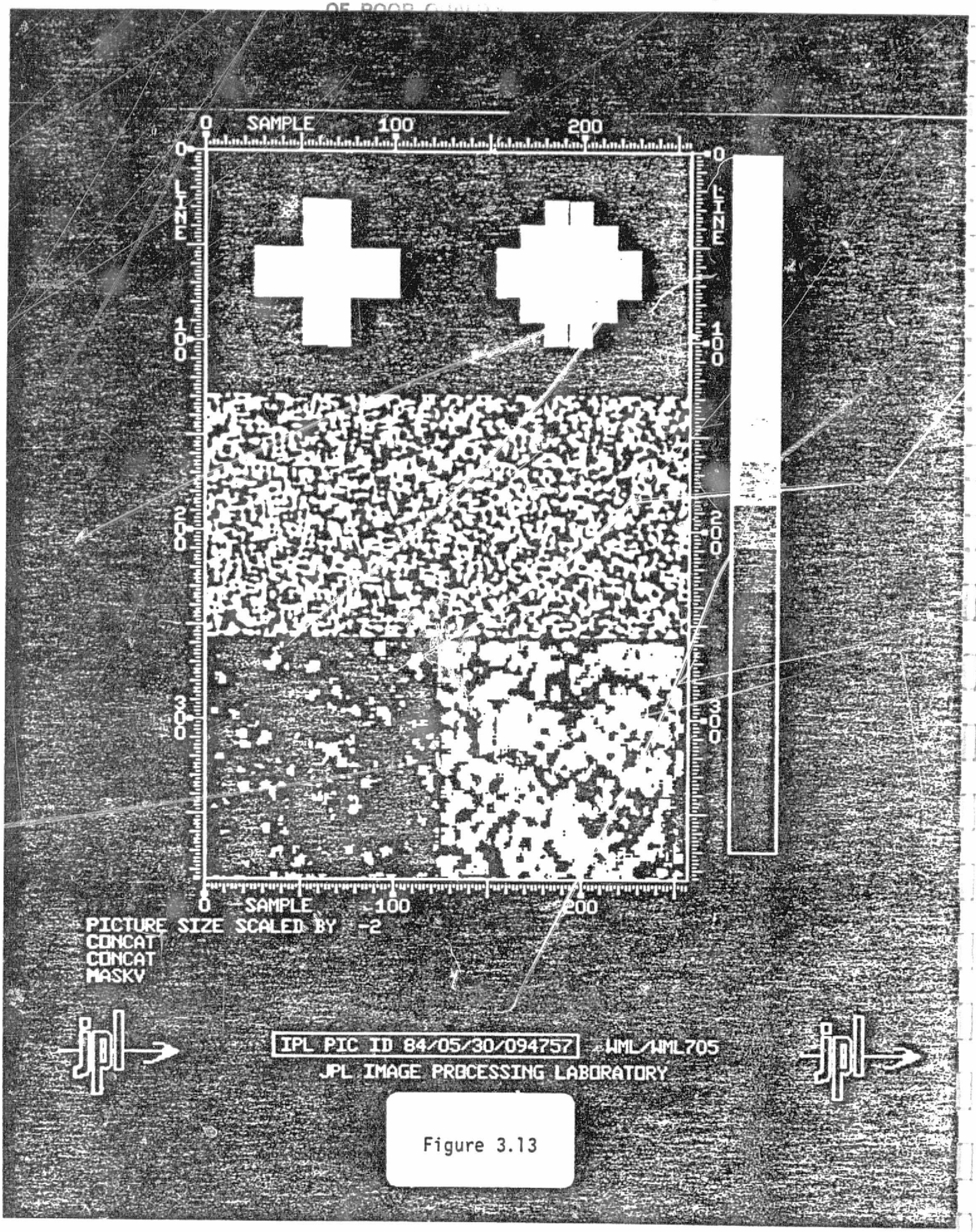

0
SAMPLE
100
200
0
LINE
100
200
300
0
LINE
100
200
300
0
SAMPLE
100
200
PICTURE SIZE SCALED BY -2
CONCAT
CONCAT
MASKV
jpl
IPL PIC ID 84/05/30/094757
WML/WML705
JPL IMAGE PROCESSING LABORATORY
jpl

Figure 3.13

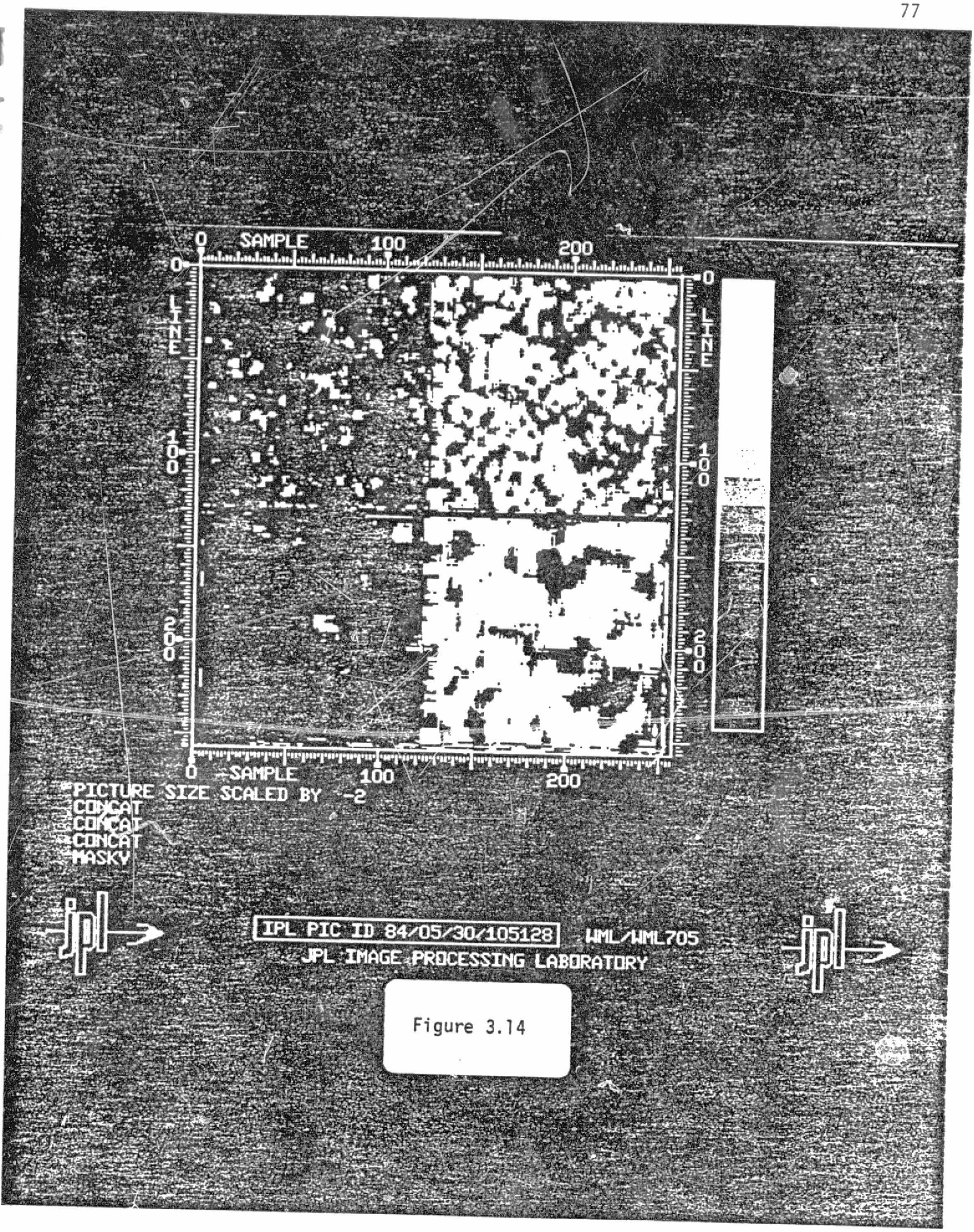
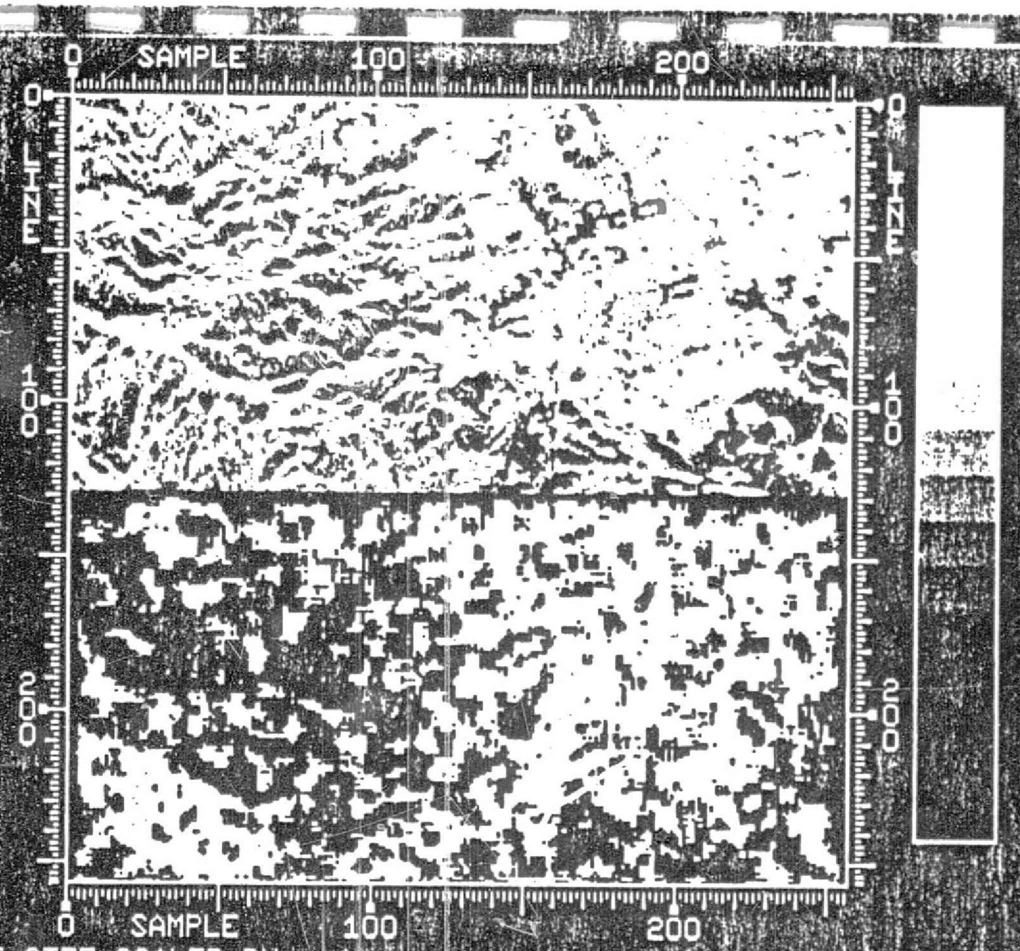

0
SAMPLE
100
200
0
LINE
100
200
0
LINE
100
200
0
SAMPLE
100
200
PICTURE SIZE SCALED BY -2
CONCAT
CONCAT
CONCAT
MASKY
IPL PIC ID 84/05/30/105128
WML/WML705
JPL IMAGE PROCESSING LABORATORY
jpl
jpl

Figure 3.14



```
PICTURE SIZE SCALED BY  -4
FLOT - COUNTERCLOCKWISE
 GAUSSIAN STRETCH
FLOT - CLOCKWISE
FLOT - VERTICAL
CONCAT
MASKV
```

IPL PIC ID 84/05/24/110430   WML/WML705

JPL IMAGE PROCESSING LABORATORY

Figure 3.15

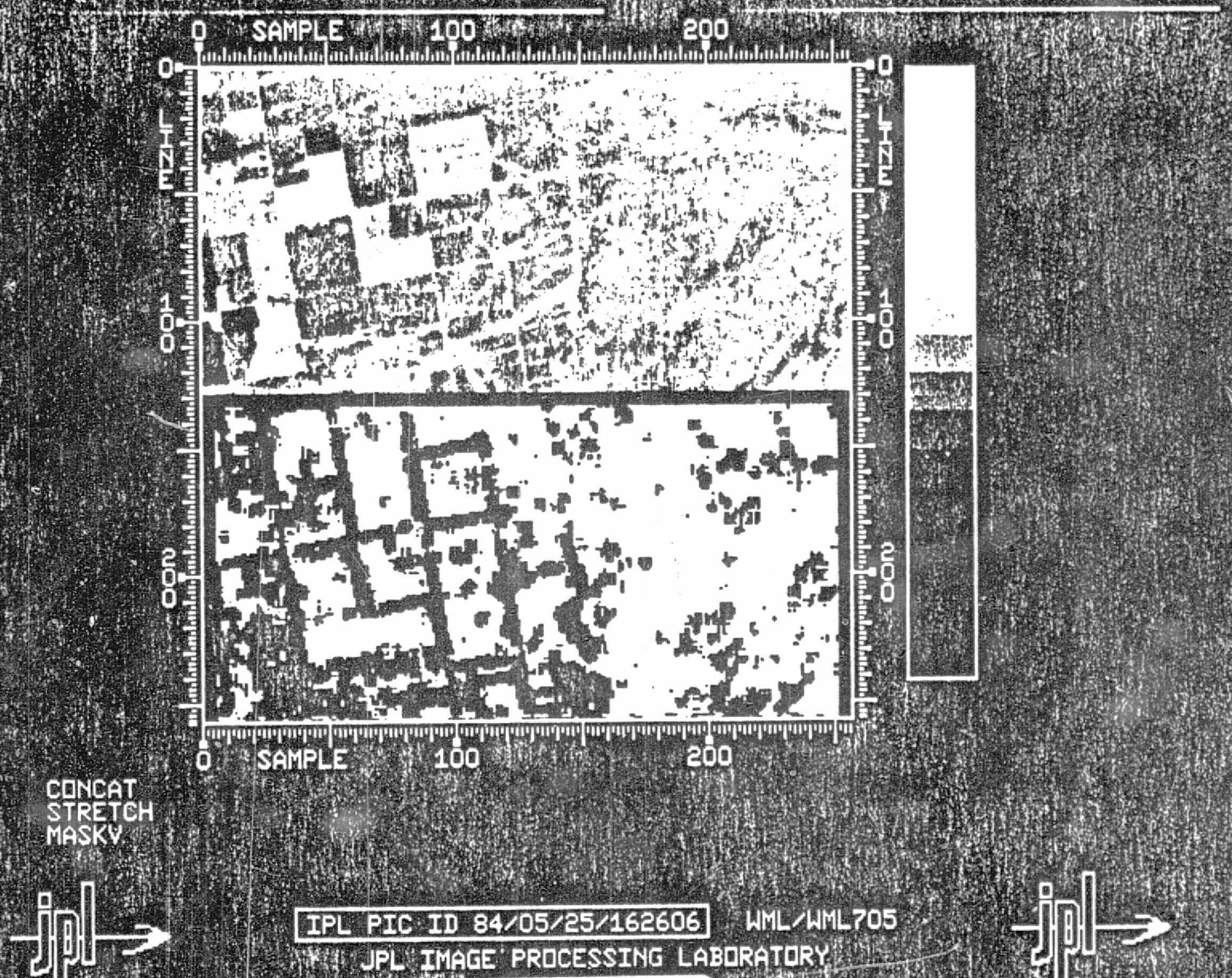

0
SAMPLE
100
200
LINE
0
100
200
CONCAT
STRETCH
MASKV
IPL PIC ID 84/05/25/162606
WML/WML705
JPL IMAGE PROCESSING LABORATORY

Figure 3.16

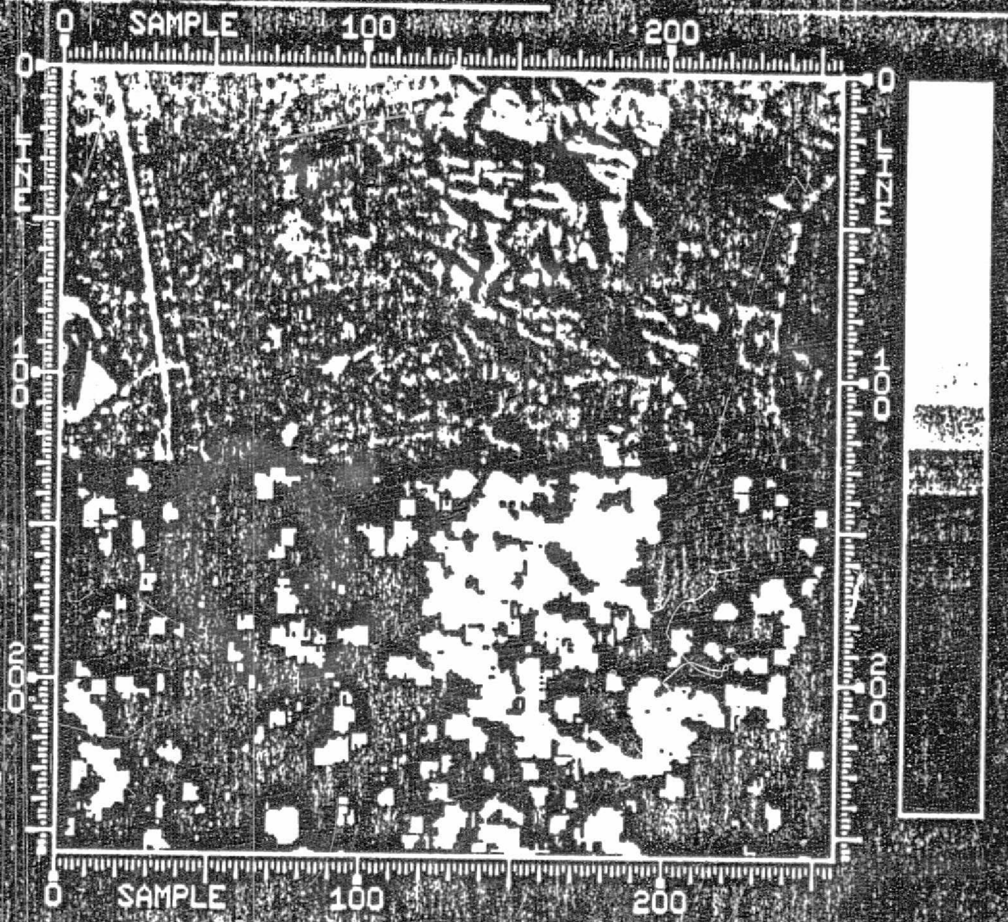


Figure 3.17

ORIGINAL PAGE IS
OF POOR QUALITY

techniques can be applied to make them modellable. This research will examine the pre-processing techniques that can convert images into modellable textures.

The second question may be answered simply that the training area should be able to represent the statistical and spectral characteristics of the entire texture class, the autoregression window size should be large enough to capture the spatial relation among pixels for the given class and the smoothing window size should be selected so that the scene class boundary confusion and false classification can be minimized. However, the precise expression of these parameter sizes as functions of the texture characteristics are not yet defined.

In order to understand the relation between the parameter sizes and the texture characteristics, a large set of images (synthesized textures and natural textures) with various statistical and spectral characteristics will be applied and the segmentation performance will be evaluated with respect to parameter sizes.

The third question can be answered only after the human perception is understood quantitatively. It may be impossible to understand the way how a human perceives textures completely. In this research, more detailed relations between the human perception and the statistical and spectral characteristics of the natural scenes will be examined.

REFERENCES

[1] R. Adler, The Geometry of Random Fields, John Wiley and Sons, Ltd., New York, 1981

[2] P. Brodatz, Textures: A. Photographic Album for Artists and Designers, Dover, New York, 1966.

[3] P. Brodatz, Wood and Wood Grains: A Photographic Album for Artists and Designers, Dover, New York, 1971.

[4] P. Brodata, Land, Sea and Sky: A Photographic Album for Artists and Designers, Dover, New York, 1976.

[5] N. Chomsky, "Form and Meaning in Natural Languages", p. 65-86, Communication: A Discussion at the Nobel Conference, edited by John D. Roslansky, North-Holland Publishing Co., New York, 1969.

[6] M. Daily, S. Williams, W. Stromberg, "Lithologic Discrimination by Fourier Analysis of Image Texture", Proceedings of the Symposium on Machine Processing of Remotely Sensed Data, Purdue, 1979, p. 430.

[7] D. Evans and W. Stromberg, "Development of Texture Signatures in Radar Images", Digest of the 1983 International Geoscience and Remote Sensing Symposium, San Francisco, CA., Aug. 31-Sept. 2, 1983.

[8] W. Finch, Jr., S. Freden, P. Lowman, Jr., N. Short, Mission to Earth: Landsat Views the World, Scientific and Technical Information Office, NASA, Washington, D. C., 1976.

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

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

[11] H. Helson, D. Lowdenslager, "Prediction Theory and Fourier Series in Several Variables, Acta Math., Vol. 99, p. 165-201, 1958.

[12] H. Helson, D. Lowdenslager, "Prediction Theory and Fourier Series in Several Variables II", Acta Math., Vol. 106, p. 175-213, 1961.

[13] B. Julesz, "Visual Pattern Discrimination", IRE Transactions on Information Theory, Vol. 8, No. 2, p. 84-92, February 1982.

[14] B. Julesz, "Perceptual Limits of Texture Discrimination and Their Implications to Figure-Ground Separation", p. 205-216, Formal Theories of Visual Perception, edited by E. Leevwenberg and H. Buffart, John Wiley and Sons, New York 1978.

[15] A. Kak, A. Rosenfeld, Digital Image Processing, Academic Press, New York 1976.

[16] R. Kardon, R. Kessel, Tissues and Organs: A Test-Atlas of Scanning Electron Microscopy, W. H. Freeman and Co., San Francisco, 1979.

[17] B. Mandelbrot, Fractals: Form, Chance, and Dimension, W. H. Freeman and Co., San Francisco, 1977.

[18]B. Mandlebrot, The Fractal Geometry of Nature, W. H. Freeman and Co., San Francisco, 1982.

[19] M. Naraghi, "Autoregressive Models for Use in Scene Segmentation", p. 93-121, Proceedings of the NASA Symposium of Mathematical Pattern Recognition and Image Analysis, July 1982.

[20] A. Pentland, "Fractal-Based Description of Natural Scenes", p. 184-192, Proceedings of Image Understanding Workshop Sponsored by the Defense Advanced Research Projects Agency, Science Applications, Inc., Report Number SAI-84-176-WA, Lee S. Baumann, editor, June 1983.

[21] J. Serra, Image Analysis and Mathematical Morphology, Academic Press, New York, 1982.

[22] D. Tjostheim, "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, January 1981.

[23] D. Tjostheim, "Unilateral Models for Stochastic Lattice Processes", p. 268-281, Lecture Notes in Control and Information Sciences, Vol. 49, edited by G. Kallianpur, Springer Verlag, New York 1983.

[24] E. Vanmarcke, Random Fields: Analysis and Synthesis, MIT Press, 1983.

[25] M. Yadrenki, Spectral Theory of Random Fields, Optimization Software, Inc., 1983.

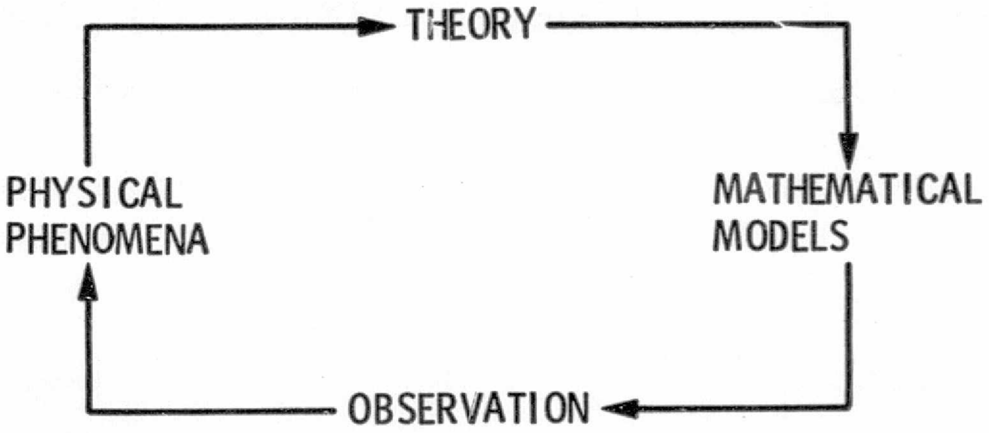
JPL

PHILOSOPHICAL INSPIRATION

"MATHEMATICS IS THE MOST POWERFUL TECHNIQUE FOR THE UNDERSTANDING OF PATTERN AND FOR THE ANALYSIS OF THE RELATIONS OF PATTERNS."

Alfred North Whitehead

PARADIGM FOR SCIENTIFIC METHOD



THEORY

PHYSICAL PHENOMENA

MATHEMATICAL MODELS

OBSERVATION

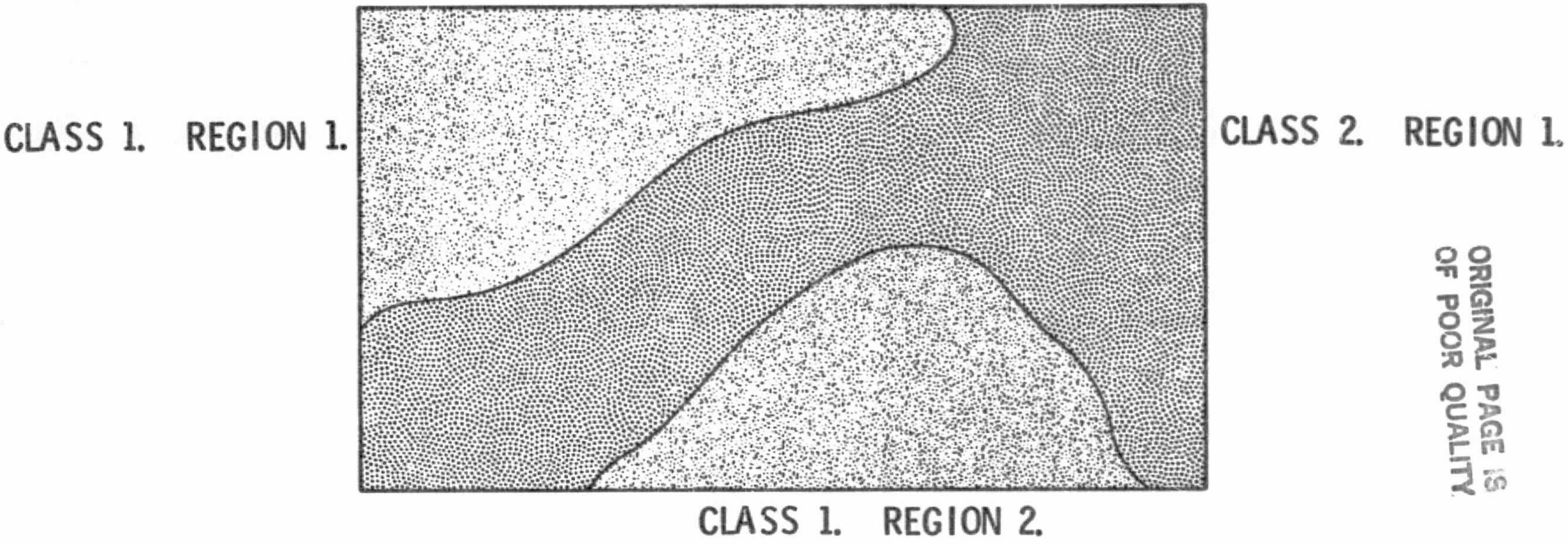
1. EXPERIENCE WITH P.P. LEADS TO A THEORY - A METAPHOR BETWEEN THE P.P. AND SET OF AXIOMS DEFINING A M.M. - BASED ON INDUCTION.

2. MATHEMATICAL ELABORATION OF THE M.M. EXPLORES THE LOGICAL CONSEQUENCES-THEOREMS - OF THESE AXIOMS - BASED ON DEDUCTION.

3. THESE THEOREMS CORRESPOND TO PREDICTABLE BEHAVIOR OF THE P.P. LEADING TO OBSERVATIONS TESTING THE THEORY - BASED ON EXPERIMENTATION.

JPL

PHYSICAL PHENOMENON: TEXTURE



MATHEMATICAL MODEL

I. EACH TEXTURE CLASS CORRESPONDS TO A STATIONARY, ERGODIC RANDOM FIELD $X: \Omega \times L \longrightarrow R$, Ω = PROBABILITY SPACE, L = LATTICE, R = REAL NUMBERS.

II. THE IMAGE BRIGHTNESS VALUES OVER EACH CONNECTED REGION $D \subset L$ FOR THAT CLASS CORRESPONDS TO A REALIZATION OF X OVER D, i.e. $X: \{\omega\} \times D \longrightarrow R, \ \omega \in \Omega$.

IMAGE SEGMENTATION PROBLEM
JPL IDENTIFICATION OF REGIONS AND CLASSES
APPROACH USING MODEL

1. UTILIZE TRAINING AREAS SELECTED BY HUMAN ANALYST TO ESTIMATE THE PARAMETERS (JOINT PROBABILITIES) FOR EACH CLASS

2. FOR EACH 'BLOCK' OF PIXELS IN THE IMAGE, APPLY BAYES' THEOREM TO CALCULATE THE A POSTERIORI (CONDITIONED ON IMAGE BRIGHTNESS VALUES OVER THE BLOCK) PROBABILITIES FOR THE BLOCK TO LIE WITHIN A REGION BELONGING TO EACH CLASS

3. USE 'CONTEXTUAL CUES' TO PERFORM FINAL SEGMENTATION

JPL MATHEMATICAL ELABORATION OF MODEL

LET B BE ANY M BY M BLOCK OF PIXELS LYING WITHIN A REGION FOR SOME ONE OF N TEXTURE CLASSES. LET $\underline{v}$ BE A VECTOR REPRESENTING THE IMAGE VALUES OVER B, LET $p(k)$, $k = 1, \ldots, N$ BE THE A PRIORI PROBABILITIES FOR THE REGION CONTAINING B TO BELONG TO THE k-th TEXTURE CLASS, LET $p(\underline{v} | k)$, $k = 1, \ldots, N$ BE THE CONDITIONAL PROBABILITIES (OR DENSITIES) FOR $\underline{v}$, AND LET $p(k | \underline{v})$ BE THE A POSTERIORI PROBABILITIES.

THEOREM 1 (BAYES) $p(k | \underline{v}) = p(k)\, p(\underline{v} | k) \Big/ \sum_{j=1}^{N} p(j)\, p(\underline{v} | j)$

DEFINITION (SHANNON) CONDITIONAL ENTROPY $H(\text{CLASS} | \underline{v}) = \underset{\underline{v}}{\text{EXP}} \left(- \sum_{j=1}^{N} p(j|\underline{v}) \log p(j|\underline{v}) \right)$

THEOREM 2 (VERY EASY) $0 \leqq H(\text{CLASS}\ \ \underline{v}) \leqq \text{LOG}\ N$ AND ⟶ 0 iff PROB CORRECT CLASSIFICATION ⟶ 1

LET m_k = MEAN OF k-th CLASS AND LET C_k = COVARIANCE OF $\underline{v}$ FOR k-th CLASS

THEOREM 3 (CALCULUS OF VARIATIONS) FOR m_k, C_k AND $p(k)$ FIXED,

$\mathbf{H}(\underline{\mathbf{V}}) + H(\text{CLASS} | \underline{v})$ IS <u>MAXIMIZED</u> iff $p(\underline{v} | j)$ IS GAUSSIAN FOR ALL $j = 1, \ldots, N$

SECOND ORDER STATISTICS AND COMPUTATIONAL COMPLEXITY

DEFINITION. LET $X: \Omega \times L \longrightarrow R$ BE A STATIONARY RANDOM FIELD. THE MEAN $m = \mathrm{EXP}\,(X\,(\omega,\ q))$ FOR ANY $q \in L$ AND THE AUTOCOVARIANCE FUNCTION IS

$$A: L \longrightarrow R \text{ WHERE } A\,(p) = \underset{\omega \in \Omega}{\mathrm{EXP}} \left(\left[X\,(\omega,\ p + q) - m \right] \left[X\,(\omega,\ q) - m \right] \right), \text{ ANY } q \in L.$$

THE MEAN AND AUTOCOVARIANCE FUNCTION COMPRISE THE SECOND ORDER STATISTICS OF A STATIONARY RANDOM FIELD.

WE CONSIDER SECOND ORDER STATISTICS BECAUSE THEY

1) COMPLETELY CHARACTERIZE (JOINTLY) GAUSSIAN RANDOM FIELDS

2) ANY ALGORITHM WHICH YIELDS A HIGH PROBABILITY OF CORRECT CLASSIFICATION USING SECOND ORDER STATISTICS, UNDER THE 'GAUSSIAN ASSUMPTION', WILL ALSO PERFORM WELL IF THIS ASSUMPTION IS NOT VALID (BY THEOREM 3)

3) RESTRICTION TO SECOND ORDER STATISTICS REDUCES COMPUTATIONAL COMPLEXITY IMMENSELY

JPL FOURIER TRANSFORMS AND SPECTRAL ANALYSIS

DEFINITION. $F_1 : L \longrightarrow R$ AND $F_2\ (u,\ v),\ 0 \leq u \leq 1,\ 0 \leq v \leq 1$

ARE FOURIER TRANSFORM PAIRS, $F_1 = \mathscr{F}(F_2),\ F_2 = \mathscr{F}^{-1}(F_1)$, if

$$F_1\ (a,\ b) = \int_0^1 \int_0^1 F_2\ (u,\ v)\ \mathrm{EXP}\ (-2\pi iau - 2\pi ibv)\, du dv,\quad i^2 = -1$$

$$F_2\ (u,\ v) = \sum_{(a,\ b) \in L} F_1\ (a,\ b)\ \mathrm{EXP}\ (2\pi iau + 2\pi ibv)$$

DEFINITION. THE SPECTRUM S OF A STATIONARY RANDOM FIELD X IS $S\ (u,\ v) = \mathscr{F}\ (A)\ (u,\ v)$ WHERE A IS THE AUTOCOVARIANCE FUNCTION FOR X.

REMARK. THE SPECTRUM IS ABSOLUTELY NOT THE FOURIER TRANSFORM OF AN IMAGE

JPL

FILTERING AND AUTOREGRESSION

DEFINITION. LET $F: L \longrightarrow R$ BE ZERO EXCEPT ON A FINITE SET $S \subset L$ AND LET $X: \Omega \times L \longrightarrow R$ BE A RANDOM FIELD. THE F-FILTER OF X IS A RANDOM FIELD $F * X: \Omega \times L \longrightarrow R$ GIVEN BY

$$F * X(\omega, p) = \sum_{q \in L} F(q) X(\omega, p-q)$$

THE F-FILTER IS <u>QUARTER</u> OR <u>HALF-PLANE</u> IF S LIES IN A QUARTER OR HALF-PLANE, IT IS CALLED <u>S-AUTOREGRESSIVE</u> IF $(0, 0) \in S$ AND $F(0, 0) = 1$ AND THE VARIANCE OF $F * X$ (AUTOCOVARIANCE FUNCTION OF $F * X$ AT (0, 0)) IS MINIMIZED (WITH RESPECT TO ALL FILTERS WHICH ARE 1 AT (0, 0) AND ZERO OUTSIDE S) THEOREM 4 (CLASSICAL) SPECTRUM $(F * X) = |\mathscr{F}(F)|^2$ SPECTRUM (X)

QUARTER PLANE AUTOREGRESSIVE 'MODELS'

A POPULAR MODEL FOR IMAGE TEXTURES CONSISTS OF A STATIONARY RANDOM FIELD X SATISFYING THE FOLLOWING:

- THERE EXIST A QUARTER PLANE $Q \subset L$, A FINITE SUBSET $S \subset Q$, AND AN S-AUTOREGRESSIVE FILTER F SUCH THAT THE AUTOCOVARIANCE OF $F * X$ IS ZERO EXCEPT AT $(0, 0) \in L$
- THE VARIANCE OF $F * X$ IS MINIMUM WITH RESPECT TO ALL T-AUTOREGRESSIVE FILTERS $G * X$ FOR ALL $T \subset Q$

THEOREM 5 (EASY) SPECTRUM (X) $= \sigma_S^{\,2} / |\mathscr{F}(F)|^2$

THEOREM 6 (SEVERAL COMPLEX VARIABLES) THE FILTER F IS STABLE (ROOTS OF $\mathscr{F}(F)(Z_1, Z_2)$ HAVE MODULUS > 1)

THEOREM 7 (ALGEBRAIC FIELD THEORY) $\mathscr{F}\Big(\text{LOG (SPECTRUM (X))}\Big)(p) = 0$

IF $p \notin Q$ AND $-p \notin Q$

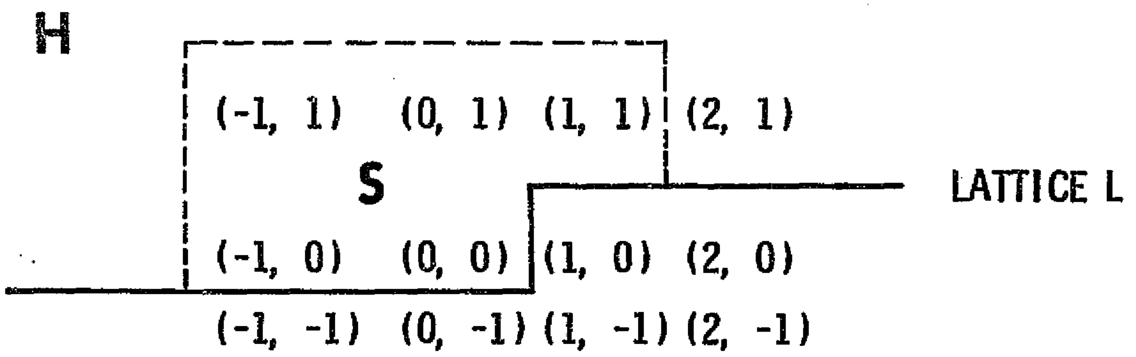
REMARK. THESE MODELS SHOULD BE LESS POPULAR

JPL HALF-PLANE AUTOREGRESSIVE FILTERING

THEOREM 8 (HELSON, H. AND LOWDENSLAGER, D.). IF X IS <u>ANY</u> RANDOM FIELD AND H ⊂ L IS ANY HALF-PLANE THEN

$$\lim_{\substack{S \subset H \\ S \text{ GETS LARGE}}} \text{SPECTRUM}\,(F_S * X) \longrightarrow \sigma^2$$

WHERE (SZEGÖ) $\sigma^2 = \text{EXP}\left(\int_0^1\int_0^1 \text{LOG}\,(\text{SPECTRUM}\,(X)\,(u,\ v))\,du\,dv\right)$



(-1, 1) (0, 1) (1, 1) (2, 1)

(-1, 0) (0, 0) (1, 0) (2, 0)

(-1, -1) (0, -1) (1, -1) (2, -1)

JPL

STATISTICAL SCENE SEGMENTATION ALGORITHM

STEP 1. SELECT SCENE - NATURAL OR SYNTHESIZED

STEP 2. SELECT TRAINING AREAS AND ESTIMATE MEANS AND AUTOCOVARIANCES

STEP 3. CHOOSE A FILTER WINDOW S LYING IN A HALF-PLANE AND CALCULATE AUTOREGRESSIVE FILTERS $F_k: S \rightarrow R$ FOR EACH TEXTURE CLASS $1 \leqq k \leqq N$

STEP 4. FILTER ORIGINAL SCENE WITH F_k TO OBTAIN X_k FOR $1 \leqq k \leqq N$

STEP 5.* APPLY TRANSFORMATION $X_k \rightarrow Y_k = \mathrm{LOG}\sqrt{2\pi V_k} + (X_k - M_k)^2/2\,V_k$ FOR $1 \leqq k \leqq N$

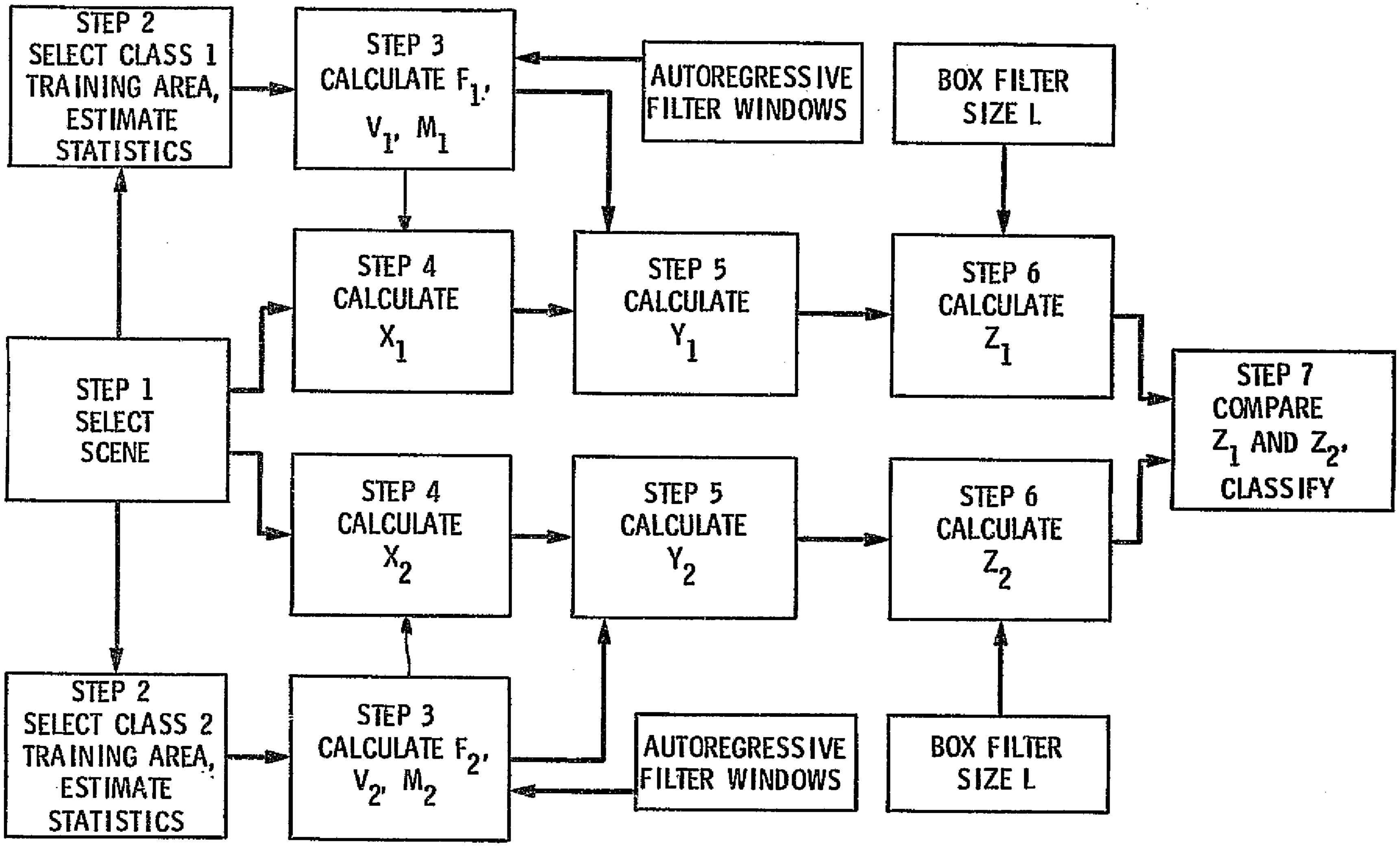
STEP 6. CHOOSE BOX SIZE L AND APPLY L x L BOX FILTER TO Y_k TO OBTAIN Z_k FOR $1 \leqq k \leqq N$

STEP 7. CLASSIFY PIXEL (a, b) INTO j-th CLASS IF $Z_j\,(a, b) \leqq Z_k\,(a, b)$ FOR ALL $k \neq j$

* V_k AND X_k IS VARIANCE AND MEAN OF X_k AS CALCULATED USING PARAMETERS FROM STEPS 2 AND 3.

JPL

ALGORITHM FLOW CHART



STEP 1 SELECT SCENE

STEP 2 SELECT CLASS 1 TRAINING AREA, ESTIMATE STATISTICS

STEP 3 CALCULATE F_1, V_1, M_1

AUTOREGRESSIVE FILTER WINDOWS

STEP 4 CALCULATE X_1

STEP 5 CALCULATE Y_1

BOX FILTER SIZE L

STEP 6 CALCULATE Z_1

STEP 2 SELECT CLASS 2 TRAINING AREA, ESTIMATE STATISTICS

STEP 3 CALCULATE F_2, V_2, M_2

AUTOREGRESSIVE FILTER WINDOWS

STEP 4 CALCULATE X_2

STEP 5 CALCULATE Y_2

BOX FILTER SIZE L

STEP 6 CALCULATE Z_2

STEP 7 COMPARE Z_1 AND Z_2, CLASSIFY

JPL

ALGORITHM PROPERTIES

- AS AUTOREGRESSIVE WINDOW SIZE AND BOX SIZE GET LARGE, PERFORMANCE APPROACHES BAYESIAN (OPTIMAL) CLASSIFIER FOR JOINTLY GAUSSIAN TEXTURES
- PERFORMANCE STILL VALID FOR NON-GAUSSIAN TEXTURES, AND ALGORITHM IS OPTIMAL FOR UTILIZING SECOND ORDER STATISTICS
- ALGORITHM ACHIEVES PERFORMANCE WITH MINIMAL COMPUTATIONAL COMPLEXITY

ALL THESE ASSERTIONS ARE MATHEMATICALLY PROVABLE!

JPL

EXTENSIONS OF THEORETICAL RESEARCH

- DEVELOP OPTIMAL NONLINEAR CLASSIFICATION ALGORITHM (DONE)
- EXTEND MODEL TO MULTISPECTRAL IMAGERY (DONE)
- EXTEND MODEL TO SUBSUME FRACTAL MODELS (DONE)
- EXTEND MODEL TO SUBSUME STRUCTURAL MODELS (DONE)
- DETERMINE OPTIMAL AUTOREGRESSIVE WINDOW SIZE, BOX FILTER SIZE, AND THEIR DEPENDENCE ON CONTEXTUAL KNOWLEDGE (INCOMPLETE)
- SYNTHESIZE STABLE HALF-PLANE AUTOREGRESSIVE FILTERS (DONE) AND UTILIZE THEM FOR OPTIMAL IMAGE COMPRESSION (INCOMPLETE)

BAYESIAN ESTIMATION OF NORMAL MIXTURE PARAMETERS

by

Charles Peters
Department of Mathematics
University of Houston

ABSTRACT

A Bayesian, or penalized maximum likelihood, approach to the problem of estimating the parameters of a mixture of multivariate normal distributions is proposed. The Bayesian formulation eliminates the problem of singularities in the likelihood function and results in an attractive EM-like procedure. Although the question of consistency is not settled, it is suggested that the proposed method has certain advantages over both the constrained and unconstrained maximum likelihood procedures.

1. INTRODUCTION

Let $X_1, \cdots, X_n$ be a random sample from a finite mixture density

$$f(x|\theta) = \sum_{i=1}^{m} q_i f_i(x|\theta_i),$$

where the component densities are d-dimensional multivariate normal and the mixing propostions q_i satisfy $q_i \geq 0$, $\sum_{i=1}^{m} q_i = 1$. We let $\theta_i = (\mu_i, \Sigma_i)$ denote the mean and covariance of the $i^{\underline{th}}$ component density and let θ denote the aggregate of all the parameters involved in the mixture density, including $q = (q_1, \cdots, q_m)$. We assume throughout that m is known. It will be convenient to consider also the precision matrix $\tau_i = \Sigma_i^{-1}$, and we sometimes let $\theta_i = (\mu_i, \tau_i)$.

Maximum lieklihood is the method of estimating the parameters θ which has recently attracted the most interest, [8]. According to this method, the estimate $\hat{\theta} = \hat{\theta}(X_1, \cdots, X_n)$ is the parameter value which maximizes the log likelihood function

$$\ell(\theta) = \sum_{i=1}^{n} \log f(X_i(\theta).$$

Unfortunately, as simple examples show, the function $\ell(\theta)$ is unbounded, and one must consider local maximizers of $\ell(\theta)$ or else modify $\ell(\theta)$ in some way so as to produce a global maximizer. Hathaway [5] took the second approach in proposing a constrained maximum likelihood estimator. For mixtures of univariate normal densities, he developed an effective computational procedure for finding a maximum of $\ell(\theta)$ subject to the constraints

$$\sigma_i \geq c\,\sigma_{i+1} \qquad i = 1, \cdots, m,$$

where σ_i is the $i^{\underline{th}}$ standard deviation, $\sigma_{m+1} = \sigma_1$, and $c > 0$ is a constant, chosen by the user. He also proved that $\ell(\theta)$ has a global maximizer, subject to the above constraints, and that the global maximizer is a strongly consistent estimator, as long as the true parameter satisfies the given constraints. Redner [7], mentions a penalized likelihood function of the form

$$\ell(\theta) - \lambda \sum_{i=1}^{m} ||\tau_i||^k ,$$

where λ, $k > 0$ and $||\tau_i||$ is a norm on symmetric $d{\times}d$ matrices.

Bayes solutions for common loss functions, such as quadratic loss, appear to be computationally infeasible [3]. For example, assuming that the mixing propostions are the only unknown parameters, and using the Dirichlet prior distribution given in the next section, there is an explicit formula for the Bayes solution with quadratic loss. However, it contains m^n terms and is not useful except for very small sample sizes. The method proposed in the next section utilizes a prior density $g(\theta)$ of a certain form on the parameter θ and takes as the estimator the mode of the posterior density

$$g(\theta|X_1,\cdot\cdot,X_n) = \frac{[\prod_{j-1}^{n} f(X_j|\theta)]g(\theta)}{\int_\theta [\prod_{j=1}^{n} f(X_j|\vartheta)]g(\theta)d\theta} .$$

Equivalently, the estimator maximizes the <u>penalized</u> log likelihood function,

$$\ell_1(\theta) = \ell(\theta) + \log g(\theta) .$$

Such a procedure can be justified in Bayesian theory as being the limit as $\epsilon \to 0$ of Bayes solutions $\hat{\theta}_\epsilon$ corresponding to 0-1 loss functions

$$L_\epsilon(\theta,\hat{\theta}) = \begin{cases} 0 & \text{if } ||\theta-\hat{\theta}|| < \epsilon \\ 1 & \text{if } ||\theta-\hat{\theta}|| \geq \epsilon . \end{cases}$$

It will be seen that $\ell_1(\theta)$ is similar to, but is more elaborate than the penalized likelihood function suggested by Redner.

2. THE PRIOR DISTRIBUTION

Recall that $q = (q_1, \cdots, q_m)$ is the vector of mixing propostions and that $\theta_i = (\mu_i,\tau_i)$ is the pair consisting of the mean vector and precision matrix of the $i^{\underline{th}}$ component normal density.

Assumption 1 : $q, \theta_1, \cdots, \theta_m$ are mutually independent.

Assumption 2 : q has a Dirichlet distribution with hyperparameters $\lambda_i, \cdots, \lambda_m$, all > 0. The prior density of q is

$$f_0(q) = \frac{\Gamma(\lambda_1 + \cdots + \lambda_m)}{\Gamma(\lambda_1) \cdots \Gamma(\lambda_m)} q_1^{\lambda_1 - 1} \cdots q_{m-1}^{\lambda_{m-1}-1} q_m^{\lambda_m - 1} .$$

Assumption 3 : Given τ_i, the prior distribution of μ_i is d-variate normal $N_d(\alpha_i, c_i\tau_i)$ with mean $\alpha_i \in R^d$ and precision matrix $c_i\tau_i$, where $c_i > 0$ is a hyperparameter. The prior distribution of τ_i is Wishart with $v_i > d-1$ degrees of freedom and expected value $v_i h_i^{-1}$, where h_i is a positive definite matrix. Thus the joint prior density of

$\theta_i = (\mu_i, \tau_i)$ is

$$f_i(\mu_i, \tau_i) = c_i^{d/2} |h_i|^{\frac{v_i}{2}} |\tau_i|^{\frac{v_i - d}{2}} \times$$

$$\times \exp \{- \frac{c_i}{2} (\mu_i - \alpha_i)^T \tau_i (\mu_i - \alpha_i) - \frac{1}{2} \mathrm{tr} h_i \tau_i\} .$$

The prior distributions given in Assumptions 2 and 3 are the standard conjugate priors for multinomial probabilities and the parameters of the normal-Wishart distribution of the sample mean and covariance, [1]. Their use here is for mathematical convenience, rather than because of any prior conviction as to their suitability. However, it is apparent that the large number of hyperparameters involved $(\lambda_i, v_i, c_i, \alpha_i, h_i)$ allows a great deal of flexibility in applications.

The penalized likelihood function corresponding to this prior is

$$\ell_1(\Theta) = \sum_{j=1}^{n} \log f(X_j|\Theta) + \sum_{i=1}^{m} \lambda_i \log q_i$$

$$+ \frac{1}{2} \sum_{i=1}^{m} (v_i - d) \log|\tau_i| - \frac{1}{2} \sum_{i=1}^{m} c_i (\mu_i - \alpha_i)^T \tau_i (\mu_i - \alpha_i)$$

$$- \frac{1}{2} \sum_{i=1}^{m} \mathrm{tr} h_i \tau_i .$$

Here, we have eliminated terms which depend neither on the parameters, nor on the samples and, for convenience, have also replaced λ_i in the original definition of $f_0(q)$ by $\lambda_i + 1$.

3. GLOBAL AND LOCAL MAXIMA OF $\ell_1(\theta)$

The prior density of θ given in the preceding section is unbounded, as is $\ell_1(\theta)$, unless the hyperparameters satisfy $\lambda_i \geq 0$, $v_i \geq d$. Therefore, these restrictions will be assumed for the remainder of this paper. The ordinary likelihood function can be obtained by allowing $\lambda_i = 0$, $v_i = d$, $c_i = 0$, $h_i = 0$ for each i. This corresponds to a posterior distribution derived from an improper, noninformative prior.

Choices of the hyperparameters which guarantee a global maximizer of $\ell_1(\theta)$ are given in the following theorem.

THEOREM 1. If $v_k > d$ and h_k is positive definite for each k, then $\ell_1(\theta)$ has a maximum.

PROOF: Since $\lambda_i \geq 0$,

$$\ell_1(\theta) \leq \sum_{j=1}^{n} \log \max_i f_i(x_j|\theta_i) + \frac{1}{2}\sum_{i=1}^{m}(v_i-d)\log|\tau_i|$$

$$-\frac{1}{2}\sum_{i=1}^{m} \operatorname{tr} h_i\tau_i$$

$$= \frac{1}{2}\{\sum_{j=1}^{n} \max_i [\log|\tau_i|-(x_j-\mu_i)^T\tau_i(x_j-\mu_i)]$$

$$+\sum_{i=1}^{m}[(v_i-d)\log|\tau_i|- \operatorname{tr} h_i\tau_i]\}$$

For each i, let $C_i(\theta) = \{x \in R^d | \log|\tau_i|-(x-\mu_i)^T\tau_i(x-\mu_i) \geq \log|\tau_k| - (x-\mu_k)^T\tau_k(x-\mu_k)$ for each $k\}$, let $\phi_i(\theta)$ be the number of samples in $c_i(\theta)$, and let

$$s_i(\theta) = \sum_{x_j \epsilon c_i(\theta)} (x_j-\mu_i)^T \tau_i (x_j-\mu_i).$$

Then

$$\ell_1(\theta) \le \frac{1}{2} \sum_{i=1}^{m} [A_i(\theta)\log|\tau_i| - trB_i(\theta)\tau_i]$$

where $A_i(\theta) = v_i-d+\phi_i(\theta)$ and $B_i(\theta) = h_i + s_i(\theta)$.

$$\ell_1(\theta) \le \frac{1}{2} \sum_{|\tau_i|\le 1} [(v_i-d)\log|\tau_i| - trh_i\tau_i]$$

$$+ \frac{1}{2} \sum_{|\tau_i|>1} [(v_i+d+n)\log|\tau_i| - trh_i\tau_i]$$

Let $\eta(\tau_i)$ and $\rho(\tau_i)$ denote the largest and smallest eigenvalues of τ_i respectively. If $\rho(\tau_k) \to \infty$ or $\eta(\tau_k) \to 0$ for some k, then the term corresponding to τ_k in the inequality above tends to $-\infty$ while the other terms are bounded. Therefore, there is an $r > 0$ such that

$\sup_{\theta} \ell_1(\theta) = \sup_{\theta \epsilon \Theta r} \ell_1(\theta) < \infty$, where

$$\Theta_r = \{\theta \mid \frac{1}{r} \le \eta(\tau_k) \le \rho(\tau_k) \le r \text{ for each } k\}.$$

Represent Θ_r as $Q \times \overline{\psi}_1 \times \cdots \times \overline{\psi}_m$, where $Q = \{q \epsilon R^m | q_i \ge 0$ for each i and $\sum_{i=1}^{m} q_i = 1\}$, and $\psi_i = \{(\mu_i,\tau_i) \mid \frac{1}{r} \le \eta(\tau_i), \rho(\tau_i) \le r\}$. Let $\overline{\psi}_i$ be the one point compactification of ψ_i, so that $\theta_i \epsilon \psi_i$ tends to ∞ if and only if $||\mu_i|| \to \infty$. If $\theta_i \to \infty$, then $f_i(x_j|\theta_i) \to 0$ for all j; thus, by allowing $-\infty$ as a value, $\ell_1(\theta)$ can be extended continuously to $\overline{\Theta}_r = Q \times \overline{\Psi}_1 \times \cdots \times \overline{\Psi}_m$, and has a maximum on that set, say at $\overline{\theta}$.

Suppose $\overline{\theta}$ is a point at infinity; i.e., that $\overline{\mu}_k = \infty$ for some k. Then $c_k = 0$, because otherwise $\ell_1(\overline{\theta}) = -\infty$. $\ell_1(\overline{\theta})$ is obviously not decreased by replacing $\overline{\mu}_k$ by any finite value. Therefore, $\ell_1(\theta)$ is maximized by a point in Θ_r. QED.

Unfortunately, as with other penalized likelihood functions the circumstances under which a consistent global maximizer of $\ell_1(\theta)$ exists are not known. Even if one exists there is no procedure for finding the global maximizer. Therefore, we must consider local maximizers. The necessary conditions for a local maximizer of $\ell_1(\theta)$ are, for $i = 1, \cdots, m$:

$$q_i = \frac{\sum_{j=1}^{n} \frac{q_i f_i(x_j|\theta_i)}{f(x_j|\theta)} + \lambda_i}{n + \lambda} , \tag{2.2}$$

where $\lambda = \sum_{i=1}^{m} \lambda_i$,

$$\mu_i = \frac{c_i\alpha_i + \sum_{j=1}^{n} \frac{q_i f_i(x_j|\theta_i)}{f(x_j|\theta)} x_j}{c_i + \sum_{j=1}^{n} \frac{q_i f_i(x_j|\theta_i)}{f(x_j|\theta)}} , \tag{2.3}$$

$$(2.4) \qquad \Sigma_i = \frac{h_i + c_i(\mu_i-\alpha_i)(\mu_i-\alpha_i)^T + \sum_{j=1}^{n} \frac{q_i f_i(x_j|\theta_i)}{f(x_j|\theta)}(x_j-\mu_i)(x_j-\mu_i)^T}{v_i-d + \sum_{j=1}^{n} \frac{q_i f_i(x_j|\theta_i)}{f(x_j|\theta)}}$$

These equations are the basis for an EM-like iteration procedure defined by evaluating the right hand sides with the current values of the parameters to obtain updated values of the parameters. Each of the updated parameters is a convex combination of some prior estimate and the EM update for ordinary maximum likelihood estimation. Interestingly, the updated q_i is a convex combination of the EM update and the prior mode $\frac{\lambda_i}{\lambda}$ of q_i, whereas the updated Σ_i is a convex combination of the EM update and the prior conditional mean

$$\frac{h_i+c_i(\mu_i-\alpha_i)(\mu_i-\alpha_i)^T}{v_i-d}$$

of Σ_i given μ_i, not the prior mode. Obviously, the larger the sample size, the greater will be the weight given to the EM updates and the less given to the prior estimates. When the update equation (2.4) for Σ_i is evaluated using the just updated value of μ_i in the products $(x_i-\mu_i)(x_i-\mu_i)^T$ and $(\mu_i-\alpha_i)(\mu_i-\alpha_i)^T$ this successive substitutions procedure is equivalent to the modified EM procedure suggested by Dempster, Laird, and Rubin [4] for finding posterior modes. Hereafter, we shall refer to this procedure as the generalized EM procedure (GEM). The general convergence properties of the GEM procedure follow from

[8, Theorem 4.1], more specifically, starting from any point $\theta^{(0)}$ in parameter space, the sequence $\{\theta^{(k)}\}_{k=0}^{\infty}$ produced by the GEM procedure converges to a nonempty, connected, compact subset of parameter space on which the penalized likelihood $\ell_1(\theta)$ is constant, and on which the equations (2.2)-(2.4) are satisfied.

The next theorem assures that the GEM procedure converges to a consistent local maximizer of $\ell_1(\theta)$, given a good enough starting value.

THEOREM 2. If the true parameter $\overline{\theta}$ is in the interior of the parameter set, then there is a neighborhood N of $\overline{\theta}$ such that with probability 1, if n is sufficiently large there is a unique solution $\hat{\theta}$ of (2.2)-(2.4) in N and $\hat{\theta} \to \overline{\theta}$ as $n \to \infty$. Furthermore, with probability 1, for large n the GEM procedure converges to $\hat{\theta}$ if the starting point is near enough to $\hat{\theta}$.

PROOF. The existence and uniqueness of a consistent local maximizer is a consequence of a consistency theorem due to Chanda [2], (see also Peters and Walker [6]). A simple modification of the proof of that theorem shows that the Hessian $d^2\ell_1(\theta)$ is negative definite at $\theta = \hat{\theta}$ for large n. Therefore, $\ell_1(\theta)$ is strictly concave in a neighborhood of $\hat{\theta}$. The local convergence of the GEM procedure to $\hat{\theta}$ now follows from the consistency theorem and Lemmas 1 and 2 of [7].

4. OVERMODELED MIXTURES

For mixture problems in which the number of normal components is not precisely known, the present model is not appropriate from a Bayesian point of view. However, it is possible that the penalized likelihood function exhibits better numerical and statistical properties in this situation than the ordinary likelihood function. To illustrate, suppose that the model contains m normal components, but the true density is a mixture of $k < m$ normal components. Thus,

$$f(x|\overline{\theta}_{(k)}) = \sum_{i=1}^{k} \overline{q}_i f_i(x|\overline{\theta}_i) \qquad (\overline{q}_i > 0)$$

is the true density, and

$$f(x\ |\theta_{(m)}) = \sum_{i=1}^{m} q_i f_i(x|\theta_i)$$

is the model. Let the hyperparameters for the model satisfy $\lambda_i = 0$, $v_i > d$, $c_i > 0$, $\alpha_i \in R^d$, and h_i positive definite for $i = 1, \cdots, m$. By Theorem 2, there is a consistent solution $\overline{\theta}_{(k)} = (q_1, \cdots, q_k, \hat{\theta}_1, \cdots, \hat{\theta}_k)$ of equations (2.2)-(2.4) for the k component mixture. Let $\hat{q}_i = 0$, $\hat{\mu}_i = \alpha_i$, $\hat{\Sigma}_i = h_i/(v_i - d)$ for $i = k+1, \cdots, m$, and let $\hat{\theta}_{(m)} = (\hat{q}_1, \cdot\cdot, \hat{q}_m, \hat{\theta}_1, \cdots, \hat{\theta}_m)$. Clearly $\hat{\theta}_{(m)}$ is a solution of (2.2)-(2.4) for the m component mixture which is consistent in the sense that $f(x|\hat{\theta}_{(n)}) \to f(x|\hat{\theta}_{(k)})$ as $n \to \infty$. In contrast, it is not known if there is a consistent solution of the ordinary likelihood equations in this situation.

5. REMARKS AND CONCLUSIONS

The remarks at the end of the preceding section suggest that in cases where the number m of normal components is unknown, but a reasonable upper bound can be assumed, one should take $\lambda_i = 0$, $v_i > d$, $c_i > 0$, h_i positive definite. Otherwise, the choice of the hyperparameters may be guided by prior guesses at location and dispersion of the mixture parameters. For example

$$\mathbb{E}(q_i) = \frac{\lambda_i+1}{\lambda+m}$$

$$\mathrm{cov}(q_i,q_k) = -\frac{(\lambda_i+1)(\lambda_k+1)}{(\lambda+m)^2(\lambda+2)}$$

$$\mathrm{var}(q_i) = \frac{(\lambda_i+1)(\lambda-\lambda_i+m-1)}{(\lambda+m)^2(\lambda+m+1)}$$

can be used to aid in choosing the λ_i, while the equation

$$\mathbb{E}(\Sigma_i) = c_i\ \mathrm{var}(\mu_i)$$

(provided $v_i > d+1$) can aid in choosing c_i.

The procedures outlined herein may be especially useful in applications such as crop inventories from satellite data. There, spectral measurements may be sampled from a large ground area (segment) which is itself chosen from a large number of possibilities. The normal mixture model has often been used for the distribution of spectral responses from particular segments. Thus the parameters $(q, \theta_1, \cdot\cdot, \theta_m)$ can be considered characteristic of segments, while the prior distribution of these parameters can reflect their variability among the possible choices of segments. Since there are "ground truth" segments available in which each pixel has a known class identity, it is possible that the hyperparameters of the prior distribution could be estimated from the ground truth segments.

Further research into the numerical and statistical properties of the GEM procedure is planned. The properties to be studied include the consistency of the global maximizer, the behavior of the GEM procedure for overmodeled mixtures, and the sensitivity of the procedure to starting values, for various choices of the hyperparameters.

REFERENCES

[1] Aitchison, J. and Dunsmore, I. R., Statistical Prediction Analysis, (Cambridge University Press, 1975).

[2] Chanda, K. C., A note on the consistency and maxima of the roots of the likelihood equations, Biometrika, 41 (1954), 56-61.

[3] Day, N. E., Estimating the components of a mixture of normal distributions, Biometrika, 56 (1969), 463-474.

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

[5] Hathaway, R., Contrained Maximum Likelihood Estimation for a Mixture of m Univariate Normal Distributions, Statistics Technical Report No. 92, Department of Mathematics and Statistics, University of South Carolina, March 1983.

[6] Peters, B. C. and Walker, H. F., An iterative procedure for obtaining maximum likelihood estimates of the parameters for a mixture of normal distributions, SIAM J. Appl. Math., 35 (1978), 362-378.

[7] Redner, R. A., Maximum Likelihood Estimation for Mixture Models, NASA/JSC Technical Report, Sept. 1980.

[8] Redner, R. A. and Walker, H. F., Mixture densities, maximum likelihood and the EM algorithm, SIAM Review, 26 (1984), 195-239.

SPATIAL ESTIMATION FROM REMOTELY SENSED DATA
VIA EMPIRICAL BAYES MODELS

by

Joe R. Hill, David V. Hinkley, Hubert Kostal, Carl N. Morris
Center for Statistical Sciences
Department of Mathematics
The University of Texas
Austin, Texas 78712

ABSTRACT

Multi-channel satellite image data, available as LANDSAT imagery, are recorded as a multivariate time series (four channels, multiple passovers) in two spatial dimensions. We consider here the application of parametric empirical Bayes theory to classification of, and estimating the probability of, each crop type at each of a large number of pixels. This theory involves both the probability distribution of imagery data, conditional on crop types, and the prior spatial distribution of crop types. For the latter we use Markov models indexed by estimable parameters. A broad outline of the general theory reveals several questions for further research. Some detailed results are given for the special case of two crop types when only a line transect is analyzed. There is also a detailed discussion of estimation of an underlying continuous process on the lattice, which would be applicable to such quantities as crop yield.

1. Introduction

Multi-channel satellite image data, available as LANDSAT imagery, are recorded as a multivariate time series (four channels, multiple passovers) in two spatial dimensions. We consider here the application of parametric empirical Bayes theory to estimating the probability of each crop type at each of a large number of pixels.

Parametric empirical Bayes modeling has proven effective in various spatial applications previously. For example, applications have been made in revenue sharing (small census areas), insurance (territories), false alarm estimation (neighborhoods in a city), epidemiology (cities), and forestry (areas), as reported respectively by Fay and Herriot [11] and Morris [16], Carter and Rolph [7], Efron and Morris [9], and Burk and Ek [6]. Such applications could be further improved by better use of neighborhood information, as by use of affinity matrices (Morris and Kostal [17]), based on ideas in Stein [19]. Much more substantial advances are needed for LANDSAT data.

We define "Empirical Bayes Modeling" to be the convolution of two families of distributions, one for the measurement data x given unknown parameters θ,

$$(1.1) \qquad \underset{\sim}{x}|\underset{\sim}{\theta} \sim f(\underset{\sim}{x}|\underset{\sim}{\theta}) \qquad \underset{\sim}{\theta} \varepsilon \Theta ,$$

and the second for the unknown parameters

$$(1.2) \qquad \underset{\sim}{\theta} \sim \pi_\alpha(\underset{\sim}{\theta}) , \qquad \alpha \varepsilon \mathcal{A} .$$

Expression (1.1) provides the likelihood, whilst (1.2) defines a family Π of densities or, in the "Parametric Empirical Bayes" case, a parametric family indexed by $\alpha \in a$. The problem is to make an inference about $\underset{\sim}{\theta}$ from $\underset{\sim}{x}$, knowing a. If a corresponds to all possible distributions on $\underset{\sim}{\theta}$, then standard frequentist methods are appropriate; and, if a has but one element, then a pure Bayes approach for the single known prior distribution is appropriate. Different (empirical Bayes) procedures arise if a contains many elements, but not too many. The key is that one can learn about α by considering the marginal density of the data

$$(1.3) \qquad h_{\alpha}(\underset{\sim}{x}) = \int f(\underset{\sim}{x}|\underset{\sim}{\theta})\,\pi_{\alpha}(\underset{\sim}{\theta})\,d\,\underset{\sim}{\theta}.$$

This empirical Bayes approach provides some conceptual advantages for LANDSAT data, because it suggests that we separate the satellite measurement process $\underset{\sim}{x}$ and the ground truth image process $\underset{\sim}{\theta}$. The major part of spatial correlation is in the ground truth process: neighboring pixels are likely to have the same crop type. The distribution (1.1) of the measurement process normally would involve much less correlation, although the theory does not require this.

The empirical Bayes approach accepts the fact that available satellite and ground truth data (U.S. sites) can be used to estimate the functions $f(\cdot|\theta)$. Thus, for example, extraction of "greenness" and/or "brightness" functions from the four channels is a legitimate operation from LANDSAT data, as are the appropriate time series transformations, e.g. "Badhwar numbers". However, the appropriate prior distribution $\pi_{\alpha}(\cdot)$ for the ground truth (crop types) for a target site may be very different from that π_{α_0} at a training site. Crop sizes and relative proportions may vary widely in different states and countries, and thus it would be unwise to assume $\alpha = \alpha_0$. The empirical Bayes approach then provides an appropriate method of estimation, in which (1.2) enables us to restrict drastically the probable spatial arrangements on $\underset{\sim}{\theta}$.

With these considerations in mind, in this paper we analyze situations where the density (1.1) is assumed known, and where particular families $\pi_\alpha(\cdot)$ are chosen. (For further comment, see Section 5.)

Section 2 concentrates on the relatively simple situation of discrete Markovian (correlated) $\underset{\sim}{\theta}$ corresponding to two crop types, on a transect, and takes $f(\cdot|\theta)$ to be known. We learn there that the posterior log odds on each crop type can be approximated by a "moving average"; that the correct moving average depends on $\alpha=(L_0,L_1)$, the average lengths of fields of type 0 and type 1; and that α can be estimated without access to ground truth data. The theory suggests how one might proceed in more realistic, more complicated situations. Section 4 addresses the more complicated two-dimensional situations with discrete $\underset{\sim}{\theta}$ using Gibbsian Markovian distributions, as described by Besag [4] and Geman and Geman [12]. Section 3 considers related empirical Bayes theory for continuous autoregressive ground truth parameters $\underset{\sim}{\theta}$. Section 5 contains general remarks, including an outline of the further research needed to bring our approach to the point of providing viable software for automatic processing of LANDSAT data.

2. Two-Crop Models on the Transect

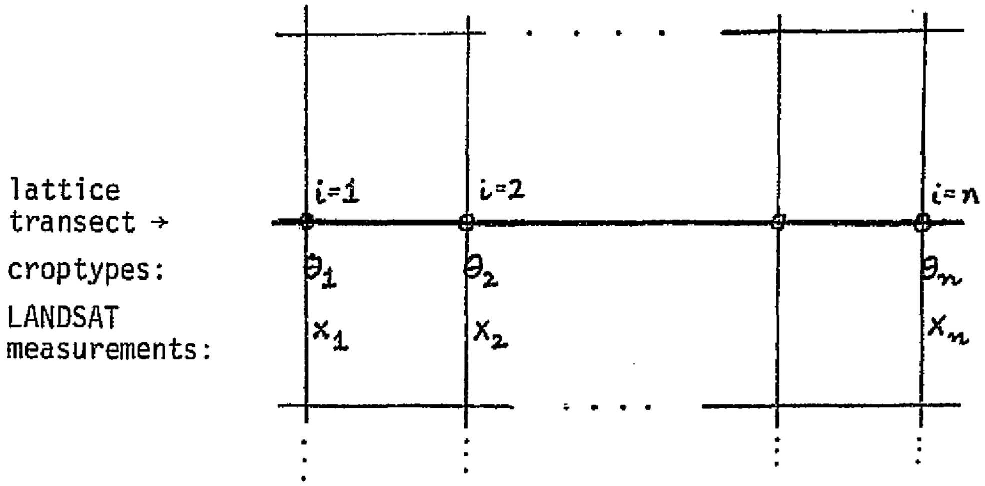
On a (lattice) transect, as illustrated in Figure 2.1, let $\underset{\sim}{\theta} = (\theta_1, \theta_2, \ldots)$ indicate the sequence of crop types, where $\theta_i = 0$ or 1. The measurement vector x_i (possibly multivariate) corresponds to a LANDSAT reading for pixel i. We assume that x_i has density $f_{\theta i}(\cdot)$, with both f_0

and f_1 known. For fixed $\underset{\sim}{\theta}$, the x_i's are assumed to be independent. Then, letting $Z_i = \log\{f_1(x_i)/f_0(x_i)\}$ be the log likelihood ratio for the $i^{\underline{th}}$ pixel, we can write the whole likelihood function as

$$(2.1) \qquad L_Z(\underset{\sim}{\theta}) = \exp(\sum_i \theta_i Z_i) ,$$

ignoring an irrelevant constant multiplier.

FIGURE 2.1 Transect model notation



A simple class Π of prior distributions on the binary crop indicator parameters is the stationary random walk (first-order binary Markov process) with transition probabilities

$$(2.2) \qquad \Pr(\theta_{i+1} = 1|\theta_i) = p_{\theta_i} \equiv 1 - q_{\theta_i} , \quad \theta_i = 0, 1.$$

This defines a two-parameter family Π indexed by $p_1 = 1 - q_1$ and $p_0 = 1 - q_0$. Equivalently, one may parameterize with $L_1 \equiv 1/p_0$ and $L_0 \equiv 1/q_1$, the expected lengths of segments for crop type 1 and crop type 0; or with $\pi_1 = L_1/(L_1 + L_0) \equiv 1 - \pi_0$ and $\nu = 2/(L_0 + L_1)$ the relative proportion of crop

1 (π_1 = marginal probability of type 1 at a given pixel) and the segment intensity parameter ($n\nu$ = expected number of sigments in a transect of length n).

The induced prior density on $\underset{\sim}{\theta}$ is proportional to the natural exponential family density

$$(2.3) \qquad \exp(\phi_1 N_1 - \phi_2 F), \text{ where}$$

$$N_1 \equiv \Sigma\,\theta_i = \#\text{ crops type }1, \quad F = \Sigma\,(\theta_i - \theta_{i+1})^2 = \#\text{ segments} - 1$$

and $\phi_1 \equiv \log(p_1/q_0)$, $\phi_2 \equiv \frac{1}{2}\log(p_1q_0/q_1p_0)$; endpoints are ignored in (2.3). Note that N_1 and F are not independent: F is restricted by the value of N_1. Also, $EN_1 = n\pi_1$ and $EF = n\nu$ if there are n pixels in the transect.

The posterior density of $\underset{\sim}{\theta}$, given both $\underset{\sim}{x}$ and the unknown $(p_0, p_1) \equiv \underset{\sim}{\alpha}$, is obtained by multiplying (2.1) and (2.3), and has logarithm (ignoring an additive constant that does not depend on $\underset{\sim}{\theta}$)

$$(2.4) \qquad \ell(\underset{\sim}{\theta}\,|\,\underset{\sim}{x}, \underset{\sim}{\alpha}) = \Sigma\,\theta_i(Z_i + \phi_1) - \phi_2\Sigma\,(\theta_i - \theta_{i+1})^2.$$

The empirical Bayes viewpoint acknowledges that ϕ_1, ϕ_2 are unknown, but that the marginal distribution of Z_i may be used to estimate them. For example, consider the case of normal, homoscedastic measurements, $x_i|\theta_i \sim N(\mu_{\theta_i}, \sigma^2)$, for which $Z_i = \delta(x_i - \bar{\mu})/\sigma$ with $\delta = (\mu_1 - \mu_0)/\sigma$ and $\bar{\mu} = (\mu_0 + \mu_1)/\sigma$. Assuming n pixels, define $\bar{Z} = \Sigma Z_i/n$ and $r_j \equiv \Sigma_i (Z_{i+j} - \bar{Z}) \cdot (Z_i - \bar{Z})/n$ to be, respectively, the mean and jth autocorrelation. With respect to the marginal distribution of $\underset{\sim}{Z}$,

$$(2.5) \qquad E\bar{Z} = \delta^2(\pi_1 - .5), \quad Er_j \doteq \delta^4\pi_0\,\pi_1(p_1 - p_0)^j, \quad j = 1, 2, \ldots .$$

One then can estimate π_1 and $p_1 - p_0$, hence $\underset{\sim}{\alpha}$, from $\bar{Z}$ and r_1 by equating to expectations, that is <u>without ground truth data</u>. The autocorrelations

$r_2, r_3 \ldots$ can be used to check the order of the Markov model. One way to develop a parametric empirical Bayes estimate, which is satisfactory for large samples, is to use the estimated $\hat{\underset{\sim}{\alpha}}$ in (2.4), and proceed in a Bayesian fashion.

There are at least two approaches to the use of (2.4). Various authors have recommended choosing $\underset{\sim}{\theta}$ to maximize (2.4), subject to $\theta_i = 0, 1$. We have taken a different approach, using (2.4) to develop a formula for

$$P(\theta_i = 1 | \text{Data}, \hat{\alpha}) \equiv P_{1i} = 1 - P_{0i} \,.$$

An approximation to the exact formula, good in the normal case if δ is small or moderate, is the logistic spatial moving average

$$(2.6) \qquad \log\left(\frac{P_{1i}/\pi_1}{P_{0i}/\pi_0}\right) = W_0 Z_i + W_1(Z_{i-1} + Z_{i+1}) + W_2(Z_{i-2} + Z_{i+2}) + \ldots \equiv \underset{\sim}{W}'\underset{\sim}{Z}$$

with

$$(2.7) \qquad W_0 = 1,\ W_j = (p_1 - p_0)^j\ ,\ j = 1, 2, \ldots .$$

The P_{1i} value determined in this way, with p_1 and p_0 estimated from (2.5), is a parametric empirical Bayes estimate for the required probability.

One possible approximation to the $\underset{\sim}{\theta}^*$ which maximizes (2.4) would be to set $\theta_i^* = 1$ if $P_{1i} > \frac{1}{2}$, otherwise zero. This is equivalent to setting $\theta_i^* = 1$ if $\underset{\sim}{W}'\underset{\sim}{Z} > \log(\pi_0/\pi_1) = \log(p_0/q_1)$. The scanty numerical work we have done thus far does not contradict the near equivalence of the two methods. If this approximation is good it would improve any complicated algorithm required to maximize (2.4); see Section 4. Note that the maximization problem (2.4) can be thought of in non-Bayesian terms as maximizing the likelihood function $\Sigma\theta_i Z_i$ subject to two constraints, with $\bar{\theta} = \pi_1$ and with ϕ_2 a penalty coefficient limiting the number of segments.

The weight $\tilde{W}_j$ may be estimated by $\tilde{W}_j = r_j\ \delta^{-4}/\tilde{\pi}_0\tilde{\pi}_1$, with $\tilde{\pi}_i$ the estimate for π_i, in the normal distribution case. This may be more robust than (2.7) if the actual Markov process for $\underset{\sim}{\theta}$ has order higher than one.

Formula (2.6) develops probabilities from the "spatially moving average",

$$(2.8) \qquad SMA_i = W_0 Z_i + \sum_{j \geq 1} W_j(Z_{i+j} + Z_{i-j}) .$$

Many proposed spatial estimation methods have been based on (2.8), with $Z_{i+j} + Z_{i-j}$ replaced by more complicated forms in two dimensions. Switzer ([20]) suggested estimating the W_i by discriminant analysis, or logistic regression, using ground truth data. This is unsatisfactory, from the empirical Bayes viewpoint, if the true crop distribution characteristic (L_0, L_1) in the site to which the formula is to be applied differs substantially from the characteristics pertaining to the available ground truth data.

The empirical Bayes viewpoint is that the weights $W_0, W_1, \ldots$ should be estimated from the marginal distribution of the data in the targeted area. Discriminant analysis will be satisfactory only if ground truth patterns are similar for training and application sites.

3. Empirical Bayes Modelling with Spatial Autoregressive Priors

Parametric empirical Bayes applications are more common, and more easily developed, in situations with continuous parameters. Such rules also provide moving average estimates of the form (2.8). Our earlier work (Morris and Kostal, [17]) studied both cases where the weights W_j are partially determined by "affinities", independent of the data, and cases where localized shrinkage factor estimators were obtained with the W_j determined separately for each pixel. Each of those approaches resulted in pixel-dependent moving averages. In this section we extend the earlier results to include parametric empirical Bayes estimators which derive from explicit spatial models.

We suppose that on a $p \times q$ lattice there is a mean process $\{\mu_{ij} : i=1, \dots, p;\ j=1, \dots, q\}$, which is not directly observed. For example, μ_{ij} might be the yield of a particular crop at the pixel labeled (i,j). Corresponding to each μ_{ij} is an observation x_{ij} obtained from LANDSAT imagery data. Our Bayesian model then consists of two parts: a distribution for the observations, namely that the x_{ij} are independent $N(\mu_{ij}, \sigma^2)$, and a structural prior distribution

$$(3.1) \qquad \mu \sim N_n(m, \tau^2 A) .$$

Here μ is a vector version of the $n = pq$ means μ_{ij}.

The posterior distribution for μ is

$$(3.2) \qquad \mu | X \sim N_n((I-B)X + Bm, (I-B)\sigma^2) ,$$

where I is the $n \times n$ identity matrix and

$$(3.3) \qquad B = \sigma^2(\sigma^2 I + \tau^2 A)^{-1} .$$

The Bayesian estimate of μ is taken to be the posterior mean. The whole posterior distribution may be used for inference or discrimination, as appropriate to the context.

The empirical Bayes approach presumes that while we accept the form of the structural distribution (3.1), the parameter values are not known a priori. The observation vector X, through its marginal distribution

$$X \sim N_n(m, \sigma^2 I + \tau^2 A) , \tag{3.4}$$

provides information on $\underset{\sim}{\alpha} = (m, \tau^2, A)$. Usually σ^2 is taken to be known, as in Section 2, or independently estimable.

Some knowledge of the structure of m and A involving a limited number of parameters is necessary for (3.4) to be useful. The mean m may be a regression surface, $m = V\beta$, or a trend surface, m_{ij} being a polynomial function of the coordinates i and j. More complex spatial models for m involve describing fields. Since our objective in this section is to describe methods for estimating the unknown variance matrix A, we shall take $m = 0$. Thus the posterior mean of (3.2) will be of the form

$$\tilde{\mu} = (I-\hat{B})X . \tag{3.5}$$

A spatial autoregressive distribution arises from the simultaneous model

$$\mu_{ij} = \rho(\mu_{i-1,j} + \mu_{i,j-1} + \mu_{i+1,j} + \mu_{i,j+1})/4 + \varepsilon_{ij} \tag{3.6}$$

with appropriate modifications on the boundary of the lattice, $\varepsilon \sim N(0, \tau^2 I)$, and $|\rho| < 1$ to ensure stationarity (Cliff and Ord, [8]). Similar models have been considered by Besag ([5]), who uses a conditional analog of (3.6), and by Ord ([18]), in non-Bayesian contexts.

Hence, with C an $n \times n$ matrix determined by (3.6), $\mu = \rho C \mu + \varepsilon$, or $\mu = (I-\rho C)^{-1} \varepsilon$. Thus in (3.1) we have

(3.7) $\quad A = (I-\rho C)^{-1}(I-\rho C')^{-1}$,

which depends only on the parameter ρ.

Since (3.4) implies that $E(XX') = \sigma^2 I + \tau^2 A$, Haff ([13]), in the context of autoregressive priors for time series, suggested estimating B in (3.2) and (3.3) by $\tilde{B} = \sigma^2 \{tXX' + (1-t)M\}^{-1}$, where $0 < t < 1$ and M is an a priori value for $\sigma^2 I + \tau^2 A$. This artifice is required because XX' is singular. We follow instead the more standard practice of obtaining maximum likelihood estimates for τ^2 and ρ (Whittle [21]).

We follow the suggestion of Herzberg (in the discussion of Barlett [3] for design matrices, and now write C in terms of Kronecker products

(3.8) $\quad C = I_p \otimes C_q + C_p \otimes I_q$,

where I_r is the $r \times r$ identity matrix and C_r depends on how (3.5) is modified at boundary points. If we let $\mu_{ij} = 0$ in (3.5) when either $i \notin \{1, \ldots, p\}$ or $j \notin \{1, \ldots, p\}$, then C_r will be the $r \times r$ matrix with ¼ appearing on the first upper and lower off-diagonals,

$$C_r = \begin{bmatrix} 0 & \frac{1}{4} & 0 & 0 & \cdots & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} & 0 & & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & & 0 & 0 \\ \cdot & & & & & & \\ \cdot & & & & & & \\ \cdot & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & \frac{1}{4} & 0 \end{bmatrix}$$

Then $C_r = P_r \Lambda_r P_r'$, where P_r is orthogonal and Λ_r is a diagonal matrix, so that (3.8) may be written as

$$C = (P_p \otimes P_q)(I_p \otimes \Lambda_q + \Lambda_p \otimes I_q)(P_p \otimes P_q)' = P\Lambda P' .$$

Now it is easy to show that $\sigma^2 I + \tau^2 A = P\Gamma P'$ where $\Gamma = \sigma^2 I + \tau^2 (I-\rho\Lambda)^{-2}$. Hence $|\sigma^2 I + \tau^2 A| = \Pi\gamma_{ij} = \Pi\{\sigma^2 + \tau^2(1-\rho\lambda_{ij})^{-2}\}$.

The log likelihood arising from (3.4) is then

$$\ell(\tau^2, \rho; Y) = -\tfrac{1}{2}(\Sigma \log\gamma_{ij} + \Sigma Z_{ij}^2/\gamma_{ij}) ,$$

where $Y = P'X$. The partial derivatives of ℓ with respect to τ^2 and ρ, when equated to zero, give the likelihood equations

$$\Sigma(1-\hat\rho\lambda_{ij})^{-2}\ (\hat\gamma_{ij} - \gamma_{ij}^2)\hat\gamma_{ij}^{-2} = 0 ,$$

(3.9)

$$\Sigma\hat\tau^2\ \lambda_{ij}(1-\hat\rho\lambda_{ij})^{-3}\ (\hat\gamma_{ij} - \gamma_{ij}^2)\hat\gamma_{ij}^{-2} = 0 ,$$

where $\hat\gamma_{ij} = \sigma^2 + \hat\tau^2(1-\hat\rho\lambda_{ij})^{-2}$.

Note that P_r, Λ_r, and so Z and Λ, can be easily found (Anderson [1]). The numerical solution of (3.9) for $\hat\tau^2$ and $\hat\rho$ is obtained using an iterative procedure such as Newton-Raphson or the method of scoring. Reasonable initial values are

(3.10) $\hat\rho_0 = 0$ and $\hat\tau_0^2 = X'X/pq - \sigma^2$.

The computed values of $\hat\tau^2$ and $\hat\rho$ are then used to find $\hat A$ and $\hat B$ in (3.7) and (3.3), leading to the estimate (3.5) for μ.

These results are a special case of results in Kostal ([15]) which include other autoregressive structures, spatial moving averages, space-time processes, and non-zero m.

4. Markov Prior Models and Their Use

The line transect analysis in Section 2 involves a special case of a general class of Markov prior models for crop codes $\underset{\sim}{\theta}$. In this section we outline the general class, and discuss its use in construction of Bayes and empirical Bayes estimates for $\underset{\sim}{\beta}$. A similar development would be possible in the somewhat different context of Section 3, providing an alternative to models such as (3.6).

We begin by noting that the first-order binary Markov model on the transect, as in (2.2) or (2.3), can be written in two other, equivalent forms. Again let $\underset{\sim}{\theta} = (\theta_1, \ldots, \theta_n)$ and write $\underset{\sim}{\theta}^{(i)} = \{\theta_j : j \neq i\}$. Then, for some normalizing constant K_n depending on $\underset{\sim}{\alpha} = (\beta, \gamma)$,

$$(4.1) \qquad \Pr(\underset{\sim}{\theta}) = K_n(\underset{\sim}{\alpha}) \exp(-\beta\Sigma\theta_i - \gamma\Sigma\theta_i\theta_{i+1})$$

and

$$(4.2) \qquad \Pr(\theta_i|\underset{\sim}{\theta}^{(i)}) = \Pr(\theta_i|\theta_{i-1}, \theta_{i+1}) \propto \exp\{-\beta\theta_i - \gamma(\theta_{i-1}\theta_i + \theta_1\theta_{i+1})\}.$$

where, in the notation of Section 2, $\beta = -\log_e(p_0q_1/q_0^2)$ and $\gamma = -2\phi_2$; simple boundary modifications of (4.2) are needed at $i = 1, n$. The formulae (4.1) and (4.2) correspond respectively to a Gibbs distribution and its local, conditional representation, both of which are capable of wide generalization to accomodate the more complex situations with which we wish to work.

The general structure illustrated by (4.1) can be described as follows (Besag, [4] and Isham, [14]). Suppose that θ_i depends, stochastically, on values of θ_j at the neighboring pixels, $j \in N(i)$, in the sense that $\Pr(\theta_i|\underset{\sim}{\theta}^{(i)}) = \Pr(\theta_i|\theta_j, j \in N(i))$; for example, we see in (4.2) that $N(i) = \{i-1,i+1\}$ except at the boundaries, where $N(1) = \{2\}$ and $N(n) = \{n-1\}$. Next, define a set C of pixels to be a clique if C contains a single pixel or if all pairs of pixels in C are neighbors of each other in the system of neighborhoods $N(\cdot)$; for example, with (4.2) the list of sets C is $\{1\}, \{2\}, \ldots, \{n\}, \{1,2\}, \{2,3\}, \ldots, \{n-1,n\}$. Then the general form of Markov model for $\underset{\sim}{\theta}$ is

$$(4.3) \qquad \Pr(\underset{\sim}{\theta}) = K_n(\underset{\sim}{\alpha}) \exp\{-\sum_C d_C(\underset{\sim}{\theta}_C, \underset{\sim}{\alpha})\},$$

wherein $d_C(\cdot)$ depends on $\underset{\sim}{\theta}$ only through $\underset{\sim}{\theta}_C = \{\theta_j : j \in C\}$; for example, in (4.1) $d_{\{i\}}(\theta_i, \underset{\sim}{\alpha}) = \beta\theta_i$ and $d_{\{i,i+1\}}(\theta_i, \theta_{i+1}, \underset{\sim}{\alpha}) = \gamma\theta_i\theta_{i+1}$.

Corresponding to (4.3) is the local, conditional model

$$(4.4) \qquad \Pr(\theta_i|\theta^{(i)}) = \Pr(\theta_i|\theta_j, j \in N(i)) = \frac{\exp\left\{-\sum_{C:i\in C} d_C(\theta_i, \underset{\sim}{\theta}_C^{(i)}, \alpha)\right\}}{\sum_s \exp\left\{-\sum_{C:i\in C} d_C(s, \underset{\sim}{\theta}_C^{(i)}, \alpha)\right\}}$$

where $\underset{\sim}{\theta}_C^{(i)} = \{\theta_j : j \in C, j \neq i\}$; equation (4.2) is an example.

Both (4.3) and (4.4) can be used for discrete or continuous values of θ_i, so long as the normalizing constant $K_n(\underset{\sim}{\alpha})$ exists. Also, the pixel label i can be two-dimensional, so that general lattice models are included. Since the function $d_C(\cdot)$ can depend on C, it is also possible to define nonhomogeneous models for $\underset{\sim}{\theta}$ within this Markov framework.

By way of illustration, we outline the two-dimensional analog of our earlier binary transect model. Ignoring boundary effects, suppose that the neighborhood $N(i)$ of pixel $i = (j,k)$ is the set of four closest pixels, labelled $(j-1,k), (j+1,k), (j,k-1), (j,k+1)$. Then the set of cliques, C , consists of all single pixels (j,k) , all pairs $\{(j,k) , (j,k+1)\}$. Now take the following simple forms for $d_C(\cdot)$:

$\underline{C}$	$d_C(\underset{\sim}{\theta}_C, \underset{\sim}{\alpha})$
$\{(j,k)\}$	$\beta\theta_{j,k}$
$\{(j,k) , (j+1,k)\}$	$\gamma_1\theta_{j,k}\theta_{j+1,k}$
$\{(j,k) , (j,k+1)\}$	$\gamma_2\theta_{j,k}\theta_{j,k+1}$

The conditional representation (4.14) then gives the autologistic form

$$(4.5) \qquad \log_e \frac{\Pr(\theta_{j,k}=1|\underset{\sim}{\theta}^{(i)})}{\Pr(\theta_{j,k}=0|\underset{\sim}{\theta}^{(i)})} = -\beta - \gamma_1(\theta_{j-1,k} + \theta_{j+1,k}) - \gamma_2(\theta_{j,k-1} + \theta_{j,k+1}) ,$$

which is isotropic if $\gamma_1 = \gamma_2$.

Recall that these Markov prior models for $\underset{\sim}{\theta}$ are intended to model the spatial correlation that exists among crop codes, this correlation being a stochastic way of describing the phenomenon of fields, within which crop codes are identical. To illustrate the kind of patterns that can be generated by the models, we simulated the autologistic model (4.5) on a 10×10 lattice (with fixed boundary) with $\beta = 2$, $\gamma_1 = \gamma_2 = -1$ and obtained the pattern in Figure 4.1.

FIGURE 4.1 Simulated map of binary crop codes using model (4.5) with $\beta = 2$, $\gamma_1 = \gamma_2 = -1$.

0	0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	0	0	1	0	1
0	0	0	1	0	1	0	0	0	0	1
0	0	0	0	0	0	1	1	0	1	1
0	1	0	1	1	1	1	1	0	1	1
0	0	0	1	1	1	1	1	0	1	1
0	1	1	0	1	1	1	0	1	0	1
0	0	0	1	0	0	1	0	0	1	1
0	0	1	1	0	0	0	1	0	1	1
0	0	0	0	0	1	1	1	0	1	1
0	0	0	0	0	1	1	1	1	1	1

To return to the general development, our aim is to use models such as (4.3) as priors in conjunction with the sampling model for satellite data $\underset{\sim}{X} = (X_1, \ldots, X_n)$, so as to obtain Bayes or empirical Bayes inferences about crop codes $\underset{\sim}{\theta}$. We assume the x_j to be conditionally independent, with joint density $\Pi f(x_j|\theta_j)$. Then, by Bayes's Theorem,

$$(4.6) \qquad \Pr(\underset{\sim}{\theta}|\underset{\sim}{X} = \underset{\sim}{x}) = K_n(\underset{\sim}{x},\underset{\sim}{\alpha})\cdot\exp\{-\sum_C d_C(\underset{\sim}{\theta}_C,\underset{\sim}{\alpha}) + \sum_i \log f(x_i|\theta_i)\}$$

$$= K_n(\underset{\sim}{x},\underset{\sim}{\alpha})\cdot\exp\{-\sum_C d_C^*(\underset{\sim}{x},\underset{\sim}{\theta},\underset{\sim}{\alpha})\},$$

say, where $d_C^*(\underset{\sim}{x},\underset{\sim}{\theta},\underset{\sim}{\alpha}) = d_C(\underset{\sim}{\theta}_C,\underset{\sim}{\alpha})$ except when C is a singleton $C = \{i\}$, in which case

$$d^{*}_{\{i\}}(\underset{\sim}{x},\underset{\sim}{\theta},\underset{\sim}{\alpha}) = d_{\{i\}}(\theta_i,\underset{\sim}{\alpha}) - \log f(x_i|\theta_i) .$$

The normalizing constant $K_n(\underset{\sim}{x},\underset{\sim}{\alpha})$ in (4.6) is the reciprocal of the marginal likelihood of $\underset{\sim}{\alpha}$, from which we would, in principle, estimate $\underset{\sim}{\alpha}$ using data $\underset{\sim}{x}$.

Equation (4.6) reduces to (2.4) for the binary transect model of Section 2, and empirical Bayes inference proceeds as described there. In general, however, it is difficult to use (4.6) even to obtain the posterior mode for $\underset{\sim}{\theta}$ when the prior parameters $\underset{\sim}{\alpha}$ are specified. Geman and Geman ([12]) present an ingenious iterative "relaxation-annealing" algorithm for obtaining this posterior mode, taking advantage of the fact that the mode is invariant under scale change of the exponential family (4.6). The algorithm does not yield marginal posterior probabilities $\Pr(\theta_i|\underset{\sim}{x},\underset{\sim}{\alpha})$, for which the direct approach of Section 2 led to logistic spatial moving averages in the binary transect case.

At the present stage of development, several important questions remain to be answered before the promising Markov prior approach yields a clean empirical Bayes algorithm for image reconstruction from LANDSAT data. In the next section we discuss these questions briefly and indicate where further research is going.

5. Discussion and Outline of Further Research

We have emphasized the importance of modeling the underlying structure of fields, where crop types occur in pixel clusters. In order that new geographical areas be amenable to adaptive statistical analysis, empirical Bayes methods are needed. Use of the class of Markov priors for the crop-type "map" $\underset{\sim}{\theta}$ leads us to the following questions: (i) How can we determine the order of the Markov prior model?, (ii) Can one usefully add an "edge" distribution, as described by Geman and Geman ([12]) ? (iii) What methods or algorithms are possible for estimating the parameters $\underset{\sim}{\alpha}$ which determine distribution (4.3) ?, (iv) Can we, and need we, estimate $\Pr(\theta_i | \underset{\sim}{x})$ in addition to calculating the posterior modal estimates of θ_i ?

To take these in reverse order, we note first that the advantage of having $\Pr(\theta_i | \underset{\sim}{x})$ is that aggregate characteristics, such as proportion of area covered by a particular crop type, could be more efficiently estimated. We conjecture that generalizations of (2.6) will prove useful. On question (iii), one possibility is to use some type of EM algorithm (with $\underset{\sim}{\theta}$ as the missing value vector) embedded in the relaxation-annealing algorithm referred to in Section 4. There are three points to bear in mind here: (a) the iterative algorithm is slow, especially in the context of the simple model of Section 2, (b) likelihood estimation of $\underset{\sim}{\alpha}$ from $\underset{\sim}{\theta}$ is non-trivial (see Besag [4]). (c) marginal correlation properties of x, as used in Section 2, are made difficult on the lattice by the poor understanding of correlation properties for $\underset{\sim}{\theta}$ (Bartlett [2]) .

On question (i), standard contingency table methods would be applicable if results for ground-truth data could be extrapolated. Large-scale ground-truth data are available for the U.S. and will be analyzed.

References

[1] Anderson, T.W., The Statistical Analysis of Time Series (Wiley, New York, 1971).

[2] Bartlett, M.S., The Statistical Analysis of Spatial Pattern (Chapman & Hall, London, 1975).

[3] Bartlett, M.S., Nearest neighbor models in the analysis of field experiments, JRSS B 40 (1978) 147-174.

[4] Besag, J., Spatial interaction and the statistical analysis of lattice systems, JRSS B 36 (1974) 192-236.

[5] Besag, J., Errors-in-variables estimation for Gaussian Lattice schemes, JRSS B 39 (1977) 73-78.

[6] Burk, T.E. and Ek, A.R., Application of James-Stein and empirical Bayes procedures to simultaneous estimation problems in forest inventory, Forest Sci. 28/4 (1982) 753-771.

[7] Carter, G. and Rolph J., Empirical Bayes methods applied to estimating fire alarm probabilities, JASA 69 (1974) 880-885.

[8] Cliff, A.D. and Ord, J.K., Spatial Processes (Pion Limited, London).

[9] Efron, B. and Morris, C., Data analysis using Stein's estimator and its generalizations, JASA 70 (1975) 311-319.

[10] Efron, B. and Morris, C., Stein's estimation rule and its competitors--an empirical Bayes approach, JASA 68 (1973) 117-130.

[11] Fay, R.E., III and Herriot, R.A., Estimates of income for small places: an application of James-Stein procedures to census data, JASA 74 (1979) 269-277.

[12] Geman, S. and Geman, D., Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images, Unpublished Report, Dept. of Math., Brown Univ. (1983).

[13] Haff, L.R., Minimax estimators of the multinormal mean: autoregressive priors, J. Multi. Anal. 6 (1976) 265-280.

[14] Isham, V., An introduction to spatial point processes and Markov random fields, Rev. I.S.I. 49 (1981) 21-44.

[15] Kostal, H., Empirical Bayes Methods for Spatial Data, Ph.D. Thesis, Dept. of Math., Univ. of Texas at Austin (to appear -- 1984).

[16] Morris, C.N., Parametric empirical Bayes inference: theory and applications, JASA 78 (1983) 47-65.

[17] Morris, C.N., and Kostal, H., An empirical Bayes approach to spatial analysis, in Guseman, L.F., Jr. (ed.), NASA Symp. on Math. Pattern Recognition and Image Analysis (College Station, Texas, 1983).

[18] Ord, J.K., Estimation methods for models of spatial interaction, JASA 70 (1975) 120-126.

[19] Stein, C., Estimate of the mean of a multivariate normal distribution, Ann. of Stat. 9 (1981) 1135-1151.

[20] Switzer, P., Extensions of linear discriminant analysis for spatial classification of remotely sensed satellite imagery, J. Int. Assoc. Math. Geol. 12 (1980) 367-376.

[21] Whittle, P., On stationary processes in the plane, Biometrika 41 (1954) 434-449.

MULTIVARIATE SPLINE METHODS IN SURFACE FITTING

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 bivariate density functions where the density function is described by a histogram of measurements. The resulting approximations are then incorporated into a Bayesian classification procedure for which the Bayes decision regions and the probability of misclassification can be readily computed. Some preliminary numerical results are presented to illustrate the method.

§1. Introduction.

This paper is a continuation of our earlier work [13] on the use of spline functions as a tool in statistical pattern classification. Whereas in [13] we dealt only with univariate problems, our aim here is to develop methods which are suitable for 2 or more dimensions.

The main mathematical tool to be used here is the tensor-product splines (see Section 3 below). In particular, we show how splines can be used to estimate multivariate conditional density functions for the classes of interest. Using tensor-product B-splines, we then develop efficient algorithms for finding the associated classification regions. Moreover, we also show how to compute the probability of misclassification associated with the classification method.

The paper is divided into 8 sections. In Section 2 we discuss the general Bayes classification procedure. In Section 3 we introduce the tensor-product splines and discuss their use in the general problem of density estimation. In Section 4 we present a specific method for estimating densities based on biquadratic splines. The problems of computing the related classification regions and the probability of misclassification are treated in Sections 5 and 6, respectively. We close the paper with examples and remarks.

§2. The Bayes Classification Procedure.

Suppose that some group Π of objects can be divided into NC classes which we will denote by $\Pi_1, \Pi_2, \ldots, \Pi_{NC}$. Now suppose that we are trying to decide which class a given randomly selected object belongs to on the basis of d measurements which have been taken on the object. In particular, suppose X is a mapping from $\Pi = \Pi_1 \cup \cdots \cup \Pi_{NC}$ into R^d such that if $w \in \Pi$, then $X(w) = (x_1,\ldots,x_d)$ is the measurement taken on w. Finally, suppose that for each $i = 1,\ldots,NC$, we know the apriori probability α_i that an object will fall in class Π_i and that we also know the conditional density function P_i associated with measurements taken from the i-th class.

Given this stochastic framework, the Bayes optimal classifier is defined as follows:

> Assign an element w to the ith class Π_i if and only if its measurement X(w) belongs to the set R_i,

where $R_1,\ldots,R_{NC}$ are the Bayes decision regions defined by

$$(2.1) \qquad R_i = \{x \in R^d : \alpha_i P_i(x) > \alpha_j P_j(x) \quad \text{for all } j \neq i\} .$$

The numerical problem of identifying the Bayes decision regions is equivalent to finding the boundaries of the sets R_i. These in turn are defined by the equations $\alpha_i P_i(x) - \alpha_j P_j(x) = 0$ for $i, j = 1,\ldots,NC$.

There are several well-known ways of measuring the quality of the Bayes classification scheme described above. One convenient way is to compute the probability of misclassification (pmc) (cf. [1,2]) defined by

$$(2.2)\quad G = 1 - \int_{R^d} \max_i[\alpha_i P_i(x)]dx = 1 - \sum_{i=1}^{NC} \alpha_i \int_{R_i} P_i(x)dx \ .$$

In general, the evaluation of the pmc G is a difficult problem since it involves integration over irregularly-shaped regions in d-space.

To apply the Bayes classification procedure in a practical setting, the following steps need to be carried out:

1) estimate NC = number of classes,

2) estimate the a priori probabilities $\alpha_1,\ldots,\alpha_{NC}$,

3) estimate the density functions $P_1,\ldots,P_{NC}$,

4) estimate the decision regions $R_1,\ldots,R_{NC}$,

5) estimate the value G of the pmc.

In this paper we shall concentrate on steps 3) - 5). For some methods dealing with problems 1) and 2), see [15,24].

§3. Estimating Densities Using Splines.

In this section we discuss the problem of estimating a multivariate probability density function on the basis of a finite number of measurements taken on the underlying random variable. This is a standard statistical problem, and there are many parametric and non-parametric methods available (see e.g. [4,6,7,11,15,21-24]. Since in many applications the densities of interest are not standard parametric densities, we focus on non-parametric methods, and in particular on methods which are based on tensor-product splines.

There are several compelling reasons for selecting splines to approximate density functions. These include among others (cf. [8,19]): 1) splines are easy to store and manipulate in a digital computer, 2) there are stable efficient algorithms for evaluating splines as well as their derivatives and integrals, 3) splines are smooth but at the same time flexible, and 4) splines are capable of approximating smooth functions to high orders of accuracy.

Our method for constructing a spline s approximating an unknown density function P on the basis of a finite number of measurements will proceed in two steps:

1) use the data to construct a histogram,

2) approximate the histogram by a spline.

The problem of histograming data has been the subject of considerable study by statisticians. Hence, throughout the remainder of this section we concentrate on step 2, assuming that the histogram has been constructed.

Before we can proceed any further, we need to introduce some notation. In order to avoid undue notational complications, we shall now restrict our discussion to the case of two dimensions; i.e., $d = 2$. For the extension to $d > 2$, see Remark 3 .

We begin by introducing some parameters to describe the histogram which we wish to fit. Let nbx and nby be positive integers denoting the number of bins in the x- and y-directions, respectively. Suppose that $xt_1 < \dots < xt_{nbx+1}$ and $yt_1 < \dots < yt_{nby+1}$. Let h_{ij} be nonegative real numbers. Then the function

$$(3.1)\quad h(x,y) = \begin{cases} h_{ij} \; , & \text{if } xt_i \leq x < xt_{i+1} \text{ and } yt_j \leq y < yt_{j+1} \\ & \text{with } 1 \leq i \leq nbx \text{ and } 1 \leq j \leq nby \\ 0 \; , & \text{otherwise} \end{cases}$$

represents the histogram we want to fit.

We now introduce B-splines. We begin with univariate B-splines. Suppose that $yx_1 < \ldots < yx_{nx+mx}$ is a set of real numbers, where nx and mx are positive integers. Then associated with these points, there exists a set of mx-th order normalized B-splines $N_1^{mx}(x),\ldots,N_{nx}^{mx}(x)$ with the properties (cf. [8,19]):

$N_i^{mx}(x)$ is a piecewise polynomial of order mx with join points (knots) located at the points $yx_i,\ldots,yx_{i+mx}$;

$N_i^{mx}(x)$ has mx-2 continuous derivatives on R;

$N_i^{mx}(x)$ is positive on (yx_i, yx_{i+mx}) and vanishes elsewhere;

$N_i^{mx}(x)$ can be computed efficiently and accurately.

Suppose my and ny are also positive integers, and that $N_1^{my}(y),\ldots,N_{ny}^{my}(y)$ are similar B-splines associated with a knot sequence $yy_1 < \ldots < yy_{ny+my}$. We now define tensor-product B-splines as follows:

$$(2.4)\quad N_{ij}^{mx,my}(x,y) = N_i^{mx}(x)N_j^{my}(y) \quad , \quad i = 1,\ldots,nx \text{ and } j = 1,\ldots,ny.$$

The knot sequences $yx_1,\dots,yx_{nx+mx}$ and $yy_{ny},\dots,yy_{ny+my}$ divide the rectangle $\Omega = [yx_1\ ,yx_{nx+mx}] \times [yy_1\ ,yy_{ny+my}]$ into subrectangles $\Omega_{ij} = [yx_i\,,yx_{i+1}) \times [yy_j\,,yy_{j+1})$, $i = 1,\dots,nx+mx-1$ and $j = 1,\dots,ny+my-1$. It follows from the properties of the univariate B-splines that the tensor-product B-splines have the properties:

$N_{ij}^{mx,my}(x,y)$ is a polynomial of order mx in x and order my in y on each subrectangle $\Omega_{\nu\mu}$, $\nu = 1,\dots,nx+mx-1$, $\mu = 1,\dots,ny+my-1$;

$N_{ij}^{mx,my} \in C^{mx-2,my-2}(R^2)$;

$N_{ij}^{mx,my}(x,y) \geq 0$ for all (x,y);

$N_{ij}^{mx,my}\ (x,y) > 0$ for $yx_i < x < yx_{i+mx}$ and $yy_j < y < yy_{j+my}$.

Figure 1 shows a view of a biquadratic B-spline (mx = my = 3).

Our aim is to approximate histograms (and thus the underlying densities) by a linear combination of the tensor-product B-splines of the following form

$$(3.2) \qquad s(x,y) = \sum_{i=1}^{nx} \sum_{j=1}^{ny} c_{ij}\, N_{ij}^{mx,my}(x,y)\ .$$

To construct an approximation as in (3.2), we must select the orders mx,my, the knot sequences yx and yy, and finally the coefficients c_{ij}. In the following section we discuss how to compute these coefficients.

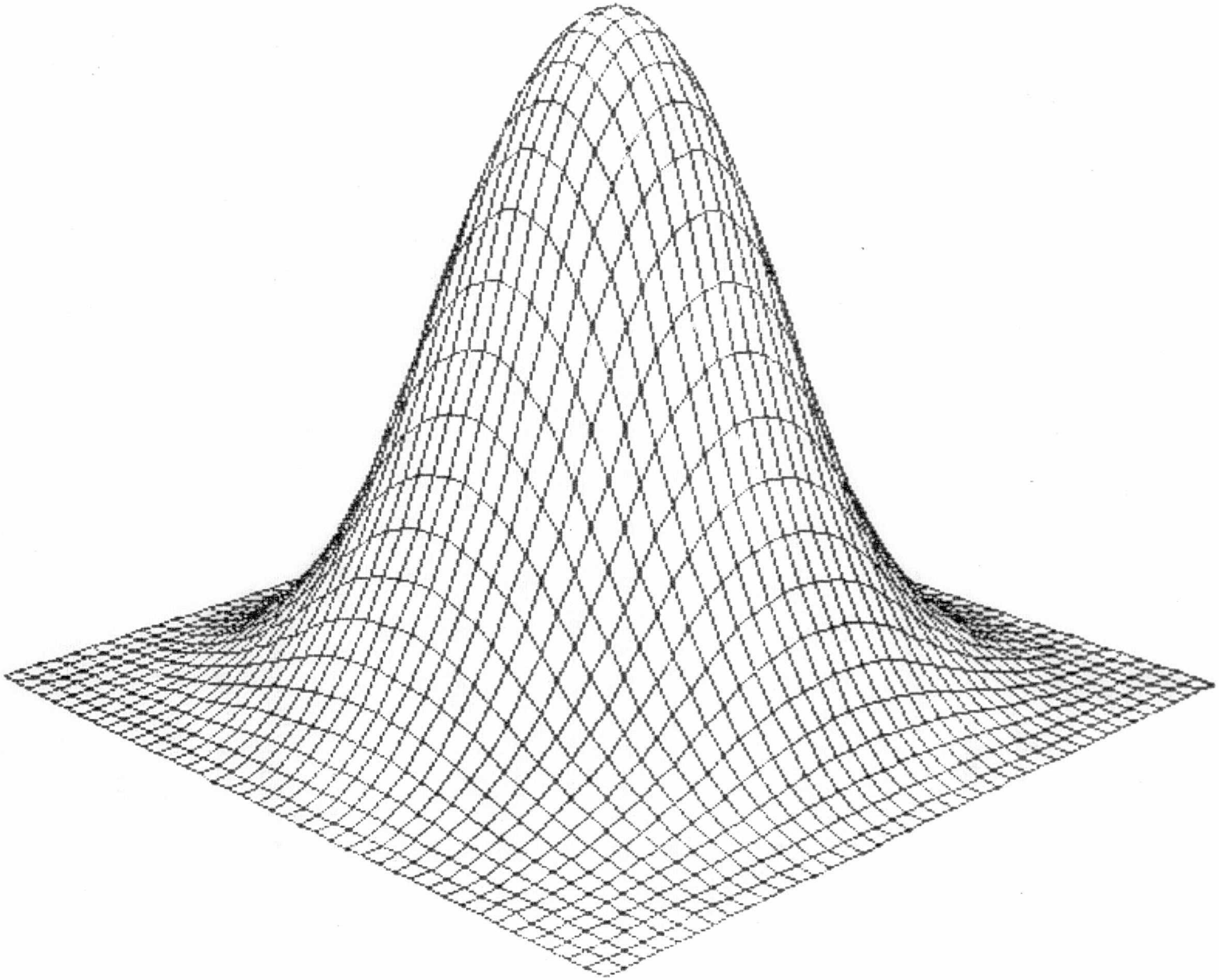


Fig. 1. A biquadratic tensor-product spline.

§4. A Biquadratic Spline Density Estimator.

In this section we discuss a specific density estimation algorithm based on biquadratic splines. First we list the input and output.

Input:

nbx = number of bins in the x-direction

nby = number of bins in the y-direction

$xt_1 < \dots < xt_{nbx+1}$ the bin edges in the x-direction

$yt_1 < \dots < yt_{nby+1}$, the bin edges in the y-direction

h_{ij}, $i = 1,\dots,nbx$ and $j = 1,\dots,nby$, the histogram values.

Output:

mx = order of the spline in the x-direction

my = order of the spline in the y-direction

nx = number of B-splines used in the x-direction

ny = number of B-splines used in the y-direction

$yx_1 < \dots < yx_{nx+mx}$, the knots in the x-direction

$yy_1 < \dots < yy_{ny+my}$, the knots in the y-direction

c_{ij}, $i = 1,\dots,nx$ and $j = 1,\dots,ny$, the coefficients of the spline.

Algorithm 4.1. (Biquadratic spline density estimation).

1. Set $mx = my = 3$.

2. Set $nx = nbx$ and $ny = nby$.

3. Set

$$yx_1 = xt_1 - (xt_2 - xt_1)$$

$$yx_i = xt_{i-1},\ i = 2,..,nbx+2$$

$$yx_{nbx+3} = xt_{nbx+1} + (xt_{nbx+1} - xt_{nbx}).$$

4. Set

$$yy_1 = yt_1 - (yt_2 - yt_1),$$

$$yy_i = yt_{i-1} , \; i = 2,\dots,nby+2,$$

$$yy_{nby+3} = yt_{nby+1} + (yt_{nby+1} - yt_{nby}).$$

5. Compute the c_{ij}'s which solve the linear system of equations

$$\sum_{i=1}^{nx} \sum_{j=1}^{ny} c_{ij} \int_{xt_\nu}^{xt_{\nu+1}} \int_{yt_\mu}^{yt_{\mu+1}} N_i^3(x)\, N_j^3(y)\, dxdy$$

$$= \quad h_{\nu\mu}(xt_{\nu+1} - xt_\nu)(yt_{\mu+1} - yt_\mu) \quad ,$$

for $\nu = 1,\dots,nx$ and $\mu = 1,\dots,ny$.

Discussion: This algorithm produces a biquadratic spline which belongs to $C^{1,1}(R^2)$ and whose support is on the rectangle $[yx_1, yx_{nbx+3}] \times [yy_1, yy_{nby+3}]$. This rectangle is slightly larger than the support of the histogram which is the rectangle $H = [xt_1, xt_{nbx+1}] \times [yt_1, yt_{nby+1}]$.

The system of equations in step 5 can be arranged in matrix form as follows:

$$(4.1) \qquad \Phi_x \, C \, \Phi_y^T = Z ,$$

where $C = (c_{ij})_{i=1,j=1}^{nx,\ ny}$, $Z = (z_{ij})_{i=1,j=1}^{nx,\ ny}$, and

$$(4.2) \qquad z_{ij} = h_{ij}(xt_{i+1} - xt_i)(yt_{j+1} - yt_j) \quad , \quad \begin{array}{l} i = 1,\dots,nx \\ j = 1,\dots,ny \; . \end{array}$$

Here Φ_x is the nx by nx matrix with entries

$$(4.3) \qquad (\Phi_x)_{ik} = \int_{xt_i}^{xt_{i+1}} N_k^3(x)\, dx \quad , \quad i,k = 1,\dots,nx,$$

and Φ_y is the analogous ny by ny matrix. The entries of these matrices can be generated by standard B-spline methods (cf. [19,Corollary 5.18]). It is not hard to show (see Lemma 4.2 below) that the matrices Φ_x and Φ_y are tridiagonal, nonsingular, and totally positive.

Following deBoor [9], we may solve the system (4.1) in two steps:

1) Solve Φ_x U = Z for the nx by ny matrix U,

2) Solve $\Phi_y C^T = U^T$ for the nx by ny matric C.

In carrying out step 1) it is desirable to first compute the LU decomposition of Φ_x, and then to compute the columns of U by back-substitution using one column of Z at a time. Because of the bandedness of Φ_x, the decomposition can be done with special software designed for banded matrices. Because of the total positivity, the decomposition can be done without pivoting (cf, [8]). For appropriate linear algebra packages, see the FORTRAN subroutines BANDET and BANSLV in [8]. The same comments apply to step 2).

The technique suggested above is more than simply a convenient way to arrange the solution of the system of equations in step 5) of the algorithm -- the approach takes essential account of the tensor-product nature of the problem and results in major savings in storage and operation counts. In particular, in equation (4.1) we need only store Z and the three diagonals of the matrices Φ_x and Φ_y. The total operation count for the algorithm is $O(nx^3 + ny^3)$.

The equations in step 5) of this algorithm are precisely the conditions that the volume under the spline surface in bin $\nu\mu$ should be exactly equal to the volume of the histogram in bin $\nu\mu$, all $\nu = 1,\ldots,nbx$ and $\mu = 1,\ldots,nby$.

Lemma 4.2. The nx by nx matrix Φ_x defined in (4.3) is tridiagonal, nonsingular, and totally positive.

Proof: The tridiagonal nature of Φ_x is obvious from the support properties of the B-splines. In particular, because of the choice of knots, the only B-splines with nonzero values in the interval $[xt_i, xt_{i+1}]$ for $1 < i < nx$ are the B-splines N^3_{i-1}, N^3_i, and N^3_{i+1}. (On the first and last intervals only two B-splines have nonzero values).

We turn now to the assertion of total positivity. Suppose that we select $1 \leq \nu_1 < \ldots < \nu_k \leq nx$ and $1 \leq \mu_1 < \ldots < \mu_k \leq nx$ with $1 \leq k \leq nx$. We need to show that the determinant

$$D\binom{\nu_1,\ldots,\nu_k}{\mu_1,\ldots,\mu_k} = \det\left(\int_{xt_{\nu_i}}^{xt_{\nu_i+1}} N^3_{\mu_j}(x)dx \right)^{k,\ \ k}_{i=1,j=1}$$

$$= \int_{xt_{\nu_1}}^{xt_{\nu_1+1}} \cdots \int_{xt_{\nu_k}}^{xt_{\nu_k+1}} [\det(N^3_{\mu_j}(\xi_i))^{k,\ \ k}_{i=1,j=1}]d\xi_1\ldots d\xi_k$$

is nonnegative. But by the total positivity of the B-splines (see e.g. [19,Theorem 4.65]), the determinant in the integrand of this multiple integral is nonnegative for all $\xi_1 < \ldots < \xi_k$. Since the intervals over which the integration is performed are in increasing order, the total postivity assertion follows.

Finally, the nonsingularity of Φ_X follows from the fact (cf. [19,Theorem 4.65]) that the determinant in the integrand of $D\binom{1,\ldots,nx}{1,\ldots,ny}$ is a continuous function which is strictly positive for certain values of the ξ's in the intervals over which the integration takes place. □

§5. Computing The Bayes Decision Regions.

In this section we discuss the problem of computing approximations to the Bayes decision regions $R_1,\ldots,R_{NC}$ in the case d = 2. Since in general, we do not know the densities $P_1,\ldots,P_{NC}$, it is natural to approximate the Bayes regions by using the approximating splines s_i in place of the P_i in the definition (2.1) of the R_i. Thus, we define

$$(5.1) \qquad R_i^* = \{x \in R^2 : \alpha_i s_i(x) \geq \alpha_j s_j(x), \text{ all } j \neq i\} \quad , \quad i = 1,..,NC.$$

When the equality $\alpha_i s_i(x) = \alpha_j s_j(x)$ holds, we put x in the set R_i^* provided i is the least integer j for which $\alpha_i P_i(x) = \alpha_j P_j(x)$. The boundaries of the decision regions are contour lines defined by the equations $\delta_{ij}(x) = \alpha_i P_i(x) - \alpha_j P_j(x) = 0$.

The problem of locating the 0-level contours of the δ_{ij} can be handled by any standard contouring algorithm (see [20] for a review of several methods). We now present an algorithm which is particularly efficient for our applications since tensor-product splines can be evaluated very efficiently on grids.

Algorithm 5.1. (To find the zero contour of a spline surface s):

1. Choose positive integers ngx and ngy.
2. Choose $xg_1 < \dots < xg_{ngx}$ and $yg_1 < \dots < yg_{ngy}$.
3. Compute $s_{ij} = s(xg_i, yg_j)$, $i = 1,\dots,ngx$; $j = 1,\dots,ngy$,
4. Triangulate the grid by drawing in upward sloping diagonals (see Figure 2).
5. Identify the edges of the triangulation whose endpoints have opposite signs.
6. Use inverse interpolation to compute an approximate crossing point for each of the edges in step 5).
7. Order the points on each contour and thread a curve through them.

In practice one would normally choose the xg's and yg's to be equally spaced. The choice of ngx and ngy controls the resolution of the contouring process -- large values of these parameters will give a fine grid and correspondingly better resolution. Step 3) of the algorithm can be made very efficient for tensor-product splines (the algorithm can be used on any function s which can be evaluated at the corners of the grid). Step 7) can be accomplished using straight line interpolation if a contour consisting of a polygon (with sharp corners) is acceptable. Otherwise it is recommended that a parametric spline curve (possibly with some tension) be used. Steps 6) and 7) can be eliminated altogether if one is willing to accept a polygonal contour. In particular, if C is a contour of s, we can replace it by the boundary of the set D, where D is the smallest union of rectangles drawn from the set

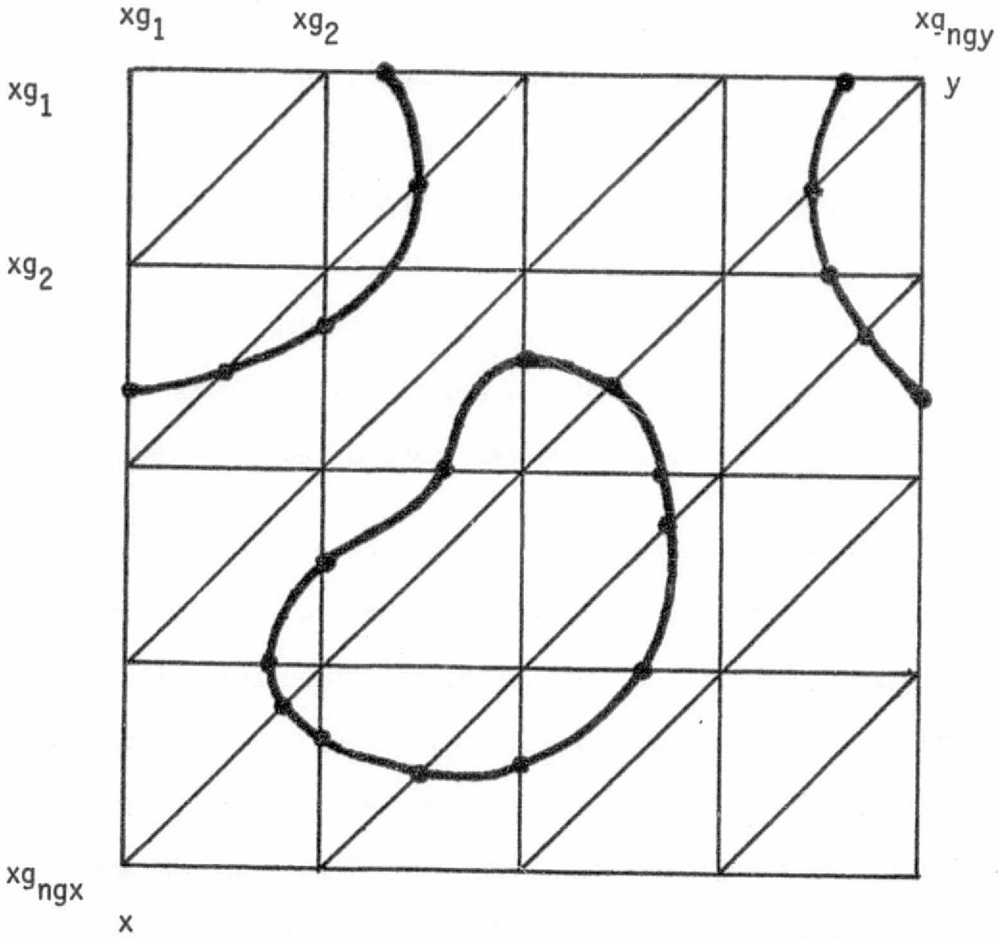


Fig. 2. A contouring algorithm.

(5.2) $\{[xg_i, xg_{i+1}] \times [yg_j, yg_{j+1}] : 1 \underline{<} i \underline{<} ngx,\ 1 \underline{<} j \underline{<} ngy\}$

such that $C \subseteq D$. Figure 3 shows a typical situation when a smooth contour has been replaced by a polygonal one.

§6. Computing The Probability Of Misclassification.

Suppose once again that $s_1, \ldots, s_{NC}$ are spline estimates of the conditional probability densities $P_1, \ldots, P_{NC}$ associated with a classification problem. Suppose in addition that $R_1^*, \ldots, R_{NC}^*$ are the approximate Bayes decision regions defined in (5.1). Then an estimate G^* for the pmc G defined in (2.2) can be computed as follows:

$$(6.1) \qquad G^* = 1 - \sum_{i=1}^{NC} \alpha_i \int_{R_i^*} s_i(x)dx \ .$$

As already observed earlier, it is possible to integrate tensor-product splines exactly over rectangular sets. Thus, if we replace the R_i^* in (6.1) by $\overline{R}_i$, where $\overline{R}_i$ is the smallest union of rectangles drawn from the set D in (5.2) such that $R^* \subseteq \overline{R}_i$, we can compute all of the the integrals in the expression

$$\underline{G} = 1 - \sum_{i=1}^{NC} \alpha_i \int_{\overline{R}_i} s_i(x)dx$$

exactly. The estimate $\underline{G}$ can be made arbitrarilly close to G by taking a sufficiently fine grid.

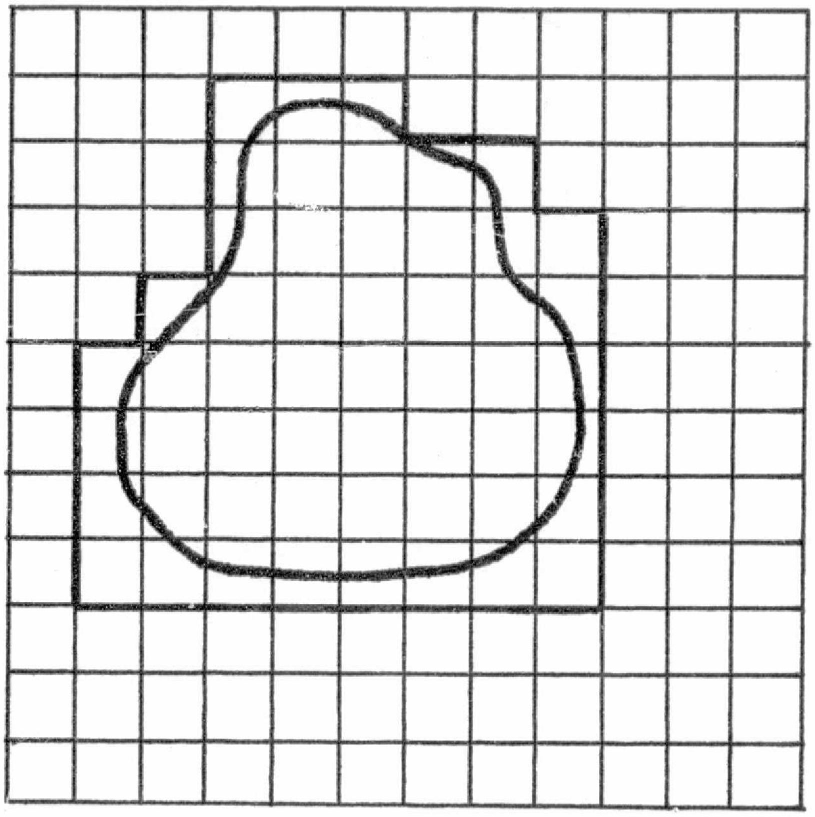


Fig. 3. A polygonal approximation of a contour.

If the s_i's are nonnegative (as they are supposed to be, although in practice they may miss by a little -- see Remark 4), then since $R_i^* \subseteq \overline{R}_i$ for all i, we concluded that $\underline{G} \leq G^*$. We can obtain an upper bound for G^* by replacing (for all i) R_i^* by a the largest $\underline{R}_i$ which is a union of rectangles drawn from the set (5.2) and which satisfies $\underline{R}_i \subseteq R_i^*$.

§7. Numerical Results

The methods discussed above have been implemented in FORTRAN, and we have begun a testing program utilizing both known distributions and actual Landsat data. Figure 4 shows a standard bivariate normal distribution with mean (1,1) and variance (.04,.04). A sample of 500 points was drawn from this distribution using a random number generator, and a histogram with bin width (.25,.25) was constructed. The result of fitting this histogram with a biquadratic spline is shown in Figure 5. A comparison of the views clearly indicates a very good fit has been achieved, although it should be noted that the spline in Figure 5 does take small negative values at some points.

Our second example involves actual Landsat data. Figure 6 shows a histogram which was generated from 500 samples taken from Channels 1 and 4 of an agricultural scene. Figure 7 shows that the corresponding spline fit is very good, despite the complexity of the histogram.

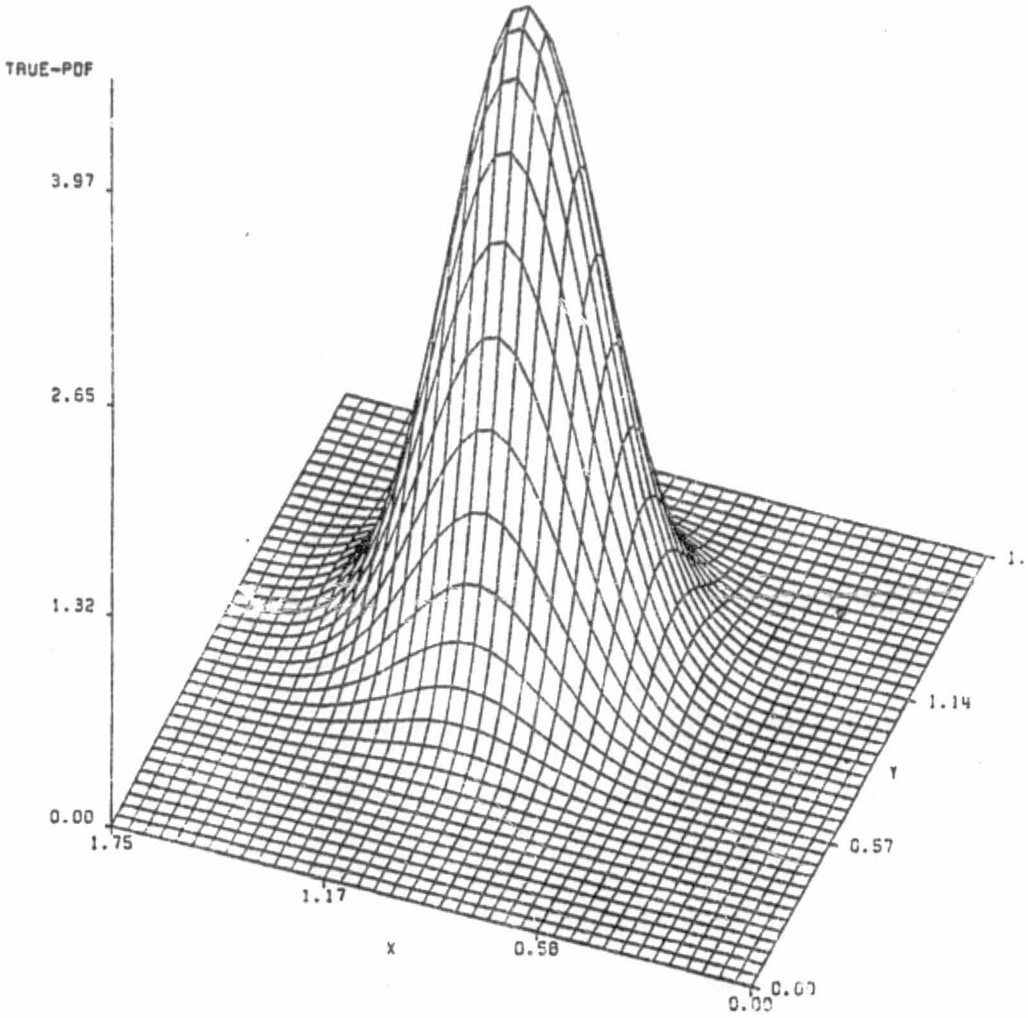


Fig. 4. Bivariate Normal Distribution mean = (1,1), variance = (.04,.04).

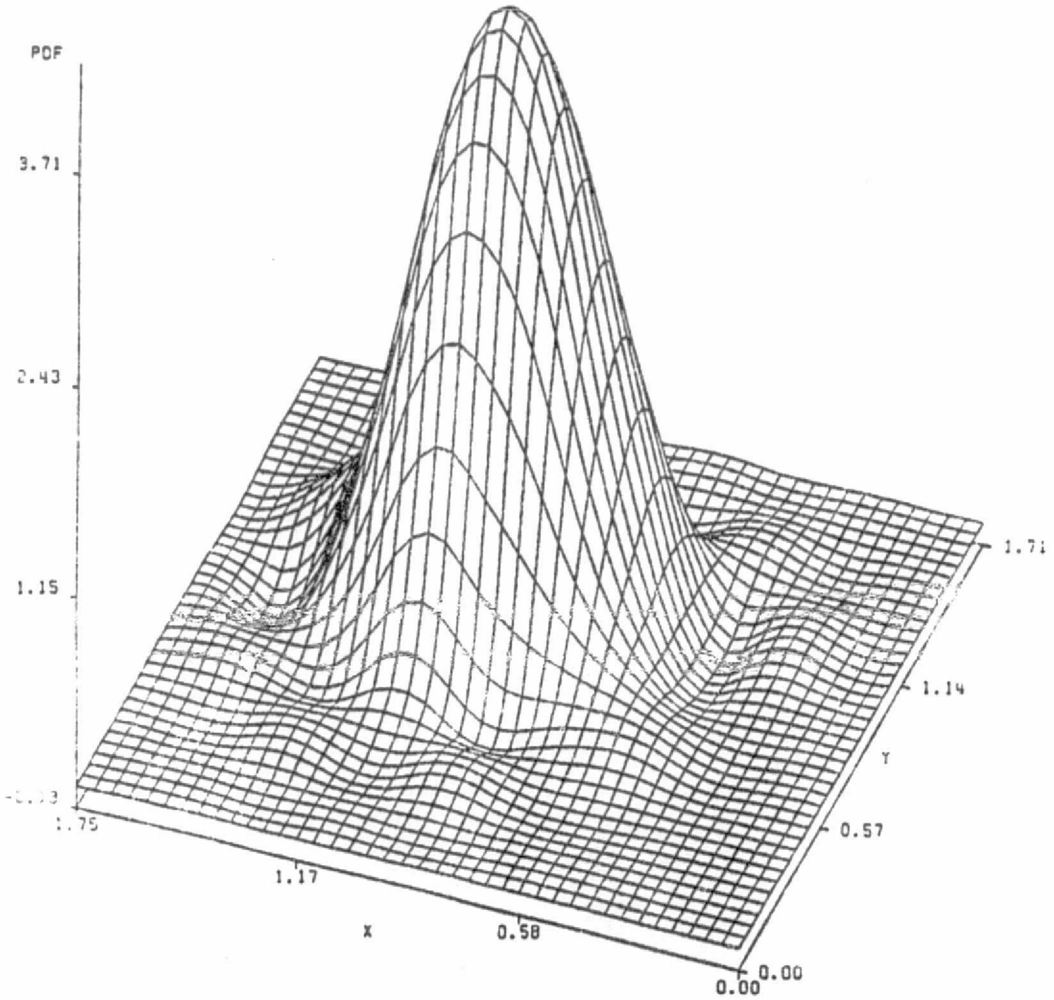


Fig. 5. Biquadratic spline fit to the normal bin width = (.25,.25).

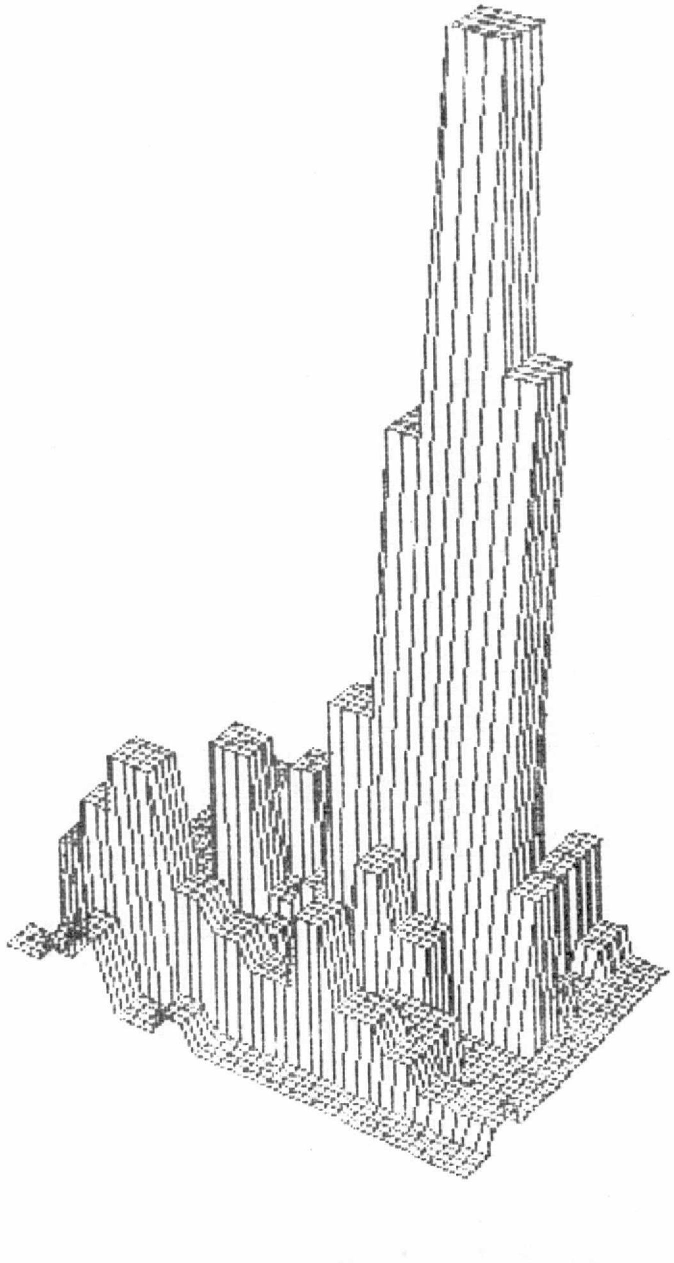


Fig. 6. Histogram of pixels from channels 1 and 4 of an agricultural scene.

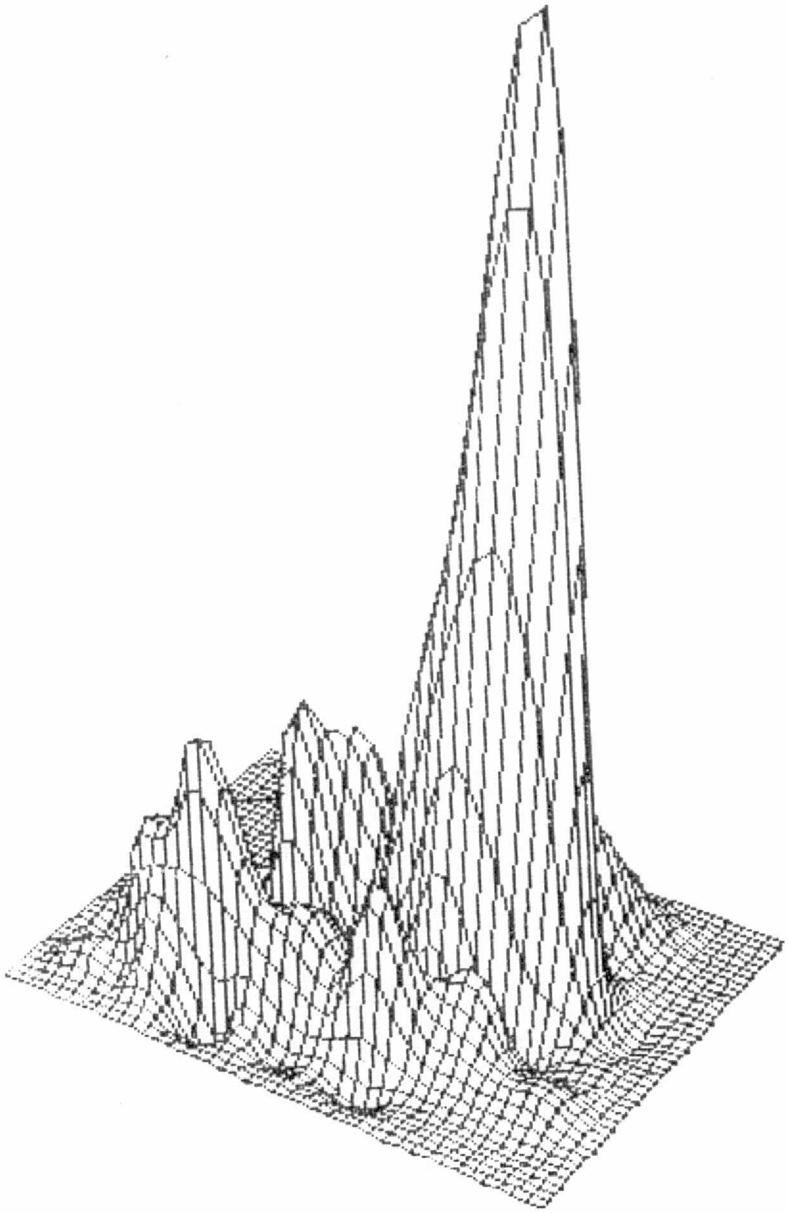


Fig. 7. The spline fit to the histogram in Figure 6.

§8. Remarks

1. The idea of fitting a spline to a histogram in such a way that the area of the histogram in each bin matches the corresponding area of the spline is due to Boneva, Kendall, & Stefanov [6]. Schoenberg [18] showed how the idea could be carried over to bivariate histograms (see also [7]).

2. The usual approach to fitting splines with area matching (cf. [7,18]) is to compute a cubic spline which fits the cumulative histogram, and then to take its derivative. In the bivariate setting one may fit a bicubic spline S to the cumulative histogram, and then the desired density fit is given by $s = D_x D_y S$. In Section 4 we have presented an algorithm which works directly with the biquadratic B-splines and the original histogram.

3. Our discussion in this paper has concentrated on the case of two dimensions. Except for some notational difficulties, there is no theoretical problem with carrying over the methods presented here to higher dimensional cases. There are, however, some practical problems. Clearly as the dimension d increases, the storage requirements for the parameters of the histogram and spline increase correspondingly. The cost of solving the analogous linear system to (4.1) also increases with d, but the tensor technique of deBoor [8,9] can still be used. Additional difficulties arise in connection with finding the Bayes regions and in computing the pmc. In particular, the Bayes regions now become subsets of d-dimensional space, and their boundaries are sets in d-1 dimensional space. We are currently planning experiments with d = 3.

4. The volume-matching algorithm presented in Section 4 is not guaranteed to produce a spline which is nonnegative. (Figure 5 shows a typical example where s takes on small negative values at some points). This is not an ideal situation since, after all, s is supposed to be approximating a density function. There are several approaches to adjusting s to obtain nonnegativity. For example, one can simply replace all negative values by zero. Alternatively, one can add small positive multiples of selected B-splines to achieve nonnegativity. We are currently exploring such post-processing schemes.

5. Another approach to achieving nonnegative spline density estimates is to determine the coefficients of the spline as the solution of some constrained optimization problem with side constraints. This is the approach which we used in [13] in the univariate case. A similar approach can be carried out in the bivariate case, but the resulting linear programming problem involves a very large tableau and hence requires considerable storage space as well as computational time. We are currently exploring special optimization methods for approximation by tensor-product splines which will take advantage of the tensor nature.

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 Verwendung 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. 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.

7. 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.

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

9. de Boor, C., Efficient computer manipulation of tensor products, ACM Trans. Math. Software 5 (1979), 173-182.

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. Cacoullos, T., Estimation of a multivariate density, Annals Inst. Stat. Math. 18 (1966), 179-189.

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 Schumaker, L. L., Spline Classification Methods, Proceedings of the NASA Symposium on Math. Pattern Recog. & Image Anal., L. F. Guseman, Jr., ed., 1983.

14. 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.

15. Lennington, R. K. and Rassbach, M. E., CLASSY - An adaptive maximum likelihood clustering algorithm, Report LEC 12145, Lockheed Electronics, 1978.

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

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

18. 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.

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

20. Schumaker, L. L. , Fitting surfaces to scattered data, in Approximation Theory II, C. K. Chui, L. L. Schumaker and G. G. Lorentz, eds., Academic Press, N. Y., 203-268.

21. Scott, D. W., Multivariate density estimation and remote sensing, Proceedings of the NASA Symposium on Math. Pattern Recog. & Image Anal., L. F. Guseman, Jr., ed., 1983, 77-92.

22. Silverman, B. W., Estimation for univariate and bivariate data, in Interpreting Multivariate Data, V. Barnett, ed., Wiley, N.Y., 1981, 37-53.

23. Tapia, R. A. and Thompson, J.R., Nonparametric Probability Density Estimation, John Hopkins Univ. Press, Baltimore, 1978.

24. Wolfe, J. H., Pattern clustering by multivariate mixture analysis, Multivariate Behavorial Research 5(1970), 329-350.

25. Yamazaki, H., Nonparametric and parametric estimation of wave statistics and spectra, Ph.D. dissertation, Texas A&M Univ., 1984.

D7

AUTOREGRESSIVE SPECTRAL ESTIMATION FOR TWO DIMENSIONAL TIME SERIES

H. Joseph Newton
and
William B. Smith

Texas A&M University

ABSTRACT

The method of determining asymptotic confidence bands for autoregressive spectra due to Newton and Pagano [3] is extended to the case of data observed in the plane. One Quadrant Autoregressive Models are used as a basis for the method.

1. INTRODUCTION

The analysis of data observed over a regular grid of points in multi-dimensional space has become an important problem in recent years; particularly in the area of remotely sensed data such as data observed by satellite. Viewing such data as a realization from a time series having a vector index set representing the location of the observations, one seeks to extend the results for scalar index set to the more general case. The purpose of this paper is to extend the method of one dimensional autoregressive spectral confidence bands to the two dimensional case. The extension to more than two dimensions is straightforward.

2. TWO DIMENSIONAL TIME SERIES

We say that the collection of random variables $\{X_{t,\tau}, t,\tau=0,\pm 1,\pm 2, \ldots\}$ is a (weakly) stationary two dimensional time series if

i) $E(X_{t,\tau}) = \mu \quad \forall t,\tau$

ii) There exists a function R(called the autocovariance function of X) having integer valued arguments such that

$R(s,u) = \mathrm{Cov}\,(X_{t,\tau}, X_{t+s,\tau+u}) \quad \forall t,\tau$.

An important function for modeling and interpreting two dimensional time series is the spectral density function f of X which is given by (if it exists) the two dimensional Fourier Transform of R:

$$f(\omega,\lambda) = \left(\frac{1}{2\pi}\right)^2 \sum_{s=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} R(s,u)\, e^{-is\omega-iu\lambda} \quad , \omega,\lambda\varepsilon[-\pi,\pi] \ .$$

Given data $\{X_{t,\tau}\}$, $t=1,\ldots,n_1$, $\tau=1,\ldots,n_2$ from a zero mean series X, we may estimate (without assuming any parametric model for X) R by the sample autocovariance function

$$\hat{R}(s,u) = \frac{1}{n_1 n_2} \sum_{t=1}^{n_1-|s|} \sum_{\tau=1}^{n_2-|u|} X_{t,\tau} X_{t+|s|,\tau+|u|} \; , \quad |s|<n_1, |u|<n_2, \quad (2.1)$$

and f by the windowed periodogram spectral estimate

$$\hat{f}(\omega,\lambda) = (\frac{1}{2\pi})^2 \sum_{s=-(n_1-1)}^{n_1-1} \sum_{u=-(n_2-1)}^{n_2-1} k(s,u)\hat{R}(s,u)e^{-is\omega-iu\lambda}$$

$$= \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} K(\omega-\mu,\lambda-\nu) I(\mu,\nu) d\mu d\nu \quad ,$$

where the periodogram I of X is given by

$$I(\mu,\nu) = \frac{1}{(2\pi)^2 n_1 n_2} \left| \sum_{t=1}^{n_1} \sum_{\tau=1}^{n_2} X_{t,\tau} e^{-it\mu-i\tau\nu} \right|^2 \quad ,$$

and k is a suitably chosen lag window, while the spectral window K is given by

$$K(\mu,\nu) = (\frac{1}{2\pi})^2 \sum_{s=-(n_1-1)}^{n_1-1} \sum_{u=-(n_2-1)}^{n_2-1} k(s,u) \; e^{-is\mu-iu\nu} \quad .$$

In the one dimensional case an alternative model-oriented method of estimating f is to assume that X can be adequately approximated by a causal autoregressive model in which case f can be expressed as a function of a few parameters which can be well estimated from the data.

the idea of modeling an observation at time t as a linear function of finitely-many previously observed values is intuitively appealing when the index set is time. However, defining the causal analogue for spatial series is notoriously difficult (see Whittle [6] and Besag [1] for example) as there is no natural ordering of the data. Tjφstheim [4], [5] has investigated analytically and numerically the efficacy of what are called one-quadrant autoregressive models as approximating models for X. In this paper we adopt this approach and find simultaneous confidence bands for the spectral density function of such a process.

3. ONE QUADRANT AUTOREGRESSIVE PROCESSES

The two dimensional time series X is called a one quadrant autoregressive process of orders p_1, p_2 if

$$\sum_{j=0}^{p_1} \sum_{k=0}^{p_2} a(j,k)\, X_{t-j,\tau-k} = \varepsilon_{t,\tau} \tag{3.1}$$

where a(0,0)=1 and ε is a two dimensional white noise time series, i.e. the ε's are zero mean, uncorrelated random variables with common variance σ^2. We write X QAR (p_1,p_2,a,σ^2). Tjφstheim [5] discusses sufficient conditions for a spatial series to be representable as a QAR and suggests that the class of processes so representable is quite large.

If the complex valued polynomial

$$g(z_1,z_2) = \sum_{j=0}^{p_1} \sum_{k=0}^{p_2} a(j,k)\, z_1{}^j z_2{}^k$$

satisfies $g(z_1,z_2)=0 \Rightarrow |z_1|>1$, $|z_2|>1$ then the QAR is called stable and X is stationary. Also one has Yule-Walker type equations (obtained by multiplying both sides of (3.1) by $X_{t-r,\tau-s}$ for $r,s\geq 0$, taking expectations on both sides, and noting that $\varepsilon_{t,\tau}$ is uncorrelated with $X_{t-r,\tau-s}$ unless r and s are both zero)

$$\sum_{j=0}^{p_1}\sum_{k=0}^{p_2} a(j,k)R(j-s,k-s) = \delta_r\delta_s\sigma^2 , \quad r,s\geq 0 \tag{3.2}$$

where δ_v is the Kronecker delta, i.e. δ_v is one if V=0 and zero otherwise. Also, the spectral density of X is given by

$$f(\omega,\lambda) = \frac{\sigma^2}{(2\pi)^2} \frac{1}{\left|\sum_{j=0}^{p_1}\sum_{k=0}^{p_2} a(j,k)\, e^{-ij\omega-ik\lambda}\right|^2} \tag{3.3}$$

Solving (3.2) for $r=0,\ldots,p_1$ and $s=0,\ldots,p_2$ with the $\hat{R}$'s defined by (2.1) replacing the R's, one can obtain estimators $\hat{\sigma}^2$ and $\hat{a}(j,k)$ which can be inserted into (3.3) to obtain an estimator $\hat{f}$ of f. Justice [2] describes a Levinson type algorithm for efficiently solving (3.2) recursively for varying values of p_1 and p_2. Tjφstheim [5] discusses methods of optimally choosing the orders p_1 and p_2 but we shall derive our inferences contingent on having adequately chosen them.

4. ASYMPTOTIC CONFIDENCE BANDS FOR QAR

In the one dimensional setting, the reciprocal of the autoregressive spectral density is a linear combination of a finite number of parameters that can be estimated by asymptotically normal estimators. Thus Scheffé projections can be used to find asymptotic confidence bands on the entire autoregressive spectral density (see Newton and Pagano [3]). In this section we describe the analogous procedure for a AQR(p_1,p_2,a,σ^2) process.

There are $p=(p_1+1)(p_2+1)-1$ a's to be estimated (a(j,k) for j=0,..., p_1, k=0,...,p_2, but a(0,0)=1). Define the vec operator on an n×m matrix A having columna $\underset{\sim}{a}_1,\ldots,\underset{\sim}{a}_m$ to be the process of forming the nm×1 vector $\underset{\sim}{a}$ = vec (A) by $\underset{\sim}{a} = (\underset{\sim}{a}_1^T,\ldots,\underset{\sim}{a}_m^T)^T$. Let tvec (A) be the result of removing the top element of vec (A).

Let A and R be (p_1+1) (p_2+1) matrices having (j,k)th elements a(j,k), and R(j,k), j=0,...,p_1, k=0,...,p_2 respectively. Then one can write the Yule-Walker equations (3.2) in amtrix form as

$$\mathcal{R}\,\underset{\sim}{a} = -\,\underset{\sim}{r}$$

where $\underset{\sim}{a}$ = tvec (A), $\underset{\sim}{r}$ = tvec (R), the first p_1 rows of $\mathcal{R}$ correspond to (3.2) for s=0 and r=1,...,p_1, the next p_1+1 are for s=1 and r=0,...,p_1, and the last p_1+1 are for s=p_2 and r=0,...,p_1. To form $\mathcal{R}$ explicitly, note that $\underset{\sim}{a}$ and $\underset{\sim}{r}$ can be partitioned into a p_1×1 vector $\underset{\sim}{a}(0)$, $\underset{\sim}{r}(0)$ followed by p_2 vectors $\underset{\sim}{a}(1),\ldots,\underset{\sim}{a}(p_2)$; $\underset{\sim}{r}(1),\ldots,\underset{\sim}{r}(p_2)$ each of length p_1+1. Thus $\mathcal{R}$ can be partitioned into a matrix containing p_2+1 rows and columns of blocks. Call the (i,j)th block C(i,j), i,j=0,...,p_2. Then C(0,0) is $p_1 \times p_1$; C(0,j) is $p_1 \times (p_1+1)$, j=1,...,p_2; C(j,0) is $(p_1+1) \times p_1$, j=1,...,p_2; while for j,k$\geq$1, C(j,k) is $(p_1+1)\times(p_1+1)$. Thus

$$\sum_{k=0}^{p_2} C(j,k)\ \underset{\sim}{a}(k) = -\underset{\sim}{r}(j) \quad , \quad j=0,\ldots,p_2 \quad .$$

A careful inspection of this equation shows that

$$C_{\ell m}(0,0) = R(m-\ell,0) \quad , \quad \ell,m=0,\ldots,p_1-1$$

$$C_{\ell m}(0,j) = C_{m\ell}(j,0) = R(m-\ell-j,j) \quad , \quad \begin{array}{l} j=1,\ldots,p_2 \\ \ell=0,\ldots,p_1-1 \\ m=0,\ldots,p_1 \end{array}$$

$$C_{\ell m}(j,k) = R(m-\ell,\ k-j) \quad , \quad \begin{array}{l} j,k=1,\ldots,p_2 \\ \ell,m=0,\ldots,p_1 \end{array}$$

and in fact $\mathcal{R}$ is the block Toeplitz matrix having (j,k)th block $G(k-j)$ where $G_{\ell m}(v) = R(m-\ell,v)$, $|v|\leq p_2, \ell,m=0,\ldots,p_1$, except that the first row and column have been removed. Thus to form $\mathcal{R}$ one needs $R(r,s)$ for $|r|\leq p_1$, $|s|\leq p_2$.

Theorem (Tjøstheim [4])

Let a be the solution to the Yule-Walker equations (3.2) $\hat{\mathcal{R}}\ \hat{\underset{\sim}{a}} = -\hat{\underset{\sim}{r}}$ where $\hat{\mathcal{R}}$ and $\hat{\underset{\sim}{r}}$ are the same as $\mathcal{R}$ and $\underset{\sim}{r}$ with $\hat{R}(j,k)$ replacing $R(j,k)$. Suppose X is a stable QAR(p_1,p_2,a,σ^2) where the white noise series ε is of independent, identically distributed random variables. Then as $n_i\to\infty$, i=1,2, we have that

$$\sqrt{n_1 n_2}\ \ (\hat{\underset{\sim}{a}}-\underset{\sim}{a}) \xrightarrow{\mathcal{D}} N_p(\underset{\sim}{0},\sigma^2\mathcal{R}^{-1})$$

(recall that $p = (p_1+1)(p_2+1)-1$) .

Now the reciprocal of the QAR spectral density is given by

$$f_Z(\omega,\lambda) = \frac{1}{f(\omega,\lambda)} = \frac{(2\pi)^4/\sigma^2}{(2\pi)^2} \left| \sum_{j=0}^{p_1} \sum_{k=0}^{p_2} a(j,k)\, e^{-ij\omega-ik\lambda} \right|^2$$

which is the spectral density function of a two dimensional moving average process Z of orders p_1 and p_2 having coefficients $a(j,k)$ and noise variance $(2\pi)^4/\sigma^2$ (See Tjϕstheim [4]). Thus

$$Z_{t,\tau} = \sum_{j=0}^{p_1} \sum_{k=0}^{p_2} a(j,k)\, \eta_{t-j,\tau-k}$$

where η is a white noise process having variance $(2\pi)^4/\sigma^2$. Now clearly $R_Z(r,s) = \mathrm{Cov}\,(Z_{t,\tau}, Z_{t+r,\tau+s})$ is zero for $|r|>p_1$ and $|s|>p_2$ while $R_Z(r,s)$ is a complicated function g of the a's and σ^2, i.e $R_Z(r,s) = g(a,\sigma^2)$. Thus

$$f_Z(\omega,\lambda) = \frac{1}{(2\pi)^2} \sum_{|r|\le p_1} \sum_{|s|\le p_2} R_Z(r,s)\, e^{-ir\omega-is\lambda}$$

$$= \frac{1}{(2\pi)^2} \sum_{|r|\le p_1} \sum_{|s|\le p_2} R_Z(r,s)\, [\mathrm{Cos}\, r\omega\, \mathrm{Cos}\, s\lambda - \mathrm{Sin}\, r\omega \mathrm{Sin}\, s\lambda]$$

which is a linear function of $R_Z(r,s)$ for $|r|\le p_1$, $|s|\le p_2$. Since $\hat{\underset{\sim}{a}}$ and $\hat{\sigma}^2$ are asymptotically normal by the above theorem, we have that the $\hat{R}_Z(r,s) = g(\hat{a},\hat{\sigma}^2)$ are also asymptotically normal and one can use Scheffé projections to get simultaneous confidence bands on f_Z and thus on f. It remains to obtain convenient expressions for the asymptotic covariances of the $\hat{R}_Z$ and to determine the best way to display the bands graphically.

REFERENCES

[1] Besag, J. (1974), "Spatial Interaction and the Statistical Analysis of Lattice Systems," Journal of the Royal Statistical Society, B36, 192-236.

[2] Justice, J. H. (1977), "A Levinson-Type Algorithm for Two-Dimensional Wiener Filtering Using Bivariate Szegö Polynomials," Proc. IEEE, 65, 822-886.

[3] Newton, H. J., and Pagano, M. (1984), "Simultaneous Confidence Bands for Autoregressive Spectra," Biometrika, 71, 197-202.

[4] Tjϕstheim, D. (1978), "Statistical Spatial Series Modeling," Adv. Appl. Prob., 10, 130-154.

[5] ____________ (1981), "Autoregressive Modeling and Spectral Analysis of Array Data in the Plane," IEEE Trans on Geoscience and Remote Sensing, G#19, 15-24.

[6] Whittle, P. (1954), "On Stationary Processes in the Plane," Biometrika, 41, 434-449.

EXPLORING THE USE OF STRUCTURAL MODELS TO IMPROVE REMOTE-SENSING AGRICULTURAL ESTIMATES

Richard F. Gunst and Mani Y. Lakshminarayanan
Department of Statistics
Southern Methodist University

ABSTRACT

Satellite estimates of agricultural characteristics often are not sufficiently precise for reliable use in small geographical regions. The precision of estimates of agricultural characteristics such as crop proportions and leaf area indexes can be increased by modelling ground observations as a function of satellite estimates. Linear regression models using least squares estimators of the model parameters is most often advocated as an appropriate methodology; however, least squares estimation requires that the predictor variables are measured without error, an unreasonable assumption for this application. An alternative estimation methodology which assumes that both the response variables (ground observations) and the predictor variables (satellite estimates) are measured with error involves the use of linear structural models. In this paper the application of linear structural models to the estimation of agricultural characteristics using satellite spectral measurements is examined.

1. Introduction

Satellite remote sensing is an important technology for rapid collection and processing of spectral information on agricultural and vegetation characteristics. Estimation of crop acreage and biomass are but two of the many potential applications of satellite remote-sensing technology. However, the precision needed for reliable estimation of agricultural and vegetation characteristics often is not obtainable solely from satellite estimates, especially for geographical regions as small as counties. For this reason ground observations from selected sample locations are often used in conjunction with satellite spectral measurements to obtain estimates for geographical regions of interest. In this paper the use of linear structural models to obtain estimates of agricultural characteristics from ground observations and satellite measurements is investigated.

Linear structural models assume that a variable of interest, the response variable (Y), is a linear function of another measurement, a stochastic predictor variable (X):

$$(1.1) \qquad Y = \alpha + \beta X .$$

In addition, both the response variable and the predictor variable are assumed to be measured with error; i.e., x and y are observable, where

$$(1.2) \qquad x = X + u \quad \text{and} \quad y = Y + v.$$

Classical linear regression models assume that the error in x is zero (i.e., u = 0) or at least negligible and that the predictor variables are known constants.

Although ground observations can be expected to have less error than estimates obtained from satellite spectral measurements, both ground observations and satellite estimates are subject to measurement error. For example, there are many sources of measurement error in the calculation of leaf area indexes from ground observation: trees must be felled and the leaves collected, weighed, and their individual areas calculated. Likewise, satellite spectral measurements are subject to several sources of error including registration error, randomness associated with the selection of segments and pixels with which to train classifiers, and technician error in the identification of pixels. Thus structural models in which both the response and predictor variables are assumed to be subject to measurement error present a more realistic framework from which to estimate many agricultural and vegetation quantities from satellite measurements.

In Section 2 of this paper the theoretical properties of maximum likelihood estimators of the parameters in linear structural models which assume independent normal distributions for X, u, and v are outlined and conditions under which these estimators reduce to least squares estimators are noted. In Section 3 the dependence of linear structural estimators on knowledge of the ratio of error variances (i.e., var(v)/var(u)) is assessed. Two applications of this methodology are discussed in Section 4. Section 5 contains concluding remarks and mentions several extensions which are currently under investigation.

2. Linear Structural Models

Either of two assumptions can be added to the linear model defined by equations (1.1) and (1.2) to define the nature of the true (unobservable) predictor variable X. If the true values of the predictor variable X are assumed to be constants, the model is referred to as a linear functional model. If the true values of the predictor variable are assumed to be stochastic, the model is referred to as a linear structural model. The focus of this investigation is on linear structural models for which the measurement errors and the predictor variable X are assumed to be independent normal random variables:

(2.1) $X \sim N(\mu_X,\sigma_X^2)$, $u \sim N(0,\sigma_u^2)$ and $v \sim N(0,\sigma_v^2)$.

Together equations (1.1), (1.2), and (2.1) constitute the linear structural model of interest in this work.

Under the assumptions stated above, the joint distribution of a random sample of n observations (x_i,y_i) is bivariate normal; i.e.,

$$(2.2) \quad \begin{pmatrix} x_i \\ y_i \end{pmatrix} \sim N \left\{ \begin{pmatrix} \mu_X \\ \alpha+\beta\mu_X \end{pmatrix}, \begin{pmatrix} \sigma_X^2+\sigma_u^2 & \beta\sigma_X^2 \\ \beta\sigma_X^2 & \beta^2\sigma_X^2+\sigma_v^2 \end{pmatrix} \right\} \quad i=1,2,\ldots,n$$

The maximum likelihood estimating equations for the parameters in the joint distribution (2.2) are:

$$(2.3) \quad \begin{array}{ll} \hat{\mu}_X = \bar{x} & \hat{\alpha}+\hat{\beta}\hat{\mu}_X = \bar{y} \\ \hat{\sigma}_X^2+\hat{\sigma}_u^2 = s_x^2 & \hat{\beta}^2\hat{\sigma}_X^2+\hat{\sigma}_v^2 = s_y^2 \\ \hat{\beta}\hat{\sigma}_X^2 = s_{xy} & \end{array}$$

where $\bar{x}$ and $\bar{y}$ are sample means, s_x^2 and s_y^2 are sample variances, and s_{xy} is the sample covariance. There are six model parameters which must be estimated from these five estimating equations; equivalently, there are five sufficient statistics from which to estimate the six model parameters. Without (a) knowledge of one or more model parameters, (b) replication, or (c) the availability of one or more additional variables ("instrumental variables") which are correlated with X the regression coefficients in (1.1) cannot be estimated consistently under the normality assumptions (2.1) because the model lacks identifiability (Reiersol [8]).

Satellite remote sensing does not generally allow the type of experimental control which permits the collection of independent replicated observations on both x and y. Likewise, satellite spectral readings are usually converted to a single estimate of the characteristic of interest, thereby precluding an instrumental variables analysis. In these situations consistent estimation of the model parameters requires some knowledge of the parameters themselves.

The model parameters which are of primary interest in the study of the structural model defined by (1.1), (1.2), and (2.1) are α, β, μ_X, and σ_X^2. Since $\hat{\mu}_X = \bar{x}$ and $\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$, the estimation of β presents the only serious problem to the estimation of model (1.1). Consequently one must be able to assume some knowledge of the error variances σ_u^2 and σ_v^2 in order to consistently estimate the remaining model parameters, in particular to estimate β. Kendall and Stuart ([4], Chapter 29) detail the solutions to the likelihood equations when one or both of the error variances is known, as well as the solution when the ratio of error variances $\lambda = \sigma_v^2/\sigma_u^2$ is known.

The assumption that the ratio of error variances is known is perhaps the most frequently-cited condition which is imposed to solve the likelihood equations. This assumption does not require explicit knowledge of either of the error variances and one often encounters analyses for which it is reasonable to conclude that the error variances are equal (i.e., $\lambda = 1$). Knowledge of the error variance ratio also insures that the variance estimates calculated from equations (2.3) are nonnegative, a property which is not guaranteed when one or both of the error variances are assumed known.

The solution for $\hat{\beta}$ when λ is assumed known is

(2.4) $\hat{\beta} = s(\lambda) + \mathrm{sign}(s_{xy})\{s^2(\lambda) + \lambda\}^{1/2}, \; s(\lambda) = (s_y^2 - \lambda s_x^2)/(2s_{xy}).$

Due to the nonnegativity of the error variance estimators when λ is known, the estimating equations (2.3) provide the following bounds on the magnitude of the structural model slope estimator:

(2.5) $|s_{xy}|/s_x^2 \le |\hat{\beta}| \le s_y^2/|s_{xy}|$.

The lower bound in inequality (2.5) is the least squares estimator of the slope parameter for the regression of y on x and the upper bound is the inverse of the least squares estimator for the regression of x on y. These two limits corresponed to structural model estimators when it is known that there is no error in the predictor or the response variable, respectively. The latter estimator is also referred to in the literature on linear calibration as an "classical" least squares estimator (e.g., Lwin and Maritz [7]).

Using the method of statistical differentials (e.g., Serfling [10], Chapter 6) one can reexpress estimator (2.4) in a Taylor series expansion about the true parameter value. By truncating this series one can approximate the distribution and moments of $\hat{\beta}$. Anderson [1]) cautions that the asymptotic properties so derived pertain to the approximation to $\hat{\beta}$ and not to $\hat{\beta}$ itself; nevertheless, the asymptotic moments of the Taylor series expansion provide a potentially useful description of the behavior of the structural model estimator.

Replacing the sample moments in (2.4) by their corresponding parameter values one readily establishes the consistency of $\hat{\beta}$. Applying the method of statistical differential to a first-order approximation to (2.4), the asymptotic variance of this approximation to $\hat{\beta}$ is to $O(n^{-2})$ (Lakshminarayanan and Gunst [5]):

(2.6) $n^{-1}\{(\beta^2+\lambda)\gamma + \lambda\gamma^2\}$,

where $\gamma = \sigma_u^2/\sigma_X^2$ is the "noise-to-signal" ratio for the observable predictor variable x. For comparative purposes, the asymptotic mean squared error of the least squares estimator $\hat{\beta}_{LS} = s_{xy}/s_x^2$ is (c.f. Richardson and Wu [9], equations (2.24) and (2.25)):

(2.7) $\beta^2\gamma(n^{-1}+\gamma)(1+\gamma)^{-2} + n^{-1}\lambda\gamma(1+\gamma)^{-1}$.

When the error variance ratio λ is incorrectly specified, the structural model estimator (2.4) is no longer consistent. Again ignoring terms to $O(n^{-2})$, the asymptotic expectation and variance of a first-order approximation to (2.4) using an assumed value of λ^* for the error variance ratio is (Lakshminarayanan and Gu:st, [5]):

(2.8) $E(\hat{\beta}) = g_\lambda(\lambda^*) + \text{sign}(\beta)\{g_\lambda^2(\lambda^*) + \lambda^*\}^{1/2}$

(2.9) $\text{Var}(\hat{\beta}) = n^{-1}(\beta^2+\lambda^*)^{-2}[3\beta^2\gamma^2(\lambda-\lambda^*)^2+(\beta^2+\lambda^*)^2\{(\beta^2+\lambda)\gamma+\lambda\gamma^2\}]$

where $g_\lambda(\lambda^*) = \{(\beta^2-\lambda^*)\sigma_X^2+(\lambda-\lambda^*)\sigma_u^2\}/(2\beta\sigma_X^2)$. When $\lambda^* = \lambda$, $E(\hat{\beta}) =$

β and equation (2.9) reduces to equation (2.6).

While the structural model estimator (2.4) and approximate asymptotic properties such as equation (2.6) are routinely used, few theoretical or empirical studies have been conducted to evaluate the performance of $\hat{\beta}$ when (i) the error variances ratio is incorrectly chosen, or (ii) only small samples of data are available. Such evaluations are especially important for the present study since one generally encounters small numbers of sample locations for which both ground observations and satellite estimates are available and the true variance ratio is not known. The next section presents a detailed investigation of these two issues.

3. Variance Ratio and Sample Size Effects

Lakshminarayanan and Gunst [5] report the results of an investigation on the effects of two factors on the performance of the structural model estimator (2.4): the choice of the variance ratio λ and sample size. In this section the results reported by Lakshminarayanan and Gunst are presented in greater detail. In particular, (i) the structural model estimator is shown to be insensitive to the choice of the variance ratio only when λ is large and γ is small, (ii) the structural model estimator is shown to possess a smaller mean squared error than least squares only when the variance ratio is chosen in a relatively narrow neighborhood of the true value, and (iii) satisfactory use of asymptotic formulae for variance estimation requires a sample size in excess of 200.

Asymptotically (i.e., replacing the sample moments by their parameter values), (3.1) $\partial\hat{\beta}/\partial\lambda = -\beta\gamma/(\beta^2 + \lambda)$

indicating that the rate of change of $\hat{\beta}$ with respect to λ depends on the true values of β, λ, and γ. Figure 1 is a graph of the relative rate of change $|\partial\hat{\beta}/\partial\lambda|/\beta$. This figure illustrates the general features of equation (3.1): $\hat{\beta}$ is relatively insensitive to the true value of λ for large values of λ and small values of γ (holding β fixed). Together these two conditions imply that σ_u^2, the error variance for the observable variable x, is small. In other words if σ_u^2 is not negligible the linear structural estimator can be very sensitive to the true value of λ, suggesting that an incorrect choice of λ could substantially alter the estimator.

Alternatively, one might wish to assess the sensitivity of $\hat{\beta}$ to λ when the variance ratio is assumed to be stochastic rather than constant. Lindley and El-Sayyad [6] propose that a Uniform(k^{-1},k) prior for λ be assumed if the measurement errors are believed to be of the same magnitude. Other reasonable priors include $N(k,\sigma_\lambda^2)$ and Chisquare(k) distributions. If one approximates the expectation of (3.1) using a three-term Taylor series expansion, the approximate expectations for the above three priors are, respectively,

(3.2) $-2\beta\gamma[(2\beta^2+k+k^{-1})^{-1} + (k-k^{-1})^2\{3(2\beta^2+k+k^{-1})^3\}^{-1}]$

(3.3) $-\beta\gamma\{(\beta^2+k)^{-1} + \sigma_\lambda^2(\beta^2+k)^{-3}\}$

(3.4) $-\beta\gamma\{(\beta^2+k)^{-1} + 2k(\beta^2+k)^{-3}\}$.

Equations (3.2)-(3.4) (divided by β) are graphed in Figures 2-4. The same overall conclusions drawn from Figure 1 are apparent in these graphs: the structural model estimator (2.4) is relatively insensitive to the true value of the variance ratio only when $\lambda(k)$ is large and γ is small.

The asymptotic moments (2.8) and (2.9) demonstrate two important properties of the structural model estimator when the variance ratio is chosen incorrectly. When λ is chosen incorrectly equation (2.8) shows that $\hat{\beta}$ is biased. In addition, $\hat{\beta}$ has a larger variance (compare equations (2.6) and (2.9)) than the structural model estimator which is obtained with the correct value of λ. Thus not only is the structural model estimator sensitive to the choice of the variance ratio but its mean squared error properties are also affected by both the true value of λ and by an incorrect choice of the variance ratio.

Figure 5 is a graph of the ratio of the asymptotic variance, equation (2.6), of the structural model estimator to the mean squared error, equation (2.7), of the least squares estimator (recall that the structural model estimator is asymptotically unbiased). In this figure β, σ_X^2 and σ_v^2 are fixed at 3,5, and 10, respectively, so that by varying σ_u^2 both λ and γ are simultaneously varied; in particular, small values of σ_u^2 correspond to large values of λ and small values of γ. As the figure indicates, unless λ is very small, corresponding to a large error variance for the predictor variable relative to that of the response variable, the structural model estimator has a smaller asymptotic mean squared error than the least squares estimator.

Figures 6-9 display ratios of the asymptotic mean squared errors of the structural model estimator with an incorrect choice of the variance ratio to that of the least squares estimator. These figures use the same model parameters as does Figure 5 but with σ_u^2 selected so that the true variance ratio is 1, 6, and 10, respectively. The figures demonstrate that the assumed variance ratio must be chosen in a re-

C-3

latively narrow interval around the true value in order for the structural model estimator to have a smaller mean squared error than least squares.

The foregoing theoretical properties are asymptotic and do not necesssarily indicate properties of the structural model estimator for finite sample sizes. It is particularly important to assess the behavior of $\hat{\beta}$ for finite sample sizes because the asymptotic moments which were derived in the last section pertain to a Taylor series approximation to $\hat{\beta}$ and not to the true distribution of the structural model estimator.

In Table 1-4 1000 replications of samples of size n were generated from the structural model defined by equations (1.1), (1.2), and (2.1) using normal variates from I.M.S.L. subroutine GGNML on a C.D.C. 6600 computer. Unless otherwise specified, β, σ_u^2, and σ_X^2 are fixed at 3, 5, and 5, respectively (thus $\gamma = 1$). By varying σ_v^2 the results are only a function of λ and n. The values in the tables are displayed as a function of the true value of λ and an assumed value, λ^*. Correct choices of the variance ratios correspond to entries for which $\lambda^* = \lambda$.

Table 1 displays ratios of the average of 1000 $\hat{\beta}$ values calculated from equation (2.4) to the true value of β. For samples of size n = 50 and 100 the maximum relative error in estimating $\hat{\beta}$ using the correct value of the variance ratio is 4%. Incorrectly choosing λ^* larger than the true variance ratio results in underestimation of β whereas too small a value of λ^* results in overestimation of β.

Estimated and asymptotic mean squared errors for the structural model estimator are compared in Table 2. Estimated mean squared errors are calculated from the usual formula:

$$\text{mse} = \Sigma(\hat{\beta}_i - \beta)^2/1000$$

and asymptotic mean squared errors are calculated from equation (2.6) using the true parameter values. The ratios in Table 2 corresponding to the correct choice of the variance ratio indicate that use of asymptotic formulae for moments of structural model estimators cannot be recommended for samples of size n = 100 or less. Errors of 15-30% between sample and theoretical mean squared errors occur for samples of size 100 when the variances ratio is correctly chosen; much larger errors occur when the variance ratio is incorrectly chosen.

Tables 3 and 4 display ratios of sample and asymptotic mean squared errors for samples of size n = 200 and several values of β, γ, λ, and λ^*. When λ is correctly chosen the ratios are much closer to 1.0 in these tables than in Table 2. If relative errors of approximately 10% or less are acceptable, these tables indicate that samples of size n = 200 could be considered minimally acceptable for a wide range of model parameters. These tables also demonstrate that λ^* must be selected near its true value for the asymptotic variance formula (2.6) to provide a reasonable assessment of the variability of $\hat{\beta}$. When β is small it is especially undesirable to choose values of λ^* which are less than the true ones. The deleterious effects of erroneous selection of the variance ratio decrease with larger values of β and smaller values of γ; moreover, when γ = 0.1 and β= 10 the ratios in Table 3 indicate a relative error of 15% or less for most of the cases in which $\lambda^* \neq \lambda$.

4. Applications

In this section the use of the structural model estimator (2.4) is examined on two data sets for which ground observations and satellite classifier estimates are available for the proportion of corn grown in each of several segments or portions of segments. No attempt is made to obtain a final prediction equation for these examples; rather, they are used merely to illustrate several important features of structural model estimators.

The first example is taken from Badhwar, Carnes, and Austin [3]. The data set consists of 41 segments for which the proportion of corn in the segments has been determined from ground observation. The satellite estimates are obtained from a temporal model of crop greenness (Badhwar [2]). Figure 9 is a scattergram of the ground truth proportions versus the Badhwar estimates of the proportions taken from the raw data listed in Table 5.

For illustration purposes, assume that the structural model defined by equations (1.1), (1.2), and (2.1) is an adequate representation of the relationship between the true proportions and their satellite estimates. Note that the variance ratio λ is not known. Inequality (2.5) provides bounds on the structural model slope estimate:

$$1.001 \leq \hat{\beta} \leq 1.291.$$

Thus the slope estimate is bounded in a relatively narrow interval. Table 6 lists estimates for several values of the variance ratio. Observe that the greatest change in the estimates occurs for variance ratios in the interval [0,1]. If it is believed that ground observation is subject to less error than satellite estimates, the variance ratio would be expected to be in this interval.

In a forthcoming report, uniform and beta distributions are assumed for the unobservable predictor variable X. The use of nonnormal distributional assumptions for X enables one to obtain estimates of the variance ratio without any knowledge of the error variances. The estimates obtained for the variance ratio under the uniform and beta assumptions are .501 and 1.146, respectively. These values of λ correspond to slope estimates of 1.206 and 1.146. The similarity of these estimates and those in Table 6 suggests that knowledge of the exact value of the variance ratio may not be critical for this data set.

The second data set was obtained from a modelling of greenness values using a mixture of Weibull distributions as discussed in Woodward et al. [11]. Random samples of 200 pure pixels were obtained from a single segment of corn and soybean crops, the proportion of corn in each of the 61 samples was calculated, and minimum distance estimates of the corn proportions were obtained. Figure 10 exhibits a scattergram plotted from the raw data in Table 7. Note that large amount of variability in the classifier estimates (x) relative to the true corn proportions (y).

The purpose in examining this data set is to illustrate the behavior of the structural model estimator when the data contains more variability in the predictor variable than in the response variable. Again using inequality (2.5), the structural model slope estimate is bounded by the following values:

$$.100 \le \hat{\beta} \le 1.717.$$

This interval is much wider than the interval for the previous example and suggests that greater uncertainty surrounds the choice of the variance ratio. Table 8 displays estimates for a range of λ values in the interval [0,1]. The dramatic drop in the estimates for very small variance ratios makes choice of an appropriate structural model estimate difficult if the variance ratio is believed to be small, an assumption which is supported by Figure 10. If one again calculates the estimates under uniform and beta assumptions on X, the variance ratio estimates are .271 and .091, respectively, corresponding to structural model slope estimates of .222 and .909.

5. Concluding Remarks

Linear structural models acknowledge the presence of measurement error in both the response variable and the predictor variable, thereby allowing a more realistic representation of the relationship between ground observations and satellite estimates than least squares estimation of the parameters of linear regression models. In this paper the potential for application of linear structural models to the estimation of agricultural and vegetation characteristics is investigated. Assuming normal probability distributions for the unobservable predictor variable and the two measurement errors necessitates some knowledge of the error variances in order to estimate the model parameters. The focus of this study is on the assumption that the ratio of error variances is known.

The asymptotic properties presented in Section 2 demonstrate that when an incorrect variance ratio is used the structural model estimator is biased and has a larger variance than the corresponding estimator which uses the correct variance ratio. The simulations in Section 3 show that samples as small as 50 or so allow acceptable estimation of the slope parameter when the variance ratio is known. Samples as large as 200 or more are necessary for asymptotic variance formulae to provide good measures of the variability of the estimator. Likewise, assumed values of the error variance ratio in a narrow interval around the true value are necessary both for accurate estimation of the slope parameter and for acceptable estimation of the estimator variability.

The structural model estimator was applied to two data sets on crop proportion estimation in Section 4. In one of the data sets the precise selection of the variance ratio was not found to be critical to the obtaining of suitable parameter estimates because the structural model estimator changed relatively little over a wide range of values of the variance ratio. In the second example choice of the variance ratio the estimator more, leaving greater uncertainty surrounding the appropriate value to use.

Many opportunities exist for improving the application of structural model estimators to the estimation of agricultural and vegetation characteristics. When the three predictor variable X is normally distributed estimation of the slope parameter can be accomplished with replication or with the use of instrumental variables. The application of structural estimation in these two situations will be detailed in future reports.

Theoretically the estimation problems described in this paper are not encountered if the unobservable predictor variable is nonnormally distributed. Estimates of the variance ratio for both of the examples in Section 4 were obtained under assumptions of uniform and beta distributions on X. Nonnormal assumptions for X present challenging theoretical and computational problems which will also be documented in a future report.

REFERENCES

[1] Anderson, T. W., Estimation of linear functional relationships: Approximate distributions and connections with simultaneous equations in econometrics, J. Royal Statist. Soc. Ser. B 38 (1976) 1-36.

[2] Badhwar, G. C., Crop emergence date determination from spectral data, Photogrammetric Engin. and Remote Sensing 46 (1980), 369-377.

[3] Badhwar, G. D., Carnes, J. G., and Austin, W. W., Corn-soybean feature extraction and classification using Landsat data, Lockheed Engin. and Mgt. Serv. Co., Inc., Houston (1981).

[4] Kendall, M. G., and Stuart, A., The Advanced Theory of Statistics, (Macmillan Co., New York, 1977).

[5] Lakshminarayanan, M. Y. and Gunst, R. F., Estimation of parameters in linear structural relationships: Sensitivity to the choice of the ratio of error variances, Biometrika 71 (1984) to appear.

[6] Lindley, D. V. and El-Sayyad, G. M., The Bayesian estimation of a linear functional relationship, J. Royal Statist. Soc. Ser. B 30 (1968) 190-202.

[7] Lwin, T. and Maritz, J. S., An analysis of the linear calibration problem from the perspective of compound estimation, Technometrics 24 (1982) 235-242.

[8] Reiersol, O., Identifiability of a linear relation between variables which are subject to error, Econometrica 18 (1950) 375-389.

[9] Richardson, D. H. and Wu, De-Min, Least squares and grouping method estimators in the errors in variables model, J. Amer. Statist. Assn. 65 (1970) 724-748.

[10] Serfling, R. J., Approximation Theorems of Mathematical Statistics (John Wiley, New York 1980).

[11] Woodward, W. A., Gunst, R. F., Lindsey, H., and Gray, H. L., Proportion estimation in mixtures of asymmetric distributions, NASA Technical Report SR-63-04409 (1983).

Table 1. Ratio of Simulated and Asymptotic Expectations of Structural Model Slope Estimators

	Assumed λ*					
True λ	0.2	0.5	1.0	2.0	4.0	10.0
(a) n = 20						
0.2	1.16	1.04	0.99	0.89	0.77	0.62
1.0	1.21	1.13	0.92	0.99	0.81	0.66
2.0	1.40	1.29	1.15	1.09	0.90	0.69
10.0	2.49	2.08	1.93	2.30	2.26	1.10
(b) n = 50						
0.2	1.02	0.99	0.94	0.86	0.74	0.62
1.0	1.11	1.09	1.02	0.93	0.81	0.64
2.0	1.23	1.19	1.14	1.02	0.87	0.67
10.0	2.27	2.19	2.13	1.98	1.62	1.04
(c) n = 100						
0.2	1.01	0.98	0.93	0.85	0.74	0.61
1.0	1.10	1.07	1.01	0.92	0.79	0.64
2.0	1.22	1.17	1.12	1.02	0.87	0.67
10.0	2.16	2.12	2.03	1.85	1.56	1.03

Table 2. Ratio of Simulated and Asymptotic Mean Squared Errors of Structural Model Slope Estimators

	Assumed λ*					
True λ	0.2	0.5	1.0	2.0	4.0	10.0
(a) n = 20						
0.2	108.11	2.69	1.94	1.31	2.48	3.23
1.0	9.91	3.32	550.84	1.99	1.66	2.68
2.0	35.78	10.48	80.04	3.42	2.38	2.24
10.0	4553.13	996.18	3087.32	357.29	1563.05	25.27
(b) n = 50						
0.2	1.27	1.15	1.20	1.86	3.81	7.40
1.0	2.21	1.92	1.53	1.25	2.25	5.70
2.0	3.79	3.08	2.35	1.29	1.44	4.18
10.0	36.15	30.49	30.84	26.79	18.65	1.69
(c) n = 100						
0.2	1.19	0.99	1.37	3.07	6.89	14.69
1.0	2.27	1.63	1.15	1.44	4.28	11.00
2.0	5.07	3.69	2.29	1.15	2.06	7.98
10.0	48.73	46.22	38.95	27.20	12.54	1.29

Table 3. Ratio of Simulated and Asymptotic Mean Squared Errors of Structural Model Slope Estimator: n = 200, γ = 0.1

	Assumed λ*					
True λ	0.2	0.5	1.0	2.0	4.0	10 .0
			(a) β = 0.1			
0.2	1.10	1.23	1.44	1.38	1.57	1.62
1.0	9.31	1.40	1.10	0.97	1.08	1.06
2.0	118.04	3.44	1.38	0.98	0.97	0.94
10.0	3046167.58	2823896.24	6151.68	3.85	1.55	0.98
			(b) β = 1.0			
0.2	1.08	1.65	3.49	6.35	9.13	12.11
1.0	5.45	2.30	1.06	1.88	4.13	6.45
2.0	16.43	7.97	2.90	1.07	1.98	3.87
10.0	137.29	95.94	53.21	18.68	4.15	1.06
			(c) β = 10.0			
0.2	1.03	1.06	1.03	1.04	1.07	1.03
1.0	1.06	1.03	1.03	1.04	1.06	1.15
2.0	1.07	1.07	1.09	1.02	1.07	1.15
10.0	1.26	1.22	1.23	1.22	1.16	1.10

Table 4. Ratio of Simulated and Asymptotic Mean Squared Errors of Structural Model Slope Estimator: n = 200, γ = 1.0

	Assumed λ*					
True λ	0.2	0.5	1.0	2.0	4.0	10.0
			(a) β = 0.1			
0.2	1.08	0.99	1.18	1.33	1.40	1.42
1.0	1.57	31592.99	1.08	0.56	0.51	0.50
2.0	11789245.73	238241.92	57182.60	1.10	0.51	0.39
10.0	38414594.50	4864779.56	2293212.56	703991.66	89744.73	1.12
			(b) β = 1.0			
0.2	1.03	5.62	15.38	24.16	29.89	33.34
1.0	39.90	11.69	1.17	5.10	10.67	14.54
2.0	128.86	72.34	19.83	1.19	3.97	7.46
10.0	12090.97	11811.55	31704.10	837.30	227.65	1.12
			(c) β = 10.0			
0.2	1.01	1.09	1.05	1.02	1.20	2.28
1.0	1.13	1.13	1.10	1.06	1.17	2.21
2.0	1.25	1.21	1.07	1.11	1.15	1.91
10.0	2.67	2.83	2.76	2.48	1.91	1.11

Table 5. Badhwar Estimates of Corn Proportions

Ground Truth	Satellite Estimate	Ground Truth	Satellite Estimate
.312	.314	.426	.546
.494	.479	.323	.372
.311	.357	.392	.387
.241	.240	.324	.353
.393	.200	.524	.529
.162	.181	.270	.409
.129	.220	.349	.323
.087	.215	.492	.427
.056	.154	.294	.278
.070	.142	.282	.329
.172	.263	.452	.413
.083	.173	.331	.381
.066	.068	.446	.430
.244	.308	.435	.398
.485	.531	.429	.375
.277	.353	.296	.352
.256	.308	.468	.397
.532	.458	.482	.480
.506	.579	.537	.573
.431	.306	.485	.580
.217	.215		

Table 6. Structural Model Slope Estimates: Badhwar Data

Variance Ratio	Slope Estimate	Variance Ratio	Slope Estimate
0.0	1.291	1.0	1.156
0.1	1.270	2.0	1.104
0.2	1.251	3.0	1.077
0.3	1.234	4.0	1.062
0.4	1.219	5.0	1.051
0.5	1.206	6.0	1.044
0.6	1.194	7.0	1.038
0.7	1.183	8.0	1.033
0.8	1.173	9.0	1.030
0.9	1.165	10.0	1.028
		∞	1.001

Table 7. Weibull MD Estimates of Corn Proportions

Ground Truth	Satellite Estimate	Ground Truth	Satellite Estimate	Ground Truth	Satellite Estimate
.42	.38	.48	.32	.45	.47
.45	.63	.45	.50	.50	.53
.44	.52	.47	.47	.43	.63
.49	.61	.47	.47	.48	.57
.39	.44	.42	.46	.43	.74
.42	.61	.49	.58	.53	.63
.49	.53	.43	.46	.46	.60
.47	.58	.42	.51	.46	.48
.50	.50	.51	.67	.52	.56
.50	.58	.46	.47	.43	.44
.48	.60	.48	.61	.46	.50
.44	.56	.46	.59	.49	.67
.45	.59	.51	.55	.43	.54
.48	.46	.49	.69	.47	.39
.50	.50	.49	.66	.44	.65
.48	.45	.40	.56	.45	.60
.42	.57	.51	.63	.46	.60
.46	.60	.52	.58	.43	.56
.40	.64	.44	.40	.46	.56
.48	.59	.56	.69	.47	.60
.40	.55				

Table 8. Structural Model Slope Estimates: Weibull MD Estimates

Variance Ratio	Slope Estimate	Variance Ratio	Slope Estimate
0.0	1.717	0.6	.136
0.1	.836	0.7	.129
0.2	.327	0.8	.125
0.3	.202	0.9	.121
0.4	.163	1.0	.119
0.5	.146	∞	.100

Fig 1: Relative Rate of Change of Structural Model Estimator
Variance Ratio Constant

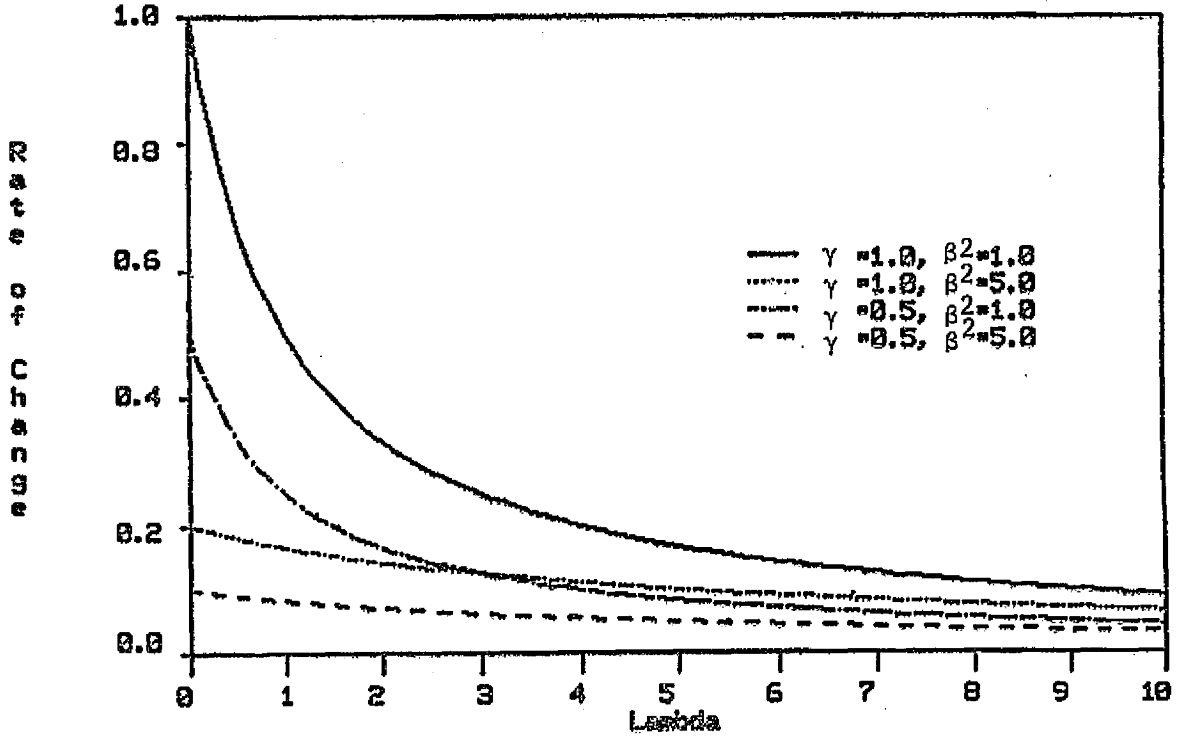


Fig 2: Relative Rate of Change of Structural Model Estimator
Uniform Prior for Variance Ratio

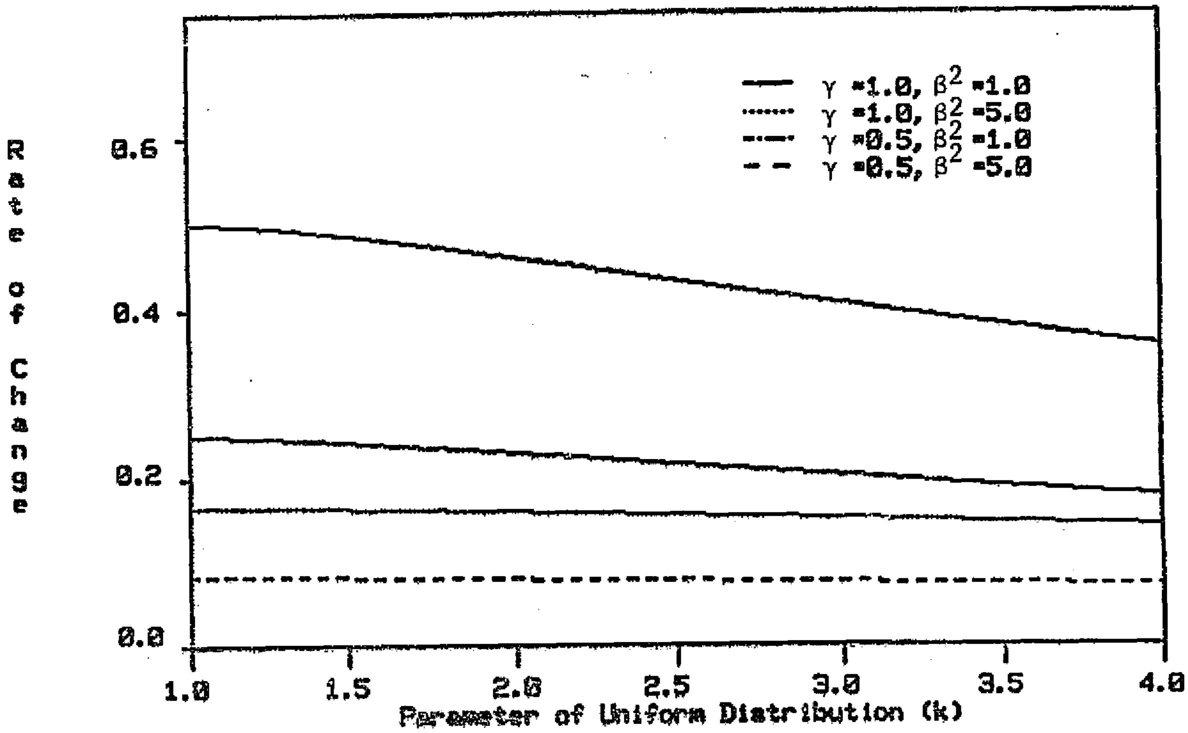


Fig 3: Relative Rate of Change of Structural Model Estimator Normal Prior for Variance Ratio

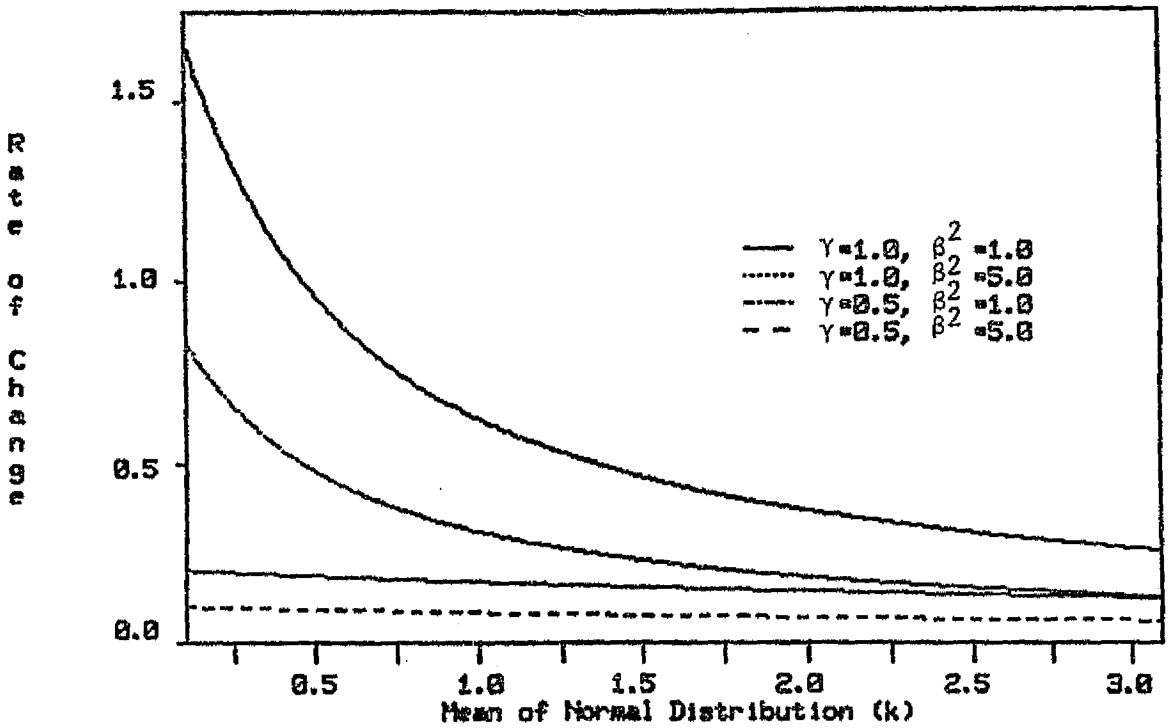


Fig 4: Relative Rate of Change of Structural Model Estimator Chisquare Prior for Variance Ratio

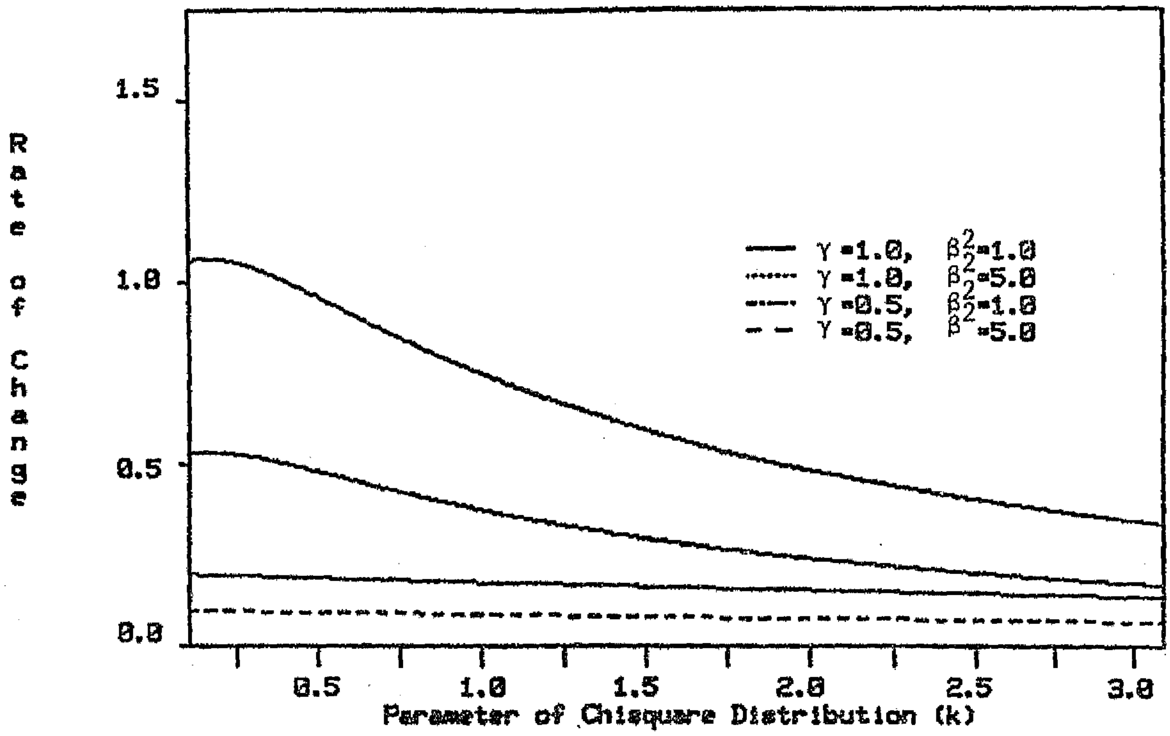


Fig 5: Ratio of Asymptotic Mean Squared Error of Structural and Least Squares Estimators

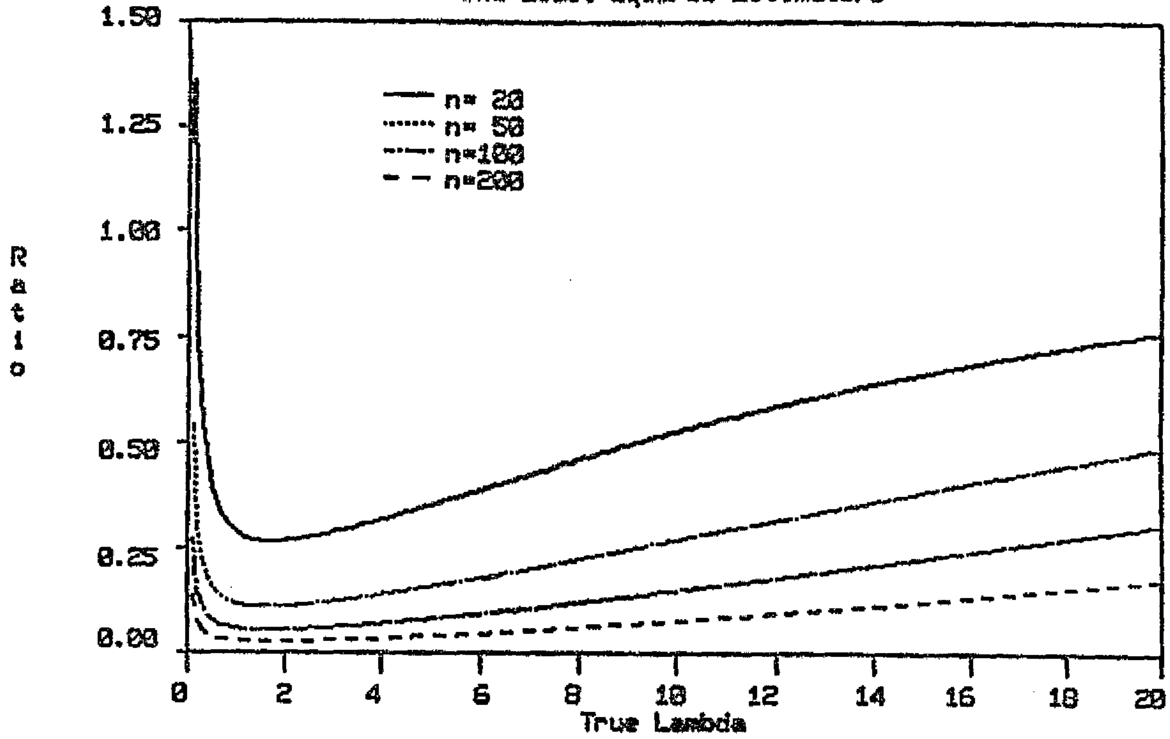


Fig 6: Ratio of Asymptotic Mean Squared Error of Structural and Least Squares Estimators, True Lambda=1.0

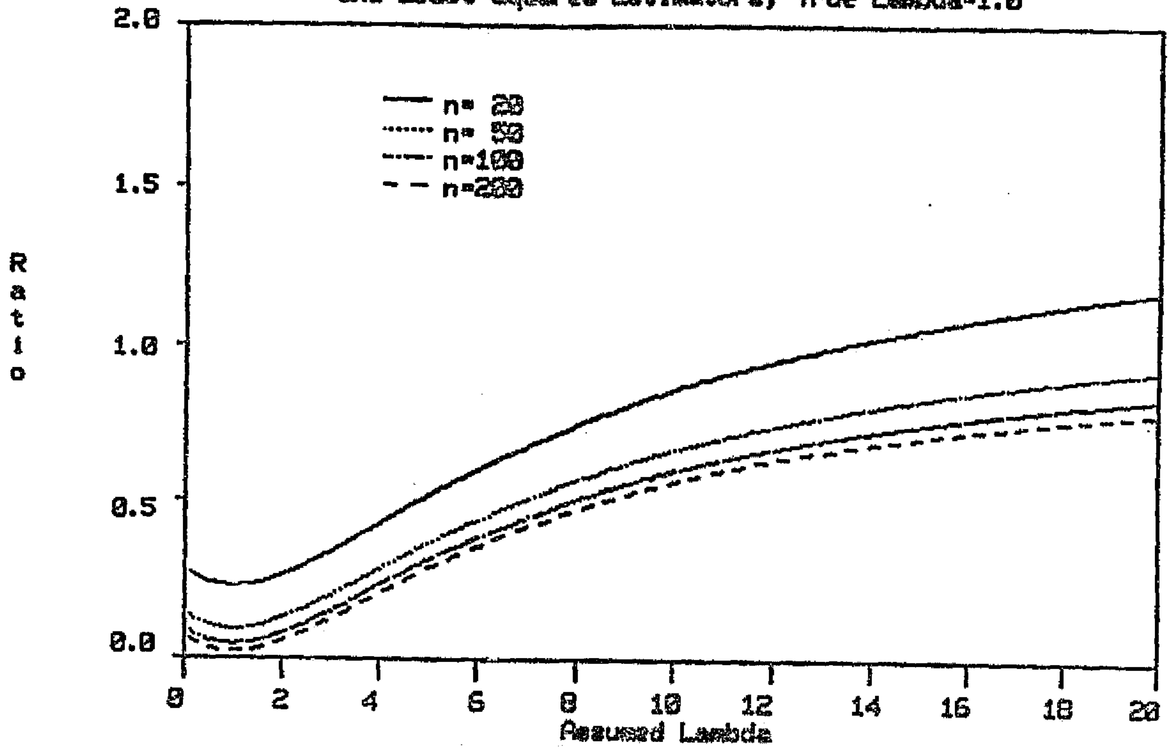


Fig 7: Ratio of Asymptotic Mean Squared Error of Structural and Least Squares Estimators, True Lambda = 6.0

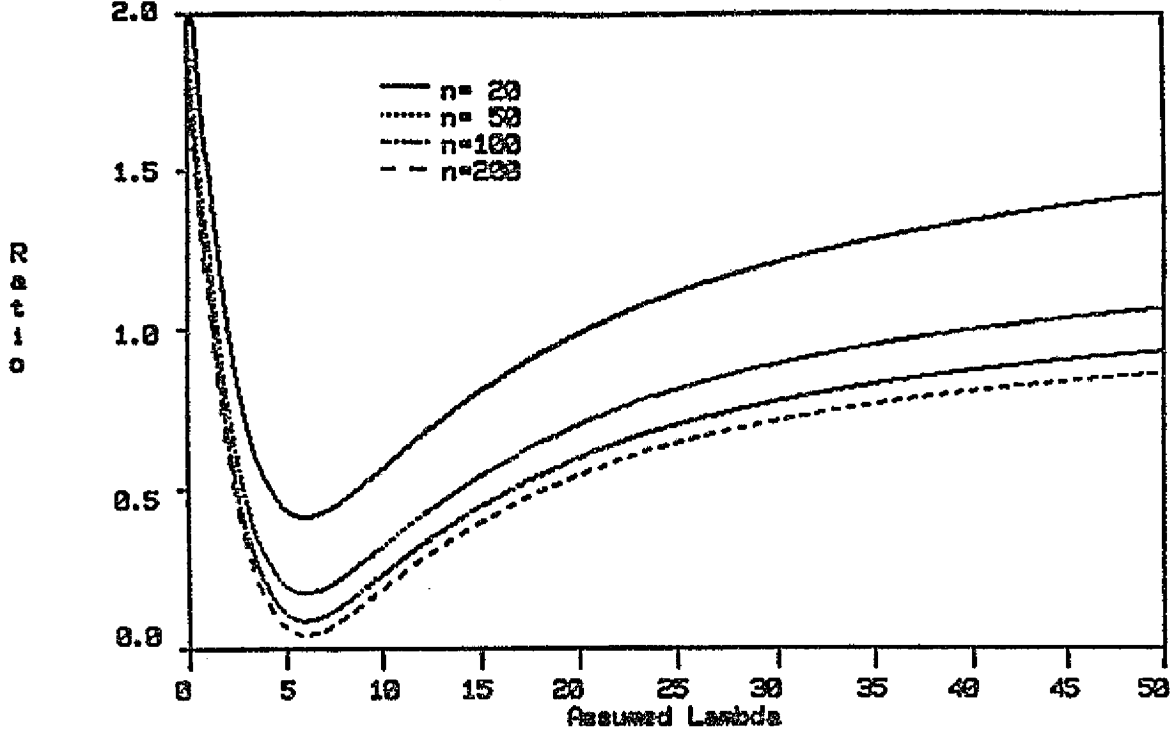


Fig 8: Ratio of Asymptotic Mean Squared Error of Structural and Least Squares Estimators, True Lambda=10.0

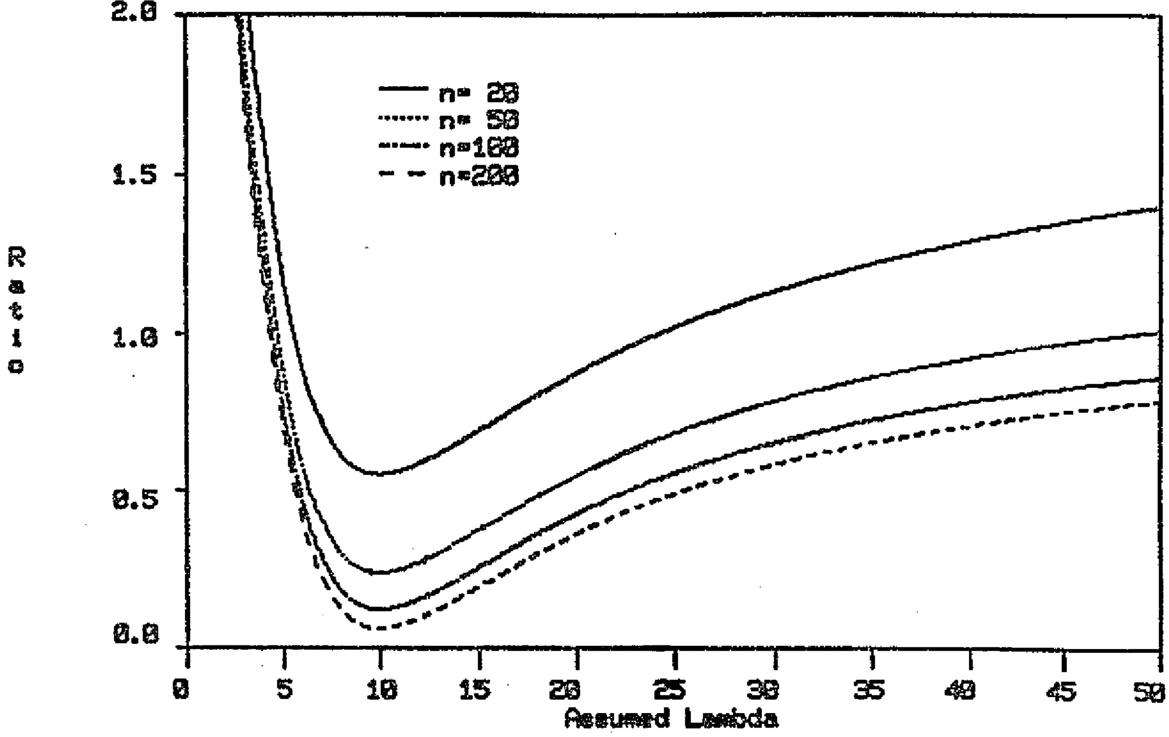


Fig 9: Scatterplot of Corn Proportions: Ground Truth versus Badhwar Estimates

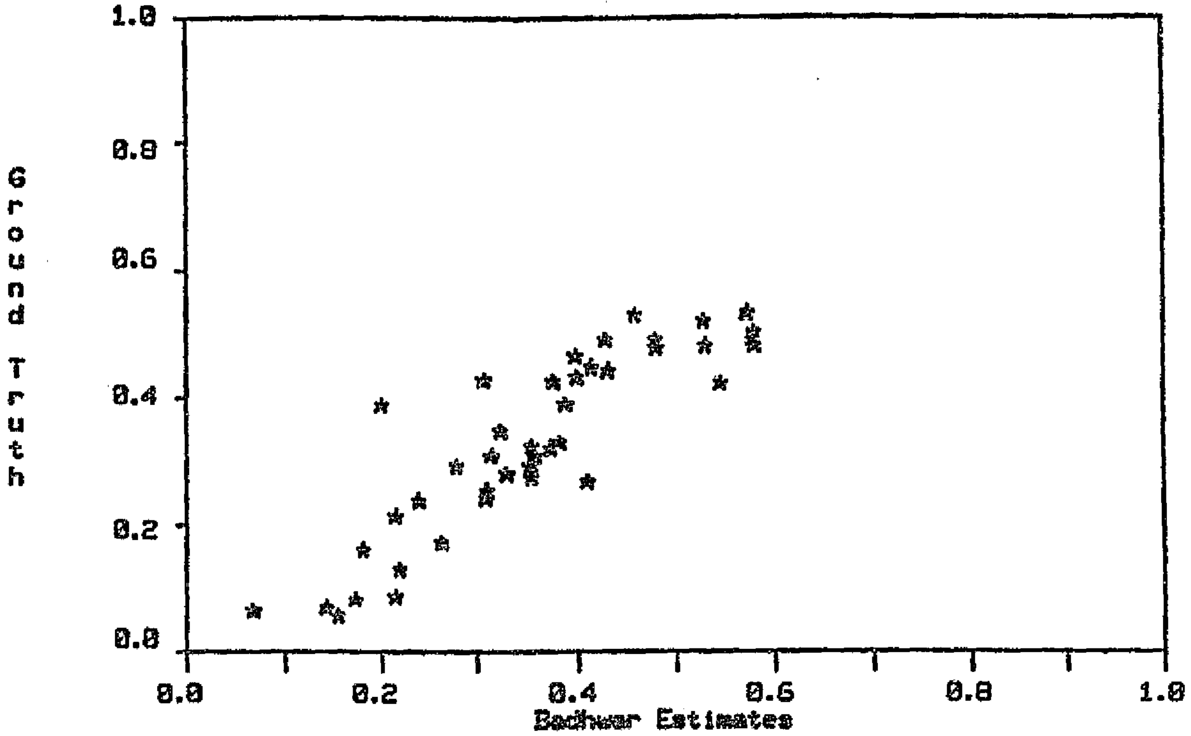
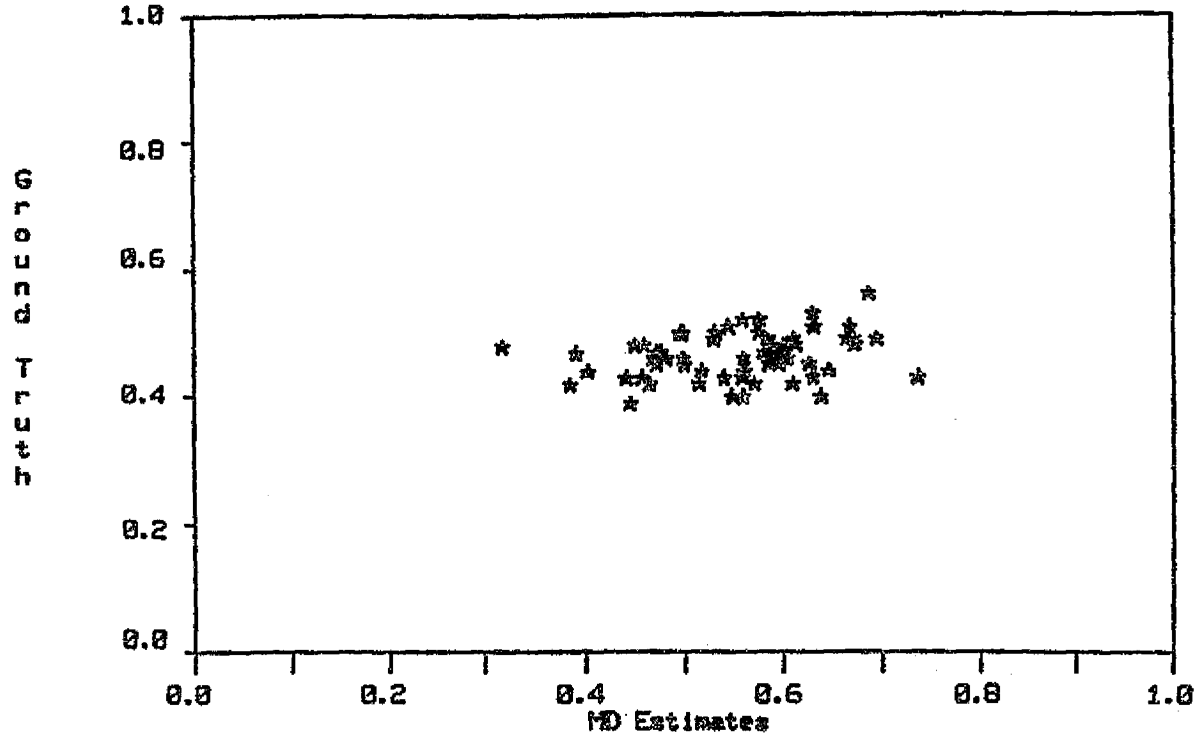


Fig 10: Scatterplot of Corn Proportions: Ground Truth versus Weibull MD Estimates



D9

CALIBRATION OR INVERSE REGRESSION: WHICH IS APPROPRIATE FOR CROP SURVEYS USING LANDSAT DATA?

R. S. Chhikara*
Lockheed Engineering and Management Services Company, Inc.
Houston, Texas

and

A. G. Houston
National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas

*Current Address: Department of Mathematical Sciences
University of Houston - Clear Lake
2700 Bay Area Blvd.
Houston, Texas 77058

ABSTRACT

Calibration and inverse regression estimators of crop proportions are investigated where the auxiliary variable is obtained from binary classification of multivariate Landsat data. We argue that the appropriate model relating classifier proportions and ground observed proportions for a given crop type is the calibration model. We then show, however, that under this model the inverse regression estimator is superior to the calibration estimator in estimating the crop acreage or proportion for a region of interest.

1. INTRODUCTION

The Statistical Reporting Service of the United States Department of Agriculture (USDA) applies probability sample survey methodology to obtain crop acreage estimates. Each year, a survey, known as the June Enumerative Survey (JES), is conducted in the United States to collect land use and crop acreage data. These data are collected for randomly selected area segments. The sampling error at the national level is believed to be about 2 percent. At the state and lower levels, the sampling error is considerably larger. Sigman et al. [8] proposed using a regression estimation approach, based on Landsat data in conjunction with the sample survey data, to decrease the sampling error at these lower levels.

Basically, the approach is to acquire Landsat data over a stratum, called an analysis district, containing a number of JES sample segments. The Landsat data are classified, using data from the sample segments for training, and selected crop acreage or proportion estimates are obtained for each sample segment in the stratum as well as for the entire stratum. The crop acreages for the sample segments observed in the JES are regressed onto the corresponding estimates obtained from the classification of the Landsat data and the resulting relationship is used to obtain an estimate of crop acreage for the stratum from the classifier estimate for the stratum.

In general, this sample survey problem can be stated as follows. Consider a population made up of N clusters. Assume that each cluster contains a large number, M, of units. (In the above context, a unit is a pixel.) Let C_1 represent the class of interest and suppose a unit either belongs to C_1 or its complement C_0. For cluster i, let Y_i be the proportion of units belonging to C_1. Next, let $\underline{Z}$ be a p x 1 measurement vector observed for each unit in the population.

Suppose n clusters are randomly selected from the N clusters and their units are enumerated and correctly identified with respect to the two classes C_1 and C_0. The actual proportion of units in C_1 is then known for each of the n sampled clusters. The set of observations on $\underline{Z}$ for the units of the sampled clusters is used to obtain a discriminant function and a classification rule. Each unit in the population is then classified based on their observations for the measurement vector $\underline{Z}$. Let X_i denote the proportion of units classified into C_1 for cluster i. The problem is to estimate the population mean

$$\bar{Y} = \sum_{1}^{N} Y_i/N$$

For the n sampled clusters, suppose $y_1, y_2, \cdots, y_n$ are the actual proportions of units in C_1 and $x_1, x_2, \cdots, x_n$ are the corresponding estimates obtained from classification of units. Let

$$\bar{X} = \sum_{1}^{N} X_i/N$$

be the average proportion obtained from classification of units in the N clusters making up the population.

The estimator of $\bar{Y}$ considered in Sigman et al. [8] is the standard regression estimator (Cochran [1]) which is based on the regression of the y_i onto the x_i for the sampled clusters. This is an inverse regression since the Y_i are independent variables, that is, variables that take on values that can be observed but not controlled, and the X_i are dependent variables, variables whose values depend on changes in the independent variables. The actual relationship between the X_i and the Y_i depends upon the overlap between the class distributions of the measurement vector $\underline{Z}$ for C_1 and C_0, and the classification procedure.

Since the X_i are dependent on the independent variables Y_i, another estimator of $\bar{Y}$ can be obtained using calibration, which is based on the direct regression of the x_i on the y_i for the sampled clusters. In 1967, Krutchkoff [5] advocated the use of inverse regression based on the results of extensive simulation studies. Since then, the controversy over the properties and hence the utility of these two estimators have been extensively discussed in the literature (Lwin and Maritz [6]). However, the studies in the literature assume a random sample (x_i, y_i), $i = 1, \cdots, n$, from an infinite population with the goal of estimating an individual y_i for a given value x_i. Moreover, none of the previous studies address the problem of classification. Hence, the present study is novel.

As in previous studies, assume that the following linear model holds true:

$$X_i = \gamma + \delta Y_i + \eta_i \tag{1.1}$$

where γ and δ are unknown parameters and the η_i are random errors (independent of the Y_i) having mean zero and finite variance. If the η_i are normally distributed, Shukla [7] has shown that the calibration estimator of Y_i for a given X_i is constant but has infinite variance, whereas the inverse regression estimator is biased but has finite variance. Without assuming normal distribution for the η_i, Lwin and Maritz [6] have shown that the inverse regression estimator has lower mean squared error than the calibration estimator in estimating Y_i for a given X_i, if the Y_i lies in the range of the sample $y_1, \cdots, y_n$. This is very likely to be the case in the present study where an estimate of $\bar{Y}$ is desired, since $y_1, \cdots, y_n$ are a random sample from the finite population whose mean is $\bar{Y}$.

The calibration and inverse regression estimators of $\bar{Y}$ are described in section 2. In section 3, we discuss the classification of units based on the measurement vector $\underline{Z}$ and investigate the two models relating the Y_i and the X_i. It is shown that the calibration model given by (1.1) is linear when the X_i are obtained using the maximum likelihood classification rule, but not the reverse model. A simulation study was conducted to compare the calibration and inverse regression estimators of $\bar{Y}$. A description of the simulation study and the results are presented in section 4. The results are summarized in section 5.

2. THE ESTIMATORS

For measurement units, define the random variable

$$\eta(\underline{Z}) = \begin{cases} 1, & u \in C_1 \\ 0, & u \in C_0 \end{cases} \tag{2.1}$$

where $\underline{Z}$ is the observation vector of unit u. Suppose units in the population are stratified using a classification rule and C_1' and C_0' are the strata corresponding to classes C_1 and C_0, respectively. Define another random variable

$$\psi(\underline{Z}) = \begin{cases} 1, & u \in C_1' \\ 0, & u \in C_0' \end{cases} . \tag{2.2}$$

The pair of random variables $(\eta(\underline{Z}), \psi(\underline{Z}))$ characterizes the two-way classification of units - actual vs. classified. If $\eta(\underline{Z})=\psi(\underline{Z})$ for all units, the classification is perfect; otherwise, it is fallible.

Assume that the cluster size is large and for a cluster, $Y=E[\eta(\underline{Z})]$ and $X=E[\psi(\underline{Z})]$. Considering the relative frequencies of units in C_1 and C_1' approximated by the expected values of the random variables $\eta(\underline{Z})$ and $\psi(\underline{Z})$, one can write Y and X as probabilities,

$$Y = P[\eta(\underline{Z}) = 1] \qquad (2.3)$$
$$X = P[\psi(\underline{Z}) = 1].$$

Similarly, the two classification error rates are approximated by the conditional probabilities,

$$\theta_0 = P[\psi(\underline{Z})=0 \mid \eta(\underline{Z})=1] \qquad (2.4)$$
$$\theta_1 = P[\psi(\underline{Z})=1 \mid \eta(\underline{Z})=0]$$

where θ_0 is called the omission error and θ_1 is called the commission error. From (2.3) and (2.4), it follows that

$$X = \theta_1 + (1-\theta_0-\theta_1)Y. \qquad (2.5)$$

Thus, for a cluster, X is a linear function of Y. Upon considering the variability in θ_1 and θ_0 across clusters, suppose the regression function of X onto Y is of linear form, say

$$E[X|Y=y] = \gamma + \delta y. \qquad (2.6)$$

Suppose n clusters are randomly selected and the (x_i, y_i), $i=1,2,\cdots,n$, are the pairs of observations for the random variables X and Y. Then by regressing x_i on y_i, a regression estimator of the population mean $\bar{Y} = \sum_1^N Y_i/N$ is given by

$$\hat{Y}_C = \bar{y} + (\bar{X} - \bar{x})/\hat{\delta} \tag{2.7}$$

where

$$\bar{X} = \sum_1^N X_i/N,$$

$$\hat{\delta} = \sum_1^N (x_i - \bar{x})(y_i - \bar{y}) / \sum_1^n (y_i - \bar{y})^2$$

$$\bar{x} = \sum_1^n x_i/n, \quad \bar{y} = \sum_1^n y_i/n \ .$$

In (2.7), $\bar{X}$ is assumed to be known and the subscript C stands for calibration. This estimator will be called the calibration estimator.

On the other hand, Y can be written in terms of X for a cluster by inverting (2.5) as follows:

$$\begin{aligned} Y &= -\theta_1/(1-\theta_0-\theta_1) + X/(1-\theta_0-\theta_1) \\ &= \phi_0 + \phi_1 X \end{aligned} \tag{2.8}$$

where

$$\begin{aligned} \phi_0 &= -\theta_1/(1-\theta_0-\theta_1) \\ \phi_1 &= 1/(1-\theta_0-\theta_1). \end{aligned} \tag{2.9}$$

Again ϕ_0 and ϕ_1 would vary across clusters and thus one may consider the inverse regression function for Y on X given by

$$E[Y|X=x] = \alpha + \beta x. \tag{2.10}$$

Then by regressing y_i on x_i, a regression estimator of $\bar{Y}$ is given by

$$\hat{Y}_{IR} = \bar{y} + \hat{\beta}(\bar{X}-\bar{x}) \tag{2.11}$$

where

$$\hat{\beta} = \sum_{1}^{n} (x_i-\bar{x})(y_i-\bar{y}) / {}_{1}^{n} (x_i-\bar{x})^2 \ .$$

The estimator in (2.11) will be called the <u>inverse regression estimator</u>.

In the next section, we discuss the cluster proportion X resulting from the classification of units based on their observed data on $\underline{Z}$ and investigate the two regression models corresponding to (2.6) and (2.10).

3. REGRESSION MODELS

3.1 Determination of X

Suppose that $\underline{Z}$ is a p x 1 random vector distributed normally with mean vectors $\underline{\mu}_1$ and $\underline{\mu}_0$ for classes C_1 and C_0, respectively, and common covariance matrix Σ. By a set of linear transformations, the class structures can be expressed in the canonical form:

$$\underset{\sim}{Z} \sim \begin{cases} N(-((\Delta/2)\underset{\sim}{e}, I) \\ N((\Delta/2)\underset{\sim}{e}, I) \end{cases} \tag{3.1}$$

where

$$\begin{aligned} \Delta^2 &= (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_0)'\Sigma^{-1}(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_0) \\ \underset{\sim}{e} &= (1, 0, \cdots, 0)'. \end{aligned}$$

When Δ is known, the discriminant function based on the log likelihood ratio is linear and the maximum likelihood classification rule is to classify a measurement unit in C_1 if $Z_1 < 0$, and in C_0 otherwise, where Z_1 is the first component of measurement vector $\underset{\sim}{Z}$. In terms of random variable ψ defined in (2.2), we have

$$\psi(\underset{\sim}{Z}) = \begin{cases} 1, & \text{if } Z_1 < 0 \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

Then the omission and commission error rates for C_1 are each equal to $\Phi(-\Delta/2)$, where Φ denotes the cdf for the standard normal distribution.

In the present context, one needs to evaluate the classification error rates for individual clusters. Because the cluster size is assumed large, the class distributions for each cluster can be approximated by the normal. For cluster i, let $\xi_i\underset{\sim}{e}$ and $(\xi_i + \Delta_i)\underset{\sim}{e}$ be the mean vectors of C_1 and C_0, respectively, and I is the common covariance matrix. This distributional assumption allows variation in class distributions across clusters in the population. To have the population means as assumed in (3.1), we assume that the average value of ξ_i across clusters is $-\Delta/2$ and that of $(\xi_i + \Delta_i)$ is $\Delta/2$. The actual classification error rates θ_{0i} and θ_{1i} and the classified proportion X_i for cluster i, denoted by B_i, are easily obtained as follows:

$$\begin{aligned} \theta_{0i} &= P[Z_1 \geq 0 | \eta(\underline{Z})=1,\ B_i] \\ &= \Phi(-\Delta/2 + \xi_i) \\ \theta_{1i} &= P[Z_1 < 0 | \eta(\underline{Z})=0,\ B_i] \qquad (3.3) \\ &= \Phi(\Delta/2 - \xi_i - \Delta_i) \end{aligned}$$

and

$$X_i = \Phi(\Delta/2 - \xi_i - \Delta_i) + [\Phi(\Delta/2 - \xi_i) - \Phi(\Delta/2 - \xi_i - \Delta_i)]Y_i$$

The sample analogue of the linear discriminant function is obtained by replacing the parameters by their estimators and is given by

$$\lambda(\underline{Z}) = [\underline{Z} - (1/2)(\bar{\underline{Z}}_1 + \bar{\underline{Z}}_0)]'\underline{S}^{-1}(\bar{\underline{Z}}_1 - \bar{\underline{Z}}_0)$$

where $\bar{\underline{Z}}_1$ and $\bar{\underline{Z}}_0$ are the sample mean vectors for C_1 and C_0, respectively, and $\underline{S}$ is the common sample covariance matrix obtained from the training samples $\underline{Z}_{1j}$ and $\underline{Z}_{0j}$ of C_1 and C_0, respectively. Suppose a set of clusters are randomly selected and the training samples consist of all the units in the sampled clusters. Because the training sample size is large, the statistics $\bar{\underline{Z}}_1$, $\bar{\underline{Z}}_0$ and $\underline{S}$ are approximately equal to the class parameters, therefore the classifier and the error rates discussed above will approximately hold true.

In practice, the parameters ξ_i, Δ_i, and Y_i are unknown except for the sampled clusters, and hence the error rates and X_i cannot be obtained for all clusters using (3.3). However, once the classification of data is completed using a classifier, X_i can be computed directly as the proportion of units from B_i that belong to C_1'. Accordingly, the values of the two estimators of $\bar{Y}$ described in section 2 can be computed.

3.2 CALIBRATION MODEL

When Δ is known, one can write from (2.5) for cluster i that

(3.4) $$X_i = \gamma_i + \delta_i Y_i$$

where

$$\gamma_i = \Theta_{1i} \text{ and } \delta_i = 1-\Theta_{0i}-\Theta_{1i} .$$

with Θ_{1i} and Θ_{0i} given in (3.3).

Thus, the regression function in (2.6) can be associated with the model

(3.5) $$X_i = \gamma + \delta Y_i + e_i$$

where

(3.6) $$e_i = (\gamma_i - \gamma) + (\delta_i - \delta)Y_i \ .$$

Since the classification rule (3.2) is independent of the Y_i for known Δ, γ_i and δ_i are independent of Y_i. Let

(3.7) $$\gamma = E[\gamma_i]$$
$$\delta = E[\delta_i].$$

Then

$$E[e_i|Y_i] = 0$$

so that the regression function in (2.6) holds and the model given in (3.5) is linear.

In the case of a finite population of N clusters, one may consider

$$\gamma = \bar{\gamma} = \sum_1^N \gamma_i/N$$

and

$$\delta = \bar{\delta} = \sum_1^N \delta_i/N$$

Now, let us consider the more likely case for which parameters are unknown. These parameters are estimated from the training samples resulting in the classification rule given by (3.2) with 0 replaced by the estimated boundary value. Then, corresponding to (3.3), we have

$$\hat{X}_i = \hat{\theta}_{1i} + (1 - \hat{\theta}_{0i} - \hat{\theta}_{1i})Y_i \ .$$

where θ_{1i} and θ_{0i} denote the estimates of θ_{1i} and θ_{0i}, respectively. Further, let

$$d\theta_{0i} = \hat{\theta}_{0i} - \theta_{0i}$$

$$d\theta_{1i} = \hat{\theta}_{1i} - \theta_{1i}$$

$$\begin{aligned} dX_i &= \hat{X}_i - X_i \\ &= d\theta_{1i} - (d\theta_{0i} + d\theta_{1i})Y_i \ . \end{aligned}$$

Again, the deviations $d\theta_{0i}$, $d\theta_{1i}$ and dX_i are independent of Y_i.

Since $\hat{X}_i = X_i + dX_i$, it follows from (3.5) that we now have the model

(3.8) $$\hat{X}_i = \gamma + \delta Y_i + \eta_i$$

where

(3.9) $$\eta_i = e_i + dX_i$$

with e_i given by (3.6).

The distributions of the deviations $d\Theta_{0i}$ and $d\Theta_{1i}$ are quite complicated; however, asymptotically, that is as the training sample size becomes large, their means go to zero (Efron [2]). Thus, as the training sample size gets large, $E(dX_i|Y_i)$ goes to zero. Hence, in the case of large training samples,

$$E(\eta_i|Y_i) = E(e_i|Y_i) + E(dX_i|Y_i)$$

goes to zero and the model (3.8) is linear.

Suppose σ_γ^2, σ_δ^2 and $\sigma_{\gamma\delta}$ are the variances and covariance respectively of γ_i and δ_i across clusters. Then the conditional error variance, $V(e_i|Y_i) = \sigma_i^2$, is given by

(3.10) $$\sigma_i^2 = \sigma_\gamma^2 + Y_i^2\sigma_\delta^2 + 2Y_i\sigma_{\gamma\delta}$$

The conditional variance due to the training sample, $V(dX_i|Y_i) = \sigma_d^2$, is of order 1/m, where m is the number of units in the training sample. However, this variance is negligible when the training sample size is large. Of course, the total conditional error variance for model (3.8), given by

$$\begin{aligned} V_i &= V(\eta_i|Y_i) \\ &= \sigma_i^2 + \sigma_d^2, \end{aligned}$$

is non-constant.

3.3 INVERSE REGRESSION MODEL

One can express (3.4) inversely as

$$Y_i = -\frac{\gamma_i}{\delta_i} + \frac{1}{\delta_i} X_i$$

Let $\alpha_i = -\gamma_i/\delta_i$ and $\beta_i = 1/\delta_i$ so that

$$Y_i = \alpha_i + \beta_i X_i.$$

Thus, the regression function in (2.10) can be conceptualized in terms of the model

(3.10) $$Y_i = \alpha + \beta X_i + \varepsilon_i$$

where

(3.11) $$\varepsilon_i = (\alpha_i - \alpha) + (\beta_i - \beta)X_i.$$

In the case of training a classifier, the model becomes

(3.12) $$Y_i = \alpha + \beta\hat{X}_i + \zeta_i$$

where

(3.13) $$\zeta_i = \varepsilon_i - \beta\, dX_i\ .$$

Models (3.10) and (3.12) are not linear. This can be easily seen by showing that the covariance of X_i and ε_i in (3.10) is not necessarily zero. Hence, the regression function in (2.10) cannot be obtained from models (3.10) and (3.12). One would therefore wonder about the basis of the inverse regression estimator of $\hat{Y}$ given in (2.11) in section 2.

In general, Lwin and Maritz [6] showed that the inverse regression estimator is a compound estimator of the sample mean and the calibration estimator obtained assuming that the model (3.8) holds. They assume a constant error variance whereas, in the present situation, the error variance is non-constant. Their formulation does not require the inverse regression model as in (3.10) or (3.12) to be linear; hence the inverse regression estimator of $\hat{Y}$ put forth in section 2.0 can be justified without reference to this assumption as was done by Lwin and Maritz.

Presently, no further analytical investigation of the models and the estimators is undertaken. Instead, a simulation study was conducted to evaluate the models and compare the two estimators. This simulation is described in the next section.

4. SIMULATION STUDY

To investigate the linearity of the calibration and inverse regression models given by (3.8) and (3.12), respectively, and to compare the performance of the two estimators of the population mean, the following simulation study was conducted: A hypothetical population consisting of N clusters was considered. The number of units per cluster is assumed infinite. For a given population mean, $\bar{Y}$, the beta distribution (IMSL subroutine GGBTR) was used to generate the actual proportions, Y_i, of the class of interest for each of the N clusters. For each cluster, the distribution of the auxiliary measurement variable Z_1 for the class of interest was assumed normal with mean $\xi_i = \mu_i - \Delta_i/2$ and variance is 1. For the other class, the distribution was assumed normal with mean $\xi_i + \Delta_i = \mu_i + \Delta_i/2$ and a variance of 1. The normal distribution (IMSL subroutine GGNML) with mean 0 and variance σ^2 was used to generate the μ_i and the triangular distribution (IMSL subroutine GGTRA) over the interval $(\Delta-p, \Delta+p)$, which has mean Δ and range 2p, was used to generate the Δ_i with σ^2 and p specified. To ensure $\Delta_i>0$, it was assumed that $\Delta>p$.

The indices of each of the variables for the N clusters were randomly permutated (IMSL subroutine GGPER) and the first n indices were selected as the sample for which the actual proportions were assumed known.

The discriminant boundary parameter, say τ, was then estimated by

$$(4.1) \qquad \hat{\tau} = \frac{\sum_1^n Y_i(\mu_i-\Delta_i/2)}{\sum_1^n Y_i} + \frac{\sum_1^n (1-Y_i)(\mu_i+\Delta_i/2)}{\sum_1^n (1-Y_i)}$$

and the classification rule for all measurement units u in the population was taken as

$$(4.2) \qquad \psi(\underline{Z}) = \begin{cases} 1, & \text{if } Z_1 < \hat{\tau}/2 \\ 0, & \text{otherwise} \end{cases}$$

for the measurement vector $\underline{Z}$. Hence, a unit is classified in C_1 if $\underline{Z}_1 < \hat{\tau}/2$ and into C_0 otherwise. Note that we did not actually generate the measurement vector $\underline{Z}$. These results correspond to having an infinite number of measurement units for training, since we assumed the number of units per cluster to be infinite. The actual errors of misclassification were computed for each of the N clusters from (3.3) as

$$\hat{\Theta}_{0i} = \text{Prob } [Z_1 \geq \hat{\tau}/2 \mid u\varepsilon C_1, B_i, \hat{\tau}]$$

$$= \Phi(-\frac{\hat{\tau}+\Delta_i}{2} + \mu_i)$$

and

$$\hat{\Theta}_{1i} = \text{Prob } [Z_1 < \hat{\tau}/2 \mid u\varepsilon C_0, B_i, \hat{\tau}]$$

$$= \Phi(\frac{\hat{\tau}-\Delta_i}{2} - \mu_i)$$

Likewise, using the relationships in sections 3.2 and 3.3, $\hat{X}_i$, $\hat{Y}_i$, $\hat{\delta}_i$, $\hat{\alpha}_i$ and $\hat{\beta}_i$ were computed for each of the N clusters.

This process was replicated 500 times for each combination of parameters considered in order to compute the bias, variance, and mean squared error for the calibration and inverse regression estimators of $\bar{Y}$. One hundred replications were made to compute the model errors, η_i and ξ_i, and their means and variances. Table 4-1 shows the values of the parameters used in the simulation.

Figure 4-1(a) shows a histogram of the 500 actual proportions generated from a beta distribution with mean .25. The actual mean and variance of these 500 proportions are .2575 and .02366, respectively. Figure 4-1(b) shows a corresponding histogram of one realization of the classified proportions resulting from using n=10, σ=.1, Δ=1.5, and p=1. A scatterplot of the actual versus the classified proportions for this realization is given in figure 4-1(c). In this case, the relationship

TABLE 4-1.- PARAMETER INPUT VALUES

N	= 500
n	= 4, 10, 30
$\bar{Y}$	= .05, .1, .25, .5
σ	= .01, .10, .50
Δ	= 1.5, 3.0
P	= 0, .5, 1

is approximately linear and a linear regression model should hold reasonably well.

A plot of the model errors, η_i, for the calibration model (3.8), versus the actual proportions, Y_i, for all 500 clusters for one realization generated using $\bar{Y}=.25$, $n=10$, $\sigma=.1$, $\Delta=1.5$ and $p=1$ is given in figure 4-2. Note that no obvious relationship exists between η_i and Y_i supporting the linear model requirement that $E[\eta_i|Y_i] = 0$. Also note that the variance of the errors tends to decrease with increasing values of Y_i, that is, we have a non-constant conditional error variance. Figure 4-3 contains a plot of the inverse regression model errors, ζ_i, versus the classifier proportions, $\hat{X}_i$, for the same parameter values. Note the linear dependence indicating $E[\zeta_i|\hat{X}_i]$ is non-zero and the inverse regression model is not linear. The variance of these errors is also non-constant, tending to increase as the classifier proportion increases.

Table 4-2 contains the means and average variances of the calibration and inverse regression model errors computed from 100 replications based on $\bar{Y}=.25$, $\sigma=.1$, $\Delta=1.5$, $p=1$ and three sample sizes, $n=4$, 10 and 30. The mean error is zero for the calibration model and it is non-zero for the inverse regression model as expected. Moreover, the error variance is much larger in the case of the later model. Notice that the means and average variances are about the same for all three sample sizes. This is due to the fact that the number of units per cluster is infinite for this simulation study. In the USDA problem which motivated this study, the number of units per cluster is large; there are 600 or more pixels per sample segment used for training. Results similar to those in figures 4-2 and 4-3 and table 4-2 were obtained for the other combinations of parameters presented in table 4-1. In summary, the calibration model given by (3.8) was found to be the appropriate linear model relating the classifier and the actual proportions in this simulation.

The coefficients γ, δ in model (3.8) and α, β in model (3.12) were computed directly as averages of γ_i, δ_i and α_i, β_i, respectively, for the 500 segments for each replication. Then their averages were obtained from the 500 replications. Also, the corresponding least square fits were obtained in each case and these coefficients were

TABLE 4-2.- MODEL ERROR STATISTICS BASED ON 100 REPLICATIONS*

		Average for 100 Replications		
		n		
Model	Error statistic	4	10	30
Calibration	Mean	0	0	0
	Variance	22.5×10^{-4}	22.6×10^{-4}	22.6×10^{-4}
Inverse regression	Mean	.0184	.0185	.0185
	Variance	93.7×10^{-4}	94.0×10^{-4}	93.9×10^{-4}

*$\bar{Y} = .25$, $\sigma = .1$, $\Delta = 1.5$, and $p = 1$.

estimated for each of the 500 replications. The results for the average computed (actual) and estimated values are given in table 4-3.

The results in table 4-3 show that the coefficients γ, δ in the calibration model (3.8) are unbiasedly estimated by the least square estimates obtained from regressing $\hat{X}_i$ on Y_i, whereas, the coefficients α, β in the inverse regression model (3.12) are biasedly estimated when regressing Y_i on $\hat{X}_i$. This again shows that the model (3-8) is linear, but model (3.12) is not.

The summary results for the calibration and inverse regression estimators of the population mean $\bar{Y}$ for 500 replications for the parametric case, i.e., $\bar{Y} = .25$, $n = 10$, $\sigma = .10$, $\Delta = 1.5$ and $p = 1$, are presented in table 4-4. The two estimators were truncated at 0 and 1 for estimates outside this range before computing the summary statistics. Truncation was actually needed only for the calibration

TABLE 4-3.- ACTUAL AND ESTIMATED VALUES OF REGRESSION COEFFICIENTS

Model	Coefficient	Actual	Estimate
Calibration	γ	.237	.235
	δ	.528	.534
Inverse Regression	α	-.515	-.261
	β	2.024	1.383

*Averaged from 500 replications.

estimator. The MSE ratio in table 4-4 is the ratio of the mean squared error for an estimator to the mean squared error using the sample mean of the n sampled actual proportions. This, of course, is an estimate of the relative efficiency of the sample mean relative to the estimator.

The bias is negligible for each estimator, though it is statisfied significant in the case of the inverse regression estimator. The MSE ratio of .342 for the inverse regression estimator is much smaller than the calibration estimator, which performs rather poorly as its MSE ratio is greater than 1.

The summary results for various combinations of parameter values showed that the MSE ratio for the inverse regression estimator was always less than or equal to that of the calibration estimator. The class separability and sample size influenced the performance of the estimators the most. The calibration estimator performed very poorly for n=4. The inverse regression estimator tended to have significant, yet negligible, bias for n=10 unless σ=.5 in which case it was not

TABLE 4-4.- SUMMARY STATISTICS FOR 500 REPLICATIONS*

Estimator	Bias	Variance	MSE	MSE Ratio	t-Statistic
Sample Mean	.0020	.0021	.0021	1.000	.97
Inverse Regression	-.0033	.0007	.0007	.342	-2.77
Calibration	.0002	.0023	.0023	1.088	.12

*γ = .25, n = 10, σ = .1, Δ = 1.5, and p = 1.0

necessarily negligible. Note that an increase in σ value leads to an increase in variability of the means for the class of interest across clusters and, hence, an increase in the variability of the classified proportions.

The relationship between the actual and classified proportions is greatly influenced by the class separability parameters Δ , p, and σ. When the mean separability Δ is large and the variability in class separability and their mean locations across clusters (that is, p and σ) is small, the relationship is fairly linear, resulting in a high correlation coefficient between the actual and classified proportions for the clusters in the population. The population correlation coefficient decreases as Δ decreases and/or either of the other two parameters p and σ increases.

In each simulation run, the square of the correlation coefficient was computed for both the population and the sample. Variation in the population correlation coefficient arises due to the decision rule for the maximum likelihood classifier varying from sample to sample for the 500 replications. Three scatterplots of the sample R^2 versus the population R^2 corresponding to three sample sizes are presented in figure 4-4 showing the values obtained for the combinations of parameter values in table 4-2. The scatterplot has a fairly high population R^2 showing very small variation regardless of sample size; whereas, the sample R^2 is highly variable with its variability decreasing as the sample size increases. These scatterplots show the potential hazard in using the sample R^2 as an indicator of the actual linear relationship for the population.

Figures 4-5, 4-6, 4-7, and 4-8 summarize the MSE ratios for the two estimators for all combinations of parameter values given in table 4-1. Figures 4-5 and 4-6 contain scatterplots of the MSE ratio versus the mean population R^2 for $\bar{\gamma}$=.5 and Δ values of 3 and 1.5, respectively. Figures 4-7 and 4-8 show the MSE ratio for $\bar{\gamma}$=.25. The symbol '*' in each scatterplot represents the inverse regression estimator and the symbol '+' is for the calibration estimator. MSE ratios for the calibration estimator were truncated at 1.6 to provide uniform plots.

Figures 4-5 and 4-7 indicate that when the separability is large across clusters and the sample size is large, the two estimators have similar MSE ratios, which significantly improve the efficiency relative to the sample mean. When the sample size is reduced, both estimators are affected. The calibration estimator is significantly degraded yielding MSE ratios much larger than 1 for n=4. The inverse regression estimators performs well overall with a modest decrease in efficiency when n=4.

When the separability between classes is reduced, as is the case in figures 4-6 and 4-8, there is a tremendous effect on the mean population R^2, which is reflected in the performance of the two estimators. For Δ=3, R^2 ranged from .75 to 1.0. Reducing Δ to 1.5, it ranged from .23 to 1.0. Note that the inverse regression estimator is still superior, except when n=4, in which case both estimators perform worse than the sample mean corresponding to low values of R^2.

5. CONCLUDING REMARKS

In the present study the individual random values for $\underline{Z}$ for the within cluster measurement units were not simulated and the current results were obtained by generating the parametric values directly for the clusters. As mentioned previously, this is equivalent to having an infinite number of units per cluster. However, the introduction of the range in class separability (i.e., parameter p) and the variability in the two class means ξ_i and $\xi_i+\Delta_i$, $i=1,2, \cdots, N$, across clusters allowed a realistic simulation of the data structure likely to arise at the cluster level in the context of the problem of crop acreage estimation described in section 1. Further study is underway to generate the random values for $\underline{Z}$ and to investigate the inverse regression and other estimators of crop acreages using results from classification of the individual units within clusters.

Scatterplots of model errors versus actual proportions for the calibration model and model errors versus classifier proportions for the inverse regression model were made for all 500 clusters for various parametric cases. These plots indicated that the assumption of linearity for the calibration model holds true; however, the model error has a non-constant variance when the separability of the two class distributions for the population vary highly across clusters. The inverse regression model is not linear.

Comparisons of the two estimators of the population mean $\bar{Y}$ indicated the inverse regression estimator to be better than the calibration estimator. The calibration estimator is unreliable unless the population R^2 between the actual and classifier proportions is high (.8 or larger). The inverse regression is significantly biased in some cases; however, in each of these cases the bias was negligible. The bias was statistically significant due to the variance of the inverse regression estimator being quite small.

ACKNOWLEDGMENT

Research of R. S. Chhikara was supported by the Statistical Reporting Service of the U.S. Department of Agriculture and NASA/JSC Contract NAS-15800.

REFERENCES

[1] Cochran, W. G., Sampling techniques (John Wiley, New York, 1977).

[2] Efron, B., The efficiency of logistic regression compared to normal discriminant analysis, J. of Amer. Stat. Assn., 70 (1975) 892-898.

[3] Hung, Hsien-Ming, Use of transformed Landsat data in regression estimation of crop acreages, Ph.D. Thesis, Iowa State University (1983).

[4] IMSL Library, Edition 9, Houston, (International Mathematical and Statistical Libraries, Inc., 1982)

[5] Krutchkoff, R. G., Classical and inverse regression methods of calibration, Technometrics, 9 (1967). 425-440.

[6] Lwin, T. and Maritz, J. S., An analysis of the linear-calibration controversy from the perspective of compound estimation, Technometrics 24 (1982). 235-242.

[7] Shukla, G. K., On the problem of calibration, Technometrics, 14 (1972). 547-553.

[8] Sigman, R., Hanuschak, G., Craig, M., Cook, P., and Cardenas, M., The use of regression estimation with Landsat and probability ground sample data, Amer. Stat. Assn. Proceedings of the Section on Survey Research Methods, (1978) 165-168.

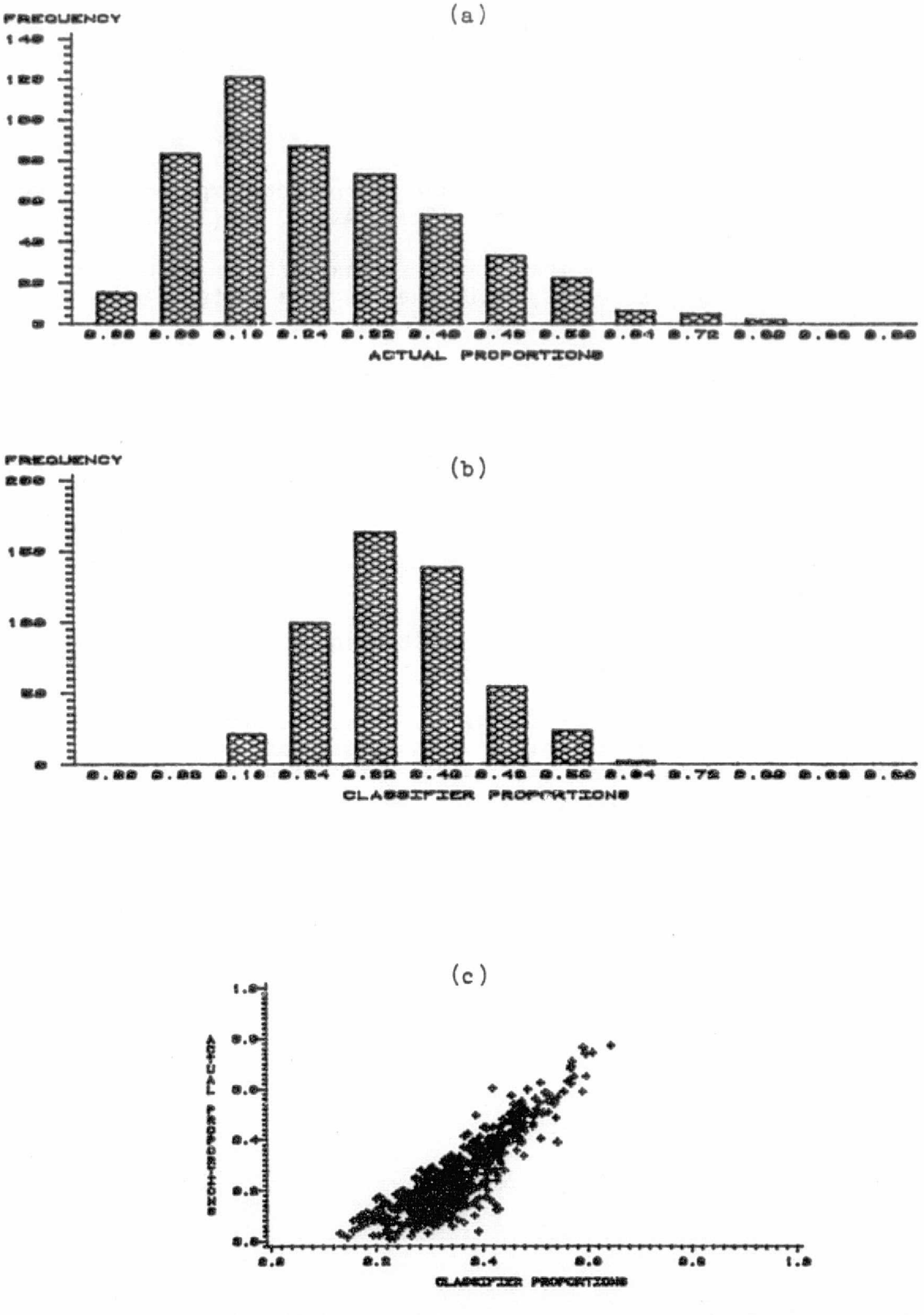


Figure 4-1.- Histograms and scatterplot of class proportions when $\bar{\gamma}$ = .25, σ = .1, Δ = 1.5, p = 1, and n = 10.

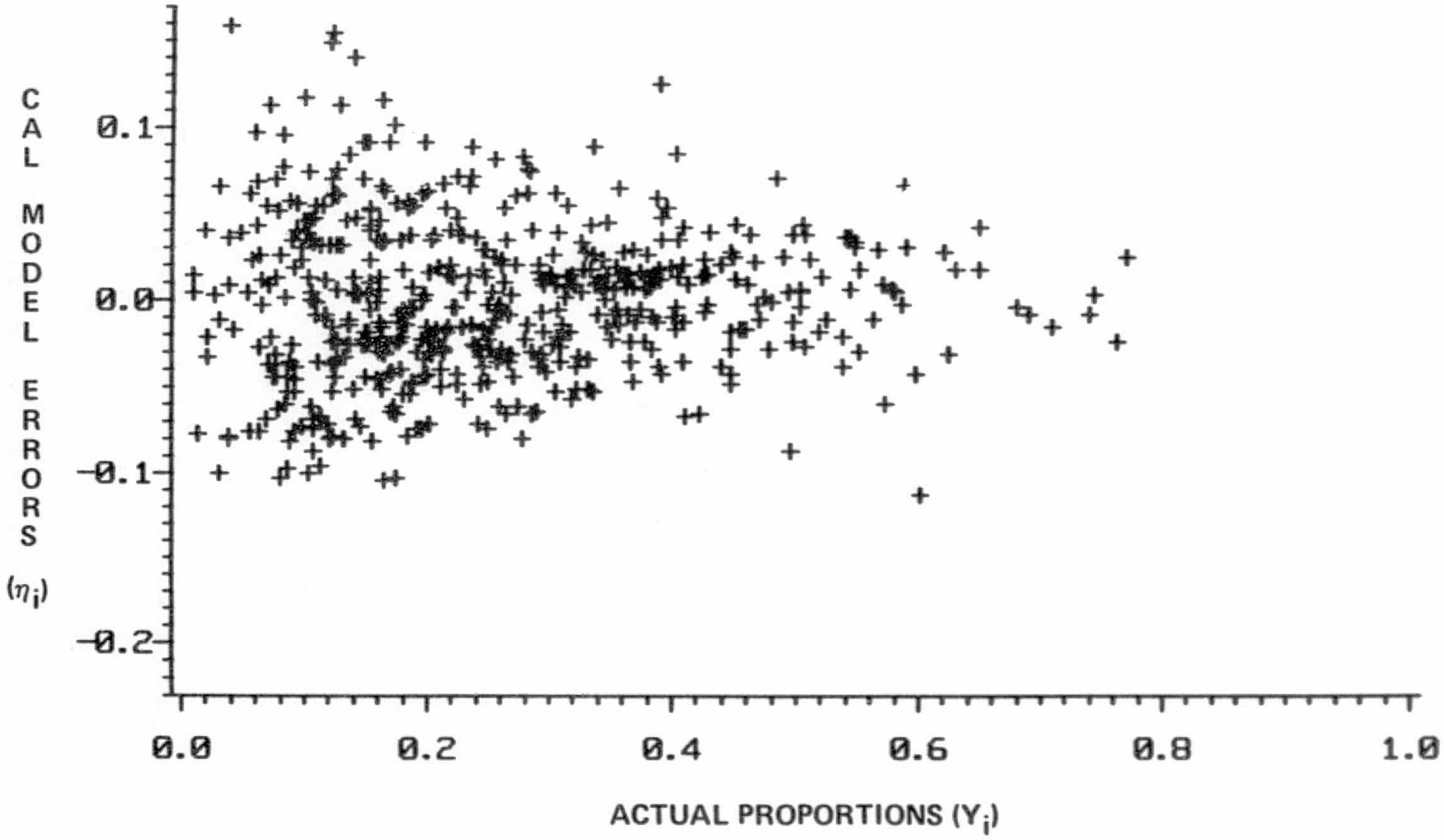


Figure 4-2.- Scatterplot of calibration model errors versus actual proportions with $\bar{Y}$ = .25, n = 10, σ =.1, Δ = 1.5 and p=1.

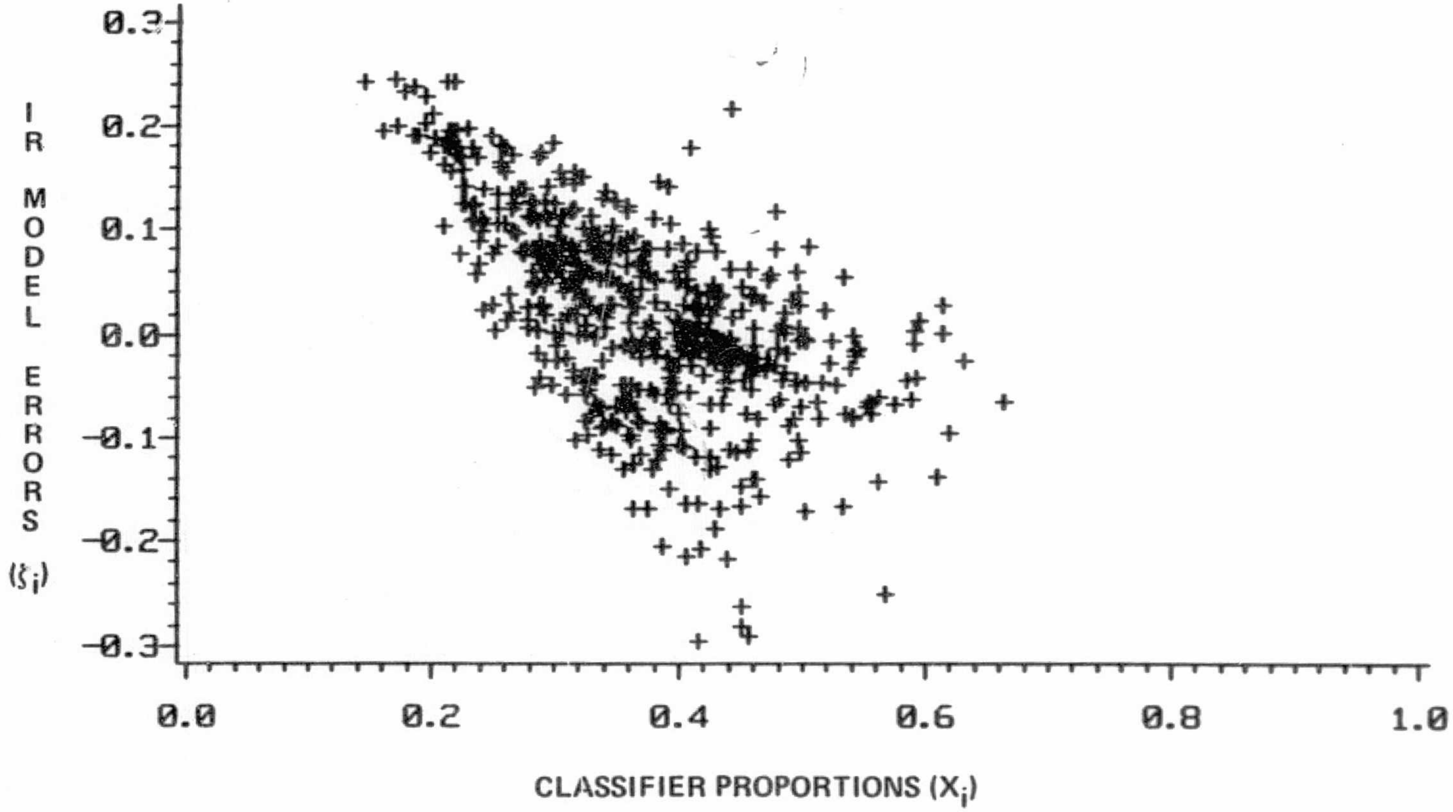


Figure 4-3.- Scatterplot of inverse regression model errors versus classifier proportions with $\bar{Y}$ = .25, n = 10, σ = .1, Δ = 1.5 and p = 1.

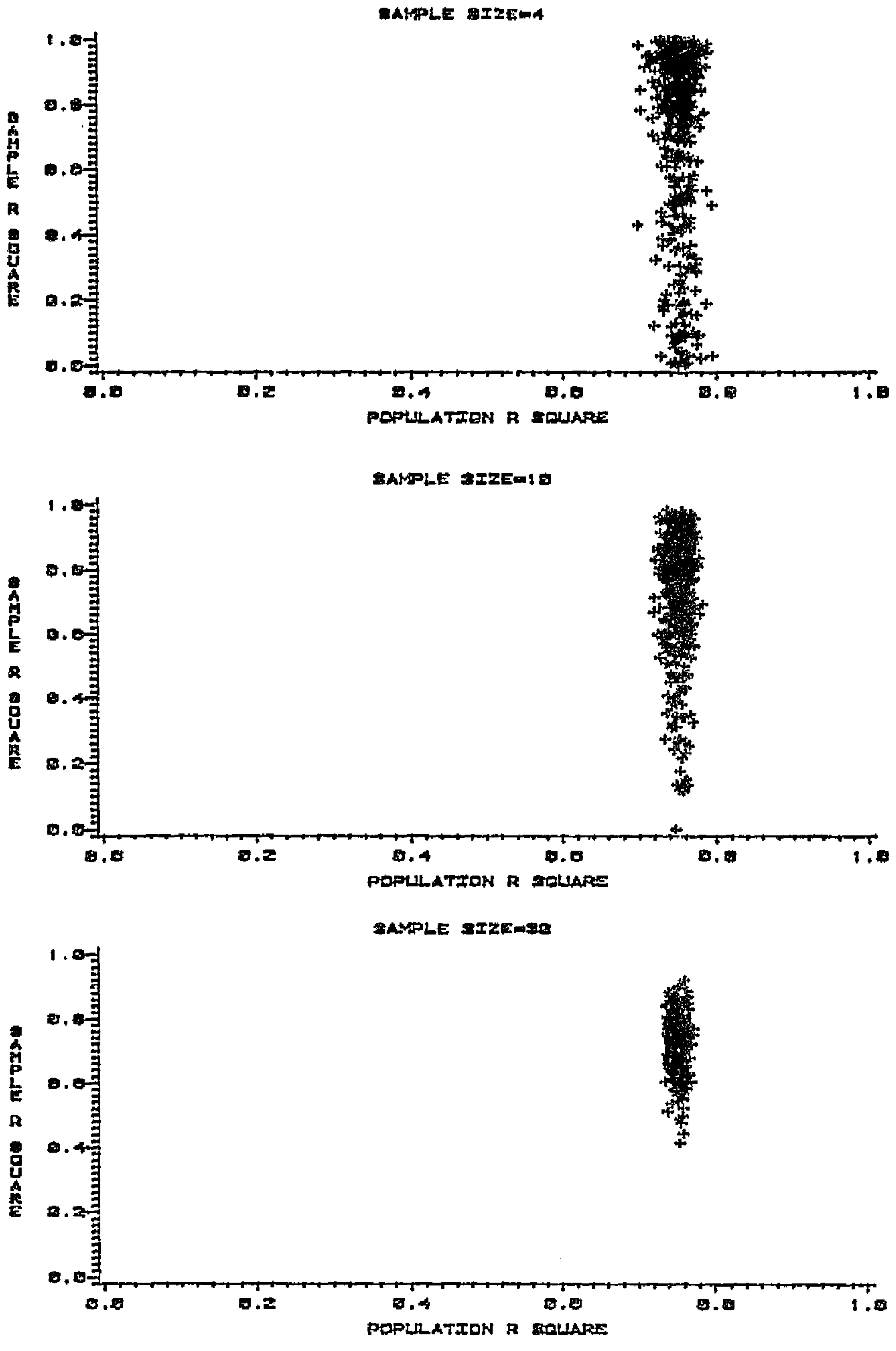
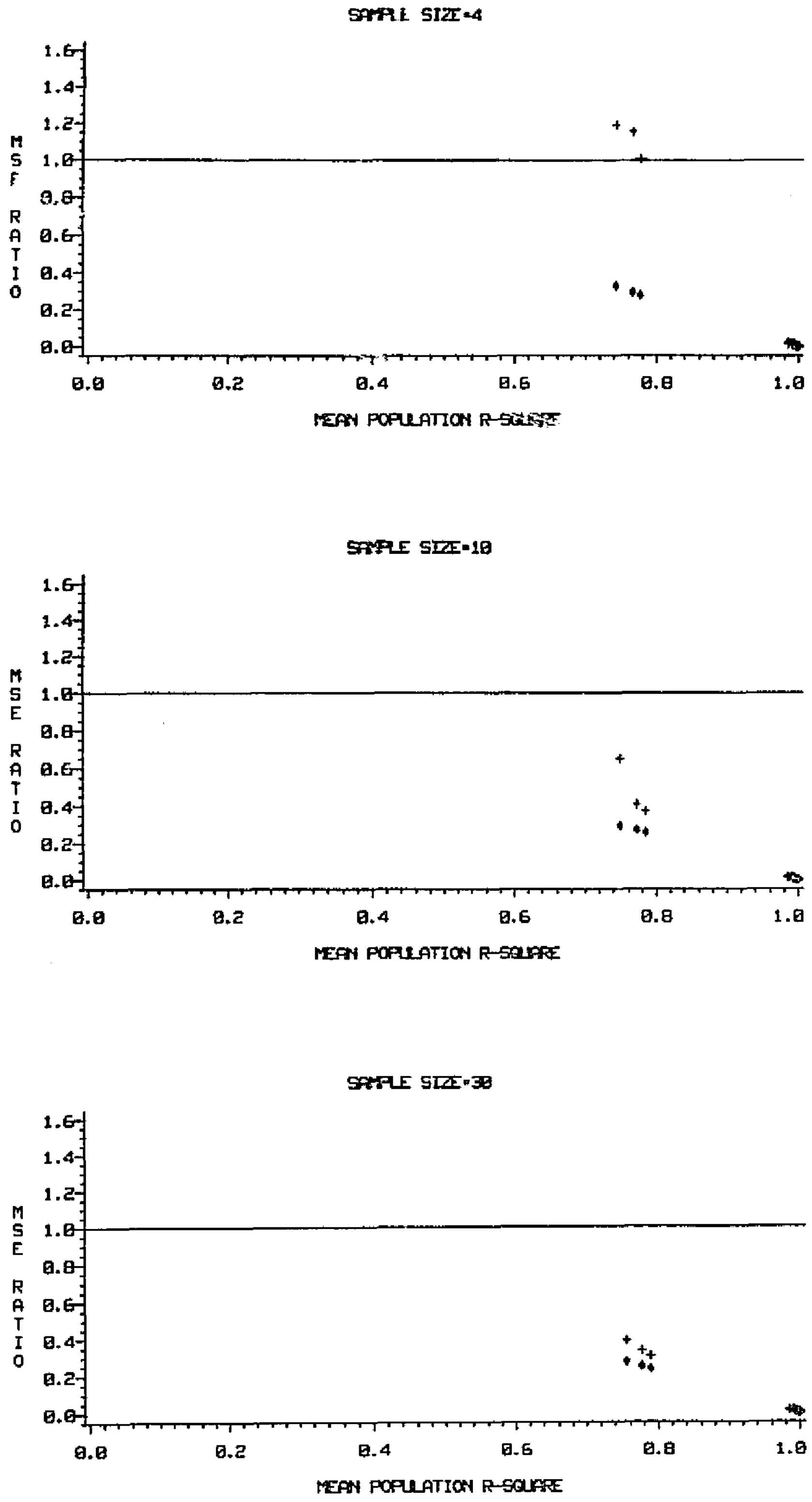


Figure 4-4.- Sample R^2 versus population R^2 when $\bar{Y}$ = .25 σ = .1, Δ = 1.5, and p = 1.0.



SAMPLE SIZE=4

SAMPLE SIZE=10

SAMPLE SIZE=30

Figure 4-5.- Scatterplots of the MSE ratio versus the mean population R^2 for the inverse regression estimator (*) and the calibration estimator (+) for $\bar{Y}$ = .5 and Δ = 3.

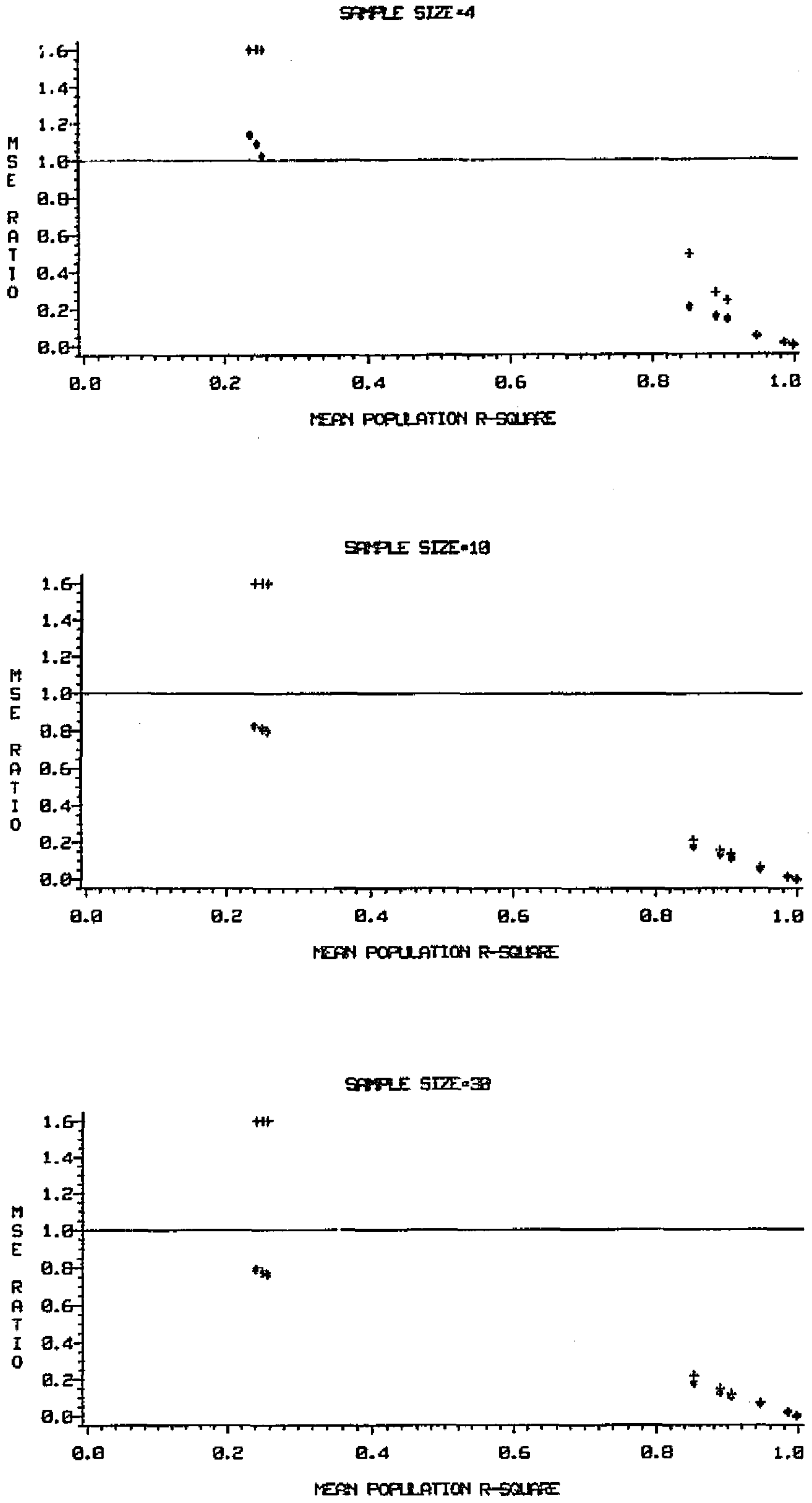


Figure 4-6.- Scatterplots of the MSE ratio versus the mean population R^2 for the inverse regression estimator (*) and the calibration estimator (+) for $\bar{Y}$ = .5 and Δ = 1.5.

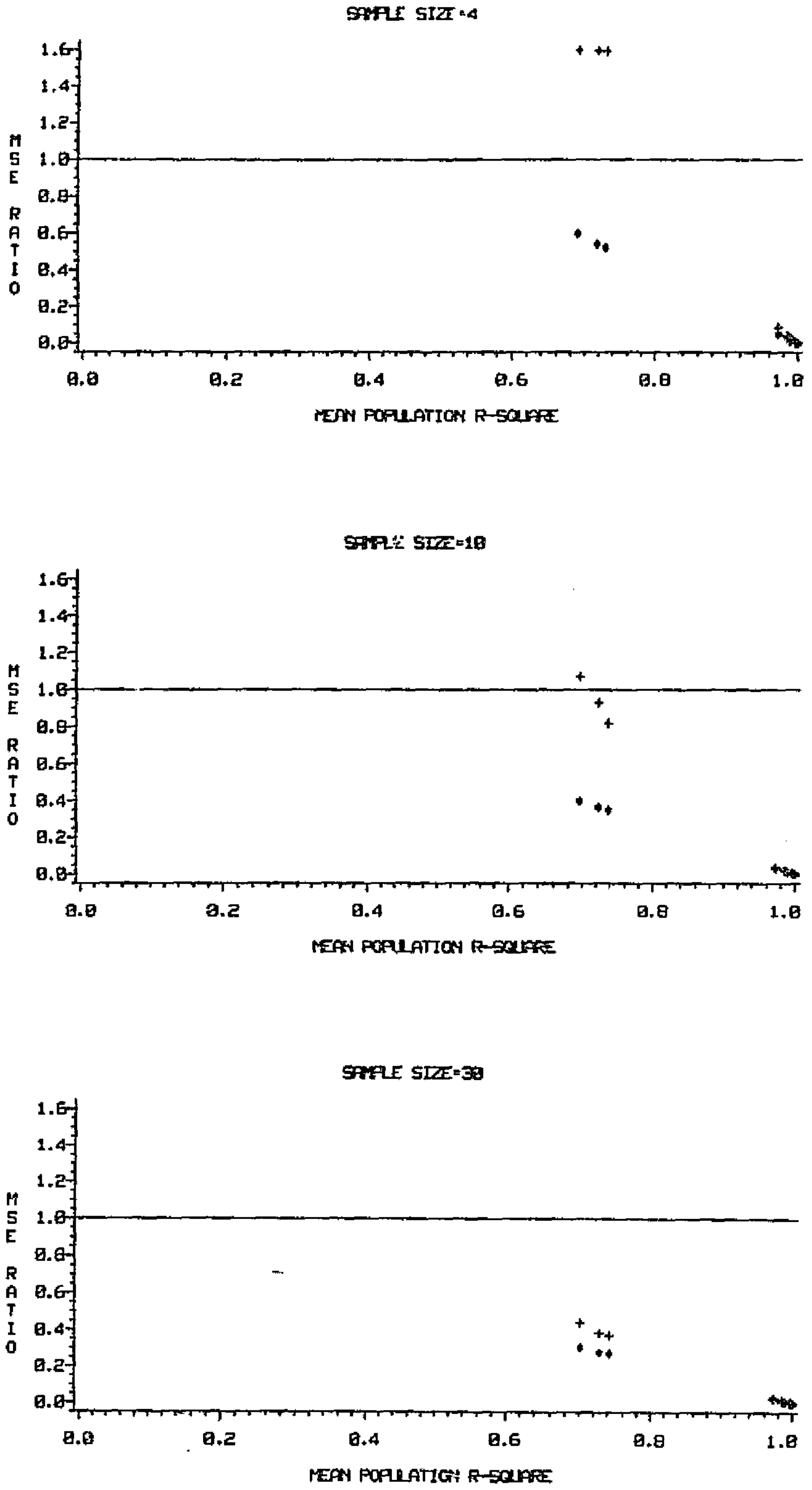


Figure 4-7.- Scatterplots of the MSE ratio versus the mean population R^2 for the inverse regression estimator (*) and the calibration estimator (+) for $\bar{Y}$ = .25 and Δ = 3.

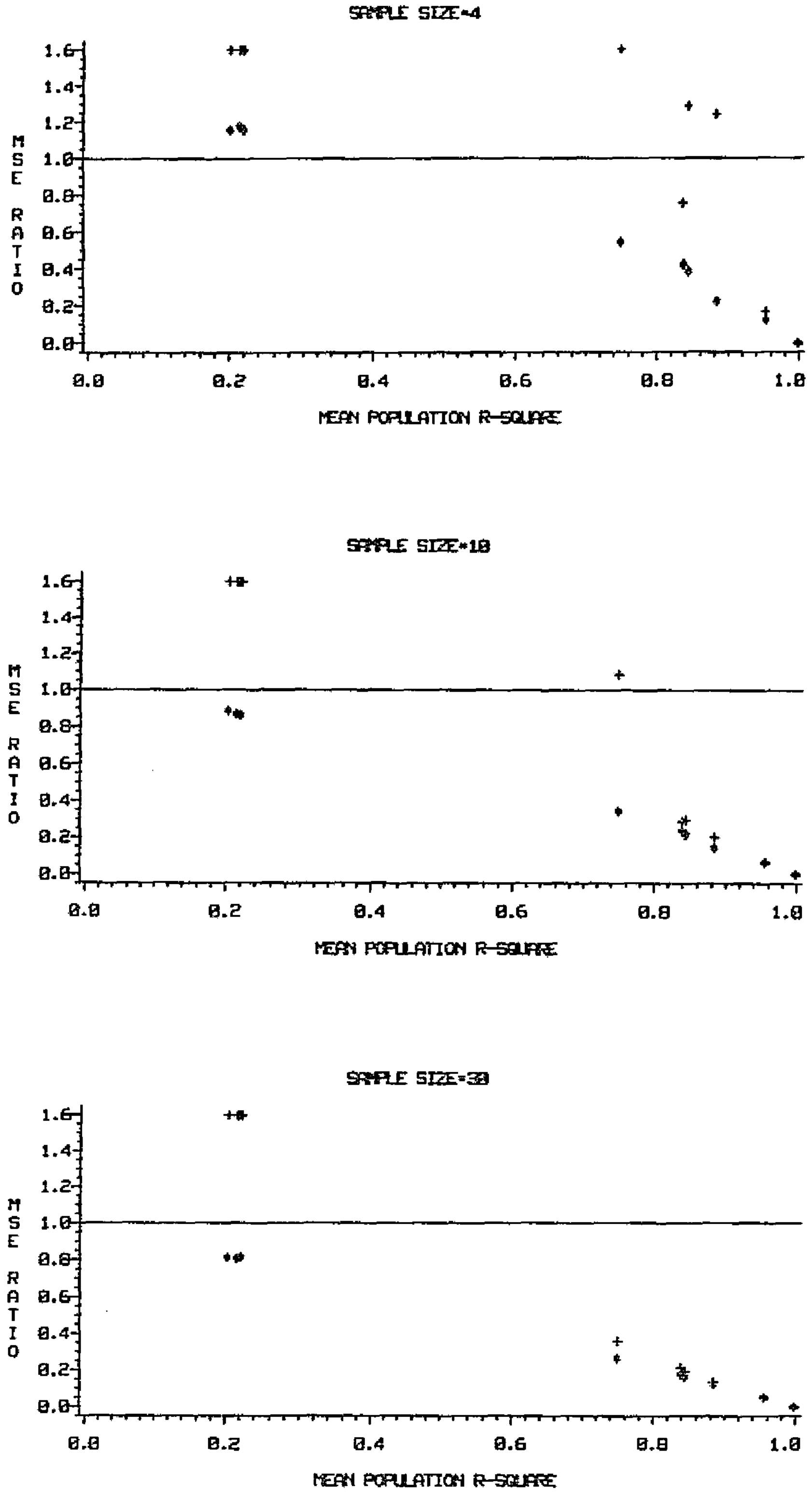


Figure 4-8.- Scatterplots of the MSE ratio versus the mean population R^2 for the inverse regression estimator(*) and the calibration estimator (+) for $\bar{Y}$ = .25 and Δ = 1.5.

Evidence Accumulation for Spatial Reasoning

Takashi Matsuyama
Vincent Shang-Shouq Hwang
and
Larry S. Davis

Center for Automation Research
University of Maryland
College Park, MD 20742

Original photography may be purchased
from EROS Data Center
Sioux Falls, SD 57198

ABSTRACT

This paper describes the evidence accumulation process of an image understanding system first described in [1], which enables the system to perform top-down(goal-oriented) picture processing as well as bottom-up verification of consistent spatial relations among objects.

1. Introduction

In a previous report[1], we described the organization of an aerial image analysis system. There are three levels of representation and control in that system: A High Level Expert(HLE) that utilizes a symbolic hierarchical model for the possible spatial organization of objects in the image to build partial, local interpretations of the image and to determine where to further analyze the image and what analyses to perform; a Model Selection Expert(MSE) that determines, on the basis of contextual information provided by the HLE, the most promising appearance descriptions to use in searching for objects and structures in the image; and a Low Level Vision Expert(LLVE) that finds pictorial entities that satisfy these appearance descriptions by selecting image processing methods to find the appropriate entities.

Our emphasis has been on the High Level Expert, which is based on a general method of "evidence accumulation" to perform flexible spatial reasoning. This paper contains a detailed description of our evidence accumulation process and its associated consistency checking process.

2. Motivation

In general, two different types of information can be used to interpret a pictorial entity: its intrinsic properties(size, shape, color etc.) and its relations to other entities. Our primary interest is the representation of geometric relations among objects and their utilization for image interpretation. This is especially important in recognition of man-made objects. Moreover, although shape can often be regards as an intrinsic object property, a complex shape is often described structurally in terms of geometric relations among its components. Thus shape recognition often requires spatial analysis.

Let REL(O1, O2) denote a binary geometric relation between two classes of objects, O1 and O2. This relation can be used as a constraint to recognize objects from these two classes by first extracting pictorial entities which satisfy the intrinsic properties of O1 and O2, and then checking that the geometric relation is satisfied by these candidate objects(Figure 1). In this bottom-up recognition scheme, analysis based on geometric relations cannot be performed until pictorial entities corresponding to objects are extracted.

In general, however, some of the correct pictorial entities often fail to be extracted by the initial image segmentation. So one must, additionally, incorporate top-down control to find pictorial entities missed by the initial segmentation. Such top-down processes use geometric relations to predict the locations of missing objects, as in the system described by Selfridge[2].

It is, of course, generally accepted that image understanding systems should incorporate both bottom-up and top-down analyses. As noted above, the use of geometric relations is very different in the two analysis processes:consistency verification in bottom-up analysis and hypothesis generation in top-down analysis. An important characteristic of our evidence accumulation method is that it enables the system to integrate both bottom-up and top-down processes into a single flexible spatial reasoning process. As will be described later, the system first establishes local environments. Then, either bottom-up or top-down processes are activated depending on the nature of the local environment. The following sections describe the concepts and characteristics of this process.

3. Representation of Geometric Relations and Hypothesis Formation

3.1. Functional Representation of Relations

A relation REL(O1, O2)(O1 and O2 are object classes) is represented using two functional expressions:

$$O1 = f(O2) \text{ and } O2 = g(O1).$$

Given an instance of O2, say r, function f maps it into a description of an instance of O1, f(r), which satisfies the geometric relation, REL, with r. The analogous interpretation holds for the other function g.

In our system, knowledge about a class of objects is reoresented by a frame[31, and a slot in that frame is used to store a function such as f or g. The function is represented by a computational procedure(which produces the description of the related instance) and a set of conditions to specify when that function can be activated. Whenever an instance of an object is created, and the conditions are satisfied, the function is applied to the instance to generate a hypothesis (expectation) for another object which would, if found, satisfy the geometric relation with the original instance. The function can use any properties of the instance to create the hypothesis.

A hypothesis is associated with a prediction area where the related object instance may be located(Figure 2). In addition to this area specification, a set of constraints on the target instance is associated with the hypothesis. Figure 3 shows the description of a road hypothesis. All hypotheses and instances are stored in a common database(the iconic database) where accumulation of evidence (i.e., recognition of overlapping sets of consistent hypotheses and instances) is performed. Similar ideas have been proposed to solve spatial layout problems[4] and to answer queries about map information[5].

3.2. Spatial Relations, Part-Whole Relations, and A-Kind-Of Relations

Two types of geometric relations are used in our system: "spatial relation"(SP) and "part-whole relation"(PW). These two types of relations are used differently by the system. The PW relations specify AND/OR hierarchies which represent objects with complex internal structure. The SP relations represent geometric and topological relations between objects. In addition, "A-kind-of relations"(AKO) are used to construct object specialization hierarchies.

There are several restrictions on the usage of these types of relations. A hierarchy defined by the PW relation must be a tree structure. Although SP relations can be established across objects in different PW hierarchies, an object cannot have an SP relation with another object in the same PW hierarchy, nor can it establish multiple SP relations to any other PW hierarchy. These restrictions were adopted to avoid redundant generation of hypotheses.

Consider the knowledge representations shown in Figures 4(a) and (b). If object A had an SP relation to object B in the same part-whole hierarchy(Figure 4(a)), there would be two paths from object A to generate a hypothesis of object B: one by the SP relation and the other by the PW relation. This means that if an instance of object A were constructed, two hypotheses for object B would be generated from the same instance. The same argument holds in the case shown in Figure 4(b). Figure 4(c) shows a circular path consisting of SP relations between objects A, B, and C. This is allowed since no redundant hypotheses are formed.

Hypothesis generation by an SP relation is done as explained above, i.e., when an object is instantiated and the set of conditions needed to generate a hypothesis are satisfied, then the function associated with the SP relation is activated to produce an expectation area and an associated set of constraints for a target object. Although, syntactically, SP relations represent binary relations, it is possible to use them to represent n-ary relations. For example, a left eye can create a hypothesis for a nose, and can use the known location of a potential right eye to generate the nose hypothesis.

The system uses PW relations both to group parts into a whole and to predict missing parts. If an instantiated object corresponds to a leaf node in the PW hierarchy, then it can directly instantiate (again, if prespecified conditions hold) its parent node through the PW relation(Figure 5).

Objects at the leaves of PW hierarchies are instantiated first, since they correspond directly to low-level image structures. The presence of a higher level object is represented by an instantiated

PW hierarchy. The parent may then <u>hypothesize</u> the presence of other missing object parts. For computational simplicity, there are no hypotheses generated between siblings in the PW hierarchy.

4. Combining Evidence

4.1. The Interpretation Cycle of the High Level Expert

Figure 6 shows the organization of the entire system. The High Level Expert iterates the following steps.

(1) Each instance of an object generates hypotheses about related objects using functions stored in the object model(frame).

(2) All pieces of evidence(both instances and hypotheses) are stored in a common database(iconic database). They are represented using an iconic data structure which associates highly structured symbolic descriptions of the instances and hypotheses with regions in a two-dimensional array.

(3) Pieces of evidence are combined to establish situations. A situation consists of consistent pieces of evidence.

(4) Focus of attention : since there are many situations, the most reliable situation is selected.

(5) The selected situation is resolved, which results either in verification of predictions on the basis of previously detected/constructed image structures or in top-down image processing to detect missing objects.

The system also has two additional processes:

(1) Instantiation of objects at the very beginning of interpretation This process is performed by the Model Selection Expert which searches for object models that have simple appearances, and directs the Low Level Vision Expert to detect pictorial entities which satisfy the appearances. The instances constructed by this process

are seeds for reasoning by the High Level Expert.

(2) Selection of the maximum consistent interpretation

During the analysis by the High Level Expert, inconsistent pieces of evidence may be constructed. The High Level Expert maintains all possible interpretations throughout the search process until no further changes are made in the iconic database. A final interpretation then selects the maximal consistent interpretation.

The following subsections provide detailed discussion of the operation of the High Level Expert.

4.2. Overview

Given a set of instances of objects, each of them activates functions to generate hypotheses about related objects. Each instance and hypothesis is represented as a region in the iconic data structure. Suppose instance s creates hypothesis f(s)(based on relation R) for object class O1, which overlaps with an instance of O1, t(Figure 7(a)). If the set of constraints associated with f(s) is satisfied by t, these two pieces of evidence are combined to form what we call a situation. The more pieces of evidence that are combined, the more reliable the situation becomes. The High Level Expert unifies f(s) and t, and establishes the relation R from s to t as the result of resolving the situation.

On the other hand, a situation may consist of overlapping hypotheses, if their constraints are consistent(Figure 7(b)). Then their unification leads the expert to search for an instance of the required object in the image. The High Level Expert asks the Model Selection Expert to detect the instance, which in turn activates the Low Level Vision Expert. If the instance is detected, it is inserted into the database. Hypothesis generation by the newly detected instance is performed at the next interpretation cycle.

4.3. Handling PW relations

Additional complications arise from resolving situations involving instances generated via PW relations. Suppose s is an instance of an object corresponding to a leaf node in a PW hierarchy(Figure 8(a)). As described above, it may instantiate its parent object. Let p denote this instance. Then p generates a hypothesis for a missing part, f(p). If there is already an instance corresponding to the missing part, say t, f(p) and t will be unified, and a part-whole relation will be established between p and t. However, since t is also an instance, it may also have instantiated its parent object. Let u denote this instance. As the result of the unification, instance t has two parent instances, p and u. This leads the High Level Reasoning Expert to another unification. The expert examines p and u, and if they are consistent, it unifies them(Figure 8(b)). This unification may trigger still another unification for higher level instances in the hierarchy. Note that after the unification, instance p can use properties of r and t to generate hypotheses for other part objects whose geometric properties could not previously be specified due to a lack of sufficient information.

If the two parent instances(p and u) were found not to be consistent, the expert records such mutually conflicting interpretations, and will perform reasoning independently based on each interpretation. The process of reasoning with alternative interpretations, is not described in detail in this paper.

There can be a still more complicated situation created by a PW relation. As shown in Figure 9(a), suppose the grandparent object has also been instantiated by an instance of a leaf object, r. Let p and q denote instances of the parent and grandparent objects, respectively; q as well as p generates hypotheses for its missing parts, say f(q). Suppose that f(q) itself has parts and one of them has already been instantiated. Let s denote that instance. Then, if instances r and s are really parts of the same object, regions of f(q) and s will overlap with each other and will be consistent.(A detailed discussion of consistency will be given in the next subsection.) In this case, the system first constructs a situation based on the intersection of f(q) and s, even if their description levels in the PW hierarchy are different, and then unifies f(q) and t(the parent instance of s). Note that instance t cannot intersect with f(q) directly since no iconic region is associated with t in the database. As a result, r, p, q, s, and t are organized into one hierarchical structure(Figure 9(b)). If, as shown in Figure 9(c), the levels of f(q) and t in the hierarchy are different(in Figure 9(b), they are at the same level), a series of parent objects are instantiated from instance s.

4.4. Forming Consistent Situations

Consistent pieces of evidence from different sources are combined into situations. The consistency among pieces of evidence is based on:

(1) prediction areas of hypotheses

(2) object categories of evidence

(3) constraints imposed on properties of hypotheses and instances

(4) relations among sources of evidence

These criteria are discussed in the next four subsections.

4.4.1. Intersections of Prediction Areas

Figure 10(a) shows all intersections formed from pieces of evidence E1, E2, E3, and E4. A partial ordering on intersections can be constructed on the basis of region containment. Intersection OP1 is less than OP2 if region OP1 is contained in region OP2. Figure 10(b) shows the lattice representing the intersection in Figure 10(a). Each intersections consists of some set of hypotheses and instance. Situations are only formed among intersecting pieces of evidence.

4.4.2. Object Categories of Evidence

In our domain, some pairs of objects cannot occupy the same location in an image. For instance, a region cannot be interpreted as both house and road at the same time(although it could be interpreted both as road and shadow). Pairs of frames representing object classes which cannot occupy the same region are linked with an <u>in-conflict-with</u> relation.

Let OP be the intersection arising from evidence {E1, E2} and let OBJ1 and OBJ2 denote the object categories of E1 and E2, respectively. If OBJ1 and OBJ2 are linked by an in-conflict-with relation, then E1 and E2 are said to be conflicting, and OP is removed from the lattice. The removal of OP is propagated through the lattice, and any intersections contained in OP are also removed, since they must also have arisen from conflicting evidence. To find all conflicting intersections, it is clearly sufficient to examine all intersections containing only a pair of pieces of evidence and then to propagate the results through the lattice.

In the above case, if both E1 and E2 are instances, the High Level Reasoning Expert records them as conflicting and use that fact to establish the inconsistency of situations containing hypotheses generated by conflicting instances. (See section 4.4.4.)

A shortcoming of our approach to evidence accumulation is that negative sources of evidence are not considered in assessing the strength of a situation. For example, in medical diagnosis, some measurements are used to deny the possibility of certain classes of diseases. Incorporation of sources of negative evidence is an important issue for future research.

4.4.3. Constraint Consistency

After eliminating all conflicting intersections, the remaining intersections are checked to determine if their associated sets of constraints are consistent. Let E1 and E2 denote the non-conflicting evidence under consideration. One of the following conditions must hold:

(1) The object categories of E1 and E2 are the same,

(2) there is a path between the two categories consisting of PW relations,

or

(3) one piece of evidence is a subcategory of the other, according to the specialization/generalization hierarchy.

In the second case, since the names of the attributes used in the constraints associated with E1 and E2 may be different, they cannot, in general, be directly compared. Suppose the object category of E1 is at a higher level in the hierarchy than that of E2. The constraints associated with E2 are translated into those for the object category of E1 by using part-whole/a-kind-of relations. Then the translated constraints are compared with those associated with E1.

Figure 11 illustrates the translation of constraints using PW relations. Constraint C1 on a road piece object is translated into constraint C2 on a road object. Currently, this translation is done simply by rewriting the attributes(slot names) of C1 into appropriate attributes(slot names) of C2 using a "slot name translation table" for the PW relation(Figure 11.b).

The properties and/or constraints associated with both pieces of evidence must be consistent. Both constraints associated with a hypothesis and properties associated with an instance are represented by sets of linear inequalities in one variable. A simple constraint manipulation system is used to check the consistency between the sets of inequalities by generating the solution space(also represented by inequalities) to the intersection of sets. If this solution space is empty, then the constraints are inconsistent. If C1 are the constraints for E1, C2 for E2, and C for O, the object category to which both E1 and E2 belong, then we must check that

$$(C_1 \cap C_2) \text{ and } C \neq \emptyset$$

We do this by first computing C3 = C1 ∩C2, and if this is non-empty, finally computing C3 and C.

4.4.4. Relations Between Sources of Evidence

The sources of accumulated evidence about a situation must not be conflicting. Let S1 and S2 denote the source evidence of E1 and E2, respectively. If a piece of evidence is a hypothesis, its source evidence is the instance which generated the hypothesis. An instance is the source evidence for itself. It is possible that S1 and S2 are mutually conflicting(Figure 12), but that E1 and E2 themselves are consistent. In such a case, we do not combine E1 and E2 into a situation; analysis based on such conflicting interpretations is performed independently.

4.5. Focus of Attention

After examining the consistency among evidence, we next evaluate the reliability of each consistent situation by summing numerical reliability measures for each piece of evidence, and select the most reliable one for further analysis. This is the focus of attention mechanism.

4.5.1. Controlling the Intermediate Interpretation Process

Recall that there are two different types of evidence in our system: instances and hypotheses. It is possible to control the direction of the interpretation process by assigning different reliabilities to them.

If a higher reliability is assigned to an instance than to a hypothesis, a situation including an instance tends to be selected as the most reliable one rather than one consisting only of hypotheses. Therefore the system first builds partial interpretations by establishing relations among instances before trying to perform top-down picture processing.

5. Resolving a Situation

As described in Section 4.2, one of two actions is taken in order to resolve a situation: confirm relations between instances or activate top-down analysis.

How a situation is resolved depends on the nature of its constituent evidence. If the pieces of evidence are all hypotheses, then a composite hypothesis is constructed for transmittal to the MSE, and any instance extracted from the image is then examined by the source instances of those hypotheses. If a situation includes both hypotheses and instances, then the instances are, in turn, examined by the sources of the hypotheses, and if none satisfy the hypotheses, then a composite hypothesis can, in turn, be transmitted to the MSE.

5.1. Resolution Process

The system provides a description of its proposed resolution to a situation to all instances involved in that situation. Each instance then evaluates the proposed solution according to its specific expectations.

In what follows, the process of resolving a situation is illustrated by the example shown in Figure 13. Suppose the consistency reasoner selected the overlapping region between two hypotheses generated from two road-piece instances RP1 and RP2(Figure 13(a)). In the symbolic data structure, RP1 and RP2 are linked to their parent road instances RD1 and RD2 by PW relations, respectively. The hypotheses for adjacent road pieces have been generated by these parent instances.

Since this situation consists only of hypotheses, the system activates top-down analysis to find a road piece in the overlapping region. This request is issued to the Model Selection Expert together with the supporting evidence(i.e. RD1 and RD2), so that the expert can use any available contextual information.

Assume that a new road-piece instance, RP3, is created(Figure 13(b)). Then, the system provides this result to the instances involved in the situation, namely RD1 and RD2.

Suppose RD1 is the first to be informed of the proposed resolution. RD1 examines whether or not RP3 satisfies all constraints required to establish relation R1. In this case, however, RP3 fails, because RP3 is not adjacent to RP1. This failure activates an exception handler, which issues a top-down request to find a road-piece between RP1 and RP3(see Figure 13(c)).

Assume that another new road-piece instance, RP4, is detected(Figure 13(d)). Since RP4 is adjacent to RP1, RD1 establishes a PW relation to RP4, and then to RP3.

Figure 13(e) shows the data organization after the same analysis is performed by RD2. In this case, however, when RD2 establishes a PW relation to RP3, an exception handler in RP3 is triggered, because RP3 has two different parents. More specifically, after RD2 establishes a PW relation to RP3, RD2 asks RP3 to check its reverse relation from RP3. An exception handler is activated as a result of this checking process. This handler issues a request to the system to examine the consistency between two parents. If they are consistent, the system merges the two PW hierarchies below them into one(Figure 13(f)). An exception handler of this kind is associated with each PW relation in order to construct a complete PW hierarchy by merging a pair of partial hierarchies.

There are several stages in the above example where the top-down request might have failed. In general, the Model Selection Expert has the ability to deal with such failures. Figure 14 shows a partial knowledge structure for suburban scenes. The Model Selection Expert analyzes the request to find RP3 (Figure 13(a)) by first assuming the road piece to be detected is a visible road, and issues a request to the Low Level Vision Expert. If this request fails, the Model Selection Expert switches to the other appearance of a road piece, i.e. an occluded road. The selection between overpass and shadowed road is done based on the cause of the failure. For example, if the cause of the failure is that the gray level in the overlapping region is too dark compared to the expected gray level, then the expert will hypothesize a shadowed road. If all efforts by the Model Selection Expert fail, this is reported to the High Level Expert. Then the system reports this to RD1 and RD2, which trigger their relevant exception handlers. Since different new hypotheses may be generated by such exception handlers, no immediate further analysis is activated. Instead, these hypotheses are combined in the next interpretation cycle. In the case of Figure 13, RD1 and RD2 would both generate hypotheses for a road terminator.

If a top-down request issued by an instance fails, the instance activates another exception handler, if anv. If all trials fail, the instance reports this to the system. Then the system activates another instance involved in the focused situation. The initial failure is not taken into account in anv way by the svstem, this is a shortcoming of the present system.

5.2. Merging a Pair of Partial PW Hierarchies

If a part instance is shared by two parent instances, the part issues a request to check the "similarity" between the parents. If they are similar, the system merges them into one.

Similarity examination involves checking whether or not the two parent instances denote(perhaps different pieces of) the same object. For example, RD1 and RD2 in Figure 13(e) should be merged into one road, although they do not denote the same (portion of) road. Knowledge about the continuity of roads is crucial in this example.

The more reliable of the two instances to be merged checks whether or not the part instances of the other instance are consistent with that more reliable parent. The more reliable parent may decide to merge with the other parent, that such a merge is not(and will never be) possible(which places them in conflict) or that sufficient information is not available to make a decision.

Figure 15 illustrates an example of the third case. The definition of a house group is a group of regularly arranged houses which face the same side of the same road. As shown in Figure 15, if two house group instances share a house instance, the similarity examination is performed. If both house group instances face the same side of the same road instance, then they are similar and are merged into one. On the other hand, if one of them has not established such a "facing" relation, then it is not possible to verifv the similarity between them. Moreover, even if the two house group instances have established "facing" relations to different road instances, it is still possible for them to be similar, because those road instances may be merged later. The house group instances can be regarded as conflicting only if their facing road instances are in conflict.

If the result of the similarity examination is "inconclusive", the system records the causes of the failure and suspends the action of establishing a new PW relation from a parent instance to the shared part instance. In the case shown in Figure 15, the relation between HG1 and H3 is suspended. The system records all suspended actions together with their causes. The suspended action can be reactivated if its cause is resolved by analyzing other situations.

6. Experimental Results

The image used in our experiment is a 320 bv 160 portion of an aerial photograph(Figure 16) with intensities in the range of 0 to 63. The scene contains houses, roads, road intersections, trees, and driveways.

The appearance models are a subset of the possible models for suburban housing developments. Currently, we deal only with houses, road pieces, road intersections, and the spatial relations among them. Figure 14 shows the suburban housing development model used. In this section, we describe how our system proceeds to construct a road network interpretation from the image.

The system's analysis starts with a segmentation of the image. Since the houses and road pieces are modeled by compact and elongated rectangles, such rectangles are first extracted from the image. A simple blob finder and ribbon finder are used to find blobs and ribbons in the image.

Elongated rectangles are extracted and instantiated as road piece instances. These instances constitute the initial entries in the iconic database. Figures 16 shows the initial road-piece instances extracted from the image. As can be seen, roads are broken into pieces.

In the first cycle of the interpretation cycles, the system checks each instance and, for each relation, creates a hypothesis(for an SP relation or a top-down usage of a PW relation) or an instance(for a bottom-up usage of a PW relation), if possible, and inserts it into the database. Since some of the relations may depend on yet undetermined values stored in frame slots, not all relations may be hypothesized at this point.

In the second cycle, the system's focus of attention mechanism selects the most promising situation. After a situation is selected, the system resolves it by first proposinq a solution to it and then broadcasting messages to the source instances. Each source instance checks the proposed solution and requests the MSE to do top-down analysis if necessary. Also, the system may reorganize the database(e.q., unification of instances) during the resolution process.

In the current experiment, the MSE is simulated by a human. The descriptions of the action and the situation are displayed on the screen. The description of the result is entered from the terminal and is instantiated as an object instance and returned to the system.

Figures 17 - 23 show how the system proceeds to select a situation, resolve the selected situation, and reorganize the database as the result of resolving that situation. Figures 17 and 18 show two road-piece instances RP1, RP2, their parent instances RD1, RD2, and the hypotheses that RD1 and RD2 generate. During the hypothesis creation cycle, instances RD1 and RD2 create hvpotheses H1, ... , H8. Hypotheses H4 and RP2 overlap(Figure 19.a). The system picks this situation(H4 and RP2 are consistent) and proceeds to resolve it.

Let C be the summarized constraints derived from the constraints of H4 and RP2. Since RP2 satisfies the constraint C, the system uses it as a proposed answer . RD1 checks the proposed solution, RP2, for adjacency. However, RP2 is not adjacent to RD1. RD1 issues a top down request to the MSE to find a road piece instance to connect RD1 and RP2. Currently, such a request is displayed on the screen and the result is entered from the terminal. The result can either be success, in which the description of the instance(object type and region description) is entered, or failure.

The description of a road piece instance(RP3) is entered from the terminal. MSE instantiates the instance and inserts it into the database. MSE reports RP3 to RD1. RD1 checks if RP3 is adjacent to RD1. Since RP3 is adjacent to RD1, RD1 establishes a PW link to RP3(Figure 20.b). Finally, RD1 checks MSG-A again and succeeds(since RD1 contains RP1 and RP3.) A PW link is established between RP2 and RD1(Figure 20.c). As a result, RP2 belongs to two parents. The system tries to unify them by checking if RD1 and RD2 are similar. this case, they are similar. The system unifies RD1 and RD2 into a single instance(say RD´.) After the unification, road instance RD´ has three parts(RP1, RP2, and RP3). Figure 21 shows the road instance RD´ and its three parts. Figure 22 shows all the road instances after the selected situation is resolved.

During the unification process, several instances are merged into a single instance. The hypotheses generated by the merged instances are removed from the database. A new set of hypotheses is generated in the next hypothesis creation cycle. Figure 23 shows the new hypotheses generated by RD'. Note that the original hypotheses H1, ... , H8 generated by RD1 and RD2 have been removed from the database.

Figure 24 shows a case where alternate hypotheses are generated. A road can either be extended continuously, or stop at a road terminator. One way to conduct the search is to look for the adjacent road piece first. If that search fails, then the search for a road terminator can start. Such a strategy is illustrated in Figure 24.a. Figure 24.b shows a road instance and the alternate hypotheses it generates during the process.

Figure 25.a shows the final result of constructing the road network interpretations by the system. The interpretation graphs are shown in Figure 25.b. Each node represents an instance. There are 29 road piece instances, 10 road instances, and 5 road terminator instances. Figure 26 shows the road joint instance J1 and all road instances meeting there. Figure 27 shows road instance R2, the road terminator instances adjacent to it, and its part objects.

REFERENCES

[1] V. Hwang, T. Matsuyama, L.S. Davis, and A. Rosenfeld, "Evidence Accumulation for Spatial Reasoning in Aerial Image Understanding", Technical Report CAR-TR-28, Center for Automation Research, University of Maryland, 1983.

[2] P.G. Selfridge, "Reasoning about Success and Failure in Aerial Image Understanding", Ph. D. thesis, University of Rochester, 1982.

[3] M. Minsky, "A Framework for Representing Knowledge", in The Psychology of Computer Vision (P. Winston, ed.), McGraw-Hill, NY, 1975.

[4] R.L. Haar, "The Representation and Manipulation of Position Information Using Spatial Relations", Technical Report TR-923, Computer Vision Laboratory, University of Maryland, 1980.

[5] D. McDermott, "A Theory of Metric Spatial Inference", Proc. of Natl. Artifical Intelligence Conference, pp.246-248, Aug. 1980.

Check the validity of REL(O1, O2) for a pair of pictorial entities which may be instances of O1 and O2.

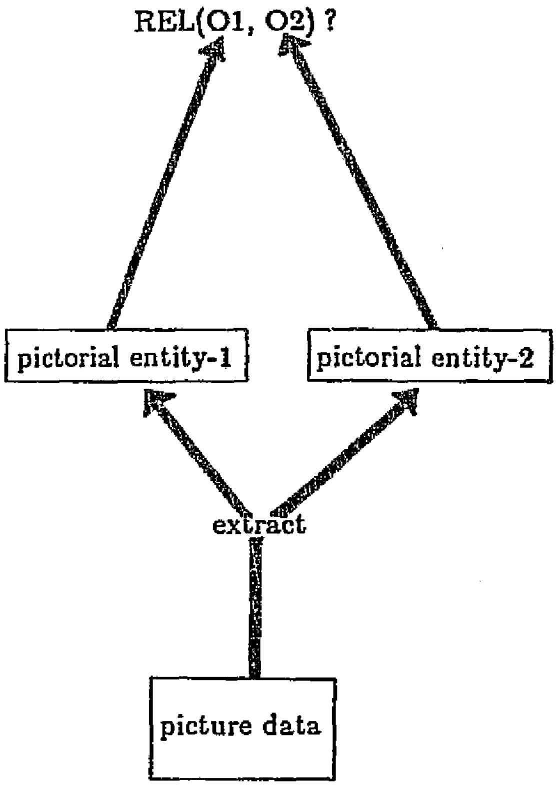


Fig. 1 Using a relation as a constraint.

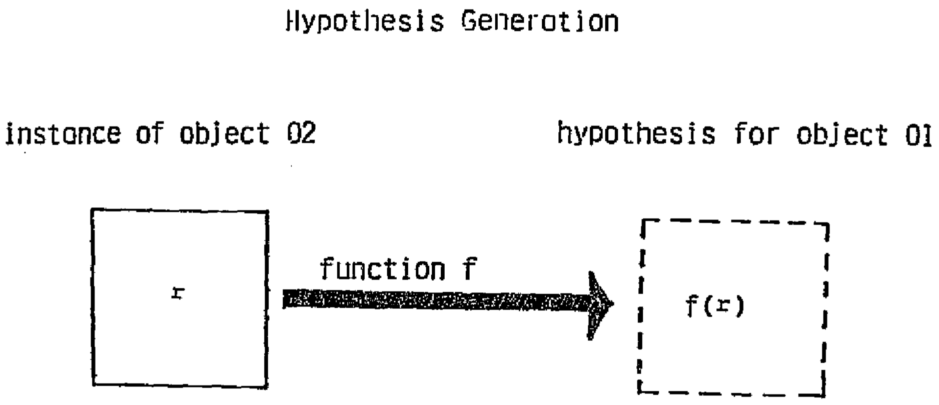


Fig. 2 Hypothesis generation based on functional representation of a relation

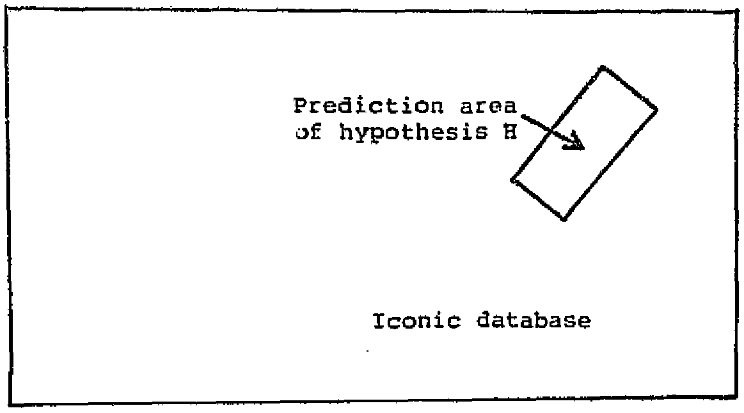
```
Frame name : Road piece
Slot name  : Length
             Width
             Direction
             Coordinate of the local coordinate system
             Father
```

(1) The description of the road piece frame

```
Frame name : Road
Slot name  : Total-length
             Average-direction
             Left-adjacent-road-piece
             Right-adjacent-road-piece
             Left-connecting-road-terminator
             Right-connecting-road-terminator
             Left-neighboring-house-group
             Right-neighboring-house-group
```

(2) The description of the road frame

Figure 3 : (a) The description of the road frame and the
road piece frame.

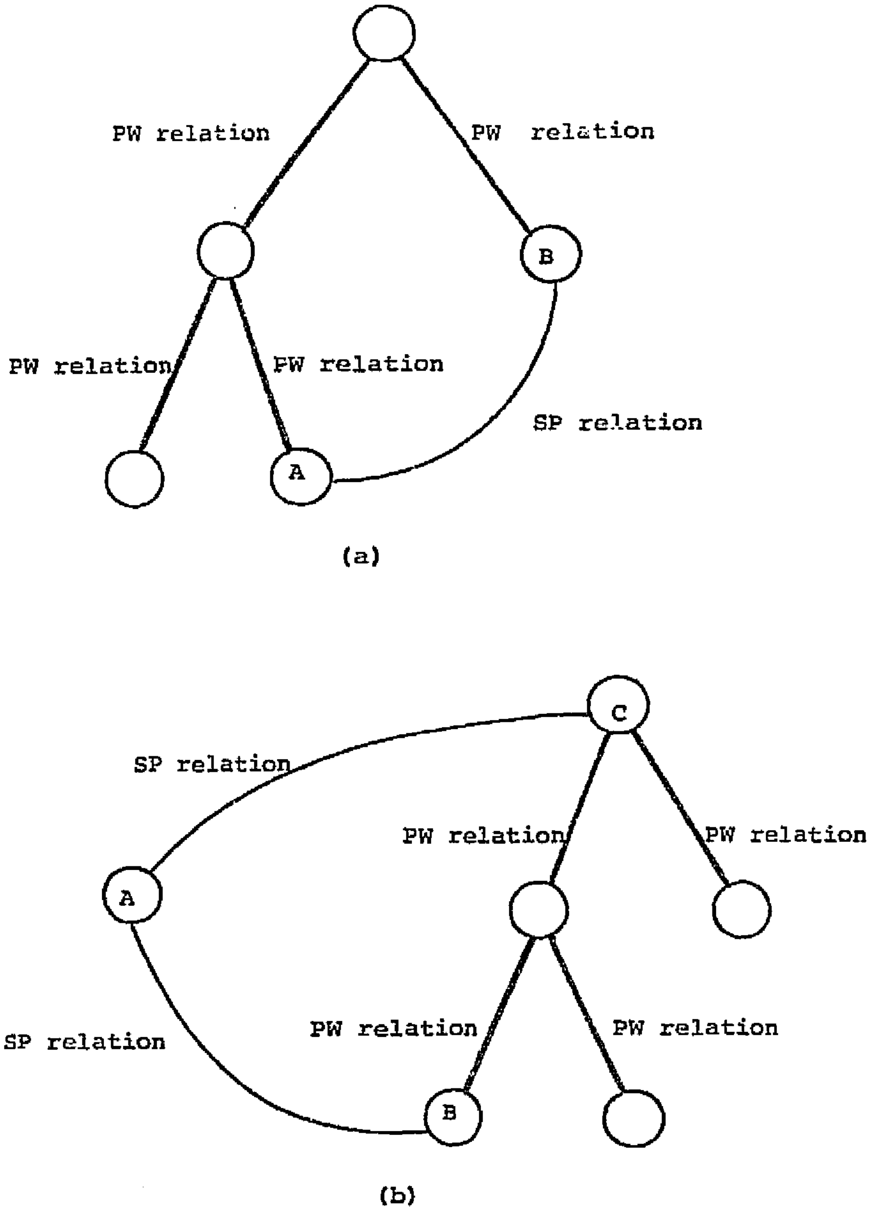


(1) Iconic description of hypothesis H

```
(AND (EQUAL OBJECT-TYPE ROAD)
     (AND (LESSP TOTAL-LENGTH 100)
          (GREATERP TOTAL-LENGTH 50))
     (AND (LESSP AVERAGE-WIDTH 15)
          (GREATERP AVERAGE-WIDTH 10))
     (AND (LESSP AVERAGE-DIRECTION 50)
          (GREATERP AVERAGE-DIRECTION 30)))
```

(2) Symbolic description of hypothesis H

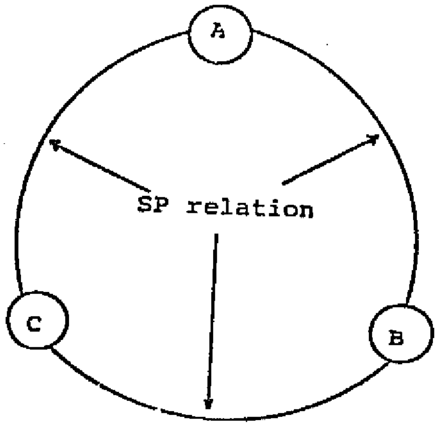
Figure 3 : (b) The description of a road hypothesis H.



(a)

(b)

Fig. 4 Avoiding redundant accumulation of evidence



(c)

Figure 4 (cont.)

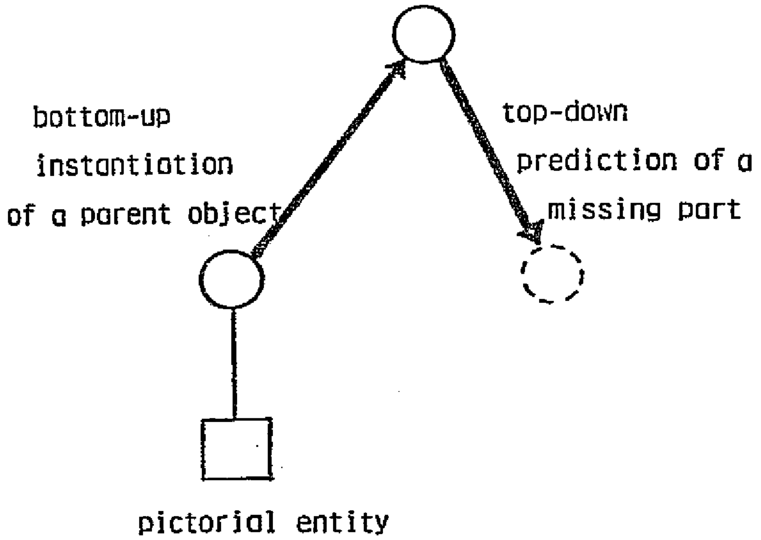


Fig. 5 Hypothesis generation by a part-whole relation

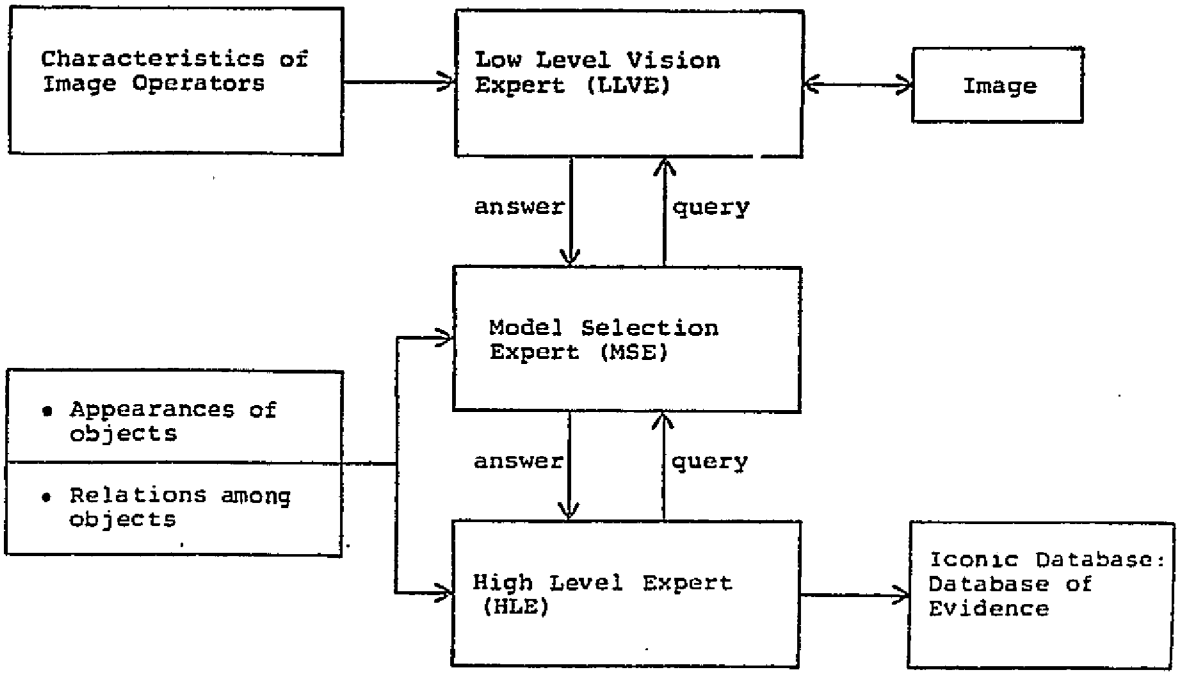
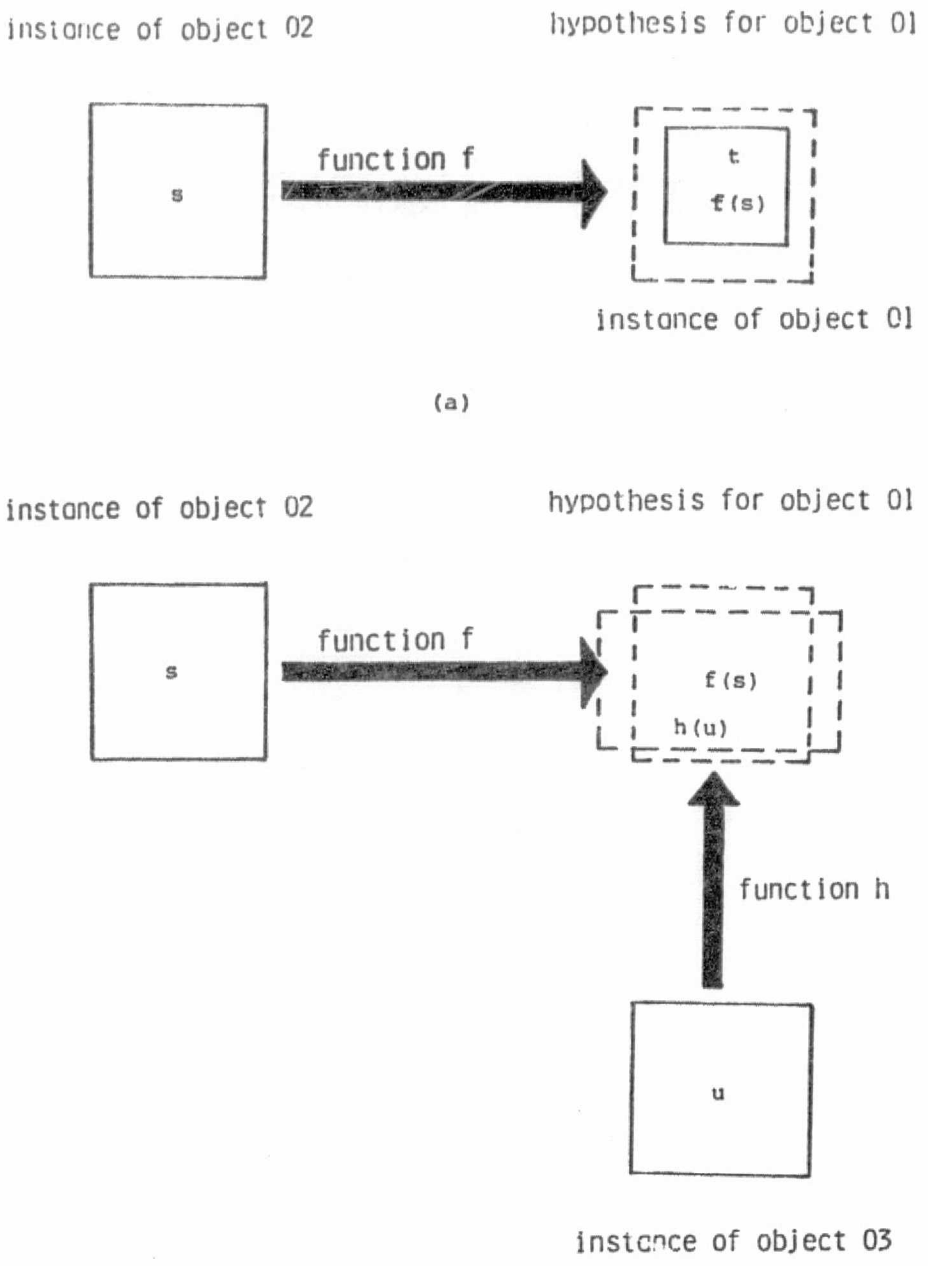

Characteristics of Image Operators
Low Level Vision Expert (LLVE)
Image
answer
query
Model Selection Expert (MSE)
• Appearances of objects
• Relations among objects
answer
query
High Level Expert (HLE)
Iconic Database: Database of Evidence

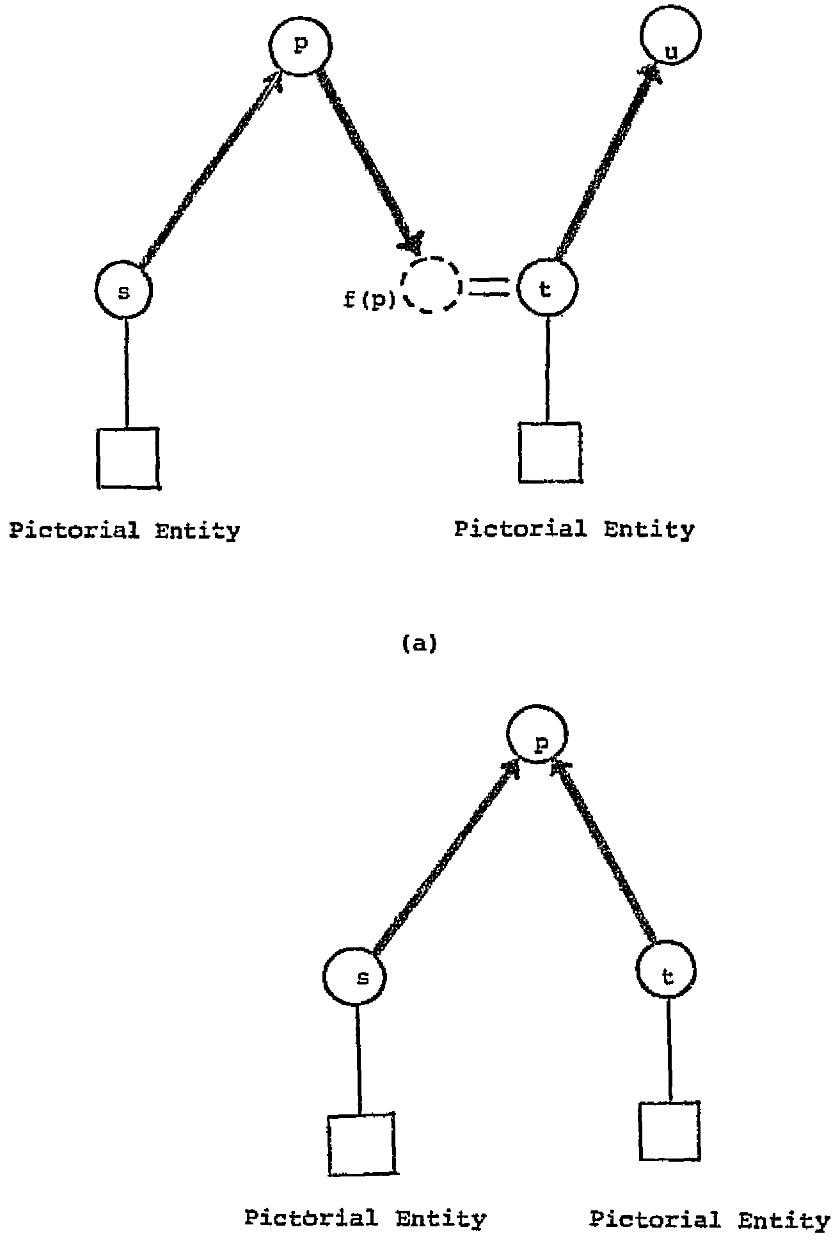
Figure 6 An Image Understanding System



(a)

(b)

Fig. 7 Principle of evidence accumulation for spatial reasoning



(a)

(b)

Fig. 8 Constructing a part-whole hierarchy

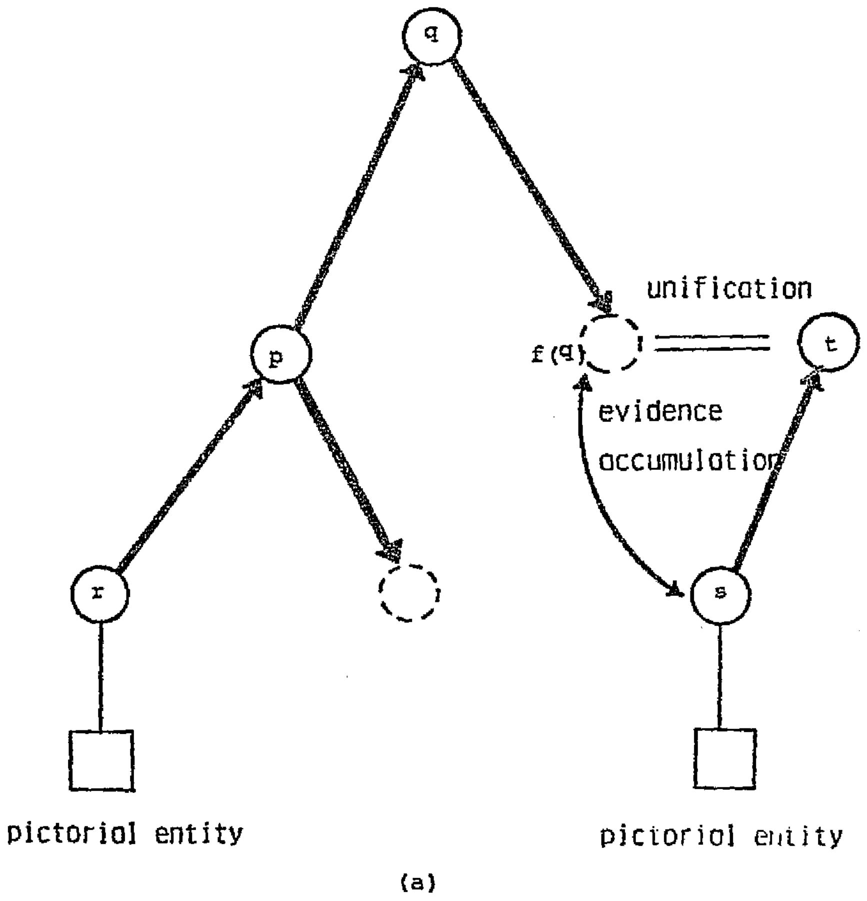


Fig. 9 (a) Another example of constructing a part-whole hierarchy

C-4

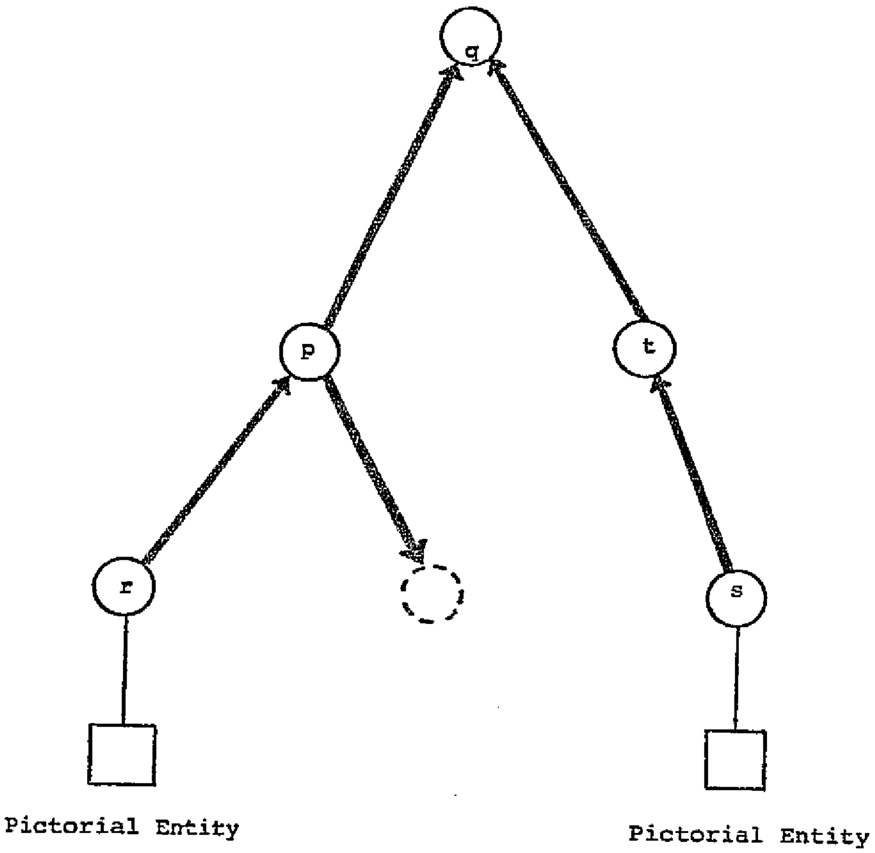


Fig. 9 (b) Result of the unification

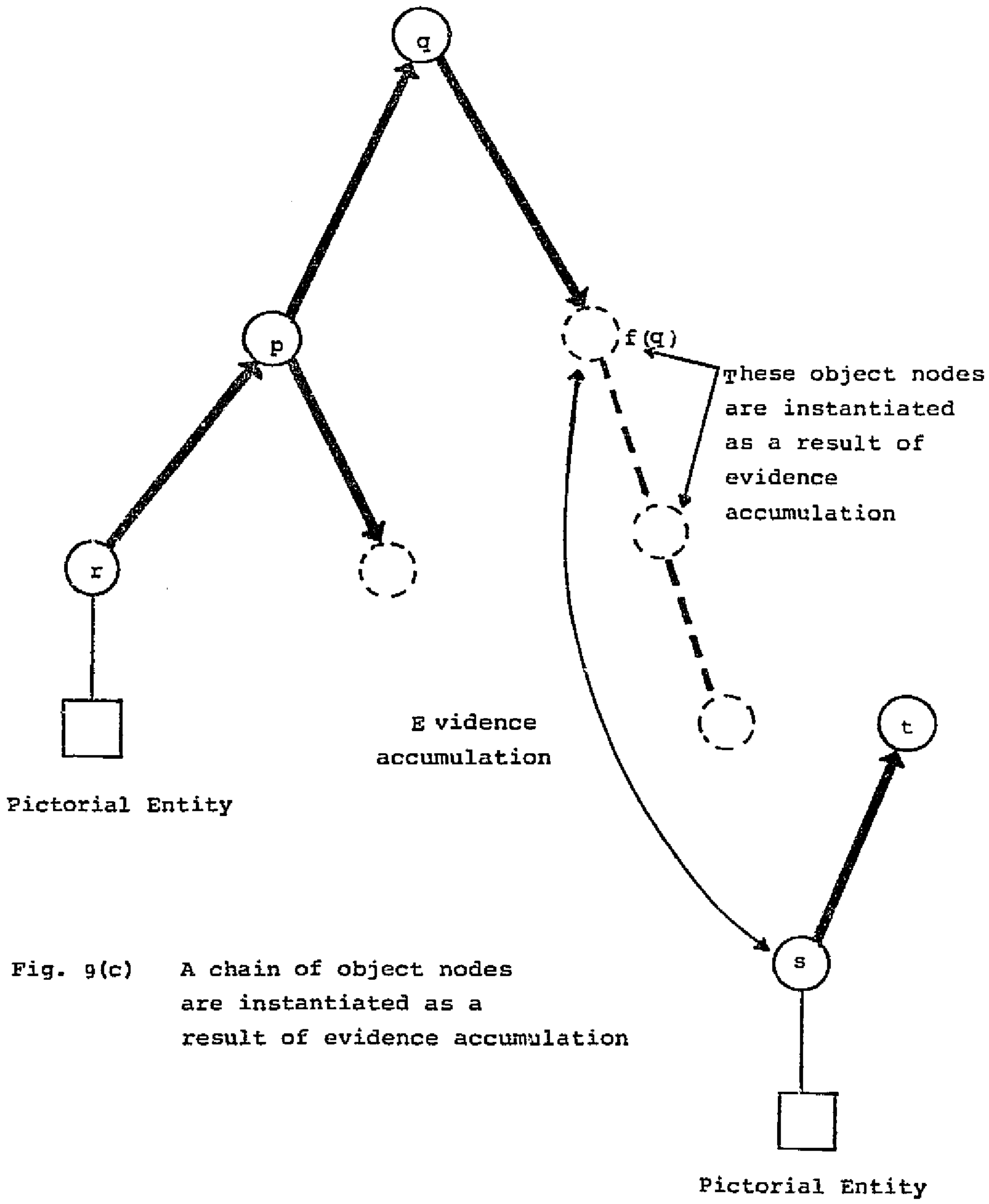
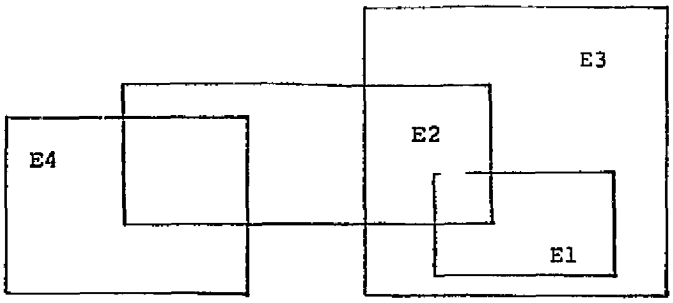
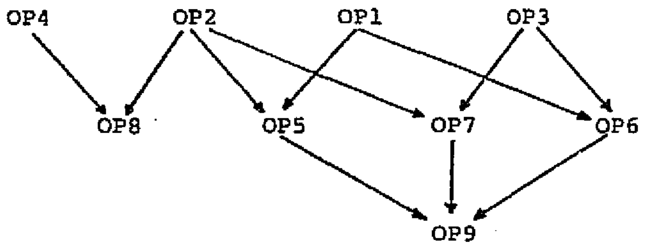


Fig. 9(c) A chain of object nodes are instantiated as a result of evidence accumulation



(a) Four overlapping pieces of evidence

Overlap	Constituent Evidence
OP1	E1
OP2	E2
OP3	E3
OP4	E4
OP5	E1, E2
OP6	E1, E3
OP7	E2, E3
OP8	E2, E4
OP9	E1, E2, E3



(b)

Fig. 10 Lattice structure to represent overlaps among pieces of evidence

```
(AND (EQUAL OBJECT-TYPE ROAD-PIECE)
     (AND (LESSP LENGTH 19)
          (GREATERP LENGTH 14))
     (AND (LESSP DIRECTION 60)
          (GREATERP DIRECTION 45)))
```

(a) The description of constraint C1.

Slot name translation table

Slot name of road-piece frame	Slot name of road frame
Length	Total-length
Width	Average-width
Direction	Average-direction

(b) Slot name translation table for the PW relation between the road frame and the road piece frame.

```
(AND (EQUAL OBJECT-TYPE ROAD)
     (AND (LESSP AVERAGE-LENGTH 19)
          (GREATERP AVERAGE-LENGTH 14))
     (AND (LESSP AVERAGE-DIRECTION 60)
          (GREATERP AVERAGE-DIRECTION 45)))
```

(c) The description of constraints C1 after translation.

Figure 11 : Translation of constraints.

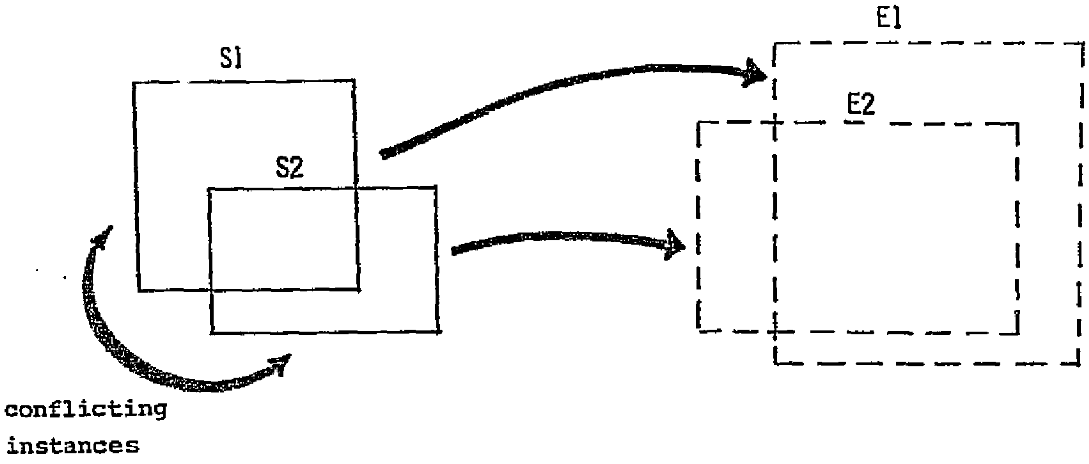


Fig. 12   Hypotheses generated by co~~n~~flicting instances

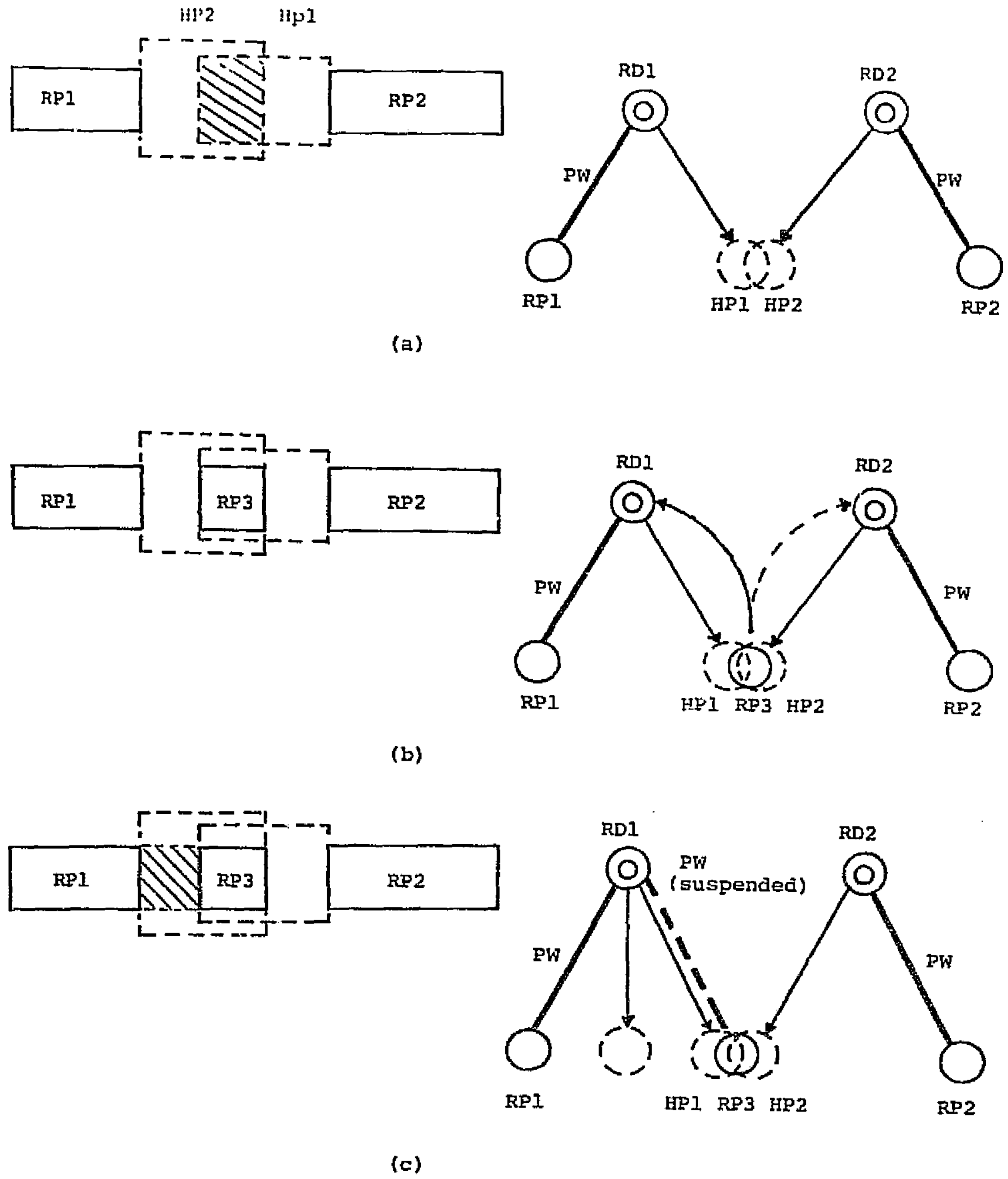


Figure 13 Resolving a situation(an example)

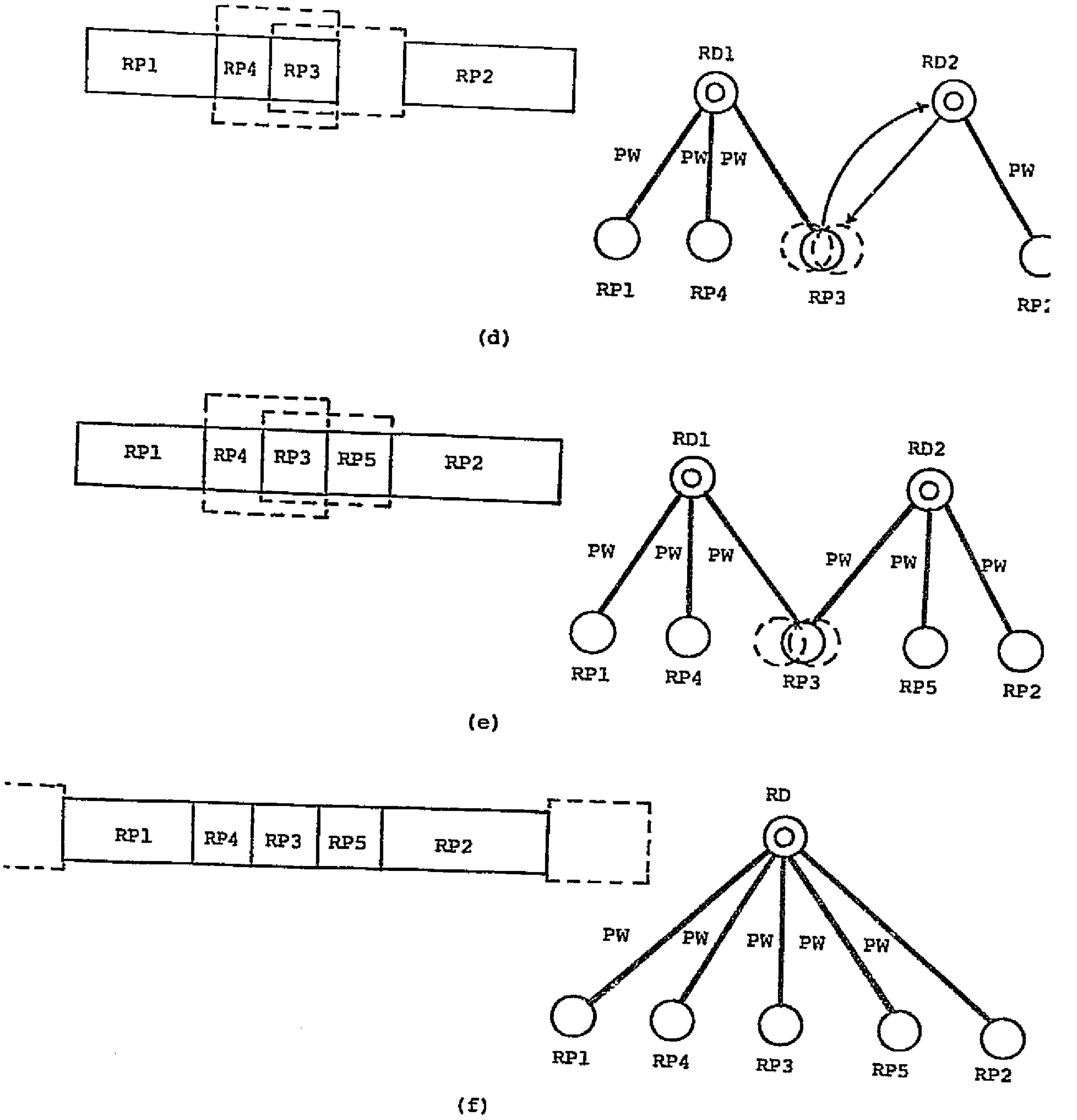
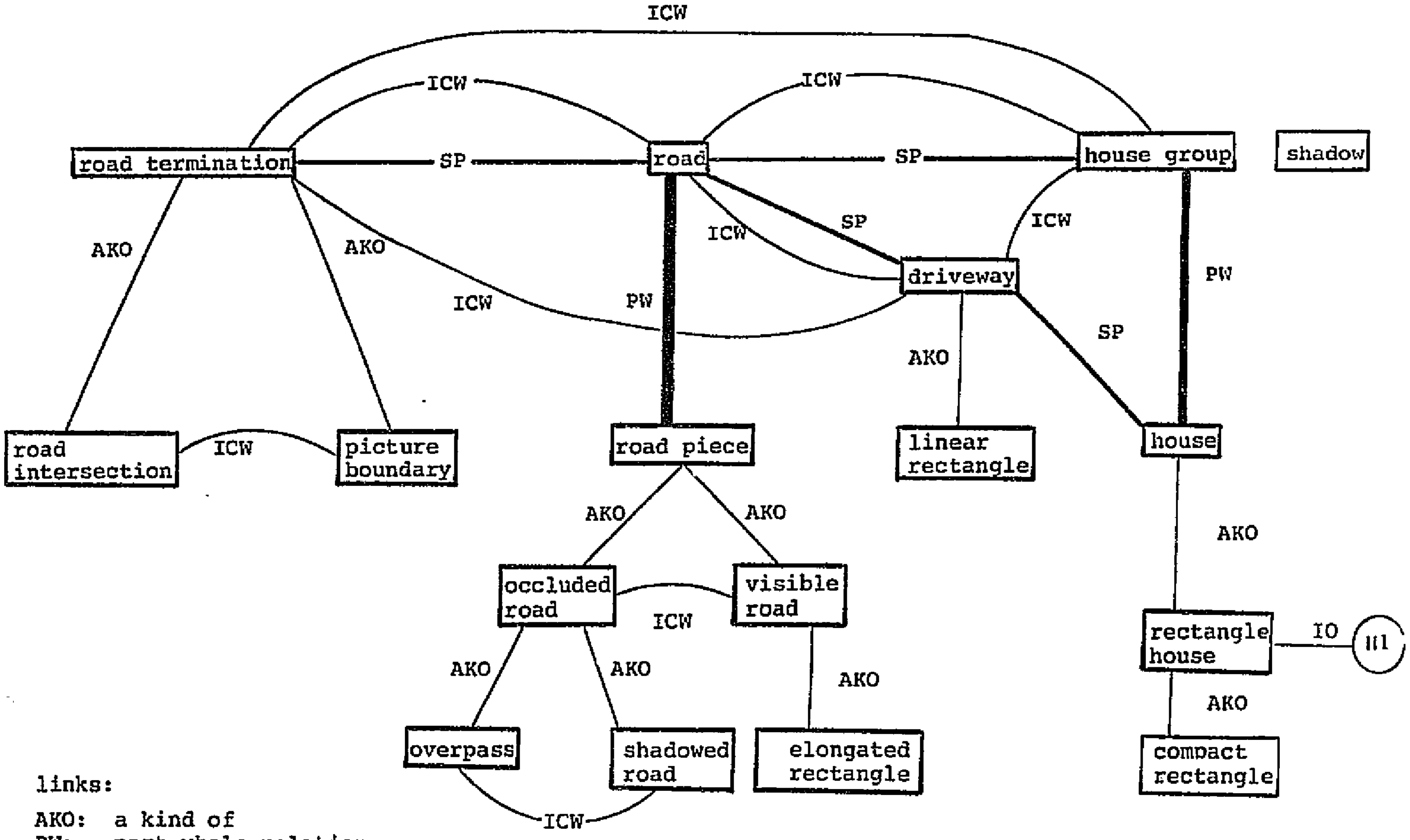


Fig. 13 Resolving a situation (see text)



links:

AKO: a kind of
PW: part-whole relation
SP: spatial relation
IO: instance of
ICW: in conflict with

Fig. 14 Knowledge organization about suburban scenes

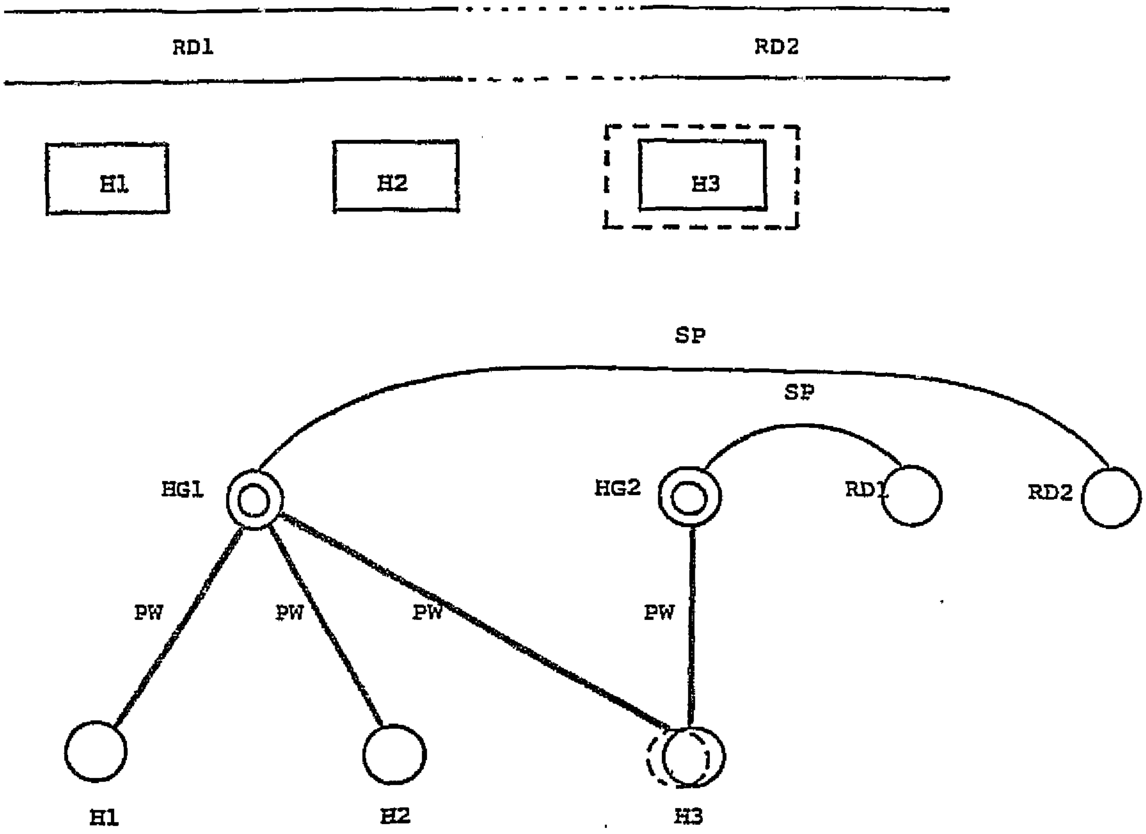


Fig. 15 Merging two partial PW hierarchies

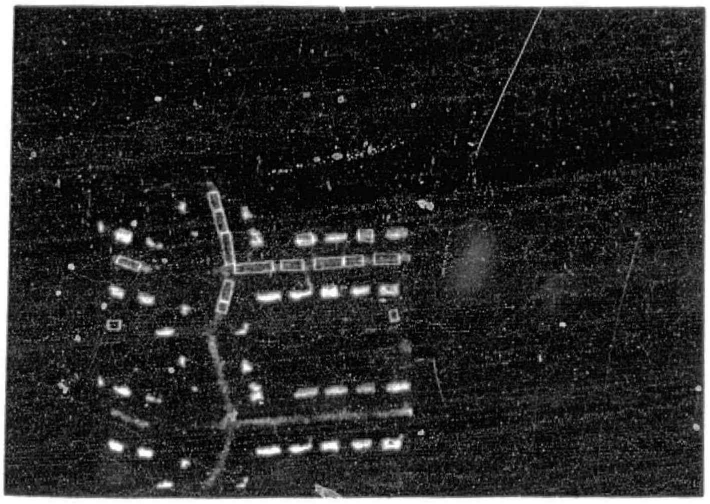
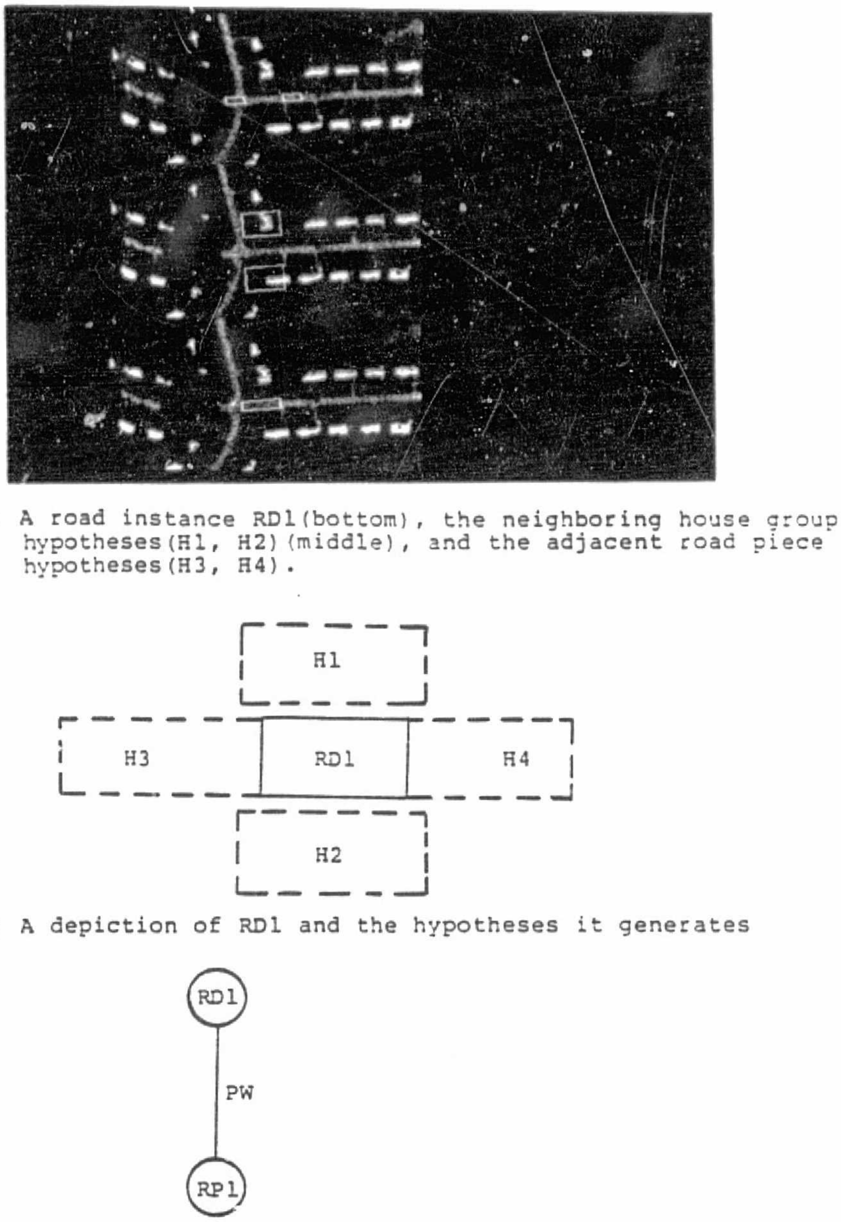


Figure 16 : Original image(bottom) and initial road piece instances overlapped on the image(top).

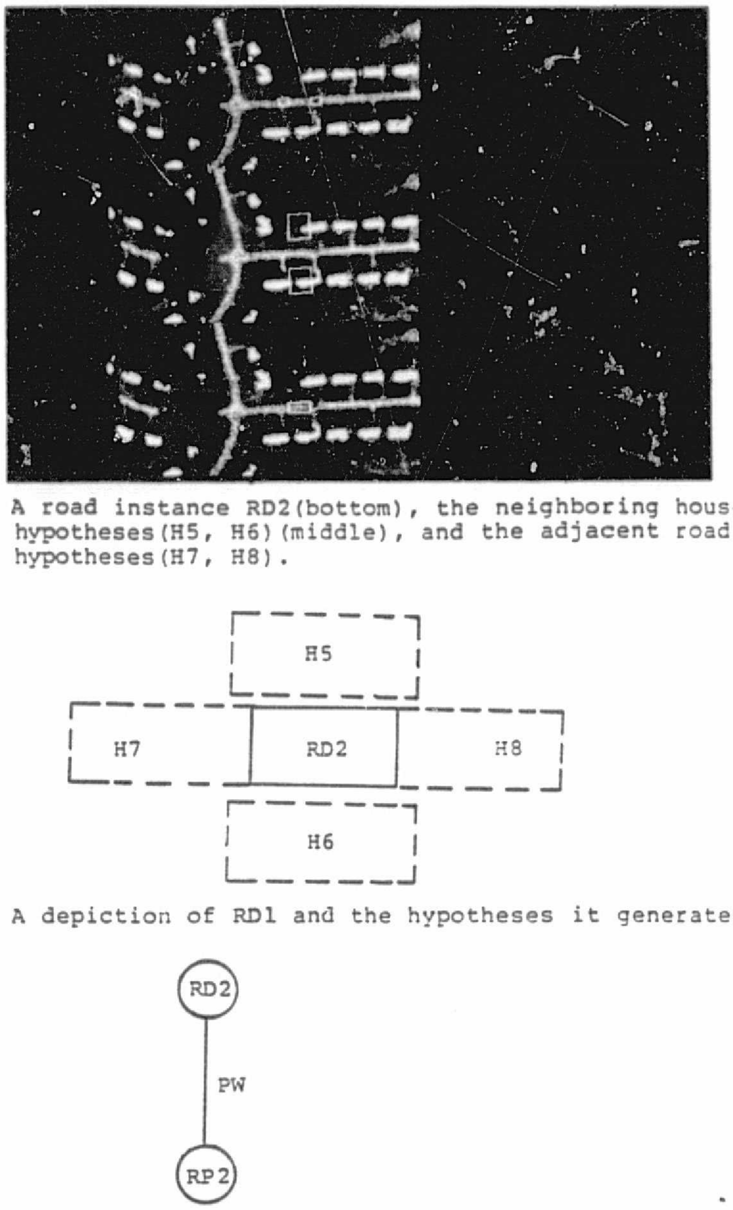


(a) A road instance RD1(bottom), the neighboring house group hypotheses(H1, H2)(middle), and the adjacent road piece hypotheses(H3, H4).

(b) A depiction of RD1 and the hypotheses it generates

(c) The interpretation graph of RD1.

Figure 17 : A road piece instance RP1, its parent RD1, and the hypotheses RD1 generates.

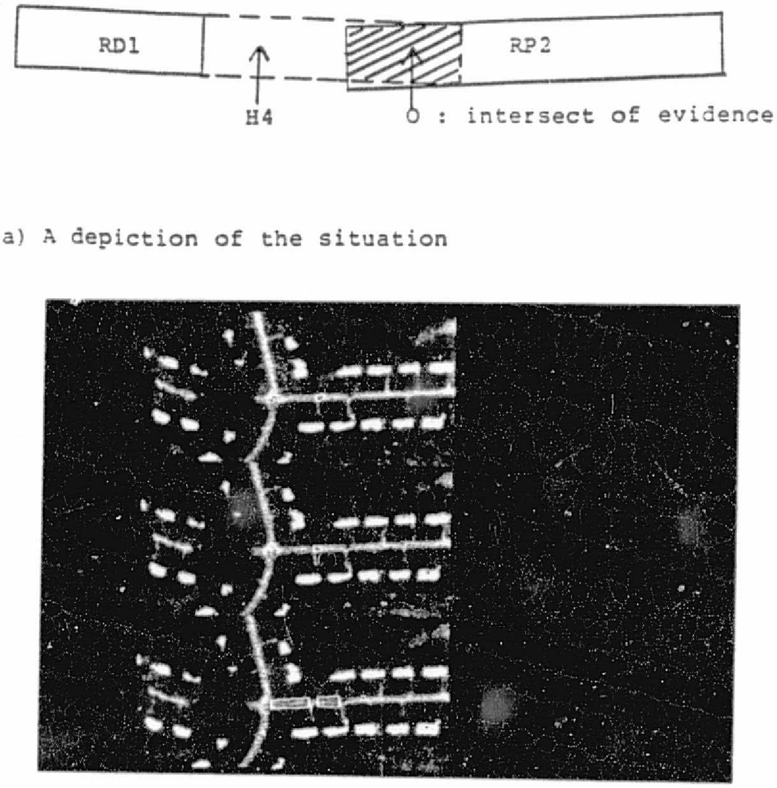


(a) A road instance RD2(bottom), the neighboring house group hypotheses(H5, H6)(middle), and the adjacent road piece hypotheses(H7, H8).

(b) A depiction of RD1 and the hypotheses it generates

(c) The interpretation graph of RD2.

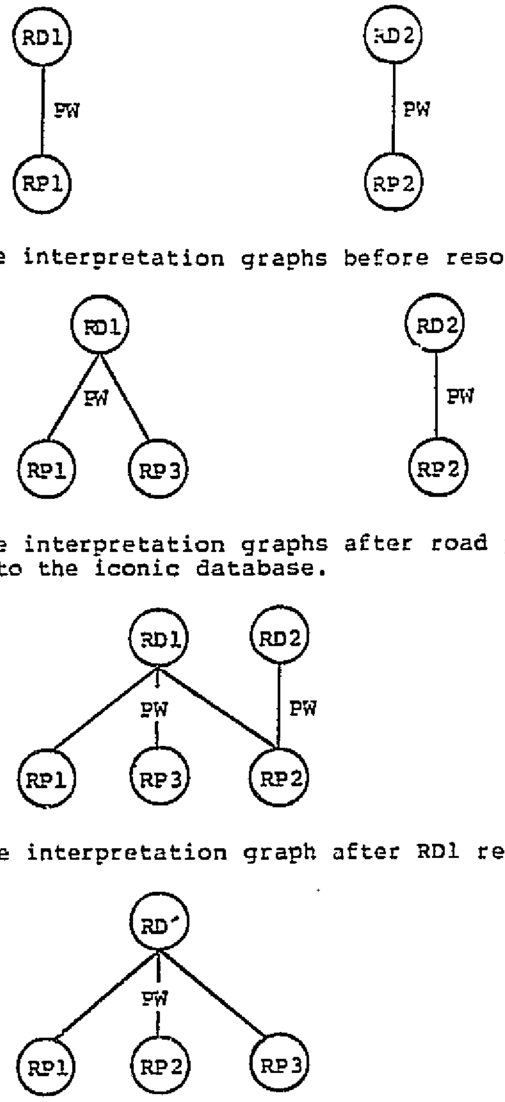
Figure 18 : A road piece instance RP2, its parent RD2, and the hypotheses RD2 generates.



(a) A depiction of the situation

(b) The supporting sources of the situation(bottom), the region of top-down prediction request(middle), the road piece instance entered from the terminal(top).

Figure 19 : A situation



(a) The interpretation graphs before resolving the situation.

(b) The interpretation graphs after road piece RP3 is entered into the iconic database.

(c) The interpretation graph after RD1 rechecks its message.

(d) The interpretation graph after the unification of RD1 and RD2.

Figure 20 : The interpretation graphs during the resolution of a situation.

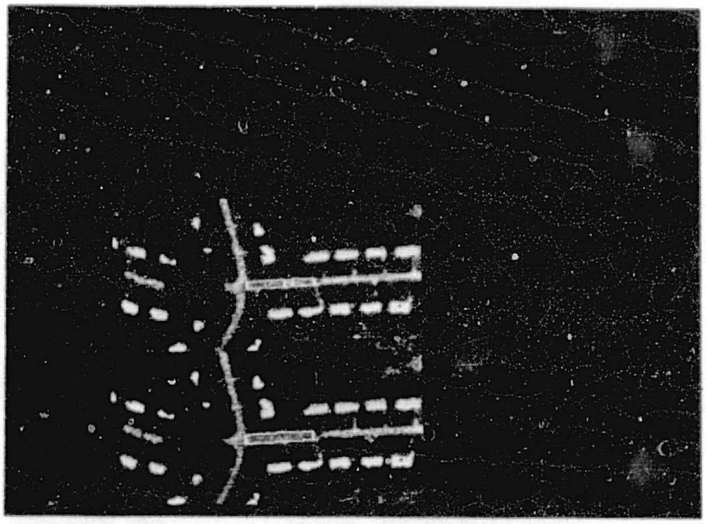


Figure 21 : Resolving a situation. Road instance RD3(bottom) and its part objects(RP1, RP2, and RP3)(top).

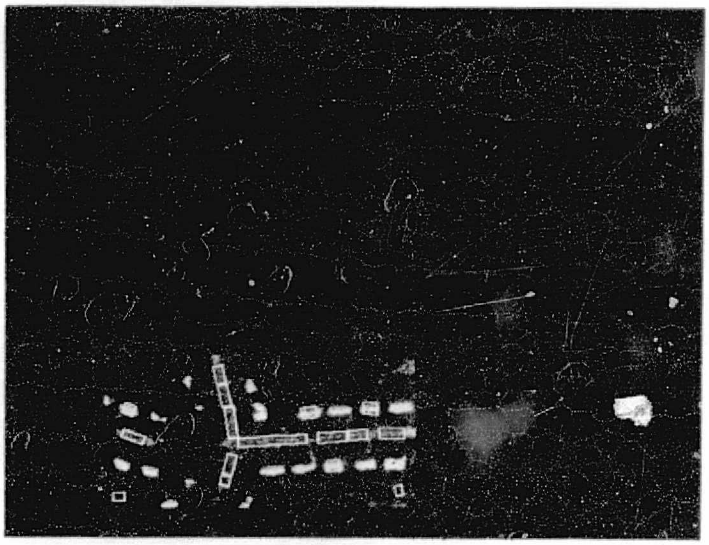


Figure 22 : All road instances after the situation is resolved.

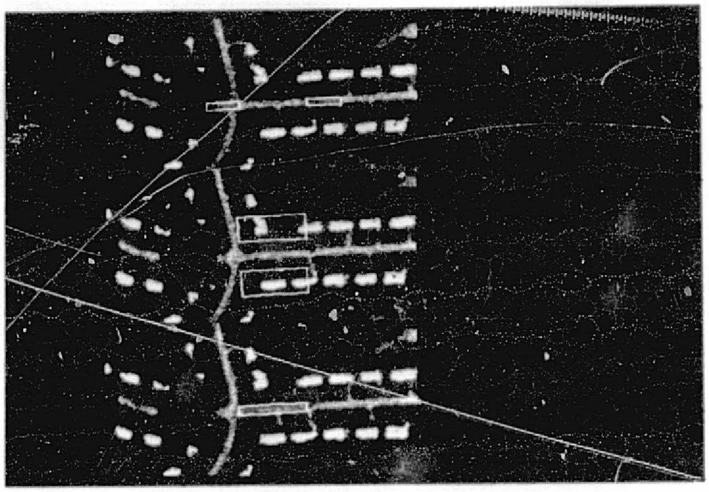
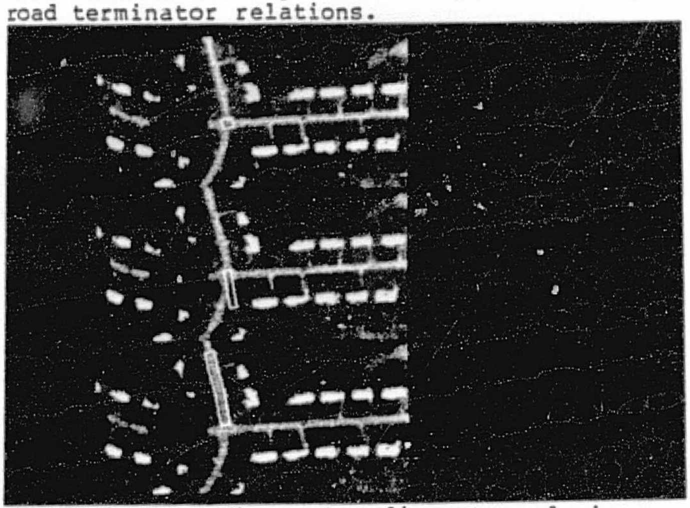


Figure 23 : Update of hypotheses : road instance RD3(bottom) neighboring house group hypotheses(middle), and adjacent road piece hypotheses(top).

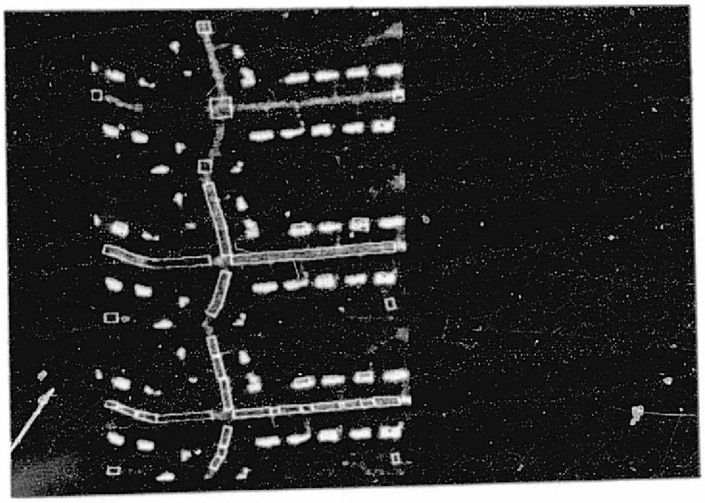
Relation	Precondition
Adjacent Road piece	Always
Adjacent Road terminator	When search for adjacent road piece has failed

(a) Precondition for adjacent road piece and adjacent road terminator relations.



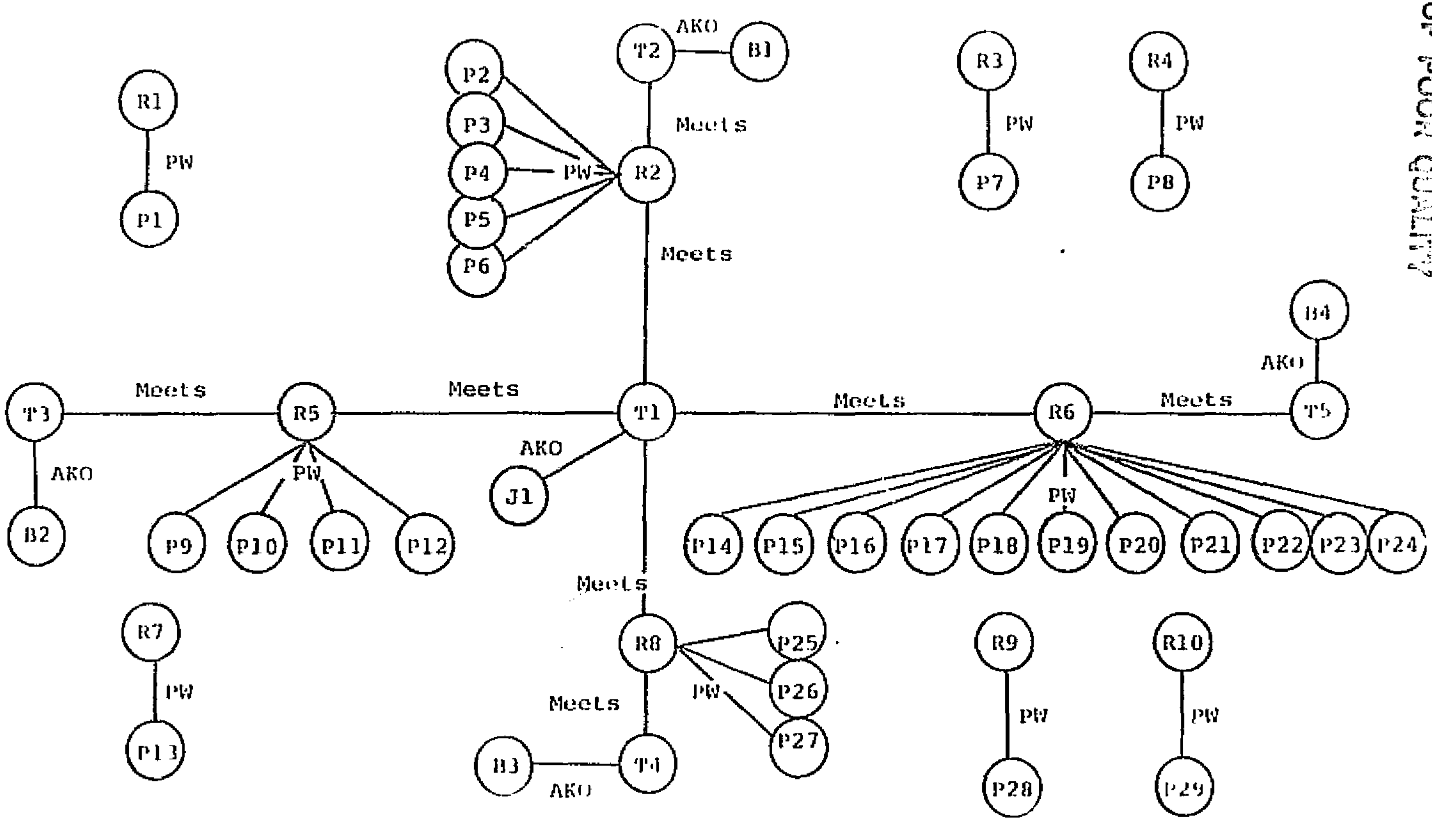
(b) A road instance(bottom), adjacent road piece hypotheses(middle), and adjacent road terminator hypothesis(top).

Figure 24 : Change of hypotheses.



(a) All road piece instances(bottom), all road instances(middle), and all road terminator instances(top).

Figure 25 : Final interpretation of the road network.



Legend : R -Road instance
P - Road piece instance
T - Road terminator instance
B - Picture boundary instance
J - Road joint instance

Figure 25 (b) The interpretation graph constructed by the system.

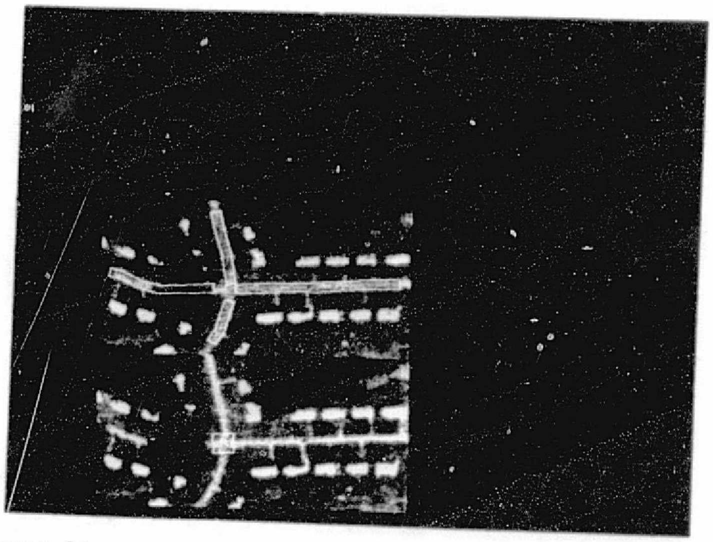


Figure 26 : Road joint instance J1(bottom) and all road instances intersecting at J1(top).

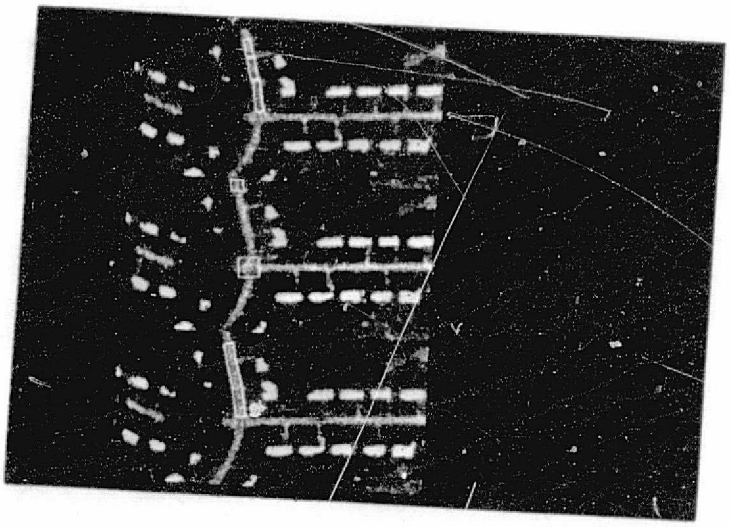


Figure 27 : Road instance R2(bottom), road terminator instance instances adjacent to it(middle), and road piece instances contained in it(top).

POWER SPECTRAL DENSITY OF MARKOV TEXTURE FIELDS

K. Sam Shanmugan and J. C. Holtzman
Telecommunications and Information Sciences Laboratory
Department of Electrical Engineering
University of Kansas
Lawrence, Kansas 66045, U.S.A.

ABSTRACT

Texture is an important image characteristic, and a variety of spatial domain techniques have been proposed for extracting and utilizing textural features for segmenting and classifying images. For the most part, these spatial domain techniques are ad hoc in nature. In this paper, we discuss a markov random field model for image texture, and derive a frequency domain description of image texture in terms of the power spectral density. This model can be used for designing optimum frequency domain filters for enhancing, restoring and segmenting images based on their textural properties.

I. INTRODUCTION

Image texture is made up of two components: a set of primitive elements and a structural arrangement [1, 2, 3, 4]. For example, in a photographic image of a residential area the primitive elements are roads, houses, and trees, and the structural arrangement is the lay out of the area. With an appropriate set of primitive elements and statistical models for the structure, it is possible to describe image texture as a markov random process, and derive a frequency

This work was supported by a NASA grant (NASA 9-16664).

domain model in the form of power spectral densities. The power spectral densities can be used to derive frequency domain algorithms for processing texture information.

For purposes of illustrating the modeling approach that can be used, consider the one dimensional textural pattern shown in Figure 1. This texture is made up of two primitive elements, rectangular and triangular in shape with the heights representing tonal variations. The location of the primitive elements is given by the sequence $\{t_i\}$. We can model this class of textural patterns as

$$X(t) = \sum_{-\infty}^{\infty} A_i \, p_{m_i} \, (t - t_i) \tag{1}$$

where

$\{A_i\}$ = Amplitude sequence,

$\{t_i\}$: Location of the i-th primitive,

$p_1, p_2, \cdots, p_N$ are N primitive elements, and

$\{m_i\}$; $m_i \in [1, 2, \cdots, N]$ indicates which one of the N primitives is present at the i-th location.

The complexity of the textural patterns and their models depend on the nature of $\{t_i\}$, $\{A_i\}$, $\{M_i\}$ and $\{p_i\}$ and the following assumptions can be made about these sequences:

1.a. $\{t_i\}$ - uniformly distributed locations or

1.b. $\{t_i\}$ - Poisson sequence with an exponential distribution for inter-location distance

2.a. $\{A_i\}$: Constant

2.b. $\{A_i\}$: i.i.d. sequence of random variables, or

2.c. $\{A_i\}$: Correlated sequence

3.a. $p_1, p_2, \cdots, p_N$: deterministic shapes, or

3.b. $p_1, p_2, \cdots, p_N$: random processes

4.a. $\{m_i\}$: Independence sequence, or
$\{m_i\}$: Homogeneous markov sequence

2. POWER SPECTRAL DENSITY OF MARKOVIAN TEXTURE FIELDS

If we assume that

$\{t_i\}$: uniformly distributed
$\{A_i\}$: constant
$p_1, p_2, \cdots, p_N$: deterministic
$\{m_i\}$: markov
$P\{\text{occurrence of } p_k\} = \pi_k$
$P(\text{occurrence of } p_i \text{ followed by } p_j \text{ after n locations}) = p_{ij}$

then the texture can be described by the markov random field (process)

$$X(t) = \sum_{-\infty}^{\infty} p_{m_i} (t - mT_s) \qquad (2)$$

where

T_s = average spacing between elements

It can be shown [5, 6] that the power spectral density of X(t) is given by

$$G_X(f) = \frac{1}{T_s^{\,2}} \sum_{n=-\infty}^{\infty} \left| \sum_{j=1}^{N} \pi_j \, S_j(\frac{n}{T_s}) \right|^2 \delta(f - \frac{n}{T_s})$$

$$+ \frac{1}{T_s} \sum_{j=1}^{N} \pi_j \, | S_j'(f)|^2$$

$$+ \frac{2}{T_s} \; \mathrm{Re} \left\{ \sum_{j=1}^{N} \sum_{k=1}^{N} \pi_j \, S_j'^{*}(f) \, S_k'(f) \, Q_{jk}(f) \right\}$$

where

$$S_k(f) = F\{p_k(t)\}$$

$$S_k'(f) = F\{p_k(t) - \sum_{j=1}^{M} \pi_j p_j(t)\}$$

and

$$Q_{jk}(f) = \sum_{n=1}^{\infty} p_{jk}(n) \exp(-i \; 2 \; \pi \; n \; fT_s)$$

By letting $t = (x, y)$, $f = (f_x, f_y)$, $p(t) = p(x, y)$, $S(f) = S(f_x, f_y)$, the model can be extended to the two dimensional case. It can also be generalized to the cases where $\{t_i\}$, $\{A_i\}$ and $\{p_i\}$ satisfy the assumptions mentioned in the previous section.

3. DISCUSSION

A generalized markov model for image texture can be derived using the formulation given in this paper. Frequency domain properties of textural patterns can be obtained from the model and the power spectral density can be used to develop frequency domain algorithms for extracting and processing textural information.

4. REFERENCES

[1] R. M. Haralick, K. Sam Shanmugan and I. Dinstein, "Textural Features for Image Classification," IEEE SMC Trans., vol. SMC-3, pp. 610, 621, 1973.

[2] R. M. Haralick, "Statistical and Structural Approaches to Texture," Proceedings of the IEEE, vol. 67, pp. 786-805, 1979.

[3] E. Wong, "Two-Dimensional Random Fields and Representation of Images," SIAM Journal of Applied Math., vol. 16, 1968.

[4] A. Rosenfeld and L. Davis, "Image Segmentation and Image Models," Proceedings of the IEEE, vol. 67, pp. 764-772, 1979.

[5] K. Sam Shanmugan, Digital and Analog Communication Systems, John Wiley and Sons, N. Y., 1979.

[6] J. H. Holmes, Spread Spectrum Systems, John Wiley and Sons, N.Y., 1982.

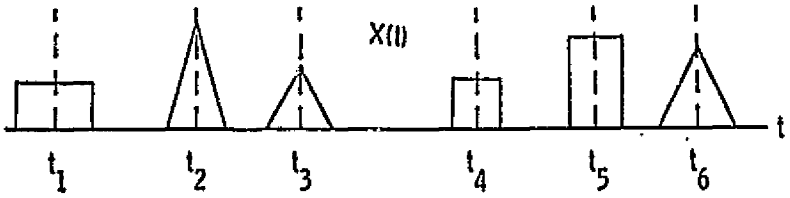


Figure 1: One dimensional textural pattern

RECTIFICATION OF SINGLE AND MULTIPLE FRAMES OF SATELLITE SCANNER IMAGERY USING POINTS AND EDGES AS CONTROL

by

Fidel C. Paderes, Jr.
Edward M. Mikhail
Wolfgang Förstner*

School of Civil Engineering
Purdue University
West Lafayette, Indiana 47907

NASA SYMPOSIUM ON
MATHEMATICAL PATTERN RECOGNITION AND IMAGE ANALYSIS

June 6-8, 1984

* Dr. Wolfgang Förstner is an Assistant Professor of Photogrammetry, Stuttgart University, West Germany; he was a Visiting Scientist at Purdue University at the time of writing this paper.

ABSTRACT

Rectification of single and overlapping multiple scanner frames is carried out using a newly developed comprehensive parametric model. Tests with both simulated and real image data have proven, that this model in general is superior to the widely used polynomial model; and that the simultaneous rectification of overlapping frames using least squares techniques yields a higher accuracy than single frame rectification due to the inclusion of tie points between the image frames. Used as control, edges or lines, which are much more likely to be found in images, can replace conventional control points and can easily be implemented into the least squares approach. An efficient algorithm for finding corresponding points in image pairs has been developed which can be used for determining tie points between image frames and thus increase the economy of the whole rectification procedure.

1. INTRODUCTION

1.1 General

Imaging, using scanners as sensors, yields the sensed data about the object in the form of pixels. Knowledge of the relative and/or absolute locations of these pixels in the object space is necessary for mapping, classification, and change detection or monitoring. Of primary interest is scanner imagery of the surface of the earth. The process of finding the location of pixels on the ground for this type of imagery is called rectification. If the reference is another image, the process is known as registration. This research covers rectification and registration of scanner imagery produced by satellite-borne scanners such as LANDSAT MSS imagery. An important element of this research concerns correspondence between two images or between an image and a representation of the terrain (i.e. a map).

If the position of the sensor platform (i.e. satellite) and the attitude of the sensor at the moment of sampling a given pixel is known, and if the interior geometry of the scanner at the same instant can be reconstructed, then the ground position of a pixel can be derived with some assumptions regarding the shape of the terrain. The satellite position can be derived from satellite tracking data. The sensor attitude can be supplied by attitude sensors on-board the satellite. The geometry of the sensor is reconstructed using calibration data and the imagery. Unfortunately, the accuracy of the satellite position and sensor attitude measurements is not sufficient to produce sub-pixel rectification accuracy.

An alternative method for rectifying satellite scanner imagery is through the use of control information. Control can be in the form of points or edges with known ground and image locations. In this method, given a suitable mathematical model, the parameters needed for relating the image positions of pixels to their ground positions are first computed using control points and applying an appropriate adjustment procedure. Then the ground positions of pixels are computed using the same model and the derived parameters. The same method, with slight modifications, can also utilize edges as control instead of points.

The above method can be further subdivided into two approaches. The first is the interpolative or the surface fitting approach. This approach uses a mathematical series (e.g., polynomial, harmonic) to approximate the true mathematical model relating the image position of pixels to their corresponding ground position. This approach requires an excessive number of control points for uniform rectification accuracy.

The second approach is commonly called "parametric". In this approach, the mathematical model used is based on the geometry of the imaging process. Because of this, it is possible to develop highly accurate models. However, usually simplifying assumptions are made to make the resulting model tractable, since the geometry of the satellite scanner image is very weak. In this approach, it is possible to exploit a-priori knowledge of the satellite position and sensor attitude, effectively combining the two main methods discussed above.

Both methods mentioned are normally used for rectifying single frames of satellite scanner imagery. This requires that some assumptions be made regarding the shape of the terrain covered by one image frame. Improvement in accuracy can be gained if overlapping frames of imagery are rectified simultaneously in a procedure commonly known as block adjustment.

1.2 Review of the Literature

The earliest approach to rectification utilized interpolative or surface fitting models such as polynomials. This model is easy to implement and gives results comparable to most early forms of the parametric model for satellite imagery (Forrest [13], Trinder [42], Bähr [1], Dowman [10]).

The parametric model based on the geometry of the scanner imaging process has many variations depending on the simplifying assumptions made. The simplest model, which is really designed for aircraft scanner data, assumes that the satellite orbit is a straight line and that the earth is projected onto a mapping plane (Kratky [23], Konecny [21], Dowman [10]). The orbit of the satellite has been assumed to be a circle (Forrest [12], Levine [26], Synder [41]) or an ellipse (Bähr [2], Sawada [39]). The earth has been assumed to be a sphere (Caron and Simon [9], Bähr [2], Sawada [39]) or an ellipsoid of revolution (Puccinelli [36], Forrest [12], Levine [26], Synder [41]).

The satellite orbit and position can be defined simultaneously in terms of the satellite position and velocity vectors (Caron and Simon [9], Puccinelli [36]). The position of a satellite along an assumed orbit can be defined in terms of time varying orbital parameters (Bähr [2]). Alternatively, the orbital parameters can be assumed constant which results in an ideal orbit. Small deviations of the actual satellite position from the ideal are then modeled as arbitrary functions of time, usually a polynomial series (Levine [26], Mikhail and Paderes [32]).

The sensor, without the scanning action, is nominally pointed along the vertical. Small deviations of the sensor attitude with respect to the vertical are modeled as polynomial functions of time. Bridging of long strips with control at each end only is feasible through the effective use of a-priori attitude information (Friedmann [16]).

1.3 Scope of Investigation

In the early phase of this research, we derived a comprehensive model considering that the earth is an ellipsoid of revolution and the orbit of the satellite is an ellipse (Mikhail and Paderes [32]). All three components of the deviation of the satellite position from the ideal and the three components of the deviation of the sensor attitude from the nominal are incorporated into the model.

Using this model, we developed a system for simulating scanner image data both in the direct and inverse modes. In the direct mode, given the parameters defining the orbit, time, satellite position deviation, sensor attitude and internal sensor geometry, and given the ground coordinate of points of interest, the corresponding image row and column numbers are derived. In the inverse mode, the ground planimetric coordinates of points are computed given the corresponding image row and column numbers, the parameters mentioned above, and the shape of the terrain.

This model has been extensively tested using simulated data and reported on in last year's Symposium (Mikhail and Paderes [32]). Five different sets of experiments were performed to study the following factors: (1) the effect of error in parameter estimates on rectification accuracy; (2) the relative performance between our extensive model, three special cases with simplifying assumptions, and the polynomial model; (3) the effect of different control densities on rectification accuracy; (4) the effect of errors in derived image position on rectification accuracy; and (5) the effect of errors in measured ground position of control points on rectification accuracy.

In Chapter 2 of this report, additional tests of this model using two frames of real data and the corresponding frames of simulated data employing the same characterizing parameters as the real data are included. Previous conclusions using purely synthetic data were generally confirmed.

With the comprehensive model fully developed and tested for the rectification of single images, effort was directed to the implementation of an extensive block adjustment program. It is based on the same mathematical model and is designed to accommodate data from overlapping satellite scanner imageries. Block adjustment reduces the required amount of control needed to meet a specified level of rectification accuracy. Synthetic data was used to verify the algorithm and the results are included in this report.

From the experience gained by analyzing both synthetic as well as real data, acceptable rectification results require from 20 to 30 control points. Securing this number of points is often difficult and costly because well identifiable "point" features are not abundant. Furthermore, high image and ground positional accuracy for control points is difficult to achieve. Therefore, the research effort was next directed toward an alternative type control. In Chapter 3 of this report, the novel concept of "edge point" is developed and tested and found to be quite promising. The idea is rather simple in that a control point can be equivalently thought of as a pair of perpendicular edges. Therefore, one edge, which may be considerably easier to find and locate accurately, can be used as control. A point on an edge, which we shall term "edge point", will have 2x2 covariance matrix which is almost singular. This is because such a point provides precise information only in the direction normal to the edge.

Having generalized somewhat the approach to control by introducing the edge points, effort is then directed to the overall problem of correspondence. Chapter 4 reviews the general problem of correspondence and develops an algorithm for locating corresponding objects in image pairs. The algorithm is based on a robust estimation procedure for the parameters of an affine transformation. It has been tested on real image data with simulated distortions, and this early result is given.

2. IMAGE RECTIFICATION

2.1 Theory

The comprehensive model we derived (Mikhail and Paderes [32]) can be used for simulating data both in the direct and inverse modes and for rectification. This model has the following form:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \lambda M \begin{bmatrix} X-X_s \\ Y-Y_s \\ Z-Z_s \end{bmatrix} \tag{1}$$

where:

x, y, z are the coordinates of a given point in the image space. These coordinates are functions of image row and column numbers and the internal sensor geometry;

X, Y, Z are the corresponding ground coordinates of the given point;

X_s, Y_s, Z_s are the ground coordinates of the satellite position when the pixel containing the given point is sampled. These coordinates are the sum of the ideal or predicted satellite position and the deviation of the actual satellite position from the predicted one. The ideal position is a function of orbital parameters and time (t) while the deviations are functions of time (t) only;

t is time which is a function of pixel row and column numbers and the internal sensor geometry;

M is an orthogonal rotation matrix which brings the ground coordinate system into the sensor coordinate system. This is a function of time, sensor attitude, deviation of the satellite position from the ideal, orbital parameters and earth geometry;

λ is a proportional constant which varies from pixel to pixel (i.e. a scale factor).

In this model, small deviations of the satellite position from the ideal (3 components) and the small sensor attitude deviations from nominal (3 components) are modeled as third degree polynomial functions of time. Usually, λ, which is a nuisance parameter, is eliminated resulting in:

$$f_1 = \frac{x}{z} - \frac{m_{11}\,(X-X_{s)} + m_{12}\,(Y-Y_{s)} + m_{13}\,(Z-Z_{s)}}{m_{31}\,(X-X_{s)} + m_{32}\,(Y-Y_{s)} + m_{33}\,(Z-Z_{s)}} = 0$$

(2)

$$f_2 = \frac{y}{z} - \frac{m_{21}\,(X-X_{s)} + m_{22}\,(Y-Y_{s)} + m_{23}\,(Z-Z_{s)}}{m_{31}\,(X-X_{s)} + m_{32}\,(Y-Y_{s)} + m_{33}\,(Z-Z_{s)}} = 0$$

These two equations, which are now in a form suitable for rectification, are then linearized with respect to four groups of variables: (1) the row and column numbers of a given point; (2) the parameters defining time and satellite orbit; (3) the parameters defining the satellite position deviation from the ideal and the sensor attitude; and (4) the ground coordinate of the corresponding point. Other variables defining the internal geometry of the sensor and the geometry of the earth's shape are held constant. Variables in the first and fourth groups vary from point to point, while variables in the second and third groups are constant throughout a whole frame. The linearized equation has the following general form:

$$A\,v + B_2\Delta_2 + B_3\Delta_3 + \ddot{B}\ddot{\Delta} = f \tag{3}$$

where:

v is a 2 element vector of residuals for the first group of variables (i.e. observed row and column numbers for a given point);

A is a 2x2 matrix of partial derivatives with respect to the first group of variables;

Δ_2 is an 8 element vector of corrections to the approximations for the second group of variables;

B_2 is a 2x8 matrix of partial derivatives with respect to the second group of variables;

Δ_3 is a 24 element vector of corrections to the approximations for the third group of variables;

B_3 is a 2x24 matrix of partial derivatives with respect to the third group of variables;

$\ddot{\Delta}$ is a 3 element vector of corrections to the approximations for the fourth group of variables (i.e., ground coordinates);

$\ddot{B}$ is a 2x3 matrix of partial derivatives with respect to the fourth group of variables;

f is a 2 element vector of constants resulting from the linearization.

The first and fourth group of variables in the linearization are known because they are supplied by ground control points. In rectification, the values of the unknown parameters in the second and third group of variables are recovered in an adjustment procedure using control points and the linearized model shown in equation (3).

Because of weak satellite scanner image geometry, not all the unknown parameters can be solved for simultaneously. Instead, unknown parameters in the second group of variables are first recovered under the assumption that all parameters in the third group are zero. This is reasonable since the model is designed such that the parameters in the third group are as close to zero as possible. Then, using the same set of control points and the computed values of the parameters in the second group, estimates of the parameters in the third group are derived. Once estimates of all unknown parameters are available, the ground coordinates of any other image point can be solved for with some assumptions regarding the shape of the terrain.

2.2 Experiments With Real Single Frame Data

Two MSS frames taken by LANDSAT 2 are used in this experiment. The first frame covers Kansas State which is relatively hilly. It has 153 uniformly distributed control points. The second frame principally covers the state of Louisiana which is flat. About 1/3 of this frame on the south-east corner is over the sea. It has 192 well distributed control points, although not as uniformly as in the Kansas frame.

Ten cases were run for each frame corresponding to two types of model (collinearity and polynomial) and five control configurations. For each case, withheld control points were used as check points. Table 1 shows the results. The collinearity model is superior to the polynomial model when the control points are few especially in hilly terrain such as the Kansas frame. Also, increasing the number of control points beyond 25 has only a marginal effect on rectification accuracy. This confirms in general our previous results using simulated data (Mikhail and Paderes [32]). Two additional cases for each frame were also run where all the control points were exercised in the adjustment. The RMS of the residuals on control points for the Kansas frame were 58.8 and 57.8 m for the collinearity and polynomial models, respectively. The corresponding values for the Louisiana frame were 61.2 and 60.1 m. These values are the upper bounds of the quality of the data. They are used in the second experiment to determine the precision of the image measurements input into the simulation.

2.3 Experiments With Single Frame Synthetic Data

Using our extensive simulation program, the characteristics of the two real image frames were used to produce simulated images which reproduce as closely as possible the real images with respect to control configuration and accuracy. Simulation was done in the inverse mode, where perfect ground coordinates are calculated from the

Table 1 RMS Error on Check Points in Meters Using Real Data

Number of Control Points	Kansas		Louisiana	
	Collinearity	Polynomial	Collinearity	Polynomial
10*	68.8	117.1	90.4	96.6
15*	67.9	73.6	72.3	71.7
25	67.6	70.4	69.3	67.3
40	67.9	69.5	66.0	65.4
81/70**	63.8	65.5	68.4	68.4

* When the number of control points is low, the number of parameters in the model is reduced to avoid convergence problems.

** 81 control points for Kansas frame and 70 for Louisiana frame.

given image coordinates and derived rectification parameters. Then the calculated ground position of control points for both frames were perturbed using normal distribution with 15 m standard deviation in each of the three coordinates. The image positions were perturbed using a combination of normal and uniform distribution. The uniform distribution used for perturbing both frames has a range of -0.5 to +0.5 pixel, and is used to account for round off errors. The normal distribution used for perturbing the Kansas frame has standard deviations of 0.44 pixel in row and 0.40 pixel in column direction. These are the values which when used in the simulation program produced the RMS values given at the end of the preceding section for the full-control case. The corresponding standard deviations for the Louisiana frame were 0.40 pixel in row and 0.64 pixel in column direction. Several sets of simulated data with the described perturbations but with different "seeds" in the random number generator were produced and rectified. Table 2 shows the results of rectification using a representative simulated data set. Comparing Tables 1 and 2, it can be seen that the trends in Table 1 which resulted from rectification of real data are duplicated in Table 2.

Simulated data using the control configuration of the two real data frames but without perturbations were produced (i.e. perfect data sets). The rectification results using this perfect data set are shown in Table 3. From this table, two significant results can be seen. First, it is possible to recover the correct set of exterior orientation

Table 2 RMS Error on Check Points in Meters Using Simulated Data

Number of Control Points	Kansas		Louisiana	
	Collinearity	Polynomial	Collinearity	Polynomial
10*	84.0	134.4	80.9	89.9
15*	76.9	82.0	78.7	79.6
25	75.4	74.8	72.5	73.8
40	64.6	64.6	65.0	64.8
81/70**	61.9	62.9	60.5	61.0

* When the number of control points is low, the number of parameters in the model,is reduced to avoid convergence problems.

** 81 control points for Kansas frame and 70 for Louisiana frame.

Table 3 RMS Error in Check Points in Meters Using Perfect Data

Number of Control Points	Kansas		Louisiana	
	Collinearity	Polynomial	Collinearity	Polynomial
10*	11.8	102.5	10.9	15.4
15*	0.6	13.2	0.3	11.2
25	0.5	10.8	0.3	9.6
40	0.5	10.4	0.3	9.6
81/70**	0.5	9.9	0.3	9.8

* When the number of control points is low, the number of parameters in the model is reduced to avoid convergence problems.

** 81 control points for Kansas frame and 70 for Louisiana frame.

elements using the collinearity model if the data is perfect. Second, and more importantly, it shows that the systematic error inherent in the polynomial model is about 10 meters.

2.4 Theory of Block Adjustment

Given overlapping strips of scanner imagery, instead of performing rectification frame by frame, all frames can be rectified simultaneously in one block adjustment. The main advantage of this approach is that conventional points and edge points common to many frames, even those with unknown ground coordinates, can be exploited to increase rectification accuracy. These points are known as tie points. Another advantage in using this method is that mosaicking of large areas is facilitated.

We implemented a block adjustment procedure for satellite scanner imagery utilizing the same mathematical model used for simple frame rectification. In block adjustment, each point appearing in any frame results in a pair of equations similar to equation (3). This is the linearized form of the mathematical model used for single frame rectification.

Using the method of least squares adjustment (Mikhail [34]), the resulting system of normal equation is of the form:

(4) $$\begin{bmatrix} \dot{N} & \overline{N} \\ \overline{N}^T & \ddot{N} \end{bmatrix} \begin{bmatrix} \dot{\Delta} \\ \ddot{\Delta} \end{bmatrix} = \begin{bmatrix} \dot{t} \\ \ddot{t} \end{bmatrix}$$

where:

$\dot{N}$, $\ddot{N}$, $\overline{N}$ are submatrices of the normal equations coefficient matrix;

$\dot{\Delta}$ is a vector of corrections to the approximations for the unknown parameters in all frames (i.e., Δ_2 and Δ_3);

$\ddot{\Delta}$ is a vector of corrections to the approximations for the ground coordinates of all points;

$\dot{t}$ and $\ddot{t}$ are the resulting constant vectors.

As an example, consider the block of overlapping imagery shown in Figure 1. There are 5 image strips overlapping by approximately 60%. Every strip has 4 frames of imagery and every frame has 9 points in it. The frames are numbered consecutively in the vertical direction along the direction of the strips. The detailed form of the normal equations coefficient matrix is shown in Figure 2.

The contribution to the normal equations of the coordinates of ground points ($\ddot{\Delta}$) are usually eliminated first, resulting in a set of reduced normal equations, which has the form:

(5) $$N\dot{\Delta} = t$$

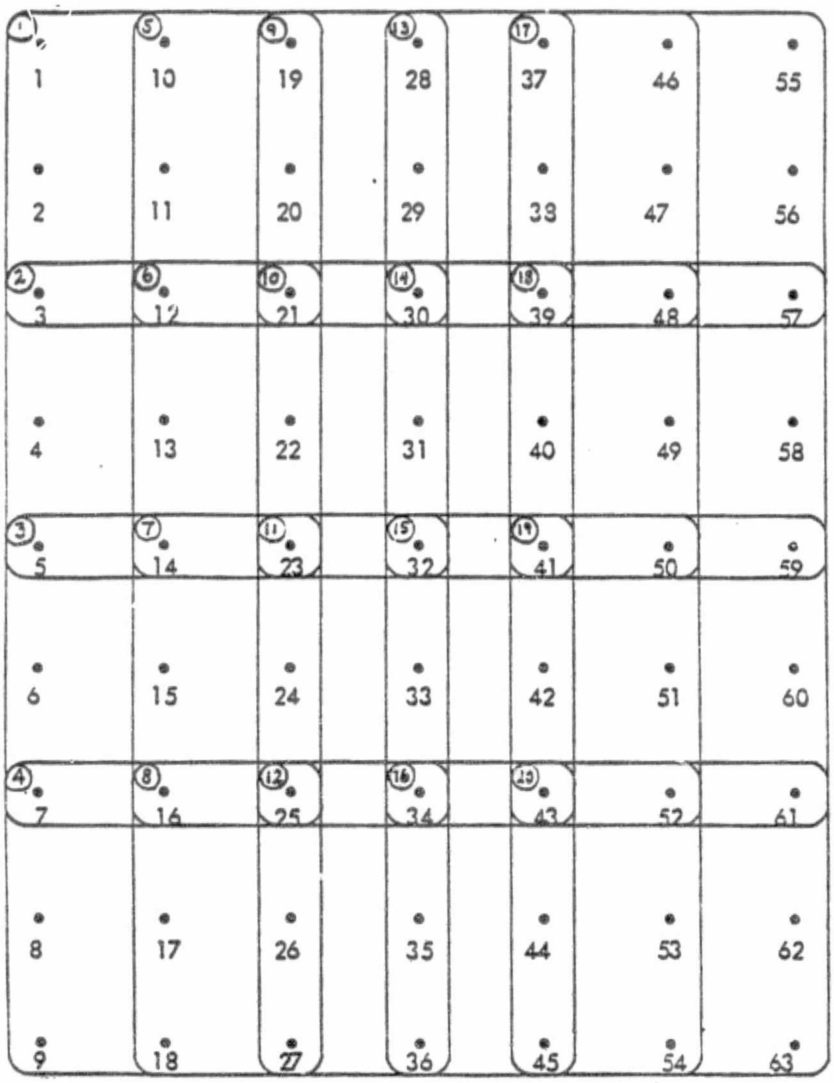


Fıgure 1 A Block of 5 Strips With 4 Frames Per Strip.

NOTE: (1) the number in circles are the frame numbers
(2) the dots represent common points and the numbers below them are the point numbers
(3) strips are in the vertical direction

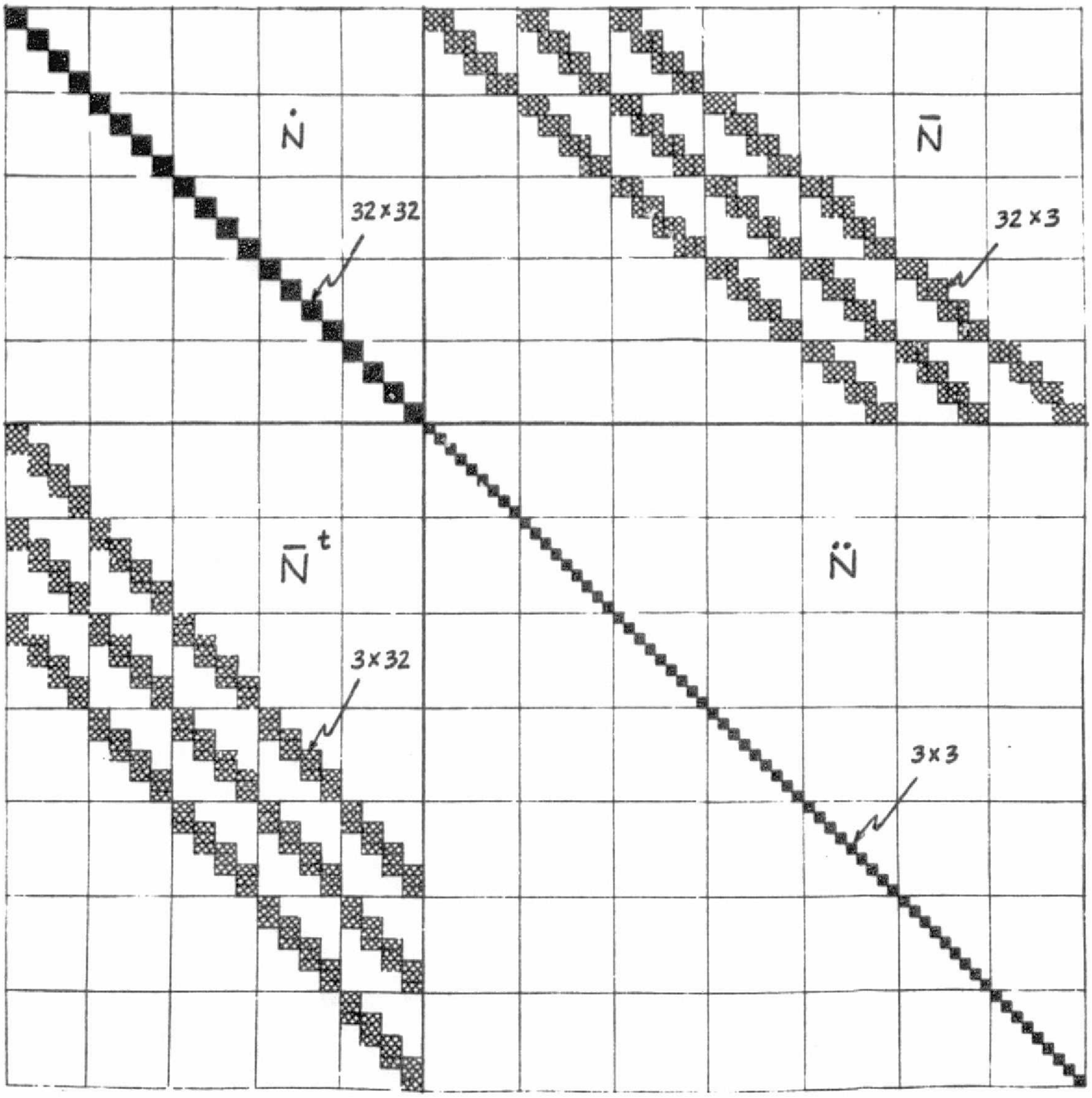


Figure 2 Detailed Form of the Normal Equations Coefficient Matrix.

where:

$$N = \quad \dot{N} - \overline{N}\ddot{N}^{-1}\overline{N}^{T}$$

$$t = \quad \dot{t} - \overline{N}\ddot{N}^{-1}\ddot{t}$$

The reduced normal equations can be formed directly without having to form the total normal equation. Proper numbering of frames results in a banded structure for the reduced normal equation coefficient matrix N. For the block shown in Figure 1, the detailed structure of the reduced normal coefficient matrix N is shown in Figure 3. Each off-diagonal sub-block in Figure 3 is due to points common to a given frame pair. The existence of these subblocks is the reason why block adjustment is more efficient than single frame rectification. As a matter of fact, block adjustment without tie points is equivalent to multiple single frame rectification. Efficient algorithms exist to solve for $\dot{\Delta}$ in equation (5).

2.5 Experiments With A Block of Overlapping Synthetic Image Data

A block of a total of 9 frames, composed of 3 adjacent strips and 3 frames per strip were simulated. The center of the block is approximately at 58.5°N latitude. The frames have about 60% sidelap between strips and 15% overlap along each strip. There are 454 control points at a grid interval of 20 km, and 453 check points also at a grid interval of 20 km. The check point grid is displaced by 10 km in both Easting and

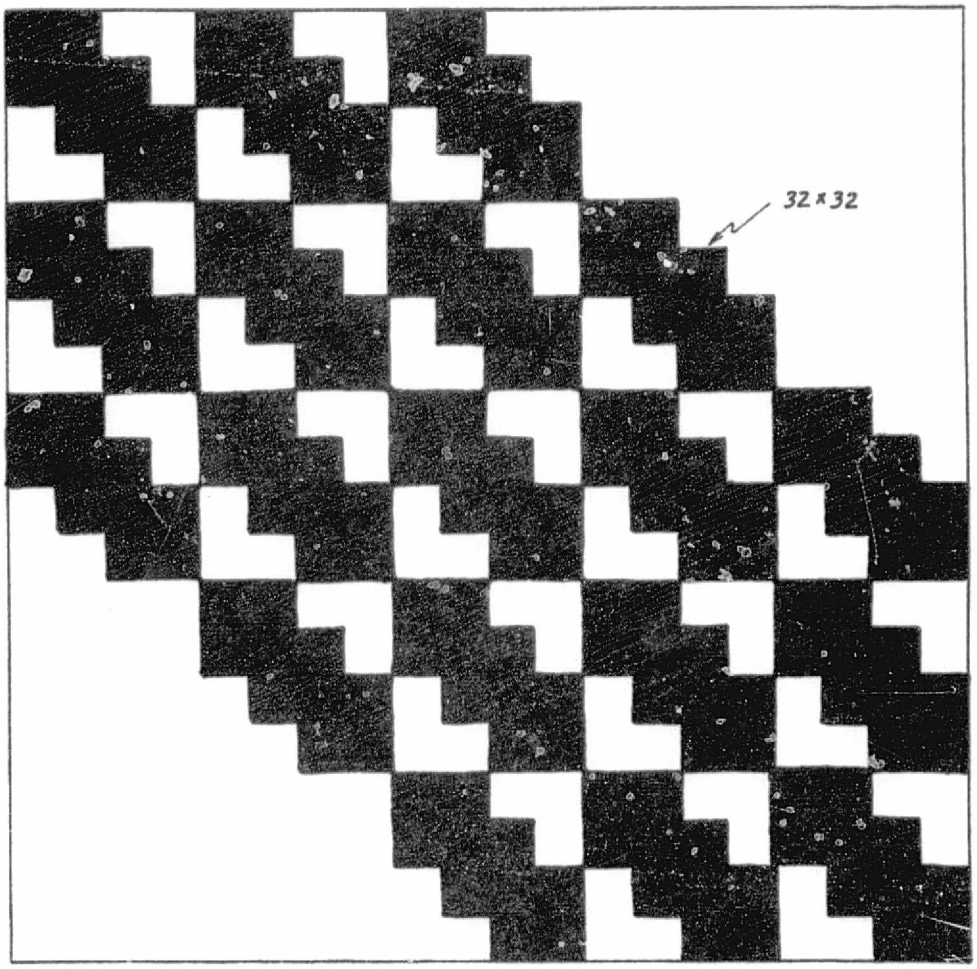


Figure 3 Detailed Form of the Reduced Normal Equations Coefficient Matrix.

Northing with the result that each control point is surrounded by 4 check points and vice versa. The ground position of both sets of points were perturbed by 15 m standard deviation in each of the three coordinate directions using the normal distribution. The image position of both sets were also perturbed using a combination of uniform and normal distribution. The uniform distribution has a range of -0.5 to +0.5 pixel. The normal distribution has a standard deviation of 0.5 pixel in both row and column direction. Five cases of block adjustment were run with different control configuration. Table 4 shows the number of control and check points for each frame and for the whole block for each of the 5 cases. It also shows the number of tie points in the block for all cases. A tie point is any point common to two or more image frames which has known image positions but unknown ground position and is included in the block adjustment. In this experiment, the ground elevation of tie points were constrained to its a-priori value. This is necessary because it was previously shown that elevations cannot be recovered with sufficient accuracy using block adjustment techniques for aircraft scanner data (McGlone and Mikhail [30]) and aircraft scanner imagery has a much stronger geometry compared to satellite scanner imagery.

Table 5 shows a relative comparison of RMS errors on check points on a frame by frame basis between block adjustment and single frame rectification for all five cases. The case where the parameters are perfectly known is included as a reference. It

Table 4 Number of Control, Tie, and Check Points Used in Block Adjustment Experiments

Cases Frames	Number of Control Points					Number of Check Points
	1	2	3	4	5	
1	11	15	27	45	91	90
2	9	13	24	39	91	89
3	9	13	26	41	88	87
4	11	15	25	45	89	86
5	11	15	26	42	89	86
6	10	14	26	41	90	86
7	10	14	25	42	88	88
8	10	15	26	44	89	87
9	11	16	26	42	85	88
Block *	42/224	66/212	125/180	214/134	454/0	453

* Control points/tie points.

Table 5 A Comparison of Check Point RMS Error Between Block Adjustment (σ_{BA}) and Single Frame Rectification (σ_{SF}).

Cases Frames	The Ratio σ_{BA}/σ_{SF} in Meters					Perfect Parameters
	1*	2*	3	4	5**	
1	93/-	79/92	66/76	67/70	66/66	65
2	77/-	76/-	68/79	74/80	69/69	62
3	117/-	100/-	73/81	80/79	79/79	68
4	87/-	77/98	65/73	67/66	65/65	63
5	76/-	74/142	67/73	70/72	68/68	64
6	79/-	74/142	63/69	69/70	63/63	62
7	113/-	70/85	65/72	65/68	65/65	59
8	92/-	97/-	64/81	69/76	68/68	60
9	83/-	72/82	65/78	67/69	68/68	62
Ave.	90.8/-	79.9/-	66.2/75.8	69.8/72.2	67.9/67.9	62.8

* Single frame rectification did not converge because of few control points (no model parameter reduction is exercised in this case).

** Block adjustment for case 5 is the same as single frame rectification because there are no tie points.

clearly shows that tie points, which are much more readily available (and less expensive) than control points, have a beneficial effect on rectification accuracy especially when control points are few. This improvement in accuracy is essentially due to tie points because block adjustment without tie points is equivalent to single frame rectification.

3. EDGES AS CONTROL

3.1 Edge Points

For a typical image frame, the necessary number of control points with the desired distribution and accuracy is difficult and sometimes impossible to secure because features that can serve as control points are few. By comparison, edges and lines occur more often and in combination with points, the necessary amount of control can more easily be satisfied if a method is devised that can utilize lines and edges as control.

A straight edge or line can be represented by a single point on that edge, preferably near the middle, and a direction. We call that point an *edge point.* Edge points on the ground, or maps representing the ground, can be identified and transferred into the corresponding image manually. The position of edge points on the image can then be measured in a direction perpendicular to the edge with an accuracy comparable to conventional points or even better. The covariance matrix for the position of the edge point in the (l,p) coordinate system is

$$\Sigma_{\mathrm{lp}} = \begin{bmatrix} (\kappa\ \sigma_{\mathrm{p}})^2 & 0 \\ 0 & \sigma_{\mathrm{p}}^2 \end{bmatrix} \tag{6}$$

where:

l is parallel to the line

p is perpendicular to the line

σ_p is the standard deviation of edge point position perpendicular to the edge

σ_l is the standard deviation of edge point position along the edge

κ is equal to σ_l/σ_p and is assigned a very large value

The direction of the edge, θ, can also be measured on the image. The covariance matrix of the edge point in the (r,c) coordinate system is

$$\Sigma_{rc} = R_\theta \, \Sigma_{lp} \, R_\theta^t \tag{7}$$

where:

r is the row direction in the image

c is the column direction in the image

R_θ is the rotation matrix with argument θ

Another method of finding the edge point on the image is through the use of digital correlation. First a window centered on the edge point on the map is digitized approximately in the row-column direction of the image. This window is then correlated with the image, with or without image pre-processing such as edge detection, resulting in image position of the edge point. The corresponding position covariance matrix can then be computed using (Forstner [14]):

(8) $$\Sigma_{rc} = \sigma_n^2 \begin{bmatrix} \Sigma\,(\partial g/\partial r)^2 & \Sigma\,(\partial g/\partial r)\,(\partial g/\partial c) \\ \text{sym.} & \Sigma\,(\partial g/\partial c)^2 \end{bmatrix}^{-1}$$

where:

σ_n is the standard deviation of image noise

g is the density of the image inside a window containing the edge point

r, c are the row and column numbers

$\partial g/\partial r$, $\partial g/\partial c$ are the partial derivatives of g with respect to r and c.

Before the location of edge points are transferred into the image, their locations are first defined in the map or ground, hence edge points can be treated as ordinary points as far as their ground positions are concerned. Once their image postions are defined, edge points can be easily incorporated into existing rectification programs.

In theory a single edge point is enough to represent a straight edge segment, but in practice more than one point may be necessary, especially if the segment is not really straight.

3.2 Experiments With Edge Points as Control In Single Frame Rectification

In our experiments using edges as control for rectification, we ran ten cases with different edge distributions. Figure 4 shows a schematic representation of all ten cases. In case (1), edge point pairs have the same coordinates and are positioned at a regular grid. The angle between the edges in an edge pair is fixed at 90°. Case (2) is the same as case (1), except that the acute angle between the edges in an edge pair varies randomly within the range 60° to 90°. Case (3) is the same as case (2), except that the range for this case is from 30° to 90°. Case (4) is the same as the previous cases except that the direction of edges in this case is totally arbitrary. Cases (5) to (8) are the same as in cases (1) to (4), respectively, except that the position of one edge point in an edge pair is randomly perturbed within the range -100 to +100 pixel. Case (9) is the same as in case (1) except that the position of each edge pair is now randomly distributed over the whole image frame. Case (10) is the most general case. In this case both the position of the edge points and the direction of edges are totally arbitrary.

The amount of contamination applied to all ten cases to simulate random errors was the same. In the image, the ideal coordinates of edge points were perturbed using a mixture of uniform and normal distribution along the edge direction and perpendicular to it. The uniform distribution has a range of -0.5 to +0.5 pixel in both directions representing the discretization errors. The normal distribution has a standard deviation of 0.5 pixel perpendicular to the edge and 25 pixels along the edge representing the identification errors. The ground position of edge points were perturbed using the nor-

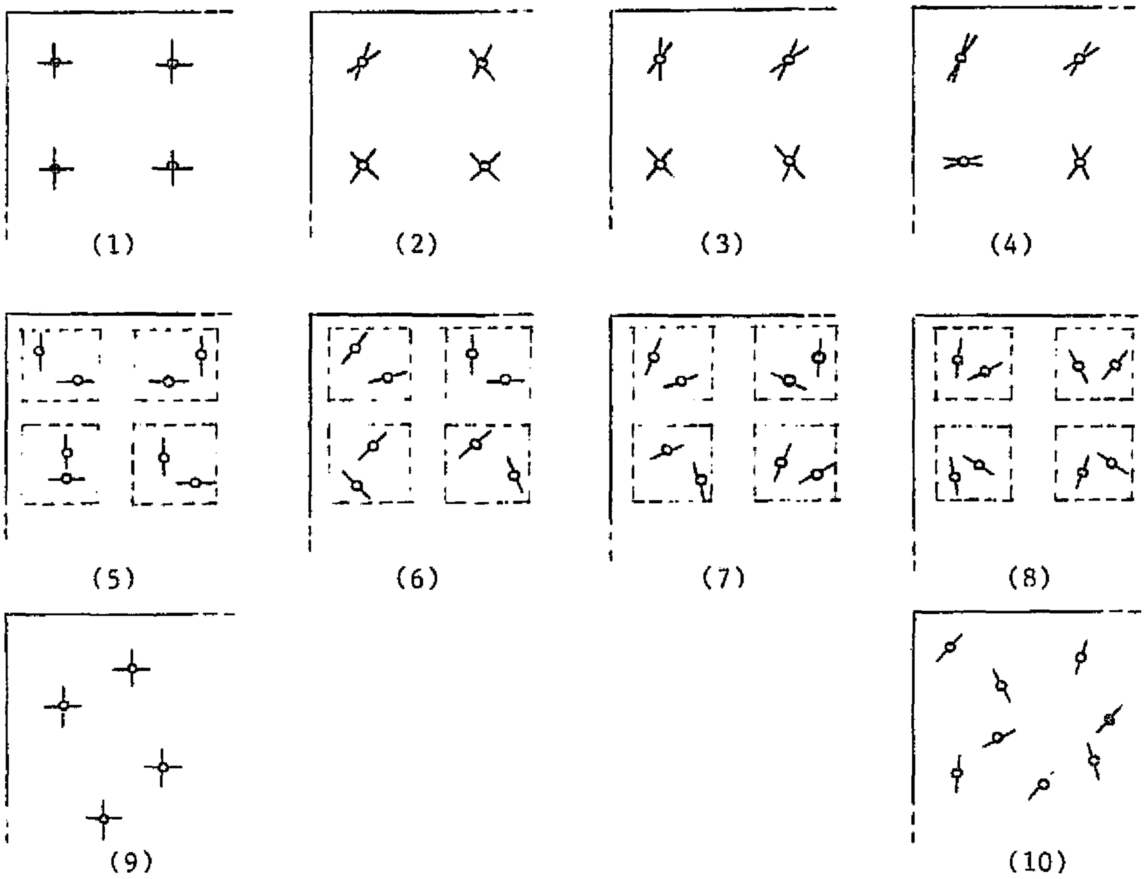


Figure 4 Distribution of Edges for Different Cases of Rectification With Edges as Control.

NOTE: (1) Pairs of edge points, having the same coordinates, are positioned at regular grid. The angle between the edges is 90°.

(2) Same as (1), except that the angle is at least 60°.

(3) Same as (1), except that the angle is at least 30°.

(4) Same as (1), except that the angle is arbitrary.

(5)-(8) Same as (1)-(4) respectively, except that the coordinates of edge points randomly deviate from regular grid up to 100 pixel.

(9) Same as (1), except that the position of a pair of edge points is totally random.

(10) Both the position of an edge point and the direction of the corresponding edge are arbitrary.

mal distribution with standard deviation of 15 meters in each of the three coordinate directions. The number of edge pairs for all cases varied from 25 to 145.

Check points were used to measure the accuracy of rectification. There were 144 check points situated on a uniform grid. For comparison purposes, the same check points were used for all cases. The image position of check points were perturbed in the same manner as edge points, except that the perturbations were applied in the row and column direction instead of along the edge and perpendicular to it and that the standard deviation for the normal component for the row and column direction were both 0.5 pixel. The ground position of check points were perturbed in exactly the same manner as those for edge points.

Each case in Figure 4 is replicated ten times using independent perturbations. A tabulation of the average rectification accuracy and the corresponding standard deviation are shown in Table 6. The average rectification accuracy for all cases are also shown in Figures 5 to 8. In these figures, the abscissa is the number of edge pairs and the ordinate is the average rectification accuracy of the ten replicates in meters. Each curve corresponds to the case number as annotated in the figures.

Figure 5 shows the results from cases (1) to (4). The only difference between these cases is the angle between the edges in an edge pair. It can be seen from the figure that decreasing the angle between edge pairs results in a corresponding decrease in

Table 6 Mean and Standard Deviation in Meters of the RMS Errors at Check Points Using Edge Points as Control

(Each Case Consists of Ten Replicates)

	Case 1		Case 2		Case 3		Case 4		Case 5	
No. of Line Pairs	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
25	73.17	1.13	79.85	2.22	77.92	1.28	105.66	8.86	69.92	0.74
41	69.24	0.67	75.24	2.38	74.51	1.21	85.76	2.82	67.94	0.56
81	65.81	0.74	69.26	1.22	70.42	0.76	75.22	1.46	65.63	0.52
145	64.59	0.71	-	-	-	-	69.29	1.08	-	-

	Case 6		Case 7		Case 8		Case 9		Case 10	
No. of Line Pairs	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
25	77.23	2.32	80.41	2.39	95.35	4.57	154.34	27.06	211.57	29.27
41	72.33	1.50	76.69	1.79	81.52	3.06	72.71	2.30	94.14	5.15
81	66.70	1.27	71.75	1.35	70.47	1.21	66.49	1.41	73.29	1.56
145	-	-	-	-	67.61	0.83	64.30	1.53	68.83	1.14

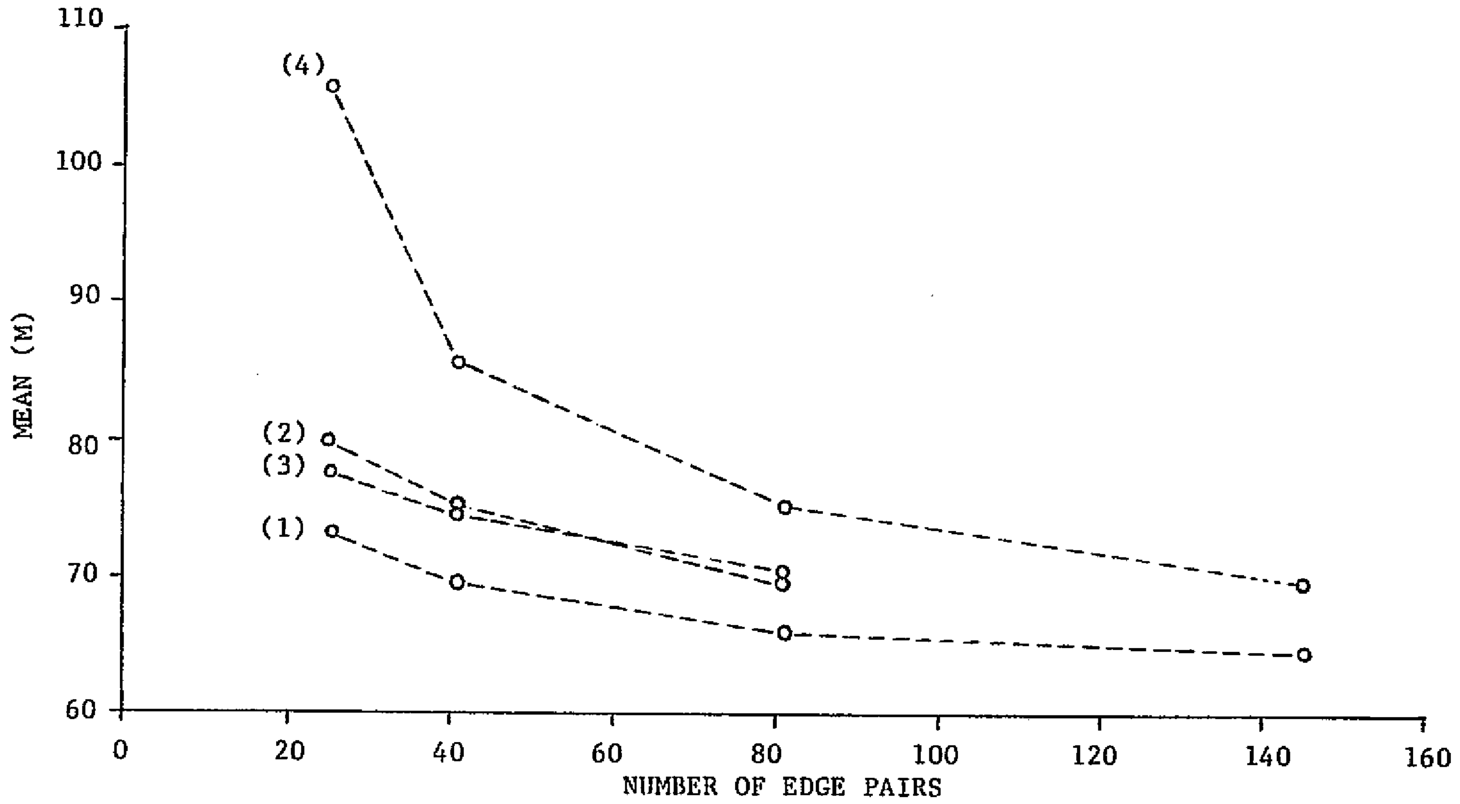


Figure 5 Plot of Rectification Results for Cases (1) to (4)

NOTE: Case (1): Pairs of edge points, having the same coordinates, are positioned at regular grid. The angle between the edges is 90°.
Case (2): Same as (1), except that the angle is at least 60°.
Case (3): Same as (1), except that the angle is at least 30°.
Case (4): Same as (1), except that the angle is arbitrary.

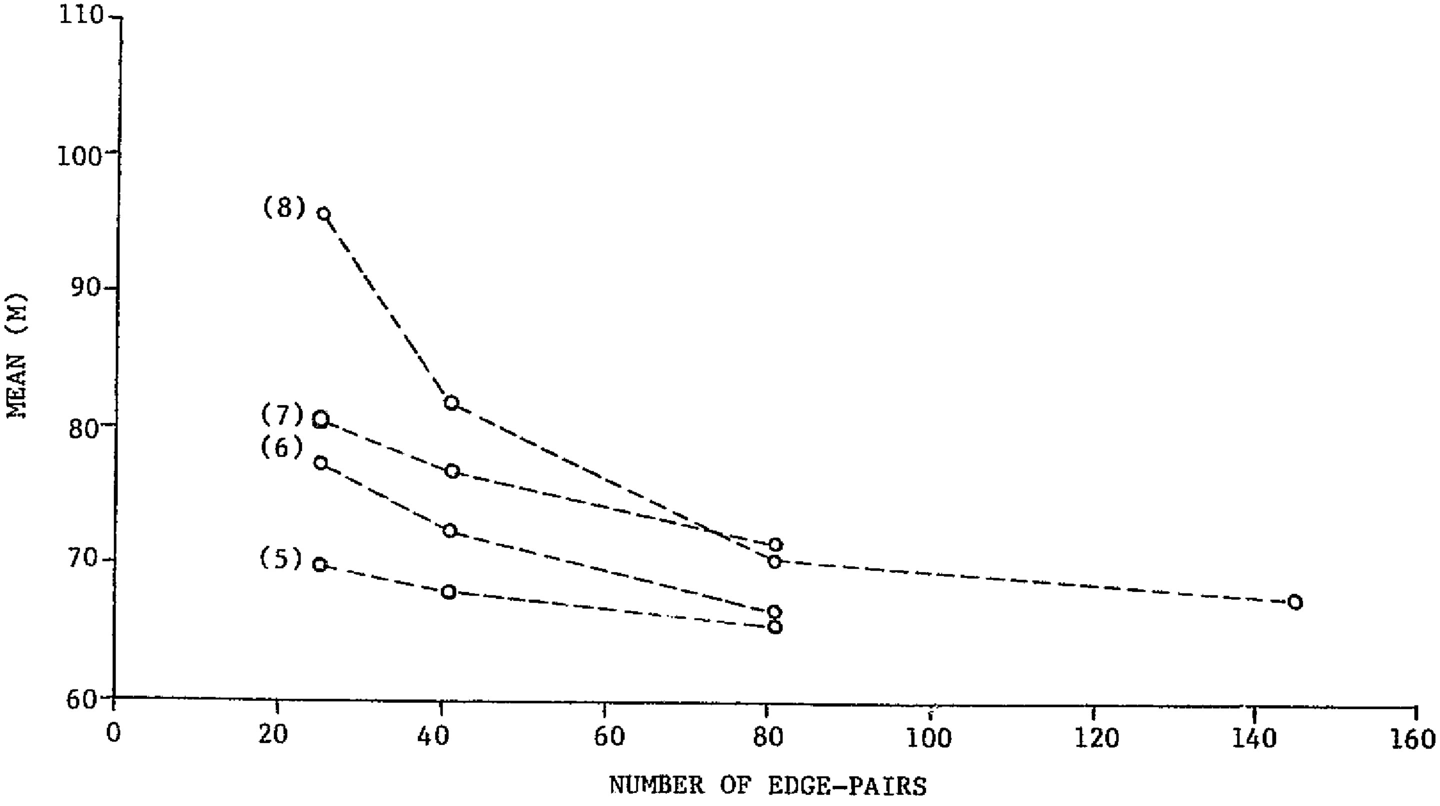


Figure 6 Plot of Rectification Results for Cases (5) to (8).

NOTE: Case (5): Edge points in an edge pair deviate randomly from regular grid up to 100 pixel. The angle between edges is 90°.
Case (6): Same as (5), except that the angle is at least 60°.
Case (7): Same as (5), except that the angle is at least 30°.
Case (8): Same as (5), except that the angle is arbitrary.

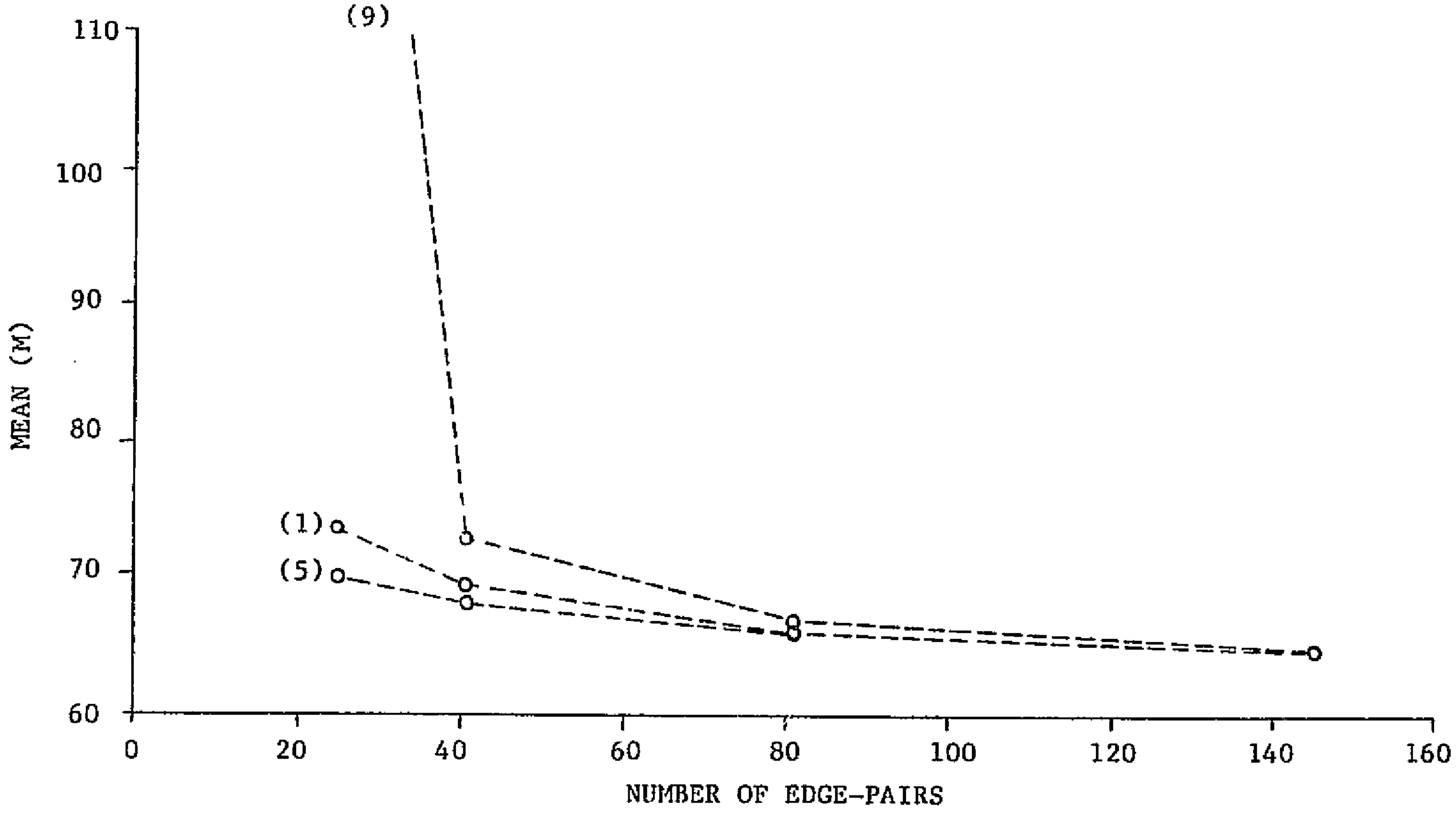


Figure 7 Plot of Rectification Results for Cases (1), (5), and (9).

NOTE: Case (1): Pairs of edge points, having the same coordinates, are positioned at regular grid. The angle between edges is 90°.

Case (5): Same as (1), except that the coordinates of edge points randomly deviate from regular grid up to 100 pixel.

Case (9): Same as (1), except that the position of a pair of edge points is totally random.

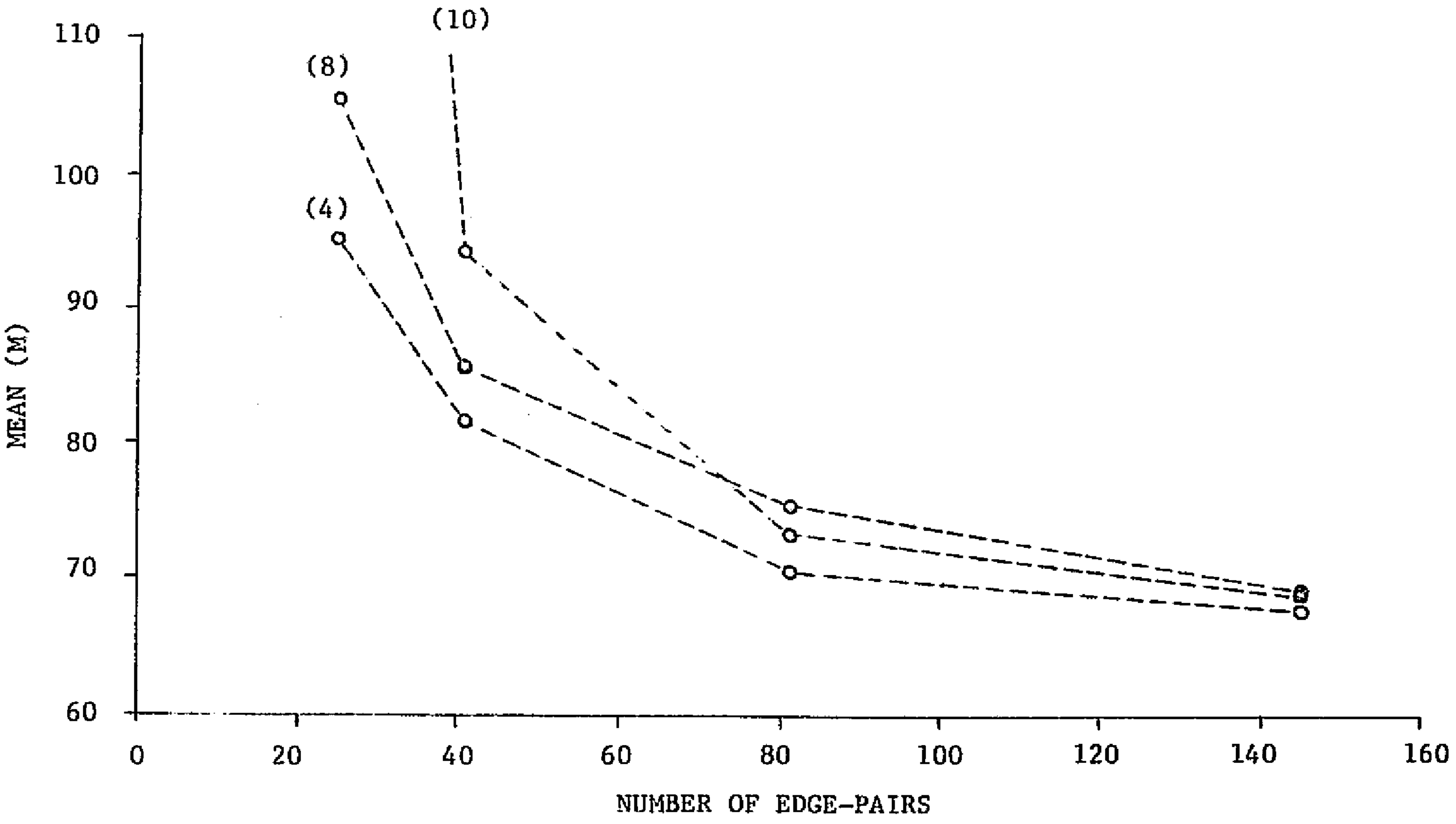


Figure 8 Plot of Rectification Results for Cases (4), (8), and (10).

NOTE: Case (4): Pairs of edge points have the same coordinates and are positioned at a regular grid. The angle between edges is arbitrary.

Case (8): Same as (4), except that the coordinates of edge points randomly deviate from regular grid up to 100 pixel.

Case (10): The position of edge points and the direction of edges are totally arbitrary.

rectification accuracy. This result is essentially repeated in Figure 6, because the only difference between these two figures is that the distance between edge points in an edge pair in all cases shown in Figure 5 is fixed at 0 while that for Figure 6 ranges up to 200 pixels. Comparing cases (1) and (4), or (5) and (8) in Figures 5 and 6 shows , that about 2 times more edge pairs are necessary to achieve the same accuracy as with conventional control points alone.

The effect of the distance between edge points in an edge pair is shown in Figures 7 and 8. Figure 7 shows cases (1), (5), and (9) where the angle between edges in an edge pair is fixed as 90°. Figure 8 shows cases (4), (8), and (10) where the angle is totally arbitrary. Separating the edges in edge pairs is beneficial up to a certain point. Total random distribution of edges over the whole image frame is inferior to other distribution when control edges are few.

Figure 9 is a comparison between cases (9) and (10). In case (9), where edge points in an edge pair have the same image coordinates and the pair of edges intersect at 90°, an edge pair is equivalent to a single control point. Case (10), where edges have totally arbitrary direction and distribution over the whole image frame, is the most extreme of all the ten cases studied. It can be seen from the figure that in order to achieve rectification accuracy when using edges comparable to that achieved when using conventional points, the number of edge pairs should be approximately 3 times the number

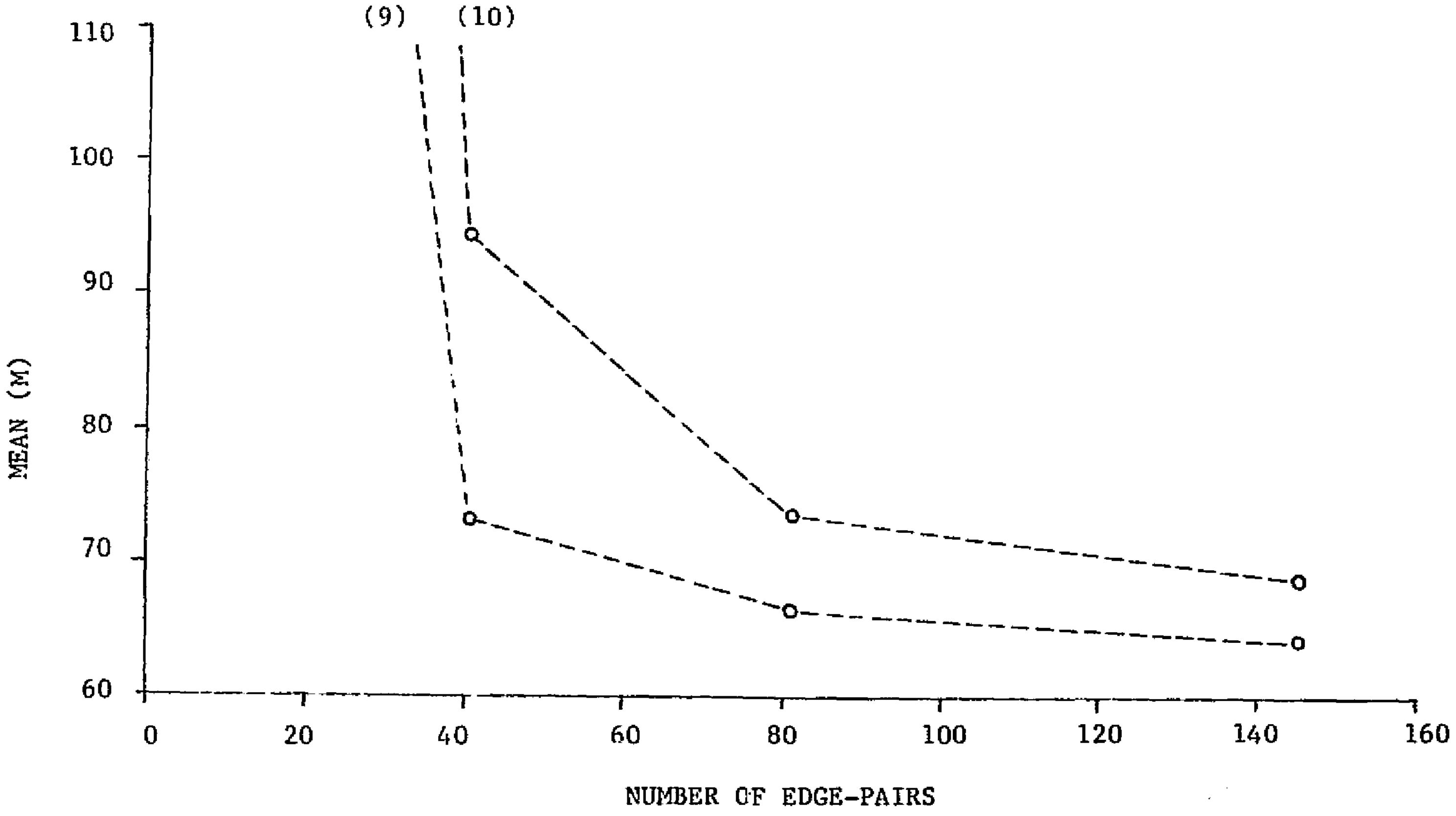


Figure 9 Plot of Rectification Results for Cases (9) and (10).

NOTE: Case (9): Pairs of edge points, having the same coordinates, are randomly distributed over the whole image frame. The angle between edges is 90°.
Case (10): The position of edge points and the direction of edges are totally arbitrary.

of control points. Thus it is worthwhile to attempt selecting well distributed control features.

Summarising the results of this approach, it has been shown that edge points can efficiently replace conventional control points. As they are much more likely to be found in an image and can be measured with at least the same precision as conventional points, one can expect that the overall rectification accuracy may even be improved.

For a practical implementation, especially to decrease the requirements on the skill of the operator, one should provide automatic algorithms for finding corresponding control features, for both conventional and edge points. This addresses the problem of scene matching. The next section is devoted to this problem and presents an algorithm, which is developed first for finding corresponding tie points in overlapping image frames.

4. ROBUST ESTIMATION FOR CORRESPONDENCE

4.1 A Hierarchical Approach to Correspondence

Scene matching is a basic requisite for different tasks which use the geometric properties of images, such as terrain classification, the derivation of digital height models or, map production. It is also the first step in applications where images are used for determining individual points in three dimensions as in photogrammetric triangulation. In all these cases either one image is related to another image (i.e. registration) or to a topographic map, (i.e. rectification).

Obviously there is no simple way to accomplish this task in one step. One rather has to pass several levels in a hierarchical way, where the results of one are the approximations for the next level. This is similar to the way the human visual system is believed to behave (Marr [28]). If one starts with a satellite image with a relative resolution of, say, 10^{-4}, i.e. 10^4 pixels per line, one could imagine a 4 step procedure, where each step increases the precision of rectification by about one order of magnitude:

1. A global image match which defines the position and the orientation to an accuracy of 2-10%, i.e. 200-1000 pixels, and 1–5°. This task is usually done by an operator but may use the very efficient algorithm by Lambird et.al. [25] (see also Stockmann, et.al. [40]).

2. In order to approximately compensate for unknown sensor position and attitude, and for relief displacements due to undulations of the terrain, one might continue with the matching of image patches. The number of these patches will depend on the roughness of the terrain in comparison to the flying height, and range from a few, say 5 or 10 to a hundred. The size of the image patches will be chosen in a way that the expected displacement will be less than half the linear patch size, thus between 4 and 20% of the side of the image. In order to keep the amount of data in a reasonable range one will use a reduced resolution, say between 1/2 and 1/8, leading to linear patch sizes of 50-200 pixels. The algorithm should be able to compensate for at least linear, i.e. affine distortions, and should lead to accuracies of 2-10 pixels, referring to the original image. Since high accuracy is not required, one might effectively use methods of structural pattern recognition to advantage by extracting scene features. One of the most promising algorithms for this step is the one by (Barnard and Thompson [4]).

3. Since fine correlation using differential methods requires approximate values which are within 1.5 pixels of the final match (Förstner [14]), an intermediate step is necessary. Here, all correlation-based methods can efficiently be used as the search area is very small. The window size will range beween 16 and 32 pixels (linear). The aim in this step is to choose a fast, robust, and reliable algorithm.

Possible candidates for this task are sequential algorithms (Barnea and Silverman [5]), the phase correlation technique (Kuglin and Hines [24]) or binary correlation. But of course the algorithm of step 2 could be used here too.

4. Fine correlation in the last step may yield subpixel accuracy, if the texture in the image allows and if it is required for the final product. Here, differential algorithms (Cafforio and Rocca [8], cf. Förstner [14]) are most efficient. The window size, depending on the texture, will range between 8 and 32 pixels. If the pixel size is adapted to the (spatial) spectrum of the images, accuracies of 0.2 pixels or better can be reached under production conditions (Bernstein [7], McGillem and Svedlow [29], Förstner [15]).

This sequence of steps has of course to take the special boundary conditions of the production into account, and may be varied accordingly. The main steps, however, will have to use similar algorithms. The concept is quite different from those used for the rectification of aerial images for orthohoto production, e.g. using the Gestalt Photo Mapper or the approach by Panton [35]. These systems do not have to cope with the weak geometry of satellite imagery, thus need only few control points. They can therefore use the internal geometry of the stereo pair for recursively updating the approximations for the fine correlation. Unlike these procedures, the above described hierarchical set up allows a great deal of parallelism in steps 2-4.

From the above mentioned algorithms the one by Barnard and Thompson needs further discussion. Its general line of thought can also be found in the approach by Lambird et.al. [25] and in the procedure by Marr, Poggio and Grimson (Marr [28], Grimson [17], cf. Kak [20]) .

With respect to its application in registration and rectification, the generality of Barnard and Thompson's geometric model turns out to be a disadvantage due to the resulting high numerical effort. Since the second step in the hierarchical procedure for registration and rectification is decisive for its reliability, this step has to be designed such that the actual data structure is taken into account, and it must also be flexible. Therefore a new algorithm has been developed, which can be used for registration and in particular for selecting tie points for rectifying overlapping image frames.

We will first formulate the problem of matching image patches of moderate sizes in subsection 4.2, discuss two of the algorithms and sketch the new one. Subsection 4.3 then describes the concept of the algorithm in detail. Subsection 4.4 is devoted to the actual implementation and subsection 4.5 contains an example, to demonstrate the performance.

4.2 The Problem of Correspondence

Let two images, or image patches, I' and I'' be given. Points i' and i'' in images I' and I'' may have the coordinates $z' = (x',y')^T$ and $z'' = (x'',y'')^T$, thus, z' and z'' are vectors, where T stands for transposed.

It is assumed that if i' and i'' are corresponding points, their coordinates can be related by

$$z'' = t(z';p) \tag{9}$$

where:

t is an arbitrary mapping function; it might reflect the knowledge about the geometric relation between the images I' and I''; and

p is a vector of unknown parameters $p_1, \dots, p_u$

It may be viewed as a severe restriction, that the mapping function must have an analytical form. But one should keep in mind that also a stochastic and/or segment-wise continuous function can be brought into the form of eq. (9). Eq. (9) will cause no difficulties, particularly in small scale imagery.

For an arbitrary pair of points (i', i'') there are two states of interest:

A. i' and i'' are corresponding points

B. i' and i'' are not corresponding points

The problem of correspondence now simply consists of: 1) finding the corresponding points; and 2) determining the parameter vector, p, of the mapping function. Theoretically the solutions to 1) and 2) are equivalent, as 1) implies 2) if applied to all pixels, say in I'. But this is neither feasible nor necessary, as the mapping function can reasonably be assumed to be smooth, i.e. roughly speaking bandlimited, and only a small number of corresponding points is sufficient to describe the mapping function. Whereas these pairs of corresponding points might replace the parameters p, the mapping function is necessary, if interpolation is required.

The known approaches actually use only a very limited number of points and explicitly or implicitly a mapping function of the type in eq. (9). In order to reduce the numerical effort and at the same time increase the reliability, objects o' and o'' are used in both images with feature vectors f' and f'' in addition to the coordinates z' and z'' attached to it:

$$o' = o'(z';f') \quad \text{and} \quad o'' = o''(z'';f'') \tag{10}$$

The procedures typically consist of three steps:

a. selection of appropriate objects o' and o'';

b. determining the similarity between all objects o' in image I' and all objects o'' in image I'', yielding possible candidates for corresponding objects

c. using some context information to find the pairs (o',o'') of corrrsponding objects.

4.2.1 The LNK-Method (Lambird et.al. [25], Stockmann et.al. [40])

a. Using edge detection procedures, this method selects objects which are either points or point pairs. Points belong to 4 classes. Pairs of such points are called abstract edges, abstract because the connecting line need not be a real edge in the image. For simplicity, we restrict the discussion to the point objects. Thus, an object o', say, in image f' is represented by its coordinates z' and its class $f' = \omega'$: $o' = o'(z',\omega')$.

b. Among all possible points (o',o'') of objects, only those which belong to the same classes are selected as possible candidates. Thus, if $\omega' = \omega''$ the objects o' and o'' are said to be similar.

c. The aim of the procedure is to determine the unknown parameters of the geometric transformation, which in this case consist of the two shifts in x- and y-directions. Each pair of similar objects leads to an equation $t(z',p) = z'-p$ which can be solved for p. The estimate $\hat{p}$ for the true shift p is taken from the histogram of all $p = z'-z''$ by searching for the peak-value representing the most probable shift. At the same time one obtains a classification of the object pairs into the two classes: ω_A of corresponding points and ω_B of non-corresponding points.

The approach is a direct solution, where no iterations are necessary. A further advantage is the sharp peak in the histogram, which guarantees a reliable solution even if the numbers of objects is large, i.e. even when the background noise is considerable. On the other hand, the method requires that eq. (9) in an extended form $(z'',f'') = t(z',f';p)$ is solvable for p. Thus, if more than two parameters have to be estimated, the object has to contain additional geometric features such as length and orientation, in f' and f'', thus, requiring more complex objects, such as lines, triangles, etc., to be extracted from the image. This might not only increase the number of combinations but also requires an additional dimension of the histogram for each additional unknown parameter. Nevertheless, a primary advantage is the absence of requirements for approximate values. Thus, with say 4 parameters, the images might have any relative orientation and scale. This method is therefore highly recommended for step 1 in the hierarchical scene matching procedure.

4.2.2 The Barnard-Thompson Algorithm [4]

a. This algorithm starts from objects which are represented by the gray level matrix g', say, centered at distinct points: $o'(z',g')$. The selection uses the interest operator by Moravec, namely the minimum variance of the gray level differences in the four main directions. This guarantees that no points on edges are selected, which are not discernible from neighbouring points on the same edge.

b. The similarity measure uses the coordinates and the gray level differences under consideration, deriving an initial probability that two objects $o'(z',g')$ and $o''(z'',g'')$ correspond, i.e. (o',o'') belongs to the class ω_A of corresponding points: $P((o',o'') \epsilon\ \omega_A) = f(z'-z'',g'-g'') \simeq 1/|g'-g''|^2$ if the shift $|\ z'-z''\ |$ is less than a threshold and $P((o',o'') \epsilon\ \omega_A) = o$ elsewhere.

c. The model of the geometric transformation is a differential one. They assume that the scene is regionwise smooth: $z''-z' = t(z')$ with the derivative $\partial t/\partial z'$ being bounded, except for the borders of the regions. The bound for $\Delta t/\Delta z'$ (being 1 pixel for $\Delta z' < 15$ pixels) is used to update the initial probabilities using a relaxation scheme (Rosenfeld et.al. [38]).

The model is extremely flexible, due to the randomness of the derivative within the admissible bound. The method can further be generalized by using more complex objects, e.g. the abstract edges of the LNK-method and thus can be an excellent solution for step 2 in the hierarchy. The numerical effort and the quality of the result, however, are highly scene dependent. In particular, the number and distribution of the selected objects are critical for the reliability of the result. Also the complexity of the geometrical model might not be necessary for satellite or aerial imagery of moderate scale (say, <1:20,000), thus questioning whether the numerical effort resulting from the relaxation process, cannot be reduced, if one takes the simpler geometry of "far-range" imagery into account.

Though both procedures follow the same general concept, their techniques are essentially different. The simple geometric model on which the LNK-Method is based, allows a fully consistent line of thought. This makes a statistical evaluation of the result feasible, e.g. using the broadness of the peak in the histogram. On the other hand, though the procedure of Barnard and Thompson is excellently motivated, it is heuristic. This prevents a thorough evaluation of its results.

4.2.3 Outline of the New Procedure

The new solution for the correspondence problem essentially aims at a maximum-likelihood estimation of the unknown parameters p of the geometrical transformation. It follows the same three steps of the procedures described above. An attempt has been made to derive the three steps on a common theoretical basis, and at the same time make it amenable to generalizations for rectification:

a. The same objects are used as in the Barnard-Thompson algorithms, namely points with their gray level matrix. The selection is guided by the theoretical precision expected from cross-correlation. It turns out, that this selection principle is closely related to Moravec's interest operator.

b. The similarity of pairs of objects is also based on the theoretical precision. In addition to the gray level difference between the two objects, the texture is taken into account, namely the variance of the gradient. Moreover, the formulation allows the introduction of correlation measures from any feature vectors, possibly including structural features. Thus, very general similarity measures can be used without losing the relation to the geometrical model.

c. The maximum likelihood estimation for the parameters p of the mapping function requires the knowledge of the probability density function of the observations. Observations in this case are the coordinate differences Δz from the modified form $\Delta z = z'' - z' = t(z'; p)$ of eq. (9). The coordinate differences of corresponding points can reasonably be assumed to be normally distributed, whereas the coordinate differences of non-corresponding points are approximately equally distributed between -d and +d where d is the dimension of the image patch. These observations therefore can be interpreted as outliers or blunders with respect to the model eq. (9). As the redundancy of the system is rather high, robust estimation procedures should work efficiently in this case. The high percentage of outliers, i.e. non-correspondence is compensated by the non-similarity of the objects, which lead to a low initial weight of these observations.

4.3 Mathematical Model

This section provides the mathematical model for the correspondence algorithm. We will start with the mapping functions and the robust estimation procedure for the determination of the unknown parameters. The similarity measure then leads us to the interest operator used for the point selection.

4.3.1 Mapping Functions

The relation between two image segments of a satellite or aerial image can be approximated by a low degree polynomial:

Shift only

(11 a,b) $$\underline{z}'' = a + \underline{z}' \quad \text{or} \quad \underline{\Delta z} = a$$

(Stochastical variables are underscored.)

Affine transformation

(12 a,b) $$\underline{z}'' = a + B\,\underline{z}' \quad \text{or} \quad \underline{\Delta z} = a + \bar{B}\,z'$$

Second order polynomial

(13 a,b) $$\underline{z}'' = a + B\,\underline{z}' + C\,\underline{z}' \otimes \underline{z}' \quad \text{or} \quad \underline{\Delta z} = a + \bar{B}z' + C\,z' \otimes z'$$

with

$$a = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}; \; B = \begin{bmatrix} p_3 & p_4 \\ p_5 & p_6 \end{bmatrix}; \; C = \begin{bmatrix} p_7 & p_8 & p_8 & p_9 \\ p_{10} & p_{11} & p_{11} & p_{12} \end{bmatrix}$$

$$(z' \otimes z')^T = (x'x' \;\; x'y' \;\; y'x' \;\; y'y')$$

and

$$\overline{B} = B - I.$$

By introducing conditions on the parameters p_i, one may restrict the mapping functions to conformal ones. For example, the conditions $p_3 = p_6$ and $p_4 = -p_5$ in eq. (12) lead to a similarity transformation with shifts, scale, and rotation only. The transformation parameters only occur linearly in the mapping function, thus could be solved in one step using the least squares technique.

4.3.2 Robust Estimation

The least squares technique starts from the linear (or linearized) model

$$(14) \qquad E(\underline{l}) = A\,\tilde{x} = \sum_{i=1}^{n} a_i^T\,\tilde{x}\,; \quad D(\underline{l}) = C_{ll} = \sigma_o^2\, Q_{ll}$$

where the nx1 vector $\underline{l}$ contains the observations, in our case the coordinate differences $\underline{\Delta z}$, with their covariance matrix C_{ll}. It is usually split into the unknown variance factor σ_o^2 and the known coefficient matrix Q_{ll}. The nxu design matrix A, having rows a_i, is supposed to be known. $\tilde{x}$ are the unknown parameters.

If the observations can be assumed to be uncorrelated, then one uses the weights w_i or the weight matrix $W = \mathrm{diag}\,(w_i) = \mathrm{diag}\,(1/q_{ii})$ to advantage, to solve the minimum problem

$$\Sigma\,(a_i'\,\underline{\hat{x}} - \underline{l}_i)^2\,w_i = \Sigma\,\underline{v}_i^2\,w_i \rightarrow \min. \tag{15}$$

It is known that the estimated parameters $\underline{\hat{x}}$ are sensitive to errors in the model eq. (9), especially gross errors or outliers, in the observations. This is due to the fact that the solution to eq. (15) is also the maximum likelihood estimator for $\tilde{x}$, if the observations are normally distributed. Observations with outliers, however, can be viewed to belong to longer tailed distributions. Examples are the Laplace-Distribution $f(x) = c\,e^{-|x|}$ and the Cauchy-Distribution $f(x) = c/(1+x^2)$.

In order to eliminate the effect of outliers on the result one can use maximum-likelihood type estimators. Then, instead of the sum of the squares of the residuals v_i the sum of a less increasing function $\rho(v_i)$ is minimized (Huber [19]):

$$\sum_i \rho(a_i^T\,\underline{\hat{x}} - \underline{l}_i) = \sum_i \rho(\underline{v}_i) \rightarrow \min \tag{16}$$

Discussion:

1. Choosing $\rho(v) = v^2/2$ gives the least squares estimator

2. Choosing $\rho(v) = \frac{1}{p} |v|^{p}$ yields the estimator minimizing the L_p-norm. A special case is obtained for $p = 1$: Minimizing $\rho(v) = |v|$ is the well known least sum method, being the ML-estimate for the Laplace-Distribution. It is the multiparameter version of the median. Barnea and Silverman [5] used it for cross correlation.

3. The choice of ρ can be guided by the evaluation of the "Influence-Curve" IC(v) (Hampel [18]) being proportional to the derivative $\psi(v) = \partial\rho/\partial v$ of the minimum function. IC(v) or $\psi(v)$ give an indication of how strong is the influence of an outlier on the estimate $\hat{\underline{x}}$.

4. The solution of eq. (16) can use existing programs for least squares solution, by either modifying the residuals, $v^* = \sqrt{\rho(v)}$ or by modifying the weights:

$$\Sigma\, \rho(v_i) = \Sigma \frac{\rho(v_i)}{v_i^2/2} \quad \frac{v_i^2}{2} = \Sigma\, w(v_i) \quad \frac{v_i^2}{2} \rightarrow \min. \tag{17}$$

using the weight function

$$w(v_i) = \frac{\rho(v_i)}{v_i^2/2+c} \quad (c \ll 1) \tag{18}$$

In an iterative solution the weights of all observations are updated depending on their residuals from the previous iteration:

$$w_i^{(\nu+1)} = w_i^{(0)}\, w(v_i^{(\nu)}) \tag{19}$$

5. If the function $\rho(v)$ is convex, thus $\psi(v)$ non-decreasing, and the model is linear, then convergence is guaranteed under broad conditions.

Minimizing the L_1-norm thus seems to be optimal, as it is robust, and convergence is guaranteed. This method however has two disadvantages:

1. $\rho(v)$ has no derivative at 0, thus, the influence curve is not continuous, which does not guarantee a unique solution.
2. The influence curve $\psi(v) = \text{sign}(v)$ is not zero for large values, thus large outliers have still an influence onto the result, which is not desirable.

We therefore propose to use the following weight functions.

1. In order to ascertain convergence we slightly modify the minimum function of the L_1-norm (cf. Figure 10).

$$\rho_1(v) = 2\,(\sqrt{1+v^2/2} - 1) \tag{20 a}$$

$$w_1(v) = \frac{4\,(\sqrt{1+v^2/2} - 1)}{v^2} \tag{20 b}$$

$$\psi_1(v) = \frac{v}{\sqrt{1+v^2/2}} \tag{20 c}$$

$\rho_1(v)$ is strictly convex with decreasing curvature for large v.

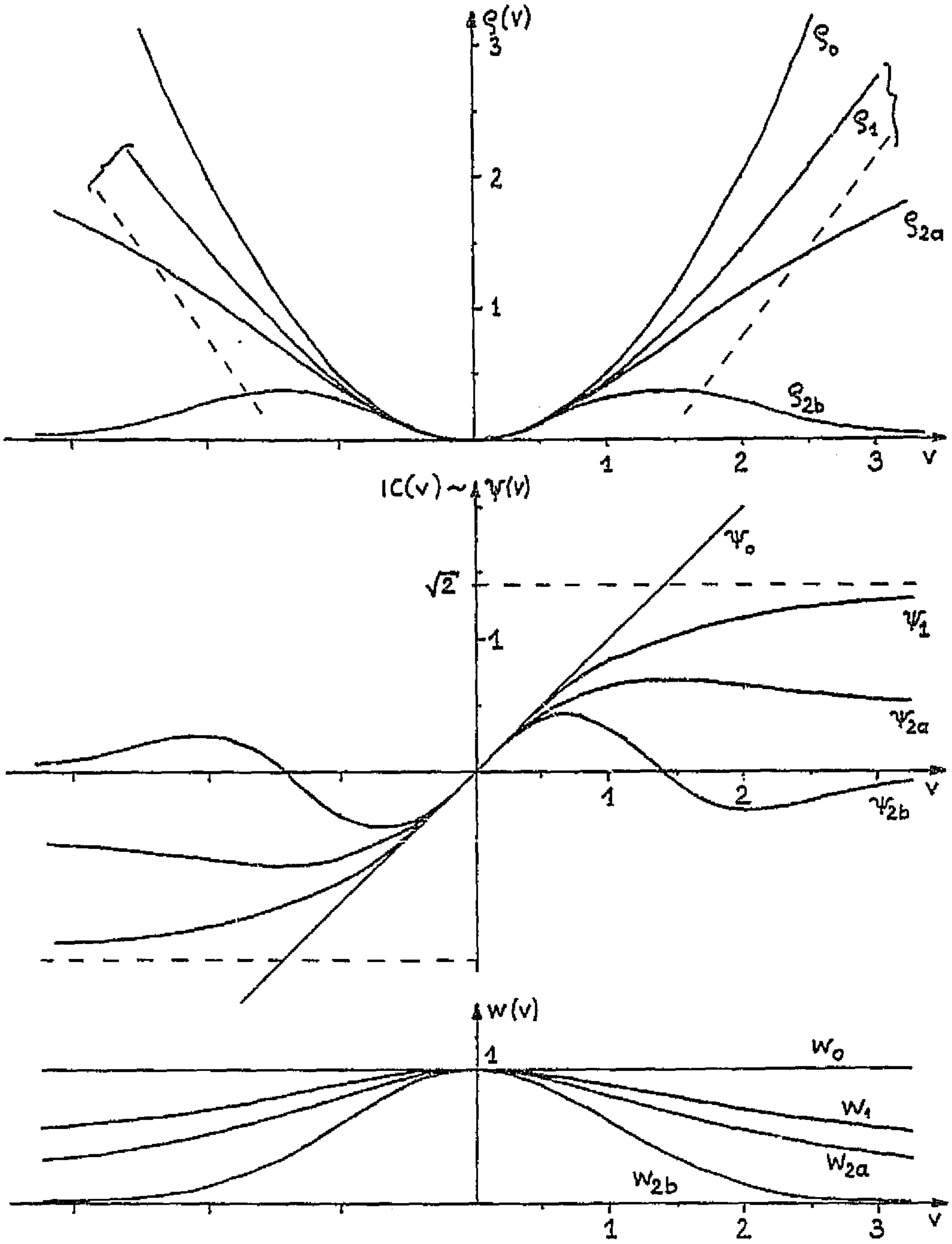


Figure 10 Minimum, Influence, and Weight Functions $\rho(v)$, $IC(v) \sim \psi(v)$, and $w(v)$.

NOTE:
0: least squares, non robust ($\psi(v)$ not bounded)
1: L_1-Norm, robust, convergence guaranteed
2: redescending IC,
a: ML-estimator for Cauchy-Distribution
b: exponential weight-function (Krarup [22])

2. After having reached convergence, one can assume to have good approximate values for the parameters. In order to eliminate the influence of large outliers one could take one of the following two minimum functions:

ρ_{2a} leads to maximum-likelihood estimators starting from a Cauchy-Distribution

$$\rho_{2a}(v) = \ln(1+v^2/2) \tag{21 a}$$

$$w_{2a}(v) = \frac{2\ln(1+v^2/2)}{v^2} \tag{21 b}$$

$$\psi_{2a}(v) = \frac{v}{(1+v^2/2)} \tag{21 c}$$

No convergence is guaranteed in the *general* case. Also, as ψ is descending for large v, no unique solution is guaranteed if arbitrary approximate values are allowed. This is meaningful as the Cauchy-Distribution has neither mean nor variance.

The following minimum function is proposed by Krarup et.al. [22] which considerably reduces the weights of false observations due to its exponential form:

(22 a) $$\rho_{2b}(v) = \frac{v^2}{2}\, e^{-v^2/2}$$

(22 b) $$w_{2b}(v) = e^{-v^2/2}$$

(22 c) $$\psi_{2b}(v) = v\ \left(1 - \frac{v^2}{2}\right)\ e^{-v^2/2}$$

This weight function fulfills practically all requirements for a well behaved weight function. (Hampel [18], Werner [43]). The functions are shown in Figure 10 together with the minimum weight and influence function of the least squares, $\rho_o(v) = v^2/2$.

4.3.3 Similarity Measure

The estimation procedure requires initial weights for the observations which in our case are the coordinate differences Δz of object or point pairs, which need not correspond. Hence, the majority of the observations are outliers and assuming equal weight would prevent the solution from getting started.

Reasonable weights can be obtained from the covariance matrix of the estimated shifts $\underline{\Delta z}$, if we would apply cross correlation to all pairs of points. It is given by (Förstner [14])

$$(23) \qquad V(z''-z') = \mathrm{Cov}\begin{bmatrix} x''-x' \\ y''-y' \end{bmatrix} = \hat{\sigma}^2_{\Delta g}\begin{bmatrix} \Sigma g_x^2 & \Sigma g_x g_y \\ \Sigma g_y g_x & \Sigma g_y^2 \end{bmatrix}^{-1} = \hat{\sigma}^2_{\Delta g}\ Q$$

where:

g is the gray level function of the object, restored from g' and g''

$\hat{\sigma}^2_{\Delta g}$ the estimated variance of the gray level differences, and

g_x, g_y are the gradients of g in x- and y-directions respectively.

The covariance matrix fully describes the precision of the match between the gray level function g' and g'' of the two objects o' and o''. This precision depends on:

1. The number of pixels used.
2. The noise variance.
3. The texture of the object, namely the edge business. It can be shown that this measure is directly related to the bandwidth of the signal and the curvature of the cross correlation function (Förstner [14]).

The covariance matrix can be visualized by an error ellipse (cf. Mikhail [34]), giving the precision of the match for all directions. A good match therefore must fulfill the following two requirements:

C1: The error ellipse should be close to a circle, otherwise the match is not well determined in one direction, e. g. at an edge.

C2: The error ellipse should be small.

Both criteria will be used for the measure of similarity between two objects and the selection of interesting points.

If the ellipse is close to a circle the weight can be directly derived from the trace of the covariance matrix

(24) $$w = \frac{1}{\mathrm{tr}(V)} = \frac{1}{\hat{\sigma}^2_{\Delta g}\ \mathrm{tr}\ (Q)}$$

Observe, that the trace is invariant to rotations. Taking the gray level differences directly to estimate $\hat{\sigma}^2_{\Delta g}$ has the disadvantage of being biased if the two images have different brightness and contrast. The correlation coefficient is known to be a better measure. Now, if one for simplicity assumes the images g' and g'' to be related to the true image g by $g' = a' (g+n') + b'$ and $g'' = a'' (g+n'') + b''$, with $\sigma^2_{n'} = \sigma^2_{n''} = \sigma^2_n$ where a and b represent contrast and brightness, the signal to noise ratio $SNR^2 = \sigma^2_g/\sigma^2_n$ is functionally related to the correlation coefficient by:

(25 a)
$$\rho = \frac{\sigma_{g'g''}}{\sigma_{g'}\sigma_{g''}} = \frac{\sigma_g^2}{\sigma_g^2+\sigma_n^2} = \frac{\mathrm{SNR}^2}{\mathrm{SNR}^2+1}$$

or

(25 b)
$$\mathrm{SNR}^2 = \frac{\sigma_g^2}{\sigma_n^2} = \frac{\rho}{1-\rho}$$

By using the approximations

(26)
$$\sigma_g^2 \simeq \sigma_{g'}\ \sigma_{g''}$$

(27)
$$\mathrm{tr}\ Q = \sqrt{\mathrm{tr}\ Q'\ \ \mathrm{tr}\ Q''}$$

and

(28)
$$\sigma_{\Delta g}^2 = 2\ \sigma_n^2$$

we obtain the following relation for the weight of the observation Δz:

(29)
$$w(o',o'') \simeq \frac{1}{2}\ \frac{\rho}{1-\rho}\ \frac{1}{\sigma_{g'}\sigma_{g''}\cdot\sqrt{\mathrm{tr}\ Q'\ \ \mathrm{tr}\ Q''}}$$

Discussion:

1. The weight depends on two terms. The first term reflects the similarity between the two objects and needs to be calculated for all object pairs. The second term depends on values obtained separately from both images.

2. The traces tr Q' and tr Q'' measure the distinctness or the locatability of the objects and are critical for the selection of appropriate points. The reason is, the noise level $\hat{\sigma}^2_{\Delta g}$ can be realistically assumed to be constant in both images.

3. The weight is a generalization of the one used by Barnard and Thompson. It differs in two ways. First, it is independent of brightness and contrast, as we are only interested in the weight ratios. Second, it takes the texture of the object into account.

4. A simple and reasonable criterion to reject object pairs based on the correlation coefficient is $\rho \geq \frac{1}{2}$. This is equivalent to requiring the SNR to be larger than 1.

5. The main advantage of the separation of the different terms in eq. (29) lies in its ability to include other measures for similarity. The correlation coefficient need not be derived from the gray levels but may use other features f' and f'' of the objects, e.g.:

   a. One could use rotation and scale invarant features, as the moments proposed by Wong and Hall [44].

b. One could use a small set of features just to decrease the computation time, e.g. the low frequency terms of a cosine transform.

c. One could use structural information, the result of a classification or a linguistic description in combination with statistical measures. The only requirement for the measure is to have the properties of a correlation coefficient.

The separation of the correlation coefficient from the variance and the texture of the gray level function, allows one to generalize the weight determination without losing the information about the geometric distinctness of the object.

4.3.4 Interest Operator

We have assumed that the error ellipse representing the covariance matrix of the coordinate difference is close to a circle. Moreover, we require that the point can be well located. Measures of both requirements should, in a simple way, be derivable from the gray level function of the image patch, as they have to be determined for all pixels. They should also be invariant to rotation; a scale factor will not change too much the ranking of the different pixels.

As the eigenvalues of the covariance matrix are invariant to rotations, and the trace equals the sum of the eigenvalues, we will use them also for determining the closeness of the error ellipse to a circle. Moreover, the eigenvalues of the coefficient matrix, say Q', and those of its inverse $N' = (Q')^{-1}$ are related by $\lambda_i(Q') = 1/\lambda_i(N')$. Thus, let λ_1 and λ_2 be the eigenvalues of N', then the ratio

$$q = \frac{4 \det N'}{(\mathrm{tr}\, N')^2} = \frac{4\,\lambda_1\lambda_2}{(\lambda_1 + \lambda_2)^2} = 1 - (\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2})^2 \tag{30}$$

is an adequate measure for the closeness of the error ellipse to a circle. If $q = 0$ (and not both λ_1 and λ_2 are zero), then $\det N'$ is zero and the matrix is singular. This means that g_x and g_y are linearly dependent thus the point may lie on an edge. The case $g = 1$ is reached, only if the eigenvalues are equal ($\lambda_1 - \lambda_2 = 0$) thus representing a circular error ellipse. The calculation of q need not use the eigenvalues, but rather the determinant and the trace of N':

$$\det N' = \Sigma(g_x')^2 \,.\, \Sigma(g_y')^2 - (\Sigma g_x' g_y')^2 \tag{31 a}$$

$$\mathrm{tr}\, N' = \Sigma(g_x')^2 + \Sigma(g_y')^2 \tag{31 b}$$

The sums can be readily derived from the squared and multiplied gradient images by convolution.

Similarly, one can derive an expression for tr Q$'$:

$$\text{tr } Q' = \frac{\text{tr } N'}{\det N'} \tag{32}$$

Thus the selection of interesting points can be accomplished for both images separately in the following steps:

1. Determination of Σg_x^2, $\Sigma g_x g_y$, and Σg_y^2;

2. Determination of tr Q and q using eq. (30) - (32);

3. Determination of the interest value, being a preliminary weight,

$$\overline{w} = \begin{cases} \dfrac{1}{\text{tr } Q} = \dfrac{\det N}{\text{tr } N} & \text{for } q > \text{threshold} \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

   for each pixel;

4. Suppressing all non-maxima in the function $\overline{w}(i,j)$;

5. All values $\overline{w}(i,j)$ give rise to an object o.

4.4 Algorithmic Solution

4.4.1 The Selection of Objects of Interest

The interest oprator eq. (30) to (33) requires the variance and covariance of the gradient image at each pixel. The used window size should be adaptable to the texture of the image patch. If one uses a square (in general, a rectangle) window the number of operations per pixel needed for the interest operator can be made independent of the window size. This is due to the fact, that the array $I(\Sigma g_x^2)$, say, containing the sums Σg_x^2 can be derived from g_x^2 by convolution with a separable window of size n_{s1} x n_{s2}, $W(i,j) = 1$ with $w = e_1\, e_2^T$ and $e_i^T = (11...1)$ containing n_{si} elements 1. As the convolution with e, or e^T, needs only 2 additions, if done recursively, only 4 additions per pixel are necessary for the determination of the array $I(\Sigma g_x^2)$ independent of the window size. The gradients g_x and g_y are calculated with the Roberts operator.

Now two thresholds $q_{min.}$ and $\overline{w}_{min.}$ are necessary to check the form and the size of the ellipse:

C1: $q_i > q_{min.}$ (form)

C2: $\overline{w}_i > \overline{w}_{min.}$ (size)

If both conditions are fulfilled, the interest value of that pixel is set to the preliminary weight $\overline{w} = 1/\text{tr } Q$, otherwise it is zero.

The threshold $q_{min.}$ is scale independent, a value of $q_{min.} = 0.25$ turned out to be reasonable. The condition C2 should also be independent of scale. Therefore, we used $w_{min.} = f \cdot \Sigma \overline{w}_i / n$, relating the preliminary weights $\overline{w}_i$ to their mean value. A value $f = 1.5$ was chosen for all tests performed.

From the resulting interest values, $\overline{w}$ or 0, the relative maximum within a certain window n_m x n_m are extracted. The window size n_m for this non-maximum suppression is independent from the one used for the sums. If the window size n_m is larger than 3 the non-maximum suppression is accomplished in two steps, the first using a 3x3 window and the second performing the comparisons in a spiral manner in the large window to keep the number of comparisons independent of the window size n_m. The selected objects are then stored in a list, containing the coordinates and the preliminary weight $\overline{w} = 1/\mathrm{tr}(Q)$. They are needed for the similarity measure.

4.4.2 The Selection of Object Pairs

The initial weight $\overline{w}$ from eq. (29) in addition to tr Q' and tr Q'', requires the standard deviations σ_g' and σ_g'' and the correlation coefficient $\rho = \sigma_{g'g''}/(\sigma_{g'}\sigma_{g''})$ where:

(34 a)
$$\sigma_{g'}^2 = \frac{\Sigma(g')^2 - (\Sigma g')^2/n}{n-1}$$

(34 b)
$$\sigma_{g''}^2 = \frac{\Sigma(g'')^2 - (\Sigma g'')^2/n}{n-1}$$

and

$$\sigma_{g'g''} = \frac{\Sigma g'g'' - (\Sigma g')(\Sigma g'')/n}{n-1} \quad (34\ c)$$

The sums $\Sigma g'$, $\Sigma g''$, $\Sigma(g')^2$, and $\Sigma(g'')^2$ are calculated for each object. The mixed sum $\Sigma g'g''$ is only calculated for pairs of objects with a distance $|z'-z''|$ less than a given threshold $d_{max.}$, which reflects the maximum expected distances between corresponding objects. All pairs of objects for which the correlaion coefficient ρ is greater than 0.5 are collected in a list, containing their coordinates z' and z'' and their weights.

4.4.3 The Selection of Corresponding Points

The selection of the corresponding points is based on the assumed geometrical relation between the images. In our context an affine transformation seems to be adequate, and therefore, has been employed. The robust adjustment is split into two steps. First, only the shift between the images is determined. This leads to better approximations, both for the shifts and the weights in the following 6 parameter transformation. Both adjustments have the same structure.

In each iteration, the parameters, the residuals, the precision of the shift, and the average weight are determined, and the weights are adapted for the next iteration. If a weight is smaller than a certain percentage (say, 10%) of the average weight, it is set to zero, eliminating that observation. The first 4 iterations are performed with the weight function given by eq. (20b), after which th redescending function in eq. (22b) is applied. The agorithm stops if either the required precision of the shift is reached, not enough corresponding points are left, or a pre-set number of iterations is reached. The residuals of the last iteration are tested, and with all residuals passing this test one additional iteration with equal weights is performed to obtain the final transformation parameters.

The obtained list of corresponding points may then still be ambiguous, as the same point in one image might correspond to several points of a cluster in the other image. The list of pairs of points is then cleaned keeping those correspondences which have the smaller residuals.

4.5 Two Examples

The following two examples are presented to show the performance of the new algorithm. In both cases, the two images I' and I'' are derived from an original image by extracting two separate windows and distorting them by an affine transformation according to eq. (12b) with random numbers in $\overline{B}$ ranging up to 0.15. Thus, the average linear distortion is approximately 10% or 6°. The extracted windows are contaminated with white Gaussian noise with a standard deviation of $\sigma = 15$ gray levels. Both windows are then smoothed with a 3x3 Hanning filter $(1\ 2\ 1)^T * (1\ 2\ 1)$.

1. The first example is based on an artificial image (cf. Figure 11). It may represent a part of a rural scene with some light roads between fields of different brightness. The dark pixels are the points selected by the interest operator. Table 7 contains the preliminary weight $\overline{w}$ and the values of q in percent describing the closeness of the error ellipse to a circle.

Observe, that some points, e.g. point 1 in the right image, lie on an edge, but due to the irregularity of the edges have been selected. Both values, $\overline{w}$ and q are small ($\overline{w} = 311$, $q = 36$ % in this case). From the 15x16 = 240 possible point pairs 59 were selected as possible candidates for correspondence. Their weights vary considerably, namely due to the correlation coefficient (cf. Table 8). The robust shift adjustment yields the pairs listed in Table 9. It shows the ambiguity of the result, as for example point 3 in the left image is connected with points 7 and 9 in the right image. As the residuals of the pair (3,7) are smaller than those of (3,9) the pair (3,7) is kept. The cleaned list in Table 9b would be the result with the shift parameters only, showing that even with a wrong geometric model nearly all corresponding objects can be found (cf. Table 10b).

The result of the robust affine transformation (cf. Table 10) shows a slightly different result. The final correspondences are shown in Figure 12. If one compares the final result with the list of the candidate pairs (Table 8), obviously the

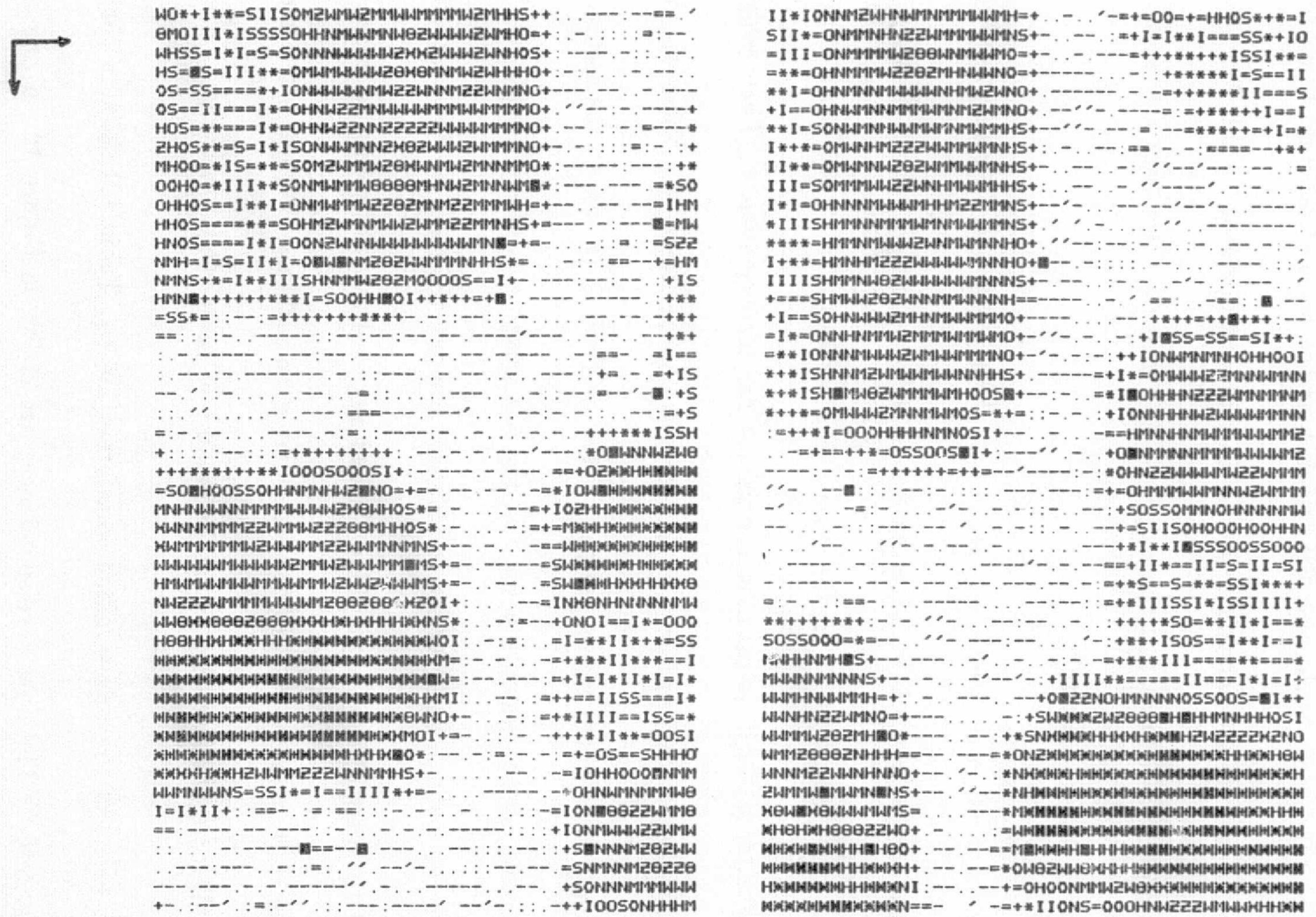


Figure 11 Example 1: Artificial Image Pair With Selected Points (black pixels)

NOTE: Window size of interest operators: 7

Window size for non-maximum suppression: 3

Left Image

i	x	y	$\bar{w}$	q
1	4	4	357	91
2	10	34	410	49
3	12	45	907	87
4	13	31	364	43
5	14	15	410	67
6	[illegible]	17	450	59
7	[illegible]	4	714	88
8	15	21	483	39
9	16	31	345	56
10	21	45	961	62
11	24	41	964	48
12	26	4	316	37
13	26	19	622	64
14	26	40	963	51
15	30	23	448	53
16	31	38	1871	94
17	36	25	1518	82
18	40	22	867	50
19	41	45	311	85
20	43	40	640	92
21	45	14	324	50
22	45	19	327	57
23	45	39	686	78

Right Image

i	x	y	$\bar{w}$	q
1	14	25	311	36
2	16	45	300	40
3	17	42	288	39
4	18	36	688	71
5	21	7	586	72
6	21	22	469	65
7	21	33	423	62
8	24	18	445	61
9	24	33	708	89
10	26	8	280	48
11	29	38	325	60
12	35	8	718	74
13	37	27	1374	65
14	37	45	368	46
15	38	36	508	50
16	38	38	540	54
17	39	11	314	37
18	42	6	289	86
19	42	11	396	42
20	43	4	301	81
21	45	5	299	77
22	45	10	446	59
23	45	24	1683	88
24	45	29	344	45

Table 7 Example 1: List of Selected Points

NOTE: x,y coordinates

$\bar{w}$ interest value

q measure for isotropy of error ellipse (in percent)

	i j	w	$\bar{w}_i$	$\bar{w}_j$	rho
1	201	4.757	410.	311.	0.813910
2	206	18.507	410.	469.	0.934145
3	307	6.047	907.	423.	0.743084
4	309	9.129	907.	708.	0.769720
5	406	3.555	364.	469.	0.733859
6	408	8.666	364.	445.	0.872919
7	505	27.938	410.	586.	0.932111
8	510	2.052	410.	280.	0.572882
9	605	16.436	450.	586.	0.883114
10	608	2.755	450.	445.	0.623202
11	708	16.076	714.	445.	0.895975
12	808	11.647	483.	445.	0.899352
13	810	1.838	483.	280.	0.589571
14	906	4.533	345.	469.	0.777062
15	908	11.360	345.	445.	0.899266
16	911	2.187	345.	325.	0.599452
17	1002	2.891	961.	300.	0.611310
18	1003	2.454	961.	288.	0.611800
19	1004	2.578	961.	688.	0.519524
20	1007	2.692	961.	423.	0.562607
21	1102	2.335	964.	300.	0.613341
22	1103	8.660	964.	288.	0.874139
23	1104	55.190	964.	688.	0.966564
24	1107	2.528	964.	423.	0.601318
25	1113	135.224	964.	1374.	0.983662
26	1114	3.310	964.	368.	0.681337
27	1115	13.244	964.	508.	0.891956
28	1116	7.161	964.	540.	0.812974
29	1212	4.827	316.	719.	0.776955
30	1312	22.732	622.	718.	0.919691
31	1313	7.178	622.	1374.	0.764347
32	1317	1.891	622.	314.	0.579353
33	1403	2.995	963.	288.	0.689592
34	1404	11.141	963.	688.	0.843669
35	1407	5.303	963.	423.	0.745271
36	1409	2.978	963.	708.	0.557126
37	1413	15.047	963.	1374.	0.861033
38	1415	4.759	963.	508.	0.732863
39	1416	3.142	963.	540.	0.638196
40	1512	7.280	448.	718.	0.800946
41	1515	2.030	448.	508.	0.584841
42	1516	2.125	448.	540.	0.589507
43	1517	12.494	448.	314.	0.908975
44	1519	24.556	448.	396.	0.943966
45	1522	11.948	448.	446.	0.874251
46	1609	9.858	1871.	708.	0.746461
47	1611	3.551	1871.	325.	0.556535
48	1623	129.159	1871.	1683.	0.970396
49	1706	16.139	1518.	469.	0.879777
50	1708	2.468	1518.	445.	0.533513
51	1717	6.066	1518.	314.	0.764804
52	1719	4.692	1518.	396.	0.683444
53	1810	1.567	867.	280.	0.504529
54	1824	5.183	867.	344.	0.779709
55	1914	3.144	311.	368.	0.693980
56	1915	7.749	311.	508.	0.843612
57	1916	6.314	311.	540.	0.810626
58	2013	5.123	640.	1374.	0.651036
59	2323	29.446	686.	1683.	0.911693

Table 8 Example 1: List of Selected Pairs

NOTE: ij point No. in left and right image (201 ≅ (2,1)), w initial weight, $\bar{w}$ preliminary weights, rho correlation coefficient.

```
estimated shift:    11.077   -12.615

a)
  i  left right         xl         yl         xr         yr         dx         dy


   1     2     6     10.000     34.000     21.000     22.000     -0.077      0.615
   2     3     7     12.000     45.000     21.000     33.000     -2.077      0.615
   3     3     9     12.000     45.000     24.000     33.000      0.923      0.615
   4     4     8     13.000     31.000     24.000     18.000     -0.077     -0.385
   5     8    10     16.000     21.000     26.000      8.000     -1.077     -0.385
   6     9     8     16.000     31.000     24.000     18.000     -3.077     -0.385
   7    11    13     24.000     41.000     37.000     37.000      1.923     -1.385
   8    13    12     26.000     19.000     35.000      8.000     -2.077      1.615
   9    14    13     26.000     40.000     37.000     27.000     -0.077     -0.385
  10    15    17     30.000     23.000     39.000     11.000     -2.077      0.615
  11    15    19     30.000     23.000     42.000     11.000      0.923      0.615
  12    15    22     30.000     23.000     45.000     10.000      3.923     -0.385
  13    16    23     31.000     38.000     45.000     24.000      2.923     -1.385

   clean list


b)
  i  left right         xl         yl         xr         yr         dx         dy


   1     2     6     10.000     34.000     21.000     22.000     -0.077      0.615
   2     3     9     12.000     45.000     24.000     33.000      0.923      0.615
   3     4     8     13.000     31.000     24.000     18.000     -0.077     -0.385
   4     8    10     16.000     21.000     26.000      8.000     -1.077     -0.385
   5    13    12     26.000     19.000     35.000      8.000     -2.077      1.615
   6    14    13     26.000     40.000     37.000     27.000     -0.077     -0.385
   7    15    19     30.000     23.000     42.000     11.000      0.923      0.615
   8    16    23     31.000     38.000     45.000     24.000      2.923     -1.385
```

Table 9 Example 1: Result of Robust Shift Adjustment

a) uncleaned list containing ambiguities
b) cleaned list

NOTE: 6 iterations

$$\begin{pmatrix} 1.19529 & 0.10307 & ; & 3.58 \\ -0.08967 & 0.88201 & ; & -6.45 \end{pmatrix} = (\hat{B} ; \hat{a})$$

a)

```
 i  left right        xl        yl        xr        yr        dx        dy
                                 .

  1    2    6     10.000    34.000    21.000    22.000    -1.961     0.637
  2    3    7     12.000    45.000    21.000    33.000     1.563    -0.840
  3    3    9     12.000    45.000    24.000    33.000    -1.437    -0.840
  4    4    8     13.000    31.000    24.000    18.000    -1.684     1.722
  5    8   10     16.000    21.000    26.000     8.000    -1.129     2.633
  6    9    8     16.000    31.000    24.000    18.000     1.902     1.453
  7   11   13     24.000    41.000    37.000    27.000    -0.505     0.556
  8   13   12     26.000    19.000    35.000     8.000     1.618    -0.027
  9   14   13     26.000    40.000    37.000    27.000     1.782    -0.505
 10   15   17     30.000    23.000    39.000    11.000     2.811     0.142
 11   15   19     30.000    23.000    42.000    11.000    -0.189     0.142
 12   15.  22     30.000    23.000    45.000    10.000    -3.189     1.142
 13   16   23     31.000    38.000    45.000    24.000    -0.448     0.282
 14    5    5     14.000    15.000    21.000     7.000     0.862    -1.480
 15   13   17     26.000    19.000    39.000    11.000    -2.382    -3.027
 16    6    5     14.000    17.000    21.000     7.000     1.068     0.284
 17    4    6     13.000    31.000    21.000    22.000     1.316    -2.278


 clean list
```

b)

```
 i  left right        xl        yl        xr        yr        dx        dy


  1    2    6     10.000    34.000    21.000    22.000    -1.961     0.637
  2    3    9     12.000    45.000    24:000    33.000    -1.437    -0.840
  3    8   10     16.000    21.000    26.000     8.000    -1.129     2.633
  4    9    8     16.000    31.000    24.000    18.000     1.902     1.453
  5   11   13     24.000    41.000    37.000    27.000    -0.505     0.556
  6   13   12     26.000    19.000    35.000     8.000     1.618    -0.027
  7   15   19     30.000    23.000    42.000    11.000    -0.189     0.142
  8   16   23     31.000    38.000    45.000    24.000    -0.448     0.282
  9    6    5     14.000    17.000    21.000     7.000     1.068     0.284
```

Table 10 Example 2: Result of Robust Affine Transformation

a) uncleaned list, containing ambiguities
b) cleaned list, final result
(cf. Figure 12)

NOTE: 6 iterations

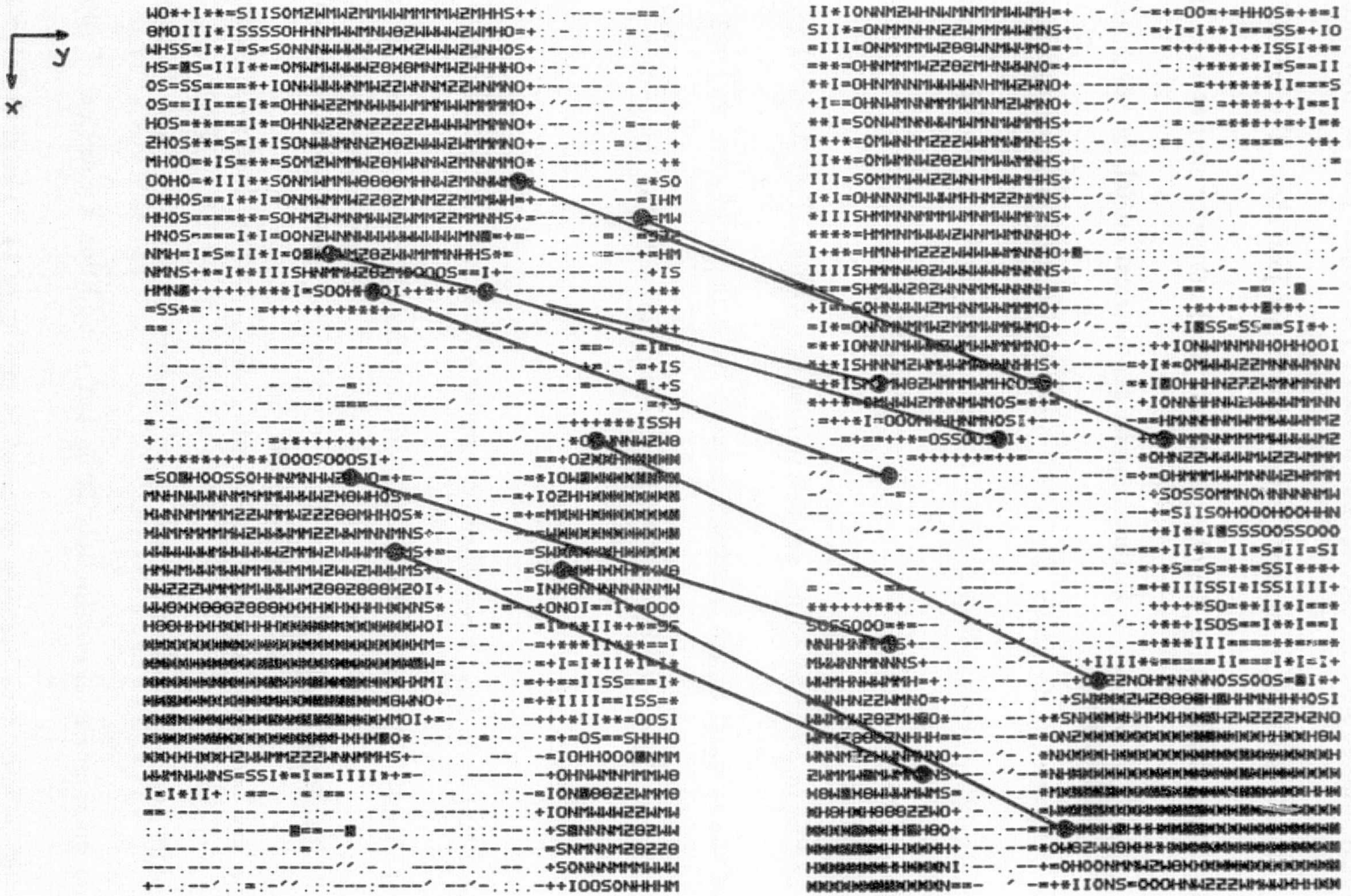


Figure 12 Example 1: Result of Correspondence Algorithm.

most similar objects are also correspondent ones. The object pair 3(8,10) with the largest residuals is found by chance, as both points are just above the level of distinctness. But observe that the objects in pair (5,5) are more similar than those in pair (6,5). The context, i. e. the common geometrical model, however, selects the pair (6,5) due to its better fit, which seem to be reasonable as can be seen from Figure 12. The final transformation parameters show scale differences up to 20% between the two images.

2. The second example is based on an image from the Arizona Test Area. The resolution of the original image has been reduced by a factor of two, yielding pixel sizes of 50 μm. The selected windows of 80x80 pixels with the interesting points are shown in Figure 13.

39 and 50 points have been selected, almost all having error ellipses close to a circle (cf. Table 11). From the 1950 possible pairs 127 were retained as candidates (cf. Table 12). Observe that the weights in this case do not vary so much as in the first example, and are considerably smaller. The final result yields 18 object pairs and is shown in Figure 14 (cf. Tables 13 and 14). Also in this case the scale difference is approximately 20%, but in addition a rotation of approximately 10° in both axes becomes apparent from Figure 14. The shifts of +4 and +18 pixels correspond to an overlap of the two windows of approximately 70%.

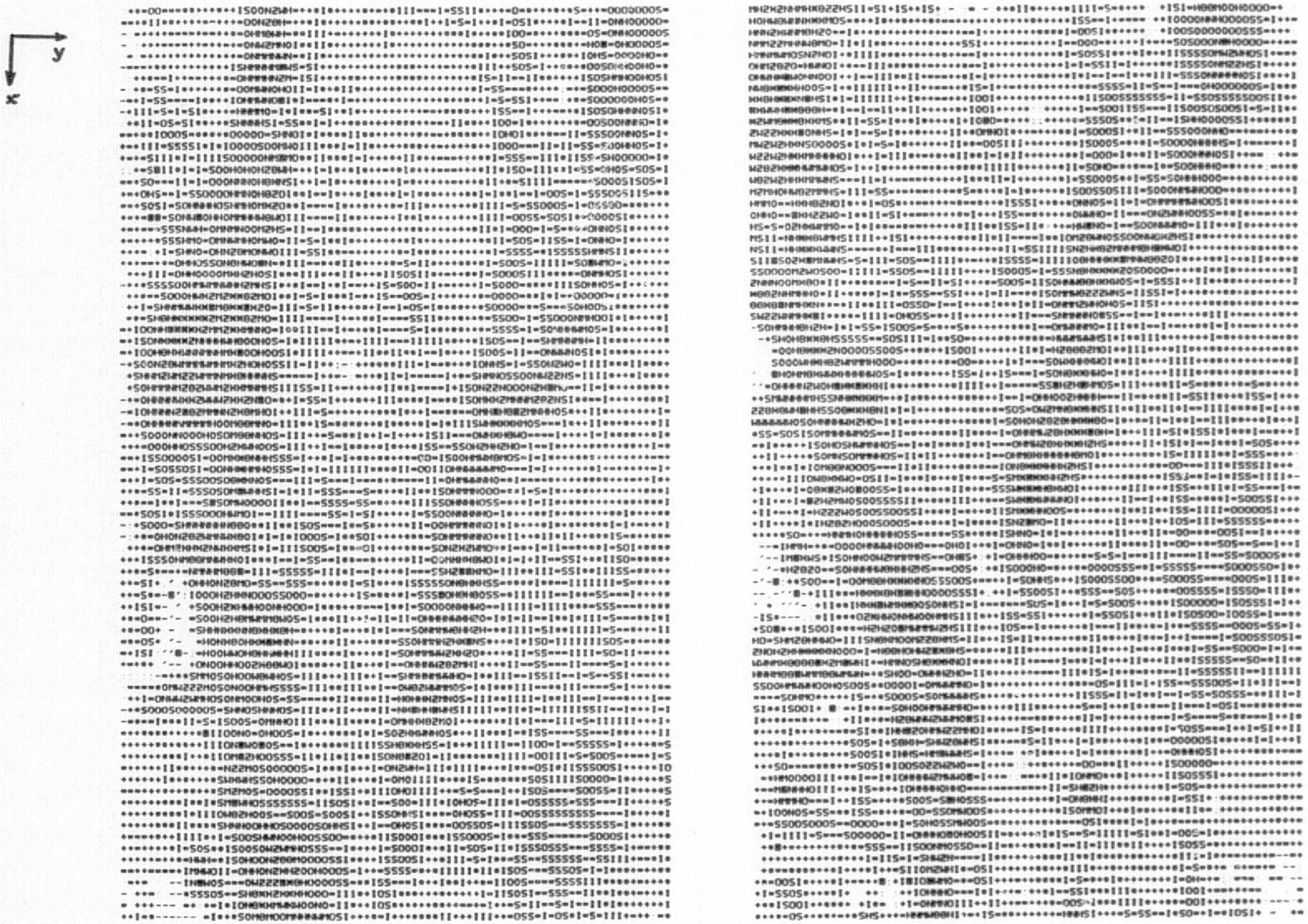


Figure 13 Example 1: Image Pair from Arizona Test Area With Selected Points (black pixels).

ORIGINAL PAGE IS OF POOR QUALITY

Left Image

i	x	y	$\overline{w}$	q
1	4	71	280	79
2	6	24	439	83
3	9	24	392	82
4	14	24	399	61
5	15	6	184	66
6	19	5	271	81
7	19	6	271	84
8	19	12	327	89
9	23	21	319	57
10	23	69	196	68
11	27	14	520	85
12	27	19	385	64
13	29	8	802	85
14	31	18	320	55
15	34	63	287	53
16	35	21	180	57
17	36	10	208	93
18	36	55	642	91
19	36	58	530	86
20	43	18	563	69
21	44	14	355	92
22	50	18	552	70
23	50	50	397	67
24	50	51	397	57
25	52	8	964	87
26	52	47	307	87
27	56	22	832	94
28	56	51	365	49
29	57	9	1258	80
30	62	43	475	68
31	62	46	355	74
32	64	13	287	95
33	65	18	381	72
34	65	21	324	93
35	66	18	381	65
36	66	42	450	58
37	70	17	519	76
38	77	12	978	91
39	77	24	660	93

Right Image

i	x	y	$\overline{w}$	q
1	4	70	206	96
2	7	6	474	72
3	9	9	673	89
4	11	35	231	75
5	12	8	660	91
6	19	7	589	91
7	20	50	355	87
8	22	60	521	75
9	23	4	496	94
10	23	9	622	87
11	23	55	475	81
12	27	5	489	88
13	28	10	363	39
14	28	52	275	95
15	33	$	950	94
16	34	13	597	96
17	34	16	778	96
18	34	45	348	87
19	34	49	335	55
20	36	8	328	98
21	38	50	612	82
22	43	12	563	86
23	43	16	261	81
24	44	9	494	89
25	46	41	669	86
26	49	7	959	74
27	49	31	197	78
28	51	4	751	91
29	52	7	899	69
30	52	22	686	88
31	53	19	629	82
32	55	4	635	77
33	55	22	568	75
34	57	27	411	74
35	58	10	617	55
36	58	14	495	68
37	59	7	392	82
38	62	12	404	67
39	63	32	204	90
40	64	23	745	98
41	66	28	330	97
42	68	32	216	84
43	69	5	355	90
44	69	49	539	92
45	70	29	310	98
46	73	29	279	88
47	74	4	224	96
48	77	19	367	96
49	77	23	350	96
50	77	26	339	92

Table 11 Example 2: List of Selected Points

NOTE: x,y coordinates

$\overline{w}$ interest value

q measure for isotropy of error ellipse (in percent)

no	i j	w	no	i j	w	no	i j	w
1	118	5.643	51	2109	9.838	101	3238	5.483
2	314	4.394	52	2115	4.774	102	3245	2.673
3	321	2.968	53	2213	18.835	103	3343	4.388
4	417	5.736	54	2227	3.340	104	3423	3.961
5	419	2.998	55	2239	4.372	105	3522	4.566
6	624	2.901	56	2242	4.866	106	3527	2.576
7	626	7.761	57	2334	5.221	107	3540	4.418
8	726	6.448	58	2340	3.506	108	3543	5.201
9	903	5.922	59	2344	6.389	109	3625	18.249
10	921	2.360	60	2350	6.412	110	3631	3.709
11	923	5.297	61	2408	2.662	111	3722	6.343
12	927	1.790	62	2419	13.608	112	3725	4.658
13	1008	2.880	63	2427	4.470	113	3726	8.724
14	1112	3.905	64	2434	4.626	114	3731	4.081
15	1205	4.182	65	2440	3.616	115	3734	4.062
16	1213	2.141	66	2444	5.546	116	3737	2.934
17	1217	4.604	67	2528	3.528	117	3740	5.449
18	1219	2.110	68	2529	4.389	118	3741	5.045
19	1227	3.880	69	2538	10.427	119	3743	4.729
20	1304	2.721	70	2548	4.565	120	3826	18.680
21	1317	4.953	71	2618	22.025	121	3834	3.980
22	1322	5.687	72	2649	4.370	122	3837	3.512
23	1326	8.102	73	2717	39.595	123	3840	9.094
24	1334	4.166	74	2719	3.926	124	3843	4.102
25	1410	6.034	75	2722	7.434	125	3930	15.418
26	1413	2.392	76	2726	7.321	126	3935	3.754
27	1425	5.155	77	2734	9.565	127	3944	8.294
28	1427	2.200	78	2735	4.318			
29	1508	13.592	79	2740	5.840			
30	1511	2.105	80	2743	6.578			
31	1521	2.783	81	2744	15.460			
32	1525	2.925	82	2814	2.866			
33	1603	9.843	83	2821	23.014			
34	1605	2.750	84	2825	3.689			
35	1621	3.211	85	2827	3.425			
36	1623	4.138	86	2839	2.692			
37	1627	2.230	87	2915	5.572			
38	1639	9.894	88	2949	4.512			
39	1726	3.290	89	3017	4.004			
40	1737	4.264	90	3031	3.112			
41	1741	2.716	91	3040	4.683			
42	1804	8.803	92	3041	3.685			
43	1807	4.351	93	3044	8.097			
44	1819	3.058	94	3050	2.444			
45	1834	4.684	95	3121	12.112			
46	1911	39.671	96	3123	2.489			
47	1921	2.798	97	3127	2.834			
48	1939	2.405	98	3130	3.692			
49	2010	17.253	99	3139	3.171			
50	2025	14.639	100	3229	2.622			

Table 12 Example 2: List of Selected Pairs

NOTE: ij point No. in left and right image (201 ≅ (2,1))
w initial weight

```
estimated shift:    -18.111     -3.333

 i  left right          xl          yl          xr          yr          dx          dy

   1   16     7     36.000      55.000      20.000      50.000       2.111      -1.667
   2   19    11     36.000      58.000      23.000      55.000       5.111       0.333
   3   24    19     50.000      51.000      34.000      49.000       2.111       1.333
   4   26    18     52.000      47.000      34.000      45.000       0.111       1.333
   5   27    17     56.000      22.000      34.000      16.000      -3.889      -2.667
   6   28    21     56.000      51.000      38.000      50.000       0.111       2.333
   7   34    23     65.000      21.000      43.000      16.000      -3.889      -1.667
   8   36    25     66.000      42.000      46.000      41.000      -1.889       2.333
   9   38    37     77.000      12.000      59.000       7.000       0.111      -1.667
```

Table 13 Example 2: Result of Robust Shift Adjustment

NOTE: 7 iterations, list had not been cleaned

$$\left(\begin{array}{ccc} 0.81926 & 0.13883 \; ; & -14.90 \\ 0.17803 & 1.17112 \; ; & -19.92 \end{array} \right) = (\hat{B} ; \hat{a})$$

```
 i  left right          xl          yl          xr          yr          dx          dy

   1   18     7     36.000      55.000      20.000      50.000       2.225       0.899
   2   19    11     36.000      58.000      23.000      55.000      -0.359      -0.588
   3   24    19     50.000      51.000      34.000      49.000      -0.861      -0.293
   4   26    18     52.000      47.000      34.000      45.000       0.222      -0.622
   5   27    17     56.000      22.000      34.000      16.000       0.029      -0.188
   6   28    21     56.000      51.000      38.000      50.000       0.055      -0.225
   7   34    23     65.000      21.000      43.000      16.000      -1.737       0.244
   8   36    25     66.000      42.000      46.000      41.000      -1.002       0.015
   9   12     5     27.000      19.000      12.000       8.000      -2.147      -0.864
  10   15     8     34.000      63.000      22.000      60.000      -0.303      -0.088
  11   20    10     43.000      18.000      23.000       9.000      -0.177      -0.186
  12   21     9     44.000      14.000      23.000       4.000       0.087       0.307
  13   22    13     50.000      18.000      28.000      10.000       0.558       0.060
  14   29    15     57.000       9.000      33.000       4.000       0.043      -3.234
  15   35    22     66.000      18.000      43.000      12.000      -1.334       0.908
  16   38    26     77.000      12.000      49.000       7.000       0.845       0.840
  17    9     3     23.000      21.000       9.000       9.000      -2.146      -0.234
  18   39    30     77.000      24.000      52.000      22.000      -0.489      -0.107
```

Table 14 Example 2: Result of Robust Affine Transformation

NOTE: 6 iterations, cleaned list (final result), only 2 correspondencies were ambiguous.

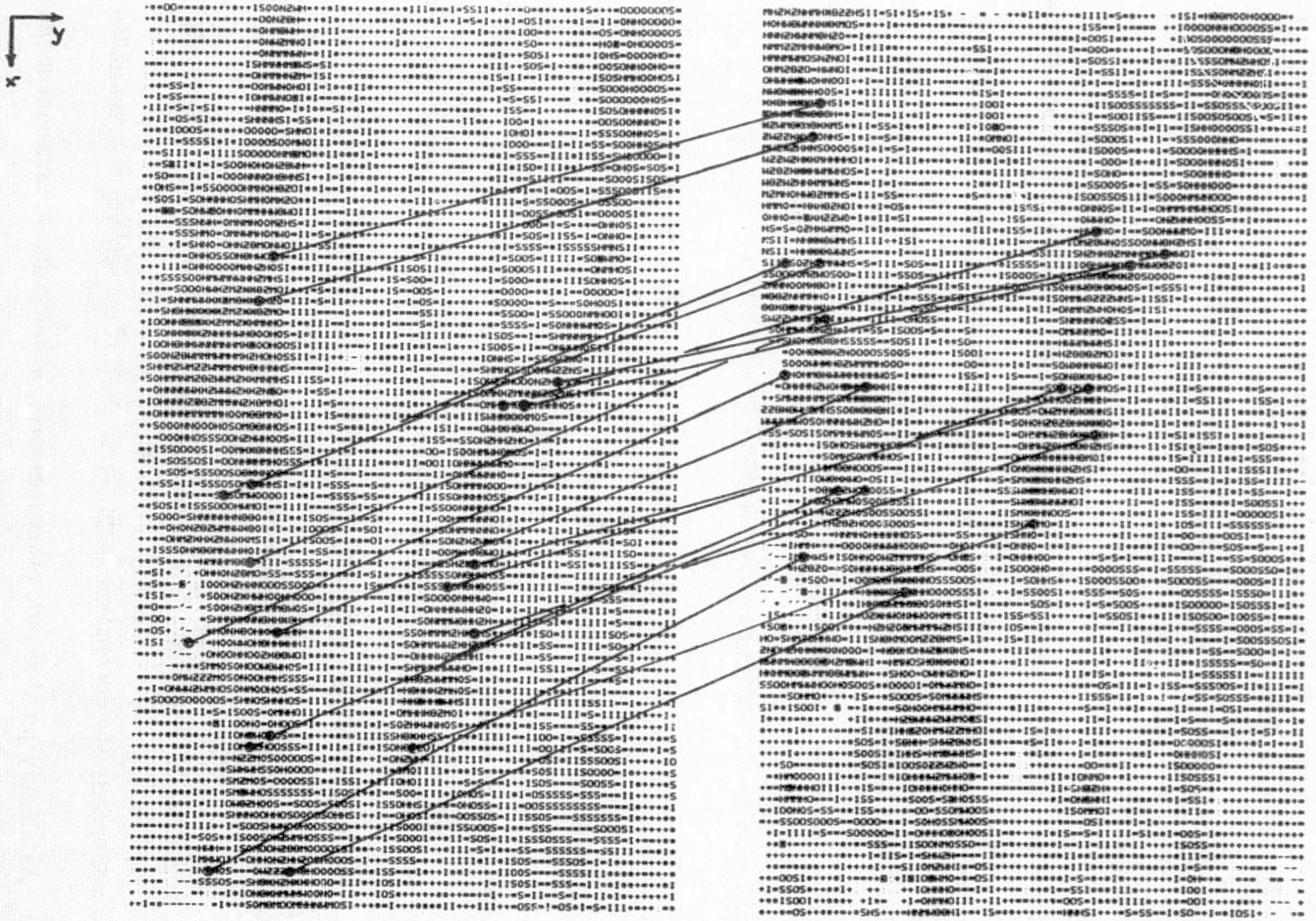


Figure 14 Example 2: Result of Correspondence Algorithm

4.6 Preliminary Conclusions

From other experiments with simulated and real data the following preliminary conclusions may be drawn:

a. The algorithm in its present form works well if the relative distortions of the images are not larger than 20-30% (corresponding to a rotation of up to 20°) and the overlapping area contains enough distinct points. These conditions can always be met if an operator provides the approximate values, or the images are oriented with an automatic procedure as the LNK-Method.

b. The results are accurate up to 1-2 pixels, if the deviation of the geometrical model from the real distortion is not too large.

c. The algorithm is fast enough to replace the first iterations in a correlation-based algorithm for high precision registration or rectification. The total computing time for a pair of images with 128x128 pixels is approximately 2-3 seconds on a VAX 11/780 and is nearly proportional to the number of pixels.

d. The limitation of the algorithm in its present form results from the similarity measure, namely the correlation coefficient, which is not scale or rotation independent. As already pointed out, other measures, as for example invariant moments might solve this problem.

e. Further research should be directed towards a link with the features of the LNK-method. There are two ways to do that which are complementary: One could use robust estimation procedures to refine the estimation of the LNK-method and one could use the abstract features, especially the abstract lines, as input for the correspondence algorithm. In this case each abstract line would give rise to four observation equations, derived from the coordinates of one end point of the line and the coordinate differences to the other end point. If consideration is restricted to rotation and scale differences only, the angular difference and the logarithm of the scale ratio of the pairs of abstract edges would lead to a robust estimation of the means of the shift, the rotation and the logarithm of the scale difference of the two images. The inclusion of line features into the algorithm would allow its application for rectification of satellite images.

5. CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

From the research performed so far, the following are the conclusions to be drawn:

1. The collinearity (or parametric) model is superior to the polynomial (or interpolative) model particularly when the number of control points is small.

2. Through simulations, it is shown that the parametric model adequately describes the real data.

3. Rectification of single image scanner data is more sensitive to image position errors than ground position errors.

4. Uncertainty in attitude estimate is the main source of error in system-corrected images.

5. In general, when more than about 25 well distributed control points are used, the effect in rectification accuracy is marginal.

6. The distribution of control features is critical to the rectification accuracy; to obtain the same accuracy about three times the number of well distributed control features are needed when such features are randomly distributed.

7. The block adjustment procedure based in the parametric rectification model was successful. Tie points between overlapping images improved rectification accuracy, particularly when few control points are used.

8. Edges proved to be an effective type of control for single image rectification. In general, about three edge pairs are needed for each conventional control point.

9. An efficient new algorithm for finding corresponding points in image pairs has been developed. The unknown parameters of the geometric transformation between the two images are derived using robust estimation techniques.

10. Tests with simulated and real data show that the present correspondence algorithm can accomodate geometric distortions up to 20 to 30 %, which corresponds to an average distortion of 3 to 7 pixels in an image of size 128x128 pixels.

11. The correspondence algorithm incorporates a new operator for finding distinct objects in an image based on the expected precision of locating such object by cross correlation.

5.2 Recommendations

1. Continue to investigate other non-conventional control such as geometric constraints and relative control (e. g. distances, angles etc.).

2. Extend the block adjustment program to accomodate edge control and perform tests.

3. Continue to develop the correspondence algorithm and apply it to remote sensing data both for registration and rectification.

4. Study the rectification/registration sequence.

5. Investigate rectification accuracy assessment.

6. Analyse blunder detection and identification procedures.

7. Research the problem of merging remote sensing data and digital terrain models.

6. ACKNOWLEDGEMENTS

This research was supported by NASA Contract No. 9-16664 through Subcontract No. L200074 with Texas A&M Research Foundation. The authors wish to thank Dr. R. P. Heydorn of NASA Johnson Space Center and Professor L. F. Guseman, Jr., of Texas A&M University for their cooperation and counsel. They would also like to express their appreciation to Dr. D. W. Mooneyhan and Dr. David Dow of NASA National Space Technology Laboratories for providing the two real data sets.

References

1. Bähr, H. P., "Geometrical Analysis and Rectification of LANDSAT MSS Imagery: Comparison of Different Methods", Nachr. Kart. u. Vermess. wes., No. 36, Institut fur Angewandte Geodasie, Frankfurt, 1978, pp. 25-46.
2. Bähr, H. P., "Geometrical Models for Satellite Scanner Imagery", ISP Commission III Congress, Helsinki, July, 1976.
3. Bähr, H. P., "Interpolation and Filtering of ERTS Imagery", ISP Commission III Symposium, Stuttgart, September, 1974.
4. Barnard, S. T. and Thompson, W. B., "Disparity Analysis of Images", IEEE, Vol. PAMI-2, 1981, pp. 333-340.
5. Barnea, D. T. and Silverman, H. F., "A Class of Algorithmus for Fast Digital Image Registration, IEEE, Vol. C-21, 1972, pp. 179-186.
6. Bernstein, R., "Digital Image Processing of Earth Observation Sensor Data", IBM J. Research Development, January, 1976, pp. 40-57.
7. Bernstein, R., "Scene Correction (Precision Processing) of ERTS Sensor Data Using Digital Image Processing Techniques", 3rd ERTS Symposium, Vol. 1-A, NASA SP-351, 1973.
8. Cafforio, C. and Rocca F., "Tracking Moving Objects in Television Images", Signal Processing 1, 1979.
9. 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.
10. 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.
11. Ewing, C. E. and Mitchell, M. M., *Introduction to Geodesy,* New York: American Elsevier, 1969.
12. Forrest, R. B., "Simulation of Orbital Image-Sensor Geometry", Photogrammetric Engineering and Remote Sensing, Volume 47, Number 8, August, 1981, pp. 1187-1193.

13. Forrest, R. B., "Geometric Correction of ERTS-1 MSS Images",ISP Commission III Symposium, Stuttgart, September, 1974.

14. Förstner, W., "Quality Assessment of Object Location and Point Transfer Using Digital Image Correlation Techniques", ISP Commission III Congress, International Archives of Photogrammetry, Volume 25-III, Rio de Janeiro, July, 1984.

15. Förstner, W., "On the Geometric Precision of Digital Correlation", International Archives for Photogrammetry, Vol. 24-III, Helsinki, 1982, pp. 176-189.

16. Friedmann, D. E. , Friedel, J. P., Magnussen, K. L., Kwok, R., and Richardson, S., "Multiple Scene Precision Rectification of Spaceborne Imagery With Very Few Ground Control Points", Photogrammetric Engineering and Remote Sensing, Volume 48, Number 12, December, 1983, pp. 1657-1667.

17. Grimson, W. E. L., *From Images to Surfaces: A Computational Study of the Human Early Visual System*, M.I.T. Press, 1981.

18. Hampel, F. R., "Robust Estimation: A Condensed Partial Survey", Zeitschrift fur Wahrscheinlichkeits Theorie, 1973.

19. Huber, P. J., *Robust Statistics*, New York, 1981.

20. Kak, A. C., *Depth Perception for Robots*, School of Electrical Engineering, Purdue University, TR-EE 83-44, 1983.

21. Konecny, G., "Mathematical Models and Procedures for the Geometric Restitution of Remote Sensing Imagery", ISP Commission III Congress, Helsinki, July, 1976.

22. Krarup, T., Juhl, J., and Kubik, K., "Gotterdammerung Over Least Squares", International Archives for Photogrammetry, Vol. 23-B3, Humburg, 1980, pp. 369-378.

23. Kratky, V., "Photogrammetric Solution for Precision Processing of ERTS Images", ISP Commission II Congress, Ottawa, July, 1972.

24. Kuglin, C. D. and Hines, D. C., "The Phase Correlation Image Alignment Method", Proceedings of the IEEE International Conference on Cybernetics and Society, 1975, pp. 163-165.

25. Lambird, B. A., Lavine, D., Stockmann, G. C., Hayes, K. C., and Kanal, L. N., Study of Digital Matching of Dissimilar Images, ETL Report, No. ETL-0248, 1980.

26. 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.

27. Levine, I., "The MSS Control Point Location Error Filter for LANDSAT-D", Sixth Annual Flight Mechanics/Estimation Theory Symposium, NASA/GSFC, October, 1981.

28. Marr, D., *Vision: A computational Investigation of the Human Representation and Processing of Visual Information*, W. H. Freemann, San Francisco, 1981.

29. McGillem, C. D. and Svedlow, M., "Image Registration Error Variance as a Measure of Overlay Quality", IEEE, Vol. GE-14, 1976, pp. 44-49.

30. McGlone, J. C. and Mikhail, E. M., *Photogrammetric Analysis of Aircraft Multispectral Scanner Data*, Technical Report Number CE-PH-81-3, School of Civil Engineering, Purdue University, West Lafayette, Indiana, October, 1981.

31. Mikhail, E. M., Akey, M. L., and Mitchell, O. R., "Detection and Sub-Pixel Location of Photogrammetric Targets in Digital Images", Presented Paper, Specialist Workshop on Pattern Recognition in Photogrammetry, Graz, 1983.

32. Mikhail, E. M. and Paderes, F. C., "Simulation Aspects in the study of Rectification of Satellite Scanner Data", Proceedings of the NASA Symposium on Mathematical Pattern Recognition and Image Analysis, Johnson Space Center, Houston, Texas, June, 1983, pp. 413-483.

33. 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.

34. Mikhail, E. M., *Observations and Least Squares*, New York: Harper and Row, 1976.

35. Panton, D. J., "A Flexible Approach to Digital Stereo Mapping", Proceedings of the DTM Symposium, ASP, St. Louis, 1978, pp. 32-60.

36. Puccinelli, E. F., "Ground Location of Satellite Scanner Data", Photogrammetric Engineering and Remote Sensing, Volume 42, Number 4, April, 1976, pp. 537-543.

37. 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.

38. Rosenfeld, A., Hummel, R. A., and Zucker, S. W., "Scene Labeling by Relaxation Operations", IEEE, Vol. SMC-6, 1976, pp. 420-433.

39. Sawada, N., Kidode, M., Shinoda, H., Asada, H., Iwanaga, M., Watanabe, S.,Mori, K., and Akiyama, M., "An Analytic Correction Method for Satellite MSS Geometric Distortions", Photogrammetric Engineering and Remote Sensing, Volume 47, Number 8 ,August, 1981, pp. 1195-1203.

40. Stockmann, G. C., Kopstein, S., and Benett, S., "Matching Images to Models for Registration and Object Location via Clustering", IEEE, Vol. PAMI-4, No. 3, 1982, pp. 229-241.

41. Synder, J. P., "Geometry of a Mapping Satellite", Photogrammetric Engineering and Remote Sensing, Volume 48, Number 10, October, 1982, pp. 1593-1602.

42. Trinder, J. C. and Nasca, S. U., "Tests on the Mapping Application of LANDSAT Imagery", ISP Commission III Congress, Helsinki, July, 1976.

43. Werner, H., "Automatic Gross Error Detection by Robust Estimators", International Archives for Photogrammetry, Vol. 25-III, Rio de Janeiro, 1984.

44. Wong, R. and Hall, E., "Scene Matching with Invariant Moments", CGIP, Vol. 8, 1978, p. 16.

THE INFLUENCE OF THE NUMBER OF GROUND CONTROL POINTS ON THE SCENE-TO-MAP REGISTRATION ACCURACY

David D. Dow

National Aeronautics and Space Administration
Earth Resources Laboratory
National Space Technology Laboratories
NSTL Station, Mississippi 39529

ABSTRACT

An investigation of the optimum number of ground control points required to rectify a full scene or a portion of a Landsat MSS scene was conducted on data from southeastern Louisiana/southwestern Mississippi and eastern Kansas. The ground control points utilized were randomly distributed across the partial or full scene. This work suggest that 24 ground control points is more than adequate to rectify a partial or full scene of Landsat MSS data. An additional study examined the error incurred in choosing ground control points representing artificial versus natural features.

Introduction

This study involves an investigation of the geometric accuracy of scene-to-map registration products of Landsat multispectral scanner (MSS) data. The rectification of Landsat MSS data to a Universal Transverse Mercator (UTM) Map base is an important pre-processing step in the analysis of earth resources science data. Potential applications that flow from an accurate scene-to-map rectification process includes:

1. component of a multisource data base
2. development of change detection products
3. input to a habitat classification

The accuracy with which ground control points (GCP) can be selected is an important source of error in the construction of the mapping equations which relate Landsat scene coordinates to map coordinates (northings and eastings in the UTM system). The use of a non-linear transformation in the mapping equations may not be justified, when one considers the accuracy with which ground control points can be selected (Steiner and Kirby [9]). A study of ground control point selection accuracy revealed that (Steiner and Kirby [9]):

1. GCPs can be selected more accurately on maps than Landsat images
2. GCPs can be measured more accuraely on man-made features (road intersection) than on natural features (land-water interfaces).

A commonly utilized mapping equation is the affine transformation which is equivalent to a first degree polynomial. The properties of the affine transformation in relation to geometrical rectification of Landsat data are discussed by Kirby and Steiner [5], Steiner and Kirby [9], Van Wie and Stein [10], Horn and Woodham [4], Emmert and McGillem [2], and Wong [12]. In comparing the UTM map control points and the Landsat scene control points of the same objects utilizing an affine transformation, the linear least squares approximation is used which generates residuals which measure how well the data fits the mapping equation. The root mean square (RMS) value is a measure of the degree of fit. The residuals stem from nonlinear distortions in satellite orbit and attitude, errors attributable to curvature of lines due to earth rotation and map projection, scanner mirror velocity non-linearity, and random variation. The affine transformation accounts for distortions due to: translation, scale change, rotation, aspect ratio, and skew (Van Wie and Stein [10]). An analysis of two Landsat MSS frames of the component sources of error in the residual error term found that the "other" category (attitude errors) were generally larger than the transformation error component or the point measurement error component (Steiner and Kirby [9]). Another source of distortion considered by the same authors (Kirby and Steiner [5]) is the differences in geometry between the UTM projection and the Landsat MSS scene. The affine transformation does not compensate for this distortion (called geometric base problem).

A number of investigators have employed polynomials of a higher degree as mapping equations. Wong [12] reported an RMS value of ±57m for a 20 term polynomial, while the RMS value of a first degree polynomial applied to the same Landsat frame was ±115m. There is a tradeoff involved, however, in that at least 20 GCPs must be used per frame to provide a least squares solution for a 20 term polynomial (up to 30 GCPs would have to be used in practice). Not only must the higher degree polynomial use a large number of GCPs, but the GCPs must be well distributed near the edges and corners of the frame (Van Wie and Stein [10]). For products issued by the EROS master data processor (MDP) to produce P-format tapes (spatially and radiometrically corrected), the number of GCPs used can be related to the scene-to-map registration accuracy. If 25 to 50 GCPs are used, the rectification accuracy should be within 1 pixel more than 99% of the time. For 8 to 24 GCPs the rectification accuracy should be within 10 pixels, while for 1 to 7 GCPs the rectification accuracy should be within 20 pixels more than 99% of the time (Nelson and Grebowsky [7]).

A recent study of Landsat-4 P-format rectification accuracy analyzed the sources of error due to locating GCPs accurately, digitizing and map distortions, and relief variations (Welch and Usery [11]). For MSS data the location error was ±30-40m (rmse_{xy} or root mean square error vector), the map and digitizing error was ±10-15m, and the terrain relief error was roughly ±10-30m. The root mean square error vector is computed from the deviation between the mapping equation and the withheld ground test point locations. When a first degree polynomials was employed with 10 or more GCPs for a whole scene, the rmse_{xy} value was ±80m. The use of a third degree polynomials with 30 or more GCPs produced an rmse_{xy} value of ±55m. For a 1024 by 1024 pixel area, 15 GCPs were used with first and second degree polynomials to produce a minimum rmse_{xy} value of ±45m. The rmse $_{xy}$ value increased to ±60m when only 10 GCPs were utilized. For a 256 by 256 pixel area, 5 GCPs and a linear first order polynomial yielded an rmse $_{xy}$ value of ±40m.

The affine transformation and higher degree polynomials are an example of interpolative or surface fitting models. The other type of model used in rectification is the parametric model which incorporates information on satellite position and sensor attitude (Mikhail and Paderes [6]). The use of parametric modelling in rectification has been described by Mikhail and Paderes [6], Horn and Woodham [4], and Sawada et. al [8]. Parametric modelling includes two components: sensor modelling and platform modelling. Sensor modelling corrects for panoramic effect (pixel projection on a plane), non-linearity of scanning, and unequal number of pixels per scan. Platform modelling deals with problems associated with sensor attitude and the satellite position in orbit. Mikhail and Paderes [6] describe this approach in some detail. The satellite collinearity equation was used to combine the sensor and platform models. Ground control points are used to estimate the unknown parameters in the sensor and platform models (19 unknown parameters existed in this study). The conclusions of the investigation conducted by Mikhail and Paderes [6] include:

1. The maximum rectification accuracy for a polynomial model is about half a pixel.
2. Rectification accuracy is not significantly improved when the number of GCPs utilized exceeds 25.

3. The rectification accuracy is sensitive to the identification accuracy of a GCP in the Landsat image, but is insensitive to the accuracy of identifying a GCP on the map.

Recent work by Mikhail and Paderes (personal communication) reported that the collinearity model gave equal or lower RMS values for the same number of ground control points than did a polynomial model. The differences in RMS values between the collinearity and polynomial models is more pronounced for 10 GCPs than it is for greater than 40 GCPs. The same conclusions were arrived at using synthetic data as were determined from using real Landsat MSS data from Kansas and Louisiana.

Methods

The Landsat MSS frames used in this study were acquired over path:23 and row:39 of the worldwide reference system (southeastern Louisiana - coastal Mississippi) and over path:29 and row:33 (western Missouri - eastern Kansas). The Kansas data was collected on 11/9/81, while the Louisiana data was collected on 11/21/81. Both Landsat MSS scenes had 10% cloud cover. The Louisiana scene was relatively flat (elevation: 0 to 362 feet above sea level) and contained up to 35% open water. The Kansas scene was hilly (elevation: 730 to 1450 feet above sea level) with neglible amounts of open water. The extensive amount of open water and wetlands in the Louisiana scene present a significant challenge for accurate rectification when compared to the Kansas Landsat frame.

The points to be utilized as ground control points (GCPs) and ground reference points (GRPs) were chosen on 1:24,000 scale, 7.5 minute quadrangle sheets produced by the U.S. Geological Survey (USGS). The GCPs are used to generate the mapping equations used in the georegistration procedure, while the GRPs were employed as test points to independently assess the accuracy of the georegistration procedure. The ground points map coordinates were recorded in the UTM system as northings and eastings, while the Landsat scene coordinates were recorded as rows and elements. The same points were identified on the 7.5 minute USGS quadrangle sheet and the Landsat A-format MSS scene. The types of features used as ground points included manmade (road intersections) and natural (river intersections) categories. Table 1 gives some examples of ground points utilized in the Louisiana Landsat frame.

For the whole scene analysis, 356 ground points were used in the Louisiana data set and 359 ground points were used in the Kansas data set. For the half scene analysis the number of ground points available was 242 for Louisiana and 241 for Kansas. For the quarter scene analysis the number of ground points utilized was 182 to 198 for Louisiana (Areas A and B) and 150 to 158 for Kansas (Areas A and B). The ground points available were divided into GCPs and GRPs.

The mapping equation utilized was a linear polynomial and the fit of the GCPs to the mapping equation was quantified by the computation of the RMS value (in meters). To evaluate the georegistration accuracy of the Landsat MSS product, the procedure of Graham and

Luebbe [3] was employed. This procedure quantifies the georegistration accuracy in terms of RBIAS (row offset), CBIAS (column offset), RSD (row standard deviation), and CSD (column standard deviation). Good georegistration accuracy would be characterized by sub-pixel offsets and standard deviation values. The equations for computing bias and standard deviation are:

$$(1)\ \text{RBIAS} = \frac{\sum_{i=1}^{NP} (\text{ROW1}_i - \text{ROW2}_i)}{NP}$$

$$(2)\ \text{RSD} = \sqrt{\frac{\sum_{i=1}^{NP} (\text{ROW1}_i - \text{ROW2}_i - \text{RBIAS})^2}{NP - 1}}$$

where NP is the number of GRPs chosen, ROW1 is the Landsat row predicted from the mapping equation, and ROW2 is the Landsat row read from the MSS imagery. The units of RBIAS and RSD are in pixels.

The ERL computer software module GNRI was utilized to take a random sample of GCPs from the overall ground point list for both the Kansas and Louisiana data sets. The random samples were chosen in groups of eight and combined with the previously chosen GCPs. Groups of eight were utilized because the quality assessment number associated with Landsat P-format MSS tape's registration accuracy employs multiples of eight. The module CSPA was utilized to compute "R" values which give a measure of the spatial distribution of ground control points (Dow [1]). For the purposes of this paper "R" values between 0.7 and 1.3 are indicative of a random spatial distribution. The module BMGC was utilized to compute the bias and standard deviation values as well as the RMS numbers.

A separate test was conducted, in conjunction with the overall study, to see how accurately a given ground control point could be selected. Ten ground control points representing natural features and ten ground control points representing manmade features were picked at random from the Louisiana P-format MSS data set. This experiment was the only case in this paper in which P-format data was used. These ground control points were reselected ten times in order to see how much operator error was introduced in GCP selection. The operator located the ground control point on a 1:24,000 scale USGS map and then moved the track ball cursor on the digital image display device until the same ground point was identified on the display screen. This procedure was replicatd ten times with the data processor, who recorded the Landsat scene coordinates location, not informing the track ball operator of the results. Table 1 lists the characteristics of the ground control points used in this study.

Most of the statistical analysis utilized in this report was generated using the BMDP Statistical Package (Dixon et al. [0]). The descriptive statistics (mean, standard deviation, standard error of the mean) and analysis of variance were run using program BMDP7D. The analysis of variance model was tested for equality of variances using Levene's test and if the Levene's test results were statistically significant, then the Brown-Forsythe procedure was used for the analysis of variance computations (Dixon et al. [0]). The Duncan's multiple range test was employed to separate out significant treatment effects in those cases where the analysis of variance results were statistically significant at the 5% level. The correlation analysis was carried out with program BMDP6D.

Results and Discussion

The ground control point accuracy experiment found a row bias of 0.04 pixels for natural features and a column bias of 0.12 pixels. For manmade features the row bias was 0.12 pixels and the column bias was 0.04 pixels. It appears from these results that manmade and natural features can be chosen with equal accuracy. Also the operator bias in ground control point selection does not represent a serious source of error in the scene-to-map registration procedure.

The results of the optimum number of GCPs needed to rectify a given portion of a Landsat MSS scene in Louisiana and Kansas is given in Tables 2 through 6. The "N" column gives the number of ground control points used to develop the mapping equation. The "R" column gives an indication of the type of spatial distribution that the GCPs exhibit across the Landsat scene. The "RMS" is a measure of how well the GCPs fit the mapping equation. The accuracy of the georegistration procedure is measured by the RBIAS, RSD, CBIAS and CSD values (measured as fractions of a pixel). The bias and standard deviation values are computed from the GRPs. The values in the last row of each column represent the mean and 95% confidence interval about the mean. This row is presented for a general descriptive overview of the results, but should not be interpreted literally in those cases where the analysis of variance (ANOVA) results are statistically significant (indicated by *). The results presented represent the outcome of 40 replicates for each the "N" equals 8 through 40.

For the whole scene analysis the R values for Kansas are larger than those for Louisiana (see Table 2). The reason for this is that the Louisiana scene has large areas of open water in which it is impossible to choose GCPs. The RMS column shows what appears to be a counter-intuitive result in that the RMS value goes up as the number of GCPs utilized increases from 8 to 40. The reason for this appears to be that as the number of GCPs increases, it is more likely to encounter outlier GCPs which distort the overall RMS value. Another possible contibuting factor is that for N equals 8, there is only one degree of freedom left over to make the estimate of the mean and thus the mapping equation lacks the redundancy in GCPs necessary to make a precise estimate of the mean. The RBIAS and CBIAS values decrease in magnitude as the number of GCPs used (N) increases. In this case outliers do not distort the results because there are many more GRPs used to check rectification accuracy than the GCPs employed to generate the mapping equation (GRPs = total available ground points - GCPs). The RSD and CSD values are fairly constant in magnitude with increasing N values. This being the case it was decided to concentrate on the RBIAS and CBIAS values in order to decide what the optimum number of GCPs required to register a whole scene of Landsast data was.

The results of the Duncan's Multiple range test were utilized to choose the optimum number of GCPs required. The N values for which the Duncan's multiple range test gave significant differences were separated out from those treatment effects which were non-significant. The range of N values which were not significant were viewed as delineating the number of GCPs which gave equivalent results and the optimum number of GCPs was the lowest N which gave non-significant results. For a whole scene the optimum number of GCPs for the Louisiana data was N=24(RBIAS) and N=16(CBIAS), while for Kansas the results were N=24(RBIAS) and N=32(CBIAS). The RBIAS, RSD, CBIAS, and CSD values were roughly the same for both the Louisiana and Kansas data sets. In both data sets the RBIAS and RSD numbers were less than the CBIAS and CSD values. Thus, registration accuracy is more accurate in the row direction than in the column direction. Good rectification accuracy is indicated by the sub-pixel bias and standard deviation values for both Louisiana and Kansas.

The results of the half scene analysis are presented in Table 3. The RMS values show the usual trend of increasing as the number of GCPs utilized increases. The RBIAS and CBIAS values decrease as the N value increases. For the Louisiana Landsat frame the RMS, RBIAS and CBIAS values are less in the half scene analysis than is the case for the whole scene analysis, but the Kansas data is the same for these parameters in the half and whole scene analysis. The optimum number of GCPs for the Louisiana data in the half scene analysis is N=16 (CBIAS and RBIAS), while the results for Kansas are N=16(RBIAS) and N=24(CBIAS). Once again the sub-pixel bias and standard deviation values indicate that good scene-to-map registration accuracy has been obtained.

Tables 4 through 6 present the results of the quarter scene analysis for two different areas (A and B) of the Landsat MSS Frame. Both areas A and area B in Louisiana were chosen in such a manner that a minimal amount of open water was available. For both area A and area B in Louisiana and Kansas the GCPs were well distributed across the area covered (given by L, scan lines, and E, elements) which resulted in R values in excess of 0.94. In Table 4 which presents the Area B results for Louisiana and Kansas, the R value is higher and the RMS value is lower than is the case for the half or whole scene analysis. The RMS values increase in magnitude as the N value increases, while the bias (row and column) figures decrease in value as N increases. The explanation for this phenomena is the same one that was provided earlier. The optimum number of GCPs required to rectify a quarter of a Landsat scene are for Louisiana N=24(RBIAS) and N=16(CBIAS), while for Kansas the numbers are N=16(RBIAS) and N=8(CBIAS). Once again good scene-to-map registration accuracy is indicated by sub-pixel bias and standard deviation values.

Tables 5 and 6 compare areas A and B for Louisiana and Kansas. For Louisiana the RMS, bias, and standard deviation values are higher in magnitude for area A than they are for area B (see Table 5). These numbers are the same for areas A and B in Kansas (see Table 6). The trends in RMS, bias and standard deviation values are the same for area A for both Kansas and Louisiana as those previously explained for area B for both Landsat frames. Thus analysis of two different portions of a Landsat frame yielded similar results which suggests that the previous conclusions may not be data specific.

Tables 7 and 8 present the results of correlation and regression analysis for Louisiana and Kansas for data from the whole scene analysis. The columns represent the dependent and independent variables (y and x) in the regression equation, r is the correlation coefficient (the square of r represents the amount of the total variation explained by the regression analysis), m is the slope of the regression equation, and b is the intercept of the regression equation. The last column presents the statistical significance of r (non-sigificant= N.S.; or significant at the 1% or 5% levels). In this analysis the r value may be significant at the 1% or 5% levels because of the large number of replicates employed, but the regression equation may not be meaningful because of low values of r and r^2. It was decided that the later situation prevailed in this data. It was concluded that there is no apparent relationship between RMS values and the bias and standard deviation figures. Furthermore there is no apparent relationship between CBIAS and RBIAS or CSD and RSD. This suggests that all of these variables (RMS, RBIAS, RSD, CBIAS, and CSD) are independent of one another and that the variables measure different properties of the scene-to-map registration process. One would expect this result from the background information discussed in the introduction.

Conclusions

The ground control point accuracy experiment quantified the error associated with choosing GCPs. This error did not seem to differ between manmade and natural features. The RMS values increased in magnitude as the N number increased, while the bias and standard deviation values decreased as N increased. This result coupled with the correlation/regression analysis suggested that the RMS number measures a different property of the scene-to-map, registration process (how well GCPs fit the mapping equation) than does the bias and standard deviation figures. The bias and standard deviation values should be utilized to estimate the accuracy of the scene rectification process. It appears from this study that 24 GCPs should be more than adequate to rectify a Landsat scene-to-map, for portions of a Landsat frame (quarter of a scene up to a whole scene) using a relatively simple linear polynomial as a mapping equation. It is possible that more complex mapping equations may yield better results, a consideration if one will be performing scene-to-map registration of Landsat thematic mapper data (30 meter pixels) in the future.

References

[0] Dixon, W.J., M.B. Brown, L. Engelman, J.W. Frane, M.A. Hill, R.I. Jennrich, and J.D. Toporek. 1983. BMDP Statistical Software. Univ. of California Press; Berkeley, California; 733 p.

[1] Dow, D.D. 1983. Progress in the scene-to-map registration investigation. p. 485-506 in Proceedings of the NASA Symposium on Pattern Recognition and Image Analysis ed. by L. F. Guseman, Jr.

[2] Emmert, R.A. and C.D. McGillem. 1973. Multitemporal geometric distortion correction utilizing the affine transformation. p. 153-161 in IEEE Conference on Machine Processing of Remotely Sensed Data.

[3] Graham, M.H. and R. Luebbe. 1981. An evaluation of MSS P-format data registration. NSTL/ERL-197; AgRISTARS Report No. DC-Y1-04069.

[4] Horn, B.K.P. and R.J. Woodham. 1979. Landsat MSS coordinate transformations. p. 59-68 in 1979 Machine Processing of Remotely Sensed Data Symposium.

[5] Kirby, M.E. and D. Steiner. 1977. A theorectical model for the evaluation of the interactions between Landsat MSS data and UTM maps in geometric transformations. p. 407-421 in Remote Sensing of Earth Resources, Volume VI; ed. by F. Shahrokhi; Univ. Tennessee, Space Institute; Tullahoma, Tn.

[6] Mikhail, E.M. and F.C. Paderes, Jr. 1983. Simulation aspects in the study of rectification of satellite scanner data. p. 413-483 in Proceedings of the NASA Symposium on Pattern Recognition and Image Analysis ed. by L. F. Guseman, Jr.

[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] Van Wie, P. and M. Stein. 1977. A Landsat digital image rectification system. IEEE Trans. Geosci. Electron. GE-15: 130-137.

[11] Welch, R. and E. L. Usery. 1984. Cartographic accuracy of Landsat-4 MSS and TM image data. IEEE Trans. Geosci. Remote Sensing GE-22:281-288.

[12] Wong, K.W. 1975. Geometric and cartographic accuracy of ERTS-1 imagery. Photogramm. Engng. and Remote Sensing 41: 621-635.

Table 1. Ground Control Points Used in Accuracy Experiment

POINT #	MAP	NATURAL/MANMADE	DESCRIPTION
191	Savannah SW	Natural	Site is the corner of a forest block
230	Bay St. Louis	Natural	Site is two tributaries joining together
047	Folsom NE	Natural	Site is a field corner against a forest
366	Happy Jack	Natural	Site is two bayous coming together
372	Point a la Hatche(62)	Natural	Site is two marsh bayous coming together
033	Ponchatoula SE	Natural	Site is two marsh bayous coming together
137	Haaswood	Natural	Site is two wetland rivers coming together
371	Lake Batola	Natural	Site is two marsh rivers coming together
209	Malheureux Point	Natural	Junction of two marsh bayous
328	Oak Mound Bayou	Natural	Junction of two marsh bayous
045	Folsom NE	Manmade	Powerline/dirt road junction
001	Franklinton SW	Manmade	Road intersection
171	Logtown	Manmade	Interstate/4-lane highway intersection
350	Vancleave (62)	Manmade	Highway intersections
317	McHenry (62)	Manmade	Highway/dirt road intersection
275	Bush	Manmade	Dirt road intersection
279	St. Tammany	Manmade	Pipeline/highway intersection
354	Pascagoula (62)	Manmade	Dirt road/highway intersection
246	Carnes NW	Manmade	Interstate/dirt road intersection
264	Bougalousa NE	Manmade	Highway/dirt road intersection

Table 2. LOUISIANA - WHOLE SCENE

(40 replicates)

N	R	RMS	RBIAS	RSD	CBIAS	CSD
8	0.77	94.58	0.38	0.06	0.82	0.14
16	0.77	119.18	0.20	0.06	0.39	0.12
24	0.73	129.02	0.17	0.06	0.42	0.12
32	0.71	132.72	0.16	0.06	0.36	0.12
40	0.71	133.95	0.14	0.06	0.37	0.12
All	*	*	*	*	*	*
	0.74 ± 0.02	121.89 ± 3.94	0.21 ± 0.03	0.060	0.47 ± 0.06	0.12 ± 0.002

*: SIGNIFICANT AT 5% LEVEL IN ANOVA

KS - WHOLE SCENE

(40 replicates)

N	R	RMS	RBIAS	RSD	CBIAS	CSD
8	0.86	112.60	0.27	0.07	0.70	0.16
16	0.85	140.80	0.21	0.06	0.44	0.14
24	0.83	144.88	0.17	0.06	0.39	0.14
32	0.82	148.72	0.16	0.06	0.30	0.14
40	0.83	146.30	0.15	0.06	0.27	0.14
All	0.84 ± 0.02	*	*	*	*	*
		138.66 ± 5.59	0.19 ± 0.02	0.06 ± 0.002	0.42 ± 0.06	0.14 ± 0.002

*: SIGNIFICANT AT 5% LEVEL IN ANOVA

Table 3.

LA. - HALF SCENE (40 REPLICATES)

N	R	RMS	RBIAS	RSD	CBIAS	CSD
8	0.75	84.18	0.19	0.06	0.54	0.14
16	0.70	100.55	0.14	0.05	0.34	0.13
24	0.67	108.65	0.12	0.05	0.31	0.12
32	0.66	110.45	0.10	0.05	0.28	0.12
40	0.66	111.50	0.10	0.05	0.27	0.13
All	0.69 ± 0.02	*	*	*	*	*
		103.07 ± 2.64	0.13 ± 0.02	0.05 ± 0.001	0.35 ± 0.04	0.13 ± 0.002

*: SIGNIFICANT AT 5% LEVEL IN ANOVA

KS. - HALF SCENE (40 REPLICATES)

N	R	RMS	RBIAS	RSD	CBIAS	CSD
8	0.90	111.55	0.43	0.08	0.70	0.16
16	0.81	133.88	0.26	0.08	0.58	0.16
24	0.79	140.38	0.22	0.08	0.44	0.16
32	0.80	143.75	0.19	0.08	0.38	0.16
40	0.79	146.30	0.16	0.08	0.33	0.16
All	*	*	*	0.08 ± 0.002	*	0.16 ± 0.002
	0.82 ± 0.02	135.17 ± 6.59	0.25 ± 0.05		0.48 ± 0.05	

*: SIGNIFICANT AT 5% LEVEL IN ANOVA

Table 4. LOUISIANA 1/4 SCENE - AREA B L:200 - 1691
E:700 - 2474

(replicates = 40)

N	R	RMS	RBIAS	RSD	CBIAS	CSD
8	1.06	60.68	0.26	0.06	0.36	0.08
16	1.07	71.70	0.17	0.06	0.22	0.07
24	1.05	75.35	0.10	0.06	0.21	0.07
32	1.00	76.00	0.12	0.06	0.21	0.07
40	1.00	76.58	0.11	0.06	0.17	0.07
All	1.03 ± 0.02	*	*	*	*	*
± 95%CI		72.06 ± 2.01	0.15 ± 0.02	0.062 ± 0.001	0.24 ± 0.02	0.074 ± 0.002

*: ANOVA SIGNIFICANT AT 5% LEVEL

KANSAS 1/4 SCENE - AREA B L:500 - 1991
E:700 - 2474

(replicates = 40)

N	R	RMS	RBIAS	RSD	CBIAS	CSD
8	1.20	53.78	0.26	0.08	0.27	0.13
16	1.15	78.48	0.15	0.07	0.25	0.13
24	1.14	85.38	0.13	0.07	0.27	0.13
32	1.14	87.23	0.12	0.07	0.24	0.13
40	1.12	88.08	0.12	0.07	0.21	0.14
All	1.15 ± 0.02	*	*	0.07 ± 0.002	0.25 ± 0.03	0.133 ± 0.002
± 95%CI		78.58 ± 5.21	0.16 ± 0.02			

*: ANOVA SIGNIFICANT AT 5% LEVEL

Table 5. Louisiana 1/4 Scene: Area A: L:1-1500 E:1-1774

N	R	RMS	RBIAS	RSD	CBIAS	CSD
8	1.05	123.60	0.48	0.12	0.84	0.23
16	1.01	131.58	0.26	0.11	0.53	0.21
24	1.03	136.25	0.21	0.12	0.37	0.22
32	1.04	140.25	0.20	0.12	0.35	0.22
40	1.06	144.85	0.19	0.12	0.35	0.23
All	1.04 ± 0.02	135.30 ± 7.38	* 0.27 ± 0.04	0.117 ± 0.002	* 0.49 ± 0.07	* 0.222 ± 0.004

*: ANOVA Significant at 5% level
No. Replicates: 40

Louisiana 1/4 Scene - Area B: L:200-1691 E:700-2474

N	R	RMS	BIAS	S.D.	BIAS	S.D.
8	1.06	60.68	0.26	0.06	0.36	0.08
16	1.07	71.70	0.17	0.06	0.22	0.07
24	1.05	75.35	0.10	0.06	0.21	0.07
32	1.00	76.00	0.12	0.06	0.21	0.07
40	1.00	76.58	0.11	0.06	0.17	0.07
All	1.03 ± 0.02	* 72.06 ± 2.01	* 0.15 ± 0.02	* 0.062 ± 0.001	* 0.24 ± 0.02	* 0.074 ± 0.002

*: ANOVA Significant at 5% Level
No. Replicates = 40

Table 6. Kansas 1/4 Scene - Area A: L:1-1500 E:1-1774

N	R	RMS	RBIAS	RS D	CBIAS	CSD
8	1.00	79.30	0.21	0.06	0.64	0.14
16	0.96	93.58	0.15	0.06	0.35	0.12
24	0.96	96.90	0.13	0.06	0.29	0.12
32	0.95	97.60	0.10	0.06	0.25	0.13
40	0.94	98.68	0.10	0.06	0.22	0.13
All	0.97 ± 0.02	* 93.21 ± 2.12	* 0.14 ± 0.02	0.059 ± 0.001	* 0.35 ± 0.04	* 0.130 ± 0.002

*: ANOVA significant at 5% level
No. Replicates: 40

Kansas 1/4 Scene - Area "B" L:500-1991 E:700-2474

N	R	RMS	BIAS	S.D.	BIAS	S.D.
8	1.20	53.78	0.26	0.08	0.27	0.13
16	1.15	78.48	0.15	0.07	0.25	0.13
24	1.14	85.38	0.13	0.07	0.27	0.13
32	1.14	87.22	0.12	0.07	0.24	0.13
40	1.12	88.08	0.12	0.07	0.21	0.14
All	1.15 ± 0.02	* 78.58 ± 5.21	* 0.16 ± 0.02	0.071 ± 0.002	0.25 ± 0.03	* 0.133 ± 0.002

*: ANOVA Significant at 5% Level

Table 7. LOUISIANA CORRELATION ANALYSIS - WHOLE Scene (N = 8 - 40, 200 replicates)

y	x	r	m	b	Significance of "r"
RBIAS	RMS	-0.193	0.0014	0.379	1%
RSD	RMS	-0.045	$-0.906\text{x}10^{-5}$	0.061	N.S.
CBIAS	RMS	-0.174	0.0027	0.801	5%
CSD	RMS	-0.529	$-0.359\text{x}10^{-3}$	0.164	1%
CBIAS	RBIAS	-0.257	0.548	0.359	1%
CSD	RSD	-0.082	0.276	0.104	N.S.

KANSAS CORRELATION ANALYSIS - WHOLE SCENE (N = 8 - 40, 200 replicates)

y	x	r	m	b	Significance of "r"
RBIAS	RMS	0.028	$0.102\text{x}10^{-3}$	0.178	N.S.
RSD	RMS	0.245	$0.830\text{x}10^{-4}$	0.051	1%
CBIAS	RMS	0.074	$0.762\text{x}10^{-3}$	0.316	N.S.
CSD	RMS	-0.222	$0.108\text{x}10^{-3}$	0.156	1%
CBIAS	RBIAS	0.249	0.701	0.287	1%
CSD	RSD	0.134	0.192	0.129	N.S.

Table 8. LOUISIANA CORRELATION ANALYSIS - WHOLE SCENE

(N = 40; 40 replicates)

y	x	r	m	b	Significance of "r"
RBIAS	RMS	0.051	$0.466\text{x}10^{-3}$	0.076	N.S.
RSD	RMS	-0.448	$-0.706\text{x}10^{-4}$	0.069	1%
CBIAS	RMS	-0.004	$-0.757\text{x}10^{-4}$	0.075	N.S.
CSD	RMS	-0.322	$-0.165\text{x}10^{-3}$	0.137	5%
CBIAS	RBIAS	-0.013	-0.028	0.374	N.S.
CSD	RSD	0.162	0.526	0.084	N.S.

KANSAS CORRELATION ANALYSIS - WHOLE SCENE

(N = 40; 40 replicates)

y	x	r	m	b	Significance of "r"
RBIAS	RMS	0.349	0.0012	-0.031	5%
RSD	RMS	-0.023	$-0.124\text{x}10^{-5}$	0.060	N.S.
CBIAS	RMS	0.144	0.0010	0.124	N.S.
CSD	RMS	-0.311	$-0.708\text{x}10^{-4}$	0.148	5%
CBIAS	RBIAS	0.054	0.109	0.258	N.S.
CSD	RSD	-0.173	-0.744	0.182	N.S.

IMAGE VARIANCE AND SPATIAL STRUCTURE IN REMOTELY SENSED SCENES

CURTIS E. WOODCOCK

DEPARTMENT OF GEOGRAPHY
BOSTON UNIVERSITY, BOSTON, MA 02135

ALAN H. STRAHLER

DEPT. OF GEOLOGY AND GEOGRAPHY, HUNTER COLLEGE
CITY UNIVERSITY OF NEW YORK, NEW YORK, NY, 10021

ABSTRACT

Empirical studies of digital images derived by scanning air photos and through acquiring aircraft and spacecraft scanner data shows that spatial structure in scenes can be measured and logically related to texture and image variance. Local variance, measured as the average standard deviation of brightness values within a three-by-three moving window, reaches a peak at a resolution cell size about two-thirds to three-fourths the size of the objects within the scene. If objects are smaller than the resolution cell size of the image, this peak does not occur and local variance simply decreases with increasing resolution as spatial averaging occurs. Variograms, which measure the average squared difference in pairs of brightness values as a function of the distance and direction between them, can also reveal the size, shape, and density of objects in the scene.

INTRODUCTION

This paper presents the continuation of research described in the 1982 issue of these Proceedings [1]. To avoid redundancy, the reader is referred to that paper for a more thorough background. The primary research goal is to develop a better understanding of spatial patterns in image data as they relate to the characteristics of the ground scene. The long term objective is to develop methods of scene inference that directly exploit the relationship between the ground scene and the spatial patterns in images. Before that goal can be accomplished, an improved understanding of the meaning of the results of various measures of spatial pattern must be developed. In particular, identifying those characteristics of the ground scene that can be recovered from measures of spatial pattern is of interest.

In this paper, images from a variety of environments and spatial resolutions are examined using two methods of measuring spatial pattern. In addition, a new direction of research that involves simulating remotely sensed images will be discussed. The use of simulated images allows for control of the ground scene, which aids the interpretation of spatial pattern measurements.

The two methods used to measure spatial patterns are (1) graphs of local variance as a function of spatial resolution, and (2) two-dimensional variograms. Local variance is measured in images as the mean value of a texture image. The texture value at a given pixel is the standard deviation of the surrounding 3 X 3 window of pixels. Note that this definition is only one of many possible definitions of texture [2]. To evaluate local variance over a range of spatial resolutions, the imagery was degraded to successively coarser resolutions by simply averaging the resolution cells to be combined into a new, larger resolution cell.

An alternative method of examining spatial structure in images is through the variogram [3]. The variogram is calculated as the mean squared deviation between two pixels a given distance and direction apart. This can be thought of as a measure of the expected difference between two pixels given the spatial relationship between them. The results of these calculations are plotted as two-dimensional contour plots. A more detailed description of both methods can be found in last year's paper.

Another brief note of background concerns the description of ground scenes. The concept of a scene model, or the development of a generalization of the nature of a scene, is essential to this project. A scene can be described as being composed of objects on a plane, or as in our case, as elements on a background. Scene models can have numerous types of elements and be very complex, and even have a nested structure in which smaller elements are used to describe or define other elements. Reference to the elements in the scene and their characteristics occurs throughout the discussion of the results.

RESULTS

Both methods of measuring spatial pattern in image data were used to evaluate spatial patterns acquired from three types of environments in images at two resolutions. Imagery was analyzed at two different resolutions in order to be able to cover a wider range of resolutions in the local variance/resolution graphs and to help illustrate that the formulation of scene models is related to the resolution of the data. The imagery at very fine resolutions for each environment was digitally scanned from aerial transparencies using a microdensitometer, thus allowing analysis of spatial pattern at finer resolutions than are available from conventional spaceborne sensors. Thematic Mapper (TM) or Thematic Mapper Simulator (TMS) data were used as the coarse resolution data for each environment. The three types of environments used in the analysis are forested, large-field agricultural, and urban/suburban. The presentation of the results and their discussion is organized around the individual images analyzed, beginning with the finer resolution imagery for each environment.

South Dakota Forest Image

The color aerial transparency of the area used to create this image was obtained from the Nationwide Forestry Applications Program office at the Johnson Space Center in Houston, Texas. The exact location of the area in South Dakota covered by the photograph is unknown, but it serves as a good example of a simple forested environment composed of trees on a relatively uniform background.

Figure 1 A-D shows the digitized photograph at the original resolution and as averaged to calculate the graph of local variance as a function of resolution (Figure 2). The graph shows that local variance is low at the resolution that the photo was scanned, or Ø.75 m (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 values 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 windows 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.

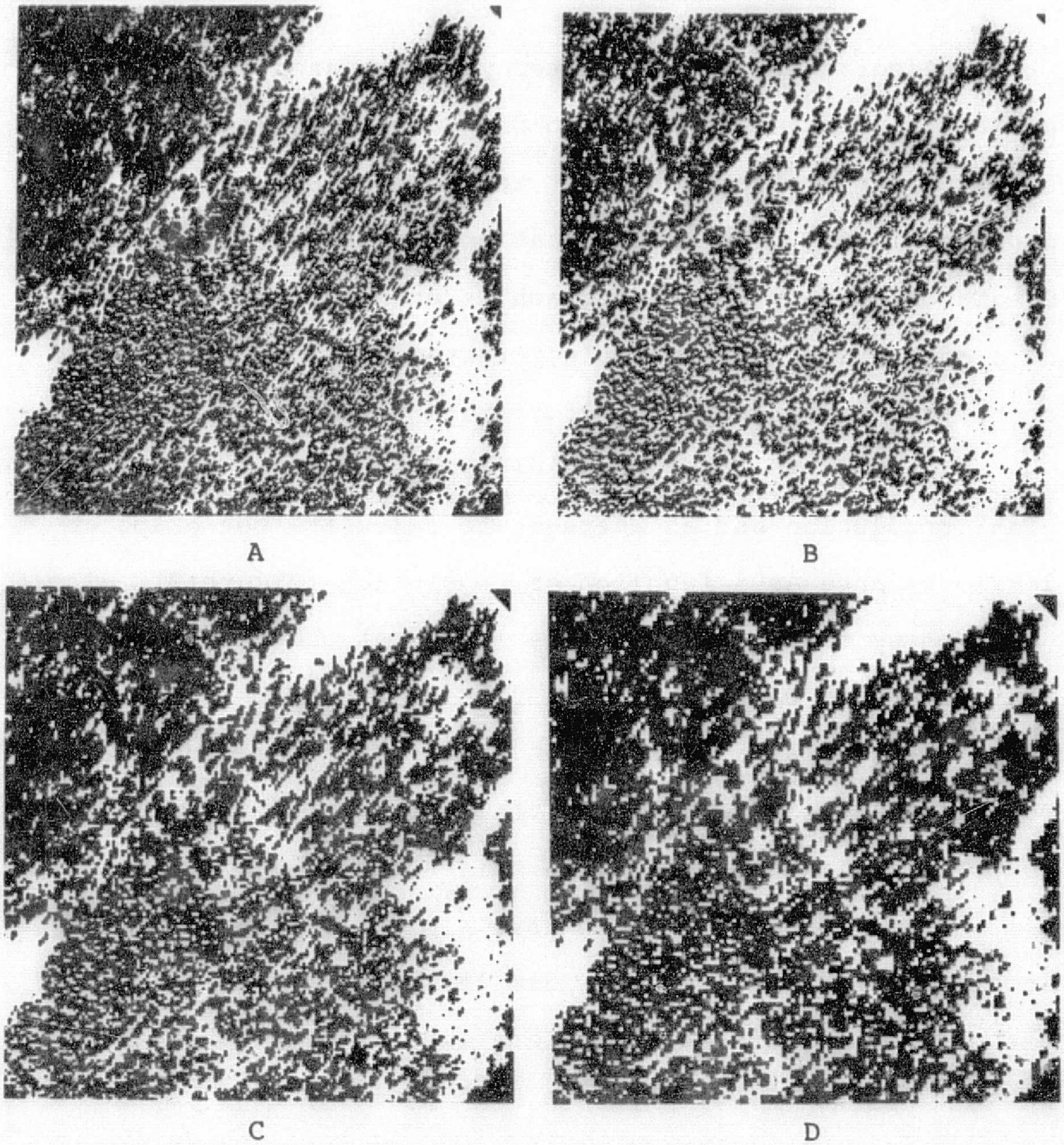


Figure 1. South Dakota forest image shown at resolutions as scanned (Ø.75 m) (A), and as averaged to yield resolutions of 3 m (B), 6 m (C), and 9.0 m (D).

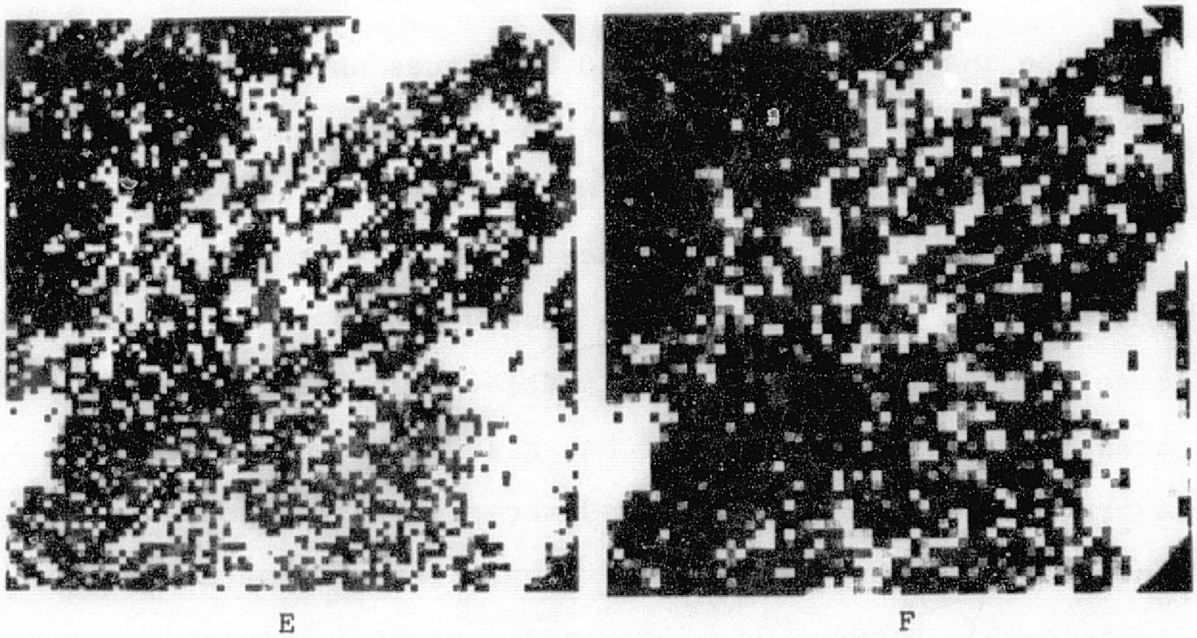


Figure 1, continued. South Dakota forest image, scanned and averaged to resolutions of 12.0 m (E) and 16 m (F).

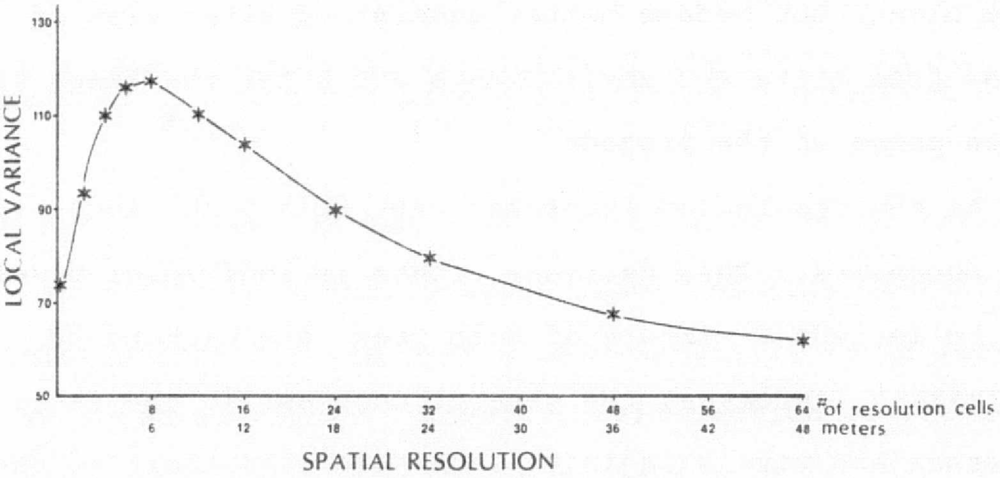


Figure 2. Local variance (average standard deviation within a three-by-three window) as a function of resolution cell size for the South Dakota forest image.

As the size of individual resolution cells increases, the number of pixels comprising an individual tree decreases, and the likelihood that surrounding pixels will be similar decreases (Figure 1B). In this situation, local variance increases. This trend continues until the pattern becomes very mottled and a peak in local variance is observed at 6 m (Figure 1C). While it was originally hypothesized that local variance would peak at the size of the elements in the scene, the observed peak occurs when the resolution cells are somewhat smaller than the trees in the scene. Close examination of the size of trees reveals an average size of between 8 and 9 meters, whereas local variance peaks at about 6 meters. Thus, there is not a simple relationship between local variance, spatial resolution, and the size of elements. An explanation for the location of the peak in local variance could not be determined from this image alone, but became better understood after viewing graphs from different environments and after the image simulation phase of the project.

As the resolution increases past this peak, local variance decreases. This decrease occurs as individual pixels come to include a mixture of both trees and background. As this mixing increases, the general contrast in the image decreases and pixels begin to look more like their neighbors. Local variance thus continues to decrease (Figures 1D-1F).

There is considerable structure in the contour plot of the variogram derived from the South Dakota forest image (Figure 3). The strength of the relationship between a given pixel and its neighbors tends to decrease with dis-

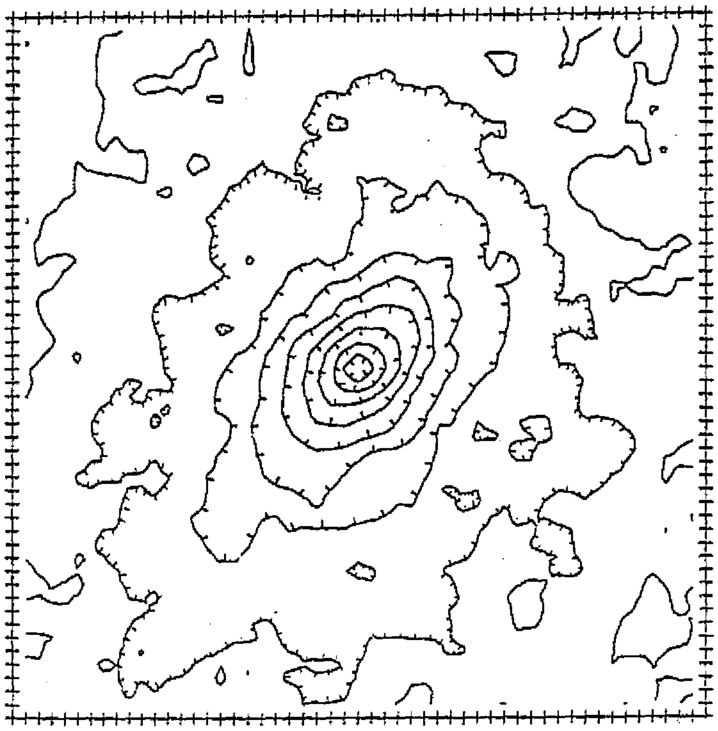


Figure 3. Two-dimensional variogram of the South Dakota forest scene. Units are pixels at original resolution (Ø.75 m).

tance until it reaches the sill, or the level of no interaction, 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 pits. This effect may be due 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. The zone of influence in the variogram seems related to the size of the elements (trees) in the scene, as the width of the area inside the sill approximates twice the size of a tree in the image.

Another notable feature of the variogram is its anisotropy, which is attributable to the shadowing in the image (Figure 1A). The variogram is markedly elongated along a diagonal from the upper right corner to the lower left corner, which corresponds to the orientation of illumination. Since shadows look more like trees than background, the shadow of a tree tends to reduce the variance measured in the direction of the shadow.

Canoga Park Residential Image

The image of a housing development in Canoga Park, California, was obtained through NASA Ames Research Center (Figure 4A). The data were collected by multispectral scanner; the red band was used for this analysis. This scene presents an interesting change from the forest environment in that it is a complex scene, having several kinds of elements. Associated with the complex nature of the scene is a change in the way the scene is organized. In this environment there is not a well-developed background similar to the forest environment. Instead, there are several different kinds of elements that are arranged in a mosaic to comprise the scene. The most obvious elements in the real scene are houses, trees, streets, lawns, and cars. However, close examination of a blowup of the image (Figure

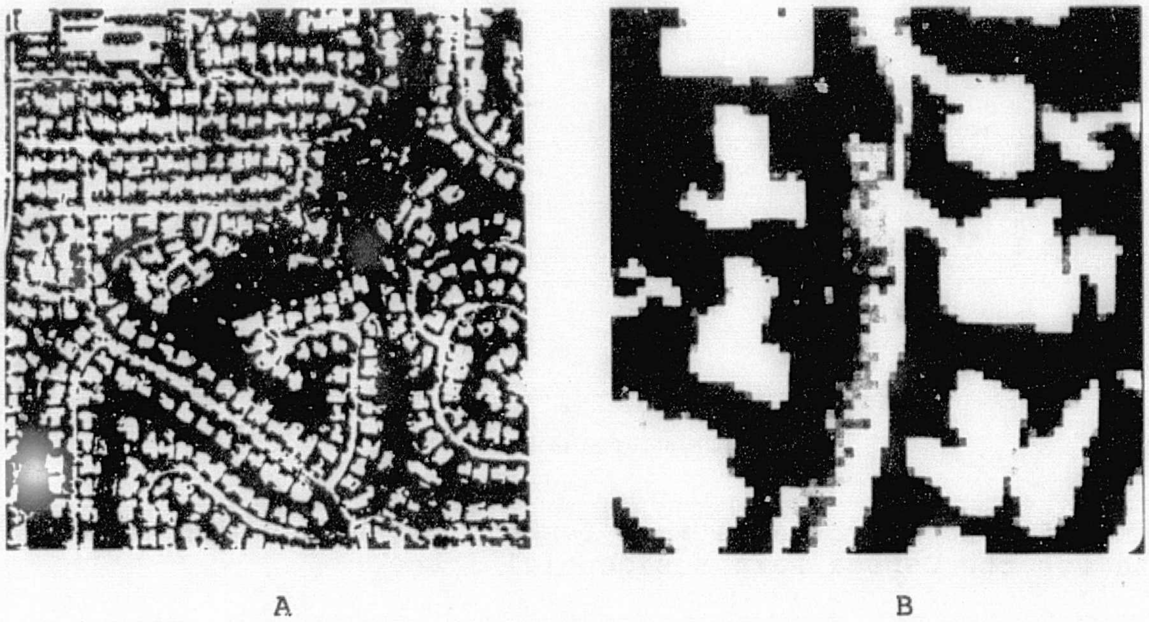


Figure 4. Red band of multispectral scanner image of a housing tract in Canoga Park, CA. (A) Portion of image; (B) enlargement showing detailed structure.

4B) reveals three kinds of elements: houses (actually their roofs), streets, and vegetation or very dark areas. Vegetation covers most of the spaces between the houses and the streets. While it is undoubtably composed of many types of plants with different life-forms, they all appear very dark in the image and cannot be differentiated. Due to their dark appearance, shadows can not be distinguished from the vegetation either, contributing to a description of the scene using three elements.

Figure 5 shows the graph of local variance as a function of resolution for this image. This graph is similar in appearance to the graph for the forest scene in that the local variance is low at the original resolution of the

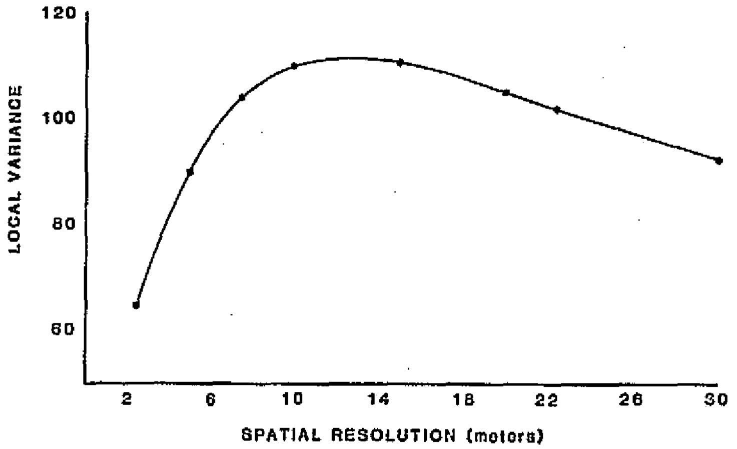


Figure 5. Local variance as a function of resolution cell size for the Canoga Park residential image.

data, rises to a peak, and then decreases. However, the general shape of the curve is different and the resolution where local variance peaks is different. The shape of the curve connecting the points on the graph is broader, not having as sharp or as well-defined a peak. The broad nature of the graph is probably attributable to the complex nature of the scene, with different elements being of different sizes. The broad distribution of sizes results in high local variance over a wider range of spatial resolutions. The peak in local variance occurs at about 13 - 15 m, or at five or six times the original resolution of the imagery. The general size of the elements again is larger than the spatial resolution where local variance peaks. The average size of houses is approximately 14 pixels in the original image, while the streets are approximately 11 pixels wide and the spaces between houses and between houses and the streets averages about 6 - 8 pixels.

The variogram of the Canoga Park image is nearly circular in the zone of influence (Figure 6). This isotropy indicates that there are not any well-defined directional effects in the image. One can see in Figure 4 that the roads run in several directions in the scene. If one or more directions predominated, there could easily be anisotropy in such an image -- reduced variance in the direction the roads are oriented. Also, there could be effects related to the shapes of houses that might be recovered

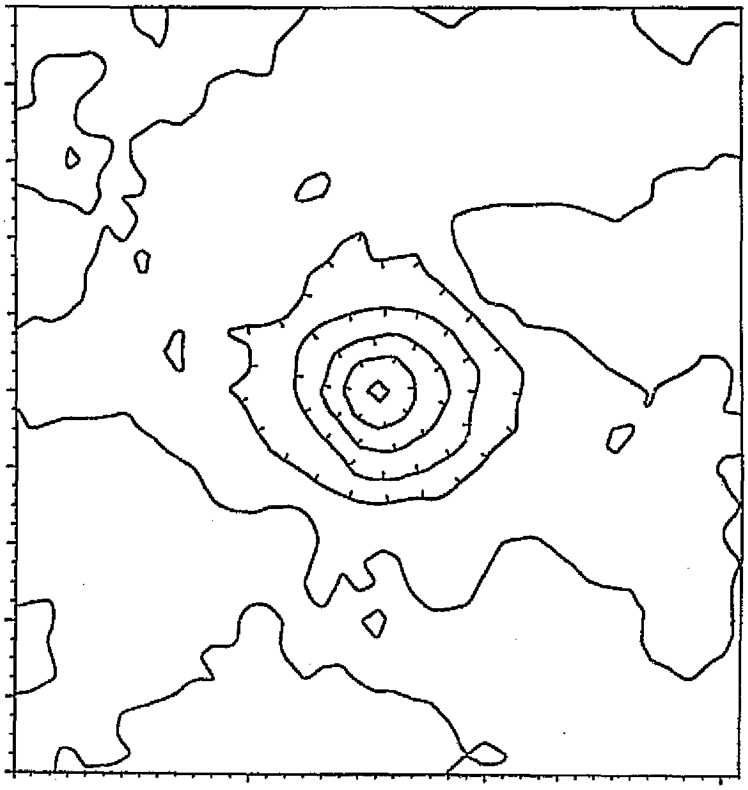


Figure 6. Two-dimensional variogram of Canoga Park residential scene. Units are pixels at original resolution (3Ø m).

through anisotropies in the variogram.

The sill of the variogram probably falls somewhere between the fifth and the sixth contour lines, although an attempt was not made to formally define the sill. The fifth contour line is the last one to hold definite structure and lies about 8 or 9 pixels from the center, indicating the existence of elements in the image that are at least that large. This size acrees reasonably well with the sizes of objects in the scene.

Agricultural Image

The imagery used for the computation of data for the local variance function were scanned at a resolution of approximately Ø.15 m (Figure 7A). Although they are not shown in the figure, the image includes portion of two other fields. Such fine resolution was used because it was hypothesized that an individual agricultural field could be characterized as being composed of elements such as individual plants, crop rows, and a background of soil. In this formulation, if the resolution cells were smaller than individual plants, or the width of a crop row, then the initial local variance would be low. As spatial resolution increased to approximately the size of the elements, an increase in local variance would be expected, similar to the findings in the forest and residential environments. Local

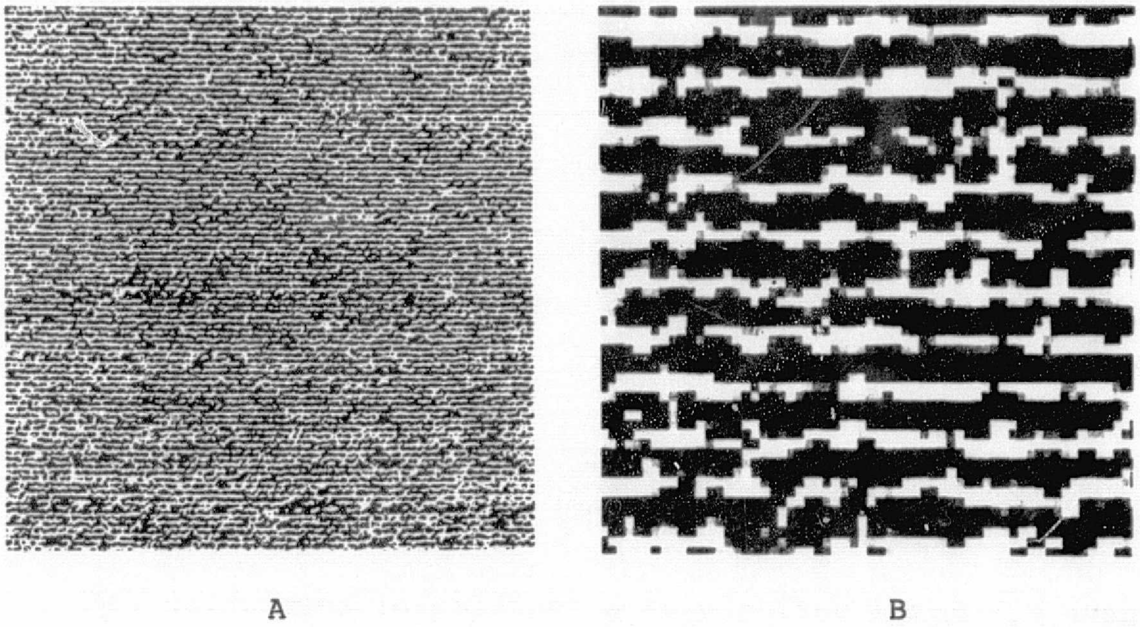


Figure 7. Agricultural image showing row structure. (A) Portion of image; (B) detail.

variance would then be expected to decline as spatial resolution increased.

The observed results (Figure 8) do not follow the hypothesized form because the imagery was not scanned at a resolution fine enough for individual elements (rows, shadows, and furrows) to be characterized by many pixels. Instead, the graph begins with local variance already high. The distance between crop rows is approximately 5 resolution cells at the original resolution of Ø.15 m. In those five pixels are included the well-illuminated portion of the crop row, the shaded side of the crop row, and the soil furrow between the rows. As a result, very few 3 X 3 windows in the image will have low variance. This problem can be seen

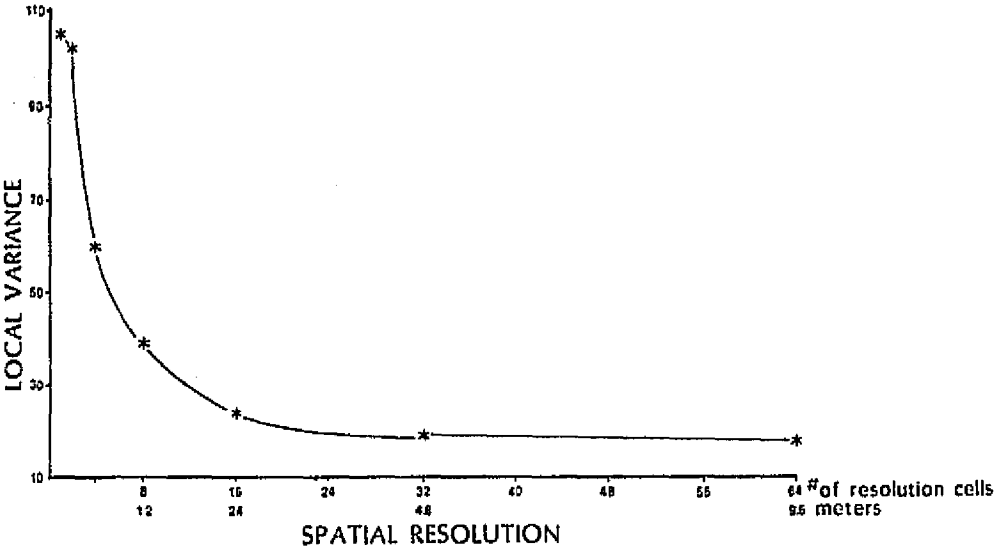


Figure 8. Local variance as a function of resolution cell size for the row-crop image.

in a blowup of a portion of the image shown in Figure 7B. If resolution were considerably finer, variance within both the shaded and well-illuminated portions of a single crop row would be low. A spatial resolution on the order of 5 cm would be required for this effect to be observed. Another factor that may contribute 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 contrasting.

Another noteworthy feature of the local variance graph is the rapid decline past the peak to a very low level. This feature is the result of a scene that becomes very homogeneous once the resolution cells are larger than the crop rows.

Variograms were computed for two different agricultural fields in the image and then the entire image as a whole. These variograms exhibit considerable structure related to the orientation and spacing of the rows. Figure 9 shows the variogram of a portion of the field shown in 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

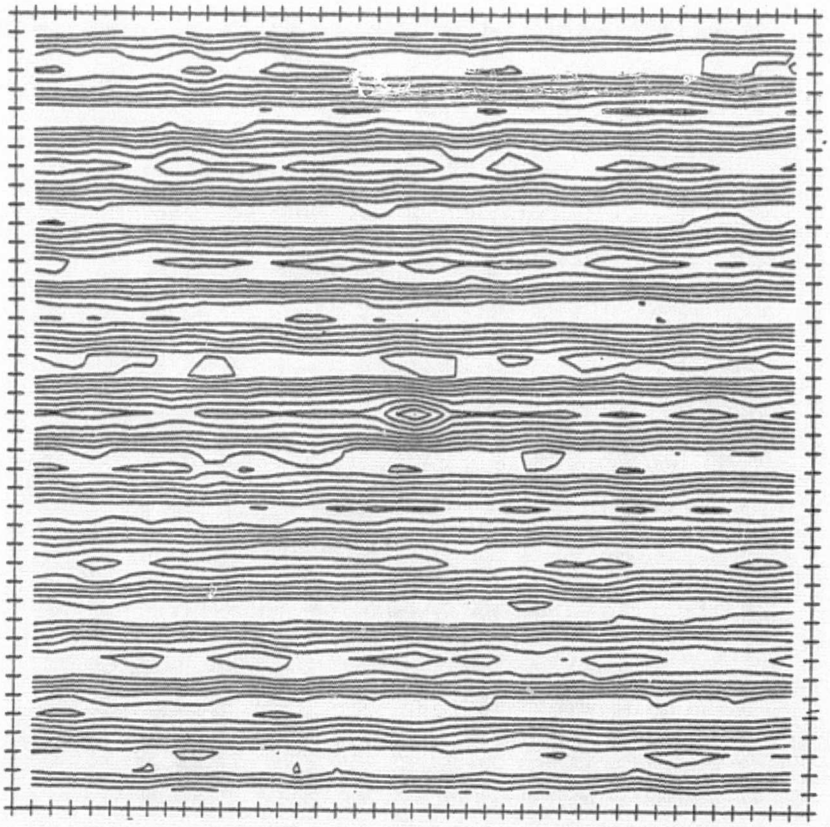


Figure 9. Two-dimensional variogram of a portion of the agricultural field shown in Figure 7. Units are pixels at original resolution cell size (15 cm).

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 half-widths 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. It obviously arises from the repetitive pattern in the image itself produced by the row structure. The

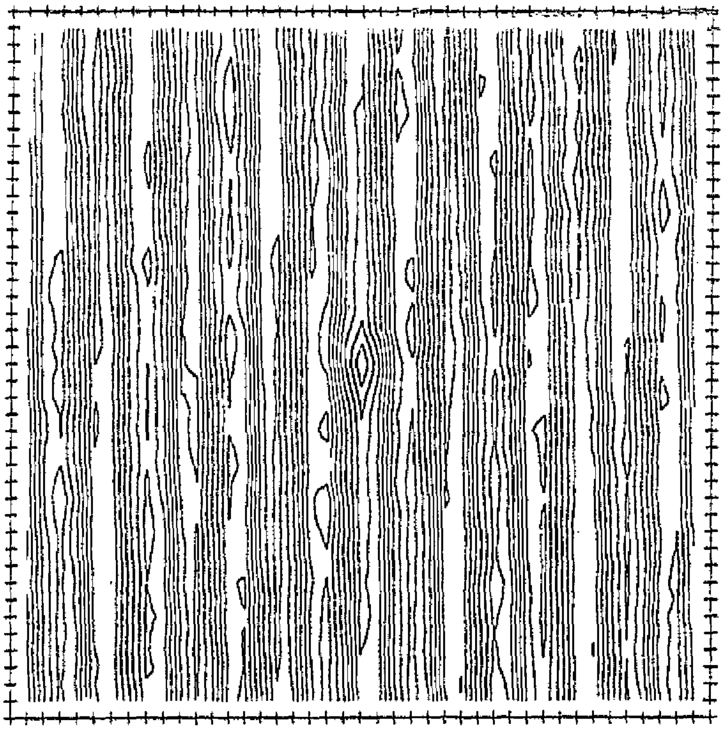


Figure 10. Variogram of a portion of the agricultural scene showing "vertical" row structure.

distance between rows can be determined by counting the number of pixels between the ridges or valleys in the variogram.

For the field in the lower left portion of the image, the variogram (Figure 1Ø) exhibits structure similar to the previous variogram except that the row direction is rotated 9Ø 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 11) is easier to understand after looking at the variograms for the individual fields. The variograms for the entire image can

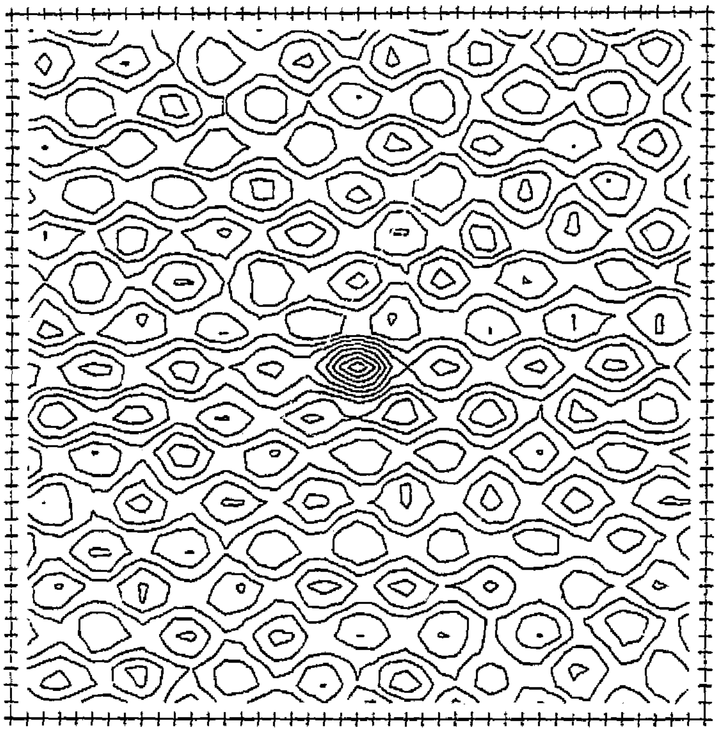


Figure 11. Variogram for entire agricultural image showing combined effects of orthogonal rows.

be thought of as the result of superimposing the variograms from fields with rows oriented perpendicularly.

The variograms for these agricultural fields illustrate the strength of the methods to illustrate underlying structure in images. Figure 7A shows a large portion of the lower left agricultural field used to estimate the variogram in Figure 9, and the linear structure can be easily seen in this picture. However, Figure 7B is a blowup of a portion of this image and illustrates how noisy this linear structure is. The variogram estimated from the image clearly identifies the linear structure despite the large noise component in the image. These results suggest the similarity between the variogram and spectral analysis, which is another method of finding periodicities in data.

Thematic Mapper Agricultural Image

The image used to analyze spatial patterns in an agricultural scene at coarse resolution is a Thematic Mapper image (Band 3) obtained from Johnson Space Center (Figure 12). This image is from the area near the corners of Missouri, Louisiana, Kentucky, and Tennessee, and the subimage selected is west of the Mississippi River. This area is ideal for this project because the scene is composed almost entirely of agricultural fields. In addition, many of the fields are planted in different crops or are at different

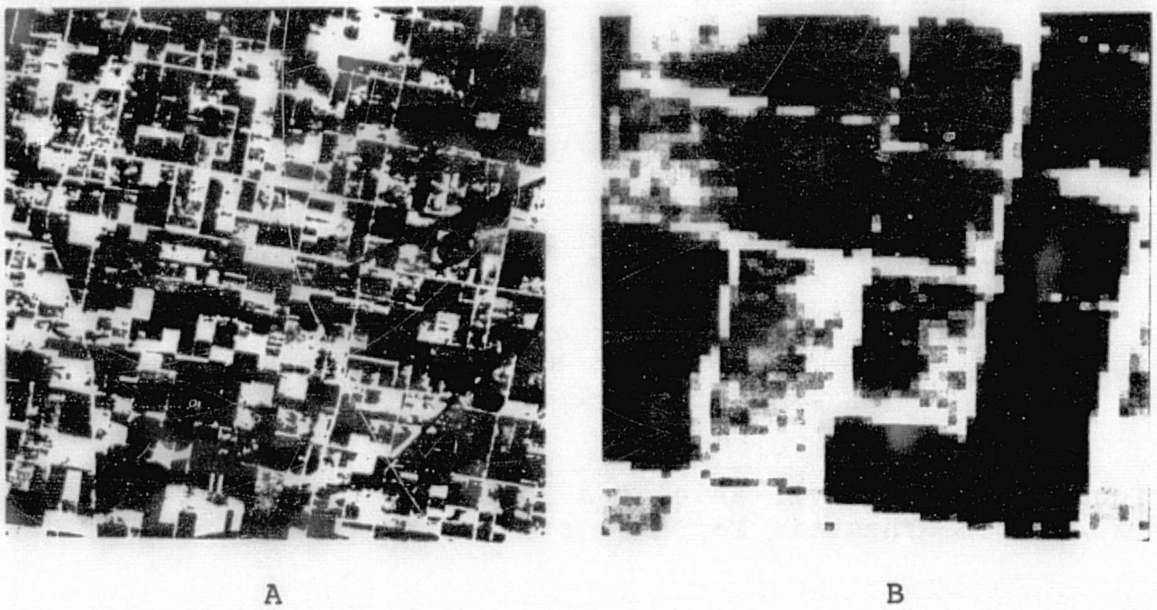


Figure 12. Midwestern agricultural scene imaged by Thematic Mapper (A). (B) Detail.

stages of development, and thus are contrasting in appearance.

The graph of local variance as a function of resolution has a similar form to the fine resolution forest scene, although it covers a different range of spatial resolutions (Figure 13). Local variance starts reasonably low at the original resolution of the data (30 m), but increases to a broad, general peak at about 240 m, and then begins a gradual decline. This shape indicates that the elements in the scene are larger than the resolution cells of the original data. In this scene, there are a variety of field sizes and shapes, but the most common field size is a quarter-section, which is 14 resolution cells on a side at the original resolution of the data (Figure 12B). Thus, the peak occurs

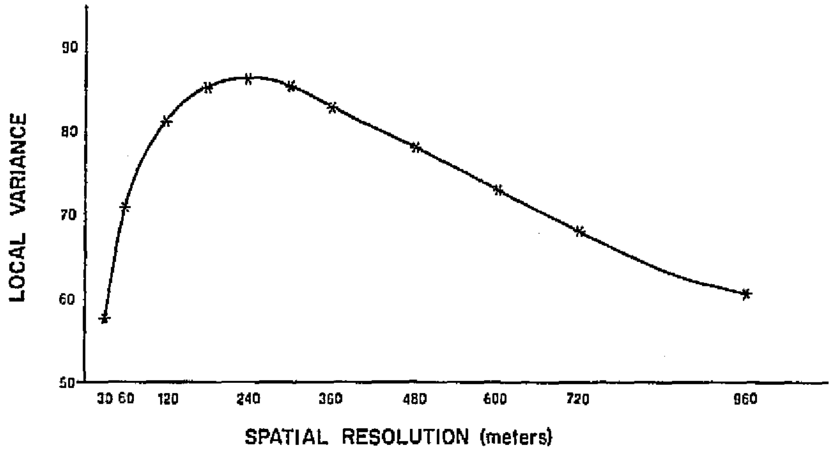


Figure 13. Local variance as a function of resolution cell size for the Thematic Mapper agricultural image.

before the size of the elements in this scene, similar to the findings with the fine resolution forest image.

The variogram derived from this image is shown in Figure 14. The variogram generally decreases as a function of distance, reflecting the homogeneity within fields. The size of the zone of influence is related to the size of the fields in the image, and there is a slight anisotropy in the variogram. This anisotropy is related to the general trend toward rectangular fields in the image that are longer in the north-south direction than the east-west direction. This characteristic can be detected in Figure 12A.

Washington, D. C. Thematic Mapper Image

The area used for the calculation of the local variance/resolution graph and the variogram in this TM image

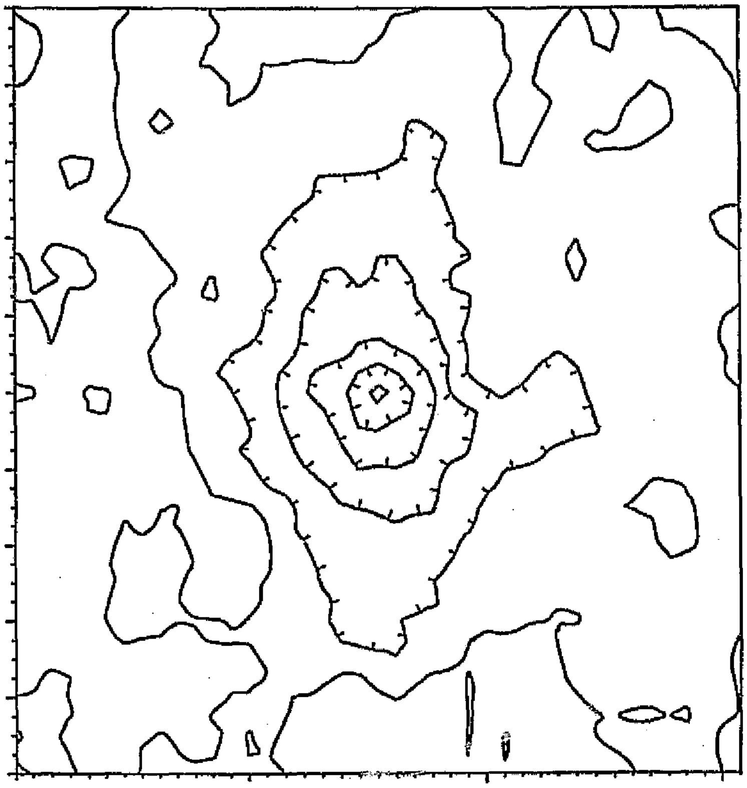


Figure 14. Two-dimensional variogram for Thematic Mapper midwestern agricultural image. Units are pixels at original resolution (30 m).

is taken from the city of Washington (Figure 15). Thus, it is not as simple or as well-defined a scene as the previous examples. The graph of local variance as a function of spatial resolution (Figure 16) does not have the familiar structure of initial low values, a peak, and eventual decline. Instead, there is a general decline in the local variance over the range of spatial resolutions covered by the graph. This indicates that the elements in the scene are generally smaller than the original resolution of the data. There are some multipixel objects in the image, as

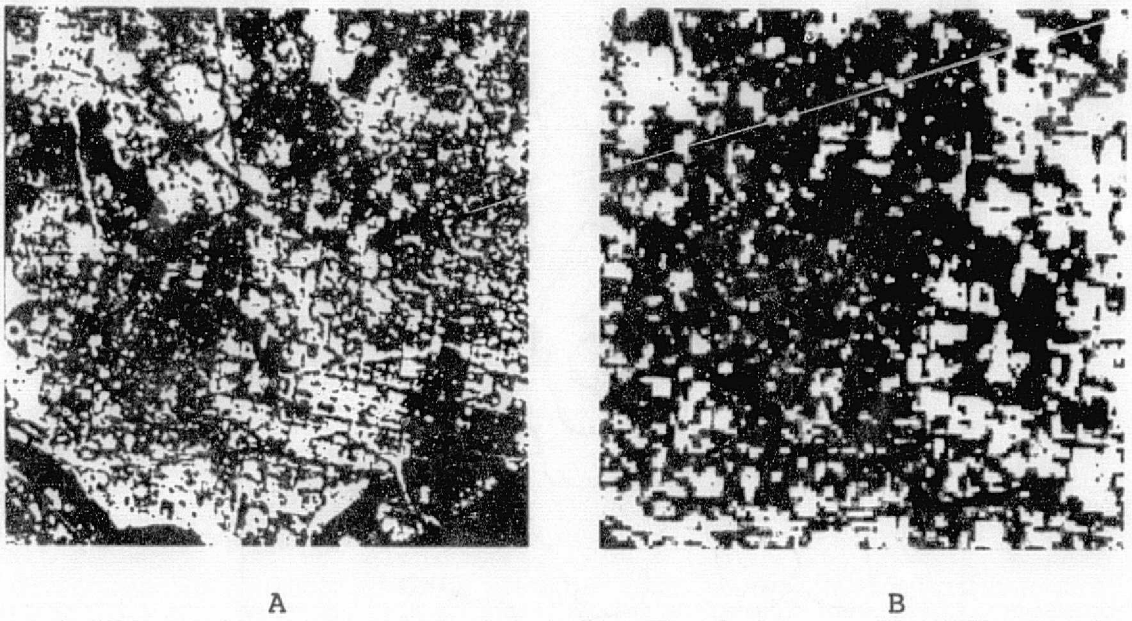


Figure 15. Portion of Thematic Mapper image of Washington, D. C. (A) with detail (B). (Images are reversed left-for-right.)

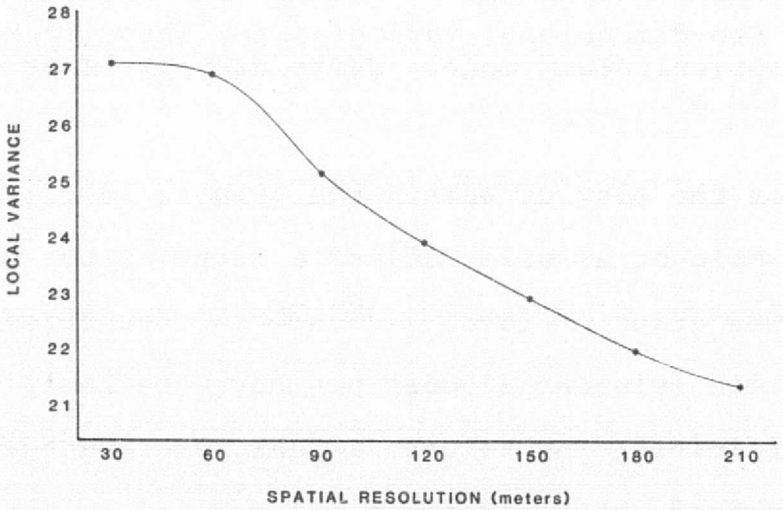


Figure 16. Local variance as a function of spatial resolution for the Thematic Mapper image of Washington, D. C.

can be seen in Figure 15B. These blocks of bright pixels are large buildings and may help explain the flat beginning of the graph in Figure 16. Other than these large buildings, there are few homogeneous regions in the image. In particular, the residential area of southeast Washington in the left part of Figure 15 has a mottled and random appearance.

The variogram derived from the Washington D.C. image does not exhibit any structure that is particularly interesting (Figure 17). The small size of the zone of influence indicates the relatively small size of the few

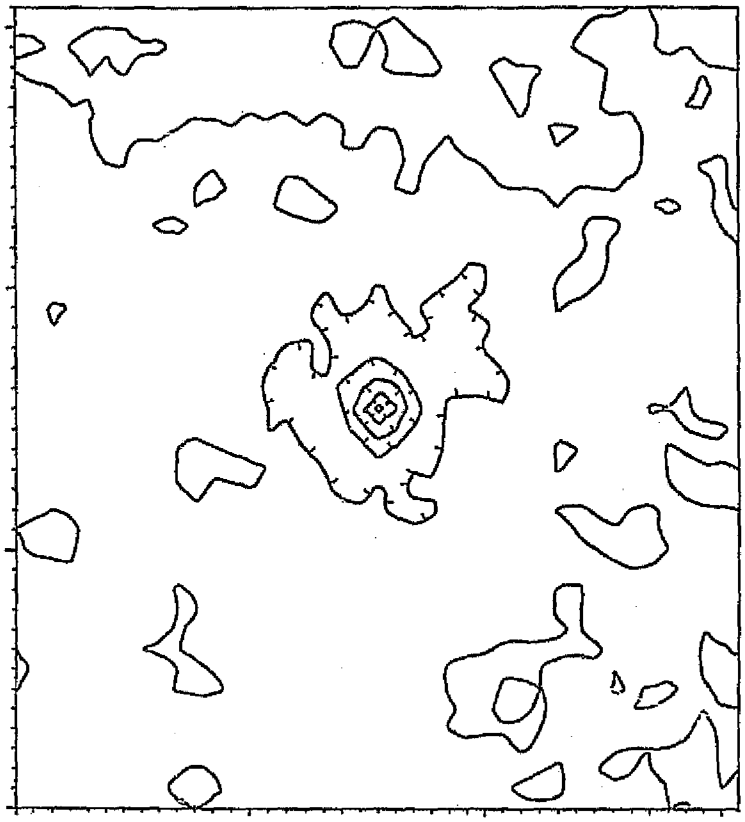


Figure 17. Two-dimensional variogram of Thematic Mapper image of Washington, D. C. Units are pixels at original resolution (30 m).

elements that are larger than the resolution cells in this scene. There is not an obvious explanation for the complex shape defined by the fifth contour line. There are not any features that have a similar shape or whose combined orientations would produce this pattern. These variations are most likely due to random effects. However, it is possible that they represent subtle characteristics of the image.

Klamath Forest Thematic Mapper Simulator Image

As an example of a forested scene at 3Ø-meter resolution, a Thematic Mapper Simulator image of a portion of the Klamath National Forest was obtained from Ames Research Center. (Unfortunately, prints of this digital image were not available at the time of preparation of this manuscript.) The results of the two methods of measuring spatial pattern for this image are similar to the results for the Washington D.C. image. The graph of local variance as a function of spatial resolution shows a marked decline as resolution increases (Figure 18). The results indicate that there are not any spatially homogeneous elements in the image that are composed of many trees. Initially it was expected that stands of trees, which can be identified by human interpreters, might cause a second peak in local variance at a resolution related to the size of the stands. However, such a peak did not occur for this image.

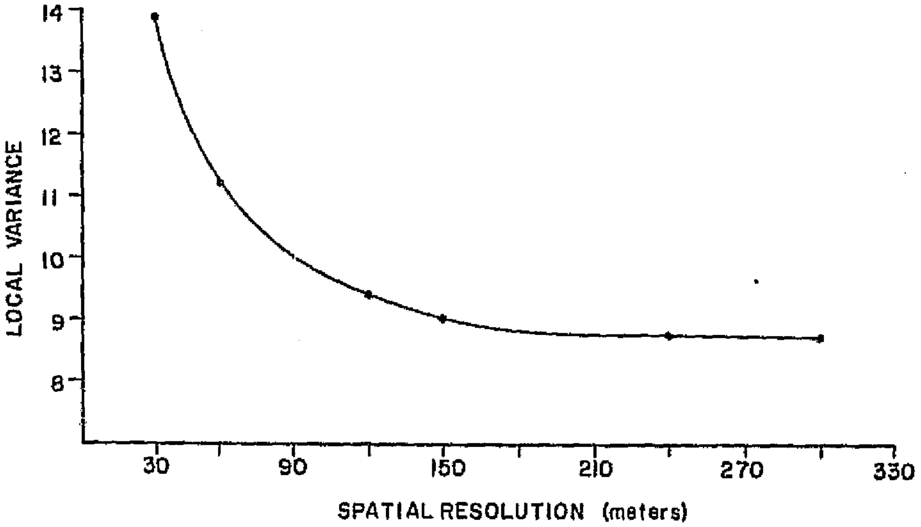


Figure 18. Local variance as a function of resolution cell size for a Thematic Mapper simulator image of a portion of the Klamath National Forest.

The drafting of the variogram for this image was not completed in time for this publication. However, it looks very similar to the results for the Washington D.C. image, exhibiting a small zone of influence. In addition, there are no well-defined anisotropies that reveal any directional orientations to the elements in the scene. The variogram thus confirms the conclusion that there are no large, spatially homogeneous elements obvious within the image.

Image Simulation

The results presented in the last section are interpreted in an intuitive manner. The empirical nature of remotely sensed images makes it difficult to control scene parameters in a way that allows experimentation to verify or help clarify the meaning of the results. Thus, it became important to develop a method of acquiring images from scene with known characteristics. One way to approach this problem is to simulate ground scenes and model their reflectance characteristics to produce an image. This approach has several advantages. First, it allows complete control, and thus knowledge, of the ground scene. Second, simulation allows examination of simple scenes, which is important for developing a solid foundation in such an exploratory and empirical study.

Forest scenes were selected to serve as the basis of the image simulation phase of the project. Forests were selected for several reasons. Past remote sensing research experience in forestry directly contributed to the development of the ideas for this project. Also, forests present a simple scene model that can be simulated relatively easily. And, as part of another line of research by Strahler and Li [4], a simulation program existed that could be modified for the purposes of this project.

The image simulation procedure is based on Monte Carlo methods and uses a two-resolution concept. The first level of resolution is the size of the units in the ground scene at which elements are differentiated. The second is the level of aggregation used to simulate the image. For the simulated image used in this project, these two resolutions were the same, which means that each resolution cell was assigned to a single type of element (crown, shadowed crown, understory, or shadowed understory). This approach produced images similar to the fine-resolution imagery that was scanned from aerial photography.

The ground scene is modeled as trees on a plane. The trees are conical in shape with a known apex angle and a lognormal height distribution. The apex angle used is based on the results of field measurements. The use of a lognormal height distribution was selected on the basis of the results presented in the ecological literature. Field measurements of tree heights have confirmed the lognormal shape of the distribution, but have illustrated the variability of the means and variances characterizing the distributions.

The trees (or cones) are distributed randomly on the surface with one exception -- the center of a new tree is not located within the cone of a previously located tree. This modification to the random model was based on the expectation that competition between trees would result in the likelihood of finding trees very close together being lower than the random model would produce [5]. Subsequent field measurements have not supported this hypothesis, and have indicated that the simple random model is a good approximation to spacing in conifer stands.

Following the random location of trees and the lognormal assignment of their heights, an elevation map is created with the heights of the surface above the base level representing the height of the forest canopy. By specifying a direction of illumination and a solar zenith angle, shadows are produced. The result is the definition of four kinds of surfaces in the scene: trees, background, shadowed trees, and shadowed background. From these, a digital image can be synthesized that resembles an image drawn from a real scene (Figure 19).

The results of the graph of local variance as a function of resolution for a simulated forest image are shown in Figure 20. The size of the image that was simulated limits the number of times that spatial resolution can be degraded. Thus, it was not possible to evaluate local variance for the full range of resolutions used in the scanned forest image. However, the shape of the curve is very similar to the results for the scanned forest image (Figure 2). There is a prominent peak at 6 m and then a decline in local variance.

The diameter of the tree crowns in the simulated image has a mean of 7 m and a very low variance (approximately 0.5 m). However, because the shadows look more like the trees than the background, the effect of shadows is to make the trees appear elongated in the direction opposite the illumination source. If shadows were considered part of a single dark element with trees, then their size along one axis would be 11 m. Thus, the peak in local variance occurs at a size somewhat lower that that of the elements in the scene,

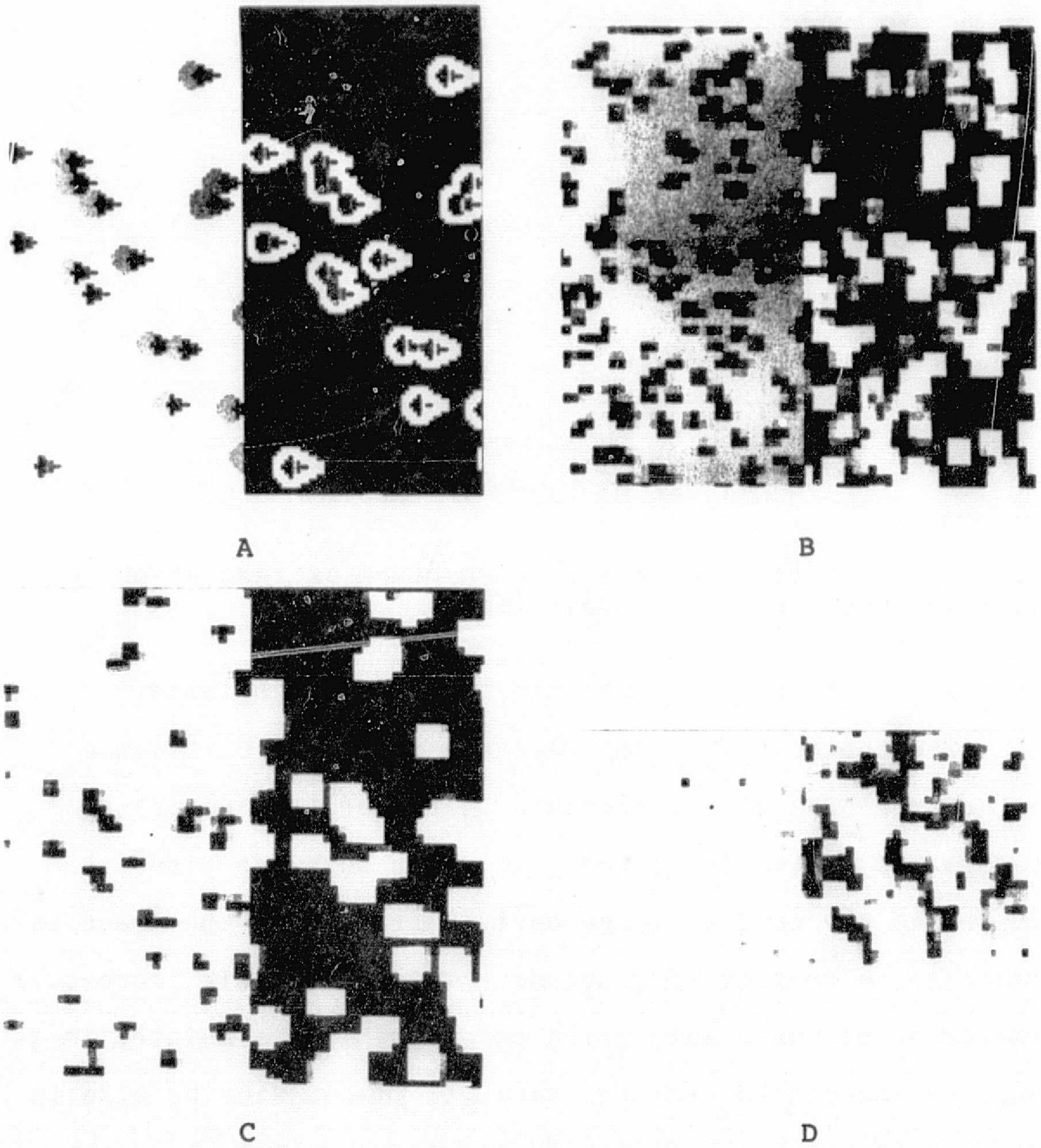


Figure 19. Portions of simulated forest images. On left, simulated image. On right, texture image derived from it. (A) Image as simulated at 1-meter resolution. Other photos show image degraded to 3-meter cells (B); 6-meter cells (C); and 9-meter cells (D).

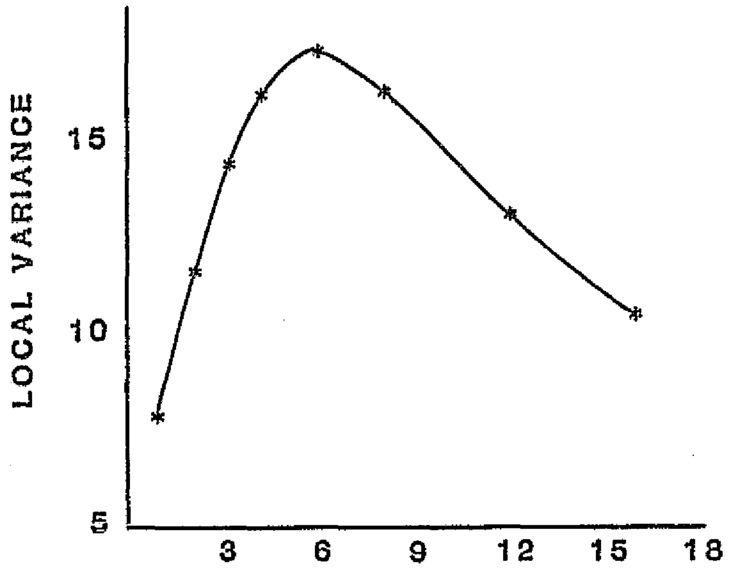


Figure 2Ø. Local variance as a function of resolution cell size for the simulated forest image.

similar to the results obtained for the forest image.

The reason that local variance peaks for this image before the size of the elements is reached can be better understood by examining the changes in both the simulated image and the texture image derived from it as the image is degraded to coarser resolutions. To display this process, a series of pictures with portions of both the simulated image and its associated texture image are placed side by side in Figures 19 A-D. The first picture (Figure 19A) shows the simulated image at its original resolution. In the texture image, high local variance occurs primarily around the perimeter of the tree and its shadow, behaving like an edge detector. The area inside the perimeter still has relatively low local variances and the area between trees is black, as the background has the same value in all locations.

Figure 19B shows the results after the image has been degraded to 3 m. At this resolution the trees can not be distinguished from their shadows and begin to appear out of focus. The dark areas inside trees in the texture image have disappeared because the size of the trees in terms of number of pixels has decreased. Similarly, the distances between trees shrinks and local variance becomes increasingly influenced by effects from neighboring trees. Comparison with the first texture image (Figure 19A) shows a larger area covered by bright values, indicating high local variance.

The resolution of peak local variance (6 m) is shown in Figure 19C. At this resolution, trees have become very small, and a large area of the texture image is bright, indicating high local variance. An interesting characteristic, which becomes very important, is that there are a considerable number of pixels with intermediate values in the texture image. In the previous texture images, pixels were either near edges and very bright, or in homogeneous areas and very dark. These intermediate texture values are the result of the effect of the degradation of resolution on the appearance of trees.

At a resolution of 9 m, local variance has begun to decline (Figure 19D). The texture image has begun to look like a continuous tone image, quite different from the edge detector in Figure 19A. While a greater proportion of the texture image has values other than black, the mean value of the image is lower. This observation is the key to understanding the reason that local variance peaks before the size of the elements. As the imagery is degraded, the model for the appearance of a tree is different than originally expected. As the resolution cells become larger, trees tend to look more and more out of focus, with many pixels being composed of a mixture of both dark tree or shadow and light background. Thus, as the size of a tree is approached, instead of having alternating light and dark pixels for tree or background, there are several intermediate tone pixels. The reflectance of any given tree is spread through many pixels. This effect can be seen in Figure 19C and 19D. The effect of numerous intermediate-tone pixels on the texture image is the production of only a few high texture values. The contrast between pixels in the image is not large enough for high texture values.

When viewed from this perspective, the result that local variance peaks before the size of the elements makes sense. The sampling theorem states that a resolution cell half the size of the element would be necessary to assure brightly contrasting pixels in the image. This perspective, combined with the increasing area covered by texture values that are not black as resolution increases, produces a peak in local variance at about 2/3 to 3/4 the size of trees in both the simulated and scanned forest images.

The variogram for the simulated image is shown in Figure 21 and serves to confirm the interpretation of the previous variograms. The sill occurs at about the width of a tree from the center. Also, the anisotropic shape is related to the orientation of illumination, similar to the results for the fine resolution forest image. It is interesting that the var. gram of the simulated image has peaks and pits outside the zone of influence. The significance of these features is unknown, but it is possible that they are indicative of periodicity induced by the constraint placed on the location of trees. Substantiation of such an

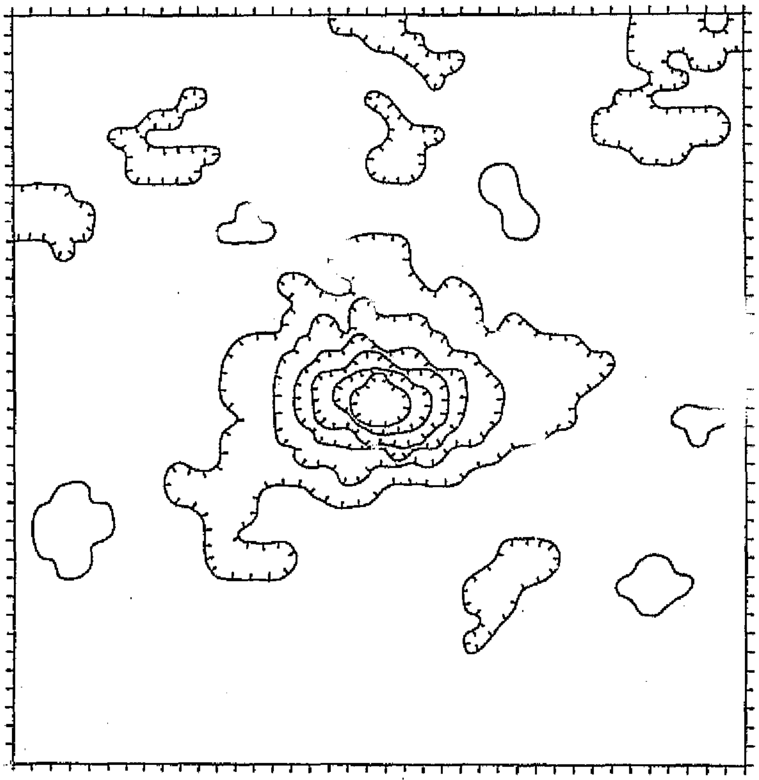


Figure 21. Two-dimensional variogram of simulated forest scene. Units are pixels at highest resolution (1 m).

effect will require further testing.

CONCLUSION

This study shows that spatial variance in digital images depends on the nature of objects within the scene itself, including their size, shape, and spacing. When the resolution cell size of an image is sufficiently smaller than the objects that dominate the scene, overall image texture remains low, the objects can be resolved, and the two-dimensional variogram will easily reveal their shape. As resolution cell size increases, local variance will peak at a resolution cell size near two-thirds of the size of the object. If the resolution is too coarse to reveal individual objects, local variance will never peak as the image is degraded, and the variogram will show little structure.

Figure 22 presents all the local variance graphs derived for real images shown on a single graph. Note that the abscissa is logarithmic; note also that the heights on the y-axis, which measures local variance, are dependent on the contrast of each image and are thus not directly comparable. However, the figure clearly identifies the sensor-scene combinations for which classification and clustering are appropriate (where resolution cell size is significanlty smaller than the objects in the scene) as opposed to mixture modeling (where resolution cell size is significantly larger

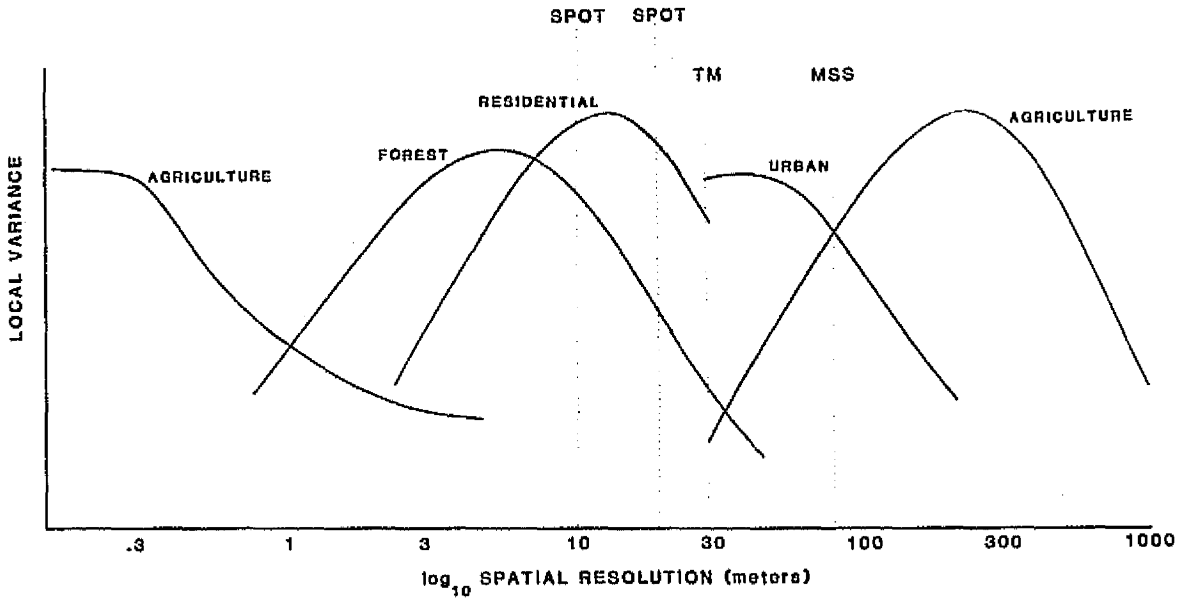


Figure 22. Combined local variance graphs for the scenes studied in this paper. Note that the height of the peaks is arbitrary, since it depends on relative image brightness.

than the objects in the scene). Thus, it will be easy to identify fields with TM data, but urban scenes will require a different approach than classification. SPOT panchromatic data, at 1Ø-meter resolution, will delineate urban objects and forest trees, but will still not reveal the periodic structure of crop rows within agricultural fields.

Future work, anticipated for the third year of this project, will involve the explicit formulation of variograms for scenes composed of simple objects in regular and random arrangements. We will also formulate the exact relationship bwtween parametric variograms and variograms of images in which spatial averaging within picture elements occurs. From this formulation, we should be able to link the variogram directly with the graph of local variance as a function of resolution cell size. Clarifying these relationships should allow us to come to a better understanding of how variograms and other spatial statistics may be used in remote sensing for better scene modeling, image enhancement, and image understanding.

REFERENCES

[1] Strahler, A. H. and C. E. Woodcock (1983). Relating Spatial Patterns in Image Data to Scene Characteristics. Proc. NASA Conf. on Mathematical Pattern Recognition and Image Analysis, Houston, TX, June 1982, pp. 5Ø7-541.

[2] Haralick, R. M. (1979). Statistical and Structural Approaches to Image Texture. Proc. I. E. E. E. 67(5):768-8Ø4.

[3] 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. John Wiley & Sons, New York, pp. 38-53.

[4] Strahler, A. H. and X. Li (1981). An Invertible Coniferous Forest Canopy Reflectance Model. Proc. Fifteenth Int. Symp. on Remote Sensing of Environment, Environmental Research Institute of Michigan, Ann Arbor, MI, pp. 1237-1242.

[5] Franklin, J. (1983). _An Empirical Study of the Spatial Pattern of Coniferous Trees_. M. A. Thesis, University of California, Santa Barbara.

Image-to-Image Correspondence: Linear-Structure Matching

Grahame B. Smith and Helen C. Wolf

Artificial Intelligence Center, SRI International
Menlo Park, California 94025

Abstract

We examine the task of matching images of a scene when they are taken from very different vantage points, when there is considerable scale change, and when the image orientations are unknown. We use the linear structures in the scene as the basis of our correspondence procedure. This paper considers the problem of describing the linear structures in a manner that is invariant relative to the variations that can occur among images, and discusses a method of finding the best description of the linear structures.

1. Introduction

When the human visual system is presented with two views of a single scene, it determines the relative viewing positions of the two images and brings the latter into correspondence. That is, the relationship of each image to the scene is understood so that both images can be used as information sources for further processing. This human ability functions well over a wide range of viewing positions and conditions. It is this ability to place two very different views of a single scene into correspondence that we address in this paper.

We should draw a distinction between two forms of the image correspondence task. Traditionally, image registration has been a task undertaken by photogrammetrists. One application involves registering an image to a map so that new information, present in the image, may be transferred to the map. Another is the registration of the two images of a stereo pair so that disparity information can be extracted. In each of these tasks the two images, (or, in the first instance, the image and the map), are similar in terms of both their viewing position and their scale. The techniques used for registering the two images are point-based. A feature point in one image is matched to the same feature point in the other image. In automated systems this is achieved by selecting a small window about the feature in one image and then correlating this window with one in the second image. If there is little distortion or occlusion, this technique performs well; it has become the basis of current automated image-registration systems.

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.

The other form of the image correspondence task seeks to find the relationship among views that differ widely in vantage point, scale, etc. We will refer to this as the correspondence task, and use registration as the name for the form of the task outlined above. In correspondence tasks there is significant distortion between the images, the scale may differ and may not even be constant across a single image, as is the case in oblique aerial imagery, occlusion is common, and the response of the various sensors to a single feature differs greatly. Feature point matching, as used in image registration, is prone to error. However, feature point matching is not the only means of placing images into correspondence. It appears that the human visual system makes use of nonpoint features, such as linear structures and extended landmarks. The aspects of our investigation reported here utilize the linear structures of the images as the prime elements for achieving correspondence.

In classifying the methods that could be employed to find linear structures in images, we draw a distinction between techniques that use semantic information and those that do not. If, for example, we apply a road operator to locate some of the linear structures in an image, that operator has had built into it semantic knowledge about the appearance of roads. We could proceed in this manner and build comparable operators for all the scene objects that manifest themselves as linear structures in images. Alternatively, we could seek to find the linear structures in an image without "identifying" their nature. In this case, we identify the image behaviour interpreted by us as a linear structure without knowledge of the world objects that gave rise to that structure. We choose this latter course because we wish to establish the correspondence among images without first having to identify the scene objects.

The correspondence task is carried out in three stages: we must find the linear structures, we must build their descriptions and, finally, we must match these descriptions. The details of the first stage is reported in Fischler and Wolf [1]. In this paper we explain how those procedures are employed in the correspondence task. We present a detailed account of our implementation of the second stage – namely, building structure descriptions – along with an outline showing how these descriptions are to be used in the final matching stage.

2. Finding the Linear Structures

Descriptions of the semantically free procedures we use to find linear structures in images can be found in Fischler and Wolf[1]. In essence, these procedures first find those pixels whose intensity levels are local maximums and minimums, then cluster such pixels and identify the minimal spanning tree for each cluster. The long paths in each of the spanning trees are found, whereupon these form the basis for the linear structure reported by the procedures. The results of applying these procedures are shown in Figures 1-4. Figure 1 is a natural-color oblique view of the Eel river in northern California; Figure 2 is a vertical infrared view of the same scene. Each was scanned through red, green, and blue filters; the results of the procedures for finding linear structures in each of these separation images are shown in Figures 3(a),3(c),3(e) and 4(a),4(c),4(e). In addition, the red, green, and blue separation images were combined into images of hue, saturation, and intensity; these were also processed to find the linear structures contained in them. The results are shown in Figures 3(b),3(d),3(f) and 4(b),4(d),4(f).

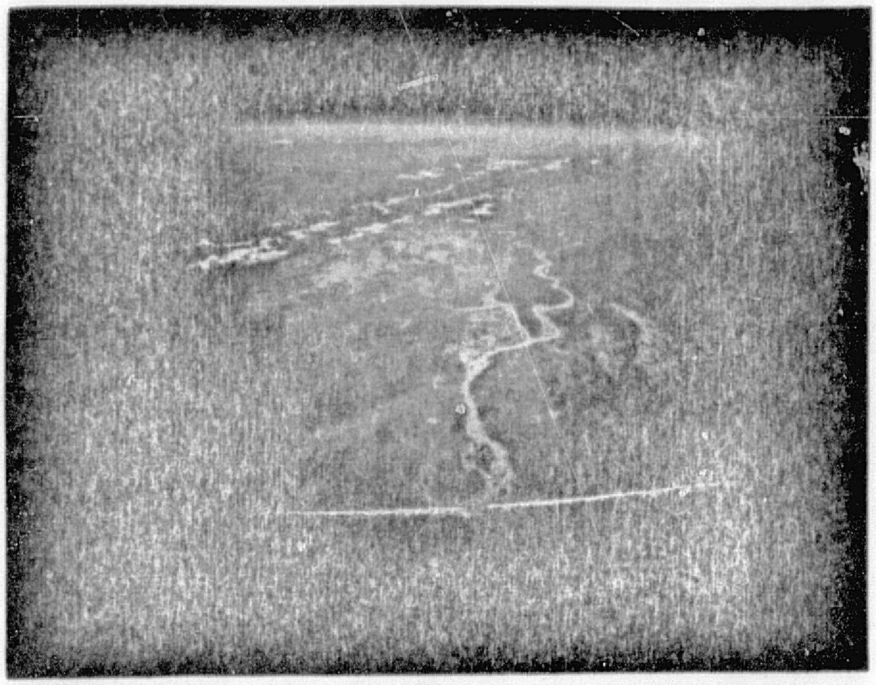


Figure 1. Oblique Natural-Color Image of the Eel River

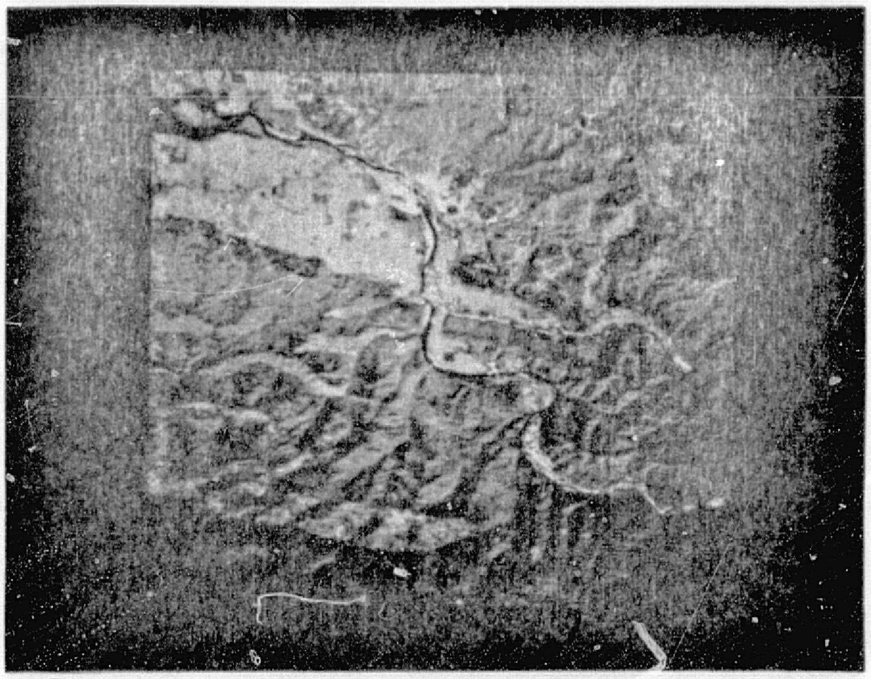


Figure 2. Vertical Infrared Image of the Eel River

These separation images differ appreciably in their linear structure. Certainly no one separation image can be selected as providing a complete delineation of the river. The philosophy we adopt is to view the original image from as many

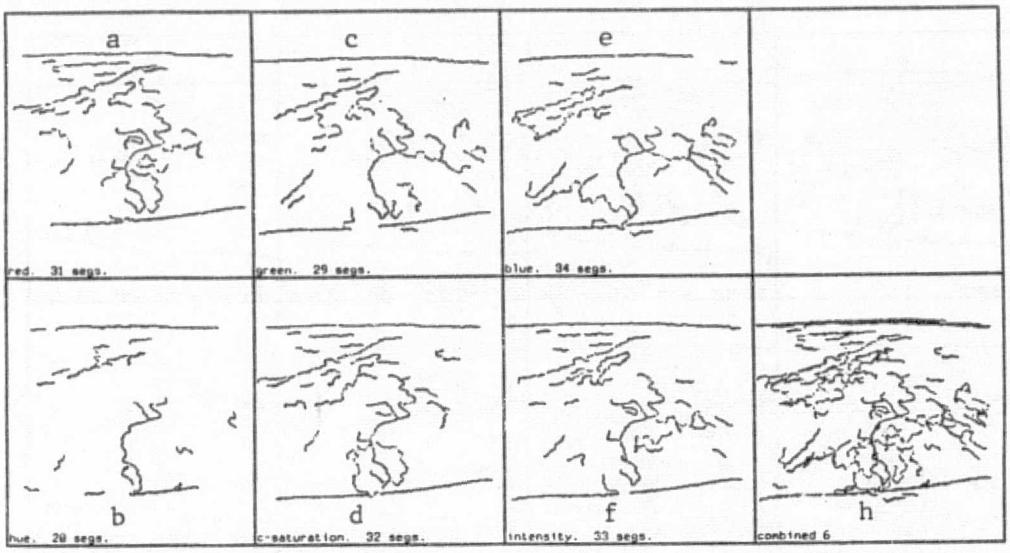


Figure 3. Linear Structure in the Oblique Image

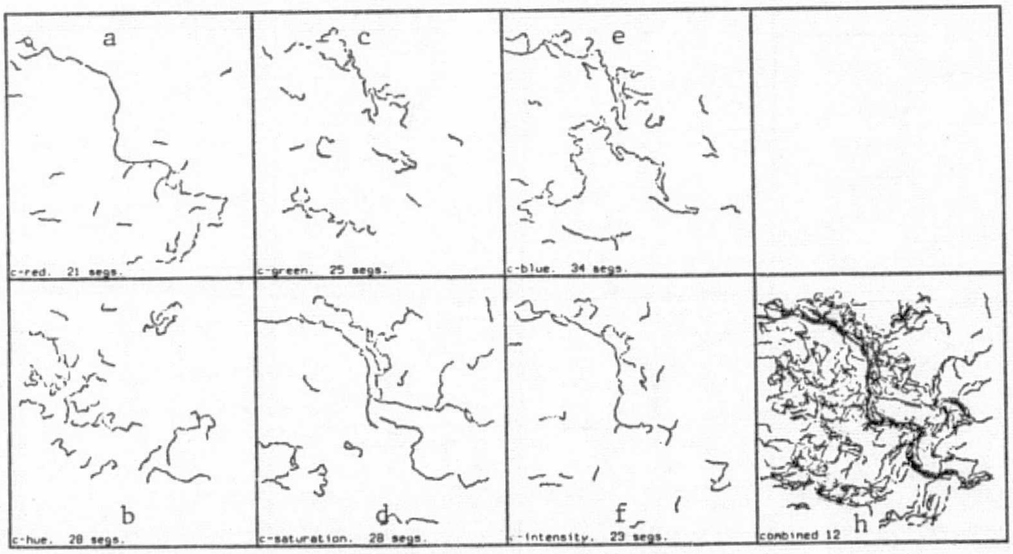


Figure 4. Linear Structure in the Vertical Image

perspectives as possible, obtaining the linear structures as seen from each of these. That is, we look for structures in hue, in the green spectral band, and so on. Of course, the hue image is derived from the red, green, and blue images, and contains only redundant information, but this presentation of the information may show structure that was masked in other presentations. In this sense, the additional

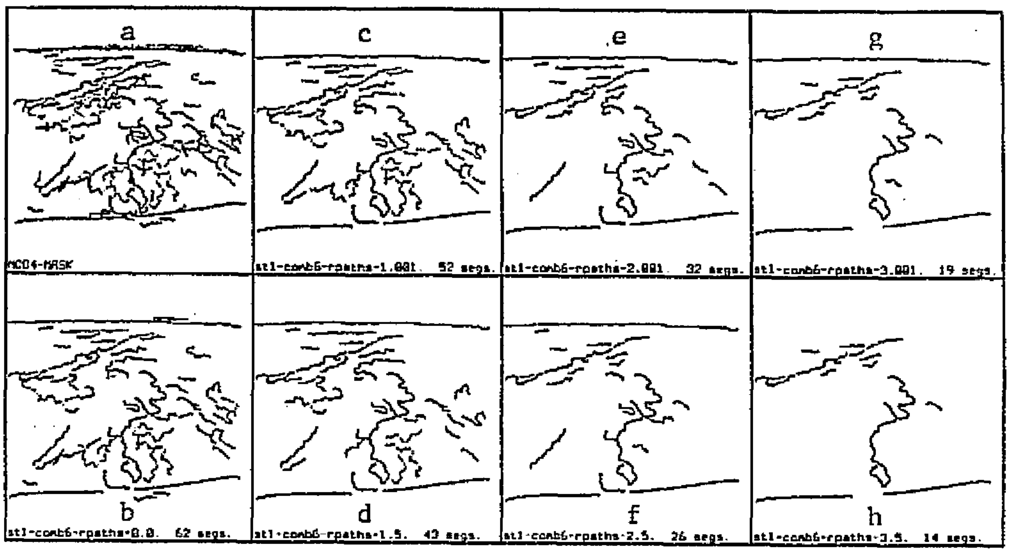

a
c
e
g
MC04-MASK
stl-comb6-rpaths-1.001. 52 segs.
stl-comb6-rpaths-2.001. 32 segs.
stl-comb6-rpaths-3.001. 19 segs.
b
d
f
h
stl-comb6-rpaths-0.0. 62 segs.
stl-comb6-rpaths-1.5. 43 segs.
stl-comb6-rpaths-2.5. 26 segs.
stl-comb6-rpaths-3.5. 14 segs.

Figure 5. Linear Structure in the Composite Oblique Image

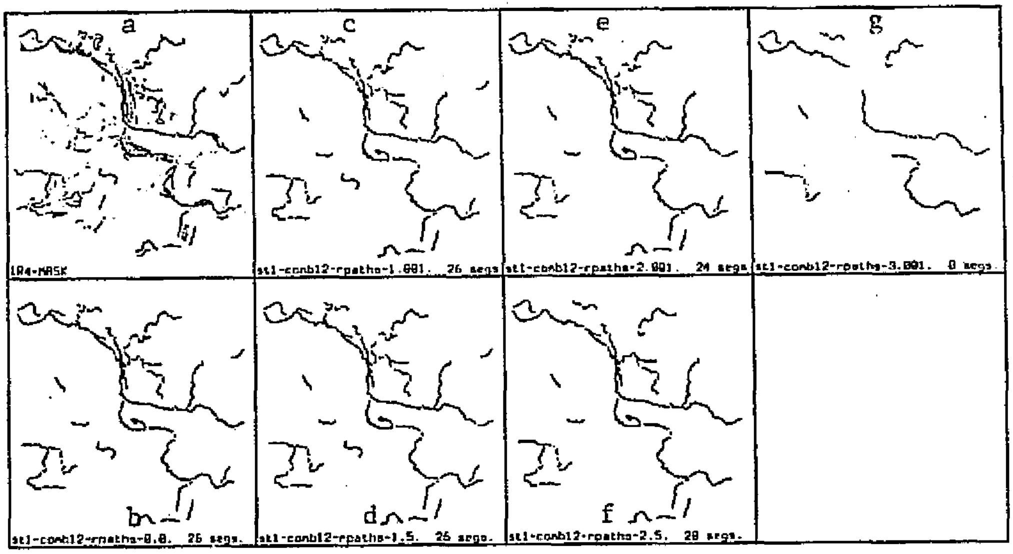

a
c
e
g
1R4-MASK
stl-comb12-rpaths-1.001. 26 segs
stl-comb12-rpaths-2.001. 24 segs
stl-comb12-rpaths-3.001. 0 segs.
b
d
f
stl-comb12-rpaths-0.0. 26 segs.
stl-comb12-rpaths-1.5. 26 segs.
stl-comb12-rpaths-2.5. 28 segs.

Figure 6. Linear Structure in the Composite Vertical Image

perspectives provide new information on which the linear-structure finders can act. The results of combining the linear structures extracted in all the various perspectives are shown in Figures 3(h) and 4(h). Clearly, some of this structure comes from shading effects rather than from physical structure in the scene. We need to separate the real physical structure from all else.

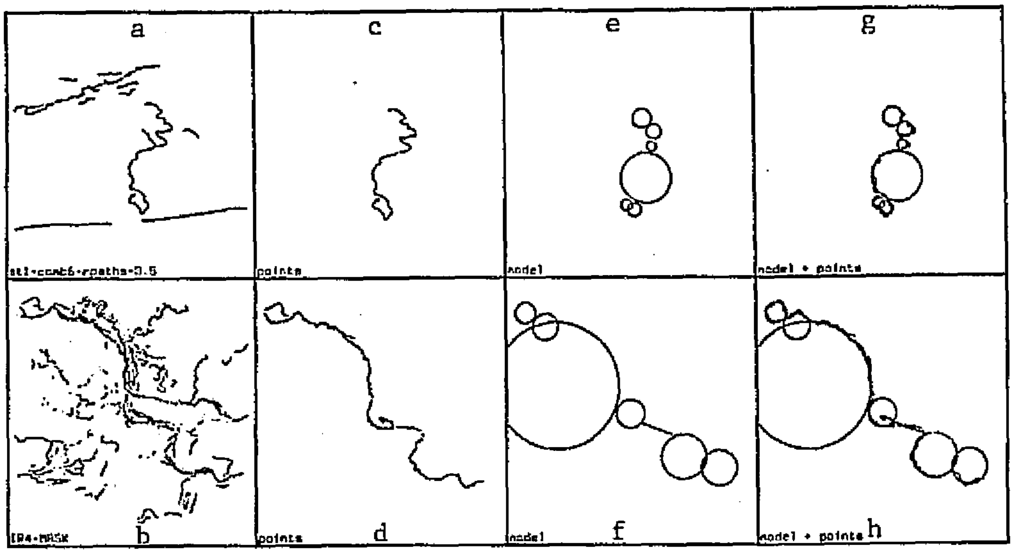


Figure 7. Structure Descriptions

Figures 3(h} and 4(h) were obtained by adding the binary images produced by the linear-structure finders. Consequently, in the combined image the values are greater than one at those pixel positions where linear structure was seen in more than one separation image. We treat this combined produce as a new "grey-level" image and, once again, apply the linear-structure finders. The results obtained from applying these procedures to Figures 3(h) and 4(h) are depicted in Figures 5(b) and 6(b). Figures 5(a) and 6(a) show an intermediate result before we cull short structures. For each of the structures in Figures 5(b) and 6(b), we calculate the average "intensity", that is the average number of original separation images exhibiting that linear structure. Figures 5(c),5(d),5(e),5(f),5(g),5(h) and 6(c),6(d),6(e),6(f),6(g),6(h) reveal which segments would remain if we thresholded the "intensity" values at 1, 1.5, 2, 2.5, 3, and 3.5, respectively.

We build a description of the linear structures from one of these images. The image we use will depend on the final matching procedure. If we wish to attempt to first match the "strongest" structures we use the image resulting from a high threshold. On the other hand, if we wish to match the complete structure, the unthresholded image would appear to be more appropriate. In the next section, where we discuss the nature of the structure description, we use as examples the foregoing two extremes. In the case of the oblique image, we have used the "intensity" image at a threshold of 3.5 (Figure 7a), while for the other case, the vertical infrared image, we employ the unthresholded image (Figure 7b).

3. Describing the Linear Structures

The means used to describe a linear structure is not independent of the use to which this description will be put. A description that makes it possible to reproduce the structure is different from one that is sufficient to recognize it. As matching is our goal, we want a description that is general enough to be unaffected by noise in the data, but specific enough to distinguish among structures that the human visual system would classify as different. To the extent feasible, the description must be invariant with respect to the variations that can occur in the data. Specifically, we want the description to be independent of orientation, scale, and vantage point.

Our matching process will compare graphs of symbolic descriptions. We will use as little metric information as possible. Consequently, the descriptions we employ are symbolic ones, the primitive entities in each of which have qualities that are themselves symbolic. For example, a primitive may be a straight-line segment whose properties, such as an intersection angle (with some other primitive), have values *acute*, *near-colinear*, etc. rather than a value in degrees.

The primitives we have chosen to use are straight-line segments, arcs of circles, and model-less, that is, data we prefer to describe as indescribable, data for which the data set itself is the most apt description. The choice of these few primitives stems from the observation that human description of linear structures seems to be based on curves and straight lines – moreover on whether adjoining curves curve the same or opposite ways and whether adjoining pieces of the structure intersect in particular ways. It is also a fact that humans find certain parts of the structure too difficult to describe, and assign them some generic term like "wiggles".

Selection of the description primitives is only half the task of description building. We need to be able to divide the linear structure into parts and assign a primitive to each. Usually the task of dividing the linear structure into parts and describing each of these parts has been handled as two relatively independent processes in which partitioning has preceded parts description. The difficulty with this approach is that some characterization of the breakpoints between parts has to be found. Generally, this characterization is based only on local properties of the linear structure, even though neighborhood information or local inhibition may be employed so as to benefit from more broadly based information. In this respect, the task of describing a structure in terms of its primitive parts appears to have been replaced by the more difficult undertaking of describing breakpoints. Our concern is to find the "best" description without first having to find the "best" subdivision. Furthermore, we would like "best" to be defined in terms of a global criterion rather than local properties of the structure.

The advantage of defining best in terms of a local criterion is that many candidates for the definition of "best" spring to mind. The disadvantage of defining "best" in a global sense is the lack not only of likely definitions, but also of computationally effective algorithms for finding this optimal solution. However, a description that views the data from a "gestalt" perspective seems more likely to be independent of image orientation, scale, and vantage point than one that applies local data measures to define the optimal description. We define best description, as the one that minimizes the number of symbols needed to encode the linear structure in terms of our description primitives.

4. Minimal Encoding

The need to match data to description primitives is a central aspect of decision theory and pervades artificial intelligence research. It is a human's ability to abstract data in terms of descriptive models that distinguishes human information processing from its electronic namesake. Effective data abstraction is a balance between two competing requirements. On the one hand a descriptive model must fit the data adequately, while, on the other, the descriptive model must not be needlessly complex. The criterion we use to select among competing descriptions is based on the work of Georgeff and Wallace [2], in which the description considered "best" is the one that can be encoded in the fewest symbols.

Suppose we wish to send data to some receiver so that he can recreate the data to some preselected level of resolution. The sender and receiver have agreed on a language for this communication that consists of a set of primitive elements. What is the most efficient encoding of the data; which message has the minimal encoding length? Consider the example of sending a message that describes a linear structure. The latter can be thought of as a list of x and y coordinates. Let us further suppose that the language of communication contains three primitives: straight-line-segments, arcs-of-circles, and model-less-segments. Is it more efficient to send the data as a single model-less-segment primitive, that is, as a list of (x,y) coordinates, or might it be more efficient to describe the data by one or more of the other primitives, specifying sufficient information to describe how the actual data differ from the primitives?

The message can be viewed as a list

$$((M_1, D_1), (M_2, D_2), ...) \qquad ,$$

where M is the specification of the primitive, D the specification of the data in terms of the selected primitive M. Let us consider an example. Suppose we have a data set that approximates a straight-line segment. We could communicate this by specifying a straight-line-segment primitive M, where M consists of a code for the straight-line-segment primitive and parameters that specify the actual straight line segment. These parameters might be the endpoints of the line. We also need to specify the actual data in terms of this primitive M. The data specification D might, for each data point, speciíy its coordinates as a distance along the line (from its centre) and the perpendicular distance from the point to the line.

As the expected distances from the points to the line are small, we shall choose an encoding of these distances so that the more probable of these, the smaller distances, are encoded in fewer symbols (or bits) than those that are less likely. In the actual examples we shall describe later, we assumed a Gaussian distribution for these perpendicular distances and we encoded optimally for that distribution. The optimal encoding length is just the negative logarithm of the probability, i.e., the function denoted as "information" in information theory.

If we have a small number of data points fewer symbols may be needed to communicate the data as a list of points; if, however, there is a large number of data points that exhibit behaviour consistent with a primitive, it will probably be cheaper to encode this data set as the primitive and then specify the data in terms of that primitive. Of course we are not just comparing the encoding of all the data with either one primitive or another. It might be more efficient to encode the data as a few primitives, with each primitive "explaining" a different part of the data. The encoding we select is the one that is globally best in explaining all the data.

A way of viewing the message form outlined above,.

$$((M_1, D_1), (M_2, D_2), \ldots) \qquad ,$$

is to look upon M as the overhead of introducing another primitive while D represents the quality of the fit between the data and the primitive. Of course, since different primitives have different M's, M also weights each primitive's efficiency for encoding data. In comparing message length we are balancing the complexity introduced by adding an extra primitive to the description of the data against the quality of fit between the assembled primitives and the data values.

Although the above discussion focused on encoding messages for communication, we use minimal encoding length as the criterion for finding the best description of a linear structure – without any interest on our part in actually transmitting the data. This of course means that we only have to decide how many symbols would be used if we were to encode the linear structure in a particular manner rather than actually doing the encoding. We can use the results of information theory to determine the optimal encoding length without even having to understand what the optimal encoding scheme is. That is, information theory gives us an operator, or a measure, that we can apply to a description to determine how many symbols we would need if we were to encode it optimally, without any consideration of the actual encoding scheme and without the need to do the encoding.

Let us consider our application, encoding linear structures in terms of three primitives: straight-line-segments, arcs-of-circles, amd model-less-segments. We will assume that the data are specified on a NxM grid, and that the noise in the data will induce a Gaussian distribution of the data points around the generating primitive. Given that all grid points are equally likely, the cost in bits of encoding a grid point is $logN + logM$, (log is to the base 2). Now consider the three alternative ways of encoding r data points (using one primitive only).

Model-less-segment:

We need a code to specify that the primitive being used is the model-less-segment. As there are only three primitives, and we assume that they are all equally likely, it costs $log3$ bits to specify the code. Specification of the data in terms of this primitive will require in turn that we specify r grid coordinates, that is, a cost of $r(logN + logM)$ bits.

Straight-line-segment:

We can specify the straight-line-segment primitive by specifying the endpoints of the line segment. This costs $2(logN + logM)$ bits. In addition, the cost of specifying the code for this primitive is $log3$. To specify the data in terms of this primitive we will, for each data point, specify a distance along the line and the perpendicular distance from the point to the line. If the line segment is of length l (in grid units) then, to specify r distances, if we assume that all distances are equally likely, will cost $rlogl$ bits. If it is also assumed that the data points have a Gaussian distribution about the primitive model, the cost of specifying r perpendicular distances is

$$\sum_{r\,pts} -log\left(\frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-d^2}{2\sigma^2}}\right) \qquad ,$$

where d is the perpendicular distance from the point to the line, and σ the standard deviation associated with the distribution. When the above expression is expanded, the sum over the d's is just the sum of the residuals squared that is calculated when the line is fitted to the data by least-squares methods.

Arcs-of-circles:

We specify the arcs-of-circles primitive by specifying the endpoints of the arc and one other point on the arc. This costs $3(logN + logM)$ bits, while the cost of specifying the code for this primitive is $log3$ bits. To specify the data we use the same scheme as we did for the straight-line-segment primitive.

Using these costing functions and a search algorithm that examines the various ways for partitioning a linear structure into primitives, we find the best description of that structure.

5. Results

The results of using the foregoing procedure on some of the linear segments found in Figures 1 and 2, (and shown in Figures 7(a) and 7(b)), are depicted in the remaining panes of Figure 7. From Figures 7(a) and 7(b) we have selected some linear structures. The selected structures, which form the main course of the Eel river, are shown in Figures 7(c) and 7(d). Our interest is in determining whether the description built from one image is the same as that from the other. Of course, in the final version of the structure builder we would need to handle all the segments simultaneously, but this will necessitate considerable improvement in the search algorithm to keep computational costs down to a reasonable level.

Figures 7(e) and 7(f) show the primitives returned. The arc of circles are shown as full circles to improve readability. In Figures 7(g) and 7(h) the primitives have been overlaid on the data to show the quality of fit. In assessing these results, one should keep the purpose of this description in mind. We want to extract a description of the linear structure in terms of lines and curves, in terms of the manner in which parts intersect (acute angles, near-colinearity, etc.), in terms of relative curvature (tight curves, gentle curves, and the like), and in terms of the sequencing of parts in the structure. Given that the two images are viewed from very different vantage points, that the scale is quite different (not even constant in one image), that one image was taken in the infrared band and one in the visible band, that the images were taken one-and-a-half years apart during different seasons, and that no semantic information was used in the processing, the closeness of the resulting descriptions is noteworthy. This points to the usefulness of processing the data in the above manner; namely, the method of finding the linear structures; the primitives used to encode the structure; and the encoding-length measure as a criterion for best description.

Figure 7 shows the results obtained with real data. Similar results have been obtained in experiments that employ other real data sets. Justification of the

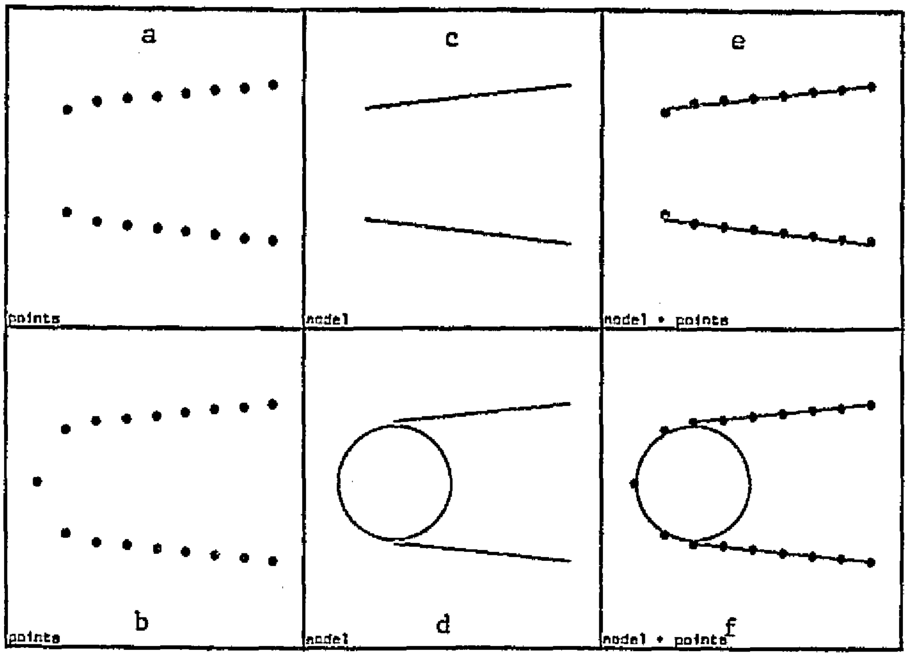


Figure 8. Encoding of Synthetic Data

method, however, requires further extensive experimentation. To better understand the behavior of the description builder we include an example using synthetic data. The data points are shown in Figures 8(a) and 8(b). In Figure 8(b) one extra data point has been added to those shown in Figure 8(a). The resulting descriptions are shown in Figures 8(c) and 8(d) and overlaid on the data in Figures 8(e) and 8(f). The addition of one critical point alters the description, an effect not unknown in the human visual system. The resulting descriptions seem to match those perceived by humans when they are presented with Figures 8(a) and 8(b). While we could not claim that minimal encoding is the criterion used by the human visual system for description building, we note that this criterion conforms to the type of behavior we would want to achieve if we were modeling the visual system. Of course, if the resultant description is sensitive to every addition or deletion of a data point it is of little use. In general, the minimal-encoding-length description appears to be stable with respect to data changes, except when "critical" points are added or deleted.

6. Matching the Descriptions

If the description we obtain from the description builder characterizes the data and is invariant with respect to orientation, scale, and vantage point, the burden of matching descriptions is lightened considerably. It is our intent to match descriptions at the symbolic level, to represent the descriptions found by minimal encoding as graphs of symbolic entities, and to match those graphs on the basis of their structure. Of course, it is unlikely that the graphs derived from different images will match perfectly. Nevertheless, from a prospective match we can find correspondences in the original images, and calculate the camera transformation between the images.

This procedure allows data in one image to be transformed into the other. It means that we can transform a linear structure found in one image into the other image. For those parts of the graph where there is a mismatch we can ask the question: how would the linear structure that is associated with the mismatch be encoded if it were first transformed into the other image and then encoded? In this manner we can attempt to explain the graph mismatches. If we cannot explain the mismatches we should consider another match of the graphs. Through this process of hypothesis and verification, we seek to avoid acceptance of a transformation that does not explain "all" the data.

7. Conclusion

Having found the linear structures in an image, we are faced with two major tasks before we can use these structures to find the correspondence between different images of a scene. We need to be able to describe these structures in a way that is independent of the variations that can occur between the images, and we need to be able to match these descriptions to find the relationship between the images.

In considering structure description we show that the usual technique of dividing the structure into parts and then describing the latter can be replaced by a procedure that finds the "best" description of the data on the basis of a global view of that data. This technique simultaneously divides the structure into parts and describes them. "Best" is defined as the cheapest encoding of the data when we consider the trade-off between the quality of explanation of the data and the complexity of that explanation.

This approach produces a description of linear structures that appears relatively insensitive to the vantage point, scale, and orientation of the original images. It may prove to be a description that enables easy matching, and hence an effective approach to solving the problem of image-to-image correspondence.

References

1. Fischler, M.A. and Wolf, H.C., Linear Delineation, *Proceedings of Computer Vision and Pattern Recognition Conference*, Washington, D.C. 1983, pp 351-356.

2. Georgeff, M.P. and Wallace, C.S., A General Selection Criterion for Inductive

Inference, in: O'Shea, T. (Ed.), *Proceedings of Advances in Artificial Intelligence, Pisa, Italy, September* 1984, North-Holland, Amsterdam, 1984.

Analysis of Subpixel Registration

Carlos A. Berenstein+*

Laveen N. Kanal+*

David Lavine*

Eric C. Olson*

Eric Slud+*

+ L.N.K Corporation
* University of Maryland

Abstract

This report describes work in the area of subpixel accuracy in image registration and edge detection. Two main directions of research were pursued: edge detection and matching based on the digital geometry of edges and random field models for probabilistic analysis of registration error. In the edge detection approach, error bounds and error probabilities were computed using theoretical models. Algorithms were developed and tested on simulated imagery. The methods appear promising for high accuracy edge position estimation and registration, though further refinement of the procedures will be required. Using random field models, a statistical measure of the quality of the cross correlation peak as an estimate of the offset between a sensed and a reference image was developed. Simulations were performed to determine the validity of this estimate with real imagery and to study the results of interpolating digital correlation functions to estimate the translation offset to subpixel accuracy.

Table of Contents

Section Title

0 .... Introduction
1 .... Introduction to geometric registration
2 .... Digital straight line segment parameter estimation
3 .... Digital line formula proofs
4 .... Feasible region shape
5 .... Infinite digital lines
6 .... Invariant line measure
7 .... Digital line-probabilistic analysis
8 .... Digital line fitting with known slope
9 .... Expected error for vertical offset estimation using slope
10 .... Expected error with pixels incorrect
11 .... One-dimensional edge projections
12 .... Fitting a digital edge to an image
13 .... Pairs of lines
14 .... Geometric registration summary
15 .... Random fields and subpixel accuracy
16 .... Objectives of the simulation study
17 .... The six reference images
18 .... Simulation results
19 .... Interpretation of results: conclusions
20 .... Conclusions
References

Section 0. Introduction

Subpixel accuracy in the registration of images and the location of objects within images is a topic of growing importance. Many users with high accuracy mensuration and classification requirements are faced with the difficulty of using high resolution monochromatic imagery or lower resolution multispectral imagery such as LANDSAT. The accuracy requirements have driven many users from the lower resolution imagery, but recently the possibility of using both types of imagery together has been considered. An additional interest in subpixel registration accuracy results from the need for high accuracy interband registration to improve classification accuracy.

This report describes our continuing efforts in the analysis of subpixel registration accuracy. Two main directions of analysis have been pursued. The first approach uses the digital structure of straight edges in imagery to aid in the matching of images. This work has two basic applications. One is registration, the major interest of our work. The second application is to the location to subpixel accuracy of structures in images. The two applications should be carefully distinguished. Subpixel registration accuracy only insures subpixel alignment based on control objects such as points or lines. Subpixel feature detection is of direct interest in features which may not even appear in previous images. Our methods are of particular interest in mensuration problems for new features such as measuring road widths and building sizes.

The study on the application of the digital structure of edges to registration accuracy took two major directions in the current study. First, we continued our theoretical analysis of the structure of digital lines. One key result of this effort was the development of proofs for the formulas describing the set of lines which could give rise to an observed digital edge. During the first year effort, we used formulas which were published [Do-Sm] without proof. Efforts to obtain further details from the authors were unsuccessful. Due to the key importance of these formulas and the difficulty of verifying them, we derived the results. The derivations as well as the formulas themselves are important when we consider the problem of digital edges in which some pixels are incorrect. This work thus provides a cornerstone for the theoretical analysis of noisy edges. Due to the complexity of the enumeration problem for digital lines, it is useful to develop asymptotic formulas for the count of digital lines. Asymptotic results are described here together with limited empirical verification. These results are in turn used to develop asymptotic error estimates for the accuracy of edge location given the correct digital edge.

The second part of our work on the digital structure of edges for registration accuracy involved the computation of average registration accuracy for various models. This portion of the study also led to the formulation of procedures for subpixel edge estimation which used the grey levels along the edge to estimate subpixel accuracy. One particularly promising approach led to an average accuracy of well under 0.1 pixel in a limited simulation study. This approach, which appears quite promising, is a natural extension of the digital edge matching of the first year and we are in the process of trying to extend the analysis to cover this method.

The second main approach taken in our study is based on the correlation structure of imagery. Using the theory of stationary random fields, we derived the probability of the peak of the cross-correlation function between a sensed and reference image being more than a specified distance from the true offset. Simulations were performed to determine the validity of this estimate and to determine the quality of the estimation of the offset using the peak of a quadric surface fitted to the correlation function.

The results described in this report provide a foundation for the modelling of subpixel accuracy. In addition, methods are developed which appear promising on an experimental level. The theoretical methods developed have been applied to simplified versions of the methods and work is currently underway to extend this analysis and test the methods more thoroughly.

NOTATION

$\lfloor x \rfloor$ - greatest integer <= x
$\lceil x \rceil$ - least integer >= x
$m \wedge n$ - greatest common divisor of m and n
$L(a,b)$ - line joining points a and b
$\psi(n)$ - 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 $n = p_\kappa^{a_1} \ldots p_\kappa^{a_\kappa}$ be the prime decomposition of n. Then
$\mu(n) = (-1)^\kappa$ if $a_1 = a_2 = \ldots = a_\kappa = 1$
$\mu(n) = 0$ otherwise

Section 1. INTRODUCTION TO GEOMETRIC REGISTRATION

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 one's 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 - Da]. Hybrid approaches may also be adapted. 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 model for geometric accuracy, it is useful to consider those aspects of the registration problem we wish to capture in our model. 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 function. 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 available 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 paper 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 $y = m(x - \Delta x) + b + y$ 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 here assume a set of pixels labelled edge pixels are provided by an edge detection procedure. Three cases can be 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, one could 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 is in doubt. Finally, we could 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 (Section 4). This allows us to give some probabilistic error estimates for the family of all digital lines (Section 7). This analysis provides the firm basis for the study of the second and third models, but since our results in these cases are not so complete yet, we leave them to future reports. We notice nevertheless that in many applications one only finds rather short digital lines, with information only on 10 to 20 pixels, and hence occasionally, e.g. if no analytic formulas are available, it might be perfectly justified to rely on computer-aided counting of possibilities when this counting is not too time consuming.

As a first step in our analysis we parametrize the chain codes of digital lines (see section 2 for definitions) by four parameters N,q,p,s as proposed by [Do-Sm] and use some formulas from the same paper. Regretfully the report [Do-1] in which these formulas are proved is not apparently available and hence we supply our own proofs in the text (Section 3). There is an excellent report [Ro-We] which seems to be generally unnoticed, and where there are several characterizations of those chain codes corresponding to digital lines among all possible strings of 0's and 1's. We do not use their results explicitly but they seem essential in the analytic study of the second and third model. We point out that both in [Do-Sm], [Ro-We] as in other work in the literature, no attention is paid to the counting of all digital lines of finite length. It is not enough to count lines through the origin as done in [Ro-We], and since our probabilistic analyses require such count, we give an exact formula for the number of all lines in section 7, which is not a straightforward generalization of the formula for lines through the origin. We also provide asymptotic bounds for the number of lines of a given length as well as provide grounds for the reasonable conjecture that this number L(N) is of the form

$$L(N) = N^3/\pi^2 + O(N^2 \log N).$$

The proof of this conjecture remains an open problem.

Section 2. 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 and strictly less than 1, we define its digitization as follows: To each intersection (a,b) between the line and a line $x = a$, a an integer, we associate the pixel with lower left hand corner $(a, \lfloor b \rfloor)$. (see figure 2.1). The chain code of the sequence of pixels with lower left hand coordinates $(0,b_0)$, $(1,b_1)$, $\ldots,(N,b_N)$ is the sequence $c_1,\ldots,c_N$ where

$$c_i = \left\{ \begin{array}{ll} 0 \text{ if } \lfloor b_i \rfloor & = \; b_{i-1} \\ \\ 1 \text{ otherwise} & \end{array} \right\}$$

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 parametrization 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 parametrization 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 length of the chain code, i.e., the number of 0's and 1's. We note that not every string of 0's and 1's is generated by a line segment. For a characterization of those that are, see [W-R].

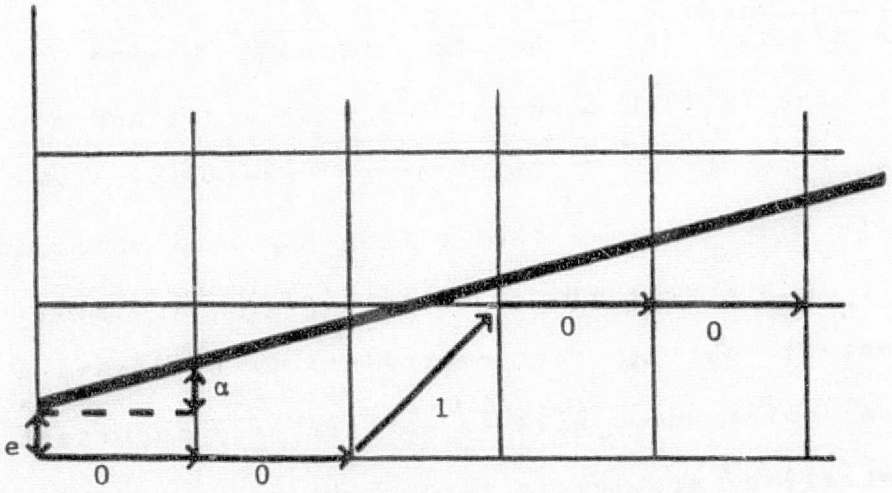


Figure 2.1 Chain code of a digital line. The digitization of the dark diagonal line has pixels with lower lefthand vertices (0,0), (1,0), (2,0), (3,1), (4,1), (5,1). The resulting chain code indicated by the arrows is 00100.

Next, q is defined to be the smallest integer such that there exists an extension c_{N+1}, c_{N+2}, ..., with $c_1, c_2, c_3, \ldots$ periodic with period q. Define p to be the number 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 $p \wedge q = 1$ 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 i-th element c_i, of this chain code is given by

$$c_i = \lfloor i(p/q) \rfloor - \lfloor (i-1)(p/q) \rfloor , \; i=1,2,...N$$

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 code string can be obtained by

$$c_i = \lfloor (i-s)(p/q) \rfloor - \lfloor (i-s-1)(p/q) \rfloor , \; i = 1,2,...,N$$

The parameters satisfy the constraints $0<=p<=q<=N$ and $0<= s<=q-1$. A point which will be particularly important for the registration problem is that there are constraints on the parameters other than the above inequalities. These additional constraints are described in Section 3. Our interest in these matters stems from the need to enumerate the digital lines satisfying various conditions. If it were 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 $0<=p<=q<=N$ and $0<=s<=q-1$.

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.

Since the smallest period plays an important role, let us point out two ways of computing it. The first one might be easier to use for long strings with the help of the FFT, the second one is very convenient for direct computation in short strings.

For the first algorithm extend the chain code to the right, with period N, i.e. $c_{i+N} = c_i$. Then

$$q = \inf \{j:\ 1<=j<=N \text{ such that } 1/N \sum_{i=1}^{N} (-1)^{c_i + c_{i+j}} =1\}.$$

Notice that the maximum value of the average in the definition of q is precisely 1. In the second algorithm, we extend the code chain in both directions by zeroes and consider

$$q = \inf\ j:\{1<=j<=N \text{ such that } 1/(N-j) \sum_{i=1}^{N} (-1)^{c_i + c_{i+j}} =1$$

with the understanding that if the set of j's is empty we take q = N. What this really means is that we slide successively to the right the chain code and compare the tail end of the original chain code with the first portion of the shifted chain code, the value q corresponds to the first perfect match, if there are no matches then q = N.

The primary result of [Do-Sm] is a description of the set of all lines whose digitization over the 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. We will call this plane the dual plane. The proof of the following formulas will be found in the next section.

Define functions F and L by:

(2) $F(s) = s - \lfloor s/q \rfloor q$

and

(3) $L(s) = s + \lfloor (N - s)/q \rfloor q$

Let ℓ be defined by the equation:

(4) $1 + \lfloor \ell(p/q) \rfloor - \ell(p/q) = 1/q$ and $0 < \ell < q$,

or, what is the same, by the fact that $\ell p \equiv -1 \pmod{q}$. The set of feasible lines is a convex quadrilateral in (e,α)-space with vertices A, B, C, D given by

(5) $A = (\lfloor F(s)p/q \rfloor - F(s)p^{+}/q^{+}, p^{+}/q^{+})$

(6) $B = (\lceil F(s)p/q \rceil - F(s)p/q, \ p/q)$

(7) $C = (1 + \lfloor F(s + \ell)p/q \rfloor - F(s + \ell)p/q, p/q)$

(8) $D = (1 + \lfloor F(s + \ell)p/q \rfloor - F(s + \ell)p^- /q^- , p^- /q^-)$

where .

(9) $q^+ = L(s + \ell) - F(s)$, $p^+ = (pq^+ + 1)/q$

(10) $q^- = L(s) - F(s + \ell)$, $p^- = (pq^- - 1)/q$

The above expressions for the vertices of the feasible quadrilateral will be discussed in greater detail in later sections. We note here that neither of the vertices A, C, D nor the points in the two sides of the quadrilateral determined by them correspond to lines that have the chain code (N, q, p, s) after digitization. It is also very important to note that (since we are working with lines of non-negative slope < 1 and non-negative ordinate to the origin < 1) the quantities p^+, q^+, q^- are strictly positive, while $p^- >= 0$ (in fact, from (10) it follows that $p^- = 0$ only if $p = q^- = 1$). This remark, which is omitted in [Do-Sm], is crucial to provide a correct count of all distinct digital lines of length N (cf. Proposition 10). It will be proved in the next section.

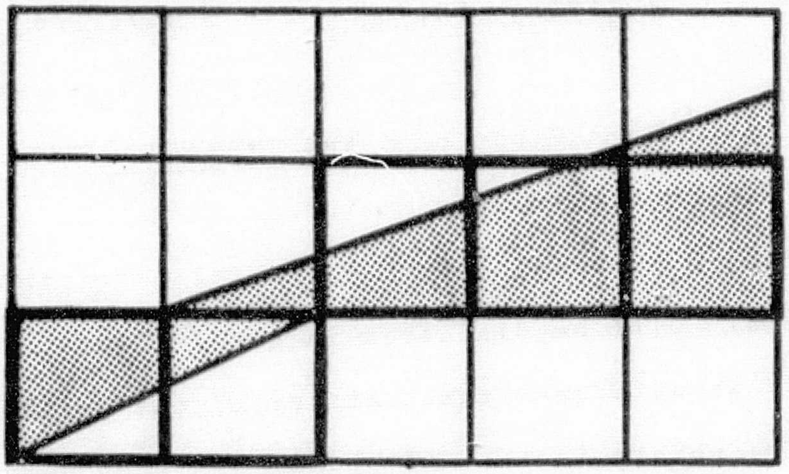


Figure 2.2 Feasible region for a digital line. The digital line consisting of those pixels with darkened boundaries has the shaded area as its feasible region.

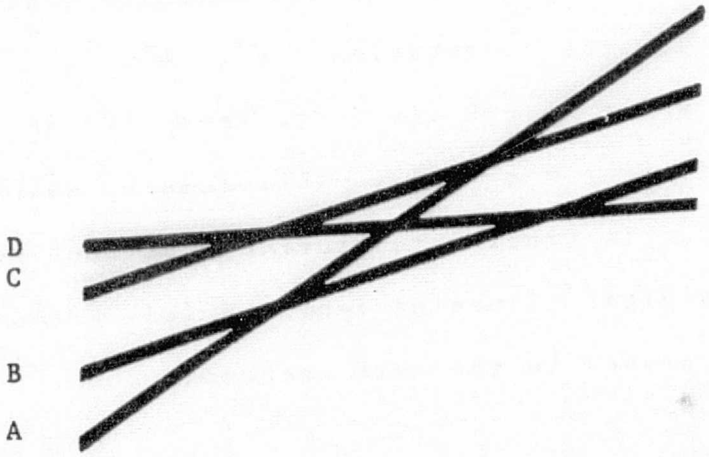


Figure 2.3 Intersections for the feasible region. The four boundary lines A, B, C, and D of a feasible region are shown. The intersection of A and D always lies between the parallel lines B and C. These lines in the x,y space correspond to the vertices A,B,C,D of the feasible quadrilateral in the (e,α) parameter space.

Section 3. Digital Line Formula Proofs

We provide here the proofs of the formulas (4) - (10) of previous section, since as we pointed out the manuscript [Do- 1] is unavailable and some errors might have occured in the original derivations of these formulas [Do-Sm] . The reader may safely skip this section without lack of continuity.

We begin with an observation from [R-W] which remains valid for lines of finite length N due to the fact that all digital lines arise out of the digitization of lines of the form $y = p/q\ x + m/q$, $0 <= m < q$, $p \wedge q = 1$ (We assume $q>1$ since for $q = 1$ we only consider the line $y = 0$).

Lemma 1: For a line of slope p/q, vertical displacement upwards by $1/q$ units results in a cyclic shift of the code by ℓ digits to the right within each segment of length q, where ℓ is the solution of the equation

(11) $\ell p \equiv -1(\mathrm{mod}\ q),\ 0 < \quad < q.$

Proof: For the purpose of this lemma we can consider a digital line of infinite length generated by the line of equation $y = (p/q)x + e$, $0 <= e < 1$. When $e = 0$ the line contains exactly those points in the lattice $\mathbb{Z}^2$ of the form $(kq,\ kp)$, $k \in \mathbb{Z}$. When e is increased the code remains the same until new lattice points lie on the line, if a new lattice point $(q',\ p')$ appears for a value e_0, then one gets a transposition of the 0 which corresponds to $x = q'$ and the 1 that corresponds to $x = q' + 1$ for values of $e < e_0$, $e \rightarrow e_0$, as a quick look at the picture shows. The points $(q' + kq, p' + kp)$, $k \in \mathbb{Z}$, belong to the line $y = (p/q)x + e_o$ and no other lattice point does, otherwise the slope could not be p/q with $p \wedge q = 1$. Notice that because of the upward shift we have $0 < q' < q$ and $0 < p' < p$ for the first value e_o where the above transposition takes place. This implies that the code of the line $y = (p/q)x + e_o$ is the same as that of the line $y = (p/q)x$ with a right cyclic shift in each period of size q'. The same fact will hold between any two successive upward shifts of the same magnitude e . It remains to identify this magnitude e_o and the value q'. Since e_o is the first positive value for which such a shift recurs, we have that the parallelogram of vertices (o,o), (q,p), (q',p') and $(q+q', p+p')$ is a minimal parallelogram on the lattice (see [H-W, p. 28]) and hence its area is exactly one, i.e.

(12) $p'q - q'p = qe_o = 1$

From this it follows that $e_o = 1/q$ and q' is the quantity defined as ℓ by (11). After successive transitions of this size, (or what is the same after q successive cyclic shifts of size ℓ) we go back to the original code. ■

We are now ready to relate the code (N,q,p,s) to the family of lines that induce the same code. First, we know that it is induced by a line of the form

(13) $y = (p/q)x + m/q \quad 0 <= m < q$

and we would like to find the relation between s and m. The lemma 1 tells us that

(14) $\ell m \equiv s (\mathrm{mod}\ q)$.

Hence, we have

(15) $s = \ell m - kq$ for some $k >= 0$,

in fact, using the function F introduced in (2) we can write

$$s = F(\ell m)$$

since all the function F does is to select a representative in $0,1,\ldots,q-1$ for every element in $\mathbb{Z}/q\mathbb{Z}$. Substituting the expression (15) into (13) we obtain

(16) $y = (p/q)s + m/q = (p\ m)/q + m/q - kp = p'm-kp \in \mathbb{Z}$

where the third identity was obtained using (12) (Recall $\ell = q'$ in (13)). That is, we see that the value s has the property that for $x = s$ the point in the line (13) is a lattice point. Furthermore, this is the first lattice point in the interval $0 <= x < \infty$ which lies on the line, otherwise the slope of the line will be rational with denominator strictly smaller than q and in contradiction to the fact, we are assuming that the chain code has smallest period q (this justifies the letter F to denote the function on $\mathbb{Z}/q\mathbb{Z}$ as defined by (2)). We can also conclude that the value y in (16) is given by

$$y = \lceil (p/q)s \rceil \text{ and } m/q = \lceil (p/q)s \rceil - (p/q)s$$

since $0 <= m/q < 1$. This tells us that the line (13) coincides with the line B given by the dual coordinates (6), i.e.

$$e = \lceil F(s)(p/q) \rceil - F(s)(p/q), \quad \alpha = p/q.$$

As a corollary of this representation and lemma 1 we obtain that the line

$$y = (p/q)x + (m+1)/q$$

coincides with the line C described by (7), namely the infinite line will have first lattice point when $x = F(s + \ell)$ and since $0 < (m+1)/q <= 1$, we have for the corresponding value of y that

$$y = \lfloor F(s+\ell)(p/q) \rfloor + 1$$

hence, it follows that the dual coordinates of C are in fact

$$e = 1 + \lfloor F(s+\ell)(p/q) \rfloor - F(s+\ell)(p/q), \alpha = p/q.$$

We could by abuse of language denote the chain code of C as $(N,q,p,F(s+\ell))$ but it might not be the case that q is the smallest period of this code, as we have in the example:

$$B \leftrightarrow (11,11,3,0) \leftrightarrow y = (3/11)x$$

so that $\ell = 7$ and

$$C \leftrightarrow (11,11,3,7) \leftrightarrow y = (3/11)x + 1/11$$

which has code

0001 001 0001

whose smallest period is 7.

Let us call "last lattice point of a line" the one with largest abscissa still <= N. We are going to consider now two other lines defined by:

(17) A: is the line passing through the first lattice point of the line B and the last lattice point of the line C.

(18) D: is the line through the first lattice point of C and the last lattice point of B

Those two lines are well-defined and not vertical since no point of B coincides with a point of C and these lattice points cannot be above each other. Neither of these two lines nor the line C have code (N,q,p,s) since they pass through lattice points different than those corresponding to B. Let us first derive an important property of this collection of four lines A,B,C,D. We note that if we have two points in the dual space $L_1 = (e_1 , \alpha_1)$, $L_2 = (e_2 , \alpha_2)$ which correspond to lines with code (N,q,p,s) then the point L of coordinates

$$e = \lambda e_1 + (1-\lambda)e_2 , \quad \alpha = \lambda\alpha_1+(1-\lambda)\alpha_2 , \quad 0 <= \lambda <= 1$$

corresponds to a line which passes through the same pixels as L_1 and L_2, in fact for a given x the ordinate y of the corresponding point in the line L is just $y = \lambda y + (1 - \lambda)y_2$, with $(x,y_1) \in L_1$, $(x,y_2) \in L_2$. So the set of lines with code (N,q,p,s) forms a convex set in the dual space. Furthermore, it is easy to see that this convex set Ω must contain an open neighborhood of the open segment defined by B and C, this is simply the fact that a line between B and C passes through the same pixels as B but passes through no lattice points (by lemma 1) hence its slope can be jiggled a bit and keep the same code.

We are going to look at the (possibly degenerate) quadrilateral with vertices A,B,C,D. For that purpose we need to find the equations of the sides, e.g. the side AB. We are looking for the equation of a line in the dual space, that is an equation of the form

$$a\alpha + be = c, \; a^2 + b^2 <> 0,$$

The definition of A shows that A and B have a point in common, namely the first lattice point of the line B, say (x_0 ,y_0), and hence every line which corresponds to a point in AB passes through the same point, i.e. it satisfies the equation

$$x_0\alpha + e = y_0$$

which has the desired form. Calling (z_0 ,w_0) the last lattice point of B, (x_1 ,y_1) the first lattice point of C, (z_1 ,w_1) the last lattice point of C we have the following equations

$$(19) \quad \begin{array}{ll} AB: & x_0\alpha + e = y_0 \\ BD: & z_0\alpha + e = w_0 \\ DC: & x_1\alpha + e = y_1 \\ CA: & z_1\alpha + e = w_1 \end{array}$$

We note that on one side of the line AB we have $x_0\alpha + e > y_0$ and on the other side we have $x_0\alpha + e < y_0$. On this second side we have that no line passes through the same pixels as B, hence it cannot have code (N,q,p,s), therefore $\{e,\alpha): x_0\alpha + e >= y_0\}$. We can conclude by a similar reasoning that:

$$\Omega \subseteq \{x_0\alpha + e >= y_0\} \cap \{z_0\alpha + e >= w_0\} \cap \{x_1\alpha + e < y_1\} \cap \{z_1\alpha + e < w_1\}$$

which is the quadrilateral determined by A,B,C,D.

To finish the proof all we need to know is that the half-open segments (A,B] and [B,D) are in Ω. For the first one it follows from the fact that there are no lattice points in the open triangle whose sides are the y-axis, the line A and line B, otherwise we consider the line through such a lattice point (x_2 ,y_2) and (x_0 ,y_0), it will have the same code as B but clearly has period $x_0 - x_2$ which is strictly less than q (recall $x_0 = s < q$). Similar reasoning holds for the other segment. Summarizing, we have:

Lemma 2: The convex set Ω of all lines coincides with the (possibly degenerate) quadrilateral of vertices A,B,C,D.

Lemma 3: It is never the case that A = D, i.e. it is impossible that we have simultaneously that $(x_0, y_0) = (z_0, w_0)$ and $(x_1, y_1) = (z_1, w_1)$.

Proof: In this case Ω is a triangle (it cannot be a segment since $A \in BC$ says that A is parallel to B, which contradicts (17)). One of the sides is BC. Hence Ω cannot contain an open neighborhood of the open segment (B,C), this contradicts an observation made above. ■

It remains to write down the dual coordinates of A and D. For that we need to consider which is the abscissa of the last lattice point on the lines B or C. For the line B we have that the abscissa of the first lattice point is x = s, hence the last lattice point is $x = s + kq$, $k >= 0$, $x <= N$. This implies that $k = \lfloor (N-s)/q \rfloor$, so we get

$$x = s + \lfloor (N-s)/q \rfloor q = L(s) \text{ (as defined by (3)).}$$

since the function L turns out to be a function well-defined on $\mathbb{Z}/q\mathbb{Z}$, we have that the abscissa of the last lattice point in C is $L(s + \ell)$. We get the following formulas companion to (19), which we will need in a later section:

$$(20) \quad \begin{array}{lll} x_0 = F(s) & y_0 = \lceil (p/q)F(s) \rceil = & (p/q)F(s) + m/q \\ z_0 = L(s) & w_0 = \lceil (p/q)L(s) \rceil = & (p/q)L(s) + m/q \\ x_1 = F(s+\ell) & y_1 = 1 + \lfloor (p/q)F(s+\ell) \rfloor = & (p/q)F(s+\ell) \\ & & + (m+1)/q \end{array}$$

$$z_1 = L(s+\ell) \quad w_1 = 1 + \lfloor (p/q)L(s+\ell) \rfloor = (p/q)L(s+\ell) + (m+1)/q$$

The line A passes through the points (x_0, y_0) and (z_1, w_1) hence

$$\alpha = (w_1 - y_0)/(z_1 - x_0)$$

Define

$$q^+ = z_1 - x_0 \; , \; p^+ = w_1 - y_0$$

Then

$$q^+ = L(s+\ell) - F(s)$$

and $p^+ = w_1 - y_0 = (p/q)(z_1 - z_0) + 1/q = (p/q)q^+ + 1/q$ verifying (9) and also showing $p^+ \wedge q^+ = 1$. We already know that $q^+ <> 0$ we want to show that Lemma 3 implies $q^+ > 0$. In fact, we have

$$q^+ = \ell + \lfloor (N-(s+\ell))/q \rfloor q$$

and the only problem could occur if $s + \ell > N$. Then we would have $N - s < q$ and $s + \ell > q$ which implies that $L(s) = F(s)$ and $F(s+\ell) = L(s+\ell)$. This is precisely the situation forbidden by Lemma 3. Now we want to find the ordinate to the origin of A, we have

$$A: \quad y - y_0 = (p^+/q^+)(x - x_0)$$

hence, using (20) we obtain

$$e = y_0 - (p^+/q^+)x_0 = \lceil F(s)(p/q) \rceil - F(s)(p^+/q^+).$$

This finishes the verification of (5). Going through the same reasoning for the line D we see that its slope is given by

$$\alpha = (p^-/q^-), \quad q^- = z_0 - x_1, \quad p^- = w_0 - y_1$$

so that $q^- = L(s) - F(s+\ell)$ as required and

$$p^- = (p/q)z_0 + m/q - ((p/q)x_1 + (m+1)/q)$$

$$= (p/q)(z_0 - x_1) - 1/q = (p/q)q^- - 1/q$$

verifying the relation (10). Writing

$$D: \quad y - y_1 = (p^-/q^-)(x-x_1)$$

and using (20) again we get

$$e = y_1 - (p^-/q^-)x_1 = 1 + \lfloor (p/q)F(s+\ell) \rfloor$$
$$- (p^-/q^-)F(s+\ell)$$

which is the only thing left to check in (8) except for seeing that $q^- > 0$. But this is again Lemma 3. Since $q^- <= 0$ only could occur if simultaneously $N-s < q$ and $s+\ell < q$. A computation shows this leads to $F(s) = L(s)$ and $F(s+\ell) = L(s+\ell)$ which cannot happen.

Summarizing, we have proved:

Proposition 4: The formulas (2) - (10) defining the quadrilateral A,B,C,D are correct and furthermore $q^+ > 0$, $q^- > 0$, $p^+ > 0$ and $p^- >= 0$.

Section 4. 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 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 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 the parallel lines B and C of the feasible region (Section 2) have 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 introduce boundary effects which cause the spread to be greater near the ends of the chain code, but the following lemma shows that at least one point of the 1/q width channel is preserved.

Lemma 5: Using the notation of section 2, 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: It can be obtained from the way the line C was defined in Section 3 or otherwise from a direct computation, which we omit, using the ordinates to the origin that appear in (6) and (7). ■

We have established that lines B and C are separated by a vertical distance $1/q$. We have already pointed out in Section 3 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 these four values.

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 $[a,b] \subseteq [0,N]$ such that the lines A and D lie between the lines B and C over the interval $[a,b]$. To do this, we establish the following facts (see Fig. 2.3):

Let $I(.,.)$ 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) $I(D,C) <= I(A,C)$
4) $I(A,B) <= I(D,B)$
5) $I(D,C) <= N$, $I(A,B) <= N$

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 non-empty interval $[a,b]$.

Lemma 6. 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

$$\begin{aligned} Y_B - Y_A &= \lceil F(s)p/q \rceil - F(s)p/q - \lceil F(s)p/q \rceil + F(s)p^+/q^+ \\ &= F(s)(p^+/q^+ - p/q) \end{aligned}$$

Since $F(s) = s >= 0$, we are done if we show $p^+/q^+ - p/q > 0$. By the definition of p^+, q^+,

$$\begin{aligned} p^+/q^+ - p/q &= (pq^+ + 1)/(qq^+) - p/q \\ &= p/q + 1/(qq^+) - p/q \\ &= 1/(qq^+) \end{aligned}$$

By Proposition 4, we have $q^+ > 0$, hence we are done. ■

Lemma 7: 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

$$Y_D - Y_C = F(s+\ell)(p/q - p^-/q^-) = F(s+\ell)/(qq^-)$$

and we are in the same situation as in the previous lemma. ■

Lemma 8: $I(D,C) <= I(A,C)$

Proof: By (17) and (18) we have that $I(A,C)$ is the abscissa of the last lattice point of C, i.e. $I(A,C) = L(s+\ell)$, while $I(D,C)$ is the abscissa of the first lattice point of C, $F(s+\ell)$. From this it follows immediately the conclusion of the lemma. This can also be done by using the dual coordinates of A,D,C but at the cost of considerable computation. ■

The same proof yields:

Lemma 9: $I(A,B) <= I(D,B)$

From what we have just said, it follows that

$$0 \le a = \max(I(A,B),I(D,C)) \le \min\,(I(A,C),I(B,D)) = b \le N$$

hence, we are guaranteed that there exists an $X \in [0,N]$ such that the feasible region over x has height $1/q$. Therefore, if we pick the line L which is the average of B and C, we have

$$(21) \qquad \min_{0 \le x \le N} \; \max_{L \in (N,q,p,s)} |L(x) - L_0(x)| \le 1/(2q)$$

where $L(x)$, $L_0(x)$ represent the ordinate of the point in L, resp. L_0, with abscissa x.

The meaning of (21) is that given a digital line with period q in the sensed image and such that the underlying real edge has slope between zero and one, then we can determine the vertical aspect between sensed and reference images to an accuracy of $1/2q$ pixels.

Section 5. Infinite Digital Lines

The feasible region for infinite digital lines is easily computed using the results of Section 4. 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.

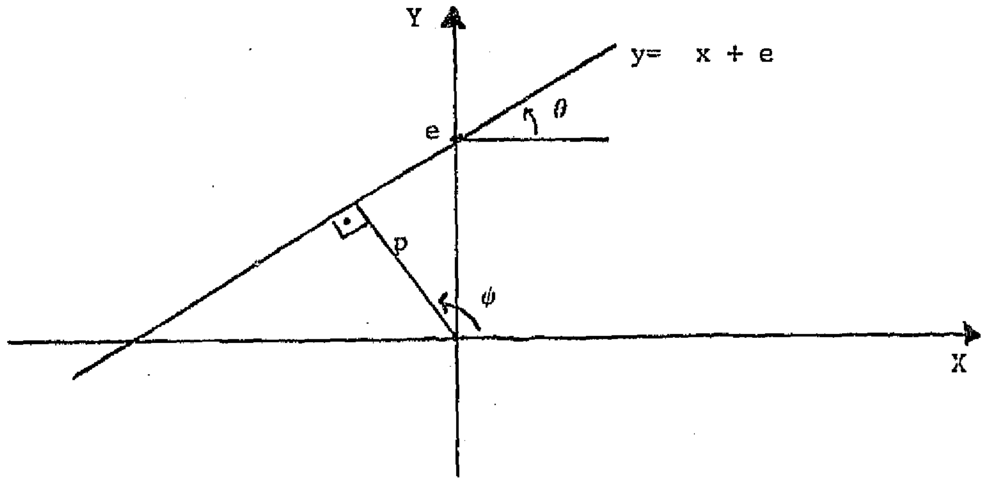
Before considering the periodic and aperiodic lines separately, we note that any two infinite lines with the same digitization are parallel. Let $y = mx + b$ and $y = nx + c$ be two lines. Then the difference, $h(x)$, in the y values of these lines at x is given by $h(x) = (m-n)x + (b-c)$. If m and n 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, 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 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 4 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 requires a different approach. We first cite a version of a classical result [H-W] 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 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 $\epsilon > 0$ 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 $mK_1 + b - \lfloor mK_1 + b\rfloor < \epsilon$ and $mK_2 + b - \lfloor mK_2 + b\rfloor > 1 - \epsilon$. 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 $\epsilon > 0$, we cannot change b by more than without changing the digitization. Hence b is fixed. Since m is also fixed, the channel is the single line $y = mx + b$.

Section 6. 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 quandary 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 (p,φ) polar coordinates of a line, in which the distribution is uniform with respect to the parameters as shown in [S, p. 28]. To write this measure in terms of the dual coordinates we appeal to the following figure:



We clearly have

$$p = e.\cos(\varphi - \pi/2) \text{ and } \pi/2 - \theta = \pi - \varphi$$

hence $p = e.\cos\theta$ and

$$dp \wedge d\varphi = (\cos\theta \, de - e\sin\theta \, d\theta) \wedge d\theta = \cos\theta \, de \quad d\theta$$

Using $\tan\theta = \alpha$ we obtain $d\theta = \cos^2\theta \, d\alpha = (1 + \alpha^2)^{-1} \; d\alpha$,

so finally

$$dp \wedge d\varphi = (1+ \alpha^2)^{-3/2} \; de \wedge d\alpha \tag{22}$$

is the invariant measure. We want to normalize (22) so that total measure of $0 <= e < 1$, $0 <= \alpha < 1$ is exactly 1. From

$$\int (1+ \alpha^2)^{-3/2} \; d\alpha = \alpha(1+\alpha^2)^{-1/2} \tag{23}$$

$$\int \alpha(1+ \alpha^2)^{-3/2} \; d\alpha = -(1+\alpha^2)^{-1/2} \tag{24}$$

we obtain that the normalized invariant measure is:

$$(25) \qquad d\mu = \sqrt{2}(1+\alpha^2)^{3/2} \quad de\,d\alpha$$

It is now easy to compute $\int_{\Omega} f(e,\alpha)\,d\mu$, where Ω is the quadrilateral A,B,C,D formed by the lines of code (N,q,p,s). It is just necessary to recall the equations (19) and (20) of the sides of this quadrilateral:

$$(26) \qquad \int_{\Omega} f d\mu = \sqrt{2}\left\{\int_{p/q}^{p^+/q^+}\left(\int_{y-x}^{w_1-\alpha z_1} f(e,\alpha)de\right)(d\alpha/(1+\alpha^2)^{3/2}) + \int_{p^-/q^-}^{p/q} \quad {}_{w_0 - \ z_0}^{y_1-\alpha x_1} f(e,\alpha)de \ (d\alpha/(1+\alpha^2)^{3/2})\right\}$$

In particular, using the definitions of p^+,q^+,p^-,q^- which appear after (20):

$$\mu(\Omega) = \sqrt{2}\left\{\int_{p/q}^{p^+/q^+}((w_1-y_0)+(x_0-z_1))(d\alpha/(1+\alpha^2)^{3/2}) + \int_{p^-/q^-}^{p/q}((y_1-w_0)+\alpha(z_0-x_1)(d\alpha/1+\alpha^2)^{3/2})\right\}$$

$$= \sqrt{2}\left\{\int_{p/q}^{p^+/q^+}(p^+-q^+\alpha)(d\alpha/(1+\alpha^2)^{3/2}) + {}_{p^-/q^-}^{p/q}(-p^-+\alpha q^-)(d\alpha/(1+\alpha^2)^{3/2})\right\}$$

$$= \sqrt{2}\left\{1/((p^+)^2+(q^+)^2)^{1/2} - (pp^+ + qq^+)/(p^2+q^2)^{1/2} + 1/((p^-)^2+(q^-)^2)^{1/2} - (pp^- + qq^-)/(p^2+q^2)^{1/2}\right\}$$

Regretfully, this expression is a bit complicated. One can compare it with the Lebesgue measure of Ω without much difficulty and finds

$$|\Omega| \le \mu(\Omega) \leqslant 2|\Omega|$$

In fact, for the analysis of next section we would like to compare the μ-measures or the Lebesgue measure with the measure on digital lines which assumes all of them are equally likely. Computations show that the invariant line measure tends to favor lines with small q, especially the horizontal line has a rather big weight; while the uniform measure on digital lines tends to favor lines with higher q. The following table gives our computations for lines of length N = 10.

TABLE 6.1

DIGITAL LINE INFORMATION

q	% TOTAL AREA	% PROBABILITY	% DIGITAL LINES
1	18.182	17.469	0.74
2	5.051	5.137	1.48
3	7.684	7.809	4.44
4	6.782	6.868	5.93
5	14.250	14.438	14.81
6	9.524	9.514	8.89
7	19.444	19.626	26.67
8	8.514	8.552	14.81
9	7.684	7.720	16.30
10	2.886	2.868	5.93

In the next section we discuss the error analysis of the procedure outlined by (21), under the measure that gives all digital lines equal weight, we leave for later the comparison with other error estimates based on the invariant measure.

Section 7. Digital Line-Probabilistic Analysis

A worst case bound on registration accuracy using a digital edge was developed in Section 4. 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 foımulated. 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 simple 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 asymtotic 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 4. 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. 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.

The problem of enumerating linear chain codes of lines through the origin was discussed in [R-W] where also an algorithm for generating the set of linear chain codes was presented. We have not found any estimates in the literature of the number of chain codes of a given length. The problem is that the the shortest period of the digital line of length N corresponding to a line

$$y = (p/q)x + m/q$$

might be strictly smaller than q. Since such lines generate all the possible digital lines and we can associate to each a code (N,q,p,s), the problem reduces to characterize those values of s for which this code does not coincide with $(N,\overline{p},\overline{q},\overline{s})$ with $\overline{q} < q$. The answer lies in the following.

Proposition 10: Given a code (N,q,p,s), the necessary and sufficient condition that it does not coincide with a code of strictly smaller period is that $q^{+} > 0$ and $q^{-} > 0$, where q^{+}, q^{-} are defined by (9) and (10).

Proof: The necessity of this condition is guaranteed by Proposition 4. In order to go further we have to analyze what condition on s ensures that $q^{+} > 0$ and $q^{-} > 0$. We have

$$\begin{aligned} q^{+} = L(s+\ell) - F(s) &= s+\ell+ \lfloor (N-s+\ell)/q \rfloor q - s \\ &= \ell + \lfloor (N-s+\ell)/q \rfloor q \\ q^{-} = L(s) - F(s+\ ) &= s + \lfloor (N-s)/q \rfloor q - (s+\ell) + \lfloor (s+\ell)/q \rfloor q \\ &= \lfloor (N-s)/q \rfloor q + (s+\ell)/q \rfloor q - \ell \end{aligned}$$

Note that if N-s >= q then we have that the digital line has period q, since the digits in the chain code corresponding to x = s+1,...,x = s+q <= N, form the chain code of the standard line, i.e. of y = (p/q)x. Of course in this case we also have $N - s - \ell >= q - \ell > 0$ hence $q^{+} >= \ell > 0$ and $q^{-} >= q - \ell > 0$. Suppose now N - s < q then the condition $q^{-} > 0$ implies that $s+\ell >= q$ and hence we have $F(s+\ell) = s+\ell-q$. Note that

$$N-F(s+\ell) = N-(s+\ell)+q >= q$$

since $q^{+} > 0$ implies that $N-(s+\ell) >= 0$. So we only have to prove that a line for which N-s<q, and $N-F(s+\ell) >= q$ has smallest period q. Notice that this says that the line $y = (p/q)x + m/q$ passes through a single lattice point at $x = s$, while the line $y = (p/q)x + (m+1)/q$ contains two lattice points, the first one with abscissa $F(s+\ell) < s$. We know hence that this second line has period exactly q since if we restrict ourselves to $F(s+\ell) + 1 <= x <= F(s+\ell) + q <= N$ the q digits in the chain code of the second line are those of the standard chain code. To prove that the original line has smallest period q it is enough to show that the same portion of its chain code has smallest period q since the period of a chain code cannot be smaller than that of any subchain. That is, we have reduced ourselves to show that the chain code of the digital line of length q corresponding to $y = (p/q)x + (q-1)/q$ has smallest period q. Calling c_i the standard chain code and c_i^* the chain code of this other line it would be enough to prove that the sequence $\{c_1^* ,\ldots,c_q^*\}$ is exactly the sequence $\{c_q,\ldots,c_1\}$ making an appeal to (1). Now, as we have argued in Lemma 1, the code $\{c_1^* ,\ldots,c_q^*\}$ is obtained from $\{c_1 ,\ldots,c_q\}$ by making $c_1^* =1$, $c_q^* =0$, $c_j^* =c$, $1<=j<=q-1$, while $c_1 =0$, $c_q =1$. To finish the proof we only need to show that the sequence $c_2,\ldots,c_{q-1}$ is symmetric, i.e.

$$c_j = c_{q-j+1}$$

But $c_j = \lfloor (p/q)j \rfloor - \lfloor (p/q)(j-1) \rfloor$ and $c_{q-j+1} = \lfloor (p/q)(q-j-1) \rfloor - \lfloor (p/q)(q-j) \rfloor = p + \lfloor -(p/q)(j-1) \rfloor - (p + \lfloor -(p/q)j \rfloor)$. As long as x is not an integer we have $\lfloor -x \rfloor = - \lfloor x \rfloor -1$. But $2 <=j<=q-1$ indicates that neither $(p/q)(j-1)$ nor $(p/q)j$ are integers, hence

$$c_{q-j+1} = -\lfloor (p/q)(j-1) \rfloor - 1 - (\lfloor (p/q)j \rfloor - 1) = c_j \ .$$

Due to the above characterization of c_j^* we have

$$c_j^* = c_{q-j+1} \quad , \quad 1 <= j <= q$$

which shows that the digital line of length N corresponding to $y = (p/q)x + (q-1)/q$ has smallest period q, and hence the same is true for the original line. ■

Proposition 10 and its proof gives us a way to compute the number L(N,q) of digital lines of length N and smallest period q. In fact L(N,1) = 1, so we can consider q>1, then the situation N-s<q can only arise if N<=q+s-1<=2q-2, that is, (N+2)/2 <= q. Hence, if q < (N+2)/2, s can take arbitrary values and it follows that

(27) $L(N,q) = q\varphi(q)$ for $2 <= q < (N+2)/2$

where $\varphi(q)$ is the Euler function that counts the number of values p, 1<=p<=q, p q=1. This formula is clearly valid for q=1 since $\varphi(1)=1$. In the remaining range of q we can use that when p runs over all the values considered in $\varphi(q)$, so does ℓ, where we remind the reader ℓ is defined by (4). We fix ℓ and divide the range of s into two classes

$$0 <= s <= N-q, \ N-q+1 <= s <= q-1$$

The second class is not empty since we are assuming N+2<=2q. In the first class every line has smallest period q, this accounts for N-q+1 lines. In the second class we have two subclasses, $s+\ell < q$ and $q <= s+\ell$. The first one cannot introduce any lines of period q due to the condition $q^- > 0$. In the second one we have to consider whether

$$N-(s+\ell-q) >= q$$

or not. Only if this inequality is true we get new lines (due to the condition $q^+ > 0$). Hence we must have

$$\max\{q-\ell,\ N-q+1\} <= s <= \min\{q-1,\ N-\ell\}$$

which gives us $1+\min\{\ell-1,\ N-q,\ 2q-N-2,\ q-\ell-1\}$ lines (notice that this minimum is non-negative). Therefore, in this range of values of q we have

$$L(N,q) = (N-q+2)\ \varphi(q) + \sum_{\ell}\min\{2q-N-2,\ q-\ell-1,\ \ell-1,\ N-q\} \tag{28}$$

where the sum takes place over all values ℓ, $1 <= \ell <= q-1$, $\ell \wedge q = 1$. Since this expression is a little bit hard to work with, we can use upper and lower estimates, $L^{**}(N,q) = q\ \varphi(q)$, $L_*(N,q) = (N-q+2)\varphi(q)$ for q in this range. Finally, setting L(N) = total number of digital lines of length N, we get the estimates

$$L_*(N) = \sum_{q=1}^{\lceil N/2 \rceil} q\varphi(q) + \sum_{(N/2)+1=q}^{N} (N-q+2)\varphi(q) \tag{29}$$

$$<= L(N) <= L^{**}(N) = \sum_{q=1}^{N} q\,\varphi(q)$$

Using the above formulas we can produce the following table for N = 10

TABLE 7.1

q	(q)	L (N,q)	L(N,q)	L (N,q)
1	1	1	1	1
2	1	2	2	2
3	2	6	6	6
4	2	8	8	8
5	4	20	20	20
6	2	12	12	12
7	6	30	36	42
8	4	16	20	32
9	6	18	22	54
10	4	8	8	40
TOTAL:		21	135	217

We notice that L(N) is fairly close to $L_*(N)$ and very different from $L^{**}(N)$. $L^{**}(N)$ would have been the count if no digital lines drop their period when considered to have finite length. Since we want to develop some asymptotic bounds for the error of the choice (21) for subpixel accuracy we introduce a different upper bound function $L^*(N,q)$ defined as follows:

$$
\begin{aligned}
(30)\qquad & L^*(N,q) = L(N,q) \quad 1 <= q <= \lceil(N/2)\rceil \\
& L^*(N,q) = L_*(N,q) + (2q-N-2)(\varphi(q)-2), \quad (N/2)+1 <= q <= (2/3)N + 2/3 \\
& L^*(N,q) = L_*(N,q) + (N-q)(\varphi(q)-2), \quad (2/3)N+2/3 < q <= N
\end{aligned}
$$

The choice is motivated by choosing the smallest of the two terms independent of ℓ in the minimum that appears in (28). Since the values $\ell = 1$, $\ell = q-1$ make this minimum zero we only have $(\varphi(q)-2)$ terms in the sum. We also note that $L^*(N,q) = L_*(N,q)$ for $q = (N/2) + 1$ (if this value is an integer) and for $q = N$. For $N = 10$, we have only three values to compute

$$L^*(N,7) = 38,\ L^*(N,8) = 20,\ L^*(N,9) = 22$$

which gives $L^*(10) = 137$ in this case, a very good approximation!

(We have used $L^{*}(N) = \sum_{q=1}^{N} L^{*}(N,q)$).

PROPOSITION 11: The exact number of digital lines is given by

the formula $L(N) = \sum_{q=1}^{n} L(N,q)$, $L(N,q)$ defined by (27) and (28).

It satisfies the inequalities

$$L_{*}(N) <= L(N) <= L^{*}(N),$$

where the functions $L_{*}(N)$, $L^{*}(N)$ have been defined above and satisfy the asymptotic estimates:

(31) $\quad L_{*}(N) = (3/4\pi^{2})N^{3} + O(N^{2}\log N) \quad 0.076N^{3}$

(32) $\quad L^{*}(N) = (10/9\pi^{2})N^{3} + O(N^{2}\log N) \quad 0.112N^{3}$

PROOF: We only have to prove the estimates (31) and (32). We use the methods used in [H-W] to prove the following asymptotic formula:

(33) $\quad \Phi(N) = \sum_{1}^{N} \phi(q) = 3N^{2}/\pi^{3} + O(N\log N).$

The idea is to write using the Moebius function μ

(34) $\quad \phi(q) = q \sum_{d|q} \mu(d)/d.$

It will be useful to find first the asymptotics of $L^{**}(N)$.

For any N,

$$L^{**}(N) = \sum_{q=1}^{N} q\phi(q) = \sum_{q=1}^{N} q^{2} \sum_{d|q} \mu(d)/d$$

We now write $q = dd'$ and substitute in the last term:

$$L^{**}(N) = \sum_{dd'<=N} d^{2}(d')^{2} \mu(d)/d$$

$$= \sum_{d=1}^{N} d\mu(d) \sum_{d'<=N/d} (d')^{2}$$

The term $\sum_{d' <= N/d} (d')^2 = (1/3)(N/d)^3 + O(N^2/d^2)$. Inserting this in $L^{**}(N)$, we obtain

$$L^{**}(N) = (1/3)N^3 \sum_{d=1}^{N} \mu(d)/d^2 + O(N^2 \log N).$$

Note we have used

$$\left| \sum_{d=1}^{N} d\mu(d)N^2/d^2 \right| <= N^2 \sum_{d=1}^{N} 1/d = O(N^2 \log N)$$

But we have $\sum_1^{\infty} \mu(d)/d^2 = 6/\pi^2$ [H-W]. Hence, $\sum_1^N \mu(d)/d^2 = 6/\pi^2 + O(1/N)$. Substituting this into $L^{**}(N)$, we get

(35) $\quad L^{**}(N) = 2N^3/\pi^2 + O(N^2 \log N)$.

We can now get the asymptotic formula for $L_*(N)$. Recall that we have, from (29),

$$L_*(N) = N^3/(4\pi^2) + O(N^2 \log N)$$

$$+ (N + 2) \sum_{(N/2)+1}^{N} \phi(q) - \sum_{(N/2)+1}^{N} q\phi(q)$$

We can write $\sum_{(N/2)+1}^{N} \phi(q) = \Phi(N) - \Phi(N/2)$

$$= 3N^2/\pi^2 - 3N^2/4\pi^2 + O(N\log N)$$

$$= 9N^2/(4\pi^2) + O(N\log N)$$

Similarly $\sum_{(N/2)+1}^{N} q\,\phi(q) = L^{**}(N) - L^{**}(N/2)$

$$= 2N^3/\pi^2 - (2/\pi^2)(N/2)^3 + O(N^2 \log N)$$

$$= 7N^3/(4\pi^2) + O(N^2 \log N)$$

So that we finally get $L_*(N) = 3N^3/(4\pi^2) + O(N^2 \log N)$.

Using the definition (30) we obtain:

$$L^*(N) = \sum_{1}^{(2/3)N+2/3} q\phi(q) - 2 \sum_{(N/2)+1}^{(2/3)N+2/3} (2q-N-2)$$

$$+ (2N+2) \sum_{(2/3)N+2/3}^{N} \phi(q) - 2 \sum_{(2/3)N+2/3}^{N} q\phi(q)$$

$$- 2 \sum_{(2/3)N+2/3}^{N} (N-q)$$

$$= L^{**}((2/3)N+2/3) + 2N(\Phi(N) - \Phi((2/3)N+2/3))$$

$$- 2(L^{**}(N) - L^{**}((2/3)N+2/3)) + 0(N^2)$$

We introduce now (33) and (34) into this expression:

$$L^{*}(N) = (6/\pi^2)((2/3)N)^3 - 4N^3/\pi^2 + 6N^3/\pi^2$$

$$-(6N/\pi^2)((2/3)N)^2 + 0(N^2 \log N)$$

$$= (10/(9\pi^2))N^3 + 0(N^2 \log N)$$

We note that $L_{*}(N) \simeq 0.076N$ and $L^{*}(N) \simeq 0.112N$ if we disregard the $0(N^2 \log N)$ term, for N = 10 these approximations are not very good. Nevertheless for the coming estimates it is only the leading term that counts.

Remark: On purely heuristic grounds one can propose an approximate formula L(N) to the correct value L(N). It consists in assuming that the values that appear in (28) are uniformly distributed with density $\phi(q)/q$. Then

$$\tilde{L}(N) = L_{*}(N) + \sum_{(N/2)+1}^{(2/3)N+2/3} (\phi(q)/q)(2q-N-2)(N-q)$$

$$+ \sum_{(2N/3)+1}^{N} (\phi(q)/q)(N-q)(2q-N)$$

It is clear that $L_{*}(N) <= \tilde{L}(N)$ and also $L(N) \lesssim L^{*}(N)$, since N-q<q and 2q-N<q in both sums. It is not apparent how to find the correct relation between L(N) and $\tilde{L}(N)$ but we note that for N = 10 we immediately get from Table 7.1 the remarkable value

$\tilde{L}(10) = 135.47$

Besides, one can show, by the same methods used in Proposition 11, that the following asymptotic development holds

$$L(N) = N^3/\pi^2 + O(N^2 \log N)$$

which fits right between the values in Proposition 11. It is tempting to conjecture that L(N) has the same asymptotic behavior. In fact, we computed L(N), using (27) and (28), and $\tilde{L}(N)$ for N = 100 and found the following values

$$L(N) = 104,359$$
$$\tilde{L}(N) = 104,949$$
$$L(N)/N^3 = 0.104359$$
$$1/\pi^2 = 0.101321$$

which clearly reinforces the conjecture.

Let S(N) be given by

$$(36) \qquad S(N) = \sum_{q=1}^{N} (1/q)L(N,q)$$

Then the offset error incurred by using the line parallel to B passing through the middle of the channel is given by

$$(37) \quad E(N) = ((1/2)S(N))/L(N)$$

when we use the uniform distribution on digital lines.

PROPOSITION 12: Up to terms of the form $O((\log N)/N^2)$ the offset error defined in (37) satisfies the estimates

$$(38) \quad (29/40)(1/N) <= E(N) <= (59/54)(1/N)$$

PROOF: We start with the lower bound for E(N). It is clear that E(N) >= 1/(2N). To improve on this we note that up to q = N/2 the sum of the terms in S(N) is exactly $\Phi(N/2)$, hence

$$2E(N) = \frac{\Phi(N/2) + \sum_{(N/2)+1}^{N} (1/q)L(N,q)}{L^{**}(N/2) + (L(N) - L^{**}(N/2))} >=$$

$$\frac{1}{N}\,\frac{N\Phi(N/2) + (L(N) - L^{**}(N/2))}{L^{**}(N/2) + (L(N) - L^{**}(N/2))}$$

Now $N\Phi(N/2) > L^{**}(N/2)$ because in $S(N)$ we divide by q, and here are considering $1 <= q <= N/2$ only. It is easy to see that the function $(a+x)/(b+x)$ is strictly decreasing if $a > b$, hence the above expression diminishes if we replace $L(N)$ by $L^{*}(N)$ and we obtain

$$2E(N) >= \frac{1}{N}\,\frac{N\Phi(N/2) + (L^{*}(N) - L^{**}(N/2))}{L^{**}(N/2) + (L^{*}(N) - L^{**}(N/2))}$$

$$= \frac{\Phi(N/2) + (L^{*}(N) - L^{**}(N/2))/N}{L^{*}(N)}$$

$$= \frac{(3/\pi^{2})(N^{2}/4) + (((10/9)N^{3})/\pi^{2} - (2/\pi^{2})(N^{3}/8))/N}{((10/9)N^{3})/\pi^{2}} + 0(\log N/N^{2})$$

$$= (29/20)(1/N) + 0(\log N/N^{2})$$

Therefore we get

$$E(N) >= (29/40)(1/N) + 0(\log N)/N^{2}.$$

Let us now work an upper bound for $E(N)$. We use a slightly more complicated method. Replacing $L(N,q)$ by $L^{*}(N,q)$ in the expression of $S(N)$ we have

$$S^{*}(N) = \sum_{1}^{(2/3)N+2/3} \phi(q) - 2 \sum_{(2/3)N+2/3}^{N} \phi(q)$$

$$+ 2N \sum_{(2/3)N+2/3}^{N} (\phi(q)/q) + 0(N)$$

The only new difficulty consists in estimating the term $\Sigma(\phi(q)/q)$.

Using the formula (34) we get

$$\sum_{(2/3)N+2/3}^{N} (\phi(q)/q) = \sum_{q} \sum_{d|q} \mu(d)/d$$

$$= \sum_{d=1}^{N} (\mu(d)/d)((1/3)(N/d) + O(1/d))$$

since by writing $q = dd'$, we get $((2/3)N + 2/3)/d <= d' <= N/d$. By the same argument we used to obtain (35) we see that this term is exactly

$$(N/3)(6/\pi^2) + O(1)$$

The first two terms in $S^*(N)$ can be computed using (33), and we finally get

$$S^*(N) = 2N^2/\pi^2 + O(N)$$

On the other hand,

$$S(N) + (L^*(N) - L(N))/N <= S^*(N)$$

Dividing by L^* (N) we obtain (up to $O(\log N/N^2)$)

$$(L(N)/L^*(N))((S(N)/L(N))-1/N) + 1/N \quad (18/10)(1/N)$$

But $\quad L(N)/L^*(N) >= L_*(N)/L^*(N) \simeq 27/40$

hence $\quad 2E(N) - 1/N \lesssim (8/10)(40/27)(1/N)$

which leads to the estimate (38). ■

REMARK: Corresponding to the heuristic estimate $\tilde{L}(N)$ given above for the correct number of lines L(N) we can construct a heuristic formula for the asymptotic error, $\tilde{E}(N)$,

$$\tilde{E}(N) = 1/2\ \tilde{S}(N)/\tilde{L}(N),$$

where

$$\tilde{S}(N) = \sum_{1}^{N/2} \phi(q) + \sum_{N/2+1}^{N} \phi(q)((4N+2)/q - 3 - (N^2/q^2))$$

One finds, using the same type of reasoning as in Proposition 12,

$$\tilde{S}(N) = (6(1-\log 2)N^2)/\pi^2 + O(N\log N)$$

$$\tilde{E}(N) = (3(1-\log 2))/\pi^2\ (1/N) + O(\log N/N^2)$$

$$\approx 0.92/N$$

which is in fact in tune with the upper and lower bounds obtained in Proposition 12, namely 0.72/N and 1.09/N respectively. It would be very interesting to show that E(N) has the same asymptotic behavior as $\tilde{E}(N)$.

We remark that though the asymptotic behavior of the expected value of the offset with respect to the invariant measure μ is very hard to obtain due to the nature of the formulas from section 6, for any concrete value of N it is perfectly possible to compute this expected value using the explicit nature of the formulas for the measure $\mu(\Omega)$ of the quadrilateral associated to any digital line. We have done this for N =10 and obtained:

TABLE 7.2

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
0.0000	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 7.2 Error probabilities for digital lines without points missing

Section 8 Digital Line Fitting with Known Slope

The procedure for estimating the best real edge giving rise to a digitization is improved considerably if we make use of the fact that the slope of the underlying real line is known. In our algorithm from the first year, discussed in Sections 2-7, we used the point on the best edge estimate to match with a corresponding point on the reference edge. This unfortunately assumed the horizontal offset was known and the vertical estimate was to be estimated. By using intersecting lines we wanted to minimize the problem. In a new version of the algorithm the problem of needing to know the x_*-offset has been eliminated and the errors are considerably reduced.

The original algorithm uses the line midway between B and L where (A,B,C,D) are the vertices of the feasible quadrilateral corresponding to the digital line. This line will not in general have the correct slope so there is no way to map it into the reference edge by a translation. That is why we selected a point on the midline to use in interimage matching. Our new algorithm is identical to the old one except that we select the midline among all lines with the proper slopes. In terms of the feasible quadrilateral for the digital edge, since we know the slope, we know we are on a fixed horizontal line in y-intercept slope space. Thus we can restrict our attention to the intersection of this horizontal line with the feasible quadrilateral and select the midpoint of this line segment as our best edge in image space. We can now map our best edge estimate to the edge in the reference image, since the two lines have the same slope. Since we are mapping the entire estimated line and not a single point on it, we are not using subpixel information about the offset in the x-direction. The worst-case and expected error bounds previously derived provide bounds, which in general are unduly pessimistic. Geometrically, one can see this as follows. The worst case error in the previous algorithm is half the vertical distance between lines B and C. For a real edge with the same slope as B and C, this maximum error can be realized in the new algorithm. For any other slope, the maximum error is half the vertical distance between the uppermost and lowermost line of that slope lying in the channel formed by A,B,C, and D. But this maximum separation decreases monotonically as the slope moves away from that of B.

The above result on the decrease in error can be easily given analytically. Assume the line B has slope $m_1 = p/q$ and the real edge has slope $m_2 > m_1$. The proof for $m_2 < m_1$ is simpler. Let the intersection of A and B be (x_1 , y_1) and let the intersection of C and D by (x_2 , y_2). Then (x_1 , y_1) is the left end point of the bottom of the channel and (x_2 , y_2) is the right end point of the top. Since $m_2 > m_1$, the lowermost line with slope m_2 and going through the channel passes through (x_1 , y_1). Similarly, the uppermost such line passes through (x_2 , y_2). Let B have y-intercept b. Then B has the equation

$$y = m_1x + b$$

Since (x_1 , y_1) is on B, we see that $h = y_1 - m_1x_1$. The vertical

separation between B and C is 1/q so C has the equation

$$\begin{aligned} y &= m_1x + b + 1/q \\ &= m_1x + y_1 - m_1x_1 + 1/q. \end{aligned}$$

The lowermost line with slope m_1 has the equation

$$y = m_2x + y_1 - m_2x_1$$

since it has slope m_2 and is constrained to pass through (x_1 , y_1). In a similar fashion, the uppermost line with slope m_2 has the equation

$$y = m_2x + y_2 - m_2x_2.$$

The difference in y intercepts between the two lines with slope m is:

$$\begin{aligned} h_1 &\equiv y_2 - m_2x_2 - (y_1 - m_1x_1) \\ &= m_1 x_2 + y_1 - m_1x_1 + 1/q - m_2x_2 - y_1 + m_1 x_1 \\ &= 1/q + m_1(x_2 - x_1) + m_2(x_1 - x_2) \\ &= 1/q + (m_1 - m_2)(x_2 - x_1) \end{aligned}$$

Since $x_2 > x_1$ and $m_1 < m_2$, $(m_1 - m_2)(x_2 - x_1) < 0$. Hence $h_1 < h$ and we have shown that the maximum error goes down as $(m_1 - m_2)(x_2 - x_1)$. The quantities x_2 and x_1 can be calculated in terms of the vertices of the feasible quadrilateral.

Section 9. Expected Error for Vertical Offset Estimation Using Slope

This section provides further probablistic analysis of our procedure for estimating the position of real lines. This procedure made use of the known slope of the real line to restrict the set of feasible lines to lie on a horizontal line segment in the feasible quadrilateral. We now describe various error expectations associated with these procedures. First we examine the expected error given the digitization and real line slope. Next we integrate this error over the quadrilateral corresponding to the digital line to obtain the expected error over all real lines with the specified digitization. Finally this is summed over the entire image. If we view the horizontal offset as known, this gives the expected error in the vertical offset. If we do not know the horizontal offset, the computation gives the error in the linear relation between x and y. A second line is then required to solve for values of the x and y offset.

The expected error given the correct digital line and slope of the real line is a trivial computation. We have shown that the set of all feasible lines is a horizontal line segment in y-intercept, slope space and that our estimate of the real line position is the real line in the image space corresponding to the midpoint of the line segment. Since the invariant measure on lines, when restricted to a horizontal line in y-intercept, slope space, is the uniform measure on the line segment, the expected error is merely the expected distance of a point on the line segment from the center of the line segment. The computation of the expectation is a simple computation using elmentary calculus and the result is r/4 where r is the length of the line segment.

The next problem is to determine, for a given quadrilateral corresponding to a digital line, what is the expected error in vertical offset estimation. This is the expected error in offset estimation given that we use the known slope of the real line to restrict ourselves to the appropriate horizontal line segment. More precisely, let y_o , y_1 denote the y-coordinate of the lowermost and uppermost vertices of a feasible quadrilateral and let h(y) denote the width of the feasible quadrilateral at height y. Then the expected error given that we are at height y is h(y)/4 and the expected error over the entire quadrilateral, Q, is

$$E(Q)=(1/k)\ {}_{y_o}^{y_1}(h(y)/(1+y^2)^{3/2})dy$$

where

$$k={}_{y_o}^{y_1}(1+y^2)^{-3/2}dy.$$

Values of E(Q) for the various quadrilaterals are given in Table 9.1, columns 1 and 2.

Next the expected error, E, over the whole image is computed by multiplying the error associated with each quadrilateral, Q, by the probability of Q occurring (using the invariant measure). The value was computed to be .056 pixels. Thus the average accuracy in vertical intercept estimation given the correct digital line is approximately 1/20 pixel.

Table 9.1. Expected errors vs Slope for Digital lines with zero and two missing pixels.

	Number pixels missing	
Slope	0	2
0.00000	0.00000	0.00000
0.01000	0.00161	0.00188
0.02000	0.00247	0.00326
0.03000	0.00282	0.00425
0.04000	0.00265	0.00494
0.05000	0.00244	0.00560
0.06000	0.00188	0.00624
0.07000	0.00181	0.00716
0.08000	0.00147	0.00840
0.09000	0.00222	0.01047
0.10000	0.00222	0.01190
0.11000	0.00281	0.01339
0.12000	0.00258	0.01444
0.13000	0.00384	0.01599
0.14000	0.00321	0.01618
0.15000	0.00415	0.01660
0.16000	0.00461	0.01730
0.17000	0.00393	0.01756
0.18000	0.00448	0.01977
0.19000	0.00847	0.02497
0.20000	0.00496	0.02282
0.21000	0.00512	0.02262
0.22000	0.00467	0.02371
0.23000	0.00758	0.02395
0.24000	0.01280	0.02392
0.25000	0.00804	0.02504
0.26000	0.00500	0.02529
0.27000	0.00651	0.02556
0.28000	0.00613	0.02723
0.29000	0.00642	0.03034
0.30000	0.00593	0.02889
0.31000	0.01135	0.02734
0.32000	0.01877	0.02472
0.33000	0.01684	0.02651

0.34000	0.00981	0.02937
0.35000	0.00718	0.03146
0.36000	0.00633	0.03304
0.37000	0.00815	0.03376
0.38000	0.00801	0.03684
0.39000	0.01545	0.04415
0.40000	0.00797	0.03730
0.41000	0.00812	0.03576
0.42000	0.01043	0.03772
0.43000	0.00840	0.03977
0.44000	0.00786	0.03755
0.45000	0.00826	0.03590
0.46000	0.01114	0.03732
0.47000	0.01909	0.03996
0.48000	0.02859	0.04245
0.49000	0.04435	0.04435
0.50000	0.02909	0.04325
0.51000	0.01976	0.04143
0.52000	0.01173	0.03940
0.53000	0.00885	0.03850
0.54000	0.00857	0.04091
0.55000	0.00932	0.04410
0.56000	0.01179	0.04263
0.57000	0.00934	0.04111
0.58000	0.00935	0.04374
0.59000	0.01885	0.05277
0.60000	0.01018	0.04502
0.61000	0.00983	0.04186
0.62000	0.00807	0.04183
0.63000	0.00901	0.04075
0.64000	0.01270	0.03893
0.65000	0.02283	0.03590
0.66000	0.02605	0.03430
0.67000	0.01613	0.03892
0.68000	0.00865	0.04199
0.69000	0.00962	0.04532
0.70000	0.00943	0.04195
0.71000	0.01032	0.04049
0.72000	0.00816	0.04133
0.73000	0.01354	0.04216
0.74000	0.02229	0.04167
0.75000	0.01367	0.04324
0.76000	0.00917	0.04453
0.77000	0.01023	0.04416
0.78000	0.01010	0.04612
0.79000	0.01870	0.05317
0.80000	0.01025	0.04422
0.81000	0.00874	0.04074
0.82000	0.01134	0.04258
0.83000	0.01104	0.04315
0.84000	0.00901	0.04484
0.85000	0.01158	0.04749
0.86000	0.00825	0.04599
0.87000	0.00971	0.04588

0.88000	0.00838	0.04490
0.89000	0.00927	0.04370
0.90000	0.00684	0.03918
0.91000	0.00957	0.03792
0.92000	0.01155	0.03836
0.93000	0.01789	0.04105
0.94000	0.02418	0.04498
0.95000	0.03403	0.05119
0.96000	0.04451	0.05874
0.97000	0.05776	0.06730
0.98000	0.07230	0.07740
0.99000	0.08883	0.08883
	-------	-------
Total <error>	0.04291	0.11799

Section 10 Expected Errors with Pixels Incorrect

Much of the theoretical analysis in the present study deals with the problem of edge location when a digital edge is known, but the position of the underlying real edge is unknown. In this section we consider the analysis of edge location estimation in the presence of incorrect pixels in the digital edge. We have been unable to derive general formulas for the effects of these errors so sampling was required to develop an expected error. Our expectation calculation assumes at most two pixels are incorrect in our estimated digital edge where the length of the edge is ten pixels. At the time these computations were performed, we weren't certain how accurately we could find the digital line, but computational considerations made the examination of additional incorrect pixels rather expensive.

The expected error for this phase of our study was defined as follows. Assume a given digital edge has been decided to be the digital edge corresponding to a real edge. Since we know the slope of the underlying real edge, this constrains the position of the underlying edge significantly. In y-intercept, slope space the point corresponding to the real edge must lie on a horizontal line whose height is the known slope of the line. We consider as feasible digital edges, those which intersect this line and differ in no more than two pixels from the correct quadrilateral. An expectation is now taken over these digital edges.

We now describe the above ideas more formally and define the expected error. Assume a quadrilateral Q_1 corresponds to the computed digital line L. Let the underlying real line in the reference image have slope α. Let $Q_1, \ldots, Q_K$ denote the set of all quadrilaterals such that the digital line corresponding to each Q_i, $i=2,\ldots,k$, differs from the digital line corresponding to Q_1 by no more than two pixels and such that the line $y=\alpha$ intersects Q_i. The set

$$S = \bigcup_{i=1}^{K} Q_i \cap L$$

is connected. Hence S is a line segment. Relabel the subscripts on the Q 's so that $Q \cap L$ is to the left of $Q_j \cap L$ if $i<j$. Let i_0 denote the index of the computed digital line. The estimated position of the real line is the midpoint, (x_0, α) of $Q_i \cap L$. The error in y-intercept location if the real line was some other point (x_1, α) on S is just $|x_0 - x_1|$.

The expected error is obtained by multiplying $|x-x_o|$ by its probability and integrating over S. Since the invariant measure on lines in x-y space is uniform on horizontal lines in y-intercept, slope space, we need only integrate $|x-x_o|$ with respect to the Euclidean measure and divide by the length of S. This calculation was done for each of 100 slopes between 0 and 1. The expected error was 0.12 pixels. Results are given in Table 9.1, columns 1 and 3.

Section 11 One-dimensional Edge Projections

The fitting of a continous edge to a digital image is a promising approach to subpixel edge location. In [Ha], discrete orthogonal polynomials are used to fit a continous surface to an image and the vanishing of the second directional derivative is used to locate edge points. Under the assumptions of our current research, we are using a high resolution reference image with a straight line of known position. In addition, it is assumed that no rotation is present between sensed and reference images. This constrains the fitting problem considerably.

Two basic types of fitting approaches can be applied to this problem. First, we can fit a two-dimensional surface to the image near the edge and then find the best fit of a straight edge with known slope to the surface. This problem is complicated by the unusual shape of the region near a straight edge in a digital image. Discrete orthogonal polynomial are most easily fit to rectangular regions in which the sides of the rectangle are horizontal or vertical. Square patches could be fit to neighborhoods of pieces of the edge neighborhood and edge points extracted, but the computational costs would be much higher than in the second fitting approach, the one-dimensional fit.

The one-dimensional fitting which we adopted for experimentation uses heavily the fact that we know both the slope of the edge and its approximate position. The fitting procedure is quite simple, but the present computation has not been optimized. Pixel centers for all pixels near the edge are projected onto a line perpendicular to the known di.ection of the edge. To each projected center we associate the grey level of the pixel. Thus we end up with a possibly multi-valued function on a finite subset of a line. We now fit an edge projection to the line. By performing the above operations on a number of digital edges without noise, we observed that the one-dimensional projection of an ideal digital edge could be represented by a continous curve consisting of two horizontal line segments connected by a slanting line segment.

To perform the fit on the projected points we used the mean grey levels in the regions above and below the edge to estimate the height of the two horizontal segment, for the fitted 1-d edge. The only remaining quantities to be estimated are the horizontal coordinates of the ends of the slanted segment. We are presently using exhaustive search to compute the best, in the sense of least squares, fit of the three segment piecewise linear edge to the projected data. Since the only variation in the fitting of the horizontal pieces results from the varying end-points, the square errors for the two horizontal segments, separately with all possible endpoints are computed prior to computing the total error of a fitted piecewise linear edge. This greatly facilitates the computation since the square error obtained in using a piece of the horizontal line segment to represent the data is obtained by a simple updating of the corresponding calculation for shorter pieces.

We now describe the algorithm more precisely, discuss theoretical considerations affecting the performance of the algorithm and give experimental results. We assume an initial esimate of the translation offset is known to within about a pixel. In particular we assume we have a real translation from the high resolution reference plane to the image plane. Thus the transformation can be used to map the reference line into a real line in the image plane. We assume that each point on this mapped line lies within about a pixel of the corresponding current subpixel location on the image plane.

The mapped real line segment is now digitized. We differ from our previous definition of digitization only for this section, and assume the digitization consists of all pixels which the line intersects. A neighborhood of the digitization is now grown. We defined our neighborhood to consist of all pixels lying on the assumed digitization or which were 8-neighbors of such a pixel. Let (x_i, y_i), $i=1,\ldots,N$ denote the coordinates of the centers of these pixels, and let α denote the slope of the reference line. Let L denote a line with slope $\beta = 1/\alpha$ and going through the origin. Then the perpendicular projections of the (x_i, y_i) on L are given by $S \equiv \{x_i \sin\gamma + y_i \cos\gamma\}$, $i=1,\ldots N$, $\gamma = \arctan\alpha$. We are only specifying the coordinate along L. Note that the points of S are not, in general unique, as can easily be seen if the original line is horizontal and L is vertical. Let $a_1,\ldots,a_N$ denote the points of S in non-descending order (duplications are allowed).

We now describe the fitting of the piecewise linear curve to the data. Let m_1 denote the mean grey level on the side of the edge corresponding to the smaller a_i's and let m_2 denote the mean grey level on the side of the edge corresponding to the larger a_i's. One approach to the estimation of these means is to begin at the midpoint of the mapped real edge and move several (5 or 6) pixels from this point in each direction away from the mapped real edge and use these pixels as the centers for small windows used for estimating the region means. Due to an experimental set up, we avoided this issue but many approaches are available. The regions abutting an edge in the reference image can be roughly outlined when the reference edge is first delineated. Alternatively, a window in each abutting region which is not close to the region boundary can be outlined. The mapped versions of these windows in the sensed image can then be used to estimate the means. Note that we are not comparing the grey levels in the sensed image with those in the reference image.

Once the region means m_1 and m_2 have been computed, we can compute the best edge fit. For each 1 <= i, j<= a, with i < j we computed the merit of the fit obtained using a_i and a_j as the endpoints of the middle segment of the fit. For each (i,j) as above we define the error e(i,j) associated with (i,j) by:

$$e(i,j): \quad \sum_{p=i}^{i} (g(a_p)-m_1)^2 + \sum_{p=i}^{j} (g(a_p)-(ma_p+b))^2$$

$$+ \quad \sum_{p=j+1}^{N} (g(a_p)-m_2)^2$$

where

$$m = (g(a_j) - g(a_i))/(a_j - a_i)$$

and $$b = - ma_i + g(a_i)$$

The middle of the three summands represents the quality of fit for the slanted segment. We define the optimal piece-wise linear fit to be given by these following segments (where $(\hat{i}, \hat{j})$ minimum e(i,j)):

1) the segment from (a_1, m_1) to $(a_{\hat{i}}, m_1)$

2) the segment from $(a_{\hat{i}}, m_1)$ to $(a_{\hat{j}}, m_2)$

3) the segment from $(a_{\hat{j}}, m_2)$ to (a_N, m_2)

We are currently using $a_{\hat{i}} + a_{\hat{j}}$ as the estimate of the intersection of the real edge with L. This estimate provides an estimate of the translation offset between the sensed and reference image in the direction along L. A second estimate in another direction (preferably perpendicular to L) is necessary to obtain an offset estimate for the x and y translations. The procedure for and analysis for combining, such estimates into a single estimate is described in Section 13.

Several factors offset the accuracy of the estimate obtained using the above procedure. First, we must consider the distribution of the points a_i on the line L. The fact that the line segments begin and end on a_i 's presents a limitation on the accuracy attainable using this procedure. The a_i 's are of the form $x_i \sin\gamma + y_i \cos\gamma$ where (x_i , y_i) is the center of a pixel near the edges. For simplicity, assume the real edge goes through the origin. Then, in order for the point (x_i , y_i) to be within a pixel of the edge we must have something like $|y_i - \alpha x_i| < 2$. The exact inequalities appear rather complicated but the above approximation is based on the fact that if we look at the vertical separation between the edge and the point $(x_i ,$ $y_j)$, the vertical separation must be less than 2 if $(x_i ,$

y_i) is to be adjacent to an edge pixel. Points (x_i , y_i) can actually be further away vertically and still be close to the line since we are really interested in the perpendicular distance to the line. Thus the above approximation becomes more accurate, the closer α is to zero. We can approximate the y_i by $|\alpha x_i + k|$ where -2 <= k <= 2. This gives us the following expression for the projection points:

$$x_i \sin\gamma + y_i \ |\alpha x_i + k| \cos\gamma$$

where -2 <= k <= 2 and 0 <= x <= h$\cos\gamma$ where h is the length of the edge. While this gives an explicit expression for the projections of pixels close to an edge, we have not been able to analyze the projections, even under this more restrictive model. Number theoretic results related to integer linear combinations of irrational numbers offer some promise of shedding more light on this problem. An alternative approach is to compute the projection points for a large number of angles and lengths.

The exact manner in which the distribution of gap between a_i 's would be used is not entirely clear though error bounds can be readily estimated. If we know that the maximum separation between any two consecutive a_i's is d, and we further assume that the algorithm is accurate to the nearest a_i then the maximum error due to the spacing of the a_i's is d.

One approach to improving the above algorithm might be to better model the form used to fit the projection points. The model consisting of two horizontal segments separated by a slanted segment was based on the observation that this behavior occured in projecting an ideal edge. Two types of refinements could be explored. First, for a given pair of mean values for the two abutting regions and for the given slope, one could compute the best fitting piecewise linear segment for an ideal image with homogenious regions using the two means. In this situation, only the y-intercept of the real line could be varied and the corresponding variation in the slope of the slanted line segment could be recorded. Only slopes in that range would then be used in fitting to the noisy image data. The best fit, in the least squares sense, as currently done in the algorithm may be better than the best least square fit in this new approach but it would then represent an impossible edge digitization. Thus the proposed procedure could both reduce computation by reducing the number of middle-section end points pairs examined and increase the accuracy of the procedure.

A second possible refinement to the current algorithm is to find the best estimate for the intersection of the real edge and L given the fitting projection edge. We selected the mid-point of the mid section for computational simplicity, but it may not be optimal for all combinations of slopes and mean grey levels. Both refinements could be investigated by extensive sampling but theoretical computations would be preferable.

The effect of noise on the above algorithm has not been modeled. Since the effect of noise seems to be strongly coupled with the geometry of projections, the problem appears to be quite difficult. Removal of outlying grey values may improve the signal to noise ratio but disturb the geometry.

Experimentation was performed using the above 1-d edge fitting method. As the initial results were promising, more extensive experimentation is planned. In order to perform the above experiments, it was necessary to have an image in which the position of an underlying real edge was known to very high precision. Two windows, an 8x8, and a 4x4, were selected from two agrigcultural fields in a Landsat image and each was repeated to provide two 32x32 windows, each representing a different type of field. A procedure was developed to splice the two images to form a third image with an edge whose position is known to very high accuracy. The procedure accepts as input a real line with slope between 0 and 1 which hits opposite sides of the 32x32 window. All pixels lying entirely below the edge are taken from the corresponding positions in window 1. All pixels lying entirely above the real line are taken from the corresponding position in window 2. Each pixel intersecting the real line is given a weighted average of the corresponding pixels in the first two windows. The weights are simply the areas of the parts of the pixel lying above and below the real edge. A real edge length of 10 was used. Fourteen angles at equal increments between 1 and 25 degrees were used. Table 11.1 gives the magnitude of the error for each line. The average error is .30 pixels.

Table 11.1 Errors in one-dimensional line fitting. Lines are at angles varying between 1 and 25 degrees in equal increments.

Line #	Error
1	.065
2	.185
3	.239
4	.045
5	.125
6	.215
7	.365
8	.360
9	.360
10	.340
11	.410
12	.171
13	.370
14	.324

Section 12 Fitting a Digital Edge to an Image

This section describes a grey level generalization of our digital edge fitting procedure. From the beginning of our work in the area of subpixel accuracy, it was felt that grey level information should ultimately be used in the locating of edges to subpixel accuracy rather than merely using the grey levels to get the digital line and then using the geometric methods. The analysis of the grey level approach appeared formidable, so we restricted ourselves initially to the investigation of the geometric methods. In Section 11 we studied one means of incorporating grey-level information, namely by projecting grey levels onto a real line perpendicular to the known edge direction. A piecewise linear ideal edge was then fit to the data and the offset in a direction perpendicular to the edge was estimated. We now describe several subpixel edge procedure directly using the two-dimensional image.

The basic idea of our first fitting procedure is to generate ideal two-dimensional edge images based on digitizing various real edges and find the one which best fits the sensed image. The means for the areas above and below the edge in the sensed image are first computed using the techniques outlined in Section 11 . Next the approximate pixel location of the edge is determined. An estimate of the correct pixel from a digital registration procedure is assumed available. Without loss of generality, we may assume the lower left-hand corner of the pixel has coordinates (0,0). A real line with the correct slope and a y-intercept of 0.5 is used to generate an edge image. Grey levels for all pixels intersecting this edge are computed. The grey level for an edge pixel is defined to be the weighted average of the average of the grey levels for the regions above and below the edge in the sensed image. As usual, the weights are the areas in the mixed pixel above and below the real edge.

The algorithm compares the generated digitization of an edge with the corresponding pixels in the sensed image. The generated edge is in the same coordinate system as the sensed image so it is meaningful to compare corresponding pixels. For each pixel, we compute the difference between the sensed and generated grey level. The sum of these differences is used to locate the edge. If the real edge is correct then we expect this sum to be close to zero. In general, the sign of the sum can be used to guide the search. From the sensed image we know whether the lower or upper region has a higher average grey level. Without loss of generality, we may assume the upper region has a higher average grey level. If the sum is a large positive number then the mixed pixels are producing, on the average, too low a grey level. Thus the estimated real edge position should be shifted down. Similarly, with a large negative value for the sum, the estimated real edge should be shifted up. This procedure is carried out in increments of a pixel until the sum changes sign. Upon termination we have a refined estimate for the pixel location of the edge.

The next phase of the algorithm attempts to locate the edge to subpixel accuracy. A new real edge is generated with an intercept which is the average of the current intercept and the nearest previous intercept in the direction indicated by the sign of the merit sum. As in the pixel level edge location method, the sum of differences is computed for corresponding pixels and the search is terminated when the possible change in y-intercept is less than a specified tolerance.

The above procedure was carried out on simulated imagery formed from LANDSAT data as described in Section 11 . Real lines of slope ranging from 0.0 to 1.0 in increments of 0.01 were used. The results are given in Table 12.1 . Of the 100 slopes tested, only 5 were worse than 0.2 pixels. Seventy-seven percent of the cases were 0.1 pixels or better.

A second procedure was based on the idea of fitting a digital edge to an image. Note that in the previous procedure, the digitization of an edge, not a digitial edge was used. Recall that the digitization of an edge contains all pixels intersecting the edge while the digital edge corresponding to a real edge contains the bottommost pixel in each column of the digitization of the edge. In the second procedure, the set of all digital edges which could be generated by an edge with the specified real edge slope were generated. For each such digital edge, the quality of fit of the digital edge to the image was computed and the digital edge with the best fit was selected.

The quality of fit measure is a non-negative real valued function which provides a rough measure of the edge quality of a set of pixels. Larger values indicate that the pixels are likely to lie on an edge. The merit m(S), of a set, S, of pixels is defined to be

$$m(S)= \qquad |u(P)-l(P)|*(1-\min(|(low+high)/2-g(P)|,1)$$

where the sum is over all pixels, P, in the set, u(P) and l(P) are the grey levels of the pixels immediately above and below P, g(P) is the grey level for the pixel P, and low and high are the means for the regions below and above the edge. This approach led to subpixel accuracy but the results were much poorer than for the digitization of the real edge. The results are given in Table 12.1 .

The procedure using the digitization of the edge, as opposed to the digital edge, has the advantage that it is extendible to region with curved boundaries, since the digitization scheme can be applied to any boundary. We are not pursuing this in the current study since the investigation of the straight-edge method still requires considerable investigation. We are beginning the study of probabilistic models for straight-edge error analysis using the methods of this section.

Several basic sources of error should be considered in our edge-fitting procedure. First, the performance of the method deteriorates with an increase in the noise in the abutting regions. This noise has two facets. It can result in inaccuracies in the calculation of the means for adjacent areas and it can result in poor fitting due to noise in the edge pixels themselves. Another source of error is quantization. If the two regions each had constant gray levels and the grey levels were not quantized, then it is easy to show the edge positions could be determined exactly. As soon as quantization is introduced, the results deteriorate, since shifts in the underlying edge position do not necessary result in shifts in the quantized grey levels. Analysis of this source of error is planned for future work.

The edge fitting methods discussed in this section represent an initial effort at subpixel edge estimation using a mixture of grey level information and digital geometry. In the very preliminary experiments performed in this section, it appears that a high level of subpixel registration accuracy may be possible using this basic approach. Considerable refinement of these methods is possible by refinements of the merit functions and search procedures.

Table 12.1. Comparison of two subpixel edge dectection algorithms. Although the number of incorrect pixels can be large, the real intercept difference can remain small. The Real Line Digitization proved to be the most accurate of the algorithms, where the directly estimated intercept is used.

Error1 = distance between estimated intercept and the correct intercept

Error2 = distance between average digital line intercept and the correct intercept

Pix = number of pixels generated that are not exactly on the edge generated by the underlying real line.

	Real Line Digitization			Digital Line Mask	
Slope	Error1	Pix	Error2	Pix	Error1
0.00000	0.09457	4	0.09500	10	0.61000
0.01000	0.05584	0	0.28000	3	0.17000
0.02000	0.07180	0	0.17000	3	0.25500
0.03000	0.05992	0	0.06000	10	0.38000
0.04000	0.05260	0	0.05000	10	0.32500
0.05000	0.04984	0	0.16000	10	0.19000

0.06000	0.00188	0	0.01500	10	0.27000
0.07000	0.00266	0	0.00000	10	0.38000
0.08000	0.00055	0	0.03500	10	0.49000
0.09000	0.02109	0	0.05000	10	0.45000
0.10000	0.00465	0	0.04500	10	0.29000
0.11000	0.00090	0	0.02000	10	0.30000
0.12000	0.00984	0	0.00500	2	0.13500
0.13000	0.01391	0	0.04000	10	0.25000
0.14000	0.03359	0	0.00000	10	0.62500
0.15000	0.03180	0	0.02000	10	0.50000
0.16000	0.01750	0	0.05500	10	0.29000
0.17000	0.03992	1	0.07000	0	0.03000
0.18000	0.04073	0	0.03000	10	0.11500
0.19000	0.00257	0	0.10000	10	0.20000
0.20000	0.03258	0	0.02000	10	0.41500
0.21000	0.00734	0	0.03000	10	0.53000
0.22000	0.02326	0	0.03000	1	0.14500
0.23000	0.03141	0	0.02000	2	0.26000
0.24000	0.04974	1	0.12500	2	0.25000
0.25000	0.07797	0	0.01000	0	0.01000
0.26000	0.02547	0	0.01000	10	0.37500
0.27000	0.01516	0	0.02000	10	0.50000
0.28000	0.01891	0	0.03500	10	0.69000
0.29000	0.00990	0	0.05000	1	0.05000
0.30000	0.02055	0	0.00500	2	0.13500
0.31000	0.01193	0	0.02000	10	0.22000
0.32000	0.04302	0	0.13000	10	0.30500
0.33000	0.03505	0	0.10000	10	0.58000
0.34000	0.03555	0	0.00000	4	0.35000
0.35000	0.02758	0	0.00000	5	0.46000
0.36000	0.02807	0	0.01500	4	0.38500
0.37000	0.00984	0	0.01000	10	0.49000
0.38000	0.03297	0	0.01500	10	0.40000
0.39000	0.02355	0	0.10000	10	0.40000
0.40000	0.09234	1	0.10500	10	0.62500
0.41000	0.01427	0	0.01000	3	0.33000
0.42000	0.03234	2	0.09000	6	0.51500
0.43000	0.00484	0	0.00000	10	0.46000
0.44000	0.00859	0	0.00000	1	0.10000
0.45000	0.00035	0	0.02000	2	0.18000
0.46000	0.10598	0	0.06500	10	0.55000
0.47000	0.00359	0	0.04000	4	0.34000
0.48000	0.03141	0	0.14500	4	0.42000
0.49000	0.01660	2	0.25000	3	0.25000
0.50000	0.04802	0	0.14500	10	0.36500
0.51000	0.04005	0	0.04000	10	0.48000
0.52000	0.04585	0	0.06500	10	0.59500
0.53000	0.03453	1	0.06000	10	0.71000
0.54000	0.03270	0	0.00000	5	0.45000
0.55000	0.04138	1	0.10000	6	0.54000
0.56000	0.02547	0	0.05500	10	0.63000
0.57000	0.05051	0	0.01000	3	0.25000
0.58000	0.01617	0	0.01000	3	0.28500
0.59000	0.03880	0	0.10000	3	0.40000

0.60000	0.00805	0	0.01500	6	0.60000
0.61000	0.04227	0	0.01000	1	0.13000
0.62000	0.03703	0	0.01500	10	0.43000
0.63000	0.00367	0	0.00000	10	0.36000
0.64000	0.04678	0	0.00000	0	0.00000
0.65000	0.00966	0	0.10000	0	0.10000
0.66000	0.01427	0	0.13000	4	0.20500
0.67000	0.02234	0	0.02000	3	0.18000
0.68000	0.00484	0	0.00500	10	0.31500
0.69000	0.00257	0	0.05000	10	0.45000
0.70000	0.02797	0	0.03500	10	0.58500
0.71000	0.05385	1	0.06000	10	0.70000
0.72000	0.07453	1	0.07000	3	0.26000
0.73000	0.03141	0	0.01000	4	0.38000
0.74000	0.01465	2	0.12500	6	0.50000
0.75000	0.08383	1	0.10000	10	0.38000
0.76000	0.04266	1	0.08500	10	0.43000
0.77000	0.06203	1	0.08000	10	0.53000
0.78000	0.07359	1	0.08500	10	0.66500
0.79000	0.12552	0	0.10000	10	0.80000
0.80000	0.08793	0	0.03000	5	0.50500
0.81000	0.10242	1	0.07000	10	0.61000
0.82000	0.13352	1	0.20500	10	0.71500
0.83000	0.12555	2	0.16000	10	0.82000
0.84000	0.08549	1	0.07500	10	0.85000
0.85000	0.09761	2	0.10000	10	0.96000
0.86000	0.11922	2	0.13500	9	1.07000
0.87000	0.11223	1	0.12000	10	0.50000
0.88000	0.08180	1	0.06500	10	0.62500
0.89000	0.10443	1	0.15000	10	0.75000
0.90000	0.10883	2	0.14500	10	0.87500
0.91000	0.12430	2	0.16000	10	0.92000
0.92000	0.10070	2	0.12500	9	1.05500
0.93000	0.13062	1	0.13000	10	1.19000
0.94000	0.11927	0	0.05000	10	1.27500
0.95000	0.10805	0	0.06000	10	0.62000
0.96000	0.10447	0	0.17000	10	0.71500
0.97000	0.09927	0	0.28000	10	0.81000
0.98000	0.11930	0	0.39000	10	0.90500
0.99000	0.04419	6	0.50000	10	1.00000

Table 12.2. Summary of subpixel edge detection algorithm errors.

	Real Line Digitization			Digital Line Mask	
	Error1	Pix	Error2	Pix	Error1
Maximums	0.13352	6	0.50000	10	1.27500
Averages	0.04741	0.46000	0.07570	7.52000	0.46450
St. Dev.	0.03832	0.92109	0.08155	3.50280	0.26417

Section 13 Pairs of Lines

The matching of a line in a reference image with a line in a sensed image only determines a linear relation between the x and y offsets for the sensed image. A second linear relation resulting from a matching of a second line between reference and sensed images can then be used to get an estimate for the x and y offsets. In this section we examine the offset estimation accuracy resulting from this approach.

We consider an image in which two perpendicular edges are used to estimate the offset between sensed and reference images. Let line L_1 in the images lead to a linear relationship $y=mx+b_1$ between the offsets. Note that this is not the equation of the real line L_1 but the equation relating the x and y offsets resulting from trying to locate L_1 in the reference image. If L_2 is perpendicular to L_1, then the corresponding relationship between the offsets for x and y using L_2 is given by $y+(l/m)x+b_2$. The knowledge of the correct digital lines for L_1 and L_2 give rise to error bounds on b_1 and b_2. Thus the correct offset relations are actually in the set

$$S_1 = \{(x, mx+b_1+h) | e < h < e\}$$

and

$$S_2 = \{(x, (1/m)x+b_2 +k) | e < k < e\} ,$$

where e_1, e_2, e_3, and e_4 are the error bounds on the linear relationship between the x and y offset estimates.

The set of feasible offset estimates is the intersection of the two infinite strips, S_1 and S_2. The intersection of these two strips is a quadrilateral in x-offset, y-offset space. In the event that the real world edges are perpendicular, the resulting quadrilateral in offset space will be a rectangle. The error in the x and y estimated offsets is a function of the angle between the image edges, the slope of the edge and the error bounds on the linear relationships between the x and y shifts resulting from the individual edge matchings. As an idea of the magnitude of the error, perpendicular edges with equal bounds, say r, on the error in the x-y offset linear relation estimation and with slopes 1 and -1 will have a maximum error of r 2. Keeping all parameters but the slopes fixed, the error increases as the slopes move away from 1 and -1. Intuitively, we are considering a square in offset space where the sides of the square are parallel to the edges in the image. The sides of the square represent the error in the linear relationship and the horizontal and vertical extent of the square give the variation in the possible correct x and y offsets.

More detailed error analysis has been carried out but this analysis is not directly useful until more extensive analysis of the digital edge fitting methods have been performed. This section has described a procedure for taking the bounds from individual edge matching and producing bounds on the offset estimation error resulting from a pair of matching edges.

Section 14 Geometric Registration Summary

The previous fourteen sections give an overview of our work on geometric methods in registration. In the latter sections, grey level information was directly incorporated into the edge location process. In this section, we attempt to put matters into perspective.

Subpixel edge position estimation can be used for registration and scene analysis. Strictly geometric methods based on the observation of the correct digital line can be quite accurate (averaging about 1/20 of a pixel). As the number of incorrect pixels is allowed to increase the estimation errors, of course, increase. The average error over all lines, given that the digitization has at most two incorrect pixels is .118. To make these figures useful, we must know how well we can find the correct digital line.

In the process of developing methods for finding the correct digital line, we came up with methods which generalized our original algorithms and directly estimated line positions. In limited experimentation, one of these methods resulted in an average error of .047 pixels with a standard deviation of .038. This approach appears quite promising though experimentation is in a very early stage. This work was done on grey level simulated images. We hope to extend the analytical study of geometric registration error to this procedure and perform more comprehensive experimental studies. If the algorithm continues to appear promising, we will examine various means to improve its efficiency and reliability.

Section 15 Random Fields and Subpixel Accuracy

In the previous report [La] the problem of subpixel translation-registration was posed in the context of sensed and reference random fields in the plane for which the correlation statistic (C(•) defined below) forms approximately a Gaussian random field. For such sensed and reference fields, a theoretical upper bound was found for the probability of local misregistration by τ pixels or more. In this section, we summarize briefly and specialize the most useful models and results from that previous report for further comparison with empirical results.

All our models are based on the assumption that a nonrandom reference field $Z_A(\underline{x})$ is specified at all lattice coordinates $\underline{x} = (x,y) \in h\mathbb{Z}^2$ (i.e., integer multiples of the fixed pixel-dimension h), and that the sensed image $Z_S(\underline{x})$ (again at all $\underline{x} \in h\mathbb{Z}^2$) has the form

$$Z_S(\underline{x}) = Z_R(\underline{x} + \underline{\theta}) + Z_N(\underline{x})$$

where θ is the unknown offset vector, not necessarily in $h\mathbb{Z}^2$, which is the object of inference in registration problems; and where Z (•) is a strictly stationary mean 0 random field which is also assumed to satisfy the ϕ-mixing condition of [De] mentioned in [La].

Further assumptions are required to describe the continuous variation of the fields Z_R, Z_N, Z_S between pixel corners. First of all, we assume (denoting $(\lfloor t_1 \rfloor, \lfloor t_2 \rfloor)$ by $\lfloor \underline{t} \rfloor$ and $\{\underline{t}\} \equiv \underline{t} - \lfloor \underline{t} \rfloor$)

$$(39) \qquad Z_N(t) = (1-\{t_1\})(1-\{t_2\})Z_N(\lfloor \underline{t} \rfloor) + (1-\{t_1\})\{t_2\}.Z_N(\lfloor \underline{t}$$

$$+ \underline{e}_2) + \{t_1\}(1-\{t_2\})Z_N(|\underline{t}| + \underline{e}_1) + \{t_1\}$$
$$\{t_2\}\, Z_N(\underline{t} + \underline{1})$$

where we have defined units by letting $h = 1$ and $\underline{e}_1 = (1,0)$, $\underline{e}_2 = (0,1)$, $\underline{1} = (1,1)$. This assumption means that Z_N at a point $\underline{t}$ interior to a given pixel J takes value which is a weighted average of the values at the corners of J with weights proportional to the area of overlap of a unit square with lower-left corner $\underline{t}$ with squares the lower-left corners of which are the four corners of J. In addition, we make one of two model assumptions on Z_R:

(40) with respect to the given pixel-lattice, $Z_R(\cdot)$ satisfies (39)

(41) $Z_R(\underline{t}) = Z_R(|\underline{t}|)$ for $\underline{t} = (t_1, t_2) \in \mathbb{R}^2$.

Assumption (41) means that we regard the reference-image grey-level as homogeneous within each pixel.

Next suppose that based on a large "window" $[-T, T] \times [-T, T]$ in our plane coordinates, we form the "correlation-statistic"

$$C(\underline{t}) = 1/(4T^2) \int_{-T}^{T}\int_{-T}^{T} Z_R(\underline{x}) Z_S(\underline{x}-\underline{t})\, d\underline{x} \quad , \; t \in \mathbb{R}^2$$

which will have mean

$$D(\underline{t}) = 1/(4T^2) \int_{-T}^{T}\int_{-T}^{T} Z_R(\underline{x} + \underline{\theta} - \underline{t})\, d\underline{x}.$$

Assuming that Z_R itself, although known, arose from a realization of a strictly stationary ergodic random field, then $D(t)$ has a well-defined limit as T gets large and it follows from work of [De] that $C(\cdot)$ considered as a plane random field is approximately Gaussian. The main result of [La] was the following (Lemma 3.1 and Corollary 3.2 specialized question of subpixel estimation by bounding the accuracy of interpolation possible for $D(\underline{r})$, i.e. by describing the features of Z_R (assumed to be a fixed realization of a smooth strictly stationary random field) which in the absence of Z_N limit the accuracy of recovery of $\underline{\theta}$ from observations of $D(\cdot)$ at pixel vertices. Under a further regularity condition on the stationary random field generating Z_R (existence of second specral moments), [La] found that as T gets large the error in determining $\underline{\theta}$ by maximizing the local (Taylor series) quadric approximant to $D(\cdot)$ is at most the smaller of

$$K_1 = h^2((1/12)(\hat{\xi}_h/\hat{a}_1)^{1/2} \min\{\sec\hat{\beta}, \csc\hat{\beta}\}$$

and

$$K_2 = (12\hat{a}_1/\hat{\xi}_h)^{1/2}$$

where h is the pixel-width as before, $L \equiv [T/h]$,

$$\hat{\xi}_h \equiv (2L+1)^{-2} \sum_{j=-L}^{L} \sum_{k=-L}^{L} [(\nabla_1 + \nabla_2)^2 Z_R(jh, kh)]^2$$

$\nabla_1 Z(x,y) \equiv Z(x,y) - \nabla_2 Z(x-h, y)$, $\nabla_2 Z(x,y) \equiv Z(x,y) - Z(x, y-h)$, and $\hat{a}_1$ is the smallest eigenvalue (with $\hat{\beta}$ the angle the corresponding eigenvector makes with the horizontal) of the quadratic form

$$q(\underline{y}) = (2L+1)^{-2} \sum_{j=-L}^{L} \sum_{k=-L}^{L} [(y_1\nabla_1 + y_2\nabla_2)Z_R(jh, kh)]^2$$

The size of $\min(K_1, K_2)$ might a priori be expected to determine how much more accurate than $\hat{\underline{\theta}}$ it is to estimate $\underline{\theta}$ by the maximizer $\underline{\theta}^{LS}$ of the local least-squares quadric surface approximant to $C(\cdot)$ interpolating a 3x3 array of neighboring pixel vertices.

Section 16. Objectives of the Simulation Study

The present simulation study had the following major objectives:

i) to compile "empirical" results concerning performance of $\hat{\underline{\theta}}$ and $\underline{\theta}^{LS}$ on real and simulated reference images, in the form of histograms of $\|\hat{\underline{\theta}} - \underline{\theta}\|$ and $\|\underline{\theta}^{LS} - \underline{\theta}\|$ for various values of offset $\underline{\theta}$,

ii) to compare the performance of $\hat{\underline{\theta}}$ chosen among pixel vertices to give the largest value of C(·) with that of the continuous-valued estimator $\underline{\theta}^{LS}$, and to check whether the greater accuracy can simply be ascribed to allowing $\underline{\theta}^{LS}$ to take values inside pixel-squares;

iii) to gain information on how large the standard deviation of additive noise must be compared to grey-level standard deviation in various reference images before pixel-level and subpixel registration (estimation of θ) is seriously degraded;

iv) to check the validity and usefulness of the theoretical results of [La] for 35x35 reference images, window size T=10, and T_o = 5.

In the remainder of this Section, we specify some notational conventions and tell what exactly was computed in the simluation. To begin with, each (of six) refernece image used was standardized to a 35x35 array (j,k = -17,...,

+17) with average 0 and sample variance 1 (thus

$$\sum_{j=-17}^{17} \sum_{K=-17}^{17} Z_R^2(j,k) = 1 ,$$

where we have adopted pixelwidth h=1). The offset-vector $\underline{\theta}$ for each iteration in each simulation was chosen uniformly in 0,1 x 0,1 .

The correlation-statistic C(·) was computed, for each lattice-point in the square $[-5,5]^2$, as follows. First, the expectation- term D($\underline{t}$) was calculated as a sum rather than the integral in its definition above:

$$(42) \qquad D(t) = (1/21^2) \sum_{j=-10}^{10} \sum_{k=-10}^{10} Z_R(j,k) Z_R((j,k) + \underline{\theta} - \underline{t})$$

This modification was made for two reasons: (1) although the integral could, under either assumption (40) or (41), be expressed as a weighted sum of terms $Z_R(\underline{x})$, $Z_R(\underline{y})$, the weights would depend on $\underline{\theta}$, and it was computationally much easier to make use of the equally plausible definition (42); (2) in actually practice, in the absense of a validated model assumption like (40) or (41), (42) is the definition one would use, with sums similarly replacing integrals in the definition of C(·). In each iteration of each simulation $Z_N(\cdot)$ was simulated at lattice points (in 35x35 array) as

$$Z_N(\underline{t}) \equiv \sum_{j=-1}^{1} \sum_{k=-1}^{1} Z_{t_1+j,\, t_2+j} \quad W(j,k)$$

where $\{Z_{x,y}\}$ is an array of independent identically normally distributed random variables with mean 0 and variance σ, and the W(j,k) are fixed weights which took one of two forms:

$$W_1 = \begin{matrix} 1/36 & 1/9 & 1/36 \\ 1/9 & 1/4 & 1/9 \\ 1/36 & 1/9 & 1/36 \end{matrix} \quad \text{when (40) was assumed, and}$$

$$W_2 = \begin{matrix} 0 & 1/4 & 1/4 \\ 0 & 1/4 & 1/4 \\ 0 & 0 & 0 \end{matrix} \quad \text{when (41) was assumed.}$$

Then $C(\underline{t}) - D(\underline{t})$ was calculated as

$$(43) \quad C(\underline{t}) - D(\underline{t}) = (1/(21)^2) \sum_{j=-10}^{10} \sum_{k=-10}^{10} Z_R(j,k) Z_N(j,k)-\underline{t})$$

In this definition we have replaced $(4T^2)^{-1}$ by $(21)^{-2}$ and modified some boundary terms, but (43) is otherwise the same as in its double-integral definition if $Z_N(\cdot)$ had been made up of independent $N(0,\sigma^2)$ variables at lattice points and had been interpolated according to (39) while Z_R was interpolated according to (40) or (41). (For example, under (40),

$$1/4T^2 \iint Z_R(\underline{x}) Z_N(\underline{x}-\underline{t}) d\underline{x} \simeq 1/(2\lfloor T/h \rfloor+1)^2 \sum_{\underline{i}} Z_R(\underline{i}) \{(4/9) Z_N(\underline{i}-\underline{t}) + 1/9(Z_N(\underline{i}-\underline{t}+\underline{e}_1)+ Z_N(\underline{i}-\underline{t}-\underline{e}_1)+ Z\ (\underline{i}-\underline{t}+\underline{e}_2)+ Z_N(\underline{i}-\underline{t}-\underline{e}_2)) + 1/36(Z_N(\underline{i}-\underline{t}+\underline{1})+ Z_N(\underline{i}-\underline{t}-\underline{1})+ Z_N(\underline{i}-\underline{t}+\underline{e}_1-\underline{e}_2) + Z_N(\underline{i}-\underline{t}+\underline{e}_2-\underline{e}_1))\}]$$

Two simulation experiments were performed on the DEC 2060, one with 450 iterations using weight-matrix W_1 and the other with 250 iterations using weights W_2. For each iteration, one offset θ and one array $\{Z_{jk}\}$ was generated, and for each of six reference images $D(\underline{t})$ and $C(\underline{t}) - D(\underline{t})$ calculated according to (42) and (43) with $\sigma = 1$. Then for each of a number of different values of σ, the arrays $\{D(\underline{t}) + \sigma(C(\underline{t}) - D(\underline{t}))\}$ (correlation-statistic arrays corresponding to the noise-fields $\sigma Z_N(\cdot)$ generated from the same random numbers) were used to calculate estimators $\underline{\hat{\theta}}$ (the lattice-point t corresponding to the largest array element) and $\underline{\theta}^{LS}$ (the maximum-point (x,y) for the least-squares quadric surface for the nine array values at $\underline{\hat{\theta}} + (j,k)$, $j,k = -1, 0, 1$). In addition, a third estimator was defined as

$$\underline{\theta}^* \equiv \underline{\hat{\theta}} + .5^* (\mathrm{sign}(\theta_1^{LS} - \hat{\theta}_1), \quad \mathrm{sign}(\theta_2^{LS} - \hat{\theta}_2)).$$

For each of several values of σ and each reference image on each iteration, the distances $\|\underline{\hat{\theta}} - \underline{\theta}\|$, $\|\underline{\theta}^{LS} - \underline{\theta}\|$, and $\|\underline{\theta}^* - \underline{\theta}\|$. were recorded. Output for the simulations consisted of histograms of these distances, with bin-width .1 pixel for simulation 1 and .125 pixel for simulation 2. The outputs are tabulated and interpreted in Section 18 and 19. Section 17 describes the six reference images (three artificial, three real) along with the corresponding quantities Γ, H_τ, K_1, K_2 relevant for the theoretical predictions of Section 15.

Section 17. The six reference images

Six reference images were used in our simulation study. Three of them (those numbered 3,4, and 5) were real 35x35 grey-level arrays chosen more or less arbitrarily from an 80x125 LANDSAT image of a rural area in the United States including cultivated fields, some wooded areas, and some roads. The other three images were artificially constructed, as follows:

For image 1, $Z_R(j,k) = 55.0 - 1.5*(|j| + |k|)$, $j,k=-17,-16,\ldots,+17$;

For image 2,

$$Z_R\,(j,k) = \begin{cases} 0 & \text{if} \quad \max(|j|, \ |k|) \ >= \ 3 \\ 40 & \text{if} \quad \max(|j|, \ |k|) \ <= \ 2 \end{cases}$$

For image 6,

$$Z_R\,(j,k) = \begin{cases} 20 & \text{if} \quad \max(j, \ k) \ <= \ 0 \\ 10 & \text{if} \quad \max(j, \ k) \ > \ 0 \end{cases}$$

Table 17.1 contains the values of H_τ , K_1, and K_2 for the six reference images. We recall that these quantities depend only on the reference image and not on the noise-covariances. On the other hand, Γ and $\psi(u)$ do depend on the covariances of $Z_N(\cdot)$: their values are displayed in Table 17.2 for all six reference images under the assumption (40) with weight-matrix W_1 and $Z_{jk} \sim \eta(0,1)$.

TABLE 17.1

(A) τ vs. H_τ for six reference images

Image	1	2	3	4	5	6
τ	H_τ					
.7	.014	.212	.197	.187	.244	.034
1.4	.027	.382	.461	.401	.434	.069
2.1	.062	.551	.662	.518	.530	.103
2.8	.098	.636	.662	.518	,608	.103
3.5	.150	.806	.669	.580	.643	.137
4.2	.202	.890	.669	.590	.678	.172
4.9	.235	.975	.669	.590	.678	.172
5.6	.329	1.0	.720	1.0	.734	.322
6.3	.399	1.0	.850	1.0	.770	.372
7.0	.469	1.0	.895	1.0	.866	.422

(B) K_1 and K_2 for reference images

Image	1	2	3	4	5	6
K_1	.183	.816	1.26	1.30	1.54	.60
K_2	5.48	1.22	.813	.784	.747	1.66

TABLE 17.2. Γ, $\psi(1)$, and $\psi($ 2) values

Image	Γ	$\psi(1)$	$\psi(1.414)$
1	.0453	.0075	.0105
2	.0615	.0207	.0284
3	.0568	.028	.034
4	.0603	.030	.035
5	.0581	.025	.029
6	.0536	.016	.022

TABLE 17.3. Smallest τ (in multiples of .7) for which $x^*(\tau)$ >= 4.5, for six reference images and four values of σ.

σ =	1	.5	.25	.125
Image				
1	4.9	3.5	2.1	2.1
2	1.4	.7	.7	.7
3	1.4	.7	.7	.7
4	1.4	.7	.7	.7
5	1.4	.7	.7	.7
6	5.6	4.2	2.1	1.4

We have given values for $\psi(1)$ and $\psi(1.414)$. Actually, only values of $\psi(u)$ for 0 <= <= 1/3 are relevant to estimating

$x(\tau)$. For purposes of approximate calculation, we treat $\psi(\cdot)$ as being linear on [0,1], in which case $2(\sqrt{2} - 1)^{-1} \int_1^\infty \psi(3^{-u^2})du < .75\psi(1)$. In further calculations, we therefore replace $x(\tau)$ by

$$x^*(\tau) \quad H_\tau/(\Gamma + .75\psi(1))$$

Now according to the Proposition of Section 1, with $T_o = 5$, $T = 10$,

$$P[\| \hat{\underline{\theta}} - \underline{\theta}\| >= \tau] <= 1.35 \, (e^{-(x^*(\tau))^2/2} / x^*(\tau)) \; 10^4$$

The right-hand side of this inequality is approximately 1.1 for $x^* = 4$, .12 for $x^* = 4.5$ and .01 for $x^* = 5$. Thus we have tabulated (in Table 17.3) for all six reference images, the smallest τ (in multiples of .7 pixels) for which $x^*(\tau) >= 4.5$. Note that reducing σ by a factor 1/2 does not change H_τ but multiplies both Γ and ψ by 1/2, so that x^* is inversely proportional to σ. Also note that by our definitions $\{C(\underline{t}) - D(\underline{t}) = t_i \in h\mathbb{Z}, \; i = 1,2\}$ is a strictly stationary Gaussian random field not depending on $\underline{\theta}$.

Section 18. Simulation Results

The histograms produced for $\|\hat{\underline{\theta}} - \underline{\theta}\| \; \| \underline{\theta}^{LS} - \theta \|$, and $\|\underline{\theta}^* - \underline{\theta}\|$, according to the simulation design described in Section 16, are tabulated in a slightly different form in Tables 18.1 and 18.2.

Defining empirical distribution functions for each simulation by

$$\hat{F}(x) \equiv (\#\text{iterations for which } \|\hat{\underline{\theta}} - \underline{\theta}\| <= x)/(\#\text{itera-}$$

tions)

$F^{LS}(x) \equiv$ (#iterations for which $\|\underline{\theta}^{LS} - \underline{\theta}\|$ <= x)/(#iterations),

we display for each simulation and each reference image, for three selected values of σ, the values $\hat{F}(x)$, $F^{LS}(x)$ for .1 <= x <= 1.7 in increments of .1 in simulation 1, and .125 <= x <= 1.5 in increments of .125 in simulation 2. We have not tabulated the results with the artificial estimator $\underline{\theta}^{*}$, which was introduced to see if $\underline{\theta}^{LS}$ derived its accuracy simply by allowing values inside pixels, because $\underline{\theta}^{*}$ turned out to be so conclusively inferior to $\underline{\theta}^{LS}$ and to $\hat{\underline{\theta}}$. To show this vividly, we consider Table 18.3, in which are displayed the empirical upper quartile points (75th percentiles) of $\|\hat{\underline{\theta}} - \underline{\theta}\|$, $\|\underline{\theta}^{LS} - \underline{\theta}\|$, and $\|\underline{\theta}^{*} - \underline{\theta}\|$ for each reference image and each of three values of σ. These were calculated by linearly interpolating the empirical distribution functions from simulation 1 to find the x corresponding to distribution function value .75.

TABLE 18.1. Empirical distribution functions $\hat{F}$ and F^{LS} (in parentheses) for each of six reference images and three values of σ, from simulation 1 (450 iterations)

(18.1a) Image 1

σ =	.10	.20	.30
.1	.016 (.038)	.007 (.009)	.004 (.004)
.2	.091 (.173)	.027 (.036)	.013 (.016)
.3	.207 (.278)	.067 (.091)	.038 (.036)
.4	.318 (.464)	.107 (.131)	.058 (.064)
.5	.429 (.573)	.173 (.200)	.102 (.098)
.6	.560 (.711)	.224 (.278)	.127 (.131)
.7	.660 (.802)	.267 (.351)	.169 (.178)
.8	.724 (.884)	.336 (.431)	.193 (.236)
.9	.800 (.938)	.400 (.500)	.244 (.291)

1.0	.840 (.969)	.458 (.569)	.289 (.311)
1.1	.896 (.987)	.540 (.609)	.349 (.393)
1.2	.927 (.993)	.622 (.678)	.400 (.442)
1.3	.956 (1.0)	.664 (.740)	.440 (.480)
1.4	.989	.724 (.804)	.484 (.522)
1.5	.993	.776 (.849)	.527 (.567)
1.6	.996	.824 (.880)	.582 (.616)
1.7	.996	.856 (.907)	.616 (.66)

(18.1b) Image 2 $\hat{F}(x)$ ($F^{LS}(x)$ in parentheses)

σ =	.2	.4	.6
x =			
.1	.029 (.247)	.029 (.100)	.024 (.022)
.2	.142 (.664)	.138 (.293)	.109 (.171)
.3	.322 (.878)	.304 (.516)	.229 (.284)
.4	.524 (.969)	.462 (.678)	.358 (.411)
.5	.753 (.998)	.644 (.793)	.461 (.522)
.6	.907 (1.0)	.782 (.864)	.564 (.598)
.7	.980	.900 (.927)	.669 (.696)
.8	.993	.940 (.958)	.729 (.747)
.9	1.0	.967 (.969)	.767 (.773)
1.0		.976 (.987)	.796 (.813)
1.1		.980 (.989)	.820 (.827)
1.2		.984 (.989)	.829 (.856)
1.3		.989 (.989)	.851 (.880)
1.4		.989 (.993)	.867 (.898)
1.5		.989 (.996)	.887 (.907)
1.6		.991 (.996)	.900 (.916)
1.7		.993 (.996)	.916 (.920)

(18.1c) Image 3 $\hat{F}(x)$ ($F^{LS}(x)$ in parentheses)

σ =	.4	.8	1.2
x=			
.1	.029 (.353)	.029 (.171)	.029 (.089)
.2	.142 (.816)	.142 (.524)	.142 (.316)
.3	.322 (.967)	.320 (.802)	.311 (.558)
.4	.527 (.998)	.509 (.911)	.480 (.738)
.5	.787 (1.0)	.733 (.969)	.676 (.838)
.6	.938	.889 (.989)	.827 (.911)
.7	.989	.962 (.996)	.911 (.942)
.8	.998	.987 (.998)	.958 (.967)
.9	1.0	1.0 (1.0)	.982 (.978)
1.0			.987 (.982)
1.1			.991 (.987)
1.2			.991 (.987)

1.3			.991 (.987)
1.4			.991 (.987)
1.5			.991 (.987)
1.6			.991 (.987)
1.7			.993 (.989)

(18.1d) Image 4 $\hat{F}(x)$ ($F^{LS}(x)$ in parentheses)

σ=	.4	.8	1.2
x=			
.1	.029 (.180)	.029 (.140)	.029 (.067)
.2	.142 (.556)	.142 (.400)	.140 (.276)
.3	.322 (.869)	.318 (.658)	.302 (.484)
.4	.527 (.980)	.513 (.811)	.478 (.638)
.5	.789 (.996)	.733 (.924)	.651 (.751)
.6	.933 (1.0)	.873 (.971)	.800 (.847)
.7	.980	.931 (.987)	.871 (.911)
.8	.996	.969 (.989)	.924 (.938)
.9	1.0	.991 (.993)	.956 (.960)
1.0		.996 (.993)	.969 (.971)
1.1		.996 (.993)	.973 (.976)
1.2		.996 (.993)	.976 (.976)
1.3		.996 (.998)	.978 (.980)
1.4		.998 (.998)	.982 (.980)
1.5		.998 (.998)	.987 (.982)
1.6		.998 (.998)	.989 (.989)
1.7		.998 (1.0)	.991 (.989)

(18.1e) Image 5 $\hat{F}(x)$ ($F^{LS}(x)$ in parentheses)

σ=	.4	.8	1.2
x =			
.1	.029 (.318)	.029 (.127)	.029 (.064)
.2	.142 (.798)	.142 (.418)	.138 (.229)
.3	.322 (.964)	.318 (.682)	.300 (.424)
.4	.527 (.998)	.509 (.847)	.458 (.611)
.5	.791 (1.0)	.742 (.927)	.660 (.731)
.6	.933	.876 (.971)	.784 (.807)
.7	.991	.978 (.982)	.882 (.864)
.8	1.0	.991 (.993)	.916 (.902)
.9		.998 (.996)	.951 (.924)
1.0		.998 (.996)	.962 (.940)
1.1		.998 (1.0)	.962 (.956)
1.2		.998	.964 (.967)
1.3		.998	.971 (.969)
1.4		.998	.976 (.978)
1.5		.998	.976 (.978)

1.6		.998	.978 (.980)
1.7		.998	.980 (.980)

(18.1f) Image 6 $\hat{F}(x)$ ($F^{LS}(x)$ in parentheses)

$\sigma =$	.4	.8	1.2
x =			
.1	.029 (0.)	.029 (.016)	.024 (.013)
.2	.142 (.018)	.127 (.073)	.102 (.102)
.3	.307 (.120)	.264 (.193)	.220 (.198)
.4	.447 (.369)	.396 (.360)	.331 (.333)
.5	.604 (.631)	.562 (.516)	.467 (.438)
.6	.720 (.864)	.673 (.704)	.573 (.567)
.7	.856 (.978)	.784 (.829)	.664 (.664)
.8	.942 (.998)	.878 (.911)	.749 (.740)
.9	.980 (1.0)	.931 (.944)	.804 (.791)
1.0	.998	.964 (.964)	.836 (.824)
1.1	1.0	.976 (.971)	.856 (.856)
1.2		.978 (.980)	.860 (.871)
1.3		.978 (.982)	.867 (.884)
1.4		.982 (.982)	.871 (.887)
1.5		.984 (.987)	.884 (.898)
1.6		.984 (.987)	.889 (.907)
1.7		(.987)(.987)	.893 (.918)

Table 18.2. Empirical d.f.'s $\hat{F}$ and F^{LS} (in parentheses) for each of six reference images and two values of σ, from simulation 2 (250 iterations).

(18.2ab) Images 1 and 2

	Image 1		Image 2	
$\sigma =$	.1	.3	.2	.6
x=				
.125	.028(.080)	.000(.008)	.032(.308)	.024(.028)
.25	.092(.256)	.008(.036)	.160(.796)	.112(.164)
.375	.248(.424)	.052(.088)	.448(.944)	.280(.296)
.5	.396(.604)	.096(.108)	.748(.992)	.428(.460)
.625	.572(.728)	.136(.152)	.916(1.0)	.536(.596)
.75	.700(.832)	.176(.200)	.988	.660(.672)
.875	.788(.908)	.216(.248)	1.0	.724(.712)
1.0	.840(.972)	.264(.328)		.756(.752)
1.125	.908(.988)	.324(.388)		.780(.796)
1.25	.952(.992)	.42 (.46)		.796(.820)
1.375	.984(.996)	.508(.512)		.816(.844)
1.5	.996(.996)	.556(.564)		.836(.852)

(18.2 cdef) Images 3,4,5,6

Image 3 Image 4

σ =	.4	1.2	.4	1.2
x=				
.125	.032(.412)	.032(.128)	.032(.128)	.032(.12)
.25	.16 (.900)	.152(.392)	.16 (.736)	.152(.332)
.375	.444(.992)	.412(.656)	.448(.952)	.416(.556)
.5	.748(1.0)	.644(.816)	.776(.996)	.676(.732)
.625	.932	.852(.904)	.940(1.0)	.812(.832)
.75	.996	.936(.956)	1.0	.936(.876)
.875	1.0	.968(.964)		.968(.92)
1.		.984(.984)		.98 (.944)
1.125		.992(.988)		.984(.956)
1.25		.996(.988)		.988(.968)
1.5		.996(.992)		.988(.976)

	Image 5		Image 6	
σ =	.4	1.2	.12	.36
x=				
.125	.032(.452)	.032(.096)	.032(0.)	.028(.024)
.25	.16 (.90)	.148(.288)	.156(.044)	.124(.088)
.375	.444(.996)	.388(.512)	.324(.204)	.292(.224)
.5	.764(1.0)	.628(.676)	.488(.576)	.432(.42)
.675	.952	.792(.804)	.672(.876)	.568(.564)
.75	.996	.884(.884)	.86(.98)	.70(.692)
.875	1.0	.908(.896)	.968(.996)	.78(.74)
1.		.92(.908)	.996(1.0)	.82(.816)
1.125		.928(.92)	1.0	.84(.86)
1.25		.94(.936)		.856(.892)
1.5		.948(.96)		.888(.916)

Table 18.3. Triples of empirical 75th percentile values for $(\|\hat{\underline{\theta}} - \underline{\theta}\|, \|\underline{\theta}^{LS} - \underline{\theta}\|, \|\underline{\theta}^{*} - \underline{\theta}\|)$ from simulation 1 (450 iterations), for each reference image and each of three values of σ.

σ	Image 1	σ	Image 2	σ	Image 6
.10	(.83,.64,.80)	.2	(.50,.24,.58)	.12	(.62,.55,.90)
.20	(1.45,1.32,1.31)	.4	(.58,.46,.73)	.24	(.67,.64,.97)

.30	(2.1,1.86,1.94)	.6	(.86,.8,1.02)	.36	(.8,.82,1.1)
σ	Image 3	σ	Image 4	σ	Image 5
.4	(.49,.19,.53)	.4	(.49,.26,.59)	.4	(.48,.19,.55)
.8	(.51,.28,.60)	.8	(.51,.36,.66)	.8	(.50,.44,.64)
1.2	(.55,.41,.68)	1.2	(.57,.50,.75)	1.2	(.57,.52,.80)

To complete this Section, we now discuss the accuracy of the empirically estimated numbers in Tables 18.1-18.3. All the distribution functions values p are with approximate probability $1-\alpha$ contained in the symmetric interval of length $p(1-p)\ \Phi^{-1}(1-\alpha/2)\sqrt{n}$ around the empirically estimated values where Φ is the standard normal distribution function and n is the number of iterations in the simulation. With n=450, substituting 1/2 for p, we find the conservative $(1-\alpha)$-quantiles for each t:

$$\text{percentage points for } |F_{est}(t)-F(t)| \cong \begin{cases} .019 & \alpha \lesssim .10 \\ .023 & \alpha \lesssim .05 \\ .026 & \alpha \lesssim .02 \end{cases}$$

In order to take account of our having estimated d.f.-values F(t) by empirical estimates $F_{est}(t)$ for many t simultaneously, the Kolmogoroff-Smirnoff approximate percentage points for n=450 are relevant:

$$\text{percentage points for } \sup_t |F_{est}(t)-F(t)| \cong \begin{cases} .058 & \alpha = .10 \\ .064 & \alpha = .05 \\ .077 & \alpha = .01. \end{cases}$$

Finally, in Table 6 we have empirically estimated upper quartiles for random variables like $\|\hat{\underline{\theta}} - \underline{\theta}\|$. Although it's hard to assess the accuracy of the linear interpolation we have used, the ordinary binomial-normal confidence interval (with n=450) for any t near the upper quartile of F(•) (with F(t) near 3/4) yields F(t) with 98% probability in the range $F_{est}(t) \pm .02$. Therefore we can ascribe extremely high confidence to the first decimal place of the upper-quartile estimates, and if F( ) (e.g. the d.f. of $\|\hat{\underline{\theta}} - \underline{\theta}\|$) were approximately linear within increments of .1 for x between 0 and 1.7, we could have approximately 98% confidence that the error in upper quartile estimates would be at most ± .02.

Section 19 Interpretation of Results: Conclusions

Our first and immediate conclusion from comparing Table 17.1(B) with Table 18.1 is that the figures-of-merit K_1 and K_2 for subpixel estimation are on the one hand too crude to be of use, since the order of subpixel estimation they allow without noise is .8 pixel or worse for images 2-5, and on the other hand not at all predictive of either the size of $\|\hat{\underline{\theta}} - \underline{\theta}\|$ or the improvement of $\|\underline{\theta}^{LS} - \underline{\theta}\|$ over $\|\hat{\underline{\theta}} - \theta\|$. Although one could hope to refine these figures-of-merit by estimation of further spectral moments of $Z_R(\cdot)$, the payoff would seem to be much too small for the stringency of assumptions on reference images which one would have to impose. We therefore do not recommend the use of such noise-free figures of merit for subpixel registration.

The second obvious conclusion of our study has already been mentioned but should be expanded: not only do the simulation results in Tables 18.1-18.3 establish the superiority of over $\underline{\theta}^{LS}$ in estimating $\underline{\theta}$, but the artificial estimator $\underline{\theta}^*$ (which attempts to bridge the gap between $\underline{\theta}$ and $\underline{\theta}^{LS}$ by shifting $\hat{\underline{\theta}}$ to the center nearest $\underline{\theta}^{LS}$ of a pixel with vertex $\underset{\sim}{\hat{\theta}}$) is markedly worse than both! In other words, for the types of moving-average Gaussian noise fields studied, the subpixel improvement of $\underset{\sim}{\hat{\theta}}$ by $\underset{\sim}{\theta}^{LS}$ makes $\underset{\sim}{\theta}^{LS}$ the estimator of choice for $\underset{\sim}{\theta}$ (in the absence of more detailed geometric information about Z_R).

Some quantitative discussion of the simulation results will give a sharper focus to our conclusions. Considering Table 6 first, we see that the accuracy of $\underset{\sim}{\theta}$ is relatively insensitive to the noise-level parameter σ for the real reference images (3-5) and that $||\underset{\sim}{\hat{\theta}}-\theta||$ is less than .5 pixel, for σ between .4 and 1.2, roughly 75% of the time. For these images, $||\underset{\sim}{\theta}^{LS}-\theta||$ has upper-quartile ranging from .2 to .5 pixels as σ ranges from .4 to 1.2, and the advantage of $\underset{\sim}{\theta}^{LS}$ over $\underset{\sim}{\hat{\theta}}$ deteriorates as σ gets larger than 1.0. Indeed, Tables 18.1 and 18.2 strongly support the following generalization: for many images 2,3,4, and 5, when $|| \underset{\sim}{\hat{\theta}} - \underset{\sim}{\theta} ||$ is less than about .6 pixel, $|| \underset{\sim}{\theta}^{LS} - \underset{\sim}{\theta} ||$ is (stochastically) smaller than $||\underset{\sim}{\hat{\theta}}-\underset{\sim}{\theta}||$ by .1 pixel or more for small (but this advantage is diluted by larger σ). Quite generally, for all six images, there seems to be no advantage of $\underset{\sim}{\theta}^{LS}$ over $\underset{\sim}{\hat{\theta}}$ when $||\underset{\sim}{\hat{\theta}}-\underset{\sim}{\theta}||$ is .9 pixel or more.

Images 1 and 6 (both artificial, with strong geometric structure, and quite nonstationary) are special in (i) showing very little advantage for $\underset{\sim}{\theta}^{LS}$ over $\underset{\sim}{\hat{\theta}}$, except for the smallest value of σ, and (ii) showing very rapid loss of accuracy as σ increases (e.g., the upper quartiles in Table 18.3 for $||\underset{\sim}{\theta}^{LS} -\underset{\sim}{\theta}||$ are larger for Images 1 and 6 than for the other images, with σ only half as large or less).

The last topic requiring detailed comment is the comparison of predictions in Table 17.3 with empirical results in Tables 18.1 and 18.5. The special features of Imagesl and 6 are, if anything more sharply brought out in Table 17.3 than in the empirical results, reflecting in part the conservatism of the probability inequality of Section 15. However, the theoretical inequality together with Table 17.3 very satisfactorily shows the subpixel accuracy of registration attainable on images 2-5. This, of course, is borne out strongly both in simulation 1 (to which Tables 17.1(a), 17.2, and 17.3 are directly relevant and in simulation 2.

Summary

According both to theoretical inequalities and the simulation study reported here, automatic subpixel registration with respect to real grey-level reference images (assumed to be observed translated, with a stationary noise field added to the pixel grey-levels) seems quite feasible. The present simulation study, one of the first systematic performance evaluations of the maximum-correlation method of image-registration and of a known effective variant based on maximizing a least-squares quadric surface locally approximating the (discrete) correlation-statistic near its (discrete) maximum, shows that even if the additive noise has standard deviation as large as that of the (35x35) reference image, the upper qauartile of the error in registration need be no more (and may be much less) than .25 to .5 pixel.

Section 20. Conclusions

We have developed geometric and probabilistic models for subpixel accuracy in image registration and edge location. These models have been used to develop and analyze procedures to perform these tasks. Initial experiments indicate a high level of subpixel accuracy may be attainable with the grey level geometric methods, though considerable experimentation will be required to validate this. Our analysis of digital lines methods is reasonably complete and indicates an average error of about 1/20 pixel. This result, which was based on restrictive assumptions, led to direct edge estimation procedures using the digitization of an edge (including grey levels). This method, which was briefly tested on grey-level imagery formed from Landsat data, gave similar accuracy without relying on the restrictions of the strictly geometric method.

An estimate for determining the error in using the peak of the cross-correlation between sensed and reference images as an estimate of the offset was developed. Simulations were used to determine the reliability of the error estimate and to determine the errors resulting from interpolation of the correlation function to locate a subpixel peak. The level of subpixel accuracy as a function of the signal noise was analyzed using simulations.

The primary direction for future work will be the analysis and testing of the procedure for estimating real edge location using the artificial edge digitizations as masks. Reasonable directions of research include further testing on noise imagery, computation optimization, and extension of our previous analysis of geometric registration to incorporate random grey level noise for the analysis of geometric registration to incorporate random grey level noise for the analysis of this new procedure.

References

[De] Deo, C. A functional central limit theorem for stationary random fields, Ann. Prob. 3 (1975) 708-715.

[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.

[Do-1] Dorst, L. On digitized straight line segments, internal report, Delft 1982.

[Do2] Dorst, L. personal communication

[Ha] Haralick, R.M. Digital step edges from zero crossing of second directional derivatives, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 6, No. 1, (1984) 58-68.

[Ha-Wr] Hardy, G.H. and Wright, E.M., An Introduction to the Theory of Numbers, (Oxford at the Clarendon Press, 1971).

[Hy-Da] Hyde, P.D. and Davis, L.S., Subpixel edge estimation, Pattern Recognition, Vol. 16, No. 4 (1983) 413-420.

[La] Lavine, D., Kanal, L.N., Berenstein, C.A., Slud, E., Herman, C., Analysis of subpixel registration accuracy, in Proceedings of the NASA Symposium on Mathematical Pattern Recognition and Analysis, Houston, Texas, June 1983, 327-412.

[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.

[S] Santalo, L.A. Integral Geometry and Geometric Probability, Addison-Wesley, 1976.

[W-R] Weiman, C. and Rothstein, J., Pattern recognition by retina-like devices, Technical Report No. OSU-CISRC-TR-72-8, Department of Computer and Information Science, Ohio State University, 1972.

ORT
79
[illegible]

APPENDIX

AGENDA

Wednesday, June 6:

9:00 - 9:30 Program Overview
M. Kristine Butera, Remote Sensing Science Program Manager, NASA Headquarters, Washington, D.C.

R. P. Heydorn, Science Manager, Fundamental Research Program: MPRIA, NASA/Johnson Space Center, Houston, Texas

Math/Stat: First Session

9:30 - 10:15 R. P. Heydorn, NASA/JSC and M. V. Martin, LEMSCO
"Estimating Location Parameters in a Mixture"

10:15 - 10:45 Break

10:45 - 11:30 David Scott and Rod Jee, Rice University
"Nonparametric Analysis of Minnesota Pine Tree Data and Landsat Data"

11:30 - 1:30 Lunch

Math/Stat: Second Session

1:30 - 2:15 Wayne Lawton and Meemong Lee, Jet Propulsion Lab
"Texture Classification Using Autoregressive Filtering"

2:15 - 3:00 Charles Peters, University of Houston
"Bayesian Estimation of Normal Mixture Parameters"

3:00 - 3:30 Break

3:30 - 4:15 J. Hill, D. V. Hinkley, H. Kostal, and C. Morris, University of Texas at Austin
"Advances in Empirical Bayes Modeling and Estimation for Spatial Data"

4:15 - 5:00 L. F. Guseman, Jr. and L. Schumaker, TAMU
"Multivariate Spline Methods in Surface Fitting"

6:00 - 10:00 Social Hour and Banquet, Gilruth Center

Thursday, June 7:

9:15 - 9:30 Announcements

Math/Stat: Third Session

9:30 - 10:15 H. J. Newton and W. B. Smith, TAMU
"Autoregressive Spatial Estimation for Two-dimensional Times Series"

10:15 - 10:30 Break

10:30 - 11:15 R. F. Gunst and M. Y. Lakshminarayanan, Southern Methodist University
"Exploring the Use of Linear Structural Models to Improve Remote Sensing Agricultural Estimates"

11:15 - 12:00 R. S. Chhikara, LEMSCO and A. G. Houston, NASA/JSC
"Calibration or Inverse Regression: Which is Appropriate for Crop Surveys Using Landsat Data?"

12:00 - 1:30 Lunch

Pattern Recognition: First Session

1:30 - 2:15 T. Matsuyama, V. Hwang, and Larry S. Davis, University of Maryland
"Evidence Accumulation for Spatial Reasoning"

2:15 - 3:00 K. S. Shanmugan, University of Kansas
"Textual Edge Detection and Sensitivity Analysis"

3:00 - 3:30 Break

3:30 - 4:15 F. Paderes, E. Mikhail, W. Forstner, Purdue University
"Rectification of Single and Multiple Frames of Satellite Scanner Imagery Using Points on Edges as Control"

4:15 - 5:00 David Dow, National Space Technology Labs
"The Influence of the Number of Ground Control Points on the Scene-to-map Registration Accuracy"

Friday, June 8:

8:30 - 8:45 Announcements

Pattern Recognition: Second Session

8:45 - 9:30 Curtis E. Woodcock and Alan H. Strahler, Hunter College
"Exploring Spatial Variance in Images Through Scene Simulation"

9:30 - 10:15 Grahame Smith, SRI International
"Image-to-Image Correspondence: Linear Structure Matching"

10:15 - 11:15 Carlos Berenstein, Laveen N. Kanal, David Lavine, and Eric Olsen, LNK Corporation
"Subpixel Registration Accuracy: Geometrical and Statistical Results"

11:15 - 12:00 Discussion

LIST OF ATTENDEES

Paul E. Anuta, Purdue University

M. Kristine Butera, NASA Headquarters

R. S. Chhikara, Lockheed
Kelly Cunningham

Larry S. Davis, University of Maryland
H. Decell, University of Houston
David D. Dow, NASA Earth Resources Laboratory
Joy Duncan, TAMU Research Foundation

David L. Egle, TAMU

W. Forstner, Purdue University

Richard F. Gunst, SMU
L. F. Guseman, Jr., TAMU

C. Hallum, University of Houston - Clear Lake City
James Henry, University of Florida
Virginia R. Hetrick, University of Florida
R. P. Heydorn, NASA/JSC
J. Hill, UT-Austin
D. V. Hinkley, UT-Austin
Chris Hlavka, NASA-Ames Research Center
Glen Houston, NASA/JSC
Vincent Hwang, University of Maryland

Rod Jee, Rice University
Johnny Johnston, UT-Austin

Laveen Kanal, LNK Corp.
H. Kostal, UT-Austin

M. Y. Lakshminarayanan, SMU
Richard Latty, University of Maryland
Wayne Lawton, JPL
Meemong Lee, JPL

Anne Marie McAndrew, NASA/JSC
E. Mikhail, Purdue University
Thomas C. Minter, Lockheed
Carl Morris, UT-Austin

H. J. Newton, TAMU

F. Paderes, Purdue University
Charles Peters, University of Houston

Marion Reilman
John W. Rouse, Jr., University of Texas at Arlington

Larry Schumaker, TAMU
David Scott, Rice University
K. S. Shanmugan, University of Kansas
Grahame B. Smith, SRI

List of Attendees (continued)

William B. Smith, TAMU
Alan Strahler, Hunter College
Curtis Woodcock, Hunter College