

7-1-1995

An Examination of distributional assumptions in landsat tm imagery

Elizabeth Frey

Follow this and additional works at: <http://scholarworks.rit.edu/theses>

Recommended Citation

Frey, Elizabeth, "An Examination of distributional assumptions in landsat tm imagery" (1995). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the Thesis/Dissertation Collections at RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.

**AN EXAMINATION OF DISTRIBUTIONAL ASSUMPTIONS
IN
LANDSAT TM IMAGERY**

by
Elizabeth G. Frey

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Center of Imaging Science of the
Rochester Institute of Technology

July 1995

Signature of the Author _____

Accepted by _____ *Aug. 25, 1995*

Coordinator, M.S. Degree Program

Center for Imaging Science
Rochester Institute of Technology
Rochester, New York

CERTIFICATE OF APPROVAL

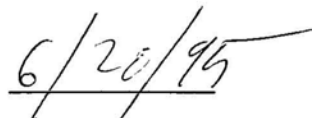
M.S. DEGREE THESIS

The M.S. degree thesis of Elizabeth Frey
has been examined and approved by the thesis
committee as satisfactory for the thesis requirement
for the Masters of Science Degree

Dr. Carl Salvaggio, Thesis Advisor

Sterling Mason

Dr. John Schott


Date

THESIS RELEASE PERMISSION FORM

Title of thesis:

**AN EXAMINATION OF DISTRIBUTIONAL
ASSUMPTIONS IN
LANDSAT TM IMAGERY**

I, Elizabeth Frey, grant permission to the Wallace Memorial Library of the Rochester Institute of Technology to reproduce this thesis in whole or in part provided any reproduction will not be of commercial use or for profit.

Elizabeth G. Frey

7/6/95
Date

AN EXAMINATION OF DISTRIBUTIONAL ASSUMPTIONS
IN
LANDSAT TM IMAGERY

by

Elizabeth G. Frey

ABSTRACT

The first portion of this study checked water, vegetation, and urban class features of LANDSAT TM data for univariate normality using Pearson's system of frequency curves. Results indicated that of the 144 image bands tested 135 were determined to be normal in distribution. The second part of the study developed an image generator that uses the mean, covariance matrix and intraband correlation of LANDSAT TM images to create synthetic class scenes. Imagery composed of multiple synthetic class scenes, which ranged from normal to non-normal in their distributions, were classified using a maximum likelihood classifier. No significant difference in classification accuracy was found between the normally distributed data and the non-normal image data.

ACKNOWLEDGEMENTS

The author wished to extend great appreciation to Dr. Carl Salvaggio for the insight and guidance he brought to this effort. His knowledge of image classification and broad research experience provided the necessary structure for a successful study. The author also wishes to thank Sterling Mason for the countless hours he gave towards this effort. His superior statistical knowledge and patient teaching skills will always be appreciated by the author. The author would also like to thank the support and guidance that Dr. John Schott extended to this study. His insight proved invaluable to this effort. Finally, the author wishes to thank Rolando Raqueño for the unparalleled support he provided towards the development of the author's computer programming skills. There will never be enough quarters to repay him.

DEDICATION

This thesis is dedicated to my grandmother, Fannie M. Frey, who taught me the importance of a sound education, and the value of a good handbag.

Table of Contents

Table of Contents	vii
Table of Tables	viii
Table of Figures	ix
1.0 Introduction	1
1.1 Historical Background	2
1.2 Characterizing Remotely Sensed Data.....	4
1.3 Current Remote Sensing Classification Algorithms	6
1.3.1 Supervised Multivariate Classification	6
1.3.2 Unsupervised Multivariate Classification.....	11
2.0 Image Selection	13
2.1 Multispectral LANDSAT Thematic Mapper Images.....	13
2.2 Subscene Locations	14
3.0 Univariate Data Analysis.....	15
3.1 Pearson's Systems of Frequency Curves.....	15
4.0 Multivariate Scene Generation	21
4.1 Overview of Synthetic Image Generator.....	21
4.2 Determining Intraband Correlation	24
4.3 Sample Covariance	25
4.4 Description of Multivariate Synthetic Image Generator	26
4.5 Description of the Imagery Created by the Synthetic.....	29
4.6 Inducing Non-normality in Synthetic Images	30
5.0 Analysis of Synthetic Imagery	33
5.1 Overview of Classification Procedures	33
5.2 Determining Classification Accuracy.....	33
5.3 Determining Signature Divergence	33
5.4 Determining Significant Changes in Classification Accuracy	34
6.0 Results and Discussion	35
6.1 Results of Univariate Analysis of LANDSAT TM Images	35
6.2 Results of Synthetic Image Classification with the	37
6.3 Conclusions and Discussion of Study Results.....	40
7.0 Recommendations for Future Work.....	43
APPENDIX A	A-1
APPENDIX B	B-1
APPENDIX C	C-1
APPENDIX D	D-1

Table of Tables

Table 2.1 Thematic Mapper sensor bands	13
Table 2.2 Residual Error after LANDSAT Parameter Correction.....	14
Table 2.3 LANDSAT TM Scene Location Data.....	14
Table 3.1 Pearson's seven "types" of distributions where	17
Table 3.2 Distributional Data for a Normal Population	19
Table 3.3 Distributional Data for a Non-normal Population.....	20
Table 4.1 Non-normal Samples Generated for Set One.....	30
Table 4.2 Non-normal Samples Generated for Set Two.....	31
Table 4.3 Non-normal Samples Generated for Set Three.....	31
Table 4.4 Non-normal Samples Generated for Set Four.....	31
Table 5.1 r x c Table	34
Table 6.1 k Values for Vegetation	36
Table 6.2 k Values for Urban	36
Table 6.3 k Values for Water.....	37

List of Figures

Figure 1 Statistical variation of reflectance for vegetation.	4
Figure 2.0 Three classes of separable spectral data.....	5
Figure 2.1 Overlapping spectral data clusters.	5
Figures 3.0 and 3.1 Minimum distance to the mean classifier	7
Figure 4 Parallelepiped classifier	8
Figure 5 a-c Sum-of-the-square-error unsupervised classifier.	12
Figure 6 Pearson's β_1 , β_2 field	18
Figure 7 High level view of synthetic image generator	22
Figures 8 Depiction of the synthetic image matrix.....	23
Figure 9 Pixel pairs for determining intraband correlation.....	24
Figure 10 Pixel pairs used to determine intraband correlation.....	25
Figure 11 Overview of synthetic image generator.	29
Figure 12 Distributions of non-normal data	31
Figure 13 Classification results for image set one	38
Figure 14 Classification results for image set two.....	38
Figure 15 Classification results for image set three.....	39
Figure 16 Classification results for image set four	39
Figure 17 Measure of Divergence Values.....	41

The advent of satellite imagery can be likened to opening the floodgates of the Grand Coulee Dam. Out spilled an unprecedented amount of data in to a community little prepared to handle it. Faced with analyzing massive quantities of data, the remote sensing community quickly adapted preexisting statistical algorithms to make the process more efficient.

One of the first statistical processes adopted by the field of remote sensing was discriminant analysis, otherwise known as classification theory. A classification algorithm assigns data from an unknown population to one of several known alternative populations based on the measurements from the unknown data¹. This quantitative decision-making procedure is well suited to today's image data when performed by a computer. The output of this process is a thematic map in which every pixel from the input image is classified into several themes or classes². Thematic maps have proven to be useful to agriculture, urban and military planners.

Present day classification maps are often derived from multispectral data. While multivariate data analysis increases the power of a classifier, it also makes the process more complex. In order to simplify the process among parametric classifiers, the remote sensing community has adopted the assumption that image data is generally normally distributed³. While there has been extensive research in the field of remote sensing regarding classification accuracy, no study has been conducted to validate the assumption regarding normality.

The intent of this study is to check for univariate normality and to determine how a parametric classifier performs on multivariate non-normal image data. This first objective is achieved by testing real image data for univariate normality via the first four moments of the class distribution. The second objective is accomplished by creating a series of multivariate non-normal synthetic images and classifying them using a parametric classifier

(maximum likelihood classifier). This study involves the theoretical development of the synthetic image generator and implementation. The goal of this work is to characterize the univariate distribution of real LANDSAT TM data, as well as establish the robustness of a parametric classifier.

1.1 Historical Background

The roots of discriminant analysis can be traced back to the fifth century BC when the historian Thucydides noted that the Athenians used averages and modal values to estimate their risk before entering warfare. These early scholars noticed that certain events usually had the same results. Aristotle summarized this by noting that "some things always come to pass in the same way", which foreshadows today's modern probability language⁴.

However modern discriminant analysis didn't begin until the first part of this century. Early work involved establishing a parameter to measure the divergence between different distributions. These first attempts at distinguishing between different populations took the general form of:

$$\frac{(x - \mu)}{\sigma} \quad (1.1)$$

Karl Pearson was one of the first mathematicians to incorporate this formula into his work. In 1921 he proposed a measurement he termed the "coefficient of radical likeness (CRL)" which he expressed as:

$$\frac{n_1 n_2}{n_1 + n_2} (\bar{X}_1 - \bar{X}_2)' S^{-1} (\bar{X}_1 - \bar{X}_2) \quad (1.2)$$

In the above formula, \bar{X}_i represents a sample mean vector from population $i = 1, 2$, n_i the sample size, and S^{-1} the pooled sample covariance. His work caught the attention of others

in the field of discriminant analysis including Mahalanobis who in 1927 presented a very similar formula:

$$D^2 = (\bar{X} - \bar{X}_2)'S^{-1}(\bar{X}_1 - \bar{X}_2) \quad (1.3)$$

which uses the sample squared distance (D^2) between two populations to separate different populations⁵.

At about the same time Mahalanobis was deriving D^2 , Fisher suggested a univariate approach to multivariate discriminant analysis. He proposed creating an optimum linear combination (later called a "linear discriminant function") of a distribution's components which maximized the squared distance between the means. This resulted in a univariate representation of a multivariate population where the univariate means are separated as much as possible relative to the populations variance⁶. This expression is expressed as follows:

$$y_0 = (\bar{X} - \bar{X}_2)'S^{-1}\bar{X}_0 \quad (1.4)$$

All of these works attempted to separate different populations based on the means and the spread of the distributions. These are the same principles which govern today's modern classification algorithms as will be demonstrated in Section 1.3.

Early multivariate discriminant analysis was used primarily in the field of anthropology and geology⁷. The success of this research can be attributed to the fact that the populations under study were well defined and separable from one another. The same attributes are desirable in remotely sensed imagery where multispectral signatures of specific ground cover classes cluster together to form distinct patterns. Showengerdt (1983) explains that the observed patterns are caused by the spectral properties of different surface materials. These spectral signatures are best characterized not as a single curve but as a family of curves whose shapes map out a spectral envelope that corresponds to a specific ground-cover class (Figure 1). The variability of these spectral signatures within the envelope can be caused by number of natural factors. Among the most prominent are atmospheric scattering, texture of land surfaces, sun and view angles⁸.

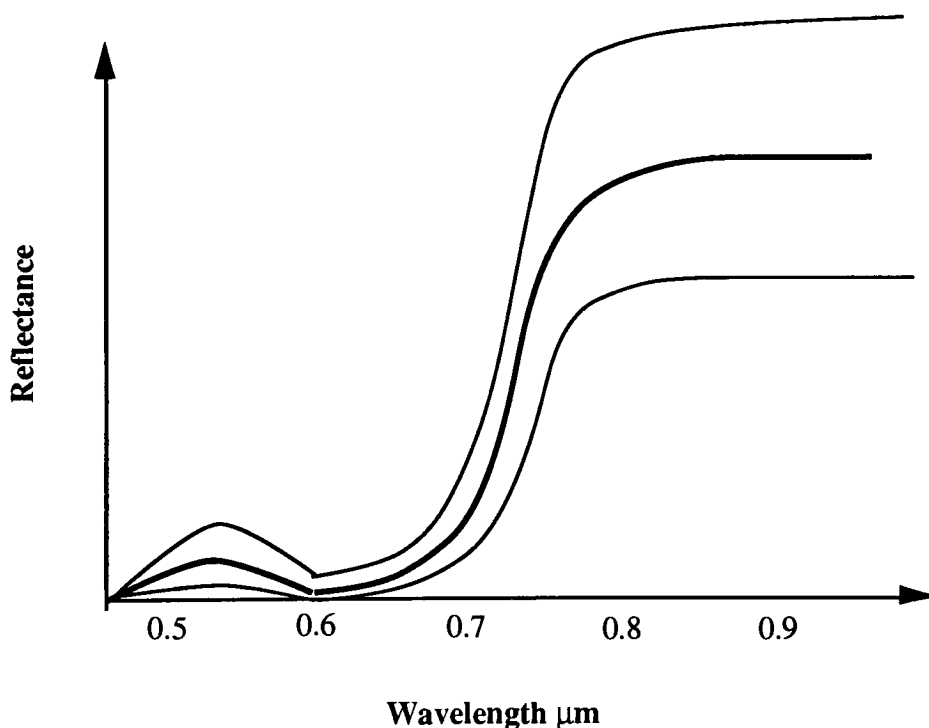


Figure 1 Statistical variation of reflectance for vegetation.

However, distinctive spectral shapes of ground features do not ensure an accurate classification. Consider Figures 2.0 and 2.1. Figure 2.0 displays three class features that could be separated (i.e. classified) successfully using a simple thresholding technique. This classification method would not work as well with Figure 2.1 since the soil and vegetation classes overlap. In cases where sample populations overlap a more sophisticated and accurate classification algorithm is required. Several multivariate techniques qualify for these cases and will be discussed in the next section.

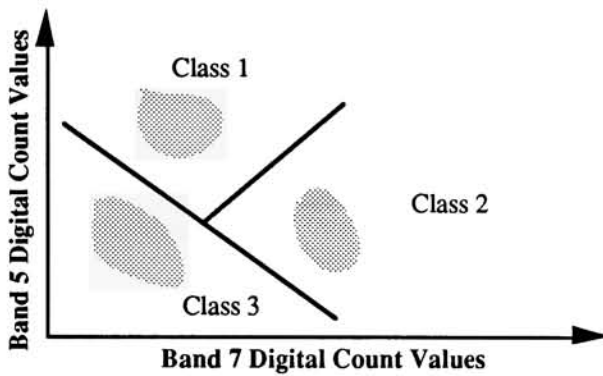


Figure 2.0

Figure 2.0 Three classes of separable spectral data

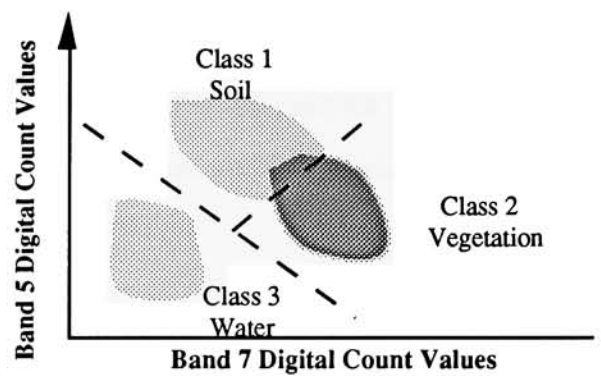


Figure 2.1

Figure 2.1 Overlapping spectral data clusters.

There are essentially two categories of classification algorithms: supervised and unsupervised classification. The first requires the user to select a representative sample of each feature of interest (e.g. vegetation, water, soil) and determine its spectral signature. Image pixels most resembling a feature in vector space are assigned to that class according to some decision rule. The second procedure uses no training data, but rather relies on a computer algorithm to determine naturally occurring clusters of feature vectors. This is called unsupervised training, where the computer picks the number and type of class signatures from parameters selected by the operator. The decision of which training procedure to use depends on the types of features under consideration, and the desired degree of accuracy⁹.

1.3.1

Supervised Multivariate Classification

Once the type of classification method has been selected, one of two decision making algorithms can be selected. These are parametric and nonparametric computer algorithms. A parametric classifier assumes a particular statistical distribution, usually a normal distribution, while a nonparametric algorithm makes no such presupposition. An example of the later is the minimum distance to the mean classifier. Pixels are compared to the mean vectors in feature space, and assigned to the class with the shortest Euclidean distance. While the minimum distance to the mean classifier uses less machine time it does not behave well when the natural variability of spectral signatures cause features to overlap one another. This phenomenon is illustrated in Figures 3.0 and 3.1. In Figure 3.0 the shortest Euclidean distance is between the candidate pixel and M_2 , resulting in the pixel being classified to class 2. However if the candidate pixel actually belonged to a class with a large variance (Figure 3.1), a classification error would occur because the minimum distance to the mean classifier does account for variability in the image data.

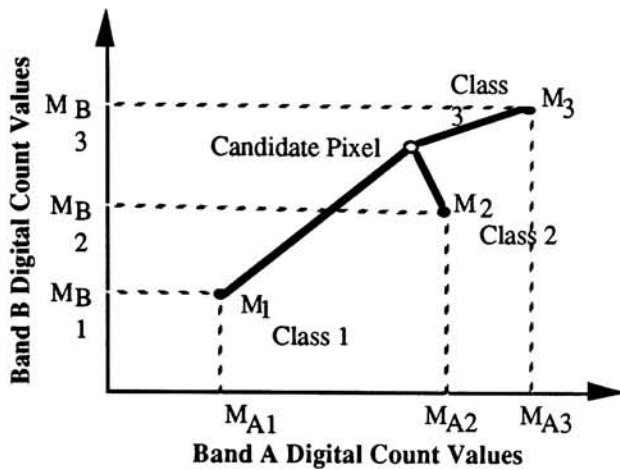


Figure 3.0

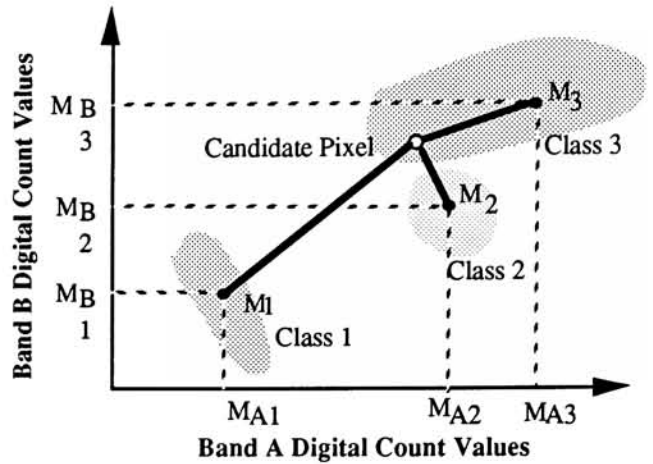


Figure 3.1

Figures 3.0 and 3.1 Minimum distance to the mean classifier

The variability of the sample distribution is taken into account by another classifier; the parallelepiped classifier. This algorithm determines the range of each class, which in the two dimensional example shown in Figure 4, is represented by a rectangle surrounding the data. Unknown pixels are classified according to what boundary the samples fall within. This classifier is as fast as the minimum distance to the mean classifier, but has the same difficulty classifying overlapping spectral signatures. In addition this classifier does not handle highly correlated data well. In Figure 4 class 4 depicts a ground feature sample from two highly correlated bands. The resulting ellipsoidal shape of the correlated data forces the range of the parallelepiped boundaries to increase thereby overestimating the size of the classification region. This increases the possibility of a misclassification error by claiming pixels that belong to another classes. While this classifier can be modified to improve its accuracy with correlated data¹⁰, it still can not match the accuracy of the parametric classifiers.

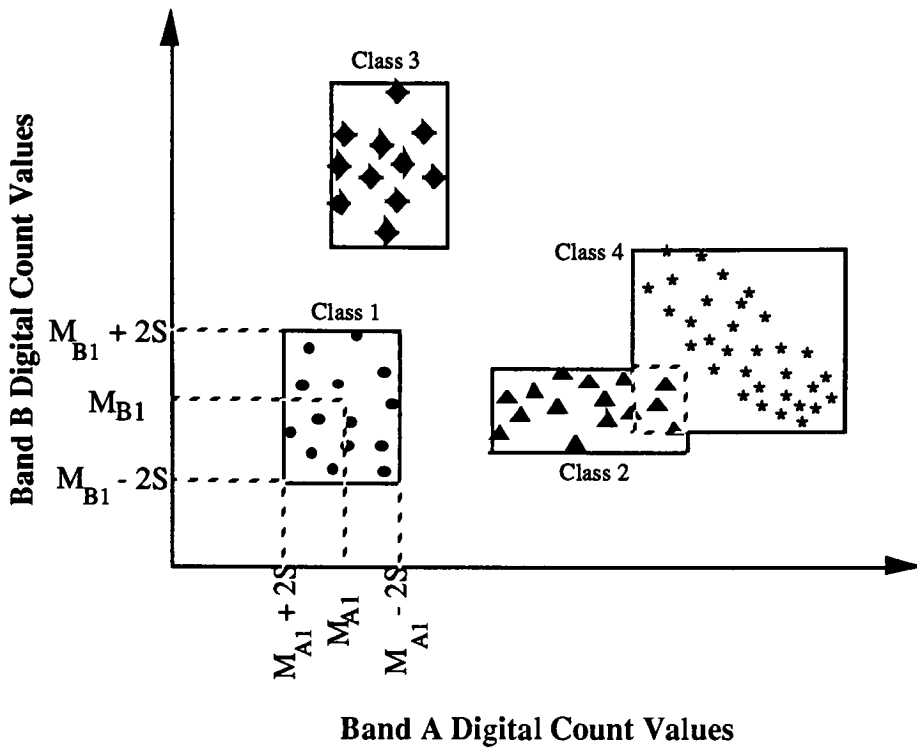


Figure 4 Parallelepiped classifier

Parametric classifiers are widely used in the remote sensing community, the most common being the maximum likelihood classifier. This algorithm is considered to be more accurate since its decision making process is based on the probability of a pixel belonging to a particular class. Simply put, each pixel of an image is treated like a column vector of brightness values. If the spectral classes of interest within an image can be represented by

$$\omega_r, r = 1, \dots, M \quad (1.5)$$

where M represents the number of classes, then the classification of vector \mathbf{x} using a parametric model becomes strictly a matter of comparing conditional probabilities of the M classes. Each conditional probability $p(\omega_r | \mathbf{x}), r = 1, \dots, M$ gives the likelihood that vector \mathbf{x} belongs to a particular class. The multivariate probability distributions can be established from the land cover training data. Final vector assignment goes to the class with the largest conditional probability (1.6).

$$\mathbf{x} \in \omega_r \text{ if } p(\omega_r|\mathbf{x}) > p(\omega_s|\mathbf{x}) \text{ for all } r \neq s \quad (1.6)$$

The above formula (1.6) can be modified to include *a priori* probabilities i.e. the probability $p(\omega_r)$ that a spectral class ω_r occurs in an image. This Bayesian approach can be written as follows:

$$p(\omega_r|\mathbf{x}) = p(\mathbf{x}|\omega_r)p(\omega_r) / p(\mathbf{x}) \quad (1.7)$$

In equation (1.7) the probability $p(\mathbf{x})$ is the probability of finding a pixel from any class at the location \mathbf{x} . The probabilities $p(\omega_r|\mathbf{x})$ are called the *posteriori* probabilities because they are the probabilities of \mathbf{x} belonging to a class ω_r after a decision has been made. Since $p(\mathbf{x})$ is a common factor in (1.7) it can be removed and the above expression rewritten as

$$\mathbf{x} \in \omega_r \text{ if } p(\mathbf{x}|\omega_r)p(\omega_r) > p(\mathbf{x}|\omega_s)p(\omega_s) \text{ for all } r \neq s \quad (1.8)$$

The decision rule in (1.8) is more acceptable since the conditional probabilities can be obtained from the training data and the classification analyst can approximate the *a priori* values($p(\omega_r)$) from the image. Note: *a priori* values are often assumed to be equal.

For mathematical convenience, $g_s(\mathbf{x})$ can be written as

$$\begin{aligned} g_s(\mathbf{x}) &= \ln\{p(\mathbf{x}|\omega_s)p(\omega_s)\} \\ &= \ln\{p(\mathbf{x}|\omega_s)\} + \ln\{p(\omega_s)\} \end{aligned} \quad (1.9)$$

and the decision rule rewritten as:

$$\mathbf{x} \in \omega_r \text{ if } g_r(\mathbf{x}) > g_s(\mathbf{x}) \text{ for all } r \neq s \quad (1.10)$$

In this form, the formula (1.10) becomes the decision rule (or discriminant function) for a univariate model. Since most classification processes use multivariate data that is assumed to be normally distributed, the formula for a multivariate conditional probability can be written for N spectral bands as:

$$p(\mathbf{x}|\omega_r) = \frac{1}{(2\pi)^{\frac{N}{2}} |\Sigma_r|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_r)^t \Sigma_r^{-1}(\mathbf{x}-\mu_r)\right) \quad (1.11)$$

where μ and Σ_r are the mean vector and covariance matrix for a given class. The premise that the training data originates from a normal population is the subject of this thesis. Both the assumption of normality expressed in (1.11) and the robustness of its discriminant function (1.13) were tested during the course of this study.

The discriminant function $g_r(\mathbf{x})$ shown in (1.10) can be written as:

$$g_r(\mathbf{x}) = \ln p(\omega_r) - \frac{1}{2} \ln |\Sigma_r| - \frac{1}{2}(\mathbf{x}-\mu_r)^t \Sigma_r^{-1}(\mathbf{x}-\mu_r) \quad (1.13)$$

If no *a priori* probabilities are available, then this formula reduces to

$$g_r(\mathbf{x}) = -\ln |\Sigma_r| - \left(\frac{1}{2}\right)(\mathbf{x}-\mu_r)^t \Sigma_r^{-1}(\mathbf{x}-\mu_r) \quad (1.14)$$

This final formula (1.14) was the one used for the image classification portion of this thesis. It presupposes a normal distribution and treats all *a priori* probabilities as being equal¹¹.

In the previous classification examples, the selection of class features is required prior to image classification. The opposite is true in unsupervised classification, where classes are chosen after the classification process according to the natural grouping or “clustering” of the sample vectors in sample space¹². Each cluster represents a probability distribution for one class that is spectrally separable from other features. Determining the relationship between the clusters in vector space and the actual image data is the responsibility of the analyst.

There are many clustering algorithms and all share the characteristic of minimizing the distances between the points within a cluster, and maximizing the spacing between clusters. One widely used clustering algorithm is "sum-of-the-square-error" criterion. This is defined by the expression:

$$SSE = \sum_{i=1}^c \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mu_i\|^2 \quad (1.15)$$

where μ_i represent the mean vector of a cluster and C_i denotes the set of data point belonging to the cluster (Note $\|\mathbf{x} - \mu_i\|$ represents the minimum Euclidean distance between \mathbf{x} and μ_i). This algorithm works by continually changing the mean of the clusters and determining Euclidean distance between the points and the means. Once a minimum Euclidean distance is reached the process is halted since the vectors are as "tight" as possible and therefore the most separable¹³. Figure 5 a-c demonstrates this iterative process of determining the minimum distance between the sample point and the mean.

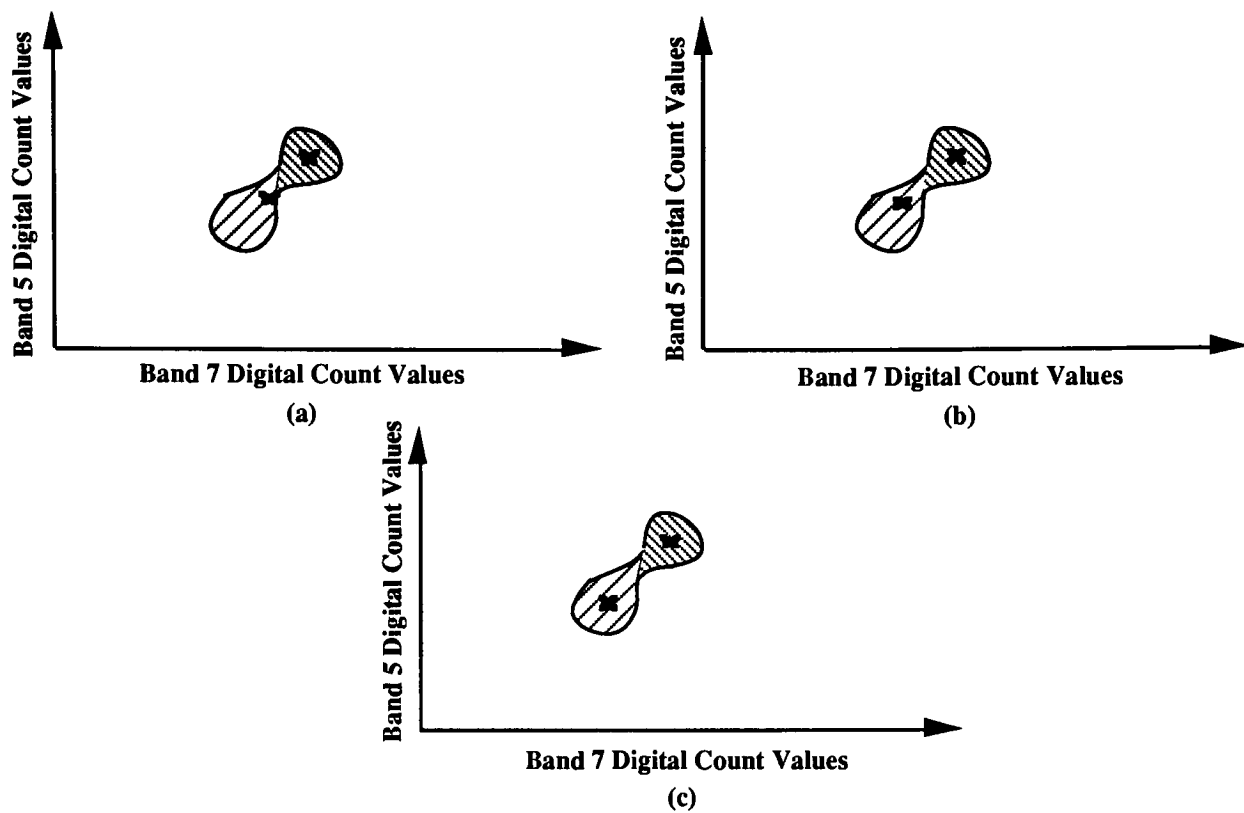


Figure 5 a-c Sum-of-the-square-error unsupervised classifier.

2.1 Multispectral LANDSAT Thematic Mapper Images

This study's analysis of univariate normality was conducted on real LANDSAT TM images. Each image contains seven bands covering visible through reflected infrared radiation with one additional band sensitive to the thermal infrared portion of the spectrum. The specific spectral bands of this TM imagery are shown in Table 2.1 and were taken from Freden and Gordon (1983).

This thesis did not use the thermal infrared portion of the spectrum (band 6) because the spatial resolution of this band differs from the other six and it is not typically correlated with the other LANDSAT bands. All the TM imagery used here was corrected at the ground processing facility for the following conditions (Table 2.2) according to Freden and Gordon (1983)¹⁴.

BANDS	RANGE(μm)
1	0.45-0.52
2	0.52-0.60
3	0.63-0.69
4	0.76-0.90
5	1.55-1.75
6	10.4-12.5
7	2.08-2.35

Table 2.1 Thematic Mapper sensor bands

Radiometric Error Correction	1 quantum level over full range
Geometric Error Correction	0.5 sensor pixels
Temporal Registration Error	0.3 sensor pixels

Table 2.2 Residual Error after LANDSAT Parameter Correction

2.2

Subscene Locations

Eight subscene images were selected for this study. Each image measures 512 x 512 pixels and contained urban, vegetation and water features. They were purposely selected from different locations within North America to provide statistical diversity among the class samples. The pertinent image data is summarized in Table 2.3.

CITY	DATE	IMAGE		PIXEL
		SIZE	TAPE ID	COORDINATES*
San Francisco CA	August 12, 1983	512 x 512	1409	1900, 1200
Charleston SC	November 9, 1982	512 x 512	2583	3700, 4150
Washington DC	November 2, 1982	512 x 512	1916	2500, 2775
Baltimore MD	November 2, 1982	512 x 512	1916	3500, 1000
Toronto Canada	May 24, 1985	512 x 512	NA	NA
Buffalo NY	June 22, 1984	512 x 512	NA	900, 2750
Rochester NY	September 13, 1982	512 x 512	NA	2390, 2727
Hartford CT	NA	512 x 512	NA	NA

* Row and column of the upper left corner of the 512 x 512 pixel subscene

Table 2.3 LANDSAT TM Scene Location Data

3.1

Pearson's Systems of Frequency Curves

This study's check for normality of LANDSAT images began with a selection of water, urban and vegetation samples from each of the eight test images described in Section 2.0. Each sample was checked for univariate normality using Karl Pearson's System of Frequency Curves. This system is described by W. P. Elderton in his book Systems of Frequency Curves (1969). Pearson, he states, developed a method of determining what distribution a sample was selected from based on the first four moments of the sample. His theory began with the simple differential equation shown in (3.1).

$$\frac{1}{y} \frac{dy}{dx} = \frac{-(x+a)}{c_0 + c_1 + c_2^2} \quad (3.1)$$

Pearson reasoned that the constants c_0, c_1, c_2 could be viewed as parameters that controlled the shape of a distribution or family of distributions. Since the moments of a mathematical function define the shape of the curve, Pearson solved the above differential in terms of the first four moments. This was accomplished by first expressing the moments as β_1 and β_2 (3.2) and then relating them to the constants c_0, c_1, c_2 by the expressions seen in (3.3).

$$\beta_1 = \frac{\mu_3^2}{\mu_2^3} \quad \beta_2 = \frac{\mu_4}{\mu_2^2} \quad (3.2)$$

where μ_2, μ_3, μ_4 = the second, third and fourth moments respectively

$$c_0 = \frac{4\beta_2 - 3\beta_1}{2(5\beta_2 - 6\beta_1 - 9)} \sigma^2$$

$$c_1 = \frac{\sqrt{\beta_1}(\beta_2 + 3)}{2(\beta_2 - 6\beta_1 - 9)} \sigma \quad (3.3)$$

$$c_2 = \frac{2\beta_2 - 3 - 6}{2(5\beta_2 - 6\beta_1 - 9)} \sigma$$

In his work, Pearson defined seven main "types" of distributions, each of which encompasses a family of distributions. A list of these types are shown in Table 3.1. Six of Pearson's types represent known sampling distributions (Table 3.1). Because the collection of Pearson's "types" originate from the same differential equation, the different solutions are continuous across the boundaries of the types. The lack of discrete boundaries is apparent in the β_1 , β_2 field as shown in Figure 6. Pearson, in an effort to simplify distributional assignments, collapsed the β_1 , β_2 field into a one dimensional κ space using the formula seen in (3.4)¹⁵. κ values were used in this study to determined the normality of LANDSAT TM samples are reported in Section 6.1.

$$\kappa = \frac{\beta_1(\beta_2 + 3)}{4(2\beta_2 - 3\beta_1 - 6)(4\beta_2 - 3\beta_1)} \quad (3.4)$$

Type	Description of Sampling Distribution	Equation
Type 1	the distribution of two independent χ^2 variables	$c(1 + \frac{x}{a_1})^{m_1-1}(1 - \frac{x}{a_2})^{m_1-1}$ for $-a_1 \leq x \leq a_2$
Type 2	the distribution of the correlation between two independent normally distributed populations	$c(1 - \frac{x^2}{a^2})^m$ for $-a \leq x \leq a$
Type 3	χ^2 distribution	$\frac{-mx}{a} e^{-\frac{mx}{a}} (1 + \frac{x}{a})^{m-1}$ for $-a \leq x < \infty$
Type 4	represents no known sampling distribution	$c(1 + \frac{x^2}{a^2})^{-m} e^{-b \tan^{-1}(\frac{x}{a})}$ for $-\infty \leq x < \infty$
Type 5	$\frac{1}{\chi^2}$ distribution	$c(1 + \frac{x}{a})^{-m} e^{\frac{-am}{x+a}}$ for $-a \leq x < \infty$
Type 6	distribution of the variance ratio i.e. F distribution	$c(1 + \frac{x}{a_1})^{-q_1} (1 + \frac{x}{a_2})^{-q_2}$ for $-a_i \leq x < \infty$
Type 7	distribution of the student ratio i.e. t distribution	$c(1 + \frac{x^2}{a_2})^{-m}$ for $-\infty \leq x < \infty$

Table 3.1 Pearson's seven "types" of distributions where:

c = a constant to make the area under the curve equal to one

m, a, q = constants

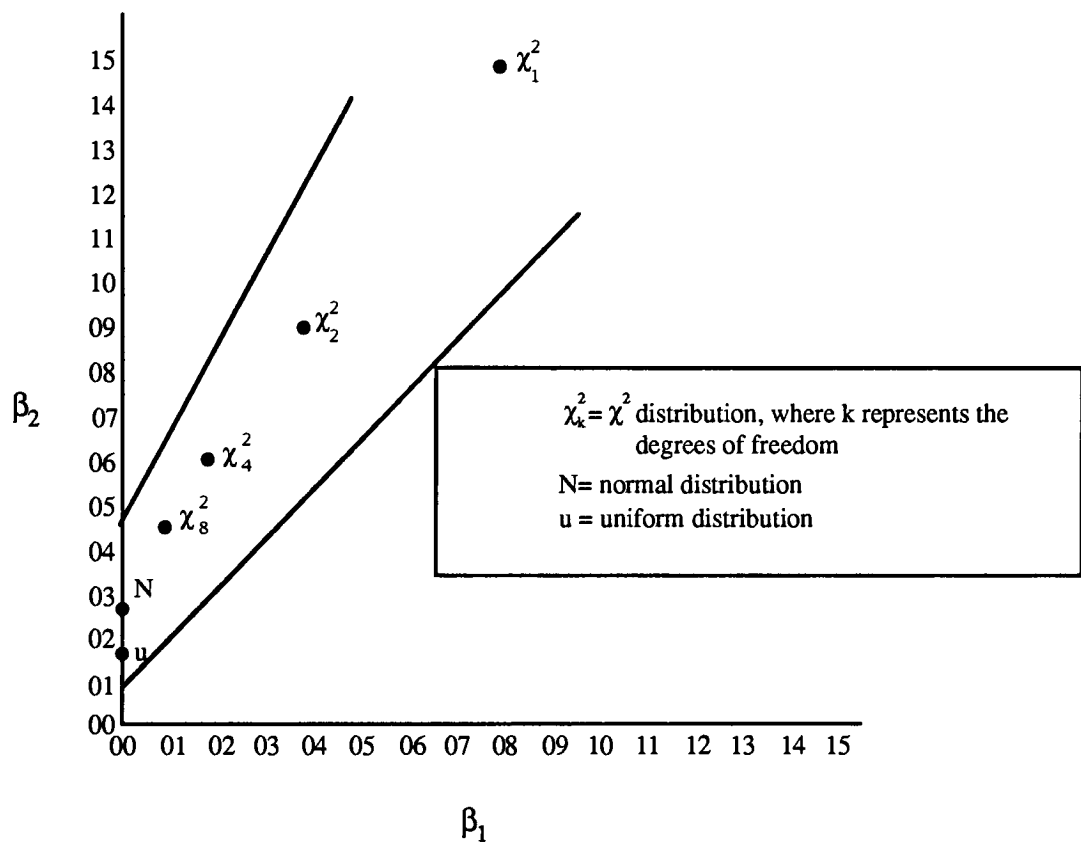


Figure 6 Pearson's β_1 , β_2 field

Pearson's System of Frequency Curves can be used to check for normality or to determine the distribution of the data under analysis. For example, consider Table 3.2 which lists the number of unexpired insurance policies against the frequency of policy holders:

Unexpired Policies in Terms of Years	Frequency of Policy Holders
0-4	11
5-9	116
10-14	274
15-19	451
20-24	432
25-29	267
30-34	116
35-39	16

Table 3.2 Distributional Data for a Normal Population

From the data listed in Table 3.2 the first four moments can be calculated. The results for μ_1 through μ_4 are listed below along with β_1 , β_2 , and κ :

$$\begin{array}{llll} \mu_1 = 19.99 & \mu_2 = 1.829 & \mu_3 = 0.12 & \mu_4 = 8.52 \\ \beta_1 = 0.002 & \beta_2 = 2.54 & \kappa = -0.007 & \end{array}$$

Reviewing Pearson's distributions, κ values with an absolute value less than or equal to 0.92 are considered to have originated from a normal distribution. Since κ is close to zero, the sample represented in Table 3.2 most likely came from population with a normal distribution. Using Pearson's System of Frequency Curves it is possible to check a sample population for normality before analyzing the data. Consider Table 3.3, which lists eight age groups of policy holders and their frequencies.

Age of Policy Holders	Frequency of Policy Holders
0-10	1
10-20	50
20-30	168
30-40	100
40-50	36
50-60	10
60-70	2
70-80	0.5

Table 3.3 Distributional Data for a Non-normal Population

$$\begin{array}{llll} \mu_1 = 0.40 & \mu_2 = 0.92 & \mu_3 = 0.89 & \mu_4 = 4.08 \\ \beta_1 = 0.99 & \beta_2 = 4.73 & \kappa = 1.89 & \end{array}$$

The statistics of this distribution indicates that this data sample has been extracted from a non-normal population.. In fact, the parent population was probably a "type VI" distribution which resembles a F distribution. Analysts using this data may want to review pending statistical analysis for assumption of normality (e.g. a test of hypothesis using a z table).

The κ values were determined for the LANDSAT images listed in Table 2.1.

Three classes were examined for normality : water, urban, and vegetation. The results of analysis are listed in Section 5.1.

4.0 MULTIVARIATE SCENE GENERATION

4.1 Overview of Synthetic Image Generator

The robustness of the maximum likelihood classifier was examined by classifying a series of multivariate synthetic images, which ranged from normal to non-normal in their distribution. Every multivariate synthetic image contained three different classes, each of which was generated from the covariance matrix (Σ) and intraband correlation (ρ) of real LANDSAT TM class data.. For example, a class scene representing water in a multivariate synthetic image would have been generated using the Σ and ρ obtained from a water sample of a LANDSAT TM image.

The generation of the synthetic class scenes began with the creation of a matrix which represented the spectral, and to some degree spatial relationship that each pixel vector had to one another for a given block of LANDSAT TM image data¹⁶. This study modeled a 5 pixel x 5 pixel x 6 band portion of LANDSAT TM image data as illustrated in Figure 7a (Note: each pixel in Figure 7a and 7b represents a six band column vector). Figure 7b presents a high level look of how the matrix, which will be referred to as the synthetic image matrix, presented the spectral and spatial relationship of the 25 pixels in Figure 7a. Every two way pixel vector combination of Figure 7a is statistically modeled as a function of Σ and ρ (Figure 7b). Elements on the diagonal represent a pixel's spectral and intraband correlation to itself. Since the intraband correlation of a pixel vector to itself is one, the diagonal elements of the synthetic image matrix were represented only by the

covariance matrix. Elements off the diagonal were modified by ρ_i to reflect the intraband correlation between two specific pixel. Pixels with the same intraband correlation were arranged in a diagonal format paralleling the main diagonal elements (Figure 7b). For simplicity, the interband correlation values of the synthetic image data was assumed to be the same the intraband correlation values.

The synthetic image matrix used in this study measured 150 x 150. A more detailed representation of it can be seen in Figure 8. A further description of the how the intraband correlation coefficients and covariance matrix were determined for the synthetic image matrix will be shown in the next section.

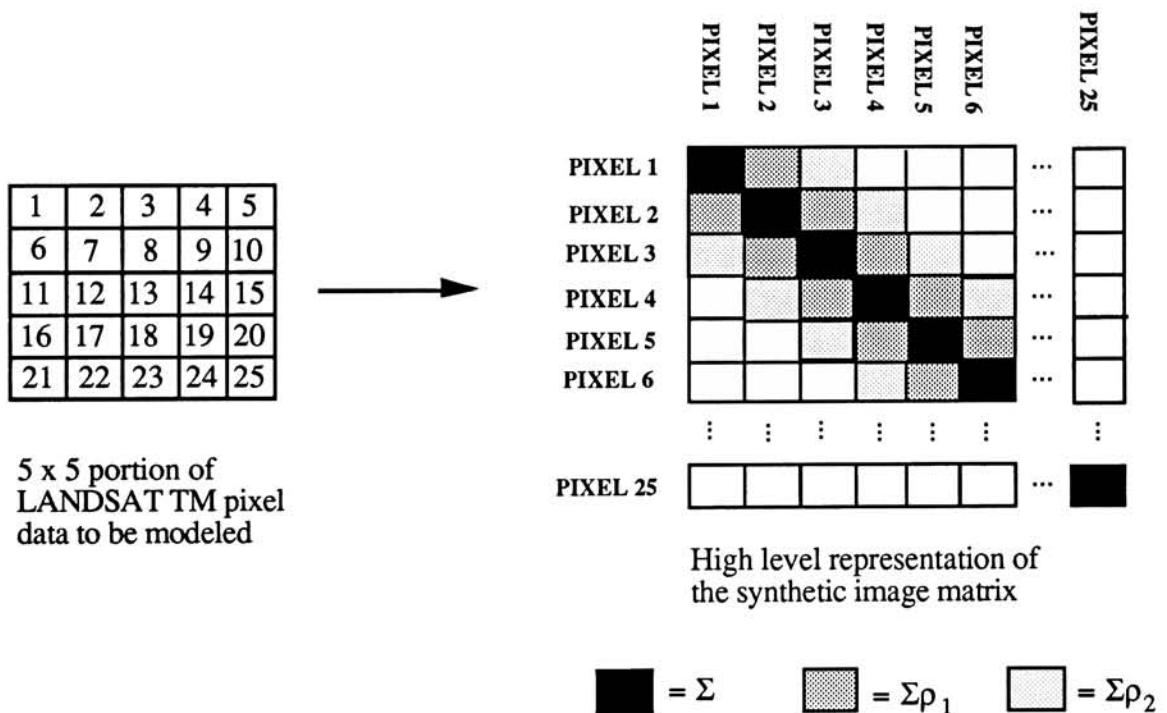


Figure 7 High level view of synthetic image generator

Note: each pixel represents a six band column vector

Synthetic image generation began by determining the intraband correlation (ρ) of the three LANDSAT TM scenes selected to be statistically reproduced for this study. The values for ρ were determined in the following manner.

First, an unsupervised classification was conducted on all imagery that contained scenes of interest. All unsupervised classifications were performed using ERDAS software. The classifications maps generated during this process were used as image masks to isolate the features of interest within a LANDSAT TM image on a band by band basis. The correlation algorithm developed for this study searched for continuous strings of class data that measured 50 pixels or longer. After locating strings of appropriate length, pixels contained in the string were grouped the into pixel pairs as shown in Figure 9.

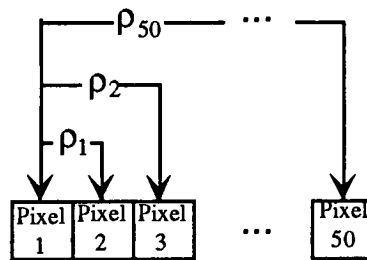


Figure 9 Pixel pairs for determining intraband correlation

The delta spacing for the intraband correlation pixel pairs ranged from 0 (ρ_1) to 49 (ρ_{50}). Intraband correlation was established for both the column and row directions as can be seen in Appendix A of this document.

The formula used to obtain intraband correlation is show in (4.1)¹⁷. Here X and Y make up one of the pixel pairs whose correlation is being determined. For example, if the intraband correlation for ρ_2 was being calculated (Figure 9), for a given band, pixel pairs

with a delta separation of 1 would be collected from the class of interest. The X variable would represent pixel 1 and the Y variable would represent pixel 3.

$$\rho_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y})}{\left\{ \sum_{i=1}^n (x_i - \bar{X})^2 \right\}^{\frac{1}{2}} \left\{ \sum_{i=1}^n (y_i - \bar{Y})^2 \right\}^{\frac{1}{2}}} \quad (4.1)$$

Once the intraband correlation values had been determined as a function of class feature and spacing between pixel pairs, the ρ values were used to modify the coefficients of the covariance matrix to represent the spatial information in the synthetic scene. Pixel pairs with an integer distance value between the pair (e.g. ρ_2 in Figure 9) simply used the intraband correlation generated by the formula in (4.1). Pixel pairs whose distance was represented by a real value (Figure 10) used an interpolation process to determine the intraband correlation of the pair.

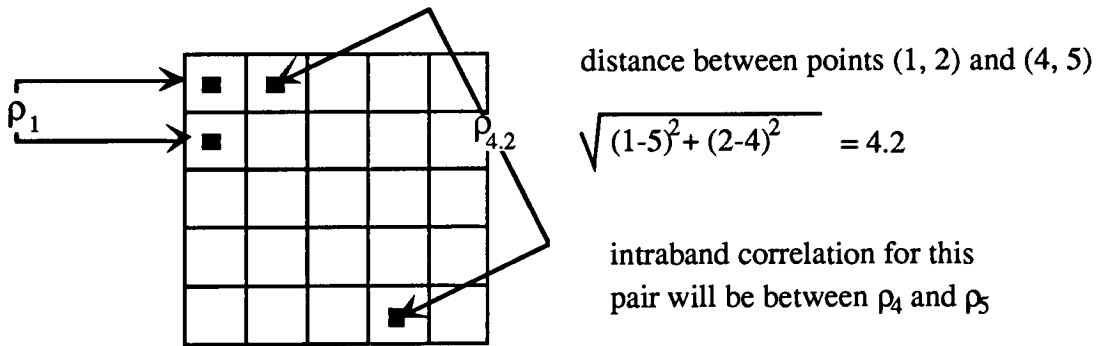


Figure 10 Pixel pairs used to determine intraband correlation

4.3 Sample Covariance

The sample covariance describes the how different bands vary with each other in relation to the means of their respective bands. This is expressed mathematically as:

$$\text{Cov}_{QR} = \frac{\sum_{i=1}^k (Q_i - x_Q)(R_i - x_R)}{k} \quad (4.2)$$

where Q and R are image data values from two different bands, x represents the mean of the respective bands and k indicates the number of pixels in a band¹⁸.

Covariance coefficients used in this study were obtained with ERDAS software. They can be found in Appendix B along with the class means.

4.4 Description of Multivariate Synthetic Image Generator

The theory behind this study's image generating algorithm is derived by T.W. Anderson in his book An Introduction to Multivariate Statistical Analysis (1958). Anderson describes a method that can be used to create a new multivariate normal population of a particular mean and spread, from another multivariate normal population which possesses a different mean and spread¹⁹. A univariate example of this theory is described below. If z represents a standardized normal variable

$$z \approx n(0,1) \quad (4.3)$$

a new population can be created from z by adjusting the mean and spread of the z

$$x = \mu + \sigma z. \quad (4.4)$$

The resulting in a new population, x , can be expressed as follows:

$$x \approx n(\mu, \sigma^2). \quad (4.5)$$

The spread of the x population data is created by multiplying the z values from a normal population by the square root of the desired variance (σ^2) i.e. the standard deviation (σ). A similar theory could be applied to a multivariate model. If \mathbf{Z} represents a vector of length k comprised of standardized normal variates

$$\mathbf{Z} \approx n(\mathbf{O}, \mathbf{I}) \quad (4.6)$$

then the multivariate expression of (4.4) could be expressed as

$$\mathbf{X} = \mu + \mathbf{P}\mathbf{Z}. \quad (4.7)$$

This new population composed of \mathbf{X} variables could be expressed as

$$\mathbf{X} \approx n(\mu, \Sigma) \quad (4.8)$$

where: μ = a vector of length k

\mathbf{P} = a $k \times k$ matrix.

Σ = the covariance matrix

The value of \mathbf{P} , capable of satisfying the expression in (4.7), can be derived by decomposing Σ . Σ is a positive definite matrix with eigenvalues $\lambda_1, \lambda_2, \lambda_3 \dots \lambda_k$ and corresponding column eigenvectors $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3 \dots \mathbf{B}_k$ each of length k . For the decomposition of Σ , the eigenvalues will be represented as a $k \times k$ matrix (λ), with eigenvalues along the diagonal and off diagonal elements equaling zero. Eigenvectors will also be treated as a $k \times k$ matrix (\mathbf{B}) of the eigenvectors presented as column elements. The decomposition of Σ begins with a well known property of eigenvectors (4.9):

$$\mathbf{B}'\Sigma\mathbf{B} = \lambda. \quad (4.9)$$

Since the eigenvectors selected for this research were of unit length $\mathbf{B}\mathbf{B}' = \mathbf{I}$, the left side of expression (4.9) may be multiplied by \mathbf{B} and the right may be multiplied by \mathbf{B}' . After this process the expression (4.9) becomes:

$$BB'\Sigma BB' = B\lambda B' \quad (4.10)$$

which can be rewritten as

$$\Sigma = B\lambda B'. \quad (4.11)$$

λ can be decomposed further into $\lambda^*\lambda^*$, where λ^* is a $k \times k$ matrix with $\sqrt{\lambda_i}$ along the diagonal and zeros off the diagonal. By substituting $\lambda^*\lambda^*$ for λ expression (4.11) becomes

$$\Sigma = B\lambda^*\lambda^*B' \quad (4.12)$$

If $P = B\lambda^*$, then the expression (4.12) can be rewritten as

$$\Sigma = PP'. \quad (4.13)$$

Having determined the value of P for expression (4.7), the population generated from (4.7) can be described as follows:

$$X \approx n(\mu, PP'). \quad (4.14)$$

An overview of how the value P fits into the synthetic image generator can be seen in Figure 11.

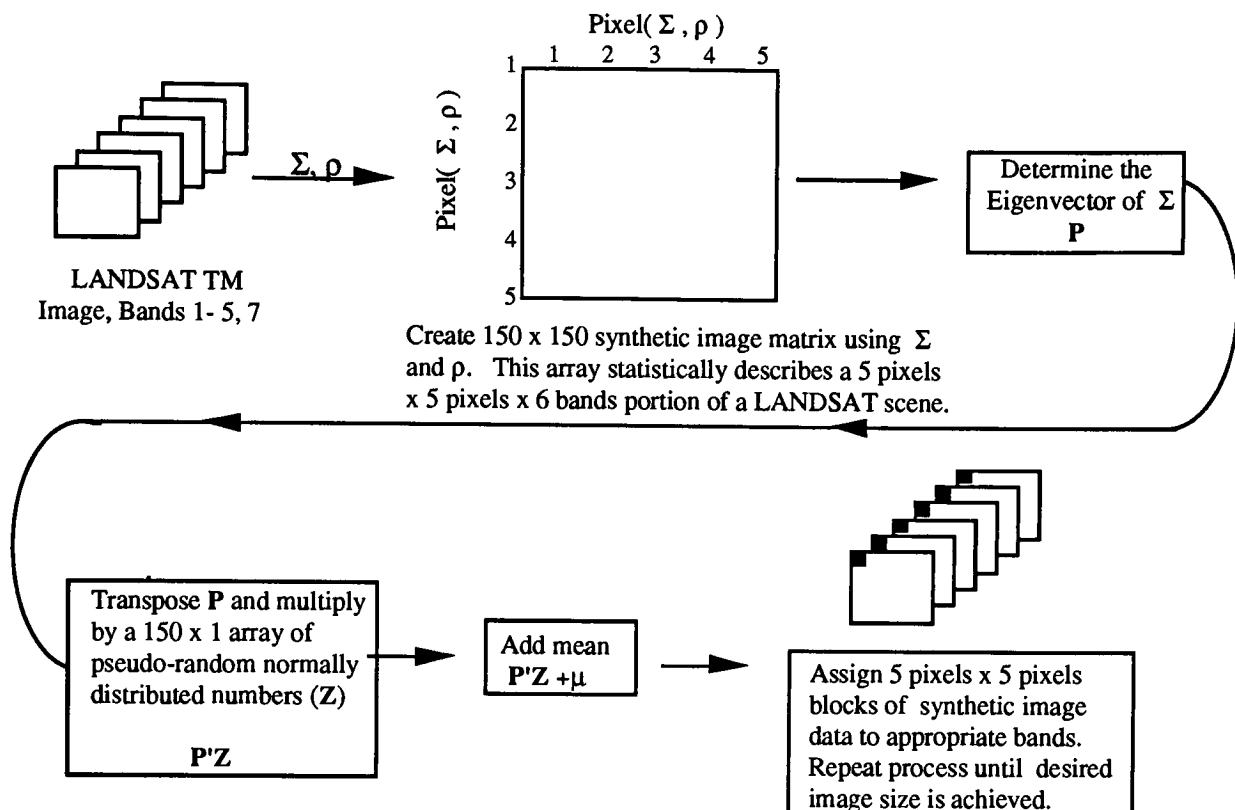


Figure 11 Overview of synthetic image generator.

4.5 Description of the Imagery Created by the Synthetic Image Generator

Synthetic image generation began by creating 256 pixel x 256 pixel blocks of class samples (e.g. water, vegetation, urban) as described in Section 4.4. After the class samples were produced, they were placed next to other synthetic class samples to form rough examples of LANDSAT TM imagery. In total, four different sets of image data were created to test the robustness of the maximum likelihood classifier, each containing three different class samples. Bands 5, 3 and 2 of these synthetic LANDSAT TM images are shown in Plates 1 through 19. The first set of images (Plates 1 through 4), spectrally represent urban, vegetation and water classes. Plate 1 in the series depicts a normally distributed image, with the progression towards non-normality shown in Plates 2 through 4. The description for how non-normality was induced is seen in Section 4.6.

The second set of images represent three different classes of water. These class features have similar means, but different covariances. They are shown in Plates 5 through 8, and mimic the same progression towards non-normality that the first image set followed.

The third image set, shown in Plates 9 through 15, contains a series of hybrid test samples created using the means of three water classes and the covariances of three vegetation scenes. These samples also range from normal to non-normal distributions.

The final set of synthetic images created for this study is shown in Plates 16 through 19 contain three types of vegetation: deciduous trees, grass and agriculture. All of the covariance matrices and class means were selected from the same LANDSAT image. The images generated followed the same range normal to non-normal as mentioned above.

4.6 Inducing Non-normality in Synthetic Images

Non-normality was induced in images for this study by corrupting a specific percentage of pixels from the normally distributed image. The pixels were selected at random, and were corrupted by adding 2, 3 or 5 standard deviations to the gray level values of 20%, 40%, or 60 % of the pixel data. The matrix in Tables 4.1, 4.2 and 4.3 shows the specific parameters used per image set: Figure 12 illustrates how the distributions of the synthetic sample changed as non-normality was induced.

	20% of Image Data	40% of Image Data	60% of Image Data
2 Standard Deviations	X		
3 standard Deviations		X	
5 Standard Deviations			X

Table 4.1 Non-normal Samples Generated for Set One

	20% of Image Data	40% of Image Data	60% of Image Data
2 Standard Deviations	X		
3 standard Deviations		X	
5 Standard Deviations			X

Table 4.2 Non-normal Samples Generated for Set Two

	20% of Image Data	40% of Image Data	60% of Image Data
2 Standard Deviations	X	X	X
3 standard Deviations	X	X	X
5 Standard Deviations			

Table 4.3 Non-normal Samples Generated for Set Three

	20% of Image Data	40% of Image Data	60% of Image Data
2 Standard Deviations	X		
3 standard Deviations		X	
5 Standard Deviations			X

Table 4.4 Non-normal Samples Generated for Set Four

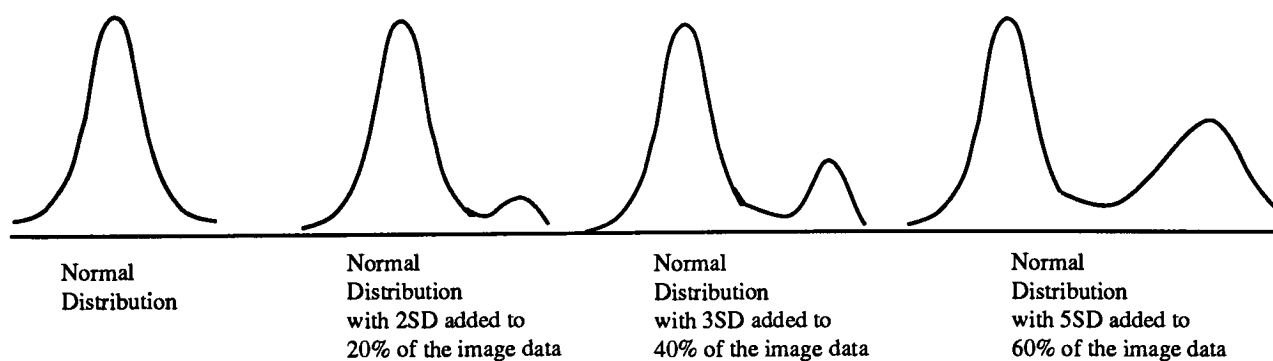


Figure 12 Distributions of non-normal data

IMAGE PLATES



Image Set 1

Synthetic image generated with all three classes normally distributed

Top left patch represents urban
Top right patch represents vegetation
Bottom middle patch represents water

PLATE 1

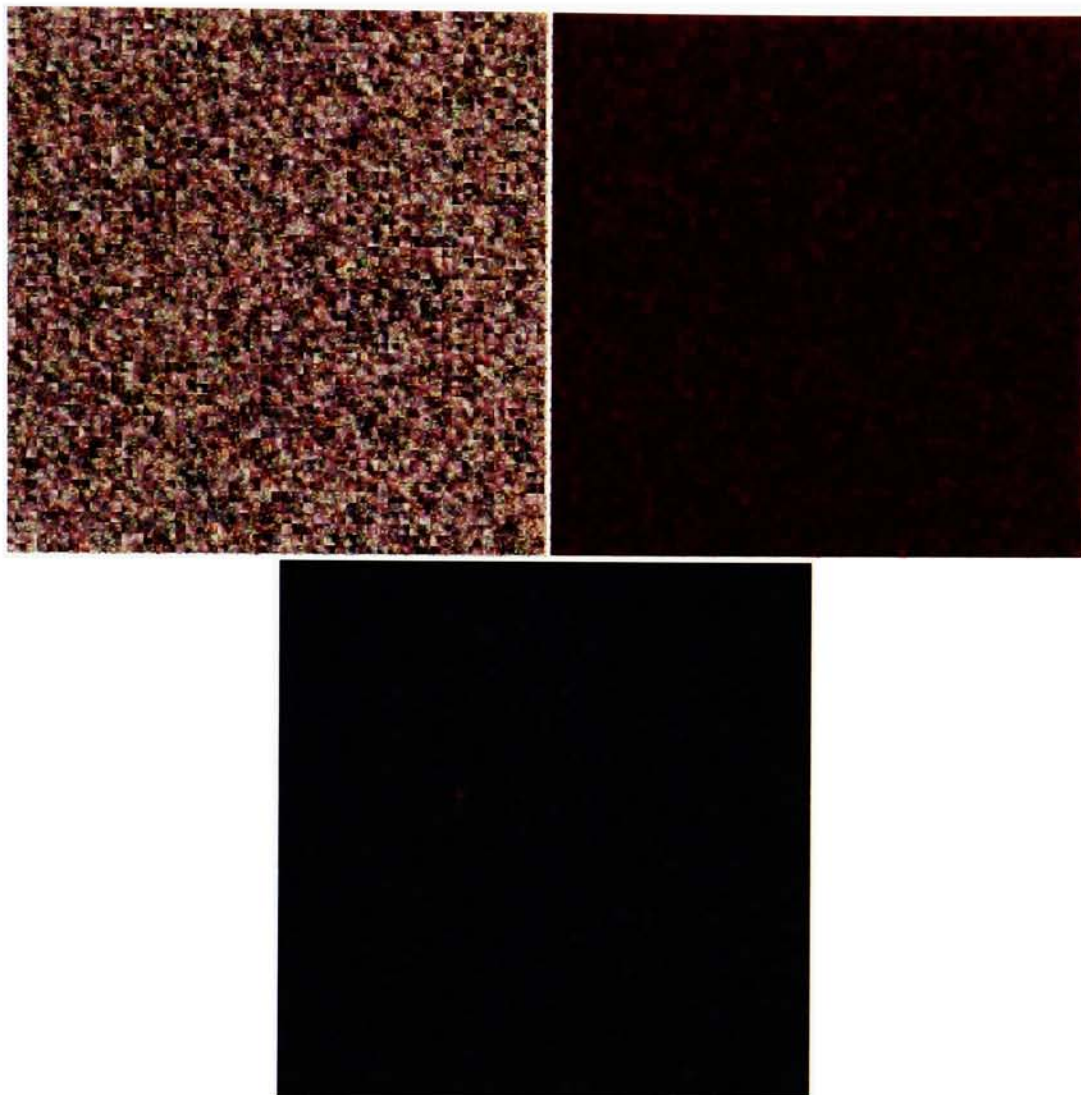


Image Set 1

Synthetic image generated with 20% of the image data corrupted by 2 standard deviations

Top left patch represents urban
Top right patch represents vegetation
Bottom middle patch represents water

PLATE 2



Image Set 1

Synthetic image generated with 60% of the image data corrupted by 3 standard deviations

Top left patch represents urban
Top right patch represents vegetation
Bottom middle patch represents water

PLATE 3

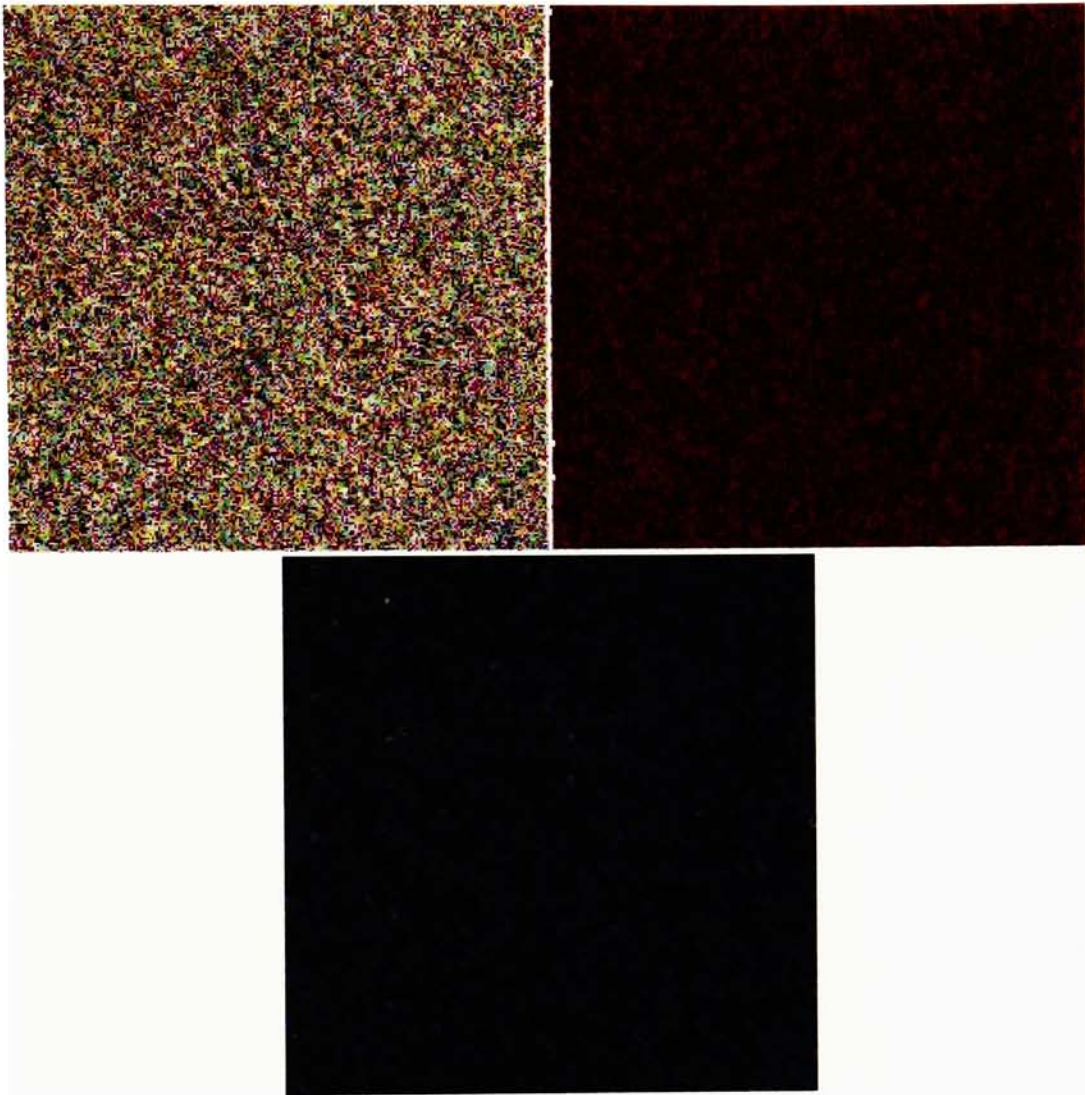


Image Set 1

Synthetic image generated with 60% of the image data corrupted by 5 standard deviations

Top left patch represents urban
Top right patch represents vegetation
Bottom middle patch represents water

PLATE 4

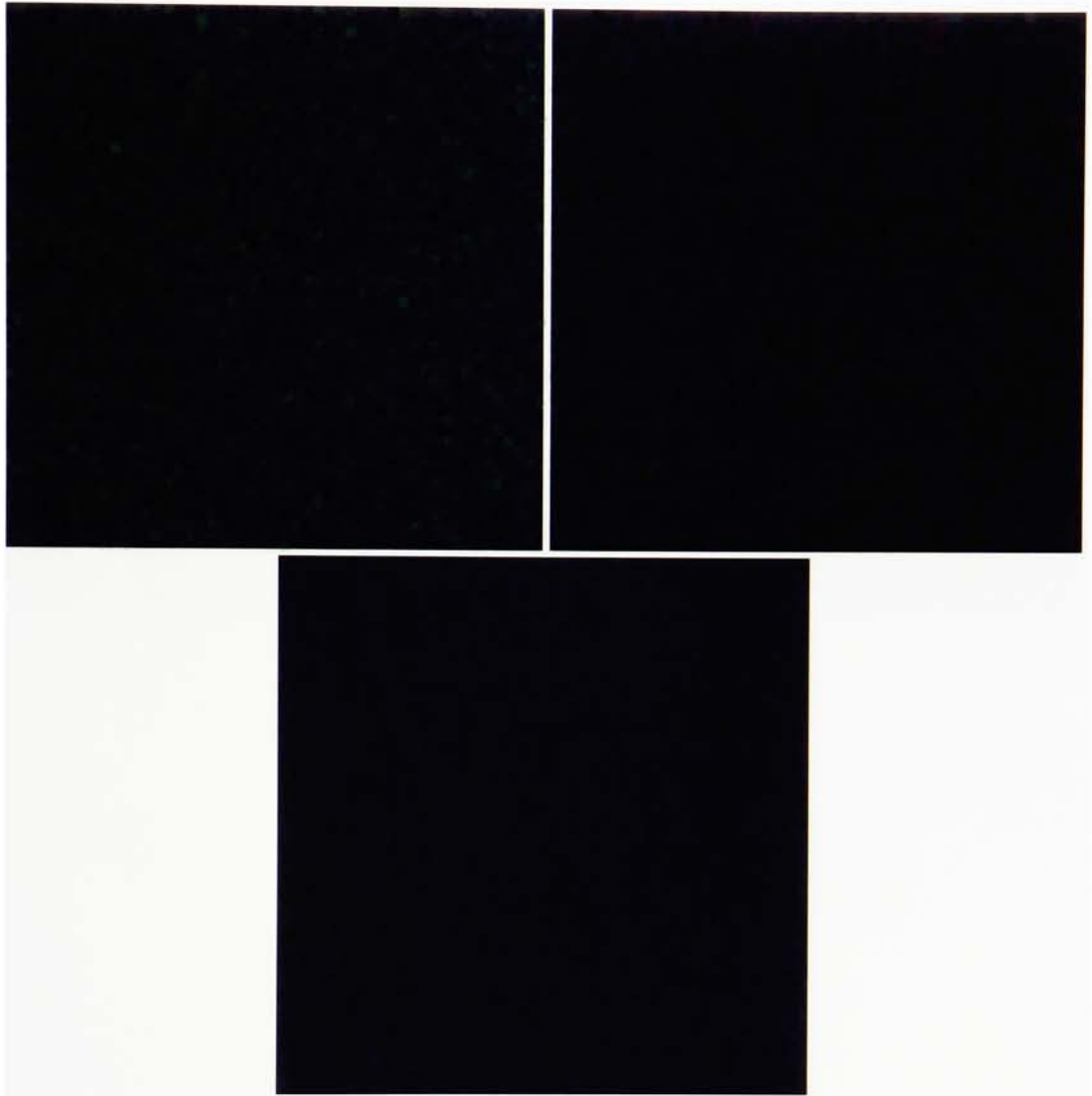


Image Set 2

Synthetic image generated with all three classes normally distributed

Top left patch represents water sample from Charleston NC

Top right patch represents water sample from Hartford CT

Bottom middle patch represents water sample from Washington DC

PLATE 5

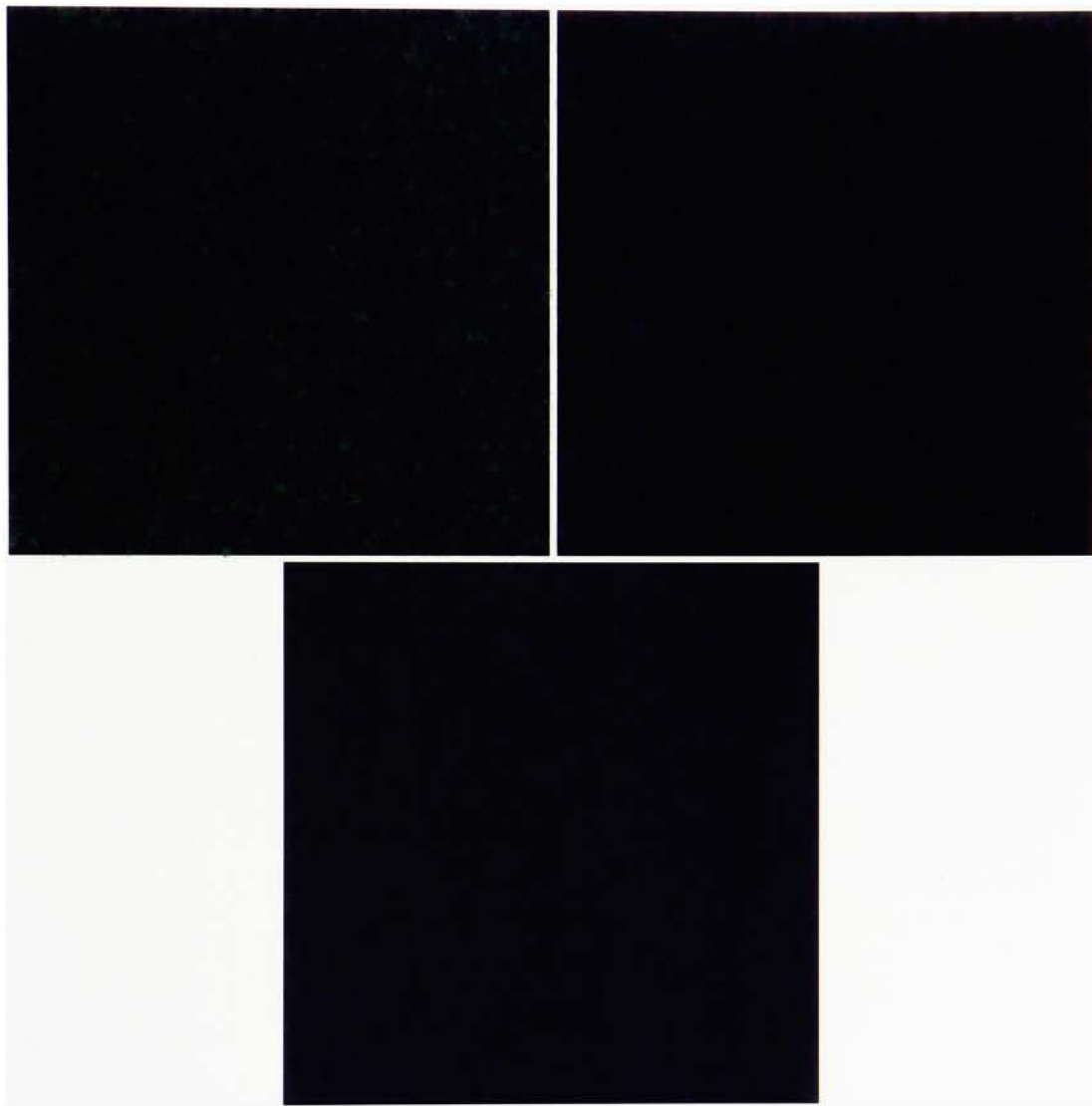


Image Set 2

Synthetic image generated with 20% of the image data corrupted by 2 standard deviations

Top left patch represents water sample from Charleston NC
Top right patch represents water sample from Hartford CT
Bottom middle patch represents water sample from Washington DC

PLATE 6

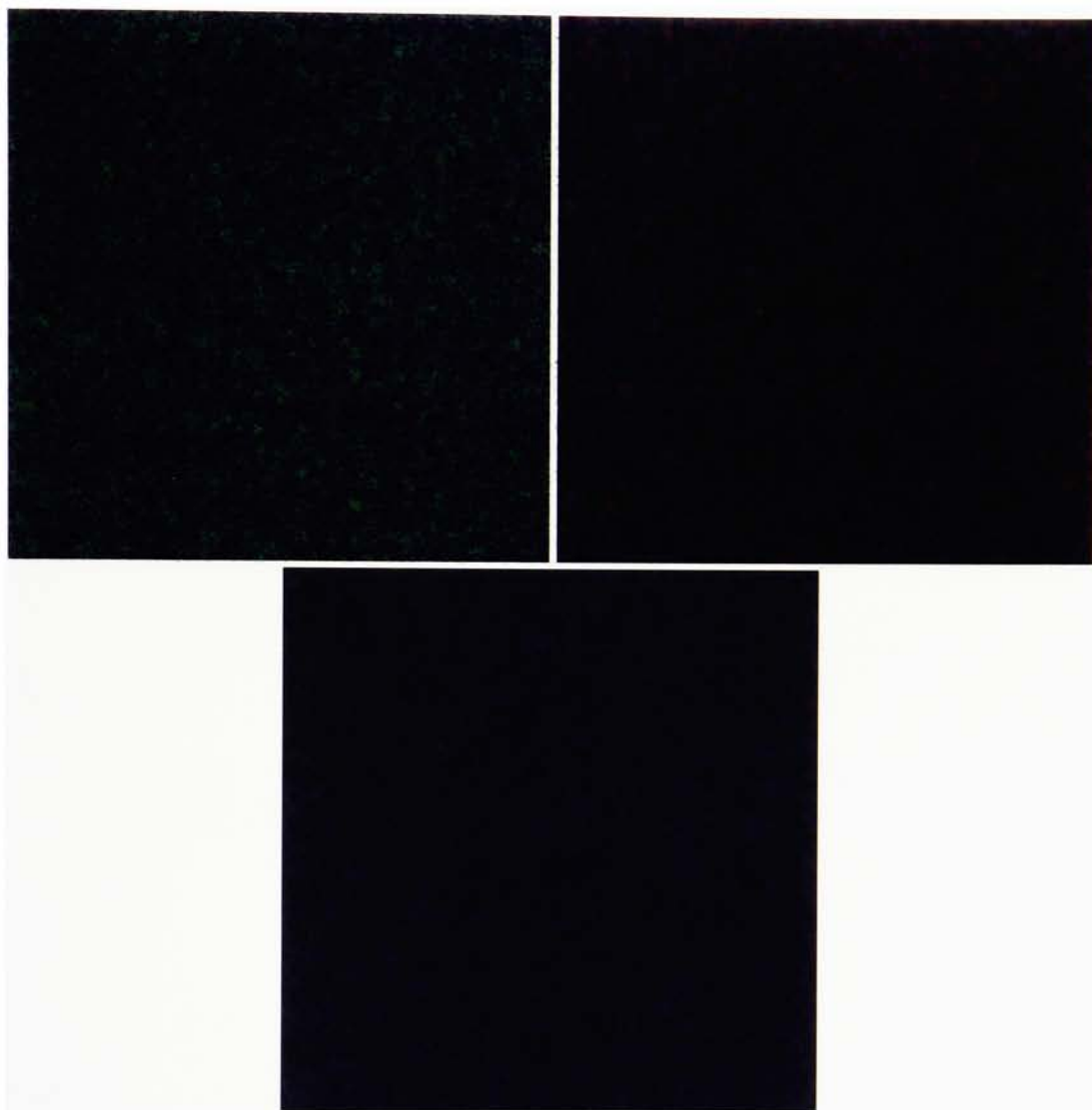


Image Set 2

Synthetic image generated with 60% of the image data corrupted by 3 standard deviations

Top left patch represents water sample from Charleston NC
Top right patch represents water sample from Hartford CT
Bottom middle patch represents water sample from Washington DC

PLATE 7

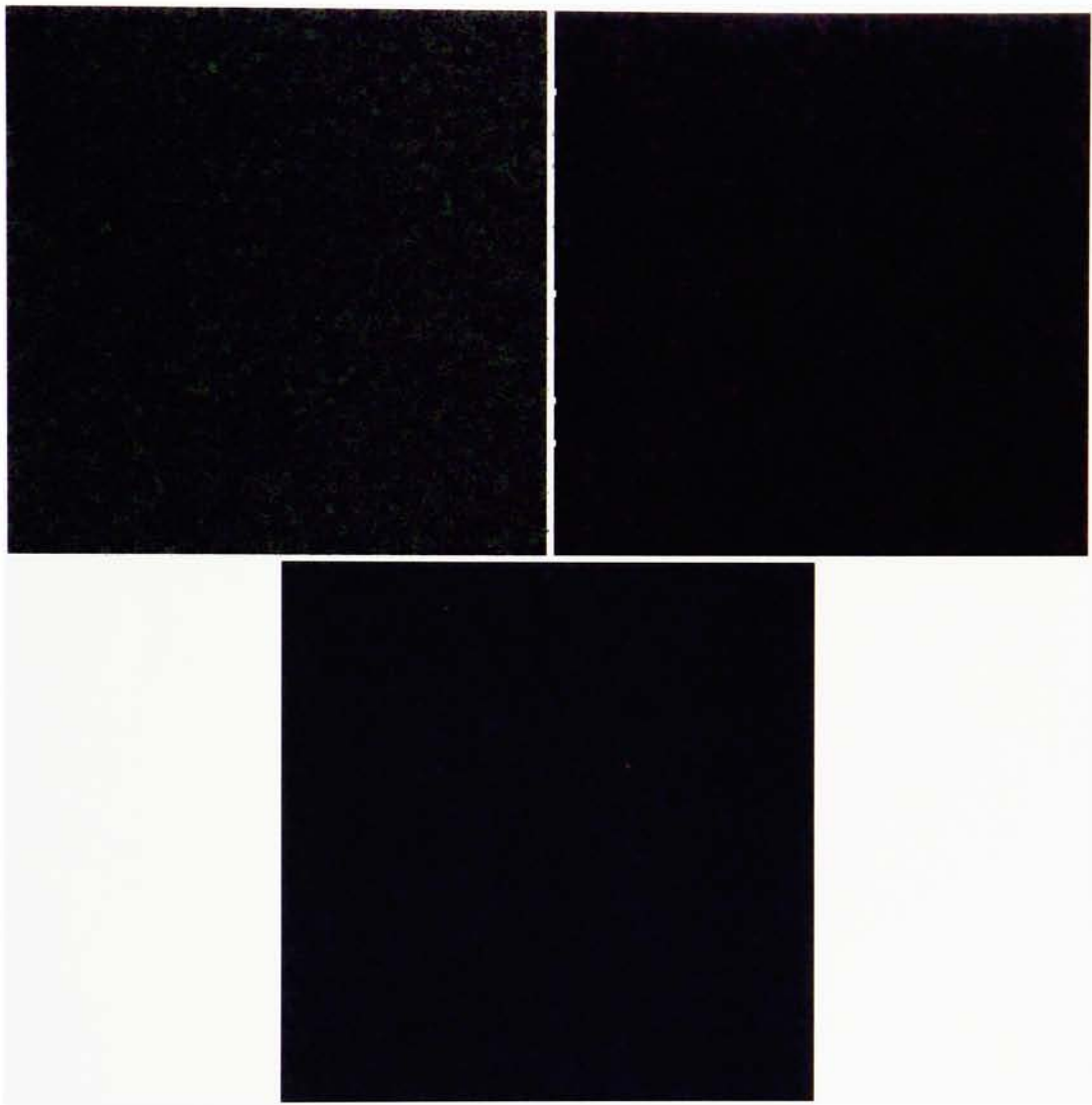


Image Set 2

Synthetic image generated with 60% of the image data corrupted by 5 standard deviations

Top left patch represents water sample from Charleston NC
Top right patch represents water sample from Hartford CT
Bottom middle patch represents water sample from Washington DC

PLATE 8

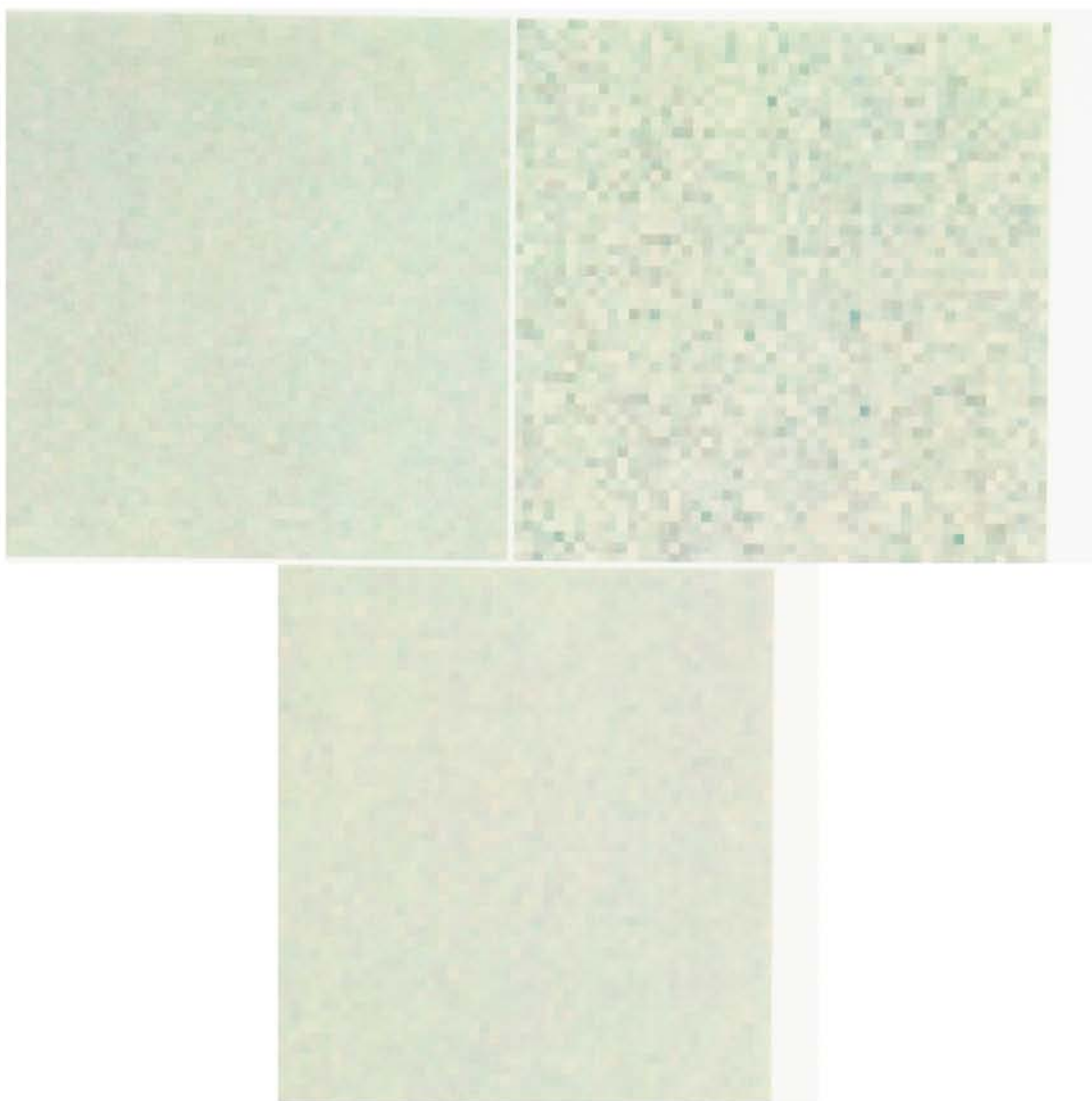


Image Set 3

Synthetic image generated with all three classes normally distributed

Imagery was created using a $\mu_{\text{water}} + 128$ digital counts, and a $\Sigma_{\text{vegetation}}$. The following lists the source of the μ_{water} and $\Sigma_{\text{vegetation}}$:

Top left patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from Baltimore MD

Top right patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from San Francisco CA

Bottom middle patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from Washington DC

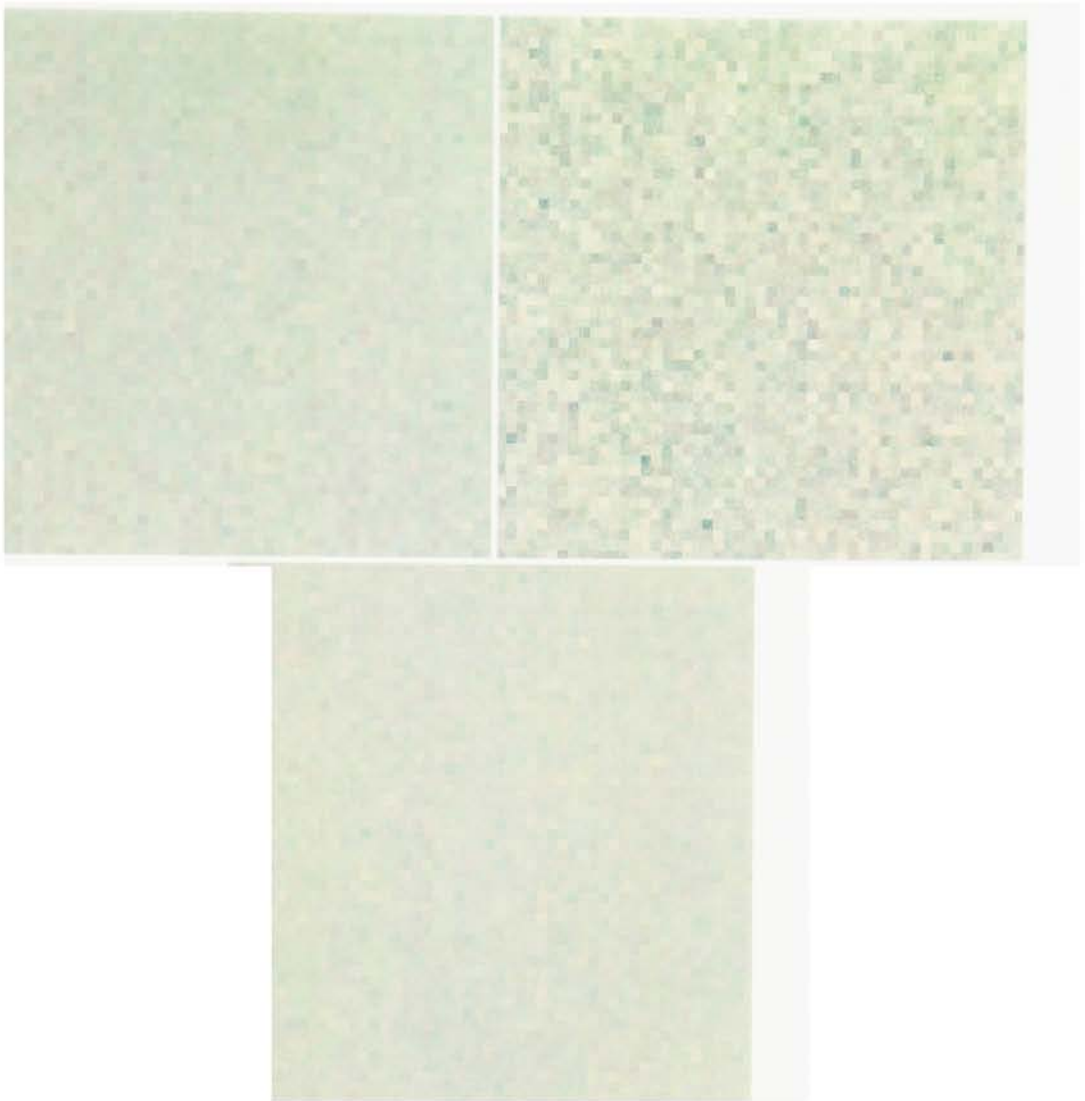


Image Set 3

Synthetic image generated with 20% of the image data corrupted by 2 standard deviations

Imagery was created using a $\mu_{\text{water}} + 128$ digital counts, and a $\Sigma_{\text{vegetation}}$. The following lists the source of the μ_{water} and $\Sigma_{\text{vegetation}}$:

Top left patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from Baltimore MD

Top right patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from San Francisco CA

Bottom middle patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from Washington DC

PLATE 10

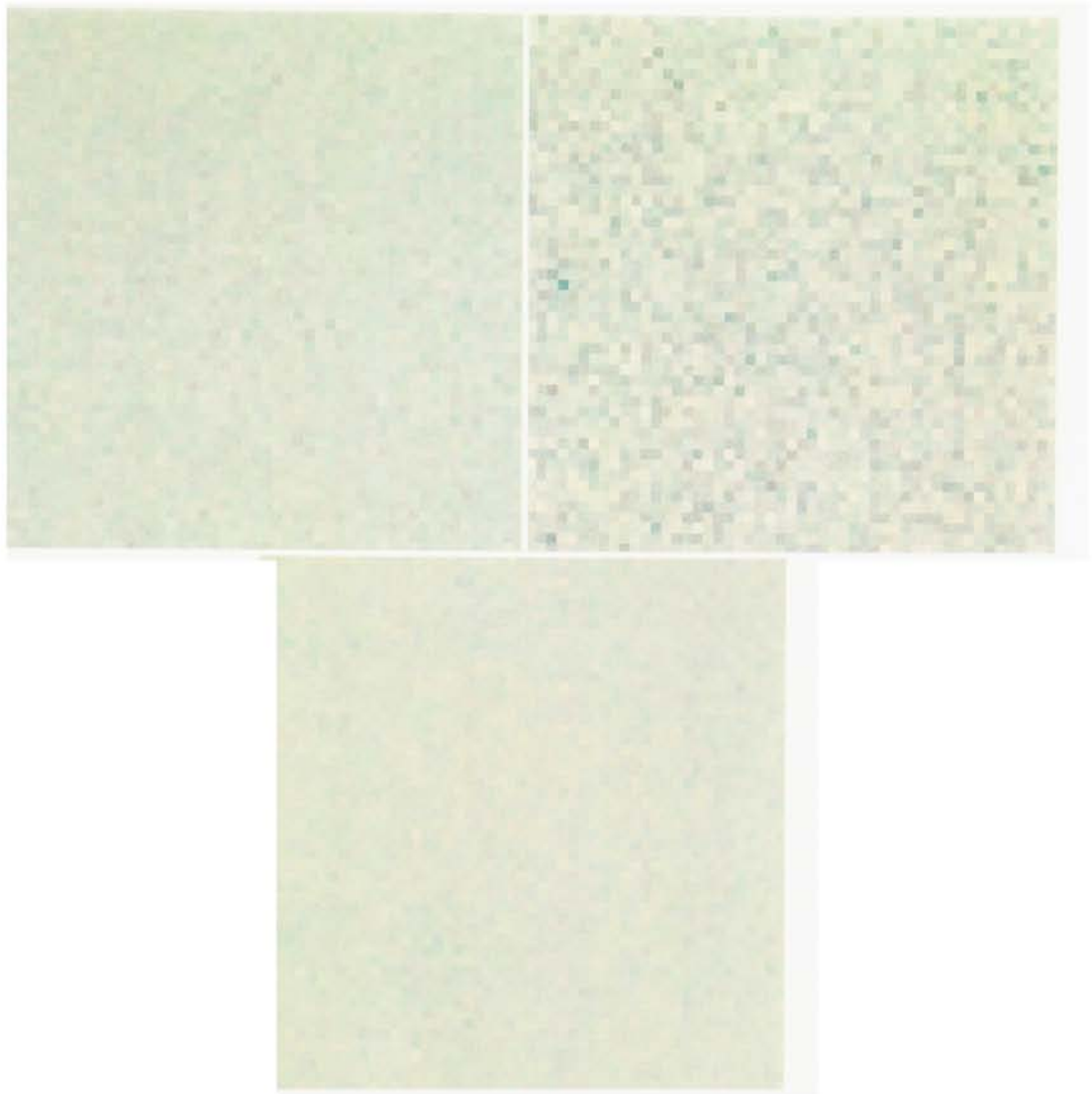


Image Set 3

Synthetic image generated with 40% of the image data corrupted by 2 standard deviations

Imagery was created using a $\mu_{\text{water}} + 128$ digital counts, and a $\Sigma_{\text{vegetation}}$. The following lists the source of the μ_{water} and $\Sigma_{\text{vegetation}}$:

Top left patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from Baltimore MD

Top right patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from San Francisco CA

Bottom middle patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from Washington DC

PLATE 11

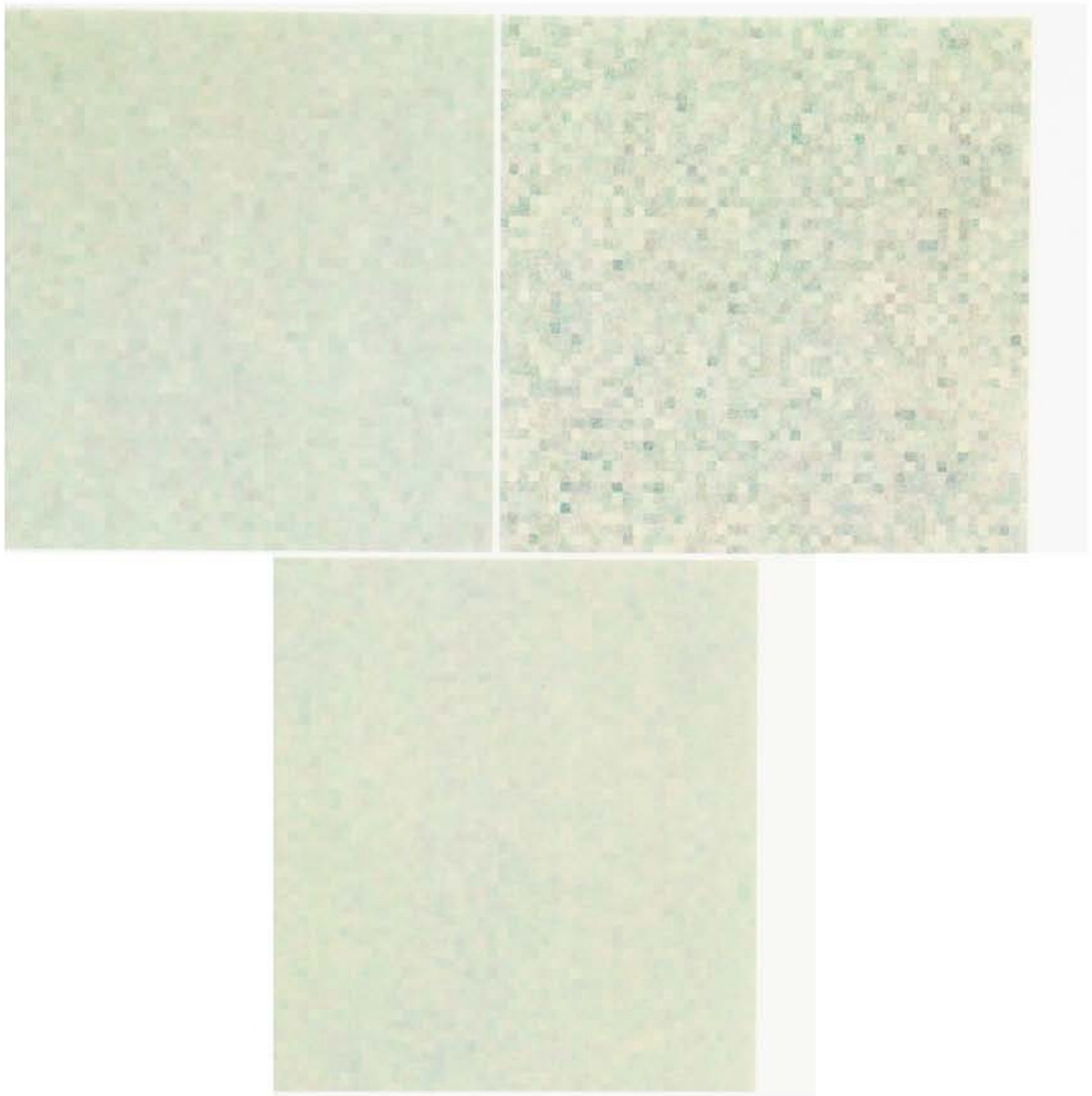


Image Set 3

Synthetic image generated with 60% of the image data corrupted by 2 standard deviations

Imagery was created using a $\mu_{\text{water}} + 128$ digital counts, and a $\Sigma_{\text{vegetation}}$. The following lists the source of the μ_{water} and $\Sigma_{\text{vegetation}}$:

Top left patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from Baltimore MD

Top right patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from San Francisco CA

Bottom middle patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from Washington DC

PLATE 12

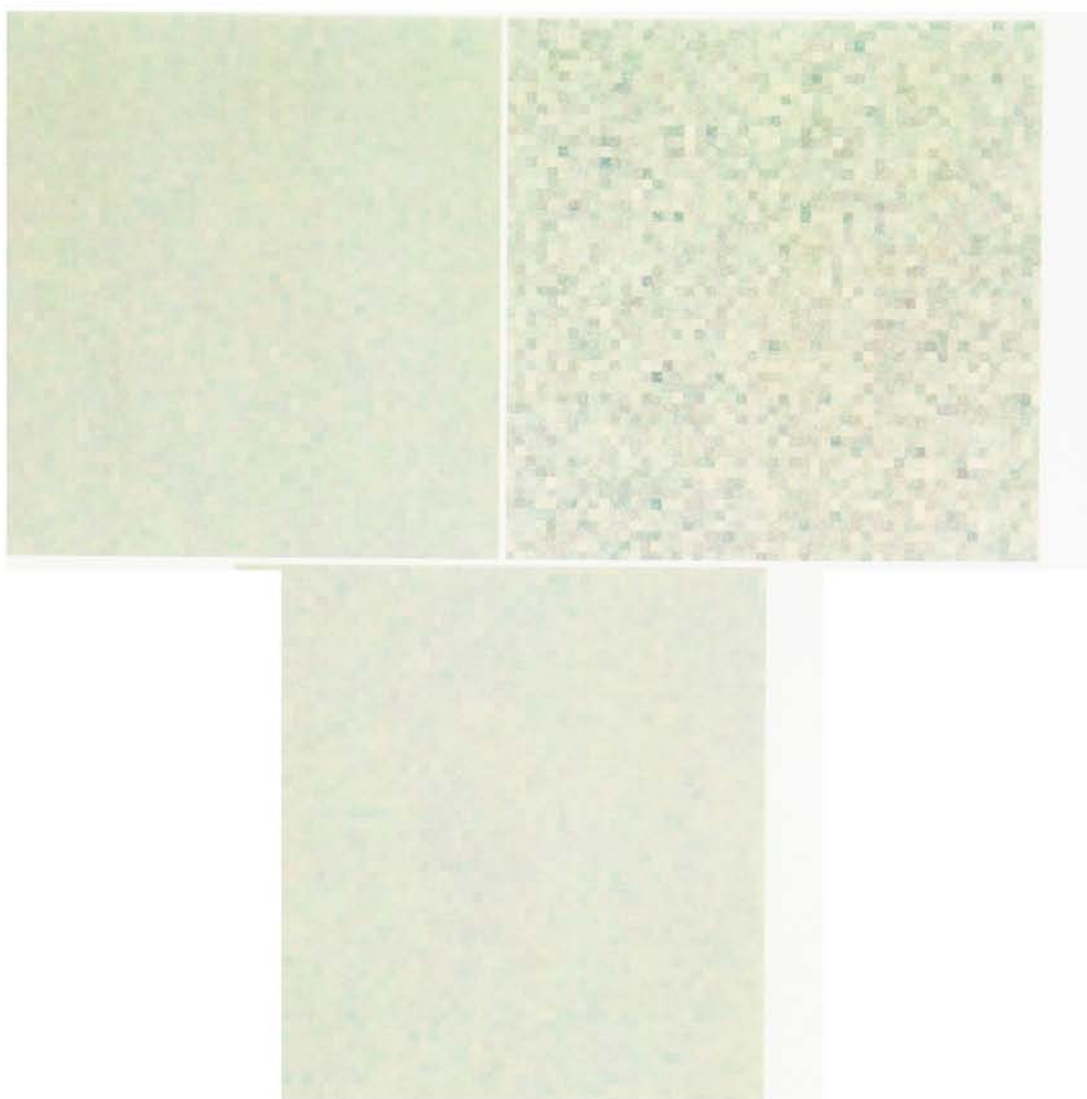


Image Set 3

Synthetic image generated with 20% of the image data corrupted by 3 standard deviations

Imagery was created using a $\mu_{\text{water}} + 128$ digital counts, and a $\Sigma_{\text{vegetation}}$. The following lists the source of the μ_{water} and $\Sigma_{\text{vegetation}}$:

Top left patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from Baltimore MD

Top right patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from San Francisco CA

Bottom middle patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from Washington DC

PLATE 13

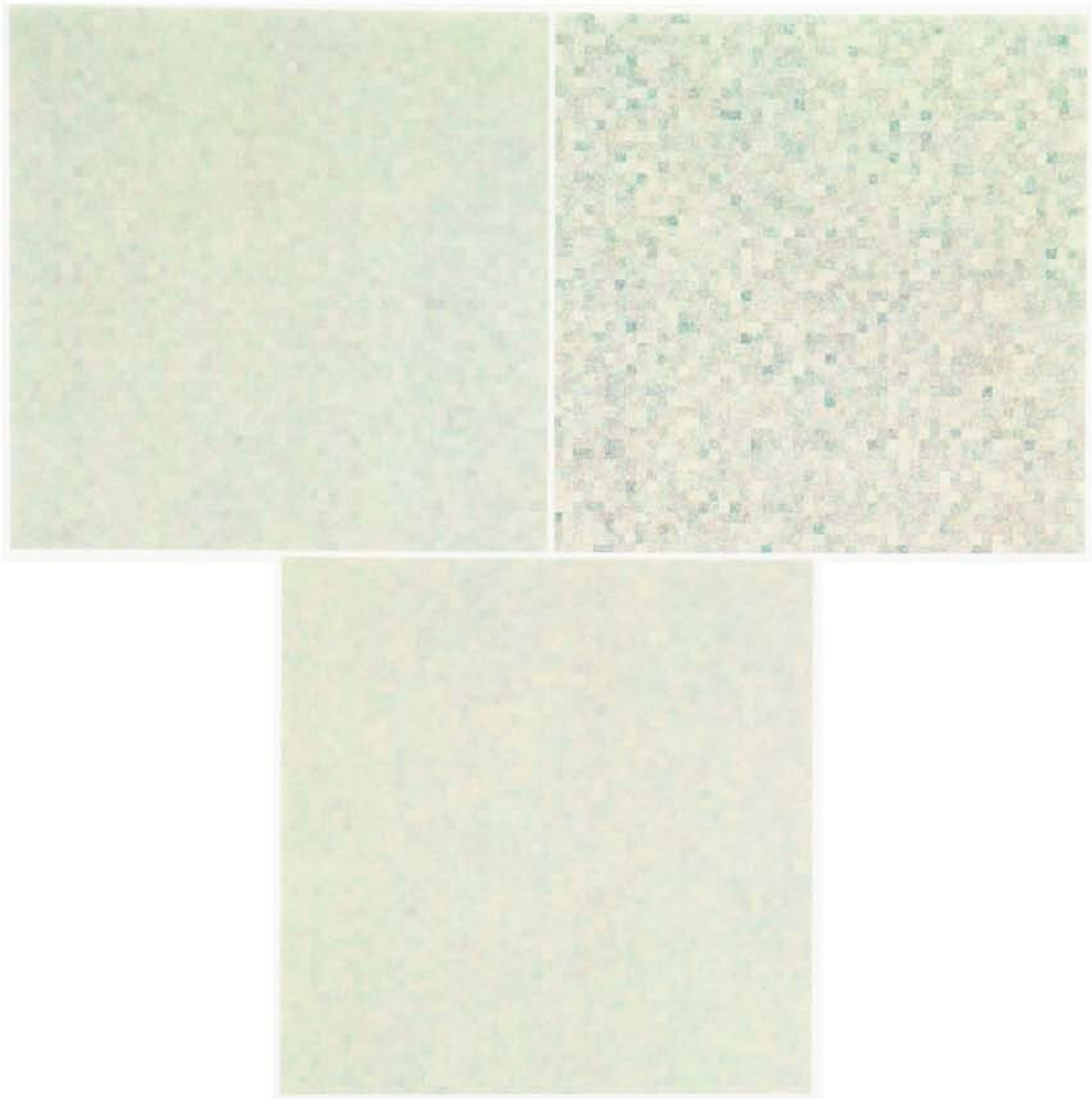


Image Set 3

Synthetic image generated with 40% of the image data corrupted by 3 standard deviations

Imagery was created using a $\mu_{\text{water}} + 128$ digital counts, and a $\Sigma_{\text{vegetation}}$. The following lists the source of the μ_{water} and $\Sigma_{\text{vegetation}}$:

Top left patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from Baltimore MD

Top right patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from San Francisco CA

Bottom middle patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from Washington DC

PLATE 14

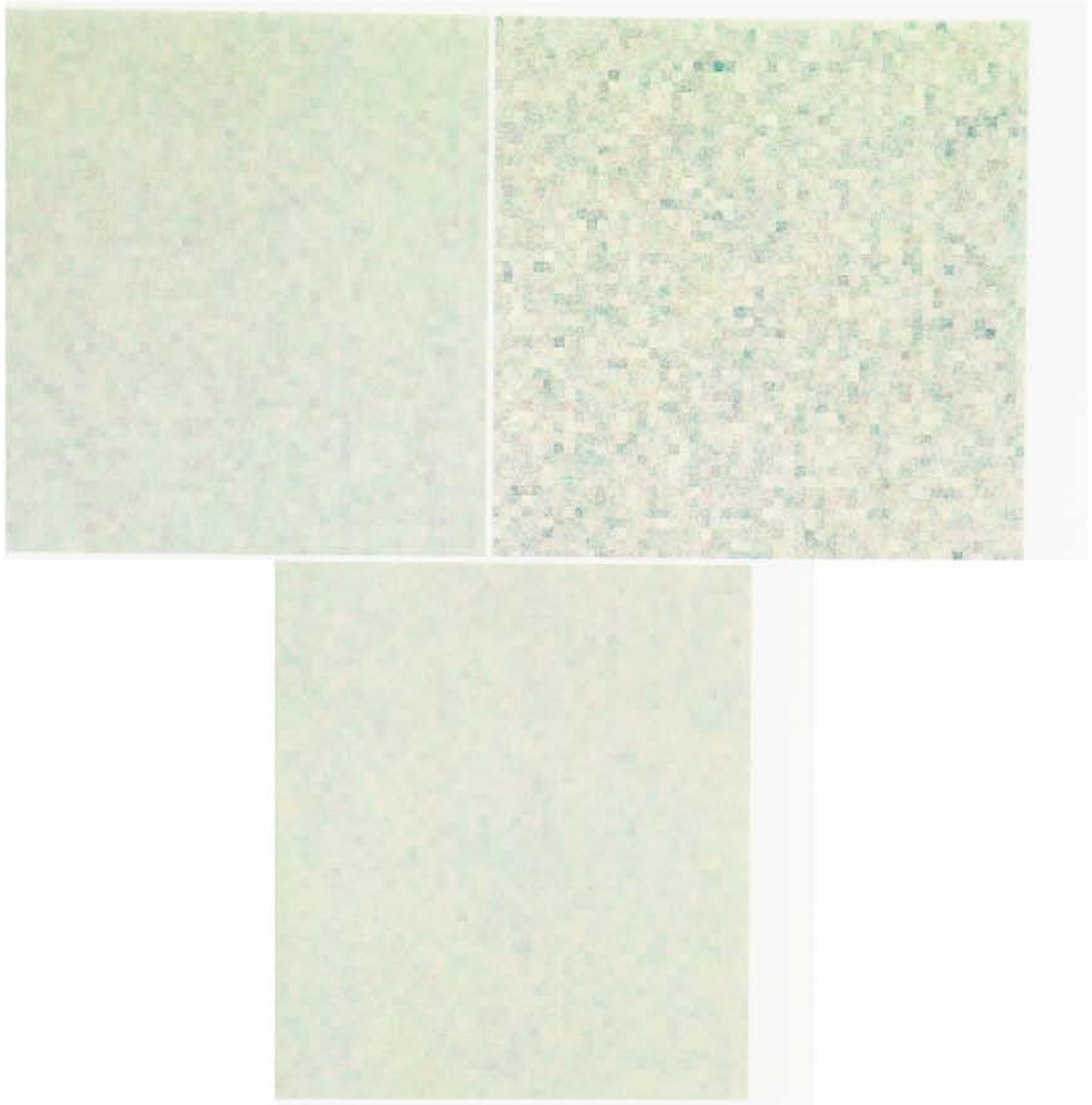


Image Set 3

Synthetic image generated with 60% of the image data corrupted by 3 standard deviations

Imagery was created using a $\mu_{\text{water}} + 128$ digital counts, and a $\Sigma_{\text{vegetation}}$. The following lists the source of the μ_{water} and $\Sigma_{\text{vegetation}}$:

Top left patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from Baltimore MD

Top right patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from San Francisco CA

Bottom middle patch μ_{water} and $\Sigma_{\text{vegetation}}$ extracted from Washington DC

PLATE 15

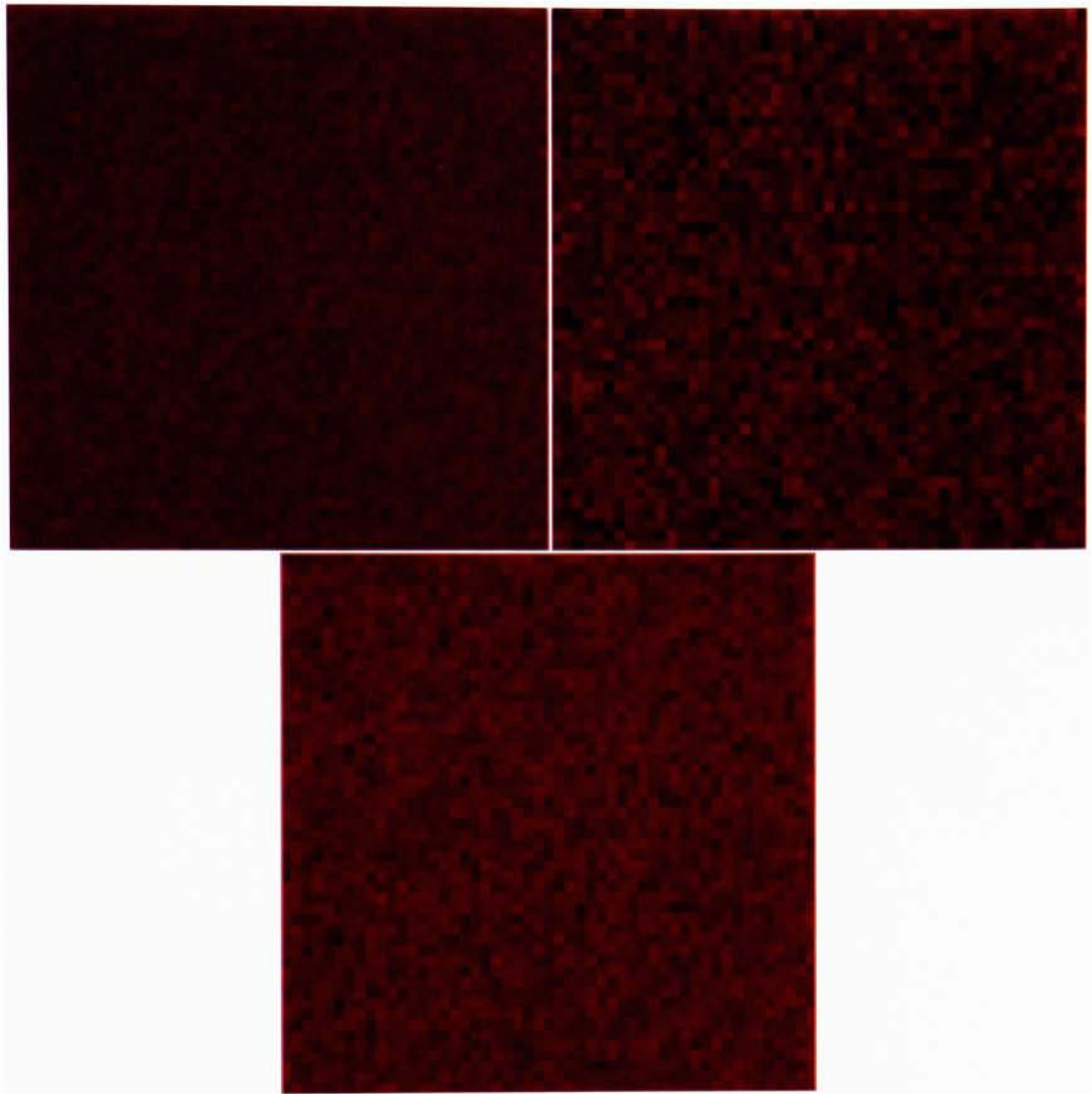


Image Set 4

Synthetic image generated with all three classes normally distributed

Three classes of vegetation, all selected from the same Rochester LANDSAT TM image

Top left patch μ and Σ obtained from grass area of Rochester image

Top right patch μ and Σ obtained from deciduous trees of Rochester image

Bottom middle patch μ and Σ obtained from agricultural area of Rochester image

PLATE 16

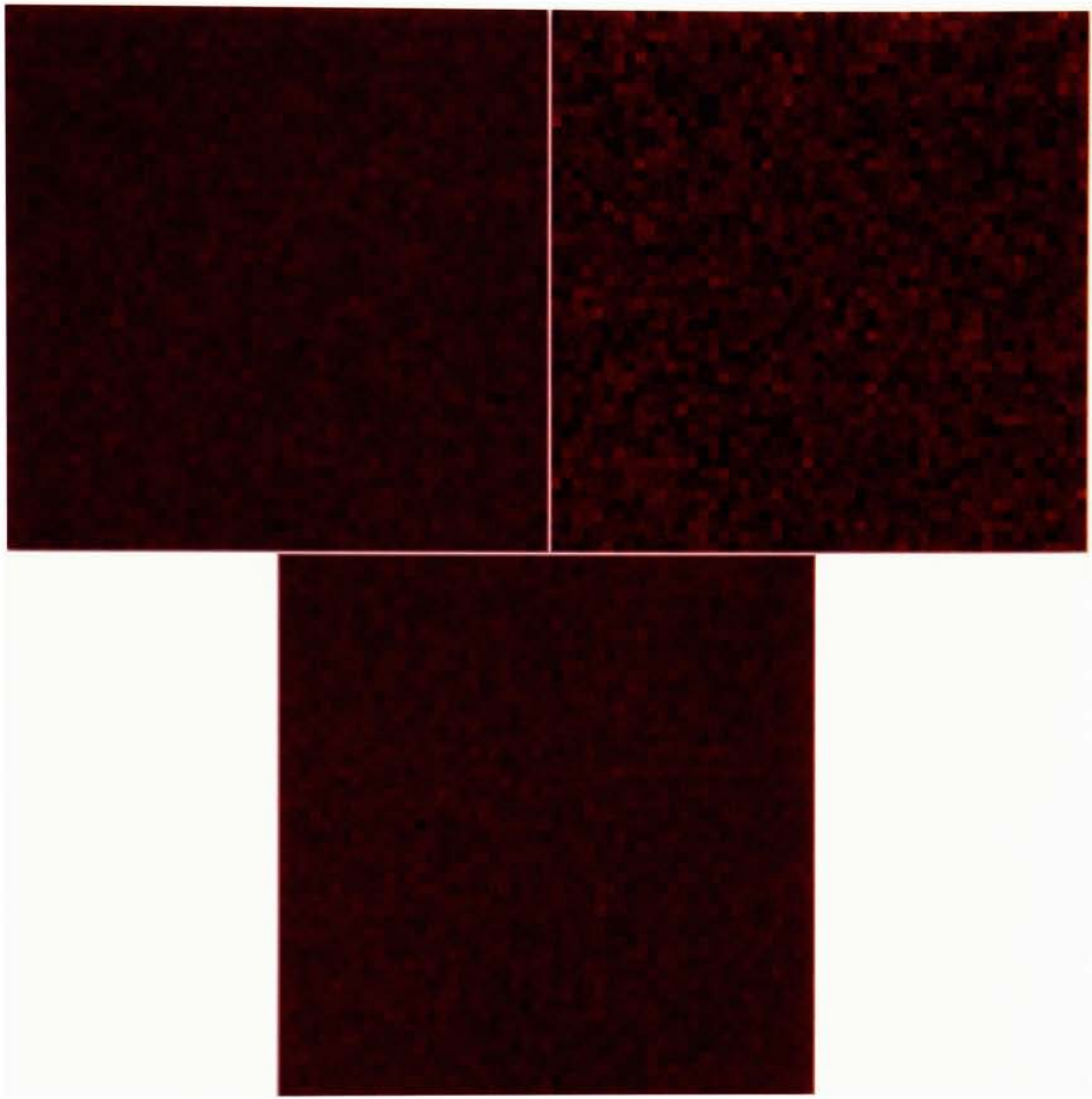


Image Set 4

Synthetic image generated with 20% of the image data corrupted by 2 standard deviations

Three classes of vegetation, all selected from the same Rochester LANDSAT TM image

Top left patch μ and Σ obtained from grass area of Rochester image

Top right patch μ and Σ obtained from deciduous trees of Rochester image

Bottom middle patch μ and Σ obtained from agricultural area of Rochester image

PLATE 17

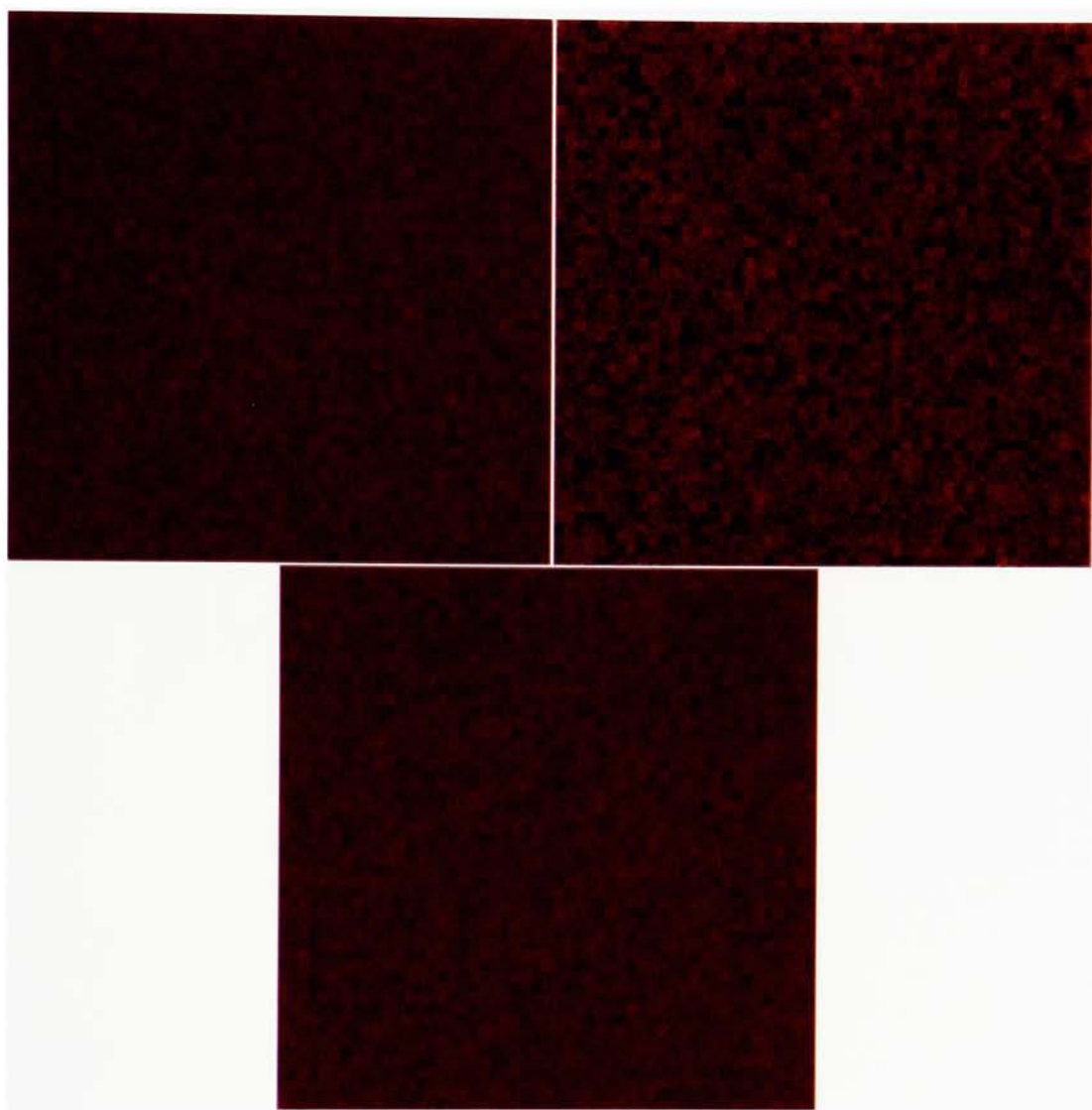


Image Set 4

Synthetic image generated with 40% of the image data corrupted by 3 standard deviations

Three classes of vegetation, all selected from the same Rochester LANDSAT TM image

Top left patch μ and Σ obtained from grass area of Rochester image

Top right patch μ and Σ obtained from deciduous trees of Rochester image

Bottom middle patch μ and Σ obtained from agricultural area of Rochester image

PLATE 18

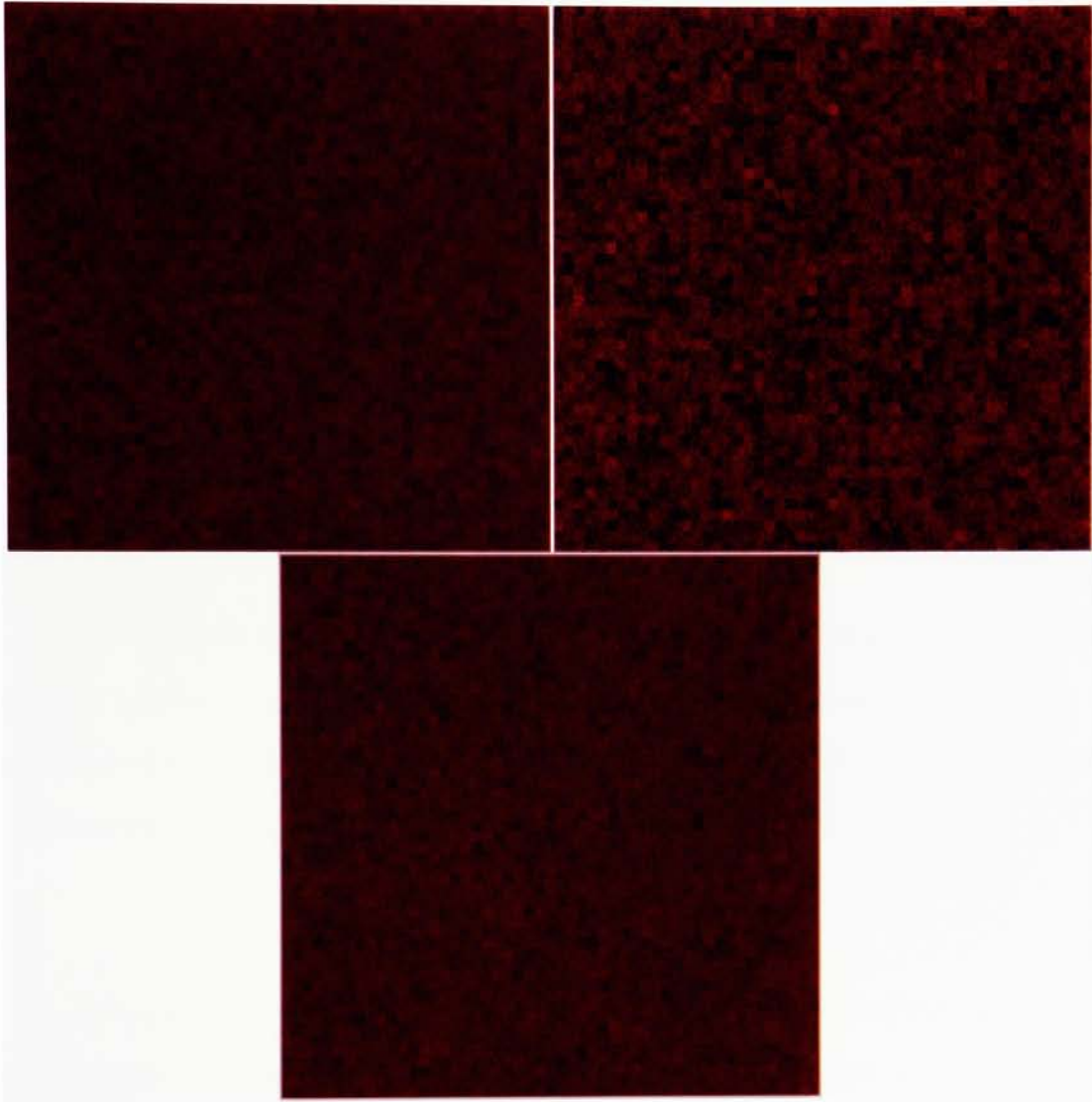


Image Set 4

Synthetic image generated with 60% of the image data corrupted by 5 standard deviations

Three classes of vegetation, all selected from the same Rochester LANDSAT TM image

Top left patch μ and Σ obtained from grass area of Rochester image

Top right patch μ and Σ obtained from deciduous trees of Rochester image

Bottom middle patch μ and Σ obtained from agricultural area of Rochester image

PLATE 19

5.1

Overview of Classification Procedures

Once this study's image sets had been create, they underwent classification using ERDAS software. This process began by selecting training samples using a polygon algorithm (DIGSCRN) to select portions of feature data. Since each image only had three class samples, the same set of polygons were used from image to image to ensure the sample size remained the same. After sample selection, the probability functions of each class were determined (SIGEXT) and stored to a file for reference during image classification.

5.2

Determining Classification Accuracy

To determine the classification accuracy of each image, the image areas sampled by the training set polygons were classified to determine the percentage of pixels classified correctly within this region (CMATRIX). The results of these classifications were listed in a contingency matrix and are included in this document in Appendix C.

5.3

Determining Signature Divergence

Class signatures generated from the synthetic ground features were examined to see how separable or diverse they were in vector space. This was accomplished by computing the Jeffries-Matusita Distance (JM). First α is determined

$$\alpha = \frac{1}{8}(\mu_i - \mu_j)^T \frac{(\Sigma_i + \Sigma_j)^{-1}}{2} (\mu_i - \mu_j) + \frac{1}{2} \ln \left(\frac{[(\Sigma_i + \Sigma_j / 2)]}{(\sqrt{|\Sigma_i| \times |\Sigma_j|})} \right) \quad (5.1)$$

and then substitutes into the following formula:

$$JM_{ij} \sqrt{2(1 - e^\alpha)} \quad (5.2)$$

where:

i and j represent the two class signatures being compared,

Σ_i is the covariance matrix of signature i ,

μ_i is the mean of signature i ,

$|\Sigma_i|$ is the determinant of Σ_i

The values for JM have a lower and upper bound of 0 - 1414. A value of zero indicates that the class signatures are inseparable. The upper bound value indicates the signatures are totally separable. The results for the class signatures generated for this study can be found in Section 6.3²⁰.

5.4 Determining Significant Changes in Classification Accuracy

This study examined the classification results for statistical difference using an $r \times c$ table and a χ^2 test for equality. These tests were designed to test for multiple equalities among the classification results listed in Appendix C. The $r \times c$ table used in this test had the following form:

	Class 1	Class 2	Class 3
Pixels correctly classified			
Pixels incorrectly classified			

Table 5.1 $r \times c$ Table

After the above table was filled, the expected frequencies, e_{rc} , were computed for each row and column as follows:

$$e_{rc} = \frac{(\text{total for row } r) (\text{total for column } c)}{(\text{total for table})} \quad (5.3)$$

Using the expected frequencies the following test for hypothesis was tested:

H_0 = all classification accuracies are equal

H_1 = all classification accuracies are not equal

A χ^2 test was used to test the statistic. χ^2 was computed as follows:

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(o_{rc} - e_{rc})^2}{e_{rc}}$$

where: o_{ij} = the observed frequency in the respective row and column

Results of the χ^2 test were compared to a χ^2 (1- α) statistics table with (rows-1)(columns-1) degrees of freedom. If the χ^2 value exceeded the table statistic, the null hypothesis was rejected at a (1- α) level of confidence²¹.

6.0 RESULTS AND DISCUSSIONS

6.1 Results of Univariate Analysis of LANDSAT TM Images

The results checking for univariate normality were obtained using Pearson's System of Frequency Curves as described in Section 3.1. According to the Elderton's interpretation of this system, an absolute κ value of 0.92 or less indicates the sample originated from a normal distribution. Tables 6.1 through 6.3 demonstrate that of the 144 image bands tested using Pearson's System of Frequency Curves, nine were determined to be non-normal. The histograms of these samples were plotted and examined to see how they compared to the other distributions tested. Plots showed that these samples possessed a much more narrow digital count range than the other class samples. Digital counts within these short ranges peaked quickly resulting in an increase to the fourth moment. It was concluded that the non-normality of these image bands were mostly a function of the length of the dynamic range.

	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
Charleston NC	9.013E-02	1.723	0.448	9.147E-03	-5.394E-04	-1.500E-02
San Francisco CA	-1.511E-03	-3.399E-03	-1.242E-03	-3.300E-03	-2.774E-04	-8.724E-04
Toronto Canada	-2.925E-03	-8.835E-04	2.344E-02	-5.166E-05	-2.054E-05	6.456E-05
Buffalo NY	-6.289E-04	3.894E-03	-7.896E-04	-1.524E-04	-2.281E-06	-7.511E-03
Rochester NY	2.549E-03	1.567E-03	4.492E-04	-1.627E-03	-8.314E-04	-1.632E-03
Hartford CT	-5.276E-03	-6.123E-02	0.411	-7.171E-04	-2.705E-04	-1.180E-02
Washington DC	-3.355E-03	-3.369E-03	-1.090E-02	-6.398E-04	-6.339E-04	-1.366E-03
Baltimore MD	-2.585E-03	-1.001E-02	-3.531E-03	3.825E-08	3.644E-05	-1.320E-03

Table 6.1 κ Values for Vegetation
Results of Univariate Check for Normality

	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
Charleston NC	-4.773E-04	-1.959E-03	-1.853E-03	-2.902E-03	-4.749E-04	-6.031E-04
San Francisco CA	-3.743E-04	-8.330E-04	-2.846E-04	-9.754E-04	1.006E-03	-6.868E-04
Toronto Canada	-3.976E-05	-5.228E-05	-8.974E-06	-9.115E-06	-2.874E-05	-4.012E-05
Buffalo NY	-5.877E-05	-1.711E-04	8.444E-05	-4.707E-05	-2.350E-05	-1.135E-04
Rochester NY	-9.876E-04	-2.791E-03	-1.258E-03	-9.084E-04	-1.147E-04	-3.280E-04
Hartford CT	-3.200E-04	-7.151E-04	-6.620E-04	-6.401E-05	-6.323E-05	-4.099E-04
Washington DC	-6.375E-04	-1.757E-03	-9.686E-04	-1.036E-03	-4.221E-04	-1.214E-03
Baltimore MD	-3.735E-03	-1.652E-03	-8.236E-04	-6.648E-04	-2.756E-04	-4.372E-04

Table 6.2 κ Values for Urban
Results of Univariate Check for Normality

	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
Charleston NC	-2.243E-03	0.500	-4.703E-03	1.085	0.511	0.299
San Francisco CA	-1.120E-02	-7.370E-02	-7.584E-03	-0.520	1.285	-2.770E-03
Toronto Canada	-8.785E-06	-7.828E-03	-1.749E-02	4.176E-03	-4.827	4.893E-03
Buffalo NY	7.361E-03	2.929E-02	-1.716E-02	-9.058E-03	-6.148E-03	-4.945E-02
Rochester NY	1.290E-02	1.867E-04	-5.642E-05	-7.854E-04	-8.233E-04	-5.321E-03
Hartford CT	2.54E-04	6.433E-06	3.834E-04	-2.273E-03	-5.108E-03	-7.738
Washington DC	-3.994E-04	-4.863E-04	-6.264E-03	6.079	0.227	-3.494
Baltimore MD	-4.338E-03	-0.996	0.937	0.627	0.502	0.243

Table 6.3 κ Values for Water
Results of Univariate Check for Normality

6.2 Results of Synthetic Image Classification with the Maximum Likelihood Classifier

The classification accuracy results of the first set of images (Plates 1 to 4) demonstrated that the classifier was able to distinguish easily between the urban, vegetation and water classes. Four different images ranging from normal to non-normal distribution were tested for classification accuracy using the maximum likelihood classifier. The percentage of pixels classified correctly is shown in Figure 13 and were obtained from the contingency matrixes listed in Appendix C.

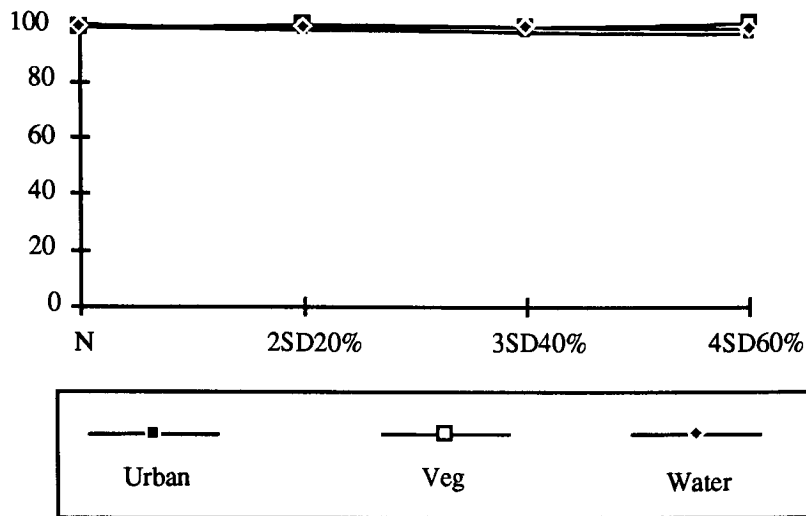


Figure 13 Classification results for image set one. Image contained urban, vegetation and water features.

The classification accuracy results for image set two (Plates 5-8) were only slightly worse than those seen in image set one as shown in Figure 14. The imagery used in this set represent three classes of water which possessed the same range of normal to non-normal distributions seen in set one.

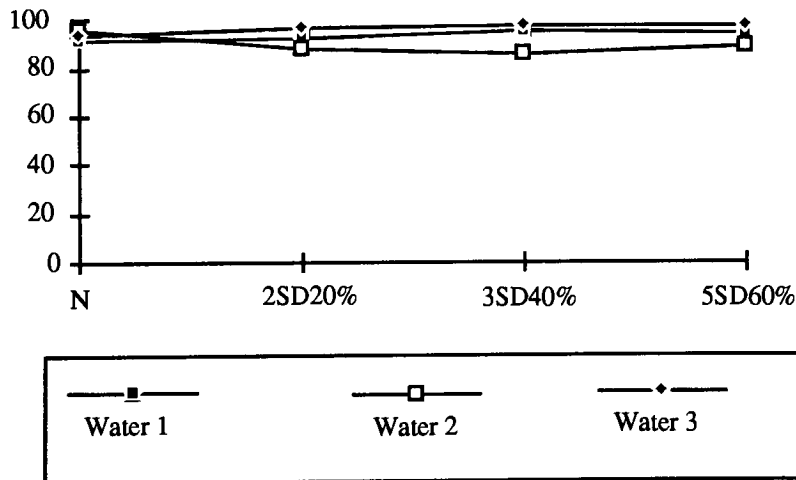


Figure 14 Classification results for image set two. Image contained three different water samples.

Image set three (Plates 8-15) demonstrated the greatest loss to classification accuracy. These results are shown in Figure 15. This image set consisted of water means, but used a vegetation covariance matrix.

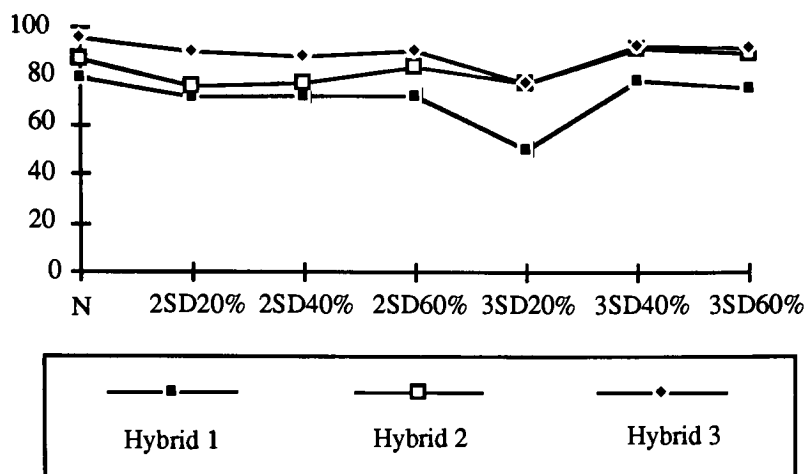


Figure 15 Classification results for image set three. Image contained three hybrid sample consisting of water μ and vegetation Σ

Set four (plates 16-19) demonstrated some loss in classification accuracy. This image set consisted of three classes of vegetation (deciduous trees, grass, agriculture) taken from the same LANDSAT scene. The classification results of this image set are illustrated in Figure 16.

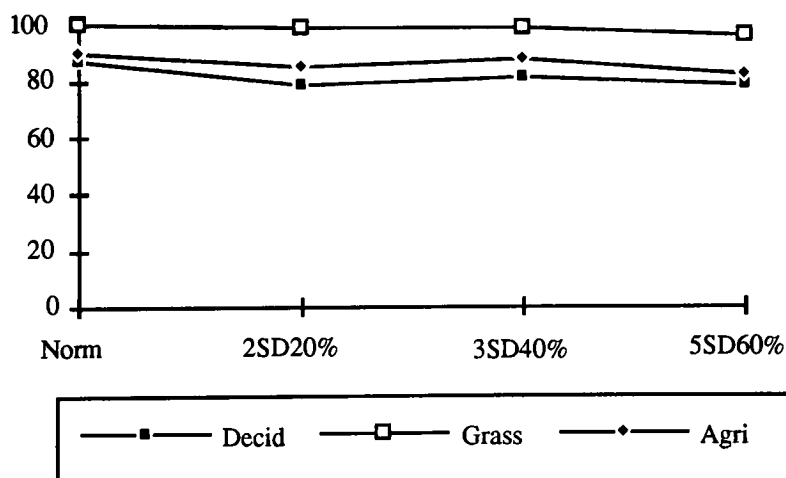


Figure 16 Classification results for image set four. Image contained three vegetation scenes. μ and Σ were selected from the same LANDSAT image

All of the classification results shown above were tested for statistical difference using a $r \times c$ table and a χ^2 test for equality. The results of these tests showed that, with

the exception of image set one, all of the other image sets had classification accuracies statistically different from one another.

6.3 Conclusions and Discussion of Study Results

The intent of the study was to check for normality of LANDSAT TM bands and to challenge the robustness of a parametric classifier. The first objective was accomplished by using Pearson's System of Frequency Curves to check for normality. Of the 144 image bands tested, nine were proven to be non-normal. This indicates that for common classes such as water, vegetation, and urban it is likely that most LANDSAT class bands are normally distributed.

The second objective of the study was accomplished by using a parametric classifier (maximum likelihood classifier) to classify image data from both normal and non-normal distributions. The classification accuracy results (section 6.2) indicate that the maximum likelihood classifier worked equally well on normally and non-normally distributed data. Classification accuracy results were higher overall for class samples with significantly different means and little distributional overlap (image sets one and two) and lower for class samples with similar means and large amounts of distributional overlap (image sets three and four). This is logical since the more separate class populations are in vector space, the less likely misclassification of image pixels will occur. The fact that the classifier experienced no additional confusion assigning pixels to non-normal populations is further confirmed by examining the measure of divergence values (Jeffries-Matusita Distance (JM)) shown in Figure 17.

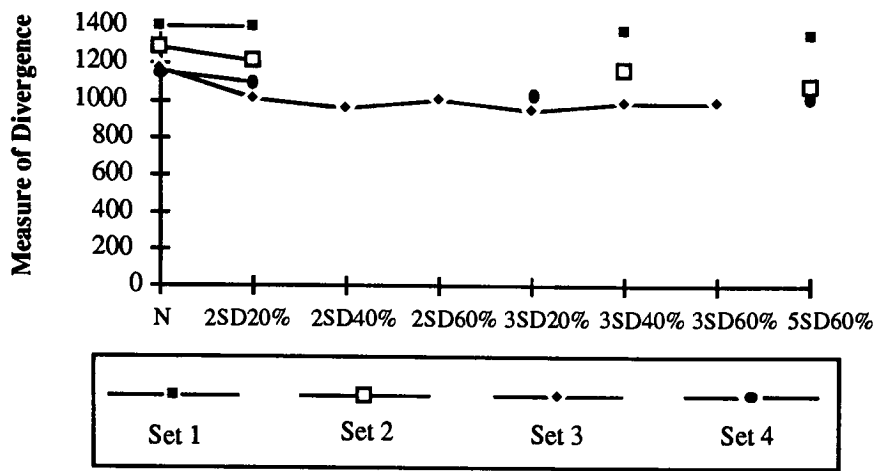


Figure 17 Measure of Divergence Values

Based on these results it would appear that the decision making property of the maximum likelihood classifier depends more on the mass of the population and not on the shape of it. To better understand this concept, consider the normal and non-normal populations depicted in Figure 18a and 18b. As the cumulative histograms of each population pair illustrates (Figures 18b and 18d), the probability of a candidate pixel to belong to a either class is not affected by the shape of the class populations, but instead is driven by where the bulk of the distribution lies in spectral space. Since the type of non-normality depicted in Figure 16 is the same non-normality induced in the imagery generated for this text, it is likely that the effect seen in Figure 18 accounts for the robustness of the maximum likelihood classifier performance seen in this study.

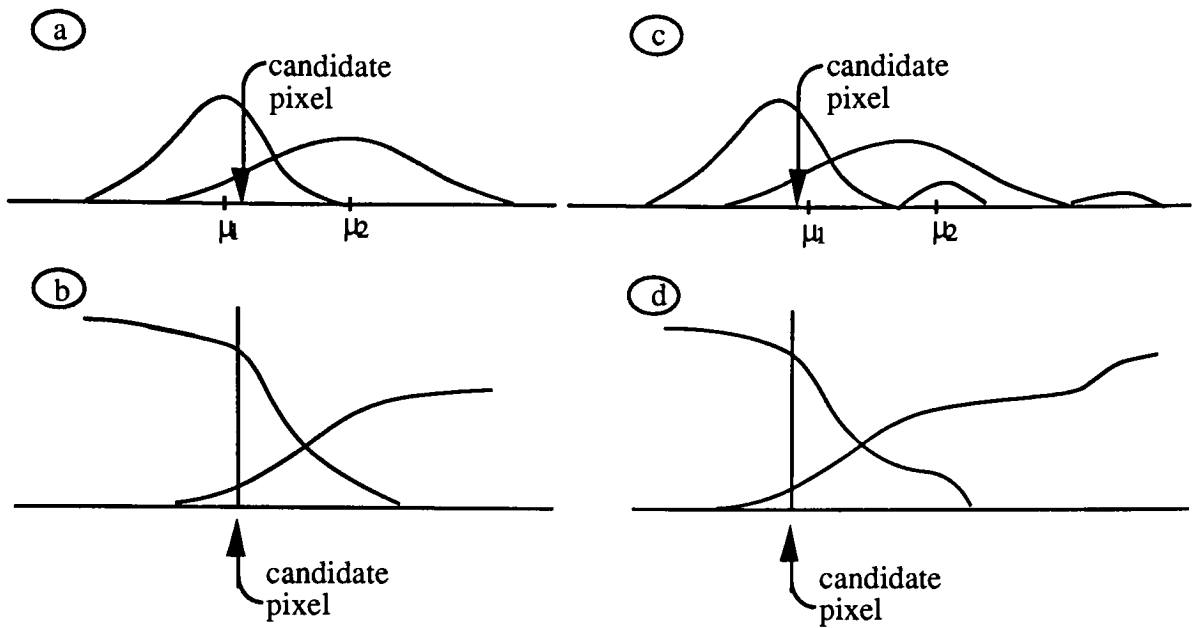


Figure 18 Comparison normal and non-normal cumulative histograms

While the maximum likelihood classifier may be robust enough to handle the type of non-normality used in this study, it may be greater challenged by other types of non-normally distributed data, for example a trimodal distribution of the population where the risk is greater for classification error. In this case, the shape of the trimodal distribution combined with its relation to other distributions in vector space could affect classification accuracies.

The primary weakness of this study was that only one type of non-normally distributed data was generated for this text. To completely determine the extent of a parametric classifier's abilities, other types of non-normal image data should be generated and classified. While some may consider this an academic point, the increased addition of texture data to multivariate image classification may require a better understanding of what classification robustness means. Texture data, which is derived from spectral imagery, can exhibit distributions very different from those commonly associated with LANDSAT data. Additionally, the current trend towards commercially available imagery with higher resolution may significantly affect class distributions²². Since these pixel footprints will average smaller portions of the earth's surface, the current assumptions of normality

associated with averaging process of the detector may be false. Further distributional research using higher resolution imagery should be conducted to see if any impact on classification accuracy is apparent.

7.0 Recommendations for Future Work

The first part of this study examined the assumptions that LANDSAT TM image data are normally distributed. While this was found to be true in the image examples examined for this study, it should be noted that all of the samples tested came from temperate climates. Additional imagery containing desert, marshlands, and snow scenes, water depths and urban areas as well as texture data derived from these classes should be examined.

The results of the second portion of the study indicated that non-normal distributions did not adversely impact image classification. However this study examined only one form of non-normality. Other types of non-normal distributions, especially those formed by texture data derived image scenes and imaging systems with better ground resolution would be helpful in fully understanding distributional effects on parametric classifiers.

REFERENCES

- ¹ T. Cacoullos ed., Discriminant Analysis and Application, Academic Press, New York and London, 1973, p. xiv.
- ² R.A. Showengerdt, Techniques for Image Processing and Classification in Remote Sensing, Academic Press Inc., New York, 1983, pp. 13-16.
- ³ R.L. Kettig, D.A. Landgrebe, "Classification of Multispectral Image Data by Extraction and Classification of Homogeneous Objects", IEEE Transactions on Geoscience Electronics, GE-14, 1, pp. 19-26.
- ⁴ Reference 1, p v
- ⁵ S. Das Gupta, "Theories and Methods in Classification: A Review", pp. 77-96 in Discriminant Analysis and Applications, T. Cacoullos ed., Academic Press, New York and London, 1973.
- ⁶ R.A. Johnson, and D.W. Wichern, Applied Multivariate Statistical Analysis. Prentice-Hall, Englewood Cliffs, NJ, 1982, pp. 470 - 478.
- ⁷ Reference 1, p xv
- ⁸ Reference 2, p 136
- ⁹ T.M. Lillesand and R.W. Kiefer, Remote Sensing and Image Interpretation, John Wiley & Sons, Inc., New York, 1979 p 445.
- ¹⁰ Reference 9, pp. 464-465.
- ¹¹ C. Salvaggio, J.R. Schott, Automated Pseudovariant Feature Normalization and Land Cover Classification, RIT/DIRS Report 87/88-63-123, 1987, pp. 21 24.
- ¹² Reference 2, pp. 145-147
- ¹³ P.H. Swain, and S.M. Davis, eds., Remote Sensing: The Quantitative Approach, McGraw-Hill, New York, 1978, pp. 178 - 184.
- ¹⁴ Reference 3, p 120
- ¹⁵ W.P. Elderton and N.L. Johnson, System of Frequency Curves, Cambridge University Press, 1969, pp. 35 - 46.
- ¹⁶ Personal conversation with Sterling Mason
- ¹⁷ I. Miller, J.E. Freund, Probability and Statistics for Engineers, Prentice-Hall, Inc., 1985, p 324.

¹⁸ ERDAS Field Guide, Second Edition, Version 7.5, July 1991, p 256.

¹⁹T.W. Anderson, An Introduction to Multivariate Statistics, John Wiley & Sons, Inc., 1958, pp. 24-27.

²⁰ Reference 18, pp. 129-130

²¹ Reference 17, pp. 162-163

²² J.R. Asker, "Pressure Builds to Free Satellite Imaging Sales", Aviation Week & Space Technology, November 15, 1993, pp. 26-27.

APPENDIX B

Class means and covariance matrixes used in the generation of Set 1 synthetic LANDSAT TM images

URBAN COVARIANCE MATRIX

	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
Band 1	191.00	108.35	133.19	104.20	145.22	89.66
Band 2	108.35	333.55	328.63	340.30	301.06	339.30
Band 3	133.19	328.63	336.50	342.08	316.76	336.60
Band 4	104.20	340.30	342.08	377.33	340.08	357.67
Band 5	145.22	301.06	316.76	340.08	369.08	334.53
Band 7	89.66	339.30	336.60	357.67	334.53	371.40
Band Means	118.77	66.04	71.76	64.33	82.23	61.97

VEGETATION COVARIANCE MATRIX

	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
Band 1	33.31	0.78	4.38	-16.64	3.04	5.24
Band 2	0.78	3.95	5.36	-1.41	6.30	5.19
Band 3	4.38	5.36	11.98	-17.33	9.86	11.05
Band 4	-16.64	-1.41	-17.33	113.58	13.74	-14.76
Band 5	3.04	6.30	9.86	13.74	48.35	19.79
Band 7	5.24	5.19	11.05	-14.76	19.79	37.13
Band Means	64.62	28.01	27.61	58.52	63.43	24.05

APPENDIX B

WATER COVARIANCE MATRIX

	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
Band 1	137.65	83.55	78.69	56.53	51.17	25.06
Band 2	83.55	64.04	61.05	47.86	45.11	30.34
Band 3	78.69	61.05	61.21	50.09	47.73	32.56
Band 4	56.53	47.86	50.09	46.05	44.15	31.75
Band 5	51.17	45.11	47.73	44.15	48.32	32.01
Band 7	25.06	30.34	32.56	31.75	32.01	27.55
Band Means	102.56	40.43	29.76	19.08	13.14	9.59

Class means and covariance matrixes used in the generation of Set 2 synthetic LANDSAT TM images

WATER 1 COVARIANCE MATRIX

	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
Band 1	2.86	0.60	0.38	0.26	-0.04	-0.01
Band 2	0.60	0.77	0.49	0.13	-0.003	0.02
Band 3	0.38	0.49	31.88	0.43	0.18	0.46
Band 4	0.26	0.13	0.43	0.62	0.13	0.06
Band 5	-0.04	-0.003	0.18	0.13	0.74	0.05
Band 7	-0.01	0.02	0.04	0.06	0.05	1.35
Band Means	60.67	22.79	18.29	9.18	5.32	2.67

APPENDIX B

WATER 2 COVARIANCE MATRIX

	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
Band 1	4.96	3.25	3.88	3.65	2.43	2.28
Band 2	3.25	4.27	4.60	5.08	3.50	2.94
Band 3	3.88	4.60	6.12	5.56	3.67	3.13
Band 4	3.65	5.08	5.56	43.49	28.72	9.46
Band 5	2.43	3.50	3.67	28.72	22.85	8.02
Band 7	2.28	2.94	3.13	9.46	8.02	5.32
Band Means	56.26	20.30	15.81	10.87	7.10	4.01

WATER 3 COVARIANCE MATRIX

	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
Band 1	2.64	0.68	0.82	0.30	0.23	0.18
Band 2	0.68	1.06	0.74	0.18	0.85	0.04
Band 3	0.82	0.74	1.29	0.26	0.19	0.11
Band 4	0.30	0.18	0.26	1.42	1.08	0.49
Band 5	0.23	0.08	0.19	1.08	2.61	0.87
Band 7	0.18	0.04	0.11	0.49	0.87	1.55
Band Means	60.51	23.40	19.70	10.85	6.52	3.54

Class means and covariance matrixes used in the generation of Set 3 synthetic LANDSAT TM images

HYBRID 1 COVARIANCE MATRIX

	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
Band 1	33.31	0.78	4.38	-16.64	3.04	5.24
Band 2	0.78	3.95	5.36	-1.41	6.30	5.19
Band 3	4.38	5.36	11.98	-17.33	9.86	11.05
Band 4	-16.64	-1.41	-17.33	113.58	13.74	-14.76
Band 5	3.04	6.30	9.86	13.74	48.35	19.79
Band 7	5.24	5.19	11.05	-14.76	19.79	37.13
Band Means	190.3	148.78	145.65	138.64	134.48	131.63

HYBRID 2 COVARIANCE MATRIX

	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
Band 1	45.05	26.53	47.11	-10.05	76.32	51.18
Band 2	26.53	19.37	32.13	0.31	57.90	35.77
Band 3	47.11	32.13	61.49	-4.88	112.37	68.46
Band 4	-10.05	0.31	-4.88	95.26	27.21	-2.82
Band 5	76.32	57.90	112.37	27.21	269.07	142.78
Band 7	51.13	35.77	68.46	-2.82	142.78	86.00
Band Means	230.24	166	160.11	141.17	135.41	132.42

APPENDIX B

HYBRID 3 COVARIANCE MATRIX

	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
Band 1	5.68	2.03	3.80	-3.69	5.74	4.55
Band 2	2.03	1.72	2.13	0.43	4.43	2.64
Band 3	3.80	2.13	5.64	-8.75	6.87	5.89
Band 4	-3.69	0.43	-8.75	88.83	24.05	-4.40
Band 5	5.74	4.43	6.87	24.05	50.98	19.33
Band 7	4.55	2.64	5.89	-4.40	19.33	12.79
Band Means	188.51	151.4	147.7	138.85	134.52	131.54

Class means and covariance matrixes used in the generation of Set 4 synthetic LANDSAT TM images

DECIDUOUS TREES COVARIANCE MATRIX

	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
Band 1	9.15	6.07	7.34	-4.39	12.07	6.70
Band 2	6.07	7.76	6.84	6.64	15.12	6.08
Band 3	7.34	6.84	8.28	-2.79	13.79	6.89
Band 4	-4.39	6.64	-2.79	133.53	31.89	-1.29
Band 5	12.07	15.12	13.79	31.89	56.23	18.52
Band 7	6.70	6.08	6.89	-1.29	18.52	10.31
Band Means	83.28	36.07	27.63	132.77	82.88	23.90

APPENDIX B

GRASS COVARIANCE MATRIX

	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
Band 1	1.88	0.89	0.85	1.86	0.89	1.20
Band 2	0.89	1.83	1.08	3.01	2.44	1.51
Band 3	0.85	1.08	1.27	2.29	0.01	0.95
Band 4	1.86	3.01	2.29	16.88	4.72	1.99
Band 5	0.89	2.44	-0.01	4.72	25.90	5.77
Band 7	1.20	1.51	0.95	1.99	5.77	4.06
Band Means	88.65	40.29	31.94	159.41	118.29	36.59

AGRICULTURE COVARIANCE MATRIX

	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
Band 1	3.23	1.59	1.55	4.92	5.60	2.39
Band 2	1.59	2.16	1.61	5.76	5.78	2.16
Band 3	1.55	1.61	2.14	2.85	5.43	2.17
Band 4	4.92	5.76	2.85	131.32	24.43	6.05
Band 5	5.60	5.78	5.43	24.43	29.58	10.17
Band 7	2.39	2.16	2.17	6.05	10.17	5.16
Band Means	86.43	37.85	30.27	143.58	85.35	25.77

APPENDIX C

Contingency Matrixes

Classification Accuracy of Set 1 Normally Distributed Image

	URBAN		VEG		WATER	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
URBAN	61478	100%	142	0.2%	247	0.4%
VEG	0	0.0%	59656	99.8%	52	0.1%
WATER	0	0.0%	0	0.0%	64678	99.5%

Classification Accuracy of Set 1, 20% of the Image Data Corrupted by 2 SD

	URBAN		VEG		WATER	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
URBAN	60802	98.9%	2	0.0%	45	0.7%
VEG	123	0.2%	59798	100%	2	0.0%
WATER	553	0.9%	0	0.0%	64030	99.2%

Classification Accuracy of Set 1, 40% of the Image Data Corrupted by 3 SD

	URBAN		VEG		WATER	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
URBAN	59880	97.4%	0	0.0%	130	0.2%
VEG	307	0.5%	59738	99.9%	130	0.2%
WATER	1291	2.1%	60	0.1%	64717	99.7%

APPENDIX C

Classification Accuracy of Set 1, 60% of the Image Data Corrupted by 5SD

	URBAN		VEG		WATER	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
URBAN	59327	96.5%	0	0.0%	260	0.4%
VEG	307	0.5%	59798	100%	585	0.9%
WATER	1844	3.0%	0	0.0%	64132	98.7%

Classification Accuracy of Set 2 Normally Distributed Image

	WATER 1		WATER 2		WATER 3	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
WATER 1	56314	91.6%	478	0.8%	3509	5.4%
WATER 2	369	0.6%	57347	95.9%	780	1.2%
WATER 3	4795	7.8%	1973	3.3%	60688	93.4%

Classification Accuracy of Set 2, 20% of the Image Data Corrupted by 2 SD

	WATER 1		WATER 2		WATER 3	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
WATER 1	56806	92.4%	4126	6.9%	910	1.4%
WATER 2	4611	7.5%	52861	88.4%	1105	1.7%
WATER 3	61	0.1%	2811	4.7%	62962	96.9%

APPENDIX C

Classification Accuracy of Set 2, 40% of the Image Data Corrupted by 3 SD

	WATER 1		WATER 2		WATER 3	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
WATER 1	58773	95.6%	5920	9.9%	390	0.6%
WATER 2	2705	4.4%	51366	85.9%	845	1.3%
WATER 3	0	0.0%	2512	4.2%	63742	98.1%

Classification Accuracy of Set 2, 60% of the Image Data Corrupted by 5SD

	WATER 1		WATER 2		WATER 3	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
WATER 1	58097	94.5%	4126	6.9%	2	0.0%
WATER 2	3381	5.5%	53400	89.3%	1494	2.3%
WATER 3	0	0.0%	2272	3.8%	64481	97.7%

Classification Accuracy of Set 3 Normally Distributed Image

	HYBRID 1		HYBRID 2		HYBRID 3	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
HYBRID 1	48814	79.4%	4784	8.0%	1624	2.5%
HYBRID 2	6701	10.9%	52203	87.3%	910	1.4%
HYBRID 3	5963	9.7%	2811	4.7%	62443	96.1%

APPENDIX C

Classification Accuracy of Set 3, 20% of the Image Data Corrupted by 2 SD

	HYBRID 1		HYBRID 2		HYBRID 3	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
HYBRID 1	44511	72.3%	9807	16.4%	5588	8.6%
HYBRID 2	5717	9.3%	45686	76.4%	910	1.4%
HYBRID 3	11250	18.3%	4305	7.2%	58479	89.9%

Classification Accuracy of Set 3, 40% of the Image Data Corrupted by 2 SD

	HYBRID 1		HYBRID 2		HYBRID 3	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
HYBRID 1	44326	72.1%	5501	9.2%	4159	6.4%
HYBRID 2	7131	11.6%	45985	77.0%	3444	5.3%
HYBRID 3	10021	16.3%	8312	13.9%	57374	88.3%

Classification Accuracy of Set 3, 60% of the Image Data Corrupted by 2 SD

	HYBRID 1		HYBRID 2		HYBRID 3	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
HYBRID 1	44264	72.0%	7834	13.1%	5653	8.7%
HYBRID 2	4242	6.9%	50230	84.1%	455	0.7%
HYBRID 3	12972	21.1%	1734	2.9%	58870	90.6%

APPENDIX C

Classification Accuracy of Set 3, 20% of the Image Data Corrupted by 3 SD

	HYBRID 1		HYBRID 2		HYBRID 3	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
HYBRID 1	28833	50.0%	8491	14.2%	13905	21.4%
HYBRID 2	3996	6.5%	46344	77.5%	520	0.8%
HYBRID 3	28649	46.6%	4963	8.3%	50552	77.8%

Classification Accuracy of Set 3, 40% of the Image Data Corrupted by 3 SD

	HYBRID 1		HYBRID 2		HYBRID 3	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
HYBRID 1	48322	78.6%	4126	6.9%	4873	7.5%
HYBRID 2	2582	4.2%	44416	91.1%	195	0.3%
HYBRID 3	10574	17.2%	1256	2.1%	59909	92.2%

Classification Accuracy of Set 3, 60% of the Image Data Corrupted by 3 SD

	HYBRID 1		HYBRID 2		HYBRID 3	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
HYBRID 1	46477	75.6%	4724	7.9%	5068	7.8%
HYBRID 2	2644	4.3%	53579	89.5%	260	0.4%
HYBRID 3	12357	20.1%	1495	2.5%	59649	91.8%

APPENDIX C

Classification Accuracy of Set 4 Normally Distributed Image

	DECIDUOUS		GRASS		AGRICULTURE	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
DECIDUOUS	53916	87.7%	0.0	0.0%	6692	10.3%
GRASS	0.0	0.0%	59738	100%	0.0	0.0%
AGRICULTURE	7562	12.3%	0.0	0.0%	58284	89.7%

Classification Accuracy of Set 4, 20% of the Image Data Corrupted by 2SD

	DECIDUOUS		GRASS		AGRICULTURE	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
DECIDUOUS	48506	78.9%	0.0	0.0%	9032	13.9%
GRASS	122	0.2%	59678	99.9%	195	0.3%
AGRICULTURE	12849	20.9%	60	0.1%	55750	85.8%

Classification Accuracy of Set 4, 40% of the Image Data Corrupted by 3 SD

	DECIDUOUS		GRASS		AGRICULTURE	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
DECIDUOUS	50473	82.1%	0.0	0.0%	7147	11.0%
GRASS	369	0.6%	59439	99.5%	780	1.2%
AGRICULTURE	10636	17.3%	299	0.5%	57050	87.8%

APPENDIX C

Classification Accuracy of Set 4, 60% of the Image Data Corrupted by SSD

	DECIDUOUS		GRASS		AGRICULTURE	
	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy	Number of Pixels	Classification Accuracy
DECIDUOUS	48445	78.8%	0.0	0.0%	6563	10.1%
GRASS	2336	3.8%	57587	96.4%	4548	7.0%
AGRICULTURE	10697	17.4%	2151	3.6%	53866	82.9%

APPENDIX D

This appendix contains the computer code used to create and analyze the data used in this study. The first program listed in this appendix (crick) was used to determine the intraband correlation of the LANDSAT TM classes studied in this thesis. A more complete description of how intraband correlation was determined can be found in section 4.2 of this text.

Crick requires a class map and a single band of LANDSAT data as input. The program calls several programs from the DIRS image processing library, and one subroutine (Test_for_Significance) written for this study. The following lists the DIRS image processing library call programs:

Ipx_SetPicsOps

Ipx_SetPicsDimens

Ipi_Openfile

Ipi_DiskPic

Ipi_getPicData

The code for the subroutine, Test_for_Significance, is presented after the crick program. This subroutine was used to determine if the correlation values generated in crick were significantly different from one another.

APPENDIX D

*Crick

THIS PROGRAM PAIRS DIGITAL COUNTS OF DIFFERENT DELTA SPACINGS
IN A SINGLE BAND AND DETERMINES THEIR R (CORRELATION) VALUE.

IMPLICIT NONE
CHARACTER*80 BAND1, CLASSIFIED_IMAGE,FILENAME,FILENAME1,FILENAME3
CHARACTER*1 TAB
INTEGER*4 I,N,K,L, TOTAL_COUNT,GRAND_TOTAL_COUNT
INTEGER*2 PIXEL(512),COUNTER,INDEX, START_INDEX, END_INDEX

INTEGER*4 NUMBER_OF_PAIRS, TOTAL_NUMBER_OF_PAIRS_PER_DELTA

INTEGER*2 DERIVATIVE_ARRAY(512), START_ARRAY_INPUT,STRING_LENGTH,LENGTH

INTEGER*2 XY_ARRAY(0:1,0:2000000), PAIRS(50),DELTA(50)

INTEGER*2 DC1,DC2,DC3,DC4, CLASS,Del,M,J

INTEGER SEED(100)

INTEGER*4 Row, Col, Bandmask

INTEGER*4 PicOps, IpiBlk_M, FilePtr1,FilePtr2,FilePtr3, FilePtr4

INTEGER*4 X_Dim, Y_Dim

REAL*8 SXY,XSUMSQ,YSUMSQ,YSUM,XSUM,XY,y_bar,x_bar, top

REAL*8 FILE_XSQ,XSQ,YSQ,x_bottom,y_bottom, x_bottom_sqr,y_bottom_sqr

REAL*8 SXX,SYX

REAL*8 R(50),denom

REAL*4 MEAN

DATA X_Dim / 512/

DATA Y_Dim /512/

Include 'Ipi_IpiLib'

TAB = CHAR(9)

Bandmask = 1

```

-----
*
      INPUT FILENAMES AND CLASSIFIED IMAGE CLASS VALUES
-----

      CALL Ipx_SetPicOps ( PicOps )
      CALL Ipx_SetPicDimens ( PicOps, X_dim, Y_dim)

      TYPE *, 'WHAT IS THE NAME OF THE FIRST BAND?'
      Accept ll, band1
11      format (a80)

      CALL Ipi_Openfile ( IpiBlk_M, band1 )
      CALL Ipi_DiskPic ( IpiBlk_M, filePtr1, PicOps )

      TYPE *, 'WHAT IS THE NAME OF THE CLASSIFIED IMAGE?'
      Accept ll, CLASSIFIED_IMAGE

      CALL Ipi_Openfile ( IpiBlk_M, classified_image)
      CALL Ipi_DiskPic ( IpiBlk_M, filePtr3, PicOps )

      TYPE *, 'WHAT IS THE DIGITAL COUNT YOU ARE LOOKING FOR?'
      READ(*,*) CLASS

      TYPE *, 'WHAT IS THE NAME OF THE FILE THAT HOLDS THE R VALUES?'
      Accept ('(A80)'), FILENAME1

      TYPE *, 'WHAT IS THE CLASS MEAN?'
      READ(*,*) MEAN

      TYPE *, 'WHAT IS THE FILENAME THAT HOLDS THE TEST_FOR_SIGN. VALUES?'
      Accept ('(A80)'), FILENAME3

      YSUM = 0.0
      XSUM = 0.0
      number_of_pairs = 0
      start_array_input = 0
      top = 0.0
      x_bar = 0.0
      y_bar = 0.0
      x_bottom = 0.0
      x_bottom_sqr = 0.0
      y_bottom = 0.0
      y_bottom_sqr = 0.0
      START_INDEX = 0

```

APPENDIX D

DO LOOP TO CHANGE DELTA VALUES

DO L = 1,50

DEL = L

TYPE*, 'DELTA EQUALS', DEL

DO LOOP TO OBTAIN DC COUNT PAIRS, DELTA PIXELS APART

DO J = 0,511,5

Do I = 0, 511

Call Ipi_getPicData (%val(filePtr3), PicOps, Bandmask, DC3,
i, j)

PIXEL(I) = DC3

END DO

DO I = 0,511

DERIVATIVE_ARRAY(I) (PIXEL(I)**2) (PIXEL(I+1)**2)

END DO

start_index = 0

end_index 0

DO I = 0,511

IF (DERIVATIVE_ARRAY(I) .NE. 0) THEN

END_INDEX = I

LENGTH = END_INDEX - START_INDEX

DO K = START_INDEX, (END_INDEX)

IF ((LENGTH .GT. DEL).AND.
(PIXEL(K+DEL).EQ.CLASS).AND.
(PIXEL(K) .EQ. CLASS)) THEN

STRING_LENGTH = END_INDEX - START_INDEX

CALL Ipi_getPicData (%val(fileptr1),picOps, Bandmask, DC1,
K, j)

CALL Ipi_getPicData (%val(fileptr1),picOps, Bandmask, DC2,
K + del, j)

APPENDIX D

```

COUNTER = END_INDEX    START_INDEX

NUMBER_OF_PAIRS    NUMBER_OF_PAIRS + 1

xy_array(0,START_ARRAY_INPUT) = dc1
xy_array(1,START_ARRAY_INPUT) = dc2

START_ARRAY_INPUT = START_ARRAY_INPUT+1

ENDIF

END DO

START_INDEX = END_INDEX +1

ENDIF

END DO

TOTAL_NUMBER_OF_PAIRS_PER_DELTA = TOTAL_NUMBER_OF_PAIRS
    _PER_DELTA + NUMBER_OF_PAIRS

```

R CORRELATION VALUE DETERMINED

```

DO I= 1, TOTAL_NUMBER_OF_PAIRS_PER_DELTA

    XSUM = XSUM + float(XY_ARRAY(0,I))

    YSUM = YSUM + float(XY_ARRAY(1,I))

end do

x_bar = xsum/total_number_of_pairs_per_delta
y_bar = ysum/total_number_of_pairs_per_delta

Do I = 1, Total_number_of_pairs_per_delta

    top = top + ((float(xy_array(0,I)) - x_bar)*
        (float(xy_array(1,I)) - y_bar))

    x_bottom = x_bottom + ((float(xy_array(0,i)) - x_bar)**2)
    y_bottom = y_bottom + ((float(xy_array(1,i)) - y_bar)**2)

End do

x_bottom_sqr = x_bottom**0.5

```

APPENDIX D

```

y_bottom_sqr = y_bottom**0.5

denom = x_bottom_sqr * y_bottom_sqr

if (denom .eq. 0.0) then

    R(L)    0.0

else

    R(L)    top/denom

endif

PAIRS(l) = total_number_of_pairs_per_delta

DELTA(L) = DEL
TOTAL_NUMBER_OF_PAIRS_PER_DELTA = 0
start_array_input=1
NUMBER_OF_PAIRS = 0
GRAND_TOTAL_COUNT    0.0
YSUM = 0.0
XSUM    0.0
top    0.0
x_bar = 0.0
y_bar = 0.0
x_bottom    0.0
x_bottom_sqr = 0.0
y_bottom = 0.0
y_bottom_sqr = 0.0

END DO

10      OPEN(1,FILE=FILENAME1,STATUS='NEW')
        DO I=1,50

            WRITE(1,10)DELTA(I),TAB,R(I)
            FORMAT(1X,I4,A1,F8.5)
            WRITE(1,*) R(I),PAIRS(I)
        END DO

        CLOSE (1)

        CALL TEST_FOR_SIGNIFICANCE (PAIRS,R,MEAN,DELTA,FILENAME3)

End

```

APPENDIX D

```

SUBROUTINE TEST_FOR_SIGNIFICANCE(PAIRS,R,MEAN,DELTA,FILENAME3)

IMPLICIT NONE

CHARACTER*80 FILENAME3

CHARACTER*1 TAB

REAL*8 R(50),Z(50),MEAN

INTEGER*2 PAIRS(50),DELTA(50)

INTEGER*2 N,I

REAL CAP_Z

TAB = CHAR(9)

-----

DO I = 1, 50

TYPE *, 'R(I)', R(I)

CAP_Z = (1.0/2.0) * (DLOG((1.0+R(I))/(1.0-R(I))))

TYPE *, 'I', I
TYPE *, 'CAP_Z', CAP_Z

z(i) = ((pairs(i) - 3)**0.5) * cap_z

TYPE *, 'Z(I)', Z(I)

END DO


OPEN (3,FILE=FILENAME3,STATUS='NEW')

DO I = 1,50

WRITE (3,10) I,TAB,Z(I)

10  FORMAT(1X,I4,A1,F8.5)

END DO

END

```

This computer code (Beta) was created to check the normality of individual LANDSAT TM class bands as described in section 3.1. Beta accepts single LANDSAT bands, and uses the following call programs from the DIRS image processing library,

Ipx_SetPicsOps

Ipx_SetPicsDimens

Ipi_Openfile

Ipi_DiskPic

Ipi_getPicData

and one program was called from IMSL:

UVSTA

APPENDIX D

```

*BETA
IMPLICIT NONE

CHARACTER*80 BAND1, CLASSMASK, FILENME

INTEGER*4 I, J, K, IDO, NROW, NVAR, LDX, IWT, IFRQ, MOPT, IPRINT, NRMISS, LDSTAT

PARAMETER (NVAR = 1, LDSTAT = 15)

REAL X1(262144,1), CONPRM, CONPRV, STAT(LDSTAT, NVAR), SD, FOURTH, THIRD

REAL SECOND, KAPPA, TOP, BOTTOM

REAL BETA1, BETA2, SD3, SD4

INTEGER*4 X_DIM, Y_DIM

DATA X_DIM /512/

DATA Y_DIM /512/

INTEGER*2 DC1, DC3, FILE1 (262144,1)

INTEGER*4 Row, Col, Bandmask

INTEGER*4 PicOps, IpiBlk_M, FilePtr1, fileptr3

Include 'Ipi_IpiLib'

Bandmask = 1
-----

CALL Ipx_SetPicOps ( PicOps )
CALL Ipx_SetPicDimens ( PicOps, X_dim, Y_dim)

WRITE(6,*)
TYPE *, 'WHAT IS THE NAME OF THE LANDSAT BAND?'
Accept 11, band1
11      format (a80)

CALL Ipi_Openfile ( IpiBlk_M, band1 )
CALL Ipi_DiskPic ( IpiBlk_M, filePtr1, PicOps )

WRITE(6,*)
TYPE *, 'WHAT IS THE NAME OF THE CLASS MASK?'
Accept 11, CLASSMASK

CALL Ipi_Openfile ( IpiBlk_M, classmask )
CALL Ipi_DiskPic ( IpiBlk_M, filePtr3, PicOps )

CONPRM = 95.0
CONPRV = 95.0
LDX = 262144
IDO = 0
IWT = 0
IFRQ = 0
MOPT = 1
IPRINT = 2

Do I = 0, 511

    Do J = 0, 511

        Call Ipi_getPicData (%val(filePtr3), PicOps, Bandmask, DC3,

```


APPENDIX D

```

        i, j )

    If ( DC3 .eq. 127) Then

        Call Ipi_getPicData (%val(filePtr1), PicOps, Bandmask, DC1,
            i, j )

        K = K+1

        X1(K,1) = DC1

        END IF
    End Do

End do

NROW = K

WRITE (6,*)

CALL UVSTA (IDO,NROW,NVAR,X1,LDX,IFRQ,IWT,MOPT,CONPRM,CONPRV,IPRINT,
    STAT,LDSTAT,NRMISS)

sd = stat(3,1)
third = stat(4,1)
fourth = stat(5,1)
second = stat(2,1)

BETA1 = THIRD**2/(SECOND**3.0)

BETA2 = FOURTH/(SECOND**2.0)

TOP = (BETA1*((BETA2+3)**2))

BOTTOM = (4*((2*BETA2)-(3*BETA1)-6)*((4*BETA2)-(3*BETA1)))

KAPPA=TOP/BOTTOM

WRITE(6,*)
TYPE *, 'WHAT IS THE NAME OF THE BETA FILE?'
READ(5,'(a)') FILENME
OPEN (UNIT=1, FILE = FILENME, ACCESS='APPEND', STATUS='OLD')

WRITE(1,*) BETA1,BETA2

WRITE(6,*)

write(6,*) 'beta1',beta1

write(6,*) 'beta2',beta2

WRITE(6,*) 'KAPPA =', KAPPA

WRITE(6,*)

CLOSE (1)

End

```

The main program and corresponding subroutines were used to create the synthetic LANDSAT imagery for this study. The main program requires the multivariate mean and covariance matrix of the class feature being generated as inputs. The following table listed the names of programs and function.

Program name	Function
Main_program	Main program of synthetic image generator
Create_mega_array	subroutine that creates the 150 x 150 matrix which statistically describes a 5 pixel x 5 pixel x 6 band block of image data
Test_punt	this subroutine inputs the 150 x 150 matrix and outputs a 256 x 256 block synthetic image data (one band)
Create_corrected_eigenvectors1	a subroutine that scales the eigenvectors derived from the 150 x 150 matrix by the eigenvalues
make_bands_punt	a subroutine used to place the 256 x 256 blocks of synthetic image data into an ERDAS acceptable format.
Image_into_bands	a subroutine which places all 6 synthetic image into one multivariate image file
Corrupt_normal_data	the subroutine used to create the non-normal image data tested in this study

APPENDIX D

```

*Main_program
  INTEGER NOUT

  REAL A(6,6), B(6,1), STACKED_ARRAY(1:5,1:30,1:30),D(6),E(9),
    CORR(6,6), AA(6,6), V(6,6),W(6,6), C(6), PRINCE(6,6),
    CHAR(6,6), HART(6,6), BALT(6,6), WASH(6,6), vsf(6,6),
    vwash(6,6)

  REAL C1,C2,C3,C4,C5,C6,MEGA_ARRAY(150,150)

  EXTERNAL EPISF,EVCSF,UMACH,WRRRN,TRNRR,MRRRR

*      COV MATRIX OF URBAN SF-----

  DATA A/191.0, 108.35, 133.19, 104.20, 145.22, 89.66, 108.35, 333.55,
    328.63, 340.30, 301.06, 339.30, 133.19, 328.63, 336.50, 342.08,
    316.76, 336.60, 104.20, 340.30, 342.08, 377.33, 340.08, 357.67,
    145.22, 301.06, 316.76, 340.08, 369.08, 334.53, 89.66, 339.30,
    336.60, 357.67, 334.53, 371.40/

*      CLASS MEANS-----

*urban      DATA C/118.0, 66.0, 71.0, 64.0, 82.0, 61.0/

*      CORRELATION COEF-----

*urban DATA D/0.861, 0.683, 0.562, 0.481, 0.419, 0.376/

-----

  CREATE THE 150 X 150 "MEGA_ARRAY" TO GENERATE THE CLASS

-----

  CALL CREATE_MEGA_ARRAY(STACKED_ARRAY,A,D,MEGA_ARRAY)

  CALL test_punt(MEGA_ARRAY,C)

  END

```

APPENDIX D

```

SUBROUTINE CREATE_MEGA_ARRAY(STACKED_ARRAY,COV_ARRAY,ROE_ARRAY,MEGA_ARR
    AY)

IMPLICIT NONE

CHARACTER*80 FILENAME1,FILENAME2,FILENAME3,FILENAME4,FILENAMES5

INTEGER K,M,L,J,I,X,COUNT,COUNTER,BLOCK_NUMBER,INDEX,ROW,COLUMN

INTEGER BEGINNING_COLUMN,BEGINNING_ROW

REAL STACKED_ARRAY(1:5,1:30,1:30), COV_ARRAY(6,6),ROE_ARRAY(6),ROEX

REAL MEGA_ARRAY (150,150)

X = 1

DO K = 1,5
    COUNT = 0

    DO M = 1,5

        DO L = 1,6
            COUNTER = 0
            COUNT = COUNT + 1

            DO J = 1,5

                DO I = 1,6
                    COUNTER = COUNTER + 1

                    IF ((X.EQ.K) .AND. (J.EQ.M)) THEN

                        STACKED_ARRAY(K,COUNT,COUNTER) =
                            COV_ARRAY(L,I)

                    ELSE

                        CALL ROE(X,K,M,J,L,I,ROEX,ROE_ARRAY)

                        STACKED_ARRAY(K,COUNT,COUNTER) =
                            ROEX * COV_ARRAY(L,I)

                    ENDIF

                END DO

            END DO

        END DO

    END DO

    COUNT = 0

    COUNTER = 0

END DO

```

```

DO J = 1,5

    DO I = 1,5

        TYPE*, ' WHAT IS THE BLOCK NUMBER?'
        READ(*,*) BLOCK_NUMBER
        BEGINNING_ROW = 0 + COUNT
        BEGINNING_COLUMN = 0 + COUNTER
        DO K = 1,30

            COUNT = COUNT + 1

            DO L = 1,30

                COUNTER = COUNTER + 1

                MEGA_ARRAY(COUNT,COUNTER) = STACKED_ARRAY
                    (BLOCK_NUMBER,K,L)

            END DO

            COUNTER = BEGINNING_COLUMN

        END DO
        COUNTER = BEGINNING_COLUMN + 30
        COUNT = BEGINNING_ROW
    END DO

    COUNTER = 0
    COUNT = BEGINNING_ROW + 30

END DO
END

```

APPENDIX D

```
SUBROUTINE test_punt(MEGA_ARRAY,MEAN)
```

```
IMPLICIT NONE
```

```
CHARACTER *80 file
```

```
INTEGER I, J,L,M, NRM, NCM, NRC, NCC, NCO, NRS, NCS, NRN, NCN, COUNTER,  
COUNT, TALLY, NRPIX, NCPIX, start_row, start_column,point,INDEX,  
PLACE, PZ
```

```
PARAMETER (NRM = 150, NCM = 150, NRC = 6, NCC = 6, NCO = 1, NRS = 120,  
NCS = 120, NRN = 30, NCN = 30, NRPIX = 256, NCPIX = 256)
```

```
REAL BAND1(NRPIX, NCPIX), BAND2(NRPIX, NCPIX), BAND3(NRPIX, NCPIX),  
BAND4(NRPIX, NCPIX), BAND5(NRPIX, NCPIX), BAND7(NRPIX, NCPIX)
```

```
REAL ARRAYS USED IN SYNTHETIC IMAGE CALCULATIONS
```

```
REAL MEGA_ARRAY(NRM,NCM), RANDOM_ONE(NRM,NCO), MEAN(NRC,NCO),  
EIGENVECTOR(NRM,NCM), EIGENVALUE(NRM,NCO),  
EIGENVECTOR_CORRECTED(NRM,NCM),  
TRANS_EIGENVECTOR_CORRECTED(NRM,NCM), PZ_ONE(NRM,NCO),  
CORRECTED_PZ_ONE(NRM,NCO)
```

```
PART ONE
```

```
THIS SECTION WILL PRODUCE X1 = P'Z + MEAN.  
THIS (150X1) ARRAY WILL BE LABELED "CORRECTED_PZ_ONE" IN THIS PROGRAM &  
WILL PRODUCE A SERIES OF 5 PIXEL X 5 PIXELS X 6 BANDS BLOCKS OF  
IMAGE DATA
```

```
BEGIN BY CREATING EIGENVECTORS, EIGENVALUES. P = (150,150) MATRIX OF  
EIGENVECTORS
```

```
CALL EVCSF (NRM,mega_array, NRM, eigenvalue, eigenvector, NRM)
```

```
CHANGE EIGENVECTOR TO UNIT LENGTH
```

```
NOTE: EIGENVECTOR_CORRECTED WILL BE TRANSPOSED IN SUBROUTINE CREATE_CORRECTED_  
EIGENVECTOR
```

```
CALL CREATE_CORRECTED_EIGENVECTORS1(eigenvalue,eigenvector,  
eigenvector_corrected)
```

APPENDIX D

```
GET (150,1) MATRIX OF PSEUDO NORMALLY DISTRIBUTED NUMBERS
```

```
Start_row = 1
```

```
Start_column = 1
```

```
Do L = 1,51
```

```
    Do M = 1,51
```

```
    CALL RNNOR(NRM, random_one)
```

```
CREATE P'Z
```

```
CALL MRRRR(NRM,NCM,eigenvector_corrected, NRM,NRM,NCO,random_one,  
           NRM, NRM, NCO, pz_one, NRM)
```

```
ADD MEAN TO P'Z TO CREATE X1
```

```
point = 1  
COUNTER= 0
```

```
DO INDEX 1,25
```

```
    DO J = 1,6
```

```
    COUNTER = COUNTER + 1
```

```
    corrected_pz_one(COUNTER,NCO) = mean(J,NCO) + pz_one(COUNTER,NCO)
```

```
    END DO
```

```
point = point + 1  
END DO
```

FINALLY, STORE THE FIRST 5X5 PIXELS INTO THEIR INDIVIDUAL BANDS

```

COUNTER = 0

DO I = start_row, start_row + 4

    DO J = start_column, start_column + 4

        COUNTER = COUNTER + 1
        BAND1(I,J) = corrected_pz_one(COUNTER, NCO)

        counter = counter + 1
        BAND2(I,J) = corrected_pz_one(counter, NCO)

        counter = counter + 1
        BAND3(I,J) = corrected_pz_one(COUNTER, NCO)

        counter = counter + 1
        BAND4(I,J) = corrected_pz_one(COUNTER, NCO)

        counter = counter + 1
        BAND5(I,J) = corrected_pz_one(COUNTER, NCO)

        counter = counter + 1
        BAND7(I,J) = corrected_pz_one(COUNTER, NCO)

    END DO

END DO

start_column = start_column + 5

End do

start_row = start_row + 5
start_column = 1

End do

CALL make_bands_punt(BAND1, BAND2, BAND3, BAND4, BAND5, BAND7, mean)

type*, 'THE END!!!'

END

```


APPENDIX D

```
SUBROUTINE CREATE_CORRECTED_EIGENVECTORS1 (EVAL,EVEC,EVEC_CORRECTED)

IMPLICIT NONE

INTEGER I,J, COUNTER

REAL EVAL(150),EVEC(150,150),EVEC_CORRECTED(150,150)

REAL CONSTANT(150),TEE(150,150),TRANS_EVEC(150,150)

-----

OBTAIN THE TRANSPOSE OF THE ARRAY 'EVEC' AND PRINT TO SCREEN

-----

CALL TRNRR(150,150,EVEC,150,150,150,TRANS_EVEC,150)

-----

MULTIPLY THE EVEC' (THE TRANSPOSE OF EVEC) BY EVEC, AND PRINT TO SCREEN

-----

CALL MRRRR(150,150,TRANS_EVEC,150,150,150,EVEC,150,150,150,TEE,150)

      DO I = 1,150

          DO J = 1,150

              IF (I.EQ.J ) THEN

                  CONSTANT(I) = (EVAL(I)/TEE(I,J))**0.5

              END IF

          END DO

      END DO

DO I = 1, 150

    DO J = 1,150

        EVEC_CORRECTED(J,I) = EVEC(J,I)*CONSTANT(I)

    END DO

END DO

END
```

APPENDIX D

```
SUBROUTINE MAKE_BANDS_PUNT(INPUT_ONE,INPUT_TWO,INPUT_THREE,INPUT_FOUR,
    INPUT_FIVE,INPUT_SEVEN,mean)
```

```
INTEGER I,J,NOUT,K,NOBS,IOPT,NBAR,ISP
```

```
REAL INPUT_ONE(256,256), INPUT_TWO(256,256),INPUT_THREE(256,256),
    INPUT_FOUR(256,256),INPUT_FIVE(256,256),INPUT_SEVEN(256,256),
.      MOPT,STAT(15,256),STATS_ARRAY(65025,1), RANDOM_TEST(65025,1),
.      mean (6,1)
```

```
INTEGER*2 BAND_ONE(512,512), BAND_TWO(512,512), BAND_THREE(512,512),
.      BAND_FOUR(512,512),BAND_FIVE(512,512),BAND_SEVEN(512,512),
.      NRMISS
```

```
INTEGER*2 INPUT_BAND_ONE(256,256), INPUT_BAND_TWO(256,256),
    INPUT_BAND_THREE(256,256),INPUT_BAND_FOUR(256,256),
    INPUT_BAND_FIVE(256,256),INPUT_BAND_SEVEN(256,256)
```

```
THIS SECTION PLACES NORMAL IMAGE DATA INTO A FILE FORMAT THAT
CAN BE RECEIVED BY ERDAS. THIS SECTION IS COMMENTED OUT IF IMAGE
DATA IS INTENDED TO BE CORRUPTED
```

```
*      DO I = 1, 255

*          DO J = 1, 255

*              IF (INPUT_ONE(I,J) .LT. 0.0) THEN

*                  INPUT_ONE(I,J) = 0.0

*              ELSE IF (INPUT_ONE(I,J) .GT. 255.0) THEN

*                  INPUT_ONE(I,J) = 255.0

*              END IF

*          END DO

*      END DO

*      DO I = 1, 255

*          DO J = 1, 255

*              IF (INPUT_TWO(I,J) .LT. 0.0) THEN

*                  INPUT_TWO(I,J) = 0.0

*              ELSE IF (INPUT_TWO(I,J) .GT. 255.0) THEN

*                  INPUT_TWO(I,J) = 255.0

*              END IF

*          END DO

*      END DO
```

APPENDIX D

```
*      DO I   1, 255      ,
*
*          DO J = 1, 255
*
*              IF (INPUT_THREE(I,J) .LT. 0.0) THEN
*
*                  INPUT_THREE(I,J) = 0.0
*
*              ELSE IF (INPUT_THREE(I,J) .GT. 255.0) THEN
*
*                  INPUT_THREE(I,J) = 255.0
*
*              END IF
*
*          END DO
*
*      END DO

*      DO I = 1, 255
*
*          DO J = 1, 255
*
*              IF (INPUT_FOUR(I,J) .LT. 0.0) THEN
*
*                  INPUT_FOUR(I,J) = 0.0
*
*              ELSE IF (INPUT_FOUR(I,J) .GT. 255.0) THEN
*
*                  INPUT_FOUR(I,J) = 255.0
*
*              END IF
*
*          END DO
*
*      END DO

*      DO I = 1, 255
*
*          DO J = 1, 255
*
*              IF (INPUT_FIVE(I,J) .LT. 0.0) THEN
*
*                  INPUT_FIVE(I,J) = 0.0
*
*              ELSE IF (INPUT_FIVE(I,J) .GT. 255.0) THEN
*
*                  INPUT_FIVE(I,J) = 255.0
*
*              END IF
*
*          END DO
*
*      END DO
```

```

*      DO I = 1, 255
*
*          DO J = 1, 255
*
*              IF (INPUT_SEVEN(I,J) .LT. 0.0) THEN
*
*                  INPUT_SEVEN(I,J) = 0.0
*
*              ELSE IF (INPUT_SEVEN(I,J) .GT. 255.0) THEN
*
*                  INPUT_SEVEN(I,J) = 255.0
*
*              END IF
*
*          END DO
*
*      END DO
*
*      CALL IMAGE_into_bands(INPUT_ONE,INPUT_TWO,INPUT_THREE,
*      INPUT_FOUR,INPUT_FIVE,INPUT_SEVEN)
*
*      CALL CORRUPT_NORMAL_DATA(INPUT_ONE,INPUT_TWO,INPUT_THREE,INPUT_FOUR,
*      INPUT_FIVE,INPUT_SEVEN,mean)

```

APPENDIX D

```

SUBROUTINE CORRUPT_NORMAL_DATA(INPUT_ONE,INPUT_TWO,INPUT_THREE,INPUT_FOUR,
    INPUT_FIVE,INPUT_SEVEN, mean)

EXTERNAL RNUNF,rnun

INTEGER I, J, corrupt_number1, corrupt_number2, corrupt_number3,
    corrupt_number4, corrupt_number5, corrupt_number7,pixel_counter

REAL INPUT_ONE(256,256), INPUT_TWO(256,256),INPUT_THREE(256,256),
. INPUT_FOUR(256,256),INPUT_FIVE(256,256),INPUT_SEVEN(256,256),
. MOPT,STAT(15,256),STATS_ARRAY(65025,1), RANDOM_TEST(65025,1),
. mean(6,1)

REAL corrupt_total1, corrupt_total2, corrupt_total3, corrupt_total4,
    corrupt_total5, corrupt_total7, mean_shift1, mean_shift2,
    mean_shift3, mean_shift4, mean_shift5, mean_shift7

REAL BAND1_CORRUPT(256,256),BAND2_CORRUPT(256,256),
    BAND3_CORRUPT(256,256),BAND4_CORRUPT(256,256),
    BAND5_CORRUPT(256,256),BAND7_CORRUPT(256,256)

REAL PIXEL_SHIFT1, PIXEL_SHIFT2,PIXEL_SHIFT3, PIXEL_SHIFT4, PIXEL_SHIFT5,
    PIXEL_SHIFT7, R(1,1),C

* 2SD vsf

* PIXEL_SHIFT1 = 13.42
* PIXEL_SHIFT2 = 8.80
* PIXEL_SHIFT3 = 15.68
* PIXEL_SHIFT4 = 19.52
* PIXEL_SHIFT5 = 32.80
* PIXEL_SHIFT7 24.36

* 3SD vsf

* PIXEL_SHIFT1 = 20.13
* PIXEL_SHIFT2 = 13.2
* PIXEL_SHIFT3 = 23.52
* PIXEL_SHIFT4 = 29.28
* PIXEL_SHIFT5 = 49.2
* PIXEL_SHIFT7 = 36.54

```

CORRUPT BAND ONE

```

PIXEL_COUNTER = 0
CORRUPT_TOTAL1 = 0

DO I = 1,255
    DO J = 1,255
        CALL RNUN (1,R)
        IF (R(1,1) .LE. 0.2) THEN
            BAND1_CORRUPT(I,J) = INPUT_ONE(I,J) + PIXEL_SHIFT1
        ELSE
            BAND1_CORRUPT(I,J) = INPUT_ONE(I,J)
        END IF
        IF (BAND1_CORRUPT(I,J) .LT. 0.0) THEN
            BAND1_CORRUPT(I,J) = 0.0
        ELSE IF (BAND1_CORRUPT(I,J) .GT. 255.0) THEN
            BAND1_CORRUPT(I,J) = 255.0
        END IF
    END DO
END DO

DO I = 1, 255
    DO J = 1, 255
        CORRUPT_TOTAL1 = CORRUPT_TOTAL1 + BAND1_CORRUPT(I,J)
        PIXEL_COUNTER = PIXEL_COUNTER + 1
    END DO
END DO

MEAN_SHIFT1 = (CORRUPT_TOTAL1/PIXEL_COUNTER) - MEAN(1,1)

DO I = 1, 255
    DO J = 1, 255
        BAND1_CORRUPT(I,J) = BAND1_CORRUPT(I,J) - MEAN_SHIFT1
    END DO

```

APPENDIX D

END DO

type*, ' '

```
type*, 'Corrupted Band One Completed '
```

type*, ' '

CORRUPT BAND TWO

```
PIXEL_COUNTER = 0
CORRUPT TOTAL2 = 0
```

DO I 1,255

DO J = 1,255

CALL RNUN (1,R)

```
IF (R(1,1) .LE. 0.2) THEN
```

```
BAND2 CORRUPT(I,J) = INPUT_TWO(I,J)+ PIXEL_SHIFT2
```

ELSE

BAND2 CORRUPT(I,J) = INPUT_TWO(I,J)

END IF

```
IF (BAND2 CORRUPT(I,J) .LT. 0.0) THEN
```

BAND2 CORRUPT(I,J) = 0.0

```
ELSE IF (BAND2 CORRUPT(I,J) .GT. 255.0) THEN
```

BAND2 CORRUPT(I,J) = 255.0

END IF

END DO

END DO

DO I = 1, 255

DO J = 1, 255

CORRUPT_TOTAL2 = CORRUPT_TOTAL2 + BAND2_CORRUPT(I,J)

```
PIXEL_COUNTER = PIXEL_COUNTER + 1
```

END DO

END DO

```
MEAN_SHIFT2 = (CORRUPT_TOTAL2/PIXEL_COUNTER) - MEAN(2,1)
```

APPENDIX D

```
DO I = 1, 255

    DO J = 1, 255

        BAND2_CORRUPT(I,J) = BAND2_CORRUPT(I,J)    MEAN_SHIFT2

    END DO

END DO

type*.'      '

type*, 'Corrupted Band Two Completed '

type*, '      '

-----

CORRUPT BAND THREE

-----

PIXEL_COUNTER = 0
CORRUPT_TOTAL3 = 0

DO I = 1,255

    DO J = 1,255

        CALL RNUN (1,R)

        IF (R(1,1) .LE. 0.2) THEN

            BAND3_CORRUPT(I,J) = INPUT_THREE(I,J) + PIXEL_SHIFT3

        ELSE

            BAND3_CORRUPT(I,J)    INPUT_THREE(I,J)

        END IF

        IF (BAND3_CORRUPT(I,J) .LT. 0.0) THEN

            BAND3_CORRUPT(I,J) = 0.0

        ELSE IF (BAND3_CORRUPT(I,J) .GT. 255.0) THEN

            BAND3_CORRUPT(I,J) = 255.0

        END IF

    END DO

END DO

DO I = 1, 255

    DO J = 1, 255

        CORRUPT_TOTAL3 = CORRUPT_TOTAL3 + BAND3_CORRUPT(I,J)

        PIXEL_COUNTER = PIXEL_COUNTER + 1
```


APPENDIX D

```

        END DO

END DO

MEAN_SHIFT3 = (CORRUPT_TOTAL3/PIXEL_COUNTER)    MEAN(3,1)

DO I = 1, 255

    DO J = 1, 255

        BAND3_CORRUPT(I,J)    BAND3_CORRUPT(I,J)    MEAN_SHIFT3

    END DO

END DO

type*, '      '

type*, 'Corrupted Band Three Completed '

type*, '      '

-----

CORRUPT BAND FOUR

-----

PIXEL_COUNTER = 0
CORRUPT_TOTAL4 = 0

DO I = 1,255

    DO J = 1,255

        CALL RNUN (1,R)

        IF (R(1,1) .LE. 0.2) THEN

BAND4_CORRUPT(I,J) = INPUT_FOUR(I,J)+ PIXEL_SHIFT4

        ELSE

        BAND4_CORRUPT(I,J) = INPUT_FOUR(I,J)

        END IF

        IF (BAND4_CORRUPT(I,J) .LT. 0.0) THEN

        BAND4_CORRUPT(I,J) = 0.0

        ELSE IF (BAND4_CORRUPT(I,J) .GT. 255.0) THEN

        BAND4_CORRUPT(I,J) = 255.0

        END IF

    END DO

END DO

DO I = 1, 255

```

APPENDIX D

```
DO J = 1, 255

CORRUPT_TOTAL4 = CORRUPT_TOTAL4 + BAND4_CORRUPT(I,J)

PIXEL_COUNTER = PIXEL_COUNTER + 1

END DO

END DO

MEAN_SHIFT4 = (CORRUPT_TOTAL4/PIXEL_COUNTER)    MEAN(4,1)

DO I = 1, 255

DO J = 1, 255

BAND4_CORRUPT(I,J) = BAND4_CORRUPT(I,J)    MEAN_SHIFT4

END DO

END DO

type*, '      '

type*, 'Corrupted Band Four Completed '

type*, '      '

-----

CORRUPT BAND FIVE

-----

PIXEL_COUNTER = 0
CORRUPT_TOTAL5 = 0

DO I = 1,255

DO J = 1,255

CALL RNUN (1,R)

IF (R(1,1) .LE. 0.2) THEN

BAND5_CORRUPT(I,J) = INPUT_FIVE(I,J)+ PIXEL_SHIFT5

ELSE

BAND5_CORRUPT(I,J) = INPUT_FIVE(I,J)

END IF

IF (BAND5_CORRUPT(I,J) .LT. 0.0) THEN

BAND5_CORRUPT(I,J) = 0.0

ELSE IF (BAND5_CORRUPT(I,J) .GT. 255.0) THEN

BAND5_CORRUPT(I,J) = 255.0
```

APPENDIX D

```

        END IF
    END DO
END DO

DO I = 1, 255
    DO J = 1, 255
        CORRUPT_TOTAL5 = CORRUPT_TOTAL5 + BAND5_CORRUPT(I,J)
        PIXEL_COUNTER = PIXEL_COUNTER + 1
    END DO
END DO

MEAN_SHIFT5 = (CORRUPT_TOTAL5/PIXEL_COUNTER)    MEAN(5,1)

DO I = 1, 255
    DO J = 1, 255
        BAND5_CORRUPT(I,J) = BAND5_CORRUPT(I,J)    MEAN_SHIFT5
    END DO
END DO

type*, '      '

type*, 'Corrupted Band Five Completed '

type*, '      '

-----

CORRUPT BAND SEVEN

-----

PIXEL_COUNTER = 0
CORRUPT_TOTAL7 = 0

DO I = 1,255
    DO J = 1,255
        CALL RNUN (1,R)

        IF (R(1,1) .LE. 0.2) THEN

BAND7_CORRUPT(I,J) = INPUT_SEVEN(I,J) + PIXEL_SHIFT7

        ELSE

        BAND7_CORRUPT(I,J) = INPUT_SEVEN(I,J)

        END IF

        IF (BAND7_CORRUPT(I,J) .LT. 0.0) THEN

```

APPENDIX D

```
BAND7_CORRUPT(I,J) = 0.0

ELSE IF (BAND7_CORRUPT(I,J) .GT. 255.0) THEN

BAND7_CORRUPT(I,J) = 255.0

END IF

END DO

END DO

DO I = 1, 255

DO J = 1, 255

CORRUPT_TOTAL7 = CORRUPT_TOTAL7 + BAND7_CORRUPT(I,J)

PIXEL_COUNTER = PIXEL_COUNTER + 1

END DO

END DO

MEAN_SHIFT7 = (CORRUPT_TOTAL7/PIXEL_COUNTER)    MEAN(6,1)

DO I = 1, 255

DO J = 1, 255

BAND7_CORRUPT(I,J) = BAND7_CORRUPT(I,J) - MEAN_SHIFT7

END DO

END DO

type*, '      '

type*, 'Corrupted Band Seven Completed '

type*, '      '

CALL IMAGE_into_bands(band1_corrupt,band2_corrupt,band3_corrupt,
band4_corrupt,band5_corrupt,band7_corrupt)

END
```

APPENDIX D

```

SUBROUTINE image_into_bands(INPUT_ONE,INPUT_TWO,INPUT_THREE,INPUT_FOUR,
. INPUT_FIVE,INPUT_SEVEN)

INTEGER I, J

REAL INPUT_ONE(256,256), INPUT_TWO(256,256),INPUT_THREE(256,256),
. INPUT_FOUR(256,256),INPUT_FIVE(256,256),INPUT_SEVEN(256,256),
. MOPT,STAT(15,256),STATS_ARRAY(65025,1), RANDOM_TEST(65025,1)

REAL BAND_ONE_CLIPPED(256,256), BAND_two_CLIPPED(256,256),
. BAND_three_CLIPPED(256,256), BAND_four_CLIPPED(256,256),
. BAND_five_CLIPPED(256,256), BAND_seven_CLIPPED(256,256)

Integer*2 BAND_ONE(256,256), BAND_TWO(256,256), BAND_THREE(256,256),
. BAND_FOUR(256,256),BAND_FIVE(256,256),BAND_SEVEN(256,256),
. NRMISS

-----

CREATE BAND ONE

-----

DO I = 1,256

DO J = 1,256

if (input_one(i,j) .lt. 0.0) then

band_one_clipped(i,j) = 0.0

else

band_one_clipped(i,j) = input_one(i,j)

END IF

end do

end do

DO I = 1,255

DO J = 1,255

BAND_ONE(I,J) = NINT(band_ONE_clipped(I,J))

END DO

END DO

CALL IMAGE_OUT(BAND_ONE,256)

TYPE*, 'BAND ONE COMPLETED'
TYPE*, ' '

```

```
CREATE BAND TWO
```

```
DO I = 1,256
    DO J = 1,256

        if (input_two(i,j) .lt. 0.0) then
            band_two_clipped(i,j) = 0.0
        else
            band_two_clipped(i,j) = input_two(i,j)
        END IF
    end do
end do

DO I = 1,255
    DO J = 1,255

        BAND_two(I,J) = NINT(band_two_clipped(I,J))

    END DO

END DO

CALL IMAGE_OUT(BAND_TWO,256)

TYPE*, 'BAND two COMPLETED'
TYPE*, '      '
```

```
CREATE BAND THREE
```

```
DO I = 1,256
    DO J = 1,256

        if (input_three(i,j) .lt. 0.0) then
```

APPENDIX D

```
band_three_clipped(i,j) = 0.0
else
band_three_clipped(i,j) = input_three(i,j)

END IF

end do

end do

DO I = 1,255
DO J 1,255

BAND_three(I,J) = NINT(band_three_clipped(I,J))

END DO

END DO

CALL IMAGE_OUT(BAND_THREE,256)

TYPE*, 'BAND THREE COMPLETED'
TYPE*, ' ',
```

```
CREATE BAND FOUR
```

```
DO I = 1,256
DO J 1,256

if (input_four(i,j) .lt. 0.0) then
band_four_clipped(i,j) = 0.0
else
band_four_clipped(i,j) = input_four(i,j)

END IF

end do

end do
```

APPENDIX D

```
DO I = 1,255

    DO J = 1,255

        BAND_four(I,J) = NINT(band_four_clipped(I,J))

    END DO

END DO

CALL IMAGE_OUT(BAND_FOUR,256)

TYPE*, 'BAND FOUR COMPLETED'
TYPE*, '    '
```

```
CREATE BAND FIVE
```

```
DO I = 1,256

    DO J = 1,256

        if (input_five(i,j) .lt. 0.0) then
            band_five_clipped(i,j) = 0.0
        else
            band_five_clipped(i,j) = input_five(i,j)
        end if
    end do

end do

DO I = 1,255

    DO J = 1,255

        BAND_five(I,J) = NINT(band_five_clipped(I,J))

    END DO

END DO

CALL IMAGE_OUT(BAND_FIVE,256)

TYPE*, 'BAND FIVE COMPLETED'
TYPE*, '    '
```



```
-----  
CREATE BAND SEVEN  
-----
```

```
DO I = 1,256  
    DO J = 1,256  
  
        if (input_seven(i,j) .lt. 0.0) then  
            band_seven_clipped(i,j) = 0.0  
        else  
            band_seven_clipped(i,j) = input_seven(i,j)  
        END IF  
    end do  
end do  
  
DO I = 1,255  
    DO J = 1,255  
  
        BAND_seven(I,J) = NINT(band_seven_clipped(I,J))  
  
    END DO  
  
END DO  
  
CALL IMAGE_OUT(BAND_SEVEN,256)  
  
    TYPE*, 'BAND SEVEN COMPLETED'  
    TYPE*, '      '  
  
END
```

