(NASA-CR-179876) LANDSAT D THEMATIC MAPPER IMAGE DIMENSIONALITY REDUCTION AND GEOMETRIC CORRECTION ACCURACY Final Report (California Univ.) 111 p CSCL 05B

N87-11336

Unclas G3/43 43882

Landsat D Thematic Mapper Image Dimensionality Reduction and Geometric Correction Accuracy

Final Report
NASA Contract NAS5-27577

Gary E. Ford
Department of Electrical and Computer Engineering
University of California, Davis
Davis, CA 95616

ABSTRACT

To characterize and quantify the performance of the Landsat thematic mapper (TM), we have studied and evaluated techniques for dimensionality reduction by linear transformation and have analyzed the accuracy of the correction of geometric errors in TM images.

Theoretical evaluations and comparisons for existing methods for the design of linear transformations for dimensionality reduction are presented. These methods include the discrete Karhunen Loéve (KL) expansion, Multiple Discriminant Analysis (MDA), Thematic Mapper (TM)-Tasseled Cap Linear Transformation and Singular Value Decomposition (SVD).

A unified approach to these design problems is presented in which each method involves optimizing an objective function with respect to the linear transformation matrix. From the these studies, four modified methods are proposed. They are referred to as the Space Variant Linear Transformation, the KL Transform-MDA hybrid method, and the First and Second Version of the Weighted MDA method. The modifications involve the assignment of weights to classes to achieve improvements in the class conditional probability of error for classes with high weights.

Experimental evaluations of the existing and proposed methods have been performed using the six reflective bands of the TM data. It is shown that in terms of probability of classification error and the percentage of the cumulative eigenvalues, the six reflective bands of the TM data require only a three dimensional feature space. It is shown experimentally as well that for the proposed methods, the classes with high weights have improvements in class conditional probability of error estimates as expected.

An analysis of the accuracy of the correction of geometric errors in TM imagery is also presented. The approach to this task is to perform a ground control point (GCP) based bivariate polynomial coordinate transformation to rectify the TM image to a map projection, and to analyze the errors in this transformation.

Landsat D Thematic Mapper Image Dimensionality Reduction and Geometric Correction Accuracy

Contents

1. Introduction

The objective of our work is to characterize and quantify the performance of the Landsat thematic mapper (TM) by analyzing the quality of the image data generated by the ground data processing system. Our primary concern is with a study of the techniques for dimensionality reduction of TM data and with the analysis of the accuracy of the correction of geometric errors in TM images.

In Part I of this report, we discuss methods for dimensionality reduction. The increased dimensionality of the 7-channel Landsat TM imagery presents machine processing problems in terms of storage, analysis, and display. We have applied and analyzed objective procedures for the reduction of the data dimensionality by linear transformation. The motivations for dimensionality reduction include the attainment of simplicity of understanding, visualization, interpretation, and the retention of sufficient detail for adequate representation. The circumstances under which dimensionality reduction is required are data exploration, stabilizing the data statistical properties, aiding significance assessment, preparing the data for classification and detection of possible functional dependencies among the observations. We were primarily concerned with the use of dimensionality reduction as a means of feature selection or extraction in a pattern recognition or classification system. Our criterion for the performance of a dimensionality reduction is the classification accuracy that can be attained by processing the dimensionality reduced data.

We have analyzed, evaluated, and compared existing methods for designing the linear transformations for dimensionality reduction, including the Karhunen Loeve (KL) transform, multiple discriminant analysis (MDA), the TM-tasseled cap linear transformation, and the singular valued decomposition (SVD). From these evaluations, several modified methods are proposed, including the Space Variant Linear Transformation and the KL transform-MDA hybrid methods, and two versions of weighted MDA methods. Two design methods are proposed for the design of linear transformations for dimensionality reduction for noisy data or observations. From the theoretical studies of the existing methods and our modifications of these methods, we propose a unified approach to these design problems, in which each method involves optimizing an objective function with respect to the linear transformation matrix. The existing and proposed methods are then evaulated experimentally by applying them to Landsat TM data and performing land use classification on the dimensionality reduced data.

In section 2, we provide an introduction to dimensionality reduction in the context of pattern recognition and classification systems. In section 3, we describe and analyze existing methods for linear transformation for dimensionality reduction, including the KL transformation, a physically-based linear transformation (the TM tasseled cap), MDA, and SVD. We then provide a comparison and discussion of these related methods. In section 4, we discuss methods for dimensionality reduction for noisy observations, based on factor analysis, minimum mean squared error methods, and signal to noise ratio methods. In section 5, we describe a unified approach to dimensionality reduction by linear transformation, in which each of the methods in shown to involve the optimization of an objective function with respect to the linear transformation matrix. Objective functions are derived for each of the existing methods. In section 6, we propose a new method for dimensionality reduction, which we have called the Space Variant Linear Transformation. In this method, different linear transformations are used for different regions of the feature space, in an attempt to optimize the classification performance in each region of the feature space. In section 7, we provide an experimental evaluation of the dimensionality reduction techniques. We make performance comparisions for these techniques, based on the probability of classification error for the same reduced dimensions for each methods. Finally, in section 8, we provide and discuss several general conclusions based on our theoretical and experimental analyses of dimensionality reduction methods.

Part II of this report is concerned with the quantification of the accuracy of the correction of geometric errors in TM imagery. Our approach to this task is to perform a ground control point (GCP) based bivariate polynomial coordinate transformation to rectify the TM image to a map projection, and to analyze the errors in this transformation. In section 9, we describe a method for coordinate transformation based on the method of least squares, employing orthonormalized polynomial basis functions to avoid numerical instabilities inherent in previously described methods. Associated with this method are procedures for the analysis of the accuracy of the geometric transformation and of the rectification of the image to a map projection as a function of the number, location, and local accuracy of the GCPs used to characterize the transformation. This method is then applied to the geometric rectification of a Landsat 4 TM subscene.

Part I

Dimensionality Reduction

2. Introduction to Dimensionality Reduction

2.1. Pattern Recognition/Classification Systems

Pattern Recognition/Classification Systems have three subsystems[1] as shown in Fig.2.1. These subsystems are the transducer, the feature selector/extractor and the classifier.

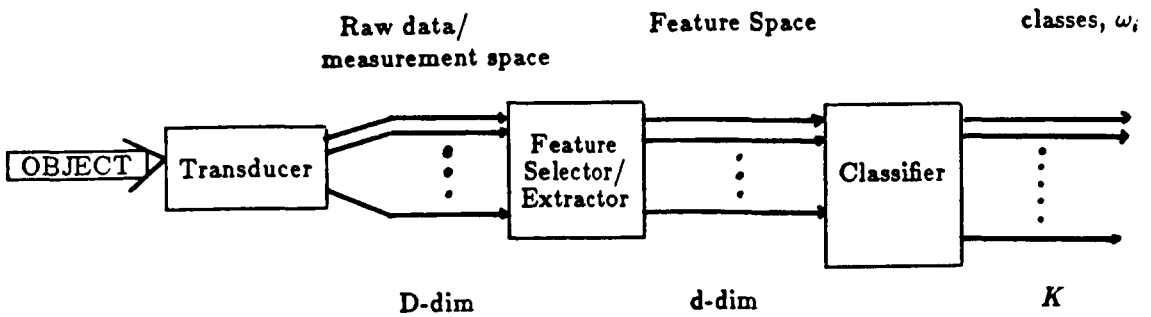

Fig.2.1. Pattern Recognition/Classification Systems

The main purpose of the feature selector/extractor is to reduce the dimensionality of the measurements produced by the transducer, by extracting *features* or *properties* of the patterns of interest. The feature selector/extractor is designed to transform the $D \times 1$ measurement vectors into $d \times 1$ feature vectors, with $d < D$, such that the probability of classification error in the d dimensional feature space is no worse than the probability of classification error in the original D dimensional measurement space. Linear transformations are widely used in feature extractors because they are analytically tractable.

2.2. Motivations for Dimensionality Reduction

The motivations for dimensionality reduction have been discussed in a large number of publications. Gnanadesikan[2] describes the issue of dimensionality reduction of data as being the attainment of simplicity of understanding, visualization, intepretation and the retention of sufficient detail for adequate representation. The circumstances under which dimensionality reduction is required are data exploration, stabilizing the data statistical properties, aiding significance assesment, preparing the data for classification and detection of possible functional dependencies among the observations. Hence the important aspects here are simplicity versus sufficient detail of representation. Fukunaga[3] defines dimensionality reduction as a product of feature extraction, which is a mapping from the original measurement space into more effective features, where effectiveness refers either to the quality of data representation or of class separability.

The motivations for reducing dimensionality cited by deViejver and Kittler[4] are to reduce the computational complexity of the classifier and to reduce the error of the parameter estimations when the number of training samples for each class is finite. They describe the second motivation as the problem of the generalization capability of the classification system, due to an increase in the number of parameters with an increase of dimensionality. For a finite number of

training samples, the accuracy of these parameter estimates might be low with respect to the true parameters of the distributions with which the classes have been modelled. The parameter estimates will be accurate, in the case of finite number of training samples, for the training samples only but might not be accurate for the rest of the data. That is why this is called the lack of generalization capability problem. However, in some cases, the design of the feature extractor to reduce the dimensionality requires the estimates of the parameters as well, so the resulting features may have the same problem. The problems, caused by the finite number of training samples, are frequently observed in practice[1], of which increasing dimensionality leads to worse rather than better classification performance.

Kanal[5] describes the goals of feature selection and extraction as (1): finding key features for regeneration or reconstruction, (2): finding features which can parsimoniously characterize the pattern, and (3): finding effective features for class discrimination or combination of the above goals. One example of a method to meet the first goal is what is known as the Karhunen Loéve expansion, which will be discussed in the following section, where the data vectors are represented by lower dimensional vectors with minimum mean squared error representations. For the second goal, one example for Landsat Thematic Mapper (TM) data are features which characterize their physical properties i.e. out of the six reflective bands, which will be discussed in detail in the following section, the data can be represented in a three dimensional feature space. These features are called[6] brightness, greenness and wetness. Example of a method to meet the third goal is the well-known method of multiple discriminant analysis (MDA) which also will be discussed in the latter section, which basically tries to provide features for which the among class scatter is maximized and the within class scatter is minimized.

Merembeck and Turner[7] state that storage requirements for large dimensional data can cause machine processing problems. Certainly if the dimensionality of the data vectors can be reduced, the storage requirements will be decreased as well. One thing worth noting, however,is to assume that the data in the original measurement space are represented in byte format for a particular dimensionality. The dimensionality is reduced but the data representation in the reduced dimensionality feature space are in real format. The machine representation of real variables requires four bytes storage, therefore, the dimensionality reduction itself might not necessarily reduce the storage requirements. The storage requirements can be reduced if the representation of the data in the reduced dimensionality feature space is also in byte format, or we need to quantize the data in the new feature space.

If we are given a set of data to be classified, there are several steps in the process that motivate the needs for dimensionality reduction, all of which have been discussed in the quoted references. The references quoted here are not exhaustive but they cover all the circumstances which require feature selection or extraction for dimensionality reduction.

A summary of these steps include:

1. Data exploration.

In this step we want to learn how the data are scattered or distributed. This may include visual observation. Certainly a managable dimensionality will simplify this process and the class training areas might be easier to find and the distribution of the class might be easier to observe. Features in the reduced dimensionality space which have a physical intepretation for a specific set of data certainly are very helpful in this step. This exploration will yield the class definitions and the training areas for each class in the supervised classification method. In unsupervised classification such as clustering, this exploration step might be done simultaneously with the classification [8]. Note that in this step, the class definitions or information may not be available yet, and therefore the method for reducing the dimensionality will be based more on data representation, or the use of features which have physical intepretations for the particular set of data.

2. Estimation of class parameters.

In this step, where the problems of a finite number of training samples have been discussed previously, dimensionality reduction can avoid the problem of lack of generalization.

3. Data classification.

Here the problem is the computational complexity of the classifier.

In general, therefore, reducing the dimensionality is closely related to achieving data simplification. The detailed motivation for reducing the dimensionality and the criterion to design the feature extractor may be different for each step in the overall classification process.

2.3. Spectral Characteristics of Thematic Mapper

The Thematic Mapper (TM) is the earth resources sensor built to improve the preceeding sensor i.e. the Multi Spectral Scanner (MSS) system. The TM is put is the satellite platform called the Landsat-D whose orbit is a circular near polar sun synchronous with a 98.22 degree inclination to the equatorial plane[9]. The altitude is about 705.3 km above equator, the equator crossing time is about 9:45 a.m. with repeat period of 16 days.

The TM has seven sensors occupying the visible, near and middle infra red and far infra red. The visible, near and middle infra red sensors measure the light reflectance of the earth while the far infra red sensor measure the thermal characteristics or temperature of the earth surface by measuring the emitted energy in that frequency band. In Table 2.1. the spectral coverages and their spatial resolutions or the Instantaneous Fields of Views (IFOV) are shown[9].

Table 2.1.
Thematic Mapper
Spectral and Spatial Characteristics

Band	Spectral Coverage (μm)	IFOV (m)
1	0.45-0.52	30
2	0.52-0.60	30
3	0.63-0.69	30
4	0.76-0.90	30
5	1.55-1.75	30
7	2.08-2.35	30
6	10.4-12.5	120

The quantization is performed toward the output data, the number of bit of the output data is eight.

From Table 2.1. we can see that assuming that the TM data are vectors then they are seven dimensional vectors. The IFOV is relatively better than the previous earth resources sensor i.e. the MSS, therefore for the same size area the number of data from the TM will be about four times of the MSS since the IFOV of the MSS is about 80 m. From this reason only the need of study of the dimensionality reduction for the TM data is justifiable.

2.4. The Objectives of the Research Project.

The first objective of the research is to perform some theoretical evaluations and comparisons of the existing methods for designing the linear transformations for dimensionality reduction. The existing methods include the Karhunen Loéve (KL) Transform, Multiple Discriminant Analysis (MDA) Method, the TM-Tasseled Cap Linear Transformation and the Singular Value Decomposition (SVD) linear transformation methods. From these evaluations, we propose some modified methods which we refer to as the Space Variant Linear Transformation and the KL

Transform-MDA Hybrid Methods, modifying the KL transform method, and the First and Second Versions of the Weighted MDA Methods, modifying the the MDA method. The modifications basically involve the assignments of weights to classes to achieve lower class conditional probability of error for the classes which are assigned high weights. We want to observe these class conditional probability of error improvements in the experiments.

Also we are proposing the two design methods for linear transformations for dimensionality reduction for noisy data or observations. They are referred to as the Minimum Mean Squared Error Criterion Based Factor Analysis, a modified version of the ordinary Factor Analysis Method, and the Signal to Noise Ratio Based Dimensionality Reduction.

From the theoretical studies of the existing methods for the design of the linear transformations for dimensionality reduction, we also propose a unified approach to these design problems, where each method will involve optimizing an objective function with respect to the linear transformation matrix .

Experimental evaluations of these existing and proposed methods are also will be performed. Final conclusions of the theoretical and experimental evaluations of the existing and the proposed methods will then be given. Because we will use the Landsat Thematic Mapper (TM) data in the experiments, the conclusions will be more applicable to the TM data rather to a general data set.

This report follows closely the Ph.D. Dissertation[10] of Kartasasmita, where part of the research associated to that Dissertation is performed to implement this research project.

References

1. Richard O. Duda and Peter E. Hart, *Pattern Classification and Scene Analysis,* John Wiley & Sons, New York, London, Sydney, Toronto, 1973.
2. R. Gnanadesikan, *Methods for Statistical Data Analysis of Multivariate Observations,* John Wiley & Sons, New York,London,Sydney,Toronto, 1977.
3. Keinosuke Fukunaga, *Introduction to Statistical Pattern Recognition,* Academic Press, New York, San Francisco, London, 1972.
4. Pierre A. deViejver and Joseph Kittler, *Pattern Recognition: A Statistical Approach,* Prentice-Hall, London, 1982.
5. Laveen Kanal, "Patterns in Pattern Recognition: 1968-1974," *IEEE Transactions on Information Theory*, vol. IT-20, no. 6, pp. 697-722, November 1974.
6. E.P. Crist and R. C. Cicone, "A Physically-Based Transformation of Thematic Mapper Data-The TM Tasseled Cap," *IEEE Transaction on Geoscience and Remote Sensing*, vol. GE-22, no. 3, May 1984.
7. Benjamin F. Merembeck and Brian J. Turner, "Directed Canonical Analysis and the Performance of Classifiers Under Its Associated Linear Transformation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. GE-18, no. 2, pp. 190-196, April 1980.
8. Gary E. Ford, V. Ralph Algazi, and Doreen I. Meyer, "A Noninteractive Procedure for Land Use Determination," *Remote Sensing of Environment*, vol. 13, no. 1, March 1983.
9. Jack L. Engel and O. Weinstein, "The Thematic Mapper-An Overview," *IEEE Transaction on Geoscience and Remote Sensing*, vol. GE-21, no. 3, July 1983.
10. Mahdi Kartasasmita, "Dimensionality Reduction by Linear Transformation for Pattern Classification with Applications to Thematic Mapper Data," *Ph. D. Dissertation*, Univ. California Davis, Davis, California, June 1986.

3. Linear Transformation for Dimensionality Reduction

3.1 Introduction

For the Feature Selector/Extractor shown in Fig.1.1, the input measurement vector or raw data vector $\underline{x}$ is a $D \times 1$ vector and the output feature vector $\underline{y}$ is a $d \times 1$ vector, where $d < D$. The general relationship between $\underline{x}$ and $\underline{y}$ is

$$\underline{y} = A\left(\underline{x}\right) \tag{3.1}$$

To determine $A\left(.\right)$, a $d \times 1$ vector function, a criterion function J must be optimized. Ideally, we want to minimimize the probability of classification error, P_e, when we use the feature vector $\underline{y}$ for the classification. This P_e should not be much larger than if we were to use the measurement or raw data vector, $\underline{x}$. In general the function $A\left(.\right)$ could be any function of the measurement vector but most existing methods[1] assume A to be a linear transformation, as follows

$$\underline{y} = A^T \underline{x} \tag{3.2}$$

where A is a $D \times d$ matrix. The general problem definition therefore is to find the $D \times d$ matrix A such that the criterion fuction $J\left(A\right)$ is optimized with respect to matrix A. The resulting transformed vector will be the $d \times 1$ vector $\underline{y}$ given in Eq.(3.2). The ideal criterion function is the probability of classification error P_e, however, methods using other criterion functions also will be considered and discussed.

Since the ideal criterion function for finding A is P_e, we will first discuss the minimum risk Bayesian classifier which, after making some assumptions, will become the minimum probability of error classifier.

To proceed, we need to establish some definitions and assumptions :

1. There are K classes ω_i; $i = 1, \ldots, K$

2. The class-conditional probability density function of $\underline{y}$, given that it comes from class ω_i is $p\left(\underline{y} \mid \omega_i\right)$.

3. The apriori probability of class ω_i is $P\left(\omega_i\right)$.

4. The mixture density of $\underline{y}$ is

$$p\left(\underline{y}\right) = \sum_{i=1}^{K} P\left(\omega_i\right) p\left(\underline{y} \mid \omega_i\right)$$

5. We assign a cost θ_{ij}, the cost of assigning $\underline{y}$ to class ω_i, where actually it is a member of class ω_j.

From Bayes rule, the aposteriori probability is

$$P\left(\omega_i \mid \underline{y}\right) = \frac{P\left(\omega_i\right) p\left(\underline{y} \mid \omega_i\right)}{p\left(\underline{y}\right)} \tag{3.3}$$

This is the probability that vector $\underline{y}$ comes from class ω_i, given the observation of $\underline{y}$. The cost or risk of making the decision to assign the observed $\underline{y}$ to class ω_i is:

$$R\left(\omega_i \mid \underline{y}\right) = \sum_{j=1}^{K} \theta_{ij} P\left(\omega_j \mid \underline{y}\right) \tag{3.4}$$

The average risk over all observations is

$$R(\omega_i) = \int R(\omega_i \mid \underline{y})\, p(\underline{y})\, d\underline{y} \tag{3.5}$$

To minimize the average risk $R(\omega_i)$ we need to minimize $R(\omega_i \mid \underline{y})$ with respect to the classes ω_i. This can be done by applying the decision rule that will assign $\underline{y}$ into class ω_i iff

$$R(\omega_i \mid \underline{y}) \leq R(\omega_j \mid \underline{y}) \quad i \neq j \tag{3.6}$$

If all classification errors impose the same risk, and there is no risk or loss in making a correct decision, the cost θ_{ij} is:

$$\theta_{ij} = \begin{cases} 1 & i \neq j \\ 0 & i = j \end{cases} \tag{3.7}$$

and the risk of assigning $\underline{y}$ to class ω_i becomes:

$$R(\omega_i \mid \underline{y}) = 1 - P(\omega_i \mid \underline{y}) \tag{3.8}$$

But since $P(\omega_i \mid \underline{y})$ is the probability that the vector $\underline{y}$ comes from class ω_i, the risk $R(\omega_i \mid \underline{y})$ is simply the probability that the vector $\underline{y}$ does not come from class ω_i. If we assign $\underline{y}$ to class ω_i, then the right hand side of Eq.(3.8) is the probability of classification error given that vector $\underline{y}$ is observed.

$$P_e(\underline{y}) = 1 - P(\omega_i \mid \underline{y})$$

The error rate or the probability of classification error, P_e, is

$$P_e = \int P_e(\underline{y})\, p(\underline{y})\, d\underline{y} = \int \left\{ 1 - P(\omega_i \mid \underline{y}) \right\} p(\underline{y})\, d\underline{y} \tag{3.9}$$

To minimize P_e the decision rule then becomes: assign $\underline{y}$ to class ω_i iff

$$P(\omega_i \mid \underline{y}) \geq P(\omega_j \mid \underline{y}) \quad i \neq j \tag{3.10}$$

and P_e now becomes

$$P_e = \int \left\{ 1 - \max_i P(\omega_i \mid \underline{y}) \right\} p(\underline{y})\, d\underline{y} \tag{3.11}$$

Given this expression for the dimensionality reduction criterion function, we return to the problem of finding the matrix A in Eq.(3.2). The objective now can be rephrased such that for a particular d, the feature vector dimension, we would like to find the matrix A such that P_e is minimized

$$\min_A P_e(A) = \min_A \int \left\{ 1 - \max_i P(\omega_i \mid A^T \underline{x}) \right\} p(A^T \underline{x})\, dA^T \underline{x} \tag{3.12}$$

The optimization of Eq.(3.12) to find the matrix A will involve multiple numerical integrations in which the maximum aposteriori probability of each vector $A^T \underline{x}$ in the integrand has to be selected and numerically minimized to find the elements of matrix A, assuming that we can find

the closed form of the probability density function $p\left(A^T \underline{x} \mid \omega_i\right)$ from a given $p\left(\underline{x} \mid \omega_i\right)$. This in general is a very complex process.

3.2 Probabilistic Distance Based Linear Transformation

Given the difficulty of finding the transformation matrix A that minimizes the probability of classification error, criterion functions which are related to the probability of error have been investigated. One such class of criterion functions are the probability distance measures[1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Those probability distance measures had been listed in several books and the current list can be found in[1] and they are :

1. Chernoff

$$J_C = -\ln \int p\left(\underline{x} \mid \omega_1\right)^s p\left(\underline{x} \mid \omega_2\right)^{1-s} d\underline{x}$$

2. Battacharya

$$J_B = -\ln \int \left[p\left(\underline{x} \mid \omega_1\right) p\left(\underline{x} \mid \omega_2\right)\right]^{1/2} d\underline{x}$$

3. Matushita

$$J_T = \left\{\int \left[\sqrt{p\left(\underline{x} \mid \omega_1\right)} - \sqrt{p\left(\underline{x} \mid \omega_2\right)}\right]^2 d\underline{x}\right\}^{1/2}$$

4. Divergence

$$J_D = \int \left[p\left(\underline{x} \mid \omega_1\right) - p\left(\underline{x} \mid \omega_2\right)\right] \ln \frac{p\left(\underline{x} \mid \omega_1\right)}{p\left(\underline{x} \mid \omega_2\right)} d\underline{x}$$

5. Pattrick-Fischer

$$J^P = \left\{\int \left[p\left(\underline{x} \mid \omega_1\right) P\left(\omega_1\right) - p\left(\underline{x} \mid \omega_2\right) P\left(\omega_2\right)\right]^2 d\underline{x}\right\}^{1/2}$$

6. Lissack-Fu

$$J_L = \int \left| p\left(\underline{x} \mid \omega_1\right) P\left(\omega_1\right) - p\left(\underline{x} \mid \omega_2\right) P\left(\omega_2\right) \right|^s \left\{p\left(\underline{x}\right)\right\}^{1-s} d\underline{x}$$

7. Kolmogorov variational distance

$$J_K = \int \left| p\left(\underline{x} \mid \omega_1\right) P\left(\omega_1\right) - p\left(\underline{x} \mid \omega_2\right) P\left(\omega_2\right) \right| d\underline{x}$$

The probabilistic distance measure can be described as an average distance between two probability density functions where the averaging is done for all possible realizations of the random vectors of those two density functions. In general these measures can be written as follows

$$J(i,j) = G\left\{\int g\left\{P\left(\omega_i\right), p\left(\underline{y} \mid \omega_i\right), P\left(\omega_j\right), p\left(\underline{y} \mid \omega_j\right), s\right\} d\underline{y}\right\} \tag{3.13}$$

Relationships with P_e, either analytically or experimentally, have been demonstrated for some of these measures, but primarily for the two class ($K=2$) problem. For example Yablon and

Chu[3] have shown the existence of strictly monotonic relationships between the minimum Bayes risk and some of the probabilistic distance measures under certain conditions, for two multivariate normal classes with equal covariance matrices.

For more than two classes[1], the criterion function is:

$$J = \sum_{i=1}^{K} \sum_{j=1}^{K} P(\omega_i)\ P(\omega_j)\ J(i,j) \tag{3.14}$$

We can substitute the vector $A^T \underline{x}$ for $\underline{y}$ and try to find A such that J of Eq.(3.14) is maximized, but from the form of Eq.(3.13) it can be seen that this will not be simple even though it might be simpler than the method using P_e as the criterion function.

For normally distributed random vectors some of the distance measures can be reduced to functions of the class conditional density parameters only[1, 7, 11]. This certainly can simplify the maximization of the criterion function J given in Eq.(3.14) to find the transformation matrix A. However, the problem of this approach is that the matrix A found from the maximization of the criterion function J is the best only for a particular dimension d. Thus we have to maximize J for several values of d and then select the matrix A that gives smallest d where the maximum value of J will not change much more by increasing the value of d. Therefore the use of the probabilistic distance measures is more appropriate for feature selection, for example,as shown in[10], rather than for feature extraction.

We conclude that the probabilistic distance criterion functions lead to computational difficulties in the determination of the optimal linear transformation matrix A. To find computationally tractable methods, other criterion functions must be considered. Basically we will discuss three approaches, where the first minimizes the mean squared error (MSE) of the representation of the measurement vector by the dimensionality reduced features. The second approach maximizes the class separability by defining the criterion function to be the ratio of between class scatter to within class scatter. The third approach uses the invariant property of the minimum probability of classification error of the multivariate Gaussian classes for the dimensionality reduced data vectors by linear transformation. The third approach, under certain conditions, has a relationship with the probability of classification error. However, for the first and the second approaches, the formal relationship between these criterion functions and the probability of classification error has not been established. But it is apparent that a relationship exists, so these approaches have some merit.

These methods will be discussed in the following sections. First, we discuss the minimum MSE representation, known as Karhunen Loéve (KL) expansion, followed by a discussion of the method of maximization of a scatter ratio, known as Multiple Discriminant Analysis (MDA) or Fisher's Linear Discriminant Analysis, and conclude with a discussion of the third approach, known as the Singular Value Decomposition Linear Transformation (SVDLT) method.

3.3 Discrete Karhunen Loéve Expansion

The objective of the discrete KL expansion is to find a set of orthonormal basis vectors such that if some of the basis vectors are *discarded,* the approximate representation of the measurement vector $\underline{x}$ will have minimum MSE. The criterion function in this method therefore is the MSE, and the intended matrix A will have the orthonormal basis vectors as its columns. The word discarded is put in Italics because in one of the representations, to be discussed in Sec.3.3.2., some of the basis vectors are not really discarded.

3.3.1 The First KL Representation

For a given $D \times 1$ random vector $\underline{x}$,

$$\underline{x} = \sum_{i=1}^{D} y_i \, \underline{a}_i \tag{3.15}$$

where $\underline{a}_i$ are the orthonormal basis vectors and y_i are the linear coefficients,

$$y_i = \underline{x}^T \underline{a}_i \tag{3.16}$$

The approximation representation[1] is given by

$$\hat{\underline{x}} = \sum_{i=1}^{d} y_i \, \underline{a}_i \tag{3.17}$$

where $d < D$. The elements of the $d \times 1$ vector $\underline{y}$ are the linear coefficients $\left\{ y_i \right\}$ given by Eq.(3.16). By assigning the d basis vectors $\underline{a}_i$ of Eq.(3.17) as the columns of the $D \times d$ matrix A, the expression for the vector $\underline{y}$ is again given by Eq.(3.2). The problem now is to find $\underline{a}_i$ such that the MSE of the representation, chosen to be the criterion function,

$$e = E\left\{ \left(\underline{x} - \hat{\underline{x}}\right)^T \left(\underline{x} - \hat{\underline{x}}\right) \right\} \tag{3.18}$$

is minimized with respect to each basis vector $\underline{a}_i$. The minimization yields an eigen equation of the autocorrelation matrix, $R_{\underline{x}}$, i.e.

$$R_{\underline{x}} \, \underline{a}_i = \lambda_i \, \underline{a}_i \tag{3.19}$$

Thus the set of orthonormal basis vectors are the eigenvectors of $R_{\underline{x}}$. The MSE becomes

$$e = \sum_{i=d+1}^{D} \lambda_i \tag{3.20}$$

Therefore if the eigenvalues λ_i are ordered

$$\lambda_1 \geq \cdots \geq \lambda_i \geq \cdots \geq \lambda_D \tag{3.21}$$

and we discard the eigenvectors $\underline{a}_i$ associated eith the $D-d$ smallest eigenvalues, the minimum MSE representation is achieved.

Using the $\underline{a}_i$ for $1 \leq i \leq d$ as the basis vectors, the $D \times 1$ random vector $\underline{x}$ can be approximately represented by the $d \times 1$ random vector $\underline{y}$,

$$\underline{y} = A^T \underline{x} \tag{3.22}$$

where A is a $D \times d$ matrix, having columns being the selected basis vectors $\underline{a}_i$,

$$A = \left(\underline{a}_1, \ldots, \underline{a}_i, \ldots, \underline{a}_d \right) \tag{3.23}$$

3.3.2 The Second KL Representation

The second approximation representation is[7]

$$\tilde{\underline{x}} = \sum_{i=1}^{d} y_i \, \underline{a}_i + \sum_{i=d+1}^{D} b_i \, \underline{a}_i \tag{3.24}$$

where $d < D$, and b_i are unknown constants, with the orthonormal expansion of $\underline{x}$ again given by Eq.(3.15), having the linear coefficients y_i given by Eq.(3.16). For reasons that will be discussed later the $D \times 1$ vector $\underline{x}$ can be represented by a $d \times 1$ vector $\underline{y}$ whose elements are the linear coefficients y_i given by Eq.(3.16). This vector $\underline{y}$ can be found by taking the linear transformation of the vector $\underline{x}$ using the matrix A whose columns are the set of selected orthonormal basis vectors $\underline{a}_i$ exactly following Eq.(3.2).

The problem now is to find $\underline{a}_i$ and b_i such that the MSE, which is similar to one in the first representation

$$\tilde{e} = E\left\{ (\underline{x} - \tilde{\underline{x}})^T (\underline{x} - \tilde{\underline{x}}) \right\} \tag{3.25}$$

is minimized. This yields an eigen equation of the covariance matrix of $\underline{x}$, $\Sigma_{\underline{x}}$,

$$\Sigma_{\underline{x}} \underline{a}_i = \beta_i \underline{a}_i \tag{3.26}$$

and the constants b_i are given by

$$b_i = \underline{a}_i^T \underline{m}_{\underline{x}} \tag{3.27}$$

where $\underline{m}_{\underline{x}}$ is the mean vector of $\underline{x}$. Thus the set of orthonormal basis vectors $\left\{ \underline{a}_i \right\}$ are the eigenvectors of $\Sigma_{\underline{x}}$. The MSE becomes

$$\tilde{e} = \sum_{i=d+1}^{D} \beta_i \tag{3.28}$$

Therefore if the eigenvalues β_i are ordered

$$\beta_1 \geq \cdots \geq \beta_i \geq \cdots \geq \beta_D \tag{3.29}$$

and using the $\underline{a}_i$ for $i \leq i \leq d$ as the basis vectors for the linear coefficients y_i and using the constants b_i for the rest of the basis vectors, the random vector $\underline{x}$ can be approximately represented with minimum MSE by a $D \times 1$ random vector $\hat{\underline{y}}$,

$$\hat{\underline{y}} = \begin{bmatrix} \underline{y} \\ \underline{b} \end{bmatrix} \tag{3.30}$$

where

$$\underline{y} = A^T \underline{x} \tag{3.31}$$

$$\underline{b} = A_{d+1,D}^T \underline{m}_{\underline{x}} \tag{3.32}$$

and where

$$A = \left(\underline{a}_1, \cdots, \underline{a}_i, \cdots, \underline{a}_d \right) \tag{3.33}$$

$$A_{d+1,D} = \left(\underline{a}_{d+1}, \cdots, \underline{a}_j, \cdots, \underline{a}_D \right)$$

For classification purposes, since the $(D-d) \times 1$ constant vector $\underline{b}$ is the same for all $\underline{x}$, we can just use the $d \times 1$ vector $\underline{y}$ to represent $\underline{x}$, as given by Eq.(3.31) which is exactly the same as the one given by Eq.(3.2).

From the above discussion, the approximation representations given by Eq.(3.17) and Eq.(3.24) will be the same if the random vector $\underline{x}$ has zero mean. But if $\underline{x}$ is not a zero mean

random vector, the mean squared errors defined in Eq.(3.20) and Eq.(3.28) will be different. In the following section, it will be shown that the approximation given by Eq.(3.24) will have smaller mean squared error.

3.3.3 The Choice of the KL Representation

The two preceeding sections show two different KL representations. We want to compare those two representations to find the one that has the minimum MSE representation. Before making the comparison, we will show that the eigen equation of the covariance matrix $\Sigma_{\underline{x}}$ given in Eq.(3.26) implies that any other set of orthonormal basis vectors will not yield a minimum mean squared error representation. We will show that it is true, and to do this we have to show that

$$t_{\max} \leq \beta_{\max} \tag{3.34}$$

and

$$t_{\min} \geq \beta_{\min} \tag{3.35}$$

where

$$t_i = \underline{v}_i^{\,T} \Sigma_{\underline{x}} \underline{v}_i \tag{3.36}$$

and

$$\sum_{i=1}^{D} t_i = \sum_{i=1}^{D} \beta_i = Trace\ \Sigma_{\underline{x}} \tag{3.37}$$

where $\underline{v}_i$'s are a set of arbitrary orthonormal basis vectors.

Define a $D \times D$ matrix A_D as follows

$$A_D = \left(\underline{a}_1, \ldots, \underline{a}_i, \ldots, \underline{a}_D \right) \tag{3.38}$$

where $\underline{a}_i$ is the eigenvector of the matrix $\Sigma_{\underline{x}}$ and define a $D \times D$ matrix V_D as follows

$$V_D = \left(\underline{v}_1, \ldots, \underline{v}_i, \ldots, \underline{v}_D \right) \tag{3.39}$$

From the eigen equation Eq.(3.26) we have

$$\Sigma_{\underline{x}} = A_D\, Diag\left(\beta_i\right) A_D^{\,T} \tag{3.40}$$

Substituting $\Sigma_{\underline{x}}$ from Eq.(3.40) into Eq.(3.36) yields

$$\begin{aligned} t_i &= \underline{v}_i^{\,T} A_D\, Diag\left(\beta_i\right) A_D^{\,T} \underline{v}_i \\ &= \sum_{k=1}^{D} \sum_{j=1}^{D} \sum_{l=1}^{D} v_{ij}\, v_{il}\, a_{kj}\, a_{kl}\, \beta_k \\ &= \sum_{k=1}^{D} \beta_k \left(\sum_{j=1}^{D} v_{ij}\, a_{kj} \right)^2 \end{aligned} \tag{3.41}$$

where

$$v_{ij} = \left\{ \underline{v}_i \right\}_j$$

$$a_{kj} = \left\{ A_D \right\}_{jk} \tag{3.42}$$

If we substitute the β_k in Eq.(3.41) by $\beta_{\max}$, we will get the inequality

$$t_i \leq \beta_{\max} \sum_{j=1}^{D} \sum_{l=1}^{D} v_{ij}\, v_{il} \sum_{k=1}^{D} a_{kj}\, a_{kl} \tag{3.43}$$

It is further known that:

$$A_D{}^T A_D = A_D A_D{}^T = I_D \tag{3.44}$$

where I_D is a $D \times D$ identity matrix. Therefore from the identity given in Eq.(3.44) we have

$$\sum_{k=1}^{D} a_{kj}\, a_{kl} = \delta_{jl} \tag{3.45}$$

If we substitute the summation in Eq.(3.43) by the one in Eq.(3.45), the inequality in Eq.(3.43) becomes

$$t_i \leq \beta_{\max} \sum_{j=1}^{j=D} v_{ij}{}^2 = \beta_{\max}$$

Therefore

$$t_i \leq \beta_{\max}$$

for $i = 1, \ldots, D$.

Similarly,

$$t_i \geq \beta_{\min} \sum_{j=1}^{D} \sum_{l=1}^{D} v_{ij}\, v_{il} \sum_{k=1}^{D} a_{kj}\, a_{kl} \tag{3.46}$$

which yields

$$t_i \geq \beta_{\min}$$

for $i = 1, \ldots, D$. Therefore the inequalities in Eq.(3.34) and (3.35) are true.

Now we want to prove Eq.(3.37), using the definition of t_i in Eq.(3.41) :

$$\sum_{i=1}^{D} t_i = \sum_{i=1}^{D} \sum_{k=1}^{D} \beta_k \sum_{j=1}^{D} \sum_{l=1}^{D} v_{ij}\, v_{il}\, a_{kl}\, a_{kl} \tag{3.47}$$

But from the orthonormality and the completeness of the set of the $\underline{v}_i$ vectors, we will have

$$V_D{}^T V_D = V_D\, V_D{}^T = I_D \tag{3.48}$$

which means that

$$\sum_{i=1}^{i=D} v_{ij}\, v_{il} = \delta_{jl} \tag{3.49}$$

Substituting the summation in Eq.(3.49) into Eq.(3.47) yields

$$\sum_{i=1}^{D} t_i = \sum_{k=1}^{D} \beta_k \sum_{j=1}^{D} a_{kj}{}^2 = \sum_{k=1}^{D} \beta_k = Trace\ \Sigma_{\underline{x}} \tag{3.50}$$

Therefore Eq.(3.37) is also true.

Now if the t_i and β_i are ordered as follows,

$$t_{max}=t_1 \geq, \ldots, \geq t_i \geq, \ldots, \geq t_D = t_{min}$$
$$\beta_{max}=\beta_1 \geq, \ldots, \geq \beta_i \geq, \ldots, \geq \beta_D = \beta_{min} \tag{3.51}$$

When we plot those t_i's and β_i's, see Fig.3.1., the *cross over* must exist.

In the region to the left of the cross over, the eigenvalues β_i will always be larger than t_i. If for example we select the $\underline{a}_i$ and $\underline{v}_i$ up to $i=k$, where k is to the left of the cross over, the mean squared error of the selected set of $\underline{a}_i$ is

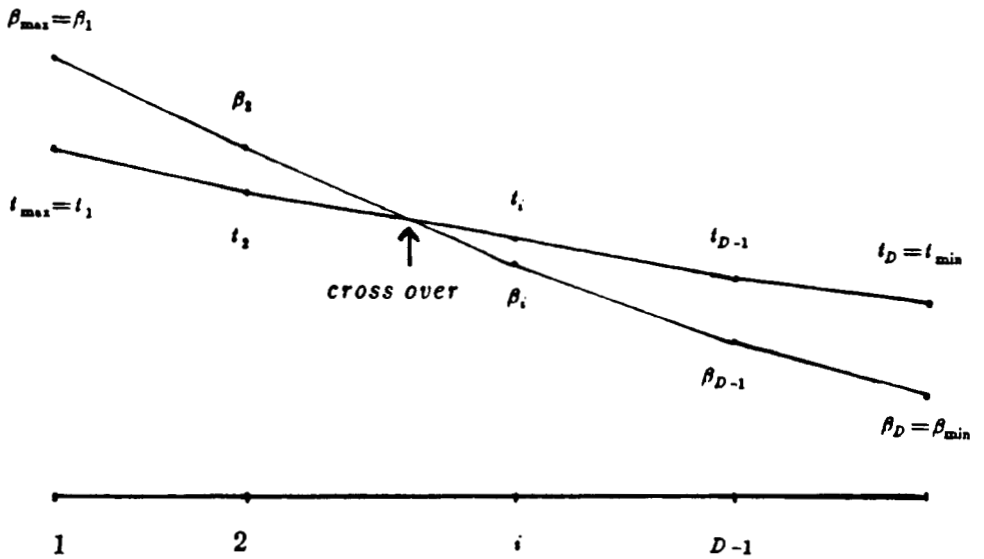

Fig.3.1. The plot of β_i and t_i.

$$e_{\underline{a}}(k)=\sum_{i=k+1}^{D}\beta_i = Trace\ \Sigma_{\underline{x}} - \sum_{i=1}^{k}\beta_i$$

and the mean squared error of the selected set of $\underline{v}_i$ is

$$e_{\underline{v}}(k)=\sum_{i=k+1}^{D} t_i = Trace\ \Sigma_{\underline{x}} - \sum_{i=1}^{k} t_i$$

Since for i up to k, β_i is always larger than or equal to t_i, then

$$e_{\underline{a}}(k) \leq e_{\underline{v}}(k) \tag{3.52}$$

For the region to the right of the cross over, the β_i are always smaller than or equal to t_i, which will yield a similar situation as in Eq.(3.52), and therefore the proof is complete.

Now we can proceed with determining which of the two KL representations has the minimum mean squared error. It is known that for a random vector $\underline{x}$,

$$R_{\underline{x}} = \Sigma_{\underline{x}} + \underline{m}_{\underline{x}}\,\underline{m}_{\underline{x}}^{T} \tag{3.53}$$

Suppose now we select d eigenvectors $\hat{\underline{a}}_i$ of the autocorrelation matrix $R_{\underline{x}}$ associated with the d largest eigenvalues λ_i. The MSE will be, rewriting Eq.(3.20),

$$e_{R_{\underline{x}}} = \sum_{i=d+1}^{D}\lambda_i \tag{3.54}$$

and this is the same as, using Eq.(3.53),

$$e_{R_{\underline{x}}} = \sum_{i=d+1}^{D} \underline{\hat{a}}_i^T R_{\underline{x}} \underline{\hat{a}}_i = \sum_{i=d+1}^{D} \underline{\hat{a}}_i^T \Sigma_{\underline{x}} \underline{\hat{a}}_i + \sum_{i=d+1}^{D} \underline{\hat{a}}_i^T \underline{m}_{\underline{x}} \underline{m}_{\underline{x}}^T \underline{\hat{a}}_i \tag{3.55}$$

Following the preceeding discussion, the first summation in the last equality of Eq.(3.55) is always larger than or equal to the MSE if we select the d eigenvectors $\underline{a}_i$ of the covariance matrix $\Sigma_{\underline{x}}$ associated with the d largest eigenvalues β_i, or,

$$e_{\Sigma_{\underline{x}}} = \sum_{i=d+1}^{D} \underline{a}_i^T \Sigma_{\underline{x}} \underline{a}_i \leq \sum_{i=d+1}^{D} \underline{\hat{a}}_i^T \Sigma_{\underline{x}} \underline{\hat{a}}_i \tag{3.56}$$

This is always true independently of how we select the d vectors $\underline{\hat{a}}_i$, because these vectors are not the eigenvectors of the covariance matrix $\Sigma_{\underline{x}}$. Moreover, the second summation in the last equality of Eq.(3.55) will yield a number that is always larger than or equal to zero, because of

$$\underline{\hat{a}}_i^T \underline{m}_{\underline{x}} \underline{m}_{\underline{x}}^T \underline{\hat{a}}_i = \left(\underline{\hat{a}}_i^T \underline{m}_{\underline{x}} \right)^2 \geq 0 \tag{3.57}$$

Therefore,

$$e_{\Sigma_{\underline{x}}} \leq e_{R_{\underline{x}}} \tag{3.58}$$

which means that the MSE of the second KL representation, shown in Eq.(3.24), for a given reduced dimensionality d, will always be smaller than or equal to the MSE of the first KL representation, shown in Eq.(3.17). From now on, when we refer to the KL transform or expansion we will be referring to the second representation, i.e. the one that uses the eigenvectors of the covariance matrix $\Sigma_{\underline{x}}$.

The property of KL expansion i.e.

$$\sum_{i=d+1}^{i=D} \beta_i \leq \sum_{i=d+1}^{i=D} t_i ,$$

where the t_i are from the set of arbitrary orthonormal basis vectors but not the eigenvectors of $\Sigma_{\underline{x}}$, implies that the KL transform yields the best variance or energy compaction. This also implies that the representation entropy[1, 12] produced by KL transform is minimum, where the representation entropy is defined as :

$$H_R = -\sum_{i=1}^{D} \tilde{\beta}_i \log \tilde{\beta}_i \tag{3.59}$$

where

$$\tilde{\beta}_i = \frac{\beta_i}{Trace\, \Sigma_{\underline{x}}}$$

If we use for the $D \times d$ transformation matrix A in Eq.(3.2) the set of the eigenvectors of the covariance matrix $\Sigma_{\underline{x}}$, the covariance matrix of the transformed vector $\underline{y}$ will be

$$\Sigma_{\underline{y}} = A^T \Sigma_{\underline{x}} A = Diag\left(\beta_i ; i = 1, \ldots, d\right) \tag{3.60}$$

which indicates that the resulting transformed vector $\underline{y}$ has uncorrelated elements.

The diagonalization of the covariance matrix $\Sigma_{\underline{x}}$ by the KL transform, shown in Eq.(3.60), implies an interesting property. If the covariance matrix of the original $D \times 1$ data vectors $\underline{x}$ is already a diagonal matrix then the KL transformation would be identity. On the other hand, if

the original data vectors $\underline{x}$ are highly correlated which can be indicated by high absolute values of the off-diagonal elements of the matrix $\Sigma_{\underline{x}}$ i.e. the absolute value of $\left\{\Sigma_{\underline{x}}\right\}_{ij}$ is close to (but never larger than) $\left(\left\{\Sigma_{\underline{x}}\right\}_{ii}\left\{\Sigma_{\underline{x}}\right\}_{jj}\right)^{1/2}$ where $\left\{\Sigma_{\underline{x}}\right\}_{ii}$ is the i^{th} diagonal element, then the diagonalization of the covariance matrix, or the KL transformation, will yield an effective dimensionality reduction. We can intepret the high correlation as high redundancy in the data representation. Therefore the KL transform method can reduce the dimensionality effectivelly by removing the redundancy but with minimum MSE representation.

3.3.4 The Physically-Based Linear Transformation of the TM Data - TM Tasseled Cap.

The transformed features which have identifiable physical properties for TM data have been introduced by Crist and Cicone[13, 14, 15]. They show that the information content of the six reflective bands of the Landsat TM earth images (see Sec.2 for the spectral characteristics of the TM data) lie primarily in a three dimensional space.

Those three features are refered to as:

1. Brightness, which is the weighted sum of the six reflective bands such that it will respond to the change in the total reflectance and those physical processes which affect the total reflectance, such as the particle size of the soil.
2. Greenness, which is the contrast between the weighted sum of the visible bands and the near infrared band. This feature responds to the combination of the low reflectance in the visible bands and high reflectance in the near infrared band of the green vegetation. In this feature the two longer infrared bands practically cancell each other.
3. Wetness, which is the contrast of the weighted sum of the visible and near infrared bands with the two longer infrared bands. Since it has been suggested that the longer infrared bands are sensitive to the soil and plant moisture, then this feature is expected to be responsive to the change of moisture content of the object.

The brightness and greenness features support a plane refered to as the Plane of Vegetation and the brightness and wetness features support a plane refered to as the Plane of Soil. These planes are perpendicular to each other. The TM data are in the space supported by these two planes, which is refered to as the Transition Region, where the position of a datum depends on the characteristics of the object represented by that datum with respect to the physical properties of the three features. Besides the three major features, there are three less significant features. However, the vast majority of the TM data lie in the three dimensional space described above.

The three major features are usually referred to as the TM taselled cap linear transformation because it is found through the extension of the taselled cap concept used for MSS data[16] which states that the MSS data are lie in a two dimensional space spanned by two orthogonal features indicating the physical properties of agricultural objects. These TM taselled cap features, however,[13] cannot be found by the KL transform method because the resulting features of the KL transform method do not necessarily have physical intepretations. However the TM tasseled cap features are found by first applying the KL transform to the original data of the six reflective bands. Those KL features are then rotated two or three at a time while maintaining the orthogonality of those features, and associating the variations of the resulting rotated transformed data with the physical characteristics of the crop canopy or soil.

In terms of data classification, this linear transformation has a similarity with the one found through KL transform method, both having more concern with data representation, i.e. the TM taselled cap is concerned with the physical properties and the KL transform is concerned with the minimum MSE representation, instead of class separations. Therefore these two methods may be most useful in the data exploration step.

3.4 Linear Transformation Based on Scatter Matrices

Instead of directly using the class conditional density functions to find the linear transformation as had been suggested in the General Introduction there is a well known method that uses the class mean vectors and covariance matrices. This method is usually called Fisher's Linear Discriminant Analysis or Multiple Discriminant Analysis (abbreviated by MDA)[17, 18, 19, 20].

The objective of the MDA approach is to find a vector or a set of vectors $\underline{a}_i$, such that if the random vector $\underline{x}$ is projected on $\underline{a}_i$ the ratio of among class to within class scatter is maximized.

The objective can be written as,

$$\max_{\underline{a}_i} \alpha_i = \max_{\underline{a}_i} \frac{\underline{a}_i^T S_A \underline{a}_i}{\underline{a}_i^T S_W \underline{a}_i} \tag{3.61}$$

where S_A is the among class scatter matrix and S_W is the within class scatter matrix.

3.4.1 Definition and Analysis

The among class scatter matrix S_A is defined as

$$S_A = \sum_{i=1}^{K} P(\omega_i) (\underline{m}_i - \underline{m}) (\underline{m}_i - \underline{m})^T \tag{3.62}$$

and the within class scatter matrix S_W is defined as,

$$S_W = \sum_{i=1}^{K} P(\omega_i) C_i \tag{3.63}$$

where C_i is the covariance matrix of class ω_i.

Both S_A and S_W are $D \times D$ symmetric matrices but S_A is at least semi positive definite. However S_W is positive definite since it is assumed that the class covariance matrices, C_i, are positive definite. Therefore the within class scatter matrix, S_W, is non singular.

Maximizing, more generally, optimizing α_i of Eq.(3.61) to find $\underline{a}_i$ can be done by taking the first derivative of α_i with respect to vector $\underline{a}_i$ and setting it equal to the zero vector. This yields,

$$S_A \underline{a}_i = \alpha_i S_W \underline{a}_i \tag{3.64}$$

$$S_W^{-1} S_A \underline{a}_i = \alpha_i \underline{a}_i \tag{3.65}$$

Eq.(3.65) is an eigen equation of the matrix $S_W^{-1} S_A$. From this we can draw the conclusion that there will be D values of α_i which satisfy Eq.(3.64). Moreover, from Eq.(3.61), the semi positive definiteness of S_A, and the positive definiteness of S_W, α_i is always larger than or equal to zero.

We can now order the eigenvalues,

$$\alpha_1 \geq, \ldots, \geq \alpha_i \geq, \ldots, \geq \alpha_D \tag{3.66}$$

and select the d eigenvectors $\underline{a}_i$ associated with the d largest eigenvalues α_i and create the $D \times d$ matrix A as follows

$$A = \left[\underline{a}_1, \ldots, \underline{a}_d \right] \tag{3.67}$$

where $\underline{a}_i$ is the eigenvector associated with α_i.

From the ratio of Eq.(3.61), the vector $\underline{a}_i$ which satisfies Eq.(3.64) and also Eq.(3.65) is independent of its norm, and therefore we can let the norm, $\| \underline{a}_i \| = \left(\underline{a}_i^T \underline{a}_i \right)^{1/2}$, equal to one.

$$\| \underline{a}_i \|^2 = \left(\underline{a}_i^T \underline{a}_i \right) = 1 \tag{3.68}$$

Using Eq.(3.68) and (3.65) we have the relationship

$$\underline{a}_i^T S_W^{-1} S_A \underline{a}_i = \alpha_i \tag{3.69}$$

However, we also can factor the matrix S_W,

$$S_W = \left(S_W^{1/2} \right)^T \left(S_W^{1/2} \right) \tag{3.70}$$

and

$$S_W^{-1/2} = \left(S_W^{1/2} \right)^{-1} \tag{3.71}$$

where the existences of the factorization of Eq.(3.70) and the inverse of the factor given by Eq.(3.71) are guaranteed because S_W is a symmetric positive definite matrix. Using Eq.(3.70) and (3.71) in Eq.(3.64) yields another eigen equation,

$$\left(S_W^{-1/2} \right)^T S_A S_W^{-1/2} \hat{\underline{a}}_i = \alpha_i \hat{\underline{a}}_i \tag{3.72}$$

where

$$\hat{\underline{a}}_i = \left(S_W^{1/2} \right) \underline{a}_i \tag{3.73}$$

or

$$\underline{a}_i = \left(S_W^{-1/2} \right) \hat{\underline{a}}_i \tag{3.74}$$

We can again constrain the norm of the eigenvector $\hat{\underline{a}}_i$,

$$\| \hat{\underline{a}}_i \|^2 = \hat{\underline{a}}_i^T \hat{\underline{a}}_i = 1 \tag{3.75}$$

and from this we will have the relationship

$$\underline{a}_i^T S_W \underline{a}_i = 1 \tag{3.76}$$

Using Eq.(3.75), (3.74) and (3.72) we will have another relationship

$$\underline{a}_i^T S_A \underline{a}_i = \alpha_i \tag{3.77}$$

We can now select the d vectors $\underline{a}_i$ to create the $D \times d$ transformation matrix A,

$$A = \left[\underline{a}_1, \ldots, \underline{a}_d \right] = S_W^{-1/2} \hat{A} \tag{3.78}$$

where

$$\hat{A} = \left[\underline{\hat{a}}_1, \ldots, \underline{\hat{a}}_d \right] \tag{3.79}$$

and $\underline{\hat{a}}_i ; i = 1, \ldots, d$ are the eigenvectors of Eq.(3.72) associated with the d largest eigenvalues α_i. Also, since the matrix involved in the eigen equation of Eq.(3.72) is symmetric, the eigenvectors $\underline{\hat{a}}_i$ are orthogonal i.e.,

$$\underline{\hat{a}}_i^{\,T} \underline{\hat{a}}_j = \delta_{ij} \tag{3.80}$$

Therefore using Eq.(3.78), (3.79), (3.80) and (3.72) we can conclude that

$$A^T S_A A = Diag\left(\alpha_i \; ; i = 1, \ldots, d\right) \tag{3.81}$$

and using Eq.(3.78), (3.79), (3.76) and (3.80) we also can conclude that

$$A^T S_W A = I_d \tag{3.82}$$

Thus the transformation matrix A defined in Eq.(3.78) will simultaneously diagonalize the matrices S_A and S_W.

The resulting transformation is then

$$\underline{y} = A^T \underline{x} \tag{3.83}$$

where $\underline{y}$ is an $d \times 1$ vector.

In this MDA method, the class mean vectors are important parameters for class separation which consequently the distribution of the data of each class has to be unimodal[20]. Therefore unimodal analysis of each class is a very important step before applying this method. The unimodality of the distribution of data of each class can be observed from the class training samples, and if the training samples of a class shows that the distribution is not unimodal then that class should be devided into subclasses where each subclasses is unimodal[21] and then those classes will be considered as ordinary classes.

From the relationships among the matrices $\Sigma_{\underline{x}}$, S_A and S_W the transformation matrix A defined in Eq.(3.78) will diagonalize the covariance matrix $\Sigma_{\underline{x}}$ or the transformed features will be uncorrelated. However, the column vectors of matrix A are not orthonormal. This may be contrasted with the transformation matrix produced by the KL expansion method which also yields uncorrelated features but the column vectors of the transformation matrix are orthonormal.

If $S_W = I_D$, the transformation matrices produced by the MDA and KL expansion methods will be the same. Also, if the number of classes, K, is very large but the space can still accomodate the scatter of each class, then the scatter of data of each class around its mean vector becomes very small which yields the matrix S_W approaching the zero matrix. In this case the MDA and KL expansion methods will yield the same transformation matrix.

In general if the number of classes is large the resulting features of MDA method might not be accurate[11] for separating the classes. However, for two equal covariance classes with equal apriori probabilities, the resulting feature will give minimum probability of error using a minimum Euclidian distance classifier.

3.5 Singular Value Decomposition Linear Transformation Method

3.5.1 Invariant Property of Bayes Classification by Linear Transformation

For multivariate Gaussian classes,[22, 23] Decell *et al* and Peters *et al* have shown the condition of the linear transformation matrix A such that the dimensionality reduction using the matrix A will not change the probability of error. Where the classifier is either the maximum likelihood or maximum aposteriori classifier. That condition and the proof will be discussed in the following.

For a $D \times d$ full rank matrix A, it is given that

$$A\left(A^T A\right)^{-1} A^T \underline{m}_{\underline{x}i} = \underline{m}_{\underline{x}i} \tag{3.84}$$

and

$$A\left(A^T A\right)^{-1} A^T \left(\Sigma_{\underline{x}i} - I\right) = \left(\Sigma_{\underline{x}i} - I\right) \tag{3.85}$$

where $\underline{m}_{\underline{x}i}$ and $\Sigma_{\underline{x}i}$ is the mean vector and the covariance matrix of $\underline{x} \epsilon \omega_i$ respectively.
If we are also given that

$$\max_j P\left(\omega_j\right) p\left(\underline{x} \mid \omega_j\right) = P\left(\omega_i\right) p\left(\underline{x} \mid \omega_i\right) \tag{3.86}$$

then for the transformation,

$$\underline{y} = A^T \underline{x} \tag{3.87}$$

we will have,

$$\max_j P\left(\omega_j\right) p\left(\underline{y} \mid \omega_j\right) = P\left(\omega_i\right) p\left(\underline{y} \mid \omega_i\right) \tag{3.88}$$

or the class assignment of the untransformed vector $\underline{x}$ and the transformed vector $\underline{y}$ will be the same.

Before we prove the above result, we want to observe characteristic of a matrix defined as follows,

$$P = I - Q = I - A\left(A^T A\right)^{-1} A^T \tag{3.89}$$

where the matrix A is the one of Eq.(3.84) and (3.85).
The matrix P is a $D \times D$ symetric idempotent matrix, therefore its eigenvalues are[24] either one or zero.

Assume now we have[18] a $D \times 1$ vector $\underline{w}$ such that

$$A^T \underline{w} = \underline{0} \tag{3.90}$$

then

$$P\underline{w} = \underline{w} \tag{3.91}$$

or $\underline{w}$ is the eigenvector of the matrix P with multiple eigenvalues one.

Assume now we have a $D \times 1$ vector $\underline{v}$ such that

$$\underline{v} = A\tilde{\underline{v}} \tag{3.92}$$

where $\tilde{\underline{v}}$ is a $d \times 1$ vector, which means that $\underline{v}$ is linear combination of the columns of the matrix A, then

$$P\underline{v} = \underline{0} \tag{3.93}$$

or $\underline{v}$ is the eigenvector of the matrix P with multiple eigenvalues zero and thus the columns of A are also the eigenvectors of P with multiple eigenvalues zero.

Since the size of the matrix A is $d \times D$, therefore there will be d number zero eigenvalues and $D-d$ number of one eigenvalues of matrix P. Hence the rank of matrix P is $D-d$.

Proof of Eq.(3.88) will be given as follows. First we will create a full rank $D \times D$ matrix $\tilde{A}$,

$$\tilde{A} = \left[A \, , \, R \right] \tag{3.94}$$

where the matrix A is shown in Eq.(3.84) and (3.86) and the $D \times \left(D-d \right)$ matrix R is defined as,

$$R = P \; C \tag{3.95}$$

where C is an arbitrary full rank $D \times \left(D-d \right)$ matrix and the matrix P is given in Eq.(3.89). Observe that,

$$A^T R = A^T P \; C = A^T \left(I - A \left(A^T A \right)^{-1} A^T \right) \; C = \underline{0} \tag{3.96}$$

which means that the columns of A and of R are orthogonal confirming that the matrix $\tilde{A}$ is full rank. And since the matrix C is arbitrary, except it has to be full rank, then the matrix R is still general.

We will transform the vector $\underline{x}$ by the matrix $\tilde{A}$,

$$\underline{\tilde{y}} = \tilde{A}^T \underline{x} = \begin{bmatrix} A^T \underline{x} \\ R^T \underline{x} \end{bmatrix} = \begin{bmatrix} \underline{y} \\ \underline{z} \end{bmatrix} \tag{3.97}$$

The class mean vectors of $\underline{z}$ is

$$\underline{m}_{\underline{z}i} = R^T \underline{m}_{\underline{x}i} = C^T \left(\underline{m}_{\underline{x}i} - A \left(A^T A \right)^{-1} A^T \underline{m}_{\underline{x}i} \right) \tag{3.98}$$

but from Eq.(3.84) we have

$$\underline{m}_{\underline{z}i} = C^T \left(\underline{m}_{\underline{x}i} - \underline{m}_{\underline{x}i} \right) = \underline{0} \tag{3.99}$$

or the class mean vectors of $\underline{z}$ are the same for all classes i.e. they are zero vectors.
The class covariance matrix of $\underline{z}$ is

$$\begin{aligned} \Sigma_{\underline{z}i} &= R^T \Sigma_{\underline{x}i} R = C^T \left(I - A \left(A^T A \right)^{-1} A \right) \Sigma_{\underline{x}i} R = \\ &= C^T \left(\Sigma_{\underline{x}i} - A \left(A^T A \right)^{-1} A \, \Sigma_{\underline{x}i} \right) R \end{aligned} \tag{3.100}$$

Substituting Eq.(3.85) into Eq.(3.100) yields

$$\begin{aligned} \Sigma_{\underline{z}i} &= C^T \left(\Sigma_{\underline{x}i} - \Sigma_{\underline{x}i} + I - A \left(A^T A \right)^{-1} A^T \right) R = \\ &= C^T \left(I - A \left(A^T A \right)^{-1} A^T R = R^T R \right. \end{aligned} \tag{3.101}$$

which shows that the class covariance matrices of $\underline{z}$ do not depend on the class.

The class cross covariance matrix between vectors $\underline{y}$ and $\underline{z}$ is

$$\Sigma_{\underline{yz}i} = R^T \Sigma_{\underline{x}i} A = C^T \left(\Sigma_{\underline{x}i} - A\left(A^T A\right)^{-1} A^T \Sigma_{\underline{x}i}\right) A \tag{3.102}$$

substituting Eq.(3.85) into Eq.(3.102) yields

$$\begin{aligned}\Sigma_{\underline{yz}i} &= C^T \left(\Sigma_{\underline{x}i} - \Sigma_{\underline{x}i} + I - A\left(A^T A\right)^{-1} A^T\right) A = \\ &= C^T \left(A - A\left(A^T A\right)^{-1} A^T A\right) = \underline{0}\end{aligned} \tag{3.103}$$

or for every class the vectors $\underline{y}$ and $\underline{z}$ are uncorrelated which because of the normality assumption are also independent. Hence the class conditional probability density function of the transformed vectors becomes

$$p\left(\tilde{\underline{y}} \mid \omega_i\right) = p\left(\underline{y} \mid \omega_i\right) p\left(\underline{z} \mid \omega_i\right) \tag{3.104}$$

But Eq.(3.99) and (3.101) show that the class mean vectors and the class covariance matrices of vector $\underline{z}$ are not dependent on the classes, therefore

$$p\left(\tilde{\underline{y}} \mid \omega_i\right) = p\left(\underline{y} \mid \omega_i\right) p\left(\underline{z}\right) \tag{3.105}$$

Since the matrix $\tilde{A}$ is full rank, then its determinant will be non zero which means that it is non singular. It have been shown[24] that for a non singular matrix $\tilde{A}$, if

$$\max_j P\left(\omega_j\right) p\left(\underline{x} \mid \omega_j\right) = P\left(\omega_i\right) p\left(\underline{x} \mid \omega_i\right) \tag{3.106}$$

then

$$\max_j P\left(\omega_j\right) p\left(\tilde{\underline{y}} \mid \omega_j\right) = P\left(\omega_i\right) p\left(\tilde{\underline{y}} \mid \omega_i\right) \tag{3.107}$$

where the relationship between vectors $\underline{x}$ and $\tilde{\underline{y}}$ is given in Eq.(3.97).
However from Eq.(3.104),

$$P\left(\omega_i\right) p\left(\tilde{\underline{y}} \mid \omega_i\right) = P\left(\omega_i\right) p\left(\underline{y} \mid \omega_i\right) p\left(\underline{z}\right) \tag{3.108}$$

which means that if Eq.(3.106) is true then

$$\max_j P\left(\omega_j\right) p\left(\underline{y} \mid \omega_j\right) = P\left(\omega_i\right) p\left(\underline{y} \mid \omega_i\right) \tag{3.109}$$

or the class assignment when classifying vectors $\underline{x}$ will be the same as when classifying the transformed vectors $\underline{y}$, which means that the probability of error will not change as well.

3.5.2 Method to Find the Linear Transformation Matrix A

We want to find the $D \times d$ transformation matrix A which satisfies Eq.(3.84) and (3.85). Decell *et al* and Peters *et al* show[22, 23] that we can append the class mean vectors and class covariance matrices as follows,

$$G = \left[\underline{m}_{\underline{x}1}, \ldots, \underline{m}_{\underline{x}K}, \Sigma_{\underline{x}1} - I, \ldots, \Sigma_{\underline{x}K} - I\right] \tag{3.110}$$

which is a $D \times \left(K+1\right) D$ matrix.
If the matrix G is not full rank and its rank is d where $d < D$, then it can be decomposed as

$$G = A\ B \tag{3.111}$$

where A is a $D \times d$ and B is a $d \times (K+1)D$ full rank matrices.
If we do post multiplication of the matrix G as follows

$$A(A^T A)^{-1} A^T G = A(A^T A)^{-1} A^T AB = A\ B = G \tag{3.112}$$

which shows that matrix A defined in Eq.(3.111) will satisfy Eq.(3.84) and (3.85).

However there are still two problems left in finding the matrix A :

1. How to decompose the matrix G following Eq.(3.111)
2. What if the matrix G is full rank.

The singular value decomposition method[18, 25, 26] can be used to solve those problems. Therefore this method will be called the singular value decomposition linear transformation (SVDLT) method, which will be discussed in the following.

3.5.3 Singular Value Decomposition Method

The singular value decomposition (SVD) method is a method to decompose a rectangular matrix as follows,

$$G = U\ Diag(\lambda_i^{1/2}; i = 1, \ldots, d)\ V^T \tag{3.113}$$

where G is a $D \times L$ matrix where its rank is $d, d \leq D < L$. The matrix U is the eigenvector matrix of the matrix GG^T and the matrix V is the eigenvector matrix of the matrix $G^T G$. The λ_i are the eigenvalues of both the GG^T and $G^T G$ matrices. For the matrix G defined in Eq.(3.110) and assuming that it is not full rank i.e. $Rank(G) = d; d < D$, the transformation matrix A can be selected as follows,

$$G = U\ Diag(\lambda_i^{1/2}; i = 1, \ldots, d)\ V^T = \tag{3.114}$$

$$= A\ B$$

where

$$B = Diag(\lambda_i^{1/2}; i = 1, \ldots, d)\ V^T$$

Thus

$$A = U \tag{3.115}$$

which is the eigenvector matrix of GG^T where the eigenvectors are the ones associated with the non zero eigenvalues λ_i. Since the matrix GG^T is symetric then the columns of the matrix U can always be chosen to be orthonormal.

If the matrix G is in fact full rank i.e. there is no eigenvalue λ_i which is zero then we can assume that the matrix G is not full rank if some of the eigenvalues λ_i are *small.* The assumption means that effectivelly the small eigenvalues are zero or the effective rank of the matrix G is equal to the number of the significant eigenvalues. Therefore we can find the $D \times d$ transformation matrix A whose colums are the eigenvectors of the matrix GG^T associated with the significant eigenvalues.

3.5.4 Geometric Intepretation and Modification

For a general situation i.e. distinct class covariance matrices, it is rather unclear what is the geometric intepretation of the above linear transformation with respect to the invariant of the class assignments of the transformed vectors. However for a restrictive case i.e. for equal class covariance matrices case, a geometrical intepretation might be clear out.

For equal covariance matrices case, assuming that the matrices are non singular, we can always find a non singular linear transformation which will yield the covariance matrices to be identity matrices. This process will not change any class assignment.

Next we can shift all the data vectors by the total mean vector defined as,

$$\underline{m}_{\tilde{x}} = \sum_{i=1}^{K} P\left(\omega_i\right) \underline{m}_{\tilde{x}i} \tag{3.116}$$

where the vector $\underline{\tilde{x}}$ is the transformed vector of $\underline{x}$ which yields the class covariance matrices which are identity matrices. This shifting will not change the class assignment as well.

The class mean vectors now becomes,

$$\underline{m}_{\hat{x}i} = \underline{m}_{\tilde{x}i} - \underline{m}_{\tilde{x}} \tag{3.117}$$

where the vector $\underline{\hat{x}}$ is the shifted version of the vector $\underline{\tilde{x}}$.

The matrix G defined in Eq.(3.110) now has columns the shifted class mean vectors only i.e. its columns are the class mean vectors $\underline{m}_{\hat{x}i}$, since the class covariance matrices are identity,

$$G = \left[\underline{m}_{\hat{x}1}, \ldots, \underline{m}_{\hat{x}i}, \ldots, \underline{m}_{\hat{x}K}\right] \tag{3.118}$$

which is a $D \times K$ rectangular matrix.

Because of the definition of the class mean vectors $\underline{m}_{\hat{x}i}$ shown in Eq.(3.116) and (3.117), these K vectors are linearly dependent. Hence the rank of matrix G is,

$$Rank\left(G\right) = \min\left(D\ ,\ K-1\right) \tag{3.119}$$

Situation shown by Eq.(3.119) can have the intepretation that the class mean vectors $\underline{m}_{\hat{x}i}$ are *well distributed,* i.e. if $D > K-1$ then those class mean vectors lie on $K-1$ dimensional space and if $K-1 > D$ then those class mean vectors lie on D dimesional space.

However it is possible that those class mean vectors are not well distributed. They may lie on the space whose dimension is less than either $K-1$ or D. What is the dimension of the space occupied by those mean vectors can be indicated by the rank of the matrix G. And these in turn is the same as the number of the non zero eigenvalues of the singular value decomposition of the matrix G. From Eq.(3.113) or (3.114) it is shown that the columns of the matrix G i.e. the class mean vectors $\underline{m}_{\hat{x}i}$, are linear combination of the orthonormal basis vectors of the columns of the matrix U. Therefore the columns of the matrix U in the decomposition of Eq.(3.113) or (3.114) can be defined as the orthonormal basis vectors which spanned the space occupied by the class mean vectors $\underline{m}_{\hat{x}i}$.

Thus the number of the basis vectors i.e. the number of the columns of the matrix U, depends on how well distributed the class mean vectors $\underline{m}_{\hat{x}i}$. If they are not well distributed then the linear transformation without changing the class assignment of probability of error is possible.

With respect to the geometric intepretation discussed above, modification of the definition of the matrix G given in Eq.(3.110) is in order. What we want is that the matrix G is not full rank. For the restrictive case i.e. the equal class covariance matrices, the matrix G is defined

following Eq.(3.118) where its columns are the shifted class mean vaectors given in Eq.(3.117). For this case if $D > K-1$ then the matrix G is guaranttee not to be full rank i.e. $Rank\left(G\right)=K-1$.

For more general case i.e. distinct class covariance matrices, we will still modifiy the definition of the matrix G by shifting all the original data vectors $\underline{x}$ by the total mean vector $\underline{m}_{\underline{x}}$,

$$\underline{x}_s = \underline{x} - \underline{m}_{\underline{x}} \tag{3.120}$$

which yields the shifted class mean vectors,

$$\underline{m}_{\underline{x}si} = \underline{m}_{\underline{x}i} - \underline{m}_{\underline{x}} \tag{3.121}$$

where the total mean vector $\underline{m}_{\underline{x}}$ is defined as,

$$\underline{m}_{\underline{x}} = \sum_{i=1}^{K} P\left(\omega_i\right) \underline{m}_{\underline{x}i} \tag{3.122}$$

where $\underline{m}_{\underline{x}i}$ are the class mean vectors of the original data vectors $\underline{x}$.
The modified definition of the matrix G will be

$$G = \left[\underline{m}_{\underline{x}s1}, \ldots, \underline{m}_{\underline{x}sK}, \Sigma_{\underline{x}1} - I, \ldots, \Sigma_{\underline{x}K} - I\right] \tag{3.123}$$

With this difinition of the matrix G, for the restrictive case i.e. identity class covariance matrices and $D < K-1$, it is guaranttee that the matrix G is not full rank i.e. $Rank\left(G\right) \leq K-1$. If the class covariance matrices are not identity but they are equal they always can be transformed into identity matrices. More over this definition can be used to compare this SVDLT method with the MDA method which will be done later.

3.5.5 Comparison with the MDA method.

Before we do comparison with the MDA method and in this respect comparisons with any other method, we need to specify the criteria which are going to be used in the comparisons. The ideal criterion certainly is the probability of error, however as mentioned earlier this criterion is very difficult to be used either for design or comparison.

The singular value decomposition linear transformation (SVDLT) method uses the probability of error criterion or more specifically the invariant of the probability of error property with the dimensionality reduction. This property can be maintained iff the two given conditions are satisfied:

1. The classes are multivariate Gaussian distributed.
2. The matrix G defined in Eq.(3.122) is not a full rank matrix.

If $Rank\left(G\right)=d, d<D$, then the invariant of the probability of error property can be maintained while reducing the dimensionality from D to d, using the decomposition of the matrix GG^T, where G is defined in Eq.(3.122). If this is the case therefore the SVDLT method is the optimum one compares to any other method.

Although the multivariate Gaussian class distribution assumption might be satisfied especially for the satellite ground images, but the second condition i.e. the rank of the matrix G may not be satisfied. Even for a restrictive case i.e. equal class covariance matrices, the matrix G may be a full rank matrix. Comparison will be done between the SVDLT and MDA methods only because both methods use the class parameters i.e. the class mean vectors and class covariance matrices in the designs of the linear transformations. However the SVDLT method requires a more stringent condition which is the classes have to be multivariate Gaussian distributed,

whereas the MDA method does not require that condition.

One possible case to have the same resulting linear transformations of the SVDLT and MDA methods is when the class covariance matrices are equal. In this case, it is always guarantteed that the class covariance matrices can be made identity by doing non singular linear transformation toward the data vectors. Therefore following the notations of Eq.(3.62) and (3.63) in the preceeding section, the MDA method now will require the solution of the eigenvalue problem of the among scatter matrix

$$S_{\hat{A}} = \sum_{i=1}^{K} P(\omega_i)\, \underline{m}_{\hat{x}i}\, \underline{m}_{\hat{x}i}^{\,T} = M Diag\left(P(\omega_i); i=1, \ldots, K\right) M^T \tag{3.124}$$

where the matrix M is defined as,

$$M = \left[\underline{m}_{\hat{x}1}, \ldots, \underline{m}_{\hat{x}K}\right] \tag{3.125}$$

where the shifted class mean vectors $\underline{m}_{\hat{x}i}$ are defined in Eq.(3.117). Noted here is that the within scatter matrix $S_{\hat{W}}$ is identity since the class covariance matrices have been transformed into identity matrices.

For this restrictive case the SVDLT method requires the solution of the eigenvalue problem of the GG^T matrix. However as shown in Eq.(3.118) the definition of the matrix M shown in Eq.(3.125) is the same as the definition of the matrix G. Hence Eq.(3.124) can be rewritten as,

$$S_{\hat{A}} = G Diag\left(P(\omega_i); i=1, \ldots, K\right) G^T \tag{3.126}$$

Therefore the SVDLT and MDA methods will have the same resulting linear transformation matrices in the case of equal covariance matrices and equal class apriori probabilities. This conclusion will also applicable whether the matrix G is full rank or not.

3.6. Comparisons and Discussions

1. The method to find the linear transformation using the probability of classification error criterion, appears to be computationally untractable. The method using the probabilistic distance measures does not seem simpler either, although for multivariate Gaussian classes some of the probabilistic distance measures have simpler forms i.e. only functions of class mean vectors and class covariance matrices. But the draw back of this method is that the transformation matrix A found by maximizing the probabilistic distance measures, is only optimum for a particular reduced dimensionality d. For different dimensionality, the matrix A has to be searched again by maximizing the probabilistic distance measure for the given dimensionality d. Therefore the method using the probabilistic distance measures are more appropriate to be used in feature selection than in feature extraction. Where in feature selection, we try to find the best group of measurements among all the available measurements to classify several classes and this information can be used to reduced the number of transducers in later applications. While in feature extraction we try to find the best functional form of the original measurements either for data representation or class separation, where in our study we limit the functional form to be linear.

2. Therefore considering the computational complexity, the method to find the transformation matrix A using the solutions of the eigenvalue problems are optimum. More over these methods can give the indications of the numbers of the transformed features which could be used for the dimensionality reductions. These will be done by ordering the eigenvalues and discarding the features, given by the eigenvectors, with non significance eigenvalues. This characteristics of the methods using the solution of the eigenvalue problem can be contrasted with the

least squared method[27, 28, 29, 30] which is not discussed in our study, where in this method the assignment of the reduced dimensionality d depends on the number of classes only. Whereas in the methods using the solution of the eigenvalue problems, especially the ones which aimed at the class separation such as in the MDA and SVDLT methods, the dimensionality reductions not only depends on the number of classes but also depends among others on how the classes are distributed i.e. more specifically how the class mean vectors are distributed in the original measurement space.

3. The KL transform method is the best method in minimum MSE criterion to represent the data in lesser dimensionality. However it is not designed for class separation, therefore it appears that this method is more appropriate for data exploration step which might include the unsupervised classifications such as clustering. In this sense, for TM data, the TM tasseled cap linear transformation is also more appropriate for data exploration step.

4. For data exploration step if we will use the KL transform method to reduce the dimensionality we need to calculate the covariance matrix of each data set. In other words the KL features are dependent on the data. On the other hand for TM data the TM tasseled cap linear transformation is claimed to be invariant of the data as long as the data are TM data. Therefore also in term of computational complexity, for the TM data, the TM tasseled cap linear transformation is better.

5. In terms of class separation and simplicity of the procedure, the MDA and the proposed SVDLT methods appears to be optimal. The objective functions of both methods are functions of the class parameters i.e. the class mean vectors and class covariance matrices. This can be contransted with KL transform or TM tasseled cap linear transformation methods, which based on data representation either using minimum MSE criterion or using physical properties of the data as the basic considerations.

6. The MDA and KL methods will give a set of uncorrelated features of the mixture of the probabilistics model of the data. However the set of features of the KL transform methods are orthonormal while the set of features of the MDA method are not orthonormal.

7. If the within scatter matrix S_W is identity, which can be caused by the situation where the class covariance matrices are equal, the transformation matrices produced by the MDA and KL transform methods will be the same. It is because the existing relationship between the total covariance matrix Σ_x of the data with the among class scatter matrix S_A and the within class scatter matrix S_W.

8. Also, if the number of classes K is very large but the space can still accomodate the scatter of each class, then the scatter of the data around its mean vectors becomes very small which yields the within scatter matrix S_W approaching the zero matrix. In this case the resulting transformation matrices of the MDA and of the KL transform methods will be the same. This kind of situation is reported[31] by Bricker *et al* where it is said that when the MDA method is applied and the number of class is increased, the resulting features, after some number of class, becomes *stable*. Our conclusion is that after some number of class, where the number of class is much larger than the original dimensionality D, the addition of more classes will not change the covariance matrix Σ_x of the data so much which yields that those *stabilized* features are those of the KL transform, although this conclusion is neither mentioned nor checked in the above report.

9. In term of probability of classification error, if both the conditions are satisfied i.e. the classes are multivariate Gaussian distributed and the matrix G in Sec.2.5. is not full rank, then the SVDLT method is optimum since the resulting features will not change the class assignments for maximum likelihood or maximum aposteriori classifier. The reduced dimensionality of this method cannot be lower than the rank of the matrix G, to maintain the class assignments

in the dimensionality reduced space. If the matrix G is a full rank matrix, then dimensionality reduction will be done by assuming that the rank of G is equal to the number of the significant eigenvalues of the decomposition of the matrix G.

10. For a restrictive case i.e. equal class covariance matrices, the SVDLT method will only use the class mean vectors to find the features. In this case, dimensionality reduction without affecting the class assignments can be achieved if the class mean vectors are not well distributed i.e. the class mean vectors actually occupy a space with lower dimensionality than of the original measurement space. If this is true than the probability of classification error is not affected by dimensionality reduction and this is applicable for number of classes more than two. Also in this case the minimum weighted euclidian distance classifier can substitute the above mentioned two classifiers, where the weights are the class apriori probabilities. This can be contrasted with the MDA method which can use the minimum euclidian distance classifier instead of the above mentioned two classifier for equally likely two classes only, provided both have equal covariance matrices and both clases are Gaussian distributed.

11. For equal covariance matrices and equal class apriori probabilities, the MDA and the SVDLT will yield equal transformations. However both methods start with different assumptions, the SVDLT assumes that the classes are multivariate Gaussian distributed while the MDA only requires that the distribution of the data of each class is unimodal. In this sense the SVDLT method is more restrictive than the MDA method.

12. The MDA and SVDLT methods both use the class mean vectors and class covariance matrices to find the linear transformation for the dimensionality reduction. However this can represent the *lack of generalization capability* because of the limited number of class training samples and both methods require the estimate of the class mean vectors and class covariance matrices. In this case the KL transform might give better linear transformation since the number of data which required to estimate the covariance matrix Σ_x is large i.e. the total number of the data. Similar situation has been shown by Muasher and Landgrebe[32] where classification accuracy using linear transformation produced by KL transform method is better than the classification accuracy from the selection of the features using the probabilistic distance measures i.e. the divergence.

13. If a method to find a linear transformation for dimensionality reduction is optimum to separate all available classes then it is possible that different transformations for different regions of the data space might be better for class separation. This is because a point in the data space might be closed to only several classes among all the available classes and the possibility of making classification error is high only for the classes which are close. This possibility will be pursued further in Sec.6.

References

1. Pierre A. deViejver and Joseph Kittler, *Pattern Recognition: A Statistical Approach,* Prentice-Hall, London, 1982.
2. Tsvi Lissack and King Sun Fu, "Error Estimation in Pattern Recognition via L-Distance Between Posteriory Density Function," *IEEE Transaction on Information Theory*, vol. IT-22, no. 1, January 1976.
3. Marvin Yablon and J. T. Chu, "The Relationship of Bayes Risk to Certain Separability Measures in Normal classification," *IEEE Transaction on Pattern Analysis and Machine Intelegence*, vol. PAMI-2, no. 2, March 1980.
4. K. S. Fu, "Pattern Recognition in Remote Sensing of the Earth Resources," *IEEE Transactions on Geoscience Electronics*, vol. GE-14, no. 1, pp. 10-18, January 1976.

5. Thomas Kailath, "The Divergence and Bhattacharya Distance Measures in Signal Selection," *IEEE Transactions on Communication Technology*, vol. COM-15, no. 1, pp. 52-60, February 1967.

6. Edward A. Patrick and Frederic P. Fischer II, "Nonparametric Feature Selection," *IEEE Transaction on Information Theory*, vol. IT-15, no. 5, Sept. 1969.

7. Keinosuke Fukunaga, *Introduction to Statistical Pattern Recognition,* Academic Press, New York, San Francisco, London, 1972.

8. K. S. Fu, Pyung June Min, and Timothy J. Li, "Feature Selection in Pattern Recognition," *IEEE Transaction on Systems Science and Cybernetics*, vol. SSC-6, no. 1, January 1970.

9. Godfried T. Toussaint, "Some Inequalities Between Distance Measures for Feature Evaluation," *IEEE Transaction on Computers*, vol. C-21, no. 4, April 1972.

10. T. Marill and D. M. Green, "On the Effectiveness of Receptron in Recognition Systems," *IEEE Transaction on Information Theory*, vol. IT-9, Jan. 1963.

11. Richard O. Duda and Peter E. Hart, *Pattern Classification and Scene Analysis,* John Wiley & Sons, New York, London, Sydney, Toronto, 1973.

12. Satosi Watanabe, "Karhunen-Loeve Expasion and Factor Analysis Theoritical Remarks and Applications," *Trans.Fourth Prague Conf.Inform.Theory,Statist.Decision Functions,and Random Processes*, pp. 635-660, Prague, 1965.

13. Eric P. Crist and Richard C. Cicone, "Application of the Tasseled Cap Concept to Simulated Thematic Mapper Data," *Photogrammetric Engineering & Remote Sensing*, vol. L, no. 3, March 1984.

14. Eric P. Crist and Richard C. Cicone, "Comparisons of the Dimensionality and Features of the Simulated Landsat-4 MSS and the TM Data," *Remote Sensing of Environment*, vol. 14, no. 1-3, January 1984.

15. E.P. Crist and R. C. Cicone, "A Physically-Based Transformation of Thematic Mapper Data-The TM Tasseled Cap," *IEEE Transaction on Geoscience and Remote Sensing*, vol. GE-22, no. 3, May 1984.

16. R. J. Kauth and G. S. Thomas, "The Tasseled Cap-A Graphic Description of the Spectral-Temporal Development of Agricultural Crops as seen by Landsat," *Symposium Proceedings on Machine Processing of Remotely Sensed Data, Purdue University W Lafayette, IN.*, June 29 - July 1 , 1976.

17. Paul E. Green and J. Douglass Carroll, *Mathematical Tools for Applied Multivariate Analysis,* Academic Press, New York, San Fransisco, London, 1976.

18. K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis,* Academic Press, London,New York,Toronto,Sydney,San Fransisco, 1979.

19. R. Gnanadesikan, *Methods for Statistical Data Analysis of Multivariate Observations,* John Wiley & Sons, New York,London,Sydney,Toronto, 1977.

20. Donald H. Foley and John W. Sammon, Jr, "An Optimal Set of Discriminant Vectors," *IEEE Transactions on Computers*, vol. C-24, pp. 281-289, March 1975.

21. K. S. Fu, D. A. Landgrebe, and T. L. Phillips, "Information Processing of Remotely Sensed Agricultural Data," *Proceedings of the IEEE*, vol. 57, no. 4, pp. 639-653, April 1969.

22. Henry P. Decell, Jr, P. L. Odell, and William A. Coberly, "Linear Dimension Reduction and Bayes Classification," *Pattern Recognition*, vol. 13, no. 3, pp. 241-243, 1981.

23. B. Charles Peters, Jr, Richard Redner, and Henry P. Decell, Jr, "Characterization of Linear Sufficient Statistics," *Sankhya: The Indian Journal of Statistics*, vol. 40 Series A, pp.

303-309, 1978.

24. Donald F. Morrison, *Multivariate Statistical Methods,* McGraw Hill Book Company, New York St.Louis San Fransisco, 1976.
25. Jan J. Gerbrands, "On the Relationships Between SVD, KLT and PCA," *Pattern Recognition*, vol. 14, no. 1-6, pp. 375-381, Pergamon Press Ltd., 1981.
26. Anil K. Jain, *ECE 206 Text UC Davis*, Davis, 1982.
27. Nobuyuki Otsu, "Optimal Linear and Nonlinear Solutions for Least-Square Discriminant Feature Extraction," *Proceedings IEEE 1982*, 1982.
28. K Fukunaga and S. Ando, "The Optimum Nonlinear Features for a Scatter Criterion in Discriminant Analysis," *IEEE Transaction on Information Theory*, vol. IT-23, July 1977.
29. D. F. Mix and R. A. Jones, "A Dimensionality Reduction Technique Based on a Least Squared Error Criterion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-4, no. 5, pp. 537-544, September 1982.
30. William A. Gardner, "A Unifying View of Second-Order Measures of Quality for Signal Classification," *IEEE Transaction on Communication*, vol. Com-28, no. 6, June 1980.
31. P. D. Bricker, R. Gnanadesikan, M. V. Mathews, Miss S. Pruzansky, P. A. Tukey, K. W. Wachter, and J. L. Warner, "Statistical Techniques for Talker Identification," *The Bell System Technical Journal*, vol. 50, no. 4, April 1971.
32. Marwan Jamil Muasher and David A. Landgrebe, "The K-L Expansion as an Effective Feature Ordering Technique for Limited Training Sample Size," *IEEE Transactions on Geoscience and Remote Sensing*, vol. GE-21, no. 4, October 1983.

4. Dimensionality Reduction for Noisy Observations

In most classification procedures, a data exploration step is required, for example to acquire the training samples for supervised classification. If the data to be processed are suspected to be corrupted by additive noise uncorrelated with the data, methods to reduce the dimensionality will have to consider the characteristics of the noise. However, in the data exploration step, the class definitions usually have been finalized. Therefore the methods which will be discussed will consider the signal as a single class noisy data set although some discussion of the MDA method with noisy data will be given as well.

The discussion will begin with the well known method Factor Analysis (FA), and an alternative for finding the factors using a minimum MSE criterion will be proposed. Next we discuss concisely the method proposed by Hsu and Womble[1], which is a method to reduce the dimensionality of the noisy data based on minimum MSE where the resulting features are assumed orthogonal. This discussion is followed by a proposed alternative method, where the dimensionality reduction is based on signal to noise ratio.

4.1. Factor Analysis

Suppose one suspects that the unknown signal, usually called the common factor[2, 3], lies in a space with lower dimensionality than that of the observed data or the observations. The full dimensionality of the observation therefore is due to an additive unknown noise, usually called the unique factor. Dimensionality reduction in such a case is to try to *recover* the signal which is believed to have lesser dimensionality than that of the observation. Therefore dimensionality reduction is done indirectly, through the recovery of the signal.

It is known that uncorrelated additive noise, which is also uncorrelated or orthogonal to the signal, will lead to a decrease in the autocorrelation coefficients between elements of the observation. This can be shown from the increase in the diagonal elements of the covariance matrix of the signal which is also the covariance matrix of the observation. With an increase in the diagonal elements of the covariance matrix while the off diagonal elements remain constant, the autocorrelation coefficients of the observation will decrease. This will result in a decrease of the redundancy of the information in the observation.

As was discussed in the KL transform section in Sec.3, dimensionality reduction can be achieved effectively if there is high redundancy in the data or observation or, equivalently, the autocorrelation coefficients of the observations are high. If we assume that low redundancy or low autocorrelation coefficients of the elements of the observation are caused by the additive noise, then an effort to discard the noise or to remove the effects of the noise to recover the unknown signal will yield an effective dimensionality reduction. In this sense, in the following we will discuss the method of dimensionality reduction using the very well known method of Factor Analysis, abbreviated by FA. In this method, there is a problem of non uniqueness of the factor and factor loading determinations. Several possibilities to overcome this problem have been proposed[2, 3], and a method to determine the factor and factor loading based on minimum mean squared error criterion is proposed.

4.1.1. Linear Model in Factor Analysis

The model in the FA method is given as

$$\underline{y} = A\underline{f} + \underline{q} \tag{4.1}$$

where

$\underline{y} = D \times 1$ observable vector

$A = D \times d$ unknown matrix called factor loading

$\underline{f} = d \times 1$ unobservable (hypothetical) vector called common factor or signal

The random vector $\underline{q}$ is uncorrelated and zero mean and the random vectors $\underline{f}$ and $\underline{q}$ are also uncorrelated.

The covariance matrix of the observation $\underline{y}$ is

$$\Sigma_{\underline{y}} = A \Sigma_{\underline{f}} A^T + \Delta \tag{4.2}$$

where $\Sigma_{\underline{f}}$ is the covariance matrix of the common factor $\underline{f}$ and $\Delta = diag\left(\delta_1^2, \ldots, \delta_D^2\right)$ is the covariance matrix of the unique factor $\underline{q}$.

If $\Sigma_{\underline{f}} = I_d$ then Eq.(4.2) becomes

$$\Sigma_{\underline{y}} = AA^T + \Delta \tag{4.3}$$

And also if the covariance matrix of $\underline{f}$ is decomposed as

$$\Sigma_{\underline{f}} = TT^T$$

where T is a lower triangular matrix[2], the covariance matrix of $\underline{y}$ given in Eq.(4.2) will become

$$\Sigma_{\underline{y}} = \tilde{A}\tilde{A}^T + \Delta$$

which has the same form as the covariance matrix of $\underline{y}$ given in Eq.(4.3) where $\tilde{A} = AT$.

On the other hand, if the loading matrix A is post multiplied by a $d \times d$ orthonormal matrix B, we will have

$$(AB)(AB)^T = AA^T$$

therefore the factor loading A is non unique under orthonormal rotation.

All these non uniqueness problems associated with the FA method might be a set back, but fortunately these non uniquenesses only occur when we want to define the factor loading A and the common factor $\underline{f}$. For any non unique factor loading A, the multiplication

$$AA^T = \Sigma_{\underline{y}} - \Delta$$

is unique. This uniqueness of the covariance matrix of $A\underline{f}$ will be exploited later to find the unique signal or common factor based on minimum mean squared error criterion. Now what has to be estimated are either of the possible matrices A or $\tilde{A}$, or the matrix Δ from the covariance matrix of observation $\Sigma_{\underline{y}}$.

There are two methods to estimate the factor loading matrix A, the principal factor method and maximum likelihood method. In the following we first will discuss the principal factor method. Let the observation vector $\underline{y}$ be scaled by a matrix C as follows,

$$\underline{t} = C^{-1/2}\underline{y} \tag{4.4}$$

where $C = diag\left(\sigma_{11}, \ldots, \sigma_{DD}\right)$ and σ_{ii} is the i^{th} diagonal element of $\Sigma_{\underline{y}}$.

The covariance matrix of $\underline{t}$ is

$$\Sigma_{\underline{t}} = R_{\underline{y}}$$

where R_y is the coefficient correlation matrix of y.

We can express the coefficient correlation matrix R_y in a matrix equation similar to that of Eq.(4.3),

$$R_y = PP^T + \Psi \tag{4.5}$$

From Eq.(4.5) we then have

$$PP^T = R_y - \Psi$$

The matrix $R_y - \Psi$ is usually called the reduced correlation matrix, from which the factor loading matrix will be estimated. We cannot proceed further before estimating the diagonal elements of the reduced correlation matrix PP^T, which are usually called the communalities, h_i^2,

$$h_i^2 = 1 - \xi_i$$

where ξ_i is an element of the diagonal matrix Ψ. The h_i^2 terms are bounded as follows: $0 < h_i^2 < 1$.

Some of the proposed estimates of the communality h_i^2 are[2, 3] :

(i) The magnitude of the highest correlation coefficient of the variable y_i with one of the remaining variables, or

$$\hat{h}_i^2 = \max_j | r_{ij} |$$

for $i \neq j$.

(ii) The square of the multiple correlation of y_i with the remaining variables,

$$\hat{h}_i^2 = 1 - \frac{1}{s_{ii}}$$

where $s_{ii} = \left\{ R_y^{-1} \right\}_{ii}$.

Once the diagonal elements of the reduced correlation matrix, $PP^T = R_y - \Psi$, have been estimated then decomposition is done as follows

$$PP^T = R_y - \Psi = \sum_{i=1}^{D} h_i \alpha_i \alpha_i^T \tag{4.6}$$

where h_i and α_i are the eigenvalues and eigenvectors of $R_y - \Psi$.

Then we order the eigenvalues as follows,

$$h_1 \geq, \ldots, \geq h_i \geq, \ldots, \geq h_D \tag{4.7}$$

It is expected that some of the h_i will be very small and possibly negative. To maintain the semi positive definiteness of the matrix PP^T, we will select only the positive and largest h_i's such that the sum of those which are selected is almost equal but less than D and then select the eigenvectors α_i accordingly.

The estimate of P will be:

$$\hat{P} = \Gamma_d H^{1/2} \tag{4.8}$$

where

$$\Gamma_d = (\alpha_1, \ldots, \alpha_d)$$

$$H^{1/2}=Diag\left\{h_i^{1/2}\right\}$$

To proceed we have to check that the elements of the estimate of the matrix Ψ, $\hat{\Psi}$, are positive or equal to zero, i.e.

$$\left\{\hat{\Psi}\right\}_i=\left\{R_y-\hat{P}\hat{P}^T\right\}_i\geq 0 \tag{4.9}$$

If Eq.(4.9) is satisfied then the estimate of the factor loading A matrix is

$$\hat{A}=C^{1/2}\hat{P} \tag{4.10}$$

where the matrix C is defined in Eq.(4.4).

The matrix $\hat{A}$ defined in Eq.(4.10) satisfies

$$\hat{A}^T C^{-1}\hat{A}=H \tag{4.11}$$

where H is a diagonal matrix, see Eq.(4.8), which is a suggested constraint[3], to get a unique factor loading A .

The other method to find the factor loading A is the maximum likelihood method. This method uses the assumption that the observation vector $\underline{y}$ is multivariate normal.

Using the Wishart density[2, 3], the log likelihood function of matrices A and Δ is

$$L\left(A,\Delta\mid S\right)=-\frac{n-1}{2}\left[\ln\mid AA^T+\Delta\mid+tr\left\{\left(AA^T+\Delta\right)^{-1}S\right\}\right] \tag{4.12}$$

where S is the sample covariance matrix. For $D\leq n-1$ where D is the dimensionality of S and n is the total number of samples, S will be non singular with probability one and will have the Wishart distribution. The indeterminacy of A up to a rotation is handled in this estimation by a constraint

$$\hat{J}=\hat{A}^T\hat{\Delta}\hat{A} \tag{4.13}$$

where $\hat{J}$ is a diagonal matrix. This will lead to an iterative method

$$\hat{J}\hat{A}^T=\hat{A}\,\hat{\Delta}^{-1}\left(S-\hat{\Delta}\right)$$

$$\hat{\Delta}=Diag\left(S-\hat{A}\hat{A}^T\right)$$

Sometimes this iteration does not converge[2].

Once the estimate of the factor loading A is found, the common factor $\underline{f}$ is given by[4]:

$$\hat{\underline{f}}=\hat{A}^T\Sigma_{\underline{y}}^{-1}\underline{y} \tag{4.14}$$

Eq.(4.14) represents the linear regression of $\underline{y}$ to find $\underline{f}$.

There are two other posibilities[3], the first called the *Bartlett's factor score,*

$$\hat{f}=\left(\hat{A}^T\hat{\Psi}^{-1}\hat{A}\right)^{-1}\hat{A}^T\hat{\Psi}^{-1}\underline{y} \tag{4.15}$$

and the second called *Thompson's factor score,*

$$\hat{f}=\left(I+\hat{A}^T\hat{\Psi}^{-1}\hat{A}\right)^{-1}\hat{A}^T\hat{\Psi}^{-1}\underline{y} \tag{4.16}$$

Which one of these three common factor estimates is the best is not clear, and even the conditions under which one is better than the rest is not clear. This confusion leads to an idea of finding the factor or the signal under the minimum mean squared error criterion.

4.1.2. Minimum Mean Squared Error Criterion Based Factor Analysis

We will define the vector $A\underline{f}$ in Eq.(4.1) as,

$$\underline{x} = A\underline{f}$$

The equation Eq.(4.1) now becomes

$$\underline{y} = \underline{x} + \underline{q} \tag{4.17}$$

Without loss of any generality, we can assume that the mean vector of $\underline{y}$ and also of $\underline{x}$ are zero vectors. If they are not, we can shift $\underline{y}$ by its mean vector, since $\underline{q}$ has been assumed to be a zero mean random vector. Using all the uncorrelatedness assumptions of the FA model the covariance matrix of $\underline{y}$ is

$$\Sigma_{\underline{y}} = \hat{\Sigma}_{\underline{x}} + \hat{\Delta}$$

and by observing Eq.(4.3), $\hat{\Sigma}_{\underline{x}} = \hat{A}\hat{A}^T$ is unique, where $\hat{A}$ is given in Eq.(4.10). From Wiener filter theory[5], the minimum mean squared error estimate of $\underline{x}$ is given by

$$\underline{\hat{x}} = \hat{\Sigma}_{\underline{x}} \Sigma_{\underline{y}}^{-1} \underline{y} \tag{4.18}$$

where $\underline{\hat{x}}$ is a $D \times 1$ vector. We can apply the Wiener filter method in this scheme because all the conditions or assumptions to applied to this method are satisfied by the FA model given in Eq.(4.1) and subsequently in Eq.(4.17). The covariance matrix of the estimate $\underline{\hat{x}}$ is

$$\Sigma_{\underline{\hat{x}}} = E\left(\underline{\hat{x}}\underline{\hat{x}}^T\right) = \hat{\Sigma}_{\underline{x}} \Sigma_{\underline{y}}^{-1} \hat{\Sigma}_{\underline{x}} \tag{4.19}$$

The covariance matrix $\Sigma_{\underline{\hat{x}}}$ has rank $d < D$, because its columns result from linear combination of the d linearly independent columns of matrix $\hat{A}$.

Following the approach of the KL transform, dimensionality reduction can be achieved by finding the eigenvectors of $\Sigma_{\underline{\hat{x}}}$ matrix having non zero eigenvalues. There will be d such eigenvectors. Let us define a $D \times d$ matrix B

$$B = \left(\underline{b}_1, \ldots, \underline{b}_d\right)$$

where $\underline{b}_i$, for $i = 1, \ldots, d$ are the eigenvectors of $\Sigma_{\underline{\hat{x}}}$ associated with non zero eigenvalues. Apply the KL expansion of $\underline{\hat{x}}$ as follows

$$\underline{\tilde{x}} = \sum_{i=1}^{i=d} z_i \underline{b}_i \tag{4.20}$$

where $z_i = \underline{\hat{x}}^T \underline{b}_i$. The mean squared error of these two steps is

$$\begin{aligned} e &= trE\left(\underline{x} - \underline{\tilde{x}}\right)\left(\underline{x} - \underline{\tilde{x}}\right)^T = trE\left(\underline{x} - \underline{\hat{x}} + \underline{\hat{x}} - \underline{\tilde{x}}\right)\left(\underline{x} - \underline{\hat{x}} + \underline{\hat{x}} - \underline{\tilde{x}}\right)^T \\ &= trE\left(\underline{x} - \underline{\hat{x}}\right)\left(\underline{x} - \underline{\hat{x}}\right)^T + trE\left(\underline{\hat{x}} - \underline{\tilde{x}}\right)\left(\underline{\hat{x}} - \underline{\tilde{x}}\right)^T \\ &= + trE\left(\underline{x} - \underline{\hat{x}}\right)\left(\underline{\hat{x}} - \underline{\tilde{x}}\right)^T + trE\left(\underline{\hat{x}} - \underline{\tilde{x}}\right)\left(\underline{x} - \underline{\hat{x}}\right)^T \end{aligned} \tag{4.21}$$

We can observe from Eq.(4.21) that the first term is the mean squared error of the Wiener filtering, and the second term is the mean squared error of the KL expasion of $\hat{\underline{x}}$. Now we will consider the third term,

$$trE\left(\underline{x}-\hat{\underline{x}}\right)\left(\hat{\underline{x}}-\tilde{\underline{x}}\right)^{T} = trE\left(\underline{x}-\hat{\underline{x}}\right)\hat{\underline{x}}^{T} - trE\left(\underline{x}-\hat{\underline{x}}\right)\tilde{\underline{x}}^{T}$$

The first term of the right most side is zero because of the orthogonality principle[5, 6], and by substituting $\tilde{\underline{x}}$ by Eq.(4.20), the second term of the right most side becomes

$$trE\left(\underline{x}-\hat{\underline{x}}\right)\tilde{\underline{x}}^{T} = \sum_{i=1}^{i=d} trE\left(\underline{x}-\hat{\underline{x}}\right)\hat{\underline{x}}^{T}\underline{b}_i\,\underline{b}_i^{T} = 0$$

Similarly the fourth term of Eq.(4.21) will be zero also.

Thus by observing the non zero terms of Eq.(3.21), the mean squared error is

$$e = e_W + e_{KL} \tag{4.22}$$

where e_W is the error from the Wiener filtering and e_{KL} is the error of the KL expansion. Both errors have been minimized, and therefore the sum of these errors is minimized as well.

The dimensionality reduction or more precisely the estimation of the common factor now can be stated as follows

$$\hat{\underline{f}} = B^{T}\hat{\underline{x}} = B^{T}\hat{\Sigma}_{\underline{x}}\Sigma_{\underline{y}}^{-1}\underline{y} \tag{4.23}$$

The proposed method gives a unique estimate of the common factor $\underline{f}$ based on the minimum mean squared error criterion, in which the dimensionality reduction is achieved since the dimensionality of the common factor is less than the dimensionality of the observation. This is done in two steps, where it has been shown that the total error of the two steps is the sum of error in the individual steps. Since each step is based on the minimum mean squared error criterion, the total mean squared error is also minimized.

4.2. Noisy Observation

In the preceeding section, we discussed the Factor Analysis method in which the additive noise components are assumed uncorrelated. Therefore the noise covariance matrix is diagonal but unknown. In this section, the noisy observation is also assumed to consist of the signal vector, $\underline{x}$, and the additive noise vector, $\underline{n}$, as follows

$$\underline{y} = \underline{x} + \underline{n}$$

The noise and signal are also assumed uncorrelated but the noise components are not. Therefore the noise covariance matrix is not diagonal. Moreover, either the covariance matrix of the noise of the signal is assumed known, so that the unknown one, because of the uncorrelatedness of the signal and the noise, can be estimated by subtracting the estimate of the covariance matrix of the observation by the known covariance matrix. With this scheme, in the following we will discuss two methods of dimensionality reduction, one of which is done using a set of orthonormal basis vectors. The one that requires the set of orthonormal basis vectors is found from minimizing the mean squared error of the signal representation. The other one is found from maximizing the signal to noise ratio on each new variable.

4.2.1. Minimum Mean Squared Error Based Dimensionality Reduction

If we would like to expand the signal vector $\underline{x}$ using a set of orthonormal basis vectors similar to the expansion shown in Eq.(3.24), the resulting mean squared error is produced by the

sum of the variances of the signal on the discarded basis vectors plus the sum of the variances of the noise on the selected basis vectors. Thus the mean squared error of the expansion is,

$$e = \sum_{i=d+1}^{i=D} \underline{\phi}_i{}^T \Sigma_{\underline{x}} \underline{\phi}_i + \sum_{i=1}^{i=d} \underline{\phi}_i{}^T \Sigma_{\underline{n}} \underline{\phi}_i = Trace\, \Sigma_{\underline{x}} - \sum_{i=1}^{i=d} \underline{\phi}_i{}^T \left(\Sigma_{\underline{x}} - \Sigma_{\underline{n}} \right) \underline{\phi}_i \tag{4.24}$$

where $\underline{\phi}_i$ are the orthonormal basis vectors, $\Sigma_{\underline{x}}$ is the covariance matrix of the signal and $\Sigma_{\underline{n}}$ is the covariance matrix of the noise.

To minimize e in Eq.(4.24), the second term in the last equality has to be maximized. Following the discussion in the KL transform section in Sec.3, the set of orthonormal basis vectors, $\underline{\phi}_i$, have to be the eigenvectors associated with the d largest eigenvalues of the matrix $\Sigma_{\underline{x}} - \Sigma_{\underline{n}}$. This matrix is not necessarily positive definite, and therefore some of its eigenvalues might be negative. Hence to minimize e in Eq.(4.24), the selected eigenvectors have to have positive eigenvalues. This is exactly what has been shown by Hsu and Womble[1].

For classification or, more precisely, dealing with more than one class of signals, the above authors[1] suggested that the average of the covariance matrices of the signal from each class and the average of the covariance matrices of the noise from each class are to replace the covariance matrices $\Sigma_{\underline{x}}$ and $\Sigma_{\underline{n}}$ respectively in Eq.(3.24). The eigenvectors and eigenvalues are then to be found accordingly. Our comment on this idea is that the resulting basis vectors might not be the best for classification because the suggested procedure neglects the class mean vectors which are often the most important class parameters in classification.

4.2.2. Signal to Noise Ratio Based Dimensionality Reduction

As an alternative to the preceeding method, in the following a method is proposed whose basic idea is to find a set of vectors such that if both the signal and the noise are projected on these vectors, the ratio of the variances of the signal and the noise is maximized.

Suppose $\underline{a}_i$ is the intended vector. The projection of the signal $\underline{x}$ on $\underline{a}_i$ is given by

$$z_i = \frac{\underline{a}_i{}^T \underline{x}}{\| \underline{a}_i \|^{1/2}} \tag{4.25}$$

The variance of z_i is

$$S_i = E\left(z_i - m_{z_i} \right)^2 = \frac{1}{\| \underline{a}_i \|} \underline{a}_i{}^T E\left(\underline{x} - \underline{m}_{\underline{x}} \right) \left(\underline{x} - \underline{m}_{\underline{x}} \right)^T \underline{a}_i = \frac{1}{\| \underline{a}_i \|} \underline{a}_i{}^T \Sigma_{\underline{x}} \underline{a}_i \tag{4.26}$$

where m_{z_i} is the mean of z_i and $\underline{m}_{\underline{x}}$ is the mean vector of $\underline{x}$.

Similarly the projection of the noise on $\underline{a}_i$ is,

$$p_i = \frac{\underline{a}_i{}^T \underline{n}}{\| \underline{a}_i \|^{1/2}} \tag{4.27}$$

The variance of p_i is

$$N_i = E\left(p_i - m_{p_i} \right)^2 = \frac{1}{\| \underline{a}_i \|} \underline{a}_i{}^T \Sigma_{\underline{n}} \underline{a}_i \tag{4.28}$$

The signal to noise ratio on the vector a_i is

$$\frac{S_i}{N_i} = \frac{\underline{a}_i{}^T \Sigma_{\underline{x}} \underline{a}_i}{\underline{a}_i{}^T \Sigma_{\underline{n}} \underline{a}_i} \tag{4.29}$$

The right hand side of Eq.(4.29) looks similar to what is called *Rayleigh quotient*[7]. The solution of this maximization requires that $\underline{a}_i$ be the eigenvector of the matrix $\Sigma_{\underline{n}}^{-1}\Sigma_{\underline{x}}$ associated with the largest eigenvalue and the ratio $\frac{S_i}{N_i}$ is this eigenvalue. One property of the matrix $\Sigma_{\underline{n}}^{-1}\Sigma_{\underline{x}}$ is that if the matrix $\Sigma_{\underline{x}}$ is (semi) positive definite then the eigenvalues of $\Sigma_{\underline{n}}^{-1}\Sigma_{\underline{x}}$ are always larger than (or equal) to zero[3]. Since both $\Sigma_{\underline{x}}$ and $\Sigma_{\underline{n}}$ are covariance matrices, at least $\Sigma_{\underline{x}}$ is semi positive definite, and $\Sigma_{\underline{n}}$ has to be positive definite to be nonsingular.

If we do the ordering as follows,

$$\frac{S_1}{N_1} \geq, \ldots, \geq \frac{S_i}{N_i} \geq, \ldots, \geq \frac{S_D}{N_D} \tag{4.30}$$

We can select $\underline{a}_i$'s associated with d largest $\frac{S_i}{N_i}$ or d non zero $\frac{S_i}{N_i}$, and the transformation will be

$$\underline{z} = A^T \underline{y} \tag{4.31}$$

where $\underline{z}$ is a $d \times 1$ vector, and

$$A = \left(\underline{a}_1, \ldots, \underline{a}_d\right)$$

and $\underline{y}$ is the $D \times 1$ noisy observation vector.

Another alternative is, instead of directly maximizing $\frac{S_i}{N_i}$ in Eq.(4.29), a prewhitening process is initially applied to the noise and is then followed by the maximization of the signal to noise ratio. This procedure amounts to simultaneous diagonalization of the matrices $\Sigma_{\underline{x}}$ and $\Sigma_{\underline{n}}$ [8, 9] which will be shown next.

For the prewhitening process we would like to find a $D \times D$ matrix Q such that if we apply the following transformation

$$Q^T \underline{y} = Q^T \underline{x} + Q^T \underline{n} \tag{4.32}$$

the covariance matrix of the random vector $Q^T \underline{n}$ is an identity matrix.

Define a $D \times D$ matrix P as follows

$$P = \left(\underline{p}_1, \ldots, \underline{p}_i, \ldots, \underline{p}_D\right) \tag{4.33}$$

where the eigenvectors of $\Sigma_{\underline{n}}$ are given by $\underline{p}_i$, and they are known to be orthonormal.

From the eigen equation of matrix $\Sigma_{\underline{n}}$ we have

$$P^T \Sigma_{\underline{n}} P = Diag\left(\alpha_i\right) \tag{4.34}$$

where α_i are the eigenvalues of $\Sigma_{\underline{n}}$, which are arranged according to their associated eigenvector in the matrix P given in Eq.(4.33).

Pre and post multiply the left hand side of Eq.(4.34) by a matrix $Diag\left(\alpha_i^{-1/2}\right)$ to yield

$$Diag\left(\alpha_i^{-1/2}\right) P^T \Sigma_{\underline{n}} P Diag\left(\alpha_i^{-1/2}\right) = I_D \tag{4.35}$$

The covariance matrix of the random vector $Q^T \underline{n}$ is

$$\Sigma_{Q\underline{n}} = Q^T \Sigma_{\underline{n}} Q \tag{4.36}$$

Observing Eq.(4.35) and (4.36), we can conclude that the matrix Q that will make the covariance matrix of the random vector $Q^T \underline{n}$ an identity matrix is

$$Q = PDiag\left(\alpha_i^{-1/2}\right) \tag{4.37}$$

Also the covariance matrix of the random vector $Q^T \underline{x}$ is

$$\Sigma_{Q\underline{x}} = Q^T \Sigma_{\underline{x}} Q \tag{4.38}$$

where the matrix Q is defined in Eq.(4.37)

The signal to noise ratio to be maximized can be written as

$$\frac{\tilde{S}_i}{\tilde{N}_i} = \frac{\tilde{\underline{a}}_i^T \Sigma_{Q\underline{x}} \tilde{\underline{a}}_i}{\tilde{\underline{a}}_i^T \Sigma_{Q\underline{n}} \tilde{\underline{a}}_i} = \frac{\tilde{\underline{a}}_i^T \Sigma_{Q\underline{x}} \tilde{\underline{a}}_i}{\tilde{\underline{a}}_i^T \tilde{\underline{a}}_i} \tag{4.39}$$

This maximization will show that $\tilde{\underline{a}}_j$ are the eigenvectors of the matrix $\Sigma_{Q\underline{x}} = Q^T \Sigma_{\underline{x}} Q$, and that the associated eigenvalues are $\frac{\tilde{S}_i}{\tilde{N}_i}$. Also, it can be observed that the variances of the noise on all of the $\tilde{\underline{a}}_i$ vectors are equal to 1, and all components of the noise vector $Q^T \underline{n}$ are uncorrelated, hence $Q^T \underline{n}$ is white noise. Therefore the signal to noise ratio in Eq.(4.39) is equal to the variance of the signal only or

$$\frac{\tilde{S}_i}{\tilde{N}_i} = \tilde{S}_i \tag{4.40}$$

If we order the $\tilde{S}_i$ as follows

$$\tilde{S}_1 \geq, \ldots, \geq \tilde{S}_i \geq, \ldots, \geq \tilde{S}_D \tag{4.41}$$

and select d eigenvectors $\tilde{\underline{a}}_i$ associated with the d largest $\tilde{S}_i$, dimensionality reduction can be achieved.

Define a $D \times d$ matrix $\tilde{A}$,

$$\tilde{A} = \left(\tilde{\underline{a}}_1, \ldots, \tilde{\underline{a}}_d\right) \tag{4.42}$$

The transformation will be

$$\tilde{\underline{z}} = \tilde{A}^T Q^T \underline{y} = \tilde{A}^T Diag\left(\alpha_i^{-1/2}\right) P^T \underline{y} = G^T \underline{y} \tag{4.43}$$

where $\tilde{\underline{z}}$ is a $d \times 1$ vector and $G = PDiag\left(\alpha_i^{-1/2}\right) \tilde{A}$.

We would like to observe the resulting covariance matrix of $\underline{y}$ after the transformation, i.e. the covariance matrix of the random vector $G^T \underline{y}$. Before we proceed we would like to define a $D \times D$ matrix $\tilde{A}_D$ as follows

$$\tilde{A}_D = \left(\tilde{\underline{a}}_1, \ldots, \tilde{\underline{a}}_i, \ldots, \tilde{\underline{a}}_D\right)$$

and also define a $D \times D$ matrix G_D,

$$G_D = PDiag\left(\alpha_i^{-1/2}\right) \tilde{A}_D \tag{4.44}$$

The transformation of $\underline{y}$ similar to the one given in Eq.(4.43) is

$$\tilde{z_D} = G_D{}^T \underline{y} = G_D{}^T \underline{x} + G_D{}^T \underline{n} \tag{4.45}$$

The covariance matrix of the random vector $G_D{}^T \underline{x}$ is

$$C_{\underline{x}} = G_D{}^T \Sigma_{\underline{x}} G_D = \tilde{A_D}{}^T \Sigma_{Q\underline{x}} \tilde{A_D} \tag{4.46}$$

where the last equality in Eq.(4.46) is found from using the definition of matrix Q given in Eq.(4.37) and of the definition of matrix $\Sigma_{Q\underline{x}}$ given in Eq.(4.38). However the columns of matrix $\tilde{A_D}$, as shown in Eq.(4.44), are the eigenvectors of the matrix $\Sigma_{Q\underline{x}}$, and the matrix $C_{\underline{x}}$ in Eq.(4.46) becomes

$$C_{\underline{x}} = \tilde{A_D}{}^T \Sigma_{Q\underline{x}} \tilde{A_D} = Diag\left(\tilde{S_i}\right) \tag{4.47}$$

The covariance matrix of random vector $G_D{}^T \underline{n}$ is

$$C_{\underline{n}} = G_D{}^T \Sigma_{\underline{n}} G_D = \tilde{A_D}{}^T Diag\left(\alpha_i{}^{-1/2}\right) P^T \Sigma_{\underline{n}} P Diag\left(\alpha_i{}^{-1/2}\right) \tilde{A_D} \tag{4.48}$$

But the columns of the matrix P as defined in Eq.(4.33) are the eigenvectors of the matrix $\Sigma_{\underline{n}}$, and hence

$$C_{\underline{n}} = \tilde{A_D}{}^T Diag\left(\alpha_i{}^{-1/2}\right) Diag\left(\alpha_i\right) Diag\left(\alpha_i{}^{-1/2}\right) \tilde{A_D} = \tilde{A_D}{}^T \tilde{A_D} = I_D \tag{4.49}$$

Therefore from Eq.(4.47) and (4.49), we can conclude that the matrix G_D diagonalizes the matrices $\Sigma_{\underline{x}}$ and $\Sigma_{\underline{n}}$ simultaneously, and more over since $\underline{x}$ and $\underline{n}$ are uncorrelated, the covariance matrix of the random vector $G_D{}^T \underline{y}$ is

$$C_{\underline{y}} = C_{\underline{x}} + C_{\underline{n}} = Diag\left(\tilde{S_i} + 1\right) \tag{4.50}$$

or the covariance matrix of the random vector $G_D{}^T \underline{y}$ is diagonal which means that the components of the random vector $G_D{}^T \underline{y}$ are uncorrelated.

One other thing worth noting is that the diagonalization of the covariance matrix $\Sigma_{\underline{y}}$ is done with matrix G_D which is not an orthonormal matrix by the definition given in Eq.(4.44). This should be contrasted with the diagonalization of the covariance matrix of the KL transformation, whose transformation matrix is an orthonormal matrix.

4.3. Relationship to Classification

From the above sections, it is clear that the FA method and the subsequent noisy observation dimensionality reduction methods are intended for processing single class random vectors. Regarding classification, there is a report[10] that shows a possible relationship between the FA method and the MDA method (discussed in Sec.3). This report says that the FA method can substitute the MDA method. This is a very useful idea since the MDA method requires that the class conditional mean vectors and covariance matrices to be known and the FA method only requires knowledge of the covariance matrices of the mixture data. However the condition that allows this substitution is rather difficult to satisfy[10], as it requires the uniqueness of the factor loading A as given in Eq.(4.3). However, the MDA method indirectly can also reduce the effect of the noise since from the relationship,

$$\Sigma_{\underline{y}} = \Sigma_{\underline{x}} + \Sigma_{\underline{n}} = S_A + S_W + \Sigma_{\underline{n}}$$

where S_A and S_W are the among class scatter and within class scatter matrices respectively, defined in the MDA method. We can lump the matrices S_W and $\Sigma_{\underline{n}}$,

$$\tilde{S}_W = S_W + \Sigma_n$$

and substitute this into the preceeding equation yielding,

$$\Sigma_y = S_A + \tilde{S}_W$$

and apply the MDA method. By doing this, what we have done is to lump the effect of the noise which tends to enlarge the within class scattering of the data into the within class scattering of the non noisy data.

Besides all the above *draw backs* with respect to classification, the application of the FA method or more generally the application of the noisy observation dimensionality reduction methods discussed above toward satellite data has not been thoroughly studied. The capability to estimate the noise covariance matrix of the FA method and the capabilities to reduce the effect of the additive noise components of those noisy observation dimensionality reduction methods might be attractive for TM data since due to the higher spatial resolution of the data acquisition system of the TM data, the class conditional variances are expected to be large which causes the data to appear to be more noisy. The application of these methods might yield a better classification error.

References

1. Kai Hsu and M. Edward Womble, "Separation and Feature Selection of Non-Stationary Processes via KL Spectra Analyses," *IEEE 1981 Man, Machine & Cybernetics*, 1981.
2. R. Gnanadesikan, *Methods for Statistical Data Analysis of Multivariate Observations,* John Wiley & Sons, New York,London,Sydney,Toronto, 1977.
3. K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis,* Academic Press, London,New York,Toronto,Sydney,San Fransisco, 1979.
4. Harry H. Harman, *Modern Factor Analysis,* University of Chicago Press, Chicago, 1960.
5. N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing,* Springer-Verlag, New York, Heidelberg, Berlin, 1975.
6. Athanosios Papoulis, *Probability, Random Variables and Stochastic Processes,* McGraw-Hill Book Company, New York,St.Louis,San Fransisco,Toronto,London,Sidney, 1965.
7. Richard O. Duda and Peter E. Hart, *Pattern Classification and Scene Analysis,* John Wiley & Sons, New York, London, Sydney, Toronto, 1973.
8. Benjamin F. Merembeck and Brian J. Turner, "Directed Canonical Analysis and the Performance of Classifiers Under Its Associated Linear Transformation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. GE-18, no. 2, pp. 190-196, April 1980.
9. Paul E. Green and J. Douglass Carroll, *Mathematical Tools for Applied Multivariate Analysis,* Academic Press, New York, San Fransisco, London, 1976.
10. Giacomo Della Riccia and Alexander Saphiro, "Fisher Discriminant Analysis and Factor Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelegence*, vol. PAMI-5, no. 1, pp. 99-104, January 1983.

5. A Unified Approach to Dimensionality Reduction by Linear Transformation

In Sec.3, several methods for finding the $D \times d$ linear transformation matrix A were discussed. The first two methods, i.e. the one based on probability of error, P_e, and the one based on the probabilistic distance, have objective functions which are functions of the matrix A, although unfortunately, the solutions appear computationally untractable. The other methods do not have objective functions which are explicit functions of the matrix A. Instead their objective functions are functions of a set of vectors. We would like to modify these objective functions so that they are functions of the $D \times d$ matrix A. By doing this we will have a unified approach to dimensionality reduction by linear transformation, as each method will involve optimizing an objective function with respect to the matrix A.

5.1 Minimum Probability of Error and Maximum Probabilistic Distances Based Objective Functions

We will first rewrite the objective functions which are already functions of the matrix A. The one that is based on probability of error, from Eq.(3.12), is

$$J(A) = P_e(A) = \int \left\{ 1 - \max_i P(\omega_i \mid A^T \underline{x}) \right\} p(A^T \underline{x})\, d(A^T \underline{x}) \tag{5.1}$$

The one that is based on the probabilistic distance method, from Eq.(3.14), is

$$J(A) = \sum_{i=1}^{K} \sum_{j=1}^{K} P(\omega_i) P(\omega_j) J(i,j,A) \tag{5.2}$$

where $J(i,j,A)$ is the probabilistic distance between classes ω_i and ω_j from Eq.(3.13) for a particular transformation matrix A, where the relationship between the transformed vector y and the original vector x is given in Eq.(3.2).

The objective function $J(A)$ of Eq.(5.1) has to be minimized and that of Eq.(5.2) has to be maximized. The optimizations are done with respect to the matrix A.

5.2 KL Transform Based Objective Function

For the KL transform method we will start with Eq.(3.25)

$$\tilde{e} = E\left\{ (\underline{x} - \tilde{\underline{x}})^T (\underline{x} - \tilde{\underline{x}}) \right\} \tag{5.3}$$

Substituting $\tilde{x}$ from Eq.(3.24) and by the orthonormality of the $\underline{a}_i$ vectors, we will have

$$\tilde{e} = E\left\{ \sum_{i=d+1}^{D} (y_i - b_i)^2 \right\} \tag{5.4}$$

To optimize with respect to b_i, we take the first derivative of $\tilde{e}$ w.r.t. b_i and set it to zero to yield

$$b_i = E\{y_i\} = \underline{a}_i^T E\{\underline{x}\} = \underline{a}_i^T \underline{m} \tag{5.5}$$

where $\underline{m} = E\{\underline{x}\}$ is the mean vector of $\underline{x}$.

Substituting b_i from Eq.(5.5) into Eq.(5.4) and performing the expectation yields

$$\tilde{e} = \sum_{i=d+1}^{D} \underline{a}_i{}^T \Sigma_{\underline{x}} \underline{a}_i$$

and this is the same as

$$\tilde{e} = \sum_{i=1}^{D} \underline{a}_i{}^T \Sigma_{\underline{x}} \underline{a}_i - \sum_{i=1}^{d} \underline{a}_i{}^T \Sigma_{\underline{x}} \underline{a}_i$$

Putting the summation into the Trace form yields

$$\tilde{e} = Trace\, \Sigma_{\underline{x}} - Trace\left(A^T \Sigma_{\underline{x}} A\right) \tag{5.6}$$

where $\Sigma_{\underline{x}}$ is the covariance matrix of $\underline{x}$ defined as

$$\Sigma_{\underline{x}} = E\left\{ \left(\underline{x}-\underline{m}\right) \left(\underline{x}-\underline{m}\right)^T \right\} \tag{5.7}$$

and the columns of matrix A are the orthonormal vectors $\underline{a}_i$ for $i = 1, \ldots, d$.

To minimize the representation error $\tilde{e}$, the second term on the right hand side of Eq.(5.6) has to be maximized. Moreover the vectors $\underline{a}_i$ are orthonormal, and therefore the objective function as a function of the $D \times d$ matrix A will be

$$J(A) = Trace\left(A^T \Sigma_{\underline{x}} A\right) \tag{5.8}$$

with the constraint,

$$A^T A = I_d \tag{5.9}$$

5.3 MDA Based Objective Function

For the MDA method we will start with the objective function of Eq.(3.61)

$$\alpha_i = \frac{\underline{a}_i{}^T S_A \underline{a}_i}{\underline{a}_i{}^T S_W \underline{a}_i} \tag{5.10}$$

Substituting $\underline{a}_i$ from Eq.(3.74) yields

$$\alpha_i = \frac{\underline{\hat{a}}_i{}^T \left(S_W{}^{-1/2}\right)^T S_A \left(S_W{}^{-1/2}\right) \underline{\hat{a}}_i}{\underline{\hat{a}}_i{}^T \underline{\hat{a}}_i} \tag{5.11}$$

where $\underline{\hat{a}}_i$ is defined in Eq.(3.73) and $S_W{}^{-1/2}$ is defined from the factorization of S_W given in Eq.(3.70). The right hand side of Eq.(5.11) is independent of the norm of the vector $\underline{\hat{a}}_i$ since this norm can be factored from both the numerator and denominator and cancelled out. Therefore we can constrain the norm of $\underline{\hat{a}}_i$ i.e. rewriting Eq.(3.75)

$$\| \underline{\hat{a}}_i \| = \underline{\hat{a}}_i{}^T \underline{\hat{a}}_i = 1 \tag{5.12}$$

which means that

$$\underline{a}_i{}^T S_W \underline{a}_i = 1 \tag{5.13}$$

And now we have the objective function

$$\alpha_i = \underline{a}_i^{\,T} S_A \underline{a}_i \tag{5.14}$$

with constraint,

$$\underline{a}_i^{\,T} S_W \underline{a}_i = 1$$

Therefore we can write the objective function as a function of the $D \times d$ matrix A, using Eq.(5.14) and Eq.(5.13),

$$J(A) = Trace\left(A^T S_A A\right) \tag{5.15}$$

with the constraint that:

$$\left\{A^T S_W A\right\}_{ii} = 1 \tag{5.16}$$

where the $D \times d$ matrix A has as its columns the vectors $\underline{a}_i ; i = 1, \ldots, d$, and the diagonal elements of the symmetric matrix $A^T S_W A$ are one. The objective function of Eq.(5.15) has to be maximized.

It is tempting to the constraint that, $A^T S_W A = I_d$, but this constraint will be too restrictive and is not necessary. For, the maximization of the objective function of Eq.(5.15) with the constraint of Eq.(5.16) will yield a matrix A such that $A^T S_W A = I_d$ as shown in Eq.(3.82).

Unfortunately, the matrix A found from the eigen equation Eq.(3.65) can not be put into an objective function which is a function of the matrix A. If we follow the steps used to get to Eq.(5.15), we will fail because of the lack of symmetry of the matrix $S_W^{-1} S_A$.

However, we can generalize the constraint of Eq.(5.16) so that

$$\left\{A^T S_W A\right\}_{ii} = g_i > 0 \tag{5.17}$$

where g_i is a constant. This means that the columns of the $D \times d$ transformation matrix A, $\underline{a}_i$, will have their directions and lengths or norm, $\| \underline{a}_i \|$, satisfy the constraint of Eq.(5.17). On the other hand the norm of $\underline{a}_i$ will not affect the ratio of Eq.(5.10). Therefore, we can conclude that the constraint of Eq.(5.17) will not restrict the search for the vector $\underline{a}_i$ to optimize Eq.(5.10) for all possible vectors $\underline{a}_i$. In fact[1], the constraint of Eq.(5.17)

$$\left\{A^T S_W A\right\}_{ii} = \underline{a}_i^{\,T} S_W \underline{a}_i = g_i \tag{5.18}$$

where $\underline{a}_i$ is the i^{th} column of the $D \times d$ matrix A, is another definition of norm.

Since the constraint has been made more general, the objective function must be modified. The modification has to follow the ratio given in Eq.(5.10) which yields

$$J(A) = Trace\left(A^T S_A A \; Diag\left(1/g_i ; i = 1, \ldots, d\right)\right) \tag{5.19}$$

where the constraint, rewriting Eq.(5.17), is

$$\left\{A^T S_W A\right\}_{ii} = g_i \tag{5.20}$$

The objective function $J(A)$ of Eq.(5.19) which has to be maximized, constrained by Eq.(5.20), will have the objective function $J(A)$ of Eq.(5.15), constrained by Eq.(5.16), as a

special case. Moreover the objective function $J(A)$ of Eq.(5.8) constrained by Eq.(5.9), i.e. the objective function of the KL transform method, can also be viewed as a special case by redefining the matrices S_A of Eq.(5.19) as Σ_z of Eq.(5.8) and S_W and g_i of Eq.(5.20) as the identity matrix, I_d, and 1 of Eq.(5.9) respectively.

5.4 Singular Value Decomposition Linear Transformation Based Objective Function

In this method, the matrix A, which is searched, is the factor of the singular value decomposition defined in Eq.(3.114) of the matrix G defined in Eq.(3.123). From Eq.(3.115) and the discussion in Sec.3.5.3, the columns of the matrix A are the eigenvectors of the matrix GG^T associated with nonzero eigenvalues or with significant eigenvalues.

If we write the eigen equation of the matrix GG^T as follows,

$$GG^T \underline{a}_i = \lambda_i \underline{a}_i \tag{5.21}$$

The matrix GG^T is a symetric matrix, so we can always make the eigenvectors $\underline{a}_i$ orthonormal. Multiplying both sides of Eq.(5.21) by $\underline{a}_i^T$ will yield,

$$\underline{a}_i^T GG^T \underline{a}_i = \lambda_i \tag{5.22}$$

where from the eigen equation in Eq.(5.21), the norm of the eigenvector $\underline{a}_i$ is

$$\| \underline{a}_i \| = \underline{a}_i^T \underline{a}_i = 1 \tag{5.23}$$

Therefore, from Eq.(5.21), (5.22) and (5.23), the unified objective function will be,

$$J(A) = Trace\left(A^T GG^T A\right) \tag{5.24}$$

with constraint

$$\left(A^T A\right)_{ii} = 1 \tag{5.25}$$

The above objective function has to be maximized. Moreover, we do not constrain the columns of the $D \times d$ matrix A to be orthonormal because the method does not require this, as discussed in Sec.3.5. However, the orthonormality of the columns of the matrix A is the result of the symmetry of the matrix GG^T.

If the matrix G is not full rank and we only want to reduce the dimensionality down to the rank of the matrix G, then for the rank of the matrix G is d, the matrix A will only have d number of columns. We will not add the number of the columns of the transformation matrix A with eigenvectors with zero eigenvalues.

The general similarity of the objective functions of the methods given from Sec.5.2 to 5.4 occurs because all of the methods, as discussed in Sec.3, are based on eigen equations of symmetric matrices. The maximizations of the trace amount to selecting the d number eigenvectors associated with the d largest eigenvalues.

References

1. Parlett B. N., *The Symmetric Eigenvalue Problem,* Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632, 1980.

6. Space Variant Linear Transformation

In the preceeding sections several linear transformations have been discussed. Some of them use class conditional information with an objective that the resulting linear transformation have high capability of separating classes with a smaller dimensionality. However all of the discussed methods use one linear transformation for the entire raw data or measurement space. The use of one linear transformation for the entire raw data or measurement space will be referred to as the space invariant linear transformation.

In the following we will propose a method which will use different linear transformations for different regions of the data space. The motivations for proposing this idea are :

1. It has been shown[1, 2] that the classification error is most affected by points close to the decision boundaries.

2. For more than two classes there is more than one boundary, so it is possible that a point is close to the boundaries of some of the classes but far from the boundaries of other classes. In this case the linear transformation at that point can be designed to separate only the classes whose boundaries are close to the point.

3. Following the idea given in 2., the design of the linear transformation at each point will involve only a subset of classes. This amounts to discarding irrelevant classes and it is expected to yield a more accurate linear transformation for the separation of the involved classes. The involved classes are therefore the classes which are close to the point to be transformed.

To implement this idea several problems are encountered, including:

1. How do we know that a point is close to the boundaries and which classes are close to that point?. This problem leads to the definition of a preliminary space (abbreviated by PS) which will be discussed below. The processes applied on the PS to solve these problems will be discussed in Sec.6.2., 6.3. and 6.4.

2. For the involved classes i.e. the classes considered close to a point, what is the appropriate linear transformation?. The choice is basically the one produced through the MDA method which has been discussed earlier. This problem will be discussed concisely in Sec.6.5.

In Sec.6.6. we will discuss the final classification step and two possible sources of classification error will be discussed in Sec.6.7. In Sec.6.8, a comparison with the space invariant linear transformation method will be discussed.

While the details will be given in the following sections, the general approach in the space variant linear transformation method consists of two steps:

1. Performing a space invariant linear transformation. In this step, a small number of features which provide the best separation will be retained. This small number of features is the space which will be called the Preliminary Space. The linear transformation in this step is that produced by the KL expansion method.

2. Adding additional features when needed by a space variant linear transformation. The linear transformation will be applied to the features which are not retained in the preceeding step. The features which are not retained will be called the Complementary Space (abbreviated by CS).

For completeness, we will define the Total Space (abbreviated by TS) as the space spanned by all the features produced by the KL expansion in the above first step.

6.1 Processes on PS

In general there will be two classes of processes on PS:

1. The preprocesses whose objectives are :

a. To choose the threshold parameter t, as described in Sec.6.2. The threshold t is used to indicate whether an unclassified point is close to decision boundaries.

b. To choose groups of classes in which the classes, which are members of a group, are considered close, as described in Sec.6.4.

2. The processes applied to the unclassified data:

a. Performing the classification.

b. If the classification indicates that the point is close to the decision boundaries then selecting one of the groups produced in process 1.b. above for that point.

The calculations done in the PS have to be computationaly simple, and therefore its dimensionality has to be low but with good class separation. The features of the KL expansion method will be used as TS and the first few features with the largest eigenvalues will be chosen as PS. The KL expansion method is chosen because the number of classes at this step is still large.

6.2 Error/Reject Relationship on PS

Classification as mentioned in the preceeding section, Sec.6.1., will be done as follows. For point $\underline{x}_{\hat{d}}$ in PS, where $\hat{d}$ is the dimensionality of PS, the classification rule is the one that minimizes the probability of error i.e. the maximum aposteriori probability, $P(\omega_i \mid \underline{x}_{\hat{d}})$, classification rule and will be applied as follows,

Assign $\underline{x}_{\hat{d}}$ to class ω_i iff :

$$P(\omega_i \mid \underline{x}_{\hat{d}}) \geq P(\omega_j \mid \underline{x}_{\hat{d}}) \quad i \neq j \tag{6.1}$$

and

$$P(\omega_i \mid \underline{x}_{\hat{d}}) \geq 1-t \quad ; 0 \leq t < \frac{K}{K-1} \tag{6.2}$$

The probability of error using this classification rule is[2, 3],

$$P_e \leq t \tag{6.3}$$

If the inequality of Eq.(6.2) is not satisfied, it means that the point is close to decision boundaries and requires further processing. Proper choice of t will prevent excessive classification error[3, 4, 5], as shown by Eq.(6.3). However, this is accomplished by substituting a rejection for a decision. The value of t will affect the rejection rate i.e. the smaller the t the higher the rejection rate which means more points need further processing. The relationship between probability of error, $P_e(t)$, and rejection rate, $R(t)$, has been given by Chow[3] :

$$P_e(t) = -\int_{\tau} \tau \, dR(\tau) \quad ; 0 < \tau \leq t \tag{6.4}$$

which is an upward concave function. Since for a large number of classes the closed form of $R(t)$ is difficult to find, we can approximate $R(t)$ by a piecewise linear approximation, as shown in Fig.6.1., where the value of R at $t = t_i$ is found by performing the classification using the rules given by Eq.(6.1) and (6.2). In this classification, we can use *unlabelled* points, and count the number of points rejected for several values of t_i for the associated rejection rate $R(t_i)$.

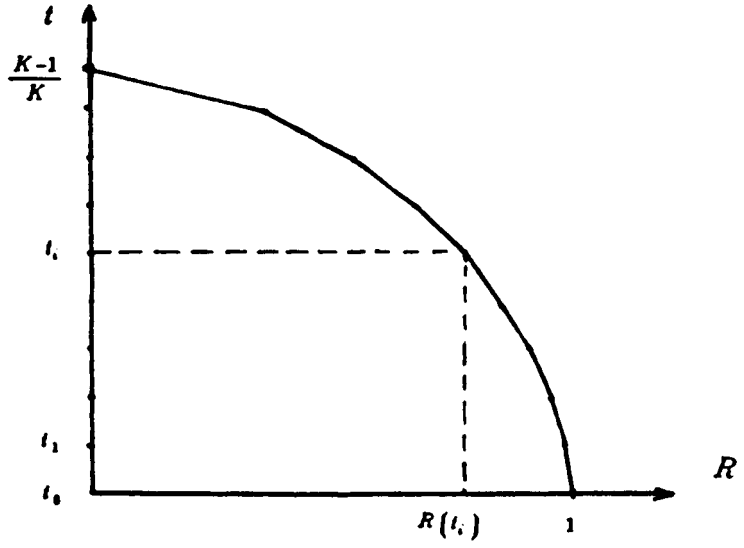

Fig.6.1. Piecewise Linear Approximation of $R(t)$.

If we substitute the above piecewise linear approximation into Eq.(6.4) we will have

$$P_e(t_i) = \sum_{j=1}^{i} \int_{R \in \gamma_j} t(R)\, dR \tag{6.5}$$

where γ_j is the interval $R(t_j) < R \leq R(t_{j-1})$. From Fig.6.1., we note that $t_0=0$ and $R(t_0)=1$. The functions $t(R)$ within the interval γ_j are linear. Thus from Eq.(6.5), we can find the approximate probability of error on PS without rejection, P_{ePS}, by choosing $t_i = \frac{K-1}{K}$.

If P_{ePS} is too large then we can increase the dimensionality of PS, $\hat{d}$. In the following section we will discuss how to assign the value of t.

6.3 Classification Threshold on PS

We will assign $\underline{x}_j$ to a class if the largest aposteriori probability satisfies the inequality of Eq.(6.2). However we want to choose the threshold $1-t$ such that the probability of error based on classification on PS, is approximately equal to the probability of error based on classification on TS. Therefore we need to know the approximate probability of error on TS, P_{eTS}. For this we will define a monotonic approximation relationship between P_e and the dimensionality as follows. We first will order the KL features by the eigenvalues β_i :

$$\beta_1 \geq \cdots \geq \beta_i \geq \cdots \geq \beta_D \tag{6.6}$$

The monotonic approximation relationship will given as follows

$$P_{ePS} = \frac{\beta}{\sum_{i=1}^{\hat{d}} \beta_i} P_{eTS} \tag{6.7}$$

where P_{eTS} = approximation of probability of error on TS and

$$\beta = \sum_{i=1}^{D} \beta_i \tag{6.8}$$

The approximation given by Eq.(6.7) says that the larger the eigenvalues β_i, the smaller the probability of error if the associated feature is used in the classification. This is the approach generally used in a feature selection process.

Given P_{ePS}, using Eq.(6.7) we can determine P_{eTS}. Now the choice of t is such that

$$\hat{P}_{ePS} = P_{eTS} = P_e\ (t) \tag{6.9}$$

where $\hat{P}_{ePS}$ = the probability of error on PS with the rejection threshold $1-t$, with t found using Eq.(6.5). Eq.(6.9) says that t is chosen such that the probability of error on PS, will be approximately equal to the probability of error on TS.

6.4 Selection of Classes Considered Close

As mentioned in Sec.6.1. point 2. for a point $\underline{x}_{\hat{d}}$ on PS which is close to a decision boundary i.e. which does not satisfy the inequality of Eq.(6.2), we need to know the classes which are close to that point so that we can provide a linear transformation for these classes. The ideal measure is the sum of the largest aposteriori probabilities which satisfies the threshold, $1-t$,

$$\sum_{i \in k} P\left(\omega_i \mid \underline{x}_{\hat{d}}\right) \geq 1-t \tag{6.10}$$

where k represents the group of classes with the largest aposteriori probabilities. With this measure we know that the point $\underline{x}_{\hat{d}}$ is a member of the classes $\omega_i \in k^{th}$ group with probability of error not larger than t. However this approach is difficult to apply because :

1. We need the information for group k before classification so we can determine the linear transformation.

2. The number of possible groupings of classes will be too large.

A possible alternative is to use the probabilistic distance measure, i.e. the Lissack-Fu distance measure[1, 2]. For classes ω_i and ω_j, it is given by

$$J_s\left(\omega_i, \omega_j\right) = \int \left| P\left(\omega_i\right) p\left(\underline{x} \mid \omega_i\right) - P\left(\omega_j\right) p\left(\underline{x} \mid \omega_j\right) \right|^s p^{1-s}\left(\underline{x}\right) d\underline{x} \tag{6.11}$$

where for the parameter $s = 1$, it becomes

$$J_1\left(\omega_i, \omega_j\right) = E\left| P\left(\omega_i \mid \underline{x}\right) - P\left(\omega_j \mid \underline{x}\right) \right| \tag{6.12}$$

by assumming that $p\left(\underline{x}\right)$ never goes to zero for all $\underline{x}$. The distance $J_1\left(\omega_i, \omega_j\right)$ of Eq.(6.12) has a direct relationship with the probability of error for two classes case[1], i.e.

$$P_{ij} = \frac{1}{2}\left[1 - J_1\left(\omega_i, \omega_j\right)\right] \tag{6.13}$$

Because of this property, we will choose the distance measure of Eq.(5.12) as the distance between two classes, ω_i and ω_j. For multivariate normal classes, this can be estimated by[1] :

$$\hat{J}_1\left(\omega_i, \omega_j\right) = \frac{1}{N_{ij}} \sum_{k=1}^{N_{ij}} \tanh \left| \frac{1}{2} d_{ij}\left(\underline{x}_k\right) \right| \tag{6.14}$$

where

$$d_{ij}\left(\underline{x}\right) = \frac{1}{2}\left(\underline{x} - \underline{m}_j\right)^T \Sigma_j^{-1}\left(\underline{x} - \underline{m}_j\right) - \frac{1}{2}\left(\underline{x} - \underline{m}_i\right)^T \Sigma_i^{-1}\left(\underline{x} - \underline{m}_i\right) - \ln\left(h_{ij}\right)$$

$$h_{ij} = \frac{P\left(\omega_j\right) \left|\Sigma_i\right|^{\frac{1}{2}}}{P\left(\omega_i\right) \left|\Sigma_j\right|^{\frac{1}{2}}}$$

and $N_{ij}=N_i+N_j$ is the sum of training samples from classes ω_i and ω_j.

We have the distances of each pair of classes so for a particular class we know the classes which are close. We can limit the number of classes that we consider to be close for any particular class. The selection of the close classes to a point $\underline{x}$ then depends on the classification on PS. The group of close classes to a point $\underline{x}$ is defined to consist of the class with the largest aposteriori probability on PS and the classes which are close to this class. With this scheme we will have at most K groups of classes and they can be identified before the classification.

6.5 Linear Transformation for the Close Classes

Basically we will use the MDA method with some modifications to determine the transformation for close classes. The linear transformation will be designed to operate on the elements of the point which lie on the complementary space, CS. The modifications are needed since now we deal with a group of classes but not with all available classes and we proceed as follows

$$\hat{P}_k\left(\omega_i\right)=\psi_k P\left(\omega_i\right) \tag{6.15}$$

where ψ_k is a constant for the k^{th} group such that

$$\sum_{i\epsilon k}\hat{P}_k\left(\omega_i\right)=1 \tag{6.16}$$

where $\hat{P}_k\left(\omega_i\right)$ is modified apriori probability of class ω_i given that $\omega_i \epsilon\, k^{th}$ group.

The group mean vector of the k^{th} group on CS

$$\underline{m}_{kCS}=\sum_{i\epsilon k}\hat{P}_k\left(\omega_i\right)\underline{m}_{iCS} \tag{6.17}$$

The among group scatter matrix becomes

$$\hat{S}_{Ak}=\sum_{i\epsilon k}\hat{P}_k\left(\omega_i\right)\left(\underline{m}_{iCS}-\underline{m}_{kCS}\right)\left(\underline{m}_{iCS}-\underline{m}_{kCS}\right)^T \tag{6.18}$$

The within group scatter matrix becomes

$$\hat{S}_{Wk}=\sum_{i\epsilon k}\hat{P}_k\left(\omega_i\right)C_{iCS} \tag{6.19}$$

where $\underline{m}_{iCS}$ is the subvector of the mean vector of class ω_i, $\underline{m}_i$, and C_{iCS} is the submatrix of the covariance matrix of class ω_i, C_i, associated with CS.

We will have at most K linear transformations and the choice of the transformation for a point depends on which class has the largest aposteriori probability at the classification on PS.

The transformed point will be

$$\underline{y}_k=\begin{bmatrix}\underline{x}_{\hat{d}}\\ \underline{z}_k\end{bmatrix}=\begin{bmatrix}\underline{x}_{\hat{d}}\\ A_k{}^T\underline{x}_{D-\hat{d}}\end{bmatrix} \tag{6.20}$$

where $\underline{y}_k$ is an $\left(\hat{d}+r_k\right)\times 1$ vector, r_k is the dimensionality of the transformed point on CS for the k^{th} group, $\underline{x}_{D-\hat{d}}$ is the element of point $\underline{x}$ on CS and A_k is the $\left(D-\hat{d}\right)\times r_k$ linear transformation matrix resulting from the applying the MDA method to the k^{th} group.

6.6 Final Classification

For each $\underline{y}_k$ given in Eq.(6.20) the classification rule to be applied is also the maximum aposteriori probability for classes $\omega_i \epsilon\, k^{th}$ group. At this point not all of the available classes are involved so some modifications have to be made. The mixture density of $\underline{x}_{\hat{d}}$ for the k^{th} group is

$$p\left(\underline{x}_{\hat{d}k}\right)=\sum_{i \epsilon k}\hat{P}_k\left(\omega_i\right)p\left(\underline{x}_{\hat{d}} \mid \omega_i\right) \tag{6.21}$$

where the modified apriori probability $\hat{P}_k\left(\omega_i\right)$ is as given in Eq.(6.15).

The aposteriori probability of class $\omega_i \epsilon\, k^{th}$ group is

$$P\left(\omega_i \mid \underline{y}_k\right)=\hat{P}_k\left(\omega_i\right)\frac{p\left(\underline{y}_k \mid \omega_i\right)}{p\left(\underline{y}_k\right)}=\tilde{P}_k\left(\omega_i\right)\frac{p\left(\underline{z}_k \mid \underline{x}_{\hat{d}},\omega_i\right)}{p\left(\underline{z}_k \mid \underline{x}_{\hat{d}}\right)} \tag{6.22}$$

where

$$\tilde{P}_k\left(\omega_i\right)=\frac{\hat{P}_k\left(\omega_i\right)p\left(\underline{x}_{\hat{d}} \mid \omega_i\right)}{p\left(\underline{x}_{\hat{d}k}\right)} \tag{6.23}$$

where $p\left(\underline{x}_{\hat{d}k}\right)$ is given in Eq.(6.26) and

$$p\left(\underline{z}_k \mid \underline{x}_{\hat{d}}\right)=\sum_{i \epsilon k}\hat{P}_k\left(\omega_i\right)p\left(\underline{z}_k \mid \underline{x}_{\hat{d}},\omega_i\right) \tag{6.24}$$

For a multivariate normal vector the density function $p\left(\underline{z}_k \mid \underline{x}_{\hat{d}},\omega_i\right)$ can be found with the method given by Srivastava and Carter[6], and to calculate $P\left(\omega_i \mid \underline{y}_k\right)$ we can use $p\left(\underline{x}_{\hat{d}} \mid \omega_i\right)$ which has been calculated on PS.

6.7 Sources of Error in the Final Classification

There are two possible sources of classification error in the final classification :

1. The point is close to the boundaries of the decision rule.
2. The point may not belong to the group of classes it has been assigned to.

The first source of error can be detected if the largest aposteriori probability is small. However, this indicates that there is a high probability that the point belongs to classes of the group where the point is assigned to initially. For the second source of error, the largest aposteriori probability is possibly high. However, that point maybe too *far* from the class mean vector of the class that has the largest aposteriori probability. This *farness* can be detected by assigning a measure of class occupation. For example, if a point is assigned to the k^{th} group and the largest aposteriori probability is of class $\omega_i \epsilon k^{th}$ group, then that point is considered too *far* from the class mean vector of the class $\omega_i \epsilon k^{th}$ or considered as not belonging to the k^{th} group if the following inequality is not satisfied[7],

$$\left(\underline{y}_k-\underline{m}_{yki}\right)^T\Sigma_{yki}{}^{-1}\left(\underline{y}_k-\underline{m}_{yki}\right)\leq\xi_{ki} \tag{6.25}$$

For $\underline{y}_k$ a multivariate normal vector, the left hand side of Eq.(6.25) is distributed $\chi^2_{\hat{d}+r_k}$ and the value of ξ_{ki} can be found by defining that $\underline{y}_k$ will be rejected if the probability of the left

hand side of Eq.(6.25) is larger than a certain value i.e.

$$P\left\{\chi^2_{\hat{d}+r_k} > \xi_{ki}\right\} = \gamma\% \tag{6.26}$$

where ξ_{ki} can be found from a χ^2 table given the value γ.

Between the two possible sources of error, from the overall strategy, the second is more important since it says that the point y_k might not come from the k^{th} group, therefore further processing needs to be done. This can be done by applying a *clean up* process on the image space, i.e. doing majority rule classification on an image block for that particular point.

6.8 Comparisons with the Space Invariant Linear Transformation Method

The disadvantage of the proposed procedure is certainly in the complexity of the computation as compared to the space invariant linear transformation method. However it can be balanced by several less complex calculations which are :

1. For a probability of error approximately equal to that on the total space,TS, some of the data are classified on a simpler space i.e. on the preliminary space, PS, whose dimensionality is smaller than the TS and also smaller than the one which is designed using the space invariant linear transformation method.

2. The space variant linear transformations which are to be used on the complementary space, CS, for each group of classes, may have less dimensionality as compared to the space invariant linear transformation method, i.e. $\hat{d}+r_k \leq d$ where $\hat{d}$ is the dimensionality of PS, r_k is the dimensionality of the transformed CS for the k^{th} group and d is the dimensionality produced by the space invariant linear transformation method. This is because the space variant linear transformations are designed for a smaller number of classes, for which less dimensionality is needed for a comparable probability of error. In this sense it is appropriate to report that from our experiments using the KL expansion method for the seven dimensional TM data, we can reduce the dimensionality to four, using the rule of thumb proposed by Merembeck and Turner[8].

3. In the final classification the number of *distances,* (aposteriori probabilities or the likelihood functions), which must be calculated are less than if we use the space invariant linear transformation method. This is because at the final classification step, we need only to calculate the *distances* of the classes that are members of a group, instead of all available classes.

Besides, from the class pair distances calculated on PS to find the close classes, we have a more detailed view of the data and can decide on the next processes based to this information.

All of these advantages have to be paid by doing complex calculations or processing on PS. However all of these processes are basically done *off line,* i.e. they are not done for each unclassified point. Also another disadvantage is that the *clean up* process is required for our proposed method in the final classification step.

References

1. Tsvi Lissack and King Sun Fu, "Error Estimation in Pattern Recognition via L-Distance Between Posteriory Density Function," *IEEE Transaction on Information Theory*, vol. IT-22, no. 1, January 1976.

2. Pierre A. deViejver and Joseph Kittler, *Pattern Recognition: A Statistical Approach,* Prentice-Hall, London, 1982.

3. C.K. Chow, "On Optimum Recognition Error and Reject Tradeoff," *IEEE Transactions on Information Theory*, vol. IT-16, no. 1, pp. 41-46, January 1970.

4. Lawrence R. Rabiner, Stephen E. Levinson, Aaron E. Rosenberg, and Jay G. Wilpon, "Speaker-Independent Recognition of Isolated Words Using Clustering Techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-27, no. 4, pp. 336-349, August 1979.

5. J. M. Holtzman, "Automatic Speech Recognition Error/No Decision Tradeoff Curves," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-32, no. 6, pp. 1232-1235, December 1984.

6. M. S. Srivastava and E. M. Carter, *An Introduction to Applied Multivariate Statistics*, North-Holland, New York. Amsterdam. Oxford, 1983.

7. K. S. Fu, "Pattern Recognition in Remote Sensing of the Earth Resources," *IEEE Transactions on Geoscience Electronics*, vol. GE-14, no. 1, pp. 10-18, January 1976.

8. Benjamin F. Merembeck and Brian J. Turner, "Directed Canonical Analysis and the Performance of Classifiers Under Its Associated Linear Transformation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. GE-18, no. 2, pp. 190-196, April 1980.

7. Experimental Evaluation of Dimensionality Reduction Techniques

Having developed a theoretical analysis of the four methods of dimensionality reduction by linear transformation, as presented in Sec.3., we then performed experiments to compare the performance of these methods. The comparisons are based on the probability of classification errors for the same reduced dimensions for each method. The need to make experimental comparison is due to the difficulties of establishing the analytical relationship between probability of error and dimensionality reduction by linear transformation for general situations. For a very restrictive situation, for example, for two multivariate normal classes with equal covariance matrices, the MDA method has a direct relationship with the probability of error. As shown in Sec.3. this problem leads to finding the linear transformation for dimensionality reduction by using objective functions which do not have a direct relationship with probability of error.

The best method in the comparisons should have the smallest number of the reduced dimensions with a probability of classification error that still is close to that of raw data, having the original dimensionality. This implies that the probability of classification error for the raw data is always the best i.e. the smallest. However, depending of the method used to estimate the probability of error, this is not always the case, as shown later in the experimental results. Discussions of methods for estimating the probability of error will be presented in this chapter as well.

Also, in this chapter, we will propose and discuss modifications of the methods presented previously. The ideas for these modifications came out of the experimental work. They are referred to as the First and Second Version of the Weighted MDA method, modifying the ordinary MDA method, and the KL Transform-MDA Hybrid method, modifying the KL Transform. Experiments with these methods will also be performed.

7.1. Evaluation Method: Probability of Error Classification Estimation

Basically there are two general methods of probability of error estimation[1]: non parametric and parametric methods. In the non parametric method the estimate of the probability of error does not depend on the probability distribution of the classes. The classification is performed on data for which the true classes are known apriori. Then the classification error is estimated by counting the misclassified samples. This estimation is also not dependent on the classification rule. What is estimated here therefore, is the probability or the rate of misclassification which, in practice, may be more useful than estimating the theoretical optimum Bayesian probability of error. However the non parametric method always requires training samples i.e. a set of known class data for classification.

In the parametric method what is usually estimated is the optimal or Bayesian probability of error which is given theoretically by

$$P_e = \int \left(1-\max_i P\left(\omega_i \mid \underline{x}\right)\right) p\left(\underline{x}\right) d\underline{x} \tag{7.1}$$

where $P\left(\omega_i \mid \underline{x}\right)$ is the aposteriori probability of class ω_i and $p\left(\underline{x}\right)$ is the mixture density. Since the aposteriori probability depends on the class apriori probability and the class conditional probability density (see Sec.3.) then the optimal probability of error is dependent on these parameters.

An example of the parametric method to estimate the optimal probability of error is[2]:

$$\hat{P}_e = \frac{1}{N}\sum_{k=1}^{N}\left(1-\max_i P\left(\omega_i \mid \underline{x}_k\right)\right) \quad ;i=1, \ldots, C \tag{7.2}$$

where N is the total number of data samples and C is the total number of classes. It is shown in the cited reference that this is an unbiased estimator and for two classes its variance is always less or equal to that of the estimator found by the non parametric method.

Another example is that known as error/rejection trade off[3] i.e. the functional relation between the probability of error and the rejection rate for any level of classification threshold t, which can be used to describe the performance of a classification system. This method, discussed in Sec.6.3., included a piecewise linear approximation of the error/reject function to approximate the optimal probability of error.

The advantage of these two parametric methods is they do not require a set of known samples except for the requirement for a set of known samples for estimating the parameters of the class conditional densities. Also these two methods estimate the theoretical optimum probability of error. However the estimate might not be reliable if the real scattering of the data of each class deviates very much from the assumed parametric form of the class probability density. Moreover, the error occuring in each class cannot be observed. The non parametric method can provide this information. In our experiments the estimate of the optimal probability of error is calculated from Eq.(7.2) for the parametric method and is listed with the estimate of the probability of misclassification using the non parametric method.

Now we are going to discuss the method for estimating the probability of classification error within the non parametric framework. Suppose we know the parametric form of the class conditional density functions but do not know the values of the parameters. We can estimate these parameters from the training set. We also require an additional set of known class data to estimate the probability of error. It is required that the set of known class samples used to estimate the class parameters, to be known as the *design samples,* be independent of the set of known class samples used to estimate the probability of error [4], known as *test samples.* The reason is that if the design set is used as test set the estimate of probability of error will be biased.

One method for using design samples as test samples is known as the *resubstitution* method[4, 1]. The *hold out* method[4, 1] is an alternative for the resubstitution method. In this method we have two exclusive sets of training samples. One set contains the design samples and the other contains the test samples. However, this method has some drawbacks in that for an insufficient number of training samples, the estimate of the probability of error is very pessimistic or biased to a high probability of error. In addition, this method does not use the training samples effectively.

The concept of the Π method[1] is a compromise between the above methods. In this method, the training samples are divided into a test set with a small number of samples, say k samples, and a design set with a large number of samples i.e. $N-k$, where N/k is integer and $k << N$. The classifier then is trained with the design set and tested with the test set. Suppose the proportion of the test samples classified incorrectly is $\hat{P}_e \left[\Pi\right]_i$. Then select another k test samples and $N-k$ design samples such that these test sets are disjoint. We will have N/k such disjoint test sets and for each pair of test and design sets we train the classifier with the design set and test with the test set. The resulting estimate of the probability of error is

$$\hat{P}_e = \frac{k}{N} \sum_{j=1}^{N/k} \hat{P}_e \left[\Pi\right]_j \tag{7.3}$$

For $k = N/2$, the above method becomes the hold out method in both directions, which is also called the *cross validation* method. For $k = 1$, the above method becomes what is called the *leave one out* method[4]. However since k is small, the disadvantage of the Π method is that it requires very heavy computation and also the estimate, although it tends to be unbiased, has large variance[1].

When the number of training samples is large the hold out method is reliable[1, 4]. We will use the hold out method together with the parametric estimate given by Eq.(7.2) when possible in our performance evaluation. In our case it is reasonable to have a large number of samples in the design and test sets. To estimate how many samples are required in the test set, we will use the following formula[5]

$$N_T = \frac{4p\left(1-p\right)}{E^2} \tag{7.4}$$

where N_T is the number of the test samples, p is the probability of error and E^2 is the allowable error of the estimate of p.

This formula is derived from the variance of a binomial random variable with parameter p, where that variance is given by[6]

$$\sigma^2 = \frac{p\left(1-p\right)}{N_T}$$

and the allowable error E is equal to twice the standard deviation σ.

From Eq.(7.4), since the maximum of the numerator is one, then for $N_T = 1000$ for example, the allowable error E^2 will be 0.001 which is a very small number. A similar method[7] showed that for a class or category probability of error about 0.5, with allowable error $E = 0.10$, the minimum number of test samples is 60. Therefore the performance measure will be based on the hold out method, by testing with a test set consisting of more than 1000 samples per class. We will use more than 1000 samples per class for the design set as well, which guarantees that the estimates of the class parameters are close to the true values.

7.2. Training Sample Acquisition

As discussed in the preceeding section to compare the performances of different linear transformations we need to have a set of known class samples. As is shown in the two dimensional histograms in KL 1 and KL 2 space in Fig.7.1 and KL 1 and KL 3 space in Fig.7.2. the TM data do not show strong multimodality. Three modes are present: the water class, the dense tree and tree/shrub classes and a mode representing data from the remaining classes. Clustering methods such as ISODATA[8] which try to find natural groupings or modes will generally perform poorly on TM data. Reasonable results for clustering of MSS data has been reported[9] but the procedure was controlled such that the clustering results are always confirmed with other sources of information. It is also shown in the above report that the cluster centers are almost uniformly distributed in feature space. This also indicates that there are no strong modes to govern the clustering. The use of *ground truth* in the form of aerial photos or ground maps can lead to choosing an unrepresentative set of training samples because of the different intepretations of land use and land cover. There can also be changes in the objects on the ground, leading to large differences between the land cover on the satellite image and on other images taken at different times. Finally, the acquisition of real ground truth can be very costly. An interactive clustering for the TM data, to avoid the above problems, is presented in the following section.

7.2.1. Interactive ML Clustering

The interactive maximum likelihood (ML) clustering is a method of clustering where the data are classified using a maximum likelihood classifier based on a Gaussian assumption with some interventions by the operator in the case of:

1. Finding the training samples of recognizable classes on the satellite image.

2. Resolving class conflicts.
3. Creating classes.

Those three interventions are done using the high resolution Gould IP8500 image processor with its software support. The first intervention is done in the initial step. In observing the image we can nearly always distinguish some of the classes. The principal distinguishing characteristics are due to the class-distinct reflectances. For example, water has low reflectance in the visual spectrum. The distinctions are also due to class-distinct spatial forms, for example, man made objects, streets, freeways. Once we can identify those classes on the image, we can create a *blotch* file using the Gould software by creating a binary map overlaying the image. Using the blotch file, we can mask the images to get the values of the pixels over the area of interest indicated by the blotch. Sets of masked points become the design sets for estimating the class mean vectors and covariance matrices. These parameters then will be used in the initial ML classification of the assumed Gaussian equally likely classes.

Before we continue with the discussion of the operator interventions we need to discuss how to represent the images such that the process of recognizing the classes from the images is made easier. The TM data are seven dimensional images and it is very difficult to observe those seven images simultaneously. To extract as much information as possible from the seven dimensional data by viewing a smaller dimensional image i.e. with the managable three dimensional data we apply the KL expansion (see Sec.3.) to the original seven dimension image.

The KL expansion or transformation is based on the eigenvectors of the sample covariance matrix of the data, calculated using:

$$C = \frac{1}{N-1} \sum_{i=1}^{N} \left(\underline{x}_i - \overline{\underline{x}}\right) \left(\underline{x}_i - \overline{\underline{x}}\right)^T \tag{7.5}$$

where $\overline{\underline{x}} = \frac{1}{N} \sum_{i=1}^{N} \underline{x}_i$ is the total mean vector of the data. The mean vector, the covariance matrix, the correlation coefficient matrix and the eigenvectors and eigenvalues are given in Table 7.1, 7.2, 7.3 and 7.4 respectively. From the correlation coefficient matrix of Table 7.4 it is observed that very high correlations occur among bands 1, 2 and 3 i.e. the visible bands and between bands 5 and 7 i.e. the near and middle infra red bands. Therefore, dimensionality reduction is feasible because of those high correlations. Since we will use only three dimensional data then the KL transformation will use only the three eigenvectors corresponding to the first three largest eigenvalues. These three eigenvalues carry more than 97 percents of the total variance, where the largest eigenvalue of the discarded eigenvectors only has a little more than one percent of the variance. This eigenvector selection follows almost exactly the *rule of thumb* suggested by Merembeck and Turner[10].

The TM image used in this experiment was acquired on Jan. 25, 1983 and its scene identification number is 40193-16315. The study area in that scene is the Walnut Creek Watershed east of Austin Texas with a size of 982×1024 pixels. One thing worth noting is that the data type in the original image is byte, while the data after KL transformation are real which require four bytes per data sample. Therefore, reducing the dimensionality from seven to three while increasing the number of bytes from one to four yields an increase in the memory requirements by almost twice that of the original data. To acquire an actual reduction, the KL transformed data has to be converted to byte as well. The byte type data represents eight bit data with dynamic range from zero to 255, uniform quantization is used in the conversion. Each KL transformed sample is scaled by 1.3784 before quantization so that the minimum and the maximum data value of the first KL transformed data are equal to zero and 255 respectively. The third KL transformed image needs to be shifted by 25 in addition to the scaling

Table 7.1
Data Mean Vector

Band	1	2	3	4	5	6	7
$\bar{x}_i$	77.45	28.63	31.97	41.88	61.89	102.48	28.54

Table 7.2
Data Covariance Matrix

Band	1	2	3	4	5	6	7
1	79.16						
2	42.95	25.42					
3	62.95	36.46	58.22				
4	34.07	22.60	32.29	75.05			
5	101.32	61.32	101.81	81.09	319.59		
6	4.28	2.95	5.70	3.29	23.57	7.99	
7	66.24	38.91	62.98	36.80	164.22	12.44	98.13

Table 7.3
Data Correlation Coefficient Matrix

Band	1	2	3	4	5	6	7
1	1.						
2	0.96	1.					
3	0.93	0.95	1.				
4	0.44	0.52	0.49	1.			
5	0.64	0.68	0.75	0.52	1.		
6	0.17	0.21	0.26	0.13	0.47	1.	
7	0.75	0.78	0.83	0.43	0.93	0.44	1.

Table 7.4
Eigenvectors and eigenvalues of Data Covariance Matrix

Eigenvector	1	2	3	4	5	6	7
	.305	0.631	-.219	-.214	0.106	-.567	-.288
	.181	0.328	-.066	-.020	0.064	0.039	0.922
	0.289	0.4	-.148	0.027	0.153	0.804	-.253
	0.221	0.229	0.93	0.152	-.062	-.031	-.049
	0.754	-.518	0.196	-.374	0.147	-.019	0.020
	0.052	-.109	-.031	0.574	0.796	-.143	-.001
	0.413	-.052	-.236	0.679	-.549	-.091	-.023
Eigenvalues	529.48	64.35	52.60	6.76	5.40	3.84	1.13
%	79.79	9.70	7.93	1.02	0.81	0.58	0.17

before quantization to maintain a transformed range of zero to 255.

It can be shown[11] that shifting will not change the classification result of the ML classifier for Gaussian classes. Moreover, scaling of each dimension, which amounts to multiplying the data vectors by a non singular diagonal matrix also will not change the classification result of the ML classifier of Gaussian classes.

The next operator intervention is to observe whether there are conflicts among the training sets for each class and how to resolve this. Class conflicts occur whenever the training samples from two or more classes are highly overlapped in feature space. There are several ways to detect this, and one of them is by observing the classification results based on the training sets. If the classification result shows a lot of small patches of different classes inside an area classified as a class then a class conflict can be assumed to occur. For example, after the initial classification we observe a lot of small patches of the barren class inside the area classified as residential. Another way to observe the conflict is by manually plotting the position of the class mean vectors and their class scattering, measured by their standard deviation, around the mean vectors on a two dimensional axis in feature space i.e. KL 1 and KL 2 space and KL 1 and KL 3 space. A computation method of confirming a conflict is by using the Lissack-Fu distance measure[2] with parameter $s=1$,

$$J_1(\omega_i,\omega_j)=E\left|P(\omega_i\mid\underline{x})-P(\omega_j\mid\underline{x})\right| \tag{7.6}$$

This distance has a direct relationship with prabability of error for a two class problem:

$$P_{eij}=\frac{1}{2}\left[1-J_1(\omega_i,\omega_j)\right] \tag{7.7}$$

For multivariate normal classes the distance measure can be estimated by

$$\hat{J}_1(\omega_i,\omega_j)=\frac{1}{N_{ij}}\sum_{k=1}^{N_{ij}}\tanh\left|\frac{1}{2}d_{ij}(\underline{x}_k)\right| \tag{7.8}$$

where

$$d_{ij}(\underline{x})=\frac{1}{2}(\underline{x}-\underline{m}_j)^T\Sigma_j^{-1}(\underline{x}-\underline{m}_j)-\frac{1}{2}(\underline{x}-\underline{m}_i)^T\Sigma_i^{-1}(\underline{x}-\underline{m}_i)-\ln(h_{ij})$$

$$h_{ij}=\frac{P(\omega_j)\left|\Sigma_i\right|^{\frac{1}{2}}}{P(\omega_i)\left|\Sigma_j\right|^{\frac{1}{2}}}$$

and N_{ij} is the total number of samples from classes ω_i and ω_j.

It is shown in the cited reference[2] that the estimate given in Eq.(7.8) is unbiased and always has smaller variance than the one found from the non parametric method. The distance between classes is considered far if the estimate is close to one. Since this distance depends on the distribution function of both classes then for multivariate normal classes, the changing of the classes parameters i.e. the mean vectors and the covariance matrices of either one of the classes or of both classes can change the distance. Moving the class mean vector and changing the covariance matrix is one way to resolve the conflict.

Other way to resolve the conflict is to observe on the Gould screen the two dimensional histogram i.e. on KL 1 and KL 2 and/or on KL 1 and KL 3 the samples taken from the classes found in conflict. From the observation we may find that there is a high density of points at the boundaries between classes. If this happens then we can assume that there is another class

shown by those dense points and we can create a new class, or subclass, for these points. The new class is created by approximating the mean vector by position of the dense points and approximating the covariance matrix by the scattering of the dense points.

In addition to creating new classes because of conflicts, we can also create new classes in the regions of the two dimensional histogram of the data that are *empty of class* definitions but which have points scattered in the region. Creating classes in this case means that we assume that there is a class in each area but it was not recognizable in the image. After we have all the mean vectors and covariance matrices both of the known and new classes, we again apply ML classification. From the result of the classification, we again observe whether there are classes still in conflict, whether the created classes are consistently occupy solid area in the classified image or whether the created classes are consistent with the ground truth data. In the next step we will discuss how to enhance the classified image such that the classes we are interested in are easier to distinguish.

After each classification we will have a map with each pixel labelled with number associated with its assigned class. The numbering will be from one to the number of classes. For example, if we have 22 classes then the numbering will be from class 1 to class 22. One way to enhance the classified image is by creating a psuedocolor display of the classified image using a program available in the Gould system. Some classes may have very little difference in their colors or low color contrasts due to the method used for class numbering. If we are interested in a close examination of a few class, we can enhance the classified image by changing the number assignments of the classes of interest. than the rest of the class numbers. For example, say we are interested in the residential class which is assigned the number five. Suppose we have 22 classes, then we change each pixel assigned to five to a large number larger than 22, say 50. Then we can recreate the pseudocolor display of the classified image. Because the number 50 is much larger than the rest of the class labels, pixels from this will be very distinct, and thus, easier to observe. A class is assumed to be properly defined whenever it has solid large patches in the classified image. Solid means that not too many pixels from other classes occupy large, homogeneous regions. The classification of the large patches can also be confirmed by use of the ground truth data.

This enhancement of the classified image by changing the class labels is very helpful in determining the class descriptions, especially when we create new classes. At times it is difficult to recognize a class directly from the image. However after samples from this *class* have been assigned to a specific number and enhanced, the spatial distribution of assigned pixels can give more information about the description of the class. Comparison with aerial photos and maps is also made easier because of the high color contrast of the enhanced image.

7.2.2. Result of the Interactive ML Clustering

As discussed in the preceeding section, the first step in the clustering process is to find the locations of the pixels of recognizable classes. For this process and subsequently for the rest of the process we will use the three dimensional KL transformed data or image. The general observation of the KL 1 image shows that some structures are clearly shown. They are streets and/or highways, blocks of residential areas, airport runways, rivers and lakes and some agricultural fields. The observation also shows some distinct spectral characteristics of different classes. For example, water is dark, barren is the most white, dense tree is dark almost like water but without a regular structure, tree/shrub is dark grey, agriculture classes are dark grey to grey and residential or man made classes are light grey. Some classes are difficult to identify. For example, residential areas have almost the same intensity as their surrounding areas.

General observation of the KL 2 image shows some distinct intensities as well. Water is white/grey, man made classes such as highways, residential, and airport runways are

white/grey, almost the same as the intensity of water. Barren is still the most white, dense tree is grey and shrub/grass i.e. low height vegetation is black. Also in this KL 2 image the residential structures are clearly distinct from the surrounding vegetation. Observation of the KL 3 image shows that the most significant intensities come from man made vegetation fields. They are white regular/rectangular patches or grey and dark patches. Water is black and the rest of the image is grey with unclear structures.

In the initial step we were able to recognize fourteen classes by observing the three KL images in a false color display on the Gould system. The aerial photos and USGS maps were also used in this step. Those fourteen classes are water, residential, transportation/young residential, concrete, dense tree, tree/shrub, savannah, crop 1, crop 2, crop 3, grass 1, grass 2, grass 3 and barren. Then ML classification and operator interventions are applied as discussed previously. The six classes which are agricultural and grass are very distinct such that they do not need any further adjustment until the end of the clustering process. This is also true for classes dense tree and tree/shrub.

The water class must be modified as we observe after the initial classification that some water is classified as either dense tree or tree/shrub. This is resolved by defining a shallow water class. Changes must also be made in the Barren class. It is observed that there are too many barren assignments in the residential and water classes. One of the reasons for this is that the variances of barren are very large since its training samples occupy a rather large space in KL space but with very low density. To resolve the misclassifications, we reduce the variances of barren by almost a half and modifiy the covariances such that the cross correlation coefficients are constant.

We also create two more classes close to the barren class in the KL transformed feature space. The region where we put these new classes is *empty* of class definitions but there is a large but low density data scattering of samples in the two dimensional scatter diagrams. We also observe that the concrete class occupies some irregular regions which is not desirable, so we create another class from these irregular regions. The training samples for the concrete class were found from the airport runways. After another classification and comparison with the maps, the newly created class appears to be the gravel class. In the interactive process, other man made classes are also defined.

The most difficult class conflicts usually occurs among man made classes, barren classes and among classes such as savannah and grass/shrub classes, which usually are not too homogeneous. Savannah is the transition from tree/shrub to grass and grass/shrub is the transition from tree/shrub to grass. Man made objects are often transitional also, because they are commonlly occupied by several different classes. For example, the transportation class is frequently a combination of trees, streets, soil and grass. The density of the surrounding vegetation can also indicate man made classes. For example[12], single family and older residential usually have a canopy of dense trees which allow them to be distinguished from multi family housing/young residential because the latter has a lower density of trees or vegetation. In our case, the young residential class is found after observing the two dimensional histograms of pixels classified as transportation, residential and shopping center. The histograms show that there is a region with a rather high density of points in the boundary between residential and shopping center. We then create a class based on the position and scattering of these dense points and after subsequent classification and comparison with the aerial photos and maps it appears that these points come from young residential.

After eleven manual iterations, or eleven class adjustments and classifications, we established that the clustering was satisfactory. We defined twenty two classes occupying a percentage of the study area as given in Table 7.5. The percentage area of a class is the number of pixels classified as that class divided by total number of pixels. The eleven manual iterations

Table 7.5
Result of Interactive M.L. Clustering

Class Number	Class Name	Number of pixels	Area %
1	Water 1(deep)	7891	0.78
2	Water 2(shallow)	15837	1.57
3	Residential 1(old)	99865	9.93
4	Residential 2(young)	38691	3.85
5	Shop/Comm. 1	13841	1.38
6	Shop/Comm. 2	8017	0.80
7	Concrete/Ind.	3505	0.35
8	Transport. 1	34215	3.40
9	Transport. 2	26139	2.60
10	Gravel 1	4466	0.44
11	Gravel 2	10932	1.09
12	Barren	6160	0.61
13	Dense Tree	127267	12.66
14	Tree/Shrub	201136	20.00
15	Savannah	171661	17.07
16	Grass/Shrub	45562	5.53
17	Grass 2(irrig.)	31615	3.14
18	Grass 1	38895	3.87
19	Grass 3(dry)	7496	0.75
20	Crop 3	45498	4.52
21	Crop 1	12760	1.27
22	Crop 2	54119	5.38

required was rather large because of the lack of experience of the author in image intepretation, especially in the early stages. In the following section an attempt to quantify the quality of the interactive ML clustering process in terms of probability of error is presented.

7.2.3. Evaluation of the Interactive ML Clustering

We want to quantify the quality of the interactive ML clustering in terms of probability of error. To do this we need to apply the classification to a subset of the data. The subset of the data have to be representative training samples. The representative training samples are acquired by random selection[13, 7]. In our case the random selection is done using a uniformly distributed $[0,1]$ random number generator. We first define how many training samples are required for a class. Then we find the ratio of the number of the required training samples to the number of pixels of that class. For example, for water 1 class we require 3000 training samples. From Table 7.5., the number of pixels available from water 1 is 7891. The ratio of the required training samples to the number of available samples is 0.38018. The classified image is scanned, and when a pixel of water 1 class is encountered a random number is generated. If the number is less than or equal to the ratio i.e. less than or equal to 0.38018 then that pixel will be included in the training set. We will get close to 3000 training samples and we will select the first 3000 of them. If we get less than 3000 samples than we can increase the ratio slightly and do the sampling again. In our experiment we require that each class be represented by 3000 training samples.

Basically our interactive ML clustering end result is produced by a ML classification. Hence we will apply ML classification to the training samples as well. Since the training samples provide a good representation of each class, the resulting probability of error of the ML classification of the training samples will give a good indication of the probability of error of the ML classification of the full image. In the ML classification of the training samples, following the discussion in Sec.7.2., the design sets and the test sets of each class will be different. One more condition to be met by the training samples, the training samples must be transformed by the KL transformation to be come a three dimensional feature space, as described in Sec.7.2.1.

The classification of the training samples discussed in the preceeding paragraph is also a way to test the sensitivity of the class assignments of the data to the changing of the decision boundaries. In the Gaussian ML classifier, the decision boundaries are dependent on the class parameters. By using the new design set to estimate the class parameters, we expect that these new parameter estimates will be different, although not greatly different, than those used in the last classification of the interactive ML clustering. If the slight changing of the decision boundaries produces large probability of error estimates, then the number of points per hypervolume or the density of points in the regions close to the decision boundaries is high, which subsequently indicates that the probability of error in the last classification of the interactive ML clustering is high as well. This is because the probability of error mainly depends on the density of points in the region close to the desicion boundaries. Therefore the classification of the training samples discussed in the preceeding paragraph, allows us to quantify the probability of error of the last classification of the interactive ML clustering method.

The design sets for each class will consist of 1200 samples and the test sets of each class will consist of 1800 samples. The classification will be done twice, so we create for each class a second design set consisting of 1200 samples, different from the design set of the first classification. The test set consists of the 1800 remaining samples.

The result of the ML classification of the training samples is shown in Table 7.6. The first and the second columns of Table 7.6 are the results of the ML classifications, and the third column is the average of the first and the second columns. The figures in the first up to the 22^{nd} row of the first and the second columns are the estimates of the class conditional

Table 7.6.
Estimate of the Probability of Error
of the Interactive M.L. Clustering
$\hat{P}_e \mid \omega_i$ (%)

Class	First Classification	Second Classification	Average
1	0.06	0.33	0.20
2	11.11	11.5	11.31
3	10.94	11.83	11.39
4	12.06	12.72	12.39
5	11.89	13.44	12.67
6	5.28	3.72	4.5
7	3.89	3.94	3.92
8	20.44	19.11	19.78
9	14.17	13.83	14.00
10	13.89	14.33	14.11
11	10.44	12.56	11.50
12	6.5	6.78	6.64
13	4.06	3.00	3.53
14	14.45	14.17	14.31
15	18.5	16.61	17.56
16	6.83	6.22	6.53
17	4.72	5.39	5.06
18	6.78	5.94	6.36
19	1.61	2.17	1.89
20	0.61	0.83	0.72
21	0.67	1.17	0.92
22	10.17	8.17	9.17
$\hat{P}_{e'}$ (%)	8.59	8.54	8.57
$\hat{P}_{eW}$ (%)	11.17	10.63	10.90
$\hat{P}_e$ (%)	18.78	18.89	18.84

probability of error[1], given by

$$\hat{P}_e \mid \omega_i = 1 - \frac{N_{Ci}}{N_{Ti}} \tag{7.9}$$

where N_{Ci} is the number of the correctly classified pixels from class ω_i and N_{Ti} is the number of samples in the test set for class ω_i. The 23^{rd} row of the first and the second columns are the averages of the class conditional probability of error, which are calculated as follows,

$$\hat{P}_e = \frac{1}{C} \sum_{i=1}^{C} \hat{P}_e \mid \omega_i \tag{7.10}$$

where C is the number of classes. The 24^{th} row of the first and the second columns are the weighted averages of the class conditional probability of error, which is calculated as follows,

$$\hat{P}_{eW} = \sum_{i=1}^{C} \hat{P}(\omega_i) \hat{P}_e \mid \omega_i \tag{7.11}$$

where $\hat{P}(\omega_i)$ is the estimate of the class apriori probability found from the percentage area classified as class ω_i in the interactive ML clustering which is shown in Table 7.5. And the last row of the first and second columns are the estimates of the optimal or Bayesian probability of error calculated from Eq.(7.2).

7.2.4. Discussion of the Interactive ML Clustering Results

Before we proceed further in the discussion, we want to emphasize what is happening in one of the operator interventions i.e. in the class creation or class synthesis. It may not be so clear in passing what is actually happening in that process. If the operator believes that there is a class in a region of the three dimensional space spanned by the three KL axes, then the operator may create a class by approximating the class mean vector from the position of the approximated center of that class and approximating the covariance matrix by the scattering of the data of the synthesized class. See section 7.2.2. for the reasons why the operator may believe that an additional class should be present in the three dimensional space. These class parameter approximations clearly are far from accurate. However after classification is applied using all the class parameters including the class parameters of the synthesized classes, we can make some observations of the spatial distribution of the displayed classified image. By using the class enhancement technique discussed in section 7.2.1. we can observe in more detail the spatial distribution of classes in the classified image. For the synthesized classes, we can select out of all the pixels assigned to the class those which constitute the most *representative* training samples for each class. We only select as training samples the pixels which come from spatially solid blocks. We will pick the training samples from several different spatially solid blocks of the classified image for each synthesized class. Therefore, after the training sample acquisitions for the synthesized classes, the class parameters are estimated as for any other class i.e. from the training samples of the associated class. A summary of the class creation or class synthesis process is as follows. Initially the class parameters are approximated roughly, then classification is performed. The classified image is then observed and the training samples for the synthesized classes are acquired and better estimates of the class parameters are computed from the acquired training samples. A synthesized class is not necessarily maintained after initial classification. It might be discarded if the observation of the classified image shows that the synthesized class does not appear to have large solid blocks in the classified image.

Although we cannot include the end result of the interactive ML clustering in the dessertation because of the restriction of color photographs, the results are encouraging. Most of the

image is occupied by blocks of classes rather than by small patches. However, some inconsistencies are seen in man made features such as shopping centers, residential or transportation categories.

From Table 7.6. we have the information needed to asses the quality of the interactive ML clustering. The first and second classifications, shown in the first and second columns of the Table 7.6., show that those two classifications have similar results. The largest difference in the estimated class conditional probability of error $\hat{P}_e \mid \omega_i$ is for class 22 which is about 2.0%. Otherwise, the two estimates are very consistent. From this consistency we can draw some conclusions. First, the samples which are assigned as design sets in both classifications clearly are representative of the classes. Remember that these two design sets, one for each classification, are totally different sets. The fact that they yield a very consistent results indicates that they are representative of the classes. The second conclusion is that since the error estimates from the two sets are similar, the entire set of training samples provide a good representation of the classes. It would be very difficult to justify this conclusion if the results of these two classifications were totally different. While we performed only two classifications, the large number of samples in the design sets for each class, which is 1200 samples/class, and in the test sets of each class, which is 1800 samples/class, provide strong support for the conclusion.

We will use the average of the results of the first and the second classifications i.e. the third column in Table 7.6. as the basis for our ensuing discussion. We next want to consider the last three rows, consisting the average of the class conditional probability of error from Eq.(7.10), the weighted average of the class conditional probability of error from Eq.(7.11) and the estimate of the optimal Bayesian probability of error from Eq.(7.2). First we need to remind ourselves that the optimal Bayesian probability of error and certainly its estimate depend on the applied classification rule. This rule is the minimum probability of error classifier or the maximum aposteriori decision rule, as described in Sec.3.1. One significant problem that arises here is that in the classification we are applying, we assume that all the classes are equally likely. We know that the aposteriori probability is dependent on the apriori probability and that the probability of error or more precisely its estimate calculated using an equally likely assumption will be different than if the correct apriori probabilities were applied. Second, the optimal Bayesian probability of error or its estimate certainly depends on the accuracy of the class conditional probability density functions in representing the distribution of each class. In terms of the TM data, the deviation from the Gaussian class distribution assumption is quite possible especially in the hypervolumes at the tails of the distribution. It is expected that the density of data at the tail fall to zero abruptly rather than following the asymptotic curve. Clearly this will yield a Bayesian probability of error estimate that is larger than it should be. Therefore the estimate of the optimum Bayesian probability of error shown in the last rows of the Table 7.6. will not be used as the quality measure. This leaves the unweighted and weighted averages of the class conditional probability of error for consideration. Recall that these two are nonparametric estimates, as described in Sec.7.1. We will use the weighted average as our quality measure. The weights are the class proportions or apriori probabilities. The principal reason for selecting the weighted average as the quality measure is that the error occuring in classes with low probability should have a small effect on the overall probability of error, and just the opposite, classes with high probability should have a strong effect on the overall probability of error.

In terms of Bayesian classification procedure, the equally likely class assumption in the non equally likely class situation (see Sec.3.) will yield a probability of error that is not minimum. However, in terms of non parametric probability of error estimation, the classification rule applied does not affect the estimate. What is important is the percentage of class ω_i samples that are correctly or incorrectly classified. We can make this observation because the classification of each of the points to be classified is known before hand. Therefore although the

classification rule applied is not necessarily optimum, but the resulting probability of error estimate is the *correct* one for that particular classification rule. In this regard therefore, the overall quality of the interactive ML clustering is characterized by the 10.9% probability of misclassification (see Table 7.6. row number 24 of the third column). There are two classes which have class conditional probaility of error larger than 15%. These are classes 8 and 15. In terms of the class proportions, from Table 7.5., the more important class is class 15 (savannah) because it occupies about 17% of the study area. Twelve classes have class conditional probability of error below 10%.

Grouping of classes may be more desirable in intepreting the classification result. For example, there may be little concern if the shallow water and deep water classes are mixed up by the classifier. On the other hand there might be more concern if the classification yielded many errors between water classes and man made feature classes. For this reason then we will group our original 22 classes into 10 new classes where the new classes and their class members are shown in Table 7.7. This grouping is similar to level I of the USGS land use/land cover classification system[5] and similar to the grouping done in Algazi[12].

Similar to Table 7.6, Table 7.8 shows the class conditional probability of errors for the new classes. The weighted average of the class conditional probability of error is also shown. Similar to Table 7.6 as well, the first and second columns of Table 7.8 show the probabilities of error of the first and second classifications respectively and the third column shows the average of the first and the second columns.

The class conditional probability of error is calculated by assuming that an error has occurred if a pixels of a new class is assigned to another class. The difference between this result and that is given in Table 7.6 i.e., for the old classes, is that the errors among classes which are members of the new class are not accounted for. The weighted average of the class conditional probability of error, shown in the last row of the third column of Table 7.8, is 9.22%, an improvement of about 1.5% over that for the original classes.

To conclude the discussion of the interactive ML clustering method we note that there are two important characteristics of this method, one a disadvantage and the other one an advantage. The disadvantage is that this method depends heavily on the interactive image processing system and its associated software. This system is needed not only as the tool to observe the intermediate classification result but also to perform some of the processing. This processing includes the *blotching* process, the *pseudocoloring* process to enhance the observation of the result of the classification process and many others which are needed in the operator interventions while applying this interactive ML clustering. On the other hand since the classification rule used in this method is basically a supervised classification rule i.e. the Gaussian maximum likelihood classifier. By applying that classification rule to the training samples which are representative of the total data we can quantitatively assess the quality of the clustering process. The quantitative measure is the probability of classification error shown in Table 7.6 or Table 7.8.

7.3. Linear Transformation Experiments

In the following subsections, we present the procedure of the experiments, implementations of each linear transformation and discussions of the results of the experimental evaluations and comparisons of the linear transformation methods for dimensionality reduction. The training samples used in the class parameter estimations and in the non parametric probability of error estimations in the experiments are the training samples which have been used to measure the quality of the interactive ML clustering method, as discussed in Sec.7.2.

Table 7.7.
Class Grouping

Class Number	Class Name	Class Members	Area %
1	Water	Water 1 Water 2	2.35
2	Residential	Residential 1 Residential 2	13.78
3	Comm/Ind./Transp.	Shop/Comm.1 Shop/Comm.2 Concrete/Ind. Transport.1 Transport.2	8.53
4	Barren/Mining	Gravel 1 Gravel 2 Barren	2.14
5	Dense Tree	Dense Tree	12.66
6	Tree/Shrub	Tree/Shrub	20.00
7	Savannah	Savannah	17.07
8	Grass/Shrub	Grass/Shrub	5.53
9	Grass Field	Grass 2 Grass 1 Grass 3	7.76
10	Crop/Pasture	Crop 3 Crop 1 Crop 2	11.17

Table 7.8.
Estimate of the Probability of Error
after class grouping shown in Table 7.7.

Class	$\hat{P}_e \mid \omega_i$ (%) First Classification	Second Classification	Average
1	0.53	0.36	0.45
2	10.06	10.81	10.44
3	7.53	7.42	7.48
4	6.20	6.80	6.50
5	4.06	3.00	3.53
6	14.44	14.17	14.31
7	18.50	16.61	17.56
8	6.83	6.22	6.53
9	1.52	1.85	1.69
10	2.15	1.85	2.00
$\hat{P}_e$ (%)	7.18	6.91	7.05
$\hat{P}_{eW}$ (%)	9.47	8.97	9.22

7.3.1. Procedure

In this section we will discuss the procedure which will be followed in the experiments to compare several linear transformation methods for dimensionality reduction. Some general observations will be made about the training samples. These samples represent the TM image of some area on the ground. From the discussion in Sec.7.2., it has been shown that these samples are *good* representations of the TM image of that area i.e. by making a *random selection* of the training samples from the overall image. Therefore the results of the experiments can be assumed to represent the results of experiments applied to the total image. The results of these experiments not only will show some differences and similarities among several linear transformation methods but also will indicate how a *typical* TM image behaves under these linear transformations. What we mean by *behaviour* is the trend of the probability of error for a certain linear transformation for a certain dimensionality.

Now the question is *how typical* is our image or more precisely how typical is the area contained in our image. One way to measure this is by observing what land cover categories are represented in the image. Do these classes represent a *broad* enough range classes such that we can say that it is a typical image? For this let us go back to Table 7.5. which shows the list of classes contained in the image. Out of the 22 classes, there are man made features, forests i.e. dense tree and tree/shrub, crops,grass fields etc, which shows that not only the number of class is rather large but the *range* of the classes is large as well. In fact if we compare the list of the classes with level I of the USGS Land Cover/Land Use Classification System[5] except for the snow category, all the categories in that classification system are represented in our image. Therefore we can say that our image is a *typical* TM image and thus our experimental results can be said to represent the *behaviour* of the TM images for certain linear transformation and for certain dimensionality.

Now we will discuss the procedure followed in the experiments:

1. Training sample preprocessing:

The experiments will be applied only to the six reflective bands of the TM data i.e. we will not use the thermal band (band 6). Based on the premise that the raw data, i.e. the six dimensional data, will contain the most information then we would like to *force* the class assignment of each training sample to be found from the six dimensional data. Therefore we apply ML classification to all the training samples i.e. the 3000 samples/class used in Sec.6.2, in the six dimensional space or raw data. The class assigned in this classification to each sample is assumed to be the true class. From this step we select 2500 samples/class as the training samples for the subsequent experiments.

2. The Linear Transformations:

The linear tranformations which will be evaluated in the experiments are (see Sec.3.):

a. KL Transform Method.
b. MDA Method.
c. Weighted MDA (First Version) Method.
d. Weighted MDA (Second Version) Method.
e. TM-Tasseled Cap Linear Tranformation Method.
f. SVD Linear Transformation Method.
g. Space Variant Linear Transformation Method.
h. KL Transform-MDA Hybrid Metod.

Detailed discussions about the implementation of those transforms will be presented in subsection 7.3.2. For each method, except for method g., the dimensionality of the transformed

data will be reduced and the feature selection, except for the Tassel Cap Method, will be made using the order of the eigenvalues or the singular values.

3. Classification Rule:

The classification rule which will be used is the maximum likelihood classifier where the class assumptions are Gaussian. The classification rule is as follows:

Assign $\underline{x}$ to class ω_i iff,

$$g_i(\underline{x}) \geq g_j(\underline{x}) \qquad i \neq j \tag{7.12}$$

where

$$g_i(\underline{x}) = -\frac{1}{2}\left[(\underline{x}-\underline{m}_i)^T C_i^{-1}(\underline{x}-\underline{m}_i) + \log|C_i|\right] \tag{7.13}$$

where $\underline{m}_i$ is the mean vector of class ω_i and C_i is the covariance matrix of class ω_i.

The right hand side of Eq.(7.13) is the log version of the Gaussian likelihood function where the class apriori probabilities are assumed equal. It was shown experimentally by Merembeck and Turner[10] that this classification rule gave the best results among other classification rules applied to ground images.

4. Probability of Error Estimation:

The resulting probability of error estimates will be used to compare the linear transformations. The estimates will be computed using the *hold out* method, discussed in Sec.7.1. Out of the 2500 samples/class, 1000 samples/class will be assigned as the design samples or design set and the 1500 samples/class left will be used as the test set. The design set will be used to train the classifier or more preciselly in our case to estimate the class mean vectors and covariance matrices. The test samples will be used to estimate the probability of error by counting the number of pixels classified incorrectly. The large number of design samples, i.e. 1000 samples/class and test samples, i.e. 1500 samples/class will guarantee that the resulting probability of error estimates are reliable.

In the experiments, the number of classes is 22, where the class list is shown in Table 7.5. But the probability of error estimates which will be considered are those of the grouped classes shown in Table 7.7. In the grouped case, the probability of error among classes in the same group will not be accounted for, and therefore the probability of error estimate will be lower compared to that of the 22 class case. However the general trends in both cases are similar for different transformations and dimensionalities. These probability of error estimates are shown in Tables 7.18. to 7.24, and example of an error matrix for the 10 classes is shown in Table 7.25.

7.3.2. Implementations

The theoretical discussions about the linear transformation methods was presented thoroughly in Sec.3 and Sec.5. In this section we will discuss some aspects of the implementations of these transformations. All of the implementations are based on the theoretical discussions in previous chapters, however some modifications have been implemented, which will be part of our discussions in this subsection.

1. The KL Transform Method:

The implementation of the KL transform method will follow closely the supporting theory. The covariance matrix of the data will be estimated from Eq.(7.5). The data here are the design set i.e. the 1000 samples/class, and the test set i.e. the 1500 samples/class. The eigenvectors and eigenvalues of the covariance matrix are shown in Table 7.9. and the probability of error estimates are shown in Table 7.18. Also an example of the error matrix for the 10 grouped

classes is shown in Table 7.25.

2. The MDA Method:

There are two different MDA methods discussed in Sec.3. These involve two different eigen equations, as discussed in Sec.3.5. The first of these is

$$S_A \ \underline{a}_i = \alpha_i \ S_W \ \underline{a}_i \tag{7.14}$$

and the second is

$$\left(S_W^{-1/2}\right)^T \ S_A \ \left(S_W^{-1/2}\right) \ \underline{\hat{a}}_i = \alpha_i \ \underline{\hat{a}}_i \tag{7.15}$$

We want to show that the transformations resulting from each version will give exactly the same classification results if we are using the Gaussian ML classifier.

It can be shown easily that,

$$\underline{\hat{a}}_i = \left(S_W^{1/2}\right) \underline{a}_i \tag{7.16}$$

We know from Eq.(7.15), since the associated matrix is symmetric, that

$$\underline{\hat{a}}_i^T \ \underline{\hat{a}}_j = \delta_{ij} \tag{7.17}$$

or the set of $\underline{\hat{a}}_i$ are orthonormal. Substituting $\underline{\hat{a}}_i$ from Eq.(7.16) into Eq.(7.15) and using Eq.(7.17) will yield

$$\underline{a}_j^T S_A \ \underline{a}_i = 0 \ ; \ i \neq j \tag{7.18}$$

and

$$\underline{a}_j^T S_W \ \underline{a}_i = 0 \ ; \ i \neq j \tag{7.19}$$

which shows that the eigenvectors, $\underline{a}_i$, will also diagonalize both matrices S_A and S_W.

Although both sets of eigenvectors $\underline{a}_i$ and $\underline{\hat{a}}_i$ are closely related as shown by Eq.(7.16), they are different in the sense that they are solutions of two different eigen equations. Usually the eigenvectors are normalized, therefore for the solution of Eq.(7.14),

$$| \ | \underline{a}_i | \ | = \underline{a}_i^T \underline{a}_i = 1 \tag{7.20}$$

and also for the solution of Eq.(7.15),

$$| \ | \underline{\hat{a}}_i | \ | = \underline{\hat{a}}_i^T \underline{\hat{a}}_i = 1 \tag{7.21}$$

These two normalized eigenvectors, using Eq.(7.16), will have relationship,

$$\underline{\hat{a}}_i = Diag\left(r_i^{-1/2}\right) S_W^{1/2} \underline{a}_i \tag{7.22}$$

where the diagonal matrix $Diag\left(r_i^{-1/2}\right)$ is a scaling matrix such that both $\underline{a}_i$ and $\underline{\hat{a}}_i$ are normalized or satisfy Eq.(7.20) and (7.21) simultaneously while also satisfying Eq.(7.16). The diagonal matrix $Diag\left(r_i^{-1/2}\right)$ is certainly non singular.

Let us define the $D \times d$ matrices

$$A = \left[\underline{a}_1, \ldots, \underline{a}_i, \ldots, \underline{a}_d\right] \tag{7.23}$$

and

$$\hat{A} = \left[\underline{\hat{a}}_1, \ldots, \underline{\hat{a}}_i, \ldots, \underline{\hat{a}}_d \right] \tag{7.24}$$

where the eigenvectors $\underline{a}_i$ and $\underline{\hat{a}}_i$ are associated with the d largest eigenvalues, α_i.

Suppose the $D \times 1$ data vector which will be tranformed is $\underline{x}$, and the transformations yield,

$$\underline{y} = A^T \underline{x} \tag{7.25}$$

and

$$\underline{\hat{y}} = \hat{A}^T \underline{x} \tag{7.26}$$

Substituting Eq.(7.20) into Eq.(7.24) yields,

$$\underline{\hat{y}} = A^T \left(S_W^{1/2} \right)^T Diag\left(r_i^{-1/2} \right) \underline{x} = A^T \underline{z} \tag{7.27}$$

where

$$\underline{z} = \left(S_W^{1/2} \right)^T Diag\left(r_i^{-1/2} \right) \underline{x} \tag{7.28}$$

$$\underline{z} = P^T \underline{x}$$

where the matrix P is defined as follows,

$$P = Diag\left(r_i^{-1/2} \right) S_W^{1/2}$$

Matrix $S_W^{1/2}$, as discussed in Sec.3., is non singular and the matrix $Diag\left(r_i^{-1/2} \right)$ is also non singular, therefore the matrix P is also non singular.

Now we will apply the ML gaussian classifier to the $D \times 1$ vector $\underline{z}$ i.e. by substituting vector $\underline{z}$ of Eq.(7.28) into Eq.(7.13),

$$g_i\left(\underline{z}\right) = -\frac{1}{2}\left[\left(\underline{z} - \underline{m}_{zi}\right)^T C_{zi}^{-1} \left(\underline{z} - \underline{m}_{zi}\right) + \log\left| C_{zi} \right| \right]$$

$$= -\frac{1}{2}\left[\left(\underline{x} - \underline{m}_i\right)^T P P^{-1} C_i^{-1} \left(P^T\right)^{-1} P^T \left(\underline{x} - \underline{m}_i\right) + \log\left|P\right|^2 + \log\left|C_i\right| \right]$$

$$= g_i\left(\underline{x}\right) - \log\left|P\right| \tag{7.29}$$

Therefore if

$$g_i\left(\underline{z}\right) = \max_j g_j\left(\underline{z}\right)$$

then from Eq.(7.29),

$$g_i\left(\underline{x}\right) = \max_j g_j\left(\underline{x}\right)$$

as well.

Therefore classification of $\underline{y}$ of Eq.(7.25) and $\underline{\hat{y}}$ of Eq.(7.26) using the Gaussian ML classifier will give exactly the same results. Hence in our experiment with the MDA method we will implement the solution of the eigen equation of Eq.(7.14) only.

The matrices S_A and S_W are created under the assumption that the class apriori probabilities $P\left(\omega_i\right)$ are equal. The eigenvectors and eigenvalues of the MDA method are shown in

Table 7.10. and the probability of error estimates are shown in Table 7.19.

3. The Weighted MDA Method:

The matrices S_A and S_W in Eq.(7.14) can be modified by varying the weights as follows:

$$S_A = \sum_{i=1}^{K} W_i \left(\underline{m}_i - \underline{m}\right) \left(\underline{m}_i - \underline{m}\right)^T \tag{7.30}$$

and

$$S_W = \sum_{i=1}^{K} W_i C_i \tag{7.31}$$

In the following, two versions of the weights W_i are implemented.

3.a. First Version:

In terms of the data scattering, as shown in the two-dimensional histograms in Fig.7.1. and 7.2., there is no clear multimodality shown. On the other hand we want to give high weights to classes with high apriori probability that are also close to their adjacent classes. Because there is no clear multimodality in the data we can assume therefore that classes with high apriori probability reside in the space close to the total mean vector. Also, we can expect that those classes are close together. Based on this idea, we can expect that classes which are far away from the total mean vector will have low apriori probability and will be far away from their adjacent classes. For the latter, the density of the data appears continuously decreases away from the total mean vector, this means that the number of classes per hypervolume or the class density is also decreasing as we move farther from the total mean vector.

Based on this idea we want to have weights which depend on the *distance* of the class mean vectors to the total mean vector. The distance which will be used is the Mahalonobis distance i.e.

$$D_i = \left(\underline{m}_i - \underline{m}\right)^T \Sigma^{-1} \left(\underline{m}_i - \underline{m}\right) \tag{7.32}$$

where D_i is the Mahalonobis distance of the mean vector of class ω_i, $\underline{m}_i$, to the total mean vector, $\underline{m}$, and Σ is the total covariance matrix of the six dimensional data.

The weights W_i will be chosen as follows,

$$W_i = \frac{\left(D_i\right)^{-1/2}}{\sum_{j=1}^{K} \left(D_j\right)^{-1/2}} \tag{7.33}$$

where these weights will be substituted into Eq.(7.30) and (7.31) to define the S_A and S_W matrices. The calculated weights W_i of Eq.(7.33) are shown in Table 7.11., and the eigenvectors and eigenvalues of this method are shown in Table 7.12 and the probability of error estimates are shown in Table 7.20.

3.b. Second Version:

As discussed in 3.a. above, the weights should represent two characteristics, the class apriori and the closeness of a class to its adjacent classes. Thus, the weights in this version will be defined as follows,

$$W_i = \frac{P\left(\omega_i\right) \kappa_i}{\sum_{j=1}^{K} P\left(\omega_j\right) \kappa_j} \tag{7.34}$$

where $P(\omega_i)$ is the apriori probability of class ω_i and κ_i is the closeness measure of class ω_i toward its adjacent classes.

Assuming that we know the class apriori probabilities, $P(\omega_i)$, then the class closeness measure κ_i will be defined as follows. We calculate the class pair Mahalonobis distance

$$d_{ij} = (\underline{m}_i - \underline{m}_j)^T \Sigma_{ij}^{-1} (\underline{m}_i - \underline{m}_j) \tag{7.35}$$

where $\underline{m}_i$ is the class mean vector of class ω_i and $\Sigma_{ij} = \frac{1}{2}(C_i + C_j)$ where C_i is the covariance matrix of class ω_i in the six dimensional data. For each class we find the two smallest d_{ij} and sum them as follows, e.g. for class ω_i,

$$d_{i,jk} = d_{ij} + d_{ik} \tag{7.36}$$

where d_{ij} and d_{ik} are the two smallest distances of the two adjacent classes to class ω_i. Next among all the $d_{i,jk}$ we will select the smallest as a normalizing factor as follows,

$$c = \min_i d_{i,jk} \tag{7.37}$$

For our data, c ==7.70124 and is associated with class 10, the gravel 1 class. The value of the closeness measure κ_i is defined as:

$$\kappa_i = \frac{c}{d_{i,jk}} = \frac{7.70124}{d_{i,jk}} \tag{7.38}$$

where κ_{10} for class 10 will be equal to one. Substituting κ_i into Eq.(7.34) produces the weights W_i.

These weights, W_i, are shown in Table 7.12. The significant difference between these weights and the ones of the first version is that these weights have larger dynamic range. The eigenvectors and eigenvalues of this second version are shown in Table 7.13 and the probability of error estimates are shown in Table 7.21.

4. TM-Tasseled Cap Linear Transformation:

The transformation vectors for this method are shown in Table 7.14. and the probability of error estimates are shown in Table 7.22

5. The SVD-Linear Transformation:

The eigenvectors and singular values of the SVD-Linear Transformation method are shown in Table 7.15. and the probability of error estimates are shown in Table 7.23.

6. The Space Variant Linear Transformation Method:

The implementation for the experiment for this method differs from what is discussed in Sec.6. In this experiment we are more interested in finding the best three dimensional spaces through the space variant linear transformation method. In the original idea, discussed in Sec.6, this method combines the classification procedure with the linear transformation to achieve a *simpler* classification process and better probability of error performance. This is achieved by performing the first classification in a rather simple space i.e. in the preliminary space (PS) which is the first two KL axes. In this step there are two possible outcomes, either making the final decision if the probability of error encountered is tolerable or going to the next feature i.e the third axis otherwise. The third axis is selected on a pixel-by-pixel basis, dependent on the class assignment of the classification in the PS. For each assigned class, the next feature will be the *best* linear combination of the last four KL axes, called the complementary space (CS). This classification strategy can also be called the sequential method.

In our experiment we do not want to implement this sequential strategy, therefore we will not make any final class decisions on the PS, i.e. the threshold $t=0$ as described in Sec.6, but we will always go to the next feature which is dependent on the class assignment of each pixel on PS.

The selection for the next axis is slightly different than what is discussed in Sec.6. as well. We will not use the original MDA method in the CS but a modified form of it. The modification of the MDA method is to use only the among scatter matrix, S_{Ai}, of the associated class ω_i i.e. the class assigned to the pixel on PS. The associated within scatter matrix S_{Wi} will not be used because in the complementary space (CS), the class covariance matrices are so different that the average of them which becomes the within scatter matrix, S_{Wi}, to represent the within class scattering of data in each class member of the group (see discussion of the group of close classes in Sec.6. or the discussion in next paragraph) may not be valid. Therefore we just use the class positions represented by the class mean vectors and we only have to solve the eigen equation of the among scatter matrices, S_{Ai}.

The selection of the close classes with respect to a particular class is also different than what is discussed in Sec.6. Here, since we know the class assignment on PS from the result of the classification on two dimensional KL transform, for each class we can observe the other classes whose data are classified as that particular class. We will consider a class to be close to a particular class if at least one point of that class is classified as that particular class in PS. From this we can find the grouping of the close classes. This idea of grouping will guarantee that the error occuring in the final classification will never be caused by the error in selecting the close classes, since in the final classification the only classes considered in this step are the ones that are members of the close classes group. These groups are shown in Table 7.16.a. up to 7.16.d.

From each group we find the among class scatter matrices, S_{Ai}. For each of them we solve their eigen equations and the eigenvectors associated with the largest eigenvalues are selected to become the coefficients of the linear combination of the last four KL axes of the CS. These eigenvectors are shown in Table 7.17.

A recapitulate, we have modified the space variant linear transformation method in the experiment because we are more interested in finding a *good* three dimensional space in terms of probability of error rather than implementing a sequential classification procedure to *simplify* the classification process, although the developed software is capable of doing this. In fact, the selection of the third feature can be intepreted as a different version of the weighted MDA method. The weights for the within scatter matrices are all zero and the weights for the among scatter matrices are one for the classes considered close to the particular class and zero for the classes considered far from the particular class. The result of the experiment is shown in the last column of Table 7.24.

7. The KL Transform-MDA Hybrid Method:

If in the preceeding implementation, we select different third features for different groups of close classes, in this method we want to search for a single third feature applicable to all data by taking into consideration the class closeness information on PS and the class apriori probabilities. The class closeness measure on PS will be calculated similar to the one used in the Second Version of the Weighted MDA method (see 3.b). It will use the class pair Mahalonobis distance of Eq.(7.35) but applied on PS. Also it will use the three smallest pair distances instead of two and sum them similar to that in Eq.(7.36). Similar to that of Eq.(7.37), we select the normalizing factor, which for our data is $c=11.45587$ of the barren class or class 12. The closeness measures κ_i are calculated similar to those of Eq.(7.38) and the weights W_i are calculated similar to those of Eq.(7.34). The basic difference between this method and the Second Version of

the Weighted MDA method is that the weights W_i are calculated by considering the class closeness on the PS which is the KL 1 and KL 2 space. The search for the third feature is done by using the Weighted MDA method applied on CS i.e. the space spanned by the last four KL axes. The application of the KL Transform followed by the Weighted MDA method leads us to refer to this method as the KL Transform-MDA Hybrid method. The weights for this method are shown in Table 7.11. The coefficients, which consist of the first eigenvector of the Weighted MDA applied on CS, for the linear combination of the last four KL axes to produce the third feature is shown in Table 7.17. The three dimensional transform is shown in Table 7.17.a., where the first and second vectors are the KL 1 and KL 2 axis respectively. The probability of error estimates for this method are shown in Table 7.24.

$\hat{P}_e \mid \omega_i$ in Tables 7.18 to 7.24 are the estimates of the class conditional probabilities of error given by Eq.(7.9). $\hat{\bar{P}}_e$ and $\hat{\bar{P}}_{eW}$ in these tables are the averages of the estimates of the class conditional probability of error as given by Eq.(7.10) and the weighted averages of the estimates of the class conditional probability of error as given by Eq.(7.11) respectively.

7.3.3. Discussions of the Experimental Results

A discussion of the experimental results will be presented and relevant conclusions will be drawn. The data which have been used in the experiments of the dimensionality reductions by linear transformations are training samples of the six reflective bands of the Thematic Mapper (TM) data from the area of Walnut Creek Watershed east of Austin, Texas. It was argued in subsection 7.3.1. that these samples represent *typical* TM data because of the broad range of classes existing in the set of data. Therefore the discussions and the conclusions will be more relevant to TM data than to a general data set. From this point of view we start to discuss the experimental results:

1. From the resulting probability of error estimates, we can divide the linear transformation methods into three categories:

a. The group of the KL and MDA methods, which include the Weighted MDA, Space Variant Linear Trasformation Method, and the KL Transform-MDA Hybrid Methods. For a three dimensional feature space, the worst average of the class conditional probability of error estimates for this group is 6.94 % from the Second Version of the Weighted MDA method, and the worst weighted average of the class conditional probability of error estimates for this group is 7.52 % also from the Second Version of the Weighted MDA method, as shown in Table.7.21.

b. The TM-Tasseled Cap Linear Transformation, where for its three dimensional feature space the average of the class conditional probability of error estimates is 7.27 % and the weighted average of the class conditional probability of error estimates is 8.04 %, as shown in Table 7.22.

c. The SVD-Linear Transformation Method, where for its three dimensional feature space the average of the class conditional probability of error estimates is 10.24 % and the weighted average of the class conditional probability of error estimates is 11.31 %, as shown in Table 7.23.

Recall that, the weights used in the averaging of the class conditional probability of error estimates are the class percentage areas of the ten classes or groups shown in Table 7.7.

2. If we can rank the three categories given above, it appears that the worst category is the SVD-Linear Transformation followed by the TM-Tasseled Cap Linear Transformation and the best is the KL and MDA category.

3. The SVD-Linear Transformation is designed with the assumption that class conditional densities are Gaussian, as described in Sec.3. The rather bad results of this method compared to

Table 7.9.
Eigenvectors and Eigenvalues
of the KL Transform Method

Eigenvectors	1	2	3	4	5	6
	0.3229	0.6286	0.2232	-0.1365	-0.5774	-0.3141
	0.1979	0.3003	0.1312	-0.0697	0.0037	0.9212
	0.3182	0.3944	0.0964	-0.1703	0.8083	-0.2268
	0.2901	-0.4177	0.8445	0.1639	0.0138	-0.0341
	0.7086	-0.4224	-0.3702	-0.4118	-0.1133	0.0075
	0.4111	0.0790	-0.2710	0.8667	0.0117	-0.0100
Eigenvalues	985.55	136.85	88.17	9.66	7.01	1.80
%	80.19	11.14	7.17	0.79	0.57	0.15

Table 7.10.
Eigenvectors and Eigenvalues
of the MDA Method

Eigenvectors	1	2	3	4	5	6
	0.2849	-0.4332	0.2237	-0.5790	-0.2103	-0.0514
	0.1389	-0.3401	0.1916	0.7678	0.6742	0.8300
	0.3768	-0.4587	0.0384	0.1514	0.1267	-0.5522
	0.2382	0.5019	0.8114	0.0115	-0.1208	-0.0317
	0.6475	0.4308	-0.5033	-0.1319	0.2683	0.0287
	0.5307	-0.2211	0.0082	0.1864	-0.6314	0.0422
Eigenvalues	11.84	7.95	2.80	0.13	0.07	0.04
%	51.84	34.83	12.28	0.58	0.30	0.18

Table 7.11.
The Weights for the Weighted MDA Methods
and for the KL Transform-MDA Hybrid Method.

	Weights		
Class	First Version	Second Version	Hybrid Method
1	0.0174	0.0018	0.000088
2	0.0282	0.0151	0.00096
3	0.0952	0.0925	0.091058
4	0.0476	0.0468	0.034238
5	0.0281	0.0215	0.018273
6	0.0302	0.0086	0.006763
7	0.0208	0.0054	0.002958
8	0.0531	0.0043	0.033373
9	0.0464	0.0328	0.022365
10	0.0217	0.0121	0.006717
11	0.0264	0.0210	0.013489
12	0.0172	0.0103	0.009399
13	0.0467	0.1248	0.055323
14	0.1100	0.1796	0.218767
15	0.0618	0.1645	0.237842
16	0.1162	0.0505	0.045643
17	0.0440	0.0295	0.029758
18	0.0457	0.0435	0.035198
19	0.0335	0.0087	0.00461
20	0.0391	0.0318	0.068144
21	0.0156	0.0061	0.003894
22	0.0552	0.0496	0.061141

Table 7.12.
Eigenvectors and Eigenvalues
of the Weighted MDA Method, First Version

Eigenvectors	1	2	3	4	5	6
	0.5187	-0.4420	0.1037	0.6757	-0.2889	-0.1447
	0.4385	-0.3670	0.1492	-0.4697	0.6581	0.8655
	0.5699	-0.3862	-0.1117	-0.4385	0.2385	-0.4731
	-0.0090	0.3447	0.9100	-0.0341	-0.1265	-0.0278
	0.0870	0.6241	-0.3418	0.1992	0.2344	0.0027
	0.4542	-0.1117	-0.0983	-0.2996	-0.5963	0.0724
Eigenvalues	11.05	6.44	2.06	0.10	0.07	0.04
%	55.91	32.59	10.42	0.52	0.34	0.22

Table 7.13.
Eigenvectors and Eigenvalues
of the Weighted MDA Method, Second Version

Eigenvectors	1	2	3	4	5	6
	0.4401	-0.3746	-0.0705	0.6867	0.0812	-0.1776
	0.5528	-0.4438	0.0660	-0.4539	0.1202	0.9078
	0.5364	-0.4152	-0.2045	-0.5057	-0.6000	-0.3536
	-0.0127	0.0206	0.9571	-0.0095	0.1213	-0.0661
	0.1381	0.6845	-0.0764	0.1443	-0.2651	0.0644
	0.4402	-0.1460	-0.1644	-0.2140	0.7307	-0.1041
Eigenvalues	9.64	5.27	1.52	0.09	0.05	0.03
%	58.09	31.74	9.19	0.51	0.32	0.16

Table 7.14.
TM Tasseled Cap Coefficients

	Feature					
TM Band	Brightness	Greenness	Wetness	Fourth	Fifth	Sixth
1	0.3037	-0.2848	0.1509	-0.8242	-0.3280	0.1084
2	0.2793	-0.2435	0.1973	0.0849	0.0549	-0.9022
3	0.4743	-0.5436	0.3279	0.4392	0.1075	0.4120
4	0.5585	0.7243	0.3406	-0.0580	0.1855	0.0573
5	0.5082	0.0840	-0.7112	0.2012	-0.4357	-0.0251
7	0.1863	-0.1800	-0.4572	-0.2768	0.8085	0.0238

Table 7.15.
Eigenvectors and Singular Values
of the SVD Method

Eigenvectors	1	2	3	4	5	6
	0.5405	-0.5276	-0.5820	-0.0202	-0.0713	-0.2920
	0.2441	-0.1812	-0.0864	0.0533	0.0021	0.9472
	0.3197	-0.0733	0.3189	0.6384	0.6101	-0.1046
	0.3589	-0.2988	0.6470	-0.5977	0.0535	-0.0571
	0.5784	0.5064	0.1649	0.2259	-0.5731	-0.0486
	0.2876	0.5811	-0.3261	-0.4254	0.5397	0.0300
Singular Values	798.15	221.12	152.61	116.20	93.741	28.96
%	56.58	15.67	10.82	8.24	6.64	2.05

Table 7.16.a.
Groups of Close Classes
In the Space Variant Lin. Trans. Method.

Class	1	2	3	4	5	6
Groups of Close Classes	1	1	2	3	4	4
	2	2	3	4	5	5
		3	4	5	6	6
		6	8	6	8	8
		8	10	8	9	10
		10	12	10	10	12
		12	13	11	11	
		13	14	12	12	
		20	16	16		
			20	20		

Table 7.16.b.
Groups of Close Classes
In the Space Variant Lin. Trans. Method.

Class	7	8	9	10	11	12
Groups of Close Classes	7	2	4	7	4	5
	9	3	5	9	5	6
	10	4	7	10	6	7
	11	5	8	11	9	10
	12	6	9	12	10	11
		8	10	19	11	12
		9	11		12	
		10	12			
		11	16			
		12	17			
		16	18			
			19			

Table 7.16.c.
Groups of Close Classes
In the Space Variant Lin. Trans. Method.

Class	13	14	15	16	17	18
Groups of Close Classes	2	2	9	3	9	9
	10	3	10	4	15	10
	13	13	12	8	16	12
	14	14	15	9	17	15
	20	20	16	10	18	16
		21	17	11	19	17
		22	18	12	20	18
			20	15	21	19
			22	16	22	20
				17		
				18		
				20		
				22		

Table 7.16.d.
Groups of Close Classes
In the Space Variant Lin. Trans. Method.

Class	19	20	21	22
Groups of Close Classes	9	2	10	14
	10	3	12	15
	12	10	14	17
	17	12	17	20
	18	14	19	21
	19	15	21	22
	21	16	22	
		20		
		22		

Table 7.17.
The Coefficients of the Last Four
KL Axis in the Complementary Space

Space Variant	Class	Coefficients			
	1	0.6562	0.3800	0.6461	-0.0872
	2	0.9970	-0.0524	0.0542	-0.0165
	3	0.9976	-0.0497	0.0462	-0.0120
	4	0.9983	-0.0489	0.0295	-0.0124
	5	0.9842	-0.1486	-0.0561	-0.0778
	6	0.9810	-0.1656	-0.0628	-0.0789
	7	0.9802	-0.1522	-0.0710	-0.1045
	8	0.9932	-0.0861	-0.0485	-0.0613
	9	0.9995	0.0226	0.0008	-0.0229
	10	0.9944	-0.0606	-0.0426	-0.0751
	11	0.9817	-0.1558	-0.0623	-0.0900
	12	0.9755	-0.1877	-0.0587	-0.0983
	13	0.9869	-0.0006	0.1568	0.0365
	14	0.9997	0.0141	-0.0155	0.0082
	15	0.9990	-0.0247	0.0379	0.0028
	16	0.9993	-0.0060	0.0359	0.0019
	17	0.9993	0.0345	-0.0107	0.0112
	18	0.9987	-0.0286	0.0431	-0.0010
	19	0.9985	0.0449	-0.0319	-0.0072
	20	0.9985	-0.0356	0.0421	-0.0031
	21	0.9978	0.0306	-0.0561	-0.0158
	22	0.9996	0.0232	-0.0113	0.0106
Hybrid Method		0.9520	0.2367	0.0131	0.1937

Table 7.17.a.
The Three Dimensional Transformation
of the KL Transform-MDA Hybrid Method
The 1^{st} and 2^{nd} columns are the KL 1 and KL 2 axis respectively

1	2	3
0.3229	0.6286	0.1118
0.1979	0.3003	0.2869
0.3183	0.3944	0.0182
0.2901	-0.4177	0.8363
0.7086	-0.4224	-0.4500
0.4111	0.0790	-0.0546

Table 7.18.
Estimate of Probability of Error
of the K-L Transform.
Classes are Groups as Shown in Table 7.7.

	$\hat{P}_e \mid \omega_i$ (%)					
Class	6-Dim	5-Dim	4-Dim	3-Dim	2-Dim	1-Dim
1	2.40	2.20	1.40	1.17	2.50	99.97
2	5.73	6.20	6.63	6.23	30.70	84.67
3	4.63	4.44	5.37	7.00	21.60	30.27
4	10.07	10.47	8.93	8.91	22.82	68.09
5	3.00	3.00	3.13	4.13	8.73	18.47
6	9.87	9.73	9.80	9.87	46.80	85.27
7	8.00	8.53	9.20	10.73	54.40	76.40
8	6.33	6.73	6.73	7.07	21.87	84.53
9	2.18	2.33	2.31	3.40	11.00	50.47
10	2.18	2.27	2.07	2.44	24.93	34.76
$\hat{P}_e$ (%)	5.44	5.59	5.56	6.10	24.54	63.29
$\hat{P}_{eW}$ (%)	5.94	6.10	6.31	6.93	31.22	62.96

Table 7.19.
Estimate of Probability of Error
of the MDA Method Transformation
Classes are Groups as Shown in Table 7.7.

	$\hat{P}_e \mid \omega_i$ (%)					
Class	6-Dim	5-Dim	4-Dim	3-Dim	2-Dim	1-Dim
1	2.40	1.77	1.30	1.20	2.93	22.20
2	5.73	6.13	7.67	8.17	40.03	93.87
3	4.63	4.71	6.76	7.69	21.63	75.57
4	10.07	8.53	9.29	9.62	22.00	73.78
5	3.00	3.00	3.20	3.80	9.53	19.27
6	9.87	9.20	9.93	9.93	36.20	51.27
7	8.00	8.53	10.20	10.67	46.80	100.00
8	6.33	6.93	7.87	9.87	31.93	57.40
9	2.18	2.31	3.00	3.29	12.13	44.22
10	2.18	2.07	2.91	3.00	24.91	33.93
$\hat{P}_e$ (%)	5.44	5.32	6.21	6.72	24.81	57.15
$\hat{P}_{eW}$ (%)	5.94	5.94	6.99	7.44	29.83	61.64

Table 7.20.
Estimate of Probability of Error
of the Weighted MDA Method Transform (First Version)
Classes are Groups as Shown in Table 7.7.

	$\hat{P}_e \mid \omega_i$ (%)					
Class	6-Dim	5-Dim	4-Dim	3-Dim	2-Dim	1-Dim
1	2.40	1.87	1.30	1.33	2.57	40.67
2	5.73	6.33	7.27	7.63	27.07	97.67
3	4.63	4.76	6.04	7.51	20.87	37.00
4	10.07	9.47	9.29	9.44	23.29	58.13
5	3.00	2.73	3.20	3.73	8.07	70.00
6	9.87	9.73	10.33	9.33	49.47	90.60
7	8.00	8.33	9.27	9.93	54.80	54.80
8	6.33	6.80	7.27	8.87	19.53	55.60
9	2.18	2.24	2.80	3.16	11.18	64.98
10	2.18	2.27	2.73	2.80	26.02	27.13
$\hat{P}_e$ (%)	5.44	5.45	5.95	6.37	24.29	59.66
$\hat{P}_{eW}$ (%)	5.94	6.04	6.72	7.01	31.19	66.30

Table 7.21.
Estimate of Probability of Error
of the Weighted MDA Method Transform (Second Version)
Classes are Groups as Shown in Table 7.7

	$\hat{P}_e \mid \omega_i$ (%)					
Class	6-Dim	5-Dim	4-Dim	3-Dim	2-Dim	1-Dim
1	2.40	1.97	1.33	1.50	4.37	39.00
2	5.73	6.30	7.47	8.43	28.60	95.37
3	4.63	4.64	6.08	7.71	18.32	42.32
4	10.07	10.20	8.93	9.87	26.51	55.96
5	3.00	2.73	3.00	3.60	7.13	59.53
6	9.87	9.93	10.47	9.87	67.20	90.80
7	8.00	8.67	9.40	10.53	43.80	55.93
8	6.33	6.67	7.53	11.73	22.73	68.13
9	2.18	2.24	2.93	3.29	18.02	60.29
10	2.18	2.36	2.69	2.89	29.53	25.93
$\hat{P}_e$ (%)	5.44	5.57	5.98	6.94	26.62	59.33
$\hat{P}_{eW}$ (%)	5.94	6.15	6.79	7.52	39.95	65.45

Table 7.22.
Estimate of Probability of Error
of the Tassel Cap Transform.
Classes are Groups as Shown in Table 7.7.

	$\hat{P}_e \mid \omega_i$ (%)					
Class	6-Dim	5-Dim	4-Dim	3-Dim	2-Dim	1-Dim
1	2.40	1.90	1.50	1.53	8.20	21.90
2	5.73	6.37	6.30	7.53	62.90	89.13
3	4.63	4.44	6.01	7.59	23.65	69.81
4	10.07	10.22	10.11	10.18	31.82	73.11
5	3.00	2.73	3.13	4.40	13.07	22.13
6	9.87	9.93	9.87	11.07	42.20	53.13
7	8.00	8.67	10.13	12.20	56.40	87.20
8	6.33	6.73	7.33	11.67	66.67	64.40
9	2.18	2.27	2.96	3.87	15.24	43.47
10	2.18	2.24	2.60	2.67	9.44	57.60
$\hat{\hat{P}}_e$ (%)	5.44	5.55	5.99	7.27	32.96	58.19
$\hat{\hat{P}}_{eW}$ (%)	5.94	6.13	6.66	8.04	37.21	62.00

Table 7.23.
Estimate of Probability of Error
of the SVD Transform
Classes are Groups as Shown in Table 7.7.

	$\hat{P}_e \mid \omega_i$ (%)					
Class	6-Dim	5-Dim	4-Dim	3-Dim	2-Dim	1-Dim
1	2.40	2.03	1.87	1.57	13.47	25.17
2	5.73	6.07	10.00	12.53	47.87	100.00
3	4.63	4.40	8.25	9.96	33.45	71.91
4	10.07	10.53	13.80	13.16	32.11	63.82
5	3.00	2.93	4.40	4.93	12.47	20.73
6	9.87	9.47	13.27	14.53	30.87	49.60
7	8.00	8.73	14.60	16.87	25.13	77.73
8	6.33	6.40	17.07	18.13	54.73	69.27
9	2.18	2.31	4.78	5.76	22.31	44.22
10	2.18	2.18	3.24	4.93	35.18	31.67
$\hat{\hat{P}}_e$ (%)	5.44	5.51	9.13	10.24	30.76	55.41
$\hat{\hat{P}}_{eW}$ (%)	5.94	6.02	9.80	11.31	31.18	58.48

Table 7.24.
Estimate of Probability of Error
of the Space Variant Linear Transform and
the KL Transform-MDA Hybrid Method.
Classes are Groups as Shown in Table 7.7.

	$\hat{P}_e \mid \omega_i$ (%)					
Class	6-Dim	5-Dim KL	4-Dim KL	3-Dim KL	Sp.Var.	Hybrid
1	2.40	2.20	1.40	1.17	1.20	1.13
2	5.73	6.20	6.63	6.23	6.30	7.93
3	4.63	4.44	5.37	7.00	7.07	7.77
4	10.07	10.47	8.93	8.91	9.18	10.02
5	3.00	3.00	3.13	4.13	4.07	3.80
6	9.87	9.73	9.80	9.87	9.93	9.40
7	8.00	8.53	9.20	10.73	10.73	10.13
8	6.33	6.73	6.73	7.07	7.07	7.93
9	2.18	2.33	2.31	3.40	3.40	3.33
10	2.18	2.27	2.07	2.44	2.44	2.96
$\hat{P}_e$ (%)	5.44	5.59	5.56	6.10	6.14	6.44
$\hat{P}_{eW}$ (%)	5.94	6.10	6.31	6.93	6.96	7.12

Table 7.25.
The Error Matrix
of the 3-Dim. KL Transform.
Classes are Groups as Shown in Table 7.7.

Assigned	Reference Class									
Class	1	2	3	4	5	6	7	8	9	10
1	2965	16	16	21	29	1	0	0	0	3
2	6	2813	83	14	31	28	0	7	0	6
3	3	120	6975	315	0	0	0	63	50	0
4	3	12	326	4099	0	1	0	3	64	0
5	18	1	0	0	1438	40	0	0	0	1
6	0	5	0	0	2	1352	4	1	1	18
7	0	0	0	0	0	18	1339	3	7	14
8	0	20	20	4	0	18	32	1394	7	34
9	0	0	78	39	0	0	66	25	4347	34
10	5	13	2	8	0	42	59	4	24	4390

the other methods indicate that the class conditional densities of the classes in the TM data are not Gaussian although they may be still unimodal. This conclusion is intuitively reasonable since ground objects in the pixels, depending on the resolution of the data acquisition system are combinations of several *primary* objects such as soils, vegetation, man made objects, grass etc. These combinations are rather constant for the more homogeneous objects such as grass fields, forest or agricultural categories. But the combination will change rather abruptly from one homogeneous object to any other homogeneous object. These characteristics also occur in less homogeneous objects such as in residential areas, but with larger variations of the reflectance of the data than of the more homogeneous objects. Therefore the high peaking of the mode and the asymptotic decrease of the data distribution from a class having a Gaussian distribution may not happen in practice. What may happen is that the class data distribution is rather constant around the mean or mode, possibly with a rather small peak and jump to zero rather abruptly at the boundaries of that class with other classes. This deviation from the Gaussian class model is also indicated indirectly by the poor estimate of the Bayesian optimum probability of error, shown as $\hat{P}_e$ in Table 7.6., compared to the non parametric probability of error estimates.

4. Related to 3 above, a question that may arise is why the Gaussian ML classifier works rather well for the linear transformations other than the SVD-Linear transformation. These can be seen in Tables 7.18 to 7.24. The answer to this is that from the nature of the Gaussian ML classifier (see Eq.(7.12) and (7.13)), it becomes a measure of the *weighted* distance from the class mean vectors to particular points. It does not require that the number of points per hypervolume or the density in a particular position in the space follow the Gaussian distribution. The classification rule only shows the equidistance ellipsoidal forms around the class mean vector or more precisely that the classifier model the class data scattering as ellipsoid. But this classifier does not require the density or the number of points per hypervolume in a certain position in the space to have a Gaussian distribution. The modelling of the class data scattering in the TM image as ellipsoidal or a *blob* seems more appropriate where the position of that ellipsoid is given by its class mean vector and the measure of the *size* and *orientation* of the data scattering is given by the class covariance matrix. Therefore the application of this ML classifier as the weighted distance measure from a class mean vector is still appropriate although it is initially designed for Gaussian distributed class.

5. The TM-Tasseled Cap Linear Transformation is ranked in the middle based on our observation of the probability of error estimates. Actually the differences between the probability of error estimates for this method and those of the other methods considered to have better performance is marginal and not significant. The reason why this method is not put into the same category as the KL and MDA although its performance is only marginally worse than these methods is because its design is totally different. This transformation is based on the physical properties of the six reflective bands of the TM images[14, 15, 16] and is intended to be applicable to any six reflective band TM image. Thus the design of this transformation does not have a direct relationship with our particular TM data set. The authors of the cited references claim that any TM image can be represented by a three dimensional space where its features are the *brightness, greenness,* and *wetness* features of the TM-Tasseled Cap Linear Transformation. Our experimental result extends the result of the cited reference that the three dimensional feature of the TM-Tasseled Cap Linear Transformation not only can be used for data representation but can give comparable probability of error performance with other methods which are designed for our particular TM data set. However, its performance is consistenly slightly worse than that of the other methods designed for our particular TM data set. This experimental result for the TM-Tasseled Cap Linear Transformation is important in terms of *complexity* of the design. The complexity of the design of this method is very low.

6. In discussing the best group of linear transformations, i.e. the group which includes the KL transform, MDA, First and Second Version of the Weighted MDA method, the Space Variant Linear Transformation and the KL Transform-MDA Hybrid Method, we will relate the experimental results with the argument that our data set is a *typical* TM data set, as discussed in Sec.7.3.1. The experimental results of this group of linear transformations show that the probability of error estimates of their three dimensional features are always almost equal or are not too much worse than those of the six dimensional raw data. An example of this is for the worst method in this group i.e. the Second Version of the Weighted MDA method, where the average of the class conditional probability of error estimates of the six dimensional raw data is 5.44 % whereas for the three dimensional feature space it is 6.94 % (see Table 7.21). This phenomenon to some degree is also shown by the experimental results of the TM-Tasseled Cap Linear Transformation, where the average of the class conditional probability of error estimates of six dimensional raw data is 5.44 % whereas for the three dimensional feature space it is 7.27 % (see Table 7.22). A considerable increase in the probability of error estimates when we decrease our dimensionality to two are observed in the experimental results for any of the linear transformation methods in this group and likewise of the TM-Tasseled Cap Linear Transformation method. An example of the increase is the 24.54 % of the two dimensional feature space and 6.10 % of the three dimensional feature space for the averages of the class conditional probability of error estimates for the KL transform method (see Table 7.18). This shows that the six reflective band TM data requires a three dimensional feature space regardless of the particular linear transformations chosen from this group, including the KL, MDA, the First and Second Version of the Weighted MDA methods. In some sense the TM-Tasseled Cap experimental results show the same characteristic of the six reflective bands TM images. Because of this, the Space Variant Linear Transformation and the KL Transform-MDA Hybrid methods are designed to utilize only no more than three dimensional feature space for classification of the TM image.

7. For three dimensional spaces, the KL transform represents 98.50 % of the variation of the data (see Table 7.9), and none of the remaining eigenvalues exceeds 1 %. The MDA method represents 98.95 % of the variations and none of the remaining eigenvalues exceeds 1 % (see Table 7.10). Similarly as shown in Table 7.12. and 7.13., the First Version and the Second Version of the Weighted MDA methods show more than 95.0 % variations are represented in the three dimensional feature space and none of the remaining eigenvalues exceeds 1 %. These four methods certainly represent most of the variations in their three dimensional feature space measured by percent cummulation of their three largest eigenvalues. However, for a two dimensional feature space, the remaining eigenvalues will be very large. For example, the KL transform, for a two dimensional feature space, only represents 91.33 % of the variation and one of the remaining eigenvalues is still large i.e. 7.17 % for the third eigenvalue (see Table 7.9). It is worth noting here the *rule of thumb* suggested by Merembeck and Turner[10] which says that we should select the space with variation exceeding 95 % and none of the remaining eigenvalues should exceeds 1 %. Therefore from the point of view of the utilization of the variations given by the cummulative percentage of the eigenvalues, we should use, for these four methods applied to the TM image, their three dimensional feature space.

8. The experimental results also show that the linear transformations designed for our data set, the KL, MDA, First and Second Version of the Weighted MDA, the Space Variant Linear Transformation and the KL Transform-MDA Hybrid methods but not the SVD-Linear Transformation method, are consistently better than the result of the TM-Tasseled Cap Linear Transformation. This shows that although the TM-Tasseled Cap Linear Transformation is very simple to use, a more *complex* method may allow us to do some futher adjustments appropriate for the data on hand to achieve better performance such as done in the Weighted MDA, the Space Variant Linear Transformation and the KL Transform-MDA Hybrid methods.

9. Comparisons between methods in the best group i.e. the KL Transform and MDA methods are difficult to make. The performance differences shown by the experimental results are very insignificant. This may be because the ratios between the number of classes,i.e. 22 classes, and the dimensionality is already large for the six reflective bands of TM data. The ratio are from $\frac{22}{6}$ for the raw data up to $\frac{22}{3}$ for the three dimensional feature space. This tendency to achieve similar performance especially for the MDA method was predicted in the theoretical analysis (see Sec.3.), where the MDA method may yield similar transformation to that of the KL transform method which uses the global statistics of the data, if the number of classes is too large.

10. For the two different eigen equations of the MDA methods i.e. the ones shown in Eq.(7.14) and (7.15), it was shown analytically that for the Gaussian ML classifier of Eq.(7.13), these two different methods will yield exactly the same classification results. The proof for this is given in Sec.7.3.2. This is to be emphasized because one report prefer one to the other[10] although it applies the Gaussian ML classifier as one of their implemented classifiers.

11. Now we want to compare the four variations of the MDA method, the ordinary MDA, the First and Second Version of the Weighted MDA methods and the KL Transform-MDA Hybrid method. Except for the ordinary MDA, the methods use unequal weights in computing the among class scatter and within class scatter matrices. The weights are shown in Table 7.11. for the 22 classes. Before we continue we want to explain something to prevent unexpected confusion. The weights shown are for 22 classes because the classifications in the experiments are done for 22 classes. But the probability of error estimates shown here are of the 10 group classes, where the groupings of the 22 classes are shown in Table 7.7. Fortunately, the classes whose weights are large, where the effects of these large weights are expected to reduce the class conditional probability of error estimates, are groups containing single class. From Table 7.7, those classes are class 13 which becomes group 5, class 14 which becomes group 6, class 15 which becomes group 7 and class 16 which becomes group 8. Therefore the class conditional probability of error estimates of these classes in the 10 class case are the same with the ones of the 22 class case. The class numbers mentioned in the following are of the 10 class case.

For the KL Transform-MDA Hybrid method, classes 6 and 7 have very large weights. Both classes yield better class conditional probability of error estimates (see Table 7.24), compared to those of the 3-dimensional ordinary MDA methods (see Table 7.19). They are even better than those of the 3-dimensional KL transform method. However the classes with low weights have higher class conditional probability of error estimates such that the weighted average of the class conditional probability of error estimates is still not as good as that for the KL transform, although the difference is not significant. Similar tendencies also shown by the First and Second Version of the Weighted MDA methods, as indicated by the class conditional probability of error, especially for the 3-dimensional cases in Table 7.20. and 7.21. The point we want to make is that by varying the weights for computing the among class scatter and within class scatter matrices, we can expect to reduce class the conditional probability of error estimates by assigning high weights to particular classes. Ideas about how to incorporate the class closeness measure and class apriori probability into the weights have been discussed in the discussion of the First and Second Version of the Weighted MDA and the KL Transform-MDA Hybrid methods.

12. For the Space Variant Linear Transformation experiments, there is an interesting result. When we apply the ordinary MDA in the Complementary Space (CS), see Sec.6. for detail description, for the group of classes consisting of classes Water 1 and Water 2, we get rather large classification error between these two classes. This is not expected because the number of class involve in this MDA method is only two. Therefore we suspect that the class

covariance matrices of these two classes are very different such that the within scatter matrix which basically is the average of those two class covariance matrices is not a good representation of either of the two class covariance matrices. It turns out using the likelihood ratio test for testing the equality of covariance matrices[17], these two matrices are significantly different which yields a within scatter matrix that will not represent any of these two covariance matrices properly which subsequently will not represent the class data scattering properly as well. This can affect the resulting transformation significantly because in the MDA method the class scattering is maximized where the classes are represented by the class mean vectors incorporated in the among scatter matrix, S_A, and the within class scatter is minimized where the within class scatter S_W is the average of the class covariance matrices. If the within class scatter matrix S_W does not represent all the class covariance matrices then the resulting transformation will be invalid as shown by our experiment.

Because of this large difference in the class covariance matrices in the CS in particular for the classes Water 1 and Water 2, and also because of the variations of the data in the CS occur primarily in the third KL axis, as shown by the precent eigenvalues in Table 7.9., we decide that the MDA applied to each group of close classes in CS will only depend on the class mean vectors of the classes member of that group. This means we will neglect the class variation information which are in the class covariance matrices. Therefore the coefficient for the linear combination to search for the next feature for each group will be of the eigenvector associated with the largest eigenvalues of the among class scatter matrices of each group. The resulting coefficients are shown in Table 7.17. This strategy gives better results than applying the ordinary MDA to each close class group. One interesting result worth noting is that except for the Water 1 group, the rest of the groups give very high weights to the third KL axis (see Table 7.17). This means that the search of the best next feature of each group points to the third KL axis regardless of the number of classes and the class members of the group. The class members of the close class groups are shown in Tables 7.16.a. to 7.16.d. Since the third KL axis is used primarily as the third features in the Space Variant Linear Tranformation then the performance of this method is practically the same as that uses the three dimensional KL features.

13. One thing is worth consideration from the above discussion. That discussion reveals that there are other characteristics that can cause the MDA method to fail. In the theoretical analysis discussed in Sec.3., we suggest that the number of classes affects the performance of the MDA method such that if the number of classes is too large relative to the dimensionality, then the MDA may converge to the KL transform method which is based on the global statistics of the data. However, the discussion of 12 above, shows that for small number of classes, in fact the experiments only involve two classes i.e. classes Water 1 and Water 2 in the CS, the MDA method yields poorer performance compared to that produced by the KL transform method. It turns out that the problem is caused by too large a difference of the two class covariance matrices.

The class covariance matrices should not vary too much from class to class so that the within scatter matrix, S_W, which is the average of the class covariance matrices, can represent the within class data scattering properly. If the class covariance matrices vary too much then the representation of the within class data scattering by the S_W matrix will not be valid. This can be very critical if the number of classes is very small, for example, in the two class case. However, it appears that for large number of classes, the performance of the MDA method is not affected greatly by the variations in the class covariance matrices. The experimental results show, as shown in Table 7.18. for the KL Transform and in Table 7.19. for the MDA method, that the MDA method and the KL transform method give similar results where this experiment is evaluated for the 22 class.

14. The experimentation for the Space Variant Linear Transformation method has not been complete as was intended initially, as described in Sec.6. for the full implementation and in Sec.7.3.2. point 6, for the modification in the implementation for the experiments. The full implementation is worth further study and more experiments with this method, especially for various data sets, not necessarily the ground images, may be worth pursuing. The developed software used for the current experiment is able to do the full implementation of this method. Simpler classification strategies might be found by fully implementing the method. However, one disadvantage of this method is that it cannot produce a lower dimensionality image. For example in our implementation for the experiments, the three dimensional images are different from group to group. In fact this property is the reason for adopting the name Space Variant Linear Transformation.

References

1. Godfried T. Toussaint, "Bibliography on Estimation of Misclassification," *IEEE Transactions on Information Theory*, vol. IT-20, no. 4, July 1974.
2. Tsvi Lissack and King Sun Fu, "Error Estimation in Pattern Recognition via L-Distance Between Posteriory Density Function," *IEEE Transaction on Information Theory*, vol. IT-22, no. 1, January 1976.
3. C.K. Chow, "On Optimum Recognition Error and Reject Tradeoff," *IEEE Transactions on Information Theory*, vol. IT-16, no. 1, pp. 41-46, January 1970.
4. Pierre A. deViejver and Joseph Kittler, *Pattern Recognition: A Statistical Approach,* Prentice-Hall, London, 1982.
5. John R. Jensen, M. Leonard Bryan, Steven Z. Friedman, Floyd M. Henderson, Robert K. Holz, David Lindgren, David L. Toll, Roy A. Welch, and James R. Wray, "Urban/Suburban Land Use Analysis," in *Manual of Remote Sensing Second Edition,* ed. Gene A. Thorley, vol. 2, American Society of Photogrammetry, Fall Church Virginia, 1983.
6. Robert V. Hogg and Allen T. Craig, *Introduction to Mathematical Statistics,* Macmillan Publishing Co., Inc., New York, 1978.
7. G. H. Rosenfield, K Fitzpatrick- Lins, and H. S. Ling, "Sampling for Thematic Map Accuracy Testing," *Photogrammetric Engineering & Remote Sensing*, vol. 48, no. 1, pp. 131-137, January 1982.
8. Richard O. Duda and Peter E. Hart, *Pattern Classification and Scene Analysis,* John Wiley & Sons, New York, London, Sydney, Toronto, 1973.
9. Gary E. Ford, V. Ralph Algazi, and Doreen I. Meyer, "A Noninteractive Procedure for Land Use Determination," *Remote Sensing of Environment*, vol. 13, no. 1, March 1983.
10. Benjamin F. Merembeck and Brian J. Turner, "Directed Canonical Analysis and the Performance of Classifiers Under Its Associated Linear Transformation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. GE-18, no. 2, pp. 190-196, April 1980.
11. Donald F. Morrison, *Multivariate Statistical Methods,* McGraw Hill Book Company, New York St.Louis San Fransisco, 1976.
12. *Habitat Evaluation Using Landsat Data, Final Report Contract DACWO5-88-P-0288 for Hydrological Engineering Center U.S. Army Corps of Engineers Davis, California,* V.R. Algazi and Associates, Consultants.
13. Katherine Fitzpatrick- Lins, "Comparison of Sampling Procedures and Data Analysis for Land-Use and Land-Cover Map," *Photogrammetric Engineering and Remote Sensing*, vol. 47, no. 3, March 1981.

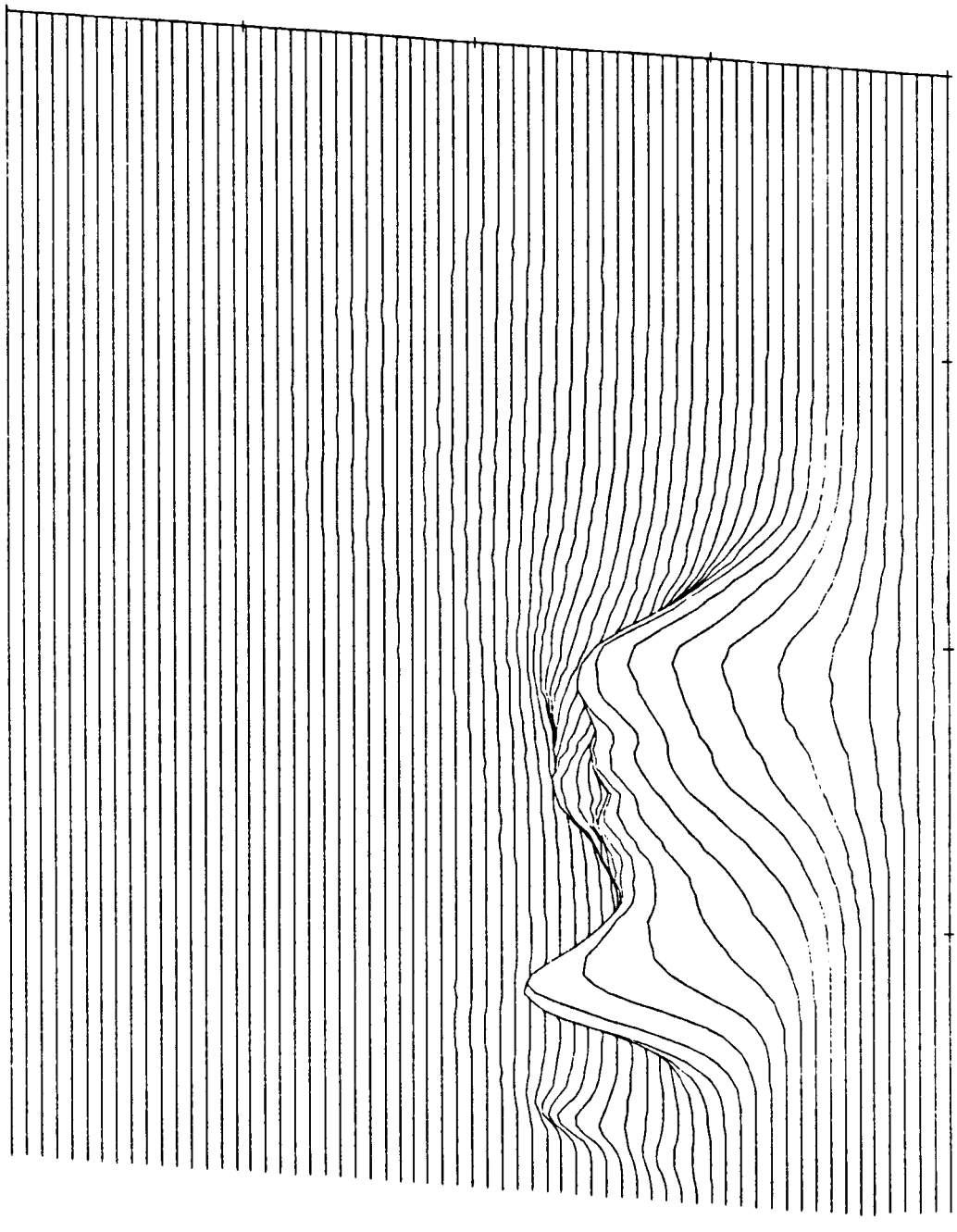

FIG.7.1. KL 1 AND KL 2 TWO-DIM. HISTOGRAM OF THE TM DATA

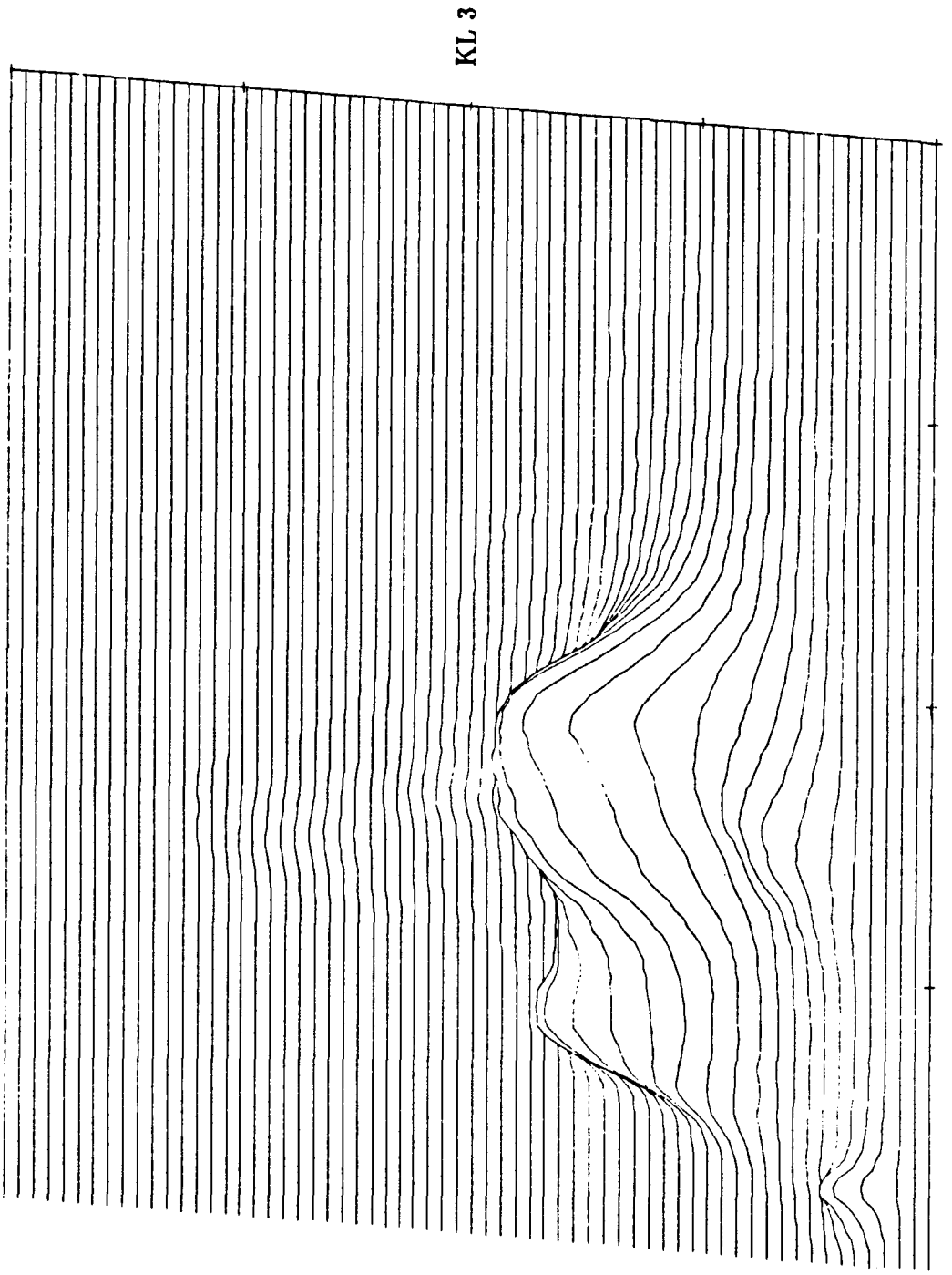

FIG.7.2. KL 1 AND KL 3 TWO-DIM. HISTOGRAM OF THE TM DATA

14. E.P. Crist and R. C. Cicone, "A Physically-Based Transformation of Thematic Mapper Data-The TM Tasseled Cap," *IEEE Transaction on Geoscience and Remote Sensing*, vol. GE-22, no. 3, May 1984.

15. Eric P. Crist and Richard C. Cicone, "Application of the Tasseled Cap Concept to Simulated Thematic Mapper Data," *Photogrammetric Engineering & Remote Sensing*, vol. L, no. 3, March 1984.

16. Eric P. Crist and Richard C. Cicone, "Comparisons of the Dimensionality and Features of the Simulated Landsat-4 MSS and the TM Data," *Remote Sensing of Environment*, vol. 14, no. 1-3, January 1984.

17. K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis*, Academic Press, London,New York,Toronto,Sydney,San Fransisco, 1979.

C-2

8. Summary and General Conclusions of Dimensionality Reduction Part

We have studied the properties of several linear transformation methods for dimensionality reduction. Theoretical derivations as well as experimental studies have been presented. Several methods have been proposed and experimental results for the existing and some of the proposed methods have been acquired. The summary and conclusions to be given in this section are those which we consider important, however more detailed conclusions have been given at the end of most of the preceeding sections.

The summary and general conclusions are as follows:

1. The analytical tractability, computational complexity, and the experimental results show that methods to find the transformation matrix A for reducing the dimensionality using the solutions of eigen equations are optimum. Moreover, these methods give an indication of the appropriate reduced dimensionality. This is done by ordering the eigenvalues and discarding the features, represented by the eigenvectors, having non significant eigenvalues.

2. A unified approach to dimensionality reduction by linear transformation is presented where each method involves optimizing an objective function with respect to the linear transformation matrix A. For the methods based on the solutions of the eigen equations, where the matrices involved in the equations are symmetric, the unified approach has a general form. It is the optimization of the objective function (see Sec.5. for details):

$$J(A) = Trace\left(A^T \tilde{S}_A A \;\; Diag\left(1/g_i ; i = 1, \ldots, d\right)\right) \tag{8.1}$$

with the constraint,

$$\left\{A^T \tilde{S}_W A\right\}_{ii} = g_i \tag{8.2}$$

where the matrices $\tilde{S}_A$, $\tilde{S}_W$ and the constants g_i are dependent on the particular method.

3. For the two different KL representations, as shown in Eq.(3.15) and (3.24), we show that under the minimum mean squared error criterion the KL transform based on the eigen equation of the covariance matrix, rather than the correlation matrix, of the data is the optimal.

4. For the two different forms of the eigen equations for the MDA methods, as shown in Eq.(7.14) and (7.15), we show that the resulting linear transformation matrices of both eigen equations will produce the same classification results if the Gaussian maximum likelihood classifier is applied.

5. For noisy data or observations subjected to additive noise, we propose two different methods (see Sec.4). We refer to them as the Minimum Mean Squared Error Criterion Based Factor Analysis and the Signal to Noise Ratio Based Dimensionality Reduction. The first method is a modified solution of the standard Factor Analysis method which consists of a sequence of two standard methods, the Wiener filter followed by the KL transform. We show that the mean squared errors of the two methods are additive which means that the errors of the steps are orthogonal or, more precisely, uncorrelated. Since the Wiener filter and KL transform minimize the mean squared errors then this proposed method will also minimize the total mean squared error.

6. For the Signal to Noise Ratio Based Dimensionality Reduction method, the unified approach discussed in point 2. can be applied where the matrix $\tilde{S}_A$ is the data covariance matrix, the matrix $\tilde{S}_W$ is the noise covariance matrix, and the constants g_i are unity. We did not perform any experiments with these two proposed methods for noisy observations.

7. In Sec.6. we propose a method refered to as the Space Variant Linear Transformation, where different transformations are applied in different regions of the feature space. This method basically applies a sequential classification method in which the next classification if required, will be done in the space designed specifically for the location of that sample in feature space. However, one *drawback* of this method is that we do not have a fixed reduced dimensionality image for all of the data. Therefore for classification, this method requires the original raw data i.e. the data with the original dimensionality. In the experiment for this method, only a partial implementation was made and the results show that it performs, in terms of probability of error estimates, almost as well as the KL transform. Full implementation of this method to test its performance with respect to reducing the computational complexity is worth pursuing in the future especially for different types of data. The developed software used in the experiments can be used for the full implementation of the method.

8. In the effort to acquire the training samples, to be used in the experimental comparisons of the linear transformation methods, we perform a method which we refer to as interactive maximum likelihood (ML) clustering on the TM Data, as described in Sec.7.2. The method classifies the data using a Gaussian maximum likelihood classifier with some operator interventions in the case of:

a. Finding the training samples of recognizable classes in the satellite images

b. Resolving class conflicts

c. Creating additional classes

These interventions are performed using a high resolution Gould IP8500 image processor and its associated software supports. The results for this clustering method in terms of the percentage area of the classes is shown in Table 7.5.

9. An attempt to quantify the performance of the interactive ML clustering method was made, as described in Sec.7.2.3. This was done by acquiring randomly 3000 training samples per class of the classified image or the map produced by the clustering method. Then, Gaussian ML classification was applied to those training samples and the resulting average of the class conditional probability of error estimates is 8.54% and the resulting weighted average of the class conditional probability of error estimates is 10.63%, as shown in Table 6.6. What actually performed in this attempt is to perform the Gaussian ML classifier, which is the classification performed in the last step of the interactive ML clustering, to the subset of the data. The subset of the data are the class training samples which were acquired randomly.

This attempt can also be considered as a way to measure the sensitivity of the class assignments to the changes in the decision boundaries. In the Gaussian ML classifier, these decision boundaries are dependent on the class parameters. The class parameters i.e. the class mean vectors and the class covariance matrices, are estimated from the design set which is a subset of the training samples acquired for the estimations. These new class parameters are expected to be different, although not greatly different, than those used in the last step of classification in the interactive ML clustering. If changes of the decision boundaries produces high probability of errors, it means that the density of points in the region close to the decision boundaries is high, which also indicates that the probability of errors in the last classification of the interactive ML clustering is high as well. This occurs because the probability of error mainly depends on the density of points in the region close to the decision boundaries. Therefore, we can quantify the probability of error occuring in the last classification of the interactive ML clustering method by applying the Gaussian ML classification to the subset of data.

More theoretical analysis of the method to quantify the performance of the preceeding classification using the training samples acquired from the classification results of that classification is needed. This method does not require *ground truths data,* which often is difficult

or expensive to acquire, but it still allow us to use the non parametric method, which does not depend on the class probability distributions, to estimate the probability of classification error.

10. We argue, as described in Sec.7.3.1., that the training samples used in the experiments represent a *typical* TM data set because a wide range of classes exist in the data set. Therefore the conclusions of the results of the linear transformation experimental comparisons are more relevant for TM data rather than for a general data set.

11. From observations of the group of methods which yield the best performance i.e. the group containing the KL, MDA, Weighted MDA, Space Variant Linear Transformation and the KL Transform-MDA Hybrid methods, the data of the six reflective TM images can be classified in a three dimensional feature space. This conclusion is based on the cumulative eigenvalue percentage and the probability of error estimates of the three dimensional feature spaces of these methods. From Table 7.18. for the KL transform, Table 7.19. for the MDA method and Tables 7.20. and 7.21. for the First and Second Versions of the Weighted MDA methods, we can see that the probability of error estimates do not change very much up to three dimensional feature spaces and then increase considerably in the two dimensional feature spaces. Therefore we conclude that for these methods the TM data also can be classified in three dimensional feature spaces.

12. The experiment for the TM-Tasseled Cap Linear Transformation, as given in Table 7.22., shows as well that the data of the six reflective TM images can be represented by the first three features of the TM-Tasseled Cap Linear Transformation. These features are (see Sec.3.3.4. and Table 7.14.) *brightness, greenness* and *wetness*. Therefore from this conclusion and the preceeding one, point 11, we can conclude that for the six reflective TM data bands, for the methods mentioned in point 11 and the TM-Tasseled Cap Linear Transformation, only three dimensional features are required to achieve probability of error comparable to that of the six dimensional raw data.

13. The high probability of error estimates produced by the SVD-Linear Transformation method, as shown is Table 7.23., but rather low probability of error estimates produced by the other methods, show that the Gaussian class distribution assumptions may not be correct. This method relies very heavily to these assumptions, as described in Sec.3.5. The class unimodality assumptions appear to be satisfied since the MDA method which relies on these assumptions, as described in Sec.3.4., performs rather well. However the applicability of the Gaussian maximum likelihood classifier to the unimodal classes is still justifiable if we consider the Gaussian ML decision rule as the minimization of the weighted Euclidian distances from the class mean vectors to a particular sample. These weighted distances only display the ellipsoidal equidistance forms from the class mean vectors or more precisely these weighted distances model the class data scattering as *ellipsoids*. The centroids of these ellipsoids are given by the class mean vectors and the relative sizes and orientations are given by the class covariance matrices. These modells do not require that the number of points per hyper volume or the densities of the data at a particular location in the feature space meet the Gaussian distributions.

14. The performances of the KL transform and the MDA methods in terms of their probability of error estimates are almost identical. The performances of the Weighted MDA methods, the KL Transform-MDA Hybrid are also very similar. Therefore the selection of the method to be applied must be based on the simplicity of the design of the linear transformation matrix A and the availability of the class training samples. The MDA method requires class parameters, and if they are not available, they must be estimated from the class training samples. If class training samples are available, the MDA method and its variations are simpler to implement than the KL transform method. The reason is that the KL transform requires the calculation of the data covariance matrix estimate. However, in term of complexity of the design of the linear transformation matrix A, the TM-Tasseled Cap Linear Transformation is the best because the

design complexity for this method is trivial. This method only applicable to the six reflective TM data only, but not to a general data set.

15. We propose and perform experiments on two modified versions of the ordinary MDA method, which are the First and the Second Versions of the Weighted MDA methods, and one modified version of the KL transform, which is the KL Transform-MDA Hybrid method, described in Sec.7.3.2. Basically these modified methods emphasize some classes more than the others by assigning different weights for different classes in the calculations of the among class scatter S_A and the within class scatter S_W matrices. In the KL Transform-MDA Hybrid method, the weighted MDA method is applied to find the third feature as a linear combination of the last four KL features i.e. the complementary space where the first two features are the first and the second KL features i.e. the preliminary space. The factors which determine the class weights are class closeness and class apriori probabilities.

The experimental results, shown in Tables 7.20.,7.21. and 7.24., show the expected result that the classes with high weights yield lower class conditional probability of error estimates than the ones of the ordinary MDA method. Moreover, the classes with high weights in the KL Transform-MDA Hybrid method yield lower class conditional probability of error estimates than the ones of the ordinary KL transform method. Thus, improvements over the ordinary MDA methods by the First and Second Version of the Weighted MDA methods and over the ordinary KL transform by the KL Transform-MDA Hybrid method for the classes with high weights have been shown as expected. However, the average or the weighted average of the class conditional probability of error estimates of those modified methods are not improved. This is due to the increase of the class conditional probability of error estimates of the classes with lower weights which overcome the decrease of the class conditional probability of error estimates of the classes with high weights.

The most important conclusion about the performance shown by the experimental results of these modified methods is that we can improve class conditional probability of error estimates of some classes by assigning high weights to them. Which class conditional probability of error estimates we want to improve are dependent on the application.

Part II

Geometric Accuracy

9. Analysis of Geometric Accuracy

The second objective of this project is to quantify the accuracy of the correction of geometric errors in TM imagery. Our approach to this task is to perform a ground control point (GCP) based bivariate polynomial coordinate transformation to rectify the TM image to a map projection, and to analyze the errors in this transformation.

To support this task, we developed a method for coordinate transformation based on the method of least squares[1], which is summarized in section 9.1. This method has been found to be numerically unstable in some situations, so the method was modified to employ orthonormalized polynomial basis functions, as described in section 9.2. In section 9.3, we apply the modified method to the geometric rectification of a Landsat 4 TM subscene.

9.1. Least-Squares Coordinate Transformation

The method of least-squares can be used to derive a bivariate polynomial coordinate transformation to rectify the geometry of satellite images. For this project, we have derived expressions for the accuracy of the geometric transformation and of the rectification of the image to a map projection as a function of the number, location, and local accuracy of the ground control points used to characterize the transformation. This work has recently been published [1]. In this section, we provide a short summary of this work.

Geometric correction can be interpreted as a least-squares coordinate transformation problem, and the known results from least-squares methods can be applied to the problem. In this approach, the geometrical distortion in the acquired image is modeled as a mapping transformation from the desired map projection coordinates to the acquired image coordinates. Denoting the map coordinates by (x_1,x_2) and the image coordinates by (y_1,y_2), the mapping function is usually chosen to be a bivariate polynomial:

$$y_j = \boldsymbol{\phi}^T(\mathbf{x})\alpha_j, \qquad j = 1,2 \tag{9.1}$$

where $\boldsymbol{\phi}(\mathbf{x})$ is a $p \times 1$ vector of polynomial functions of the map coordinate vector $\mathbf{x}$, and α_j is a $p \times 1$ vector of unknown coefficients.

The coefficients are determined from a set of n ground control points (GCPs). The $n \times 1$ vector of image GCP observations, $\mathbf{y}_j$, is assumed to be fixed but subject to measurement errors due to the limited image resolution and the resulting difficulty in locating the GCP features. These observations are assumed to be statistically independent, so the $n \times n$ covariance matrix, Σ_j, will be diagonal. The uncertainty in the corresponding map GCP locations, $\mathbf{x}_i$, $i = 1, 2, \ldots, n$, is assumed to be negligible.

The least-squares problem is to determine the estimated transformation vector, $\hat{\alpha}_j$, that minimizes the weighted sum of the squares of the residuals

$$J_j = \mathbf{r}_j^T \mathbf{W}_j \mathbf{r}_j \tag{9.2}$$

where $\mathbf{W}_j$ is the $n \times n$ weight matrix, taken to be the inverse of the image GCP covariance matrix, $\Sigma^{-1}{}_j$; and $\mathbf{r}_j$ is the $n \times 1$ vector of residuals

$$\mathbf{r}_j = \mathbf{y}_j - \hat{\mathbf{y}}_j \tag{9.3}$$

and $\hat{\mathbf{y}}_j$ is the estimated image GCP location vector.

Defining the $n \times p$ matrix of transformed observed map GCPs as $\boldsymbol{\Phi}$, where the ith row of $\boldsymbol{\Phi}$ is $\boldsymbol{\phi}^T(\mathbf{x}_i)$, the estimated image GCP location vector is

$$\hat{\mathbf{y}}_j = \boldsymbol{\Phi}\, \hat{\alpha}_j . \tag{9.4}$$

The estimated transformation coefficient vector is then given by

$$\hat{\boldsymbol{\alpha}}_j = [\boldsymbol{\Phi}^T \mathbf{W}_j \boldsymbol{\Phi}]^{-1} \boldsymbol{\Phi}^T \mathbf{W}_j \mathbf{y}_j . \tag{9.5}$$

An indication of the precision of this transformation is given by an estimate of the covariance of the of the coefficient estimate

$$\mathbf{S}_{\hat{\alpha}_j} = [\boldsymbol{\Phi}^T \mathbf{W}_j \boldsymbol{\Phi}]^{-1} \tag{9.6}$$

This covariance estimate is a function of the locations of the map GCPs through $\boldsymbol{\Phi}$ and the variances of the image GCP measurement errors through $\mathbf{W}_j$.

The precision of the transformation is indicated by an estimate of the variance of the estimated value of the image coordinate

$$s^2_{\hat{y}_j} = \boldsymbol{\phi}^T(\mathbf{x})[\boldsymbol{\Phi}^T \mathbf{W}_j \boldsymbol{\Phi}]^{-1} \boldsymbol{\phi}(\mathbf{x}). \tag{9.7}$$

This expression provides an estimate of the error variance at any point in the map space for a specific set of GCP observations.

The "goodness of fit" of the transformation can be assessed from the weighted sums of squared residual error, J_1 and J_2. These sums have a chi-squared distribution with $n - p$ degrees of freedom, and the confidence region at a significance level α is

$$J_j < \chi^2_{\alpha, n-p}$$

where $\chi^2_{\alpha,n-p}$ is the value of the chi-square distribution at significance level α and $n - p$ degrees of freedom.

The problem encountered in implementing this method as a computer algorithm is that the least-squares normal matrix, $\boldsymbol{\Phi}^T \mathbf{W}_j \boldsymbol{\Phi}$, is often unstable, leading to numerical problems in computing its inverse. The instability is caused by the large dynamic range of the element values. The problem can be solved by orthonormalizing the basis functions, which is described in the following section.

9.2. Use of Orthonormalized Basis Functions

To resolve the numerical problems associated with the least-squares coordinate transformation, a method based on orthonormalizing the basis functions has been developed[2]. In this method, the basis functions for the transformation are chosen such that the normal equation matrix is an identity. The new $p \times 1$ vector basis function $\mathbf{v}(\mathbf{x})$ can be chosen to be a linear function of the original basis functions

$$\mathbf{v}(\mathbf{x}) = \mathbf{A}\, \boldsymbol{\phi}(\mathbf{x}) \tag{9.8}$$

where $\mathbf{A}$ is a $p \times p$ matrix. The matrix of transformed observed map GCPs is then

$$\mathbf{V} = \boldsymbol{\Phi}\, \mathbf{A}^T . \tag{9.9}$$

With this new set of basis functions, the estimated image GCP location is

$$\hat{\mathbf{y}}_j = \mathbf{V}\, \hat{\boldsymbol{\beta}}_j , \tag{9.10}$$

where $\hat{\boldsymbol{\beta}}_j$ is a $p \times 1$ vector of coefficients, which, through the use of equation (9.5), is given by

$$\hat{\boldsymbol{\beta}}_j = [\mathbf{V}^T \mathbf{W}_j \mathbf{V}]^{-1} \mathbf{V}^T \mathbf{W}_j \mathbf{y}_j \tag{9.11}$$

The problem is now to choose $\mathbf{A}$ such that

$$\mathbf{V}^T \mathbf{W}_j \mathbf{V} = (\mathbf{W}_j^{1/2} \mathbf{V})^T (\mathbf{W}_j^{1/2} \mathbf{V}) = \mathbf{I} \tag{9.12}$$

This is equivalent to orthonormalizing the weighted basis functions, $\mathbf{W}_j^{1/2} \boldsymbol{\Phi}$. The orthornormalized basis functions, $\mathbf{v}(\mathbf{x})$ and the associated matrix $\mathbf{A}$ can be generated from the original basis

functions, $\phi(\mathbf{x})$, through the use of the Graham-Schmidt procedure[3]. With this choice of $\mathbf{A}$, the expression for the orthonormal transformation coefficients becomes

$$\hat{\boldsymbol{\beta}}_j = \mathbf{V}^T \mathbf{W}_j \mathbf{y}_j . \tag{9.13}$$

Substituting equation (9.9) into equation (9.10), the expression for the estimated image vector becomes

$$\hat{\mathbf{y}}_j = \boldsymbol{\Phi}\mathbf{A}^T \hat{\boldsymbol{\beta}}_j . \tag{9.14}$$

By comparing this with equation (9.4), we find the relationship between the coefficient vectors

$$\hat{\boldsymbol{\alpha}}_j = \mathbf{A}^T \hat{\boldsymbol{\beta}}_j . \tag{9.15}$$

Thus, the estimate of the coefficient vector based on the original basis functions is given by

$$\hat{\alpha}_j = \mathbf{A}^T \mathbf{V}^T \mathbf{W}_j \mathbf{y}_j = \mathbf{A}^T (\boldsymbol{\Phi}\mathbf{A}^T)^T \mathbf{W}_j \mathbf{y}_j = \mathbf{A}^T \mathbf{A} \boldsymbol{\Phi}^T \mathbf{W}_j \mathbf{y}_j . \tag{9.16}$$

There is another property of the orthonormal transformation that will prove useful. From equations (9.9) and (9.12), we have

$$[\mathbf{W}_j^{1/2} \boldsymbol{\Phi}\mathbf{A}^T]^T \mathbf{W}_j^{1/2} \boldsymbol{\Phi}\mathbf{A}^T = \mathbf{I}. \tag{9.17}$$

By algebraic manipulation of this equation, we find

$$\mathbf{A}^T \mathbf{A} = [\boldsymbol{\Phi}^T \mathbf{W}_j \boldsymbol{\Phi}]^{-1} \tag{9.18}$$

Thus, the quantity on the right hand side of the equation, which is the inverse of the normal equation matrix for the original basis functions, can be computed from the transformation matrix $\mathbf{A}$ obtained by Graham Schmidt orthonormalization.

The orthonormalized approach is also useful in evaluating the precision of the transformation. The estimate of the covariance of the original coefficient vector, from equations (9.6) and (9.18), becomes

$$\mathbf{S}_{\hat{\alpha}_j} = \mathbf{A}^T \mathbf{A} \tag{9.19}$$

The estimate of the error variance at any point in map space, from equations (9.7) and (9.18), becomes

$$s^2_{\hat{y}_j} = \boldsymbol{\phi}^T(\mathbf{x}) \mathbf{A}^T \mathbf{A} \boldsymbol{\phi}(\mathbf{x}) \tag{9.20}$$

Thus, we are able to replace the troublesome computation of the inverse of the normal matrix with the computation of the transformation matrix $\mathbf{A}$ by the Graham Schmidt procedure. We are able to use this matrix to estimate the transformation coefficients, the covariance of this estimate, and the estimate of the error covariance in terms of the original basis functions.

In summary, the algorithm for computing the transformation and evaluating its precision, using orthonormal basis functions, is the following:

- Apply the Graham-Schmidt procedure to orthonormalize the weighted original basis functions, generating the linear transformation matrix $\mathbf{A}$ of equation (9.8).
- Estimate the original coefficient vector using equation (9.16).
- Estimate the covariance of the estimate of the original coefficient vector using equation (9.19).
- Estimate the error variance of the transformation at any point in map space using equation (9.20).

9.3. Evaluation of the Geometric Accuracy of a Landsat TM Image

The orthonormal coordinate transformation algorithm was applied to a Landsat thematic mapper (TM) image to determine the accuracy with which the geometry could be rectified to a Universal Transverse Mercator (UTM) map projection, and to determine if a bilinear, or affine, transformation was adequate in producing this correction. The image analyzed was a 982 by 1024 pixel Landsat 4 TM subimage of a region including Austin, Texas, acquired January 25, 1983 (scene ID 40193-16315, row 39, path 27). This scene is characterized by a blend of urban and rural land use features with low terrain relief.

To identify GCPs, the KL transform was applied to the image (as described in section 7.3.2 of this section), and the first three transformed components were displayed in false color on an International Imaging Systems (IIS) model 70 image processing system. This image was compared with USGS topographic maps (7 1/2 minute series) to identify readily observable GCPs. Image GCP locations were determined using a trackball-controlled interactive program on the IIS system. Map GCP locations were determined using a 30 inch by 40 inch digital tablet interfaced to a graphics terminal. The varied nature of the landscape and the absence of a regular, systematic pattern of roads and transportation features complicated the identification of GCPs. Typically, GCPs included road intersections, bridges across rivers, or other natural features that were readily identifiable on the USGS maps.

The image GCP standard deviations were roughly estimated from the visual difficulty encountered in locating them on the image display. Experience has shown that easily identified features can be located with a standard deviation of 0.9 pixel. The relative difficulty in locating less obvious GCPs was estimated and the corresponding standard deviations were appropriately scaled from the base value. GCPs that could be located with medium difficulty were estimated to have standard deviations of 1.35 pixel, and GCPs presenting moderate difficulty were estimated to have standard deviations of 1.8 pixel.

The accuracy of the acquired GCPs was then assessed by applying the orthonormal coordinate transformation and checking the residual errors, $\mathbf{r}_j$, and the residuals weighted by the inverse of the GCP observation standard deviation. GCPs for which the weighted residual was greater than three were considered as "suspect," and were examined to determine if an error was made in determining its location in the image or maps. Graphics overlays on the image, showing the observed and estimated image GCP locations, were also used to determine errors. When we determined that we had acquired the location of the wrong feature on either the image or the map, the GCP locations were measured again. Using this method, we were able to locate 58 GCPs, well separated spatially on the image.

We applied the biquadratic transformation ($p = 6$), using the following mapping functions:

$$\phi(\mathbf{x}) = \begin{bmatrix} 1 \\ x_1 - \bar{x}_1 \\ x_2 - \bar{x}_2 \\ (x_1 - \bar{x}_1)^2 \\ (x_2 - \bar{x}_2)^2 \\ (x_1 - \bar{x}_1)(x_2 - \bar{x}_2) \end{bmatrix}, \tag{9.21}$$

where $\bar{(\cdot)}$ denotes a sample average over the observed GCPs. We compared this with the bilinear or affine transformation, using the first three of the six basis functions given in equation (9.21).

For the biquadratic transformation, the estimates of the coefficient vectors and their standard deviations are given in Table 9.1. The transformation of the GCPs is detailed in Table

9.2, where the map and image GCPs, image GCP standard deviation estimates, the transformed map GCPs, the error standard deviation at the GCP locations, the residual errors, and the weighted residual errors are listed. The weighted sums of square errors per degree of freedom are

$$J_1/(n-p) = 1.222$$

$$J_2/(n-p) = 0.848$$

The confidence limit per degree of freedom for α=0.05 is 1.343, so the chi-squared test is passed at the 0.05 significance level in both coordinates, indicating a good fit. However, the uncertainties of the quadratic coefficients (k=3,4,5) are of the same order of magnitude as the coefficients. This indicates that we have little confidence in the quadratic coefficients and that we would obtain better results from a bilinear or affine transformation.

Table 9.1. Transformation Vectors — Biquadratic

	Coordinate y_1		Coordinate y_2	
	Coefficients	Uncertainty	Coefficients	Uncertainty
k	$\hat{\alpha}_{1k}$	$s_{\hat{\alpha}_{1k}}$	$\hat{\alpha}_{2k}$	$s_{\hat{\alpha}_{2k}}$
1	478.693	0.250	518.060	0.267
2	34.6631	0.0183	-5.4218	0.0201
3	-5.4376	0.0149	-34.6564	0.0163
4	0.001821	0.00250	0.002062	0.00271
5	0.001832	0.00194	-0.003253	0.00210
6	0.004880	0.00208	-0.002241	0.00230

For the bilinear or affine transformation, the estimates of the coefficient vectors and their standard deviations are given in Table 9.3. The transformation of the GCPs is detailed in Table 9.4, which provides the same information given for the biquadratic transformation in Table 9.2. Note that the weighted residuals are higher than that for the biquadratic case, although they are less than 3 in every case, with the largest being 2.8. The weighted sums of squared errors per degree of freedom are

$$J_1/(n-p) = 1.288$$

$$J_2/(n-p) = 0.861$$

which are also slightly higher than for the biquadratic case. The confidence limit per degree of freedom for α=0.05 is 1.333, so again the chi-squared test is passed in both coordinates, indicating a good fit.

Table 9.3. Transformation Vectors — Bilinear

	Coordinate y_1		Coordinate y_2	
	Coefficients	Uncertainty	Coefficients	Uncertainty
k	$\hat{\alpha}_{1k}$	$s_{\hat{\alpha}_{1k}}$	$\hat{\alpha}_{2k}$	$s_{\hat{\alpha}_{2k}}$
1	478.9392	0.1360	517.9066	0.1461
2	34.6600	0.0182	-5.4181	0.0198
3	-5.4360	0.0148	-34.6586	0.0160

To assess the accuracy of the transformation, consider the values of $s_{\hat{y}_j}$, the estimated standard deviation of the transformation, given in Table 9.4. These are very low, with the largest being 0.373 for GCP number 36, located near the edge of the subimage, at line 991 and pixel 39. The total transformation uncertainty is given by

$$s = (s_{\hat{y}_1}^2 + s_{\hat{y}_2}^2)^{1/2} \tag{9.22}$$

These values range from 0.207 for GCP 43, near the spatial centroid of the GCPs, to 0.514 for GCP 36, near the edge of the subimage. Using a Gaussian assumption, this indicates, in the worst case, that the geometric accuracy is within 0.5 pixel 68 percent of the time, and is within 1 pixel 90 percent of the time.

The transformation coefficients of Table 9.3 indicate that the geometric distortion in the image is adequately modeled by translation, scaling, and rotation, since $\hat{\alpha}_{12}$ is nearly equal to $-\hat{\alpha}_{23}$, and $\hat{\alpha}_{13}$ is nearly equal to $\hat{\alpha}_{22}$. Further computation shows that the scaling factor is 35.0825 meters per pixel, and the rotation angle is 8.903 degrees with respect to the UTM map projection coordinate system.

9.4. Summary

We have developed a technique for bivariate coordinate transformation to rectify the geometry of satellite images based on the method of least-squares. To resolve the numerical problems associated with this method, we have modified it by orthonormalizing the basis functions, thus avoiding the need to invert a possibly unstable least-squares normal matrix. Expressions have been derived for the accuracy of the geometric transformation and of the rectification of the image to a map projection as a function of the number, location, and local accuracy of the ground control points used to characterize the transformation. Thus, the technique can also be used to evaluate the geometric registration of a coordinate-transformed image.

The technique was applied to a portion of a Landsat TM image. The image analyzed was a 982 by 1024 pixel Landsat 4 TM subimage of a region including Austin, Texas, acquired January 25, 1983 (scene ID 40193-16315). Using an interactive image display system to acquire image control points and a digital tablet to acquire the corresponding map control points from USGS topographic maps, 58 GCPs were acquired. Computation of a biquadratic coordinate transformation from these GCPs showed that the second degree terms were insignificant. Using a biquadratic, or affine, transformation, the total transformation uncertainty at the GCPs ranged from 0.207 to 0.514 pixels. The most significant geometric errors are translation, scaling, and rotation, with a scaling factor of 35.0825 meters per pixel and a rotation angle of 8.903 degrees with respect to the UTM map projection coordinate system.

References

1. Gary E. Ford and Claudio I. Zanelli, "Analysis and Quantification of Errors in the Geometric Correction of Satellite Images," *Photogrammetric Engineering and Remote Sensing*, vol. 51, no. 11, pp. 1725-1734, November 1985.
2. Subrahmanyam V. Vaddiparty, "Orthonormalized Basis Functions in Geometric Correction," M.S. Thesis, University of California, Davis, Davis, CA, June 1985.
3. H. Anton, *Elementary Linear Algebra,* Wiley, Canada, 1977.

Table 9.2. Transformation of GCPs — Biquadratic

GCP No.	Observed Map GCPs Locations (UTM — kilometers)		Observed Image GCPs Locations (pixels)		Observed Image GCPs Std Devs (pixels)		Estimated Image GCPs Locations (pixels)		Estimated Error Std Devs (pixels)		Residual Error (pixels)		Weighted Residual Error	
i	x_{1i}	x_{2i}	y_{1i}	y_{2i}	σ_{1i}	σ_{2i}	$\hat{y}_{1i}$	$\hat{y}_{2i}$	$s_{\hat{y}_1}$	$s_{\hat{y}_1}$	r_{1i}	r_{2i}	$\frac{r_{1i}}{\sigma_{1i}}$	$\frac{r_{2i}}{\sigma_{2i}}$
1	612.773	3369.618	32.	87.	0.9	0.9	33.33	87.97	0.51	0.53	-1.33	-0.97	1.47	1.07
2	617.384	3367.049	208	151.	1.8	1.8	207.26	151.89	0.29	0.30	0.74	-0.89	0.41	0.49
3	615.521	3371.073	121	22.	0.9	0.9	120.77	22.33	0.46	0.47	0.23	-0.33	0.25	0.36
4	616.993	3363.683	213.	271.	0.9	0.9	211.99	270.84	0.26	0.27	1.01	0.16	1.12	0.18
5	612.287	3362.600	55	334.	0.9	0.9	54.75	334.17	0.41	0.42	0.25	-0.17	0.28	0.18
6	614.455	3358.009	155	482.	1.8	1.8	154.98	481.46	0.30	0.32	0.02	0.54	0.01	0.30
7	621.316	3357.388	396	465.	1.4	1.4	396.11	465.58	0.25	0.27	-0.11	-0.58	0.08	0.43
8	622.406	3360.651	418.	348.	1.4	0.9	416.19	346.52	0.24	0.25	1.81	1.48	1.34	1.65
9	617.897	3359.217	267.	421.	1.8	1.8	267.63	420.78	0.25	0.27	-0.63	0.22	0.35	0.12
10	620.500	3363.675	335.	251.	0.9	0.9	333.66	251.95	0.23	0.24	1.34	-0.95	1.49	1.06
11	622.113	3367.230	369	120.	0.9	0.9	370.40	119.75	0.27	0.28	-1.40	0.25	1.56	0.27
12	625.055	3368.100	468	74.	0.9	1.4	467.86	73.51	0.30	0.33	0.14	0.49	0.16	0.36
13	623.024	3363.367	424	247.	0.9	1.4	422.90	248.91	0.23	0.25	1.10	-1.91	1.22	1.41
14	620.636	3370.620	301	10.	0.9	0.9	300.83	10.04	0.38	0.40	0.17	-0.04	0.19	0.04
15	636.485	3365.930	877.	87.	0.9	1.4	876.63	86.97	0.49	0.57	0.37	0.03	0.41	0.02
16	633.453	3367.606	762.	46.	0.9	0.9	762.29	45.11	0.42	0.48	-0.29	0.89	0.32	0.99
17	627.625	3366.367	566.	120.	1.4	0.9	566.44	119.75	0.28	0.31	-0.44	0.25	0.32	0.28
18	629.477	3368.868	616.	23.	0.9	1.4	617.35	22.78	0.38	0.43	-1.35	0.22	1.50	0.16
19	627.466	3363.551	577.	219.	0.9	0.9	576.07	218.41	0.23	0.26	0.93	0.59	1.03	0.66
20	631.766	3364.357	722	167.	1.4	1.4	721.09	167.10	0.29	0.33	0.91	-0.10	0.67	0.07
21	635.504	3363.257	856.	185.	1.4	1.8	856.82	185.16	0.39	0.45	-0.82	-0.16	0.61	0.09
22	633.059	3360.107	791.	309.	1.8	1.4	788.81	307.71	0.28	0.31	2.19	1.29	1.22	0.96
23	630.368	3355.877	720.	468.	0.9	1.4	718.21	468.94	0.24	0.26	1.79	-0.94	1.99	0.70
24	634.108	3355.575	850.	458.	0.9	1.4	849.58	459.29	0.31	0.35	0.42	-1.29	0.46	0.96
25	636.472	3357.554	922.	377.	1.4	1.4	921.03	377.93	0.41	0.47	0.97	-0.93	0.72	0.69
26	624.969	3356.888	526.	463.	0.9	1.4	525.47	463.10	0.25	0.26	0.53	-0.10	0.59	0.07
27	625.786	3360.700	532.	326.	0.9	1.4	533.17	326.45	0.23	0.25	-1.17	-0.45	1.30	0.33
28	610.654	3356.096	35.	573.	1.8	1.8	33.84	568.49	0.44	0.46	1.16	4.51	0.64	2.50
29	616.329	3355.999	231.	542.	0.9	1.4	230.88	540.86	0.26	0.28	0.12	1.14	0.14	0.84
30	616.823	3351.072	273.	708.	1.4	1.4	274.97	708.78	0.24	0.26	-1.97	-0.78	1.46	0.58
31	619.689	3348.759	386.	773.	0.9	1.4	386.87	773.26	0.23	0.25	-0.87	-0.26	0.97	0.19
32	621.923	3354.374	433	568	0.9	1.4	433.58	566.73	0.25	0.27	-0.58	1.27	0.64	0.94
33	619.513	3353.488	354.	611.	1.4	0.9	354.92	610.52	0.24	0.26	-0.92	0.48	0.68	0.54
34	613.773	3349.269	178.	788.	1.4	1.4	179.32	787.76	0.29	0.30	-1.32	0.24	0.98	0.18
35	614.751	3353.546	192.	637	1.4	1.4	189.69	634.41	0.28	0.29	2.31	2.59	1.71	1.92
36	611.197	3349.547	86.	794.	1.4	1.4	88.70	792.21	0.39	0.40	-2.70	1.79	2.00	1.32
37	608.891	3344.139	39.	991.	0.9	0.9	38.90	991.74	0.59	0.61	0.10	-0.74	0.11	0.83
38	612.978	3345.382	174.	926.	0.9	0.9	173.30	926.53	0.37	0.38	0.70	-0.53	0.78	0.58
39	615.836	3346.646	267.	866	0.9	0.9	265.15	867.27	0.28	0.29	1.85	-1.27	2.05	1.41
40	618.953	3343.941	389.	943.	1.4	1.4	387.86	943.91	0.32	0.33	1.14	-0.91	0.84	0.67
41	621.632	3346.973	466.	826	1.4	1.4	463.91	824.53	0.24	0.26	2.09	1.47	1.55	1.09
42	616.105	3342.871	294.	996	1.4	1.4	295.27	996.31	0.39	0.40	-1.27	-0.31	0.94	0.23
43	613.662	3356.840	133.	529.	1.8	1.8	133.90	526.28	0.32	0.33	-0.90	2.72	0.50	1.51
44	624.211	3353.073	520.	598.	0.9	0.9	519.92	599.40	0.24	0.26	0.08	-1.40	0.09	1.56
45	628.562	3354.336	664.	531.	1.4	0.9	663.91	532.11	0.23	0.25	0.09	-1.11	0.07	1.23
46	631.142	3350.063	776.	667	0.9	0.9	776.51	666.26	0.24	0.27	-0.51	0.74	0.57	0.83
47	632.772	3353.071	816.	553.	1.4	1.4	816.76	553.30	0.27	0.31	-0.76	-0.30	0.56	0.22
48	625.650	3348.370	594.	754	0.9	0.9	595.40	754.50	0.22	0.24	-1.40	-0.50	1.56	0.55
49	626.761	3350.992	619.	657	1.4	0.9	619.64	657.71	0.22	0.24	-0.64	-0.71	0.47	0.79
50	636.311	3351.418	948	591	0.9	0.9	948.43	591.58	0.42	0.49	-0.43	-0.58	0.48	0.64
51	633.646	3347.896	873.	727	0.9	1.4	875.04	727.87	0.32	0.38	-2.04	-0.87	2.26	0.65
52	629.151	3346.922	724.	787	0.9	1.4	724.58	785.73	0.23	0.26	-0.58	1.27	0.65	0.94
53	632.123	3345.170	837.	831	0.9	1.4	837.07	830.41	0.31	0.37	-0.07	0.59	0.07	0.44
54	633.015	3340.399	896.	992	0.9	1.4	893.90	990.69	0.50	0.60	2.10	1.31	2.33	0.97
55	628.695	3343.244	729.	916.	0.9	0.9	728.83	915.46	0.30	0.35	0.17	0.54	0.19	0.60
56	627.218	3340.839	691.	1007	0.9	0.9	690.84	1006.55	0.40	0.45	0.16	0.45	0.17	0.50
57	622.923	3345.153	517.	879	0.9	1.4	518.56	880.51	0.26	0.28	-1.56	-1.51	1.74	1.12
58	623.946	3342.799	567.	957	0.9	0.9	566.89	956.39	0.33	0.36	0.11	0.61	0.13	0.68

Table 9.4. Transformation of GCPs — Bilinear

GCP No.	Observed Map GCPs Locations (UTM — kilometers)		Observed Image GCPs Locations (pixels)		Observed Image GCPs Std Devs (pixels)		Estimated Image GCPs Locations (pixels)		Estimated Error Std Devs (pixels)		Residual Error (pixels)		Weighted Residual Error	
i	x_{1i}	x_{2i}	y_{1i}	y_{2i}	σ_{1i}	σ_{2i}	$\hat{y}_{1i}$	$\hat{y}_{2i}$	$s_{\hat{y}_1}$	$s_{\hat{y}_1}$	r_{1i}	r_{2i}	$\frac{r_{1i}}{\sigma_{1i}}$	$\frac{r_{2i}}{\sigma_{2i}}$
1	612.773	3369.618	32.	87.	0.9	0.9	33.79	87.82	0.32	0.34	-1.79	-0.82	1.99	0.91
2	617.384	3367.049	208.	151.	1.8	1.8	207.58	151.88	0.25	0.26	0.42	-0.88	0.24	0.49
3	615.521	3371.073	121.	22.	0.9	0.9	121.11	22.49	0.31	0.32	-0.11	-0.49	0.12	0.55
4	616.993	3363.683	213.	271.	0.9	0.9	212.33	270.65	0.22	0.23	0.67	0.35	0.75	0.38
5	612.287	3362.600	55.	334.	0.9	0.9	55.10	333.69	0.27	0.28	-0.10	0.31	0.12	0.35
6	614.455	3358.009	155.	482.	1.8	1.8	155.20	481.07	0.22	0.23	-0.20	0.93	0.11	0.52
7	621.316	3357.388	396.	465.	1.4	1.4	396.36	465.41	0.15	0.15	-0.36	-0.41	0.27	0.30
8	622.406	3360.651	418.	348.	1.4	0.9	416.42	346.42	0.16	0.17	1.58	1.58	1.17	1.76
9	617.897	3359.217	267.	421.	1.8	1.8	267.92	420.54	0.18	0.19	-0.92	0.46	0.51	0.26
10	620.500	3363.675	335.	251.	0.9	0.9	333.90	251.92	0.19	0.20	1.10	-0.92	1.22	1.02
11	622.113	3367.230	369.	120.	0.9	0.9	370.49	119.98	0.22	0.24	-1.49	0.02	1.66	0.03
12	625.055	3368.100	468.	74.	0.9	1.4	467.72	73.89	0.23	0.25	0.28	0.11	0.31	0.08
13	623.024	3363.367	424.	247.	0.9	1.4	423.06	248.93	0.18	0.19	0.94	-1.93	1.04	1.43
14	620.636	3370.620	301.	10.	0.9	0.9	300.88	10.47	0.27	0.29	0.12	-0.47	0.13	0.52
15	636.485	3365.930	877.	87.	0.9	1.4	875.68	87.15	0.31	0.35	1.32	-0.15	1.46	0.11
16	633.453	3367.606	762.	46.	0.9	0.9	761.48	45.50	0.29	0.32	0.52	0.50	0.57	0.55
17	627.625	3366.367	566.	120.	1.4	0.9	566.22	120.03	0.22	0.24	-0.22	-0.03	0.16	0.04
18	629.477	3368.868	616.	23.	0.9	1.4	616.81	23.30	0.26	0.29	-0.81	-0.30	0.90	0.22
19	**627.466**	**3363.551**	**577.**	**219.**	**0.9**	**0.9**	**576.01**	**218.50**	**0.19**	**0.21**	**0.99**	**0.50**	**1.10**	**0.56**
20	631.766	3364.357	722.	167.	1.4	1.4	720.70	167.23	0.24	0.26	1.30	-0.23	0.97	0.17
21	635.504	3363.257	856.	185.	1.4	1.8	856.21	185.14	0.28	0.31	-0.21	-0.14	0.16	0.08
22	633.059	3360.107	791.	309.	1.8	1.4	788.61	307.55	0.23	0.25	2.39	1.45	1.33	1.07
23	630.368	3355.877	720.	468.	0.9	1.4	718.33	468.72	0.18	0.20	1.67	-0.72	1.85	0.53
24	634.108	3355.575	850.	458.	0.9	1.4	849.59	458.94	0.23	0.26	0.41	-0.94	0.46	0.70
25	636.472	3357.554	922.	377.	1.4	1.4	920.79	377.54	0.27	0.30	1.21	-0.54	0.89	0.40
26	624.969	3356.888	526.	463.	0.9	1.4	525.69	462.95	0.14	0.15	0.31	0.05	0.34	0.03
27	625.786	3360.700	532.	326.	0.9	1.4	533.30	326.40	0.16	0.17	-1.30	-0.40	1.44	0.29
28	610.654	3356.096	35.	573.	1.8	1.8	33.86	567.94	0.28	0.29	1.14	5.06	0.63	2.81
29	**616.329**	**3355.999**	**231.**	**542.**	**0.9**	**1.4**	**231.07**	**540.57**	**0.20**	**0.20**	**-0.07**	**1.43**	**0.08**	**1.06**
30	616.823	3351.072	273.	708.	1.4	1.4	274.97	708.65	0.20	0.21	-1.97	-0.65	1.46	0.48
31	619.689	3348.759	386.	773.	0.9	1.4	386.88	773.29	0.19	0.20	-0.88	-0.29	0.98	0.21
32	621.923	3354.374	433.	568.	0.9	1.4	433.81	566.57	0.14	0.15	-0.81	1.43	0.90	1.06
33	619.513	3353.488	354.	611.	1.4	0.9	355.10	610.36	0.16	0.17	-1.10	0.64	0.81	0.72
34	613.773	3349.269	178.	788.	1.4	1.4	179.05	787.66	0.25	0.26	-1.05	0.34	0.78	0.25
35	614.751	3353.546	192.	637.	1.4	1.4	189.73	634.13	0.22	0.23	2.27	2.87	1.68	2.13
36	611.197	3349.547	86.	794.	1.4	1.4	88.28	792.00	0.29	0.30	-2.28	2.00	1.69	1.48
37	608.891	3344.139	39.	991.	0.9	0.9	37.74	991.92	0.35	0.37	1.26	-0.92	1.40	1.02
38	612.978	3345.382	174.	926.	0.9	0.9	172.66	926.70	0.29	0.30	1.34	-0.70	1.49	0.78
39	615.836	3346.646	267.	866.	0.9	0.9	264.83	867.40	0.24	0.25	2.17	-1.40	2.41	1.55
40	618.953	3343.941	389.	943.	1.4	1.4	387.57	944.27	0.24	0.25	1.43	-1.27	1.06	0.94
41	621.632	3346.973	466.	826.	1.4	1.4	463.94	824.66	0.19	0.21	2.06	1.34	1.53	0.99
42	616.105	3342.871	294.	996.	1.4	1.4	294.68	996.77	0.28	0.29	-0.68	-0.77	0.50	0.57
43	613.662	3356.840	133.	529.	1.8	1.8	134.06	525.86	0.23	0.24	-1.06	3.14	0.59	1.74
44	624.211	3353.073	520.	598.	0.9	0.9	520.15	599.27	0.14	0.15	-0.15	-1.27	0.17	1.41
45	628.562	3354.336	664.	531.	1.4	0.9	664.12	531.91	0.16	0.18	-0.12	-0.91	0.09	1.02
46	631.142	3350.063	776.	667.	0.9	0.9	776.77	666.02	0.20	0.23	-0.77	0.98	0.85	1.09
47	632.772	3353.071	816.	553.	1.4	1.4	816.91	552.97	0.21	0.24	-0.91	0.03	0.68	0.02
48	625.650	3348.370	594.	754.	0.9	0.9	595.60	754.49	0.17	0.19	-1.60	-0.49	1.78	0.54
49	626.761	3350.992	619.	657.	1.4	0.9	619.88	657.59	0.16	0.18	-0.88	-0.59	0.65	0.65
50	636.311	3351.418	948.	591.	0.9	0.9	948.56	591.07	0.27	0.30	-0.56	-0.07	0.62	0.08
51	633.646	3347.896	873.	727.	0.9	1.4	875.32	727.57	0.25	0.28	-2.32	-0.57	2.58	0.42
52	629.151	3346.922	724.	787.	0.9	1.4	724.84	785.68	0.21	0.23	-0.84	1.32	0.93	0.98
53	632.123	3345.170	837.	831.	0.9	1.4	837.37	830.30	0.25	0.28	-0.37	0.70	0.41	0.52
54	633.015	3340.399	896.	992.	0.9	1.4	894.21	990.83	0.30	0.34	1.79	1.17	1.98	0.86
55	628.695	3343.244	729.	916.	0.9	0.9	729.03	915.65	0.24	0.27	-0.03	0.35	0.03	0.39
56	627.218	3340.839	691.	1007.	0.9	0.9	690.90	1006.99	0.26	0.29	0.10	0.01	0.11	0.02
57	622.923	3345.153	517.	879.	0.9	1.4	518.57	880.74	0.21	0.22	-1.57	-1.74	1.74	1.29
58	623.946	3342.799	567.	957.	0.9	0.9	566.84	956.78	0.23	0.25	0.16	0.22	0.17	0.25