RSD-TR-17-85

OBJECTIVE DIMENSIONALITY REDUCTION USING OUT-OF-CLASS COVARIANCE[1]

Paul Besl

Kurt Skifstad

Ramesh Jain

Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, Michigan 48109

November 1985

CENTER FOR RESEARCH ON INTEGRATED MANUFACTURING

Robot Systems Division

COLLEGE OF ENGINEERING

THE UNIVERSITY OF MICHIGAN

ANN ARBOR, MICHIGAN 48109-1109

[1] This work was supported by IBM Corp., Data Systems Div., Kingston, NY under the monitorship of Dr. Jack Contino.

ensa

UMR0338

TABLE OF CONTENTS

Abstract

Non-hierarchical statistical decision algorithms spend a significant portion of their time entertaining incorrect hypotheses in multiple class, pattern recognition problems. Maximum-likelihood multivariate-Gaussian (MLMVG) hypothesis testing is a common example of such a statistical pattern recognition technique. It is shown that the use of "out-of-class" covariance matrices can significantly reduce the run-time computations required to make MLMVG decisions. The analysis directly leads to an objective dimensionality reduction (ODR) technique that indicates the preferred, intrinsic dimensionality of multiple class decision spaces given the training data. Run-time computations are reduced even further using these reduced dimension class decision spaces with no measurable loss in decision accuracy. This method is then compared to a popular subjective dimensionality reduction technique to stress the essential concepts of out-of-class covariance. The theory has been applied to a nine (9) class, twenty-seven (27) feature, automatic visual solder joint inspection problem with excellent results; run-time computations are reduced by more than a factor of three while maintaining excellent decision performance.

Index Terms: Maximum-likelihood hypothesis testing, multivariate Gaussian assumption, dimensionality reduction, out-of-class covariance matrix, simultaneous diagonalization, automatic visual inspection.

1. INTRODUCTION

There are numerous industrial processes that benefit from automation technology. In many applications, automated decision-making capability is required as an integral task in a particular operation. For example, a machine vision system may be required to inspect the integrity of automatically manufactured parts, route bad parts off the assembly line, and correct process parameters responsible for the defects in those parts. A vast variety of statistical pattern recognition techniques may be applicable when the decision problem can be posed as a multiple class, multiple feature hypothesis testing problem. The designer of an automated decision-making system must evaluate the available alternatives offered by pattern recognition techniques: parametric vs. non-parametric, hierarchical vs. non-hierarchical, normal vs. non-normal, linear vs. non-linear, supervised vs. unsupervised, optimal (Bayesian) vs. sub-optimal. Among many other factors, these choices depend on the type of input data being used, the amount of training data available, the type of the training process allowed, and the amount of *a priori* knowledge one has about the input data.

Statistical pattern recognition techniques make assumptions about the nature of the input data, and these assumptions influence the decision-making process. For example, zero-mean assumptions for features can easily be handled at run time with a negligible decrease in efficiency. Other assumptions, however, such as those regarding the amount of correlation between features or the relative contribution of each feature to the decision-making process are generally more difficult to handle, and are thus ignored in some schemes for the sake of simplicity at the cost of reduced decision accuracy.

Methods such as the k-nearest neighbor algorithm and the minimum distance (to class means) classifier use a straightforward feature-space distance metric for classification purposes. Although the performance of these algorithms may be quite good with feature data that is uncorrelated and highly relevant to the decision process, they do not incorporate ways of automatically compensating for highly correlated features or identifying features that may actually contribute little or no information to the decision process.

For many pattern recognition problems (such as automatic solder joint inspection using gray level images), the boundary between meaningful feature data and useless feature data can be quite hazy. Feature selection is limited by the abilities of the decision system designer or programmer. A small number of potentially useful features may be present in a larger set of highly correlated features requiring several higher dimensions with little gain in information relevant to the decision process. Unfortunately, as mentioned above, the simpler statistical pattern recognition techniques prove to be severely limited when confronted with this type of data. In order to perform well under these circumstances, an algorithm must provide two capabilities: (1) it must compensate for feature data that is correlated by computing new decorrelated features, and (2) it must discriminate between uninformative and meaningful features by automatically weighting them according to their importance in the decision making process. The first problem is easily solved by using the principal components (or discrete Karhunen-Loeve) transformation (eigenvalue-eigenvector decomposition) of the appropriate covariance matrix to provide uncorrelated feature data. The second problem has long been recognized as a crucial one in statistical pattern recognition research. Many subjective techniques for dimensionality reduction have been developed that allow one to obtain the best features from a given set with respect to some criterion. However, objective feature evaluation methods that state which features in a given list are informative and which features are misleading or irrelevant are rarely realized in practice.

The Objective Dimensionality Reduction algorithm is a non-hierarchical (single-stage), parametric, maximum-likelihood, multiclass decision algorithm based on the multivariate-Gaussian assumption for feature data. It is computationally efficient, and it handles the two problems mentioned above quite effectively in our automatic visual solder joint inspection application. But despite these strengths and other benefits described subsequently, it is also subject to the limitations of this type of approach. For example, all input data features must be computed in most cases, and it is not possible to trade off decision accuracy for computation costs. These factors may be a hindrance for certain applications in which case one may want to consider hierarchical (multi-stage), non-parametric, decision tree classifiers [Kulkarni and Kanal 1978], which can directly trade classification accuracy for costs and compute only the input data features as needed for particular decisions in the decision tree. Nevertheless, we believe that the out-of-class covariance concept and its dimensionality reduction implications provide important insights for statistical pattern recognition research.

2. QUALITATIVE COMMENTS

The Objective Dimensionality Reduction (ODR) technique provides a unique, new "twist" (in the form of a new rotation matrix) to the common maximum-likelihood, multivariate-Gaussian (MLMVG) pattern recognition technique by considering what we call "out-of-class" covariance matrices. Decisions are based on the minimization of a quadratic distance metric as in the MLMVG case. However, instead of the usual inverse in-class covariance matrix, the ODR method uses a variable-size, row-shuffled transformation matrix for each class. This matrix is computed using both the in-class and out-of-class covariance matrices. Although not employed by standard pattern recognition techniques, the out-of-class covariance matrix concept is intuitively appealing because every M class decision can be considered as M binary class decisions where only in-class and not-in-class hypotheses are entertained. This idea is found in Dye [1974] and Friedman [1977].

The ODR method allows a significant reduction in run-time computation over standard MLMVG techniques, and allows an objective evaluation of the decision-making relevance of individual transform features in the transformed feature vector space for each class. Each transformed vector space is simply a rotated and scaled version of the original feature vector space where the transform features are uncorrelated with respect to both in-class and out-of-class conditions.

As with the MLMVG approach, if *a priori* probabilities of the different classes are also known, this maximum likelihood technique can be easily generalized to provide minimum-error decisions. If the costs of false alarms and misses are also known, Bayes risk can be minimized.

The maximum-likelihood decision rule states that the hypothesis that an observed signal is of the class C_i is correct if the probability of that signal belonging to class C_i is greater than the probability of the signal belonging to any particular one of the other possible classes within the context of the decision problem. That is, given an unclassified signal, the conditional probability density function is evaluated for each class and the class with the largest conditional probability density is chosen as correct. A vector $\vec{x}$ composed of feature data extracted from the unclassified signal can often be assumed to be a Gaussian random vector with each feature vector component being normally distributed and all features being jointly Gaussian in nature. This assumption is usually justified under the authority of the central limit theorem, which states that if enough random variables from any given distribution are averaged, a random variable will be obtained that tends to be normally distributed (i.e., Gaussian). Given this approximate Gaussian nature of the input data, one can infer that, since the statistics of Gaussian random vectors are completely determined by their mean vectors and covariance matrices, the feature vector statistics obtained are also mostly determined by their mean vectors and covariance matrices. Following this (tenuous) line of logic, one can implement a maximum-likelihood decision algorithm if reasonable estimates can be obtained for the mean vector and covariance matrix for each class. It is then assumed that the sample mean and sample covariance matrices computed from selectively chosen, representative training samples of each class will provide the reasonable estimates necessary for the maximum likelihood decision rule under the Gaussian random vector assumption.

If the full dimensionality N of the original feature vector space is maintained, N + 1 vector inner products (of length N) must be evaluated to estimate the conditional probability density for each class in the standard multivariate-Gaussian maximum-likelihood technique. The uniqueness of the ODR algorithm lies in the computational efficiency of its decision making process by using information provided by out-of-class covariance matrices. By following the approach originated by Dye [1974], ***soft (or adaptive) dimensionality reduction*** or ***hard (or fixed) dimensionality reduction*** can be achieved. The order of the transform features considered in the decision process depend on the given class hypothesis and the training data. The number of transform features evaluated depends on these factors and the unknown being classified. Class hypotheses can often be dismissed after evaluating only one or two vector inner products. Consider a 10 class, 25 feature solder joint inspection problem. Straightforward MLMVG evaluation requires 260 vector inner products (of length 25) to reach a maximum likelihood decision. A typical ODR algorithm decision with soft dimensionality reduction might only require 60 inner products and can actually require as few as 35 inner products with no change in maximum-likelihood decision accuracy. Hard dimensionality reduction decisions generally need even fewer inner products. Computational short-cuts like this can be extremely important for real-time inspection applications using many features and many classes. The ODR method also provides other very interesting feature ranking capabilities, which are discussed in more detail later.

Covariance matrices conditioned by the in-class hypothesis are used in most techniques where uncorrelated feature data is analyzed. Dye [1974] has noted that in multiple class problems, decision algorithms spend most of their time entertaining incorrect hypotheses. This extremely important observation has dramatic consequences when properly analyzed. As shall be shown, it is advantageous to examine covariance matrices conditioned by the out-of-class hypothesis and to also uncorrelate feature data using out-of-class covariance matrices. The generalized eigenvector problem for two positive definite real symmetric matrices is solved using simultaneous diagonalization. This solution process can be applied to the out-of-class and in-class covariance matrices providing simultaneously uncorrelated features.

The motivation for decorrelating the feature data can be phrased as follows. Multivariable problems are much easier to work with when uncorrelated features are used because each uncorrelated feature can be treated as a separate, one-dimensional entity instead of just another element of some interrelated vector quantity. When features are uncorrelated with respect to both the in-class and out-of-class hypotheses, each uncorrelated feature represents a separate one-dimensional decision space. It is almost always easier to analyze N separate, easy 1-D problems than to analyze one complicated, interrelated N-dimensional problem. Mathematical diagonalization processes provide this decoupling. In addition, the multivariate Gaussian assumption implies that uncorrelated features are actually (class-conditionally) statistically independent.

The so-called "whitening" transformation [Fukunaga 1972] is commonly used to rotate and scale unknown vectors so that, when entertaining the correct hypothesis for a given class, the components of the resultant vector are uncorrelated, zero mean, unit variance. By applying the same whitening transformation to the out-of-class covariance matrices, new matrices are produced that correspond to the out-of-class covariance matrices in the new whitened vector space. Special eigenvalues can then be obtained from these new matrices through the use of standard diagonalization procedures.

Soft and hard dimensionality reduction is based on these special eigenvalues produced by the diagonalization of the whitened out-of-class covariance matrices. As will be proven in the next section, the expected squared value of a new uncorrelated feature (denoted here as a z feature) is exactly one (1.00) when the correct class is hypothesized and is equal to the corresponding special eigenvalue (denoted here as a g eigenvalue) when *any incorrect class* is hypothesized. And of course, the maximum-likelihood decision rule is equivalent to a minimum (Mahalanobis) distance rule. Eigenvalues greater than one indicate a dimension in the z feature space in which that particular class is easily distinguishable from other classes. Eigenvalues less than one indicate dimensions in the z feature space where it is very difficult to distinguish the given class from others. Given that the original features are sufficiently Gaussian in nature, the contribution of feature data in the direction of the feature space corresponding to a small eigenvalue is actually detrimental to the classification procedure based on the minimum distance metric. We refer to the process of ignoring these misleading z values as hard dimensionality reduction.

We are not aware of any other pattern recognition method that can give such an explicit, objective, class-specific ranking of features as the ODR method provides. Recent reviews, such as [Swain 1985], indicate others are not aware of any such techniques either. ODR analysis indicates the preferred dimensionality of the decision space for each class as dictated by the training data. Because of this capability, it is reported [Dye 1974] that performance of the ODR technique does not decline as the number of features increases (the so-called "peaking phenomenon") as other similar methods do owing to finite training sample size [Jain and Waller 1978]. We restrict our analysis to the dimensionality reduction properties of the ODR approach.

The soft dimensionality reduction is obtained as follows. The evaluation order in the computational loop corresponding to the calculation of the classification metric for the decision process can be "shuffled" according to these g eigenvalues so that the features corresponding to largest eigenvalues are considered first. The decision process then may conditionally jump out of the computational loop when the accumulating metric exceeds the previous minimum value. Thus, after considering but a few of the feature values for a given class, a decision can be made that the correct hypothesis is not being entertained, allowing the next hypothesis to be considered. In the solder joint inspection application discussed in the last section, it has been found that on average only about eight (7.82) z features need to be computed using hard dimensionality reduction and about ten (10.16) z features using soft dimensionality reduction even though 27 image features are being calculated for each image.

Figure 1 displays an example image of plated-through-hole solder joints on a printed circuit board and a class-labeled version of that same image. Each solder joint in a set of these images has been classified by a human operator and by the ODR classification technique using an interesting set of gray-level image features. The class labels are discussed in Section 5. The distribution of the number of z features computed by the ODR algorithm (i.e., the number inner product computations) while making classification decisions is shown in Figure 2. The standard MLMVG approach requires a full inner product computation for each hypothesis (27 inner products). The emulated MLMVG algorithm (based on the matrix square root) using the test against the smallest metric computed so far yields the top plot, which involved an average of about thirteen (12.50) inner products. Note that just the insertion of such a test reduced computations by 53.7 percent for our solder joint inspection data. The MLMVG algorithm with shuffling (soft dimensionality reduction) was used to obtain the next plot. Notice the shift towards fewer z computations. In fact, 41 percent of all hypotheses were dismissed after the evaluation of a *single* transformed z feature (one inner product). Total computations were reduced by another 18.7

percent from the emulated MLMVG case. The ODR algorithm with hard dimensionality reduction is shown in the third plot. Note the break up of the peak at the full number of features in this case. Total computations were reduced by another 23.0 percent from the soft dimensionality reduction. Despite this reduction in the average number of inner products computed per hypothesis from 27 to 7.82 (71 percent reduction), there is no real loss in decision performance as shown in Figures 3 and 4. The last plot demonstrates the equivalent behavior of the minimum distance classifier. Its results are shown in Figure 5. These results tables are explained in detail in the last section. We emphasize that the performance of the straightforward maximum likelihood approach can still be obtained even though the necessary computations have been reduced by a significant factor from the straightforward implementation. These concepts are discussed in detail in the next section.

3. QUANTITATIVE ANALYSIS

Now that the basic ideas of the ODR method have been qualitatively introduced, the entire method can be discussed in complete detail. The first step in the ODR technique involves the computation of estimates of an in-class correlation matrix (denoted $\mathbf{A}_i$) and an in-class mean vector (denoted $\vec{\mu}_i$) for each of N_C classes using N_F features from N_T training samples. The correlation matrix is calculated by summing the outer products of all feature vectors of class C_i from the training set and dividing by the total number L_i of training samples for that class. The in-class mean vector is simply formed by summing all feature vectors of each class together and dividing by the number of training samples. The total number of training samples N_T is just the sum of the L_i's. We do not assume that the same number of samples are available from each class.

$$\mathbf{A}_i = \mathrm{E}\left\{ \vec{x}\,\vec{x}^T \mid \vec{x} \in C_i \right\} = \frac{1}{L_i} \sum_{j=1}^{L_i} \left[\vec{x}_j \vec{x}_j^T \right] \qquad (\vec{x}_j \in C_i) \qquad (1)$$

$$\vec{\mu}_i = \frac{1}{L_i} \sum_{j=1}^{L_i} \vec{x}_j \qquad (\vec{x}_j \in C_i) \qquad (2)$$

where $\vec{x}$ is a feature vector from the training set containing a list of numbers computed from the observed signal. Brackets are added at times to emphasize matrix quantities.

It should be noted that the notation $\vec{x}_j \in C_i$ is not completely rigorous, as $\vec{x}_j$ is merely a random feature vector extracted from a random signal caused by a random event e_j. Therefore, $\vec{x}_j$ does not really belong to a class in the sense of a partition of the probability space. We should say that the random event e_j belongs to class C_i and $\vec{x}_j$ is a function of the random event. However since there is a direct correspondence because $\vec{x}_j$ is formed by extracting features from the signal caused by e_j, there is no danger of ambiguity. The previous notation has been adopted for convenience and will be used throughout.

The next step in the algorithm is the calculation of an in-class covariance matrix for each class. This is done by subtracting the outer product of the in-class mean vector (μ_i) from the correlation matrix.

$$\mathbf{B}_i = \mathrm{E}\left\{(\vec{x} - \vec{\mu}_i)(\vec{x} - \vec{\mu}_i)^T \mid \vec{x} \in C_i\right\} = \mathbf{A}_i - \left[\vec{\mu}_i \vec{\mu}_i^{\,T}\right] \tag{3}$$

In the case that the data is zero mean, the covariance and correlation matrices are equivalent. The diagonal elements of each **B** matrix are the variances corresponding to each feature for that particular class. The off-diagonal elements represent the covariances of the various features. High absolute values of off-diagonal elements indicate highly correlated features; low absolute values indicate less correlated features. Features are uncorrelated when their covariance is zero.

Given the **B** matrices, and making the assumption that the data is roughly Gaussian in nature, the probability that a given sample vector $\vec{x}$ will occur given that the observed signal belongs to the class C_i is given by the multivariate Gaussian (MVG) density function:

$$p_i(\vec{x}) = \frac{1}{(2\pi)^{\frac{N}{2}} \left(\det \mathbf{B}_i\right)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu}_i)^T \mathbf{B}_i^{-1}(\vec{x} - \vec{\mu}_i)\right) \tag{4}$$

The maximum likelihood decision rule can then be expressed as follows:

$$\textit{Decide}\ \ \vec{x} \in C_i \ \text{ if } p_i(\vec{x}) > p_m(\vec{x}) \ \textit{ for all } m \neq$$

Since $(2\pi)^{\frac{N}{2}}$ is a constant and $\exp(x)$ is monotonic, computational effort can be reduced by basing the decision process on the minimization of the statistic S_i given in equation (5), as opposed to the maximization of p_i in equation (4):

$$S_i(\vec{x}) = K_i + (\vec{x} - \vec{\mu}_i)^T \mathbf{B}_i^{-1} (\vec{x} - \vec{\mu}_i) \tag{5}$$

where

$$K_i = ln(det \mathbf{B}_i) \tag{5a}$$

A different, but equivalent decision rule is thus formulated:

$$Decide \;\; \vec{x} \in C_i \text{ if } S_i(\vec{x}) < S_m(\vec{x}) \;\; for \; all \; m \neq \tag{6}$$

One can implement this approach as stated thus far to obtain a maximum-likelihood classifier for multivariate Gaussian feature data. This is the straightforward approach requiring $N_F + 1$ vector inner products of length N_F for each of the N_C class hypotheses. This requires approximately $N_C(N_F^2 + N_F)$ multiplications and additions to classify each unknown feature vector.

Dye [1974] has noted that it is of significant interest to calculate an "out-of-class covariance matrix" **F** for each class. That is, compute the expected value of the outer product matrix given that the observation vector $\vec{x}$ is not in the given class. This is motivated by the fact the multiple class decision algorithms spend far more time entertaining false hypotheses than correct ones.

$$\mathbf{F} = \mathrm{E}\left\{ (\vec{x} - \vec{\mu}_i)(\vec{x} - \vec{\mu}_i)^T \mid \vec{x} \notin C \right\} \tag{7}$$

$$= \mathrm{E}\left\{\vec{x}\,\vec{x}^T \mid \vec{x} \notin C_i\right\} - \vec{\mu}_i\,\mathrm{E}\left\{\vec{x} \mid \vec{x} \notin C_i\right\} - \mathrm{E}\left\{\vec{x} \mid \vec{x} \notin C_i\right\}\vec{\mu}_i^{\,T} + \vec{\mu}_i\vec{\mu}_i^{\,T}$$

$$= \left[\frac{1}{(N_T - L_i)} \sum_{x \notin C_i} \vec{x}\,\vec{x}^T\right] - \vec{\mu}_i\vec{\nu}_i^{\,T} - \vec{\nu}_i\vec{\mu}_i^{\,T} + \vec{\mu}_i\vec{\mu}_i^{\,T}$$

where N_T is the total number of feature vectors from the training set. The **F** matrices can be calculated from information previously obtained plus the calculation of an "out-of-class mean vector $\vec{\nu}_i$. The out-of-class mean vector is given by

$$\vec{\nu} = \mathrm{E}\left\{\vec{x} \mid \vec{x} \notin C_i\right\} = \frac{1}{(N_T - L_i)} \sum_{\substack{m=1 \\ m \neq i}}^{N_C} L_m \vec{\mu}_m \qquad (8)$$

The out-of-class correlation matrix is determined from

$$\sum_{\notin C_i} \left[\vec{x}\,\vec{x}^T\right] = \sum_{\substack{m=1 \\ m \neq i}}^{N_C} \sum_{x \in C_m} \left[\vec{x}\,\vec{x}^T\right] = \sum_{\substack{m=1 \\ m \neq i}}^{N_C} \sum_{j=1}^{L_m} \left[\vec{x}_j\vec{x}_j^{\,T}\right] = \sum_{\substack{m=1 \\ m \neq i}}^{N_C} L_m \mathbf{A}_m \cdot \qquad (9)$$

This yields the final expression for the out-of-class covariance matrix:

$$\mathbf{F} = \left[\frac{1}{(N_T - L_i)} \sum_{\substack{m=1 \\ m \neq i}}^{N_C} L_m \mathbf{A}_m\right] - \vec{\nu}_i\vec{\mu}_i^{\,T} - \vec{\mu}_i\vec{\nu}_i^{\,T} + \vec{\mu}_i\vec{\mu}_i^{\,T} \qquad (10)$$

There are other methods for computing the out-of-class covariance matrices. A different method is described later. A method similar to this one is presented in [Dye 1974].

The definitions above directly imply that the **B** matrices are positive definite when no pair of features is completely correlated and that both the **B** and **F** matrices are symmetric. To find a vector space where features are uncorrelated with respect to both in-class and out-of-class hypotheses, it is necessary to perform a simultaneous diagonalization operation on the **B**'s and **F**'s. This is equivalent to solving the generalized eigenvalue problem:

$$\left[\mathbf{F}\right]\vec{x} = \lambda \left[\mathbf{B}\right]\vec{x}$$

The resulting generalized eigenvectors specify directions in the feature vector space that are, in a sense, "equally natural" for both in-class and out-of-class hypotheses. One is free in the general problem to whiten with respect to either **F** or **B** if both are positive definite, but it is customary to whiten with respect to the in-class covariance matrix in this type of problem.

The process of simultaneous diagonalization begins with the solution of the simple eigenvalue problem for the **B** matrices:

$$\mathbf{B} = \mathbf{U}\ \mathbf{b}\ \mathbf{U}^{\mathbf{T}} \qquad (11)$$

where the **U** matrices are the orthogonal eigenvector matrices and the **b** matrices are diagonal with the diagonal elements corresponding to the eigenvalues of the **B** matrices. No reordering of the eigenvalues in the **b** matrices is assumed. These matrices are then used to transform a feature vector $\vec{x}$ into a "whitened" zero mean and unit variance vector $\vec{y}$ whose components are uncorrelated with respect to the in-class hypothesis. We explicitly state the transformation, the mean vector, and the covariance matrix:

$$\vec{y}_i = \mathbf{b}^{-\frac{1}{2}} \mathbf{U}^T (\vec{x} - \vec{u}_i) \tag{12}$$

$$E\left\{\vec{y}_i \mid \vec{x} \in C_i\right\} = \mathbf{b}_i^{-\frac{1}{2}} \mathbf{U}_i^T \left[E\left\{\vec{x} \mid \vec{x} \in C_i\right\} \quad \vec{\mu} \right] = 0 \tag{13}$$

$$E\left\{\vec{y}_i \vec{y}_i^T \mid \vec{x} \in C_i\right\} = E\left\{\mathbf{b}_i^{-\frac{1}{2}} \mathbf{U}_i^T (\vec{x} - \vec{\mu}_i)(\vec{x} - \vec{\mu}_i)^T \mathbf{U}_i \mathbf{b}_i^{-\frac{1}{2}}\right\} \tag{14}$$

$$= \mathbf{b}^{-\frac{1}{2}} \mathbf{U}^T \mathbf{B}_i \mathbf{U}_i \mathbf{b}_i^{-\frac{1}{2}}$$

$$= \mathbf{b}_i^{-\frac{1}{2}} \mathbf{b}_i \mathbf{b}_i^{-\frac{1}{2}} = \mathbf{I} = \textit{Identity Matrix} \; .$$

This process is standard practice and often is referred to as whitening of the feature data [Fukunaga 1972].

The same transformation is now applied to the **F** matrices to produce a set of **G** matrices, which correspond to the out-of-class covariance matrices in the whitened $\vec{y}$ vector space:

$$\mathbf{G}_i = \mathbf{b}_i^{-\frac{1}{2}} \, \mathbf{U}_i^T \mathbf{F}_i \, \mathbf{U}_i \, \mathbf{b}_i^{-\frac{1}{2}} \; . \tag{15}$$

These **G** matrices are symmetric, but not generally diagonal. Therefore, the components of the $\vec{y}$ vectors are generally correlated with respect to the out-of-class hypothesis. The **G** matrices can however be diagonalized to produce eigenvector matrices (**V**'s) and eigenvalue matrices (**g**'s):

$$\mathbf{G} = \mathbf{V}\ \mathbf{g}\ \mathbf{V}^{\mathbf{T}} \tag{16}$$

Starting with **B** and **F**, we have solved for **b**, **U**, **g**, **V**. Computation of these quantities is the key part of the simultaneous diagonalization process.

Once the simultaneous diagonalization is completed, a transform matrix is obtained for each class to rotate and scale a feature vector observation so that it is zero mean and of unit variance conditioned by the in-class hypothesis, and its components are uncorrelated with respect to *both* the in-class and out-of-class hypotheses. Below are listed the expressions for the new $\vec{z}$ feature vector, the new transform matrices **W** and the conditional mean vectors and covariance matrices.

$$\vec{z} = \mathbf{V}^{\mathbf{T}}\,\mathbf{b}^{-\frac{}{2}}\,\mathbf{U}^{\mathbf{T}}(\vec{x} - \vec{\mu}\ ) = \mathbf{W}^{\mathbf{T}}(x - \mu\ ) = \mathbf{V}^{\mathbf{T}} \tag{17}$$

$$\mathbf{W} = \mathbf{U}\ \mathbf{b}^{-\frac{1}{2}}\mathbf{V} \tag{18}$$

$$\mathrm{E}\left\{\vec{z}\ \mid \vec{x} \in C\ \right\} = 0 \tag{19}$$

$$\mathrm{E}\left\{\vec{z}\,\vec{z}^{T} \mid \vec{x} \in C\ \right\} = \mathbf{V}^{\mathbf{T}}\,\mathbf{b}^{-\frac{1}{2}}\mathbf{U}^{\mathbf{T}}\,\mathrm{E}\left\{(\vec{x} - \vec{\mu}\ )(\vec{x} - \vec{u}\ )^{T}\right\}\mathbf{U}\ \mathbf{b}^{-\frac{1}{2}}\mathbf{V}$$

$$= \mathbf{W}^{\mathbf{T}} \mathbf{B} \ \mathbf{W} \ = \mathbf{I} = \textit{Identity Matrix} \tag{20}$$

One would naturally expect that a pure rotation applied to a whitened vector space would yield another whitened vector space. Similarly, applying the out-of-class condition yields

$$\mathrm{E}\left\{ \vec{z}_i \vec{z}_i^{\,T} \mid \vec{x} \notin C_i \right\} = \mathbf{g}_i = \mathbf{W}^{\mathbf{T}}\mathbf{F} \ \mathbf{W}_i \tag{21}$$

$$\mathrm{E}\left\{ \vec{z}_i \mid \vec{x} \notin C_i \right\} = \mathbf{W_i}^{\mathbf{T}}(\vec{\nu}_i - \vec{\mu}_i) \neq \vec{0} \quad (\textit{in general}).$$

Since $\mathbf{g}_i$ and **I** are both diagonal, we see that **W** provides simultaneous diagonalization of **B** and **F**

Let us examine this $\vec{z}$ vector space more closely. Let z_{ik} be the *k-th* component of the $\vec{z}$ vector when the *i-th* class hypothesis is being considered. Let g_{ik} be the *k-th* eigenvalue of the matrix $\mathbf{g}_i$ corresponding to the transform feature z_{ik}. It can be expected that, on average, if $\vec{x}$ represents an observed signal from the class C_i, then $z_{ik}^2 = 1$; that is,

$$\mathrm{E}\left\{ z_{ik}^2 \mid \vec{x} \in C_i \right\} = 1 \qquad \text{for } \textit{all } k$$

which means that

$$\mathrm{E}\left\{ z_i^T z_i \mid \vec{x} \in C_i \right\} = N_F = tr(\mathbf{I}) \, ,$$

where *tr* ( ) is the matrix trace operator. It can be expected that, on average, if $\vec{x}$ represents an observed signal from a class other than C_i and the C_i class hypothesis is being entertained, then $z_{ik}^2 = g_{ik}$; that is,

$$E\left\{ z_{ik}^2 \mid \vec{x} \notin C_i \right\} = g_{ik}$$

which means that

$$E\left\{ z_i^T z_i \mid \vec{x} \notin C_i \right\} = tr(\mathbf{g}_i) = \sum_{k=1}^{N_f} g_{ik}$$

We note immediately that if each g_{ik} were much greater than one, then it would be very easy to detect that a false hypothesis was being entertained.

The elements of the diagonal **g** matrices are of particular interest since each value corresponds to the variance of that corresponding feature in the whitened vector space. Since the expected value of the variance of given feature in the new, uncorrelated vector space is unity, a large element of a **g** matrix (i.e., greater than one) indicates a direction in the rotated and scaled vector space in which it should be possible to easily distinguish between a correct and incorrect hypothesis. A small eigenvalue (i.e., less than one) indicates a direction in the feature space in which it is difficult to distinguish between a correct and incorrect hypothesis. In fact, using such a transform feature value will be detrimental or misleading to the decision process (finding the minimum S). An eigenvalue of exactly one (1.00) means that the information found in the corresponding direction in the vector space is expected to contribute nothing to the decision-making process. It is not helpful or misleading, but irrelevant.

It should be noted that the statistics S_i 's in the decision rule can be expressed in terms of the $\vec{z}$'s or the $\vec{y}_i$'s. Everything that has been discussed so far is exactly equivalent to the standard maximum-likelihood decision rule.

$$S(\vec{x}) = K + \vec{z}^T\vec{z} = K_i + \vec{y}_i^T\vec{y} \tag{22}$$

$$= K_i + (\vec{x} - \vec{u})^T \mathbf{B}^{-1}(\vec{x} - \vec{\mu}_i)$$

$$= K + (\vec{x} - \vec{\mu})^T \mathbf{W} \mathbf{W}^{\mathbf{T}}(\vec{x} - \vec{u})$$

$$= K + \sum_{k=1}^{N_r} z_k^2$$

$$= K + \sum_{k=1}^{N_r} \left\{ \sum_{m=1}^{N_r} \left[\mathbf{W}^T\right]_{km} (\vec{x}_m - \vec{\mu}_m) \right\}^2 \tag{22a}$$

This last equation really summarizes the results of this analysis. If $\{K \ \ \mathbf{W} \ \ \vec{\mu}\}$ is precomputed, then $S(\vec{x})$ can be evaluated for any unknown vector and any class C . These precomputations can be done using the same type of training data required for minimum distance or K-Nearest Neighbor classifiers. And the precomputed information is equivalent to knowing $\{K , \mathbf{B}^{-1}, \vec{\mu}\}$, which are required by the straightforward computation. This is because $\mathbf{W}$ is a matrix square root of $\mathbf{B}_i^{-1}$.

But let us examine the form of the distance metric expectation under the in-class and out-of-class hypotheses:

$$\mathrm{E}\left\{ S\,(\vec{x}) \mid \vec{x} \in C \right\} = K + N_F$$

$$\mathrm{E}\left\{ S\,(\vec{x}) \mid \vec{x} \notin C \right\} = K + \sum_{k=1}^{N_f} g_{ik}$$

Note that the ith hypothesis can be rejected as soon as the accumulating sum $S\,(\vec{x})$ is greater than the lowest sum S already calculated. Also, note that it is advantageous, in terms of run-time computations, to shuffle the rows of the $\mathbf{W}^T$ matrices according to the order of the g_{ik} eigenvalues calculated previously. By placing the *k-th* rows of the $\mathbf{W}^T$ matrix corresponding to larger g_{ik} elements higher up (row-wise) in the matrices, the decision process may be allowed to jump out of the computational inner product loop very quickly when entertaining false hypotheses. It is in these manipulations that the real efficiency of the classification process is realized. We call this soft (or adaptive) dimensionality reduction because the number of z features evaluated changes for every decision, but is ordinarily much less than the maximum number possible (i.e., the full dimension of the feature space). Notice that no feature data has been disregarded, but the training data has been used more effectively to quickly dismiss false hypotheses. The decision surfaces in the N_F-dimensional feature space are still the same.

Hard (or fixed) dimensionality reduction can be used in conjunction with the soft dimensionality reduction mentioned above. At this point, a break is made from conventional multivariate-Gaussian maximum-likelihood decision making. The decision surfaces in the N_F-dimensional feature space are changed slightly due to hard dimensionality reduction. Let N_{gi} be the number of $\mathbf{g}_i$ eigenvalues greater than one for the class C_i. We call this N_{gi} the intrinsic dimensionality of the decision space for the class C_i. Hard dimensionality reduction can be achieved after the $\mathbf{W}^T$ matrices have been shuffled row-wise by only considering the first N_{gi} features in the whitened z feature space when entertaining the hypothesis that $\vec{x}$ is a member of class C_i and simply adding one (1.0) to the sum $S_i(\vec{x})$ whenever the g_{ik} eigenvalue corresponding to the z_{ik} feature is less than or equal to one. Hence, certain z_{ik} terms need never be computed, saving execution time, and the corresponding row of the $\mathbf{W}_i^T$ matrix can be discarded, saving storage space. It is possible to achieve this hard dimensionality reduction without decreasing the effective decision accuracy obtained from straightforward maximum-likelihood approach. This is reasonable because the features corresponding to the $\mathbf{g}$ eigenvalues less than one are actually detrimental to the classification process when entertaining the given hypothesis. If the sum $S_i(\vec{x})$ is to be minimized in order to make the correct decision, and if it is expected that the value one (1.0) will be added to the sum when the correct hypothesis is being entertained, then it is easy to see that if it is expected that a value of less than one is to be added when the hypothesis is known to be wrong, the chances of making the correct decision are being diminished. In fact, it does not seem unreasonable that better performance can be achieved by ignoring the misleading information. This is the key to the objectivity of the ODR technique. No subjective thresholds of any sort are required to determine which features should be used. The threshold is always one as computed analytically.

The average and standard deviation of the value of $S_i(\vec{x}) - K_i$ for the correct decisions made on a particular image were computed to get an idea of the variance of the expected value N_F as in the equation above. For soft dimensionality reduction, we obtained the value of 25 ± 12.7; for hard dimensionality reduction, 26 ± 10.1. These results are compatible with the expected value of 27.

In our limited experiments, we have not seen markedly different performance when using the hard dimensionality reduction capability of the ODR method, but we know that the computational load is further reduced. Sometimes we have seen slightly better performance and other times, slightly worse performance; none of these changes appear to be statistically significant. Our experimental results in Figures 3 and 4 show an example of this phenomenon. Dye [1974] reports that definite improvements in classification performance have been obtained owing to the hard dimensionality reduction, but that these improvements are relatively small.

In summary, Dye's theory involving out-of-class covariance is sound, and we feel it is a major contribution to computational methods for multivariate-Gaussian maximum-likelihood decision-making. It is interesting to note that some artificial intelligence research has focussed on the similar problem of measuring disbelief in a given hypothesis as well as computing belief measures [Khan and Jain 1985].

It is extremely important to note that this procedure precisely indicates the preferred dimensionality of the feature space *for each class.* As Dye states, "dimensionality reduction can differ from class to class in accordance with the differing statistical properties of each class relative to the others, rather than remaining fixed at a number imposed by compromise."

Dimensionality reduction in the original feature vector space has not yet been addressed. The features in the original $\vec{x}$ space can only be useful or useless; they cannot be directly detrimental to decision making. Useful features will be weighted with non-zero coefficients in the $\mathbf{W}_i^T$ matrices whereas useless feature receive zero (or very small) weights. To test for useless original feature data, a feature must not be used by any class for any z feature. The usefulness of the kth feature in $\vec{x}$ may be tested by summing the squares of the components of the kth column vector in the $\mathbf{W}_i^T$ matrix for each class C_i. If the components of $\vec{x}$ are zero-mean, unit-variance (not conditioned on any class hypotheses), and if this sum is below a selected small threshold (close to zero) for every class, then the kth feature is not being used and can be discarded. Otherwise, it is contributing something to at least one class decision and should be kept and labeled as useful. Moreover, that sum can be used for ranking purposes.

We can express these ideas more formally as follows. If we let $\vec{W}_{ik}$ be the kth column vector of the $\mathbf{W}_i^T$ matrix, we can compute an overall usefulness parameter α_k that can be used to rank the importance of the features in the original $\vec{x}$ space:

$$\alpha_k = \sum_{i=1}^{N_C} | \vec{W}_{ik} |^2$$

where the bars denote the vector norm. Hence, we can say that the feature vector component $\vec{x}_k$ is more useful than $\vec{x}_{k'}$ if $\alpha_k > \alpha_{k'}$. The feature $\vec{x}_k$ is useless if $\alpha_k < \epsilon$ where ϵ is a small positive number chosen to account for numerical imprecision.

4. COMPARISONS AND OBSERVATIONS

When analyzing a statistical pattern recognition technique (or any new algorithm), it is desirable to draw parallels between the new method in question and standard, well-tested algorithms. This is helpful in understanding the performance potential of the algorithm as well as allowing an easier grasp of its underlying concepts. For this reason, we compare the ODR method to a standard hard dimensionality reduction technique commonly known as Parametric Discriminant Analysis. The ideas of simultaneous diagonalization and in-class covariance are common to both, but parametric discriminant analysis uses a single between-class covariance matrix instead of the multiple out-of-class covariance matrices used by the ODR technique.

Parametric Discriminant Analysis (PDA) is the multiple-class generalization of Fisher's linear discriminant analysis [Fisher 1936] [Duda and Hart 1973] and involves the calculation of an overall *within-class covariance matrix* and an overall *between-class covariance matrix.* The representation for the within-class scatter matrix is straightforward. It simply involves the summation of the individual within-class covariance matrices calculated previously.

$$\mathbf{S}_W = \sum_{i=1}^{N_C} p_i \mathbf{B}_i = \mathrm{E}\left\{\mathbf{B}_i\right\} \tag{23}$$

where as before

$$\mathbf{B}_i = \mathrm{E}\left\{(\vec{x} - \vec{\mu}_i)(\vec{x} - \vec{\mu}_i)^T \mid \vec{x} \in C_i\right\} \tag{24}$$

and p_i is the "probability" of occurrence of the ith class in the training data and is defined as:

$$p_i = \frac{L_i}{N_T} \tag{25}$$

The computation of the overall between-class covariance matrix can also be expressed as an expectation. This matrix is defined as follows:

$$\mathbf{S}_B = \sum_{i=1}^{N_T} p_i\, (\vec{\mu}_i - \vec{\mu})(\vec{\mu}_i - \vec{\mu})^T = \mathrm{E}\left\{(\vec{\mu}_i - \vec{\mu})(\vec{\mu}_i - \vec{\mu})^T\right\} \tag{26}$$

where μ is the overall feature mean vector defined as:

$$\mu = \frac{1}{N_T} \sum_{j=1}^{N_T} \vec{x}_j = \sum_{i=1}^{N_C} p_i \vec{\mu}_i = \mathrm{E}\left\{\vec{\mu}_i\right\} \tag{27}$$

Note that $\mathbf{S}_W$ averages all "information" about the scatter of feature data under the in-class hypothesis whereas $\mathbf{S}_B$ averages all the information about the scatter of the class mean vectors relative to the central point (mean vector) of the unclassified mixture of feature data. PDA then defines a criterion function J that attempts to measure in a single scalar quantity the ratio of the spread between all of the classes to the internal spread within the classes. J will be large when all feature vectors for the given classes are tightly grouped about their respective class means and the class means are far apart compared to the tight grouping of the classes. Thus, one can expect excellent decision-making performance when J is large, and the larger the better. The two most commonly used criterion functions for J are

$$J_1 = tr(\mathbf{S}_W^{-1}\mathbf{S}_B)$$

$$J_2 = \frac{\det(\mathbf{S}_B)}{\det(\mathbf{S}_W)}$$

By introducing a new transformation matrix **W** that maps original features into new, better features, one finds that both J_1 and J_2 can be maximized by performing a simultaneous diagonalization on $\mathbf{S}_B$ and $\mathbf{S}_W$ (see [Duda and Hart 1973] or [Fukunaga 1972]). The optimizing matrix **W** is the transformation matrix resulting from the simultaneous diagonalization as in the ODR technique. Since the rank of $\mathbf{S}_B$ can be no larger than $N_C - 1$, the matrix **W** provides a mapping from the original set of N_F features to a reduced dimension set of the $N_C - 1$ best features that are linear combinations of the original features and apply to all class decisions. It is interesting to note that the optimal Bayes classifier only needs $N_C - 1$ features to produce minimum error decisions and ignores all additional features. Unfortunately, in the PDA method, there are no guarantees that the reduced dimensionality feature vectors will provide adequate decision performance. One can seldom even hope for nearly equivalent performance because there are usually many more features than classes and not enough feature information is retained. Efforts have been made to artificially introduce new classes and to reuse the same procedure on the orthogonal subspace of feature data that remains when the best $N_C - 1$ feature subspace has been extracted. However effective those methods might be, they do not offer the simplicity, elegance, and objectivity of the ODR approach. A critical difference here is that there is only one **W** matrix; no attempt is made to handle each class separately as in the ODR method. The set of features actually extracted must be used for all class decisions and is therefore a compromise of the best features for each class.

We have introduced an existing method related to the ODR technique and have seen how the $\mathbf{S}_W$ matrix and the $\mathbf{B}_i$ matrices are related. It is interesting to examine how the $\mathbf{S}_B$ and $\mathbf{F}_i$ matrices are related to each other. To express this $\mathbf{S}_B$ matrix in terms of our known ODR quantities, it is convenient to define the total (or mixture) covariance matrix as follows:

$$\mathbf{S}_T = \frac{1}{N_T} \sum_{j=1}^{N_T} (\vec{x}_j - \vec{\mu})(\vec{x}_j - \vec{\mu})^T = \mathbf{S}_W + \mathbf{S}_B \tag{28}$$

$$= \mathrm{E}\left\{ \vec{x}\,\vec{x}^T \right\} - \vec{\mu}\,\vec{\mu}^T$$

It is possible (after lengthy algebraic manipulations) to express the **F** matrices calculated in equation (7) in terms of this total covariance matrix and the probability of occurrence of each particular class in the training set:

$$\mathbf{F}_i = \frac{1}{1 - p_i} \left[\mathbf{S}_T + (\vec{\mu} - \vec{\mu}_i)(\vec{\mu} - \vec{\mu}_i)^T - p_i \mathbf{B}_i \right] \tag{29}$$

Summing the resulting expression over the total number of classes, and multiplying by the probability of occurrence of that particular class produces the expectation of the **F** matrix:

$$\mathrm{E}\left\{ \mathbf{F}_i \right\} = \sum_{j=1}^{N_C} p_i \, \mathbf{F}_i \neq \mathbf{S}_B \tag{30}$$

Taking a naive view, one might expect this to yield the overall between-class covariance matrix just as summing the **B** matrices produces the within-class covariance matrix. However, since one **F** matrix already averages over all classes that are not the assumed class, summing **F** matrices over each class produces quite a bit of duplicate averaging that must be compensated for as we show in the next equation.

Given equation (29), one can express the overall between-class scatter matrix in terms of the **B** and **F** matrices from before:

$$\frac{1}{2} \sum_{i=1}^{N_C} (1 - p_i)(p_i)(\mathbf{F}_i - \mathbf{B}_i) = \mathbf{S}_B \tag{31}$$

The overall between-class covariance matrix is one half the sum over all the classes of the product of the probability of occurrence of each class times the probability of not being in that class times the difference of the out-of-class covariance matrix and the in-class covariance matrix. The resultant summed matrix must be of rank N_C - 1 or less [Duda and Hart 1973] [Fukunaga and Mantock 1983] whereas the $\mathbf{F}_i$ matrices are typically all of full rank.

4.1. Two Class, Equal-Covariance Case

It is illuminating to point out that the ODR and PDA methods yield the same transformation matrix W in the simplest case of a two-class, equal-covariance matrix, multivariate Gaussian decision problem. Since they both simplify to the same algorithm in the simplest case, the ODR method can be viewed as a more effective alternative to the multiple-class generalization provided by the PDA technique.

Consider the case where $\mathbf{B} = \mathbf{B}_1 = \mathbf{B}_2$, and we have an equal number of training samples from each class. Then we have $\mathbf{F} = \mathbf{F}_1 = \mathbf{F}_2$ and $\mathbf{W} = \mathbf{W}_1 = \mathbf{W}_2$. This allows us to express all the basic quantities mentioned above in a much simpler form.

$$\mathbf{F} = \mathbf{B} + \left[(\vec{\mu}_1 - \vec{\mu}_2)(\vec{\mu}_1 - \vec{\mu}_2)^T \right] = \mathbf{S}_W + 4\mathbf{S}_B$$

$$\mathbf{S}_W = \mathbf{B}$$

$$\mathbf{S}_B = \frac{1}{4}(\vec{\mu}_1 - \vec{\mu}_2)(\vec{\mu}_1 - \vec{\mu}_2)^T$$

$$\vec{\mu} = \frac{1}{2}(\vec{\mu}_1 + \vec{\mu}_2)$$

Of course, by definition of simultaneous diagonalization, we still have

$$\mathbf{I} = \mathbf{W}^T\mathbf{B}\mathbf{W} \qquad \mathbf{g} = \mathbf{W}^T\mathbf{F}\mathbf{W}$$

This implies that

$$\mathbf{g} = \mathbf{I} + \left[\vec{\eta}\,\vec{\eta}^{\,T}\right]$$

where

$$\vec{\eta} = \mathbf{W}^T(\mu_1 - \mu_2)$$

Without loss of generality, there exists a transformation matrix $\mathbf{W}$ satisfying all above conditions such that

$$\vec{\eta} = [\,\eta \;\; 0 \;\; 0 \;\; \cdots \;\; 0\,]^T$$

which implies that

$$\left[\vec{\eta}\,\vec{\eta}^{\,T}\right] = diag\,(\eta^2, 0, 0, \;\cdots\;, 0)$$

where $diag\,(\cdot)$ denotes a diagonal matrix with vector argument on the diagonal. Therefore, we also have the desired equivalent simultaneous diagonalization action in terms of the within-class and between-class matrices:

$$\mathbf{W}^T\mathbf{S}_W\mathbf{W} = \mathbf{I} \qquad \mathbf{W}^T\mathbf{S}_B\mathbf{W} = diag\,(\frac{\eta^2}{4}\,, 0, 0, \;\cdots\;, 0)\,.$$

Both ODR and PDA methods result in the optimal Bayes classifier in this special case under the assumption of equal *a priori* probabilities. Both the ODR hard dimensionality reduction method and the PDA method reduce the original $N_F \times N_F$ **W** matrix to a $N_F \times 1$ matrix. The ODR hard dimensionality reduction method ignores the rest of the **W** matrix because the associated eigenvalues in the **g** matrix are less than or equal to one (1.0); in this case, all g_i eigenvalues are exactly equal to one except for $g_1 = 1+\eta^2 > 1$. The PDA method ignores the rest of the **W** matrix because the original outer product matrix of the class mean difference vector was of rank one. To be more specific, let $\vec{W}_1$ be the first column vector of the **W** matrix. Bayes test then becomes the linear discriminant test:

$$\vec{W}_1^T(\vec{x} - \vec{\mu}) \underset{>}{<} 0.$$

which is exactly equivalent to the more familiar formula:

$$\vec{x}^T \mathbf{B}^{-1}(\vec{\mu}_1 - \vec{\mu}_2) \underset{>}{<} \frac{1}{2}(\vec{\mu}_1 + \vec{\mu}_2)\mathbf{B}^{-1}(\vec{\mu}_1 - \vec{\mu}_2)$$

The resulting probability of error, given by [Duda and Hart 1973], is

$$P(e) = \frac{1}{\sqrt{2\pi}} \int_d^\infty \exp(\frac{-u^2}{2})du$$

where

$$d = \frac{1}{2} \mid \mathbf{W}^T(\vec{\mu}_1 - \vec{\mu}_2) \mid$$

Although an inner product and a comparison against a pre-computed threshold is all that is required in the standard approach, the PDA approach, and the

ODR approach for this ***very simple special case***, the benefits of the ODR approach for many class, many feature problem should be clear. Moreover, the ODR method provides greater computational benefits as the dimensionality of the feature space increases and as the number of classes increases.

4.2. Summary

We have attempted to note the similarities and differences between the ODR method and one existing related method, the PDA method. The within-class and between-class covariance matrices are simultaneously diagonalized in the PDA approach just as the in-class and out-of-class covariance matrices are simultaneously diagonalized in ODR analysis. PDA results in one transform matrix while the ODR method creates a transform matrix for each class hypothesis. Thus, the ODR method is able to treat each class "personally" whereas PDA must be "democratic" and can only deal with individual classes within the group context. Although the primary use of multiple-class parametric discriminant analysis is for the purpose of hard dimensionality reduction, this comparison hopefully can provide some insight into the nature of the ODR technique, which directly provides soft and hard dimensionality reduction and classification.

Another advanced, related, noteworthy technique that should be mentioned is the nonparametric discriminant analysis method proposed by Fukunaga and Mantock [1983]. This approach also determines a linear transformation and provides a scalar measure for each feature that indicates its quality, but allows dimensionality reduction to be controlled by the user and does not determine the intrinsic dimensionality for each class directly from the data as in the ODR technique.

5. VISUAL SOLDER JOINT INSPECTION APPLICATION

The problem of automating solder-side post-solder-wave solder joint inspection on plated-through-hole printed circuit boards is considered by many to be quite a formidable task. Jones [1985] describes this problem as "the (printed circuit board manufacturing) industry's toughest technical problem." Given a gray-scale image of solder joints, a decision must be made for each joint regarding its acceptability and defective nature if it is not acceptable. We have addressed the inspection problem by treating it as a standard pattern recognition problem where a set of usual and unusual gray-scale image features are utilized [Besl et al. 1985]. There are currently twenty-seven features calculated for each solder joint subimage, which has been automatically isolated from a larger image containing several solder joints. Figure 1 displays an example image of plated-through-hole solder joints on a printed circuit board and a class-labeled version of that same image. Each solder joint in a set of these images has been classified by a human operator to use for training and testing the ODR algorithm. The following set of nine classes has been chosen to categorize solder joint subimages according to their defects:

A = Acceptable Joint (dark to medium brightness)

B = Acceptable Joint (medium bright to bright)

C = Cup-shaped Filled Hole (no lead)

D = Disturbed/Deformed solder (lead?)

E = Excessive solder (lead?)

F = Filled Hole (flat solder surface, no lead)

H = Hole (no lead, no solder)

I = Insufficient solder (poor fillet on lead)

N = No solder (lead present but no solder fillet)

A and B type subimages represent acceptable solder joints. C and F type subimages represent plated-through-holes that the solder has completely filled even though no component lead is present. C and F type subimages do not really represent solder joint defects, but they do need to be recognized. Some applications require F type filled holes to meet certain quality specifications, and in these cases, C subimages may represent undesirable defects. D and E type subimages represent globs of solder. E type subimages represent smooth surfaced (almost hemispherical) blobs of solder where the D label is used for any blobs with unsmooth surfaces. There is usually too much solder in D and E types to tell if a component lead is present or not. H and N type subimages represent large holes that are not filled with solder: the N label implies a component lead is present whereas H implies otherwise. I type subimages represent component leads with solder where the solder fillet is not acceptable due to one of many reasons.

Because of the complexity of this multiple class problem and the difficulty in obtaining uncorrelated gray-scale image features, Automatic Visual Solder Joint Inspection is an excellent problem to address with the ODR (Objective Dimensionality Reduction) Statistical Pattern Analysis technique. The selection of informative scalar features for gray-level solder joint subimages is a difficult task, especially when a classification algorithm is used that cannot objectively rank features according to their usefulness for decision-making. Since the ODR Technique creates different weighting values for each feature depending on class hypotheses and it permits hard dimensionality reduction, features are only used when the information contained therein is helpful for making a decision about a particular class. Because of this, the addition of uninformative features does not turn out to be detrimental to the classification process. That is, we can afford to be somewhat speculative in feature selection because we know that our selection mistakes will be automatically ignored.

5.1. Summary of Solder Joint Subimage Inspection Features

Twenty-seven (27) features are currently calculated for each solder joint image. These can be divided up into five categories: (1) Basic Gray-Level Statistics Features, (2) 3-D Gray-Level Inertia Features, (3) Gray-Level Surface Area Features, (4) Differential Geometric Gray-Level Surface Curvature Features, and (5) Binary Image Connected-Region Features. The known correlated features have been removed from this list. Out of twenty-seven features, sixteen are general purpose gray-level subimage features, and the other eleven involve application specific assumptions. Of these eleven, five are gray-level subimage features and six are binary subimage features.

The numbers (labels) associated with each feature have no meaning. They resulted from the order in which different features were included in the feature extraction subroutine.

5.1.1. Basic Gray Level Statistics Features

Feature 0 is the normalized standard deviation of the gray-level surface. Feature 1 is the normalized mean gray level, or gray volume. Feature 2 is the normalized central subwindow gray volume. Feature 3 is the normalized outer frame region gray volume. Feature 7 is the minimum normalized gray level in the subimage. Feature 19 is the percentage of dark pixels in the subimage (within 5% of the 0 (dark) gray level). Feature 20 is the percentage of bright pixels in the subimage (within 5% of the maximum gray level). Features 2 and 3 are application specific features whereas the others are general purpose.

5.1.2. 3-D Gray-Level Inertia Features

Feature 4 is the first principal (spatial) moment of inertia. Feature 5 is the ratio of the brightness moment to the average of the two spatial moments of inertia. Feature 6 is the sum of all three moments of inertia.

5.1.3. Faceted Gray-Level Surface Area Features

Feature 10 is the approximate surface area obtained by summing the gray-level surface metric determinant over all pixels. Feature 11 is the faceted gray level surface area.

5.1.4. Differential Geometric Gray-Level Surface Curvature Features

Feature 8 is percentage of positive Gaussian curvature pixels. Feature 9 is percentage of negative Gaussian curvature pixels. The number of zero Gaussian curvature pixels is typically non-zero preventing these two features from being correlated. Feature 13 is the average value of positive Gaussian curvature. Feature 14 is the average value of negative Gaussian curvature. Feature 17 is percentage of positive mean curvature pixels. Feature 18 is percentage of negative mean curvature pixels. The number of zero mean curvature pixels is typically non-zero preventing these two features from being correlated. Feature 15 is the average value of positive mean curvature. Feature 16 is the average value of negative mean curvature. Feature 12 is the quadratic variation of the gray level surface.

5.1.5. Binary Image Connected-Region Features

Gray-level images of several solder joint types have very distinctive characteristics when thresholded to create binary images. For example, the tip of a component lead often appears as a bright region in the subimage, and therefore it will show up as a region in the thresholded image. All subimages were thresholded at gray level based on the overall brightness of the image (85 on a scale of 256 for our implementation), to provide the greatest amount of information. This threshold can be computed automatically via histogram analysis, but must make some domain-specific assumptions. The lead tip connected-component regions are clearly seen in the top two binary images.

The first region related feature is Feature 21, which is the number of four-connected regions in the thresholded solder joint subimage. Since those subimages containing solder leads are guaranteed to contain at least one region, this feature is effective in separating those subimages containing solder leads from those without solder leads.

Feature 22 is the number of pixels in the largest four-connected region in the thresholded image. This feature provides excellent separation of classes, and is therefore a great contributor to the classification process. Often the largest region in an acceptable type joint is the lead tip, while, for an F or C type joint, the largest region is usually several times that size. Holes usually produce small four-connected bright regions.

Feature 23 is the number of pixels in the thresholded image that are not in the largest region. This feature serves to separate those joint types with little or no solder present, for example N or H type joints, from those with solder present.

Feature 24 is the ratio of the area of the min/max box around the largest region to the number of pixels in the largest region. This feature also helps distinguish N and H type solder joints. Since the largest region present is often the border of the solder pad, the min/max box will therefore almost occupy the entire subimage, making the ratio quite large.

Feature 25 is the aspect ratio (width to height) of the min/max box surrounding the largest region.

Feature 26 is the ratio of the perimeter squared to the area for the largest region within that subimage. This feature works well in separating E type and some F type joints from the others. Since the four-connected regions in the E type joints often appear as C-shapes or rings (owing to the use of a toroidal fluorescent tube for lighting), the ratio of the perimeter to the area is large in comparison to that for other joint types, which usually contain roughly rectangular or circular regions.

5.1.6. Complete Ranking of Feature Usage

Although all features were found to be useful, some features are more useful than others. In order to make this quantitative, we computed a Euclidean metric of the weighting factors for each feature across all classes. The metric was computed with and without hard dimensionality reduction. Figure 8 shows the ranking of all the features according to their average usefulness across every class decision for both cases. The top ten and worst five subimage features are the same in both cases and are briefly discussed. Due to the symmetry properties of solder joint subimages from an overhead view, the three 3-D inertia properties are rated the highest, followed by the average normalized gray level and standard deviation. The number of pixels in the thresholded binary image was ranked sixth with the approximate surface area measure coming in seventh. The average positive Gaussian curvature feature was ranked next, which means that it actually characterizes bright spots in subimages. The quadratic variation was ranked ninth followed by the central subwindow volume feature. The worst five features are the min/max box occupancy ratio and the peround measure of the largest binary-image region, the two Gaussian curvature percentages, and the negative Gaussian curvature average.

5.2. Measurement of Performance

To measure the performance of the ODR technique using the 27 features and the 9 classes, five quantities are calculated. The first is called Correct Classifications (*CC*) and is defined as the number of joints where the classification assigned by the algorithm agrees with that assigned by the human classifier.

The second performance measure is the number of unacceptable joints classified as acceptable by the algorithm (*Misses*).

The third quantity is the number of acceptable joints classified as unacceptable by the algorithm (*False_Alarms*).

The fourth quantity calculated is the percent correctness of good / bad decisions made by the algorithm (*%GB*). This is calculated by summing the number of acceptable joints correctly classified as acceptable and the number of unacceptable joints correctly classified as unacceptable and dividing by the total number of joints being considered. We consider this to be the most important index of performance.

The final quantity calculated is the percent correctness of class decisions (*%CC*). This is calculated by taking the number of correct classifications (*CC*) and dividing by the number of solder joints being considered.

5.3. Computation Reduction

Figure 2 displays the reduction in computation due to the use of the out-of-class covariance matrices. Each plot is a histogram that shows the relative number of times that a particular number of z features (inner product computations) were needed. We can denote that number of z features as n_z and the number of times it is used as $n_d(n_z)$. We refer to such a histogram as a dimensionality histogram because it can be viewed a plot of the number of decisions made in each dimension from 0 to $N_F = 27$.

The top plot displays the MLMVG algorithm results when the $S_i(\vec{x})$ statistic is computed using the square root **W** matrices instead of the $\mathbf{B}^{-1}$ matrices and the conditional termination criteria is used. To our knowledge, most MLMVG implementations do not even use this trick and therefore, every decision would require the full N_F inner products.

The second plot displays the results when the **W** matrices have been sorted in decreasing order according the g_i eigenvalues. Notice the peak that occurs at $n_z = 1$. This means that it is quite common that a decision can be made after evaluating *only one z* feature. The peak at $n_z = 27$ still occurs because every solder joint decision must have the full computation for at least one hypothesis.

The third plot shows the effects of hard dimensionality reduction on the dimensionality histogram. The major effect is that the peak at $n_z = 27$ is broken down into several smaller peaks corresponding to the different intrinsic dimensionalities of the decision spaces for the different classes. In our experiments, the N_{gi} values were the same for several class hypotheses. The low dimension portion of this third plot looks very similar to the second plot as it should.

The fourth plot shows the ODR algorithm in the minimum distance classifier emulation mode where all $\mathbf{W}$ matrices are the identity matrix and all K_i values are zero. The number of z values then does not indicate the number of inner products in this case because the $\vec{x}$ feature vector can be used directly. However, it is interesting to note that the dimensionality of the average decision is very high.

5.4. Performance of the ODR Algorithm

A database of seven images was used for testing of the ODR Technique. Each of theses images were digitized with the camera directly over the printed circuit board, keeping exposure and lighting constant for each. Each individual image contained over 190 solder joint subimages, although only 65 were actually recorded in the database for the last image (A8).

It can be shown [Kalayeh and Landgrebe 1983] that (1) N_F+1 training samples per class are absolutely necessary, (2) $5N_F+1$ training samples per class are preferred, and (3) $10N_F+1$ samples would yield excellent estimates of the required covariance matrices needed for multivariate normal processing. Obtaining low variance results from the mean and covariance calculations is necessary to make the classification process reliable and consistent. Therefore, all 1286 of the solder joint subimages in the A image database were used in the training stage. Unfortunately, this left no remaining joints from the same database with the same image resolution and the same lighting for testing purposes outside the training set. However, we are very satisfied with our results at this stage in our research in that no other method of the several that we have tried was able to demonstrate such good performance even when trained on every subimage. Most methods performed worse when trained on all the image data.

Classification results for the A-type images without and with hard dimensionality reduction is shown in Figures 3 and 4 respectively. It is seen that the hard dimensionality reduction does little to affect performance even though computation was reduced (on average, by two inner products of length 27). These results *include errors* made on solder joints that were humanly classified with a class label that was not used in the nine classes. For comparison purposes, we altered the input to the ODR algorithm so that it would behave as an unweighted, normalized minimum-distance classifier. Figure 5 shows the marked decrease in performance even though these results are better than previous results, which indicates that the normalization process and the new features are of some help. The best overall results from the minimum-distance classifier previously used [Besl et al. 1985] are displayed in Figure 6. As can be seen, the ODR Technique results are far superior to those previously obtained.

Our latest results are from classifications of images which are subsets of the training set. However, we have no evidence to suggest that performance will diminish significantly when dealing with new data because there is a substantial amount of image variation just in the A set. For testing purposes, two new images were digitized, each having one hundred classifiable joints. An attempt was made to duplicate the resolution and lighting conditions from the A-type images. One image, K1, was taken from the same printed circuit board as the A-type images, and the other, K2, was taken from a new PC board. As can be seen in Figure 7, these results are not on the same level as those obtained from the A-images. Unfortunately, the classification of objects with reflective surfaces, such as solder joints, is extremely sensitive to the changes in lighting. Therefore, these results should not be considered indicative of the actual performance of this algorithm with unknown data, as a solder joint inspection station would maintain much more consistent lighting.

5.5. Distance Table for the Solder Joint Class Road Map

Many of the concepts presented here are not easy to grasp owing to their multidimensional nature. However, almost everyone is familiar with a distance table that lists the distances between cities on a map, and the distance table concept is not limited to two dimensions. Figure 9 shows two distance tables for the metric distances between the mean vectors of the nine solder joint classes, which are analogous to different cities. The units of distance are expressed as percentages of the average between-class distance. Table (a) shows the distances in a normalized 27 dimensional space, and summarizes the distribution of mean vectors that the minimum distance classification algorithm would have to work with. The two classes closest together and probably hardest to distinguish are class B and class I. This agrees with our qualitative visual impressions of these types. This table is naturally symmetric since the distance from point A to point B is the same as the distance from point B to point A. Also, the table does not account for the effects of internal class variance on the decision-making separability of classes because the Euclidean distance was used instead of the Bhattacharya or divergence distance measures. On the other hand, Table (b) shows a distance table that is not symmetric. It represents the distances between classes as seen by the ODR algorithm. For example, under the class B hypothesis, the class C mean is 21 units from the class B mean whereas they are separated by only 17 units under the class C hypothesis. This is due to the customized class decision structure of the ODR technique. That is, the two distances

$$| \mathbf{W}_i^T(\vec{\mu}_i - \vec{\mu}_j) |$$

and

$$| \mathbf{W}_j^T(\vec{\mu}_i - \vec{\mu}_j) |$$

are in general not equal. These tables are presented to help the reader picture general concepts in the context of a specific application with specific data. The second table points out the asymmetric nature of MLMVG class mean distances as considered by the ODR method.

6. SUMMARY

We have presented a method called the Objective Dimensionality Reduction (ODR) algorithm. This method uses the standard maximum-likelihood multivariate Gaussian (MLMVG) approach modified by the incorporation of out-of-class covariance matrix information. This modification reduces the amount of computation required to make approximate MLMVG decisions by more than a factor of three (3) and provides an objective indication of the intrinsic dimensionality of the decision space for each class hypothesis. The hard dimensionality reduction capability of the ODR approach makes it relatively insensitive to increases in input feature dimensionality because it ignores bad feature data. This method was compared to the commonly used parametric discriminant analysis technique for dimensionality reduction. Excellent experimental results for a 9 class, 27 feature solder joint inspection application were obtained with significantly reduced computations compared to the standard MLMVG method.

7. ACKNOWLEDGEMENTS

We gratefully acknowledge the helpful comments of R. Dye.

8. REFERENCES

BESL, P., DELP, E., AND JAIN. R. 1985. Automatic Visual Solder Joint Inspection. *IEEE Journal on Robotics and Automation* **1,** 1 (March), 42-56.

DUDA, R.O. AND HART, P.E. 1973. *Pattern Classification and Scene Analysis.* Wiley-Interscience, New York, NY.

DYE, R. 1974. "Multivariate Categorical Analysis - Bendix Style," Tech. Report BSR 4149, Bendix Corporation, Southfield, Mich. (June).

FISHER, R.A. 1936. The use of multiple measurements in taxonomic problems. *Ann. Eugenics,* vol. 7, part II, pp. 179-188.

FRIEDMAN, J.H. 1977. A Recursive Partitioning Decision Rule for Non-Parametric Classification. *IEEE Trans. Computers,* (April), 404-408.

FUKUNAGA, K. 1972 *Introduction to Statistical Pattern Recognition.* Academic Press, New York, NY.

FUKUNAGA, K. AND MANTOCK, J.M. 1983. Nonparametric Discriminant Analysis. *IEEE Trans. Pattern Analysis and Machine Intell.* PAMI-5, 6 (November), 671-678.

JAIN, A.K. AND WALLER, W.G. 1978. On the Optimal Number of Features in the Classification of Multivariate Gaussian Data. In *Proceedings of 4th International Joint Conference on* Pattern Recognition, (Kyoto, Japan, November 1978), pp. 265-269.

JONES, S.T. 1985. Flexible Inspection Systems for Printed Circuit Board Production: A Review of the State of the Art. Presented at *VISION '85 Conference* (Detroit, Mich., March 25-29), SME, Dearborn, Mich.; paper available from Control Automation, Inc., Princeton, N.J.

KALAYEH, H.M. AND LANDGREBE, D.A. 1983. Predicting the Required Number of Training Samples. *IEEE Trans. Pattern Analysis and Machine Intell.* PAMI-5, 6 (November), 664-667.

KHAN, N.A. AND JAIN, R.C. 1985. Uncertainty Management in a Distributed Knowledge Based System. In *Proceedings of 9th International Joint Conference on* Artificial Intelligence (Los Angeles, Calif., August 1985), pp. 318-320.

KULKARNI, A.V. AND KANAL, L.N. 1978. Admissible Search Strategies for Parametric and Nonparametric Hierarchical Classifiers. In *Proceedings of 4th International Joint Conference on* Pattern Recognition (Kyoto, Japan, November 1978), pp. 238-248.

SWAIN, P.H. 1985. Advanced Interpretation Techniques for Earth Data Information Systems. *Proc. IEEE* **73,** 6 (June), 1031-1039.

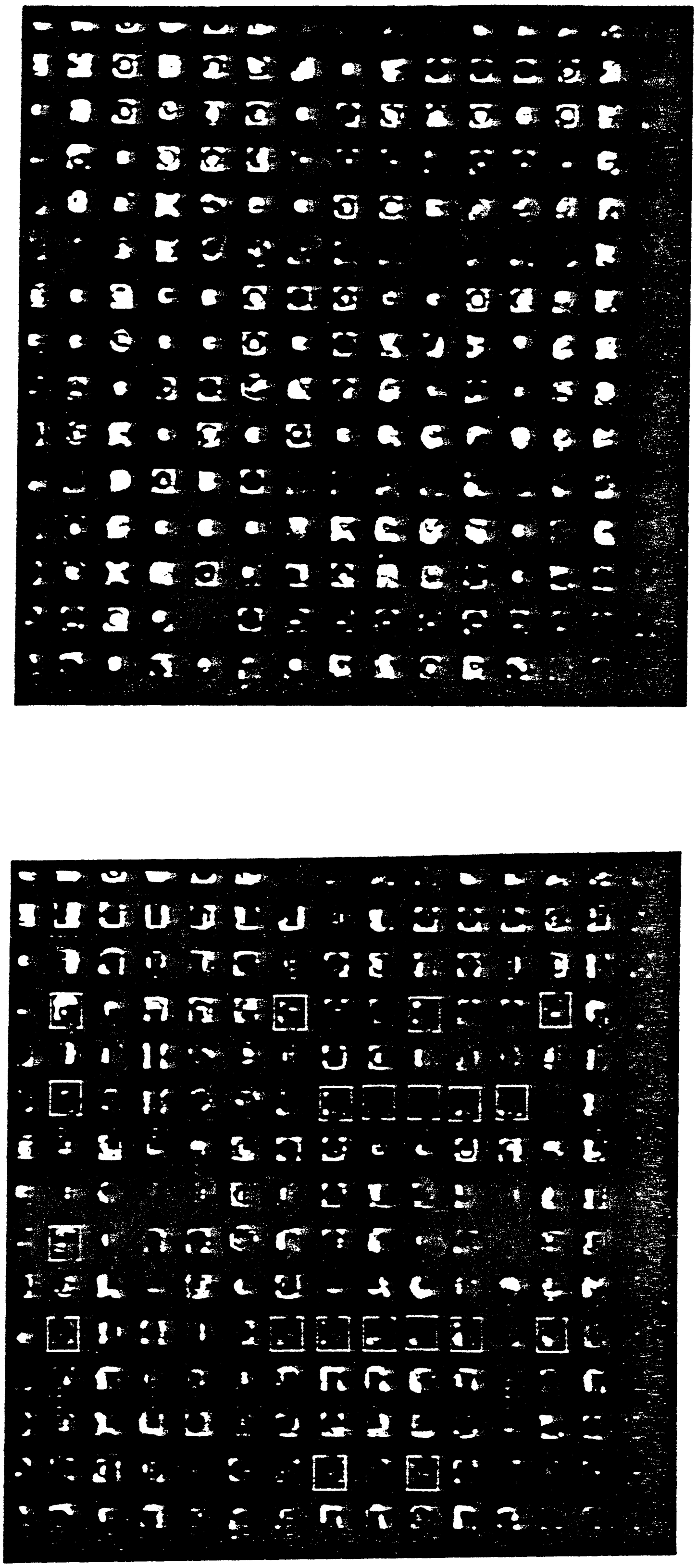

Figure 1. PCB Image and Labeled Image

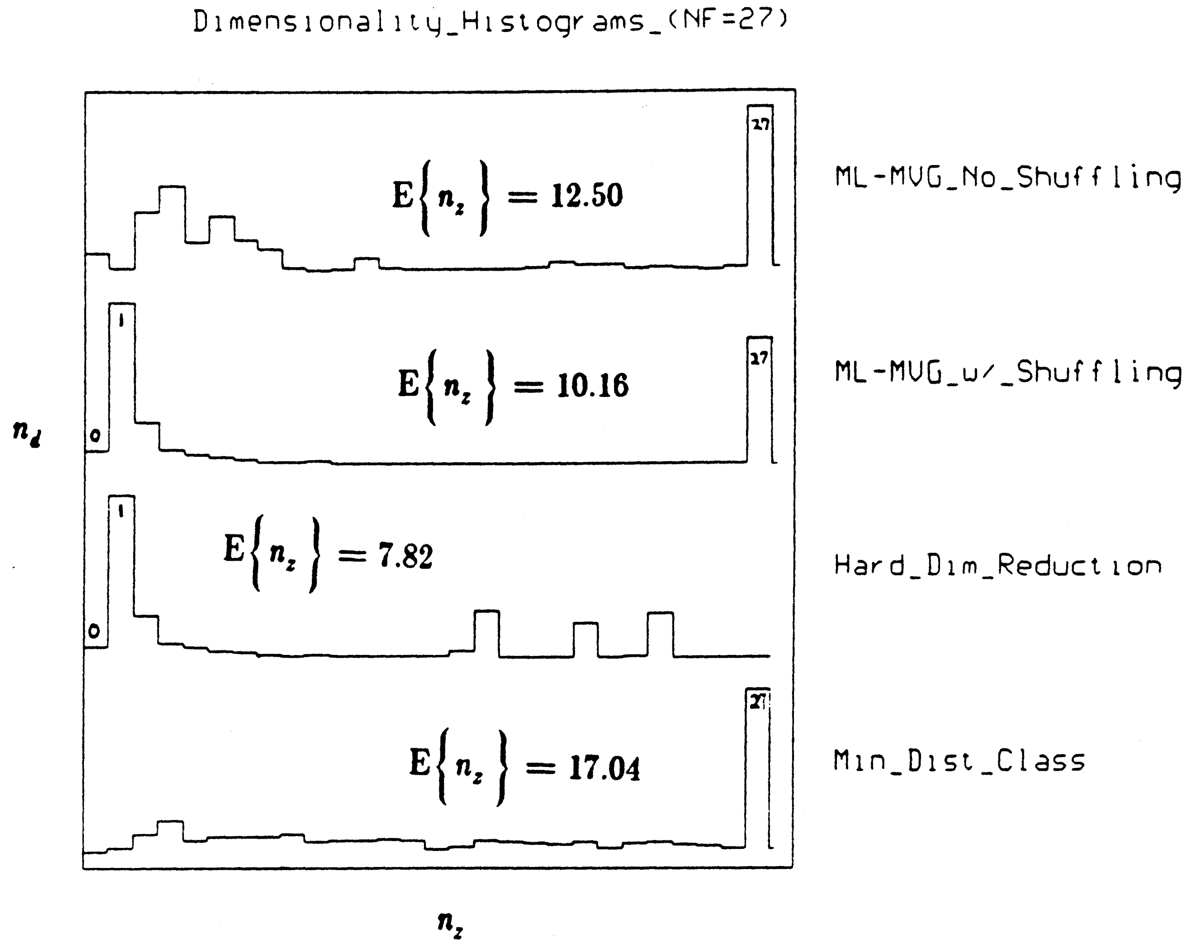

Figure 2. Dimensionality Histograms

<table>
<tr><th colspan="7">Classification Performance for The ODR Algorithm</th></tr>
<tr><th colspan="7">Soft Dimensionality Reduction Used</th></tr>
<tr><th>Image</th><th>Total_Joints</th><th>CC</th><th>Misses</th><th>False_Alarm</th><th>%GB</th><th>%CC</th></tr>
<tr><td>A1</td><td>196</td><td>168</td><td>10</td><td>0</td><td>94.9</td><td>85.7</td></tr>
<tr><td>A3</td><td>197</td><td>189</td><td>1</td><td>0</td><td>99.5</td><td>95.9</td></tr>
<tr><td>A4</td><td>196</td><td>179</td><td>1</td><td>4</td><td>97.4</td><td>91.3</td></tr>
<tr><td>A5</td><td>211</td><td>190</td><td>3</td><td>3</td><td>97.2</td><td>90.0</td></tr>
<tr><td>A6</td><td>210</td><td>193</td><td>4</td><td>2</td><td>97.1</td><td>91.9</td></tr>
<tr><td>A7</td><td>211</td><td>194</td><td>5</td><td>1</td><td>97.2</td><td>91.9</td></tr>
<tr><td>A8</td><td>65</td><td>62</td><td>0</td><td>0</td><td>100</td><td>95.4</td></tr>
<tr><td>Total</td><td>1286</td><td>1175</td><td>24</td><td>10</td><td>97.4</td><td>91.4</td></tr>
</table>

Figure 3.

<table>
<tr><th colspan="7">Classification Performance for The ODR Algorithm</th></tr>
<tr><th colspan="7">Hard Dimensionality Reduction Used</th></tr>
<tr><th>Image</th><th>Total_Joints</th><th>CC</th><th>Misses</th><th>False_Alarm</th><th>%GB</th><th>%CC</th></tr>
<tr><td>A1</td><td>196</td><td>167</td><td>9</td><td>0</td><td>95.7</td><td>85.2</td></tr>
<tr><td>A3</td><td>197</td><td>189</td><td>1</td><td>0</td><td>99.5</td><td>95.9</td></tr>
<tr><td>A4</td><td>196</td><td>178</td><td>1</td><td>4</td><td>97.4</td><td>90.8</td></tr>
<tr><td>A5</td><td>211</td><td>192</td><td>3</td><td>2</td><td>97.6</td><td>91.0</td></tr>
<tr><td>A6</td><td>210</td><td>194</td><td>4</td><td>2</td><td>97.1</td><td>92.4</td></tr>
<tr><td>A7</td><td>211</td><td>191</td><td>5</td><td>1</td><td>97.2</td><td>90.5</td></tr>
<tr><td>A8</td><td>65</td><td>63</td><td>0</td><td>0</td><td>100</td><td>96.9</td></tr>
<tr><td>Total</td><td>1286</td><td>1174</td><td>23</td><td>9</td><td>97.5</td><td>91.3</td></tr>
</table>

Figure 4.

<table>
<tr><th colspan="7">Classification Performance for The ODR Algorithm</th></tr>
<tr><th colspan="7">Minimum Distance Classifier Emulation</th></tr>
<tr><th>Image</th><th>Total_Joints</th><th>CC</th><th>Misses</th><th>False_Alarm</th><th>%GB</th><th>%CC</th></tr>
<tr><td>A1</td><td>196</td><td>135</td><td>9</td><td>2</td><td>94.4</td><td>68.8</td></tr>
<tr><td>A3</td><td>197</td><td>151</td><td>2</td><td>1</td><td>98.5</td><td>76.6</td></tr>
<tr><td>A4</td><td>196</td><td>136</td><td>13</td><td>9</td><td>88.8</td><td>69.4</td></tr>
<tr><td>A5</td><td>211</td><td>148</td><td>15</td><td>5</td><td>90.5</td><td>70.1</td></tr>
<tr><td>A6</td><td>210</td><td>146</td><td>9</td><td>15</td><td>88.6</td><td>69.5</td></tr>
<tr><td>A7</td><td>211</td><td>139</td><td>9</td><td>14</td><td>89.1</td><td>65.8</td></tr>
<tr><td>A8</td><td>65</td><td>52</td><td>2</td><td>1</td><td>95.4</td><td>80.0</td></tr>
<tr><td>Total</td><td>1286</td><td>907</td><td>59</td><td>47</td><td>91.8</td><td>70.5</td></tr>
</table>

Figure 5.

Classification Performance for Old Minimum Distance Classifier					
Twelve Member Class List					
Weight File 3					
Image	MFC	Good/Bad_%	Miss_%	False_Alarm_%	Correct_Class_%
A1	A1	96.4	2.6	1.0	74.5
A2	A2	83.2	9.6	7.1	67.0
A3	A3	95.9	2.0	2.0	76.6
A4	A4	91.8	2.6	5.6	66.8
A5	A5	81.0	12.3	6.6	65.9
A6	A6	89.0	5.7	5.2	61.9
A7	A7	88.2	0.9	10.9	57.8
B1	B1	91.1	7.1	1.8	71.4
B2	B2	92.2	0.0	7.8	79.7
B3	B3	77.3	10.6	12.1	66.7
B4	B4	98.4	0.0	1.6	60.9
B5	B5	84.4	12.5	3.1	71.9
B6	B6	82.8	3.1	14.1	73.4
AVG	---	88.6	5.3	6.1	68.8

Figure 6.

Classification Performance for The ODR Algorithm						
Soft Dimensionality Reduction Used						
Image	Total_Joints	CC	Misses	False_Alarm	%GB	%CC
K1	100	75	15	1	84.0	75.0
K2	100	89	10	3	87.0	89.0
Total	200	164	25	4	85.5	82.0

Figure 7.

Subimage Features Ranked by Usefulness				
HDR = Hard Dimensionality Reduction				
	Feature_#	Metric	Feature_#	Metric
Rank	With HDR	With HDR	Without HDR	Without HDR
0	6	803.3	6	827.9
1	4	715.5	4	734.3
2	5	562.0	5	564.5
3	1	485.3	1	493.0
4	0	349.3	0	352.2
5	22	336.1	22	339.2
6	10	278.12	10	280.3
7	13	197.5	13	197.9
8	12	165.4	12	166.4
9	2	162.2	2	163.9
10	17	149.7	17	160.0
11	18	149.5	18	159.6
12	20	128.0	20	129.7
13	3	118.3	3	120.1
14	11	89.9	11	90.7
15	23	77.5	23	79.3
16	19	66.9	19	68.2
17	7	60.0	16	61.6
18	16	59.8	7	61.4
19	15	53.0	15	54.9
20	25	52.8	21	54.7
21	21	52.2	25	54.0
22	24	47.2	24	48.4
23	26	41.2	26	42.5
24	8	39.6	8	40.7
25	9	34.8	9	36.7
26	14	31.9	14	32.8

Figure 8.

Distance in Unmodified 27 Dimentional Space
(units = percent of average distance)

	a	b	c	d	e	f	h	i	n
a	0	78	100	85	102	134	93	80	97
b		0	32	46	85	70	94	28	120
c			0	57	94	59	113	32	143
d				0	54	77	106	47	120
e					0	91	138	80	149
f						0	146	68	178
h							0	103	82
i								0	133
n									0

TABLE A.

Distance in Rotated and Scaled Domain
(units = percent of average distance)

	a	b	c	d	e	f	h	i	n
a	0	70	80	58	54	80	68	42	61
b	104	0	21	42	76	44	72	44	72
c	59	17	0	37	69	40	91	29	86
d	57	39	40	0	36	28	82	34	80
e	76	89	96	45	0	70	136	49	128
f	54	42	32	18	33	0	92	29	91
h	67	44	46	52	73	59	0	45	44
i	442	331	496	470	425	500	555	0	437
n	291	160	199	327	432	290	60	274	0

TABLE B.

Figure 9. (A) Minimum Distance and (B) ODR Algorithm Distance Tables.