A General Methodology for Simultaneous Representation and Discrimination of Multiple Object Classes

Ashit Talukder and David Casasent

Department of Electrical and Computer Engineering, Laboratory for Optical Data Processing
Carnegie Mellon University, Pittsburgh, PA 15213

Abstract

In this paper we address a new general method for linear and nonlinear feature extraction for simultaneous representation and classification. We call this approach the maximum representation and discrimination feature (MRDF) method. We develop a novel nonlinear eigenfeature extraction (NLEF) technique to represent data with closed-form solutions and use it to derive a nonlinear MRDF algorithm. Results of the MRDF on synthetic databases are shown and compared with results from standard Fukunaga-Koontz transform and Fisher discriminant function methods. The method is also applied to an automated product inspection problem (discrimination) and for classification and pose estimation of two similar objects under 3-D aspect angle variations (representation and discrimination).

Keywords: Classification, discrimination (nonlinear), feature extraction (nonlinear), pattern recognition, pose estimation, principal component analysis (nonlinear and closed-form), product inspection.

1 Introduction

Feature extraction for signal/image representation is an important issue in data processing. 1-D and 2-D signals typically have a very large number of data points. Typical speech signals contain around $10^2 - 10^3$ samples, and images have around $N \times N$ pixels where N varies from 32 to 1024. Processing of such high-dimensional data is time-consuming. Typically, features are extracted from the data using syntactic [1], statistical [2], stochastic [3] or correlation [4] methods. Syntatic pattern recognition methods use a set of structural elements or primitives within a scene or object and their interrelationships to interpret the data [5]. Statistical and stochastic methods use probability distribution measures [2] or stochastic models [3] to estimate, detect and classify data. Such methods are good for modeling of textures and regions with pseudorandom gray-scale variations. Correlation filters have been proven to be useful for distortion-invariant detection of objects [4]. Often, however, the data within an image needs to be reduced to a lower-dimensional space for purposes of analysis, while preserving information in the lower-dimensional transformed space. Such an application is known as data representation. Principal component analysis (PCA) or the Karhunen-Loeve (KL) transform [6] is useful for such applications. Neural network (NN) solutions to PCA learning have also been suggested[7,8]. PCA however can optimally represent only a single class at a time. It does not ensure discriminatory information. Others [9] have noted that standard PCA methods fail to discriminate between significantly different types of objects (such as cars, chairs, human faces, and human bodies) and have suggested a most discriminating feature (MDF) measure to allow for separation of features between classes in the transformed space. In discrimination cases, the Fukunaga Koontz (FK) transform [10] is useful, since it computes orthonormal basis functions that can represent a single class best while least representing all the other classes. However, the FK transform is not well suited for applications where several classes of objects need to be both represented well and separated. The Fisher linear discriminant [11] is also well suited for separating image/signal data for different objects or classes by a linear transformation. Distance Classifier Correlation Filters (DCCFs) [12] also reduce intra-class variations and increases inter-class separation as the Fisher measure does; but they do so by operating in the frequency plane.

In this paper, we develop a new feature extraction method called the maximum representation and discrimination feature (MRDF) that allows for simultaneous representation of each class and separation between the different classes. The measure used in the MRDF allows each class to have multiple clusters in the image/signal data, while the MDF, PCA, Fisher and DCCF methods assume only one cluster per class.

Linear transforms have been widely used in many signal processing applications for several reasons. It has been proved that the linear matched spatial filter is the best detection filter for a single object when it is corrupted by white noise [13]. Linear techniques for detection and classification are attractive since they are easy to design and typically have closed form solutions. Linear distortion-invariant filters [4] exist that can achieve recognition in the face of distortions such as aspect variations, in-plane rotation, etc. Linear methods such as the Fisher discriminant, FK transform, MDF, and linear PCA are only optimal when the data is Gaussian and symmetrically distributed about the mean. Such methods extract information from the second-order correlations in the data (covariance matrix). Therefore, such techniques implicitly assume probability density functions (PDF) that are unimodal, and are symmetrically distributed about the mean. While such linear methods are optimal when Gaussian or wide sense stationarity is assumed, they are not necessarily the best for complex data that are asymmetrically distributed or not described by Gaussian PDFs. It has been shown that many signals in the real world are inherently non-symmetric [14,15] and that linear PCA is incapable of representing such data [14]. For such cases, nonlinear transforms are necessary.

Several techniques have been suggested to capture higher-order statistics from data [14,16-19], but all methods are iterative (this requires ad hoc parameter selection, large training set sizes can be needed, and generalization and convergence problems can arise). Thus, our nonlinear non-iterative solution is of importance. Nonlinear PCA[17,20-24] is an extension of linear PCA. A linear combination of the input data can be passed through a nonlinearity (sigmoid, tanh, etc.)[17] and a set of linear weights can be computed iteratively[7] using information from the higher-order input correlations (this handles asymmetric data [21]). The disadvantage of this approach is its slow convergence to the optimal weights when the Hebbian learning rule is used[20]. The error measure used is also of concern. When the error measure is quadratic (mean square error criterion) it has been shown[17,20] that the weight update rule is similar to a Hebbian rule, and hence a linear PCA is obtained. It has been shown that when the error function increases less than quadratically, the resultant system is less susceptible to noise [17]. Thus, non-quadratic error measures and iterative stochastic gradient algorithms have been used[17,20]. An iterative nonlinear neural network (NN) method [18], independent component analysis (ICA), has been used for blind separation of source signals from their linear or nonlinear mixtures. A cost function is created using the moments of the PDFs of the signal mixtures, and this is minimized using a backpropagation type of neural network to obtain the desired nonlinear transformation. All iterative techniques to determine weights can have convergence problems to the globally optimal solution [20] and *have limitations on the rank of the decision surfaces compared to our method* (as we will show).

We discuss the concepts behind linear PCA and then present our linear MRDF technique in Sect. 2 and test results using it on several databases (Sect. 3). Our nonlinear eigenfeature extraction (NLEF) technique for representation is presented in Sect. 4, and theoretically compared to nonlinear PCA methods. These NLEF ideas are then used to develop the nonlinear MRDF in Sect. 5; initial test results using it are presented in Sect. 6.

2 Linear MRDF (Maximum Representation And Discrimination Feature)

We follow the following notation. Vectors and matrices are represented by lower and uppercase letters with underlines ($\underline{x}$ and $\underline{X}$). Random vectors are lowercase and bold ($\mathbf{x}$), and random variables and scalars are simply lowercase. When we consider a single random vector, $\mathbf{x}$, its linear transformation by $\underline{\phi}$ yields the random variable $y = \underline{\phi}^T \mathbf{x}$, and we use the expectation operator $E(\mathbf{x})$ to denote the expected value of the random vector $\mathbf{x}$. Terms E_n with different subscripts n are *not* the expectation operator. When considering a set of sample data vectors $\{\underline{x}\}$, we describe them by the sample data matrix $\underline{X} = [\underline{x}_1 \; \underline{x}_2 \; ... \; \underline{x}_N]$.

2.1 Linear PCA and Its Limitations

Principal component analysis (PCA) is a transformation primarily used for representing high-dimensional data in fewer dimensions such that the maximum information about the data is present in the transformed space. The linear PCA is now summarized to define our notation.

Given a N dimensional random vector, $\mathbf{x}$, whose mean, $\underline{\mu} = E[\mathbf{x}]$, and covariance matrix $\underline{C} = E[\mathbf{x}\mathbf{x}^T] - \underline{\mu\mu}^T$ are known, the objective is to find the transformation $\underline{\mathbf{\Phi}}_M$ such that the M-dimensional random vector $\mathbf{y}_M = \underline{\mathbf{\Phi}}_M^T\mathbf{x}$ contains maximal information about $\mathbf{x}$. We find $\underline{\mathbf{\Phi}}_M = [\underline{\phi}_1\ \underline{\phi}_2 ... \underline{\phi}_M]$ such that some error criterion is minimized, where $\underline{\mathbf{\Phi}}_M$ is an $N \times M$ matrix composed of M orthonormal vectors, $\underline{\phi}_1\ \underline{\phi}_2 ... \underline{\phi}_M$, each of size N. Each element y_m of $\mathbf{y}_M$ is the **projection** of the input random vector $\mathbf{x}$ onto the basis vector $\underline{\phi}_m$. To find $\underline{\mathbf{\Phi}}_M$, it is customary to minimize the mean square error between $\mathbf{x}$ and the approximation $\hat{\mathbf{x}} = \underline{\mathbf{\Phi}}_M\mathbf{y}_M$, i.e., we select $\underline{\mathbf{\Phi}}_M$ such that $E[(\mathbf{x} - \hat{\mathbf{x}})^2] = E[(\mathbf{x} - \underline{\mathbf{\Phi}}_M\mathbf{y}_M)^2]$ is minimized. This solution for $\underline{\mathbf{\Phi}}_M$ satisfies $\underline{C\mathbf{\Phi}}_M = \underline{\mathbf{\Phi}}_M\underline{\Lambda}$, where $\underline{\Lambda}$ is a diagonal matrix whose elements are the eigenvalues of the covariance matrix $\underline{C}$. The columns of the $\underline{\mathbf{\Phi}}_M$ solution are the M eigenvectors of $\underline{C}$ with the largest eigenvalues. This solution minimizes the mean squared representation error. It can also be shown that this solution also maximizes the variance of the output random vector in the $\mathbf{y}_M = \underline{\mathbf{\Phi}}_M^T\mathbf{x}$ transformed space, i.e. it maximizes $E[\mathbf{y}_M^T\mathbf{y}_M] = \sum_{m=1}^{M} \underline{\phi}_m^T \underline{C}\underline{\phi}_m$. This is useful for insight into the basis vectors chosen in several synthetic 2-D examples. An attractive property of PCA is that it compresses the maximum average energy in the full data set in only $M \leq N$ samples; and, in image transmission problems, PCA provides the minimum distortion rate among all unitary transforms when the transmitted signal is Gaussian.

For a given problem where a number of samples $\{\underline{x}_p\}, p = 1, ...P$ from the random vector $\mathbf{x}$ are available, the covariance matrix is computed in the following manner. Each data sample is arranged as a column of a data matrix $\underline{X} = [\underline{x}_1\ \underline{x}_2\ ..\ \underline{x}_P]$, the sample mean is $\underline{\hat{\mu}} = 1/P \sum_{p=1}^{P} \underline{x}_p$, and the sample covariance matrix is $\underline{\hat{C}} = 1/P(\underline{XX}^T) - \underline{\hat{\mu}\hat{\mu}}^T$.

The KL transform or PCA is *optimal only for representation of data.* The FK transform is *well suited for discrimination only* since it merely represents one class best and the other class worst; when a class has multiple clusters it does not perform well, and it is *not well-suited for simultaneous representation and discrimination* in the same feature space. The Fisher linear discriminant also has associated problems when classes contain multiple clusters. The MDF (most discriminating feature) method [9] is similar to the Fisher discriminant, since it estimates a transform that best separates one class from the rest of the classes. The disadvantages of this are that: only one linear transform per class can be computed and thus the maximum number of MDFs equals the number of classes; it does not consider representation of classes; it is ill-suited for cases when a class has multiple clusters; and separating one class from all other classes (all other classes are thus considered as a macro-class) is not necessarily the best approach. Our approach computes a basis set that best represents each class and at the same time best separates the classes in the new feature space. We first define two measures, one for representation (Sect. 2.2) and one for discrimination (Sect. 2.3), and combine the two measures to derive the MRDF (Sect. 2.4). This initial development considers only linear transforms and features; we later (Sect. 4) extend this to nonlinear cases.

2.2 Measure for Best Representation

We wish to determine an M-dimensional transform, i.e. M orthogonal basis function vectors $\underline{\phi}_m$. For simplicity of explanation, we consider two classes, where $\mathbf{x}_1$ and $\mathbf{x}_2$ are two random vectors corresponding to classes 1 and 2. From PCA concepts, the transform that best represents class 1 maximizes $\underline{\Phi}_M^T \underline{C}_1 \underline{\Phi}_M = \sum_{m=1}^{M} \underline{\phi}_m^T \underline{C}_1 \underline{\phi}_m$, where $\underline{C}_1 = E[\mathbf{x}_1 \mathbf{x}_1^T] - \underline{\mu}_1 \underline{\mu}_1^T$ is the covariance matrix of class 1 and $\underline{\mu}_1 = E[\mathbf{x}_1]$ is the mean of class 1. Similarly, the transform that best represents class 2 maximizes the measure $\underline{\Phi}_M^T \underline{C}_2 \underline{\Phi}_M = \sum_{m=1}^{M} \underline{\phi}_m^T \underline{C}_2 \underline{\phi}_m$. When P samples $\{\underline{x}_{1p}\}, p = 1, ...P$, from class 1 are available, the covariance matrix is computed as $\widehat{\underline{C}}_1 = 1/P(\underline{X}_1 \underline{X}_1^T) - \widehat{\underline{\mu}}_1 \widehat{\underline{\mu}}_1{}^T$, where the P samples are the columns of the data matrix $\underline{X}_1 = [\underline{x}_{11} \ \underline{x}_{12} \ .. \ \underline{x}_{1P}]$ and the sample mean is $\widehat{\underline{\mu}}_1 = 1/P \sum_{p=1}^{P} \underline{x}_{1p}$. The sample covariance matrix for class 2 is computed in a similar manner.

For a transform to represent both classes equally well, we will require it to maximize the new measure

$$E_R = \sum_{m=1}^{M} \underline{\phi}_m^T \underline{C}_1 \underline{\phi}_m \ + \ \sum_{m=1}^{M} \underline{\phi}_m^T \underline{C}_2 \underline{\phi}_m. \tag{1}$$

When there are L classes the desired transform maximizes $E_R = \sum_{l=1}^{L} \sum_{m=1}^{M} \underline{\phi}_m^T \underline{C}_l \underline{\phi}_m$.

2.3 Measure for Best Discrimination

For discrimination, we want the projections (transformed values) for each class to be separated. Measures such as the square of the difference in projections of the class means (used in the Fisher measure) are not robust enough to handle a variety of class distributions. If one class has several clusters, use of the mean does not provide a good measure of separation. We use the squared difference of the projected values for each sample in one class versus the projected values for all of the other class projections. In other words, the separation measure we use is the average squared difference for all such projection values for two classes. *We expect this to be a better measure of separation, since it does not use the means of the class projections.*

We denote the projection of class 1 on basis vector m by $y_{m1} = \underline{\phi}_m^T \mathbf{x}_1$; similarly, the projection of class 2 on basis vector m is $y_{m2} = \underline{\phi}_m^T \mathbf{x}_2$. For the projections of the two classes to be best separated for each of the M basis vectors, we desire that $E_{Dm} = E[(y_{m1} - y_{m2})^2]/E[(y_{m1} - \mu_{y_{m1}})^2 + (y_{m2} - \mu_{y_{m2}})^2]$, where $1 \leq m \leq M$, be large (or maximized) while minimizing the spread of each class projection, i.e. we maximize the mean squared separation between the projections of classes 1 and 2 on each basis vector $\underline{\phi}_m$ while minimizing the sum of the projections of the class covariances. The numerator in the discrimination measure can be further simplified by writing it in terms of the input random vectors as $\underline{\phi}_m^T E[(\mathbf{x}_1 - \mathbf{x}_2)(\mathbf{x}_1 - \mathbf{x}_2)^T]\underline{\phi}_m = \underline{\phi}_m^T \underline{R}_{12} \underline{\phi}_m$, where $\underline{R}_{12} = E[(\mathbf{x}_1 - \mathbf{x}_2)(\mathbf{x}_1 - \mathbf{x}_2)^T]$. For cases where several data samples from each class are available, the average of the squared distance from each sample in class 1 to each sample in class 2 is used. If P samples from class 1, $\{\underline{x}_{1p}\}$ with $p = 1, ..., P$, and Q samples from class 2, $\{\underline{x}_{2q}\}$ with $q = 1, ..., Q$, are available, then the estimate of $\underline{R}_{12}$ used is $\hat{\underline{R}}_{12} = 1/(PQ) \sum_{p=1}^{P} \sum_{q=1}^{Q} (\underline{x}_{1p} - \underline{x}_{2q})(\underline{x}_{1p} - \underline{x}_{2q})^T$.

The steps involved in obtaining the numerator in E_{Dm} follow. Form the class projections $y_{m1} = \mathbf{x}_1^T \underline{\phi}_m$ and $y_{m2} = \mathbf{x}_2^T \underline{\phi}_m$. Calculate $(y_{m1} - y_{m2})^T = \underline{\phi}_m^T (\mathbf{x}_1 - \mathbf{x}_2)$, from which we have $(y_{m1} - y_{m2}) = (\mathbf{x}_1^T - \mathbf{x}_2^T)\underline{\phi}_m = (\mathbf{x}_1 - \mathbf{x}_2)^T \underline{\phi}_m$. The numerator in the separation measure is then obtained $E[\ (y_{m1} - y_{m2})^2\ ] = E[\ (y_{m1} - y_{m2})^T\ (y_{m1} - y_{m2})] = \underline{\phi}_m^T E[(\mathbf{x}_1 - \mathbf{x}_2)(\mathbf{x}_1 - \mathbf{x}_2)^T]\underline{\phi}_m$, as noted above. Therefore, to determine the best $\underline{\phi}_m$ set for discrimination of classes 1 and 2, we maximize

$$E_D = \sum_{m=1}^{M} E_{Dm} = \sum_{m=1}^{M} \frac{\underline{\phi}_m^T \underline{R}_{12} \underline{\phi}_m}{\underline{\phi}_m^T (\underline{C}_1 + \underline{C}_2) \underline{\phi}_m}. \tag{2}$$

For the multi-class case (L classes), the separation measure to be maximized is $E_D = \sum_{m=1}^{M} [\underline{\phi}_m^T (\sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \underline{R}_{ij}) \underline{\phi}_m] / [\underline{\phi}_m^T (\sum_{k=1}^{L} \underline{C}_k) \underline{\phi}_m]$.

2.4 Derivation of the MRDF

To design a set of orthonormal basis functions $\underline{\phi}_m$ that can provide *both representation and classification*, we combine the two measures in Eq. (1) and Eq. (2). We introduce a factor k to denote the degree to which representation and discrimination is desired. When only discrimination between classes is desired we assign $k = 0$, and when only representation of the classes is required we use $k = 1$; in general, intermediate k values are preferable. We combine the two measures to maximize

$$E_{RD} = \sum_{m=1}^{M} \frac{\underline{\phi}_m^T [k(\underline{C}_1 + \underline{C}_2) + (1-k)\underline{R}_{12}] \underline{\phi}_m}{\underline{\phi}_m^T [k\underline{\mathbf{I}} + (1-k)(\underline{C}_1 + \underline{C}_2)] \underline{\phi}_m} \tag{3}$$

where $0 \leq k \leq 1$ and $\underline{\mathbf{I}}$ is the identity matrix. Note that when $k = 1$ in Eq. (3), we obtain $E_{RD} = \sum_{m=1}^{M} \underline{\phi}_m^T [(\underline{C}_1 + \underline{C}_2)] \underline{\phi}_m / \underline{\phi}_m^T \underline{\phi}_m$ which is the condition for best representation of two classes (Eq. (1)); when $k = 1$ in Eq. (3), we obtain $E_{RD} = \sum_{m=1}^{M} \underline{\phi}_m^T [\underline{R}_{12}] \underline{\phi}_m / \underline{\phi}_m^T (\underline{C}_1 + \underline{C}_2) \underline{\phi}_m$ which is the discrimination measure of the MRDF (Eq. (2)). Setting the derivative of Eq. (3) with respect to one basis function $\underline{\phi}_m$ to zero, we obtain

$$[k\underline{\mathbf{I}} + (1-k)(\underline{C}_1 + \underline{C}_2)]^{-1} [k(\underline{C}_1 + \underline{C}_2) + (1-k)\underline{R}_{12}] \underline{\phi}_m = \lambda_m \underline{\phi}_m \text{ for all } m = 1, 2..., M. \tag{4}$$

The $\underline{\phi}_m$ solutions to Eq. (4) are the solutions to an eigenvalue-eigenvector equation. The MRDF basis functions $\underline{\phi}_m$ are the M eigenvectors corresponding to the M largest eigenvalues of $[k\underline{\mathbf{I}} + (1-k)(\underline{C}_1 + \underline{C}_2)]^{-1} [k(\underline{C}_1 + \underline{C}_2) + (1-k)\underline{R}_{12}]$.

The MRDF has several properties that are of theoretical and practical interest. We prove in an associated publication[25] that when the class distributions are Gaussian with equal covariance matrices, the discriminatory MRDF (k=0) is the same as the Bayes classifier, and when one class is present and representation is desired (k=1), the MRDF reduces to the linear PCA solution. The MRDF technique can produce as many as N basis functions (as many as the dimensionality of the sample vectors). All classes are considered simultaneously (pairwise), as compared to the macro-class approach (MDF, FK and Fisher linear discriminant). This *MRDF approach allows multiple clusters in a single class*, due to its separation measure in Eq. (2) that computes the average squared difference between the projections of the two classes. The MRDF achieves *simultaneous representation and classification*, and therefore is very well-suited for applications where both inter-class performance (discrimination and classification) and intra-class performance (representation, or recognition of the different versions of the members in each class) are required. Both properties are useful in general pattern recognition problems such as digital library searches, pose estimation of objects in robotics and automatic target recognition, automatic reconaissance, part and material handling and inspection, active vision, face recognition, etc.

3 Initial Linear MRDF Results

To present data and decision surfaces visually, we consider two synthetic problems using two features with two classes of data (there are 2000 samples in each class; we used 1000 samples from each class as the training set and 1000 samples from each class as the test set). The objective of these tests is to illustrate the performance of the MRDF for representation and discrimination of a higher-dimensional feature set (N features) in a lower-dimensional feature space (M features) where $M < N$. For insight into the performance of the MRDF, we transform these 2-D features (N=2) into a 1-D feature space (M=1). The advantage of such a transform is evident when $N \gg M$. The data in each class are 2-D random variables described by a single Gaussian or by a mixture of two Gaussian PDFs; the latter case results in two clusters of data for one class in feature space. For each case, we show the input data samples from the test set for class 1 (o) and class 2 (+); only every tenth sample is shown to allow better presentation. We also show the 1-D linear MRDF, FK and Fisher discriminant functions in the input feature space that were determined using samples from the training set. To classify the test data, it is projected onto one of the discriminant functions and then classified using a nearest neighbor rule (based on the class of the closest training set sample). As a measure of the discriminating ability of each of the linear discriminant functions, we calculate the mean squared difference in the projection values for the two classes (this is the separation measure used in the MRDF). We also compare the representation error for each method. This is computed as follows. The projection of a 2-D sample $\underline{x}_1$ onto a basis vector $\underline{\phi}_1$ is $y_1 = \underline{\phi}_1^T \underline{x}_1$. By the orthonormal property of the transforms, the reconstructed sample is $\widehat{\underline{x}_1} = \underline{\phi}_1 y_1$. The representation error is the mean squared error between the original and reconstructed samples.

For all three discriminant functions, we used only one basis function. In calculating the MRDF discriminant (basis) function, we used $k = 0.5$ in Eq. (4) to assign equal weights for representation and discrimination. To calculate the FK discriminant (basis) function, we used the eigenvector that best represented one class (the FK eigenvector with the largest eigenvalue).

Fig. 1a shows the Case 1 data (o and +). Class 1 has a PDF that is a single Gaussian with unequal variances oriented at 90^o. Class 2 is a mixture of two Gaussians with unequal variances oriented at 0^o (horizontal) and 75^o; it thus results in two clusters in Fig. 1a. The three linear discriminant functions are also shown in Fig. 1a. The input data are projected normally onto the different linear discriminant functions (LDFs). The projections of many samples onto the Fisher LDF overlap (the Fisher measure fails when a class has more than one cluster, since it uses the mean of each class as a separation measure). The FK vector selected was the one that represents class 2 best and class 1 least (the spread of the class 2 projections on the FK vector in Fig. 1a is seen to be larger than the spread of the class 1 projections). Recall from Sect. 2 that the variance of the projection will be larger for the class that is best represented. For the projections onto the FK vector, the bottom cluster in class 2 is separated well and there is some degree of overlap in the projections of the class 1 cluster and the top cluster in class 2. Our linear MRDF discriminant separates all three clusters well; although the variance of the projection of the lower class 2 cluster onto the MRDF is large, this does not degrade discrimination. The dominant eigenvector for the MRDF measure is such that the projections of class 1 and 2 do not have significant overlap (discrimination), and the spread (representation) of class 1 and class 2 is large. To compare the results of the MRDF, FK and Fisher, we carried out a nearest neighbor classification of the test set projections on the three basis vectors. The results are shown in Table 1. The MRDF has the best classification rate P_C (the precentage of test set data correctly classified). The mean squared representation error and the mean squared separation in the test set projections for each of the three LDFs are listed in Table 1 for completeness. As seen, the MRDF has a larger separation and a smaller representation error than the other LDFs.

Fig. 1b shows Case 2 data; now, each class is a mixture of two Gaussians with different means and oriented at various angles. With two clusters per class, this is a difficult classification problem for a single linear discriminant function. The Fisher linear discriminant fails in this case since the means of classes 1 and 2 nearly coincide. The FK basis vector chosen was the one that best represented class 1 (it thus chooses a vector such that the spread of the projections of class 1 is larger than the spread of the class 2 projections). As seen, the sample projections from the top clusters in classes 1 and 2 overlap; this results in its low P_C (Table 1). The MRDF basis vector (the dominant eigenvector in Eq. (4)) separates the class projections well, and represents class 2 well. The mean squared representation error, the mean squared separation in the class projections and P_C are again best for the MRDF as seen in Table 1.

We also tested the MRDF on a database of real objects (two cars with 72 aspect views of each taken from a 25^o depression angle). These are objects 19 and 6 in the COIL-20 database [26]. Fig 2 shows several aspect views of each object. Each image is stretched in x and y until it fills one dimension of a 128×128 pixel frame (this is often used in robotic vision) and the gray values in each image are stretched to cover the full 0-255 range. For each object, the 72 different aspect views at 5^o intervals covering a 360^o aspect range are divided into 36 even aspect angles (0^o, 10^o, ...) as the training set and the 36 odd aspect views (5^o, 15^o, ...) as the test set. We designed MRDF basis vectors for discrimination ($k = 0$ in Eq. (4)) using the 36 training images. We chose the two best (dominant) MRDF basis vectors. A nearest neighbor classifier gave perfect $P_C = 100\%$ on the test set (see Table 2).

We now consider the use of this database and the MRDF features when we wish to *determine both the pose and the class of the input test objects.* This is a case where both representation of each object class in feature space (for pose estimation) and discrimination between different object classes in feature space (to determine object class) is desired. This is useful in active vision applications where objects or tools on a factory production line need to be classified and their pose estimated for use by robotic manipulators in grasping, inspection, and assembly. In this case, we used a new feature space trajectory (FST) representation [27] for different distorted versions of an object in the MRDF feature space. In the FST representation, different object aspect views are points in feature space. Points associated with adjacent aspect views are joined by straight lines to produce a distinct FST (a piecewise linear curve) in feature space for all distorted aspect views of an object; different FSTs are produced for different objects. To classify an input test object, it is mapped to a point in feature space. The closest FST determines its class estimate and the closest line segment on that FST provides an estimate of its pose (we interpolate between the aspect angle values at the vertex end points of this line segment). We used the MRDF basis functions for the two car objects with $k = 0.5$ as the feature space and constructed the FST for each car using 36 training images per class (36 vertices on each FST). We then estimated the class and pose of the test set of 72 input object aspect views. For the pose estimation problem, we must first determine the number of MRDF basis vectors to use. We included the dominant basis function MRDF vectors until these contained about 40% of the energy in the transformed training set samples. We are developing other methods to select the best number of features using nearest neighbor leave-one-out tests [28] and separation of different parts of an FST [29].

For our initial results, we present classification scores P_C and the average pose $\widehat{\theta}_{avg}$ estimation error of the test set for different numbers of MRDF features (Table 2). For each feature space choice, new FSTs are produced and the class and pose of the test set are estimated. To test the discrimination capabilities of the MRDF using an FST, we first set $k = 0$, and then analyzed the classification results using two or more dominant MRDFs. Perfect classification (P_C= 100%) was obtained using 2, 3 and 4 MRDFs with $k = 0$ (only discrimination), and the corresponding average pose estimation errors $\widehat{\theta}_{avg}$ were 37.7°, 29.04° and 16.21° respectively. We noted earlier however, that we expect intermediate values of k to provide both joint discrimination and representation. We therefore used $k = 0.5$ and tested the performance of the MRDF. The results are summarized in Table 2. Using 2, 3 and 4 MRDFs, we obtained P_C= 93.06%, 100%, and 100%, with corresponding pose estimate errors of 34.2°, 2.68°, and 2.65°. Use of 4 discriminating MRDFs ($k = 0$) gave P_C=100% and $\widehat{\theta}_{avg} = 16.21°$ proving that intermediate values of k are necessary for joint discrimination and representation. Use of 4 KL features (2 per class) in contrast gave P_C=100% and $\widehat{\theta}_{avg} = 37.2°$. When we used four FK features calculated from ten KL features (five per class), we obtained P_C=98.6% and $\widehat{\theta}_{avg} = 6.2°$ (Table 2). Therefore, our MRDFs are observed to provide better performance than standard techniques such as the KL and the FK using fewer numbers of features.

4 Nonlinear Eigenfeature Extraction (NLEF) Algorithm

All prior nonlinear PCA methods are iterative and thus do not have closed-form solutions. Here, we discuss a new algorithm (NLEF) to produce a nonlinear transformation *with a closed-form solution.* We discuss this algorithm for the case of best representation; we will later (Sect. 5) extend it to the case of both representation and discrimination. This method uses higher-order correlation information in the input data and thus is useful for representation of asymmetrically distributed data. This method is then shown to provide advantages compared to prior nonlinear PCA methods.

To formulate the problem and solution, we consider a random vector $\mathbf{x} = [x_1 \; x_2 \; x_3 \; ....x_N]^T$ that models the distorted versions of an object or variations in the features of one object. For representation, our objective is to find a nonlinear transformation (or a set of nonlinear transforms), $y = f(\mathbf{x})$, on $\mathbf{x}$ that optimally represents the random vector $\mathbf{x}$ in a reduced dimensionality space. We consider nonlinear transforms that are polynomial mappings of the input. In this initial work, we only consider a second-order polynomial mapping (quadratic transform). The quadratic transform can be written as $y = \sum_n a_n x_n + \sum_n \sum_m a_{mn} x_m x_n$ or in matrix form as $y = \mathbf{x}^T \underline{A} \mathbf{x} + \underline{b}^T \mathbf{x}$ where $\underline{A}$ is a matrix and $\underline{b}$ is a vector. We can also write a quadratic mapping as a linear transform $y = \underline{\phi}^T \mathbf{x}_H$ on the higher-order and higher-dimensional vector

$\mathbf{x}_H = [x_1\ x_2\ x_N\ x_1x_1\ x_1x_2\ ....\ x_1x_N\ x_2x_2\ x_2x_3\ ....\ x_Nx_N]^T$ that contains higher-order terms in the original input data $\mathbf{x}$. This vector is of dimension $H = N + N(N+1)/2$ with N linear terms, and $N(N+1)/2$ unique non-linear cross-product terms. We use this last formulation of a quadratic mapping to derive our NLEF, since *it allows for a closed-form solution in the* $\mathbf{x}_H$ *space that is quadratic in the original* $\mathbf{x}$ *space.* Note that this can be generalized easily to yield polynomial mappings of any arbitrary order.

We now address determining the M quadratic transforms on the input such that the random vector is well represented. The coefficients of this transform, $\{\underline{\phi}_m\}, m = 1, ..., M$, are arranged in a $H \times M$ matrix $\underline{\Phi}_M = [\underline{\phi}_1\ \underline{\phi}_2\ ...\ \underline{\phi}_M]$. The M quadratic transforms on $\mathbf{x}$ yield a new random vector $\mathbf{y}_{HM} = \underline{\Phi}_M^T \mathbf{x}_H$. Note that so far we have formulated the quadratic transform in terms of a linear transform. In PCA (Sect. 2.1), the linear transform that best represents $\mathbf{x}$ maximizes the variance of the output random vector. Similarly, for the quadratic mapping, we choose the set of orthonormal vectors $\underline{\phi}_m$ that maximize the variance of the output random vector, i.e. we maximize

$$\Sigma_{m=1}^{M} \underline{\phi}_m^T \underline{C}_H \underline{\phi}_m + \Sigma_{m=1}^{M} \lambda_m (\underline{\phi}_m^T \underline{\phi}_m - 1), \tag{5}$$

where the λ_m are the Lagrange multiplier coefficients to be chosen and the summation is over the basis function vectors $\underline{\phi_m}$ that are used ($M < H$). The second term in Eq. (5) ensures that the $\underline{\phi_m}$ are orthonormal (they are orthogonal by definition). In Eq. (5) we now use the higher-order covariance matrix $\underline{C}_H$ of the random vector $\mathbf{x}_H$; this matrix contains higher-order correlation terms (we consider only terms up to the fourth order in this quadratic transform case), i.e. $\underline{C}_H = E(\mathbf{x}_H \mathbf{x}_H^T) - \underline{\mu}_H \underline{\mu}_H^T$ where $\underline{\mu}_H = E(\mathbf{x}_H)$. The transformation matrix $\underline{\Phi}_M$ must satisfy

$$\underline{C}_H \underline{\Phi}_M = \underline{\Phi}_M \underline{\Lambda}. \tag{6}$$

This corresponds to an eigenvalue-eigenvector equation where $\underline{\Lambda}$ is a diagonal matrix whose diagonal elements are the eigenvalues of $\underline{C}_H$ and the columns of $\underline{\Phi}_M$ are the eigenvectors of $\underline{C}_H$. Therefore, the M quadratic transforms $\underline{\Phi}_M = [\underline{\phi}_1\ \underline{\phi}_2 ...\ \underline{\phi}_M]$ that best represent the random vector $\mathbf{x}$ are the M dominant eigenvectors associated with the M largest eigenvalues of the higher-order covariance matrix $\underline{C}_H$.

When several samples $\{\underline{x}_p\}, p = 1, ..., P$, of the random vector $\mathbf{x}$ are available, the sample higher-order covariance matrix $\underline{\widehat{C}}_H$ is used. To calculate $\underline{\widehat{C}}_H$, the higher-order vectors $\{\underline{x}_{pH}\}$ and the estimate $\underline{\widehat{\mu}}_H = 1/P(\sum_{p=1}^{P} \underline{x}_{pH})$ of $\underline{\mu}_H$ are used and the sample higher-order covariance matrix is found to be $\underline{\widehat{C}}_H = 1/P(\sum_{p=1}^{P} \underline{x}_{pH}\underline{x}_{pH}^T) - \underline{\widehat{\mu}}_H\underline{\widehat{\mu}}_H^T$.

This section has provided a solid foundation for the choice of the nonlinear quadratic (higher-order) transforms that optimally represent a class of data. We refer to this algorithm as the nonlinear eigenfeature (NLEF) extraction algorithm. *The NLEF has a closed form solution; thus iterative solutions and their problems are avoided.* It extracts and uses higher-order correlation information present in the input data. This new method automatically finds the best nonlinear transforms, produces orthogonal features, and orders them in terms of their order of importance for representation. By omitting data with smaller eigenvalues, outliers in the training data can be automatically omitted. Conversely, the linear PCA method considers only second-order correlations in the input data. Note that if a linear transformation provides a better spread (variance) in the data than a nonlinear transformation, the NLEF automatically sets the higher-order coefficients to zero and the NLEF then becomes the linear PCA. Therefore, the linear PCA is a special case of the NLEF.

The NLEF overcomes the various problems that are associated with nonlinear PCA (NLPCA) iterative solutions. However, the NLEF also has other advantages as we now discuss. NLPCA techniques typically compute a linear weighted combination of the input data and then pass this through a nonlinearity. This form of nonlinear transformation has some limitations compared with the nonlinear polynomial transformation we use, as we now discuss. The NLPCA computes a transformation $y = g(\underline{w}^T\mathbf{x})$, where $\mathbf{x}$ is the input vector, $\underline{w}$ is the vector of NLPCA weights and $g()$ is a nonlinear function. The Taylor's series expansion of the NLPCA is $y = c_0 + c_1(\underline{w}^T\mathbf{x}) + c_2(\underline{w}^T\mathbf{x})^2 + ...$, where the number of terms in the expansion depends on the "smoothness" of the function $g(x)$. If we neglect the third and higher-order terms (i.e. we consider the quadratic case), the expansion is $y = c_0 + c_1(\underline{w}^T\mathbf{x}) + c_2(\mathbf{x}^T\underline{w}\underline{w}^T\mathbf{x})$, where the quadratic term in the expansion is $\mathbf{x}^T\underline{w}\underline{w}^T\mathbf{x} = \mathbf{x}^T\underline{A}'\mathbf{x}$ where $\underline{A}' = \underline{w}\underline{w}^T$. A general quadratic transformation is $z = \mathbf{x}^T\underline{A}\mathbf{x}$, where the matrix $\underline{A}$ determines the shape of the quadratic mapping (hyperellipsoid, etc.) and the rank of $\underline{A}$ (or equivalently its positive definiteness) determines the number of dimensions in which the decision surface or mapping can depart from a plane. *When $\underline{A}$ is rank deficient, it implies that the mapping is a plane and is not curved in some dimensions.* The matrix $\underline{A}' = \underline{ww^T}$ is analogous to $\underline{A}$ in a general quadratic transformation; as seen, the rank of $\underline{A}'$ is one, since it is the outer-product of two vectors, each of which has rank one. In other words, the quadratic mapping produced by the NLPCA can produce curved surfaces in only one dimension. Thus the NLPCA provides very limited higher-order transformations due to the rank-deficiency of its quadratic mapping. For one quadratic mapping $\underline{\phi}$, our quadratic NLEF ouput $y_H = \underline{\phi}^T \mathbf{x}_H$ is $y_H = \sum_i \sum_j a_{ij} x_i x_j + \sum_i b_i x_i = \mathbf{x}^T \underline{A}_\phi \mathbf{x} + \underline{\mathbf{b}}^T \mathbf{x}$, where $\underline{A}_\phi$ is a symmetric matrix. The vectors $\mathbf{x}$ are N-dimensional and in the original space. The matrix $\underline{A}_\phi$ is $N \times N$ and is symmetric. Its maximum rank is thus N, the dimension of the input vector $\mathbf{x}$. If the data requires it, the rank of $\underline{A}_\phi$ will be N (full rank) and thus the representation basis function will curve in all N dimensions of the input space. Thus, *our NLEF quadratic transformation algorithm produces a quadratic transformation matrix of higher rank than the nonlinear PCA can.* Hence, it is expected to provide more general quadratic transforms.

However, the number of parameters to be estimated in our quadratic NLEF is $\mathcal{O}(N)^2$, while in other NLPCA methods the number of unknown parameters is the same as the dimensionality N of the data ($\mathcal{O}(N)$). The on-line computation time for the NLEF transform thus increases quadratically with the size of the input. When the input data are images (N is large), we thus use a nonlinear transform with no cross terms and a resultant computation time of ($\mathcal{O}(N)$). In this paper, we discuss quadratic transforms for feature input data only (with low N), in which the on-line computation time for a quadratic transform is not high.

5 Nonlinear MRDF

The linear MRDF can only provide linear transformations. Using the NLEF algorithm, we now develop a nonlinear MRDF that provides *nonlinear transformations for both representation and discrimination.*

We consider a second-order nonlinear MRDF for two classes of data described by the random vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. We create the augmented data vectors, $\mathbf{x}_{1H}$ and $\mathbf{x}_{2H}$ for each class; these contain first and second-order terms as before. We desire to determine the M best transforms $\underline{\Phi}_M = [\underline{\phi}_1\underline{\phi}_2 ... \underline{\phi}_M]$ such that the transformed data $\mathbf{y}_{1H} = \underline{\Phi}_M^T\mathbf{x}_{1H}$ and $\mathbf{y}_{2H} = \underline{\Phi}_M^T\mathbf{x}_{2H}$ are both separated and still representative of the input data. The $\underline{\phi}_m$ are constrained to be orthogonal. For representation, we maximize

$$E_R = \sum_{m=1}^{M} \underline{\phi}_m^T \underline{C}_{1H}\underline{\phi}_m + \sum_{m=1}^{M} \underline{\phi}_m^T \underline{C}_{2H}\underline{\phi}_m, \tag{7}$$

where $\underline{C}_{1H} = E[\mathbf{x}_{1H}\mathbf{x}_{1H}^T] - \underline{\mu}_{1H}\underline{\mu}_{1H}^T$ is the *higher-order covariance matrix* of class 1 and $\underline{\mu}_{1H} = E[\mathbf{x}_{1H}]$ is the mean of the augmented vector $\mathbf{x}_{1H}$; similarly, $\underline{C}_{2H}$ is the *higher-order covariance matrix* of class 2. The separation measure to be maximized is

$$E_S = \sum_{m=1}^{M} E_{Sm} = \sum_{m=1}^{M} \frac{\underline{\phi}_m^T \underline{R}_{12H}\underline{\phi}_m}{\underline{\phi}_m^T(\underline{C}_{1H} + \underline{C}_{2H})\underline{\phi}_m} \tag{8}$$

where $\underline{R}_{12H} = E[(\mathbf{x}_{1H} - \mathbf{x}_{2H})(\mathbf{x}_{1H} - \mathbf{x}_{2H})^T]$.

For the MRDF, we want to compute the best features that can jointly represent and discriminate between the input classes. The weight assigned for representation is k and for discrimination is $(1-k)$. The measure to be maximized for the nonlinear MRDF is thus

$$E_{RS} = \sum_{m=1}^{M} \frac{\underline{\phi}_m^T[k(\underline{C}_{1H} + \underline{C}_{2H}) + (1-k)(\underline{R}_{12H})]\underline{\phi}_m}{\underline{\phi}_m^T[k\underline{\mathbf{I}} + (1-k)(\underline{C}_{1H} + \underline{C}_{2H})]\underline{\phi}_m}. \tag{9}$$

Differentiating Eq. (9) with respect to the nonlinear functions $\underline{\Phi}_M$, the solution must satisfy

$$[k\underline{\mathbf{I}} + (1-k)(\underline{C}_{1H} + \underline{C}_{2H})]^{-1}[k(\underline{C}_{1H} + \underline{C}_{2H}) + (1-k)(\underline{R}_{12H})]\underline{\Phi}_M = \underline{\Phi}_M\underline{\Lambda}. \tag{10}$$

This corresponds to an eigenvalue-eigenvector equation as before. Therefore, the M best nonlinear MRDF transformation coefficients $\underline{\phi}_m$ are the M dominant eigenvectors corresponding to the M largest eigenvalues of $[k\underline{\mathbf{I}} + (1-k)(\underline{C}_{1H} + \underline{C}_{2H})]^{-1}[k(\underline{C}_{1H} + \underline{C}_{2H}) + (1-k)(\underline{R}_{12H})]$. If we wish to only discriminate between the classes we assign $k = 0$, if only representation is needed we select $k = 1$; to simultaneously represent and discriminate data we use intermediate values of k.

6 Nonlinear MRDF Results

We expect our nonlinear MRDFs to provide improved P_C for the data in Fig. 1, but this has not been verified since our linear MRDFs gave $P_C \simeq 99\%$ (Table 1). To test the nonlinear MRDF, we first consider two 2-D (two features) synthetic data cases, in which the decision surfaces can be visualized. In these cases, higher-order decision surfaces will be necessary to separate the two classes, specifically quadratic decision surfaces using our quadratic (second-order nonlinear) MRDF. We consider the best single MRDF solution for each of these two cases.

A single linear discriminant function (LDF) cannot separate many configurations of data. One example is the case when one class is completely surrounded by another class. In such cases quadratic or higher-order decision surfaces are needed. A number of LDFs can be used to approximate a quadratic decision surface, however the use of many LDFs increases computation and design costs. Conversely, our quadratic nonlinear MRDF inherently produces quadratic decision surfaces using fewer discriminant functions ($\underline{\phi}_m$), because these discriminant functions are nonlinear functions of the input feature data. In each case considered, each class has 2000 samples (1000 samples in the training set and 1000 in the test set) denoted by + and o with every fifth sample from the test set shown for better presentation. In Case 1 (Fig. 3a), class 1 samples have an uniform distribution within a certain radius from the origin and they are surrounded by class 2 samples. To separate these clusters, a circular quadratic decision surface is necessary. The best single quadratic MRDF transform was determined using the samples from the training set; the resultant decision surface (the MRDF function) is shown as a solid line in Fig. 3a. The decision surface produced by our nonlinear MRDF is seen to provide good class separation. Samples on either side of the decision surface are assigned to different classes. A P_C of 99.9% was obtained on the test set with this nonlinear MRDF.

In Case 2 (Fig. 3b), class 1 is Gaussian distributed and centered about the origin; class 2 is a mixture of two Gaussians such that it forms two clusters, one on each side of the class 1 cluster. The decision surface produced by the single best (dominant) quadratic MRDF is elliptical (solid line in Fig. 3b). Data inside the elliptical decision surface is assigned to class 1 and data outside it is assigned to class 2. In this case, the classification accuracy of 97.5% was obtained on the test set with the nonlinear (quadratic) MRDF.

We next tested the nonlinear MRDF on a product inspection problem. The problem involves classification of pistachio nuts as clean or infested based on features extracted from real-time X-Ray images of the pistachios. Infestations include worm or insect feeding damage, mold, rancidity, decay, etc. External images of the pistachios provide insufficient information about the quality of a nut (Fig. 4, top), while real-time X-ray images provide internal details from which classification is possible (Fig. 4, bottom).

We preprocessed arbitrarily oriented touching nuts to extract individual nuts; we then morphologically removed the shell edge and airgap (between the nutmeat and the shell) from the image as detailed elsewhere [30]. The resultant nutmeat-only images were then used for classification. We extracted four histogram features [31] from the histogram of the nutmeat-only image and four features from the histogram of an edge-enhanced version of this image. Infested nuts tend to be darker, and rougher than clean ones (Fig. 4, bottom left). The histogram features from the non-edge enhanced image capture information about the gray-level distribution in the nutmeat (dark or light gray values), and the features from the edge image capture texture information (roughness or smoothness). These eight features were input to our nonlinear MRDF algorithm. In this case, our nonlinear MRDF *algorithm is applied to input features rather than images.*

We used 605 clean and 686 infested pistachio nut images; the clean nuts were divided into a training set of 303 nuts and a test set of 302 nuts and the infested nuts were divided into a training set of 344 nuts and a test set of 342 nuts. The eight histogram features were computed for the training and test set images, and nonlinear quadratic MRDF features were calculated from these using only training set data. Since only discrimination is necessary (not representation), we used k=0 in Eq. (9). A piecewise-quadratic neural network (PQNN) classifier [32] was then trained using these nonlinear MRDF features calculated from the training set, and then tested on the test set. The steps involved in classifying each pistachio nut are shown in Fig. 5. The best prior results [31] on this database using eight histogram features (rather than our nonlinear MRDF features calculated from the same histogram features) gave 88% correct classification on the test set (Table 3). The nonlinear MRDF was found to (Table 3) improve this to 90.4% using only one quadratic MRDF, and to 91.3% using the twelve dominant discriminating quadratic MRDFs. This improvement is significant. The purpose of these tests is to show that higher-order correlation information exists in real data and that our nonlinear MRDF can locate such information.

7 SUMMARY

We have presented a new linear feature extraction technique for representation and discrimination (MRDF) and a new nonlinear extension of it (NLEF) that has a closed-form solution. A theoretical comparison of the NLEF and the NLPCA shows that the NLEF has more general decision surfaces of higher rank compared to the NLPCA. Tests of our new techniques on synthetic and real data showed good results, demonstrated the need for higher-order decision surfaces, that higher-order correlation information exists in real data, and that such information is useful for classification. These ideas have a wide variety of applications in digital library searches and indexing, automated surveillance, active vision, material handling and inspection, and associated areas in computer vision.

ACKNOWLEDGEMENTS

This material is based upon work supported by the Cooperative State Research, Education, and Extension Service, U.S. Department of Agriculture, under a recent agreement.

References

1. R. C. Gonzalez. *Syntactic Pattern Recognition.* Addison-Wesley Pub. Com., Reading, Mass., 1978.

2. S. Aeberhard, D. Coomans, and O. De Vel. Comparative analysis of statistical pattern recognition methods in high dimensional settings. *Pattern Recognition*, 27(8):1065–1077, Aug. 1992.

3. B. S. Manjunath and R. Chellappa. Unsupervised texture segmentation using markov random field models. *IEEE Trans. PAMI*, 13(5):478–482, May 1991.

4. D. Casasent and S. Ashizawa. SAR detection, recognition and clutter rejection with new MINACE filters. *Opt. Engr.*, October 1997. to be published.

5. P. Rothman. Syntactic pattern recognition. *AI Expert*, 7(10):40–51, Oct. 1992.

6. H. Hotelling. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychology*, 24:417–441 and 498–520, 1933.

7. E. Oja. A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15:267–273, 1982.

8. J. Hertz et. al. *Introduction to the theory of neural computation.* Addison Wesley, 1991.

9. D. L. Swets and J. J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. PAMI*, 18(8):831–836, Aug. 1996.

10. K. Fukunaga and W. L. G. Koontz. Applications of the Karhunen-Loeve expansion to feature selection and ordering. *IEEE Trans. Comp.*, C-19:917–923, 1970.

11. R. A. Fisher. *Contributions to Mathematical Statistics.* John Wiley, New York, 1950.

12. A. Mahalanobis, B. V. K. Vijaya Kumar, and S. R. F. Sims. Distance classifier correlation filters for multi-class target recognition. *Applied Optics*, 35(17):3127–3133, 1996.

13. H. L. Van Trees. *Detection, Estimation and Modulation Theory.* Wiley, New York, 1968.

14. G. Taylor and S. Coombes. Learning higher order correlations. *Neural Networks*, 6:423–427, 1993.

15. H. B. Barlow. *Possible principles underlying the transmission of sensory messages*, pages 217–234. MIT Press, Cambridge, Mass., 1961.

16. G. Deco and W. Brauer. Nonlinear higher order statistical decorrelation by volume-conserving neural networks. *Neural Networks*, 8(4):525–535, 1995.

17. J. Karhunen and J. Joutsensalo. Representation and separation of signals using nonlinear PCA type learning. *Neural Networks*, 7(1):113–127, 1994.

18. G. Burel. Blind separation of sources: A nonlinear neural algorithm. *Neural Networks*, 5:937–947, 1992.

19. G. F. Ramponi, G. L. Sicuranza, and W. Ukovich. A computational method for design of 2-D nonlinear Volterra filters. *IEEE Trans. on Circuits and Systems*, 35(9):1095–1102, Sept. 1988.

20. J. Karhunen and J. Joutsensalo. Generalizations of principal component analysis, optimization problems, and neural networks. *Neural Networks*, 8(4):549–562, 1995.

21. W. S. Softky and D. M. Kammen. Correlations in high dimensional or asymmetric data sets: Hebbian neuronal processing. *Neural Networks*, 4(3):337–348, Nov. 1991.

22. J. Karhunen and Joutsensalo. Learning of robust principal component subspace. In *Proc. Intl. Joint Conf. Neural Networks*, pages 2409–2412, Oct. 1993.

23. J. Karhunen and Joutsensalo. Nonlinear generalizations of principal component learning algorithms. In *Proc. Intl. Joint Conf. Neural Networks*, pages 2599–2602, Oct. 1993.

24. J. Joutsensalo and J. Karhunen. Nonlinear multilayer principal component type subspace learning algorithms. In C. A. Kamm et al, editor, *Neural Networks for signal processing III*, pages 68–77. IEEE Press, New York, 1993.

25. Ashit Talukder and David Casasent. Joint recognition and discrimination in nonlinear feature space. In *Proc. SPIE: Intelligent Robots and Computer Vision XVI*, volume 3208, Oct. 1997.

26. S. A. Nene, S. K. Nayar, and H. Murase. *Columbia image object library (COIL-20). Technical Report CUCS-006-96.* Dept. of Computer Science, Columbia University, New York, NY 10027, 1996.

27. D. Casasent and L. Neiberg. Classifier and shift-invariant ATR neural networks. *Neural Networks*, 8(7/8):1117–1129, 1995.

28. D. Casasent and R. Shenoy. Feature space trajectory for distorted-object classification and pose estimation in SAR. *Optical Engineering*, October 1997.

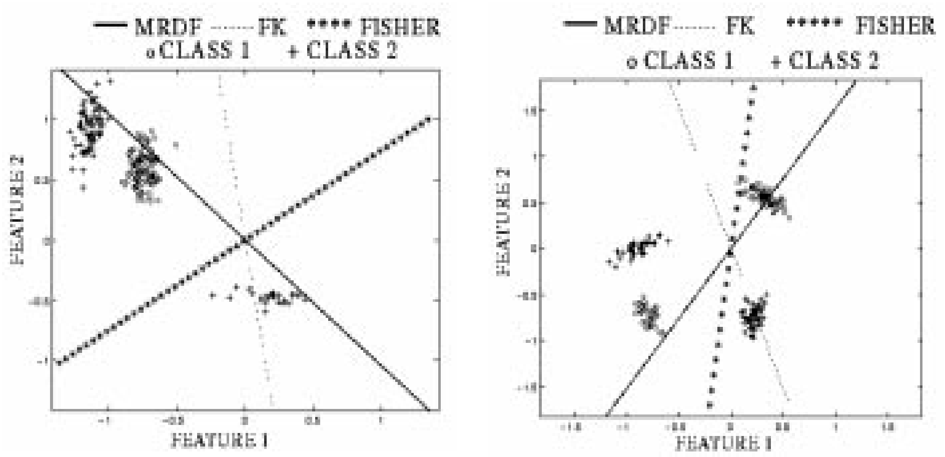
29. D. Casasent, L. Neiberg, and M. Sipe. FST distorted object representation for classification and pose estimation. *Accepted for publication in Optical Engineering*, (This Issue).

30. A. Talukder and D. Casasent. Automated segmentation and feature extraction of product inspection items. In *Proc. SPIE*, volume 3073, pages 96–107, Apr 1997.

31. D. Casasent, M. Sipe, T. Schatzki, and P.M. Keagy. Neural net classification of x-ray pistachio nut data. In *Optics in Agriculture, Forestry, and Biological Processing II, Proc. SPIE*, volume 2907, pages 217–229, Nov 1996.

32. D. Casasent and S. Natarajan. A classifier neural net with complex-valued weights and square-law nonlinearities. *Neural Networks*, 8(6):989–998, 1995.

List of Figures



(a) (b)

Figure 1: Test set data for Case 1 (a) and Case 2 (b) of 2-D random variables from two classes and the basis functions generated by the MRDF, FK, and Fisher linear discriminants.

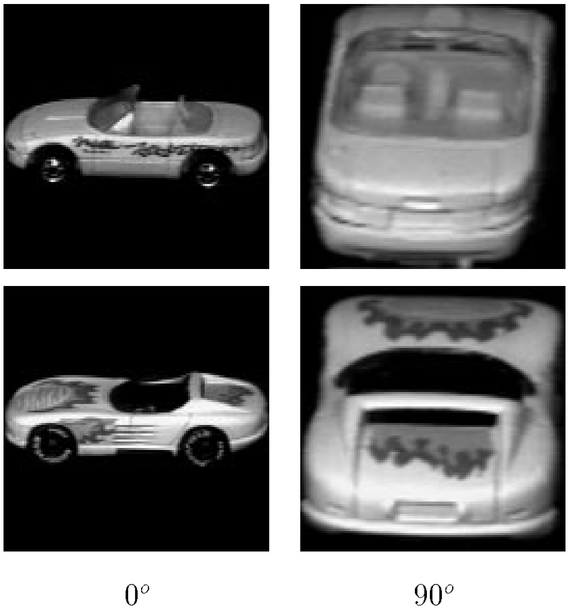
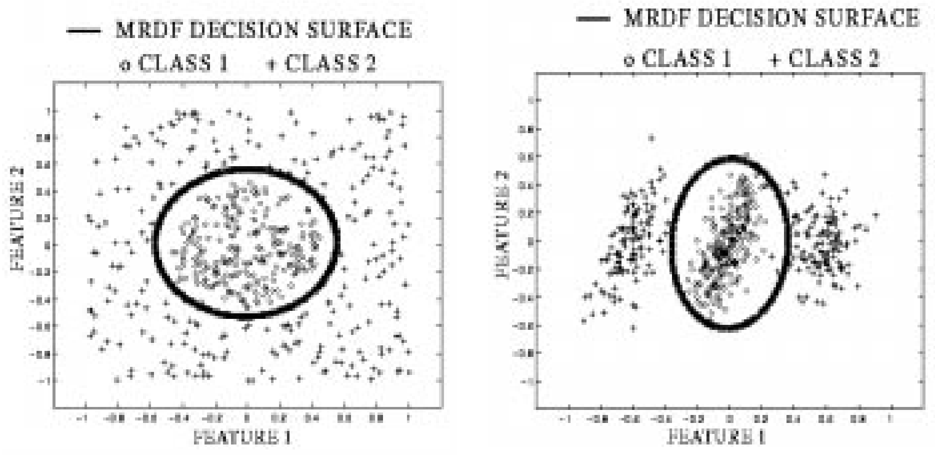


Figure 2: Aspect views of two similar cars (top object 19, bottom object 6)



(a) (b)

Figure 3: Test set data for Case 1 (a) and Case 2 (b) of 2-D classes and the quadratic MRDF functions produced.

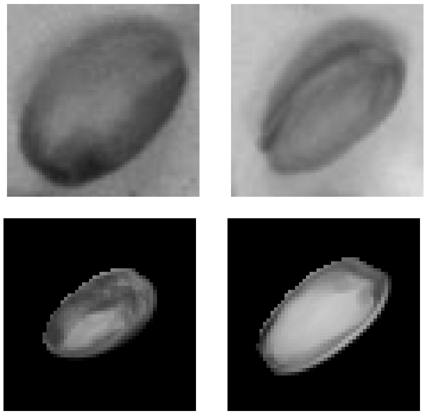


Figure 4: Visible (top) and X-ray (bottom) nut images; infested (left) and clean (right).

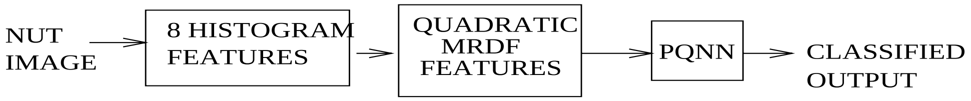


Figure 5: Block diagram of the quadratic MRDF classifier for pistachio nut inspection.

Basis Function	Inter-Class Separation		MS Representation error		P_C Results	
	Case 1	Case 2	Case 1	Case 2	Case 1	Case 2
MRDF	0.71	0.95	0.19	0.47	98.8%	99.2%
FK	0.48	0.485	0.74	0.53	82.1%	54.0%
Fisher	0.42	0.81	0.8	0.55	42.4%	58.3%

Table 1: Inter-class separation, mean-square representation error and classification accuracy using MRDF, FK and Fisher discriminant functions.

Features Used	P_C	Avg Pose
		Est. Error ($\widehat{\theta}_{avg}$)
2 MRDFs (k=0.5)	93.06%	34.2°
3 MRDFs (k=0.5)	100%	2.68°
4 MRDFs (k=0.5)	100%	2.65°
2 MRDFs (k=0)	100%	37.7^{o}
4 KL (2 KL/Class)	100%	37.2°
4 FK of 10 KLs (5 KL/Class)	98.6%	6.2°

Table 2: Classification and pose estimation of two cars using MRDFs.

Features Used	P_C (Train)	P_C (Test)
1 Quadratic MRDF	92.9%	90.4%
12 Quadratic MRDFs	91.7%	91.3%
8 Histogram Features	89.3%	88.0%

Table 3: Pistachio nut classification results using nonlinear MRDF and histogram features.