Best-Bases Feature Extraction Algorithms for Classification of Hyperspectral Data

Shailesh Kumar, Joydeep Ghosh, and Melba M. Crawford, *Member, IEEE*

***Abstract*—Due to advances in sensor technology, it is now possible to acquire hyperspectral data simultaneously in hundreds of bands. Algorithms that both reduce the dimensionality of the data sets and handle highly correlated bands are required to exploit the information in these data sets effectively. We propose a set of best-bases feature extraction algorithms that are simple, fast, and highly effective for classification of hyperspectral data. These techniques intelligently combine subsets of adjacent bands into a smaller number of features. Both top-down and bottom-up algorithms are proposed. The top-down algorithm recursively partitions the bands into two (not necessarily equal) sets of bands and then replaces each final set of bands by its mean value. The bottom-up algorithm builds an agglomerative tree by merging highly correlated adjacent bands and projecting them onto their Fisher direction, yielding high discrimination among classes. Both these algorithms are used in a pairwise classifier framework where the original C-class problem is divided into a set of $\binom{C}{2}$ two-class problems.**

The new algorithms 1) find variable length bases localized in wavelength, 2) favor grouping highly correlated adjacent bands that, when merged either by taking their mean or Fisher linear projection, yield maximum discrimination, and 3) seek orthogonal bases for each of the $\binom{C}{2}$ two-class problems into which a C-class problem can be decomposed. Experiments on an AVIRIS data set for a 12-class problem show significant improvements in classification accuracies while using a much smaller number of features. Moreover, the proposed methodology facilitates the extraction of valuable domain knowledge regarding the importance of certain bands for discriminating specific groups of classes.

I. Introduction

DISCRIMINATION among different landcover types using remotely sensed data is an important application of pattern classification. Advances in sensor technology have made possible the simultaneous acquisition of hyperspectral data in more than two hundred individual bands, where each spectral band covers a fixed range of wavelengths. Although hyperspectral data are becoming more widely available, algorithms that exploit the potential of the narrow bands while being computationally tractable, are needed. The response from each pixel in the hyperspectral image can be represented by a D-dimensional *ordered vector* or "signal" x that is characterized by highly correlated spectrally adjacent bands. A feature extractor for hyperspectral data should utilize these properties while obtaining "signatures" for discriminating among different landcover types or classes. Feature selection methods based on Bhattacharya distance [1] and feature extraction methods based on Karhunen–Loéve (K–L) transforms [2] have been proposed to reduce the number of features used in classification. However, while feature selection methods ignore the fact that adjacent bands are generally correlated, feature extraction methods do not utilize the ordering information between adjacent bands at all. Moreover, only one set of features is typically used for labeling all the classes. While selection of a single "global" set of features leads to inefficient utilization of information from multispectral data, it is even more problematic in hyperspectral data analysis as it defeats the primary motivation for acquiring hyperspectral data: to characterize the unique class specific responses of individual land cover types.

A new algorithm has been developed that extracts class-specific features for classification. First, a C-class problem is decomposed into $\binom{C}{2}$ two-class problems. For each pair of classes, features are extracted independently, and a Bayesian classifier is learned on this feature space. The results of all the $\binom{C}{2}$ classifiers are then combined to determine the class label of a pixel. This paper focuses on the feature extraction component of the algorithm for two-class problems. These techniques involve merging adjacent subsets of bands to yield a small number of highly discriminatory features. Two algorithms for finding such feature spaces are proposed: 1) a fast, greedy top-down approach that recursively partitions a set of adjacent bands into two sets and merges each final group of bands into its mean [3], and 2) a bottom-up agglomerative clustering approach that merges adjacent highly correlated bands by projecting them onto their Fisher direction that maximizes the separation between two classes [4].

The paper is organized as follows. In Section II, the characteristics of hyperspectral data are reviewed, and the pairwise classifier framework is presented. The new top-down and bottom-up algorithms are described in Sections III and IV, respectively. Experimental results highlighting the efficacy of the best-bases algorithms in improving classification accuracy, reducing the feature space, and extracting domain knowledge are presented in Section V.

II. Background

Hyperspectral sensors simultaneously acquire information in hundreds of spectral bands. A hyperspectral image is essentially a three-dimensional (3-D) array $I(p, q, d)$, where (p, q) denotes a pixel location in the image, and d denotes a spectral

Manuscript received July 24, 2000; revised March 3, 2001. This work was supported by ARO Contracts DAAD19-99-1-0012, DAAG55-98-1-0230, and DAAG55-98-1-0287, NSF Grant ECS-9900353, the Texas Advanced Technology Research Program (CSRA-ATP-009), and the NASA Solid Earth and Natural Hazards Program (NAG5-7660(LA)).

S. Kumar and J. Ghosh are with the Department of Electrical and Computer Engineering, The University of Texas, Austin, TX 78712 USA (e-mail: skumar@lans.ece.utexas.edu).

M. M. Crawford is with the Center for Space Research, The University of Texas, Austin, TX 78712 USA.

Publisher Item Identifier S 0196-2892(01)05489-4.

0196–2892/01$10.00 ©2001 IEEE

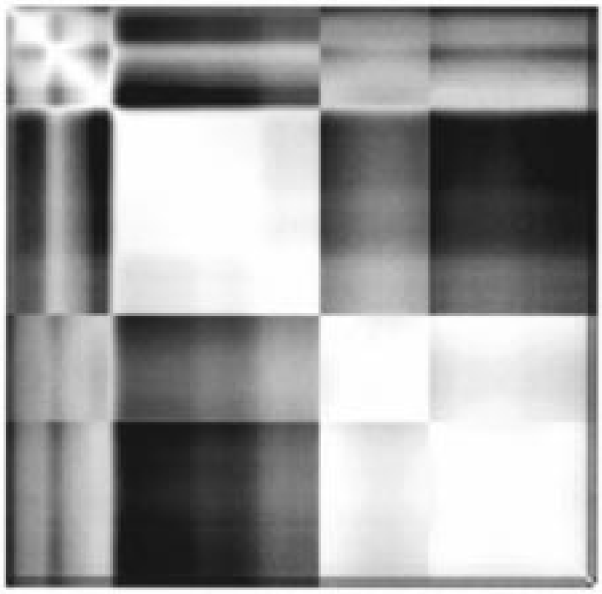

Fig. 1. Correlation matrix of AVIRIS data set: Adjacent bands typically exhibit higher correlation (white).

band (wavelength). The value stored at $I(p, q, d)$ is the response (reflectance or emittance) from the pixel (p, q) at a wavelength corresponding to spectral band d. The input space for a hyperspectral data (classification problem) is an ordered vector of real numbers of length D, the number of spectral bands, where bands that are spectrally "near" each other tend to be highly correlated (see Fig. 1). The goal of a feature extraction algorithm for hyperspectral classification is to obtain mappings of the information from the original set of bands that characterize the spectral signatures of the classes that are being discriminated.

A. Related Work in Feature Extraction for Hyperspectral Data

Analysis of hundreds of simultaneous channels of data necessitates the use of either feature selection or extraction algorithms prior to classification. Feature selection algorithms for hyperspectral classification are costly, while feature extraction methods based on K–L transforms, Fisher's discriminant, or Bhattacharya distance cannot be used directly in the input space because the covariance matrices required by all these methods are highly unreliable given the ratio of the amount of training data to the number of input dimensions.

Lee and Landgrebe [1] proposed methods for *feature extraction based on decision boundaries* for both Bayesian and neural network based classifiers. The data is projected normally to the decision boundary found by learning a classifier in the input space itself. Jia and Richards [2]–[5] proposed a feature extraction technique based on the segmented principal components transformation (SPCT) for two-class problems involving hyperspectral data. The K–L transform is applied to each group of adjacent highly correlated bands, and a subset of principal components from each group is selected based on their discrimination capacity. Recently, Jimenez and Landgrebe [6] proposed a feature reduction algorithm for hyperspectral data based on projection pursuit [7]. Using Bhattacharya distance as the projection index, a linear transformation of the input space is sought. The projection matrix is constrained to partition all the bands into smaller groups of adjacent bands and project each group into a single feature.

B. Desired Properties of a Hyperspectral Feature Extractor

The three main properties of the desired feature extraction algorithm identified in [3] are as follows.

1) ***Class dependence:*** Different subsets of classes are best distinguished by different feature sets. Hence, feature extractors for specific groups of classes should be determined separately. Most classifiers seek only one set of features that distinguishes among all the classes simultaneously. This not only increases the complexity of the potential decision boundary, but also requires a large number of features and reduces the interpretability of the resulting features.

2) ***Ordering constraint:*** The characteristics that bands are ordered and adjacent bands are correlated should be exploited by the feature extraction algorithm. A Fisher or K–L transform on all the bands does not treat the input vector as a signal and hence is not ideal for hyperspectral data feature extraction. Both the SPCT based feature extractor [2] and the projection pursuit based algorithm [6] utilize the ordering and locality properties of hyperspectral data. In general, any transformation should involve adjacent groups of bands.

3) ***Discriminating transforms:*** The transformations should try to maximize discrimination among classes, and thus use class label information. The K–L transform, used in SPCT, for example, is suited for preserving the variance in the data, but does not necessarily increase the discriminatory capacity of the feature space. Use of Fisher discriminant or Bhattacharya distance (as used in decision boundary feature extractors) is therefore more desirable for feature extraction.

In order to satisfy Property 1, a pairwise classifier with a class pair specific feature extractor is used. This pairwise classifier architecture is described in the following section.

C. The Pairwise Classification Framework

The conventional approach to $C > 2$ classification problems is to first transform an input space $\mathcal{I}$ (which is the hyperspectral signal here) to a feature space $\mathcal{F}$ in which the discrimination among all the C classes in class set $\mathbf{\Omega}$ is high then to use a single classifier Φ that distinguishes all the C classes simultaneously. In this paper, however, we use a Bayesian pairwise classifier (BPC) framework [8], [9] that we developed previously for classification problems with a moderately large number of classes. In the BPC framework shown in Fig. 2, a C-class problem is first decomposed into a set of $\binom{C}{2}$ two-class problems for all pairs $(\omega_i, \omega_j), 1 \leq i < j \leq C$. Each of the two-class problems is solved independently, and their results are combined to obtain the result for the original C-class problem.

A customized feature extraction approach for each pair of classes is especially advantageous in remote sensing applications, where extraction of domain knowledge about specific class characteristic, is as important as reducing the feature

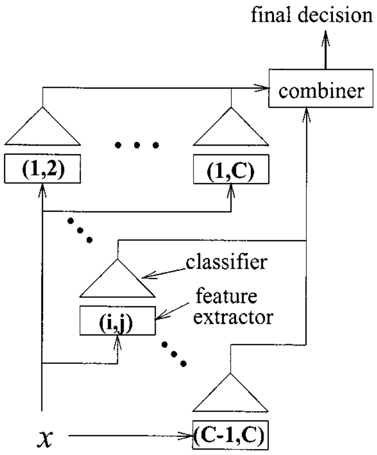

Fig. 2. Pairwise classifier architecture: $\binom{C}{2}$ pairwise classifiers with respective feature selectors.

space and improving classification accuracy [8], [9]. For classification of hyperspectral data, this framework is particularly advantageous as one can essentially "create" artificial hyperspectral sensors for discriminating each pair of classes from the original.

Each classifier ϕ_{ij} in the BPC architecture has an associated feature extractor denoted by $\psi_{ij}\,:\,\mathcal{I}\rightarrow\mathcal{F}_{ij}$ that transforms an input $\mathbf{x}\in\mathcal{I}$ into a feature vector $\psi_{ij}(\mathbf{x})\in\mathcal{F}_{ij}$ specific to class pair (ω_i,ω_j). The output of ϕ_{ij} is an estimate of the posterior probability $P_{ij}\left(\omega_i|\psi_{ij}(\mathbf{x})\right)$ (and $P_{ij}\left(\omega_j|\psi_{ij}(\mathbf{x})\right)= 1-P_{ij}\left(\omega_i|\psi_{ij}(\mathbf{x})\right)$). Each ϕ_{ij} is implemented as a maximum aposterior Bayesian classifier that models the probability density functions $p\left(\psi_{ij}(\mathbf{x})|\omega_k\right)$, $k=i,j$ as:[1]

$$\hat{p}\left(\mathbf{y}|\omega_k\right)=\frac{1}{\sqrt{(2\pi)^d\left|\Sigma_k^{(i,j)}\right|}}\times\exp\left[-\frac{1}{2}\left(\mathbf{y}-\mu_k^{(i,j)}\right)^{\mathrm{T}}\left(\Sigma_k^{(i,j)}\right)^{-1}\times\left(\mathbf{y}-\mu_k^{(i,j)}\right)\right] \tag{1}$$

where

$\mathbf{y}$ $\psi_{ij}(\mathbf{x})$;

$d=|\mathcal{F}_{ij}|$ dimensionality of the feature space $\mathcal{F}_{ij}$;

$\mu_k^{(i,j)}$ mean of class ω_k in the feature space $\mathcal{F}_{ij}$.

$$\mu_k^{(i,j)}=\frac{1}{N_k}\sum_{\mathbf{x}\in\mathcal{X}_k}\psi_{ij}(\mathbf{x}) \tag{2}$$

where $\mathcal{X}_k$ is the set of training examples in class ω_k. The corresponding covariance $\Sigma_k^{(i,j)}$ is given by

$$\Sigma_k^{(i,j)}=\frac{1}{N_k-1}\sum_{\mathbf{x}\in\mathcal{X}_k}\left(\psi_{ij}(\mathbf{x})-\mu_k^{(i,j)}\right)\times\left(\psi_{ij}(\mathbf{x})-\mu_k^{(i,j)}\right)^{\mathrm{T}}. \tag{3}$$

Using Bayes rule

$$\phi_{ij}(\mathbf{y})=\hat{P}_{ij}(\omega_i|\mathbf{y})=\frac{\hat{p}\left(\mathbf{y}|\omega_i\right)\hat{P}_{ij}(\omega_i)}{\hat{p}\left(\mathbf{y}|\omega_i\right)\hat{P}_{ij}(\omega_i)+\hat{p}\left(\mathbf{y}|\omega_j\right)\hat{P}_{ij}(\omega_j)},\quad k=i,j \tag{4}$$

where $\hat{P}_{ij}(\omega_k)$ are the estimated class priors based on the training data

$$\hat{P}_{ij}(\omega_k)=\frac{N_k}{N_i+N_j},\quad k=i,j. \tag{5}$$

These estimates are used in cases where the class distribution of the training data can be used to estimate the class priors. However, in the landcover classification problem, the class prior often cannot be estimated reliably from the class distributions of the training data and hence, equal priors were assumed.

The outputs of the $\binom{C}{2}$ classifiers can be combined (see Fig. 2) to obtain the final output either by simple voting [10] or by applying the MAP to estimates of the overall posterior probabilities obtained from the outputs of the pairwise classifiers [11]. In the voting combination scheme, a count $c(\omega_k|\mathbf{x})$ of the number of $\binom{C}{2}$ classifiers that labeled $\mathbf{x}$ as class ω_k

$$c\left(\omega_k|\mathbf{x}\right)=\sum_{i<k}I\left(\phi_{ik}(\mathbf{x})<0.5\right)+\sum_{i>k}I\left(\phi_{ki}(\mathbf{x})\geq 0.5\right) \tag{6}$$

is used. Here $I(bool)$ is the indicator function, which is 1 when the $bool$ argument is true and 0 otherwise. The input $\mathbf{x}$ is assigned the class label for which the count is maximum, i.e., $\omega(\mathbf{x})=\arg\max_{k=1,\ldots,C}c\left(\omega_k|\mathbf{x}\right)$.

In another recently proposed approach to combining pairwise classifiers [11], the overall posterior probabilities $p_i=P(\omega_i|\mathbf{x})\ \forall i=1,\ldots,C$ are estimated for some $\mathbf{x}$ from the $\binom{C}{2}$ probabilities as follows. Denote $m_{ij}=|\mathcal{X}_i|+|\mathcal{X}_j|$, $r_{ij}=\phi_{ij}\left(\psi_{ij}(\mathbf{x})\right)$, and $\nu_{ij}=\hat{p}_i/(\hat{p}_i+\hat{p}_j)$. The goal is to find an estimate $\hat{p}_i$ of true posteriors $P\left(\omega_i|\mathbf{x}\right)$ such that ν_{ij} is close to r_{ij}, $\forall i\neq j$. Since there are $C-1$ independent parameters but $\binom{C}{2}$ equations, it is not possible in general to estimate $\hat{p}_i$ so that $\nu_{ij}=r_{ij}\ \forall i\neq j$. Hence, only an approximate solution is sought. The objective for estimating $\mathbf{p}=(p_1,p_2,\ldots,p_C)$ is to minimize the weighted K–L distance between r_{ij} and ν_{ij}

$$J(\mathbf{p})=\sum_{i<j}m_{ij}\left[r_{ij}\log\frac{r_{ij}}{\nu_{ij}}+(1-r_{ij})\log\frac{1-r_{ij}}{1-\nu_{ij}}\right]. \tag{7}$$

This results in the following algorithm.

1) Initialize $\hat{p}_i=|\mathcal{X}_i|/|\mathcal{X}|$ and evaluate corresponding ν_{ij}.
2) Repeat the following updates for $i=1,2,\ldots,C,1,2,\ldots$ until convergence

$$\hat{p}_i\leftarrow\hat{p}_i\frac{\sum_{j\neq i}m_{ij}r_{ij}}{\sum_{j\neq i}m_{ij}\nu_{ij}}, \tag{8}$$

$$\hat{p}_i\leftarrow\frac{\hat{p}_i}{\sum_{j=1}^{C}\hat{p}_j} \tag{9}$$

$$\nu_{ij}\leftarrow\frac{\hat{p}_i}{\hat{p}_i+\hat{p}_j}. \tag{10}$$

[1] Alternatively, a mixture of Gaussian can be used as required.

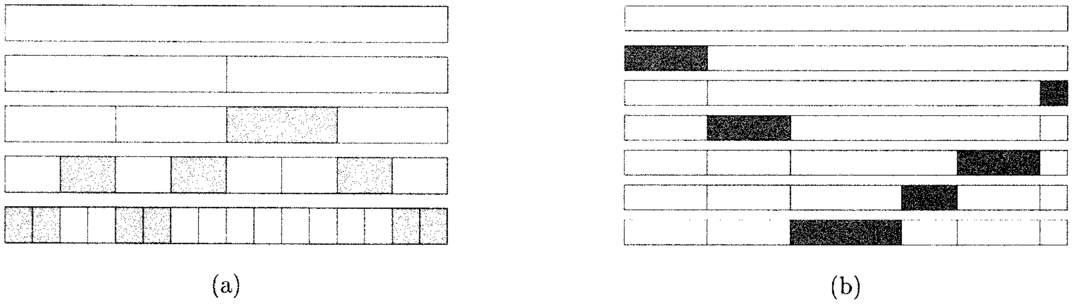

Fig. 3. (a) Example of a wavelet expansion of a 16-dimensional (16-D) signal into a full binary tree and a result of the LDB best-bases selection (dark blocks). The LDB best-bases algorithm always partitions the signal into two equal parts. (b) Example of a more general bases that cannot be obtained by such an LDB approach.

The input $\mathbf{x}$ for which $\hat{p}_i$, $i = 1, \ldots, C$ are estimated, is assigned class $\omega(\mathbf{x}) = \arg\max_{i=1,\ldots,C} \hat{p}_i$. While voting is simple and fast, the MAP approach produced slightly better results in the experiments reported in Section V.

D. Local Discriminant Bases (lDB)

Hyperspectral data are ordered in D-dimensional "signal" vectors. Coifman and Wickerhauser [12] developed a best-bases algorithm that first expands a signal into wavelet packet bases and then performs a bottom-up search for the bases that produce maximum compression. Saito and Coifman [13] extended this best-bases algorithm for classification and introduced the notion of local discriminant bases (LDB) for classification of signals and images.

The LDB algorithm first expands a given signal into a library of orthonormal bases, i.e., a redundant set of wavelet packet bases having a binary tree structure, where the nodes of the tree represent the subspaces with different time-frequency localization characteristics. Complete bases, called the best-bases, which minimize a certain information cost function such as entropy (for compression) or that maximize certain discrimination information function such as cross-entropy (for classification, as in LDB) [13], is sought in this binary tree using a divide-and-conquer algorithm. In this paper, Bhattacharya distance is used to evaluate each basis. The shaded regions of the tree in Fig. 3(a) illustrate an example of such a bases. The 16-length signal is partitioned into two eight-length segments, each of which is further partitioned into two four-length segments and so on. The LDB algorithm starts at the bottom and considers the relative goodness of the two-dimensional (2-D) space comprised of first two elements of the signal and the one-dimensional (1-D) space formed by merging these two elements. In this case, it finds that the 2-D finer resolution space has better discrimination and retains the (the first two shaded blocks at finest resolution). Next, the LDB considers the third and fourth elements and decides that merging them is better, and hence the first shaded block at the second resolution. The process continues, resulting in the kind of bases shown in Fig. 3(a). The signal is partitioned only into two equal parts at each resolution.

LDB cannot be applied directly for feature extraction for hyperspectral data, for the following reasons.

- **Fixed length subspaces:** Once a mother wavelet is fixed, the binary tree formed by the recursive expansion of the signal into a library of orthonormal bases is also fixed. As a result, bases other than the ones shown in Fig. 3(a) cannot be obtained. Fig. 3(b) shows an example of one such tree and a choice of bases that cannot be obtained from the LDB algorithm.
- **Requires full expansion:** The original best-bases algorithm of Coifman and Wickerhauser [12] requires that the signal be first decomposed completely into the finest bases possible before the second stage of finding the best-bases can be initiated. For a signal of length $n(n = 2^k)$, the complete expansion has the time complexity of $O(n \log_2 n)$. The complete expansion is necessary because the best-bases selection algorithm is bottom-up. As shown in the experiments, for classification problems, the actual number of features that is eventually selected is far less than n. That is, the tree is normally pruned near the root node and hence, a top-down approach would be more efficient.
- **No class dependence:** The LDB algorithm tries to find a set of best-bases for all the C classes by trying to maximize the overall discriminant function obtained by adding all the pairwise functions for all class pairs. As remarked in [13], for a large number of classes, it is not guaranteed that such a discrimination function yields the optimal set of bases for discriminating all classes simultaneously.

In the following sections, it will be shown that both the proposed algorithms, in conjunction with the pairwise architecture of Section II-C, are able to avoid all three problems.

III. Top-Down Generalized Local Discriminant Bases

Generalizing the LDB, the new top-down procedure builds the tree recursively partitioning the D bands into two possibly unequal length subsets or groups of bands. The fast, greedy algorithm, hereafter referred to as TD-GLDB, is different from the LDB algorithm in three ways. First, there is no predefined mother wavelet to fix the subspaces. A subspace of any number of adjacent bands is allowed so that partitions like the one shown in Fig. 3(b) are possible. Second, although the algorithm generates a binary tree, it does so in a top-down fashion, so the worst-case time complexity is only $O(DL)$, where L is the depth of the tree. Finally, a customized set of features is obtained for each pair of classes. The proposed TD-GLDB algorithm has the following three stages:

1) **Recursive partitioning of adjacent bands into nonoverlapping groups (subspaces):** All the D bands in the hyperspectral data set are recursively partitioned into a number of groups of adjacent bands as shown in Fig. 3(b). Each group is denoted by an interval $[\ell, u]$ comprised of all the bands between and including bands ℓ through u $(1 \leq \ell \leq u \leq D)$.
2) **Merging bands within each group:** Bands within each group are linearly combined to give single "group-bands". A user defined *merge function* denoted by $\mathcal{M}(\mathbf{x}|\ell, u)$ is used to merge bands in the group $[\ell, u]$. Herein, for simplicity, the mean of all the bands within a group is used as the merge function

$$y_{\ell,u}(\mathbf{x}) = \mathcal{M}(\mathbf{x}|\ell, u) = \frac{1}{u-\ell+1}\sum_{i=\ell}^{u} x_i. \tag{11}$$

In general, any linear combination such as a Fisher discriminant projection could be used as the merge function.
3) **Selection of group-bands:** A number of group-bands is obtained as a result of Steps 1 and 2. The subset that provides the best discrimination between classes α and β is selected using a forward feature selection algorithm [14] in which features are added from the available set until the increase in discrimination is not significant.

A. Class Discrimination Functions

Let $\mathcal{J}(\ell, u) = \mathcal{J}(\ell, u|\mathcal{M}, \mathcal{X}_\alpha, \mathcal{X}_\beta)$ denote a measure of discrimination between classes α and β along the group-band $y_{\ell,u}$ obtained by merging bands ℓ through u using a merge function $\mathcal{M}$. Choices for $\mathcal{J}$ are members of two broad categories.

1) **Classification performance on training/validation data:** Using a training or validation set, the classification accuracy of the two-class problem can be measured in the $y_{\ell,u}$ 1-D space assuming a maximum likelihood (ML) classifier in the 1-D space. This could also be used as a measure of discriminating capacity for group-band $y_{\ell,u}$.
2) **Differences in class probability density functions (pdfs):** If the pdfs $p_\alpha(y_{\ell,u})$ and $p_\beta(y_{\ell,u})$ are estimated from the data sets $\mathcal{X}_\alpha$ and $\mathcal{X}_\beta$, then the discrimination between classes α and β can be evaluated in terms of some difference measure between the two pdfs (e.g., the Kullback-Leibler divergence [15], Bhattacharya distance [16], etc.) In this paper, we use a discriminant measure based on the log-odds of (pairwise) class posterior probabilities [8], [9]

$$\begin{aligned}\mathcal{J}(\ell, u) =& \frac{1}{N_\alpha}\sum_{\mathbf{x}\in\mathcal{X}_\alpha} \log\frac{\hat{P}_{\alpha\beta}(\alpha|y_{\ell,u}(\mathbf{x}))}{\hat{P}_{\alpha\beta}(\beta|y_{\ell,u}(\mathbf{x}))} \\ &+ \frac{1}{N_\beta}\sum_{\mathbf{x}\in\mathcal{X}_\beta} \log\frac{\hat{P}_{\alpha\beta}(\beta|y_{\ell,u}(\mathbf{x}))}{\hat{P}_{\alpha\beta}(\alpha|y_{\ell,u}(\mathbf{x}))}\end{aligned} \tag{12}$$

where $\hat{P}_{\alpha\beta}(\alpha|y_{\ell,u}(\mathbf{x}))$ is the (estimated) pairwise class posterior probability of $\mathbf{x}$ projected on the basis $y_{\ell,u}$ for class α and $\hat{P}_{\alpha\beta}(\beta|y_{\ell,u}(\mathbf{x})) = 1 - \hat{P}_{\alpha\beta}(\alpha|y_{\ell,u}(\mathbf{x}))$. This log-odds ratio gave better performance than Bhattacharya distance and K–L divergence in our experiments. This is just an empirical observation and may not be true for all datasets.

B. Recursive Decomposition Algorithm

Once the merge function $\mathcal{M}$ and the discrimination function $\mathcal{J}$ are selected, the following top-down recursive algorithm partitions the original D-dimensional space into smaller subspaces.

```
Decompose $(\ell, u)$
  1) For each $\ell \leq k < u$ compute $\mathcal{J}(\ell, k)$ and $\mathcal{J}(k+1, u)$
     and find the best partition of the subspace $[\ell, u]$
       $\tilde{k} = \arg\max_{\ell \leq k < u} \max\{\mathcal{J}(\ell, k), \mathcal{J}(k+1, u)\}.$   (13)
  2) If $\mathcal{J}(\ell, \tilde{k}) > \mathcal{J}(\ell, u)$ and $\tilde{k} - \ell \geq 1$
       • Decompose $(\ell, \tilde{k})$.
  3 If $\mathcal{J}(\tilde{k}+1, u) > \mathcal{J}(\ell, u)$ and $u - \tilde{k} > 1$, then
       • Decompose $(\tilde{k}+1, u)$.
```

The condition in Steps 2 and 3 in this Decompose routine ensure that a subspace that does not show any improvement in its discrimination capacity from its parent node is not partitioned any further. This heuristic pruning mechanism is based on the following assumption.

Assumption I: If the discrimination $\mathcal{J}(a, b) < \mathcal{J}(A, B)$ for some $1 \leq A \leq a \leq b \leq B \leq D$, then for any subspace $[c, d]$ of $[a, b]$ i.e., $1 \leq A \leq a \leq c \leq d \leq b \leq B \leq D$, $\mathcal{J}(c, d) \leq \mathcal{J}(A, B)$.

Although we do not prove this theoretically, it was found to be true in all our experiments for both discrimination functions described in Section III-A. Using assumption I, steps 2 and 3 in the Decompose routine essentially imply that if a child node does not have a higher discrimination than the parent node, the child node need not be expanded further. This pruning mechanism leads to an efficient top-down search for a set of bands with high discrimination.

IV. Bottom-Up Generalized Local Discriminant Bases

A bottom-up algorithm (referred to as GLDB-BU [4]) that generalizes the LDB algorithm was also developed. Here, the search for the best-bases is conducted for each pair of classes separately, as opposed to the LDB where only one set of bases is extracted for distinguishing all classes simultaneously. It also allows the flexibility of merging any set of adjacent bands as opposed to merging bands obtained by recursively partitioning the set of bands into two equal groups in each level of the tree, as in LDB. Finally, the criteria used to measure the goodness of a group of bands uses both the correlation between those bands and the discrimination between the two classes when these bands are projected in the Fisher direction. The Fisher projections of a localized set of highly correlated bands yield the set of bases for a given pair of classes.

A. Criteria for Evaluating a Basis

In Section II-B, three properties of a hyperspectral feature extractor were identified. The pairwise classifier incorporates the first of those properties, i.e., class dependence, by extracting features independently for each class pair. The second and third properties are incorporated by adequately defining the criteria for evaluating a basis. One such criteria is proposed in this section.

The second property requires that the high correlation values between spectrally adjacent hyperspectral bands be utilized. Thus, the criteria should reward combining highly correlated bands rather than combining less correlated bands. There are a number of ways of defining the correlation among a group of bands in terms of the correlation between all pairs of bands given by a $D \times D$ correlation matrix $\mathbf{q}$ (e.g., Fig. 1)

$$q_{i,j} = \frac{|Q_{i,j}|}{\sqrt{Q_{i,i}Q_{j,j}}} \tag{14}$$

where $\mathbf{Q}$ is the covariance matrix

$$\mathbf{Q} = \frac{1}{|\mathcal{X}_\alpha| + |\mathcal{X}_\beta|} \sum_{\mathbf{x}\in\mathcal{X}_\alpha\cup\mathcal{X}_\beta} (\mathbf{x}-\mu)(\mathbf{x}-\mu)^{\mathrm{T}} \tag{15}$$

and μ is the mean over the two classes

$$\mu = \frac{1}{|\mathcal{X}_\alpha| + |\mathcal{X}_\beta|} \sum_{\mathbf{x}\in\mathcal{X}_\alpha\cup\mathcal{X}_\beta} \mathbf{x}. \tag{16}$$

The criteria used to quantify the correlation among the bands for any group of bands $[\ell, u]$ $(1 \leq \ell \leq u \leq D)$ is defined by the correlation measure $\mathcal{C}(\ell, u)$ as the *minimum* of all the correlations between every pair of bands in the range $[\ell, u]$

$$\mathcal{C}(\ell, u) = \min_{\ell \leq i < j \leq u} q_{i,j}. \tag{17}$$

The group of bands $[\ell, u]$ is highly correlated if $\mathcal{C}(\ell, u)$ is large.

The third property in Section II-B requires that the basis be discriminating. Thus, the criteria should reward a highly discriminating basis more than a less discriminating basis. Discrimination between the two classes can be quantified as the Fisher discriminant in the 1-D Fisher projection obtained from the subspace $[\ell, u]$. Let $\mu^\alpha_{\ell,u}$ and $\mu^\beta_{\ell,u}$ be the subvectors containing dimensions ℓ through u of the mean vectors μ^α and μ^β corresponding to the classes α and β. Let $\Sigma^\alpha_{\ell,u}$ and $\Sigma^\beta_{\ell,u}$ be the submatrices containing rows and columns ℓ through u of the class covariance matrices Σ^α and Σ^β of these classes, then the within class covariance $\mathbf{W}_{\ell,u}$ for the two-class problem is given by

$$\mathbf{W}_{\ell,u} = \frac{1}{P(\alpha) + P(\beta)} \left[P(\alpha)\Sigma^\alpha_{\ell,u} + P(\beta)\Sigma^\beta_{\ell,u}\right] \tag{18}$$

where $P(\omega)$ is the prior probability of class ω. The between class covariance $\mathbf{B}_{\ell,u}$ is given by

$$\mathbf{B}_{\ell,u} = \left(\mu^\alpha_{\ell,u} - \mu^\beta_{\ell,u}\right)\left(\mu^\alpha_{\ell,u} - \mu^\beta_{\ell,u}\right)^{\mathrm{T}}. \tag{19}$$

We define the discrimination measure $\mathcal{D}(\ell, u)$ for grouping these bands as the Fisher discriminant of the subspace induced by bands ℓ through u. Since we are considering only a two-class problem, the Fisher discriminant projects this subspace into a 1-D space through a Fisher projection vector $\mathbf{w}_{\ell,u}$ that maximizes

$$\mathcal{D}(\ell, u) = \frac{\mathbf{w}^{\mathrm{T}}_{\ell,u}\mathbf{B}_{\ell,u}\mathbf{w}_{\ell,u}}{\mathbf{w}^{\mathrm{T}}_{\ell,u}\mathbf{W}_{\ell,u}\mathbf{w}_{\ell,u}} \tag{20}$$

yielding the projection vector

$$\mathbf{w}_{\ell,u} = \mathbf{W}^{-1}_{\ell,u}\left(\mu^\alpha_{\ell,u} - \mu^\beta_{\ell,u}\right). \tag{21}$$

The product $\mathcal{J}(\ell, u) = \mathcal{C}(\ell, u)\mathcal{D}(\ell, u)$ of the correlation measure and the discrimination measure is used as the measure of goodness of the "group-band" $[\ell, u]$. The corresponding basis is given by the Fisher projection vector $\mathbf{w}_{\ell,u}$. A bottom-up search algorithm, described next, is used for finding the best set of bases for each class pair using this criteria.

B. Bottom-Up Search for the Best-bases

In the bottom-up search algorithm, let $\mathcal{B}^m(k) = [\ell^m_k, u^m_k]$ denote the set of actual bands that belong to the kth "group-band" and N_m denote the number of such group-bands at level m.

1) Initialize $m = 0$ (finest level), $\mathcal{B}^0(b) = [b, b]$, $\forall b = 1, \ldots, D$. $N_m = D$.

2) Find the best pair of bands to merge
   - for $i = 1, \ldots, N_m - 1$.

   — Form a group-band by merging group-bands $\mathcal{B}^m(i)$ and $\mathcal{B}^m(i+1)$

   $$[\ell_i, u_i] \leftarrow \left[\ell^m_i, u^m_{i+1}\right]. \tag{22}$$

   — Evaluate the group-band $[\ell_i, u_i]$: $\mathcal{J}(\ell_i, u_i) = \mathcal{C}(\ell_i, u_i)\mathcal{D}(\ell_i, u_i)$.
   - Find the position of the best pair of bands at this level

   $$I = \arg\max_{i=1,\ldots,N_m-1} \mathcal{J}(\ell_i, u_i). \tag{23}$$

3) If $\mathcal{J}(\ell_I, u_I) \geq \max\{\mathcal{J}(\ell^m_I, u^m_I), \mathcal{J}(\ell^m_{I+1}, u^m_{I+1})\}$ then continue; otherwise stop.

4) Update bands at the next level
   - If $I > 1$ then $\mathcal{B}^{m+1}(i) \leftarrow \mathcal{B}^m(i)$, i.e., $\left[\ell^{m+1}_i, u^{m+1}_i\right] \leftarrow [\ell^m_i, u^m_i]$, $\forall i = 1, \ldots, I-1$.
   - $\mathcal{B}^{m+1}(I) \leftarrow \mathcal{B}^m(I) \cup \mathcal{B}^m(I+1)$, i.e., $\left[\ell^{m+1}_I, u^{m+1}_I\right] \leftarrow \left[\ell^m_I, u^m_{I+1}\right]$.
   - If $I < N_m - 1$ then $\mathcal{B}^{m+1}(i) \leftarrow \mathcal{B}^m(i+1)$, i.e., $\left[\ell^{m+1}_i, u^{m+1}_i\right] \leftarrow \left[\ell^m_{i+1}, u^m_{i+1}\right]$, $\forall i = I+1, \ldots, N_m - 1$.

5) Move to the next level: $m \leftarrow m+1$, $N_m \leftarrow N_{m-1} - 1$. Return to Step 2.

The process of merging adjacent bands continues until a level (i.e., M), at which merging any two adjacent group-bands yields a worse (in terms of $\mathcal{J}$) group-band than either of the two group-bands being merged (Step 3). Each of the N_M group-bands $\{\mathcal{B}^{(M)}(i)\}_{i=1}^{N_M}$ obtained as a result of this process is associated with the corresponding Fisher projection vector $\left\{\mathbf{w}_{\ell^M_i, u^M_i}\right\}_{i=1}^{N_M}$ that forms the bases of projection being sought. All the selected bases are mutually orthogonal as the group-bands do not overlap. Instead of using all the N_M bases, a forward feature selection algorithm is applied to the resulting set of bases to select a subset such that adding any more bases will not lead to a significant increase (i.e., user-specified 1%) in the classification accuracy of the training set. Let $W_{\alpha\beta}$ denote the final $D \times K$

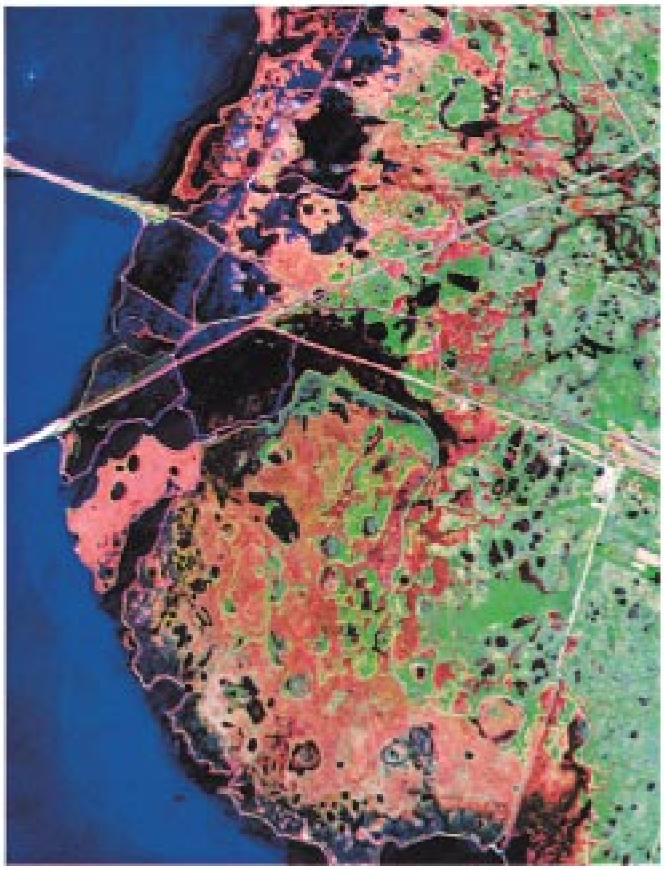

Fig. 4. Hyperspectral data: The AVIRIS Hyperspectral data obtained by NASA over Kennedy Space Center, FL. The bands corresponding to wavelengths of 1999 nm, 953 nm, and 527 nm were used for RGB channels for displaying purpose.

matrix containing the orthogonal bases after the feature selection process for distinguishing class pair (α, β).

C. A Bayesian Classifier for GLDB-BU

The classifier for class-pair (α, β) utilizes: 1) the bases vectors $W_{\alpha\beta}$ found by the algorithm presented in Section IV-B, 2) the K-dimensional means $W_{\alpha\beta}^{\mathrm{T}}\mu^{\alpha}$ and $W_{\alpha\beta}^{\mathrm{T}}\mu^{\beta}$ in the projected space, and 3) the $K \times K$ covariance matrices $W_{\alpha\beta}^{\mathrm{T}}\Sigma^{\alpha}W_{\alpha\beta}$ and $W_{\alpha\beta}^{\mathrm{T}}\Sigma^{\beta}W_{\alpha\beta}$. In terms of the pairwise classifier architecture, for any novel input $\mathbf{x}$, the feature extractor transforms it into $\mathbf{y}$ as

$$\mathbf{y} = \psi_{\alpha\beta}(\mathbf{x}) = W_{\alpha,\beta}^{\mathrm{T}}\mathbf{x}. \tag{24}$$

The pdf in the feature space for class $\gamma \in \{\alpha, \beta\}$ is given by

$$\begin{aligned} p\left(W_{\alpha\beta}^{\mathrm{T}}\mathbf{x}|\gamma\right) =& \frac{1}{\sqrt{(2\pi)^K\left|W_{\alpha\beta}^{\mathrm{T}}\Sigma^{\gamma}W_{\alpha\beta}\right|}} \\ &\times \exp\left[-\frac{1}{2}\left(\mathbf{x}-\mu^{\gamma}\right)^{\mathrm{T}}W_{\alpha\beta}\left(W_{\alpha\beta}^{\mathrm{T}}\Sigma^{\gamma}W_{\alpha\beta}\right)^{-1}\right. \\ &\qquad \left.\times W_{\alpha\beta}^{\mathrm{T}}\left(\mathbf{x}-\mu^{\gamma}\right)\right] \end{aligned} \tag{25}$$

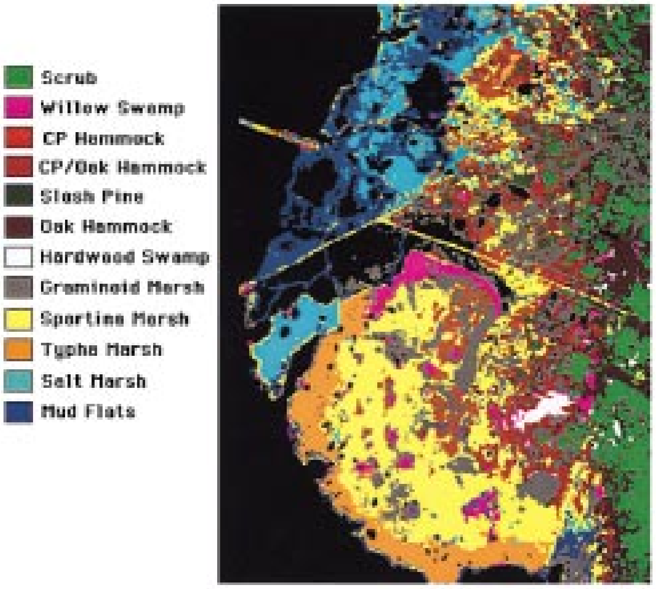

Fig. 5. Classified map of the KSC site using GLDB-BU algorithm.

The posterior probability $P_{\alpha\beta}(\alpha|\mathbf{x})$ is computed using (4). The results of all the pairwise classifiers are combined using either of the methods noted in Section II-C.

V. Experiments and Results

A. Site Description

The wetlands located on the west shore of the Kennedy Space Center (KSC), FL, and the Indian River contain critical habitat for several species of water fowl and aquatic life. Mapping the land cover and its response to wetland management practices using remotely sensed data from a variety of sensors is the focus of a multiyear project. In 1996, hyperspectral data were acquired using NASA's airborne visible infrared imaging spectrometer (AVIRIS) over the KSC complex. The test site for this study consists of a series of impounded marshes with vegetation communities ranging from low, halophyte marshes to high, graminoid savannah to forested wetlands. Discrimination between individual species of marsh vegetation and of woodland vegetation types is quite difficult because spectral signatures are similar. The capability for improved discrimination of the various vegetation types was investigated using the hyperspectral AVIRIS data. Fig. 4 shows three bands corresponding to wavelengths 1999 nm, 953 nm, and 527 nm of the AVIRIS data mapped to the RGB channels, respectively. Fig. 5 shows an example of the resulting classified map obtained by applying the GLDB-BU algorithm to the AVIRIS data.

The proposed GLDB-TD and GLDB-BU feature extraction algorithms were applied to a 183 band subset of the 224 bands (excluding water absorption bands) used in the classification. The seven upland and five wetland cover types identified for classification are listed in Table I: Classes 3–7, i.e., cabbage palm hammock (3), cabbage palm/oak hammock (4), slash pine (5), broad leaf/oak hammock (6), and Hardwood swamp (7), are all trees. Class 4 is a mixture of class 3 and oak hammock. Class 6 is a mixture of broad leaf trees (maples and laurels) and oak hammock. Class 7 also contains several species of broadleaf

TABLE I
TWELVE CLASSES IN THE AVIRIS HYPERSPECTRAL DATASET

Num	Class Name
	Upland Classes
1	Scrub
2	Willow Swamp
3	Cabbage palm hammock
4	Cabbage palm/oak hammock
5	Slash pine
6	Broad leaf/oak hammock
7	Hardwood swamp
	Wetland Classes
8	Graminoid marsh
9	Spartina marsh
10	Cattail marsh
11	Salt marsh
12	Mud flats

trees. These classes have similar spectral signatures and are very difficult to discriminate in multispectral and even hyperspectral data using traditional methods.

There were ≈ 350 examples for each class. These were randomly partitioned into 50% training and 50% test sets for each of the ten experiments. The proposed algorithms were compared to the SPCT and the LDB approach in terms of classification accuracy and reduction in the feature space. Since there is no systematic way of estimating the parameters of the training priors for each class, equal priors are assumed instead of using (5) (the number of training examples only reflects training samples that we were able to collect, but does not reflect the true prior of the classes).

B. Experimental Results

Table II contains the test set accuracies of the SPCT (upper triangular) and LDB (lower triangular) algorithms over all the 66 pairwise classifiers, while Table III contains the number of corresponding features used by each of these classifiers for SPCT and LDB. Although most pairs of classes were reasonably well discriminated by the features extracted by SPCT and LDB, the class pairs [cabbage palm/oak hammock (4), hardwood swamp (7)], [cabbage palm/oak hammock (4), broad leaf/oak hammock(6)], [slash pine(5), hardwood swamp(7)], and [cabbage palm/oak hammock(4), slash pine(5)] were not discriminated easily. Their classification accuracies over the test set were significantly less than the mean accuracy of 93% (see Table VII) over all the 66 pairwise classifiers. Further, the number of features selected for class pair (4,5) by SPCT (i.e., 15) and LDB (i.e., ten) is also largeer than the average number of features utilized (ten for SPCT and seven for LDB).

The classification accuracies over test sets for the GLDB-TD algorithm for all the 66 pairwise classifiers for both objective functions, classification accuracy on the training set (upper triangular), and log-odds ratio of posterior probabilities (lower triangular) are shown in Table IV. The corresponding number of features averaged over ten experiments are listed in Table V for the GLDB-TD algorithm. There is an average improvement of almost 5% in classification accuracies in all class pairs when using GLDB-TD feature extractor as compared to the SPCT or LDB feature extractors. Moreover, the classification accuracy over the difficult class pairs (4,5), (4,6), and (5,7) also improved by similar amounts. However, as a mixture class, class pair (4,7) still remains difficult to discriminate even by the GLDB-TD algorithm. Only 2–4 features were utilized by most of the 66 pairwise classifiers when using the GLDB-TD feature extractor compared to 7–12 features used by SPCT and 5–11 features used by LDB-based classifiers.

TABLE II
PAIRWISE CLASSIFIER ACCURACIES ON TEST SET FOR SPCT (UPPER TRIANGULAR) AND LDB (LOWER TRIANGULAR) FEATURE EXTRACTION ALGORITHMS

	1	2	3	4	5	6	7	8	9	10	11	12
1	-	93.1	94.2	91.4	92.4	86.9	94.6	93.1	92.6	91.5	92.4	96.8
2	94.2	-	95.7	91.1	90.3	93.8	89.8	94.7	95.2	92.2	94.1	95.2
3	95.1	95.9	-	92.5	92.8	95.2	97.4	95.6	96.2	95.9	97.3	95.3
4	93.0	92.6	93.6	-	86.3	88.1	69.7	94.0	95.2	97.8	95.3	96.6
5	94.2	91.8	91.3	87.1	-	94.5	80.9	97.1	94.5	97.4	93.9	95.8
6	87.1	94.5	96.9	87.9	95.1	-	87.1	93.1	91.6	93.7	96.7	97.1
7	96.4	90.3	96.1	72.4	78.3	88.2	-	93.6	97.9	98.3	95.2	97.3
8	94.7	92.1	95.9	95.1	96.7	94.6	94.3	-	92.8	89.1	96.4	91.2
9	94.3	95.9	95.1	96.7	95.2	90.9	96.3	91.2	-	94.6	95.1	96.7
10	93.6	90.1	96.4	96.1	96.1	95.2	97.3	90.4	94.6	-	91.3	87.2
11	93.4	92.3	95.1	94.8	96.5	94.8	96.7	95.7	95.1	93.1	-	92.9
12	95.7	96.4	97.2	94.1	95.2	96.2	96.9	93.1	96.7	89.6	93.8	-

TABLE III
AVERAGE NUMBER OF FEATURES USED FOR SPCT (UPPER TRIANGULAR) AND LDB (LOWER TRIANGULAR) FEATURE EXTRACTION ALGORITHMS

	1	2	3	4	5	6	7	8	9	10	11	12
1	-	9.4	12.1	8.8	7.3	12.7	7.2	10.8	8.1	11.4	12.4	12.3
2	6.1	-	12.0	14.9	16.0	11.1	10.0	9.8	8.9	9.5	6.3	10.5
3	7.4	5.0	-	9.7	12.0	10.0	12.1	9.5	11.4	10.7	7.9	11.2
4	5.9	9.9	5.7	-	15.5	10.6	9.7	11.8	14.3	11.1	11.9	8.7
5	6.2	10.1	6.9	10.2	-	7.5	9.5	11.3	6.1	12.3	9.9	8.8
6	7.5	6.2	6.1	7.3	4.3	-	10.1	13.3	14.1	12.1	7.8	10.4
7	6.0	7.6	8.2	6.1	6.1	6.2	-	7.6	8.2	11.4	10.5	6.8
8	6.4	5.2	4.9	7.5	8.8	9.5	4.7	-	17.8	10.6	9.7	9.1
9	5.6	7.0	7.3	10.2	5.4	11.3	6.3	14.3	-	15.2	8.2	7.9
10	7.7	5.1	6.5	7.6	9.1	10.2	8.7	7.3	10.3	-	7.9	12.5
11	8.2	4.2	5.3	7.2	5.2	5.7	7.4	5.8	5.2	5.2	-	11.7
12	7.5	7.6	7.1	5.9	7.1	7.2	5.7	6.2	6.1	9.3	8.3	-

Table VI contains the average classification accuracies over the test sets (upper triangular matrix) and the number of features in the reduced space (lower triangular matrix) for the GLDB-BU algorithm applied to the same data set. For most class pairs, the classification accuracy was almost 100% and the minimum of 92.5% was for class pair (4,5), one of the problematic class

TABLE IV
CLASSIFIER ACCURACIES FOR THE GLDB-TD FEATURE EXTRACTION ALGORITHM. THE UPPER TRIANGULAR MATRIX CONTAINS ACCURACIES FOR $\mathcal{J}_1$ = CLASSIFICATION ACCURACY ON TRAINING SET AND THE LOWER TRIANGULAR MATRIX CONTAINS ACCURACIES FOR $\mathcal{J}_2$ = LOG-ODDS RELEVANCE (12)

	1	2	3	4	5	6	7	8	9	10	11	12
1	-	98.3	100	97.1	98.7	91.3	100	97.2	99.5	97.9	98.8	99.5
2	97.7	-	99.4	96.0	95.3	95.6	93.1	99.1	99.5	98.6	99.9	99.6
3	100	97.9	-	96.4	96.9	98.9	100	99.4	99.3	99.1	99.6	99.6
4	95.4	96.3	95.5	-	90.8	91.4	73.3	99.0	99.4	99.1	99.7	99.2
5	98.4	94.8	95.7	89.9	-	97.3	86.2	99.2	99.6	99.5	99.8	99.7
6	89.9	95.5	97.6	90.3	95.6	-	90.9	98.9	98.9	99.6	99.1	99.7
7	99.9	96.8	97.2	82.2	83.8	91.1	-	99.9	100	100	99.9	100
8	97.2	97.9	98.7	98.7	99.1	99.3	99.9	-	96.8	93.8	99.9	95.5
9	99.3	99.0	98.7	98.9	99.5	98.8	99.9	95.9	-	99.4	99.4	99.9
10	98.2	99.0	99.0	99.1	99.5	99.7	100	92.4	95.6	-	98.7	90.9
11	99.6	99.5	99.3	99.6	100	98.6	100	99.9	98.8	98.9	-	98.5
12	98.9	99.5	99.3	99.3	99.7	99.9	100	94.4	98.6	91.8	98.4	-

TABLE V
AVERAGE NUMBER OF FEATURES USED BY THE GLDB-TD FEATURE EXTRACTION ALGORITHM. THE UPPER TRIANGULAR MATRIX IS FOR $\mathcal{J}_1$ = CLASSIFICATION ACCURACY ON TRAINING SET AND THE LOWER TRIANGULAR MATRIX IS FOR $\mathcal{J}_2$ = LOG-ODDS RELEVANCE

	1	2	3	4	5	6	7	8	9	10	11	12
1	-	2.0	2.2	1.5	1.2	3.6	1.9	2.2	1.2	2.7	2.9	3.1
2	2.0	-	3.0	3.9	5.0	2.1	3.0	2.8	2.9	2.5	2.3	2.2
3	1.4	3.0	-	2.7	3.0	2.0	4.1	2.5	2.4	2.7	1.9	2.2
4	2.2	5.5	2.1	-	4.5	2.6	2.7	2.8	3.3	3.1	2.9	2.3
5	1.3	4.2	3.2	5.3	-	1.5	2.5	3.3	3.1	3.3	2.9	2.2
6	2.3	3.6	2.0	1.3	1.4	-	2.1	3.4	3.1	3.3	2.2	2.4
7	1.9	4.5	5.5	4.1	5.2	2.0	-	1.8	2.6	2.5	2.3	1.3
8	4.3	5.3	2.7	4.0	5.1	4.7	2.6	-	4.1	2.5	2.2	2.9
9	1.2	2.3	2.6	4.9	3.6	3.6	2.0	5.2	-	3.4	2.6	2.8
10	3.1	5.3	2.5	3.1	3.3	3.7	2.0	5.7	2.3	-	2.6	3.4
11	2.0	2.0	3.0	2.0	2.0	2.7	2.2	3.8	3.1	2.8	-	2.3
12	3.3	3.9	2.0	2.6	3.1	2.4	2.0	6.1	2.7	5.0	2.0	-

pairs. Further, note that class pair (4,7) had a classification accuracy of 100% as compared to 82% by GLDB-TD and $< 73\%$ by LDB and SPCT. Class pair (4,6) was also classified easily by the GLDB-BU to yield almost 100% classification accuracy compared to $< 88\%$ by SPCT and LDB and 91% by GLDB-TD. Further, only one feature was required by GLDB-BU to discriminate class pair (4,7) compared to more than four by GLDB-TD, 15 by SPCT, and six by LDB. The number of features required by most of the classifiers based on GLDB-BU feature extraction was 1–2, with class pair (4,5) requiring more and still not being as well-discriminated as other pairs. Fig. 7(a) shows the bases and distribution of points in the three dimensional (3-D) feature space for class pair (4,5). Similarly, Fig. 7(b) shows the bases selected by GLDB-BU for class pair (4,7) and the corresponding distribution of points in the projected 1-D space.

Improvement in classification accuracy with a simultaneous reduction in the number of features in the feature space can be attributed mainly to the generalization of the LDB algorithm. While only highly correlated bands were allowed to merge, their usefulness was measured in terms of how well they can discriminate the two classes. Since SPCT uses the K–L transform to reduce the dimensionality, it does not necessarily transform the input into a highly discriminatory feature space. LDB is limited by the types of subspaces it can select, and hence its subspaces are inferior to those of the proposed algorithms. Compared to GLDB-TD, the proposed GLDB-BU performs better both in terms of classification accuracy and the number of required features. This can be attributed to the following differences between the two algorithms (i) GLDB-BU is bottom-up and hence searches through the space of possible bases more exhaustively than the top-down GLDB-TD; (ii) The merge function for GLDB-TD is just the mean while in GLDB-BU, the merge function is the Fisher direction of projection, which is expected to perform better in terms of discrimination between the two classes.

Table VII summarizes the results of Tables I–VI. The original problem was a 12-class problem that was decomposed into 66 two-class problems, then the 66 pairwise classifiers were combined using the method proposed by Tibshirani [11] (described in Section II-C). Overall test accuracies for all the four algorithms are contained in Table VII. The high values of classification accuracies from GLDB-TD and GLDB-BU for each pair of classes also resulted in higher accuracies in the overall C-class problem.

C. Computational Complexity

There are three components of the total time complexity of the feature extraction algorithms that sought a set of linear transformations or bases. The four algorithms: SPCT, LDB, TD-GLDB, and BU-GLDB are evaluated in terms of these three components.

1) **Time to evaluate a basis:** The goodness of a single basis is measured by some criteria such as the Bhattacharya distance in LDB, the amount of variance preserved or eigen values in SPCT, log-odds ratio of posterior probabilities in TD-GLDB, and Fisher discriminant in BU-GLDB algorithms. For an n-dimensional subset of bands from ℓ through u $(n = u - \ell + 1)$, computation of Bhattacharya distance (LDB), PCA projection (SPCT) and Fisher projection (GLDB-BU) take $\mathcal{O}(n^3)$ time. In GLDB-TD, only the mean of the n bands is computed, which requires $\mathcal{O}(n)$ time.
2) **Time to search for a bases set:** The feature extraction algorithms seek multiple localized set of bases. The LDB algorithm uses the wavelet packet best-bases search approach that takes $\mathcal{O}(D \log D)$ for a signal of length D. The SPCT algorithm utilizes edge detection algorithms on the correlation matrices to first isolate groups of localized highly correlated bands. This process takes $\mathcal{O}(D)$ time. The TD-GLDB algorithm does a top-down search and prunes the search based on Assumption 1. This takes $\mathcal{O}(DL)$ time, where L is the depth of the tree. The GLDB-BU algorithm requires $\mathcal{O}(D \log D)$ time as it is also bottom-up like the LDB algorithm.

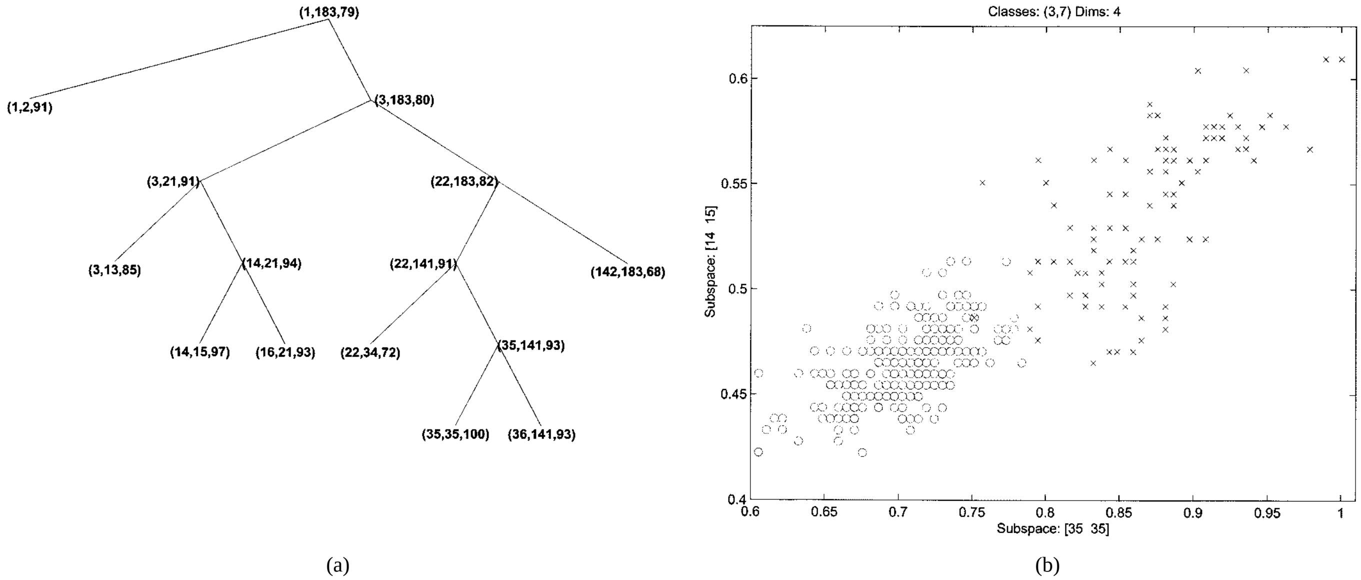

Fig. 6. (a) Tree obtained by the GLDB-TD algorithm for class pair cabbage palm hammock (3) and hardwood swamp (7). The three numbers at each node denote $(\ell, u, \mathcal{J})$, the range of the bands $[\ell, u]$, and goodness measure of each 1-D feature (the classification accuracy on training set). (b) Distribution of all data points in class 3 and 7 in the 2-D space formed by the group-bands [35,35] and [14,15].

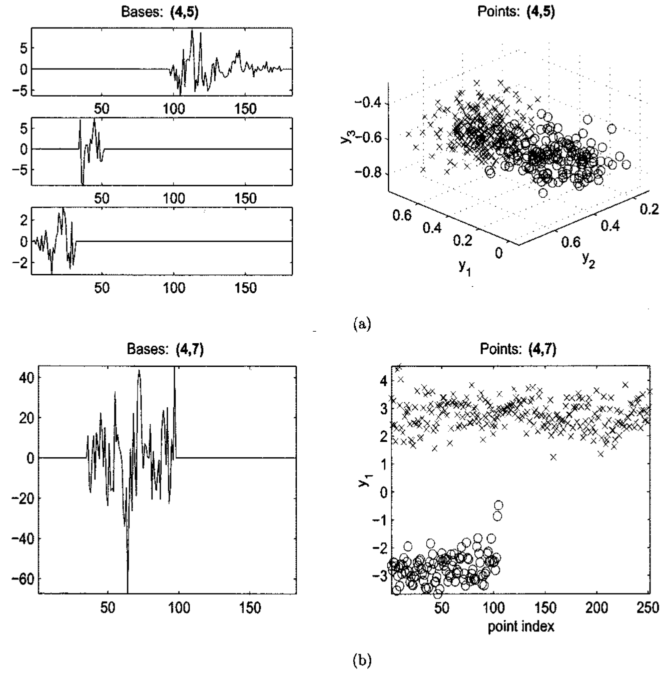

Fig. 7. (a) Bases weights selected by GLDB-BU algorithm for discriminating classes (4,5) and the corresponding distribution of points in this three-dimensional (3-D) feature space. (b) Bases weights selected by GLDB-BU algorithm for discriminating classes (4,7) and the corresponding distribution of points in this two-dimensional (2-D) space.

TABLE VI
CLASSIFIER ACCURACIES ON TEST SET (UPPER TRIANGULAR) AND THE NUMBER OF FEATURES USED (LOWER TRIANGULAR) FOR THE GLDB-BU ALGORITHM

	1	2	3	4	5	6	7	8	9	10	11	12
1	-	100.0	100.0	99.3	100.0	93.1	100.0	99.3	100.0	100.0	99.9	100.0
2	1.1	-	100.0	99.8	100.0	100.0	98.9	100.0	100.0	100.0	100.0	100.0
3	1.3	1.1	-	96.1	95.4	99.8	100.0	98.8	99.5	100.0	100.0	100.0
4	1.0	1.0	2.8	-	92.5	99.8	100.0	98.8	100.0	99.7	99.7	99.7
5	1.1	1.2	1.2	3.1	-	100.0	100.0	99.0	100.0	100.0	100.0	100.0
6	1.9	1.1	1.1	2.0	1.0	-	99.1	99.4	99.6	100.0	100.0	100.0
7	1.1	1.0	1.0	1.1	1.2	1.4	-	100.0	100.0	100.0	99.5	100.0
8	1.3	1.0	1.0	1.2	1.1	1.1	1.0	-	98.9	99.2	100.0	98.7
9	1.1	1.2	1.2	1.1	1.4	1.2	1.0	1.6	-	100.0	99.8	100.0
10	1.0	1.1	1.1	1.1	1.2	1.0	1.0	1.2	1.0	-	100.0	99.2
11	1.0	1.3	1.1	1.0	1.0	1.0	1.1	1.0	1.2	1.1	-	99.6
12	1.0	1.1	1.0	1.0	1.1	1.0	1.2	1.0	1.0	1.1	1.2	-

TABLE VII
SUMMARY OF TABLES II THROUGH VI: (I) TEST ACCURACIES, (III) NUMBER OF FEATURES AVERAGED OVER ALL THE 66 PAIRWISE CLASSIFIERS ([] = STD. DEVIATION OVER 66 PAIRWISE CLASSIFIERS), AND (III) OVERALL ACCURACY AFTER COMBINING THE 66 CLASSIFIERS ({} = STD. DEVIATION ACROSS TEN EXPERIMENTS)

	GLDB-BU	**GLDB-TD**	**SPCT**	**LDB**
Test Accuracy	99.4 [1.31]	97.52 [4.20]	93.26 [4.40]	93.53 [4.14]
Num. Features	1.2 [0.42]	2.67 [0.73]	10.57 [2.42]	7.11 [1.88]
Overall accuracy	95.3 (2.42)	86.3 (3.56)	78.7 (3.47)	79.3 (4.12)

3) **Number of Bases sets:** In conventional classifiers, a single feature space is sought for dimensionality reduction. The pairwise classifier framework, however, seeks one feature space for each pair of classes. This increases the time complexity of using a pairwise classifier by $\mathcal{O}(C^2)$ times for a C-class problems. The SPCT and LDB algorithms have been used with conventional classifiers that extract only one feature space for all C-classes. This approach does lead to faster feature extraction but as the number of classes grows, the interpretability and discrimination capability of such a feature space decreases.

Thus, in terms of evaluating a basis and searching for a set of bases, the GLDB algorithms are comparable with SPCT and LDB algorithms. However, when used in conjunction with the pairwise classifier framework, any feature extraction algorithm will have to be used $\binom{C}{2}$ times, increasing the overall time of learning.

VI. Domain Knowledge

The feature extraction algorithms used in conjunction with the pairwise classifier architecture not only increased classification accuracies and reduced input dimensionality, but also provided useful domain knowledge. In this section, some examples of the kinds of domain knowledge obtained from the best-bases algorithms are highlighted.

Focused feature extraction based on the original bands of data for each individual pair of classes results in feature spaces that are suitable for distinguishing specific pairs of classes. This knowledge can be very important in determining what sensors or bands should be used and may be related to the associated physiological characteristics those bands represent in the two land cover classes. Fig. 6(a) shows an example of the top-down tree formed by the GLDB-TD algorithm for discriminating classes cabbage palm hammock (3) and hardwood swamp (7), and both upland tree classes. The leaf nodes of the tree represent the group of bands that were merged to their means for greatest discrimination between class pair (3,7). Each node in the tree represents a subspace of the original 183-dimensional input space. The first two numbers at each node in Fig. 6(a) represent the $[\ell, u]$ values of that node. The third number shows the $\mathcal{J}(\ell, u)$ value of that node measured as classification accuracy over the training data (rounded to nearest integer) (the root node always represents the 1-D space obtained by merging all the bands [1, 183]). For the class pair (3,7), the root node has a discrimination value of only 79%, while the subspace [14,15] has a discrimination of 97% and the subspace [35,35] has a discrimination value of 100%. The pruning heuristic does not allow further decomposition of the leaf node, for example [142, 183], because the value of the discriminant function at this node, i.e., 68% is less than its parent node [22, 183] i.e., 82%. The class pair (3,7) was easily distinguished by a pair of very narrow sets of bands, i.e., band 35 and the mean of bands [14,15]. Similar trees were obtained for all $\binom{C}{2}$ class pairs.

Fig. 6(b) shows the distribution of all the labeled sample points in classes 3 (red circles) and 7 (blue crosses) projected into the space obtained by the bands [35,35] and [14,15] in the top-down tree shown in Fig. 6(a). These are the first two of the four group-bands selected by the feature selection algorithm that follows the tree building phase in the GLDB-TD algorithm. For any pair of classes, the eventual bands selected by the GLDB-TD algorithms were more or less the same for different randomly chosen training examples.

Fig. 7 shows the bases selected by the GLDB-BU algorithm for two pairs of problematic classes: (4,5) and (4,7). The left plot shows the Fisher discriminant bases weights $\mathbf{w}_{\ell,u}$ obtained by (21). Each basis is localized within the highly correlated regions shown in Fig. 1. For each pair of classes, different numbers and types of bases are obtained in general. The class pair (4,5) [Fig. 7(a))] required three bases. The distribution of all the data points in the corresponding 3-D feature space is also shown. This class pair was hard to separate even by the GLDB-BU algorithm. Class pair (4,7) that was very easily discriminated by GLDB-BU required only one basis shown in Fig. 7(b). The distribution of all the data points for these two classes in the 1-D space shows why GLDB-BU obtains 100% classification accuracy for this class pair.

The bands selected by GLDB-TD and GLDB-BU overlapped for most class pairs. However, since these algorithms merge the bands differently, in general, different bases are obtained.

VII. Conclusions

Two new best-bases algorithms adapted to the properties of hyperspectral data were presented and evaluated using an AVIRIS data set. These algorithms extend the local discriminant bases algorithm developed for signal and image classification. The GLDB-TD algorithm uses a top-down greedy search and is fast but less exhaustive. The GLDB-BU algorithm performs a more exhaustive, yet efficient bottom-up search for the best bases to discriminate any two classes and utilizes the correlation between bands as well as discrimination between classes to measure the goodness of a basis. Empirical results for a 12-class problem show improvements of 5 to 10% in classification accuracies using a smaller number of features relative to LDB and the recently proposed SPCT algorithms for each of the 66 class pairs. As expected, the GLDB-BU had slightly better performance than GLDB-TD in terms of the classification accuracy and the number of features selected but required more computation. The overall accuracy after merging the 66 pairwise classifiers to solve the original 12-class problem was also significantly higher for GLDB-BU than the other approaches. The two extensions of the LDB algorithm suited for hyperspectral data were able to significantly reduce the input space from 183 dimensions to less than four dimensions in most cases. Domain knowledge pertaining to the importance of different bands for distinguishing specific class pairs and overall importance of each band was also provided. The best-bases algorithms thus provide a means to "create" lower spectral resolution sensors, each with only a few bands, specific to a particular classification problem.

References

[1] C. Lee and D. A. Landgrebe, "Decision boundary feature extraction for neural networks," *IEEE Trans. Neural Networks*, vol. 8, pp. 75–83, Jan. 1997.

[2] X. Jia and J. A. Richards, "Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification," *IEEE Trans. Geosci. Remote Sensing*, vol. 37, pp. 538–542, Jan. 1999.

[3] S. Kumar, J. Ghosh, and M. M. Crawford, "Multiresolution feature extraction for pairwise classification of hyperspectral data," *Proc. SPIE*, vol. 3962, pp. 60–71, Jan. 2000.

[4] ——, "Classification of hyperspectral data using best-bases feature extraction algorithms," *Proc. SPIE*, vol. 4055, pp. 362–373, Apr. 2000.

[5] X. Jia, "Classification techniques for hyperspectral remote sensing image data," Ph.D. dissertation, Univ. College, ADFA, Univ. New South Wales, Australia, 1996.

[6] L. O. Jimenez and D. A. Landgrebe, "Hyperspectral data analysis and supervised feature reduction via projection pursuit," *IEEE Trans. Geosci. Remote Sensing*, vol. 37, pp. 2653–2664, Nov. 1999.

[7] P. Huber, "Projection pursuit," in *The Annals of Statistics*, 1985, vol. 13, pp. 435–475.

[8] S. Kumar, J. Ghosh, and M. M. Crawford, "A versatile framework for labeling imagery with large number of classes," in *Proc. Int. Joint Conf. Neural Networks*, Washington, DC, 1999.

[9] M. M. Crawford, S. Kumar, M. R. Ricard, J. C. Gibeaut, and A. Neuenschwander, "Fusion of airborne polarimetric and interferometric SAR for classification of coastal environments," *IEEE Trans. Geosci. Remote Sensing*, vol. 37, pp. 1306–1315, May 1999.

[10] J. H. Friedman, "On bias, variance, loss, and the curse of dimensionality," Dept. Statistics, Stanford Univ., Stanford, CA, 1996.

[11] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," in *Advances in Neural Information Processing Systems*, M. J. Karens, M. I. Jordan, and S. A. Solla, Eds. Cambridge, MA: MIT Press, 1998, vol. 10, pp. 507–513.

[12] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inform. Theory*, vol. 38, pp. 713–719, Mar. 1992.

[13] N. Saito and R. R. Coifman, "Local discriminant bases, in Mathematical Imaging: Wavelet Applications in Signal and Image Processing II," in *Proc. SPIE*, vol. 2303, 1994, pp. 2–14.

[14] A. W. Whitney, "A direct method of nonparametric measurement selection," *IEEE Trans. Comput.*, vol. 20, pp. 1100–1103, 1971.

[15] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[16] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bull. Calcutta Math. Soc.*, vol. 35, pp. 99–109, 1943.

Shailesh Kumar was born on January 27, 1974, in Meerut, India. He received the B.Tech. degree (with honors) in computer science and engineering from the Institute of Technology, Banaras Hindu University, Varanasi, India, in 1995, and the M.S. and Ph.D. degrees, in computer science and computer engineering, respectively, from the University of Texas, Austin, in 1998 and 2000, respectively.

While pursuing the Ph.D. degree, he was with the Laboratory of Artificial Neural Systems and Center for Space Research, University of Texas. He is currently a Staff Scientist in the Advanced Technology Solutions Division of HNC Software, San Diego, CA. His main research interests are in pattern recognition, machine learning, data mining, remote sensing, wavelet technology, and image processing. He has published over a dozen conference papers and several journal papers in these areas.

Joydeep Ghosh was received the B.Tech degree from the Indian Institute of Technology (IIT), Kanpur, India, in 1983, and the M.S. and Ph.D. degrees from the University of Southern California, Los Angeles, in 1988.

In 1988, he joined the the Department of Electrical and Computer Engineering, University of Texas, Austin, where he has been a Full Professor since 1998, and holder of the Archie Straiton Endowed Fellowship. He directs the Laboratory for Artificial Neural Systems (LANS), where his research group is studying the theory and applications of adaptive pattern recognition, data mining including web mining, and multilearner systems. He has published more than 200 refereed papers and edited eight books.

Dr. Ghosh has received six best paper awards, including the 1992 Darlington Prize for the Best Journal Paper from the IEEE Circuits and Systems Society, and the Best Applications Paper at ANNIE'97. He has served as the General Chairman for the SPIE/SPSE Conference on Image Processing Architectures, Santa Clara, CA, February 1990, as Conference Co-Chair of Artificial Neural Networks in Engineering (ANNIE)'93 -ANNIE'96, ANNIE '98-2001, and has been on the Program or Organizing Committees of several conferences on neural networks and parallel processing. More recently, he co-organized workshops on Web Mining (with the SIAM International Conference on Data Mining, 2001) and on Parallel and Distributed Data Mining (with KDD, 2000). He was a Plenary Speaker for ANNIE'97 and Letters Editor for IEEE Transactions on Neural Networks from 1998 to 2000. He is currently an Associate Editor of *Pattern Recognition*, *Neural Computing Surveys*, and the *International Journal of Smart Engineering Design*.

Melba M. Crawford (M'90) received the B.S. degree in civil engineering and the M.S. degree in civil and environmental engineering from the University of Illinois, Urbana, in 1970 and 1973, respectively, and the Ph.D. degree in industrial and systems engineering from The Ohio State University, Columbus, in 1981.

She has been a Member of faculty with the University of Texas (UT), Austin, since 1980. and affiliated with the UT Center for Space Research since 1988. Her primary research interests are in statistical methods in image processing and development of algorithms for analysis of remotely sensed data. Her current research projects include mapping of coastal vegetation, multisensor topographic mapping, and multiresolution methods in data integration.