A Fast Two-Stage Classification Method for High-Dimensional Remote Sensing Data

Te-Ming Tu, Chin-Hsing Chen, Jiunn-Lin Wu, and Chein-I Chang, *Senior Member, IEEE*

***Abstract*— Classification for high-dimensional remotely sensed data generally requires a large set of data samples and enormous processing time, particularly for hyperspectral image data. In this paper, we present a fast two-stage classification method composed of a band selection (BS) algorithm with feature extraction/selection (FSE) followed by a recursive maximum likelihood classifier (MLC). The first stage is to develop a BS algorithm coupled with FSE for data dimensionality reduction. The second stage is to design a fast recursive MLC (RMLC) so as to achieve computational efficiency. The experimental results show that the proposed recursive MLC, in conjunction with BS and FSE, reduces computing time significantly by a factor ranging from 30 to 145, as compared to the conventional MLC.**

***Index Terms*— Band selection (BS), canonical analysis (CA), principal components analysis (PCA), recursive ML classifier (MLC), Winograd's identity.**

I. Introduction

ONE OF THE challenging problems in processing high-dimensional data is the computational complexity resulting from processing the vast amount of data volume. This is particularly true for remotely sensed image data, such as hyperspectral images collected by as many as 224 spectral bands from the airborne visible/infrared imaging spectrometer (AVIRIS) [1]. In order to mitigate this problem, data dimensionality reduction is generally required. One of widely used methods is the Karhunen–Loeve transform or principal components analysis (PCA) [2], [3], which reorganizes data in such a manner that the principal axis is one, and the data has the maximum variance. An alternative approach is to judiciously select a subset of dominant bands at the expense of some information loss. In this case, the selection of appropriate bands to preserve desired information is crucial. This is one of the main issues in the remote sensing community [4], [5]. The need for band selection (BS) arises from the fact that adjacent bands are highly correlated and not all of them are equally important and useful. Consequently, finding such a set of bands to reduce the data volume, while preserving all wanted information is highly desirable. As a matter of fact, in real applications, strongly correlated bands can be eliminated, and only those bands bearing significantly different information are required for data processing.

Classification is one of the most often used quantitative data analysis techniques to describe ground cover types or material classes. A variety of classification methods is available in the literature [6]. Of most interest is the maximum likelihood (ML) classification. One disadvantage of ML classification is excessive computing time, required for processing data with high dimensionality. So, developing an efficient algorithm to perform the ML classification seems beneficial in the remote sensing community. Recently, Lee *et al.* [7] proposed a multistage classification method, which decomposed the classification task into several stages. In each stage, the likelihood values of classes were calculated on the basis of a partial set of features. Then, those classes with likelihood values less than a predetermined threshold would be truncated and not be processed in the subsequent stages. As reported, the multistage likelihood classification reduced the processing time by a factor of three to seven. By taking another approach, Jia *et al.* [8] developed a block-form ML classification to reduce the processing time. The idea is to break up a correlation matrix into a group of subblock matrices so that only diagonal subblock matrices can be used for ML classification. As a result, high-dimensional data can be reduced to a group of smaller data sets with low dimensionality, each of which is represented by a subblock matrix. In [9], Settle and Briggs took advantage of an eight-pipeline processor and a Model 75 image processor, to speed computing time up to 100 times that which is required for the conventional ML classification.

In this paper, we present a fast two-stage classification method, depicted in Fig. 1. It is carried out by a BS algorithm with feature extraction and selection (FSE), then followed by a recursive MLC (RMLC). The BS algorithm proposed here is based on so-called canonical analysis (CA) [10]–[12] and is different from those in [13]. Despite the fact that PCA has been in favor in many remote sensing applications, PCA is only optimal in the sense of minimum mean-squared error and not necessarily optimal for class separability [3], [6]. As a result, PCA-based approaches will not generally yield the best performance in terms of classification. The advantage of CA over PCA is that CA is developed using the principle of within-class and between-class scatter matrices: a key concept adopted in Fisher's linear discriminant analysis [14]. CA is simultaneous to maximize the between-class scatter matrix and minimize the within-class scatter matrix so as to achieve the maximum possible class discrimination. Using the eigenvalues and eigenvectors generated by CA, a loading factor matrix can

Manuscript received August 8, 1995; revised July 2, 1996. This work was supported by the National Science Council under Grants NSC 85-2213-E-006-066 and NSC 84-2213-E-006-086.

T.-M. Tu is with the Department of Electrical Engineering, Chung Cheng Institute of Technology, Tahsi, Taoyuan, Taiwan 33509, R.O.C. (e-mail: tutm@cc04.ccit.edu.tw).

C.-H. Chen and J.-L. Wu are with the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan 70101, R.O.C.

C.-I. Chang is with the Department of Computer Science and Electrical Engineering, Remote Sensing Signal and Image Processing Laboratory, University of Maryland-Baltimore County, Baltimore, MD 21228-5398 USA.

Publisher Item Identifier S 0196-2892(98)00034-5.

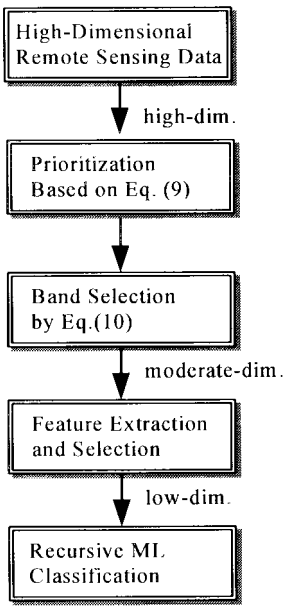

Fig. 1. Block diagram of the proposed fast two-stage MLC.

be defined [15], through which a discriminant power (DP) is calculated for each spectral band and used as the classification capability of that band. A higher DP indicates a better discrimination. By virtue of DP all bands can be prioritized in accordance with their DP's. Since the BS does not do any spectral compression, CA is applied again to the selected band data to extract features and select desired features for next stage classification. In order to further reduce computational complexity of the conventional ML classification, a RMLC, based on a Winograd's algorithm [16]–[18], is devised in the second stage to achieve computational efficiency.

Comparing the MLC's developed in [7] and [8], the proposed two-stage MLC has two distinct features that make it appealing and attractive in applications. These are BS and recursive algorithmic structure. While the former allows us to reduce high-dimensional data to low-dimensional data, the latter enables us to implement the MLC recursively on computers in an efficient way. In addition, the proposed MLC is compatible with the multistage ML [7] and the block-form ML [8], which can be easily included in our algorithm to further improve computing saving. Moreover, unlike [9], which utilized a pipeline (parallel) computer architecture, the designed RMLC does not rely on any parallel structure. It can be implemented on traditional von Neumann computers. By coupling the BS algorithm with the FSE, the RMLC can cut down the computing time by a factor ranging from 30 to 145, as compared to the conventional MLC. Although we do not have a Model 75 image processor for comparison, the advantage of our two-stage ML classification method over the work in [9] is that our MLC can be implemented on conventional sequential machines rather than parallel processors, while still achieving substantial saving in computation. Experiments demonstrate the merit of this proposed fast two-stage ML classification method.

This paper is organized as follows. Section II describes a CA-based approach to BS by using the concept of loading factors followed by FSE. Section III presents a fast recursive MLC based on a Winograd's algorithm. Section IV conducts various experiments to evaluate the performance of the proposed method in terms of classification accuracy, processing time, and computational efficiency. Section V draws a brief conclusion.

II. Band Selection Algorithm and Feature Extraction/Selection

The objective of BS is to properly select m bands from a large set of l bands in such a fashion that the data provided by the selected bands can sufficiently represent all the bands in some optimal sense. Several approaches have been proposed in the past for this purpose. For example, distance measure, such as Bhattacharyya distance, Mahalanobis distance, and Jeffreys–Matusita distance [13], [19], information measure, such as divergence and mutual information [13], [20], and eigenanalysis, such as PCA and CA [2], [3], [12], [14]. The first two approaches generally require computing measure criteria between all possible classes. Consequently, it is computationally prohibitive, since the number of subsets considered for possible classes grows exponentially with the number of bands. On the other hand, the eigenapproach makes use of the eigenvalues and eigenvectors generated by the data correlation matrix to rotate the original data coordinates along the direction of maximum variance. Since CA is optimal in terms of class separability, we adopt it for the development of our algorithm.

A. CA

Suppose that $\{\omega_1, \omega_2, \cdots, \omega_c\}$ are the classes of interest and x_{ij} is the jth sample in class i. Let N_i be the number of samples in the ith class and $N = N_1 + N_2 + \cdots + N_c$. The mean of total samples $\Xi = \{\boldsymbol{x}_{ij}\}_{i=1,\,j=1}^{c,\,N_i}$ is given by $\boldsymbol{m} = (1/N)\sum_{i=1}^{c}\sum_{j=1}^{N_i}\boldsymbol{x}_{ij}$ and $\boldsymbol{m}_i = (1/N_i)\sum_{x_{ij}\in\omega_i}\boldsymbol{x}_{ij}$ is the mean of samples in the ith class. One of important criteria used in classification is to define an appropriate rule for measuring the distance between two samples within a class as well as in separate classes. A good classification criterion ought to be able to minimize the distance between samples in one class and, in the mean time, maximize the distance between samples in separate classes. CA is basically developed to reflect this philosophy. Let the *total scatter matrix* $\boldsymbol{S}_T$, *within-class matrix* $\boldsymbol{S}_W$ and *between-class matrix* $\boldsymbol{S}_B$ be defined as follows [14]:

$$\boldsymbol{S}_T = \frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N_i}(\boldsymbol{x}_{ij}-\boldsymbol{m})(\boldsymbol{x}_{ij}-\boldsymbol{m})^T \tag{1}$$

$$\boldsymbol{S}_W = \sum_{i=1}^{c}\sum_{x_{ij}\in\omega_i}\frac{1}{N}(\boldsymbol{x}_{ij}-\boldsymbol{m}_i)(\boldsymbol{x}_{ij}-\boldsymbol{m}_i)^T \tag{2}$$

$$\boldsymbol{S}_B = \sum_{i=1}^{c}\frac{N_i}{N}(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{m}_i-\boldsymbol{m})^T \tag{3}$$

$$\boldsymbol{S}_T = \boldsymbol{S}_W + \boldsymbol{S}_B \tag{4}$$

where $\boldsymbol{S}_W$ in (2) represents the mean of within-class sample distance and $\boldsymbol{S}_B$ in (3) is the mean of between-class sample distance. Equation (4) is simply the sum of (2) and (3).

In order to simultaneously minimize $\boldsymbol{S}_W$ and maximize $\boldsymbol{S}_B$, we consider maximizing the criterion function known as the generalized Fisher linear discriminant function or Rayleigh quotient given by

$$J(\boldsymbol{W}) = tr\frac{\boldsymbol{W}^T\boldsymbol{S}_B\boldsymbol{W}}{\boldsymbol{W}^T\boldsymbol{S}_W\boldsymbol{W}}. \tag{5}$$

In general, solving the maximization of (5) over matrix $\boldsymbol{W}$ is not easy. However, if let $\boldsymbol{W}^*$ be the optimal solution and $\boldsymbol{w}_i^*$ be its ith column vector, it is easy to show that $\boldsymbol{w}_i^*$ is a generalized eigenvector, which corresponds to the ith largest eigenvalue λ_i in the following generalized eigenvalue problem:

$$\boldsymbol{S}_B\boldsymbol{w}_i^* = \lambda_i\boldsymbol{S}_W\boldsymbol{w}_i^* \tag{6}$$

or equivalently

$$\boldsymbol{S}_W^{-1}\boldsymbol{S}_B\boldsymbol{w}_i^* = \lambda_i\boldsymbol{w}_i^* \tag{7}$$

provided that $\boldsymbol{S}_W$ is nonsingular.

It is known that in a c-class classification problem, the rank of $\boldsymbol{S}_B$ is no greater than $c-1$. Accordingly, there are at most $c-1$ nonzero eigenvalues. The larger the eigenvalue, the better the discrimination. In this case, the eigenvectors associated with each nonzero eigenvalue can be used as discriminant vectors with the discriminant power determined by the magnitude of the corresponding eigenvalues. By making use of these $c-1$ eigenvectors, we can actually separate c classes.

B. BS

BS is one of the important tasks in processing multispectral/hyperspectral image data. There are at least two reasons for doing so. One is that since not all bands carry the same amount of information, the bands need to be prioritized according to the significance of their information so data dimensionality reduction can be achieved by removing bands with least priorities. A second reason is that bands adjacent to each other generally portray very high correlation, so a set of selective bands should be able to well represent other bands without loss of much information. It is indeed in many applications that a small number of spectral bands may be sufficiently enough to represent the entire band data.

As mentioned previously, the magnitude of an eigenvalue found by (6) provides an indication of importance of its corresponding eigenvector. Let $\{\lambda_i\}_{i=1}^{c-1}$ be the nonzero eigenvalues obtained by (5) and (6) to discriminate c classes and $\boldsymbol{w}_i^* = (w_{i1}^*, w_{i2}^*, \cdots, w_{il}^*)^T$ is an eigenvector corresponding to the eigenvalue λ_i, where l is the number of bands. From the factor analysis [15], we can define a loading factor matrix Γ by

$$\Gamma = [r_{ik}]_{\substack{i\in\{1,2,\cdots,c-1\}\\ k\in\{1,2,\cdots,l\}}} \tag{8}$$

where $r_{ik} = \sqrt{\lambda_i}(w_{ik}/\|\boldsymbol{w}_i\|)$ is called the loading factor of canonical component i in the kth band and $\|\cdot\|$ is the L^2-norm operator. (Note that we use the terminology of canonical component for CA to correspond to the principal component used for PCA.) Apparently, $\sum_{k=1}^{l} r_{ik}^2 = \lambda_i$. For each band $k \in \{1, 2, \cdots, l\}$, we can further define the discriminant power associated with the kth band by

$$\rho_k \equiv \sum_{i=1}^{c-1} r_{ik}^2 \tag{9}$$

which is the sum of r_{ik} over c-1 canonical components. From (9), a larger value of ρ_k implies more significance of the kth band. Without loss of generality, we assume that $\{\rho_k\}_{k=1}^{l}$ is arranged in decreasing order, i.e., $\rho_1 \geq \rho_2 \geq \cdots \geq \rho_l$. As a result, the original bands can be prioritized in accordance with their discriminant powers so that the first band has the largest discriminant power, then the second band, the third band, etc., until the last band, which has the least discriminant power.

Assume that all bands are prioritized according to (9) and m is the number of bands selected from band 1 up to band m for classification. In order to measure the m-band performance, we use (9) to define a discriminant power probability (DPP) $_m$ associated with the m-BS as

$$\text{DPP}_m = \frac{\sum_{k=1}^{m}\rho_k}{\sum_{k=1}^{l}\rho_k}. \tag{10}$$

It is obvious that $\sum_{k=1}^{l}\rho_k = \sum_{i=1}^{c-1}\lambda_i$ is the total sum of all eigenvalues, which implies that if all bands are used for classification, the DPP must be one, i.e., $\text{DPP}_l = 1$. Also, the DPP DPP_m is the monotonically increasing function of m, the number of bands to be used for classification. So, DPP_m does provide a good criterion to measure the classification performance produced by m bands. For example, if $\text{DPP}_m = 0.9$, it indicates that using the first m bands for classification yields 90% performance compared to that using entire l bands. Therefore, the DPP_m is a logical and appropriate choice for our BS criterion.

C. FSE

One of goals of the CA approach, used for developing the BS algorithm in the previous section, was to select desired bands based on spectral correlation, but no spectral compression was done during BS. In this subsection, CA is applied again to exploit the spectral correlation of the selected data for FSE so that a further data dimensionality reduction can be accomplished. Because the CA-generated eigenvectors could be used as discriminant vectors for classification as described in the BS algorithm, they can be also used as feature vectors for data dimensionality reduction. This is a common technique widely used in pattern classification literature [3], [5], [6], [10], [14], [21]. In this case, the unwanted spectral redundant feature vectors can be removed by appropriately selecting a number of desired features for next stage ML classification. Such CA-based feature selection will mitigate the Hughes phenomenon [22], an effect resulting from a small size of training samples.

Although CA was used in our BS and FSE, it is not the only transformation that can be used for this purpose. A

decision boundary-based method suggested in [21] can be another choice. However, there are advantages of using CA, such as its simplicity and popularity. More importantly, it can be easily implemented as a routine in most software programs, which allows the proposed BS algorithm with FSE to be incorporated into or used in conjunction with most of remote sensing software packages.

III. Fast Recursive Classification Algorithm

A. MLC

As assumed before, let $\omega_1, \omega_2, \cdots, \omega_c$ denote c-distinct classes. We also assume that the conditional-class probability distribution $p_i(\boldsymbol{x})$ for each $i \in \{1, 2, \cdots, c\}$ are Gaussian distributed. Then the MLC is given by

$$\boldsymbol{x} \in \omega_i \qquad \text{if } p_i(\boldsymbol{x}) = \max_{t=1,2,\cdots,c} p_t(\boldsymbol{x}) \tag{11}$$

where $p_i(\boldsymbol{x})$ is given by

$$p_i(\boldsymbol{x}) = \frac{1}{(2\pi)^{l/2}|\Sigma_i|^{1/2}} \exp\left[-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)\right] \tag{12}$$

with $\boldsymbol{m}_i$ and Σ_i denoted by the mean vector and covariance matrix of the ith class, respectively. Σ_i^{-1} and $|\Sigma_i|$ are the inverse and determinant of Σ_i, respectively.

Taking the logarithm of both sides of (12) and using the MLC given by (11) yields the following decision rule:

$$\boldsymbol{x} \in \omega_i \qquad \text{if } d_i(\boldsymbol{x}) = \max_{t=1,2,\cdots,c} d_t(\boldsymbol{x}) \tag{13}$$

where

$$d_i(\boldsymbol{x}) = \log|\Sigma_i| + Q_i(\boldsymbol{x}) \tag{14}$$

$$Q_i(\boldsymbol{x}) = (\boldsymbol{x}-\boldsymbol{m}_i)^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i). \tag{15}$$

It is obvious from (13)–(15) that the most intensive computation for $d_i(\boldsymbol{x})$ is the quadratic term $Q_i(\boldsymbol{x})$. A direct calculation of $Q_i(\boldsymbol{x})$ requires $l(l+1)$ multiplications and about the same order of additions. So, classifying a pixel vector $\boldsymbol{x}$ into one of c classes needs an order of $c \cdot l \cdot (l+1)$ multiplications.

B. Recursive Implementation of MLC

According to (14) and (15), the quadratic term $Q_i(\boldsymbol{x})$ in the MLC accounts for most computations, which are matrix–vector multiplications [23]–[25]. In the following, we develop a recursive version of the MLC to reduce the complexity required for calculating (15).

First of all, we express Σ_n in terms of Σ_{n-1}, i.e.,

$$\Sigma_n = \begin{bmatrix} \Sigma_{n-1} & \mathbf{b}_n \\ \mathbf{b}_n^T & \sigma_{nn} \end{bmatrix}. \tag{16}$$

Let $\boldsymbol{L}_{n-1}\boldsymbol{L}_{n-1}^T$ be the Cholesky decomposition of the covariance matrix Σ_{n-1} where $\boldsymbol{L}_{n-1}$ is a lower triangular matrix. Using the Cholesky decomposition, we can reexpress Σ_n as

$$\Sigma_n = \boldsymbol{L}_n\boldsymbol{L}_n^T = \begin{bmatrix} \boldsymbol{L}_{n-1} & \mathbf{0} \\ \boldsymbol{p}_n^T & \nu_{nn} \end{bmatrix}\begin{bmatrix} \boldsymbol{L}_{n-1}^T & \boldsymbol{p}_n \\ \mathbf{0} & \nu_{nn} \end{bmatrix}. \tag{17}$$

Equating (16) and (17) yields

$$\begin{aligned} \boldsymbol{L}_{n-1}\boldsymbol{p}_n &= \boldsymbol{b}_n \\ \boldsymbol{p}_n^T\boldsymbol{p}_n + \nu_{nn}^2 &= \sigma_{nn}. \end{aligned} \tag{18}$$

Substituting (17) into (15) and letting $\boldsymbol{z}_k = \boldsymbol{x}_k - \boldsymbol{m}_k$, the quadratic term $Q_i(\boldsymbol{x})$ in (15) can be reexpressed as

$$Q_i(\boldsymbol{x}) = \boldsymbol{z}^T\boldsymbol{L}_i^{-T}\boldsymbol{L}_i\boldsymbol{z} = \boldsymbol{y}^T\boldsymbol{y} = \sum_{k=1}^{l} y_k^2 \tag{19}$$

where $\boldsymbol{y}$ can be calculated by a forward substitution using $\boldsymbol{z} = \boldsymbol{L}\boldsymbol{y}$ of the form

$$z_k = \sum_{n=1}^{k} \nu_{kn} y_n \tag{20}$$

Since computing (19) requires calculation of y_k from (20) for each k, this results in total multiplications for one class, which can be calculated by $\sum_{k=1}^{l}(k+1) = (1/2)l(l+3)$.

C. An Improved RMLC Using a Winograd's Algorithm

In order to further reduce the computation of the quadratic term (19) in the discriminant function, we use a Winograd's algorithm [16]–[18] in the recursive MLC developed in the last section.

By making use of the following identity:

$$x_1y_1 + x_2y_2 = (x_1+y_2)(x_2+y_1) - x_1x_2 - y_1y_2$$

a Winograd's identity for the multiplication of two N-dimensional vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ can be obtained by the following:

$$\begin{aligned} \boldsymbol{A}^T\boldsymbol{B} = \sum_{k=1}^{2M} a_kb_k &= \sum_{u=1}^{M}(a_{2u-1}+b_{2u})(a_{2u}+b_{2u-1}) \\ &\quad - \sum_{u=1}^{M}(a_{2u-1}a_{2u}) - \sum_{u=1}^{M}(b_{2u-1}b_{2u}) \end{aligned} \tag{21}$$

where $N = 2M$ (i.e., N is an even number). If N is odd, we simply add an extra zero to both vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ of (21) to make N even so that (21) is still applicable.

The advantage of using (21) is that the number of multiplications is reduced to half of that required by the direct computation, since the second and third terms can be precomputed and stored in the beginning of the computation. Furthermore, since $\boldsymbol{y}$ in (19) is computed by a forward substitution, according to $\boldsymbol{z} = \boldsymbol{L}\boldsymbol{y}$, the computation of (19) can be speeded up with the help of the Winograd's identity, given in (21). The procedure can be derived as follows.

From (20), a close-form solution to the equation $\boldsymbol{z} = \boldsymbol{L}\boldsymbol{y}$ is given by

$$y_n = \frac{z_n - \sum_{\kappa=1}^{n-1}\nu_{n\kappa}y_\kappa}{\nu_{nn}}. \tag{22}$$

TABLE I
COMPUTATION OF $Q(\boldsymbol{x})$ AT THE CLASSIFICATION STAGE USING THE PROPOSED RMLC

Calculating $Q(\mathbf{x})$	Multiplication number for the n th iteration	Multiplication number for a class
n is odd (From (23),(24) and (19))	(23): 1 (24) : $(n-1)/2+1$ (19) : 1 total : $(n-1)/2+3$	$\sum_{n=1}^{l}(\frac{n-1}{2}+3)$
n is even (From (25),(26) and (19))	(25): 0 (26) : $(n/2)+1$ (19) : 1 total : $(n/2)+2$	$\sum_{n=1}^{l}(\frac{n}{2}+2)$
Average	$(n/2)+(9/4)$	$(1/4)(l^2+10l)$

Then, y_n can be computed by the following alternate calculations of (22), according to whether n is even or odd.

1) If n is an odd number, i.e., $n=2m+1$, let

$$p(n)=\sum_{u=1}^{m}\nu_{n,2u}\nu_{n,2u-1}$$

and

$$q(n)=q(n-1)+y_{n-1}y_{n-2} \tag{23}$$

with the initial condition $q(0)=0$, then y_n in (22) can be calculated as in (24), shown at the bottom of the page.

2) If n is an even number, i.e., $n=2m$, let

$$p(n)=\sum_{u=1}^{m-1}\nu_{n,2u}\nu_{n,2u-1} \quad \text{and} \quad q(n)=q(n-1) \tag{25}$$

with the initial condition $q(0)=0$, then y_n in (23) can be calculated as in (26), shown at the bottom of the page.

It should be noted that in the above algorithm, the $p(n)$ only needs to be calculated once in the beginning of the computing process, then it will be treated as a constant in subsequent calculations. Therefore, the calculation for each additional feature requires only $(n/2)+(9/4)$ multiplications. As a result, the total number of multiplications required for one class is $(1/4)(l^2+10)$. The number of multiplications required to compute $Q(\boldsymbol{x})$ in (19) is detailed in Table I.

It is worth noting that the Winograd's algorithm described above cuts the number of required multiplications in half, at the expense of a few additions. Such computation reduction is easily justified because a multiplication requires more computing time than an addition, particularly in hardware implementation. The speed-up rate can be calculated by

$$\text{speed-up rate}=\frac{l(l+1)}{(1/4)(l^2+10l)}=\frac{4(l+1)}{l+10}$$

D. A Sequential Partial Sum Approach

From the Cholesky decomposition of the covariance matrix Σ given by (17), the determinant of Σ is easily calculated by

$$|\Sigma|=|\boldsymbol{L}|\,|\boldsymbol{L}^T|=\prod_{k=1}^{l}\nu_{kk}^2. \tag{27}$$

Taking the logarithm of both sides of (27), we obtain

$$\ln|\Sigma|=\ln\left(\prod_{k=1}^{l}\nu_{kk}^2\right)=\sum_{k=1}^{l}2\ln|\nu_{kk}|. \tag{28}$$

Substituting (19) and (28) into (14) results in

$$\begin{aligned} d_i(\boldsymbol{x}) &= \log|\Sigma_i|+Q_i(\boldsymbol{x}) \\ &= \sum_{k=1}^{l}2\ln|\nu_{i,kk}|+\sum_{k=1}^{l}y_{i,k}^2=a_{i,l} \end{aligned} \tag{29}$$

where

$$a_{i,k}=\sum_{t=1}^{k}2\ln|\nu_{i,tt}|+\sum_{t=1}^{k}y_{i,t}^2 \tag{30}$$

is called the kth partial sum for class i, which adds $\nu_{i,tt}$ and $y_{i,tt}^2$ from $t=1$ up to k, and $\{a_{i,t}\}_{t=1}^{l}$ is a monotonically increasing sequence for each i, i.e., $a_{i,t}\geq a_{i,s}$ if $t>s$.

The formulation in (29) is very useful because the discriminant function $d_i(\boldsymbol{x})$ can be calculated by a sequential partial sum $\{a_{i,t}\}_{t=1}^{l}$, given by (30). Since the classification of a test pattern $\boldsymbol{x}$ into class i is based on the ith discriminant function $d_i(\boldsymbol{x})$, given by (13), a more efficient classification rule can be now obtained by replacing (13) with (29) through the partial sums defined by (30) [24], [25]. For example, let t^* be the class assigned to the test pattern $\boldsymbol{x}$. Class 1 through class t were tested, i.e., $d_{t^*}(\boldsymbol{x})$ achieved the minimum among $\{\omega_i\}_{i=1}^{t}$ and $t^*\in\{1,2,\cdots,t\}$. For the next class $t+1$, rather than directly computing $d_{t+1}(\boldsymbol{x})$, we compute $a_{t+1,k}$, given

$$y_n=\frac{z_n-\sum_{u=1}^{m}(\nu_{n,2u-1}+y_{2u})(\nu_{n,2u}+y_{2u-1})+p(n)+q(n)}{\nu_{nn}} \tag{24}$$

$$y_n=\frac{z_n-\sum_{u=1}^{m-1}(\nu_{n,2u-1}+y_{2u})(\nu_{n,2u}+y_{2u-1})+\nu_{n,n-1}y_{n-1}+p(n)+q(n)}{\nu_{nn}} \tag{26}$$

TABLE II
PARAMETERS OF THE FSS

Number of Bands	60 bands
Spectral Converge	0.4 - 2.4 μm
Altitude	60 m
IFOV	25 m

TABLE III
CLASS DESCRIPTION OF DATA SET 1, COLLECTED IN HAND COUNTY, SD

Species	Date	No. of Samples
Spring Wheat	10/18/77	313
Spring Wheat	5/15/78	474
Spring Wheat	6/02/78	517
Spring Wheat	7/09/78	454
Spring Wheat	7/26/78	518
Spring Wheat	8/16/78	464
Spring Wheat	9/21/78	469
Spring Wheat	10/26/78	441

TABLE IV
CLASS DESCRIPTION OF DATA SET 2, COLLECTED IN HAND COUNTY, SD

Species	Date	No. of Samples
Native Grass Pas	10/18/77	183
Native Grass Pas	5/15/78	196
Native Grass Pas	6/02/78	214
Native Grass Pas	7/09/78	170
Native Grass Pas	7/26/78	217
Native Grass Pas	8/16/78	212

by (30), for each k, then compare it to $d_{t^*}(\boldsymbol{x})$ in the following procedure.

Step 1) Check if $a_{t+1,k} \le d_{t^*}(\boldsymbol{x})$ for $k < l$. If it is not, go to Step 3. Otherwise, continue to compute $a_{t+1,k+1}$ and go to Step 2.

Step 2) Check if $a_{t+1,k+1} \le d_{t^*}(\boldsymbol{x})$ for $k+1 < l$. If it is not, go to Step 3. Otherwise, repeat Steps 1 and 2 until $k+1 = l$, in which case, $\boldsymbol{x}$ will be assigned to class $t+1$ and go to Step 4.

Step 3) In this case, $a_{t+1,k} > d_{t^*}(\boldsymbol{x})$ for some $k < l$, $d_{t+1}(\boldsymbol{x}) = a_{t+1,l} \ge a_{t+1,k} > d_{t^*}(x)$, the $\boldsymbol{x}$ cannot be assigned to class $t+1$ and go to Step 4.

Step 4) Move on to next class $t+2$ and compute $a_{t+2,1}$ for the next class $t+2$. Repeat Steps 1–4.

The above four-step process will be terminated after all classes are exhausted. In this case, class c^* is the correct class to which the test pattern is supposed to be assigned. A flowchart of this procedure is shown in Fig. 2. As demonstrated in experiments, this algorithm is very efficient and achieves substantial saving in computation, particularly, for the case in which a large number of features need to be used or the class differences are large.

Another advantage of the proposed algorithm using (29) and (30) is to alleviate the Hughes phenomenon, which often occurs in the processing of high-dimensional remote sensing data.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, experiments are conducted based on the FSS data with 60 spectral bands [26]. The major parameters of the

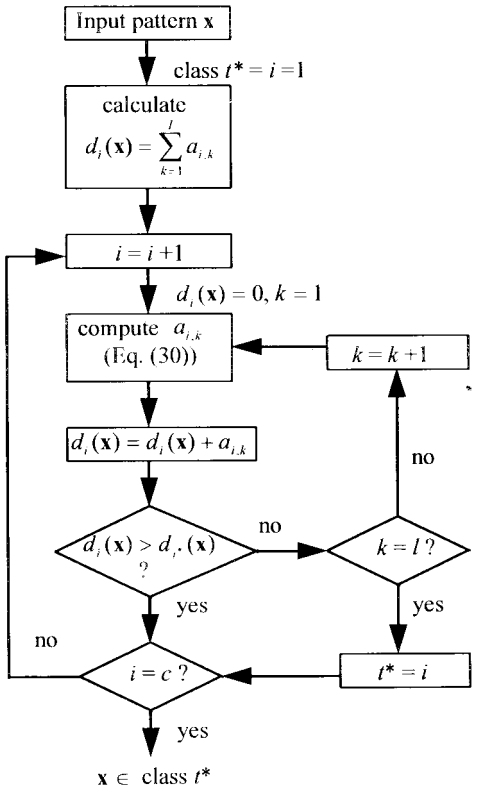

Fig. 2. Flowchart of the proposed fast recursive ML algorithm.

TABLE V
CLASS DESCRIPTION OF DATA SET 3

Species	Date	Location	No. of Samples
Spring Wheat	7/26/78	Hand Co. SD.	518
Spring Wheat	6/02/78	Hand Co. SD.	517
Spring Wheat	5/15/78	Hand Co. SD.	474
Spring Wheat	9/21/78	Hand Co. SD.	469
Spring Wheat	8/16/78	Hand Co. SD.	464
Spring Wheat	7/09/78	Hand Co. SD.	454
Spring Wheat	10/26/78	Hand Co. SD.	441
Spring Wheat	9/28/76	Hand Co. SD.	414
Winter Wheat	10/26/78	Finney Co. KS.	393
Spring Wheat	10/18/77	Hand Co. SD.	313
Winter Wheat	9/20/77	Finney Co. KS.	292
Grain Sorghum	3/08/77	Finney Co. KS.	279
Grain Sorghum	9/28/76	Finney Co. KS.	277
Pasture	10/26/78	Finney Co. KS.	217
Summer Follow	8/16/78	Finney Co. KS.	216
Native Grass Pas	6/02/78	Hand Co. SD.	214
Summer Follow	5/03/77	Finney Co. KS.	211
Native Grass Pas	5/15/78	Hand Co. SD.	196
Grain Sorghum	6/26/77	Finney Co. KS.	157

FSS data are listed in Table II. Since bands corresponding to water absorption regions have no useful energy, they are removed prior to processing, which leave 56 bands in this study. Three data sets were used and their species, data collection dates, locations, and the number of samples collected are tabulated in Tables III–V. Data set 1 contains eight classes and Data set 2 has six classes with their class means shown in Figs. 3 and 4, respectively. From Figs. 3 and 4, we can see that the mean differences between classes in

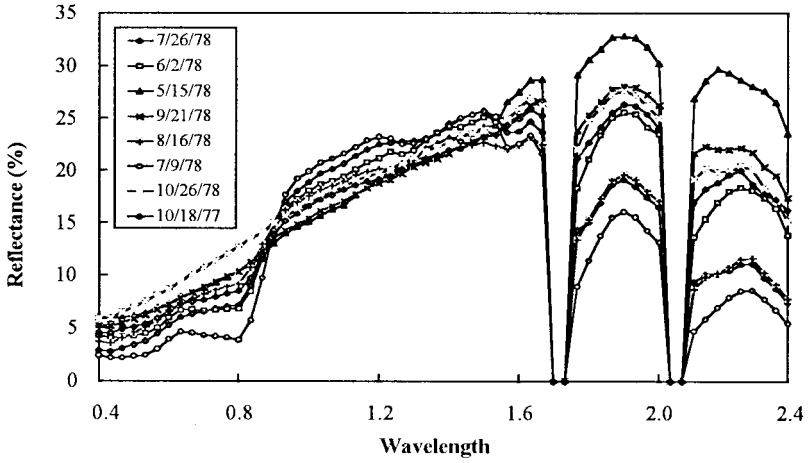

Fig. 3. Class means of data set 1 with eight multitemporal classes.

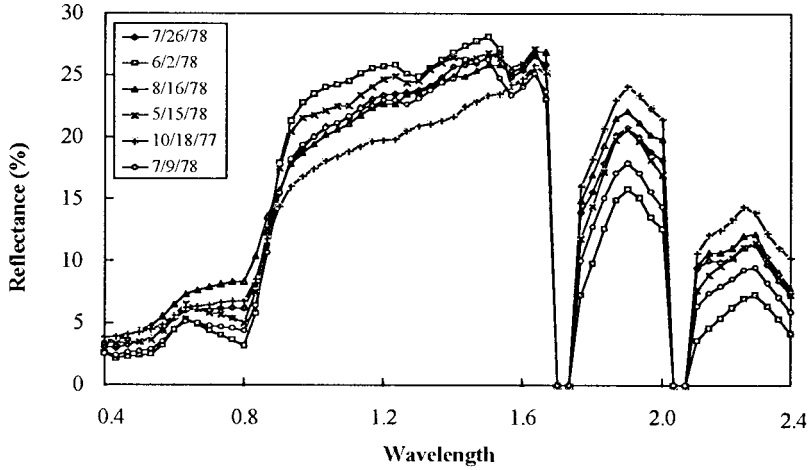

Fig. 4. Class means of data set 2 with six multitemporal classes.

Data set 2 are smaller than that in Data set 1. Using these two data sets can illustrate the role of the mean difference in classification. Unlike Data sets 1 and 2, which contain only a single substance, Data set 3 has 19 classes, which contain six different substances. By means of Data set 3, we can show that the proposed classification method can also perform well for the spatially and temporally combined data. In order to estimate the Gaussian parameters used in the MLC, half of the data were used for training, while the other half were used as test samples. All programs were written in FORTRAN language and compiled with Microsoft Fortran 5.1 compiler on an IBM PC-486.

The experiments conducted in this section were based on these three data sets and particularly designed to make various comparisons among several variations of MLC's. Seven ML classifiers compared in the experiments are a) conventional Gaussian MLC using all bands, b) RMLC using all bands, described in Fig. 2, c) Gaussian MLC with uniform band (UB-MLC), where UB selects the bands uniformly apart, d) Gaussian RMLC with UB (UB-RMLC), e) conventional Gaussian MLC with BS (BS-MLC), where BS is the BS algorithm described in Section II-B, f) Gaussian RMLC with BS (BS-RMLC), and g) Gaussian RMLC with BS and FES (BSFES-RMLC), where BSFES is the BS with FES, given in Sections II-B and II-C. Among these seven MLC's, BSFES-RMLC specified by g) is the two-stage ML classification method proposed in this paper. The reasons for choosing these classifiers for comparisons are given as follows.

1) Comparing RMLC against MLC demonstrates the computation efficiency and speed-up rate if an MLC is implemented recursively.

TABLE VI
PERFORMANCE COMPARISON AMONG 7 MLC'S FOR DATA SET 1

Type of classifiers	Number of bands used	Classification accuracy (%)	Processing time (sec)	Speed-up rate
MLC	56	98.36/96.32	432.87/437.02	----
RMLC			99.13/99.08	4.37/4.41
UB-MLC	15	92.46/89.37	36.96/36.65	11.71/11.92
UB-RMLC			8.42/8.33	51.41/52.46
BS-MLC	15	98.35/96.10	36.79/36.57	11.77/11.95
BS-RMLC			8.41/8.32	51.47/52.46
BSFES-RMLC	7	96.47/93.38	3.06 / 3.02	141.46/144.71

(where */* = performance of the training data / performance of the test data)

TABLE VII
PERFORMANCE COMPARISON AMONG 7 MLC'S FOR DATA SET 2

Type of classifiers	Number of bands used	Classification accuracy (%)	Processing time (sec)	Speed-up rate
MLC	56	99.14/82.02	107.10/106.92	----
RMLC			31.33/31.31	3.42/3.42
UB-MLC	19	94.92/85.10	23.40/23.29	4.58/4.59
UB-RMLC			6.85/6.80	15.64/15.72
BS-MLC	19	98.70/92.19	23.42/23.30	4.57/4.59
BS-RMLC			6.85/6.79	15.64/15.7
BSFES-RMLC	5	93.36/89.93	1.25 / 1.21	85.68/88.36

TABLE VIII
PERFORMANCE COMPARISON AMONG 7 MLC'S FOR DATA SET 3

Type of classifiers	Number of bands used	Classification accuracy (%)	Processing time (sec)	Speed-up rate
MLC	56	99.01/92.36	547.06/545.32	----
RMLC			127.35/125.87	4.30/4.33
UB-MLC	26	96.13/93.21	118.14/117.55	4.63/4.64
UB-RMLC			27.22/26.97	20.10/20.22
BS-MLC	26	98.90/95.35	117.91/117.66	4.64/4.63
BS-RMLC			27.23/26.99	20.09/20.20
BSFES-RMLC	18	97.65/94.42	18.26/18.19	30/30

2) Comparing UB-MLC against MLC with full bands demonstrates the classification accuracy, the computation efficiency and saving if a UB is used instead of entire bands.
3) Comparing BS-MLC and BS-RMLC against UB-MLC and UB-RMLC demonstrates the classification accuracy, the computation efficiency and saving if a BS is used rather than UB.
4) Comparing BSFES-RMLC against BS-RMLC demonstrates the classification accuracy, the computation efficiency, and saving if an extra FSE procedure is also used in conjunction with BS-RMLC.

Tables VI–VIII tabulate all detailed analysis for three data sets, respectively. The classification accuracy is calculated by the mean of correct classification rates over all classes, where a correct classification is made if a data sample is correctly classified into the class to which it belongs. It is noted that this classification accuracy is a random variable, whose value varies from trial to trial. However, the variance of this random variable will become small as the size of samples used is increased.

It should be noted that the UB described above is implemented in a natural and intuitive way. This is based on the concept that selecting widespread bands is more representative than adjacent bands, due to high spectral correlation. It is very similar to downsampling (undersampling), a technique widely used in communications and signal processing. It requires no extra processing time. For instance, if only 15 bands need to be selected from 56 FSS bands, UB will select bands that are four-bands apart. Namely, the selected bands uniformly cover the entire spectral range. Unfortunately, a major issue arising in UB is the determination of the number of bands

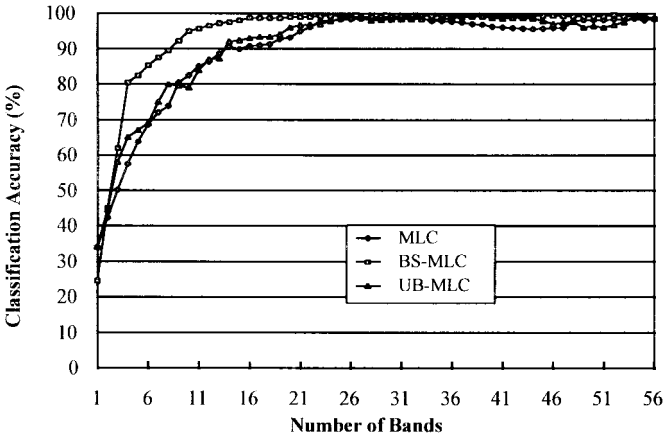

Fig. 5. Performance comparison among methods with no BS (MLC), uniform BS (UB-MLC), and the proposed BS (BS-MLC) for data set 1 (training data).

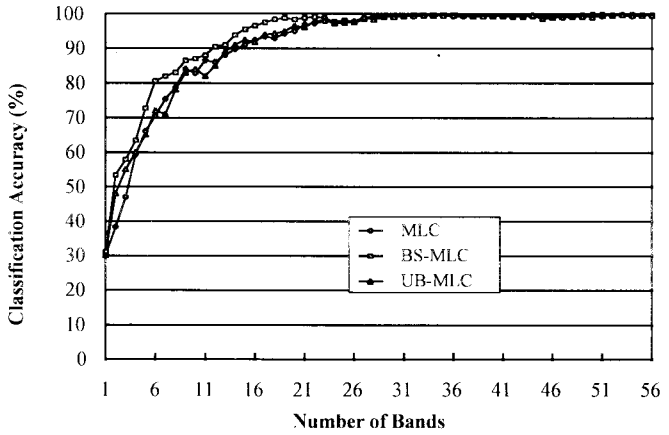

Fig. 6. Performance comparison among methods with MLC, UB-MLC, and BS-MLC for data set 2 (training data).

prior to selection. It may be done by guess or trial and error, but obviously, it is not an optimal way to do it. In contrast to UB, the proposed BS suggests a systematic scheme to search for desired bands. It first calculates the discriminant power given by (9) for each band and prioritizes all bands in order of decreasing discriminant power. Then for each m, it further computes (10) to produce the discriminant power probability DPP_m, which provides a measure of classification power with m bands used for classification. This selection procedure is designed by using CA, which minimizes misclassification error, based on second-order data statistics. As shown in Figs. 5–7 for training samples and Figs. 8–10 for test samples, such CA-based BS performs better than the UB in both cases in terms of classification accuracy by using the same number of bands. This can be anticipated because the CA-based BS algorithm took into account the statistical spectral correlation, while UB simply downsamples the spectral bands uniformly without actually calculating data statistics, in which case, it views the bands to be equally important. On the other hand, we can see in Figs. 5–7 that the performance of UB does not make much difference in training data compared to that with no BS, which assumes that the bands are processed according

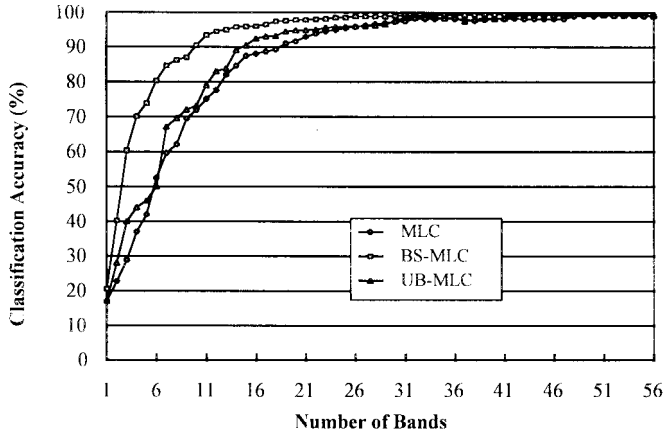

Fig. 7. Performance comparison among methods with MLC, UB-MLC, and BS-MLC for data set 3 (training data).

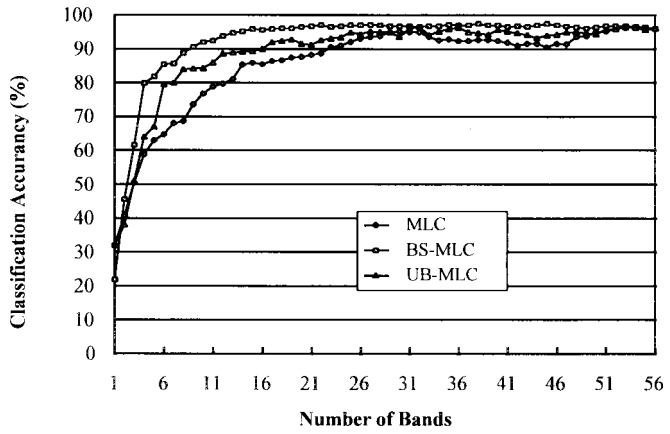

Fig. 8. Performance comparison among methods with MLC, UB-MLC, and BS-MLC for data set 1 (test data).

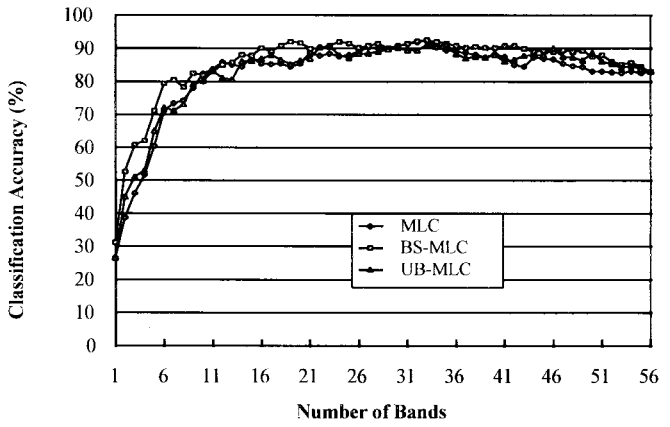

Fig. 9. Performance comparison among methods with MLC, UB-MLC, and BS-MLC for data set 2 (test data).

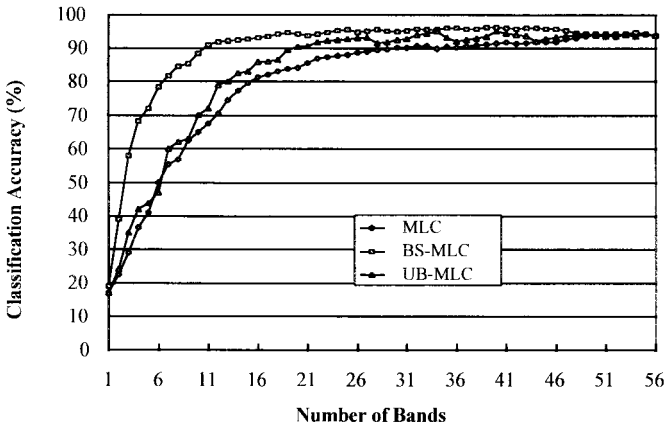

Fig. 10. Performance comparison among methods with MLC, UB-MLC, and BS-MLC for data set 3 (test data).

to increasing spectral frequency. However, UB does perform slightly better than no BS in test data, as shown in Figs. 8–10. It should be noted that the number of bands to be used for BS can be determined from training samples. From Figs. 5–7, we observe that the classification rate curves saturate gradually when their band number reach certain numbers. In the training phase, these numbers are 15, 19, and 26 for the data sets 1, 2, and 3, respectively. Such numbers provide an approximation to intrinsic dimensionality of the data. Of course, the number of bands to be selected in classification also depends on the characteristics and the number of classes.

It is also worth noting that the Hughes phenomenon appears pronounced in data set 2. This is because the mean differences between the various classes in data set 2 are relatively small and make discrimination difficult among classes. Therefore, the parameters obtained from the training data cannot sufficiently represent their characteristics if the size of the training samples is small. In this case, the effectiveness of the BS is decreased and the between-class scatter matrix $\boldsymbol{S}_B$ tends to approach zero matrix.

In addition to the gains offered by BS and FSE, another substantial saving in computation can be obtained by the fast recursive MLC, which incorporates the Cholesky decomposition, a Winograd's algorithm, and the sequential partial sum approach delineated in Section III. The recursive algorithm can be implemented independently with/without BS, thus, it is not necessarily limited to the application presented here. As shown in Tables VI–VIII, the speed-up rate of RMLC compared to that of MLC is approximately four. As expected, the saving in processing time is greatly improved in data sets 1 and 3 (Tables VI and VIII) with large class mean differences, but less significantly for data set 2 (Table VII) when the class means are close.

Finally, from Tables VI–VIII, we conclude that BSFES-RMLC has three advantages: 1) it reduces the bandwidth of transmission via BS at the expense of a few additional computations, 2) in the light of data dimensionality reduction and FSE, it reduces the Hughes phenomenon, which often occurs in the high-dimensional remotely sensed data, and 3) by making use of a recursive algorithm, it reduces the processing time significantly, thus, improving computational efficiency. These advantages enable BSFES-RMLC to handle high-dimensional remotely sensed data efficiently in terms of computation and processing time.

V. Conclusions

In this paper, a fast two-stage classification method was presented to cope with two problems generally encountered

in high-dimensional remote sensing data, a large bandwidth of transmission and excessive processing time required for classification. The proposed method consists of two stages: a BS algorithm along with FSE employed in the first stage followed by a fast RMLC in a second stage. The BS was developed on the basis of the CA and loading factors, from which the discriminant power could be defined and assigned to each band. Using their discriminant powers, all bands were prioritized for the purpose of BS. Since the BS had done nothing about spectral compression, the CA was applied again to the selected band data so that desired feature vectors could be extracted and selected for next stage classification to achieve a further data dimensionality reduction. In the phase of classification, a fast recursive MLC was designed by taking advantage of a Winograd's algorithm and a sequential partial sum procedure. As a consequence, a substantial saving in computation can be accomplished. Experimental results show that the proposed classifier cuts down computational cost significantly and performs faster than the conventional MLC by a factor of 30–145. In addition, due to the reduction of the Hughes phenomenon by the proposed sequential partial sum procedure, the RMLC produces a higher discrimination rate and improves classification performance.

ACKNOWLEDGMENT

The authors would like to thank Prof. D. A. Landgrebe of Purdue University for providing the FSS data sets and two anonymous reviewers' suggestions for improving the readability and presentation of this paper.

REFERENCES

[1] G. Vane and A. F. H. Goetz, "Terrestrial imaging spectroscopy," *Remote Sens. Environ.*, vol. 24, pp. 1–29, 1988.
[2] K. Fukanaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1990.
[3] R. A. Schowengerdt, *Techniques for Image Processing and Classifications in Remote Sensing*. New York: Academic, 1983.
[4] W. Eppler, "Canonical analysis for increased classification speed and channel selection," *IEEE Trans. Geosci. Electron.*, vol. GE-14, pp. 26–33, 1976.
[5] P. H. Swain and R. C. King, "Two effective feature selection criteria for multispectral remote sensing," in *Proc. First Int. Joint Conf. Pattern Recognition*, 1973, pp. 536–540.
[6] A. Richards, *Remote Sensing Digital Image Analysis*, 2nd ed. New York: Springer-Verlag, 1993.
[7] C. Lee and D. A. Landgrebe, "Fast likelihood classifier," *IEEE Trans. Geosci. Remote Sensing*, vol. 29, pp. 509–517, 1991.
[8] X. Jia and J. A. Richards, "Efficient maximum likelihood classification for imaging spectrometer data sets," *IEEE Trans. Geosci. Remote Sensing*, vol. 32, pp. 274–281, 1994.
[9] J. Settle and S. A. Briggs, "Fast maximum likelihood classification of remotely-sensed imagery," *Int. J. Remote Sensing*, vol. 8, pp. 723–734, 1987.
[10] H. Foley and J. W. Sammon, "An optimal set of discriminant vector," *IEEE Trans. Comput.*, vol. C-24, pp. 281–289, 1975.
[11] A. Campbell and W. R. Atchley, "The geometry of canonical variate analysis," *Syst. Zoology*, vol. 30, pp. 268–280, 1981.
[12] B. Kim and D. A. Landgrebe, "Hierarchical classifier design in high-dimensional, numerous class cases," *IEEE Trans. Geosci. Remote Sensing*, vol. 29, pp. 518–528, 1991.
[13] W. Mansel, W. J. Kramber, and J. K. Lee, "Optimum band selection for supervised classification of multispectral data," *Photogramm. Eng. Remote Sensing*, vol. 56, pp. 55–60, 1990.
[14] O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[15] A. Basilevsky, *Statistical Factor Analysis and Related Method: Theory and Application*. New York: Wiley, 1994.
[16] R. Harter, "The optimality of Winograd's formula," *Commun. ACM*, vol. 15, pp. 352–353, 1972.
[17] L. Kronsjo, *Algorithm: Their Complexity and Efficiency*, 2nd ed. Singapore: Wiley, 1987.
[18] B. Venkateswarlu and P. S. V. S. K. Raju, "Winograd's method: A perspective for some pattern recognition problems," *Pattern Recognit. Lett.*, vol. 15, pp. 105–109, 1994.
[19] H. Swain and S. M. Davis, *Remote Sensing: The Quantitative Approach*. New York: McGraw-Hill, 1978.
[20] C. Conese and F. Maselli, "Selection of optimum bands from TM scenes through mutual information analysis," *ISPRS J. Photogramm. Remote Sens.*, vol. 48, pp. 2–11, 1993.
[21] Lee and D. A. Landgrebe, "Feature extraction based on decision boundaries," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 388–400, 1993.
[22] F. Hughes, "On the accuracy of statistical pattern recognizers," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 55–63, 1968.
[23] B. Venkateswarlu and P. S. V. S. K. Raju, "Three stage ML classifier," *Pattern Recognition*, vol. 24, pp. 1113–1116, 1991.
[24] E. Hodgson, "Reducing the computational requirements of the minimum distance classifier," *Remote Sens. Environ.*, vol. 25, pp. 117–128, 1988.
[25] P. M. Mather, "Computationally-efficient maximum likelihood classifier employing prior probabilities for remotely-sensed data," *Int. J. Remote Sensing*, vol. 6, pp. 369–376, 1985.
[26] L. Biehl *et al.*, "A crops and soils data base for scene radiation research," in *Proc. Machine Processing Remotely Sensed Data Symp.*, 1982, pp. 169–177.

Te-Ming Tu, for a photograph and biography, see this issue, p. 181.

Chin-Hsing Chen, for a photograph and biography, see this issue, p. 181.

Jiunn-Lin Wu was born in Kaohsiung, Taiwan, R.O.C., in 1971. He received the B.S.E.E. and M.S.E.E. degrees from the National Cheng Kung University, Tainan, Taiwan, in 1993 and 1995, respectively, both in electrical engineering. He is currently pursuing the Ph.D. degree at the National Cheng Kung University. His interests include pattern recognition, remote sensing, and image coding.

Chein-I Chang (S'81–M'82–SM'92), for a photograph and biography, see this issue, p. 181.