Fast Algorithm for Updating the Discriminant Vectors of Dual-Space LDA

Wenming Zheng, *Member, IEEE*, and Xiaoou Tang, *Fellow, IEEE*

***Abstract*—Dual-space linear discriminant analysis (DSLDA) is a popular method for discriminant analysis. The basic idea of the DSLDA method is to divide the whole data space into two complementary subspaces, i.e., the range space of the within-class scatter matrix and its complementary space, and then solve the discriminant vectors in each subspace. Hence, the DSLDA method can take full advantage of the discriminant information of the training samples. However, from the computational point of view, the original DSLDA method may not be suitable for online training problems because of its heavy computational cost. To this end, we modify the original DSLDA method and then propose a data order independent incremental algorithm to accurately update the discriminant vectors of the DSLDA method when new samples are inserted into the training data set. We conduct experiments on the AR face database to confirm the better performance of the proposed algorithms in terms of the recognition accuracy and computational efficiency.**

***Index Terms*—Dual-space linear discriminant analysis (DSLDA), feature extraction, incremental linear discriminant analysis.**

I. Introduction

LINEAR discriminant analysis (LDA) [1] is a well-known feature extraction method in statistical pattern recognition. It computes an optimal feature space based on the Fisher's criterion, in which the projections of the training data will have the maximum ratio of the between-class distance to the within-class distance. In many applications, however, LDA often suffers from the so-called undersampled problems [2], where the dimensionality of the input space is larger than the number of available training data points such that the within-class scatter matrix becomes singular. To deal with the undersampled problem, many LDA methods, such as the principal component analysis plus LDA (PCA+LDA) method [3]–[5], the null space method [6], [7], and the direct LDA (DLDA) method [8], had been developed in recent years. However, a common drawback of the above methods is that they solve the discriminant vectors by focusing on a single data subspace rather than the full data space. Therefore, these methods may lose some useful discriminant information [9], [10] to some extent.

The LDA method based on generalized singular value decomposition (LDA/GSVD) [2] is another LDA method recently proposed to overcome the undersampled problem of LDA. Different from the previous LDA methods, LDA/GSVD solves the discriminant vectors in the full data space. It avoids the singularity problem of the scatter matrices by using the Moore–Penrose generalized inverse to replace the inverse of the scatter matrices defined in the Fisher's criterion. Nevertheless, it should be noted that the optimal discriminant vectors of LDA/GSVD lie in the range space of the total-class scatter matrix [11]. If the rank of the range space of the total-class scatter matrix is equal to the sum of the ranks of both the within-class scatter matrix and the between-class scatter matrix, the solutions of the LDA/GSVD method will be equivalent to those of the null space method [7]. In this case, the LDA/GSVD method may suffer from the same drawback of losing some useful discriminant information as the null space method. In [12] and [13], Wang and Tang used a random sampling approach to combine the two subspaces. However, the method needs to train a large number of weak classifiers.

To overcome the undersampled problem and at the same time extract more useful discriminant infromation, a new LDA method, called dual-space LDA (DSLDA), was recently proposed by Wang and Tang [9]. The basic idea of this method is to divide the whole data space into two complementary subspaces, i.e., the range space of the within-class scatter matrix and its complementary subspace, and then solve the discriminant vectors in each subspace. As a result, the DSLDA method can obtain more discriminant vectors than the other methods, and hence can extract more useful discriminant information.

Although the LDA method had been extensively studied during the last decades, only a few papers addressed the computational issues on how to reduce the computational cost of solving the discriminant vectors. One of the computational issues is the fast updating of the discriminant vectors when new training samples are inserted into the training data set. This issue is important for some online learning problems, such as object tracking [14] or face recognition [19]. To design the fast LDA algorithm, Mao and Jain proposed an iterative approach for solving LDA based on a two-layer neural network [15]. Chatterjee and Roychowdhury proposed another neural-network-based algorithm to iteratively solve the LDA discriminant vectors. In 2004, Lin *et al.* [14] proposed an incremental LDA algorithm for object tracking. This algorithm works under the circumstance that only one class of the training data set contains more than one sample. Ye *et al.* [17] proposed another incremental LDA algorithm via QR decomposition

Manuscript received August 12, 2008; revised May 09, 2009. First published June 30, 2009; current version published August 14, 2009. This work was supported in part by the Natural Science Foundation of China under Grant 60503023 and Grant 60872160, and in part by the program for new century excellent talents in university (NCET-05-0467). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Anil Jain.

W. Zheng is with the Key Laboratory of Child Development and Learning Science, Ministry of Education, Research Center for Learning Science, Southeast University, Nanjing, Jiangsu 210096, China (e-mail: wenming_zheng@seu.edu.cn).

X. Tang is with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong (e-mail: xtang@ie.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2009.2025844

1556-6013/$26.00 © 2009 IEEE

(LDA/QR), which solves the discriminant vectors by involving the subspace spanned by the class means to find the optimal discriminant vectors. Pang *et al.* [18] proposed another incremental algorithm for classification of data streams. However, they only addressed the method of updating the scatter matrices. In the above algorithms, a common drawback is that all these algorithms fail to address the undersampled problem of LDA.

To incrementally solve the optimal discriminant vectors of LDA with the undersampled problem, Zheng *et al.* [7] proposed a new LDA algorithm based on the Gram–Schmidt orthogoanlization. The problem of this algorithm is that it only considers the discriminant vectors in the null space of the within-class scatter matrix. Recently, Zhao *et al.* [19] proposed another algorithm based on the fast singular value decomposition (SVD) technique [20] to incrementally update the discriminant vectors of the LDA/GSVD method when new samples are inserted into the training data set. However, it should be noted that the computational reduction of the fast SVD algorithm is due to the use of the approximation trick. In other words, if we want to obtain more accurate SVD updating results, the reduction of the computation will be very limited.

In this paper, we use the data order independent (DOI) incremental algorithm for accurately updating the discriminant vectors of DSLDA when new samples are inserted into the training data set. To this end, we modify the original DSLDA method proposed by Wang and Tang [9] in order to reduce the computational complexity. On the other hand, to incrementally update the discriminant vectors of our modified DSLDA method when new samples are inserted, we first use the Gram–Schmidt orthogonalization technique to incrementally update the orthonormal basis of the range space of the within-class scatter matrix.[1] Then, we propose an efficient algorithm to incrementally update the between-class and within-class scatter matrices when new samples are inserted. Finally, we propose an effective algorithm to incrementally solve the discriminant vectors in each subspace based on the prior computational results.

The remainder of this paper is organized as follows: In Section II, we briefly review the DSLDA method proposed by Wang and Tang [9]. In Section III, we propose a modified DSLDA method and the incremental DSLDA algorithm. In Section IV, we address the classification problem of the DSLDA method. Section V is devoted to the experiments. Finally, we conclude the paper in Section VI.

II. Brief Review of the Original DSLDA Method

Let $\mathbf{X} = \{\mathbf{x}_i^j | j = 1, 2, \ldots, n_i; i = 1, 2, \ldots, c\}$ be a d-dimensional real sample set with n elements, where c is the number of the classes, n_i is the number of the samples of the ith class, and $\mathbf{x}_i^j$ is the jth sample of the ith class. The classical LDA method aims to find an optimal transformation matrix $\mathbf{W}$ that maximizes the Fisher's criterion

$$\mathbf{W} = \arg\max_{\mathbf{W}} \operatorname{trace}((\mathbf{W}^T\mathbf{S}_W\mathbf{W})^{-1}\mathbf{W}^T\mathbf{S}_B\mathbf{W}) \quad (1)$$

where $\operatorname{trace}(\cdot)$ denotes the trace operator, and

$$\mathbf{S}_B = \sum_{i=1}^{c} n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T = \mathbf{H}_B\mathbf{H}_{\mathbf{B}}^T \quad (2)$$

$$\mathbf{S}_W = \sum_{i=1}^{c}\sum_{j=1}^{n_i}(\mathbf{x}_i^j - \mathbf{m}_i)(\mathbf{x}_i^j - \mathbf{m}_i)^T = \mathbf{H}_W\mathbf{H}_W^T \quad (3)$$

are between-class and within-class scatter matrices, where $\mathbf{m}_i = (1/n_i)\sum_{j=1}^{n_i}\mathbf{x}_i^j$ is the mean of the ith class, $\mathbf{m} = (1/n)\sum_{i=1}^{c}\sum_{j=1}^{n_i}\mathbf{x}_i^j$ is the global mean of the data set, and $\mathbf{H}_B$ and $\mathbf{H}_W$ are defined as

$$\mathbf{H}_B = [\sqrt{n_1}(\mathbf{m}_1 - \mathbf{m}), \sqrt{n_2}(\mathbf{m}_2 - \mathbf{m}), \ldots, \sqrt{n_c}(\mathbf{m}_c - \mathbf{m})] \quad (4)$$

$$\mathbf{H}_W = [\mathbf{x}_1^1 - \mathbf{m}_1, \ldots, \mathbf{x}_1^{n_1} - \mathbf{m}_1, \ldots, \mathbf{x}_c^1 - \mathbf{m}_c, \ldots, \mathbf{x}_c^{n_c} - \mathbf{m}_c]. \quad (5)$$

The columns of the optimal transformation matrix $\mathbf{W}$ can be obtained by solving the eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$ [21]. However, as for the case of the undersampled problem, the within-class scatter matrix $\mathbf{S}_W$ becomes singular and the LDA method will fail. To overcome the singularity problem of $\mathbf{S}_W$, Wang and Tang [9] proposed the DSLDA method to solve the optimal discriminant vectors, which can be formulated into solving the following optimization problems:

$$\begin{cases} \mathbf{W}_P = \arg\max \operatorname{trace}((\mathbf{W}_P^T\mathbf{S}_W\mathbf{W}_P)^{-1}\mathbf{W}_P^T\mathbf{S}_B\mathbf{W}_P) \\ |\mathbf{W}_P^T\mathbf{S}_W\mathbf{W}_P| \neq 0 \end{cases} \quad (6)$$

$$\begin{cases} \mathbf{W}_C = \arg\max \operatorname{trace}(\mathbf{W}_C^T\mathbf{S}_B\mathbf{W}_C) \\ \mathbf{W}_C^T\mathbf{S}_W\mathbf{W}_C = \mathbf{0}. \end{cases} \quad (7)$$

The columns of $\mathbf{W}_P$ can be solved by involving the range space of $\mathbf{S}_W$ whereas the columns of $\mathbf{W}_C$ can be solved in the complementary space of the range space of $\mathbf{S}_W$.

Based on the DSLDA method, the classification for a given test sample $\mathbf{x}_{\text{test}}$ can be obtained via the optimization problem (8), shown at the bottom of the page [9], where ρ is average noise spectrum of the eigenvalues in the complementary space of the range space of $\mathbf{S}_W$.

$$c^* = \arg\max_i \left(\|\mathbf{W}_P^T\mathbf{x}_{\text{test}} - \mathbf{W}_P^T\mathbf{m}_i\|^2 + \frac{\|\mathbf{W}_C^T\mathbf{x}_{\text{test}} - \mathbf{W}_C^T\mathbf{m}_i\|^2}{\rho} \right) \quad (8)$$

III. Modified DSLDA Method and the Incremental Algorithm

In this section, we design a DOI incremental algorithm for DSLDA. We first modify the original DSLDA method so as to reduce its computational complexity. In the rest of this paper,

[1] In [7], we have proposed to use the Gram–Schmidt orthogonalization technique to incrementally update the discriminant vectors of the null space LDA method. However, we require the constraint that the training samples are independent. In this paper, we free this constraint.

we make the following conventions: $\overline{\mathbf{S}_W(0)}$ denotes the range space of $\mathbf{S}_W$, and $\mathbf{S}_W(0)$ denotes the orthogonal complementary space of $\overline{\mathbf{S}_W(0)}$.

A. Modified DSLDA Method

From (6) and (7), we observe that the most time-consuming part of the original DSLDA method is to solve the transformation matrix $\mathbf{W}_P$ from the subspace $\overline{\mathbf{S}_W(0)}$. Therefore, to reduce the computational complexity of DSLDA, we should first reduce the computational complexity of solving the discriminant vectors from $\overline{\mathbf{S}_W(0)}$. To this end, we adopt the method of Ye *et al.* [17] using the subspace spanned by the class means of the training samples after projecting onto the subspace $\overline{\mathbf{S}_W(0)}$ to find the optimal discriminant vectors rather than the whole subspace $\overline{\mathbf{S}_W(0)}$, which can greatly reduce the computational complexity.

Let $\mathbf{U}$ be a matrix whose columns form an orthonormal basis of $\overline{\mathbf{S}_W(0)}$, then the projection of the class mean $\mathbf{m}_i$ onto the subspace $\overline{\mathbf{S}_W(0)}$ can be expressed as $\mathbf{U}\mathbf{U}^T\mathbf{m}_i$ $(i = 1, \ldots, c)$. Similarly, let $\mathbf{U}^\perp$ be a matrix whose columns form an orthonormal basis of $\mathbf{S}_W(0)$; then the projection of the class mean $\mathbf{m}_i$ onto the subspace $\mathbf{S}_W(0)$ can be expressed as $\mathbf{U}^\perp{\mathbf{U}^\perp}^T\mathbf{m}_i$ $(i = 1, \ldots, c)$. In this case, we can formulate the modified DSLDA method as the following optimization problem:

$$\begin{cases} \mathbf{W}_P = \mathbf{U}\mathbf{U}^T\mathbf{B}\mathbf{L}_P \\ \mathbf{L}_P = \arg\max \mathrm{trace}((\mathbf{L}_P^T\mathbf{S}_w\mathbf{L}_P)^{-1}\mathbf{L}_P^T\mathbf{S}_b\mathbf{L}_P) \end{cases} \quad (9)$$

$$\begin{cases} \mathbf{W}_C = \mathbf{U}^\perp{\mathbf{U}^\perp}^T\mathbf{B}\mathbf{L}_C \\ \mathbf{L}_C = \arg\max \mathrm{trace}(\mathbf{L}_C^T\mathbf{S}_C\mathbf{L}_C) \end{cases} \quad (10)$$

where $\mathbf{B} = [\mathbf{m}_1, \ldots, \mathbf{m}_c]$, $\mathbf{S}_w = \mathbf{B}^T\mathbf{U}\mathbf{U}^T\mathbf{S}_W\mathbf{U}\mathbf{U}^T\mathbf{B}$, $\mathbf{S}_b = \mathbf{B}^T\mathbf{U}\mathbf{U}^T\mathbf{S}_B\mathbf{U}\mathbf{U}^T\mathbf{B}$, and $S_C = \mathbf{B}^T\mathbf{U}^\perp{\mathbf{U}^\perp}^T\mathbf{S}_B\mathbf{U}^\perp{\mathbf{U}^\perp}^T\mathbf{B}$.

In what follows, we shall propose a DOI incremental algorithm for the DSLDA method, which can accurately update the discriminant vectors of DSLDA after new samples are inserted into the training data set. To begin with, we introduce Theorem 1 below, which is useful for deriving our incremental algorithm. The proof is given in Appendix A.

Theorem 1: Suppose that $\mathbf{u}_1, \ldots, \mathbf{u}_k$ $(k \le n - c)$ form an orthonormal basis of the subspace spanned by $\mathbf{x}_i^j - \mathbf{x}_i^1$ $(i = 1, \ldots, c; j = 2, \ldots, n_i)$, then we have

$$\overline{\mathbf{S}_W(0)} = \mathrm{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}.$$

Theorem 1 provides a method to find an orthonormal basis of the subspace $\overline{\mathbf{S}_W(0)}$. In the rest of this section, we will propose three algorithms for the solution of our modified DSLDA method. The first one solves the discriminant vectors of DSLDA in a batch form, and the other two focus on designing DOI incremental algorithms to accurately update the discriminant vectors when new samples are inserted into the training data set.

B. Batch Algorithm for DSLDA

Define a $d \times (n-c)$ matrix $\mathbf{A} = [\mathbf{A}_1,\ \mathbf{A}_2,\ \ldots,\ \mathbf{A}_c]$, where $\mathbf{A}_i = [\mathbf{x}_i^2 - \mathbf{x}_i^1,\ \ldots,\ \mathbf{x}_i^{n_i} - \mathbf{x}_i^1]$. Suppose that $\mathbf{u}_1, \ldots, \mathbf{u}_k$ $(k \le n - c)$ are the corresponding orthonormal vectors of the columns of $\mathbf{A}$ using the Gram–Schmidt orthogonalization procedures. Let $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_k]$; then from theorem 1 we know that $\overline{\mathbf{S}_W(0)}$ can be spanned by the columns of $\mathbf{U}$.

1) Solving Transformation Matrix $\mathbf{W}_P$: According to the definition of $\mathbf{W}_P$ in (9), solving the transformation matrix $\mathbf{W}_P$ boils down to solving the matrix $\mathbf{L}_P$ such that $\mathbf{W}_P = \mathbf{U}\mathbf{U}^T\mathbf{B}\mathbf{L}_P$, where $\mathbf{L}_P$ is the solution of the following optimization problem:

$$\mathbf{L}_P = \arg\max \mathrm{trace}((\mathbf{L}_P^T\mathbf{S}_w\mathbf{L}_P)^{-1}\mathbf{L}_P^T\mathbf{S}_b\mathbf{L}_P) \quad (11)$$

where $\mathbf{S}_w = \mathbf{B}^T\mathbf{U}\mathbf{U}^T\mathbf{S}_W\mathbf{U}\mathbf{U}^T\mathbf{B}$ and $\mathbf{S}_b = \mathbf{B}^T\mathbf{U}\mathbf{U}^T\mathbf{S}_B\mathbf{U}\mathbf{U}^T\mathbf{B}$. Let

$$\mathbf{P} = \mathbf{U}\mathbf{U}^T\mathbf{B}. \quad (12)$$

Then we have

$$\mathbf{S}_b = \mathbf{P}^T\mathbf{H}_B\mathbf{H}_\mathbf{B}^T\mathbf{P} = \mathbf{P}^T\mathbf{S}_B\mathbf{P}$$

and

$$\mathbf{S}_w = \mathbf{P}^T\mathbf{H}_W\mathbf{H}_W^T\mathbf{P} = \mathbf{P}^T\mathbf{S}_W\mathbf{P}. \quad (13)$$

The columns of $\mathbf{L}_P$ are the eigenvectors associated with the largest eigenvalues of the eigensystem

$$\mathbf{S}_w^{-1}\mathbf{S}_b\phi = \lambda\phi. \quad (14)$$

Suppose that $\phi_1, \ldots, \phi_{c-1}$ are the eigenvectors corresponding to the $c-1$ largest eigenvalues of eigensystem (14), and let

$$\omega_i^P = \mathbf{U}\mathbf{U}^T\mathbf{B}\phi_i = \mathbf{P}\phi_i \quad \text{and} \quad \omega_i^P = \frac{\omega_i^P}{\|\omega_i^P\|}. \quad (15)$$

Then, we obtain that $\mathbf{W}_P = [\omega_1^P, \ldots, \omega_{c-1}^P]$.

2) Solving Transformation Matrix $\mathbf{W}_C$: Suppose that $(\mathbf{U}, \mathbf{U}^\perp)$ is an orthogonal matrix whose columns form an orthonormal basis of the d-dimensional data space. Then we have

$$(\mathbf{U}, \mathbf{U}^\perp)(\mathbf{U}, \mathbf{U}^\perp)^T = \mathbf{U}\mathbf{U}^T + \mathbf{U}^\perp{\mathbf{U}^\perp}^T = \mathbf{I}. \quad (16)$$

Since columns of $\mathbf{U}$ span the subspace $\overline{\mathbf{S}_W(0)}$, we obtain that the columns of $\mathbf{U}^\perp$ span the subspace $\mathbf{S}_W(0)$. Hence, from the definition of $\mathbf{W}_C$ in (10), we obtain that solving the transformation matrix $\mathbf{W}_C$ boils down to solving the matrix $\mathbf{L}_C$ such that $\mathbf{W}_C = \mathbf{U}^\perp{\mathbf{U}^\perp}^T\mathbf{B}\mathbf{L}_C$, where $\mathbf{L}_C$ is the solution of the following optimization problem:

$$\begin{aligned} \mathbf{L}_C &= \arg\max \mathbf{L}_C^T\mathbf{B}^T\mathbf{U}^\perp{\mathbf{U}^\perp}^T\mathbf{H}_B\mathbf{H}_\mathbf{B}^T\mathbf{U}^\perp{\mathbf{U}^\perp}^T\mathbf{B}\mathbf{L}_C \\ &= \arg\max \mathbf{L}_C^T\mathbf{B}^T(\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{H}_B\mathbf{H}_\mathbf{B}^T(\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{B}\mathbf{L}_C \\ &= \arg\max \mathbf{L}_C^T(\mathbf{B}^T - \mathbf{P}^T)\mathbf{H}_B\mathbf{H}_\mathbf{B}^T(\mathbf{B} - \mathbf{P})\mathbf{L}_C. \end{aligned} \quad (17)$$

Let

$$\mathbf{S}_C = (\mathbf{B}^T - \mathbf{P}^T)\mathbf{H}_B\mathbf{H}_B^T(\mathbf{B} - \mathbf{P}) \quad (18)$$

and suppose that $\gamma_1, \ldots, \gamma_{c-1}$ are the eigenvectors corresponding to the $c-1$ largest eigenvalues of $\mathbf{S}_C$. Let

$$\omega_i^C = \mathbf{U}^\perp{\mathbf{U}^\perp}^T\mathbf{B}\gamma_i. \quad (19)$$

Combining (16) and (19), we have

$$\omega_i^C = \mathbf{U}^\perp{\mathbf{U}^\perp}^T\mathbf{B}\gamma_i = (\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{B}\gamma_i = (\mathbf{B} - \mathbf{P})\gamma_i. \quad (20)$$

Let $\omega_i^C = \omega_i^C/\|\omega_i^C\|$. Then we have $\mathbf{W}_C = [\omega_1^C, \ldots, \omega_{c-1}^C]$.

The pseudocode of the batch algorithm for the modified DSLDA method is listed in Algorithm 1.

Algorithm 1: Batch Algorithm for Our DSLDA Method

Input: Data matrix $\mathbf{X}$ and class labels $\mathbf{l}$.
Begin:
1) Calculate the class mean $\mathbf{m}_i$, the global mean $\mathbf{m}$, the number n_i of the ith class samples, and the total number n of the samples.
2) Set $\mathbf{B} = [\mathbf{m}_1, \ldots, \mathbf{m}_c]$, $\mathbf{H}_B = [\sqrt{n_1}(\mathbf{m}_1 - \mathbf{m}), \ldots, \sqrt{n_c}(\mathbf{m}_c - \mathbf{m})]$, and

$$\mathbf{H}_W = [\mathbf{x}_1^1 - \mathbf{m}_1, \ldots, \mathbf{x}_1^{n_1} - \mathbf{m}_1, \ldots, \ldots, \mathbf{x}_c^1 - \mathbf{m}_c, \ldots, \mathbf{x}_c^{n_c} - \mathbf{m}_c]$$

$\mathbf{A} = [\mathbf{A}_1, \ \mathbf{A}_2, \ \ldots, \ \mathbf{A}_c]$, where $\mathbf{A}_i = [\mathbf{x}_i^2 - \mathbf{x}_i^1, \ \ldots, \ \mathbf{x}_i^{n_i} - \mathbf{x}_i^1]$.
3) Get the orthogonal matrix $\mathbf{U}$ by orthonormalizing the columns of $\mathbf{A}$ using Gram–Schmidt orthogonalization procedures.
4) Compute $\mathbf{P} = \mathbf{U}\mathbf{U}^T\mathbf{B}$.
5) Compute $\mathbf{S}_b = \mathbf{P}^T\mathbf{H}_B\mathbf{H}_{\mathbf{B}}^T\mathbf{P}$, and $\mathbf{S}_w = \mathbf{P}^T\mathbf{H}_W\mathbf{H}_W^T\mathbf{P}$.
6) Solve the $c-1$ principal eigenvectors $\phi_1, \ldots, \phi_{c-1}$ of $\mathbf{S}_w^{-1}\mathbf{S}_b$.
7) Compute $\omega_i^P = \mathbf{P}\phi_i$, $\omega_i^P \leftarrow \omega_i^P/\|\omega_i^P\|$, $\mathbf{W}_P \leftarrow [\omega_1^P, \ldots, \omega_{c-1}^P]$.
8) Compute $\mathbf{S}_C = (\mathbf{B}^T - \mathbf{P}^T)\mathbf{H}_B\mathbf{H}_{\mathbf{B}}^T(\mathbf{B} - \mathbf{P})$.
9) Solve the $c-1$ principal eigenvectors $\gamma_1, \ldots, \gamma_{c-1}$ of $\mathbf{S}_C$.
10) Compute $\omega_i^C = (\mathbf{B} - \mathbf{P})\gamma_i$, $\omega_i^C \leftarrow \omega_i^C/\|\omega_i^C\|$, $\mathbf{W}_C \leftarrow [\omega_1^C, \ldots, \omega_{c-1}^C]$.

Output: The updated matrices $\mathbf{B}$, $\mathbf{H}_B$, $\mathbf{H}_W$, $\mathbf{U}$, $\mathbf{S}_b$, $\mathbf{S}_w$, $\mathbf{P}$, the size n_j of the jth class samples, the global mean $\mathbf{m}$ of the training samples, and the transform matrices $\mathbf{W}_P$ and $\mathbf{W}_C$.

C. *Fast Algorithm for Updating DSLDA*

We have proposed the batch algorithm for a modified DSLDA method in Section III-B. In this section, we will focus on designing the DOI incremental algorithm to update the discriminant vectors of DSLDA when new samples are inserted into the training data set. For simplicity of deriving our algorithm, we use the following convention: for any variable $\mathbf{X}$, its updated version after inserting new samples is denoted by $\tilde{\mathbf{X}}$. Let $\mathbf{x}$ be an inserted instance coming from the ith class. Without loss of generality, we assume that $i > 1$. Divide $\tilde{\mathbf{S}}_B$ and $\tilde{\mathbf{S}}_W$ as the following two parts:

$$\tilde{\mathbf{S}}_B = \mathbf{S}_B + \Delta\mathbf{S}_B \quad (21)$$
$$\tilde{\mathbf{S}}_W = \mathbf{S}_W + \Delta\mathbf{S}_W. \quad (22)$$

Then, from (12), we obtain that

$$\tilde{\mathbf{S}}_b = \tilde{\mathbf{P}}^T\mathbf{S}_B\tilde{\mathbf{P}} + \tilde{\mathbf{P}}^T\Delta\mathbf{S}_B\tilde{\mathbf{P}} \quad (23)$$
$$\tilde{\mathbf{S}}_w = \tilde{\mathbf{P}}^T\mathbf{S}_W\tilde{\mathbf{P}} + \tilde{\mathbf{P}}^T\Delta\mathbf{S}_W\tilde{\mathbf{P}}. \quad (24)$$

In what follows, we update the scatter matrices according to the different cases of the inserted instance $\mathbf{x}$: 1) $\mathbf{x}$ belongs to an existing class i ($\leq c$); 2) $\mathbf{x}$ belongs to a new class i ($> c$).

1) The Inserted Instance $\mathbf{x}$ *Belongs to an Existing Class* $i(\leq c)$*:* If $\mathbf{x}$ belongs to an existing class i, then the updated matrices $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{A}}$ can be expressed as

$$\tilde{\mathbf{B}} = \mathbf{B} + (\tilde{\mathbf{m}}_i - \mathbf{m}_i)\mathbf{e}_i^T \quad \text{and} \quad \tilde{\mathbf{A}} = [\mathbf{A}, \ \mathbf{x} - \mathbf{x}_i^1] \quad (25)$$

where $\mathbf{e}_i$ represents a $c \times 1$ unit vector with the ith item equals to 1. Then we obtain that the updated orthonormal matrix $\tilde{\mathbf{U}}$ can be written as

$$\tilde{\mathbf{U}} = [\mathbf{U}, \mathbf{u}] \quad (26)$$

where $\mathbf{u} = ((\mathbf{x} - \mathbf{x}_i^1) - \mathbf{U}\mathbf{U}^T(\mathbf{x} - \mathbf{x}_i^1))/(\|(\mathbf{x} - \mathbf{x}_i^1) - \mathbf{U}\mathbf{U}^T(\mathbf{x} - \mathbf{x}_i^1)\|)$. From (25) and (26), we obtain that

$$\tilde{\mathbf{P}} = \tilde{\mathbf{U}}\tilde{\mathbf{U}}^T\tilde{\mathbf{B}} = \mathbf{P} + \mathbf{u}\mathbf{u}^T\mathbf{B} + \mathbf{U}\mathbf{U}^T(\tilde{\mathbf{m}}_i - \mathbf{m}_i)\mathbf{e}_i^T + \mathbf{u}\mathbf{u}^T(\tilde{\mathbf{m}}_i - \mathbf{m}_i)\mathbf{e}_i^T. \quad (27)$$

Combining (2), (3), (12), and (27), we have

$$\tilde{\mathbf{P}}^T\mathbf{S}_B\tilde{\mathbf{P}} = \mathbf{S}_b + \Delta\mathbf{S}_b^1 \quad (28)$$
$$\tilde{\mathbf{P}}^T\mathbf{S}_W\tilde{\mathbf{P}} = \mathbf{S}_w + \Delta\mathbf{S}_w^1 \quad (29)$$

where

$$\begin{aligned}\Delta\mathbf{S}_b^1 = &\{\mathbf{B}^T\mathbf{u}\}\{\mathbf{u}^T\mathbf{H}_B\mathbf{H}_B^T\tilde{\mathbf{P}}\} \\ &+ \mathbf{e}_i\{(\tilde{\mathbf{m}}_i - \mathbf{m}_i)^T\tilde{\mathbf{U}}\tilde{\mathbf{U}}^T\mathbf{H}_B\mathbf{H}_B^T\tilde{\mathbf{P}}\} \\ &+ \{\mathbf{P}^T\mathbf{H}_B\mathbf{H}_B^T\mathbf{u}\}\{\mathbf{u}^T\mathbf{B}\} \\ &+ \{\mathbf{P}^T\mathbf{H}_B\mathbf{H}_B^T\tilde{\mathbf{U}}\tilde{\mathbf{U}}^T(\tilde{\mathbf{m}}_i - \mathbf{m}_i)\}\mathbf{e}_i^T \quad (30) \\ \Delta\mathbf{S}_w^1 = &\{\mathbf{B}^T\mathbf{u}\}\{\mathbf{u}^T\mathbf{H}_W\mathbf{H}_W^T\tilde{\mathbf{P}}\} \\ &+ \mathbf{e}_i\{(\tilde{\mathbf{m}}_i - \mathbf{m}_i)^T\tilde{\mathbf{U}}\tilde{\mathbf{U}}^T\mathbf{H}_W\mathbf{H}_W^T\tilde{\mathbf{P}}\} \\ &+ \{\mathbf{P}^T\mathbf{H}_W\mathbf{H}_W^T\mathbf{u}\}\{\mathbf{u}^T\mathbf{B}\} \\ &+ \{\mathbf{P}^T\mathbf{H}_W\mathbf{H}_W^T\tilde{\mathbf{U}}\tilde{\mathbf{U}}^T(\tilde{\mathbf{m}}_i - \mathbf{m}_i)\}\mathbf{e}_i^T. \quad (31)\end{aligned}$$

Moreover, we have the following expressions with respect to the matrices $\Delta\mathbf{S}_B$ and $\Delta\mathbf{S}_W$

$$\begin{aligned}\Delta\mathbf{S}_B = &n\mathbf{m}\mathbf{m}^T - n_i\mathbf{m}_i\mathbf{m}_i^T + (n_i + 1)\tilde{\mathbf{m}}_i\tilde{\mathbf{m}}_i^T \\ &- (n+1)\tilde{\mathbf{m}}\tilde{\mathbf{m}}^T \quad (32) \\ \Delta\mathbf{S}_W = &\mathbf{x}\mathbf{x}^T + n_i\mathbf{m}_i\mathbf{m}_i^T - (n_i + 1)\tilde{\mathbf{m}}_i\tilde{\mathbf{m}}_i^T. \quad (33)\end{aligned}$$

The detailed derivations are given in Appendix B.

Let $\Delta\mathbf{S}_b^2 = \tilde{\mathbf{P}}^T\Delta\mathbf{S}_B\tilde{\mathbf{P}}$ and $\Delta\mathbf{S}_w^2 = \tilde{\mathbf{P}}^T\Delta\mathbf{S}_W\tilde{\mathbf{P}}$. Then from (32) and (33), we obtain

$$\begin{aligned}\Delta\mathbf{S}_b^2 = &n\{\tilde{\mathbf{P}}^T\mathbf{m}\}\{\mathbf{m}^T\tilde{\mathbf{P}}\} - n_i\{\tilde{\mathbf{P}}^T\mathbf{m}_i\}\{\mathbf{m}_i^T\tilde{\mathbf{P}}\} \\ &+ (n_i + 1)\{\tilde{\mathbf{P}}^T\tilde{\mathbf{m}}_i\}\{\tilde{\mathbf{m}}_i^T\tilde{\mathbf{P}}\} \\ &- (n+1)\{\tilde{\mathbf{P}}^T\tilde{\mathbf{m}}\}\{\tilde{\mathbf{m}}^T\tilde{\mathbf{P}}\} \quad (34) \\ \Delta\mathbf{S}_w^2 = &\{\tilde{\mathbf{P}}^T\mathbf{x}\}\{\mathbf{x}^T\tilde{\mathbf{P}}\} + n_i\{\tilde{\mathbf{P}}^T\mathbf{m}_i\}\{\mathbf{m}_i^T\tilde{\mathbf{P}}\} \\ &- (n_i + 1)\{\tilde{\mathbf{P}}^T\tilde{\mathbf{m}}_i\}\{\tilde{\mathbf{m}}_i^T\tilde{\mathbf{P}}\}. \quad (35)\end{aligned}$$

After calculating the matrices $\Delta\mathbf{S}_b^1$, $\Delta\mathbf{S}_w^1$, $\Delta\mathbf{S}_b^2$, and $\Delta\mathbf{S}_w^2$, we can update the matrices $\mathbf{S}_b$ and $\mathbf{S}_w$ as follows:

$$\tilde{\mathbf{S}}_b = \mathbf{S}_b + \Delta\mathbf{S}_b^1 + \Delta\mathbf{S}_b^2 \quad \text{and} \quad \tilde{\mathbf{S}}_w = \mathbf{S}_w + \Delta\mathbf{S}_w^1 + \Delta\mathbf{S}_w^2.$$

Moreover, the matrix $\mathbf{S}_C$ can be updated as

$$\tilde{\mathbf{S}}_C = (\tilde{\mathbf{B}}^T - \tilde{\mathbf{P}}^T)\tilde{\mathbf{H}}_B\tilde{\mathbf{H}}_{\mathbf{B}}^T(\tilde{\mathbf{B}} - \tilde{\mathbf{P}}). \tag{36}$$

Finally, the columns of the transformation matrix $\tilde{\mathbf{W}}_P$ can be obtained by solving the eigenvectors of $\tilde{\mathbf{S}}_w^{-1}\tilde{\mathbf{S}}_b$ whereas the columns of $\tilde{\mathbf{W}}_C$ can be obtained by solving the eigenvectors of $\tilde{\mathbf{S}}_C$.

The pseudocode of the fast algorithm for updating existing class is listed in Algorithm 2.

Algorithm 2: DOI Incremental Algorithm for Updating Existing Class

---

Input: The matrices $\mathbf{B}, \mathbf{H}_B, \mathbf{H}_W, \mathbf{U}, \mathbf{S}_b, \mathbf{S}_w, \mathbf{P}$, the size n_j of the jth class samples, the global mean $\mathbf{m}$ of the training samples, and the inserted sample $\mathbf{x}$ of ith class.

Begin:

1) Set $\tilde{n}_j = n_j$, $\tilde{\mathbf{m}}_j = \mathbf{m}_j$ $(j \neq i)$ and $\tilde{n}_i = n_i + 1$, $\tilde{\mathbf{m}}_i = (n_i\mathbf{m}_i + \mathbf{x})/(n_i + 1)$, $n = \sum_j n_j$, and $\tilde{\mathbf{m}} = (n\mathbf{m} + \mathbf{x})/(n + 1)$.
2) $\tilde{\mathbf{B}} \leftarrow \mathbf{B} + (\tilde{\mathbf{m}}_i - \mathbf{m}_i)\mathbf{e}_i^T$,
$\tilde{\mathbf{H}}_B = [\sqrt{\tilde{n}_1}(\tilde{\mathbf{m}}_1 - \tilde{\mathbf{m}}), \ldots, \sqrt{\tilde{n}_c}(\tilde{\mathbf{m}}_c - \tilde{\mathbf{m}})]$

$$\tilde{\mathbf{H}}_W = [\mathbf{x}_1^1 - \mathbf{m}_1, \ldots, \mathbf{x}_1^{n_1} - \mathbf{m}_1, \ldots, \mathbf{x}_i^1 - \tilde{\mathbf{m}}_i, \ldots, \mathbf{x}_i^{n_i} - \tilde{\mathbf{m}}_i, \ldots, \mathbf{x}_c^1 - \mathbf{m}_c, \ldots, \mathbf{x}_c^{n_c} - \mathbf{m}_c, \mathbf{x} - \tilde{\mathbf{m}}_i].$$

3) $\mathbf{u} \leftarrow ((\mathbf{x} - \mathbf{x}_i^1) - \mathbf{U}\mathbf{U}^T(\mathbf{x} - \mathbf{x}_i^1))/\|(\mathbf{x} - \mathbf{x}_i^1) - \mathbf{U}\mathbf{U}^T(\mathbf{x} - \mathbf{x}_i^1)\|$, $\tilde{\mathbf{U}} \leftarrow [\mathbf{U}, \mathbf{u}]$.
4) $\tilde{\mathbf{P}} \leftarrow \mathbf{P} + \mathbf{u}\mathbf{u}^T\mathbf{B} + \mathbf{U}\mathbf{U}^T(\tilde{\mathbf{m}}_i - \mathbf{m}_i)\mathbf{e}_i^T + \mathbf{u}\mathbf{u}^T(\tilde{\mathbf{m}}_i - \mathbf{m}_i)\mathbf{e}_i^T$.
5) Compute $\Delta\mathbf{S}_b^1$, $\Delta\mathbf{S}_w^1$, $\Delta\mathbf{S}_b^2$, and $\Delta\mathbf{S}_w^2$ using (30), (31), (34), (35), and then update $\mathbf{S}_b$ and $\mathbf{S}_w$: $\tilde{\mathbf{S}}_b \leftarrow \mathbf{S}_b + \Delta\mathbf{S}_b^1 + \Delta\mathbf{S}_b^2$, $\tilde{\mathbf{S}}_w \leftarrow \mathbf{S}_w + \Delta\mathbf{S}_w^1 + \Delta\mathbf{S}_w^2$.
6) Solve the $c-1$ principal eigenvectors $\phi_1, \ldots, \phi_{c-1}$ of $\tilde{\mathbf{S}}_w^{-1}\tilde{\mathbf{S}}_b$.
7) Compute $\omega_i^P = \tilde{\mathbf{P}}\phi_i$, $\omega_i^P \leftarrow \omega_i^P/\|\omega_i^P\|$, $\tilde{\mathbf{W}}_P \leftarrow [\omega_1^P, \ldots, \omega_{c-1}^P]$.
8) Compute $\tilde{\mathbf{S}}_C = (\tilde{\mathbf{B}}^T - \tilde{\mathbf{P}}^T)\tilde{\mathbf{H}}_B\tilde{\mathbf{H}}_{\mathbf{B}}^T(\tilde{\mathbf{B}} - \tilde{\mathbf{P}})$.
9) Solve the $c-1$ principal eigenvectors $\gamma_1, \ldots, \gamma_{c-1}$ of $\tilde{\mathbf{S}}_C$.
10) Compute $\omega_i^C = (\tilde{\mathbf{B}} - \tilde{\mathbf{P}})\gamma_i$, $\omega_i^C \leftarrow \omega_i^C/\|\omega_i^C\|$, $\tilde{\mathbf{W}}_C \leftarrow [\omega_1^C, \ldots, \omega_{c-1}^C]$.

Output: The updated matrices $\tilde{\mathbf{B}}, \tilde{\mathbf{H}}_B, \tilde{\mathbf{H}}_W, \tilde{\mathbf{U}}, \tilde{\mathbf{S}}_b, \tilde{\mathbf{S}}_w, \tilde{\mathbf{P}}$, the size $\tilde{n}_j$ of the jth class samples, the global mean $\tilde{\mathbf{m}}$ of the training samples, and the transform matrices $\tilde{\mathbf{W}}_P$ and $\tilde{\mathbf{W}}_C$.

2) The Inserted Instance $\mathbf{x}$ *Belongs to a New Class* $i(> c)$*:* If the inserted instance $\mathbf{x}$ belongs to a new class, then the updated matrices $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{A}}$ can be expressed as

$$\tilde{\mathbf{B}} = [\mathbf{B}, \mathbf{x}] \quad \text{and} \quad \tilde{\mathbf{A}} = \mathbf{A}. \tag{37}$$

Hence, we obtain that the updated matrix $\tilde{\mathbf{U}}$ equal to $\mathbf{U}$

$$\tilde{\mathbf{U}} = \mathbf{U}. \tag{38}$$

From (37) and (38), we have

$$\tilde{\mathbf{P}} = \tilde{\mathbf{U}}\tilde{\mathbf{U}}^T\tilde{\mathbf{B}} = [\mathbf{U}\mathbf{U}^T\mathbf{B}, \mathbf{U}\mathbf{U}^T\mathbf{x}] = [\mathbf{P}, \mathbf{U}\mathbf{U}^T\mathbf{x}]. \tag{39}$$

Combining (2), (3), (12), and (39), we obtain

$$\tilde{\mathbf{P}}^T\mathbf{S}_B\tilde{\mathbf{P}} = \begin{pmatrix} \mathbf{S}_b & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix} + \Delta\mathbf{S}_b^1 \tag{40}$$

$$\tilde{\mathbf{P}}^T\mathbf{S}_W\tilde{\mathbf{P}} = \begin{pmatrix} \mathbf{S}_w & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix} + \Delta\mathbf{S}_w^1 \tag{41}$$

where $\mathbf{0}$ denotes the $c \times 1$ zero vector, $\Delta\mathbf{S}_b^1$ and $\Delta\mathbf{S}_w^1$ are respectively given by

$$\Delta\mathbf{S}_b^1 = \begin{pmatrix} \mathbf{O}_{c\times c} & \mathbf{P}^T\mathbf{H}_B\mathbf{H}_{\mathbf{B}}^T\mathbf{U}\mathbf{U}^T\mathbf{x} \\ \mathbf{x}^T\mathbf{U}\mathbf{U}^T\mathbf{H}_B\mathbf{H}_{\mathbf{B}}^T\mathbf{P} & \mathbf{x}^T\mathbf{U}\mathbf{U}^T\mathbf{H}_B\mathbf{H}_{\mathbf{B}}^T\mathbf{U}\mathbf{U}^T\mathbf{x} \end{pmatrix} \tag{42}$$

$$\Delta\mathbf{S}_w^1 = \begin{pmatrix} \mathbf{O}_{c\times c} & \mathbf{P}^T\mathbf{H}_W\mathbf{H}_W^T\mathbf{U}\mathbf{U}^T\mathbf{x} \\ \mathbf{x}^T\mathbf{U}\mathbf{U}^T\mathbf{H}_W\mathbf{H}_W^T\mathbf{P} & \mathbf{x}^T\mathbf{U}\mathbf{U}^T\mathbf{H}_W\mathbf{H}_W^T\mathbf{U}\mathbf{U}^T\mathbf{x} \end{pmatrix} \tag{43}$$

where $\mathbf{O}_{c\times c}$ denotes the $c \times c$ zero matrix.

Moreover, we have the following expressions with respect to the matrices $\Delta\mathbf{S}_B$ and $\Delta\mathbf{S}_W$

$$\Delta\mathbf{S}_B = n\mathbf{m}\mathbf{m}^T + \mathbf{x}\mathbf{x}^T - (n+1)\tilde{\mathbf{m}}\tilde{\mathbf{m}}^T \tag{44}$$

$$\Delta\mathbf{S}_W = \mathbf{O}_{d\times d} \tag{45}$$

where $\mathbf{O}_{d\times d}$ denotes the $d \times d$ zero matrix. The derivations are given in Appendix C.

Let $\Delta\mathbf{S}_b^2 = \tilde{\mathbf{P}}^T\Delta\mathbf{S}_B\tilde{\mathbf{P}}$ and $\Delta\mathbf{S}_w^2 = \tilde{\mathbf{P}}^T\Delta\mathbf{S}_W\tilde{\mathbf{P}}$. Then from (44) and (45), we have

$$\Delta\mathbf{S}_b^2 = n\{\tilde{\mathbf{P}}^T\mathbf{m}\}\{\mathbf{m}^T\tilde{\mathbf{P}}\} + \{\tilde{\mathbf{P}}^T\mathbf{x}\}\{\mathbf{x}^T\tilde{\mathbf{P}}\} - (n+1)\{\tilde{\mathbf{P}}^T\tilde{\mathbf{m}}\}\{\tilde{\mathbf{m}}^T\tilde{\mathbf{P}}\} \tag{46}$$

$$\Delta\mathbf{S}_w^2 = \mathbf{O}_{(c+1)\times(c+1)}. \tag{47}$$

The matrices $\mathbf{S}_b$ and $\mathbf{S}_w$ can be updated according to the following expressions:

$$\tilde{\mathbf{S}}_b = \begin{pmatrix} \mathbf{S}_b & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix} + \Delta\mathbf{S}_b^1 + \Delta\mathbf{S}_b^2 \tag{48}$$

$$\tilde{\mathbf{S}}_w = \begin{pmatrix} \mathbf{S}_w & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix} + \Delta\mathbf{S}_w^1 \tag{49}$$

and the updated matrix $\tilde{\mathbf{S}}_C$ can be calculated as

$$\tilde{\mathbf{S}}_C = (\tilde{\mathbf{B}}^T - \tilde{\mathbf{P}}^T)\tilde{\mathbf{H}}_B\tilde{\mathbf{H}}_B^T(\tilde{\mathbf{B}} - \tilde{\mathbf{P}}). \tag{50}$$

The columns of the transformation matrix $\tilde{\mathbf{W}}_P$ can be obtained by solving the eigenvectors of $\tilde{\mathbf{S}}_w^{-1}\tilde{\mathbf{S}}_b$ whereas the columns of $\tilde{\mathbf{W}}_C$ can be obtained by solving the eigenvectors of $\tilde{\mathbf{S}}_C$.

TABLE I
COMPUTATIONAL COMPLEXITY OF EACH LINE IN ALGORITHM 1

No. of Line	1)	2)	3)	4)	5)	6)	7)	8)	9)	10)
Algorithm 1	$O(dc)$	$O(dc)$	$O(d(n-c)^2)$	$O(d(n-c)c)$	$O(dnc)$	$O(c^3)$	$O(dc^2)$	$O(dc^2)$	$O(c^3)$	$O(dc^2)$

TABLE II
COMPUTATIONAL COMPLEXITY OF EACH LINE IN ALGORITHMS 2 AND 3

No. of Line	1)	2)	3)	4)	5)	6)	7)	8)	9)	10)
Algorithm 2	$O(d)$	$O(dc)$	$O(d(n-c))$	$O(d(n-c))$ $+O(dc)$	$O(d(n-c))$ $+O(dc)$	$O(c^3)$	$O(dc^2)$	$O(dc^2)$	$O(c^3)$	$O(dc^2)$
Algorithm 3	$O(d)$	$O(dc)$	0	$O(d(n-c))$	$O(d(n-c))$ $+O(dc)$	$O(c^3)$	$O(dc^2)$	$O(dc^2)$	$O(c^3)$	$O(dc^2)$

The pseudocode of the fast algorithm for updating new class is listed in Algorithm 3.

Algorithm 3: DOI Incremental Algorithm for Updating New Class

Input: The matrices $\mathbf{B}, \mathbf{H}_B, \mathbf{H}_W, \mathbf{U}, \mathbf{S}_b, \mathbf{S}_w, \mathbf{P}$, the size n_j of the jth class samples, the global mean $\mathbf{m}$ of the training samples, and the inserted sample $\mathbf{x}$ of ith class.

Begin:

1) Set $\tilde{n}_j = n_j$, $\tilde{\mathbf{m}}_j = \mathbf{m}_j$ $(j = 1, \ldots, c)$ and $\tilde{n}_{c+1} = 1$, $\tilde{\mathbf{m}}_{c+1} = \mathbf{x}$, $n = \sum_j n_j$, and $\tilde{\mathbf{m}} = (n\mathbf{m} + \mathbf{x})/(n+1)$.
2) $\tilde{\mathbf{B}} \leftarrow [\mathbf{B}, \mathbf{x}]$, $\tilde{\mathbf{H}}_B = [\sqrt{\tilde{n}_1}(\tilde{\mathbf{m}}_1 - \tilde{\mathbf{m}}), \ldots, \sqrt{\tilde{n}_c}(\tilde{\mathbf{m}}_c - \tilde{\mathbf{m}}), \mathbf{x} - \tilde{\mathbf{m}}]$, $\tilde{\mathbf{H}}_W = \mathbf{H}_W$.
3) $\tilde{\mathbf{U}} \leftarrow \mathbf{U}$.
4) $\tilde{\mathbf{P}} \leftarrow [\mathbf{P}, \mathbf{U}\mathbf{U}^T\mathbf{x}]$.
5) Compute $\Delta\mathbf{S}_b^1$, $\Delta\mathbf{S}_w^1$, and $\Delta\mathbf{S}_b^2$ using (42), (43), (46), and then update $\mathbf{S}_b$ and $\mathbf{S}_w$: $\tilde{\mathbf{S}}_b \leftarrow \begin{pmatrix} \mathbf{S}_b & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix} + \Delta\mathbf{S}_b^1 + \Delta\mathbf{S}_b^2$, $\tilde{\mathbf{S}}_w \leftarrow \begin{pmatrix} \mathbf{S}_w & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix} + \Delta\mathbf{S}_w^1$.
6) Solve the c principal eigenvectors $\phi_1, \ldots, \phi_c$ of $\tilde{\mathbf{S}}_w^{-1}\tilde{\mathbf{S}}_b$.
7) Compute $\omega_i^P = \tilde{\mathbf{P}}\phi_i$, $\omega_i^P \leftarrow \omega_i^P/\|\omega_i^P\|$, $\tilde{\mathbf{W}}_P \leftarrow [\omega_1^P, \ldots, \omega_c^P]$.
8) $\tilde{\mathbf{S}}_C = (\tilde{\mathbf{B}}^T - \tilde{\mathbf{P}}^T)\tilde{\mathbf{H}}_B\tilde{\mathbf{H}}_B^T(\tilde{\mathbf{B}} - \tilde{\mathbf{P}})$.
9) Solve the c principal eigenvectors $\gamma_1, \ldots, \gamma_c$ of $\tilde{\mathbf{S}}_C$.
10) Compute $\omega_i^C = (\tilde{\mathbf{B}} - \tilde{\mathbf{P}})\gamma_i$, $\omega_i^C \leftarrow \omega_i^C/\|\omega_i^C\|$, $\tilde{\mathbf{W}}_C \leftarrow [\omega_1^C, \ldots, \omega_c^C]$.

Output: The updated matrices $\tilde{\mathbf{B}}, \tilde{\mathbf{H}}_B, \tilde{\mathbf{H}}_W, \tilde{\mathbf{U}}, \tilde{\mathbf{S}}_b, \tilde{\mathbf{S}}_w, \tilde{\mathbf{P}}$, the size $\tilde{n}_j$ of the jth class samples, the global mean $\tilde{\mathbf{m}}$ of the training samples, and the transform matrices $\tilde{\mathbf{W}}_P$ and $\tilde{\mathbf{W}}_C$.

D. Computational Analysis of the DSLDA Algorithm

In Algorithms 1, 2, and 3, we list the pseudocodes of solving the discriminant vectors of the modified DSLDA, where Algorithm 1 describes the batch approach of solving DSLDA, Algorithm 2 describes the DOI incremental algorithm for updating DSLDA when a new sample of the ith class is inserted into the training data set, and Algorithm 3 describes the DOI incremental algorithm for updating DSLDA when a new sample from a new class is inserted into the training data set. Table I lists the computational complexity of each line of the batch algorithm (Algorithm 1) whereas Table II lists the complexity of each line of the incremental algorithms (Algorithms 2 and 3).

From Table I, we can see that the computational complexity of relearning the discriminant vectors of DSLDA with $n+1$ training samples using the batch algorithm is $O(d(n-c)^2) + O(dnc) + O(dc^2)$. However, from Table II, we can see that updating the discriminant vectors of DSLDA using the incremental algorithms when a new training sample is inserted into the training data set with n elements only needs the computational complexity of $O(d(n-c)) + O(dc) + O(dc^2)$.

Based on Algorithms 2 and 3 as well as Table II, we can analyze the computational complexity of updating the discriminant vectors when more than one sample is inserted into the training data set. More specifically, if more than one sample is inserted into the training data set with n elements, the last five lines, i.e., lines 6–10 of both Algorithms 2 and 3 only need to be computed once. Suppose that there are k samples being inserted into the the training data set. For simplicity, we assume that there are Δc new classes of samples among the k inserted samples. In this case, the first five lines, i.e., 1–5, in Algorithm 2 should be repeated $k - \Delta c$ times while the first five lines in Algorithm 3 should be repeated Δc times with the insertion of the k samples. According to Table II, we can obtain that the complexity of updating the new discriminant vectors based on Algorithms 2 and 3 is $O(d(n+k-c-\Delta c)k) + O(d(c+\Delta c)k) + O(d(c+\Delta c)^2)$. By contrast, using the batch algorithm to recompute the discriminant vectors on the training data set with $n+k$ samples, we need the computational complexity of

$$O(d(n+k-c-\Delta c)^2) \\ +O(d(n+k)(c+\Delta c)) + O(d(c+\Delta c)^2).$$

IV. PATTERN CLASSIFICATION BASED ON DSLDA

In this section, we will address the classification problem using our DSLDA method. Suppose that $\mathbf{x}_{\text{test}}$ is a test sample. Then the projections of $\mathbf{x}_{\text{test}}$ onto $\mathbf{W}_P$ and $\mathbf{W}_C$ are $\mathbf{y}_{\text{test}} = \mathbf{W}_P^T\mathbf{x}_{\text{test}}$ and $\mathbf{z}_{\text{test}} = \mathbf{W}_C^T\mathbf{x}_{\text{test}}$. Let $\mathbf{y}_i^j = \mathbf{W}_P^T\mathbf{x}_i^j$ and $\mathbf{z}_i^j = \mathbf{W}_C^T\mathbf{x}_i^j$ $(i = 1, \ldots, c; j = 1, \ldots, n_i)$ be the projections of $\mathbf{x}_i^j$ onto $\mathbf{W}_P$ and $\mathbf{W}_C$, respectively. Denote the distance between $\mathbf{y}_{\text{test}}$ and $\mathbf{y}_i^j$ by $d_P(\mathbf{y}_{\text{test}}, \mathbf{y}_i^j)$ and the distance between $\mathbf{z}_{\text{test}}$ and $\mathbf{z}_i^j$ by $d_C(\mathbf{z}_{\text{test}}, \mathbf{z}_i^j)$, where

$$d_P(\mathbf{y}_{\text{test}}, \mathbf{y}_i^j) = \|\mathbf{y}_{\text{test}} - \mathbf{y}_i^j\| \quad (51)$$

$$d_C(\mathbf{z}_{\text{test}}, \mathbf{z}_i^j) = \|\mathbf{z}_{\text{test}} - \mathbf{z}_i^j\|. \quad (52)$$

It should be noted that $d_P(\mathbf{y}_{\text{test}}, \mathbf{y}_i^j)$ and $d_C(\mathbf{z}_{\text{test}}, \mathbf{z}_i^j)$ may not share the same metric measurement. To solve this problem, we adopt the scheme of Yang *et al.* [10] by defining the following hybrid distance $d(\mathbf{x}_{\text{test}}, \mathbf{x}_i^j)$ to measure the similarity between $\mathbf{x}_{\text{test}}$ and $\mathbf{x}_i^j$, where $d(\mathbf{x}_{\text{test}}, \mathbf{x}_i^j)$ is defined as

$$d(\mathbf{x}_{\text{test}}, \mathbf{x}_i^j) = (1-\mu)\frac{d_P(\mathbf{y}_{\text{test}}, \mathbf{y}_i^j)}{\sum_{i=1}^{c}\sum_{j=1}^{n_i} d_P(\mathbf{y}_{\text{test}}, \mathbf{y}_i^j)} + \mu\frac{d_C(\mathbf{z}_{\text{test}}, \mathbf{z}_i^j)}{\sum_{i=1}^{c}\sum_{j=1}^{n_i} d_C(\mathbf{z}_{\text{test}}, \mathbf{z}_i^j)} \quad (53)$$

where $\mu \in [0,1]$ is a weight parameter used to emphasize the contribution of one of the two "discriminant subspaces" and deemphasize the other one in the classification task.

Suppose that c^* is the class label of the test sample $\mathbf{x}_{\text{test}}$. Then c^* can be obtained by the following expression:

$$c^* = \arg\min_i d(\mathbf{x}_{\text{test}}, \mathbf{x}_i^j). \quad (54)$$

V. Experiments

In this section, we will conduct experiments on the AR face database [22] to test the performance of our DSLDA algorithms in terms of the computational efficiency and the recognition accuracy. For comparison, we also conduct the same experiments using the PCA+LDA method, the DLDA method, the null space method, the IDR/QR method, the two-dimensional LDA (2DLDA) [23], and the original DSLDA method proposed by Wang and Tang [9], respectively. The nearest neighbor classifier is used for the classification task in the experiments.

The AR face database consists of over 3000 facial images of 126 subjects. Each subject contains 26 facial images recorded in two different sessions separated by two weeks, where each session consists of 13 images. The original image size is 768×576 pixels, and each pixel is represented by 24 bits of RGB color values. Fig. 1 shows the 26 images of one subject, where the first 13 images are taken in the first session, and the remaining 13 images are taken in the second session. We randomly select 70 subjects (50 males and 20 females) among the 126 subjects for the experiment. For each subject, only the nonoccluded images, i.e., the images numbered 1–7 and 14–20 (see Fig. 1), are used for our experiments. To reduce the computational cost, all the images are centered and cropped into the size of 468×476 pixels, and then down-sampled into the size of 100×100 pixels.

A. Computational Efficiency Test

In this experiment, we will test the computational efficiency of the proposed DSLDA algorithms in terms of the CPU time of updating the discriminant vectors when new samples are inserted into the training data set. Among the 980 samples of the AR face database, we randomly select 900 samples as the training data set, and use the remaining 80 samples as the testing data set. We first use the batch DSLDA algorithm (i.e., Algorithm 1) to compute the discriminant vectors on the training data set. Then we design our experiments according to the following steps:

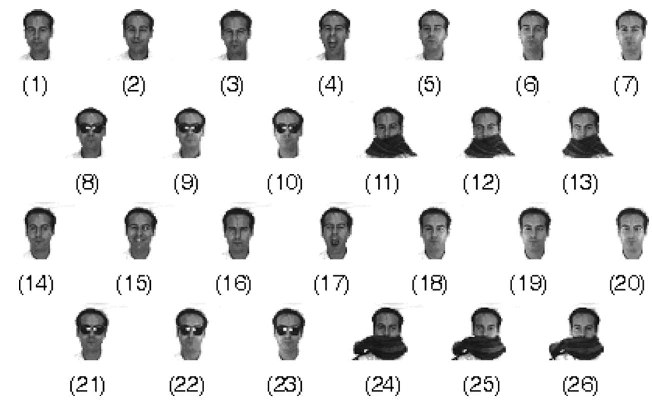

Fig. 1. Examples of the 26 images of one subject in AR face database, where only the nonocclusion images, i.e., images numbered 1–7 and 14–20, are used in our experiment.

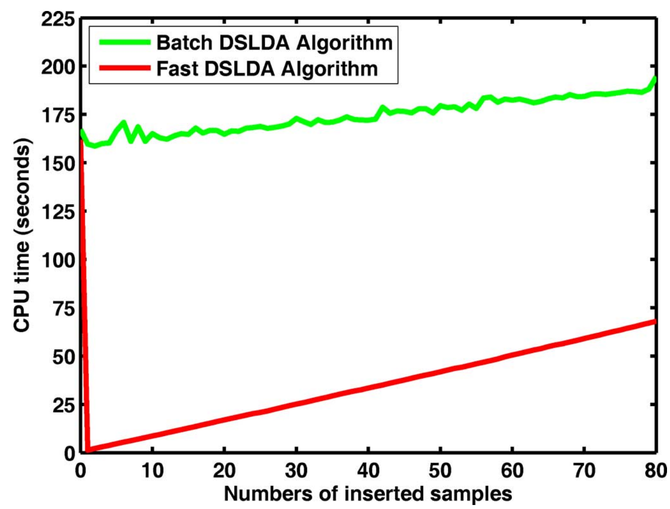

Fig. 2. Plot of the CPU time of the batch DSLDA algorithm and the fast DSLDA algorithm varying with the number of the inserted samples on the AR face database.

1) Select one sample from the testing data set.
2) Use the fast DSLDA algorithm (i.e., Algorithm 2 or 3) to calculate the CPU time of updating the discriminant vectors when the selected samples are inserted into the training data set. At the same time, use the batch algorithm (i.e, Algorithm 1) to calculate the CPU time of computing the discriminant vectors when the selected samples are inserted into the training data set.
3) Select another sample from the remaining samples of the testing data set.
4) Repeat steps 2 and 3 until all the samples of the testing data set are used as the inserted samples.

Fig. 2 shows the results of the experiments, from which we can see that the CPU time of both the fast updating algorithm and the batch algorithm increases with the increase of the number of the inserted samples. However, the fast algorithm (i.e., Algorithm 2 or 3) are more efficient in obtaining the new discriminant vectors, especially when the size of the inserted samples is much less than the size of the training samples.

TABLE III
FACE RECOGNITION ACCURACY ON THE AR DATABASE OF VARIOUS METHODS

	PCA+LDA	DLDA	Null Space	IDR/QR	2DLDA	Wang's DSLDA [9]	Our DSLDA
Recognition rate (%)	95.51	95.41	95.10	95.00	95.41	95.82	**96.53**
Standard deviation (%)	3.75	2.16	4.04	2.74	3.03	2.74	**2.02**

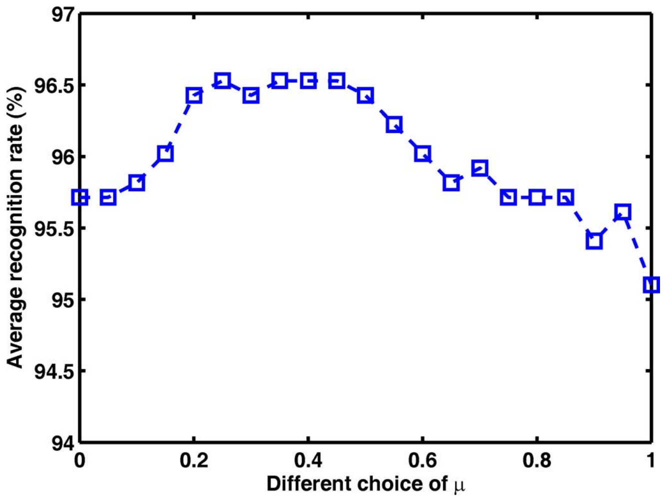

Fig. 3. Recognition rate of our DSLDA method with different choice of the parameter μ in AR the database.

B. Recognition Accuracy Test

In this experiment, we use the two-fold cross-validation strategy to test the recognition performance of our DSLDA method. First, we divide the selected AR face data set into two subsets: one subset consists of the images per subject numbered 1–4 and 14–16, and the other subset consists of the remaining images per subject. Then we choose one subset as the training data set and the other one as the testing data set. Finally, we train our DSLDA method and other discriminant analysis methods on the selected training data set and use the testing data set to evaluate the recognition accuracy. After finishing the recognition process of the various methods, we swap the training data set and the testing data set and then reconduct the experiments. The recognition rates of the two trials are averaged as the final average recognition rate (%). The experimental results are shown in Fig. 3 and Table III, where Fig. 3 shows the recognition rate of the proposed DSLDA method with the different choice of parameter μ, and Table III shows the best recognition rate of each testing method.

From Fig. 3, we can see that the best recognition rate of the proposed DSLDA method can be achieved when the parameter μ lies in [0.25, 0.45]. This also means that both "discriminant subspaces" of the DSLDA method contain useful discriminant information. Moreover, from Table III, we can see that the proposed DSLDA method achieves the best recognition rate among the various methods.

VI. CONCLUSION

We have proposed a modified DSLDA method and a DOI incremental algorithm for updating the DSLDA projection vectors when new samples are inserted into the training data set. Our modified DSLDA method aims to simultaneously extract the discriminant information from both the range space and the null space of the within-class scatter matrix. Therefore, compared with other incremental LDA algorithms, our incremental DSLDA algorithm can extract more discriminant information, and hence can achieve better recognition performance. Our experiments on the AR face database have shown the computational efficiency of our incremental algorithm compared with the batch algorithm. The experimental results also show the superiority of the our DSLDA method over other discriminant analysis methods in terms of the recognition accuracy.

APPENDIX A

From $(\mathbf{x}_i^j - \mathbf{m}_i) = (\mathbf{x}_i^j - \mathbf{x}_i^1) + (\mathbf{x}_i^1 - \mathbf{m}_i)$, we obtain

$$\text{span}\{\mathbf{x}_i^j - \mathbf{m}_i | j = 1, \ldots, n_i\} \subseteq \text{span}\{\mathbf{m}_i - \mathbf{x}_i^1, \mathbf{x}_i^j - \mathbf{x}_i^1 | j = 2, \ldots, n_i\}. \quad (55)$$

Note that $\sum_{j=2}^{n_i}(\mathbf{x}_i^j - \mathbf{x}_i^1) = \sum_{j=1}^{n_i}(\mathbf{x}_i^j - \mathbf{x}_i^1) = n_i(\mathbf{m}_i - \mathbf{x}_i^1)$, we get

$$\text{span}\{\mathbf{x}_i^j - \mathbf{x}_i^1 | j = 2, \ldots, n_i\} = \text{span}\{\mathbf{m}_i - \mathbf{x}_i^1, \mathbf{x}_i^j - \mathbf{x}_i^1 | j = 2, \ldots, n_i\}. \quad (56)$$

From (55) and (56), we get

$$\text{span}\{\mathbf{x}_i^j - \mathbf{m}_i | j=1, \ldots, n_i\} \subseteq \text{span}\{\mathbf{x}_i^j - \mathbf{x}_i^1 | j=2, \ldots, n_i\}. \quad (57)$$

On the other hand, from the fact that $\mathbf{x}_i^j - \mathbf{x}_i^1 = (\mathbf{x}_i^j - \mathbf{m}_i) - (\mathbf{x}_i^1 - \mathbf{m}_i)$, we know that $\mathbf{x}_i^j - \mathbf{x}_i^1$ can be linearly combined by $\mathbf{x}_i^j - \mathbf{m}_i$ $(j = 1, \ldots, n_i)$. Therefore, we have

$$\text{span}\{\mathbf{x}_i^j - \mathbf{x}_i^1 | j=2, \ldots, n_i\} \subseteq \text{span}\{\mathbf{x}_i^j - \mathbf{m}_i | j=1, \ldots, n_i\}. \quad (58)$$

Combining (57) and (58), we obtain

$$\text{span}\{\mathbf{x}_i^j - \mathbf{m}_i | j=1, \ldots, n_i\} = \text{span}\{\mathbf{x}_i^j - \mathbf{x}_i^1 | j=2, \ldots, n_i\}. \quad (59)$$

Moreover, from (3) and (5), we obtain

$$\overline{\mathbf{S}_W(0)} = \text{span}\{\mathbf{x}_i^j - \mathbf{m}_i | i = 1, \ldots, c; j = 1, \ldots, n_i\}. \quad (60)$$

Combining (59) and (60), we have

$$\overline{\mathbf{S}_W(0)} = \text{span}\{\mathbf{x}_i^j - \mathbf{x}_i^1 | i = 1, \ldots, c; j = 2, \ldots, n_i\}. \quad (61)$$

Since $\mathbf{u}_1, \ldots, \mathbf{u}_k$ $(k \leq n - c)$ is an orthonormal basis of $\mathbf{x}_i^j - \mathbf{x}_i^1$ $(i = 1, \ldots, c; j = 2, \ldots, n_i)$, we have

$$\text{span}\{\mathbf{x}_i^j - \mathbf{x}_i^1 | i=1, \ldots, c; j=2, \ldots, n_i\} = \text{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}. \quad (62)$$

From (61) and (62), we obtain

$$\overline{\mathbf{S}_W(0)} = \text{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}.$$

APPENDIX B

From $\mathbf{m}_i = (1/n_i)\sum_j \mathbf{x}_i^j$ and $\mathbf{m} = (1/n)\sum_i\sum_j \mathbf{x}_i^j$, we have

$$\sum_{j=1}^{n_i} \mathbf{x}_i^j = n_i\mathbf{m}_i \quad \text{and} \quad \sum_{i=1}^{c} n_i\mathbf{m}_i = n\mathbf{m}. \tag{63}$$

Thus, we obtain

$$\begin{aligned}\mathbf{S}_B &= \sum_i n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T \\ &= \sum_i n_i\mathbf{m}_i\mathbf{m}_i^T - \mathbf{m}\sum_{i=1}^{c} n_i\mathbf{m}_i^T - \left(\sum_i n_i\mathbf{m}_i\right)\mathbf{m}^T \\ +n\mathbf{m}\mathbf{m}^T &= \sum_i n_i\mathbf{m}_i\mathbf{m}_i^T - n\mathbf{m}\mathbf{m}^T \end{aligned} \tag{64}$$

$$\begin{aligned}\mathbf{S}_W &= \sum_i\sum_j(\mathbf{x}_i^j - \mathbf{m}_i)(\mathbf{x}_i^j - \mathbf{m}_i)^T \\ &= \sum_i\sum_j \mathbf{x}_i^j\mathbf{x}_i^{j^T} - \sum_i\left(\sum_j \mathbf{x}_i^j\right)\mathbf{m}_i^T \\ &\quad - \sum_i \mathbf{m}_i\left(\sum_j \mathbf{x}_i^{j^T}\right) + \sum_i n_i\mathbf{m}_i\mathbf{m}_i^T \\ &= \sum_i\sum_j \mathbf{x}_i^j\mathbf{x}_i^{j^T} - \sum_i n_i\mathbf{m}_i\mathbf{m}_i^T. \end{aligned} \tag{65}$$

From (64), we obtain

$$\tilde{\mathbf{S}}_B = \sum_{t=1}^{c} \tilde{n}_t\tilde{\mathbf{m}}_t\tilde{\mathbf{m}}_t^T - \tilde{n}\tilde{\mathbf{m}}\tilde{\mathbf{m}}^T.$$

Because the inserted sample $\mathbf{x}$ belongs to the ith class, we have

$$\begin{aligned}\tilde{\mathbf{S}}_B &= \sum_{t\neq i} n_t\mathbf{m}_t\mathbf{m}_t^T + (n_i+1)\tilde{\mathbf{m}}_i\tilde{\mathbf{m}}_i^T - (n+1)\tilde{\mathbf{m}}\tilde{\mathbf{m}}^T \\ &= \sum_t n_t\mathbf{m}_t\mathbf{m}_t^T - n_i\mathbf{m}_i\mathbf{m}_i^T + (n_i+1)\tilde{\mathbf{m}}_i\tilde{\mathbf{m}}_i^T \\ &\quad - n\mathbf{m}\mathbf{m}^T + n\mathbf{m}\mathbf{m}^T - (n+1)\tilde{\mathbf{m}}\tilde{\mathbf{m}}^T \\ &= \mathbf{S}_B - n_i\mathbf{m}_i\mathbf{m}_i^T + (n_i+1)\tilde{\mathbf{m}}_i\tilde{\mathbf{m}}_i^T + n\mathbf{m}\mathbf{m}^T \\ &\quad - (n+1)\tilde{\mathbf{m}}\tilde{\mathbf{m}}^T. \end{aligned} \tag{66}$$

Similarly, from (65), we get

$$\tilde{\mathbf{S}}_W = \sum_t\sum_j \mathbf{x}_t^j\mathbf{x}_t^{j^T} + \mathbf{x}\mathbf{x}^T - \sum_t \tilde{n}_t\tilde{\mathbf{m}}_t\tilde{\mathbf{m}}_t^T.$$

Because $\mathbf{x}$ belongs to the ith class, we have

$$\begin{aligned}\tilde{\mathbf{S}}_W &= \sum_t\sum_j \mathbf{x}_t^j\mathbf{x}_t^{j^T} + \mathbf{x}\mathbf{x}^T - \sum_{t\neq i} n_t\mathbf{m}_t\mathbf{m}_t^T - (n_i+1)\tilde{\mathbf{m}}_i\tilde{\mathbf{m}}_i^T \\ &= \sum_t\sum_j \mathbf{x}_t^j\mathbf{x}_t^{j^T} + \mathbf{x}\mathbf{x}^T \\ &\quad - \left(\sum_t n_t\mathbf{m}_t\mathbf{m}_t^T - n_i\mathbf{m}_i\mathbf{m}_i^T + (n_i+1)\tilde{\mathbf{m}}_i\tilde{\mathbf{m}}_i^T\right) \\ &= \sum_t\sum_j \mathbf{x}_t^j\mathbf{x}_t^{j^T} - \sum_t n_t\mathbf{m}_t\mathbf{m}_t^T + \mathbf{x}\mathbf{x}^T + n_i\mathbf{m}_i\mathbf{m}_i^T \\ &\quad - (n_i+1)\tilde{\mathbf{m}}_i\tilde{\mathbf{m}}_i^T \\ &= \mathbf{S}_W + \mathbf{x}\mathbf{x}^T + n_i\mathbf{m}_i\mathbf{m}_i^T - (n_i+1)\tilde{\mathbf{m}}_i\tilde{\mathbf{m}}_i^T. \end{aligned} \tag{67}$$

Therefore, from (66) and (67), we have

$$\begin{aligned}\Delta\mathbf{S}_B &= \tilde{\mathbf{S}}_B - \mathbf{S}_B = n\mathbf{m}\mathbf{m}^T - n_i\mathbf{m}_i\mathbf{m}_i^T + (n_i+1)\tilde{\mathbf{m}}_i\tilde{\mathbf{m}}_i^T \\ &\quad - (n+1)\tilde{\mathbf{m}}\tilde{\mathbf{m}}^T \\ \Delta\mathbf{S}_W &= \tilde{\mathbf{S}}_W - \mathbf{S}_W = \mathbf{x}\mathbf{x}^T + n_i\mathbf{m}_i\mathbf{m}_i^T - (n_i+1)\tilde{\mathbf{m}}_i\tilde{\mathbf{m}}_i^T. \end{aligned}$$

APPENDIX C

If $\mathbf{x}$ is from a new class, then from (64), we get

$$\begin{aligned}\tilde{\mathbf{S}}_B &= \sum_t n_t\tilde{\mathbf{m}}_t\tilde{\mathbf{m}}_t^T - \tilde{n}\tilde{\mathbf{m}}\tilde{\mathbf{m}}^T \\ &= \sum_t n_t\mathbf{m}_t\mathbf{m}_t^T + \mathbf{x}\mathbf{x}^T - (n+1)\tilde{\mathbf{m}}\tilde{\mathbf{m}}^T \\ &= \sum_t n_t\mathbf{m}_t\mathbf{m}_t^T - n\mathbf{m}\mathbf{m}^T + n\mathbf{m}\mathbf{m}^T + \mathbf{x}\mathbf{x}^T \\ &\quad - (n+1)\tilde{\mathbf{m}}\tilde{\mathbf{m}}^T \\ &= \mathbf{S}_B + n\mathbf{m}\mathbf{m}^T + \mathbf{x}\mathbf{x}^T - (n+1)\tilde{\mathbf{m}}\tilde{\mathbf{m}}^T. \end{aligned} \tag{68}$$

Similarly, from (65), we obtain

$$\begin{aligned}\tilde{\mathbf{S}}_W &= \sum_t\sum_j \mathbf{x}_t^j\mathbf{x}_t^{j^T} + \mathbf{x}\mathbf{x}^T - \sum_t \tilde{n}_t\tilde{\mathbf{m}}_t\tilde{\mathbf{m}}_t^T \\ &= \sum_t\sum_j \mathbf{x}_t^j\mathbf{x}_t^{j^T} + \mathbf{x}\mathbf{x}^T - \left(\sum_t n_t\mathbf{m}_t\mathbf{m}_t^T + \mathbf{x}\mathbf{x}^T\right) \\ &= \sum_t\sum_j \mathbf{x}_t^j\mathbf{x}_t^{j^T} - \sum_t n_t\mathbf{m}_t\mathbf{m}_t^T = \mathbf{S}_W. \end{aligned} \tag{69}$$

Therefore, from (68) and (69), we have

$$\begin{aligned}\Delta\mathbf{S}_B &= \tilde{\mathbf{S}}_B - \mathbf{S}_B = n\mathbf{m}\mathbf{m}^T + \mathbf{x}\mathbf{x}^T - (n+1)\tilde{\mathbf{m}}\tilde{\mathbf{m}}^T \\ \Delta\mathbf{S}_W &= \tilde{\mathbf{S}}_W - \mathbf{S}_W = \mathbf{O}_{d\times d}. \end{aligned}$$

REFERENCES

[1] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[2] J. Ye, R. Janardan, C. H. Park, and H. Park, "An optimization criterion for generalized discriminant analysis on undersampled problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 982–994, Aug. 2004.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces versus fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[4] X. Wang and X. Tang, "A unified framework for subspace face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1222–1228, Sep. 2004.

[5] X. Wang and X. Tang, "Unified subspace analysis for face recognition," in *Proc. 9th IEEE Int. Conf. Computer Vision*, 2003, vol. 1, pp. 13–16.

[6] H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana, "Discriminative common vectors for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 1, pp. 4–13, Jan. 2005.

[7] W. Zheng, C. Zou, and L. Zhao, "Real-time face recognition using Gram-Schmidt orthogonalization algorithm for LDA," in *17th Int. Conf. Pattern Recognition (ICPR'04)*, 2004, pp. 403–406.
[8] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data – with application to face recognition," *Pattern Recognit.*, vol. 34, pp. 2067–2070, 2001.
[9] X. Wang and X. Tang, "Dual-space linear discriminant analysis for face recognition," in *Proc. Int. Conf. Computer Vision and Pattern Recognition (CVPR'04)*, 2004, pp. 564–569.
[10] J. Yang, A. F. Frangi, J.-Y. Yang, D. Zhang, and Z. Jin, "A complete kernel Fisher discriminant framework for feature extraction and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 230–244, Feb. 2005.
[11] J. Ye, "Comments on the complete characterization of a family of solutions to a generalized Fisher criterion," *J. Mach. Learning Res.*, vol. 9, pp. 517–519, 2008.
[12] X. Wang and X. Tang, "Random sampling for subspace face recognition," *Int. J. Comput. Vision*, vol. 70, no. 1, pp. 91–104, 2006.
[13] X. Wang and X. Tang, "Random sampling LDA for face recognition," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, 2004, vol. 2, pp. 259–265.
[14] R.-S. Lin, M.-H. Yang, and S. E. Levinson, "Object tracking using incremental fisher discriminant analysis," in *17th Int. Conf. Pattern Recognition (ICPR'04)*, 2004, pp. 757–760.
[15] J. Mao and K. Jain, "Artifical neural networks for feature extraction and multivariate data projection," *IEEE Trans. Neural Netw.*, vol. 6, no. 2, pp. 296–317, Mar. 1995.
[16] C. Chatterjee and V. P. Roychowdhury, "On self-organizing algorithms and networks for class-separability features," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 663–678, May 1997.
[17] J. Ye, Q. Li, H. Xiong, H. Park, R. Janardan, and V. Kumar, "IDR:/QR: An incremental dimension reduction algorithm via QR decomposition," in *Proc. 10th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'04)*, 2004, pp. 364–373.
[18] S. Pang, S. Ozawa, and N. Kasabov, "Incremental linear discriminant analysis for classification of data streams," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 5, pp. 905–914, Oct. 2005.
[19] H. Zhao and P. C. Yuen, "Incremental linear discriminant analysis for face recognition," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 1, pp. 210–221, Feb. 2008.
[20] H. Zha and H. D. Simon, "On updating problems in latent semantic indexing," *SIAM J. Sci. Stat. Comput.*, vol. 21, no. 2, pp. 782–791, 1999.
[21] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Second ed. New York: Academic, 1990.
[22] A. M. Martinez and R. Benavente, The AR Face Database CVC Tech. Rep. 24, 1998.
[23] J. Ye, R. Janardan, and Q. Li, "Two-dimensional linear discriminant analysis," in *18th Ann. Conf. Neural Information Processing Systems (NIPS 2004)*, 2004, pp. 1569–1576.

Wenming Zheng (M'08) received the B.S. degree in computer science from Fuzhou University, Fuzhou, Fujian, China, in 1997, the M.S. degree in computer science from Huaqiao University, Quanzhou, Fujian, China, in 2001, and the Ph.D. degree in signal processing from Southeast University, Nanjing, Jiangsu, China, in 2004.

Since 2004, he has been with the Research Center for Learning Science (RCLS), Southeast University, Nanjing, Jiangsu, China. His research interests include neural computation, pattern recognition, machine learning, and computer vision.

Xiaoou Tang (S'93–M'96–SM'02–F'09) received the B.S. degree from the University of Science and Technology of China, Hefei, in 1990, and the M.S. degree from the University of Rochester, Rochester, NY, in 1991. He received the Ph.D. degree from Massachusetts Institute of Technology, Cambridge, in 1996.

He is a Professor in the Department of Information Engineering, the Chinese University of Hong Kong. He worked as the group manager of the Visual Computing Group at Microsoft Research Asia from 2005 to 2008. His research interests include computer vision, pattern recognition, and video processing.

Dr. Tang received the Best Paper Award at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009. He is a program chair of the IEEE International Conference on Computer Vision (ICCV) 2009 and an Associate Editor of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (PAMI) and *International Journal of Computer Vision* (IJCV).