Direct Kernel Biased Discriminant Analysis: A New Content-based Image Retrieval *Relevance Feedback* Algorithm

Dacheng Tao, Xiaoou Tang, *Senior Member, IEEE*, Xuelong Li, *Member, IEEE*, and Yong Rui, *Senior Member, IEEE*

***Abstract*—In recent years, a variety of relevance feedback (RF) schemes have been developed to improve the performance of content-based image retrieval (CBIR). Given user feedback information, the key to a RF scheme is how to select a subset of image features to construct a suitable dissimilarity measure. Among various RF schemes, biased discriminant analysis (BDA) based RF is one of the most promising. It is based on the observation that all positive samples are alike, while in general each negative sample is negative in its own way. However, to use BDA, the small sample size (SSS) problem is a big challenge, as users tend to give a small number of feedback samples. To explore solutions to this issue, this paper proposes a direct kernel BDA (DKBDA), which is less sensitive to SSS. An incremental DKBDA (IDKBDA) is also developed to speed up the analysis. Experimental results are reported on a real-world image collection to demonstrate that the proposed methods outperform the traditional kernel BDA (KBDA) and the support vector machine (SVM) based RF algorithms.**

***Index Terms*—Relevance feedback (RF), content-based image retrieval (CBIR), biased discriminant analysis (BDA), kernel biased discriminant analysis (KBDA), direct kernel biased discriminant analysis (DKBDA), incremental direct kernel biased discriminant analysis (IDKBDA).**

I. INTRODUCTION

WITH the explosive growth in image records and the rapid increase of computer power, retrieving images from a large-scale image database becomes one of the most active research fields [1], [2]. To give all images text annotations manually is tedious and impractible and to automatically annotate an image is beyond current techniques.

Content-based image retrieval (CBIR) is a technique to retrieve images semantically relevant to the user's query from an image database. It is based on automatically extracted visual features from an image, such as color [3], [4], [10]-[12], texture [5]-[10], [12], and shape [11]-[13]. However, the gap between these low-level visual features and high-level semantic meanings usually leads to poor performance.

Relevance feedback (RF) is a way to bridge this gap and to scale the performance in CBIR systems [14]-[17]. RF focuses on the interactions between the user and the search engine by letting the user labeling semantically positive or negative samples. RF is different from the traditional classification problem because the user is not likely to label a large number of retrieved images.

As a result, small sample learning methods, where the number of the training samples is much smaller than the dimension of the descriptive features, are important in CBIR RF. Discriminant analysis [18]-[26]and the support vector machine (SVM) method [27]-[31] are two small sample learning methods used in recent years to obtain *state-of-the-art* performances.

Discriminant analysis [18] is one of the most popular solutions for the small sample learning problem. In the last twenty years, Fisher linear discriminant analysis (LDA) has been successfully used in face recognition [19]-[23], [26]. LDA was first used in CBIR for feature selection and extracts the most discriminant subset feature for image retrieval. The remaining images in the database were then projected onto the subspace and finally, some similarity or dissimilarity measures were used to sort these images. However, with LDA all negative feedbacks are deemed equivalent, and this is a severe limitation of the method because *all positive examples are alike and each negative example is negative in its own way*. With this observation, biased discriminant analysis (BDA) [24], [25] was developed by Zhou and Huang to scale the performance of CBIR and obtained a more satisfactory result. In the BDA model, the negative feedbacks are required to stay away from the center of the positive feedbacks. Motivated by the kernel trick successfully used in pattern recognition [32], Zhou et al. also generalized the BDA to the kernel feature space as the kernel biased discriminant analysis (KBDA). KBDA performs

Manuscript received October 31, 2004. The first two authors' research related to this paper was fully supported by grants from the Research Grants Council of the Hong Kong SAR.

Dacheng Tao is with the School of Computer Science and Information Systems, Birkbeck College, University of London, Malet Street, London WC1E 7HX, United Kingdom. (e-mail: dacheng@dcs.bbk.ac.uk).

Xiaoou 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).

Xuelong Li is with the School of Computer Science and Information Systems, Birkbeck College, University of London, Malet Street, London WC1E 7HX, United Kingdom. (e-mail: xuelong@dcs.bbk.ac.uk).

Yong Rui is with the Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA. (e-mail: yongrui@microsoft.com).

much better than BDA [24], [25]. Just like LDA, BDA and KBDA also lead to the small sample size (SSS) problem [33] because the number of the sample is much smaller than the dimension of the representative features of images. Traditionally, the SSS problem is solved by the regularization method [33], as in [24], [25].

However, the regularization method to solve the SSS problem is not a good choice for LDA, as is pointed out by many papers on face recognition [20]-[23], [26]. We aim to significantly improve the performance of CBIR RF and utilize the direct idea to the BDA algorithm in the kernel feature space. We name the approach as the direct kernel BDA (DKBDA)[34]. DKBDA is motivated by: (a) direct LDA (DLDA) [23], [26], has been successfully applied to face recognition; (b) unlike face recognition, image retrieval deals with diverse images, so the nonlinear properties of image features should be considered because of the success of kernel algorithms in pattern recognition.

The DKBDA algorithm can be regarded as an enhanced KBDA. According to the kernel trick idea, the original input space is first nonlinearly mapped to an arbitrarily high dimension feature space, in which the distribution of the images' patterns is linearized. Then, the DLDA idea [23], [26] is used to obtain a set of optimal discriminant basis vectors in the kernel feature space. The BDA criterion is modified as in Liu et al. [20], so that a robust result can be gained.

The following section describes the related previous work: BDA, KBDA, and DLDA; DKBDA is then proposed in Section III; in Section IV, an image retrieval system is introduced; in Section V, a large number of experiments validate the effectiveness and efficiency of DKBDA on a large real world image database; possible future work is briefly described in Section VI; finally, Section VII draws conclusions. Detailed deduction of DKBDA is given in Appendix I; Appendix II provides full deduction of the incremental DKBDA (IDKBDA).

II. Previous Work

In this section, previous work including Direct Linear Discriminant Analysis (DLDA), Biased Discriminant Analysis (BDA), and Kernel Biased Discriminant Analysis (KBDA) are introduced.

A. *Direct Linear Discriminant Analysis* (*DLDA*)

Before describing DLDA [23], we first describe Linear Discriminant Analysis (LDA) [18].

LDA tries to find the best discriminating subspace for different classes. It is spanned by a set of vectors $\mathbf{W}$, which aims at maximizing the ratio between $\mathbf{S}_b$ and $\mathbf{S}_w$, the *within-class scatter matrix* and the *between-class scatter matrix*, respectively.

$$\mathbf{W} = \arg\max_{\mathrm{w}} \frac{\left\|\mathbf{W}^{\mathbf{T}}\mathbf{S}_b\mathbf{W}\right\|}{\left\|\mathbf{W}^{\mathbf{T}}\mathbf{S}_w\mathbf{W}\right\|}. \tag{1}$$

Assume the training set contains c individual classes and each class C_i has N_i samples. Then $\mathbf{S}_b$ and $\mathbf{S}_w$ are defined as,

$$\begin{cases} \mathbf{S}_b = \dfrac{1}{N}\displaystyle\sum_{i=1}^{c} N_i\left(\mathbf{m}_i - \mathbf{m}\right)\left(\mathbf{m}_i - \mathbf{m}\right)^T \\ \mathbf{S}_w = \dfrac{1}{N}\displaystyle\sum_{i=1}^{c}\sum_{j=1}^{N_i}\left(\mathbf{x}_j^i - \mathbf{m}_i\right)\left(\mathbf{x}_j^i - \mathbf{m}_i\right)^T, \mathbf{x}_j^i \in C^i, \end{cases} \tag{2}$$

where $N = \sum_{i=1}^{c} N_i$. $\mathbf{m} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{x}_j$ is the mean vector of the total training set. $\mathbf{m}_i = \frac{1}{N_i}\sum_{j=1}^{N_i}\mathbf{x}_j^i$ is the mean vector for the individual class C_i. $\mathbf{x}_j^i$ is the j^{th} sample belongs to class C_i. Therefore, $\mathbf{W}$ can be computed from the eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$. Given c equals 2, LDA changes to Fisher discriminant analysis (FDA); otherwise, it is multiple discriminant analysis (MDA).

LDA has the SSS problem when the number of the training samples is smaller than the dimension of the low-level visual features, which is almost always true for CBIR RF.

Yu et al. [23] propose a DLDA method. It accepts high-dimensional data as input, and optimizes Fisher's criterion directly without any feature extraction or dimension reduction steps. So, it takes advantage of all the information within and outside of the null space of $\mathbf{S}_w$. In this approach, $\mathbf{S}_b$ is first diagonalized, then the null space of $\mathbf{S}_b$ is removed,

$$\mathbf{Y}^T\mathbf{S}_b\mathbf{Y} = \mathbf{D}_b \succ 0, \tag{3}$$

where $\mathbf{Y}$ comprises eigenvectors and $\mathbf{D}_b$ comprises the corresponding non-zero eigenvalues of $\mathbf{S}_b$. $\mathbf{S}_w$ is transformed to

$$\mathbf{K}_w = \mathbf{D}_b^{-1/2}\mathbf{Y}^T\mathbf{S}_w\mathbf{Y}\mathbf{D}_b^{-1/2}. \tag{4}$$

where $\mathbf{K}_w$ is diagonalized by eigen analysis,

$$\mathbf{U}^T\mathbf{K}_w\mathbf{U} = \mathbf{D}_w. \tag{5}$$

The LDA transformation matrix for classification is defined as,

$$\mathbf{W} = \mathbf{Y}\mathbf{D}_b^{-1/2}\mathbf{U}\mathbf{D}_w^{-1/2}. \tag{6}$$

In DLDA, the null space of $\mathbf{S}_b$ is removed, and the discriminant vectors are restricted in the subspace spanned by class centers. It is assumed that the null space of $\mathbf{S}_b$ contains no discriminative information at all.

B. *Biased Discriminant Analysis* (*BDA*)

Zhou *et al.* [24], [25] developed BDA, which defines the (*1+x*)-*class classification problem*. This means there is an unknown number of classes but the user is only interested in one class.

BDA tries to find the subspace to discriminate the positive samples (the only class of concern to the user) and negative samples (unknown number of classes). It is spanned by a set of vectors $\mathbf{W}$ maximizing the ratio between the biased matrix $\mathbf{S}_y$ and the positive covariance matrix $\mathbf{S}_x$,

$$\mathbf{W} = \arg\max_{\mathbf{W}} \frac{\left\| \mathbf{W}^{\mathrm{T}} \mathbf{S}_y \mathbf{W} \right\|}{\left\| \mathbf{W}^{\mathrm{T}} \mathbf{S}_x \mathbf{W} \right\|}. \tag{7}$$

Assume the training set contains N_{x} positive and N_{y} negative samples. $\mathbf{S}_x$ and $\mathbf{S}_y$ can be defined as (8):

$$\begin{cases} \mathbf{S}_x = \sum_{i=1}^{N_x} (\mathbf{x}_i - \mathbf{m}_x)(\mathbf{x}_i - \mathbf{m}_x)^T \\ \mathbf{S}_y = \sum_{i=1}^{N_y} (\mathbf{y}_i - \mathbf{m}_x)(\mathbf{y}_i - \mathbf{m}_x)^T, \end{cases} \tag{8}$$

where $\mathbf{x}_i$ denotes the positive samples, $\mathbf{y}_i$ denotes the negative samples, and $\mathbf{m}_x = \frac{1}{N_x}\sum_{i=1}^{N_x} \mathbf{x}_i$ is the mean vector of the positive samples. $\mathbf{W}$ can be computed from the eigenvectors of $\mathbf{S}_x^{-1}\mathbf{S}_y$. Firstly, BDA minimizes the variance of the positive samples. Then it maximizes the distance between the two centers of the positive feedbacks and all negative feedbacks.

C. *Kernel Biased Discriminant Analysis* (*KBDA*)

The data is in a non-linear space, in which the kernel method is successfully used. Therefore, BDA is generalized to its kernel version, named as KBDA. To obtain the non-linear generalization, the linear input space is mapped to a non-linear kernel feature space:

$$\boldsymbol{\Phi} : R^N \rightarrow \mathbf{F}, \tag{9}$$

$$\mathbf{x} \mapsto \boldsymbol{\Phi}(\mathbf{x}). \tag{10}$$

The data $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n \in R^N$ is mapped from R^N into a potentially much higher dimensional feature space $\mathbf{F}$. Now, given a learning problem, one can consider the BDA in $\mathbf{F}$ instead of R^N. In other words, the idea behind KBDA is to perform the BDA in the feature space $\mathbf{F}$ instead of the input space R^N.

Let $\mathbf{S}_x^{\phi}$ and $\mathbf{S}_y^{\phi}$ be the *positive within-class scatter* and the *negative scatter with respect to positive centroid* matrices in the feature space $\mathbf{F}$. They can be respectively expressed as follows:

$$\begin{cases} \mathbf{S}_x^{\phi} = \sum_{i=1}^{Nx} \left(\boldsymbol{\varphi}(\mathbf{x}_i) - \bar{\boldsymbol{\varphi}}(\mathbf{x})\right)\left(\boldsymbol{\varphi}(\mathbf{x}_i) - \bar{\boldsymbol{\varphi}}(\mathbf{x})\right)^T = \boldsymbol{\Phi}_x \boldsymbol{\Phi}_x^T \\ \mathbf{S}_y^{\phi} = \sum_{i=1}^{Ny} \left(\boldsymbol{\varphi}(\mathbf{y}_i) - \bar{\boldsymbol{\varphi}}(\mathbf{x})\right)\left(\boldsymbol{\varphi}(\mathbf{y}_i) - \bar{\boldsymbol{\varphi}}(\mathbf{x})\right)^T = \boldsymbol{\Phi}_y \boldsymbol{\Phi}_y^T, \end{cases} \tag{11}$$

$$\begin{cases} \boldsymbol{\Phi}_x = \left[(\boldsymbol{\varphi}(\mathbf{x}_1) - \bar{\boldsymbol{\varphi}}(\mathbf{x}))...(\boldsymbol{\varphi}(\mathbf{x}_i) - \bar{\boldsymbol{\varphi}}(\mathbf{x}))...(\boldsymbol{\varphi}(\mathbf{x}_{Nx}) - \bar{\boldsymbol{\varphi}}(\mathbf{x})) \right] \\ \boldsymbol{\Phi}_y = \left[(\boldsymbol{\varphi}(\mathbf{y}_1) - \bar{\boldsymbol{\varphi}}(\mathbf{x}))...(\boldsymbol{\varphi}(\mathbf{y}_i) - \bar{\boldsymbol{\varphi}}(\mathbf{x}))...(\boldsymbol{\varphi}(\mathbf{y}_{Ny}) - \bar{\boldsymbol{\varphi}}(\mathbf{x})) \right] \end{cases} \tag{12}$$

where $\bar{\boldsymbol{\varphi}}(\mathbf{x}) = \frac{1}{N_x}\sum_{i=1}^{N_x} \boldsymbol{\varphi}(\mathbf{x}_i)$ is the centroid of positive samples, N_{x} is the positive samples' number, and N_{y} is the negative samples' number. KBDA determines a set of optimal discriminant basis vectors $\mathbf{W} = \{w_k\}_{k=1}^{m}$, which, according to eigenvectors of $\mathbf{S}_x^{\phi-1}\mathbf{S}_{\mathrm{y}}^{\phi}$, can be obtained to solve the following eigenvalue problem:

$$\mathbf{W} = \arg\max_{\mathbf{W}} \frac{\left\| \mathbf{W}^{\mathrm{T}} \mathbf{S}_y^{\phi} \mathbf{W} \right\|}{\left\| \mathbf{W}^{\mathrm{T}} \mathbf{S}_x^{\phi} \mathbf{W} \right\|}. \tag{13}$$

The dimension of the feature space $\mathbf{F}$ is arbitrarily high, and possibly infinite. Fortunately, there is no need to use the exact $\boldsymbol{\Phi}(\mathbf{x})$ to calculate $\mathbf{W}$, because the kernel method can be utilized to avoid mapping the feature point from the linear input space to a nonlinear kernel feature space. This mapping is based on replacing the dot product with a kernel function in the input space R^N.

In KBDA based RF, the number of feedback samples is much smaller than the dimension of the low-level visual feature. This leads to a degenerated $\mathbf{S}_x^{\phi}$, i.e. the SSS problem or the matrix singular problem. Zhou *et al.* [24], [25] solve the SSS problem by the regularized version $\mathbf{S}_x^{\phi}$ and $\mathbf{S}_y^{\phi}$, which adds small quantities to the diagonal of the scatter matrices. However, this is not an optimal solution and sometimes it may lead to an ill-posed problem, which limits the performance of their method.

III. Direct Kernel Biased Discriminant Analysis (DKBDA) and Its Incremental Version

The regularization method to solve the SSS problem is not a good choice for LDA, as is pointed out by many papers on face recognition [20]-[23], and [26]. We aim to significantly improve the performance of CBIR RF and utilize the direct idea to the BDA algorithm in the kernel feature space. This direct method is proposed based on *all positive examples are alike and each negative example is negative in its own way* [24], [25]. We name the approach as the direct kernel BDA (DKBDA).

DKBDA is motivated by: (a) the fact that direct LDA (DLDA) [23], [26], recently developed for face recognition, has made some advances; (b) Unlike face recognition, image retrieval deals with diverse images, so the nonlinear properties of image features should be considered because of the success of kernel algorithms in pattern recognition.

DKBDA can be regarded as an enhanced KBDA. According to the kernel trick idea, the original input space is first nonlinearly mapped to an arbitrarily high dimension feature space, in which the distribution of the images' patterns is linearized. Then, the DLDA idea [23], [26] is used to obtain a set of optimal discriminant basis vectors in the kernel feature space. The BDA criterion is modified as in Liu *et al.* [20], so that a robust result can be gained. First of all, the kernel matrix $\mathbf{K}$ is introduced:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xy} \\ \mathbf{K}_{yx} & \mathbf{K}_{yy} \end{bmatrix}, \tag{14}$$

where

$$\mathbf{K}_{xx} = \left[k(\mathbf{x}_i, \mathbf{x}_j) \right]_{\substack{1 \le i \le Nx \\ 1 \le j \le Nx}} \quad \mathbf{K}_{xy} = \left[k(\mathbf{x}_i, \mathbf{y}_j) \right]_{\substack{1 \le i \le Nx \\ 1 \le j \le Ny}}$$

$$\mathbf{K}_{yx} = \left[k(\mathbf{y}_i, \mathbf{x}_j) \right]_{\substack{1 \le i \le Ny \\ 1 \le j \le Nx}} \quad \mathbf{K}_{yy} = \left[k(\mathbf{y}_i, \mathbf{y}_j) \right]_{\substack{1 \le i \le Ny \\ 1 \le j \le Ny}}$$

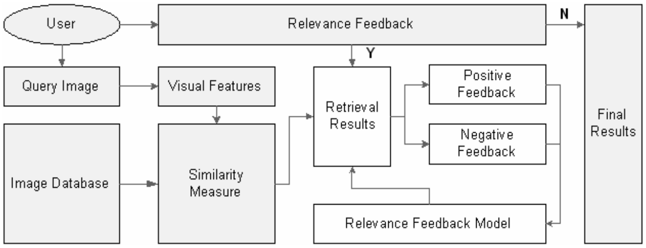

Fig. 1. System flow chart.

where $\mathbf{x}_i$ stands for positive feedback samples, and N_x is the number of positive feedback samples; $\mathbf{y}_i$ stands for negative feedback samples, and N_y is the number of negative feedback samples. $k(.,.)$ is the kernel function. Some typical kernel functions can be employed, such as Polynomial, Gaussian, or Sigmoid based kernel functions.

DKBDA begins from the analysis of the *negative scatter with respect to positive centroid* matrix (11). Since the dimension of $\mathbf{\Phi}_y$ could be arbitrarily infinitive, it is impossible to calculate $\mathbf{S}_y^{\phi} = \mathbf{\Phi}_y \mathbf{\Phi}_y^T$ directly and implement eigen analysis with $\mathbf{S}_x^{\phi}$. Fortunately, this can be avoided through the following analysis:

$$\begin{aligned} &\mathbf{\Phi}_y^T \mathbf{\Phi}_y \mathbf{e}_i = \lambda_i \mathbf{e}_i \Rightarrow \mathbf{\Phi}_y \mathbf{\Phi}_y^T \left(\mathbf{\Phi}_y \mathbf{e}_i\right) = \lambda_i \left(\mathbf{\Phi}_y \mathbf{e}_i\right), \\ &\mathbf{\Phi}_y \mathbf{\Phi}_y^T \mathbf{u}_i = \lambda_i \mathbf{u}_i \Rightarrow \mathbf{u}_i = \mathbf{\Phi}_y \mathbf{e}_i, \\ &\therefore \mathbf{U} = \mathbf{\Phi}_y \mathbf{E}. \end{aligned} \tag{15}$$

The dimension of $\mathbf{\Phi}_y^T \mathbf{\Phi}_y$ is the number of negative RF samples. The next problem is to obtain the matrix.

$$\mathbf{\Phi}_y^T \mathbf{\Phi}_y = \begin{bmatrix} \boldsymbol{\varphi}^T(\mathbf{y}_i)\boldsymbol{\varphi}(\mathbf{y}_j) - \boldsymbol{\varphi}^T(\mathbf{y}_i)\bar{\boldsymbol{\varphi}}(\mathbf{x}) \\ -\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\boldsymbol{\varphi}(\mathbf{y}_j) + \bar{\boldsymbol{\varphi}}^T(\mathbf{x})\bar{\boldsymbol{\varphi}}(\mathbf{x}) \end{bmatrix}_{\substack{i=1,2,\ldots,Ny \\ j=1,2,\ldots,Ny}}. \tag{16}$$

$\boldsymbol{\varphi}^T(\mathbf{y}_i)\boldsymbol{\varphi}(\mathbf{y}_j)$, $\boldsymbol{\varphi}^T(\mathbf{y}_i)\bar{\boldsymbol{\varphi}}(\mathbf{x})$, $\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\boldsymbol{\varphi}(\mathbf{y}_j)$, and $\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\bar{\boldsymbol{\varphi}}(\mathbf{x})$ should then be calculated. The detailed deductions can be seen from the Appendix I and the results are given by the following formulations:

$$\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\bar{\boldsymbol{\varphi}}(\mathbf{x}) = \frac{1}{N_x^2}\mathbf{1}_{Nx,1}^T \mathbf{K}_{xx} \mathbf{1}_{Nx,1}, \tag{17}$$

$$\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\boldsymbol{\varphi}(\mathbf{y}_j) = \frac{1}{N_x}\sum_{m=1}^{N_x} k(\mathbf{x}_m, \mathbf{y}_j), \tag{18}$$

$$\boldsymbol{\varphi}^T(\mathbf{y}_i)\bar{\boldsymbol{\varphi}}(\mathbf{x}) = \frac{1}{N_x}\sum_{m=1}^{N_x} k(\mathbf{y}_i, \mathbf{x}_m), \tag{19}$$

where $\mathbf{1}_{Nx,1}$ is an *Nx* by 1 column vector (all terms equal to 1).

So the following formulations can be obtained according to the kernel matrix (14).

$$\mathbf{\Phi}_y^T \mathbf{\Phi}_y = \mathbf{K}_{yy} - \frac{1}{N_x}\mathbf{K}_{yx}\mathbf{1}_{Nx,Ny} - \frac{1}{N_x}\mathbf{1}_{Ny,Nx}\mathbf{K}_{xy} + \frac{\alpha}{N_x^2}\mathbf{1}_{Ny,Ny}, \tag{20}$$

where $\alpha = \mathbf{1}_{Nx,1}^T \mathbf{K}_{xx} \mathbf{1}_{Nx,1}$. $\mathbf{1}_{Nx,Ny}$ is an *Nx* by *Ny* sized matrix (all terms equal to 1), the same for $\mathbf{1}_{Ny,Nx}$, $\mathbf{1}_{Nx,Nx}$ and $\mathbf{1}_{Ny,Ny}$.

Do eigen analysis with (20), and obtain the non-zero space $\mathbf{E}$ of $\mathbf{\Phi}_y^T \mathbf{\Phi}_y$, so that $\mathbf{E}^T \mathbf{\Phi}_y^T \mathbf{\Phi}_y \mathbf{E} = \mathbf{D}_y \succ 0$. According to (15), $\mathbf{W} = \mathbf{\Phi}_y \mathbf{D}_y^{-1/2} \mathbf{E}$ can be obtained as the normalized non-zero subspace, which can diagonalize the $\mathbf{\Phi}_y \mathbf{\Phi}_y^T$, i.e. $\mathbf{W}^T \mathbf{S}_y^{\phi} \mathbf{W} \succ 0$. Here, no need to calculate $\mathbf{W} = \mathbf{\Phi}_y \mathbf{D}_y^{-1/2} \mathbf{E}$. Similar to the DLDA, the *positive with-in class scatter* matrix is projected onto the non-zero space:

$$\mathbf{W}^T \mathbf{S}_x^{\phi} \mathbf{W} = \mathbf{D}_y^{-1/2} \mathbf{E}^T \mathbf{\Phi}_y^T \mathbf{S}_x^{\phi} \mathbf{\Phi}_y \mathbf{E} \mathbf{D}_y^{-1/2}. \tag{21}$$

From (21), to calculate $\mathbf{W} = \mathbf{\Phi}_y \mathbf{D}_y^{-1/2} \mathbf{E}$ can be avoided. The new problem is to reckon $\mathbf{\Phi}_y^T \mathbf{S}_x^{\phi} \mathbf{\Phi}_y$. With the following deduction (22)-(26), conclusion can be drawn that $\mathbf{\Phi}_y^T \mathbf{S}_x^{\phi} \mathbf{\Phi}_y$ is only related to the kernel matrix (14), just like $\mathbf{\Phi}_y^T \mathbf{\Phi}_y$.

$$\mathbf{\Phi}_y^T \mathbf{S}_x^{\phi} \mathbf{\Phi}_y = \mathbf{\Phi}_y^T \mathbf{\Phi}_x \mathbf{\Phi}_x^T \mathbf{\Phi}_y = \left(\mathbf{\Phi}_x^T \mathbf{\Phi}_y\right)^T \left(\mathbf{\Phi}_x^T \mathbf{\Phi}_y\right). \tag{22}$$

To compute $\mathbf{\Phi}_y^T \mathbf{S}_x^{\phi} \mathbf{\Phi}_y$, only need to calculate $\mathbf{\Phi}_x^T \mathbf{\Phi}_y$.

$$\mathbf{\Phi}_x^T \mathbf{\Phi}_y = \begin{bmatrix} \boldsymbol{\varphi}^T(\mathbf{x}_i)\boldsymbol{\varphi}(\mathbf{y}_j) - \boldsymbol{\varphi}^T(\mathbf{x}_i)\bar{\boldsymbol{\varphi}}(\mathbf{x}) \\ -\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\boldsymbol{\varphi}(\mathbf{y}_j) + \bar{\boldsymbol{\varphi}}^T(\mathbf{x})\bar{\boldsymbol{\varphi}}(\mathbf{x}) \end{bmatrix}_{\substack{i=1,2,\ldots,Nx \\ j=1,2,\ldots,Ny}}. \tag{23}$$

$\mathbf{\Phi}_x^T \mathbf{\Phi}_y$ should be calculated after $\boldsymbol{\varphi}^T(\mathbf{x}_i)\bar{\boldsymbol{\varphi}}(\mathbf{x})$, $\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\boldsymbol{\varphi}(\mathbf{y}_j)$, and $\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\bar{\boldsymbol{\varphi}}(\mathbf{x})$. Here, $\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\boldsymbol{\varphi}(\mathbf{y}_j)$ and $\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\bar{\boldsymbol{\varphi}}(\mathbf{x})$ are calculated in (18) and (17) respectively. In (24), $\boldsymbol{\varphi}^T(\mathbf{x}_i)\bar{\boldsymbol{\varphi}}(\mathbf{x})$ is reckoned.

$$\boldsymbol{\varphi}^T(\mathbf{x}_i)\bar{\boldsymbol{\varphi}}(\mathbf{x}) = \frac{1}{N_x}\sum_{m=1}^{N_x} k(\mathbf{x}_i, \mathbf{x}_m). \tag{24}$$

Then $\mathbf{\Phi}_x^T \mathbf{\Phi}_y$ is obtained by (25).

$$\mathbf{\Phi}_x^T \mathbf{\Phi}_y = \mathbf{K}_{xy} - \frac{1}{N_x}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny} - \frac{1}{N_x}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy} + \frac{\alpha}{N_x^2}\mathbf{1}_{Nx,Ny}. \tag{25}$$

Then $\mathbf{\Phi}_y^T \mathbf{S}_x^{\phi} \mathbf{\Phi}_y$ is obtained by (26) and detailed deduction can be found in Appendix I:

Fig. 2. User interface of the system.

$$\mathbf{\Phi}_y^T \mathbf{S}_x^{\phi} \mathbf{\Phi}_y = \mathbf{A} - \frac{1}{N_x}\mathbf{B} + \frac{1}{N_x^2}\mathbf{C} - \frac{\alpha}{N_x^3}\mathbf{D}, \tag{26}$$

where,

$$\mathbf{A} = \mathbf{K}_{yx}\mathbf{K}_{xy} + \frac{\alpha}{N_x^2}\left(\mathbf{K}_{yx}\mathbf{1}_{Nx,Ny} + \mathbf{1}_{Ny,Nx}\mathbf{K}_{xy}\right) + \frac{\alpha^2}{N_x^3}\mathbf{1}_{Ny,Ny},$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{K}_{yx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny} + \mathbf{K}_{yx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy} \\ +\mathbf{1}_{Ny,Nx}\mathbf{K}_{xx}\mathbf{K}_{xy} + \mathbf{K}_{yx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy} \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} \mathbf{1}_{Ny,Nx}\mathbf{K}_{xx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny} + \mathbf{1}_{Ny,Nx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy} \\ +\mathbf{K}_{yx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny} + N_x\mathbf{K}_{yx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy} \end{pmatrix},$$

$$\mathbf{D} = \begin{pmatrix} \mathbf{1}_{Ny,Nx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny} + N_x\mathbf{K}_{yx}\mathbf{1}_{Nx,Ny} \\ +\mathbf{1}_{Ny,Nx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny} + N_x\mathbf{1}_{Ny,Nx}\mathbf{K}_{xy} \end{pmatrix}.$$

With the idea of DLDA, do the eigen analysis of $\tilde{\mathbf{S}}_x^{\phi} = \mathbf{W}^T\mathbf{S}_x^{\phi}\mathbf{W} = \mathbf{D}_y^{-1/2}\mathbf{E}^T\mathbf{\Phi}_y^T\mathbf{S}_x^{\phi}\mathbf{\Phi}_y\mathbf{E}\mathbf{D}_y^{-1/2}$, and select the eigenvectors $\mathbf{V}$ of $\tilde{\mathbf{S}}_x^{\phi}$ with the smallest eigenvalues $\mathbf{D}_x$, *i.e.*

$$\mathbf{V}^T\tilde{\mathbf{S}}_x^{\phi}\mathbf{V} = \mathbf{D}_x. \tag{27}$$

Finally, the overall projection matrix $\mathbf{H} = \mathbf{E}\mathbf{D}_y^{-1/2}\mathbf{V}\mathbf{D}_x^{-1/2}$ is established.

It is possible that some diagonal values in the matrix $\mathbf{D}_x$ are zero, which means that $\mathbf{D}_x^{-1/2}$ does not exist. The zero eigenvalue problems can be avoided based on a modified KBDA criterion, according to [20]. The modified KBDA criterion is:

$$\mathbf{W} = \arg\max_{\mathbf{W}} \frac{\left\|\mathbf{W}^{\mathrm{T}}\mathbf{S}_y^{\phi}\mathbf{W}\right\|}{\left\|\mathbf{W}^{\mathrm{T}}\left(\mathbf{S}_x^{\phi} + \mathbf{S}_y^{\phi}\right)\mathbf{W}\right\|}. \tag{28}$$

The modified criterion equals to the original KBDA criterion according to the proof in [20]. Upon the modified KBDA criterion, the singular value problem can be avoided because $\left\|\mathbf{W}^{\mathrm{T}}\mathbf{S}_y^{\phi}\mathbf{W}\right\| = \mathbf{I}$.

With the optimal discriminant directions, which are drawn from the previous derivations, the projection of a new pattern $\mathbf{z}$ to $\mathbf{H}$ is given by:

$$\begin{aligned} f(\mathbf{z}) &= \left\{\left(\mathbf{\Phi}_y\mathbf{D}_y^{-1/2}\mathbf{E}\right)\cdot\left(\mathbf{V}\mathbf{D}_x^{-1/2}\right)\right\}^T\phi(\mathbf{z}) = \mathbf{H}^T\mathbf{\Phi}_y^T\phi(\mathbf{z}) \\ &= \mathbf{H}^T\left(\sum_{i=1}^{Ny}k(\mathbf{y}_i,\mathbf{z}) - \frac{1}{N_x}\sum_{j=1}^{Nx}k(\mathbf{x}_j,\mathbf{z})\right). \end{aligned} \tag{29}$$

DKBDA algorithm is summarized in Table 1.

TABLE II
IDKBDA ALGORITHM

Input	In the i^{th} iteration, we have N_{x} positive samples and N_{y} negative samples and in the $(i+1)^{\text{th}}$ iteration, we have L_{x} incremental positive samples and L_{y} incremental negative samples. $\mathbf{x}_i$ stands for the positive feedback samples; $\mathbf{y}_i$ stands for the negative feedback samples; $k(.,.)$ is the kernel function; $\mathbf{z}$ stands for the testing sample.
a.	Calculate the incremental kernel matrices ${}^1\mathbf{K}_{xx}$, ${}^1\mathbf{K}_{xy}$, ${}^1\mathbf{K}_{yx}$, ${}^1\mathbf{K}_{yy}$, ${}^2\mathbf{K}_{xx}$, ${}^2\mathbf{K}_{xy}$, ${}^3\mathbf{K}_{xy}$, and ${}^2\mathbf{K}_{yy}$ according to Appendix II.
b.	Calculate $\mathbf{\Phi}_y^T\mathbf{\Phi}_y$, the incremental version of $\mathbf{\Phi}_y^T\mathbf{\Phi}_y$, according to Appendix II.
c.	Extract the prime subspace of $\mathbf{\Phi}_y^T\mathbf{\Phi}_y$ by eigen analysis or the incremental SVD in theorem 2. Then $\mathbf{E}$ is extracted to satisfy $\mathbf{E}^T\mathbf{\Phi}_y^T\mathbf{\Phi}_y\mathbf{E} = \mathbf{D}_y \succ 0$.
d.	Calculate $\mathbf{\Phi}_x^T\mathbf{\Phi}_y$, the incremental version of $\mathbf{\Phi}_x^T\mathbf{\Phi}_y$, according to Appendix II.
e.	With the modified KBDA criterion, select eigenvectors $\mathbf{V}$ of $\tilde{\mathbf{S}}_x^{\phi} = \mathbf{D}_y^{-1/2}\mathbf{E}^T\mathbf{\Phi}_y^T\tilde{\mathbf{S}}_x^{\phi}\mathbf{\Phi}_y\mathbf{E}\mathbf{D}_y^{-1/2}$ with the smallest eigenvalues $\mathbf{D}_x$ by eigen analysis. Here, $\mathbf{\Phi}_y^T\tilde{\mathbf{S}}_x^{\phi}\mathbf{\Phi}_y = \left(\mathbf{\Phi}_x^T\mathbf{\Phi}_y\right)^T\left(\mathbf{\Phi}_x^T\mathbf{\Phi}_y\right)$.
f.	Calculate the kernel projection matrix $\mathbf{H} = \mathbf{E}\mathbf{D}_y^{-1/2}\mathbf{V}\mathbf{D}_x^{-1/2}$.
a.	For a given pattern, the IDKBDA transformation is: $f(\mathbf{z}) = \mathbf{H}^T\left(\sum_{i=1}^{Ny}k(\mathbf{y}_i,\mathbf{z}) - \frac{1}{N_x}\sum_{j=1}^{Nx}k(\mathbf{x}_j,\mathbf{z})\right)$.
Output	$f(\mathbf{z})$ stands for the projected testing sample.

The ranks of $\mathbf{K}$, $\mathbf{\Phi}_y^T\mathbf{\Phi}_y$ and $\mathbf{\Phi}_x^T\mathbf{\Phi}_y$ are $\operatorname{rank}(\mathbf{K}) \le N_x + N_y$, $\operatorname{rank}\left(\mathbf{\Phi}_y^T\mathbf{\Phi}_y\right) \le N_y$ and $\operatorname{rank}\left(\mathbf{\Phi}_x^T\mathbf{\Phi}_y\right) \le \min\left(N_x - 1, N_y\right)$; the rank of $\mathbf{\Phi}_y^T\mathbf{S}_x^{\phi}\mathbf{\Phi}_y$ can be calculated by $\operatorname{rank}\left(\mathbf{\Phi}_y^T\mathbf{S}_x^{\phi}\mathbf{\Phi}_y\right) = \operatorname{rank}\left(\left(\mathbf{\Phi}_x^T\mathbf{\Phi}_y\right)^T\left(\mathbf{\Phi}_x^T\mathbf{\Phi}_y\right)\right) = \operatorname{rank}\left(\mathbf{\Phi}_x^T\mathbf{\Phi}_y\right) \le \min\left(N_x - 1, N_y\right)$; $\operatorname{rank}(\mathbf{H}) \le \min\left(N_x - 1, N_y\right)$ is the rank of $\mathbf{H}$, therefore, $\dim\left(f(\mathbf{z})\right) \le N_x - 1$. DKBDA chooses the intersection space of $\overline{\mathbf{S}_x^{\phi}} \cap \mathbf{S}_y^{\phi}$, where $\mathbf{R}^L = \mathbf{S}_x^{\phi} \oplus \overline{\mathbf{S}_x^{\phi}}$ (L is determined by the kernel parameter and kernel function).

The above theoretical rank analyses show that DKBDA eliminates the SSS problem. Based on DKBDA, the CBIR RF performances can be much improved. A large number of experimental results are given in Section V.

DKBDA can be accelerated by the incremental technique. The deduction of the incremental DKBDA (IDKBDA) is given in Appendix II. The algorithm of IDKBDA is provided in Table 2 and many comparative experimental results are also provided in the Section V of this article.

IV. Image Retrieval System

With CBIR the search engine is required to feedback the most semantically relevant images after each previous RF iteration. The user will not label many images for each iteration and will usually only do a few iterations. Thus, the following CBIR framework is used into which any RF algorithm can be embedded.

As shown in Figure 1, when a query is submitted, its low-level visual features are extracted. Then, all images in the database are sorted based on a similarity metric. If the user is satisfied with the result, the retrieval process is ended. If the user is not satisfied s/he can label some images as positive feedbacks and/or some images as negative feedbacks. Using this feedback process, the system is trained based on machine learning using the embedded RF algorithm. Then, all the images are re-sorted based on the recalculated similarity metric. If the user is still not content with the result, s/he repeats the process.

Figure 2 shows our query by example (QBE) system GUI. In our experiments the user first selects an image from the gallery and this image is then shown in the Preview Image window. Secondly, the user licks the "Retrieval" button, and the images in the gallery are sorted using the similarity metric. Then, the user provides feedback by clicking on the "thumb up" or "thumb down" button according to his/her judgment of the relevance of the sorted images. Finally the user clicks the "Retrieval" button to re-sort the images in the gallery, which uses this feedback information. The last two steps can iterate to obtain a more satisfactory result. The number of iterations is shown in the Query Image/# Feedback window.

The "All Images" tab-page excludes the images marked as either relevant or irrelevant in previous iterations. For the next iteration only these images are re-sorted using the further modified metric. Consequently, the images marked for next iteration do not overlap with the previous feedback images. The "Retrieved" tab-page contains all images marked as relevant in previous iterations plus a number of the top images from the latest iteration.

V. Experiment

In this section we report the results of a large number of experiments in which we took the CBIR platform described in the previous section and compared the performance between KDBA, CSM, and our new DKBDA algorithms for RF. For the experiments we used part of the Corel image database [9], a real world database comprising 10,800 images. The images shown in Figure 2 are from this database.

In the Corel Photo Gallery, each folder includes 100 images. However, the folders' names are not suitable as conceptual classes, because many images with similar concepts are not in the same folder and some images whose semantic contents are quite different are in the same folder. The existing folders in the Corel Photo Gallery were therefore ignored and all 10,800 images were manually divided into 80 concept groups. These concept groups were only used in the evaluation of the results of our experiments.

Generally in a CBIR RF system images are represented by the three main features: color [3], [4], and [10]-[12], texture [5]-[10], [12], and shape [11]-[13]. For the color feature we select three measures, hue, saturation, and value. We use these to form a histogram [3]. Hue and saturation are both quantized into 8 bins and value into 4 bins. A 128 dimensional Color coherence vector (CCV) [4] in Lab color space and a 9 dimensional color moment feature in Luv color space are both employed. For the texture feature a pyramidal wavelet transform (PWT) is extracted from the Y component in YCbCr space. Every image is decomposed by the traditional pyramid-type wavelet transform with Haar wavelet. The mean and standard deviation are calculated in terms of the sub-bands at each decomposed level. PWT results in a feature vector of 24 values. In addition, we also extract the tree-structured wavelet transform (TWT) in form of a 104 dimensional feature.

Each of these features has its own power to characterize a type of image content. The system combines the color and texture features into a feature vector, and then normalizes it into a normal distribution.

Precision is widely used to evaluate retrieval performance. It is the ratio of the number of relevant images retrieved in the top N retrieved images. In our experiments, comparisons are made of the performances of the BDA, KBDA [24], [25], SVM based RF [28], and the direct BDA (DBDA is similar to the DLDA. We can substitute $\mathbf{S}_b$ and $\mathbf{S}_w$ by $\mathbf{S}_y$ and $\mathbf{S}_x$, respectively. With this substitution, this direct version of BDA is obtained), DKBDA and its incremental version, IDKBDA.

Experiments with 300 different query images were performed. In the experiments there were 9 iterations. For each iteration the top 48 images resulting from the re-sorted results were examined serially from the top and each image was marked as correct or incorrect. The first 5 correct images and the first 5 incorrect images were then used as feedback unless fewer such images were found among the top 48 in which case the fewer number found was used as feedback.

As can be seen in Figure 3, the proposed DBDA algorithm consistently outperforms the BDA algorithm and the SVM RF algorithm. The images for the 300 query experiments were randomly selected. The first six figures show the average precision for the 300 experiments for the top 10, 20, 30, 40, 50, and 60 results. We note that for the results DBDA clearly gives a superior performance. There is more benefit from further iterations. In the case of the top 10 results, after 4 iterations, the precision of the proposed DBDA is already higher than 90% while 7 iterations are required for the BDA algorithm and more than 9 iterations for the SVM RF. When more top results are considered, DKBDA again gives superior performance. The last six figures show the standard deviation for the 300 experiments for the top 10, 20, 30, 40, and 60 results. These six figures correspond to the above six average precision figures, respectively. We note that the standard deviation of the DBDA is the smaller than that of BDA and SVM RFs under all of our experimental conditions. This shows that the proposed DBDA is stable for the retrieval problem.

As shown in Figure 4, in the kernel space the DKBDA also outperforms the KBDA consistently. The first six figures show the average precision for the 300 experiments for the top 10, 20, 30, 40, 50, and 60 results. Comparing with the experimental

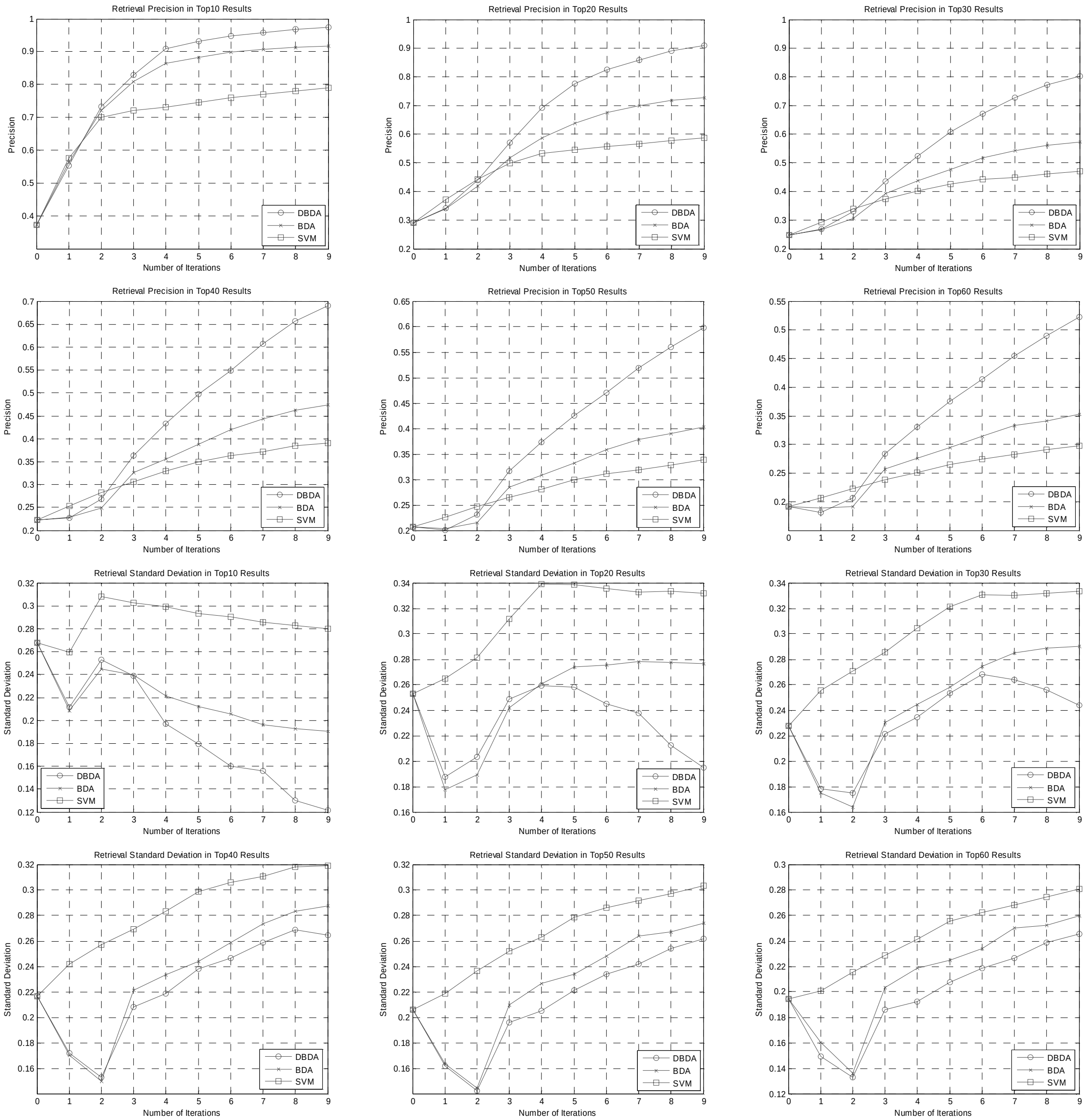

Fig. 3. The average precision and the standard deviation of DBDA, BDA, and SVM.

results in Figure 3, both KBDA and DKBDA perform better than their non-kernel space versions. In the case of the top 10 results, after 3 iterations, precisions of both KBDA and DKBDA are higher than 90%. Note that after 3 iterations there is little further improvement with any of the algorithms but that DKBDA still clearly gives a superior performance. The corresponding standard deviation is shown by the last six figures in Figure 4 for the top 10, 20, 30, 40, and 60 results, respectively. DKBDA gives smallest values in most cases. In other word, the DKBDA is more stable than KBDA.

DKBDA is also compared with its incremental version IDKBDA in all the experiments reported in Figure 4. For all situations, the curves of IDKBDA press close to the DKBDA curves. Therefore, IDKBDA is proved to be of approximately the same capabilities as DKBDA, but it can speed up the DKBDA remarkably by saving about 20% of the running time (9 hours and 11 hours respectively for all the 300 queries and 9 iterations for each query).

The problem of mislabeled samples is an open issue in small sample learning. The number of labeled samples is small. Therefore, when the number of the mislabeled samples is smaller than the correctly labeled samples, the learning machine can still obtain a correct model for the retrieval process by ignoring the minor mistake. However, if a user mislabels too many images during the relevance feedback, the learning will be misled to an incorrect retrieval model. Thereafter, the retrieval system cannot give a satisfactory performance.

In our experiments, the computer does the relevance feedback iterations automatically without mislabeled samples using the 80 concept groups described previously.

For an experiment the concept group of the randomly selected image, which was to be used as the query image, was noted. For each iteration this concept group was compared with the concept groups of the top sorted images and where they are the same the image was labelled the positive feedback (“thumbs

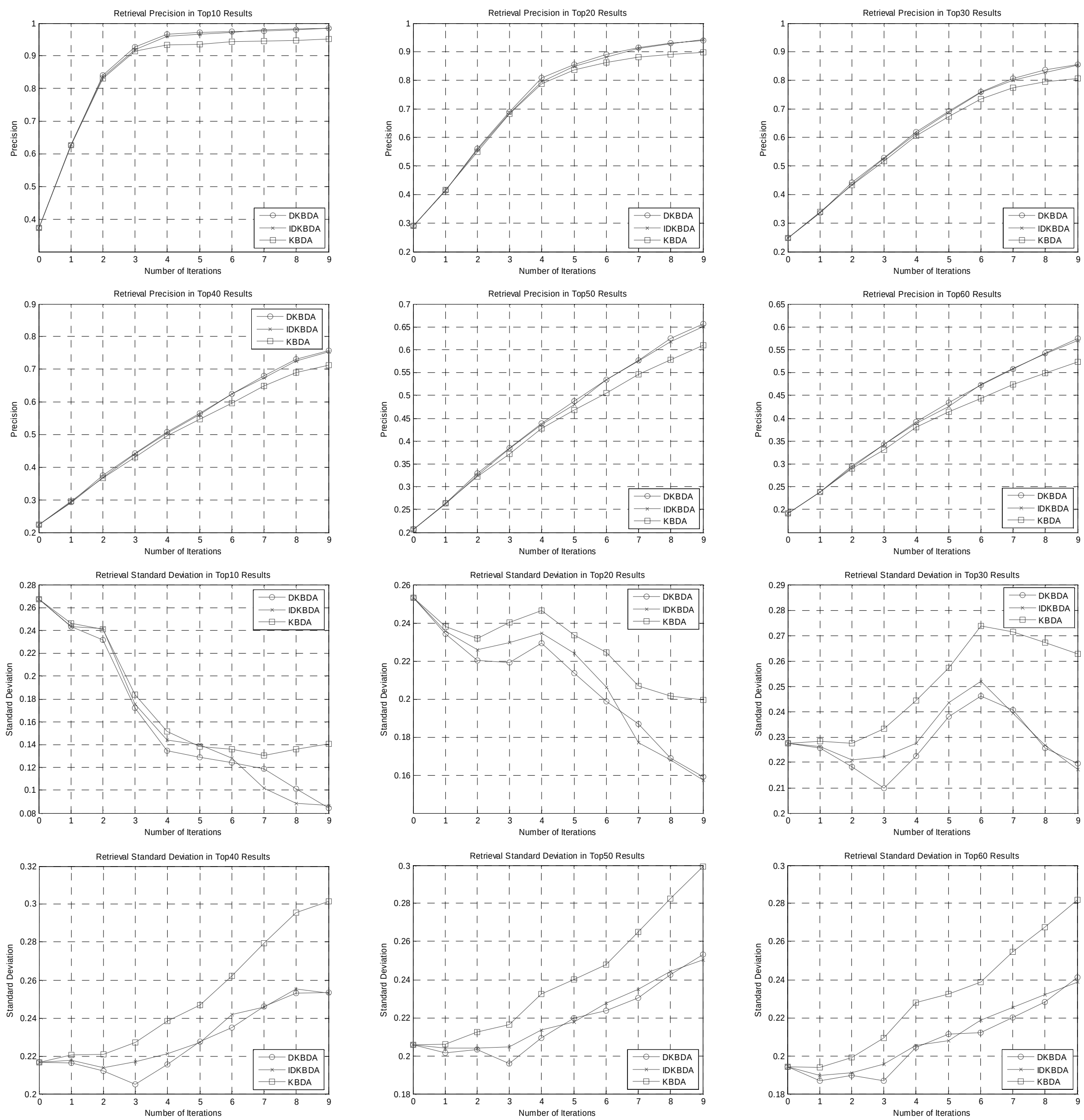

Fig. 4. The average precision and the standard deviation of DKBDA, IDKBDA, and KBDA.

up") and where they are different the negative feedback ("thumbs down").

In all experiments, the Gaussian kernel $k(\mathbf{x},\mathbf{y}) = \mathrm{e}^{-\rho\|\mathbf{x}-\mathbf{y}\|^2}$ is chosen. For SVM RF we chose $\rho = 1$ and for the KBDA, DKBDA, and IDKBDA based RFs we chose $\rho = 0.1$. These values were chosen to give the best performance for these methods, respectively.

VI. Discussion and Future Work

In the proposed CBIR system and its RF algorithms, several aspects can be improved. For example, indexing techniques can improve both the speed and the precision. More low-level visual features can help better characterize the content of an image. The parameters of the kernel-based RF algorithms can be further tuned.

A. Indexing: a much larger image database will be utilized in the current platform. To accelerate the retrieval speed, the indexing of database is important. Recently, many image-indexing algorithms have been developed. There are two major styles, each of which has its intrinsic advantage. (1) Classification based indexing [35] focuses on the improvement of retrieval precision. In this method, each image is assigned one or more distinct labels. Then, based on these labels, the indexing can be constructed through semantic classifications. Thereafter, the search results will cater to most of the users. (2) Low-level visual feature based indexing is employed to speed up the retrieval. There are many feature-based indexing approaches such as a variety of tree-based indexing structures for high-dimensional databases and VQ and VA methods [2]. A promising approach is to combine the feature and classification information in the indexing structure, so that both speed and precision can be improved.

B. Image Representation: there are many other low-level visual features for image representation. New features may outperform the traditional ones, e.g. the Gabor wavelet feature [6] and the edge direction histogram [11].

C. Kernel Parameter Selection: In DKBDA, different choices of kernel parameters affect its performance. How to select the kernel parameters is still an open issue. Recently, the tuning method was used to select the SVM parameters [28]. In the future, we plan to generalize the tuning method to select the parameters of kernel-based algorithms. For RF in CBIR, the training size of the training set is small, so the leave-one-out method to tune the parameters can be used.

VII. Conclusion

Utilizing the direct idea to the biased discriminant analysis, this paper proposed a straightforward method of direct kernel BDA to solve the small sample size problem of the modified BDA in the kernel feature space. DKBDA removes the null space of the negative scatter with respect to the positive centroid matrix, and then the eigenvectors of the positive with-in class scatter matrix corresponding to the smallest eigenvalues are extracted as the most discriminant directions in the kernel space. Incremental DKBDA is also developed to speed up the DKBDA. From a large number of evaluation experiments based on the Corel image database of 10, 800 images with 80 semantic concepts, the conclusion can be drawn that DKBDA and IDKBDA outperform both the traditional kernel BDA and the support vector machine RF.

Appendix I

$\mathbf{\Phi}_y^T\mathbf{\Phi}_y$, $\mathbf{\Phi}_x^T\mathbf{\Phi}_y$, and $\mathbf{\Phi}_y^T\mathbf{S}_x^\phi\mathbf{\Phi}_y$ are calculated.

1. Calculate $\mathbf{\Phi}_y^T\mathbf{\Phi}_y$

$$\mathbf{\Phi}_y^T\mathbf{\Phi}_y=\left[\left(\boldsymbol{\varphi}^T(\mathbf{y}_i)-\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\right)\left(\boldsymbol{\varphi}(\mathbf{y}_j)-\bar{\boldsymbol{\varphi}}(\mathbf{x})\right)\right]_{\substack{i=1,2,\ldots,Ny\\j=1,2,\ldots,Ny}}$$
$$=\left[\boldsymbol{\varphi}^T(\mathbf{y}_i)\boldsymbol{\varphi}(\mathbf{y}_j)-\boldsymbol{\varphi}^T(\mathbf{y}_i)\bar{\boldsymbol{\varphi}}(\mathbf{x})-\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\boldsymbol{\varphi}(\mathbf{y}_j)+\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\bar{\boldsymbol{\varphi}}(\mathbf{x})\right]_{\substack{i=1,2,\ldots,Ny\\j=1,2,\ldots,Ny}}.$$

$$\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\bar{\boldsymbol{\varphi}}(\mathbf{x})=\left(\frac{1}{N_x}\sum_{m=1}^{N_x}\boldsymbol{\varphi}(\mathbf{x}_m)\right)^T\left(\frac{1}{N_x}\sum_{i=n}^{N_x}\boldsymbol{\varphi}(\mathbf{x}_n)\right)$$
$$=\frac{1}{N_x^2}\sum_{m=1}^{N_x}\sum_{n=1}^{N_x}\boldsymbol{\varphi}^T(\mathbf{x}_m)\boldsymbol{\varphi}(\mathbf{x}_n)=\frac{1}{N_x^2}\sum_{m=1}^{N_x}\sum_{n=1}^{N_x}k(\mathbf{x}_m,\mathbf{x}_n)$$
$$=\frac{1}{N_x^2}\mathbf{1}_{Nx,1}^T\mathbf{K}_{xx}\mathbf{1}_{Nx,1},$$

$$\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\boldsymbol{\varphi}(\mathbf{y}_j)=\left(\frac{1}{N_x}\sum_{m=1}^{N_x}\boldsymbol{\varphi}(\mathbf{x}_m)\right)^T\boldsymbol{\varphi}(\mathbf{y}_j)$$
$$=\frac{1}{N_x}\sum_{m=1}^{N_x}\boldsymbol{\varphi}^T(\mathbf{x}_m)\boldsymbol{\varphi}(\mathbf{y}_j)=\frac{1}{N_x}\sum_{m=1}^{N_x}k(\mathbf{x}_m,\mathbf{y}_j),$$

$$\boldsymbol{\varphi}^T(\mathbf{y}_i)\bar{\boldsymbol{\varphi}}(\mathbf{x})=\boldsymbol{\varphi}^T(\mathbf{y}_i)\left(\frac{1}{N_x}\sum_{m=1}^{N_x}\boldsymbol{\varphi}(\mathbf{x}_m)\right)$$
$$=\frac{1}{N_x}\sum_{m=1}^{N_x}\boldsymbol{\varphi}^T(\mathbf{y}_i)\boldsymbol{\varphi}(\mathbf{x}_m)=\frac{1}{N_x}\sum_{m=1}^{N_x}k(\mathbf{y}_i,\mathbf{x}_m),$$

where $\mathbf{1}_{Nx,1}$ is an Nx by 1 column vector (all terms equal to 1).

Consequently, $\mathbf{\Phi}_y^T\mathbf{\Phi}_y$ is given by:

$$\mathbf{\Phi}_y^T\mathbf{\Phi}_y$$
$$=\left[\boldsymbol{\varphi}^T(\mathbf{y}_i)\boldsymbol{\varphi}(\mathbf{y}_j)-\boldsymbol{\varphi}^T(\mathbf{y}_i)\bar{\boldsymbol{\varphi}}(\mathbf{x})-\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\boldsymbol{\varphi}(\mathbf{y}_j)+\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\bar{\boldsymbol{\varphi}}(\mathbf{x})\right]_{\substack{i=1,2,\ldots,Nx\\j=1,2,\ldots,Nx}}$$
$$=\left[k(\mathbf{y}_i,\mathbf{y}_j)-\frac{1}{N_x}\sum_{m=1}^{N_x}k(\mathbf{y}_i,\mathbf{x}_m)-\frac{1}{N_x}\sum_{m=1}^{N_x}k(\mathbf{x}_m,\mathbf{y}_j)+\frac{1}{N_x^2}\mathbf{1}_{Nx,1}^T\mathbf{K}_{xx}\mathbf{1}_{Nx,1}\right]_{\substack{i=1,2,\ldots,Ny\\j=1,2,\ldots,Ny}}$$
$$=\mathbf{K}_{yy}-\frac{1}{N_x}\mathbf{K}_{yx}\mathbf{1}_{Nx,1}\mathbf{1}_{Ny,1}^T-\frac{1}{N_x}\mathbf{1}_{Ny,1}\mathbf{1}_{Nx,1}^T\mathbf{K}_{xy}+\frac{1}{N_x^2}\mathbf{1}_{Nx,1}^T\mathbf{K}_{xx}\mathbf{1}_{Nx,1}\mathbf{1}_{Ny,Ny}$$
$$=\mathbf{K}_{yy}-\frac{1}{N_x}\mathbf{K}_{yx}\mathbf{1}_{Nx,Ny}-\frac{1}{N_x}\mathbf{1}_{Ny,Nx}\mathbf{K}_{xy}+\frac{\alpha}{N_x^2}\mathbf{1}_{Ny,Ny},$$

where $\alpha=\mathbf{1}_{Nx,1}^T\mathbf{K}_{xx}\mathbf{1}_{Nx,1}$. $\mathbf{1}_{Nx,Ny}$, $\mathbf{1}_{Ny,Nx}$, $\mathbf{1}_{Nx,Nx}$, and $\mathbf{1}_{Ny,Ny}$ are Nx by Ny sized matrices with all terms equal to 1.

$$\mathbf{\Phi}_x^T\mathbf{\Phi}_y=\left[\left(\boldsymbol{\varphi}^T(\mathbf{x}_i)-\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\right)\left(\boldsymbol{\varphi}(\mathbf{y}_j)-\bar{\boldsymbol{\varphi}}(\mathbf{x})\right)\right]_{\substack{i=1,2,\ldots,Nx\\j=1,2,\ldots,Ny}}$$
$$=\left[\boldsymbol{\varphi}^T(\mathbf{x}_i)\boldsymbol{\varphi}(\mathbf{y}_j)-\boldsymbol{\varphi}^T(\mathbf{x}_i)\bar{\boldsymbol{\varphi}}(\mathbf{x})-\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\boldsymbol{\varphi}(\mathbf{y}_j)+\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\bar{\boldsymbol{\varphi}}(\mathbf{x})\right]_{\substack{i=1,2,\ldots,Nx\\j=1,2,\ldots,Ny}}.$$

$$\boldsymbol{\varphi}^T(\mathbf{x}_i)\bar{\boldsymbol{\varphi}}(\mathbf{x})=\boldsymbol{\varphi}^T(\mathbf{x}_i)\left(\frac{1}{N_x}\sum_{m=1}^{N_x}\boldsymbol{\varphi}(\mathbf{x}_m)\right)$$
$$=\frac{1}{N_x}\sum_{m=1}^{N_x}\boldsymbol{\varphi}^T(\mathbf{x}_i)\boldsymbol{\varphi}(\mathbf{x}_m)=\frac{1}{N_x}\sum_{m=1}^{N_x}k(\mathbf{x}_i,\mathbf{x}_m).$$

Therefore, $\mathbf{\Phi}_x^T\mathbf{\Phi}_y$ is obtained as

$$\mathbf{\Phi}_x^T\mathbf{\Phi}_y$$
$$=\left[\boldsymbol{\varphi}^T(\mathbf{x}_i)\boldsymbol{\varphi}(\mathbf{y}_j)-\boldsymbol{\varphi}^T(\mathbf{x}_i)\bar{\boldsymbol{\varphi}}(\mathbf{x})-\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\boldsymbol{\varphi}(\mathbf{y}_j)+\bar{\boldsymbol{\varphi}}^T(\mathbf{x})\bar{\boldsymbol{\varphi}}(\mathbf{x})\right]_{\substack{i=1,2,\ldots,Nx\\j=1,2,\ldots,Ny}}$$
$$=\left[k(\mathbf{x}_i,\mathbf{y}_j)-\frac{1}{N_x}\sum_{m=1}^{N_x}k(\mathbf{x}_i,\mathbf{x}_m)-\frac{1}{N_x}\sum_{m=1}^{N_x}k(\mathbf{x}_m,\mathbf{y}_j)+\frac{1}{N_x^2}\mathbf{1}_{Nx,1}^T\mathbf{K}_{xx}\mathbf{1}_{Nx,1}\right]_{\substack{i=1,2,\ldots,Nx\\j=1,2,\ldots,Ny}}$$
$$=\mathbf{K}_{xy}-\frac{1}{N_x}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny}-\frac{1}{N_x}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy}+\frac{\alpha}{N_x^2}\mathbf{1}_{Nx,Ny}.$$

3. Calculate $\mathbf{\Phi}_y^T\mathbf{S}_x^\phi\mathbf{\Phi}_y$

$$\mathbf{\Phi}_y^T\mathbf{S}_x^\phi\mathbf{\Phi}_y$$
$$=\left(\mathbf{K}_{xy}-\frac{1}{N_x}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny}-\frac{1}{N_x}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy}+\frac{\alpha}{N_x^2}\mathbf{1}_{Nx,Ny}\right)^T$$
$$\cdot\left(\mathbf{K}_{xy}-\frac{1}{N_x}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny}-\frac{1}{N_x}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy}+\frac{\alpha}{N_x^2}\mathbf{1}_{Nx,Ny}\right)$$
$$=\mathbf{K}_{yx}\mathbf{K}_{xy}-\frac{1}{N_x}\mathbf{K}_{yx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny}-\frac{1}{N_x}\mathbf{K}_{yx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy}+\frac{\alpha}{N_x^2}\mathbf{K}_{yx}\mathbf{1}_{Nx,Ny}$$
$$-\frac{1}{N_x}\mathbf{1}_{Ny,Nx}\mathbf{K}_{xx}\mathbf{K}_{xy}+\frac{1}{N_x^2}\mathbf{1}_{Ny,Nx}\mathbf{K}_{xx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny}+\frac{1}{N_x^2}\mathbf{1}_{Ny,Nx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy}$$
$$-\frac{\alpha}{N_x^3}\mathbf{1}_{Ny,Nx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny}-\frac{1}{N_x}\mathbf{K}_{yx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy}+\frac{1}{N_x^2}\mathbf{K}_{yx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny}$$
$$+\frac{1}{N_x^2}\mathbf{K}_{yx}\mathbf{1}_{Nx,Nx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy}-\frac{\alpha}{N_x^3}\mathbf{K}_{yx}\mathbf{1}_{Nx,Nx}\mathbf{1}_{Nx,Ny}+\frac{\alpha}{N_x^2}\mathbf{1}_{Ny,Nx}\mathbf{K}_{xy}$$
$$-\frac{\alpha}{N_x^3}\mathbf{1}_{Ny,Nx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny}-\frac{\alpha}{N_x^3}\mathbf{1}_{Ny,Nx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy}+\frac{\alpha^2}{N_x^4}\mathbf{1}_{Ny,Nx}\mathbf{1}_{Nx,Ny}$$
$$=\mathbf{A}-\frac{1}{N_x}\mathbf{B}+\frac{1}{N_x^2}\mathbf{C}-\frac{\alpha}{N_x^3}\mathbf{D},$$

where

$$\mathbf{A}=\mathbf{K}_{yx}\mathbf{K}_{xy}+\frac{\alpha}{N_x^2}\left(\mathbf{K}_{yx}\mathbf{1}_{Nx,Ny}+\mathbf{1}_{Ny,Nx}\mathbf{K}_{xy}\right)+\frac{\alpha^2}{N_x^3}\mathbf{1}_{Ny,Ny},$$

$$\mathbf{B}=\begin{pmatrix}\mathbf{K}_{yx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny}+\mathbf{K}_{yx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy}\\+\mathbf{1}_{Ny,Nx}\mathbf{K}_{xx}\mathbf{K}_{xy}+\mathbf{K}_{yx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy}\end{pmatrix},$$

$$\mathbf{C}=\begin{pmatrix}\mathbf{1}_{Ny,Nx}\mathbf{K}_{xx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny}+\mathbf{1}_{Ny,Nx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy}\\ +\mathbf{K}_{yx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny}+N_x\mathbf{K}_{yx}\mathbf{1}_{Nx,Nx}\mathbf{K}_{xy}\end{pmatrix},$$

$$\mathbf{D}=\begin{pmatrix}\mathbf{1}_{Ny,Nx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny}+N_x\mathbf{K}_{yx}\mathbf{1}_{Nx,Ny}\\ +\mathbf{1}_{Ny,Nx}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny}+N_x\mathbf{1}_{Ny,Nx}\mathbf{K}_{xy}\end{pmatrix}.$$

APPENDIX II

DKBDA can be accelerated by the incremental technique. Assume in the i^{th} iteration, we have N_{x} positive samples and N_{y} negative samples and in the $(i+1)^{\text{th}}$ iteration, we have L_{x} incremental positive samples and L_{y} incremental negative samples. The deduction is given:

In the i^{th} iteration, the kernel matrix is $\mathbf{K}=\begin{bmatrix}{}^{0}\mathbf{K}_{xx} & {}^{0}\mathbf{K}_{xy}\\ {}^{0}\mathbf{K}_{yx} & {}^{0}\mathbf{K}_{yy}\end{bmatrix}$, where ${}^{0}\mathbf{K}_{yx}={}^{0}\mathbf{K}_{xy}^{T}$, and in the $(i+1)^{\text{th}}$ iteration, the kernel matrix is $\mathbf{K}=\begin{bmatrix}\mathbf{K}_{xx} & \mathbf{K}_{xy}\\ \mathbf{K}_{yx} & \mathbf{K}_{yy}\end{bmatrix}$, where $\mathbf{K}_{yx}=\mathbf{K}_{xy}^{T}$. $\mathbf{K}_{xx}=\begin{bmatrix}{}^{0}\mathbf{K}_{xx} & {}^{1}\mathbf{K}_{xx}\\ {}^{1}\mathbf{K}_{xx}^{T} & {}^{2}\mathbf{K}_{xx}\end{bmatrix}$, $\mathbf{K}_{xy}=\begin{bmatrix}{}^{0}\mathbf{K}_{xy} & {}^{1}\mathbf{K}_{xy}\\ {}^{2}\mathbf{K}_{xy} & {}^{3}\mathbf{K}_{xy}\end{bmatrix}$, and $\mathbf{K}_{yy}=\begin{bmatrix}{}^{0}\mathbf{K}_{yy} & {}^{1}\mathbf{K}_{yy}\\ {}^{1}\mathbf{K}_{yy}^{T} & {}^{2}\mathbf{K}_{yy}\end{bmatrix}$. We denote the elements of these sub-matrices by the kernel function:

$${}^{0}\mathbf{K}_{xx}=\left[k\left(\mathbf{x}_i,\mathbf{x}_j\right)\right]_{\substack{1\le i\le Nx\\1\le j\le Nx}}\qquad {}^{0}\mathbf{K}_{xy}=\left[k\left(\mathbf{x}_i,\mathbf{y}_j\right)\right]_{\substack{1\le i\le Nx\\1\le j\le Ny}}$$

$${}^{0}\mathbf{K}_{yx}={}^{0}\mathbf{K}_{xy}^{T}\qquad {}^{0}\mathbf{K}_{yy}=\left[k\left(\mathbf{y}_i,\mathbf{y}_j\right)\right]_{\substack{1\le i\le Ny\\1\le j\le Ny}}$$

$${}^{1}\mathbf{K}_{xx}=\left[k\left(\mathbf{x}_i,\mathbf{x}_j\right)\right]_{\substack{1\le i\le Nx\\Nx+1\le j\le Nx+Lx}}\qquad {}^{1}\mathbf{K}_{xy}=\left[k\left(\mathbf{x}_i,\mathbf{y}_j\right)\right]_{\substack{1\le i\le Nx\\Ny+1\le j\le Ny+Ly}}$$

$${}^{1}\mathbf{K}_{yx}={}^{1}\mathbf{K}_{xy}^{T}\qquad {}^{1}\mathbf{K}_{yy}=\left[k\left(\mathbf{y}_i,\mathbf{y}_j\right)\right]_{\substack{1\le i\le Ny\\Ny+1\le j\le Ny+Ly}}$$

$${}^{2}\mathbf{K}_{xx}=\left[k\left(\mathbf{x}_i,\mathbf{x}_j\right)\right]_{\substack{Nx+1\le i\le Nx+Lx\\Nx+1\le j\le Nx+Lx}}\qquad {}^{2}\mathbf{K}_{xy}=\left[k\left(\mathbf{x}_i,\mathbf{y}_j\right)\right]_{\substack{Nx+1\le i\le Nx+Lx\\1\le j\le Ny}}$$

$${}^{3}\mathbf{K}_{xy}=\left[k\left(\mathbf{x}_i,\mathbf{y}_j\right)\right]_{\substack{Nx+1\le i\le Nx+Lx\\Ny+1\le j\le Ny+Ly}}\qquad {}^{2}\mathbf{K}_{yy}=\left[k\left(\mathbf{y}_i,\mathbf{y}_j\right)\right]_{\substack{Ny+1\le i\le Ny+Ly\\Ny+1\le j\le Ny+Ly}}$$

The incremental DKBDA depends on the incremental kernel matrix and eigenvalue decomposition.

The incremental version of $\mathbf{\Phi}_y^T\mathbf{\Phi}_y$:

$$\mathbf{\Phi}_y^T\mathbf{\Phi}_y=\begin{pmatrix}\mathbf{K}_{yy}-\frac{1}{N_x+L_x}\mathbf{K}_{yx}\mathbf{1}_{Nx+Lx,Ny+Ly}\\ -\frac{1}{N_x+L_x}\mathbf{1}_{Ny+Ly,Nx+Lx}\mathbf{K}_{xy}+\frac{\alpha}{\left(N_x+L_x\right)^2}\mathbf{1}_{Ny+Ly,Ny+Ly}\end{pmatrix}$$

$$=\begin{bmatrix}{}^{0}\mathbf{K}_{yy} & {}^{1}\mathbf{K}_{yy}\\ {}^{1}\mathbf{K}_{yy}^{T} & {}^{2}\mathbf{K}_{yy}\end{bmatrix}+\frac{\alpha}{\left(N_x+L_x\right)^2}\begin{bmatrix}\mathbf{1}_{Ny,Ny} & \mathbf{1}_{Ny,Ly}\\ \mathbf{1}_{Ly,Ny} & \mathbf{1}_{Ly,Ly}\end{bmatrix}$$
$$-\frac{1}{N_x+L_x}\begin{bmatrix}{}^{0}\mathbf{K}_{xy}^{T} & {}^{2}\mathbf{K}_{xy}^{T}\\ {}^{1}\mathbf{K}_{xy}^{T} & {}^{3}\mathbf{K}_{xy}^{T}\end{bmatrix}\begin{bmatrix}\mathbf{1}_{Nx,Ny} & \mathbf{1}_{Nx,Ly}\\ \mathbf{1}_{Lx,Ny} & \mathbf{1}_{Lx,Ly}\end{bmatrix}$$
$$-\frac{1}{N_x+L_x}\begin{bmatrix}\mathbf{1}_{Ny,Nx} & \mathbf{1}_{Ny,Lx}\\ \mathbf{1}_{Ly,Nx} & \mathbf{1}_{Ly,Lx}\end{bmatrix}\begin{bmatrix}{}^{0}\mathbf{K}_{xy} & {}^{1}\mathbf{K}_{xy}\\ {}^{2}\mathbf{K}_{xy} & {}^{3}\mathbf{K}_{xy}\end{bmatrix}$$

$$=\begin{bmatrix}{}^{0}\mathbf{K}_{yy} & {}^{1}\mathbf{K}_{yy}\\ {}^{1}\mathbf{K}_{yy}^{T} & {}^{2}\mathbf{K}_{yy}\end{bmatrix}+\frac{\alpha}{\left(N_x+L_x\right)^2}\begin{bmatrix}\mathbf{1}_{Ny,Ny} & \mathbf{1}_{Ny,Ly}\\ \mathbf{1}_{Ly,Ny} & \mathbf{1}_{Ly,Ly}\end{bmatrix}$$
$$-\frac{1}{N_x+L_x}\begin{bmatrix}{}^{0}\mathbf{K}_{xy}^{T}\mathbf{1}_{Nx,Ny}+{}^{2}\mathbf{K}_{xy}^{T}\mathbf{1}_{Lx,Ny} & {}^{0}\mathbf{K}_{xy}^{T}\mathbf{1}_{Nx,Ly}+{}^{2}\mathbf{K}_{xy}^{T}\mathbf{1}_{Lx,Ly}\\ {}^{1}\mathbf{K}_{xy}^{T}\mathbf{1}_{Nx,Ny}+{}^{3}\mathbf{K}_{xy}^{T}\mathbf{1}_{Lx,Ny} & {}^{1}\mathbf{K}_{xy}^{T}\mathbf{1}_{Nx,Ly}+{}^{3}\mathbf{K}_{xy}^{T}\mathbf{1}_{Lx,Ly}\end{bmatrix}$$
$$-\frac{1}{N_x+L_x}\begin{bmatrix}\mathbf{1}_{Ny,Nx}{}^{0}\mathbf{K}_{xy}+\mathbf{1}_{Ny,Lx}{}^{2}\mathbf{K}_{xy} & \mathbf{1}_{Ny,Nx}{}^{1}\mathbf{K}_{xy}+\mathbf{1}_{Ny,Lx}{}^{3}\mathbf{K}_{xy}\\ \mathbf{1}_{Ly,Nx}{}^{0}\mathbf{K}_{xy}+\mathbf{1}_{Ly,Lx}{}^{2}\mathbf{K}_{xy} & \mathbf{1}_{Ly,Nx}{}^{1}\mathbf{K}_{xy}+\mathbf{1}_{Ly,Lx}{}^{3}\mathbf{K}_{xy}\end{bmatrix}$$
$$=\begin{bmatrix}\mathbf{\Psi}_{11} & \mathbf{\Psi}_{12}\\ \mathbf{\Psi}_{21} & \mathbf{\Psi}_{22}\end{bmatrix}=\begin{bmatrix}\mathbf{\Psi}_{11} & \mathbf{\Psi}_{12}\\ \mathbf{\Psi}_{12}^{T} & \mathbf{\Psi}_{22}\end{bmatrix}\quad\begin{bmatrix}\mathbf{\Phi}_y^T\mathbf{\Phi}_y & \mathbf{\Psi}_{12}\\ \mathbf{\Psi}_{12}^{T} & \mathbf{\Psi}_{22}\end{bmatrix}$$

$$\begin{cases}\mathbf{\Psi}_{11}={}^{0}\mathbf{K}_{yy}-\frac{1}{N_x+L_x}\begin{pmatrix}{}^{0}\mathbf{K}_{xy}^{T}\mathbf{1}_{Nx,Ny}+{}^{2}\mathbf{K}_{xy}^{T}\mathbf{1}_{Lx,Ny}\\ +\mathbf{1}_{Ny,Nx}{}^{0}\mathbf{K}_{xy}+\mathbf{1}_{Ny,Lx}{}^{2}\mathbf{K}_{xy}\end{pmatrix}+\frac{\alpha}{\left(N_x+L_x\right)^2}\mathbf{1}_{Ny,Ny}\\ \mathbf{\Psi}_{12}={}^{1}\mathbf{K}_{yy}-\frac{1}{N_x+L_x}\begin{pmatrix}{}^{0}\mathbf{K}_{xy}^{T}\mathbf{1}_{Nx,Ly}+{}^{2}\mathbf{K}_{xy}^{T}\mathbf{1}_{Lx,Ly}\\ +\mathbf{1}_{Ny,Nx}{}^{1}\mathbf{K}_{xy}+\mathbf{1}_{Ny,Lx}{}^{3}\mathbf{K}_{xy}\end{pmatrix}+\frac{\alpha}{\left(N_x+L_x\right)^2}\mathbf{1}_{Ny,Ly}\\ \mathbf{\Psi}_{22}={}^{2}\mathbf{K}_{yy}-\frac{1}{N_x+L_x}\begin{pmatrix}{}^{1}\mathbf{K}_{xy}^{T}\mathbf{1}_{Nx,Ly}+{}^{3}\mathbf{K}_{xy}^{T}\mathbf{1}_{Lx,Ly}\\ +\mathbf{1}_{Ly,Nx}{}^{1}\mathbf{K}_{xy}+\mathbf{1}_{Ly,Lx}{}^{3}\mathbf{K}_{xy}\end{pmatrix}+\frac{\alpha}{\left(N_x+L_x\right)^2}\mathbf{1}_{Ly,Ly}\end{cases}$$

$$\alpha=\mathbf{1}_{Nx+Lx,1}^{T}\mathbf{K}_{xx}\mathbf{1}_{Nx+Lx,1}=\begin{bmatrix}\mathbf{1}_{1,Nx} & \mathbf{1}_{1,Lx}\end{bmatrix}\begin{bmatrix}{}^{0}\mathbf{K}_{xx} & {}^{1}\mathbf{K}_{xx}\\ {}^{1}\mathbf{K}_{xx}^{T} & {}^{2}\mathbf{K}_{xx}\end{bmatrix}\begin{bmatrix}\mathbf{1}_{Nx,1}\\ \mathbf{1}_{Lx,1}\end{bmatrix}$$
$$=\mathbf{1}_{1,Nx}{}^{0}\mathbf{K}_{xx}\mathbf{1}_{Nx,1}+\mathbf{1}_{1,Lx}{}^{1}\mathbf{K}_{xx}^{T}\mathbf{1}_{Nx,1}+\mathbf{1}_{1,Nx}{}^{1}\mathbf{K}_{xx}\mathbf{1}_{Nx,1}+\mathbf{1}_{1,Lx}{}^{2}\mathbf{K}_{xx}\mathbf{1}_{Lx,1}$$
$$=\alpha+\left(\mathbf{1}_{1,Lx}{}^{1}\mathbf{K}_{xx}^{T}\mathbf{1}_{Nx,1}+\mathbf{1}_{1,Nx}{}^{1}\mathbf{K}_{xx}\mathbf{1}_{Nx,1}+\mathbf{1}_{1,Lx}{}^{2}\mathbf{K}_{xx}\mathbf{1}_{Lx,1}\right)$$

$\left\|\mathbf{\Psi}_{11}-\mathbf{\Phi}_y^T\mathbf{\Phi}_y\right\|\le\varepsilon$, where ε is a small value. $\alpha=\mathbf{1}_{Nx,1}^{T}\mathbf{K}_{xx}\mathbf{1}_{Nx,1}$. $\mathbf{1}_{Nx,Ny}$ is an Nx by Ny sized matrix with all terms equal to 1, so are $\mathbf{1}_{Ny,Nx}$, $\mathbf{1}_{Nx,Nx}$ and $\mathbf{1}_{Ny,Ny}$.

The incremental version of $\mathbf{\Phi}_x^T\mathbf{\Phi}_y$:

$$\mathbf{\Phi}_x^T\mathbf{\Phi}_y=\begin{pmatrix}\mathbf{K}_{xy}-\frac{1}{N_x+L_x}\mathbf{K}_{xx}\mathbf{1}_{Nx+Lx,Ny+Ly}\\ -\frac{1}{N_x+L_x}\mathbf{1}_{Nx+Lx,Nx+Lx}\mathbf{K}_{xy}+\frac{\alpha}{\left(N_x+L_x\right)^2}\mathbf{1}_{Nx+Lx,Ny+Ly}\end{pmatrix}$$

$$=\begin{bmatrix}{}^{0}\mathbf{K}_{xy} & {}^{1}\mathbf{K}_{xy}\\ {}^{2}\mathbf{K}_{xy} & {}^{3}\mathbf{K}_{xy}\end{bmatrix}+\frac{\alpha}{\left(N_x+L_x\right)^2}\begin{bmatrix}\mathbf{1}_{Nx,Ny} & \mathbf{1}_{Nx,Ly}\\ \mathbf{1}_{Lx,Ny} & \mathbf{1}_{Lx,Ly}\end{bmatrix}$$
$$-\frac{1}{N_x+L_x}\begin{bmatrix}{}^{0}\mathbf{K}_{xx} & {}^{1}\mathbf{K}_{xx}\\ {}^{1}\mathbf{K}_{xx}^{T} & {}^{2}\mathbf{K}_{xx}\end{bmatrix}\begin{bmatrix}\mathbf{1}_{Nx,Ny} & \mathbf{1}_{Nx,Ly}\\ \mathbf{1}_{Lx,Ny} & \mathbf{1}_{Lx,Ly}\end{bmatrix}$$
$$-\frac{1}{N_x+L_x}\begin{bmatrix}\mathbf{1}_{Nx,Nx} & \mathbf{1}_{Nx,Lx}\\ \mathbf{1}_{Lx,Nx} & \mathbf{1}_{Lx,Lx}\end{bmatrix}\begin{bmatrix}{}^{0}\mathbf{K}_{xy} & {}^{1}\mathbf{K}_{xy}\\ {}^{2}\mathbf{K}_{xy} & {}^{3}\mathbf{K}_{xy}\end{bmatrix}$$

$$=\begin{bmatrix}{}^{0}\mathbf{K}_{xy} & {}^{1}\mathbf{K}_{xy}\\ {}^{2}\mathbf{K}_{xy} & {}^{3}\mathbf{K}_{xy}\end{bmatrix}+\frac{\alpha}{\left(N_x+L_x\right)^2}\begin{bmatrix}\mathbf{1}_{Nx,Ny} & \mathbf{1}_{Nx,Ly}\\ \mathbf{1}_{Lx,Ny} & \mathbf{1}_{Lx,Ly}\end{bmatrix}$$
$$-\frac{1}{N_x+L_x}\begin{bmatrix}{}^{0}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny}+{}^{1}\mathbf{K}_{xx}\mathbf{1}_{Lx,Ny} & {}^{0}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ly}+{}^{1}\mathbf{K}_{xx}\mathbf{1}_{Lx,Ly}\\ {}^{1}\mathbf{K}_{xx}^{T}\mathbf{1}_{Nx,Ny}+{}^{2}\mathbf{K}_{xx}\mathbf{1}_{Lx,Ny} & {}^{1}\mathbf{K}_{xx}^{T}\mathbf{1}_{Nx,Ly}+{}^{2}\mathbf{K}_{xx}\mathbf{1}_{Lx,Ly}\end{bmatrix}$$
$$-\frac{1}{N_x+L_x}\begin{bmatrix}\mathbf{1}_{Nx,Nx}{}^{0}\mathbf{K}_{xy}+\mathbf{1}_{Nx,Lx}{}^{2}\mathbf{K}_{xy} & \mathbf{1}_{Nx,Nx}{}^{1}\mathbf{K}_{xy}+\mathbf{1}_{Nx,Lx}{}^{3}\mathbf{K}_{xy}\\ \mathbf{1}_{Lx,Nx}{}^{0}\mathbf{K}_{xy}+\mathbf{1}_{Lx,Lx}{}^{2}\mathbf{K}_{xy} & \mathbf{1}_{Lx,Nx}{}^{1}\mathbf{K}_{xy}+\mathbf{1}_{Lx,Lx}{}^{3}\mathbf{K}_{xy}\end{bmatrix}$$
$$=\begin{bmatrix}\mathbf{\Pi}_{11} & \mathbf{\Pi}_{12}\\ \mathbf{\Pi}_{21} & \mathbf{\Pi}_{22}\end{bmatrix}\quad\begin{bmatrix}\mathbf{\Phi}_x^T\mathbf{\Phi}_y & \mathbf{\Pi}_{12}\\ \mathbf{\Pi}_{21} & \mathbf{\Pi}_{22}\end{bmatrix}$$

$$\begin{cases} \mathbf{\Pi}_{11} = {}^{0}\mathbf{K}_{xy} - \dfrac{1}{N_x + L_x}\begin{pmatrix} {}^{0}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ny} + {}^{1}\mathbf{K}_{xx}\mathbf{1}_{Lx,Ny} \\ +\mathbf{1}_{Nx,Nx}\,{}^{0}\mathbf{K}_{xy} + \mathbf{1}_{Nx,Lx}\,{}^{2}\mathbf{K}_{xy} \end{pmatrix} + \dfrac{\alpha}{\left(N_x + L_x\right)^2}\mathbf{1}_{Nx,Ny} \\ \mathbf{\Pi}_{12} = {}^{1}\mathbf{K}_{xy} - \dfrac{1}{N_x + L_x}\begin{pmatrix} {}^{0}\mathbf{K}_{xx}\mathbf{1}_{Nx,Ly} + {}^{1}\mathbf{K}_{xx}\mathbf{1}_{Lx,Ly} \\ +\mathbf{1}_{Nx,Nx}\,{}^{1}\mathbf{K}_{xy} + \mathbf{1}_{Nx,Lx}\,{}^{3}\mathbf{K}_{xy} \end{pmatrix} + \dfrac{\alpha}{\left(N_x + L_x\right)^2}\mathbf{1}_{Nx,Ly} \\ \mathbf{\Pi}_{21} = {}^{2}\mathbf{K}_{xy} - \dfrac{1}{N_x + L_x}\begin{pmatrix} {}^{1}\mathbf{K}_{xx}^{T}\mathbf{1}_{Nx,Ny} + {}^{2}\mathbf{K}_{xx}\mathbf{1}_{Lx,Ny} \\ +\mathbf{1}_{Lx,Nx}\,{}^{0}\mathbf{K}_{xy} + \mathbf{1}_{Lx,Lx}\,{}^{2}\mathbf{K}_{xy} \end{pmatrix} + \dfrac{\alpha}{\left(N_x + L_x\right)^2}\mathbf{1}_{Nx,Ly} \\ \mathbf{\Pi}_{22} = {}^{3}\mathbf{K}_{xy} - - \dfrac{1}{N_x + L_x}\begin{pmatrix} {}^{1}\mathbf{K}_{xx}^{T}\mathbf{1}_{Nx,Ly} + {}^{2}\mathbf{K}_{xx}\mathbf{1}_{Lx,Ly} \\ +\mathbf{1}_{Lx,Nx}\,{}^{1}\mathbf{K}_{xy} + \mathbf{1}_{Lx,Lx}\,{}^{3}\mathbf{K}_{xy} \end{pmatrix} + \dfrac{\alpha}{\left(N_x + L_x\right)^2}\mathbf{1}_{Lx,Ly} \end{cases}$$

$\left\|\mathbf{\Pi}_{11} - \mathbf{\Phi}_x^T\mathbf{\Phi}_y\right\| \le \xi$, where ξ is a small value.

Then we have the incremental version of $\mathbf{\Phi}_y^T\tilde{\mathbf{S}}_x^{\phi}\mathbf{\Phi}_y$ as:

$$\mathbf{\Phi}_y^T\tilde{\mathbf{S}}_x^{\phi}\mathbf{\Phi}_y = \left(\mathbf{\Phi}_x^T\mathbf{\Phi}_y\right)^T\left(\mathbf{\Phi}_x^T\mathbf{\Phi}_y\right) = \begin{bmatrix} \mathbf{\Phi}_x^T\mathbf{\Phi}_y & \mathbf{\Pi}_{12} \\ \mathbf{\Pi}_{21} & \mathbf{\Pi}_{22} \end{bmatrix}^T \begin{bmatrix} \mathbf{\Phi}_x^T\mathbf{\Phi}_y & \mathbf{\Pi}_{12} \\ \mathbf{\Pi}_{21} & \mathbf{\Pi}_{22} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{\Phi}_y^T\mathbf{\Phi}_x\mathbf{\Phi}_x^T\mathbf{\Phi}_y + \mathbf{\Pi}_{12}^T\mathbf{\Pi}_{21} & \mathbf{\Phi}_y^T\mathbf{\Phi}_x\mathbf{\Pi}_{12} + \mathbf{\Pi}_{12}^T\mathbf{\Pi}_{22} \\ \mathbf{\Pi}_{21}^T\mathbf{\Phi}_x^T\mathbf{\Phi}_y + \mathbf{\Pi}_{22}^T\mathbf{\Pi}_{21} & \mathbf{\Pi}_{21}^T\mathbf{\Pi}_{12} + \mathbf{\Pi}_{22}^T\mathbf{\Pi}_{22} \end{bmatrix}.$$

To conduct the incremental learning, we also need the incremental singular value decomposition (SVD).

Theorem 1 ([35]**):**

If $\mathbf{USV}^T = \mathbf{M}$, then $\begin{bmatrix}\mathbf{U} & \mathbf{J}\end{bmatrix}\begin{bmatrix}\mathbf{S} & \mathbf{U}^T\mathbf{C} \\ 0 & \mathbf{J}^T\mathbf{H}\end{bmatrix}\begin{bmatrix}\mathbf{V} & 0 \\ 0 & \mathbf{I}\end{bmatrix}^T$

$= \begin{bmatrix}\mathbf{M} & \mathbf{C}\end{bmatrix}$, where $\mathbf{H} = \left(\mathbf{I} - \mathbf{UU}^T\right)\mathbf{C} \xrightarrow{\text{QR Decomposition}} \mathbf{JK}$.

Let $\mathbf{Q} = \begin{bmatrix}\mathbf{S} & \mathbf{U}^T\mathbf{C} \\ 0 & \mathbf{J}^T\mathbf{H}\end{bmatrix} \xrightarrow{\text{SVD}} \mathbf{U'S'V'}$, then we get the incremental version of SVD as:

$$\left(\begin{bmatrix}\mathbf{U} & \mathbf{J}\end{bmatrix}\mathbf{U'}\right)\mathbf{S'}\left(\begin{bmatrix}\mathbf{V} & 0 \\ 0 & \mathbf{I}\end{bmatrix}\mathbf{V'}\right)^T = \begin{bmatrix}\mathbf{M} & \mathbf{C}\end{bmatrix}.$$

Theorem 2:

Given $\mathbf{USV}^T = \mathbf{M}$, then $\begin{bmatrix}\mathbf{M} & \mathbf{C} \\ & \mathbf{T}\end{bmatrix}$ can be decomposed by:

$$\begin{bmatrix}\mathbf{M} & \mathbf{C} \\ & \mathbf{T}\end{bmatrix} = \begin{bmatrix}\left(\begin{bmatrix}\mathbf{U} & \mathbf{J}\end{bmatrix}\mathbf{U'}\right)^T & 0 \\ 0 & \mathbf{I}\end{bmatrix}$$

$$\begin{bmatrix} \mathbf{S'} & 0 \\ \mathbf{T}\left(\begin{bmatrix}\mathbf{V} & 0 \\ 0 & \mathbf{I}\end{bmatrix}\mathbf{V'}\right) & \mathbf{H'}^T\mathbf{J'} \end{bmatrix}\begin{bmatrix}\left(\begin{bmatrix}\mathbf{V} & 0 \\ 0 & \mathbf{I}\end{bmatrix}\mathbf{V'}\right) & \mathbf{J'}\end{bmatrix}^T$$

where

$$\mathbf{H'} = \left(\mathbf{I} - \begin{bmatrix}\mathbf{V} & 0 \\ 0 & \mathbf{I}\end{bmatrix}\mathbf{V'V'}^T\begin{bmatrix}\mathbf{V}^T & 0 \\ 0 & \mathbf{I}\end{bmatrix}\right)\mathbf{T}^T \xrightarrow{\text{QR Decomposition}} \mathbf{J'K'}$$

.Let $\mathbf{Q'} = \begin{bmatrix} \mathbf{S'} & 0 \\ \mathbf{T}\left(\begin{bmatrix}\mathbf{V} & 0 \\ 0 & \mathbf{I}\end{bmatrix}\mathbf{V'}\right) & \mathbf{H'}^T\mathbf{J'} \end{bmatrix} \xrightarrow{SVD} \mathbf{U''S''V''}$ and the incremental SVD of the matrix $\begin{bmatrix}\mathbf{M} & \mathbf{C} \\ & \mathbf{T}\end{bmatrix}$ is:

$$\begin{bmatrix}\mathbf{M} & \mathbf{C} \\ & \mathbf{T}\end{bmatrix} = \left(\begin{bmatrix}\left(\begin{bmatrix}\mathbf{U} & \mathbf{J}\end{bmatrix}\mathbf{U'}\right)^T & 0 \\ 0 & \mathbf{I}\end{bmatrix}\mathbf{U''}\right)\mathbf{S''}\left(\begin{bmatrix}\left(\begin{bmatrix}\mathbf{V} & 0 \\ 0 & \mathbf{I}\end{bmatrix}\mathbf{V'}\right) & \mathbf{J'}\end{bmatrix}\mathbf{V''}\right)^T$$

.Theorem 2 can be obtained from theorem1 easily.

The incremental DKBDA can be obtained with the theorem 2. Two conditions can guarantee its rightness:

1. $\left\|\mathbf{\Psi}_{11} - \mathbf{\Phi}_y^T\mathbf{\Phi}_y\right\| \le \varepsilon$, where ε is a small value.
2. $\left\|\mathbf{\Pi}_{11} - \mathbf{\Phi}_x^T\mathbf{\Phi}_y\right\| \le \xi$, where ξ is a small value.

With the two conditions (for the incremental SVD) and the incremental computation of the kernel matrix $\mathbf{K}$, $\mathbf{\Phi}_y^T\mathbf{\Phi}_y$, $\mathbf{\Phi}_x^T\mathbf{\Phi}_y$, and $\mathbf{\Phi}_y^T\mathbf{S}_x^{\phi}\mathbf{\Phi}_y$, the incremental version DKBDA can be obtained as in Table 2.

ACKNOWLEDGMENT

We thank Professor Peter King (Sch. of Computer Science & Information Systems, Birkbeck College, University of London, UK.) for helpful discussions and assistance with the presentation of this material. We thank Miss Jing Li for assistance on the incremental learning.

REFERENCES

[1] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-based Image Retrieval at the End of the Early Years,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 22(12), pp. 1349 - 1380, Dec. 2000.

[2] W. Ma and H. Zhang, “Content-based Image Indexing and Retrieval,” in Handbook of Multimedia Computing, Borko Furht, ed. CRC Press, pp. 227 - 254, 1998.

[3] M. Swain and D. Ballard, “Color Indexing,” International Journal of Computer Vision, 7(1), pp. 11 - 32, 1991.

[4] G. Pass, R. Zabih, and J. Miller, “Comparing Images Using Color Coherence Vectors,” In Proc. ACM Multimedia, pp. 65 - 73, 1996.

[5] H. Tamura, S. Mori, and T. Yamawaki, “Texture Features Corresponding to Visual Perception,” IEEE Trans. on Systems, Man, and Cybernetics, 8(6), pp. 460-473, June 1978.

[6] B. Manjunath and W. Ma, “Texture Features for Browsing and Retrieval of Image Data,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 18(8), pp. 837 - 42, Aug. 1996.

[7] J. Mao and A. Jain, “Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models,” Pattern Recognition, 25(2), pp.173 - 188, 1992.

[8] T. Chang and C. Kuo, “Texture Analysis and Classification with Tree-Structured Wavelet Transform,” IEEE Trans. on Image Processing, 2(4), pp. 429 - 441, Oct. 1993.

[9] J. Wang, J. Li, and G. Wiederhold, “SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 23(9), pp. 947 – 963, Sep. 2001.

[10] B. Manjunath, J. Ohm, V. Vasudevan, and A. Yamada, “Color and Texture Descriptors Circuits and Systems for Video Technology,” IEEE Transactions on Circuits and Systems for Video Technology, 11(6), pp. 703 – 715, June 2001.

[11] A. Jain and A. Vailaya, “Image Retrieval Using Color and Shape,” Pattern Recognition, 29(8), pp. 1233 - 1244, Aug. 1996.

[12] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubino, “the QBIC Project: Querying Images by Content Using Color, Texture, and Shape,” Storage and Retrieval for Images and Video Databases, pp. 173-181, SPIE, Feb. 1993.

[13] A. Jain and A. Vailaya, “Shape-based Retrieval: a Case Study with Trademark Image Databases,” Pattern Recognition, 31(9), pp. 1369 - 1390, 1998.

[14] Y. Ishikawa, R. Subramanya, and C. Faloutsos, “Mindreader: Querying Databases through Multiple Examples,” In Proc. International Conference on Very Large Data Base, pp. 433-438, 1998.
[15] Y. Rui, T. Huang, M. Ortega, and S. Mehrotra, “Relevance Feedback: a Power Tool in Interactive Content-based Image Retrieval,” IEEE Trans. on Circuits and Systems for Video Technology, 8(5), pp. 644 – 655, Sep. 1998.
[16] Y. Rui, T. Huang, and S. Mehrotra, “Relevance Feedback Techniques in Interactive Content-based Image Retrieval,” In Proc. Storage and Retrieval of Images and Video Databases VI, vol. 3312, pp. 25-36, SPIE, 1998.
[17] Y. Rui, T. Huang, and S. Chang, “Image Retrieval: Current Techniques, Promising Directions and Open Issues,” Journal of Visual Communication and Image Representation, 10(1) pp.1 - 23, Mar. 1999.
[18] K. Fukunaga, “Statistical Pattern Recognition (Second Edition),” Academic Press, Boston, 1990.
[19] Y. Cheng, Y. Zhuang, and J. Yang, “Optimal Fisher Discriminant Analysis Using the Rank Decomposition,” Pattern Recognition, vol. 25, pp. 101 - 111, 1992.
[20] K. Liu, Y. Cheng, J, Yang, and X. Liu, “an Efficient Algorithm for Foley-Sammon Optimal Set of Discriminant Vectors by Algebraic Method,” Pattern Recognition, vol. 6, pp. 817 - 829, 1992.
[21] D. Swets, and J. Weng, “Discriminant Analysis and Eigenspace Partition Tree for Face and Object Recognition from Views,” In Proc. IEEE International Conference on Automatic Face and Gesture Recognition, pp.192 - 197, Oct. 1996.
[22] W. Zhao, R. Chellappa, and A. Krishnaswamy, “Discriminant Analysis of Principal Components for Face Recognition,” In Proc. IEEE International Conference on Automatic Face and Gesture Recognition, pp. 336 - 341, 1998.
[23] H. Yu and J. Yang, “a Direct LDA Algorithm for High-Dimensional Data with Application to Face Recognition,” Pattern Recognition, vol. 34, pp 2067-2070, 2001.
[24] X. Zhou, T. Huang, “Small Sample Learning during Multimedia Retrieval using Biasmap,” In Proc. IEEE International Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 11-17, 2001.
[25] X. Zhou and T. Huang, “Comparing Discriminate Transformations and SVM for Learning during Multimedia Retrieval,” In Proc. ACM Multimedia, pp. 137-146, 2001.
[26] J. Lu, K. Plataniotis, and A. Venetsanopoulos, “Face Recognition Using Kernel Direct Discriminant Analysis Algorithms,” IEEE Trans. on Neural Networks, 14(1), pp. 117 - 126, Jan. 2003.
[27] V. Vapnik, “the Nature of Statistical Learning Theory,” Springer-Verlag, New York (1995).
[28] L. Zhang, F. Lin, and B. Zhang, “Support Vector Machine Learning for Image Retrieval,” In Proc. IEEE International Conference on Image Processing, pp. 721-724, 2001.
[29] P. Hong, Q. Tian, T. Huang, “Incorporate Support Vector Machines to Content-based Image Retrieval with Relevant Feedback,” In Proc. IEEE International Conference on Image Processing, pp. 750-753, 2000.
[30] Y. Chen, X. Zhou, and T. Huang, “One-class SVM for Learning in Image Retrieval,” In Proc. IEEE International Conference on Image Processing, pp. 815 - 818, 2001.
[31] G. Guo, A. Jain, W. Ma, and H. Zhang, “Learning Similarity Measure for Natural Image Retrieval with Relevance Feedback,” IEEE Trans. on Neural Networks, 12(4), pp. 811 - 820, July 2002.
[32] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, “An Introduction to Kernel-based Learning Algorithms,” IEEE. Trans on Neural Networks, 12(2), pp. 181 - 201, Mar. 2001.
[33] A. Roger and C. Johnson, “Matrix analysis,” Cambridge University Press, 1990.
[34] D. Tao and X. Tang, “A Direct Method to Solve the Biased Discriminant Analysis in Kernel Feature Space for Content-based Image Retrieval,” IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 441-444, 2004.
[35] N. Vasconcelos, “Image Indexing with Mixture Hierarchies,” In Proc. IEEE International Conference on Computer Vision and Pattern Recognition, pp. 3 – 10, 2001.
[36] M. Brand, “Incremental Singular Value Decomposition of Uncertain Data with Missing Values,” In Proc. European Conference on Computer Vision, pp. 707 – 720, 2002.