Face Recognition using Second Order Discriminant Tensor Subspace Analysis

Su-Jing Wang[a], Chun-Guang Zhou[a], Na Zhang[a], Xu-Jun Peng[b], Yu-Hsin Chen[a], Xiaohua Liu[a,*]

[a] *College of Computer Science and Technology, Jilin University, Changchun 130012, China*
[b] *CSE department, SUNY at Buffalo, Amherst, NY 14228, USA*

Abstract

Discriminant information (DI) plays a critical role in face recognition. In this paper, we proposed a Second Order Discriminant Tensor Subspace Analysis (DTSA) algorithm to extract discriminant features from the intrinsic manifold structure of the tensor data. DTSA combines the advantages of previous methods with DI, the tensor methods preserving the spatial structure information of the samples original image matrices, and the manifold methods preserving the local structure of the samples distribution. DTSA defines two similarity matrices, namely within-class similarity matrix and between-class similarity matrix. The within-class similarity matrix is determined by the distances of point pairs in the same class, while the between-class similarity matrix is determined by the distances between the means of each pair of classes. Using these two matrices, the proposed method preserves the local structure of the samples to fit the manifold structure of facial images in high dimensional space better than other methods. Moreover, compared to the 2D methods, the tensor based method employs two-sided transformations rather than single-sided one, and yields higher compression ratio. As a tensor method, DTSA uses an iterative procedure to calculate the optimal solution of two transformation matrices. In this paper, we analyzed DTSA's connections to 2D-DLPP and TSA, theoretically. The experiments on the ORL, Yale and YaleB facial databases show the effectiveness of the proposed method.

[*] Corresponding author
Email address: liuxiaohua@yahoo.cn (Xiaohua Liu)

Keywords: Face recognition, Locality preserving projection, Discriminant information, Tensor subspace

1. Introduction

Automatic facial recognition has been a longstanding challenge in the field of computer vision and pattern recognition for several decades. Feature extraction is one of the central issues for face recognition. Subspace transformation (ST) is often used as a feature extraction method. The idea of ST is to project the feature from the original high dimensional space to a low dimensional subspace, which is called *projective subspace*. In the projective subspace, the transformed feature is easier to be distinguished than the original one.

As one of the widely used linear STs, Principal Component Analysis (PCA)[1] seeks the optimal projection directions according to maximal variances. Linear Discriminant Analysis (LDA)[2] uses DI to search for the directions which are most effective for discrimination by maximizing the ratio between the between-class and within-class scatters. Both PCA and LDA aim to preserve global structures of the samples. Seung[3] assumed that the high dimensional visual image information in real world lies on or is close to a smooth low dimensional manifold. Inspired by this assumption, multiple manifold STs that preserve local structure of samples have been proposed. Locality Preserving Projections (LPP)[4] aims to preserve the local structure of the original space in the projective subspace. Its performance is better than those of PCA and LDA for face recognition[5]. Discriminant Locality Preserving Projections (DLPP)[6] encodes DI into LPP to further improve the discriminant performance of LPP for face recognition. Some other DLPP related works can be found in [7][8][9][10].

The potential shortage of the above methods is that they are one dimensional STs (1D-STs), which vectorize a facial image of size m by n to a $(m \times n)$-dimensional vector. For example, the corresponding vector of a 32×32 pixels image is 1024 dimension. In practice, the 1D-STs applied on the 2D images have been found to have some intrinsic problems: singularity of within-class scatter matrices, limited available projection directions, high computational cost and a loss of the underlying spatial structure information of the images.

To overcome the above problems, some researchers have attempted to treat the image as a matrix instead of a vector. Yang *et al.*[11] proposed a 2D-PCA algorithm to compute the image scatter matrix from the image matrix representations directly. Li and Yuan[12] presented a 2D-LDA to extend LDA using the idea of the image matrix representations. Chen *et al.*[13] developed a 2D-LPP which directly extracts the proper features from image matrix representations by preserving the local structure of samples. Xu *et al.*[14] used DI to construct the adjacency graph based on 2D-LPP. And Yu developed [15] a 2D-DLPP, a variation of 2D-LPP which uses DI. 2D-LPP and 2D-DLPP achieved better results in recognizing face, facial expression[16], gait[17], and palm[18] than the methods preserving the global structure of samples such as 2D-PCA, 2D-LDA. These 2D-STs (two dimensional ST) not only reduce the complexities of time and space, but also preserve spatial structure information of the 2D images.

However, one disadvantage of 2D-STs (compared to 1D-STs) is that there are still more feature coefficients needed to represent an image, due to the fact that 2D-STs only employ single-sided transformations. Recently, the tensor STs, which employ two-sided transformation for a gray image, attract more attention in the field of feature extraction and dimension reduction, since many objects can be represented by multidimensional arrays, *i.e.* tensors. The number of dimensions is called the *order* of the tensor and each dimension defines one of the so-called *modes*. For example, a gray image is a second-order tensor, then its rows are called mode-1 of the tensor and its columns are called mode-2 of the tensor.

For 2nd-order tensor, Ye[19][20] proposed the generalized low-rank approximations of matrices (GLRAM) method. GLRAM was reported to consume less computation time and yield higher compression ratio than SVD in applications such as image compression and retrieval. He *et al.*[21] proposed a algorithm, Tensor Subspace Analysis (TSA), which preserves the local structure of samples using two-sided transformations. For Nth-order tensor, Liu *et al.*[22] extended PCA from vector to tensor. In order to encode the DI into the tensor subspace, GTDA[23] and DATER[24] extended LDA and MSD[25] from vector to tensor, respectively. These algorithms are summarized in Figure 1.

Most of the above existing algorithms are unified into a general graph embedding framework proposed by Yan *et al.* [26]. And a new supervised ST algorithm MFA (Marginal Fisher Analysis) was proposed by them under this framework as well. Recently, some new ST algorithms[27][28][29][15] were also developed by using the framework. In this paper, we follow the framework provided by Yan and propose a new ST algorithm.

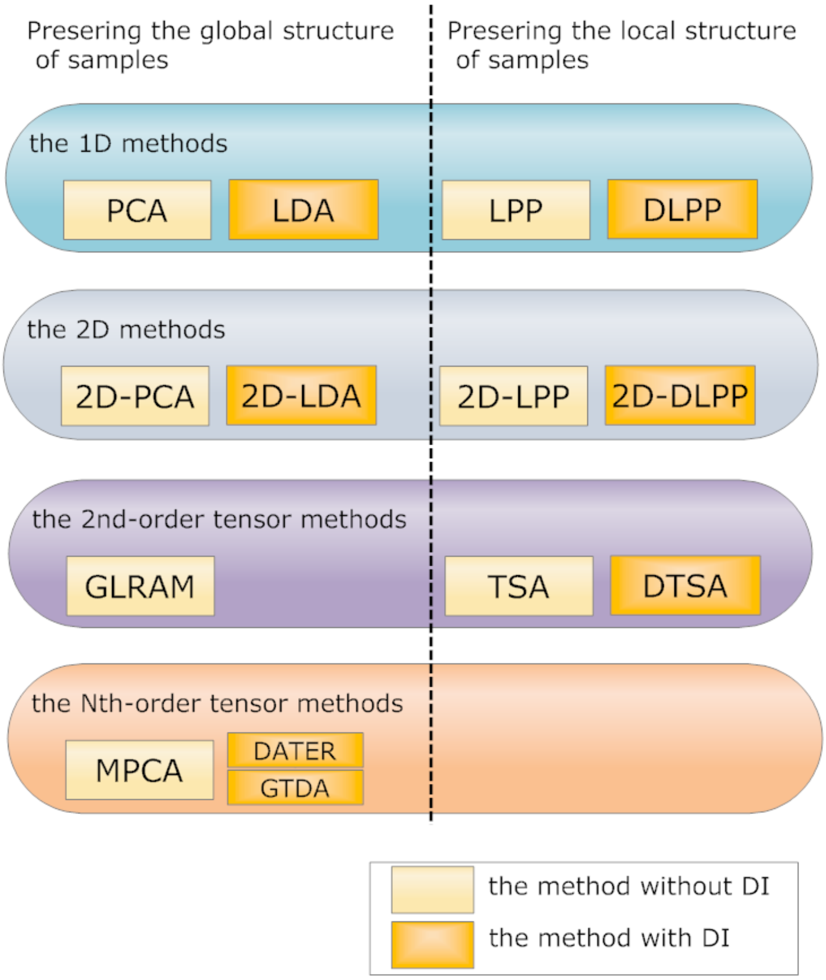

Presering the global structure of samples
Presering the local structure of samples
the 1D methods
PCA
LDA
LPP
DLPP
the 2D methods
2D-PCA
2D-LDA
2D-LPP
2D-DLPP
the 2nd-order tensor methods
GLRAM
TSA
DTSA
the Nth-order tensor methods
MPCA
DATER
GTDA
the method without DI
the method with DI

Figure 1: The classical STs are summarized.

Limiting the recognition tasks to gray-scale facial images, we only consider 2nd-order tensor in this paper. Inheriting the merits from TSA and 2D-DLPP, a novel method which is called Second Order Discriminant Tensor Subspace Analysis (DTSA) is proposed in this paper. Its advantages include:

1. The DI can further improve recognition performance.
2. More spatial information of the images are preserved by presenting the image as a matrix and higher compression ratios with the use of two-sided transformations.
3. Local structure of samples distribution is preserved.

The rest of this paper is organized as follows: in Section 2, we give related definitions and a brief review of 2D-DLPP and TSA; in Section 3, we will introduce the Second Order Discriminant Tensor Subspace Analysis and reveal its connections to 2D-DLPP and TSA, theoretically; in Section 4, the experimental results are reported and analyzed; finally in Section 5, conclusions are drawn and several issues for future works are described.

2. Related Works

In this section, we give definitions of ST and a brief review of 2D-DLPP and TSA algorithms.

2.1. Definitions of ST

With DI, we have a set $\mathcal{X}$ consisting of N samples coming from C classes:

$$\mathcal{X} = \left\{\mathbf{X}_1^1, \mathbf{X}_2^1, \ldots, \mathbf{X}_{N_1}^1, \mathbf{X}_1^2, \mathbf{X}_2^2, \ldots, \mathbf{X}_{N_2}^2, \ldots, \mathbf{X}_1^C, \mathbf{X}_2^C, \ldots, \mathbf{X}_{N_C}^C\right\} \quad (1)$$

where $\mathbf{X}_i^c \in \mathbb{R}^{I_1 \times I_2}$ means the ith sample in the cth class. N_c is the number of samples in the cth class, and $N_1 + N_2 + \ldots + N_C = N$ is satisfied. Without DI, Eq. (1) simplifies to

$$\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\} \quad (2)$$

The task of the ST is to seek a transformation mapping:

$$\mathbf{X}_i \to \mathbf{Y}_i \quad (3)$$

such that the projected point $\mathbf{Y}_i$ is easier to be distinguished in the projective subspace.

With DI, we define two similarity matrices $\mathbf{W}^c$ and $\mathbf{B}$.

Definition 1. In the *within-class similarity matrix* of cth class $\mathbf{W}^c$, each entry W_{ij}^c is the similarity between the samples $\mathbf{X}_i^c$ and $\mathbf{X}_j^c$, and it is defined as: $W_{ij}^c = \exp(-\|\mathbf{X}_i^c - \mathbf{X}_j^c\|_F^2/t)$, where $\|\cdot\|$ is the Frobenius norm of matrix, *i.e.* $\|\mathbf{A}\|_F = \sqrt{\sum_i \sum_j A_{ij}^2}$. Similarly, in the *between-class similarity matrix* $\mathbf{B}$, each entry B_{ij} is the similarity between the mean samples $\overline{\mathbf{X}}_i$ and $\overline{\mathbf{X}}_j$, and it is defined as: $B_{ij} = \exp(-\|\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j\|_F^2/t)$, where $\overline{\mathbf{X}}_i = \frac{1}{N_i}\sum_{k=1}^{N_i} \mathbf{X}_k^i$.

Definition 2. For every $\mathbf{W}^c$, there is a corresponding diagonal matrix $\mathbf{D}^c$, $D_{ii}^c = \sum_j W_{ij}^c$. Then, $\mathbf{L}^c = \mathbf{D}^c - \mathbf{W}^c$ is a laplacian matrix. Therefore, we define *total within-class laplacian matrix* $\mathbf{L}$ as:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}^1 & & & & \\ & \ddots & & & \\ & & \mathbf{L}^c & & \\ & & & \ddots & \\ & & & & \mathbf{L}^C \end{bmatrix} \tag{4}$$

$\mathbf{L}$ is a real symmetric matrix. Similarly, we can also define *between-class laplacian matrix* $\mathbf{H}$: $\mathbf{H} = \mathbf{E} - \mathbf{B}$, where $\mathbf{E}$ is a diagonal matrix, and its entries are column (or row, since $\mathbf{B}$ is symmetric) sum of $\mathbf{B}$, $E_{ii} = \sum_j B_{ij}$. Here, $\mathbf{H}$ is also a real symmetric matrix.

Theorem 1. $\mathbf{L}$ *is semi-positive definite.*

Proof. $\mathbf{L}^c$ is a Laplacian matrix. According to the properties of Laplacian matrix[30], $\mathbf{L}^c$ is real symmetric and semi-positive definite. So, $\mathbf{L}^c$ can be decomposed as $\mathbf{L}^c = (\mathbf{S}^c)^T \mathbf{S}^c$.

$$
\begin{aligned}
\mathbf{L} = & \begin{bmatrix} \mathbf{L}^1 & & & & \\ & \ddots & & & \\ & & \mathbf{L}^c & & \\ & & & \ddots & \\ & & & & \mathbf{L}^C \end{bmatrix} \\
= & \begin{bmatrix} (\mathbf{S}^1)^T \mathbf{S}^1 & & & & \\ & \ddots & & & \\ & & (\mathbf{S}^c)^T \mathbf{S}^c & & \\ & & & \ddots & \\ & & & & (\mathbf{S}^C)^T \mathbf{S}^C \end{bmatrix} \\
= & \begin{bmatrix} \mathbf{S}^1 & & & & \\ & \ddots & & & \\ & & \mathbf{S}^c & & \\ & & & \ddots & \\ & & & & \mathbf{S}^C \end{bmatrix}^T \begin{bmatrix} \mathbf{S}^1 & & & & \\ & \ddots & & & \\ & & \mathbf{S}^c & & \\ & & & \ddots & \\ & & & & \mathbf{S}^C \end{bmatrix}
\end{aligned}
\tag{5}
$$

So, $\mathbf{L}$ is semi-positive definite.

To better understand the following algorithms, a theorem is given:

Theorem 2. *Suppose a set of N matrices $\mathbf{X}_i \in \mathbb{R}^{I_1 \times I_2}$ and a matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$, then we have:*

$$
\begin{aligned}
& \sum_{i,j=1}^{N} W_{ij} \mathbf{X}_i^T \mathbf{X}_j \\
& = \left[\mathbf{X}_1^T, \mathbf{X}_2^T, \cdots, \mathbf{X}_N^T \right] \left(\mathbf{W} \otimes \mathbf{I}_{I_1} \right) \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_N \end{bmatrix}
\end{aligned}
\tag{6}
$$

where operator $\otimes$ is the Kronecher product of the matrices and $\mathbf{I}_{I_1}$ is an identity matrix of order I_1.

Proof.

$$
\begin{aligned}
&\left[\mathbf{X}_1^T, \mathbf{X}_2^T, \cdots, \mathbf{X}_N^T\right](\mathbf{W} \otimes \mathbf{I}_{I_1}) \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_N \end{bmatrix} \\
&= \left[\mathbf{X}_1^T, \mathbf{X}_2^T, \cdots, \mathbf{X}_N^T\right] \begin{bmatrix} W_{11} \otimes \mathbf{I}_{I_1} & W_{12} \otimes \mathbf{I}_{I_1} & \cdots & W_{1N} \otimes \mathbf{I}_{I_1} \\ W_{21} \otimes \mathbf{I}_{I_1} & W_{22} \otimes \mathbf{I}_{I_1} & \cdots & W_{2N} \otimes \mathbf{I}_{I_1} \\ \vdots & \vdots & \ddots & \vdots \\ W_{N1} \otimes \mathbf{I}_{I_1} & W_{N2} \otimes \mathbf{I}_{I_1} & \cdots & W_{NN} \otimes \mathbf{I}_{I_1} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_N \end{bmatrix} \\
&= \left[\sum_{i=1}^{N} \mathbf{X}_i^T W_{i1} \otimes \mathbf{I}_{I_1}, \sum_{i=1}^{N} \mathbf{X}_i^T W_{i2} \otimes \mathbf{I}_{I_1}, \cdots, \sum_{i=1}^{N} \mathbf{X}_i^T W_{iN} \otimes \mathbf{I}_{I_1}\right] \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_N \end{bmatrix} \\
&= \left[\sum_{i=1}^{N} \mathbf{X}_i^T W_{i1}, \sum_{i=1}^{N} \mathbf{X}_i^T W_{i2}, \cdots, \sum_{i=1}^{N} \mathbf{X}_i^T W_{iN}\right] \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_N \end{bmatrix} \\
&= \sum_{i,j=1}^{N} W_{ij} \mathbf{X}_i^T \mathbf{X}_j
\end{aligned}
$$

2.2. Two-dimensional discriminant locality preserving projections (2D-DLPP)

2D-DLPP is a 2D-ST with DI. In this case, Eq. (3) can be specified as:

$$\mathbf{Y}_i^c = \mathbf{X}_i^c \mathbf{V}, \quad i = 1, 2, \ldots, N_c, \quad c = 1, 2, \ldots, C. \tag{7}$$

where $\mathbf{V} \in \mathbb{R}^{I_2 \times L_2}$ $(L_2 \leq I_2)$ is the *transformation matrix.* Given N samples, the objective function of 2D-DLPP is defined as:

$$\max_{\mathbf{V}} \frac{\sum_{i,j=1}^{C} \|\overline{\mathbf{Y}}_i - \overline{\mathbf{Y}}_j\|_F^2 B_{ij}}{\sum_{c=1}^{C} \sum_{i,j=1}^{N_c} \|\mathbf{Y}_i^c - \mathbf{Y}_j^c\|_F^2 W_{ij}^c} \tag{8}$$

where $\overline{\mathbf{Y}}_i = \frac{1}{N_i}\sum_{k=1}^{N_i}\mathbf{Y}_k^i$. Subject Eq. (7) to objective function (8), then we can have the following equation:

$$\max_{\mathbf{V}} \frac{\sum\limits_{i,j=1}^{C} \|\overline{\mathbf{X}}_i\mathbf{V} - \overline{\mathbf{X}}_j\mathbf{V}\|_F^2 B_{ij}}{\sum\limits_{c=1}^{C}\sum\limits_{i,j=1}^{N_c} \|\mathbf{X}_i^c\mathbf{V} - \mathbf{X}_j^c\mathbf{V}\|_F^2 W_{ij}^c} \tag{9}$$

By maximizing the numerator of the above objective function, the projected points $\mathbf{Y}_i$ and $\mathbf{Y}_j$ move far away from each other if their corresponding mean samples of two classes $\overline{\mathbf{X}}_i$ and $\overline{\mathbf{X}}_j$ are close. In the opposite manner, the distance of feature points $\mathbf{Y}_i^c$ and $\mathbf{Y}_j^c$ keeps close if their original samples of the class $\mathbf{X}_i^c$ and $\mathbf{X}_j^c$ are close by minimizing the denominator of objective function. It is easy to observe that the optimal solution of objective function is obtained by maximizing the between class distance and minimizing the within class distance. Thus, by utilizing the DI, 2D-DLPP has better performance than 2D-LPP for recognition.

According to [15], the denominator of Eq. (9) can be simplified as:

$$\frac{1}{2}\sum_{c=1}^{C}\sum_{i,j=1}^{N_c} \|\mathbf{X}_i^c\mathbf{V} - \mathbf{X}_j^c\mathbf{V}\|_F^2 W_{ij}^c = tr\left[\mathbf{V}^T\mathbf{X}^T\left(\mathbf{L}\otimes\mathbf{I}_{I_1}\right)\mathbf{X}\mathbf{V}\right] \tag{10}$$

where

$$\mathbf{X}^c = \begin{bmatrix} \mathbf{X}_1^c \\ \mathbf{X}_2^c \\ \vdots \\ \mathbf{X}_{N_c}^c \end{bmatrix} \tag{11}$$

and

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}^1 \\ \mathbf{X}^2 \\ \vdots \\ \mathbf{X}^C \end{bmatrix} \tag{12}$$

Similarly, the numerator of Eq. (9) can be simplified as:

$$\frac{1}{2}\sum_{i,j=1}^{C} \|\overline{\mathbf{X}}_i\mathbf{V} - \overline{\mathbf{X}}_j\mathbf{V}\|_F^2 B_{ij} = tr\left[\mathbf{V}^T\overline{\mathbf{X}}^T(\mathbf{H}\otimes\mathbf{I}_{I_1})\overline{\mathbf{X}}\mathbf{V}\right] \tag{13}$$

where

$$\overline{\mathbf{X}} = \begin{bmatrix} \overline{\mathbf{X}}_1 \\ \overline{\mathbf{X}}_2 \\ \vdots \\ \overline{\mathbf{X}}_C \end{bmatrix} \tag{14}$$

Subject Eqs. (10) and (13) to (9), the objective function can be simplified as:

$$\max_{\mathbf{V}} \frac{tr\left[\mathbf{V}^T\overline{\mathbf{X}}^T(\mathbf{H}\otimes\mathbf{I}_{I_1})\overline{\mathbf{X}}\mathbf{V}\right]}{tr\left[\mathbf{V}^T\mathbf{X}^T\left(\mathbf{L}\otimes\mathbf{I}_{I_1}\right)\mathbf{X}\mathbf{V}\right]} = \max_{\mathbf{V}} \frac{tr\left(\mathbf{V}^T\mathbf{S}_H\mathbf{V}\right)}{tr\left(\mathbf{V}^T\mathbf{S}_L\mathbf{V}\right)} \tag{15}$$

where, $\mathbf{S}_H = \overline{\mathbf{X}}^T\left(\mathbf{H}\otimes\mathbf{I}_{I_1}\right)\overline{\mathbf{X}}$ and $\mathbf{S}_L = \mathbf{X}^T\left(\mathbf{L}\otimes\mathbf{I}_{I_1}\right)\mathbf{X}$. Eq. (15) can be treated as the following generalized eigenvector problem:

$$\mathbf{S}_H\mathbf{V} = \lambda\mathbf{S}_L\mathbf{V} \tag{16}$$

The transformation matrix $\mathbf{V}$ consists of the d eigenvectors corresponding to the d largest eigenvalues.

2.3. Tensor subspace analysis

TSA is a tensor ST without DI. In the case, Eq. (3) can be specified as

$$\mathbf{Y}_i = \mathbf{U}^T\mathbf{X}_i\mathbf{V} \tag{17}$$

where $\mathbf{U} \in \mathbb{R}^{I_1\times L_1}$ $(L_1 \leq I_1)$ is called *left transformation matrix* and $\mathbf{V} \in \mathbb{R}^{I_2\times L_2}$ $(L_2 \leq I_2)$ is called *right transformation matrix.* TSA aims to find a subspace in which the local structure of samples in manifold can be preserved. A reasonable transformation respecting to the graph embedding framework can be obtained by solving the following objective function:

$$\min_{\mathbf{U},\mathbf{V}} \sum_{ij} W_{ij} \parallel \mathbf{U}^T\mathbf{X}_i\mathbf{V} - \mathbf{U}^T\mathbf{X}_j\mathbf{V} \parallel_F^2 \tag{18}$$

This objective function indicates that the relative distances between the transformed samples are preserved. Except for preserving the local structure, TSA improves the separability of different classes by maximizing the global scatter on the manifold of the projective subspace [21]:

$$\max_{\mathbf{U},\mathbf{V}} \sum_{i} D_{ii} \parallel \mathbf{U}^T\mathbf{X}_i\mathbf{V} \parallel_F^2 \tag{19}$$

By using simple algebraic transformations, these two properties can be formulized as the two optimization problems:

$$\min_{\mathbf{U},\mathbf{V}} \frac{\operatorname{tr}\left[\mathbf{U}^T(\mathbf{D}_V - \mathbf{W}_V)\mathbf{U}\right]}{\operatorname{tr}(\mathbf{U}^T\mathbf{D}_V\mathbf{U})} \tag{20}$$

$$\min_{\mathbf{U},\mathbf{V}} \frac{\operatorname{tr}\left[\mathbf{V}^T(\mathbf{D}_U - \mathbf{W}_U)\mathbf{V}\right]}{\operatorname{tr}(\mathbf{V}^T\mathbf{D}_U\mathbf{V})} \tag{21}$$

where $\mathbf{D}_V = \sum_i D_{ii}\mathbf{X}_i\mathbf{V}\mathbf{V}^T\mathbf{X}_i^T$, $\mathbf{W}_V = \sum_{ij} W_{ij}\mathbf{X}_i\mathbf{V}\mathbf{V}^T\mathbf{X}_j^T$, $\mathbf{D}_U = \sum_i D_{ii}\mathbf{X}_i^T\mathbf{U}\mathbf{U}^T\mathbf{X}_i$ and $\mathbf{W}_U = \sum_{ij} W_{ij}\mathbf{X}_i^T\mathbf{U}\mathbf{U}^T\mathbf{X}_j$. According to the generalized Rayleigh−Ritz theorem [31], assuming $\mathbf{V}$ is fixed, $\mathbf{U}$ is calculated according to the following equation which is a generalized eigenvalue problem:

$$(\mathbf{D}_V - \mathbf{W}_V)\mathbf{u} = \lambda\mathbf{D}_V\mathbf{u} \tag{22}$$

Similarly, assuming $\mathbf{U}$ is fixed, the optimal $\mathbf{V}$ can be obtained according to:

$$(\mathbf{D}_U - \mathbf{W}_U)\mathbf{v} = \lambda\mathbf{D}_U\mathbf{v} \tag{23}$$

Based on the dependency between Eq. (22) and Eq (23), the optimal $\mathbf{U}$ and $\mathbf{V}$ can be calculated using an iterative technique which fixes $\mathbf{U}$ initially to compute the $\mathbf{V}$ and uses updated $\mathbf{V}$ to calculate the new $\mathbf{U}$ repeatedly.

3. Second Order Discriminant Tensor Subspace Analysis

3.1. Problem formulation

Given a set $\mathcal{X}$ in Eq. (1), Eq. (3) can be specified as

$$\mathbf{Y}_i^c = \mathbf{U}^T\mathbf{X}_i^c\mathbf{V}, \quad i = 1, 2, \ldots, N_c, \quad c = 1, 2, \ldots, C. \tag{24}$$

The task is to learn the two matrices $\mathbf{U}$ and $\mathbf{V}$ which project those N samples to a set of the projected points

$$\mathcal{Y} = \left\{\mathbf{Y}_1^1, \mathbf{Y}_2^1, \ldots, \mathbf{Y}_{N_1}^1, \mathbf{Y}_1^2, \mathbf{Y}_2^2, \ldots, \mathbf{Y}_{N_2}^2, \ldots, \mathbf{Y}_1^C, \mathbf{Y}_2^C, \ldots, \mathbf{Y}_{N_C}^C\right\} \tag{25}$$

where $\mathbf{Y}_i^c \in \mathbb{R}^{L_1 \times L_2}$.

Following the idea of 2D-DLPP, we seek $\mathbf{U}$ and $\mathbf{V}$ such that if the two samples $\mathbf{X}_i^c$ and $\mathbf{X}_j^c$ in the same class are close then the corresponding projected points $\mathbf{Y}_i^c$ and $\mathbf{Y}_j^c$ are close as well. A reasonable criterion for the projection is to minimize the following objective function:

$$\min \sum_{c=1}^{C} \sum_{i,j=1}^{N_c} \|\mathbf{Y}_i^c - \mathbf{Y}_j^c\|_F^2 W_{ij}^c \tag{26}$$

Additionally, a reasonable criterion for the projection is to maximize the following objective function:

$$\max \sum_{i,j=1}^{C} \|\overline{\mathbf{Y}}_i - \overline{\mathbf{Y}}_j\|_F^2 B_{ij} \tag{27}$$

By applying multiplicative principle[32], the multiobjective programming problem (26) and (27) is converted into a single-objective programming problem:

$$\max_{\mathbf{U},\mathbf{V}} \frac{\sum\limits_{i,j=1}^{C} \|\overline{\mathbf{Y}}_i - \overline{\mathbf{Y}}_j\|_F^2 B_{ij}}{\sum\limits_{c=1}^{C}\sum\limits_{i,j=1}^{N_c} \|\mathbf{Y}_i^c - \mathbf{Y}_j^c\|_F^2 W_{ij}^c} \tag{28}$$

3.2. Related Definitions

For the convenience of discussing on the proposed algorithm, we introduce some definitions.

Definition 3. For the cth class and the left transformation matrix $\mathbf{U}$, we define the *left transformation column matrix* of the cth class:

$$\mathbf{P}_U^c = \begin{bmatrix} \mathbf{U}^T\mathbf{X}_1^c \\ \mathbf{U}^T\mathbf{X}_2^c \\ \vdots \\ \mathbf{U}^T\mathbf{X}_{N_c}^c \end{bmatrix} \tag{29}$$

and the *total left transformation column matrix*:

$$\mathbf{P}_U = \begin{bmatrix} \mathbf{P}_U^1 \\ \mathbf{P}_U^2 \\ \vdots \\ \mathbf{P}_U^{N_c} \end{bmatrix} \tag{30}$$

Similarly, for the mean value of each class $\overline{\mathbf{X}}_c$ ($c = 1, 2, \ldots, C$), we define *mean left transformation column matrix*

$$\mathbf{Q}_U = \begin{bmatrix} \mathbf{U}^T\overline{\mathbf{X}}_1 \\ \mathbf{U}^T\overline{\mathbf{X}}_2 \\ \vdots \\ \mathbf{U}^T\overline{\mathbf{X}}_{N_c} \end{bmatrix} \tag{31}$$

Definition 4. For the cth class and the right transformation matrix $\mathbf{V}$, we define the *right transformation row matrix* of the cth class:

$$\mathbf{P}_V^c = \left[\mathbf{X}_1^c\mathbf{V}, \mathbf{X}_2^c\mathbf{V}, \ldots, \mathbf{X}_{N_c}^c\mathbf{V}\right] \tag{32}$$

and the *total right transformation row matrix*:

$$\mathbf{P}_V = \left[\mathbf{P}_V^1, \mathbf{P}_V^2, \ldots, \mathbf{P}_V^{N_c}\right] \tag{33}$$

Similarly, for the mean value of each $\overline{\mathbf{X}}_c$ ($c = 1, 2, \ldots, C$), we define *mean right transformation row matrix*

$$\mathbf{Q}_V = \left[\overline{\mathbf{X}}_1\mathbf{V}, \overline{\mathbf{X}}_2\mathbf{V}, \ldots, \overline{\mathbf{X}}_{N_c}\mathbf{V}\right] \tag{34}$$

3.3. DSTA Algorithm

Since the projection onto a second-order tensor subspace consists of two projection matrices, two optimization subproblems can be solved by finding $\mathbf{U}$ and $\mathbf{V}$ such that minimizes Eq. (28).

Theorem 3. *Let $\mathbf{U}$ and $\mathbf{V}$ be the solutions to Eq. (28). For a given $\mathbf{U}$, the matrix $\mathbf{V}$ consists of the L_2 generalized eigenvectors corresponding to the largest L_2 generalized eigenvalues of the matrix pencil $(\mathbf{S}_H^U, \mathbf{S}_L^U)$, where $\mathbf{S}_H^U = \mathbf{Q}_U^T(\mathbf{H} \otimes \mathbf{I}_{L_1})\mathbf{Q}_U$, $\mathbf{S}_L^U = \mathbf{P}_U^T(\mathbf{L} \otimes \mathbf{I}_{L_1})\mathbf{P}_U$.*

Proof. Since $\|\mathbf{A}\|_F^2 = tr(\mathbf{A}^T\mathbf{A})$ and Theorem 2, the denominator of Eq.

(28) can be simplified as:

$$
\begin{aligned}
&\frac{1}{2}\sum_{c=1}^{C}\sum_{i,j=1}^{N_c}\|\mathbf{Y}_i^c-\mathbf{Y}_j^c\|_F^2 W_{ij}^c\\
&=\frac{1}{2}\sum_{c=1}^{C}\sum_{i,j=1}^{N_c} tr\left[(\mathbf{Y}_i^c-\mathbf{Y}_j^c)^T(\mathbf{Y}_i^c-\mathbf{Y}_j^c)\right]W_{ij}^c\\
&=tr\left[\frac{1}{2}\sum_{c=1}^{C}\sum_{i,j=1}^{N_c}(\mathbf{U}^T\mathbf{X}_i^c\mathbf{V}-\mathbf{U}^T\mathbf{X}_j^c\mathbf{V})^T(\mathbf{U}^T\mathbf{X}_i^c\mathbf{V}-\mathbf{U}^T\mathbf{X}_j^c\mathbf{V})W_{ij}^c\right]\\
&=tr\left\{\mathbf{V}^T\left[\frac{1}{2}\sum_{c=1}^{C}\sum_{i,j=1}^{N_c}(\mathbf{U}^T\mathbf{X}_i^c-\mathbf{U}^T\mathbf{X}_j^c)^T(\mathbf{U}^T\mathbf{X}_i^c-\mathbf{U}^T\mathbf{X}_j^c)W_{ij}^c\right]\mathbf{V}\right\}\\
&=tr\left\{\mathbf{V}^T\left\{\sum_{c=1}^{C}\sum_{i,j=1}^{N_c}\left[\left(\mathbf{U}^T\mathbf{X}_i^c\right)^T\mathbf{U}^T\mathbf{X}_i^c-\left(\mathbf{U}^T\mathbf{X}_i^c\right)^T\mathbf{U}^T\mathbf{X}_j^c\right]W_{ij}^c\right\}\mathbf{V}\right\}\\
&=tr\left\{\mathbf{V}^T\left\{\sum_{c=1}^{C}\left[\sum_{i=1}^{N_c}\left(\mathbf{U}^T\mathbf{X}_i^c\right)^T\mathbf{U}^T\mathbf{X}_i^c\sum_{j=1}^{N_c}W_{ij}^c-\sum_{i,j=1}^{N_c}\left(\mathbf{U}^T\mathbf{X}_i^c\right)^T\mathbf{U}^T\mathbf{X}_j^cW_{ij}^c\right]\right\}\mathbf{V}\right\}\\
&=tr\left\{\mathbf{V}^T\left\{\sum_{c=1}^{C}\left[(\mathbf{P}_U^c)^T(\mathbf{D}^c\otimes\mathbf{I}_{L_1})\mathbf{P}_U^c-(\mathbf{P}_U^c)^T(\mathbf{W}^c\otimes\mathbf{I}_{L_1})\mathbf{P}_U^c\right]\right\}\mathbf{V}\right\}\\
&=tr\left\{\mathbf{V}^T\mathbf{P}_U^T\left[(\mathbf{D}-\mathbf{W})\otimes\mathbf{I}_{L_1}\right]\mathbf{P}_U\mathbf{V}\right\}\\
&=tr\left[\mathbf{V}^T\mathbf{P}_U^T\left(\mathbf{L}\otimes\mathbf{I}_{L_1}\right)\mathbf{P}_U\mathbf{V}\right]\\
&=tr\left(\mathbf{V}^T\mathbf{S}_L^U\mathbf{V}\right)
\end{aligned}
\tag{35}
$$

According to Theorem 1, $\mathbf{L}$ is semi-positive definite. The identity matrix $\mathbf{I}_{L_1}$ is semi-positive definite, so $\mathbf{L}\otimes\mathbf{I}_{L_1}$ is semi-positive definite. Suppose there is a real matrix $\mathbf{S}'$, such that $\mathbf{L}\otimes\mathbf{I}_{L_1}=\mathbf{S}'^T\mathbf{S}'$.

$$\mathbf{S}_L^U=\mathbf{P}_U^T(\mathbf{L}\otimes\mathbf{I}_{L_1})\mathbf{P}_U=\mathbf{P}_U^T(\mathbf{S}'^T\mathbf{S}')\mathbf{P}_U=(\mathbf{S}'\mathbf{P}_U)^T(\mathbf{S}'\mathbf{P}_U)\tag{36}$$

Thus, $\mathbf{S}_L^U$ is semi-positive definite.

Similarly, the numerator of Eq. (28) can be simplified as:

$$
\begin{aligned}
&\frac{1}{2}\sum_{i,j=1}^{C}\|\overline{\mathbf{Y}}_i - \overline{\mathbf{Y}}_j\|_F^2 B_{ij}\\
&= \frac{1}{2}\sum_{i,j=1}^{C} tr\left[(\overline{\mathbf{Y}}_i - \overline{\mathbf{Y}}_j)^T(\overline{\mathbf{Y}}_i - \overline{\mathbf{Y}}_j)\right] B_{ij}\\
&= \frac{1}{2}\sum_{i,j=1}^{C} tr\left[(\mathbf{U}^T\overline{\mathbf{X}}_i\mathbf{V} - \mathbf{U}^T\overline{\mathbf{X}}_j\mathbf{V})^T(\mathbf{U}^T\overline{\mathbf{X}}_i\mathbf{V} - \mathbf{U}^T\overline{\mathbf{X}}_j\mathbf{V})\right] B_{ij}\\
&= \frac{1}{2}\sum_{i,j=1}^{C} tr\left[\mathbf{V}^T(\mathbf{U}^T\overline{\mathbf{X}}_i - \mathbf{U}^T\overline{\mathbf{X}}_j)^T(\mathbf{U}^T\overline{\mathbf{X}}_i - \mathbf{U}^T\overline{\mathbf{X}}_j)\mathbf{V}\right] B_{ij}\\
&= tr\left\{\sum_{i,j=1}^{C}\mathbf{V}^T\left[(\mathbf{U}^T\overline{\mathbf{X}}_i)^T\mathbf{U}^T\overline{\mathbf{X}}_i - (\mathbf{U}^T\overline{\mathbf{X}}_i)^T\mathbf{U}^T\overline{\mathbf{X}}_j\right]\mathbf{V}B_{ij}\right\}\\
&= tr\left\{\sum_{i=1}^{C}\mathbf{V}^T\left[(\mathbf{U}^T\overline{\mathbf{X}}_i)^T\mathbf{U}^T\overline{\mathbf{X}}_i\sum_{j=1}^{C}B_{ij}\right]\mathbf{V} - \sum_{i,j=1}^{C}\mathbf{V}^T(\mathbf{U}^T\overline{\mathbf{X}}_i)^T B_{ij}\mathbf{U}^T\overline{\mathbf{X}}_j\mathbf{V}\right\}\\
&= tr\left\{\mathbf{V}^T\mathbf{Q}_U^T[(\mathbf{E}-\mathbf{B})\otimes\mathbf{I}_{L_1}]\mathbf{Q}_U\mathbf{V}\right\}\\
&= tr\left[\mathbf{V}^T\mathbf{Q}_U^T(\mathbf{H}\otimes\mathbf{I}_{L_1})\mathbf{Q}_U\mathbf{V}\right]\\
&= tr\left(\mathbf{V}^T\mathbf{S}_H^U\mathbf{V}\right)
\end{aligned}
\tag{37}
$$

where $\mathbf{S}_H^U$ is also semi-positive definite. Therefore, for a given $\mathbf{U}$, the solution to Eq. (28) can be converted into the following optimal problem about a variable $\mathbf{V}$:

$$\max_{\mathbf{V}}\frac{tr(\mathbf{V}^T\mathbf{S}_H^U\mathbf{V})}{tr(\mathbf{V}^T\mathbf{S}_L^U\mathbf{V})} \tag{38}$$

It is easy to see that the optimal $\mathbf{V}$ should be the generalized eigenvalues problem:

$$\mathbf{S}_H^U\mathbf{v} = \lambda\mathbf{S}_L^U\mathbf{v} \tag{39}$$

the matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{L_2}]$ consists of the L_2 generalized eigenvectors corresponding to the largest L_2 generalized eigenvalues of the matrix pencil $(\mathbf{S}_H^U, \mathbf{S}_L^U)$.

Theorem 4. *Let* $\mathbf{U}$ *and* $\mathbf{V}$ *be the solution to Eq. (28). For a given* $\mathbf{V}$*, the matrix* $\mathbf{U}$ *consists of the* L_1 *generalized eigenvectors corresponding to the largest* L_1 *generalized eigenvalues of the matrix pencil* $(\mathbf{S}_H^V, \mathbf{S}_L^V)$*, where* $\mathbf{S}_H^V = \mathbf{Q}_V^T(\mathbf{H} \otimes \mathbf{I}_{L_2})\mathbf{Q}_V$*,* $\mathbf{S}_L^V = \mathbf{P}_V^T(\mathbf{L} \otimes \mathbf{I}_{L_2})\mathbf{P}_V$*.*

Proof. The proof is similar to the Theorem 3.

From the above analysis, we see that the optimizations of **U** and **V** depend on each other. From Theorem 3 and Theorem 4, an iterative procedure can be utilized to solve Eq. (28), which is summarized in Algorithm 1.

Algorithm 1 DTSA

```
INPUT: a set of $N$ samples $\mathcal{X}$ with DI, the number of reduced dimensions $L_1, L_2$, and the maximum iteration times $T_{max}$
OUTPUT: a set of projected points $\mathcal{Y}$ and two transformation matrices $\mathbf{U}$, $\mathbf{V}$
Algorithm:
initialize $\mathbf{U}$ with an identity matrix
Compute the total within-class laplacian matrix $\mathbf{L}$ and the between-class laplacian matrix $\mathbf{H}$ according to Definition 2.

for $t = 0$ to $T_{max}$ do
    Compute $\mathbf{P}_U$ and $\mathbf{Q}_U$ by Definition 3.
    $\mathbf{S}_H^U = \mathbf{Q}_U^T(\mathbf{H} \otimes \mathbf{I}_{L_1})\mathbf{Q}_U$; $\mathbf{S}_L^U = \mathbf{P}_U^T(\mathbf{L} \otimes \mathbf{I}_{L_1})\mathbf{P}_U$
    the matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{L_2}]$ consists of the $L_2$ generalized eigenvectors corresponding to the largest $L_2$ generalized eigenvalues of the matrix pencil $(\mathbf{S}_H^U, \mathbf{S}_L^U)$.

    Compute $\mathbf{P}_V$ and $\mathbf{Q}_V$ according to Definition 4.
    $\mathbf{S}_H^V = \mathbf{Q}_V^T(\mathbf{H} \otimes \mathbf{I}_{L_2})\mathbf{Q}_V$; $\mathbf{S}_L^V = \mathbf{P}_V^T(\mathbf{L} \otimes \mathbf{I}_{L_2})\mathbf{P}_V$
    the matrix $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{L_1}]$ consists of the $L_1$ generalized eigenvectors corresponding to the largest $L_1$ generalized eigenvalues of the matrix pencil $(\mathbf{S}_H^V, \mathbf{S}_L^V)$.
end for

Compute a set of projected points $\mathcal{Y}$ by using Eq. (24)
```

3.4. Feature extraction and classification

A set of features $\mathcal{Y}$ are extracted for each sample in $\mathcal{X}$ by using Eq. (24). For a test sample $\mathbf{X}_{test}$, similarly, the corresponding projected point $\mathbf{Y}_{test}$ is:

$$\mathbf{Y}_{test} = \mathbf{U}^T \mathbf{X}_{test} \mathbf{V} \tag{40}$$

We choose the nearest-neighbor classifier which uses the Euclidean distance to measure the similarity and identify the target label of each test sample.

$$d(\mathbf{Y}_{test}\mathbf{Y}_i^c) = \|\mathbf{Y}_{test} - \mathbf{Y}_i^c\|_F \tag{41}$$

The $\mathbf{X}_{test}$ is assigned to cth class according to its closest class center $\mathbf{Y}_i^c$.

3.5. Connections to 2D-DLPP and TSA

In this section, the relations between the proposed algorithm and the 2D-DLPP or TSA are analyzed. Compared to the 2D-DLPP, DTSA employs two-sided transformations rather than single-sided one, and yields higher recognition rate. In Algorithm 1, we fix $\mathbf{U}$ as an identity matrix of order I_1 and set $L_1 = I_1$. $\mathbf{P}_U^c$, $\mathbf{P}_U$ and $\mathbf{Q}_U$ defined in Definition 3 are equal to $\mathbf{X}^c$ in Eq. (11), $\mathbf{X}$ in Eq. (12) and $\overline{\mathbf{X}}$ in Eq. (14), respectively. $\mathbf{S}_H^U$ and $\mathbf{S}_L^U$ in Theorem 3 are equal to $\mathbf{S}_H$ and $\mathbf{S}_L$ in Eq. (15), respectively. Consequently, Eq. (39) is equivalent to Eq. (16). Since $\mathbf{U}$ is fixed, $\mathbf{U}$ does not need to be solved using Theorem 4 through iterative procedure. Thus, the solution of DTSA is the solution of 2D-DLPP.

DTSA is a version of TSA with DI. In DTSA, if there is only one class such that $C = 1$ and $\mathbf{W} = \mathbf{W}^1$, then Eq. (26) becomes Eq. (18) and Eq. (27) becomes

$$\max \|\overline{\mathbf{Y}}\|_F^2 \tag{42}$$

which calculates the maximum value of the mean of new projected features. However, the scatter of new projected features are maximized in Eq.(19) in TSA.

4. Experiments and results

4.1. Database and experimental set

Three well-known face database ORL[1], Yale[2] and the Extended Yale Face Database B[33] (denoted by YaleB hereafter) were used in our experiments.

[1] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[2] http://cvc.yale.edu/projects/yalefaces/yalefaces.html

Figure 2: Sample images of one individual from the ORL database.

Figure 3: Sample images of one individual in the YALE database.

The ORL database collects images from 40 individuals, and 10 different images are captured for each individual. For each individual, the images with different facial expressions and details are obtained at different times. The face in the images may be rotated, scaled and be tilting in some degree. The sample images of one individual from the ORL database are shown in Figure 2.

There are total of 165 gray scale images for 15 individuals where each individual has 11 images in Yale face database. The images demonstrate variations in lighting condition, facial expression (normal, happy, sad, sleepy, surprised, and wink). The sample images of one individual from the Yale database are showed in Figure 3.

The YaleB contains 21888 images of 38 individuals under 9 poses and 64 illumination conditions. A subset containing 2414 frontal pose images of 38 individuals under different illuminations per individual is extracted. The sample images of one individual from the YaleB database are showed in Figure 4.

From each of the face database mentioned above, the image set is partitioned into the different gallery and probe sets. In this paper, the *Gm/Pn* indicates that m images per individual are randomly selected for training and the remaining n images are used for testing. For each partition, we use 50 random splits for cross-validation tests. All images are manually cropped and resized to two different resolutions, 32×32 pixels and 64×64 pixels. These cropped images and random splits can be downloaded from the Web[3].

[3] `http://www.zjucadcg.cn/dengcai/Data/FaceData.html`

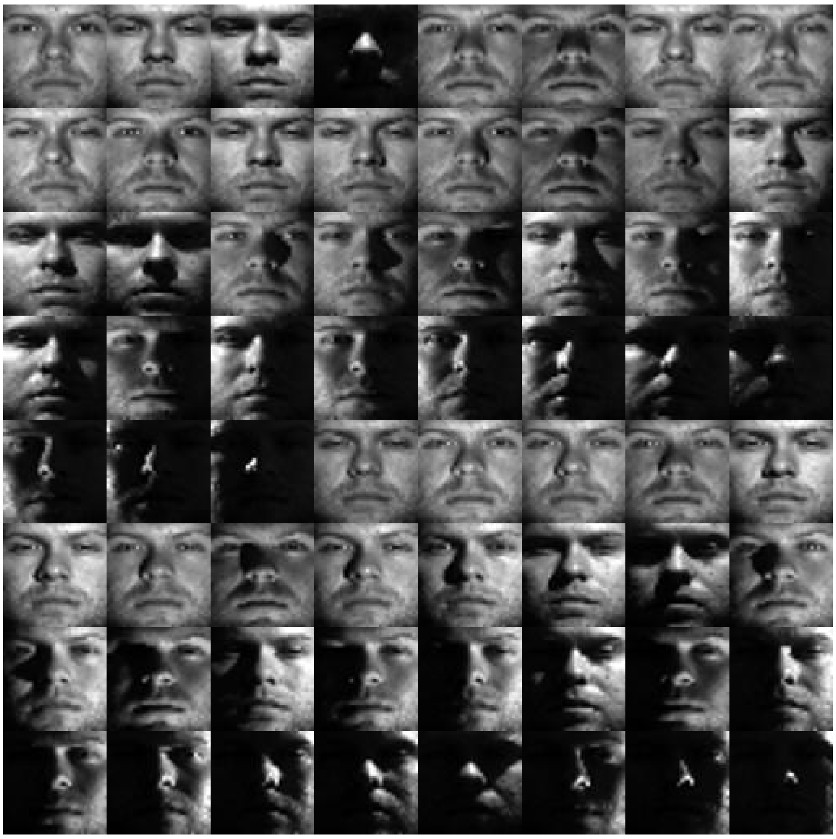


Figure 4: Sample images of one individual from the YaleB database.

4.2. Properties of algorithms for ST

In this section, we will investigate the properties of the algorithms in Figure 1. The properties include time complexity, space complexity and compression. The time and space complexities of the algorithms are illustrated in Table 1. Data obtained from references are marked correspondingly. For 1D-STs, generally, $I_1 I_2$ is far greater than the other values. So, their time and space complexities are dependent on $I_1 I_2$. For 2D-STs, the time and space complexities of the methods which are based on preserving the local structure of samples are $O(M^2 I^3)$ and $O(M^2 I_1^2)$, because of the Kronecher product. The analysis of the time complexity of DTSA is covered in Section 4.4. In Table 1, T is the number of iterations. For some, we assume $I_1 = I_2 = \ldots = I_M = I$ for simplicity.

Table 1: Time and space complexities of algorithms in Figure 1

Algorithms	time complexity	space complexity
PCA	$O((I_1 I_2)^3)$ [34]	$O((I_1 I_2)^2)$ [34]
LDA	$O((I_1 I_2)^3)$	$O((I_1 I_2)^2)$
LPP	$O((I_1 I_2)^3)$	$O((I_1 I_2)^2)$
DLPP	$O((I_1 I_2)^3)$	$O((I_1 I_2)^2)$
2D-PCA	$O(I_2^3)$ [35]	$O(I_1 I_2)$
2D-LDA	$O(I_2^3)$	$O(I_1 I_2)$
2D-LPP	$O(M^2 I^3)$	$O(M^2 I_1^2)$
2D-DLPP	$O(M^2 I^3)$	$O(M^2 I_1^2)$
GLRAM	$O(T(I_1 + I_2)^2 \max(L_1, L_2)N)$ [19]	$O(I_1 I_2)$ [19]
TSA	$O(T(I_1^3 + I_2^3 + N^2 (I_1 \times I_2)^{\frac{3}{2}}))$ [36]	$O(I_1 I_2)$
DTSA	$O(T((C^2 + N^2) I^3))$	$O(M^2 I_1^2)$
MPCA	$O((M+1)MNI^{(M+1)} + M \cdot I^3)$ [22]	$O(\prod_{m=1}^{M} I_m)$ [22]
GTDA	$O(T \sum_{m=1}^{M} I_m^3)$ [24]	$O(\sum_{m=1}^{M} I_m^3)$ [24]
DATER	$O(T(MI^{M+1} + MI^3))$ [23]	$O(\sum_{m=1}^{M} I_m^3)$

In these algorithms, the ones which preserve the global structure of samples without the DI are also used for the image compression. We investigate their compressions using the criterion: RMSE versus different values of Compression Ratios (CRs) [37]. The RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{M} \sum_{i=1}^{M} \|\mathbf{X}_i - \widetilde{\mathbf{X}}_i\|_F^2} \qquad \widetilde{\mathbf{X}}_i \text{is the reconstruction of } \mathbf{X}_i \tag{43}$$

The experiments are implemented using all images in the Yale face database as the training set to compare their RMSEs at various CRs. CRs is defined as $NI_1 I_2 / s$ with s as the scale factor required to the certain algorithm. According to [19], given a lower dimension d, we have $s = d(N + I_1 I_2)$ for PCA,

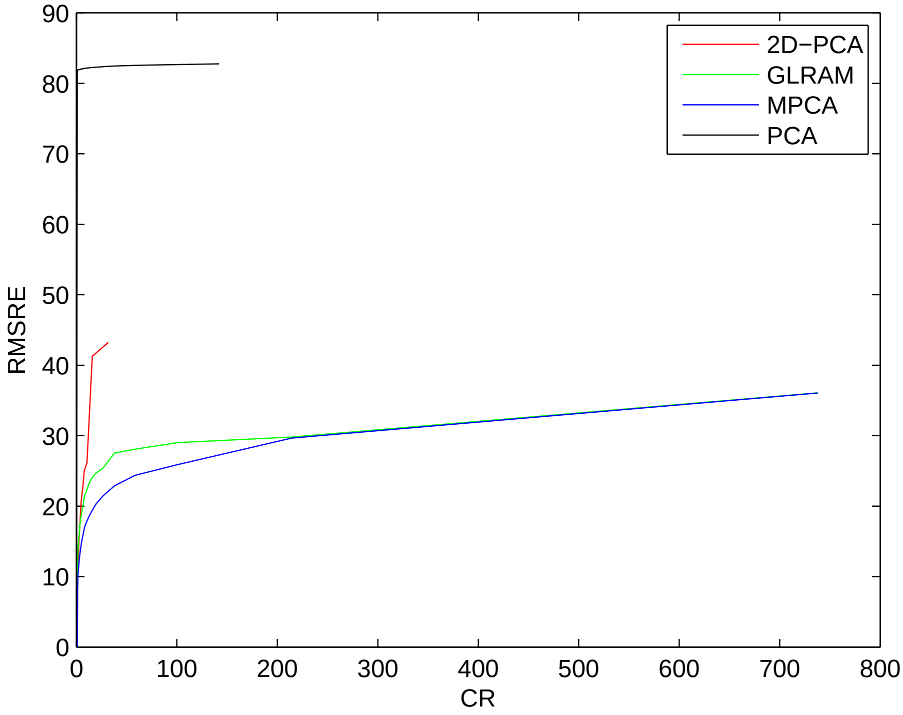


Figure 5: RMSEs vs. various CRs of PCA, 2D-PCA, GLRAM and MPCA on the Yale database.

$s = d(NI_1 + I_2)$ for 2D-PCA, and $s = d(dN + I_1 + I_2)$ for the two tensor algorithms (GLRAM and MPCA). For clarity, we set the reduced dimension L_2 is d for 2D-PCA, and set $L_1 = L_2 = d$ for the two tensor algorithms. Figure 5 plots the RMSE for different CRs. The results show that the two tensor algorithms produce the best compression performance. And we can also see better performance of 2D-PCA than PCA.

4.3. Analysis of Optimal Dimensions

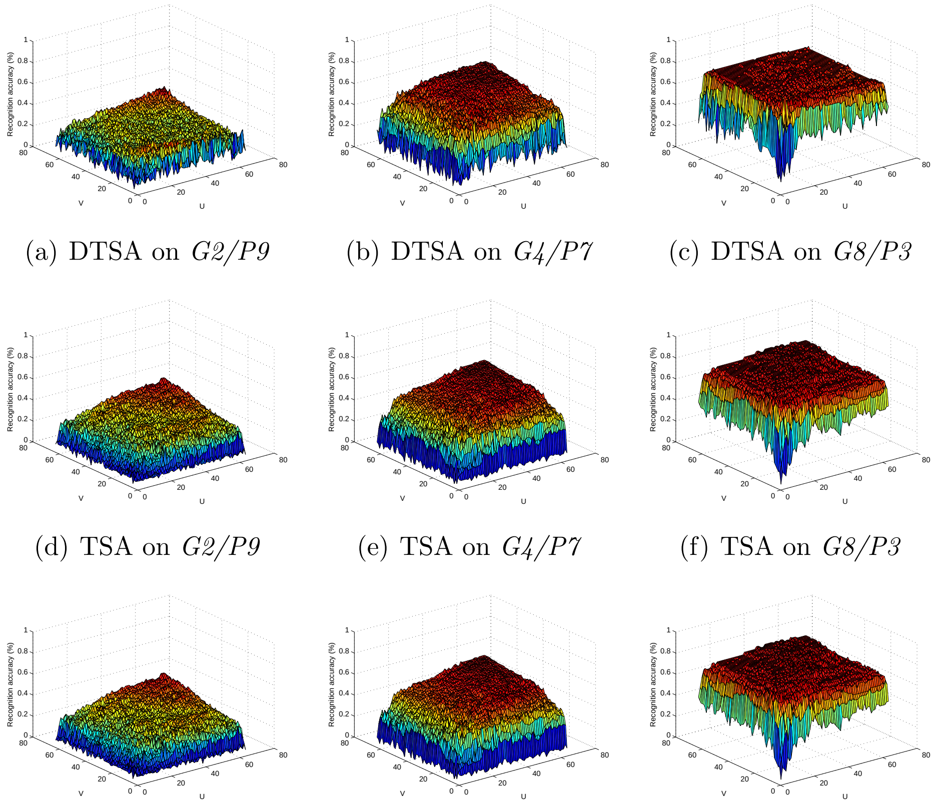
In this section, we discuss the optimal dimensions for three different tensor STs, namely DTSA, TSA[4] and GLRAM[5]. The experiments are conducted on Yale facial database using the 64×64 resolution. Three partitions (*G2/P9*,*G4/P7*,*G8/P3*) are selected and one split is randomly selected from each partition. The performances of the three algorithms versus their dimensions are shown in Figure 6. As shown in the figure, the best performances occur on *full transformation* $(L_1 = I_1, L_2 = I_2)$. Following, we use full transformation to get two full transformation matrices $\mathbf{U}$ and $\mathbf{V}$. Then the optimal dimensions are determined by running over every possible dimension. The performances versus these dimensions are illustrated in Figure 7. The optimal dimensions are about 20 for all three algorithms. Experiments with the same settings are then implemented on the 32×32 resolution as well and the results are shown in Figure 8. From the figure, we can see that the optimal dimensions of $\mathbf{U}$ and $\mathbf{V}$ are both about 10 for the three algorithms. We can draw a conclusion that the optimal dimensions doubles as the image resolution doubles. And the best performance occurs in the first one-third of columns of full transformation matrix.

4.4. Time complexity

In this section, we discuss the time complexities of DTSA and TSA. Suppose the $m \times p$ matrix $\mathbf{A}$ and the $p \times n$ matrix $\mathbf{B}$, the matrix multiplication $\mathbf{AB}$ requires about $p \times m \times n$ floating-point multiplications. Suppose the $p \times q$ matrix $\mathbf{M}$ and the $m \times n$ matrix $\mathbf{N}$, the Kronecher product of the matrices $\mathbf{M} \otimes \mathbf{N}$ requires about $p \times q \times m \times n$ floating-point multiplications. For each iteration of TSA, $\mathbf{D}_V$, $\mathbf{W}_V$, $\mathbf{D}_U$ and $\mathbf{W}_U$ need to be computed, and require $2 \times N \times L_1 \times I_1^2$, $2 \times N^2 \times L_1 \times I_1^2$, $2 \times N \times L_2 \times I_2^2$ and $2 \times N^2 \times L_2 \times I_2^2$ floating-point multiplications, respectively. The costs of $\mathbf{U}^T\mathbf{X}_i$ and $\mathbf{X}_i\mathbf{V}$ in TSA are ignored, because the same costs appear in DTSA as well. In each iteration of DTSA, $\mathbf{S}_H^U$, $\mathbf{S}_L^U$, $\mathbf{S}_H^V$ and $\mathbf{S}_L^V$ need to be computed, and require $C^2 \times L_1^2 \times (2I_1+1)$, $N^2 \times L_1^2 \times (2I_1+1)$, $C^2 \times L_2^2 \times (2I_2+1)$ and $N^2 \times L_2^2 \times (2I_2+1)$ floating-point multiplications, respectively. Generally, N is far greater than the other quantities (such as C, I_1, I_2, L_1 and L_2). The costs of $\mathbf{W}_V$ and

[4] The matlab code can be downloaded from `http://www.zjucadcg.cn/dengcai/Data/data.html`

[5] The matlab code can be downloaded from `http://www-users.cs.umn.edu/~jieping/GLRAM/`

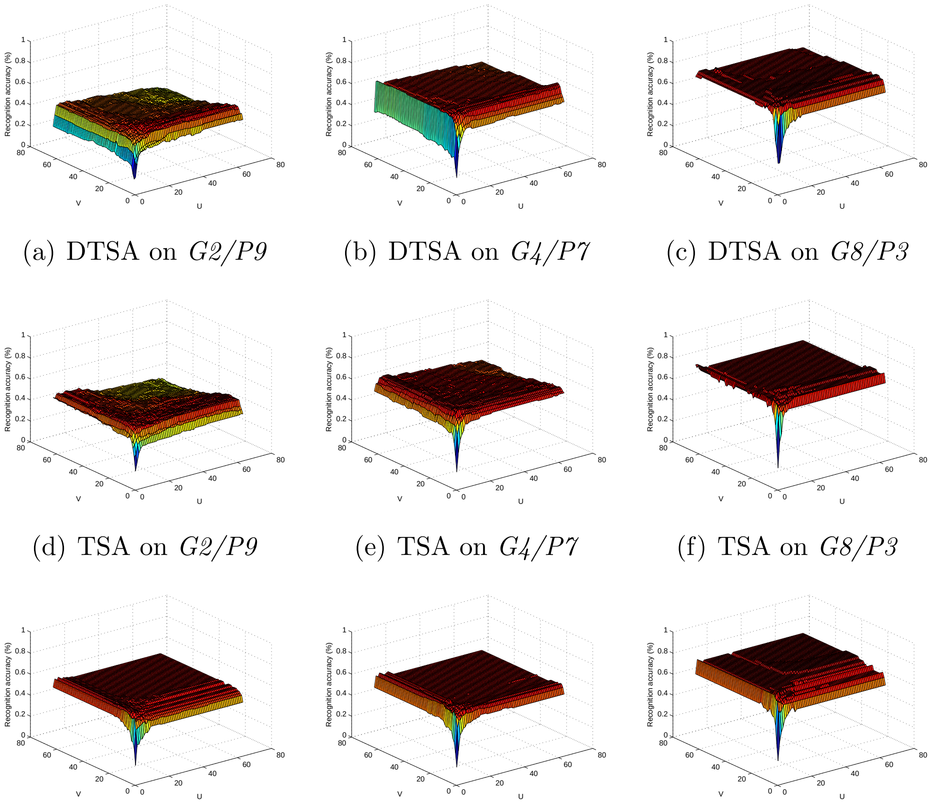


(a) DTSA on *G2/P9* (b) DTSA on *G4/P7* (c) DTSA on *G8/P3*

(d) TSA on *G2/P9* (e) TSA on *G4/P7* (f) TSA on *G8/P3*

(g) GLRAM on *G2/P9* (h) GLRAM on *G4/P7* (i) GLRAM on *G8/P3*

Figure 6: The performance of three tensor STs vs. the dimensions on 64 × 64 resolution

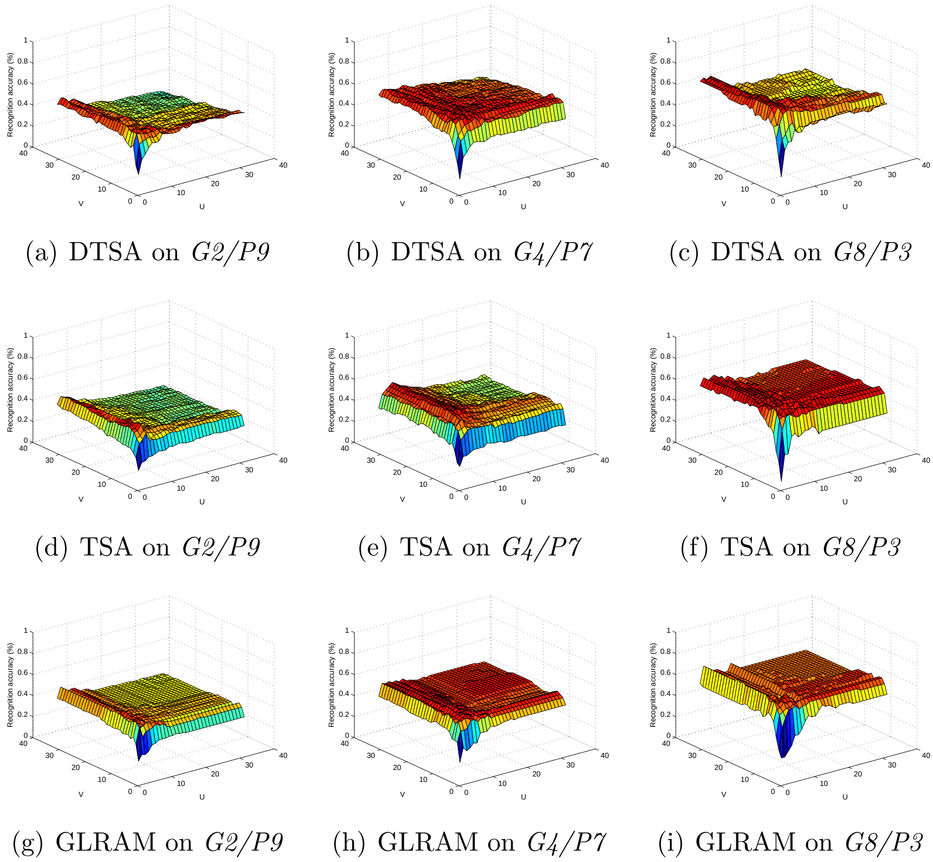


(a) DTSA on *G2/P9* (b) DTSA on *G4/P7* (c) DTSA on *G8/P3*

(d) TSA on *G2/P9* (e) TSA on *G4/P7* (f) TSA on *G8/P3*

(g) GLRAM on *G2/P9* (h) GLRAM on *G4/P7* (i) GLRAM on *G8/P3*

Figure 7: The performance of three tensor STs vs. the dimensions on full transformation on 64 × 64 resolution



(a) DTSA on *G2/P9* (b) DTSA on *G4/P7* (c) DTSA on *G8/P3*

(d) TSA on *G2/P9* (e) TSA on *G4/P7* (f) TSA on *G8/P3*

(g) GLRAM on *G2/P9* (h) GLRAM on *G4/P7* (i) GLRAM on *G8/P3*

Figure 8: The performance of three tensor STs vs. the dimensions on full transformation on 32 × 32 resolution

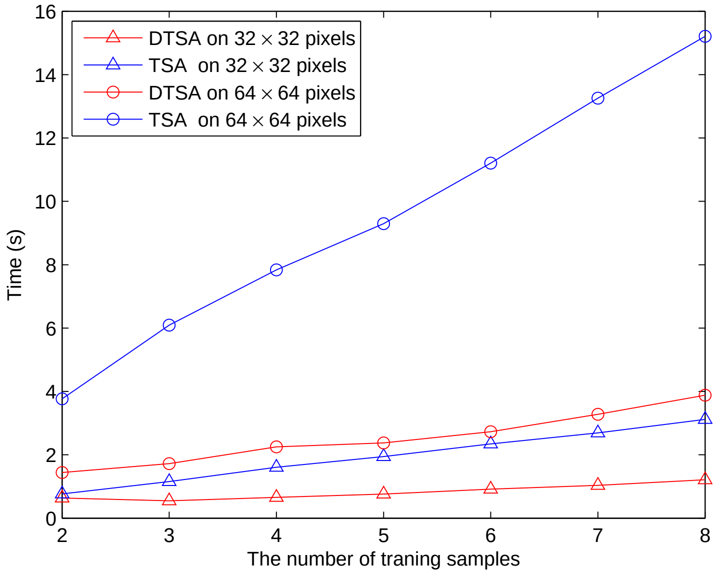


Figure 9: Comparison of TSA and DTSA with respect to the time on ORL database.

$\mathbf{W}_U$ in TSA are equivalent to those of $\mathbf{S}_L^V$ and $\mathbf{S}_L^U$ in DTSA. In most cases, however, N is far greater than C^2. So, the cost of TSA is greater than that of DTSA.

The three face database described above were used to compare the time complexities between DTSA and TSA. For each database, we run DTSA and TSA using different sizes of training set. For example, in ORL database, if we select three images for each subject as training images, we have 120 training images in total (recall that there are 40 subjects in ORL database). We run DTSA and TSA in Matlab R2009b on a PC with Intel Xeon E5430 and 16 GB DDR2 memory. For each facial database, we run experiments on the two resolutions. Figure 9 and Figure 10 show the results in ORL and Yale database. It is clear that the proposed algorithm is more efficient than TSA. As the size of training set increases, the curve of TSA grows more rapidly than DTSA. For databases, such as YaleB, where a relatively large amount of images is extracted per individual, as the size of training set increases, the time cost of the proposed method is slightly higher than TSA in 32 × 32 resolution. However, in 64 × 64 resolution, the time cost of DTSA is still lower than TSA. The conclusion can be seen from Figure 11.

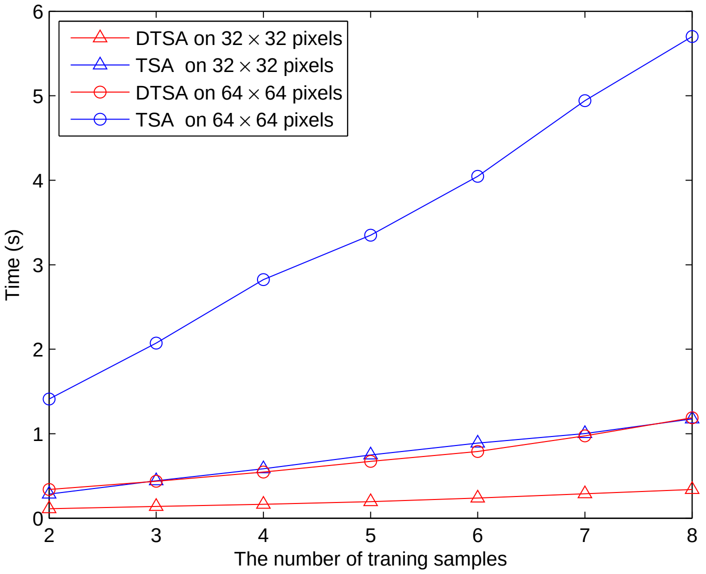


Figure 10: Comparison of TSA and DTSA with respect to the time on Yale database.

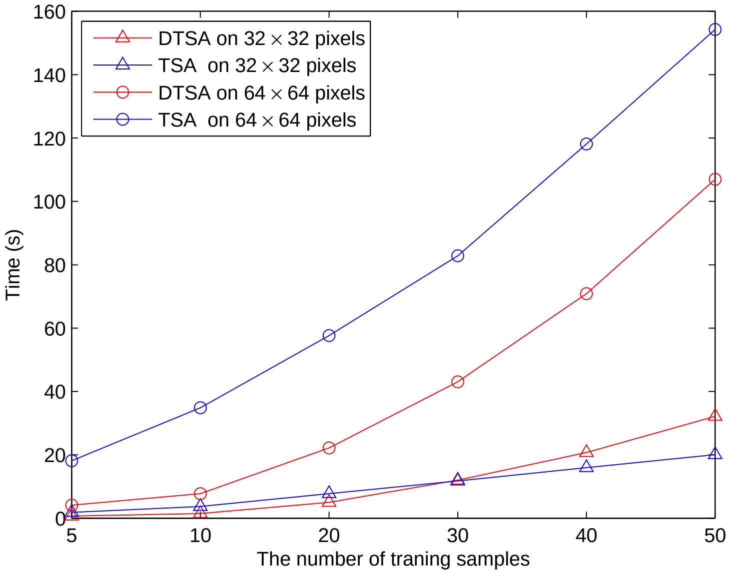


Figure 11: Comparison of TSA and DTSA with respect to the time on YaleB database.

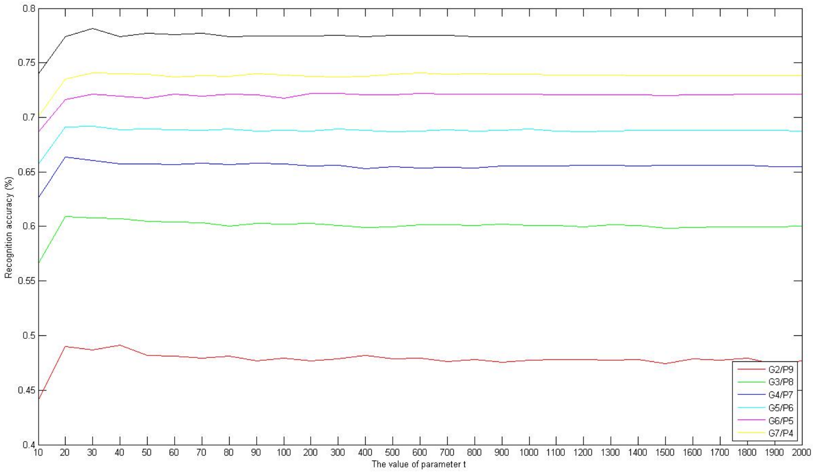


Figure 12: The various values of parameters t of DTSA on YALE face database.

4.5. Sensitivity of parameter t

For 2D-LPP, 2D-DLPP, TSA and DTSA, the heat kernel $\exp(-\|x-y\|^2/t)$ is used and a value has to be assigned to parameter t. YALE face database is used to investigate the sensitivity of the parameter t. To compare the four algorithms, tests for each algorithms are carried out 50 times and the final performance is obtained by averaging the results of all 50 tests. The results are illustrated in Figure 12, Figure 13, Figure 14 and Figure 15. As shown in the four figures, when $t > 50$, recognition accuracies of these four algorithms are not sensitive to the parameter t. It can also be found that the sensitivity of the parameter t is related to the size of the training set. As the size of the training set increases, the performance is more insensitive to t. Compared to 2D-LPP and 2D-DLPP, DTSA and TSA are more sensitive to the parameter t, because of iterations in DTSA and TSA. It is interesting that, unlike DTSA as shown in the Figure 13, the accuracy of TSA increases for 2 and 3 training samples while it decreases for others, when $t < 20$. To avoid the effects from parameter sensitivity, the parameter t is set as 1000 in the following experiments.

4.6. The performance of DTSA compared to other algorithms

In this section, we compare the performance of DTSA with other classical algorithms using the matrix representation. These algorithms are 2D-PCA,

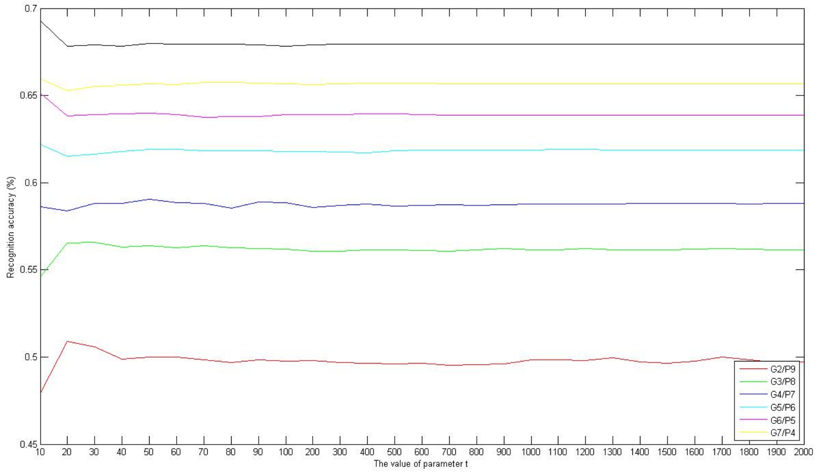


Figure 13: The various values of parameters t of TSA on YALE face database.

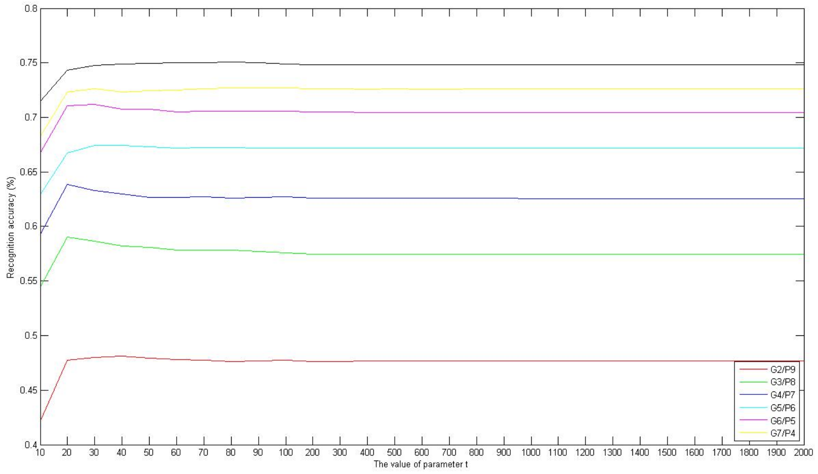


Figure 14: The various values of parameters t of 2D-DLPP on YALE face database.

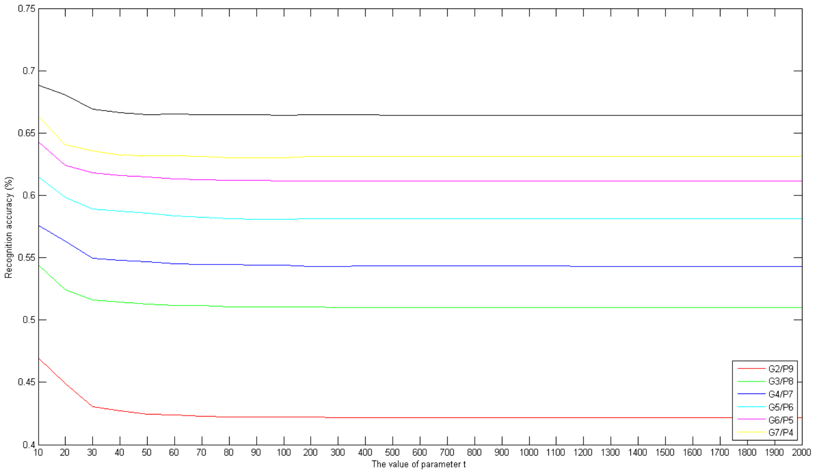


Figure 15: The various values of parameters t of 2D-LPP on YALE face database.

2D-LDA, 2D-LPP, 2D-DLPP, DATER, GTDA, GLRAM, TSA and TMFA. For 2D-LPP, 2D-DLPP, TSA and DTSA, the heat kernel $\exp(-\|x-y\|^2/t)$ is used. We use k nearest neighbors to construct the adjacency graph in 2D-LPP and TSA. The parameter k is set as 12. In TMFA, the two graphs: the intrinsic graph and the penalty graph need to be constructed using k nearest neighbors. The parameters k_1 and k_2 are set to be 12 and 200. For GTDA, we choose 0.1 for ξ and 0.001 for ϵ. For DATER, ϵ is set as 0.01. There are 10 iterations in GLRAM, TSA, TMFA and DTSA.

The first experiment is implemented on the ORL facial database as described in Section 4.1. The maximum average recognition accuracy and the standard deviation across 50 runs of tests of each algorithm are shown in Table 2. The recognition accuracy curve versus the variation in size of training set is shown in Figure 16.

As we can see in Table 2, in every partition, the performance of DTSA reaches the top, except for partition *G2/P8*, where the standard deviation of DTSA is the smallest. In the situations of smaller size of training set (*G2/P8*,*G3/P7* and *G4/P6*), the recognition accuracy of DTSA is about 10 percent more than that of TSA. The results are constant with the theoretical analysis that the PCA subspace without DI is not ideal for face recognition compared to LDA with DI[6]. For the 4 2D-STs, we see that the methods with DI outperform the methods without DI and the local methods are superior to corresponding global ones as shown in Figure 16. For the tensor STs, TSA which preserves the local structure of samples is better than GLRAM which preserves the global structure of samples. The proposed method DTSA has DI and preserves the local structure of samples. So it outperforms other Second Order Tensor methods.

Table 2: Recognition accuracy (%) on ORL database (mean±std)

Partitions	2D-PCA	2D-LDA	2D-LPP	2D-DLPP	GLRAM
G2/P8	72.76±2.71	75.67±2.65	72.66±3.02	79.34±3.62	72.00±3.11
G3/P7	80.72±2.25	82.96±2.23	80.13±2.40	87.21±2.28	80.17±2.48
G4/P6	86.09±1.91	88.22±1.78	86.24±1.84	91.37±1.70	85.81±1.93
G5/P5	89.41±2.04	91.61±1.84	89.52±2.19	93.73±1.98	89.23±2.20
G6/P4	91.93±1.88	92.90±1.90	91.91±1.91	94.65±1.55	91.74±1.93
G7/P3	94.02±2.26	94.60±1.99	93.67±2.38	96.60±1.89	93.98±2.46
G8/P2	95.63±1.98	95.63±2.57	95.52±2.14	97.20±1.74	95.58±2.15
Partitions	TMFA	GTDA	DATER	TSA	DTSA
G2/P8	81.39±3.35	72.16±3.12	81.86±3.40	73.14±3.06	**83.76±3.39**
G3/P7	90.35±1.97	80.37±2.44	90.36±1.95	80.49±2.59	**91.61±1.73**
G4/P6	93.79±1.84	86.02±1.92	93.76±1.86	86.83±1.84	**94.55±1.52**
G5/P5	95.98±1.51	89.40±2.30	96.05±1.55	90.33±1.80	**96.68±1.37**
G6/P4	96.65±1.32	91.93±1.93	96.95±1.18	92.40±1.75	**97.46±1.09**
G7/P3	97.20±1.53	94.20±2.47	97.57±1.52	94.47±2.10	**98.03±1.16**
G8/P2	97.58±1.59	95.70±2.07	98.07±1.43	96.45±1.70	**98.60±1.25**

Compared to the ORL database, the Yale face database has different illuminations. The experimental setting is the same as that of the ORL database. The results are illustrated in Table 3. And Figure 17 shows the recognition accuracy curves versus the variations of the size of training set. Two group of methods which preserve the local structure of samples are: tensor ST denoted by the red lines and 2D-ST denoted by green lines in Figure 17. Another group of methods which preserve the global structure of samples are denoted by blue lines. Each group includes 2 algorithms: the one with DI is denoted by a solid line and the one without DI is represented by dash-dot line. Other algorithms are denoted by other styles. From Figure 17, we can see that with smaller size of training set, the performance of the algorithms with DI is slightly worse than those of methods without DI. However, with the larger size of the training set, the performance of the algorithms with DI becomes better than those of algorithms without DI. Within these algorithms, the difference between the recognition accuracy of DTSA and TSA is the highest amongst the three groups, when the training set size is the largest. From Table 3, we can see that when the partition is *G2/P9*, the recognition accuracy of all algorithms is about 50%. When the partition is *G8/P3*, the recognition accuracy of DTSA is 80% and that of the others are about 70%.

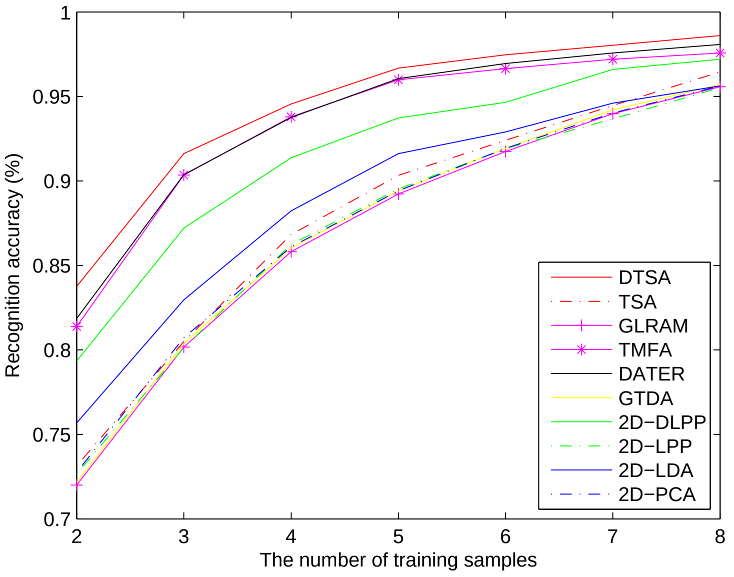


Figure 16: Recognition accuracy (%) on ORL database (mean).

Table 3: Recognition accuracy (%) on Yale database (mean)

Partitions	2D-PCA	2D-LDA	2D-LPP	2D-DLPP	GLRAM
G2/P9	52.16±3.36	51.26±4.20	53.30±4.62	46.64±5.89	51.36±3.06
G3/P8	57.83±3.25	58.10±3.76	58.65±3.05	58.70±4.83	57.03±2.79
G4/P7	60.57±3.01	62.80±4.04	62.23±3.47	63.43±4.16	59.41±3.46
G5/P6	63.56±3.58	65.07±3.41	64.22±3.39	66.16±4.07	63.04±4.05
G6/P5	65.68±3.37	58.59±3.91	66.64±3.76	70.37±4.27	65.01±4.01
G7/P4	66.87±3.91	69.80±4.36	68.00±4.03	71.57±4.48	66.83±5.00
G8/P3	69.33±4.92	72.04±5.16	70.44±5.24	74.27±4.65	68.89±4.94

Partitions	TMFA	GTDA	DATER	TSA	DTSA
G2/P9	47.87±6.41	51.66±3.68	48.06±6.00	**55.24±4.03**	52.00±5.60
G3/P8	59.92±4.41	57.82±3.20	60.45±5.12	61.48±4.12	**63.47±3.98**
G4/P7	65.70±4.33	60.76±3.62	65.68±4.52	63.58±3.49	**68.69±3.81**
G5/P6	68.69±3.82	64.62±3.88	69.13±4.07	67.22±3.42	**71.09±4.31**
G6/P5	72.05±4.12	67.47±4.08	72.19±3.83	68.32±3.63	**74.21±4.12**
G7/P4	73.77±4.49	68.97±3.51	73.90±4.11	70.33±3.79	**76.47±3.73**
G8/P3	77.42±4.54	71.69±4.83	77.24±3.80	72.31±4.76	**80.00±4.11**

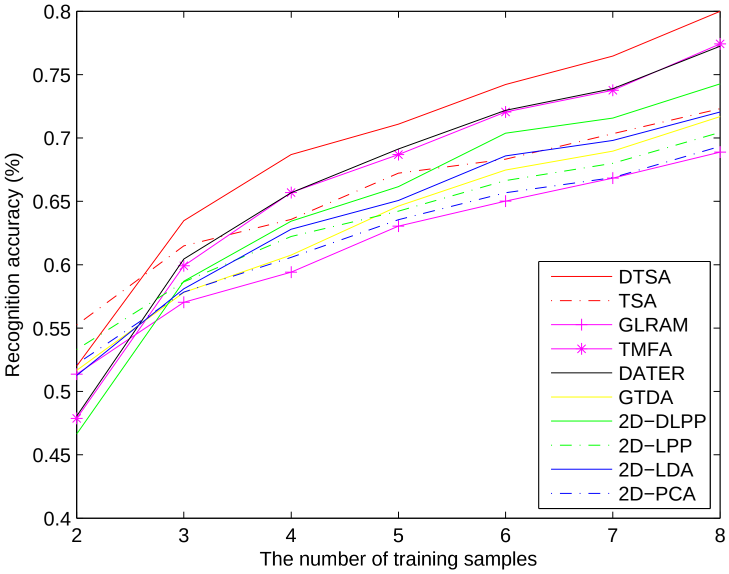


Figure 17: Recognition accuracy (%) on Yale database (mean).

Furthermore, we implemented the experiments on the more complex facial database Yale B. The results are illustrated in Table 4. Figure 18 shows that the performance of the algorithms with DI is dramatically superior to the ones without DI on this complex database. Moreover, we can see that the standard deviation of DTSA is the smallest when the partitions are *G20/P40*, *G30/P30*, *G40/P20*, *G50/P10* from Table 4. Meanwhile, when the partitions are *G30/P30*, *G40/P20*, *G50/P10* , the recognition accuracies of TMFA are higher than those of DATER.

4.7. Discussion

The experiments on three databases have been systematically performed. Their results reveal a number of interesting remarks:

1. For the tensor STs, the best performance occurs on full transformation. Meanwhile, the optimal dimension is about one-third of columns of full transformation matrix.
2. On the time complexity, DTSA is more efficient than TSA, except for cases of relatively large number of training samples, relatively large

Table 4: Recognition accuracy (%) on YaleB database (mean±std)

Partitions	2D-PCA	2D-LDA	2D-LPP	2D-DLPP	GLRAM	TMFA	GTDA	DATER	TSA	DTSA
G5/P55	30.92±1.45	55.93±1.54	31.76±1.55	60.11±2.72	30.85±1.46	67.06±2.97	30.85±1.46	67.22±2.69	29.55±1.67	**70.35±1.74**
G10/P50	44.52±0.98	70.12±1.21	45.84±1.09	74.48±1.28	44.50±1.02	75.57±1.46	44.56±1.00	78.91±1.72	44.05±1.25	**82.36±1.23**
G20/P40	58.29±1.11	80.55±0.84	59.69±1.27	84.23±0.94	57.93±1.24	84.97±0.96	57.00±1.24	86.44±1.15	58.42±1.26	**89.52±0.73**
G30/P30	66.10±1.25	85.05±0.87	66.96±1.15	88.23±0.77	65.40±1.18	91.15±0.72	65.40±1.18	89.89±0.77	66.19±1.22	**92.47±0.73**
G40/P20	70.98±1.42	87.95±0.94	71.76±1.07	90.50±0.91	70.20±1.09	93.52±0.78	70.20±1.09	91.95±0.84	71.14±1.24	**94.11±0.76**
G50/P10	75.16±2.17	90.59±1.19	75.13±1.78	92.02±1.27	73.95±1.76	95.12±0.94	73.95±1.75	93.51±0.97	74.64±1.94	**95.31±0.88**

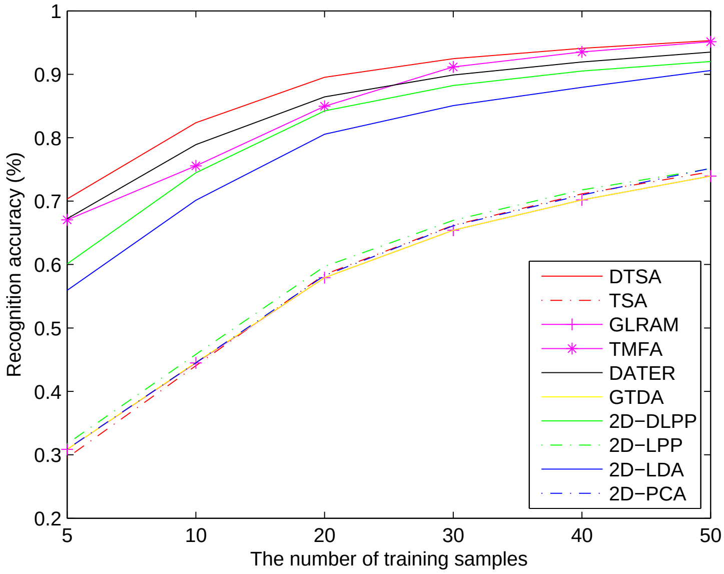


Figure 18: Recognition accuracy (%) on YaleB database (mean±std).

number of samples in every class and relatively low resolution of sample image.

3. For recognition accuracy, the methods with DI are better than the ones without DI on ORL database. The results are more evident on YaleB database where face images are more complex than those of the ORL database. On Yale database, we can also see that the methods with DI are superior to the corresponding ones without DI.

5. Conclusion

This paper presents a novel feature extraction method, Second Order Discriminant Tensor Subspace Analysis (DTSA). Unlike 2D-DLPP, DTSA employs two-sided transformations instead of single-sided transformation. D-ifferent from TSA, DTSA uses DI to increase the accuracy of recognition of facial images. Compared to the methods that preserve the global structure of samples, DTSA focuses on the local structure of samples to fit the intrinsic manifold structure of face images in high dimensional space. In this paper, we analyze its connections to 2D-DLPP and TSA, theoretically. Furthermore, DTSA benefits from three methods, i.e., tensor based methods, DI methods and local structure preserved methods. Experiments on the ORL, Yale and YaleB face database show the efficiency of the proposed method.

Our future researches will focus on extending DTSA to higher order tensor for recognition of gait and action etc.

Acknowledgements

This paper is supported by (1)the National Natural Science Foundation of China under Grant No. 60873146, 60973092, 60903097, (2)project of science and technology innovation platform of computing and software science (985 engineering), (3)the Key Laboratory for Symbol Computation and Knowledge Engineering of the National Education Ministry of China.

[1] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3 (1) (1991) 71–86.

[2] P. N. Belhumeur, J. P. Hespanha, D. J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 711–720.

[3] H. S. Seung, D. D. Lee, The manifold ways of perception, Science 290 (5500) (2000) 2268–2269.

[4] X. F. He, P. Niyogi, Locality preserving projections, in: Advances in neural information processing systems, Vol. 16, The MIT Press, 2004, pp. 153–160.

[5] X. F. He, S. C. Yan, Y. X. Hu, P. Niyogi, H. J. Zhang, Face recognition using laplacianfaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 328–340.

[6] W. W. Yu, X. L. Teng, C. Q. Liu, Face recognition using discriminant locality preserving projections, Image and Vision Computing 24 (3) (2006) 239–248.

[7] L. Yang, W. Gong, X. Gu, W. Li, Y. Liang, Null space discriminant locality preserving projections for face recognition, Neurocomputing 71 (16-18) (2008) 3644–3649.

[8] L. Zhu, S. Zhu, Face recognition based on orthogonal discriminant locality preserving projections, Neurocomputing 70 (7-9) (2007) 1543–1546.

[9] J. Gui, C. Wang, L. Zhu, Locality preserving discriminant projections, in: ICIC'09: Proceedings of the Intelligent computing 5th international conference on Emerging intelligent computing technology and applications, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 566–572.

[10] X. Zhao, X. Tian, Locality preserving fisher discriminant analysis for face recognition, in: ICIC'09: Proceedings of the 5th international conference on Emerging intelligent computing technology and applications, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 261–269.

[11] J. Yang, D. Zhang, A. F. Frangi, J. Y. Yang, Two-dimensional PCA: a new approach to appearance-based face representation and recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (1) (2004) 131–137.

[12] M. Li, B. Z. Yuan, 2D-LDA: a statistical linear discriminant analysis for image matrix, Pattern Recognition Letters 26 (5) (2005) 527–532.

[13] S. B. Chen, H. F. Zhao, M. Kong, B. Luo, 2D-LPP: a two-dimensional extension of locality preserving projections, Neurocomputing 70 (4-6) (2007) 912–921.

[14] Y. Xu, G. Feng, Y. N. Zhao, One improvement to two-dimensional locality preserving projection method for use with face recognition, Neurocomputing 73 (1-3) (2009) 245–249.

[15] W. Yu, Two-dimensional discriminant locality preserving projections for face recognition, Pattern Recognition Letters 30 (15) (2009) 1378–1383.

[16] R. C. Zhi, Q. Q. Ruan, Facial expression recognition based on two-dimensional discriminant locality preserving projections, Neurocomputing 71 (7-9) (2008) 1730–1734.

[17] E. Zhang, Y. W. Zhao, W. Xiong, Active energy image plus 2DLPP for gait recognition, Signal Processing 90 (7) (2010) 2295–2302.

[18] M. H. Wan, Z. H. Lai, J. Shao, Z. Jin, Two-dimensional local graph embedding discriminant analysis (2DLGEDA) with its application to face and palm biometrics, Neurocomputing 73 (1-3) (2009) 197–203.

[19] J. Ye, Generalized low rank approximations of matrices, Machine Learning 61 (1) (2005) 167–191.

[20] J. Liu, S. Chen, Z.-H. Zhou, X. Tan, Generalized low-rank approximations of matrices revisited, IEEE Transactions on Neural Networks 21 (4) (2010) 621–632.

[21] X. He, D. Cai, P. Niyogi, Tensor subspace analysis, in: Advances in Neural Information Processing Systems 18, MIT Press, 2005.

[22] H. Lu, K. N. Plataniotis, A. N. Venetsanopoulos, MPCA: Multilinear principal component analysis of tensor objects, IEEE Transactions on Neural Networks 19 (1) (2008) 18–39.

[23] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, H.-J. Zhang, Discriminant analysis with tensor representation, in: Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition CVPR 2005, Vol. 1, 2005, pp. 526–532.

[24] D. Tao, X. Li, X. Wu, S. J. Maybank, General tensor discriminant analysis and gabor features for gait recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (10) (2007) 1700–1715.

[25] F. Song, D. Zhang, Q. Chen, J. Wang, Face recognition based on a novel linear discriminant criterion, Pattern Analysis & Applications 10 (3) (2007) 165–174.

[26] S. C. Yan, D. Xu, B. Y. Zhang, H. J. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: A general framework for dimensionality reduction, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (1) (2007) 40–51.

[27] L. Qiao, S. Chen, X. Tan, Sparsity preserving projections with applications to face recognition, Pattern Recognition 43 (1) (2010) 331–341.

[28] M. Sun, C. Liu, J. Yang, Z. Jin, J. Yang, A two-step framework for highly nonlinear data unfolding, Neurocomputing 73 (10-12) (2010) 1801–1807.

[29] Y. Liu, Y. Liu, K. C. C. Chan, Tensor distance based multilinear locality-preserved maximum information embedding, IEEE Transactions on Neural Networks 21 (11) (2010) 1848–1854.

[30] F. R. K. Chung, Spectral Graph Theory, 1997.

[31] R. A. Horn, C. A. Johnson, Matrix Analysis, Cambridge University Press, 1985.

[32] H. Eschenauer, J. Koski, A. Osyczka, Multicriteria design optimization, Springer-Verlag, 1990.

[33] A. Georghiades, P. Belhumeur, D. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 643–660.

[34] L. van der Maaten, E. Postma, H. van den Herik, Dimensionality reduction: A comparative review, Tech. rep., Tilburg University (2009).

[35] J. Yang, C. Liu, Horizontal and vertical 2DPCA-based discriminant analysis for face verification on a large-scale database, IEEE Transactions on Information Forensics and Security 2 (4) (2007) 781–792.

[36] Z. Guan, C. Wang, Z. Chen, J. Bu, C. Chen, Efficient face recognition using tensor subspace regression, Neurocomputing 73 (13-15) (2010) 2744–2753.

[37] D. Xu, S. Yan, S. Lin, T. Huang, S. Chang, Enhancing bilinear subspace learning by element rearrangement, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (10) (2009) 1913–1920.