



## Tensor linear Laplacian discrimination (TLLD) for feature extraction

Wei Zhang<sup>a,\*</sup>, Zhouchen Lin<sup>b</sup>, Xiaoou Tang<sup>a</sup>

<sup>a</sup>Department of Information Engineering, The Chinese University of Hong Kong, Rm. 802, Ho Sin Hang Engineering Building, Shatin, Hong Kong

<sup>b</sup>Microsoft Research Asia, Sigma Building, Zhichun Road #49, Haidian District, Beijing 100190, PR China

### ARTICLE INFO

#### Article history:

Received 16 May 2008

Received in revised form 24 October 2008

Accepted 7 January 2009

#### Keywords:

Discriminant feature extraction

Tensor

Contextual distance

### ABSTRACT

Discriminant feature extraction plays a central role in pattern recognition and classification. In this paper, we propose the tensor linear Laplacian discrimination (TLLD) algorithm for extracting discriminant features from tensor data. TLLD is an extension of linear discriminant analysis (LDA) and linear Laplacian discrimination (LLD) in directions of both nonlinear subspace learning and tensor representation. Based on the contextual distance, the weights for the within-class scatters and the between-class scatter can be determined to capture the principal structure of data clusters. This makes TLLD free from the metric of the sample space, which may not be known. Moreover, unlike LLD, the parameter tuning of TLLD is very easy. Experimental results on face recognition, texture classification and handwritten digit recognition show that TLLD is effective in extracting discriminative features.

© 2009 Elsevier Ltd. All rights reserved.

### 1. Introduction

Discriminant feature extraction is an important topic in pattern recognition and classification. Principal component analysis (PCA) and linear discriminant analysis (LDA) are two traditional algorithms for linear discriminant feature extraction. Both methods involve scatters computed in the Euclidean metric, i.e., the underlying assumption is that the sample space is Euclidean. Both PCA and LDA have found wide application in pattern recognition and computer vision. For example, they are known as the famous Eigenfaces method and Fisherfaces method in face recognition [2], respectively. And many variants of LDA have shown good performance in various applications [9,12,20–22,27]. As the data manifold may not be linear, some nonlinear discriminant feature extraction algorithms, e.g., locality preserving projections (LPP) [8] and linear Laplacian discrimination (LLD) [31], have recently been developed. In addition, the kernel trick [15] is also widely applied to extend linear feature extraction algorithms to nonlinear ones by performing linear operations in a higher or even infinite dimensional space transformed by a kernel mapping function.

It is worth noting that most of the existing discriminant analysis methods are vector based, i.e., the input data are always (re)arranged in a vector form regardless of the inherent correlation among different dimensions. In practice, vector-based methods have been found

to have some intrinsic problems [26]: singularity of within-class scatter matrices, limited available projection directions and high computational cost. Much work has been done to deal with these problems [20–22,4,5]. Recently, several tensor-based methods have been proposed as alternatives to overcome these drawbacks. Tensor-based methods respect the dimensional structure of data, hence can extract better discriminant features robustly. They perform well particularly when the number of samples is relatively small, a case in which vector-based methods often suffer the singularity problem. Along this line, Ye et al.'s 2DLDA [29] and Yan et al.'s DATER [26] are the tensor extensions of the popular vector-based LDA algorithm. And tensor LPP [6,7] is an extension of LPP, also preserving local neighbor structures of tensor samples. All these methods work in tensor spaces with Euclidean metrics if metrics are to be used.

Despite the success of various subspace learning algorithms, we notice that almost all of them rely on the Euclidean assumption on the data space when computing the distance between samples, unless the appropriate metric for the data space is known, e.g., KL divergence or  $\chi^2$  distance are suitable for histogram-based data. Distance metric learning attempts to learn metrics from data. However, it has mainly focused on finding a *linear* distance metric that optimizes the data compactness and separability in a *global* sense [23,24,28]. It is computationally expensive when treating high-dimensional data, and no current nonlinear dimensionality reduction approaches can learn an explicit nonlinear metric [28]. Approximated geodesic distance [18], which attempts to estimate the distances among samples, could help alleviate, but also not resolve, the issue of metrics. For example, a slenderly distributed cluster can have large geodesic distance between the samples, which makes distance-based cluster

\* Corresponding author. Tel.: +852 3163 4327; fax: +852 2603 5032.

E-mail addresses: [wzhangee@hotmail.com](mailto:wzhangee@hotmail.com) (W. Zhang), [zhoulin@microsoft.com](mailto:zhoulin@microsoft.com) (Z. Lin), [xtang@ie.cuhk.edu.hk](mailto:xtang@ie.cuhk.edu.hk) (X. Tang).

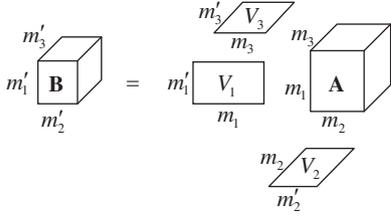


Fig. 1. Visualization of the multiplication of an  $(m_1 \times m_2 \times m_3)$ -tensor  $\mathbf{A}$  with matrices  $V_1 \in \mathbb{R}^{m_1 \times m_1}$ ,  $V_2 \in \mathbb{R}^{m_2 \times m_2}$ , and  $V_3 \in \mathbb{R}^{m_3 \times m_3}$ . This figure is adapted from [10].

analysis error-prone. Actually, what is more important is the structure of the data, rather than the absolute distance between the data samples.

From the above observations, we propose the tensor linear Laplacian discrimination (TLLD) method for nonlinear feature extraction from tensor data. TLLD could be viewed as an extension of both LDA and LLD [31] in directions of nonlinearity and tensor representation. LLD has shown its superiority of feature extraction in nonlinear spaces [31], but it still has all the abovementioned drawbacks of vector-based methods because it has the same number of available projection directions and the same null spaces of the within-class scatter matrices as LDA (the proof is in Appendix A). And although LLD has aimed at removing the metric assumption by introducing weights to the scatter matrices, nonetheless the weights are still defined as a function of the distance in the sample space. Therefore, LLD still needs the *a priori* assumption on the metric of the sample space. To further reduce the dependence on the metric of the sample space, TLLD computes the weights based on the contextual distances instead, which are measured by the *contribution to the structure* of data in the sample space. This idea is inspired by the recent work on structural perception of data [13,30]. In order to match the tensor nature of data, we further extend the vector-based coding length [13,30] to *tensor coding length* as the contextual set [30] descriptor. Another advantage of using contextual-distance-based weights is that tuning the time variable in the weights now becomes very easy by rescaling. In short, TLLD handles two kinds of structure in the sample data, the tensor structure within each individual sample and the distributional structure across all samples, in a unified way.

The rest of this paper is organized as follows. We first present TLLD in Section 2, then discuss the choice of the weights for scatter matrices in Section 3. The experimental results are presented in Section 4 and Section 5 concludes our paper.

## 2. Tensor linear Laplacian discrimination

In this section, we first give definitions of some basic tensor operations. Then we present the formulation of TLLD.

### 2.1. Preliminaries of tensor operations

An order- $n$  tensor is an element of the space  $\mathbb{R}^{m_1 \times m_2 \times \dots \times m_n}$ , where  $m_i$  ( $i = 1, 2, \dots, n$ ) are positive integers. The scalar product of tensors  $\mathbf{A}$  and  $\mathbf{B}$  with the same dimensions is  $\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{i_1=1}^{m_1} \dots \sum_{i_n=1}^{m_n} \mathbf{A}_{i_1, \dots, i_n} \mathbf{B}_{i_1, \dots, i_n}$ . The Frobenius-norm of a tensor  $\mathbf{A}$  is given by  $\|\mathbf{A}\| = \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle}$ . The  $k$ -mode product of an order- $n$  tensor  $\mathbf{A} \in \mathbb{R}^{m_1 \times m_2 \times \dots \times m_n}$  by a matrix  $V \in \mathbb{R}^{m_k \times m'_k}$ , denoted by  $\mathbf{A} \times_k V$ , is still an order- $n$  tensor whose entries are given by  $(\mathbf{A} \times_k V)_{i_1, i_2, \dots, i_{k-1}, j, i_{k+1}, \dots, i_n} = \sum_{l_k=1}^{m_k} \mathbf{A}_{i_1, i_2, \dots, i_{k-1}, l_k, i_{k+1}, \dots, i_n} V_{j, l_k}$ . Fig. 1 visualizes the equation  $\mathbf{B} = \mathbf{A} \times_1 V_1 \times_2 V_2 \times_3 V_3$  for order-3 tensors  $\mathbf{A} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$  and  $\mathbf{B} \in \mathbb{R}^{m'_1 \times m'_2 \times m'_3}$ . The mode- $k$  matrix unfolding of  $\mathbf{A}$  is denoted by  $A_{(k)} \in \mathbb{R}^{m_k \times (m_{k+1} \dots m_n m_1 \dots m_{k-1})}$  [10], where the element

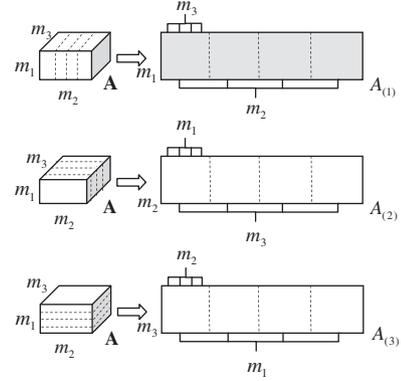


Fig. 2. Unfolding of the  $(m_1 \times m_2 \times m_3)$ -tensor  $\mathbf{A}$  to the  $(m_1 \times m_2 m_3)$ -matrix  $A_{(1)}$ , the  $(m_2 \times m_3 m_1)$ -matrix  $A_{(2)}$ , and the  $(m_3 \times m_1 m_2)$ -matrix  $A_{(3)}$  ( $m_1 = m_2 = m_3 = 4$ ). This figure is adapted from [10].

$\mathbf{A}_{i_1, \dots, i_n}$  of the original tensor appears at the  $i_k$ -th row and the  $u_k$ -th column of  $A_{(k)}$ , in which  $u_k = (i_{k+1} - 1)m_{k+2}m_{k+3} \dots m_n m_1 m_2 \dots m_{k-1} + (i_{k+2} - 1)m_{k+3} \dots m_n m_1 m_2 \dots m_{k-1} + \dots + (i_n - 1)m_1 m_2 \dots m_{k-1} + (i_1 - 1)m_2 m_3 \dots m_{k-1} + (i_2 - 1)m_3 \dots m_{k-1} + \dots + i_{k-1}$ . An illustration of an order-3 tensor's matrix unfolding is shown in Fig. 2. And the  $k$ -mode product in tensor notation  $\mathbf{B} = \mathbf{A} \times_k V$  can be expressed in terms of matrix unfolding:  $B_{(k)} = V A_{(k)}$ .

### 2.2. Discriminant scatters

The strategy of LDA and LLD for discriminative feature extraction is to simultaneously minimize the within-class variance and maximize the between-class variance of low-dimensional features after projections. We follow this strategy but with tensor representation.

Let the samples in order- $n$  tensor representation be  $\mathbf{X}_i$ ,  $i = 1, 2, \dots, N$ , where  $N$  is the total number of samples. And let  $s_i$  be the label of  $\mathbf{X}_i$  and  $N_s$  be the number of samples in the  $s$ -th class. The total number of classes is  $c$ . Our objective is to find a group of orthogonal projection matrices  $U_k \in \mathbb{R}^{m_k \times m'_k}$  ( $m'_k < m_k$ ),  $k = 1, 2, \dots, n$ , such that the projected low-dimensional tensors

$$\mathbf{Y}_i = \mathbf{X}_i \times_1 U_1^T \times_2 U_2^T \dots \times_n U_n^T, \quad i = 1, 2, \dots, N \quad (1)$$

have minimal within-class variance and maximal between-class variance.

Following LLD, it is natural to define the within-class scatter as follows:

$$\alpha = \sum_{s=1}^c \sum_{\mathbf{X}_i \in \Omega_s} w_i \|\mathbf{Y}_i - \bar{\mathbf{Y}}^s\|^2, \quad (2)$$

where  $\bar{\mathbf{Y}}^s = (1/N_s) \sum_{\mathbf{X}_i \in \Omega_s} \mathbf{Y}_i$  is the centroid of the  $s$ -th projected class,  $\Omega_s = \{\mathbf{X}_i | s_i = s\}$  is the set of the  $s$ -th class, and  $w_i$  is the weight for the  $i$ -th sample. Similarly, the between-class scatter is defined as

$$\beta = \sum_{s=1}^c w^s N_s \|\bar{\mathbf{Y}}^s - \bar{\mathbf{Y}}\|^2, \quad (3)$$

where  $\bar{\mathbf{Y}} = (1/N) \sum_{\mathbf{X}_i \in \Omega} \mathbf{Y}_i$  is the centroid of all the projected samples,  $\Omega = \{\mathbf{X}_i, i = 1, 2, \dots, N\}$  is the set of samples and  $w^s$  is the weight for the  $s$ -th class. The choice of  $w_i$  and  $w^s$  will be presented in Section 3. Our goal is to find orthogonal projection matrices  $U_k$ , such that  $\alpha$  is minimized and at the same time  $\beta$  is maximized. So also adopting Fisher's criterion, we may solve

$$\arg \max_{U_1, U_2, \dots, U_n} \frac{\beta}{\alpha}. \quad (4)$$

**Table 1**

Tensor linear Laplacian discrimination (TLLD) algorithm.

Given the sample set  $\Omega = \{\mathbf{X}_i \in \mathbb{R}^{m_1 \times m_2 \times \dots \times m_n} \mid i = 1, 2, \dots, N\}$ , their class labels  $s_i \in \{1, 2, \dots, c\}$ , and the target dimensions  $(m'_1, m'_2, \dots, m'_n)$  of features.

1. Compute the weights  $w_i, i = 1, 2, \dots, N$ , and  $w^s, s = 1, 2, \dots, c$ , and initialize  $U_k(0) = (I_{m'_k \times m'_k}, \mathbf{0}_{m'_k \times (m_k - m'_k)})^T (k = 1, 2, \dots, n)$ .
2. For  $l = 1 : L_{max}$  do
  - For  $k = 1 : n$  do
    - (a)  $\mathbf{Z}_i = \mathbf{X}_i \times_1 U_1^T(l) \dots \times_{k-1} U_{k-1}^T(l) \times_{k+1} U_{k+1}^T(l-1) \dots \times_n U_n^T(l-1)$ , and compute the matrix unfolding  $(Z_i)_{(k)}, i = 1, 2, \dots, N$ ;
    - (b)  $S_w^{(k)} = \sum_{i=1}^N w_i (Z_i - \bar{Z}^i)_{(k)} (Z_i - \bar{Z}^i)_{(k)}^T$ ,  $S_b^{(k)} = \sum_{s=1}^c w^s N_s (\bar{Z}^s - \bar{Z})_{(k)} (\bar{Z}^s - \bar{Z})_{(k)}^T$ ;
    - (c) Solve the trace-ratio problem (8) for  $U_k(l)$ .
    - (d) If  $\|U_k(l) - U_k(l-1)\| < m_k m'_k \epsilon, \forall k$ , and  $l > 1$ , break;
3. Output the projection matrices  $U_k = U_k(l) \in \mathbb{R}^{m_k \times m'_k}, k = 1, 2, \dots, n$ .

### 2.3. Solving for projection matrices

It is hard to solve (4) for  $U_i (i = 1, 2, \dots, n)$  simultaneously. So we turn to iteration methods. As [6], we extend the matrix-based deduction therein to tensors to reformulate  $\alpha$  and  $\beta$  by mode- $k$  unfolding:

$$\begin{aligned}
 \alpha &= \sum_{i=1}^N w_i \|\mathbf{Y}_i - \bar{\mathbf{Y}}^i\|^2 \\
 &= \sum_{i=1}^N w_i \text{tr}[(Y_i - \bar{Y}^i)_{(k)} (Y_i - \bar{Y}^i)_{(k)}^T] \\
 &= \sum_{i=1}^N w_i \text{tr}[U_k^T (Z_i - \bar{Z}^i)_{(k)} (Z_i - \bar{Z}^i)_{(k)}^T U_k] \\
 &= \text{tr} \left\{ U_k^T \left[ \sum_{i=1}^N w_i (Z_i - \bar{Z}^i)_{(k)} (Z_i - \bar{Z}^i)_{(k)}^T \right] U_k \right\}, \quad (5)
 \end{aligned}$$

where  $\mathbf{Z}_i = \mathbf{X}_i \times_1 U_1^T \times_2 U_2^T \dots \times_{k-1} U_{k-1}^T \times_{k+1} U_{k+1}^T \dots \times_n U_n^T$ , and  $(Z_i - \bar{Z}^i)_{(k)}$  is the mode- $k$  matrix unfolding of  $\mathbf{Z}_i - \bar{\mathbf{Z}}^i$ , in which  $\bar{\mathbf{Z}}^i$  is the centroid of  $\{\mathbf{Z}_j \mid s_j = s_i\}$ . And

$$\begin{aligned}
 \beta &= \sum_{s=1}^c w^s N_s \|\bar{\mathbf{Y}}^s - \bar{\mathbf{Y}}\|^2 \\
 &= \sum_{s=1}^c w^s N_s \text{tr}[(\bar{Y}^s - \bar{Y})_{(k)} (\bar{Y}^s - \bar{Y})_{(k)}^T] \\
 &= \sum_{s=1}^c w^s N_s \text{tr}[U_k^T (\bar{Z}^s - \bar{Z})_{(k)} (\bar{Z}^s - \bar{Z})_{(k)}^T U_k] \\
 &= \text{tr} \left\{ U_k^T \left[ \sum_{s=1}^c w^s N_s (\bar{Z}^s - \bar{Z})_{(k)} (\bar{Z}^s - \bar{Z})_{(k)}^T \right] U_k \right\}, \quad (6)
 \end{aligned}$$

where  $\bar{\mathbf{Z}}$  is the centroid of all  $\mathbf{Z}_i$ 's.

So we arrive at the forms of the within-class scatter and the between-class scatter under mode- $k$  unfolding:

$$\alpha = \text{tr}(U_k^T S_w^{(k)} U_k) \quad \text{and} \quad \beta = \text{tr}(U_k^T S_b^{(k)} U_k), \quad (7)$$

where  $S_w^{(k)} = \sum_{i=1}^N w_i (Z_i - \bar{Z}^i)_{(k)} (Z_i - \bar{Z}^i)_{(k)}^T$  is the mode- $k$  within-class scatter matrix and  $S_b^{(k)} = \sum_{s=1}^c w^s N_s (\bar{Z}^s - \bar{Z})_{(k)} (\bar{Z}^s - \bar{Z})_{(k)}^T$  is the mode- $k$  between-class scatter matrix.

Therefore, we may solve

$$\arg \max_{U_k} \frac{\beta}{\alpha} = \frac{\text{tr}(U_k^T S_b^{(k)} U_k)}{\text{tr}(U_k^T S_w^{(k)} U_k)} \quad (8)$$

for  $U_k$  successively, by fixing the rest  $U_i$ 's to prepare  $S_b^{(k)}$  and  $S_w^{(k)}$ , and repeat this procedure until convergence. For each  $k$ , this trace-ratio problem can be efficiently solved by the algorithm proposed by Wang et al. [19].

As we can only obtain a locally optimal solution, the initialization is important. Following previous literatures, e.g. [7, 11, 26], we initialize  $U_k$  as  $(I_{m'_k \times m'_k}, \mathbf{0}_{m'_k \times (m_k - m'_k)})^T$ , which we call the identity initialization. We test such initialization on real and synthetic data. We choose four datasets (From FRGC v2, CMU PIE, USC SIPI and MNIST database, respectively. The details are given in Section 4.) and randomly select 10 subjects and five images per subject in each dataset. We find that for every randomly chosen target dimensions  $m'_1 \times m'_2 \times \dots \times m'_n$ , the objective function values resulting from the identity initialization are always almost as good as the best ones resulting from 50 times of random initialization. The same observation persists on randomly generated synthetic datasets. Moreover, the experiments in Section 4 show that TLLD with the identity initialization also achieve good recognition results. So identity initialization is recommended.

Finally, we summarize the above procedure in Table 1. Note that the target dimensions should satisfy:  $1 \leq m'_k \leq \min\{m_k, (c-1) \prod_{i \neq k} m_i\}$  [26].

### 3. Definition of weights

In this section, we discuss how to choose the weights  $w_i$  and  $w^s$  so as to make the TLLD algorithm complete.

Motivated by LLD [31] and Laplacian Eigenmap [3], we define the weights in the following forms:

$$\begin{cases} w_i = \exp\left(-\frac{d^2(\mathbf{X}_i, \Omega_{s_i})}{t}\right), & i = 1, 2, \dots, N, \\ w^s = \exp\left(-\frac{d^2(\Omega_s, \Omega)}{t}\right), & s = 1, 2, \dots, c, \end{cases} \quad (9)$$

where  $d(\cdot, \cdot)$  is some distance,  $t$  is the time variable and  $s_i$  is the class label of  $\mathbf{X}_i$ .

In LLD, the weights are simply related to the distances to the centroids, using the metric of the sample space:

$$\begin{cases} w_i = \exp\left(-\frac{\|\mathbf{X}_i - \bar{\mathbf{X}}^{s_i}\|_{\mathbb{S}}^2}{t}\right), & i = 1, 2, \dots, N, \\ w^s = \exp\left(-\frac{\|\bar{\mathbf{X}}^s - \bar{\mathbf{X}}\|_{\mathbb{S}}^2}{t}\right), & s = 1, 2, \dots, c. \end{cases} \quad (10)$$

For example, the authors used  $\chi^2$  distance because they worked on histogram-based data. Such a definition of weights has two problems: (1) the metric of the original sample space may be unknown and (2) the Euclidean centroids of samples may not lie on the data manifold, hence the metric of the sample space cannot be applied

to compute the distance between samples and the centroids. So we should define the weights in another way.

### 3.1. Contextual distance

As we have argued in the Introduction, a better definition for the weights should be based on the structure of the data, rather than the absolute distances among the samples. Inspired by the recent work on structural perception of data [13,30], we deem that contextual-distance-based definition of the weights should be good choice. According to [30], contextual distances are defined on the contextual set  $X$  (the set of nearest neighbors) of a sample  $x$ . It is related to the contribution of the samples to the structural integrity of the contextual set, which is depicted by a structural descriptor  $f$  (could be either scalar or vector valued). As the descriptor  $f(X)$  is supposed to be the intrinsic structural characterization of the set  $X$ , if  $x$  complies with the structure of  $X$ , then removing  $x$  from  $X$  will not affect the structure much. In contrast, if  $x$  is an outlier or a noise sample, then removing  $x$  from  $X$  will change the structure significantly. The contribution of  $x$  to the structure of  $X$  is thus measured by

$$\delta f = f(X) - f(X \setminus \{x\}). \quad (11)$$

So we may define the distance from  $x$  to  $X$  as

$$d(x, X) = \|\delta f\| = \|f(X) - f(X \setminus \{x\})\|. \quad (12)$$

The generalization to the distance between two sets is straightforward.

From the above analysis, it becomes natural to define

$$\begin{aligned} d(\mathbf{X}_i, \Omega_{s_i}) &= \|f(\Omega_{s_i}) - f(\Omega_{s_i} \setminus \{\mathbf{X}_i\})\|, \\ d(\Omega_s, \Omega) &= \|f(\Omega) - f(\Omega \setminus \Omega_s)\|, \end{aligned} \quad (13)$$

for the weights in (9), using the idea of checking the structural variation.

### 3.2. Tensor coding length

To employ the contextual-distance-based weights, we still have to find an appropriate structural descriptor. In [30], two descriptors were introduced: centroid and coding length [13,30]. The centroid descriptor is defined as

$$f(\Omega) = \frac{1}{|\Omega|} \sum_{\mathbf{x} \in \Omega} \mathbf{x},$$

where  $|\Omega|$  denotes the cardinality of  $\Omega$ . The coding length descriptor is  $f(\Omega) = L(\Omega)$ , where  $L(\Omega)$  is the minimal number of bits to encode the data in  $\Omega$ , up to a tolerable distortion  $\varepsilon$  (see Appendix B for the expression of  $L(\Omega)$ ).

Unfortunately, neither of the above two existing descriptors is suitable for TLLD. This is because the centroid descriptor inherently assumes an Euclidean sample space, while the current formulation of coding length is vector based. To match the tensor nature of TLLD, we propose the tensor coding length.

We first mode- $k$  unfold each tensor to a matrix and then compute the mode- $k$  coding length of the set of columns of these matrices:

$$L_{(k)}(\mathbf{X}) = L(\{(X_1)_{(k)}, (X_2)_{(k)}, \dots, (X_N)_{(k)}\}), \quad (14)$$

where  $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N\}$  and  $(X_i)_{(k)}$  is the mode- $k$  matrix unfolding of  $\mathbf{X}_i$ . Then the tensor coding length of  $\mathbf{X}$  is defined as the following vector:

$$\mathbf{L}(\mathbf{X}) = [L_{(1)}(\mathbf{X}), L_{(2)}(\mathbf{X}), \dots, L_{(n)}(\mathbf{X})]^T. \quad (15)$$

**Table 2**

Metrics we adopt for different features in experiments.

| Feature type                          | Metric             |
|---------------------------------------|--------------------|
| Raw facial images                     | Euclidean distance |
| Locally binary pattern (LBP) features | $\chi^2$ distance  |
| Gabor features                        | Euclidean distance |
| Features extracted by projections     | Euclidean distance |

To compute the tensor coding length, we empirically choose the tolerable distortion as:  $\varepsilon = \sqrt{10(\sum_{k=1}^n m_k)^{n-1}/N^n \prod_{k=1}^n m_k}$ .

Now the only issue left is to determine the parameter  $t$  in (9). In LLD, this time parameter is hard to tune in order to achieve the optimal performance: it may range from 0.01 to 500 in [31] and from  $10^{-5}$  to  $10^3$  in our experiments (see Section 4). In our TLLD, we simply rescale  $t$  as:  $t = t' \sigma_w$  for  $w_i$  and  $t = t' \sigma_b$  for  $w^s$ , respectively, where  $\sigma_w = (1/N) \sum_{i=1}^N d^2(\mathbf{X}_i, \Omega_{s_i})$  and  $\sigma_b = (1/N) \sum_{s=1}^c N_s d^2(\Omega_s, \Omega)$ , and have found that the optimal  $t'$  is usually around 1. This treatment easily waives the parameter tuning on  $t$ . Note that such a rescaling trick does not work for LLD. This is because LLD uses the Euclidean distance, which can have a much larger distance variance than using the contextual distance (e.g., the neighboring points are in a long-thin area), while rescaling is effective only when the distance variance is relatively small.

## 4. Experimental results

To evaluate our TLLD algorithm, we perform experiments on facial databases (FRGC version 2 [17] and CMU PIE<sup>1</sup>), texture database (USC SIPI from the Brodatz album<sup>2</sup>) and handwritten digit database (MNIST<sup>3</sup>). We compare TLLD with PCA, LDA, LLD, TLDA (DATER [26]) and tensor LPP [7].<sup>4</sup> We also replace the weights of TLLD with (10) using the assumed metric of the sample space so as to verify the necessity of using contextual-distance-based weights. This version of “TLLD” is denoted as TLLD-0. To test on different kinds of data spaces, we use the raw facial images for face recognition, the locally binary pattern (LBP) [16] features for texture classification, and the Gabor features for handwritten digit recognition, as the input data of the abovementioned methods, respectively. The most commonly used metrics for these features are listed in Table 2. The nearest-neighbor classifier is adopted for all the experiments, where the metric used to compute the distance between the output features of these methods is dependent on the nature of the features. For example, when histogram features are directly used for classification,  $\chi^2$  distance will be used by the classifier. However, if the histogram features are projected to low-dimensional spaces by the abovementioned methods before feeding the classifier, Euclidean distance will be used instead. All parameters of the involved methods are tuned on the training set, by the full search over a relatively wide range which is discretized by some stepsize. In Section 4.3, we also compare the running time between tensor-based methods and vector-based methods, in which the difference is the most drastic.

### 4.1. Face recognition

One of the most natural forms of tensor data are raw images. In this experiment, two benchmark face databases, experiment 4 in FRGC version 2 and CMU PIE, are used. For FRGC v2, we search all images of each person in the query set and take the first 10 images

<sup>1</sup> [http://www.ri.cmu.edu/projects/project\\_418.html](http://www.ri.cmu.edu/projects/project_418.html)

<sup>2</sup> <http://sipi.usc.edu/database/database.cgi?volume=textures>

<sup>3</sup> <http://yann.lecun.com/exdb/mnist/>

<sup>4</sup> Tensor LPP was designed for matrices only but the data in the second and the third experiments are both order-3 tensors, so we only test it in the first experiment.



**Fig. 3.** The results are averaged over six target dimensions and 50 random splits. (a) Sample facial images of one person from FRGC v2. Images are  $36 \times 32$  pixels in size. (b) Sample facial images of one person from CMU PIE database. Images are  $32 \times 32$  pixels in size.

**Table 3**  
Face recognition results on FRGC v2 and CMU PIE databases.

| Database   | FRGC v2     |                                 | CMU PIE     |                                 |
|------------|-------------|---------------------------------|-------------|---------------------------------|
|            | Err (%)     | Dim                             | Err (%)     | Dim                             |
| PCA        | 9.76        | 70                              | 30.16       | 135                             |
| PCA + LDA  | 9.05        | 90                              | 6.58        | 25                              |
| PCA + LLD  | 8.81        | 50 ( $t = 5$ )                  | 6.58        | 25 ( $t = 1000$ )               |
| TLDA       | 8.10        | $20 \times 4$                   | 4.54        | $10 \times 5$                   |
| Tensor LPP | 7.86        | $25 \times 13$                  | 4.54        | $10 \times 5$                   |
| TLLD-0     | 7.86        | $10 \times 9$ ( $t = 10^{-5}$ ) | 4.31        | $10 \times 5$ ( $t = 10^{-4}$ ) |
| TLLD       | <b>7.38</b> | $15 \times 4$ ( $t' = 1$ )      | <b>4.08</b> | $10 \times 5$ ( $t' = 1$ )      |

Err and Dim denote recognition error rate and reduced feature dimensions, respectively. The time parameters for LLD, TLLD-0 and TLLD are also listed in the brackets. Dim are the optimal reduced dimensions for the corresponding method.

if the number of facial images is more than 10. Thus we collect 600 facial images of the first 60 subjects for our experiment. All the images are aligned according to the positions of eyes and mouths, and then cropped to a size of  $36 \times 32$  (top row of Fig. 3). The CMU PIE database contains more than 40,000 facial images of 68 people. The images were acquired in different poses, under various illumination conditions and with different facial expressions. In this experiment, a subset, five near frontal poses (C27, C05, C29, C09 and C07) under two illumination conditions (indexed as 08 and 11) of 63 people, is used. So each person has 10 images and in total 630 images are collected. All the images are aligned by fixing the locations of eyes, and then normalized to  $32 \times 32$  pixels (bottom row of Fig. 3). We randomly select three images of each person for the training set and gallery set and the rest images are used for querying.

Table 3 shows the recognition results.<sup>5</sup> One can see tensor approaches outperform vector-based methods, and TLLD is the best among them. TLLD is also better than TLLD-0 which assumes Euclidean distance between the facial images. Moreover, one can also see that the time variable for LLD varies dramatically, while TLLD does not require careful tuning on  $t'$ .

#### 4.2. Texture classification

Another suitable application of TLLD is features based on local regions. Such features can be organized in tensor structures, with their spatial information accounting for two dimensions of the tensor. And such features often lie in nonlinear spaces. We choose USC-SIPI image database as the data for texture classification. The image data are comprised of 13 textures from the Brodatz album shown in Fig. 4. For each texture,  $512 \times 512$  images digitized at six different rotation angles ( $0^\circ$ ,  $30^\circ$ ,  $60^\circ$ ,  $90^\circ$ ,  $120^\circ$ , and  $150^\circ$ ) are included. The images are divided into 16 disjoint  $128 \times 128$  subimages. So there are 1248 samples in total, each of the 13 classes having 96 samples. We randomly select 20 samples in each class as the training set and the gallery set, while the others are used for testing.

<sup>5</sup> In this experiment, LDA and LLD have to work with PCA because otherwise the within-class scatter matrices will be singular.

In this experiment, we use  $LBP_8^{u2}$  [16] on a  $3 \times 3$  grid<sup>6</sup> (i.e., the set of  $LBP_8^{u2}$ 's separately computed on the nine evenly partitioned subimages) as the input of the methods to be tested. So the input data are order-3 tensors. The results are shown in Table 4. We see that the vector-based methods perform much worse than tensor-based methods. This is because texture images are globally homogeneous. So LBP histograms on different grids are highly correlated, therefore simple vectorization of such features can result in a highly singular total-class scatter matrix. Even applying PCA does not help much because in this case PCA can only extract very little discriminative information from the complement of the null space of the total-class scatter matrix. This is the cause of the poor performance of vector-based methods. Again, one can see that TLLD outperforms all other methods and its optimal time parameter is still 1.

#### 4.3. Handwritten digit recognition

A third application of TLLD is dimensionality reduction on multi-resolution images, which are general purpose features for computer vision and image processing and have been very successful in many applications. The most popular multi-resolution operator is the Gabor filter. It has been frequently used in texture analysis [1], face recognition [26] and digit recognition [25]. We perform experiments on the MNIST handwritten digit database of 60,000 training samples and 10,000 testing samples. All images are  $28 \times 28$  grayscale images. We choose the first 20 images of each digit to compose a subdatabase Fig. 5. For each class, the first five samples are selected for training, and the remaining 15 images are used for testing. We extract 24 Gabor features in four different scales and six different directions as did in [14] and down-sample them to  $7 \times 7$  images. Then we get order-3 tensor features of size  $24 \times 7 \times 7$ .

To our best knowledge, we are unaware of any research reporting what the optimal metric for Gabor features is. So we have to assume Euclidean distance between the original order-3 Gabor feature tensors, so that baseline can be computed and LLD and TLLD-0 can be applied. The results are shown in Table 5. Almost the same conclusions from face recognition and texture classification can be drawn. And TLLD again shows its advantage of metric independence: TLLD performs better than TLLD-0 which blindly assumes Euclidean distances.

We also present the training time of tensor-based methods and their vector-based counterparts in Table 6. One can see that TLDA, TLLD-0 and TLLD are much faster than LDA and LLD. This testifies to the speed advantage of tensor-based methods.

<sup>6</sup> In the original paper by Ojala et al. [16], the images are not partitioned. We partition the images just for constructing tensor input data. Note that the purpose of experiments in this paper is to test the discriminative power of different methods, rather than proposing better features for specific tasks, it is harmless to do so.

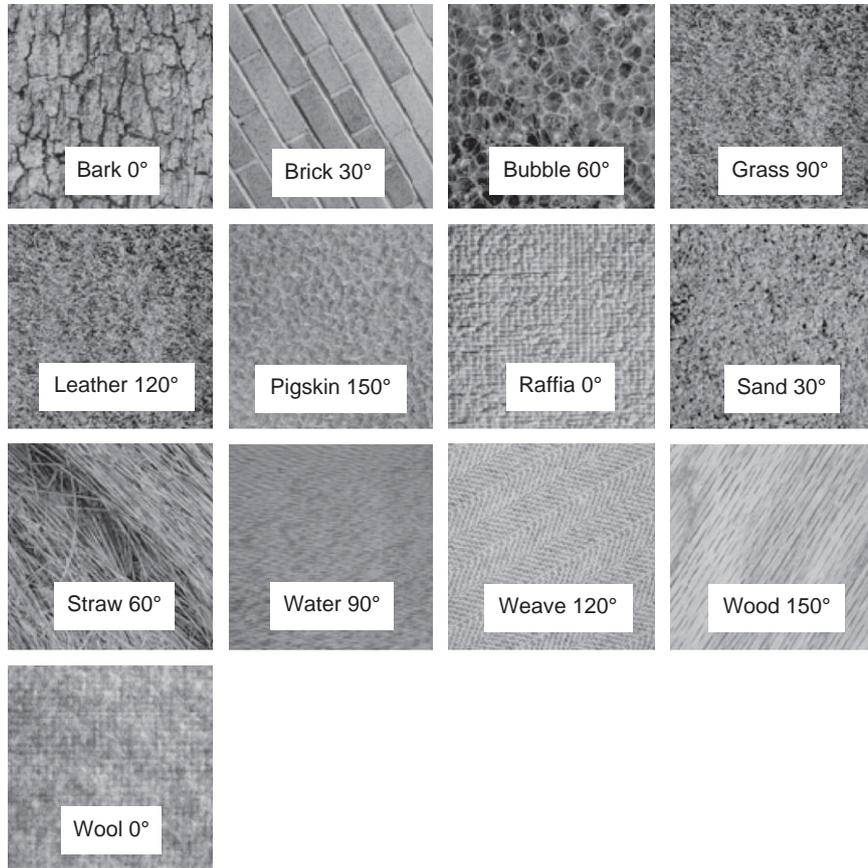


Fig. 4. Texture images in particular rotation angles. Each texture was digitized at six angles: 0°, 30°, 60°, 90°, 120°, and 150°. Images are 512 × 512 pixels in size.

Table 4  
Comparison of texture classification error rates.

| Method                            | Err (%)     | Dim              |
|-----------------------------------|-------------|------------------|
| Baseline                          | 19.54       | 531 (59 × 3 × 3) |
| PCA                               | 17.51       | 10               |
| LDA                               | 25.40       | 15               |
| PCA + LDA                         | 13.97       | 150              |
| LLD ( $t = 5 \times 10^{-5}$ )    | 23.58       | 57               |
| PCA + LLD ( $t = 500$ )           | 14.27       | 190              |
| TLDA                              | 3.24        | 57 × 1 × 1       |
| TLLD-0 ( $t = 5 \times 10^{-6}$ ) | 1.92        | 30 × 1 × 1       |
| TLLD ( $t' = 1$ )                 | <b>1.62</b> | 53 × 1 × 1       |

Using the original 3 × 3 grid  $LBP_8^{pi2}$  directly without dimensionality reduction is the baseline. Dim are the optimal reduced dimensions for the corresponding method.

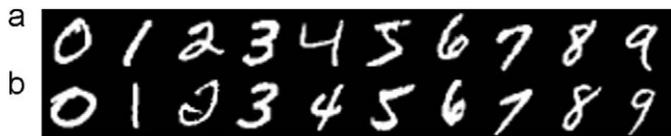


Fig. 5. Samples of handwritten digits from MNIST database. Images are 28 × 28 pixels in size. (a) Samples from the training set. (b) Samples from the testing set.

### 5. Conclusions

In this paper, a novel algorithm named TLLD is proposed for extracting discriminative features from tensor data. Contextual-distance-based weighting mechanism enables TLLD to work effectively without assuming an *a priori* metric for the tensor space. Experiments on different tasks have proven the superiority of TLLD,

Table 5  
Recognition results of handwritten digits.

| Method                   | Err (%)      | Dim               |
|--------------------------|--------------|-------------------|
| Baseline                 | 22.67        | 1176 (24 × 7 × 7) |
| PCA                      | 22.67        | 35                |
| LDA                      | 26.67        | 20                |
| PCA + LDA                | 17.33        | 25                |
| LLD ( $t = 10^{-2}$ )    | 25.33        | 20                |
| PCA + LLD ( $t = 1$ )    | 16.67        | 15                |
| TLDA                     | 17.33        | 6 × 4 × 7         |
| TLLD-0 ( $t = 10^{-1}$ ) | 17.33        | 6 × 4 × 7         |
| TLLD ( $t' = 1$ )        | <b>16.00</b> | 12 × 2 × 2        |

Using the Gabor features directly without dimensionality reduction is the baseline. Dim are the optimal reduced dimensions for the corresponding method.

Table 6  
The training time of vector-based methods and tensor-based methods on MNIST database.

| Method   | Vector-based |       | Tensor-based |        |       |
|----------|--------------|-------|--------------|--------|-------|
|          | LDA          | LLD   | TLDA         | TLLD-0 | TLLD  |
| Time (s) | 10.35        | 10.28 | 0.354        | 0.317  | 0.336 |

The time is averaged on all possible choice of target dimensions.

including higher discriminative power, metric independence, and easy parameter tuning.

As the features extracted by TLLD are also tensors, we expect that the recognition results could be further improved if TLLD cooperates with tensor-oriented classifiers. And it is possible that some contextual distances other than the tensor coding length can result in

even better performance of TLLD. It is attractive to explore in both directions.

## Acknowledgment

The first author would like to thank Deli Zhao for valuable discussions.

## Appendix A. The relationship between LDA and LLD

In this appendix, we show that LDA and LLD have the same null space of the within-class scatter matrices and the same number of available projection directions.

In LLD, two scatter matrices, called within-class and between-class scatter matrices, can be written as follows:

$$D_w = H_w W_w H_w^T \quad \text{and} \quad D_b = H_b W_b H_b^T, \quad (\text{A.1})$$

where  $W_w = \text{diag}(w_1, w_2, \dots, w_N)$ ,  $W_b = \text{diag}(w^{s_1}, w^{s_2}, \dots, w^{s_N})$ ,  $H_w$  is the data matrix and  $H_b$  is the class mean matrix. The class means and the global mean of the data have been subtracted from the  $H_w$  and  $H_b$  matrices, respectively [31]. In LDA, the two scatter matrices can be written as

$$S_w = H_w H_w^T, \quad S_b = H_b H_b^T. \quad (\text{A.2})$$

Perform singular value decomposition on  $H_w$ :

$$H_w = P A Q^T, \quad (\text{A.3})$$

where  $P^T P = P P^T = I$  and  $Q^T Q = Q Q^T = I$ . So we have  $P^T S_w P = A A^T$  and  $P^T D_w P = A Q^T W_w Q A^T$ . Note that  $W_w$ ,  $P$  and  $Q$  are full-rank matrices ( $w_i > 0$ ). So  $\text{rank}(D_w) = \text{rank}(P^T D_w P) = \text{rank}(A Q^T W_w Q A^T) = \text{rank}(W_w^{1/2} Q A^T A Q^T W_w^{1/2}) = \text{rank}(A^T A) = \text{rank}(A A^T) = \text{rank}(P^T S_w P) = \text{rank}(S_w)$ . Meanwhile, for any right eigenvector  $p$  of  $S_w$  associated with the zero eigenvalue,  $p$  is also a right eigenvector of  $H_w^T$  with the zero eigenvalue. Thus  $D_w p = H_w W_w H_w^T p = 0$ , i.e.,  $p$  is a right eigenvector of  $D_w$  with the zero eigenvalue. So the within-class scatter matrix of LLD have the same null space as that of LDA.

Similarly,  $\text{rank}(S_b) = \text{rank}(D_b)$ . As the number of available projection directions is dependent on the ranks of the within-class and the between-class scatter matrices, LLD and LDA have the same number of available projection directions.

From the above analysis, we can conclude that LLD cannot avoid the drawbacks of most vector-based subspace learning algorithms, such as singularity, curse of dimensionality and limit of available projection directions.

## Appendix B. Coding length

Coding length was introduced by Ma et al. [13] to computer vision and pattern recognition. It is defined on vector sets. For a vector set  $X = \{x_1, x_2, \dots, x_K\}$ , we center each point as  $\tilde{x}_i = x_i - \bar{x}$ , where  $\bar{x} = (1/K) \sum_{i=1}^K x_i$ , and denote  $\tilde{X} = [\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_K]$ . Then the coding length of  $X$  is

$$L(X) = \frac{K+m}{2} \log_2 \det \left( I + \frac{m}{\varepsilon^2 K} \tilde{X} \tilde{X}^T \right) + \frac{m}{2} \log_2 \left( 1 + \frac{\bar{x}^T \bar{x}}{\varepsilon^2} \right), \quad (\text{B.1})$$

where  $\varepsilon$  is the allowable distortion, which could be empirically chosen as  $\varepsilon = \sqrt{10m/(K-1)}$ , and  $m$  is the dimension of vectors.

## References

[1] S. Arivazhagan, L. Ganesan, S. Priyal, Texture classification using Gabor wavelets based rotation invariant features, *Pattern Recognition Letters* 27 (16) (2006) 1976–1982.

[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] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Computation* 15 (2003) 1373–1396.

[4] J. Friedman, Regularized discriminant analysis, *Journal of American Statistical Association* 84 (405) (1989) 165–175.

[5] T. Hastie, A. Buja, R. Tibshirani, Penalized discriminant analysis, *The Annals of Statistics* 23 (1) (1995) 73–102.

[6] X. He, D. Cai, H. Liu, J. Han, Image clustering with tensor representation, in: *MULTIMEDIA'05: Proceedings of the 13th Annual ACM International Conference on Multimedia*, ACM, New York, NY, USA, 2005.

[7] X. He, D. Cai, P. Niyogi, Tensor subspace analysis, in: *Advances in Neural Information Processing Systems*, 2005.

[8] X. He, P. Niyogi, Locality preserving projections, in: *Advances in Neural Information Processing Systems*, 2003.

[9] P. Howland, H. Park, Generalizing discriminant analysis using the generalized singular value decomposition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (8) (2004) 995–1006.

[10] L.D. Lathauwer, B.D. Moor, J. Vandewalle, A multilinear singular value decomposition, *SIAM Journal on Matrix Analysis and Applications* 21 (4) (2000) 1253–1278.

[11] X. Li, S. Lin, S. Yan, D. Xu, Discriminant locally linear embedding with high-order tensor data, *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics* 38 (2) (2008).

[12] C. Liu, Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (5) (2006) 725–737.

[13] Y. Ma, H. Derksen, W. Hong, J. Wright, Segmentation of multivariate mixed data via lossy data coding and compression, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (9) (2007) 1546–1562.

[14] B.S. Manjunath, W.Y. Ma, Texture features for browsing and retrieval of image data, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18 (8) (1996) 837–842.

[15] K. Muller, S. Mika, G. Rietsch, K. Tsuda, B. Scholkopf, An introduction to kernel-based learning algorithms, *IEEE Transactions on Neural Networks* 12 (2001) 181–201.

[16] T. Ojala, M. Pietikainen, T. Maenpaa, Gray scale and rotation invariant texture classification with local binary patterns, in: *European Conference on Computer Vision*, 2000.

[17] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek, Overview of the face recognition grand challenge, in: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.

[18] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* 290 (2000) 2319–2323.

[19] H. Wang, S. Yan, D. Xu, X. Tang, T. Huang, Trace ratio vs. ratio trace for dimensionality reduction, in: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2007.

[20] X. Wang, X. Tang, Dual-space linear discriminant analysis for face recognition, in: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2004.

[21] X. Wang, X. Tang, A unified framework for subspace face recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (9) (2004) 1222–1228.

[22] X. Wang, X. Tang, Random sampling for subspace face recognition, *International Journal of Computer Vision* (2006) 91–104.

[23] K.Q. Weinberger, J. Blitzer, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, in: *Advances in Neural Information Processing Systems*, 2005.

[24] E.P. Xing, A.Y. Ng, M.I. Jordan, S. Russell, Distance metric learning, with application to clustering with side-information, in: *Advances in Neural Information Processing Systems*, 2002.

[25] D. Xu, S. Yan, L. Zhang, H.-J. Zhang, Z. Liu, H.-Y. Shum, Concurrent subspaces analysis, in: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.

[26] S. Yan, D. Xu, Q. Yang, L. Zhang, H.-J. Zhang, Discriminant analysis with tensor representation, in: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.

[27] J. Yang, A. Frangi, J. Yang, D. Zhang, Z. Jin, KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2) (2005) 230–244.

[28] L. Yang, R. Jin. *Distance Metric Learning: A Comprehensive Survey*. Technical Report, Department of Computer Science and Engineering, Michigan State University, 2006.

[29] J. Ye, R. Janardan, Q. Li, Two-dimensional linear discriminant analysis, in: *Advances in Neural Information Processing Systems*, 2004.

[30] D. Zhao, Z. Lin, X. Tang, Contextual distance for data perception, in: *Proceedings of the International Conference on Computer Vision*, 2007.

[31] D. Zhao, Z. Lin, R. Xiao, X. Tang, Linear Laplacian discrimination for feature extraction, in: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2007.

**About the Author**—WEI ZHANG received the Bachelor degree from Tsinghua University in 2007. He is currently an MPhil-PhD stream student in the Department of Information Engineering, the Chinese University of Hong Kong. His research interests include pattern recognition and statistical learning.

**About the Author**—ZHOUCHEN LIN received the Ph.D. degree in applied mathematics from Peking University in 2000. He is currently a researcher in Visual Computing Group, Microsoft Research Asia. His research interests include computer vision, computer graphics, pattern recognition, statistical learning, document processing, and human computer interaction.

**About the Author**—XIAOOU TANG received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, in 1996. He is a professor and the director of Multimedia Lab in the Department of Information Engineering, the Chinese University of Hong Kong. He is an associate editor of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI). His research interests include computer vision, pattern recognition, and video processing.