A novel face recognition method based on sub-pattern and tensor

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

[a] *College of Computer Science and Technology, Jilin University, Changchun 130012, China*

[b] *Raytheon BBN technologies, Boston, MA, 02138, USA*

Abstract

This paper aims to address one of many problems existing in current facial recognition techniques using tensor (TensorFace Algorithm and its extensions). Current methods rasterize facial images as vectors, which result in a loss of spatial structure information. In this paper, we propose a method called Sp-Tensor to extend TensorFace by applying the sub-pattern technique. Advantages of the proposed method include: (1) a portion of spatial structure and local information of facial images is preserved; (2) dramatically reduce the computation complexity than other existing methods when building the model. The experimental results demonstrate that Sp-Tensor has better performance than the original TensorFace and Sp-PCA1, especially for facial images with un-modeled views and light conditions.

Keywords: Face recognition, Tensor subspace analysis, Sub-pattern technique, Multilinear analysis

1. Introduction

Over the past decades, extensive researches on face recognition have been carried out in both the field of pattern recognition and artificial intelligence and is widely used in many applications, including public securities, crime and terrorist detections, etc. Normally, a 2D facial image is represented

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

as a feature point in a high dimensional feature space and its perceptually structure can be characterized by using a small set of meaningful parameters. Thus, dimensionality reduction techniques are commonly used before recognition.

PCA[1] is a widely used linear dimensionality reduction method which maximizes the variance of projected features in the projective subspace. LDA[2] encodes discriminant information by maximizing the ratio between the inter-class and intra-class scatters. Different to LDA, max-min distance analysis (MMDA)[3] duly considers the separation of all class pairs. To deal with general case of data distribution, Bian and Tao [3] also extended MMDA to kernel MMDA (KMMDA). To overcome the non-smooth max-min optimization problem with orthonormal constraints which is introduced by MMDA/KMMDA, they developed a sequential convex relaxation algorithm to solve it approximately. Zhang *et al.* [4] proposed a patch alignment framework, which consists of two stages: part optimization and whole alignment. The framework reveals that 1) algorithms are intrinsically different in the patch optimization stage and 2) all algorithms share an almost identical whole alignment stage. As an application of this framework, they developed a new dimensionality reduction algorithm, namely Discriminative Locality Alignment (DLA), by imposing discriminative information into the part optimization stage. DLA can 1) attack the distribution nonlinearity of measurements; 2) preserve the discriminative ability; and 3) avoid the small-sample-size problem. Li and Tao[5] proposed a simple exponential family PCA (SePCA) to employ exponential family distributions to handle general types of observations. The method also automatically discovers the number of essential principal components by using Bayesian inference. Zhou *et al.* [6] used elastic net to find the optimal sparse solution of the dimensionality reduction algorithm which is based on manifold learning. However, these methods consider a 2D facial image as a vector. This results in the loss of spatial structure information of the facial images.

A 2D gray facial image is naturally represented by a 2nd-order tensor. Wang *et al.* [7] proposed a Discriminant Tensor Subspace Analysis (DTSA) algorithm to extract discriminant features from the intrinsic manifold structure of the 2nd-order tensor. They also treated a color facial image as a 3rd-order tensor and proposed a tensor discriminant color space (TDCS) model[8] to seek an optimal color space for face recognition. Mu *et al.*[9] treated a gray facial image as a 3rd-order tensor by using biologically inspired features. They also employed a similar idea to recognize gait[10].

Despite the achievements, facial recognition is still quite challenging due to multiple factors which cannot be predicted or controlled, such as different user identities, various user postures, facial expressions and varying lights. These variations can be modeled as a tensor, which is a multidimensional array, using multilinear analysis. A high order tensor can construct a multilinear structure to model the multiple factors of facial variation[11][12][13]. Each factor is arranged along a certain *mode* (dimension) of the tensor. A-mongst all these factors, "user identities" is the key for facial recognition. Hence, TensorFaces were proposed by Vasilescu *et al.* and were successfully used for face recognition [14][15][16][17].

Recently, a great deal of interests are aroused in the field of face recognition by using tensor[18][19][20][21][22][23]. Gao and Tian[18][21] utilized a combination of multilinear and non-linear view manifold to present a multi-view face recognition algorithm by improving on a TensorFace based method. In order to overcome the difficulty of obtaining a complete training tensor, Geng *et al.*[19] proposed a M^2SA method which can work on a training tensor with massive missing values. Rana *et al.*[22][20] presented the MPCA-JS method by exploiting the interaction of all subspaces resulting from multilinear decomposition. This method not only offered the flexibility to handle facial images at un-modeled lights or views, but also speeded up the recognition process. Park *et al.*[23] proposed a novel tensor approach based on an individual-modeling and nonlinear mapping method to deal with the difficulty in factoring the unknown parameters of a new test image, and solve the person-identity parameter by tensor-based face recognition. However, all these methods rasterize facial images as vectors. This leads to the loss of some spatial structure information of the image.

To address the problem, the techniques of reorganizing input variables of images have been extensively studied over the last decade. From the perspective of rearranging the pixels of images, these methods can be divided into two main categories. In one category, the image is treated as a matrix instead of a vector. 2D-PCA[24] employs single-sided transformation of the image matrix (only transformation of the columns of the image matrix), while $(2D)^2$PCA[25][26] employs two-sided transformation (both columns and rows). MatFLDA[27] encodes the discriminant information into 2D-PCA to enhance the recognition accuracy. In the other category, an image is partitioned into equally sized, non-overlapping blocks called *sub-pattern.* Sp-PCA[28] applies PCA on the set of sub-patterns directly. UPCA[29] unifies the methods into a single framework and illustrates that the traditional

PCA, Sp-PCA and 2D-PCA are three specific forms of UPCA determined by the number of sub-patterns k, which is a tradeoff between accuracy and stability of the estimation of covariance. It also shows that the traditional PCA and 2D-PCA are two specific forms of Sp-PCA.

Higher-order SVD (HOSVD) is regarded as a generalization of traditional PCA or SVD for higher-order tensors. Having established a clear relationship between HOSVD and traditional PCA, it is only natural to extend Sp-PCA from vector to tensor, which improves its performance on multiple factors affecting facial variation. In this paper, we introduce the sub-pattern technique into TensorFace to build a new model named *Sp-Tensor* (Sub-pattern Tensor) and propose a recognition method using Sp-Tensor. The merits of oufr model and method are:

1. A portion of spatial structure information is preserved and local information of facial images is utilized not only in building the model, but in the recognition process as well.
2. Existing sub-pattern methods of traditional PCA are extended to a higher-order tensor to present the multiple factors of facial variation by multilinear analysis or HOSVD;
3. Dramatically reduce the computation complexity than the existing methods when building a model;

The rest of this paper is organized as follows: in Section 2, the definitions related to tensor and higher-order SVD are described. The sub-pattern technique along with the building of Sp-Tensor model is introduced in Section 3; Moreover, in the same section, a recognition method using the proposed model is covered and the details of extension from two existing TensorFace based recognition methods to Sp-Tensor are discussed. In Section 4, the experimental results are reported and analyzed. The conclusions and our future work are discussed in Section 5.

2. Tensor Fundamentals

A tensor is a multidimensional array. More formally, a Nth-order tensor is an element of the tensor product of N vector spaces, each of which has its own coordinate system. It is higher-order generalization of scalar (0th-order tensor), vector (1st-order tensor), and matrix (2nd-order tensor). In this paper, lowercase italic letters (a, b, ...) denote scalars, bold lowercase letters

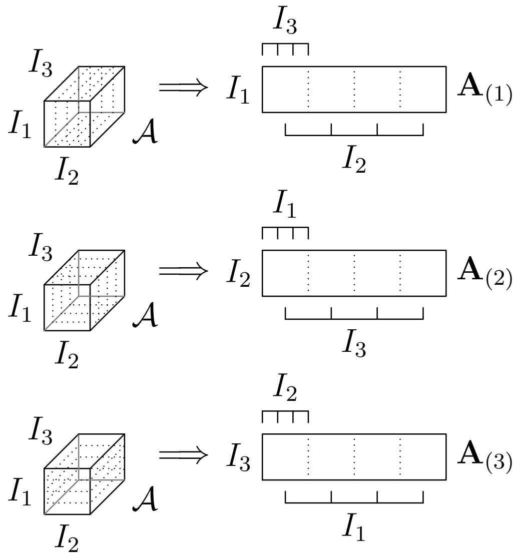


Figure 1: Unfolding of the $(I_1 \times I_2 \times I_3)$-tensor $\mathcal{A}$ to the $(I_1 \times I_2 I_3)$-matrix $\mathbf{A}_{(1)}$, the $(I_2 \times I_3 I_1)$-matrix $\mathbf{A}_{(2)}$, and the $(I_3 \times I_1 I_2)$-matrix $\mathbf{A}_{(3)}$ $(I_1 = I_2 = I_3 = 4)$.

(**a, b, ..**) denote vectors, bold uppercase letters (**A, B, ...**) denote matrices, and calligraphic uppercase letters ($\mathcal{A}$, $\mathcal{B}$, ...) denote tensors.

The *order* of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is N. An element of $\mathcal{A}$ is denoted by $\mathcal{A}_{i_1 i_2 \ldots i_N}$ or $a_{i_1 i_2 \ldots i_N}$, where $1 \leq i_n \leq I_n$, $n = 1, 2, \ldots, N$. The *mode-n fibers* of $\mathcal{A}$ are the I_n-dimensional vectors obtained from $\mathcal{A}$ by fixing every index but index i_n. The *scalar product* of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, denoted by $\langle \mathcal{A}, \mathcal{B} \rangle$, is defined in a straightforward way as $\langle \mathcal{A}, \mathcal{B} \rangle \overset{\text{def}}{=} \sum_{i_1} \sum_{i_1} \ldots \sum_{i_N} a_{i_1 i_2 \ldots i_N} b_{i_1 i_2 \ldots i_N}$. The *Frobenius norm* of a tensor $\mathcal{A}$ is then defined as $\|\mathcal{A}\| \overset{\text{def}}{=} \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$. We introduce the following definitions[30][31] which are relevant to this paper.

Definition 1. The *mode-n unfolding matrix* of a Nth-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, is a matrix $\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times I_{\overline{n}}}$, which is the ensemble of vectors in $\mathbb{R}^{I_n}$ obtained by keeping index i_n fixed and varying the other indices. Here, $I_{\overline{n}} = (\prod_{i \neq n} I_i)$. We denote the mode-n unfolding matrix of $\mathcal{A}$ as $\mathbf{A}_{(n)}$.

Fig. 1 shows an example of pictorial description of a third order tensor.

Definition 2. The *mode-n product* of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ by a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$, denoted by $\mathcal{A} \times_n \mathbf{U}$, is an $(I_1 \times I_2 \times \ldots \times I_{n-1} \times J_n \times I_{n+1} \times \ldots \times I_N)$-

tensor of which the entries are given by:

$$(\mathcal{A} \times_n \mathbf{U})_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} \overset{\text{def}}{=} \sum_{i_n} a_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} u_{j_n i_n}. \tag{1}$$

The mode-n product of tensor and matrix can be expressed in terms of unfolding matrices for simplicity purposes.

$$(\mathcal{A} \times_n \mathbf{U})_{(n)} = \mathbf{U} \cdot \mathbf{A}_{(n)} \tag{2}$$

Given a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and matrices $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}, \mathbf{V} \in \mathbb{R}^{J_m \times I_m}$, the mode-n product of tensor and matrix satisfies:

$$(\mathcal{A} \times_n \mathbf{U}) \times_m \mathbf{V} = (\mathcal{A} \times_m \mathbf{V}) \times_n \mathbf{U} = \mathcal{A} \times_n \mathbf{U} \times_m \mathbf{V} \tag{3}$$

By taking the standard multilinear algebra, any tensor $\mathcal{D} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ can be decomposed by the HOSVD [11] as

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \ldots \times_n \mathbf{U}_n \ldots \times_N \mathbf{U}_N \tag{4}$$

where $\mathbf{U}_n (n = 1, 2, \ldots, N)$, is an orthonormal *mode matrix* and contains the ordered principal components for the nth mode. $\mathcal{Z}$ is the *core tensor*. The decomposition is proceeded as:

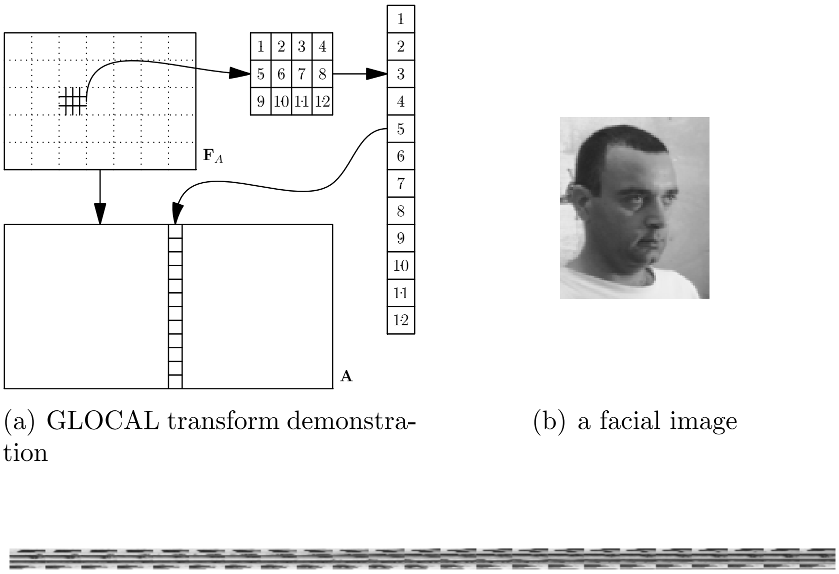
1. For $n = 1, \ldots, N$, compute matrix $\mathbf{U}_n$ in Eq.(4) by computing the SVD of the unfolding matrix $\mathbf{D}_{(n)}$ and setting $\mathbf{U}_n$ to be the left matrix of the SVD.
2. Solve the core tensor as follows:

$$\mathcal{Z} = \mathcal{D} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \ldots \times_n \mathbf{U}_n^T \ldots \times_N \mathbf{U}_N^T \tag{5}$$

3. Proposed approaches

3.1. Image partition for constructing sub-pattern image

In this paper, GLOCAL transform[32] is adopted to obtain the sub-pattern image. Suppose that the size of each facial image F_A is $w \times h$. To derive its sub-pattern image or its sub-pattern matrix $\mathbf{A}$, each facial image is initially partitioned into I_{pat} equally sized, non-overlapping sub-patterns. Then each column of $\mathbf{A}$ can be obtained by raster-scanning a sub-pattern into a vector with dimensions of $I_{pix} = (w \times h)/I_{pat}$. In this way, a $I_{pat} \times I_{pix}$



(a) GLOCAL transform demonstration

(b) a facial image

(c) responding sub-pattern image

Figure 2: GLOCAL transform

sub-pattern image **A** is obtained for each facial image. Note that the processing of sub-patterns also follows the same raster-scan order. Fig. 2 shows a demonstration of a GLOCAL transform (Fig. 2(a)), a facial image (Fig. 2(b)) and its corresponding sub-pattern image (Fig. 2(c)). Depending on the sequences of scanning the sub-patterns and the pixels in each sub-pattern, there are four possible combinations for generating a sub-pattern image. We choose to scan by row for both sub-patterns and pixels in each sub-pattern. Nevertheless, it is easy to verify that the sub-pattern images derived from different scanning orders can be transformed into one another by row or column permutations. That is, they are equivalent up to a multiplication of some suitable permutation matrices. The GLOCAL transform can be generalized from matrices to tensors. We can transform the tensor space from $\mathbb{R}^{w\times h}$ to $\mathbb{R}^{I_{pat}\times I_{pix}}$.

3.2. Sub-pattern based tensor (Sp-Tensor)

The Sp-Tensor is a multilinear extension of the subpattern-based PCA[28]. Given an ensemble of sub-pattern face images which is resulted from the confluence of multiple factors related to identity, view, light, and facial expression, a training tensor $\mathcal{D} \in \mathbb{R}^{I_i \times I_v \times I_l \times I_e \times I_{pat} \times I_{pix}}$ can be constructed, where I_i, I_v, I_l, I_e, I_{pat} and I_{pix} denote the number of identities, views, lights, expressions, sub-patterns and pixels in a sub-pattern, respectively. HOSVD is used to factorize identity, view, light, expression information, *etc.* as

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_i \times_2 \mathbf{U}_v \times_3 \mathbf{U}_l \times_4 \mathbf{U}_e \times_5 \mathbf{U}_{pat} \times_6 \mathbf{U}_{pix} \tag{6}$$

where the *core tensor* $\mathcal{Z} \in \mathbb{R}^{I'_i \times I'_v \times I'_l \times I'_e \times I'_{pat} \times I'_{pix}}$ governs the interaction among the factors represented in the 6 mode matrices. The mode matrices $\mathbf{U}_i \in \mathbb{R}^{I_i \times I'_i}$, $\mathbf{U}_v \in \mathbb{R}^{I_v \times I'_v}$, $\mathbf{U}_l \in \mathbb{R}^{I_l \times I'_l}$ and $\mathbf{U}_e \in \mathbb{R}^{I_e \times I'_e}$ span the parameter space of various identities, views, lights, and expressions, respectively. The mode matrices $\mathbf{U}_{pat} \in \mathbb{R}^{I_{pat} \times I'_{pat}}$ and $\mathbf{U}_{pix} \in \mathbb{R}^{I_{pix} \times I'_{pix}}$ constitute the space of sub-pattern eigenimages. The i'-th row of $\mathbf{U}_i$, denoted by $\mathbf{u}_i^{(i')}$, is the coefficient vector of identity i'. The v'-th row of $\mathbf{U}_v$, denoted by $\mathbf{u}_v^{(v')}$, is the coefficient vector of view v'. The l'-th row of $\mathbf{U}_l$, denoted by $\mathbf{u}_l^{(l')}$, is the coefficient vector of light l'. And the e'-th row of $\mathbf{U}_e$, denoted by $\mathbf{u}_e^{(e')}$, is the coefficient vector of expression e'.

The great merit of Sp-Tensor beyond the existing techniques of reorganizing variables for images is that Sp-Tensor explicitly represents how various factors interact to produce facial images. Here, the various factors include identity, view, light and expression. This merit contributes more to the existing techniques of recognizing variables for images than Tensorface. On the other hand, comparing with Tensorface, the merit of Sp-Tensor is that sub-pattern image represents the interaction between mode-5 and mode-6 factors.

From Eq.(6), a training image with the identity i', the view v'. the light l' and the expression e' is denoted as

$$\mathcal{D}^{(i',v',l',e')} = \mathcal{Z} \times_1 \mathbf{u}_i^{(i')} \times_2 \mathbf{u}_v^{(v')} \times_3 \mathbf{u}_l^{(l')} \times_4 \mathbf{u}_e^{(e')} \times_5 \mathbf{U}_{pat} \times_6 \mathbf{U}_{pix} \tag{7}$$

where $\mathcal{D}^{(i',v',l',e')}$ is a $1 \times 1 \times 1 \times 1 \times I_{pat} \times I_{pix}$ tensor. We reconstruct facial images by fixing v', l', e', to various i' and selecting the first 1, 5, 10, 50, 100, 200, 250, 300, 322 eigenvectors from $\mathbf{U}_{pix}$, which are illustrated in Fig. 3. In the same way, various lights, views and expressions with the increasing number of eigenvectors are shown in Fig. 4, Fig. 5 and Fig. 6, respectively. From these four figures, we can see there is very little difference between images in the first columns of Fig. 3, Fig. 5 and Fig. 6. However, the images

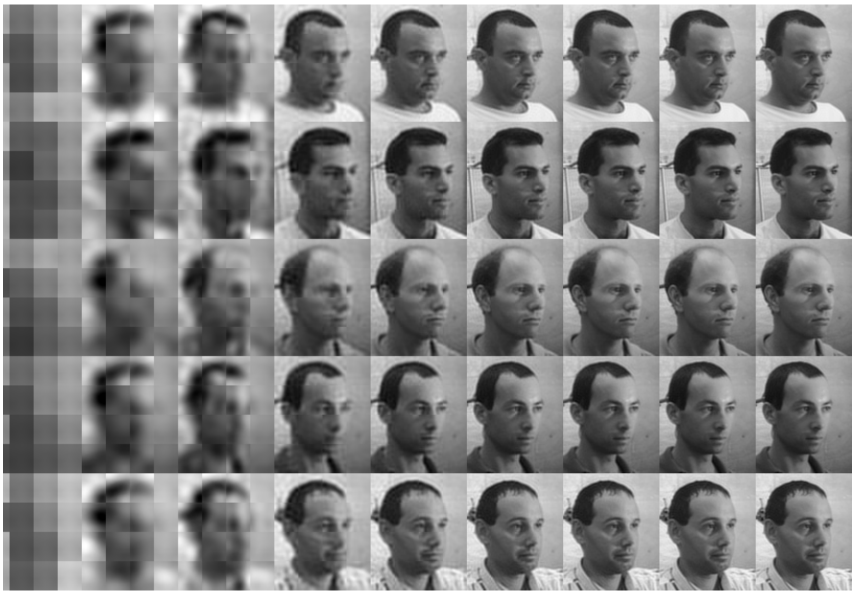


Figure 3: various people with the increasing number of eigenvectors

in the first columns of Fig. 4 show a great difference between the images due to various lights. This indicates that the light has the greatest contribution. Then, we can sort the factors according to their influences in descending order as such: light, view, identity, and expression.

In order to reduce the effect of light variation, the mean values are subtracted from each of mode-*pix* fibers of $\mathcal{D}$ before decomposing $\mathcal{D}$ as Eq.(6). We called it *Spm-Tensor* to distinguish it from *Sp-Tensor*. However, the term Sp-Tensor is used for both Sp-Tesor and Spm-Tensor if removal of mean values from each mode-*pix* fibers is not necessary.

According to Eq.(3), Eq.(6) can be transformed as follows:

$$\begin{aligned}\mathcal{D} &= (\mathcal{Z} \times_2 \mathbf{U}_v \times_3 \mathbf{U}_l \times_4 \mathbf{U}_e \times_5 \mathbf{U}_{pat} \times_6 \mathbf{U}_{pix}) \times_1 \mathbf{U}_i \\ &= \mathcal{B} \times_1 \mathbf{U}_i\end{aligned} \tag{8}$$

Every person in the training tensor can be represented by a single I_i' vector, which contains coefficients with respect to the bases comprising the tensor $\mathcal{B}$.

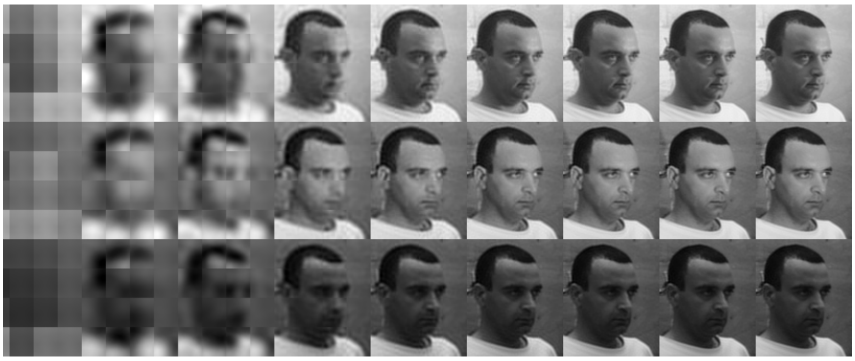


Figure 4: various lights with the increasing number of eigenvectors

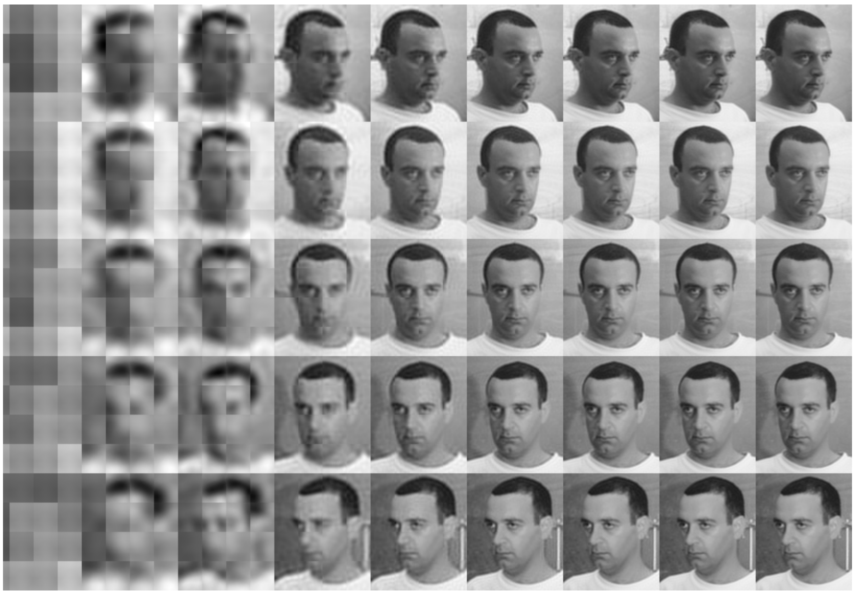


Figure 5: various views with the increasing number of eigenvectors

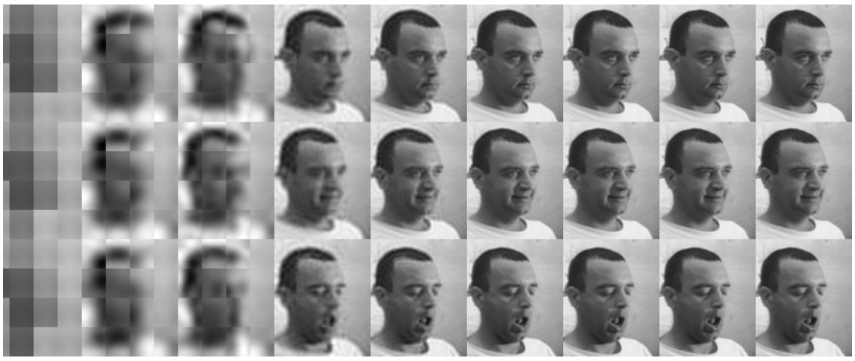


Figure 6: various expressions with the increasing number of eigenvectors

3.3. Recognition Using Sub-pattern based tensor

First, we will introduce two formal definitions before the discussion of recognition using Sp-Tensor. As an extension of *fibers*, we define *mode-n hyperslices* in the following way:

Definition 3. The *mode-n hyperslices* of N-order $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ are a set of (N-*1*)-order $\mathcal{A}_{::\ldots i_n \ldots ::} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_{n-1} \times I_{n+1} \times \ldots \times I_N}$ obtained from $\mathcal{A}$ by fixing index i_n and varying other indices, denoted as

$$\mathfrak{S}_{(I_n)}(\mathcal{A}) = \{\mathcal{A}_{::\ldots 1 \ldots ::}, \mathcal{A}_{::\ldots 2 \ldots ::}, \ldots, \mathcal{A}_{::\ldots I_n \ldots ::}\} \tag{9}$$

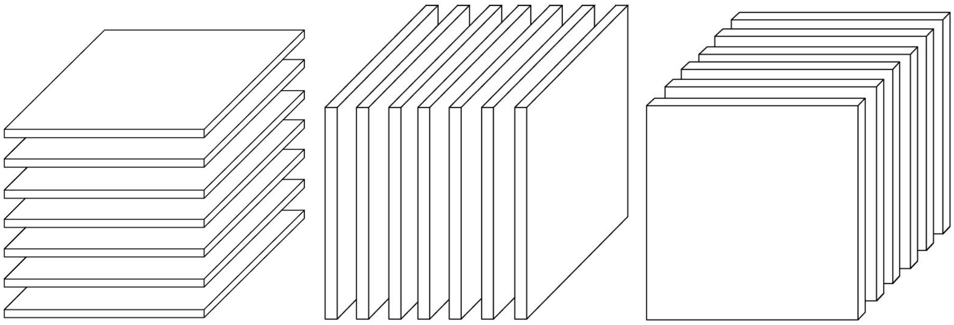
The mode-1 hyperslices, mode-2 hyperslices and mode-3 hyperslices of a 3rd-order tensor $\mathcal{X}$ are denoted as $\mathbf{X}_{i::}, \mathbf{X}_{:j:}$ and $\mathbf{X}_{::k}$, respectively. See Fig. 7 for more details. The *mode-n hyperslices* for mode-n product satisfies the following properties.

Property 1. Given the tensor $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_{n-1} \times J_n \times I_{n+1} \times \ldots \times I_N}$, the matrices $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$ and $\mathcal{B} = \mathcal{A} \times_n \mathbf{U}$, one has

$$\mathcal{B}_{::\ldots i_n \ldots ::} = \mathcal{A}_{::\ldots i_n \ldots ::} \times_n \mathbf{U} \tag{10}$$

and

$$\mathcal{B}_{::\ldots i_n \ldots ::} = \mathcal{A}_{::\ldots i_n \ldots ::} \times_n \mathbf{u}_n \tag{11}$$



(a) hyperslices $\mathbf{X}_{i::}$ (b) hyperslices $\mathbf{X}_{:j:}$ (c) hyperslices $\mathbf{X}_{::k}$

Figure 7: Hyperslices of a 3rd-order tensor

where

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \dots \\ \mathbf{u}_{J_n} \end{bmatrix}$$

Applying *hyperslices* operator along I_v, I_l, I_e on $\mathcal{D}$ and $\mathcal{B}$ in Eq.(8), respectively, we get

$$\mathfrak{S}_{(I_v)}(\mathfrak{S}_{(I_l)}(\mathfrak{S}_{(I_e)}(\mathcal{D}))) = \{\mathcal{D}_{:v'l'e'::}\} \tag{12}$$

$$\mathfrak{S}_{(I_v)}(\mathfrak{S}_{(I_l)}(\mathfrak{S}_{(I_e)}(\mathcal{B}))) = \{\mathcal{B}_{:v'l'e'::}\} \tag{13}$$

where

$$v' = 1, \dots, I_v; \quad l' = 1, \dots, I_l; \quad e' = 1, \dots, I_e$$

$\mathcal{B}_{:v'l'e'::} \in \mathbb{R}^{I_i \times 1 \times 1 \times 1 \times I_{pat} \times I_{pix}}$ is the basis tensor for a given view v', light l', and expression e'. According to Eq.(8), we obtain

$$\mathcal{D}_{:v'l'e'::} = \mathcal{B}_{:v'l'e'::} \times_1 \mathbf{U}_i \tag{14}$$

where $\mathcal{D}_{:v'l'e'::}, \mathcal{B}_{:v'l'e'::} \in \mathbb{R}^{I_i \times 1 \times 1 \times 1 \times I_{pat} \times I_{pix}}$, $\mathbf{U}_i \in \mathbb{R}^{I_i \times I'_i}$. We apply *hyperslices* operator along I_i on $\mathcal{D}_{:v'l'e'::}$ and $\mathcal{B}_{:v'l'e'::}$ in Eq.(14), According to Eq.(10), we obtain

$$\mathcal{D}_{i'v'l'e'::} = \mathcal{B}_{i'v'l'e'::} \times_1 \mathbf{u}_i^{(i')}, i' = 1, 2, \dots I_i \tag{15}$$

where $\mathcal{D}_{i'v'l'e'::}, \mathcal{B}_{i'v'l'e'::} \in \mathbb{R}^{1 \times 1 \times 1 \times 1 \times I_{pat} \times I_{pix}}$, $\mathbf{u}_i^{(i')}$,the coefficient for each person *i'*, is the *i'*th row vector extracted from the matrix $\mathbf{U}_i$. Given a test image, a $1 \times 1 \times 1 \times 1 \times I_{pat} \times I_{pix}$ tensor $\mathcal{D}_{test}$ can be constructed. The person's identity can be determined by the following criterion:

$$\underset{i',v',l',e'}{\arg} \min \|\mathcal{D}_{test} - \mathcal{D}_{i'v'l'e'::}\| \tag{16}$$

In Tensorface[14], the personal identity of a test image is obtained according to:

$$\underset{i'}{\arg} \min \|\mathbf{u}_i^{(i')} - \mathbf{c}_{test}\| \tag{17}$$

where $\mathbf{c}_{test}$ is the candidate coefficient vector obtained by computing all combinations of view, light and expression. Identity is determined by finding the value of i' that yields the minimum Euclidean distance between $\mathbf{c}_{test}$ and the vectors $\mathbf{u}_i^{(i')}$. Comparing the above equation with Eq. (16), one can see that the proposed method uses the Frobenius norm of a tensor rather to measure the distance between two high order tensors than the norm of a vector to determine a person's identity. But in TensorFace, only the distance between two vectors is used to determine a test personal identity. Obviously, a high order tensor contains more spatial structure information than a vector. (see [30][33]). As a result, a portion of the spatial structure information is preserved. To show the effects of the spatial structure information preserved through the proposed method for facial recognition, we extend the existing MPCA-LV and MPCA-JS[20] to sub-patterns in section 3.4. Experiments are then conducted to compare Sp-Tensor with the two extended methods as well as MPCA-LV and MPCA-JS in section 4.5.

3.4. Sp-Tensor-LV and Sp-Tensor-JS

Santu *et al.*[20] proposed a new optimization framework that unifies some of the existing tensor-based methods for face recognition into a common mathematical basis. They regarded tensor face recognition as a multilinear least square problem and showed four different ways to solve this optimization problem. We extend two of them to sub-pattern tensor face recognition.

For a test image $\mathcal{D}_{test}$, we have,

$$\mathcal{D}_{test} = \mathcal{Z} \times_1 \mathbf{u}_i \times_2 \mathbf{u}_v \times_3 \mathbf{u}_l \times_4 \mathbf{u}_e \times_5 \mathbf{U}_{pat} \times_6 \mathbf{U}_{pix} \tag{18}$$

where $\mathbf{u}_i$, $\mathbf{u}_v$, $\mathbf{u}_l$ and $\mathbf{u}_e$ are identity-space, view-space, light-space and expression-space projections,respectively. For a test image $\mathcal{D}_{test}$, we need to calculate $\mathbf{u}_i$, $\mathbf{u}_v$, $\mathbf{u}_l$ and $\mathbf{u}_e$. This can be formulated as,

$$\underset{\mathbf{u}_i,\mathbf{u}_v,\mathbf{u}_l,\mathbf{u}_e}{\arg} \min \|\mathcal{D}_{test} - \mathcal{Z} \times_1 \mathbf{u}_i \times_2 \mathbf{u}_v \times_3 \mathbf{u}_l \times_4 \mathbf{u}_e \times_5 \mathbf{U}_{pat} \times_6 \mathbf{U}_{pix}\| \tag{19}$$

Although we only need to obtain $\mathbf{u}_i$, the vectors $\mathbf{u}_v$, $\mathbf{u}_l$ and $\mathbf{u}_e$ still need to be calculated. To reduce the computational cost, a fixed set $\{(\mathbf{u}_v, \mathbf{u}_l, \mathbf{u}_e)\}$ is restructured, its size is $I_v \times I_l \times I_e$. Let $(\mathbf{u}_v^{(v')}, \mathbf{u}_l^{(l')}, \mathbf{u}_e^{(e')}) \in \{(\mathbf{u}_v, \mathbf{u}_l, \mathbf{u}_e)\}$, Eq.(19) can be rewritten as:

$$\arg\min_{\mathbf{u}_i} \|\mathcal{D}_{test} - \mathcal{Z} \times_1 \mathbf{u}_i \times_2 \mathbf{u}_v^{(v')} \times_3 \mathbf{u}_l^{(l')} \times_4 \mathbf{u}_e^{(e')} \times_5 \mathbf{U}_{pat} \times_6 \mathbf{U}_{pix}\| \qquad (20)$$

From Definition 1, the above equation can be rewritten as:

$$\arg\min_{\mathbf{u}_i} \|(\mathcal{D}_{test})_{(i)} - \mathbf{u}_i \times (\mathcal{Z} \times_2 \mathbf{u}_v^{(v')} \times_3 \mathbf{u}_l^{(l')} \times_4 \mathbf{u}_e^{(e')} \times_5 \mathbf{U}_{pat} \times_6 \mathbf{U}_{pix})_{(i)}\| \quad (21)$$

or:

$$\mathbf{u}_i = (\mathcal{D}_{test})_{(i)} \times (\mathcal{Z} \times_2 \mathbf{u}_v^{(v')} \times_3 \mathbf{u}_l^{(l')} \times_4 \mathbf{u}_e^{(e')} \times_5 \mathbf{U}_{pat} \times_6 \mathbf{U}_{pix})^+_{(i)} \qquad (22)$$

where the superscript + implies *Moore-Penrose pseudoinverse.* For each possible triple $(\mathbf{u}_v^{(v')}, \mathbf{u}_l^{(l')}, \mathbf{u}_e^{(e')}) \in \{(\mathbf{u}_v, \mathbf{u}_l, \mathbf{u}_e)\}$,

$$\begin{aligned} \mathbf{u}_i^{v'l'e'} &= (\mathcal{D}_{test})_{(i)} \times (\mathcal{Z} \times_2 \mathbf{u}_v^{(v')} \times_3 \mathbf{u}_l^{(l')} \times_4 \mathbf{u}_e^{(e')} \times_5 \mathbf{U}_{pat} \times_6 \mathbf{U}_{pix})^+_{(i)} \\ &= (\mathcal{D}_{test})_{(i)} \times \mathbf{A}^{v'l'e'} \\ v' &= 1, \ldots, I_v; \quad l' = 1, \ldots, I_l; \quad e' = 1, \ldots, I_e. \end{aligned} \qquad (23)$$

Later, the best matching identity is determined by:

$$\arg\min_{i',v',l',e'} \|\mathbf{u}_{i_{i'}}^{v'l'e'} - \mathbf{u}_i^{(i')}\| \qquad (24)$$

The proposed method, denoted by Sp-Tensor-LV, is an extension from MPCA-LV[20]. The difference between them is unfolding $\mathcal{D}_{test}$ on mode-i in Eq.(21).

Next, we show how to extend MPCA-JS[20] to sub-pattern tensor face recognition. Let us start from the following definition:

$$\mathcal{A} = \mathcal{Z} \times_5 \mathbf{U}_{pat} \times_6 \mathbf{U}_{pix} \qquad (25)$$

where $\mathcal{A} \in \mathbb{R}^{I'_i \times I'_v \times I'_l \times I'_e \times I_{pat} \times I_{pix}}$. Then we have:

$$\mathcal{A}(i_i, i_v, i_l, i_e) = I_{i_i i_v i_l i_e} \qquad (26)$$

where $I_{i_i i_v i_l i_e}$ is a $1 \times 1 \times 1 \times 1 \times I_{pat} \times I_{pix}$ tensor, $i_i = 1, \ldots, I'_i; i_v = 1, \ldots, I'_v; i_l = 1, \ldots, I'_l; i_e = 1, \ldots, I'_e$. According to Eq.(25), Eq.(18) can be rewritten as:

$$\mathcal{D}_{test} = \mathcal{A} \times_1 \mathbf{u}_i \times_2 \mathbf{u}_v \times_3 \mathbf{u}_l \times_4 \mathbf{u}_e \qquad (27)$$

The above equation can be rewritten as (See Appendix A):

$$\mathcal{D}_{test} = f(\mathbf{u}_i, \mathbf{u}_v, \mathbf{u}_l, \mathbf{u}_e) \cdot \mathbf{A} \tag{28}$$

where $f : \mathbb{R}^{I'_i} \times \mathbb{R}^{I'_v} \times \mathbb{R}^{I'_l} \times \mathbb{R}^{I'_e} \rightarrow \mathbb{R}^{I'_i I'_v I'_l I'_e}$ is a function over $\mathbf{u}_i, \mathbf{u}_v, \mathbf{u}_l, \mathbf{u}_e$. Let $d_{test} = f(\mathbf{u}_i, \mathbf{u}_v, \mathbf{u}_l, \mathbf{u}_e)$ be the description vector for $\mathcal{D}_{test}$, we can then reformulate Eq.(19) as:

$$\underset{d_{test}}{\arg\min} \|(\mathcal{D}_{test})_{(i)} - d_{test} \cdot \mathbf{A}\| \tag{29}$$

This is a linear problem and the least square solution for d_{test} is computed as:

$$d_{test} = (\mathcal{D}_{test})_{(i)} \times \mathbf{A}^{+} \tag{30}$$

where d_{test} is compared to the stored description vectors and the best matching training image is found. Next, we will show that the description vectors of training images can be directly calculated from the matrices $\mathbf{U}_i$, $\mathbf{U}_v$, $\mathbf{U}_l$ and $\mathbf{U}_e$.

We define C_i as,

$$C_i = \begin{bmatrix} C_i^{(1)} \\ C_i^{(2)} \\ \dots \\ C_i^{(I_i)} \end{bmatrix} \tag{31}$$

where

$$C_i^{(k)} = \begin{bmatrix} \mathbf{u}_i^{(k)} & 0 & \dots & 0 \\ 0 & \mathbf{u}_i^{(k)} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \mathbf{u}_i^{(k)} \end{bmatrix} \tag{32}$$

where $\mathbf{u}_i^{(k)}$ (kth row of U_i) is repeated diagonally for $I'_e I'_l I'_v$ times. Similarly we can define C_v, C_l and C_e. Let $\mathbf{M} = C_e \times C_l \times C_v \times C_i$. If m_k is the kth row of the matrix $\mathbf{M}$, then

$$m_k = (\mathcal{D}_{test})_{(i)} \times \mathbf{A}^{+} \tag{33}$$

the best matching identity is $mod(k-1, I_i) + 1$. The detailed proof can be found in [20]. We call the extended method as Sp-Tensor-JS.

4. Experiments

4.1. Database and Experimental Settings

Our experimental data came from the Weizmann database[1]. There were a total of 2018 facial images from 32 individuals with 26 different views, 16 different light conditions and 3 different expressions in this data set. However, only 27 out of the 32 individuals had all 3 expressions, the first five views$(0^\circ, \pm 17^\circ, \pm 34^\circ)$ and the first three light conditions. We evaluated the performance of the proposed approaches for face recognition based on the selected 27 subjects. The 27 subjects had 1874 facial images in total which were resized to 112×92 without any other preprocessing and were then divided into 16 sub-patterns, whose size was 28×23. Within the 1874 facial images from the 27 selected subjects, 1215 facial images that included all expressions, the first five poses and the first three light conditions were organized into a sixth-order tensor $\mathcal{D} \in \mathbb{R}^{27\times5\times3\times3\times16\times644}$, where the six modes were corresponding to identity, view, light, expression, sub-pattern and sub-pattern pixels, respectively. The remaining images, which had different light conditions and views from training set $\mathcal{D}$, were regarded as the test set, and denoted by $test_{mixed}$. For HOSVD and other tensor operations, we used the tensor toolbox developed by Bader and Kolda in MATLAB$^{\mathrm{TM}}$[34]. The parameters of HOSVD were set as the following: $I'_i = I_i$, $I'_v = I_v$, $I'_l = I_l$, $I'_e = I_e$, $I'_{pat} = I_{pat}$ and $I'_{pix} = \frac{1}{2} I_{pix}$, respectively. All experiments were conducted on a 2.66 GHz Intel PC with 16 GB main memory, and the OS of Microsoft Windows XP 64-bits version.

We used a 5-fold cross validation method to analyze the performance of facial recognition for an un-modeled view. In each fold, the ith view's images were selected as the testing set from $\mathcal{D}$, the remaining views were used as the training set. The final result was the sum of the number of correct recognitions in each fold divided by the sum of the total number of facial images tested in each fold. Similarly, a 3-fold cross validation method was used to analyze the performance of facial recognition for an un-modeled light condition. The same method was used in testing un-modeled expressions as well. Finally, to analyze the performance of recognizing a facial image for an un-modeled view under an unmodelled lighting condition, $test_{mixed}$ was used as the test set and $\mathcal{D}$ was used as the training set.

[1] `http://www.faculty.idc.ac.il/moses/`

4.2. time complexities of Sp-Tensor and TensorFace

In this section, the time complexities of Sp-Tensor and TensorFace are analyzed and compared. From HOSVD, the matrix $\mathbf{U}_n$ is the left matrix of SVD performing on the unfolding matrix $\mathbf{D}_{(n)}$. One is required to solve an eigenvalue problem of the matrix $\mathbf{D}_{(n)}\mathbf{D}'_{(n)}$ with the size of $I_n \times I_n$. Its time complexity is $O(I_n^3)$[35]. For Sp-Tensor, $\mathcal{D}$ is decomposed as

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_i \times_2 \mathbf{U}_v \times_3 \mathbf{U}_l \times_4 \mathbf{U}_e \times_5 \mathbf{U}_{pat} \times_6 \mathbf{U}_{pix} \tag{34}$$

Usually, the number of pixels in each sub-pattern is much larger than the number of other factors in the training tensor. So, the time complexity of the decomposition of the training tensor $\mathcal{D}$ is about $O(I_{pix}^3)$. For TensorFace, similarly, $\mathcal{D}'$ is decomposed as

$$\mathcal{D}' = \mathcal{Z}' \times_1 \mathbf{U}'_i \times_2 \mathbf{U}'_v \times_3 \mathbf{U}'_l \times_4 \mathbf{U}'_e \times_5 \mathbf{U}'_{pix} \tag{35}$$

Its time complexity is roughly $O(I'^3_{pix})$. In our experiments, the facial image had $112 \times 92 = 10304$ pixels and I'_{pix} was 10304 in the training tensor of TensorFace. In Sp-Tensor, all images were divided into 16 sub-patterns, each of which contained 644 pixels, and I_{pix} was 644 in the training tensor. So, when building a model, Sp-Tensor cost less time than TensorFace. Table 1 listed the time cost for building models using various numbers of sub-patterns. When the number of sub-patterns was 1, Sp-Tensor degenerated into TensorFace.

Table 1: The consume times for building model on various number of sub-patterns

the number of sub-patterns	64	32	16	8	4	2	1 (TensorFace)
I_{pix}	161	322	644	1288	2576	5152	10304
cost time (second)	2.08	2.11	2.81	9.15	71.29	355.36	21419

4.3. Comparing Sp-Tensor with Sp-PCA1 and TensorFace

In the following experiments, the proposed methods Sp-Tensor and Spm-Tensor[2] were tested and compared with Tensorface[14] and Sp-PCA1[36].

[2] Spm-Tensor denotes that the mean values were subtracted from each of mode-*pix* fibers of $\mathcal{D}$ before decomposing $\mathcal{D}$ as Eq.(6).

As shown in Table 2, out of the four different experiments, Sp-Tensor had the highest recognition accuracy for un-modeled expression and $test_{mixed}$ with a recognition rate of 92.34% and 72.99% respectively. Whereas, Spm-Tensor achieved the highest recognition accuracy of 99.56% in un-modeled cases, exceeding Sp-Tensor and Tensorface (methods that did not remove light variations) by 28%. The reason is that Spm-Tensor reduces the effects of light variation by removing the mean values of each mode-pix fibers of $\mathcal{D}$ in un-modeled light images. The advantage of this effect was further demonstrated when compared to Sp-PCA1 which also reduces the influence of light variations, yet the proposed method of Spm-Tensor surpassed Sp-PCA1 in recognition accuracy by 1.82%. The images from Fig. 8 were examples of original images of one subject from the Weizmann database which had the first view and various expressions and under different light conditions. The corresponding darker images beside the original images were the facial images whose mean values had been removed to eliminate the influence of shadows.

Table 2: Experimental results using Weizmann database on 28 × 23 size sub-patterns

	Sp-Tensor	Spm-Tensor	Tensorface	Sp-PCA1
view	87.82	85.60	79.59	**88.91**
light	72.66	**99.56**	71.28	98.02
expression	**99.34**	98.27	98.93	98.85
$test_{mixed}$	**72.99**	67.86	70.86	71.17

For different views, in each fold, the vth view's images (243 images) were selected as the test set from $\mathcal{D}$, the remaining images were used as the training set. Table 3 showed the number of images that were recognized correctly in each fold. We can see that Sp-tensor reached the top in view5 and Spm-tensor reached the top in view1 and view3. Ideally, to view1 and view5 the results should be symmetrical because both views share the same degree of change. However the experiment results disagreed with this hypothesis because whole head images were used instead of cropped and aligned facial images. As a result, certain details in the whole head images were not aligned symmetrically for both view1 and view5 (i.e. the position of certain facial features such as the ear might be on a certain pixel in view1 but not in the

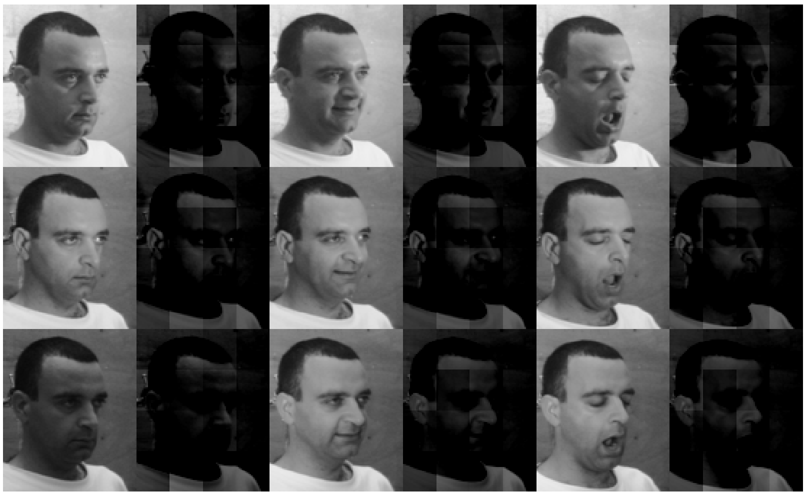


Figure 8: Original images of the 1st subject in the 1st view in Weizmann database under different light conditions with various expressions. The responding images are images whose mean values have been removed.

corresponding pixel in view5).

Table 3: The number of correct recognition in each fold for different views

	Sp-Tensor	Spm-Tensor	Sp-PCA1
view1(+34°)	228	**230**	217
view2(+17°)	207	203	**218**
view3(0°)	215	**220**	217
view4(−17°)	196	207	**217**
view5(−34°)	**221**	180	212

As we can see, Sp-Tensor showed high recognition accuracies when testing a facial image with unmodelled expressions as shown in Table 2. That is because the facial images of the same user with different expressions only show changes in certain sub-patterns, whilst information in other sub-patterns remains invariant. This indicates that Sp-Tensor method can preserve some local information for facial images such that the recognition accuracy is improved.

4.4. the relation between performance and the number of sub-patterns

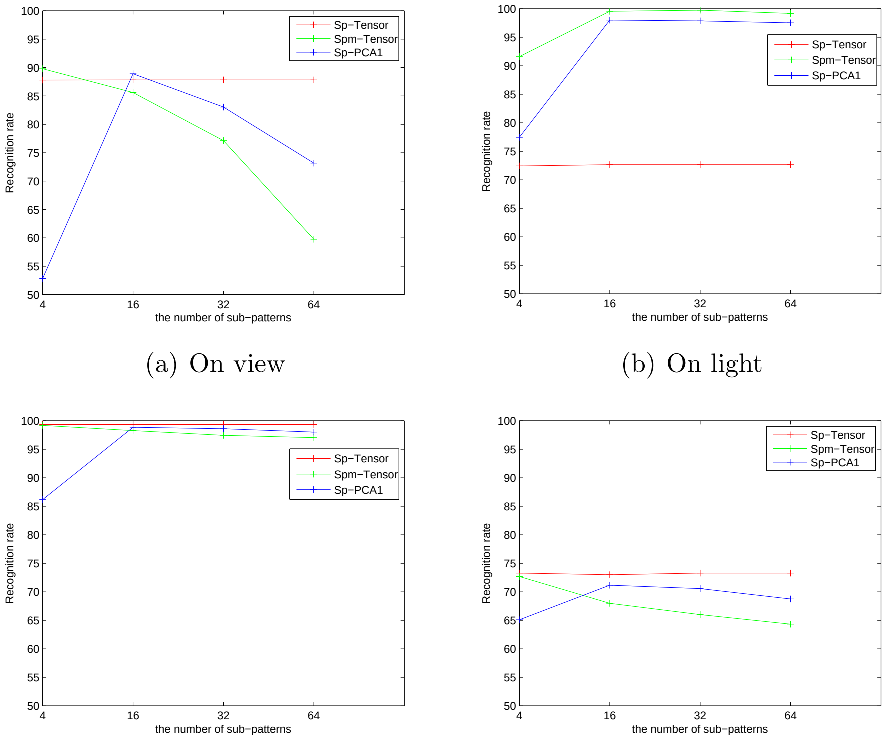
The number of sub-patterns is a tradeoff between the accuracy and stability of the estimation of covariance in the existing techniques of re-organizing variables for images[29]. In this section, the relationship between performance and the number of sub-patterns is investigated. We divided the facial images into 4 (56×46), 32 (14×23) and 64 (7×23) sub-patterns, respectively. Table 4 showed no difference between the performance of Sp-Tensor using 32 (14×23) and 64 (7×23) sub-patterns. For Spm-Tensor and Sp-PCA1, the performances were worse as the number of sub-patterns increased. Fig. 9 showed the changes of recognition accuracies with the number of the sub-patterns when testing on un-modeled view, light, expression and $test_{mixed}$. As we can see from Fig. 9, the number of sub-patterns had less influence on the performance of Sp-Tensor than the other techniques. Overall, Sp-PCA1 achieved the best performance when using 16 sub-patterns. For testing facial images under an un-modeled light condition, Spm-Tensor showed better recognition accuracy than Sp-Tensor and Sp-PCA1. With the same number of sub-patterns, Sp-Tensor's performance surpassed other methods when testing on un-modeled expression and $test_{mixed}$, while Spm-Tensor achieves the best recognition accuracy when testing on facial images under an un-modeled light condition. However, tests on images with an un-modeled view revealed that different methods yielded the best results using different size sub-patterns.

Table 4: Experimental results using Weizmann database on 56 × 46 size, 14 × 23 size and 7 × 23 size sub-patterns

		Sp-Tensor	Spm-Tensor	Sp-PCA1
56 × 46	view	87.82	**89.79**	52.84
	light	72.43	**91.60**	77.45
	expression	**99.34**	99.18	86.17
	$test_{mixed}$	**73.29**	72.69	65.10
14 × 23	view	**87.82**	77.12	83.05
	light	72.26	**99.75**	97.86
	expression	**99.34**	97.45	98.60
	$test_{mixed}$	**73.14**	66.01	70.56
7 × 23	view	**87.82**	59.75	73.17
	light	72.26	**99.18**	97.53
	expression	**99.34**	97.04	98.02
	$test_{mixed}$	**73.14**	64.34	68.74

4.5. Analysis of recognition methods Sp-Tensor-LV and Sp-Tensor-JS to Sp-Tensor

Looking deeper into Sp-Tensor's extended method, we carried out an experiment using 28 × 23 size sub-patterns to compare Sp-Tensor with Sp-Tensor-LV and Sp-Tensor-JS. For Sp-Tensor-LV, an extended method of MPCA-LV, we used the maximum *cosine distance* between $\mathbf{u}_{i_{i'}}^{v'l'e'}$ and $\mathbf{u}_{i}^{i'}$ instead of the minimum *norm* in Eq.(24). Comparing and analyzing the performances of both maximum cosine distance and minimum norm of Sp-Tensor-LV revealed that experiments conducted on images from un-modeled views, expressions and $test_{mixed}$, using norm as measure outperformed the ones using cosine distance as measure. But for experiments on images with un-modeled lights, the performance of using norm as measure was a bit worse



(a) On view

(b) On light

(c) On expression

(d) On $test_{mixed}$

Figure 9: Recognition rates vs. the number of sub-patterns

than those of using cosine distance as a measure. The results of our experiments were listed and organized in Table 5.

To further analyze and compare the difference between the methods, experiments were conducted using Sp-Tensor-JS, an extended method from MPCA-JS. Through the results shown in Table 5, one can notice that the recognition accuracy of Sp-Tensor-LV (79.51%) was dramatically higher than that of Sp-Tensor-JS (17.94%) when testing on un-modeled view. Additional experiments on images from un-modeled expressions and $test_{mixed}$, also showed that Sp-Tensor-LV has higher recognition accuracy than those of Sp-Tensor-JS. This proves that Sp-Tensor-LV had more robust than Sp-Tensor-JS when testing a facial image that combines unknown light condition and unknown view in the model.

To fully compare different methods, we then removed the mean values and obtained the following extended methods, Spm-Tensor-JS and Spm-Tensor-LV. Reducing the effects of light variation improved both extended recognition methods performances on un-modeled light. Comparing the results of the extended methods to the results of Spm-Tensor in Table 2, Spm-Tensor still surpassed other methods for recognition. However it is worth noting that Spm-Tensor-JS, though a bit lower in recognition accuracy compared to Spm-Tensor, was faster than Spm-Tensor. To obtain a better overview of all methods, we also implemented MPCA-LV and MPCA-JS methods on the Weizmann database, their results were listed in Table 6.

In conclusion from Tables 2, 5 and 6, The proposed method Sp-Tensor showed higher recognition accuracy than MPCA-LV and MPCA-JS as well as their corresponding extended methods, Sp-Tensor-LV and Sp-Tensor-JS. Even when effects of light variations were reduced, experiments illustrated that our method, Spm-Tensor, surpassed other recognition methods in terms of recognition accuracy, while being slightly slower than Spm-Tensor-JS.

Table 5: Experimental results of two extended methods

	Sp-Tensor-LV		Spm-Tensor-LV		Sp-Tensor-JS	Spm-Tensor-JS
	cos	norm	cos	norm		
view	75.06	79.51	77.61	78.68	17.94	17.28
light	80.16	70.78	99.01	98.44	99.42	99.51
expression	98.44	98.93	98.11	98.35	90.37	89.79
$test_{mixed}$	68.44	70.86	70.71	71.02	53.87	55.24

Table 6: Experimental results of MPCA-LV and MPCA-JS

	MPCA-LV		MPCA-JS
	cos	norm	
view	75.23	79.59	20.25
illumination	80.25	71.28	99.51
expression	98.44	98.93	91.03
$test_{mixed}$	68.59	70.86	57.36

5. Conclusion

Facial images change appearance due to multiple factors such as different poses, lighting variations, and facial expressions. Tensors are higher order extensions of vectors and matrices, which make it possible to analyze different appearance factors of facial variation. However, the existing methods using tensor rasterize facial images as vectors, which lead to a loss of spatial structure information for facial images. In order to solve this problem, we extend current tensor-based facial recognition method to the sub-pattern mode. The advantages of this method are:(1) Not only the spatial structure information and local information of facial images are preserved during model-building, but spatial structure information is also preserved during recognition; (2) Sp-Tensor is less expensive than the existing methods when building a model. We also extended two existing Tensorface recognition methods to sub-pattern mode. In order to compare the performances of Sp-Tensor, Sp-PCA1, original TensorFace and two extended recognition methods, the experiments were conducted on the Weizmann face database using different size sub-patterns. The experimental results prove that the Sp-Tensor not only works better than the original TensorFace, but outperforms Sp-PCA1 when testing facial images under an unknown light condition or with an unknown expression or under the combination of unknown view and unknown light condition in the model. The performance of Sp-Tensor is stable for varying size of sub-pattern for testing of images from an un-modeled view.

Our future research will focus on the matrix formed sub-pattern images instead of vector formed ones to explore the advantages of spatial structure information of sub-pattern images. Meanwhile, it would be meaningful to explore how sub-pattern tensor face methods can be used for face recognition under occlusion.

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.

The authors would like to thank Mary Tate, Lincong Wang, Mingming Sun and Jianqin Qu for their valuable comments.

Appendix A. Deduction of Sp-Tensor-JS

After applying the rule of mode multiplication, the Eq. (27) can be written as,

$$
\begin{aligned}
\mathcal{D}_{test} &= \sum_{i_e=1}^{I'_e} \mathbf{u}_e(i_e) \cdot \sum_{i_l=1}^{I'_l} \mathbf{u}_l(i_l) \cdot \sum_{i_v=1}^{I'_v} \mathbf{u}_v(i_v) \cdot \sum_{i_i=1}^{I'_i} \mathbf{u}_i(i_i) \cdot (I_{i_i i_v i_l i_e})_{(i)} \\
&= \sum_{i_e=1}^{I'_e} \mathbf{u}_e(i_e) \cdot \sum_{i_l=1}^{I'_l} \mathbf{u}_l(i_l) \cdot \sum_{i_v=1}^{I'_v} \mathbf{u}_v(i_v) \cdot \mathbf{u}_i \cdot \begin{bmatrix} (I_{1 i_v i_l i_e})_{(i)} \\ \cdots \\ (I_{I'_i i_v i_l i_e})_{(i)} \end{bmatrix} \\
&= \sum_{i_e=1}^{I'_e} \mathbf{u}_e(i_e) \cdot \sum_{i_l=1}^{I'_l} \mathbf{u}_l(i_l) \cdot \left\{ \mathbf{u}_v(1) \cdot \mathbf{u}_i \cdot \begin{bmatrix} (I_{11 i_l i_e})_{(i)} \\ \cdots \\ (I_{I'_i 1 i_l i_e})_{(i)} \end{bmatrix} + \cdots + \mathbf{u}_v(I'_v) \cdot \mathbf{u}_i \cdot \begin{bmatrix} (I_{1 I'_v i_l i_e})_{(i)} \\ \cdots \\ (I_{I'_i I'_v i_l i_e})_{(i)} \end{bmatrix} \right\} \\
&= \sum_{i_e=1}^{I'_e} \mathbf{u}_e(i_e) \cdot \sum_{i_l=1}^{I'_l} \mathbf{u}_l(i_l) \cdot \mathbf{u}_v \cdot \begin{bmatrix} \mathbf{u}_i \cdot \begin{bmatrix} (I_{11 i_l i_e})_{(i)} \\ \cdots \\ (I_{I'_i 1 i_l i_e})_{(i)} \end{bmatrix} & \cdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \cdots & \mathbf{u}_i \cdot \begin{bmatrix} (I_{1 I'_v i_l i_e})_{(i)} \\ \cdots \\ (I_{I'_i I'_v i_l i_e})_{(i)} \end{bmatrix} \end{bmatrix} \\
&= \sum_{i_e=1}^{I'_e} \mathbf{u}_e(i_e) \cdot \sum_{i_l=1}^{I'_l} \mathbf{u}_l(i_l) \cdot \mathbf{u}_v \cdot \underset{(I'_v \times I'_i I'_v)}{\begin{bmatrix} \mathbf{u}_i & \cdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \cdots & \mathbf{u}_i \end{bmatrix}} \cdot \begin{bmatrix} \begin{bmatrix} (I_{11 i_l i_e})_{(i)} \\ \cdots \\ (I_{I'_i 1 i_l i_e})_{(i)} \end{bmatrix} \\ \cdots \\ \cdots \\ \begin{bmatrix} (I_{1 I'_v i_l i_e})_{(i)} \\ \cdots \\ (I_{I'_i I'_v i_l i_e})_{(i)} \end{bmatrix} \end{bmatrix}
\end{aligned}
\tag{A.1}
$$

we denote

$$\underset{(I'_v \times I'_i I'_v)}{\begin{bmatrix} \mathbf{u}_i & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \mathbf{u}_i \end{bmatrix}} \cdot \begin{bmatrix} \begin{bmatrix} (I_{11 i_l i_e})_{(i)} \\ \dots \\ (I_{I'_i 1 i_l i_e})_{(i)} \end{bmatrix} \\ \dots \\ \dots \\ \begin{bmatrix} (I_{1 I'_v i_l i_e})_{(i)} \\ \dots \\ (I_{I'_i I'_v i_l i_e})_{(i)} \end{bmatrix} \end{bmatrix} = Y_{i_l} \tag{A.2}$$

Eq.(A.1) can be written as,

$$\begin{aligned} \mathcal{D}_{test} &= \sum_{i_e=1}^{I'_e} \mathbf{u}_e(i_e) \cdot \sum_{i_l=1}^{I'_l} \mathbf{u}_l(i_l) \cdot \mathbf{u}_v \cdot Y_{i_l} \\ &= \sum_{i_e=1}^{I'_e} \mathbf{u}_e(i_e) \cdot \mathbf{u}_l \cdot \begin{bmatrix} \mathbf{u}_v \cdot Y_{i_l=1} & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \mathbf{u}_v \cdot Y_{i_l=I'_l} \end{bmatrix} \\ &= \sum_{i_e=1}^{I'_e} \mathbf{u}_e(i_e) \cdot \mathbf{u}_l \cdot \underset{(I'_l \times I'_v I'_l)}{\begin{bmatrix} \mathbf{u}_v & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \mathbf{u}_v \end{bmatrix}} \cdot \begin{bmatrix} Y_{i_l=1} \\ \dots \\ Y_{i_l=I'_l} \end{bmatrix} \end{aligned} \tag{A.3}$$

Simlilarly, we denote

$$\underset{(I'_l \times I'_v I'_l)}{\begin{bmatrix} \mathbf{u}_v & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \mathbf{u}_v \end{bmatrix}} \cdot \begin{bmatrix} Y_{i_l=1} \\ \dots \\ Y_{i_l=I'_l} \end{bmatrix} = Z_{i_e} \tag{A.4}$$

Eq.(A.3) can be written as,

$$\begin{aligned} \mathcal{D}_{test} &= \sum_{i_e=1}^{I'_e} \mathbf{u}_e(i_e) \cdot \mathbf{u}_l \cdot Z_{i_e} \\ &= \mathbf{u}_e \cdot \begin{bmatrix} \mathbf{u}_l \cdot Z_{i_e=1} & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \mathbf{u}_l \cdot Z_{i_e=I'_e} \end{bmatrix} \\ &= \mathbf{u}_e \cdot \underset{(I'_e \times I'_l I'_e)}{\begin{bmatrix} \mathbf{u}_l & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \mathbf{u}_l \end{bmatrix}} \cdot \begin{bmatrix} Z_{i_e=1} \\ \dots \\ Z_{i_e=I'_e} \end{bmatrix} \end{aligned} \tag{A.5}$$

Substituting Z_{i_e} and Y_{i_l} with their definition and after a few more steps, we get,

$$\mathcal{D}_{test} = \mathbf{u}_e \cdot \underset{(I'_e \times I'_l I'_e)}{\begin{bmatrix} \mathbf{u}_l & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \mathbf{u}_l \end{bmatrix}} \cdot \underset{(I'_l I'_e \times I'_l I'_e I'_v)}{\begin{bmatrix} \mathbf{u}_v & \dots & \dots & 0 \\ \dots & \mathbf{u}_v & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \mathbf{u}_v \end{bmatrix}} \cdot$$

$$\underset{(I'_l I'_e I'_v \times I'_l I'_e I'_v I'_i)}{\begin{bmatrix} \mathbf{u}_i & \dots & \dots & \dots & 0 \\ \dots & \mathbf{u}_i & \dots & \dots & \dots \\ \dots & \dots & \mathbf{u}_i & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \mathbf{u}_i \end{bmatrix}} \cdot \underset{(I'_l I'_e I'_v I'_i \times I_{pat} I_{pixel})}{\begin{bmatrix} (I_{1111})_{(i)} \\ \dots \\ (I_{I'_i 111})_{(i)} \\ (I_{1211})_{(i)} \\ \dots \\ (I_{I'_i I'_v 11})_{(i)} \\ (I_{1121})_{(i)} \\ \dots \\ (I_{I'_i I'_v I'_l 1})_{(i)} \\ (I_{1112})_{(i)} \\ \dots \\ (I_{I'_i I'_v I'_l I'_e})_{(i)} \end{bmatrix}} \tag{A.6}$$

$$\Rightarrow \mathcal{D}_{test} = f(\mathbf{u}_i, \mathbf{u}_v, \mathbf{u}_l, \mathbf{u}_e) \cdot \mathbf{A}$$

where $f : \mathbb{R}^{I'_i} \times \mathbb{R}^{I'_v} \times \mathbb{R}^{I'_l} \times \mathbb{R}^{I'_e} \to \mathbb{R}^{I'_i I'_v I'_l I'_e}$ is a function over $\mathbf{u}_i, \mathbf{u}_v, \mathbf{u}_l, \mathbf{u}_e$. Let $d_{test} = f(\mathbf{u}_i, \mathbf{u}_v, \mathbf{u}_l, \mathbf{u}_e)$ be the description vector for $\mathcal{D}_{test}$.

[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] W. Bian, D. Tao, Max-min distance analysis by using sequential sdp relaxation for dimension reduction, IEEE Transactions on Pattern Analysis and Machine Intelligence (99). doi:10.1109/TPAMI.2010.189.

[4] T. Zhang, D. Tao, X. Li, J. Yang, Patch alignment for dimensionality reduction, IEEE Transactions on Knowledge and Data Engineering (2008) 1299–1313.

[5] J. Li, D. Tao, Simple exponential family PCA, AISTATS, 2010.

[6] T. Zhou, D. Tao, X. Wu, Manifold elastic net: a unified framework for sparse dimension reduction, Data Mining and Knowledge Discovery (2010) 1–32.

[7] S.-J. Wang, C.-G. Zhou, N. Zhang, X.-J. Peng, Y.-H. Chen, X.-H. Liu, Face recognition using second order discriminant tensor subspace, Neurocomputing.

[8] S.-J. Wang, J. Yang, N. Zhang, C.-G. Zhou, Tensor discriminant color space for face recognition, IEEE Transactions on Image Processing-doi:10.1109/TIP.2011.2121084.

[9] Y. Mu, D. Tao, X. Li, F. Murtagh, Biologically Inspired Tensor Features, Cognitive Computation 1 (4) (2009) 327–341.

[10] Y. Mu, D. Tao, Biologically inspired feature manifold for gait recognition, Neurocomputing 73 (4-6) (2010) 895–902.

[11] L. De Lathauwer, B. De Moor, J. Vandewalle, On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors, SIAM Journal on Matrix Analysis and Applications 21 (4) (2000) 1324–1342.

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

[13] R. Badeau, R. Boyer, Fast multilinear singular value decomposition for structured tensors, SIAM Journal on Matrix Analysis and Applications 30 (2008) 1008.

[14] M. A. O. Vasilescu, D. Terzopoulos, Multilinear image analysis for facial recognition, in: International Conference on Pattern Recognition, Vol. 16, Citeseer, 2002, pp. 511–514.

[15] M. A. O. Vasilescu, D. Terzopoulos, Multilinear analysis of image ensembles: TensorFaces, in: ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I, Springer-Verlag, London, UK, 2002, pp. 447–460.

[16] M. A. O. Vasilescu, D. Terzopoulos, Multilinear subspace analysis of image ensembles, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, 2003, pp. II–93–9.

[17] M. A. O. Vasilescu, D. Terzopoulos, Multilinear independent components analysis, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2005, Vol. 1, 2005, pp. 547–553.

[18] X. B. Gao, C. N. Tian, Multi-view face recognition based on tensor subspace analysis and view manifold modeling, Neurocomputing 72 (16-18) (2009) 3742–3750.

[19] X. Geng, K. Smith-Miles, Z.-H. Zhou, L. Wang, Face image modeling by multilinear subspace analysis with missing values, in: MM '09: Proceedings of the seventeen ACM international conference on Multimedia, ACM, New York, NY, USA, 2009, pp. 629–632.

[20] S. Rana, W. Liu, M. Lazarescu, S. Venkatesh, A unified tensor framework for face recognition, Pattern Recognition 42 (11) (2009) 2850–2862.

[21] C. Tian, G. Fan, X. Gao, Multi-view face recognition by nonlinear tensor decomposition, in: Proc. 19th International Conference on Pattern Recognition ICPR 2008, 2008, pp. 1–4.

[22] S. Rana, W. Liu, M. Lazarescu, S. Venkatesh, Recognising faces in unseen modes: A tensor based approach, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR 2008, 2008, pp. 1–8.

[23] S. W. Park, M. Savvides, Individual kernel tensor-subspaces for robust face recognition: A computationally efficient tensor framework without requiring mode factorization, IEEE Transactions on Systems, Man, and Cybernetics, Part B 37 (5) (2007) 1156–1166.

[24] J. Yang, D. Zhang, A. Frangi, J. Yang, Two-dimensional PCA: a new approach to appearance-based face representation and recognition, IEEE

Transactions on Pattern Analysis and Machine Intelligence (2004) 131–137.

[25] D. Zhang, Z.-H. Zhou, (2D)^{2}PCA: Two-directional two-dimensional P-CA for efficient face representation and recognition, Neurocomputing 69 (1-3) (2005) 224–231.

[26] W. Zuo, D. Zhang, K. Wang, Bidirectional PCA with assembled matrix distance metric for image recognition, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 36 (4) (2006) 863–872.

[27] S. Chen, Y. Zhu, D. Zhang, J. Yang, Feature extraction approaches based on matrix pattern: MatPCA and MatFLDA, Pattern Recognition Letters 26 (8) (2005) 1157–1167.

[28] S. C. Chen, Y. L. Zhu, Subpattern-based principle component analysis, Pattern Recognition 37 (5) (2004) 1081–1083.

[29] S. Shan, B. Cao, Y. Su, L. Qing, X. Chen, W. Gao, Unified principal component analysis with generalized covariance matrix for face recognition, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR 2008, 2008, pp. 1–7.

[30] D. Tao, X. Li, X. Wu, W. Hu, S. Maybank, Supervised tensor learning, Knowledge and Information Systems 13 (1) (2007) 1–42.

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

[32] H.-T. Chen, T.-L. Liu, C.-S. Fuh, Learning effective image metrics from few pairwise examples, in: Proc. Tenth IEEE International Conference on Computer Vision ICCV 2005, Vol. 2, 2005, pp. 1371–1378 Vol. 2.

[33] Y. Liu, Y. Liu, K. Chan, Tensor distance based multilinear locality-preserved maximum information embedding, IEEE Transactions on Neural Networks 21 (11) (2010) 1848 –1854. doi:10.1109/TNN.2010.2066574.

[34] B. Bader, T.G.Kolda, Tensor toolbox version 2.3, copyright 2009, sandia national laboratories, `http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/`.

[35] G.H.Golub, C. Loan, Matrix computations, Johns Hopkins University Press, 1996.

[36] P. C. Hsieh, P. C. Tung, A novel hybrid approach based on sub-pattern technique and whitened PCA for face recognition, Pattern Recognition 42 (5) (2009) 978–984.