Applied Mathematical Modelling 39 (2015) 1266–1274

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

A low-rank tensor-based algorithm for face recognition

Lăcrămioara Liţă, Elena Pelican *

Ovidius University, Dept of Math and Comp Sci, Mamaia 124, 900527 Constanta, Romania

A R T I C L E   I N F O

Article history:
Received 30 May 2014
Received in revised form 24 July 2014
Accepted 12 August 2014
Available online 28 August 2014

Keywords:
Pattern recognition
Dimension reduction
Eigenvalue
Eigenvector
Singular values
Tensors

A B S T R A C T

The face recognition problem arises in a wide range of real life applications. Our new developed face recognition algorithm, based on higher order singular value decomposition (HOSVD) makes use of only third order tensor. A convenient way of writing the commutativity of different modes of tensor-matrix multiplications leads to a new method that outperforms in terms of complexity another third order tensor method. The resulting algorithm is more successful (in terms of recognition rate) than the conventional eigenfaces algorithm. Its effectiveness is proved for two benchmark datasets (ExtYaleB and Essex datasets), which are ensembles of facial images that combine different modes, like facial geometries, illuminations, and expressions.

1. Introduction

In recent years, due to technical evolution, face recognition became an issue of increasing interest in a wide range of applications, such as security and surveillance problems, forensic problems, human–computer interface, access control problems, multimedia communications and so on. It is well known that human accuracy in identifying a face in a crowd is of 97.5%, being independent on expressions, head poses, illuminations conditions and other multiple factors. This robustness of the human perception is crucial to human social interaction. So far, this is an open problem in computer vision and pattern recognition.

Over the years, different types of algorithms have been developed for face recognition and, also, algorithms used in other branches of science were adapted to solve this problem [1,2].

Linear algebra, where the main objects are vectors and matrices, has provided with many valuable algorithms for this issue. From the Karhunen–Loeve transform [3,4] to principal component analysis [5,6], with the version of conventional eigenfaces, and its improved version, independent component analysis (ICA) [7], these algorithms take into account a single factor variations in image formation. When illumination, viewpoint, head pose, and expression vary, eigenfaces performs poorly.

Multilinear algebra, the algebra of higher-order tensors, offers powerful and sophisticated tools to approach a multifactor model of representation of images. A good survey on tensor is [8] where the authors provide an overview of higher order tensors, their decompositions and software packages for working with tensors [8,9].

We use in this paper a higher-order generalization of PCA (Tucker decomposition) and singular value decomposition (SVD) of matrices for computing principal components. If, for a matrix, the existence and uniqueness of SVD is assured, the situation for tensors are not the same: there is no true tensor SVD, with all good properties [10], there are many ways

* Corresponding author.
E-mail addresses: lgrecu@univ-ovidius.ro (L. Liţă), epelican@univ-ovidius.ro (E. Pelican).

http://dx.doi.org/10.1016/j.apm.2014.08.003

to decompose tensor orthogonally. We use in a convenient way this property in order to have representation that separates the different modes from the formation of facial images.

The rest of the paper is organized as follows. Section 2 reviews some related work; Section 3 summarizes the standard Eigenfaces algorithm whereas Section 4 recalls briefly the main results in tensor algebra that are relevant to our approach. Section 5 presents the multilinear analysis of the face recognition with the description of the proposed algorithm. Section 6 contains numerical experiments on the analysis of facial images from two benchmark datasets using our algorithm and the PCA. Section 7 concludes the paper.

2. Related work

Facial representation for recognition has been tackled by a family of PCA-based algorithm, such as Eigenfaces [6,5] and FisherFaces [11]. They compute the PCA by performing an SVD on $P \times N$ = data matrix of "vectorized" $N = m \times n$ pixel images of P people. This type of linear model is suited when the identity of the subject is the only variable that counts for image formation. In [12] it is proved that a natural representation of a collection of images is a third order tensor, rather than a simple matrix of vectorized images. N-mode analysis was first proposed by Tucker [13] and developed by Kapteyn et al. [14,15]. Also de Lathauwer in [15,16] develops the SVD for tensors, called higher order SVD (HOSVD).

Vasilescu and Terzopoulos proposed a "TensorFaces" representation in [17] which has several advantages over conventional Eigenfaces. They make use of 5-mode decomposition of a tensor (HOSVD for fifth order tensors), employing the Weizmann face database of 28 male subjects photographed in 15 different poses under 4 illuminations performing 3 different face expressions. Vasilescu in [18,19] addressing the motion analysis/synthesis problem, structured motion capture data in tensor form and developed an algorithm for extracting "human" motion signature. In [20], Vasilescu et al. introduced a tensor framework for image-based rendering and developed an algorithm called TensorTextures. The same authors introduced in [21] a multilinear generalization of ICA (MICA), and successfully applied this algorithm to a multimodel face recognition problem involving multiple people imaged under different viewpoints and illumination.

The 3-mode SVD facial representation technique (HOSVD for third order tensors) that we develop in this paper is for the case when the dataset is not fully organized. We mean that the information for a person is about only one attribute (or facial expression, or illumination, or head pose), and not about all of them as for the Weizmann face dataset. The obtained results are promising and comparable with the ones in MICA in [21] with n-order tensors.

3. Eigenfaces (PCA)

Although the Eigenfaces algorithm is already well-known, we will include a brief description of it for the sake of completeness. The eigenfaces are the eigenvectors of the covariance matrix of a set of faces [5,22]. These vectors are called eigenfaces because they resemble human faces when they are represented. These eigenfaces can be obtained by performing a mathematical process, namely the principal component analysis (PCA). The eigenvectors are chosen in descending order of their importance: the first component has the highest significance and so on. It must be taken into account that each principal component is orthogonal to all previous principal components. The idea of using principal components to represent human faces was developed by Sirovich and Kirby [4,6] and used by Turk and Pentland [5,22] for face detection and face recognition.

Every image, from the dataset, has the same resolution $M = n_1 \times n_2$ and is transformed into a vector $\Gamma_i \in \mathbb{R}^{M\times 1}$. Then we compute the average face vector Ψ of all N images and subtract the average face vector from all vectors $\varphi_i = \Gamma_i - \Psi$. The covariance matrix is $C = AA^T \in \mathbb{R}^{M\times M}$, with M the resolution of an image. Because in practice, M is very large, the computational effort to determine M eigenvalues and eigenvectors is huge. Thus, the idea is to reduce the amount of calculations by reducing the size. Let $L = A^TA, L \in \mathbb{R}^{N\times N}$. From all N vectors obtained, we keep only the first K vectors corresponding to the largest K eigenvalues. To identify a new image, Γ, we represent it using the eigenvectors $\{u_1, u_2, \ldots, u_K\}$. Thus, we have $\omega_i = u_i^T(\Gamma - \Psi), i = 1 : K$. These coefficients ω_i form the vector $\Omega^T = [\omega_1, \omega_2, \ldots, \omega_K]$. The vector Ω describes the contribution of each eigenface in representing the image Γ and is used to classify the new image Γ.

4. Fundamentals of tensor algebra

A N-order tensor is an object with N dimensions. If $N = 1$ (first order tensor) we have a vector and if $N = 2$ (second order tensor) we have a matrix. In the next section we will deal only with the case where $N = 3$. Hence, further, we consider a third order tensor $A \in \mathbb{R}^{l\times m\times n}$.

An element of tensor A is denoted by $A(i,j,k)$, where $1 \leqslant i \leqslant l,\ 1 \leqslant j \leqslant m$, and $1 \leqslant k \leqslant n$. We define the first mode fibers of a third order tensor A to be the column vectors $A(:,j,k)$, the second mode fibers as vectors $A(i,:,k)$, and the third mode fibers as vectors $A(i,j,:)$. Hence, fibers are characterized by fixing the index in all modes but one. Similarly, we define the slices of a tensor to be the matrix (for a third order tensor) obtained by fixing the index in one mode, namely we have the slices $A(i,:,:), A(:,j,:)$, and $A(:,:,k)$.

The scalar product $\langle A, B\rangle$ of two tensors $A, B \in \mathbb{R}^{l\times m\times n}$ is computed as $\langle A, B\rangle = \sum_{i=1}^{l}\sum_{j=1}^{m}\sum_{k=1}^{n} A(i,j,k)B(i,j,k)$. Two tensors A and B are orthogonal if $\langle A, B\rangle = 0$. Let $A \in \mathbb{R}^{l\times m\times n}, U \in \mathbb{R}^{l_0\times l}$ and $A\times_1 U$ a tensor of size $l_0 \times m \times n$, we have the following way of multiplication (first mode tensor-matrix multiplication)

$$(A\times_1 U)(j, i_2, i_3) = \sum_{k=1}^{l} u_{j,k} a_{k,i_2,i_3}. \tag{1}$$

For $A \in \mathbb{R}^{l\times m\times n}, U \in \mathbb{R}^{m_0\times m}$ and $A\times_2 U$ a tensor of size $l \times m_0 \times n$ we have second mode tensor-matrix multiplication

$$(A\times_2 U)(i_1, j, i_3) = \sum_{k=1}^{m} u_{j,k} a_{i_1,k,i_3}, \tag{2}$$

and for $A \in \mathbb{R}^{l\times m\times n}, U \in \mathbb{R}^{n_0\times n}$ and $A\times_3 U$ a tensor of size $l \times m \times n_0$ we have third mode tensor-matrix multiplication

$$(A\times_3 U)(i_1, i_2, j) = \sum_{k=1}^{n} u_{j,k} a_{i_1,i_2,k}. \tag{3}$$

The i-mode and j-mode multiplication commute if $i \neq j, i, j \in \{1, 2, 3\}$:

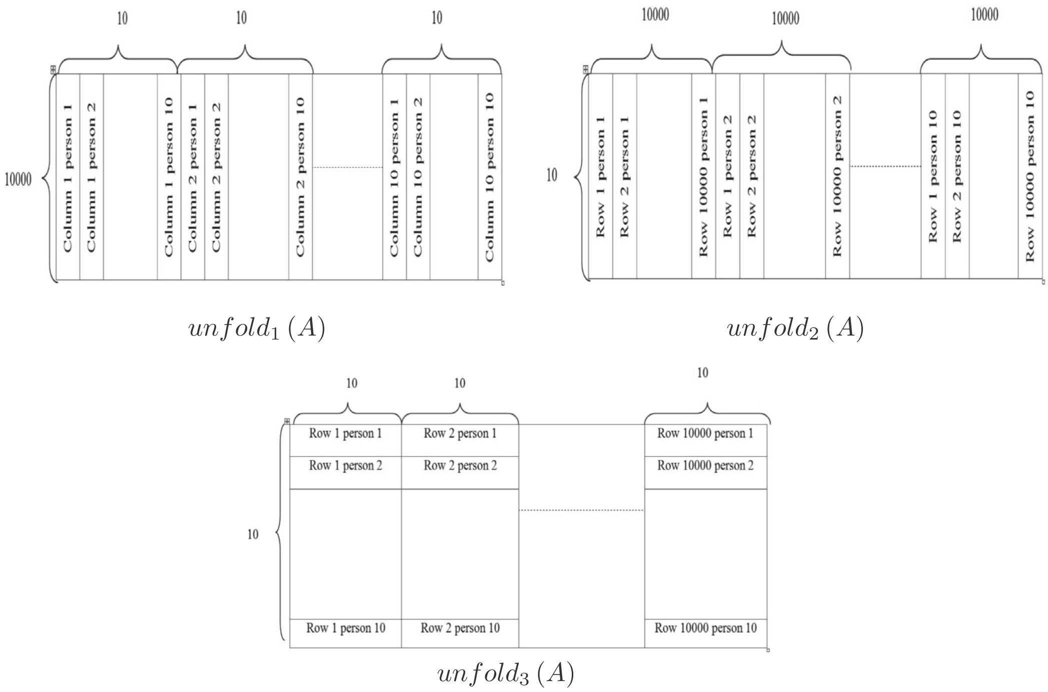
$$(A\times_i U)\times_j V = \left(A\times_j V\right)\times_i U = A\times_i U\times_j V. \tag{4}$$

It is sometimes convenient to rearrange the elements of a tensor into a matrix, this process is called unfolding a tensor into a matrix [15] or flattening a tensor [17]. In what follows, we keep the notation from [23,16]: $A_{(i)} = unfold_i(A)$. Different papers sometimes use different orderings of the columns for the n-mode unfoldings. We use the meaning and the notation from [16]. This ordering is not important as long as it is consistent across related calculations. In all unfoldings, row i of $A_{(j)}$ contains all the elements of A which have the jth index equal to i. Thus, we have

$$A_{(1)} = unfold_1(A) = (A(:, 1, :)A(:, 2, :)\ldots A(:, m, :)), \tag{5}$$

$$A_{(2)} = unfold_2(A) = \left(A(:, :, 1)^T A(:, :, 2)^T \ldots A(:, :, n)^T\right), \tag{6}$$

$$A_{(3)} = unfold_3(A) = \left(A(1, :, :)^T A(2, :, :)^T \ldots A(l, :, :)^T\right). \tag{7}$$



The inverse of the unfolding operation is folding $fold_i(unfold_i(A)) = A$. Using these unfoldings we obtain: $A\times_1 U = fold_1(U \cdot unfold_1(A)), A\times_2 U = fold_2(U \cdot unfold_2(A)), A\times_3 U = fold_3(U \cdot unfold_3(A))$.

Next, we present a generalization of the SVD theorem for matrices to SVD for tensors [15,16,23].

Theorem 1 (HOSVD [15,23]). *The tensor $A \in \mathbb{R}^{l\times m\times n}$ can be written as*

$$A = S\times_1 U^{(1)}\times_2 U^{(2)}\times_3 U^{(3)} \tag{8}$$

where $U^{(1)} \in \mathbb{R}^{l\times l}, U^{(2)} \in \mathbb{R}^{m\times m}, U^{(3)} \in \mathbb{R}^{n\times n}$ are orthogonal matrices. Matrices $U^{(i)}$ are obtained from $A_{(i)} = U^{(i)}\Sigma^{(i)}\left(V^{(i)}\right)^T$, $A_{(i)} = unfold_i(A)$, without forming the $V^{(i)}$ explicitly.

$$S = A\times_1\left(U^{(1)}\right)^T\times_2\left(U^{(2)}\right)^T\times_3\left(U^{(3)}\right)^T \tag{9}$$

is the core tensor, of the same size as A and satisfies

- any two different slices fixed in the same mode are orthogonal (all-orthogonality) $\langle S(i,:,:), S(j,:,:)\rangle = 0$, $i\neq j$, $\langle S(:,i,:), S(:,j,:)\rangle = 0, i\neq j$, $\langle S(:,:,i), S(:,:,j)\rangle = 0,\ i\neq j$;

- the norms of the slices along every mode are ordered, e.g., for the first mode we have $\|S(1,:,:)\|_F \geqslant \|S(2,:,:)\|_F \geqslant \ldots \geqslant \|S(l,:,:)\|_F \geqslant 0$.

The Frobenius norms $\|S(i,:,:)\|_F$ are n-mode singular values of A and the vector $U_i^{(n)}$ is the ith n-mode singular vector. The higher order singular value decomposition of a tensor is not unique.

5. Multilinear analysis. Tensor-*ipe* algorithm

Our datasets are organized as collections of matrices, each matrix containing the vectorized pictures of the same person. The dataset is divided into two nonoverlapping sets: the training and testing sets. Let $A \in \mathbb{R}^{n_i\times n_e\times n_p}$ be a tensor representing our training dataset and let a vector z from $\mathbb{R}^{n_i}$ represent a picture from the testing set. Here n_i is the resolution of a picture, n_p is the number of persons from the dataset, and n_e is the number of expressions per person. Hence, $n_e n_p$ is the number of pictures in the dataset. We want to find out if the algorithm identifies the person in the picture correctly. In [23], for the algorithms from page 173–174, it is assumed that $n_i \gg n_e n_p$. For a large dataset (e.g. ExtYaleB dataset), after reducing image size (as in [24], we get $n_i \ll n_e n_p$. When performing the tests for ExtYaleB dataset with the algorithms from [23] page 173–174 (without reducing the image size in order to fulfill the assumption $n_i \gg n_e n_p$) we obtain smaller recognition rate than for PCA (Eigenfaces algorithm).

In order to address this issue, we propose another algorithm for face recognition, when $n_i \ll n_e n_p$. This is the case of large datasets, when the number of pictures is huge. Briefly, our proposed multilinear recognition algorithm obtaines the HOSVD decomposition (10) (for tensor A of vectorized training images) and afterwards, using the commutativity of i-mode and j-mode tensor-matrix multiplication, extract the matrix G which has row vectors g_e^T of coefficients for each person, and a tensor C. More explicitly, for the given problem, we use the following form of the HOSVD theorem

$$A = C\times_e G, \quad C = S\times_i F\times_p H, \tag{10}$$

where $\times_i = \times_1,\ \times_e = \times_2,\ \times_p = \times_3$ and $F = U^{(1)} \in \mathbb{R}^{n_i\times n_i}, G = U^{(2)} \in \mathbb{R}^{n_e\times n_e}, H = U^{(3)} \in \mathbb{R}^{n_p\times n_p}$ from the HOSVD theorem.

For a given person p we have:

$$A(:,:,p) = C(:,:,p)\times_e G. \tag{11}$$

Tensors $A(:,:,p)$ and $C(:,:,p)$ are, in fact, matrices denoted by A_p and, respectively C_p. Hence $A_p(:,e)$ is the image of person p in expression e and the columns of matrix C_p are basis vectors for person p (let us denote this basis by person basis). It follows that

$$A_p = C_p G^T, \quad p = 1,2,\ldots,n_p. \tag{12}$$

Let $G^T = (g_1 \ldots g_{n_e})$, then

$$A_p(:,e) = C_p g_e. \tag{13}$$

Thus $g_e, e = 1,2,\ldots,n_e$ are the coordinates of the image $A_p(:,e)$ of person p in expression e in the above mentioned basis.

Now, let $z \in \mathbb{R}^{n_i}$ be a picture from the testing set. We want to see if the picture is correctly identified. If it is an image of person p in expression e, then the coordinates of z in that basis are equal to g_e. So we can classify z by computing its coordinates in all the persons basis and checking, for each person, whether the coordinates of z coincide (or almost coincide) with the elements of any row of G (i.e. finding the nearest neighbour). The coordinates of z in this basis can be found by solving

$$\min_{\alpha_p}\left\|C_p\alpha_p - z\right\|_2. \tag{14}$$

For each image z we have to solve n_p least square problems with $C_p \in \mathbb{R}^{n_i\times n_e}$. From

$$C = S\times_i F\times_p H, \tag{15}$$

we get that

$$C_p = FB_p, \tag{16}$$

where $B_p \in \mathbb{R}^{n_e n_p\times n_e}, B_p = (S\times_p H)(:,:,p)$.

In order to reduce the computational effort we can truncate the tensors and matrices so that we obtain a truncated HOS-VD decomposition for tensor A. Let k be the level of truncation and $F_k = F(:, 1:k)$ and we obtain

$$\hat{C} = (S \times_p H)(1:k, :, :) \times_i F_k. \tag{17}$$

Hence, we have to solve

$$\min_{\alpha_p} \left\| \hat{C}_p \alpha_p - z \right\|_2. \tag{18}$$

Therefore, our proposed algorithm is the next one, for $n_i \ll n_e n_p$.

Tensor-ipe Algorithm

Let z be the image we are looking for. Choose k, the level of truncation.
for $p = 1, 2, \dots, n_p$
  Let $\hat{C} = (S \times_p H)(1:k, :, :) \times_i F_k$
  Solve $\min_{\alpha_p} \left\| \hat{C}_p \alpha_p - z \right\|_2$.
  for $e = 1, 2, \dots, n_e$
    if $\|\alpha_p - g_e\|_2 < tol$, then it is person p and STOP
  end for
end for.

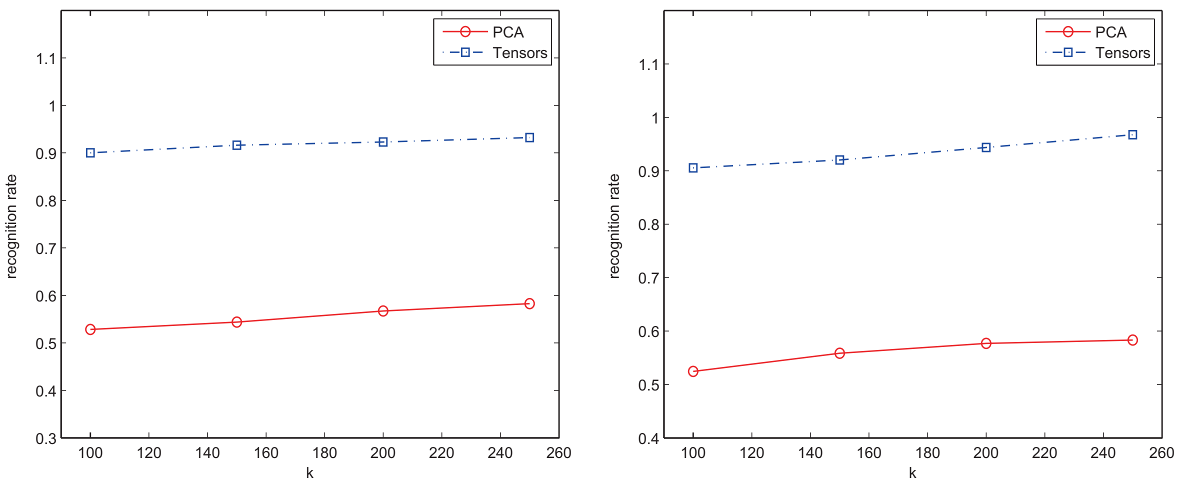
Remark 1. The level of truncation k can be chosen as in [25] or [26] or [27] or [28] and not empirically as in [23] (pages 116 and 173).

For the face recognition problem, if $n_i \ll n_e n_p$, it is better to use the new proposed Tensor-*ipe* algorithm, because it gives much better accuracy rather than the algorithm from [23] or PCA (see the Experiments section) and also the recognition running time is smaller than for the algorithm in [23]. This is because we have to solve n_p least squares problems with the matrix $\hat{C}_p \in \mathbb{R}^{n_i \times n_e}$ instead of n_e least squares problems ($n_e > n_p$ for ExtYaleB dataset) and $B_e \in \mathbb{R}^{n_e n_p \times n_p}$ [23] page 173), hence the dimension of B_e is bigger than the dimension of $\hat{C}_p$.

Since for $M \in \mathbb{R}^{s \times t}$ and $b \in \mathbb{R}^t$, the algorithm for solving $\|Mx - b\| = \min$ i.e. $x = M^+ b$ has the order of complexity $O(s^2 t + t^3)$, i.e. $O(t(s^2 + t^2))$ (for computing the Moore–Penrose pseudoinverse M^+ with SVD, like `pinv` command in Matlab),

Table 1
Datasets.

	ExtYaleB	Essex
# Subjects	38	224
# Images	2414	4480
Resolution	168 × 192	180 × 200



ExtYaleB dataset: 50 train ExtYaleB dataset: 55 train

Fig. 1. Comparisons of recognition rate for ExtYaleB dataset, Tensor-*ipe* and PCA (Eigenfaces) algorithms.

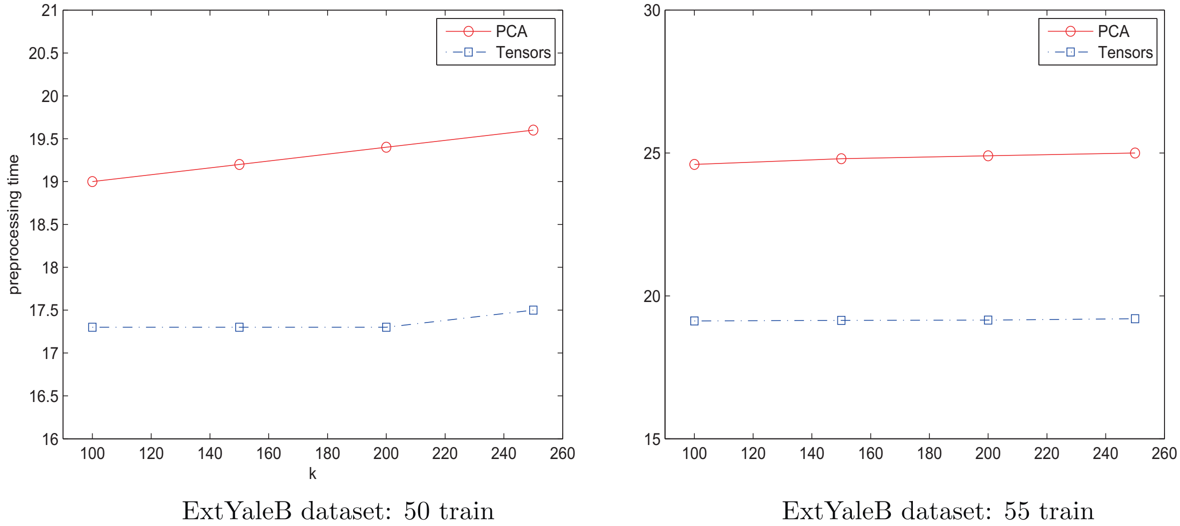


Fig. 2. Comparisons of preprocessing time (in seconds) for ExtYaleB dataset, Tensor-*ipe* and PCA (Eigenfaces) algorithms.

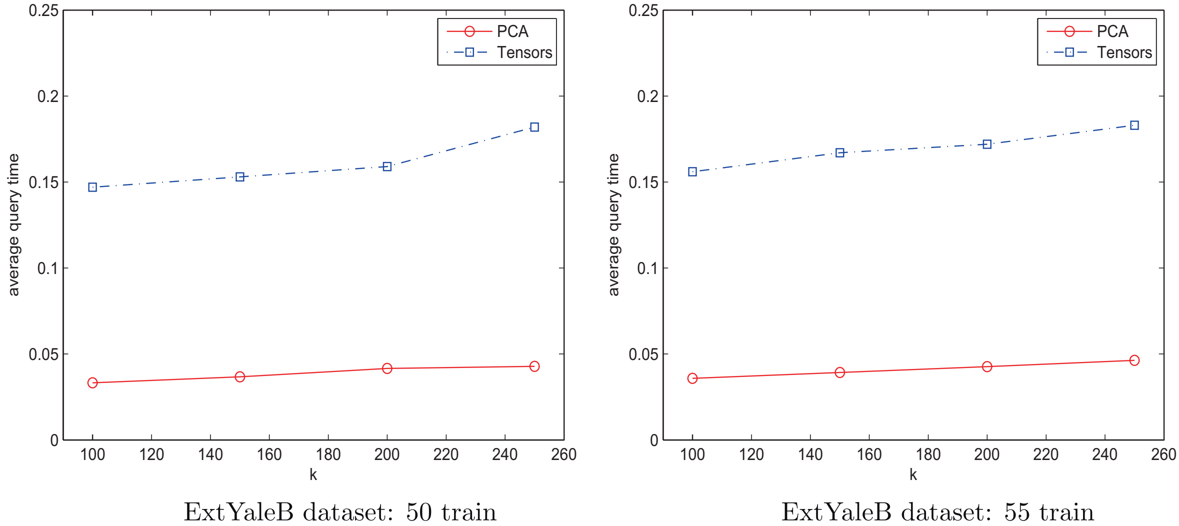


Fig. 3. Comparisons of average query time (in seconds) for ExtYaleB dataset, Tensor-*ipe* and PCA (Eigenfaces) algorithms.

plus $O(ts)$ (for matrix–vector multiplication), we conclude that the complexity for solving n_p problems with $C_p : n_i \times n_e$ is of order $O(n_p[n_e(n_i^2 + n_e^2) + n_i n_e])$ i.e.

$$O(n_p n_e(n_i^2 + n_e^2 + n_i)), \tag{19}$$

and the complexity for solving n_e problems with $B_e : n_e n_p \times n_p$ is of order $O\left(n_e\left[n_p\left(n_e^2 n_p^2 + n_p^2\right) + n_e n_p^2\right]\right)$ i.e.

$$O\left(n_e n_p\left(n_e^2 n_p^2 + n_p^2 + n_e n_p\right)\right). \tag{20}$$

Since for large dataset $n_i \ll n_e n_p$ and $n_e < n_p$, we state that our algorithm has a smaller complexity (see (19)) than the one proposed in [23] (see (20)). Of course, the conventional Eigenfaces is faster than our algorithm in terms of average query time (see Experiments section). This is happening because for tensors one has to solve many least squares problems, but in terms of recognition rate our method outperforms the standard Eigenfaces (therefore it is a promising tool for such endeavours).

6. Experiments

The motivation why tensors are used in face recognition is that often the data is stored as a tensor. Instead of depositing all the data as a matrix, we store all photos belonging to the same person as a matrix, and all matrices corresponding to all persons form a tensor. This leads to a better ordering of the pictures from the dataset. In this ordering, the faces are classified

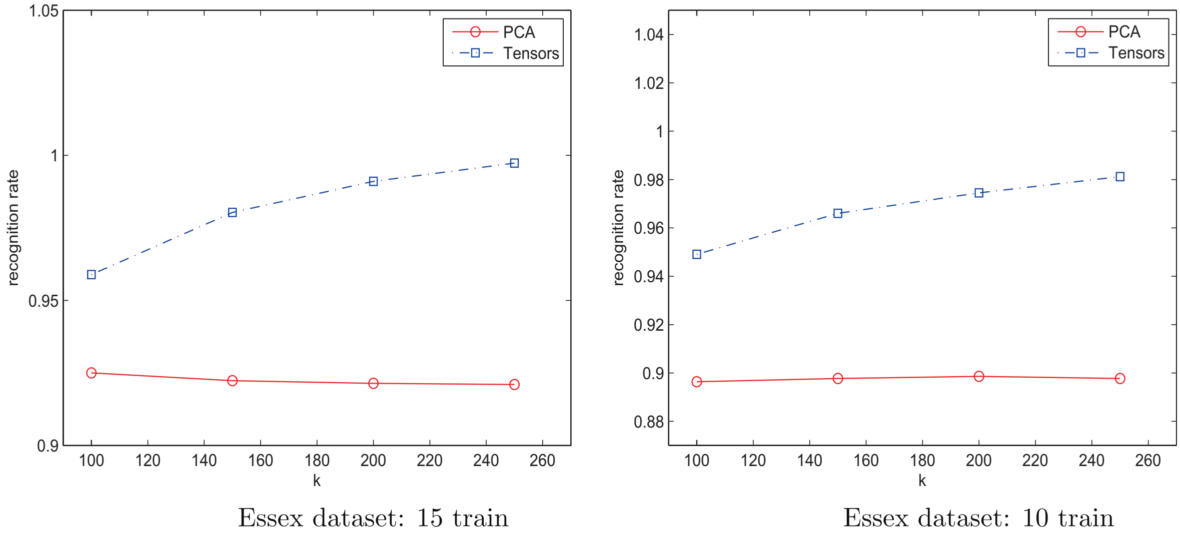


Fig. 4. Comparisons of recognition rate for Essex dataset, Tensor-*ipe* and PCA (Eigenfaces) algorithms.

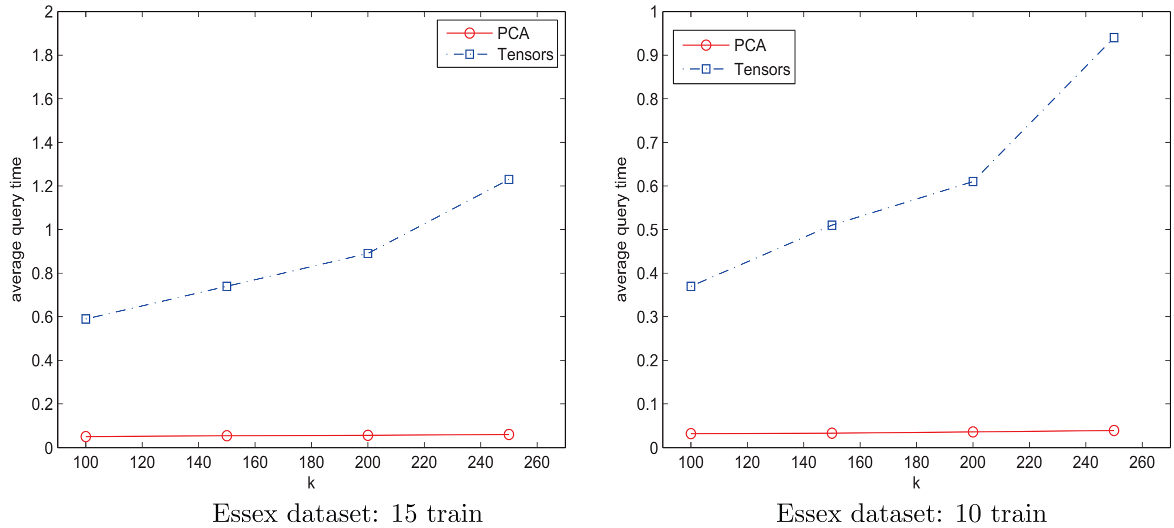


Fig. 5. Comparisons of average query time (in seconds) for Essex dataset, Tensor-*ipe* and PCA (Eigenfaces) algorithms.

into a number of groups of different "expressions". This classification refers to photographic angles (e.g. left-portrait, right-portrait, front-portrait), illumination conditions (dark, lighted, etc.), head pose, or facial expressions (happy, sad, angry, etc.) and so on. For instance, for ExtYaleB dataset the "expressions" refer to lighting conditions.

The experiments are performed in Matlab 7 on a Intel Core Duo 3.16 GHz and 2.83 GHz CPU with 3.25 GB memory.

We employ gray-level facial images from the two datasets.

The ExtYaleB[1] dataset [29,30] consists of 2414 images of the same resolution 168×192 of 38 persons. All images are cropped. In this dataset the pictures are taken under different variations of luminosity. In [31] the authors remark that better results can be obtained if the first three dominant eigenvectors are not taken into account. This note is also claimed in [32], hence for this dataset we follow this idea.

A part of the Essex Face dataset (a collection of facial images of Dr. Libor Spacek)[2] consists of 4480 images of the same resolution 180×200 of 224 persons. For the first 152 subjects "a sequence of images is taken while they sit at fixed distance from the camera and are asked to speak. The speech is used to introduce facial expression variation." For the next 72 subjects

[1] Available at http://vision.ucsd.edu/leekc/ExtYaleDatabase/ExtYaleB.html.

[2] Available at http://cswww.essex.ac.uk/mv/allfaces/.

"a sequence of 20 images per individual was taken using a fixed camera. During the sequence the subject takes one step forward towards the camera. This movement is used to introduce significant head (scale) variations between images of same individual. There is about 0.5 s between successive frames in the sequence."

The statistical information of these two datasets is tabulated in Table 1.

For the ExtYaleB dataset for the Eigenfaces and Tensor-*ipe* algorithms we resize the pictures (in order to fulfill $n_i \ll n_e n_p$). We combine the Tensor-*ipe* algorithm with the common metrics: L2 norm $d_{L2}(x,y) = \|x-y\|_2$ (Euclidian distance), L1 norm $d_{L1}(x,y) = \|x-y\|_1$ (City block), L∞ norm $d_{L\infty}(x,y) = \|x-y\|_\infty$ and cosine norm $d_{cos}(x,y) = 1 - \frac{x \cdot y}{\|x\|\|y\|}$. The Eigenfaces algorithm (PCA) gives better results in combination with the L1 norm [33]. For the Tensor-*ipe* algorithm we use the L2 norm [34]. For the ExtYaleB dataset we have two types of experiments: the first 50 images for training and the last 14 for testing, and the other is the first 55 images for training and the last 9 for testing.

The results for the recognition rate for the ExtYaleB dataset are depicted in Fig. 1. The Tensor-*ipe* algorithm has by far the best recognition rate, between 90% (training 50, $k = 100$) and 96.75% (training 55, $k = 250$), higher than the one for PCA. In terms of running time, we split the time into preprocessing time and average query time. The preprocessing time is dataset preparation stage, when we construct the smaller dimension subspace and project the dataset onto it. This stage is performed only once, before starting the search for a face/person. In Fig. 2 we have the results for the preprocessing time. In our case, the smaller preprocessing time is given by our Tensor-*ipe* algorithm. The average query time is the average of all query time when searching to identify a face/person. The results for the average query time are presented in Fig. 3. The smallest average query time is given by the PCA algorithm.

For the Essex dataset for both Eigenfaces and Tensor-*ipe* algorithms, we resize the pictures without altering the picture quality. For this dataset, we combine the Tensor-*ipe* algorithm with L2 norm $d_{L2}(x,y) = \|x-y\|_2$ (Euclidian distance) and the Eigenfaces algorithm (PCA) with the L1 norm $d_{L1}(x,y) = \|x-y\|_1$ (City block) [33]. For the Essex dataset we have two types of experiments: the first 15 images for training and the last 5 for testing, and the other is the first 10 images for training and the last 10 for testing.

The results for the recognition rate for the Essex dataset are the those shown in Fig. 4. Also for this dataset the Tensor-*ipe* algorithm has by far the best recognition rate, between 95% (training 10, $k = 100$) and 99% (training 15, $k = 250$), higher than the one for PCA. For the Essex dataset, because the number of images is quite large, the preprocessing stage takes a bit longer, but since this stage is performed only once we think this is not an inconvenient. This is because as long we do not enrich the dataset with some new images, it does not need to perform the preprocessing phase again. The results for the average query time are presented in Fig. 5. The smallest average query time is again given by the PCA algorithm.

The obtained results show that Tensor-*ipe* method is a promising tool for the face recognition problem when applied to big organized datasets with respect to one attribute (either facial expression, or illumination, or head pose).

7. Conclusions

We have approached the analysis of an ensemble of facial images organized as "expressions" as a problem in multilinear algebra in which the image ensemble is represented as a third order tensor. Using the Higher Order SVD algorithm (n-mode SVD) and the commutativity of tensor-matrix multiplication, our algorithm outperforms in terms of running time another third order algorithm, and in terms of recognition rate the above mentioned algorithm and the standard eigenfaces. Also, it is appropriate for the datasets with $n_i \ll n_e n_p$ and when there are not so many "structured" data for each individual (viewpoints, illuminations, expressions) as the one used by Vasilescu et al. in [17,21]. The results are promising, and for Essex datasets the recognition rates are almost 99% for test subjects.

As future work, we intend to perform the same analysis when the two datasets are combined, i.e. for a part of the new dataset the "expressions" refer to facial expressions, and for the other part, they refer to viewpoint or illumination.

Acknowledgement

The authors want to thank the reviewers for their insightful remarks and comments on this paper. Also, the authors are grateful to Alex Vasilescu for her valuable recommendations.

References

[1] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley-Interscience, New York, 2001.
[2] Murty M. Narasimha, Devi V. Susheela, Pattern Recognition, An Algorithmic Approach, First Edition., Springer, 2011.
[3] K. Karhunen, Über lineare Methoden in der Wahrscheinlichkeitsrechnung, Ann. Acad. Sci. Fenn., Ser. A.1. Math.-Phys. 37 (1947) 3–79.
[4] M. Kirby, L. Sirovich, Application of the Karhunen–Loeve procedure for the characterization of human faces, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1) (1990) 103–108.
[5] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cogn. Neurosci. 1 (1991) 71–86.
[6] L. Sirovich, M. Kirby, Low-dimensional procedure for the characterization of human faces, J. Opt. Soc. Am. A – Opt., Image Sci. Vision 4 (3) (1987) 519–524.
[7] R. Chellappa, C.L. Wilson, S. Sirohey, Human and machine recognition of faces: a survey, Proc. IEEE 83 (5) (1995) 705–740.
[8] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM REV. 51 (3) (2009) 455–500.
[9] B.W. Bader, T.G. Kolda, Algorithm 862: MATLAB tensor classes for fast algorithm prototyping, ACM Trans. Math. Softw. 32 (4) (2006) 635653.
[10] T.G. Kolda, Orthogonal tensor decompositions, SIAM J. Matrix Anal. Appl. 23 (1) (2001) 243–255.

[11] P.N. Belhumeur, J. Hespanha, D.J. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, in: Proceedings of the European Conference on Computer Vision, 1996, pp. 45–58.
[12] A. Shashua, A. Levin, Linear image coding for regression and classification using the tensor-rank principle, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hawai, 2001, pp. 42–49.
[13] L.R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (1966) 279–311.
[14] A. Kapteyn, H. Neudecker, T. Wansbeek, An approach to n-mode component analysis, Psychometrika 51 (2) (1986) 269–275.
[15] L. de Lathauwer, B. de Moor, J. Vandewalle, A multilinear singular value decomposition, SIAM J. Matrix Anal. Appl. 21 (2000) 1252–1278.
[16] L. de Lathauwer, B. de Moor, J. Vandewalle, On the best rank-l and rank-$(R_l, R_2, \ldots, R_n)$ approximation of higher order tensors, SIAM J. Matrix Anal. Appl. 21 (4) (2000) 132–1342.
[17] M.A.O. Vasilescu, D. Terzopoulos, Multilinear analysis of images ensembles: tensorfaces, in: A. Heyden et al. (Eds.), Proceedings of 7th European Conference on Computer Vision (ECCV'02), Copenhagen, Denmark, May, 2002, in Computer Vision – ECCV 2002, Lecture Notes in Computer Science, vol. 2350, Springer-Verlag, Berlin, 2002, pp. 447–460.
[18] M.A.O. Vasilescu, An algorithm for extracting human motion signatures, in: IEEE International Conference on Computer Vision and Pattern Recognition, Hawai, 2001.
[19] M.A.O. Vasilescu, Human motion signatures for character animation, ACM SIGGRAPH 2001 sketches and applications, Los Angeles, CA, 2001.
[20] M.A.O. Vasilescu, D. Terzopoulos, TensorTextures: multilinear image-based rendering, in: Proc. ACM SIGGRAPH 2004 Conference Los Angeles, CA, August, 2004, in Computer Graphics Proceedings, Annual Conference Series, 2004.
[21] M.A.O. Vasilescu, D. Terzopoulos, Multilinear independent components analysis, in; Proceedings of the IEEE Computer Vision and Pattern Recognition Conf. (CVPR 05), vol. 1, San Diego, CA, June 2005, p. 547553.
[22] M. Turk, A. Pentland, Face recognition using eigenfaces, Comput. Vision Pattern Recogn. Proc. (1991) 586–591.
[23] L. Elden, Matrix Methods in Data Mining and Pattern Recognition, SIAM, Philadelphia, 2007.
[24] J. Chen, Y. Saad, Lanczos Vectors versus singular vectors for effective dimension reduction, IEEE Trans. Knowl. Data Eng. 21 (8) (2009) 1091–1103.
[25] E. Pelican, L. Grecu, Low-rank matrix methods in pattern recognition, in: Balkan Conference on Operational Research (BALCOR), 2009, ISBN: 973-86979-9-9.
[26] E. Pelican, L. Grecu, Comparison between some matrix methods with applications in pattern recognition, in: Applied Linear Algebra Conference, May 24–28, Novi Sad, 2010.
[27] K. Delac, M. Grgic, P. Liatsis, Appearance-based statistical methods for face recognition, in: Proceedings of the 47th International Symposium ELMAR-2005 Focused on Multimedia Systems and Applications, Zadar, Croatia, 2005, pp. 151–158.
[28] F. Glineur, L. Lu, P. Van Dooren, X. Wang, Extended lanczos bidiagonalization for dimension reduction in information retrieval, in: 8th International Conference on Natural Computation (ICNC'12), Maui, Hawaii, 2012.
[29] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 643–660.
[30] K.C. Lee, J. Ho, D. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 684–698.
[31] A. Pentland, T. Starner, N. Etcoff, A. Masoiu, O. Oliyide, M. Turk, Experiments with Eigenfaces, in: Looking at People Workshop, IJCAI'93, 1993.
[32] A.M. Martinez, A.C. Kak, PCA versus LDA, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2) (2001) 228–233.
[33] K. Delac, M. Grgic, S. Grgic, Independent comparative study of PCA, ICA, and LDA on the FERET data set, Int. J. Imaging Syst. Technol. 15 (5) (2006) 252–260.
[34] E. Pelican, L. Liţă (Grecu), Solving the pattern recognition problem with some low-rank approximation based algorithms, XIème Colloque Franco-Roumain de Mathématiques Appliquées, 24–30 August, Bucarest, 2012.