IPSJ Transactions on Computer Vision and Applications Vol. 1 231–241 (Sep. 2009)

Research Paper

Generalized N-Dimensional Principal Component Analysis (GND-PCA) Based Statistical Appearance Modeling of Facial Images with Multiple Modes

Xu Qiao,[†1] Rui Xu,[†1] Yen-Wei Chen,[†1]
Takanori Igarashi,[†2] Keisuke Nakao[†2]
and Akio Kashimoto[†2]

This paper introduces a framework called generalized N-dimensional principal component analysis (GND-PCA) for statistical appearance modeling of facial images with multiple modes including different people, different viewpoint and different illumination. The facial images with multiple modes can be considered as high-dimensional data. GND-PCA can represent the high-order dimensional data more efficiently. We conduct extensive experiments on MaVIC Database (KAO-Ritsumeikan Multi-angle View, Illumination and Cosmetic Facial Database) to evaluate the effectiveness of the proposed algorithm and compared the conventional ND-PCA in terms of reconstruction error. The results indicated that the extraction of data features is computationally more efficient using GND-PCA than PCA and ND-PCA.

1. Introduction

To interpret images of faces, it is important to have a model of how the face can appear. Faces vary widely, but differences can be represented by texture (patterns of pixel values) across the face. The texture can vary because of differences between individuals and due to changes in expression, viewpoint, cosmetics and illumination.

Principal component analysis (PCA) method[1] is an efficient method to build statistical appearance models of human faces. In the PCA-based face representation and recognition methods, the 2D face image matrices must be previously transformed into 1D image vectors column by column[2]. Such unfolding process causes two problems: one is the huge calculation cost and another is the poor performance to generalize.

To overcome these problems, a new technique called 2-dimensional principal component analysis (2D-PCA)[3] has been proposed, which directly computes eigenvectors of the covariance matrix of the image without matrix-to-vector conversion. It was reported that the recognition accuracy with 2D-PCA on several face databases was higher than that with conventional 1D-PCA. However, the main disadvantage of 2D-PCA is that it needs many more coefficients for image representation. A method called generalized 2-dimensional principal component analysis (G2D-PCA)[4] has been proposed to find optimal basis for both row- and column-mode subspaces.

Recently, a method called N-dimensional PCA (ND-PCA) was proposed for high-dimensional data analysis[5]. In this method the high dimensional data is treated as a higher-order tensor which is directly trained to obtain the bases on one mode-subspace by the higher-order singular value decomposition (HOSVD)[6),7]. It was applied to 3D facial scanning data. The facial images with multiple modes can be considered as high-dimensional data (or a tensor). With the help of HOSVD or multilinear algebra, Vasilescu and Terzopoulos proposed tensorface method[10] to build a statistical model of facial images in different conditions, such as illumination, viewpoint and expression. Since ND-PCA only compresses the data on one mode-subspace, it also suffered from the problem that the data can't be represented efficiently, similar to the problem of 2D-PCA.

Inspired from the works of G2D-PCA and ND-PCA, we proposed a new method called generalized N-dimensional principal component analysis (GND-PCA)[11]. The high-dimensional data is treated as a series of higher-order tensors and the bases on each mode-subspace are calculated in order to approximate the tensors accurately. ND-PCA is a unilateral-projection-based scheme. Compared with ND-PCA, GND-PCA is a N-lateral-projection-based scheme. Experiments show that GND-PCA can represent the data more efficiently as compared to ND-PCA method and has good performance. Another benefit of GND -PCA is that the features obtained by GND-PCA is more meaningful than ones obtained by ND-PCA to represent the different factors such as viewpoint, expression and illumination, etc.

[†1] Graduate School of Science and Engineering, Ritsumeikan University
[†2] Beauty Cosmetic Research Lab, Kao Corporation

In this paper, we first introduce the basic knowledge of multilinear algebra and HOSVD method. Then we introduce GND-PCA method in Section 3. Experiments and the results are shown in Section 4. Comparison with other relative methods is presented in Section 5. Finally, the conclusion is presented in the last section.

2. Multilinear Algebra and HOSVD

We introduce some background knowledge and some related works in this section.

(1) Definition of Tensor

A tensor can be consider as a higher order extension of a vector (first order tensor) and a matrix (second order tensor).

An N-th order tensor $\mathcal{A}$ is defined as a multi-array with N indices, where $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and $\mathbb{R}$ is the real manifold. Elements of the tensor $\mathcal{A}$ are denoted as $a_{i_1 \ldots i_n \ldots i_N}$, where $1 \leqslant i_n \leqslant I_n$. The space of the N-th order tensor is comprised by the N mode subspaces. From the perspective of $\mathcal{A}$, scalars, vectors and matrices can be seen as zeroth-order, first order and second order tensors respectively.

Varying the $n - th$ index i_n and keeping the other indices fixed, we can obtain the "mode-n vectors" of the tensor $\mathcal{A}$. The "mode-n matrix" $\mathbf{A}_{(n)}$ can be formed by arranging all the mode-n vectors sequentially as its columns, $\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \cdot \ldots I_{n-1} \cdot I_{n+1} \cdot \ldots I_N)}$. The procedure of forming the mode-n matrix is called unfolding of a tensor.

Figure 1 gives the examples to show how to unfold the third order to their mode-n matrices. The mode-n product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$, denoted as $\mathcal{A} \times_n \mathbf{U}$, 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. Entries of the new tensor is defined by $(\mathcal{A} \times_n \mathbf{U})_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} \overset{\text{def}}{=} \sum_{i_n} a_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} u_{j_n i_n}$. These entries can also be calculated by matrix product, $\mathbf{B}_{(n)} = \mathbf{U} \cdot \mathbf{A}_{(n)}$, $\mathbf{B}_{(n)}$ is mode-n matrix of the tensor $\mathcal{B} = \mathcal{A} \times_n \mathbf{U}$.

The mode-n product has two properties. One can be expressed by $(\mathcal{A} \times_n \mathbf{U}) \times_m \mathbf{V} = (\mathcal{A} \times_m \mathbf{V}) \times_n \mathbf{U} = \mathcal{A} \times_n \mathbf{U} \times_m \mathbf{V}$; and the other is $(\mathcal{A} \times_n \mathbf{U}) \times_n \mathbf{V} = \mathcal{A} \times_n (\mathbf{V} \cdot \mathbf{U})$. The inner product of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is defined by $\langle \mathcal{A}, \mathcal{B} \rangle \overset{\text{def}}{=} \sum_{i_1} \sum_{i_2} \ldots \sum_{i_N} a_{i_1 i_2 \ldots i_N} b_{i_1 i_2 \ldots i_N}$.

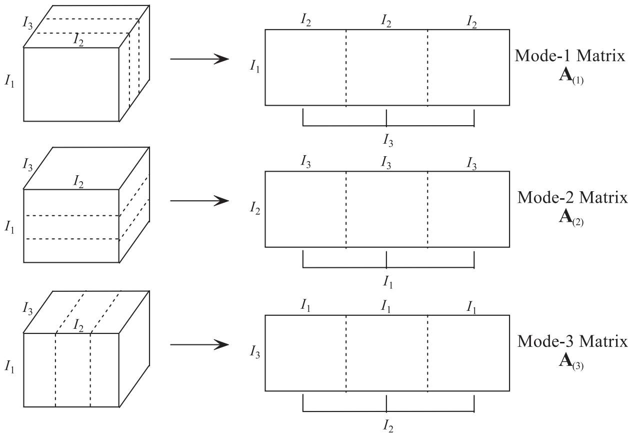


Fig. 1 Example of Unfolding the Third Order Tensor to the Three Mode-n Matrices.

The Frobenius-norm of a tensor $\mathcal{A}$ is defined by $||\mathcal{A}|| \overset{\text{def}}{=} \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$. The Frobenius-norm of a tensor can also be calculated from its mode-n matrix, $||\mathcal{A}|| = ||\mathbf{A}_{(n)}|| = \sqrt{tr(\mathbf{A}_{(n)} \cdot \mathbf{A}_{(n)}^T)}$, $tr(.)$ is the trace of a matrix.

A Nth-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ can be decomposed by the tucker model, which can be expressed by Eq. (1).

$$\mathcal{A} = \mathcal{B} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times \ldots \times \mathbf{U}^{(N)} \qquad (1)$$

where $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times J_n}$ and $\mathcal{B} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_N}$ is the core tensor.

Much more details are referred to 8), 9).

(2) HOSVD and ND-PCA Method

When the orthogonal constraint for the matrices $\mathbf{U}^{(n)}$ is required, there are two kinds of methods for the tucker decomposition. One method is called higher-order SVD (HOSVD)[6] in which the column vectors of $\mathbf{U}^{(n)}$ are obtained by the singular value decomposition (SVD) for the mode-n matrices $\mathbf{A}_{(n)}$ and $J_n = I_n$. The other method is called lower rank-$(J_1, J_2, \ldots, J_N)$ tensor approximation[7] in which the matrices $\mathbf{U}^{(n)}$ is determined by finding a lower rank-$(J_1, J_2, \ldots, J_N)$ tensor $(J_n < I_n)$ to most accurately approximate the original tensor.

In linear algebra, Singular Value Decomposition (SVD) is an important factorization of a rectangular real or complex matrix, with several applications in signal processing and statistics. SVD computes the low-rank approximation of a set of 1D vectors. This can be generalized to a two dimensional Singular Value Decomposition (2DSVD) to do low-rank approximation of a set of matrices such as a set of images. Higher Order Singular Value Decomposition (HOSVD) is a generalization of SVD for high dimensional tensor [6]. In the case of a 3D tensor executing HOSVD in 2 dimensions gives the same result as 2DSVD.

ND-PCA [5] is proposed for the modeling of higher-dimensional data. This method is based on HOSVD and the data is treated as the high-order tensor. Given a series of higher-order tensors with zero-mean value, $\mathcal{A}_i \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}, i = 1, 2, \ldots, M$ and $\sum_{i=1}^{M} \mathcal{A}_i = 0$, a $(N+1) - th$ order tensor is firstly formed from the data and then HOSVD is applied on its mode-n subspace. The first leading J, where $J < I_n$, eigenvectors $\mathbf{U}^{(n)} = [\mathbf{u}_1^{(n)}, \mathbf{u}_2^{(n)}, \ldots, \mathbf{u}_J^{(n)}] \in \mathbb{R}^{I_n \times J}$ is the bases on the mode-n subspace. A tensor $\mathcal{A}_i \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ can be compactly represented by the tensor $\mathcal{B}_i = \mathcal{A}_i \times_n {\mathbf{U}^{(n)}}^T$, $\mathcal{B}_i \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_{n-1} \times J \times I_{n+1} \times \ldots \times I_N}$ whose components are the projections (coefficients) onto the mode-n subspaces for ND-PCA. Since HOSVD can be calculated efficiently, ND-PCA does not suffers from the computing cost problem; however, ND-PCA only compress the data on the mode-n subspaces, so ND-PCA needs lots of components to represent the data, which is similar to the problem of 2DPCA.

3. GND-PCA Algorithm

(1) Theory Foundation

We formalize GND-PCA [11] as follows:

Given a series of the $N - th$ order tensors with zero-mean [⋆1] $\mathcal{A}_i \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}, i = 1, 2, \ldots, M$, $\sum_{i=1}^{M} \mathcal{A}_i = 0$, we hope to find another series of lower rank-$(J_1, J_2, \ldots, J_N)$ tensors $\hat{\mathcal{A}}_i$ which can most accurately approximate the original tensors, where $J_n < I_n$. The new series of tensors can be decomposed by the same N matrices $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times J_n}$ with orthogonal columns according to tucker model which can be shown by Eq. (2).

$$\hat{\mathcal{A}}_i = \mathcal{B}_i \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times \ldots \times_n \mathbf{U}^{(n)} \times \ldots \times_N \mathbf{U}^{(N)} \tag{2}$$

where $\mathcal{B}_i \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_n \times \ldots \times J_N}$ are the core tensors.

The orthogonal matrices $\mathbf{U}^{(n)}$ can be determined by minimizing the cost function shown by Eq. (3). The cost function is defined as a mean square error between the original tensor and the reconstructed tensor.

$$S = \sum_{i=1}^{M} ||\mathcal{A}_i - \hat{\mathcal{A}}_i||^2 = \sum_{i=1}^{M} ||\mathcal{A}_i - \mathcal{B}_i \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times \ldots \times_N \mathbf{U}^{(N)}||^2 \tag{3}$$

In Eq. (3), only the tensors $\mathcal{A}_i$ are known. However, supposing the N matrices $\mathbf{U}^{(n)}$ are known, the answer of $\mathcal{B}_i$ to minimize Eq. (3) is merely the result of the traditional linear least-square problem. Theorem 3.1 can be obtained.

Theorem 3.1. *Given fixed N matrices $\mathbf{U}^{(n)}$, the tensors $\mathcal{B}_i$ that minimize the cost function, Eq. (3), are given by $\mathcal{B}_i = \mathcal{A}_i \times_1 {\mathbf{U}^{(1)}}^T \times_2 {\mathbf{U}^{(2)}}^T \times \ldots \times_N {\mathbf{U}^{(N)}}^T$*

The proof of Theorem 3.1 is simple, so it is omitted. With the help of Theorem 3.1, Theorem 3.2 can be obtained.

Theorem 3.2. *If the tensors $\mathcal{B}_i$ are chosen as $\mathcal{B}_i = \mathcal{A}_i \times_1 {\mathbf{U}^{(1)}}^T \times_2 {\mathbf{U}^{(2)}}^T \times \ldots \times_N {\mathbf{U}^{(N)}}^T$, minimization of the cost function Eq. (3) is equal to maximization of the following cost function S', where*

$$S' = \sum_{i=1}^{M} ||\mathcal{A}_i \times_1 {\mathbf{U}^{(1)}}^T \times_2 {\mathbf{U}^{(2)}}^T \times \ldots \times_N {\mathbf{U}^{(N)}}^T||^2.$$

There is no close-form solution to simultaneously resolve the matrices $\mathbf{U}^{(n)}$ for the cost function S'; however the explicit solution for one matrix can be obtained if the other matrices are fixed. This is expressed by Lemma 3.3.

Lemma 3.3. *Given the fixed matrices, $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(n-1)}, \mathbf{U}^{(n+1)}, \mathbf{U}^{(N)}$, if the columns of the matrix $\mathbf{U}^{(n)}$ are selected as the first J_n leading eigenvectors of the matrix $\sum_{i=1}^{M} (\mathbf{C}_{i(n)} \cdot \mathbf{C}_{i(n)}^T)$, $\mathbf{C}_{i(n)}$ is the mode-n matrix of the tensor $\mathcal{C}_i =$*

[⋆1] if the tensors do not have zero-mean, we can subtract the mean-value from each tensor to obtain a new series of tensors $\mathcal{A}'_i$ which have zero-mean, i.e. $\mathcal{A}'_i = \mathcal{A}_i - \frac{1}{M} \sum_{i=1}^{M} \mathcal{A}_i$

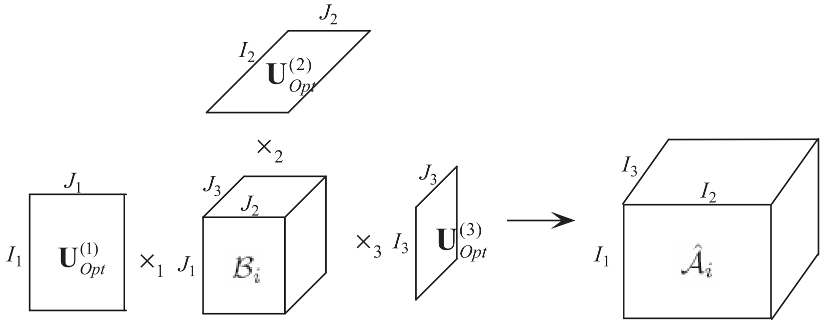


Fig. 2 Illustration of Reconstructing a Third Order Tensor by the Three Orthogonal Bases of Mode Subspaces $\mathbf{U}^{(1)}_{opt}$, $\mathbf{U}^{(2)}_{opt}$, $\mathbf{U}^{(3)}_{opt}$ and the Projections $\mathcal{B}_i$.

$\mathcal{A}_i \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} \times \ldots \times_{n-1} \mathbf{U}^{(n-1)^T} \times_{n+1} \mathbf{U}^{(n+1)^T} \times \ldots \times_N \mathbf{U}^{(N)^T}$*, the cost function* S' *can be maximized.*

The proofs of Theorem 3.2 and Lemma 3.3 are shown in Appendix A.1 and A.2 respectively.

(2) Iteration Algorithm

According to Lemma 3.3 we can use an iteration algorithm to get the N optimal matrices, $\mathbf{U}^{(1)}_{opt}$,$\mathbf{U}^{(2)}_{opt}, \ldots, \mathbf{U}^{(N)}_{opt}$, which are able to maximize the cost function S'. This algorithm is summarized as Algorithm 1.

Using the calculated matrices $\mathbf{U}^{(n)}_{opt}, n = 1, 2, \ldots, N$, each of the volume $\mathcal{A}_i$ can be approximated by $\hat{\mathcal{A}}_i$ with least errors, where $\hat{\mathcal{A}}_i = \mathcal{B}_i \times_1 \mathbf{U}^{(1)}_{opt} \times_2 \mathbf{U}^{(2)}_{opt} \times \ldots \times_N \mathbf{U}^{(N)}_{opt}$ and $\mathcal{B}_i = \mathcal{A}_i \times_1 \mathbf{U}^{(1)^T}_{opt} \times_2 \mathbf{U}^{(2)^T}_{opt} \times \ldots \times_N \mathbf{U}^{(N)^T}_{opt}$. The approximation can be illustrated by **Fig. 2** for the 3-dimensional case. In GND-PCA, the matrices $\mathbf{U}^{(n)}_{opt}, n = 1, 2, \ldots, N$ construct the bases on the N mode-subspaces; and the core tensor $\mathcal{B}_i$ are the projections on these mode-subspaces.

4. Experiments and Results

(1) MaVIC Database

The database we used is MaVIC (KAO-Ritsumeikan Multi-angle View, Illumination and Cosmetic Facial Database)[12]. The facial images in the database are captured from various angles of view and various illuminations with a "multi-angle image capturing system (MICS)"[21]. In MaVIC, there are 170 Japanese women's natural facial images and 250 Japanese women's cosmetic facial images.

Algorithm 1 Iteration Algorithm to Compute the N Matrices $\mathbf{U}^{(1)}_{opt}, \mathbf{U}^{(2)}_{opt}, \ldots, \mathbf{U}^{(N)}_{opt}$

```
IN: a series of $N - th$ order tensors, $\mathcal{A}_i \in \mathbb{R}^{I_1 \times\ I_2 \times \ldots \times I_N}, i = 1, 2, \ldots, M$.
OUT: $N$ Matrices $\mathbf{U}^{(n)}_{opt} \in \mathbb{R}^{I_n \times J_n}$ $(J_n < I_n, n = 1, 2, \ldots, N)$ with orthogonal column vectors.
(a) Initial values: $k = 0$ and $\mathbf{U}^{(n)}_0$ whose columns are determined as the first $J_n$ leading eigenvectors of the matrices $\sum_{i=1}^{M} (\mathbf{A}_{i(n)} \cdot \mathbf{A}^T_{i(n)})$.
(b) Iterate for $k$ until convergence
      • Maximize $S' = \sum_{i=1}^{M} \| \mathcal{C}_i \times_1 \mathbf{U}^{(1)^T} \|^2$, $\mathcal{C}_i = \mathcal{A}_i \times_2 \mathbf{U}^{(2)^T}_k \times \ldots \times_N \mathbf{U}^{(N)^T}_k$
        Solution: $\mathbf{U}^{(1)}$ whose columns are determined as the first $J_1$ leading eigenvectors of $\sum_{i=1}^{M} (\mathbf{C}_{i(1)} \cdot \mathbf{C}^T_{i(1)})$
        Set $\mathbf{U}^{(1)}_{k+1} = \mathbf{U}^{(1)}$.
      • Maximize $S' = \sum_{i=1}^{M} \| \mathcal{C}_i \times_2 \mathbf{U}^{(2)^T} \|^2$, $\mathcal{C}_i = \mathcal{A}_i \times_1 \mathbf{U}^{(1)}_{k+1}{}^T \times_3 \mathbf{U}^{(3)^T}_k \times \ldots \times_N \mathbf{U}^{(N)^T}_k$
        Solution: $\mathbf{U}^{(2)}$ whose columns are determined as the first $J_2$ leading eigenvectors of $\sum_{i=1}^{M} (\mathbf{C}_{i(2)} \cdot \mathbf{C}^T_{i(2)})$
        Set $\mathbf{U}^{(2)}_{k+1} = \mathbf{U}^{(2)}$.
        ...
      • Maximize $S' = \sum_{i=1}^{M} \| \mathcal{C}_i \times_n \mathbf{U}^{(n)^T} \|^2$, $\mathcal{C}_i = \mathcal{A}_i \times_1 \mathbf{U}^{(1)}_{k+1}{}^T \times \ldots \times_{n-1} \mathbf{U}^{(n-1)^T}_{k+1} \times_{n+1} \mathbf{U}^{(n+1)^T}_k \times \ldots \times_N \mathbf{U}^{(N)^T}_k$
        Solution: $\mathbf{U}^{(n)}$ whose columns are determined as the first $J_n$ leading eigenvectors of $\sum_{i=1}^{M} (\mathbf{C}_{i(n)} \cdot \mathbf{C}^T_{i(n)})$
        Set $\mathbf{U}^{(n)}_{k+1} = \mathbf{U}^{(n)}$.
        ...
      • Maximize $S' = \sum_{i=1}^{M} \| \mathcal{C}_i \times_N \mathbf{U}^{(N)^T} \|^2$, $\mathcal{C}_i = \mathcal{A}_i \times_1 \mathbf{U}^{(1)}_{k+1}{}^T \times \ldots \times_{N-1} \mathbf{U}^{(N-1)^T}_{k+1}$
        Solution: $\mathbf{U}^{(N)}$ whose columns are determined as the first $J_N$ leading eigenvectors of $\sum_{i=1}^{M} (\mathbf{C}_{i(N)} \cdot \mathbf{C}^T_{i(N)})$
        Set $\mathbf{U}^{(N)}_{k+1} = \mathbf{U}^{(N)}$.
    $k = k + 1$
(c) Set $\mathbf{U}^{(1)}_{opt} = \mathbf{U}^{(1)}_k$, $\mathbf{U}^{(2)}_{opt} = \mathbf{U}^{(2)}_k$, ..., $\mathbf{U}^{(N)}_{opt} = \mathbf{U}^{(N)}_k$.
```

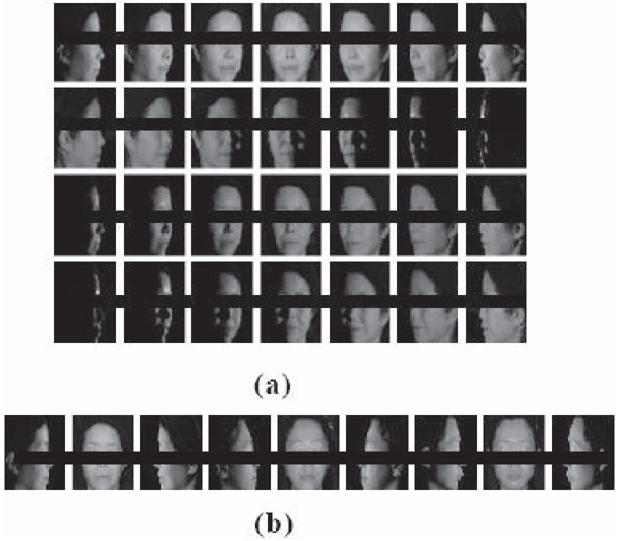


Fig. 3 Example images of MaVIC Database. (a) One sample with different poses (different columns) and different illuminations (different rows). (b) Different samples.

Each subject was photographed in 13 different view-points under 14 illuminations. Some sample images from MaVIC Database are shown in **Fig. 3**.

(2) Statistical Appearance Models by GND-PCA

The proposed GND-PCA is used to construct statistical appearance models for the 3D-Tensor data. In our experiment, we used a portion of the MaVIC database. 80 samples (subjects) with natural facial images are used and each sample has 13 viewpoints (poses) and 13 illuminations. The size of each image is 60 × 50 pixels. The 2D image is unfolded into a vector with the dimension of 3000. All images of each sample are written together as a 3D-Tensor. Then we put the vectors in different directions according to the following rules: mode-1 for different viewpoints, mode-2 for image textures (pixels) and mode-3 for different illuminations. Each sample is corresponding to a 3D tensor with the dimension of 3000 × 13 × 13. We use 79 samples for training and another 1 sample for Statistical Appearance Model Testing.

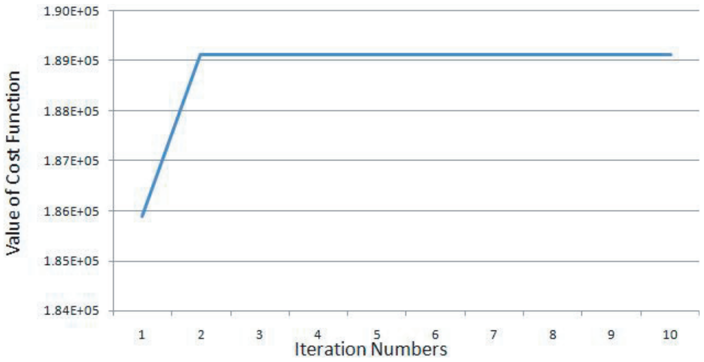


Fig. 4 Convergence of GND-PCA (when 100 × 5 × 5 mode subspace bases are trained).

As shown in **Fig. 4**, it can be seen that the cost function is not dramatically changed after two times of iterations. So that we usually set the iteration times of GND-PCA to two in the experiments.

We choose one sample as the template and use the others for training. The leave-one-out experiment is used to test the generalization ability of the constructed models. The accuracy of the reconstructed tensor from the reduced mode-subspace is measured by a normalized correlation. Normalized correlation of two tensor-formed data, the original $\mathbf{t}_{Ori}(x, y, z)$ and the reconstructed $\mathbf{t}_{Rec}(x, y, z)$, is defined as

$$NC = \frac{\sum_{x,y,z} \mathbf{t}_{Ori}(x, y, z) \cdot \mathbf{t}_{Rec}(x, y, z)}{\sqrt{\sum_{x,y,z} \mathbf{t}^2_{Ori}(x, y, z)} \cdot \sqrt{\sum_{x,y,z} \mathbf{t}^2_{Rec}(x, y, z)}}.$$

The value of NC is between 0 and 1. The more similar the two data, the larger the value of normalized correlation. **Figure 6** shows Normalized correlation between the original ensemble and the reconstructed ensembles when different number of mode-subspace bases are used. **Figure 5** shows the reconstructed results from different mode-subspace bases respectively. It can be seen that normalized correlation becomes larger and larger when the number of bases is increased. When 3000 × 13 × 13 mode-subspace bases are used, normalized correlation becomes 1, which means an untrained data can be fully reconstructed without reconstructed errors if all the mode-subspace bases are used.

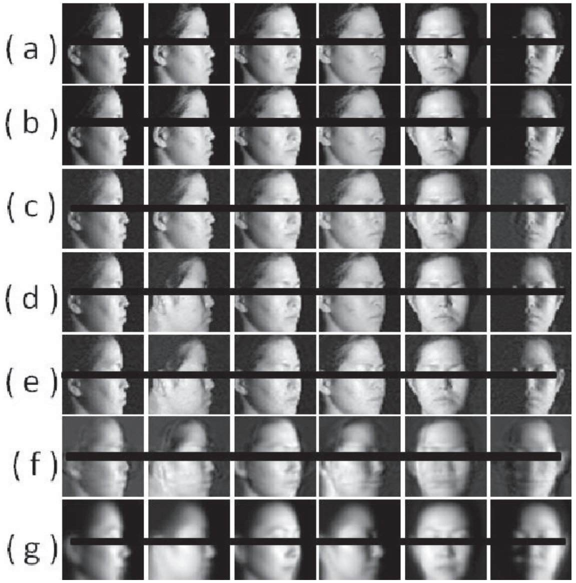


Fig. 5 Some images of reconstructed results by GND-PCA. (a) Some original images from MaVIC. First image is under illumination 0^o and viewpoint 0^o, noted as $(I0^o, V0^o)$. The following five image are under $(I30^o, V0^o)$, $(I0^o, V30^o)$, $(I30^o, V30^o)$, $(I90^o, V90^o)$, $(I180^o, V90^o)$ respectively. (b) Images reconstructed by GND-PCA (Mode-subspace bases: $3000 \times 13 \times 13$). Compressing Rate (CR) is 1 and normalized correlation (NC) is 1. (c) Images reconstructed by GND-PCA (Mode-subspace bases: $1200 \times 5 \times 5$). CR is 5.92% and NC is 0.987. (d) Images reconstructed by GND-PCA (Mode-subspace bases: $600 \times 5 \times 10$). CR is 5.92% and NC is 0.992. (e) Images reconstructed by GND-PCA (Mode-subspace bases: $600 \times 5 \times 10$). CR is 5.92% and NC is 0.989. (f) Images reconstructed by GND-PCA (Mode-subspace bases: $100 \times 5 \times 5$). CR is 0.5% and NC is 0.977. (g) Mean images of training set.

Figure 7 shows a series of normalized correlations while increasing the training samples and we can find that the constructed models have good performance for generalization even though the training samples are quite few by using the GND-PCA method. In the following experiments, we use 19 samples for training and another 5 samples for testing in order to comparing GND-PCA with PCA and ND-PCA.

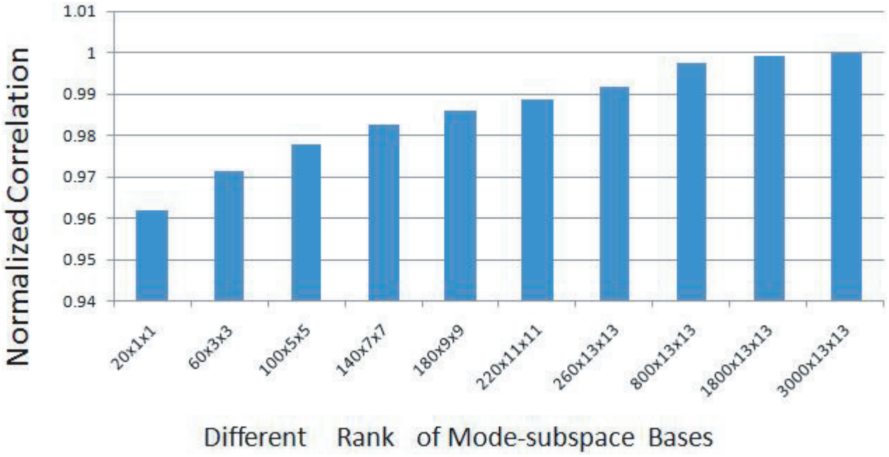


Fig. 6 Normalized correlation between the original ensemble and the reconstructed ensembles when different number of mode-subspace bases are used.

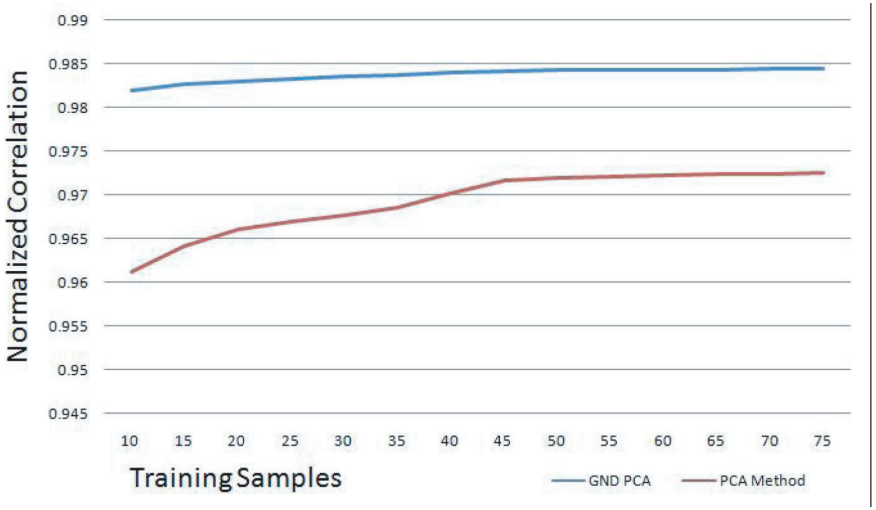


Fig. 7 Comparison of Normalized correlations between GND PCA (Mode-subspace bases: $1200 \times 5 \times 5$) and PCA method (Eigenface method, all the available bases are used in the reconstruction) when the number of training samples is increasing.

(3) Comparison with PCA-Based Method

The PCA-based methods are also tried to build the statistical appearance models for facial ensembles; however, two kinds of PCA-based methods suffers from problems. For the first way to implement PCA, the computing cost is huge. The vectors unfolded from the $3000 \times 13 \times 13$ data have the high dimension of 507000, so that the covariance matrix in the unfolded vector space has an extremely huge dimension of 507000×507000. Assuming float data type is used to store the covariance matrix in the computer, we need to allocate the memory of more than 1 MGB. This is impossible for the current desktop personal computer.

The second way to implement PCA (which is called eigenface Method) suffers from the problem of bad performance on generalization. The leave-one-out experiments are done and **Fig. 9** compares the normalized correlation between the original sample facial ensembles and the reconstructed sample facial ensembles by the eigenface method and GND-PCA method. All the 19 available bases are used in the reconstructions in the eigenface method while the mode-subspace bases trained by GND-PCA is $1200 \times 5 \times 5$. **Figure 8** shows the examples for the reconstructed results in the leave-one-out testing for the eigenface method. It can be seen that the quality of the reconstructed results are not satisfied. The reconstructed facial images are blurred and do not represent the characters of the original images. The problem for this is because the available bases by eigenface method are too few to represent an untrained vector in the unfolded vector space with large dimension.

(4) Comparison with ND-PCA

Since ND-PCA can only compress the data in one mode-subspace, it represents data not as efficient as GND-PCA method under the same compression rate. This can be demonstrated by the following experiments. With the quite similar compression rate, where $\frac{1200 \cdot 5 \cdot 5}{3000 \cdot 13 \cdot 13} \approx 5.92\%$ for GND-PCA and $\frac{230 \cdot 13 \cdot 13}{3000 \cdot 13 \cdot 13} \approx \frac{3000 \cdot 1 \cdot 13}{3000 \cdot 13 \cdot 13} \approx \frac{3000 \cdot 13 \cdot 1}{3000 \cdot 13 \cdot 13} \approx 7.20\%$ for mode-1, mode-2, mode-3 of ND-PCA respectively, we use leave-one-out experiment to compare the quality of reconstruction of GND-PCA with the ND-PCA method by normalized correlation. It can be seen in Fig. 9 that the normalized correlation for ND-PCA is lower than GND-PCA. This illustrates that the qualities of the reconstructed ensembles by ND-PCA are worse than GND-PCA when the compressing rate of two methods are nearly the same. Figure 8 also gives an example for the reconstructed ensembles by the ND-PCA method in the leave-one-out testing. It can be seen that the results of ND-PCA is better than the results of the eigenface method. But they are more blurred compared to the results of GND-PCA method.

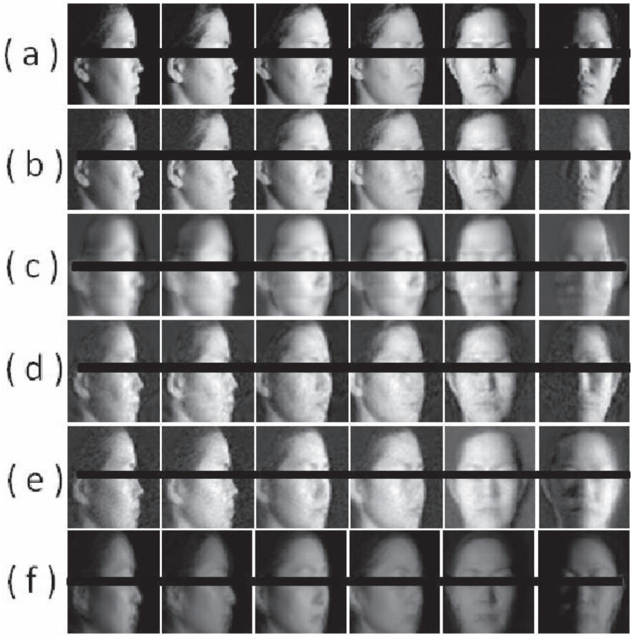


Fig. 8 (a) Some original images from MaVIC. (b) Images reconstructed by GND-PCA (Mode-subspace bases: $1200 \times 5 \times 5$). (c) Images reconstructed by ND-PCA (Mode-1 (Viewpoints) subspace bases: $3000 \times 1 \times 13$). (d) Images reconstructed by ND-PCA (Mode-2 (Pixels) subspace bases: $230 \times 13 \times 13$). (e) Images reconstructed by ND-PCA (Mode-3 (Illuminations) subspace bases: $3000 \times 13 \times 1$). (f) Images reconstructed by PCA (Eigenface method), all the 19 available bases are used in the reconstruction.

As mentioned above, ND-PCA is a unilateral-projection-based scheme, where only one mode multiplication is taken. So that the feature tensor (core tensor) which is got by ND-PCA contains the correlation information among the vectors of mode-matrice after corresponding-mode unfolding. Compared with the

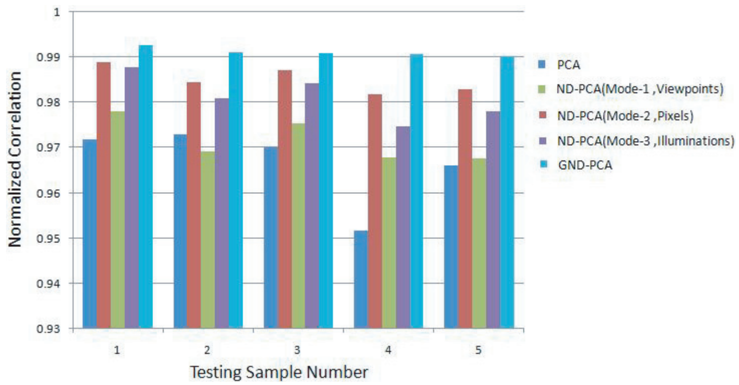


Fig. 9 Comparison of the results in the leave-one-out testing among PCA, ND-PCA and GND-PCA.

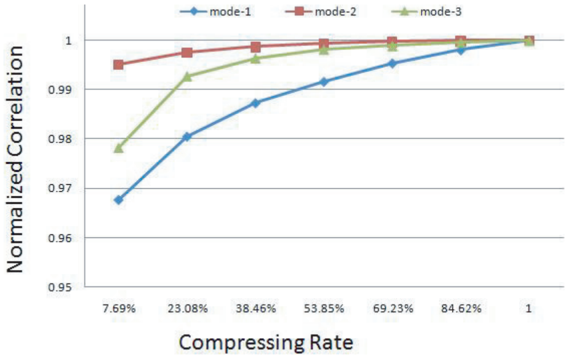


Fig. 10 Comparison of Normalized Correlation among different mode-subspace when different number of mode-subspace bases are used.

ND-PCA, GND-PCA can effectively remove the redundancies among the vectors of each mode-matrices by detecting the features on all mode subspace simultaneously and thus lower down the number of coefficients used to represent a tensor.

Another problem is that it is difficult to find the best approximation of a series of tensors rather than one tensor. We need to give a fixed number to the rank of each mode to do the feature detection. **Figure 10** gives the comparison of normalized correlations with the same compressing ratios in each mode-subspace and shows that the reconstruction results is much more sensitive on mode-1 (viewpoints) than on mode-3 (illuminations) while increasing the compressing rank number of mode-subspace. It is more stable on mode-2 (pixels) than on mode-1 (viewpoints) and on mode-3 (illuminations). It should be noted that on mode-2 (pixels) the variants among the samples can be represented as much as possible by using very small number basis. So we can choose the smaller rank number on mode-2 (pixels) and choose the larger rank number on mode-1 (viewpoints) and on mode-3 (illuminations) by GND-PCA method to keep the higher construction information in the core tensors under the same compression rates. Some researchers have focused on the optimal tensor approximation of dimensionality reduction (such as 13), 14)) and it is still an open problem to deal with.

5. Related Work

Since a tensor can be used to create efficient and intuitive representations for a high-dimension data, it is a hot topic for researchers to introduce the tensor-based method into image analysis area to deal with the image ensembles with multiple factors. The image data tensor is decomposed in order to separate and represent the constituent factors. There are mainly two types of tensor decomposition method: CANDECOMP-PARAFAC (CP) Decomposition[9)] and TUCKER Decomposition[8)]. The two methods can be extend to arbitrary ordered tensors. The first method is used for the sparse linear image coding, such as tensor-rank principle[15)] and NTF (non-negative tensor factorization)[16),17)]. But it is too complicated to compute and understanding. Our method is based on the TUCKER Decomposition which is based on tensor flattening.

The rank-R tensor approximation algorithm[19)] based on slice projection of third-order tensors were proposed for represent the multidimensional data. Similar to GND-PCA algorithm in this paper, this algorithm also needs iterations for convergence. A subtle but important difference between rank-R tensor approximation algorithm and GND-PCA algorithm is that the former mainly aims to organize all the samples together into a larger tensor for dimensionality reduction, which is extremely important for image compression in computer vision, whereas, the GND-PCA algorithm aims to compress a series of image ensembles by dealing with each sample as a separate tensor and making use of correlation information between them. As same as GND-PCA is an extension of G2D-PCA method, Rank-R tensor approximation algorithm reduces to the GLRAM (generalized low rank approximation of matrices)[18] when the tensor is second order.

The work that is most closely related to the current one is the MPCA (multilinear principal component analysis) recently proposed[20]. Like GND-PCA, MPCA works with tensor objects and determines a multilinear projection onto a tensor subspace of lower dimensionality that captures most of the signal variation present in the original tensorial representation. The key different is that the iteration algorithm which is used for calculating the optimal dimension-reduced matrices on each mode-subspace is formulate in GND-PCA definitely.

6. Conclusion

In this paper we proposed GND-PCA for statistical appearance modeling of facial images with multiple modes including different people, different pose and different illumination. Compared with ND-PCA method, it can represent the data more efficiently and the constructed models have good performance on generalization. In the future, we will apply the proposed method to statistical analysis of cosmetic facial images in order to design and control various types of facial appearance using cosmetic foundations[21]. The proposed method can also be used for generating various types of facial images in computer graphics.

References

1) Jolliffe, I.T.: *Principal Component Analysis*, Springer-Verlag, New York (2002).
2) Turk, M. and Pentland, A.: Eigenfaces for Recognition, *Journal of Cognitive Neuroscience*, Vol.3, No.1, pp.71–86 (1991).
3) Yang, J., Zhang, D., Frangi, A.F., et al.: Two-Dimensional PCA: A New Approach to Appearance-based Face Representation and Recognition, *IEEE Trans. PAMI*, Vol.26, No.1, pp.131–137 (2004).
4) Kong, H., Li, X., Wang, L., et al.: Generalized 2D Principal Component Analysis for Face Image Representation and Recognition, *Neural Networks*, Vol.18, pp.585–594 (2005).
5) Yu, H.C. and Bennamoun, M.: 1D-PCA, 2D-PCA to nD-PCA, *18th International Conference on Pattern Recognition* (*ICPR'06*) (2006).
6) Lathauwer, L.D., Moor, B.D. and Vandewalle, J.: A Multilinear Singular Value Decomposition, *SIAM Journal of Matrix Analysis and Application*, Vol.21, No.4, pp.1253–1278 (2001).
7) Lathauwer, L.D., Moor, B.D. and Vandewalle, J.: On the Best Rank-1 and Rank-(R1, R2, ..., RN) Approximation of Higher-order Tensors, *SIAM Journal of Matrix Analysis and Application*, Vol.21, No.4, pp.1324–1342 (2001).
8) Tucker, L.R.: Some Mathematical Notes of Three-mode Factor Analysis, *Psychometrika*, Vol.31, pp.279–322 (1996).
9) Bader, B.W. and Kolda, T.G.: Algorithm 862: Matlab Tensor Classes for Fast Algorithm Prototyping, *ACM Trans. Math. Softw.*, Vol.32, No.4, pp.635–653 (2006).
10) Vasilescu, M.A.O. and Terzopoulos, D.: Multilinear Subspace Analysis of Image Ensemble, *Proc. IEEE EConf. Computer Vision and Pattern Recognition*, Vol.2, pp.93–99 (2003).
11) Xu, R. and Chen, Y.W.: Appearance Models for Medical Volumes with Few Samples by Generalized 3D-PCA, *Lecture Notes in Computer Science*, Springer, LNCS 4984, pp.821–830 (2008).
12) Chen, Y.W., Fukui, T., Qiao, X., et al.: Multi-angle View, Illumination and Cosmetic Facial Image Database (MaVIC) and Quantitive Analysis of Facial Appearance, *Lecture Notes in Computer Science*, Springer, LNCS 5342, pp.411–420 (2008).
13) Wang, H. and Ahuja, N.: A Tensor Approach to Dimensionality Reduction, *IEEE Computer Vision and Pattern Recognition (ICPR'05)* (2005).
14) Wang, H. and Ahuja, N.: Compact Representation of Multidimensional Data using Tensor Rank-one Decomposition, *17th International Conference on Pattern Recognition* (*ICPR'04*), Vol.1, pp.44–47 (2004).
15) Shashua, A.: Linear Image Coding for Regression and Classification using the Tensor-rank Principle, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii (2001).
16) Lee, D. and Seung, H.: Learning the Parts of Objects by Non-negative Matrix Factorization, *Nature*, Vol.401, pp.788–791 (1999).
17) Hazan, T., Polak, S. and Shashua, A.: Sparse Image Coding using a 3D Non-negative Tensor Factorization, *International Conference on Computer Vision* (*ICCV*), Beijing (2005).
18) Ye, J.: Generalized Low Rank Approximations of Matrices, *In Proceedings of the Twenty-First international Conference on Machine Learning* (*ICML '04*), New York, NY (2004).
19) Wang, H. and Ahuja, N.: Rank-R Approximation of Tensors using Image-as-matrix Representation, *IEEE Computer Society Conference*, Vol.2, pp.346–353 (2005).
20) Lu, H., Plataniotis, K.N. and Venetsanopoulos, A.N.: MPCA: Multilinear Principal Component Analysis of Tensor Objects Neural Networks, *IEEE Transactions*, Vol.19, Issue 1, pp.18–39 (2008).
21) Igarashi, T.: Recent technical trends on cosmetic foundations, *Fragrance Journal*,

Vol.34, pp.17–28 (2006).

Appendix

A.1 The Proof of Theorem 3.2

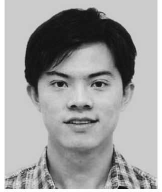
Proof. The cost function can be expanded, shown as $S = \sum_{i=1}^{M} \| \mathcal{A}_i - \hat{\mathcal{A}}_i \|^2 = \sum_{i=1}^{M}(\| \mathcal{A}_i \|^2 - 2\langle \mathcal{A}_i, \hat{\mathcal{A}}_i\rangle + \| \hat{\mathcal{A}}_i \|^2)$. According to the definition of inner product of tensor, we can write the inner product $\langle \mathcal{A}_i, \hat{\mathcal{A}}_i\rangle$ as $\langle \mathcal{A}_i, \hat{\mathcal{A}}_i\rangle = \langle \mathcal{A}_i, \mathcal{B}_i \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times \ldots \times_N \mathbf{U}^{(N)}\rangle = \langle \mathcal{A}_i \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} \times \ldots \times_N \mathbf{U}^{(N)^T}, \mathcal{B}_i\rangle = \langle \mathcal{B}_i, \mathcal{B}_i\rangle = \| \mathcal{B}_i \|^2$. Since the matrices $\mathbf{U}^{(n)}$ have orthogonal columns, they will not affect the Frobenius norm[7]. Therefore, we get $\| \hat{\mathcal{A}}_i \| = \| \mathcal{B}_i \|$. Substituting $\langle \mathcal{A}_i, \hat{\mathcal{A}}_i\rangle = \| \mathcal{B}_i \|^2$ and $\| \hat{\mathcal{A}}_i \| = \| \mathcal{B}_i \|$ into the cost function, we can get $S = \sum_{i=1}^{M}(\| \mathcal{A}_i \|^2 - \| \mathcal{B}_i \|^2) = \sum_{i=1}^{M}(\| \mathcal{A}_i \|^2 - \| \mathcal{A}_i \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} \times \ldots \times_N \mathbf{U}^{(N)^T} \|^2) = \sum_{i=1}^{M} \| \mathcal{A}_i \|^2 - S'$. The first term, $\sum_{i=1}^{M} \| \mathcal{A}_i \|^2$, has a fixed value, so minimization of cost function is equal to maximization of the last term S'. The theorem is proved. □ □

A.2 The Proof of Lemma 3.3

Proof. We can first represent S' by $S' = \sum_{i=1}^{M} \| \mathcal{A}_i \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} \times \ldots \times_N \mathbf{U}^{(N)^T} \|^2 = \sum_{i=1}^{M} \| (\mathcal{A}_i \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} \times \ldots \times_{n-1} \mathbf{U}^{(n-1)^T} \times_{n+1} \mathbf{U}^{(n+1)^T} \times \ldots \times_N \mathbf{U}^{(N)^T}) \times_n \mathbf{U}^{(n)^T} \|^2 = \sum_{i=1}^{M} \| \mathcal{C}_i \times_n \mathbf{U}^{(n)^T} \|^2 = \sum_{i=1}^{M} \| \mathbf{U}^{(n)^T} \cdot \mathbf{C}_{i(n)} \|^2 = \sum_{i=1}^{M} tr(\mathbf{U}^{(n)^T} \cdot \mathbf{C}_{i(n)} \cdot \mathbf{C}_{i(n)}^T \cdot \mathbf{U}^{(n)}) = tr(\mathbf{U}^{(n)^T} \cdot \sum_{i=1}^{M}(\mathbf{C}_{i(n)} \cdot \mathbf{C}_{i(n)}^T) \cdot \mathbf{U}^{(n)})$. Therefore, maximization of the cost function S' is the same as maximization of $tr(\mathbf{U}^{(n)^T} \cdot \sum_{i=1}^{M}(\mathbf{C}_{i(n)} \cdot \mathbf{C}_{i(n)}^T) \cdot \mathbf{U}^{(n)})$. This is a well-resolved problem. The solution is to select columns of the matrix $\mathbf{U}^{(n)}$ as the first J_n leading eigenvectors of the matrix $\sum_{i=1}^{M}(\mathbf{C}_{i(n)} \cdot \mathbf{C}_{i(n)}^T)$. □ □

(Received October 8, 2008)
(Accepted April 1, 2009)
(Released September 24, 2009)

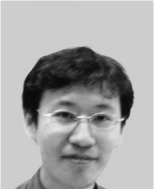
(Communicated by *Shuji Senda*)



Xu Qiao was born in 1983. He received the B.S. and M.S. degrees from School of Mathematics, Shandong University in 2004 and 2007, respectively. He is currently a graduate student in the doctoral program of Graduate School of Science and Engineering, Ritsumeikan University. His research interests include image processing and pattern recognition.

Rui Xu received the B.S. and M.S. degrees in 2001 and 2004, respectively, from School of Electronic and Information, South China University of Technology. He received the Ph.D. degree in 2007 from the Graduate School of Science and Engineering, Ritsumeikan University, Japan. Right now, he is a research staff in Digital Technology Research Center, Sanyo Electric Co., Ltd. His research fields are image processing and pattern recognition.

Yen-wei Chen received a B.E. degree in 1985 from Kobe University, Kobe, Japan, a M.E. degree in 1987, and a D.E. degree in 1990, both from Osaka University, Osaka, Japan. From 1991 to 1994, he was a research fellow with the Institute of Laser Technology, Osaka. From October 1994 to March 2004, he was an associate Professor and a professor with the Department of Electrical and Electronic Engineering, University of the Ryukyus, Okinawa, Japan. He is currently a professor with the college of Information Science and Engineering, Ritsumeikan University, Kyoto, Japan. He was a visiting professor with the Oxford University, Oxford, UK in 2003. He is an Overseas Assessor of Chinese Academy of Science and Technology. He is an associate Editor of International Journal of Image and Graphics (IJIG) and editorial board members of the International Journal of Knowledge-based and Intelligent Engineering Systems and the International Journal of Information. His research interests include pattern recognition, image processing and machine learning. He has published more than 200 research papers in these fields.



Takanori Igarashi received the B.S. and M.S. degrees in applied chemistry from Waseda University in 1996 and 1998, respectively. From 1998, he has been a research scientist at Kao Co., Ltd. During 2003–2004, he was a visiting research scientist at Department of Computer Science, Columbia University. Since joining Kao Corporation, his research interests have been analysis on skin appearance using optics and image processing, and development of cosmetic ingredients using inorganic chemistry.

Keisuke Nakao received the B.S. and M.S. degrees in 1990 and 1992, respectively, from Department of Applied Chemistry, School of Science and Engineering, Waseda University. Right now, he is a senior researcher in Beauty Cosmetic Research Lab., Kao Co., Ltd. His research fields are development of functional powder and research of powder cosmetic products.

Akio Kashimoto received the B.S. and M.S. degrees in 1988 and 1990, respectively, from faculty of science, Tokyo University of Science. Right now, he is a research staff in Beauty Research Center, Kao Co., Ltd. His research fields are consumer trends and value investigation of beauty cosmetics.