Fibre Centred Tensor Faces

Abstract

In this paper we present a reformulation of the tensorface analysis method and produce a model that is simpler (i.e. has fewer parameters), is more compact (i.e. has tighter distributions) and is less ambiguous (i.e. no 2 sets of parameters synthesise the same data vector). This is achieved by simply subtracting the fibre (row, column, etc) mean from each fibre of the training data before performing PCA analysis. Centring of tensor data via subtraction of the whole set mean is commonly used as a preprocessing step, but the fibre-centring algorithm presented here has not been suggested previously for tensorface analysis. We show how the new formulation allows an approximate linear analysis with a considerable speed improvement over previous methods. In addition, the centring allows simpler truncation of parameter vectors leading to an even more compact model. The new method is tested on image synthesis and analysis and in a simple face recognition task, in which it out performs non-centred multilinear analysis.

1 Introduction

Multilinear analysis has proved a useful tool in facial image analysis because of its ability to separate different sources of variation such as pose, expression, identity and lighting. While centring via subtraction of the whole set mean is frequently used as a preprocessing step, the choice of centring algorithm can lead to different results and more or less efficient algorithms. In this work we present a centring approach designed for face analysis. The centring approach used is to centre each “fibre” before unfolding the tensor, removing variation due to factors that are more easily accounted for using other axes of the face tensor. For example, using an identity-expression tensor, we would subtract each subject's average (across expressions) from the individual expression images and perform the expression part of the analysis on the resulting “identity free” tensor. This is repeated along the identity axis, removing the average of each expression from each individual expressive image, leaving “expression free” identity data. In this paper we first review the related literature, then explain the proposed method in detail and finally present preliminary results that demonstrate that this kind of processing not only leads to more appropriate parameters (for improved face recognition) it also leads to a faster analysis algorithm.

2 Literature Review

Multilinear analysis was introduced to the computer vision community by Vasilescou and Terzopoulos [1-4]. Multilinear data represent the natural extension from scalars (0-D tensors), through vectors (1D tensor) and matrices (2D tensors) to general n-dimensional data. For face data the images are usually scan converted into a 1-D vector (vectorised), with the data vector representing 1 dimension of the face tensor and other attributes varying along the other axes e.g. along one axis identity, along another expressions and along another could be different lighting. To synthesise a new data vector from this block it seems intuitively reasonable to use one set of weights along each axis i.e. one for the identity axis, another along the expression axis and a third set of weights along the lighting axis. Hence the linear weightings of the example data vectors are given as the outer product tensor of the weights along each axis of the data block. This leads to a considerable reduction in the number of parameters required over a linear model. For example, using 10 identities, 10 expressions and 10 lightings would only require 30 parameters (10+10+10) instead of 1000 (10x10x10) using a linear combination of the training data vectors. The reconstruction from such a data set is therefore given by:

$$t=\sum_{i=0}^{I}\sum_{j=0}^{J}\cdots\sum_{k=0}^{K} a_i b_j \cdots c_k t_{ij\cdots k} \qquad (1)$$

where *t* is the reconstructed data vector, a_i, b_j, ..., c_k are the model parameters and $t_{ij...k}$ are the model's components.

As with linear face analysis, it has proved useful to perform additional (multi)linear processing on the input data. The standard decomposition is to factor the n-way tensor into N orthogonal matrices plus a core tensor. The usual method of construction is to unfold the tensor into a matrix along each axis, perform Singular Value Decomposition (SVD) on the unfolded matrix and then take the left orthogonal matrix. This is known as high-order SVD (HOSVD) or Tucker's method [5]. The HOSVD method does not yield unique output parameters and the core tensor is not diagonal. Alternatives have been proposed, for example Parallel Factor Analysis (PARAFAC), which is also known as Canonical Decomposition (CANDECOMP) [6][7], in which the data is modelled as the sum of N outer product tensors. This has the attractive properties of a diagonal core tensor and that the decomposition is unique, but the generating matrices are not orthogonal. The development of PARAFAC/CANDECOMP was motivated by a need to discover fundamental properties of the original data. For example in chemometrics it has been used to discover the pure spectra of mixed samples. Although one can think of applications in face analysis in which the primary aim is to discover fundamental properties of the original data (e.g. to find basic components of expressions) the aim in most of the existing face research is for compact, efficient and effective analysis and synthesis methods, and so the use of more efficient orthogonal decomposition remains attractive.

In order to analyse an example data vector several methods have been proposed. In their original tensorface work [1] Vasilescou and Terzopoulos proposed finding the optimal set of identity vectors for each set of non-identity parameters (expression, lighting, viewpoint). The optimal set of these weights that best matches the target are considered the best match. This needs to be repeated across the database to find the overall best match. This method has two drawbacks, the first is that it is close to an exhaustive search and therefore slow and the second is that it does not allow for interpolated values of the non-identity parameters e.g. lighting between two of the training lighting examples.

The ideal would be to find the parameters that minimise (1) in the least squares sense i.e.

$$\chi^2=\sum_{x=0}^{X}\left(\sum_{i=0}^{I}\sum_{j=0}^{J}\cdots\sum_{k=0}^{K} a_i b_j \cdots c_k T_{ij\cdots k}(x)-T(x)\right)^2 \qquad (2)$$

This is a highly non-linear equation and difficult to solve explicitly. In [3] Vasilescou et al approximated the solution using a two step minimisation i.e. first they find the best linear parameters, $r_{ij...k}$ that minimise:

$$\chi^2 = \sum_{x=0}^{X} \left(\sum_{i=0}^{I} \sum_{j=0}^{J} \cdots \sum_{k=0}^{K} r_{ij\cdots k} T_{ij\cdots k}(x) - T(x) \right)^2 \qquad (3)$$

and then they find the best rank 1 approximation to the Tensor $r_{ij...k}$ using HOSVD. This is not guaranteed to find the best solution to (3), as other authors have noted [8], but seems a reasonable approximation.

Others have solved (2) using iterative methods. Park and Savvides [9] use the additional restriction that the magnitude of each of the weight vectors should be 1 (i.e. $\sum_i a_i^2 = \sum_j b_j^2 = \cdots = \sum_k c_k^2 = 1$). The resulting equations are solved iteratively using a Lagrange multiplier to include the additional constraint. This additional constraint is required because the original tensorface model is non unique, i.e. two (or more) sets of parameters can give the same output e.g. scaling the a_i by α and the b_j by $1/\alpha$ produces the same result. It is this non-uniqueness of the parametric description that we believe leads to under performance of the original tensor-face SVD model. Because the identity parameters are only known subject to some linear multiple, recognition is based on maximising the normalised scalar product between the known and test identity parameters. It is quite possible that two very different face images could have the same normalised identity parameters, depending on the values of the other parameters. Conversely, it is possible for the same face to have quite different parameters that lie along a line in the identity parameter space.

One difference between the tensorface analysis proposed by Vasilescou et al and the frequently used PCA analysis of facial images e.g. [12][13], is that in the latter the mean is explicitly subtracted from the set before performing the PCA analysis (e.g. using SVD or Jacobi rotations to find the Eigenvectors and Eigenvalues of the covariance matrix). This makes sense from the perspective of both the dimensionality of the linear space spanned by the data and from a Gaussian model of the distribution of the data. Given N non-collinear points in some (generally higher) M dimensional space, N-1 direction vectors are required to span the space, plus a single position vector within the linear space. Hence a space defined by N points only requires at most N-1 parameters to span the linear subspace. A convenient point to choose as the position vector within the space is the mean, as this minimises the size of the distribution.

Bro and Smilde [10] have considered the effects of centring and also scaling of the data in multilinear analysis, but they do not consider the situation in which the fibre means are retained in the model along every dimension. Lu et al [11] have used a centred multilinear approach for gait analysis, but again the centring is only performed once along a single "samples" axis. It is not clear how their method could be extended to include appropriate centring along multiple sample axes, such as different styles of gait.

In this paper we adapt the HOSVD method with the addition of fibre centring. First we outline the mathematics of the reformulated representation and then describe methods for the analysis of an input vector and synthesis of an output vector. Finally we test the new method on a face recognition task and compare the performance with a non-centred multilinear method.

3 Method

3.1 Tensor construction

We propose a simple modification to the multilinear analysis of face images in which the mean is explicitly subtracted before processing. The mean we use in this context is the "fibre" mean (we will use fibre to refer to any 1D array extracted parallel to the Cartesian axes e.g. rows, columns, etc.) Removing the fibre mean leaves difference vectors that minimise the variation within the fibre. For example, using an identity-expression-lighting set we would remove the identity and expression variation from the lighting fibres, the identity and lighting variation from the expression fibres and the expression and lighting variation from the identity fibres. The hope is to minimise the variation due to known factors when processing along each axis.

For each axis of the tensor we first unfold the tensor along that axis and then multiply by the matrix [***I-M***], where each element of ***M*** is equal to *1/N.* This subtracts the fibre average from each sample. Next we perform PCA analysis on the resulting matrix to find the rotation matrix $\boldsymbol{U}_i$ and variances w_i. At most *N-1* of the variances will be non-zero, so we can discard at least the smallest component. We construct the matrix $\boldsymbol{U}_i'$ whose first row is an averaging vector and the remaining rows calculate the rotated difference components as follows:

$$U_i' = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & u_{00} & \cdots & u_{N0} \\ \vdots & & \ddots & \\ 0 & u_{0\mathrm{N}-1} & \cdots & u_{NN-1} \end{bmatrix} \begin{bmatrix} 1/N & 1/N & \cdots & 1/N \\ 1-1/N & -1/N & \cdots & -1/N \\ \vdots & & \ddots & \vdots \\ -1/N & -1/N & \cdots & 1-1/N \end{bmatrix} \qquad (4)$$

where $\{u_{jk}\} = \boldsymbol{U}_{\mathrm{i}}$. We perform the analysis on the original data along each axis, and then as a final step multiply the original tensor by the matrices $\boldsymbol{U}_i'$ along each direction. In our implementation no processing is performed along the direction of the data vector. An alternative option is to perform the decomposition along all axes and then to undo the transform along the data vector using the inverse operator. In our implementation the inverse operator $\boldsymbol{U}_i'^{-1}$ is given by:

$$U_i'^{-1} = \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & u_{00} & \cdots & u_{0\mathrm{N}-1} \\ \vdots & & \ddots & \\ 0 & u_{N0} & \cdots & u_{N-1\mathrm{N}} \end{bmatrix} \qquad (5)$$

3.2 Image synthesis

In order to reconstruct a single data vector (e.g. an image) we use an equation of the same form as used previously, i.e.:

$$\boldsymbol{t}=\sum_{i=0}^{I}\sum_{j=0}^{J}\cdots\sum_{k=0}^{K}a_i b_j\cdots c_k \boldsymbol{t}_{ij\cdots k} \qquad (6)$$

except that the first component of each parameter vector is 1, i.e. $a_0 = b_0 = ... = c_0 = 1$. The component $\boldsymbol{t}_{00...0}$ is the overall mean of the entire multilinear set and is given a constant weighting of 1. Hence the model has one less free parameter along each axis, and so is a simpler model. In addition because of the normalisation the distribution of parameter values are tighter and so truncation of the components can be justified. Truncation of the data has previously been performed using an iterative approach that tries to minimise the reconstruction error while preserving orthonormality of the decomposition vectors [4]. The centring scheme proposed here ensures that the truncated tensor model is optimal in the sense that the component with the smallest distribution from the row mean is discarded.

3.3 Image analysis

In this paper we propose 2 methods for estimating the parameters of our updated model given in equation (2). In the first we follow [3] and approximate the solution by first finding the linear estimates that solve equation (3). Because the weight of the overall mean is 1 we subtract the component $\boldsymbol{t}_{00..0}$ before minimising. The least squares solution of (3) for our model is given by:

$$\underline{\boldsymbol{r}}=(\boldsymbol{T}^{\boldsymbol{t}}\boldsymbol{T})^{-1}\boldsymbol{T}^{\boldsymbol{t}}(t-t_{00\cdots0}) \qquad (7)$$

where $\underline{\boldsymbol{r}}$ is the vector containing the vectorised version of the tensor $R = \{r_{ij...k}\}$ and $\boldsymbol{T}$ is the matrix whose columns are the data vectors of the tensor T excluding the mean (i.e. the tensor T unfolded along the data axis, less the mean).

In previous work HOSVD has been used to factor R into the best rank-1 approximation and find the outer product vectors. For the new model we note that the weights on the origin adjacent edges of the linear weight tensor should be equal to the parameter vectors we seek i.e.

$$\begin{aligned} r_{i0\cdots0}&=a_i\\ r_{0\mathrm{j}\cdots0}&=b_j\\ &\vdots\\ r_{00\cdots k}&=c_k \end{aligned} \qquad (8)$$

As an additional optimisation we do not need to calculate all of the linear components explicitly, as these can be "projected out". We only need the rows of the matrix $\boldsymbol{P} = (\boldsymbol{TT})^{-1}\boldsymbol{T}^{t}$ that correspond to the origin adjacent edge components of $\underline{\boldsymbol{r}}$ in order to get a fast (linear) estimate of the model parameters. Once the matrix $\boldsymbol{P}$ has been calculated only $O(I+J+ ... +K)$ dot products are required to analyse an input image rather than $O(IJ...K)$ dot products required to compute the entire tensor R. Also this method does not require the additional work of using HOSVD to find the best rank 1 approximation of the tensor *R* and factor it into vectors.

If additional accuracy is found necessary we propose using an alternating least squares (ALS) approach. An initial estimate of the parameters can be found using equation (7) and then the parameters along each axis are re-estimated, keeping the others fixed i.e. estimate parameters a_i keeping b_j ... c_k fixed, then estimating parameters b_j, keeping a_i ... c_k fixed etc. This involves collapsing the model using the fixed components e.g. if we are optimising the parameters a_i the components are given by:

$$\boldsymbol{t}'_i = \sum_{j=0}^{J} \cdots \sum_{k=0}^{K} b_j \cdots c_k \boldsymbol{t}_{ij \cdots k} \qquad (9)$$

and the model is given by:

$$\boldsymbol{t} = \boldsymbol{t}'_0 + \sum_{i=1}^{I} a_i \boldsymbol{t}'_i \qquad (10)$$

which is solved for a_i. This process can be repeated until convergence (i.e. the parameters remain stable).

3.4 Face recognition method

In this section we describe how to find the best match from a set of example faces to a previously unseen facial image. In order to compute the best match, previous authors have used the maximum normalised scalar product between the known identity components and those estimated for the test image, i.e.

$$\hat{\boldsymbol{a}} \cdot \hat{\boldsymbol{b}} = \frac{\sum a_i b_i}{\sqrt{\sum a_i^2 \sum b_i^2}} \qquad (11)$$

is maximised, where ***a*** and ***b*** are the identity parameter vectors for two subjects. The normalised scalar product was used because of the unknown scaling of the parameters. In contrast the values of the fibre-centred tensor parameters are known precisely, so we have explored alternative recognition methods. One option is to build a Gaussian model of the identity parameters for each subject and calculate the maximum probability over all subjects i.e. maximise:

$$p(\boldsymbol{a}) = \frac{1}{(2\pi)^{n/2} |\boldsymbol{C}_b|} e^{-\frac{1}{2}(\boldsymbol{a}-\bar{\boldsymbol{b}})' \boldsymbol{C}_b^{-1} (\boldsymbol{a}-\bar{\boldsymbol{b}})} \qquad (12)$$

where ***a*** is the identity parameter vector of the test image and ***b*** and $\boldsymbol{C}_b$ are the mean and covariance matrix that define the Gaussian distribution of identity parameters for the training face that we are testing against. The problem with this method is that the distribution of identity parameters for an individual in the training data is usually zero or very close to zero. This could be seen as a case of over fitting of the model to the training data, and the same is true of the non-centred HOSVD, save for the unknown sign of the normalised components. We could populate the Gaussian model by using additional example images of the same subjects used to train the tensor model but as this requires an additional image set, we instead use the simpler method of finding the closest mean to the test subject i.e. we seek to minimise:

$$\sum_{i=0}^{I} (a_i - \bar{b}_i)^2 \qquad (13)$$

where ***a*** is the test identity vector and ***b*** is the mean identity vector for the training image we are comparing against.

4 Results

In this section we present the results of our reformulated tensorface methods and compare them with the original algorithms. First we describe our data preprocessing, then we demonstrate the results of analysis and synthesis of face images. Finally we use the method for a face recognition task and compare the results with standard HOSVD recognition.

The test set of images need to contain standardised images that vary along a number of dimensions (such as identity, pose, lighting). Many of the existing test sets, such as the FERET set are not suitable for multilinear training, as the poses etc are not standardised across subjects. Other more suitable images sets exists, such as the Yale database and the Weizman image set. We use two example sets for training and testing, one set of our own images and one from a standard collection. The first is a bi-linear set in which 7 subjects are performing 16 different viseme (visual phonemes). The peak viseme images were extracted from video by hand. The images in this set are face-on images with consistent lighting. We use an out-of-set (neutral) image for testing. The second set is a tri-linear subset of the Weizmann face database. The images in the subset are face-on images and show 15 identities, 3 lightings and 3 expressions. A forth image of each subject with a neutral expression and different lighting is used as the test set.

4.1 Data pre-processing

In this work we decompose the facial image data into a shape component and a shape normalised image component. Each image was delineated automatically using an active appearance model (AAM) [11][12] and any errors were corrected by hand. The average of the feature points across the set was calculated and was shifted and scaled to fit in a window of a user specified size (for the experiments shown here we used a 100 by 150 pixel image). The delineated feature template of each subject was then normalised to this average using best fit scaling, rotation and translation before being vectorised and placed in the input shape tensor. The input images were warped into the normalised average shape using piecewise linear warping over a triangulation of the template feature points. All but the face region was masked and then the resulting images were vectorised and added to the input image tensor.

4.2 Model Building Example

Figure 1 shows the decomposition process applied to the subset of the Weizmann face set. Only 3 of the 15 identities used are shown. The input data tensor comprising the shape normalised example images is on the left. The result of processing along each axis in turn with the matrix $\boldsymbol{U}_i'$ is shown going from left to right. The original data and averaged components are unmodified for display and the processed difference components are scaled and shifted so that mid-grey equals zero for display.

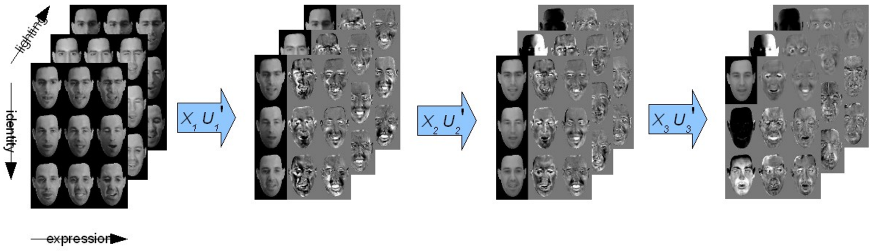

Figure 1: Example of the decomposition process on a set of (shape normalised) face images.

4.3 Analysis and Synthesis Examples

Figure 2 shows the synthesis process. The synthesis process forms a weighted sum of the centred multilinear data (left). The weights (centre) are formed from the outer product tensor of the parameter vectors. Each parameter vector starts with the value 1 and so the contribution of the overall mean is always 1. Figure 3 illustrates the linear analysis process. Figure 4 shows some examples of face synthesis from parameters found using the linear approximation. Examples are given with and without truncation of the model.

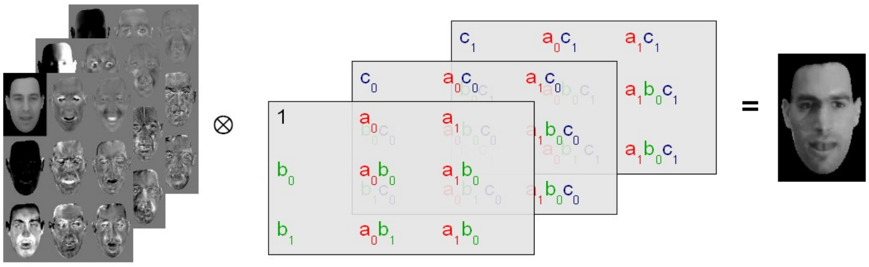

Figure 2: A diagram of the face synthesis method for the normalised image components.

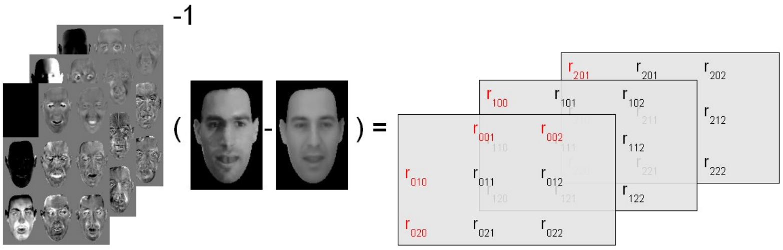
-1
(-) =
r_{201} r_{201} r_{202}
r_{212}
r_{222}
r_{100} r_{101} r_{102}
r_{112}
r_{122}
r_{001} r_{002}
r_{010} r_{011} r_{012}
r_{020} r_{021} r_{022}

Figure 3: Diagram showing the (approximate) linear analysis process. Only the components of the pseudo inverse matrix (left) corresponding to the linear coefficients shown in red (right) need to be used.

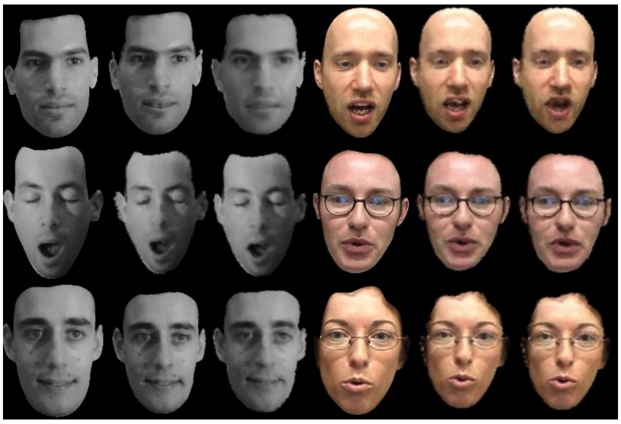

Figure 4: Face synthesis by analysis examples from the two data sets used. In each set: left original, centre 100% and right 95% of variance in each parameter explained in the two data sets.

4.4 Face Recognition Experiments

We tested the centred tensor face model's performance on a simple face recognition task using our two face sets for training. We tested the model's ability to correctly assign an identity to previously unseen images of the same individuals as used in the training set. For the viseme set this was a neutral expression image and for the Weizmann set it was a neutral image with different lighting. For the centred model we used the approximate linear analysis and truncated the model parameters to explain 95% of the variance in the training data along each axis. For the viseme set this reduced the viseme parameters from 15 to 12 in shape and from 15 to 13 in colour but did not truncate the identity parameters. For the Weizmann subset the identity parameters were reduced from 15 to 12 in both shape and colour and the lighting and expression parameters were not truncated. The results are shown in Table 1.

	Non-centred	Centred
Viseme	71%	100%
Weizmann	67%	100%

Table 1. Results of the simple face recognition experiments.

5 Conclusions and Future Work

In this paper we have developed a reformulation of the multilinear PCA analysis of face images and have demonstrated improved recognition performance over previous HOSVD face recognition in a simple test. The improved performance is almost certainly due to the unknown scaling of the identity parameters in the original HOSVD method requiring the use of the normalised scalar product or normalised nearest neighbour for recognition. This normalisation can project different subjects into a similar part of the parameter space. In contrast in the centred method the parameters are unique and so can be used directly.

Future work will focus on testing the method on larger datasets and for other recognition tasks e.g. expression recognition. We would also like to compare our new method with multilinear independent component analysis (MICA) and explore an ICA version of the algorithm presented here.

References

[1] M.A.O. Vasilescu and D. Terzopoulos. *Multilinear analysis for facial image recognition*. In Proc. Int. Conf. on Pattern Recognition, volume 2, pages 511–514, Quebec City, August 2002.

[2] M.A.O. Vasilescu and D. Terzopoulos. *Multilinear analysis of image ensembles: Tensorfaces*. In Proc. European Conf. on Computer Vision (ECCV 2002), pages 447–460, Copenhagen, Denmark, May 2002.

[3] M.A.O. Vasilescu and D. Terzopoulos, *Multilinear Independent Components Analysis*, Proc. Computer Vision and Pattern Recognition Conf. (CVPR '05),

[4] M.A.O. Vasilescu, D. Terzopoulos, *Multilinear Subspace Analysis for Image Ensembles*,' Proc. Computer Vision and Pattern Recognition Conf. (CVPR '03), Vol.2, Madison, WI, June, 2003, 93-99.

[5] P.M. Kroonenburg, *Three-mode principal component analysis. Theory and applications*. DSWO Press, Leiden, 1983.

[6] R.A. Harshman, *Foundations of the PARAFAC procedure: Model and conditions for an 'explanatory' multi-mode factor analysis.* UCLA Working Papers in phonetics, 16 (1970) 1.

[7] J.D. Carroll and J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*. Psychometrika, 35 (1970) 283.

[8] I. Macedo, E. V. Brazil, and L. Velho. *Expression Transfer between Photographs through Multilinear AAMs*. In Proceedings of SIBGRAPI 2006 - XIX Brazilian Symposium on Computer Graphics and Image Processing, Manaus, October 2006

[9] S.W. Park, M. Savvides, *Estimating Mixing Factors Simultaneously in Multilinear Tensor Decomposition for Robust Face Recognition and Synthesis*, cvprw, p. 49, 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), 2006.

[10] Rasmus Bro and Age K. Smilde, *Centering and scaling in component analysis*, Journal of Chemometrics, 2003, 17, 16-33.

[11] Haiping Lu , K.N. Plataniotis and A.N. Venetsanopoulos, *Multilinear Principal Component Analysis of Tensor Objects for Recognition*, 18th International Conference on Pattern Recognition (ICPR'06), pp. 776-779.

[12] T. F. Cootes, G. J. Edwards, and C. J. Taylor. *Active appearance models*. In 5th European Conference on Computer Vision (ECCV 1998), pages 484–498, June 1998.

[13] I. Matthews and S. Baker. *Active appearance models revisited.* International Journal of Computer Vision, 60(2):135–164, Nov. 2004.