Automatic Face Annotation by Multilinear AAM with Missing Values

Zhen-Hua Feng^{1,2}, Josef Kittler^{2}, William Christmas^{2}, Xiao-Jun Wu^{1}, and Sebastian Pfeiffer^{3}

^{1}*School of IoT Engineering, Jiangnan University, China*
^{2}*Centre for Vision Speech and Signal Processing, University of Surrey, UK*
^{3} *Buchmann Institute for Molecular Life Sciences, Goethe University, Frankfurt/Main, Germany*
{Z.Feng, J.Kittler, W.Christmas}@surrey.ac.uk, xiaojun_wu_jnu@163.com,
Sebastian.Pfeiffer@physikalischebiologie.de

Abstract

It has been shown that multilinear subspace analysis is a powerful tool to overcome difficulties posed by viewpoint, illumination and expression variations in Active Appearance Model(AAM). However, the Higher Order Singular Value Decomposition (HOSVD) in multilinear analysis requires training samples to build the training tensor, which include face images under all different variations. It is hard to obtain such a complete training tensor in practical applications. In this paper, we propose a multilinear AAM which can be generated from an incomplete training tensor using Multilinear Subspace Analysis with Missing Values (M^{2}SA). Also, the 2D appearance is used for training appearance tensor directly to reduce the memory requirements. Experimental results on the Multi-PIE face database show the efficiency of the proposed method.

1 Introduction

The Active Appearance Model (AAM) [2] has been proved to be an efficient generative model in face image analysis. Typically, an AAM is fitted to input images to achieve face annotation or to attempt face tracking, recognition and synthesis. However, the AAM is very sensitive to pose, expression and illumination variations, which seriously limits its applicability.

To counteract the harmful effects introduced by these variations and enhance the fitting performance of AAM, many approaches have been proposed in recent years. Multilinear(tensor) analysis methods, such as tensorfaces [9] [8], have been put forward for their ability to decouple face images into different subspaces. Tensor-based AAM (TAAM) [6] is a typical extension of the tensor method to AAM. TAAM estimates the pose, illumination and expression of the input image and then uses Higher Order Singular Value Decomposition (HOSVD) [3] to decompose the shape and global appearance tensors. Thus, the variation-specific shape and appearance basis tensors can be obtained and converted to basis matrices to build shape and appearance models for the AAM. However, it is normally hard to obtain enough training samples for the HOSVD used by TAAM in practical applications. Furthermore, TAAM treats the appearance images as vectors in HOSVD, which is costly in memory.

In this paper, we introduce Multilinear Subspace Analysis with Missing values (M^{2}SA) [4] to generate the shape and appearance models for AAM. 2D appearances are used to build the appearance tensor directly rather than converting them to vectors. This new structure can significantly reduce the memory costs in tensor decomposition. Compared with [4], in the fitting phase, a discrete variation estimation controls the selection of the appropriate configuration of the TAAM model.The fitting efficiency is achieved by the inverse compositional algorithm [7].

The rest of this paper is organized as follows. Section 2 introduces the basic theory of tensor algebra and the M^{2}SA algorithm for tensor decomposition with missing values. Section 3 discusses how to build the multilinear AAM and details the AAM fitting algorithm. Experimental results obtained on the Multi-PIE face database [5] are presented in Section 4. Finally, the conclusions are drawn in Section 5.

2 Multilinear subspace analysis by M^{2}SA

In this paper, scalars, vectors, matrices and higher-order tensors are denoted by lower-case letters (a,b,$\cdots$), bold lower-case letters ($\mathbf{a}$,$\mathbf{b}$,$\cdots$), bold upper-case latter ($\mathbf{A}$,$\mathbf{B}$,$\cdots$) and calligraphic upper-case letters ($\mathcal{A}$,$\mathcal{B}$,$\cdots$) respectively.

2.1 HOSVD and Dimensionality Reduction

Given an N^{th}-order tensor $\mathcal{D} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, it can be decomposed by HOSVD as:

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_N \mathbf{U}_N, \tag{1}$$

where $\mathcal{Z} \in R^{I_1 \times I_2 \cdots I_N}$ is the core tensor, which stands for the interaction between mode matrices $\mathbf{U}_1, \mathbf{U}_2, \cdots, \mathbf{U}_N$, and $'\times_n'$ is mode-n multiplication. The mode-n matrix $\mathbf{U}_n$ is the left singular matrix obtained by applying Singular Value Decomposition (SVD) to mode-n unfolding matrix $\mathbf{D}_{(n)}$ of tensor $\mathcal{D}$. The core tensor $\mathcal{Z}$ is computed by:

$$\mathcal{Z} = \mathcal{D} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \cdots \times_N \mathbf{U}_N^T. \tag{2}$$

The dimensionality reduction of tensor aims to find a lower $rank-(R_1, \cdots, R_N)$ approximation for an input tensor. The mode-n rank of tensor $\mathcal{D}$ is defined as $R_n = rank(\mathbf{D}_{(n)})$, where $\mathbf{D}_{(n)}$ is the mode-n unfolding matrix of $\mathcal{D}$. A pseudo code is given in Algorithm 1.

Algorithm 1 N-Mode Dimensionality Reduction

```
1. Set the lower rank $R_n < I_n$ for $n = 1, 2, \cdots, N$; apply HOSVD to $\mathcal{D}$; truncate each mode matrix $\mathbf{U}_n$ to $R_n$ columns and obtain the initial mode matrices $\mathbf{U}_1^0$, $\mathbf{U}_2^0, \cdots \mathbf{U}_N^0$;
2. Iterate for $k = 1, 2, \cdots$:
    2.1 Set $\tilde{\mathcal{U}}_n^k = \mathcal{D} \times_1 (\mathbf{U}_1^k)^T \cdots \times_{n-1} (\mathbf{U}_{n-1}^k)^T \times_{n+1} (\mathbf{U}_{n+1}^{k-1})^T \cdots \times_N (\mathbf{U}_N^{k-1})^T$;
    2.2 Obtain $\tilde{\mathbf{U}}_n^k$ by unfolding $\tilde{\mathcal{U}}_n^k$ along the $n-th$ mode;
    2.3 Orthonormalise the columns of $\tilde{\mathbf{U}}_n^k$ and truncate it to $R_n$ columns to obtain $\mathbf{U}_n^k$;
Until $\|\mathbf{U}_n^{k\,T} \cdot \mathbf{U}_n^{k-1}\|^2 > (1-\varepsilon)R_n$, for $n = 1, 2, \ldots, N$,
3. Compute core tensor by $\hat{\mathcal{Z}} = \tilde{\mathcal{U}}_N \times_N \hat{\mathcal{U}}_N^T$ and the rank-reduced approximation $\hat{\mathcal{D}} = \hat{\mathcal{Z}} \times_1 \hat{\mathbf{U}}_1 \times_2 \hat{\mathbf{U}}_2 \cdots \times_N \hat{\mathbf{U}}_N$.
```

However, the HOSVD can only be used when all elements in the input tensor $\mathcal{D}$ are available. In a practical application, it is very difficult to obtain this kind of dataset, especially when the training tensor includes several variations like identity, pose, illumination and expression.

2.2 The M^{2}SA algorithm

To apply Algorithm 1 to an input tensor with missing values, M^{2}SA was proposed in [4]. In the M^{2}SA algorithm, we must first define an index tensor $\mathcal{I}$ which has the same dimensionality as the training tensor $\mathcal{D}$. The value of the elements in $\mathcal{I}_{i_1 i_2 \cdots i_N} = 1$ or 0 depends on whether $\mathcal{D}_{i_1 i_2 \cdots i_N}$ is available or unavailable. M^{2}SA is summarized in Algorithm 2. According to the experimental results in [4], this algorithm performs best when the values of $R_n(n = 1 \cdots N)$ are about 2/3 of the original dimensionality.

Algorithm 2 M^{2}SA Dimensionality Reduction

```
1. Fill the missing elements in training tensor $\mathcal{D}$ with the mean captured over all the available elements sharing the same variations to obtain the initialization of the training tensor $\mathcal{D}^0$;
2. Apply Algorithm 1 to $\mathcal{D}^0$ to get the initial rank-reduced approximation $\hat{\mathcal{D}}^0 = \hat{\mathcal{Z}}^0 \times_1 \hat{\mathbf{U}}_1^0 \times_2 \hat{\mathbf{U}}_2^0 \cdots \times_N \hat{\mathbf{U}}_N^0$;
3. Iterate for $k = 0, 1, 2, \cdots$:
    3.1 Update training tensor by $\mathcal{D}^k = \mathcal{D}. \times \mathcal{I} + \hat{\mathcal{D}}^{k-1}. \times (\sim \mathcal{I})$;
    3.2 Apply Algorithm 1 to $\mathcal{D}^k$ to get the new rank-reduced approximation $\hat{\mathcal{D}}^k = \hat{\mathcal{Z}}^k \times_1 \hat{\mathbf{U}}_1^k \times_2 \hat{\mathbf{U}}_2^k \cdots \times_N \hat{\mathbf{U}}_N^k$;
Until $\|(\mathcal{D}^k - \hat{\mathcal{D}}^k). \times \mathcal{I}\| < \varepsilon$ or $k > Max\_Loop$;
4. Compute the rank-reduced approximation $\hat{\mathcal{D}} = \hat{\mathcal{Z}} \times_1 \hat{\mathbf{U}}_1 \times_2 \hat{\mathbf{U}}_2 \cdots \times_N \hat{\mathbf{U}}_N$.
```

3 Multilinear AAM with missing values

3.1 Shape and appearance models

In classical AAM, both the shape and appearance models are trained from a set of samples across different identity, pose, illumination and expression variations by applying PCA to a set of shape and global appearance training vectors. This leads to a fitting difficulty when the face images contain several different types of variation. It is also known as the generalization problem of AAM. Multilinear analysis can decouple these variations into different subspaces and successfully manage the diversity of these variations. To make the multilinear-based AAM more convenient in real applications, M^{2}SA is used in this section to generate variation-specific AAM models instead of the classical HOSVD.

For an incomplete shape training tensor $\mathcal{S} \in \mathbf{R}^{I_{id} \times I_{pe} \times I_{ill} \times I_{exp} \times I_{cor}}$ and an incomplete appearance training tensor $\mathcal{A} \in \mathbf{R}^{I_{id} \times I_{pe} \times I_{ill} \times I_{exp} \times I_{px} \times I_{py}}$, the use of M^{2}SA implies constructing:

$$\begin{aligned} \mathcal{S} =& \mathcal{Z}_S \times_1 \mathbf{V}_{id} \times_2 \mathbf{V}_{pe} \times_3 \\ & \mathbf{V}_{ill} \times_4 \mathbf{V}_{exp} \times_5 \mathbf{V}_{cor}, \quad (3) \\ \mathcal{A} =& \mathcal{Z}_A \times_1 \mathbf{W}_{id} \times_2 \mathbf{W}_{pe} \times_3 \\ & \mathbf{W}_{ill} \times_4 \mathbf{W}_{exp} \times_5 \mathbf{W}_{px} \times_6 \mathbf{W}_{py}, \quad (4) \end{aligned}$$

where: $\mathcal{Z}_S$ and $\mathcal{Z}_A$ are shape and appearance core tensors; $\mathbf{V}_{id}, \mathbf{V}_{pe}, \mathbf{V}_{ill}, \mathbf{V}_{exp}, \mathbf{V}_{cor}$ are mode matrices of the shape tensor for identity, pose, illumination, expression and the coordinates of the landmarks in shape respectively; $\mathbf{W}_{px}, \mathbf{W}_{py}$ are mode matrices of the appearance tensor based on the resolution of the 2D appearance.

Because the shape is mainly influenced by pose and expression, we can build a set of basis shape tensors:

$$\mathcal{B}_S(v_{i_{pe}}, v_{i_{exp}}) = \mathcal{Z}_S \times_2 \mathbf{v}_{i_{pe}}^T \times_4 \mathbf{v}_{i_{exp}}^T \times_5 \mathbf{V}_{cor} \quad (5)$$

where $i_{pe} = 1, 2 \cdots, I_{pe}$ and $i_{exp} = 1, 2 \cdots, I_{exp}$, $\mathbf{v}_{i_{pe}}^T$ is the $i_{pe} - th$ row of the mode-2 matrix and $\mathbf{v}_{i_{exp}}^T$ is the $i_{exp} - th$ row of the mode-4 matrix.

On the other hand, because the appearance is mainly influenced by pose and illumination, we can build a set of basis appearance tensors:

$$\begin{aligned} \mathcal{B}_A(w_{i_{pe}}, w_{i_{ill}}) =& \mathcal{Z}_A \times_2 \mathbf{w}_{i_{pe}}^T \times_3 \mathbf{w}_{i_{ill}}^T \times_5 \\ & \mathbf{W}_{px} \times_6 \mathbf{W}_{py} \quad (6) \end{aligned}$$

where $i_{pe} = 1, 2 \cdots, I_{pe}$ and $i_{ill} = 1, 2 \cdots, I_{ill}$, $\mathbf{w}_{i_{pe}}^T$ is the $i_{pe} - th$ row of the mode-2 matrix and $\mathbf{w}_{i_{ill}}^T$ is the $i_{ill} - th$ row of the mode-3 matrix.

It is easy to get the shape basis matrix $\mathbf{B}_S(v_{i_{pe}}, v_{i_{exp}})^{I_{cor} \times I_{id} I_{ill}}$ by unfolding $\mathcal{B}_S(v_{i_{pe}}, v_{i_{exp}})$ along the $5th$ mode. Thus, we can build $I_{pe}I_{exp}$ different shape models:

$$\mathbf{s}(v_{i_{pe}}, v_{i_{exp}}) = \mathbf{s}_0(v_{i_{pe}}, v_{i_{exp}}) + \sum_{k=1}^{p} \alpha_k \mathbf{s}_k(v_{i_{pe}}, v_{i_{exp}}), \quad (7)$$

where $\mathbf{s}_0(v_{i_{pe}}, v_{i_{exp}})$ is the mean shape of all the training sets under $i_{pe} - th$ pose and $i_{exp} - th$ expression, and $\mathbf{s}_k(v_{i_{pe}}, v_{i_{exp}})$ is the $k - th$ column of the shape basis matrix $\mathbf{B}_S(v_{i_{pe}}, v_{i_{exp}})$.

For the appearance basis tensor, we can get all basis appearances by fixing the first four modes $\mathcal{B}_A(w_{i_{pe}}, w_{i_{ill}})[i_{id}, i_{pe}, i_{ill}, i_{exp}, 1 \cdots I_{px}, 1 \cdots I_{py}]$. Then we can convert all basis appearances to vectors and obtain the appearance basis matrix $\mathbf{B}_A(w_{i_{pe}}, w_{i_{ill}})^{I_{px}I_{py} \times I_{id}I_{exp}}$. Thus, we can build $I_{pe}I_{ill}$ different appearance models:

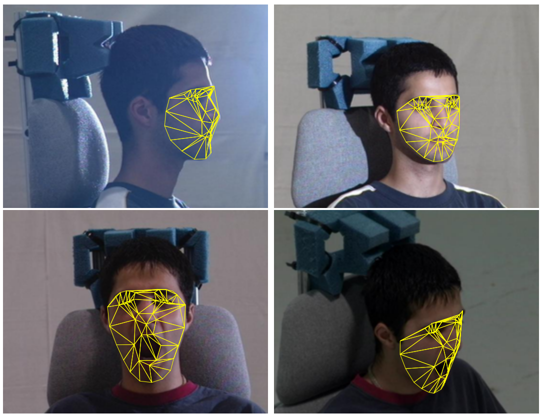


Figure 1. Typical fitted faces with 50% missing values

$$\mathbf{a}(w_{i_{pe}}, w_{i_{ill}}) = \mathbf{a}_0(w_{i_{pe}}, w_{i_{ill}}) + \sum_{k=1}^{q} \beta_k \mathbf{a}_k(w_{i_{pe}}, w_{i_{ill}}), \quad (8)$$

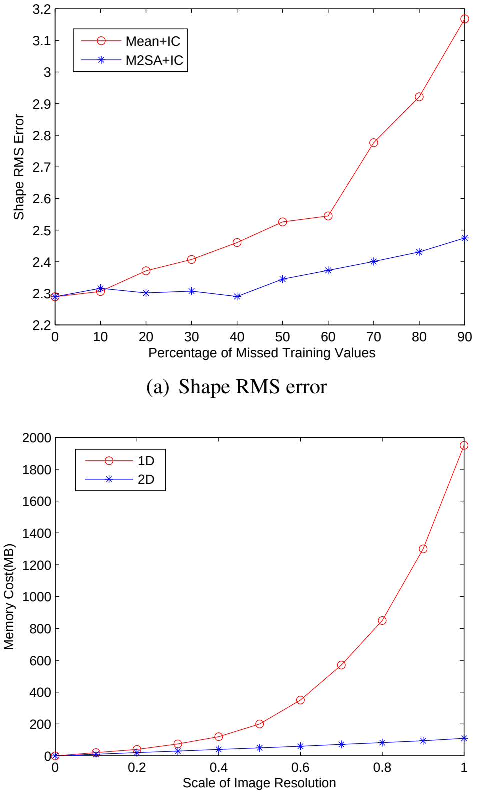
where $\mathbf{a}_0(w_{i_{pe}}, w_{i_{ill}})$ is the mean global texture of all the training sets under the $i_{pe} - th$ pose and the $i_{ill} - th$ illumination, and $\mathbf{a}_k(w_{i_{pe}}, w_{i_{ill}})$ is the $k - th$ column of the appearance basis matrix $\mathbf{B}_A(w_{i_{pe}}, w_{i_{ill}})$.

3.2 Fitting algorithm

For an input image, we assume that the face region has been already detected by a face detection algorithm with a bounding box. The discrete estimation algorithm in [6] is used to estimate the pose, expression and illumination variations. The corresponding shape and appearance models are then selected to fit this input image using the inverse compositional algorithm in [7].

4 Experimental results

The proposed method is tested on the CMU Multi-PIE [5] face database. Because it is laborious to landmark the whole database manually, we randomly select 60 identities under 4 poses (01_0, 04_1, 05_1 and 08_1), 3 expressions (neutral, smile and scream) and 4 illuminations (00, 01, 07 and 13) as our experimental dataset. We randomly choose 30 identities as the training set and the other 30 identities as the test set. Both the training set and test set are landmarked manually for building models and providing the ground truth in



(a) Shape RMS error

(b) Memory cost

Figure 2. Experimental results

the test set. The experiments are tested on a server with 16×3.0 GHz Xeon Processors, 144 GB RAM and Matlab 2011a using Tensor Toolbox 2.5 by Sandia National Laboratories[1].

We randomly remove $10\% - 90\%$ of the values from the training set to test the fitting performance of the proposed algorithm. Fig. 1 shows some typical fitted results of our proposed method when we omit 50% of the values from the training set using the inverse compositional algorithm. Fig. 2(a) shows the shape RMS errors between the fitted shape model and the ground truth against the percentage of missing values. To compare the proposed method with classical TAAM, we insert the missing values using the mean of all available values with the same variations. The fitted shape RMS error of classical TAAM grows faster than the proposed method as the proportion of missing values increases. Also, the proposed method fit the input images well even when we omit 90% of the values from the training set.

To compare the memory costs of tensor decomposition using 1D and 2D appearances, we scale the original images (640*480) to different resolutions. The memory costs of 1D representation increases much faster than that of the 2D representation as the resolution increases. Fig. 2(b) shows that the use of 2D representation can significantly reduce the memory cost.

5 Conclusions

In this paper, we applied the M^{2}SA algorithm to AAM to overcome the difficulty in building shape and appearance models when we have an incomplete training tensor. The experimental results show that the proposed modeling and fitting methods work well even when up to 90% samples are missing from the training set.

Acknowledgments

This work was supported by the Key Grant Project of Chinese Ministry of Education under Grants 311024, National Natural Science Foundation of China under Grants 60973094 and 61103128.

References

[1] T. G. K. Brett W. Bader et al. Matlab tensor toolbox version 2.5. Available online, January 2012.

[2] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *Computer Vision ECCV'98*, volume 1407 of *Lecture Notes in Computer Science*, pages 484–498. Springer Berlin / Heidelberg, 1998.

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

[4] X. Geng, K. Smith-Miles, Z. Zhou, and L. Wang. Face image modeling by multilinear subspace analysis with missing values. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41(3):881–892, 2011.

[5] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

[6] H.-S. Lee and D. Kim. Tensor-based aam with continuous variation estimation: Application to variation-robust face recognition. 31(6):1102–1116, 2009.

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

[8] M. Vasilescu and D. Terzopoulos. Multilinear subspace analysis of image ensembles. In *Proceedings of Computer Vision and Pattern Recognition,2003*, volume 2, pages II–93. IEEE, 2003.

[9] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proceedings of European Conference on Computer Vision*, pages 447–460, 2002.