Warping TensorFaces: Preprocessing Images for Multilinear Analysis of Facial Image Ensembles

Ruslan Shaydulin

May 2, 2018

1 Introduction

In this project we explore multilinear analysis of facial images. Our work is based on a series of papers by Dr. M. Alex O. Vasilescu that started with "Multilinear analysis of image ensembles: Tensorfaces" [VT02a]. Multilinear analysis of facial images is motivated by the fact that the image is a result of multiple factors: age, lighting conditions, pose etc. While we would be more interested in knowing those underlying factors, we can only observe the result (i.e. pixels). Multilinear analysis is a way to utilize those hidden variables in the facial image data. Tensorfaces are based on the idea that multilinear tensor methods (like multilinear PCA) can be used to explicitly model the image in terms of those hidden factors. For a slightly more detailed explanation of the intuition, the reader is referred to this beautifully simple post by Dr. Vasilescu [Vas].

We introduce a preprocessing step that uses Active Appearance Model to warp the training and testing faces into reference shape. This approach allowed us to achieve acceptable performance on smaller dataset than the one used in the original paper. We will discuss the differences between our approach and the approach described in [VT02b], as well as the reasons for the decrease in performance.

2 Related work

Reiterating the points made in the proposal, we want to emphasize that this project doesn't intend to compete with state-of-the-art methods. Since the series of papers by Vasilescu ([VT02a, VT02b, VT03, VT07] are just a small selection), there has been a lot of research into more complex multilinear face models, including tensor-based active appearance models [FKCW16, LK09]. Those approaches are beyond the scope of this project.

Our approach resembles the approach taken by Y.Wang [WZLJ12] in that we combine classical AAM preprocessing with tensor space analysis of processed images.

In this section, we will introduce only the necessary tensor algebra concepts and definitions. We defer the discussion of other concepts and approaches that we build upon to the later sections of the report. To not make the report too technical we will only introduce the relevant concepts as needed.

2.1 Relevant Tensor Algebra

This section is a boiled-down version of Sections 3 and 4 of [VT02a]. The reader is referred to the original paper for extended discussion and additional references.

Tensor is a higher order generalization of a vectors and matrices [VT02b]. A vector is a first-order tensor and a matrix is a second-order tensor.

The scalars are denoted by lowercase letters $(a, b, c, ...)$, vectors by bold lowercase letters $(\mathbf{a}, \mathbf{b}, \mathbf{c}, ...)$, the matrices by uppercase letters $(A, B, C, ...)$ and tensors by calligraphic uppercase letters $(\mathcal{A}, \mathcal{B}, \mathcal{C}, ...)$.

Tensor, also known as n-way array or n-mode matrix, is a multilinear mapping over a set of vector spaces. Tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_N}$ is of order N. An element of tensor $\mathcal{A}$ is denoted as $\mathcal{A}_{i_1 i_2 \cdots i_N}$ or $a_{i_1 i_2 \cdots i_N}$.

A subtensor is a part of a tensors created by fixing some indices. For example, a *fiber* is a vector-valued subtensor created by fixing all indices except one: $a = \mathcal{X}(i_1, i_2, ..., i_{j-1}, :, i_{j+1}, ..., i_N)$. A

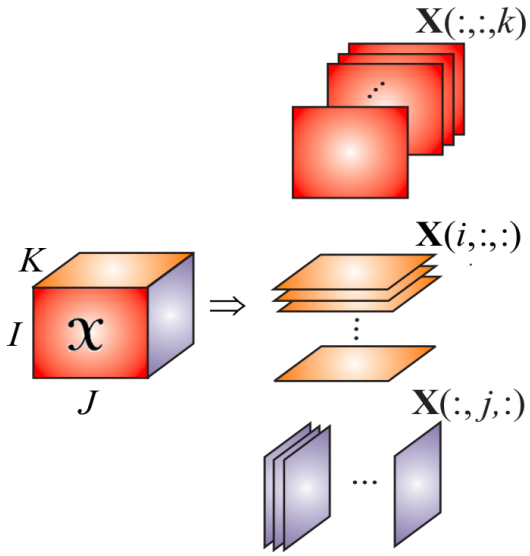

Figure 1: Three ways of visualizing a third-order tensor. A third-order tensor can be understood as a "stack" of matrices. Image from [CMDL+15]

slice is a matrix-valued subtensor created by fixing all indices except two: $A = \mathcal{X}(i_1, i_2, ..., i_{j-1}, :, :, i_{j+2}, ..., i_N)$. Mode-n fibers of the tensor A are vectors created by varying i^{th} index and keeping all others fixed.

Mode-n product of tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by matrix $M \in \mathbb{R}^{J_n \times I_N}$ can be understood as the inner product of mode-n fibers of tensor $\mathcal{A}$ (vectors) with matrix M. More formally, the mode-n product denoted by $\mathcal{A} \times_n M$ is a tensor in $\mathbb{R}^{I_1 \times I_2 \times \cdots \times J_n \times \cdots \times I_N}$ whose entries are computed as follows:

$$(\mathcal{A} \times_n M)_{i_1 i_2 \cdots j_n \cdots i_N} = \Sigma_{i_n} a_{i_1 i_2 \cdots i_n \cdots i_N} m_{j_n i_n} \tag{1}$$

To develop an intuition it is useful to consider a third-order tensor. It can be thought of as a "stack of matrices". See Figure 1 for a visualization (image from [CMDL+15]).

The tensor $\mathcal{D}$ that we are building is $28 \times 5 \times 3 \times 1 \times 8830$ (we will try as much as possible follow the notation in [VT02b] and [VT02a]). We perform tensor decomposition on it to extract the orthogonal spaces corresponding to different parameters like pose, illumination and expression. This requires some explanation.

In two-way Latent Variable Analysis and similar problems the aim is to decompose a data matrix $X \in \mathbb{R}^{I \times R}$ into the factor matrices $A = [a_1, a_2, ..., a_R] \in \mathbb{R}^{I \times R}$ and $B = [b_1, b_2, ..., b_R] \in \mathbb{R}^{J \times R}$ as

$$X = ADB^T + E \tag{2}$$

where $D = diag(\lambda_1, \lambda_2, ..., \lambda_R)$ is a scaling (normalizing) matrix, the columns of B represent the unknown source signals (factors or latent variables) and the columns of A represent the associated mixing vectors. E is noise. This decomposition assumes that the data X has hidden components b_i that are mixed together in an unknown manner through coefficients A [CMDL+15]. Singular Value Decomposition (SVD) is just a special case of 2.

For some data the two dimensional representation is not natural. For example, it is natural to represent time-series data as a third-order tensor (imagine a "stack" of matrices), where each slice is the matrix at a given time point. In our case, it is natural to stack facial images in a multidimensional structure where each dimension corresponds to a certain parameter like pose, illumination or expression.

In terms of mode-n products, decomposition in 2 can be rewritten as $X = D \times_1 A \times_2 B = D \times_1 U_1 \times_2 U_2$ (we will discard the noise matrix). Extending this to tensor of order $N > 2$, "N-mode SVD" orthogonalizes the N spaces spanned by tensor and decomposes the tensor into the mode-n product of N-orthogonal spaces:

$$\mathcal{X} = \mathcal{Z} \times_1 U_1 \times_2 U_2 \cdots \times_N U_N \tag{3}$$

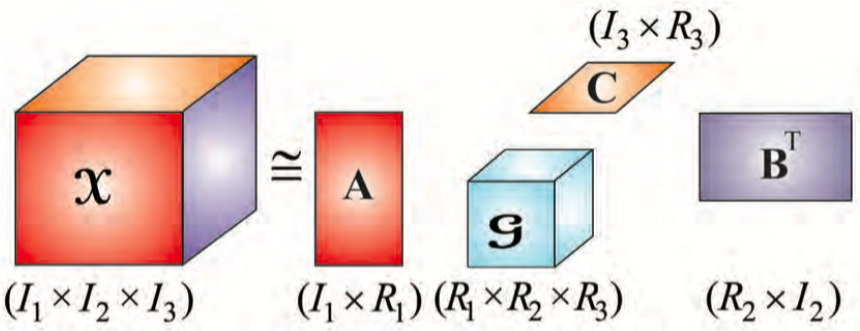

Figure 2: An illustration of tucker decomposition of a third-order tensor. Columns of A, B, C span the signal subspaces for three modes. Note that core tensor $\mathcal{G}$ is not diagonal, representing the complex interactions between tensor components. Image from [CMDL+15]

Tensor $\mathcal{Z}$ is commonly called *core tensor* and is analogous to the diagonal singular value matrix D in two-way SVD. See Figure 2 for an illustration for $N = 3$. However, unlike the two-way case the tensor $\mathcal{Z}$ ($\mathcal{G}$ in the illustration) is in general a full tensor and not diagonal.

3 Technique

In this section we will describe the approach we use. There are two parts to this project. First, we build the multilinear model (in two steps, see Section 3.1). Second, we use this model for facial recognition (Section 3.2).

3.1 Building the model

We use the following two step approach. First step is preprocessing using traditional methods and second step is multilinear analysis. We are utilizing Weizmann face image database as the dataset. Weizmann dataset was chosen because to our best knowledge it is the only freely available dataset that provides multidimensional image data for faces. By multidimensionality here we mean the availability of many different pictures of the same face, structured in the same way. In the example of Weizmann dataset, for each of the 28 male subjects it provides 45 images in 5 viewpoints, 3 illuminations and 3 expressions. In this work we only utilize a subset of the dataset for the reasons that will be described later.

The first step consists of building Active Appearance Models [CET01] for the Weizmann dataset and using them to warp all faces to reference shape. Active Appearance Model is a statistical model of the appearance of image. It learns of a set of model parameters that control the shape and gray-level variation from a training set [CET01]. Then the trained model can be used to fit those parameters to new images. Active Appearance Models (AAMs) are morphable. In this context this fact gives us the option of using the fitted landmarks to warp all faces to refence shape. In this project we take advantage of the Active Appearance Models implemented in Menpo project [AAB+14]. Menpo is a set of Python frameworks for 2D and 3D deformable modeling that includes training and fitting code for many state-of-the-art methods [AAB+14].

We train the Active Appearance Model on the landmarked images of HELEN dataset [LBL+12]. Helen dataset consists of 2000 training images with "highly accurate, detailed, and consistent annotations of the primary facial components" [LBL+12]. We then use this model to landmark the faces from the unlandmarked Weizmann dataset. When the AAM fails to correctly landmark the face, we resort to manual landmarking. For manual landmarking we're using Tim Cootes' set of tools (available here online). The tool we use from that toolset is called am_markup.

For our purposes, the precise mechanics of Active Appearance Models are not very interesting. We only use them as a landmarking tool to warp all images to the reference shape (the goal being to limit the information that our model learns to just the important parts).

To sum up, the first step of the model building process consists of the following parts:

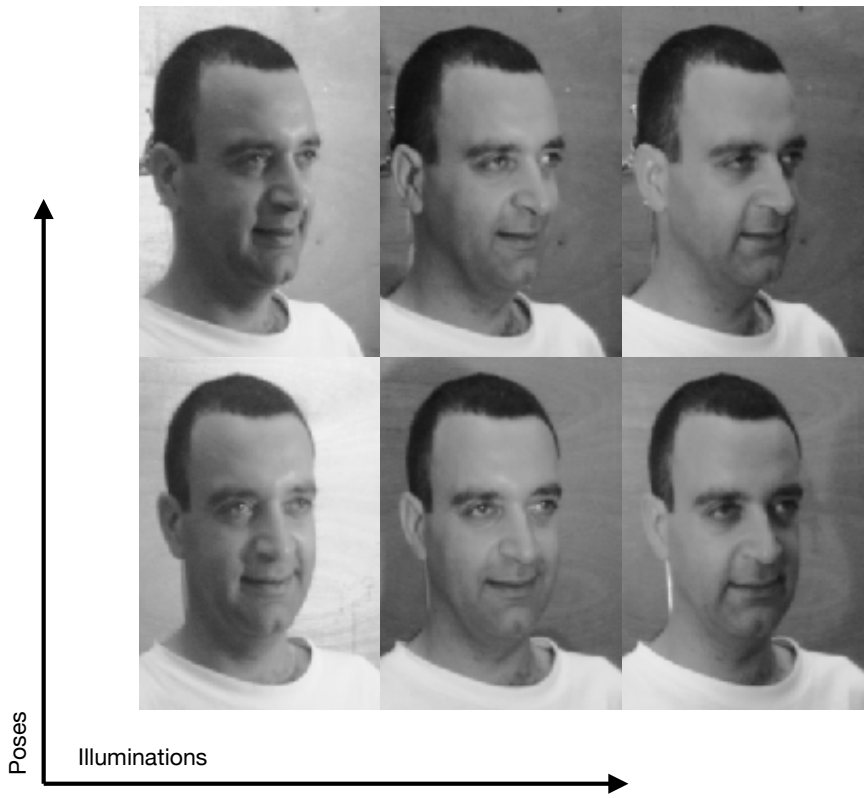

Figure 3: Weizmann dataset provides 45 images in different poses, illuminations and expressions for each person.

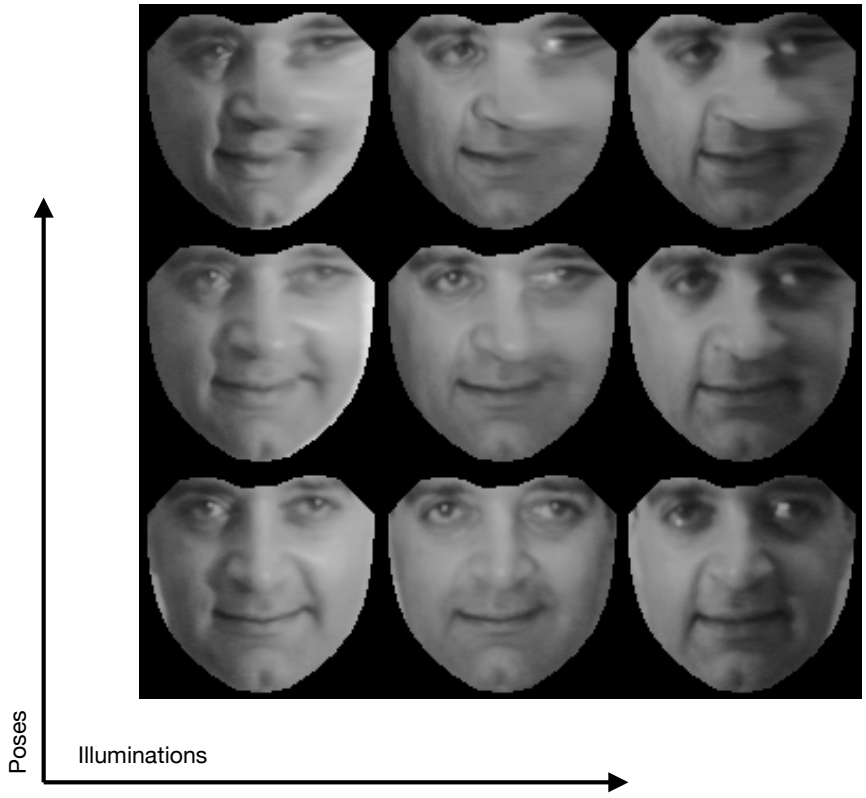

Figure 4: Warped and masked versions of the images.

1. Train Menpo's Active Appearance Model (AAM) on HELEN landmarked dataset
2. Fit the AAM to Weizmann face dataset and export the landmarks
3. Import the landmarks into am_markup and go through all faces. For each face visually check if the landmarking was correct. If it was not, adjust the landmarks manually.
4. Export the corrected landmarks from am_markup and import them back into Menpo.
5. Warp all faces to the reference shape in Menpo and export them as images

For an example of what warped faces look like, see Figure 4.

In the second step we use tensor analysis on pixels of preprocessed images from the first step to build a multilinear model. We build a tensor where each tensor dimension corresponds to a factor like lighting, pose etc and perform n-mode SVD as described in [VT02a]. This will results in a multilinear model that can be used for different applications. In this project, we explore the application to facial recognition. We will now describe the model in detail.

The tensor is built in the following way. Each fiber is a vector obtained by flattening the 110×115 masked image matrix into a vector $\mathbf{f}$ of size 8830. Each vector is normalized. Let $\mathbf{f}$ correspond to person number i, pose number p, illumination ill and expression ex. Then $\mathbf{f}$ is inserted in the tensor in the following way (using MATLAB notation):

$$\mathcal{D}(i, p, ill, ex, :) = f \tag{4}$$

The resulting tensor is of size $28 \times 5 \times 3 \times 1 \times 8830$. After building the tensor, we apply N-mode decomposition described in Section 2.1. We are using the implementation provided in MATLAB framework TensorLab [VDS+16]. The resulting 5-mode decomposition is

$$\mathcal{D} = \mathcal{Z} \times_1 U_{people} \times_2 U_{views} \times_3 U_{illums} \times_4 U_{expres} \times_5 U_{pixels} \tag{5}$$

Figure 5: First three basis vectors (columns) of $\mathcal{Z} \times_5 U_{pixels}$ varying in "people" dimension.

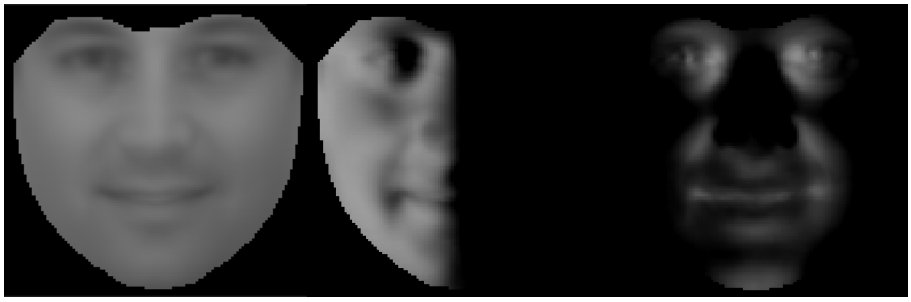

Figure 6: First three basis vectors of $\mathcal{Z} \times_5 U_{pixels}$ varying in "illumination" dimension.

A 28×28 matrix U_{people} orthonormally spans the space of people. Similarly, other matrices orthonormally span corresponding spaces.

Multilinear analysis is advantageous because the core tensor $\mathcal{Z}$ transforms the images in U_{pixels} into *eigenmodes*, representing the variation across different modes (illumination, pose, etc). Figures 5 and 6 illustrate that. For a more detailed discussion of the properties of eigenmodes and tensor decomposition the reader is referred to the original paper [VT02a].

3.2 Facial recognition

The second part of this project consists of using the model built in the first step for facial recognition. In this section is simply a retelling of section 3 of [VT02b], however we include it for completeness nonetheless.

The approach being used is a natural extension of PCA. The general idea is to take a new, previously unseen image, project it into reduced-dimensional space of row vectors of U_{people} and detect it as a person corresponding to the closest of those vectors.

Concretely, we let

$$\mathcal{B} = \mathcal{Z} \times_2 U_{views} \times_3 U_{illums} \times_4 U_{expres} \times_5 U_{pixels} \tag{6}$$

For a new image $\mathbf{d}$, we do the following steps. We index into $\mathcal{B}$ for a view (pose), illumination and expression, obtaining $28 \times 1 \times 1 \times 1 \times 8830$ tensor $\mathcal{B}_{v,i,e}$. We flatted it across people mode to get 28×8830 matrix $B_{v,i,e}$ and use $B_{v,i,e}^{-T}$ to project the new image $\mathbf{d}$ into the space of rows of U_{pixels}. We then iterate through all 28 of those rows to pick one with the least distance from $B_{v,i,e}^{-T}\mathbf{d}$. We repeat for all combinations of parameters and pick the person that results in the best fit.

4 Challenges and issues

In this section we will briefly describe some of the challenges we've faced. The main source of problems was the first step: warping all images to the reference shape. We spent a lot of time trying to produce a fully automated solution. However, it seems to us that Active Appearance Models are struggling with challenging poses and illumination. See Figure 7 for some examples of the images produced.

Figure 7: Some of the images produced by warping using incorrect landmarks. This was caused by AAM incorrectly landmarking faces.

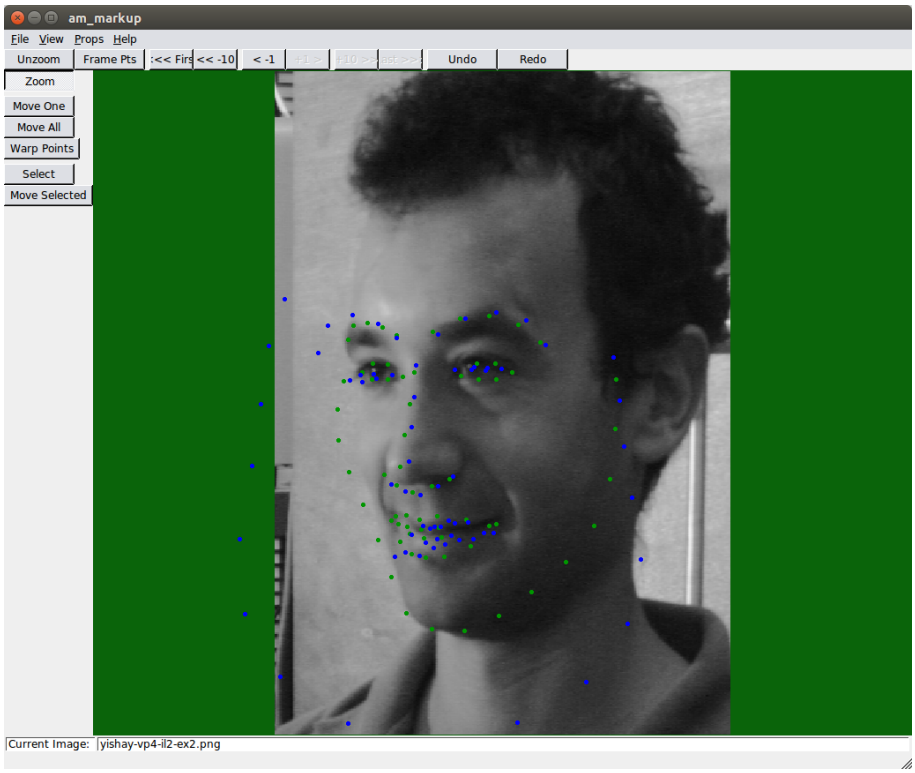

Figure 8: A screenshot of am_markup tool from Tim Cootes' set of tools [CET01]. Blue points represent incorrect landmarks produced by AAM.

Eventually, we resorted to checking all landmark placements manually. Figure 8 presents an example of incorrectly landmarked face (blue dots). We than manually adjust the points to correct locations (green dots). This was a very laborious task, so we had to limit ourselves to only a third of the dataset (420 images, just one out of three expressions). This resulted in lower (if still fairly good) quality of recognition than the one reported in the original paper.

5 Results

To test the quality of the recognition, we run two experiments that split out dataset into training and testing across different modes. The baseline is random guess, which would give $\frac{1}{28} = 3.6\%$ correct guesses. In first experiment, we train our model on images corresponding to first two illuminations (front and left) and use the images corresponding to the third illumination (right). For the first experiment, the recognition rate is 20%. In the second, more successful experiment we split the images across pose mode: the model is trained on images corresponding to poses $0, 2, 4$ (left left, center, right right) and tested on images corresponding to poses $1, 3$ (left, right). This resulted in recognition rate 68%.

Since we don't use the full Weizmann dataset (see Section 4), those numbers are below the rates observer by authors of the original paper (for the second experiment, they've observed recognition rate of 88%).

6 Conclusions

In this project, we used an additional preprocessing step (warping all images to reference shape) to improve the quality of facial recognition using method described in [VT02b]. Unfortunately, it is hard to draw conclusions from this study. We've demonstrated comparable recognition rates on much smaller dataset. However, it is hard to attribute those results to any part of our approach in particular.

Possible and very interesting extensions of this project include using the full dataset (1260 images), using other normalization techniques (for example, cropping to mask instead of warping) and using other tensor analysis techniques.

The full code for this project was made available online at `https://github.com/rsln-s/tensorfaces`.

7 Acknowledgments

I have learned a lot from this project. I would like to thank the people without whom this project would not have happened. First, I would like to thank Dr Eric Patterson for suggesting the idea for this project and providing invaluable guidance and advice. Second, I would like to thank Dr Victor Zordan for his openness and flexibility that made this project possible in the first place.

References

[AAB+14] Joan Alabort-i-Medina, Epameinondas Antonakos, James Booth, Patrick Snape, and Stefanos Zafeiriou. Menpo: A comprehensive platform for parametric image alignment and visual deformable models. In *Proceedings of the ACM International Conference on Multimedia*, MM '14, pages 679–682, New York, NY, USA, 2014. ACM.

[CET01] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001.

[CMDL+15] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2):145–163, 2015.

[FKCW16] Zhen-Hua Feng, Josef Kittler, William Christmas, and Xiao-Jun Wu. A unified tensor-based active appearance face model. *arXiv preprint arXiv:1612.09548*, 2016.

[LBL+12] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S Huang. Interactive facial feature localization. In *European Conference on Computer Vision*, pages 679–692. Springer, 2012.

[LK09] Hyung-Soo Lee and Daijin Kim. Tensor-based aam with continuous variation estimation: application to variation-robust face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(6):1102–1116, 2009.

[Vas] Alex Vasilescu. How are tensor methods used in computer vision and machine learning?

[VDS+16] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer. Tensorlab 3.0, Mar. 2016. Available online.

[VT02a] M Alex O Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *European Conference on Computer Vision*, pages 447–460. Springer, 2002.

[VT02b] M Alex O Vasilescu and Demetri Terzopoulos. Multilinear image analysis for facial recognition. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 2, pages 511–514. IEEE, 2002.

[VT03] M Alex O Vasilescu and Demetri Terzopoulos. Multilinear subspace analysis of image ensembles. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–93. IEEE, 2003.

[VT07] M Alex O Vasilescu and Demetri Terzopoulos. Multilinear (tensor) image synthesis, analysis, and recognition [exploratory dsp]. *IEEE Signal Processing Magazine*, 24(6):118–123, 2007.

[WZLJ12] Y. Wang, Z. Zhang, W. Li, and F. Jiang. Combining tensor space analysis and active appearance models for aging effect simulation on face images. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4):1107–1118, Aug 2012.