Imperial College of Science, Technology and Medicine
Department of Computing

Multilinear Methods for Disentangling Variations with Applications to Facial Analysis

Mengjiao Wang

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of Imperial College London, August 2019

Abstract

Several factors contribute to the appearance of an object in a visual scene, including pose, illumination, and deformation, among others. Each factor accounts for a source of variability in the data. It is assumed that the multiplicative interactions of these factors emulate the entangled variability, giving rise to the rich structure of visual object appearance. Disentangling such unobserved factors from visual data is a challenging task, especially when the data have been captured in uncontrolled recording conditions (also referred to as "in-the-wild") and label information is not available. The work presented in this thesis focuses on disentangling the variations contained in visual data, in particular applied to 2D and 3D faces. The motivation behind this work lies in recent developments in the field, such as (i) the creation of large, visual databases for face analysis, with (ii) the need of extracting information without the use of labels and (iii) the need to deploy systems under demanding, real-world conditions.

In the first part of this thesis, we present a method to synthesise plausible 3D expressions that preserve the identity of a target subject. This method is supervised as the model uses labels, in this case 3D facial meshes of people performing a defined set of facial expressions, to learn. The ability to synthesise an entire facial rig from a single neutral expression has a large range of applications both in computer graphics and computer vision, ranging from the efficient and cost-effective creation of CG characters to scalable data generation for machine learning purposes. Unlike previous methods based on multilinear models, the proposed approach is capable to extrapolate well outside the sample pool, which allows it to accurately reproduce the identity of the target subject and create artefact-free expression shapes while requiring only a small input dataset. We introduce global-local multilinear models that leverage the strengths of expression-specific and identity-specific local models combined with coarse motion estimations from a global model. The expression-specific and identity-specific local models are built from different slices of the patch-wise local multilinear model. Experimental results show that we achieve high-quality, identity-preserving facial expression synthesis results that outperform existing methods both quantitatively and qualitatively.

In the second part of this thesis, we investigate how the modes of variations from visual data can be extracted. Our assumption is that visual data has an underlying structure consisting of factors of variation and their interactions. Finding this structure and the factors is important as it would not only help us to better understand visual data but once obtained we can edit the

factors for use in various applications. Shape from Shading and expression transfer are just two of the potential applications. To extract the factors of variation, several supervised methods have been proposed but they require both labels regarding the modes of variations and the same number of samples under all modes of variations. Therefore, their applicability is limited to well-organised data, usually captured in well-controlled conditions. We propose a novel general multilinear matrix decomposition method that discovers the multilinear structure of possibly incomplete sets of visual data in unsupervised setting. We demonstrate the applicability of the proposed method in several computer vision tasks, including Shape from Shading (SfS) (in the wild and with occlusion removal), expression transfer, and estimation of surface normals from images captured in the wild.

Finally, leveraging the unsupervised multilinear method proposed as well as recent advances in deep learning, we propose a weakly supervised deep learning method for disentangling multiple latent factors of variation in face images captured in-the-wild. To this end, we propose a deep latent variable model, where we model the multiplicative interactions of multiple latent factors of variation explicitly as a multilinear structure. We demonstrate that the proposed approach indeed learns disentangled representations of facial expressions and pose, which can be used in various applications, including face editing, as well as 3D face reconstruction and classification of facial expression, identity and pose.

Acknowledgements

I would like to express my thanks to my supervisor Dr. Stefanos Zafeiriou who guided me through this Ph.D. journey. This work would not have been possible without his constant guidance and support.

Similarly I would like to thank my collaborators for their contributions and support. Particular thanks go to Dr. Yannis Panagakis at Imperial College who has filled the role of a second supervisor for me. I am also thankful to my collaborators at

- Imperial College: Dr. Patrick Snape and Dr. Shiyang Cheng
- Stony Brook University: Zhixin Shu and Dr. Dimitris Samaras
- Disney Research: Dr. Derek Bradley and Dr. Thabo Beeler

It has been an amazing journey with them.

Lastly, I would like to thank the most important people in my life: my family.

Contents

Abstract **i**

Acknowledgements **iii**

1 Introduction **3**

1.1 Motivation and Objectives 3

1.2 Contributions 5

1.3 Statement of Originality 6

1.4 Copyright Declaration 6

1.5 Publications 7

2 Literature Review **8**

2.1 Notations and Multilinear Algebra Basics 8

2.2 General Statistical Methods 11

2.2.1 Singular Value Decomposition (SVD) 12

2.2.2 Principal Component Analysis (PCA) 13

2.2.3 High Order Singular Value Decomposition (HOSVD) 15

2.3 Multilinear Analysis of Faces 17

2.4 Photometric Stereo 19

2.4.1 Calibrated Photometric Stereo 19

2.4.2 Uncalibrated Photometric Stereo 21

2.4.3 Class-specific Uncalibrated Photometric Stereo 23

2.5 Deep Learning for Disentangled Representations Learning 24

2.5.1 Representation Learning 25

2.5.2 Autoencoder (AE) 25

2.5.3 Generative Adversarial Network (GAN) 27

2.5.4 Recent Advances 27

3 Global-Local Multilinear Framework for High-Fidelity 3D Facial Expression Synthesis 30

3.1 Introduction 30

3.2 Related Work 32

3.2.1 3D Face Datasets 32

3.2.2 Expression Synthesis 32

3.3 Global-Local Expression Synthesis 34

3.3.1 Global Model 35

3.3.2 Local Models 36

3.3.3 Local Model Optimization and Reconstruction 39

3.4 Experiments 41

3.4.1 Qualitative Evaluation 44

3.4.2 Ablation Study 47

3.4.3 Quantitative Evaluation 48

3.4.4 Perceptual User Study 51

3.4.5 Data Augmentation 51

3.5 Conclusions 53

4 Unsupervised Tensor Decomposition for Learning the Multilinear Structure of Visual Data 54

4.1 Introduction 54

4.2 Methodology 58

4.2.1 Basic Model 58

4.2.2 Robust Decomposition 61

4.2.3 Rank-constrained Decomposition 66

4.2.4 Graph-regularised Decomposition 71

4.3 Experimental Evaluation 73

4.3.1 Disentangling Illumination and Shape 75

4.3.2 Disentangling Expression and Identity 78

4.3.3 Disentangling Illumination, Expression and Identity 82

4.3.4 Robust Disentanglement of Illumination and Shape 83

4.3.5 Disentanglement of Illumination and Shape with Low-rank Constraints . 87

4.3.6 Semi-Supervised Disentanglement of Expression and Identity 88

4.3.7 Unsupervised Normal Estimation using Deep Learning 89

4.4 Limitations 93

4.5 Conclusions 94

5 Neuro-Tensorial Approach for Learning Disentangled Representations 95

5.1 Introduction 95

5.2 Methodology 97

5.2.1 Facial Texture 97

5.2.2 3D Facial Shape 99

5.2.3 Facial Normals 99

5.2.4 3D Facial Pose 99

5.2.5 Network Architecture 100

5.2.6 Training 104

5.3 Proof of Concept Experiments 108

5.3.1 Disentangling Expression and Identity 108

5.3.2 Disentangling Pose, Expression and Identity 111

5.4 Experiments in-the-wild 111

5.4.1 Expression, Pose and Identity Editing in-the-wild 112

5.4.2 Expression and Identity Interpolation 121

5.4.3 Illumination Editing 121

5.4.4 3D Reconstruction 122

5.4.5 Normal Estimation 124

5.4.6 Quantitative Evaluation of the Latent Space 125

5.5 Limitations 127

5.6 Conclusions 127

6 Conclusion **128**

6.1 Summary of Thesis Achievements 128

6.2 Future Work 131

Bibliography **143**

Abbreviations

AAE Adversarial Autoencoder. 1, 24

AE Autoencoder. v, 1, 24, 25

CNN Convolutional Neural Network. 1

DL Deep Learning. 1

DNN Deep Neural Network. 1, 24

GAN Generative Adversarial Network. v, 1, 24, 27

HOSVD High Order Singular Value Decomposition. iv, 1, 11, 15, 17, 18

KR Khatri-Rao. 1, 11, 23, 24

PCA Principal Component Analysis. iv, 1, 11, 13, 26

SfS Shape from Shading. 1, 19

SH Spherical Harmonics. 1, 22, 23

SVD Singular Value Decomposition. iv, 1, 11, 12, 15, 18

UPS Uncalibrated Photometric Stereo. 1, 19, 22–24

Chapter 1

Introduction

1.1 Motivation and Objectives

Statistical methods that explain variability among observed measurements (data) in terms of a potentially lower number of unobserved, latent, variables are cornerstones in data analysis, image and signal processing, and computer vision.

Factor analysis [FW11] and the closely related Principal Component Analysis (PCA) [Hot33] and Singular Value Decomposition (SVD) are probably the most popular statistical methods to find a single mode of variation that explains the data. Nevertheless, most forms of (visual) data have many different and possibly independent, modes of variations and hence methods such as the PCA are not able to identify them. Consider, for example, a population of faces with differing identities and expressions observed under different views (poses) where the appearance of each face is a result of some multifactor confluence due to identity, expression, and pose variation. In order to disentangle multiple but independent modes of variations, several multilinear (tensor) decompositions have been employed [Tuc66; DDV00; KD80; KB09; Kru89]. For instance, the High Order SVD (HOSVD) [DDV00] is able to identify different modes of variation for identities, expressions, and poses per pixel, from a population of faces, by decomposing a carefully designed data tensor. This method is known as TensorFaces [VT02b].

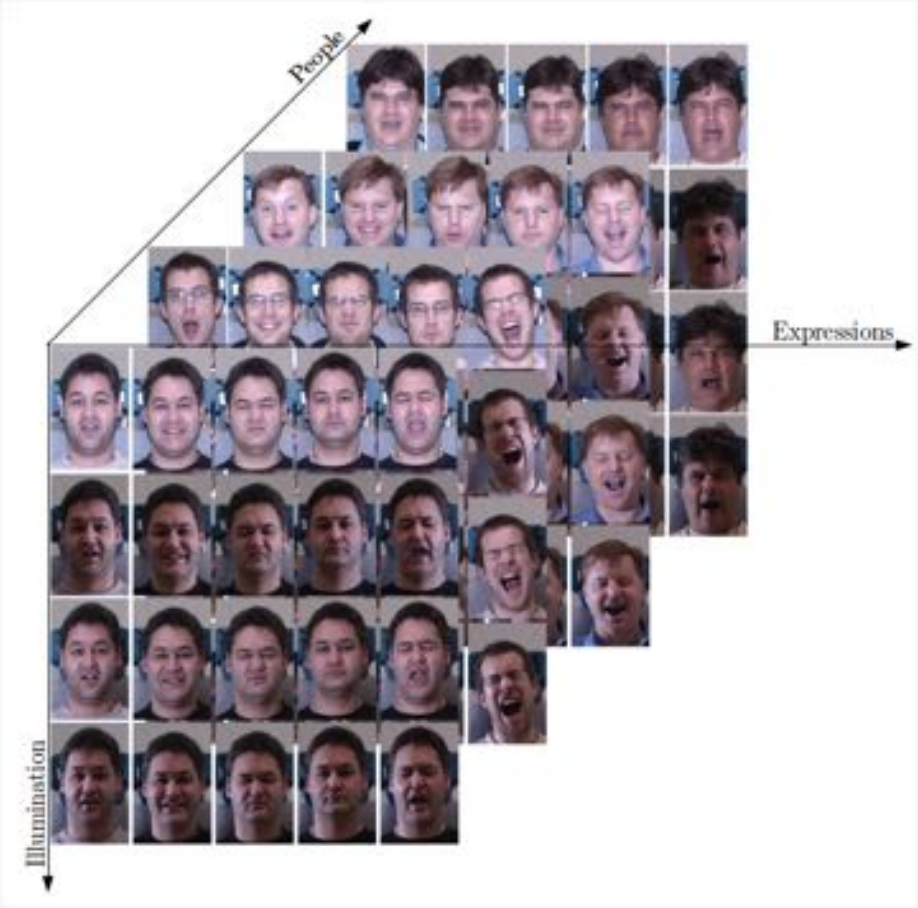


Figure 1.1: Visualisation of the Multi-PIE [Gro+10] dataset. Collecting data where every person is present in all the lighting and expression variations is an expensive process that does not scale well.

The above multilinear decompositions have two main limitations:

1. Traditional multilinear models consider the data globally, assuming that any new sample may be encoded as a linear combination of the basis vectors of the tensor. While this assumptions holds reasonably well for lower resolution geometry, it does not scale well to higher resolution shapes when the number of data samples is remain the same. For higher resolution shapes, the dimensionality of the problem becomes too large and new samples lie far outside of the convex hull spanned by the given data tensor.

2. Multilinear decompositions require a complete data tensor, which has to be built using labels for each mode of variation. That is, in the aforementioned example of faces with varying expression, identity and pose, one needs facial images for every possible expression and pose for each and every person in order to build the required complete tensor. Meth-

ods for completing the tensor[Liu+13; SDS10; GRY11] have been proposed but they are not in the scope of this investigation. Clearly, these requirements limit the applicability of multilinear decompositions to data captured in controlled conditions (e.g., PIE [SBB03], Multi-PIE [Gro+10], visualised in Figure 2.1, and BU-3DFE [Yin+06]), where all the necessary data variations along with their labels are available.

In this thesis, we consider both limitations and investigate ways to overcome them using a variety of methods related to multilinear decompositions.

1.2 Contributions

In this section, we describe the main contributions of this Ph.D. thesis in more detail.

- **Global-Local Multilinear Framework**: Chapter 3 addresses the issue of high-fidelity 3D expression synthesis and argues for a local-global multilinear framework in order to better generalise. We propose a novel global-local multilinear framework. Applied on 3D facial expression synthesis, the model combines a global tensor model to synthesise the coarse deformation with local patch-wise models to generate the detailed shape deformation. The synthesised shapes exhibit expression specific detail while preserving the identity of the subject.

- **Unsupervised Multilinear Model**: Chapter 4 focuses on the problem of disentangling the modes of variation in unlabelled and possibly incomplete data. We consider sets of data that are incomplete in the sense that access to samples exhibiting every possible type of variation is not guaranteed. These sets of data do not contain label information, hence only unsupervised methods, models which do not require label supervision, can be applied to them. To this end, we propose the first *unsupervised multilinear decomposition* which uncovers the potential multilinear structure of incomplete sets of data and the corresponding low-dimensional latent variables (coefficients) explaining different types of variation.

- **Weakly Supervised Autoencoder**: Chapter 5 aims to leverage the unsupervised multilinear method proposed in Chapter 4 as well as recent advances in deep learning. We propose a weakly supervised deep learning method for disentangling multiple latent factors of variation in face images captured in-the-wild. This novel method combines unsupervised tensor decomposition with deep neural networks.

1.3 Statement of Originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work. All sources used have been appropriately referenced.

1.4 Copyright Declaration

1.5 Publications

In this section, we provide a list of publications that were authored during the course of this Ph.D. thesis. The work presented in this thesis is directly related to the following publications:

- **Mengjiao Wang**, Yannis Panagakis, Patrick Snape, and Stefanos Zafeiriou. Learning the Multilinear Structure of Visual Data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- **Mengjiao Wang**, Yannis Panagakis, Patrick Snape, and Stefanos Zafeiriou. Disentangling the Modes of Variation in Unlabelled Data. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.
- **Mengjiao Wang**, Zhixin Shu, Shiyang Cheng, Yannis Panagakis, Dimitris Samaras, and Stefanos Zafeiriou. An Adversarial Neuro-Tensorial Approach for Learning Disentangled Representations. In *International Journal of Computer Vision (IJCV)*, 2019.
- **Mengjiao Wang**, Derek Bradley, Stefanos Zafeiriou, and Thabo Beeler. A Local-Global Multilinear Framework for Facial Expression Synthesis. *Under review*, 2019.

Chapter 2

Literature Review

Learning disentangled representations of data comprises a large corpus of works. In this chapter we will present existing techniques upon which we build methods in the later chapters. We first look into the area of statistical methods e.g. multilinear methods and then delve into photometric stereo, a field focused on disentangling shape and illumination from images. Photometric stereo frequently uses data decomposition techniques to achieve this goal. Finally we investigate recent advances in the field which use deep learning methods. We will briefly introduce each of these techniques.

2.1 Notations and Multilinear Algebra Basics

Tensors are considered as the multidimensional equivalent of matrices (second-order tensors) and vectors (first-order tensors) and denoted by calligraphic letters, e.g., $\boldsymbol{\mathcal{X}}$. Matrices are denoted by uppercase boldface letters (e.g. $\boldsymbol{X}$) and vectors by lowercase boldface letters (e.g. $\boldsymbol{x}$).

$\mathbf{I}$ denotes the identity matrix of compatible dimensions. The ith column of $\mathbf{X}$ is denoted as $\mathbf{x}_i$.

The *outer product* of vectors $\boldsymbol{u} \in \mathbb{R}^m$ and $\boldsymbol{v} \in \mathbb{R}^n$ is defined as

$$\boldsymbol{u} \circ \boldsymbol{v} = \boldsymbol{u}\boldsymbol{v}^T = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_m \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_3 & \dots & v_n \end{bmatrix} = \begin{bmatrix} u_1v_1 & u_1v_2 & \dots & u_1v_n \\ u_2v_1 & u_2v_2 & \dots & u_2v_n \\ \vdots & \vdots & \ddots & \vdots \\ u_mv_1 & u_mv_2 & \dots & u_mv_n \end{bmatrix}$$

The *order* of a tensor is the number of indices needed to address its elements i.e. the number of dimensions. The dimensions of a tensor are also called ways or modes. Each element of an Mth-order tensor $\boldsymbol{\mathcal{X}}$ is addressed by M indices, i.e., $(\boldsymbol{\mathcal{X}})_{i_1,i_2,\dots,i_M} \doteq x_{i_1,i_2,\dots,i_M}$.

The sets of real and integers numbers is denoted by $\mathbb{R}$ and $\mathbb{Z}$, respectively.

A set of M real matrices (vectors) of varying dimensions is denoted by $\{\mathbf{X}^{(m)} \in \mathbb{R}^{I_n \times N}\}_{m=1}^{N}$ ($\{\mathbf{x}^{(m)} \in \mathbb{R}^{I_m}\}_{m=1}^{M}$).

An Mth-order real-valued tensor $\boldsymbol{\mathcal{X}}$ is defined over the tensor space $\mathbb{R}^{I_1 \times I_2 \times \dots \times I_M}$, where $I_m \in \mathbb{Z}$ for $m = 1, 2, \dots, M$.

An Mth-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_M}$ has *rank*-1, when it is decomposed as the outer product of M vectors $\{\mathbf{x}^{(m)} \in \mathbb{R}^{I_m}\}_{m=1}^{M}$. That is,

$$\boldsymbol{\mathcal{X}} = \mathbf{x}^{(1)} \circ \mathbf{x}^{(2)} \circ \dots \circ \mathbf{x}^{(M)} \doteq \bigcirc_{m=1}^{M} \mathbf{x}^{(m)},$$

where $\circ$ denotes for the vector outer product. Hence each element of the tensor is a product of vector elements:

$$x_{i_1,i_2,\dots,i_M} = x_{i_1}^{(1)} x_{i_2}^{(2)} \dots x_{i_M}^{(M)}$$

The *mode-m matricisation* of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_M}$ maps $\boldsymbol{\mathcal{X}}$ to a matrix $\mathbf{X}_{(m)} \in \mathbb{R}^{I_m \times \bar{I}_m}$ with $\bar{I}_m = \prod_{\substack{k=1 \\ k \neq m}}^{M} I_k$ such that the tensor element $x_{i_1,i_2,\dots,i_M}$ is mapped to the matrix element $x_{i_m,j}$ where $j = 1 + \sum_{\substack{k=1 \\ k \neq m}}^{M} (i_k - 1) J_k$ with $J_k = \prod_{\substack{n=1 \\ n \neq m}}^{k-1} I_n$. For example, let a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{3 \times 2 \times 2}$

be

$$\boldsymbol{X}_{::1} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, \boldsymbol{X}_{::2} = \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix}$$

The mode-m matricisations are

$$\boldsymbol{X}_{(1)} = \begin{bmatrix} 1 & 2 & 7 & 8 \\ 3 & 4 & 9 & 10 \\ 5 & 6 & 11 & 12 \end{bmatrix}, \boldsymbol{X}_{(2)} = \begin{bmatrix} 1 & 3 & 5 & 7 & 9 & 11 \\ 2 & 4 & 6 & 8 & 10 & 12 \end{bmatrix}, \boldsymbol{X}_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

The *mode-m (matrix) product* of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ with a matrix $\boldsymbol{U} \in \mathbb{R}^{J \times I_m}$ is denoted by $\boldsymbol{\mathcal{X}} \times_m \boldsymbol{U} \in \mathbb{R}^{I_1 \times \cdots \times I_{m-1} \times J \times I_{m+1} \times \cdots \times I_M}$. Element-wise, it is defined as

$$(\boldsymbol{\mathcal{X}} \times_m \boldsymbol{U})_{i_1 \ldots i_{m-1} j i_{m+1} \ldots i_M} = \sum_{i_m=1}^{I_m} x_{i_1 i_2 \ldots i_M} u_{j i_m}.$$

This can be translated to:

$$\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{X}} \times_m \boldsymbol{U} \iff \boldsymbol{Z}_{(m)} = \boldsymbol{U} \boldsymbol{X}_{(m)}.$$

The *mode-m (vector) product* of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$ with a vector $\mathbf{x} \in \mathbb{R}^{I_m}$ is denoted by $\boldsymbol{\mathcal{X}} \times_m \mathbf{x} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N}$. The result is of order $M-1$ and is defined element-wise as

$$(\boldsymbol{\mathcal{X}} \times_m \mathbf{x})_{i_1, \ldots, i_{m-1}, i_{m+1}, \ldots, i_M} = \sum_{i_m=1}^{I_m} x_{i_1, i_2, \ldots, i_M} x_{i_m}.$$

In order to simplify the notation, we denote

$$\boldsymbol{\mathcal{X}} \times_1 \mathbf{x}^{(1)} \times_2 \mathbf{x}^{(2)} \times_3 \cdots \times_M \mathbf{x}^{(M)} = \boldsymbol{\mathcal{X}} \prod_{m=1}^{m} \times_m \mathbf{x}^{(m)}.$$

The *Kronecker product* of matrices $\mathbf{A} \in \mathbb{R}^{I \times N}$ and $\mathbf{B} \in \mathbb{R}^{J \times M}$ is denoted by $\mathbf{A} \otimes \mathbf{B}$ and yields

a matrix of dimensions $(IJ) \times (NM)$:

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \dots & a_{1N}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \dots & a_{2N}\boldsymbol{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\boldsymbol{B} & a_{I2}\boldsymbol{B} & \dots & a_{IN}\boldsymbol{B} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 & \boldsymbol{a}_1 \otimes \boldsymbol{b}_2 & \dots & \boldsymbol{a}_N \otimes \boldsymbol{b}_{M-1} & \boldsymbol{a}_N \otimes \boldsymbol{b}_M \end{bmatrix}.$$

The *Khatri-Rao (KR)* product of matrices $\mathbf{A} \in \mathbb{R}^{I \times N}$ and $\mathbf{B} \in \mathbb{R}^{J \times N}$ is a column-wise Kronecker product denoted by $\mathbf{A} \odot \mathbf{B}$ and yields a matrix of dimensions $(IJ) \times N$:

$$\boldsymbol{A} \odot \boldsymbol{B} = \begin{bmatrix} \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 & \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 & \dots & \boldsymbol{a}_N \otimes \boldsymbol{b}_N \end{bmatrix}.$$

Furthermore, the Khatri-Rao of a set of matrices $\{\mathbf{X}^{(m)} \in \mathbb{R}^{I_m \times N}\}_{m=1}^{N}$ is denoted by $\mathbf{X}^{(1)} \odot \mathbf{X}^{(2)} \odot \cdots \odot \mathbf{X}^{(M)} \doteq \bigodot_{m=1}^{M} \mathbf{X}^{(m)}$.

Finally, $\| \cdot \|_F$ denotes the Frobenius norm, $\| \cdot \|_*$ the nuclear norm and $\| \cdot \|_1$ the l_1-norm.

More details on tensors and multilinear operators can be found in [KB08].

2.2 General Statistical Methods

Factor analysis [FW11] and the closely related Principal Component Analysis (PCA) [Hot33] and Singular Value Decomposition (SVD) are probably the most popular statistical methods to find a single mode of variation that explains the data. This data is considered to be a data matrix. Each image in the dataset has been vectorised and then stacked together to form this data matrix.

In many fields, multidimensional data called higher-order tensors have been introduced to reformulate and solve problems. In order to apply similar techniques such as matrix SVD to multilinear data, High Order Singular Value Decomposition (HOSVD) has been proposed as a generalisation.

2.2.1 Singular Value Decomposition (SVD)

The matrix SVD is an important factorisation method used in a variety of applications.

The SVD takes a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \tag{2.1}$$

where $\boldsymbol{U} \in \mathbb{R}^{m \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r \times n}$ are orthonormal matrices. The columns of $\boldsymbol{U}$ are called the left-singular vector and the columns of $\boldsymbol{V}$ are called the right-singular vectors. $\boldsymbol{\Sigma} \in \mathbb{R}^{r \times r}$ is a diagonal matrix of singular values σ_i.

$\sigma_i \geq 0$ is a singular value of $\boldsymbol{A}$ if and only if there exist unit vectors $\boldsymbol{u} \in \mathbb{R}^m$ and $\boldsymbol{v} \in \mathbb{R}^n$ s.t.

$$\boldsymbol{A}\boldsymbol{v} = \boldsymbol{\sigma_i}\boldsymbol{u} \tag{2.2}$$

and

$$\boldsymbol{A}^T\boldsymbol{u} = \boldsymbol{\sigma_i}\boldsymbol{v} \tag{2.3}$$

The eigenvectors of $\boldsymbol{A}^T\boldsymbol{A}$ are the right-singular vectors making up the columns of $\boldsymbol{V}$. Similarly, the eigenvectors of $\boldsymbol{A}\boldsymbol{A}^T$ are the left-singular vectors making up the columns of $\boldsymbol{U}$. The singular values in $\boldsymbol{\Sigma}$ are square roots of eigenvalues from $\boldsymbol{A}\boldsymbol{A}^T$ or $\boldsymbol{A}^T\boldsymbol{A}$ and are arranged in descending order.

SVD can be thought of as decomposing the data matrix into a weighted, ordered sum of separable matrices.

$$\boldsymbol{A} = \sum_i \boldsymbol{A}_i = \sum_i \sigma_i \boldsymbol{u}_i \otimes \boldsymbol{v}_i, \tag{2.4}$$

where $\boldsymbol{A}_i$ are separable matrices. $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are the i-th columns of the corresponding SVD matrices $\boldsymbol{U}$ and $\boldsymbol{V}$. σ_i are the ordered singular values.

Another application of SVD is low-rank matrix approximation. We want to approximate $\boldsymbol{A}$

with a matrix $\tilde{\boldsymbol{A}}$ of rank r. The solution is returned by the SVD of $\boldsymbol{A}$:

$$\tilde{\boldsymbol{A}} = \boldsymbol{U}\tilde{\boldsymbol{\Sigma}}\boldsymbol{V}^T, \tag{2.5}$$

where $\tilde{\boldsymbol{\Sigma}}$ contains the r largest singular values of $\boldsymbol{\Sigma}$. The remaining singular values are set to 0.

2.2.2 Principal Component Analysis (PCA)

The PCA is an unsupervised technique to extract variance from datasets. It is an orthogonal projection of the data into a subspace in order to maximise the variance of the projected data.

Given a set of observations $\{(x)_i\}_1^n$ where $\boldsymbol{x}_i \in \mathbb{R}^d$. We aim to project the data onto a subspace with dimensionality $m < d$ where the variance of the projected data is maximised. In the case of $m = 1$, we can define an unit vector $\boldsymbol{w}$. Each data point $\boldsymbol{x}_i$ is projected as $\boldsymbol{w}^T\boldsymbol{x}_i$. Then the mean of the projected data is

$$\boldsymbol{w}^T\overline{\boldsymbol{x}} = \boldsymbol{w}^T\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i \tag{2.6}$$

Hence the variance of the projected data is

$$\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{w}^T\boldsymbol{x}_i - \boldsymbol{w}^T\overline{\boldsymbol{x}})^2 = \boldsymbol{w}^T\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T\boldsymbol{w} = \boldsymbol{w}^T\boldsymbol{S}\boldsymbol{w}, \tag{2.7}$$

where $\boldsymbol{S}$ is the data covariance matrix. In order to maximise the projected variance $\boldsymbol{w}^T\boldsymbol{S}\boldsymbol{w}$, we need to include the normalisation condition $\boldsymbol{w}^T\boldsymbol{w} = 1$ which will be included as a Lagrange multiplier. The maximisation is then on

$$\boldsymbol{w}^T\boldsymbol{S}\boldsymbol{w} + \lambda(1 - \boldsymbol{w}^T\boldsymbol{w}). \tag{2.8}$$

The resulting stationary point is at

$$\boldsymbol{S}\boldsymbol{w} = \lambda\boldsymbol{w}, \tag{2.9}$$

so $\boldsymbol{w}$ is an eigenvector of $\boldsymbol{S}$. Left-multiplying this by $\boldsymbol{w}^T$, returns

$$\boldsymbol{w}^T \boldsymbol{S} \boldsymbol{w} = \lambda. \tag{2.10}$$

So the projected variance will be maximal when $\boldsymbol{w}$ is the eigenvector of $\boldsymbol{S}$ with the largest eigenvalue λ. Projections of the data on this eigenvector is called the first principal component. In the case of $m > 1$, additional principal components are found by maximising the projected variance on all directions orthogonal to the previous principal components. We want to find a set of projections $\{\boldsymbol{w}_i\}_{i=1}^m$, then a data point $\boldsymbol{x}_i$ is projected as

$$\begin{bmatrix} \boldsymbol{w}_1^T \boldsymbol{x}_i \\ \vdots \\ \boldsymbol{w}_i^T \boldsymbol{x}_i \end{bmatrix} = \boldsymbol{W}^T \boldsymbol{x}_i \tag{2.11}$$

The sum of the projected variance in each dimension is

$$\sum_{i=1}^{m} \boldsymbol{w}_i^T \boldsymbol{S} \boldsymbol{w}_i = \mathrm{tr}\,[\boldsymbol{W}^T \boldsymbol{S} \boldsymbol{W}], \tag{2.12}$$

where $\boldsymbol{S}$ is the data covariance matrix as defined in (2.7). In order to maximise $\mathrm{tr}\,[\boldsymbol{W}^T \boldsymbol{S} \boldsymbol{W}]$, we need to include the normalisation condition $\boldsymbol{W}^T \boldsymbol{W} = \boldsymbol{I}$ which will be included as a Lagrange multiplier. The maximisation is then on

$$\mathrm{tr}\,[\boldsymbol{W}^T \boldsymbol{S} \boldsymbol{W}] + \mathrm{tr}\,[\boldsymbol{\Lambda}(\boldsymbol{I} - \boldsymbol{W}^T \boldsymbol{W})]. \tag{2.13}$$

The result is

$$\boldsymbol{S} \boldsymbol{W} = \boldsymbol{W} \boldsymbol{\Lambda}, \tag{2.14}$$

Hence, $\{\boldsymbol{w}_i\}_{i=1}^m$ are the m eigenvectors of $\boldsymbol{S}$ with the m largest nonzero eigenvalues. The eigendecomposition of $\boldsymbol{S}$ would be:

$$\boldsymbol{S} = \boldsymbol{Q} \boldsymbol{\Lambda} \boldsymbol{Q}^{-1}, \tag{2.15}$$

where the columns of $\boldsymbol{Q}$ are eigenvectors and $\boldsymbol{\Lambda}$ is a diagonal matrix containing the eigenvalues

λ_i in decreasing order.

SVD, explained in Section 2.2.1, can also be used to compute PCA. Let $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ be the centred data matrix. The SVD of $\boldsymbol{X}$ is:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T. \tag{2.16}$$

Then the data covariance matrix $\boldsymbol{S}$ of $\boldsymbol{X}$ is:

$$\boldsymbol{S} = \frac{\boldsymbol{X}^T\boldsymbol{X}}{n-1} = \boldsymbol{V}\frac{\Sigma^2}{n-1}\boldsymbol{V}^T = \boldsymbol{V}\frac{\Sigma^2}{n-1}\boldsymbol{V}^{-1}, \tag{2.17}$$

as $\boldsymbol{V}$ is orthonormal. Thus $\boldsymbol{Q} = \boldsymbol{V}$ where the columns of $\boldsymbol{V}$ are eigenvectors of $\boldsymbol{S}$ and $\lambda_i = \frac{\sigma_i^2}{n-1}$ are the eigenvalues of $\boldsymbol{S}$.

A well-known face recognition technology called eigenfaces [TP91] uses PCA to compute its facial representation. Eigenfaces are the principal components of a dataset of faces. [SK87] developed this representation and [TP91] applied it to face detection and recognition.

2.2.3 High Order Singular Value Decomposition (HOSVD)

The HOSVD of a tensor can be viewed as a multilinear generalisation of the matrix SVD.

Let us consider the definition of matrix SVD of $\boldsymbol{A} \in \mathbb{R}^{I_1 \times I_2}$.

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T. \tag{2.18}$$

This can be written as

$$\boldsymbol{A} = \boldsymbol{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)}, \tag{2.19}$$

where $\boldsymbol{U}^{(1)} \in \mathbb{R}^{I_1 \times I_1}$, $\boldsymbol{U}^{(2)} \in \mathbb{R}^{I_2 \times I_2}$ are orthogonal and $\boldsymbol{S} \in \mathbb{R}^{I_1 \times I_2}$. $\boldsymbol{S} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\min(I_1,I_2)})$ is pseudodiagonal and ordered i.e. $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(I_1,I_2)} \geq 0$.

[DDV00] suggested the following generalization: Any tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ can be written

as

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{S}} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \cdots \times_N \boldsymbol{U}^{(N)}, \tag{2.20}$$

where $\{\boldsymbol{U}^{(n)}\}_{n=1}^{N}$ are orthogonal matrices and $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is all-orthogonal and ordered. A subtensor of $\boldsymbol{\mathcal{S}}$, $\boldsymbol{\mathcal{S}}_{i_n=\alpha}$, can be obtained by fixing the nth index to α. All-orthogonality means that two subtensors $\boldsymbol{\mathcal{S}}_{i_n=\alpha}$ and $\boldsymbol{\mathcal{S}}_{i_n=\beta}$ are orthogonal for all possible values of n, α and β where $\alpha \neq \beta$:

$$< \boldsymbol{\mathcal{S}}_{i_n=\alpha}, \boldsymbol{\mathcal{S}}_{i_n=\beta} >= 0 \text{ when } \alpha \neq \beta. \tag{2.21}$$

Ordering means that $\|\boldsymbol{\mathcal{S}}_{i_n=1}\| \geq \|\boldsymbol{\mathcal{S}}_{i_n=2}\| \geq \cdots \geq \|\boldsymbol{\mathcal{S}}_{i_n=I_n}\| \geq 0$.

From (2.20), the following can be deduced:

$$\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{A}} \times_1 \boldsymbol{U}^{(1)^T} \times_2 \boldsymbol{U}^{(2)^T} \cdots \times_N \boldsymbol{U}^{(N)^T}. \tag{2.22}$$

This can be written in matrix format as

$$\boldsymbol{A}_{(n)} = \boldsymbol{U}^{(n)} \boldsymbol{S}_{(n)} (\boldsymbol{U}^{(n+1)} \otimes \boldsymbol{U}^{(n+2)} \ldots \boldsymbol{U}^{(N)} \otimes \boldsymbol{U}^{(1)} \otimes \boldsymbol{U}^{(2)} \ldots \boldsymbol{U}^{(n-1)})^T. \tag{2.23}$$

$\boldsymbol{U}^{(n)}$ can be obtained from the SVD of $\boldsymbol{A}_{(n)}$:

$$\boldsymbol{A}_{(n)} = \boldsymbol{U}^{(n)} \Sigma^{(n)} \boldsymbol{V}^{(n)^T}. \tag{2.24}$$

$\{\boldsymbol{U}^{(n)}\}_{n=1}^{N}$ can be constructed in a similar way by SVD on $\{\boldsymbol{A}^{(n)}\}_{n=1}^{N}$. The resulting $\boldsymbol{\mathcal{S}}$ would satisfy the properties of all-orthogonality and ordering.

Hence the HOSVD can be computed as

1. For all n, construct the n-mode matrix unfolding of $\boldsymbol{\mathcal{A}}$, $\boldsymbol{A}_{(n)}$, and use SVD to find the n-mode left singular matrix $\boldsymbol{U}_{(n)}$.
2. Compute $\boldsymbol{\mathcal{S}}$ as $\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{A}} \times_1 \boldsymbol{U}^{(1)^T} \times_2 \boldsymbol{U}^{(2)^T} \cdots \times_N \boldsymbol{U}^{(N)^T}$.

An application of HOSVD is presented in the next section.

2.3 Multilinear Analysis of Faces

For the past fifteen years, the computer vision community has made considerable efforts to collect databases in controlled conditions that can capture the variations of visual objects such as human faces. Arguably, the most comprehensive efforts were made in order to collect the so-called PIE [SBB03] and Multi-PIE [Gro+10] databases. These databases contain a number of people (i.e., multiple identities) captured under different poses and illuminations, displaying a variety of facial expressions. Thus, this dataset contains many different modes of variation and motivated the use of multilinear decompositions, such as HOSVD [DDV00], in order to disentangle the different modes of variations. The TensorFaces [VT02b] is probably the most popular method in this category.

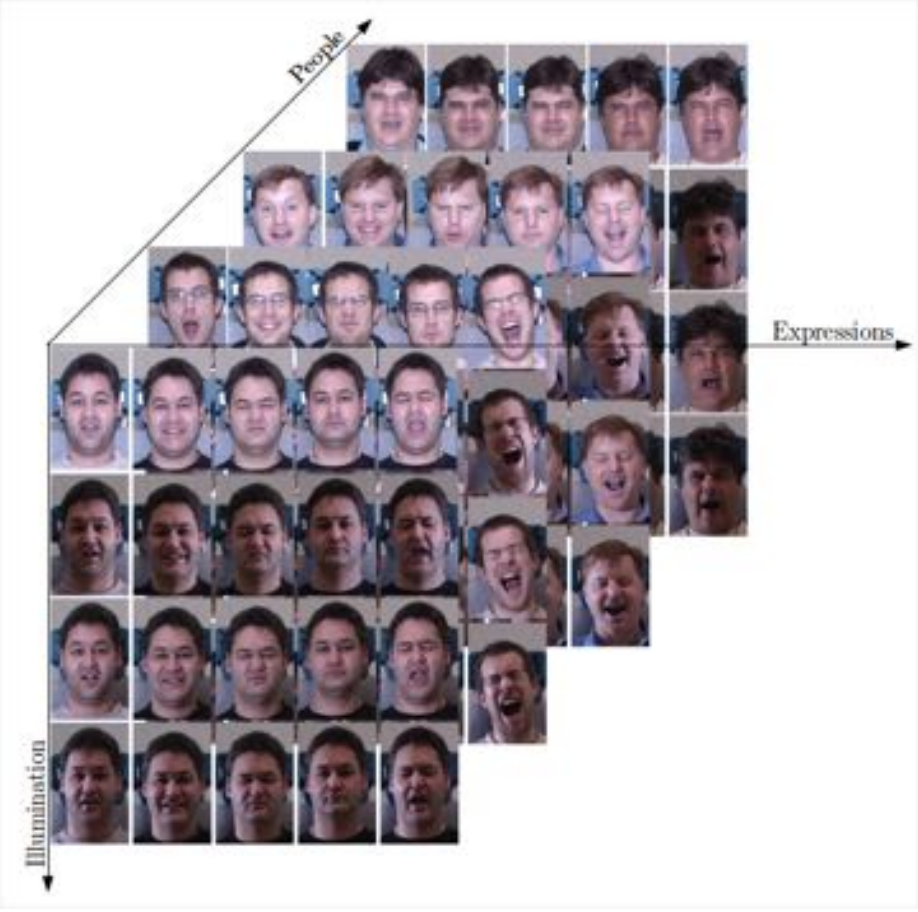


Figure 2.1: Visualisation of the Multi-PIE [Gro+10] dataset. Collecting data where every person is present in all the lighting and expression variations is an expensive process that does not scale well.

Concretely, let $\mathcal{X}$ be a complete data tensor (see Fig 2.1), TensorFaces [VT02b] disentangles

the modes of variation by seeking a decomposition of the form;

$$\mathcal{X} = \mathcal{B} \times_1 \boldsymbol{A}_1 \times_2 \boldsymbol{A}_2 \times_3 \boldsymbol{A}_3 \cdots \times_N \boldsymbol{A}_N, \tag{2.25}$$

where $\mathcal{B}$ is the core tensor of the same size as $\mathcal{X}$ representing the interactions between the factors $\boldsymbol{A}_n$, for $n = 1, \ldots, N$. Equation (2.25) directly corresponds to the HOSVD [DDV00] formulation explained in Section 2.2.3.

For example, on the Weizmann face dataset of 28 subjects in 5 viewpoints, 4 illuminations, 3 expressions and 7943 pixels per image, $\mathcal{X} \in \mathbb{R}^{28\times5\times4\times3\times7943}$ is a tensor. The aim is then to decompose $\mathcal{X}$ as

$$\mathcal{X} = \mathcal{B} \times_1 \boldsymbol{A}_{people} \times_2 \boldsymbol{A}_{views} \times_3 \boldsymbol{A}_{illumns} \times_4 \boldsymbol{A}_{expres} \times_5 \boldsymbol{A}_{pixels}, \tag{2.26}$$

where $\mathcal{B} \in \mathbb{R}^{28\times5\times4\times3\times7943}$ is the core tensor. [VT02b] proposes the following N-mode SVD algorithm to recover this representation:

1. For $n \in \{people, views, illumns, expres, pixels\}$, flatten $\mathcal{X}$ into the matrix $\boldsymbol{X}_{(n)}$ and compute the SVD: $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = SVD(\boldsymbol{X}_{(n)})$. Then set $\boldsymbol{A}_n = \boldsymbol{U}$.

2. Solve for $\mathcal{B}$ as: $\mathcal{B} = \mathcal{X} \times_1 \boldsymbol{A}^T_{people} \times_2 \boldsymbol{A}^T_{views} \times_3 \boldsymbol{A}^T_{illumns} \times_4 \boldsymbol{A}^T_{expres} \times_5 \boldsymbol{A}^T_{pixels}$

Even though TensorFaces and related methods e.g., [QC15] succeed in recovering the modes of variation, their applicability is rather limited since they not only require the data to be labelled but also the data tensor must contain all samples in all different variations. This is the primary reason that such methods are still mainly applied to tightly controlled databases such as PIE and Multi-PIE, visualised in Figure 2.1, and not to possibly not complete data captured "in-the-wild" data.

In addition, traditional multilinear models consider the data globally, assuming that any new sample may be encoded as a linear combination of the basis vectors of the tensor. Given a sufficiently large dataset this assumptions holds reasonably well for lower resolution data, but

it does not scale easily. For higher resolution data, the dimensionality of the problem becomes too large for the same amount of data samples and new samples cannot be well approximated by the multilinear model. Hence, local methods have been gaining in popularity in order to capture the fine-scale details of faces. [BBW14] proposed a local method based on multilinear models. Despite these advances, the problem of high-fidelity 3D facial expression synthesis still requires more investigation. We propose a global-local multilinear framework in Chapter 3 to achieve state-of-the-art high-fidelity 3D facial expression synthesis.

2.4 Photometric Stereo

Facial Shape from Shading (SfS) [Woo80] and Uncalibrated Photometric Stereo in General Lighting [BJK07] also relies heavily on data decomposition. The recovery of 3D shape from images represents an ill-posed and challenging problem. In its most difficult form, this involves recovering a representation of shape for an object from a single image, under arbitrary illumination. However, for any given image, there are an infinite number of shape, illumination and reflectance inputs that can reproduce the image [AP96]. Therefore, shape recovery is commonly performed by relaxing the problem by introducing prior information or by adding constraints, such as in SfS [Woo80]. In particular, Class-specific UPS seeks to recover the shape of the object by exploiting the similarity within the object class. In the case of faces, there are millions of available images that can be utilised to build in-the-wild models. However, recovering shape from these images is incredibly challenging, as they have been captured in completely unconstrained conditions. No knowledge of the lighting conditions, the facial location or the camera geometric properties are provided with the images.

2.4.1 Calibrated Photometric Stereo

In detail, Photometric Stereo aims at estimating the surface normals. A surface normal $\boldsymbol{n} = (n_x, n_y, n_z)$ is the orientation of a vector perpendicular to the tangent plane on the object surface. This is visualised in Fig. 2.2. Surface normals are then used to estimate the surface

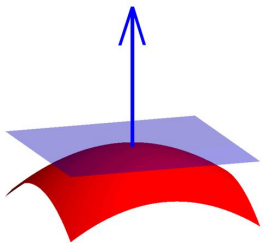


Figure 2.2: Surface normal: orientation of a vector perpendicular to the tangent plane on the object surface

gradients which can be integrated into a depth map $\boldsymbol{z}$. The surface gradients are:

$$(p, q) = (\frac{\partial z(x, y)}{\partial x}, \frac{\partial z(x, y)}{\partial y}). \tag{2.27}$$

The surface normals can be formulated using surface gradients as:

$$\boldsymbol{n} = (\frac{-p}{\sqrt{p^2 + q^2 + 1}}, \frac{-q}{\sqrt{p^2 + q^2 + 1}}, \frac{1}{\sqrt{p^2 + q^2 + 1}})^T, \tag{2.28}$$

where $\boldsymbol{n}$ is a surface normal unit vector at the point (x, y).

Calibrated Photometric Stereo often makes use of the Lambertian surface assumption. In the Lambertian reflectance model, surfaces only emit diffuse reflectance. This means that they reflect light in all direction and obey Lambert's cosine law: "The emitted light is proportional to the cosine of the incidence angle θ_i: the angle between surface normal $\boldsymbol{n}$ and light source $\boldsymbol{l}$." Formally, a convex lambertian surface illuminated by a single light can be expressed as:

$$I(x, y) = \rho cos(\theta_i) = \rho(\boldsymbol{n} \cdot \boldsymbol{l}), \tag{2.29}$$

where $I(x, y)$ is the image irradiance at the point (x, y) and ρ is the surface albedo at the given point (x, y).

Fig. 2.3 visualises the incidence angle θ_i.

Figure 2.3: The incidence angle θ_i: the angle between surface normal $\boldsymbol{n}$ and light source $\boldsymbol{l}$

In order to estimate the surface normals, the formulation is as follows:

$$\boldsymbol{I} = \rho(\boldsymbol{NL}), \tag{2.30}$$

where $\boldsymbol{I}$ is the matrix of known image intensities, $\boldsymbol{L}$ the matrix of light directions and $\boldsymbol{N}$ is the matrix of normals. With 3 or more images,

$$\rho\boldsymbol{N} = (\boldsymbol{L}^T\boldsymbol{L})^{-1}\boldsymbol{L}^T\boldsymbol{I}. \tag{2.31}$$

The surface albedo can be recovered from $\rho = norm(\rho\boldsymbol{N})$. In the next section, we will discuss the cases where the lighting $\boldsymbol{L}$ is unknown.

2.4.2 Uncalibrated Photometric Stereo

Instead of a single point light source illuminating the object, we now consider a collection of directional light source placed at infinity. The lighting intensity can be expressed as a non-negative function of the unit sphere using a sum of spherical harmonics. Spherical harmonics is an orthonormal basis for all functions on the surface of a sphere.

$$I(x,y) = \sum_{n=0}^{\infty}\sum_{m=-n}^{n} \alpha_n l_{nm}\rho(x,y)Y_{nm}(\boldsymbol{n}(x,y)), \tag{2.32}$$

where $\alpha = \pi, \frac{2\pi}{3}, \frac{\pi}{4}\ldots,$ l_{nm} are lighting coefficients and Y_{nm} are surface spherical harmonics functions evaluated at the surface normal $\boldsymbol{n}(x,y)$. As $n \to \infty$, the coefficients tend to zero so

the spherical harmonics can be approximated using the lower order harmonics.

By using the first order SH, 87.5% of the low-frequency component of the lighting is approximated. The first order SH can then be used to recover 3D shape as their discrete approximation directly incorporates the normals of the object:

$$Y_{nm}(\boldsymbol{n}(x,y)) = \rho(x,y)[1, \boldsymbol{n}_x(x,y), \boldsymbol{n}_y(x,y), \boldsymbol{n}_z(x,y)]^T, \tag{2.33}$$

The normals can be integrated to provide a dense 3D surface [FC88a].

Spherical harmonics can be approximated by a low-dimensional linear subspace [BJ03; RH01]. The aim of Uncalibrated Photometric Stereo (UPS) is to approximate $\boldsymbol{X}$, a matrix of images of the same object under unknown lighting:

$$\boldsymbol{X} = \boldsymbol{B}\boldsymbol{P}, \tag{2.34}$$

where $\boldsymbol{B}$ contains the first order spherical harmonics basis images and $\boldsymbol{P}$ is the matrix of lighting coefficients. $\boldsymbol{X}$ can be factorised using SVD (Section 2.2.1):

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \boldsymbol{B} = \boldsymbol{U}\sqrt{\boldsymbol{\Sigma}}, \boldsymbol{P} = \sqrt{\boldsymbol{\Sigma}}\boldsymbol{V}^T. \tag{2.35}$$

This decomposition suffers from ambiguity. Prior work [BJK07] suggested a method to resolve this ambiguity up to a 4×4 Lorentz transformation as follows:

1. Normalise the columns of $\boldsymbol{B}$
2. Construct $\boldsymbol{Q}$ using quadratic terms computed from the rows of $\boldsymbol{B}$
3. Use SVD to construct $\boldsymbol{S}$ that approximates the null space of $\boldsymbol{Q}$
   (a) If $\boldsymbol{S}$ has exactly 1 positive eigenvalue (execute $\boldsymbol{S} = -\boldsymbol{S}$) or 1 negative eigenvalue: Construct $\boldsymbol{A} = \sqrt{\boldsymbol{\Sigma}}\boldsymbol{W}^T$ from eigendecomposition $\boldsymbol{S} = \boldsymbol{W}\boldsymbol{J}\boldsymbol{\Sigma}\boldsymbol{W}^T$
   (b) Else: Find $\boldsymbol{A}$ that minimises $\|\boldsymbol{S} - \boldsymbol{A}^T\boldsymbol{J}\boldsymbol{A}\|$

4. Compute $\boldsymbol{A}\boldsymbol{B}^T$ which is the shape up to a 4×4 Lorentz transformation

UPS is useful as lighting estimations are often unknown but it still requires a number of images of the exact same object under different illumination conditions.

2.4.3 Class-specific Uncalibrated Photometric Stereo

Class-specific UPS seeks to recover the shape of the object by exploiting the similarity within the object class. Recent class-specific UPS techniques [Kem13b; SPZ15] proposed to recover a class-specific Spherical Harmonics (SH) basis that exploits the low-rank structure of faces [BJ03; GBK01]. The recovered SH basis can be robustly learnt from automatically aligned, "in-the-wild" images. The images can be automatically aligned using Procrustes Analysis.

[Kem13b; SPZ15] attempt to build a subspace that explicitly separates shape and appearance by performing a rank constrained Khatri-Rao(KR) factorization [KR68]. KR product has been defined in 2.1. The first paper where the decomposition has been proposed and applied in 3D facial shape reconstruction is [Kem13b]. The method in [Kem13b] was inspired by the decomposition techniques employed in the related area of Structure-from-Motion [BHB00]. [Kem13b] imposes a rank-1 constraint on every matrix formed using a row of $\boldsymbol{P}$:

1. For every image i:
   (a) Construct $4 \times k$ matrix $\boldsymbol{P'}$ from $1 \times 4k$ row $\boldsymbol{p}_i$
   (b) SVD on $\boldsymbol{P'} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, set $\boldsymbol{\alpha} = \boldsymbol{u}(:,1)\boldsymbol{\sigma}(1,1)$ and $\boldsymbol{l} = \boldsymbol{v}(:,1)$
   (c) $\boldsymbol{p}_i = [\boldsymbol{\alpha}\boldsymbol{l}^T]_{1\times 4k}$
2. Estimate $\boldsymbol{B}$ s.t. $min\|\boldsymbol{X} - \boldsymbol{B}\boldsymbol{P}\|^2$

[RH10] proposed that the best least-squares factorisation solution to a KR product is the rank-1 approximation of the matrices formed from each row. [SPZ15] made the link between the KR

factorisation and the UPS and utilised the method by [RH10] to solve the KR factorisation. [SPZ15] reformulated $\boldsymbol{P}$ as $\boldsymbol{L} \odot \boldsymbol{C}$ and proposed the following robust decomposition:

$$\boldsymbol{X} = \boldsymbol{B}(\boldsymbol{L} \odot \boldsymbol{C}), \tag{2.36}$$

where $\boldsymbol{B}$ is the linear basis, $\boldsymbol{L}$ the matrix of first order spherical harmonics lighting coefficients and $\boldsymbol{C}$ the matrix of shape coefficients. While [Kem13b] used optical flow [KS12] based registration, [SPZ15] applied the above robust decomposition to remove outliers from the images. This decomposition suggests a potential tensor formulation. We show that the decompositions proposed in [Kem13b; SPZ15] are very special cases of the unsupervised tensor decomposition proposed in Chapter 4.

2.5 Deep Learning for Disentangled Representations Learning

Another promising line of research for discovering latent representations is unsupervised Deep Neural Network (DNN). Unsupervised DNN architectures include the Autoencoder (AE) [BCV13b], as well as the Generative Adversarial Network (GAN) [Goo+14] or adversarial versions of AE, e.g., the Adversarial Autoencoder (AAE) [Mak+15]. Even though GAN, as well as AAE, provide very elegant frameworks for discovering powerful low-dimensional embeddings without having to align the faces, due to the complexity of the networks, unavoidably all modes of variation are multiplexed in the latent representation. In order to edit an image through the latent representation e.g. make a person smile, a set of images with known label information such as people smiling is needed. By averaging over the latent representation of this set, the smile manifold can be deduced to edit a new image. Only with the use of these labels it is possible to model/learn the manifold over the latent representation, usually as a post-processing step [Shu+17]. Truly disentangle variations in an unsupervised manner is still a key challenge for deep learning. Unsupervised disentanglement methods do exist which we will introduce in this section.

2.5.1 Representation Learning

Representation learning relies on the assumption that any real-world observation $\boldsymbol{x}$ has been generated through a two-step process:

1. a multivariate latent random variable $\boldsymbol{z}$ is sampled from a distribution $P(\boldsymbol{z})$. $\boldsymbol{z}$ is assumed to correspond to meaningful factors of variation of $\boldsymbol{x}$ e.g. content and position of the object in an image.
2. the observation $\boldsymbol{x}$ is sampled from the conditional distribution $P(\boldsymbol{x}|\boldsymbol{z})$

In this way, the high-dimensional data $\boldsymbol{x}$ can be explained by a lower dimensional and semantically meaningful latent variable $\boldsymbol{z}$. At the same time, there is a direct mapping between $\boldsymbol{z}$ and $\boldsymbol{x}$.

A recent line of work has argued that learning a disentangled $\boldsymbol{z}$ is important for representation learning. A disentangled representation should fulfil the following:

- A disentangled representation should separate the distinct, informative factors of variations in the data [BCV13a].
- A change in a single underlying factor of variation $\boldsymbol{z}_i$ should lead to a change in a single factor in the learned representation $r(\boldsymbol{x})$. This assumption can be extended to groups of factors .
- The learnt representation should be useful for downstream tasks or the (semi-)supervised learning of downstream tasks.

2.5.2 Autoencoder (AE)

An autoencoder is a neural network trained to copy its input to its output. The autoencoder consists of two parts: an encoder $\boldsymbol{h} = f(\boldsymbol{x})$ that encodes the data $\boldsymbol{x}$ into a low-dimensional feature space and a decoder $\boldsymbol{r} = g(\boldsymbol{h})$ that decodes it back to the original space. The aim

of autoencoders is for the information to be preserved. Hence, the transformed reconstructed data should be as close to the original data as possible. This property is represented by the reconstruction error:

$$\|\boldsymbol{X} - g(f(\boldsymbol{X}))\|_F^2 \tag{2.37}$$

Autoencoders are information-preserving dimensionality reduction or feature learning methods. In order for the lower dimensional code to reconstruct the original data, it has to contain the most important information of the data. If the encoder f and decoder g are both linear functions, the optimal autoencoder is equivalent to PCA, presented in Section 2.2.2. Instead of linear functions f and g can be any usually any non-linear function in the form of a neural network. The optimisation of (2.37) does not have any explicit constraints on the structure of the latent code. A naive optimisation of (2.37) could recover a solution where f and g are identity functions and thus not learn any meaningful mappings on the original data.

To prevent this, autoencoders are designed to be unable to learn to copy perfectly. In the original formulation, autoencoders use 4 densely connected layers with a sigmoid non-linearity, and a linear bottleneck layer (the last layer of the encoder f) [HS06]. The encoder function f was designed to map the original data to a lower dimensionality code which forced the encoding and decoding functions to uncover meaningful components of the original data distribution. [HS06] was able to reconstruct the data samples with a dimensionality of 28×28 pixels using only a 30-dimensional code.

Recent approaches for unsupervised disentanglement learning are largely based on Variational Autoencoders (VAEs)[KW13]: First a specific prior $P(\boldsymbol{z})$ is assumed on the latent space and the conditional probability $P(\boldsymbol{x}|\boldsymbol{z})$ is parametrised using a deep network. This is the decoder. On the side of the encoder, the distribution $P(\boldsymbol{z}|\boldsymbol{x})$ is approximated using a variational distribution $Q(\boldsymbol{z}|\boldsymbol{x})$. The complete model is then trained end to end by minimising a suitable approximation to the negative log-likelihood. The representation for $r(\boldsymbol{x})$ is the mean of the approximate posterior distribution $Q(\boldsymbol{z}|\boldsymbol{x})$.

2.5.3 Generative Adversarial Network (GAN)

Generative Adversarial Network (GAN) consists of two neural networks: a generator and a discriminator. The generator generates new data instances and the discriminator decides whether this data instance belongs to the original training dataset or not. The goal of the generator is to fool the discriminator into believing that the synthetic samples belong to the true dataset. In order to do this, the generator has to generate samples that are believable enough to fool the discriminator. The discriminator on the other hand aims to correctly distinguish the fake data samples from the true ones.

In detail, the generator takes in a vector of random numbers and returns a generated image. The discriminator sees both the generated images and the images from the ground truth dataset and returns a probability p of authenticity for each image ($0 < p < 1$). 0 represents fake and 1 represents authenticity.

The generator and the discriminator are trained in tandem. Both networks optimise opposing objective functions in a zero-sum game.

2.5.4 Recent Advances

More recently, both supervised and unsupervised deep learning methods have been developed for disentangled representations learning. Transforming autoencoders [HKW11] is among the earliest methods for disentangling latent factors by means of autoencoder capsules. Variational autenencoders (VAE) are frequently applied to disentangled representation learning. Many methods augment the VAE loss with a regulariser: The β-VAE [Hig+] penalises the Kullback-Leiber-Divergence (KL) term in the Evidence Lower Bound (ELBO) between the prior and $Q(\boldsymbol{z}|\boldsymbol{X})$ to constrain the capacity of the VAE bottleneck. The challenge with this approach is that the reconstruction quality decreases when disentanglement improves. Different approaches have been proposed to remedy this. The AnnealedVAE [Bur+18] progressively increase the bottleneck capacity so that the encoder can focus on learning one factor of variation at the time. β-TCVAE [Che+18a] and FactorVAE [KM18] decompose the KL term into mutual information

and total correlation and penalise the total correlation with a biased Monte-Carlo estimator and with adversarial training respectively. The DIP-VAE-I and the DIP-VAE-II [KSB17] penalise the mismatch between the aggregated posterior and a factorized prior.

In [DCB12] hidden factors of variation are disentangled via inference in a variant of the restricted Boltzmann machine called hossRBM. hossRBM can disentangle emotion from identity on the Toronto Face Dataset in an unsupervised manner. Disentangled representations of input images are obtained by the hidden layers of deep networks with supervision in [Che+14] and through a higher-order Boltzmann machine in disBM [Ree+14]. disBM learns a disentangled representation by clamping a part of the hidden units for a pair of data points that are known to match in all but one factors of variation. The Deep Convolutional Inverse Graphics Network (DC-IGN) [Kul+15] extends this clamping idea to VAE and successfully learns a representation that is disentangled with respect to transformations such as out-of-plane rotations and lighting variations. Both disBM and DC-IGN yield impressive results but require supervision.

Many methods also employ Generative Adversarial Network (GAN) for disentangled representation learning. InfoGAN[Che+16] penalises the mutual information of a categorical input to the generator and the output of an auxiliary network identical to the discriminator apart from the last layer. InfoGAN achieves good disentangling results despite being an unsupervised method. [Mat+16] combines both GAN and VAE to separate style and content in a weakly supervised fashion. With supervision, [Wan+17a] is able to disentangle the factors of the Multi-PIE dataset in a GAN framework. Methods in [Tew+17; TYL17] also extract disentangled and interpretable visual representations such as 3D shape and pose respectively by employing adversarial training. StyleGAN [KLA19] is also able to separate content and style at different scales via its proposed generator architecture.

Recent works in face modeling [Tew+18; TL18] also employ self-supervision or weak supervision (i.e. pseudo-supervision) to learn 3D Morphable Models from images. They rely on the use of a 3D to 2D image rendering layer to separate shape and texture. As multilinear analysis is a powerful technique, recent works have looked into combining multilinear models with deep learning. [AWB18] proposed an autoencoder with a convolutional neural network-based encoder

and a multilinear model-based decoder.

The method in [Shu+17] disentangles the latent representations of illumination, surface normals, and albedo of face images using an image rendering pipeline. Trained with pseudo-supervision, [Shu+17] undertakes multiple image editing tasks by manipulating the relevant latent representations. Nonetheless, this editing approach still requires labels on which expression is in which image, as well as sufficient sampling of a specific expression.

Contrarily to [Tew+18; TL18] the proposed network in Chapter 5 does not render the 3D shape into a 2D image. Nor does the proposed methods use a multilinear model-based decoder as in [AWB18]. Trained using weak supervision from a 3D morphable model, our method is able to disentangle variations such as illumination, expression, pose and identity from face images. In addition, it also learns the components of a 3D morphable model due to the supervision.

Chapter 3

Global-Local Multilinear Framework for High-Fidelity 3D Facial Expression Synthesis

3.1 Introduction

Most approaches for facial animation [DN08], model-based face reconstruction [Zol+18], or facial performance capture employ some form of blendshapes [Lew+14] as a deformation subspace. For high-quality facial rigs it is not uncommon to use hundreds or even thousands of blendshapes for a single person[1]. Building such a rig is highly involved, and typically requires to capture a human subject under a large range of expressions, or to manually sculpt these shapes when creating an imaginary person. Hence these high-quality person-specific facial rigs are currently only practical for hero assets in high budget feature films.

On the other end of the spectrum, consumer-based facial capture methods typically employ pre-built multilinear blendshape models rather than person-specific ones. In order to remain generic, however, these multilinear models must contain many identities with a shape per expression per identity, and hence require even more shapes to be acquired. As a result, these

[1] https://www.3lateral.com

multi-identity models typically contain only a very limited number of expressions or identities, and as a consequence inherently limit the quality that may be achieved.

When capturing or manually sculpting shapes is not an option, shape synthesis can provide a viable alternative. Starting from a single neutral scan or sculpt of the target subject, a synthesis technique can generate the corresponding expression meshes automatically. A good synthesis algorithm should satisfy several criteria - 1) the identity of the target subject must be maintained within all synthesized expressions, 2) the generated expressions should contain plausible deformation at both the global and local scale, and 3) the resulting meshes should be free from geometric artifacts like local pinching, abnormal stretching and triangle flipping. Previous methods for synthesizing facial expressions tend to fall short on one or more of these criteria. Our proposed method can synthesize the full set of expression blendshapes while generally satisfying all criteria, which we will demonstrate both quantitatively as well as qualitatively through a user study.

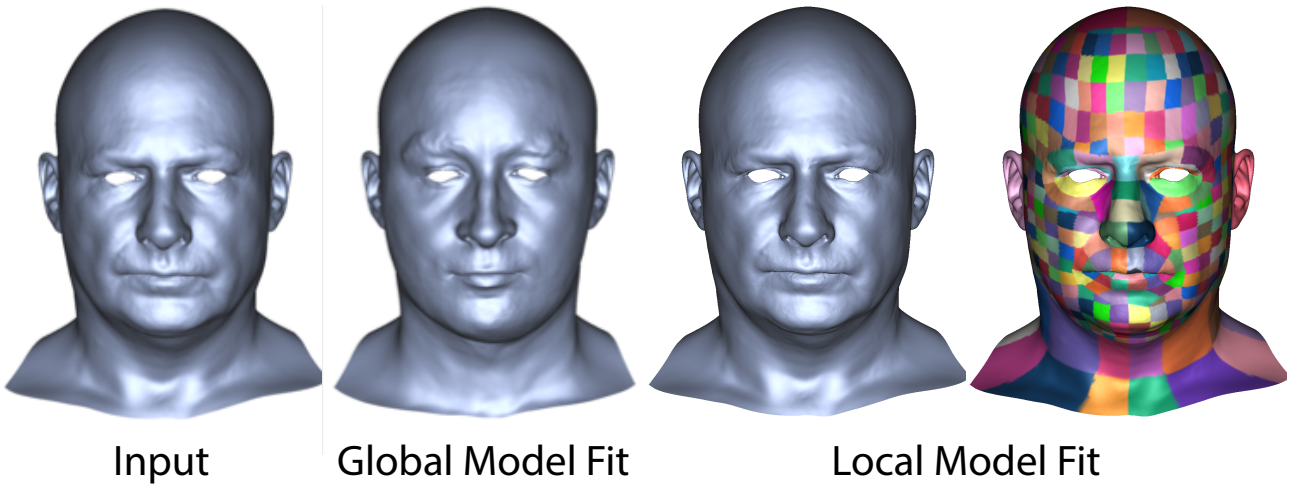


Figure 3.1: Unlike traditional global models, our proposed global-local model is able to extrapolate outside of the training set.

Our method is data-driven based on a limited number of samples (∼50), but unlike traditional multilinear models it does create plausible shapes that lie outside of this dataset since it combines a global tensor model to synthesize the coarse deformation with local patch-wise models to generate the detailed shape deformation, visualized in Fig. 3.1. The expression-specific and identity-specific local models are built from different slices of the patch-wise local multilinear model and provides a way to generate natural-looking detailed expressions which stay plausible

for the identity of the target subject. This allows on the one hand to effortlessly create high-quality facial rigs also for background characters or lower budget productions, and on the other hand to augment existing multi-identity models such as [Boo+16b] with expressions, which will in turn benefit methods that leverage multilinear morphable models for fitting and performance capture.

3.2 Related Work

3.2.1 3D Face Datasets

In recent years, various databases have been collected for expressive 3D faces. One of the earliest, BU-3DFE [Yin+06] contains 100 subjects over 7 expressions (neutral and the six prototypic expressions). More recently 4D facial expression databases such as BU-4DFE [Yin+08] and 4DFAB[Che+18c] have been released. 4DFAB[Che+18c] is currently the largest with 180 people captured. Despite these big efforts, the amount of available 3D facial expression data is still limited, hindering the performance of multilinear models built from them. On the other hand, 3D face datasets containing only neutral scans are often several orders of magnitude larger. Booth et al. [Boo+16b], for example, collected 10,000 faces to build a 3D morphable face model. Our proposed method would be able to augment these datasets with expressions.

3.2.2 Expression Synthesis

Facial expression synthesis has been an active research topic. Prior work can be mainly summarized in two categories. The first category focuses on generative models such as multilinear models whereas the second category comprises of computer graphics techniques which directly warp the input face mesh to generic target expressions. We split the discussion into 2D and 3D expression synthesis.

2D Expression Synthesis

In 2D, a lot of work has been completed on facial expression synthesis. Especially, Generative Adversarial Networks (GANs) have shown a lot of promise for this task. StarGAN [Cho+18] displays successful results for facial expression synthesis by conditioning the generation with images of a specific domain. Here, such a domain would be a set of images of persons sharing the same expression. Pumarola et al [Pum+18] proposed an unsupervised extension of this using Action Units (AUs). Other works [Shu+17; Wan+19] have investigated the potential of incorporating 3D models in editing the expression of the face in 2D. Some work on 2D facial reenactment [Thi+15] have relied on deformation transfer in for 3D meshes [SP04] to transfer facial expressions.

3D Expression Synthesis

By carefully designing a data tensor according to modes of variation such as identity and expression, TensorFaces [VT02a] is able to analyze each mode linearly, since each mode is allowed to vary in turn, while the remaining modes are held constant. A new identity of a given expression can be projected into the core tensor by fixing the expression mode. Once the identity component is estimated, the identity mode can be fixed while the expression mode can be varied to generate new expressions for the input identity.

Blanz and Vetter [B+99] proposed 3D Morphable Models (3DMMs) to model the shape and texture of the human face. Based on this, Blanz et al. [Bla+03] transferred expressions from a common expression framework to reanimate a face in a still image or video. Later, Vlasic et al. [Vla+05] proposed multilinear face models for facial expression tracking and transfer. In order to construct their multilinear models from an incomplete set of face scans they applied an expectation-maximization approach to fill in the missing data. Wang et al. [Wan+18] proposed an unsupervised method to build a multilinear model from partial data using a custom tensor decomposition. FaceWarehouse [Cao+14b] introduced a 3D facial expression database. They built a bilinear face model from the data and showed that it can be used to estimate face identities and expressions for facial images and videos. Recently [Li+17] introduced a linear

model of expression, which they trained on very limited number of people (∼11) on a low-dimensional head model. Local methods based on multilinear models have been proposed by [BBW14] which would be able to capture fine-scale details but it is not guaranteed to produce high-quality expression synthesis.

Expression cloning where one person's expression is transferred onto another person's neutral face is a popular technique for synthesizing facial expressions in computer graphics. Sumner and Popovic [SP04] proposed nonlinear deformation transfer to transfer the 3D deformations from a source mesh to a target mesh. An elastic model was proposed by [Zha+14] to balance the global and local warping effects aimed at 2D facial expression synthesis. While approaches based on global multilinear models often leverage the whole dataset and estimate person-specific expressions, they do not capture details well. Approaches based on local multilinear models are able to capture details but do not necessarily achieve high-quality expression synthesis. On the other hand, expression cloning approaches do not attempt to reflect the expression of the target person but can achieve high-quality expression synthesis by simply cloning the expression details of the source. In this work, we propose a novel method that leverages the advantages of multilinear models in predicting person-specific expressions and at the same time achieves high-quality synthesis that preserves the identity of the target subject.

3.3 Global-Local Expression Synthesis

We now present our global-local multilinear framework for expression synthesis and discuss how it can be used to generate novel expressions given a target subject. Figure 3.2 shows an overview of the framework.

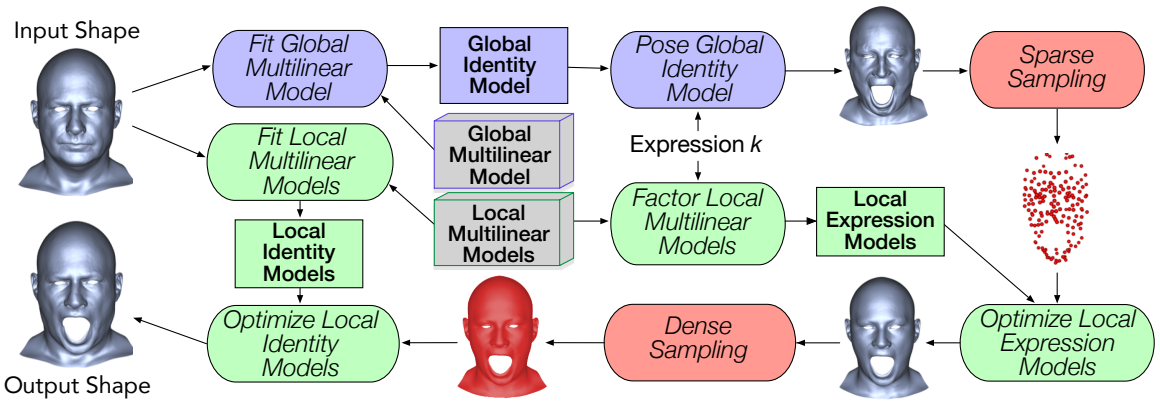


Figure 3.2: Our approach uses a global multilinear model (blue) to estimate coarse deformation of the new expression. A sparse sampling of the global model result is used to then optimize the local models (green). The local expression models are optimized first to return a plausible new expression. The resulting shape is densely sampled to provide constraints when optimizing the local identity models which predict the identity of the input shape. The result is a high-quality, plausible new facial expression of the input shape. Actions such as model fitting are represented in beveled boxes.

3.3.1 Global Model

We start by building a traditional multilinear model [VT02a] using our training data $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{n_v \times n_{id} \times n_{exp}}$.

$$\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{C}} \times_1 \boldsymbol{U}_{vertices} \times_2 \boldsymbol{U}_{identities} \times_3 \boldsymbol{U}_{expressions}, \tag{3.1}$$

where the core tensor $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{n_v \times n_{id} \times n_{exp}}$ governs the interactions between the different factors. The $n_{id} \times n_{id}$ mode matrix $\boldsymbol{U}_{identities}$ spans the space of people parameters, whereas the $n_{exp} \times n_{exp}$ mode matrix $\boldsymbol{U}_{expressions}$ spans the space of expression parameters. The $n_v \times n_v$ mode matrix $\boldsymbol{U}_{vertices}$ spans the space of 3D meshes. As the model is built from the entire face mesh, we call this our *global multilinear model*.

Global Model Fitting. The global model can be fit to a target neutral mesh $\boldsymbol{t}$ by selecting the row from $\boldsymbol{U}_{expressions}$ that corresponds to the neutral expression coefficients, denoted $\boldsymbol{u}_{exp}^{neutral}$, and then estimating the unknown identity coefficients $\boldsymbol{u}_{id}^{\boldsymbol{t}}$ of $\boldsymbol{t}$ within the model as

$$\boldsymbol{t} = (\mathcal{C} \times_1 \boldsymbol{U}_{vertices} \times_3 \boldsymbol{u}_{exp}^{neutral}) \times_2 \boldsymbol{u}_{id}^{\boldsymbol{t}}, \tag{3.2}$$

where $\boldsymbol{u}_{id}^{\boldsymbol{t}}$ is solved in a least-squares sense.

Global Model Posing. Once the identity coefficients are known, we can then predict the set of expressions $\boldsymbol{E^t} \in \mathbb{R}^{n_v \times n_{exp}}$ corresponding to $\boldsymbol{t}$ using the global model by solving

$$\boldsymbol{E^t} = \mathcal{C} \times_1 \boldsymbol{U}_{vertices} \times_2 \boldsymbol{u}_{id}^{\boldsymbol{t}} \times_3 \boldsymbol{U}_{expressions}, \tag{3.3}$$

or for a particular expression k

$$\boldsymbol{e}^{\boldsymbol{t},k} = \mathcal{C} \times_1 \boldsymbol{U}_{vertices} \times_2 \boldsymbol{u}_{id}^{\boldsymbol{t}} \times_3 \boldsymbol{u}_{exp}^{k}. \tag{3.4}$$

Sampling Coarse Deformation. The global model can represent important coarse deformation of the face when posing into expressions, but the local details tend to be inaccurate (see Fig. 3.4 and the discussion in Section 3.3.3). Thus we wish to extract only the coarse expression deformation, which we do by sampling the surface at a sparse set of locations, inspired by the commonly-used paradigm of marker-based motion capture. Specifically, for a given expression k we select a subset of J vertices on the input mesh $\boldsymbol{t}$: $V^{\boldsymbol{t}} = \{\boldsymbol{v}_1^{\boldsymbol{t}}, ..., \boldsymbol{v}_J^{\boldsymbol{t}}\}$, and the corresponding vertices on the neutral expression $\boldsymbol{e}^{\boldsymbol{t},n}$ as well as $\boldsymbol{e}^{\boldsymbol{t},k}$, denoted V^n and V^k respectively. We then compute the sparse motion deltas induced by expression k and add them to the input mesh $\boldsymbol{t}$ to obtain sparse vertex samples V^c, as

$$\boldsymbol{v}_j^c = \boldsymbol{v}_j^{\boldsymbol{t}} + (\boldsymbol{v}_j^k - \boldsymbol{v}_j^n), \tag{3.5}$$

for each $\boldsymbol{v}_j^c \in V^c$ and corresponding $\boldsymbol{v}_j^{\boldsymbol{t}} \in V^{\boldsymbol{t}}$, $\boldsymbol{v}_j^k \in V^k$ and $\boldsymbol{v}_j^n \in V^n$. These sparse vertex positions encoding the coarse expression deformation are then passed to our local model fitting stages.

3.3.2 Local Models

A key component of our approach is to obtain the finer scale expression details using local multilinear models, built on subsets of the mesh vertices, distributed spatially across the face. We refer to a local subset of vertices as a *patch*. Fig. 3.3 illustrates a semantically-meaningful

artist-generated patch layout that we used (although our method can be applied with any layout), and note that patches are increased in size by 20% to provide sufficient overlap between neighboring patches for retaining smoothness in the final reconstruction. For each patch i, a local multilinear model is created from the data tensor $\mathcal{D}_i \in \mathbb{R}^{n_{v_i} \times n_{id} \times n_{exp}}$, where n_{v_i} is the number of vertices in patch i. This is analogous to the global model,

$$\mathcal{D}_i = \mathcal{C}_i \times_1 \boldsymbol{U}_{vertices_i} \times_2 \boldsymbol{U}_{identities_i} \times_3 \boldsymbol{U}_{expressions_i}, \tag{3.6}$$

however it is important to note that the patches in $\mathcal{D}_i$ are first rigidly aligned to the mean patch shape $\boldsymbol{m}_i$ computed over all identities and expressions, thus completely removing rigid patch motion and retaining only local deformations like stretching, compression, bulging, etc. The individual local multilinear models each contain expression and identity modes, as illustrated in Fig. 3.3.

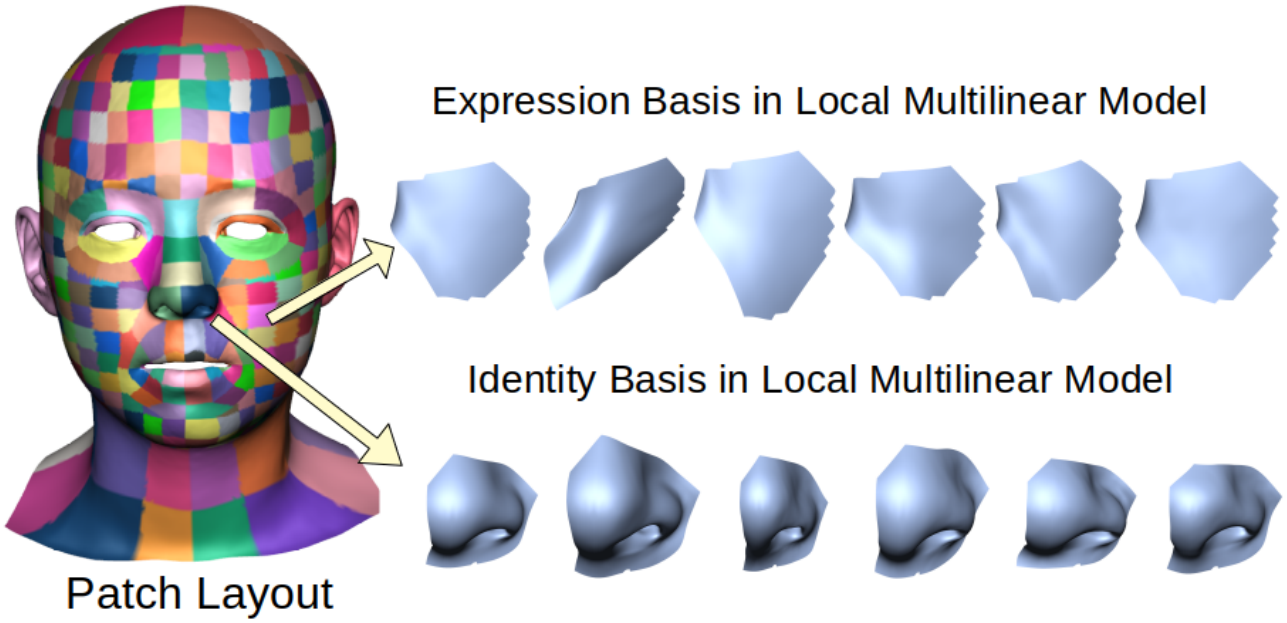


Figure 3.3: We construct a local multilinear model for each patch in the given patch layout. Each patch has its own expression basis and identity basis. A patch containing the nasolabial fold is visualized with its mean shape and top 5 PCA expression basis. We also visualize a nose patch in terms of its mean and top 5 PCA identity basis. Basis vectors are shown at $+3\sigma$.

Patch i can then be posed into a shape $\boldsymbol{x}_i$ using the patch reconstruction model (inspired by [Wu+16])

$$\boldsymbol{x}_i = M_i(\boldsymbol{m}_i + \sum_{f=1}^{F} \alpha_i^f \boldsymbol{b}_i^f), \tag{3.7}$$

where M_i is the rigid motion of the patch, $\{\boldsymbol{b}_i^1, ..., \boldsymbol{b}_i^F\} = \boldsymbol{B}_i$ is a deformation subspace for the patch consisting of F components, and $\{\alpha_i^1, ..., \alpha_i^F\}$ are the coefficients of the deformation basis. In terms of the deformation basis, we purposely choose to decouple identity and expression and solve for each one independently so as not to enforce explicit correlation between local identity and expression deformation. To accomplish this we can define the deformation subspace in two different ways, where $\boldsymbol{B}_i$ is as described next.

Local Expression Model

Given a particular expression k, we factor the local multilinear models to obtain a set of linear models, which we refer to as local expression models. The deformation space of these models spans the set of identities. Specifically, for expression k and each patch i we set

$$\boldsymbol{B}_i = \mathcal{C}_i \times_1 \boldsymbol{U}_{vertices_i} \times_2 \boldsymbol{U}_{identities_i} \times_3 \boldsymbol{u}^k_{exp_i}. \tag{3.8}$$

Optimizing Local Expression Models. The main idea is that we can use the sparse vertices V^c computed from the global model as constraints and then we can optimize the local expression models to obtain per-patch model parameters M_i and $\{\alpha_i^1, ..., \alpha_i^{n_{id}}\}$. Finally, we combine the patches to obtain a global face mesh with the desired expression. The details of the optimization procedure and mesh reconstruction are given in Section 3.3.3.

Sampling Dense Expression Deformation. The resulting mesh contains the desired local expression details with the desired coarse deformation, however it will not retain the local identity details of the target subject $\boldsymbol{t}$. Therefore, we proceed to fit and solve a local identity model as described next, however this time densely sampling the estimated expression mesh to create a dense set of vertex constraints, which allows to preserve the local expression deformations as closely as possible while solving for the identity.

Local Identity Model

We obtain the identity of the target subject by building a set of local identity models, one per patch, analogous to the local expression models but this time spanning the space of expressions for the desired identity.

Local Model Fitting. This is accomplished by solving for the local identity coefficients that best fit the local shape $\boldsymbol{t}_i$ of $\boldsymbol{t}$ corresponding to patch i. Specifically, we select the neutral expression coefficients $\boldsymbol{u}_{exp_i}^{neutral}$ from $\boldsymbol{U}_{expressions_i}$ and solve for $\boldsymbol{u}_{id_i}^{\boldsymbol{t}}$ in a least-squares sense:

$$\boldsymbol{t}_i = (\mathcal{C}_i \times_1 \boldsymbol{U}_{vertices_i} \times_3 \boldsymbol{u}_{exp_i}^{neutral}) \times_2 \boldsymbol{u}_{id_i}^{\boldsymbol{t}}, \tag{3.9}$$

Once obtaining $\boldsymbol{u}_{id_i}^{\boldsymbol{t}}$, we can then form the expression patch shapes, which becomes the new deformation subspace $\boldsymbol{B}_i$

$$\boldsymbol{B}_i = \mathcal{C}_i \times_1 \boldsymbol{U}_{vertices_i} \times_2 \boldsymbol{u}_{id_i}^{\boldsymbol{t}} \times_3 \boldsymbol{U}_{expressions_i}. \tag{3.10}$$

Optimizing Local Identity Models. Using the dense vertex positions from the local expression solve as constraints, we optimize the local identity models to obtain final per-patch model parameters M_i and $\{\alpha_i^1, ..., \alpha_i^{n_{exp}}\}$, this time corresponding to the local identity models, and combine the patches to a global face mesh that forms the final synthesized expression k for target $\boldsymbol{t}$ (again, refer to Section 3.3.3 for optimization and mesh reconstruction details).

3.3.3 Local Model Optimization and Reconstruction

In the problem formulation for both the local expression model and the local identity model above, we must solve for the model parameters including the rigid local patch motion $\{M_i\}_{i=1}^{n_p}$, and the local blend coefficients $\{\boldsymbol{a}_i\}_{i=1}^{n_p}$ for the n_p patches. We denote $\boldsymbol{x}$ as our result mesh. We formulate the solution as an energy minimization problem.

$$\underset{\{M_i\}_{i=1}^{n_p}, \{\boldsymbol{a}_i\}_{i=1}^{n_p}}{\text{minimize}} \quad E_P + E_O + E_C, \tag{3.11}$$

where E_p is the *position* constraint (either the sparse or dense vertex positions defined above), E_o is the *overlap* constraint and E_c is the *subspace consistency* constraint.

Position Constraint

For the constrained subset of vertices V^c, obtained e.g. from the sparse sampling of the coarse deformation or the dense sampling of the expression deformation Eq. 3.5, we formulate the position constraint as:

$$E_P = \sum_{\boldsymbol{v}_j^c \in V^c} \sum_{i \in \Omega(\boldsymbol{v}_j^c)} \|\boldsymbol{v}_j^c - \boldsymbol{x}_i(\boldsymbol{v}_j^c)\|^2, \tag{3.12}$$

where $\boldsymbol{v}_j^c$ denotes the positional constraint, $\Omega(\boldsymbol{v}_j^c)$ is the set of patches which contain vertex $\boldsymbol{v}_j^c$ and $\boldsymbol{x}_i(\boldsymbol{v}_j^c)$ refers to the corresponding vertex position of $\boldsymbol{v}_j^c$ within patch $\boldsymbol{x}_i$ from Eq. 3.7.

Overlap Constraint

We employ an overlap constraint as defined in [Wu+16], which functions as spatial regularizer as it encourages neighbouring patches to take on similar shapes in the overlapping area, defined as:

$$E_O = \lambda_O \sum_{\boldsymbol{v} \in S} \sum_{(i,j) \in \Omega(\boldsymbol{v}), i>j} \|\boldsymbol{x}_i(\boldsymbol{v}) - \boldsymbol{x}_j(\boldsymbol{v})\|^2, \tag{3.13}$$

where S is the set of vertices shared by patches, $\Omega(\boldsymbol{v})$ is the set of patches that contain vertex $\boldsymbol{v}$, $\boldsymbol{x}_i(\boldsymbol{v})$ is the 3D position of vertex $\boldsymbol{v}$ in patch i and $\lambda_O (= 0.1)$ is a weighting factor.

Subspace Consistency Constraint

In addition to the overlap constraint, we also add a subspace consistency term for neighbouring patches, incentivizing them to take on similar coefficients of deformation.

$$E_C = \lambda_C \sum_{\boldsymbol{v} \in S} \sum_{(i,j) \in \Omega(\boldsymbol{v}), i>j} \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j\|^2, \tag{3.14}$$

where S and $\Omega(\boldsymbol{v})$ are the same as the overlap constraint, and $\boldsymbol{\alpha}_i$ corresponds to the deformation coefficients $\{\alpha_i^f\}_{f=1}^F$ of patch i and $\lambda_C(= 0.3)$ is a weighting factor.

Mesh Reconstruction

Once the model parameters are solved, all individual patches can be posed to form a global face shape, and we follow the approach of [Wu+16] to integrate the patches into a coherent shape.

As illustrated in Fig. 3.4, our global-local synthesis method is able to produce plausible facial expressions with realistic local details that preserve the input identity, unlike the global multilinear model, which exhibits artifacts and looses the identity.

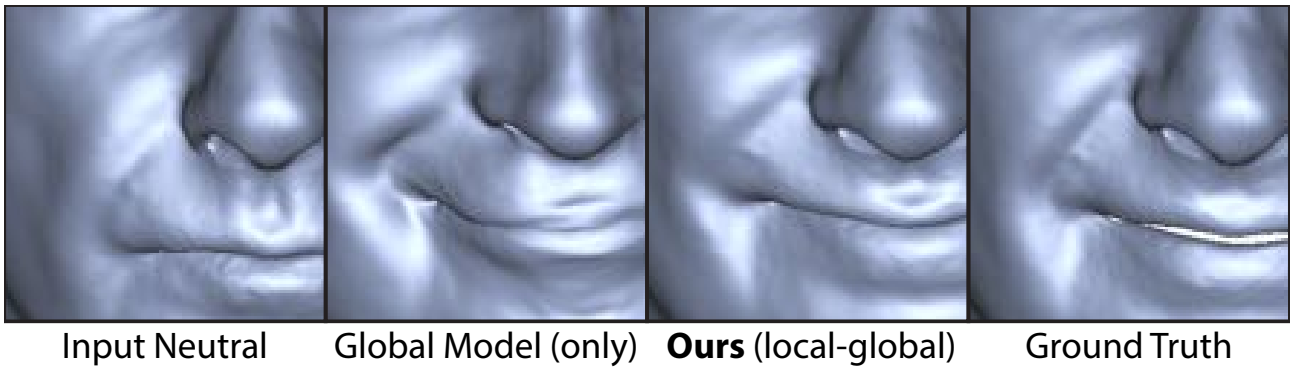


Figure 3.4: **Local Details -** Given the input neutral shape, the global multilinear model contains artifacts when predicting a smile expression. Note the local details of the nose are incorrect, and the nasolabial fold is very different from the ground truth expression. Our global-local framework achieves much more plausible results.

3.4 Experiments

Figure 3.5: We show the results of our method on [Che+18c] and compare it against the baselines [VT02a; SP04]. While [VT02a] produces unnatural expressions and [SP04] exhibits artefacts not specific to the target identity, our proposed method can robustly generate a plausible expression for the target identity. Our results are qualitatively and quantitatively closer to the ground truth.

Figure 3.6: We show the results of our method and compare it against the baselines [VT02a; SP04]. We note that the baselines produce unnatural expressions or expressions with artefacts while our results look more natural and plausible for the target identity.

3.4.1 Qualitative Evaluation

We compare the results of our method against two baselines. One is the global multilinear model [VT02a] and another is deformation transfer [SP04]. For deformation transfer, we use the results of [VT02a] as reference expression to transfer the deformations from.

We show the results of our method on two datasets: our own dataset containing high-quality facial meshes of 95k vertices and 4DFAB [Che+18c], a dataset with meshes of 53k vertices. 4DFAB contains 6 expressions and 110 different subjects. There is no neutral expression so we use the anger expression as our input meshes. We train our method on 100 subjects and test it on the remaining 10 subjects. Fig. 3.5 shows how our method compares against the baselines [VT02a; SP04] on 4DFAB. Our results are noticeably less noisy, preserve the identity and exhibit plausible expressions. Fig. 3.6 shows how our proposed method outperforms the baselines [VT02a; SP04] on our own dataset, where we used 50 subjects for training. We further show in Fig. 3.7 and Fig. 3.8 that we can add displacement map onto our method to achieve more realistic expression synthesis given a single neutral mesh of the target person. Displacement mapping uses the texture map to displace points on a mesh along the local surface normal. A displacement map can be estimated from the input neutral face mesh and would add very fine-grained details such as pores to the result mesh. The given neutral mesh has been highlighted.

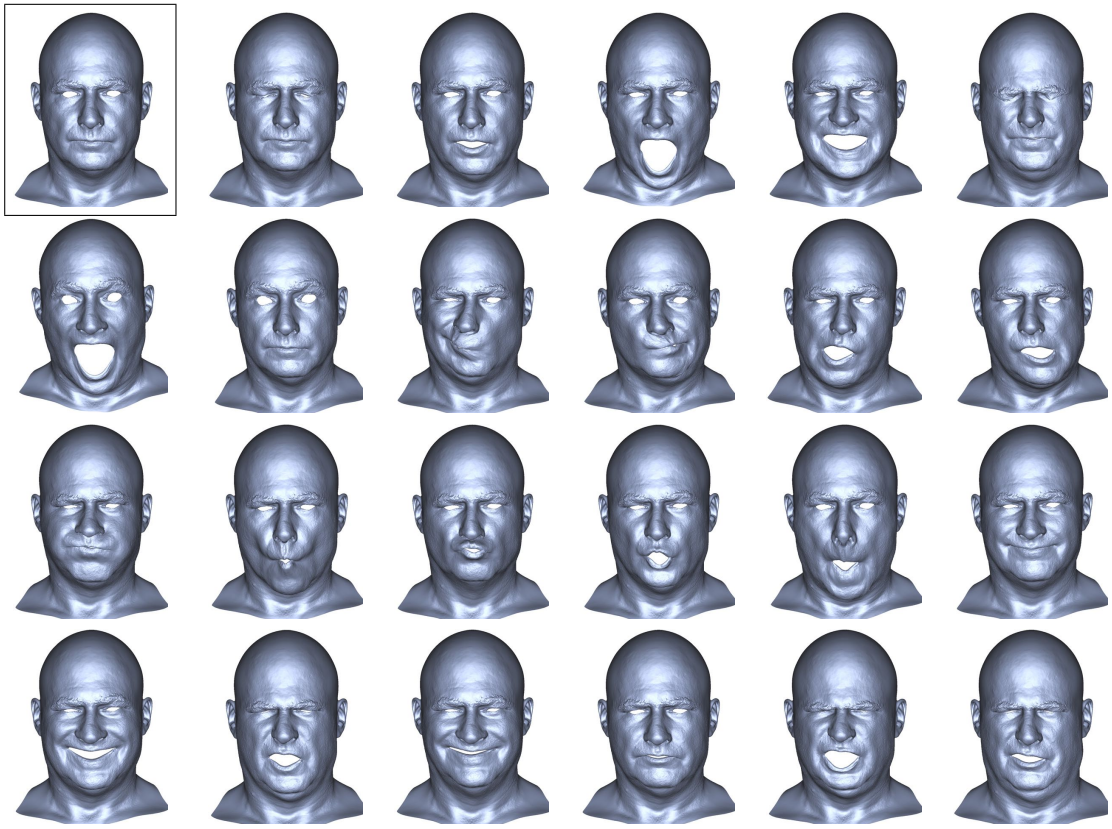


Figure 3.7: Person A with Displacement Map. The highlighted mesh is the input neutral face mesh. From the texture map and the normal map associated with this mesh, a displacement map can be estimated which would simulate very fine details on the face. Note how the fine hairs of the eyebrows are visible.

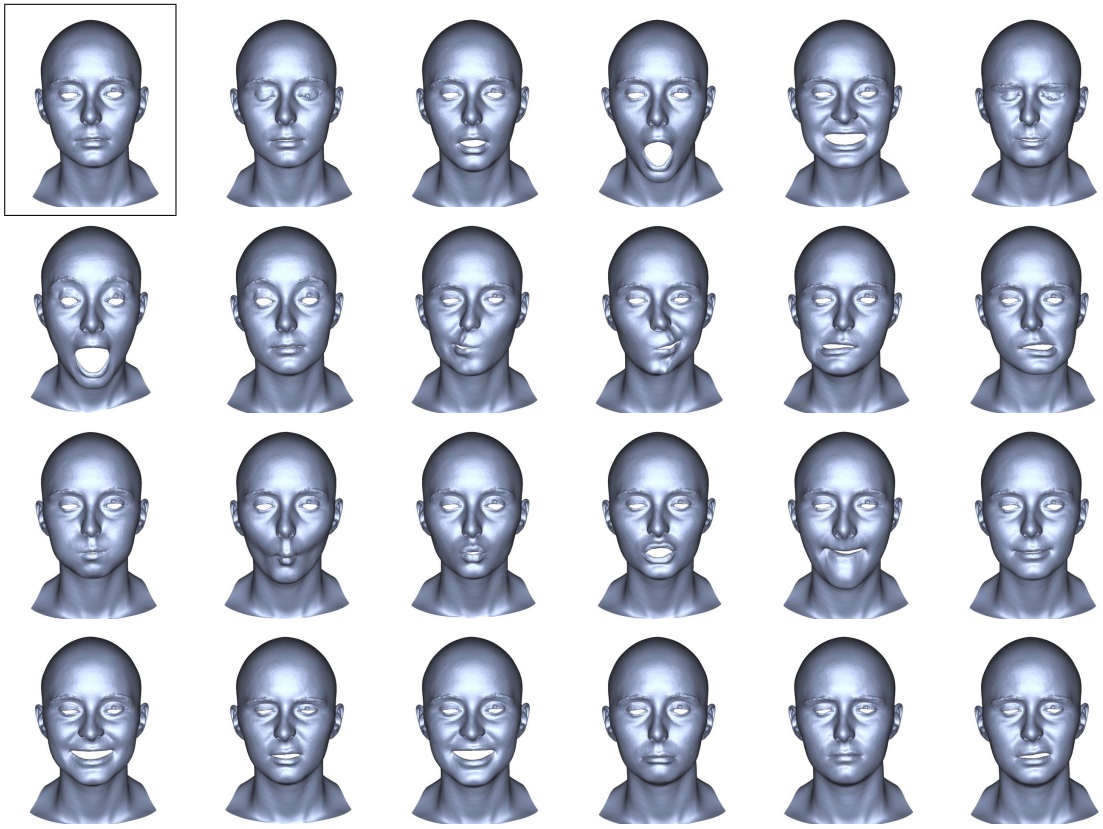


Figure 3.8: Person B with Displacement Map. The highlighted mesh is the input neutral face mesh. From the texture map and the normal map associated with this mesh, a displacement map can be estimated which would simulate very fine details on the face. Note how the fine hairs of the eyebrows are visible.

3.4.2 Ablation Study

Fig. 3.9 shows the results of our ablation study, assessing the impact of the different parts of the model. Method 3 represents using a local multilinear model (similar to the one proposed in [BBW14]) to predict expressions for the target person. We observe that despite conserving fine-scale details, this method does not synthesise a natural-looking facial expression. Our proposed method is method 4 which outperforms all other combinations in this ablation study. The synthesised facial expression maintain fine-scale details and look realistic and natural at the same time.

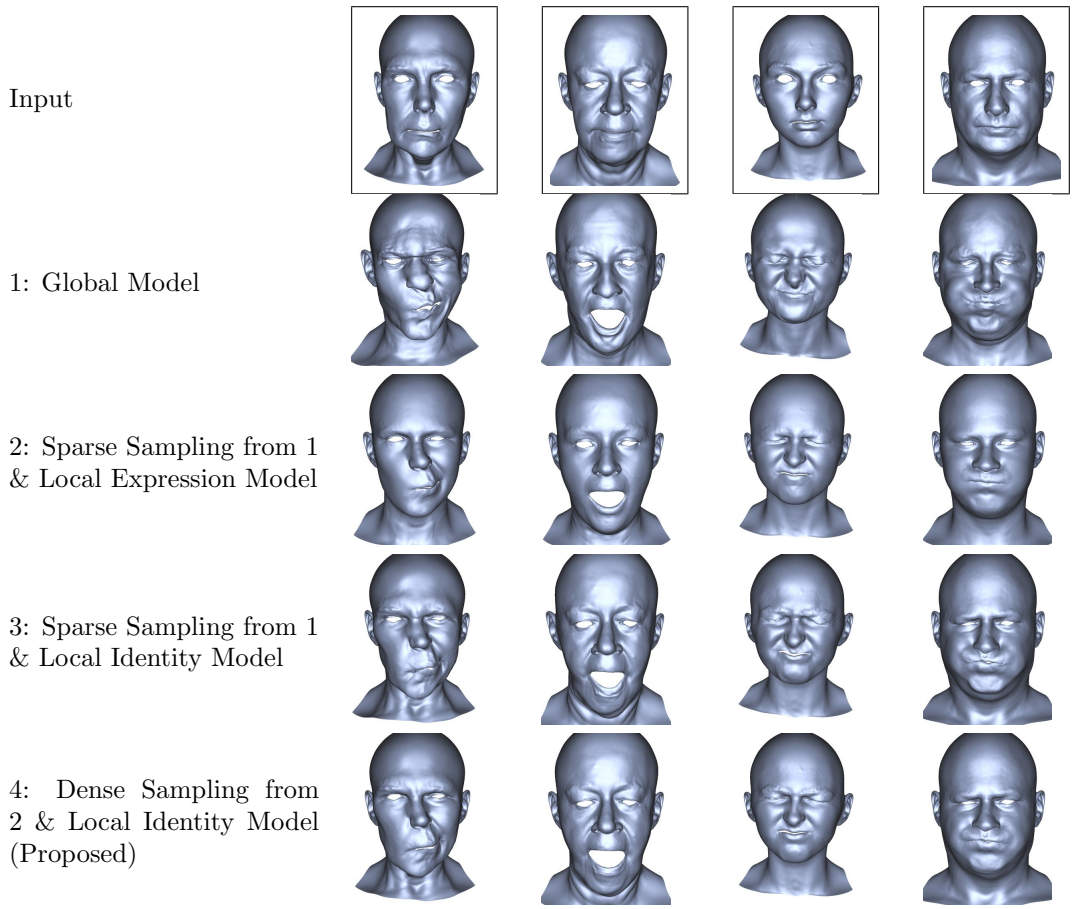


Figure 3.9: Ablation Study: Our proposed method returns plausible expressions with the minimum amount of artefacts while maintaining the identity.

3.4.3 Quantitative Evaluation

In order to quantitatively compare to existing methods, we predict 19 different expressions for 30 new test identities. We have removed high-level expressions such as sadness, happiness, and anger from the comparison since these are performed highly subjective and are hence not suited for quantitative analysis. For the 540 test meshes, we have ground truth results from the dataset. We compare our proposed methods against the multilinear model [VT02a] and deformation transfer [SP04] on 2 different metrics: 3D point position error in Fig. 3.10 and angular error of the normals in Fig. 3.11. Table 3.1 summarises the comparison. We observe that our proposed method outperforms the baselines in both metrics.

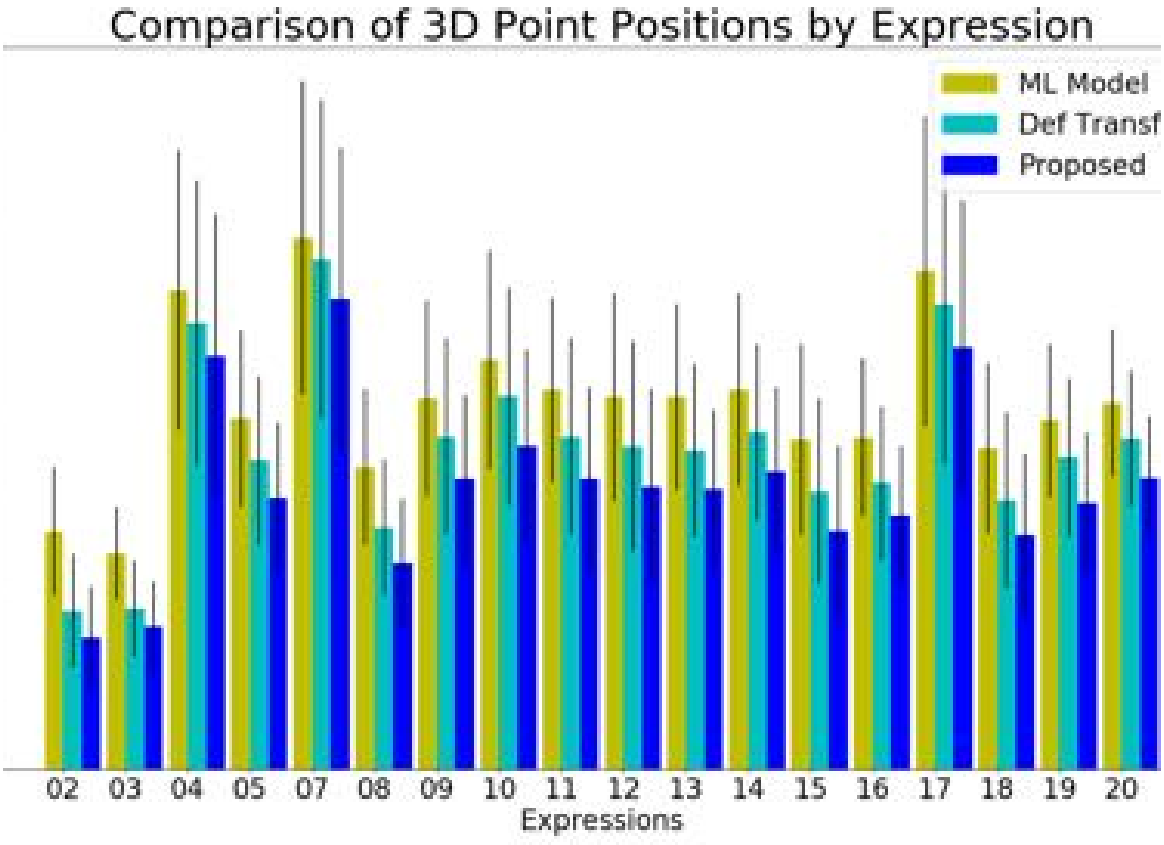


Figure 3.10: Comparison of our proposed method with multilinear model [VT02a] and deformation transfer [SP04] in terms of 3D position difference to the ground truth shape. Our proposed method consistently outperforms the baseline methods.

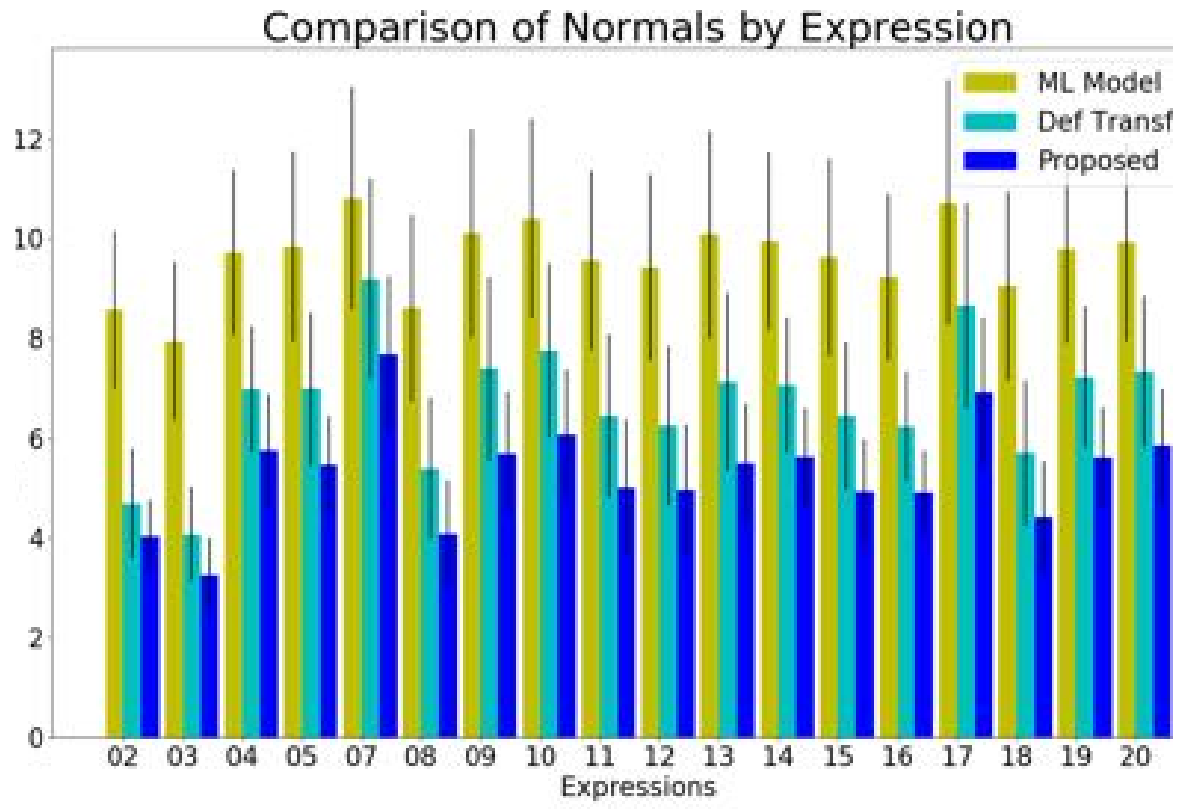


Figure 3.11: Comparison of our proposed method with multilinear model [VT02a] and deformation transfer [SP04] in terms of angular error between the normals of the predicted mesh and the normals of the ground truth shape. Our proposed method consistently outperforms the baseline methods.

Metric		[VT02a]	[SP04]	Ours
Position error on our data	[mm]	4.03	3.54	**3.12**
Position error on [Che+18c]	[mm]	2.32	2.23	**1.98**
Normal error on our data	[°]	9.63	6.72	**5.32**
Normal error on [Che+18c]	[°]	6.01	5.75	**4.75**

Table 3.1: Quantitative comparison reporting the mean position and mean normal angular error. Our method outperforms the baselines in terms of both position error and normal error on both datasets.

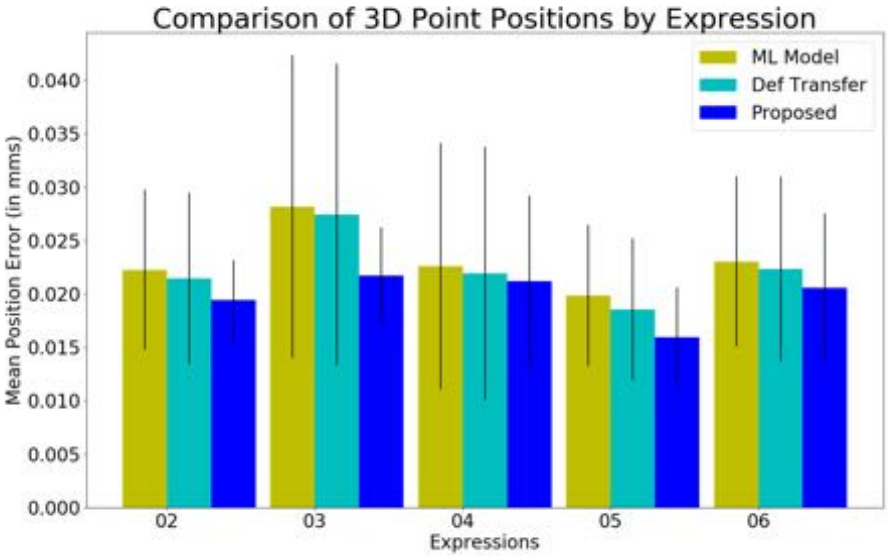


Figure 3.12: Comparison of our proposed method with multilinear model [VT02a] and deformation transfer [SP04] in terms of 3D position difference to the ground truth on [Che+18c]. Our proposed method consistently outperforms the baseline methods.

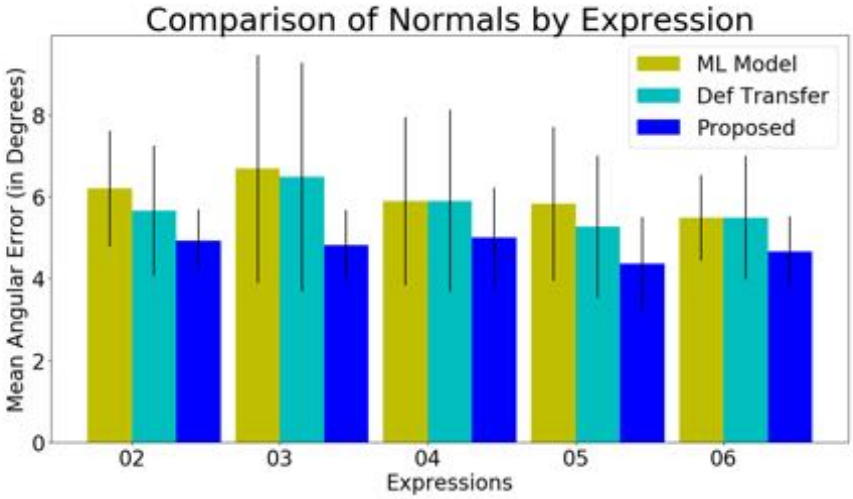


Figure 3.13: Comparison of our proposed method with multilinear model [VT02a] and deformation transfer [SP04] in terms of angular error between the normals of the predicted mesh and the normals of the ground truth on [Che+18c]. Our proposed method consistently outperforms the baseline methods.

3.4.4 Perceptual User Study

We conducted an anonymous user study where each user was presented with an image depicting a neutral expression of a random subject alongside the expressions generated by our proposed method and the two baseline methods [VT02a; SP04]. Then the user was asked to choose which among the three options is the most plausible expression for the person in the neutral image. Each user was asked to rate a total of 50 samples spanning 24 different expressions and 25 different identities. The results are reported in Table 3.2. Our proposed method was chosen 57.60% of the time, more than twice as often than any of the baseline methods. A total of 50 users participated in the user study, yielding 2429 selections. The results are statistically significant with $p < 10^{-4}$.

Method	Top Selection Rate
Ours	**57.60%**
[SP04]	20.95%
[VT02a]	21.45%

Table 3.2: Users tend to select shapes synthesised by the proposed method as their preferred result twice as likely than shapes synthesised by the baselines.

3.4.5 Data Augmentation

Our expression synthesis method can be used for data augmentation purposes. Many available 3D face datasets contain only faces in neutral expression. By registering the template mesh of the new dataset to the template mesh of our dataset, the trained global-local model can be applied directly to meshes from a new dataset. Even more variability may be achieved by bypassing the global model estimate and directly transferring the coarse motion from the individual subjects instead, allowing to synthesize a wide range of expression nuances as shown in Fig. 3.14, while still preserving the target identity.

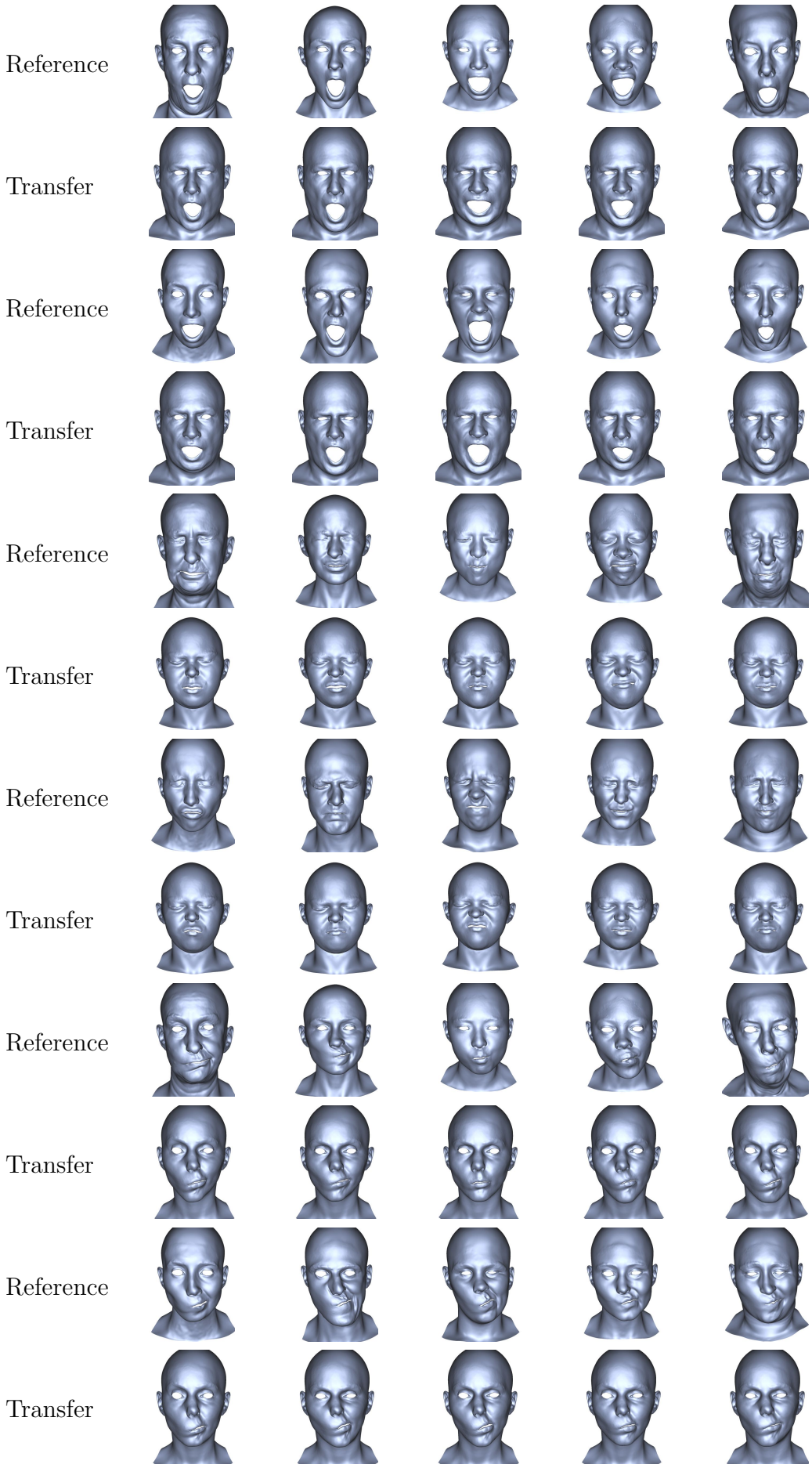


Figure 3.14: Extracting the coarse deformation from individual subjects instead of the global model allows to synthesize a wide range of expression nuances while preserving the target identity, which is very valuable for data augmentation purposes.

3.5 Conclusions

We present a method for synthesising realistic facial expressions given a target neutral face mesh. Leveraging a new global-local multilinear model, our method produces novel expressions that retain the target identity, contain plausible expression details at various scales, and are free from geometric artefacts. Our method both quantitatively and qualitatively outperforms current state-of-the-art expression synthesis algorithms, and can be used to generate high-quality facial rigs or augment existing multi-identity facial datasets.

Chapter 4

Unsupervised Tensor Decomposition for Learning the Multilinear Structure of Visual Data

4.1 Introduction

Statistical methods that explain variability among observed measurements (data) in terms of a potentially lower number of unobserved, latent, variables are cornerstones in data analysis, image and signal processing, and computer vision.

Factor analysis [FW11] and the closely related Principal Component Analysis (PCA) [Hot33] and Singular Value Decomposition (SVD) are probably the most popular statistical methods to find a single mode of variation that explains the data. Nevertheless, most forms of (visual) data have many different and possibly independent, modes of variations and hence methods such as the PCA are not able to identify them. Consider, for example, a population of faces with differing identities and expressions observed under different views (poses) where the appearance of each face is a result of some multifactor confluence due to identity, expression, and pose variation. In order to disentangle multiple but independent modes of variations, several multilinear

(tensor) decompositions have been employed [Tuc66; DDV00; KD80; KB09; Kru89]. For instance, the High Order SVD (HOSVD) [DDV00] is able to identify different modes of variation for identities, expressions, and poses per pixel, from a population of faces, by decomposing a carefully designed data tensor. This method is known as TensorFaces [VT02b].

In this chapter, we investigate the problem of disentangling the modes of variation in unlabelled and possibly incomplete data. In particular, we focus on sets of data that are incomplete in the sense that access to samples exhibiting every possible type of variation is not guaranteed. To this end, we propose the first *unsupervised multilinear decomposition* which uncovers the potential multilinear structure of incomplete sets of data and the corresponding low-dimensional latent variables (coefficients) explaining different types of variation. The proposed model is schematically summarised in Figure 4.1. In the depicted example, each image $\mathbf{x}_i$ is generated as tensor to vector product [KB09] of a tensor $\boldsymbol{\mathcal{B}}$ capturing the multilinear structure of the data and some coefficients corresponding to a meaningful variation. Here, $\mathbf{l}_i$ represents lighting coefficients, $\mathbf{e}_i$ expression coefficients and $\mathbf{c}_i$ identity coefficients. The number of differed types of variation is assumed to be known and specifies the order of the multilinear basis $\boldsymbol{\mathcal{B}}$.

The contributions are organised as follows:

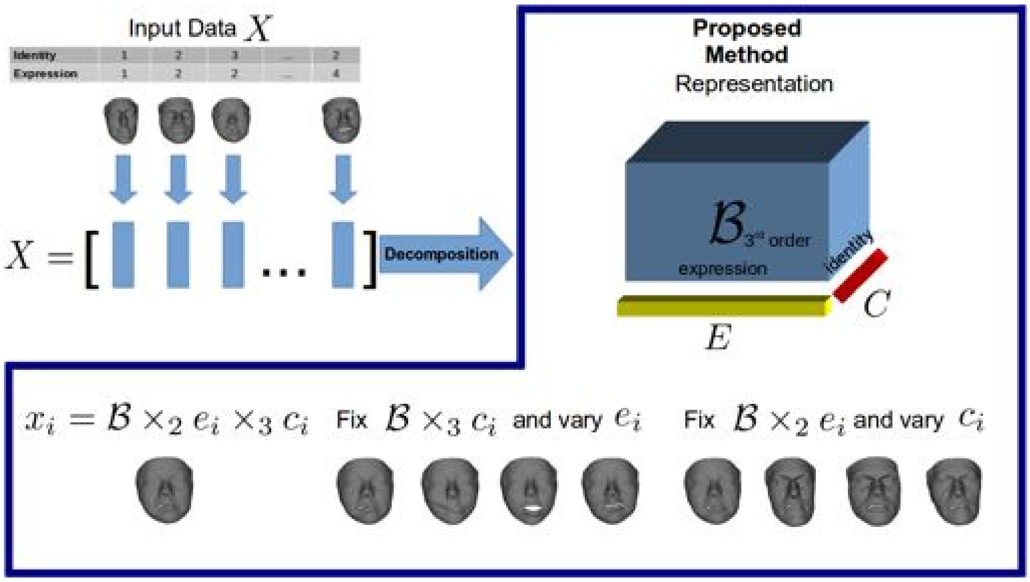
- A novel multilinear decomposition of matrices that recovers the multilinear structure and hence disentangles an arbitrary number of different modes of variation from possibly incomplete set of (visual) data is proposed in Section 4.2.1.
- To compute the proposed multilinear matrix decomposition, an efficient alternating least squares type of algorithm is developed in Section 4.2.1.
- The proposed method is extended to handle data contaminated by sparse noise of large magnitude and outliers in Section 4.2.2. To this end, a suitable ℓ_1 -norm regularised problem is solved allowing the estimation of different modes of variation in the presence of noise.
- A second variant of the proposed decomposition allowing the estimation of low-rank latent coefficients is introduced in Section 4.2.3. Latent coefficients with low-rank structure

naturally appear in applications such as video analysis where consecutive video frames are highly correlated

- In practice, partial information regarding labels or the geometry of a subset of modes of variation is available. To exploit such information a graph-regularised extension of the proposed decomposition is proposed in Section 4.2.4.

- To demonstrate the generality of the proposed models, in Section 4.3 extensive experiments on computer vision tasks are conducted including facial expression transfer, Shape from Shading (SfS), and estimation of surface normals directly from "in-the-wild" images. In the latter task, we demonstrate that by feeding the estimated normals from the proposed decomposition into a deep neural network, facial reconstruction can be achieved using a single non-aligned image captured in the wild. Furthermore, it is worth mentioning that the methods for SfS in [Kem13b; SPZ15] are only special cases of the proposed multilinear decomposition.

Example 3D Data



Example 2D Data

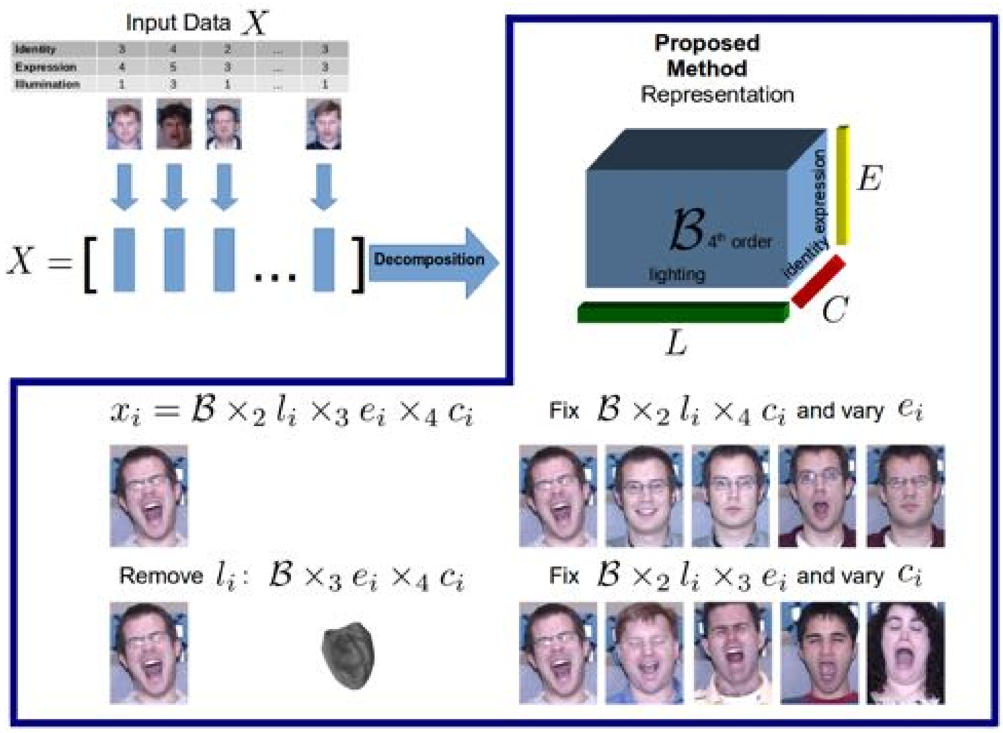


Figure 4.1: Visualisation of the unsupervised multilinear decomposition and its applications. A sample vector $\mathbf{x}_i$ is assumed to be generated by a common multilinear structure $\mathcal{B}$ and sample specific weights e.g. $\mathbf{l}_i$, $\mathbf{e}_i$ and $\mathbf{c}_i$. We assume the weights correspond to variations in the data ($\mathbf{l}_i$ to lighting, $\mathbf{e}_i$ to expression and $\mathbf{c}_i$ to identity). By varying $\mathbf{e}_i$ only, we expect to see changes in expression but no change in identity or lighting. Similarly, if we vary $\mathbf{c}_i$ only we expect the expression and lighting to remain the same but the identities to change. Additionally if we remove the lighting $\mathbf{l}_i$, we expect the remaining information to correspond to the 3D shape of the object.

4.2 Methodology

4.2.1 Basic Model

Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{d \times N}$ be a matrix of observations, where each of the N columns represent a vectorised image of d pixels. In order to discover $M-1$ different modes of variation we propose the following decomposition:

$$\boldsymbol{x}_i = \boldsymbol{\mathcal{B}} \times_2 \boldsymbol{a}_i^{(2)} \times_3 \boldsymbol{a}_i^{(3)} \cdots \times_m \boldsymbol{a}_i^{M} = \boldsymbol{\mathcal{B}} \prod_{m=2}^{M} \times_m \mathbf{a}_i^{(m)}, \tag{4.1}$$

where $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{d \times K_2 \times \cdots \times K_M}$ representing the common multilinear basis of $\boldsymbol{X}$ and the set of vectors $\{\boldsymbol{a}_i^{(m)} \in \mathbb{R}^{K_m}\}_{m=2}^{M}$ represents the variation coefficients in each mode specific to the vectorised image $\boldsymbol{x}_i$.

Therefore, for the observation matrix $\boldsymbol{X}$, and by exploiting the properties of multilinear operators e.g., [KB08], the above decomposition is written in matrix form as

$$\boldsymbol{X} = \boldsymbol{B}_{(\mathbf{1})}(\boldsymbol{A}^{(2)} \odot \boldsymbol{A}^{(3)} \cdots \odot \boldsymbol{A}^{(M)}) = \boldsymbol{B}_{(\mathbf{1})}\big(\bigodot_{m=2}^{M} \mathbf{A}^{(m)}\big), \tag{4.2}$$

where $\boldsymbol{B}_{(\mathbf{1})}\mathbb{R}^{d \times K_2 \cdot K_3 \cdot K_M}$ is the mode-1 matricisation of $\boldsymbol{\mathcal{B}}$ and $\{\boldsymbol{A}^{(m)}\}_{m=2}^{M} \in \mathbb{R}^{K_m \times N}$ gathers the variation coefficients for all images across $M-1$ modes of variation. Clearly, this formulation is different from the Tucker decomposition [Tuc66] and the HOSVD [DDV00].

To find the unknown multilinear basis $\boldsymbol{\mathcal{B}}$ and the variation coefficients $\{\boldsymbol{A}^{(m)}\}_{m=2}^{M}$, we propose to solve:

$$\begin{aligned} \underset{\boldsymbol{B}_{(\mathbf{1})}, \{\boldsymbol{A}^{(m)}\}_{m=2}^{M}}{\arg\min} \quad & \|\boldsymbol{X} - \boldsymbol{B}_{(\mathbf{1})}\big(\bigodot_{m=2}^{M} \mathbf{A}^{(m)}\big)\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{B}_{(\mathbf{1})}^T \boldsymbol{B}_{(\mathbf{1})} = \mathbf{I}. \end{aligned} \tag{4.3}$$

Optimisation problem (4.3) is non-convex. Therefore, we propose to solve (4.3) by employing an Alternating Least Squares (ALS) scheme, where each variable is updated in an alternating fashion. Let t denote the iteration index, given $\boldsymbol{B}_{(\mathbf{1})}[0]$ and $\{\boldsymbol{A}^{(m)}[0]\}_{m=2}^{M}$, the iteration of the ALS solver reads as follows:

$$\begin{aligned} \boldsymbol{B}_{(1)}[t+1] = \underset{\boldsymbol{B}_{(1)}}{\arg\min} \, \|\boldsymbol{X} - \boldsymbol{B}_{(1)}\big(\bigodot_{m=2}^{M} \mathbf{A}^{(m)}[t]\big)\|_F^2 \\ \text{s.t. } \boldsymbol{B}_{(1)}^T \boldsymbol{B}_{(1)} = \mathbf{I}. \end{aligned} \tag{4.4}$$

$$\begin{aligned} \{\boldsymbol{A}^{(m)}[t+1]\}_{m=2}^{M} = \\ \underset{\{\boldsymbol{A}^{(m)}\}_{m=2}^{M}}{\arg\min} \, \|\boldsymbol{X} - \boldsymbol{B}_{(1)}[t+1]\big(\bigodot_{m=2}^{M} \mathbf{A}^{(m)}\big)\|_F^2 \end{aligned} \tag{4.5}$$

Solving (4.4): Problem (4.4) is an orthogonal Procrustes problem, whose solution is given by [GD04]: $\boldsymbol{B}_{(1)}[t+1] = \boldsymbol{U}\boldsymbol{V}^T$, where $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \boldsymbol{X}\big(\bigodot_{m=2}^{M} \mathbf{A}^{(m)}[t]\big)^T$ is the SVD.

Solving (4.5): Due to the unitary invariance of the Frobenius norm (4.4) is equivalent to

$$\underset{\{\boldsymbol{A}^{(m)}\}_{m=2}^{M}}{\arg\min} \, \|\boldsymbol{B}_{(1)}[t+1]^T\boldsymbol{X} - \bigodot_{m=2}^{M} \mathbf{A}^{(m)}\|_F^2, \tag{4.6}$$

which is a Khatri-Rao factorisation problem [RH10]. Let $\boldsymbol{Q} = \boldsymbol{B}_{(1)}[t+1]^T\boldsymbol{X} \in \mathbb{R}^{K_2 \cdot K_3 \cdots K_M \times N}$, then each column of Q is written as:

$$\boldsymbol{q}_i = \bigodot_{m=2}^{M} \mathbf{a}_i^{(m)} \tag{4.7}$$

Let us partition $\boldsymbol{q}_i$ into a set $S = K_{M-1} \cdot K_{M-2} \cdots K_2$ vectors $\{\boldsymbol{q}_i^{B_b} \in \mathbb{R}^{K_M}\}_{b=1}^{K_{M-1} \cdot K_{M-2} \cdots K_2}$ such that $\boldsymbol{q}_i = [\boldsymbol{q}_i^{B_1 T} \boldsymbol{q}_i^{B_2 T} \cdots \boldsymbol{q}_i^{B_S T}]^T$. This partitioning enables us to rearrange the elements of $\boldsymbol{q}_i$ into a tensor $\boldsymbol{\mathcal{Q}}_i \in \mathbb{R}^{K_M \times K_{M-1} \times \cdots \times K_2}$ such that $\boldsymbol{Q}_{\boldsymbol{i}(\mathbf{1})} = [\boldsymbol{q}_i^{B_1}, \boldsymbol{q}_i^{B_2}, \cdots, \boldsymbol{q}_i^{B_{K_{M-1} \cdot K_{M-2} \cdots K_2}}] \in \mathbb{R}^{K_M \times (K_{M-1} \cdot K_{M-2} \cdots K_2)}$. Therefore, based on (4.7), $\boldsymbol{\mathcal{Q}}_i$ is written as

$$\boldsymbol{\mathcal{Q}}_i = \mathbf{a}_i^{(M)} \circ \mathbf{a}_i^{(M-1)} \circ \cdots \circ \mathbf{a}_i^{(2)} \tag{4.8}$$

Equation (4.8) indicates that we can recover the set of vectors $\{\mathbf{a}_i^{(m)}\}_{m=2}^{M}$ and therefore the set of matrices $\{\mathbf{A}^{(m)}\}_{m=2}^{M}$, by seeking a best (in the least squares sense) rank-1 approximation of $\boldsymbol{\mathcal{Q}}_i$, for $i = 1, 2, \ldots, N$. An efficient way to find the best rank-1 approximation of $\boldsymbol{\mathcal{Q}}_i$ is to exploit the truncated HOSVD [DDV00]. That is,

Algorithm 1 Multilinear Data Decomposition Algorithm

Input: Data Matrix $\boldsymbol{X} \in \mathbb{R}^{d \times N}$, dimensions K_2, K_3, $\dots K_M$ **Output:** $\boldsymbol{\mathcal{B}}$, $\boldsymbol{A}^{(2)}$, $\boldsymbol{A}^{(3)}$, $\dots \boldsymbol{A}^{(M)}$

Initialisation: $t \leftarrow 0$,
$[\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}] \leftarrow$SVD$(\boldsymbol{X})$,
$\boldsymbol{B_{(1)}}[0] = \boldsymbol{U}\sqrt{\boldsymbol{\Sigma}}$, $\boldsymbol{Q}[0] = \sqrt{\boldsymbol{\Sigma}}\boldsymbol{V}^T$
while not converged **do**
  for all image $i = 1 \dots N$ **do**
    construct $\boldsymbol{\mathcal{Q}}_i \in \mathbb{R}^{K_M \times K_{M-1} \cdots \times K_2}$ from $\boldsymbol{q}_i[t]$
    $[\boldsymbol{S}_i, \boldsymbol{U}_i] \leftarrow$HOSVD$(\boldsymbol{\mathcal{Q}}_i)$
    for each mode m $= 2 \dots$ M **do**
      $\boldsymbol{a}_i^{(m)}[t+1] = (\boldsymbol{S}_i)_1^{\frac{1}{M-1}} \boldsymbol{U}_i^{(M-m+1)}$
    end for
  end for
  $[\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}] \leftarrow$SVD$(\boldsymbol{X}\big(\bigodot_{m=2}^{M} \mathbf{A}^{(m)}[t]\big)^T)$
  $\boldsymbol{B_{(1)}}[t+1] = \boldsymbol{U}\boldsymbol{V}^T$
  $\boldsymbol{Q}[t+1] = \boldsymbol{B}_{(1)}[t+1]^T \boldsymbol{X}$
  Check convergence condition:
  $\frac{\|\boldsymbol{X} - \boldsymbol{B_{(1)}}[t+1]\boldsymbol{Q}[t+1]\|_F^2}{\|\boldsymbol{X}\|_F^2} < \epsilon$
  $t \leftarrow t + 1$
end while
Tensorise $\boldsymbol{B_{(1)}}$ into $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{d \times K_2 \times \cdots \times K_M}$

$$\boldsymbol{\mathcal{Q}}_i = s \prod_{n=1}^{M-1} \times_n \mathbf{u}_i^{(n)}, \tag{4.9}$$

where $\{\mathbf{u}_i^{(n)} \in \mathbb{R}^{K_{M-n+1}}\}_{n=1}^{M-1}$ is the the set of the first higher order singular vector along $M-1$ modes of tensor $\boldsymbol{\mathcal{Q}}_i$ and $s = (\boldsymbol{\mathcal{S}})_{1,1,\dots,1}$ is the first high-order singular value stored as a first element in the core tensor $\boldsymbol{\mathcal{S}}$. Consequently, the columns of the variation coefficient matrices $\{\mathbf{A}^{(m)}\}_{m=2}^{M}$ can be estimated by

$$\boldsymbol{a}_i^{(m)} = s^{\frac{1}{M-1}} \mathbf{u}_i^{(M-m+1)}, \tag{4.10}$$

for $m = 2, 3, \dots, M$. Interestingly, the estimation of the variation coefficients according to (4.10) resolves the inherent scaling ambiguity in (4.6) by assigning the same Euclidean-norm to each column of $\mathbf{A}^{(m)}$. The procedure of solving (4.3) is summarised in Algorithm 1.

Remarks: In the special case of 2 modes and where $k_2 = 4$, (4.2) becomes:

$$\boldsymbol{X} = \boldsymbol{B_{(1)}}(\boldsymbol{L} \odot \boldsymbol{C}), \tag{4.11}$$

where $\boldsymbol{L} = \boldsymbol{A}_2 \in \mathbb{R}^{4\times n}$, $\boldsymbol{C} = \boldsymbol{A}_3 \in \mathbb{R}^{k\times n}$.

Let $\boldsymbol{P} = \boldsymbol{L} \odot \boldsymbol{C}$ then,

$$\boldsymbol{X} = \boldsymbol{B}_{(1)}\boldsymbol{P}. \tag{4.12}$$

Equation (4.12) corresponds to the formulation used by [Kem13b]. $\boldsymbol{P} = \boldsymbol{L}\odot\boldsymbol{C}$ has been implied by [Kem13b] but not explicitly formulated as such. Hence this shows that [Kem13b] represents a special case of our general decomposition.

4.2.2 Robust Decomposition

Equation (4.11) corresponds to the formulation used by [SPZ15] with the only difference being the separation of $\boldsymbol{X}$ into a low-rank part $\boldsymbol{BP}$ and sparse, non-Gaussian error $\boldsymbol{E}$. [SPZ15] used these to add robustness to the decomposition due to occlusion that is present in images captured in unconstrained conditions (also referred to as "in-the-wild"), as well as to account for the high frequency errors introduced by the coarse geometric alignment of the images. Generalising this to the case of arbitrary number of modes, a robust decomposition can be found by solving:

$$\begin{aligned} \underset{\boldsymbol{B}_{(1)},\{\boldsymbol{A}^{(m)}\}_{m=2}^{M}}{\arg\min} \quad & \|\boldsymbol{P}\|_* + \lambda\|\boldsymbol{E}\|_1 \\ \text{s.t.} \quad & \boldsymbol{X} = \boldsymbol{B}_{(1)}\boldsymbol{P} + \boldsymbol{E}, \\ & \boldsymbol{P} = \big(\bigodot_{m=2}^{M} \mathbf{A}^{(m)}\big), \\ & \boldsymbol{B}_{(1)}^T\boldsymbol{B}_{(1)} = \mathbf{I}. \end{aligned} \tag{4.13}$$

The nuclear norm and the l_1-norm promote low-rank and sparsity respectively. To solve (4.15), the Alternating Directions Method (ADM) [Ber82] is employed. To this end, the following augmented Lagrangian function should be minimised:

$$\begin{aligned}\mathcal{L}(\boldsymbol{P},\boldsymbol{E},\boldsymbol{B_{(1)}},\{\boldsymbol{A}^{(m)}\}_{m=2}^{M},\boldsymbol{\Lambda}_1,\boldsymbol{\Lambda}_2) = \\ \|\boldsymbol{P}\|_* + \lambda\|\boldsymbol{E}\|_1 + \frac{\mu}{2}\|\boldsymbol{X} - \boldsymbol{B_{(1)}}\boldsymbol{P} - \boldsymbol{E} + \frac{\boldsymbol{\Lambda}_1}{\mu}\|_F^2 + \\ \frac{\mu}{2}\|\boldsymbol{P} - \big(\bigodot_{m=2}^{M}\mathbf{A}^{(m)}\big) + \frac{\boldsymbol{\Lambda}_2}{\mu}\|_F^2,\end{aligned} \tag{4.14}$$

with respect to $\boldsymbol{B}_{\boldsymbol{(1)}}^T\boldsymbol{B_{(1)}} = \mathbf{I}$. $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$ denote the Lagrangian multipliers.

A robust decomposition can be found by solving:

$$\begin{aligned}\underset{\boldsymbol{B_{(1)}},\{\boldsymbol{A}^{(m)}\}_{m=2}^{M}}{\arg\min} \quad & \|\boldsymbol{P}\|_* + \lambda\|\boldsymbol{E}\|_1 \\ \text{s.t.} \quad & \boldsymbol{X} = \boldsymbol{B_{(1)}}\boldsymbol{P} + \boldsymbol{E}, \\ & \boldsymbol{P} = \big(\bigodot_{m=2}^{M}\mathbf{A}^{(m)}\big), \\ & \boldsymbol{B}_{\boldsymbol{(1)}}^T\boldsymbol{B_{(1)}} = \mathbf{I}.\end{aligned} \tag{4.15}$$

The nuclear norm and the l_1-norm promote low-rank and sparsity respectively. To solve (4.15), the Alternating Directions Method (ADM) [Ber82] is employed. To this end, the following augmented Lagrangian function should be minimised:

$$\begin{aligned}\mathcal{L}(\boldsymbol{P},\boldsymbol{E},\boldsymbol{B_{(1)}},\{\boldsymbol{A}^{(m)}\}_{m=2}^{M},\boldsymbol{\Lambda}_1,\boldsymbol{\Lambda}_2) = \\ \|\boldsymbol{P}\|_* + \lambda\|\boldsymbol{E}\|_1 + \frac{\mu}{2}\|\boldsymbol{X} - \boldsymbol{B_{(1)}}\boldsymbol{P} - \boldsymbol{E} + \frac{\boldsymbol{\Lambda}_1}{\mu}\|_F^2 + \\ \frac{\mu}{2}\|\boldsymbol{P} - \big(\bigodot_{m=2}^{M}\mathbf{A}^{(m)}\big) + \frac{\boldsymbol{\Lambda}_2}{\mu}\|_F^2,\end{aligned} \tag{4.16}$$

with respect to $\boldsymbol{B}_{\boldsymbol{(1)}}^T\boldsymbol{B_{(1)}} = \mathbf{I}$. $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$ denote the Lagrangian multipliers. Let t denote the iteration index. Given $\boldsymbol{P}[t]$, $\boldsymbol{B_{(1)}}[t]$, $\boldsymbol{E}[t]$, $\boldsymbol{A}^{(m)}[t]$, $\boldsymbol{\Lambda}_1[t]$, $\boldsymbol{\Lambda}_2[t]$ and $\mu[t]$, the iteration of ADM for Equation (4.15) reads:

$$\begin{aligned}\boldsymbol{P}[t+1] &= \arg\min_{\boldsymbol{P}[t]} \mathcal{L}(\boldsymbol{P}[t], \boldsymbol{E}[t], \boldsymbol{B_{(1)}}[t], \{\boldsymbol{A}^{(m)}[t]\}_{m=2}^{M}, \boldsymbol{\Lambda}_1[t]) \\ &= \|\boldsymbol{P}[t]\|_* + \frac{\mu}{2}\Big(\|\boldsymbol{P}[t] - (\bigodot_{m=2}^{M} \mathbf{A}^{(m)}[t]) + \frac{\boldsymbol{\Lambda}_2[t]}{\mu}\|_F^2 \\ &+ \|\boldsymbol{X} - \boldsymbol{B_{(1)}}[t]\boldsymbol{P}[t] - \boldsymbol{E}[t] + \frac{\boldsymbol{\Lambda}[t]}{\mu}\|_F^2\Big),\end{aligned} \tag{4.17}$$

$$\begin{aligned}\boldsymbol{E}[t+1] = \arg\min_{\boldsymbol{E}[t]} &\lambda\|\boldsymbol{E}[t]\|_1 \\ &+ \frac{\mu}{2}\|\boldsymbol{X} - \boldsymbol{B_{(1)}}[t]\boldsymbol{P}[t] - \boldsymbol{E}[t] + \frac{\boldsymbol{\Lambda}[t]}{\mu}\|_F^2,\end{aligned} \tag{4.18}$$

$$\begin{aligned}\boldsymbol{B_{(1)}}[t+1] = \arg\min_{\boldsymbol{B_{(1)}}[t]} &\frac{\mu}{2}\|\boldsymbol{X} - \boldsymbol{B_{(1)}}[t]\boldsymbol{P}[t] - \boldsymbol{E}[t] + \frac{\boldsymbol{\Lambda}[t]}{\mu}\|_F^2, \\ \text{s.t. } &\boldsymbol{B}_{\boldsymbol{(1)}}^T\boldsymbol{B_{(1)}} = \mathbf{I}\end{aligned} \tag{4.19}$$

$$\{\boldsymbol{A}^{(m)}[t+1]\}_{m=2}^{M} = \arg\min_{\boldsymbol{A^{(m)}[t]}} \frac{\mu}{2}\|\boldsymbol{P}[t] - (\bigodot_{m=2}^{M} \mathbf{A}^{(m)}[t]) + \frac{\boldsymbol{\Lambda}_2[t]}{\mu}\|_F^2, \tag{4.20}$$

In order to solve (4.17), we need to minimise (4.16) with respect to $\boldsymbol{P}$. Let $f(\boldsymbol{P})$ be the smooth term in (4.16) i.e., $f(\boldsymbol{P}) = \frac{\mu}{2}(\|\boldsymbol{P} - \big(\bigodot_{m=2}^{M} \mathbf{A}^{(m)}\big) + \frac{\boldsymbol{\Lambda}_2}{\mu}\|_F^2 + \|\boldsymbol{X} - \boldsymbol{B_{(1)}}\boldsymbol{P} - \boldsymbol{E} + \frac{\boldsymbol{\Lambda}}{\mu}\|_F^2)$. $f(\boldsymbol{P})$ is linearly approximated with respect to $\boldsymbol{P}$ at $\boldsymbol{P}[t]$ as:

$$f(\boldsymbol{P}) \approx f(\boldsymbol{P}[t]) + tr((\boldsymbol{P} - \boldsymbol{P}[t])^T \bigtriangledown f(\boldsymbol{P}[t])) + \frac{\mu}{2}\|\boldsymbol{P} - \boldsymbol{P}[t]\|_F^2, \tag{4.21}$$

where

$$\begin{aligned}\bigtriangledown f(\boldsymbol{P}[t]) = &-\mu\boldsymbol{B_{(1)}}[t]^T\big(\boldsymbol{X} - \boldsymbol{B_{(1)}}[t]\boldsymbol{P}[t] - \boldsymbol{E}[t] + \frac{\boldsymbol{\Lambda}[t]}{\mu}\big) \\ &+ \mu\big(\boldsymbol{P}[t] - (\bigodot_{m=2}^{M} \mathbf{A}^{(m)}[t]) + \frac{\boldsymbol{\Lambda}_2[t]}{\mu}\big).\end{aligned} \tag{4.22}$$

An approximate solution to (4.17) can therefore be obtained by minimising the linearised aug-

mented Lagrangian function:

$$\begin{aligned}\boldsymbol{P}[t+1] \approx & \arg\min_{\boldsymbol{P}} \|\boldsymbol{P}\|_* + f(\boldsymbol{P}[t]) \\ & + tr((\boldsymbol{P}-\boldsymbol{P}[t])^T \bigtriangledown f(\boldsymbol{P}[t])) + \frac{\mu[t]}{2}\|\boldsymbol{P}-\boldsymbol{P}[t]\|_F^2 \\ & = \arg\min_{\boldsymbol{P}} \|\boldsymbol{P}\|_* + \frac{\mu[t]}{2}\|\boldsymbol{P}-(\boldsymbol{P}[t]-\frac{1}{\mu}\bigtriangledown f(\boldsymbol{P}[t]))\|_F^2 \\ & = \mathcal{D}_{\mu^{-1}}\big[\boldsymbol{B_{(1)}}[t]^T\boldsymbol{X}-\boldsymbol{P}[t]-\boldsymbol{B_{(1)}}[t]^T\boldsymbol{E}+\frac{\boldsymbol{B_{(1)}}[t]^T\boldsymbol{\Lambda}_1[t]}{\mu} \\ & + (\bigodot_{m=2}^{M}\mathbf{A}^{(m)}[t]) - \frac{\boldsymbol{B_{(1)}}[t]^T\boldsymbol{\Lambda}_2[t]}{\mu}\big],\end{aligned} \tag{4.23}$$

where $\mathcal{D}_\tau[\boldsymbol{Q}] = \boldsymbol{U}\mathcal{S}_\tau\boldsymbol{V}^T$ is the singular value thresholding operator for any matrix $\boldsymbol{Q}$ with singular value decomposition (SVD) $\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. $\mathcal{S}_\tau[q] = \text{sgn}(q)\ \ \max(|q|-\tau, 0)$ is the shrinkage operator that applied element-wise can be extended to matrices [Can+11].

Subproblem (4.18) has a unique solution obtained via the shrinkage operator $\mathcal{S}_\tau[q]$. The solution of (4.18) is

$$\boldsymbol{E}[t+1] = \mathcal{S}_{\lambda\mu^{-1}}\big[\boldsymbol{X}-\boldsymbol{B_{(1)}}[t]\boldsymbol{P}[t+1]+\frac{\boldsymbol{\Lambda}_1[t]}{\mu}\big]. \tag{4.24}$$

Subproblem (4.19) is a reduced rank Procrustes Rotation problem [ZHT06]. The solution is $\boldsymbol{B_{(1)}}[t+1] = \boldsymbol{U}\boldsymbol{V}^T$ with

$$\big(\boldsymbol{X}-\boldsymbol{E}[t+1]+\frac{\boldsymbol{\Lambda}_1[t]}{\mu}\big)\boldsymbol{P}[t+1]^T = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \tag{4.25}$$

being the SVD of $\big(\boldsymbol{X}-\boldsymbol{E}[t+1]+\frac{\boldsymbol{\Lambda}_1[t]}{\mu}\big)\boldsymbol{P}[t+1]^T$.

Subproblem (4.20) is a Khatri-Rao factorisation problem [RH10] which can be solved in the same way as for (4.6). In this case $\boldsymbol{Q} = \boldsymbol{P}[t]+\frac{\boldsymbol{\Lambda}_2[t]}{\mu}$. The ADM for solving (4.15) is outlined in Algorithm 2.

Algorithm 2 Robust Multilinear Decomposition Algorithm

Input: Data Matrix $\boldsymbol{X} \in \mathbb{R}^{d \times N}$, dimensions K_2, K_3, $\dots K_M$ and parameter λ **Output:** $\boldsymbol{\mathcal{B}}$, $\boldsymbol{E}$, $\boldsymbol{P}$, $\boldsymbol{A}^{(2)}$, $\boldsymbol{A}^{(3)}$, $\dots \boldsymbol{A}^{(M)}$

Initialisation:
$t \leftarrow 0$,
$\boldsymbol{P}[0] = 0$, $\boldsymbol{E}[0] = 0$, $\boldsymbol{B_{(1)}}[0] = 0$, $\{\boldsymbol{A}^{(m)} = 0\}_{m=2}^{M}$,
$\boldsymbol{\Lambda_1}[0] = 0$, $\boldsymbol{\Lambda_2}[0] = 0$, $\mu = 10^{-6}$, $\rho = 1.1$, $\epsilon = 10^{-8}$
while not converged **do**
  Update $\boldsymbol{P}[t+1]$ by

$$\boldsymbol{P}[t+1] = \mathcal{D}_{\mu^{-1}}\big[\boldsymbol{B_{(1)}}[t]^T\boldsymbol{X} - \boldsymbol{P}[t] - \boldsymbol{B_{(1)}}[t]^T\boldsymbol{E} + \frac{\boldsymbol{B_{(1)}}[t]^T\boldsymbol{\Lambda}_1[t]}{\mu} + (\bigodot_{m=2}^{M}\mathbf{A}^{(m)}[t]) - \frac{\boldsymbol{B_{(1)}}[t]^T\boldsymbol{\Lambda}_2[t]}{\mu}\big]$$

  Update $\boldsymbol{E}[t+1]$ by

$$\boldsymbol{E}[t+1] = \mathcal{S}_{\lambda\mu^{-1}}\big[\boldsymbol{X} - \boldsymbol{B_{(1)}}[t]\boldsymbol{P}[t+1] + \frac{\boldsymbol{\Lambda}_1[t]}{\mu}\big]$$

  Update $\boldsymbol{B_{(1)}}[t+1]$ by

$$\big(\boldsymbol{X} - \boldsymbol{E}[t+1] + \frac{\boldsymbol{\Lambda}_1[t]}{\mu}\big)\boldsymbol{P}[t+1]^T = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T,$$
$$\boldsymbol{B_{(1)}}[t+1] = \boldsymbol{U}\boldsymbol{V}^T$$

  Set $\boldsymbol{Q} = \boldsymbol{P}[t+1] + \frac{\boldsymbol{\Lambda}_2[t]}{\mu}$
  for all image $i = 1 \dots N$ **do**
    construct $\boldsymbol{\mathcal{Q}}_i \in \mathbb{R}^{K_M \times K_{M-1} \cdots \times K_2}$ from $\boldsymbol{q}_i[t]$
    $[\boldsymbol{S}_i, \boldsymbol{U}_i] \leftarrow$HOSVD$(\boldsymbol{\mathcal{Q}}_i)$
    for each mode m $= 2 \dots$ M **do**
      $\boldsymbol{a}_i^{(m)}[t+1] = (\boldsymbol{S}_i)_1^{\frac{1}{M-1}}\boldsymbol{U}_i^{(M-m+1)}$
    end for
  end for
  Update Lagrange multipliers by

$$\boldsymbol{\Lambda_1}[t+1] = \boldsymbol{\Lambda_1}[t] + \mu(\boldsymbol{X} - \boldsymbol{B_{(1)}}[t+1]\boldsymbol{P}[t+1] - \boldsymbol{E}[t+1])$$
$$\boldsymbol{\Lambda_2}[t+1] = \boldsymbol{\Lambda_2}[t] + \mu(\boldsymbol{P}[t+1] - (\bigodot_{m=2}^{M}\mathbf{A}^{(m)}[t+1]))$$

  Update μ by $\mu = min(\rho\mu, 10^{-6})$
  Check convergence condition:

$$\frac{\|\boldsymbol{X} - \boldsymbol{B_{(1)}}[t+1]\boldsymbol{P}[t+1] - \boldsymbol{E}[t+1]\|_F^2}{\|\boldsymbol{X}\|_F^2} < \epsilon$$
$$\frac{\|\boldsymbol{P}[t+1] - (\bigodot_{m=2}^{M}\mathbf{A}^{(m)}[t+1])\|_F^2}{\|\boldsymbol{X}\|_F^2} < \epsilon$$

  $t \leftarrow t+1$
end while
Tensorise $\boldsymbol{B_{(1)}}$ into $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{d \times K_2 \times \cdots \times K_M}$

4.2.3 Rank-constrained Decomposition

We propose another variation to the problem formulated in (4.3) in order to incorporate additional low-rank constraints so that the methodology is suitable for image analysis. A sequence of images of a face from a single viewpoint, under varying illumination, can be nearby completely explained by 5 or 6 principal components [Ram02]. This justifies the addition of an additional low-rank constraint in the case of specific data such as videos of a single person under illumination change.

We propose to solve the following problem:

$$\begin{aligned} \underset{\boldsymbol{B_{(1)}},\{\boldsymbol{A}^{(m)}\}_{m=2}^{M}}{\arg\min} \quad & \{\|\boldsymbol{A}^{(m)}\|_*\}_{m=2}^{M} + \lambda\|\boldsymbol{E}\|_1 \\ \text{s.t.} \quad & \boldsymbol{X} = \boldsymbol{B_{(1)}}\big(\bigodot_{m=2}^{M}\mathbf{L}^{(m)}\big) + \boldsymbol{E}, \\ & \boldsymbol{L}^{(m)} = \boldsymbol{A}^{(m)}, \\ & \boldsymbol{B}_{\mathbf{(1)}}^{T}\boldsymbol{B_{(1)}} = \mathbf{I}. \end{aligned} \tag{4.26}$$

Similarly to (4.15), Problem (4.28) is solved by employing the ADM. That is, the following augmented Lagrangian function is minimised:

$$\begin{aligned} &\mathcal{L}(\{\boldsymbol{A}^{(m)}\}_{m=2}^{M}, \boldsymbol{E}, \boldsymbol{B_{(1)}}, \{\boldsymbol{\Lambda}^{(m)}\}_{m=1}^{M}) = \\ &\{\|\boldsymbol{A}^{(m)}\|_*\}_{m=2}^{M} + \lambda\|\boldsymbol{E}\|_1 + \\ &\frac{\mu}{2}\|\boldsymbol{X} - \boldsymbol{B_{(1)}}\big(\bigodot_{m=2}^{M}\mathbf{L}^{(m)}\big) - \boldsymbol{E} + \frac{\boldsymbol{\Lambda}_1}{\mu}\|_F^2 + \\ &\sum_{m=2}^{M}\frac{\mu}{2}\|\boldsymbol{L}^{(m)} - \boldsymbol{A}^{(m)} + \frac{\boldsymbol{\Lambda}_m}{\mu}\|_F^2, \end{aligned} \tag{4.27}$$

with respect to $\boldsymbol{B}_{\mathbf{(1)}}^{T}\boldsymbol{B_{(1)}} = \mathbf{I}$. $\boldsymbol{\Lambda}^{(m)}$ are the Lagrangian multipliers. Let t denote the iteration index.

We propose to solve the following problem:

$$\begin{aligned} \underset{\boldsymbol{B}_{(1)},\{\boldsymbol{A}^{(m)}\}_{m=2}^{M}}{\arg\min} \quad & \{\|\boldsymbol{A}^{(m)}\|_*\}_{m=2}^{M} + \lambda\|\boldsymbol{E}\|_1 \\ \text{s.t.} \quad & \boldsymbol{X} = \boldsymbol{B}_{(1)}\big(\bigodot_{m=2}^{M}\mathbf{L}^{(m)}\big) + \boldsymbol{E}, \\ & \boldsymbol{L}^{(m)} = \boldsymbol{A}^{(m)}, \\ & \boldsymbol{B}_{(1)}^T\boldsymbol{B}_{(1)} = \mathbf{I}. \end{aligned} \tag{4.28}$$

Similarly to (4.15), Problem (4.28) is solved by employing the ADM. That is, the following augmented Lagrangian function is minimised:

$$\begin{aligned} &\mathcal{L}(\{\boldsymbol{A}^{(m)}\}_{m=2}^{M}, \boldsymbol{E}, \boldsymbol{B}_{(1)}, \{\boldsymbol{\Lambda}^{(m)}\}_{m=1}^{M}) = \\ &\{\|\boldsymbol{A}^{(m)}\|_*\}_{m=2}^{M} + \lambda\|\boldsymbol{E}\|_1 + \\ &\frac{\mu}{2}\|\boldsymbol{X} - \boldsymbol{B}_{(1)}\big(\bigodot_{m=2}^{M}\mathbf{L}^{(m)}\big) - \boldsymbol{E} + \frac{\boldsymbol{\Lambda}_1}{\mu}\|_F^2 + \\ &\sum_{m=2}^{M}\frac{\mu}{2}\|\boldsymbol{L}^{(m)} - \boldsymbol{A}^{(m)} + \frac{\boldsymbol{\Lambda}_m}{\mu}\|_F^2, \end{aligned} \tag{4.29}$$

with respect to $\boldsymbol{B}_{(1)}^T\boldsymbol{B}_{(1)} = \mathbf{I}$. $\boldsymbol{\Lambda}^{(m)}$ are the Lagrangian multipliers. Let t denote the iteration index.

Given $\boldsymbol{A}^{(m)}[t]$, $\boldsymbol{B}_{(1)}[t]$, $\boldsymbol{E}[t]$, $\boldsymbol{L}^{(m)}[t]$, $\boldsymbol{\Lambda}_m[t]$ and $\mu[t]$, the iteration of ADM for Equation (4.28) reads:

$$\begin{aligned} \boldsymbol{A}^{(m)}[t+1] &= \underset{\boldsymbol{A}^{(m)}[t]}{\arg\min}\,\mathcal{L}\big(\{\boldsymbol{A}^{(m)}[t]\}_{m=2}^{M}, \boldsymbol{E}[t], \boldsymbol{B}_{(1)}[t], \\ &\qquad \{\boldsymbol{L}^{(m)}[t]\}_{m=2}^{M}, \{\boldsymbol{\Lambda}_m[t]\}_{m=1}^{M}\big) \\ &= \|\boldsymbol{A}^{(m)}[t]\|_* + \frac{\mu}{2}\|\boldsymbol{L}^{(m)}[t] - \boldsymbol{A}^{(m)}[t] + \frac{\boldsymbol{\Lambda}_m[t]}{\mu}\|_F^2, \end{aligned} \tag{4.30}$$

$$\begin{aligned}\boldsymbol{E}[t+1] = \arg\min_{\boldsymbol{E}[t]} \lambda\|\boldsymbol{E}[t]\|_1 \\ + \frac{\mu}{2}\|\boldsymbol{X} - \boldsymbol{B_{(1)}}[t]\big(\bigodot_{m=2}^{M}\mathbf{L}^{(m)}[t]\big) - \boldsymbol{E}[t] + \frac{\Lambda_1[t]}{\mu}\|_F^2,\end{aligned} \tag{4.31}$$

$$\begin{aligned}&\boldsymbol{L}^{(m)}[t+1] = \\ &\arg\min_{\boldsymbol{L}^{(m)}[t]} \frac{\mu}{2}\|\boldsymbol{X} - \boldsymbol{B_{(1)}}[t]\big(\bigodot_{m=2}^{M}\mathbf{L}^{(m)}[t]\big) - \boldsymbol{E}[t+1] + \frac{\Lambda_1[t]}{\mu}\|_F^2 \\ &+ \frac{\mu}{2}\|\boldsymbol{L}^{(m)}[t+1] - \boldsymbol{A}^{(m)}[t] + \frac{\Lambda_m[t]}{\mu}\|_F^2,\end{aligned} \tag{4.32}$$

$$\begin{aligned}&\boldsymbol{B_{(1)}}[t+1] = \\ &\arg\min_{\boldsymbol{B_{(1)}}[t]} \frac{\mu}{2}\|\boldsymbol{X} - \boldsymbol{B_{(1)}}[t]\big(\bigodot_{m=2}^{M}\mathbf{L}^{(m)}[t+1]\big) - \boldsymbol{E}[t+1] + \frac{\Lambda_1[t]}{\mu}\|_F^2, \\ &\text{s.t. } \boldsymbol{B}_{\boldsymbol{(1)}}^T\boldsymbol{B_{(1)}} = \mathbf{I}\end{aligned} \tag{4.33}$$

Subproblem (4.30) admits the following closed-form solution:

$$\boldsymbol{A}^{(m)}[t+1] = \mathcal{D}_{\mu^{-1}}(\boldsymbol{L}^{(m)}[t] + \frac{\boldsymbol{\Lambda}_m[t]}{\mu}), \tag{4.34}$$

where $\mathcal{D}_\tau[\boldsymbol{Q}] = \boldsymbol{U}\mathcal{S}_\tau\boldsymbol{V}^T$ is the singular value thresholding operator for any matrix $\boldsymbol{Q}$ with singular value decomposition (SVD) $\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. $\mathcal{S}_\tau[q] = sgn(q)\ max(|q| - \tau, 0)$ is the shrinkage operator that applied element-wise can be extended to matrices [Can+11].

(4.31) has an unique solution using the shrinkage operator $\mathcal{S}_\tau[q]$ as follows:

$$\boldsymbol{E}[t+1] = \mathcal{S}_{\frac{\lambda}{\mu}}(\boldsymbol{X} - \boldsymbol{B_{(1)}}[t]\big(\bigodot_{m=2}^{M}\mathbf{L}^{(m)}[t]\big) - \boldsymbol{E}[t] + \frac{\Lambda_1[t]}{\mu}), \tag{4.35}$$

(4.32) can be solved in the following: Let $\boldsymbol{O}_m = \boldsymbol{A}^{(m)} - \frac{\Lambda_m[t]}{\mu}$. Then rewriting (4.32) using (4.1) to solve per image gives:

$$\begin{aligned}\boldsymbol{L}^{(s)}[t+1] = \\ \underset{\boldsymbol{L}^{(m)}[t]}{\arg\min} \sum_{i=1}^{n} \frac{\mu}{2} \|\boldsymbol{x}_i - \boldsymbol{\mathcal{B}}[t] \prod_{m=2, m\neq s}^{M} \times_m \mathbf{l}_i^{(m)}[t] \times_s \mathbf{l}_i^{(s)}[t] \\ - \boldsymbol{e}_i[t+1] + \boldsymbol{\lambda}_i[t]\|_F^2 + \sum_{i=1}^{n} \frac{\mu}{2} \|\boldsymbol{l}_i[t] - \boldsymbol{o}_i[t]\|_F^2,\end{aligned} \tag{4.36}$$

So we need to solve (4.36) for each image i:

$$\begin{aligned}\boldsymbol{l}_i^{(s)}[t+1] = \\ \underset{\boldsymbol{l}_i^{(m)}[t]}{\arg\min} \frac{\mu}{2} \|\boldsymbol{x}_i - \boldsymbol{\mathcal{B}}[t] \prod_{m=2, m\neq s}^{M} \times_m \mathbf{l}_i^{(m)}[t] \times_s \mathbf{l}_i^{(s)}[t] \\ - \boldsymbol{e}_i[t+1] + \boldsymbol{\lambda}_i[t]\|_F^2 + \frac{\mu}{2} \|\boldsymbol{l}_i[t] - \boldsymbol{o}_i[t]\|_F^2,\end{aligned} \tag{4.37}$$

Let $f(\boldsymbol{l}_i^{(s)}[t]) = \frac{\mu}{2} \|\boldsymbol{x}_i - \boldsymbol{\mathcal{B}}[t] \prod_{m=2, m\neq s}^{M} \times_m \mathbf{l}_i^{(m)}[t] \times_s \mathbf{l}_i^{(s)}[t] - \boldsymbol{e}_i[t+1] + \boldsymbol{\lambda}_i[t]\|_F^2 + \frac{\mu}{2} \|\boldsymbol{l}_i[t] - \boldsymbol{o}_i[t]\|_F^2$. We derive $f(\boldsymbol{l}_i^{(s)}[t])$ to find the minimum. Let $\boldsymbol{B'} = \boldsymbol{\mathcal{B}}[t] \prod_{m=2, m\neq s}^{M} \times_m \mathbf{l}_i^{(m)}[t]$:

$$\begin{aligned} & \frac{df(\boldsymbol{l}_i^{(s)}[t])}{d\boldsymbol{l}_i^{(s)}[t]} = 0 \\ \Leftrightarrow -2\boldsymbol{B'}^T \boldsymbol{x}_i + 2\boldsymbol{B'}^T \boldsymbol{B'} \times_s \mathbf{l}_i^{(s)}[t] + \mu \mathbf{l}_i^{(s)}[t] - \mu \boldsymbol{o}_i[t] = 0 \\ \Leftrightarrow (2\boldsymbol{B'}^T \boldsymbol{B'} + \mu \boldsymbol{I})^{-1} (2\boldsymbol{B'}^T \boldsymbol{x}_i + \mu \boldsymbol{o}_i[t]) = \mathbf{l}_i^{(s)}[t] \end{aligned} \tag{4.38}$$

This element-wise update solves (4.32).

Subproblem (4.33) can be solved uniquely as it is a reduced rank Procrustes Rotation problem [ZHT06]. The solution is $\boldsymbol{B_{(1)}}[t+1] = \boldsymbol{U}\boldsymbol{V}^T$ with

$$\big(\boldsymbol{X} - \boldsymbol{E}[t+1] + \frac{\boldsymbol{\Lambda}_1[t]}{\mu}\big)\big(\bigodot_{m=2}^{M} \mathbf{L}^{(m)}[t+1]\big)^T = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \tag{4.39}$$

being the SVD of $\big(\boldsymbol{X} - \boldsymbol{E}[t+1] + \frac{\boldsymbol{\Lambda}_1[t]}{\mu}\big)\big(\bigodot_{m=2}^{M} \mathbf{L}^{(m)}[t+1]\big)^T$.

The ADM solving (4.28) is outlined in Algorithm 3.

Algorithm 3 Rank-constrained Multilinear Decomposition Algorithm

Input: Data Matrix $\boldsymbol{X} \in \mathbb{R}^{d \times N}$, dimensions K_2, K_3, $\dots K_M$ and parameter λ **Output:** $\boldsymbol{\mathcal{B}}$, $\boldsymbol{E}$, $\boldsymbol{A}^{(2)}$, $\boldsymbol{A}^{(3)}$, $\dots \boldsymbol{A}^{(M)}$

1: Initialisation:
$t \leftarrow 0$,
$\boldsymbol{E}[0] = 0$, $\boldsymbol{B_{(1)}}[0] = 0$, $\{\boldsymbol{A}^{(m)} = 0\}_{m=2}^{M}$, $\{\boldsymbol{L}^{(m)} = 0\}_{m=2}^{M}$,
$\boldsymbol{\Lambda_1}[0] = 0$, $\{\boldsymbol{\Lambda_m}[0] = 0\}_{m=2}^{M}$, $\mu = 10^{-6}$, $\rho = 1.1$, $\epsilon = 10^{-8}$
2: **while** not converged **do**
3: Update $\{\boldsymbol{A}^{(m)}[t+1]\}_{m=2}^{M}$ by

$$\boldsymbol{A}^{(m)}[t+1] = \mathcal{D}_{\mu^{-1}}(\boldsymbol{L}^{(m)}[t] + \frac{\boldsymbol{\Lambda}_m[t]}{\mu})$$

4: Update $\boldsymbol{E}[t+1]$ by

$$\boldsymbol{E}[t+1] = \mathcal{S}_{\frac{\lambda}{\mu}}(\boldsymbol{X} - \boldsymbol{B_{(1)}}[t]\big(\bigodot_{m=2}^{M} \mathbf{L}^{(m)}[t]\big) - \boldsymbol{E}[t] + \frac{\Lambda_1[t]}{\mu})$$

5: Set $\{\boldsymbol{O}_m = \boldsymbol{A}^{(m)}[t+1] - \frac{\boldsymbol{\Lambda}_m[t]}{\mu}\}_{m=2}^{M}$
6: **for all** image $i = 1 \dots N$ **do**
7: **for** each mode s = 2 …M **do**
8: $\boldsymbol{B'} = \boldsymbol{\mathcal{B}}[t] \prod_{m=2, m\neq s}^{M} \times_m \mathbf{l}_i^{(m)}[t]$
9: $\mathbf{l}_i^{(s)}[t+1] = (2\boldsymbol{B'}^T\boldsymbol{B'} + \mu\boldsymbol{I})^{-1}(2\boldsymbol{B'}^T\boldsymbol{x}_i + \mu\boldsymbol{o}_i[t])$
10: **end for**
11: **end for**
12: Update $\boldsymbol{B_{(1)}}[t+1]$ by

$$\big(\boldsymbol{X} - \boldsymbol{E}[t+1] + \frac{\boldsymbol{\Lambda}_1[t]}{\mu}\big)\big(\bigodot_{m=2}^{M} \mathbf{L}^{(m)}[t+1]\big)^T = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T,$$
$$\boldsymbol{B_{(1)}}[t+1] = \boldsymbol{U}\boldsymbol{V}^T$$

13: Update Lagrange multipliers by

$$\boldsymbol{\Lambda_1}[t+1] = \boldsymbol{\Lambda_1}[t] + \mu(\boldsymbol{X} - \boldsymbol{B_{(1)}}[t+1]\big(\bigodot_{m=2}^{M} \mathbf{L}^{(m)}[t+1]\big) - \boldsymbol{E}[t+1])$$
$$\big\{\boldsymbol{\Lambda_m}[t+1] = \boldsymbol{\Lambda_m}[t] + \mu(\boldsymbol{L}^{(m)}[t+1] - \mathbf{A}^{(m)}[t+1])\big\}_{m=2}^{M}$$

14: Update μ by $\mu = min(\rho\mu, 10^{-6})$
15: Check convergence condition:

$$\frac{\|\boldsymbol{X} - \boldsymbol{B_{(1)}}[t+1]\big(\bigodot_{m=2}^{M} \mathbf{L}^{(m)}[t+1]\big) - \boldsymbol{E}[t+1]\|_F^2}{\|\boldsymbol{X}\|_F^2} < \epsilon$$
$$\big\{\frac{\|\boldsymbol{L}^{(m)}[t+1] - \mathbf{A}^{(m)}[t+1]\|_F^2}{\|\boldsymbol{X}\|_F^2} < \epsilon\big\}_{m=2}^{M}$$

16: $t \leftarrow t+1$
17: Tensorise $\boldsymbol{B_{(1)}}[t+1]$ into $\boldsymbol{\mathcal{B}}[t+1] \in \mathbb{R}^{d \times K_2 \times \dots \times K_M}$
18: **end while**

4.2.4 Graph-regularised Decomposition

In practical applications, there might be available side information about the geometric and topological properties of some modes of variation or even available labels. A typical example is a set of facial images with known identities captured under unknown illuminations conditions. To capture such geometric or label information, the graph embedding framework [Yan+07] can be employed by defining a suitable Laplacian graph capturing the available geometric or discriminant information. Therefore, a graph-regularized version of the proposed method is as follows:

$$\begin{aligned} \underset{\boldsymbol{B_{(1)}},\{\boldsymbol{A}^{(m)}\}_{m=2}^{M}}{\arg\min} & \|\boldsymbol{X} - \boldsymbol{B_{(1)}}\big(\bigodot_{m=2}^{M} \mathbf{A}^{(m)}\big)\|_F^2 \\ & + \sum_{s\in S\subseteq\{2,\ldots,M\}} \lambda_s tr(\boldsymbol{A}^{(s)}\boldsymbol{L}^{(s)}\boldsymbol{A}^{(s)T}) \\ & \text{s.t. } \boldsymbol{B}_{\boldsymbol{(1)}}^T\boldsymbol{B_{(1)}} = \mathbf{I}. \end{aligned} \tag{4.40}$$

S is the subset of the modes which contain side information. $\boldsymbol{L}^{(s)} \in \mathbb{R}^{N\times N}$ corresponds to the Laplacian matrix containing either the labelled information or other constraints of a specific mode s. The λ_s are input weight parameters which balance the reconstruction error term with the graph constraints.

This decomposition can be applied in unsupervised, semi-supervised and supervised manner. Depending on the Laplacian matrix, different graph embeddings can be incorporated [Yan+07]. To apply this in a unsupervised manner, $\boldsymbol{L}^{(s)}$ can be specified to learn a manifold structure that conserves local structure such as used in Laplacianfaces [He+05]. In ISOMAP [TSL00] the Laplacian is specified to preserve the geodesic distances of the data points.

In the case of semi-supervised learning, we specify $\boldsymbol{L}^{(s)}$ for specific modes s and include labelled information in $\boldsymbol{L}^{(s)}$ where available. For the samples where the labels are absent, we complete their entries in $\boldsymbol{L}^{(s)}$ by adding connections to their nearest neighbours. This type of semi-supervision has been previously used in [ZGL03]. For supervised learning, we can use the labels to form $\boldsymbol{L}^{(s)}$. The Laplacian can also be specified as the graph embedding corresponding

to LDA [MK01].

(4.40) can be rewritten as the following per-sample formulation:

$$\begin{aligned}\underset{\boldsymbol{B}_{(1)},\{\boldsymbol{A}^{(m)}\}_{m=2}^{M}}{\arg\min} & \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{\mathcal{B}} \prod_{m=2}^{M} \times_m \boldsymbol{a}_i^{(m)}\|_F^2 \\ & + \sum_{S\subseteq\{2,\ldots,M\}} \sum_{k=1}^{N}\sum_{j=1}^{N} \lambda_s tr(l_{kj}\boldsymbol{a}_k^{(s)}\boldsymbol{a}_j^{(s)}) \\ & \text{s.t. } \boldsymbol{B}_{(1)}^T\boldsymbol{B}_{(1)} = \mathbf{I}.\end{aligned} \tag{4.41}$$

The updates for $\mathcal{B}$ and for the unconstrained $\boldsymbol{A}^{(m)}$s where $m \notin S$ would stay exactly the same as in our original method in Section 4.2.1. For the constrained $\boldsymbol{A}^{(s)}$s where $s \in S$, we would need to update them per sample. Using derivation, we can solve for a constrained sample $\boldsymbol{a}_i^{(s)}$ where $s \in S$ by the following update:

$$\begin{aligned}\boldsymbol{a}_i^{(s)} =& (2(\boldsymbol{\mathcal{B}} \prod_{m=2,m\neq s}^{M} \times_m \boldsymbol{a}_i^{(m)})^T(\boldsymbol{\mathcal{B}} \prod_{m=2,m\neq s}^{M} \times_m \boldsymbol{a}_i^{(m)}) + \lambda_s l_{ii}\boldsymbol{I})^{-1} \\ & \times (2(\boldsymbol{\mathcal{B}} \prod_{m=2,m\neq s}^{M} \times_m \boldsymbol{a}_i^{(m)})^T\boldsymbol{x}_i - \lambda_s \sum_{j\neq i} l_{ij}\boldsymbol{a}_j^{(s)} - \lambda_s \sum_{k\neq i} l_{ki}\boldsymbol{a}_k^{(s)})\end{aligned} \tag{4.42}$$

The complete procedure is summarised in Algorithm 4 below.

Algorithm 4 Semi-supervised Multilinear Data Decomposition Algorithm using Graph Constraints

Input: Data Matrix $\boldsymbol{X} \in \mathbb{R}^{d \times N}$, dimensions K_2, K_3, $\dots K_M$, the constrained modes $S \subseteq \{2, ..., M\}$, for each constrained mode $s \in S$: Laplacian matrix $\boldsymbol{L}^{(s)} \in \mathbb{R}^{N \times N}$ and parameter λ_s
Output: $\boldsymbol{\mathcal{B}}$, $\boldsymbol{A}^{(2)}$, $\boldsymbol{A}^{(3)}$, $\dots \boldsymbol{A}^{(M)}$

Initialisation: $t \leftarrow 0$,
$[\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}] \leftarrow$SVD($\boldsymbol{X}$),
$\boldsymbol{B_{(1)}}[0] = \boldsymbol{U}\sqrt{\boldsymbol{\Sigma}}$, $\boldsymbol{Q}[0] = \sqrt{\boldsymbol{\Sigma}}\boldsymbol{V}^T$
for all image $i = 1 \dots N$ **do**
  construct $\boldsymbol{\mathcal{Q}}_i \in \mathbb{R}^{K_M \times K_{M-1} \cdots \times K_2}$ from $\boldsymbol{q}_i[0]$
  $[\boldsymbol{S}_i, \boldsymbol{U}_i] \leftarrow$HOSVD($\boldsymbol{\mathcal{Q}}_i$)
  for each mode m **do**
    $\boldsymbol{a}_i^{(m)}[0] = (\boldsymbol{S}_i)_1^{\frac{1}{M-1}} \boldsymbol{U}_i^{(M-m+1)}$
  end for
end for
while not converged **do**
  for all image $i = 1 \dots N$ **do**
    construct $\boldsymbol{\mathcal{Q}}_i \in \mathbb{R}^{K_M \times K_{M-1} \cdots \times K_2}$ from $\boldsymbol{q}_i[t]$
    $[\boldsymbol{S}_i, \boldsymbol{U}_i] \leftarrow$HOSVD($\boldsymbol{\mathcal{Q}}_i$)
    for each mode m $\notin$ S **do**
      $\boldsymbol{a}_i^{(m)}[t+1] = (\boldsymbol{S}_i)_1^{\frac{1}{M-1}} \boldsymbol{U}_i^{(M-m+1)}$
    end for
    for each mode s $\in$ S **do**
      $\boldsymbol{y} = \boldsymbol{\mathcal{B}} \prod_{m=2, m \neq s}^{M} \times_m \boldsymbol{a}_i^{(m)}[t]$
      $\boldsymbol{a}_i^{(s)}[t+1] = (2\boldsymbol{y}^T\boldsymbol{y} + \lambda_s l_{ii}\boldsymbol{I})^{-1}(2\boldsymbol{y}^T\boldsymbol{x}_i - \lambda_s \sum_{j \neq i} l_{ij}\boldsymbol{a}_j^{(s)}[t] - \lambda_s \sum_{k \neq i} l_{ki}\boldsymbol{a}_k^{(s)}[t])$
    end for
  end for
  $[\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}] \leftarrow$SVD($\boldsymbol{X}\big(\bigodot_{m=2}^{M} \mathbf{A}^{(m)}[t]\big)^T$)
  $\boldsymbol{B_{(1)}}[t+1] = \boldsymbol{U}\boldsymbol{V}^T$
  $\boldsymbol{Q}[t+1] = \boldsymbol{B}_{(1)}[t+1]^T\boldsymbol{X}$
  Check convergence condition:
  $\frac{\|\boldsymbol{X} - \boldsymbol{B_{(1)}}[t+1]\boldsymbol{Q}[t+1]\|_F^2 + \sum_{s \in S} \lambda_s tr(\boldsymbol{A}^{(s)}[t+1]\boldsymbol{L}^{(s)}\boldsymbol{A}^{(s)T}[t+1])}{\|\boldsymbol{X}\|_F^2} < \epsilon$
  Tensorise $\boldsymbol{B_{(1)}}$ into $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{d \times K_2 \times \cdots \times K_M}$
  $t \leftarrow t + 1$
end while

4.3 Experimental Evaluation

In this section, we provide a number of experimental results in order to demonstrate the ability of the proposed method in recovering meaningful modes of variations. Unless otherwise stated, all the data used have been aligned to a reference shape to achieve pixel-wise correspondence.

- We first consider data containing lighting variations i.e., objects under different lights and

decompose the data into illumination and shape/identity components. We demonstrate that our method requires neither complete well-organised data (e.g. all the objects under the same number of lighting conditions), nor labels to find the underlying multilinear structure. We also show that this decomposition can be applied to "in-the-wild" datasets of different objects.

- Then we investigate synthetic 3D facial data that contains both facial expression and identity variations. As there is no texture variability, we provide a proof of concept of our methodology on 3D faces disentangling expression and identity.

- Thirdly we consider 2D data captured in controlled conditions that simultaneously contains lighting, expression and identity variations. As a proof of concept, we show that the decomposition is able to disentangle the 3 different modes of variations. These experiments prove that the model is able to extract meaningful modes of variations from visual data.

- We investigate how to robustly disentangle shape and illumination for "in-the-wild" datasets of faces and ears. The l_1 optimisation providse robustness as it is able to disregard noise and occlusions in the data. We also show that the robust decomposition is able to achieve better reconstruction results than the non-robust version. In a separate experiment, we consider "in-the-wild" videos of a single person. The use of the low-rank constraint allows us to disentangle shape and illumination and reconstruct the face despite synthetic or natural occlusions.

- We also show how our graph-regularised decomposition can be applied to disentangle expression and identity in a semi-supervised setting. We then test the resulting expression components for classification and find that they become more discriminative.

- Finally, we show that the low-rank subspace of shape we obtained from prior decompositions is extremely powerful. We create an unsupervised learning normal estimation pipeline in which we feed the estimated normals from our decomposition method on an "in-the-wild" dataset of faces as the input data to a deep neural network. The resulting

deep network is then able to reconstruct faces from a single non-aligned "in-the-wild" image.

Overall, we demonstrate that our method requires neither complete well-organised data (e.g. all the objects under the same number of lighting conditions), nor labels to find the underlying multilinear structure. We also show that the extended decomposition methods can be applied to "in-the-wild" datasets of different objects to achieve superior performance.

4.3.1 Disentangling Illumination and Shape

Given a dataset of objects in piecewise correspondence (e.g. warped to a mean reference shape) but containing light and identity variations, we seek to recover the illumination mode of variation. We model illumination using first order spherical harmonics consisting of 4 components [BJ01]:

$$\boldsymbol{X} = \boldsymbol{B}_{(1)}(\boldsymbol{L} \odot \boldsymbol{C}), \tag{4.43}$$

where $\boldsymbol{B}_{(1)} \in \mathbb{R}^{d \times 4k}$ is the orthogonal mode-1 matricisation of our proposed tensor $\boldsymbol{\mathcal{B}}$, $\boldsymbol{L} \in \mathbb{R}^{4 \times n}$ is the matrix of first order spherical harmonic light coefficients and $\boldsymbol{C} \in \mathbb{R}^{k \times n}$ is a matrix of shape and identity coefficients. Evidently, this is a special case of our proposed decomposition in (4.2). The choice of k is subject to a trade-off between reconstruction detail of the images and the ability of the decomposition to separate illumination and shape/identity.

Given this setting and an appropriate choice for k, we performed a number of experiments to show that our decomposition is able to separate lighting from shape and identity. Our model indeed recovers illumination as the first mode of variation. The recovered basis $\boldsymbol{B}_{(1)}$, subject to orthogonality constraints, corresponds to a spherical harmonics basis and can be applied to estimate the normals and albedo of the object. The estimated normals are then warped back into the original space of the image and integrated using the method of [FC88b] to recover the 3D reconstruction. We run this experiment on a variety of a number datasets including Photoface [Zaf+13], HELEN [Le+12] and a collected set of human ear images.

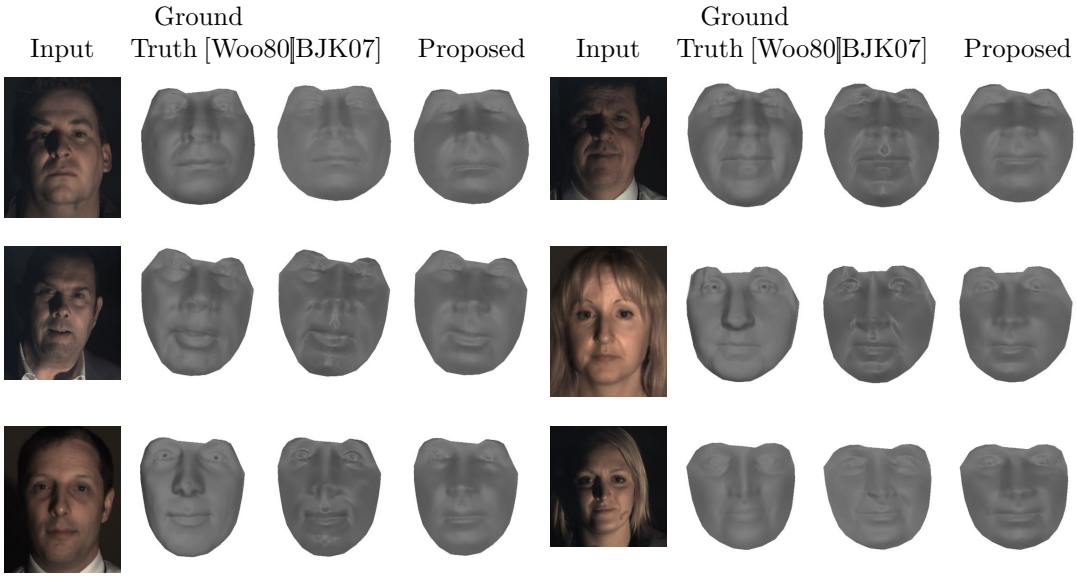


Figure 4.2: 3D shape reconstruction: Comparison of our proposed method with photometric stereo [Woo80] and the person-specific photometric stereo in general lighting of [BJK07]. Images from the Photoface [Zaf+13] dataset.

Method	**Mean angular error against [Woo80]**
[BJK07]	38.35°± 15.63°
Ours	**33.37°± 3.29°**

Table 4.1: Comparison of estimated normals

Comparison using Photoface

Photoface [Zaf+13] is a photometric stereo dataset containing single-view images of people taken under 4 different illumination conditions. We annotated 68 facial landmarks on 273 people from the dataset. The landmarks are used for the warping of the images into/from the mean reference shape. In the absence of ground truth depth or normal data, we use normals recovered from Photometric Stereo (PS) [Woo80] as our ground truth. However, the normals from PS may be biased by outliers so these normals serve as a weak ground truth.

We wish to show that our decomposition works even in the case of an incomplete tensor. To this end, we apply our algorithm to a subset of the dataset: for each person we randomly choose 2 out of the 4 images. The data is incomplete as we do not have the sameset of images for each person. For this experiment we set $k = 40$.

We compare our results against the person-specific results of [BJK07] which utilises all 4 lighting

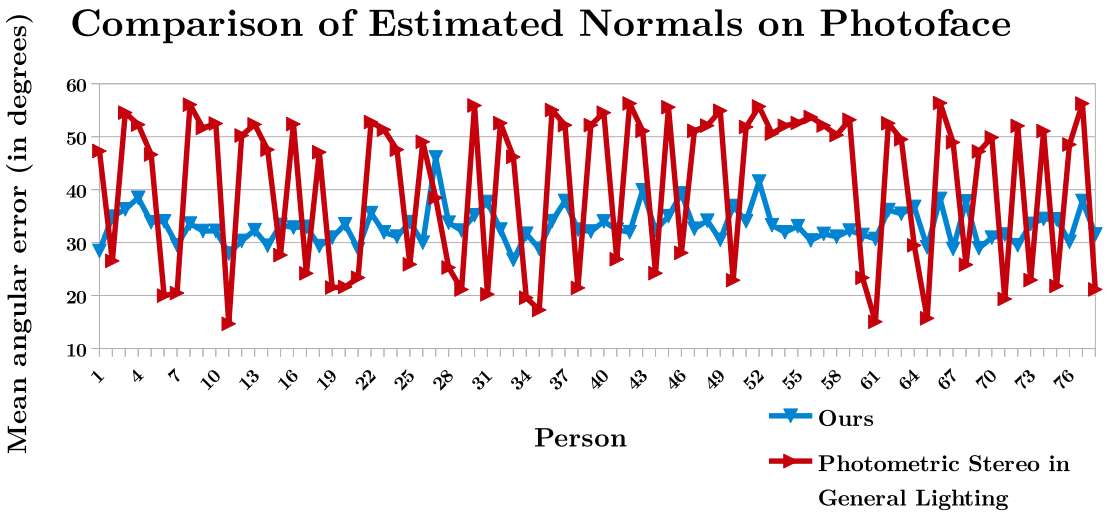


Figure 4.3: Comparison of our proposed method with person-specific photometric stereo in general lighting of [BJK07]. The error has been calculated against the estimated normals from photometric stereo [Woo80].

conditions and applied the method per person. Figure 4.2 shows the sample reconstructions from this experiment. We also plotted the mean angular error between our results and the "ground truth" ones from PS [Woo80] in Figure 4.17 and compare against [BJK07]. We can see that even with missing light information and across multiple identities, our model achieves competitive results, see quantitative results in Table 5.1. We obtain a mean angular error of 33.27°across all 273 people against 38.35°using [BJK07]. In addition our method tends to be more robust with ±3.29° of standard deviation compared with ±15.63° from [BJK07]. These results are the only quantitative results we can obtain as the other datasets do not provide the necessary light information to compute "ground truth" normals from PS.

Comparison using "in-the-wild" Datasets

In this experiment, we show that our method is able to reconstruct a large number of in-the-wild images. In the first experiment, we use the HELEN [Le+12] dataset containing 2000 identities with 1 image per person. We used the 68 facial landmarks from [Sag+16] for the warping to/from the mean reference shape. Figure 4.4 shows the results on a number of challenging images for $k = 40$. [BJK07] cannot be run on this dataset as it would require at least 4 different

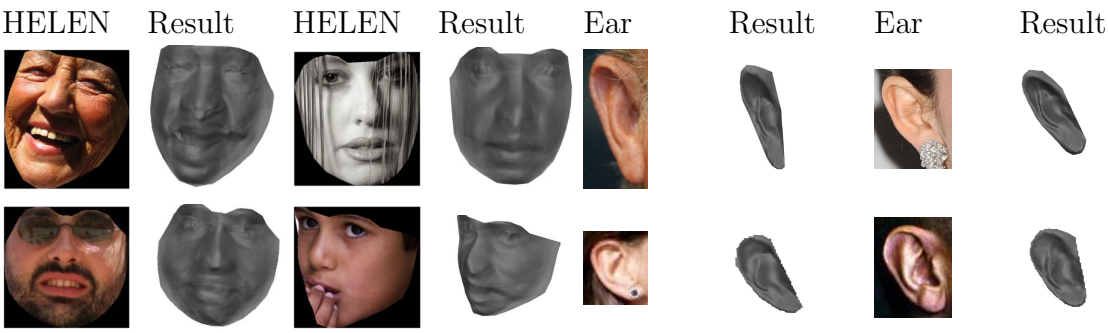


Figure 4.4: Face and ear reconstructions. Sample images from the HELEN [Le+12] and Ear datasets.

lightings per identity.

In the second experiment, we show that our method works on other objects apart from faces. We apply the same methodology to in-the-wild images of ears. We collected 605 images of ears and annotated them with 55 landmarks. Setting $k = 75$, we apply our decomposition and show the results in Figure 4.4.

4.3.2 Disentangling Expression and Identity

In this set of experiments we synthetically generate a dataset of 3D faces where the only variations are identity and expression. The dataset has been created using the Large Scale 3D Morphable Model [Boo+16a] and put in correspondence with the blendshapes of the FaceWarehouse [Cao+14a] so that we can allow for expressions. The dataset with 2000 3D facial meshes consists of 10 facial expressions and 200 identities. We wanted to examine whether our decomposition is able to find a space of identity variation that did not contain expressions. To this purpose we ensured that the facial expressions included in the data did not contain the neutral expression. A sample of the dataset is shown in Figure 4.6.

The decomposition becomes:

$$\boldsymbol{X} = \boldsymbol{B}_{(1)}(\boldsymbol{E} \odot \boldsymbol{C}), \tag{4.44}$$

where $\boldsymbol{B}_{(1)} \in \mathbb{R}^{d \times ek}$ is the orthogonal mode-1 matricisation of tensor $\boldsymbol{\mathcal{B}}$. $\boldsymbol{E} \in \mathbb{R}^{e \times n}$ is the matrix of expression coefficients. e should be set to the approximate number of differing expressions in the data. $\boldsymbol{C} \in \mathbb{R}^{k \times n}$ is assumed to be a matrix of identity coefficients. Evidently, this is

a special case of our proposed decomposition in (4.2). The choice of k is subject to a trade-off between reconstruction detail of the data and the ability of the decomposition to separate expression and identity.

Given this setting and an appropriate choice for k, we performed a number of experiments to show that our decomposition is able to separate expression from identity. Setting $e = 10$ and $k = 50$, we apply the decomposition to discover that $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{d \times e \times k}$ becomes a basis of expression and identity. We note that $\pm\boldsymbol{\mathcal{B}}_{:i:}$ are bases corresponding to expressions in the dataset. The first 10 components of the first 3 bases are plotted in Figure 4.5. We also discover that the first basis $\pm\boldsymbol{\mathcal{B}}_{:0:}$, visualised in Figure 4.5a, is a basis of neutral expressions. This is impressive as the neutral expression did not exist in the original dataset [1].

Thus we can use the neutral expression basis to create synthetic neutral faces of people using the following method. Let $\boldsymbol{B}_0$ denote the neutral expression basis $\boldsymbol{\mathcal{B}}_{:0:}$:

$$\boldsymbol{x}_i' = \boldsymbol{B}_0(\boldsymbol{B}_0^T\boldsymbol{B}_0)^{-1}\boldsymbol{B}_0^T\boldsymbol{x}_i, \tag{4.45}$$

where $\boldsymbol{x}_i'$ denotes the resulting neutral face of the person in $\boldsymbol{x}_i$. The results are visualised in Figure 4.7.

By decoupling $\boldsymbol{E}$, the matrix of expression coefficients and $\boldsymbol{C}$, the matrix representing identities, the decomposition allows us to transfer expressions across identities. Facial expression transfer results are in Figure 4.8.

[1] Nevertheless some, e.g. the 5th column of Figure 4.5 from the left do show some expression. This is mainly because we applied arbitrary scaling to the component (we could have normalized the scaling to the variance associated with this component). The experiments that can conclusively show that our method indeed decoupled identity and expression are shown in Figures 4.7 and 4.8 (transferring expression by changing only the components in $\boldsymbol{E}$).

(a) Basis of first expression $\pm\boldsymbol{\mathcal{B}}_{:0:}$. The first row corresponds to positive variance and the second row corresponds to negative variance along the basis.

(b) Basis of second expression $\pm\boldsymbol{\mathcal{B}}_{:1:}$

(c) Basis of third expression $\pm\boldsymbol{\mathcal{B}}_{:2:}$

Figure 4.5: The 3 first expression bases from the decomposition of the synthetic 3D data. We note that despite the data not containing the neutral expression, the first expression basis corresponds to the neutral expression. The other basis display different expression variations.

Figure 4.6: Sample data of the synthetic 3D dataset. Images 1 to 3 from the left show different identities and images 4 to 6 different expressions.

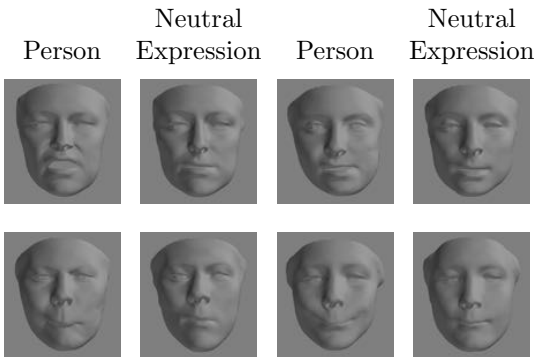


Figure 4.7: Neutralising expressions: We show the result of generating synthetic neutral faces by using the neutral expression basis. The results look promising.

Figure 4.8: We show the results of the expression transfer experiment. The transferred expressions look convincing.

4.3.3 Disentangling Illumination, Expression and Identity

In this experiment, we test our decomposition on data that simultaneously contains lighting, facial expression variations as well as multiple identities such as the Multi-PIE [Gro+10] dataset. We select 147 identities, 5 expressions and 5 illuminations from the overall dataset. Our subset consists of 3675 images. We rigidly align the data to a mean shape in order to conserve the facial expression variations. Frequently, lighting becomes the first mode of variation in visual data.

We model illumination using first order spherical harmonics consisting of 4 components [BJ01]. The decomposition can be adapted in this manner:

$$\boldsymbol{X} = \boldsymbol{B}_{(1)}(\boldsymbol{L} \odot \boldsymbol{E} \odot \boldsymbol{C}), \tag{4.46}$$

where $\boldsymbol{L} \in \mathbb{R}^{4\times n}$ is the matrix of first order spherical harmonic light coefficients, $\boldsymbol{E} \in \mathbb{R}^{e\times n}$ and $\boldsymbol{C} \in \mathbb{R}^{k\times n}$ represent expression and identity coefficients respectively.

Setting $e = 5$ and $k = 40$, we obtain a resulting tensor $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{d\times 4\times e\times k}$. Our model indeed recovers illumination as the first mode of variation. The recovered basis $\boldsymbol{B}_{(1)}$, subject to orthogonality constraints, corresponds to a spherical harmonics basis and can be applied to estimate the normals and albedo of the object. The estimated normals are then warped back into the original space of the image and integrated using the method of [FC88b] to recover the 3D reconstruction, see Figure 4.9.

Figure 4.9: 3D Reconstruction on Multi-PIE [Gro+10] dataset. The results show that 3-way disentanglement is possible.

Figure 4.10: Expression transfer on Multi-PIE. As our decomposition reduces the dimensionality of the images in the dataset, we show the images with the transferred expression next to the reconstructed image of the ground truth from the dataset. Given the decomposition, the reconstruction represents the result of a plausible expression transfer.

4.3.4 Robust Disentanglement of Illumination and Shape

As the decomposition also decouples expression and identity variations into $\boldsymbol{E}$ and $\boldsymbol{C}$, we can use this to transfer facial expressions from one person to another person. Adapting the equation (4.1) to this decomposition (4.46), we specify for images $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ where the two images are of different people and expressions:

$$\boldsymbol{x}_i = \boldsymbol{\mathcal{B}} \times_2 \boldsymbol{l}_i \times_3 \boldsymbol{e}_i \times_4 \boldsymbol{c}_i, \quad \boldsymbol{x}_j = \boldsymbol{\mathcal{B}} \times_2 \boldsymbol{l}_j \times_3 \boldsymbol{e}_j \times_4 \boldsymbol{c}_j \tag{4.47}$$

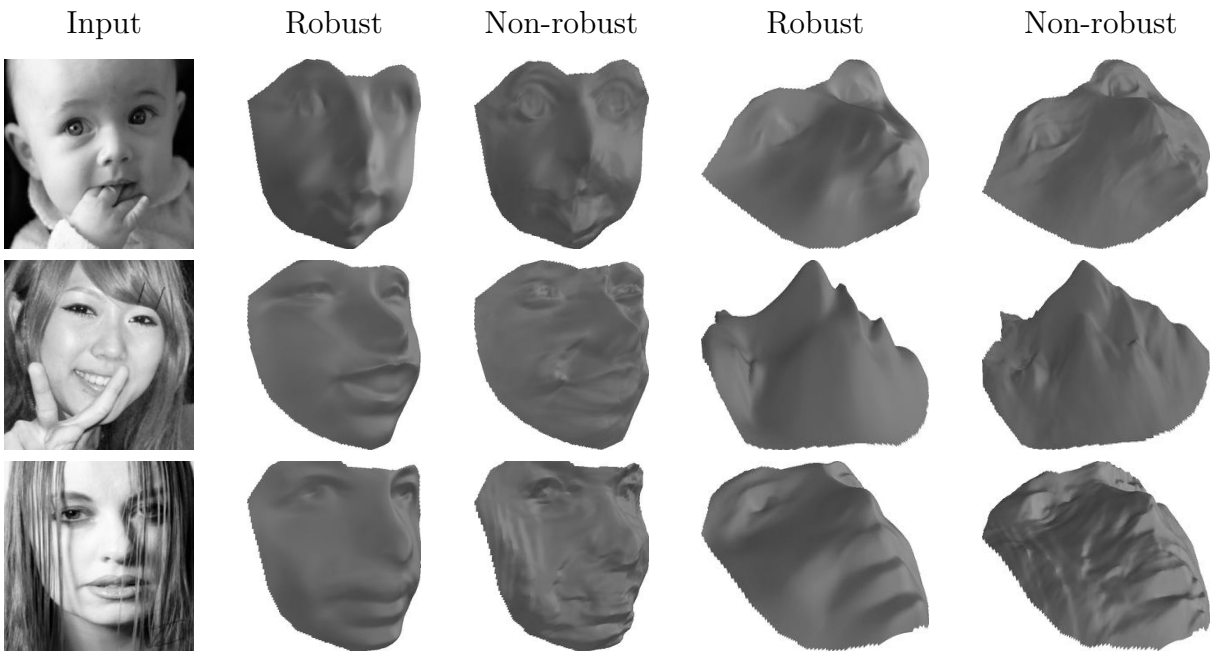


Figure 4.11: **Comparison of the robust and non-robust decomposition**. Images from the HELEN [Le+12] dataset.

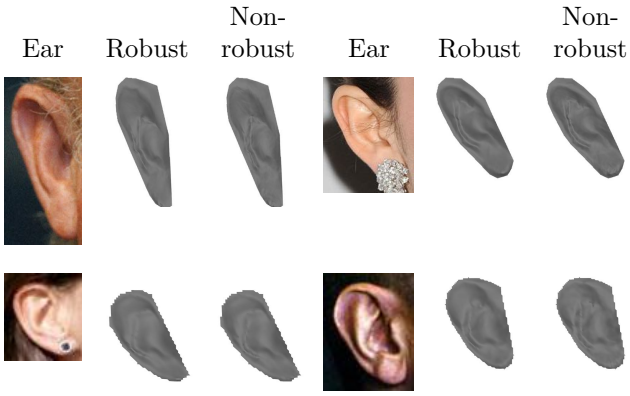


Figure 4.12: Ear reconstructions. Sample images from the Ear dataset.

By swapping $\boldsymbol{c}_i$ with $\boldsymbol{c}_j$, the identity coefficients, we can create a synthetic image $\boldsymbol{x}_{i_j}$ containing the expression of person i and identity of person j.

$$\boldsymbol{x}_{i_j} = \boldsymbol{\mathcal{B}} \times_2 \boldsymbol{l}_i \times_3 \boldsymbol{e}_i \times_4 \boldsymbol{c}_j. \tag{4.48}$$

In this way, a synthetic dataset of people with new expressions are created. Sample results of the expression transfer experiment are shown in Figure 4.10. Some of the examples are challenging ones such as transferring expressions across gender. The Multi-PIE [Gro+10] dataset contains a number of people wearing glasses which lead to artefacts in the area around the eyes in the

synthetic images.

We test this synthetic data via an expression classification experiment to verify that the new synthetic expressions are recognisable. Specifically, we trained a linear SVM model with the original dataset and respective expression labels and used the synthetic dataset as test data. The prediction results are listed in Table 4.2. The high accuracy of 85.1% shows that the synthetic data manages to model the expressions contained in the original data.

Data	**Prediction accuracy**
Synthetic expressions data	0.851

Table 4.2: Prediction accuracy on synthetic dataset

Using the robust decomposition method from Section 4.2.2, we show on two different dataset that the method is able to robustly reconstruct objects "in-the-wild". As "in-the-wild" data often contain noise and natural occlusions, a robust decomposition seems to be ideal in this case to separate the noise from the actual shape.

The decomposition applied in the below experiments is:

$$\boldsymbol{X} = \boldsymbol{B_{(1)}}(\boldsymbol{L} \odot \boldsymbol{C}) + \boldsymbol{E}, \tag{4.49}$$

where $\boldsymbol{L} \in \mathbb{R}^{4\times n}$ is the the matrix of first order spherical harmonic light coefficients, $\boldsymbol{C} \in \mathbb{R}^{k\times n}$ is a matrix of shape and identity coefficients and $\boldsymbol{E} \in \mathbb{R}^{d\times n}$ represents the matrix of sparse errors. A sparsity constraint has been put on $\boldsymbol{E}$ in the problem formulation. This is a special case of our proposed decomposition in (4.15).

In order to validate the usefulness of the robust decomposition, we have added 1% salt&pepper noise on the Photoface dataset. We then compare the estimated normals of our robust and non-robust decompositions on this noisy Photoface dataset. The "ground truth" normals were obtained from the clean Photoface dataset using PS [Woo80]. Figure 4.13 shows the sample reconstructions from this experiment. From the 3D shape, we observe that the non-robust reconstruction also reconstructs the noise. The robust reconstruction obtains a smooth shape without the noise. The result of the quantitative evaluation can be found in Table 4.3. This

Method	Mean±Std	<30°	<35°
Ours- Non-Robust	39.81°± 12.36°	0.7%	42.7%
Ours- Robust	**33.86°± 4.84°**	**16.4%**	**71.5%**

Table 4.3: Angular error for our method with and without robustness on Photoface containing 1% salt&pepper noise. Our robust method outperforms our basic method in terms of 3D reconstruction from noisy data.
clearly demonstrates that the robust decomposition outperforms the non-robust decomposition on noisy data.

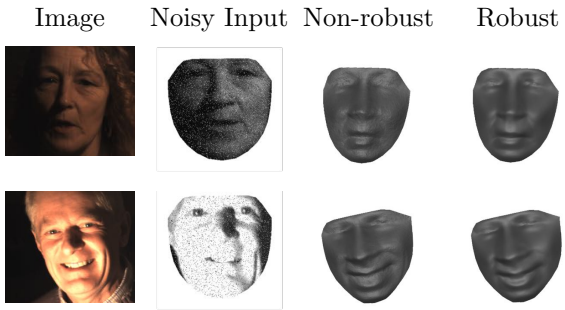


Figure 4.13: Sample reconstruction from the Photoface dataset with 1% salt&pepper noise using non-robust and robust decomposition.

We then show on two different "in-the-wild" datasets that the robust method outperforms the non-robust version.

Faces "In-the-wild"

In this experiment, we show that our method is able to robustly reconstruct a large number of "in-the-wild" images. We use the HELEN [Le+12] dataset containing 2000 identities with 1 image per person. We used the 68 facial landmarks from [Sag+16] for the warping to/from the mean reference shape. Figure 4.11 shows the results on a number of challenging images for $k = 200$. Clearly the robust method is able to separate illumination and appearance better than the non-robust method.

Ears "In-the-wild"

In this experiment, we show that our method works on other objects apart from faces. We collected 605 "in-the-wild" images of ears and annotated them with 55 landmarks. The land-

Method	**Mean±Std**	**<5°**	**<10°**
Ours-Without Low-rank Constraints	8.97°± 2.02°	0%	88.5%
Ours-With Low-rank Constraints	**6.10°± 1.74°**	**55.5%**	**100%**

Table 4.4: Angular error for our method with and without low-rank constraints on videos containing baboon patch occlusions.

marks were used for the warping to/from the mean reference shape. Setting $k = 100$, we apply our decomposition and show the results in Figure 4.12. The results indicate that the robust decomposition method outperforms the non-robust method.

4.3.5 Disentanglement of Illumination and Shape with Low-rank Constraints

Videos of a single person can specifically profit from the rank-constrained decomposition method from Section 4.2.3. As the rank-constrained decomposition also incorporates the l_1 norm, we can apply this method on noisy data.

We show this using two experiments: In the first experiment, we synthetically occlude part of a video with baboon patches. 20% of the frames in the video has been occluded by baboon patches. For each of those occluded frames, the baboon patch covers 10% of the frame. In the second experiment, we run the same method on videos where some frames have been naturally occluded by hands or hair.

Figure 4.14 shows the qualitative result of the two experiments. In both situations, we were able to reconstruct the faces quite well despite the occlusions. We can see how the reconstruction strongly resembles the person in the video.

Table 4.4 shows the quantitative result in the case of synthetic occlusion. The ground truth are the normals estimated without low-rank constraints on videos without occlusion ($k = 20$). We compare our methods with and without low-rank constraints ($k = 20$) on the videos containing occlusions. Clearly, our method with low-rank constraints is robust to occlusions and outperforms our method without low-rank constraints.

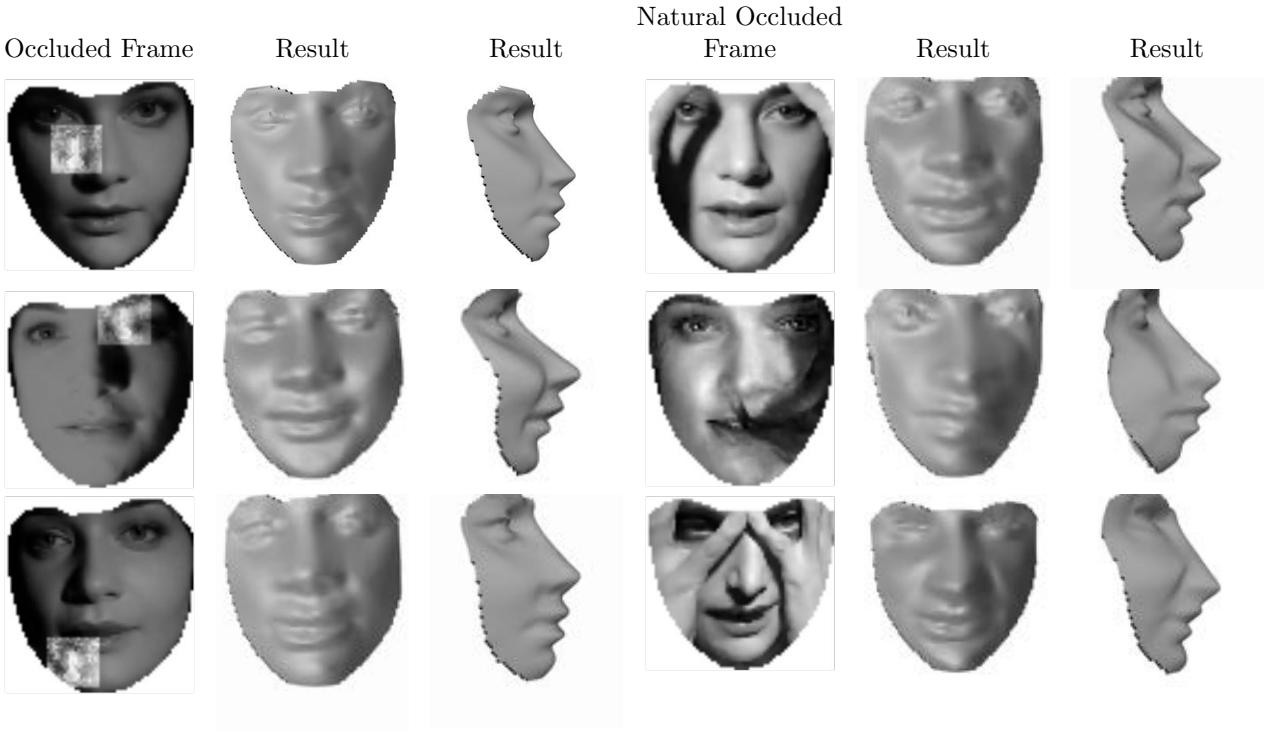


Figure 4.14: Face reconstructions from occluded video frames using the rank-constrained decomposition.

4.3.6 Semi-Supervised Disentanglement of Expression and Identity

We have collected a new 3D database of people displaying 6 different expressions (happiness, disgust, anger, surprise, sadness and fear). We used NICP [Zaf+17] to bring them in correspondence with the Basel face model. In total we collect samples from 200 people, each sample was annotated with the expression label of the 6 different expressions. We applied the unsupervised version of the decomposition without graph-regularisation using the 6 labels. Keeping the identity parameters fixed to the ones of the mean face, we randomly sample values for $\boldsymbol{E}$. Figure 4.15 shows how we can generate 3D faces corresponding to each of the 6 expressions using this approach.

Then we split our dataset into 5 random splits, each time keeping 1000 samples for training and 200 for testing. We apply both the unsupervised and graph-regularised semi-supervised decomposition on the data and use a nearest neighbour classifier on $\boldsymbol{E}$ to predict the expressions of the test set. Table 4.5 shows how incorporating supervision via graph-regularisation strongly improves the expression classification accuracy.

Method	Accuracy
Unsupervised Decomposition	84.5%
Graph-regularised Semi-supervised Decomposition	**96.0%**

Table 4.5: Expression classification results using unsupervised and semi-supervised decomposition.

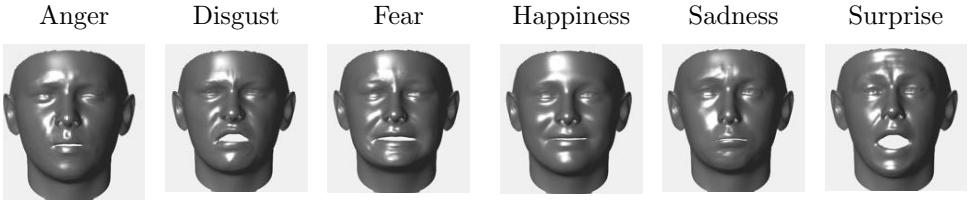


Figure 4.15: 3D faces generated by keeping the identity component $\boldsymbol{C}$ fixed and randomly sampling the expression component $\boldsymbol{E}$.

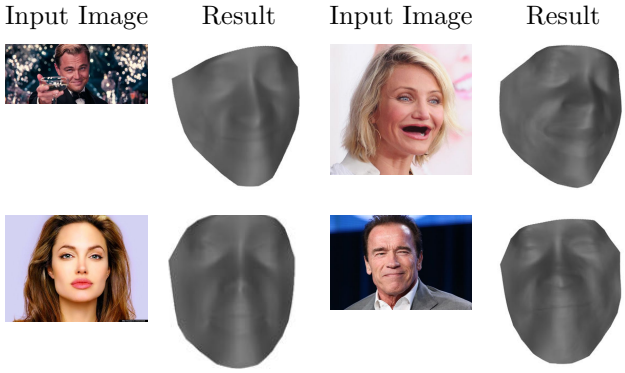


Figure 4.16: Face reconstructions from single "in-the-wild" images using the deep unsupervised model trained on HELEN. Though the results may not seem impressive, they are obtained from a pipeline of two unsupervised methods. In this aspect, the results are good.

4.3.7 Unsupervised Normal Estimation using Deep Learning

In this experiment, we use the normals estimated on the HELEN dataset [Le+12] using the proposed method to train a 'fully convolutional' network to perform normal estimation on faces. This fully unsupervised pipeline consists of obtaining normals estimated using our decomposition and then feeding them as input data to a deep network. We use the network based on ResNet-50 [He+16] and the UberNet architecture [Kok16] for surface normal estimation. UberNet used a Deep Convolutional Neural Network (DCNN) for surface normal estimation among a series of other tasks.

The architecture is as follows: We use skip layers [HAG15] to incorporate both low- and high-level features into the task. We pool features from layers conv1, block2/unit4, block3/unit6, block4/unit3 which correspond to C1, . . . , C4 in Fig. 4.18. At each layer, linear mappings

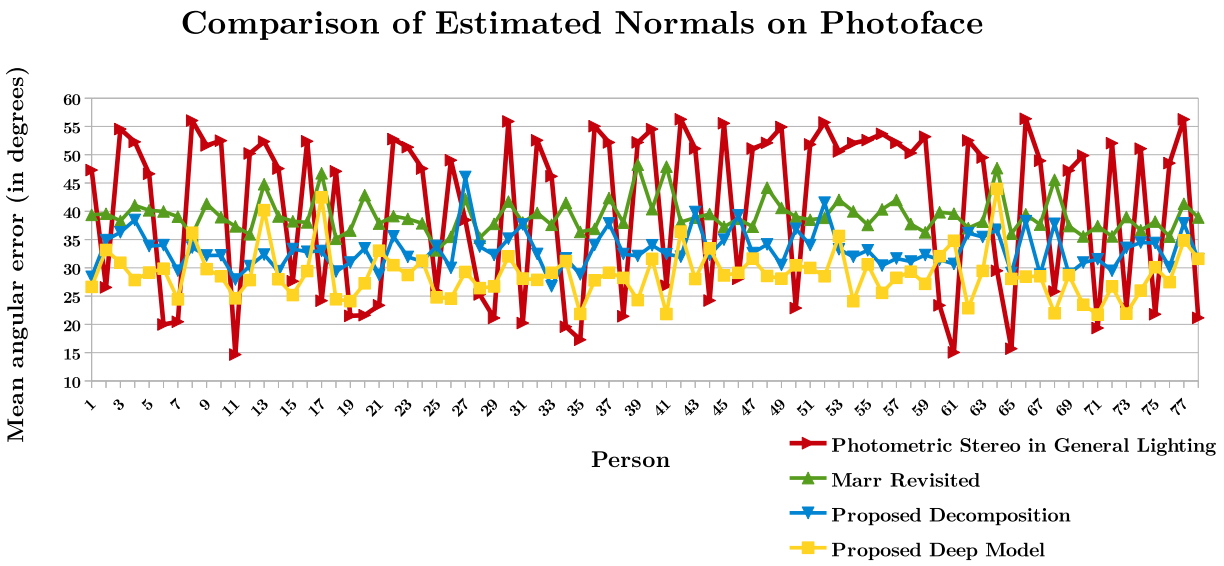


Figure 4.17: Comparison of our two proposed methods with person-specific photometric stereo in general lighting of [BJK07] and a generic state-of-the-art network [BRG16]. The error has been calculated against the estimated normals from photometric stereo [Woo80].

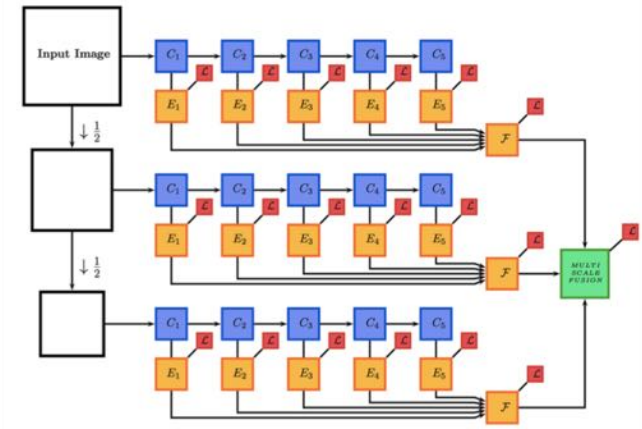


Figure 4.18: The deep model: ResNet-50 based architecture.

from the high-dimensional intermediate neuron activation space to the three-dimensional output space for normal estimation is learned. These intermediate layers are processed with batch normalisation [IS15] to bring the intermediate activations into a common scaling. To be able to work on varying face sizes in images, a 3-scale pyramid is used. The image is down-sampled at scales 2 and 3 by half and a quarter times respectively. The whole architecture of the model is visualised in Figure 4.18.

In order to quantitatively estimate the performance of the learned deep model, we require some level of ground truth data. Photoface [Zaf+13] is a photometric stereo dataset containing single-

Method	**Mean±Std against [Woo80]**	**<35°**	**<40°**
[BJK07]	38.35°± 15.63°	46.4%	46.8%
[BRG16]	38.77°± 3.27°	4.5%	73.0%
Ours-Decomposition Method	33.37°± 3.29°	75.3%	96.3%
Ours-Unsupervised Deep Model	**28.53°± 4.23°**	**93.6%**	**97.8%**

Table 4.6: Angular error for the various surface normal estimation methods on the Photoface [Zaf+13] dataset

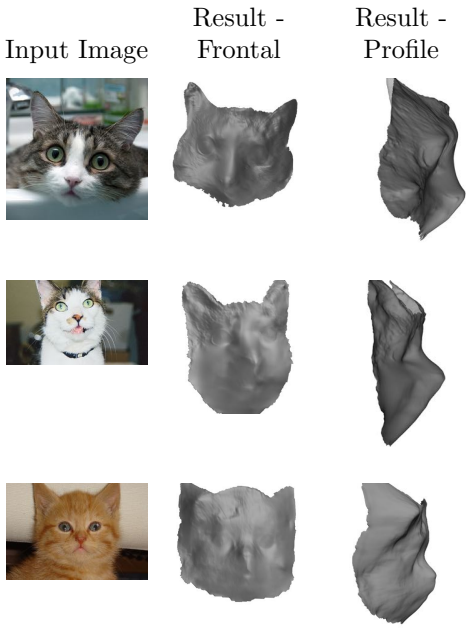


Figure 4.19: Face reconstructions from single "in-the-wild" cats images using the deep unsupervised model trained on human faces and ears.

view images of people taken under 4 different illumination conditions. We annotated 68 facial landmarks on 57 people from the dataset. The landmarks are used for the warping of the images into/from the mean reference shape for our proposed decomposition from Section 4.2.1. In the absence of ground truth depth or normal data, we use normals recovered from Photometric Stereo (PS) [Woo80] as our ground truth. However, the normals from PS may be biased by outliers so these normals serve as a weak ground truth.

By testing our learned deep model on a previously unseen dataset such as Photoface is challenging as the images have been taken under different conditions to the ones in the HELEN training dataset. We plotted the mean angular error between our decomposition method (Section 4.2.1) results and the "ground truth" ones from PS [Woo80] in Figure 4.17 and compare against our learned deep model.

Our decomposition method uses 2 randomly selected images of a person under different lighting

conditions. The method in [BJK07] requires 4 images of a person under different lighting conditions. [BRG16] is a generic state-of-the-art network which reconstructs from one image. Our deep model similarly only requires one image per person. From the quantitative results in Table 5.1, our deep model obtains a mean angular error of 28.53°across 273 people against 38.77°using [BRG16]. It clearly shows that the deep model works very well on the Photoface test data and performs comparably and even slightly better than our decomposition method. The deep model obtains a mean angular error of 28.53°against 33.37°using the decomposition method. The reason for the strong performance is the variation of k. As the decomposition method is restricted by the number of annotated images in the Photoface dataset, the k used is 40. The deep model is trained on the larger HELEN dataset with $k = 400$. This suggests that the deep model may be able to extract more reconstruction details from the Photoface images than the decomposition method.

We show some sample reconstructions using the state-of-the-art general reconstruction network [BRG16] versus our proposed deep network in Figure 4.20.

The results are extremely encouraging as they indicate that we can apply this unsupervised deep model directly to "in-the-wild" internet images of faces. Unlike the proposed decomposition method, the deep model does not require any warping of the images to/from the reference frame. This also is very beneficial for "in-the-wild" images. Figure 4.16 shows the reconstruction results of our deep model on internet images. The reconstructions nicely mirrors the facial expressions contained in the images.

In addition, we trained a separate network with the images from the HELEN and Ear datasets. The ground truth normals used were again the result of our decomposition method. We tested the model on human faces and found that it reconstructs more details. Then we tested this additionally on internet images of cats and found that due to the similar facial structure (eyes and nose), the model is able to reconstruct cat faces. The reconstructions can be seen in Figure 4.19. The model even manages to reconstruct the fur details, which is impressive.

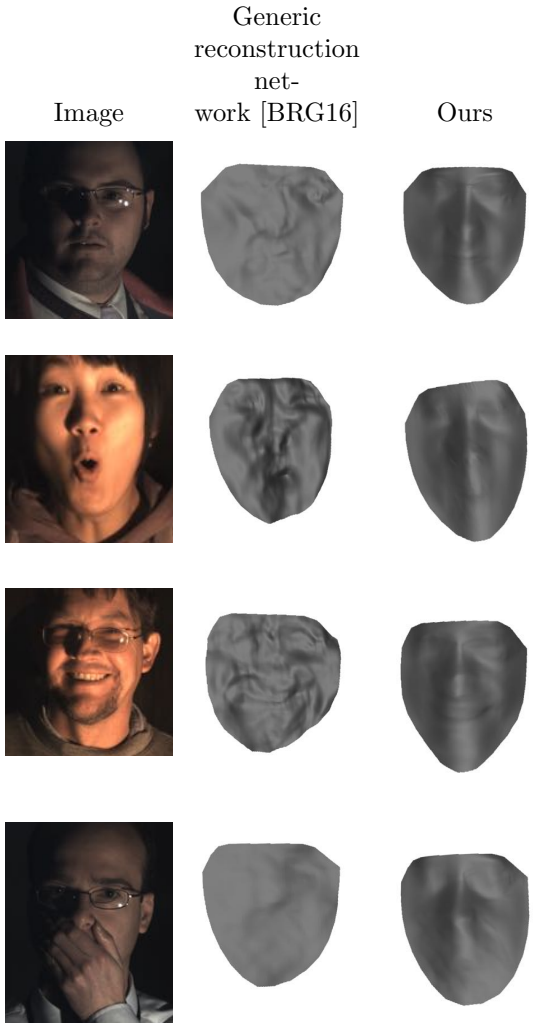


Figure 4.20: Sample reconstruction from the Photoface dataset with state-of-the-art general network [BRG16](middle) and our deep network trained in a unsupervised fashion (right).

4.4 Limitations

Our proposed method is unsupervised in the way that it does not require label information. However a certain level of knowledge about the dataset is required. That knowledge should be in the form of

1. the number and types of modes of the data. In this chapter we focused on illumination, expression and identity as modes of variation.

2. an estimation of how many components is sufficient to capture the variations contained in each mode. We assumed that for lighting 4 components would be sufficient, whereas for expression 6 would be appropriate.

2D image datasets should also be class-specific and aligned. In terms of illumination disentanglement, a Lambertian assumption is incorporated in the formulation. This means that the surface to be reconstructed is assumed to be Lambertian. Datasets containing a specific type of objects with Lambertian-like surfaces are ideal for this method.

We have shown that 2D and 3D datasets of faces are well-suited for this unsupervised multilinear decomposition.

4.5 Conclusions

In this chapter, we have proposed an unsupervised method able to discover the assumed multilinear structure in visual data. To this end an alternating least squares algorithm has been developed. We extended this method to incorporate robustness and rank constraints. Our experiments show that the method is able to discover the multilinear structure of "in-the-wild" visual data without the presence of labels or well-organised input data. Additional experiments using an unsupervised deep learning pipeline show the application of the method directly on internet images of human as well as cat faces.

Chapter 5

Neuro-Tensorial Approach for Learning Disentangled Representations

5.1 Introduction

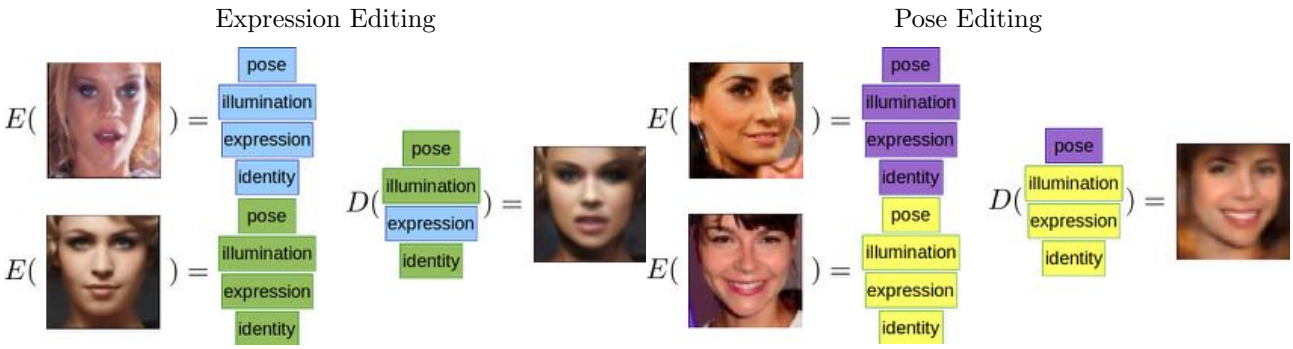


Figure 5.1: Given a single in-the-wild image, our network learns disentangled representations for pose, illumination, expression and identity. Using these representations, we are able to manipulate the image and edit the pose or expression.

The appearance of visual objects is significantly affected by multiple factors of variability such as, for example, pose, illumination, identity, and expression in case of faces. Each factor accounts for a source of variability in the data, while their complex interactions give rise to the observed entangled variability. Discovering the modes of variation, or in other words disentangling the latent factors of variations in visual data, is a very important problem in the intersection of statistics, machine learning, and computer vision.

Recent unsupervised tensor decompositions methods [TSH13; Wan+17b] automatically discover the modes of variation in unlabelled data. In particular, the most recent one [Wan+17b] assumes that the original visual data have been produced by a hidden multilinear structure and the aim of the unsupervised tensor decomposition is to discover both the underlying multilinear structure, as well as the corresponding weights (coefficients) that best explain the data. Special instances of the unsupervised tensor decomposition are the Shape-from-Shading (SfS) decompositions in [Kem13a; SPZ15] and the multilinear decompositions for 3D face description in [Wan+17b]. In [Wan+17b], it is shown that the method indeed can be used to learn representations where many modes of variation have been disentangled (e.g., identity, expression and illumination etc.). Nevertheless, the method in [Wan+17b] is not able to find pose variations and bypasses this problem by applying it to faces which have been frontalised by applying a warping function (e.g., piece-wise affine warping [MB04]).

Another promising line of research for discovering latent representations is unsupervised Deep Neural Networks (DNNs). Unsupervised DNNs architectures include the Auto-Encoders (AE) [BCV13b], as well as the Generative Adversarial Networks (GANs) [Goo+14] or adversarial versions of AE, e.g., the Adversarial Auto-Encoders (AAE) [Mak+15]. Even though GANs, as well as AAEs, provide very elegant frameworks for discovering powerful low-dimensional embeddings without having to align the faces, due to the complexity of the networks, unavoidably all modes of variation are multiplexed in the latent-representation. Only with the use of labels it is possible to model/learn the manifold over the latent representation, usually as a post-processing step [Shu+17].

In this chapter, we show that it is possible to learn a disentangled representation of the human face captured in arbitrary recording conditions in an pseudo-supervised manner[1] by imposing a multilinear structure on the latent representation of an AAE [Shu+17]. We define pseudo-supervision as using information from another approach which is also not fully supervised. To the best of our knowledge, this is the first time that unsupervised tensor decompositions have been combined with DNNs for learning disentangled representations. We demonstrate

[1]Our methodology uses the information produced by an automatic 3D face fitting procedure [Boo+17] but it does not make use of any labels in the training set.

the power of the proposed approach by showing expression/pose transfer using only the latent variable that is related to expression/pose. We also demonstrate that the disentangled low-dimensional embeddings are useful for many other applications, such as facial expression, pose, and identity recognition and clustering. An example of the proposed approach is given in Fig. 5.1. In particular, the left pair of images have been decomposed, using the encoder of the proposed neural network $E(\cdot)$, into many different latent representations including latent representations for pose, illumination, identity and expression. Since our framework has learned a disentangled representation we can easily transfer the expression by only changing the latent variable related to expression and passing the latent vector into the decoder of our neural network $D(\cdot)$. Similarly, we can transfer the pose merely by changing the latent variable related to pose.

5.2 Methodology

In this section, we will introduce the main multilinear models used to describe three different image modalities, namely texture, 3D shape and 3D surface normals. To this end, we assume that for each different modality there is a different core tensor but all modalities share the same latent representation of weights regarding identity and expression. During training all the core tensors inside the network are randomly initialised and learnt end-to-end. In the following, we assume that we have a set of n facial images (e.g., in the training batch) and their corresponding 3D facial shape, as well as their normals per pixel (the 3D shape and normals have been produced by fitting a 3D model on the 2D image, e.g., [Boo+17]).

5.2.1 Facial Texture

The main assumption here follows from [Wan+17b]. That is, the rich structure of visual data is a result of multiplicative interactions of hidden (latent) factors and hence the underlying multilinear structure, as well as the corresponding weights (coefficients) that best explain the data can be recovered using the unsupervised tensor decomposition [Wan+17b]. Indeed, fol-

lowing [Wan+17b], disentangled representations can be learnt (e.g., identity, expression, and illumination, etc.) from frontalised facial images. The frontalisation process is performed by applying a piecewise affine transform using the sparse shape recovered by a face alignment process. Inevitably, this process suffers from warping artifacts. Therefore, rather than applying any warping process, we perform the multilinear decomposition only on near frontal faces, which can be automatically detected during the 3D face fitting stage. In particular, assuming a near frontal facial image rasterised in a vector $\boldsymbol{x}_f \in \mathbb{R}^{k_x \times 1}$, given a core tensor $\mathcal{Q} \in \mathbb{R}^{k_x \times k_l \times k_{exp} \times k_{id}}$ [2], this can be decomposed as

$$\boldsymbol{x}_f = \mathcal{Q} \times_2 \boldsymbol{z}_l \times_3 \boldsymbol{z}_{exp} \times_4 \boldsymbol{z}_{id}, \tag{5.1}$$

where $\boldsymbol{z}_l \in \mathbb{R}^{k_l}, \boldsymbol{z}_{exp} \in \mathbb{R}^{k_{exp}}$ and $\boldsymbol{z}_{id} \in \mathbb{R}^{k_{id}}$ are the weights that correspond to illumination, expression and identity respectively. The equivalent form in case that we have a number of images in the batch stacked in the columns of a matrix $\mathbf{X}_f \in \mathbb{R}^{k_x \times n}$ is

$$\boldsymbol{X}_f = \boldsymbol{Q}_{(1)}(\boldsymbol{Z}_l \odot \boldsymbol{Z}_{exp} \odot \boldsymbol{Z}_{id}), \tag{5.2}$$

where $\boldsymbol{Q}_{(1)}$ is a mode-1 matricisation of tensor $\mathcal{Q}$ and $\boldsymbol{Z}_l$, $\boldsymbol{Z}_{exp}$ and $\boldsymbol{Z}_{id}$ are the corresponding matrices that gather the weights of the decomposition for all images in the batch. That is, $\boldsymbol{Z}_{exp} \in \mathbb{R}^{k_{exp} \times n}$ stacks the n latent variables of expressions of the images, $\boldsymbol{Z}_{id} \in \mathbb{R}^{k_{id} \times n}$ stacks the n latent variables of identity and $\boldsymbol{Z}_l \in \mathbb{R}^{k_l \times n}$ stacks the n latent variables of illumination.

Although the modelling of facial texture is only applied on near-frontal faces, the model is able to leverage this information in the case of non-frontal faces for disentangling.

[2] Tensors notation: Tensors (i.e., multidimensional arrays) are and denoted by calligraphic letters, e.g., $\boldsymbol{\mathcal{X}}$. The *mode-m matricisation* of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ maps $\boldsymbol{\mathcal{X}}$ to a matrix $\mathbf{X}_{(m)} \in \mathbb{R}^{I_m \times \bar{I}_m}$. The *mode-m vector product* of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$ with a vector $\mathbf{x} \in \mathbb{R}^{I_m}$, denoted by $\boldsymbol{\mathcal{X}} \times_n \mathbf{x} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N}$.

The *Kronecker product* is denoted by $\otimes$ and the *Khatri-Rao* (i.e., column-wise Kronecker product) product is denoted by $\odot$. More details on tensors and multilinear operators can be found in [KB08].

5.2.2 3D Facial Shape

It is quite common to use a bilinear model for disentangling identity and expression in 3D facial shape [BW16]. Hence, for 3D shape we assume that there is a different core tensor $\mathcal{B} \in \mathbb{R}^{k_{3d} \times k_{exp} \times k_{id}}$ and each 3D facial shape $\boldsymbol{x}_{3d} \in \mathbb{R}^{k_{3d}}$ can be decomposed as:

$$\boldsymbol{x}_{3d} = \mathcal{B} \times_2 \boldsymbol{z}_{exp} \times_3 \boldsymbol{z}_{id}, \tag{5.3}$$

where $\boldsymbol{z}_{exp}$ and $\boldsymbol{z}_{id}$ are exactly the same weights as in the texture decomposition (5.2). The tensor decomposition for the n images in the batch is therefore written as

$$\boldsymbol{X}_{3d} = \boldsymbol{B}_{(1)}(\boldsymbol{Z}_{exp} \odot \boldsymbol{Z}_{id}), \tag{5.4}$$

where $\boldsymbol{B}_{(1)}$ is a mode-1 matricization of tensor $\mathcal{B}$.

5.2.3 Facial Normals

The tensor decomposition we opted to use for facial normals was exactly the same as the texture, hence we can use the same core tensor and weights. The difference is that since facial normals do not depend on illumination parameters (assuming a Lambertian illumination model), we just need to replace the illumination weights with a constant[3]. Thus, the decomposition for normals can be written as

$$\boldsymbol{X}_N = \boldsymbol{Q}_{(1)}(\frac{1}{k_l}\mathbf{1} \odot \boldsymbol{Z}_{exp} \odot \boldsymbol{Z}_{id}), \tag{5.5}$$

where $\mathbf{1}$ is a matrix of ones.

5.2.4 3D Facial Pose

Finally, we define another latent variable regarding 3D pose. This latent variable $\boldsymbol{z}_p \in \mathbb{R}^9$ represents a 3D rotation. We denote by $\boldsymbol{x}^i \in \mathbb{R}^{k_x}$ an image at index i. The indexing is denoted

[3]This is also the way that normals are computed in [Wan+17b] up to a scaling factor

in the following by the superscript. The corresponding $\boldsymbol{z}_p^i$ can be reshaped into a rotation matrix $\boldsymbol{R}^i \in \mathbb{R}^{3\times 3}$. As proposed in [Wor+17], we apply this rotation to the feature of the image $\boldsymbol{x}^i$ created by 2-way synthesis (explained in Section 5.2.5). This feature vector is the i-th column of the feature matrix resulting from the 2-way synthesis $(\boldsymbol{Z}_{exp} \odot \boldsymbol{Z}_{id}) \in \mathbb{R}^{k_{exp}k_{id}\times n}$. We denote this feature vector corresponding to a single image as $(\boldsymbol{Z}_{exp} \odot \boldsymbol{Z}_{id})^i \in \mathbb{R}^{k_{exp}k_{id}}$. Next $(\boldsymbol{Z}_{exp} \odot \boldsymbol{Z}_{id})^i$ is reshaped into a $3 \times \frac{k_{exp}k_{id}}{3}$ matrix and left-multiplied by $\boldsymbol{R}^i$. We assume that the product of the size of the latent variables of expression and identity $k_{exp} \times k_{id}$ is divisible by 3. After another round of vectorisation, the resulting feature $\in \mathbb{R}^{k_{exp}k_{id}}$ becomes the input of the decoders for normal and albedo. This transformation from feature vector $(\boldsymbol{Z}_{exp} \odot \boldsymbol{Z}_{id})^i$ to the rotated feature is called **rotation**.

5.2.5 Network Architecture

We incorporate the structure imposed by Equations (5.2), (5.4) and (5.5) into an auto-encoder network, see Figure 5.3. For some matrices $\boldsymbol{Y}_i \in \mathbb{R}^{k_{yi}\times n}$, we refer to the operation $\boldsymbol{Y}_1 \odot \boldsymbol{Y}_2 \in \mathbb{R}^{k_{y1}k_{y2}\times n}$ as **2-way synthesis** and $\boldsymbol{Y}_1 \odot \boldsymbol{Y}_2 \odot \boldsymbol{Y}_3 \in \mathbb{R}^{k_{y1}k_{y2}k_{y3}\times n}$ as **3-way synthesis**. The multiplication of a feature matrix by $\boldsymbol{B}_{(1)}$ or $\boldsymbol{Q}_{(1)}$, mode-1 matricisations of tensors $\mathcal{B}$ and $\mathcal{Q}$, is referred to as **projection** and can be represented by an unbiased fully-connected layer.

Our network follows the architecture of [Shu+17]. The encoder E receives an input image $\boldsymbol{x}$ and the convolutional encoder stack first encodes it into $\boldsymbol{z}_i$, an intermediate latent variable vector of size 128×1. $\boldsymbol{z}_i$ is then transformed into latent codes for background $\boldsymbol{z}_b$, mask $\boldsymbol{z}_m$, illumination $\boldsymbol{z}_l$, pose $\boldsymbol{z}_p$, identity $\boldsymbol{z}_{id}$ and expression $\boldsymbol{z}_{exp}$ via fully-connected layers.

$$E(\boldsymbol{x}) = [\boldsymbol{z}_b, \boldsymbol{z}_m, \boldsymbol{z}_l, \boldsymbol{z}_p, \boldsymbol{z}_{id}, \boldsymbol{z}_{exp}]^T. \tag{5.6}$$

The decoder D takes in the latent codes as input. $\boldsymbol{z}_b$ and $\boldsymbol{z}_m$ (128×1 vectors) are directly passed into convolutional decoder stacks to estimate background and face mask respectively. The remaining latent variables follow 3 streams:

1. $\boldsymbol{z}_{exp}$ (15×1 vector) and $\boldsymbol{z}_{id}$ (80×1 vector) are joined by 2-way synthesis and projection to estimate facial shape $\hat{\boldsymbol{x}_{3d}}$.

2. The result of 2-way synthesis of $\boldsymbol{z}_{exp}$ and $\boldsymbol{z}_{id}$ is rotated using $\boldsymbol{z}_p$. The rotated feature is passed into 2 different convolutional decoder stacks: one for normal estimation and another for albedo. Using the estimated normal map, albedo, illumination component $\boldsymbol{z}_l$, mask and background, we render a reconstructed image $\hat{\boldsymbol{x}}$.

3. $\boldsymbol{z}_{exp}$, $\boldsymbol{z}_{id}$ and $\boldsymbol{z}_l$ are combined by a 3-way synthesis and projection to estimate frontal normal map and a frontal reconstruction of the image.

Streams 1 and 3 drive the disentangling of expression and identity components, while stream 2 focuses on the reconstruction of the image by adding the pose components. The decoder D then outputs the reconstructed image from the latent codes.

$$D(\boldsymbol{z}_b, \boldsymbol{z}_m, \boldsymbol{z}_l, \boldsymbol{z}_p, \boldsymbol{z}_{id}, \boldsymbol{z}_{exp}) = \hat{\boldsymbol{x}}. \tag{5.7}$$

Our input images are aligned and cropped facial images from the CelebA database [Liu+15] of size 64×64, so $k_x = 3 \times 64 \times 64$. $k_{3d} = 3 \times 9375$, $k_l = 9$, $k_{id} = 80$ and $k_{exp} = 15$.

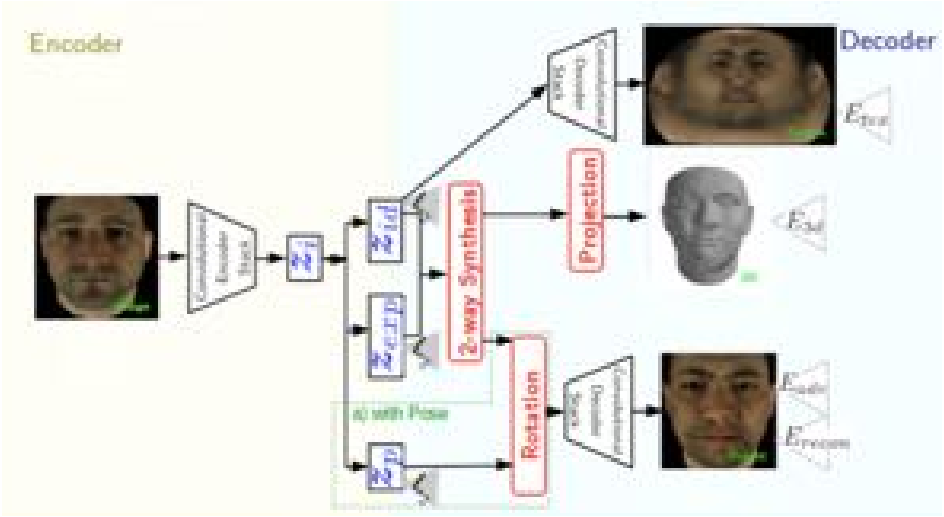


Figure 5.2: Our proof-of-concept network is an end-to-end trained auto-encoder. The encoder E extracts latent variables corresponding to expression and identity from the input image $\boldsymbol{x}$. These latent variables are then fed into the decoder D to reconstruct the image. A separate stream also reconstructs facial texture from $\boldsymbol{z}_{id}$. We impose a multilinear structure and enforce the disentanglement of variations. In the extended version a) the encoder also extracts a latent variable corresponding to pose. The decoder takes in this information and reconstructs an image containing pose variations.

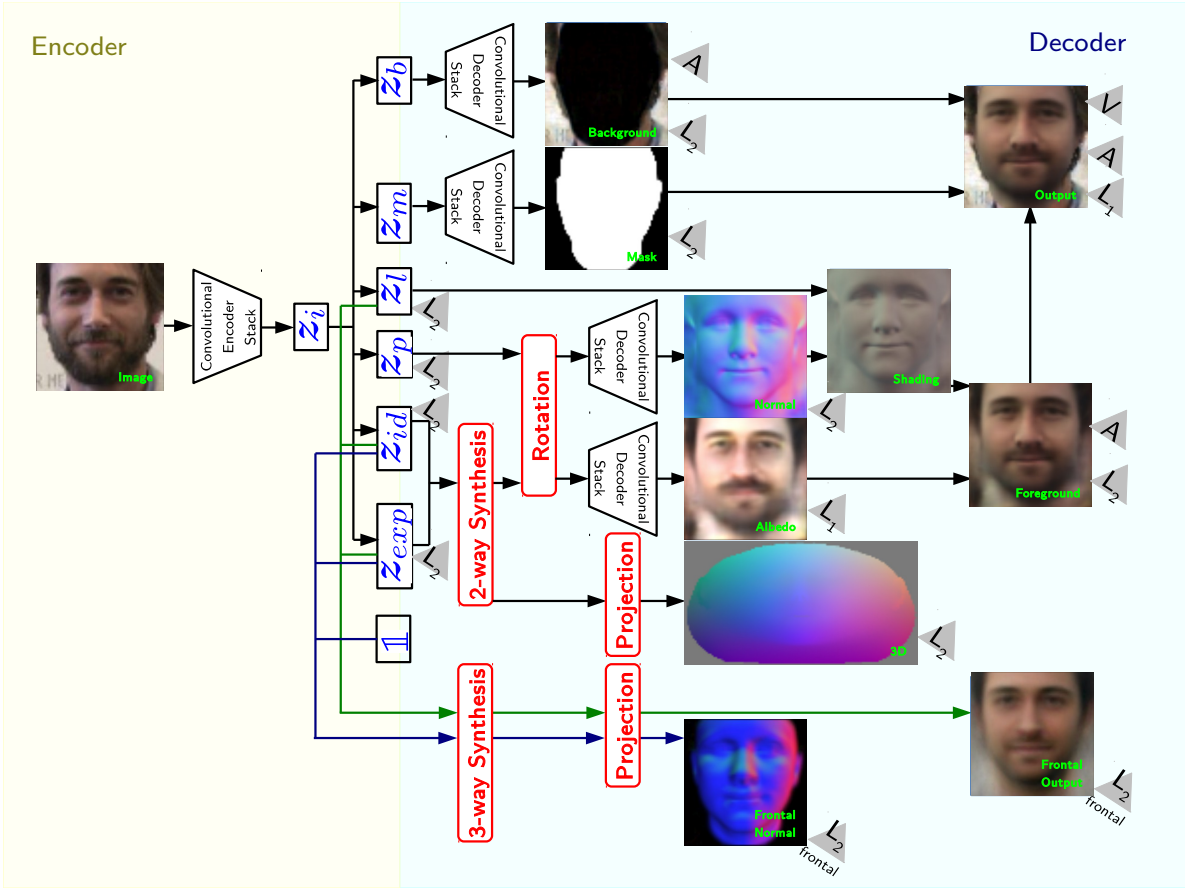


Figure 5.3: Our network is an end-to-end trained auto-encoder. The encoder E extracts latent variables corresponding to illumination, pose, expression and identity from the input image $\boldsymbol{x}$. These latent variables are then fed into the decoder D to reconstruct the image. We impose a multilinear structure and enforce the disentangling of variations. The grey triangles represent the losses: adversarial loss A, verification loss V, L_1 and L_2 losses.

5.2.6 Training

We use in-the-wild face images for training. Hence, we only have access to the image itself ($\boldsymbol{x}$) while ground truth labelling for pose, illumination, normal, albedo, expression, identity or 3D shape is unavailable. The main loss function is the reconstruction loss of the image x:

$$E_x = E_{recon} + \lambda_{adv} E_{adv} + \lambda_{veri} E_{veri}, \tag{5.8}$$

where $\hat{\boldsymbol{x}}$ is the reconstructed image, $E_{recon} = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_1$ is the reconstruction loss, λ_{adv} and λ_{adv} are regularisation weights, E_{adv} represents the adversarial loss and E_{veri} the verification loss. They are defined in the following. We use the pre-trained verification network $\mathcal{V}$ [WHS15] to find face embeddings of our images $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$. As both images are supposed to represent the same person, we minimise the cosine distance between the embeddings: $E_{veri} = 1 - cos(\mathcal{V}(\boldsymbol{x}), \mathcal{V}(\hat{\boldsymbol{x}}))$. Simultaneously, a discriminative network $\mathcal{D}$ is trained to distinguish between the generated and real images [Goo+14]. We incorporate the discriminative information by following the auto-encoder loss distribution matching approach of [BSM17]. The discriminative network $\mathcal{D}$ is itself an auto-encoder trying to reconstruct the input image $\boldsymbol{x}$ so the adversarial loss is $E_{adv} = \|\hat{\boldsymbol{x}} - \mathcal{D}(\hat{\boldsymbol{x}})\|_1$. $\mathcal{D}$ is trained to minimise $\|\boldsymbol{x} - \mathcal{D}(\boldsymbol{x})\|_1 - k_t\|\hat{\boldsymbol{x}} - \mathcal{D}(\hat{\boldsymbol{x}})\|_1$.

As fully unsupervised training often results in semantically meaningless latent representations, Shu et al. [Shu+17] proposed to train with pseudo ground truth values for normals, lighting and 3D facial shape. We adopt here this technique and introduce further pseudo ground truth values for pose $\hat{\boldsymbol{x}_p}$, expression $\hat{\boldsymbol{x}_{exp}}$ and identity $\hat{\boldsymbol{x}_{id}}$. $\hat{\boldsymbol{x}_p}$, $\hat{\boldsymbol{x}_{exp}}$ and $\hat{\boldsymbol{x}_{id}}$ are obtained by fitting coarse face geometry to every image in the training set using a 3D Morphable Model [Boo+17]. We incorporated the constraints used in [Shu+17] for illumination, normals and albedo. Hence, the following new objectives are introduced:

$$E_p = \|\boldsymbol{z}_p - \hat{\boldsymbol{x}_p}\|_2^2, \tag{5.9}$$

where $\hat{\boldsymbol{x}_p}$ is a 3D camera rotation matrix.

$$E_{exp} = \|fc(\boldsymbol{z}_{exp}) - \hat{\boldsymbol{x}_{exp}}\|_2^2, \tag{5.10}$$

where fc$(\cdot)$ is a fully-connected layer and $\hat{\boldsymbol{x}_{exp}} \in \mathbb{R}^{28}$ is a pseudo ground truth vector representing 3DMM expression components of the image $\boldsymbol{x}$.

$$E_{id} = \|fc(\boldsymbol{z}_{id}) - \hat{\boldsymbol{x}_{id}}\|_2^2 \tag{5.11}$$

where fc$(\cdot)$ is a fully-connected layer and $\hat{\boldsymbol{x}_{id}} \in \mathbb{R}^{157}$ is a pseudo ground truth vector representing 3DMM identity components of the image $\boldsymbol{x}$.

Multilinear Losses

Directly applying the above losses as constraints to the latent variables does not result in a well-disentangled representation. To achieve a better performance, we impose a tensor structure on the image using the following losses:

$$E_{3d} = \|\hat{\boldsymbol{x}_{3d}} - \mathcal{B} \times_2 \boldsymbol{z}_{exp} \times_3 \boldsymbol{z}_{id}\|_2^2, \tag{5.12}$$

where $\hat{\boldsymbol{x}_{3d}}$ is the 3D facial shape of the fitted model.

$$E_f = \|\boldsymbol{x}_f - \mathcal{Q} \times_2 \boldsymbol{z}_l \times_3 \boldsymbol{z}_{exp} \times_4 \boldsymbol{z}_{id})\|_2^2, \tag{5.13}$$

where $\boldsymbol{x}_f$ is a semi-frontal face image. During training, E_f is only applied on near-frontal face images filtered using $\hat{\boldsymbol{x}_p}$.

$$E_n = \|\hat{\boldsymbol{n}_f} - \mathcal{Q} \times_2 \frac{1}{k_l}\mathbf{1} \times_3 \boldsymbol{z}_{exp} \times_4 \boldsymbol{z}_{id})\|_2^2 \tag{5.14}$$

where $\hat{\boldsymbol{n}_f}$ is a near frontal normal map. During training, the loss E_n is only applied on near frontal normal maps. The tensor $\mathcal{Q}$ is the same as the one defined in 5.13.

The model is trained end-to-end by applying gradient descent to batches of images, where

Equations (5.12), (5.13) and (5.14) are written in the following general form:

$$E = \|\boldsymbol{X} - \boldsymbol{B}_{(1)}(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)})\|_F^2, \tag{5.15}$$

where M is the number of modes of variations, $\boldsymbol{X} \in \mathbb{R}^{k \times n}$ is a data matrix, $\boldsymbol{B}_{(1)}$ is the mode-1 matricisation of a tensor $\mathcal{B}$ and $\boldsymbol{Z}^{(i)} \in \mathbb{R}^{k_{zi} \times n}$ are the latent variables matrices.

The partial derivative of (5.15) with respect to the latent variable $\boldsymbol{Z}^{(i)}$ are computed as follows: Let $\hat{\boldsymbol{x}} = vec(\boldsymbol{X})$ be the vectorised $\boldsymbol{X}$, $\hat{\boldsymbol{z}}^{(i)} = vec(\boldsymbol{Z}^{(i)})$ be the vectorised $\boldsymbol{Z}^{(i)}$,

$\boldsymbol{Z}^{(\hat{i-1})} = \boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(i-1)}$ and $\boldsymbol{Z}^{(\hat{i+1})} = \boldsymbol{Z}^{(i+1)} \odot \cdots \odot \boldsymbol{Z}^{(M)}$, then (5.15) is equivalent with:

$$\begin{aligned} &\|\hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)})vec(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)})\|_F^2 \\ =&\|\hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)})(\boldsymbol{I} \odot \boldsymbol{Z}^{(\hat{i-1})}) \otimes \boldsymbol{I} \\ &\cdot \boldsymbol{I} \odot (\boldsymbol{Z}^{(\hat{i+1})}(\boldsymbol{I} \otimes \not\vdash)) \cdot \hat{\boldsymbol{z}^{(i)}}\|_2^2 \end{aligned} \tag{5.16}$$

Consequently the partial derivative of (5.15) with respect to $\boldsymbol{Z}^{(i)}$ is obtained by matricising the partial derivative of (5.16) with respect to $\boldsymbol{Z}^{(i)}$. The derivation details are in the subsequent section.

Derivation Details

The model is trained end-to-end by applying gradient descent to batches of images, where (5.12), (5.13) and (5.14) are written in the following general form:

$$E = \|\boldsymbol{X} - \boldsymbol{B}_{(1)}(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)})\|_F^2, \tag{5.15}$$

where $\boldsymbol{X} \in \mathbb{R}^{k \times n}$ is a data matrix, $\boldsymbol{B}_{(1)}$ is the mode-1 matricisation of a tensor $\mathcal{B}$ and $\boldsymbol{Z}^{(i)} \in \mathbb{R}^{k_{zi} \times n}$ are the latent variables matrices.

The partial derivative of (5.15) with respect to the latent variable $\boldsymbol{Z}^{(i)}$ are computed as follows: Let $\hat{\boldsymbol{x}} = vec(\boldsymbol{X})$ be a vectorisation of $\boldsymbol{X}$, then (5.15) is equivalent with:

$$\begin{aligned}&\|\boldsymbol{X} - \boldsymbol{B}_{(1)}(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)})\|_F^2 \\ =&\|vec(\boldsymbol{X} - \boldsymbol{B}_{(1)}(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)}))\|_2^2 \\ =&\|\hat{\boldsymbol{x}} - vec(\boldsymbol{B}_{(1)}(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)}))\|_2^2,\end{aligned} \tag{5.17}$$

as both the Frobenius norm and the L_2 norm are the sum of all elements squared.

$$\begin{aligned}&\|\hat{\boldsymbol{x}} - vec(\boldsymbol{B}_{(1)}(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)}))\|_2^2 \\ =&\|\hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)})vec(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)})\|_2^2,\end{aligned} \tag{5.18}$$

as the property $vec(\boldsymbol{BZ}) = (\boldsymbol{I} \otimes \boldsymbol{B})vec(\boldsymbol{Z})$ holds [Neu69].

Using $vec(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)}) = (\boldsymbol{I} \odot \boldsymbol{Z}^{(1)}) \otimes \boldsymbol{I} \cdot vec(\boldsymbol{Z}^{(2)})$ [Roe12] and let $\boldsymbol{Z}^{\hat{(i-1)}} = \boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(i-1)}$ and $\hat{\boldsymbol{Z}^{(i)}} = \boldsymbol{Z}^{(i)} \odot \cdots \odot \boldsymbol{Z}^{(M)}$ the following holds:

$$\begin{aligned}&\|\hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)})vec(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)})\|_2^2 \\ =&\|\hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)})(\boldsymbol{I} \odot \boldsymbol{Z}^{\hat{(i-1)}}) \otimes \boldsymbol{I} \cdot vec(\hat{\boldsymbol{Z}^{(i)}})\|_2^2\end{aligned} \tag{5.19}$$

Using $vec(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)}) = \boldsymbol{I} \odot (\boldsymbol{Z}^{(2)}(\boldsymbol{I} \otimes \nVdash)) \cdot vec(\boldsymbol{Z}^{(1)})$ [Roe12] and let $\boldsymbol{Z}^{\hat{(i+1)}} = \boldsymbol{Z}^{(i+1)} \odot \cdots \odot \boldsymbol{Z}^{(M)}$:

$$\begin{aligned}&\|\hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)})(\boldsymbol{I} \odot \boldsymbol{Z}^{\hat{(i-1)}}) \otimes \boldsymbol{I} \cdot vec(\hat{\boldsymbol{Z}^{(i)}})\|_2^2 \\ =&\|\hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)})(\boldsymbol{I} \odot \boldsymbol{Z}^{\hat{(i-1)}}) \otimes \boldsymbol{I} \\ &\cdot \boldsymbol{I} \odot (\boldsymbol{Z}^{\hat{(i+1)}}(\boldsymbol{I} \otimes \nVdash)) \cdot vec(\boldsymbol{Z}^{(i)})\|_2^2\end{aligned} \tag{5.20}$$

Let $\hat{\boldsymbol{z}}^{(i)} = vec(\boldsymbol{Z}^{(i)})$ be a vectorisation of $\boldsymbol{Z}^{(i)}$, this becomes:

$$\begin{aligned}&\|\hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)})(\boldsymbol{I} \odot \boldsymbol{Z}^{\hat{(i-1)}}) \otimes \boldsymbol{I} \\ &\cdot \boldsymbol{I} \odot (\boldsymbol{Z}^{\hat{(i+1)}}(\boldsymbol{I} \otimes \nVdash)) \cdot \hat{\boldsymbol{z}^{(i)}}\|_2^2\end{aligned} \tag{5.16}$$

We then compute the partial derivative of (5.16) with respect to $\hat{\boldsymbol{z}^{(i)}}$:

$$\frac{\partial \|\hat{\boldsymbol{x}} - \boldsymbol{A}\hat{\boldsymbol{z}^{(i)}}\|_2^2}{\partial \hat{\boldsymbol{z}^{(i)}}} = 2\boldsymbol{A}^T(\boldsymbol{A} \cdot \hat{\boldsymbol{z}^{(i)}} - \hat{\boldsymbol{x}}), \tag{5.21}$$

where $\boldsymbol{A} = (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)})(\boldsymbol{I} \odot \boldsymbol{Z}^{(\hat{i}-1)}) \otimes \boldsymbol{I} \cdot \boldsymbol{I} \odot (\boldsymbol{Z}^{(\hat{i}+1)}(\boldsymbol{I} \otimes \nVdash))$.

The partial derivative of (5.15) with respect to $\boldsymbol{Z}^{(i)}$ is obtained by matricising (5.21).

5.3 Proof of Concept Experiments

We develop a lighter version of our proposed network, a proof-of-concept network (visualised in Figure 5.2), to show that our network is able to learn and disentangle pose, expression and identity.

In order to showcase the ability of the network, we leverage our newly proposed 4DFAB database [Che+18b], where subjects were invited to attend four sessions at different times in a span of five years. In each experiment session, the subject was asked to articulate 6 different facial expressions (*anger, disgust, fear, happiness, sadness, surprise*), and we manually select the most expressive mesh (i.e. the apex frame) for this experiment. In total, 1795 facial meshes from 364 recording sessions (with 170 unique identities) are used. We keep 148 identities for training and leave 22 identities for testing. Note that there are no overlapping of identities between both sets. Within the training set, we synthetically augment each facial mesh by generating new facial meshes with 20 randomly selected expressions. Our training set contains in total 35900 meshes. The test set contains 387 meshes. For each mesh, we have the ground truth facial texture as well as expression and identity components of the 3DMM model.

5.3.1 Disentangling Expression and Identity

We create frontal images of the facial meshes. Hence there is no illumination or pose variation in this training dataset. We train a lighter version of our network by removing the illumination and pose streams, a proof-of-concept network, visualised in Figure 5.2, on this synthetic dataset.

Expression Editing

We show the disentanglement between expression and identity by transferring the expression of one person to another.

For this experiment, we work with unseen data (a hold-out set consisting of 22 unseen identities) and no labels. We first encode both input images $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$:

$$\begin{aligned} E(\boldsymbol{x}^i) &= \boldsymbol{z}^i_{exp}, \boldsymbol{z}^i_{id}, \\ E(\boldsymbol{x}^j) &= \boldsymbol{z}^j_{exp}, \boldsymbol{z}^j_{id}, \end{aligned} \tag{5.22}$$

where $E(\cdot)$ is our encoder and $\boldsymbol{z}_{exp}$ and $\boldsymbol{z}_{id}$ are the latent representations of expression and identity respectively.

Assuming we want $\boldsymbol{x}^i$ to emulate the expression of $\boldsymbol{x}^j$, we decode on:

$$D(\boldsymbol{z}^j_{exp}, \boldsymbol{z}^i_{id}) = \boldsymbol{x}^{ji}, \tag{5.23}$$

where $D(\cdot)$ is our decoder. The resulting $\boldsymbol{x}^{ji}$ becomes our edited image where $\boldsymbol{x}^i$ has the expression of $\boldsymbol{x}^j$. Figure 5.4 shows how the network is able to separate expression and identity. The edited images clearly maintain the identity while expression changes.

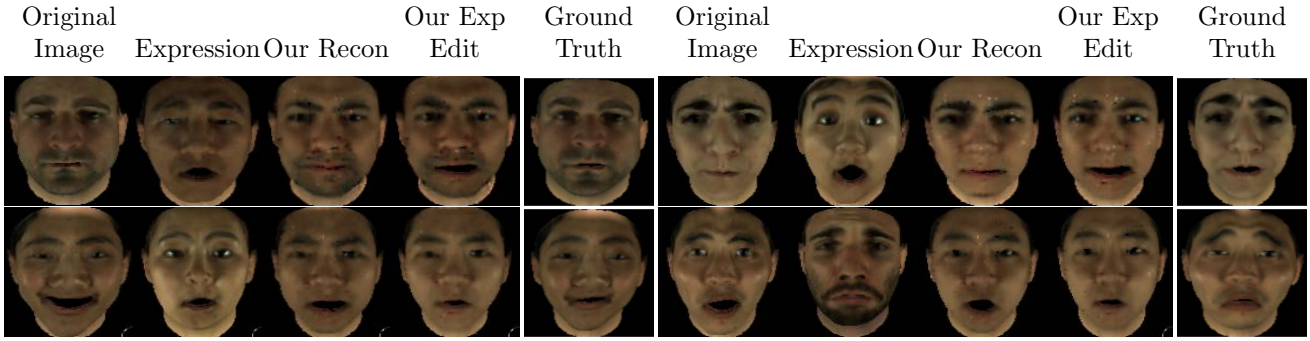


Figure 5.4: Our network is able to transfer the expression from one face to another by disentangling the expression components of the images. The ground truth has been computed using the ground truth texture with synthetic identity and expression components.

3D Reconstruction and Facial Texture

The latent variables $\boldsymbol{z}_{exp}$ and $\boldsymbol{z}_{id}$ that our network learns are extremely meaningful. Not only can they be used to reconstruct the image in 2D, but also they can be mapped into the expression ($\boldsymbol{x}_{exp}$) and identity ($\boldsymbol{x}_{id}$) components of a 3DMM model. This mapping is learnt inside the network. By replacing the expression and identity components of a mean face shape with $\hat{\boldsymbol{x}_{exp}}$ and $\hat{\boldsymbol{x}_{id}}$, we are able to reconstruct the 3D mesh of a face given a single input image. We compare these reconstructed meshes against the ground truth 3DMM used to create the input image in Figure 5.5.

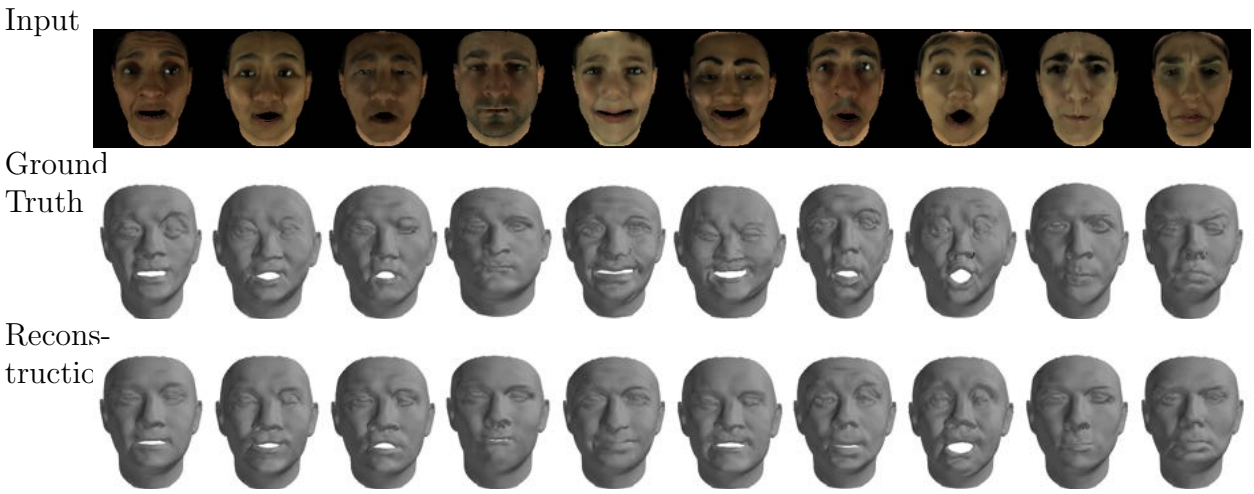


Figure 5.5: Given a single image, we infer meaningful expression and identity components to reconstruct a 3D mesh of the face. We compare the reconstruction (last row) against the ground truth (2^{nd} row).

At the same time, the network is able to learn a mapping from $\boldsymbol{z}_{id}$ to facial texture. Therefore, we can predict the facial texture given a single input image. We compare the reconstructed facial texture with the ground truth facial texture in Figure 5.6.

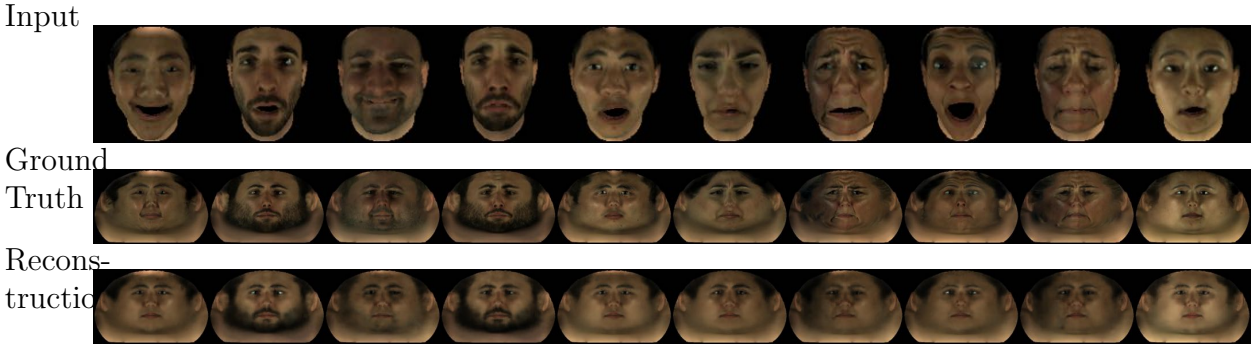


Figure 5.6: Given a single image, we infer the facial texture. We compare the reconstructed facial texture (last row) against the ground truth texture (2^{nd} row).

5.3.2 Disentangling Pose, Expression and Identity

Our synthetic training set contains in total 35900 meshes. For each mesh, we have the ground truth facial texture as well as expression and identity components of the 3DMM, from which we create a corresponding image with one of 7 given poses. As there is no illumination variation in this training set, we train a proof-of-concept network by removing the illumination stream, visualised in Figure 5.2a, on this synthetic dataset.

Pose Editing

We show the disentanglement between pose, expression and identity by transferring the pose of one person to another. Figure 5.7 shows how the network is able to separate pose from expression and identity. This experiment highlights the ability of our proposed network to learn large pose variations even from profile to frontal faces.

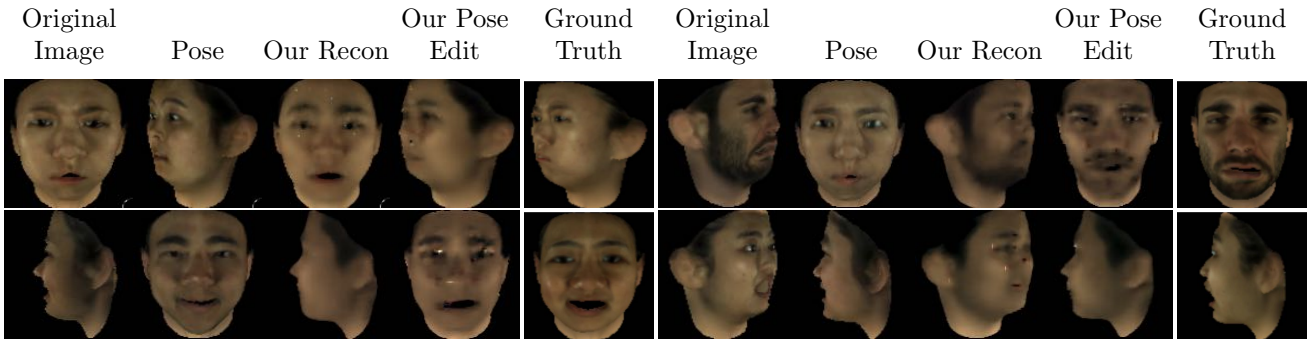


Figure 5.7: Our network is able to transfer the pose from one face to another by disentangling the pose, expression and identity components of the images. The ground truth has been computed using the ground truth texture with synthetic pose, identity and expression components.

5.4 Experiments in-the-wild

We train our network on in-the-wild data and perform several experiments on unseen data to show that our network is indeed able to disentangle illumination, pose, expression and identity.

We edit expression or pose by swapping the latent expression/pose component learnt by the

encoder E (Eq. (5.6)) with the latent expression/pose component predicted from another image. We feed the decoder D (Eq. (5.7)) with the modified latent component to retrieve our edited image.

5.4.1 Expression, Pose and Identity Editing in-the-wild

Given two in-the-wild images of faces, we are able to transfer the expression, pose of one person to another. We are also able to swap the face of the person from one image to another. Transferring the expression from two different facial images without fitting a 3D model is a very challenging problem. Generally, it is considered in the context of the same person under an elaborate blending framework [Yan+11] or by transferring certain classes of expressions [Sag+17].

For this experiment, we work with completely unseen data (a hold-out set of CelebA) and no labels. We first encode both input images $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$:

$$\begin{aligned} E(\boldsymbol{x}^i) &= \boldsymbol{z}^i_{exp}, \boldsymbol{z}^i_{id}, \boldsymbol{z}^i_p \\ E(\boldsymbol{x}^j) &= \boldsymbol{z}^j_{exp}, \boldsymbol{z}^j_{id}, \boldsymbol{z}^j_p, \end{aligned} \tag{5.24}$$

where $E(\cdot)$ is our encoder and $\boldsymbol{z}_{exp}$, $\boldsymbol{z}_{id}$, $\boldsymbol{z}_p$ are the latent representations of expression, identity and pose respectively.

Assuming we want $\boldsymbol{x}^i$ to take on the expression, pose or identity of $\boldsymbol{x}^j$, we then decode on:

$$\begin{aligned} D(\boldsymbol{z}^j_{exp}, \boldsymbol{z}^i_{id}, \boldsymbol{z}^i_p) &= \boldsymbol{x}^{jii} \\ D(\boldsymbol{z}^i_{exp}, \boldsymbol{z}^i_{id}, \boldsymbol{z}^j_p) &= \boldsymbol{x}^{iij} \\ D(\boldsymbol{z}^i_{exp}, \boldsymbol{z}^j_{id}, \boldsymbol{z}^i_p) &= \boldsymbol{x}^{iji} \end{aligned} \tag{5.25}$$

where $D(\cdot)$ is our decoder.

The resulting $\boldsymbol{x}^{jii}$ then becomes our result image where $\boldsymbol{x}^i$ has the expression of $\boldsymbol{x}^j$. $\boldsymbol{x}^{jii}$ is the edited image where $\boldsymbol{x}^i$ changed to the pose of $\boldsymbol{x}^j$. $\boldsymbol{x}^{iji}$ is the edit where $\boldsymbol{x}^i$'s face changed to

the face of $\boldsymbol{x}^j$.

As there is currently no prior work for this expression editing experiment without fitting an AAM [CET01] or 3DMM, we used the image synthesised by the 3DMM fitted models as a baseline, which indeed performs quite well. Compared with our method, other very closely related works [Wan+17b; Shu+17] are not able to disentangle illumination, pose, expression and identity. In particular, [Shu+17] disentangles illumination of an image while [Wan+17b] disentangles illumination, expression and identity from "frontalised" images. Hence they are not able to disentangle pose. None of these methods can be applied to the expression/pose editing experiments on a dataset that contains pose variations such as CelebA. If [Wan+17b] is applied directly on our test images, it would not be able to perform expression editing well, as shown by Figure 5.9.

For the 3DMM baseline, we fit a shape model to both images and extract the expression components of the model. This fitting step has high overhead of 20 seconds per image. We then generate a new face shape using the expression components of one face and the identity components of another face in the same 3DMM setting. This technique has much higher overhead than our proposed method as it requires time-consuming 3DMM fitting of the images. Our expression editing results and the baseline results are shown in Figure 5.8. Though the baseline is very strong, it does not change the texture of the face which can produce unnatural looking faces shown with original expression. Also, the baseline method can not fill up the inner mouth area. Our editing results show more natural looking faces.

For pose editing, the background is unknown once the pose has changed, thus, for this experiment, we mainly focus on the face region. Figure 5.10 shows our pose editing results. For the baseline method, we fit a 3DMM to both images and estimate the rotation matrix. We then synthesise $\boldsymbol{x}_i$ with the rotation of $\boldsymbol{x}_j$. This technique has high overhead as it requires expensive 3DMM fitting of the images.

Figure 5.11 shows our results on the task of face swapping where the identity of one image has been swapped with the face of another person from the second image.

Figure 5.8: Our network is able to transfer the expression from one face to another by disentangling the expression components of the images. We compare our expression editing results with a baseline where a 3DMM has been fit to both input images.

Quantitative Studies

We conducted a quantitative measure on the expression editing experiment. We ran a face recognition experiment on 50 pairs of images where only the expression has been transferred. We then passed them to a face recognition network [DGZ18] and extracted their respective embeddings. All 50 pairs of embeddings had cosine similarity larger than 0.3. In comparison, We selected 600 pairs of different people from CelebA and computed their average cosine similarity which is 0.062. The histogram of these cosine similarities is visualised in Figure 5.14. This indicates that the expression editing does conserve identity in terms of machine perception.

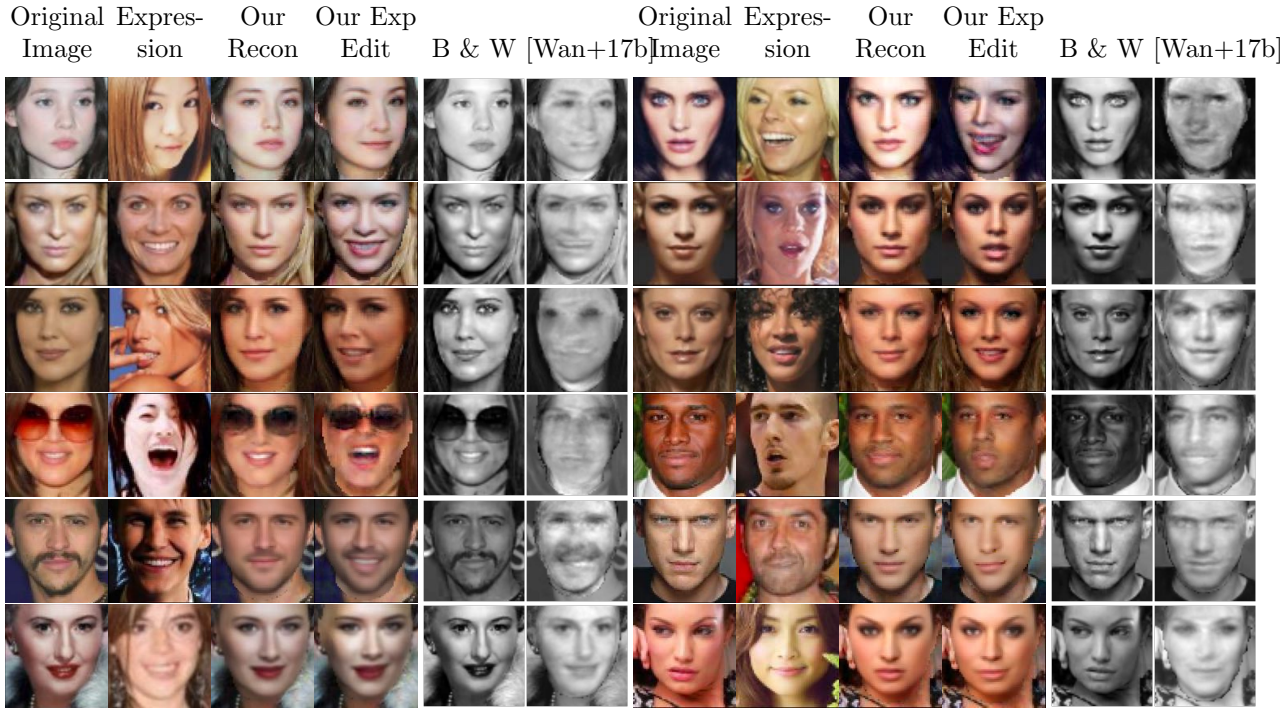


Figure 5.9: We compare our expression editing results with [Wan+17b]. As [Wan+17b] is not able to disentangle pose, editing expressions from images of different poses returns noisy results.

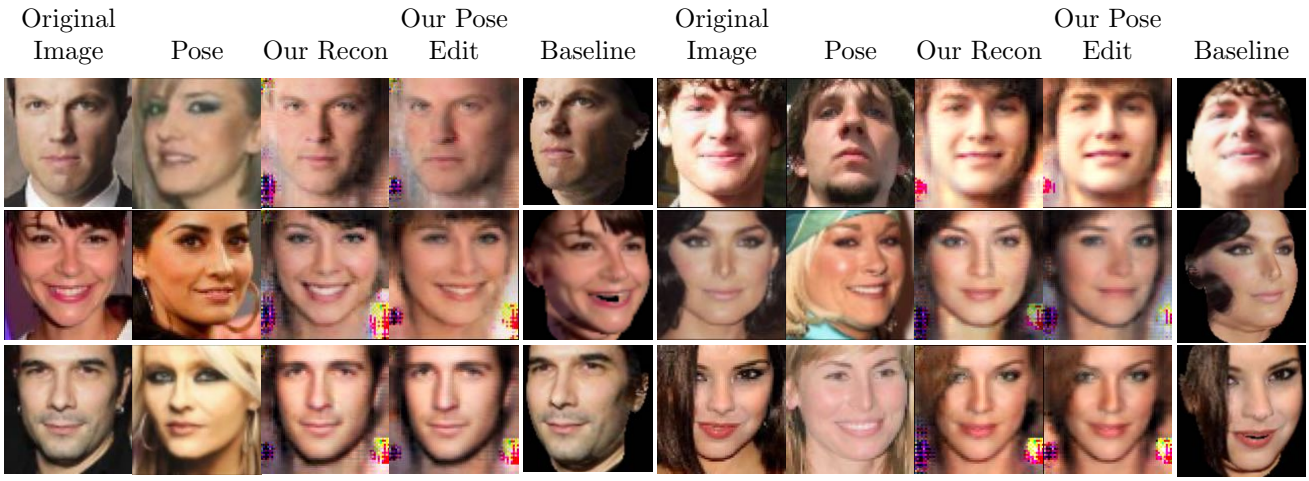


Figure 5.10: Our network is able to transfer the pose of one face to another by disentangling the pose components of the images. We compare our pose editing results with a baseline where a 3DMM has been fit to both input images.

Ablation Studies

We performed a series of ablation studies. We first trained a network without multilinear losses by simply feeding the concatenated parameters $\boldsymbol{p} = [\boldsymbol{z}_{pose}, \boldsymbol{z}_{exp}, \boldsymbol{z}_{id}]$ to the decoder, thus the training of the network is only driven by the reconstruction loss and pseudo-supervision from 3DMM on pose, expression and identity latent variables, i.e., $\boldsymbol{z}_{pose}, \boldsymbol{z}_{exp}$ and $\boldsymbol{z}_{id}$. Next, we

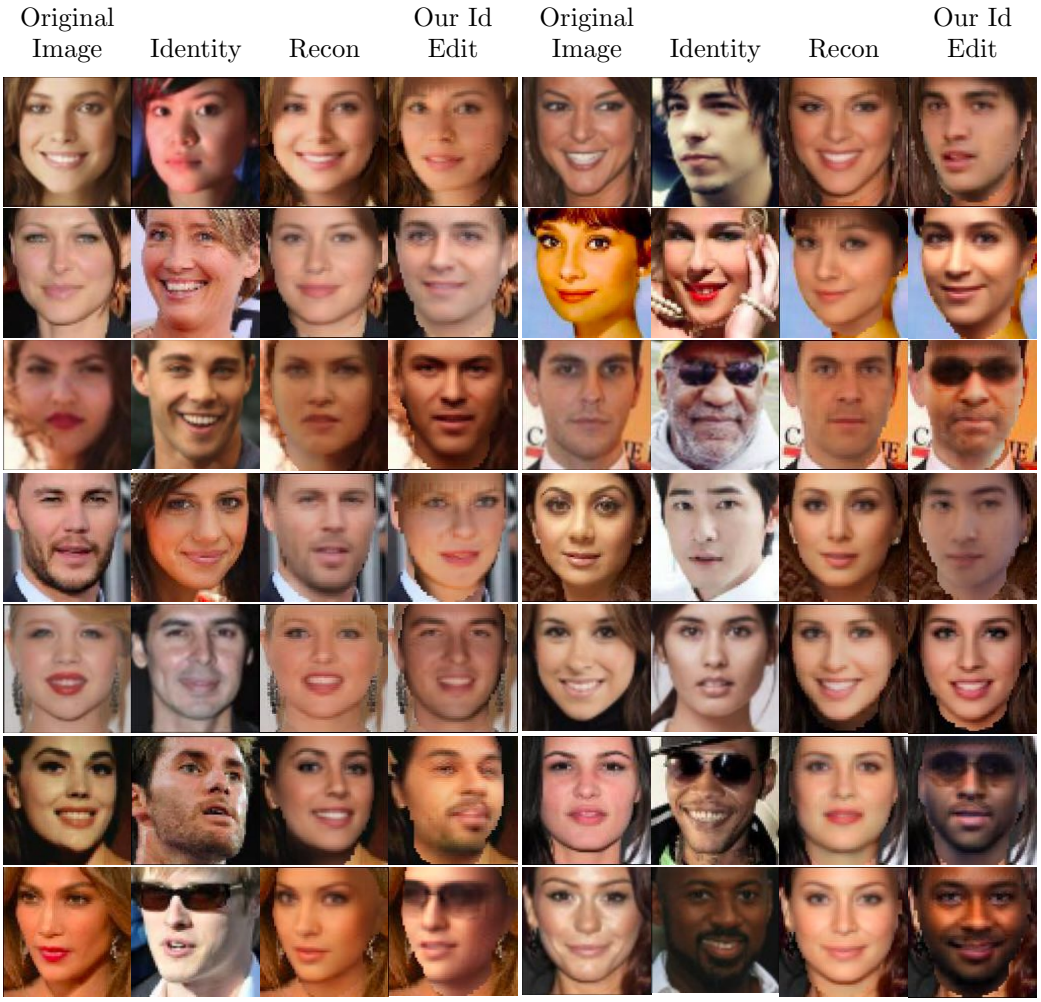


Figure 5.11: Our network is also able to transfer the identity of one image to another by disentangling the identity components of the images.

started to incorporate other losses (i.e., multilinear losses, adversarial loss, verification loss) step by step in the network and trained different models. In this way, we can observe at each step how additional loss may improve the result.

In Figure 5.12 and 5.13, we compare the expression and pose editing results. We find that the results without multilinear losses shows some entanglement of the variations in terms of illumination, identity, expression and pose. In particular, the entanglement with illumination is strong, examples can be found in second and ninth row of Figure 5.12. Indeed, by incorporating multilinear losses in the network, the identity and expression variations are better disentangled. Furthermore, the incorporation of adversarial and verification losses enhances the quality of images, making them look more realistic but do not contribute in a meaningful way to the disentanglement.

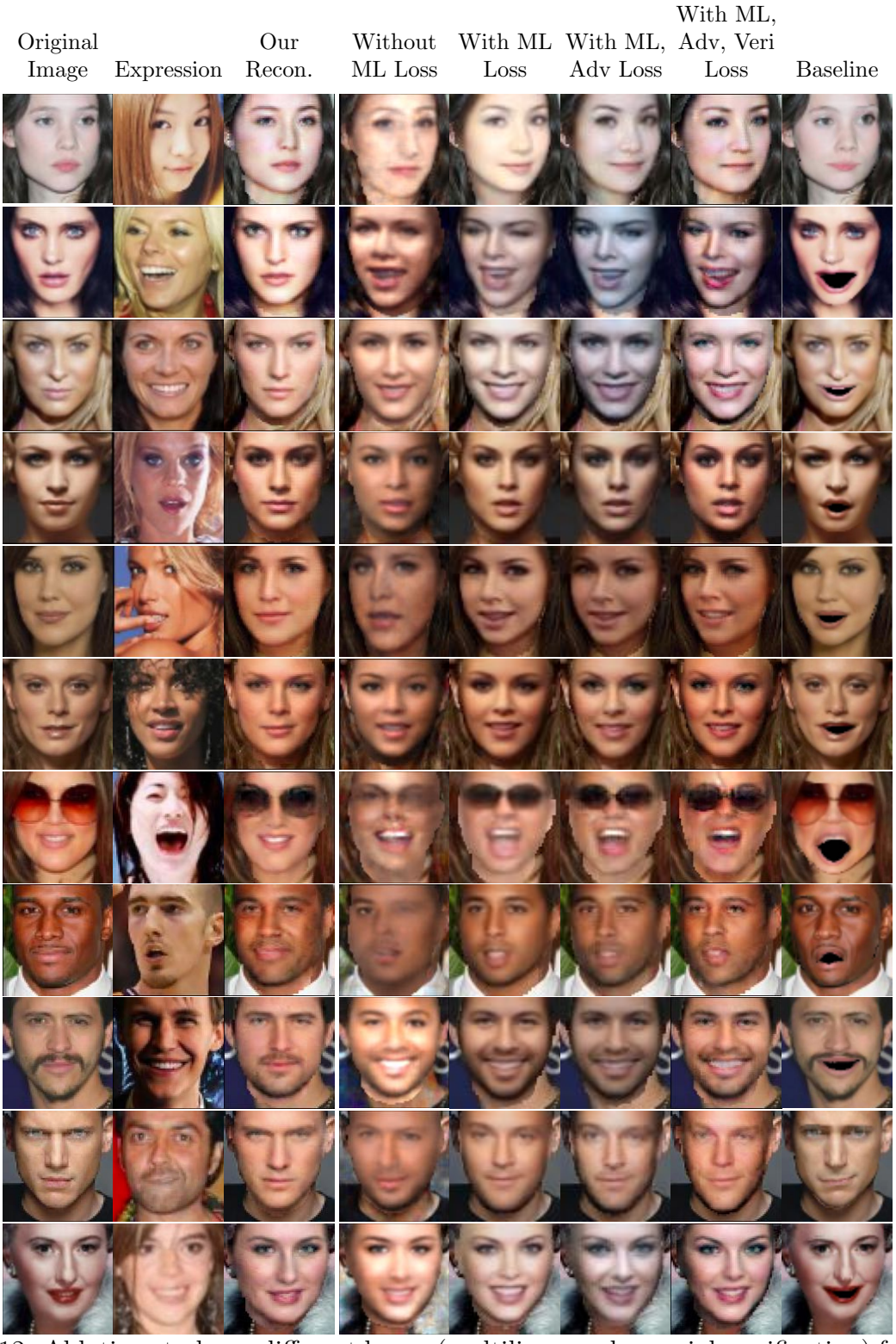


Figure 5.12: Ablation study on different losses (multilinear, adversarial, verification) for expression editing. The results show that incorporating multilinear losses indeed helps the network to better disentangle the expression variations.

Discussion on Texture Quality

It has to be noted that our baseline 3DMM method [Boo+17] does not change facial texture. It directly samples the original texture and maps it to a 3D face. Hence, the texture quality is

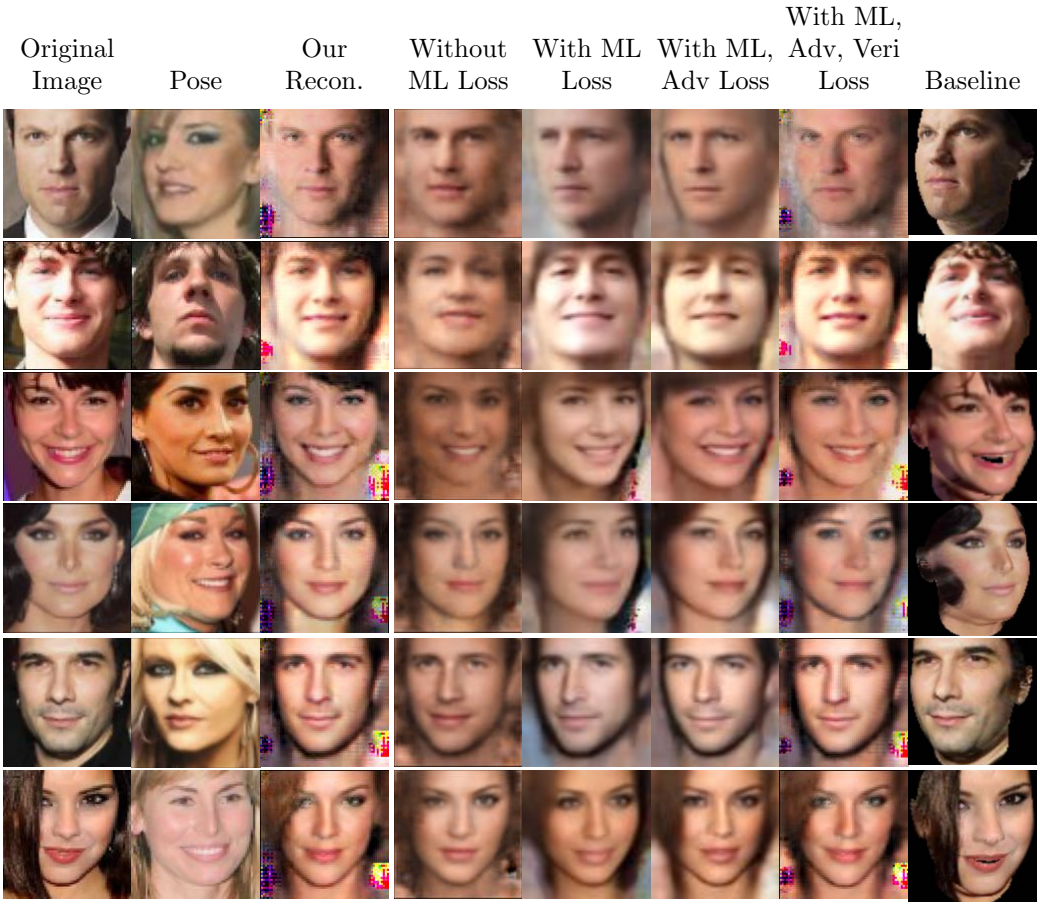


Figure 5.13: Ablation study on different losses (multilinear, adversarial, verification) for facial pose editing. The results show that incorporating multilinear losses helps the network to better disentangle the pose variations.

exactly the same as that of the original image as no low-dimensional texture representation is used. In terms of texture quality, direct texture mapping has an edge over our proposed method which models the texture using a low-dimensional representation. But direct texture mapping is also prone to artefacts and does not learn the new expression in the texture. Looking at Figure 5.8 column 2, rows 4, 5 and 7, we observe that the texture itself did not change in the baseline result. The eyes and cheeks did not adjust to show a smiling or neutral face. The expression change results from the change in the 3D shape but the texture itself remained the same as in the input. Low-dimensional texture representation does not have this issue and can generate new texture with changed expression.

Generally methods similar to ours which estimate facial texture is not able to extract the same amount of details as the original image. Figure 5.15 visualises how our texture reconstruction compares to state-of-the-art works which have been trained on images of higher resolutions.

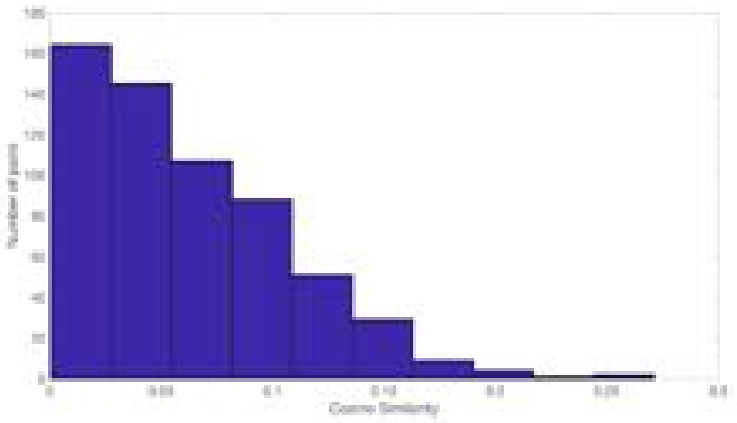


Figure 5.14: Histogram of cosine similarities on 600 pairs of "non-same" people from CelebA

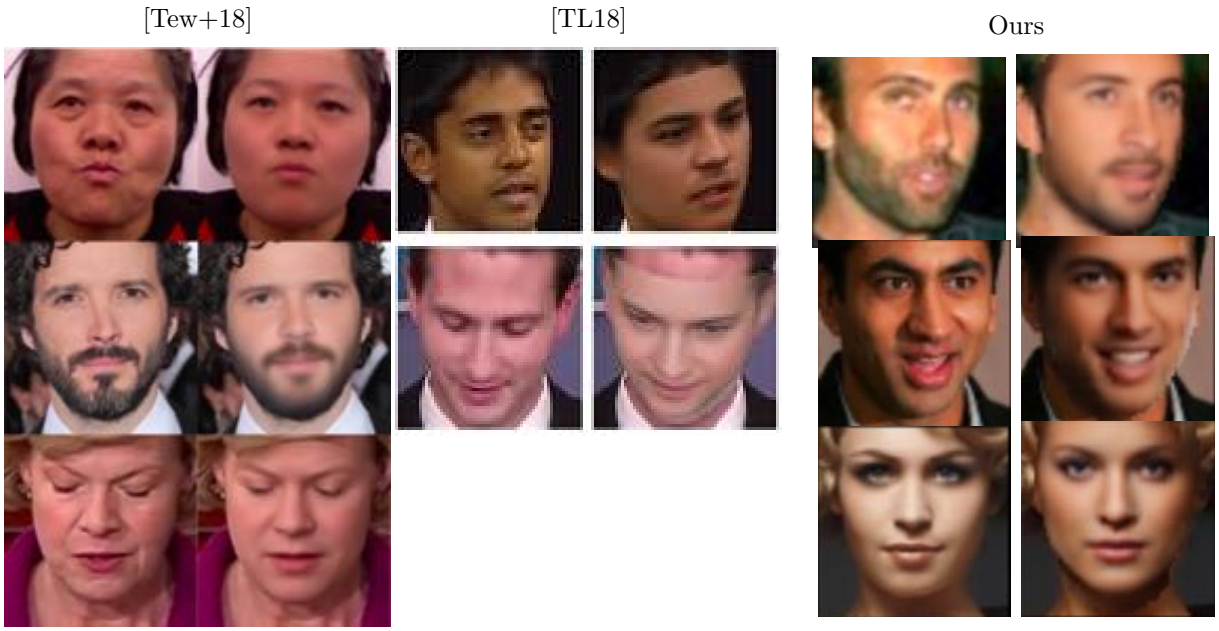


Figure 5.15: Texture reconstruction compared with [Tew+18; TL18]. [Tew+18; TL18] have been trained with images of higher resolutions of 240×240 and 128×128 respectively. In comparison our model has only been trained with images of size 64×64 pixels.

Figure 5.16: We show how to interpolate the expression latent code from the rightmost person to the expression on the leftmost person. As the interpolation is only done on the expression code, the identity remains the identity of the rightmost person.

Figure 5.17: We show how to interpolate the identity latent code from the rightmost person to the leftmost person. As the interpolation is only done on the identity code, the expression remains the one of the rightmost person.

5.4.2 Expression and Identity Interpolation

We interpolate $\boldsymbol{z}^i_{exp}$ / $\boldsymbol{z}^i_{id}$ of the input image $\boldsymbol{x}^i$ on the right-hand side to the $\boldsymbol{z}^t_{exp}$ / $\boldsymbol{z}^t_{id}$ of the target image $\boldsymbol{x}^t$ on the left-hand side. The interpolation is linear and at 0.1 interval. For the interpolation we do not modify the background so the background remains that of image $\boldsymbol{x}^i$.

For expression interpolation, we expect the identity and pose to stay the same as the input image $\boldsymbol{x}^i$ and only the expression to change gradually from the expression of the input image to the expression of the target image $\boldsymbol{x}^t$. Figure 5.16 shows the expression interpolation. We can clearly see the change in expression while pose and identity remain constant.

For identity interpolation, we expect the expression and pose to stay the same as the input image $\boldsymbol{x}^i$ and only the identity to change gradually from the identity of the input image to the identity of the target image $\boldsymbol{x}^t$. Figure 5.17 shows the identity interpolation. We can clearly observe the change in identity while other variations remain limited.

5.4.3 Illumination Editing

We transfer illumination by estimating the normals $\hat{\boldsymbol{n}}$, albedo $\hat{\boldsymbol{a}}$ and illumination components $\hat{\boldsymbol{l}}$ of the source ($\boldsymbol{x}^{source}$) and target ($\boldsymbol{x}^{target}$) images. Then we use $\hat{\boldsymbol{n}}^{target}$ and $\hat{\boldsymbol{l}}^{source}$ to compute the transferred shading $\boldsymbol{s}^{transfer}$ and multiply the new shading by $\hat{\boldsymbol{a}}^{target}$ to create the relighted image result $\boldsymbol{x}^{transfer}$. In Figure 5.18 we show the performance of our method and compare against [Shu+17] on illumination transfer. We observe that our method outperforms [Shu+17] as we obtain more realistic looking results.

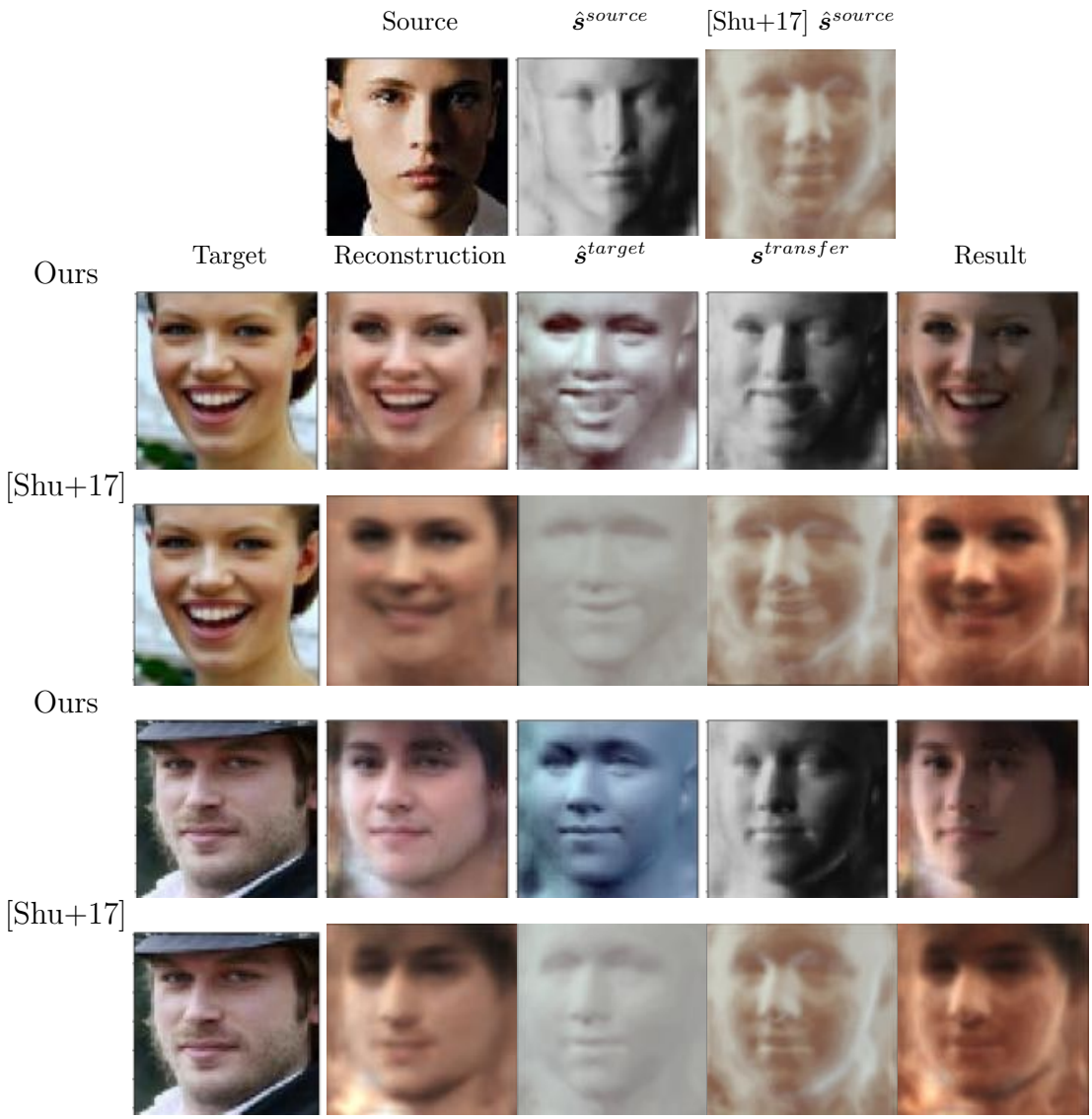


Figure 5.18: Using the illumination and normals estimated by our network, we are able to relight target faces using illumination from the source image. The source $\hat{\boldsymbol{s}}^{source}$ and target shading $\hat{\boldsymbol{s}}^{target}$ are displayed to visualise against the new transferred shading $\boldsymbol{s}^{transfer}$. We compare against [Shu+17].

5.4.4 3D Reconstruction

The latent variables $\boldsymbol{z}_{exp}$ and $\boldsymbol{z}_{id}$ that our network learns are extremely meaningful. Not only can they be used to reconstruct the image in 2D, they can be mapped into the expression ($\boldsymbol{x}_{exp}$) and identity ($\boldsymbol{x}_{id}$) components of a 3DMM. This mapping is learnt inside the network. By replacing the expression and identity components of a mean face shape with $\hat{\boldsymbol{x}_{exp}}$ and $\hat{\boldsymbol{x}_{id}}$, we are able to reconstruct the 3D mesh of a face given a single in-the-wild 2D image. We compare these reconstructed meshes against the fitted 3DMM to the input image.

The results of the experiment are visualised in Figure 5.19. We observe that the reconstruction is comparable to other state-of-the-art techniques [Jac+17; Fen+18]. None of the techniques though capture well the identity of the person in the input image due to a known weakness in 3DMM.

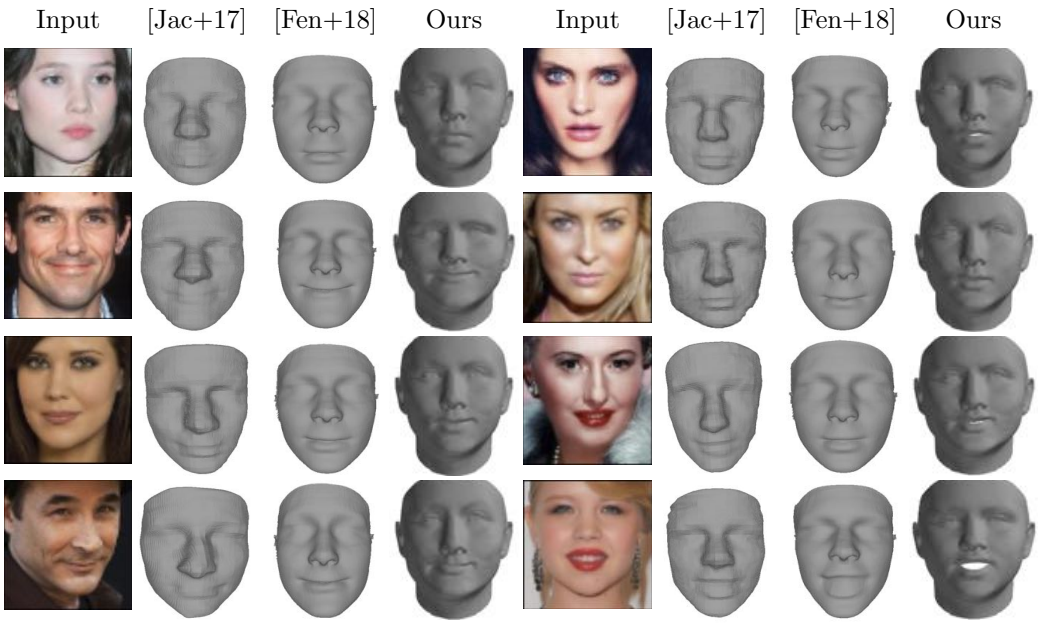


Figure 5.19: Given a single image, we infer meaningful expression and identity components to reconstruct a 3D mesh of the face. We compare our 3D estimation against recent works [Jac+17; Fen+18].

Method	**Mean±Std against [Woo80]**	**<35°**	**<40°**
[Wan+17b]	33.37°± 3.29°	75.3%	96.3%
[Shu+17]	30.09°± 4.66°	84.6%	98.1%
Proposed	**28.67°± 5.79°**	**89.1%**	**96.3%**

Table 5.1: Angular error for the various surface normal estimation methods on the Photoface [Zaf+13] dataset. We also show the proportion of the normals below 35° and 40°.

Features	Identity	Expression	Pose
SIFT and Visual Bag of Words, K=50	14.60%	58.33%	55.50%
SIFT and Visual Bag of Words, K=100	18.71%	59.36%	59.46%
Standard CNN Model	94.68%	96.54%	98.78%
Ours ($\boldsymbol{z}_{identity}$, $\boldsymbol{z}_{expression}$, $\boldsymbol{z}_{pose}$)	88.29%	84.85%	95.55%

Table 5.2: Classification accuracy results: we try to classify 54 identities using $\boldsymbol{z}_{id}$, 6 expressions using $\boldsymbol{z}_{exp}$ and 7 poses using $\boldsymbol{z}_{p}$. We compare against standard baseline methods such as SIFT and CNN.

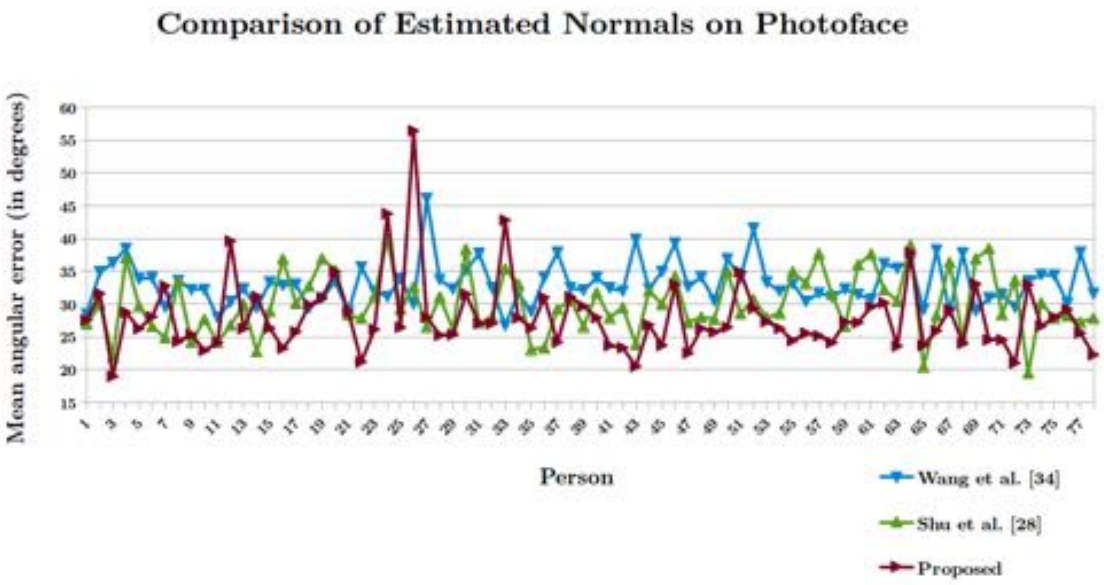


Figure 5.20: Comparison of the estimated normals obtained using the proposed model vs the ones obtained by [Wan+17b] and [Shu+17] on the Photoface dataset. The proposed method outperforms the baselines on the majority of cases.

5.4.5 Normal Estimation

We evaluate our method on the surface normal estimation task on the Photoface [Zaf+13] dataset which has information about illumination. Assuming the normals found using calibrated Photometric Stereo [Woo80] as "ground truth", we calculate the angular error between our estimated normals and the "ground truth". Figure 5.20 and Table 5.1 quantitatively evaluates our proposed method against prior works [Wan+17b; Shu+17] in the normal estimation task. We observe that our proposed method performs on par or outperforms previous methods.

5.4.6 Quantitative Evaluation of the Latent Space

We want to test whether our latent space corresponds well to the variation that it is supposed to learn. For our quantitative experiment, we used Multi-PIE [Gro+10] as our test dataset. This dataset contains labelled variations in identity, expressions and pose. Disentanglement of variations in Multi-PIE is particularly challenging as its images are captured under laboratory conditions which is quite different from that of our training images. As a matter of fact, the expressions contained in Multi-PIE do not correspond to the 7 basic expressions and can be easily confused.

Features	Identity
Without verification loss	87.94%
Ours ($\boldsymbol{z_{identity}}$)	**88.29%**
Without verification loss (frontal only)	99.96 %
Ours ($\boldsymbol{z_{identity}}$, frontal only)	**99.98%**

Table 5.3: Identity classification accuracy results: we classify 54 identities using $\boldsymbol{z}_{id}$ with and without verification loss.

We encoded 10368 images of the Multi-PIE dataset with 54 identities, 6 expressions and 7 poses and trained a linear SVM classifier using 90% of the identity labels and the latent variables $\boldsymbol{z}_{id}$. We then test on the remaining 10% $\boldsymbol{z}_{id}$ to check whether they are discriminative for identity classification. We use 10-fold cross-validation to evaluate the accuracy of the learnt classifier. We repeat this experiment for expression with $\boldsymbol{z}_{exp}$ and pose with $\boldsymbol{z}_{p}$ respectively. Our results in Table 5.2 show that our latent representation is indeed discriminative. We compare against some standard baselines such as Bag-of-Words (BoWs) models with SIFT feature [SZ09] and standard CNN. Our model does not outperform the standard CNN model, which is fully supervised and requires a separate model for each variation classification. Still our results are a strong indication that the latent representation found is discriminative. This experiment showcases the discriminative power of our latent representation on a previously unseen dataset.

As an ablation study, we test the accuracy of the identity classification of $\boldsymbol{z}_{id}$ from a model trained without the verification. The results in Table 5.3 show that though adding the verification loss improves the performance, the gain is not significant enough to prove that this loss is a substantial contributor of the information.

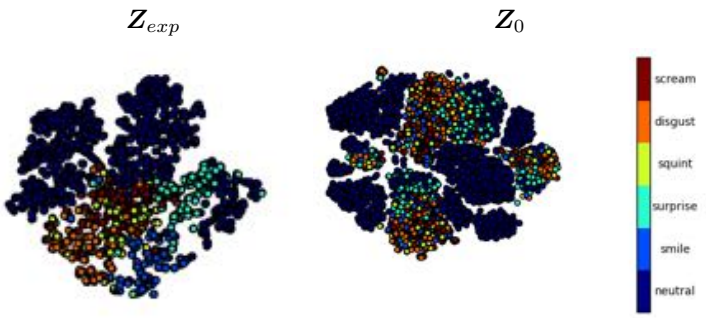


Figure 5.21: Visualisation of our $\boldsymbol{Z}_{exp}$ and baseline $\boldsymbol{Z}_0$ using t-SNE. Our latent $\boldsymbol{Z}_{exp}$ clusters better with regards to expression than the latent space $\boldsymbol{Z}_0$ of an auto-encoder.

In order to quantitatively compare with [Wan+17b], we run another experiment on only frontal images of the dataset with 54 identities, 6 expressions and 16 illuminations. The results in Table 5.4 shows how our proposed model outperforms [Wan+17b] in these classification tasks. Our latent representation has stronger discriminative power than the one learnt by [Wan+17b].

Identity

	$\boldsymbol{z_{identity}}$	$\boldsymbol{C}$ [Wan+17b]
Accuracy	**99.33%**	19.18 %

Expression

	$\boldsymbol{z_{expression}}$	$\boldsymbol{E}$ [Wan+17b]
Accuracy	**78.92%**	35.49

Illumination

	$\boldsymbol{z_{illumination}}$	$\boldsymbol{L}$ [Wan+17b]
Accuracy	**64.11%**	48.85%

Table 5.4: Classification accuracy results in comparison with [Wan+17b]: As [Wan+17b] works on frontal images, we only consider frontal images in this experiment. We try to classify 54 identities using $\boldsymbol{z}_{id}$ vs. $\boldsymbol{C}$, 6 expressions using $\boldsymbol{z}_{exp}$ vs. $\boldsymbol{E}$ and 16 illumination using $\boldsymbol{z}_{ill}$ vs. $\boldsymbol{L}$.

We visualise, using t-SNE [MH08], the latent $\boldsymbol{Z}_{exp}$ and $\boldsymbol{Z}_p$ encoded from Multi-PIE according to their expression and pose label and compare against the latent representation $\boldsymbol{Z}_0$ learnt by an in-house large-scale adversarial auto-encoder of similar architecture trained with 2 million faces [Mak+15]. Figures 5.21 and 5.22 show that even though our encoder has not seen any images of Multi-PIE, it manages to create informative latent representations that cluster well expression and pose (contrary to the representation learned by the tested auto-encoder).

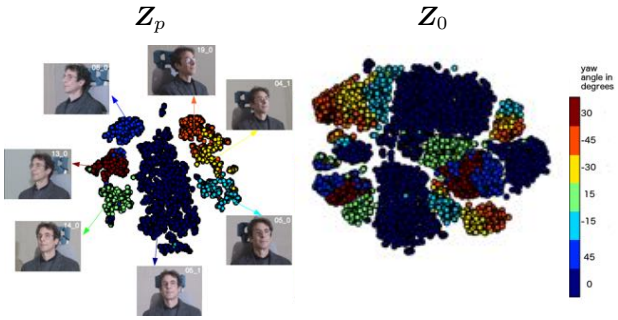


Figure 5.22: Visualisation of our $\boldsymbol{Z}_p$ and baseline $\boldsymbol{Z}_0$ using t-SNE. It is evident that the proposed disentangled $\boldsymbol{Z}_p$ clusters better with regards to pose than the latent space $\boldsymbol{Z}_0$ of an auto-encoder.

5.5 Limitations

Some of our results do still show entanglement in the variations. Sometimes despite only aiming to change expression only, pose or illumination have been modified as well. This happens mainly in very challenging scenarios where for example one of the image shows extreme lighting conditions, is itself black and white or displays large pose variations. Due to the dataset (CelebA) we used, we do struggle with large pose variations. The proof of concept experiments do show that this is possible to be learned with a more balanced dataset.

5.6 Conclusions

We proposed the first, to the best of our knowledge, attempt to jointly disentangle modes of variation that correspond to expression, identity, illumination and pose using no explicit labels regarding these attributes. More specifically, we proposed the first, as far as we know, approach that combines a powerful Deep Convolutional Neural Network (DCNN) architecture with unsupervised tensor decompositions. We demonstrate the power of our methodology in expression and pose transfer, as well as discovering powerful features for pose and expression classification. For future work, we believe that designing networks with skip connections for better reconstruction quality and which at the same time can learn a representation space where some of the variations are disentangled would be a promising research direction.

Chapter 6

Conclusion

6.1 Summary of Thesis Achievements

In this thesis, we investigated multilinear methods for disentangling variations from data. We showed that by extending multilinear methods as well as combine it with deep learning methods, we are able to achieve large improvements in a range of applications.

In Chapter 3, we have proposed a novel local-global multilinear framework that is able to synthesising realistic 3D facial expressions given a target neutral face mesh. This framework combines a global tensor model to synthesise the coarse deformation with local patch-wise expression and identity models to generate the detailed expression shape deformation. By first optimising sparsely for expression and then fitting densely for identity, this new framework is able to produce plausible expression but still maintain the target identity. Leveraging this new local-global multilinear framework, our method produced novel expressions that retain the target identity, contain plausible expression details at various scales, and are free from geometric artefacts. We performed experiments on two separate 3D face datasets. One is publicly available but is of lower resolution than the other which is high-resolution. Our method both quantitatively and qualitatively outperformed current state-of-the-art expression synthesis algorithms such as global multilinear model and deformation transfer. Global multilinear model can predict an expression for a specific target identity but has difficulty generalising to out-of-

sample identities. Deformation transfer is the technique currently used in the movie industry to synthesise facial expressions. Though it performs well, it sometimes also transfers identity detail from the reference identity over to the target identity. We showed that our proposed framework leverages the strength of both type of methods without any of the disadvantages. The synthesised expressions are plausible, free from artefacts and contain fine-scaled details. They retain the target identity and generalise well to out-of-sample faces. The synthesised 3D faces can be used to generate high-quality facial rigs for digital avatars in the movie and entertainment industry. Similarly it can also be used to augment existing multi-identity facial datasets with expressions.

In the previous chapter, we focused on how to improve the performance of supervised multilinear models. Next in Chapter 4, we proposed an unsupervised method able to discover the multilinear structure in visual data. In practice, visual data exhibit several modes of variations. For instance, the appearance of faces varies in identity, expression, pose etc. To extract these modes of variations from visual data, several supervised methods that rely on multilinear decomposition have been developed. The main drawbacks of such methods is that they require both labels regarding the modes of variations and the same number of samples under all modes of variations. Therefore, their applicability is limited to well-organised data, usually captured in well-controlled conditions. We focus on the problem of disentangling the modes of variation in unlabelled and possibly incomplete data. We consider sets of data that are incomplete in the sense that access to samples exhibiting every possible type of variation is not guaranteed. To this end, we proposed the first unsupervised multilinear decomposition which uncovers the potential multilinear structure of incomplete sets of data and the corresponding low-dimensional latent variables (coefficients) explaining different types of variation. We developed an alternating least squares algorithm to discover the multilinear structure of data. Our experiments showed that the method is able to discover the multilinear structure of "in-the-wild" visual data without the presence of labels or well-organised input data. One application of the method is in Shape from Shading where the method is able to disentangle illumination and shape from images of faces. By separating illumination from shape, we were able to reconstruct the 3D shape of the face given its 2D image. Our method quantitatively outperformed another

uncalibrated photometric stereo method on a benchmark dataset containing illumination information. Furthermore we were able to reconstruct the 3D shape of ears given its 2D image due to the general nature of the method. A second application is expression transfer. Given synthetic 3D faces with expression variations, our method was able to disentangle identity and expression and transfer the expression of one face to another. Expression transfer can also be applied to 2D facial images where illumination, expression and identity need to be decomposed to transfer the expression of one image to another. Then we extended our proposed method to incorporate robustness constraints and show how this allows the reconstruction of 3D faces from noisy images. Our method can also be modified to incorporate low-rank constraints which allows us to recover the 3D shape of an object from videos containing occlusions. Additional experiments using an unsupervised deep learning pipeline showed the application of the method directly on internet images of human faces as well as cat faces. Our unsupervised multilinear decomposition method is the first to be able to discover the multilinear structure in visual data. It is a general method applicable for different types of visual data (2D as well as 3D) and very versatile. We were able to extend it with robustness and rank constraints, as well as incorporate it as pseudo-ground truth in a deep learning pipeline.

Finally in Chapter 5, we proposed the first, to the best of our knowledge, attempt to jointly disentangle modes of variation that correspond to expression, identity, illumination and pose using no explicit labels regarding these attributes. More specifically, we proposed the first, as far as we know, approach that combines a powerful Deep Convolutional Neural Network (DCNN) architecture with unsupervised tensor decompositions. Our proposed method is a deep latent variable model, where the multiplicative interactions of multiple latent factors of variation are explicitly modelled by means of multilinear (tensor) structure. Our model is aided by pseudo-supervision obtained from a 3D morphable model. We demonstrated that the proposed approach indeed learns disentangled representations of facial expressions and pose, which can be used in various applications, including face editing, as well as 3D face reconstruction and classification of facial expression, identity and pose. For face editing, we can transfer the latent representation of expression or pose from one face to another. We also show how our approach can be used to relight the image. Due to the pseudo-supervision from a 3D morphable model,

our learnt representation can be used to reconstruct a 3D mesh of the face. Furthermore we show that the learnt latent representations are powerful features for identity, expression and pose classification.

6.2 Future Work

Overall we have shown that multilinear methods are powerful techniques which can be customised in various ways to discover the hidden multilinear structure of data. The work presented in this thesis can also be extended in a number of ways.

The framework in Chapter 3 is a powerful method to synthesise 3D facial expressions. A limitation of the framework is the need for a complete 3D data tensor to create the multilinear model. In the case of missing data, the given multilinear model could be replaced by a multilinear decomposition that does not require a full data tensor such as in Chapter 4.

With a larger high-quality 3D facial expression dataset, the method could be combined with recent advances in geometric deep learning. Local and global meshes could be learned using geometric deep learning and combined in a similar way as presented in the chapter to build a high-performance face model. Local approaches can also address the current issues in geometric deep learning for learning high-resolution facial meshes.

The unsupervised multilinear model presented in Chapter 4 presents an investigation into unsupervised multilinear decompositions. Some limitations of the approach are that the dataset has to be class-specific and requires Lambertian-like objects in cases of 2D datasets with illumination variation. One future direction to alleviate this would be to combine this with recent approaches in representation learning using VAE methods.

Another limitation of the method are the lower quality results reconstruction and expression transfer results. The method could also be adapted with local methods to achieve a higher quality decomposition. An alternative direction to consider would be to find a similar custom multilinear decomposition in case of supervised/semi-supervised data for better disentangle-

ment. This could also lead to better disentanglement and higher result quality. An interesting research direction would be to adapt the model for different variations such as temporal data.

Finally in Chapter 5, we have combined unsupervised multilinear decomposition with deep autoencoder. Recent work [AWB18] also explored the possibility of combining a multilinear decoder within deep learning architectures. Though pose variations cannot be captured in the multilinear decomposition presented in Chapter 4, it can be incorporated in the deep autoencoder. Though successful at different tasks, the limitations of the method is its custom build. The entire network is custom-built for 2D human faces and assumes specific variations within the given dataset. A research direction would be the adaption of this idea to multiple generic objects. The approaches in deep representation learning methods would give a clue in how to proceed for this. Another idea would be to investigate how variations such as human age can be included in the same custom structure. An issue of this work is image quality. Due to the lower-dimensional disentangled representation, the edited images are of lower quality. Frequently the underlying multilinear structure cannot reconstruct the high-frequency details of the data. Hence we believe that designing networks with skip connections for better reconstruction quality and which at the same time can learn a representation space where some of the variations are disentangled would be a promising research direction.

List of Figures

1.1 Visualisation of the Multi-PIE [Gro+10] dataset. Collecting data where every person is present in all the lighting and expression variations is an expensive process that does not scale well. 4

2.1 Visualisation of the Multi-PIE [Gro+10] dataset. Collecting data where every person is present in all the lighting and expression variations is an expensive process that does not scale well. 17

2.2 Surface normal: orientation of a vector perpendicular to the tangent plane on the object surface 20

2.3 The incidence angle θ_i: the angle between surface normal $\boldsymbol{n}$ and light source $\boldsymbol{l}$. 21

3.1 Unlike traditional global models, our proposed global-local model is able to extrapolate outside of the training set. 31

3.2 Our approach uses a global multilinear model (blue) to estimate coarse deformation of the new expression. A sparse sampling of the global model result is used to then optimize the local models (green). The local expression models are optimized first to return a plausible new expression. The resulting shape is densely sampled to provide constraints when optimizing the local identity models which predict the identity of the input shape. The result is a high-quality, plausible new facial expression of the input shape. Actions such as model fitting are represented in beveled boxes. 35

3.3 We construct a local multilinear model for each patch in the given patch layout. Each patch has its own expression basis and identity basis. A patch containing the nasolabial fold is visualized with its mean shape and top 5 PCA expression basis. We also visualize a nose patch in terms of its mean and top 5 PCA identity basis. Basis vectors are shown at $+3\sigma$. 37

3.4 **Local Details -** Given the input neutral shape, the global multilinear model contains artifacts when predicting a smile expression. Note the local details of the nose are incorrect, and the nasolabial fold is very different from the ground truth expression. Our global-local framework achieves much more plausible results. 41

3.5 We show the results of our method on [Che+18c] and compare it against the baselines [VT02a; SP04]. While [VT02a] produces unnatural expressions and [SP04] exhibits artefacts not specific to the target identity, our proposed method can robustly generate a plausible expression for the target identity. Our results are qualitatively and quantitatively closer to the ground truth. 42

3.6 We show the results of our method and compare it against the baselines [VT02a; SP04]. We note that the baselines produce unnatural expressions or expressions with artefacts while our results look more natural and plausible for the target identity. 43

3.7 Person A with Displacement Map. The highlighted mesh is the input neutral face mesh. From the texture map and the normal map associated with this mesh, a displacement map can be estimated which would simulate very fine details on the face. Note how the fine hairs of the eyebrows are visible. 45

3.8 Person B with Displacement Map. The highlighted mesh is the input neutral face mesh. From the texture map and the normal map associated with this mesh, a displacement map can be estimated which would simulate very fine details on the face. Note how the fine hairs of the eyebrows are visible. 46

3.9 Ablation Study: Our proposed method returns plausible expressions with the minimum amount of artefacts while maintaining the identity. 47

3.10 Comparison of our proposed method with multilinear model [VT02a] and deformation transfer [SP04] in terms of 3D position difference to the ground truth shape. Our proposed method consistently outperforms the baseline methods. . . 48

3.11 Comparison of our proposed method with multilinear model [VT02a] and deformation transfer [SP04] in terms of angular error between the normals of the predicted mesh and the normals of the ground truth shape. Our proposed method consistently outperforms the baseline methods. 49

3.12 Comparison of our proposed method with multilinear model [VT02a] and deformation transfer [SP04] in terms of 3D position difference to the ground truth on [Che+18c]. Our proposed method consistently outperforms the baseline methods. 50

3.13 Comparison of our proposed method with multilinear model [VT02a] and deformation transfer [SP04] in terms of angular error between the normals of the predicted mesh and the normals of the ground truth on [Che+18c]. Our proposed method consistently outperforms the baseline methods. 50

3.14 Extracting the coarse deformation from individual subjects instead of the global model allows to synthesize a wide range of expression nuances while preserving the target identity, which is very valuable for data augmentation purposes. . . . 52

4.1 Visualisation of the unsupervised multilinear decomposition and its applications. A sample vector $\mathbf{x}_i$ is assumed to be generated by a common multilinear structure $\mathcal{B}$ and sample specific weights e.g. $\mathbf{l}_i$, $\mathbf{e}_i$ and $\mathbf{c}_i$. We assume the weights correspond to variations in the data ($\mathbf{l}_i$ to lighting, $\mathbf{e}_i$ to expression and $\mathbf{c}_i$ to identity). By varying $\mathbf{e}_i$ only, we expect to see changes in expression but no change in identity or lighting. Similarly, if we vary $\mathbf{c}_i$ only we expect the expression and lighting to remain the same but the identities to change. Additionally if we remove the lighting $\mathbf{l}_i$, we expect the remaining information to correspond to the 3D shape of the object. 57

4.2 3D shape reconstruction: Comparison of our proposed method with photometric stereo [Woo80] and the person-specific photometric stereo in general lighting of [BJK07]. Images from the Photoface [Zaf+13] dataset. 76

4.3 Comparison of our proposed method with person-specific photometric stereo in general lighting of [BJK07]. The error has been calculated against the estimated normals from photometric stereo [Woo80]. 77

4.4 Face and ear reconstructions. Sample images from the HELEN [Le+12] and Ear datasets. 78

4.5 The 3 first expression bases from the decomposition of the synthetic 3D data. We note that despite the data not containing the neutral expression, the first expression basis corresponds to the neutral expression. The other basis display different expression variations. 80

4.6 Sample data of the synthetic 3D dataset. Images 1 to 3 from the left show different identities and images 4 to 6 different expressions. 80

4.7 Neutralising expressions: We show the result of generating synthetic neutral faces by using the neutral expression basis. The results look promising. 80

4.8 We show the results of the expression transfer experiment. The transferred expressions look convincing. 81

4.9 3D Reconstruction on Multi-PIE [Gro+10] dataset. The results show that 3-way disentanglement is possible. 82

4.10 Expression transfer on Multi-PIE. As our decomposition reduces the dimensionality of the images in the dataset, we show the images with the transferred expression next to the reconstructed image of the ground truth from the dataset. Given the decomposition, the reconstruction represents the result of a plausible expression transfer. 83

4.11 **Comparison of the robust and non-robust decomposition**. Images from the HELEN [Le+12] dataset. 84

4.12 Ear reconstructions. Sample images from the Ear dataset. 84

4.13 Sample reconstruction from the Photoface dataset with 1% salt&pepper noise using non-robust and robust decomposition. 86

4.14 Face reconstructions from occluded video frames using the rank-constrained decomposition. 88

4.15 3D faces generated by keeping the identity component $\boldsymbol{C}$ fixed and randomly sampling the expression component $\boldsymbol{E}$. 89

4.16 Face reconstructions from single "in-the-wild" images using the deep unsupervised model trained on HELEN. Though the results may not seem impressive, they are obtained from a pipeline of two unsupervised methods. In this aspect, the results are good. 89

4.17 Comparison of our two proposed methods with person-specific photometric stereo in general lighting of [BJK07] and a generic state-of-the-art network [BRG16]. The error has been calculated against the estimated normals from photometric stereo [Woo80]. 90

4.18 The deep model: ResNet-50 based architecture. 90

4.19 Face reconstructions from single "in-the-wild" cats images using the deep unsupervised model trained on human faces and ears. 91

4.20 Sample reconstruction from the Photoface dataset with state-of-the-art general network [BRG16](middle) and our deep network trained in a unsupervised fashion (right). 93

5.1 Given a single in-the-wild image, our network learns disentangled representations for pose, illumination, expression and identity. Using these representations, we are able to manipulate the image and edit the pose or expression. 95

5.2 Our proof-of-concept network is an end-to-end trained auto-encoder. The encoder E extracts latent variables corresponding to expression and identity from the input image $\boldsymbol{x}$. These latent variables are then fed into the decoder D to reconstruct the image. A separate stream also reconstructs facial texture from $\boldsymbol{z}_{id}$. We impose a multilinear structure and enforce the disentanglement of variations. In the extended version a) the encoder also extracts a latent variable corresponding to pose. The decoder takes in this information and reconstructs an image containing pose variations. 102

5.3 Our network is an end-to-end trained auto-encoder. The encoder E extracts latent variables corresponding to illumination, pose, expression and identity from the input image $\boldsymbol{x}$. These latent variables are then fed into the decoder D to reconstruct the image. We impose a multilinear structure and enforce the disentangling of variations. The grey triangles represent the losses: adversarial loss A, verification loss V, L_1 and L_2 losses. 103

5.4 Our network is able to transfer the expression from one face to another by disentangling the expression components of the images. The ground truth has been computed using the ground truth texture with synthetic identity and expression components. 109

5.5 Given a single image, we infer meaningful expression and identity components to reconstruct a 3D mesh of the face. We compare the reconstruction (last row) against the ground truth (2^{nd} row). 110

5.6 Given a single image, we infer the facial texture. We compare the reconstructed facial texture (last row) against the ground truth texture (2^{nd} row). 110

5.7 Our network is able to transfer the pose from one face to another by disentangling the pose, expression and identity components of the images. The ground truth has been computed using the ground truth texture with synthetic pose, identity and expression components. 111

5.8 Our network is able to transfer the expression from one face to another by disentangling the expression components of the images. We compare our expression editing results with a baseline where a 3DMM has been fit to both input images. 114

5.9 We compare our expression editing results with [Wan+17b]. As [Wan+17b] is not able to disentangle pose, editing expressions from images of different poses returns noisy results. 115

5.10 Our network is able to transfer the pose of one face to another by disentangling the pose components of the images. We compare our pose editing results with a baseline where a 3DMM has been fit to both input images. 115

5.11 Our network is also able to transfer the identity of one image to another by disentangling the identity components of the images. 116

5.12 Ablation study on different losses (multilinear, adversarial, verification) for expression editing. The results show that incorporating multilinear losses indeed helps the network to better disentangle the expression variations. 117

5.13 Ablation study on different losses (multilinear, adversarial, verification) for facial pose editing. The results show that incorporating multilinear losses helps the network to better disentangle the pose variations. 118

5.14 Histogram of cosine similarities on 600 pairs of "non-same" people from CelebA 119

5.15 Texture reconstruction compared with [Tew+18; TL18]. [Tew+18; TL18] have been trained with images of higher resolutions of 240×240 and 128×128 respectively. In comparison our model has only been trained with images of size 64×64 pixels. 119

5.16 We show how to interpolate the expression latent code from the rightmost person to the expression on the leftmost person. As the interpolation is only done on the expression code, the identity remains the identity of the rightmost person. . 120

5.17 We show how to interpolate the identity latent code from the rightmost person to the leftmost person. As the interpolation is only done on the identity code, the expression remains the one of the rightmost person. 120

5.18 Using the illumination and normals estimated by our network, we are able to relight target faces using illumination from the source image. The source $\hat{\boldsymbol{s}}^{source}$ and target shading $\hat{\boldsymbol{s}}^{target}$ are displayed to visualise against the new transferred shading $\boldsymbol{s}^{transfer}$. We compare against [Shu+17]. 122

5.19 Given a single image, we infer meaningful expression and identity components to reconstruct a 3D mesh of the face. We compare our 3D estimation against recent works [Jac+17; Fen+18]. 123

5.20 Comparison of the estimated normals obtained using the proposed model vs the ones obtained by [Wan+17b] and [Shu+17] on the Photoface dataset. The proposed method outperforms the baselines on the majority of cases. 124

5.21 Visualisation of our $\boldsymbol{Z}_{exp}$ and baseline $\boldsymbol{Z}_0$ using t-SNE. Our latent $\boldsymbol{Z}_{exp}$ clusters better with regards to expression than the latent space $\boldsymbol{Z}_0$ of an auto-encoder. . 126

5.22 Visualisation of our $\boldsymbol{Z}_p$ and baseline $\boldsymbol{Z}_0$ using t-SNE. It is evident that the proposed disentangled $\boldsymbol{Z}_p$ clusters better with regards to pose than the latent space $\boldsymbol{Z}_0$ of an auto-encoder. 127

List of Tables

3.1 Quantitative comparison reporting the mean position and mean normal angular error. Our method outperforms the baselines in terms of both position error and normal error on both datasets. 49

3.2 Users tend to select shapes synthesised by the proposed method as their preferred result twice as likely than shapes synthesised by the baselines. 51

4.1 Comparison of estimated normals 76

4.2 Prediction accuracy on synthetic dataset 85

4.3 Angular error for our method with and without robustness on Photoface containing 1% salt&pepper noise. Our robust method outperforms our basic method in terms of 3D reconstruction from noisy data. 86

4.4 Angular error for our method with and without low-rank constraints on videos containing baboon patch occlusions. 87

4.5 Expression classification results using unsupervised and semi-supervised decomposition. 89

4.6 Angular error for the various surface normal estimation methods on the Photoface [Zaf+13] dataset 91

5.1 Angular error for the various surface normal estimation methods on the Photoface [Zaf+13] dataset. We also show the proportion of the normals below 35° and 40°. 123

5.2 Classification accuracy results: we try to classify 54 identities using $\boldsymbol{z}_{id}$, 6 expressions using $\boldsymbol{z}_{exp}$ and 7 poses using $\boldsymbol{z}_{p}$. We compare against standard baseline methods such as SIFT and CNN. 123

5.3 Identity classification accuracy results: we classify 54 identities using $\boldsymbol{z}_{id}$ with and without verification loss. 125

5.4 Classification accuracy results in comparison with [Wan+17b]: As [Wan+17b] works on frontal images, we only consider frontal images in this experiment. We try to classify 54 identities using $\boldsymbol{z}_{id}$ vs. $\boldsymbol{C}$, 6 expressions using $\boldsymbol{z}_{exp}$ vs. $\boldsymbol{E}$ and 16 illumination using $\boldsymbol{z}_{ill}$ vs. $\boldsymbol{L}$. 126

Bibliography

[AP96] Edward H Adelson and Alex P Pentland. "The perception of shading and reflectance". In: *Perception as Bayesian inference* (1996), pp. 409–423.

[AWB18] Victoria Fernández Abrevaya, Stefanie Wuhrer, and Edmond Boyer. "Multilinear autoencoder for 3d face model learning". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 1–9.

[B+99] Volker Blanz, Thomas Vetter, et al. "A morphable model for the synthesis of 3D faces." In: *Siggraph*. Vol. 99. 1999. 1999, pp. 187–194.

[BBW14] Alan Brunton, Timo Bolkart, and Stefanie Wuhrer. "Multilinear Wavelets: A Statistical Shape Space for Human Faces". In: *ECCV*. 2014.

[BCV13a] Y. Bengio, A. Courville, and P. Vincent. "Representation Learning: A Review and New Perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (Aug. 2013), pp. 1798–1828. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2013.50.

[BCV13b] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[Ber82] DP Bertsekas. *Constrained optimization and Lagrange multiplier methods*. 1982. DOI: 10.1002/net.3230150112.

[BHB00] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. "Recovering non-rigid 3D shape from image streams". In: *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*. Vol. 2. IEEE. 2000, pp. 690–696.

[BJ01] Ronen Basri and David Jacobs. "Lambertian reflectances and linear subspaces". In: *IEEE International Conference on Computer Vision* 00.C (2001), pp. 383–390.

[BJ03] Ronen Basri and David W Jacobs. "Lambertian reflectance and linear subspaces". In: *IEEE T-PAMI* 25.2 (2003), pp. 218–233.

[BJK07] Ronen Basri, David Jacobs, and Ira Kemelmacher. "Photometric stereo with general, unknown lighting". In: *International Journal of Computer Vision* 72.3 (2007), pp. 239–257. ISSN: 09205691. DOI: 10.1007/s11263-006-8815-7.

[Bla+03] Volker Blanz et al. "Reanimating faces in images and video". In: *Computer graphics forum*. Vol. 22. 3. Wiley Online Library. 2003, pp. 641–650.

[Boo+16a] James Booth et al. "A 3D Morphable Model learnt from 10'000 faces". In: *Computer Vision and Pattern Recognition, IEEE Conference on (CVPR)* (2016), pp. 5543–5552. ISSN: 10636919. DOI: 10.1109/CVPR.2016.598.

[Boo+16b] J. Booth et al. "A 3D Morphable Model learnt from 10,000 faces". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[Boo+17] James Booth et al. "3D Face Morphable Models" In-the-Wild"". In: *arXiv preprint arXiv:1701.05360* (2017).

[BRG16] Aayush Bansal, Bryan C. Russell, and Abhinav Gupta. "Marr Revisited: 2D-3D Alignment via Surface Normal Prediction". In: *CVPR* (2016).

[BSM17] David Berthelot, Tom Schumm, and Luke Metz. "Began: Boundary equilibrium generative adversarial networks". In: *arXiv preprint arXiv:1703.10717* (2017).

[Bur+18] Christopher P. Burgess et al. *Understanding disentangling in $fffdfffd$-VAE*. 2018. arXiv: 1804.03599 [stat.ML].

[BW16] Timo Bolkart and Stefanie Wuhrer. "A Robust Multilinear Model Learning Framework for 3D Faces". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4911–4919.

[Can+11] Emmanuel J Candes et al. "Robust principal component analysis?" In: *Journal of the ACM* 58.3 (2011), pp. 1–37.

[Cao+14a] Chen Cao et al. "FaceWarehouse: A 3D facial expression database for visual computing". In: *IEEE Transactions on Visualization and Computer Graphics* 20.3 (2014), pp. 413–425.

[Cao+14b] Chen Cao et al. "Facewarehouse: A 3d facial expression database for visual computing". In: *IEEE Transactions on Visualization and Computer Graphics* 20.3 (2014), pp. 413–425.

[CET01] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. "Active appearance models". In: *IEEE Transactions on pattern analysis and machine intelligence* 23.6 (2001), pp. 681–685.

[Che+14] Brian Cheung et al. "Discovering hidden factors of variation in deep networks". In: *arXiv preprint arXiv:1412.6583* (2014).

[Che+16] Xi Chen et al. "Infogan: Interpretable representation learning by information maximizing generative adversarial nets". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2172–2180.

[Che+18a] Tian Qi Chen et al. "Isolating sources of disentanglement in variational autoencoders". In: *Advances in Neural Information Processing Systems*. 2018, pp. 2610–2620.

[Che+18b] Shiyang Cheng et al. "4DFAB: A Large Scale 4D Database for Facial Expression Analysis and Biometric Applications". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5117–5126.

[Che+18c] S. Cheng et al. "4DFAB: A Large Scale 4D Database for Facial Expression Analysis and Biometric Applications". In: *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018)*. Salt Lake City, Utah, US, June 2018.

[Cho+18] Yunjey Choi et al. "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8789–8797.

[DCB12] Guillaume Desjardins, Aaron Courville, and Yoshua Bengio. "Disentangling factors of variation via generative entangling". In: *arXiv preprint arXiv:1210.5474* (2012).

[DDV00] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. "A multilinear singular value decomposition". In: *SIAM journal on Matrix Analysis and Applications* 21.4 (2000), pp. 1253–1278.

[DGZ18] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. "ArcFace: Additive Angular Margin Loss for Deep Face Recognition". In: *CoRR* abs/1801.07698 (2018). arXiv: 1801.07698. URL: http://arxiv.org/abs/1801.07698.

[DN08] Zhigang Deng and Junyong Noh. "Computer facial animation: A survey". In: *Data-driven 3D facial animation*. Springer, 2008, pp. 1–28.

[FC88a] Robert T Frankot and Rama Chellappa. "A method for enforcing integrability in shape from shading algorithms". In: *IEEE T-PAMI* 10.4 (1988), pp. 439–451.

[FC88b] Robert T Frankot and Rama Chellappa. "Method for Enforcing Integrability in Shape from Shading Algorithms". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.4 (1988), pp. 439–451. ISSN: 01628828. DOI: 10.1109/34.3909.

[Fen+18] Yao Feng et al. "Joint 3D Face Reconstruction and Dense Alignment with Position Map Regression Network". In: *The European Conference on Computer Vision (ECCV)*. Sept. 2018.

[FW11] Leandre R Fabrigar and Duane T Wegener. *Exploratory factor analysis*. Oxford University Press, 2011.

[GBK01] A S Georghiades, P N Belhumeur, and D J Kriegman. "From few to many: illumination cone models for face recognition under variable lighting and pose". In: *IEEE T-PAMI* 23.6 (2001), pp. 643–660.

[GD04] John C. Gower and Garmt B. Dijksterhuis. *Procrustes problems*. Vol. 30. Oxford Statistical Science Series. Oxford, UK: Oxford University Press, Jan. 2004.

[Goo+14] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[Gro+10] Ralph Gross et al. "Multi-PIE". In: *Image and Vision Computing* 28.5 (2010), pp. 807–813.

[GRY11] Silvia Gandy, Benjamin Recht, and Isao Yamada. "Tensor completion and low-n-rank tensor recovery via convex optimization". In: *Inverse Problems* 27.2 (2011), p. 025010.

[HAG15] Bharath Hariharan, Pablo Arbel, and Ross Girshick. "Hypercolumns for Object Segmentation and Fine-grained Localization". In: *CVPR* (2015), pp. 447–456.

[He+05] Xiaofei He et al. "Face recognition using Laplacianfaces". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.3 (Mar. 2005), pp. 328–340. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2005.55.

[He+16] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CVPR* 7.3 (2016), pp. 171–180.

[Hig+] Irina Higgins et al. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework." In: ().

[HKW11] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. "Transforming auto-encoders". In: *International Conference on Artificial Neural Networks*. Springer. 2011, pp. 44–51.

[Hot33] Harold Hotelling. "Analysis of a complex of statistical variables into principal components." In: *Journal of educational psychology* 24.6 (1933), p. 417.

[HS06] Geoffrey E Hinton and Ruslan R Salakhutdinov. "Reducing the dimensionality of data with neural networks". In: *science* 313.5786 (2006), pp. 504–507.

[IS15] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. 2015, pp. 448–456.

[Jac+17] Aaron S Jackson et al. "Large Pose 3D Face Reconstruction from a Single Image via Direct Volumetric CNN Regression". In: *International Conference on Computer Vision* (2017).

[KB08] Tamara G. Kolda and Brett W. Bader. "Tensor Decompositions and Applications". In: *SIAM Review* 51.3 (2008), pp. 455–500. ISSN: 0036-1445. DOI: 10.1137/07070111X.

[KB09] Tamara G Kolda and Brett W Bader. "Tensor decompositions and applications". In: *SIAM review* 51.3 (2009), pp. 455–500.

[KD80] Pieter M Kroonenberg and Jan De Leeuw. "Principal component analysis of three-mode data by means of alternating least squares algorithms". In: *Psychometrika* 45.1 (1980), pp. 69–97.

[Kem13a] Ira Kemelmacher-Shlizerman. "Internet based morphable model". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 3256–3263.

[Kem13b] Ira Kemelmacher-Shlizerman. "Internet-based Morphable Model". In: *IEEE International Conference on Computer Vision (ICCV)* (2013). DOI: 10.1109/ICCV.2013.404.

[KLA19] Tero Karras, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4401–4410.

[KM18] Hyunjik Kim and Andriy Mnih. "Disentangling by Factorising". In: *International Conference on Machine Learning*. 2018, pp. 2649–2658.

[Kok16] Iasonas Kokkinos. "UberNet: Training a 'Universal' Convolutional Neural Network for Low-, Mid-, and High-Level Vision using Diverse Datasets and Limited Memory". In: *CoRR* abs/1609.02132 (2016).

[KR68] CG Khatri and C Radhakrishna Rao. "Solutions to some functional equations and their applications to characterization of probability distributions". In: *Sankhyā: The Indian Journal of Statistics, Series A* (1968), pp. 167–180.

[Kru89] Joseph B Kruskal. "Rank, decomposition, and uniqueness for 3-way and N-way arrays". In: *Multiway data analysis*. North-Holland Publishing Co. 1989, pp. 7–18.

[KS12] I Kemelmacher-Shlizerman and S M Seitz. "Collection flow". In: *CVPR*. IEEE, 2012, pp. 1792–1799.

[KSB17] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. "Variational inference of disentangled latent concepts from unlabeled observations". In: *arXiv preprint arXiv:1711.00848* (2017).

[Kul+15] Tejas D Kulkarni et al. "Deep convolutional inverse graphics network". In: *Advances in Neural Information Processing Systems*. 2015, pp. 2539–2547.

[KW13] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[Le+12] Vuong Le et al. "Interactive facial feature localization". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 7574 LNCS. PART 3. 2012, pp. 679–692.

[Lew+14] John P Lewis et al. "Practice and Theory of Blendshape Facial Models." In: *Eurographics (State of the Art Reports)* 1.8 (2014), p. 2.

[Li+17] Tianye Li et al. "Learning a model of facial shape and expression from 4D scans". In: *ACM Transactions on Graphics (TOG)* 36.6 (2017), p. 194.

[Liu+13] Ji Liu et al. "Tensor completion for estimating missing values in visual data". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.1 (2013), pp. 208–220.

[Liu+15] Ziwei Liu et al. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015.

[Mak+15] Alireza Makhzani et al. "Adversarial autoencoders". In: *arXiv preprint arXiv:1511.05644* (2015).

[Mat+16] Michael F Mathieu et al. “Disentangling factors of variation in deep representation using adversarial training”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 5040–5048.

[MB04] Iain Matthews and Simon Baker. “Active appearance models revisited”. In: *International journal of computer vision* 60.2 (2004), pp. 135–164.

[MH08] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of Machine Learning Research* 9.Nov (2008), pp. 2579–2605.

[MK01] A. M. Martinez and A. C. Kak. “PCA versus LDA”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.2 (Feb. 2001), pp. 228–233. ISSN: 0162-8828. DOI: 10.1109/34.908974.

[Neu69] Heinz Neudecker. “Some theorems on matrix differentiation with special reference to Kronecker matrix products”. In: *Journal of the American Statistical Association* 64.327 (1969), pp. 953–963.

[Pum+18] Albert Pumarola et al. “Ganimation: Anatomically-aware facial animation from a single image”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 818–833.

[QC15] Qiang Qiu and Rama Chellappa. “Compositional dictionaries for domain adaptive face recognition”. In: *IEEE Transactions on Image Processing* 24.12 (2015), pp. 5152–5165.

[Ram02] Ravi Ramamoorthi. “Analytic PCA construction for theoretical analysis of lighting variability in images of a Lambertian object”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.10 (2002), pp. 1–12.

[Ree+14] Scott Reed et al. “Learning to Disentangle Factors of Variation with Manifold Interaction”. In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1431–1439.

[RH01] Ravi Ramamoorthi and Pat Hanrahan. "On the relationship between radiance and irradiance: determining the illumination from images of a convex Lambertian object". In: *JOSA* 18.10 (2001), pp. 2448–2459.

[RH10] Florian Roemer and Martin Haardt. "Tensor-based channel estimation and iterative refinements for two-way relaying with multiple antennas and spatial reuse". In: *IEEE Transactions on Signal Processing* 58.11 (2010), pp. 5720–5735. ISSN: 1053587X. DOI: 10.1109/TSP.2010.2062179.

[Roe12] Florian Roemer. "Advanced algebraic concepts for efficient multi-channel signal processing". PhD thesis. Universitätsbibliothek Ilmenau, 2012.

[Sag+16] Christos Sagonas et al. "300 faces in-the-wild challenge: Database and results". In: *Image and Vision Computing* 47 (2016), pp. 3–18.

[Sag+17] Christos Sagonas et al. "Robust joint and individual variance explained". In: *Proceedings of IEEE InternationalConference on Computer Vision & Pattern Recognition (CVPR)*. 2017.

[SBB03] Terence Sim, Simon Baker, and Maan Bsat. "The CMU Pose, Illumination, and Expression Database". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.12 (Dec. 2003), pp. 1615–1618.

[SDS10] Marco Signoretto, Lieven De Lathauwer, and Johan AK Suykens. "Nuclear norms for tensors and their use for convex multilinear estimation". In: *Linear Algebra and Its Applications* 43 (2010).

[Shu+17] Z. Shu et al. "Neural Face Editing with Intrinsic Image Disentangling". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*. 2017.

[SK87] Lawrence Sirovich and Michael Kirby. "Low-dimensional procedure for the characterization of human faces". In: *Josa a* 4.3 (1987), pp. 519–524.

[SP04] Robert W. Sumner and Jovan Popović. "Deformation Transfer for Triangle Meshes". In: *ACM Trans. Graph.* 23.3 (Aug. 2004), pp. 399–405. ISSN: 0730-0301. DOI:

10.1145/1015706.1015736. URL: http://doi.acm.org/10.1145/1015706.1015736.

[SPZ15] Patrick Snape, Yannis Panagakis, and Stefanos Zafeiriou. "Automatic construction of robust spherical harmonic subspaces". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 91–100.

[SZ09] Josef Sivic and Andrew Zisserman. "Efficient visual search of videos cast as text retrieval". In: *IEEE transactions on pattern analysis and machine intelligence* 31.4 (2009), pp. 591–606.

[Tew+17] Ayush Tewari et al. "MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction". In: *The IEEE International Conference on Computer Vision (ICCV)*. 2017.

[Tew+18] Ayush Tewari et al. "Self-supervised Multi-level Face Model Learning for Monocular Reconstruction at over 250 Hz". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[Thi+15] Justus Thies et al. "Real-Time Expression Transfer for Facial Reenactment". In: *ACM Trans. Graph.* 34.6 (Oct. 2015). ISSN: 0730-0301. DOI: 10.1145/2816795.2818056.

[TL18] Luan Tran and Xiaoming Liu. "Nonlinear 3D Face Morphable Model". In: *In Proceeding of IEEE Computer Vision and Pattern Recognition*. Salt Lake City, UT, June 2018.

[TP91] Matthew A Turk and Alex P Pentland. "Face recognition using eigenfaces". In: *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*. IEEE. 1991, pp. 586–591.

[TSH13] Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey Hinton. "Tensor analyzers". In: *International Conference on Machine Learning*. 2013, pp. 163–171.

[TSL00] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. "A Global Geometric Framework for Nonlinear Dimensionality Reduction". In: *Science* 290.5500 (2000), pp. 2319–2323. ISSN: 0036-8075. DOI: 10.1126/science.290.5500.2319.

[Tuc66] Ledyard R Tucker. "Some mathematical notes on three-mode factor analysis". In: *Psychometrika* 31.3 (1966), pp. 279–311.

[TYL17] Luan Tran, Xi Yin, and Xiaoming Liu. "Disentangled representation learning gan for pose-invariant face recognition". In: *CVPR*. Vol. 4. 5. 2017, p. 7.

[Vla+05] Daniel Vlasic et al. "Face Transfer with Multilinear Models". In: *ACM Trans. Graph.* 24.3 (July 2005), pp. 426–433. ISSN: 0730-0301. DOI: 10.1145/1073204.1073209. URL: http://doi.acm.org/10.1145/1073204.1073209.

[VT02a] M. Alex O. Vasilescu and Demetri Terzopoulos. "Multilinear Analysis of Image Ensembles: TensorFaces". In: *Computer Vision — ECCV 2002*. Ed. by Anders Heyden et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 447–460. ISBN: 978-3-540-47969-7.

[VT02b] M Alex O Vasilescu and Demetri Terzopoulos. "Multilinear analysis of image ensembles: Tensorfaces". In: *European Conference on Computer Vision*. Springer. 2002, pp. 447–460.

[Wan+17a] Chaoyue Wang et al. "Tag Disentangled Generative Adversarial Network for Object Image Re-rendering". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. 2017, pp. 2901–2907. DOI: 10.24963/ijcai.2017/404.

[Wan+17b] Mengjiao Wang et al. "Learning the multilinear structure of visual data". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 4592–4600.

[Wan+18] Mengjiao Wang et al. "Disentangling the Modes of Variation in Unlabelled Data". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.11 (Nov. 2018), pp. 2682–2695. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2017.2783940.

[Wan+19] Mengjiao Wang et al. "An Adversarial Neuro-Tensorial Approach for Learning Disentangled Representations". In: *International Journal of Computer Vision* (Feb. 2019). ISSN: 1573-1405. DOI: 10.1007/s11263-019-01163-7. URL: https://doi.org/10.1007/s11263-019-01163-7.

[WHS15] Xiang Wu, Ran He, and Zhenan Sun. "A Lightened CNN for Deep Face Representation". In: *arXiv preprint arXiv:1511.02683* (2015).

[Woo80] R J Woodham. *Photometric method for determining surface orientation from multiple images*. 1980. DOI: 10.1117/12.7972479.

[Wor+17] Daniel E. Worrall et al. "Interpretable Transformations With Encoder-Decoder Networks". In: *The IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.

[Wu+16] Chenglei Wu et al. "An Anatomically-constrained Local Deformation Model for Monocular Face Capture". In: *ACM Trans. Graph.* 35.4 (July 2016), 115:1–115:12. ISSN: 0730-0301. DOI: 10.1145/2897824.2925882. URL: http://doi.acm.org/10.1145/2897824.2925882.

[Yan+07] S. Yan et al. "Graph Embedding and Extensions: A General Framework for Dimensionality Reduction". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.1 (Jan. 2007), pp. 40–51. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2007.250598.

[Yan+11] Fei Yang et al. "Expression flow for 3D-aware face component transfer". In: *ACM Transactions on Graphics (TOG)*. Vol. 30. 4. ACM. 2011, p. 60.

[Yin+06] Lijun Yin et al. "A 3D facial expression database for facial behavior research". In: *7th international conference on automatic face and gesture recognition (FGR06)*. IEEE. 2006, pp. 211–216.

[Yin+08] L. Yin et al. "A high-resolution 3D dynamic facial expression database". In: *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*. Sept. 2008, pp. 1–6. DOI: 10.1109/AFGR.2008.4813324.

[Zaf+13] Stefanos Zafeiriou et al. "Face recognition and verification using photometric stereo: The photoface database and a comprehensive evaluation". In: *IEEE Transactions on Information Forensics and Security* 8.1 (2013), pp. 121–135. ISSN: 15566013. DOI: 10.1109/TIFS.2012.2224109.

[Zaf+17] S. Zafeiriou et al. "Large Scale 3D Morphable Models". In: *International Journal of Computer Vision* (2017).

[ZGL03] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. "Semi-supervised Learning Using Gaussian Fields and Harmonic Functions". In: *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*. ICML'03. Washington, DC, USA: AAAI Press, 2003, pp. 912–919. ISBN: 1-57735-189-4.

[Zha+14] Yihao Zhang et al. "Facial expression cloning with elastic and muscle models". In: *Journal of Visual Communication and Image Representation* 25.5 (2014), pp. 916–927.

[ZHT06] Hui Zou, Trevor Hastie, and Robert Tibshirani. "Sparse principal component analysis". In: *Journal of Computational and Graphical Statistics* 15.2 (2006), pp. 265–286. ISSN: 1061-8600. DOI: 10.1198/106186006X113430. arXiv: 1205.0121v2.

[Zol+18] Michael Zollhöfer et al. "State of the art on monocular 3D face reconstruction, tracking, and applications". In: *Computer Graphics Forum*. Vol. 37. 2. Wiley Online Library. 2018, pp. 523–550.