Large-scale learning of shape and motion models for the 3D face

Victoria Fernandez-Abrevaya

Université Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : Mathématiques Appliquées

Arrêté ministériel : 25 mai 2016

Présentée par

Victoria FERNANDEZ ABREVAYA

Thèse dirigée par **Edmond BOYER**
et codirigée par **Stefanie WUHRER**, CR, INRIA

préparée au sein du **Laboratoire Laboratoire Jean Kuntzmann**
dans **l'École Doctorale Mathématiques, Sciences et technologies de l'information, Informatique**

Apprentissage à grande échelle de modèles de formes et de mouvements pour le visage 3D

Large-scale learning of shape and motion models for the 3D face

Thèse soutenue publiquement le **23 novembre 2020**,
devant le jury composé de :

Monsieur EDMOND BOYER
DIRECTEUR DE RECHERCHE, INRIA CENTRE DE GRENOBLE RHÔNE-ALPES, Directeur de thèse
Monsieur WILLIAM A. P. SMITH
PROFESSEUR ASSISTANT, UNIVERSITE D'YORK - ROYAUME-UNI, Rapporteur
Monsieur KIRAN VARANASI
PROFESSEUR, UNIVERSITE DE LEIPZIG - ALLEAMGNE, Rapporteur
Monsieur THOMAS VETTER
PROFESSEUR, UNIVERSITE DE BALE - SUISSE, Examinateur
Monsieur RADU HORAUD
DIRECTEUR DE RECHERCHE, INRIA CENTRE DE GRENOBLE RHÔNE-ALPES, Président
Madame STEFANIE WUHRER
CHARGE DE RECHERCHE HDR, INRIA CENTRE DE GRENOBLE RHÔNE-ALPES, Co-directrice de thèse

Abstract

Data-driven models of the 3D face are a promising direction for capturing the subtle complexities of the human face, and a central component to numerous applications thanks to their ability to simplify complex tasks. Most data-driven approaches to date were built from either a relatively limited number of samples or by synthetic data augmentation, mainly because of the difficulty in obtaining large-scale and accurate 3D scans of the face. Yet, there is a substantial amount of information that can be gathered when considering publicly available sources that have been captured over the last decade, whose combination can potentially bring forward more powerful models.

This thesis proposes novel methods for building data-driven models of the 3D face geometry, and investigates whether improved performances can be obtained by learning from large and varied datasets of 3D facial scans. In order to make efficient use of a large number of training samples we develop novel deep learning techniques designed to effectively handle three-dimensional face data. We focus on several aspects that influence the geometry of the face: its shape components including fine details, its motion components such as expression, and the interaction between these two subspaces.

We develop in particular two approaches for building generative models that decouple the latent space according to natural sources of variation, *e.g.*identity and expression. The first approach considers a novel deep autoencoder architecture that allows to learn a multilinear model without requiring the training data to be assembled as a complete tensor. We next propose a novel non-linear model based on adversarial training that further improves the decoupling capacity. This is enabled by a new 3D-2D architecture combining a 3D generator with a 2D discriminator, where both domains are bridged by a geometry mapping layer.

As a necessary prerequisite for building data-driven models, we also address the problem of registering a large number of 3D facial scans in motion. We propose an approach that can efficiently and automatically handle a variety of sequences while making minimal assumptions on the input data. This is achieved by the use of a spatiotemporal model as well as a regression-based initialization, and we show that we can obtain accurate registrations in an efficient and scalable manner.

Finally, we address the problem of recovering surface normals from natural images, with the goal of enriching existing coarse 3D reconstructions. We propose a method that can leverage all available image and normal data, whether paired or not, thanks to a new cross-modal learning architecture. Core to our approach is a novel module that we call deactivable skip connections, which allows to transfer the local details from the image to the output surface without hurting the performance when autoencoding modalities, achieving state-of-the-art results for the task.

Keywords. 3D face modeling • Decoupled generative models • 4D face registration • Surface normal estimation

Résumé

Les modèles du visage 3D fondés sur des données sont une direction prometteuse pour capturer les subtilités complexes du visage humain, et une composante centrale de nombreuses applications grâce à leur capacité à simplifier des tâches complexes. La plupart des approches basées sur les données à ce jour ont été construites à partir d'un nombre limité d'échantillons ou par une augmentation par données synthétiques, principalement en raison de la difficulté à obtenir des scans 3D à grande échelle. Pourtant, il existe une quantité substantielle d'informations qui peuvent être recueillies lorsque l'on considère les sources publiquement accessibles qui ont été capturées au cours de la dernière décennie, dont la combinaison peut potentiellement apporter des modèles plus puissants.

Cette thèse propose de nouvelles méthodes pour construire des modèles de la géométrie du visage 3D fondés sur des données, et examine si des performances améliorées peuvent être obtenues en apprenant à partir d'ensembles de données vastes et variés. Afin d'utiliser efficacement un grand nombre d'échantillons d'apprentissage, nous développons de nouvelles techniques d'apprentissage profond conçues pour gérer efficacement les données faciales tri-dimensionnelles. Nous nous concentrons sur plusieurs aspects qui influencent la géométrie du visage : ses composantes de forme, y compris les détails, ses composants de mouvement telles que l'expression, et l'interaction entre ces deux sous-espaces.

Nous développons notamment deux approches pour construire des modèles génératifs qui découplent l'espace latent en fonction des sources naturelles de variation, *e.g.*identité et expression. La première approche considère une nouvelle architecture d'auto-encodeur profond qui permet d'apprendre un modèle multilinéaire sans nécessiter l'assemblage des données comme un tenseur complet. Nous proposons ensuite un nouveau modèle non linéaire basé sur l'apprentissage antagoniste qui davantage améliore la capacité de découplage. Ceci est rendu possible par une nouvelle architecture 3D-2D qui combine un générateur 3D avec un discriminateur 2D, où les deux domaines sont connectés par une couche de projection géométrique.

En tant que besoin préalable à la construction de modèles basés sur les données, nous abordons également le problème de mise en correspondance d'un grand nombre de scans 3D de visages en mouvement. Nous proposons une approche qui peut gérer automatiquement une variété de séquences avec des hypothèses minimales sur les données d'entrée. Ceci est réalisé par l'utilisation d'un modèle spatio-temporel ainsi qu'une initialisation basée sur la régression, et nous montrons que nous pouvons obtenir des correspondances précises d'une manière efficace et évolutive.

Finalement, nous abordons le problème de la récupération des normales de surface à partir d'images naturelles, dans le but d'enrichir les reconstructions 3D grossières existantes. Nous proposons une méthode qui peut exploiter toutes les images disponibles ainsi que les données normales, qu'elles soient couplées ou non, grâce à une nouvelle architecture d'apprentissage cross-modale. Notre approche repose sur un nouveau module qui permet de transférer les détails locaux de l'image vers la surface de

sortie sans nuire aux performances lors de l'auto-encodage des modalités, en obtenant des résultats de pointe pour la tâche.

Contents

List of Figures

1 Introduction

The face is our main vehicle for communication, whether verbally, non-verbally, or even involuntarily via micro-expressions. Our ability to produce at least some of the facial expressions is an innate and not a cultural trait [Ekman and Keltner, 1970], and thus deeply connected to our most basic instincts. It is also our principal source for recognizing people: it is the face that we will recall first when we think of someone. As such, it has long captivated researchers from numerous domains, ranging from biology and psychology to computer vision and computer graphics.

Digital faces have received similar attention and are equivalently challenging. They play a central role in the film and gaming industry, which has pushed for impressive advances with the goal of creating believable characters. Highly realistic faces can currently be modeled, rendered and animated, although the process is still very complex and expensive, requiring significant manual input from the artists. Digital 3D faces are also ubiquitous in consumer-grade applications. From more trivial tasks such as creating effects during a video-conference, to the more promising applications of telepresence, avatar generation and fully autonomous virtual agents, all of these rely on an underlying model of the three-dimensional face. They are also part of numerous applications in the medical field, including early diagnosis of craniofacial disorders [Suttie et al., 2013], reconstruction of missing parts for implant design [Mueller et al., 2011], and pain detection [Zhang et al., 2015], to name a few.

The key enabler for the majority of these applications is the underlying parametric model: a function that generates a 3D face based only on a few parameters, thus reducing the complexity of the task at hand. First proposed almost fifty years ago by Parke [1974], parametric models are widely used in the film industry thanks to their efficiency [Lewis et al., 2014b], as well as in numerous computer vision tasks where the low-dimensional representation can simplify ill-posed problems, *e.g.*recovering a 3D face from a single image [Blanz and Vetter, 1999].

Building accurate models of the 3D face as well as its motion is a hard task, due to two main reasons. First, the anatomy of the face is very complex. Its shape is influenced by the underlying bone structure as well as the surrounding tissue, and movement is induced by small muscles that are attached close to the surface skin. When these muscles contract they create subtle but noticeable

changes that include creases and wrinkles, deforming the surface in a mostly non-rigid manner. To complicate things more, this highly flexible system varies from person to person according to their underlying shape.

A second source of complexity comes from our remarkably tuned ability to read expressions. Already in 1872 Charles Darwin argued that facial expressions are a biological and not a cultural trait [Darwin, 1872], and this was shown to be true for at least certain "universal" expressions [1] in the seminal work of psychologist Paul Ekman [Ekman and Keltner, 1970]. Since this is a skill more instinctive than learned, we can easily recognize subtle errors made by a computer generated face. This phenomenon has even been formalized in the so-called *uncanny valley* hypothesis. Introduced by Mori et al. [1970], the hypothesis relates the level of realism of a character with the emotional response of the observer. The more realistic the better the response, but when it reaches a point of almost-realistic, but not entirely, it produces a highly unpleasant effect due to our ability to recognize the missing aspects. Only when the face is truly realistic one can overcome this "uncanny valley".

Data-driven models are a promising direction for dealing with the aforementioned complexities. The goal here is to learn the particularities of the face from a database of real 3D scans, such that new identities and/or expressions can be generated by manipulating a few parameters based on the statistics of the database. Building such models involves several steps: scanning a large 3D facial dataset with sufficient variation in terms of identities and expressions; establishing *correspondences* among the captured data, such that they all share the same mathematical representation; and correctly modeling the different factors that affect the facial geometry using a meaningful parameterization. Each of these steps has challenges that remain unsolved.

First, capturing and processing the necessary 3D scans for building data-driven models is a laborious task, which is why most current models were learned from relatively limited datasets. Yet, given the interest that the facial shape has received from the research community, there is a large collection of publicly available databases that were acquired throughout the last decades. Combined, they cover a wide range of identities and expressions, whether static (*e.g.* [Cao et al., 2013, Yin et al., 2006, Savran et al., 2008]) or in motion (*e.g.* [Yin et al., 2008, Fanelli et al., 2010, Cosker et al., 2011, Zhang et al., 2014]), and can even capture fine-scale details (*e.g.* [Stratou et al., 2011]). These datasets are a valuable source of information for building data-driven models of the 3D face, and the ability to learn patterns from all of them can potentially result in more powerful models. But handling large 3D datasets coming from various sources presents additional challenges, requiring methods that are at the same time efficient, scalable and robust.

Establishing correspondences among raw scans is also a difficult problem, challenged by noisy data, lack of distinctive structures in large areas of the facial surface, and significant shape variations arising from different ethnicities,

1. Anger, disgust, fear, happiness, sadness and surprise.

age or expressions. The problem is even harder when dealing with large datasets for which its capture setup is not fully controlled, such as the ones previously mentioned.

Finally, there are many factors that influence the facial shape, and accurately capturing these poses too several challenges. One aspect that is often overlooked, and yet necessary for correctly animating a face, is how to model the *interactions* that occur between the shape and motion-related components of the face geometry; that is, how to correctly model the expressions taking into account the underlying shape. This is typically approached by either building independent models for the shape and expression spaces (thus ignoring any possible interaction), or by building tensor-based models which, although taking into account these interactions, are hard to scale to large datasets. Another challenging aspect is how to encode and recover *detailed* surfaces using a data-driven approach, since low-dimensional parametric models usually struggle to recover high-frequency information.

In this thesis we investigate whether better models for the 3D face geometry can be obtained by learning from a large number of real 3D scans. We are interested here in all aspects that influence the geometry of the face: its identity-related components, including its details, as well as those that arise during motion. To this end, we develop novel techniques for building *decoupled* models (*i.e.*those that capture the interaction between the different subspaces) in Chapters 3 and 5; address the problem of establishing dense correspondences among a large number of datasets of 3D faces in motion in Chapter 4; and tackle the problem of recovering finer details through a surface normal representation in Chapter 6. Motivated by the need to handle large-scale datasets coming from publicly available sources, we propose efficient and scalable methods that take advantage of recent deep learning techniques, building performant models that can profit from all available data.

1.1 Contributions

This thesis contributes novel methods for building data-driven models of the 3D face, as well as a novel image/shape prior for the problem of surface normal recovery. We focus here on models that can decouple the latent space such that we can improve applications related to the facial motion. In the final chapter we move the focus onto the problem of recovering surface details, an aspect that is typically not encoded in low-dimensional parametric models. In all cases we developed algorithms that can profit from both large and realistic datasets, making use of deep learning frameworks to efficiently handle a large and varied source of information.

We make in particular the following contributions:

- A novel approach for building *multilinear models* from large datasets, in order to decouple the parameterization in a scalable manner and without the need to assemble the dataset into a complete tensor.
- A novel registration approach for 3D faces in motion, designed to efficiently and robustly put a large number of examples into a common parameterization.
- A novel modeling framework for learning non-linear decoupled models of the 3D face using adversarial learning. To enable this, we contribute a new architecture based on a *geometry mapping layer*, that allows to perform efficient convolutions on the 3D face while leveraging advances in 2D neural networks.
- A novel method for the estimation of surface normals from unconstrained images, which can learn from large and unpaired datasets of high-quality normals and in-the-wild 2D images of the face. To this end we contribute the *deactivable skip connections*, a novel module that enhances the capabilities of cross-modal learning and proved to be key for training from unpaired data.

1.2 Outline

We begin by reviewing the related work in **Chapter 2**, including all the necessary steps for building data-driven models.

Chapter 3 presents the *multilinear autoencoder*, an approach for building tensor-based decoupled models from large datasets that cannot be assembled into a complete tensor. We leverage the capacity of deep neural networks to optimize efficiently over large datasets and propose a new autoencoder architecture that allows to refine an initial multilinear model, such that it can better capture all available data.

Chapter 4 considers the problem of *registration*, and introduces a novel approach designed to handle large datasets of 3D faces in motion. Key to the method is the use of a spatiotemporal model as well as a regression-based initialization approach, which allows to efficiently process a large number of frames and sequences.

With a large and registered dataset now in hand, **Chapter 5** revisits the problem of building decoupled models through a novel framework that makes use of recent generative adversarial learning techniques. This is enabled by a new 3D-2D architecture that allows to generate three-dimensional data while profiting from advances in 2D neural networks, where both domains are bridged by a geometry mapping layer.

In **Chapter 6** we move the focus to the problem of estimating surface normals from unconstrained images, and propose a novel approach based on cross-modal learning which allows to learn a rich latent space that encodes

both 2D facial image and 3D surface normal information, achieving state-of-the-art results for the task.

Finally, **Chapter 7** concludes with a summary of our main contributions as well as considerations for future directions that extend this work.

1.3 Publications

The material presented in this thesis is based on the following publications:

Chapter 3:

- FERNANDEZ ABREVAYA, V., WUHRER, S., BOYER, E. Multilinear Autoencoder for 3D Face Model Learning. *IEEE Winter Conference on Applications of Computer Vision (WACV)*. (2018)

Chapter 4:

- FERNANDEZ ABREVAYA, V., WUHRER, S., BOYER, E. Spatiotemporal Modeling for Efficient Registration of Dynamic 3D Faces. *IEEE International Conference on 3D Vision (3DV)*. (2018)

Chapter 5:

- FERNANDEZ ABREVAYA, V., BOUKHAYMA, A., WUHRER, S., BOYER, E. A Decoupled 3D Facial Shape Model by Adversarial Training. *IEEE International Conference on Computer Vision (ICCV)*. (2019)

Chapter 6:

- FERNANDEZ ABREVAYA, V., BOUKHAYMA, A., TORR, P.H.S., BOYER, E. Cross-modal Deep Face Normals with Deactivable Skip Connections. *IEEE conference on computer vision and pattern recognition (CVPR)*. (2020)

2 Background

There are several ways by which we can discretize a 3D surface for computational purposes. A commonly used approach, and the one we follow throughout most of this thesis, is to represent the surface as a triangular mesh: a collection of vertices $\mathcal{V} = \{v_i \in \mathbb{R}^3, i \in [1 \dots n]\}$ and triangular facets $\mathcal{F} = \{(vi, vj, vk), v_i, v_j, v_k \in \mathcal{V}\}$ that approximate the real, continuous, surface. An example of a facial mesh can be seen in Figure 2.1. This allows to represent the discrete 3D points as encoded by $\mathcal{V}$, as well as any intermediate point p inside a triangle (v_i, v_j, v_k) through its barycentric coordinates (α, β), such that $p = \alpha v_i + \beta v_j + \gamma v_k$, with $\alpha + \beta + \gamma = 1, \alpha, \beta, \gamma \geq 0$. Triangular meshes can also be parameterized into 2D by a one-to-one mapping $\phi : \mathcal{V} \rightarrow D$, in which each vertex $v_i \in \mathcal{V}$ is associated with a coordinate $(u, v)_i$ in the unit square domain D. This so-called UV parameterization allows to store in a regular grid the associated color textures, displacement maps, normal maps, or even the geometry [Gu et al., 2002], encoding dense surface information by interpolating with the barycentric coordinates.

If we want to recover a mesh of n vertices with known triangulation $\mathcal{F}$ from an input (*e.g.*a 2D image) we need to estimate $3n$ parameters –a large number of degrees of freedom if the mesh has a reasonable resolution. Fortunately, this can be significantly reduced when dealing with a specific class of shapes such as the face. Here the amount of variation among different instances is much less than when dealing with arbitrary objects, and one can presume that the shapes live in a manifold of dimension $d << 3n$. Numerous methods have been proposed that exploit this fact using dimensionality reduction techniques over a training dataset [Blanz and Vetter, 1999, Vlasic et al., 2005] or by hand-crafting a set of d basic facial shapes that can then be linearly combined [Lewis et al., 2014b]. Not only this reduces the dimensionality, but also provides a strong prior knowledge on which facial meshes are plausible that can greatly benefit tasks such as 3D reconstruction from images or real-time facial animation.

Several steps are needed before one can study the properties and common patterns of the face geometry from a dataset of 3D scans. The first step is to acquire such a dataset from real people, which should ideally contain detailed scans of the geometry and be diverse in terms of age, sex, ethnicity, or facial expressions (Section 2.1). The next step is to *register* the scans such that they share a common parameterization: each face should contain the same number of vertices and triangulation $\mathcal{F}$, and each three-dimensional point should have

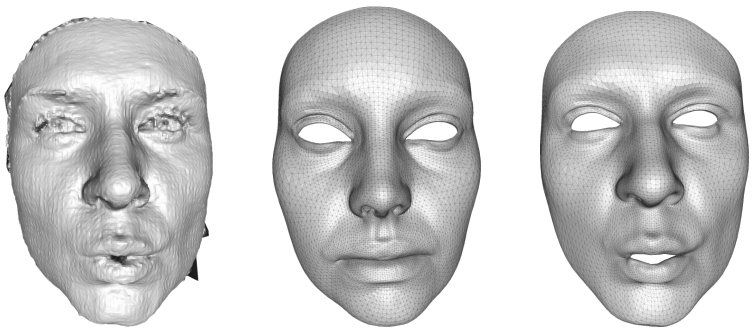


Figure 2.1 – Example of facial mesh. From left to right: raw scan (from Savran et al. [2008]), template mesh (adapted from Alexander et al. [2010]), registered mesh.

the same anatomical meaning (Section 2.2). Once in correspondence the dataset can be rigidly aligned (*e.g.*using Generalized Procrustes Analysis [Davies et al., 2008]), removing rotation, translation, and optionally scale differences [1] such that only variations due to shape remain.

With a registered and rigidly aligned dataset we can now study how each of the vertices vary under different types of deformations. There are several aspects of the face that can be captured by a model. Differences in morphological features give place to variations in **identity**: the traits that distinguish one person from the other. These will be covered in Section 2.3.1. Another important source of variation is that which occurs when performing different **expressions**: changes that occur when a single identity is set in motion (Section 2.3.2). A particular kind of models, that we call here **decoupled** models, allow to capture variations in both identity and expression simultaneously, while taking into account the interactions that occur between these two spaces; these are discussed in Section 2.3.3. Another aspect is the encoding of **details**, as parameteric models usually struggle with higher-frequency components due to the low-dimensional representation; these are reviewed in Section 2.3.4. Finally, the appearance of the face is also crucial for applications such as realistic rendering and analysis-by-synthesis algorithms. We will not review appearance models here as this work is focused on the geometric aspects of the face –interested readers are referred to Klehm et al. [2015], Egger et al. [2019].

This chapter is intended only as a brief overview that will serve as background information for the chapters to come. For a more detailed treatment we refer to Brunton et al. [2014b], Egger et al. [2019] regarding data-driven facial shape models, Lewis et al. [2014b], Orvalho et al. [2012] for 3D facial animation and Zollhöfer et al. [2018] for the application of 3D face reconstruction from monocular input.

1. For anatomical shapes, scale is usually preserved.

2.1 Acquisition

There exists a variety of scanners that can accurately measure the 3D surface of the face. A typical classification divides them into *active* and *passive*, where the former acquires data by emitting a signal which is then measured by a sensor, while the latter employ sensing devices alone such as two or more RGB cameras. Each of these has its own strengths and weaknesses, as well as its own noise characteristics. Active systems include laser scanners [Levoy et al., 2000] which were used for faces in *e.g.* Blanz and Vetter [1999], structured light scanners [Geng, 2011], which were used in *e.g.* Savran et al. [2008] and time-of-flight devices [Hansard et al., 2012], used for example in Cao et al. [2014a]. The accuracy and speed of these sensors can vary greatly, ranging from accurate at the cost of slower frame rates (e.g. the structured light scanner used in [Paysan et al., 2009]), to real-time at the cost of lower resolution (*e.g.*the Kinect sensor [Mutto et al., 2012]).

Passive methods estimate depth by reasoning about the reflected light as captured by one or more sensing devices. A commonly used approach operates by triangulating corresponding pixels from two (*e.g.* [Valgaerts et al., 2012]) or more (*e.g.* [Beeler et al., 2010, Bradley et al., 2010]) RGB cameras, a technique known as multi-view stereo. This allows the acquisition of both texture and depth at high frame rates and are thus particularly well suited for dynamic captures –many publicly available motion data such as [Cosker et al., 2011, Yin et al., 2008] were acquired using this technique. Although the quality of multi-view stereo methods has lagged behind active sensors for many years, recent work *e.g.* [Beeler et al., 2010, 2011, Wu et al., 2011, Fyffe et al., 2017] has demonstrated that high quality 3D scans of the face can be obtained with purely passive approaches, under well-controlled studio setups.

As opposed to the previously mentioned *geometric* methods which estimate surface positions, *photometric* methods are concerned with the estimation of the surface normals, and have been used with both active and passive scanners. This is achieved by analyzing the interaction of the light with the surface, either from one image (a technique called shape-from-shading [Horn and Brooks, 1989]) or multiple images showing different illuminations, a technique called photometric stereo [Woodham, 1980]. Both cases estimate dense per-pixel orientations from which the shape can be recovered by integration [Quéau et al., 2018], or combined with depth to improve an initial reconstruction [Nehab et al., 2005]. These techniques are suitable for acquiring high frequency details and perform well with the facial shape as they do not require a highly textured surface. The most accurate facial acquisitions are typically obtained with a combination of geometric and photometric methods, *e.g.* Ma et al. [2007], Zivanov et al. [2009], Ghosh et al. [2011], Seck et al. [2016].

Although much research effort has been dedicated to reduce the quality gap between high-end systems and consumer-grade devices (*e.g.*RGB or RGB-D cameras), good quality acquisitions still need to be done using expensive systems with complex setups and often costly offline computations. Thus,

it is to date difficult and expensive to acquire a large dataset of 3D scans that will allow the use of modern deep learning techniques. However, the facial shape has always attracted much research interest, and as a consequence numerous databases of 3D facial scans have been acquired and made publicly available. For example, BU-3DFE [Yin et al., 2006], Bosphorus [Savran et al., 2008], ND-2006 [Faltemier et al., 2007] or FaceWarehouse [Cao et al., 2013] contain thousands of scans showing multiple identites in multiple expressions; ICT-3DRFE [Stratou et al., 2011] and Photoface [Zafeiriou et al., 2011] were captured using photometric systems and thus exhibit high frequency details; and dynamic datasets such as BU-4DFE [Yin et al., 2008], D3DFACS [Cosker et al., 2011], BP4D-Spontaneous [Zhang et al., 2014], B3D(AC)^2 [Fanelli et al., 2010] or CoMA [Ranjan et al., 2018] open up the possibility of analyzing the facial surface in motion. In this work we study techniques to make use of this large but varied source of information in order to learn richer models of the 3D shape, as well as its motion-related components such as expression.

2.2 Registration

Capture systems typically return meshes that are inconsistent with each other: each scan contains a different number of vertices, and each of these vertices may have a different anatomical meaning (for example the i-th vertex of one scan might be located at the tip of the nose, while the i-th vertex of another scan might be located in the eye corner). This unordered representation is clearly ill-suited for studying shape variations, as we do not know how each point really varies from one shape to the next. In order to solve this we need to put the data in correspondence, a process that is sometimes referred to as *registration*.

Following the definition of Van Kaick et al. [2011], given a set of input shapes $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n$ the problem of finding correspondences is defined as that of finding a meaningful relation $\mathcal{R}$ between the shapes. Correspondences can be sparse, establishing a connection between a few, typically distinctive points; or dense, where the correspondence is defined for all the primitive elements of the shape (*e.g.*all the vertices of a mesh). Sparse correspondence for 3D faces are commonly referred to as landmarks, and are defined in terms of morphologically relevant and distinguishable features such as the eye corners or the tip of the nose. Example works include Passalis et al. [2011], Creusot et al. [2013], Bolkart and Wuhrer [2015a], Gilani et al. [2017]. We are mostly concerned here with *dense* correspondences for the 3D face, as this will allow to study the properties of entire facial surfaces. Example works include Blanz and Vetter [1999], Amberg et al. [2007], Li et al. [2008] to name a few, and will be elaborated in the following paragraphs.

The nature of the relation $\mathcal{R}$ also gives rise to different types of algorithms. Rigid registration methods study the case where $\mathcal{R}$ is a rigid transformation: the goal here is to find a global rotation $\mathbf{R} \in \mathbb{R}^{3\times3}$ and translation $\mathbf{t} \in \mathbb{R}^3$ that

will minimize some distance function between two shapes $\mathcal{X}$ and $\mathcal{Y}$. A widely used solution is the *Iterative Closest Point* (ICP) algorithm [Besl and McKay, 1992], which iterates between selecting the closest point as a matching correspondence, and computing the optimal rigid transformation for these correspondences by minimizing the *point-to-point* error function:

$$\underset{\mathbf{R}\in\mathbb{R}^{3\times3},\mathbf{t}\in\mathbb{R}^3}{\arg\min} \sum_{\mathbf{p}_i\in\mathcal{X}} \|\mathbf{q}_i - (\mathbf{R}\mathbf{p}_i + \mathbf{t})\|_2^2, \qquad s.t. \mathbf{R}^T\mathbf{R} = \mathbf{I}, det(\mathbf{R}) = 1 \tag{2.1}$$

where $\mathbf{q}_i \in \mathcal{Y}$ is the closest point to $(\mathbf{R}\mathbf{p}_i + \mathbf{t})$ given the current estimate for $\mathbf{R}, \mathbf{t}$. The closest point search can be accelerated by using appropriate data structures such as a kd-tree [Bentley, 1975] or by inverse calibration [Blais and Levine, 1995]. An alternative to Equation 2.1 is to minimize the *point-to-plane* function [Chen and Medioni, 1992, Low, 2004], which considers instead the distance between $(\mathbf{R}\mathbf{p}_i + \mathbf{t})$ and the tangent plane to the closest point:

$$\underset{\mathbf{R}\in\mathbb{R}^{3\times3},\mathbf{t}\in\mathbb{R}^3}{\arg\min} \sum_{\mathbf{p}_i\in\mathcal{X}} \left(\mathbf{n}_{\mathbf{q}_i}^T(\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i)\right)^2, \qquad s.t. \mathbf{R}^T\mathbf{R} = \mathbf{I}, det(\mathbf{R}) = 1 \tag{2.2}$$

with $\mathbf{n}_{\mathbf{q}_i}$ the normal vector at the closest point $\mathbf{q}_i \in \mathcal{Y}$. Both cases will converge to a local solution and as such require a good initialization, although it has been shown that the point-to-plane formulation will typically converge faster. Numerous variants have been proposed that tackle different aspects of the algorithm, see *e.g.* [Castellani and Bartoli, 2012].

Non-rigid registration considers the case where $\mathcal{R}$ can be an arbitrary relation; a significantly more complex problem with a larger solution space. A common approach when working with a specific class of shapes is to do *template fitting*: a pre-computed mesh (*e.g.*an artist-quality facial mesh) is warped towards each of the scans, automatically establishing dense correspondences among all shapes through the common template. Non-rigid variants for this purpose differ mostly in the choice of deformation parameterization. This includes per-vertex affine transformations [Allen et al., 2003, Amberg et al., 2007], vertex displacements [Weise et al., 2009, Salazar et al., 2014], free-form deformations (FFD) [Huang et al., 2003, Wang et al., 2004], thin-plate splines (TPS) [Chui and Rangarajan, 2000, Patel and Smith, 2009, Hutton et al., 2001], deformation graph [Li et al., 2008], physics-based models [Passalis et al., 2005], gaussian mixture models [Myronenko et al., 2007] and gaussian process deformation model [Gerig et al., 2018].

An alternative, used particularly for faces, is to perform the registration in the 2D domain by leveraging the associated UV texture map. Blanz and Vetter [1999] proposed the use of optical flow on the texture maps of 200 subjects in neutral expressions, which worked well on their database of uniform ethnicities. To handle more variations, Patel and Smith [2009] used instead a TPS that interpolates manually annotated landmarks in the UV domain. Manual annotations were replaced by an Active Appearance Model (AAM) in Cosker et al. [2011] and Cheng et al. [2017a]. The work of Booth et al. [2018]

compared the non-rigid ICP approach of Amberg et al. [2007] with the two UV-based approaches (TPS and optical flow), and showed superior results for non-rigid ICP, in the context of automatic registration of a large number of neutral 3D faces. Dai et al. [2017] proposed instead to combine the two alternatives, initially fitting a template using coherent point drift [Myronenko et al., 2007] and afterwards refining using optical flow.

Non-rigid registration can also be aided by the use of a statistical model, such as those outlined in Section 2.3. This was considered in *e.g.* Albrecht et al. [2008], Schneider and Eisert [2009], Cheng et al. [2017b] which used a linear model of shape or expression, Brunton et al. [2014a], Bolkart and Wuhrer [2015a] that used a multilinear model, and Lüthi et al. [2017] that considered a gaussian process model. Note how this is a chicken-and-egg problem: while statistical models can be useful priors for registration, in order to build such a model one requires a training set of registered meshes in the first place. One solution is to use a bootstrapping approach as considered in *e.g.* [Blanz and Vetter, 1999, Li et al., 2017]. Here an initial registration is used to build a statistical model which in turn is used to improve the registrations, iterating for a few times. Another option is to jointly optimize for the registration and the model, a harder problem that was addressed in Bolkart and Wuhrer [2015b], Zhang et al. [2016].

Finally, a line of work focuses on the case where the face is undergoing a certain motion, a problem usually referred to as *tracking*. Examples of this include Beeler et al. [2011], Bolkart and Wuhrer [2015a], Li et al. [2017]. If the same template is used to track multiple motions then correspondences are established both in space and time, expanding the scope of applications. We contribute in Chapter 4 an approach for efficiently registering a large number of 3D facial scans in motion; a more detailed review of the topic will be presented there.

2.3 Models

Once registered, each face can be represented as a vector $\mathbf{x} = (x_1, y_1, z_1, \ldots, x_n, y_n, z_n) \in \mathbb{R}^{3n}$ with n the number of vertices. We are now in position to analyze a population of 3D faces $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ and study how the surface varies according to the different factors that influence the shape. The next subsections will elaborate on the identity, expression, and high frequency detail models that were built for this purpose.

2.3.1 Modeling Identity Variations

By far the most common approach for modeling identity variations is to use Principal Component Analysis (PCA) [Jolliffe, 1986], assuming a normal distribution of the shapes.

The process of building such a model begins by removing the average face $\bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}_i$ from each sample, *i.e.* $\tilde{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}$. The now centered data is assem-

bled into the columns of a matrix $\mathbf{X} \in \mathbb{R}^{3n \times m}$, and an eigenvalue decomposition of the covariance matrix is performed, *e.g.*by singular value decomposition (SVD) of $\mathbf{X}$. The eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_p$ associated with the non-zero eigenvalues provide an orthogonal basis for a new vector space that is decorrelated and is typically of much lower dimensionality compared to $\mathbb{R}^{3n}$, while the eigenvalues $\lambda_1, \ldots, \lambda_p$ contain the variance of the data in the direction of each eigenvector. The dimensionality can thus be further reduced by keeping the $d < p$ eigenvectors with largest eigenvalues, such that a certain percentage of variance is retained. This process yields a generative model in which novel faces can be synthesized from a vector $\mathbf{w} \in \mathbb{R}^d$ by:

$$x(\mathbf{w}) = \bar{\mathbf{x}} + \mathbf{B}\mathbf{w}, \tag{2.3}$$

where $\mathbf{B} \in \mathbb{R}^{3n \times d}$ contains the d principal eigenvectors and $\mathbf{w}$ is the low dimensional representation of $x(\mathbf{w}) \in \mathbb{R}^{3n}$. Thanks to the Gaussian assumption we can easily estimate the likelihood of each parameter $\mathbf{w}_i$ in $\mathbf{w}$, knowing that $P(\mathbf{w}_i) = exp[-1/2(\frac{\mathbf{w}_i}{\sqrt{\lambda_i}})^2]$. This can be leveraged for regularization, as was done in *e.g.* [Blanz and Vetter, 1999, Aldrian and Smith, 2013, Lewis et al., 2014a, Patel and Smith, 2016, Thies et al., 2016].

The seminal work of Blanz and Vetter [1999] was the first to propose the use of this model for a set of densely corresponded faces. Their so-called 3D Morphable Model (3DMM) consisted of two independent PCA models, one for shape and one for texture, trained from 200 scans of Caucasian subjects. An improved version was made publicly available in Paysan et al. [2009], known as the Basel face model. This is still widely used to date –see *e.g.* [Richardson et al., 2016, Tran et al., 2017a, Zhu et al., 2017, Tewari et al., 2017, Genova et al., 2018, Bas and Smith, 2019], and newer versions have been recently proposed [Gerig et al., 2018]. Other publicly available alternatives include the multi-resolution model of Huber et al. [2016], the models of Brunton et al. [2014b] and the full-head models of Dai et al. [2017] and Li et al. [2017].

Equation 2.3 denotes a *global* model, in that a single parameter vector generates the entire shape. *Local* models of identities have also been explored, where several sub-models encode different regions of the face. This allows more diversity in the generation process and better generalization to unseen data, but the representation is less compact and harder to fit to ambiguous input such as 2D images. The work of Blanz and Vetter [1999] already proposed to build PCA models for five regions of the face, and several other works followed this idea, *e.g.* ter Haar and Veltkamp [2008], De Smet and Van Gool [2011], Brunton et al. [2011].

A drawback of PCA models is that they can only represent shapes that are linear combinations of the training data, and hence require a large training set with sufficient coverage. Booth et al. [2016] built a model from almost 10.000 identities, showing that indeed a larger set can yield models of higher quality. Alternatively, Lüthi et al. [2017] model shape variations as a Gaussian Process, which allows to compensate for the lack of data with manually designed

kernels. In this thesis we approach this by learning models from a large scale dataset acquired from multiple sources as explained in Section 2.1.

2.3.2 Modeling Expression Variations

Modeling expressions is considerably harder as the movement induced by the muscles on the skin can be highly non-linear.

In the area of facial animation the use of parametric models was introduced by Parke [1974], and since then many approaches have been proposed for manipulating a 3D face based on a few parameters. Frequently employed in the film industry is the *blendshape* model [Lewis et al., 2014b], used for example to create Gollum in *The Lord of the Rings* or Benjamin Button in *The Curious Case of Benjamin Button*. The blendshape model is based on a set of d meshes in correspondence showing a single subject in multiple expressions, and generates new faces as a linear combination of these:

$$x(\mathbf{w}) = \mathbf{B}\mathbf{w}, \tag{2.4}$$

where $\mathbf{B} \in \mathbb{R}^{3n \times d}$ contains each expression mesh (the blendshapes) on the columns. This is very similar to the PCA model of Equation 2.3 and follows the same goal: to reduce the dimensionality and limit the range of deformations allowed. Yet, there is an important difference: unlike PCA the bases are not orthogonal with each other, but carry instead semantic meaning. For example, the first basis might generate a smile and the second one a blink. This makes it easier for artists to manipulate and animate the facial mesh, modifying expressions by simply "sliding" the components of the parameter vector $\mathbf{w}$. Such semantic control is not possible with PCA, which in exchange offers better compactness since the basis vectors have no redundancy.

The model can also be expressed as displacements from a selected neutral face of the subject $\mathbf{x}_0$, known as the *delta* blendshape model:

$$x(\mathbf{w}) = \mathbf{x}_0 + \mathbf{B}\mathbf{w}, \tag{2.5}$$

where $\mathbf{B}$ now contains displacements from the neutral face, *i.e.*$(\mathbf{x}_i - \mathbf{x}_0)$ on each column. This allows a more localized control if the blendshapes are focused on a region, and to easily replace the identity by simply modifying the neutral face $\mathbf{x}_0$. In order to build the basis vectors the blendshapes can be either manually modeled by an artist (usually following the Facial Action Coding System (FACS) [Ekman and Friesen, 1978] [2]), scanned from an actor (*e.g.* [Weise et al., 2009]), transferred from the expressions of a different actor [Li et al., 2010] or automatically discovered [Bouaziz et al., 2013, Li et al., 2013].

The blendshape model has been widely adopted thanks to its simplicity and semantic parameterization, but can present artifacts due to its linear nature. In particular, it is easy to obtain unrealistic shapes when activating multiple

2. The Facial Action Coding System is a description of the visible facial movements, classifying each into Action Units (AUs) which describe a contraction or relaxation of a specific muscle.

blendshapes at the same time, and care must be taken when combining them so that they do not interfere with each other. To address this some authors proposed to include bilinear "correction" shapes $\mathbf{b}_{i,j}$ [Lewis et al., 2014b] such that $x(\mathbf{w}) = \mathbf{B}\mathbf{w} + w_i w_j \mathbf{b}_{i,j}$, which are only considered when two blendshapes are simultaneously non-zero. Nonlinear corrections were also considered, *e.g.* Seol et al. [2012], as well as the combination of blendshape models with physical simulation [Barrielle et al., 2016, Cong et al., 2016, Kozlov et al., 2017, Ichim et al., 2017, Barrielle and Stoiber, 2019].

In the computer vision community several works proposed extensions to 3DMM that handle expressive faces. Blanz et al. [2003] encoded expressions as displacements from the corresponding neutral face and built a PCA model from 35 scanned expressions of a single subject. This idea was extended to several subjects in the work of Amberg et al. [2008], while Yang et al. [2011] built instead one PCA model per-expression. Local expression models have also been considered, *e.g.* Decarlo and Metaxas [2000], Tena et al. [2011], Neumann et al. [2013], Wu et al. [2016], Cheng et al. [2017b]. While these can generalize better, they usually cannot capture the co-articulation effects that occur between different parts of the face, *e.g.*the simultaneous movement of the eyes and mouth during a smile. A few non-linear models were recently proposed, such as Li et al. [2017] which includes jaw articulation and corrective blendshapes, or Ranjan et al. [2018], Lombardi et al. [2018], Tran et al. [2019] that used deep neural networks to model the expression space.

2.3.3 Modeling Identity and Expression Variations

Models that can simultaneously encode variations due to identity and expression have clear advantages, as they can generalize to a larger scope of scenarios.

Although it is possible to learn a single PCA model where both novel identities and expressions are synthesized from a unique latent vector, this would require a large amount of training data to properly generalize to unseen instances [Yang et al., 2011]. Only recently a few non-linear models were proposed with this property by leveraging deep learning techniques, *e.g.* Tran and Liu [2018], Bagautdinov et al. [2018], Shamai et al. [2019], Zhou et al. [2019]. More importantly, such an entangled representation does not allow to independently control the generation of identities and expressions, excluding applications that take benefit of this such as animation, tracking and recognition.

The most straightfoward and commonly adopted way of combining these two spaces is by addition:

$$x(\mathbf{w}_{id}, \mathbf{w}_{exp}) = f_{id}(\mathbf{w}_{id}) + f_{exp}(\mathbf{w}_{exp}), \tag{2.6}$$

where $\mathbf{w}_{id} \in \mathbb{R}^{d_{id}}, \mathbf{w}_{exp} \in \mathbb{R}^{d_{exp}}$ are the latent vectors for identity and expression respectively, $f_{id} : \mathbb{R}^{d_{id}} \longrightarrow \mathbb{R}^{3n}$ is a function that decodes the vertices of the

neutral face, and $f_{exp} : \mathbb{R}^{d_{expr}} \longrightarrow \mathbb{R}^{3n}$ decodes displacement vectors from a neutral expression. In the case of a linear model, typically $f_{id}(\mathbf{w}_{id}) = \bar{\mathbf{x}} + \mathbf{B}_{id}\mathbf{w}_{id}$ with $\mathbf{B}_{id}$ a PCA basis for the identity space, and $f_{exp}(\mathbf{w}_{exp}) = \mathbf{B}_{exp}\mathbf{w}_{exp}$ with $\mathbf{B}_{exp}$ a PCA or a blendshape basis for expression. This is the approach followed for example by Amberg et al. [2008], Garrido et al. [2016a], Thies et al. [2016], Tewari et al. [2017], Liu et al. [2018], Kim et al. [2018a], Tewari et al. [2019]. A handful of works have also modeled either the identity or the expression functions as nonlinear, such as Ranjan et al. [2018] where the PCA identity model of Flame [Li et al., 2017] is combined with a deep mesh autoencoder for expressions, or Tran et al. [2018, 2019], Li et al. [2020] that train a deep neural network for each space.

The additive model of Equation 2.6 assumes that the expression displacements are the same for all identities, ignoring the fact that the way an expression is performed depends on the underlying shape. Bilinear [Tenenbaum and Freeman, 2000, Chuang et al., 2002] and multilinear [Vasilescu and Terzopoulos, 2002, Vlasic et al., 2005] models address this through the use of factorization techniques.

Multilinear models are mappings from a *set* of latent variables to the output shape, such that the different factors modulate each other multiplicatively. The model becomes linear when all but one of the factors is held constant.

Let us first recall the basic concepts of multilinear algebra, on which the model is based[3]. A N-th order tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$ is a multi-dimensional array that is indexed by N integers, generalizing the concept of vectors (first-order tensors) and matrices (second-order tensors) to higher dimensions. The higher order analog of rows and columns is called a fiber, obtained by fixing all but one of the indices. For example, given a 3-rd order tensor $\mathcal{Y} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, the vector $\mathbf{y}_{i,:,k} \in \mathbb{R}^{d_2}$, with $i \in [1..d_1], k \in [1..d_3]$ is a mode-2 fiber of $\mathcal{Y}$. A tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times \cdots \times d_n \times \cdots \times d_N}$ can be converted into a matrix through a process called *matricization*, *unfolding* or *flattening*. The mode-n matricization of $\mathcal{X}$ is a matrix $\mathbf{X}_{(n)} \in \mathbb{R}^{d_n \times (d_1 d_2 \ldots d_{n-1} d_{n+1} \ldots d_N)}$ obtained by arranging the mode-n fibers as columns. The mode-n product between a tensor $\mathcal{X}$ and a matrix $\mathbf{U} \in \mathbb{R}^{K \times d_n}$, denoted as $\mathcal{X}' = \mathcal{X} \times_n \mathbf{U}$, is a tensor $\mathcal{X}' \in \mathbb{R}^{d_1 \times \cdots \times K \times \cdots \times d_N}$ obtained by replacing all mode-n fibers $\mathbf{x}$ by $\mathbf{U}\mathbf{x}$.

To build a multilinear model of 3D faces the training set $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ is assembled into a tensor, where one of the dimensions represents the vertices and the rest is arranged according to the modeled factors of variation. For example, a training set of m_i identities each performing m_e expressions can be arranged as a tensor $\mathcal{Y} \in \mathbb{R}^{3n \times m_i \times m_e}$ such that the fiber $\mathcal{Y}_{:,i,j}$ contains the vertices of the i^{th} identity and e^{th} expression, and the fiber $\mathcal{Y}_{k,:,e}$ contains the k-th vertex of all identities when performing the e-th expression.

Higher-Order SVD (HOSVD) [De Lathauwer et al., 2000a] generalizes the concept of SVD to tensors, and can be applied to the centered version $\mathcal{X}'$ of $\mathcal{X}$

3. A more detailed review can be found in Kolda and Bader [2009].

to obtain a decomposition of the form:

$$\mathcal{X}' \approx \mathcal{C} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \ldots \times_N \mathbf{U}_N \tag{2.7}$$

where $\mathbf{U}_i \in \mathbb{R}^{d_i \times p_i}, p_i \leq d_i$ are orthogonal matrices, and $\mathcal{C} \in \mathbb{R}^{p_1 \times \cdots \times p_N}$, called the *core* tensor, models the interaction between the different vector spaces spanned by $\mathbf{U}_i$. As with PCA, the training tensor is first centered such that $\mathcal{X}'_{:,i_1,\ldots,i_N} = \mathcal{X}_{:,i_1,\ldots,i_N} - \bar{\mathbf{x}}$, with $\bar{\mathbf{x}}$ the average vertices among the training set. Since the goal is to model vertices as a function of parameters for the different factors (*e.g.*identity, expression), multilinear models of the face typically do not factor along the mode that corresponds to the vertices [Vlasic et al., 2005], such that

$$\mathcal{X}' \approx \mathcal{M} \times_2 \mathbf{U}_2 \ldots \times_N \mathbf{U}_N. \tag{2.8}$$

A multilinear model of identity and expression will thus generate new faces as

$$x(\mathbf{w}_2, \mathbf{w}_3) = \bar{\mathbf{x}} + \mathcal{M} \times_2 \mathbf{w}_2 \times_3 \mathbf{w}_3, \tag{2.9}$$

where *e.g.*$\mathbf{w}_2 \in \mathbb{R}^{d_{id}}$ are the identity coefficients, $\mathbf{w}_3 \in \mathbb{R}^{d_{exp}}$ the expression coefficients, and d_{id}, d_{exp} the corresponding dimensions.

The model was first applied for 3D faces in Vlasic et al. [2005], where the authors built a bilinear model for identity and expression and a trilinear model that also considers visemes. It has since been used to address multiple tasks, including mesh animation [Wampler et al., 2007], video editing [Dale et al., 2011, Yang et al., 2012], 3D reconstruction [Shi et al., 2014], avatar reconstruction [Cao et al., 2016] and statistical modeling of motion [Bolkart and Wuhrer, 2015a]. A localized version has been proposed in Brunton et al. [2014a], and publicly available models were made in the works of Cao et al. [2013], Brunton et al. [2014b] and Bolkart and Wuhrer [2016]. The recent work of Yang et al. [2020] builds a multilinear model from a dataset of around 900 identities in 20 expressions each.

A main disadvantage of multilinear models is the need to build a full training data tensor, requiring very careful capture conditions in which *e.g.*all identities perform all of the expressions. Already the work of Vlasic et al. [2005] approached this through the use of Probabilistic PCA for tensor completion. Bolkart and Wuhrer [2016] proposed a method to robustly learn a model considering missing data, corrupt data, and incorrect semantic correspondence. Recent work has also explored fully unsupervised approaches: Wang et al. [2017] recover the core tensor using an alternating least squares approach, while Wang et al. [2019] train a deep autoencoder with multilinear structure on three modalities. In Chapter 3 we will present a novel deep autoencoder architecture that allows to train a multilinear model from incomplete data tensors, allowing to handle significantly larger training sets.

Nonlinear models of identity and expression have also been proposed. Wang et al. [2004] use manifold learning techniques to model the different expression styles of individuals. More recently, deep learning approaches have allowed to learn such non-linear models in a scalable manner, including the

fusion network of Jiang et al. [2019], and the work developed in the context of this thesis, which will be introduced in Chapter 5.

2.3.4 Modeling Details

While useful for handling complex problems, low-dimensional parametrizations often restrict the deformations to its coarsest features: mid-scale and high-frequency details are mostly lost in the process. Only recently researchers began to explore a low-dimensional space that can still encode higher-frequency geometry, again thanks to advances in deep learning, *e.g.* [Tran et al., 2018, 2019]. Still, the usual approach is to complement the coarse shape from a parametric model with one or two extra layers that encode out-of-model details. This can include a medium-scale layer for wrinkles and a fine-scale layer for mesoscopic geometry, and the deformations can be represented directly on the (possibly refined) mesh [Garrido et al., 2013, 2016a], as displacement maps [Golovinskiy et al., 2006, Thomas and Taniguchi, 2016, Tran et al., 2019], or as normal maps [Ichim et al., 2015, Lattas et al., 2020].

At the coarsest level, corrective layers can be added to a parametric model to more closely match the input data and capture person-specific idiosyncrasies. To keep the model tractable the correctives can too be encoded in a low-dimensional space, such as the linear spectral basis of Bouaziz et al. [2013], Garrido et al. [2016a] or the automatically learned corrective function of Tewari et al. [2018].

When available, a database of high-resolution scans can be used to learn a generative model of wrinkle formation. Such an approach was followed for example by Golovinskiy et al. [2006] who synthesized displacement maps based on statistical analysis, Ma et al. [2008], Bickel et al. [2008], Bermano et al. [2014] that model displacements as a function of the coarse shape, and Cao et al. [2015] that regress wrinkles based on local appearance. The training dataset can be person-specific [Bickel et al., 2008, Bermano et al., 2014] or person-independent [Golovinskiy et al., 2006, Cao et al., 2015], and can be acquired from less constrained setups such as a single high-quality model [Li et al., 2015] or video data [Garrido et al., 2016a]. When large datasets are available deep learning techniques can also be leveraged, *e.g.* Huynh et al. [2018], Yamaguchi et al. [2018], Tran et al. [2018], Chen et al. [2019].

Since medium- and fine-scale layers correlate with the deformations of the coarser shape, several works proposed to build a mapping between the coarse shape parameters and the finer layers. This was considered for example in Ma et al. [2008] through the use of deformation-driven polynomial displacement maps, Bickel et al. [2008], Ichim et al. [2015] that train a mapping between edge strain and displacement or normal maps, and Garrido et al. [2016a], Yang et al. [2020] by regressing displacements from the coefficients of a parametric model.

Fine-scale details have been traditionally obtained using photometric approaches, *e.g.*shape-from-shading (SfS) [Horn and Brooks, 1989] in the case of a single image. This recovers a normal map which can then be integrated [Quéau

et al., 2018] or used to enhance a pre-computed depth map using methods such as Nehab et al. [2005]. While SfS normally relies on generic priors (such as smoothness or integrability), face-specific models can also be leveraged to alleviate some of its intrinsic limitations. For example, Smith and Hancock [2006] constrain the output to lie in the space of a statistical model of surface normals. The shading-based refinement methods of Valgaerts et al. [2012], Garrido et al. [2013], Shi et al. [2014], Garrido et al. [2016a] follow a coarse-to-fine approach where the results from coarser levels are used to constrain SfS. Data-driven approaches, and in particular recent deep learning techniques, can potentially provide a strong prior to SfS by learning from a large set of examples. Yet, the works presented to date [Shu et al., 2017, Sengupta et al., 2018, Trigeorgis et al., 2017] can only recover overly-smoothed normal distributions. We will present in Chapter 6 a novel approach for normal estimation using deep learning that can recover significantly more accurate results.

3 A Multilinear Autoencoder for 3D Face Model Learning from Large Datasets

Generative models of the 3D facial shape are extensively used in a number of fields that include computer vision –where they serve as priors for ill-posed problems such as 3D reconstruction from images–, computer graphics –where they can be used for the animation of digital characters–, and medical image analysis, where they can be used to distinguish normal from pathological structures. They proved to be benefitial for these tasks as they provide a low-dimensional parameterization for an otherwise complex problem, simplifying both synthesis and inference tasks.

A special kind of models are those that can *decouple* the changes due to natural factors of variation, for instance identity, expression or even age in the case of faces. Such models provide an (ideally) independent parameterization for each of the factors, incorporating a degree of semantic control that enables numerous applications. For example, a decoupled parameterization allows to transfer the expression from one digital character to another one by simply replicating the expression vector, potentially preserving the target's individuality in the performance of the expression [Vlasic et al., 2005, Dale et al., 2011]. Knowing that the identity is constant allows for strong regularization in tracking [Dale et al., 2011, Shi et al., 2014] or learning [Sanyal et al., 2019, Tewari et al., 2019] tasks, a property that will also be leveraged in Chapter 4. Other applications include 3D face and expression recognition [Mpiperis et al., 2008, Liu et al., 2018], expression rectification [Yang et al., 2011], automatic blendshape generation [Cao et al., 2013, 2014a, 2016, Wang et al., 2020] and synthetic data generation for machine learning applications [Han et al., 2017, Shamai et al., 2019].

Multilinear models were proposed with this task in mind. The model, first employed for 3D faces by Vlasic et al. [2005], extends the widely used 3D Morphable Model [Blanz and Vetter, 1999] by assembling the training dataset as a tensor that is organized based on the specified factors, and by replacing PCA with a tensor decomposition method, typically Higher-Order SVD (HOSVD) [1]. This results in a compact model where the different latent vectors influence only one factor (*e.g.*identity or expression), and the interaction between these is modeled in a multiplicative manner.

1. An introduction to multilinear models is provided in Section 2.3.3.

Multilinear models have however one major drawback: in order to build them one needs to assemble the training data into a complete tensor. In the case of faces showing different identities and different expressions, this means that each subject must be scanned performing each of the training expressions. This is clearly not scalable and prohibits the use of scans for which the full set of expressions is not present.

The problem is more pronounced when considering publicly available 3D datasets, whose combination can be large and whose sources and capture protocols can be varied, *e.g*. Cosker et al. [2011], Savran et al. [2008], Yin et al. [2008]. Considering the effort that is involved in capturing accurate scans of the 3D face, these datasets are a valuable source of information and the ability to learn patterns from them can potentially bring forward more powerful models. Yet, there is evidently no control on the type of data that was captured, and as a consequence one cannot build a complete data tensor to train a multilinear model. The large number of scans also demands for computationally efficient and scalable methods, as well as fully automatic registration approaches that could hence result in partially corrupted training data (either in the geometry or in the semantics). The question is then **how to leverage all available training data to build richer multilinear models** –without assuming a complete training data tensor, requiring little to no pre-processing, and taking into account that the scans may be corrupted by geometric noise and/or erroneous labels.

This chapter takes a step towards these goals by proposing a novel framework that can learn a multilinear model of the 3D face from such data. This is achieved through a new autoencoder architecture that combines a CNN-based encoder –thus assuring robustness to corrupt and incomplete data– with a *multilinear decoder*, that can effectively decouple the shape variations over data attributes. It additionally inherits the benefit of scalability that is characteristic of autoencoders. Moreover, using a multilinear model as decoder rather than a generic network allows to explicitly take advantage of redundant training data showing the same factor, and to effectively decouple shape variations in the learned representation.

The proposed approach builds on recent works that use deep neural networks for 3D face modeling. In particular, two of them [Laine et al., 2017, Tewari et al., 2017] have successfully explored the combination of a CNN-based encoder with a linear generative model as decoder for the task of 3D reconstruction of faces from 2D images. We follow a similar strategy however, unlike Tewari et al. [2017] the decoder is *learned* with the rest of the network, and unlike Laine et al. [2017], our learned model generalizes to various factors captured for different subjects.

The method takes as input a set of 3D face scans annotated with labels for each factor, *e.g*.identities and expressions, and provides: (i) A multilinear model, which is able to accurately reconstruct the training data and decouples shape changes due to different factors; (ii) A trained autoencoder capable of

regressing from any 3D face scan to the registered model, thus allowing to efficiently compute correspondences for new data.

Our model performs favorably against other recent approaches that learn multilinear face models from incomplete training data tensors, namely Bolkart and Wuhrer [2016] and Wang et al. [2017]. In particular, we show experimentally that the proposed method is capable of building rich models which achieve a better decoupling of factors. This is demonstrated by a classification rate of synthetically transferred expressions that is over 5% higher than competing methods. While the experiments focus on identity and expression attributes, our formalism readily generalizes to other factors as well.

3.1 Related Work

There is an extensive amount of work on 3D human face modeling, many of which were reviewed in Section 2.3. Here we focus the discussion on works that are most closely related to the proposed approach.

Generative modeling of 3D faces Linear models were first introduced to model face shape in neutral expression along with appearance information in Blanz and Vetter [1999], and later extended to include expression change as a linear factor in Amberg et al. [2008]. These linear models are often called 3D morphable models (3DMM), and have recently been learned from large training sets [Booth et al., 2016] and from craniofacial scans [Dai et al., 2017]. These models do not account for correlations of expression and identity spaces.

Multilinear models were introduced to independently represent the influence of different factors on the facial shape, which allows for expression transfer [Vlasic et al., 2005]. They were later used to edit 2D images and videos with the help of 3D face reconstructions [Dale et al., 2011, Cao et al., 2014b]. FaceWarehouse [Cao et al., 2014b] is a popular publicly available multilinear 3D face model. While multilinear models effectively decouple shape variations due to different factors, they require carefully acquired training data where each subject is captured in every factor.

Li et al. [2017] introduced a generative model learned from a large collection of 3D motion sequences of faces. Pose changes due to skeletal motion is modeled using a skinning approach, while shape changes due to identity, expression, and pose correction are modeled as linear factors similar to 3DMM. Interestingly, they note that it is an open problem to extend tensor-based multilinear models to handle dynamic training data.

We take a step in this direction by deriving an efficient method to learn a multilinear model from an incomplete tensor of training data, that effectively decouples factor effects.

Learning a multilinear model from partial or noisy data Traditionally, multilinear models are learned by assembling a dataset into a tensor and performing tensor decomposition [De Lathauwer et al., 2000a]. This requires each training

face to be present in all factors. Furthermore, noise in the data, registration or labeling affect the quality of the model. While tensor completion methods can be used to solve the problem of incomplete data, they do not scale well in practice, especially if the tensor is dense as in our case [Song et al., 2019].

Two recent methods were proposed to address these problems. Bolkart and Wuhrer [2016] introduced a groupwise optimization to handle both missing and noisy data and was shown to outperform tensor completion methods. However, the approach is computationally costly and hence does not scale to large datasets with high dimensionality in two or more factors. Another work proposed an unsupervised method to compute a multilinear model from partial data [Wang et al., 2017]. While computationally more efficient, it uses a non-standard tensor decomposition that leads to a generative model that does not fully decouple the modes. We will compare to both methods in Section 3.4.

Deep neural networks for 3D face modeling Deep neural networks have experimentally been shown to summarize large groups of data and automatically extract only the relevant features for a large variety of problems, providing an efficient structure for the optimization of large datasets. This motivates the use of deep learning as a scalable and robust alternative for training a multilinear model.

Initial works that used CNN frameworks to recover the 3D shape from a single photograph include Zhu et al. [2016], Richardson et al. [2016, 2017], Tran et al. [2017a], Güler et al. [2017], Sela et al. [2017]. This was predominantly achieved by supervised regression towards the coefficients of a 3DMM, restricting the accuracy of the solution due to the use of synthetic data. The work of Tewari et al. [2017] was the first to frame the generative model as decoder of a neural network. This allowed for end-to-end self-supervised training using an analysis-by-synthesis loss function. Subsequent methods focused on improving the loss formulation, such as Genova et al. [2018], Sengupta et al. [2018], Sanyal et al. [2019]. Unlike the present work, none of these model-based decoders were trained or improved during the CNN optimization.

By the time of publication, only the work of Laine et al. [2017] observed that optimizing an initial PCA model within a deep learning task can yield better results than a fixed model. The work on this chapter follows a similar path, yet unlike Laine et al. [2017] our model is not person-specific and can generalize to arbitrary subjects and expressions. Since then the approach has been adopted by several works. In particular, the simultaneous training of a generative model and a regressor from RGB images holds great potential, as it allows to build fully-unsupervised 3D models from large-scale 2D datasets. In this context, Tewari et al. [2018] learned a corrective space for a fixed identity model, Tewari et al. [2019] learned the full identity space, and Tran et al. [2018, 2019] learned both identity and expression models directly from images.

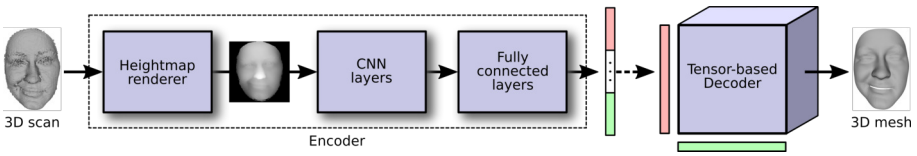


Figure 3.1 – Multilinear Autoencoder architecture. The encoder takes as input a 3D mesh, which is rendered into a heightmap, processed by a deep CNN, and transformed into a latent representation by the fully-connected layers. The decoder splits the latent representation according to the specified factors and performs a multilinear transformation in order to get the output mesh. Both encoder and decoder are optimized during training.

3.2 Overview

The goal of this work is to learn a generative model of faces from a set of labeled 3D scans, that are possibly corrupted by both geometric noise and label errors. To achieve this, we propose an autoencoder architecture with a CNN-based encoder and a multilinear model-based decoder, as illustrated in Figure 3.1 and detailed in the following section.

Input Data To train the autoencoder we consider 3D face scans showing variations in different factors, *e.g.*identity and expression, along with the corresponding labels. Not all combinations of factors are required in the input scans, and part of the training data can be without labels. The input scans are first registered, enabling reconstruction errors between the output meshes and the input scans to be estimated in a consistent way. These registrations need not be precise, as the global nature of training will ensure that isolated errors are averaged out.

Encoder The CNN encoder maps each 3D face scan into a low-dimensional representation that decouples the influence of the different factors on the final shape. Extending CNNs to unorganized 3D geometric data is an active field of research (see *e.g.* Wu et al. [2020]) and beyond the scope of this work. Instead, we take advantage of the fact that 3D faces can be mapped onto 2D images for which regular CNNs apply. Hence, the first step of the encoder is to project input 3D scans into grayscale images that contain depth information. The remainder of the encoder consists of a convolutional neural network followed by fully connected layers, which transform the depth image into a d-dimensional vector with the concatenated coefficients for each mode.

Decoder The multilinear decoder splits the output of the encoder according to the factors, applies mode-n multiplication between these latent vectors and the core tensor, and adds a previously computed average face, as normally done with multilinear models (see Sections 3.3.1 and 2.3.3). The output of the

decoder are 3D vertex coordinates that, combined with the connectivity of the average face, define a 3D mesh. The key here is that all of these operations can be written as layers of a neural network, thus allowing to update the values of the core tensor along with the rest of the network.

Training During training both the CNN encoder and the multilinear decoder are optimized. In addition to a generative loss that accounts for reconstruction errors, we optimize a latent loss that measures whether input faces with the same labels are mapped onto close-by points in parameter space, hence enforcing shape variations to be decoupled with respect to the different factors. The space that models face variations is large compared to the available training data and a good initialization is thus required. To this aim, both encoder and decoder are pre-trained, as detailed in Section 3.3.3.

Once the autoencoder has been trained, the multilinear model can be extracted from the decoder and treated as a classic multilinear model. In addition, the trained encoder can be used to regress any 3D scan to the model, thereby allowing to efficiently register new data.

3.3 Multilinear Autoencoder

We now describe the proposed autoencoder architecture that allows to learn k modes of variation in the input face data through a multilinear model.

3.3.1 Multilinear Model as a Decoder

In a multilinear model a face is represented by a set of vectors $\{\mathbf{w}_2, \ldots, \mathbf{w}_{k+1}\}$, $\mathbf{w}_j \in \mathbb{R}^{d_j}$, where k is the number of linear modes attached to faces in the model. Let $\mathbf{x} \in \mathbb{R}^{3n}$ be the vector of 3D coordinates associated with the n vertices of a face mesh, then the multilinear model relates the latent k factors $\mathbf{w}_j$ with the 3D face $\mathbf{x}$ by:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathcal{M} \times_2 \mathbf{w}_2^T \times_3 \mathbf{w}_3^T \ldots \times_{k+1} \mathbf{w}_{k+1}^T, \tag{3.1}$$

where $\bar{\mathbf{x}}$ is the average training face, $\mathcal{M} \in \mathbb{R}^{3n \times d_2 \times d_3 \times \ldots d_{k+1}}$ is a tensor that combines the linear modes $\mathbf{w}_j$ called the *core tensor*, and $\times_j$ is the product of $\mathcal{M}$ and a vector along mode j. The model is therefore represented by the entries of $\mathcal{M}$ in addition to the set of coefficients $\mathbf{w}_j^{(i)}$ for the i-th face and the j-th factor in the training set.

An interesting property of tensors states that [Kolda and Bader, 2009]

$$\begin{aligned} \mathcal{Y} = \mathcal{C} \times_1 \mathbf{A}^1 \times_2 \mathbf{A}^2 \ldots \times_{k+1} \mathbf{A}^{k+1} \Leftrightarrow \\ \mathbf{Y}_{(n)} = \mathbf{A}^n \mathbf{C}_{(n)} \left(\mathbf{A}^{k+1} \otimes \ldots \otimes \mathbf{A}^{n+1} \otimes \mathbf{A}^{n-1} \otimes \ldots \mathbf{A}^1 \right), \end{aligned} \tag{3.2}$$

where $\mathcal{Y}$ and $\mathcal{C}$ are tensors, $\otimes$ is the Kronecker product, $\mathbf{A}^n$ are matrices of appropriate dimensions, and $\mathbf{C}_{(n)}, \mathbf{Y}_{(n)}$ are the matricizations of $\mathcal{C}$ and $\mathcal{Y}$ containing

the mode-n fibers as columns. In particular, for the model vector coefficents $\mathbf{w}_j$, a training face mesh $\mathbf{x}$, and mode $n = 1$,

$$\begin{aligned} \mathbf{x} = \bar{\mathbf{x}} + \mathcal{C} \times_1 \mathbf{w}_1^T \times_2 \mathbf{w}_2^T \dots \times_{k+1} \mathbf{w}_{k+1}^T \Leftrightarrow \\ \mathbf{x} = \bar{\mathbf{x}} + \mathbf{w}_1^T \mathbf{C}_{(1)} \left(\mathbf{w}_{k+1}^T \otimes \dots \otimes \mathbf{w}_2^T \right). \end{aligned} \tag{3.3}$$

Recall that the vector $\mathbf{w}_1$ corresponds to the vertices factor, and as such it is "absorbed" by the core tensor of the model (see Section 2.3.3). Hence, $\mathcal{M} = \mathcal{C} \times_1 \mathbf{w}_1 \Leftrightarrow \mathbf{M}_{(1)} = \mathbf{w}_1^T \mathbf{C}_{(1)}$, and thus Equation 3.1 can be written as

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{M}_{(1)} \left(\mathbf{w}_{k+1}^T \otimes \dots \otimes \mathbf{w}_2^T \right). \tag{3.4}$$

By writing the transformation $\mathbf{M}_{(1)}(\otimes_{j=k+1}^{2} \mathbf{w}_j)$ as layers of a neural network, we can refine the multilinear model $\mathcal{M}$ at the same time we optimize for the reconstructions. After training, it suffices to recover $\mathbf{M}_{(1)}$ from the network and fold it back into a tensor $\mathcal{M}$ to obtain the new multilinear model.

3.3.2 Learning the Multilinear Model

The training process seeks to obtain good reconstructions of the data, while at the same time decoupling the latent representation with respect to the factors of variation. Hence, we will use two loss functions: a geometric loss that measures the reconstruction error, and a latent loss that softly evaluates how decoupled the latent space is, by measuring how close two embeddings with the same label are.

Generative loss Given a training set $\mathbf{X}$ of faces, the loss of a multilinear model $\mathcal{M}$ over a mini-batch $\mathbf{X}_b \subseteq \mathbf{X}$ is measured as the average error between the reconstructions of the model and the observed faces $\mathbf{x}_i$:

$$\mathcal{L}_G = \frac{1}{|\mathbf{X}_b|} \sum_{\mathbf{x}_i \in \mathbf{X}_b} \left\| \mathbf{x}_i - \left(\bar{\mathbf{x}} + \mathcal{M} \times_2 \mathbf{w}_2^{(i)} \dots \times_{k+1} \mathbf{w}_{k+1}^{(i)} \right) \right\|_2^2, \tag{3.5}$$

or equivalently (Eq. 3.4)

$$\mathcal{L}_G = \frac{1}{|\mathbf{X}_b|} \sum_{\mathbf{x}_i \in \mathbf{X}_b} \left\| \mathbf{x}_i - \left(\bar{\mathbf{x}} + \mathbf{M}_{(1)} \left(\bigotimes_{j=k+1}^{2} \mathbf{w}_j^{(i)} \right) \right) \right\|_2^2. \tag{3.6}$$

Note that Equation 3.6 is not a decomposition of the data tensor, but a soft constraint that allows to represent a given label in mode j by different coefficients $\mathbf{w}_j^{(i)}$ for different faces $\mathbf{x}_i$. This can be an advantage when the labeling is not trust-worthy, allowing for flexibility in the factor separation.

Latent loss We observed that a simple reconstruction loss is not sufficient to ensure a decoupled space, as originally guaranteed by the tensor decomposition. This is expected, since Equation 3.6 does not evaluate the coefficients

$\mathbf{w}_j$ directly but the reconstruction they yield, therefore allowing the output vertices to be arbitrarily affected by any mode.

To overcome this, we define a loss function that softly constrains the latent parameters. Given a training batch $\mathbf{X}_b$, for each mode j we consider the *labeled* subset $\mathbf{W}_{lbl}^{(j)} = \{\mathbf{w}_j^{(1)}, \ldots, \mathbf{w}_j^{(m)}\}$, $m \leq |\mathbf{X}_b|$, of mode-j latent weights in the batch, obtained after encoding $\mathbf{X}_b$; for example, all the batch weights that have an "expression" label associated with it. Let $\tilde{\mathbf{W}}_j^{(i)} = \{\mathbf{w}_j^{(i_1)}, \mathbf{w}_j^{(i_2)}, \ldots\}$ be the set of all mode-j coefficients *in the full training dataset* that share the same label in mode j as $\mathbf{w}_j^{(i)} \in \mathbf{W}_{lbl}^{(j)}$ (for example, all the training faces that were labeled "happy"). Then the function writes:

$$\mathcal{L}_L^{(j)} = \frac{1}{|\mathbf{W}_{lbl}^{(j)}|} \sum_{\mathbf{w}_j^{(i)} \in \mathbf{W}_{lbl}^{(j)}} \frac{1}{|\tilde{\mathbf{W}}_j^{(i)}|} \sum_{\mathbf{w}_j^{(p)} \in \tilde{\mathbf{W}}_j^{(i)}} \left\| \mathbf{w}_j^{(i)} - \mathbf{w}_j^{(p)} \right\|_2^2, \tag{3.7}$$

where the average over coefficients accounts for very different sizes of the sets $\mathbf{W}_j^{(i)}$. The latent loss is then the sum of the loss of each mode:

$$\mathcal{L}_L = \sum_{j=2}^{k+1} \mathcal{L}_L^{(j)}. \tag{3.8}$$

Note that the loss is calculated on each batch *over the full training set*, and thus the gradient must also be globally computed as:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}_j^{(i)}} \mathcal{L}_L^{(j)} &= 2 \sum_{\mathbf{w}_j^{(p)} \in \tilde{\mathbf{W}}_j^{(i)}} \frac{1}{|\tilde{\mathbf{W}}_j^{(i)}|} \left(\mathbf{w}_j^{(i)} - \mathbf{w}_j^{(p)} \right) - 2 \sum_{\mathbf{w}_j^{(p)} \in \tilde{\mathbf{W}}_j^{(i)}} \frac{1}{|\tilde{\mathbf{W}}_j^{(p)}|} \left(\mathbf{w}_j^{(p)} - \mathbf{w}_j^{(i)} \right) \\ &= 2 \sum_{\mathbf{w}_j^{(p)} \in \tilde{\mathbf{W}}_j^{(i)}} \frac{1}{|\tilde{\mathbf{W}}_j^{(i)}|} \left(\mathbf{w}_j^{(i)} - \mathbf{w}_j^{(p)} \right) + 2 \sum_{\mathbf{w}_j^{(p)} \in \tilde{\mathbf{W}}_j^{(i)}} \frac{1}{|\tilde{\mathbf{W}}_j^{(p)}|} \left(\mathbf{w}_j^{(i)} - \mathbf{w}_j^{(p)} \right) \\ &= 4 \frac{1}{|\tilde{\mathbf{W}}_j^{(i)}|} \sum_{\mathbf{w}_j^{(p)} \in \tilde{\mathbf{W}}_j^{(i)}} \left(\mathbf{w}_j^{(i)} - \mathbf{w}_j^{(p)} \right). \end{aligned} \tag{3.9}$$

Here, the second term accounts for the fact that $\mathbf{w}_j^{(i)}$ will appear in the set $\tilde{\mathbf{W}}_j^{(p)}$, and the last line considers $|\tilde{\mathbf{W}}_j^{(p)}| = |\tilde{\mathbf{W}}_j^{(i)}|$.

3.3.3 Architecture

CNN Encoder The encoder transforms the 3D face input data into a vector $\mathbf{w} \in \mathbb{R}^{d_2 + \cdots + d_{k+1}}$ that contains the concatenated model coefficients, *i.e.*the latent parameters of the face. The first layer of the network takes as input a 3D scan and converts it into a 2D image that encodes heights from a fixed plane. The regression from the 2D heightmap to the model coefficients is implemented using a ResNet-18 [He et al., 2016] which reduces the image to a 256-dimensional

vector, after which three fully-connected layers perform the regression towards the coefficient vector $\mathbf{w}$ of the specified dimensions.

Multilinear Decoder The multilinear decoder takes as input the vector $\mathbf{w}$, which is seen as a concatenation of mode coefficients $\mathbf{w} = \{\mathbf{w}_2, \mathbf{w}_3, \ldots, \mathbf{w}_{k+1}\}$, and transforms it into 3D vertex coordinates by performing mode multiplications with the core tensor. As explained in Section 3.3.1, this operation can be written as the product between the matricized version of the tensor $\mathbf{M}_{(1)}$ and the Kronecker product of each mode coefficient (Equation 3.4). Therefore, in order to learn the parameters of the core tensor $\mathcal{M}$ we implement each of these operations as a layer in the network, and allow the linear module $\mathbf{M}_{(1)}$ to be optimized with the rest of the parameters. This way we benefit from the capacity of neural networks to robustly summarize the representative aspects of an entire dataset, and from the associated optimization machinery to find the model in a scalable manner.

Estimation The multilinear autoencoder estimation proceeds in two stages. First, we initialize both CNN encoder and multilinear decoder, as our training data is limited with respect to the number of parameters. Initializing the multilinear decoder with random values did not yield good results in our experiments, particularly in terms of decoupling. Hence we initialize by performing Higher Order Singular Value Decomposition (HOSVD) [De Lathauwer et al., 2000a] on a complete subset of the data, *i.e.*a subset in which all the factors of variation are present for all elements. Note that this enforces a limit on the dimensionality of the latent vectors, since now they cannot be greater than the amount of samples for each factor in the initial tensor. To subsequently pre-train the CNN encoder, we optimize it separately using the generative loss in Equation 3.5 with the fixed initial multilinear model, and with both registered and unregistered scans to augment the training data.

In the second stage the full network is optimized with all available face data. This is achieved by minimizing the following combined generative and latent loss:

$$\underset{\mathbf{M}_{(1)}, \{\mathbf{w}_j^{(i)}\}}{\arg\min}\ \mathcal{L}_G + \lambda \mathcal{L}_L, \tag{3.10}$$

where λ is a scalar that weighs the contribution of the latent loss.

3.4 Evaluation

We evaluate both the generative model that is extracted from the decoder as well as the full autoencoder that can be used for regression.

We begin by presenting implementation details (Section 3.4.1), the datasets employed (Section 3.4.2), and the proposed evaluation protocol that analizes the quality of the generative model (Section 3.4.3). In Section 3.4.4 we show results using these metrics over alternative latent weights and model dimensions, as well as comparisons to state-of-the art methods that learn multilinear 3D

face models from incomplete data. Section 3.4.5 evaluates next the multilinear autoencoder and its ability to register raw scans into the new model. Finally, Section 3.4.6 showcases a few applications of the multilinear autoencoder.

3.4.1 Implementation Details

To pre-train the encoder and to learn the generative model during fine-tuning we use the *AdaDelta* algorithm [Zeiler, 2012], with parameters as provided in the paper. We use a mini-batch size of 64, a learning rate of 0.01 for pre-training and a learning rate of 1 for training the autoencoder. The encoder was pre-trained for 100 epochs and the autoencoder was fine-tuned for 200 epochs. Unless otherwise specified, we use $\lambda = 1$ in Equation 3.10, and set the dimensions of identity and expression spaces to 89 and 25 respectively, which is the maximum allowed by the initial tensor (see Section 3.3.3). The framework was implemented in PyTorch v1.0.1 [Paszke et al., 2019], and the experiments were run using a NVidia GeForce GTX 1080 GPU. For the facial mesh template we used a cropped version of the publicly available Digital Emily [Alexander et al., 2010], which consisted of $n = 10057$ vertices (see Figure 2.1).

3.4.2 Datasets

Training data for initialization We use BU-3DFE [Yin et al., 2006] and Bosphorus [Savran et al., 2008] datasets for initialization, as these come with manually annotated landmarks that simplify pre-processing. The data is registered using Optimal Step NICP [Amberg et al., 2007], initialized with Laplacian deformation using the provided landmarks. In the case of BU-3DFE we register against the "raw" version of the scans, switching to the post-processed version provided by the authors only in the cases where this failed. In total we registered 2499 scans from BU-3DFE and 2698 from Bosphorus. To initialize the decoder we run HOSVD [De Lathauwer et al., 2000a] on a data tensor built from BU-3DFE, which provides 100 identities performing 25 expressions each: the seven prototypical expressions [2], with each non-neutral expression in four levels of intensity. We consider each intensity as a distinct expression and remove subjects that belong to the testing set, resulting in a training data tensor of size $3 * 10057 \times 89 \times 25$ [3]. The CNN encoder is pretrained with both BU-3DFE and Bosphorus. To augment the training data we randomly rotate each face by an angle $\theta \in [-10°; 10°]$ in yaw, pitch or roll axes, and apply a random scale in $[0.95; 1.05]$. Furthermore, we use both the registered data and the corresponding raw 3D scans, for which the registered versions allow to recover ground truth vertex correspondences for training. This augmentation allows the CNN encoder to learn richer feature extractors, as the raw scans

2. Neutral, Angry, Disgust, Fear, Happy, Sad, Surprise.
3. We used 89 out of 100 identities: ten subjects were left out for testing, and one subject was left out of the initial tensor as one of the expressions did not register succesfully.

contain larger geometric errors, holes and extra parts such as hair and the neck. This results in a set of 99,000 heightmap images for pretraining.

Training data for model optimization We demonstrate the capabilities of the multilinear autoencoder (MAE) trained on two different datasets. A first MAE is learned from static data, using the combined Bosphorus and BU-3DFE databases, for a total of 4500 meshes. We will refer to this MAE as *Bu3+Bosph*. We use seven labels from BU-3DFE which correspond to the highest intensity of each expression; the lower intensities are left unlabeled. For Bosphorus we label the seven prototypical expressions as well as the action units, with the exception of action units 43 and 44 which are not correctly captured by the registration. The second MAE is learned by combining the previous with the dynamic database D3DFACS [Cosker et al., 2011] using the publicly available registrations of Li et al. [2017] [4]. Although there is redundancy in consecutive frames, this allows to test a scenario where MAE is trained on a considerably larger training set. We will refer to this as *Bu3+Bosph+D3D*. The dataset is sparsely labeled by considering the first three frames of each sequence as the neutral expression, and the five frames located around the middle as peak frames, which are assigned the facial action unit of the sequence (establishing a semantic correspondence with the expressions in Bosphorus). In total, *Bu3+Bosph+D3D* is trained from 49811 scans, an order of magnitude larger than the training sets used in previous methods [Bolkart and Wuhrer, 2016, Wang et al., 2017].

Test data We leave 10 subjects out from BU-3DFE and 10 from Bosphorus, and test both *Bu3+Bosph* and *Bu3+Bosph+D3D* on these. The testing subjects were selected among those whose registrations were of good quality (manually verified), while keeping a balance between male/female subjects as well as different ethnicities. [5]

3.4.3 Evaluation Protocol

We measure the quality of the generative models using the metrics *generalization* and *specificity* [Davies et al., 2008]. *Generalization* measures the ability of the model to adapt to unseen data, and is evaluated by projecting test data into the model space and calculating the reconstruction error. To provide a common framework for comparisons, this is implemented by iteratively fixing one space and finding the optimal coefficients for the other one [Vlasic et al., 2005]. We ignore border vertices during evaluation as these contain noise due to the registration process. *Specificity* measures whether only valid members of the shape class are modeled, or in other words, the model's suitability for

4. We registered our template to one of the frames using Amberg et al. [2007] and transferred the correspondences to the rest of the dataset.

5. We use the following identities: F0007, F0013, F0043, F0045, F0056, M0012, M0015, M0027, M0037, M0038 for BU-3DFE, bs003, bs024, bs032, bs038, bs081, bs086, bs090, bs092, bs095, bs101 for Bosphorus.

generating synthetic data. To evaluate specificity, we assume the data to follow independent normal distributions in identity and expression spaces and sample 1000 faces. For each randomly drawn sample we measure its mean vertex distance to all elements in the training data and keep the minimum value; specificity is defined as the average of this process over all synthetically generated faces. To compute the normal distribution we consider the sample mean and standard deviation based on the training data. We account for an imbalanced number of labels by first grouping the coefficients by label, summarizing each group by its medoid, and computing the normal distribution based on per-group values.

There is no standard metric to evaluate decoupling, and hence we propose here a protocol that was adapted from Ghosh et al. [2017]. We first train an external classifier to recognize the seven prototypical expressions (anger, happiness, disgust, sadness, fear, surprise and neutral) given an input image with a rendered mesh [6]. The evaluation proceeds as follows. We obtain identity and expression weights for each sample in the testing split of BU-3DFE by iterative registration. We regularize this process with a Tikhonov regularization term,

$$\sum_{k=1}^{d_{id}} \left(\frac{\mathbf{w}_2(k) - \bar{\mathbf{w}}_2(k)}{\sigma_2(k)} \right)^2 \tag{3.11}$$

for identity and similarly defined for expression, where $\sigma_2(k)$ denotes the standard deviation for identity coefficients. We further regularize by simultaneously registering all the expressions of a same identity (*i.e.*we solve for a unique identity weight $\mathbf{w}_2$ while fixing several expression weights $\mathbf{w}_3^{(i)}$ in a single system), and again ignore the border vertices due to noisy registrations. Once the identity and expression coefficients were recovered, we transfer the seven expressions of one test identity to all the other identities, and repeat the process for each subject in the testing set. Expression transfer is performed by replacing the expression weight $\mathbf{w}_3$ with that of the current source face. Finally, we let the classifier measure whether the known transferred expression was preserved, and report the average accuracy of the classifier.

3.4.4 Generative Model Evaluation

This section shows results on the quality of the learned generative model under different configurations, as well as comparisons to classic tensor decomposition and two state-of-the-art methods on multilinear model learning of 3D faces from incomplete data.

Influence of the latent loss We first measure how different values of λ affect the output model, both for *Bu3+Bosph* and *Bu3+Bosph+D3D* . Results are shown in Tables 3.1 and 3.2. As expected, greater values of λ result in progressively

6. We render normal maps as we found this to work better for recognition. The classifier is trained using BU-3DFE and Bosphorus which provide the necessary labels.

	Bu3+Bosph		
λ	Generalization	Specificity	Expression
0.1	1.06	**3.33**	32.54
1	**1.04**	3.47	**58.41**
10	1.09	3.41	57.30

Table 3.1 – Influence of the latent loss on the *Bu3+Bosph* model. Median generalization error (mm), specificity error (mm) and percentage of correct classifications after expression transfer. Best values in bold.

	Bu3+Bosph+D3D		
λ	Generalization	Specificity	Expression
0.1	1.00	3.38	47.46
1	**1.00**	3.61	**50.16**
10	0.99	**3.32**	45.87

Table 3.2 – Influence of the latent loss on the *Bu3+Bosph+D3D* model. Median generalization error (mm), specificity error (mm) and percentage of correct classifications after expression transfer. Best values in bold.

better decoupling of the spaces, but it appears to saturate at one point. An illustration of the effect of λ on the *Bu3-Bosph* model is shown in Figure 3.2. All selected models produce plausible synthetic faces, but there is a clear decrease in the quality of the transfers when the value of λ is too low. Interestingly, larger values of λ also appear to improve generalization, which suggests that the latent loss is acting as a regularizer that can help to better reconstruct unseen data. We select $\lambda = 1$ for the following experiments.

Compactness Ideally a model should perform well using a small number of parameters. We thus evaluate generalization, specificity and expression transfer under different latent dimensions, varying either the identity or the expression space. For this experiment we re-use the pretrained encoder from the previous evaluation (trained on dimensions $89 - 25$ for identity and expression) by simply removing the corresponding rows in the last linear layer of the encoder. The results can be found in Table 3.3. As expected, lower dimensions increase the error in generalization, although this is more significant when varying the identity space. On the other hand, the expression transfer capacity, as captured by the current evaluation, does not appear to vary much among the different dimensions. A change can be observed under 20 dimensions for the identity space (last row in Table 3.3). This could be explained by the fact that Euclidean distances between vectors, as required by Equation 3.7, are more significant when the dimensionality is lower. On the other hand, a correct expression transfer depends also on correctly recovering the coefficients and it is thus

#Id	*#Expr*	Generalization	Specificity	Expression
89	25	1.04	3.47	58.41
89	20	1.07	3.57	57.62
89	15	1.11	3.51	59.52
89	10	1.19	3.77	57.46
65	20	1.23	3.43	60.00
50	20	1.42	3.18	51.59
35	20	1.64	3.33	53.33
20	20	1.87	3.57	64.93

Table 3.3 – Influence of latent size for identity and expression spaces: median generalization error (mm), specificity error (mm) and percentage of correct classifications after expression transfer.

related to how well the model generalizes, which might explain why the value sometimes fluctuates.

Comparison to standard tensor decomposition We compare here to standard tensor decomposition methods, namely higher-order SVD (HOSVD) [De Lathauwer et al., 2000a] and higher-order orthogonal iteration (HOOI) [De Lathauwer et al., 2000b]. Both methods require complete data tensors to perform a Tucker decomposition. For a fair comparison MAE is both pre-trained and trained using BU-3DFE alone, with distinct labels assigned to the different expression intensities. We refer to this version as "MAE - `bu3`".

Results are shown in the top three rows of Table 3.4. Note that HOSVD is the method we use for initialization of the decoder, and thus we show here

(a) $\lambda = 0.1$

(b) $\lambda = 1$

Figure 3.2 – Influence of the latent loss on expression transfer. From left to right: input mesh, transferred expressons: angry, disgust, fear, happy, sad, surprise. Lower values of λ can sometimes fail to properly decouple the latent space, and hence transferring expressions does not preserve semantics.

Model	Generalization	Specificity	Expression
HOSVD	1.20	4.05	51.59
HOOI	1.20	4.01	51.75
MAE - `bu3`	**1.13**	**3.76**	**61.43**
MAE - `bu3-bosph`	1.04	**3.47**	58.41
MAE - `bu3-bosph-d3d`	**1.00**	3.61	50.16

Table 3.4 – Comparison against standard tensor decomposition methods, and influence of training data size. The top three rows show comparisons against classic tensor decomposition methods, using the same training dataset. The bottom rows show improvements obtained when training with larger datasets that cannot be assembled as a tensor. In terms of median generalization error (mm), specificity error (mm), and percentage of correct classifications for expression transfer.

that the proposed training indeed improves the initial model, even when no additional data is used. We further compare against HOOI since it shares our goal of enhancing an initial model provided by HOSVD, which is achieved through an iterative approach. We observe from Table 3.4 that all metrics are improved compared to both tensor decomposition methods. This includes the expression transfer capacity thanks to the addition of the latent loss, with a correct classification value that is almost 10% higher.

Effect of training data The main benefit of our approach is the ability to train with large datasets that do not necessarily form a complete data tensor. Hence, we show in the bottom of Table 3.4 the improvements that can be attained by comparing against different training data sizes. While MAE-`bu3`, HOSVD and HOOI were trained on 2225 samples, MAE-`bu3-bosph` was trained on 4500 and MAE-`bu3-bosph-D3D` on 49811 scans. We can see from Table 3.4 that the ability to generalize to unseen data is greatly improved, as well as the specificity values, showing the benefit of leveraging all available training data.

Comparison to state-of-the-art Finally, we compare to two closely related works that learn multilinear models of 3D faces from incomplete data tensors: RMM [Bolkart and Wuhrer, 2016] and Wang et al. [2017]. We run RMM on our own registration using the publicly available code with default parameters. We use the same subset from BU-3DFE and Bosphorus as in the published model since this was already proven to work correctly for RMM, except we remove the testing identities that were used for the previous experiments (see Section 3.4.2). In particular, we use 184 identites from BU-3DFE and Bosphorus, and 7 expressions from BU-3DFE (with the highest intensities) plus 23 expressions from Bosphorus. The latent dimensions are set to 23 and 6 for identity and expression respectively, as in the published RMM model. We build a model using this setting for the method of Wang et al. [2017] with code provided by the authors, and train MAE on this data and with same

Method	Generalization	Specificity	Expression
Bolkart and Wuhrer [2016]	1.71	3.44	56.51
Wang et al. [2017]	**1.53**	3.57	13.65
MAE	2.17	**3.42**	**61.59**

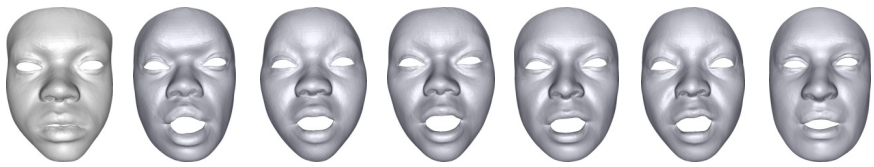
Table 3.5 – Comparison between state-of-the-art and the MAE decoder, in terms of median generalization error (mm), specificity error (mm), and percentage of correct classifications for expression transfer.

dimensions.

Table 3.5 shows the results obtained. We can see that our method outperforms the other two in terms of specificity and expression transfer. Figure 3.3 shows an example of expression transfer results for the three methods. Note that while RMM and MAE achieve visually plausible results, Wang et al. [2017] gives noisy faces that do not preserve identities, as their tensor decomposition does not yield a good decoupling of the different modes.

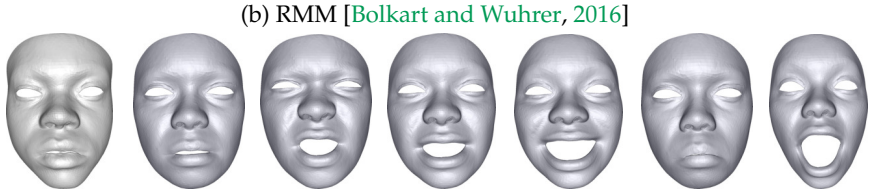
3.4.5 Multilinear Autoencoder Evaluation

We now evaluate the multilinear autoencoder, including the encoder that can efficiently regress into the learned multilinear model. We start by dis-



(a) Wang et al. [2017]

(b) RMM [Bolkart and Wuhrer, 2016]



(c) MAE

Figure 3.3 – Qualitative comparison between state-of-the-art and MAE. From left to right: original scan, transferred expressions: angry, disgust, fear, happy, sad, surprise.

Fitting method	Generalization
Encoder only	3.17
Iterative - average initialization	1.04
Iterative - encoder initialization	**1.01**

Table 3.6 – Generalization value (mm) under different fitting methods: using only the encoder; using the iterative approach with average identity and expression weights for initialization; and using the iterative approach with encoder initialization.

cussing the computation times of the method, and afterwards consider the capacity of the model to register unseen data.

Computation times Computing the core tensor using HOSVD for $89 - 25$ dimensions requires on average 3 seconds. Pre-training the encoder takes about $40ms$ per mini-batch and ~ 3 minutes per epoch for our data (including data-loading time). Fine-tuning the *Bu3+Bosph* model takes about 13 seconds on average per epoch, while fine-tuning *Bu3+Bosph+D3D* takes around 2 minutes per epoch. Generating each depth image takes $\sim 20ms$ for the registered data. Once the training is finished, regressing from a single raw scan to 3D vertices requires around 250ms for a batch of size 64.

Generalization with the autoencoder For a fair comparison, all generalization values presented in the previous section were obtained by iterative fitting initialized with the mean identity or expression vector. We show in Table 3.6 two alternatives for this that leverage the encoder: by using the encoder for registration, and by using the encoder as initialization for the iterative method. We can see from Table 3.6 that the generalization value using the encoder alone is significantly higher than the rest. On the other hand this can be done very efficiently when performed on the GPU. Thus, we leverage the encoder for efficiently initializing the iterative process, achieving an even better generalization value as shown in the bottom row of Table 3.6.

Registration of raw scans We evaluate the reconstructions of the test set obtained by regressing with the multilinear autoencoder using the original raw scan images. We consider the initial registered versions of the scans as ground-truth even though this might not be exact, since the registrations were manually verified to be globally correct. This gives a median per-vertex Euclidean error of $3.86mm$ for *Bu3+Bosph*, and a median per-vertex Euclidean error of $3.85mm$ for *Bu3+Bosph+D3D*. Figure 3.4 shows one example of raw scan registration (from a different dataset). Even though the error is relatively high, we observe that the outputs are in general visually close to the expected identity and expression, and could be used as initializations for optimization-based refinements.

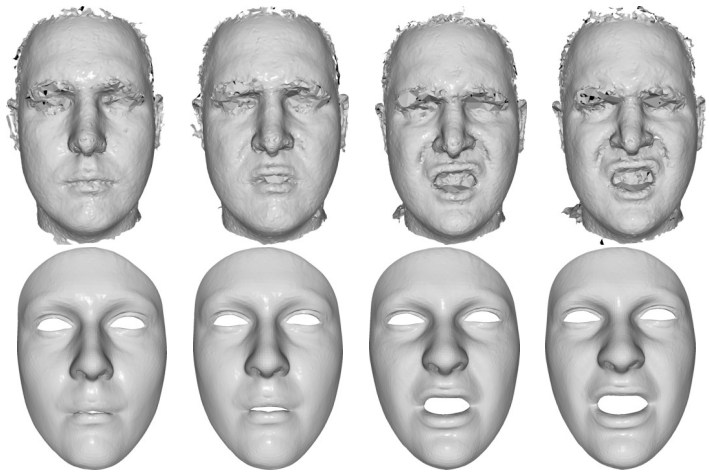


Figure 3.4 – Registration of raw scans using the encoder. Top: input scans. Bottom: registered results.

3.4.6 Applications

We finish this section by showcasing two possible applications of the multilinear autoencoder.

Identity recognition from 3D mesh sequences Face recognition from 3D or 4D data has received increasing attention thanks to both a wider availability of depth data, as well as the potential it holds in overcoming the inherent limitations of 2D images. We test the ability of MAE to perform recognition as follows. Given an input mesh sequence, we use the encoder on each frame to recover identity and expression coefficients. We then measure the Euclidean distance between the recovered identity coefficient and each of the coefficients in a training database, keeping the identity label of the closest sample. Finally, we set as identity label of the sequence the label that was guessed by the majority of the frames.

We base the evaluation on the protocol used in Sun et al. [2010] and Alashkar et al. [2016]. In particular, we take 60 identities from the BU-4DFE dataset [Yin et al., 2008] and perform two experiments: *Expression Dependent* (ED) and *Expression Independent* (EI). For the ED experiment we split each mesh sequence in half, using the first half as training database and the second half for testing. For the EI experiment we use one expression sequence for training and the rest of the expressions for testing, repeating for each of the six expressions. In both cases we report the percentage of correct classifications.

We show results in Tables 3.7 and 3.8, where we also compare to the values reported by Sun et al. [2010] and Alashkar et al. [2016]. Our simple recognition method achieves the best results for the ED experiment, as well as competitive results on the EI experiment. Note that unlike these methods our approach is very efficient, as it requires only one pass of the encoder and an L2 distance

Method	Recognition Rate (%)
LLE on static 3D (reported in [Sun et al., 2010])	82.34
PCA on static 3D (reported in [Sun et al., 2010])	80.78
LDA on static 3D (reported in [Sun et al., 2010])	91.37
Sun et al. [2010]	97.47
Alashkar et al. [2016]	**100.00**
MAE	**100.00**

Table 3.7 – Identity recognition from 3D mesh sequences (ED experiment): comparison against Sun et al. [2010], Alashkar et al. [2016] and approaches using static 3D data reported in Sun et al. [2010].

Method	AN	DI	FE	HA	SA	SU	Avg.
Sun et al. [2010]	94.12	94.09	94.45	94.52	93.87	95.02	94.37
Alashkar et al. [2016]	85.20	87.70	83.49	83.36	84.86	80.49	84.13
MAE	87.00	90.33	86.67	86.67	87.33	86.00	87.33

Table 3.8 – Identity recognition from 3D mesh sequences (EI experiment): Recognition rate (percentage) for each of the training expressions: angry (AN), disgust (DI), fear (FE) happy (HA), sad (SA), surprise (SU), and comparisons to Sun et al. [2010], Alashkar et al. [2016].

calculation against the database, where both can be done efficiently on the GPU.

Expression Synthesis on Raw Data The multilinear autoencoder can also be used to plausibly deform 3D facial scans, *e.g.*for automatic creation of a blendshape rig. We propose to this end the following method. Given an input raw scan in neutral expression, we first register our template in order to obtain identity and expression coefficients of the model. Here we registered in particular using Amberg et al. [2007] and then projected into the model using the iterative approach, initialized with the known neutral expression coefficient. We next recover the target expression coefficients by registering multiple expressions from a different identity. In this case we regularize both with Equation 3.11 and by fitting a unique identity weight to multiple expressions. We then combine the source identity coefficient with the target expression coefficients, thus obtaining deformed templates that perform the required expressions. Finally, we deform the original mesh based on the positions of the deformed template and with Laplacian regularization. Point-to-point correspondences between the template and the scan can be easily obtained thanks to the initial registration. An example of results obtained using this method on a subject from Bosphorus can be found in Figure 3.5, where different action units where transferred to a source mesh (taken from the testing set). For comparison, we show in the top row of Figure 3.5 the ground-truth scans of the source performing the target action units.

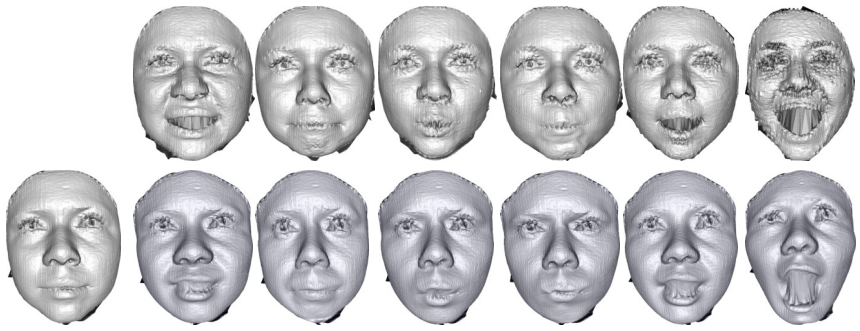


Figure 3.5 – Expression synthesis on raw data. Top: ground-truth scans, bottom: our results.

3.5 Conclusion

In this chapter we demonstrated that it is possible to obtain an expressive multilinear model from large and diverse datasets, by leveraging a novel architecture that we call Multilinear Autoencoder. The proposed approach is capable of making better use of all available data, learning a generative model that can better decouple the latent space, and an encoder that can perform fast regression into this model from raw, unregistered scans. Throughout the experimental evaluation we showed that the Multilinear Autoencoder outperforms current state-of-the-art methods that learn multilinear models from incomplete data, particularly in terms of decoupling the spaces. We believe this work opens up possibilities for learning rich generative 3D face models from large training sets, which in turn can enhance numerous applications including recognition and animation. The next chapter will present one of these possible applications: the registration of large datasets of 3D scans.

The proposed method has a few limitations that are worth mentioning.

A first limitation lies in the need to initialize the decoder with a complete data tensor, requiring at least a subset of the data to be capable of being assembled as such. It is clear that removing this would further simplify the requirements on the training set. Moreover, it would also eliminate the restrictions on the latent space dimensions (see Section 3.3.3), which are bounded by the size of this subset. As mentioned, a random initialization of the decoder did not yield good results in our experiments, yet better initialization approaches can still be explored.

Another disadvantage is related to the choice of dimensionality of the different spaces. Not only this choice is bounded by the size of the initial tensor, but there is also no principled approach for selecting the appropriate model dimensions. Unlike PCA or HOSVD where the percentage of retained vari-

ance can guide model selection, here it must be done by ad-hoc procedures, *e.g.*parameter sweep as in Table 3.3. It is not clear yet how model selection can be performed for this type of deep-learning based models, but a more principled approach would certainly be desirable, as compactness is an important aspect.

A final limitation is the large number of parameters involved in the model, a problem that comes from the use of a multilinear model itself. The amount of entries in the core tensor is typically very large and grows exponentially with each new factor that is added. In our implementation, the number of trainable parameters in the decoder is an order of magnitude larger than those in the convolutional encoder. This also results in large disk space usage for the trained model, on the order of hundreds of megabytes.

Most of these limitations will be addressed by considering a novel strategy for modeling decoupled spaces in Chapter 5.

4 Large-Scale Registration of Faces in Motion

Registration is an essential step in the process of learning a model such as the one presented in the previous chapter. Given two or more scans of a 3D face (not necessarily of the same subject) this technique ensures that anatomically corresponding points are consistently identified, such that the vertex located at *e.g.*the tip of the nose will always be found at that location [1]. Without this step, the different 3D scans have no coherent structure and their common patterns cannot be studied.

We focus here on the registration of *spatiotemporal* data, *i.e.*sequences of 3D face scans, otherwise called 4D data. The interest in the context of this work is two-fold. First, this allows to parameterize not only the individual facial shapes but also their temporal evolution, expanding the scope and performance of automatic facial analysis systems such as expression recognition [Alashkar et al., 2016, Fang et al., 2012, Sandbach et al., 2011], pain detection [Zhang et al., 2015] and realistic expression synthesis [Yu et al., 2012]. Second, motion acquisitions allow to capture a larger range of expressions including spontaneous ones, which would be much harder to elicit in a static capture. This can give access to a rich and diverse source of information for building generative models of the 3D face, and registration is a fundamental first step towards this goal.

When dealing with motion sequences one could in principle apply a purely spatial registration algorithm to each frame, *e.g.* [Amberg et al., 2007, Salazar et al., 2014], yielding a static face pose parametrization that is agnostic to time information. Yet, in the case of faces in motion, the registration can account for the temporal aspects through a spatiotemporal parametrization. The interest here is to better capture face deformations with a temporal tracking, where static registrations provide only coarse and noisy motion information. Spatiotemporal registration is however more complex than its static counterpart, since tracking robustly and reliably is still challenging in practice.

The process of collecting 4D scans is expensive and time-consuming. Yet, as mentioned before, multiple research groups have captured and released large 4D face databases throughout the last decade, *e.g.* Yin et al. [2008], Cosker et al. [2011], Zhang et al. [2014]. We aim here to harness such source of information by registering the large corpus into a single parameterization. This in turn poses additional challenges to the registration process. Not only the amount of data

1. An introduction is provided in Section 2.2.

is considerably larger, but the capture systems and acquisition protocols differ from one dataset to the other. Face registration methods in this context should be able to handle datasets in a fully automatic way, as manual intervention is not feasible for thousands of frames, in addition to being robust to different types of noise resulting from different acquisition scenarios. Furthermore, they must be efficient in order to process thousands of frames in a reasonable time.

The work presented in this chapter addresses the aforementioned objectives by proposing a novel method to register spatiotemporal 3D face data. The approach is based purely on geometry to allow leveraging any available temporal 3D face scan, even when the associated RGB images are not provided *e.g.*for privacy reasons. We do not require pre-determined landmarks as input, which are more challenging to obtain for 3D data, thus removing a possible source of error. The main innovation here is the use of a *spatiotemporal model* as opposed to a purely static one, which combined with a regression-based approach allows to exploit the spatial and temporal coherence of the data in an efficient manner. The use of such model enables registrations that both fix identities over temporal sequences and regularize observed motions to prevent high-frequency flickering. The approach presents the following advantages: it can register multiple datasets into a single representation; it does not require color information as in *e.g.* Cosker et al. [2011], Cheng et al. [2017a], Fyffe et al. [2017], and is robust to occlusions by construction; it runs an order of magnitude faster than recently proposed methods based on parametric face models [Bolkart and Wuhrer, 2015a, Li et al., 2017] while achieving comparable accuracy; and provides compact representations of the results.

The method is evaluated qualitatively and quantitatively on three publicly available datasets, namely D3DFACS [Cosker et al., 2011], BU-4DFE [Yin et al., 2008] and BP4D-Spontaneous [Zhang et al., 2014], demonstrating it can efficiently obtain accurate registrations as well as compact representations. Comparisons to Bolkart and Wuhrer [2015a], Li et al. [2017] and Cosker et al. [2011] show that the proposed approach can achieve similar or better results in terms of vertex-to-scan error and in terms of semantic parametrization, while remaining either more general in terms of requirements of the datasets, or more efficient in terms of computational times.

4.1 Related Work

Numerous works have studied the registration of static 3D face scans, and an overview can be found in Section 2.2. While a static method can be applied independently to each frame of a motion sequence, this is known to lead to artifacts including high-frequency jitter. We focus therefore on methods that take advantage of the temporal redundancy captured by 4D data. A related line of research that has recently received considerable attention is the reconstruction of 4D facial motion based on monocular 2D video, *e.g.* [Cao et al., 2015, Garrido et al., 2016a]. These works solve an underconstrained reconstruction instead of a 3D registration as addressed in this work; the

interested reader is referred to the survey of Zollhöfer et al. [2018]. In the following we discuss strategies for the registration of 4D face data.

Registration of 4D face data

Initial geometry-based methods used a coarse-to-fine approach combined with Free Form Deformations [Wang et al., 2004], or through harmonic [Wang et al., 2008] or conformal [Sun et al., 2010] maps that reduce the problem to 2D registration. For expression recognition, Fang et al. [2012] performed pairwise registration of consecutive frames using an Annotated Face Model (AFM) [Kakadiaris et al., 2007], where temporal information was exploited by initializing with the result of the previous frame.

For real-time expression transfer, Weise et al. [2009] introduced a system based on non-rigid Iterative Closest Point (ICP), from which a person-specific blendshape model was built and used to sequentially track sequences of the same actor in real-time. Follow-up work [Weise et al., 2011] improved on this by using color cues and a probabilistic animation prior which can handle noisier input from an RGB-D camera. The methods of Li et al. [2013], Bouaziz et al. [2013] further removed the need for calibration by updating an initial blendshape model on-the-fly. Other real-time tracking approaches from RGB-D video include Zollhöfer et al. [2014] that deform a template using an as-rigid-as-possible prior, and Thies et al. [2015] that track blendshape weights through an analysis-by-synthesis framework. Further improvements on this line of work included robustness to occlusions and pose [Hsieh et al., 2015], detailed blendshape models through the use of displacement maps [Thomas and Taniguchi, 2016], eye-gaze control [Thies et al., 2018a,b], and full head and upper body tracking [Thies et al., 2018a].

More recent alternatives, used mainly for high-quality acquisition setups, follow two main lines. The first performs registration in texture space by computing correspondences between sparse landmarks predicted using an Active Appearance Model (AAM), which are densified using thin-plate spline deformations [Cosker et al., 2011, Cheng et al., 2017a]. The method of Cosker et al. [2011] achieves inter-sequence correspondence by registering each frame towards a manually selected neutral expression, and intra-sequence correspondence by registering these neutral frames to a template. To better handle texture variations, Cheng et al. [2017a] extend the previous by using session-and-subject specific AAM, and non-rigid ICP [Amberg et al., 2007] between manually selected neutral frames. Since these methods operate on color information, they require careful acquisition setups with controlled lighting conditions, as *e.g.*moving shadows can lead to inaccuracies.

The second line of work takes advantage of low dimensional parametric shape spaces learned from large databases of static 3D face scans and used as prior during registration. Most related to our work, multilinear models of identity and expression [Bolkart and Wuhrer, 2015a] and a linear articulated model with expressions [Li et al., 2017] have been used for this purpose. These

works achieve registrations of relatively high accuracy and report running times of 30 seconds to 2 minutes per frame. Although the overall facial shape is recovered, fine-scale details such as wrinkles are not modeled. Our work shares this property, as well as the robustness and accuracy of these methods while allowing for a gain in efficiency.

Joint registration and reconstruction

Performance capture is concerned with the recovery of both the 3D shape and its temporally coherent deformations, and it is hence also related to this work. In this context, several authors use optical flow to recover a consistent geometry from passive [Bradley et al., 2010, Beeler et al., 2011] or active [Zhang et al., 2004] systems using synchronized multi-view 2D videos. Bradley et al. [2010] jointly solve for registration and reconstruction by sequentially tracking the initial frame. Optical flow with sequential tracking is known to be prone to drift (*i.e.*the accumulation of tracking errors), and thus Beeler et al. [2011] propose instead to do optical flow on sub-sequences defined by automatically selected key-frames. Results are of very high quality and achieve pore-level details. Non-sequential tracking has also been explored, by using a minimum spanning tree [Klaudiny and Hilton, 2012], a performance flow graph [Fyffe et al., 2014], or by independent optical flow between a template and each frame [Fyffe et al., 2017]. All of these methods require a dense setup of synchronized video cameras. Valgaerts et al. [2012] simplify these requirements by introducing a method that achieves results of similar quality from a single pair of stereo cameras, combining sequential scene flow to compute the global registration with shading-based refinement to compute fine-scale details. More recently, Wu et al. [2018] proposed an incremental approach in which a person-specific neural network is used for initialization and gradually improved as more frames are registered. These methods achieve temporally coherent results which are of high quality and include fine-scale details. However, they are limited to specific acquisition setups as the input to the methods are synchronized and calibrated 2D videos. In this work, we consider the more general problem of registering the geometry of 4D face scans without the need for reliable color information.

4.2 Method

We aim here at registering a large number of sequences of 3D face scans, displaying a varied range of identities and emotions. Each of these sequences may contain many frames, and each frame a large number of vertices, making the problem high-dimensional and difficult to optimize. Furthermore, datasets captured with different acquisition setups present different levels of noise, missing data and occlusions, hence the naive application of a frame-by-frame template fitting approach is prone to failure. To keep the method as general as possible we do not assume availability of either landmark or color information,

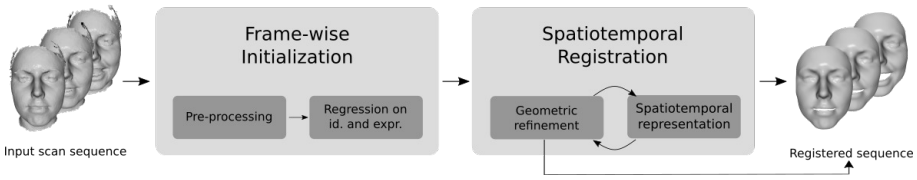


Figure 4.1 – Overview of the proposed spatiotemporal registration approach. The input to the system is a sequence of 3D scans showing a 3D face in motion, and the output is a registered version of the sequence encoded in a compact representation. The registration is initialized by a frame-by-frame automatic regression approach that gives identity and expression coefficients of a multilinear model. The identity coefficients are then combined in order to build a spatiotemporal model of the sequence, and an iterative process alternates between projecting into this model and refining the geometry.

which allows to register data coming from sources for which privacy is a concern.

In order to process large datasets we seek a strategy that is centered around robustness and efficiency. To this end, we follow a model-based approach as was previously considered by *e.g.* Amberg et al. [2008], Schneider and Eisert [2009], Cheng et al. [2017b] in the static case, or Bolkart and Wuhrer [2015a], Li et al. [2017] in the temporal case. But unlike these, we propose instead to use a **spatiotemporal model**, combining a shape space that regularizes the spatial information with a temporal space that regularizes the trajectories of each vertex, thus capitalizing on the redundancies present both in space and time. This allows not only for more accurate registrations, but also faster computational times and a very compact representation of the output.

A second step towards robustness and efficiency is taken by the use of a **regression-based initialization**. This builds on the work presented in the previous chapter by leveraging the multilinear encoder for a fast initialization of the spatial component of the model. The regression is done independently on each frame and outputs for each time step the identity and expression coefficients of a static multilinear model. Per-frame identity coefficients are then combined into a single identity weight and used to build the shape basis of the spatiotemporal model. The algorithm proceeds by alternating between fitting the current estimation to the possibly noisy correspondences, and projecting the result back into the spatiotemporal model. Figure 4.1 summarizes the process.

We will begin by describing the spatiotemporal model in Section 4.2.1, while the details of the algorithm are presented in Section 4.2.2.

4.2.1 Spatiotemporal Model

The model used in this approach is an extension of the multilinear model introduced in Section 2.3.3 and Chapter 3. Recall that, given $\mathbf{x} \in \mathbb{R}^{3n}$ a vector of coordinates associated with the n vertices of a registered mesh, $\mathcal{M} \in \mathbb{R}^{3n \times d_{id} \times d_{exp}}$ a core tensor, and $\mathbf{w}_{id} \in \mathbb{R}^{d_{id}}$, $\mathbf{w}_{exp} \in \mathbb{R}^{d_{exp}}$ the identity and expression coefficients respectively, the multilinear model relates these to the 3D face by:

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathcal{M} \times_2 \mathbf{w}_{id} \times_3 \mathbf{w}_{exp}, \tag{4.1}$$

where $\bar{\mathbf{x}}$ is the mean face over the model's training data, and $\times_i$ denotes mode-i multiplication.

When the data is a *sequence* of F 3D faces in correspondence, $[\mathbf{x}^1, \ldots, \mathbf{x}^F]$, $\mathbf{x}^i \in \mathbb{R}^{3n}$, one could encode it within the multilinear model by using a unique identity representation $\mathbf{w}_{id}$ plus an array of expression weights $[\mathbf{w}^1_{exp}, \ldots, \mathbf{w}^F_{exp}]$. This is the approach followed for example by Bolkart and Wuhrer [2015a], and while it gives a relatively compact representation, it has a few drawbacks. First, unless a prior is imposed on the curve, the expression weights can take any form which in practice results in flickering of the reconstructed vertices. Second, the formulation does not take into account the high temporal regularity that each vertex exhibits, resulting in redundancy of the representation.

As originally proposed by Akhter et al. [2012], a bilinear model can be built that leverages both spatial and temporal redundancies. To this end, the sequence is organized into a matrix $\mathbf{S} \in \mathbb{R}^{F \times 3n}$ containing each frame in a row, $\mathbf{S} = [\mathbf{x}^1, \ldots, \mathbf{x}^F]^T$, or equivalently each vertex trajectory in a column. Let $\mathbf{B} \in \mathbb{R}^{3n \times d_s}$ be a matrix with the shape basis vectors in its columns, encoding each frame into a space of dimension d_s. Similarly, let $\mathbf{\Theta} \in \mathbb{R}^{F \times d_t}$ be a matrix with the temporal basis vectors in its columns, encoding the *trajectory* of each vertex into a space of dimension d_t. Then the sequence matrix $\mathbf{S}$ can be decomposed as

$$\mathbf{S} \approx \mathbf{\Theta} \mathbf{C} \mathbf{B}^T, \tag{4.2}$$

where $\mathbf{C} \in \mathbb{R}^{d_t \times d_s}$ is a matrix of spatiotemporal coefficients that compactly encode $\mathbf{S}$. The dimensions d_s and d_t allow to trade off the compactness of the representation and the approximation error of the input sequence.

We can easily incorporate this model into the multilinear framework. Given the unique identity coefficients $\mathbf{w}_{id}$, multiplying it with the core tensor results in a shape matrix for that particular subject:

$$\begin{aligned} \mathbf{x} &\approx \bar{\mathbf{x}} + (\mathcal{M} \times_2 \mathbf{w}_{id}) \times_3 \mathbf{w}_{exp} && (4.3) \\ &= \bar{\mathbf{x}} + \mathbf{w}^T_{exp} (\mathcal{M} \times_2 \mathbf{w}_{id})_{(3)} && (4.4) \\ &= \bar{\mathbf{x}} + \mathbf{w}^T_{exp} \mathbf{B}^T && (4.5) \end{aligned}$$

where $\mathcal{X}_{(3)}$ denotes mode-3 matricization of a tensor $\mathcal{X}$ (see Kolda and Bader [2009]). We can thus obtain the shape matrix $\mathbf{B}$ of the spatiotemporal model by simply multiplying $\mathcal{M} \times_2 \mathbf{w}_{id}$, assuming we know $\mathbf{w}_{id}$.

For the temporal basis $\mathbf{\Theta}$ we follow Akhter et al. [2012] and fix it to the Discrete Cosine Transform (DCT), since this approaches the optimal PCA-learned basis when the data is generated from a stationary first-order Markov process, and was empirically demonstrated to hold for sparse facial data in Akhter et al. [2012]. Note that the dimensions d_t of the temporal basis need to be chosen carefully, as a very low-dimensional space will not allow high-frequency trajectories and will flatten the original motion. In this work we set d_t to be a factor of the sequence length (*i.e.*$d_t = F/k$ for some constant k), which allows a certain degree of independence from the sampling rate if we assume that the motions are of approximately the same speed. This approach worked well for the experiments shown in this chapter where this property holds, but a more accurate method for model selection should be investigated. We leave this as future work.

4.2.2 Registration

As depicted in Figure 4.1, we proceed in two major steps. First, we perform an initialization independently on each frame that robustly and efficiently regresses each scan against identity and expression coefficients of the multilinear model. The resulting meshes correctly capture the general structure of the motion and shape but are still overly smooth and, because each frame is treated independently, exhibit high-frequency jitter. To remedy this we use the multilinear face model to build the spatiotemporal model described in Section 4.2.1. The second step makes use of this model to iteratively improve the initial approximations, regularizing the motion of the vertices and turning the problem into a much lower-dimensional one compared to a frame-wise formulation. We now describe each of these steps in more detail.

Frame-wise Initialization

Given a sequence of observations $[\mathbf{o}^1, \ldots, \mathbf{o}^F]$ consisting of F frames, we register each frame independently by regressing identity and expression coefficients using the encoder part of the multilinear autoencoder presented in Chapter 3. We thus obtain a sequence of identity and expression weights that represent the face motion:

$$W_{id} = [\mathbf{w}_{id}^1, \ldots, \mathbf{w}_{id}^F], \text{and} \tag{4.6}$$

$$W_{exp} = [\mathbf{w}_{exp}^1, \ldots, \mathbf{w}_{exp}^F]. \tag{4.7}$$

Even though the network was trained with raw scans presenting different types of noise, the quality of the registration degrades when the input data differs in form and orientation from the original training data. To ensure results of high quality, the input 3D scan is therefore pre-processed as follows. We first detect the nose tip by training a neural network for this task on depth data, using the same architecture and training data as in the previous chapter

(Section 3.4.2) [2]. We next crop the face a radius of 100mm around this point, and perform a coarse frontalization step so that it approximately looks towards the z-direction. For frontalization, we consider the direction of the normals of each vertex, which gives a distribution of orientations over the sphere that sample a semi-sphere for a cropped face. The directional mean of this distribution gives a coarse approximation of where the face is "looking at", and we align this to the directional mean of the model's mean face. This makes the weak assumption that the face is not upside-down and works well as long as the cropped face does not contain too many holes or extra parts. Note that only a coarse alignment is required here since the autoencoder was trained to have some degree of robustness to pose variation in the input image. As a result, this pre-processing step can be replaced with any other method that will produce a cropped face and a rough frontalization.

The resulting multilinear model representations $[\mathbf{x}^1, \ldots, \mathbf{x}^F]$, obtained by reconstructing the faces using Equation 4.1 with the coefficients from Equations 4.6 and 4.7, are in the coordinate system in which the multilinear model was learned. For further refinement of these approximations they need to be compared to the original scans. To align the observations $[\mathbf{o}^1, \ldots, \mathbf{o}^F]$ to the model coordinate system we take advantage of the depth images generated during regression to find initial correspondences. In particular, we consider the depth image of the cropped and frontalized scan, and the depth image of the registered mesh, and establish preliminary correspondences by assigning pixels at the same location. This correspondence is used to rigidly transform $\mathbf{o}^i$ to $\mathbf{x}^i$ $(i = 1, \ldots, F)$. We then perform a few iterations of regular ICP alignment [3]. Once each frame is aligned, we discard the cropped version and go back to the original raw scan; this allows to remove the quality of the crop as possible source of error in the subsequent steps.

Given the identity coefficients of Equation 4.6 we next proceed to build the model outlined in Section 4.2.1. Unlike the original formulation of Akhter et al. [2012] in which both the shape basis $\mathbf{B}$ and the model coefficients $\mathbf{C}$ need to optimized, we leverage the multilinear model to obtain a person-specific spatiotemporal representation. Specifically, we summarize the regressed identity coefficients W_{id} into a unique coefficient $\mathbf{w}_{id}$ for the entire sequence, given that the identity of the subject is fixed for any given motion. We compute $\mathbf{w}_{id}$ as a mean over the regressed results, $\mathbf{w}_{id} = \overline{W_{id}}$, and create the shape basis $\mathbf{B}$ by multiplying the core tensor with $\mathbf{w}_{id}$ as $\mathbf{B}^T = \mathcal{M} \times_2 \mathbf{w}_{id}$. The temporal basis $\mathbf{\Theta}$ is fixed to the DCT basis according to the specified dimensions d_t.

2. We use the `bu-bosph` version which has ground-truth nose tip locations. The input is the same heightmap used for regressing multilinear coefficients. Note that nose tip detection on a heightmap image is a relatively easy task.

3. We accelerate the nearest point search using the libigl implementation of the axis-aligned bounding box (AABB) data structure [Jacobson et al., 2018]. We stop when the difference between iterations is small, or a maximum of 30 iterations is reached.

Iterative Refinement

Up to this point we have computed, for each frame i, spatially aligned observations $\mathbf{o}^i$ along with registered faces $\mathbf{x}^i$ that approximate the geometry of $\mathbf{o}^i$. By iterating between refining the geometry of $\mathbf{x}^i$ to match $\mathbf{o}^i$, and projecting into the spatiotemporal model, we further improve the geometric approximation of the registrations, obtaining temporally smooth results that can be represented compactly.

Geometric refinement To improve the quality of the approximation $\mathbf{x}^i$ of $\mathbf{o}^i$ we non-rigidly deform the registrations to the scans. The following discussion omits the frame index i to simplify notation. The registration $\mathbf{x}$ is warped to $\mathbf{o}$ by optimizing for displacements $\delta^{\mathbf{x}}$ of the vertices of $\mathbf{x}$ along their normal directions with Laplacian regularization. In particular, we solve for

$$\min_{\{\delta_1^{\mathbf{x}},\dots,\delta_n^{\mathbf{x}}\}} \alpha \sum_{j=1}^{n} w_j \|\mathbf{v}_j^{\mathbf{x}} + \delta_j^{\mathbf{x}} \mathbf{n}_j^{\mathbf{x}} - \mathbf{p}_j^{\mathbf{x}}\|^2 + \beta \sum_{j=1}^{n} \|\mathcal{L}(\delta_j^{\mathbf{x}})\|^2, \tag{4.8}$$

where $\mathbf{v}_j^{\mathbf{x}} \in \mathbb{R}^3$ is a vertex in $\mathbf{x}$, $\mathbf{p}_j^{\mathbf{x}} \in \mathbb{R}^3$ the closest point to $\mathbf{v}_j^{\mathbf{x}}$ in $\mathbf{o}$, $\mathbf{n}_j^{\mathbf{x}} \in \mathbb{R}^3$ the normal vector of $\mathbf{v}_j^{\mathbf{x}}$, $\mathcal{L}$ the cotangent discretization of the Laplace-Beltrami operator [Meyer et al., 2003], and w_j, α, β are scalar weights. We discard closest points whose Euclidean distance is greater than 5mm and whose deviation in normal vector is greater than 45° by setting w_j to zero. This formulation can be efficiently minimized by solving a linear system of equations.

Spatiotemporal sequence projection For each iteration, we gather the approximations $\mathbf{x}^i$ obtained in the previous step into a sequence matrix $\mathbf{S}$, and use Equation 4.2 to compute $\mathbf{C}$ by solving

$$\mathbf{B}\mathbf{C}^T = \mathbf{S}^T\mathbf{\Theta}. \tag{4.9}$$

This can be performed efficiently, as $\mathbf{B}$ is fixed and can be factorized once for all the iterations.

Final refinement The two previous steps are iterated a few times until convergence. To obtain more detailed results we complete the iterative process with a final geometric refinement step. This allows to leave the bounds of the multilinear model, thereby providing more accurate approximations of $\mathbf{o}^i$. To prevent artifacts we use a stronger regularization weight β in the last geometric refinement. However, this loses both the compactness of the representation and the motion regularization. To rectify this we project the trajectory of the displacements $\delta_i^{\mathbf{x}}$ into a second DCT basis (of possibly different dimensionality d_t'), obtaining a displacement coefficient vector $\mathbf{d}_i \in \mathbb{R}^{d_t'}$ for each of the n vertices, with $d_t' << F$. We thus retain compactness while allowing for more detailed registrations, as well as preventing flickering in the final trajectory of the vertices.

Sequence Representation

After registration we can compactly store the results given the multilinear model. For each registered motion the following information suffices to reconstruct the sequence $[\mathbf{x}^1, \ldots, \mathbf{x}^F]$: (1) the identity coefficient $\mathbf{w}_{id} \in \mathbb{R}^{d_{id}}$ that compactly encodes the shape matrix $\mathbf{B}$; (2) the spatiotemporal coefficients $\mathbf{C} \in \mathbb{R}^{d_t \times d_s}$; (3) the dimensions of the temporal bases d_t and d_t'; and (4) the m displacement coefficients $\{\mathbf{d}_1, \ldots, \mathbf{d}_m\}, \mathbf{d}_i \in \mathbb{R}^{d_t'}$. This significantly reduces the storage requirements of large datasets (*e.g.*from 9.1GB to less than 1MB in the example from Figure 4.5), while still retaining a reasonable level of detail.

4.3 Evaluation

We validate the approach on D3DFACS [Cosker et al., 2011], BU-4DFE [Yin et al., 2008] and BP4D-Spontaneous [Zhang et al., 2014] datasets. D3DFACS contains 519 sequences of 10 subjects performing different types of facial action units, while BU-4DFE contains 101 subjects with 6 sequences each performing the six prototypical emotions; in both cases the average sequence length is around 100 frames and the meshes contain around 30K and 35K vertices respectively. BP4D-Spontaneous contains 328 sequences with 41 subjects performing 8 tasks each, which were designed to elicit spontaneous emotions. The average sequence length is around 1100 frames and the average number of vertices is around 37K. In the following we provide both qualitative and quantitative evaluations over these.

Implementation details The code was implemented in C++ using Eigen3 [Guennebaud et al., 2010] and libigl [Jacobson et al., 2018]. We use the autoencoder from Chapter 3 that was trained on BU-3DFE [Yin et al., 2006] and Bosphorus [Savran et al., 2008] datasets for 500 epochs [4]. The dimensions of identity and expression spaces are set to 65 and 20, and the dimension of the temporal basis is set to $F/5$ (where F is the number of frames). We set $d_t' = 5$, and unless otherwise specified, we fix the number of iterations to 5. For Equation 4.8 we set $\beta = 1$, $\alpha = 0.9$ during iterations and $\alpha = 0.8$ for the final step. Since BU-4DFE contains noisier scans, we set $\alpha = 0.5$ during iterations and $\alpha = 0.2$ for the final step to avoid overfitting. The template mesh has 5996 vertices and is depicted in Figure 4.4.

4.3.1 Qualitative results

Figure 4.3 shows an example of the results obtained on each of the steps of the method: regression, spatiotemporal registration, and final refinement. Figure 4.2 shows a few more examples of registrations obtained on D3DFACS, BU-4DFE and BP4D-Spontaneous. They illustrate that accurate cross-dataset registrations can be obtained, while still being robust to different types of noise in the data.

4. We use in fact the version that was originally published in Fernández Abrevaya et al. [2018].

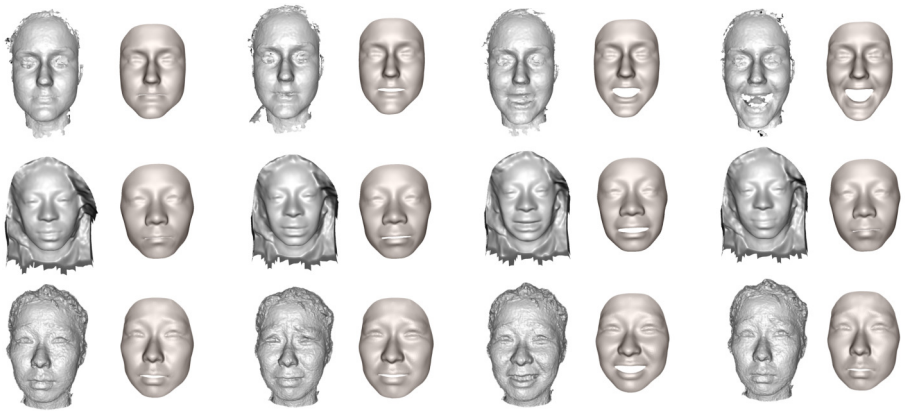


Figure 4.2 – Registration examples on (from top to bottom): D3DFACS, BU-4DFE and BP4D datasets.

4.3.2 Quantitative results

We evaluate the quality of the registrations with two commonly used metrics: median per-vertex error towards the input scan, and landmark distances. The median per-vertex error is taken across all registered frames in the dataset, and shows how close the registrations are to the real scans. We also evaluate semantic accuracy by manually placing landmarks on five key-frames over 10 randomly selected sequences of D3DFACS, and measuring the Euclidean distances between these and the landmarks defined over the template. We use in particular 11 landmarks, which can be visualized in Fig 4.4. The chosen key-frames sample the sequence by taking the first and last frame, the peak frame, and two intermediate ones.

We evaluate the stability of the motion by using a *compactness* measure (see Davies et al. [2008]) as follows. For each sequence, we align the frames using generalized Procrustes analysis, perform PCA, and measure the amount of variabilty captured by each principal component. If the registrations exhibit high-frequency jitter, we expect to see less variability retained by the first principal components, as the variations coming from flickering vertices would have to be encoded by higher-order principal components. To summarize

Figure 4.3 – Results for the successive steps (left to right): raw scan, regression, 1 iteration, 5 iterations, final.

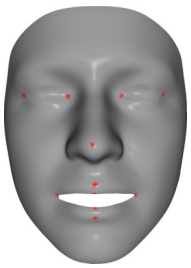


Figure 4.4 – Template mesh, and landmarks used for evaluation.

this over the entire dataset we determine the mean variability obtained as a function of the percentage of principal components considered.

Figure 4.6 shows a cumulative plot of the median per-vertex error on D3DFACS obtained after the initial regression, after 1, 3, 5 and 10 iterations of spatiotemporal registration, and for the final result. We can see that the iterative process improves the initial regressions in terms of surface fit. Furthermore, Table 4.1 shows the mean landmark error over the 11 landmarks for the final results obtained after 1, 3, 5 and 10 iterations. Despite being a landmark-free registration method, the method obtains a good semantic accuracy that is improved with each iteration.

We evaluate the benefits of the temporal regularization by comparing the full model with a static version of our framework. For this, instead of projecting onto the spatiotemporal model we independently project each frame onto the shape basis $\mathbf{B}$, and measure the results in terms of vertex error and compactness. Figure 4.7 shows cumulative plots obtained for these registrations. Note that while using a spatiotemporal model achieves similar accuracies in terms of vertex error, the compactness of each sequence improves with the spatiotemporal model, implying less high-frequency jitter with the latter. This results can also be qualitatively assessed in the accompanying video.

Finally, we show the ability of the method to track long videos by registering the sequences from BP4D-Sponteanous, many of which consist of more than 1000 frames. We obtain a mean vertex error of 0.33mm over all registered sequences and frames. Figure 4.5 further shows the median per-vertex error for each frame on one example. This error stays between 0.1 and 0.4mm and does not increase with time, suggesting that no drift is occuring. This is expected, as the regression step is performed independently on each frame.

With respect to the running times, we report a mean per-frame processing time of 578ms on the D3DFACS dataset, 637ms on BU-4DFE and 399ms for BP4D, for five iterations in all cases. Computation times were measured on an Intel Xeon 3.30GHz with NVidia GeForce GTX 1080 GPU.

4.3.3 Comparisons

We compare our method to the previous works of Bolkart and Wuhrer [2015a], Li et al. [2017] and Cosker et al. [2011] using registrations provided by

	It. 1	It. 3	It. 5	It. 10	Li et al. [2017]
Mean error	2.92	2.77	2.69	**2.65**	3.13

Table 4.1 – Mean landmark error in mm., for 1, 3, 5 and 10 iterations, and comparison with Li et al. [2017], over 10 selected sequences of the D3DFACS dataset.

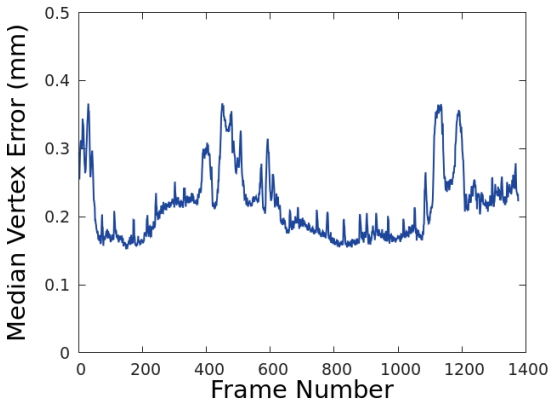


Figure 4.5 – Median per-vertex error for each frame of a long sequence in BP4D.

the authors.

Bolkart and Wuhrer Bolkart and Wuhrer [2015a] also register motion sequences in a fully-automatic manner by using a multilinear model and geometric information only. We compare to this method on 497 sequences from

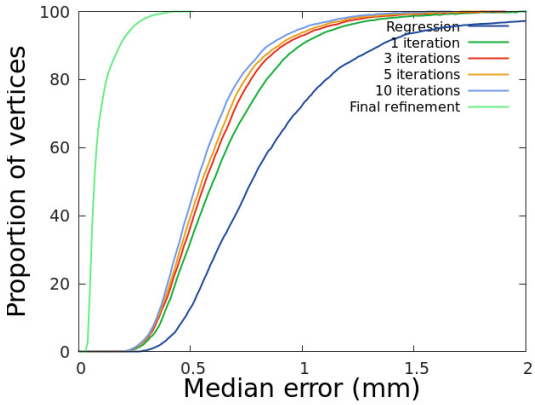
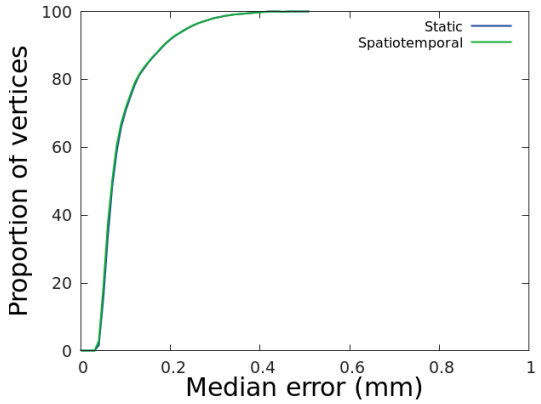
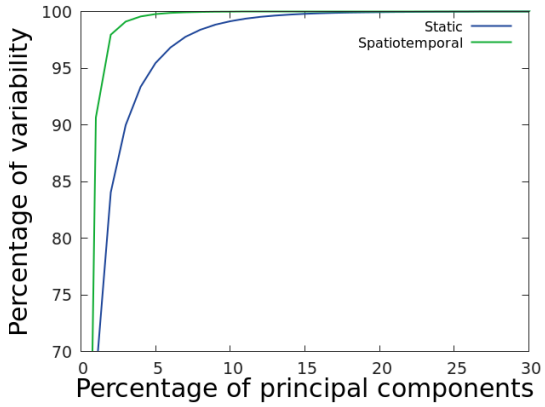


Figure 4.6 – Cumulative plot of median per-vertex error over D3DFACS (46028 frames) for: regression results, spatiotemporal registration (1, 3, 5 and 10 iterations) and final refinement.



(a) Median per-vertex error



(b) Mean compactness

Figure 4.7 – Comparison between our method and a static version, in terms of vertex error and sequence compactness.

BU-4DFE, which are the sequences that were correctly registered by Bolkart and Wuhrer [2015a]. Figure 4.8a shows cumulative plots of the median per-vertex error on all the registered sequences in BU-4DFE, comparing Bolkart and Wuhrer [2015a] to our registration without and with the final refinement step, since Bolkart and Wuhrer [2015a] has no refinement step. Figure 4.9a further shows a qualitative comparison over a challenging example. Results reveal similar accuracy for both methods without the refinement step, whereas Bolkart and Wuhrer [2015a] requires around 30 seconds per-frame to process.

Li et al. The method of Li et al. [2017] was used to register D3DFACS, and thus we compare our results over this dataset. For a fair comparison we crop their full-head model so that it contains only the face, to be similar to our registrations. We obtain a mean vertex error of 0.13 mm for our method, and 0.33mm for Li et al. [2017]. In Figure 4.8b we show the cumulative plots for the median per-vertex distance for both methods. They demonstrate that our

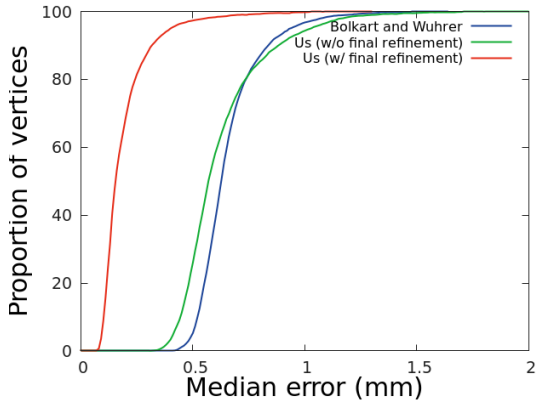
approach achieves higher accuracy while reducing both the running time (Li et al. [2017] reported 155 seconds per-frame) and the requirements on the dataset. We also compare with respect to landmark errors; results can be found in Table 4.1. Our approach achieves better semantic accuracy with a single iteration, even though it requires no pre-determined landmarks to guide the process, confirming that our registrations faithfully preserve the anatomic semantics. Figure 4.9b shows a qualitative comparison.

Cosker et al. Finally, we compare to the work of Cosker et al. [2011]. Comparisons are done over 3 sequences of D3DFACS that were provided by the authors. We obtain a mean error of 0.13mm for our method and 0.18 mm for Cosker et al. [2011]. Figure 4.8c shows the results in terms of median per-vertex error. They demonstrate similar accuracy although our method is more general as it does not require a controlled capture setup. Figure 4.9c shows a qualitative comparison.

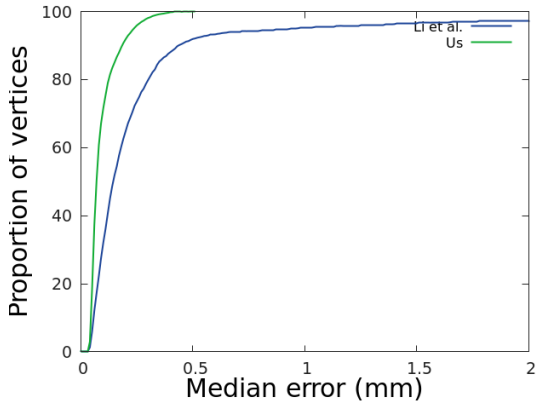
Comparisons on efficiency The efficiency of our method comes from both the regression step and the spatiotemporal model optimization. The regression is essential to get a good starting point that is already close to a local minimum, and it can be performed efficiently on the GPU thanks to the heightmap representation. Furthermore, due to the spatiotemporal model we need to optimize for much less parameters, while still remaining in a global sequence formulation. In particular, on each iteration we need to solve for the matrix $\mathbf{C}$ which is of size $d_s \times d_t$, with $d_s = d_{exp}$. In our implementation $d_t = F/5$ and thus we optimize for $d_{exp}F/5$ parameters, reducing by a factor of 5 compared to a frame-by-frame formulation. The method of Bolkart and Wuhrer [2015a] optimizes for the parameters of each frame, which amounts to $d_{id} + Fd_{exp}$ variables to be solved. The main data term on the method of Li et al. [2017] optimizes for shape and expression parameters plus per-joint pose parameters of an articulated model on a frame-by-frame basis, increasing the complexity. Moreover, each frame is initialized from the previous one and thus it cannot be parallelized. As for UV-based methods such as Cosker et al. [2011], the computational complexity depends on the number of pixels of the image; while these are usually more efficient than 3D-based ones, we have shown that we can achieve similar accuracy, while remaining more general with respect to the acquisition setup.

4.3.4 Limitations

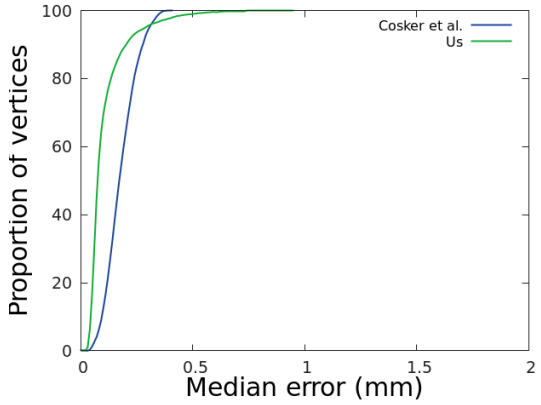
The regression-based initialization allows the method to be robust to noise in the input data, but it comes with drawbacks. In particular, the use of a depth map implies that the method is not rotation-invariant, and thus a proper pre-processing is needed to ensure that the face is "looking front". Although this does not require accurate pose detection (the network was trained with data showing $\pm 30^\circ$ of pose variation), the output will be more accurate the



(a) Bolkart and Wuhrer [2015a] on 49757 frames

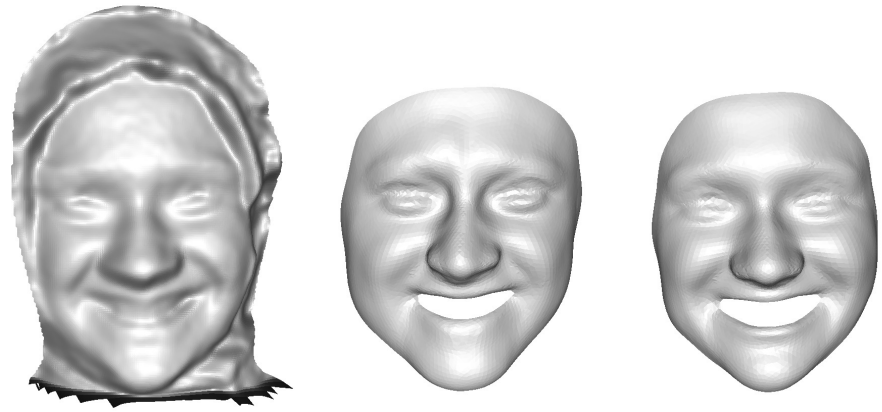


(b) Li et al. [2017] on 46028 frames

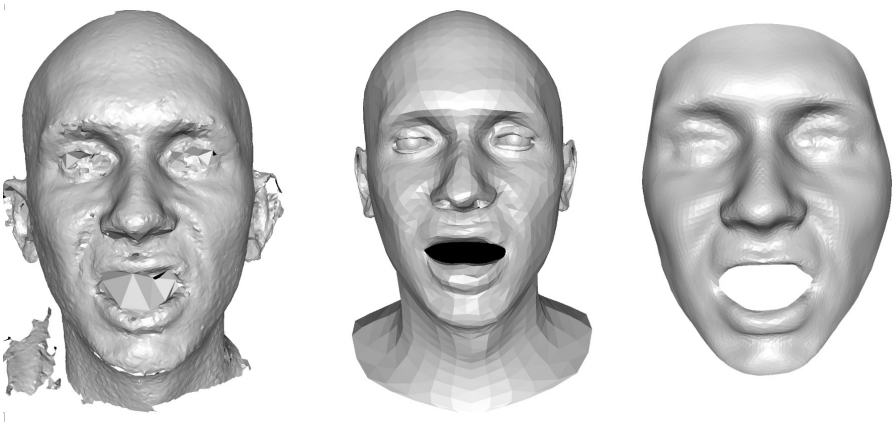


(c) Cosker et al. [2011] on 292 frames

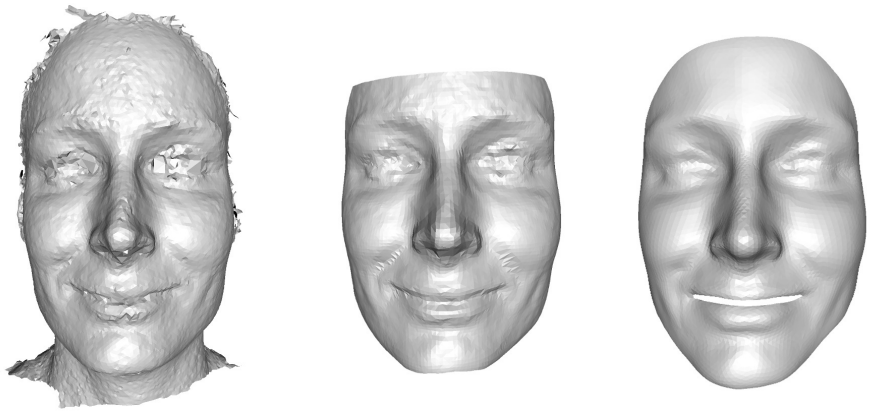
Figure 4.8 – Comparisons to Bolkart and Wuhrer [2015a] on BU-4DFE, Li et al. [2017] on D3DFACS, and Cosker et al. [2011] on a subset of D3DFACS. Cumulative plots for median per-vertex error.



(a) Bolkart and Wuhrer [2015a]

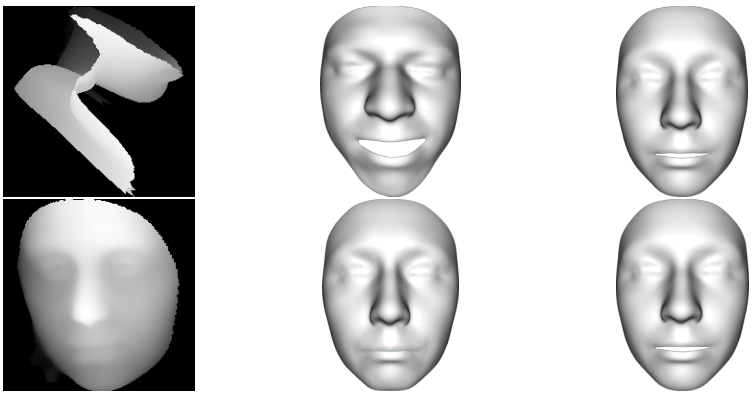


(b) Li et al. [2017]



(c) Cosker et al. [2011]

Figure 4.9 – Qualitative comparisons. From left to right: original scan, compared method, our result.



(a) (b) (c)

Figure 4.10 – Failure example. (a) Heightmap obtained after bad nose tip detection (top) and the following frame (bottom); (b) Regression results; (c) Recovery by interpolation.

closer the input is to a frontal pose, and this in turn affects the final result. Furthermore, our choice of initialization can sometimes be a source of failure, particularly with the nose tip detection; see *e.g.*Figure 4.10. When this step fails all subsequent steps fail too, since regressions are inaccurate and initial correspondences cannot be found. In our experiments, this resulted in erroneous registration of some of the frames in BU-4DFE and BP4D datasets. On the other hand, since we are dealing with motion data, unsuccessful frames that are isolated can be ignored without resulting in failure of the entire sequence. In our implementation we automatically detect failed frames after ICP diverges, and this is fixed by interpolating pose and shape parameters using correct neighbouring frames. With this simple approach all sequences from BU-4DFE and 95% from BP4D were registered (no errors were found during registration of D3DFACS).

Another intrinsic limitation comes from the restricted scope of our trained model. Particularly for expressions that are far from this scope, the framework will provide only a coarse approximation and even the final refinement step can fail to compensate. A related problem, already mentioned in Section 4.3.4, is the simplistic approach for temporal model selection, in which the dimensions are only dependent on the number of frames. Even though this worked well for most of the sequences registered here, a few of the sequences from BP4D include speech, which occurs faster than pre-defined expressions. In our experiments the chosen temporal resolution is not sufficient for this type of motion, and some of the visemes get smoothed out.

4.4 Conclusion

We described in this chapter a novel method for the automatic registration of large datasets of 3D face scans in motion. Having access to a fully automatic, efficient and accurate registration approach not only enables the study of both spatial and temporal patterns of the face, but also allows to do so at a larger scale. This in turn allows to benefit from the efforts of multiple researchers that captured and made publicly available different aspects of the 3D face in motion. The technique proposed in this chapter holds several properties that are appealing in this context: it is fully automatic, robust to different noise characteristics, has minimum requirements on the input scans and as such it is not limited to a specific capture system, it is efficient, and yields compact representations of the data. We successfully registered in Section 4.3 three standard datasets, without losing accuracy and with significantly better time performances than competing methods. The approach shows how the use of a global *spatiotemporal* model -as opposed to a purely static shape model, or a sequential motion prior- can benefit the task of 4D registration. It also indicates how a regression-based approach can help to achieve a robust initialization, despite its limited accuracy against more classic methods.

There are several aspects that should be considered in a future work. First, as mentioned in Section 4.2.1, the approach we used for temporal model selection is quite simplistic and a more accurate strategy, capable of handling motions of variable speed, would be more valuable. In particular, we observed that speech-related motions –which are performed at higher speed than pre-defined expressions or action units– can be easily "washed out" due to the low dimensional temporal space. Setting higher dimensions for the temporal space on the other hand results in high-frequency jitter, and thus a better approach –*e.g.*an automatic selection of the best temporal dimension, or a semi-local model covering a certain time window– should be explored. An alternative that is also worth exploring is to consider a different type of temporal model, *e.g.*a data-driven one.

Another interesting extension would be to consider a refinement not only of the expression and geometry factors, but also the identity which is used to build the shape basis $\mathbf{B}$. Recall that this is constructed purely from the regression results –a bad identity initialization could result in poor registrations. Although we did not observe such problems, it is clear that simultaneously refining the shape would further benefit the accuracy of the technique.

A final extension should devote particular attention to the most expressive parts of the face: the mouth and the eyes. These areas have very distinct motion, geometry and appearance, and are key for transmitting emotions. Our current method is not capable of distinguishing closed from open eyes, or capturing subtle eyelid motion, and as previously mentioned the temporal dimensions are sometimes insufficient to correctly capture speech. A local treatment of these (as considered in *e.g.* Garrido et al. [2016b], Bermano et al. [2015]) should be considered for more accurate results.

5 A Decoupled 3D Facial Shape Model by Adversarial Learning

The previous chapter provided access to a large dataset of registered 3D faces. With this at hand, we revisit the problem of building decoupled models that was addressed in Chapter 3, this time in light of more advanced modeling techniques.

We are interested here in building generative models that can capture the space of realistic three-dimensional faces, while also differentiating the various factors that influence the generation of this shape, *e.g.*the individual identity or the expression. As mentioned, these *decoupled* models offer an independent parameterization to each of the sources of variation while at the same time modeling the interactions that occur among them. While the interest is typically to disentangle identity from expression, other factors may come to play too. For instance, a model that can disentangle shape, expression and viseme [1] can have applications in visual 3D speech synthesis, enabling systems that generalize to multiple identities and can control the emotion in which a viseme is performed.

The identity and expression subpaces are typically modeled as two independent linear factors which are additively combined [Amberg et al., 2008]. While simple and effective for inference applications, these models can produce artifacts when transferring expressions among very different facial shapes. Another commonly used alternative for decoupling the latent space is the *multilinear model*, and we have seen in Chapter 3 how it can be learned from a large dataset of 3D facial scans. Yet, there are several challenges that remain.

First, while scalability in terms of size of the training set was already addressed in this thesis, multilinear models are still not scalable in terms of *number of factors*. The size of the core tensor grows exponentially with each new dimension, a property that is shared by the Multilinear Autoencoder of Chapter 3. Furthermore, acquiring the data becomes harder with each new factor: a model that decouples for example identity, expression and viseme would require the capture of multiple subjects each performing all the expressions in all of the visemes; a high cost both in time and money. Multilinear Autoencoders also suffer from this, as they still require a subset of the data to be assembled as a tensor for initialization.

A second downside is the assumption of linearity. During motion the face undergoes complex deformations that experts believe cannot be correctly

1. The visual counterpart of a phoneme (a unit of sound).

captured by a restricted linear space [Pighin and Lewis, 2006, Cosker et al., 2010, Trutoiu et al., 2014]. To relax the linear assumption in modeling 3D faces, deep generative models with autoencoder architectures have recently been proposed. They demonstrate benefits in modeling geometric details [Bagautdinov et al., 2018, Tran et al., 2019], joint appearance and shape [Lombardi et al., 2018, Zhou et al., 2019] and non-linear deformations present in extreme facial expressions [Ranjan et al., 2018]. Yet, none of these approaches are capable of decoupling the factors of variation. [2]

A final drawback, this time more specific to the technique presented in Chapter 3, relates to the loss function that was used to enforce decoupling while training the network. The loss in Equation 3.7 induces an error solely based on the structure of the latent space. Yet, we believe that the *perception* of the output should also be taken into acount if we want to model subtler changes. For example, given two meshes with a same expression code, they should be perceived by a human observer as having the same expression even if the geometric displacements are significantly different, and this should be valid not only for *e.g.*the coarse "smile" expression label, but also for all the subtle variations that exist. Ideally, the latent loss should focus too on this aspect.

We present here an altogether different technique for learning a decoupled 3D face model, and investigate the use of Generative Adversarial Networks (GANs) [Goodfellow et al., 2014] for this task. GANs are well suited to our problem since the loss function is designed to evaluate the *output* and not the structure of the latent space, while performing a non-linear transformation between the latent variables and the resulting mesh. Furthermore, the proposed approach is scalable in terms of the number of factors to be modeled, requiring only sparsely labeled data and a new discriminator for each additional factor.

A key challenge is how to best represent the data to enable stable training of GANs. While current deep learning techniques have shown impressive results in the image domain, extending these to 3D data is not straightforward. We propose here a novel 3D-2D architecture in which a multilayer perceptron (MLP) generates the 3D face shape given a latent code, while a regular convolutional network is used as a 2D discriminator. This is allowed by an intermediate *geometry mapping* layer that transforms a 3D surface mesh into a geometry image encoding the mesh vertex locations.

To effectively decouple the factors of variation we build on auxiliary classifiers [Odena et al., 2017] whose task is to correctly guess the label associated with each factor (*e.g.*"happy" expression), and introduce a loss on the classifier features for unlabeled samples. Comparisons with recent approaches based on autoencoder architectures [Fernández Abrevaya et al., 2018, Ranjan et al., 2018] demonstrate that the proposed model can better decouple identity and expression, and exhibit more variability in the generated data.

In summary, this chapters contributes:

2. A concurrent work was published that tackles this, see Jiang et al. [2019].

1. A new generative 3D face model that captures non-linear deformations due to expression, as well as the relationship between identity and expression subspaces.
2. A novel 3D-2D architecture that allows to generate 3D meshes while leveraging the discriminative power of CNNs, by introducing a *geometry mapping layer* that acts as bridge between the two domains.
3. A training scheme that enables to effectively decouple the factors of variation, leading to significant improvements with respect to the state of the art.

5.1 Related Work

We focus in the following on closely related deep learning works for 3D face modeling and disentanglement. A more detailed review on classic data-driven models such as the 3D Morphable Model (3DMM) can be found in Section 2.3.

Autoencoders for 3D faces Recent works leverage deep learning methods to overcome the limitations of (multi-)linear models. Ranjan et al. [2018] proposed an autoencoder architecture that learns a single global model of the 3D face, and as such the different factors cannot be decoupled directly. However, an extension called DeepFLAME was proposed that combines a linear model of identity [Li et al., 2017] with the autoencoder trained on expression displacements. While expressions are modeled non-linearly, the relationship between identity and expression is not addressed explicitly. In Chapter 3 we developed the multilinear autoencoder (MAE) [Fernández Abrevaya et al., 2018] in which the decoder is a multilinear tensor structure. While the relationship between the two spaces is accounted for, transferring expressions still presents artifacts. We compare our proposed approach to DeepFLAME and MAE, as they achieve state-of-the-art results on decoupling identity and expression variations.

Bagautdinov et al. [2018] proposed a multiscale model of 3D faces at different levels of geometric detail. Two recent works [Tran et al., 2018, Tewari et al., 2018] use autoencoders to learn a global or corrective morphable model of 3D faces and their appearance based on 2D training data. However, none of these methods allow to disentangle factors of variation in the latent space. Unlike the aforementioned works, we investigate the use of GANs to learn a decoupled model of the 3D face.

GANs for 3D faces Some recent works have proposed to combine a 3DMM with an appearance model obtained by adversarial learning. Slossberg et al. [2018] train a GAN on aligned facial textures and combine this with a linear 3DMM to generate realistic synthetic data. Gecer et al. [2019] train a similar model and show that GANs can be used as a texture prior for accurate fitting to 2D images. Deng et al. [2018] fit a 3DMM to images and use a GAN to complete the missing parts of the resulting UV map. All of these methods rely on linear 3DMMs, and hence to shape spaces limited in expressiveness. While

the focus is on improving the appearance, we follow a different objective with a generative shape model that decouples identities and expressions.

To the best of our knowledge, by the time of publication the only work that learned 3D facial shape variations using a GAN was Shamai et al. [2019], which is an extension of Slossberg et al. [2018]. The authors proposed to learn identity variations by training a GAN on geometry images, but unlike our work they do not model the non-linear variations due to expression nor the correlation between identity and expression, since the main focus is on appearance. Recently, Moschoglou et al. [2020] combined an autoencoder architecture with a GAN adversarial loss to learn shape variations, where geometry is again encoded in a UV map. Unlike these works, we propose to generate the shapes directly in the 3D domain, and use geometry images only for discrimination.

Two other methods learn to enhance an input 3D face geometry with photometric information using a GAN. Given a texture map and a coarse mesh, Huynh et al. [2018] augment the latter with fine scale details, and given an input image and a base mesh, Yamaguchi et al. [2018] infer detailed geometry and high quality reflectance. Both works require the conditioning of an input, and unlike us they do not build a generative 3D face model.

Generative models with disentangled representations The problem of learning disentangled representations has received considerable attention in the machine learning community, see *e.g.* [Bengio et al., 2013, van Steenkiste et al., 2019]. When full label supervision is available, bilinear [Tenenbaum and Freeman, 2000] and multi-linear [Vasilescu and Terzopoulos, 2002, Vlasic et al., 2005] models were initially proposed to disentangle known factors of variation. More recently, Reed et al. [2014] extended a Restricted Boltzmann Machine by clamping parts of the hidden units assigned to a specific factor, and Dosovitskiy et al. [2015], Kulkarni et al. [2015] trained deep neural networks to generate 2D projections of 3D objects from high-level descriptions. Weaker forms of labeling were considered in Reed et al. [2015], Mathieu et al. [2016], Jha et al. [2018], and fully unsupervised approaches were proposed by Chen et al. [2016], Higgins et al. [2017], Kim and Mnih [2018]. Because of the lack of supervision these methods cannot control which factors are encoded.

In their original form, GANs are unable to explicitly disentangle latent factors according to known features or attributes. Numerous works have been proposed that modify certain factors of an input image using GANs conditioned on an image and control labels, *e.g.* Tran et al. [2017b], Pumarola et al. [2018], Shen et al. [2018], Zhao et al. [2018], Usman et al. [2019]. They all require explicit conditioning on a key factor (*e.g.*expression, rotation, lighting) as well as identity in the form of an input image, whereas we aim here to learn the latent spaces implicitly. To our knowledge only a few works decouple without conditioning on an input shape. Mathieu et al. [2016] combine an encoder-decoder generator and a reconstruction loss with swapped latent vectors to disentangle identity, but only experimented with very low resolution

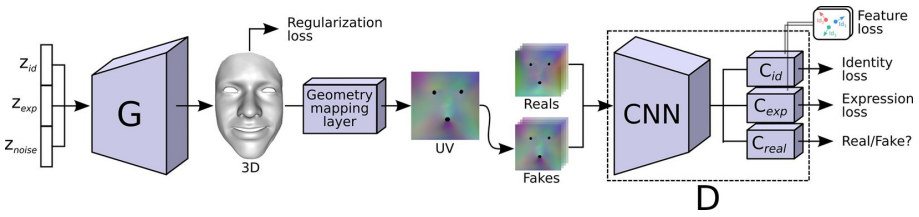


Figure 5.1 – Proposed architecture. A MLP generates the 3D coordinates of the mesh, while discrimination occurs in 2D space thanks to the *geometry mapping* layer. Identity and expression codes z_{id}, z_{exp} are used to control the generator, and classification losses are added to decouple between the two. A feature loss is introduced to ensure consistency over features with fixed identities or expressions.

images. Donahue et al. [2018] decouples by classifying pairs with a common identity. Neither of these are symmetrical with respect to the two factors to disentangle as they focus on preserving identity only. We propose here an alternative that succeeds in decoupling latent codes into a constant number of separate factors.

5.2 Background

Generative Adversarial Networks (GANs) [Goodfellow et al., 2014] are based on a minimax game, in which a discriminator D and a generator G are optimized for competing goals. The discriminator is tasked with learning the difference between real and fake samples, while the generator is trained to maximize the mistakes of the discriminator. At convergence, G approximates the real data distribution. Training involves the optimization of the following:

$$\min_G \max_D \mathcal{L}_{GAN} = \mathbb{E}_{x\sim p_{data}}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))], \tag{5.1}$$

where p_{data} denotes the distribution of the training set, and p_z denotes the prior distribution for G, typically $\mathcal{N}(0, I)$.

GANs have been shown to be very challenging to train with the original formulation and prone to low diversity in the generated samples. To address this, Arjovsky et al. [2017] propose to minimize instead an approximation of the Earth Mover's distance between generated and real data distributions, which is the strategy we adopt in this work:

$$\mathcal{L}_{GAN} = \mathbb{E}_{x\sim p_{data}}[D(x)] - \mathbb{E}_{z\sim p_z}[D(G(z))]. \tag{5.2}$$

In particular we use the extension of Gulrajani et al. [2017] which uses a gradient penalty in order to enforce that D is 1-Lipschitz.

When labels are available, using them has proven to be beneficial for GAN performance. Odena et al. [2017] proposed the Auxiliary Classifier GANs (AC-GAN), in which D is augmented so that it outputs the probability of an image

belonging to a pre-defined class label $c \sim p_c$. In this case, the loss function for G and D is extended with:

$$\mathcal{L}_C^{real} = \mathbb{E}_{x \sim p_{data}, c \sim p_c}[\log P(C = c|x)], \tag{5.3}$$

$$\mathcal{L}_C^{fake} = \mathbb{E}_{z \sim p_z, c \sim p_c}[\log P(C = c|G(z, c))]. \tag{5.4}$$

In order to evaluate if a model is correctly decoupling, we need to be able to distinguish whether two identites or expressions sharing the same latent code are perceptually similar. Thus, our work builds on the idea of auxiliary classifiers in order to learn a decoupling of the shape variations into factors, as will be explained in the next section.

5.3 Method

We consider as input a dataset of registered and rigidly aligned 3D facial meshes, where each mesh is defined by $(\mathcal{V}, \mathcal{F})$, the set of 3D vertices $\mathcal{V} \in \mathbb{R}^{3 \times n_v}$ and the set of triangular faces $\mathcal{F} \in \mathbb{N}^{3 \times n_f}$ that connect the vertices. Our goal is to build an expressive model that can decouple the representation based on known factors of variation. In contrast to classical approaches in which a reconstruction error is optimized, we rely instead on the adversarial loss enabled by a convolutional discriminator. To this end, we introduce an architecture in which a *geometry mapping layer* serves as bridge between the generated 3D mesh and the 2D domain, for which convolutional layers can be applied (Section 5.3.1). To learn a decoupled parameterization, we build on the idea of Auxiliary Classifiers and introduce a feature loss to further improve the results (Section 5.3.3). We will consider here a model that decouples between identity and expression, however the principle can be easily extended to more factors.

5.3.1 Geometry Mapping Layer

While deep learning can be efficiently used on regularly sampled signals, such as 2D pixel grids, applying it to 3D surfaces is more challenging due to their irregular structure. In this work we propose to generate the 3D coordinates of the mesh using a multilayer perceptron, while the discriminative aspects are handled in the 2D image domain. This allows to benefit from efficient and well established architectures that have been proven to behave adequately under adversarial training, while still generating the 3D shape in its natural domain.

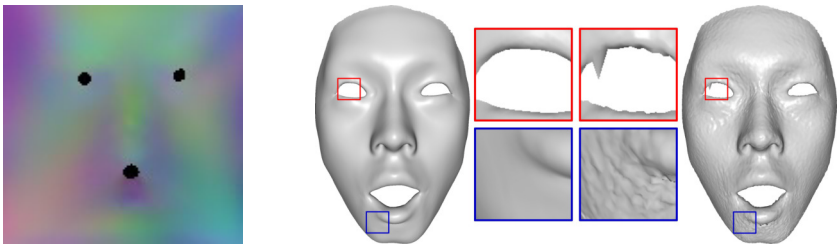
In particular, a 2D representation of a mesh can be achieved through a UV parameterization $\phi : \mathcal{V} \rightarrow D$ that associates each vertex $v_k \in \mathcal{V}$ with a coordinate $(u, v)_k$ in the unit square domain D. Continuous images can be obtained by interpolating the (x, y, z) vertex values according to the 2D barycentric coordinates, and storing them in the image channels. Borrowing the term from Gu et al. [2002], we call this a *geometry image* (see Figure 5.2a).

Note that although our method could generate geometry images instead of 3D meshes, this would introduce an unnecessary additional reconstruction step that is likely to cause information loss and artifacts in the final meshes, as illustrated in Figure 5.2b. This is due to the fact that a single planar unfolding of a mesh may create distortions such as triangle flipping [Sheffer et al., 2006], and a many-to-one mapping may be obtained even with a bijective parameterization due to the finite size of images. In addition, as elaborated in Gu et al. [2002], unless border vertices are preassigned to distinct pixels which can be challenging for large meshes, sampling these locations results in erroneous interpolations. Generating 3D point coordinates instead allows to avoid reconstruction artifacts, and to apply common mesh regularization techniques that simplify and improve the learning process. We use geometry images only as the representation for the discriminative component that evaluates the 3D generator through CNNs.

The mapping layer operates as follows. Given a mesh made of vertices $\mathcal{V} = \{v_k / k = 1..n_v\}$, a target image size $n \times n$, and a pre-computed UV parameterization ϕ, we build two images $I^{\mathcal{U}}$, $I^{\mathcal{V}}$ of dimension $n \times n$, and three images I^{v_1}, I^{v_2} and I^{v_3} of dimension $n \times n \times 3$ each. For each pixel (i, j), we consider the ϕ-projected mesh triangle $(\hat{v}_1, \hat{v}_2, \hat{v}_3)$ containing it. The barycentric abscissa and ordinate of pixel (i, j) in triangle $(\hat{v}_1, \hat{v}_2, \hat{v}_3)$ are then stored in images $I^{\mathcal{U}}$ and $I^{\mathcal{V}}$ respectively, and the original face vertex coordinates v_1, v_2 and v_3 are stored in images I^{v_1}, I^{v_2} and I^{v_3}. The mapping layer computes the output geometry image $\mathcal{I}$ as:

$$\mathcal{I} = I^{\mathcal{U}} * I^{v_1} + I^{\mathcal{V}} * I^{v_2} + (\mathbf{1} - I^{\mathcal{U}} - I^{\mathcal{V}}) * I^{v_3}, \tag{5.5}$$

where $*$ denotes element-wise multiplication and $\mathbf{1} \in \mathbb{R}^{n \times n}$ is the matrix of ones. Since this layer simply performs indexing and linear combinations on the elements of $\mathcal{V}$ using the predefined parameters in $I^{\mathcal{U}}$ and $I^{\mathcal{V}}$, all operations are differentiable and the gradients can be back-propagated from the discriminated image to the generated mesh.



(a) Geometry image (b) Original and reconstructed meshes

Figure 5.2 – While a GAN could be used to generate geometry images, recovering the mesh from them is prone to artifacts, e.g. erroneous boundary interpolations (red) and precision loss (blue) in 5.2b. In this work we generate instead the 3D mesh, while geometry images are used only for discrimination.

Convolving the UV map obtained through the geometry mapping layer corresponds to convolving the original mesh, such that the convolutional kernel covers the surface in a possibly un-even manner, as illustrated in Figure 5.3. This process allows to take advantage of the efficiency of regular grids while still generating shapes in 3D space, retaining knowledge of the surface connectivity.

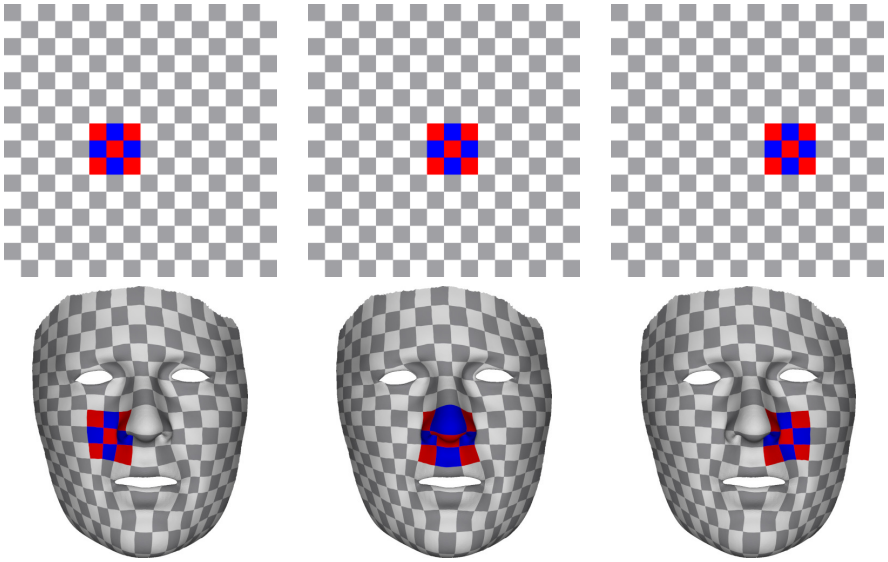


Figure 5.3 – Illustration of mesh convolutions using the geometry mapping layer. The top row represents a UV map with a convolutional kernel being applied, while the bottom row shows the effect on the original mesh. The geometry mapping layer allows to backpropagate results from the image representation to the generated mesh.

5.3.2 Architecture

Figure 5.1 depicts our proposed architecture. The generator consists of two fully connected layers that map the latent code z to a vector of size $3n_v$ containing the stacked 3D coordinates of displacements from a reference face mesh. The output vertex positions are passed through the mapping layer to generate a geometry image of size $n \times n$, which is then processed by the discriminator in order to classify whether the generated mesh is real or fake. We also consider auxiliary classifiers for the discriminator, denoted as C_{id} and C_{exp}. The design of D shows two main differences with respect to the original AC-GAN. First, instead of classifying only one type of label, we use here classifiers for both identity and expression. This favors decoupling, since the classification of one factor is independent of the choice of parametrization for the other factors. Second, we provide distinct convolutional layers for the real/fake, identity and expression blocks. This is motivated by the observation that the

features required to classify identities and expressions are not necessarily the same.

5.3.3 Decoupled Model Learning

We rely on the discriminator not only to generate realistic faces, but also to decouple the factors of variation. For this, we optimize D such that it maximizes

$$\mathcal{L}_D = \mathcal{L}_{GAN} + \lambda_C(\mathcal{L}_{ID} + \mathcal{L}_{EXP}). \tag{5.6}$$

Here, $\mathcal{L}_{GAN}$ denotes the standard adversarial loss (see Equation 5.2), and $\mathcal{L}_{ID}, \mathcal{L}_{EXP}$ the classification losses measured against the labels provided with the dataset and weighted by scalar λ_C. These losses are defined similarly to Equation 5.3 as:

$$\begin{aligned} \mathcal{L}_{ID} &= \mathbb{E}_{x\sim p_{data}, c\sim p_c^{id}}[\log P(C = c|x)], \\ \mathcal{L}_{EXP} &= \mathbb{E}_{x\sim p_{data}, c\sim p_c^{exp}}[\log P(C = c|x)], \end{aligned} \tag{5.7}$$

where p_c^{id} and p_c^{exp} denote the distribution of identity and expression labels, respectively. We ignore the sample contribution in the classification loss if it is not labeled.

The generator G takes as input a random vector $z = \{z_{id}, z_{exp}, z_{noise}\}$, which is the concatenation of the identity code $z_{id} \sim p_{id}$, the expression code $z_{exp} \sim p_{exp}$ and a random noise $z_{noise} \sim p_{noise}$. It produces the location of n_v displacement vectors from a reference mesh, and is trained by minimizing:

$$\begin{aligned} \mathcal{L}_G = \lambda_1 \mathcal{L}_{GAN} - \lambda_2 \left(\mathcal{L}_{CL}^{id} + \mathcal{L}_{CL}^{exp}\right) \\ +\lambda_3 \left(\mathcal{L}_{FEAT}^{id} + \mathcal{L}_{FEAT}^{exp}\right) + \lambda_4 \mathcal{L}_{reg}, \end{aligned} \tag{5.8}$$

where $\mathcal{L}_{GAN}$ is the standard GAN loss (Equation 5.2); $\mathcal{L}_{CL}^{id}$ and $\mathcal{L}_{CL}^{exp}$ are classification losses; $\mathcal{L}_{FEAT}^{id}$ and $\mathcal{L}_{FEAT}^{exp}$ are feature losses that aim to further increase the decoupling of the factors; $\mathcal{L}_{reg}$ is a regularizer; and $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are weights for the different loss terms. We explain each of these in the following.

Classification Loss In addition to the adversarial loss, the generator is trained to classify its samples with the correct labels by maximizing:

$$\begin{aligned} \mathcal{L}_{CL}^{id} &= \mathbb{E}_{z\sim p_z, c\sim p_c^{id}}[\log P(C = c|G(z))], \\ \mathcal{L}_{CL}^{exp} &= \mathbb{E}_{z\sim p_z, c\sim p_c^{exp}}[\log P(C = c|G(z))]. \end{aligned} \tag{5.9}$$

In order to generate data belonging to a specific class, we sample one identity/expression code z_{id}, z_{expr} for each label and fix it throughout the training; this becomes the input for G each time the classification loss must be evaluated. We denote the set of fixed codes for identity and expression as $\mathcal{T}^{id}$ and $\mathcal{T}^{exp}$ respectively.

Feature Loss The classification loss is limited to codes in $\mathcal{T}^{id}/\mathcal{T}^{exp}$, which have associated labels. We found that better decoupling results can be obtained if we

include a loss on the classifier features. We measure this by generating samples in pairs which share the same identity or expression vector, and measuring the error as:

$$\mathcal{L}_{FEAT}^{id} = \frac{2}{N}\sum_{z_{id}}\Big(1 - cos(\mathbf{f}_{1,z_{id}}, \mathbf{f}_{2,z_{id}})\Big), \tag{5.10}$$

$$\mathcal{L}_{FEAT}^{exp} = \frac{2}{N}\sum_{z_{exp}}\Big(1 - cos(\mathbf{f}_{1,z_{exp}}, \mathbf{f}_{2,z_{exp}})\Big). \tag{5.11}$$

Here, N is the batch size, and $\mathbf{f}_{i,z_{id}} = \mathbf{f}\Big(G(z_{id}, z_{exp,i}, z_{noise,i})\Big)$ are feature vectors obtained by inputting the sample $G(z_{id}, z_{exp,i}, z_{noise,i})$ through the classifier C_{id} and extracting the features from the second to last layer. That is, given two inputs which were generated with the same identity vector, $\mathcal{L}_{FEAT}^{id}$ enforces that their feature vectors in the identity classifier are also aligned. The definition is analogous for $\mathbf{f}_{i,z_{exp}}$ with C_{exp}.

To enable training with both classification and feature loss, for each batch iteration we alternate between the sampling of labeled identity codes $z_{id} \in \mathcal{T}^{id}$ with unlabeled expression codes $z_{exp} \sim p_{exp}$, and the sampling of unlabeled identity codes $z_{id} \sim p_{id}$ with labeled expression codes $z_{exp} \in \mathcal{T}^{exp}$. The classification is evaluated for the labeled factor only, while the feature loss is used for unlabeled codes, and the alternation allows to better cover the identity and expression sub-spaces during training.

Regularization Generating a 3D mesh allows us to reason explicitly at the surface level and define high order loss functions using the mesh connectivity. In particular, we enforce spatial consistency over the generated faces by minimizing the following term on the output displacements $\mathbf{v} = G(z)$:

$$\mathcal{L}_{reg} = \|L\mathbf{v}\|_2^2, \tag{5.12}$$

where L is the cotangent discretization of the Laplace-Beltrami operator [Meyer et al., 2003].

5.4 Evaluation

We provide in this section results obtained with the proposed framework, which demonstrate its benefits particularly in decoupling. We first clarify our set-up with implementation details in Section 5.4.1 and the datasets used in 5.4.2. We explain in Section 5.4.3 the proposed metrics for the evaluation of a 3D face model, and introduce a new measure for analyzing the diversity of the generated samples. In Section 5.4.4 we perform ablation studies to verify that all the components are necessary to effectively train an expressive model. Finally, in Section 5.4.5 we compare our results to state-of-the-art 3D face models that can decouple the latent space, and show that our approach outperforms with respect to decoupling and diversity.

5.4.1 Implementation Details

We set the weights to $\lambda_C = 0.1$ (Equation 5.6), $\lambda_1 = \lambda_2 = 1$, $\lambda_3 = 0.5$ and $\lambda_4 = 100$ (Equation 5.8). The classification losses are further weighted to account for imbalanced labels [King and Zeng, 2001]. For the generator, we use two fully connected layers with an intermediate representation of size 512 and ReLU non-linearity. For the discriminator we use a variant of DC-GAN [Radford et al., 2016], with the first two convolutional blocks shared between C_{real}, C_{id} and C_{expr}, while the remaining are duplicated for each module (more details can be found in Appendix A.1). The models were trained for 200 epochs using ADAM optimizer [Kingma and Ba, 2015] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, a learning rate of 0.0002 and a batch size of 64. During training we add instance noise [Sønderby et al., 2017] with $\sigma = 0.1$ to the input of D. The discriminator is trained for 3 iterations each time we train the generator. The models take around 2 hours to train on a NVidia GeForce GTX 1080 GPU.

The template mesh contains 22129 vertices. We pre-compute the UV map ϕ using harmonic parameterization [Eck et al., 1995], setting the outer boundary face vertices to a unit square to ensure full usage of the image domain. We generate geometry images of size 64×64; we experimented with other image sizes but the best decoupling results were obtained with this resolution. The dimensions for $(z_{id}, z_{exp}, z_{noise})$ are set to $(65, 15, 5)$ to facilitate comparison with Fernández Abrevaya et al. [2018], and the feature vectors used in Equations 5.10 and 5.11 are of size 2048.

5.4.2 Datasets

All models were trained using a combination of four publicly available 3D face datasets. In particular, we use two datasets containing static 3D scans of multiple subjects: BU-3DFE [Yin et al., 2006] and Bosphorus [Savran et al., 2008], and combine these with two datasets of 3D motion sequences of multiple subjects: BP4D-Spontaneous [Zhang et al., 2014] and BU-4DFE [Yin et al., 2008]. The static datasets provide variability of identities, while the motion datasets provide variability of expressions and a larger number of training samples. We registered BU-3DFE and Bosphorus with a template fitting approach [Salazar et al., 2014], and the motion datasets with the spatiotemporal approach introduced in Chapter 4.

The final dataset contains 30559 registered 3D faces and was obtained by subsampling the motion sequences. We provide identity labels for all meshes, while the expression labels are limited to the seven basic emotional expressions, which appear in both static datasets. For BU-4DFE, expression labels are assigned to three frames per sequence: the neutral expression to the first and last frame, and the labeled expression of the sequence to the peak frame. For BP4D, one neutral frame is manually labeled per subject (this is a requirement for comparison to Ranjan et al. [2018]). Overall, due to the use of motion data, only 7% of it is assigned expression labels.

5.4.3 Evaluation Metrics

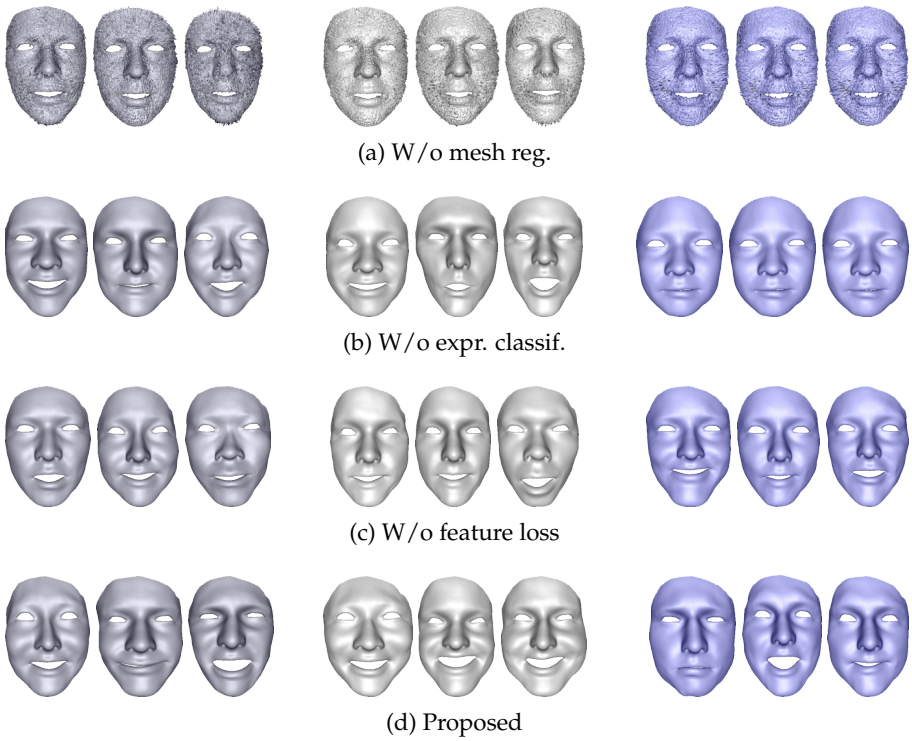
We evaluate the models in terms of *diversity* of the generated samples, *decoupling* of identity and expression spaces, and *specificity* to the 3D facial shape. We believe it is necessary to simultaneously consider all the metrics, as they provide complementary information on the model. For instance, a good decoupling value can be obtained when the diversity is poor, since small variations facilitate the classification of samples as "same". Conversely, a large diversity value can be obtained when decoupling is poor, since the identities/expressions sharing the same code can yield very different shapes. We detail these in the following.

Diversity We consider it important to measure the diversity of the 3D face shapes generated by a model, particularly with GANs that are known to be prone to mode collapse. To the best of our knowledge, this has not yet been considered in the context of 3D face models and we propose therefore to evaluate as follows. We sample p pairs of randomly generated meshes and compute the mean vertex distance among the pairs; *diversity* is then defined as the average of distances over the p pairs. We expect here to see higher values for more diverse models. We evaluate on three sets of sampled pairs: (1) among pairs chosen randomly (*global diversity*), (2) among pairs that share the same identity code (*identity diversity*) and (3) among pairs that share the same expression code (*expression diversity*). For all cases we evaluate on 10000 pairs. For comparison, the training set is also evaluated on these three metrics by leveraging the labels.

Decoupling To evaluate decoupling in both identity and expression spaces we follow the protocol proposed in Donahue et al. [2018]. In particular, we first train two networks, one for identity and one for expression, that transform an image representation of the mesh to an n-dimensional vector using triplet loss [Schroff et al., 2015], where $n = 128$ in our experiments. The trained networks allow to measure whether two meshes share the same identity or expression by checking whether the distance between their embeddings is below a threshold τ.

To measure identity decoupling, we generate n random faces $\mathbf{x}_i = G(z^i_{id}, z^i_{exp}, z^i_{noise})$, and for each random face we fix the identity code and sample m faces $\mathcal{Y}(\mathbf{x}_i) = \{G(z^i_{id}, z^j_{exp}, z^j_{noise}), j = 1..m\}$. We then use the embedding networks to evaluate whether the original faces $\mathbf{x}_i$ and their corresponding samples in $\mathcal{Y}(\mathbf{x}_i)$ correspond to the same identity, and report the percentage of times the pairs were classified as "same". We proceed analogously for expression decoupling. We set $n = 100$, $m = 100$, $\tau = 0.14$ for identity and $\tau = 0.226$ for expression; more implementation details are given in Appendix A.2.

Specificity Specificity is a metric commonly used for the evaluation of statistical shape models [Davies et al., 2008] and whose goal is to quantify whether



(a) W/o mesh reg.

(b) W/o expr. classif.

(c) W/o feature loss

(d) Proposed

Figure 5.4 – Qualitative results for alternative approaches. From left to right: randomly generated samples (dark gray), random samples with a same expression code (light gray), random samples with a same identity code (purple).

all the generated samples belong to the original shape class, faces in our case. For this, n samples are randomly drawn from the model and for each the mean vertex distance to each member of the training set is measured, keeping the minimum value. The metric then reports the mean of the n values. We use here $n = 1000$.

5.4.4 Ablation Tests

We start by demonstrating that each of the proposed components is necessary to obtain state-of-the-art results according the metrics previously defined. To this end, we compare our approach against the following alternatives: (1) without mesh regularization (Equation 5.12); (2) with identity classification only; (3) with expression classification only; and (4) without feature loss (Equations 5.10 and 5.11).

Table 5.1 gives the evaluation metrics for each of these options, and Figure 5.4 provides qualitative examples. From the results we observe that: (1) The mesh regularization is crucial to generate samples that are realistic facial shapes. This is reflected by a very large value in specificity as well as low diversity, due to the fact that the model never converged to realistic faces (see Figure 5.4a). (2)

	Dec-Id↑	Dec-Exp↑	Div↑	Div-Id↑	Div-Exp↑	Sp.↓
Training data	–	–	4.89	3.30	5.04	–
w/o mesh regularization	99.6	99.1	1.41	0.65	1.25	3.61
w/o expr. classification	100.0	42.8	4.81	0.11	4.87	2.01
w/o id. classification	7.8	98.9	5.28	4.87	2.05	2.22
w/o feature loss	96.0	80.3	4.47	1.75	4.01	2.00
3DMM	99.6	65.6	3.53	1.95	2.89	2.30
MAE	99.5	53.3	3.89	0.92	3.76	2.00
CoMA	97.5	65.5	3.38	1.71	2.90	2.47
Ours	98.6	89.7	4.74	1.94	4.22	2.01

Table 5.1 – Quantitative evaluation with respect to decoupling of identity and expression (*Dec-*, percentage), diversity (*Div-*, in mm) and specificity (*Sp.*, in mm.); and comparisons to 3DMM [Amberg et al., 2008], MAE [Fernández Abrevaya et al., 2018] and CoMA [Ranjan et al., 2018]. Higher is better, except for specificity.

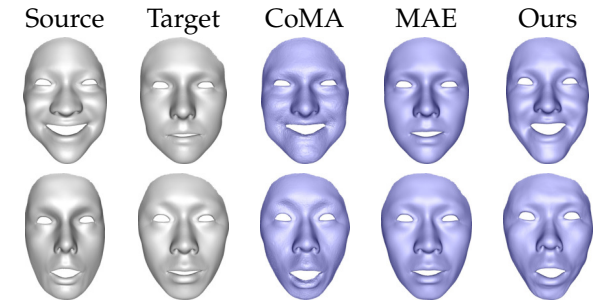
Considering classification in only one factor significantly reduces the capacity of the model to preserve semantic properties in the other factor, as indicated by the very low decoupling values obtained in the corresponding rows. This justifies the use of classifiers for each of the factors. (3) Without the feature loss the model can still achieve good results, but both expression decoupling and diversity are lower than with the full model and the inclusion of the feature loss improves expression classification by almost 10%. Note that decoupling the expression space is significantly more challenging than identity, as the provided labels are very sparse. This effect is illustrated on Figure 5.4c, where models with the same expression code can lead to faces with slightly different expressions. Our approach provides more coherent faces, as shown in Figure 5.4d.

5.4.5 Comparisons

We compare the proposed approach against state-of-the-art generative 3D face models. Our goal is to build a decoupled latent space, and thus we focus the comparison to works that either enforce this explicitly [Fernández Abrevaya et al., 2018], or combine a model trained on expressions with a linear space of identities [Ranjan et al., 2018, Amberg et al., 2008]. We train all models using the same dimensions (65 for identity and 20 for expression).

The model proposed in Fernández Abrevaya et al. [2018], called MAE in the following, was trained with the same dataset and the same label information (Section 5.4.2) for 200 epochs, with the default parameters given in the paper. We initialize the encoder and the decoder from the publicly available models.

The model proposed in Ranjan et al. [2018], called CoMA in the following, does not explicitly favor decoupling and thus we use the DeepFLAME



(a) Transferring expression to a target face

(b) Sampling novel identities from the transferred expression.

Figure 5.5 – Qualitative comparison in terms of expression transfer. Top: expression code z_{expr} transferred to a target identity. Bottom: using z_{expr} from the source in the top row, we sample novel identities (left to right: CoMA, MAE, ours).

alternative [Li et al., 2017], which we also train with the same dataset. This results in a PCA model built from 299 identities and an autoencoder trained on 30330 displacements from the corresponding neutral face. For the identity space we manually selected one neutral frame for each sequence in BP4D-Spontaneous, as this dataset does not provide labels. The model was trained using the publicly available code for 200 epochs.

We also trained an additive linear model as described in Amberg et al. [2008] using our dataset, and the same neutral/expression separation selected for CoMA (see above). We refer to this model as 3DMM.

Model quality We show quantitative results with respect to decoupling, diversity and specificity in the bottom of Table 5.1. Note that the proposed approach significantly outperforms the others in terms of expression decoupling, which is more challenging than identity due to the sparse labeling. This is shown qualitatively in Figure 5.5, where we transferred expressions by simply exchanging the latent code z_{exp}. We can see here that the expression is well preserved by our model.

With respect to identity decoupling the four methods perform similarly well, with 3DMM achieving the highest value. Note that, in the case of MAE, the large decoupling value is combined with the lowest diversity in identity (*Div-Id*), which suggests limited generative capabilities.

Our model outperforms all methods in terms of diversity. Combined with

Method	$\lambda = 0$	$\lambda = 0.01$	$\lambda = 10$
3DMM [Amberg et al., 2008]	6.62	4.64	2.46
MAE [Fernández Abrevaya et al., 2018]	4.46	4.06	2.78
CoMA [Ranjan et al., 2018]	3.05	3.02	2.83
Ours	**2.62**	**2.55**	**2.42**

Table 5.2 – Reconstruction of sparse data under different regularization weights (RMSE, in mm).

a specificity value that is among the best, this implies that it has learned to generate significant variations that remain valid facial shapes.

Reconstruction of Sparse Data We test here model generalization when reconstructing partial face data given sparse constraints. To this purpose, we use the dataset provided by Ranjan et al. [2018], which contains 12 subjects performing 12 extreme expressions. We take the middle frame of each sequence and manually label 85 landmarks (see Figure 5.9b), resulting in a testing set of 144 subjects. The face model is fitted by minimizing:

$$\arg\min_{z} \sum_{i=1}^{p} \|\tilde{\mathbf{v}}_i(z) - \mathbf{v}_i\|_2^2 + \lambda \|z\|_2^2, \tag{5.13}$$

where $\mathbf{v}_i$ are the 3D locations of the p key-points in the testing set, $\tilde{\mathbf{v}}_i(z)$ are the corresponding key-points in the face model generated with code z, and λ the regularization weight. We optimize using a gradient descent approach [Kingma and Ba, 2015] starting from a randomly sampled code z. Note that this is a challenging scenario since the training set does not contain such expressions, and the correspondences are very sparse.

We compare our results with those obtained with 3DMM, MAE and CoMA, using the same optimization for all methods. We measure the reconstruction error against the ground-truth surface and report the RMSE. Quantitative results can be found in Table 5.2 for different regularization weights λ. Our method outperforms in all cases, including without regularization ($\lambda = 0$). We found that our model can produce reasonable faces in most cases, while MAE and CoMA easily produce un-realistic faces when the regularization is not strong enough. Qualitative examples can be seen in Figure 5.9a.

5.4.6 Extension to other factors

One of the benefits of our framework lies in its ability to easily extend to other factors of variation. As an illustration, we trained a model that decouples identity, expression and viseme (the visual counterpart of a phoneme). The results can be found in Figure 5.6, where we show qualitative examples obtained by modifying the different factors of variation individually.

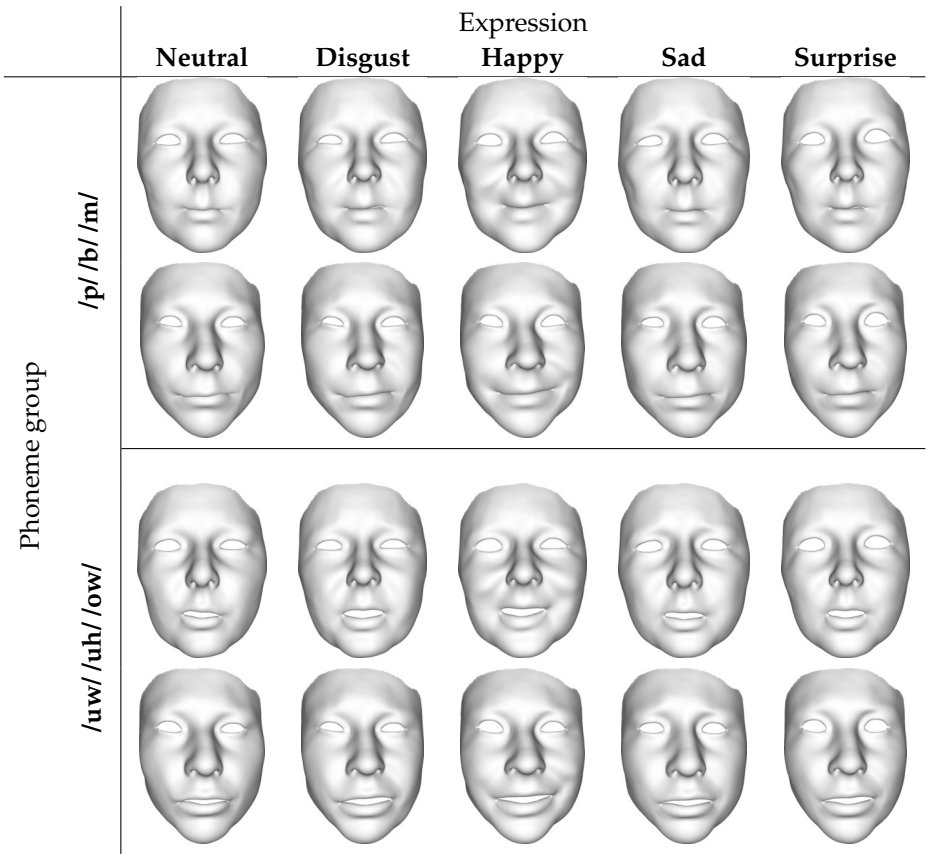


Figure 5.6 – Example of decoupling between identity, expression and viseme.

We trained the model using the audiovisual 3D dataset of Fanelli et al. [2010], which contains 14 subjects performing 40 speech sequences in neutral and "expressive" mode. We assign phoneme labels using the Montreal Forced Aligner tool [McAuliffe et al., 2017] with the provided audio, which are mapped to visemes following Neti et al. [2000]. For expression, we manually labeled 699 frames with the aid of the provided expression ratings of each sequence. This resulted in a database with 100% labeled identites, 68% labeled visemes, and 3% labeled expressions. We set the latent dimensions to $(50, 50, 50, 5)$ for identity, expression, viseme and noise, respectively.

Note that this is a simplified model of speech, since the temporal information is not taken into account. Yet, we can see in Figure 5.6 that a decoupling between the aspects affected by phoneme production, and those affected by expressions such as happiness or surprise can be easily distinguished by our framework. Note for example the change in eye expression and the subtle mouth movements that occur to accommodate the viseme under different expressions. It is also worth noting that these results were obtained with fully automatic labels for viseme, and very sparse manual labels for expression, thus simplifying the efforts required to annotate the dataset. Unlike the identity

and expression factors, which are intuitively easier to separate, the viseme and expression factors are more intertwined and decoupling them is very challenging even for a human annotator. In spite of this, our results show that we can reasonably decouple the three factors.

5.4.7 Latent Space Manipulation

Thanks to the decoupling of identity and expression spaces, we can synthesize new expressions by simple manipulation of the latent space. We show here two possibilities for this.

Given a source mesh obtained with $G(z_{id}^{src}, z_{expr}^{src}, z_{noise}^{src})$ and a target mesh obtained with $G(z_{id}^{target}, z_{expr}^{target}, z_{noise}^{target})$, we generate new expressions for the target mesh by either

1. Replacing the expression with that of the source: $G(z_{id}^{target}, \mathbf{z}_{\mathbf{expr}}^{\mathbf{src}}, z_{noise}^{target})$
2. Adding the expression vectors: $G(z_{id}^{target}, \mathbf{z}_{\mathbf{expr}}^{\mathbf{src}} + \mathbf{z}_{\mathbf{expr}}^{\mathbf{target}}, z_{noise}^{target})$

Results can be seen in Figure 5.7. In particular, note how adding the latent vectors results in plausible expressions which preserve the semantics of both sources.

Finally, we show that the latent space is smooth with an example of interpolation and extrapolation in Figure 5.8.

Figure 5.7 – Example of expression space manipulation. In gray a source mesh and a target mesh. In purple the result of (1) replacing the expression code of the target with that of the source *(replaced)*, and (2) adding the source and target expression codes *(added)*.

5.5 Conclusion

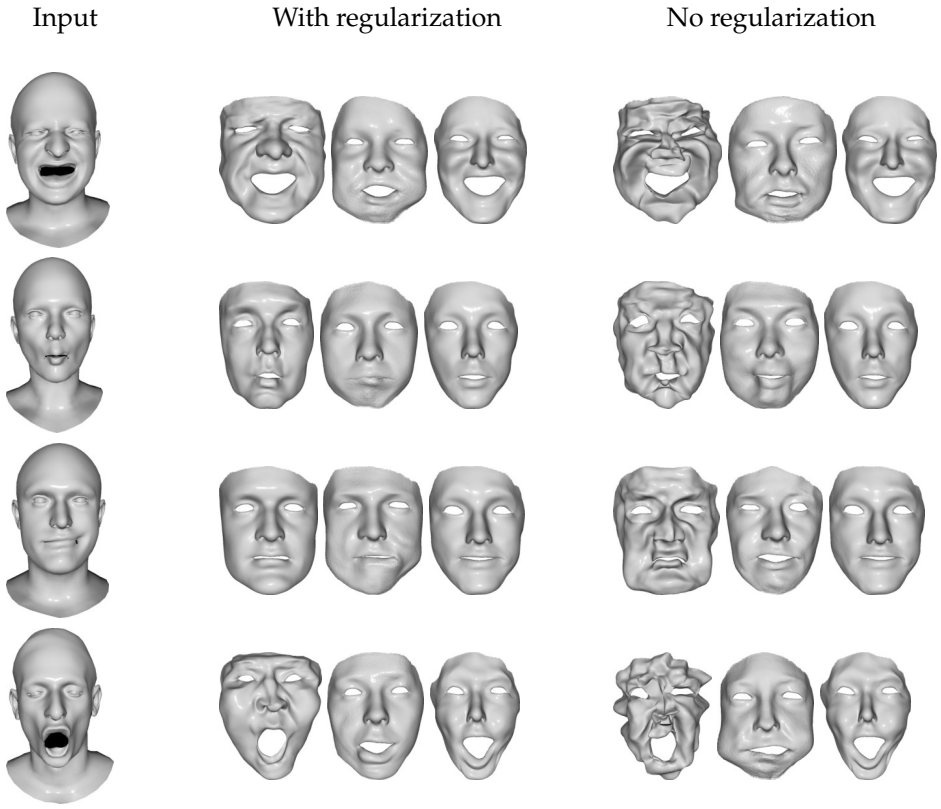
We explored in this work the use of adversarial training for learning decoupled 3D facial models. Our results show that purely discriminative losses are well suited for the decoupling task, achieving state-of-the-art performance

Figure 5.8 – From top to bottom: interpolation (purple) and extrapolation (gray) of expression code, identity code, and the full latent.

in terms of decoupling and diversity of the generated samples. Although the expressiveness of the model remains limited by the diversity of the training data and the accuracy of its labels, we show that adversarial learning has strong potential in building performant 3D facial models.

Our framework is enabled by a novel method for processing 3D data using deep learning approaches, namely the 3D-2D architecture. This architecture allows to benefit from advances in 2D convolutional networks while still generating the data in the 3D domain. The results of this work suggest that such 3D-2D approach is a viable alternative to other mesh processing frameworks, *e.g.* Verma et al. [2018]. Exploring its true capacity in light of other tasks is an interesting direction that we leave for future work.

GANs have shown an impressive ability to retain fine-scale details in 2D facial images. Unfortunately, due to the lack of proper training data the present work does not answer the question of whether similar progress can be obtained in 3D through the proposed approach. Yet, we believe that it should be possible given the proper data and enough network capacity, and leave this too as a future direction.



(a) Comparison against MAE and COMA, with and without regularization. From left to right: MAE, COMA, our result.

(b) 85 landmarks used for fitting

Figure 5.9 – Reconstruction of sparse data

6 Estimating 3D Face Normals from Natural Images

The previous chapters dealt with the *global* properties of the face: how to model and track the geometry of the 3D shape and motion, as well as the interactions between the identity and expression subspaces. In this final chapter we consider a slightly different but related problem: how to estimate local geometric details. To this end, we depart from the global shape models used until now and build instead a prior that can simultaneously encode information about the facial normals and natural face images. The goal here is to estimate accurate surface normals from images in-the-wild, which can be used to enhance coarser estimations from parametric models as illustrated in Figure 6.1.

3D reconstruction of the human face is a long-standing problem in computer vision, with a wide range of applications including biometrics, forensics, animation, gaming, and human digitalization. In many of these applications monocular inputs are considered in order to limit the acquisition constraints,

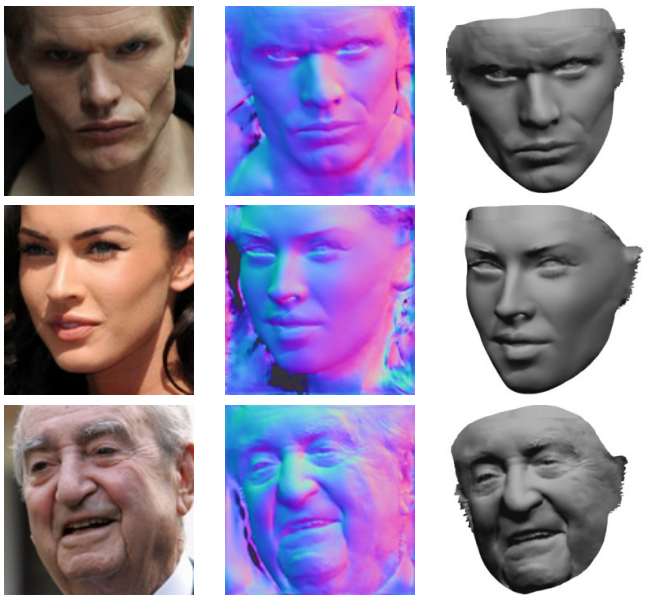


Figure 6.1 – Our model predicts accurate normals from a single input image that can be used to enhance a coarse geometry (e.g. PRN [Feng et al., 2018]).

hence enabling uncontrolled environments as well as efficient information usage for *e.g.*facial telecommunication and entertainment. Although significant progress has been recently made by the scientific community, recovering detailed 3D face models given only single images is still an open problem.

Monocular face reconstruction is in essence an ill-posed problem which requires strong prior knowledge. Assuming a simple shading model, seminal shape-from-shading (SfS) approaches [Horn and Brooks, 1989, Zhang et al., 1999] were estimating shape normals by considering local pixel intensity variations. Fine scale surface details can be recovered using this strategy, however the applicability to in-the-wild images is limited by the simplified image formation model that is assumed. Later on, a more global strategy was proposed with parametric face models, such as the ones considered in the previous chapters. They allow fitting a template face controlled by only a few coefficients, resulting hence in improved robustness. While being widely adopted, parametric models are inherently restricted in expressiveness and have difficulties in recovering small surface details, as a consequence of their low dimensional representation. Recently, deep learning methods that exploit large-scale face image datasets have been investigated with the aim of better generalization. While most works in this category are trained to estimate the coefficients of a parametric model [Tewari et al., 2017, 2018, Genova et al., 2018, Kim et al., 2018a, Sanyal et al., 2019], a few other approaches infer directly per-pixel depth [Sela et al., 2017], UV position maps [Feng et al., 2018] or surface normals [Trigeorgis et al., 2017, Sengupta et al., 2018].

As observed in previous work [Smith et al., 2019, Zhang and Funkhouser, 2018], regressing depth information alone can lead to suboptimal results, especially detail-wise, as the inherent scale ambiguity with single images can make convergence difficult for neural networks. On the other hand, the estimation of normals appears to be an easier task for such networks, given that normals are strongly correlated to pixel intensities and depend mostly on local information, a fact already exploited by SfS techniques. Still, only a few approaches have been proposed in this line for facial images [Shu et al., 2017, Sengupta et al., 2018], mostly due to the limited available ground-truth data. We propose here a method that overcomes this limitation and can leverage all data available through the use of cross-modal learning. Our experiments demonstrate that this strategy can estimate more accurate and sharper facial surface normals from single images.

The proposed approach recovers accurate normals corresponding to the facial region within an RGB image, with the goal of enhancing an existing coarse reconstruction, Feng et al. [2018] in our experiments. We cast the problem as a color-to-normal image translation, which can be in principle solved by combining an image encoder E_I with a normal decoder D_N as in Trigeorgis et al. [2017], and including skip connections between E_I and D_N [Ronneberger et al., 2015] in order to transfer details from the image domain to the normals domain. However, training such a network can prove difficult unless a large dataset of image/normal pairs, that ideally contains images in-the-wild, is available. In

practice few such datasets are currently publicly available, *e.g.* Zafeiriou et al. [2011], which were moreover captured under controlled conditions. To improve generalization, we propose to augment the architecture with a normal encoder E_N and an image decoder D_I, where all encoders/decoders share the same latent space. This augmented architecture provides additional constraints on the latent space with the auto-encoded image-to-image and normal-to-normal branches, effectively building a prior over both realistic facial shapes and realistic facial images. In order to keep advantage of the skip connections between E_I and D_N, while avoiding the resulting bonded connections between E_N with D_N that hamper the architecture, we introduce the *deactivable skip connections*. This allows skip connections to be turned on and off during training according to the type of data.

To summarize, we contribute in this chapter:

1. A framework that leverages cross-modal learning for the estimation of normals from a single face image in-the-wild.
2. The introduction of the *deactivable skip connection*.
3. An extensive evaluation that shows that our approach outperforms state-of-the-art methods on the Photoface [Zafeiriou et al., 2011] and Florence [Bagdanov et al., 2011] datasets, with up to nearly 10% improvements in angular error on the Florence dataset, as well as visually compelling reconstructions.

6.1 Related Work

We focus the discussion below on methods that consider 3D face reconstruction, or normal estimation, given single RGB images.

Reconstruction with Parametric Models 3D reconstruction from a single image is ill-posed and many methods resort therefore to strong priors with parametric face models such as blendshape [Garrido et al., 2013, Cao et al., 2015, Thomas and Taniguchi, 2016] or statistical models, typically the 3D Morphable Model (3DMM) [Blanz and Vetter, 1999]. These models are commonly used within an analysis-by-synthesis optimization [Romdhani and Vetter, 2005, Huber et al., 2016, Egger et al., 2018, Booth et al., 2018, Gecer et al., 2019] or, more recently, using deep learning to regress model parameters [Richardson et al., 2016, 2017, Tewari et al., 2017, Tran et al., 2017a, Genova et al., 2018, Feng et al., 2018, Kim et al., 2018a, Tewari et al., 2019, Sanyal et al., 2019], or alternatively to regress other face information using 3DMM training data, for instance volumetric information [Jackson et al., 2017], UV position map [Feng et al., 2018], normal map [Trigeorgis et al., 2017], depth map [Sela et al., 2017], or the full image decomposition [Shu et al., 2017, Sengupta et al., 2018, Kim et al., 2018b]. This strategy has proven robustness, however it is constrained by the parametric representation that offers limited expressiveness and fails in recovering fine scale details.

In order to improve the quality of the reconstructions several works have proposed to add medium-scale correctives on top of the parametric model [Li et al., 2013, Garrido et al., 2016a, Tewari et al., 2018], to train a local wrinkle regressor [Cao et al., 2015], or to learn deep non-linear 3DMMs [Tran et al., 2019, Zhou et al., 2019] that can capture higher-frequency details. Our method also enables to enhance a face prediction through the estimation of more accurate normals.

Normal Estimation with Shape from Shading Shape-from-shading (SfS) [Horn and Brooks, 1989, Zhang et al., 1999] is a well-studied technique that aims at recovering detailed 3D surfaces from a single image based on shading cues. It estimates surface normals using the image irradiance equation, as well as illumination model parameters when these are unknown. SfS is inherently limited by the simplified image formation model assumed but has inspired numerous works that build on the correlation between pixel intensity and normals, either explicitly or implicitly. For instance, a few works on faces combined SfS with a data-driven model, *e.g.* [Smith and Hancock, 2006, Kemelmacher-Shlizerman and Basri, 2010, Snape and Zafeiriou, 2014], which helps to avoid some of the limitations such as ill-posedeness and ambiguities *e.g.* [Belhumeur et al., 1999]. The recent works of Shu et al. [2017] and Sengupta et al. [2018] use deep neural networks to decompose in-the-wild facial images into surface normals, albedo and shading, assuming Lambertian reflectance and using a semi-supervised learning approach inspired by SfS. Our work follows a similar direction and estimates the normal information from a single image, but unlike Shu et al. [2017] and Sengupta et al. [2018] we do not rely on an image formation model and let instead the network learn such transformation from real data.

Normal Estimation with Deep Networks Closely related to our work are methods that recover surface normals from an image using deep neural networks, *e.g.* [Wang et al., 2015, Eigen and Fergus, 2015, Yoon et al., 2016, Kokkinos, 2017, Bednarik et al., 2018, Qi et al., 2018, Zhang and Funkhouser, 2018, Qiu et al., 2019, Du et al., 2019, Smith et al., 2019, Alldieck et al., 2019]. Yoon et al. [2016] and Bansal et al. [2016] focus on the normal prediction task in order to recover detailed surfaces. Eigen and Fergus [2015] simultaneously regress depth, normal and semantic segmentation using a multi-scale approach. Zhang and Funkhouser [2018] predict surface normal and occlusion boundaries to later optimize for depth completion; a similar direction was followed by Qiu et al. [2019] for outdoor scenes. Trigeorgis et al. [2017] estimate facial normals with a supervised approach trained on synthetic data. Our approach differs from the aforementioned methods with a new architecture that enables cross-modal learning, hence improving performances in monocular 3D face normal estimation.

Geometry Enhancement using Deep Networks Methods have been proposed

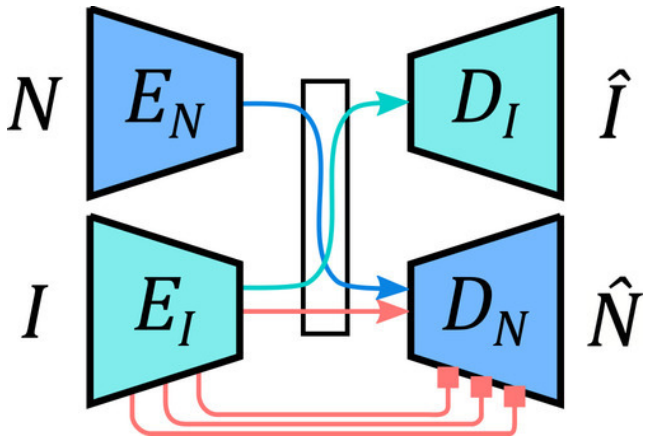


Figure 6.2 – Overview of the proposed approach. Our cross-modal architecture allows exploitation of paired and unpaired image/normal data for image-to-normal translation (red), by means of further image-to-image (green) and normal-to-normal (blue) regularizations during training. The *deactivable skip connections* allow to transfer details from the image encoder E_I to the normal decoder D_N without having to link the normal encoder E_N to the normal decoder D_N.

that directly enhance face models using deep neural networks. Richardson et al. [2017] use two networks where the first estimates a coarse shape and the second one refines the depth map from the previous branch, using an SfS-inspired unsupervised loss function. Sela et al. [2017] recover the depth and correspondence maps coupled with an off-line refinement step. The works of Yamaguchi et al. [2018], Huynh et al. [2018] estimate high frequency details by training with very accurate ground-truth data, which requires a careful acquisition process and high-quality inputs. Tran et al. [2018] estimate a per-pixel bump map, where the ground-truth data is obtained by applying an SfS method offline. The work of Chen et al. [2019] learns to estimate a geometric proxy and a displacement map for details primarily for high resolution images (2048×2048). While they mention limitations with low resolution images, we show results with resolutions as low as 256×256.

6.2 Method

We propose to predict face normals from a single color image using a deep convolutional encoder-decoder network. A natural solution to this purpose is to combine an image encoder E_I with a normal decoder D_N, as in *e.g.* Trigeorgis et al. [2017]. However training such an architecture requires pairs of normal and color images in correspondence. Although a few public datasets are available that contain high-quality 3D or normal ground-truth information for faces, for instance ICT-3DRFE [Stratou et al., 2011] or Photoface [Zafeiriou et al., 2011], they were obtained under controlled conditions and do not, therefore, really

cover the distribution of images in-the-wild. On the other hand, numerous large datasets of natural images are publicly available, for example CelebA [Liu et al., 2015] and AffectNet [Mollahosseini et al., 2017], yet without the associated accurate and detailed ground-truth normal values. Whereas other works have approached this by augmenting the training corpus with synthetic ground-truth [Trigeorgis et al., 2017, Sengupta et al., 2018], we propose instead a method based on cross-modal learning that can leverage all available data, even unpaired.

6.2.1 Cross-modal Architecture

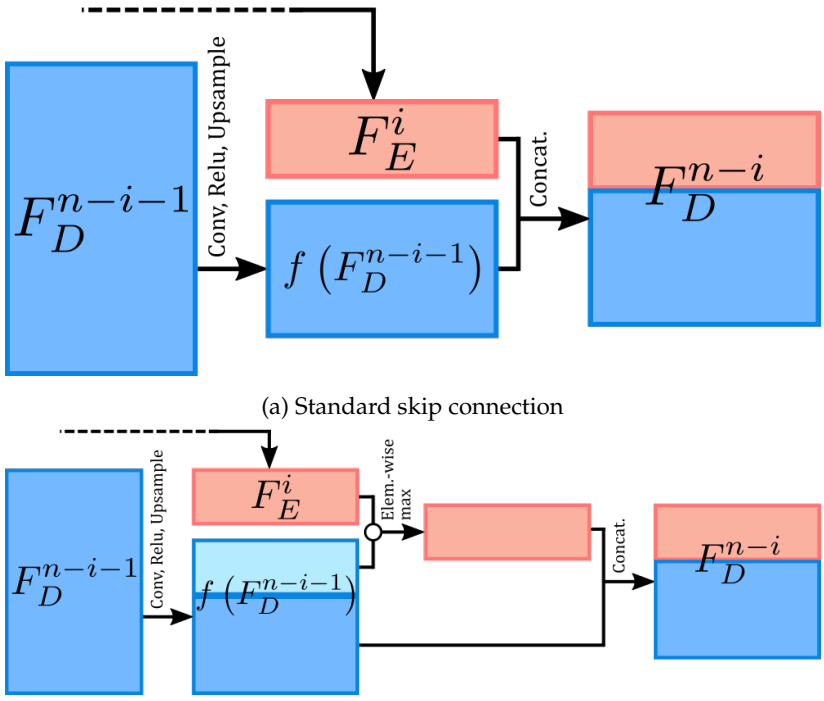
As depicted in Figure 6.2, we use two encoder/decoder networks, one for images E_I/D_I and one for normals E_N/D_N, sharing the same latent space. This architecture is trained with image-to-image, normal-to-normal and image-to-normal supervision simultaneously in order to obtain a robust and rich latent representation. To this purpose, we exploit paired images of normal and color information on faces, as available from [Stratou et al., 2011, Zafeiriou et al., 2011], in addition to individual images of either color or normal information, from *e.g.*CelebA-HQ [Karras et al., 2017] and BJUT-3D [bju, 2005]. To improve the overall performance we augment this architecture with long skip connections between E_I and D_N, as it favors the transfer of details between the image and normal domains, and since it has been shown to significantly increase performance in several image translation tasks *e.g.* Isola et al. [2017]. In practice we use a U-Net+ResNet [Ronneberger et al., 2015, He et al., 2016] architecture that combines the benefits of both short and long skip connections.

Training such an architecture end-to-end raises an obstacle: the skip connections from E_I to D_N ($E_I \rightarrow D_N$), which are based on concatenating feature maps, impose by construction to also have skip connections between the encoder and decoder of the normal modality, i.e. $E_N \rightarrow D_N$. This is counterproductive in practice: by setting skip connections within the same modality, it is in fact easier for the normal autoencoder to transfer features from the earliest layers of its encoder to the last layers of its decoder through the skip connection, thus depriving the deeper layers of any meaningful gradients during training. Not only will this fail to improve the latent face representation, but it will also alter the coefficients of the normal decoder for the image-to-normal inference task.

For this reason, we introduce the *deactivable skip connections* as shown in Figure 6.3 and detailed in Section 6.2.2. This allows us to train the framework end-to-end by setting long connections solely between E_I and D_N, thus learning a rich latent space that encodes facial features from both color and normal images while profiting from all available data.

6.2.2 Deactivable Skip Connections

As mentioned earlier, skip connections are well suited to our problem as they allow sharing of low-level information at multiple scales while still preserving the general structure. In the implementation of standard skip con-



(a) Standard skip connection

(b) Deactivable skip connection

Figure 6.3 – Instead of concatenating the encoder features (red) and decoder features (blue), as with standard skip connections, we fuse the encoder features with part of the decoder features (light blue), to be able to deactivate this operation when needed.

nections, as in Ronneberger et al. [2015], Isola et al. [2017], the decoder features at the $(n-i)^{th}$ layer F_D^{n-i} are the concatenation of the processed previous layer features $f(F_D^{n-i-1})$ and the encoder features at layer i, F_E^i, where n is the total number of layers (see Figure 6.3a).

Let $m(F_{E_I}^i)$ be the number of feature maps at the i^{th} layer of E_I. The proposed architecture (Figure 6.2) requires to set connections from the image encoder E_I to the normal decoder D_N, and as a consequence, each layer features $F_{D_N}^{n-i}$ of D_N are expected to always have an additional $m(F_{E_I}^i)$ channels. In order to gain generalization over each domain, both the color and the normal images can be auto-encoded during training. However, since the concatenation is expected during training on the decoder D_N side, features of the normal encoder E_N must be concatenated as well, which as discussed is detrimental to our model.

The **deactivable skip connections** are designed such that, during training, the transfer of feature maps from encoders to decoders can be selectively activated or deactivated. Compared to a decoder equipped with standard skip connections, the processed features $f(F_D^{n-i-1})$ of our decoder include $m(F_E^i)$ extra channels (light blue in Figure 6.3b). During a normal-to-normal pass, the skip connections are deactivated and the $(n-i)^{th}$ layer features of the

normal decoder correspond to the processed previous layer features *e.g.* $F_D^{n-i} = f(F_D^{n-i-1})$. During an image-to-normal pass, the skip connection is activated: we first perform an element-wise max-pooling between the i^{th} layer features of the encoder F_E^i and the last $m(F_E^i)$ channels of the processed $(n-i-1)^{th}$ layer features of the decoder $f(F_D^{n-i-1})$, as illustrated in Figure 6.3b. The result is stacked back with the remaining of the processed previous layer features thus forming the final $(n-i)^{th}$ decoder layer features F_D^{n-i}. Doing so allows to transfer the information from encoder to decoder without degrading performances when the transfer operation does not occur, as when auto-encoding normals.

6.2.3 Training

We train the framework end-to-end using both supervised and unsupervised data, where the latter includes individual image and normal datasets. During training, the skip connections are deactivated when doing a normal-to-normal pass. For the supervised case, and for unsupervised normals, the loss function is the cosine distance between the output and the ground-truth, which in our experiments gave better results than the $L1/L2$ norm:

$$\mathcal{L}_{nrm}(N,\hat{N}) = 1 - \frac{1}{|N|}\sum_{(i,j)} \frac{N(i,j)^\top \cdot \hat{N}(i,j)}{\|N(i,j)\|_2 \|\hat{N}(i,j)\|_2}, \tag{6.1}$$

where $N(i,j)$ and $\hat{N}(i,j)$ are the normal vectors at pixel (i,j) in the ground-truth and output normal images N and $\hat{N}$ respectively, and $|N|$ is the number of pixels in N. For unsupervised image data we use the $L2$ loss:

$$\mathcal{L}_{img}(I,\hat{I}) = \|I - \hat{I}\|_2^2, \tag{6.2}$$

where $\hat{I}$ is the output color image and I the ground-truth. In both cases, the loss is applied only on facial regions segmented using masks obtained as explained in Section 6.3.1.

In practice, as we can only perform a training iteration for one input modality at a time, either an input batch of images or normals, we train our model as follows: when loading a batch of images with image/normal ground-truth pairs, we perform a normal-to-normal iteration first, followed by an image-to-normal plus image-to-image iteration, where both losses in the latter iteration are summed with equal weights. When loading a batch of images only, we perform an image-to-image iteration. Finally, with a batch of normals only, we naturally proceed with a normal-to-normal iteration alone.

6.3 Evaluation

We report below on the accuracy of the normals estimated with our approach on standard datasets [Zafeiriou et al., 2011, Bagdanov et al., 2011]. We compare against state-of-the-art methods on normal estimation and 3D reconstruction, and show significant improvements in terms of normal prediction accuracy. This is supported by compelling reconstructions of images

in-the-wild from 300-W [Sagonas et al., 2013], as can be seen in Figures 6.4 and 6.5.

Following previous work [Sengupta et al., 2018, Trigeorgis et al., 2017], we use as metric the mean angular error between the output and the ground-truth normals, as well as percentage of pixels within the facial region with an angular error of less than 20°, 25° and 30°. For qualitative comparisons we show both the output normal map, as well as the mesh results obtained by enhancing the output of PRN [Feng et al., 2018] using normal mapping [Cohen et al., 1998]: we append the predicted normals to the the PRN mesh thus rendering enhanced geometric shading.

6.3.1 Implementation Details

The framework was implemented in PyTorch [Paszke et al., 2019], and all experiments were run on a GTX TITAN Black. The networks were trained for 40 epochs using ADAM solver [Kingma and Ba, 2015] with a learning rate of 10^{-4}. We use a ResNet-18 [He et al., 2016] architecture and set five skip connections, one at the output of the initial layer and the rest at the output of each of the four residual blocks. Each mini-batch during training consists of data of the same type, *i.e.*images only, normals only or image-normal pairs, as this worked best for us empirically.

Similar to prior work, input images are crops of fixed size around the face. We extract 2D keypoints with a face detector [King, 2009] and create masks on the facial region by finding the tightest square of edge size l around the convex hull of the points. The images are then cropped with a square patch of size $1.2 \times l$ centered at the same 2D location as the previously detected box, and subsequently resized to 256×256.

6.3.2 Datasets

Our training set comprises multiple datasets: ICT-3DRFE [Stratou et al., 2011] and Photoface [Zafeiriou et al., 2011] which provide image/normal pairs, CelebA-HQ [Karras et al., 2017] which only contains 2D images, and BJUT-3D [bju, 2005], which consists of high-quality 3D scans.

We generated 8625 image/normal pairs from ICT-3DRFE by randomly rotating the 345 3D models and relighting them using the provided albedos. We sampled random rotation axes and angles in $[-\pi/4, \pi/4]$, random lighting directions with positive z, and random intensities in $[0, 2]$. For Photoface, following the setting in [Trigeorgis et al., 2017, Sengupta et al., 2018], we randomly selected a training subset of 353 people resulting in 9478 image/normal pairs. We also generated 5000 high resolution facial images from CelebA-HQ, which is used to train the image-to-image branch exclusively. In addition, we render 3000 normal images from the 500 scans of BJUT-3D, rotated with random axes and angles in $[-\pi/4, \pi/4]$. We only render normal images from this dataset as the original scan color images are not provided.

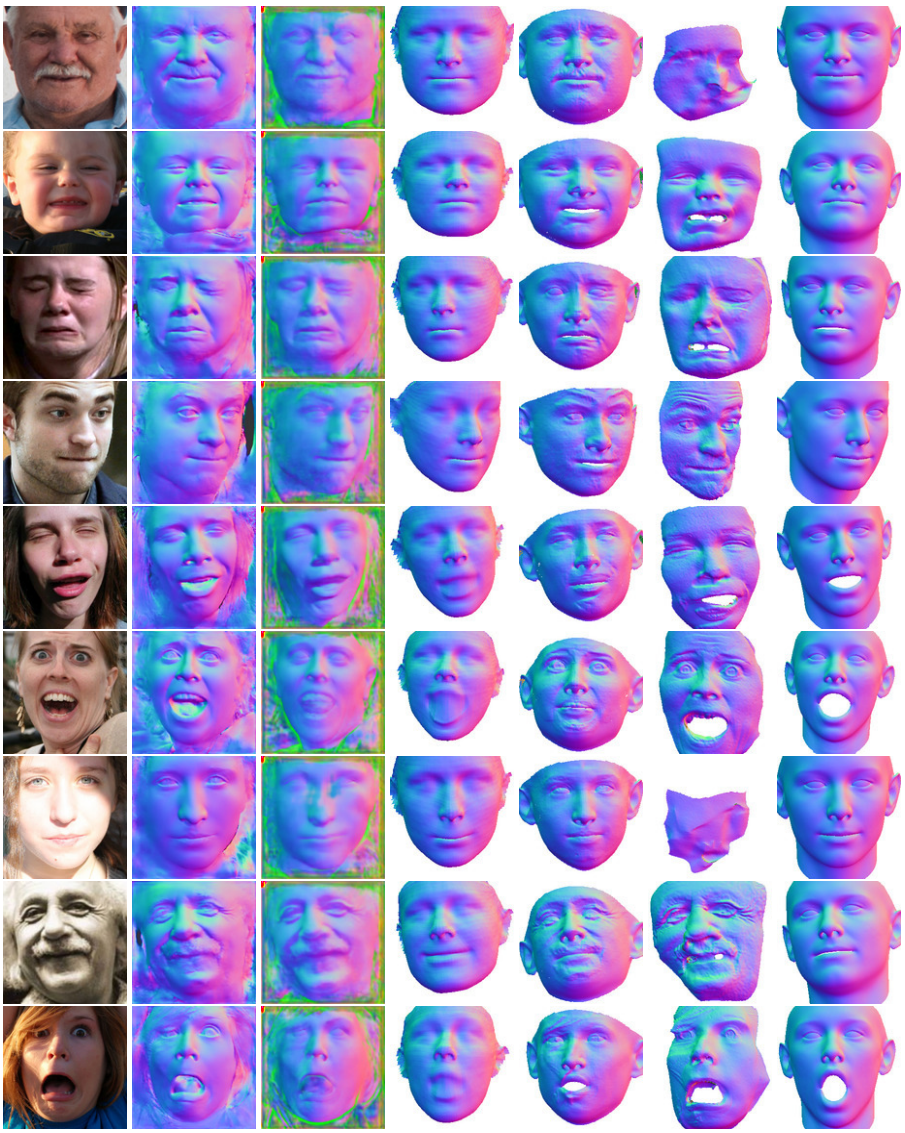
For evaluation purposes we use the remaining testing subset of Photoface, which consists of 100 subjects not seen during training and 1489 image/normal pairs. This subset challenges the reconstruction with very severe lighting conditions. Following the work of Feng et al. [2018], we create an additional evaluation set by rendering 530 color and normal facial images from the 53 3D models of the Florence dataset [Bagdanov et al., 2011], rotated with random axes and angles in $[-\pi/4, \pi/4]$. This allows to evaluate on a completely unseen dataset. Finally, we use the 300-W dataset [Sagonas et al., 2013] of 2D face images to assess qualitative performances in-the-wild. Note that for both training and testing, we limited ourselves to 3D face datasets of high quality and details.

	Mean±std	$< 20^o$	$< 25^o$	$< 30^o$
Pix2Vertex [Sela et al., 2017]	33.9±5.6	24.8%	36.1%	47.6%
Extreme [Tran et al., 2018]	27.0±6.4	37.8%	51.9%	64.5%
3DMM [Trigeorgis et al., 2017]	26.3±10.2	4.3%	56.1%	**89.4**%
3DDFA [Zhu et al., 2017]	26.0±7.2	40.6%	54.6%	66.4%
SfSNet [Sengupta et al., 2018]	25.5±9.3	43.6%	57.5%	68.7%
PRN [Feng et al., 2018]	24.8±6.8	43.1%	57.4%	69.4%
Ours	**22.8±6.5**	**49.0%**	**62.9%**	74.1%
UberNet [Kokkinos, 2017]	29.1±11.5	30.8%	36.5%	55.2%
NiW [Trigeorgis et al., 2017]	22.0±6.3	36.6%	59.8%	79.6%
Marr Rev [Bansal et al., 2016]	28.3±10.1	31.8%	36.5%	44.4%
SfSNet-ft [Sengupta et al., 2018]	12.8±5.4	83.7%	90.8%	94.5%
Ours-ft	**12.0±5.3**	**85.2%**	**92.0%**	**95.6%**

Table 6.1 – Quantitative comparisons on the Photoface dataset with mean angular errors (degrees) and percentage of errors below 20°, 25° and 30°. `-ft` means that the method was fine-tuned on Photoface.

	Mean±std	$< 20^o$	$< 25^o$	$< 30^o$
Extreme [Tran et al., 2018]	19.2±2.2	64.7%	75.9%	83.3%
SfSNet [Sengupta et al., 2018]	18.7±3.2	63.1%	77.2%	86.7%
3DDFA [Zhu et al., 2017]	14.3±2.3	79.7%	87.3%	91.8%
PRN [Feng et al., 2018]	14.1±2.16	79.9%	88.2%	92.9%
Ours	**11.3±1.5**	**89.3%**	**94.6%**	**96.9%**

Table 6.2 – Quantitative comparisons on the Florence dataset with mean angular errors (degrees) and percentage of errors below 20°, 25° and 30°.



(a) Input (b) Ours (c) SfSnet (d) PRN (e) Extreme (f) Pix2V (g) 3DDFA

Figure 6.4 – Qualitative comparisons on normals in the 300-W dataset

6.3.3 Comparisons

We compare our results to methods that explicitly recover surface normals, either for facial images (SfSNet [Sengupta et al., 2018], NiW [Trigeorgis et al., 2017]) or for general scenes (Marr Rev [Bansal et al., 2016], UberNet [Kokkinos, 2017]). We also compare against state-of-the-art approaches for 3D face reconstruction, namely the classic 3DMM fitting method used in Trigeorgis et al. [2017], 3DDFA [Zhu et al., 2017], the bump map regression based approach of Tran et al. [2018] and the combined regression+shape-from-shading approach

Figure 6.5 – Qualitative comparisons on geometries in the 300-W dataset. (a) Input, (b) Ours+PRN, (c) SfSNet+PRN, (d) PRN, (e) Extreme, (f) Pix2Vertex, (g) 3DDFA

of Sela et al. [2017].

Quantitative results can be found in Table 6.1 for Photoface and Table 6.2 for Florence datasets. We show results of our method both with (`Ours-ft`) and without (`Ours`) fine-tuning of the training split of Photoface in the upper and lower parts of Table 6.1 respectively. The same is done with SfSNet. The error values on Photoface for the methods of Sengupta et al. [2018], Trigeorgis et al. [2017], Sela et al. [2017], Bansal et al. [2016] and Kokkinos [2017] are as reported in Sengupta et al. [2018], and we use the publicly available implementations of Tran et al. [2018], Zhu et al. [2017] and Feng et al. [2018] for the others. For the Florence dataset we use the publicly available implementations. Note that, to be able to evaluate the per-pixel normal accuracy, we can only compare to 3D reconstruction methods whose output is aligned with the image. For a fair comparison, all methods were given facial images of size 256×256 as input, resized if necessary.

The proposed approach shows the best values both in mean angular error and percentage under 20°, 25° and 30° degrees, only outperformed by 3DMM on errors under 30°. As noted by the authors in Trigeorgis et al. [2017], 3DMM fitting performs well under 30° because of the coarseness of the model and the

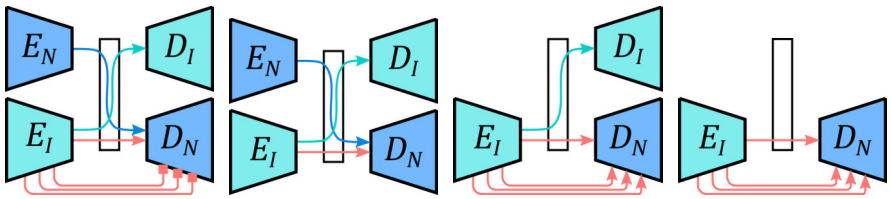
keypoint supervision, but its performance on tighter angles drops drastically as it lacks precision. We found that, although Sela et al. [2017] and Tran et al. [2018] usually provide seemingly detailed reconstructions, the actual normals of these methods lack accuracy as witnessed by their numbers.

Our good performance is also confirmed by qualitative comparisons over images in-the-wild in various head poses and under arbitrary lighting conditions as can be seen in Figures 6.4 and 6.5. For comparisons with mesh results (Figure 6.5), we show for both our approach and SfSNet [Sengupta et al., 2018] the normal mapping over the same base mesh, obtained using PRN [Feng et al., 2018], and we refer to these as Ours+PRN and SfSNet+PRN respectively. We show our meshes from two views to illustrate that the output is not optimized for a particular viewpoint, a known limitation with SfS. Compared to SfSNet we recover much more refined details that significantly enhance the base mesh. Compared to Extreme [Tran et al., 2018] our approach does not include unnecessary additional noise. As observed by other authors, Pix2Vertex [Sela et al., 2017] cannot handle difficult poses or illuminations, and sometimes simply fails to converge. Both PRN and 3DDFA [Zhu et al., 2017] can correctly recover the general structure of the face, although their goal was not to recover surface details as we do.

We believe our improved results are due to the fact that we do not rely on a parametric model for training data generation, as was done in *e.g.* Sengupta et al. [2018], as well as the strongly regularized latent space that is learned through the two encoder/decoder networks, in addition to the skip connections that can transfer the necessary details.

6.3.4 Ablation

We evaluate here the influence of the proposed architectural components. In particular, we compare against the alternatives shown in Figure 6.6: our model without skip connections (Figure 6.6b), without the normal encoder E_N (Figure 6.6c), and without both the normal encoder E_N and image decoder D_I (Figure 6.6d), *i.e.*a basic encoder-decoder architecture. Since there is no need in the last two cases for deactivable skip connections we use standard ones. We show quantitative results in Table 6.7, and qualitative examples in Figure 6.8.



(a) Ours (b) w/o skip co. (c) w/o E_N (d) w/o E_N, D_I

Figure 6.6 – Architectures for the ablation test: (a) our proposed architecture, (b) without skip connections, (c) without the normal encoder and (d) without the normal encoder and the image decoder.

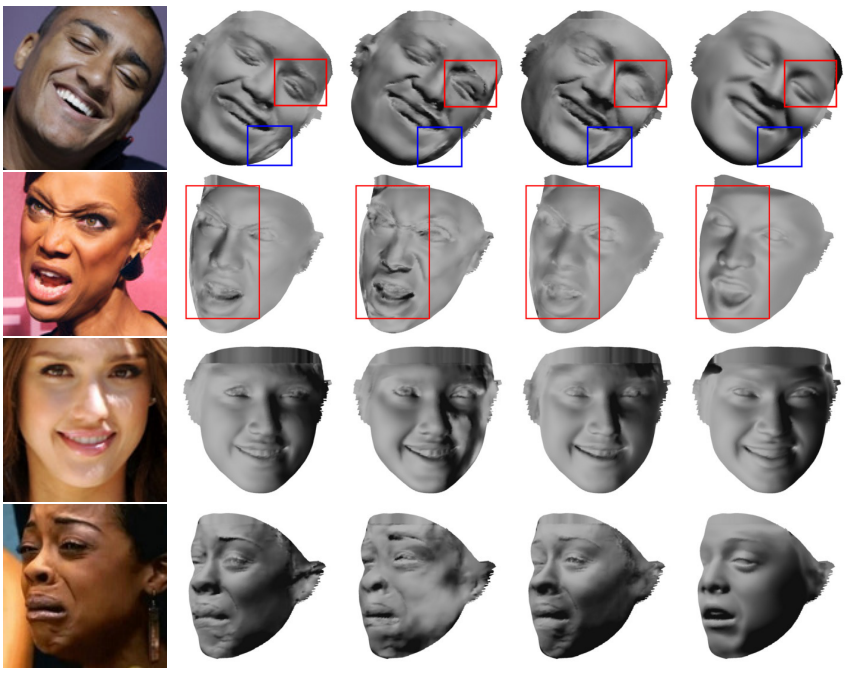
	Mean±std	$< 20^o$	$< 25^o$	$< 30^o$
w/o skip co. (Fig.6.6b)	24.4 ± 6.7	46.6%	60.6%	72.0%
w/o E_N, D_I (Fig.6.6d)	23.3 ± 6.3	47.7%	61.9%	73.3%
w/o E_N (Fig.6.6c)	23.0 ± 6.8	47.6%	61.5%	73.1%
Ours (Fig.6.6a)	$\mathbf{22.8 \pm 6.5}$	**49.0%**	**62.9%**	**74.1%**

(a) On Photoface [Zafeiriou et al., 2011]

	Mean±std	$< 20^o$	$< 25^o$	$< 30^o$
w/o skip co. (Fig.6.6b)	12.6 ± 1.4	85.8%	92.6%	95.8%
w/o E_N (Fig.6.6c)	12.4 ± 1.6	86.0%	92.6%	95.9%
w/o E_N, D_I (Fig.6.6d)	12.0 ± 1.2	87.8%	94.1%	96.7%
Ours (Fig.6.6a)	$\mathbf{11.3 \pm 1.5}$	**89.3%**	**94.6%**	**96.9%**

(b) On Florence [Bagdanov et al., 2011]

Figure 6.7 – Quantitative comparisons between architectures: the proposed architecture (*Ours*), without skip connections (*w/o skip co.*), without the normal encoder (*w/o* E_N) and without the normal encoder and the image decoder (*w/o* E_N, D_I).



(a) Input (b) Ours (c) w/o E_N D_I (d) w/o E_N (e) w/o skip co.

Figure 6.8 – Qualitative comparisons between architectures: (b) our proposed architecture, (c) without the normal encoder and the image decoder, (d) without the normal encoder, and (e) without skip connections.

Our final model outperforms the alternatives both quantitatively and quali-

tatively which validates the proposed cross-modal architecture design, and the benefit of the introduced deactivable skip connections.

For example, we can see in the geometric shape of the eyelids in the first row of Figure 6.8 and the shading in the second row that our final model gets the best from each of the alternatives. Our correct global shape estimate is comparable to that of the cross-modal model without skip connections, although the latter is smoother and clearly lacks details. Additionally we can see that removing the image decoder D_I and normal encoder E_N (*i.e.*a standard encoder-decoder with skip connections) gives poor results for images in-the-wild, due to the domain gap between training and evaluation. This can be visualized particularly in the artifacts appearing on the third and fourth examples, or the inaccurate shadings of the second example. Finally, our fine details are comparable to those of the model with skip connections but without the normal encoder E_N, which in turn has a reduced ability to represent the shape accurately, since it has not learned an additional prior on the geometric aspects of the face.

6.3.5 Low-cost depth enhancement

We can use our model to enhance the appearance of the noisy depth data coming from low-cost RGB-D sensors, e.g. Kinect. We show an example of this using the FaceWarehouse dataset [Cao et al., 2013], where we use the accompanying RGB image to predict normals with our method, and append these normals to the raw depth image pixel-wise using normal mapping, thus rendering enhanced geometric shading. In Figure 6.9 we show the RGB images in the first row, the raw depth in the second, and the same depth enhanced with our model's predictions in the last one. The ability to recover accurate normals allows to enhance the depth appearance significantly.

6.3.6 Limitations

The proposed method still has limitations, some of which are shown in Figure 6.10. These belong to extreme situations that represent outliers to the training data, including faces in very severe lighting/shades (Figure 6.10a,6.10b), occlusion (Figures 6.10c,6.10d), very low quality images (Figure 6.10e) and unusual facial textures (Figure 6.10f).

6.4 Conclusion

This chapter presented a novel deep learning based approach for the estimation of facial normals in-the-wild. The proposed method is centered on a new architecture that combines the robustness of cross-modal learning and the detail transfer ability of skip connections, enabled thanks to the new *deactivable skip connections*. By leveraging both paired and unpaired data of image and normal modalities during training, we learn a strong prior knowledge on the distribution of both natural images and facial shape in the form of surface

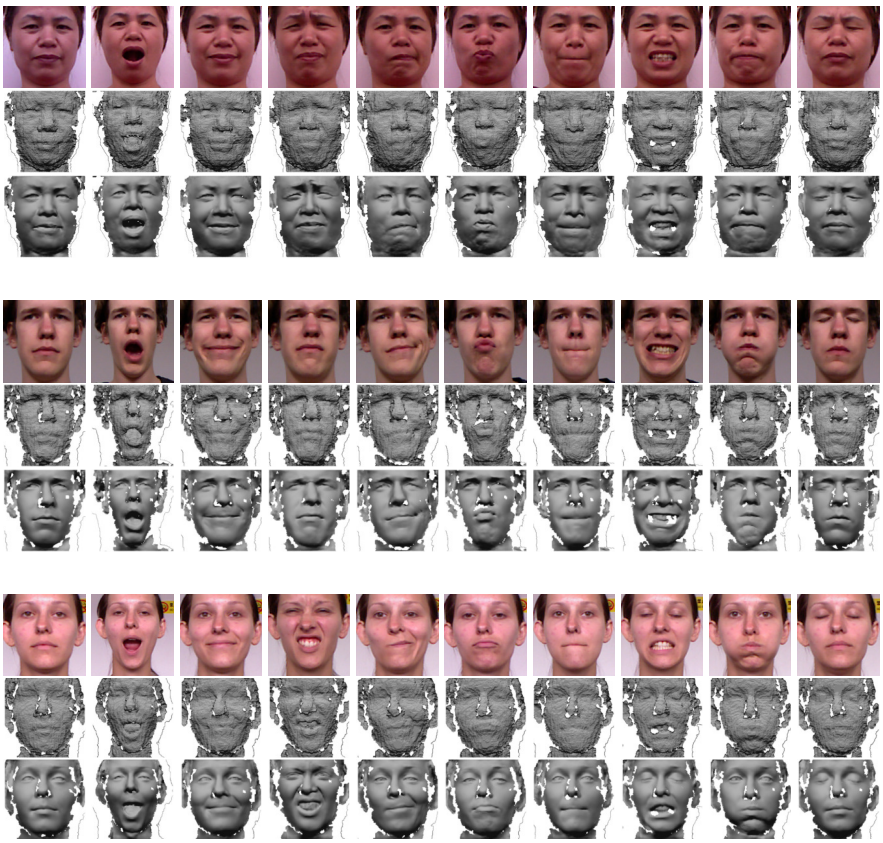


Figure 6.9 – Raw Kinect depth enhancement using our normals on the Facewarehouse dataset [Cao et al., 2013].

(a) (b) (c) (d) (e) (f)

Figure 6.10 – Failure cases

normals. Thanks to this, we achieve state-of-the-art results on angular estimation errors and obtain visually compelling enhanced 3D reconstructions on challenging images in-the-wild.

Compared to classic SfS approaches, we achieve accurate estimations even under hard conditions imposed by natural images; and we can further do

it efficiently thanks to the use of a neural network. Compared to methods based on parametric approaches, we are able to recover much finer details as a result of the skip connections and our novel cross-modal architecture that can leverage all available data. Through ablation studies and comparisons to other approaches, this chapter also confirms that (1) normal estimation is a task well suited for convolutional networks, and (2) training exclusively on real images and high-quality scans is highly benefitial, which was again allowed by the use of cross-modal learning and the deactivable skip connections.

Among the limitations of our work are the inability to properly handle occlusions (as it is mostly a local method) and to recover finer-details, *e.g.*pore-level details, which are directions that will be tackled in future work. Moreover, unlike parametric models our results are not temporally coherent, and thus cannot be used to study motion-related aspects without a registration step. We believe however that this framework can be leveraged to enhance generative models such as the ones presented in previous chapters. An interesting future direction would be to harness this in order to learn fine-scale details that correlate with both identity and expression. Considering that the proposed architecture is rather generic, a final future direction will be to investigate its use on other tasks that exhibit similar data conditions.

7 Conclusions

This thesis presented novel methods for learning data-driven models of the 3D facial shape from large-scale datasets. There are many aspects of the face geometry that are interesting and challenging to explore. We focused here on the following aspects: (1) how to build *decoupled* models that can capture the interaction between the shape and motion components; (2) how to place large datasets of 3D faces in motion into a single parameterization; and (3) how to obtain finer details by focusing on the problem of surface normal recovery from natural images. In each of these methods we investigated whether better performances could be obtained by working on large-scale datasets. We circumvented the difficulty of acquiring a large number of 3D scans by profiting from the numerous publicly available sources, and proposed techniques designed to handle the challenges that come with such data while harnessing the underlying information.

To conclude this work we summarize next the main contributions of this thesis, as well as directions for future work.

7.1 Summary of Contributions

The main contributions of this thesis were the following:

Chapter 3 proposed the *multilinear autoencoder*, a novel method for building multilinear models that does not require complete data tensors for training. The approach was based on a new architecture that enabled the use of the deep learning optimization machinery to refine an initial tensor model. This allowed to better encode all available training data, thus demonstrating that expressive multilinear models can be learned from large-scale sources. Additionally, we proposed a loss function on the latent space which allowed to retain the decoupling capabilities of multilinear models, and even improve them when compared to competing methods.

Chapter 4 contributed a *spatiotemporal registration approach* for 3D faces in motion, designed to automatically process datasets coming from multiple acquisition systems. Through the combination of a spatiotemporal model that globally handles entire motion sequences, and a regression-based approach that can efficiently and robustly initialize the registration process, we demonstrated accurate performances while remaining both efficient and scalable. The method was tested on three publicly available datasets showing different types of

motion including spontaneous ones, and combined allowed us to register more than $300,000$ facial frames.

Chapter 5 explored the use of *adversarial learning for decoupled models*, contributing an alternative technique to linear and multilinear models that achieved significantly better decoupling results. We enabled the use of adversarial learning for 3D faces by proposing the *geometry mapping layer*, which acts as bridge between the 3D generator and 2D discriminators. This strategy allowed to leverage recent progress in generative adversarial learning while still generating three-dimensional faces in their natural domain. Our purely discriminative loss functions demonstrated significantly better decoupling capabilities, and were able to capture subtle expression differences in the latent space.

Since there is no standard evaluation protocol, both Chapters 3 and 5 further proposed metrics to assess the degree of decoupling of generative models based on external classifiers. Chapter 5 also proposed a diversity metric, which was not considered before in the evaluation of 3D face models.

Finally, Chapter 6 introduced a new approach for the problem of estimating *facial normals from images in-the-wild*. Improved performances were obtained through the use of a novel cross-modal learning technique that enabled training exclusively from high quality data, whether paired or not. This was achieved by a novel module that we called *deactivable skip connections*, which allowed to integrate both the auto-encoded and image-to-normal branches within a same architecture, while still transferring the local details from the input image to the output surface. We showed how this strategy can learn a rich latent space of both natural images and surface normals that enabled accurate reconstructions, as well as state-of-the-art results and visually compelling enhancements in challenging cases.

Each of these methods have a few drawbacks that were discussed by the end of each chapter. We summarize here what we consider to be the main points:

- For methods that learn a global latent space such as those in Chapters 3 and 5, it remains a question what is the optimal dimension of each space, and how to properly choose these dimensions. This was also a problem in the spatiotemporal model used in Chapter 4, where a low number of temporal coefficients can lead to overly-smoothed motions. The problem is shared by many of the recent deep-learning based techniques that operate as "black-box" machines. A principled way of choosing the optimal dimensionality such that it balances model expressiveness against compactness is certainly desirable, particularly for telecommunication systems.

- The registration method in Chapter 4 globally considers the entire facial surface and motion, but a more local treatment of certain attributes

of the face might be worth exploring. This is valid both in terms of space and time. In terms of time, some motions occur faster than others, and a unique temporal space for the entire sequence might not be able to capture higher frequencies. A more local approach, *e.g.*a sliding window with variable dimensionality, could be necessary for accurate results in different types of motion. In terms of space, areas such as the mouth and eyes are key for transmitting emotions, and a faithful recovery of their shape is essential for applications such as animation and re-targeting. Specific optimization terms for these areas, as it was considered in *e.g.* Bermano et al. [2015], Garrido et al. [2016a], could further improve the results of Chapter 4.

- The approach for surface normal estimation in Chapter 6 does not yield temporally coherent results, and thus it cannot be used to study how finer details evolve over time. Furthermore, because of the local formulation the results are not robust to occlussions. To address this, a combination of global models (like those considered in previous chapters) and local approaches such as the one presented here should be considered.

More general directions for future work are discussed in the next section.

7.2 Future work

The models presented here were learned from publicly available sources that allowed to consider datasets of larger scale than most related work. Yet, these are still expensive and time consuming to capture, and can only be acquired under controlled setups which limit the range of motions that can be studied. Learning high-quality models from cheaper acquisition devices such as RGB or RGB-D sensors would not only give access to a larger corpus of training data, but will also allow to model human behaviour that cannot be recorded in controlled scenarios. In the case of faces, this means modeling truly spontaneous expressions and micro-expressions, as well as their relationship with the environment. First works that learn 3D models purely from RGB images have begun to appear, *e.g.* Tran and Liu [2018], Tran et al. [2019], but are still limited in the amount of variations they can capture. Extending the directions considered in this thesis such that equivalent or improved results can be obtained using less constrained data is an exciting avenue for future work.

For decoupled models there are two interesting extensions that were not addressed here. First is the *unsupervised* discovery of the latent factors: instead of training with labeled data as considered in this thesis, it would be benefitial to explore the case were the labels corresponding to each of the factors are unknown. This would allow for example to build the identity-expression-viseme model of Figure 5.6 without the need to provide any manual annotation,

which is very time consuming and prone to errors. While numerous work have explored this with 2D applications in mind (*e.g.* Chen et al. [2016]), the direction is mostly unexplored in the case of 3D data.

Another extension is the fine-grained encoding of semantics in the latent space. The decoupling considered here is concerned with the independent manipulation of identity and expression, but there is no mechanism to control for example the size of the nose or how closed the eyes are. Having access to semantically-based parameters that can modify for example the intensity of an action unit (as it is commonly done in the film industry through the blendshape parameterization) would allow for widespread adoption of more complex models like the one in Chapter 5. Furthermore, such models could be use for inferring semantic information from *e.g.*2D images, which can in turn be useful for fine-grained recognition and manipulation.

The models of Chapters 3 and 5 are global models that encode the entire face, while the approach of Chapter 6 considers mostly local information. There is a trade-off for each: using global models gives robustness to different tasks by providing a strong knowledge of what a face should look like and how the different spaces interact with each other, but it is not capable of capturing details. On the other hand, the local approach of Chapter 6 allows to recover finer details [1], but the results are neither robust nor temporally coherent. A future direction in terms of modeling will be to explore a combination of these two, in order to benefit from the best of each.

Some of the technical contributions of this thesis can be applied to other problems and it would be interesting to explore their capacity for this. This is the case of the geometry mapping layer of Chapter 5 and the deactivable skip connections of Chapter 6. For example, the geometry mapping layer can be useful for general tasks on 3D shapes such as correspondence and classification, while the deactivable skip connections can be leveraged on other applications that both involve multiple modalities and can profit from the use of skip connections.

1. Note that there is still a notion of global shape, see *e.g.*Figure 6.8b.

Bibliography

The bjut-3d large-scale chinese face database. Technical report, Multimedia and Intelligent Software Technology Beijing Municipal Key Laboratory, Beijing University of Technology, Beijing, China, 2005. 90, 93

I. Akhter, T. Simon, S. Khan, I. Matthews, and Y. Sheikh. Bilinear spatiotemporal basis models. *ACM Transactions on Graphics*, 31(2):17, 2012. 50, 51, 52

T. Alashkar, B. B. Amor, M. Daoudi, and S. Berretti. A grassmann framework for 4d facial shape analysis. *Pattern Recognition*, 57:21–30, 2016. 40, 41, 45

T. Albrecht, M. Luthi, and T. Vetter. A statistical deformation prior for non-rigid image and shape registration. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 14

O. Aldrian and W. A. P. Smith. Inverse rendering of faces with a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5): 1080–1093, 2013. 15

O. Alexander, M. Rogers, W. Lambeth, J.-Y. Chiang, W.-C. Ma, C.-C. Wang, and P. Debevec. The digital emily project: Achieving a photorealistic digital actor. *IEEE Computer Graphics and Applications*, 30(4):20–31, 2010. 10, 32

T. Alldieck, G. Pons-Moll, C. Theobalt, and M. Magnor. Tex2shape: Detailed full human body geometry from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 88

B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM transactions on graphics (TOG)*, 22(3):587–594, 2003. 13

B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 12, 13, 14, 32, 33, 41, 45, 47

B. Amberg, R. Knothe, and T. Vetter. Expression invariant 3d face recognition with a morphable model. In *Conference on Automatic Face and Gesture Recognition*, pages 1–6, 2008. 17, 18, 25, 49, 65, 78, 79, 80

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 69

T. Bagautdinov, C. Wu, J. Saragih, P. Fua, and Y. Sheikh. Modeling Facial Geometry using Compositional VAEs. In *Conference on Computer Vision and Pattern Recognition*, volume 1, page 1, 2018. 17, 66, 67

A. D. Bagdanov, A. Del Bimbo, and I. Masi. The florence 2d/3d hybrid face dataset. In *ACM Workshop on Human Gesture and Behavior Understanding*, 2011. 87, 92, 94, 98

A. Bansal, B. Russell, and A. Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *CVPR*, 2016. 88, 94, 95, 96

V. Barrielle and N. Stoiber. Realtime performance-driven physical simulation for facial animation. In *Computer Graphics Forum*, volume 38, pages 151–166. Wiley Online Library, 2019. 17

V. Barrielle, N. Stoiber, and C. Cagniart. Blendforces: A dynamic framework for facial animation. In *Computer Graphics Forum*, volume 35, pages 341–352. Wiley Online Library, 2016. 17

A. Bas and W. A. Smith. What does 2d geometric information really tell us about 3d face shape? *International Journal of Computer Vision*, 127(10):1455–1473, 2019. 15

J. Bednarik, P. Fua, and M. Salzmann. Learning to reconstruct texture-less deformable surfaces from a single view. In *Proceedings of International Conference on 3D Vision*, 2018. 88

T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross. High-quality single-shot capture of facial geometry. In *ACM SIGGRAPH 2010 papers*, pages 1–9. 2010. 11

T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross. High-quality passive facial performance capture using anchor frames. *ACM Transactions on Graphics*, 30(4):75:1–75:10, 2011. 11, 14, 48

P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille. The bas-relief ambiguity. *International Journal of Computer Vision*, 1999. 88

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 68

J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975. 13

A. Bermano, T. Beeler, Y. Kozlov, D. Bradley, B. Bickel, and M. Gross. Detailed spatio-temporal reconstruction of eyelids. *ACM Transactions on Graphics (TOG)*, 34(4):1–11, 2015. 63, 105

A. H. Bermano, D. Bradley, T. Beeler, F. Zund, D. Nowrouzezahrai, I. Baran, O. Sorkine-Hornung, H. Pfister, R. W. Sumner, B. Bickel, and M. Gross. Facial performance enhancement using dynamic shape space analysis. *ACM Trans. Graph.*, 33(2):13:1–13:12, Apr. 2014. ISSN 0730-0301. 20

P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992. 13

B. Bickel, M. Lang, M. Botsch, M. A. Otaduy, and M. Gross. Pose-space animation and transfer of facial details. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 57–66. Eurographics Association, 2008. 20

G. Blais and M. D. Levine. Registering multiview range data to create 3d computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):820–824, 1995. 13

V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Siggraph*, volume 99, pages 187–194, 1999. 3, 9, 11, 12, 13, 14, 15, 23, 25, 87

V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. In *Computer graphics forum*, volume 22, pages 641–650. Wiley Online Library, 2003. 17

T. Bolkart and S. Wuhrer. 3d faces in motion: Fully automatic registration and statistical analysis. *Computer Vision and Image Understanding*, 131:100–115, 2015a. 12, 14, 19, 46, 47, 49, 50, 56, 57, 58, 59, 60, 61

T. Bolkart and S. Wuhrer. A groupwise multilinear correspondence optimization for 3d faces. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3604–3612, 2015b. 14

T. Bolkart and S. Wuhrer. A robust multilinear model learning framework for 3d faces. In *Conference on Computer Vision and Pattern Recognition*, pages 4911–4919, 2016. 19, 25, 26, 33, 37, 38

J. Booth, A. Roussos, S. Zafeiriou, A. Ponniah, and D. Dunaway. A 3d morphable model learnt from 10,000 faces. In *Conference on Computer Vision and Pattern Recognition*, pages 5543–5552, 2016. 15, 25

J. Booth, A. Roussos, A. Ponniah, D. Dunaway, and S. Zafeiriou. Large scale 3d morphable models. *International Journal of Computer Vision*, 126(2-4):233–254, 2018. 13, 87

S. Bouaziz, Y. Wang, and M. Pauly. Online modeling for realtime facial animation. *ACM Transactions on Graphics (ToG)*, 32(4):1–10, 2013. 16, 20, 47

D. Bradley, W. Heidrich, T. Popa, and A. Sheffer. High resolution passive facial performance capture. *ACM Transactions on Graphics*, 29(4):41:1–41:10, 2010. 11, 48

A. Brunton, J. Lang, E. Dubois, and C. Shu. Wavelet model-based stereo for fast, robust face reconstruction. In *2011 Canadian Conference on Computer and Robot Vision*, pages 347–354. IEEE, 2011. 15

A. Brunton, T. Bolkart, and S. Wuhrer. Multilinear wavelets: A statistical shape space for human faces. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014a. 14, 19

A. Brunton, A. Salazar, T. Bolkart, and S. Wuhrer. Review of statistical shape spaces for 3d data with comparative analysis for human faces. *Computer Vision and Image Understanding*, 128:1–17, 2014b. 10, 15, 19

C. Cao, Y. Weng, S. Lin, and K. Zhou. 3d shape regression for real-time facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013. 4, 12, 19, 23, 99, 100

C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on graphics (TOG)*, 33(4):1–10, 2014a. 11, 23

C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. *ACM Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2014b. 25

C. Cao, D. Bradley, K. Zhou, and T. Beeler. Real-time high-fidelity facial performance capture. *ACM Transactions on Graphics (ToG)*, 34(4):1–9, 2015. 20, 46, 87, 88

C. Cao, H. Wu, Y. Weng, T. Shao, and K. Zhou. Real-time facial animation with image-based dynamic avatars. *ACM Transactions on Graphics*, 35(4), 2016. 19, 23

U. Castellani and A. Bartoli. 3d shape registration. In *3D Imaging, Analysis and Applications*, pages 221–264. Springer, 2012. 13

A. Chen, Z. Chen, G. Zhang, K. Mitchell, and J. Yu. Photo-realistic facial details synthesis from single image. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 20, 89

X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2016. 68, 106

Y. Chen and G. G. Medioni. Object modeling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, 1992. 13

S. Cheng, I. Kotsia, M. Pantic, and S. Zafeiriou. 4dfab: A large scale 4d facial expression database for biometric applications. *arXiv preprint arXiv:1712.01443*, 2017a. 13, 46, 47

S. Cheng, I. Marras, S. Zafeiriou, and M. Pantic. Statistical non-rigid icp algorithm and its application to 3d face alignment. *Image and Vision Computing*, 58:3–12, 2017b. 14, 17, 49

E. S. Chuang, F. Deshpande, and C. Bregler. Facial expression space learning. In *10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings.*, pages 68–76. IEEE, 2002. 18

H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2, pages 44–51. IEEE, 2000. 13

J. Cohen, M. Olano, and D. Manocha. Appearance-preserving simplification. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, 1998. 93

M. Cong, K. S. Bhat, and R. Fedkiw. Art-directed muscle simulation for high-end facial animation. In *Symposium on Computer Animation*, pages 119–127, 2016. 17

D. Cosker, E. Krumhuber, and A. Hilton. Perception of linear and nonlinear motion properties using a facs validated 3d facial model. In *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, pages 101–108, 2010. 66

D. Cosker, E. Krumhuber, and A. Hilton. A facs valid 3d dynamic action unit database with applications to 3d dynamic morphable facial modeling. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2296–2303. IEEE, 2011. 4, 11, 12, 13, 24, 33, 45, 46, 47, 54, 56, 59, 60, 61

C. Creusot, N. Pears, and J. Austin. A machine-learning approach to keypoint detection and landmarking on 3d meshes. *International journal of computer vision*, 102(1-3):146–179, 2013. 12

H. Dai, N. Pears, W. Smith, and C. Duncan. A 3d morphable model of craniofacial shape and texture variation. In *International Conference on Computer Vision*, pages 3085–3093, 2017. 14, 15, 25

K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister. Video face replacement. *ACM Transactions on Graphics*, 30(6):#130:1–10, 2011. 19, 23, 25

C. Darwin. *The expression of the emotions in man and animals*. London :John Murray,, 1872. 4

R. Davies, C. Twining, and C. Taylor. *Statistical models of shape: Optimisation and evaluation*. Springer Science & Business Media, 2008. 10, 33, 55, 76

L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal of Matrix Analysis and Applications*, 21: 1253–1278, 2000a. 18, 25, 31, 32, 36

L. De Lathauwer, B. De Moor, and J. Vandewalle. On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors. *SIAM journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000b. 36

M. De Smet and L. Van Gool. Optimal regions for linear model-based 3d face reconstruction. In R. Kimmel, R. Klette, and A. Sugimoto, editors, *Computer Vision – ACCV 2010*, pages 276–289. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-19318-7. 15

D. Decarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 38: 99–127, 2000. 17

J. Deng, S. Cheng, N. Xue, Y. Zhou, and S. Zafeiriou. Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In *CVPR*, pages 7093–7102, 2018. 67

C. Donahue, Z. C. Lipton, A. Balsubramani, J. McAuley, S. Rezvani, N. Mokari, M. R. Javan, M. H. Salas-Olmedo, J. C. Garcia-Palomares, J. Gutierrez, et al. Semantically decomposing the latent spaces of generative adversarial networks. *International Conference on Learning Representations*, 2018. 69, 76

A. Dosovitskiy, J. Tobias Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1538–1546, 2015. 68

D. Du, L. Wang, H. Wang, K. Zhao, and G. Wu. Translate-to-recognize networks for rgb-d scene recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 88

M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH*, 1995. 75

B. Egger, S. Schönborn, A. Schneider, A. Kortylewski, A. Morel-Forster, C. Blumer, and T. Vetter. Occlusion-aware 3d morphable models and an illumination prior for face image analysis. *International Journal of Computer Vision*, 2018. 87

B. Egger, W. A. Smith, A. Tewari, S. Wuhrer, M. Zollhoefer, T. Beeler, F. Bernard, T. Bolkart, A. Kortylewski, S. Romdhani, et al. 3d morphable face models–past, present and future. *arXiv preprint arXiv:1909.01815*, 2019. 10

D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, 2015. 88

P. Ekman and W. V. Friesen. Facial action coding system: a technique for the measurement of facial movement. 1978. 16

P. Ekman and W. V. Friesen. *Unmasking the face: A guide to recognizing emotions from facial clues*. Ishk, 2003.

P. Ekman and D. Keltner. Universal facial expressions of emotion. *California Mental Health Research Digest*, 1970. 3, 4

T. C. Faltemier, K. W. Bowyer, and P. J. Flynn. Using a multi-instance enrollment representation to improve 3d face recognition. In *2007 First IEEE International Conference on Biometrics: Theory, Applications, and Systems*, pages 1–6. IEEE, 2007. 12

G. Fanelli, J. Gall, H. Romsdorfer, T. Weise, and L. Van Gool. A 3-d audio-visual corpus of affective communication. *IEEE Transactions on Multimedia*, 12(6): 591–598, 2010. 4, 12, 81

T. Fang, X. Zhao, O. Ocegueda, S. K. Shah, and I. A. Kakadiaris. 3d/4d facial expression analysis: An advanced annotated face model approach. *Image and vision Computing*, 30(10):738–749, 2012. 45, 47

Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV*, 2018. 85, 86, 87, 93, 94, 96, 97

V. Fernández Abrevaya, S. Wuhrer, and E. Boyer. Multilinear autoencoder for 3d face model learning. In *Winter Conference on Applications of Computer Vision*, pages 1–9, 2018. 54, 66, 67, 75, 78, 80

G. Fyffe, A. Jones, O. Alexander, R. Ichikari, and P. Debevec. Driving high-resolution facial scans with video performance capture. *ACM Transactions on Graphics (TOG)*, 34(1):1–14, 2014. 48

G. Fyffe, K. Nagano, L. Huynh, J. Busch, A. Jones, and H. L. H. L. Debevec. Multi-view stereo on consistent face topology. In *Eurographics*, 2017. 11, 46, 48

P. Garrido, L. Valgaerts, C. Wu, and C. Theobalt. Reconstructing detailed dynamic face geometry from monocular video. In *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2013)*, volume 32, pages 158:1–158:10, November 2013. doi: 10.1145/2508363.2508380. URL http://doi.acm.org/10.1145/2508363.2508380. 20, 21, 87

P. Garrido, M. Zollhöfer, D. Casas, L. Valgaerts, K. Varanasi, P. Pérez, and C. Theobalt. Reconstruction of personalized 3d face rigs from monocular video. *ACM Transactions on Graphics (TOG)*, 35(3):1–15, 2016a. 18, 20, 21, 46, 88, 105

P. Garrido, M. Zollhöfer, C. Wu, D. Bradley, P. Pérez, T. Beeler, and C. Theobalt. Corrective 3d reconstruction of lips from monocular video. *ACM Trans. Graph.*, 35(6):219–1, 2016b. 63

B. Gecer, A. Lattas, S. Ploumpis, J. Deng, A. Papaioannou, S. Moschoglou, and S. Zafeiriou. Synthesizing coupled 3d face modalities by trunk-branch generative adversarial networks. *arXiv preprint arXiv:1909.02215*, 2019. 67, 87

J. Geng. Structured-light 3d surface imaging: a tutorial. *Advances in Optics and Photonics*, 3(2):128–160, 2011. 11

K. Genova, F. Cole, A. Maschinot, A. Sarna, D. Vlasic, and W. T. Freeman. Unsupervised training for 3d morphable model regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8377–8386, 2018. 15, 26, 86, 87

T. Gerig, A. Morel-Forster, C. Blumer, B. Egger, M. Luthi, S. Schönborn, and T. Vetter. Morphable face models-an open framework. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 75–82. IEEE, 2018. 13, 15

A. Ghosh, G. Fyffe, B. Tunwattanapong, J. Busch, X. Yu, and P. Debevec. Multiview face capture using polarized spherical gradient illumination. In *ACM Transactions on Graphics (TOG)*, volume 30, page 129. ACM, 2011. 11

P. Ghosh, J. Song, E. Aksan, and O. Hilliges. Learning human motion models for long-term predictions. In *Conference on 3D Vision*, pages 1–9, 2017. 34

S. Z. Gilani, A. Mian, and P. Eastwood. Deep, dense and accurate 3d face correspondence for generating population specific deformable models. *Pattern Recognition*, 69:238–250, 2017. 12

A. Golovinskiy, W. Matusik, H. Pfister, S. Rusinkiewicz, and T. Funkhouser. A statistical model for synthesis of detailed facial geometry. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1025–1034. ACM, 2006. 20

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 66, 69

X. Gu, S. J. Gortler, and H. Hoppe. Geometry images. In *SIGGRAPH*, 2002. 9, 70, 71

G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010. 54

R. A. Güler, G. Trigeorgis, E. Antonakos, P. Snape, S. Zafeiriou, and I. Kokkinos. DenseReg: fully convolutional dense shape regression in-the-wild. In *Conference on Computer Vision and Pattern Recognition*, pages 6799–6808, 2017. 26

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017. 69

X. Han, C. Gao, and Y. Yu. Deepsketch2face: a deep learning based sketching system for 3d face and caricature modeling. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017. 23

M. Hansard, S. Lee, O. Choi, and R. P. Horaud. *Time-of-flight cameras: principles, methods and applications*. Springer Science & Business Media, 2012. 11

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 30, 90, 93

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *Iclr*, 2(5):6, 2017. 68

B. K. Horn and M. J. Brooks. *Shape from shading*. 1989. 11, 20, 86, 88

P.-L. Hsieh, C. Ma, J. Yu, and H. Li. Unconstrained realtime facial performance capture. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 47

X. Huang, N. Paragios, and D. Metaxas. Establishing local correspondences towards compact representations of anatomical structures. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 926–934. Springer, 2003. 13

P. Huber, G. Hu, R. Tena, P. Mortazavian, P. Koppen, W. J. Christmas, M. Ratsch, and J. Kittler. A multiresolution 3d morphable face model and fitting framework. In *Proceedings of the 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2016. 15, 87

T. J. Hutton, B. Buxton, and P. Hammond. Dense surface point distribution models of the human face. In *Proceedings IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA 2001)*, pages 153–160. IEEE, 2001. 13

L. Huynh, W. Chen, S. Saito, J. Xing, K. Nagano, A. Jones, P. Debevec, and H. Li. Mesoscopic facial geometry inference using deep neural networks. In *CVPR*, 2018. 20, 68, 89

A. E. Ichim, S. Bouaziz, and M. Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Transactions on Graphics (ToG)*, 34(4):45, 2015. 20

A.-E. Ichim, P. Kadleček, L. Kavan, and M. Pauly. Phace: physics-based face modeling and animation. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 17

P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 90, 91

A. S. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 87

A. Jacobson, D. Panozzo, et al. libigl: A simple C++ geometry processing library, 2018. https://libigl.github.io/. 52, 54

A. H. Jha, S. Anand, M. Singh, and V. Veeravasarapu. Disentangling factors of variation with cycle-consistent variational auto-encoders. In *European Conference on Computer Vision*, pages 829–845. Springer, 2018. 68

Z.-H. Jiang, Q. Wu, K. Chen, and J. Zhang. Disentangled representation learning for 3d face shape. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11957–11966, 2019. 20, 66

I. T. Jolliffe. *Principal Component Analysis and Factor Analysis*, pages 115–128. Springer New York, 1986. ISBN 978-1-4757-1904-8. doi: 10.1007/978-1-4757-1904-8_7. 14

I. A. Kakadiaris, G. Passalis, G. Toderici, M. N. Murtuza, Y. Lu, N. Karampatziakis, and T. Theoharis. Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *IEEE Transactions on pattern analysis and machine intelligence*, 29(4):640–649, 2007. 47

T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 90, 93

I. Kemelmacher-Shlizerman and R. Basri. 3d face reconstruction from a single image using a single reference face shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010. 88

H. Kim and A. Mnih. Disentangling by factorising. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018. 68

H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, M. Niessner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):163, 2018a. 18, 86, 87

H. Kim, M. Zollhöfer, A. Tewari, J. Thies, C. Richardt, and C. Theobalt. Inversefacenet: Deep monocular inverse face rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018b. 87

D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. 93

G. King and L. Zeng. Logistic regression in rare events data. *Political analysis*, 9 (2):137–163, 2001. 75

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2015. 75, 80, 93

M. Klaudiny and A. Hilton. High-detail 3d capture and non-sequential alignment of facial performance. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 17–24. IEEE, 2012. 48

O. Klehm, F. Rousselle, M. Papas, D. Bradley, C. Hery, B. Bickel, W. Jarosz, and T. Beeler. Recent advances in facial appearance capture. In *Computer Graphics Forum*, volume 34, pages 709–733. Wiley Online Library, 2015. 10

I. Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017. 88, 94, 95, 96

T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009. 18, 28, 50

Y. Kozlov, D. Bradley, M. Bächer, B. Thomaszewski, T. Beeler, and M. Gross. Enriching facial blendshape rigs with physical simulation. In *Computer Graphics Forum*, volume 36, pages 75–84. Wiley Online Library, 2017. 17

T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in neural information processing systems*, pages 2539–2547, 2015. 68

S. Laine, T. Karras, T. Aila, A. Herva, S. Saito, R. Yu, H. Li, and J. Lehtinen. Production-level facial performance capture using deep convolutional neural networks. In *Symposium on Computer Animation*, pages #10:1–10, 2017. 24, 26

A. Lattas, S. Moschoglou, B. Gecer, S. Ploumpis, V. Triantafyllou, A. Ghosh, and S. Zafeiriou. Avatarme: Realistically renderable 3d facial reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 760–769, 2020. 20

M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, et al. The digital michelangelo project: 3d scanning of large statues. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 131–144, 2000. 11

J. Lewis, Z. Mo, K. Anjyo, and T. Rhee. Probable and improbable faces. In *Mathematical Progress in Expressive Image Synthesis I*, pages 21–30. Springer, 2014a. 15

J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. H. Pighin, and Z. Deng. Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)*, 1(8):2, 2014b. 3, 9, 10, 16, 17

H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Computer graphics forum*, volume 27, pages 1421–1430. Wiley Online Library, 2008. 12, 13

H. Li, T. Weise, and M. Pauly. Example-based facial rigging. *Acm transactions on graphics (tog)*, 29(4):1–6, 2010. 16

H. Li, J. Yu, Y. Ye, and C. Bregler. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph.*, 32(4):42–1, 2013. 16, 47, 88

J. Li, W. Xu, Z. Cheng, K. Xu, and R. Klein. Lightweight wrinkle synthesis for 3d facial modeling and animation. *Computer-Aided Design*, 58:117–122, 2015. 20

R. Li, K. Bladin, Y. Zhao, C. Chinara, O. Ingraham, P. Xiang, X. Ren, P. Prasad, B. Kishore, J. Xing, et al. Learning formation of physically-based face attributes. In *CVPR*, 2020. 18

T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics*, 36(6): 194:1–194:17, 2017. 14, 15, 17, 18, 25, 33, 46, 47, 49, 56, 57, 58, 59, 60, 61, 67, 79

F. Liu, R. Zhu, D. Zeng, Q. Zhao, and X. Liu. Disentangling features in 3d face shapes for joint face reconstruction and recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5216–5225, 2018. 18, 23

Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 90

S. Lombardi, J. Saragih, T. Simon, and Y. Sheikh. Deep appearance models for face rendering. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018. 17, 66

K.-L. Low. Linear least-squares optimization for point-to-plane icp surface registration. *Chapel Hill, University of North Carolina*, 4(10):1–3, 2004. 13

M. Lüthi, T. Gerig, C. Jud, and T. Vetter. Gaussian process morphable models. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1860–1873, 2017. 14, 15

W.-C. Ma, T. Hawkins, P. Peers, C.-F. Chabert, M. Weiss, and P. E. Debevec. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. *Rendering Techniques*, 2007(9):10, 2007. 11

W.-C. Ma, A. Jones, J.-Y. Chiang, T. Hawkins, S. Frederiksen, P. Peers, M. Vukovic, M. Ouhyoung, and P. Debevec. Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Transactions on Graphics (TOG)*, 27(5):1–10, 2008. 20

M. F. Mathieu, J. J. Zhao, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in neural information processing systems*, pages 5040–5048, 2016. 68

M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, volume 2017, pages 498–502, 2017. 81

M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003. 53, 74

A. Mollahosseini, B. Hasani, and M. H. Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, 2017. 90

M. Mori et al. The uncanny valley. *Energy*, 7(4):33–35, 1970. 4

S. Moschoglou, S. Ploumpis, M. Nicolaou, A. Papaioannou, and S. Zafeiriou. 3dfacegan: Adversarial nets for 3d face representation, generation, and translation. *International Journal of Computer Vision*, 2020. 68

I. Mpiperis, S. Malassiotis, and M. G. Strintzis. Bilinear models for 3-d face and facial expression recognition. *IEEE Transactions on Information Forensics and Security*, 3(3):498–511, 2008. 23

A. Mueller, P. Paysan, R. Schumacher, H.-F. Zeilhofer, B.-I. Berg-Boerner, J. Maurer, T. Vetter, E. Schkommodau, P. Juergens, and K. Schwenzer-Zimmerer. Missing facial parts computed by a morphable model and transferred directly to a polyamide laser-sintered prosthesis: an innovation study. *British journal of oral and maxillofacial surgery*, 49(8):e67–e71, 2011. 3

C. D. Mutto, P. Zanuttigh, and G. M. Cortelazzo. *Time-of-flight cameras and microsoft kinect (TM)*. Springer Publishing Company, Incorporated, 2012. 11

A. Myronenko, X. Song, and M. A. Carreira-Perpinán. Non-rigid point set registration: Coherent point drift. In *Advances in neural information processing systems*, pages 1009–1016, 2007. 13, 14

D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi. Efficiently combining positions and normals for precise 3d geometry. *ACM transactions on graphics (TOG)*, 24(3):536–543, 2005. 11, 21

C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, D. Vergyri, J. Sison, and A. Mashari. Audio visual speech recognition. Technical report, IDIAP, 2000. 81

T. Neumann, K. Varanasi, S. Wenger, M. Wacker, M. Magnor, and C. Theobalt. Sparse localized deformation components. *ACM Transactions on Graphics (TOG)*, 32(6):1–10, 2013. 17

A. Odena, C. Olah, and J. Shlens. Conditional Image Synthesis with Auxiliary Classifier GANs. In *International Conference on Machine Learning*, 2017. 66, 69

V. Orvalho, P. Bastos, F. I. Parke, B. Oliveira, and X. Alvarez. A facial rigging survey. In *Eurographics*, 2012. 10

F. I. Parke. *A Parametric Model for Human Faces*. PhD thesis, 1974. 3, 16

G. Passalis, I. A. Kakadiaris, T. Theoharis, G. Toderici, and N. Murtuza. Evaluation of 3d face recognition in the presence of facial expressions: an annotated deformable model approach. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, pages 171–171. IEEE, 2005. 13

G. Passalis, P. Perakis, T. Theoharis, and I. A. Kakadiaris. Using facial symmetry to handle pose variations in real-world 3d face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1938–1951, 2011. 12

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf. 32, 93

A. Patel and W. A. Smith. 3d morphable face models revisited. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1327–1334. IEEE, 2009. 13

A. Patel and W. A. Smith. Manifold-based constraints for operations in face space. *Pattern recognition*, 52:206–217, 2016. 15

P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 296–301. Ieee, 2009. 11, 15

F. Pighin and J. P. Lewis. Facial motion retargeting. In *ACM SIGGRAPH 2006 Courses*, pages 2–es. 2006. 66

A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 818–833, 2018. 68

X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 88

J. Qiu, Z. Cui, Y. Zhang, X. Zhang, S. Liu, B. Zeng, and M. Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 88

Y. Quéau, J.-D. Durou, and J.-F. Aujol. Normal integration: a survey. *Journal of Mathematical Imaging and Vision*, 60(4):576–593, 2018. 11, 20

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016. 75

A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black. Generating 3d faces using convolutional mesh autoencoders. In *European Conference on Computer Vision*, 2018. 12, 17, 18, 66, 67, 75, 78, 80

S. Reed, K. Sohn, Y. Zhang, and H. Lee. Learning to disentangle factors of variation with manifold interaction. In *International Conference on Machine Learning*, pages 1431–1439, 2014. 68

S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee. Deep visual analogy-making. In *Advances in neural information processing systems*, pages 1252–1260, 2015. 68

E. Richardson, M. Sela, and R. Kimmel. 3d face reconstruction by learning from synthetic data. In *Conference on 3D Vision*, pages 460–469, 2016. 15, 26, 87

E. Richardson, M. Sela, R. Or-El, and R. Kimmel. Learning detailed face reconstruction from a single image. *Conference on Computer Vision and Pattern Recognition*, pages 1259–1268, 2017. 26, 87, 89

S. Romdhani and T. Vetter. Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 87

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 86, 90, 91

C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 397–403, 2013. 93, 94

A. Salazar, S. Wuhrer, C. Shu, and F. Prieto. Fully automatic expression-invariant face correspondence. *Machine Vision and Applications*, 25(4):859–879, 2014. 13, 45, 75

G. Sandbach, S. Zafeiriou, M. Pantic, and D. Rueckert. A dynamic approach to the recognition of 3d facial expressions and their temporal models. In *Face and Gesture 2011*, pages 406–413. IEEE, 2011. 45

S. Sanyal, T. Bolkart, H. Feng, and M. Black. Learning to regress 3D face shape and expression from an image without 3D supervision. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 7763–7772, June 2019. 23, 26, 86, 87

A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3d face analysis. *Biometrics and identity management*, pages 47–56, 2008. 4, 10, 11, 12, 24, 32, 54, 75

D. C. Schneider and P. Eisert. Fast nonrigid mesh registration with a data-driven deformation prior. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 304–311. IEEE, 2009. 14, 49

F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 76

A. Seck, W. A. Smith, A. Dessein, B. Tiddeman, H. Dee, and A. Dutta. Ear-to-ear capture of facial intrinsics. *arXiv preprint arXiv:1609.02368*, 2016. 11

M. Sela, E. Richardson, and R. Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. *International Conference on Computer Vision*, pages 1576–1585, 2017. 26, 86, 87, 89, 94, 96, 97

S. Sengupta, A. Kanazawa, C. D. Castillo, and D. W. Jacobs. Sfsnet: Learning shape, reflectance and illuminance of facesin the wild'. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6296–6305, 2018. 21, 26, 86, 87, 88, 90, 93, 94, 95, 96, 97

Y. Seol, J. Seo, P. H. Kim, J. Lewis, and J. Noh. Weighted pose space editing for facial animation. *The Visual Computer*, 28(3):319–327, 2012. 17

G. Shamai, R. Slossberg, and R. Kimmel. Synthesizing facial photometries and corresponding geometries using generative adversarial networks. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 15(3s):1–24, 2019. 17, 23, 68

A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.*, 2(2):105–171, 2006. 71

Y. Shen, P. Luo, J. Yan, X. Wang, and X. Tang. Faceid-gan: Learning a symmetry three-player gan for identity-preserving face synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 821–830, 2018. 68

F. Shi, H.-T. Wu, X. Tong, and J. Chai. Automatic acquisition of high-fidelity facial performances using monocular videos. *ACM Transactions on Graphics (TOG)*, 33(6):1–13, 2014. 19, 21, 23

Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras. Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5541–5550, 2017. 21, 86, 87, 88

R. Slossberg, G. Shamai, and R. Kimmel. High quality facial surface and texture synthesis via generative adversarial networks. In *European Conference on Computer Vision*, pages 498–513. Springer, 2018. 67, 68

D. Smith, M. Loper, X. Hu, P. Mavroidis, and J. Romero. Facsimile: Fast and accurate scans from an image in less than a second. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5330–5339, 2019. 86, 88

W. A. Smith and E. R. Hancock. Recovering facial shape using a statistical model of surface normal direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1914–1930, 2006. 21, 88

P. Snape and S. Zafeiriou. Kernel-pca analysis of surface normals for shape-from-shading. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 88

C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. Amortised map inference for image super-resolution. In *ICLR*, 2017. 75

Q. Song, H. Ge, J. Caverlee, and X. Hu. Tensor completion algorithms in big data analytics. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(1):1–48, 2019. 26

G. Stratou, A. Ghosh, P. Debevec, and L.-P. Morency. Effect of illumination on automatic expression recognition: a novel 3d relightable facial database. In *Face and Gesture 2011*, pages 611–618. IEEE, 2011. 4, 12, 89, 90, 93

Y. Sun, X. Chen, M. Rosato, and L. Yin. Tracking vertex flow and model adaptation for three-dimensional spatiotemporal face analysis. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(3):461–474, 2010. 40, 41, 47

M. Suttie, T. Foroud, L. Wetherill, J. L. Jacobson, C. D. Molteno, E. M. Meintjes, H. E. Hoyme, N. Khaole, L. K. Robinson, E. P. Riley, et al. Facial dysmorphism across the fetal alcohol spectrum. *Pediatrics*, 131(3):e779–e788, 2013. 3

J. R. Tena, F. De la Torre, and I. Matthews. Interactive region-based linear 3d face models. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH âĂŹ11. Association for Computing Machinery, 2011. 17

J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural computation*, 12(6):1247–1283, 2000. 18, 68

F. B. ter Haar and R. C. Veltkamp. 3d face model fitting for recognition. In *European conference on computer vision*, pages 652–664. Springer, 2008. 15

A. Tewari, M. Zollhöfer, H. Kin, P. G. G. Bernard, P. Pérez, and C. Theobalt. MoFa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *International Conference on Computer Vision*, pages 3715–3724, 2017. 15, 18, 24, 26, 86, 87

A. Tewari, M. Zollhöfer, P. Garrido, F. Bernard, H. Kim, P. Pérez, and C. Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2549–2559, 2018. 20, 26, 67, 86, 88

A. Tewari, F. Bernard, P. Garrido, G. Bharaj, M. Elgharib, H.-P. Seidel, P. Pérez, M. Zollhofer, and C. Theobalt. Fml: face model learning from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10812–10822, 2019. 18, 23, 26, 87

A. Tewari, M. Elgharib, G. Bharaj, F. Bernard, H.-P. Seidel, P. Pérez, M. Zöllhofer, and C. Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images, cvpr 2020. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, june 2020.

J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. Real-time expression transfer for facial reenactment. *ACM Trans. Graph.*, 34 (6):183–1, 2015. 47

J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016. 15, 18

J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Headon: Real-time reenactment of human portrait videos. *ACM Transactions on Graphics 2018 (TOG)*, 2018a. 47

J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Facevr: Real-time gaze-aware facial reenactment in virtual reality. *ACM Transactions on Graphics 2018 (TOG)*, 2018b. 47

D. Thomas and R.-I. Taniguchi. Augmented blendshapes for real-time simultaneous 3d head modeling and facial motion capture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3299–3308, 2016. 20, 47, 87

A. T. Tran, T. Hassner, I. Masi, and G. Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. *Conference on Computer Vision and Pattern Recognition*, pages 5163–5172, 2017a. 15, 26, 87

A. T. Tran, T. Hassner, I. Masi, E. Paz, Y. Nirkin, and G. G. Medioni. Extreme 3d face reconstruction: Seeing through occlusions. In *CVPR*, pages 3935–3944, 2018. 18, 20, 26, 67, 89, 94, 95, 96, 97

L. Tran and X. Liu. Nonlinear 3d face morphable model. In *In Proceeding of IEEE Computer Vision and Pattern Recognition*, Salt Lake City, UT, June 2018. 17, 105

L. Tran, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In *CVPR*, 2017b. 68

L. Tran, F. Liu, and X. Liu. Towards high-fidelity nonlinear 3d face morphable model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1126–1135, 2019. 17, 18, 20, 26, 66, 88, 105

G. Trigeorgis, P. Snape, I. Kokkinos, and S. Zafeiriou. Face normals" in-the-wild" using fully convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 340–349. IEEE, 2017. 21, 86, 87, 88, 89, 90, 93, 94, 95, 96

L. C. Trutoiu, E. J. Carter, N. Pollard, J. F. Cohn, and J. K. Hodgins. Spatial and temporal linearities in posed and spontaneous smiles. *ACM Transactions on Applied Perception (TAP)*, 11(3):1–15, 2014. 66

B. Usman, N. Dufour, K. Saenko, and C. Bregler. Puppetgan: Cross-domain image manipulation by demonstration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9450–9458, 2019. 68

L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, and C. Theobalt. Lightweight binocular facial performance capture under uncontrolled lighting. *ACM Transactions on Graphics*, 31(6):187:1–11, 2012. 11, 21, 48

O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1681–1707. Wiley Online Library, 2011. 12

S. van Steenkiste, F. Locatello, J. Schmidhuber, and O. Bachem. Are disentangled representations helpful for abstract visual reasoning? In *Advances in Neural Information Processing Systems*, pages 14222–14235, 2019. 68

M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Computer Vision — ECCV 2002*, pages 447–460, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. 18, 68

N. Verma, E. Boyer, and J. Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2598–2606, 2018. 83

D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. *ACM Transactions on Graphics*, 24(3):426–433, 2005. 9, 18, 19, 23, 25, 33, 68

K. Wampler, D. Sasaki, L. Zhang, and Z. Popović. Dynamic, expressive speech animation from a single mesh. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 53–62, 2007. 19

M. Wang, Y. Panagakis, P. Snape, and S. Zafeiriou. Learning the multilinear structure of visual data. In *Conference on Computer Vision and Pattern Recognition*, pages 4592–4600, 2017. 19, 25, 26, 33, 37, 38

M. Wang, Z. Shu, S. Cheng, Y. Panagakis, D. Samaras, and S. Zafeiriou. An adversarial neuro-tensorial approach for learning disentangled representations. *International Journal of Computer Vision*, 127(6-7):743–762, 2019. 19

M. Wang, D. Bradley, S. Zafeiriou, and T. Beeler. Facial expression synthesis using a global-local multilinear framework. In *Eurographics*, 2020. 23

X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 88

Y. Wang, X. Huang, C.-S. Lee, S. Zhang, Z. Li, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang. High resolution acquisition, learning and transfer of dynamic 3-d facial expressions. In *Computer Graphics Forum*, volume 23, pages 677–686. Wiley Online Library, 2004. 13, 19, 47

Y. Wang, M. Gupta, S. Zhang, S. Wang, X. Gu, D. Samaras, and P. Huang. High resolution tracking of non-rigid motion of densely sampled 3d data using harmonic maps. *International Journal of Computer Vision*, 76(3):283–300, 2008. 47

T. Weise, H. Li, L. Van Gool, and M. Pauly. Face/off: Live facial puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation*, pages 7–16, 2009. 13, 16, 47

T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. In *ACM transactions on graphics (TOG)*, volume 30, page 77. ACM, 2011. 47

R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):191139, 1980. 11

C. Wu, B. Wilburn, Y. Matsushita, and C. Theobalt. High-quality shape from multi-view stereo and shading under general illumination. In *CVPR 2011*, pages 969–976. IEEE, 2011. 11

C. Wu, D. Bradley, M. Gross, and T. Beeler. An anatomically-constrained local deformation model for monocular face capture. *ACM Transactions on Graphics (TOG)*, 35(4):1–12, 2016. 17

C. Wu, T. Shiratori, and Y. Sheikh. Deep incremental learning for efficient high-fidelity face tracking. *ACM Transactions on Graphics (TOG)*, 37(6):1–12, 2018. 48

Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020. 27

S. Yamaguchi, S. Saito, K. Nagano, Y. Zhao, W. Chen, K. Olszewski, S. Morishima, and H. Li. High-fidelity facial reflectance and geometry inference from an unconstrained image. *ACM Transactions on Graphics (TOG)*, 37(4): 162, 2018. 20, 68, 89

F. Yang, J. Wang, E. Shechtman, L. Bourdev, and D. Metaxas. Expression flow for 3d-aware face component transfer. In *ACM SIGGRAPH 2011 papers*, pages 1–10. 2011. 17, 23

F. Yang, L. Bourdev, E. Shechtman, J. Wang, and D. Metaxas. Facial expression editing in video using a temporally-smooth factorization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 861–868. IEEE, 2012. 19

H. Yang, H. Zhu, Y. Wang, M. Huang, Q. Shen, R. Yang, and X. Cao. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 19, 20

L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3d facial expression database for facial behavior research. In *Conference on Automatic Face and Gesture Recognition*, pages 211–216, 2006. 4, 12, 32, 54, 75

L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3d dynamic facial expression database. In *IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, 2008. 4, 11, 12, 24, 40, 45, 46, 54, 75

Y. Yoon, G. Choe, N. Kim, J.-Y. Lee, and I. S. Kweon. Fine-scale surface normal estimation using a single nir image. In *Proceedings of the European Conference on Computer Vision*, 2016. 88

H. Yu, O. G. Garrod, and P. G. Schyns. Perception-driven facial expression synthesis. *Computers & Graphics*, 36(3):152–162, 2012. 45

S. Zafeiriou, M. Hansen, G. Atkinson, V. Argyriou, M. Petrou, M. Smith, and L. Smith. The photoface database. In *CVPRW*, 2011. 12, 87, 89, 90, 92, 93, 98

M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 32

C. Zhang, W. A. Smith, A. Dessein, N. Pears, and H. Dai. Functional faces: Groupwise dense correspondence using functional maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5033–5041, 2016. 14

L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: High-resolution capture for modeling and animation. In *ACM Annual Conference on Computer Graphics*, pages 548–558, August 2004. 48

R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape-from-shading: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8):690–706, 1999. 86, 88

X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, and J. M. Girard. Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706, 2014. 4, 12, 45, 46, 54, 75

X. Zhang, L. Yin, and J. F. Cohn. Three dimensional binary edge feature representation for pain expression analysis. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 1, pages 1–7. IEEE, 2015. 3, 45

Y. Zhang and T. Funkhouser. Deep depth completion of a single rgb-d image. In *CVPR*, 2018. 86, 88

B. Zhao, X. Wu, Z.-Q. Cheng, H. Liu, Z. Jie, and J. Feng. Multi-view image generation from a single-view. In *ACM International Conference on Multimedia*, 2018. 68

Y. Zhou, J. Deng, I. Kotsia, and S. Zafeiriou. Dense 3d face decoding over 2500fps: Joint texture & shape convolutional mesh decoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1097–1106, 2019. 17, 66, 88

X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Li. Face alignment across large poses: A 3d solution. In *Conference on Computer Vision and Pattern Recognition*, pages 146–155, 2016. 26

X. Zhu, X. Liu, Z. Lei, and S. Z. Li. Face alignment in full pose range: A 3d total solution. *IEEE transactions on pattern analysis and machine intelligence*, 41 (1):78–92, 2017. 15, 94, 95, 96, 97

J. Zivanov, P. Paysan, and T. Vetter. Facial normal map capture using four lights–an effective and inexpensive method of capturing the fine scale detail of human faces using four point lights. In *In Proc. GRAPP*. Citeseer, 2009. 11

M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, et al. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Transactions on Graphics (ToG)*, 33(4):1–12, 2014. 47

M. Zollhöfer, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Pérez, M. Stamminger, M. Nießner, and C. Theobalt. State of the Art on Monocular 3D Face Reconstruction, Tracking, and Applications. *Computer Graphics Forum (Eurographics State of the Art Reports 2018)*, 2018. 10, 47

A Appendix

A.1 Chapter 5 - Architecture

Figure A.1 shows the architecture for the Generator and Discriminator (the latter with the classification branches). Here, d_{id}, d_{exp} and d_{noise} are the dimensions for identity, expression and noise, respectively; n_{id} is the number of distinct labels for identity, and n_{exp} the number of distinct labels for expression. We use Leaky ReLU with a slope of 0.2.

A.2 Chapter 5 - Decoupling Evaluation

We train the embedding networks using a Resnet-18 architecture with input images of size 224×224. The images contain the orthographic projection of the facial mesh, and the values in the RGB channels encode the normal direction of each vertex, as we found this to give better results than the UV images. The networks were trained using the datasets described in Section 5.4.2 with the provided labels. The threshold is selected such that it maximizes the accuracy on the validation set, while keeping the False Acceptance Rate (FAR) below 10%. We build the validation set by randomly choosing an equal number of positive and negative pairs from the testing split. We choose 0.14 as threshold for identity, which achieves 98.66% accuracy and a FAR of 1.21%. For expression we use 0.226 as threshold, which achieves 84.2% of accuracy and a FAR of 8.03%.

Operation	Activation	Output Shape
$z \sim \mathcal{N}(0, I)$	–	$d_{id} + d_{exp} + d_{noise}$
Linear	LReLU	512
Linear	–	66387
Reshape	–	22129×3

(a) Generator

Operation	Activation	Output Shape
Input	–	22129×3
Geometry mapping	–	$3 \times 64 \times 64$
Common branch		
Conv 3×3	LReLU	$16 \times 32 \times 32$
Conv 3×3	LReLU	$32 \times 16 \times 16$
Discriminator branch		
Conv 3×3	LReLU	$64 \times 8 \times 8$
Conv 3×3	LReLU	$128 \times 4 \times 4$
Reshape	–	2048
Linear	–	1
Identity branch		
Conv 3×3	LReLU	$64 \times 8 \times 8$
Conv 3×3	LReLU	$128 \times 4 \times 4$
Reshape	–	2048
Linear	–	n_{id}
Expression branch		
Conv 3×3	LReLU	$64 \times 8 \times 8$
Conv 3×3	LReLU	$128 \times 4 \times 4$
Reshape	–	2048
Linear	–	n_{exp}

(b) Discriminator and Classifiers.

Figure A.1 – Generator and Discriminator used for the GAN architecture of Chapter 5