WHAT DOES 2D GEOMETRIC INFORMATION REALLY TELL US ABOUT 3D FACE SHAPE?

ANIL BAS

PHD

UNIVERSITY OF YORK

COMPUTER SCIENCE

JUNE 2018

Abstract

A face image contains geometric cues in the form of configurational information (semantically meaningful landmark points and contours). In this thesis, we explore to what degree such 2D geometric information allows us to estimate 3D face shape.

First, we focus on the problem of fitting a 3D morphable model to single face images using only sparse geometric features. We propose a novel approach that explicitly computes hard correspondences which allow us to treat the model edge vertices as known 2D positions, for which optimal pose or shape estimates can be linearly computed. Moreover, we show how to formulate this shape-from-landmarks problem as a separable nonlinear least squares optimisation.

Second, we show how a statistical model can be used to spatially transform input data as a module within a convolutional neural network. This is an extension of the original spatial transformer network in that we are able to interpret and normalise 3D pose changes and self-occlusions. We show that the localiser can be trained using only simple geometric loss functions on a relatively small dataset yet is able to perform robust normalisation on highly uncontrolled images. We consider another extension in which the model itself is also learnt.

The final contribution of this thesis lies in exploring the limits of 2D geometric features and characterising the resulting ambiguities. 2D geometric information only provides a partial constraint on 3D face shape. In other words, face landmarks or occluding contours are an ambiguous shape cue. Two faces with different 3D shape can give rise to the same 2D geometry, particularly as a result of perspective transformation when camera distance varies. We derive methods to compute these ambiguity subspaces, demonstrate that they contain significant shape variability and show that these ambiguities occur in real-world datasets.

List of Contents

List of Tables

List of Figures

List of Symbols

Symbol	Description	Object type
N	No. 3D vertices	$\in \mathbb{Z}$
D	No. model dimensions	$\in \mathbb{Z}$
L	No. 2D landmarks	$\in \mathbb{Z}$
n	No. vertices/images	$\in \mathbb{Z}$
$\mathbf{Q}$	Principal components	$\in \mathbb{R}^{3N\times D}$ or $\in \mathbb{R}^{2N\times D}$
$\boldsymbol{\alpha}$	Shape parameter vector	$\in \mathbb{R}^{D}$
$\bar{\mathbf{s}}$	Mean face shape	$\in \mathbb{R}^{3N}$ or $\in \mathbb{R}^{2N}$
σ	Standard deviation	$\in \mathbb{R}^{D}$
$\mathbf{v}_i$	ith 3D point (vertex)	$\in \mathbb{R}^{3}$
$\mathbf{x}_i$	ith 2D point	$\in \mathbb{R}^{2}$
$\mathbf{P}$	Orthographic projection matrix	$\in \mathbb{R}^{2\times 3}$
$\mathbf{R}$	Rotation matrix	$\in SO(3)$
$\mathbf{r}$	Axis-angle vector	$\in \mathbb{R}^{3}$
ϕ	Rotation angle	$\in \mathbb{R}$
$\mathbf{t}$	Translation vector	$\in \mathbb{R}^{2}$ or $\mathbb{R}^{3}$
s	Scale	$\in \mathbb{R}_{>0}$
f	Focal length	$\in \mathbb{R}_{>0}$
$\mathbf{K}$	Camera intrinsics	$\in \mathbb{R}^{3\times 3}$
E	Reprojection error	$\in \mathbb{R}_{\geq 0}$
ε	Objective function	$\in \mathbb{R}_{\geq 0}$
ℓ	Loss function	$\in \mathbb{R}_{\geq 0}$
$\mathbf{d}$	Vector of residuals	$\in \mathbb{R}^{2L}$ or $\mathbb{R}^{3L}$
$\mathbf{I}_n$	Identity matrix	$\in \{0,1\}^{n\times n}$
$\mathbf{1}_n$	Column vector of ones	$\in \{1\}^{n}$
$\mathbf{J}$	Jacobian matrix	$\in \mathbb{R}^{2L\times 4}$ or $\mathbb{R}^{3L\times 4}$
t_z	Face-camera distance	$\in \mathbb{R}_{>0}$
k	Threshold value	$\in \mathbb{R}_{>0}$
γ	Scaling factor	$\in \mathbb{R}$
$\boldsymbol{\Pi}$	2D projection	$\in \mathbb{R}^{2L\times D}$ or $\mathbb{R}^{3L\times D}$
$\mathbf{f}$	Flexibility modes	$\in \mathbb{R}^{D}$
λ_i	ith eigenvalue	$\in \mathbb{R}$
$\mathcal{B}$	Occluding boundary vertices	$\subset \{1,\ldots,N\}$
w	Weight	$\in [0,1]$
$\otimes$	Kronecker product	Operator

Acknowledgements

Doing a PhD is an amazing experience and sometimes it looks insanely complicated. Here, I would like to offer a gentle reminder to myself and others, in large, friendly letters: DON'T PANIC.

I would like to thank my colleagues and friends for their input, advice and support. Special thanks go to the co-authors of our papers for inspiring collaboration and contribution.

I would like to thank all the members of the CVPR group and Department of Computer Science for providing a comfortable and productive atmosphere.

I would like to thank Marmara University and the Council of Higher Education, Turkey, for funding my study.

I would like to express sincere gratitude to my assessor, Professor Richard Wilson, for his invaluable feedback at every stage of the process, as well as to my external examiner, Dr Moi Hoon Yap, for carefully reviewing my thesis and providing constructive and insightful comments. It is immensely appreciated.

Finally, I would like to thank my beloved family, without whom this thesis would never have been started; my exceptional, inspiring supervisor, Dr William Smith, who always made time for me and dealt with all my questions in long meetings, without whom it would never have been built; and to my dearest, soon-to-be-wife, Imke, without whom it would never have been finished.

As a great man once said, “our true mentor in life is science”.

Author's Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References.

Most parts of this thesis have been published by the author. A complete list of publications is below.

- A. Bas, W.A.P. Smith. **What Does 2D Geometric Information Really Tell Us About 3D Face Shape?** arXiv preprint arXiv:1708.06703, 2018.

- A. Bas, W.A.P. Smith. **Statistical Transformer Networks: Learning Shape and Appearance Models via Self Supervision**. arXiv preprint arXiv:1804.02541, 2018.

- A. Bas, P. Huber, W.A.P. Smith, M. Awais and J. Kittler. **3D Morphable Models as Spatial Transformer Networks**. In Proc. ICCV Workshop on Geometry Meets Deep Learning, pp. 904-912, 2017.

- A. Bas, W. A. P. Smith, T. Bolkart and S. Wuhrer. **Fitting a 3D Morphable Model to Edges: A Comparison Between Hard and Soft Correspondences**. In Proc. ACCV Workshop on Facial Informatics, pp. 377-391, 2016.

- A. Bas and W. A. P. Smith. **CCTV Revisited: Face Analysis Under Challenging Conditions**. YorkTalks, Finalist – Science Category, University of York PhD Research Spotlight Competition, 2016.

Chapter 1

Introduction

With one look, humans can recognise a face, understand emotions as well as estimate age, gender and ethnicity. This complex phenomenon comprises two functions: capturing and understanding. In terms of the former, we have already exceeded human-level capabilities by means of cameras with cutting-edge lenses and highly sophisticated sensors. In terms of the latter, however, computers still have a long way to go before successfully replicating the human vision system.

A specific domain of interest for the understanding of faces is the estimation of face shape. This is a fundamental problem for many scenarios, ranging from lie detection to lip reading, which requires algorithms to exploit the potential use of every single cue to be able to operate on everyday images. Within this context, face analysis rises as a challenging research area in the field of computer vision and machine learning.

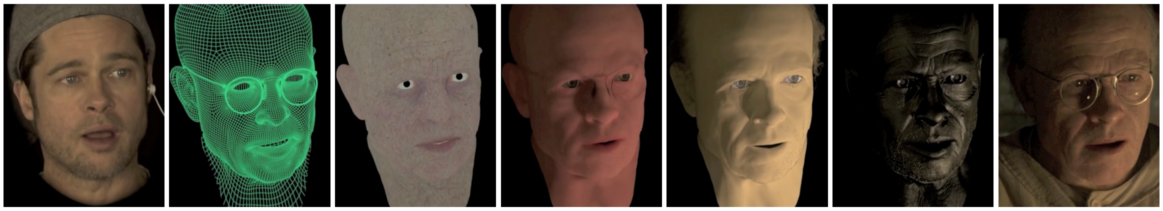


Figure 1.1: An example use of a 3D face model in a movie production. Close-up shot of actor's face, model's shape, texture, illumination components and final rendering result (from left to right). *The Curious Case of Benjamin Button*, a movie by Paramount Pictures and Warner Bros. Pictures that stars Brad Pitt, won the Academy Award for Best Visual Effects in 2009. Cropped clips from "Making of *The Curious Case of Benjamin Button*" courtesy of Digital Domain.

1.1 Benefits of 3D Face Modelling

Face analysis using 3D models has many benefits. A 3D model allows intrinsic properties of a face (shape and texture) to be explicitly disentangled from the other environmental factors that determine the appearance of a face. This simplifies analysis tasks since the need for invariance to these external factors is avoided and facilitates synthesis since the forward rendering process can be explicitly modelled. Figure 1.1 shows an example use of a 3D model in a movie production setting, involving computer vision for 3D capture, tracking for animation of a rigged 3D model and photorealistic rendering for the output image.

Pose normalisation for faces is an important factor in tackling the problem of recognition and alignment. Although this can be solved partially in 2D models, it is handled more systematically since pose is separated from shape (using camera models) in 3D models. Moreover, 3D models can generate face images under any pose, expression and illumination conditions. This process sometimes introduces self-occlusions on appearance due to nonvisible regions which can be restored by synthesising textures [Saito et al., 2017]. Alternatively, we can reconstruct the model combining multiple images taken from different views.

Some models come with an additional **expression model** [Li et al., 2017] and are more suitable to work with real-world images. Therefore, expression invariant data can be used effectively. Moreover, the identity can be fixed across the video sequence while optimising for the expression, leading to increased performance. We can also remove the **effects of lighting** by separating shape model from other factors. This makes the processing algorithm immune to illumination or appearance changes, using only geometric information in the data.

The human face varies due to personal preferences such as weight and hairstyle changes, facial hair and tanned skin. It also changes over time due to the effects of ageing. For example, the hairline could recede, muscles atrophy, the face skin gets less elastic which leads to wrinkles and could be damaged by ultraviolet exposure (photoageing) as well. 3D face models allow us to derive **parametric descriptions** of face attributes and produce variations in their generative capacity. In [Paysan et al., 2009], it is possible to manipulate the model by varying attributes individually. Similarly, [Cao et al., 2015] proposed a generic modelling technique that is capable of describing fine scale facial details.

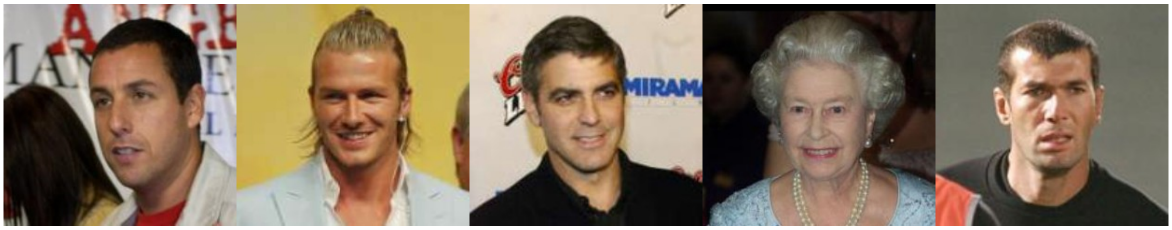


Figure 1.2: Uncontrolled face image examples from LFW dataset [Huang et al., 2007].

These 3D face models have been widely used in many computer vision tasks including face recognition, attribute (e.g. age, gender and race) estimation, data augmentation, avatar design and facial animation.

1.2 Benefits of Estimating 3D Face Models from 2D Images

Surely, having a 3D model of a face would be beneficial since 3D shape is invariant to any pose and illumination conditions, while expression can be explicitly modelled as a deformation of the neutral face. This is particularly useful for surveillance applications or in the search for missing children (with modelling age progression [Scherbaum et al., 2007] as well). However, acquiring such a model is difficult and usually requires a special capture device. Furthermore, 3D scans are often unattainable because of limited accessibility of the subject (e.g. in a security setting) along with high-cost and speed limitations or simply preferences. (Perhaps this will change in the near future since having 3D baby scans in pregnancy has become mainstream [Roberts et al., 2017].)

Alternatively, we can reconstruct 3D models from conventional images. This vastly broadens the scope of application and potentially leads to new applications. A 2D face image contains various cues that can be exploited to estimate 3D shape. These cues could be appearance-based (texture, colour, pixel intensity) or photometry-based (albedo, shading, specularities) or geometry-based (landmarks, internal and external edges).

The process of extracting intrinsic face properties from uncontrolled images is an ill-posed problem since appearance is strongly influenced by extrinsic factors such as pose, motion blur, saturation, sensor noise and illumination changes. In addition to these factors, one should consider the fact that the face itself is complex, as a result of combining expression and identity. Figure 1.2 shows a set of uncontrolled face images from [Huang et al., 2007].

Estimating 3D face models from images offers a wide range of functionality to applications such as face manipulation [Thies et al., 2016], expression analysis [Mpiperis et al., 2008], face beautification [Scherbaum et al., 2011], occlusion handling [Egger et al., 2016], driver monitoring [Vicente et al., 2015] and human-computer interaction [Grupp et al., 2016].

1.3 Research Question

The extreme variation of scene parameters and modelling every factor of appearance in these images has proven a very challenging task. Methods that attempt to model all of these factors are likely to be fragile and break when assumptions are not met. For example, photometric information provides a cue to the 3D shape of a face [Smith and Hancock, 2006], however, it requires estimates of lighting as well as camera and reflectance properties which make it difficult to apply to in-the-wild images. Moreover, in some conditions, the shape-from-shading cue may be entirely absent. Perfectly ambient light cancels out all shading other than ambient occlusion which provides only a very weak shape cue [Prados et al., 2009].

We therefore chose to condense our research theme by asking to what extent we could use *geometric information* for 3D face shape recovery. 2D geometric information holds out the hope of estimating 3D face shape without having to model and explain real-world appearance. This is sometimes referred to as "configurational" information and includes the relative layout of features (usually encapsulated in terms of the position of semantically meaningful landmark points) and contours (caused by occluding boundaries or texture edges). Geometric information has been used in 3D face reconstruction [Blanz et al., 2004, Knothe et al., 2006, Patel and Smith, 2009, Aldrian and Smith, 2013, Cao et al., 2014a], though most commonly only for a rough initialisation, while we seek to understand whether accurate 3D models can be estimated using this information alone. We benefit from the fact that landmark detection on highly uncontrolled face images is now a mature research field with benchmarks [Sagonas et al., 2016] providing an indication of likely accuracy. The primary advantage of using geometric cues is that they provide direct information about the shape of the face, without having to model the photometric image formation process or interpret appearance. As an example, a profile view derived from the occluding boundary reveals strong information about the shape of the nose.

1.4 Contributions

On this basis, there are three main contributions of this thesis. The thesis has a separate chapter for each proposed idea. Each chapter covers a brief summary of the novelty, investigates thoroughly the body of research, provides qualitative and quantitative evaluation and concludes with implications for future research.

Classical Optimisation-based Fitting The face shape estimation problem from a single image can be approached by using a statistical shape model. This transforms the problem to one of analysis-by-synthesis and provides a strong statistical prior to constrain the problem. This *model fitting* problem is particularly challenging in the presence of weak data when the model prior dominates and the result tends to end up resembling the mean face.

To our knowledge, model fitting has not previously been approached as a non-rigid alignment problem. The challenge here is that model-image correspondence is unknown, so previous methods have avoided computing an explicit correspondence and relied on a soft correspondence objective function. We approach the model fitting problem as one of establishing correspondence when there is no prior knowledge available. This relies solely on geometric information, namely the set of 2D landmark points and edges. Edges are an attractive feature to exploit because they are relatively insensitive to changes in illumination and camera parameters. Thus, our first contribution is a fully automatic method for fitting a 3D morphable model to a single image using landmarks and edge features.

Geometry Meets Deep Learning The recent advancements in deep learning have resulted in strong performance gains in the face analysis domain. Although it is common to build face recognition/classification systems using convolutional neural network (CNN) architectures, the idea of incorporating a 3D morphable model into a neural network as a module which learns pose normalisation, surprisingly, still has not been implemented yet. Our second contribution is a state-of-the-art approach to use a 3D face model within a CNN based on purely geometric transformations. This is an extension of the original Spatial Transformer Networks (STN) [Jaderberg et al., 2015] in that we are able to interpret and normalise 3D pose changes and self-occlusions. The trained localisation part of the network is independently useful since it learns to fit a 3D morphable model to a single image.

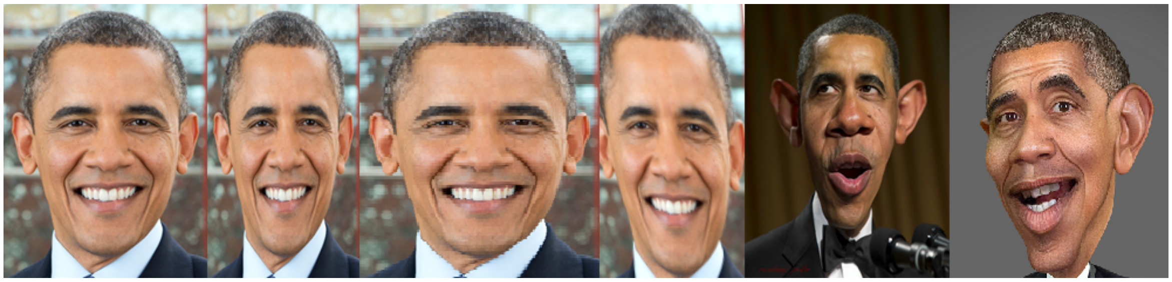


Figure 1.3: The distortion of 2D face shape. A set of different geometries of the same image (left four). Caricatures of the same person (right two).

We extend this idea further by replacing the fixed morphable model with a statistical shape model which is itself learnt. By training a network containing such a module end-to-end for a particular task, the network learns the optimal non-rigid alignment of the input data. Moreover, the statistical shape model is learnt with no direct supervision and can be reused for other tasks. Besides training for a specific task, we also show that our network can learn a shape model using generic loss functions, including a loss inspired by the minimum description length principle in which an appearance model is also learnt from scratch with no supervision at all, even identity labels.

Geometric Ambiguities Face recognition in humans is remarkably insensitive to distortions in 2D shape, so that face images which are very different from each other in terms of facial geometry could be easily recognised as the same person [Sandford and Burton, 2014], even when caricatured [Rhodes et al., 1987] or heavily blurred [Hole et al., 2002], shown in Figure 1.3. On the other hand, machine face recognition systems have incorporated extremely elaborate schemes for estimating and normalising shape variation [Taigman et al., 2014], and our first two contributions are based on the idea that 2D geometric features contain enough information to uniquely estimate 3D face shape. Apparently, human perception works differently than how we model face recognition systems.

While it is unclear whether a 3D model is needed for face recognition and whether humans estimate one as part of the recognition process, we can still ask if the task of estimating one from a 2D image is possible. Our final contribution is to investigate the limits and ambiguities of using geometric features to compute 3D face shape. In other words, we answer the question: what does 2D geometric information really tell us about 3D face shape?

1.5 Thesis Structure

The structure of this thesis is as follows. In Chapter 2, we start by reviewing prior studies from the vantage point of our particular perspective. We give an overview of the state-of-the-art methods in optimisation-based model fitting and in convolutional neural networks. In Chapter 3, we investigate the model fitting and correspondence problem. We propose a novel algorithm for fitting 3D face model to single images using geometric features. We also include a comparison between alternating linear least squares and separable nonlinear least squares. We shift our focus from a classical optimisation-based approach to neural networks in Chapter 4. We explore the problem of face shape construction in the deep learning domain and propose networks based on purely geometric transformations that can be trained in an unsupervised fashion. In Chapter 5, we address the limitations of 2D geometric features to compute 3D shape. Computing flexibility modes under orthographic and perspective projections, we demonstrate that ambiguities exist in both synthetic data and real images. Finally, we summarise our research and discuss possible research directions in Chapter 6.

Chapter 2

Related Work

2.1 Introduction

The main aim of this thesis is to explore the use of 2D geometric information on 3D face shape recovery. This issue is directly related to face modelling, rigid and non-rigid alignment, the correspondence problem and model fitting techniques from various perspectives, including optimisation-based fitting and deep learning; in conjunction with the use of geometric cues. In accordance, this chapter provides a review of relevant literature on these topics. The objective here is not to cover all the aspects of the shape estimation problem, but to acknowledge selectively some of the works which are closely related to the contents of this thesis.

2.2 Face Modelling

A face model refers to a generic model which describes features of faces such as shape, texture, etc. Built from 2D images or 3D scans, face models allow us to recover desired properties (e.g. 3D face shape, 2D pose, skin reflectance) by finding the optimal parameters.

Principal Component Analysis (PCA) is a statistical method to analyse the correlational relationship of the example data [Jolliffe, 2002]. It is a standard technique for characterising the distribution of large amounts of data and a useful tool for finding the patterns in the set (correlations). These patterns are identified as *principal components* and expressed by a linear combination of correlated variables.

In the context of statistical face shape modelling, PCA provides a mean (a mean face shape) and variance (principal components that represent the shape variation across the dataset). Therefore, it is possible to generate new faces that are not included in the input data by adding linear combinations of the principal components to the mean face.

2.2.1 2D Face Modelling

[Turk and Pentland, 1991] proposed a pioneering method for face imaging based on principal component analysis. The predominant idea of their study pertains to the conversion of face image data into a set of characteristic features known as *eigenfaces*. These are the eigenvectors which are derived from the covariance matrix of a set of face images. Although there are certain drawbacks, such as working with only one pose angle or illumination condition or working in a holistic manner (as opposed to feature-based), eigenfaces is widely seen as a major milestone and created a foundation for many new studies.

[Belhumeur et al., 1997] extended the eigenfaces approach to *fisherfaces* by employing a special classifier called Fisher's Linear Discriminant [Fisher, 1936] to categorise different species of irises. They used the classifier in their model as a feature extraction immune to facial expression and illumination changes.

While these early approaches had difficulty with major feature changes in appearance such as an open mouth or glasses, later studies used statistical methods to obtain more accurate and efficient results when dealing with challenging variations. [Cootes et al., 1995] built an Active Shape Model (ASM) based on a face shape variation from examples. This statistical model, previously known as Smart Snake Algorithm [Cootes and Taylor, 1992], described each shape in the given sets of training images as a set of labelled landmark points which are consistent between shapes. By applying PCA to the training set, the model formed with a reduced number of parameters and held a wide variety of face shapes.

There has been many influential extensions of Active Shape Model. One of the most remarkable is the Active Appearance Model (AAM) originally proposed in [Edwards et al., 1998] and later extended in [Cootes et al., 1998, Cootes et al., 2001]. The idea of AAM is to merge shape and texture information as *appearance* and take advantage of intensity and colour as well as face structure.

[Cristinacce and Cootes, 2006] presented a combined model called the Constrained Local Model which is a mixture of shape constraints and local feature templates. It shows similarities to AAM but alternatively, the face model generated by a set of local feature templates rather than by modelling the whole face.

2.2.2 3D Face Modelling

Although 2D models were commonly used until recently, 3D face models potentially provide a more structured route. For example, the fitting robustness and rate of convergence of the 3D algorithm are higher than 2D [Matthews et al., 2007]. Another benefit of 3D models is that pose and shape parameters are separated from each other whereas in 2D models the pose is often represented as non-rigid motion. Likewise, the 3D model usually includes appearance information which helps to remove the effects of illumination. The main disadvantage is that the building a 3D model requires a large amount of well-aligned 3D scans.

Blanz and Vetter extended shape model to 3D by applying PCA to shape and texture data from 3D scans in two inter-related articles [Blanz and Vetter, 1999, Blanz and Vetter, 2003]. This model, named 3D Morphable Model (3DMM), consists of a vector space representation of faces defined by parameterised triangular meshes which are in dense correspondence. This seminal work has since been used widely in the fields of computer vision, graphics and machine learning.

[Amberg et al., 2008] proposed a model for expression-invariant face recognition. Built by merging neutral and expressive face components, the expression model was used for manipulations in videos. Moreover, their approach combined the 3D morphable model and the idea of [Romdhani et al., 2006] that allows to obtain pose and lighting-invariant face recognition by separating shape and albedo parameters from pose and lighting.

Similarly, [Vlasic et al., 2005] presented a multilinear face model to transfer facial movements from a source face to a target face. The model uses a collection of face meshes and sets up point-to-point correspondence similar to 3DMM. Moreover, it includes estimated identity and expression variations of these face scans. This allows the model to adjust facial motion by varying one attribute while keeping others constant. Figure 2.1 shows the visualised data of varied forms of 3D face scans.

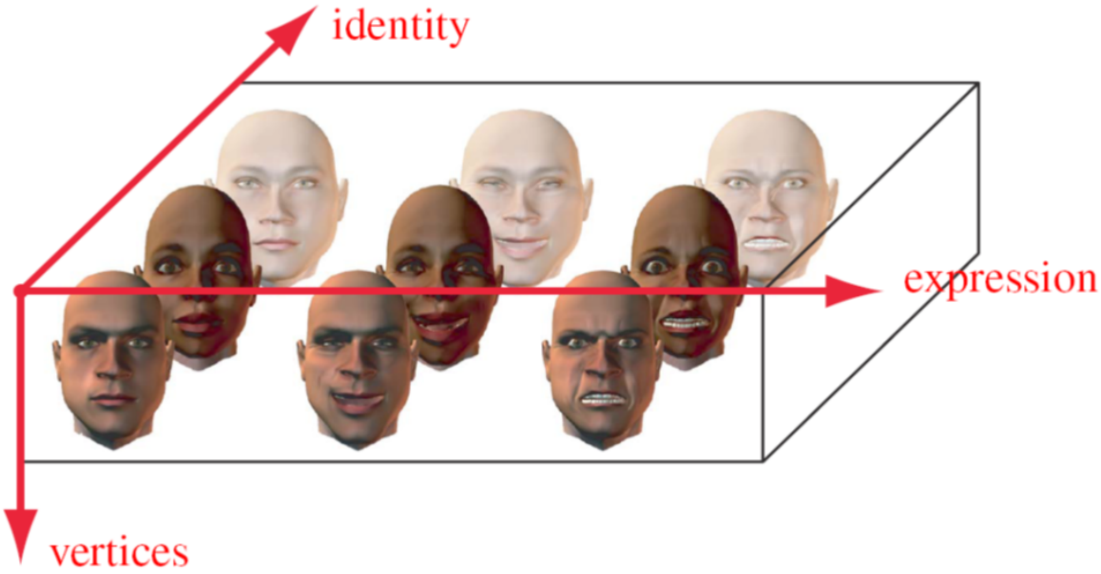


Figure 2.1: Multilinear model attribute variation. The first mode contains vertices, while the second and third modes correspond to expression and identity respectively [Vlasic et al., 2005].

Recently, we see the use of the Gaussian process rather than PCA to handle facial details and shape variations in model construction [Lüthi et al., 2017]. Similarly, [Koppen et al., 2018] proposed the Gaussian Mixture 3DMM which was constructed using diverse ethnic groups (Caucasian, Chinese and African) and extended the model with a wide range of age and gender modes.

In this thesis, we mainly use the Basel Face Model (BFM), a version of 3DMM which was made publicly available in [Paysan et al., 2009]. BFM includes separate mean shape and texture vectors along with the principal components of shape and texture variations. One can estimate new faces by combining the mean face with variation modes since the model is linear. The model is later extended to Probabilistic and Semantic Morphable Models [Gerig et al., 2018, Egger et al., 2018] that contain expression and occlusion models, better illumination estimation as well as improved age distribution.

The fundamental challenge in constructing a model is to create dense correspondence between all facial meshes. In the case of 3DMM, an optical flow algorithm was used to compute point-to-point correspondence sourced from the texture information of 3D face scans. Likewise, some studies worked with assumed correspondence points or manual initialisation.

Thus far, we explained what a face model is and examined 2D and 3D face modelling studies. In the following section, we focus on the model fitting problem, largely between 3D model and 2D image, and articulate the problem by investigating various classifications of relevant literature.

2.3 Model Fitting

The problem of fitting a statistical face model can be viewed as one of simultaneous non-rigid alignment and inverse rendering. The non-rigid alignment goal is to find the face shape by simultaneously finding correspondences between model and data as well as non-rigidly deforming the model (by adjusting its parameters). The inverse rendering goal is to recover the face appearance by reformulating the image formation process and modelling each contributing factor separately (by decomposing appearance into lighting, shading and face texture).

We can write down a very general expression of the model fitting problem:

$$\min_{a,b} \mathrm{dist}(\, I, F(\, M(a), b\,)\,)$$

where the terms are defined as follows:

- a: Model parameters (e.g. face shape, texture)
- b: Extrinsic/scene parameters (e.g. pose, camera, lighting)
- M: Model
- $M(a)$: An instance of the model (e.g. a 2D shape, shape + appearance, a 3D mesh)
- I: Data (e.g. an image, a 3D scan, multiple images, features derived from an image)
- $F(M(a), b)$: A rendering function or projection – essentially, this is predicted data
- dist: A distance/cost/energy function that evaluates the difference between input and model prediction (e.g. difference between two images or distance between landmarks)

The model fitting objective can be formulated in different ways. Throughout the years, an extensive range of diverse techniques has been proposed to find an optimal way to align observed and predicted data. This vast body of literature can be divided into the following categories:

- **Rigid versus non-rigid alignment:** Rigid alignment is known as a fitting process of two elements (in our case, we could define these two as input data and target model) without any "shape change". Shape change stands for only rotation, translation and scale transformations that can be applied to the model. Inversely, this means that any shape deformation is considered as non-rigid alignment, including functions like stretching, bending or twisting.

- **Perspective versus orthographic camera model:** This categorisation is about the observation of 3D points and their projection onto 2D space. In the perspective case, the projected shape varies with the distance between camera and object where, in the orthographic case, the variation in depth is small relative to the distance from camera to object. For example, when a human face is viewed under perspective projection, the features closest to the camera (e.g. nose) appear larger and those furthest to the camera (e.g. ears) appear smaller and partially occluded.

- **Single image versus multiview:** The fitting methods can be divided by the number of input they use. Estimating parameters from a single image is a challenging task and often requires additional constraints or strong assumptions. Fitting methods that use multiple images taken from different angles can handle the occlusion better and recover shape and appearance effectively. Similarly, methods working with videos can exploit multiple observations with the help of consistency within frames.

- **Optimisation versus regression:** We can make a distinction, although not salient, based on the type of minimisation approach. In some methods, the fitting process is solved by optimising the objective function to estimate parameters whereas in other methods, this can be done by directly regressing parameters from input data which can be referred to as a learning problem.

- **Linear versus nonlinear cost function:** The complexity of the cost function is another way to classify methods. Nonlinear fitting methods usually try to optimise for combined parameters and their success depends on good initial estimate or the similarity of the model and the object. A linear approximation breaks down the problem and focuses on a particular point which can be solved efficiently and accurately.

- **Hard versus soft correspondence:** Another distinction can be made between methods that depend on an explicit correspondence between model and data and those that do not explicitly compute such a correspondence. This has a direct effect on the complexity of the cost function, meaning that the hard correspondence works on the linear and global optimum problem, where the soft correspondence deals with nonlinear optimisation.

- **Appearance-based versus geometry-based cost:** We can separate fitting methods based on the features they exploit. These include appearance-based (e.g. texture and colour) and geometry-based information (e.g. landmarks and edges).

We believe a more fundamental (and useful) distinction is between methods that use cues derived from appearance versus geometry. This has several advantages. First, the geometric cues convey direct information about the face shape. This allows us to clearly separate shape reconstruction from appearance. Second, estimating shape from geometric information alone is relatively straightforward which leads to more robust optimisation problems. Finally, using specific features helps to tackle difficult problems individually (specifically, unknown pose and expression, illumination variation and occlusion) which often occur in uncontrolled images.

The pose of an object can be seen as a rigid transformation and is often described by its translation and rotation (i.e. position and orientation). We solve the rigid alignment as a sub-problem within the model fitting procedure. Therefore, it is also important to include approaches based on rigid alignment here. Moreover, these methods potentially provide insight and inspiration to develop better techniques. Hence, we start with a compact review of some of the fundamental studies of rigid alignment in the following subsection.

2.3.1 Rigid Alignment

Object detection and recognition systems are some of the first and fundamental applications in modern computer vision. Over the decades, numerous studies have been conducted in this area, using rigid alignment especially. In one of the earliest studies [Huttenlocher and Ullman, 1987], the object was aligned to the image applying translation, rotation and scaling to correspondence points, which concludes that three points observed from a single 2D image are sufficient to determine the pose of the rigid object in 3D. [Granger et al., 2001] used rigid registration on the surfaces of teeth and jaw bone as part of their application of oral implantology.

Commonly, in the relationship between two sets of point data (from objects that can be similar, overlapped or have no relation) the scan alignment approach aims to find an ideal way to match one set to another. A very popular solution to the alignment problem is the Iterative Closest Point (ICP) algorithm. It was introduced in [Besl and McKay, 1992] with the aim of minimising distances between data and shape points. The key idea is to find matching correspondences while refining rigid-body transformations. These two steps are iterated to convergence (until the difference in the error is below a certain threshold value). [Rusinkiewicz and Levoy, 2001] investigated many variants of the ICP algorithm focusing on convergence speed and accuracy. They discussed these variants, which affect each stage of the algorithm from the selection of points to the minimisation strategy.

[Mitra et al., 2004] asserted an optimised alignment method based on applying rigid transformation for the registration of partial surface points. In the Digital Michelangelo Project [Levoy et al., 2000], 3D rigid transformation was implemented in the stage of some part of calibration and scan alignment. It was used to find matching points on two meshes and refine overlapping scans.

There are several notable examples of rigid transformations on 3D models. [Kemelmacher and Basri, 2005] presented a shape indexing method to recognise 3D objects from a single image. Their approach embedded poses and lighting conditions within their method. [Simon et al., 1994] applied the ICP algorithm to capture 3D pose estimation of a rigid object.

[Rusinkiewicz et al., 2002] proposed a model acquisition system that includes aligning acquired range images and merging them to create a 3D model. They established optimisation through small translations and rotations to eliminate the effects caused by object movement. Parallel to this study, [Liu and Heidrich, 2003] introduced a method for 3D model acquisition and registration using scanned surface data of objects.

[Lamond and Watson, 2004] presented a hybrid rendering system that integrates photogrammetric modelling techniques and laser acquired range data. Their approach uses image edges for the alignment process and view-dependent texture mapping for the rendering process to create geometrically accurate and photorealistic models.

2.3.2 Appearance Error Optimisation

The model fitting process can be achieved by minimising the difference between the template and the observed image iteratively, which is known as the analysis-by-synthesis approach. This was used for fitting 2D AAMs in a framework called the Inverse Compositional Algorithm. It is proposed by Baker and Matthews in a series of studies [Baker and Matthews, 2001, Baker and Matthews, 2004, Matthews and Baker, 2004] and extended to 3D morphable model in [Blanz and Vetter, 1999, Blanz and Vetter, 2003].

[Gleicher, 1997] introduced an Image Difference Decomposition technique based on the assumption of linear dependency between the model and the input image. The main idea is to convert the image difference into a linear combination in order to solve the registration problem in tracking.

[Romdhani and Vetter, 2003] and recently [Booth et al., 2017] extended the inverse compositional algorithm to 3D face models. By the power of optimising the cost function iteratively, both approaches are robust and accurate in their own right. However, since a 3D morphable model is a dense model, which means that it consists of thousands of vertices, the fitting process is very slow.

In such cases, when the error function is non-convex or the noisy data is used, the fitting algorithm can get stuck in *local minima*. The local minima problem can be defined as a failed situation that is locked in the local minimum while trying to detect the true (global) minimum in optimising nonlinear functions, shown in Figure 2.2.

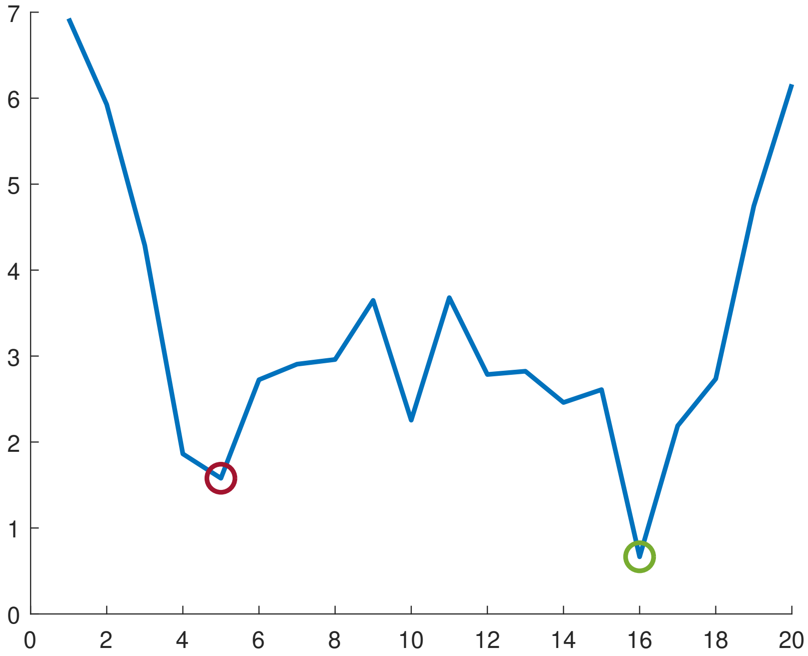


Figure 2.2: Local minima problem. When trying to minimise general nonlinear functions, algorithms may be trapped in local minima (in red), which do not correspond to the true minimum value of the function (in green).

[Keller et al., 2007] focused on recovering the 3D shape and pose of a human face from a single contour image. They demonstrated how to fit a morphable model to outer and inner contours (due to texture, shape and shadowing).

Recently, we have seen that noisily detected landmarks can be filtered using a model [Amberg and Vetter, 2011] and automatic landmark detection can be integrated into a fitting algorithm [Schönborn et al., 2013].

2.3.3 Non-rigid Alignment: Geometry Error Optimisation

Although appearance-based approaches estimate the 3D face shape to a close degree, they suffer from fundamental complications arising from interpreting appearance and modelling large variations in illumination. Instead, geometric cues such as landmarks and edges are well suited to this problem given that they are relatively insensitive to changes in illumination as well as camera parameters. Using only such 2D geometric information, model fitting can be viewed as a non-rigid, 2D/3D alignment problem. For some features, the model/image correspondences are known whereas for others they must also be estimated.

Facial landmarks are used in a number of ways in face processing, including registration and normalisation. Motivated by the recent improvements in the robustness and efficiency of 2D facial feature detectors, a number of researchers have used the position of facial landmarks in a 2D image as a cue for 3D face shape, in particular, by fitting a 3D morphable model to these detected landmarks [Blanz et al., 2004, Knothe et al., 2006, Patel and Smith, 2009]. [Breuer et al., 2008] extended the refinement process in [Blanz and Vetter, 1999] using a landmark detector to align 3D model, providing a fully automatic system. Landmarks have been shown to be sufficient for obtaining useful shape estimates in their own right [Aldrian and Smith, 2013]. Separating geometric features from illumination and reflectance effects, model fitting can be achieved efficiently.

The problem of interpreting 3D face shape from 2D landmark positions is related to the problem of non-rigid structure from motion [Hartley and Vidal, 2008]. However, in that case, the basis set describing the non-rigid deformations is unknown, but multiple views of the deforming object are available. In our case (i.e. only a single view of the face is available), the basis set is known as "face space" – represented by a 3D morphable model.

Some work has considered other 2D shape features besides landmark points. An early example of using image edges for face model fitting is ASM [Cootes et al., 1995] where a 2D boundary model is aligned to image edges. In 3D, contours have been used directly for 3D face shape estimation [Atkinson et al., 2009]. [Zhu et al., 2015] presented a method that can be seen as a hybrid of landmark and edge fitting. Fixed landmark points that define boundaries are allowed to slide over the 3D face surface during fitting. From a theoretical standpoint, [Lüthi et al., 2009] explored to what degree face shape is constrained when contours are fixed.

Shape-from-silhouette is another technique to reconstruct 3D shapes. Since silhouettes immune to internal markings (holes) and noise, accurate geometry of a face can be obtained based on shape similarity. [Moghaddam et al., 2003] was the first to use this as a cue for morphable model fitting. The 3D face shape was optimised using a boundary-weighted cost function for matching partial contour segments derived from silhouette images which are independent from lighting and texture changes. Similarly, [Cashman and Fitzgibbon, 2013] learned a 3DMM from 2D images by fitting to silhouettes.

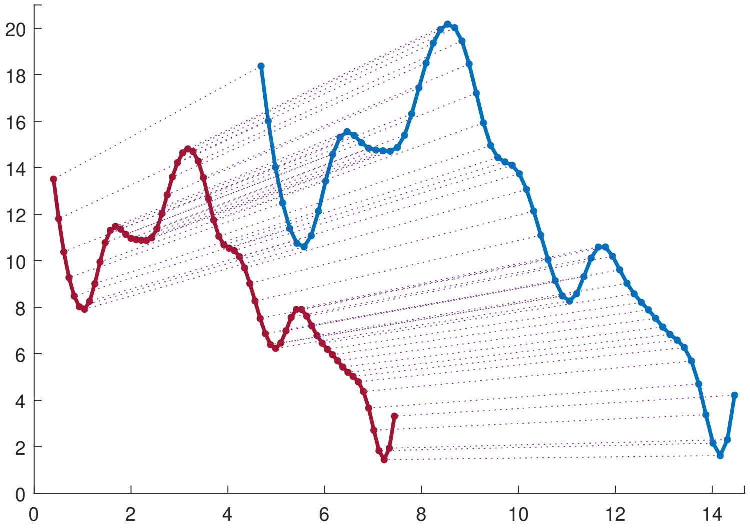


Figure 2.3: Point-to-point correspondence. The purple dotted lines indicate the mapping between one surface and another.

It is important to note a distinction related to the use of correspondence which can be defined as hard and soft correspondences. By hard correspondence, we mean a one-to-one matching process for each point on the image, shown in Figure 2.3. By soft correspondence, we mean an energy term that is being minimised that does not depend on an explicit correspondence hypothesis. This process would depend on an operator (e.g. distance transform) which provides the strongest feature. Therefore, points are not based on computing point-to-point fitting and have multiple candidate correspondences.

[Romdhani and Vetter, 2005] extended the fitting progress in [Blanz and Vetter, 2003] by combining geometry and texture based features. Cost functions of each feature; such as edges, texture constraints or specular highlights derived from the input image, were combined to obtain a smoother cost function that is easier to minimise. As part of a hybrid energy function, an edge distance cost was obtained by using texture and outer (silhouette) contours in a similar way to LM-ICP [Fitzgibbon, 2003] where correspondence between image edges and model contours is soft. This is achieved by applying a distance transform to an edge image, which provides a smoothly varying cost surface whose value at a pixel indicates the distance (and its gradient, the direction) to the closest edge. Similarly, [Amberg et al., 2007] extended this idea to multi-view setting (multiple images and videos) and smoothed the edge distance cost by averaging results with different parameters.

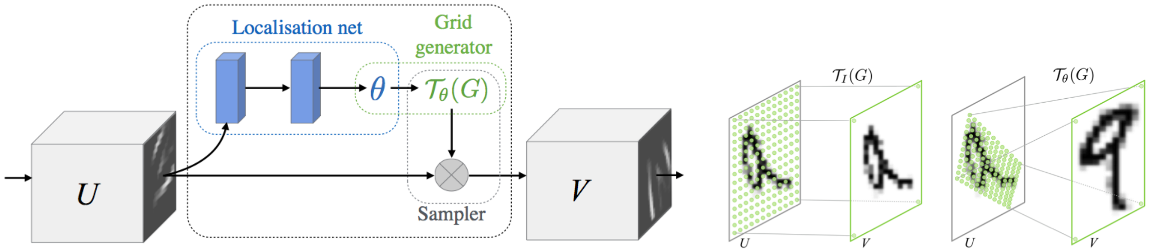
Fitting a 3DMM to a 2D image using only geometric features (i.e. landmarks and edges) is essentially a non-rigid alignment problem. Surprisingly, the idea of employing an ICP [Besl and McKay, 1992] approach with hard edge correspondences (in a similar manner to ASM fitting) has been discounted in the literature [Romdhani and Vetter, 2005].

Methods that use hard correspondences tend to be more efficient and less prone to becoming stuck in local minima. However, they also tend to be more fragile since they depend upon the correspondences being accurate. [Keller et al., 2007] showed that using soft correspondences lead to a cost function that is neither continuous nor differentiable. This suggests the optimisation method must be carefully chosen.

2.3.4 Regression-based Fitting

A recent alternative to optimisation-based approaches is to learn a regressor directly from face images or features. Many of these are based on deep learning (and are covered in the next section). However, some use more traditional machine learning approaches. Of particular relevance, [Sánchez-Escobedo et al., 2016] learned a regressor from extracted face contours to 3DMM shape parameters. They estimated 3D face surface based on the connection between multi-view 2D contours and corresponding 2D pixel projections of 3D vertices around the occluding boundaries. In a similar manner to landmarks, local features can be used to aid the fitting process [Huber et al., 2015]. They used a learning-based cascaded regression method to simultaneously estimate shape and pose parameters by directly learning the gradient direction from data.

Several recent works [Cao et al., 2013, Cao et al., 2014a, Saito et al., 2016] used landmark fitting to generate ground truth to train a direct image-to-shape parameters regressor. Again, the landmark fitting optimisation is performed using alternating minimisation, this time under perspective projection with a given focal length. Of interesting relevance to the subsequent discussion on perspective ambiguity (Section 2.5), [Cao et al., 2014a] explicitly noted that varying the focal length leads to different shapes and used binary search to find the one that gives the lowest residual error.



(a) (b)

Figure 2.4: (a) Overview of a spatial transformer module (a localiser network, a grid generator and a sampler). (b) Two examples of applying the parameterised sampling grid to the image U producing the output V. Identity transformation (left) and affine transformation (right) [Jaderberg et al., 2015].

2.4 Connections Between Geometry and Deep Learning

Establishing correspondence between images of objects from the same class is a fundamental task in computer vision. It enables appearance to be disentangled from the effects of pose and shape deformation, simplifying the task of comparing objects. Likewise, the accuracy of the recognition can be improved by aligning the objects to similar pose.

Recently, with increased processing abilities and the availability of high scale data, deep learning architectures have achieved state-of-the-art performance in several benchmarks ranging from handwritten digit classification [LeCun et al., 1998] to object recognition [Russakovsky et al., 2015] to face recognition [Huang et al., 2007]. The general problem of alignment has been approached from a learning perspective previously [Huang et al., 2012]. The specific question of whether convolutional neural networks learn a notion of correspondence implicitly has been considered in [Long et al., 2014].

In a lot of applications, the process of pose normalisation and object recognition are disjoint. For example, in the breakthrough deep learning face recognition paper, DeepFace [Taigman et al., 2014] used a 3D mean face as preprocessing before feeding the pose-normalised image to a CNN. We now examine related studies to solve this problem in the following subsection.

2.4.1 Spatial Transformer Networks

The Spatial Transformer Networks (STN) [Jaderberg et al., 2015] aimed to combine these two processes into a single network that is trainable end-to-end. The localiser network estimated a 2D affine transformation that was applied to the regular output grid, meaning the network could only learn a fairly restricted space of transformations. The overview of a spatial transformer module and an illustration of how it works are shown in Figure 2.4.

They also proposed the concept of a 3D transformer, which takes 3D voxel data as input, applies 3D rotation and translation, and outputs a 2D projection of the transformed data. Working with 3D (volumetric data) removes the need to model occlusion or camera projection parameters.

A number of subsequent works were inspired by the original STN. [Yan et al., 2016] used an encoder-decoder architecture in which the encoder estimates a 3D volumetric shape from an image and is trained by combining with a decoder which uses a perspective transformer network to compute a 2D silhouette loss. [Handa et al., 2016] presented the geometric vision with neural networks (gvnn) toolbox that has layers that explicitly implement 3D geometric transformations. Their goal was to use 3D transformations in low level vision tasks such as relative pose estimation.

[Chen et al., 2016] used a spatial transformer that applies a 2D similarity transform as part of an end-to-end network for face detection. [Henriques and Vedaldi, 2017] applied a spatial warp prior to convolutions such that the convolution result is invariant to a class of two-parameter spatial transformations. [Yu et al., 2016] incorporated a parametric shape model, though their basis was 2D (and trainable), modelled only sparse shape and combined pose and shape into a single basis. They used a second network to locally refine position estimates and train end-to-end to perform landmark localisation.

[Wu et al., 2017] applied recursive STN in the context of face recognition. [Dai et al., 2017] introduced Deformable Convolutional Networks based on the idea of augmenting the spatial sampling locations in the modules with additional offsets and then learning the offsets from target tasks without additional supervision. [Bhagavatula et al., 2017] fitted a generic 3D face model and estimated 2D face landmarks in a 3D-aware fashion, though they required known landmarks for training. [Zhong et al., 2017] implemented the spatial transformer module for alignment learning in end-to-end face recognition.

2.4.2 Supervised CNN Regression

As covered in Section 2.3, the task of fitting a 3DMM to a single image has traditionally been posed as a problem of analysis-by-synthesis and solved by optimisation. The original method [Blanz and Vetter, 1999] used stochastic gradient descent to minimise an appearance error, regularised by statistical priors. Subsequent work used a more complex feature-based objective function [Romdhani and Vetter, 2005] and the state-of-the-art method used Markov Chain Monte Carlo for probabilistic image interpretation [Schönborn et al., 2017].

Analysis-by-synthesis approaches are computationally expensive, prone to convergence on local minima and fragile when applied to in-the-wild images (as noted in Section 2.3). For this reason, there has been considerable recent interest in using CNNs to directly regress 3DMM parameters from images. The majority of such work is based on supervised learning.

[Jourabloo and Liu, 2016] fitted a 3DMM to detected landmarks and then trained a CNN to directly regress the fitted pose and shape parameters. [Tran et al., 2017] used a recent multi-image 3DMM fitting algorithm [Piotraschke and Blanz, 2016] to obtain pooled 3DMM shape and texture parameters (i.e. the same parameters for all images of the same subject). They then trained a CNN to directly regress these parameters from a single image. They did not estimate pose and hence did not compute an explicit correspondence between the model and image.

[Kim et al., 2018] went further by also regressing illumination parameters (effectively performing inverse rendering) though they trained on synthetic, rendered images (using a breeding process to increase diversity). They estimated a 3D rotation but relied on precisely cropped input images such that scale and translation is implicit.

[Richardson et al., 2016] also trained on synthetic data though they used an iteratively applied network architecture and a shape-from-shading refinement step to improve the geometry. [Jackson et al., 2017] regressed shape directly using a volumetric representation.

[Kanazawa et al., 2016] proposed a weakly supervised architecture that learns correspondences by shape deformation in a fine-grained dataset. [Rocco et al., 2017] proposed a neural network for geometric matching between two images by estimating an affine transformation that strongly relies on fully supervised training.

[Ranjan et al., 2017a, Ranjan et al., 2017b] proposed a multi-task architecture that simultaneously predicts landmark locations, estimates pose and identifies faces.

The DenseReg [Güler et al., 2017] approach used fully convolutional networks to directly compute dense correspondence between a 3D model and a 2D image. The network did not explicitly estimate or model 3D pose or shape (though these are implied by the correspondence) and was trained by using manually annotated 2D landmarks to warp a 3D template onto the training images (providing the supervision).

[Sela et al., 2017] also used a fully convolutional network to predict correspondence and also depth. They then merged the model-based and data-driven geometries for improved quality.

The weakness of all of these supervised approaches is that they require labelled training data (i.e. images with fitted morphable model parameters). If the images are real-world images, then the parameters must come from an existing fitting algorithm, in which case the best the CNN can do is learn to replicate the performance of an existing algorithm. If the images are synthetic with known ground truth parameters, then the performance of the CNN on real-world input is limited by the realism and variability present in the synthetic images. Alternatively, we must rely on 3D supervision provided by multiview or depth images, in which case the available training data is vastly reduced.

2.4.3 Unsupervised CNN Regression

[Thewlis et al., 2017a, Thewlis et al., 2017b] followed an unsupervised approach to establish correspondence between different object instances in a category by learning a sparse set of landmarks and later a dense model. [Richardson et al., 2017] took a step towards removing the need for labels by presenting a semi-supervised approach. They still relied on supervised training for learning 3DMM parameter regression but then refined the coarse 3DMM geometry using a second network that is trained in an unsupervised manner.

Very recently, [Tewari et al., 2017] presented MoFA, a completely unsupervised approach for training a CNN to regress 3DMM parameters, pose and illumination using an autoencoder architecture. The regression is done by the encoder CNN. The decoder then uses a hand-crafted differentiable renderer to synthesise an image. The unsupervised loss is the error between the rendered image and the input, with convergence aided by losses for priors and landmarks. Note that the decoder is exactly equivalent to the differentiable cost function used in classical analysis-by-synthesis approaches. Presumably, the issues caused by the non-convexity of this cost function are reduced in a CNN setting since the gradient is averaged over many images.

While the ability of [Tewari et al., 2017] to learn from unlabelled data is impressive, there are a number of limitations. The complexity required to enable the hand-crafted decoder to produce photorealistic images of any face under arbitrary real-world illumination, captured by a camera with arbitrary geometric and photometric properties, is huge. Arguably, this has not yet been achieved in computer graphics. Moreover, the 3DMM texture should only capture intrinsic appearance parameters such as diffuse and specular albedo (or even spectral quantities to ensure independence from the camera and lighting). Such a model is not currently available.

2.5 Ambiguities in the Interpretation of 3D Face Shape

The projection of a 3D face to 2D results in a loss of information (e.g. occlusion and depth information) so that the 3D shape cannot be unambiguously reconstructed. Hence, existing methods rely on some mode of prior knowledge or visual cues to select from the space of possible solutions. For example, a statistical model provides constraints on shape and texture variation or, more generally, faces are approximately bilaterally symmetric in shape and (ignoring the effects of illumination) appearance. Nevertheless, there are many interpretations of monocular 2D geometric information which vary significantly in 3D shape. Modelling these ambiguities is important for understanding the uncertainty in face shape estimations.

2.5.1 Flexibility of Partial Fixed Models

Shape-from-correspondence partially constrains a 3D face shape estimate. Knowing the 2D projection of a point only constrains its 3D position to any point along the camera projection ray. This can be viewed as a partial constraint on the 3D shape, but flexibility may remain and this leads to ambiguities in shape estimation.

The problem of describing the flexibility in a statistical shape model that is partially fixed has been considered before in 3D. The idea is to characterise the space of shapes that approximately fit the given observations of the 3D position of some points, curves or subsets of the surface. [Albrecht et al., 2008] proposed an efficient solution to this problem using a generalised Eigenvalue decomposition. Of particular relevance, they computed the subspace of faces with a fixed 3D profile. [Lüthi et al., 2009] extended this approach into a fully probabilistic setting.

2.5.2 Faces Under Perspective Projection

The majority of 3D model fitting methods to 2D images assume an affine camera (such as scaled orthographic or "weak perspective") [Blanz et al., 2004, Knothe et al., 2006, Patel and Smith, 2009, Aldrian and Smith, 2013]. Such a camera does not introduce any nonlinear perspective transformation. This means that the effects of perspective must be taken into account for any situation where a face may be viewed from a small distance (particularly common due to the popularity of the "selfie" format).

[Smith, 2016] delivered a useful and thorough overview on this issue. For the purposes of this subsection, we refer to parts of its review as it provides a clear description of the problem. There are several studies from psychology, art history and computer science that focus on the effect of perspective. In psychology, [Liu and Chaudhuri, 2003] found that there is a strong dependence of human face recognition performance on perspective transformation. [Perona, 2007, Bryan et al., 2012] considered another effect, concluding that perspective distortion impacts the social perception of faces. In the context of art history, [Latto and Harper, 2007] investigated how subject-artist distance leads to perspective distortion as well as how subject-camera distance influences estimations of body weight from face images.

Two recent papers addressed the estimation problem of subject-camera distance from monocular, perspective views of a face. They observed that the structure of projected 2D features reveals information about perspective transformation and this can be used for estimating the distance. [Flores et al., 2013] recovered the distance using exemplar 3D face models under the assumption of a calibrated camera with known intrinsic parameters. The estimation is based on the general distribution of landmarks across subjects while adjusting the focal length which keeps face silhouette at a constant size. [Burgos-Artizzu et al., 2014] proposed an automatic method for estimating subject-camera distance from a single 2D frontal image. Their approach does not require a 3D shape model or any prior knowledge on camera or subject. The method, which is based on a regressor that mapped face shapes at different distances, measures the changes in the position of a number of facial landmarks over distance. They also noted that human observers are bad at estimating the distance from a single image while the method does so precisely.

[Fried et al., 2016] explored the effect of perspective in a synthesis application. They used a 3D head model to compute a 2D warp to simulate the effect of changing the subject-camera distance, allowing them to approximate appearance at any distance given a single image. [Valente and Soatto, 2015] also proposed a method to warp a 2D image to compensate for perspective. However, their goal was to improve the performance of face recognition systems that they showed are sensitive to such transformations.

2.5.3 Other Ambiguities

There are other known ambiguities in the monocular estimation of 3D shape. The bas relief ambiguity [Belhumeur et al., 1999] arises in photometric stereo with unknown light source directions. A continuous class of surfaces (differing by a linear transformation) can produce the same set of images when an appropriate transformation is applied to the illumination and albedo. For the particular case of faces, [Georghiades et al., 2001] took advantage of symmetries and similarities to resolve this ambiguity. [Hill and Bruce, 1994] interpreted shaded images of concave faces as convex faces with inverted illumination. This is a binary version of the bas relief ambiguity, occurring when both convex and concave faces are interpreted as convex so as to be consistent with prior knowledge.

More generally, ambiguities in surface reconstruction have been considered in a number of settings. [Ecker et al., 2008] considered the problem of reconstructing a smooth surface from local information that contains a discrete ambiguity. The ambiguities studied here are in the local surface orientation or gradient, a problem that occurs in photometric shape reconstruction. [Moreno-Noguer and Fua, 2013] used stochastic sampling to explore the set of possible solutions to non-rigid, monocular shape reconstruction. They attempted to select from within this space using additional information provided by motion or shading. [Salzmann et al., 2007] studied the ambiguities that arise in monocular non-rigid structure from motion under perspective projection.

2.5.4 Deep Face Correspondence

A very recent trend has been to train image-to-image CNNs to directly estimate correspondence between a 3DMM and a 2D face image. Unlike landmarks, this correspondence is dense, providing a 2D location for every visible vertex. This was first proposed by [Güler et al., 2017] who used a fully convolutional network. [Yu et al., 2017] took a similar approach but went further by using the correspondences to estimate 3D face shape by fitting a 3DMM. [Sela et al., 2017] took a multitask learning approach by training a CNN to predict both correspondence and facial depth. In all cases, this estimated dense correspondence provides an ambiguous shape cue, for the reasons described above.

2.6 Face Datasets

Throughout this thesis, we use several datasets in order to evaluate our algorithms both quantitatively and qualitatively. In this section, we would like to introduce these datasets.

We use synthetic rendering of faces supplied with the Basel Face Model (BFM) [Paysan et al., 2009]. This is particularly useful because it allows us to evaluate surface error by computing distance between ground truth and reconstructed meshes.

The CMU Pose, Illumination, and Expression (PIE) database [Sim et al., 2003] is famous for its use on face recognition studies. It includes images of subjects with various pose, illumination and expression conditions under controlled environments. We use images from this dataset for the frontalisation experiment in Chapter 3.

We use the Labeled Faces in the Wild (LFW) dataset [Huang et al., 2007] to visualise model fitting in an uncontrolled setting. Although LFW is a relatively small dataset, it is one of the well-studied face recognition benchmarks in unconstrained environments.

FaceWarehouse [Cao et al., 2014b] is a comprehensive 3D facial expression database that contains 150 subjects from various ethnic backgrounds. It provides each image with a corresponding ground truth model that enables us to compare model fitting algorithms in a realistic setting in Chapter 3. Because it has 3D face meshes for each subject and expression, shape and expression models can be built by applying PCA.

The Annotated Facial Landmarks in the Wild (AFLW) dataset [Martin Koestinger and Bischof, 2011] provides a large-scale image collection with a wide variety in appearance. Each image is accompanied by 21 annotated landmarks. In Chapter 4, we train our 3DMM-STN network on this dataset.

UMDFaces [Bansal et al., 2017] contains face images of famous people with corresponding identity and estimated 21 landmarks. We use the UMDFaces dataset for testing the 3DMM-STN network since it provides multiple images of the same person. Moreover, we train and test our StaTN network on this dataset in Chapter 4.

We use the CAT dataset [Zhang et al., 2008] to demonstrate transfer learning on unsupervised data. It includes a large number of cat images which is provided for training and evaluation of detection algorithms.

The Caltech Multi-Distance Portraits (CMDP) dataset [Burgos-Artizzu et al., 2014] provides frontal portraits captured from seven distances spanning the typical range between camera and subject. The dataset itself displays the effect of perspective distortion and the deformation in the image plane related to distance. We present geometric ambiguities on real images from this dataset in Chapter 5.

2.7 Conclusions

In this chapter, we presented a systematic literature review to describe crucial techniques that have been developed over the years. We dwelt on the optimisation- and learning-based model fitting problem and its limitations.

From this review, we can draw a number of conclusions:

- Appearance-based methods provide a full explanation of a face's appearance. However, they require camera calibration, manual initialisation or estimation of combined parameters (e.g. pose, shape, albedo, illumination). Existing methods make strong assumptions about illumination or reflectance or involve solving a complex nonlinear optimisation problem with no guarantee of finding the global minimum. Moreover, they are typically extremely computationally expensive. For example, [Schönborn et al., 2017] require 10,000 samples to produce a good estimate of the posterior.

- Geometric cues hold certain benefits for face shape recovery. Separating the shape reconstruction from appearance has shown promise, avoiding the potentially fragile process of modelling as well as estimating appearance and illumination properties. In some cases, coarse shape fitting can be established using only very sparse landmark information. Edges also provide a strong cue for establishing correspondence, directly conveying information about 3D pose and shape.

- Deep learning techniques are showing great promise for direct regression of face shape parameters. However, they only work effectively with either supervised training or clever self supervision costs. There are clear disadvantages of supervised training. First, it requires labelled dataset. Second, if the supervision includes annotated data, its quality has a direct effect on the performance of the network.

- A spatial transformer within a CNN explicitly estimates an affine transformation and resamples a specific part of the input image to a fixed-size output image. This gives the network the ability to explicitly compensate the effects of pose and, to some extent, non-rigid deformations. By exploiting a model of geometric transformation, the amount of training data and the complexity of the network can be vastly reduced.

- Ambiguities in face shape estimation have been almost completely ignored in all model fitting methods. Although a limited amount of studies has investigated the effects of perspective by estimating the subject-camera distance, the problem of interpreting 3D face shape from ambiguous cues has not been explored in detail.

Chapter 3

Optimisation-based Fitting Using Geometric Features

3.1 Introduction

Estimating 3D face shape from one or more 2D images is a longstanding problem in computer vision. It has a wide range of applications from pose-invariant face recognition [Blanz and Vetter, 2003] to creation of 3D avatars from 2D images [Ichim et al., 2015]. One of the most successful approaches to this problem is to use a statistical model of 3D face shape [Blanz and Vetter, 1999]. This transforms the problem of shape estimation to one of model fitting and provides a strong statistical prior to constrain the problem.

Image edges convey important information about a face. The occluding boundary provides direct information about 3D shape, for example a profile view reveals strong information about the shape of the nose. Internal edges, caused by texture changes, high curvature or self-occlusion, provide information about the position and shape of features such as lips, eyebrows and the nose. This information provides a cue for estimating 3D face shape from 2D images or, more generally, for fitting face models to images.

In this chapter, we present a fully automatic algorithm for fitting to image edges with hard correspondence. By hard correspondence, we mean that an explicit correspondence is computed between projected model vertex and edge pixel. For comparison, we describe our variant of previous methods [Romdhani and Vetter, 2005, Amberg et al., 2007, Keller et al., 2007] that fit to edges using soft correspondence. By soft correspondence, we mean that an energy term that captures many possible edge correspondences is minimised. We present quantitative and qualitative evaluations on both synthetic and real images.

3.2 Contributions

Recent face shape estimation methods are able to obtain considerably higher quality results than the purely model-based approaches. They do so by using pixel-wise shading or motion information to apply finescale refinement to an initial shape estimate. For example, [Suwajanakorn et al., 2014] used photo collections to build an average model of an individual which was later fitted to a video and added finescale detail by optical flow and shape-from-shading. [Cao et al., 2015] took a machine learning approach and trained a regressor that predicts high resolution shape detail from local appearance.

Our aim in this chapter is not to compete directly with these methods. Specifically, we do not claim to achieve qualitatively or quantitatively better shape estimation results. Since our methods are model-based, their accuracy is limited by the generalisation capability of the model. However, our methods do not require any training data, black box regressors or learning. Furthermore, compared to analysis-by-synthesis approaches, our methods make no assumptions about appearance and illumination. The objective is to explore the potential accuracy of a face model that can be built from a single input image using solely sparse, geometric information. The output of our method may provide a better initialisation for state-of-the-art refinement techniques or remove the need to have a person specific model.

As mentioned in Chapter 2, fitting a 3D morphable model to a 2D image using only geometric features (i.e. landmarks and edges) is essentially a non-rigid alignment problem. Surprisingly, the idea of employing an iterated closest point approach with hard edge correspondences has been discounted in the literature [Romdhani and Vetter, 2005]. In this chapter, we pursue this idea and develop an iterative 3DMM fitting algorithm that is fully automatic, simple and efficient. Instead of working in a transformed distance-to-edge space, we compute an explicit correspondence between model and image edges. This allows us to treat the model edge vertices as a landmark with known 2D position, for which optimal pose or shape estimates can be easily computed.

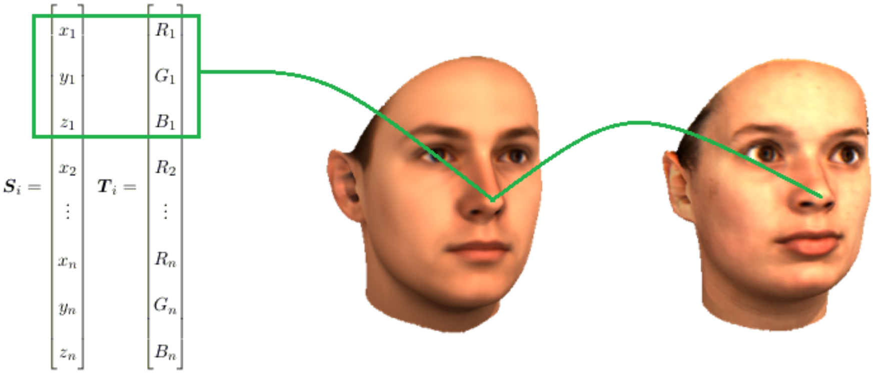


Figure 3.1: Shape and texture vector representation. Vertices at the same position in each mesh correspond to the texture at the same location on the face.

We also observe later in this chapter that, under both orthographic and perspective projection, model fitting can be posed as a separable nonlinear least squares optimisation problem that can be solved efficiently without requiring any problem specific optimisation method, initialisation or parameter tuning.

3.3 3D Morphable Model

In Subsection 2.2.2, we briefly describe 3D face modelling approaches. Here, we focus on the model construction and its main characteristics.

A 3D morphable model is a statistical representation of a set of 3D faces and every face can be represented by shape ($\mathbf{S} \in \mathbb{R}^{3N}$) and texture ($\mathbf{T} \in \mathbb{R}^{3N}$) vectors. One can define the geometry of a face as:

$$\mathbf{S}_i = (x_1, y_1, z_1, x_2, ..., x_n, y_n, z_n)^{\mathrm{T}}, \tag{3.1}$$

where x, y and z are the coordinates of its n vertices. Similarly, the texture of a face can be defined as:

$$\mathbf{T}_i = (R_1, G_1, B_1, R_2, ..., R_n, G_n, B_n)^{\mathrm{T}}, \tag{3.2}$$

where R, G and B are the colour values of the n corresponding vertices. Shape and texture vectors and their correspondence between faces are shown in Figure 3.1.

Shape and texture are described by a linear subspace model learnt from data using principal component analysis. In other words, by applying PCA to the data matrix stacked with meshes, separate shape and texture models can be obtained which each includes the mean and a set of principal components (plus the standard deviation $\sigma \in \mathbb{R}^D$). In this thesis, we only work with the shape model. The shape model comprises the mean shape $\bar{\mathbf{s}} \in \mathbb{R}^{3N}$ and D principal components $\mathbf{Q} \in \mathbb{R}^{3N \times D}$. (Accordingly, the texture model would have similar expressions.) We can approximate the shape of any object from the same class as the training data as:

$$\mathbf{s}(\boldsymbol{\alpha}) = \mathbf{Q}\boldsymbol{\alpha} + \bar{\mathbf{s}}, \tag{3.3}$$

where the vector $\mathbf{s}(\boldsymbol{\alpha}) \in \mathbb{R}^{3N}$ contains the coordinates of the N vertices, stacked to form a long vector: $\mathbf{s} = [u_1, v_1, w_1, \ldots, u_N, v_N, w_N]^{\mathrm{T}}$ determined by the shape parameters $\boldsymbol{\alpha} \in \mathbb{R}^D$. Hence, the ith vertex is given by: $\mathbf{v}_i = [s_{3i-2}, s_{3i-1}, s_{3i}]^{\mathrm{T}}$.

For convenience, we denote the sub-matrix corresponding to the ith vertex as $\mathbf{Q}_i \in \mathbb{R}^{3\times D}$ and the corresponding vertex in the mean face shape as $\bar{\mathbf{s}}_i \in \mathbb{R}^3$, such that the ith vertex is given by:

$$\mathbf{v}_i = \mathbf{Q}_i\boldsymbol{\alpha} + \bar{\mathbf{s}}_i. \tag{3.4}$$

Similarly, we define the row corresponding to the u component of the ith vertex as $\mathbf{Q}_{iu}$ (and v and w) and define the u component of the ith mean shape vertex as $\bar{s}_{iu}$ (and v and w).

3.4 Projection Models

As introduced in Section 2.5, the vast majority of 2D face analysis methods that involve estimation of 3D face shape assume an affine camera (such as scaled orthographic or "weak perspective") [Blanz et al., 2004, Knothe et al., 2006, Patel and Smith, 2009, Aldrian and Smith, 2013]. This would be suitable for applications where the subject-camera distance is likely to be large, however, any situation where a face may be viewed from a small distance must account for the effects of perspective.

In this chapter, our approach is based on fitting a 3DMM to face images under the assumption of a scaled orthographic projection. We mainly focus on the perspective projection in Chapter 5 where we examine the ambiguity arising from the effect of perspective transformation. However, in order to maintain cohesion, we present both orthographic and perspective camera models to the reader in this section.

3.4.1 Scaled Orthographic Projection

The scaled orthographic, or weak perspective, projection model assumes that variation in depth over the object is small relative to the mean distance from camera to object. Under this assumption, the projection of a 3D point $\mathbf{v} = [u, v, w]^{\mathrm{T}}$ onto the 2D point $\mathbf{x} = [x, y]^{\mathrm{T}}$ is given by $\mathbf{x} = \mathbf{SOP}[\mathbf{v}, \mathbf{R}, \mathbf{t}_{2\mathrm{d}}, s] \in \mathbb{R}^2$ which does not depend on the distance of the point from the camera, but only on a uniform scale s given by the ratio of the focal length of the camera and the mean distance from camera to object:

$$\mathbf{SOP}[\mathbf{v}, \mathbf{R}, \mathbf{t}_{2\mathrm{d}}, s] = s\mathbf{P}\mathbf{R}\mathbf{v} + s\mathbf{t}_{2\mathrm{d}}, \tag{3.5}$$

where

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.6}$$

is a projection matrix and the pose parameters $\mathbf{R} \in SO(3)$, $\mathbf{t}_{2\mathrm{d}} \in \mathbb{R}^2$ and $s \in \mathbb{R}^+$ are a rotation matrix, 2D translation and scale respectively. In order to constrain optimisation to valid rotation matrices, we parameterise the rotation matrix by an axis-angle vector $\mathbf{R}(\mathbf{r})$ with $\mathbf{r} \in \mathbb{R}^3$. The conversion from an axis-angle representation to a rotation matrix is given by:

$$\mathbf{R}(\mathbf{r}) = \cos\theta\mathbf{I} + \sin\theta \left[\bar{\mathbf{r}}\right]_{\times} + (1 - \cos\theta)\bar{\mathbf{r}}\bar{\mathbf{r}}^{\mathrm{T}}, \tag{3.7}$$

where $\theta = \|\mathbf{r}\|$ and $\bar{\mathbf{r}} = \mathbf{r}/\|\mathbf{r}\|$ and

$$\left[\mathbf{a}\right]_{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{3.8}$$

is the cross product matrix.

3.4.2 Perspective Projection

The nonlinear perspective projection of the 3D point $\mathbf{v} = [u, v, w]^{\mathrm{T}}$ onto the 2D point $\mathbf{x} = [x, y]^{\mathrm{T}}$ is given by the pinhole camera model $\mathbf{x} = \mathbf{pinhole}[\mathbf{v}, \mathbf{K}, \mathbf{R}, \mathbf{t}_{\mathrm{3d}}] \in \mathbb{R}^2$ where $\mathbf{R} \in SO(3)$ is a rotation matrix and $\mathbf{t}_{\mathrm{3d}} = [t_x, t_y, t_z]^{\mathrm{T}}$ is a 3D translation vector which relate model and camera coordinates (the extrinsic parameters). The matrix:

$$\mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

contains the intrinsic parameters of the camera, namely the focal length f and the principal point (c_x, c_y). We assume that the principal point is known (often the centre of the image is an adequate estimate) and parameterise the intrinsic matrix by its only unknown $\mathbf{K}(f)$. Note that varying the focal length amounts only to a uniform scaling of the projected points in 2D. This corresponds exactly to the scenario in Figure 5.1 in Chapter 5. There, subject-camera distance was varied before rescaling each image such that the interocular distance was constant, effectively simulating a lack of calibration information. This nonlinear projection can be written in linear terms by using homogeneous representations $\tilde{\mathbf{v}} = [u, v, w, 1]^{\mathrm{T}}$ and $\tilde{\mathbf{x}} = [x, y, 1]^{\mathrm{T}}$:

$$\gamma\tilde{\mathbf{x}} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}_{\mathrm{3d}}\end{bmatrix}\tilde{\mathbf{v}}, \tag{3.10}$$

where γ is an arbitrary scaling factor.

3.5 Shape from Correspondence

We begin by showing how to fit a morphable model to L observed 2D positions $\mathbf{x}_i = [x_i \ y_i]^{\mathrm{T}}$ $(i = 1 \dots L)$ arising from the projection of corresponding vertices in the morphable model. Without loss of generality, we assume that the ith 2D position corresponds to the ith vertex in the morphable model. The objective of fitting a morphable model to these observations is to obtain the shape and pose parameters that minimise the reprojection error, E_{lmk}, between observed and predicted 2D positions:

$$E_{\mathrm{lmk}}(\boldsymbol{\alpha}, \mathbf{R}, \mathbf{t}_{\mathrm{2d}}, s) = \frac{1}{L} \sum_{i=1}^{L} \|\mathbf{x}_i - \mathbf{SOP}\left[\mathbf{Q}_i \boldsymbol{\alpha} + \bar{\mathbf{s}}_i, \mathbf{R}, \mathbf{t}_{\mathrm{2d}}, s\right] \|^2. \tag{3.11}$$

The scale factor in front of the summation makes the magnitude of the error invariant to the number of landmarks. This problem is multilinear in the shape parameters and the SOP transformation matrix. It is also nonlinearly constrained, since $\mathbf{R}$ must be a valid rotation matrix. Although minimising E_{lmk} is a non-convex optimisation problem, a good initialisation can be obtained using alternating linear least squares and this estimate subsequently refined using nonlinear optimisation.

3.5.1 Landmarks

We use landmarks both for initialisation and as part of our overall objective function as one cue for shape estimation. We apply a facial landmark detector that is suitable for operating on in-the-wild images. This provides approximate positions of facial landmarks for which we know the corresponding vertices in the morphable model. We use these landmark positions to make an initial estimate of the pose and shape parameters. Note that any facial landmark detector can be used at this stage. In our experiments, we show results with a recent landmark detection algorithm [Zhu and Ramanan, 2012] that achieves state-of-the-art performance and for which code is provided by the authors.

3.5.2 Pose Estimation

We make an initial estimate of $\mathbf{R}$, $\mathbf{t}_{\mathrm{2d}}$ and s using a simple extension of the POS algorithm [Dementhon and Davis, 1995]. Compared to POS, we additionally enforce that $\mathbf{R}$ is a valid rotation matrix. We begin by solving an unconstrained system in a least squares sense. We stack two copies of the 3D points in homogeneous coordinates:

$$\mathbf{A}_{2i-1} = [u_i \ v_i \ w_i \ 1 \ 0 \ 0 \ 0 \ 0] \quad \text{and} \quad \mathbf{A}_{2i} = [0 \ 0 \ 0 \ 0 \ u_i \ v_i \ w_i \ 1] \tag{3.12}$$

and form a long vector of the corresponding 2D points:

$$\mathbf{d} = [x_1 \ y_1 \ \cdots \ x_L \ y_L]^{\mathrm{T}}. \tag{3.13}$$

We then solve for $\mathbf{k} \in \mathbb{R}^8$ in $\mathbf{Ak} = \mathbf{d}$ using linear least squares. We define $\mathbf{r}_1 = [k_1\ k_2\ k_3]$ and $\mathbf{r}_2 = [k_5\ k_6\ k_7]$. Scale is given by $s = (\|\mathbf{r}_1\| + \|\mathbf{r}_2\|)/2$ and the translation vector by $\mathbf{t}_{\text{2d}} = [k_4/s\ k_8/s]^{\text{T}}$. We perform singular value decomposition on the matrix formed from $\mathbf{r}_1$ and $\mathbf{r}_2$:

$$\mathbf{USV}^{\text{T}} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_1 \times \mathbf{r}_2 \end{bmatrix}. \tag{3.14}$$

The rotation matrix is given by $\mathbf{R} = \mathbf{UV}^{\text{T}}$. If $\det(\mathbf{R}) = -1$ then we negate the third row of $\mathbf{U}$ and recompute $\mathbf{R}$. This guarantees that $\mathbf{R}$ is a valid rotation matrix. This approach gives a good initial estimate which we subsequently refine with nonlinear optimisation of E_{lmk} with respect to $\mathbf{R}$, $\mathbf{t}_{\text{2d}}$ and s.

3.5.3 Shape Estimation

With a fixed pose estimate, shape parameter estimation under scaled orthographic projection is a linear problem. The 2D position of the ith vertex as a function of the shape parameters is given by:

$$s\mathbf{R}_{1..2}(\mathbf{Q}_i\boldsymbol{\alpha} + \bar{\mathbf{s}}_i) + s\mathbf{t}_{\text{2d}}. \tag{3.15}$$

Hence, each observed vertex adds two equations to a linear system. Concretely, for each image we form the matrix $\mathbf{C} \in \mathbb{R}^{2L\times D}$ where

$$\begin{aligned} \mathbf{C}_{2i-1} &= s(\mathbf{R}_{11}\mathbf{Q}_{iu}^{\text{T}} + \mathbf{R}_{12}\mathbf{Q}_{iv}^{\text{T}} + \mathbf{R}_{13}\mathbf{Q}_{iw}^{\text{T}}) \\ \mathbf{C}_{2i} &= s(\mathbf{R}_{21}\mathbf{Q}_{iu}^{\text{T}} + \mathbf{R}_{22}\mathbf{Q}_{iv}^{\text{T}} + \mathbf{R}_{23}\mathbf{Q}_{iw}^{\text{T}}) \end{aligned} \tag{3.16}$$

and vector $\mathbf{h} \in \mathbb{R}^{2L}$ where

$$\mathbf{h}_{2i-1} = x_i - s(\mathbf{R}_1\bar{\mathbf{s}}_i + \mathbf{t}_{\text{2d1}}) \quad \text{and} \quad \mathbf{h}_{2i} = y_i - s(\mathbf{R}_2\bar{\mathbf{s}}_i + \mathbf{t}_{\text{2d2}}). \tag{3.17}$$

We solve $\mathbf{C}\boldsymbol{\alpha} = \mathbf{h}$ in a least squares sense subject to an additional constraint to ensure plausibility of the solution. We follow [Brunton et al., 2014] and use a hyperbox constraint on the shape parameters. This avoids having to choose a regularisation weight but ensures that each parameter lies within σ standard deviations of the mean by introducing a constraint on the shape parameters (we use $\sigma = 3$ in our experiments). Hence, the problem can be solved in closed form as an inequality constrained linear least squares problem.

3.5.4 Nonlinear Refinement

Having alternated pose and shape estimation for a fixed number of iterations, finally we perform nonlinear optimisation (using the trust-region-reflective algorithm [Coleman and Li, 1996]) of E_{lmk} over $\boldsymbol{\alpha}$, $\mathbf{R}$, $\mathbf{t}_{\text{2d}}$ and s simultaneously. We represent $\mathbf{R}$ in axis-angle space to ensure that it remains a valid rotation matrix and we retain the hyperbox constraint on $\boldsymbol{\alpha}$.

3.6 Shape from Contours

The method in Section 3.5 enables a 3DMM to be fitted to 2D landmark positions if the correspondence between landmarks and model vertices is known. Edges, for example caused by occluding boundaries, do not have a fixed correspondence to model vertices. Hence, fitting to edges requires shape and pose estimation to happen in conjunction with establishing correspondence between image and model edges. Our proposed approach establishes these correspondences explicitly by finding the closest image edge to each model boundary vertex (subject to additional filtering to remove unreliable matches). Our method comprises the following steps:

1. Initialise shape and pose estimates by fitting to landmarks only.
2. Improve initialisation using iterated closest edge fitting (iterate these three steps until convergence).
   (a) Compute occluding boundary vertices for current shape and pose estimate and project to 2D.
   (b) Find correspondence between edges detected in the image and the projection of model vertices that lie on the occluding boundary. This is done in a nearest neighbour fashion with some filtering for robustness.

(c) With the correspondences to hand, treat edge vertices like landmarks with known correspondence and refit the model (initialising with the nonlinear parameters obtained in the previous iteration and retaining the original landmarks).

3. Nonlinear optimisation of hybrid objective function containing landmark, edge and prior terms.

3.6.1 Edge Cost

We assume that a subset of pixels have been labelled as edges and stored as the set $\psi = \{(x, y)|(x, y) \text{ is an edge}\}$. In practice, we compute edges by applying the Canny edge detector with a fixed threshold to the input image. More robust performance would be obtained by using a problem-specific edge detector such as boosted edge learning. This was recently done for fitting a morphable tooth model to contours in uncontrolled images [Wu et al., 2016].

Model contours are computed based on the pose and shape parameters as the occluding boundary of the 3D face. The set of occluding boundary vertices, $\mathcal{B}(\boldsymbol{\alpha}, \mathbf{R}, \mathbf{t}_{2\mathrm{d}}, s)$, are defined as those lying on a mesh edge whose adjacent faces have a change of visibility. This definition encompasses both outer (silhouette) and inner (self-occluding) contours. Since the viewing direction is aligned with the z axis, this is tested simply by checking if the sign of the z-component of the triangle normal changes on either side of the edge. In addition, we check that potential edge vertices are not occluded by another part of the mesh (using z-buffering) and we ignore edges that lie on a mesh boundary since they introduce artificial edges. We deal only with occluding contours (both inner and outer). If texture contours were defined on the surface of the morphable model, it would be straightforward to include these in our approach.

We define the objective function for edge fitting with hard correspondence as the sum of squared distances between each projected occluding boundary vertex and the closest edge pixel:

$$E_{\mathrm{edge}}(\boldsymbol{\alpha}, \mathbf{R}, \mathbf{t}_{2\mathrm{d}}, s) = \tfrac{1}{|\mathcal{B}(\boldsymbol{\alpha},\mathbf{R},\mathbf{t}_{2\mathrm{d}},s)|} \textstyle\sum_{i\in\mathcal{B}(\boldsymbol{\alpha},\mathbf{R},\mathbf{t}_{2\mathrm{d}},s)} \min_{(x,y)\in\psi} \|[x\ y]^T - \mathbf{SOP}\left[\mathbf{Q}_i\boldsymbol{\alpha} + \bar{\mathbf{s}}_i, \mathbf{R}, \mathbf{t}_{2\mathrm{d}}, s\right]\|^2. \quad (3.18)$$

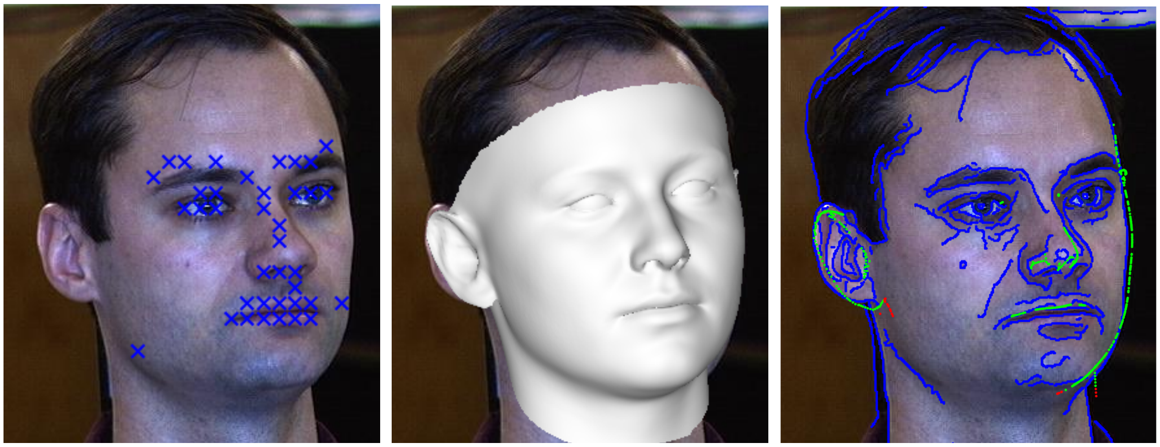


Figure 3.2: Iterated closest edge fitting for initialisation of the edge fitting process. Input image with automatically detected landmarks (left). Overlaid shape obtained by fitting only to landmark (middle). Image edges in blue, model boundary vertices with image correspondences in green, unreliable correspondences in red (right).

Note that the minimum operator is responsible for computing the hard correspondences. This objective is non-convex since the minimum of a set of convex functions is not convex [Grant et al., 2006]. Hence, we require a good initialisation to ensure convergence to a minimum close to the global optimum. Fitting to landmarks only does not provide a sufficiently good initialisation. For this reason, we describe a method for obtaining a good initial fit to edges, before incorporating the edge cost into a hybrid objective function.

3.6.2 Iterated Closest Edge Fitting

We propose to refine the landmark-only fit with an initial fit to edges that works in an iterated closest point manner. That is, for each projected model contour vertex, we find the closest image edge pixel and treat this as a known correspondence. In conjunction with the landmark correspondences, we again run the method in Section 3.5. This leads to updated pose and shape parameters and, in turn, to updated model edges and correspondences. We iterate this process for a fixed number of iterations. We refer to this process as Iterated Closest Edge Fitting (ICEF) and provide an illustration in Figure 3.2. On the left, we show an input image with the initial landmark detection result. In the middle, we show the initial shape and pose obtained by fitting only to landmarks. On the right, we show image edge pixels in blue and projected model contours in green (where nearest neighbour edge correspondence is considered reliable) and in red (where correspondence is considered unreliable). The green/blue correspondences are used for the next iteration of fitting.

Finding the image edge pixel closest to a projected contour vertex can be done efficiently by storing the image edge pixels in a kd-tree. We filter the resulting correspondences using two commonly used heuristics. First, we remove 5% of the matches for which the distance to the closest image edge pixel is largest. Second, we remove matches for which the image distance divided by s exceeds a threshold (chosen as 10 from our empirical experiments). The division by scale factor s makes this choice invariant to changes in image resolution.

3.6.3 Prior

Under the assumption that the training data of the 3DMM forms a Gaussian cloud in high dimensional space, then we expect that each of the shape parameters follows a normal distribution with zero mean and variance given by the eigenvalue, λ_i, associated with the corresponding principal component. We find that including a prior term that captures this assumption significantly improves performance over using the hyperbox constraint alone. The prior penalises deviation from the mean shape as follows:

$$E_{\text{prior}}(\boldsymbol{\alpha}) = \sum_{i=1}^{D} \left(\frac{\alpha_i}{\sqrt{\lambda_i}} \right)^2 . \tag{3.19}$$

3.6.4 Nonlinear Refinement

Finally, we perform nonlinear optimisation of a hybrid objective function comprising landmark, edge and prior terms:

$$E(\boldsymbol{\alpha}, \mathbf{R}, \mathbf{t}_{2\text{d}}, s) = w_1 E_{\text{lmk}}(\boldsymbol{\alpha}, \mathbf{R}, \mathbf{t}_{2\text{d}}, s) + w_2 E_{\text{edge}}(\boldsymbol{\alpha}, \mathbf{R}, \mathbf{t}_{2\text{d}}, s) + w_3 E_{\text{prior}}(\boldsymbol{\alpha}), \tag{3.20}$$

where w_1, w_2 and w_3 weight the contribution of each term to the overall energy. The landmark and edge terms are invariant to the number of landmarks and edge vertices which means we do not have to tune the weights for each image (for example, for the results in Table 3.1 we use fixed values of: $w_1 = 0.15$, $w_2 = 0.45$ and $w_3 = 0.4$). We retain the hyperbox constraint and so the hybrid objective is a constrained nonlinear least squares problem and we again optimise using the trust-region-reflective algorithm.

For efficiency and to avoid problems of continuity and differentiability of the edge cost function, we follow [Amberg et al., 2007] and keep occluding boundary vertices, $\mathcal{B}$, fixed for a number of iterations of the optimiser. After a number of iterations, we recompute the vertices lying on the occluding boundary and restart the optimiser.

3.6.5 Fitting with Soft Edge Correspondence

We compare our approach with a method based on optimising an edge cost function, in the same spirit as previous work [Romdhani and Vetter, 2005,Amberg et al., 2007,Keller et al., 2007]. The original method of [Romdhani and Vetter, 2005] used a cost surface that was the Euclidean distance transform of a binary edge image. This provides a smoothly varying cost surface whose value at a pixel indicates the distance (and its gradient, the direction) to the closest edge. In a slightly different setting, [Amberg et al., 2007] suggested using a more robust cost surface which removes the need to commit to a particular set of edge detection parameter values. The idea is to iterate over a range of parameter values (in this case, thresholds for non-maxima suppressed gradient magnitude values) and average the resulting distance transform maps.

We follow the same approach as [Amberg et al., 2007] to compute the edge cost function, however we remove an implicit assumption of their method. Integrating over edge threshold improves robustness to illumination changes or overall brightness changes by avoiding the selection of a hard threshold. However, the gradient magnitudes are always calculated using finite difference at a particular scale. This means there is an implicit assumption that the edges of interest are at the same scale as the finite difference window used. Therefore, if the input was a very high resolution image, face boundary edges may be missed and only high frequency, local noise picked up by the edge detector. We overcome this assumption by also integrating over scale, shown in Figure 3.3.

For our edge detector, we use gradient magnitude thresholding with non-maxima suppression. Given a set of edge detector sensitivity thresholds, thres, and scales, scal, we compute $n = \text{thres} \times \text{scal}$ edge images, $\text{Edge}^1, \dots, \text{Edge}^n$, using each pair of image scale and

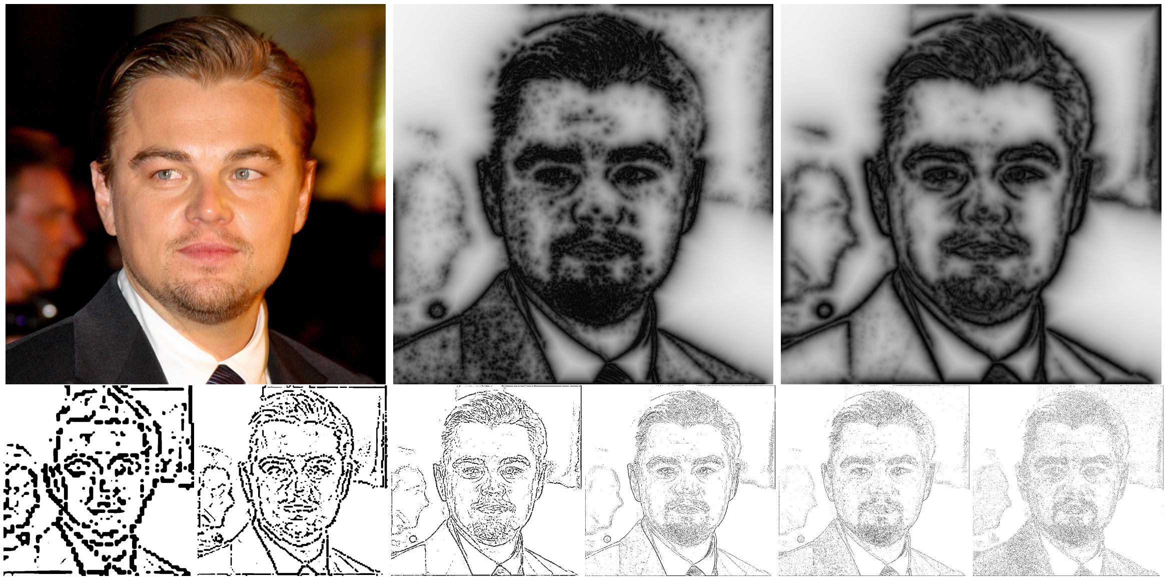


Figure 3.3: Edge cost surface with soft correspondence. Top row shows input image, ECS using original scale and ECS integrating over scale (from left to right). Bottom row shows edges at different scales to confirm that no single scale presents all desired boundary edges while accurately excluding undesired edges.

threshold values. We compute the Euclidean distance transform, $\text{Dist}^1, \dots, \text{Dist}^n$, for each edge image (i.e. the value of each pixel in Dist^i is the distance to the closest edge pixel in Edge^i). Finally, we compute the edge cost surface as:

$$ECS(x, y) = \frac{1}{n} \sum_{i=1}^{n} \frac{\text{Dist}^i(x, y)}{\text{Dist}^i(x, y) + \kappa}. \tag{3.21}$$

The parameter κ determines the influence range of an edge in an adaptive manner. [Amberg et al., 2007] suggest a value for κ of 1/20th the expected size of the head in pixels. We compute this parameter automatically from the scale s. To evaluate the edge cost, we compute model contour vertices as in Subsection 3.6.1, project them into the image and interpolate the edge cost function using bilinear interpolation:

$$E_{\text{softedge}}(\boldsymbol{\alpha}, \mathbf{R}, \mathbf{t}_{2\text{d}}, s) = \frac{1}{|\mathcal{B}(\boldsymbol{\alpha}, \mathbf{R}, \mathbf{t}_{2\text{d}}, s)|} \sum_{i \in \mathcal{B}(\boldsymbol{\alpha}, \mathbf{R}, \mathbf{t}_{2\text{d}}, s)} ECS(\mathbf{SOP}\left[\mathbf{Q}_i \boldsymbol{\alpha} + \bar{\mathbf{s}}_i, \mathbf{R}, \mathbf{t}_{2\text{d}}, s\right]). \tag{3.22}$$

As with the hard edge cost, we found that the best performance was achieved by also including the landmark and prior terms in a hybrid objective function. Hence, we minimise:

$$E(\boldsymbol{\alpha}, \mathbf{R}, \mathbf{t}_{2\text{d}}, s) = w_1 E_{\text{lmk}}(\boldsymbol{\alpha}, \mathbf{R}, \mathbf{t}_{2\text{d}}, s) + w_2 E_{\text{softedge}}(\boldsymbol{\alpha}, \mathbf{R}, \mathbf{t}_{2\text{d}}, s) + w_3 E_{\text{prior}}(\boldsymbol{\alpha}). \tag{3.23}$$

We again initialise by fitting to landmarks only using the method in Subsection 3.5.1, retain the hyperbox constraint and optimise using the trust-region-reflective algorithm. We use the same weights as for the hard correspondence method in our experiments.

3.7 Separable Nonlinear Least Squares

In this section, we demonstrate that the model fitting problem can be posed as a separable nonlinear least squares (SNLS) optimisation that can be solved efficiently without requiring any problem-specific optimisation method, initialisation or parameter tuning.

We reshape our fitting approach described in Sections 3.5 and 3.6 as a vector of residuals for clarity and introduce objective functions under perspective projection as well. Next, we show that these objective functions can be written in a form that is linear in some of the parameters (including shape) and nonlinear in the remainder. This special form of least squares problem can be solved more efficiently than general least squares problems and may converge when the original problem would diverge [Golub and Pereyra, 2003].

SNLS problems are solved by optimising a nonlinear least squares problem only in the nonlinear parameters, hence the problem dimensionality is reduced and the number of parameters that require initial guesses reduced. For convenience, henceforth we denote by $\mathbf{Q}_L \in \mathbb{R}^{3L\times D}$ the submatrix of $\mathbf{Q}$ containing the rows corresponding to the L landmarks (i.e. the first $3L$ rows of $\mathbf{Q}$).

We now present the objective functions as residual vectors for the orthographic and perspective cases and then show how they can be expressed as separable nonlinear least squares problems.

3.7.1 Orthographic Objective Function

In the orthographic case, we seek to minimise the following objective function:

$$\varepsilon_{\text{ortho}}(\mathbf{r}, \mathbf{t}_{\text{2d}}, s, \boldsymbol{\alpha}) = \mathbf{d}_{\text{ortho}}(\mathbf{r}, \mathbf{t}_{\text{2d}}, s, \boldsymbol{\alpha})^{\text{T}} \mathbf{d}_{\text{ortho}}(\mathbf{r}, \mathbf{t}_{\text{2d}}, s, \boldsymbol{\alpha}), \tag{3.24}$$

where the vector of residuals $\mathbf{d}_{\text{ortho}}(\mathbf{r}, \mathbf{t}_{\text{2d}}, s, \boldsymbol{\alpha}) \in \mathbb{R}^{2L}$ are given by:

$$\mathbf{d}_{\text{ortho}}(\mathbf{r}, \mathbf{t}_{\text{2d}}, s, \boldsymbol{\alpha}) = \begin{bmatrix} \mathbf{x}_1 - \mathbf{SOP}\left[\mathbf{Q}_1\boldsymbol{\alpha} + \bar{\mathbf{s}}_1, \mathbf{R}(\mathbf{r}), \mathbf{t}_{\text{2d}}, s\right] \\ \vdots \\ \mathbf{x}_L - \mathbf{SOP}\left[\mathbf{Q}_L\boldsymbol{\alpha} + \bar{\mathbf{s}}_L, \mathbf{R}(\mathbf{r}), \mathbf{t}_{\text{2d}}, s\right] \end{bmatrix}. \tag{3.25}$$

These residuals are linear in the shape parameters, translation vector and scale but nonlinear in the rotation vector. In Section 3.5, we treated this as a multilinear optimisation problem and used alternating linear least squares and subsequently refined using nonlinear optimisation. Instead, we observe that the problem can be treated as linear in the shape and translation parameters simultaneously and nonlinear in scale and rotation.

The vector of residuals (3.25) in the orthographic objective function (3.24) can be written in SNLS form as

$$\mathbf{d}_{\text{ortho}}(\mathbf{r}, \mathbf{t}_{\text{2d}}, s, \boldsymbol{\alpha}) = \mathbf{A}(\mathbf{r}, s)\begin{bmatrix} \boldsymbol{\alpha} \\ \mathbf{t}_{\text{2d}} \end{bmatrix} - \mathbf{y}(\mathbf{r}, s), \tag{3.26}$$

where $\mathbf{A}(\mathbf{r}, s) \in \mathbb{R}^{2L \times D+2}$ is given by

$$\mathbf{A}(\mathbf{r}, s) = s\left[\left(\mathbf{I}_L \otimes [\mathbf{PR}(\mathbf{r})]\right)\mathbf{Q}_L \quad \mathbf{1}_L \otimes \mathbf{I}_2\right], \tag{3.27}$$

and $\mathbf{y}(\mathbf{r}, s) \in \mathbb{R}^{2L}$ is given by

$$\mathbf{y}(\mathbf{r}, s) = s\left(\mathbf{I}_L \otimes [\mathbf{PR}(\mathbf{r})]\right)\bar{\mathbf{s}} - \begin{bmatrix} x_1 & y_1 & \dots & y_L \end{bmatrix}^{\text{T}}. \tag{3.28}$$

Note that this vector of residuals is exactly equivalent to the original one. The optimal solution to the original objective function (3.24) in terms of the linear parameters is given by:

$$\begin{bmatrix} \boldsymbol{\alpha}^* \\ \mathbf{t}_{\text{2d}}^* \end{bmatrix} = \mathbf{A}^+(\mathbf{r}, s)\mathbf{y}(\mathbf{r}, s), \tag{3.29}$$

where $\mathbf{A}^+(\mathbf{r}, s)$ is the pseudoinverse. Substituting (3.29) into (3.26) we get a vector of residuals that is exactly equivalent to (3.25) but which depends only on the nonlinear parameters:

$$\mathbf{d}_{\text{ortho}}(\mathbf{r}, s) = \mathbf{A}(\mathbf{r}, s)\mathbf{A}^+(\mathbf{r}, s)\mathbf{y}(\mathbf{r}, s) - \mathbf{y}(\mathbf{r}, s). \tag{3.30}$$

Substituting this into (3.24), we get an equivalent objective function, $\varepsilon_{\text{ortho}}(\mathbf{r}, s)$, again depending only on the nonlinear parameters. This is a nonlinear least squares problem of very low dimensionality ($[\mathbf{r}\ \ s]$ is only 4D). We solve this using the trust-region-reflective algorithm for which we require $\mathbf{J}_{\mathbf{d}_{\text{ortho}}}(\mathbf{r}, s) \in \mathbb{R}^{2L\times 4}$, the Jacobian of the residual function. In Appendix A, we analytically derive $\mathbf{J}_{\mathbf{d}_{\text{ortho}}}$. Although computing these derivatives is quite involved, in practice it is still faster than using finite difference approximations. Once optimal parameters have been obtained by minimising $\varepsilon_{\text{ortho}}(\mathbf{r}, s)$ then the parameters $\boldsymbol{\alpha}^*$ and $\mathbf{t}^*$ are obtained by (3.29).

If we wish to impose a statistical prior on the shape parameters we can use Tikhonov regularisation, as in [Blanz et al., 2004], during the solution of (3.29).

3.7.2 Perspective Objective Function

In the perspective case, we seek to minimise the following objective function:

$$\varepsilon_{\text{persp}}(\mathbf{r}, \mathbf{t}_{3\text{d}}, f, \boldsymbol{\alpha}) = \mathbf{d}_{\text{persp}}(\mathbf{r}, \mathbf{t}_{3\text{d}}, f, \boldsymbol{\alpha})^{\text{T}} \mathbf{d}_{\text{persp}}(\mathbf{r}, \mathbf{t}_{3\text{d}}, f, \boldsymbol{\alpha}), \tag{3.31}$$

where the vector of residuals $\mathbf{d}_{\text{persp}}(\mathbf{r}, \mathbf{t}_{3\text{d}}, f, \boldsymbol{\alpha}) \in \mathbb{R}^{2L}$ is given by:

$$\mathbf{d}_{\text{persp}}(\mathbf{r}, \mathbf{t}_{3\text{d}}, f, \boldsymbol{\alpha}) = \begin{bmatrix} \mathbf{x}_1 - \textbf{pinhole}\left[\mathbf{Q}_1\boldsymbol{\alpha} + \bar{\mathbf{s}}_1, \mathbf{K}(f), \mathbf{R}(\mathbf{r}), \mathbf{t}_{3\text{d}}\right] \\ \vdots \\ \mathbf{x}_L - \textbf{pinhole}\left[\mathbf{Q}_L\boldsymbol{\alpha} + \bar{\mathbf{s}}_L, \mathbf{K}(f), \mathbf{R}(\mathbf{r}), \mathbf{t}_{3\text{d}}\right] \end{bmatrix}. \tag{3.32}$$

These residuals are nonlinear in all parameters and non-convex due to the perspective projection. However, we can use the direct linear transformation (DLT) [Hartley and Zisserman, 2003] to transform the problem to a linear one. The solution of this easier problem provides a good initialisation for nonlinear optimisation of the true objective.

From (3.3) and (3.10) we have a linear similarity relation for each landmark point:

$$\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} \sim \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t}_{3\text{d}} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_i\boldsymbol{\alpha} + \bar{\mathbf{s}}_i \\ 1 \end{bmatrix}, \tag{3.33}$$

where $\sim$ denotes equality up to a non-zero scalar multiplication. We rewrite as a collinearity condition:

$$\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}_\times \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t}_{3\text{d}} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_i \boldsymbol{\alpha} + \bar{\mathbf{s}}_i \\ 1 \end{bmatrix} = \mathbf{0}, \tag{3.34}$$

where $\mathbf{0} = [0\ 0\ 0]^{\mathrm{T}}$. This means that each landmark yields three equations that are linear in the unknown shape parameters $\boldsymbol{\alpha}$ and the translation vector $\mathbf{t}_{3\text{d}}$. The perspective residual function (3.32), linearised via (3.34), can be written in SNLS form as

$$\mathbf{d}_{\text{persp}}^{\text{DLT}}(\mathbf{r}, \mathbf{t}_{3\text{d}}, f, \boldsymbol{\alpha}) = \mathbf{B}(\mathbf{r}, f) \begin{bmatrix} \boldsymbol{\alpha} \\ \mathbf{t}_{3\text{d}} \end{bmatrix} - \mathbf{z}(\mathbf{r}, f), \tag{3.35}$$

where $\mathbf{B}(\mathbf{r}, f) \in \mathbb{R}^{3L \times D+3}$ is given by:

$$\mathbf{B}(\mathbf{r}, f) = \mathbf{D}\mathbf{E}(f)\mathbf{F}(\mathbf{r}), \tag{3.36}$$

with

$$\mathbf{D} = \operatorname{diag}\left(\begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}_\times, \dots, \begin{bmatrix} \mathbf{x}_L \\ 1 \end{bmatrix}_\times \right), \quad \mathbf{E}(f) = \mathbf{I}_L \otimes \mathbf{K}(f), \tag{3.37}$$

and

$$\mathbf{F}(\mathbf{r}) = \left[\left(\mathbf{I}_L \otimes \mathbf{R}(\mathbf{r}) \right) \mathbf{Q}_L \quad \mathbf{1}_L \otimes \mathbf{I}_3 \right]. \tag{3.38}$$

The vector $\mathbf{z}(\mathbf{r}, f) \in \mathbb{R}^{3L}$ is given by:

$$\mathbf{z}(\mathbf{r}, f) = -\mathbf{D} \left(\mathbf{I}_L \otimes [\mathbf{K}(f)\mathbf{R}(\mathbf{r})] \right) \bar{\mathbf{s}}. \tag{3.39}$$

Exactly as in the orthographic case, we can write optimal solutions for the linear parameters in terms of the nonlinear parameters and solve a 4D nonlinear minimisation problem in $(\mathbf{r}, f)$. In contrast to the orthographic case, this objective is not equivalent to minimisation of the original objective, i.e. the sum of squared perspective reprojection distances in (3.31). Therefore, we use the SNLS solution to initialise a nonlinear least squares optimisation of the original objective over all parameters, again using trust-region-reflective. In practice, we find that the SNLS solution is already very close to the optimum and that the subsequent nonlinear least squares optimisation usually converges in 2-5 iterations.

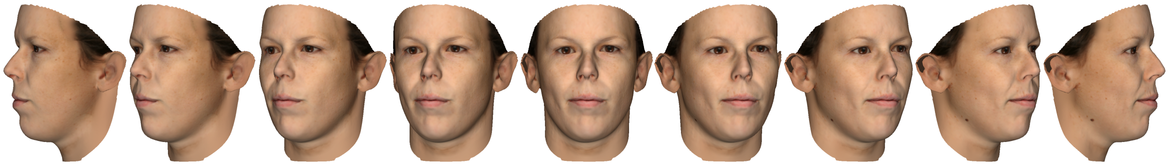


Figure 3.4: Synthetic input images for one subject.

3.8 Experimental Results

We present two sets of experimental results. First, we use synthetic images with known ground truth 3D shape in order to quantitatively evaluate our method and provide comparison to previous work. Second, we use real images to provide qualitative evidence of the performance of our method in uncontrolled conditions. For the 3DMM in both sets of experiments we use the Basel Face Model [Paysan et al., 2009].

3.8.1 Quantitative Evaluation

We begin with a quantitative comparative evaluation on synthetic data. We use the 10 out-of-sample faces supplied with the BFM and render orthographic images of each face in 9 poses (rotations of 0°, ±15°, ±30°, ±50° and ±70° about the vertical axis). We show sample input images for one subject in Figure 3.4. In all experiments, we report the mean Euclidean distance between ground truth and estimated face surface in mm after Procrustes alignment.

In the first experiment, we use ground truth landmarks. Specifically, we use the 70 Farkas landmarks, project the visible subset to the image (yielding between 37 and 65 landmarks per image) and round to the nearest pixel. In Table 3.1, we show results averaged over pose angle and over the whole dataset.

	Rotation angle									
Method	−70°	−50°	−30°	−15°	0°	15°	30°	50°	70°	**Mean**
Average face	3.35	3.35	3.35	3.35	3.35	3.35	3.35	3.35	3.35	3.35
Proposed (landmarks only)	2.67	2.60	2.58	2.64	2.56	2.49	2.50	2.54	2.63	2.58
[Aldrian and Smith, 2013]	2.64	2.60	2.55	2.54	**2.49**	2.42	2.43	2.44	2.54	2.52
[Romdhani and Vetter, 2005] (soft)	2.65	2.59	2.58	2.61	2.59	2.50	2.50	2.46	2.51	2.55
Proposed (ICEF)	2.38	2.40	2.51	**2.38**	2.52	2.45	2.43	2.38	2.3	2.42
Proposed (hard)	**2.35**	**2.26**	**2.38**	2.40	2.51	**2.39**	**2.40**	**2.20**	**2.26**	**2.35**

Table 3.1: Mean Euclidean vertex distance (mm) with ground truth landmarks.

Method	Landmark noise std. dev.					
	$\sigma = 0$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 4$	$\sigma = 5$
Proposed (landmarks only)	2.58	2.60	2.61	2.68	2.76	2.85
[Aldrian and Smith, 2013]	2.52	2.53	2.55	2.62	2.65	2.73
[Romdhani and Vetter, 2005] (soft)	2.55	2.57	2.57	2.62	2.70	2.76
Proposed (ICEF)	2.42	2.43	2.43	2.50	2.57	2.60
Proposed (hard)	**2.35**	**2.36**	**2.35**	**2.39**	**2.47**	**2.50**

Table 3.2: Mean Euclidean vertex distance (mm) with noisy landmarks.

Method	Rotation angle									Mean
	$-70°$	$-50°$	$-30°$	$-15°$	$0°$	$15°$	$30°$	$50°$	$70°$	
Proposed (landmarks only)	6.79	6.84	5.19	5.74	5.68	6.34	6.48	7.04	7.74	6.43
[Zhu et al., 2015]	N/A	N/A	4.63	5.09	4.19	5.22	4.92	N/A	N/A	N/A
[Romdhani and Vetter, 2005] (soft)	4.46	3.42	3.66	3.78	3.77	3.57	4.31	4.19	4.73	3.99
Proposed (ICEF)	3.70	3.32	3.26	3.23	3.37	3.50	3.43	4.07	3.52	3.49
Proposed (hard)	**3.43**	**3.20**	**3.19**	**3.09**	**3.30**	**3.36**	**3.36**	**3.84**	**3.41**	**3.35**

Table 3.3: Mean Euclidean vertex distance (mm) with automatically detected landmarks.

As a baseline, we show the error if we simply use the average face shape. We then show the result of fitting only to landmarks, i.e. the method in Section 3.5. We include two comparison methods. The approach of [Aldrian and Smith, 2013] uses only landmarks but with an affine camera model and a learnt model of landmark variance. The soft edge correspondence method of [Romdhani and Vetter, 2005] is described in Subsection 3.6.5. The final two rows show two variants of our proposed methods: the fast iterated closest edge fitting version and the full version with nonlinear optimisation of the hard correspondence cost. Average performance over the whole dataset is best for our method and, in general, using edges over landmarks only and applying nonlinear optimisation improves performance. The performance of our methods over landmark-only methods improves with pose angle. This suggests that edge information becomes more salient for non-frontal poses.

The second experiment is identical to the first except that we add Gaussian noise of varying standard deviation to the ground truth landmark positions. In Table 3.2, we show results averaged over all poses and subjects.

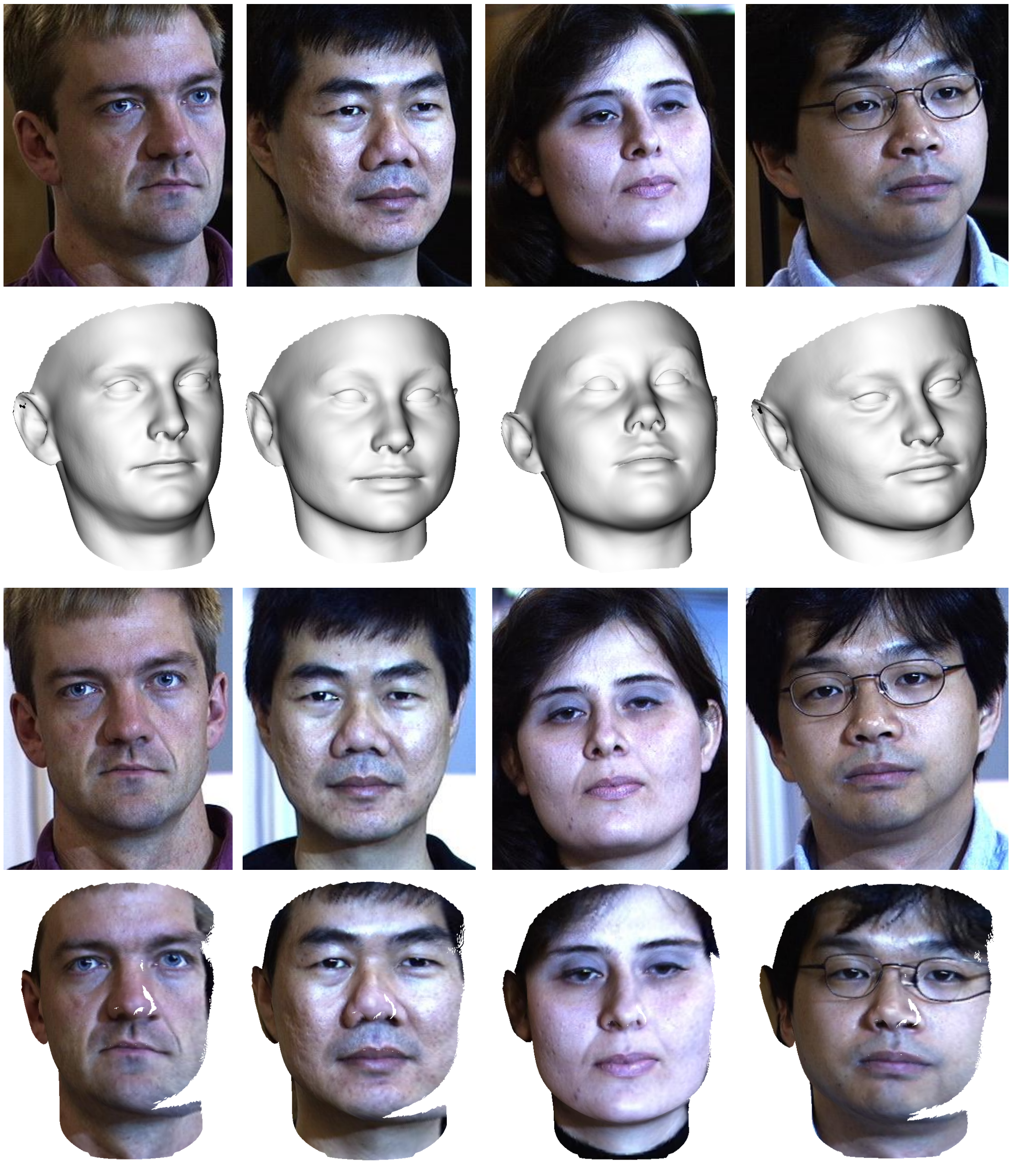


Figure 3.5: Qualitative frontalisation results. First row shows input images. Second row shows fitted models. Third and fourth rows show frontal images of samples and reconstruction with textured frontalisation for comparison.

In the final experiment, we use landmarks that are automatically detected using the method of [Zhu and Ramanan, 2012]. This enables us to include comparison with the fitting algorithm of [Zhu et al., 2015]. We use the author's own implementation which only works with a fixed set of 68 landmarks. This means that the method cannot be applied to the more extreme pose angles where fewer landmarks are detected. In this more challenging scenario, our method again gives the best overall performance and is superior for all pose angles, shown in Table 3.3.

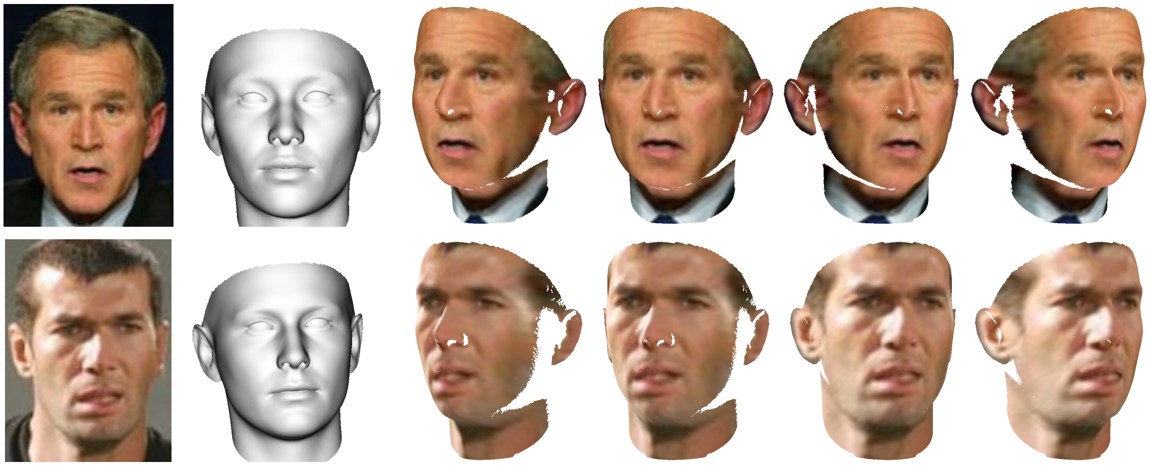


Figure 3.6: Qualitative pose editing results. First and second columns show input images and fitted models. Third to sixth columns show textured models with different poses.

3.8.2 Qualitative Evaluation

In Figure 3.5, we show qualitative examples from the CMU PIE dataset [Sim et al., 2003]. Here, we fit to images (first row) in a non-frontal pose using automatically detected landmarks [Zhu and Ramanan, 2012] and show the reconstruction (second row). We texture map the image onto the mesh, rotate to frontal pose (fourth row) and compare to an actual frontal view (third row).

Finally, we show qualitative examples from the Labeled Faces in the Wild dataset [Huang et al., 2007] in Figure 3.6. Again, we texture map the image to the mesh and show a range of poses. These results show that our method is capable of robustly and fully automatically fitting to unconstrained images.

3.8.3 SNLS Evaluation

We now compare our separable nonlinear least squares approach with alternating least squares as used in this chapter and previous work [Aldrian and Smith, 2013, Cao et al., 2013, Cao et al., 2014a, Zhu et al., 2015, Saito et al., 2016]. In order to evaluate in a realistic setting, we require images with corresponding ground truth 3DMM fits. For this reason, we use the FaceWarehouse dataset and model [Cao et al., 2014b]. We use leave-one-out testing, building each model on 149 subjects and testing on the remaining one and use the 74 landmarks provided with the dataset. For this evaluation we test only the orthographic setting.

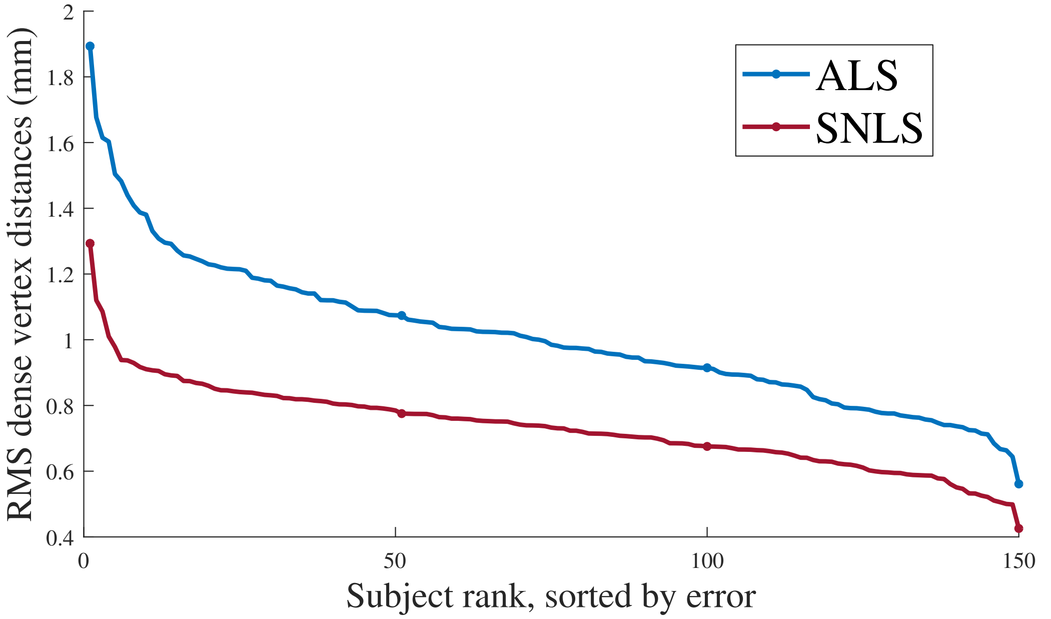


Figure 3.7: Quantitative comparison between alternating linear least squares (ALS) and separable nonlinear least squares (SNLS) on 150 subjects in the FaceWarehouse dataset. The average dense surface error is 1.01mm for ALS and 0.73mm for SNLS.

Figure 3.7 shows the mean Euclidean distance between dense ground truth and estimated face surface in mm after Procrustes alignment. We do not use any regularisation for either algorithm and therefore do not need to choose the weight parameter. For all subjects SNLS achieves a lower error, on average reducing it by about 30%.

3.9 Conclusions

In this chapter, we have presented a fully automatic algorithm for fitting a 3DMM to single images using hard edge correspondence and compared it to existing methods using soft correspondence.

In 3D-3D alignment, the soft correspondence of LM-ICP [Fitzgibbon, 2003] is demonstrably more robust than hard ICP [Besl and McKay, 1992]. However, in the context of 3D-2D non-rigid alignment, a soft edge cost function is neither continuous nor differentiable since contours appear, disappear, split and merge under parameter changes [Keller et al., 2007]. This makes its optimisation challenging, unstable and highly dependent on careful choice of optimisation parameters. Although our proposed algorithm relies on potentially brittle hard correspondences, solving for shape and pose separately requires only solution of a linear problem and, together, optimisation of a multilinear problem. We have observed that this makes iterated closest edge fitting faster and provides an initialisation that allows the subsequent nonlinear optimisation to converge to a better optimum. This explains the improved performance over edge fitting with soft correspondence.

Further, we have shown that our proposed algorithms for fitting a 3D morphable model to 2D landmarks or contours under either orthographic or perspective projection can be posed as a separable nonlinear least squares problem and solved efficiently. We have provided quantitative comparison between alternating linear least squares and separable nonlinear least squares to demonstrate that this reformulation is indeed superior.

Chapter 4

Geometry Meets Deep Learning

4.1 Introduction

The methods in the previous chapter use classical methods from image processing (e.g. edge detection), face analysis (e.g. landmark detection) and various optimisation algorithms. Such combinations of handcrafted features and methods typify computer vision approaches into the early 2010s. They rely on human ingenuity to design suitable features, tune parameters and select optimisation algorithms, with no guarantee that any of these decisions are optimal. For example, the use of edge features in the previous chapter relies on accurate edge detection which in turn depends on suitable parameter selections and, in some images, meaningful edges may not even exist.

In this chapter, we shift our focus to deep learning techniques, in particular, Convolutional Neural Networks. Recently, CNNs have achieved state-of-the-art performance in many computer vision tasks, including image classification [Krizhevsky et al., 2012,He et al., 2016], object detection [Ren et al., 2015,Redmon et al., 2016,He et al., 2017] and face recognition [Taigman et al., 2014,Sun et al., 2015,Parkhi et al., 2015]. Rather than rely on handcrafted feature design, CNNs are trained end-to-end and learn low level features, intermediate representations and high level abstractions in the process of being trained to solve a particular task.

CNNs are usually trained with such large amounts of data that they can learn invariance to scale, translation, in-plane rotation and, to a certain degree, out-of-plane rotations, without using any explicit geometric transformation model. However, most networks do require a rough bounding box estimate as input and do not work for larger variations. Recently, [Jaderberg et al., 2015] proposed the Spatial Transformer Network – a module that can be incorporated into a neural network architecture, giving the network the ability to explicitly account for the effects of pose and non-rigid deformations (which we refer to simply as "pose"). An STN explicitly estimates pose and then resamples a specific part of the input image to a fixed-size output image. It is thus able to work on inputs with larger translation and pose variation in general, since it can explicitly compensate for it, and feed a transformed region of interest to the subsequent neural network layers. By exploiting and "hard-coding" knowledge of geometric transformation, the amount of training data and the required complexity of the network can be vastly reduced.

In this chapter, we show how to use 2D and 3D statistical models as a spatial transformer within a convolutional neural network. Rather than rely on generic layers within a CNN to learn invariance to various kinds of spatial transformation, an STN includes expert layers that predict and apply a parametric transformation to an input feature map. By including a shape model as a component within a CNN, the network can learn its own notion of alignment that is optimal for the task that it is learning to solve.

First, we show how to use a 3D morphable model as a spatial transformer network (we refer to this as a 3DMM-STN). In this setting, the locations in the input image that are resampled are determined by the 2D projection of a 3D deformable mesh. Hence, our 3DMM-STN estimates both 3D shape and pose. This allows us to explicitly estimate and account for 3D rotations as well as self-occlusions. The output of our 3DMM-STN is a resampled image in a flattened 2D texture space in which the images are in dense, pixel-wise correspondence. Hence, this output can be fed to subsequent CNN layers for further processing. Although we focus on face images and use a 3D morphable face model [Blanz and Vetter, 1999,Paysan et al., 2009], our idea is general and could be applied to any object for which a statistical 3D shape model is available (though note that the symmetry and Siamese loss functions proposed in Subsection 4.3.4 do assume that the object is bilaterally symmetric).

Second, we generalise the idea of spatial transformer networks by replacing the parametric transformation of a fixed, regular sampling grid with a deformable, statistical shape model which is itself learnt. We call this a Statistical Transformer Network (StaTN). By training a network containing a StaTN end-to-end for a particular task, the network learns the optimal non-rigid alignment of the input data for the task. Moreover, the statistical shape model is learnt with no direct supervision (such as landmarks) and can be reused for other tasks. Besides training for a specific task, we also show that a StaTN can learn a shape model using generic loss functions. This includes a loss inspired by the minimum description length principle in which an appearance model is also learnt from scratch. In this configuration, our model learns an active appearance model and a means to fit the model from scratch with no supervision at all.

4.2 Contributions

In this chapter, we propose a purely geometric approach in which only the shape component of a 3DMM is used to geometrically normalise an image. Unlike [Jourabloo and Liu, 2016, Richardson et al., 2016, Tran et al., 2017, Güler et al., 2017, Sela et al., 2017, Jackson et al., 2017, Kim et al., 2018], our method can be trained in an unsupervised fashion, and thus does not depend on synthetic training data or the fitting results of an existing algorithm. In contrast to [Tewari et al., 2017], we avoid the complexity and potential fragility of having to model illumination and reflectance parameters. Moreover, our 3DMM-STN can form part of a larger network that performs a face processing task and is trained end-to-end. Finally, in contrast to all previous 3DMM fitting networks, the output of our 3DMM-STN is a 2D resampling of the original image which contains all of the high frequency, discriminating detail in a face rather than a model-based reconstruction which only captures the gross, low frequency aspects of appearance that can be explained by a 3DMM.

Similarly, we propose a new adaptation of a spatial transformer network to replace fixed transformation model that explicitly learns dense, non-rigid correspondence by incorporating 2D shape and appearance model into our network. The mean and principal components of a statistical model are subject to constraints (e.g. orthogonality of principal components).

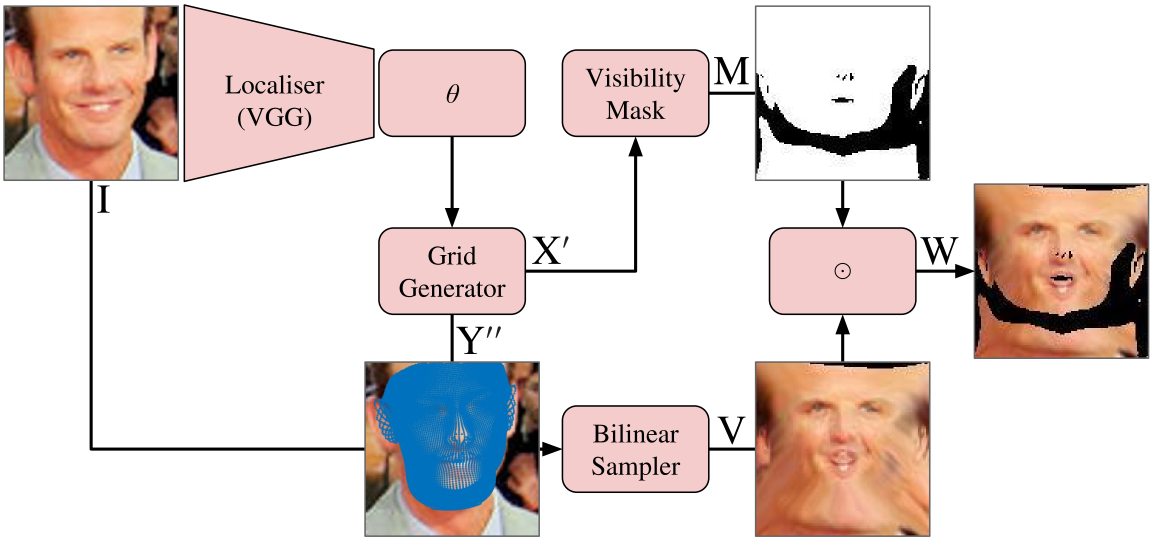


Figure 4.1: Diagram of the 3DMM-STN. The localiser predicts 3DMM shape parameters and pose. The grid generator projects the 3D geometry to 2D. The bilinear sampler resamples the input image to a regular output grid which is then masked by an occlusion mask computed from the estimated 3D geometry.

We show how these can be enforced by incorporating manifold gradient descent into backpropagation. We introduce generic losses that can be used to train a StaTN without supervision (i.e. not even identity labels for computing a classification loss).

4.3 3D Morphable Models as Spatial Transformer Networks

Our proposed 3DMM-STN has the same components as a conventional STN, however each component must be modified to incorporate the statistical shape model, 3D transformations, projection and self-occlusion. In this section, we describe each component of a 3DMM-STN and the layers that are required to construct it. A diagram of our architecture is shown in Figure 4.1.

4.3.1 Localiser Network

The localiser network is a CNN that takes an image as input and regresses the pose and shape parameters, θ, of the face in the image. Specifically, we predict the following vector of parameters:

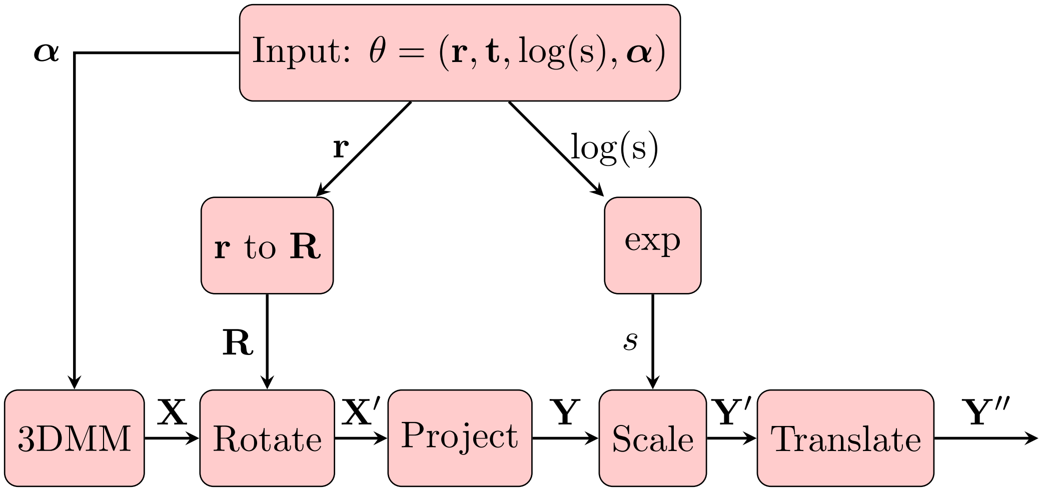


Figure 4.2: The grid generator network within the 3DMM-STN.

$$\theta = (\underbrace{\mathbf{r}, \mathbf{t}, \log(\mathrm{s})}_{\text{pose}}, \underbrace{\boldsymbol{\alpha}}_{\text{shape}}). \tag{4.1}$$

Here, $\mathbf{t} \in \mathbb{R}^2$ is a 2D translation, $\mathbf{r} \in \mathbb{R}^3$ is an axis-angle representation of a 3D rotation with rotation angle $\|\mathbf{r}\|$ and axis $\mathbf{r}/\|\mathbf{r}\|$. Since scale must be positive, we estimate log scale and later pass this through an exponentiation layer, ensuring that the estimated scale, s, is positive. The shape parameters $\boldsymbol{\alpha} \in \mathbb{R}^D$ are the principal component weights used to reconstruct the shape.

For our localiser network, we use the pretrained VGG-Faces [Parkhi et al., 2015] architecture, delete the classification layer and add a new fully connected layer with $6 + D$ outputs. The weights for the new layer are randomly initialised but scaled so that the elements of the axis-angle vector are in the range $[-\pi, \pi]$ for typical inputs. The whole localiser is then fine-tuned as part of the subsequent training.

4.3.2 Grid Generator Network

In contrast to a conventional STN, the warped sampling grid is not obtained by applying a global transformation to the regular output grid. Instead, we apply a 3D transformation and projection to a 3D mesh that comes from the morphable model. The intensities sampled from the source image are then assigned to the corresponding points in a flattened 2D grid. For this reason, the grid generator network in a 3DMM-STN is more complex than in a conventional STN, although we emphasise that it remains differentiable and hence suitable for use in end-to-end training. The sample points in our grid generator are determined by the transformation parameters θ estimated by the localiser network.

Our grid generator combines a linear statistical model with a scaled orthographic projection as shown in Figure 4.2. Note that we could alternatively use a perspective projection (modifying the localiser to predict a 3D translation as well as camera parameters such as focal length). We now describe the transformation applied by each layer in the grid generator and provide derivatives.

3D Morphable Model Layer The 3DMM layer generates a shape $\mathbf{X} \in \mathbb{R}^{3\times N}$ comprising N 3D vertices by taking a linear combination of D basis shapes (principal components) stored in the matrix $\mathbf{Q} \in \mathbb{R}^{3N\times D}$ and the mean shape $\bar{\mathbf{s}} \in \mathbb{R}^{3N}$ according to shape parameters $\boldsymbol{\alpha} \in \mathbb{R}^{D}$:

$$\mathbf{X}(\boldsymbol{\alpha})_{i,j} = \mathbf{s}(\boldsymbol{\alpha})_{3(j-1)+i}, \quad i \in [1,3], j \in [1,N], \tag{4.2}$$

where

$$\mathbf{s}(\boldsymbol{\alpha}) = \mathbf{Q}\boldsymbol{\alpha} + \bar{\mathbf{s}}, \tag{4.3}$$

and the derivatives are given by:

$$\frac{\partial \mathbf{s}}{\partial \boldsymbol{\alpha}} = \mathbf{Q}, \quad \frac{\partial X_{i,j}}{\partial \alpha_k} = Q_{3(j-1)+i,k}. \tag{4.4}$$

Note that such a linear model is exactly equivalent to a fully connected layer (and hence a special case of a convolutional layer) with fixed weights and biases. This is not at all surprising since a linear model is exactly what is implemented by a single layer linear decoder.

In this interpretation, the shape parameters play the role of the input map, the principal components the role of weights and the mean shape the role of biases. This means that this layer can be implemented using an existing implementation of a convolution layer and also, following our later suggestion for future work, that the model could itself be made trainable simply by having non-zero learning rate for the convolution layer.

In our network, we use some of the principal components to represent shape variation due to identity and the remainder to represent deformation due to expression. We assume that expressions are additive and we can thus combine the two into a single linear model. Note that the shape parameters relating to identity may contain information that is useful for recognition, so these could be incorporated into a descriptor in a recognition network after the STN.

Axis-angle to Rotation Matrix Layer This layer converts an axis-angle representation of a rotation, $\mathbf{r} \in \mathbb{R}^3$, into a rotation matrix:

$$\mathbf{R}(\mathbf{r}) = \cos\theta\mathbf{I} + \sin\theta\left[\bar{\mathbf{r}}\right]_\times + (1-\cos\theta)\bar{\mathbf{r}}\bar{\mathbf{r}}^T, \tag{4.5}$$

where $\theta = \|\mathbf{r}\|$ and $\bar{\mathbf{r}} = \mathbf{r}/\|\mathbf{r}\|$ and

$$\left[\mathbf{a}\right]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{4.6}$$

is the cross product matrix. The derivatives are given by [Gallego and Yezzi, 2015]:

$$\frac{\partial\mathbf{R}}{\partial r_i} = \begin{cases} [\mathbf{e}_i]_\times & \text{if } \mathbf{r} = \mathbf{0} \\ \frac{r_i[\mathbf{r}]_\times + [\mathbf{r}\times(\mathbf{I}-\mathbf{R}(\mathbf{r}))\mathbf{e}_i]_\times}{\|\mathbf{r}\|^2}\mathbf{R} & \text{otherwise} \end{cases}, \tag{4.7}$$

where $\mathbf{e}_i$ is the ith vector of the standard basis in $\mathbb{R}^3$.

3D Rotation Layer The rotation layer takes as input a rotation matrix $\mathbf{R}$ and N 3D points $\mathbf{X} \in \mathbb{R}^{3\times N}$ and applies the rotation:

$$\begin{gathered} \mathbf{X}'(\mathbf{R},\mathbf{X}) = \mathbf{R}\mathbf{X}, \\ \frac{\partial X'_{i,j}}{\partial R_{i,k}} = X_{k,j}, \quad \frac{\partial X'_{i,j}}{\partial X_{k,j}} = R_{i,k}, \quad i,k \in [1,3], j \in [1,N]. \end{gathered} \tag{4.8}$$

Orthographic Projection Layer The orthographic projection layer takes as input a set of N 3D points $\mathbf{X}' \in \mathbb{R}^{3\times N}$ and outputs N 2D points $\mathbf{Y} \in \mathbb{R}^{2\times N}$ by applying an orthographic projection along the z axis:

$$\begin{aligned} &\mathbf{Y}(\mathbf{X}') = \mathbf{P}\mathbf{X}', \quad \mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \\ &\frac{\partial Y_{i,j}}{\partial X'_{i,j}} = 1, \quad i \in [1,2], j \in [1,N]. \end{aligned} \tag{4.9}$$

Scaling Layer The log scale estimated by the localiser is first transformed to scale by an exponentiation layer:

$$s(\log(\mathrm{s})) = \exp(\log(\mathrm{s})), \quad \frac{\partial s}{\partial \log(\mathrm{s})} = \exp(\log(\mathrm{s})). \tag{4.10}$$

Then, the 2D points $\mathbf{Y} \in \mathbb{R}^{2\times N}$ are scaled:

$$\mathbf{Y}'(s, \mathbf{Y}) = s\mathbf{Y}, \quad \frac{\partial Y'_{i,j}}{\partial s} = Y_{i,j}, \quad \frac{\partial Y'_{i,j}}{\partial Y_{i,j}} = s. \tag{4.11}$$

Translation Layer Finally, the 2D sample points are generated by adding a 2D translation $\mathbf{t} \in \mathbb{R}^2$ to each of the scaled points:

$$\mathbf{Y}''(\mathbf{t}, \mathbf{Y}') = \mathbf{Y}' + \mathbf{1}_N \otimes \mathbf{t}, \quad \frac{\partial Y''_{i,j}}{\partial t_i} = 1, \quad \frac{\partial Y''_{i,j}}{\partial Y'_{i,j}} = 1, \tag{4.12}$$

where $\mathbf{1}_N$ is the row vector of length N containing ones and $\otimes$ is the Kronecker product.

4.3.3 Sampling

In the original STN, the sampler component used bilinear sampling to sample values from the input image and transform them to an output grid. We make a number of modifications. First, the output grid is a texture space flattening of the 3DMM mesh. Second, the bilinear sampler layer will incorrectly sample parts of the face onto vertices that are self-occluded, so we introduce additional layers that calculate which vertices are occluded and mask the sampled image appropriately.

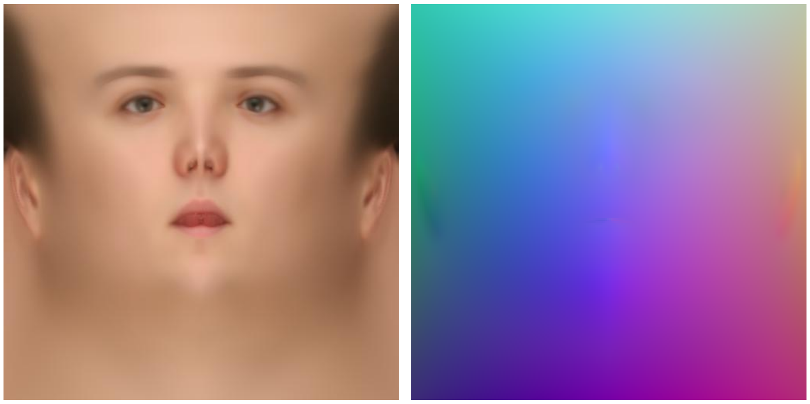


Figure 4.3: The output grid of the 3DMM-STN: a Tutte embedding of the mean shape of the Basel Face Model. A visualisation using the mean texture (though note that our 3DMM-STN does not use a texture model) (left) and the mean shape as a geometry image (right) [Gu et al., 2002].

Output Grid The purpose of an STN is to transform an input image into a canonical, pose-normalised view. In the context of a 3D model, one could imagine a number of analogous ways that an input image could be normalised. For example, the output of the STN could be a rendering of the mean face shape in a frontal pose with the sampled texture on the mesh. Instead, we choose to output sampled textures in a 2D embedding obtained by flattening the mean shape of the 3DMM. This ensures that the output image is approximately area uniform with respect to the mean shape and also that the whole output image contains face information.

Specifically, we compute a Tutte embedding [Floater, 1997] using conformal Laplacian weights and with the mesh boundary mapped to a square. To ensure a symmetric embedding, we map the symmetry line to the symmetry line of the square, flatten only one side of the mesh and obtain the flattening of the other half by reflection. We show a visualisation of our embedding using the mean texture in Figure 4.3. In the order that the output warped image produces a regularly sampled image, we regularly re-sample (i.e. re-mesh) the 3DMM (mean and principal components) over a uniform grid of size $H' \times W'$ in this flattened space. This effectively makes our 3DMM a deformable geometry image [Gu et al., 2002]. The re-sampled 3DMM that we use in our STN therefore has $N = H'W'$ vertices and each vertex i has an associated UV coordinate (x_i^t, y_i^t). The corresponding sample coordinate produced by the grid generator is given by $(x_i^s, y_i^s) = (Y''_{1,i}, Y''_{2,i})$.

Bilinear Sampling We use bilinear sampling exactly as in the original STN such that the re-sampled image V_i^c at location (x_i^t, y_i^t) in colour channel c is given by:

$$V_i^c = \sum_{j=1}^{H} \sum_{k=1}^{W} I_{jk}^c \max(0, 1 - |x_i^s - k|) \max(0, 1 - |y_i^s - j|), \tag{4.13}$$

where I_{jk}^c is the value in the input image at pixel (j, k) in colour channel c. I has height H and width W. This bilinear sampling is differentiable (see [Jaderberg et al., 2015] for derivatives) and so the loss can be backpropagated through the sampler and back into the grid generator.

Self-occlusions Since the 3DMM produces a 3D mesh, parts of the mesh may be self-occluded. The occluded vertices can be computed exactly using ray-tracing or z-buffering or they can be precomputed and stored in a lookup table. For efficiency, we approximate occlusion by only computing which vertices have backward facing normals.

This approximation would be exact for any object that is globally convex. For objects with concavities, the approximation will underestimate the set of occluded vertices. Faces are typically concave around the eyes, the nose boundary and the mouth interior but we find that typically only around 5% of vertices are mislabelled and the accuracy is sufficient for our purposes.

This layer takes as input the rotation matrix $\mathbf{R}$ and the shape parameters $\boldsymbol{\alpha}$ and outputs a binary occlusion mask $\mathbf{M} \in \{0, 1\}^{H' \times W'}$. The occlusion function is binary and hence not differentiable at points where the visibility of a vertex changes; everywhere else the gradient is zero. Hence, we simply pass back zero gradients:

$$\frac{\partial \mathbf{M}}{\partial \boldsymbol{\alpha}} = 0, \quad \frac{\partial \mathbf{M}}{\partial \mathbf{R}} = 0. \tag{4.14}$$

Note that this means that the network is not able to learn how changes in occlusion help to reduce the loss. Occlusions are applied in a forward pass but changes in occlusion do not backpropagate.

Masking Layer The final layer in the sampler combines the sampled image and the visibility map via pixel-wise products:

$$W_i^c = V_i^c M_{x_i^t, y_i^t}, \quad \frac{\partial W_i^c}{\partial V_i^c} = M_{x_i^t, y_i^t}, \quad \frac{\partial W_i^c}{\partial M_{x_i^t, y_i^t}} = V_i^c. \tag{4.15}$$

4.3.4 Geometric Losses for Localiser Training

An STN is usually inserted into a network as a preprocessor of input images and its output is then passed to a classification or regression CNN. Hence, the pose normalisation that is learnt by the STN is the one that produces optimal performance on the subsequent task. In the context of a 3D morphable face model, an obvious task would be face recognition. While this is certainly worth pursuing, we have observed that the optimal normalisation for recognition may not correspond to the correct model-image correspondence one would expect. For example, if context provided by hair and clothing helps with recognition, then the 3DMM-STN may learn to sample this.

Instead, we show that it is possible to train an STN to perform accurate localisation using only some simple geometric priors without even requiring identity labels for the images. We describe these geometric loss functions as follows.

Bilateral Symmetry Loss Faces are approximately bilaterally symmetric. Ignoring the effects of illumination, this means that we expect sampled face textures to be approximately bilaterally symmetric. We can define a loss that measures the asymmetry of the sampled texture over visible pixels:

$$\ell_{\text{sym}} = \sum_{i=1}^{N} \sum_{c=1}^{3} M_{x_i^t, y_i^t} M_{x_{\text{sym}(i)}^t, y_i^t} (V_i^c - V_{\text{sym}(i)}^c)^2, \tag{4.16}$$

where $\text{sym}(i)$ is the index of the pixel with location $(W' + 1 - x_i^t, y_i^t)$ in the resampled image.

Siamese Multi-view Fitting Loss If we have multiple images of the same face in different poses (or equivalently from different viewpoints), then we expect that the sampled textures will be equal (again, neglecting the effects of illumination). If we had such multiview images,

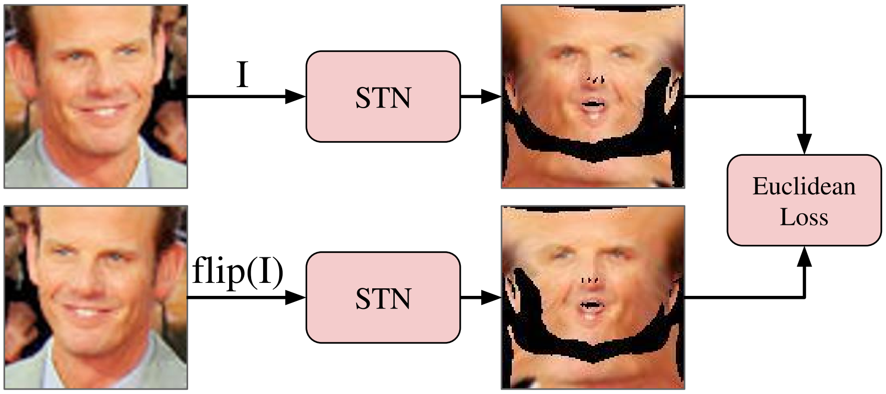


Figure 4.4: Siamese multiview loss. An image and its horizontal reflection yield two sampled images. We penalise differences in these two images.

this would allow us to perform Siamese training where a pair of images in different poses were sampled into images V_i^c and W_i^c with visibility masks $\mathbf{M}$ and $\mathbf{N}$ giving a loss:

$$\ell_{\text{multiview}} = \sum_{i=1}^{N} \sum_{c=1}^{3} M_{x_i^t, y_i^t} N_{x_i^t, y_i^t} (V_i^c - W_i^c)^2. \tag{4.17}$$

Ideally, this loss would be used with a multiview face database or even a face recognition database where images of the same person in different in-the-wild conditions are present. We use an even simpler variant which does not require multiview images; again based on the bilateral symmetry assumption. A horizontal reflection of a face image approximates what that face would look like in a reflected pose. Hence, we perform Siamese training on an input image and its horizontal reflection. This is different to the bilateral symmetry loss and is effectively encouraging the localiser to behave symmetrically.

Landmark Loss As has been observed elsewhere [Tewari et al., 2017], convergence of the training can be speeded up by introducing surrogate loss functions that provide supervision in the form of landmark locations. It is straightforward to add a landmark loss to our network. First, we define a selection layer that selects $L < N$ landmarks from the N 2D points outputted by the grid generator:

$$\mathbf{L} = \mathbf{Y}''\mathbf{S}, \tag{4.18}$$

where $\mathbf{S} \in \{0,1\}^{N \times L}$ is a selection matrix with $\mathbf{S}^T\mathbf{S} = \mathbf{I}_L$. Given L landmark locations

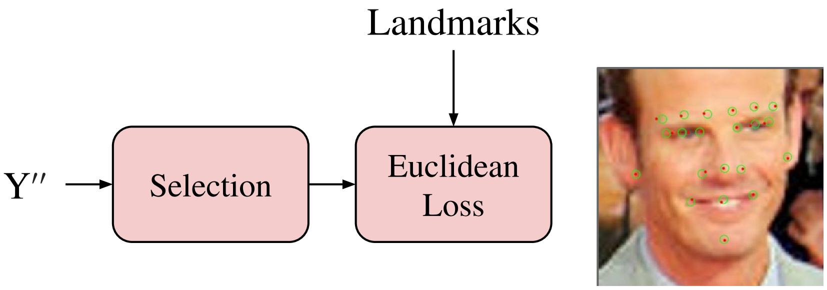


Figure 4.5: Landmark loss. The diagram shows the implementation of the regression layer that computes the Euclidean distance between selected 2D points and ground truth positions (left). Predicted positions are in red and landmark positions are in green (right).

$\mathbf{l}_1, \dots, \mathbf{l}_L$ and associated detection confidence values $w_1, \dots, w_L$, we computed a weighted Euclidean loss:

$$\ell_{\text{landmark}} = \sum_{i=1}^{L} w_i \|\mathbf{L}_i - \mathbf{l}_i\|^2. \tag{4.19}$$

Landmarks that are not visible (i.e. were not hand-labelled or detected) are simply assigned zero confidence.

Statistical Prior Loss The statistical shape model provides a prior. We scale the shape basis vectors such that the shape parameters follow a standard multivariate normal distribution: $\boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$. Hence, the statistical prior can be encoded by the following loss function:

$$\ell_{\text{prior}} = \|\boldsymbol{\alpha}\|^2. \tag{4.20}$$

4.3.5 Experimental Results

Figure 4.6 shows the pipeline of an image passing through a 3DMM-STN. For our statistical shape model, we use $D = 10$ dimensions of which five are the first five (identity) principal components from the Basel Face Model [Paysan et al., 2009]. The other five are expression components which come from FaceWarehouse [Cao et al., 2014b] using the correspondence to the Basel Model provided by [Zhu et al., 2016]. We re-mesh the Basel Model over a uniform

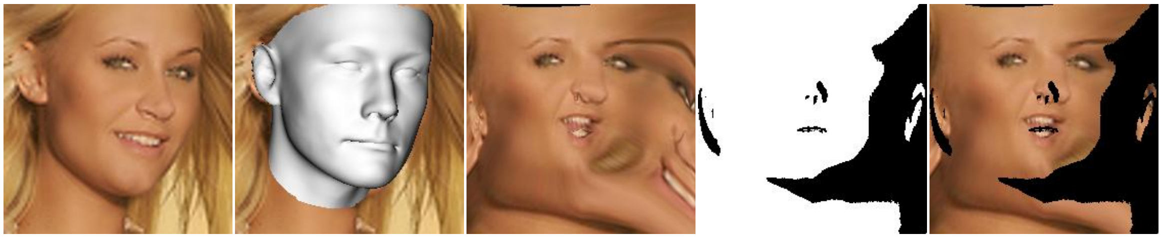


Figure 4.6: Overview of the 3DMM-STN. Input image, rendering of estimated shape in estimated pose, sampled image, occlusion mask, final output of 3DMM-STN (from left to right).

grid of size 224 × 224. We trained our 3DMM-STN with the four loss functions described in Subsection 4.3.4 using the AFLW dataset [Martin Koestinger and Bischof, 2011]. This provides up to 21 landmarks per subject for over 25k in-the-wild images. This is a relatively small dataset for training a deep network so we perform fine-tuning by setting the learning rate on the last layer of the localiser to four times that of the rest of the network.

A by-product of the trained 3DMM-STN is that it can also act as a 2D landmark localiser. After training, the localiser achieves an average landmarking error of 2.35 pixels on the part of AFLW used as validation set, over the 21 landmarks, showing that the training converges well overall.

We begin by demonstrating that our 3DMM-STN learns to predict consistent correspondence between model and image. In Figure 4.7, we show 3DMM-STN output for multiple images of the same person. Note that the features are consistently mapped to the same location in the transformed output.

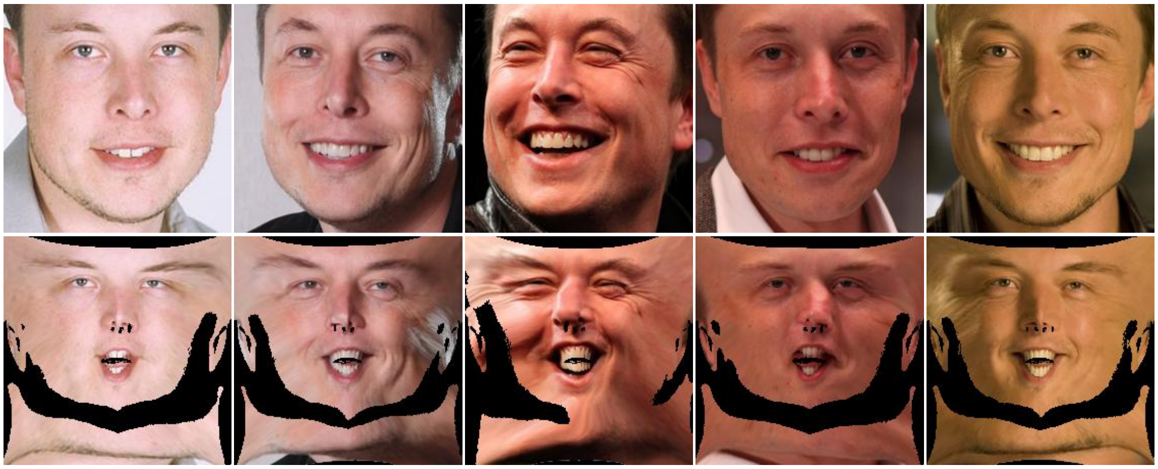
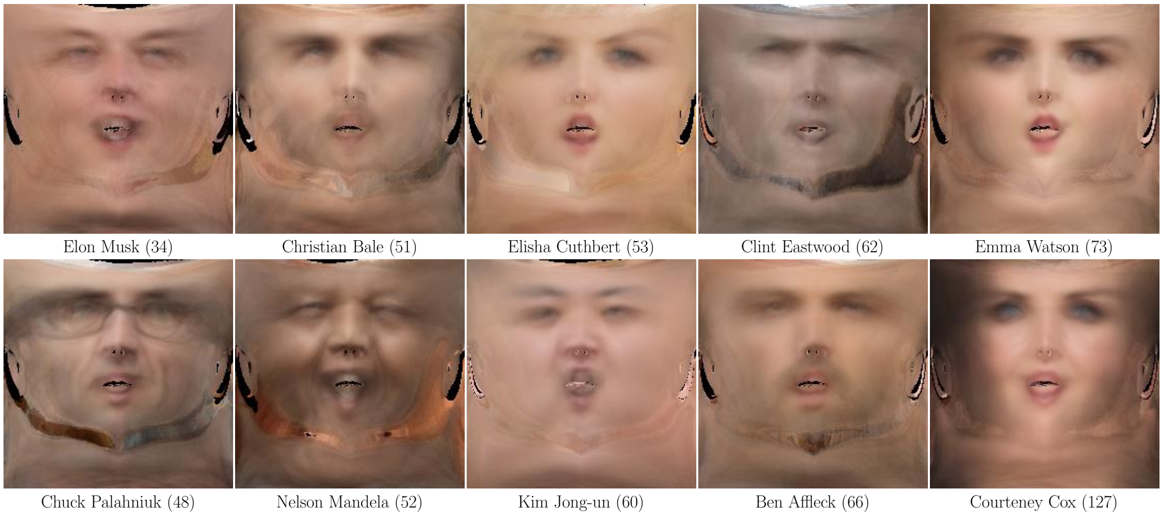


Figure 4.7: 3DMM-STN output for multiple images of the same person in different poses.



Elon Musk (34) Christian Bale (51) Elisha Cuthbert (53) Clint Eastwood (62) Emma Watson (73)

Chuck Palahniuk (48) Nelson Mandela (52) Kim Jong-un (60) Ben Affleck (66) Courteney Cox (127)

Figure 4.8: A set of mean flattened images per subject. Real images are obtained from the UMDFaces dataset. The number of images that are used for averaging is stated next to subject's name.

In Figure 4.8, we go further by applying the 3DMM-STN to multiple images of the same person and then average the resulting transformed images. We show results for 10 subjects from the UMDFaces dataset [Bansal et al., 2017]. The number of images for each subject is shown in parentheses. The averages have well-defined features despite being computed from images with large pose variation.

In Figure 4.9, we provide a qualitative comparison to [Tran et al., 2017]. This is the only previous work on 3DMM fitting using a CNN for which the trained network is made publicly available. In columns one and five, we show input images from the AFLW dataset. In columns two and six, we show the reconstruction provided by [Tran et al., 2017]. While the reconstruction captures the rough appearance of the input face, it lacks the discriminating detail of the original image. This method regresses shape and texture directly but not illumination or pose. Hence, we cannot directly compare the model-image correspondence provided by this method. To overcome this, we use the landmark detector used by [Tran et al., 2017] during training and compute the optimal pose to align their reconstruction to these landmarks. We replace their cropped model with the original BFM shape model and sample the image. This allows us to create the flattened images in columns three and seven. The output of our proposed 3DMM-STN is shown in columns four and eight. We note that our approach less frequently samples background and yields a more consistent

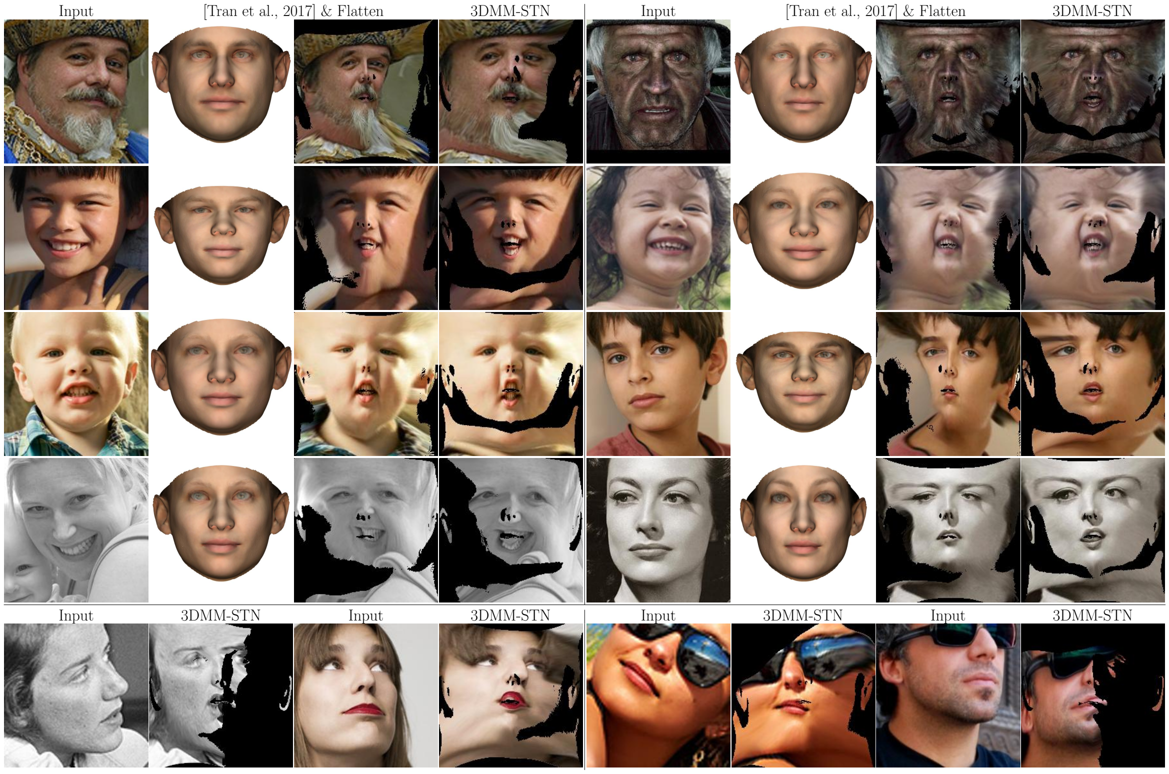


Figure 4.9: Qualitative comparison to [Tran et al., 2017]. The bottom row shows examples for which [Tran et al., 2017] failed to fit due to failure of the landmark detector.

correspondence of the resampled faces. In the bottom row of the figure we show challenging examples where [Tran et al., 2017] did not produce any output because the landmark detector failed. Despite occlusions and large out-of-plane rotations, the 3DMM-STN still does a good job of producing a normalised output image.

4.4 Statistical Transformer Networks

In this section, we address an important drawback of our 3DMM-STN as well as the original STN. The geometric transformation model used by the STNs must be hand-picked and remains fixed. We propose to replace the transformation model by a learnable statistical shape model which we call a Statistical Transformer Network.

The power of statistical shape and appearance models is well known, for example Active Appearance Models [Cootes et al., 2001] in 2D and 3D Morphable Models [Blanz and Vetter, 1999] in 3D. These models can be built from a few hundred or thousand samples and then deployed to solve problems ranging from tracking to recognition to synthesis. Usually,

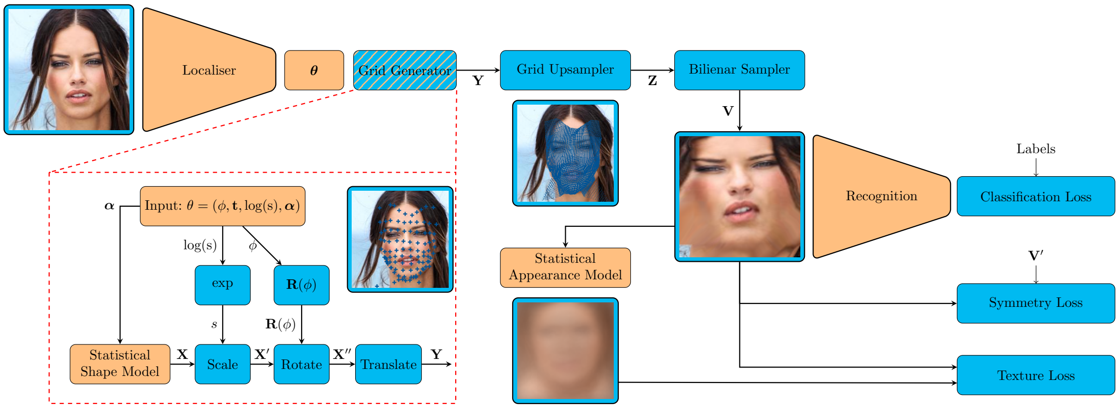


Figure 4.10: Overview of the StaTN. Learnable components of the network are shown in orange.

constructing such models requires hand labelling of landmark points so that correspondence can be established between training samples. Then the variability in shape and appearance is learnt, typically using PCA.

A StaTN learns such a statistical shape model (and optionally a statistical appearance model) with no landmark supervision and also learns to fit the model. Hence, a StaTN learns an explicit representation of a particular object class in an interpretable way (the parameters of the statistical model can be explicitly accessed and understood). We start explaining how each component of a conventional STN must be modified. An overview of our proposed StaTN is shown in Figure 4.10.

4.4.1 Localiser Network

The localiser network is a black box CNN that takes an image (or, more generally, a feature map) as input and regresses a semantically meaningful vector of parameters θ:

$$\theta = (\underbrace{\phi, \mathbf{t}, \log(\mathrm{s})}_{\text{rigid pose}}, \underbrace{\boldsymbol{\alpha}}_{\text{shape}}). \tag{4.21}$$

Similar to the localiser part of the 3DMM-STN (see Subsection 4.3.1), $\mathbf{t} \in \mathbb{R}^2$ is a 2D translation. log(s) is a log scale that later passes through an exponentiation layer, ensuring that the estimated scale, s, is positive. $\phi \in \mathbb{R}$ is a rotation angle and the shape parameters $\boldsymbol{\alpha} \in \mathbb{R}^D$ are the weights of the principal components of the statistical shape model described below. The architecture of the localiser network is not critical. For all our experiments, we use a very simple architecture comprising six blocks of convolution, ReLU and pooling, followed by a fully connected layer with 1024 units followed by the final regression layer implemented as a fully connected layer with $D + 4$ units.

4.4.2 Grid Generator

The purpose of the grid generator is to compute a sampling coordinate (x_i^s, y_i^s) for each corresponding point (x_i^t, y_i^t) in the regular output grid from the transformation parameters provided by the localiser network. The output grid comprises $M = H'W'$ points, regularly sampled over $-1 \dots 1$ with height H' and width W'. Our grid generator begins by generating a shape from a linear shape model (which is learnt as part of the StaTN training), then a rigid transformation is applied to this before it is finally upsampled to a high resolution sampling grid.

Linear Shape Model A linear shape model is an orthonormal basis enabling compact representation of a class of shapes. Specifically, a shape comprised of N 2D vertices, $\mathbf{x} \in \mathbb{R}^{2N}$, is written as a sum of a mean shape $\bar{\mathbf{s}} \in \mathbb{R}^{2N}$ and a linear combination of a set of D orthonormal bases $\mathbf{Q} \in \mathbb{R}^{2N \times D}$:

$$\mathbf{s}(\boldsymbol{\alpha}) = \mathbf{Q}\alpha + \bar{\mathbf{s}}, \tag{4.22}$$

where $\alpha \in \mathbb{R}^D$ is a vector of shape parameters and $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_D$. Typically, such models are built statistically by labelling a set of training images and using PCA to extract the mean and basis vectors. Instead, here we will learn the model in an unsupervised manner simultaneously with learning to fit the model.

Note that the linear model in (4.22) can be interpreted as a fully connected layer of a CNN (or equivalently, a special case of a convolution layer) in which the orthonormal basis plays the role of the filters, the mean shape plays the role of the biases and the parameter vector plays the role of the input feature map. To make this explicit, we rewrite each component of the model in tensor form such that the output shape is $\mathcal{X} \in \mathbb{R}^{1 \times 1 \times 2N}$ and the model is given by the orthonormal basis $Q \in \mathbb{R}^{D \times 1 \times 1 \times 2N}$ and the parameter vector $\alpha \in \mathbb{R}^{D \times 1 \times 1}$. In this form, the familiar definition of a convolution operation yields the same model as (4.22):

$$\mathcal{X}_{i',j',k'} = \bar{\mathbf{s}}_{k'} + \sum_{i,j,k} Q_{i,j,k,k'} \alpha_{i+i',j+j',k}. \tag{4.23}$$

In practice, we implement the linear shape model as a convolution layer and learn the filters and biases as normal. We initialise the biases (mean shape) as a regular square grid and the filters (principal components) as a random orthonormal matrix.

Subsequently, it is notationally convenient to rewrite the output shape and mean shape in matrix form as $\mathbf{X} \in \mathbb{R}^{2\times N}$ and $\bar{\mathbf{S}} \in \mathbb{R}^{2\times N}$.

Scaling Layer The log scale estimated by the localiser is first transformed to scale by an exponentiation layer:

$$s(\log(\mathrm{s})) = \exp(\log(\mathrm{s})), \quad \frac{\partial s}{\partial \log(\mathrm{s})} = \exp(\log(\mathrm{s})). \tag{4.24}$$

Then, the 2D points $\mathbf{X} \in \mathbb{R}^{2\times N}$ are scaled:

$$\mathbf{X}'(s, \mathbf{X}) = s\mathbf{X}, \quad \frac{\partial X'_{i,j}}{\partial s} = X_{i,j}, \quad \frac{\partial X'_{i,j}}{\partial X_{i,j}} = s. \tag{4.25}$$

2D Rotation Matrix Layer This layer outputs a 2D rotation matrix as a function of a rotation angle $\mathbf{R} : \mathbb{R} \mapsto \mathbb{R}^{2\times 2}$:

$$\mathbf{R}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}, \quad \frac{\partial \mathbf{R}}{\partial \phi} = \begin{bmatrix} -\sin\phi & -\cos\phi \\ \cos\phi & -\sin\phi \end{bmatrix}. \tag{4.26}$$

2D Rotation Layer The rotation layer takes as input a rotation matrix $\mathbf{R}$ and N 2D points $\mathbf{X}' \in \mathbb{R}^{2\times N}$ and applies the rotation:

$$\begin{gathered} \mathbf{X}''(\mathbf{R}, \mathbf{X}') = \mathbf{R}\mathbf{X}', \\ \frac{\partial X''_{i,j}}{\partial R_{i,k}} = X'_{k,j}, \quad \frac{\partial X''_{i,j}}{\partial X'_{k,j}} = R_{i,k}, \quad i,k \in \{1,2\}, j \in \{1,\dots,N\}. \end{gathered} \tag{4.27}$$

Translation Layer Finally, the 2D sample points are generated by adding a 2D translation $\mathbf{t} \in \mathbb{R}^2$ to each of the scaled points:

$$\mathbf{Y}(\mathbf{t}, \mathbf{X}'') = \mathbf{X}'' + \mathbf{1}_N \otimes \mathbf{t}, \quad \frac{\partial Y_{i,j}}{\partial t_i} = 1, \quad \frac{\partial Y_{i,j}}{\partial X''_{i,j}} = 1, \tag{4.28}$$

where $\mathbf{1}_N$ is the row vector of length N containing ones and $\otimes$ is the Kronecker product.

4.4.3 Grid Upsampler

The resolution at which we wish to resample the image may be higher than the resolution at which we wish to statistically model shape. For example, in our experiments, our statistical shape model comprises $N = 10 \times 10$ grid vertices whereas our resampled images comprise $M = 112 \times 112$ pixels, i.e. two orders of magnitude more. This keeps the dimensionality of the statistical model (that must be learnt from data) down, whilst still allowing sufficient detail to be sampled from the input images. To achieve this, we precompute the barycentric weights of each high resolution output grid point in the low resolution output grid. We then use these weights to compute sample locations for every high resolution point, $\mathbf{Z} \in \mathbb{R}^{2\times M}$, from the computed low resolution sample grid points. In other words, we perform a linear interpolation of the low resolution sample grid. In practice, this can be written as:

$$\mathbf{Z}(\mathbf{Y}) = \mathbf{Y}\mathbf{W}, \quad \frac{\partial Z_{i,j}}{\partial Y_{i,k}} = W_{k,j}, \quad i \in \{1, 2\}, \, j \in \{1, \dots, M\}, \, k \in \{1, \dots, N\}. \tag{4.29}$$

where $\mathbf{W} \in \mathbb{R}^{N\times M}$ is constant, sparse (each row contains three non-zero values) and each row sums to one: $\mathbf{W}\mathbf{1}_M = \mathbf{1}_N$. The sample points for each point in the output grid are given by $(x_i^s, y_i^s) = (Z_{1,i}, Z_{2,i})$. See Figure 4.10 for a visualisation of the low and high resolution sampling grids overlaid on an input image.

4.4.4 Bilinear Sampling

As stated earlier in this chapter, we use bilinear sampling exactly as in the original STN and 3DMM-STN such that the re-sampled image V_i^c at location (x_i^t, y_i^t) in colour channel c is given by:

$$V_i^c = \sum_{j=1}^{H} \sum_{k=1}^{W} I_{jk}^c \max(0, 1 - |x_i^s - k|) \max(0, 1 - |y_i^s - j|), \tag{4.30}$$

where I_{jk}^c is the value in the input image at pixel (j, k) in colour channel c.

4.4.5 Backpropagation with Manifold Gradient Descent

In a StaTN, some learnable parameters are subject to constraints. If, during backpropagation, an unconstrained step in the direction of the negative gradient of the loss function is taken, then these parameters will no longer satisfy the constraints. In this subsection, we show how manifold gradient descent can be used to ensure the constraints on learnable parameters remain satisfied during training.

Constrained Parameters In our network, the shape model is subject to such constraints and hence requires special treatment during training. First, the shape basis is required to be orthonormal, i.e. that $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_D$. Second, we require that the mean shape is centred, i.e. that $\bar{\mathbf{S}}\mathbf{1}_N = \mathbf{I}_2$. Otherwise there is an ambiguity between the translation estimated by the localiser, $\mathbf{t}$, and the centering of the mean (i.e. the same shape can be obtained by translating the mean or translating the output shape from our model). Without constraint, this gives SGD redundant search directions during training.

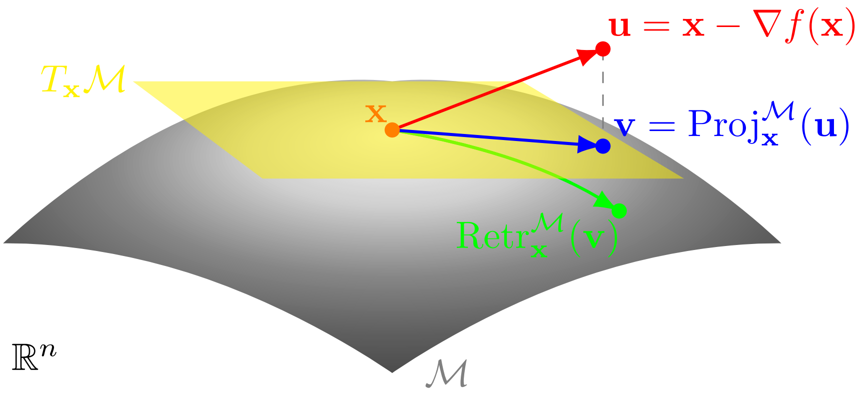

$\mathbf{u} = \mathbf{x} - \nabla f(\mathbf{x})$
$T_\mathbf{x}\mathcal{M}$
$\mathbf{x}$
$\mathbf{v} = \mathrm{Proj}_\mathbf{x}^\mathcal{M}(\mathbf{u})$
$\mathrm{Retr}_\mathbf{x}^\mathcal{M}(\mathbf{v})$
$\mathbb{R}^n$
$\mathcal{M}$

Figure 4.11: Manifold optimisation: the Euclidean descent direction $-\nabla f(\mathbf{x})$ is transformed to the tangent plane at $\mathbf{x}$, $T_\mathbf{x}\mathcal{M}$, via orthogonal projection and then to the manifold $\mathcal{M} \subset \mathbb{R}^n$ via a retraction.

Both of these constraints can be encoded by viewing the parameters as belonging to a Riemannian manifold and using manifold optimisation for these parameters during training. This idea is not new and has been considered, for example, in [Harandi and Fernando, 2016]. Here, we show how to use manifold optimisation for the two model parameters in our STN that are subject to constraints.

Manifold Gradient Descent Suppose $M \subset \mathbb{R}^n$ is a Riemannian manifold embedded in $\mathbb{R}^n$ and $f : \mathbb{R}^n \mapsto \mathbb{R}$ a cost function on $\mathbb{R}^n$. If $\mathbf{x} \in \mathbb{R}^n$ is some learnable parameter then $-\nabla f(\mathbf{x})$ is a (Euclidean) descent direction for $\mathbf{x}$. In practice, this Euclidean gradient would be provided by backpropagation. Usually, some variation of stochastic gradient descent is used to reduce the loss by taking a step in the negative gradient direction. However, if our learnable parameters are subject to constraints then taking a step in the unconstrained gradient direction will lead to parameters that do not satisfy the constraints.

Manifold optimisation (see Figure 4.11) relies on two operations: *orthogonal projection* from the ambient space to the tangent space of the manifold and *retraction* to transform from the tangent space onto the manifold. The Euclidean gradient computed via backpropagation is first projected to the tangent space, then a retraction is applied to this tangent vector, giving a new point on the manifold. Note that the geometric exponential map is a particular kind of retraction but often we can use alternatives that are cheaper to compute.

Centred matrices manifold: The mean shape must lie on the manifold of centred matrices $C_{m,n} = \{\mathbf{X} \in \mathbb{R}^{m \times n} | \mathbf{X}\mathbf{1}_n = \mathbf{0}_m\}$. Specifically, $\bar{\mathbf{S}} \in C_{2,N}$. Without this constraint there is a translational ambiguity between the translation vector $\mathbf{t}$ and the mean shape. Projection and retraction on this manifold are particularly simple. The orthogonal projection $\text{Proj}_{\mathbf{X}}^{C_{m,n}} : \mathbb{R}^{m \times n} \mapsto T_{\mathbf{X}}C_{m,n}$ of a displacement $\mathbf{U} \in \mathbb{R}^{m \times n}$ in the ambient space onto the tangent space $T_{\mathbf{X}}C_{m,n}$ at $\mathbf{X}$ is obtained simply by centering $\mathbf{U}$:

$$\text{Proj}_{\mathbf{X}}^{C_{m,n}}(\mathbf{U}) = \mathbf{U} - \mathbf{U}\mathbf{1}_n. \tag{4.31}$$

The retraction $\text{Retr}_{\mathbf{X}}^{C_{m,n}} : T_{\mathbf{X}}C_{m,n} \mapsto C_{m,n}$ of a tangent vector $\mathbf{V} \in T_{\mathbf{X}}C_{m,n}$ is simply:

$$\text{Retr}_{\mathbf{X}}^{C_{m,n}}(\mathbf{V}) = \mathbf{X} + \mathbf{V}. \tag{4.32}$$

So, we initialise with a centred shape then, when updating the mean shape during SGD, we simply centre the gradient provided by backpropagation before adding it to the current mean shape.

Stiefel manifold: The orthonormal shape basis must lie on the Stiefel manifold $V_k(\mathbb{R}^n) = \{\mathbf{X} \in \mathbb{R}^{n \times k} | \mathbf{X}^T\mathbf{X} = \mathbf{I}_k\}$. This is the manifold of k dimensional orthonormal bases in $\mathbb{R}^n$. Specifically, $\mathbf{Q} \in V_D(\mathbb{R}^{2N})$. The orthogonal projection $\text{Proj}_{\mathbf{X}}^{V_k(\mathbb{R}^n)} : \mathbb{R}^{n \times k} \mapsto T_{\mathbf{X}}V_k(\mathbb{R}^n)$ of a displacement $\mathbf{U} \in \mathbb{R}^{n \times k}$ in the ambient space onto the tangent space $T_{\mathbf{X}}V_k(\mathbb{R}^n)$ at $\mathbf{X}$ is given by:

$$\text{Proj}_{\mathbf{X}}^{V_k(\mathbb{R}^n)}(\mathbf{U}) = \mathbf{U} - \mathbf{X}\,\text{sym}(\mathbf{X}^T\mathbf{U}), \tag{4.33}$$

where $\text{sym}(\mathbf{M}) = 0.5(\mathbf{M} + \mathbf{M}^T)$. A retraction $\text{Retr}_{\mathbf{X}}^{V_k(\mathbb{R}^n)} : T_{\mathbf{X}}V_k(\mathbb{R}^n) \mapsto V_k(\mathbb{R}^n)$ of a tangent vector $\mathbf{V} \in T_{\mathbf{X}}V_k(\mathbb{R}^n)$ can be obtained by finding the closest orthogonal matrix to $\mathbf{V}$:

$$\text{Retr}_{\mathbf{X}}^{V_k(\mathbb{R}^n)}(\mathbf{V}) = \mathbf{U}, \tag{4.34}$$

where $\mathbf{V} = \mathbf{U}\mathbf{P}$ is the polar decomposition of $\mathbf{V}$.

Implementation In practice, we make a small modification to the implementation of backpropagation. Where layer parameters are updated, we test whether the layer is one with constraints. If it is, we apply projection and retraction before the updates are added to the parameters.

4.4.6 Losses for Training a StaTN

As with the original STN, a StaTN can be used as a component within a larger network that is trained end-to-end. In this section, we consider some different ways that this can be achieved and design loss functions that help the StaTN learn a meaningful statistical model.

Learning by Task The most obvious way to use a StaTN is as part of a network that is trained to solve a task such as recognition or classification. Here, the StaTN acts to normalise the effects of pose and shape, making the subsequent task easier to solve. Concretely, the output of the StaTN (i.e. the resampled image) is fed to a classification network with, for example, its own softmax loss (see Figure 4.10). This loss is propagated back through the classification network, through the resampler and into the statistical shape model and the localisation networks.

In this setting, the StaTN will learn a notion of correspondence that is optimal for the task being solved. This may not coincide with intuitive notions of correspondence, nor will attention necessarily focus only on the object of interest. For example, if training for face recognition, a StaTN may learn that there is important contextual information in clothing or background and so the statistical model (and hence sample grid) may not attend only to the face. In our experiments, we use a softmax classification loss, ℓ_{class}, in the context of a face classification task.

Appearance Model with an Autoencoder We now propose a loss that can be used to train a StaTN in a much more general setting. Inspired by the minimum description length principle [Davies et al., 2002], which is based on the idea that the correct correspondence is the one that leads to the best compression of the data, we measure a loss as the reconstruction error of the images resampled by a StaTN in order to learn a statistical appearance model. This loss will be minimised by the network learning to establish correspondence that leads to the most compressible appearance. An image collection containing images of a particular object class but no further information, i.e. not even having identity labels for each image, would be suitable for this type of unsupervised training. Intuitively, the StaTN then searches for objects with the most redundant appearance in the image collection.

Specifically, we learn a linear statistical appearance model of the resampled images $\mathbf{V} \in \mathbb{R}^{H'W' \times c}$, where c is the number of colour channels (usually the StaTN and hence the appearance model will be applied to RGB image input and so $c = 3$, however in general this approach could be applied to feature maps with any number of channels). We use the same linear model as in (4.22), where $\bar{\mathbf{s}}$ is the mean texture and $\mathbf{Q}$ the texture principal components. The projection of an input image onto the model is given by:

$$\mathbf{w} = \mathbf{Q}\mathbf{Q}^T(\text{vec}(\mathbf{V}) - \bar{\mathbf{s}}) + \bar{\mathbf{s}}. \tag{4.35}$$

This is simply a linear autoencoder. A more complex, nonlinear autoencoder could be used here for the texture model but it has been shown many times previously that a linear model is an efficient representation for the appearance of many object classes [Cootes et al., 2001]. The texture loss is then given by the squared Euclidean distance between the source and reconstructed textures:

$$\ell_{\text{tex}} = \|\mathbf{w} - \text{vec}(\mathbf{V})\|^2. \tag{4.36}$$

The principal components of the texture model are subject to the same orthogonality constraint as the shape model, i.e. they lie on the Stiefel manifold. Hence, we use the same manifold optimisation strategy for these parameters as in Subsection 4.4.5. The mean texture does not need constraining since there is no texture translation to cause an ambiguity. Note that, when trained in this way, a StaTN is effectively learning an Active Appearance Model [Cootes et al., 2001] and the means to fit the model to an image with no supervision.

Regularisation Besides the above two losses, we may wish to regularise the process of training a StaTN such that the obtained shape and appearance models exhibit desirable properties.

Symmetry loss: Many natural and man-made objects exhibit bilateral symmetry. Usually, statistical shape and appearance models would be symmetric by construction since the chosen landmarks would be symmetric. However, we do not use landmarks and neither the classification loss nor the texture loss require this to be the case. To encourage a symmetric model we penalise asymmetry, measured as the difference between a sampled image and its reflection:

$$\ell_{\text{sym}} = \sum_{i=1}^{M} \sum_{c=1}^{3} (V_i^c - V_{\text{sym}(i)}^c)^2, \tag{4.37}$$

where $V_{\text{sym}(i)}^c$ is the value in the resampled image at location $(W' + 1 - x_i^t, y_i^t)$. This ignores the effect of illumination (which may introduce asymmetries in appearance) but is still a useful regulariser when averaged over batches.

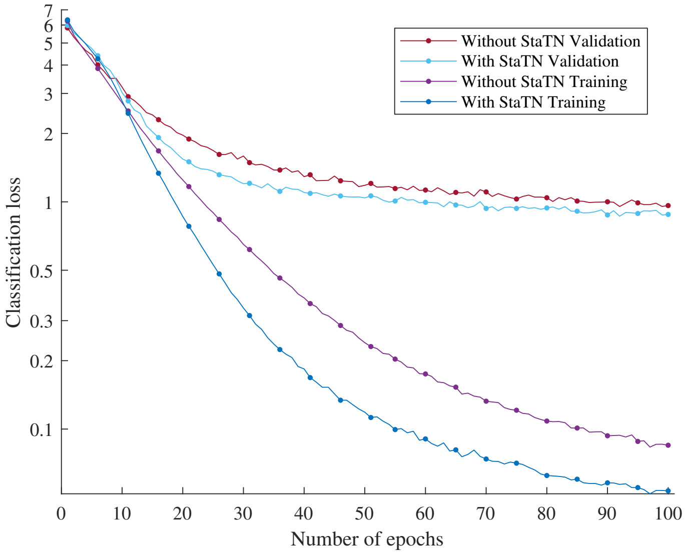


Figure 4.12: Training and validation curves with/without StaTN. A level of improvement can be seen in validation performance, even though the proposed network uses less information.

Area loss: When training without a classification loss, i.e. using only the texture loss, a trivial solution is to collapse the grid to a single pixel. This makes the appearance constant and hence compressible. To avoid this we propose a second regularisation. In a triangulation of our sample grid, we would like the area of the triangles to be preserved (i.e. not collapse to zero). More generally, we would like our shape model to be diffeomorphic, i.e. avoid triangles folding over themselves. Hence, for a sample grid we compute the signed area, a_t, for each triangle t and penalise areas close to zero or that are negative (i.e. have flipped) as follows:

$$\ell_{\text{area}} = \sum_t \max(0, \exp(-a_t) - k), \tag{4.38}$$

where $0 < k \leq 1$ is a constant which determines how small a triangle must be before the penalty is applied. $k = 1$ means only negative areas are penalised. k close to zero means even large triangle areas are penalised. We use a value of $k = 0.99$ in our experiments.

Hybrid Loss In our experiments, we use a hybrid loss function comprising a weighted sum of the four losses (where a loss is switched off by setting the corresponding weight to zero):

$$\ell = w_{\text{class}}\ell_{\text{class}} + w_{\text{tex}}\ell_{\text{tex}} + w_{\text{sym}}\ell_{\text{sym}} + w_{\text{area}}\ell_{\text{area}}. \tag{4.39}$$

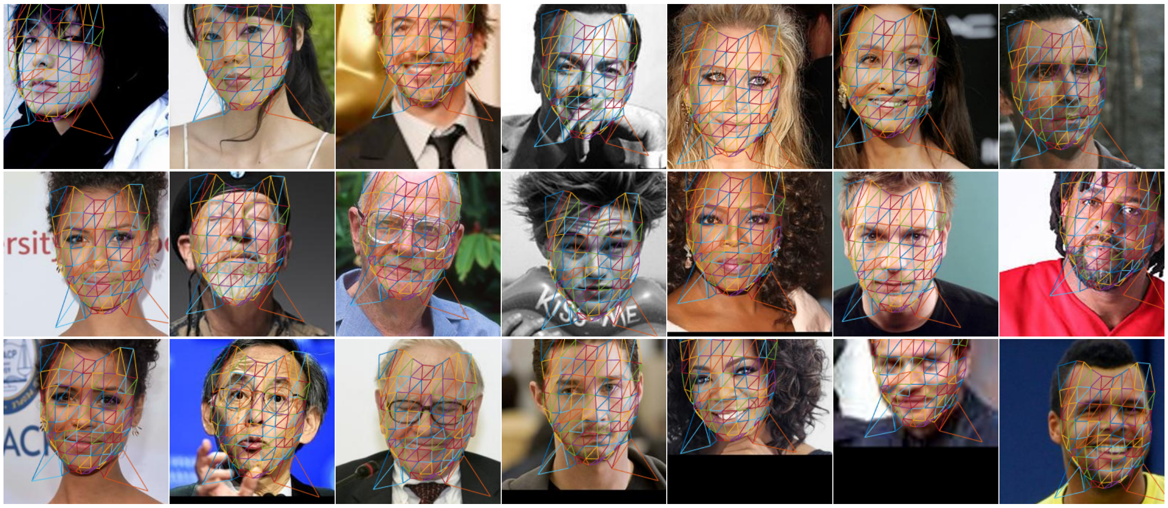


Figure 4.13: Qualitative StaTN fitting results. We show a triangulation of the low resolution sample grid predicted by the grid generator. The deformable grid and fitting process have been learnt from scratch in an end-to-end trained face classification network with no landmark supervision.

4.4.7 Experimental Results

We use the UMDFaces dataset [Bansal et al., 2017] in our experiments. We choose 750 identities with the highest number of images, comprising 61311 in total. This is a rather small dataset; however, we apply random cropping on images and batch normalisation to the convolutional layers as data augmentation.

We follow very similar architecture (5 or 6 convolutional layers with ReLU and pooling followed by a fully connected layer) for our localiser and recognition parts of the network. We use 10 dimensions for both our statistical shape and appearance models. We trained our network with classification, texture and symmetry losses. The learning rates of the localiser and recognition layers are 0.001 whereas shape and texture layers are 0.01 and 1, respectively.

In Figure 4.12, we show the training and validation curves for our proposed StaTN network and an equivalent recognition network without spatial transformation. There is a modest but clear improvement in validation performance, even though our proposed network performs recognition with less information. (Since the grid is smaller than the image, part of the image data is discarded prior to recognition.)

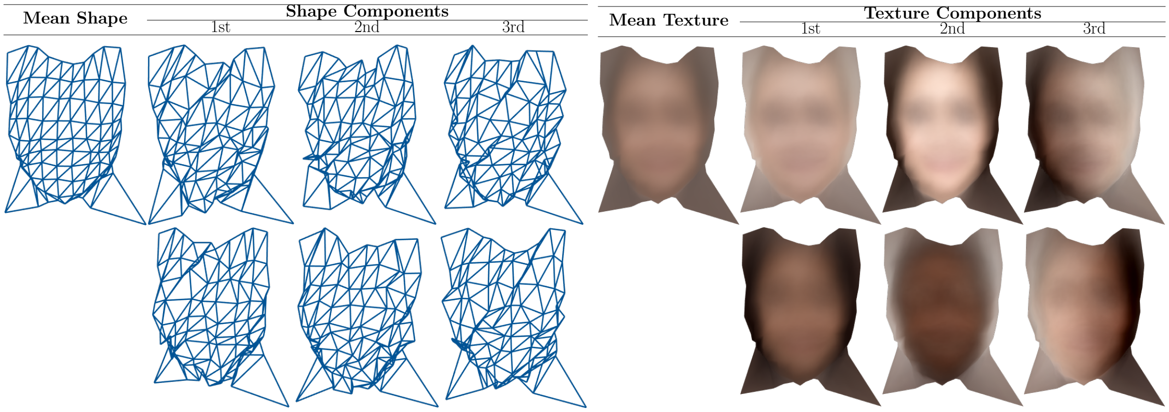


Figure 4.14: Shape and appearance models learnt whilst training a StaTN on the dataset shown in Figure 4.13.

Figure 4.13 shows qualitative fitting results predicted by our network's grid generator. The sparse grid successfully locates the face even in images that are highly cluttered, noisy and badly cropped. Note that we trained our network without any supervision.

In Figure 4.14, we show the shape and appearance model learnt by our network. The shape model clearly resembles a face shape. Interestingly, the shape model does not appear to include the ears, but does sample a region of the shoulders and neck. The texture model has clearly interpretable principal components. The first two capture global lighting or skin colour changes; the third captures side to side lighting variation. The shape components are less easily interpretable but the second mode appears to capture side-to-side 3D rotation of the face.

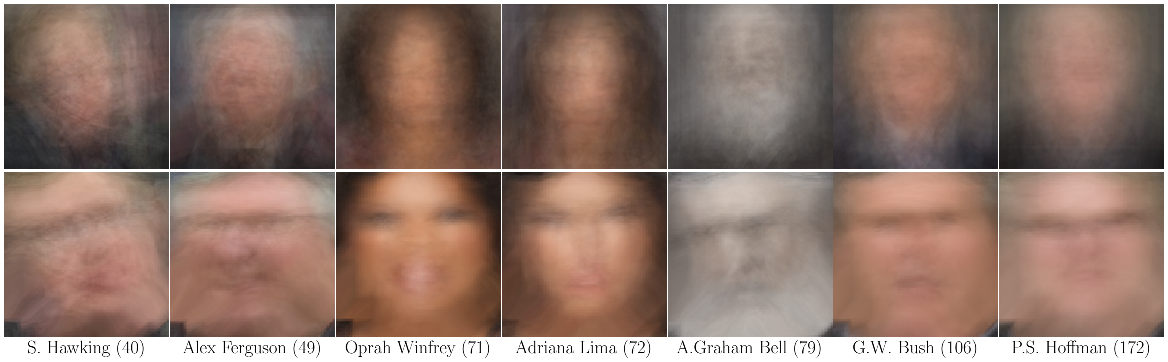


Figure 4.15: A set of averaged images per subject from the UMDFaces dataset. Averaging raw face images of the same person (top row). Images that are obtained by applying the StaTN to multiple images of the same person (bottom row). The number of images that are used for averaging is stated next to the subject's name.

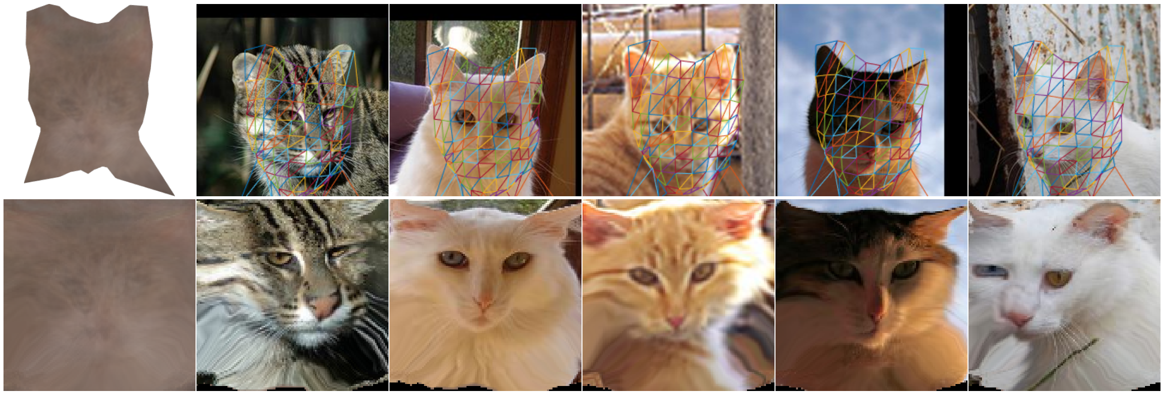


Figure 4.16: Transfer learning on the CAT dataset [Zhang et al., 2008]. The first image of the first row shows the shape model with the mean texture. The other images in the first row illustrate the sparse grid fitting results. The first image of the second row shows the mean texture only. The other images in the second row illustrate the output of the bilinear sampler of our network.

In Figure 4.15, we apply the StaTN to multiple images of the same person and then average the output of the bilinear sampler of our network. We show comparison between the raw average (top row) and the sampled average (bottom row). The number of images for each subject is shown in parentheses. The averages of the resampled images are much sharper and more recognisable than the averages of the raw images. This shows that the StaTN is successfully establishing correspondence between the images.

Finally, in Figure 4.16, we show results for a completely unsupervised dataset. Here, we train on 10k images from the CAT dataset [Zhang et al., 2008]. These images have no identity labels so we use only the texture and regularisation losses. We initialise with the face network trained in the previous experiment and fine-tune. Again, the network learns to consistently fit a meaningful grid to each image and constructs a plausible appearance model.

4.5 Conclusions

In this chapter, we have presented two approaches that attempt to combine model and learning-based computer vision. In the first part, we have shown how to use a 3D morphable model as a spatial transformer within a CNN. The network (specifically, the localiser part of the network) learns to fit a 3D morphable model to a single 2D image without needing labelled examples of fitted models. Since the problem of fitting a morphable model to an image is an unsolved problem (and therefore no existing algorithm could be assumed to provide reliable ground truth fits), this kind of unsupervised learning is desirable.

The morphable model itself is fixed in our first architecture. However, there is no reason that this could not also be learnt. In the second part, we have tried to demonstrate this. By incorporating an explicit shape and appearance model along with a rigid transformation model, our StaTN network is able to explicitly learn dense, non-rigid correspondence. Moreover, the shape, appearance and pose parameters are interpretable and the shape and appearance models form components that can be reused in other networks or other settings. This reduces the “black box” nature of a CNN to some extent.

Using a StaTN as part of a network that is learning to solve a task, e.g. with a classification loss, then the network learns a notion of correspondence that is optimal for that task. This may be revealing about what information is most important for solving a particular task. When the texture loss is used in conjunction with learning a task, then the network learns to trade off sampling more of the image (and potentially sampling useful contextual information in the background) against attending to more easily compressible objects in the image. When the texture loss is used on its own, the network seeks the most compressible object class present in the training images.

Chapter 5

Limits and Ambiguities

5.1 Introduction

In the previous two chapters, we presented methods for fitting a 3D morphable model to a single image using only geometric cues (specifically, landmarks, edges and bilateral symmetry). In doing so, we did not consider whether this is a well-posed problem, we did not model uncertainty and we did not reveal ambiguities in the reconstruction.

In this chapter, we seek to recover a subspace of possible 3D face shapes that are all consistent with the 2D data, rather than try to explain 2D geometric data with a single, best-fitting 3D face. "Consistent" here means that the model explains the data within the tolerance with which we can hope to locate these features within a 2D image. For example, state-of-the-art automatic face landmarking provides a mean landmark error under 4.5% of interocular distance for only 50% of images (according to the second conduct of the 300 faces in-the-wild challenge [Sagonas et al., 2016]). We show how to compute this subspace and show that it contains very significant shape variation.

The ambiguity arises for two reasons. The first is that, within the space of possible faces (as characterised by a 3D morphable model) there are degrees of flexibility that do not change the 2D geometric information when projection parameters are fixed (this applies to both orthographic and perspective projection).

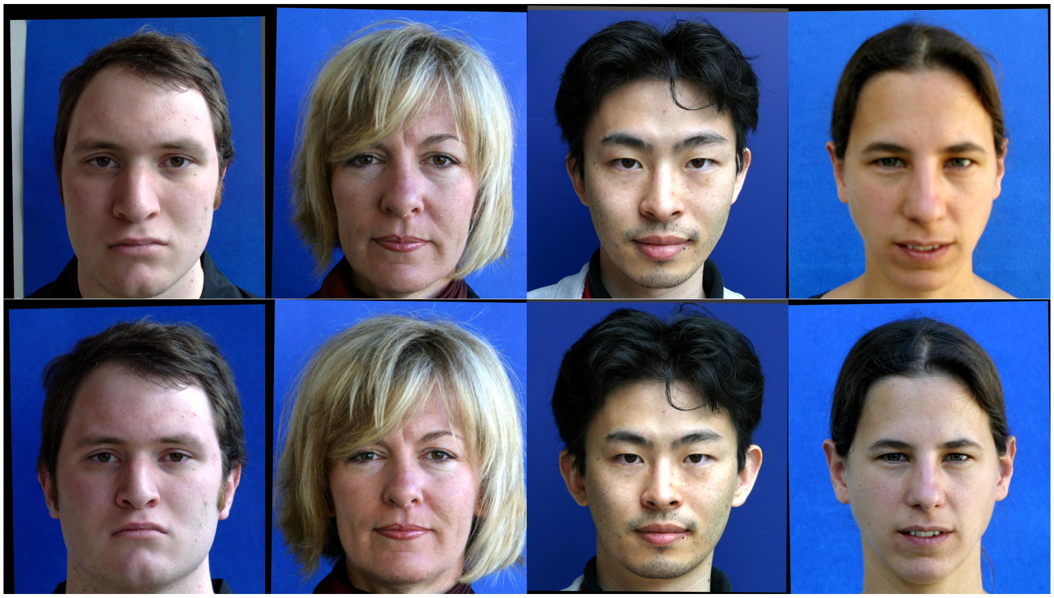


Figure 5.1: Perspective transformation of real faces from the CMDP dataset [Burgos-Artizzu et al., 2014]. The subject is the same in each column and the same camera and lighting is used. The change in viewing distance (60cm top row, 490cm bottom row) causes a significant difference in projected shape.

The second is caused by the nonlinear effect of perspective. When a human face is viewed under perspective projection, its 2D shape varies depending on the subject-camera distance. This effect distorts the relative distances between facial features; the features closest to the camera (e.g. nose) appear larger and those furthest to the camera (e.g. ears) appear smaller with respect to the rest of the face. The face shape appears elongated in general. Figure 5.1 shows the effect of perspective transformation. Cropped faces taken of subjects at 60cm and 490cm are rescaled in order to keep the interocular distance uniform. This effect leads to the second ambiguity, namely that two different (but natural) 3D face shapes viewed at different distances can give rise to the same 2D geometric features.

5.2 Contributions

In this chapter, we show that 2D geometric information only provides a partial constraint on 3D face shape. In other words, face landmarks or occluding contours are an ambiguous shape cue. We use real face images to verify that the ambiguity is present in actual faces. We show that, on average, 2D geometry is more similar between different faces viewed at the same distance than it is between the same face viewed at different distances. We present quantitative and qualitative results on synthetic 2D geometric data created by projection of real 3D scans.

[Smith, 2016] previously considered the effect of perspective projection under the assumption of fixed rotation and translation. The model fitting process was based on landmark points only. We go further by also considering orthographic projection and showing how to compute flexibility modes. Moreover, we improve the model fitting process by posing as a separable nonlinear least squares problem, including solving for rotation and translation. We present more comprehensive experimental results to demonstrate the ambiguities. Finally, we consider not only landmarks but also show how to fit to contours where model-image correspondence is not known.

We verify that multiple explanations of observed 2D shape features are possible. This is the case even for dense data, i.e. where the 2D position of every vertex in the face mesh is known. We show that two faces with significantly different 3D shape can produce almost identical 2D landmarks. This means that the differences are much smaller than the accuracy of either human or machine labelled landmarks. We compute these flexibility modes under both orthographic or perspective projection. We also present qualitative results on real images from the Caltech Multi-Distance Portraits (CMDP) dataset [Burgos-Artizzu et al., 2014].

5.3 Model Fitting

In order to demonstrate the geometric ambiguity, we use algorithms proposed in Chapter 3 for fitting a 3DMM to 2D geometric information and extracting the subspace of possible 3D shapes. There is no need to repeat here the details of the separable nonlinear least squares optimisation or orthographic/perspective projection models, since we explained them thoroughly in Chapter 3. However, we include some of the key equations as a quick reference for the reader. Hence, we begin by investigating the perspective ambiguity and flexibility modes.

5.4 Perspective Ambiguities

The objective for the model fitting is to find the shape, pose and camera parameters that, when projected to 2D, minimise the sum of squared distances over all landmarks. We seek to minimise the objective function $\varepsilon_{\text{persp}}(\mathbf{r}, \mathbf{t}_{3\text{d}}, f, \boldsymbol{\alpha})$ under perspective projection. Solving this optimisation problem yields a least squares estimate of the pose and shape of a face, given 2D landmark positions. As mentioned, we introduced objective functions for the orthographic and perspective cases and showed how they can be expressed as separable nonlinear least squares problems in Chapter 3.

In Section 5.5, we show that for both orthographic and perspective cases, with pose fixed there remain degrees of flexibility that allow the 3D shape to vary without significantly increasing the objective value. However, for the perspective case there is an additional degree of freedom related to the subject-camera distance, i.e. t_z. If, instead of allowing t_z to be optimised along with other parameters, we fix it to some chosen value k, then we can obtain different shape and pose parameters:

$$\boldsymbol{\alpha}^*(k) = \arg_{\boldsymbol{\alpha}} \min_{\mathbf{r}, \mathbf{t}_{3\text{d}}, f, \boldsymbol{\alpha}} \varepsilon_{\text{persp}}(\mathbf{r}, \mathbf{t}_{3\text{d}}, f, \boldsymbol{\alpha}), \quad \text{s.t.} \quad t_z = k. \tag{5.1}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^D$ is a vector contains the number of D shape parameters which determines the vertex positions of a 3DMM by:

$$\mathbf{s}(\boldsymbol{\alpha}) = \mathbf{Q}\boldsymbol{\alpha} + \bar{\mathbf{s}}, \tag{5.2}$$

where the vector $\mathbf{s}(\boldsymbol{\alpha}) \in \mathbb{R}^{3N}$ contains the coordinates of the N vertices, stacked to form a long vector: $\mathbf{s} = [u_1, v_1, w_1, \dots, u_N, v_N, w_N]^{\text{T}}$, $\mathbf{Q} \in \mathbb{R}^{3N \times D}$ contains the D retained principal components and $\bar{\mathbf{s}} \in \mathbb{R}^{3N}$ is the mean shape.

Given 2D landmark observations, we therefore have a continuous (nonlinear) space of solutions $\boldsymbol{\alpha}^*(k)$ as a function of subject-camera distance. This is the perspective face shape ambiguity. If the mean reprojection error with a value of k other than the optimal one is still smaller than the tolerance of our landmark detector, then shape recovery is ambiguous.

5.5 Flexibility Modes

We now show that there are remaining modes of flexibility in the model fit. Our assumption is that a least squares model fit has been obtained which amounts to a shape, $\mathbf{Q}\boldsymbol{\alpha}+\bar{\mathbf{s}}$, determined by the estimated shape parameter and a pose $(\mathbf{r}, s, \mathbf{t}_{2\mathrm{d}})$ or $(\mathbf{r}, f, \mathbf{t}_{3\mathrm{d}})$ for orthographic or perspective, respectively. Keeping pose parameters fixed, we wish to find perturbations to the shape parameters that change the projected 2D geometry as little as possible (i.e. minimising the increase in the reprojection error of landmark vertices) while changing the 3D shape as much as possible.

Our approach to computing these flexibility modes is an extension of the method of [Albrecht et al., 2008]. They considered the problem of flexibility only in a 3D setting where the model is partitioned into a disjoint fixed part and a flexible part. We extend this so that the constraint on the fixed part acts in 2D after orthographic or perspective projection while the flexible part is the 3D shape of the whole face.

In the orthographic case, we define the 2D projection of the principal component directions for the L landmark vertices as:

$$\boldsymbol{\Pi}_{\mathrm{ortho}} = \left(\mathbf{I}_L \otimes \left(\mathbf{P}\mathbf{R}(\mathbf{r})\right)\right)\mathbf{Q}_L, \tag{5.3}$$

where $\mathbf{r}$ is the rotation vector that was estimated during fitting. Intuitively, we seek modes that move the landmark vertices primarily along the projection axis, which depends only on the rotation, and therefore do not move their 2D projection much. Hence, the flexibility modes do not depend on the scale or translation of the fit or even the landmark positions. For the perspective case, we again use the DLT linearisation in (3.34), leading to the following expression:

$$\boldsymbol{\Pi}_{\mathrm{persp}} = \mathbf{D}\left(\mathbf{I}_L \otimes \left(\mathbf{K}(f)\begin{bmatrix}\mathbf{R}(\mathbf{r}) & \mathbf{t}_{3\mathrm{d}}\end{bmatrix}\mathbf{O}\right)\right)\mathbf{Q}_L, \tag{5.4}$$

where

$$\mathbf{O} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}. \tag{5.5}$$

Again, $\mathbf{r}$, f and $\mathbf{t}_{\mathrm{3d}}$ are the rotation vector, focal length and translation that were estimated during fitting. By using the DLT linearisation, the intuition here is that we want the camera rays to the landmark vertices to remain as parallel as possible with the homogeneous vectors representing the observed landmarks.

Concretely, we seek flexibility modes, $\mathbf{f} \in \mathbb{R}^D$, such that $\mathbf{Qf}$ changes as much as possible whilst the 2D projection of the landmarks, given by $\mathbf{\Pi}_{\mathrm{ortho}}\mathbf{f}$ or $\mathbf{\Pi}_{\mathrm{persp}}\mathbf{f}$, changes as little as possible. This can be formulated as a constrained maximisation problem:

$$\max_{\mathbf{f} \in \mathbb{R}^D} \|\mathbf{Qf}\|^2 \quad \text{subject to } \|\mathbf{\Pi f}\|^2 = c, \tag{5.6}$$

where $\mathbf{\Pi}$ is one of the projection matrices and $c \in \mathbb{R}^+$ controls how much variation in the 2D projection is allowed (this value is arbitrary since it does not appear in the subsequent flexibility mode computation). Introducing a Lagrange multiplier and differentiating with respect to $\mathbf{f}$ yields:

$$\mathbf{Q}^{\mathrm{T}}\mathbf{Qf} = \lambda \mathbf{\Pi}^{\mathrm{T}}\mathbf{\Pi f}. \tag{5.7}$$

This is a generalised eigenvalue problem whose solution is a set of flexibility modes $\mathbf{f}_1, \dots, \mathbf{f}_D$ along with their corresponding generalised eigenvalue $\lambda_1, \dots, \lambda_D$, sorted in descending order. Therefore, $\mathbf{f}_1$ is the flexibility mode that changes the 3D shape as much as possible while minimising the change to the projected 2D geometry. If a face was fitted with shape parameters $\boldsymbol{\alpha}$ then its shape is varied by adjusting the weight w in: $\mathbf{Q}(\boldsymbol{\alpha} + w\mathbf{f}) + \bar{\mathbf{s}}$.

We can truncate the number of flexibility modes by setting a threshold k_1 on the mean Euclidean distance by which the surface should change and testing whether the corresponding change in mean landmark error is less than a threshold k_2. We retain only those flexibility modes where this is the case.

5.6 Experimental Results

We now present experimental results to demonstrate the ambiguities that arise in estimating 3D face shape from 2D geometry. We use the shape component of the Basel Face Model [Paysan et al., 2009] only. The model is supplied with 10 out-of-sample faces which are scans of real faces that are in correspondence with the model. We use these for quantitative evaluation on synthetic data. Unusually, the model does not factor out scale, i.e. faces are only aligned via translation and rotation. This means that the vertex positions are in absolute units of distance. This allows us to specify subject-camera distance in physically meaningful units. For all fittings we use Tikhonov regularisation with a low weight. For sparse (landmark) fitting, where overfitting is more likely, we use $D = 70$ dimensions and constrain parameters to be within $k = 2$ standard deviations of the mean. For dense fitting, we use all $D = 199$ model dimensions and constrain parameters to be $k = 3$ standard deviations of the mean.

We make use of two quantitative error measures in our evaluation. d_S is the mean Euclidean distance between the ground truth and reconstructed surface after aligning with Procrustes analysis. d_L is the mean distance between observed landmarks and the corresponding projection of the reconstructed landmark vertices, expressed as a percentage of the interocular distance.

5.6.1 Perspective Ambiguity

We begin by investigating the perspective ambiguity using synthetic data. We use the out-of-sample BFM scans to create input data by choosing pose parameters and projecting the faces to 2D. For sparse landmarks, we use the 70 anthropometric landmarks (due to [Farkas, 1994]) whose indices in the BFM are known. These landmarks are particularly appropriate as they were chosen so as to best measure the variability in craniofacial shape over a population. In Figure 5.2a, we show over what range of distances perspective transformation has a significant effect on 2D face geometry. For each face, we project the 70 landmarks to 2D under perspective projection and measure d_L with respect to the orthographic projection of the landmarks. As t_z increases, the projection converges towards orthography and the error

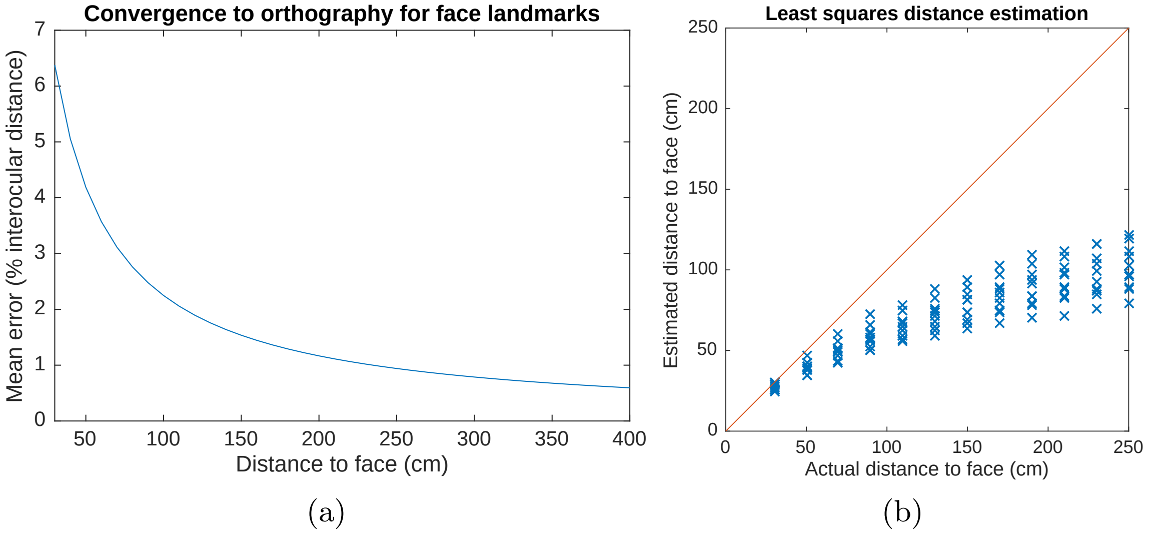


Figure 5.2: (a) Mean landmark error (y axis) between perspective and orthographic projection, averaged over 10 BFM scans, as subject-camera distance (x axis) is varied. (b) Subject-camera distance estimation by least squares optimisation.

tends to zero. The landmark error falls below 1% when the distance is around 2.5 metres. Hence, we experiment with distances ranging from selfie distance (30cm) up to this distance.

Our first evaluation of the perspective ambiguity is based on estimating the subject-camera distance as one of the parameters in the least squares fitting process. We use the BFM scans as target faces, vary the subject-camera distance and project the 70 Farkas landmarks to 2D under perspective projection. We use a frontal pose ($\mathbf{r} = [0\ 0\ 0]$) and arbitrarily set the focal length to $f = 1$. We initialise the optimisation with the correct focal length and rotation, giving it the best possible chance of estimating the correct distance. We plot estimated versus ground truth distance in Figure 5.2b. Optimal performance would see all points falling on the diagonal red line. The distance is consistently under-estimated and the mean percentage error in the estimate is 42%. It is clear that the 2D landmarks alone do not contain enough information to accurately estimate subject-camera distance as part of the model fitting process.

Our second experiment is that landmarks produced by a real 3D face shape at one distance can be explained by 3D shapes at multiple different distances. We show quantitative results in Table 5.1. Each row of the table corresponds to a distance at which we place each of the BFM scans in a frontal pose before projecting to 2D. We then fit to these landmarks with

Actual distance (cm)		Fitting distance (cm)				
		30	**60**	**120**	**240**	**Ortho**
30	d_L	0.21	0.24	0.26	0.27	0.28
	d_S	7.23	9.70	13.07	14.55	14.47
60	d_L	0.30	0.26	0.27	0.27	0.28
	d_S	8.07	6.29	6.60	6.99	7.48
120	d_L	0.37	0.29	0.28	0.28	0.28
	d_S	9.52	6.17	5.38	5.39	5.62
240	d_L	0.42	0.32	0.29	0.29	0.28
	d_S	10.16	6.72	5.59	5.37	5.38
Ortho	d_L	0.47	0.35	0.31	0.30	0.29
	d_S	11.02	7.43	6.01	5.54	5.29

Table 5.1: Quantitative results for the perspective ambiguity on synthetic data. Each cell shows the landmark error, d_L in %, top and surface error, d_S in mm, bottom.

the subject-camera distance assumed to be the value shown in the column. The results show that we are able to explain the data almost as well at the wrong distance as the correct one but the 3D shape is very different, differing by over a 1cm on average. Note that [Burgos-Artizzu et al., 2014] found that the difference between landmarks on the same face placed by two different humans was typically 3% of the interocular distance. Similarly, the 300 faces in-the-wild challenge [Sagonas et al., 2016] found that even the best methods did not obtain better than 5% accuracy for more than 50% of the landmarks. Hence, the difference between target and fitted landmarks is substantially smaller than the accuracy of either human or machine placed landmarks. Importantly, this means that the fitting energy could not be used to resolve the ambiguity. The residual difference between target and fitted landmarks is too small to meaningfully choose between the two solutions.

We now show qualitative examples from the same experiment. In Figure 5.3, we show orthographic renderings of perspective fits to the face shown in the first column. In the top row the target landmarks were generated by viewing the face at 30cm, in the bottom row the face was at 120cm. In each column we show fitting results at different distances. In the final column we show the landmarks of the real face (circles) overlaid with the landmarks from the fitted faces (dots) showing that highly varying 3D faces can produce almost identical 2D landmarks.

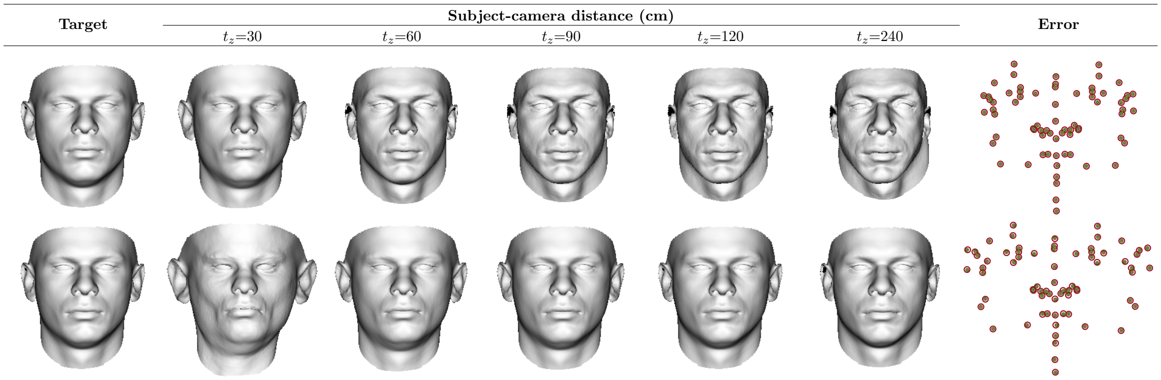


Figure 5.3: Qualitative perspective face shape ambiguity. There is a subspace of possible 3D face shapes with varying subject-camera distance within the landmark tolerance. Target face is at 30cm (top row) and 120cm (bottom row).

In Figures 5.4 and 5.5, we go further by showing the results of fitting to sparse 2D landmarks (the Farkas feature points), landmarks/edges and all vertices for 4 of the BFM scans (i.e. the targets are real faces). In Figure 5.4, the target face is close to the camera ($t_z = 30$cm) and we fit the model at a far distance ($t_z = 120$cm). This configuration is reversed in Figure 5.5 (200cm to 60cm). Since we are only interested in the spatial configuration of features in the image, we show both target and fitted mesh with the texture of the real target face. The target perspective projection to which we fit is shown in the first and fifth columns. The fitting result under perspective projection is shown in the second to fourth columns and sixth to eight columns. To enable comparison between the target and fitted faces, we render them under orthographic projection in rows two and four respectively. The landmarks from the target (plotted as blue circles) and fitted (shown as red dots) face are shown under perspective projection in column nine. We illustrate edge correspondence (model contours) between faces in the tenth column. In the last column, we average the target and fitted face texture from the dense fitting result, showing that there is no visible difference in the 2D geometry of these two images.

The implication of these results is that, in a sample of real faces, we might expect that two different identities with different face shapes could give rise to approximately the same 2D landmarks when viewed from different distances. We show in Figure 5.6 that this is indeed the case. The CMDP dataset contains images of 53 subjects viewed at 7 different distances. 55 landmarks are placed manually on each face image. We search for pairs of faces whose

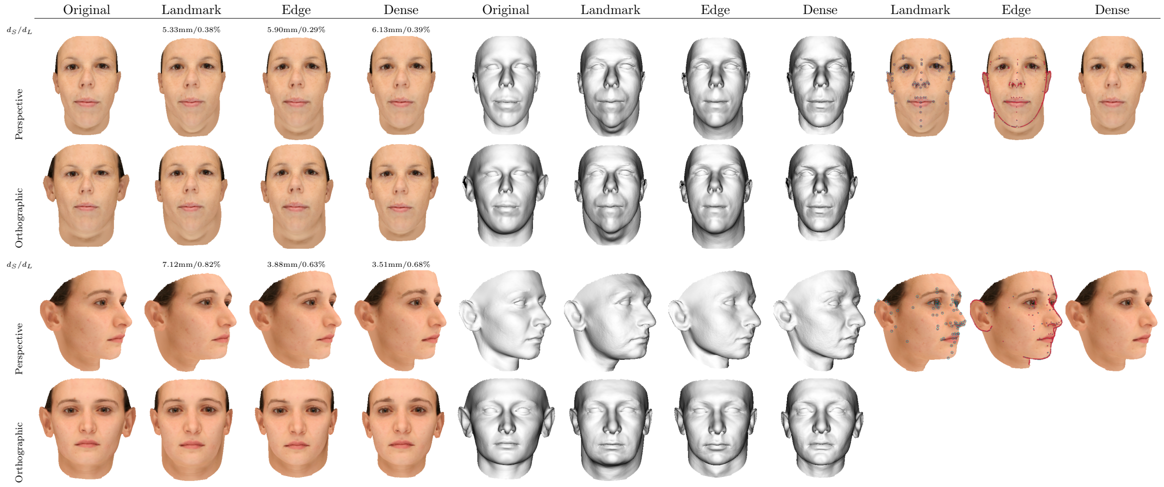


Figure 5.4: Sparse and dense fitting of the synthetic images. Target at 30cm, fitted results at 120cm.

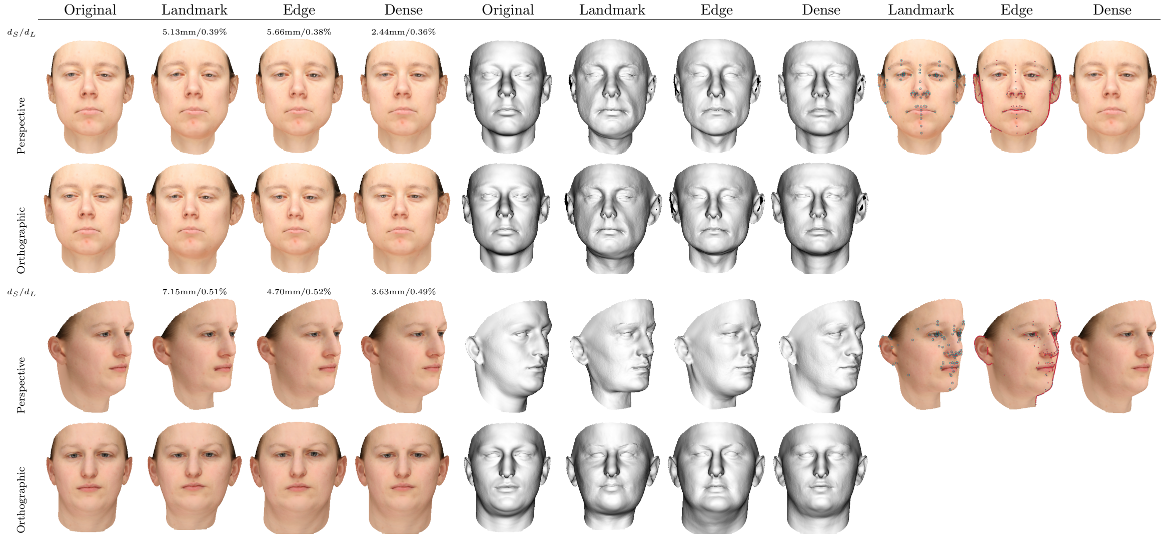


Figure 5.5: Sparse and dense fitting of the synthetic images. Target at 200cm, fitted results at 60cm.

landmarks (when viewed at different distances) are close in a Procrustes sense. Despite the small sample size, we find a pair of faces whose mean landmark error is 2.48% (i.e. they are within the expected accuracy of a landmark detector [Sagonas et al., 2016]) when they are viewed at 61cm and 488cm respectively (second and fourth image in the figure). In the third image, we blend these two images to show that their 2D features indeed align well. To highlight that their face shape is in fact quite different, we show their appearance with distances reversed in columns one and five (allowing direct comparison between columns one

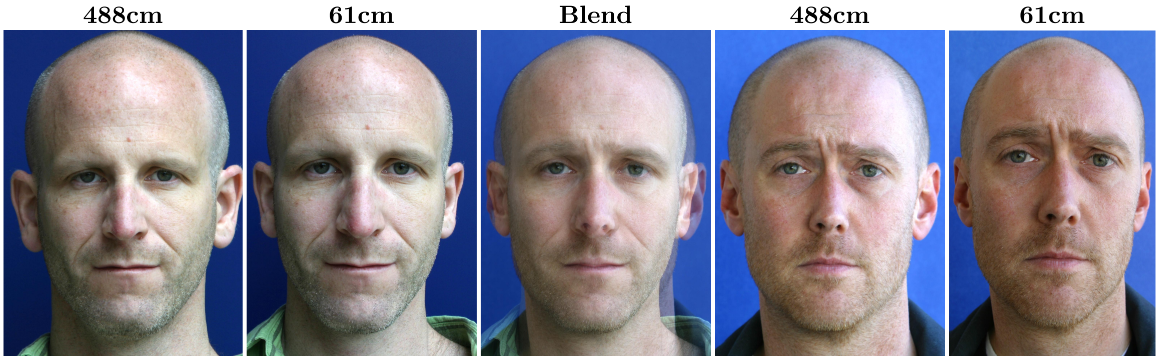


Figure 5.6: Perspective ambiguity in real faces. Two faces are shown at two different distances. The blend in the middle shows that their 2D geometry is similar when viewed at very different distances.

and four or two and five). For example, the face in column one has larger ears and inner features that are more concentrated towards to the centre of the face compared to the face in column four.

The CMDP data can also be used to demonstrate a surprising conclusion. For all 53 subjects, we compute the mean landmark error between the same identity at 61cm and 488cm which is 3.11%. Next, for each identity we find the identity at the same distance with the smallest landmark error. Averaged over all identities, this gives a value of 2.86% for 61cm and 2.83% for 488cm. We therefore conclude that 2D geometry between different identities at the same distance is more similar than between the same identity at different distances. If the number of identities was increased, the size of this effect would likely increase since the chance of finding closely matching different identity pairs would increase.

This demonstrate clearly that two faces with significantly different 3D shape can give rise to almost identical 2D landmark positions under perspective projection.

5.6.2 Flexibility Modes

We now explore the flexibility that remains when a model has been fitted to 2D geometric information. There is a surprising amount of remaining flexibility. Using the 70 Farkas landmark points under orthographic projection in a frontal pose, the BFM has around 50 flexibility modes that change the 3D shape by $k_1 = 2$mm while inducing a mean change in landmark position of less than $k_2 = 2$ pixels. Restricting consideration to those flexibility

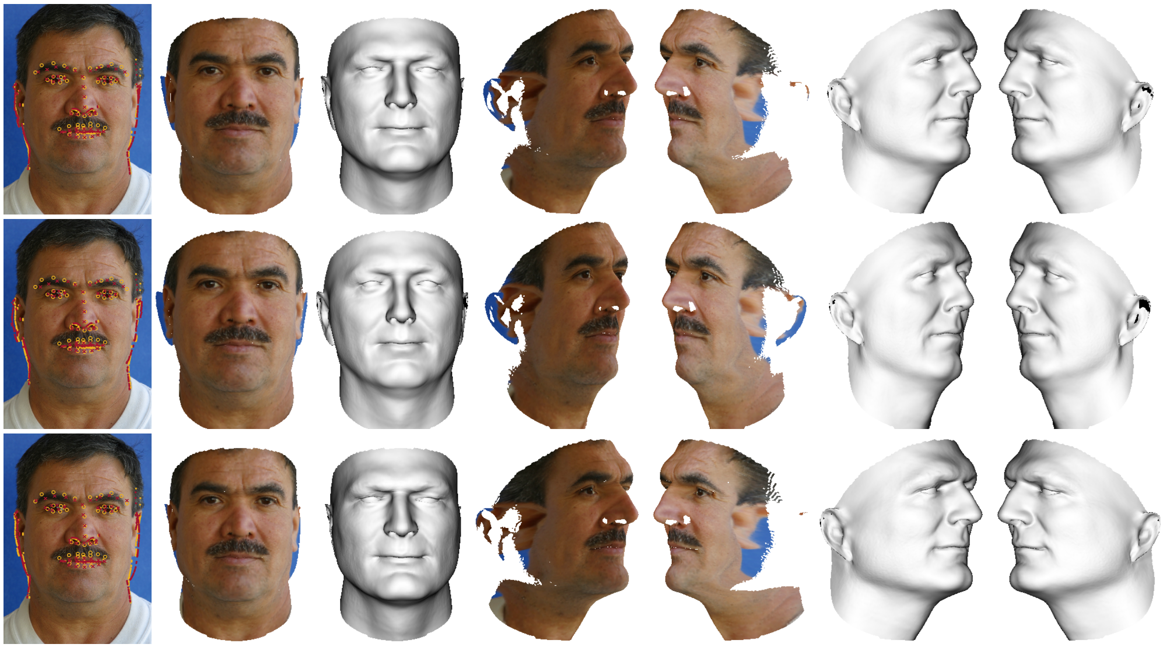


Figure 5.7: Orthographic fitting with flexibility modes. Landmark and edge fitting (first row). The first plus and minus flexibility components (second and third rows). Landmark distance is 1.14% and surface distance is 10mm.

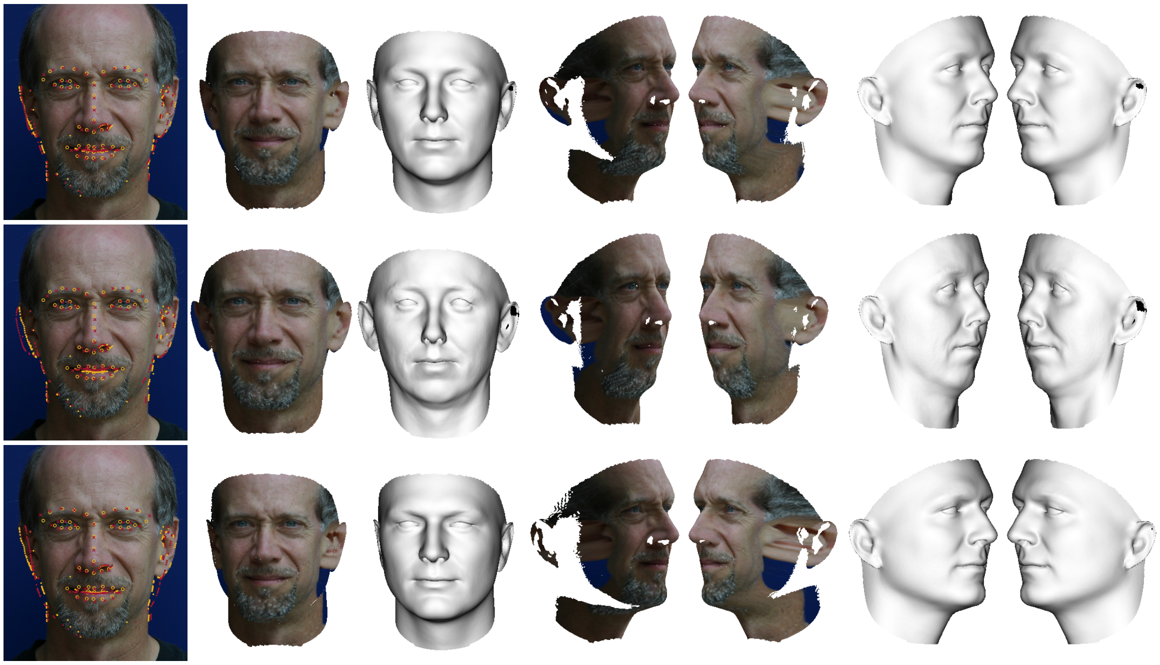


Figure 5.8: Perspective fitting with flexibility modes. Landmark and edge fitting (first row). The first plus and minus flexibility components (second and third rows). Landmark distance is 1.79% and surface distance is 10mm.

modes where the shape parameter vector remains plausible (i.e. stays within 3 standard deviations of the expected Mahalanobis length [Patel and Smith, 2016]), the number reduces to 7. This still means that knowing the exact 2D location of 70 landmark points only reduces the space of possible 3D face shapes to a 7D subspace of the morphable model.

In Figures 5.7 and 5.8, we show qualitative examples of the flexibility modes. We fit to a real image under both orthographic and perspective projection. We then compute the first flexibility mode and vary the shape in both directions such that the mean surface distance is 10mm. Despite the large change in the surface, the landmarks only vary by 1.14% for orthographic and 1.79% for perspective fitting. The correspondence when the texture is sampled onto the mesh remains similar. In other words, three very different surfaces provide plausible 3D explanations of the 2D data.

5.7 Conclusions

In this chapter, we have studied ambiguities that arise when 3D face shape is estimated from monocular 2D geometric information. We have shown that 2D geometry (either sparse landmarks, semi-dense contours or dense vertex information) can be explained by a space of possible faces which vary significantly in 3D shape.

We consider it surprising that the natural variability in face shape should include variations consistent with perspective transformation and that there are degrees of flexibility in face shape that have only a small effect on 2D geometry when pose is fixed. There are a number of interesting implications of these ambiguities.

In forensic image analysis, metric distances between features have been used as a way of comparing the identity of two face photographs. For example, [Porter and Doran, 2000] normalise face images by the interocular distance before using measurements such as the width of the face, nose and mouth to compare identities. We have shown that, after such normalisation, all distances between anthropometric features can be equal (up to the accuracy of landmarking) for two very different faces. This casts doubt on the use of such techniques in forensic image analysis and perhaps partially explains the studies that have demonstrated the weakness of these approaches [Kleinberg et al., 2007].

Clearly, any attempt to reconstruct 3D face shape using 2D geometric information alone (such as in [Blanz et al., 2004,Knothe et al., 2006,Patel and Smith, 2009,Aldrian and Smith, 2013]) will be subject to the ambiguity. Hence, the range of possible solutions is large and the likely accuracy low. If estimated 3D face shape is to be used for recognition, then the dissimilarity measure must account for the ambiguities we have described.

For some face analysis problems, the purpose of fitting a statistical shape model is simply to establish correspondence. For example, it may be that face texture will be processed on the surface of the mesh, or that correspondence is required in order to compare different face textures for recognition. In such cases, these ambiguities are not important. Any solution that fits the dense 2D shape features (i.e. any from within the space of solutions described by the ambiguity) will suffice to correctly establish correspondence.

Chapter 6

Conclusions

In this thesis, we addressed the question of to what extent we could use geometric information for face shape recovery. We explored the correspondence problem focusing on purely geometric transformations in the optimisation and deep learning domain. The contributions made in this work are summarised as follows.

In Chapter 3, we presented a fully automatic method for fitting a 3D morphable model to single face images in arbitrary pose and lighting. Our approach relies on geometric features (edges and landmarks) and, inspired by the iterated closest point algorithm, is based on computing hard correspondences between model vertices and edge pixels. We demonstrated that this is superior to previous work that uses soft edge correspondences to form an edge-derived cost surface that is minimised by nonlinear optimisation.

Next, we upgraded our alternating linear least squares optimisation to separable nonlinear least squares which allow us to solve the least squares problem more efficiently. We show that the model fitting problem can be posed as a separable nonlinear least squares form and solved efficiently under both orthographic and perspective projection.

In Chapter 4, we proposed to use a 3D face model together with the spatial transformer, which explicitly incorporates model knowledge into the deep learning network. The architecture is based on a purely geometric approach in which only the shape component of the model is used. Our method can be trained in an unsupervised fashion and does not depend on synthetic training data or previous fitting results. We demonstrated the approach in the task of face pose estimation, frontalisation and as a potential input for face recognition.

In the second part of Chapter 4, we further investigated the idea of incorporating a 2D statistical shape model (that is itself learnt) in a spatial transformer network with no direct supervision. Introducing generic loss functions, statistical shape and appearance model are learnt by our network as well as a notion of correspondence that is optimal for the task being solved (in this case, fitting the model to an image).

Finally, in Chapter 5, we showed that 2D geometric information (landmarks and occluding contours) is an ambiguous cue for estimating 3D face shape if no further constraints are enforced. This means that one can recover subspace of possible 3D face shapes that are within the tolerance with which we can hope to locate these features in a 2D image. Moreover, the 2D shape of a human face that is viewed under perspective projection varies in relation to the distance between the camera and subject. This arises from the effect of perspective transformation which distorts the relative distances between facial features.

6.1 Future Work

The contributions of this thesis could be strengthened and developed further as follows:

- We could incorporate any of the refinements of standard ICP to our model fitting approach. A comprehensive study can be found in [Rusinkiewicz and Levoy, 2001]. We could also explore other ways in which the notion of correspondence is formulated.

- The fitting accuracy is directly related to the accuracy of landmark points. In this thesis, we used landmarks labelled by the landmark detection algorithm in [Zhu and Ramanan, 2012] or provided by datasets. We can further examine the accuracy of several other publicly available landmark detectors [Uřičář et al., 2012, Kazemi and Josephine, 2014, Asthana et al., 2015]. It is also important to investigate the effect and quality (as well as visibility) of each individual landmark and modify its contributions to model fitting accordingly.

- Similar to landmarks, detection of image edges is quite fragile and it is often hard to extract reliable information, depending on noise and cluttered background. There is also ambiguity in interpreting, which could stem from many factors such as shadow, reflectance or simply because of visibility. Currently, we compute edges by applying the Canny edge detector with a fixed threshold. We can extend this to the machine learning domain [Dollár and Zitnick, 2015] or to deep learning architecture [Bertasius et al., 2015]. A very possible future direction is to substitute an edge detector with superpixels [Ren and Malik, 2003], similar to [Kae et al., 2013]. This transforms the problem of edge detection into a combinatorial problem of labelling superpixels as face/non-face and also provides information about the location of edges using global interpretation of image rather than local, pixel-level information.

- We can benefit from using multiple images or videos by exploiting the consistency within frames and multiple observations from different angles. Moreover, we can have several assumptions (e.g. fixed identity, fixed scene lighting, similar background) that would allow us to constrain the model fitting problem.

- Future research might investigate a detailed face model which includes all facial characteristics (e.g. dimples, wrinkles, crow feet) that could generate more accurate face shape. This can be achieved by further exploration on modelling these deformations or applying shape-from-shading techniques.

- In some cases, an ambiguous face shape is acceptable to some extent. For instance, if the goal is to establish robust face recognition, estimating precise face shape is not always necessary. Recently, [Hassner et al., 2015] showed this by employing a single 3D shape for face frontalisation. Sometimes, rough approximation of the 3D surface would be sufficient for face recognition if there were elements other than geometry to exploit. [Pierrard and Vetter, 2007] achieved accurate face identification by exploiting local skin irregularities (e.g. moles, birthmarks). This means that combining geometric features with texture, shading and colour cues could lead to better face analysis. This also means that the geometry derived from appearance and photometry could potentially provide additional cues that could be integrated for improved correspondence. Of course, both approaches come with certain drawbacks of complex nonlinear optimisation.

- The most obvious way to use our transformer networks is as part of a network that is trained to solve a task such as recognition or classification in an end-to-end manner. We would hope that the normalisation of pose and shape effects means that a recognition network could be trained on less data and with less complexity than existing networks that must learn pose invariance implicitly. The shape parameters estimated by the localiser may contain discriminative information and these could be combined into subsequent descriptors for recognition. Further exploration of the multiview fitting loss on a multiview face database would provide a rich source of data for learning accurate localisation.

- Recently, generative adversarial networks [Goodfellow et al., 2014] have shown great promise in training generative models to be indistinguishable from real data [Makhzani et al., 2016]. Our work includes fixed, prebuilt generative models (i.e. a 3DMM) as well as learnt generative models (the shape and appearance models in the StaTN). This means that we can exploit an additional cost function based on an adversarial loss that minimises the difference between real and generated samples. This could help our methods learn better generative models or to fit fixed models better (by backpropagating adversarial losses through the generative model and into the discriminative model fitting network). Moreover, our proposed networks can be improved further by learning a generative distribution of faces in an unsupervised scenario.

- The ambiguity work in this thesis can be extended beyond geometric cues. For example, it would be promising to investigate whether additional cues resolve the ambiguities. An interesting follow-up to the work of [Amberg et al., 2007] would be to examine whether there is an ambiguity in uncalibrated stereo face images. Alternatively, we could investigate whether photometric cues (shading, shadowing and specularities) or statistical texture cues help to resolve the ambiguity. In the case of shading, it is not clear that this will be the case. Assuming illumination is unknown, it is possible that a transformation of the lighting environment could lead to shading which is consistent with (or at least close to) that of the target face [Smith, 2016].

6.2 Concluding Remarks

On the one hand, this thesis adds to the prevailing belief that deep learning has a remarkable ability to solve challenging tasks on completely uncontrolled data. So, a single forward pass of an in-the-wild image through our trained CNN outputs an accurately fitted model. However, this thesis also argues that we should be cautious. The ambiguities described in Chapter 5 are significant and presumably networks in previous work, such as [Tewari et al., 2017], have learnt some kind of prior to choose from the space of ambiguous solutions. Once the network is trained, we have no way of knowing what this prior is or understanding the space of possible solutions. This should concern us if we were to use such networks in a security setting or even in a safety-critical setting.

Although the current trend in deep learning is to make use of every single cue for better recognition and segmentation systems, fundamental issues need to be taken into account. For example, information extracted from the background, clothing and hair may at times be fruitful for face recognition. However, one should consider the fact that these features can be easily manipulated, imitated or reproduced. Moreover, these networks can be dramatically fooled by making small changes in the test images [Goodfellow et al., 2015] or generating fooling examples [Nguyen et al., 2015].

One seemingly unattractive yet potentially visionary idea would be to start over, modelling human perception and understanding from scratch. The main focus should fall on the capacity of extracting meaningful information as the human brain does rather than on pixel-based processing. This would likely cause a huge performance hit if our approach to comparative benchmarking is to achieve the highest accuracy at any cost (even if the cues we exploit are unreliable or false, in general). In the long run though, this would lead to more generic and transparent algorithms that are fully understood and practical to use in everyday systems. Perhaps this is the way to prevent the next AI winter.

Appendix A

SNLS Derivatives

Here we provide all of the derivatives required to optimise the SNLS objective functions. Specifically, we show how to compute the Jacobian matrices of the residual functions for the orthographic (3.26) and perspective case linearised via the DLT (3.35) from Chapter 3.

Matrix Derivative Identities: The following identities are used in our derivations.

The derivatives of the axis-angle to rotation matrix function in (3.7) are given by [Gallego and Yezzi, 2015]:

$$\frac{\partial \mathbf{R}}{\partial r_i} = \begin{cases} [\mathbf{e}_i]_\times & \text{if } \mathbf{r} = \mathbf{0} \\ \frac{r_i[\mathbf{r}]_\times + [\mathbf{r} \times (\mathbf{I} - \mathbf{R}(\mathbf{r}))\mathbf{e}_i]_\times}{\|\mathbf{r}\|^2} \mathbf{R}(\mathbf{r}) & \text{otherwise} \end{cases}, \tag{A.1}$$

where $\mathbf{e}_i$ is the ith vector of the standard basis in $\mathbb{R}^3$.

The scalar derivative of the Kronecker product is:

$$\frac{\partial (\mathbf{X} \otimes \mathbf{Y})}{\partial x} = \frac{\partial \mathbf{X}}{\partial x} \otimes \mathbf{Y} + \mathbf{X} \otimes \frac{\partial \mathbf{Y}}{\partial x}. \tag{A.2}$$

For the special case involving the identity matrix, i.e. where $\mathbf{X} = \mathbf{I}$, this simplifies to:

$$\frac{\partial (\mathbf{I} \otimes \mathbf{Y})}{\partial x} = \mathbf{I} \otimes \frac{\partial \mathbf{Y}}{\partial x}. \tag{A.3}$$

The scalar derivative of the pseudoinverse $\mathbf{A}^{+}(x)$ of $\mathbf{A}$ at x is given by:

$$\frac{\partial \mathbf{A}^{+}}{\partial x} \;=\; -\mathbf{A}^{+}\frac{\partial \mathbf{A}}{\partial x}\mathbf{A}^{+} \;+\; \mathbf{A}^{+}\mathbf{A}^{+\mathrm{T}}\frac{\partial \mathbf{A}^{\mathrm{T}}}{\partial x}(\mathbf{I} \;-\; \mathbf{A}\mathbf{A}^{+}) \;+\; (\mathbf{I} \;-\; \mathbf{A}^{+}\mathbf{A})\frac{\partial \mathbf{A}^{\mathrm{T}}}{\partial x}\mathbf{A}^{+\mathrm{T}}\mathbf{A}^{+}. \quad \text{(A.4)}$$

Orthographic Case: The derivatives of the matrix $\mathbf{A}(\mathbf{r}, s)$ are given by:

$$\frac{\partial \mathbf{A}}{\partial s} = \left[(\mathbf{I}_L \otimes \mathbf{P}\mathbf{R}(\mathbf{r}))\,\mathbf{Q}_L \quad \mathbf{1}_L \otimes \mathbf{I}_2\right], \quad \text{(A.5)}$$

$$\frac{\partial \mathbf{A}}{\partial r_i} = \left[s\left(\mathbf{I}_L \otimes \mathbf{P}\tfrac{\partial \mathbf{R}}{\partial r_i}\right)\mathbf{Q}_L \quad \mathbf{0}_{2L\times 2}\right]. \quad \text{(A.6)}$$

The derivatives of the vector $\mathbf{y}(\mathbf{r}, s)$ are given by:

$$\frac{\partial \mathbf{y}}{\partial s} = (\mathbf{I}_L \otimes \mathbf{P}\mathbf{R}(\mathbf{r}))\,\bar{\mathbf{s}}, \quad \text{(A.7)}$$

$$\frac{\partial \mathbf{y}}{\partial r_i} = s\left[\left(\mathbf{I}_L \otimes \mathbf{P}\frac{\partial \mathbf{R}}{\partial r_i}\right)\bar{\mathbf{s}}\right]. \quad \text{(A.8)}$$

From the components above we can compute the derivatives of the residual function:

$$\frac{\partial \mathbf{d}_{\mathrm{ortho}}}{\partial s} \;=\; \left(\mathbf{A}(\mathbf{r}, s)\frac{\partial \mathbf{A}^{+}}{\partial s} + \frac{\partial \mathbf{A}}{\partial s}\mathbf{A}^{+}(\mathbf{r}, s)\right)\mathbf{y}(\mathbf{r}, s) \;+\; \mathbf{A}(\mathbf{r}, s)\mathbf{A}^{+}(\mathbf{r}, s)\frac{\partial \mathbf{y}}{\partial s} \;-\; \frac{\partial \mathbf{y}}{\partial s}, \quad \text{(A.9)}$$

$$\frac{\partial \mathbf{d}_{\mathrm{ortho}}}{\partial r_i} \;=\; \left(\mathbf{A}(\mathbf{r}, s)\frac{\partial \mathbf{A}^{+}}{\partial r_i} + \frac{\partial \mathbf{A}}{\partial r_i}\mathbf{A}^{+}(\mathbf{r}, s)\right)\mathbf{y}(\mathbf{r}, s) \;+\; \mathbf{A}(\mathbf{r}, s)\mathbf{A}^{+}(\mathbf{r}, s)\frac{\partial \mathbf{y}}{\partial r_i} \;-\; \frac{\partial \mathbf{y}}{\partial r_i}. \quad \text{(A.10)}$$

Finally, the Jacobian, $\mathbf{J}_{\mathbf{d}_{\mathrm{ortho}}}(\mathbf{r}, s)$, is obtained by stacking these four vectors into a $2L \times 4$ matrix:

$$\mathbf{J}_{\mathbf{d}_{\mathrm{ortho}}}(\mathbf{r}, s) = \left[\tfrac{\partial \mathbf{d}_{\mathrm{ortho}}}{\partial r_1} \quad \tfrac{\partial \mathbf{d}_{\mathrm{ortho}}}{\partial r_2} \quad \tfrac{\partial \mathbf{d}_{\mathrm{ortho}}}{\partial r_3} \quad \tfrac{\partial \mathbf{d}_{\mathrm{ortho}}}{\partial s}\right]. \quad \text{(A.11)}$$

Perspective Case: The derivatives of the matrix $\mathbf{B}(\mathbf{r}, f)$ are given by:

$$\frac{\partial \mathbf{B}}{\partial f} = \mathbf{D}\frac{\partial \mathbf{E}}{\partial f}\mathbf{F}(\mathbf{r}) \quad \text{and} \quad \frac{\partial \mathbf{B}}{\partial r_i} = \mathbf{D}\mathbf{E}(f)\frac{\partial \mathbf{F}}{\partial r_i}, \tag{A.12}$$

where

$$\frac{\partial \mathbf{E}}{\partial f} = \mathbf{I}_L \otimes \frac{\partial \mathbf{K}}{\partial f} \quad \text{and} \quad \frac{\partial \mathbf{F}}{\partial r_i} = \left[\left(\mathbf{I}_L \otimes \tfrac{\partial \mathbf{R}}{\partial r_i}\right)\mathbf{Q}_L \quad \mathbf{0}_{3L\times 3}\right], \tag{A.13}$$

and

$$\frac{\partial \mathbf{K}}{\partial f} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{A.14}$$

The derivatives of the vector $\mathbf{z}(\mathbf{r}, f)$ are given by:

$$\frac{\partial \mathbf{z}}{\partial f} = -\mathbf{D}\left(\mathbf{I}_L \otimes \left[\frac{\partial \mathbf{K}}{\partial f}\mathbf{R}(\mathbf{r})\right]\right)\bar{\mathbf{s}}, \tag{A.15}$$

$$\frac{\partial \mathbf{z}}{\partial r_i} = -\mathbf{D}\left(\mathbf{I}_L \otimes \left[\mathbf{K}(f)\frac{\partial \mathbf{R}}{\partial r_i}\right]\right)\bar{\mathbf{s}}. \tag{A.16}$$

From the components above we can compute the derivatives of the residual function:

$$\frac{\partial \mathbf{d}^{\text{DLT}}_{\text{persp}}}{\partial f} = \left(\mathbf{B}(\mathbf{r}, f)\frac{\partial \mathbf{B}^+}{\partial f} + \frac{\partial \mathbf{B}}{\partial f}\mathbf{B}^+(\mathbf{r}, f)\right)\mathbf{z}(\mathbf{r}, f) + \mathbf{B}(\mathbf{r}, f)\mathbf{B}^+(\mathbf{r}, f)\frac{\partial \mathbf{z}}{\partial f} - \frac{\partial \mathbf{z}}{\partial f}, \tag{A.17}$$

$$\frac{\partial \mathbf{d}^{\text{DLT}}_{\text{persp}}}{\partial r_i} = \left(\mathbf{B}(\mathbf{r}, f)\frac{\partial \mathbf{B}^+}{\partial r_i} + \frac{\partial \mathbf{B}}{\partial r_i}\mathbf{B}^+(\mathbf{r}, f)\right)\mathbf{z}(\mathbf{r}, f) + \mathbf{B}(\mathbf{r}, f)\mathbf{B}^+(\mathbf{r}, f)\frac{\partial \mathbf{z}}{\partial r_i} - \frac{\partial \mathbf{z}}{\partial r_i}. \tag{A.18}$$

Finally, the Jacobian, $\mathbf{J}_{\mathbf{d}^{\text{DLT}}_{\text{persp}}}(\mathbf{r}, f)$, is obtained by stacking these four vectors into a $3L \times 4$ matrix:

$$\mathbf{J}_{\mathbf{d}^{\text{DLT}}_{\text{persp}}}(\mathbf{r}, f) = \begin{bmatrix} \frac{\partial \mathbf{d}^{\text{DLT}}_{\text{persp}}}{\partial r_1} & \frac{\partial \mathbf{d}^{\text{DLT}}_{\text{persp}}}{\partial r_2} & \frac{\partial \mathbf{d}^{\text{DLT}}_{\text{persp}}}{\partial r_3} & \frac{\partial \mathbf{d}^{\text{DLT}}_{\text{persp}}}{\partial f} \end{bmatrix}. \tag{A.19}$$

Abbreviations

PCA	Principal Component Analysis
ASM	Active Shape Model
AAM	Active Appearance Model
3DMM	3D Morphable Model
BFM	Basel Face Model
ICP	Iterative Closest Point
LM-ICP	Levenberg-Marquardt Iterative Closest Point
SOP	Scaled Orthographic Projection
POS	Pose from Orthography and Scaling
ICEF	Iterated Closest Edge Fitting
ALS	Alternating Linear Least Squares
SNLS	Separable Nonlinear Least Squares
DLT	Direct Linear Transformation
CNN	Convolutional Neural Network
SGD	Stochastic Gradient Descent
STN	Spatial Transformer Networks
3DMM-STN	3D Morphable Models as Spatial Transformer Networks
StaTN	Statistical Transformer Networks
LFW	Labeled Faces in the Wild dataset
AFLW	Annotated Facial Landmarks in the Wild dataset
CMU PIE	The CMU Pose, Illumination, and Expression database
CMDP	Caltech Multi-Distance Portraits dataset

References

[Albrecht et al., 2008] Albrecht, T., Knothe, R., and Vetter, T. (2008). Modeling the remaining flexibility of partially fixed statistical shape models. In *Proc. MICCAI Workshops*, pages 160–169.

[Aldrian and Smith, 2013] Aldrian, O. and Smith, W. A. (2013). Inverse rendering of faces with a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(5):1080–1093.

[Amberg et al., 2007] Amberg, B., Blake, A., Fitzgibbon, A., Romdhani, S., and Vetter, T. (2007). Reconstructing high quality face-surfaces using model based stereo. In *Proc. ICCV*, pages 1–8.

[Amberg et al., 2008] Amberg, B., Knothe, R., and Vetter, T. (2008). Expression invariant 3d face recognition with a morphable model. In *Proc. FG*, pages 1–6.

[Amberg and Vetter, 2011] Amberg, B. and Vetter, T. (2011). Optimal landmark detection using shape models and branch and bound. In *Proc. ICCV*, pages 455–462.

[Asthana et al., 2015] Asthana, A., Zafeiriou, S., Tzimiropoulos, G., Cheng, S., and Pantic, M. (2015). From pixels to response maps: Discriminative image filtering for face alignment in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(6):1312–1320.

[Atkinson et al., 2009] Atkinson, G. A., Smith, M. L., Smith, L. N., and Farooq, A. R. (2009). Facial geometry estimation using photometric stereo and profile views. In *Proc. ICB*, pages 1–11.

[Baker and Matthews, 2001] Baker, S. and Matthews, I. (2001). Equivalence and efficiency of image alignment algorithms. In *Proc. CVPR*, pages 1–8.

[Baker and Matthews, 2004] Baker, S. and Matthews, I. (2004). Lucas-kanade 20 years on: A unifying framework. *Int. J. Comput. Vis.*, 56(3):221–255.

[Bansal et al., 2017] Bansal, A., Nanduri, A., Castillo, C. D., Ranjan, R., and Chellappa, R. (2017). Umdfaces: An annotated face dataset for training deep networks. In *Proc. IJCB*, pages 464–473.

[Belhumeur et al., 1997] Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):711–720.

[Belhumeur et al., 1999] Belhumeur, P. N., Kriegman, D. J., and Yuille, A. L. (1999). The bas-relief ambiguity. *Int. J. Comput. Vis.*, 35(1):33–44.

[Bertasius et al., 2015] Bertasius, G., Shi, J., and Torresani, L. (2015). Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *Proc. CVPR*, pages 4380–4389.

[Besl and McKay, 1992] Besl, P. and McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256.

[Bhagavatula et al., 2017] Bhagavatula, C., Zhu, C., Luu, K., and Savvides, M. (2017). Faster than real-time facial alignment: A 3d spatial transformer network approach in unconstrained poses. In *Proc. ICCV*, pages 3980–3989.

[Blanz et al., 2004] Blanz, V., Mehl, A., Vetter, T., and Seidel, H.-P. (2004). A statistical method for robust 3D surface reconstruction from sparse data. In *Proc. 3DPVT*, pages 293–300.

[Blanz and Vetter, 1999] Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH*, pages 187–194.

[Blanz and Vetter, 2003] Blanz, V. and Vetter, T. (2003). Face recognition based on fitting a 3D morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1063–1074.

[Booth et al., 2017] Booth, J., Antonakos, E., Ploumpis, S., Trigeorgis, G., Panagakis, Y., and Zafeiriou, S. (2017). 3D face morphable models “in-the-wild”. In *Proc. CVPR*, pages 48–57.

[Breuer et al., 2008] Breuer, P., Kim, K., Kienzle, W., Schölkopf, B., and Blanz, V. (2008). Automatic 3D face reconstruction from single images or video. In *Proc. FG*, pages 1–8.

[Brunton et al., 2014] Brunton, A., Salazar, A., Bolkart, T., and Wuhrer, S. (2014). Review of statistical shape spaces for 3D data with comparative analysis for human faces. *Comput. Vis. Image Underst.*, 128:1–17.

[Bryan et al., 2012] Bryan, R., Perona, P., and Adolphs, R. (2012). Perspective distortion from interpersonal distance is an implicit visual cue for social judgments of faces. *PloS one*, 7(9):e45301.

[Burgos-Artizzu et al., 2014] Burgos-Artizzu, X. P., Ronchi, M. R., and Perona, P. (2014). Distance estimation of an unknown person from a portrait. In *Proc. ECCV*, pages 313–327.

[Cao et al., 2015] Cao, C., Bradley, D., Zhou, K., and Beeler, T. (2015). Real-time high-fidelity facial performance capture. *ACM Trans. Graph.*, 34(4):46.

[Cao et al., 2014a] Cao, C., Hou, Q., and Zhou, K. (2014a). Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. Graph.*, 33(4):43.

[Cao et al., 2013] Cao, C., Weng, Y., Lin, S., and Zhou, K. (2013). 3d shape regression for real-time facial animation. *ACM Trans. Graph.*, 32(4):41.

[Cao et al., 2014b] Cao, C., Weng, Y., Zhou, S., Tong, Y., and Zhou, K. (2014b). Facewarehouse: A 3D facial expression database for visual computing. *IEEE Trans. Vis. Comput. Graphics*, 20(3):413–425.

[Cashman and Fitzgibbon, 2013] Cashman, T. J. and Fitzgibbon, A. W. (2013). What shape are dolphins? building 3D morphable models from 2D images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):232–244.

[Chen et al., 2016] Chen, D., Hua, G., Wen, F., and Sun, J. (2016). Supervised transformer network for efficient face detection. In *Proc. ECCV*, pages 122–138.

[Coleman and Li, 1996] Coleman, T. F. and Li, Y. (1996). An interior trust region approach for nonlinear minimization subject to bounds. *SIAM J. Optim.*, 6(2):418–445.

[Cootes et al., 1998] Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active appearance models. In *Proc. ECCV*, pages 484–498.

[Cootes et al., 2001] Cootes, T. F., Edwards, G. J., and Taylor, C. J. (2001). Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):681–685.

[Cootes and Taylor, 1992] Cootes, T. F. and Taylor, C. J. (1992). Active shape models-'smart snakes'. In *Proc. BMVC*, pages 266–275.

[Cootes et al., 1995] Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1995). Active shape models-their training and application. *Comput. Vis. Image Underst.*, 61(1):38–59.

[Cristinacce and Cootes, 2006] Cristinacce, D. and Cootes, T. F. (2006). Feature detection and tracking with constrained local models. In *Proc. BMVC*, pages 929–938.

[Dai et al., 2017] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. (2017). Deformable convolutional networks. In *Proc. ICCV*, pages 764–773.

[Davies et al., 2002] Davies, R. H., Twining, C. J., Cootes, T. F., Waterton, J. C., and Taylor, C. J. (2002). A minimum description length approach to statistical shape modeling. *IEEE Trans. Med. Imag.*, 21(5):525–537.

[Dementhon and Davis, 1995] Dementhon, D. F. and Davis, L. S. (1995). Model-based object pose in 25 lines of code. *Int. J. Comput. Vis.*, 15(1-2):123–141.

[Dollár and Zitnick, 2015] Dollár, P. and Zitnick, C. L. (2015). Fast edge detection using structured forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(8):1558–1570.

[Ecker et al., 2008] Ecker, A., Jepson, A. D., and Kutulakos, K. N. (2008). Semidefinite programming heuristics for surface reconstruction ambiguities. In *Proc. ECCV*, pages 127–140.

[Edwards et al., 1998] Edwards, G. J., Taylor, C. J., and Cootes, T. F. (1998). Interpreting face images using active appearance models. In *Proc. FG*, pages 300–305.

[Egger et al., 2016] Egger, B., Schneider, A., Blumer, C., Forster, A., Schönborn, S., and Vetter, T. (2016). Occlusion-aware 3d morphable face models. In *Proc. BMVC*, pages 64.1–64.11.

[Egger et al., 2018] Egger, B., Schönborn, S., Schneider, A., Kortylewski, A., Morel-Forster, A., Blumer, C., and Vetter, T. (2018). Occlusion-aware 3d morphable models and an illumination prior for face image analysis. *Int. J. Comput. Vis.*, pages 1–19.

[Farkas, 1994] Farkas, L. G. (1994). *Anthropometry of the head and face.* Raven Press.

[Fisher, 1936] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Ann. Eugen.*, 7(2):179–188.

[Fitzgibbon, 2003] Fitzgibbon, A. W. (2003). Robust registration of 2D and 3D point sets. *Image Vis. Comput.*, 21(13-14):1145–1153.

[Floater, 1997] Floater, M. S. (1997). Parametrization and smooth approximation of surface triangulations. *Comput. Aided Geom. Des.*, 14(3):231–250.

[Flores et al., 2013] Flores, A., Christiansen, E., Kriegman, D., and Belongie, S. (2013). Camera distance from face images. In *Proc. ISVC*, pages 513–522.

[Fried et al., 2016] Fried, O., Shechtman, E., Goldman, D. B., and Finkelstein, A. (2016). Perspective-aware manipulation of portrait photos. *ACM Trans. Graph.*, 35(4):128.

[Gallego and Yezzi, 2015] Gallego, G. and Yezzi, A. (2015). A compact formula for the derivative of a 3-D rotation in exponential coordinates. *J. Math. Imaging Vis.*, 51(3):378–384.

[Georghiades et al., 2001] Georghiades, A. S., Belhumeur, P. N., and Kriegman, D. J. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):643–660.

[Gerig et al., 2018] Gerig, T., Morel-Forster, A., Blumer, C., Egger, B., Luthi, M., Schönborn, S., and Vetter, T. (2018). Morphable face models - an open framework. In *Proc. FG*, pages 75–82.

[Gleicher, 1997] Gleicher, M. (1997). Projective registration with difference decomposition. In *Proc. CVPR*, pages 331–337.

[Golub and Pereyra, 2003] Golub, G. and Pereyra, V. (2003). Separable nonlinear least squares: the variable projection method and its applications. *Inverse Probl.*, 19(2):R1.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Proc. NIPS*, pages 2672–2680.

[Goodfellow et al., 2015] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *Proc. ICLR*, pages 1–11.

[Granger et al., 2001] Granger, S., Pennec, X., and Roche, A. (2001). Rigid point-surface registration using an em variant of icp for computer guided oral implantology. In *Proc. MICCAI*, pages 752–761.

[Grant et al., 2006] Grant, M., Boyd, S., and Ye, Y. (2006). Disciplined convex programming. In *Global optimization*, pages 155–210. Springer.

[Grupp et al., 2016] Grupp, M., Kopp, P., Huber, P., and Rätsch, M. (2016). A 3d face modelling approach for pose-invariant face recognition in a human-robot environment. In *Proc. Robot World Cup*, pages 121–134.

[Gu et al., 2002] Gu, X., Gortler, S. J., and Hoppe, H. (2002). Geometry images. *ACM Trans. Graph.*, 21(3):355–361.

[Güler et al., 2017] Güler, R. A., Trigeorgis, G., Antonakos, E., Snape, P., Zafeiriou, S., and Kokkinos, I. (2017). DenseReg: Fully convolutional dense shape regression in-the-wild. In *Proc. CVPR*, pages 6799–6808.

[Handa et al., 2016] Handa, A., Bloesch, M., Pătrăucean, V., Stent, S., McCormac, J., and Davison, A. (2016). gvnn: Neural network library for geometric computer vision. In *Proc. ECCV*, pages 67–82.

[Harandi and Fernando, 2016] Harandi, M. and Fernando, B. (2016). Generalized backpropagation, étude de cas: Orthogonality. *arXiv preprint arXiv:1611.05927*.

[Hartley and Vidal, 2008] Hartley, R. and Vidal, R. (2008). Perspective nonrigid shape and motion recovery. In *Proc. ECCV*, pages 276–289.

[Hartley and Zisserman, 2003] Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

[Hassner et al., 2015] Hassner, T., Harel, S., Paz, E., and Enbar, R. (2015). Effective face frontalization in unconstrained images. In *Proc. CVPR*, pages 4295–4304.

[He et al., 2017] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proc. ICCV*, pages 2980–2988.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778.

[Henriques and Vedaldi, 2017] Henriques, J. F. and Vedaldi, A. (2017). Warped convolutions: Efficient invariance to spatial transformations. In *Proc. ICML*, pages 1461–1469.

[Hill and Bruce, 1994] Hill, H. and Bruce, V. (1994). A comparison between the hollow–face and 'hollow-potato' illusions. *Perception*, 23(11):1335–1337.

[Hole et al., 2002] Hole, G. J., George, P. A., Eaves, K., and Rasek, A. (2002). Effects of geometric distortions on face-recognition performance. *Perception*, 31(10):1221–1240.

[Huang et al., 2012] Huang, G., Mattar, M., Lee, H., and Learned-Miller, E. G. (2012). Learning to align from scratch. In *Proc. NIPS*, pages 764–772.

[Huang et al., 2007] Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.

[Huber et al., 2015] Huber, P., Feng, Z.-H., Christmas, W., Kittler, J., and Ratsch, M. (2015). Fitting 3D morphable face models using local features. In *Proc. ICIP*, pages 1195–1199.

[Huttenlocher and Ullman, 1987] Huttenlocher, D. P. and Ullman, S. (1987). Object recognition using alignment. In *Proc. ICCV*, volume 87, pages 102–111.

[Ichim et al., 2015] Ichim, A. E., Bouaziz, S., and Pauly, M. (2015). Dynamic 3D avatar creation from hand-held video input. *ACM Trans. Graph.*, 34(4):45.

[Jackson et al., 2017] Jackson, A. S., Bulat, A., Argyriou, V., and Tzimiropoulos, G. (2017). Large pose 3D face reconstruction from a single image via direct volumetric cnn regression. In *Proc. ICCV*, pages 1031–1039.

[Jaderberg et al., 2015] Jaderberg, M., Simonyan, K., and Zisserman, A. (2015). Spatial transformer networks. In *Proc. NIPS*, pages 2017–2025.

[Jolliffe, 2002] Jolliffe, I. (2002). *Principal component analysis*. Springer.

[Jourabloo and Liu, 2016] Jourabloo, A. and Liu, X. (2016). Large-pose face alignment via CNN-based dense 3D model fitting. In *Proc. CVPR*, pages 4188–4196.

[Kae et al., 2013] Kae, A., Sohn, K., Lee, H., and Learned-Miller, E. (2013). Augmenting crfs with boltzmann machine shape priors for image labeling. In *Proc. CVPR*, pages 2019–2026.

[Kanazawa et al., 2016] Kanazawa, A., Jacobs, D. W., and Chandraker, M. (2016). Warpnet: Weakly supervised matching for single-view reconstruction. In *Proc. CVPR*, pages 3253–3261.

[Kazemi and Josephine, 2014] Kazemi, V. and Josephine, S. (2014). One millisecond face alignment with an ensemble of regression trees. In *Proc. CVPR*, pages 1867–1874.

[Keller et al., 2007] Keller, M., Knothe, R., and Vetter, T. (2007). 3D reconstruction of human faces from occluding contours. In *Proc. MIRAGE*, pages 261–273.

[Kemelmacher and Basri, 2005] Kemelmacher, I. and Basri, R. (2005). Indexing with unknown illumination and pose. In *Proc. CVPR*, pages 909–916.

[Kim et al., 2018] Kim, H., Zollhöfer, M., Tewari, A., Thies, J., Richardt, C., and Theobalt, C. (2018). InverseFaceNet: Deep single-shot inverse face rendering from a single image. In *Proc. CVPR*.

[Kleinberg et al., 2007] Kleinberg, K. F., Vanezis, P., and Burton, A. M. (2007). Failure of anthropometry as a facial identification technique using high-quality photographs. *J. Forensic Sci.*, 52(4):779–783.

[Knothe et al., 2006] Knothe, R., Romdhani, S., and Vetter, T. (2006). Combining pca and lfa for surface reconstruction from a sparse set of control points. In *Proc. FG*, pages 637–644.

[Koppen et al., 2018] Koppen, P., Feng, Z.-H., Kittler, J., Awais, M., Christmas, W., Wu, X.-J., and Yin, H.-F. (2018). Gaussian mixture 3d morphable face model. *Pattern Recognit.*, 74:617–628.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proc. NIPS*, pages 1097–1105.

[Lamond and Watson, 2004] Lamond, B. and Watson, G. (2004). Hybrid rendering-a new integration of photogrammetry and laser scanning for image based rendering. In *Proc. TPCG*, pages 179–186.

[Latto and Harper, 2007] Latto, R. and Harper, B. (2007). The non-realistic nature of photography: Further reasons why turner was wrong. *Leonardo*, 40(3):243–247.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324.

[Levoy et al., 2000] Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., and Ginsberg, J. (2000). The digital michelangelo project: 3d scanning of large statues. In *Proc. SIGGRAPH*, pages 131–144.

[Li et al., 2017] Li, T., Bolkart, T., Black, M. J., Li, H., and Romero, J. (2017). Learning a model of facial shape and expression from 4D scans. *ACM Trans. Graph.*, 36(6):194.

[Liu and Chaudhuri, 2003] Liu, C. H. and Chaudhuri, A. (2003). Face recognition with perspective transformation. *Vision Res.*, 43(23):2393–2402.

[Liu and Heidrich, 2003] Liu, Y. and Heidrich, W. (2003). Interactive 3d model acquisition and registration. In *Proc. PG*, pages 115–122.

[Long et al., 2014] Long, J. L., Zhang, N., and Darrell, T. (2014). Do convnets learn correspondence? In *Proc. NIPS*, pages 1601–1609.

[Lüthi et al., 2009] Lüthi, M., Albrecht, T., and Vetter, T. (2009). Probabilistic modeling and visualization of the flexibility in morphable models. In *Proc. Math. of Surf.*, pages 251–264.

[Lüthi et al., 2017] Lüthi, M., Gerig, T., Jud, C., and Vetter, T. (2017). Gaussian process morphable models. *IEEE Trans. Pattern Anal. Mach. Intell.*

[Makhzani et al., 2016] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2016). Adversarial autoencoders. In *Proc. ICLR*, pages 1–16.

[Martin Koestinger and Bischof, 2011] Martin Koestinger, Paul Wohlhart, P. M. R. and Bischof, H. (2011). Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *Proc. ICCV Workshops*, pages 2144–2151.

[Matthews and Baker, 2004] Matthews, I. and Baker, S. (2004). Active appearance models revisited. *Int. J. Comput. Vis.*, 60(2):135–164.

[Matthews et al., 2007] Matthews, I., Xiao, J., and Baker, S. (2007). 2d vs. 3d deformable face models: Representational power, construction, and real-time fitting. *Int. J. Comput. Vis.*, 75(1):93–113.

[Mitra et al., 2004] Mitra, N. J., Gelfand, N., Pottmann, H., and Guibas, L. (2004). Registration of point cloud data from a geometric optimization perspective. In *Proc. SIGGRAPH*, pages 22–31.

[Moghaddam et al., 2003] Moghaddam, B., Lee, J., Pfister, H., and Machiraju, R. (2003). Model-based 3D face capture with shape-from-silhouettes. In *Proc. AMFG*, pages 20–27.

[Moreno-Noguer and Fua, 2013] Moreno-Noguer, F. and Fua, P. (2013). Stochastic exploration of ambiguities for nonrigid shape recovery. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(2):463–475.

[Mpiperis et al., 2008] Mpiperis, I., Malassiotis, S., and Strintzis, M. G. (2008). Bilinear models for 3-d face and facial expression recognition. *IEEE Trans. Inf. Forensic Secur.*, 3(3):498–511.

[Nguyen et al., 2015] Nguyen, A., Yosinski, J., and Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proc. CVPR*, pages 427–436.

[Parkhi et al., 2015] Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. In *Proc. BMVC*, pages 41.1–41.12.

[Patel and Smith, 2009] Patel, A. and Smith, W. A. (2009). 3D morphable face models revisited. In *Proc. CVPR*, pages 1327–1334.

[Patel and Smith, 2016] Patel, A. and Smith, W. A. (2016). Manifold-based constraints for operations in face space. *Pattern Recognit.*, 52:206–217.

[Paysan et al., 2009] Paysan, P., Knothe, R., Amberg, B., Romdhani, S., and Vetter, T. (2009). A 3D face model for pose and illumination invariant face recognition. In *Proc. AVSS*, pages 296–301.

[Perona, 2007] Perona, P. (2007). A new perspective on portraiture. *J. Vis.*, 7(9):992–992.

[Pierrard and Vetter, 2007] Pierrard, J.-S. and Vetter, T. (2007). Skin detail analysis for face recognition. In *Proc. CVPR*, pages 1–8.

[Piotraschke and Blanz, 2016] Piotraschke, M. and Blanz, V. (2016). Automated 3D face reconstruction from multiple images using quality measures. In *Proc. CVPR*, pages 3418–3427.

[Porter and Doran, 2000] Porter, G. and Doran, G. (2000). An anatomical and photographic technique for forensic facial identification. *Forensic Sci. Int.*, 114(2):97–105.

[Prados et al., 2009] Prados, E., Jindal, N., and Soatto, S. (2009). A non-local approach to shape from ambient shading. In *Proc. SSVM*, pages 696–708.

[Ranjan et al., 2017a] Ranjan, R., Patel, V. M., and Chellappa, R. (2017a). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*

[Ranjan et al., 2017b] Ranjan, R., Sankaranarayanan, S., Castillo, C. D., and Chellappa, R. (2017b). An all-in-one convolutional neural network for face analysis. In *Proc. FG*, pages 17–24.

[Redmon et al., 2016] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proc. CVPR*, pages 779–788.

[Ren et al., 2015] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. NIPS*, pages 91–99.

[Ren and Malik, 2003] Ren, X. and Malik, J. (2003). Learning a classification model for segmentation. In *Proc. ICCV*, pages 10–17.

[Rhodes et al., 1987] Rhodes, G., Brennan, S., and Carey, S. (1987). Identification and ratings of caricatures: Implications for mental representations of faces. *Cogn. Psychol.*, 19(4):473–497.

[Richardson et al., 2016] Richardson, E., Sela, M., and Kimmel, R. (2016). 3D face reconstruction by learning from synthetic data. In *Proc. 3DV*, pages 460–469.

[Richardson et al., 2017] Richardson, E., Sela, M., Or-El, R., and Kimmel, R. (2017). Learning detailed face reconstruction from a single image. In *Proc. CVPR*, pages 5553–5562.

[Roberts et al., 2017] Roberts, J., Griffiths, F., and Verran, A. (2017). Seeing the baby, doing family: Commercial ultrasound as family practice? *Sociology*, 51(3):527–542.

[Rocco et al., 2017] Rocco, I., Arandjelović, R., and Sivic, J. (2017). Convolutional neural network architecture for geometric matching. In *Proc. CVPR*, pages 6148–6157.

[Romdhani et al., 2006] Romdhani, S., Ho, J., Vetter, T., and Kriegman, D. J. (2006). Face recognition using 3-d models: Pose and illumination. *Proc. IEEE*, 94(11):1977–1999.

[Romdhani and Vetter, 2003] Romdhani, S. and Vetter, T. (2003). Efficient, robust and accurate fitting of a 3d morphable model. In *Proc. ICCV*, pages 59–66.

[Romdhani and Vetter, 2005] Romdhani, S. and Vetter, T. (2005). Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *Proc. CVPR*, volume 2, pages 986–993.

[Rusinkiewicz et al., 2002] Rusinkiewicz, S., Hall-Holt, O., and Levoy, M. (2002). Real-time 3D model acquisition. *ACM Trans. Graph.*, 21(3):438–446.

[Rusinkiewicz and Levoy, 2001] Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the icp algorithm. In *Proc. 3DIM*, pages 145–152.

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., and Bernstein, M. (2015). Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252.

[Sagonas et al., 2016] Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. (2016). 300 faces in-the-wild challenge: Database and results. *Image Vis. Comput.*, 47:3–18.

[Saito et al., 2016] Saito, S., Li, T., and Li, H. (2016). Real-time facial segmentation and performance capture from rgb input. In *Proc. ECCV*, pages 244–261.

[Saito et al., 2017] Saito, S., Wei, L., Hu, L., Nagano, K., and Li, H. (2017). Photorealistic facial texture inference using deep neural networks. In *Proc. CVPR*, pages 5144–5153.

[Salzmann et al., 2007] Salzmann, M., Lepetit, V., and Fua, P. (2007). Deformable surface tracking ambiguities. In *Proc. CVPR*, pages 1–8.

[Sánchez-Escobedo et al., 2016] Sánchez-Escobedo, D., Castelán, M., and Smith, W. A. (2016). Statistical 3D face shape estimation from occluding contours. *Comput. Vis. Image Underst.*, 142:111–124.

[Sandford and Burton, 2014] Sandford, A. and Burton, A. M. (2014). Tolerance for distorted faces: Challenges to a configural processing account of familiar face recognition. *Cognition*, 132(3):262–268.

[Scherbaum et al., 2011] Scherbaum, K., Ritschel, T., Hullin, M., Thormählen, T., Blanz, V., and Seidel, H.-P. (2011). Computer-suggested facial makeup. *Comput. Graph. Forum*, 30(2):485–492.

[Scherbaum et al., 2007] Scherbaum, K., Sunkel, M., Seidel, H.-P., and Blanz, V. (2007). Prediction of individual non-linear aging trajectories of faces. *Comput. Graph. Forum*, 26(3):285–294.

[Schönborn et al., 2017] Schönborn, S., Egger, B., Morel-Forster, A., and Vetter, T. (2017). Markov chain monte carlo for automated face image analysis. *Int. J. Comput. Vis.*, 123(2):160–183.

[Schönborn et al., 2013] Schönborn, S., Forster, A., Egger, B., and Vetter, T. (2013). A monte carlo strategy to integrate detection and model-based face analysis. In *Proc. GCPR*, pages 101–110.

[Sela et al., 2017] Sela, M., Richardson, E., and Kimmel, R. (2017). Unrestricted facial geometry reconstruction using image-to-image translation. In *Proc. ICCV*, pages 1585–1594.

[Sim et al., 2003] Sim, T., Baker, S., and Bsat, M. (2003). The CMU pose, illumination, and expression database. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(12):1615–1618.

[Simon et al., 1994] Simon, D., Hebert, M., and Kanade, T. (1994). Real-time 3-d pose estimation using a high-speed range sensor. In *Proc. ICRA*, pages 2235–2241.

[Smith, 2016] Smith, W. A. (2016). The perspective face shape ambiguity. In *Perspectives in Shape Analysis*, pages 299–319. Springer.

[Smith and Hancock, 2006] Smith, W. A. and Hancock, E. R. (2006). Recovering facial shape using a statistical model of surface normal direction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(12):1914–1930.

[Sun et al., 2015] Sun, Y., Liang, D., Wang, X., and Tang, X. (2015). Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*.

[Suwajanakorn et al., 2014] Suwajanakorn, S., Kemelmacher-Shlizerman, I., and Seitz, S. M. (2014). Total moving face reconstruction. In *Proc. ECCV*, pages 796–812.

[Taigman et al., 2014] Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proc. CVPR*, pages 1701–1708.

[Tewari et al., 2017] Tewari, A., Zollhöfer, M., Kim, H., Garrido, P., Bernard, F., Pérez, P., and Theobalt, C. (2017). MoFA: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proc. ICCV*, pages 1274–1283.

[Thewlis et al., 2017a] Thewlis, J., Bilen, H., and Vedaldi, A. (2017a). Unsupervised learning of object frames by dense equivariant image labelling. In *Proc. NIPS*, pages 844–855.

[Thewlis et al., 2017b] Thewlis, J., Bilen, H., and Vedaldi, A. (2017b). Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proc. ICCV*, pages 5916–5925.

[Thies et al., 2016] Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., and Nießner, M. (2016). Face2face: Real-time face capture and reenactment of rgb videos. In *Proc. CVPR*, pages 2387–2395.

[Tran et al., 2017] Tran, A. T., Hassner, T., Masi, I., and Medioni, G. (2017). Regressing robust and discriminative 3d morphable models with a very deep neural network. In *Proc. CVPR*, pages 1493–1502.

[Turk and Pentland, 1991] Turk, M. and Pentland, A. P. (1991). Face recognition using eigenfaces. In *Proc. CVPR*, pages 586–591.

[Uřičář et al., 2012] Uřičář, M., Franc, V., and Hlaváč, V. (2012). Detector of facial landmarks learned by the structured output svm. In *Proc. VISAPP*, pages 547–556.

[Valente and Soatto, 2015] Valente, J. and Soatto, S. (2015). Perspective distortion modeling, learning and compensation. In *Proc. CVPR Workshops*, pages 9–16.

[Vicente et al., 2015] Vicente, F., Huang, Z., Xiong, X., De la Torre, F., Zhang, W., and Levi, D. (2015). Driver gaze tracking and eyes off the road detection system. *IEEE Trans. Intell. Transp. Syst.*, 16(4):2014–2027.

[Vlasic et al., 2005] Vlasic, D., Brand, M., Pfister, H., and Popović, J. (2005). Face transfer with multilinear models. *ACM Trans. Graph.*, 24(3):426–433.

[Wu et al., 2016] Wu, C., Bradley, D., Garrido, P., Zollhöfer, M., Theobalt, C., Gross, M., and Beeler, T. (2016). Model-based teeth reconstruction. *ACM Trans. Graph.*, 35(6):220.

[Wu et al., 2017] Wu, W., Kan, M., Liu, X., Yang, Y., Shan, S., and Chen, X. (2017). Recursive spatial transformer (REST) for alignment-free face recognition. In *Proc. CVPR*, pages 3792–3800.

[Yan et al., 2016] Yan, X., Yang, J., Yumer, E., Guo, Y., and Lee, H. (2016). Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In *Proc. NIPS*, pages 1696–1704.

[Yu et al., 2017] Yu, R., Saito, S., Li, H., Ceylan, D., and Li, H. (2017). Learning dense facial correspondences in unconstrained images. In *Proc. CVPR*, pages 4723–4732.

[Yu et al., 2016] Yu, X., Zhou, F., and Chandraker, M. (2016). Deep deformation network for object landmark localization. In *Proc. ECCV*, pages 52–70.

[Zhang et al., 2008] Zhang, W., Sun, J., and Tang, X. (2008). Cat head detection-how to effectively exploit shape and texture features. In *Proc. ECCV*, pages 802–816.

[Zhong et al., 2017] Zhong, Y., Chen, J., and Huang, B. (2017). Toward end-to-end face recognition through alignment learning. *IEEE Signal Process. Lett.*, 24(8):1213–1217.

[Zhu et al., 2016] Zhu, X., Lei, Z., Liu, X., Shi, H., and Li, S. Z. (2016). Face alignment across large poses: A 3D solution. In *Proc. CVPR*, pages 146–155.

[Zhu et al., 2015] Zhu, X., Lei, Z., Yan, J., Yi, D., and Li, S. Z. (2015). High-fidelity pose and expression normalization for face recognition in the wild. In *Proc. CVPR*, pages 787–796.

[Zhu and Ramanan, 2012] Zhu, X. and Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *Proc. CVPR*, pages 2879–2886.