3D Face and Body Reconstruction via Volumetric Regression Networks



Aaron S. Jackson

Supervised by Dr. Georgios Tzimiropoulos

Computer Vision Laboratory

School of Computer Science

University of Nottingham

This dissertation is submitted for the degree of

Doctor of Philosophy

September 2019

Acknowledgements

First and foremost, I would like to begin by thanking my supervisor, Georgios Tzimiropoulos, for his tireless encouragement, advice, guidance and support throughout my PhD. I would also like to thank the Computer Vision Laboratory as a whole for being a welcoming, friendly and supportive environment - in particular, B86: Dimitris, Ioanna, Jing, Joy, Keerthy, Kike and Siyang.

I would also like to thank my friends over in `symbollox` for keeping me entertained throughout the day, but also for their support in early 2018: `j‘ey`, `mgsk`, `mister_m`, `moparisthebest`, `penryu`, `pie__`, `qu1j0t3` and `stylewarning`.

Thank you to Joe, most importantly for being a friend I can talk to about anything, but also an excellent colleague who has always helped me. Also, thanks to my housemate and friend Tom, although I'm not *really* sure what for – just a bit of everything I guess.

Finally, thank you to my mum (Charlotte), dad (Ian), brother (Wes) and sister (Nic), for their encouragement and support over the past few years.

bnl

1995–2018

Abstract

3D Face reconstruction is the process of estimating the full 3D geometry of a human's face from one or more images. Applications of 3D face reconstruction span many areas, from personalisation of video games and trying on accessories online, to measuring emotional arousal for psychological studies and in medicine, such as simulating the result of reconstructive surgery. Approaches to 3D face reconstruction generally depend on a 3D Morphable Model (3DMM) - a parametric model, where the shape, pose and expression can be adjusted using a small number of parameters. While methods based on such techniques can work well on frontal images, they often begin to fail on cases of large pose, difficult expression, occlusion, and bad lighting. Additionally, encoding detail in so few parameters is not possible.

In this thesis, we propose a novel approach to the problem of 3D face reconstruction: *Volumetric Regression Networks*. Our non-parametric approach constrains the problem to the spatial domain using an end-to-end network which directly regresses the 3D geometry using a volumetric representation. This avoids the need for 3DMM generation, which involves finding correspondence between all vertices of all training samples, but also the fitting stage, which requires solving a difficult optimisation problem. We demonstrate that doing so can not only provide state-of-the-art results, but also be adapted to other deformable objects, such as the full human body.

Table of contents

List of figures **x**

List of tables **xiii**

1 Introduction **1**

1.1 Problem 1

1.2 Where do we fit in? 6

1.3 Contributions 8

1.4 Outline 10

1.5 Publications 11

2 Background **12**

2.1 Deep Learning 12

2.1.1 Convolutional Neural Networks 13

2.1.2 Batch Normalisation 14

2.1.3 Frameworks 15

2.2 3D Volumes and Volumetric Algorithms 16

2.2.1 Voxelisation 17

2.2.2 Surface Extraction 18

2.2.3 Size, Compression and Formats 19

3 Literature Review 21

3.1 Face Reconstruction 21

3.1.1 3D Morphable Models (3DMM) 22

3.1.2 Synthetic Training Data for 3DMM 24

3.1.3 Shape from Shading 24

3.1.4 Multiview Reconstruction and Structure from Motion 25

3.1.5 Depth Estimation 26

3.1.6 Where does our work fit in? 26

3.2 Human Body Reconstruction 27

3.3 Landmark Detection 28

3.3.1 Face Alignment 29

3.3.2 Human Pose Estimation 30

4 Large Pose 3D Face Reconstruction via Volumetric Regression 33

4.1 Introduction 34

4.1.1 Main contributions 36

4.2 Method 37

4.2.1 Training Datasets 37

4.2.2 Proposed volumetric representation 38

4.2.3 Volumetric Regression Networks 40
4.2.4 Training 44
4.3 Results 44
4.3.1 Evaluation Datasets 44
4.3.2 Evaluation metric 46
4.3.3 Evaluation 47
4.3.3.1 Comparison with state-of-the-art 48
4.3.4 Training and Run Times 51
4.3.5 Results with ICP Alignment 52
4.4 Importance of spatial alignment 52
4.5 Ablation studies 55
4.5.1 Effect of pose 55
4.5.2 Effect of expression 56
4.5.3 Effect of Gaussian size for guidance 58
4.6 Architectural Experiments 58
4.6.1 On intermediate supervision 59
4.6.2 On increasing the number of stacks vs the number of modules . 59
4.6.3 On varying the network parameters 60
4.6.4 A Volumetric Implementation 61
4.6.5 Architectural Conclusions 62
4.7 Conclusions 64

5 3D Human Body Reconstruction from a Single Image via Volumetric Regression 65

5.1 Introduction . . . 66

5.2 Method . . . 68

5.2.1 Volumetric Regression . . . 68

5.2.2 Dataset and Voxelisation . . . 69

5.2.3 Training . . . 70

5.3 Architecture . . . 71

5.3.1 Multistack . . . 71

5.3.2 Image Only . . . 72

5.3.3 Landmarks Only . . . 73

5.3.4 Mask Only . . . 73

5.3.5 3D Convolution . . . 73

5.4 Results . . . 74

5.5 High Quality Training Data . . . 77

5.5.1 Performance . . . 79

5.6 Conclusion . . . 79

6 A CNN Cascade for Landmark Guided Semantic Part Segmentation 81

6.1 Introduction . . . 82

6.1.1 Main contributions . . . 83

6.2 Literature Review . . . 84

6.2.1 Semantic Segmentation 84
6.2.2 Part Segmentation 88
6.2.3 Facial Part Segmentation 88
6.3 Datasets 89
6.4 Method 90
6.5 Experiments 93
6.5.1 Overview of Results 93
6.5.2 Unguided Facial Part Segmentation 94
6.5.3 Guided Facial Part Segmentation with Groundtruth 95
6.5.4 Guided Facial Part Segmentation with Detected Landmarks . . 96
6.6 Conclusion 97

7 Conclusion **100**

7.1 Future Work 101
7.1.1 Detail Refinement 101
7.1.2 Texture Synthesis 102
7.1.3 Fixed Correspondence 102
7.1.4 Temporal Consistency 102
7.1.5 Performance and Speed 103

References **104**

List of figures

1.1 Visual examples of challenging images 3

1.2 3D Morphable Model Fitting Procedure 6

2.1 Volumetric antialiasing at different resolutions 17

2.2 Voxelisation error due to limiting ray tracing directions 18

4.1 A 3D Reconstruction Preview 33

4.2 Dataset voxelisation procedure 39

4.3 Error due to voxelisation 40

4.4 Overview of proposed architectures 42

4.5 Binary vs Real volumes 43

4.6 Example BU-4DFE and Florence renderings 46

4.7 NME performance on AFLW2000-3D images 48

4.8 NME performance on BU-4DFE renderings 49

4.9 NME performance on Florence renderings 49

4.10 Visual results on AFLW2000-3D dataset 50

4.11 Some very difficult failure cases 50

4.12 Visual results on the 300VW dataset 51

4.13 NME performance on AFLW2000-3D with ICP Alignment 53

4.14 NME performance on BU4DFE with ICP Alignment 53

4.15 NME performance on Florence with ICP Alignment 54

4.16 Visual results when spatial alignment is ignored 55

4.17 Effect of pose 56

4.18 The six facial expressions available in the BU-4DFE dataset 57

4.19 Effect of facial expressions 57

4.20 NME performance from architectural experiments 62

4.21 Visual comparison of number of parameters 63

5.1 Example reconstructions 65

5.2 Proposed stacked architecture 72

5.3 A “flat” volumetric residual block 74

5.4 Visual results from our main architecture 75

5.5 Visual results when trained on high quality data 78

6.1 Examples from our facial part segmentation method 83

6.2 VGG-16 Network 84

6.3 Example groundtruth facial part masks 89

6.4 Our proposed end-to-end architecture for facial part segmentation . . . 91

6.5 Unguided vs. Guided facial part segmentation 94

6.6 Groundtruth Guided vs. Unguided facial part segmentation 95

6.7 Groundtruth vs. Detected landmarks for guidance 96

6.8 Average images and their output from facial part segmentation 97

6.9 Challenging images and their output from facial part segmentation . . 98

List of tables

4.1 Numerical performance of 3D face reconstruction 45

4.2 Numerical performance of 3D face reconstruction with ICP 52

4.3 Overview of architectural experiments 58

5.1 Numerical performance of human body reconstruction 76

6.1 Layers of VGG-FCN 90

Chapter 1

Introduction

The landscape of computer vision has shifted dramatically over the past decade. Prior to this, general purpose solutions to difficult problems had low success rates on unconstrained images. This is especially true for the primary focus of this thesis: 3D face reconstruction. This chapter begins by giving an introduction to the problem of 3D face reconstruction. We attempt to justify the need for research in this area by highlighting many of the important applications, as well as the limitations of existing work. Additionally, this chapter attempts to explain where our own work on this problem fits into the area of 3D face reconstruction. Finally, we give an overview of the structure of this thesis.

1.1 Problem

3D face reconstruction is the process of estimating the 3D geometry of a person's face, but what exactly do we even mean by geometry? Fundamentally, this geometry is a set of 3D vertices, connected by edges, which together defines a 3D surface. This is more generally referred to as a mesh. This mesh should capture the shape, expression

and pose of the face. In a perfect world, it should also capture finer details, such as facial hair, eye brows, eye lashes, wrinkles, spots, blemishes and anything else we as humans can see on a person's face. Ideally, it should also capture facial accessories, such as glasses and jewellery.

Well, even with the laser precision of commercial 3D scanners, satisfying all of these desires is currently impossible. Despite this, 3D scanners are the go to solution when it comes to obtaining a high resolution facial mesh. However, the use of such equipment comes with many limitations. Perhaps worst of all is that the person being scanned has to sit very still while the scan is being performed, surrounded by some large and intimidating machine. In fact, haven't we been here before? This all sounds very similar to photography in the early 1800's. Mercury vapour aside - "Sir! Stand still, please! There is at least five more minutes until our photo has exposed!"

Many applications of 3D face reconstruction cannot tolerate these constraints. Such examples span a wide range of areas. The more obvious being personalisation of computer games, or trying on accessories online, such as glasses. However, this area has many other applications, such as facial expression analysis for measuring emotional arousal, where 3D face reconstruction can be a useful tool in psychological studies. Facial behaviour transfer is another potential application of this area of work and is widely used in creative industries, such as the development of computer games and animated films where typically a large number facial point markers have to be attached to the actor's face. Additionally, standing perfectly still could certainly have a negative impact on the actor's behaviours. There is also the potential for medical applications, such as simulating the result of reconstructive surgery after an accident.

Due to the nature of these applications, there is a clear demand for reliability under unconstrained environments, which regularly cannot be catered to by 3D scanning. As such, there are a variety of approaches which attempt to perform 3D face reconstruction



Occlusion Large Pose Illumination Expression

Figure 1.1 Some visual examples of common problems which 3D Face Reconstruction methods have to be able to tolerate, taken from the AFW dataset [121].

under more typical settings. The inputs to these methods vary widely, from sets of images in predefined poses and video, to reconstruction from a single unconstrained image. Some of the difficulties which all 3D face reconstructions would *like* to be able to address are displayed in Figure 1.1, and discussed in detail the next few paragraphs.

Occlusion. In computer vision, occlusion refers to the visual obstruction of an object. In the case of faces, this could be due to any number of reasons, from nose scratching to accessory wearing and beyond. As humans, we can look past this occlusion. Faces do not suddenly stop being faces just because there's a hand in the way. We can also make assumptions about what the shape or appearance of a face might be based on the information available with a high level of confidence. Unfortunately, this is much harder for a computer. We have to encode this information some how, and tie pieces together with knowledge, such as how faces are generally symmetric. An approach which deals with this problem quite well is to use a 3D Morphable Model (3DMM), which we'll discuss in much more detail later in this chapter.

Large Pose refers to any facial pose which defers too much from a frontal face, that is, a face which is looking directly at the camera. For many methods this is between $45° - 90°$, at which point a large portion of the face is self occluded. Many methods

attempt to work around this problem by using multiple images from different views. This is referred to as multiview 3D face reconstruction and has its own set of problems which are discussed later in this section, other approaches are model based, and are therefore able to make a reliable assumption about the self occluded part of the face.

Illumination can cause issues for many methods. For some methods, a perfectly illuminated face without any shadows is optimal. For other methods, such as those based on a Shape-from-Shading (SfS) approaches, having no shadows at all is equally as big a problem.

Facial Expression. Some facial expressions are more challenging than others, a neutral expression typically being the easiest to model and often the most prevalent in facial scan datasets. Take smiling for instance, smiling result in large deformation of the 3D geometry around the mouth, but also to the sides of the nose and around the eyes. The extent of this deformation varies largely from person to person, making many facial expressions challenging and the relationship between areas of the face under different expressions difficult to model.

Detail is difficult for two main reasons. First, an approach which is capable of encoding a high level of detail is required. This is often challenging with model fitting based approaches because the inputs need to be constrained to a small number of principal components which vary aspects such as shape, pose and rotation. This constraint usually leads to post-processing steps to extract the detail, which can destroy correspondence between vertices usually present in a parametric model, as well as pose and scale information. Second, there need to be some way to distinguish between what is actually *detail*, such as wrinkles and spots which affect 3D shape versus what is noise, makeup (potentially tattoos), dirt or anything else affecting only textural information.

Lab Conditions vs. In-the-Wild. Due to the difficulties involved with using 3D scanning equipment, discussed earlier, facial scan datasets are always collected under lab conditions. This typically limits the amount of variation available across many aspects, for example lighting and illumination and background scenery. 3D face reconstruction methods which are then trained on such data may struggle to transfer their knowledge to what are known as "In-the-Wild" images. Such images are very diverse compared to what may be available under constrained lab conditions.

Single Image vs. Multiview or Video. Methods capable of using or requiring multiple views or video often have a lot more information available to them. Parts of the face which are typically self occluded under the single image setting may be available to the method if multiple images are available. However, taking multiple images of fixed pose and expression is a problem in its own right, and it is therefore preferable use only a single image without having a detrimental effect on reconstruction performance.

Our approach to 3D face reconstruction has proven resilient to many of the aforementioned challenges. We demonstrate good performance on occluded faces, show only a small reduction in performance for large poses and difficult facial expressions, work across challenging illumination on rendered datasets, and work on *in-the-wild* images and use only a single image. In the case of detail, we demonstrate that our approach is capable of estimating the fine details in Chapter 5, in which we demonstrate our method also working on human body reconstruction with high levels of detail.

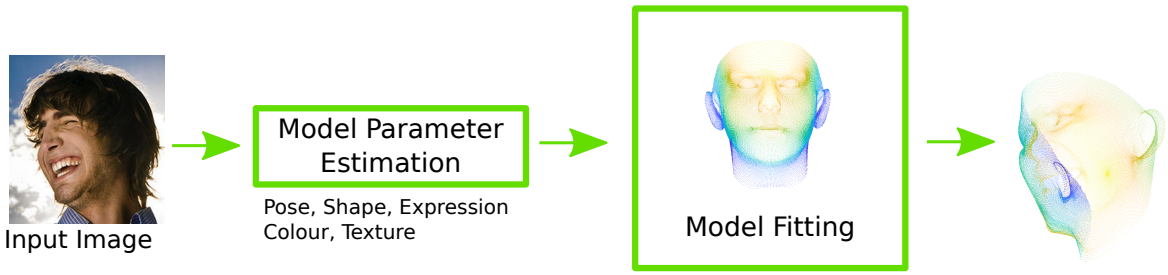


Figure 1.2 The fitting procedure of a 3D Morphable Model (3DMM)

1.2 Where do we fit in?

There are a number of approaches to 3D face reconstruction. Each general approach has its own set of advantages and disadvantages. The aim of this section is to give a brief overview of the general approaches to this problem. By doing so, we aim to show where and how our own work fits in, along with why this is important.

3D Morphable Models (3DMM) are perhaps the most popular approach to estimating facial 3D geometry. These methods use a vector of parameters for a pre-computed face model [52, 43, 120, 63], which varies the shape, expression, pose and optionally textural information. We discuss 3DMM in more detail in Chapter 3, although a brief explanation of the fitting procedure is depicted in Figure 1.2. These methods are increasingly using Convolutional Neural Networks as a means to regress the parameters, and in general, work very reliably on *normal* faces. However, 3DMM based approaches gradually perform worse as factors such as pose and expression increase. We show in Section 4.5 that our method has a very uniform and low error over all expressions, with only a small increase in error on large poses (such as around 80 degrees). Another disadvantage of 3DMM based approaches is that the 3DMM must be generated - this requires finding the dense correspondence between all vertices of all samples, which also becomes difficult as pose and expression changes. Finally,

since these methods are model based, they are unable to directly produce finer details, such as wrinkles. In an extension to our own work, we show in Chapter 5 (where we perform 3D reconstruction of humans), that our method is able to directly regress fine details in clothing.

Multiview and Photogrammetry based methods require multiple images from many different angles in order to estimate the 3D geometry of a face (or other object) [31, 29, 81, 72]. The primary drawback of such methods is, of course, that multiple images are not always available, especially in unconstrained recording conditions (under the aforementioned environmental constraints) associated with 3D face reconstruction. While the final 3D reconstruction from such methods can be of very high resolution and detailed, finding corresponding features between each image is both memory and CPU intensive. Our own method uses only a single RGB image, without any additional depth data - from this single image, we regress the full 3D facial geometry.

Shape-from-Shading (SfS) methods attempt to recover the 3D geometry by analysing the shading and reflectance of a face relative to a *albedo* (mean) face. Similarly to 3DMM based methods, SfS methods often rely on a 3D model, inheriting many of the same problems, such as finding a dense correspondence [98, 50]. Additionally, many of these methods also require more than one image, which means SfS methods often also inherit the problems associated with multiview methods. Finally, more so than other methods, SfS methods struggle with lighting - over or under illuminated faces may produce poor results when compared to other methods. Our approach to 3D face reconstruction works under almost all kinds of lighting, and we show this on a challenging synthetic dataset.

Depth Estimation is arguable more popular among problems such as room reconstruction, however, there are several methods which attempt to reconstruct the frontal part of the facial 3D geometry using this technique [96, 97]. It would be fair to claim that these methods have declined in popularity. The biggest flaw of depth estimation based approaches is that only the visible parts can be reconstructed, unless other methods are employed afterwards (such as 3DMM, again, inheriting many of the aforementioned problems). While also working from a single image, our method is able to hallucinate the non-visible parts of the face, including, to a certain extent, those which have been self occluded without having to handle this cases on an individual basis.

Given the numerous applications of such a technology and the significant disadvantages of the already existing methods, we feel there is a profound requirement for something more robust to the challenges in this area. In this thesis we approach the problem using a single image, and propose a new non-parametric approach to 3D face reconstruction which attempts to account for as many of these challenges as possible.

1.3 Contributions

The contributions of this thesis can be summarised as follows:

- We propose an alternative representation for the 3D geometry which, unlike X, Y, Z vertices, can be directly regressed using a Convolutional Neural Network (CNN). Our method is non-parametric and does not use a shape model. This offers the advantage of being able to regress detail without further post processing steps.

- We present a deeply learnt method for 3D face reconstruction, which can be trained directly in an end-to-end fashion. Our method reconstructs the full 3D facial structure, including parts which are occluded. We also show that our method works from just a single image without requiring accurate alignment or establishing a dense correspondence between all training samples. This allows us to bypass the construction (during training) and fitting (during inference) of a 3D Morphable Model (3DMM). We refer to this method as the **Volumetric Regression Network (VRN)**.

- We show how a related task of facial landmark localisation can be incorporated into our method (also in an end-to-end fashion) to improve the reconstruction quality, particularly on expressive faces in non-frontal views.

- We evaluate our method on both constrained and heavily unconstrained images from the web, demonstrating that our method **outperforms all prior work** on single image 3D face reconstruction by a large margin.

- We show that our method works on other deformable objects, such as in the task of **human body reconstruction**, and that provided high quality training data is available, our method is able to both reconstruct difficult human poses and **fine details such as wrinkles** in clothing.

- We release our code for 3D face reconstruction under the MIT license, enabling our method to be used for both commercial and personal purposes. In addition, we have provided a free online service for automatic generation of 3D reconstructions from an uploaded image.

1.4 Outline

In **Chapter 2**, an introduction to the background material is provided, discussing volumetric representations and algorithms, such as voxelisation, surface extraction and some of the difficulties associated with volumetric representations.

Chapter 3 gives an overview of related literature, starting with face alignment, which is often a prerequisite to 3D face reconstruction methods. Continuing on, a discussion of the various existing 3D face reconstruction is given. The remainder of this chapter gives an overview of human pose estimation and human body reconstruction.

Chapter 4 is the primary focus of this thesis. This chapter describes our approach to 3D face reconstruction, which includes details about the datasets used, the volumetric representation and the (small) amount of error it introduces, along with our approach and network architectures. This chapter also describes the numerous experiments carried out on the various datasets, comparisons with other methods, and ablation studies on pose, expression and influence of parameters used for guidance. Finally, we explore some of the architectural options available for this method, which result in performance degradation or boosts.

An extension to our work on 3D face reconstruction is described in **Chapter 5**, in which we show how our method can be applied to on another category of deformable objects: the full human body. We carry out experiments on two datasets, showing that our approach can handle both difficult poses and finer details.

Our first work, focusing on landmark guided semantic part segmentation of the human face is described in **Chapter 6**. It is important to mention here that this thesis does not focus on semantic segmentation, but this work *heavily* influenced our approach to the problem of 3D reconstruction, since our method shifts the 3D reconstruction problem into a volumetric segmentation problem.

Finally, we conclude with a reflection and discussion of our work, what impact it may have had and the numerous areas worthy of attention as future work, in **Chapter 7**.

1.5 Publications

Several publications came as a result of the work leading up to this thesis:

- Jackson, A.S., Valstar, M. and Tzimiropoulos, G., 2016, October. A CNN cascade for landmark guided semantic part segmentation. In European Conference on Computer Vision (Workshops) (pp. 143-155). Springer, Cham.
- Jackson, A.S., Bulat, A., Argyriou, V. and Tzimiropoulos, G., 2017, October. Large pose 3D face reconstruction from a single image via direct volumetric CNN regression. In Computer Vision (ICCV), 2017 IEEE International Conference on (pp. 1031-1039). IEEE.
- Jackson, A.S., Manafas, C., & Tzimiropoulos, G., 2018 3D Human Body Reconstruction from a Single Image via Volumetric Regression. In European Conference on Computer Vision (Workshops).

Chapter 2

Background

This chapter gives an introduction to deep learning, the volumetric representation we use for encoding 3D geometry, along with some of the algorithms associated with volumetric data, such as voxelisation and surface extraction.

2.1 Deep Learning

Neural networks have significantly grown in size since their inception. A neural network model of today may contain as many as fifty million parameters. This establishes a difficult, extremely large, non-convex optimisation problem. The only way to sensibly train these models, ignoring minor implementation details, is to use gradient descent optimisation.

To assist in the explanation of gradient descent, suppose a classification problem with annotated training data $Z = \{(x_i, y_i)\}$, containing images x_i each with a corresponding label y_i. We define f as the classifier, mapping image x_i to label y_i given a real vector

of parameters θ,

$$y_i \approx f(x_i; \theta). \tag{2.1}$$

To measure the performance this classifier, a loss function ℓ is defined, which is entirely dependant on the problem at hand. Hence, our loss over x is defined as,

$$loss = \sum_i \ell(y_i, f(x_i; \theta). \tag{2.2}$$

Initially parameters θ will be unknown. Hence, in order to find θ (so that we can establish a trained classifier), we must minimise the loss function,

$$\min_\theta \sum_i \ell(y_i, f(x_i; \theta)). \tag{2.3}$$

This minimisation is typically performed using gradient descent, where θ is gradually learnt over many iterations. This is at rate η (known as the *learning rate*), and is directed by the gradient of the f with parameters θ,

$$\theta = \theta - \eta \nabla_\theta \ell(y_i, f(x_i; \theta). \tag{2.4}$$

More often than not, the above is extended to support batches of images per iteration, which results in a more stable gradient due to the averaging across several images.

2.1.1 Convolutional Neural Networks

In the previous subsection, an overview of gradient descent was presented, where parameters θ were used to *in some way* to apply a mapping from an input to a

label. Such a function might be defined as simply as a linear function, $f(x;\theta) = \theta x$. Unfortunately, despite being appropriate in many areas, a single linear function will struggle to map a non-linear problem, which is almost certainly the case with all computer vision based applications. As such, convolutions are often used to both reduce the number of parameters, and to allow convenient and efficient stacking, allowing the network to learn difficult mappings. These convolution operations perform a weighted sum of a window which moves across the input with a particular stride.

Convolutional Neural Networks (CNNs) are constructed, typically, from stacks of convolutional filters, each followed by some form of activation function. These activation functions discard unwanted information, making it easier for future layers to make use of the extracted features. A commonly used activation function is the Rectified Linear Unit (ReLU), which discards negative values, setting them to zero. The tailing end of a CNN is often specific to each application. For example, for a numerical regression task, linear layers may be added, and for segmentation, a 1×1 convolution may be added. As these networks have gradually increased in size, the term *Deep Learning* has become popular. Networks of today may consist of several hundred convolutional layers, such as ResNet-152 [39].

2.1.2 Batch Normalisation

As networks grew from 15 or so layers upwards, the job of gradient descent became very difficult. During training, the distribution of every input to every layer changes slightly. This is because the parameters of the layer before each layer change. This problem is referred to as *internal covariate shift*. A solution known as Batch Normalisation was proposed in [44], enabling networks with many layers (such as the 152 layers in ResNet [39]) to be trained in much shorter times. Some additional advantages of using

batch normalisation include being able to use larger learning rates and reducing the need for such careful initialisation.

Batch normalisation works as the name might suggest, by normalising the output of each parameterised layer using learnable scale and shift parameters, as well as the size of each minibatch, for each minibatch. Understanding exactly how batch normalisation works is not imperative to understanding the contents of this thesis. However, in general, batch normalisation has been applied either before or after every parameterised layer of the CNN models proposed in this thesis. Additionally, as these networks are very large, they are unlikely to have been possible to train without employing batch normalisation.

2.1.3 Frameworks

While designing and developing a framework to work with deep neural networks *might* be an interesting thought experiment for some, it is far from the focus of this thesis. The work presented throughout this thesis was developed primarily under the Torch [1]. There are a number of reasons for this, but to name just a few:

1. Efficient GPU support. This is important, of course, in any framework intended for deep learning. Torch enables networks to be run in parallel (for speed) or sequentially (for additional memory) across multiple GPUs.

2. Elegant memory backend. Data can be loaded directly into memory and views can be applied to create N-dimensional Tensors without shuffling memory. This is very important when reading in large volumetric training samples.

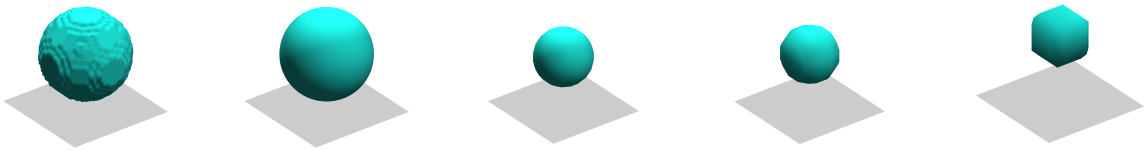
[1] http://torch.ch/, except for the work on facial part segmentation, which was developed under Cafe [47]

3. Multithreading. The trade offs between I/O and CPU are discussed in more detail in Section 2.2.3. However, regardless of how the data is optimised (in terms of compression or just raw data), volumes are big. Loading them sequentially and applying augmentation, even on very fast hardware, remains a computationally expensive task.

2.2 3D Volumes and Volumetric Algorithms

Typically, 3D models are provided as a mesh, defined by vertices and faces, in a particular file format such as *Wavefront Obj*. These files typically contain additional data, such as texture mapping coordinates and surface normals. However, to represent the shape of an object, only the mesh is required. Directly regressing the vertices for this mesh using a CNN, along with a predefined set of faces, would be a very elegant approach to 3D face reconstruction. However, such an approach has never been demonstrated before, and it is likely that it isn't possible using a CNN unless the dimensionality of the problem is first reduced (hence the use of 3DMM).

An alternative approach to representing the shape of an object is to encode it in a three dimensional volume. Typically, each voxel stores an intensity value, much like a grey scale image, collected using an MRI scanner for example. However, by storing only binary values, we can encode the 3D shape of an object. This may be thought of as a 3D matrix, where *inside* the object is denoted by a one, and *outside* is denoted with a zero. This allows us to constrain the problem of 3D face reconstruction to one that spatial, by directly learning a mapping between image and the 3D volume, avoiding the need to learn mappings to a parametric or Euclidean space, which is much more difficult.



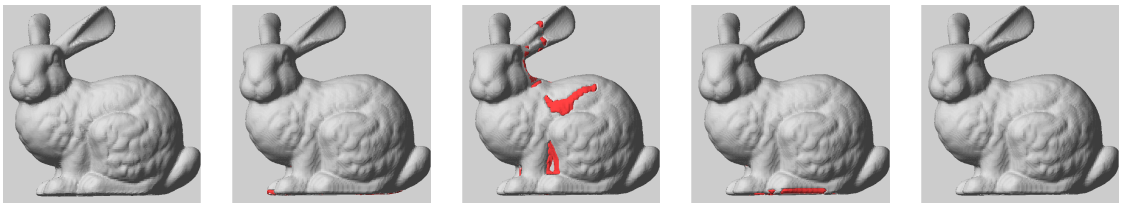
Binary 32^3 | Real 32^3 | Real 16^3 | Real 8^3 | Real 4^3

Figure 2.1 A volumetric sphere, with and without antialiasing, at several different resolutions.

2.2.1 Voxelisation

The process of transforming a mesh into a volumetric structure is referred to as voxelisation. Intuitively, this voxelisation process discretises the mesh. It is no longer possible to reproduce a non-integer form of the original mesh - a surface noticeably constructed out of small cubes. Ideally, there should be minimal visual degradation between viewing the original mesh and the voxelised object, otherwise, how would it be possible to encode detail? Just as anti-aliasing is possible in 2D graphics, it is also possible with 3D volumes. This is achieved by using real numbers inside a voxel, instead of just a binary value. This results in very low distortion, provided the resolution is sensibly selected. This is shown in Figure 2.1, where a sphere encoded using binary values is clearly built out of cubes, where as even a small real value volume remains representative.

Generating a 2D image from 2D vector graphics (such as polygons) is called rasterisation. This involves tracing a ray while counting the number of times the ray intersects a boundary of the object or shape. If the count is odd, colour in pixel and move onto the next. If the count is even, just move onto the next pixel. Voxelisation follows a similar idea, except it is performed in three dimensions instead of two. This is done by tracing rays through the X, Y and Z planes separately to produce three



Original Mesh | Voxelised Traced X | Voxelised Traced Y | Voxelised Traced Z | Voxelised Traced XYZ

Figure 2.2 Voxelised results of the Stanford Bunny (original shown on left). The first three volumes have been voxelised by tracing through one of the X, Y, or Z planes (e.g. from front to back). The final volume (right most) is generated by voting. Red patches fill in voxels which have been missed. Each volume is 128^3 in size.

separate volumes. These volumes are then combined in some way, typically by ensuring each XYZ coordinate has been set in at least two volumes (voting). We show this visually in Figure 2.2, where error (shown in red) has been introduced by only voxelising the Stanford Bunny[2] from a single direction.

2.2.2 Surface Extraction

Surface extraction is the process of extracting a mesh which encloses a 3D object represented in volumetric space. Occasionally such methods require thresholds to extract separate regions of the volume, for example, in an MRI scan, where the brain needs to be visualised independently of the skull (and hence, independently extracted from the volume).

Perhaps the most popular method for computing the isosurface of a volume is Marching Cubes [67], developed in 1987. The algorithm traverses through all voxels, using the value of eight neighbouring voxels to form eight binary values, a single byte. This byte can be used to lookup a known polygon configuration from a table of $2^8 = 256$ entries. However, in the first version of Marching Cubes, only 15 unique

[2]http://graphics.stanford.edu/data/3Dscanrep

polygon configurations were stored. This lead to cases of ambiguity when performing a lookup for certain voxel configurations. The side effect of this is that meshes would sometimes end up having holes in them. In [25], additional unique configurations were proposed, leading to a total of 33. This removes the ambiguity, and results in the meshes being closed or *water tight*.

2.2.3 Size, Compression and Formats

There are many more factors which have to be considered when working with volumetric representations of objects, as opposed to a standard mesh.

To start with, the **size** of the volume (in terms of bytes) grows much more rapidly than it would in a 2D image. In order to represent a sensible amount of depth in a 128×128 image, a volume of 128^3 voxels is needed. If this is uncompressed and encoded as bytes, it will require 2MB of storage. Scale this up to a dataset of maybe 50,000 samples, use a slightly larger volume, and it is clear that this may begin to cause problems.

One solution might be to compress the volumes. However, compression introduces a significant computational overhead. It is a common trade off between processing power, available storage and I/O speeds. While compression of images has come a long way (even JPEG decoding is hardware accelerated on some platforms now), the same is not true for 3D volumes. Generic file compression options are of course available (Run Length Encoding, Huffman Coding and LZ77, just to name a few), but on large volumes being read many times per second, the CPU could easily become the bottleneck. In this work, we do not compress the volumes and instead use large SSDs to store our volumetric training data.

The format in which the volume is stored is another important consideration. There are many formats for storing volumetric data. Typically, these formats have been developed for the medical field for use with MRI, CT scans and the like. As such, these formats often include space for details such as patient information, along with a metadata describing the resolution and bit depth of the volumetric data. An argument can be made that none of this information is required if the volumetric representation is used as an intermediary state and doesn't have to be processed by *humans*. This applies to the work presented in this thesis. The volumetric representation used in the work described throughout this thesis is a contiguous array of bytes, which can be very efficiently read into memory.

Chapter 3

Literature Review

This section attempts to provide a comprehensive literature review of the works relevant to this thesis. This thesis focuses primarily on the 3D reconstruction of human faces, but includes an extension which shows our method working on full human bodies. Methods for 3D reconstruction of the face and body often depend on methods for extracting landmarks first. As such, an overview of literature relating to face alignment and human pose estimation is given in this literature review, providing some context for the works relating to 3D face and human body reconstruction.

3.1 Face Reconstruction

The literature on 3D face reconstruction typically falls into several categories. However, there is often also a lot of interaction or intersection between multiple techniques in an attempt to garner better results.

3.1.1 3D Morphable Models (3DMM)

Arguably the most popular approach to estimating the 3D geometry of the human face is using 3D Morphable Models (3DMM). A model s consist of a 3D mesh of N vertices, defined as a $3N$ vector:

$$s = [x_1, y_1, z_1, \ldots, x_N, y_N, z_N] \tag{3.1}$$

This model is constructed by finding the dense correspondence between all vertices. This ensures that any particular vertex (e.g. the tip of the nose) has the same index in all meshes. Additionally, it ensures that all meshes have the same number of vertices. These meshes must then be brought into a shape space, which ensures that the meshes are aligned *somewhere* against an unknown mean shape and is performed by applying Generalised Procrustes Analysis. Principal Component Analysis (PCA) is then performed which results in $\{\overline{\boldsymbol{s}}_{id}, \boldsymbol{U}_{id}, \Sigma_{id}\}$, where $\overline{\boldsymbol{s}}_{id}$ is a vector containing a mean shape, $\boldsymbol{U}_{id}$ is the first n principal components for each vertex, and hence has the shape $3N \times n$. Finally, Σ_{id} is a diagonal matrix containing the standard deviations of each principal component. Finally, there are the shape parameters $\boldsymbol{p} = [p_1, \ldots, p_n]$, which can be used to generate a new face $\mathcal{S}$ [12]:

$$\mathcal{S}(\boldsymbol{p}) = \overline{\boldsymbol{s}}_{id} + \boldsymbol{U}_{id}\Sigma_{id}\boldsymbol{p} \tag{3.2}$$

There are many methods based on this approach, including [52, 43, 120, 63, 104, 50, 49]. In particular, Jourabloo and Liu use a cascade of CNNs, which first regress the camera projection, followed by the pose, followed by shape and expression. Features are extracted from their model fitting which can assist in facial landmark localisation [52]. A method capable of working with models of different resolutions is proposed in [43], along with several models of different resolution, ranging from roughly 3500 to 30,000

vertices. This method does not use a CNN, instead regressing parameters for shape and pose using only the landmarks. This is done by iteratively minimising a cost function which evaluates the fitting.

Constraining the pose, shape and (optionally) appearance into a small number of parameters has the undesirable side effect of being unable to capture finer details, such as wrinkles and bumps. This is due to the small number of parameters used and can be seen in many methods, for example [86], where the 3DMM fitting stage is only used to extract a coarse shape. In [92], a CNN is used for 3DMM fitting, followed by a refinement step which iteratively improves the finer details. However, during this iterative refinement, the correspondence between vertices, as well as the scale and spatial location of the fitting is lost, which disregards one of the main advantages of 3DMM based approaches. Another iterative approach attempting to *also* recover both the 3D geometry and facial landmarks is described in [63]. For initialisation, the method passes the image and a mean shape from the facial landmarks. This mean shape is generally generated by first aligning with a method such as Procrustes, followed by taking the average location for each point. These landmarks are gradually refined while learning a 3D to 2D mapping, which updates the 3D face shape. Unlike our own method (described in Chapter 4), the method of Liu et al. does not estimate the facial pose, and so always produces a frontal reconstruction [63]. In many cases, the pose is an important component, and as such, a separate method, likely based on the facial landmarks, would have to be used to find the transformation for the 3D model.

A partially synthetic approach to 3D face reconstruction (indirectly for face alignment) is proposed in [120]. The method employs a single CNN that is iteratively applied to estimate the model parameters using as input the 2D image and 3D shape and pose representation from the previous iteration. The network consists of four

convolution layers and two fully connected layers. The first two convolutions share parameters for memory efficient feature extraction. This shape and pose representation can best be summarised as a depth map of the fitting generated by parameters from the previous iteration. The training data is augmented by artificially rotating faces between $\{-90, 90\}$ degrees using groundtruth parameters for a frontal fitting. One of the byproducts of this work is a large dataset consisting of large pose synthetically rotated faces, each with a corresponding 3D mesh.

3.1.2 Synthetic Training Data for 3DMM

The data required for training 3D reconstruction methods typically consists of an image with a corresponding 3D scan. Such pairs can be difficult to obtain. Richardson et al. alleviate this issue by generating synthetic training data, comprising of renderings of randomly generated 3DMM with the corresponding mesh [85]. Faces are rendered under random lighting conditions and pose. An iterative fitting method is then used during inference, optimising both geometry and lighting to ensure a good match with the given image. Synthetic data was also used for training a general purpose 3D reconstruction method [61]. Methods trained using synthetic training data often fail on real or "in-the-wild" images due to unrealistic renderings or constrained lighting.

3.1.3 Shape from Shading

A method which performs 3D face reconstruction from video is presented in [98], which can also work from a set of static images. The method combines their proposed 3D flow estimation algorithm with a Shape from Shading (SfS) method. The method first estimates the facial pose, which is used to rotate an average face shape to the correct pose. For each frame, 3D flow updates the mesh, which both refines the detail, and

updates aspects such as expression and pose. Another Shape from Shading approach is proposed in [50], which first aligns a coarse facial model using 2D landmarks. This coarse model is then *corrected* using the smaller eigenvalues of a Laplacian eigenfunction, which results in small adjustments to the coarse geometry. Finally, the shape from shading step refines the mesh and introduces finer details.

3.1.4 Multiview Reconstruction and Structure from Motion

Several methods exist which rely on multiple images to reconstruct the 3D geometry of the face. These again fall into two more categories. These are known as Multiview [72, 31] or Structure from Motion (SfM) [31, 29, 81] based methods, although again, there are some similarities between these methods.

A keypoint based approach is presented in [72], where Scale Invariant Feature Transform (SIFT) [69] is first used to extract dense features from each image. These are then aligned using an Iterative Closest Point (ICP) algorithm to build a point cloud of a face, from which a surface could be extracted later. However, due to the simplicity of this approach, and the limitations of ICP, this method will not account for changes in facial expression between images. The work of [31] proposes to first pass each image through a CNN to extract feature maps, which are then passed to a recurrent neural network (RNN) to regress parameters for a 3DMM. The result is altered with each frame, making it a suitable method for videos, where facial expression changes over time.

A two stage (coarse to fine) approach is presented in [29], where first a sparse set of landmarks are first estimated, which are then used to estimate pose to assist in fitting a dense set of points to the face to create a coarse reconstruction. During this coarse reconstruction step, 14 images from specific poses are extracted by rendering the coarse

reconstruction, which are used to refine the model and extract details. Finally, in [81], a method is proposed which reconstructs individual 3D shapes from multiple single images of a face. The quality of each shape is evaluated by measuring the distance between each rendering and corresponding image. The best of these shapes are them combined into a single 3D model. The aim of doing this is to better handle variations in pose, lighting, occlusions and facial expressions.

3.1.5 Depth Estimation

Depth estimation refers to the problem of estimating how far away each pixel of an image would be, either relative to the other objects in the image, or from the camera (i.e. real world space). Depth estimation is typically applied to problems such as room reconstruction, where only distances are required and not the full geometry of every object in the room. Despite this, there are a couple of methods which attempt to reconstruct the face by estimating the depth from an image [96, 97]. In [96], a constrained Independent Component Analysis (ICA) model is used to analyse multiple facial images of the same subject, from which the 3D geometry is obtained. An almost identical method is presented in [97], except instead of using an ICA model, a nonlinear Least-Squares model is used, offering a small performance improvement over [96].

3.1.6 Where does our work fit in?

In this section on 3D face reconstruction we have discussed several different general approaches to reconstructing the shape of a person's face. However, our method for 3D face reconstruction, described in Chapter 4 is very different to the aforementioned methods. First, our method does not rely on a 3D shape model for either the pre-processing or "fitting" procedure, instead we rely on a volumetric representation. This

avoids the requirement of estimating correspondence between vertices, which is a challenging problem in itself. Second, our method is direct, producing a volumetric representation of the facial geometry in just a single step. Third, unlike [104] and multiview approaches, we do not lose any pose or scale information, and show potential for detailed reconstruction, demonstrated in Chapter 5 on human bodies.

3.2 Human Body Reconstruction

Human body reconstruction has a wide variety of applications, from motion capture to trying on clothing in virtual environments. This problem involves estimating the geometry for the full body, sometimes with clothing, of all limbs and extremities. Many human reconstruction methods estimate the geometry from one or more images. For example, [5, 37, 38] fit a model based on a single RGB or grey scale image. In particular, [37] fit a skeleton model to the image by estimating the scale and pose of each body part separately. In [38], they fit a shape model initialised by a user clicking on separate body parts, assisted by a segmentation mask. Another shape model based approach is proposed in [5], using the SCRAPE model [1], which is fitted with a stochastic optimisation step. A general shape fitting method for reconstruction is proposed in [24], where two Gaussian models are used - one for shape and one for pose, by solving non-linear optimisation problems. The authors demonstrate this method on human bodies and sharks. In [48], a single image and its corresponding landmarks are used to lookup a similar human pose using a kd-tree, containing about 4 million examples. A method intended for multi-instance model fitting from a single image is described in [111].

Several methods aim to estimate the 3D geometry using only the landmarks extracted via human pose estimation [11, 84]. Particularly, SMPLify [11] (which uses

the SMPL model [66]), which was extended to also include further guidance from a segmentation mask in [107]. However, such an approach will never be able to capable of regressing finer details, unless information from the image is also captured.

Aside from SCRAPE [1] and SMPL [66], mentioned earlier, Dyna, the shape model capable of capturing large variations in body shape is presented in [82], but without an accompanying fitting method from a single image. A very recent shape model called Total Capture [51] captures many aspects of the body which are typically ignored by other shape models, including the face and hands.

The work we present in Chapter 5 is different from all of the aforementioned methods in that we do not regress parameters for a shape model, nor do we regress the vertices directly. Further more, our method skips the model generation step entirely, which avoids the need to find dense correspondence between all training examples. Instead, we constrain the problem to the spatial domain, and directly regress the 3D structure using spatial convolutions using a CNN, via a volumetric representation from which the full 3D geometry can be recovered.

3.3 Landmark Detection

Both face alignment and human pose estimation can be considered as a special cases of landmark detection. The work presented in this thesis optionally uses landmarks to guide our 3D reconstruction method, which offers a slight performance improvement, although really shows its effectiveness on difficult poses.

3.3.1 Face Alignment

Face alignment is the process of detecting a set of facial points to a facial image. Face alignment is a prerequisite to many 3D face reconstruction methods - in fact, methods such as [43] depend only on facial landmarks, discarding the image entirely.

Until recently, the best performing methods for face alignment were based on a cascaded regression approach. These methods estimate the location of facial landmarks by applying a sequence of regressors. The input to these regressors is typically SIFT [68] and other hand selected features. These regressors are applied in a cascaded fashion where the input to regressor k is the estimate from regressor $k-1$. More details of such approaches to face alignment can be found in [30, 32, 17, 109, 119, 106].

The remainder of this section will focus on deep learning based approaches to face alignment. Deep convolutional networks have proven to be an excellent choice for this kind of problem. One of the primary reasons for this is due to the way global and local context can be incorporated to provide both accuracy and precision.

Several approaches directly regress the locations of the landmarks using a tail of linear layers. For example, a CNN cascade is presented in [95] to directly produce the landmark locations. The output of this second stage is a set of 68 (x, y, z) landmarks. A multitask approach is proposed in [114], regressing the location of the facial landmarks while performing facial attribute classification.

Many CNN based alignment methods estimate the facial landmarks by regressing a 2D Gaussian located at each landmark location [14, 16, 71, 57, 74]. This spatial approach is commonly referred to as heatmap regression. One transitioning approach for 3D face alignment is described in [14], where 2D landmarks are regressed as heatmaps. These heatmaps are then passed through a second network, performing regression to obtain 3D x, y, z landmarks. This was extended in [16], where the second regressor is

replaced with a CNN producing a 3D heatmap representation. A similar heatmap based approach was used in a multi-stage approach described in [57], where the landmark detections are refined after each stage. Between each stage, the current landmark prediction and some generated features are passed again through the network. At the beginning of each stage, the inputs are transformed using the landmarks to make the problem less challenging.

An indirect approach is discussed in [120], where a 3DMM is first fitted, from which 3D landmarks are taken. This work is discussed in further detail in Section 3.1 as it is performing 3D reconstruction, but for the purposes of face alignment. A similar idea is proposed in [52], where a dense 3D model is fit to the face first.

3.3.2 Human Pose Estimation

Modern approaches to estimating human pose are based on methods employing CNNs. These methods generally fall into one of two categories. The first is to directly regress the coordinates of the joints using an L2 (or similar) loss, see for example [60, 77, 99, 100, 117, 22, 36, 103, 102]. In particular, Park et al. estimate the 3D pose by combining the 2D predictions with image features [77]. Tekin et al. employ an autoencoder to constrain pose to something plausible [99]. Similarly, Zhou et al. have the same goal but achieve this by using a kinematic model [117]. Synthetic training data is used by Chen et al. for the full training procedure, to ensure that the network is trained with accurate data [22]. However, Ghezelghieh et al. only augment their existing training set with synthetic data [36]. Toshev and Szegedy propose an iterative method, in which images are first passed through an initalisation CNN. The output from this is then combined again with the input image and passed through a second CNN multiple times iteratively [103]. Tompson et al. propose the use of a multi-resolution CNN,

trained jointly with a Markov Random Field, which helps to refine the prediction from the CNN by constraining pose [102]. An iterative method is presented by Carreira et al. [18]. However, unlike the method of Toshev and Szegedy, which directly outputs the locations of the joints after each iteration, this method outputs a displacement vector for each joint. The displacement is applied and the points are *rendered* to heatmaps, which fed into the network again for the next iteration. For the first iteration, a mean pose is provided to the network [18].

The second approach to CNN based human pose estimation is to regress heatmaps, as done in [75, 80, 118, 78, 73, 115]. In these methods, an intensity value is regressed for each pixel, giving a likelyhood of the joint in that pixel. Newell et al. present a novel "Stacked Hourglass" network is proposed [75]. This architecture in particular, despite being intended for human pose estimation, has strongly influenced our work on 3D reconstruction, which is discussed further in Chapters 4 and 5, as such the remainder of this paragraph will serve to explain the general idea of this work. Ignoring the stacking aspect for just a moment, the "hourglass" architecture, also called an Encoder-Decoder network, gradually reduces and increases the spatial resolution while combining features from different scales. Combining these features from different scales allows the network to produce better predictions, as both global (e.g. *What's the main object in this image?*) and local context (e.g. *Where is the boundary between these two objects?*) is available to the last few layers of the network. Stacking these hourglass modules together with a loss function between each one allows the network to refine and reevaluate the pose as it passes through the network. This is a similar idea to the iterative methods described in [103, 18]. In the case of [75], the loss function used for human pose estimation is Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2, \tag{3.3}$$

which as the name suggests takes the mean of the squared error between elements of Y and $\hat{Y}$. For their experiments, performance is measured for several configurations of stacking, 2, 4 or 8 hourglasses. In each case, reducing the number of residual modules to compensate for parameter count. The performance between each configuration after the last layer is fairly close, but the 8 hourglass configuration beats the 4 hourglass configuration, which also beats the two hourglass configuration.

A 3D heatmap is regressed in [78], which is a similar idea to our own 3D reconstruction, but for pose, not geometry. A part based heatmap regression method is presented in [73]. In [80], video is used instead of a single image. The method accepts an arbitrary number of neighbouring frames for each time t. Features between frames are combined using what they call "Spatial Fusion Layers". For each frame at time t, a pose heatmap. Optical flow is used from the neighbouring frames to time t to refine the prediction at. Finally, in [115], the authors present their method for *seeing through walls*, by using a standard CNN trained on 2D radio frequency heatmaps, for both horizontal and vertical polarizations.

Chapter 4

Large Pose 3D Face Reconstruction via Volumetric Regression

3D face reconstruction is a fundamental Computer Vision problem with many associated difficulties. Current systems often assume the availability of multiple facial images as input, sometimes from the same subject. Additionally, there are a number of methodological challenges involved, which must be addressed. These challenges include establishing a dense correspondence across large poses, expressions and non-uniform illumination. In general, these methods require complex and inefficient pipelines for model building and fitting. In this work, we propose to address many of these

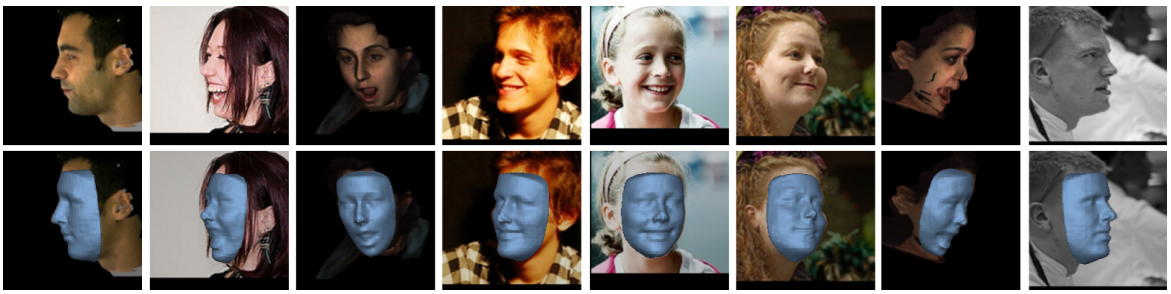


Figure 4.1 A preview of our 3D reconstruction method, applied on a wide variety of poses.

limitations by constraining the problem of 3D face reconstruction to the spatial domain. We do this by training a Convolutional Neural Network (CNN), on an appropriate dataset, consisting of 2D images and 3D facial models, or scans. Our CNN works with just a single 2D facial image and does not require accurate alignment. Our method works for arbitrary facial poses and expressions, and can be used to reconstruct the whole 3D facial geometry, including non-visible parts. Our method bypasses the construction (during training) and fitting (during inference) of a 3D Morphable Model. We achieve this via a simple CNN architecture that performs direct regression of a volumetric representation of the 3D facial geometry from a single 2D image. We also demonstrate how the related task of facial landmark localisation can be incorporated into the proposed framework and help improve reconstruction quality, especially in the cases of large poses and facial expressions.

4.1 Introduction

3D face reconstruction is the problem of recovering the 3D facial geometry from 2D images. Despite many years of research, it is still an open problem in Computer Vision and Graphics research. Depending on the setting and the assumptions made, there are many variations of it, as well as a multitude of approaches to solve it. This work is on 3D face reconstruction using only a single image. Under this setting, the problem is considered far from being solved. In this chapter, we propose to approach it, for the first time to the best of our knowledge, by directly learning a mapping from pixels to 3D coordinates using a Convolutional Neural Network (CNN). Besides its simplicity, our approach works with totally unconstrained images downloaded from the web, including facial images of arbitrary poses, facial expressions and occlusions, as shown in Figure 4.1.

Motivation. No matter what the underlying assumptions are, what the input(s) and output(s) to the algorithm are, 3D face reconstruction requires in general complex pipelines and solving non-convex difficult optimization problems for both model building (during training) and model fitting (during testing). In the following paragraphs, we provide examples from 5 predominant approaches:

1. In the 3D Morphable Model (3DMM) [9, 88], the most popular approach for estimating the full 3D facial structure from a single image, training includes an iterative flow procedure for dense image correspondence which is prone to failure. Additionally, testing requires a careful initialisation for solving a difficult highly non-convex optimization problem in order to estimate the pose and shape parameters, which is slow.

2. The work of [54], a popular approach for 2.5D reconstruction from a single image, formulates and solves a carefully initialised (for frontal images only) non-convex optimization problem for recovering the lighting, depth, and albedo in an alternating manner where each of the sub-problems is a difficult optimization problem per se.

3. In [55], a recent approach for creating a neutral subject-specific 2.5D model from a near frontal image, an iterative procedure is proposed, which entails localising facial landmarks, face frontalization, solving a photometric stereo problem, local surface normal estimation, and finally shape integration.

4. In [98], a state-of-the-art pipeline for reconstructing a highly detailed 2.5D facial shape for each video frame, an average shape and an illumination subspace for the specific person is firstly computed (offline), while testing is an iterative process requiring a sophisticated pose estimation algorithm, 3D flow computation

between the model and the video frame, and finally shape refinement by solving a shape-from-shading optimization problem.

5. More recently, the state-of-the-art method of [89] that produces the average (neutral) 3D face from a collection of personal photos, firstly performs landmark detection, then fits a 3DMM using a sparse set of points, then solves an optimization problem similar to the one in [55], then performs surface normal estimation as in [55] and finally performs surface reconstruction by solving another energy minimisation problem.

Simplifying the technical challenges involved in the aforementioned works is the main motivation of this work.

4.1.1 Main contributions

We describe a very simple approach which bypasses many of the difficulties encountered in 3D face reconstruction by using a novel volumetric representation of the 3D facial geometry, and an appropriate CNN architecture that is trained to regress directly from a 2D facial image to the corresponding 3D volume. An overview of our method is shown in Figure 4.4. In summary, our contributions are:

- Given a dataset consisting of 2D images and 3D face scans, we investigate whether a CNN can learn directly, in an end-to-end fashion, the mapping from image pixels to the full 3D facial structure geometry (including the non-visible facial parts). Indeed, we show that the answer to this question is positive.

- We demonstrate that our CNN works with just a single 2D facial image, does not require accurate alignment nor establishes dense correspondence between images, works for arbitrary facial poses and expressions, and can be used to reconstruct

the whole 3D facial geometry bypassing the construction (during training) and fitting (during testing) of a 3DMM.

- We achieve this via a simple CNN architecture that performs *direct* regression of a volumetric representation of the 3D facial geometry from a single 2D image. 3DMM fitting is not used. Our method uses only 2D images as input to the proposed CNN architecture.

- We show how the related task of 3D facial landmark localisation can be incorporated into the proposed framework and help improve reconstruction quality, especially for the cases of large poses and facial expressions.

- We report results for a large number of experiments on both controlled and completely unconstrained images from the web, illustrating that our method outperforms prior work on single image 3D face reconstruction by a large margin.

4.2 Method

This section describes our framework for 3D reconstruction, including the proposed data representation used.

This section begins by describing the training data used for our 3D reconstruction approach. This is followed by the proposed volumetric representation, and the 3D reconstruction method itself.

4.2.1 Training Datasets

Our aim is to regress the full 3D facial structure from a 2D image. To this end, our method requires an appropriate dataset consisting of 2D images and 3D facial scans.

As our target is to apply the method on completely unconstrained images from the web, we chose the dataset of [120] for forming our training and test sets. The dataset has been produced by fitting a 3DMM built from the combination of the Basel [79] and FaceWarehouse [17] models to the unconstrained images of the 300W dataset [91] using the multi-feature fitting approach of [88], careful initialisation and by constraining the solution using a sparse set of landmarks. Face profiling is then used to render each image to 10-15 different poses resulting in a large scale dataset (more than 60,000 2D facial images and 3D meshes) called 300W-LP. Each mesh consists of approximately 53000 vertices, enclosing the face entirely. However, no vertices exist around the mouth, base of neck and eyes, and as such, this mesh is not watertight. All images are 450×450, although these are downsampled to 386×386 during training. Note that because each mesh is produced by a 3DMM, the vertices of all produced meshes are in dense correspondence; however this is not a prerequisite for our method and unregistered raw facial scans could be also used if available (e.g. the BU-4DFE dataset [113]).

4.2.2 Proposed volumetric representation

Our goal is to predict the coordinates of the 3D vertices of each facial scan from the corresponding 2D image via CNN regression. As a number of works have pointed out (see for example [101, 80]), direct regression of all 3D points concatenated as a vector using the standard L2 loss might cause difficulties in learning because a single correct value for each 3D vertex must be predicted. Additionally, such an approach requires interpolating all scans to a vector of a fixed dimension, a pre-processing step not required by our method. Note that similar learning problems are encountered when a CNN is used to regress model parameters like the 3DMM parameters rather than the actual vertices. In this case, special care must be taken to weight parameters appropriately using the Mahalanobis distance or in general some normalisation method,

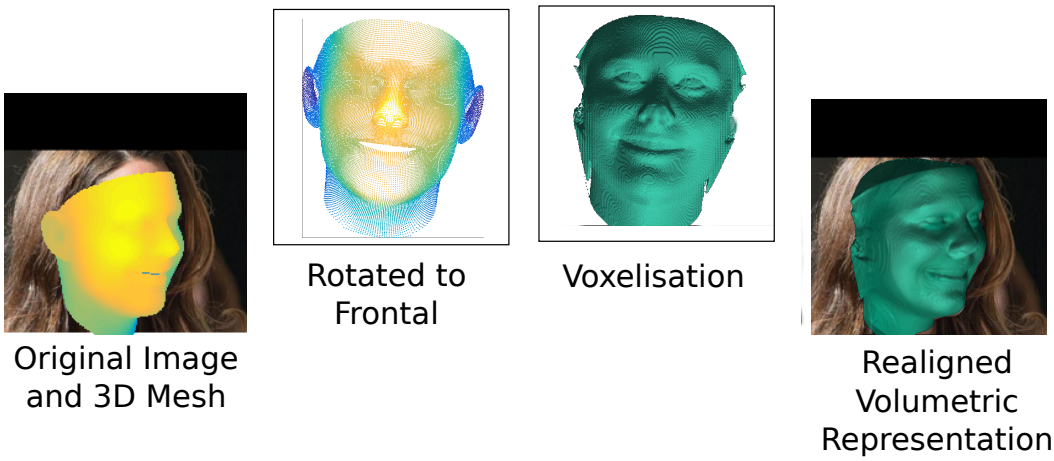


Figure 4.2 The voxelisation process creates a volumetric representation of the 3D face mesh, aligned with the 2D image.

see for example [120]. We compare the performance of our method with that of a similar method [120] in Section 4.3.

To alleviate the aforementioned learning problem, we propose to reformulate the problem of 3D face reconstruction as one of 2D to 3D image segmentation: in particular, we convert each 3D facial scan into a 3D binary volume $\mathbf{V}_{whd}$ by discretizing the 3D space into voxels $\{w, h, d\}$, assigning a value of 1 to all points enclosed by the 3D facial scan, and 0 otherwise. That is to say V_{whd} is the ground truth for voxel $\{w, h, d\}$ and is equal to 1, if voxel $\{w, h, d\}$ belongs to the 3D volumetric representation of the face and 0 otherwise (i.e. it belongs to the background). The conversion is shown in Figure 4.2. Contrary to standard approaches, this data was voxelised by rotating to frontal and estimating a depth map using bilinear interpolation, from which the volume is filled back to a fixed distance from the front of the face. This results in the mouth becoming filled. Notice that the process creates a volume fully aligned with the 2D image. The importance of spatial alignment is analysed in more detail in Section 4.4. The error caused by discretization for a randomly picked facial scan as a function of the volume size is shown in Figure 4.3. Given that the error of state-of-the-art methods [89, 63] is of the order of a few mms, we conclude that discretization by $192 \times 192 \times 200$

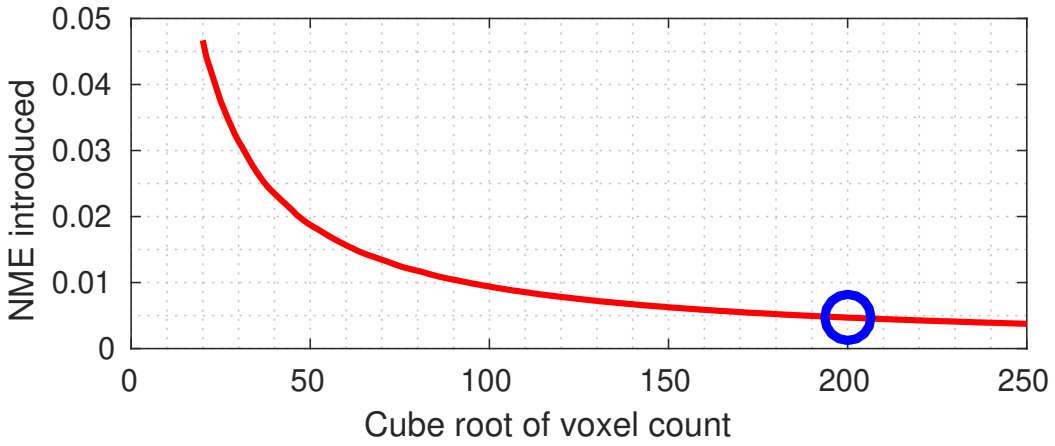


Figure 4.3 The error introduced due to voxelisation, shown as a function of volume density.

produces negligible error. Such a spatial resolution is ideal for the type of architecture we have chosen to use as it can be reduced by half many times. The depth, 200, was chosen to capture as much depth information as possible without sacrificing the spatial resolution or increasing the memory requirements too much.

Given our volumetric facial representation, the problem of regressing the 3D coordinates of all vertices of a facial scan is reduced to one of 3D binary volume segmentation. We approach this problem using recent CNN architectures from semantic image segmentation [65] and their extensions [75], as described in the next subsection.

4.2.3 Volumetric Regression Networks

In this section, we describe the proposed volumetric regression network, exploring several architectural variations described in detail in the following subsections:

Volumetric Regression Network (VRN). We wish to learn a mapping from the 2D facial image to its corresponding 3D volume $f : \mathbf{I} \rightarrow \mathbf{V}$. Given the training set of 2D images and constructed volumes, we learn this mapping using a CNN. Our CNN architecture for 3D segmentation is based on the "hourglass network" of [75] an

extension of the fully convolutional network of [65] using skip connections and residual learning [39]. Our volumetric architecture consists of two hourglass modules which are stacked together without intermediate supervision. The input is an RGB image and the output is a volume of $192 \times 192 \times 200$ of real values. This architecture is shown in Figure 4.4a. As it can be observed, the network has an encoding/decoding structure where a set of convolutional layers are firstly used to compute a feature representation of fixed dimension. This representation is further processed back to the spatial domain, re-establishing spatial correspondence between the input image and the output volume. Features are hierarchically combined from different resolutions to make per-pixel predictions. The second hourglass is used to refine this output, and has an identical structure to that of the first one.

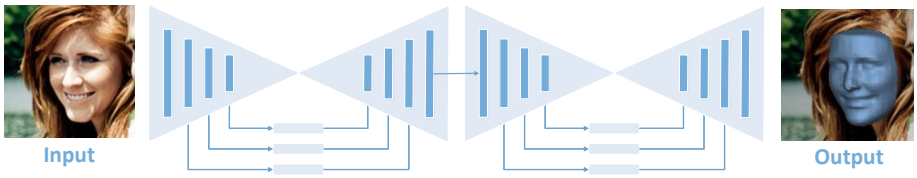
We train our volumetric regression network using the sigmoid cross entropy loss function:

$$l_1 = \sum_{w=1}^{W} \sum_{h=1}^{H} \sum_{d=1}^{D} [V_{whd} \log \widehat{V}_{whd} + (1 - V_{whd}) \log(1 - \widehat{V}_{whd})], \tag{4.1}$$

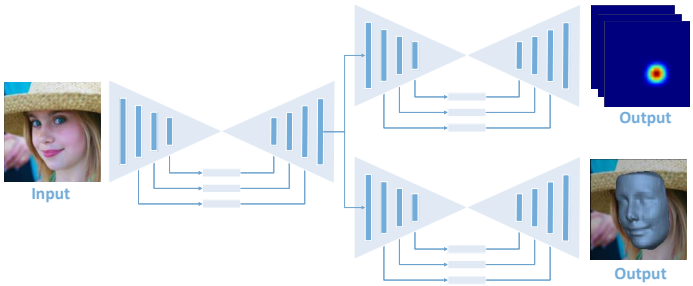
where $\widehat{V}_{whd}$ is the corresponding Sigmoid output at voxel $\{w, h, d\}$ of the regressed volume. In the case of *VRN - Multitask*, the same loss is applied to both tails of the network with even weightings. Similarly, in *VRN - Guided*, the loss is also applied to the heatmap regression and hence the loss is summed during backpropagation.

Each residual module of all three of our architectures consists of 256 feature spatial convolutions.

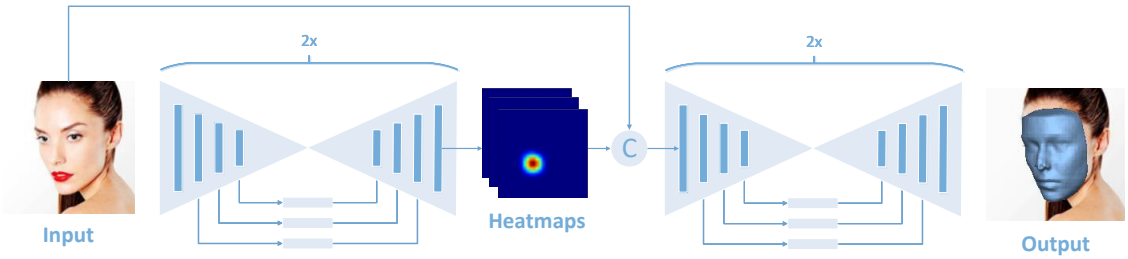
At test time, and given an input 2D image, the network regresses a 3D volume from which the outer 3D facial mesh is recovered. Rather than making hard (binary) predictions at pixel level, we found that the soft sigmoid output is more useful for further processing. Both representations are shown in Figure 4.5 where clearly the latter results in smoother results. Finally, from the 3D volume, a mesh can be formed



(a) The proposed *Volumetric Regression Network (VRN)* accepts as input an RGB input and directly regresses a 3D volume completely bypassing the fitting of a 3DMM. Each rectangle is a residual module of 256 features.



(b) The proposed *VRN - Multitask* architecture regresses both the 3D facial volume and a set of sparse facial landmarks.



(c) The proposed *VRN - Guided* architecture firsts detects the 2D projection of the 3D landmarks, and stacks these with the original image. This stack is fed into the reconstruction network, which directly regresses the volume.

Figure 4.4 An overview of the proposed three architectures for Volumetric Regression: *Volumetric Regression Network (VRN)*, *VRN - Multitask* and *VRN - Guided*.

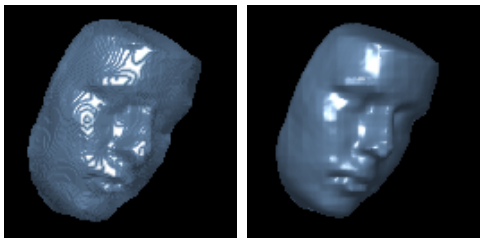


Figure 4.5 Comparison between making hard (binary) vs soft (real) predictions. The latter produces a smoother result.

by generating the iso-surface of the volume. If needed, correspondence between this variable length mesh and a fixed mesh can be found using Iterative Closest Point (ICP) [8].

VRN - Multitask. We also propose a Multitask VRN, shown in Figure 4.4b, consisting of three hourglass modules. The first hourglass provides features to a fork of two hourglasses. The first of this fork regresses the 68 iBUG landmarks [91] as 2D Gaussians, each on a separate channel. The second hourglass of this fork directly regresses the 3D structure of the face as a volume, as in the aforementioned unguided volumetric regression method. The goal of this multitask network is to learn more reliable features which are better suited to the two tasks.

VRN - Guided. We argue that reconstruction should benefit from firstly performing a simpler face analysis task; in particular we propose an architecture for volumetric regression guided by facial landmarks. To this end, we train a stacked hourglass network which accepts guidance from landmarks during training and inference. This network has a similar architecture to the unguided volumetric regression method, however the input to this architecture is an RGB image stacked with 68 channels, each containing a Gaussian ($\sigma = 1$, approximate diameter of 6 pixels) centred on each of the 68 landmarks. This stacked representation and architecture is demonstrated in Figure 4.4c. During training we used the ground truth landmarks while during testing

we used a stacked hourglass network trained for facial landmark localisation. We call this network *VRN - Guided*.

4.2.4 Training

Each of our architectures was trained end-to-end using RMSProp with an initial learning rate of 10^{-4}, which was lowered after 40 epochs to 10^{-5}. This training protocol remained fixed for all architectures to avoid bias. The number of epochs were chosen based on the rate at which the learning rate fell during training of *VRN* and some tests using the testing set of the HELEN [59] images from 300W [91] as validation. During training, random augmentation was applied to each input sample (face image) and its corresponding target (3D volume): we applied in-plane rotation $r \in [-45°, ..., 45°]$, translation $t_z, t_y \in [-15, ..., 15]$ and scale $s \in [0.85, ..., 1.15]$ jitter. In 20% of cases, the input and target were flipped horizontally. Finally, the input samples were adjusted with some colour scaling on each RGB channel selected uniformly $c_r, c_g, c_b \in \{0.6, ..., 1.4\}$.

In the case of the *VRN - Guided*, the landmark detection module was trained to regress Gaussians with standard deviation of approximately 3 pixels ($\sigma = 1$). We noticed near negligible reduction in 3D reconstruction performance when training with different standard deviations.

4.3 Results

4.3.1 Evaluation Datasets

We used three different datasets for quantifying the performance of our method. The dataset AFLW2000-3D allows for unbiased and fair comparison with other methods.

Table 4.1 Reconstruction accuracy on AFLW2000-3D, BU-4DFE and Florence in terms of NME. Lower is better.

Method	**AFLW2000-3D**	**BU-4DFE**	**Florence**
VRN	0.0676	0.0600	0.0568
VRN - Multitask	0.0698	0.0625	0.0542
VRN - Guided	**0.0637**	**0.0555**	**0.0509**
3DDFA [120]	0.1012	0.1227	0.0975
EOS [43]	0.0971	0.1560	0.1253

The remaining two, BU-4DFE and Florence, have to be rendered, but allows us to quantify performance across pose and expression with greater accuracy. These three datasets are discussed in more detail below.

(a) AFLW2000-3D As our target was to test our network on totally unconstrained images, we firstly conducted experiments on the AFLW2000-3D [120] dataset which contains 3D facial meshes for the first 2000 images from AFLW [56].

(b) BU-4DFE We also conducted experiments on rendered images from BU-4DFE [113]. We rendered each participant for both Happy and Surprised expressions with three different pitch rotations between -20 and 20 degrees. For each pitch, seven roll rotations from -80 to 80 degrees were also rendered. Large variations in lighting direction and colour were added randomly to make the images more challenging. We show some example renderings in the first row (a) of Figure 4.6.

(c) Florence Finally, we conducted experiments on rendered images from the Florence [4] dataset. Facial images were rendered in a similar fashion to the ones of BU-4DFE but for slightly different parameters: Each face is rendered in 20 difference poses, using a pitch of -15, 20 or 25 degrees and each of the five evenly spaced rotations between -80 and 80. We show some example renderings in the second (b) row of Figure 4.6.



Figure 4.6 Example renderings from the BU-4DFE **(a)** and Florence **(b)**.

4.3.2 Evaluation metric

To measure the accuracy of reconstruction for each face, we used the Normalised Mean Error (NME) defined as the average per vertex Euclidean distance between the estimated and ground truth reconstruction normalised by the outer 3D interocular distance:

$$\text{NME} = \frac{1}{N} \sum_{k=1}^{N} \frac{||\mathbf{x}_k - \mathbf{y}_k||_2}{d}, \tag{4.2}$$

where N is the number of vertices per facial mesh, d is the 3D outer interocular distance and $\mathbf{x}_k$,$\mathbf{y}_k$ are vertices of the grouthtruth and predicted meshes. This outer interocular distance is the distance between the two lateral canthus, which is the most outer part of the eye. As this is in 3D, the distance remains fixed even when the face is rotated. The error is calculated on the face region only on approximately 19,000 vertices per facial mesh. Notice that when there is no point correspondence between the ground truth and the estimated mesh, ICP was used but only to establish the correspondence, i.e. the rigid alignment was not used. If the rigid alignment is used,

we found that, for all methods, the error decreases but it turns out that the relative difference in performance remains the same. For completeness, we included these results in Section 4.3.5.

4.3.3 Evaluation

We performed cross-database experiments only, on 3 different databases, namely AFLW2000-3D [120], BU-4DFE [113], and Florence [4] reporting the performance of all the proposed networks (*VRN*, *VRN - Multitask* and *VRN - Guided*) along with the performance of two state-of-the-art methods, namely 3DDFA [120] and EOS [43]. Both methods perform 3DMM fitting (3DDFA uses a CNN), a process completely bypassed by *VRN*.

Our results can be found in Table 4.1 and Figures 4.7, 4.8 and 4.9. Visual results of the proposed *VRN - Guided* on some very challenging images from AFLW2000-3D can be seen in Figure 4.10. We show some failure cases in Figure 4.11 where the images are very different to what has been seen during training time. Finally, we show some example results from some frames in the 300-VW [90] video dataset in Figure 4.12. From all of these results, we can conclude the following:

1. Volumetric Regression Networks largely outperform 3DDFA [120] and EOS [43] on all datasets used for evaluation, statistically significant in all cases ($p < 0.01$, paired t-test). This verifies that directly regressing the 3D facial structure using the volumetric representation, is a much easier problem for CNN learning.

2. All VRNs perform well across the whole spectrum of facial poses, expressions and occlusions. Also, there are no significant performance discrepancies across different datasets (ALFW2000-3D seems to be slightly more difficult).

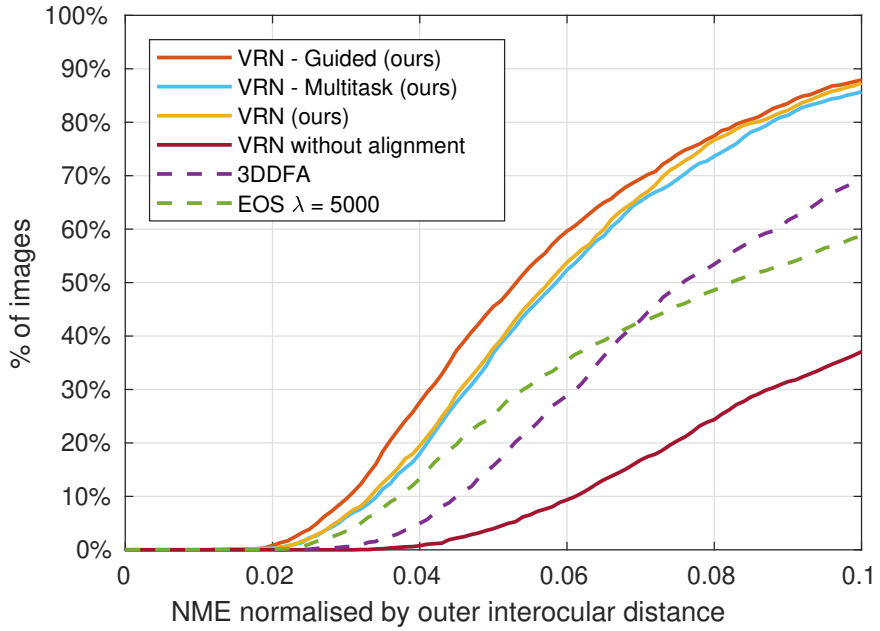


Figure 4.7 NME-based (Equation 4.3.2) performance on in-the-wild ALFW2000-3D dataset. The proposed *Volumetric Regression Networks*, and EOS and 3DDFA are compared. Please note that "VRN - without alignment" refers to a VRN where a frontal face is always regressed, regardless of input pose, discussed in Section 4.4.

3. The best performing VRN is the one guided by detected landmarks (*VRN - Guided*), however at the cost of higher computational complexity: *VRN - Guided* uses another stacked hourglass network for landmark localization.

4. *VRN - Multitask* does not always perform better than the plain VRN (in fact on BU-4DFE it performs worse), not justifying the increase of network complexity. The results are not significant on any dataset ($p > 0.10$, paired t-test). It seems that it might be preferable to train a network to focus on the task in hand.

Details about our experiments are given in the proceeding subsections.

4.3.3.1 Comparison with state-of-the-art

We compared against state-of-the-art 3D reconstruction methods for which code is publicly available. These include the very recent methods of 3DDFA [120], and EOS [43].

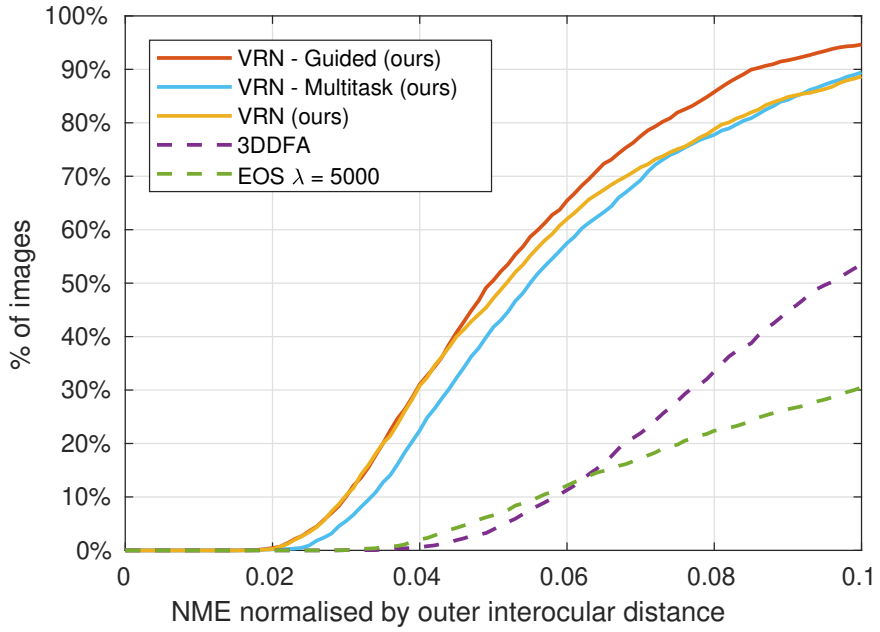


Figure 4.8 NME-based (Equation 4.3.2) performance on renderings from BU-4DFE. The proposed *Volumetric Regression Networks*, and EOS and 3DDFA are compared.

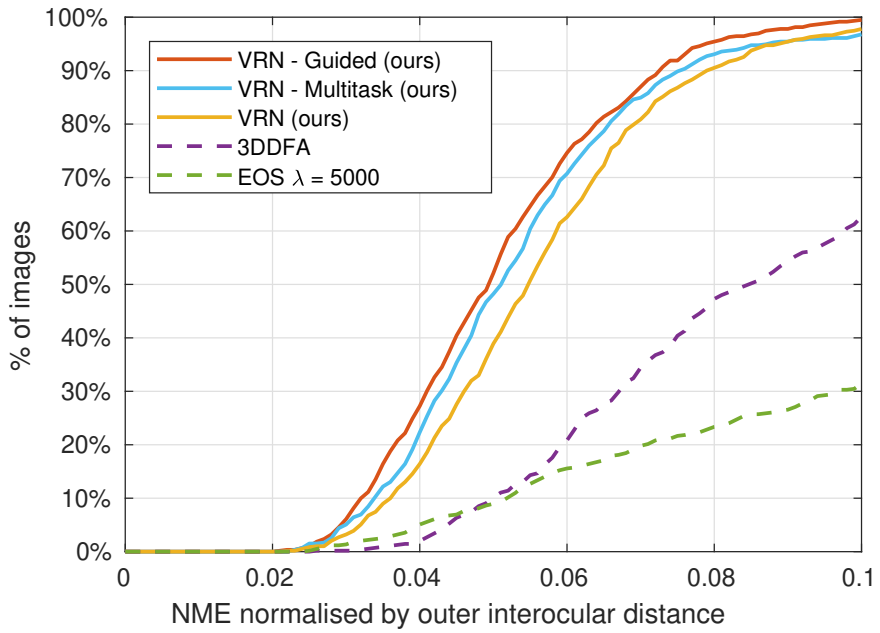


Figure 4.9 NME-based (Equation 4.3.2) performance on our large pose renderings of the Florence dataset. The proposed *Volumetric Regression Networks*, and EOS and 3DDFA are compared.

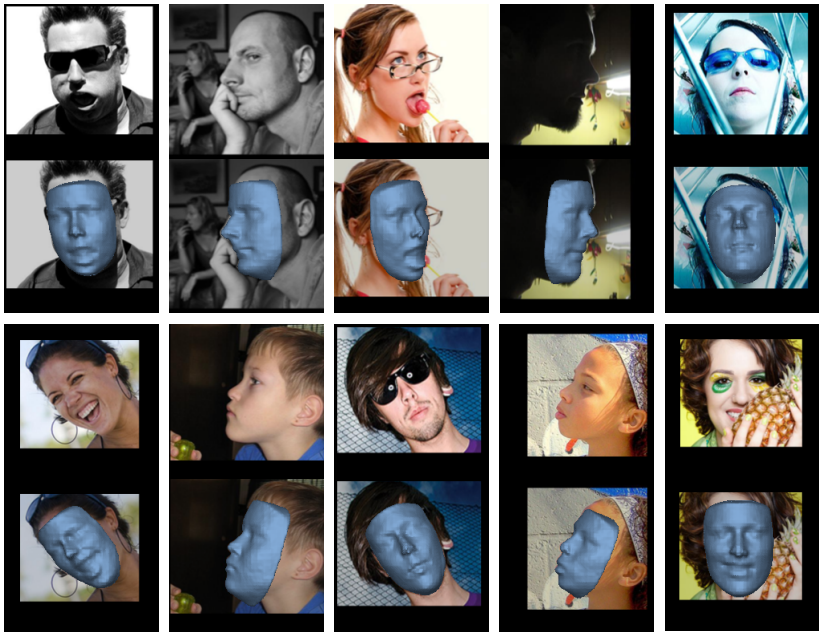


Figure 4.10 Some visual results from the AFLW2000-3D dataset generated using our *VRN - Guided* method.

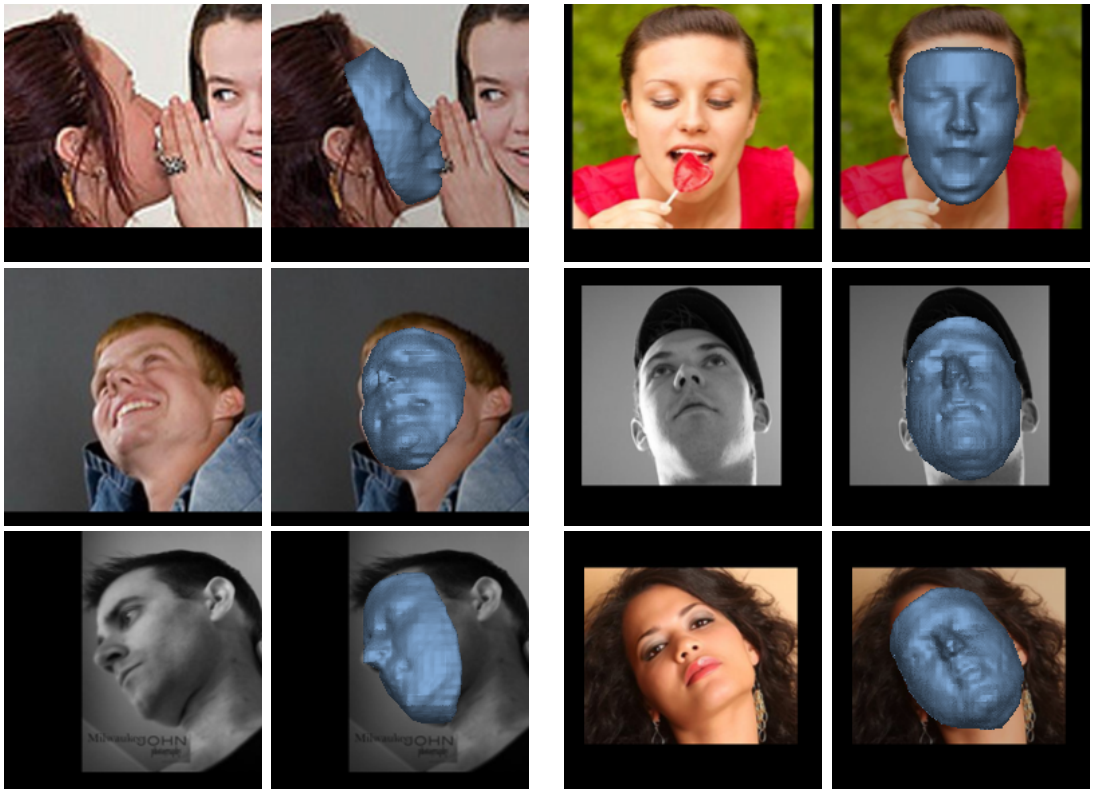


Figure 4.11 Some failure cases on AFLW2000 from our *VRN - Guided* network. In general, these images are poses which are not seen in the training set.



Figure 4.12 Some example results from frames of the 300VW dataset.

For EOS we used a large regularisation parameter $\lambda = 5000$ which we found to offer the best performance for most images. The method uses 2D landmarks as input, so for the sake of a fair comparison a stacked hourglass for 2D landmark detection was trained for this purpose. Our tests were performed using version 0.12 of EOS.

4.3.4 Training and Run Times

Due to the size of the volumetric representation, CPU and RAM can rapidly become a bottleneck during the training procedure. The voxelisation procedure took approximately 3 days using 200 parallel instances. This was carried out on the University of Nottingham's HPC service. Training takes a further 3 to 5 days. During inference, the face should ideally be well centred and of a similar scale to that used during training. Deep learning based face alignment methods take on the order of $100 - 200$ms. Our

Table 4.2 Reconstruction accuracy on AFLW2000-3D, BU4DFE and Florence in terms of NME (Equation 4.3.2) where ICP has been used to remove the rigid transformation. Lower is better.

Method	**AFLW2000**	**BU4DFE**	**Florence**
VRN	0.0605	0.0514	0.0470
VRN - Multitask	0.0625	0.0533	0.0439
VRN - Guided	**0.0543**	**0.0471**	**0.0429**
3DDFA [120]	0.1012	0.1144	0.0784
EOS [43]	0.0890	0.1456	0.1200

volumetric regression method takes approximately 200ms on an NVIDIA Titan X (Pascal). A final step of surface extraction is then needed to generate the mesh from the volumetric output. A fast implementation (such as the one implemented in MATLAB) takes approximately 100ms.

4.3.5 Results with ICP Alignment

We present results where ICP has been used not only to find the correspondence between the groundtruth and predicted vertices, but also to remove the rigid transformation between them. ICP was applied after the surface extraction was used to create the mesh. We find that this offers a marginal improvement to all methods. However, the relative performance remains mostly the same between each method. Results on AFLW2000 [120], BU4DFE [113] and Florence [4] can be seen in Figs. 4.13, 4.14 and 4.15 respectively. Numeric results can be found in Table 4.2.

4.4 Importance of spatial alignment

Our approach to 3D face reconstruction preserves the spatial alignment between input and output. That is to say that orthographic projection the regressed 3D geometry should be capable of segmenting the input face from the background. A 3D

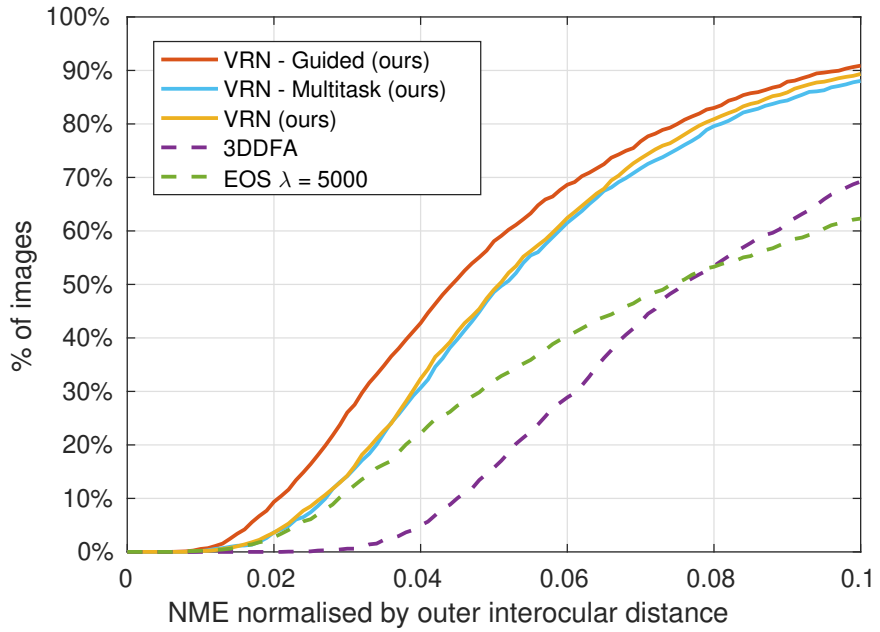


Figure 4.13 NME-based (Equation 4.3.2) performance on the in-the-wild AFLW2000-3D dataset, where ICP has been used to remove the rigid transformation. The proposed *Volumetric Regression Networks*, and EOS and 3DDFA are compared.

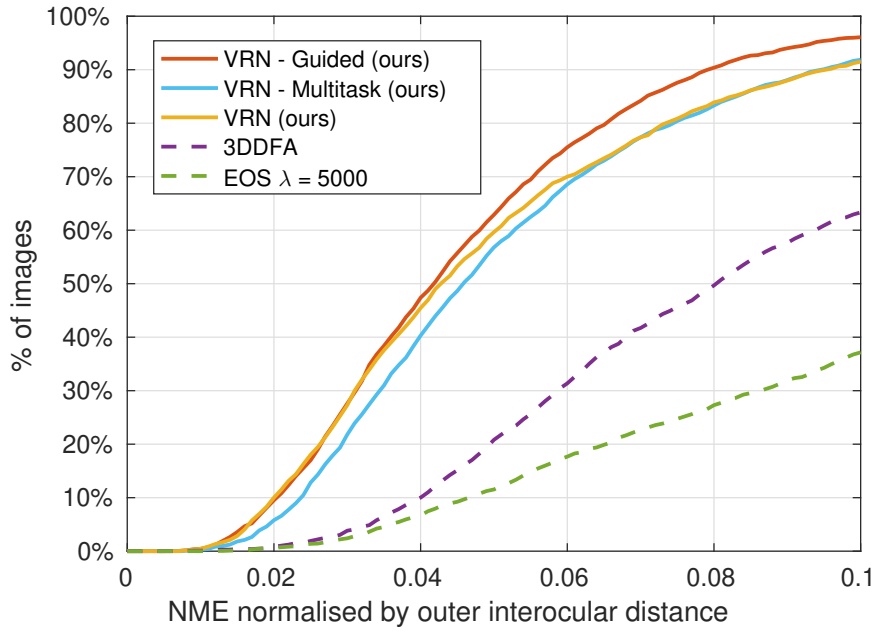


Figure 4.14 NME-based (Equation 4.3.2) performance on our large pose and expression renderings of the BU4DFE dataset, where ICP has been used to remove the rigid transformation. The proposed *Volumetric Regression Networks*, and EOS and 3DDFA are compared.

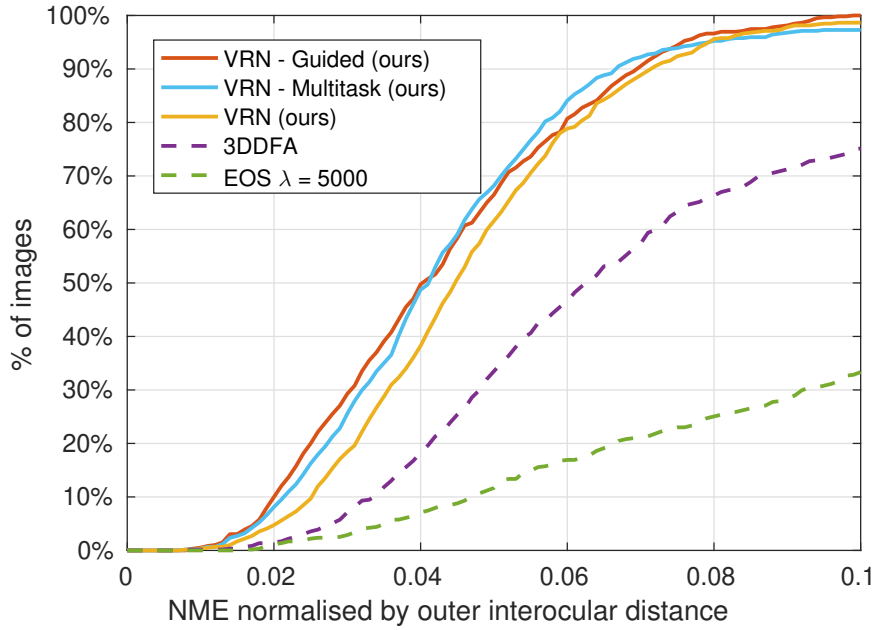


Figure 4.15 NME-based (Equation 4.3.2) performance on our large pose renderings of the Florence dataset, where ICP has been used to remove the rigid transformation. The proposed *Volumetric Regression Networks*, and EOS and 3DDFA are compared.

reconstruction method is described in [26] which accepts as input one or more images, and as output, produces a small 3D volume of $32 \times 32 \times 32$ using an LSTM [41]. In an attempt to explore what the repercussions of ignoring spatial alignment are, we trained a variant of *VRN* which regresses a frontal version of the face, i.e. a face of fixed orientation, regardless of input pose, as in [26]. We call this fixed-pose variant of our method *VRN - without alignment*. We also attempted to train a network using the code from [26] on downsampled versions of our volumes. Unfortunately we were unable to get the network to learn anything and found that the loss would not reduce.

Although *VRN - without alignment* produces a reasonable face, it captures only a diminished expression. The shape for all faces appears to remain almost identical across all images. This is very noticeable in Figure 4.16. Numeric comparison is shown in Figure 4.7, as *VRN - without alignment*. We believe that this further

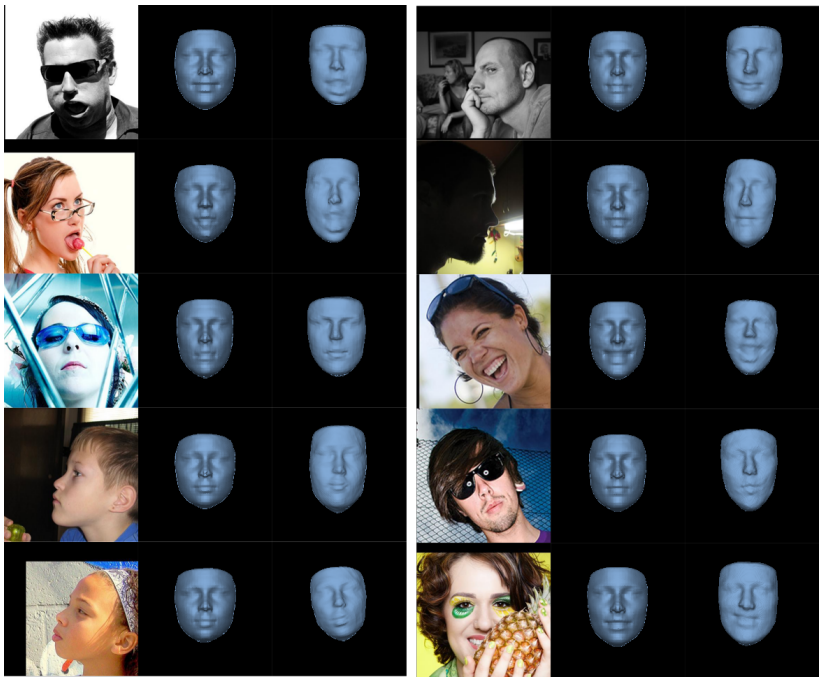


Figure 4.16 Result from *VRN without alignment* (second columns), and a frontalised output from *VRN - Guided* (third columns).

confirms that **spatial alignment is of paramount importance** when performing 3D reconstruction in this way.

4.5 Ablation studies

In this section, we report the results of experiments aiming to shed further light into the performance of the proposed networks. For all experiments reported, we used the best performing *VRN - Guided*.

4.5.1 Effect of pose

To measure the influence of pose on the reconstruction error, we measured the NME (Equation 4.3.2) for different amounts of yaw using all of our Florence [4] renderings. These were rendered at $\{-80, -40, 0, 40, 80\}$ degrees of yaw for each of $\{-5, 0, 5\}$ degrees

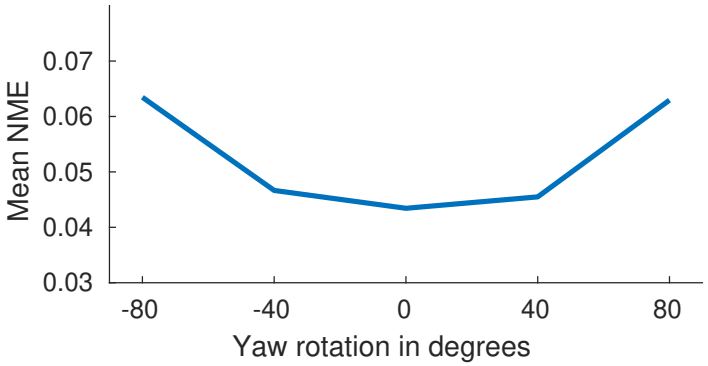


Figure 4.17 The effect of pose on reconstruction accuracy in terms of NME (Equation 4.3.2) on the Florence dataset. The *VRN - Guided* network was used.

of pitch, for all 50 subjects in Florence. As shown in Figure 4.17, the performance of our method decreases as the pose increases. This is to be expected, due to less of the face being visible which makes evaluation for the invisible part difficult. We believe that our error is still very low considering the magnitude of these poses.

4.5.2 Effect of expression

Certain expressions are usually considered harder to accurately reproduce in 3D face reconstruction. To measure the effect of facial expressions on performance, we rendered frontal images in difference expressions from BU-4DFE and measured the performance for each expression. We show the six acted facial expressions which are available in BU-4DFE in Figure 4.18. These expressions are available for all participants, of which there are 102. This kind of extreme acted facial expressions generally do not occur in the training set, yet as shown in Figure 4.19, the performance variation across different expressions is quite minor.



Figure 4.18 The six facial expressions available in the BU-4DFE dataset [113], rendered for the same participant.

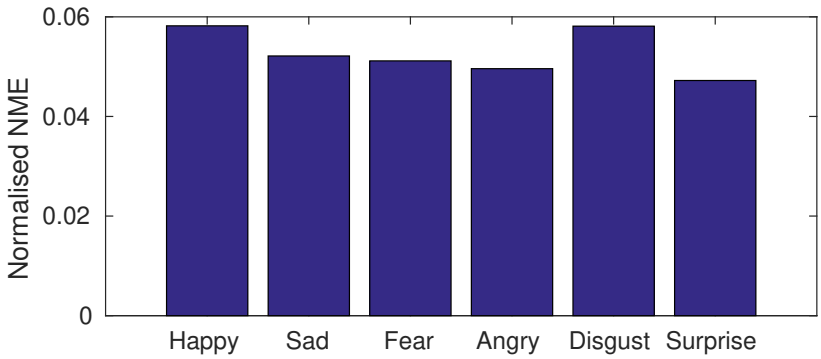


Figure 4.19 The effect of facial expression on reconstruction accuracy in terms of NME (Equation 4.3.2) on the BU-4DFE dataset. The *VRN - Guided* network was used.

Table 4.3 An overview of the architectural experiments conducted. **Feats** refers to the number of features per convolution layer. Similarly, **VFeats** refers to the number of volumetric features per volumetric convolution (in most cases this has no value as the majority of the networks used in our experiments are 2D only). **Mod** refers to the number of Residual modules used sequentially in any one place of the hourglass. **Stack** refers to the number of hourglass networks in a single stack. The right most column points to the relevant section of the chapter. We also show the mean Normalised Mean Error (NME, Equation4.3.2) for all images of the AFLW2000 dataset in column **nNME**.

Name	**Feats**	**VFeats**	**Mod**	**Stack**	**Para**	**mNME**	**Sec.**
VRN - Guided	256	—	4	2	2.5×10^7	0.0637	
Intermediate Supervis.	256	—	4	2	2.5×10^7	0.0630	4.6.1
Hourglass	256	—	8	1	2.5×10^7	0.0621	
Three Hourglass	256	—	3	3	2.8×10^7	0.0632	4.6.2
Four Hourglass	256	—	2	4	2.7×10^7	0.0639	
Hourglass Heavy	384	—	8	1	5.3×10^7	0.0627	
Hourglass Light	128	—	8	1	6.3×10^6	0.0633	4.6.3
Hourglass Lighter	64	—	8	1	1.8×10^6	0.0661	
Volumetric VRN	128	32	2	2	7.7×10^5	0.0697	4.6.4

4.5.3 Effect of Gaussian size for guidance

We trained a *VRN - Guided*, however, this time, the facial landmark detector network of the *VRN - Guided* regresses larger Gaussians ($\sigma = 2$ as opposed to the normal $\sigma = 1$). The performance of the 3D reconstruction dropped by a negligible amount, suggesting that as long as the Gaussians are of a sensible size, guidance will always help.

4.6 Architectural Experiments

In this section we explore several variations and modifications to the *VRN - Guided* method, which may improve or impede performance. We provide an overview of experiments along with NME (Equation 4.3.2) in Table 4.3.

4.6.1 On intermediate supervision

As described in the original stacked hourglass paper [75], intermediate supervision is typically used providing significant boost in performance. However, it was not implemented for the basic *VRN* or *VRN - Guided* networks. As an attempt to boost performance, we have trained an identical network to *VRN - Guided*, except with a supervising loss after the first hourglass network. A very minor performance improvement can be observed, as shown in Table 4.3. Additionally, the corresponding curve can be seen in Figure 4.20. We conclude that intermediate supervision is not necessary to achieve high performance for our 3D reconstruction experiment.

4.6.2 On increasing the number of stacks vs the number of modules

Given a fixed computational budget one has two options when implementing a Volumetric Regression Network: to increase the number of stacks while decreasing the number of modules or vice versa. An interesting question which has not been answered yet is whether it is better to have more stacked hourglasses or more modules per hourglass. In this section, we attempt to quantify any performance variation, at least for the task of 3D face reconstruction using volumetric regression while keeping the number of parameters approximately the same.

We evaluate three additional networks, each with different number of hourglasses and modules, such that the total number of parameters remains very similar. These results are shown in detail in Table 4.3 as One, Three or Four Hourglass. The first of these additional networks consists of only one hourglass network and eight modules. Interestingly, this performs the best out of the three, suggesting that for this particular problem, perhaps it is more optimal to increase the number of modules over the number

of hourglasses. This may also be observed from a single hourglass outperforming *VRN - Guided*, which again, may be due to higher performance gains from more modules over stacked hourglasses. However, it can also be observed that the performance difference between all networks is very minor, suggesting that for our experiment as long as the number of parameters remains roughly fixed good performance can be achieved independently of the exact configuration of the hourglass network.

4.6.3 On varying the network parameters

Having investigated the importance of network configuration in the previous section, in this section we keep the architecture fixed and instead vary the number of network parameters to quantify the the performance impact. Hence, in this section we explore the influence of number of parameters, and whether the performance will continue to increase with the number of parameters, or the same performance can be achieved with fewer parameters. If performance saturation has occurred, the dataset has likely become the limiting factor, likely due to variance in the training set, suggesting that it may be possible to achieve a similar level of performance with fewer parameters.

We explore 3 variants of the VRN, corresponding to different number of parameters, namely *Heavy*, *Light* and *Lighter*. Their exact number of parameters and performance is shown in Table 4.3. It can be observed that the performance difference between *Heavy* and *Light* are very minor. However, *Lighter* resulted in a significant drop in performance. With this information we can hypothesise that the optimal number of parameters is the one corresponding to the *Light* network. We show a visual comparison between these three networks in Figure 4.21: the visual performance of the *Lighter* network is clearly worse - the surface is not as smooth and the facial expression is not captured as clearly as in both *Heavy* and *Light*.

4.6.4 A Volumetric Implementation

One of the main contributions of this chapter is to show that good performance for 3D face reconstruction can be achieved even when 2D spatial convolutions are used. However, one might wonder why we decided to use 2D convolutions for a 3D (volumetric) regression task. The short answer is that training a 3D CNN is difficult and highly resource intensive. Firstly, the activations are much larger, consuming more memory leaving less space for parameters. Secondly, the complexity of volumetric convolution is much higher than that of standard spatial convolution, hence the time required to train such a network is far greater, despite the potential for fewer parameters.

For completeness, we have trained a fully volumetric version of VRN where we attempt to match this time the complexity in theoretical floating point operations (FLOPS) for a single forward pass (as opposed to matching the number of parameters) with that of a standard VRN. We use theoretical FLOPS, as opposed to actual or raw number of FLOPS, as frameworks such as CuDNN make numerous optimisations, which often result in fewer operations. Unfortunately, this resulted in having only 32 features per volumetric convolution.

Despite containing far fewer parameters than the standard VRN, the training took over 10 days on two Titan X GPUs and had to be performed with a batch size of 2. This is significantly slower than training the standard VRN, which takes approximately 2 days on the same hardware. We show the NME (Equation 4.3.2) for this network in Table 4.3, along with a curve in Figure 4.20. Unsurprisingly, the performance of this method is quite poor, further illustrating the difficulty in training 3D CNNs despite the problem being a volumetric one.

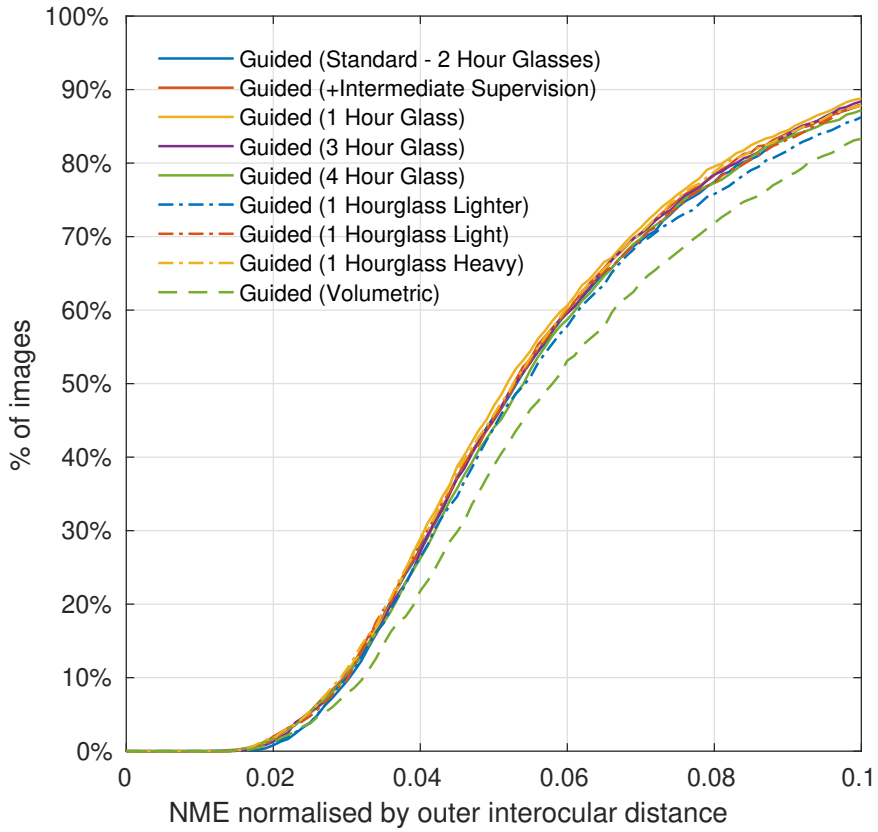


Figure 4.20 Results in NME (Equation 4.3.2) from architectural experiments described in Section 4.6 on the AFLW2000 dataset, showing that in the majority of cases the performance is almost the same.

4.6.5 Architectural Conclusions

As we have seen from the past few experiments, performance in general remains very similar despite architectural changes. Only after reducing the number of parameters significantly were we able observe any significant change. We conclude that with the guidance from detected landmarks only, we are almost able to attain the best performance on these datasets.



Figure 4.21 A visual comparison between the *Heavy*, *Light* and *Lighter* methods described in Section 4.6.3 on three images from the AFLW2000 dataset.

4.7 Conclusions

We proposed a direct approach to 3D facial reconstruction from a single 2D image using volumetric CNN regression. To this end, we proposed and exhaustively evaluated three different networks for volumetric regression, reporting results that show that the proposed networks perform well for the whole spectrum of facial pose, and can deal with facial expressions as well as occlusions. We also compared the performance of our networks against that of recent state-of-the-art methods based on 3DMM fitting reporting large performance improvement on three different datasets. Future work may include improving detail and establishing a fixed correspondence from the isosurface of the mesh.

Chapter 5

3D Human Body Reconstruction from a Single Image via Volumetric Regression



Figure 5.1 Some example results using our method when trained with a high quality detailed training set.

5.1 Introduction

3D reconstruction is the process of estimating the 3D geometry from one or more 2D images. In this chapter, we focus on reconstruction of the human body from a single image, including the non-visible parts which have been self-occluded. Our method builds on the work in Chapter 4, where a 3D face is directly regressed from a single image, using we have already referred to as a "Volumetric Regression Network" (VRN). In this chapter, we show that the same idea can be applied to other deformable objects, in particular, the human body. This poses an array of challenges which are not present when reconstructing the face. While we are still only reconstructing an object of a single class, the body has many more axes of rotation compared to a face. As such, human body reconstruction is often considered to be a very difficult problem.

Motivation. The pipelines required for 3D human reconstruction are typically based on solving difficult non-convex optimisation problems. Perhaps the most common approach to 3D human body reconstruction is to fit a shape model. For example, the recent method of [11], fits a 3D shape model to 2D body joints by jointly optimising the pose and shape. However, optimisation methods are sensitive to initialization and are easily trapped to local minima, both of which are exacerbated by occlusions and potential scale changes.

In this work, we aim to significantly reduce the complexity of standard 3D human reconstruction techniques - to the point where it could just as easily be treated as a segmentation task. We do this by directly regressing a volumetric representation of the 3D geometry using a standard, spatial, CNN architecture, where the regressed volumetric structure is spatially aligned with the input. Notably, we do **not** regress a depth map; the 3D structure is regressed as slices and recovered from its volumet-

ric representation using a standard surface extraction algorithm, such as Marching Cubes [67]. In summary, our main contributions in this work are as follows:

1. We show that volumetric regression networks can also work very well on objects other than the human face, in this case, the full human body.

2. We propose several improvements to the network architecture described in Chapter 4, showing performance improvements. These include introducing intermediate supervision, using more advanced residual modules and altering the network structure by increasing the number of hourglass modules, by reducing the number of residual modules, freeing memory on the GPUs.

3. We show that VRN is capable of reconstructing complex poses when trained on a suitable dataset.

4. Finally, we show that given high quality training data, our method can learn to produce previously unseen, highly detailed, full 3D reconstructions from only a single image. To the best of our knowledge, there is no other method capable of obtaining results with such high fidelity and reliability as ours.

The remainder of the chapter is structured as follows: First, we describe our method, including the volumetric representation we have already mentioned briefly, followed by the datasets and the training procedure. Next we will discuss several architectural variants of VRN, followed by results from a network trained with pose-variant data, but little detail. Finally, we will show results which have been generated by training a model with highly detailed data.

5.2 Method

This section describes our proposed method, including the voxelisation and alignment procedures.

5.2.1 Volumetric Regression

In this work, our goal is to reconstruct the full geometry of a human body from just a single image. There are several ways of estimating the geometry using deep learning. The first is to directly regress the vertices using a top-down network such as VGG [94] or ResNet [39] trained with an l_2 loss. This has at least two disadvantages: firstly it requires the training data to be resampled to have a fixed number of vertices, which implies finding correspondence between all vertices of all meshes. Secondly, and more importantly, training a network to directly regress a very large number of vertices is hard because a single wrong vertex will increase error significantly. A common, and more efficient alternative is to regress the parameters of a 3D shape model. However, as these parameters are not scaled equally, it is necessary to employ normalisation methods, such as weighting the outputs using the Mahalanobis distance which has been also proven challenging to get it working well [46]. Additionally, 3D shape model based approach are known to be good at capturing the coarse shape but less able at capturing fine details (in the case of detailed 3D reconstruction).

To eliminate the aforementioned learning challenges, we reformulate the problem of 3D reconstruction by constraining it to the spatial domain, using a standard convolutional neural network. Our approach can be thought of as a type of image segmentation where the output is a set of slices capturing the 3D geometry. Hence architecturally one can use standard architectures for (say, semantic) segmentation. Following on from the work in Chapter 4, on human faces, we do this by encoding

the geometry of the body in a volumetric representation. In this representation, the 3D space has been discretised with a fixed dimensionality. Space which is *inside* the object is encoded as a voxel with value equal to one. All other space (i.e. background or unknown object classes) are encoded with a voxel with a value equal to zero. For this particular application, the dimensionality of our volumes are $128 \times 128 \times 128$, which given the level of detail in our training set, is more than adequate (although we show in Section 5.5 results with much greater detail, and only a slightly larger volume). One of the main advantages of this representation is that it allows the non-visible (self-occluded or otherwise) parts of the geometry to also be reconstructed. This is not the case in methods attempting to reconstruct the body using depth map regression.

One of the most important aspects to note in the case of training a volumetric regression network is that the input and output must be spatially aligned to avoid requiring the network from learning a difficult transformation between the input image and output volume. Put simply, the 2D projection of the target object should do a reasonable, if not very good, job at segmenting the input. Through experimentation, we have found that it is possible to ignore spatial alignment, as long as the pose is fixed (i.e. always frontal). However, ignoring spatial alignment will severely impact the performance of the method.

When trained to receive guidance from human pose estimation, landmarks are passed to the network as separate channels, each containing a Gaussian centred over the landmark's location. The Guassians have a diameter of approximately 6 pixels.

5.2.2 Dataset and Voxelisation

While Human3.6M [20, 45] does include its own 3D scans, they are not in correspondence with the video frames. As such, we produced our training data by running SMPLify [11]

on the Human3.6M dataset. SMPLify is a 3D body model fitting method which uses the SMPL model [66] by optimising the pose and shape given 2D landmarks as input. The landmarks required by SMPLify were generated using the code made available with [13]. The fitted meshes were voxelised at a resolution of $128 \times 128 \times 128$. The X and Y components of the volume are aligned with the images width and height respectively. However, as we are going from 2D $\rightarrow$ 3D, we must ensure that our third component, Z, is handled sensibly. In terms of depth, the meshes are first aligned by the mean of the Z component. However, through experimentation, we found that as long as the Z alignment is performed in a seemingly sensible way, and remains stable across all images, the network will learn to regress the 3D structure without issue. Random scale augmentation was performed in advance of the training procedure, as doing this on-the-fly (for 3D volumes) can be quite demanding in terms of CPU usage.

An unfortunate side effect of using SMPLify to generate our training data is that it is not possible to regress features such as fingers or facial expressions. SMPLify does not model these, and as such, their pose remains fixed across all images. It also becomes a bottleneck in terms of performance. We show in Section 5.5, using a different dataset, that very high quality reconstruction is also possible with our proposed method.

5.2.3 Training

Our end-to-end network was trained using RMSProp [40] optimisation with a learning rate of 10^{-4}, which was reduced after approximately 20 epochs to 10^{-5} for 40 epochs. We did not observe any performance improvement by reducing this learning rate further. A batch size of 6 was used across 2 NVIDIA 1080 Ti graphics cards and our dataset consisted of 35,000 frames from the 7 participants in Human3.6m. During the voxelisation, random scale augmentation was applied. Applying scale augmentation to

a 3D volume on the fly, is very CPU intensive and slows down the training procedure too much. During training, augmentation to the input image was applied. This on-the-fly augmentation included colour channel scaling, random translation and random horizontal flipping.

5.3 Architecture

In the following subsections, we introduce the several architectural options we have explored as extensions to the original architecture described in Chapter 4. Our first network is the same as the one used for faces, referred to as *VRN - Guided*, which establishes our baseline. This network employs two Encoder-Decoder ("hourglass") networks in a stack. We follow a similar design, aside for the changes described in this section. All of our architectures were trained with the same loss function as used for faces:

$$l_1 = \sum_{w=1}^{W} \sum_{h=1}^{H} \sum_{d=1}^{D} [V_{whd} \log \widehat{V}_{whd} + (1 - V_{whd}) \log(1 - \widehat{V}_{whd})], \tag{5.1}$$

where $\widehat{V}_{whd}$ is the corresponding sigmoid output at voxel $\{w, h, d\}$. In all of the architectures presented, wherever a loss function has been introduced, the default behaviour of summing the losses is used.

5.3.1 Multistack

This network makes the following changes to the *VRN - Guided* baseline network. We half the number of residual modules from four to two. In doing so, we also halved the memory requirements, allowing us to increase the number of hourglass modules in the stack, from two to four. Next, we replace the original residual module used

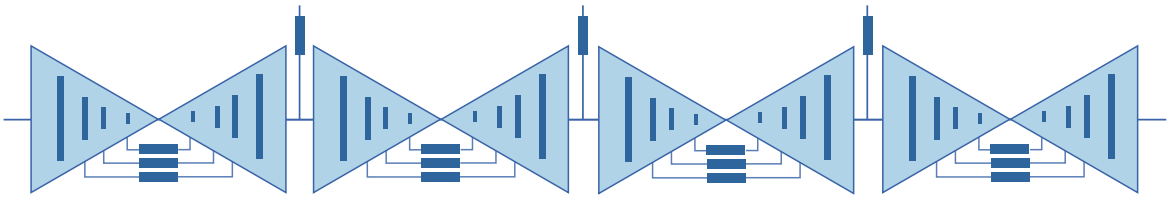


Figure 5.2 The *Ours - Multistack* network. Dark blue boxes represent residual modules. Each Encoder-Decoder module has its own loss, while still passing features to the next module.

in *VRN - Guided* with the multi-scale residual module proposed in [15]. While this module is intended for binarized networks (intended for high speed), we found that this can offer performance gains in non-binary, floating point valued architectures. This module aims to increase the receptive field and improve gradient flow while keeping the number of parameters close to the original. We also show the performance improvement from introducing just this component in the results section. Finally, we introduce supervision after each hourglass module. We therefore have four losses. Each hourglass module forks to provide features for the next hourglass, and to regress the volumetric representation. The performance after each hourglass improves. We found that there was no benefit to adding more than four hourglass networks as the performance just fluctuates as more are added. This network is depicted in Figure 5.2.

5.3.2 Image Only

Our standard network (*Ours - Multistack*) is trained to receive guidance from the landmarks, while also using useful information from the images. With this network, we try to measure the impact of training with just images, while keeping the architecture identical. We call this network *Ours - Image Only*. We expect that the performance of this network be significantly lower than when guidance from the human pose is also provided.

5.3.3 Landmarks Only

Many methods, such as [11, 84], use only the landmarks as input during training and inference. Hence, it is an interesting investigation to measure the performance of our method when only landmarks are provided, without the image. As such, we trained *Ours - Landmarks Only*. However, using only landmarks to fit a shape model results in generic appearing fittings. Provided high quality training data is available, our method can regress these fine details and match the body shape when also provided with the image.

5.3.4 Mask Only

Our method does not rely on a segmentation mask, as is the case in [93]. However, there is no reason why our method cannot reconstruct 3D geometry from a single segmentation mask, or silhouette. To show this, we train another network, *Ours - Segmentation Mask* which accepts only a single channel, containing the mask of the target object. From this, the network reconstructs the 3D geometry in the same way. Once again, this network has an identical configuration to *Ours - Multistack*, apart from the first layer receiving a different number of inputs. We expect this network to perform quite well since the segmentation mask we are providing to the network is the projection of the target volume.

5.3.5 3D Convolution

While volumetric CNNs can likely outdo a spatial network in terms of performance, on this task, the memory requirements are much higher than that of a spatial CNN. So much so, that employing volumetric CNNs at a suitable resolution is not currently

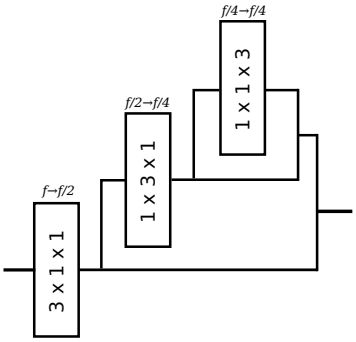


Figure 5.3 A "flat" volumetric residual block

possible. However, we were interested to test a compromise between the two and train a volumetric CNN where the filters are flat. More concretely, where f is the number of features, our filters had sizes $f \times 3 \times 1 \times 1$, $f \times 1 \times 3 \times 1$ or $f \times 1 \times 1 \times 3$. These were combined into a flat volumetric residual module, as shown in Figure 5.3, heavily inspired by [83]. This network also takes as input the image with corresponding landmarks. To provide a fair comparison with the other methods, we match the number of floating point operations of this network to *Ours - Multistack* by reducing the number of parameters (which also allows the network to fit into memory).

5.4 Results

In this section we will give an overview of the performance of the architectures we have described above. For each network, we give our results as an Intersection over Union (IoU) score, which is defined as the number of intersecting set voxels over the number voxels set in either volume. Our test set consisted 500 frames from Session 11 of Human3.6m [45], across all known poses in the dataset. These numerical results can be found in Table 5.1. We will discuss these results in more detail in the proceeding paragraphs.



Figure 5.4 Visual results from our main network, *Ours - Multistack*, on a test split of Human3.6m [45]. These results demonstrate VRN's ability to deal with large and complex pose. We also show the reconstructions with the texture projected onto them.

Table 5.1 Numerical performance of our proposed method and additional architectural experiments, all on data generated using SMPLify.

Method	**IoU @ Epoch 30**	**IoU @ Epoch 60**
VRN - Guided (Baseline)	61.6%	63.9%
Ours - Multistack	61.1%	**68.3%**
Ours - Old Residual	60.5%	66.1%
Ours - Landmarks Only	58.6%	61.0%
Ours - Image Only	46.8%	48.3%
Ours - Mask Only	52.8%	53.0%
Ours - 3D Convolution	57.3%	61.6%

We show visual results for *Ours - Multistack* in Figure 5.4. The quantitative results suggest that the changes we made to the baseline network *VRN - Guided* helped quite significantly, offering a performance increase of over 4% in terms of IoU ($p < 0.01$, paired t-test). From this performance improvement, more than 2% was due to using the residual module proposed in [15], this can be seen from the results for *Ours - Old Residual*. As our data is generated by SMPlify [11], we are unable to provide a quantitative comparison with this method.

As expected, removing either the landmarks *or* the image reduces performance. The best performance is attainable by providing the network with both the image and landmarks, as seen quantitatively between *Ours - Multistack*, *Ours - Landmarks Only* and *Ours - Image Only*. Also unsurprisingly, landmarks alone offers better performance than the image alone. This is true at least in this case, as the groundtruth model has no detail. We also show performance where only the segmentation mask is provided to the network (this is not provided in the case of *Ours - Multiststack*). These results are labelled *Ours - Mask Only*. We expected this network to perform better than the landmarks or image only networks, as the mask we provided was a direct 2D projection of the target volume.

Notes on Performance. A single forward pass through our network takes approximately 200ms on an NVIDIA 1080 Ti GPU. This produces the volumetric representation. Surface extraction introduces 200-600ms overhead depending on the implementation used. Significantly higher performance may be achieved with smaller volumes, but this will result in a lower level of detail. Training typically takes about two days.

5.5 High Quality Training Data

In the previous section, we showed that our method can reconstruct bodies of very large pose. However, due to the dataset we trained on, we are only able to regress the coarse geometry without any detail. Detailed 3D reconstruction was also not demonstrated in the case of faces in [46], which was also due to the lack of a detailed dataset. Hence, in this section, we demonstrate that VRN *is* capable of regressing details when a high quality dataset is provided. For this experiment, we use our best performing network *Ours - Multistack*.

This dataset consists of highly detailed 3D scans from 40 participants, 4 of which were reserved for quantitative testing, but all of which are quite restricted in terms of pose. Only one scan per participant was available. These models do not have a corresponding image which is aligned with the model. As such, we rendered and voxelised these models under a large variety of different lighting conditions, scales and views to create our training set consisting of approximately 20,000 samples which are spatially aligned. The voxelisation was performed at a resolution of $128 \times 256 \times 96$, which efficiently encapsulates the poses found in the dataset. As in our previous experiment, the Z alignment was performed by the mean Z component. Unfortunately we are not able to publicly release this dataset.



Figure 5.5 Example 3D reconstructions from the web (Creative Commons) using our method trained with high quality training data. The first row shows the input image, the second shows the 3D reconstruction from the front, and the third row shows the 3D reconstruction when views from behind (i.e. the hallucinated side, in the case of these images). The final row shows the frontal reconstruction with the projected texture. These results show that VRN is capable of regressing finer details.

5.5.1 Performance

The four models which we reserved for testing were also rendered and voxelised in the same way as above, to produce 60 testing images. Our method reconstructs these with an IoU of 78%. This is significantly higher than the reconstructions in our previous experiment. This is likely due to the better spatial alignment between the training images and target. Additionally, we show qualitative results on real-world images taken from the web [1]. These reconstructions can be found in Figure 5.5. We show the backsides of these reconstructions, which demonstrate the networks ability to reconstruct the self-occluded body parts. Finer details can be seen in the wrinkles of clothing. As our method was trained on synthetic data, we believe there may be some performance degradation on real-world images. Additionally, several of the poses found in the reconstructions in Figure 5.5 are not found in the 36 training samples. This suggests that VRN is somewhat tolerant to previously unseen poses.

5.6 Conclusion

In this chapter, we have shown that using Volumetric Regression Networks, as described in [46], for the task of 3D reconstruction, is not restricted to the simpler task of face reconstruction, nor is this a limiting factor in terms of detail, despite the small size of the volumes we are working with. We have proposed several improvements to the original VRN which improve the performance quite substantially. Finally, we have shown, by using two different datasets, that VRN can regress both unusual poses (in networks trained on Human3.6m), and high levels of detail (in the case of our private but detailed dataset). We believe that given a large enough dataset containing many

[1] These images are licensed under Creative Commons. Attribution, where required, has been provided our website.

pose variations, and high levels of detail, the network will be capable of large pose 3D human reconstruction, while also capturing fine details, from a single image.

Chapter 6

A CNN Cascade for Landmark Guided Semantic Part Segmentation

The exact relationship between this work on facial part segmentation and the main focus of this thesis may not appear obvious, nor is it particularly relevant to the area of *Volumetric Regression Networks*. However, this work strongly influenced the future work, since in many ways our 3D reconstruction approach can be thought of a type of two class segmentation problem (inside the face or body, or outside). Additionally, this work was where we first began to explore how the use of guidance, in the form of facial landmarks, can assist the problem of facial part segmentation, and later, the problem of 3D face reconstruction.

This chapter proposes a CNN cascade for semantic part segmentation of guided by pose-specific information encoded in terms of a set of landmarks. There is large amount of prior work on each of these tasks separately, yet, to the best of our knowledge, this is the first time in literature that the interplay between pose estimation and semantic

part segmentation is investigated. To address this limitation of prior work, in this chapter, we propose a CNN cascade of tasks that firstly performs landmark localisation and then uses this information as input for guiding semantic part segmentation. We applied our architecture to the problem of facial part segmentation and report large performance improvement over the standard unguided network on the most challenging face datasets.

6.1 Introduction

Pose estimation refers to the task of localising a set of landmarks (or keypoints) on objects of interest like faces [27], the human body [110] or even birds [112]. Locating these landmarks help establish correspondences between two or more different instances of the same object class which in turn has been proven useful for fine-grained recognition tasks like face and activity recognition. Part segmentation is a special case of semantic image segmentation which is the task of assigning an object class label to each pixel in the image. In part segmentation, the assigned label corresponds to the part of the object that this pixel belongs to. In this chapter, we investigate whether pose estimation can guide contemporary CNN architectures for semantic part segmentation. This seems to be natural yet to the best of our knowledge this is the first work that addresses this problem. To this end, we propose a Convolutional Neural Network (CNN) cascade for landmark guided part segmentation and report large performance improvement over a standard CNN for semantic segmentation that was trained without guidance.

Although the ideas and methods presented in this chapter can probably be applied to any structured deformable object (e.g. faces, human body, cars, birds), we will confine ourselves to human faces. The main reason for this is the lack of annotated



Figure 6.1 Example faces and their corresponding output from the CNN cascade.

datasets. To the best of our knowledge, there are no datasets providing pixel-level annotation of parts and landmarks at the same time. While this is also true for the case of human faces, one can come up with pixel-level annotation of facial parts by just appropriately connecting a pseudo-dense set of facial landmarks for which many datasets and a very large number of annotated facial images exist, see for example [91]. Note that during testing we do not assume knowledge of the landmarks' location, and what we actually show is that a two-step process in which a CNN firstly predicts the landmarks and then uses this information to segment the face largely outperforms a CNN that was trained to directly perform facial part segmentation.

6.1.1 Main contributions

In summary, this chapter addresses the following research questions:

1. Is a CNN for facial part segmentation needed at all? One might argue that by just predicting the facial landmarks and then connecting them in the same way as we created the part labels, we could get high quality facial part segmentation thus completely by-passing the part segmentation task. Our first result in this chapter is that indeed the latter method slightly outperforms a CNN trained for facial part segmentation (without guidance though).

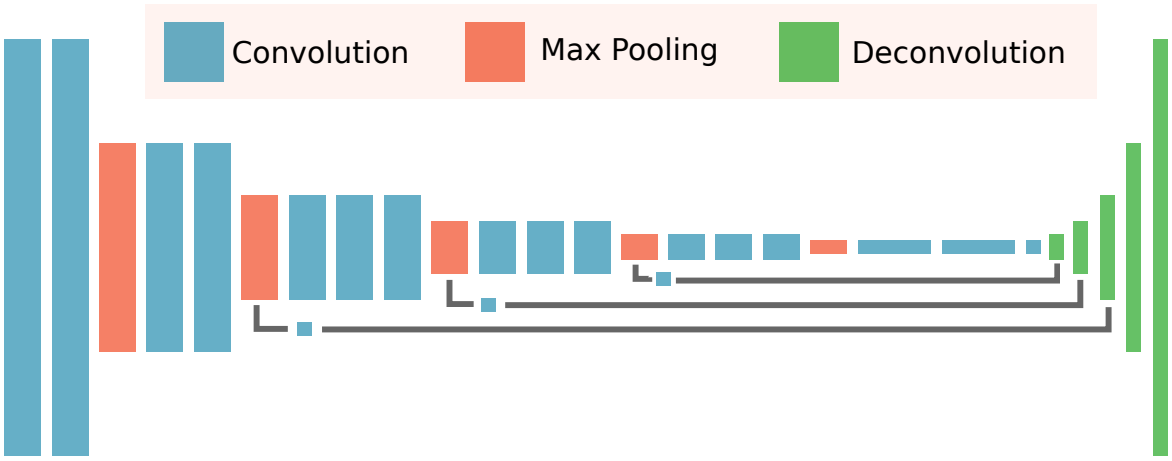


Figure 6.2 Overview of the Fully Convolutional Network [65], low level information providing refinement are reintroduced into the network during deconvolution.

2. Can facial landmarks be used for guiding facial part segmentation, thus reversing the result mentioned above? Indeed, we show that the proposed CNN cascade for landmark guided facial part segmentation largely outperforms both methods mentioned above without even requiring very accurate localisation of the landmarks. Some example output can be seen in Fig 6.1.

6.2 Literature Review

This literature review is included here, instead of Chapter 3 to avoid distracting from the primary focus of this thesis.

6.2.1 Semantic Segmentation

Semantic segmentation is the process by which pixels of an image are labelled by the objects contained in the image. For example, all dogs may be labelled as red, while cats may be labelled as green. In order for this to work, context is required, both locally and globally. Global context tells us what is in the image and roughly where a

particular object may be located. Local context then refines this global information, defining boundaries between classes with greater precision.

Classical approaches to semantic segmentation involved first segmenting the image using statistical models, followed by individually classifying the segmented objects. See for example [2, 19], which use a Conditional Random Field (CRF) to separate the image. However, these methods generally struggle to adequately incorporate high and low level feature descriptors necessary for accurate segmentation and labelling.

CNNs drastically changed the landscape of semantic segmentation. The first work producing satisfactory semantic segmentation was the Fully Convolutional Network (FCN) described in [65], which is based on VGG-16 [94]. In this work, information is forked from different layers throughout the network. Each fork is used to produce an intensity map for all known class (initially 59, targeted at the PASCAL VOC 2010 dataset [35]). Forks earlier on in the network provide very detailed but noisy predictions about the location of classes, where as the latter forks are have very little noise, but can only approximate the location of a class. A diagram of the structure of FCN is shown in Figure 6.2. FCN was not intended to be trained from scratch. Instead, training was initialised from the parameters of VGG-16, once the network had been trained on the ImageNet dataset [58] for the task of classification. The parameters in the last few layers of the network were reshaped from linear layers into a 1×1 convolution. An interesting side effect of which, is the ability for the network to accept inputs of arbitrary sizes. The size of the input no longer dictates the number of parameters in the linear layers. Due to its simplicity and (at the time) state of the art results, it is no surprise the ideas behind FCN have become the standard for modern semantic segmentation methods.

Still, one of the limitations of FCN is that the resolution of its predictions are quite low. As such, several methods have proposed extensions to FCN to compensate for

this limitation. The most common addition has been to add a CRF to the output of the network, to provide further refinement. The work of [21] first upsamples the predicted scores using bilinear interpolation and then refines the output by applying a dense CRF. The method of [116] performs recurrent end-to-end training of the FCN and the dense CRF. Finally, the work in [76] employs learnt deconvolution layers, as opposed to fixing the parameters with an interpolation filter (as in FCN). These filters learn to reconstruct the object's mask, in addition to upsampling its input.

In the past year or so, Encoder-Decoder style network architectures (such as [75]) have gained a great deal of popularity. This architecture reduces the spatial dimensionality of the feature maps as they are passed through the *encoder*. These features are then upsampled again using a *decoder* part of the network, while being recombined with lower level information, much like in FCN [65]. This architecture has also been referred to as an hourglass network. This name was first used in [75], and performed human pose estimation with great precision. This level of precision is due to the being suited to combining features from the high and low parts of the network efficiently (i.e. combining both global and local context). As such, similar architectures have been used for semantic segmentation problems. For example, SegNet [3] uses an Encoder-Decoder architecture to perform semantic segmentation on 11 classes for autonomous driving in real time.

Architectures aside, a main drawback of this supervised approach to semantic segmentation is producing large quantities of detailed groundtruth semantic masks. There are several large datasets already available, such as PASCAL VOC [35], Microsoft COCO [62] and Cityscapes [28]. All of these datasets are fairly constrained in terms of the number of known classes. Respectively, these datasets have 59, 183 and 30 known classes. This is problematic when it comes to making semantic segmentation a viable tool for general semantic segmentation *in-the-wild*. As such, numerous approaches have

been devised to train these models in ways requiring less human intervention. In [87], hooks are added to visually detailed computer games, to allow automatic generation of instance aware semantic segmentation masks. A drawback of this is that most games are far from realistic enough to work on real images. As such, models trained on this data will also require domain adaptation - which is an entire research area in itself. In [6], a weakly supervised approach is presented, where groundtruth semantic segmentation masks are avoided entirely. Instead, the method learns the semantic segmentation task from just a single point on each object.

Another approach is described in [42], where the training set is formed of images where all classes are annotated with bounding boxes. However, only some classes have groundtruth masks. The network is trained to produce both bounding boxes and semantic segmentation masks simultaneously by learning a mapping between the visual embedding from the bounding box feature extractors, to the semantic segmentation feature extractors. They refer to this approach as *partially* supervised, as some groundtruth data is required. Additionally, the method has been shown to be able to segment up to 3000 classes - far higher than [65], which was trained on the 59 classes from VOC PASCAL 2010 [35].

This chapter describes our method for facial part segmentation, in which each component of the face (such as eyes, lips, nose, etc) are assigned a pixel wise label. Deep learning based approaches to semantic segmentation, particularly the fully convolutional approach described in [65], have been heavily influential in our work on 3D reconstruction due to the spatial nature of our approach.

6.2.2 Part Segmentation

There have been also a few works that extend semantic segmentation to part segmentation with perhaps the most well-known being the Shape Boltzman Machine [33, 34]. This work has been recently extended to incorporate CNN refined by CRF features (as in [21]) in [105]. Note that this work aims to refine the CNN output by applying a Restricted Boltzmann Machine on top of it and does not make use of pose information as provided by landmarks. In contrast, we propose an enhanced CNN architecture which is landmark-guided, can be trained end-to-end and yields large performance improvement without the need of further refinement.

6.2.3 Facial Part Segmentation

One of the first face segmentation methods prior to deep learning is known as LabelFaces [108] which is based on patch classification and further refinement via a hierarchical face model. Another hierarchical approach to face segmentation based on Restricted Boltzmann Machines was proposed in [70]. More recently, a multi-objective CNN has been shown to perform well for the task of face segmentation in [64]. The method is based on a CRF the unary and pairwise potentials of which are learnt via a CNN. Softmax loss is used for the segmentation masks, and a logistic loss is used to learn the edges. Additionally, the network makes use of a non-parametric segmentation prior which is obtained as follows: first facial landmarks on the test image are detected and then all training images with most similar shapes are used to calculate an average segmentation mask. This mask is finally used to augment RGB. This segmentation mask might be blurry, does not encode pose information and results in little performance improvement.

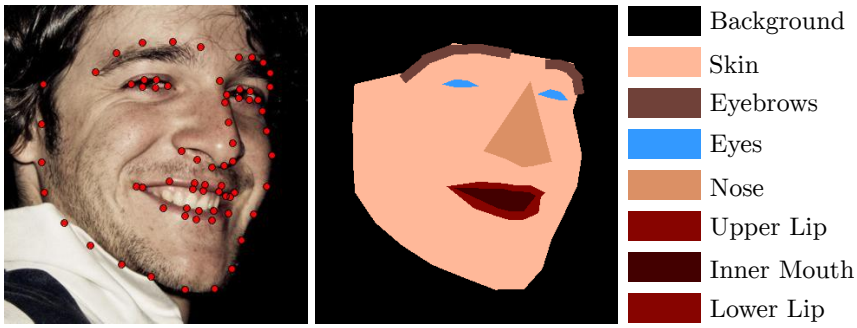


Figure 6.3 Example groundtruth segmentation mask produced from the groundtruth landmarks.

6.3 Datasets

There are a few datasets which provide annotations of pixel-level parts [10, 53, 23] but to the best of our knowledge there are no datasets containing both part and landmark annotations. Hence, in this work, we rely on datasets for facial landmarking. These datasets provide a pseudo-dense set of landmarks. Segmentation masks are constructed by joining the groundtruth landmarks together to fully enclose each facial component. The eyebrows are generated by a spline with a fixed width relative to the normalised face size, to cover the entire eyebrow. The selected classes are background, skin, eyebrows, eyes, nose, upper lip, inner mouth and lower lip. While this results in straight edges between landmarks, the network can learn a mean boundary for each class. The output from the network will be actually smoother than the groundtruth. This process is illustrated in Figure 6.3.

For our experiments we used the 68-point landmark annotations provided by the 300W challenge [90]. In particular the training sets of LFPW [7] (811 images), Helen [59] (2000 images), AFW [121] (337 images) and iBUG [90] (135 images) are all used for training while the 300W test set (600 images) is used for testing. Both training and test sets contain very challenging images in terms of appearance, pose, expression and occlusion.

Table 6.1 The VGG-FCN [94, 65] architecture employed by our landmark detection and semantic part segmentation network.

Layer Name	Kernel	Stride	Outputs	Layer Name	Kernel	Stride	Outputs
conv1_1	3×3	1×1	64	pool4	2×2	2×2	–
conv1_2	3×3	1×1	64	conv5_1	3×3	1×1	512
pool1	2×2	2×2	–	conv5_2	3×3	1×1	512
conv2_1	3×3	1×1	128	conv5_3	3×3	1×1	512
conv2_2	3×3	1×1	128	pool5	2×2	2×2	–
pool2	2×2	2×2	–	fc6_conv	7×7	1×1	4096
conv3_1	3×3	1×1	256	fc7_conv	1×1	1×1	4096
conv3_2	3×3	1×1	256	fc8_conv	1×1	1×1	68 or 7
conv3_3	3×3	1×1	256	deconv_32	4×4	2×2	68 or 7
pool3	2×2	2×2	–	score_pool4	1×1	1×1	68 or 7
conv4_1	3×3	1×1	512	deconv_16	4×4	2×2	68 or 7
conv4_2	3×3	1×1	512	score_pool3	1×1	1×1	68 or 7
conv4_3	3×3	1×1	512	deconv_8	16×16	8×8	68 or 7

This collection of images undergoes some pre-processing before they are used to train the network. The faces are normalised to be of equal size and cropped with some noise added to the position of the bounding box. Not all images are the same size, but their height is fixed at 350 pixels. With probability $p = 0.5$, a randomly sized black rectangle, large enough to occlude an entire component is layered over the input image. This assists the network in learning a robustness to partial occlusion.

6.4 Method

We propose a CNN cascade (shown in Figure 6.4 and listed in Table 6.1) which performs landmark localisation followed by facial part segmentation. Our cascade was based on the VGG-FCN [94, 65] using Caffe [47] and consists of two main components:

1. Firstly, an FCN is trained to detect facial landmarks using Sigmoid Cross Entropy Loss.

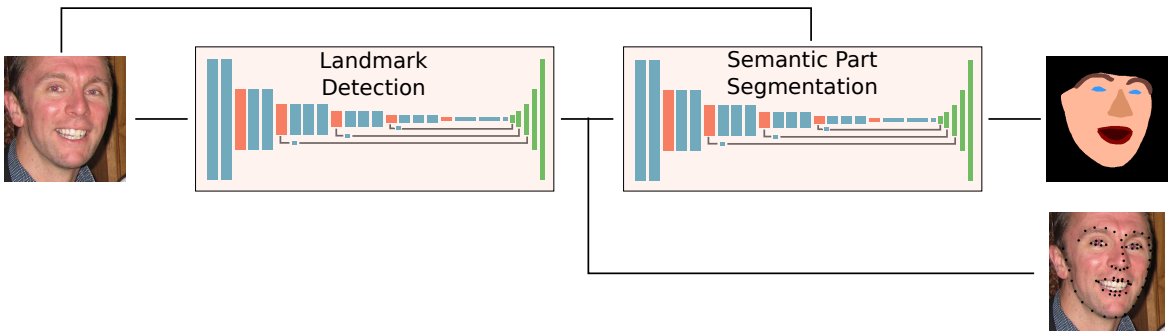


Figure 6.4 The proposed architecture, comprising of two separate Fully Convolutional Networks. The first performs Landmark Detection, the output of which is encoded as multichannel representation which is then passed into the Semantic Part Segmentation network.

2. Secondly, inspired by the human pose estimation method of [18], the detected 68 landmarks are encoded as 68 separate channels each of which contains a 2D Gaussian centred at the corresponding landmark's location. The 68 channels are then stacked along with the original image and passed into our segmentation network. This is a second FCN trained for facial part segmentation using as input the stacked representation of 2D Gaussians and image, and a standard Softmax loss.

Overall we encode pose specific information by augmenting the input to our segmentation network with a multi-channel confidence map representation using Gaussians centred at the predicted landmarks' location. Hence, our FCN for semantic segmentation is trained to produce high quality, refined semantic masks by incorporating low level information with globally aware information. Each of the aforementioned components is now discussed in more detail:

Facial Landmark Detection The training procedure for landmark detection is similar to training FCN for part segmentation. Landmarks are encoded as 2D Gaussians centred at the provided landmarks' location. Each landmark is allocated its own channel

to prevent overlapping with other landmarks and allow the network to more easily distinguish between each point. The main difference with part segmentation is the loss function. Sigmoid Cross Entropy Loss [112] was chosen to regress the likelihood of a pixel containing a point. More concretely, given our groundtruth Gaussians $\hat{p}$ and predicted Gaussians p, each of equal dimensions $N \times W \times H$, we can define the Sigmoid Cross Entropy loss l as follows:

$$l = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{W} \sum_{j=1}^{H} [p_{i,j}^{n} \log(\hat{p}_{i,j}^{n}) + (1 - p_{i,j}^{n}) \log(1 - \hat{p}_{i,j}^{n})]. \tag{6.1}$$

The loss was scaled by $1e^{-5}$ and a learning rate of 0.0001 was used. The network was trained in steps as previously described, for approximately 400,000 iterations, until convergence.

Guided Facial Part Segmentation To train our guided FCN part segmentation network we followed [65]. Softmax Loss was also used. If N is the number of outputs (in our case, classes), $p_{i,j}$ is the predicted output for pixel (i, j), and n is the true label for pixel (i, j), then the Softmax loss l can be defined as:

$$l = \frac{-1}{N} \sum_{i=1}^{W} \sum_{j=1}^{H} \log(p_{i,j}^{n}). \tag{6.2}$$

We firstly trained an unguided FCN for facial part segmentation following [65]. Initially, the network was trained as 32 stride, where no information from the lower layers is used to refine the output. In this configuration, each pixel of the final output comes only from 32×32 of the input, resulting in only an approximate segmentation. This followed by introducing information from pool4, followed by pool3. A learning rate of 0.0001 was chosen, and a momentum of 0.9. The network was trained for approximately 300,000 iterations until convergence.

Then, our guided FCN was initialised from the weights of the unguided one, by expanding the first layer to accommodate the additional 68 input channels. As mentioned earlier, each channel contains a 2D Gaussian centred at the corresponding landmark's location. A key aspect of our cascade is how the landmarks' location is determined during training. We cannot use the groundtruth landmark locations nor the prediction of our facial landmark detection network on our training set as those will be significantly more accurate than those observed during testing. Hence, we applied our facial landmark detection network on our validation set and recorded the landmark localisation error. We used this error to create a multivariate Gaussian noise model that was added to the groundtruth landmark locations of our training set. This way our guided segmentation network was initialised with much more realistic input in terms of landmarks' location. Furthermore, the same learning rate of 0.0001 was used. For the first 10,000 iterations, training was disabled on all layers except for the first. This allowed the network to warm up slightly, and prevent the parameters in other layers from getting destroyed by a high loss.

6.5 Experiments

6.5.1 Overview of Results

In all experiments we used the training and test sets detailed in Section 6.3. As a performance measure, we used the familiar intersection over union measure [65]. We report a comparison between the performance of four different methods of interest:

1. The first method is the VGG-FCN trained for facial part segmentation. We call this method **Unguided**.

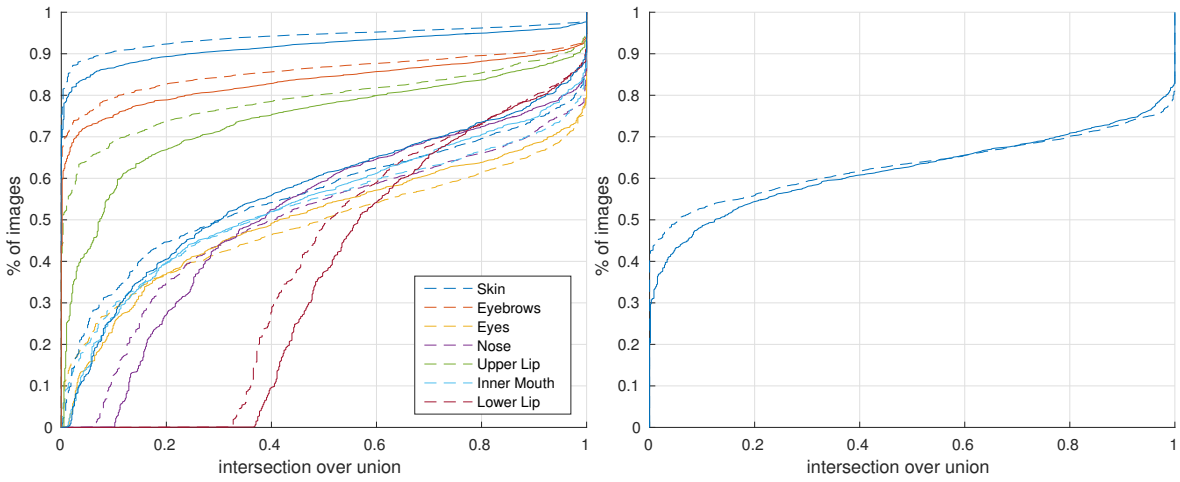


Figure 6.5 Comparison of Unguided (—) and Connected Landmarks (- -). Per-class averages shown on the right.

2. The second method is the part segmentation result obtained by joining the landmarks obtained from VGG-FCN trained for facial landmark detection. We call this method **Connected Landmarks**.

3. The third method is the proposed landmark guided part segmentation network where the input is the groundtruth landmarks' location. We call this method **Guided by Groundtruth**.

4. Finally, the fourth method is the proposed landmark guided part segmentation network when input is detected landmarks' location. We call this method **Guided by Detected**.

The first two methods are the baselines in our experiments while the third one provides an upper bound in performance. The fourth method is the proposed CNN cascade.

6.5.2 Unguided Facial Part Segmentation

To establish a baseline, an unguided fully convolutional network was firstly trained. This was done as described in the FCN paper [65] and Section 6.4. Some visual results

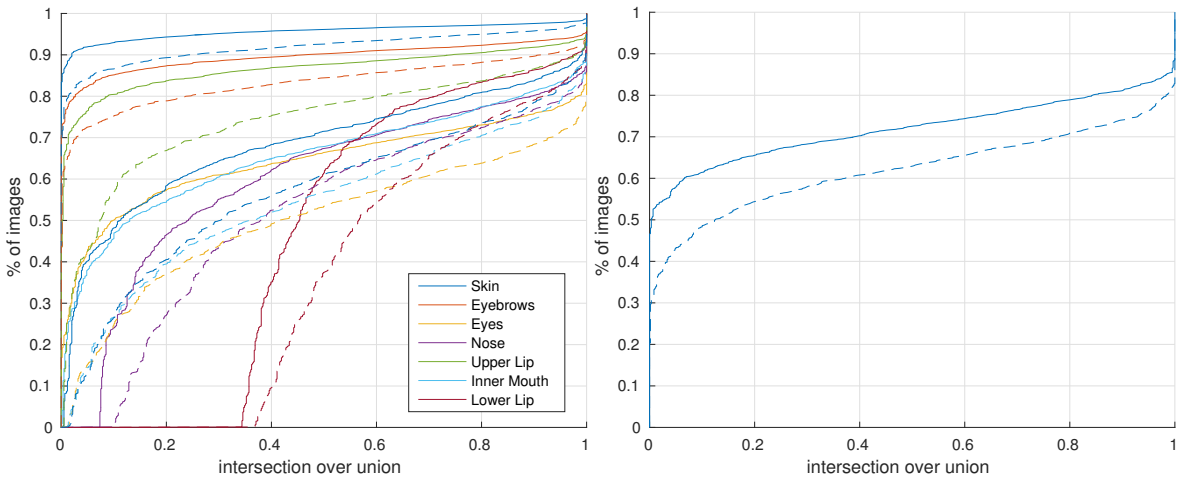


Figure 6.6 Comparison of guided by groundtruth (—) and unguided (- -) facial part segmentation. Per-class averages shown on the right.

can be seen in Figure 6.9. Additionally, a second baseline was obtained by simply connecting the landmarks of our facial landmark detection network also described in Section 6.4. The performance of both baselines can be seen in Figure 6.5. We may observe that connecting the landmarks appears to offer slightly better performance than FCN for part segmentation alone. Nevertheless, we need to emphasise that the groundtruth masks were obtained by connecting the landmarks and hence there is some bias towards the connecting the landmarks approach.

6.5.3 Guided Facial Part Segmentation with Groundtruth

To establish an upper bounds to our performance, a fully convolutional network was trained to accept guidance from groundtruth landmarks. As described in Section 6.4, the guidance is provided in the form of landmarks encoded as 2D Gaussians. The performance difference between unguided and groundtruth guided part segmentation can be seen in Figure 6.6. These results are not surprising given that the groundtruth semantic masks are generated from the landmarks guiding the network. Furthermore, landmark detection offers an advantage because, in the case of faces, there can only be

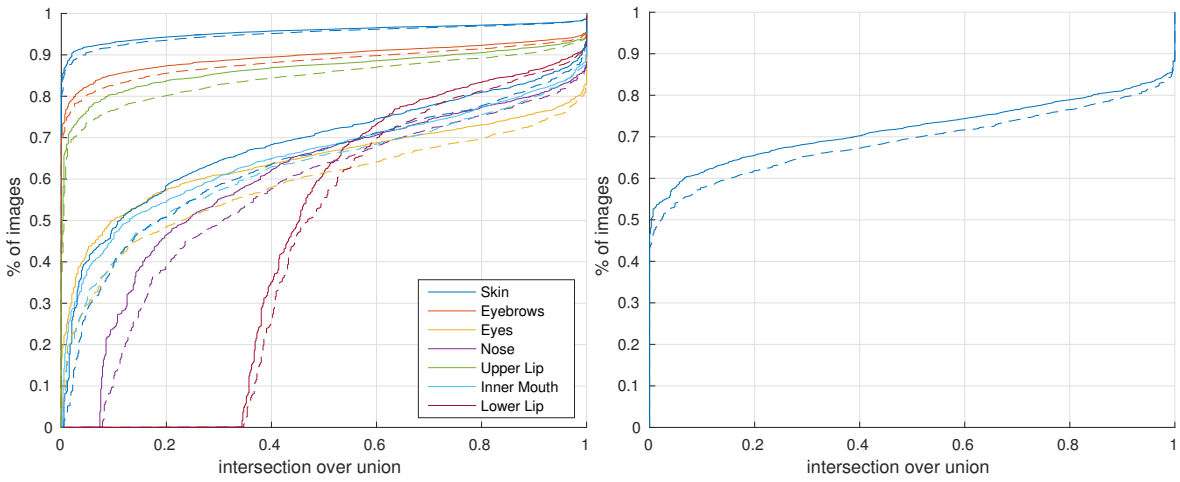


Figure 6.7 Comparison of guidance from groundtruth landmarks (—) and guidance from detected landmarks (- -). Per-class averages shown on the right.

one tip of the nose, and one left side of the mouth. Giving some information to the network about where it is likely to be located can offer a significant advantage. Our next experiment shows that this is still the case when detected landmarks are used instead of groundtruth landmarks.

6.5.4 Guided Facial Part Segmentation with Detected Landmarks

With our upper bound and baselines defined, we can now see how much of an improvement we can achieve by guiding the network with our detected landmarks. The output of the landmark detection network is passed into the part segmentation network along with the original input image. We acknowledge that the performance of our landmark detector is far from groundtruth, despite this we observe performance gains by using this as guidance. We measure the performance as mean point to point Euclidean distance normalised by the outer interocular Euclidean distance, as in [90]. This results in an error of 0.0479. However, we show that the performance of the segmentation is improved significantly. The results of facial part segmentation guided by the detected

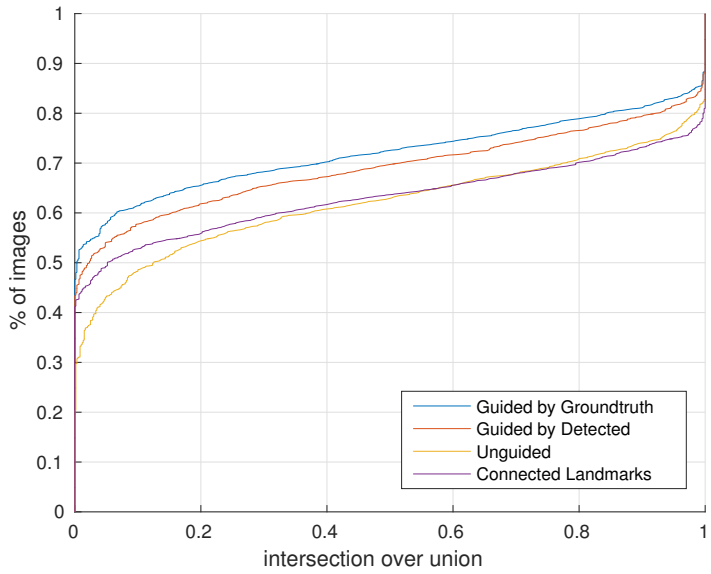


Figure 6.8 Average performance of the four tested methods over all facial components: part segmentation guided by groundtruth landmarks, part segmentation guided by detected landmarks, unguided part segmentation, part segmentation by joining up the detected landmarks.

landmarks, compared to the network guided by groundtruth landmarks can be seen in Fig 6.7. Guided by groundtruth landmarks offers an upper bound to our experiments, but we note a fixed improvement in terms of performance compared to unguided, and a small improvement on those which already have low error. Our main result is that performance of the guided by detected network is very close to the that of the guided by groundtruth, illustrating that in practice accurate landmark localisation is not really required to guide segmentation. Some visual results can be seen in Figure 6.9. Finally, performance over all components for all methods is given in Figure 6.8.

6.6 Conclusion

In this chapter we proposed a CNN architecture to improve the performance of part segmentation by task delegation. In doing so, we provided both landmark localisation and semantic part segmentation on human faces. However, our method should be

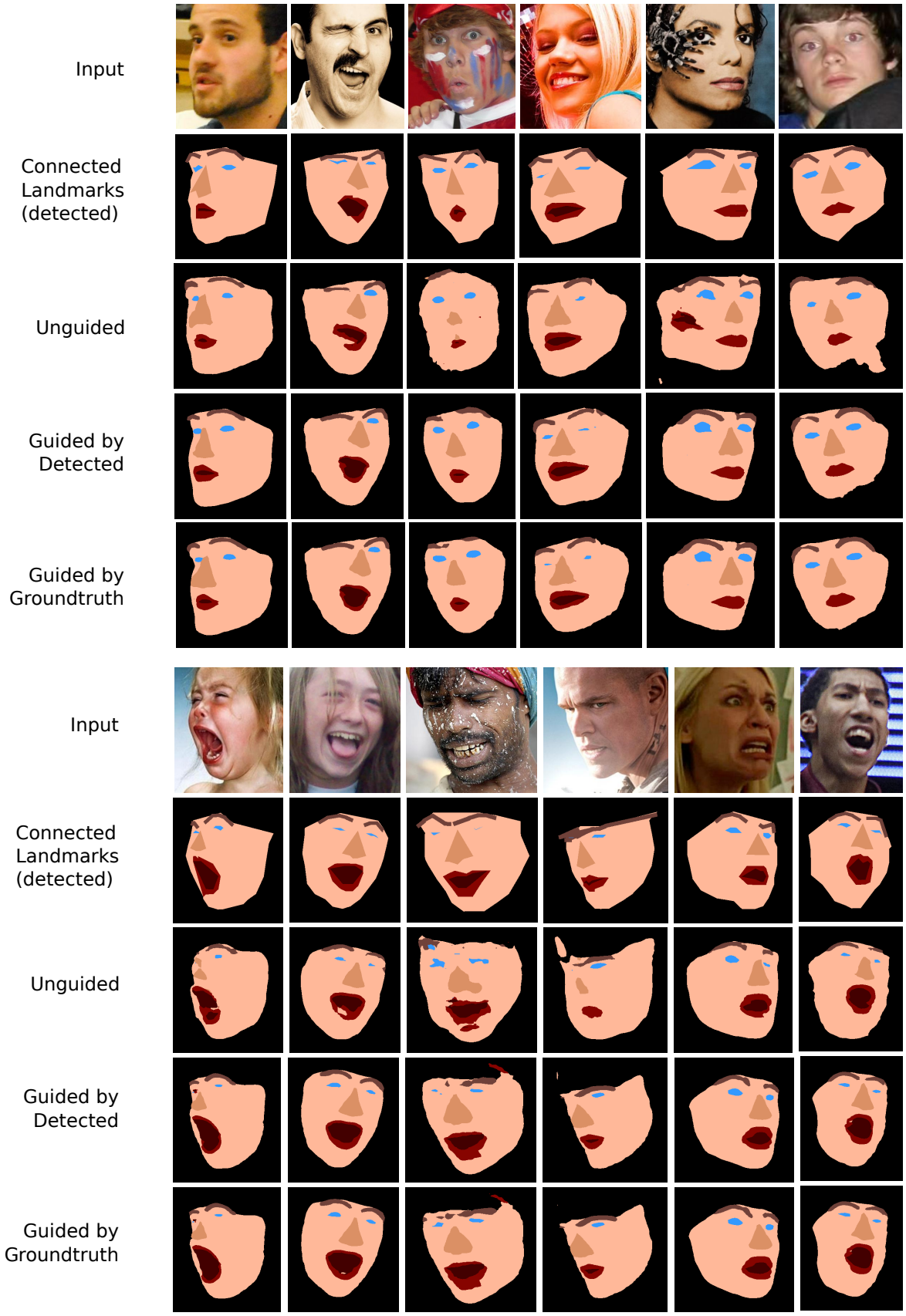


Figure 6.9 Some visual results showing where the unguided network begins to fail, and where the guidance begins to pay off. Observe how visually close the results of the guided by groundtruth landmarks and the guided by detected landmarks networks are.

applicable to our objects as well. This is the focus of our ongoing work. We are also looking into how the segmentation masks can be further used to improve landmark localisation accuracy, thus leading to a recurrent architecture. Future work may also compare the performance of this method with a multitask architecture.

Chapter 7

Conclusion

In this thesis, we have proposed a novel approach to 3D face reconstruction, which can be learnt in an end-to-end fashion and is resilient to large pose, unusual expressions and occlusion. Additionally, our method is insensitive to alignment, minimising the requirements for careful initialisation, improving reliability overall. We compared our method to several other state-of-the-art methods, and out performed all of them by a large margin. We showed how the related task of face alignment can be used to assist the 3D reconstruction process. We also demonstrated that our method is able to work well on the task of human body reconstruction. In doing so, we also demonstrated that our method is able to reconstruct fine details, such as wrinkles in clothing, when trained with high quality data. This strongly suggests that our method will also be capable of working on other deformable objects.

To demonstrate our confidence in the 3D face reconstruction method presented in Chapter 4, a website was developed[1], allowing users to upload an image of a face and be presented with an interactive 3D model. The impact of which far exceeded any of our expectations. To date, this website has reconstructed more than 1 million faces.

[1] http://cvl-demos.cs.nott.ac.uk/vrn

Feedback from users has been positive and the resultant models have been used for a multitude of purposes; from comforting those who have lost loved ones, to being used in music videos and assisting in teaching school children about 3D printing. Additionally, code for our 3D face reconstruction method has been made available[2] under the MIT license.

7.1 Future Work

In Chapter 4 we presented results which demonstrate Volumetric Regression Networks as being a high performing, state-of-the-art method for 3D face reconstruction. Despite these results, 3D face reconstruction will continue to be an active research area for quite some time. In this section, we will discuss some potential directions for future work, specifically aimed at Volumetric Regression Networks.

7.1.1 Detail Refinement

Volumetric Regression Networks require a significant amount of training data to produce high quality results - this is a limitation of our method. Furthermore, obtaining high quality *and* detailed data, containing blemishes and wrinkles, in a natural (or *in-the-wild* setting is difficult, if not impossible. As such, there is a clear requirement for a method which is able to learn this in a semi-supervised, if not entirely unsupervised manner.

[2]https://github.com/AaronJackson/vrn

7.1.2 Texture Synthesis

Our method for 3D face reconstruction does not attempt to produce any kind of texture map or colour information, instead we focused on the reconstruction aspect only. The online demo of this method, mentioned earlier, does render faces with a texture. However, this is simply the original image projected onto the 3D model using the x, y coordinates. This occasionally results in strange artefacts around the sides of the reconstruction. An interesting direction for future work may be to improve this, by simultaneously regressing a warped texture image which can wrap around the regressed 3D model.

7.1.3 Fixed Correspondence

One of the advantages of 3DMM based approaches to 3D face reconstruction is that vertices have a fixed index across all meshes (except for a few cases, such as [104], where the mesh is modified in a refinement step). As VRN produces a volumetric representation, both the number of vertices and the order of them varies, depending on the face or surface extraction algorithm used. This somewhat hinders the ease-of-use for applications, as vertices would have to be selected manually. There are a number of potential solutions to this problem, such as regressing a vertex map, in a similar fashion to aforementioned texture synthesis. Alternatively, as a post-processing step, resampling and ordering the vertices.

7.1.4 Temporal Consistency

As our method was designed to work from just a single image, no consideration was really given to how well it would work over temporal data. While we feel the

performance is reasonably good on videos, it is possible that the shape of the face might change, potentially making it unsuitable for a variety of purposes. Hence, a nice addition to our method would be to optionally accept features from several previous frames. This would allow some temporal consistency to be established, hopefully constraining the face shape over time.

7.1.5 Performance and Speed

The performance of our 3D reconstruction method is limited by what makes it an effective method, the volumetric representation. Unfortunately, 3D volumes are generally quite large. The drawback of this is that it takes time to read and write them to disk, as well as apply transformations for data augmentation. However, there are perhaps several optimisations which could be made to improve this. Examples might include a network configuration which is using half precision floats, which would also allow the models to be saved in half precision floats, instead of converting them from bytes (which also takes time). There is plenty of room for performance improvements towards the way they are stored on disk - Run Length Encoding (RLE) could certainly be employed to compact the contiguous blocks of ones or zeros into something significantly faster to read.

References

[1] Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. (2005). Scape: shape completion and animation of people. In *Proceedings of ACM Transactions on Graphics (TOG)*, volume 24, pages 408–416.

[2] Arbeláez, P., Hariharan, B., Gu, C., Gupta, S., Bourdev, L., and Malik, J. (2012). Semantic segmentation using regions and parts. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3378–3385.

[3] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 39(12):2481–2495.

[4] Bagdanov, A. D., Masi, I., and Del Bimbo, A. (2011). The florence 2d/3d hybrid face datset. In *Proceedings of ACM Multimedia International Workshop on Multimedia Access to 3D Human Objects (MA3HO)*, pages 79–80.

[5] Balan, A. O., Sigal, L., Black, M. J., Davis, J. E., and Haussecker, H. W. (2007). Detailed human shape and pose from images. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

[6] Bearman, A., Russakovsky, O., Ferrari, V., and Fei-Fei, L. (2016). What's the point: Semantic segmentation with point supervision. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 549–565.

[7] Belhumeur, P., Jacobs, D., Kriegman, D., and Kumar, N. (2011). Localizing parts of faces using a consensus of exemplars. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 545–552.

[8] Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–606.

[9] Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3d faces. In *Proceedings of Computer Graphics and Interactive Techniques*, pages 187–194.

[10] Bo, Y. and Fowlkes, C. C. (2011). Shape-based pedestrian parsing. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2265–2272.

[11] Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., and Black, M. J. (2016). Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 561–578.

[12] Booth, J., Roussos, A., Ververas, E., Antonakos, E., Ploumpis, S., Panagakis, Y., and Zafeiriou, S. (2018). 3d reconstruction of "in-the-wild" faces in images and videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(11):2638–2652.

[13] Bulat, A. and Tzimiropoulos, G. (2016a). Human pose estimation via convolutional part heatmap regression. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 717–732.

[14] Bulat, A. and Tzimiropoulos, G. (2016b). Two-stage convolutional part heatmap regression for the 1st 3d face alignment in the wild (3dfaw) challenge. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 616–624.

[15] Bulat, A. and Tzimiropoulos, G. (2017a). Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 3706–3714.

[16] Bulat, A. and Tzimiropoulos, G. (2017b). How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *Proceedings of International Conference on Computer Vision*, pages 1021–1030.

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

[18] Carreira, J., Agrawal, P., Fragkiadaki, K., and Malik, J. (2016). Human pose estimation with iterative error feedback. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4733–4742.

[19] Carreira, J., Caseiro, R., Batista, J., and Sminchisescu, C. (2012). Semantic segmentation with second-order pooling. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 430–443.

[20] Catalin Ionescu, Fuxin Li, C. S. (2011). Latent structured models for human pose estimation. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 2220–2227.

[21] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2015). Semantic image segmentation with deep convolutional nets and fully connected crfs. In *Proceedings of International Conference on Learning Representations (ICLR)*.

[22] Chen, W., Wang, H., Li, Y., Su, H., Wang, Z., Tu, C., Lischinski, D., Cohen-Or, D., and Chen, B. (2016). Synthesizing training images for boosting human 3d pose estimation. In *Proceedings of Conference on 3D Vision (3DV)*, pages 479–488.

[23] Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., and Yuille, A. (2014). Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1971–1978.

[24] Chen, Y., Kim, T.-K., and Cipolla, R. (2010). Inferring 3d shapes and deformations from single views. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 300–313.

[25] Chernyaev, E. (1995). Marching cubes 33: Construction of topologically correct isosurfaces. Technical report, Institute for High Energy Physics.

[26] Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S. (2016). 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 628–644.

[27] Cootes, T., Edwards, G., and Taylor, C. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(6):681–685.

[28] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223.

[29] Dai, P., Wang, X., and Zhang, W. (2018). Coarse-to-fine multiview 3d face reconstruction using multiple geometrical features. *Multimedia Tools and Applications*, 77(1):939–966.

[30] Dollár, P., Welinder, P., and Perona, P. (2010). Cascaded pose regression. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1078–1085.

[31] Dou, P. and Kakadiaris, I. A. (2018). Multi-view 3d face reconstruction with deep recurrent neural networks. *Image and Vision Computing*, 80:80–91.

[32] E. Sánchez-Lozano, B. Martinez, G. T. and Valstar, M. (2016). Cascaded continuous regression for real-time incremental face tracking. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 645–661.

[33] Eslami, S. and Williams, C. (2012). A generative model for parts-based object segmentation. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 100–107.

[34] Eslami, S. A., Heess, N., Williams, C. K., and Winn, J. (2014). The shape boltzmann machine: a strong model of object shape. *International Journal of Computer Vision (IJCV)*, 107(2):155–176.

[35] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338.

[36] Ghezelghieh, M. F., Kasturi, R., and Sarkar, S. (2016). Learning camera viewpoint using cnn to improve 3d body pose estimation. In *Proceedings of International Conference on 3D Vision (3DV)*, pages 685–693.

[37] Grest, D., Herzog, D., and Koch, R. (2005). Human model fitting from monocular posture images. In *Proceedings of Vision Modeling and Visualization (VMV)*, pages 665–1344.

[38] Guan, P., Weiss, A., Balan, A. O., and Black, M. J. (2009). Estimating human shape and pose from a single image. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 1381–1388.

[39] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

[40] Hinton, G., Srivastava, N., and Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent.

[41] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[42] Hu, R., Dollár, P., He, K., Darrell, T., and Girshick, R. (2018). Learning to segment every thing. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4233–4241.

[43] Huber, P., Hu, G., Tena, R., Mortazavian, P., Koppen, W. P., Christmas, W., Rätsch, M., and Kittler, J. (2016). A multiresolution 3d morphable face model and fitting framework. In *Proceedings of International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*.

[44] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

[45] Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. (2014). Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 36(7):1325–1339.

[46] 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 *Proceedings of International Conference on Computer Vision (ICCV)*, pages 1031–1039.

[47] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

[48] Jiang, H. (2010). 3d human pose reconstruction using millions of exemplars. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 1674–1677.

[49] Jiang, L., Wu, X., and Kittler, J. (2018a). Pose invariant 3d face reconstruction. *arXiv preprint arXiv:1811.05295*.

[50] Jiang, L., Zhang, J., Deng, B., Li, H., and Liu, L. (2018b). 3d face reconstruction with geometry details from a single image. *IEEE Transactions on Image Processing*, 27(10):4756–4770.

[51] Joo, H., Simon, T., and Sheikh, Y. (2018). Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *Proceedings of Confernence Computer Vision and Pattern Recognition (CVPR)*, pages 8320–8329.

[52] Jourabloo, A. and Liu, X. (2016). Large-pose face alignment via cnn-based dense 3d model fitting. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4188–4196.

[53] Kae, A., Sohn, K., Lee, H., and Learned-Miller, E. (2013). Augmenting crfs with boltzmann machine shape priors for image labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2019–2026.

[54] Kemelmacher-Shlizerman, I. and Basri, R. (2011). 3d face reconstruction from a single image using a single reference face shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(2):394–405.

[55] Kemelmacher-Shlizerman, I. and Seitz, S. M. (2011). Face reconstruction in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 1746–1753.

[56] Koestinger, M., Wohlhart, P., Roth, P. M., and Bischof, H. (2011). Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark

localization. In *First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, pages 2144–2151.

[57] Kowalski, M., Naruniec, J., and Trzcinski, T. (2017). Deep alignment network: A convolutional neural network for robust face alignment. In *Proceedings of Conference on Computer Vision & Pattern Recognition (CVPR-W), Faces-in-the-wild Workshop/Challenge*, pages 88–97.

[58] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105.

[59] Le, V., Brandt, J., Lin, Z., Bourdev, L., and Huang, T. S. (2012). Interactive facial feature localization. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 679–692.

[60] Li, S. and Chan, A. B. (2014). 3d human pose estimation from monocular images with deep convolutional neural network. In *Proceedings of Asian Conference on Computer Vision (ACCV)*, pages 332–347.

[61] Li, Y., Su, H., Qi, C. R., Fish, N., Cohen-Or, D., and Guibas, L. J. (2015). Joint embeddings of shapes and images via cnn image purification. *ACM Transactions on Graphics (TOG)*, 34(6):234.

[62] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 740–755.

[63] Liu, F., Zeng, D., Zhao, Q., and Liu, X. (2016). Joint face alignment and 3d face reconstruction. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 545–560.

[64] Liu, S., Yang, J., Huang, C., and Yang, M.-H. (2015). Multi-objective convolutional learning for face labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3451–3459.

[65] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440.

[66] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. (2015). Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248.

[67] Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of ACM Siggraph Computer Graphics*, pages 163–169.

[68] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110.

[69] Lowe, D. G. et al. (1999). Object recognition from local scale-invariant features. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 1150–1157.

[70] Luo, P., Wang, X., and Tang, X. (2012). Hierarchical face parsing via deep learning. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2480–2487.

[71] Mahpod, S., Das, R., Maiorana, E., Keller, Y., and Campisi, P. (2018). Facial landmark point localization using coarse-to-fine deep recurrent neural network. *arXiv preprint arXiv:1805.01760*.

[72] Mayo, M. and Zhang, E. (2009). 3d face recognition using multiview keypoint matching. In *Proceedings of International Conference on Advanced Video and Signal Based Surveillance*, pages 290–295.

[73] Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H.-P., Xu, W., Casas, D., and Theobalt, C. (2017). Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):44.

[74] Merget, D., Rock, M., and Rigoll, G. (2018). Robust facial landmark detection via a fully-convolutional local-global context network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 781–790.

[75] Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 483–499.

[76] Noh, H., Hong, S., and Han, B. (2015). Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528.

[77] Park, S., Hwang, J., and Kwak, N. (2016). 3d human pose estimation using convolutional neural networks with 2d pose information. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 156–169.

[78] Pavlakos, G., Zhou, X., Derpanis, K. G., and Daniilidis, K. (2017). Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7025–7034.

[79] Paysan, P., Knothe, R., Amberg, B., Romdhani, S., and Vetter, T. (2009). A 3d face model for pose and illumination invariant face recognition. In *Proceedings of Asian Conference on the Social Sciences (AVSS)*, pages 296–301.

[80] Pfister, T., Charles, J., and Zisserman, A. (2015). Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1913–1921.

[81] Piotraschke, M. and Blanz, V. (2016). Automated 3d face reconstruction from multiple images using quality measures. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3418–3427.

[82] Pons-Moll, G., Romero, J., Mahmood, N., and Black, M. J. (2015). Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics*, 34(4):120:1–120:14.

[83] Qiu, Z., Yao, T., and Mei, T. (2017). Learning spatio-temporal representation with pseudo-3d residual networks. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 5533–5541.

[84] Ramakrishna, V., Kanade, T., and Sheikh, Y. (2012). Reconstructing 3d human pose from 2d image landmarks. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 573–586.

[85] Richardson, E., Sela, M., and Kimmel, R. (2016). 3d face reconstruction by learning from synthetic data. In *Proceedings of Conference on 3D Vision (3DV)*, pages 460–469.

[86] Richardson, E., Sela, M., Or-El, R., and Kimmel, R. (2017). Learning detailed face reconstruction from a single image. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1259–1268.

[87] Richter, S. R., Vineet, V., Roth, S., and Koltun, V. (2016). Playing for data: Ground truth from computer games. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 102–118.

[88] Romdhani, S. and Vetter, T. (2005). Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 986–993.

[89] Roth, J., Tong, Y., and Liu, X. (2016). Adaptive 3d face reconstruction from unconstrained photo collections. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4197–4206.

[90] Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. (2013a). 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of International Conference on Computer Vision Workshops (ICCV-W)*, pages 397–403.

[91] Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. (2013b). A semi-automatic methodology for facial landmark annotation. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 896–903.

[92] Sela, M., Richardson, E., and Kimmel, R. (2017). Unrestricted facial geometry reconstruction using image-to-image translation. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 1576–1585.

[93] Sigal, L., Balan, A., and Black, M. J. (2008). Combined discriminative and generative articulated pose and non-rigid shape estimation. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 1337–1344.

[94] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[95] Sun, Y., Wang, X., and Tang, X. (2013a). Deep convolutional network cascade for facial point detection. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3476–3483.

[96] Sun, Z.-L. and Lam, K.-M. (2011). Depth estimation of face images based on the constrained ica model. *IEEE Transactions on Information Forensics and Security*, 6(2):360–370.

[97] Sun, Z.-L., Lam, K.-M., and Gao, Q.-W. (2013b). Depth estimation of face images using the nonlinear least-squares model. *IEEE Transactions on Image Processing*, 22(1):17–30.

[98] Suwajanakorn, S., Kemelmacher-Shlizerman, I., and Seitz, S. M. (2014). Total moving face reconstruction. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 796–812.

[99] Tekin, B., Katircioglu, I., Salzmann, M., Lepetit, V., and Fua, P. (2016a). Structured prediction of 3d human pose with deep neural networks. *arXiv preprint arXiv:1605.05180*.

[100] Tekin, B., Rozantsev, A., Lepetit, V., and Fua, P. (2016b). Direct prediction of 3d body poses from motion compensated sequences. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 991–1000.

[101] Tompson, J., Goroshin, R., Jain, A., LeCun, Y., and Bregler, C. (2015). Efficient object localization using convolutional networks. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 648–656.

[102] Tompson, J. J., Jain, A., LeCun, Y., and Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 1799–1807.

[103] Toshev, A. and Szegedy, C. (2014). Deeppose: Human pose estimation via deep neural networks. In *Proceedings of conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1653–1660.

[104] Tran, A. T., Hassner, T., Masi, I., Paz, E., Nirkin, Y., and Medioni, G. (2018). Extreme 3d face reconstruction: Seeing through occlusions. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR-W)*, pages 3935–3944.

[105] Tsogkas, S., Kokkinos, I., Papandreou, G., and Vedaldi, A. (2015). Deep learning for semantic part segmentation with high-level guidance. *arXiv preprint arXiv:1505.02438*.

[106] Tzimiropoulos, G. (2015). Project-out cascaded regression with an application to face alignment. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3659–3667.

[107] Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., and Schmid, C. (2017). Learning from synthetic humans. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 109–117.

[108] Warrell, J. and Prince, S. J. (2009). Labelfaces: Parsing facial features by multiclass labeling with an epitome prior. In *Proceedings of International Conference on Image Processing (ICIP)*, pages 2481–2484.

[109] Xiong, X. and De la Torre, F. (2013). Supervised descent method and its applications to face alignment. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 532–539.

[110] Yang, Y. and Ramanan, D. (2011). Articulated pose estimation with flexible mixtures-of-parts. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1385–1392.

[111] Zanfir, M., Popa, A.-I., Zanfir, A., and Sminchisescu, C. (2018). Human appearance transfer. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5391–5399.

[112] Zhang, N., Shelhamer, E., Gao, Y., and Darrell, T. (2015). Fine-grained pose prediction, normalization, and recognition. *arXiv preprint arXiv:1511.07063*.

[113] Zhang, X., Yin, L., Cohn, J. F., Canavan, S., Reale, M., Horowitz, A., and Liu, P. (2013). A high-resolution 3d dynamic facial expression database. In *Proceedings of Automatic Face and Gesture Recognition (FG)*, pages 1–6.

[114] Zhang, Z., Luo, P., Loy, C. C., and Tang, X. (2014). Facial landmark detection by deep multi-task learning. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 94–108. Springer.

[115] Zhao, M., Li, T., Alsheikh, M. A., Tian, Y., Zhao, H., Torralba, A., and Katabi, D. (2018). Through-wall human pose estimation using radio signals. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7356–7365.

[116] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. (2015). Conditional random fields as recurrent neural networks. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1529–1537.

[117] Zhou, X., Sun, X., Zhang, W., Liang, S., and Wei, Y. (2016a). Deep kinematic pose regression. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 186–201.

[118] Zhou, X., Zhu, M., Leonardos, S., Derpanis, K. G., and Daniilidis, K. (2016b). Sparseness meets deepness: 3d human pose estimation from monocular video. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4966–4975.

[119] Zhu, S., Li, C., Change Loy, C., and Tang, X. (2015). Face alignment by coarse-to-fine shape searching. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4998–5006.

[120] Zhu, X., Lei, Z., Liu, X., Shi, H., and Li, S. Z. (2016). Face alignment across large poses: A 3d solution. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 146–155.

[121] Zhu, X. and Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2879–2886.