Imperial College London

Department of Computing

Deep Deformable Models for 3D Human Body

Author:
Haoyang Wang

Supervisor:
Stefanos Zafeiriou

Submitted in partial fulfillment of the requirements for the Doctor of Philosophy in Computing and Diploma of Imperial College London

I confirm that this thesis is my own work and that all else is appropriately referenced.

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Abstract

Deformable models are powerful tools for modelling the 3D shape variations for a class of objects. However, currently the application and performance of deformable models for human body are restricted due to the limitations in current 3D datasets, annotations, and the model formulation itself. In this thesis, we address the issue by making the following contributions in the field of 3D human body modelling, monocular reconstruction and data collection/annotation.

Firstly, we propose a deep mesh convolutional network based deformable model for 3D human body. We demonstrate the merit of this model in the task of monocular human mesh recovery. While outperforming current state of the art models in mesh recovery accuracy, the model is also light weighted and more flexible as it can be trained end-to-end and fine-tuned for a specific task.

A second contribution is a bone level skinned model of 3D human mesh, in which bone modelling and identity-specific variation modelling are decoupled. Such formulation allows the use of mesh convolutional networks for capturing detailed identity specific variations, while explicitly controlling and modelling the pose variations through linear blend skinning with built-in motion constraints. This formulation not only significantly increases the accuracy in 3D human mesh reconstruction, but also facilitates accurate in the wild character animation and retargetting.

Finally we present a large scale dataset of over 1.3 million 3D human body scans in daily clothing. The dataset contains over 12 hours of 4D recordings at 30 FPS, consisting of 7566 dynamic sequences of 3D meshes from 4205 subjects. We propose a fast and accurate sequence registration pipeline which facilitates markerless motion capture and automatic dense annotation for the raw scans, leading to automatic synthetic image and annotation generation that boosts the performance for tasks such as monocular human mesh reconstruction.

Acknowledgements

I would like to express my sincere gratitude to everyone who has supported and helped me in my PhD journey.

First of all, all of this will not be possible without my supervisor Prof. Stefanos Zafeiriou. I first met Stefanos in his course when I was a master student, after which I have completed my master thesis under his supervision. His passion and extensive knowledge in the subject has inspired me to pursue a research degree. 3D human body modelling and its relevant applications was a completely new topic to me when I started my PhD, and without Stefanos' consistent support and guidance I would have never gained the confidence and this level of understanding in my research. It has been an honour for me to be able to work with him.

I would also like to say a big thank you to Dr. Riza Alp Güler. I worked with Alp closely on almost all of my projects, especially for the museum data and the BLSM paper. The countless discussions we had on the technical details and the experience of sitting together to implement difficult pieces of the project has benefited me both in the research perspective and in my future career.

Moreover, I would like to thank everyone from the iBUG group for the inspiring discussions and collaborations we had, and every member and alumni of Huxley 302 for all the fun times we had together.

Apart from my Imperial colleagues, I would also like to thank Prof. Iasonas Kokkinos and Dr. George Papandreou who have offered their priceless help on the BLSM paper; as well as everyone from Ariel AI, with whom I have spent most of my time with virtually since the pandemic began. The working from home experience became less lonely with you guys.

Furthermore, I would like to say thank you to my friends Yuliya Gitlina, Shuang Xia, Yi-Ling Liu and Chen Chen. The tea breaks we had in Huxley and the weekend adventures we had in London had been the most enjoyable times I had outside my research life. I have also received a lot of support from my friend Xi Chen, despite the distance, we have been through so many happy or difficult times together, which

I will never forget.

I would have not became who I am without the unconditional trust and continuous support of my parents. Thank you for supporting me for every decision that I have made in my life, and believing in me in even the most difficult situations.

Last but not least, I would like to thank Dimitris for his love and support. Thank you for being there for me in the extremely difficult circumstances and patiently absorbing all my negative energies. I feel very lucky to have you by my side.

Contents

1 Introduction **15**

1.1 Motivation 15

1.2 Contribution 16

1.2.1 Reparameterising 3D Morphable Models 16

1.2.2 Single Image 3D Reconstruction with Mesh Convolutions 17

1.2.3 A Bone-Level Skinned Model of the Human Mesh 18

1.2.4 MeDigital: A Large Scale 4D Dataset Of Human Body 18

1.3 Publications 19

2 Background **20**

2.1 3D Data Representation 20

2.2 Parametric Models of Human Bodies 22

2.2.1 Model Formulation 22

2.2.2 Model Training and Data Pre-processing 30

2.3 Applications 33

3 Reparameterising 3D Morphable Models **36**

3.1 Introduction 36

3.2 Related Work 37

3.3 Method 38

3.3.1 Problem Formulation 38

3.3.2 Computing the **W** matrix 39

3.3.3 Blend Skinning Models 39

3.4 Evaluation 40

3.4.1 PCA-based Shape Models 41

3.4.2 Blend Skinning Models 43

3.5 Conclusion 44

4 Single Image 3D Human Body Reconstruction with Mesh Convolutions 46
4.1 Introduction . . . 47
4.2 Related Work . . . 49
4.3 Mesh Convolutional Networks . . . 51
4.3.1 Spiral Convolution . . . 51
4.3.2 Mesh Pooling . . . 53
4.3.3 Mesh Convolutional Autoencoder . . . 54
4.4 3D Reconstruction of Human Body from Single Image . . . 55
4.5 Evaluation . . . 57
4.5.1 Implementation Details . . . 58
4.5.2 Experimental Setup . . . 58
4.5.3 Results and Discussion . . . 61
4.5.4 Results on in the wild images . . . 65
4.6 Conclusion . . . 66

5 BLSM: A Bone-Level Skinned Model of the Human Mesh 69
5.1 Introduction . . . 69
5.2 Related Work . . . 72
5.3 Bone-Level Skinned Model . . . 73
5.3.1 Skeleton Modeling . . . 73
5.3.2 Template Synthesis . . . 76
5.3.3 Linear Blend Skinning . . . 77
5.4 Model training . . . 78
5.4.1 Unconstrained Landmark-based Alignment . . . 78
5.4.2 Bone Basis and Bone-corrective Blendshapes . . . 82
5.4.3 Shape Blendshapes . . . 83
5.4.4 Blending Weights . . . 83
5.5 Evaluation . . . 84
5.5.1 Implementation Details . . . 84
5.5.2 Quantitative Evaluation . . . 84
5.5.3 Qualitative Evaluation . . . 89
5.6 Conclusion and Future Work . . . 90

6 MeDigital: A Large Scale 4D Dataset Of Human Body 94
6.1 Introduction . . . 95
6.2 Dataset Overview . . . 97

6.2.1 Data Acquisition 97
6.2.2 Comparison to Existing Datasets 99
6.3 Registration 101
6.3.1 Stage 1: 3D Keypoints Based Alignment 102
6.3.2 Stage 2: Shape Initialisation with Sparse Correspondences . . . 103
6.3.3 Shape Refinement Based on Multiview Dense Correspondences . 103
6.3.4 Stage 3: Refining Surface Details 104
6.4 Evaluation 105
6.4.1 Registration Quality 105
6.4.2 Attribute-driven 3D Mesh Synthesis 108
6.4.3 Single Image Mesh Recovery with Synthetic Training 111
6.5 Conclusion 115

7 Conclusion **120**
7.1 Summary 120
7.2 Future Work 122
7.2.1 High Fidelity Clothed Human Modelling 122
7.2.2 Robust Registration of Scans 123
7.2.3 Monocular Human Mesh Recovery 124

List of Figures

2.1 Illustration of a typical binary voxel occupancy encoding (used by [1]), where for the completed shape representation, the a voxel is either with in the observed surface or free. The side view marked as the yellow slice in (2) is used for visualisation in (3). 21

2.2 An example of deformation transfer using similar approach as the SCAPE model. Deformation of the mesh is modeled as per-triangle transformations. Simply solving the new mesh with neighbouring triangle smoothness constraint gives a discontinued mesh B. Enforcing the shared vertices to be transformed to the same location (or computing the new vertices locations to be as close as possible to B as in the SCAPE model) gives a continued smooth surface C [2]. 24

2.3 A blend skinning human body model proposed by [3]. Given a rigged template mesh (a), the shape and pose-dependent variations are added to template as vertex offsets, and the joint location is updated (b and c), then linear blend skinning is applied to animate the updated template to desired pose. 25

2.4 A human body shape space learned by PCA [3], the first two principal components are visualised, varying from -2 to +2 standard deviation . 26

2.5 Shell PCA compared to Euclidean PCA on human body pose modelling [4]. PCA in non-linear manifolds better captures the non-linear nature of human body articulations comparing to PCA in Euclidean space. . . 27

2.6 A template mesh deformed by the samples of the GPMM with a Gaussian kernel [5]. Meshes synthesised from GPMMs are not necessarily valid shapes, but they are flexible enough and could benefit downstream optimisation steps. 28

2.7 Mesh convolutional based models struggle to model large articulated poses (top row: COMA [6]). The result is improved when the number of learnable weights in the network is increased (bottom row: Neural 3DMM [7]), however the reconstruction error is still high for some parts. 29

2.8 Network structure of implicit function based model used in [8]. Given a feature vector, the network outputs a binary value indicating if the queried point is inside or outside the object. Surface is reconstructed by sampling points in the 3D space and query with the network. 31

3.1 Visualisation of the first three principal components at -3 and +3 standard deviations. The red meshes indicate the original model, while the gray meshes indicate the new model. 41

3.2 Compactness of the Original and Reparameterised Models. 42

3.3 Generalisation Error of the Original and Reparameterised Models. . . . 42

3.4 Specificity Error of the Original and Reparameterised Models. 43

3.5 Visualisation of the animated blend skinning models. Red meshes: animation result of the original model with 6890 vertices. Gray meshes: animation result of the reparameterised model with 10412 vertices. . . . 45

4.1 Overview of our proposed approach. Given an input RGB image, our network outputs a 3D mesh reconstruction of the person in the image through an encoder-decoder structure. During training, the 3D mesh reconstruction loss and 2D keypoint reprojection loss is minimised. We also use a mesh autoencoder based discriminator network during training to ensure that the image to mesh network outputs plausible human bodies. 47

4.2 Our choice of reference mesh and the downsampled meshes with factor of 4, 2, 2, 2. By posing the template mesh as illustrated, vertices around the joints are preserved after down-sampling, subsequently helping the network to learn human body articulations in a coarse level. 54

4.3 Illustration of our mesh convolutional autoencoder architecture. 54

4.4 Illustration of our image to mesh network architecture. A CNN is used to encode image features given a RGB image. The image features are then sent to two separate fully connected network modules to compute the mesh embedding and camera embedding. The 3D mesh is then decoded from the mesh embedding with the spiral mesh decoder, and projected back to the image with the predicted camera parameters. . . 56

4.5 Left two columns: results from network D (our image to mesh network). Right two columns: results on the same image from network A (SMPL baseline). 63

4.6 Left two columns: results from network F (our proposed approach). Right two columns: results on the same image from network B (HMR). 64

4.7 Qualitative results on Human3.6M Protocol 1 testset (1st row) and UP-3D testset by error quartile in terms of per-vertex mean reconstruction error. The columns show examples from different error quartiles, from left to right: 0-25%, 25-50%, 50-75%, 75-100%. 66

4.8 Qualitative results from network Config F (with mesh decoder pretraining) and Config E (training from scratch). From left to right: results of Config F network overlaid on the input image, Config F result from a different angle, results of Config E network, Config E result from a different angle. 68

5.1 Overview of our Bone-Level Skinned Model (BLSM): The top row shows skeleton synthesis: starting from a canonical, bind pose, we first scale the bone lengths and then apply an articulated transformation. The bottom row shows shape control: the canonical mesh template is affected by the bone scaling transform through Bone-Scaling Blend Shapes, and then further updated to capture identity-specific shape variation. The skeleton drives the deformation of the resulting template through Linear Blend Skinning, yielding the posed shape. 70

5.2 7 out of the 47 degrees of freedom corresponding to kinematically feasible joint rotations for our skeleton. 75

5.3 Impact of bone length variation on the template. Plain linear blend skinning results in artifacts. The linear, bone-corrective blendshapes eliminate these artifacts, and capture correlations of bone lengths with gender and body type. 76

5.4 Example of one registered CAESAR instance. We first fit our rigged template to the landmarks on the scan surface by optimising over the joint angles and bone scales, then non-rigidly deform the vertices freely to align the scan surface. 79

5.5 Comparison between mutual nearest neighbour (MNN) loss and chamfer loss. Red represents points on the scan, and blue for points on the model. Left: MNN loss. Right: chamfer loss. With chamfer loss the blue points would collapse around the edge of holes, while MNN loss only considers more confident correspondences. 80

5.6 Mean absolute vertex error on the CAESAR dataset (left) and our in-house testset (right) against number of shape coefficients. 85

5.7 Shape generalisation error (left) and pose generalisation error (right) on D-FAUST dataset against number of shape coefficients. 87

5.8 Mean absolute vertex error and example of reconstructions on the test-set. Left to right: SMPL-reimpl, BLSM-Linear, BLSM-Spiral. For linear models we show result with 125 coefficients allowed. For BLSM-spiral the latent size is 128. 88

5.9 Mean absolute vertex error of gender specific models on the CAESAR dataset (left) and DFAUST dataset (right) against number of shape coefficients. 89

5.11 Image-driven character animation: we rig two characters from [9] using our model's bone structure. This allows us to transform any person into these characters, while preserving the pose and body type of the person in the image. 89

5.10 Samples from reconstructions of D-Faust and our testset. Top to bottom: ground truth, SMPL-reimpl (125), BLSM-lienar (125), BLSM-spiral (128) 91

5.12 Pixel-accurate, image-driven character animation in-the-wild. 92

5.13 Linear (top two rows) vs. graph convolutional (bottom two rows) modeling of shape variation. 93

6.1 Sample scans from our large scale 4D scan dataset. The dataset contains over 1.3 million 3D scans of human body with high resolution textures, capturing with over 4200 identities and 7500 dynamic sequences, the dataset surpasses most existing scan datasets in terms of subject and pose varieties, as well as texture quality. 94

6.2 The data capturing system consists of 14 Modular Camera Units of 42 cameras synchronised with a lighting system of 12 LED panels. 97

6.3 An example scan frame from our dataset. At the top row we show the geometry of the scan, the textured scan and the UV parameterization. At the bottom row, we show the collage of raw captured images of this frame, and wireframe of the scan. 98
6.4 Subjects age, gender, height, weight, and ethinic group distribution of our dataset and CAESAR. 100
6.5 Our proposed registration pipeline. 101
6.6 (a) Example of a scan instance where vertices at bottom of the feet are missing. (b) BLSM fitting result at stage 2. Wrong correspondences are established due to bad initialisation from stage 1. (c) DensePose result from multiple views. (d) BLSM fitting result with DensePose reprojection loss. 104
6.7 Registration result of one example frame. (a) Triangles with area larger than a certain threshold is omitted to deal with missing parts and noise in the scans. (b) Registration result with pointcloud evenly sampled from the scan surface. (c) Registration result with pointcloud sampled from triangles smaller than $5 \times 10^{-4}\ m^3$ 105
6.8 Left: Error heatmaps on the multi-subject (top row) and multi-pose (bottom row) subset visualised on the mean BLSM template. Colors are shown in millimetres. Right: cumulative per-vertex error plot. Blue line: multi-subject subset, red line: multi-pose subset. 106
6.9 Visualisation of some example frames from the multi-pose subset where the registration error is large. Samples are from protocol 15, 16, 12, 02 respectively. For each frame we show: raw scan with texture (with face of the subject occluded), registration result, error heatmap visualised on the registration overlaid with the raw scan from front and back view. 107
6.10 First 3 bone bases and first 5 shape bases from -3σ to $+3\sigma$ 109
6.11 Variations of synthetic image generation pipelines using our scan dataset. 112
6.12 Samples from our synthetic image dataset. 113
6.13 Qualitative results on LSP dataset. For each example in the figure: left: input image, middle: result from **real + synthetic** images trained network, right: result from **real** images trained network. Qualitatively the network trained with real + synthetic images outperforms the network trained with only real images. 115
6.14 Shapes synthesised by regressing from the input attribute values to BLSM shape parameters. 117

6.15 Shapes synthesised by regressing the PCA coefficents of the input attribute values to the BLSM shape parameters. 118
6.16 Shapes synthesised with a conditional GAN. 119

List of Tables

2.1 Comparison of different types of parametric models for 3D human body 35

4.1 Architecture of our mesh convolutional autoencoder. 55

4.2 Our ablative experiments setup. 'AE' refers to our proposed mesh convolutional autoencoder. 'Decoder' refers to the mesh convolutional decoder. 'IEF' refers to the iterative error feedback component used in HMR[10]. 'BEGAN adv' refers to our proposed dense mesh autoencoder adversarial loss. 59

4.3 Mean joint reconstruction errors and mean vertex reconstruction errors on Human3.6M dataset for the configurations of our ablation study described in Table 4.2. Using the proposed convolutional mesh decoder, simple mesh regression training (**Config C**) performs better compared to linear blend skinning based models (**Config A** and **B**). Adversarial training with mesh autoencoder further improves the results (**Config E** and **F**). 61

4.4 Comparison to state-of-the-art on UP-3D dataset and Human3.6M dataset (Protocol 2). Errors are measured in millimetres. (*) indicates that the vertex loss is measured on a sparse set of points (landmark or keypoint loss). 66

5.1 Reconstruction error on groups of subjects with different BMI value, height, weight and gender 86

5.2 Generalisation error and AUC for cumulative error distribution on our in-house testset. 88

6.1 List of motion protocols and number of sequences captured for each protocol. The protocols were designed to capture most of the feasible human joint motions. 99

6.2 Overview of existing 3D/4D human body datasets 99

6.3 Mean vertex to scan registration error in (mm) on the multi-subject and multi-pose subset of the dataset. Protocols (01-23) are as listed in Table 6.1. Here (*) indicates that sequences of this protocol contains two persons. 106
6.4 Joint reconstruction results on UP-3D dataset and segmentation results on LSP dataset. Segmentation results are evaluated for both foreground vs background segmentation as well as part segmentations (6 parts + background). Network trained on synthetic + real image outperforms the network trained only with real images on all tasks. 114

Chapter 1

Introduction

1.1 Motivation

Human understanding from images has been one of the core problems of computer vision due to its wide range of applications in human computer interaction. Since the recent development of deep learning methods and hardware, the level of perception of human bodies in images has progressed from detection, classification and keypoints localisation in 2D to 3D pose estimation and surface reconstruction, facilitating applications such as monocular motion capture, 3D full body avatar digitisation, character animation and virtual/augmented reality. Leveraging 3D representations from 2D images however remains a challenging problem due to the ambiguities introduced while projecting a 3D object to the 2D image plane.

Many works have attempted to tackle the problem with deep neural networks, where a mapping from input images and the human joints in 3D is learnt. The limitation of such model free method is the lack of prior knowledge of the human body structure, which could lead to implausible reconstructions in some challenging cases. A model free reconstruction of the full 3D surface is even more challenging as the degree of freedom increases.

Deformable models, or parametric models, incorporate prior knowledge about objects, restricting the generated shapes to lie within a plausible space. Such models have been used widely in the task of 3D pose estimation as the problem could be reduced to predicting the model parameters. Moreover, a deformable model provides a surface reconstruction while solving the pose estimation problem alone, which can be further refined for high fidelity surface reconstruction with additional supervision signals. The

resulting parameterisation of the human in the image can be used for motion and texture transfer, which is not straightforward with a model free approach.

However, building compact and realistic 3D parametric models of human body is challenging. Existing 3D datasets of human body scans are either captured in restricted conditions or have small number of participants and pose variations, leading to models with limited representation power. Moreover, manual annotation of 3D datasets requires a lot of labour, therefore there does not exist a dataset with reliable ground truth registrations, subsequently the learned parametric model suffers from noise and misalignment of the vertices, leading to insufficient representation of details.

Texture model is also a important component of deformable models, however currently most of the human body models only consider the shape component due to the lack of high quality texture information in the existing 3D datasets. Datasets such as [11][12][13] capture human body in minimum clothing in different experimental setup, as a result, texture models built with these datasets can only be used to improve registration results within the datasets, and cannot be used for in the wild tasks, such as detailed surface reconstruction from images.

1.2 Contribution

Motivated by the aforementioned problems, we make the following contributions in this thesis.

1.2.1 Reparameterising 3D Morphable Models

Building parametric models of 3D human bodies remains a difficult problem due to limitations of dataset, realiability of registration method and computational resources. Here we consider linear shape models which is built by performing Principal Component Analysis (PCA) on a set of registrations aligned to a pre-defined template, and linear blend skinning models for human bodies. Once a shape model is trained, the topology of the instances in the model space is then fixed. However, in many applications, a model with a different topology might be useful. High resolution models can be used for synthesising high fidelity meshes, while low resolution models could reduce the computational overhead in downstream applications. For these use cases, one may prefer to reuse a model instead of repeating the model training pipeline with another template, since it could be quite demanding in terms of computational time.

In other cases it would not even be possible because access to the original training data is prohibited.

In Chapter 3 of the thesis, we propose a simple yet efficient method of reparameterising statistical shape models given a new template of a different topology. The proposed method is based on the probabilistic nature of statistical shape models. Given a model and a new template, we solve for a covariance matrix for the new model directly without using or generating any training data. We provide both qualitative and quantitative evaluation, demonstrating that our proposed method is able to reparameterise models while preserving the surface details with no information loss in terms of intrinsic properties of the model. This is particularly useful while comparing models of different topology, as we demonstrate in Chapter 5.

1.2.2 Single Image 3D Reconstruction with Mesh Convolutions

In Chapter 4 we propose a method for recovering 3D representation of human body from single RGB image. Previous works rely on the use of a parametric model of human body, either in the form of optimisation based fitting method, or deep learning based method that regress the model parameters directly from the input image. In both cases, supervision with model parameters are required for plausible and robust reconstruction results, requiring extra learning or fitting steps for preparing the pose priors.

We propose a network structure that operates directly on 3D mesh vertices instead. Given an input RGB image, our network compute a latent code through an image encoder, which is then sent to a mesh convolutional decoder that outputs the 3D vertex locations. During training, we use a mesh convolutional autoencoder based discriminator network to enforce plausible reconstructions, which lead to significantly better results compared to the direct vertex regression method through mesh convolutions. We perform quantitative evaluations on the task of 3D pose estimation, and show that our proposed method outperforms comparable linear blend skinning model based methods. We also show our results on in-the-wild images, demonstrating the effectiveness and potential of our proposed approach.

1.2.3 A Bone-Level Skinned Model of the Human Mesh

Currently a common approach of data-driven parametric models of rigged mesh representations for 3D human body first synthesise the template mesh in a canonical pose, then estimate skeleton joints post-hoc by regressing from the synthesised mesh. We aim at increasing the accuracy of modelling by revisiting the template synthesis process prior to rigging. Our major contribution in Chapter 5 consists of disentangling the modelling of bone length variability from acquired body traits dependent. This also facilitates modelling of human body with clothes, for which the bone structure of the modelled body does not depend on the clothes they wear.

We further control and strengthen the individual components of the model: Firstly, we constrain joint angles to respect the kinematic constraints of human body, reducing body motion to 47 pose atoms. Secondly, we introduce accurate mesh convolution-based networks to capture identity-specific surface variation. We show that these largely outperform their linear basis counterparts, demonstrating for the first time the merit of mesh convolutions in rigged full-body modelling.

We provide quantitative results on the problem of reconstructing a collection of 3D human scans, and show that we obtain systematic gains in average vertex reconstruction accuracy when comparing to a SMPL-type baseline. Beyond quantitative evaluation, we also show that our decoupled bone and shape representation facilitates accurate character animation in-the-wild.

1.2.4 MeDigital: A Large Scale 4D Dataset Of Human Body

Existing data-driven human body models are restricted to model human body shapes under clothes due to the lack of high quality 3D scan data of clothed human bodies. Thus beyond the above contributions with regards to modelling methods, we also present the first ever large scale 4D dataset of clothed human body scans. The dataset consists of 1.3 million scans together with high resolution texture maps. Over 4000 adults and children spanning different age, body type and ethinic groups were captured in their daily clothes, resulting in over 7500 sequences with different motions covering a wide range of actions. We propose a robust and fast registration pipeline and evaluate the performance both qualitatively and quantitatively.

We further demonstrated two use cases of our dataset in the task of attribute driven mesh synthesis and synthetic image generation with automatic 3D annotations, which

improves the performance of model based 3D human mesh recovery in monocular images.

1.3 Publications

Publications related to this thesis

- Dominik Kulon, Haoyang Wang, Riza Alp Güler, Michael M. Bronstein, and Stefanos Zafeiriou. Single Image 3D Hand Reconstruction with Mesh Convolutions. In *30th British Machine Vision Conference 2019 (BMVC)*, 2019.

- Haoyang Wang, Riza Alp Güler, Iasonas Kokkinos, George Papandreou, and Stefanos Zafeiriou. BLSM: A Bone-Level Skinned Model of the Human Mesh. In *European Conference on Computer Vision (ECCV)*, 2020.

- Haoyang Wang and Stefanos Zafeiriou. Reparameterising 3D Statistical Shape Models. In *2019 14th IEEE International Conference on Automatic Face Gesture Recognition (FG 2019)*, 2019.

Other publications

- Stylianos Ploumpis, Evangelos Ververas, Eimear O'Sullivan, Stylianos Moschoglou, Haoyang Wang, Nick Pears, William A. P. Smith, Baris Gecer, and Stefanos Zafeiriou. Towards a complete 3D morphable model of the human head. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

- Stylianos Ploumpis, Haoyang Wang, Nick Pears, William A. P. Smith, and Stefanos Zafeiriou. Combining 3D Morphable Models: A Large Scale Face-And-Head Model. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Chapter 2

Background

In this chapter we present a literature review of parametric 3D human body models and it's applications. We first describe the commonly used data representation for 3D shapes, then we introduce different types of formulations that are commonly used for parametric modelling of 3D human bodies, as well as the challenges that arises when training such models. Finally we provide a brief overview of the applications of parametric models of human bodies in various problems.

2.1 3D Data Representation

Pointcloud and Mesh

3D shapes can be represented as an unordered set of points $\{\mathbf{P}_i | i = 1, 2, ..., N\}$ where each $\mathbf{P}_i$ is a vector (x, y, z) of Cartesian coordinate of the point. This can be represented as a matrix $\mathbf{P} \in \mathbb{R}^{N\times 3}$. While pointclouds can store the location of a large number of points, accessing the surface properties such as normals of the shape is not straightforward and requires searching for the neighbours for each point, leading to extra computational cost. Moreover, to represent a detailed shape, a large amount of points are needed and thus can be memory inefficient.

Another commonly used representation for 3D data is mesh. Meshes are defined as a graph $\mathcal{G} = (\mathbf{V}, \mathcal{E})$, where $\mathbf{V} \in \mathbb{R}^{N\times 3}$ is the set of N vertices, each represented by the Cartesian coordinates describing the location of the vertex; and $\mathcal{E}$ is the set of edges representing the connectivities between the vertices. The sets of closed edges give a set of faces, and each face can contain more than 2 edges. However the most commonly used setting is 3 edges per-face, resulting in triangle meshes.

Edges explicitly defined on meshes facilitates operations that explores neighbouring properties of vertices, as neighbourhood of a vertex can be obtained simply by indexing with the edge list. Moreover, faces defined by the edges can act as an approximation of the surface, thus reduce the number of points needed to represent an object as opposite to pointclouds.

Occupancy Grid

Occupancy grid is a 3D tensor where each element is a binary value indicating whether a unit cube (voxel) is filled. Fig. 2.1 shows a chair represented with occupancy grid, where size of the grid is $30 \times 30 \times 30$. Probablisic representation of occupancy grid is also possible [14]. The advantage of using occupancy grid is that deep learning methods defined on 2D images that use convolutional neural networks can be directly adopted to operate on 3D objects. However a huge amount of memory is required to represent an object with high resolution details and smooth surface. [15] attempted to use hierarchical octree representation to refine surface details of the modelled objects, however the resolution is still limited.

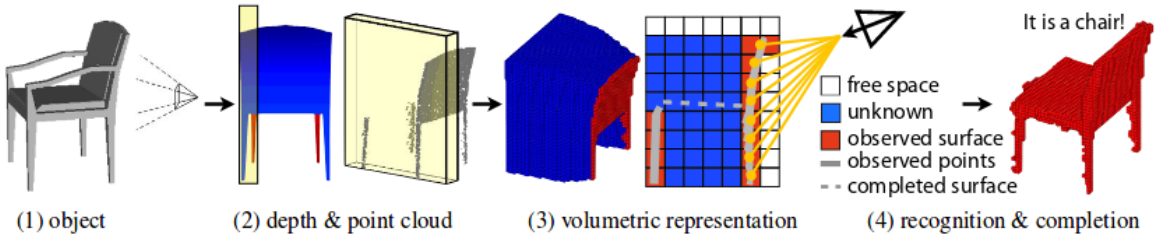


Figure 2.1: Illustration of a typical binary voxel occupancy encoding (used by [1]), where for the completed shape representation, the a voxel is either with in the observed surface or free. The side view marked as the yellow slice in (2) is used for visualisation in (3).

Implicit Representations

Except for the explicit representations that directly describes the shape in 3D space, there are also implicit representations that do not involve 3D coordinates. One example of such representations is shape descriptors. The property of shape descriptors depends on the definition of the specific shape descriptor that is used. However they are generally obtained from the object's geometry, topology, surface and other characteristics [16].

Another type of implicit representation of shape makes use of a function F which takes as input a latent vector, and estimate the inside/outside or signed-distance function given a query location in 3D space [17][18][8][19][20]. The shape is however implicitly described by the latent vector and the function, and it takes an extra step to reconstruct the surface of the object.

Other Representations

There exists other representations of 3D objects where the objects are stored in '2.5D' instead of actual 3D. Depth images and multiview representations are typical '2.5D' representations of 3D shapes. Depth images are typically combined with RGB images and are inexpensive to store and capture due to the popular RGB-D sensors. However depth maps do not represent the full 3D geometry of the object, and lack of topological information.

2D images from multiple views can also be used to represent a 3D object, however geometry of the 3D shape need to be learned or induced from the set of 2D images. Also the number of views that are needed is unclear, as small number of views might not capture the full 3D property of the object.

2.2 Parametric Models of Human Bodies

In this section we provide a literature survey on parametric modelling of human bodies. We start with explaining what is a parametric model, and describe the formulations of some popular human body models. We then give an overview of the general model training pipeline, emphasising the challenge with pre-processing the 3D data that are crucial for the training.

2.2.1 Model Formulation

Parametric models of 3D shapes are compact representations that is able to map a single or a set of parameter values to a specific shape. Here we consider shapes that are explicitly represented by 3D coordinates of a collection of points or vertices. In other words, we would like to find a statistical space such that the function

$$F(\mathbf{c}) : \mathbb{R}^d \rightarrow \mathbb{R}^{N\times 3}$$

which takes the vector of d coefficients $\mathbf{c}$, will generate a mesh of the desired shape and pose [21]. Such models can be hand engineered, which is common in graphic applications. However due to the need to represent a wide variety of body shapes and poses, researchers have focused on learning such models from data. In the following, we will describe different categories of parametric models of the human body, classified by the different types of generation functions that are used: triangle based models, basis/manifold based models, blend skinning models, deep neural network based models and models that use implicit functions.

Triangle Based Models

The triangle based models generate model instances by deforming each of the triangle faces in a template mesh. In this case the coefficients $\mathbf{c}$ define the transformation matrices of each face in the template. The SCAPE model [30] represents the shape and pose changes of human body separately. The shape changes are modelled by the per-triangle deformations $\mathbf{C}_t$ where t is the index of the triangle in the mesh. The pose changes are represented with two transformations: $\mathbf{R}_t$ which are computed by the deformation of neighbouring joints of triangle t, and transformations $\mathbf{Q}_t$ which encode the non-rigid pose dependent deformations such as muscle bulging. Neighbouring triangles are forced to have similar $\mathbf{Q}_t$ so that the problem is well-constrained. Given the joint rotations and the shape coefficients, a new mesh can then be computed by first computing $\mathbf{R}_t$, $\mathbf{Q}_t$ as a function of the neighbouring joints, and $\mathbf{C}_t$ as a function of the shape coefficients, then the edges $e_t^k, k = 1, 2, 3$ of triangle t can be deformed as:

$$e_t^k = \mathbf{R}_t \mathbf{C}_t \mathbf{Q}_t \hat{e_t^k} \tag{2.1}$$

then the N vertex locations y_t^k of the resulting mesh, where t and k here denotes the k-th vertex of triangle t, are solved by minimising the overall least squares error:

$$\underset{y_1,\ldots,y_N}{argmin} \sum_t \sum_k \left\| \mathbf{R}_t \mathbf{C}_t \mathbf{Q}_t \hat{e_t^k} - (y_t^k - y_t^1) \right\|^2 \tag{2.2}$$

The SCAPE model is flexible, but generating a new mesh with the model is computationally expensive, since it does not encode vertex positions, and to recover a smooth surface the least square problem need to be solved to "stitch" the triangles (see figure 2.2 for a similar example). Similarly [31] reconstructed the mesh using poisson surface reconstruction algorithm [32] after solving the per-triangle transformations. The computational overhead makes the model unsuitable for applications where speed is important. The problem can be overcame by directly learning a model of vertices



Figure 2.2: An example of deformation transfer using similar approach as the SCAPE model. Deformation of the mesh is modeled as per-triangle transformations. Simply solving the new mesh with neighbouring triangle smoothness constraint gives a discontinued mesh B. Enforcing the shared vertices to be transformed to the same location (or computing the new vertices locations to be as close as possible to B as in the SCAPE model) gives a continued smooth surface C [2].

instead of triangle deformations, as described in the paragraphs below.

Blend Skinning Models

Blend skinning methods (rigging) are widely used in the animation industry. It attaches the mesh surface to an underlying skeleton. Given the transformation of each part of the skeleton, the vertices can then be computed using the weighted influence of its neighbouring bone (figure 2.3). The aforementioned SCAPE model uses a similar approach, except it is based on triangles instead of vertices. A detailed review of blend skinning methods can be found in [3]. We will mainly focus on modelling shapes and poses with blend skinning here, instead of the variations of blend skinning methods.

The blend skinning method transforms the template mesh to give a new mesh of different pose, but do not change the shape of the body. Therefore the shape changes, modelled as vertex displacements from the template mesh, need to be blended into the mesh before skinning. The vertex displacements are computed as the linear combination of the blend shapes, weighted by some coefficients. The S-SCAPE model [33] learns the body shape changes with PCA, then perform rigging on the shaped mesh (figure 2.3 **(b)**). This simplified variation of the SCAPE model does not consider the pose-dependent deformations such as muscle bulging, but is efficient to compute. The SMPL model (Skinned Multi-Person Linear model) [3] learns the pose-dependent deformations as pose blend shapes (figure 2.3 **(c)**). The model is more realistic and efficient to animate compared to other models based on blend skinning method, but it is hard to train. The SMPL pose blend shapes are weighted by the joint rotation matrices. Thus during training $N \times 3 \times 9 \times K$ parameters for the pose blend shapes

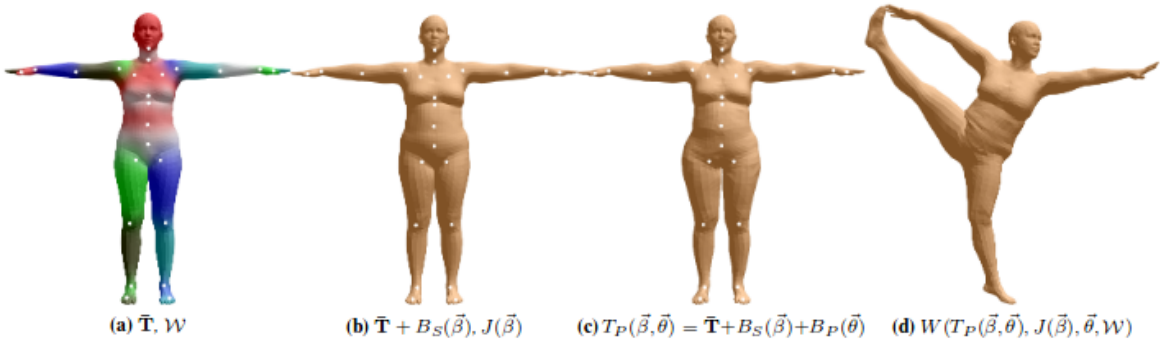


Figure 2.3: A blend skinning human body model proposed by [3]. Given a rigged template mesh (a), the shape and pose-dependent variations are added to template as vertex offsets, and the joint location is updated (b and c), then linear blend skinning is applied to animate the updated template to desired pose.

need to be learned, where N is the number of vertices in the template mesh, and K is the number of joints.

Since the blend skinning methods models the full pose already, the models based on this method need to ensure that the learned shape variations are purely based on individual variations, the pose changes need to be excluded completely. Therefore before learning the shape variations, the data need to be 'pose normalised' so that the meshes are in exactly the same pose. Different pose normalisation methods can be used depending on the model formulation. For example, the S-SCAPE model performs PCA on localised Laplacian coordinates [34] of the registrations. The SMPL model first solves for the pose defined by the joint rotations, then solve for the shape, which when posed by the previously learned pose, will match the registration by minimising the sum of vertex distances between the model instance and the data. [35] normalise the pose by introducing a skeleton model to the scans, and performs Laplacian surface deformation.

Basis/Manifold Based Models

The basis based models learn a set of orthonormal basis vectors, such that the linear combination of the basis vectors gives a new 3D mesh of the desired pose and shape. In this case the coefficients of the generation function F are the weights of the basis vectors.

The most common form of this approach is computing the basis vectors via PCA. Assuming a set of data of rigidly aligned 3D meshes $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_D]$ is given, where $\mathbf{m}_i = [x_1, y_1, z_1, ..., x_N, y_N, z_N]^\mathbf{T}$ is a $3N \times 1$ vector of concatenated Cartesian

coordinates of the N vertices of the i-th mesh. PCA can then be applied to $\mathbf{M}$, resulting in a mean mesh $\bar{\mathbf{m}}$ and a set of orthonormal basis $\mathbf{U}_M \in \mathbb{R}^{3N \times d}$ such that the variance of $\mathbf{M}$ is maximised in $\mathbf{U}_M$. Then the shape generation function becomes:

$$F(\mathbf{c}) = \bar{\mathbf{m}} + \mathbf{U}_M \mathbf{c}$$

The PCA approach has been applied to human face modelling to capture identity and expression variations in both 2D [36] and 3D [37][38][39][40]. The same approach has also been applied to learn model of human body shapes. [41] used a model similar to [37] to learn shapes of human body in a similar pose. Similar models has been used by [42][43] and [44] to reconstruct 3D shapes from single view images. However these models only capture the shape variations of human bodies, not pose.

Models allowing pose variations such as the SCAPE model [30], the S-SCAPE model [33] and the SMPL model [3] use PCA only to learn shape variations, the pose variations are modelled separately with rigging. Models such as [45] use skeletons to model the poses, then apply PCA on the poses represented by the joint rotation matrices to reduce the number of components of the model. Such models are powerful in terms of expressive abilities, however generating a new mesh with these models requires a more complex generation function rather than the simple linear combination of basis.

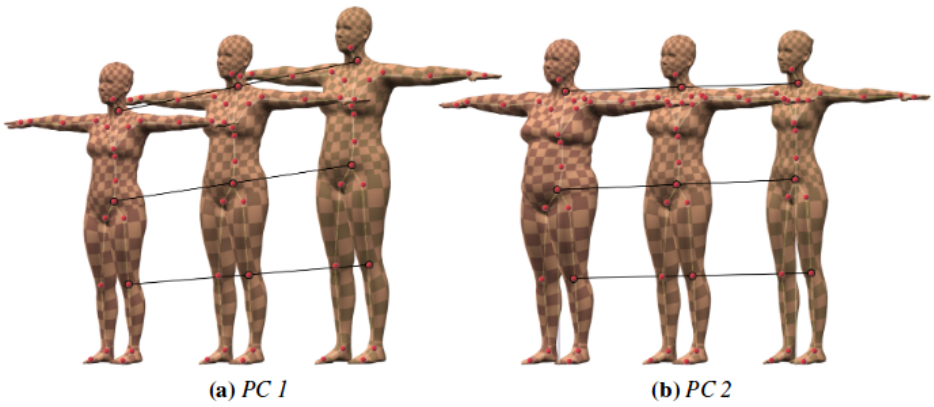
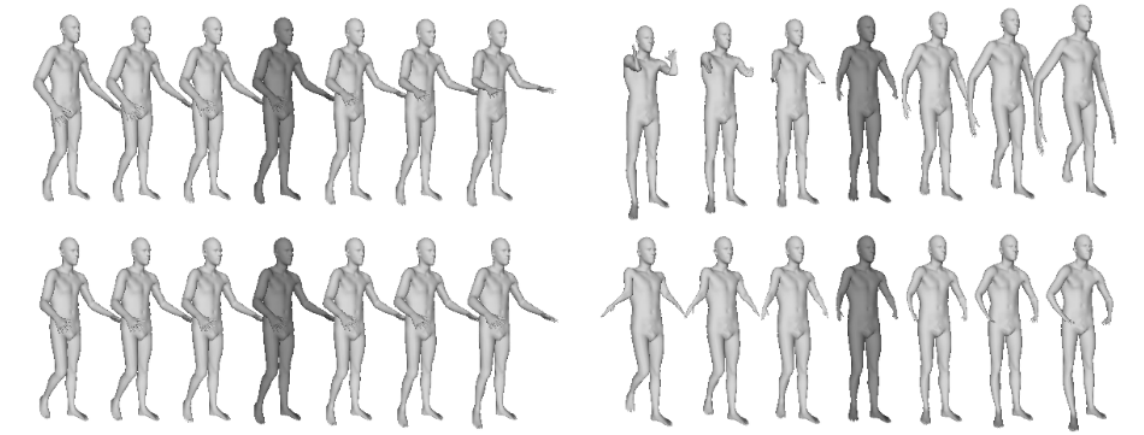


Figure 2.4: A human body shape space learned by PCA [3], the first two principal components are visualised, varying from -2 to +2 standard deviation

PCA assumes the data can be well-approximated by a hyper-planar manifold, which makes it unsuitable for data with high non-linear variations such as articulated pose variation of the human body (see figure 2.5 (b) for an example). Using PCA to learn human body pose variations will result in unrealistic artefacts. Thus non-linear approaches have been proposed [4] to perform PCA in non-linear manifold. In this

case the manifold need to be defined or learnt, and the statistics for computing the mean, covariance and eigenvectors for PCA also need to be defined (see [4] and [46] for details).



(a) Shell PCA modes (elastic average in dark grey)

(b) Euclidean PCA modes (linear average in dark grey)

Figure 2.5: Shell PCA compared to Euclidean PCA on human body pose modelling [4]. PCA in non-linear manifolds better captures the non-linear nature of human body articulations comparing to PCA in Euclidean space.

Apart from learning basis from data, shape modelling can also be done with Gaussian process [47]. While PCA learns a discrete covariance matrix from the data matrix, Gaussian process defines a continuous kernel function. Then the eigenfunctions of the kernel function can be estimated to form a set of basis functions via the Nystrom approximation algorithm given just a single template mesh [5].

The kernel defines the correlation between two points x and y based on their euclidean distances, such that closer points on the mesh are more correlated. Moreover, while most of the PCA approaches model the vertex locations of the mesh, the Gaussian Process Morphable Model (GPMM) directly models the deformations of the vertices, therefore the samples of the model are deformation fields instead of meshes. The deformation of the template mesh in the model space defined by this kernel is smooth, since neighbouring points are more correlated, they are more likely to deform similarly.

Note that the resulting meshes given by the samples of this model are not necessarily valid faces or bodies (see figure 2.6), but they are flexible enough to help the following optimisation steps to converge faster. For example, [48] uses the GPMM to adapt the template mesh to be as close to the target as possible by matching some target key points, then use Coherent Point Drift (CPD) to perform mesh registration. The kernel



Figure 2.6: A template mesh deformed by the samples of the GPMM with a Gaussian kernel [5]. Meshes synthesised from GPMMs are not necessarily valid shapes, but they are flexible enough and could benefit downstream optimisation steps.

function can also be defined as the Gaussian kernel multiplied by the data covariance matrix, which allows the results to be closer to valid faces, but with more flexible variations than the PCA models.

Deep Neural Network Based Models

Another line of work attempts to model 3D human bodies with deep learning approaches. In this case, the generation function F takes the form of neural networks with a decoder structure, and the input coefficients $\mathbf{c}$ is a latent vector. The network can either be fully connected or convolutional. The network is trained either as part of a variational autoencoder (VAE) or a generative adversarial network (GAN). Then the decoder part of the VAE or the generator part of the GAN is retained for 3D mesh synthesis.

[25] proposed a variational autoencoder that operates on the shape descriptor domain. They use the rotation-invariant mesh difference (RIMD) feature, and attempt to reconstruct the feature with a VAE. While synthesising shapes, 3D meshes are reconstructed from the output features by minimising the RIMD energy function over vertex locations. The use of shape descriptor guarantees accurate and detailed modelling compared to the baseline method which outputs directly globally aligned 3D vertex coordinates in the authors' experiments, however size of the model is increased as a feature vector of dimension 60K is needed to represent a mesh of 5K faces. Moreover, an extra step of optimisation is needed in order to reconstruct a 3D shape from the output feature vector. [49] proposed to use VAE to model only the shape variations of human bodies. While the VAE outputs 3D vertex locations, the output is then fed into a blend skinning layer to produce articulated body shapes.

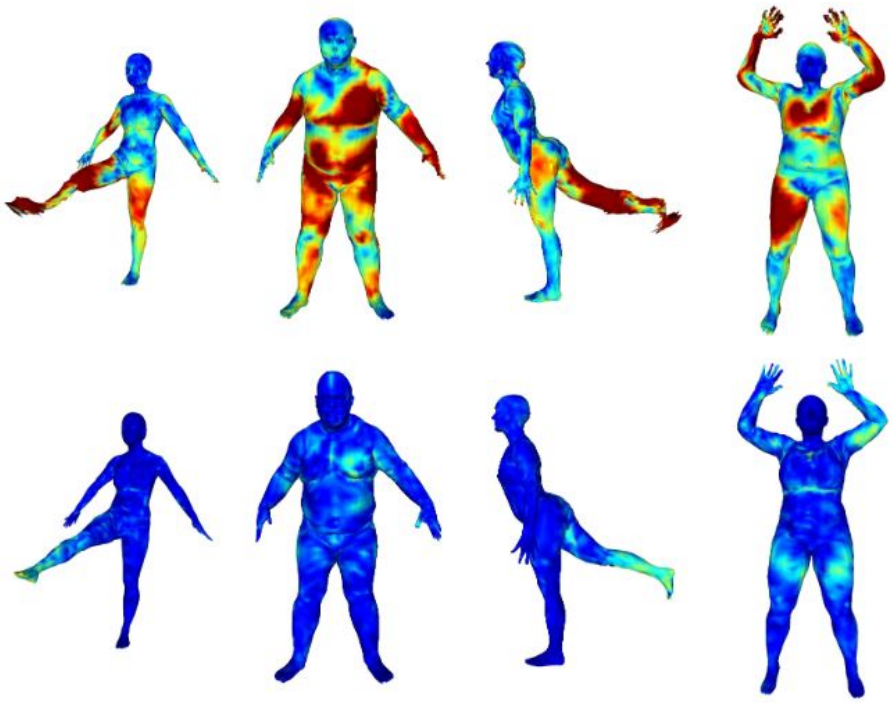


Figure 2.7: Mesh convolutional based models struggle to model large articulated poses (top row: COMA [6]). The result is improved when the number of learnable weights in the network is increased (bottom row: Neural 3DMM [7]), however the reconstruction error is still high for some parts.

To exploit the graph structure of meshes, many works has been proposed to generalise convolutional operator to 3D shape learning and analysis. [50][26][27] has attempted model 3D human bodies with graph convolutional VAEs on the shape descriptors. [6] proposed to build mesh autoencoders for 3D faces by performing convolutions on the spectral domain [51]. [24] used VAE on the spectral domain to model clothes deformations from a T-pose template of the human body. [28] attempted to regress 3D human body meshes from latent codes computed by an image encoder, and combined the result with linear blend skinning (LBS) based models. [7] proposed neural 3D morphable models with spiral convolutional autoencoders, and evaluated the model on the task of 3D reconstruction of human body meshes.

The challenge in building a deep neural network based parametric model is to generate plausible shapes. As shown in [7], graph convolutional based models struggle to produce a smooth surface when large articulated motions occur (Fig. 2.7). [28] used a linear blend skinning model post hoc to regularise the generated shapes. [24] used mesh convolutional networks to model only deformations at T-pose. While adversar-

ial losses have been used in [10] to learn valid joint rotations for linear blend skinning models, [52] proposed the first generative adversarial network to generate directly 3D shapes with mesh convolutions, and demonstrated its superiority over autoencoder based models. However the work has not been extended for body modelling, and could be an interesting direction for future work.

Models Using Implicit Shape Representations

Implicit functions [53] are a type of representation of surface where a function F takes as input a point p, and outputs a value indicating if the point is inside/outside/on the surface of an object. The indicator value varies depending on the implementation, while the function F generally takes the form of a deep neural network in the era of deep learning. [18] used a VAE which takes as input the the point location and a latent vector to generate the probability of occupancy, which is further processed to produce a high resolution mesh using octree. [8] used a fully connected decoder network and employed latent-GANs on the input feature vectors to generate 3D objects. [19] used a auto-decoder structure where latent vectors paired with each shape and the network parameters are optimised jointly during training.

The aforementioned work aims to generate 3D shapes of objects such as chairs and tables. [17] attempted to generate 3D human bodies by combining local analytic implicit functions with deep implicit functions. Each local implicit function is a gaussian that provides a coarse shape and region of influence for each shape element, and the deep implicit function networks are used to refines the shape details. The results of each part are then combined to produce the full shape. The authors have attempted to reconstruct samples from the SMPL model for evaluation, however details of the body are not well preserved. [29] proposed to use an effective spatial sampling strategy that is more suitable for meshes. Points are sampled from the surface and perturbed before combined with points uniformly sampled from the object's bounding box. The resulting shape is then able to preserve details on the surface such as clothes folds.

2.2.2 Model Training and Data Pre-processing

Training data-driven parametric models involves solving for the parameters ϕ of the generation function F such that for instances $\mathbf{S}_n$, $n = 1, ..., N$ in the training set, the following loss is minimised:

$$\sum_{n=1}^{N} d(F_\phi(\mathbf{c}_n), \mathbf{S}_n) \tag{2.3}$$

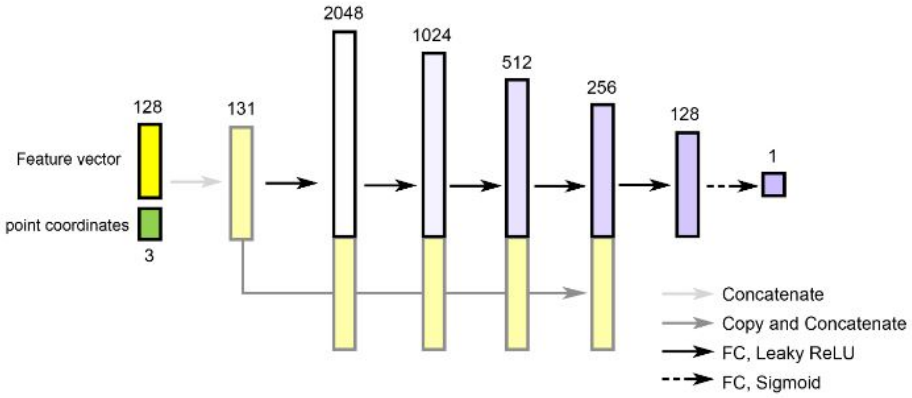


Figure 2.8: Network structure of implicit function based model used in [8]. Given a feature vector, the network outputs a binary value indicating if the queried point is inside or outside the object. Surface is reconstructed by sampling points in the 3D space and query with the network.

where d is the distance function, generally takes the form of L1 or L2 norm. $\mathbf{c}_n$ is the vector of optimal coefficients that gives the best fit for instance n within the model space. ϕ and $\mathbf{c}_n$ can be optimised jointly or in an iterative manner.

3D data used to train the models are often required to share the same topology and semantic ordering of points, with the exception of implicit function based models where the 3D objects are treated as surface instead of a collection of points. Here we consider the case of using meshes as training data. Meshes in different datasets often do not correspond, in the sense that they have different number of points and ordering of points. Also, the desired parametric models are often based on meshes of relatively low resolutions for simplicity and efficiency, while the scans are often of high resolutions. Moreover, there is typically noise in the scans, resulting in holes on the surface which need to be filled or outlier points that need to be removed. Thus pre-processing of the data is a crucial step for training a high fidelity model.

To jointly tackle the aforementioned problems, surface registration techniques can be used, where a low resolution template mesh will be aligned and deformed to match the scans. The problem can be formulated as an optimisation problem. Assuming a template mesh $\mathbf{T}$ and a scan $\mathbf{S}$ are given, the goal of surface registration is to find the deformation D such that:

$$E = \|D(\mathbf{T}) - U(\mathbf{S})\|^2$$

is minimised, where $D(\mathbf{T})$ is the template mesh deformed by D, and $U(\mathbf{S})$ is the set of corresponding points of each point in $\mathbf{T}$ on $\mathbf{S}$ [54].

Searching for the correspondences is the key of surface registration algorithms. Sometimes sparse ground truth correspondences are available from the dataset. For example, markers can be attached to the subjects prior to data acquisition, and they can be corresponded to manually defined landmarks on the template mesh [55]. When ground truth correspondences are not available, correspondence can be established by searching for pairs of the closest points between the target scan and the template mesh. [5] finds the closest points on the scan for each point in the template mesh (template to scan correspondence). [45] and [40] also included the scan to template correspondence in their cost function. [33] also computes the compatibility of the closest points where the compatibility is defined by the angles between the surface normals of a pair of candidate. [48] considers only the "mutual nearest neighbours" between the template and scan. While applicable, texture or colour information can also be used as evidence for establishing correspondences [56].

Correspondences established by searching for nearest neighbours are only reliable when the template and target scan are close to each other. Several approaches can be used to address this issue. The Iterative Closest Points method (ICP) [57] searches for the closest points in an iterative manner. After each iteration of deformation, the correspondences are updated using the latest deformed template. Parametric models with manual initialisation of parameters (e.g. the skeleton structure of linear blend skinning based models) can also be used to estimate an initial fitting by first solving for the optimal coefficients with sparse correspondences. The results can then be further refined with closest point approaches outside the model space [58][59][45]. Without a parametric model initialisation, the template can be deformed using some mesh editing algorithms as priors such as the Laplacian mesh editing algorithm [60], Free-form Deformation [61], or Gaussian Process Regression [5].

To deal with missing data and noises in the scans, several approaches have been proposed. For example the "mutual nearest neighbours" correspondence used in [48] also helps to deal with missing data. [54] and [33] include a smoothness or stiffness term in their cost function so that neighbouring points are deformed similarly to prevent overfitting. Other model based approaches such as [40] and [45] regularise the registrations towards the model instances which matches the scans best, so that the registrations cannot deform arbitrarily away from the model space.

After registration, the registered meshes need to be rigidly aligned so that the models

trained with these meshes learn only the shape and pose variations; rotation, translation, and scaling in the dataset will be excluded. Generalised Procrustes Analysis [62] can be applied in this case. A detailed survey on the rigid registration methods can be found in [63].

2.3 Applications

Parametric models of 3D human body has many applications. In this section, we briefly discuss it's application in the task of 3D pose estimation/shape reconstruction from images, 3D shape completion/synthesis and 2D image synthesis.

Parametric models have been widely used in the task of 3D reconstruction from single view images. The problem itself is challenging due to the ambiguity introduced when projecting a real world 3D scene into the 2D image plane. Many works have focused on recovering the poses from images only, resulting in skeletons instead of 3D shapes. Without using a parametric model, a mapping between appearance and body pose need to be learned, i.e. use the image pixels as the input to some regressors to induce the resulting pose [64]. Some of the recently popular deep learning approaches fall into this category, where the 2D image pixels are used as input of Convolutional Neural Networks (CNNs) to compute the 3D joint locations which define the pose [65]. The limitation of such model free method is the lack of prior knowledge of the object's structure and characteristics. In the case of human body poses, this will lead to implausible 3D poses due to the misprediction of joints, as well as self occlusions of the body in the image. A model free reconstruction of the full 3D surface is even more challenging as the degree of freedom increases, and many works have attempted to incorporate a parametric model post hoc to the model free prediction to recover a plausible 3D prediction [28].

Parametric models incorporate prior knowledge about the body. This could be learned from data, manually defined or in the form of adversarial training. The parametric model itself restricts the generated shapes to lie within the model space, thus regularisation only need to be enforced on the model parameters, instead of the whole shape. The model can be used for 3D shape reconstruction from images in a multi-staged system or one stage system. One example of multi-stage system is [66], which first detects 2D keypoints from the image, then optimise over the model and camera parameters to minimise the keypoint reprojection loss. [10][67][68] use instead a one stage system, which directly regress the model parameters from images using CNNs.

On the other hand, shapes given by parametric 3D reconstructions often shares the same topology, therefore landmark localisation and semantic segmentation problems can be solved automatically from the resulting shape. Also, since the models represent the human pose and shape with only a few parameters, the human motion and shape in 2D images can be parameterised easily with the fitting results, which facilitates markerless motion capture and transfer in the wild.

The unified mesh topology resulted from parametric 3D reconstructions provides a handle to textured 3D shape digitization. [22] used SMPL model plus vertex offset induced from silhouette to reconstruct a person in clothes from a video, the texture is then painted from multiview images to give a textured reconstruction of the person. [69] used a texture inpainting network which allows full body texture generation from reconstruction of a single image. Since the digitization result is automatically rigged as an instance of the parametric model, it can be subsequently controlled and rendered to synthesis personalised animations. Without a parametric model, the problem of motion transfer between images are often tackled with image to image translation GANs [70][71]. With parametric reconstructions of a textured human body, the problem simply becomes rendering and blending the person into a background image [72]. This also allows image synthesis from different view point, which is not straightforward with image to image translation methods. The parameterisation can also be used to reshape the person in the image as demonstrated in [73].

Parametrising human body shapes and poses also facilitates the synthesis of garments associated with the body, as such, the model can be used to power virtual try on applications. In graphics applications, cloth simulation focus on modelling the physical property of the garment. However with parametric models, the deformation and wrinkles of the clothes can be associated with the pose parameters of the body [23][24][74][75].

Model	Template	Input	F	Output	Learned parameters	Dataset	Size	Application
SCAPE	Mesh	Joint angles + Shape coefficient	Linear + Surface reconstruction	Mesh	Triangle deformations	Pose (70) + Identity (37)	$O(N)$	Shape completion
S-SCAPE	Mesh	Joint angles + Shape coefficients	LBS+Linear	Mesh	Vertex based shape basis	Identity (CAESAR)	$O(ND + NJ)$	3D shape reconstruction
SMPL	Mesh	Joint angles + Shape coefficients	LBS+Linear	Mesh	Shape basis, LBS weights, Pose-dependent basis	Pose (FAUST) + Identity (CAESAR)	$O(ND + NJ)$	3D shape reconstruction, pose estimation [10], image digitization [22], virtual try on[23][24]
FCN	Features, Mesh	Latent vector	Fully connected network	Features, Mesh	FCN weights	-	Varies	Shape reconstruction/synthesis [25]
GCN	Features, Mesh	Latent vector	Graph convolutional network	Features, Mesh	Convolutional kernel weights	-	Varies	Correspondence [26][27], pose estimation [28]virtual try on [24]
Implicit	-	Point coordinate	FCN + Surface reconstruction	Inside outside value	FCN weights	-	Varies	Image digitization [29]

Table 2.1: Comparison of different types of parametric models for 3D human body

Chapter 3

Reparameterising 3D Morphable Models

The limitation of statistical shape models is that, once built, the model can only represent 3D shape instances of a fixed mesh topology. While some applications may require a shape model of a different mesh topology, the model building pipeline has to be repeated with the new template, which could be time and computational resource consuming. In other cases only the statistical model is available and access to the original data is not possible. In this chapter, we present a method to reparameterise a given 3D statistical shape model to any topology without using any training data. We also show that the reparameterised model achieves comparable performance as the original model.

3.1 Introduction

3D statistical shape models are widely used for modelling human faces [37][76], bodies [77][78][79] as well as objects such as human bones and organs [80]. The idea of building statistical shape model is to perform PCA on a set of registrations aligned to a pre-defined template, the shape space is then parameterised by the principal components. Once a statistical shape model is trained, the topology of the instances in the model space is then fixed. However, in many applications, a model with a different topology might be useful. For example, in [38], in order to reconstruct the 3D structures of human faces in 2D images, an accurate high resolution per-vertex texture model is needed. In [3], the pose dependent shape variations are modelled with a matrix of dimension $3N \times 9n$, where N is the number of vertices in the registration and n is the number of joints in the skeleton structure of the template. For these applications,

one may prefer to reuse a model instead of repeating the model training pipeline with another template, since it could be quite demanding in terms of computational time. In other cases it would not even be possible because access to the original training data is prohibited.

Motivated by the aforementioned problems, we propose a simple, yet very efficient, method of reparameterising statistical shape models given a new template of a different topology. Our method is based on the probabilistic nature of statistical shape models. Given a model and a new template, we solve for a covariance matrix for the new model directly without using or generating any training data. We present the formulation and solution of the problem in section 3.3. In section 3.4, we present both qualitative and quantitative evaluation of our method. Finally in section 3.5, we summarise our contribution and provide ideas for future work.

3.2 Related Work

The original formulation of 3D Morphable Models (3DMM) was proposed by [37], where they construct 3D face shape models by performing Principal Component Analysis on a set of training face meshes in full correspondence. 3DMMs have since been widely applied to human face modelling. While [76] and [81] showed the linear PCA based 3DMMs can capture the identity dependent variations in human faces, such as gender, ethnicity and age, the work of [38] and [40] also modelled the expression variations. The success of 3DMM is due to the fact that the identity and expression variations in human faces can be well approximated by a hyper-planar manifold in Euclidean space [4], the face shape space can then be parameterised by the axes of the manifold where the captured variance is maximised. The same assumption holds for human body shape variations related to identity [77][78][79][3].

For nonlinear variations such as human body pose changes, the most commonly used approach is blend skinning, where each vertex in the pre-defined template is transformed as a function of the neighbouring bones. In this case the model is parameterised by the parameters of the blend skinning function. We refer the reader to [3] for a detailed review on blend skinning methods. [40] also modelled variations such as jaw movement in faces with blend skinning. [82] used blend skinning method to model the pose changes of hands.

We aim to reparameterise 3D statistical models. For PCA based models, our goal is

to reparameterise the principal components. For blend skinning based models, the goal is to reparameterise the per-vertex parameters of the blend skinning function. To the best of our knowledge, this is the first work on reparameterising statistical shape models.

3.3 Method

3.3.1 Problem Formulation

Assuming a set of D aligned 3D meshes are given, where each mesh $\mathbf{m}_i$ with N vertices is represented by an $3N \times 1$ vector $[x_1, y_1, z_1, ..., x_N, y_N, z_N]^T$. From the probabilistic view, a PCA-based statistical shape model assumes the shape variations can be modelled with a normal distribution [47]:

$$\mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{3.1}$$

where the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ are computed from the data as:

$$\boldsymbol{\mu} = \frac{1}{D}\sum_{i=1}^{D}\mathbf{u}_i \tag{3.2}$$

$$\boldsymbol{\Sigma} = \frac{1}{D-1}\sum_{i=1}^{D}(\mathbf{u}_i - \boldsymbol{\mu})(\mathbf{u}_i - \boldsymbol{\mu})^T \tag{3.3}$$

Given an N_1-vertexed model $\boldsymbol{M}_1$: $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and a new template $\mathbf{m}_2$ represented as $\boldsymbol{\mu}_2$ with N_2 vertices, the problem of reparameterising $\boldsymbol{M}_1$ to be used with $\mathbf{m}_2$ can be considered as an optimisation problem, where the covariance matrix $\boldsymbol{\Sigma}_2$ which minimises the difference between the old model $\boldsymbol{M}_1$ and the new model $\boldsymbol{M}_2$ need to be solved. With the probabilistic formulation of PCA-based shape models, we solve for $\boldsymbol{\Sigma}_2$ by minimising the Kullback-Leibler (KL) divergence between $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$:

$$\boldsymbol{\Sigma}_2 = \underset{\boldsymbol{\Sigma}}{argmin}\ D_{KL}(\mathbf{W}\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)||\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})) \tag{3.4}$$

where $\mathbf{W}$ is a $3N_2 \times 3N_1$ matrix which maps $\boldsymbol{M}_1$ to the same $3N_2$-dimension as $\boldsymbol{\mu}_2$.

By setting the derivative of Eq 3.4 w.r.t. $\boldsymbol{\Sigma}$ to 0, the new covariance matrix can then be obtained:

$$\boldsymbol{\Sigma}_2 = \mathbf{W}\boldsymbol{\Sigma}_1\mathbf{W}^T + (\boldsymbol{\mu}_2 - \mathbf{W}\boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \mathbf{W}\boldsymbol{\mu}_1)^T \tag{3.5}$$

3.3.2 Computing the W matrix

The performance of the new model depends on the choice of the matrix $\mathbf{W}$. Ideally, the new model $\boldsymbol{M}_2$ can model the exact distribution of $\mathbf{W}\boldsymbol{M}_1$ if $\mathbf{W}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 = \mathbf{0}$, therefore we want $\mathbf{W}\boldsymbol{\mu}_1$ to be as close to $\boldsymbol{\mu}_2$ as possible.

Assuming the new template $\mathbf{m}_2$ has been non-rigidly aligned to the template $\mathbf{m}_1$ of $\boldsymbol{M}_1$. The vertex to surface correspondences from $\mathbf{m}_2$ to $\mathbf{m}_1$ can be computed. We can then define $\mathbf{W}$ as the matrix which maps the N_1 vertices of $\mathbf{m}_1$ to the N_2 points on its surface corresponding to the vertices of $\mathbf{m}_2$, in this way the distance between $\mathbf{W}\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ is minimised while preserving the surface property of $\mathbf{m}_1$.

Such matrix $\mathbf{W}$ can be arranged as a sparse matrix of $N_2 \times N_1$ blocks, where each block is a 3×3 submatrix. Suppose the closest point of vertex $\mathbf{v}_i \in \mathbf{m}_2$ on the surface of $\mathbf{m}_1$ is $(\mathbf{t}_i, \mathbf{x}_i)$, where $\mathbf{t}_i = (a, b, c)$ is the index of the corresponding triangle face defined by three point with index a, b and c, and $\mathbf{x}_i = (u_i, v_i, w_i)$ is the barycentric coordinate of this point within triangle $\mathbf{t}_i$. We then set block (i, a), (i, b) and (i, c) of matrix $\mathbf{W}$ as follows:

$$
\begin{array}{c}
 \\
i
\end{array}
\begin{array}{c}
\begin{array}{cccccccccccc}
 & & a & & & & b & & & & c & \\
\end{array} \\
\begin{bmatrix}
\cdots & & \cdots & & \cdots & & \cdots & & \cdots & & \cdots & & \cdots \\
 & u_i & 0 & 0 & & v_i & 0 & 0 & & w_i & 0 & 0 & \\
\cdots & 0 & u_i & 0 & \cdots & 0 & v_i & 0 & \cdots & 0 & w_i & 0 & \cdots \\
 & 0 & 0 & u_i & & 0 & 0 & v_i & & 0 & 0 & w_i & \\
\cdots & & \cdots & & \cdots & & \cdots & & \cdots & & \cdots & & \cdots
\end{bmatrix}
\end{array}
$$

Setting the blocks in such way for all N_2 vertices gives us the final $\mathbf{W}$ matrix, where the rest of the entries remain zero.

3.3.3 Blend Skinning Models

For human bodies, PCA based shape models are commonly used to model only the identity dependent shape changes, while the pose changes are modelled with blend skinning methods. Here we consider the formulation proposed by [3], in which they used linear blend skinning method to model the pose changes of human bodies, as well as pose blend shape to model the pose-dependent shape changes. We use the matrix $\mathbf{W}$

to transform the vertex based parameters in their model such as the blend skinning weights and pose blend shapes. The pose blend shape models the pose dependent deformations of the N vertices in the template while the n joints rotate. The pose blend shape is linear with respect to the joint rotation matrices, therefore it is represented by a $3N \times 9n$ matrix. We compute the pose blend shape of our new model as:

$$\mathbf{P}_2 = \mathbf{W}\mathbf{P}_1 \tag{3.6}$$

The blend skinning weights define how much the transformation of the n joints affect the transformation of the N vertices. It is represented by an $N \times n$ matrix. We rearrange our $\mathbf{W}$ matrix as a $N_2 \times N_1$ matrix $\mathbf{W}'$. For each pair of vertex to surface correspondence $\mathbf{v}_i$ and $((a, b, c), (u_i, v_i, w_i))$, we set the entries (i, a), (i, b), and (i, c) of $\mathbf{W}'$ as:

$$\begin{array}{c} \\ i \end{array} \begin{array}{c} \begin{array}{ccccccc} \quad & a & \quad & b & \quad & c & \quad \end{array} \\ \begin{bmatrix} \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & u_i & \dots & v_i & \dots & w_i & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \end{array}$$

We then compute the new blend skinning weights as:

$$\mathbf{B}_2 = \mathbf{W}'\mathbf{B}_1 \tag{3.7}$$

3.4 Evaluation

We evaluate the proposed method with statistical models of faces and bodies. For faces, we build a model of 8082 vertices with 2500 randomly selected scans from the MeIn3D [76] dataset following the pipeline proposed in the original paper. We aim to reparameterise the model to have 15176 vertices. 2000 registrations were used for training, and 500 registrations were also aligned to the new template for testing. For bodies, we train a shape model of 6890 vertices with 1657 randomly selected meshes from the MPII Human Shape dataset [83] following the pipeline proposed in [3]. More specifically, we align the meshes to the template used by [3] with non-rigid icp [84], then normalise the registrations to T-pose before performing PCA to build the body model. We then reparameterise the model to have 10412 vertices, and test the models with 500 registrations.

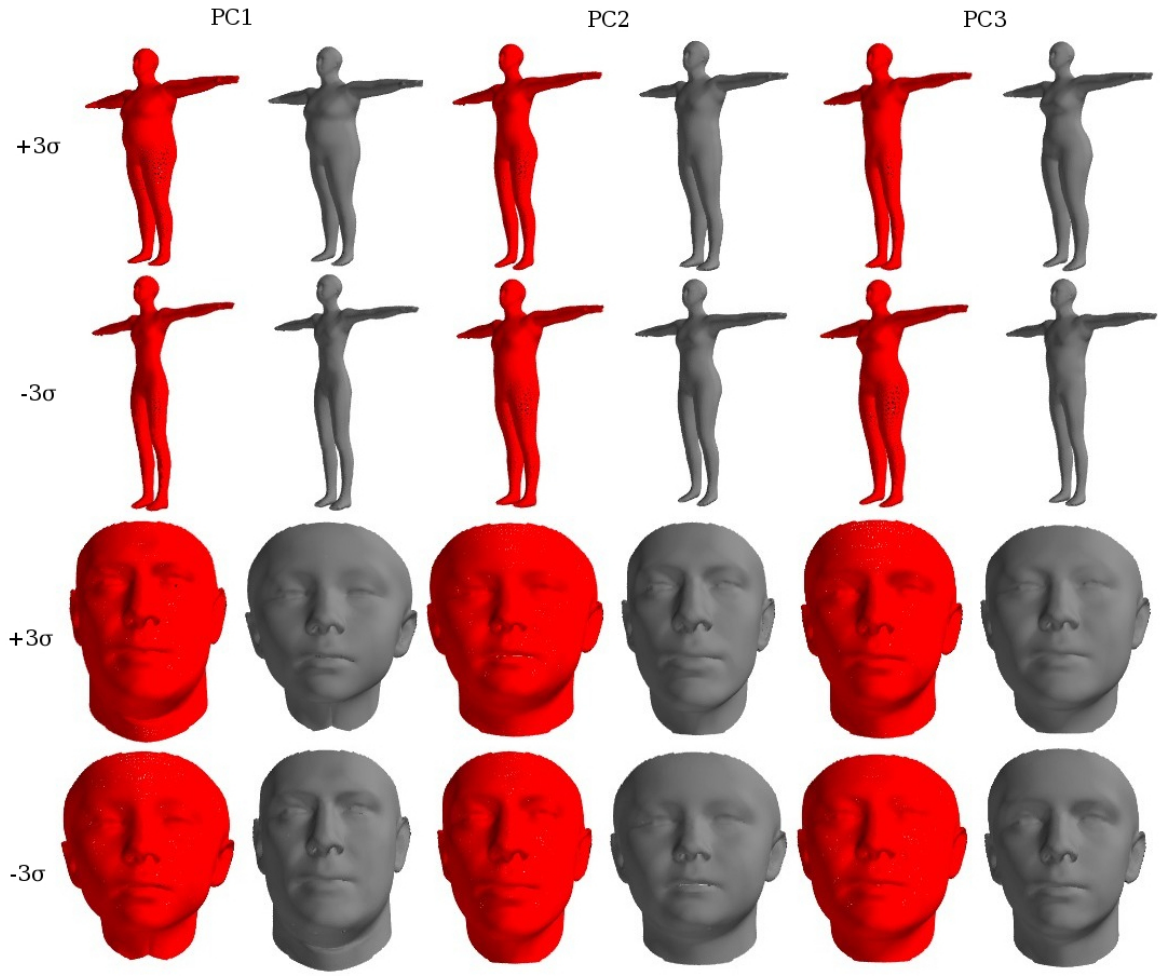


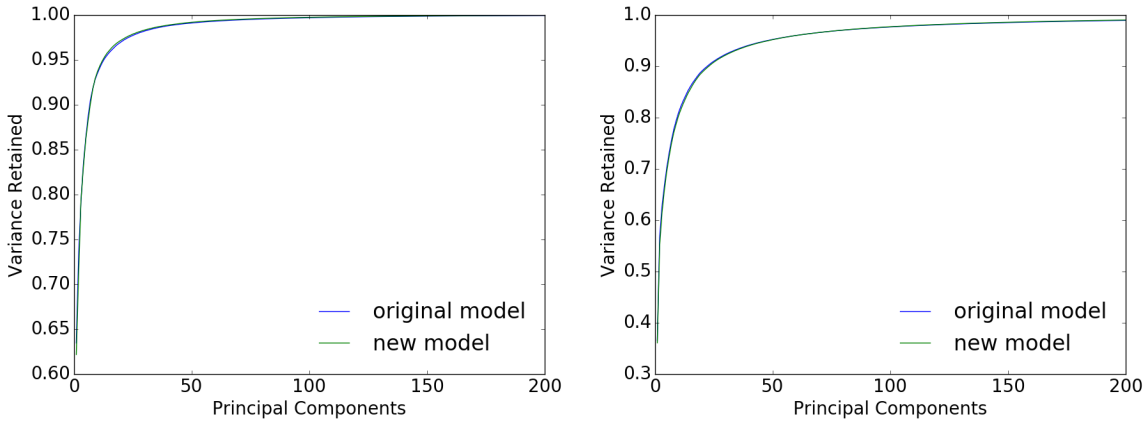
Figure 3.1: Visualisation of the first three principal components at -3 and +3 standard deviations. The red meshes indicate the original model, while the gray meshes indicate the new model.

3.4.1 PCA-based Shape Models

We first visually inspect the quality of the new model by comparing its first few principal components with the components of the original model. In figure 3.1 we visualise the first three principal components of the original and new models from -3 to +3 standard deviation. We observe that each principal component of the new models retain similar variations as in the original models.

We then compare the original model and the new model in terms of their compactness, generalisation, and specificity as proposed by [85]. Compactness is the percentage of variance in the training data that is explained by the model. Figure 3.2 shows the variance retained by the original and new model while a certain number of principal components are kept. For both the body and face models, the principal components of the new model are able to explain nearly the same percentage of variance as the

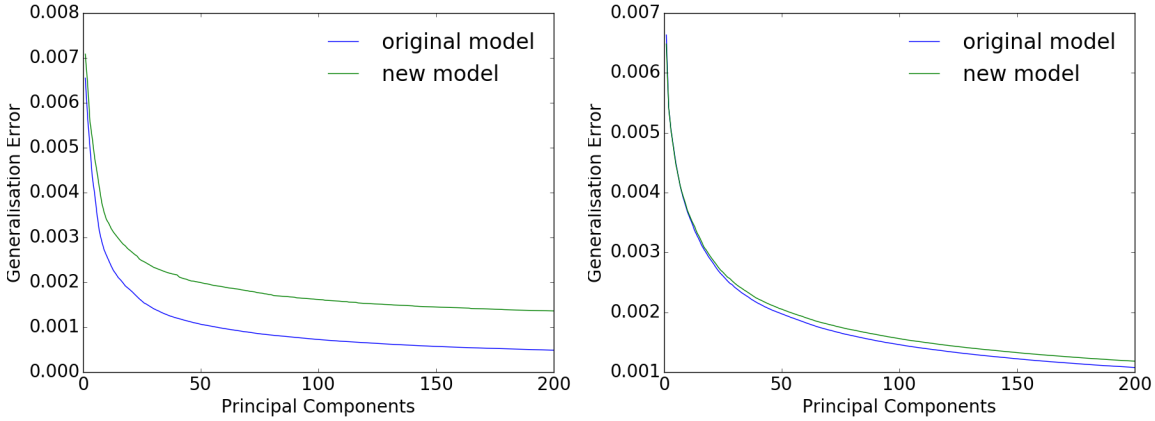
original model. Therefore we consider the new model to be as compact as the original model.



(a) Body Model (b) Face Model

Figure 3.2: Compactness of the Original and Reparameterised Models.

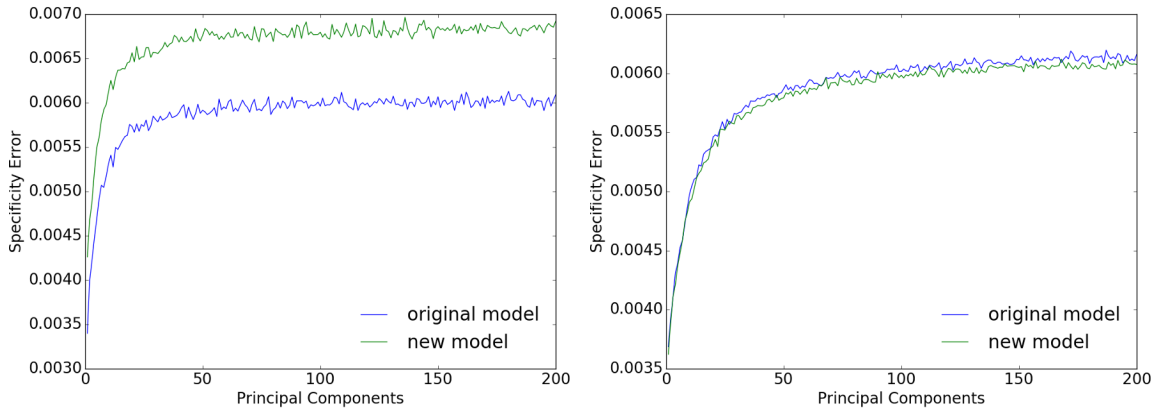
Generalisation measures the model's ability of generalising to new instances that are unseen during the training. We project each of the models to the test set, and compute the mean of the average per-vertex euclidean distance between the model reconstruction and the testing instances to obtain the generalisation error for each model. Note that during training, we scale our templates to fit inside box of diagonal 1. Thus the generalisation and specificity error are measured at this scale. For both the body and face models, the difference between the generalisation error of the new and original model is smaller than 0.001. Therefore we believe that the new models computed with our method can achieve comparable generalisation ability as the original models.



(a) Body Model (b) Face Model

Figure 3.3: Generalisation Error of the Original and Reparameterised Models.

Specificity evaluates the validity of instances generated by the model. For each model, we randomly generate 1000 samples from the model, and compute the averaged per-vertex euclidean distance between each instance and its nearest neighbour in the test set. We then average this error over all 1000 samples as the specificity error of this model. The specificity errors for the face and body models are plotted in figure 3.4. We observe that for the face model, the new model computed with our method have smaller specificity errors. And for the body model, the new model have specificity error larger than the original model, but the difference between the specificity error is smaller than 0.001. Therefore we believe that the new models can generate similar random instances as the original models.



(a) Body Model (b) Face Model

Figure 3.4: Specificity Error of the Original and Reparameterised Models.

3.4.2 Blend Skinning Models

We animate the body models with poses from Unite The People (UP-3D) dataset [86] to inspect the quality of the new blend skinning parameters. Some examples of the animated meshes are presented in figure 3.5. We observe that the animated meshes have smooth surface, which suggests the reparameterised blend skinning weights are able to represent the relations between each vertex in the new template and the joints correctly. Also the pose blend shapes are able to preserve the muscle bulging details given by the original model and the bending artefact around the joints resulted from the linear blend skinning method is corrected accordingly.

3.5 Conclusion

In this chapter we have presented a method to reparameterise 3D statistical shape models given a new template. We have computed a transformation matrix to transform the model template that we wish to reparameterise to be at the same dimension as the new template. For PCA based models, we then computed a new covariance matrix by minimising the KL-divergence between the original model and the reparameterised model. For blend skinning based models, we have transformed the per-vertex parameters with our transformation matrix. We have showed that the reparameterised models are as compact as the original model, and generalise equally well while being more specific than the original model. We have also demonstrated that the reparameterised blend skinning models are able to preserve the surface details of the original model.

In future work, we will explore further the performance of our method by applying the reparameterised models to mesh registration, 3D reconstruction of 2D images, and dense shape regression problems.

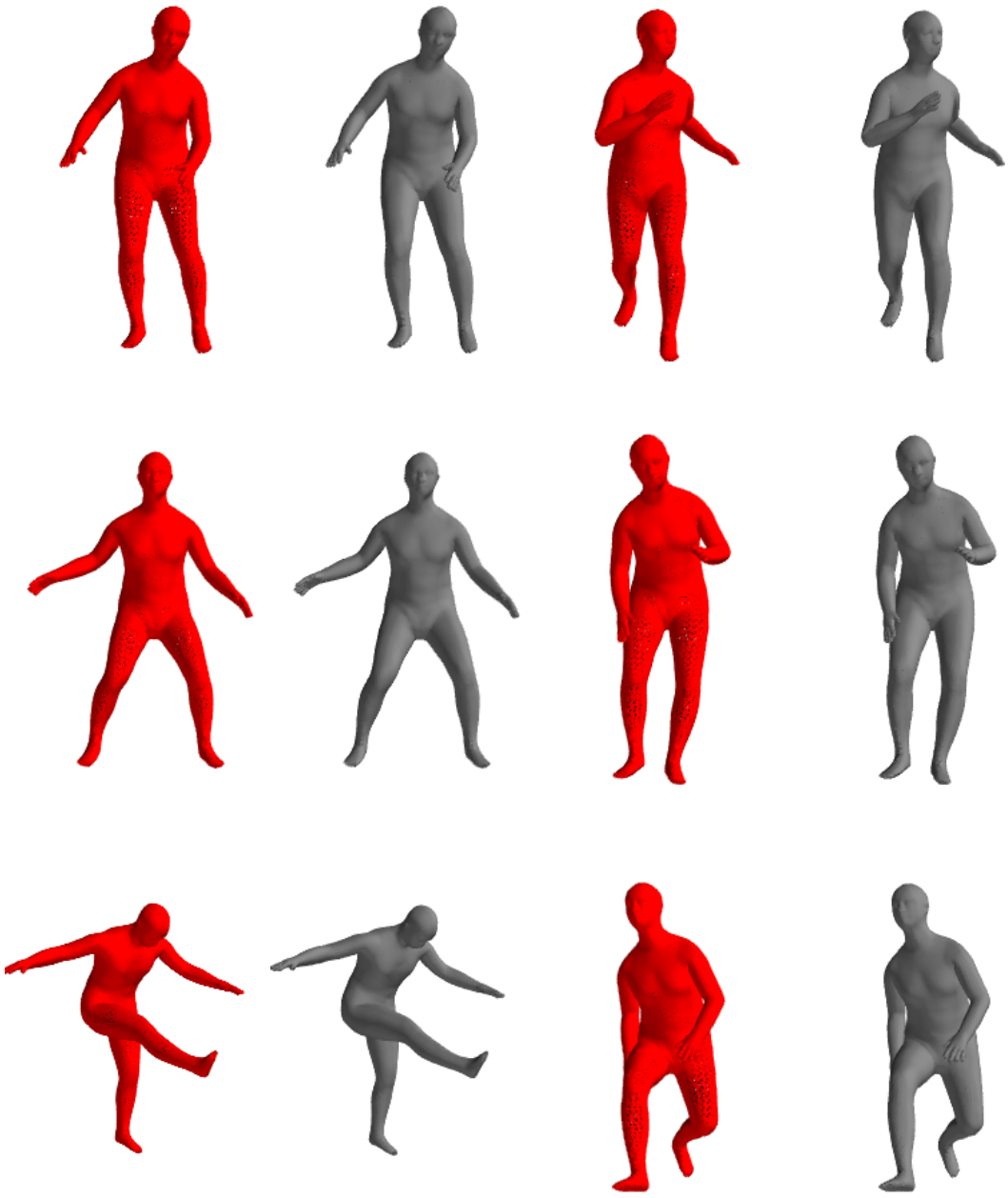


Figure 3.5: Visualisation of the animated blend skinning models. Red meshes: animation result of the original model with 6890 vertices. Gray meshes: animation result of the reparameterised model with 10412 vertices.

Chapter 4

Single Image 3D Human Body Reconstruction with Mesh Convolutions

In this chapter we propose a method for recovering 3D representation of human body from single RGB image. Previous works rely on the use of a parametric model of human body, either in the form of optimisation based fitting method, or deep learning based method that regress the model parameters directly from the input image. In both cases, supervision with model parameters are required for plausible and robust reconstruction results, requiring extra learning or fitting steps for preparing the pose priors.

We propose a network structure that operates directly on 3D mesh vertices instead. Given an input RGB image, our network compute a latent code through an image encoder, which is then sent to a light weighted mesh convolutional decoder that outputs the 3D vertex locations. During training, we use a mesh convolutional autoencoder based discriminator network to enforce plausible reconstructions, which lead to significantly better results compared to the direct vertex regression method through mesh convolutions. We perform quantitative evaluations on the task of 3D pose estimation, and show that our proposed method outperforms comparable linear blend skinning model based methods. We also show our results on in-the-wild images, demonstrating the effectiveness and potential of our proposed approach.

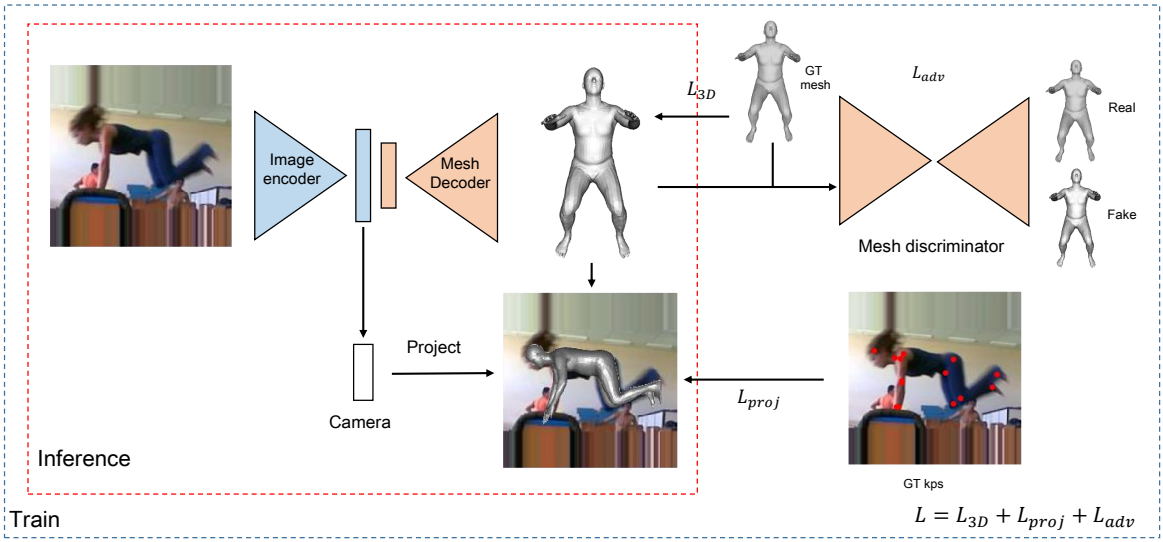


Figure 4.1: Overview of our proposed approach. Given an input RGB image, our network outputs a 3D mesh reconstruction of the person in the image through an encoder-decoder structure. During training, the 3D mesh reconstruction loss and 2D keypoint reprojection loss is minimised. We also use a mesh autoencoder based discriminator network during training to ensure that the image to mesh network outputs plausible human bodies.

4.1 Introduction

Leveraging 3D representations of the human body from a single RGB image has attracted many researcher's attention recently due to its wide range of applications. While many approaches successfully recover 3D joint locations, recovering a full 3D mesh remains a challenge. The problem itself is ill-posed due to the ambiguities between 3D and 2D mapping, self occlusion and the articulated nature of the human body.

Parametric models of human bodies incorporate prior knowledge of the shape and pose variabilities, limiting the predicted 3D mesh to lie within the model space which guarantees plausible outputs. While used for the task of 3D reconstructions, the problem is reduced to predicting the low dimensional model parameters from the input image, instead of the full set of 3D vertex locations. The most widely used parametric human body model is SMPL [3] where the human mesh is parameterised by a set of shape coefficients and joint rotation matrices. Such models are heavy weighted due to the need of storing the blend skinning weight matrix and the linear basis, consequently the size grows linearly while the resolution of the template increases.

Early works that utilise the SMPL model typically involves an optimisation based pipeline [66], where the model parameters are solved in an iterative manner to optimise the model's reprojection loss based on some image observations (e.g. 2D keypoints or silhouettes). The major drawbacks of such approach are the slow running time during inference and the need to solve a non-convex optimisation problem which could easily lead to local minima while dealing with challenging poses. Therefore, focus has been shifted to learning based method, which uses convolutional neural networks to directly regress the SMPL model parameters. However the axis-angle joint rotations used by the SMPL model is difficult to regress, as such, many methods have been proposed to improve the results either by using an iterative error feedback regressor [10] or intermediate representation of the image such as joint heatmaps and part segmentation maps [87][88].

Another issue with the SMPL model is that it does not restrict the human body pose space, which could potentially lead to kinematically implausible results. This issue is addressed by learning a separate pose prior either in the form of mixture of gaussian [66] or with adversarial training [10]. Learning such pose priors requires additional MoCap data that are represented as the SMPL pose parameters, requiring extra step of model fitting.

Motivated by the aforementioned problems, we propose a simple yet effective network structure that operates directly on 3D mesh vertices. The overview of our approach is illustrated in Fig. 4.1. Instead of using linear blend skinning based models, we use a powerful deep mesh convolutional decoder for generating 3D meshes. The shape and pose variabilities of the 3D human body mesh is fully parameterised by a low dimentional latent vector space, which is obtained by pretraining a mesh convolutional autoencoder.

We then combine the mesh decoder with an image encoder and train the network end-to-end to perform the 3D reconstruction task from images. The network is trained to minimise the vertex reconstruction loss, avoiding the need of additional supervision on model parameters. While reliable 3D ground truth is available, our network can be fine-tuned to learn to reconstruct the detailed shapes, overcoming the limited representational power of LBS based parametric models.

We further improve the reconstruction quality of our network by utilising the pre-

trained mesh autoencoder as a mesh prior to enforce plausible reconstructions. While training the image to mesh network, the mesh autoencoder is fine-tuned to learn a vertex reconstruction error based adversarial loss on the meshes, demonstrating for the first time the merit of convolutional mesh autoencoders in adversarial training.

We perform detailed evaluation on the task of 3D pose estimation and compare our proposed approach to several baseline methods. We show that our mesh convolutional decoder based network outperforms the baseline method that regresses SMPL model parameters, while our mesh autoencoder based discriminator further improves the reconstruction results. We also evaluate and compare to several state of the art method on in the wild images, demonstrating the effectiveness and potential of our proposed approach.

4.2 Related Work

Human Body Mesh Recovery

The approaches for recovering a 3D human body mesh from a single input image can be categorised into optimisation based method and learning based method. Optimisation based methods fit a parametric model to the image by finding the optimal set of model parameters that minimises some objective function based on features such as 2D keypoints and silhouette. Whereas learning based methods typically use convolutional neural networks to regress the model parameters or the 3D location of the mesh vertices.

Early works of optimisation based approaches mainly rely on manual annotations [89][90][91]. These methods have restricted usage in the case of in the wild images where manual annotations of silhouette or dense correspondences cannot be obtained. Beyond the limited use case, since optimisation based approaches typically require solving a challenging non-convex optimisation problem, these works have also focused on developing approaches to solve the optimisation problem itself. SMPLify [66] attempts to automate this pipeline where 2D keypoints detected using CNNs are used as the main objective when fitting the SMPL model to the image. They also use an automatic differentiation tool [92] to solve the optimisation problem. [86] improved the shape estimation of SMPLify by also including landmarks on the body surface and a silhouette matching term in the objective function.

Using 2D keypoints and segmentations as objective could lead to implausible reconstructions due to ambiguities. This issue has been addressed in the form of pose priors [66] in the model parameter space. Another major drawback of such optimisation based methods is the inference speed. Since for each input image, an iterative optimisation process is required to solve the parameters, the methods cannot be used for real time applications. The results also rely on reasonable initialisation and keypoints detection or segmentations.

On the contrary, learning based method seeks to train a network with the 2D or 3D supervisions that minimises the reconstruction objective function. During inference, the results can be obtained by simply forward-passing the image through the network, significantly reducing the inference time. Moreover, while additional manual annotation or component for automatic inference of the 2D keypoints/silhouettes is needed during inference for the optimisation based method, learning based methods are typically independent from these as all the inference is done by one network.

Some learning based reconstruction approach are independent from parametric models, where the network directly outputs 3D vertex locations [28], voxelised representation of the mesh [93][94] or dense image to vertex correspondences [95]. These non-parametric approaches do not typically regularise the output, which could lead to implausible or noisy reconstructions for challenging poses. Another line of work utilise parametric models for mesh generation, while the network attempts to regress pose and shape parameters from the input image. To deal with pose ambiguities, either adversarial training on the parameters [10] or pose priors [96] are required. [88] extended the network to perform part segmentation which is then used for model parameter regression.

While optimisation based method often produce reliable results and learning based method can perform fast real time inference, some works have attempted to combine these two. [67] use a model parameter regression network [10] to produce initial parameter estimations, which is then improved by optimisation based method [66], this enables the possibility to use 3D mesh supervision when only 2D keypoints annotations are available. [96] use a multitask network to regress model parameters as well as dense correspondences, 2D and 3D keypoints, which are then used as complementary cues to refine the model based reconstruction.

In this work, instead of focusing on improving the parametric model predictions by using different source of supervision or network architecture, we introduce the use of

mesh convolutional networks that directly outputs the vertex coordinates given an input image encoding. Our approach do not rely on supervision of parametric model parameters, and the network is trained end to end given 3D ground truth of the training images, hence the performance is not limited by the representational power of the parametric model itself.

Geometric Deep Learning

We have presented a detailed survey on deep neural network based 3D human body modelling in Chapter 2, here we focus instead the development on geometric deep learning for 3D shape modelling with convolutional operations on meshes.

One direction of work uses the spectrum of the mesh given by the eigenvectors of the mesh laplacian [97]. The disadvantage of such method is the high computational complexity due to the expensive matrix multiplications required. [51] reduced the complexity by representing the filters as polynomial expansion and using Chebyshev expansion to compute the polynomials recursively. [98] utilised the fast spectral convolution filters to build convolutional mesh autoencoders for generating and reconstructing 3D human faces. [52] further extended the use of mesh autoencoder to build a 3D mesh GAN that generates more detailed high resolution faces.

Another category of approaches such as [99][100] and [101] focus on generalising convolutional operations on meshes in the spatial domain, where a set of weighting functions are defined for the each position in a patch localised to a point. [27] proposed a spiral operator that constructs the local patch in the spiral ordering of the neighbouring vertices for the task of learning correspondences between 3D meshes. [7] extended this work to build a neural 3D morphable model of human faces and bodies. In this work, we employ the setting of [7], but for the task of 3D reconstruction from RGB images instead of directly from 3D shapes.

4.3 Mesh Convolutional Networks

4.3.1 Spiral Convolution

Suppose a 3D mesh is represented by a graph $\mathcal{G} = (\mathbf{V}, \mathcal{E})$, where $\mathbf{V} \in \mathbb{R}^{N \times 3}$ represents the vertices coordinates in 3D Euclidean space, and $\mathcal{E}$ is the set of edges that defines the connectivity of the vertices. Mesh convolutional filters can either be defined from the spatial approach or the spectral approach. Here we consider mesh convolutional

filters defined on the spatial domain, where a set of local weighting functions $w_1, ..., w_L$ is defined on the local system of coordinates $\mathbf{u}(x, y)$ for each vertex y in the neighbourhood $N(x)$ of a vertex x. The convolutional filter is then defined as a patch operator where the features $f(y)$ are aggregated as:

$$(f * g)_x = \sum_{l=1}^{L} g_l \sum_{y \in N(x)} w_l(\mathbf{u}(x, y)) f(y) \tag{4.1}$$

where g_l is the filter weights. Here $w_l(\mathbf{u}(x, y))$ are a set of learnable soft-attention weights that performs a all-to-all mapping since a one-to-one mapping cannot be obtained due to the absence of a global coordinate system.

However, when considering a mesh of fixed topology, the neighbouring vertices of a point can be ordered in a way such that the local order is fixed. In this case, the convolutional operation can be defined as follows:

$$(f * g)_x = \sum_{l=1}^{L} g_l f(x_l) \tag{4.2}$$

where $x_1, ..., x_L$ are the neighbours of x ordered in a fixed way, which is equivalent of having a patch operator on a single neighbouring vertex. Following [7], we use the setting where the vertex ordering is obtained using spiral trajectories. Suppose $R^d(x)$ is the d-ring of vertex x, and $R^d_j(x)$ is the jth element in the ring. The spiral patch operator is defined as the ordered sequence:

$$S(x) = x, R^1_1(x), R^1_2(x), ..., R^h_{\|R^h\|(x)} \tag{4.3}$$

where h is the patch radius or kernal size. The spiral convolution is then computed as:

$$(f * g)_x = \sum_{l=1}^{L} g_l f(S_l(x)) \tag{4.4}$$

The ordering is defined by the direction of ring and the starting vertex $R^1_1(x)$. Here the ordering is pre-computed on a template mesh, where $R^1_1(x)$ is chosen to be the closest vertex of each x, and the rest of the sequence is deduced by going through the d-ring counterclockwise.

4.3.2 Mesh Pooling

The pooling operation on mesh is equivalent to mesh down-sampling. We adopt the implementation of [6]. At each pooling layer, the feature map $\mathbf{X} \in \mathbb{R}^{n \times F}$ is multiplied by selection matrix $\mathbf{Q}^d \in \mathbb{R}^{n \times m}$, where $\mathbf{Q}^d_{ij} = 1$ indicates that the i-th vertex is preserved during the down-sampling operation, and correspond to the j-th vertex in the next layer. $\sum_j \mathbf{Q}^d_{ij} = 0$ means that the vertex is discarded.

The unpooling operation can be seen as subdividing the surface to increase the resolution of the mesh. However in order to preserve the template topology and enforce symmetry when used in a mesh autoencoder, an up-sampling matrix $\mathbf{Q}^u$ can be built simultaneously as the down-sampling matrix.

For each vertex that is preserved in the down-sampling operation, i.e. $\mathbf{Q}^d_{ij} = 1$, $\mathbf{Q}^u_{ji}$ is set to be 1. For each discarded vertex, the vertex is projected to the nearest triangle in the down-sampled mesh, and retrieved with the corresponding barycentric coordinate during up-sampling. Assume a vertex i is discarded and its closest triangle on the down-sampled mesh is (p, q, r), and its corresponding barycentric coordinate is $[u, v, w]$, then the up-sampling matrix is constructed such that $\mathbf{Q}^u_{pi} = u$, $\mathbf{Q}^u_{qi} = v$, and $\mathbf{Q}^u_{ri} = w$.

Since all the input meshes share the same topology and semantic ordering of vertices, the down-sampling and up sampling matrix can be computed only once on the template mesh. [6] computed the down-sampled vertices using a quadric error based edge collapse method on the reference face. We observe that when applied to mesh of human bodies, the choice of reference mesh is crucial and has considerable impact on the reconstruction quality of the result. Fig. 4.2 shows our choice of reference mesh and the down-sampled meshes with factor of 4, 2, 2, 2 respectively. The reference mesh is posed such that after down-sampling, vertices around the joints are preserved.

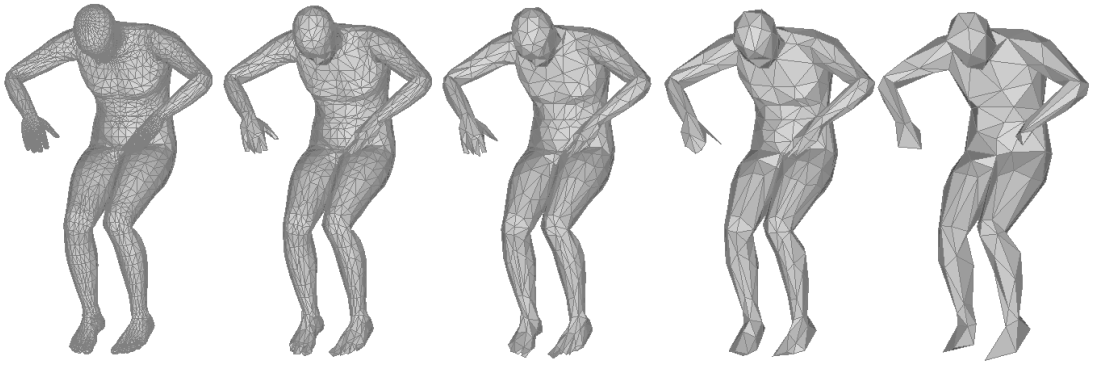


Figure 4.2: Our choice of reference mesh and the downsampled meshes with factor of 4, 2, 2, 2. By posing the template mesh as illustrated, vertices around the joints are preserved after down-sampling, subsequently helping the network to learn human body articulations in a coarse level.

4.3.3 Mesh Convolutional Autoencoder

Having specified all the components of the network, we now describe the structure we used to build the mesh convolutional autoencoder.

Each building block of the network consists of one spiral convolutional layer and one pooling/unpooling layer. The input layer is a convolutional layer with kernal size 2 and 16 channels, followed by 4 blocks of layers with channel size (16, 32, 64, 128), kernel size (2, 1, 1, 1) and down-sampling factor of (4, 2, 2, 2) respectively. Then we use a fully connected layer to obtain the latent code of size 128. The decoder is symmetric to the encoder. We use biased Elu activation for the convolutional layers, except for the output layer. Figure Fig. 4.3 illustrate the network structure, and the hyperparameters of the network is summarised in table Table 4.1.

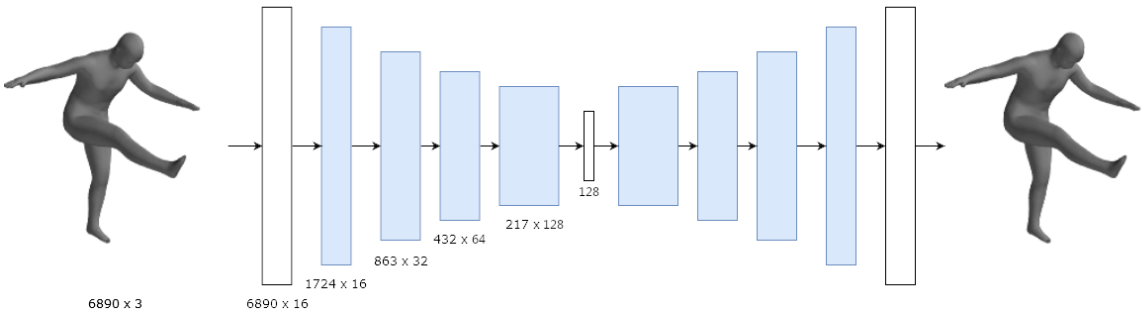


Figure 4.3: Illustration of our mesh convolutional autoencoder architecture.

Layer	Input Size	Output Size	Kernal size	Layer	Input Size	Output Size	Kernal size
Conv-elu	6890x3	6890x16	2	FC	128	217x128	-
Conv-elu	6890x16	6890x16	2	Unpool	217x128	432x128	-
Pool	6890x16	1724x16	-	Conv-elu	432x128	432x64	1
Conv-elu	1724x16	1724x32	1	Unpool	432x64	863x64	-
Pool	1724x32	863x32	-	Conv-elu	863x64	863x32	1
Conv-elu	863x32	863x64	1	Unpool	863x32	1724x32	-
Pool	863x64	432x64	-	Conv-elu	1724x32	1724x16	1
Conv-elu	432x64	432x128	1	Unpool	1724x16	6890x16	-
Pool	432x128	217x128	-	Conv-elu	6890x16	6890x16	2
FC	217x128	128	-	Conv-elu	6890x16	6890x3	2

Table 4.1: Architecture of our mesh convolutional autoencoder.

Given a set of N meshes $\{\mathbf{V}_i\}$, we first normalise the vertices to obtain the normalised deformations $\hat{\mathbf{V}}_i$ by substracting the mean shape and dividing by the per-vertex standard deviation computed over all the training samples. The autoencoder is then trained to minimise the L1 reconstruction error:

$$\sum_{i=1}^{N} \|\hat{\mathbf{V}}_i - AE(\hat{\mathbf{V}}_i)\| \tag{4.5}$$

where AE denotes the autoencoder network.

4.4 3D Reconstruction of Human Body from Single Image

The mesh autoencoder can act as a parametric model for shape generation and reconstruction. Similar to the approaches that utilise the SMPL model in deep neural networks [88][10], we use the decoder part of the autoencoder for reconstructing 3D meshes of human body from monocular images. More specifically, we use an image CNN to compute a latent code given an image, then the latent code is sent to the decoder to output a mesh. Except for the latent code, the CNN also predicts parameters of a weak perspective camera that projects the output 3D mesh back to the original input image. The network architecture is as illustrated in Fig. 4.4.

Given pairs of 2D image and the ground truth 3D mesh $(\mathbf{I}_i, \mathbf{V}_i)$ with 2D keypoint locations in the image as $\mathbf{J}_i^{2D}$, we train the network by minimising the following error:

$$L_{data} = L_{3D} + L_{proj} \tag{4.6}$$

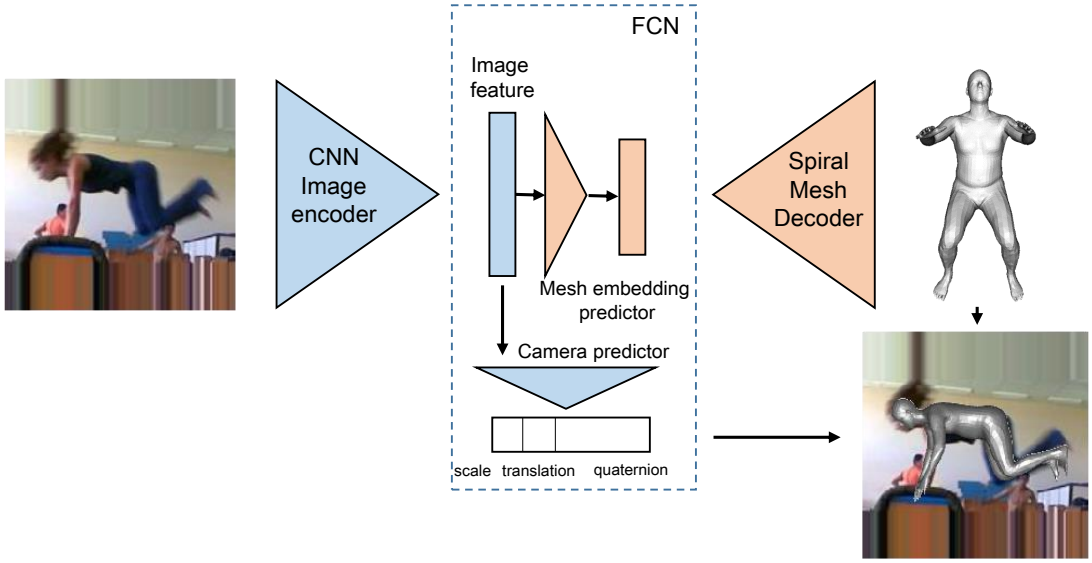


Figure 4.4: Illustration of our image to mesh network architecture. A CNN is used to encode image features given a RGB image. The image features are then sent to two separate fully connected network modules to compute the mesh embedding and camera embedding. The 3D mesh is then decoded from the mesh embedding with the spiral mesh decoder, and projected back to the image with the predicted camera parameters.

where

$$L_{3D} = \sum_{i=1}^{N} \|\mathbf{V}_i - D_{mesh}(E_{im}(\mathbf{I}_i))\| \tag{4.7}$$

is the L1 reconstruction loss of the 3D mesh and

$$L_{proj} = \sum_{i=1}^{N} \|\mathbf{J}_i^{2D} - J\mathbf{\Pi}(D_{mesh}(E_{im}(\mathbf{I}_i)), E_{cam}(E_{im}(\mathbf{I}_i)))\| \tag{4.8}$$

encourages the projected keypoints to match the ground truth annotations. Here $\mathbf{\Pi}(\mathbf{V}, \mathbf{cam})$ is the weak perspective projection operation.

During training, the network learns to regress vertex locations given the input images, however, this is a challenging task as the autoencoder is designed for the task of reconstruction instead of generation. In this case, the output shape might have a low vertex reconstruction error, but low fidelity, resulting in shapes with non-smooth or noisy surface. In order to enforce the network to generate valid shapes, we introduce a discriminator network. Unlike approaches that are based on the SMPL model, the latent space of our autoencoder do not correspond to any semantically meaningful

parameters, thus ground truth supervision for the latent space of 'real shape' is not available. Thus instead of discriminating the latent space like [10], we directly train the discriminator network on meshes.

While many GAN works use discriminator networks that output a logit indicating whether the input sample is real or fake, we use an mesh autoencoder network to compute the dense adversarial loss. Here we employ the setup of [102]. Suppose we have discriminator network D_{AE}, where for an input sample $\mathbf{V}$, the reconstruction loss given by D_{AE} is defined as follows:

$$L_{AE} = \|\mathbf{V} - D_{AE}(\mathbf{V})\| \tag{4.9}$$

The discriminator network is then trained to minimise the following loss:

$$L_{D_{AE}} = L_{AE}(\mathbf{V}_i) - k_t L_{AE}(D_{mesh}(E_{im}(\mathbf{I}_i)) \tag{4.10}$$

and an adversarial loss term is added to the image to mesh generator objective:

$$\begin{aligned} L &= L_{data} + L_{Gadv} \\ &= L_{data} + L_{AE}(D_{mesh}(E_{im}(\mathbf{I}_i)) \end{aligned} \tag{4.11}$$

The term k_t maintains the balance between the real and fake loss, at each training step t, the value of k_t is updated as:

$$k_t = k_{t-1} + \lambda_k(\gamma L_{AE}(\mathbf{V}_i) - L_{AE}(D_{mesh}(E_{im}(\mathbf{I}_i)))) \tag{4.12}$$

where γ and λ_k are hyperparameters that can be set experimentally.

4.5 Evaluation

In this section we provide experimental results of our proposed mesh recovery approach. We compare the results to several baseline method on the Human3.6M dataset, so as to investigate the effect of our proposed losses and different training setups. For all experiments we evaluate the performance quantitatively by reporting the per-vertex mean reconstruction error. Additionally we report the mean joint reconstruction error since this is the most standard error metric for most of the 3D body reconstruction methods. Finally we perform experiments on in the wild images and compare the

results to several popular approaches. We also provide qualitative results for all experiments.

4.5.1 Implementation Details

For all of our experiments we use ResNet-50 pretrained on ImageNet as the image encoder. The last layer of the ResNet is replaced with 2 fully connected layers that outputs 1024D features. The feature is sent to the latent code regressor to produce a latent vector of 128D, which is then used in the mesh decoder to generate a mesh. The camera branch takes the extracted image features as input, and outputs a 7D camera vector which contains a 4D quaternion rotation, 2D translation and a scaling factor. For the mesh decoder and autoencoder discriminator we use the architecture specified in Table 4.1.

All input images are cropped and scaled to 224x224 using the tight bounding boxes of the ground truth 2D keypoints while preserving the aspect ratio of the person. For training we use batch size of 16. The mesh generator is trained with Adam optimizer of learning rate 5×10^{-5}, while the autoencoder discriminator is trained with learning rate 10^{-5}. We use weight decay of 10^{-5} for both networks. We choose hyperparameter values of $\lambda_k = 0.001$ and $\gamma = 0.7$ for the discriminator, with the initial $k_0 = 0$.

4.5.2 Experimental Setup

In order to investigate the merit of our proposed mesh convolutional network over linear blend skinning based models, as well as the effectiveness of the autoencoder based discriminator, we train multiple networks with different mesh generation component (mesh convolutional decoder vs SMPL) and loss functions. State-of-the-art pose estimation methods often train on different combinations of datasets, as well as with different input formats (RGB, RGBD, part segmentations, 2D + 3D supervision, etc.). In order to obtain comparable results, we use only Human3.6M [103] for training and evaluation in this experiment. The ground truth 3D mesh annotations are obtained from MoSh [104]. We remove global rotations of the ground truth 3D mesh by setting the root joint rotation provided as SMPL parameter by MoSh to zero. For all configurations, We train on subjects S1, S5, S6, S7 and S8 for 20 epochs, and evaluate on subjects S9 and S11. For fast training we downsample the training set from 50fps to 5fps. The training images are randomly rotated and translated.

	Network	Discriminator	Loss	Pretrain
Config A	ResNet + SMPL	-	3D+parameter+2D	-
Config B [10]	ResNet + SMPL + IEF	[10]	3D+parameter+2D+adv	-
Config C	ResNet + Decoder	-	3D+2D	-
Config D	ResNet + Decoder	-	3D+2D	Decoder
Config E	ResNet + Decoder	Mesh AE	3D+2D+BEGAN adv	-
Config F	ResNet + Decoder	Mesh AE	3D+2D+BEGAN adv	Decoder + AE

Table 4.2: Our ablative experiments setup. 'AE' refers to our proposed mesh convolutional autoencoder. 'Decoder' refers to the mesh convolutional decoder. 'IEF' refers to the iterative error feedback component used in HMR[10]. 'BEGAN adv' refers to our proposed dense mesh autoencoder adversarial loss.

Table 4.2 lists the network structure and training loss that are used for our ablation study. Note that all configurations use the same ResNet pretrained on ImageNet, and the 'Pretrain' column only list the pretraining configuration of additional components.

Config A To assess the reconstruction ability of our mesh convolutional decoder, we train a baseline network that regresses the SMPL model parameters from ResNet. For this configuration, the latent code regressor outputs a 79D parameter vector, which correspond to the 10 shape parameters β and the 23 axis-angle rotation for the joints θ of SMPL model. Since the camera branch outputs a quaternion rotation, we did not use root joint rotation of the SMPL model. For training loss, we added the following SMPL parameter regression term:

$$\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|^2 + \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|^2 \tag{4.13}$$

where $\boldsymbol{\beta}, \boldsymbol{\theta}$ are the network's latent code prediction and $\boldsymbol{\beta}', \boldsymbol{\theta}'$ are the ground truth parameter vector obtained from MoSh. We have observed that without the parameter supervision the network would take longer to converge and tend to generate invalid shapes at the early stage of training. The network is trained with the Adam optimizer of learning rate 10^{-5} following [10].

Config B This configuration is the exact implementation of Human Mesh Recovery (HMR) [10], except that we now only train on our crops and downsamples of Human3.6M. Compare to **Config A**, an itarative error feedback (IEF) layer is added to the latent code regressor. Instead of regressing directly $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, the latent code regressor takes the image feature and current estimate of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, and outputs a

residual $\Delta\boldsymbol{\beta}$ and $\Delta\boldsymbol{\theta}$. The residual is then added to the previous estimate to produce an estimate of the current iteration. In our experiments, we use 3 iterations for the IEF layer. The final estimate are used for computing the 3D, parameter regression and 2D keypoint loss. The intermediate parameter estimates are sent to the factorised discriminator network to ensure each update produce a valid human body mesh. We use the same discriminator network implementation as [10], and train the discriminator with an Adam optimizer of learning rate 10^{-4} while training the generator network simultaneously as specified by **Config A**.

Config C For **Config C** we use the network structure as described in section 4.4 without using the discriminator network. This setup is to be compared with **Config A** to show the effectiveness of our mesh convolutional decoder over linear blend skinning method. However for the mesh decoder, there does not exist ground truth latent vector supervision, thus only 3D mesh reconstruction and 2D keypoint loss are used. While training this network, weights and bias for the mesh decoder are randomly initialised with a normal distribution.

Config D In this configuration the network structure and losses used for training are the same as **Config C**, except that we pretrain the mesh convolutional decoder and freeze its weights while training with images. Therefore the only learnable parameters during training are the ResNet parameters, the fully connected layers for image feature extraction, the latent code regressor and the camera parameter regressor.

The mesh decoder is pretrained by training a mesh convolutional autoencoder on a collection of ground truth 3D meshes plus synthetic meshes. We use randomly sampled SMPL shape and pose parameters from a mixture of experts prior proposed by [96] for generating synthetic meshes of valid body shape and poses. At each training step, we construct the input mini batch in a balanced manner such that it consists of half Human3.6M meshes and half synthetic meshes. The decoder part of this autoencoder is then plugged into the image to mesh network for further training.

Config E This configuration is our proposed method as described in section 4.4 and 4.5.1.

Config F The network structure used in this experiment is the same as **Config E**, except that the mesh decoder and mesh convolutional discriminator network is pretrained as in **Config D**. After pretraining the mesh autoencoder, the decoder

weights are used to initialise the mesh decoder in the image to mesh network, and the autoencoder weights are used to initialise the discriminator network. When training on images, we freeze the weights of the decoder part in the image to mesh network, as well as in the discriminator network.

4.5.3 Results and Discussion

Recons. Err.	Config A	Config B	Config C	Config D	Config E	Config F
Joint (mm)	96.37	88.75	88.62	90.66	**81.76**	85.76
Vertex (mm)	109.10	100.83	95.08	102.44	**93.69**	96.34

Table 4.3: Mean joint reconstruction errors and mean vertex reconstruction errors on Human3.6M dataset for the configurations of our ablation study described in Table 4.2. Using the proposed convolutional mesh decoder, simple mesh regression training (**Config C**) performs better compared to linear blend skinning based models (**Config A** and **B**). Adversarial training with mesh autoencoder further improves the results (**Config E** and **F**).

In Table 4.3 we report the mean joint and vertex reconstruction error on the Human3.6M testset. We observe that with our proposed mesh decoding method, naive mesh regression training (**Config C**) achieves comparable results to SMPL parameter regression with adversarial training (**Config B**) in terms of 3D joint reconstruction error, while performing better on per-vertex reconstruction error. Our proposed dense mesh autoencoder based adversarial training further improves the result (**Config E**).

Mesh Convolution vs Linear Blend Skinning

Fig. 4.5 visualises results of our proposed image to mesh network (**Config D**) compared to results obtained from the SMPL parameter regression baseline (**Config A**). As suggested by [10], directly predicting SMPL parameters is a difficult problem. While using SMPL parameters as regression target of the image encoder during training, small error in the axis angle representation of the joint rotations could lead to visually very different results. Such method also relies on ground truth parameter supervision while the discriminator network in **Config B** is not used. We have observed that when training without parameter supervision, even with per-vertex reconstruction loss, the network cannot be trained stably and will produce invalid shapes. On the contrary, our mesh decoder do not rely on latent vector supervision and can be directly trained to regress 3D vertex locations, which is more stable and reliable compared to the parameter regression method.

Adversarial Prior on parameters vs Dense adversarial loss

In Fig. 4.6 we visualise the reconstruction results of our proposed dense mesh autoencoder based adversarial training compared to the factorised adversarial prior. While the SMPL parameter regression result is improved by performing adversarial training on all the parameters predicted by the iterative error feedback loop, our autoencoder based dicriminator also improves the reconstruction quality of the mesh decoder. As a result, our proposed method achieves better reconstruction quality for both joints and vertices.

With or without pretraining

The numerical evaluation on Human3.6M suggests that our proposed network achieves lower reconstruction error when trained from scratch (**Config C** and **Config E**). In Fig. 4.8, we visualise some examples of reconstruction predicted by networks **Config F** (pretrained) and **Config E** (trained from scratch). During training, we observed that although the pretrained mesh autoencoder can produce perfect reconstruction by forward passing an input mesh through the encoder-decoder structure, while attempting to reconstruct a mesh with the decoder only, the reconstruction loss is often relatively high and requires more iterations to converge. In other words, error backpropagation from the output layer to the latent code layer is not as effective. Since the autoencoder is pretrained only for the reconstruction task, the latent space is highly unstructured due to the complexity of the human body variation space. As a result of lack of ground truth latent code supervision, it is difficult for the image encoder to learn to regress the latent code that will produce the desired output through the mesh decoder. This holds even when we attempt to finetune the pretrained mesh decoder together with the image decoder. On the other hand, the training converges faster while using randomly initialised weights for the mesh decoder. However, without pretraining, the mesh decoder often produces noisy and spiky surface, as the discriminator network needs more samples to be properly trained.

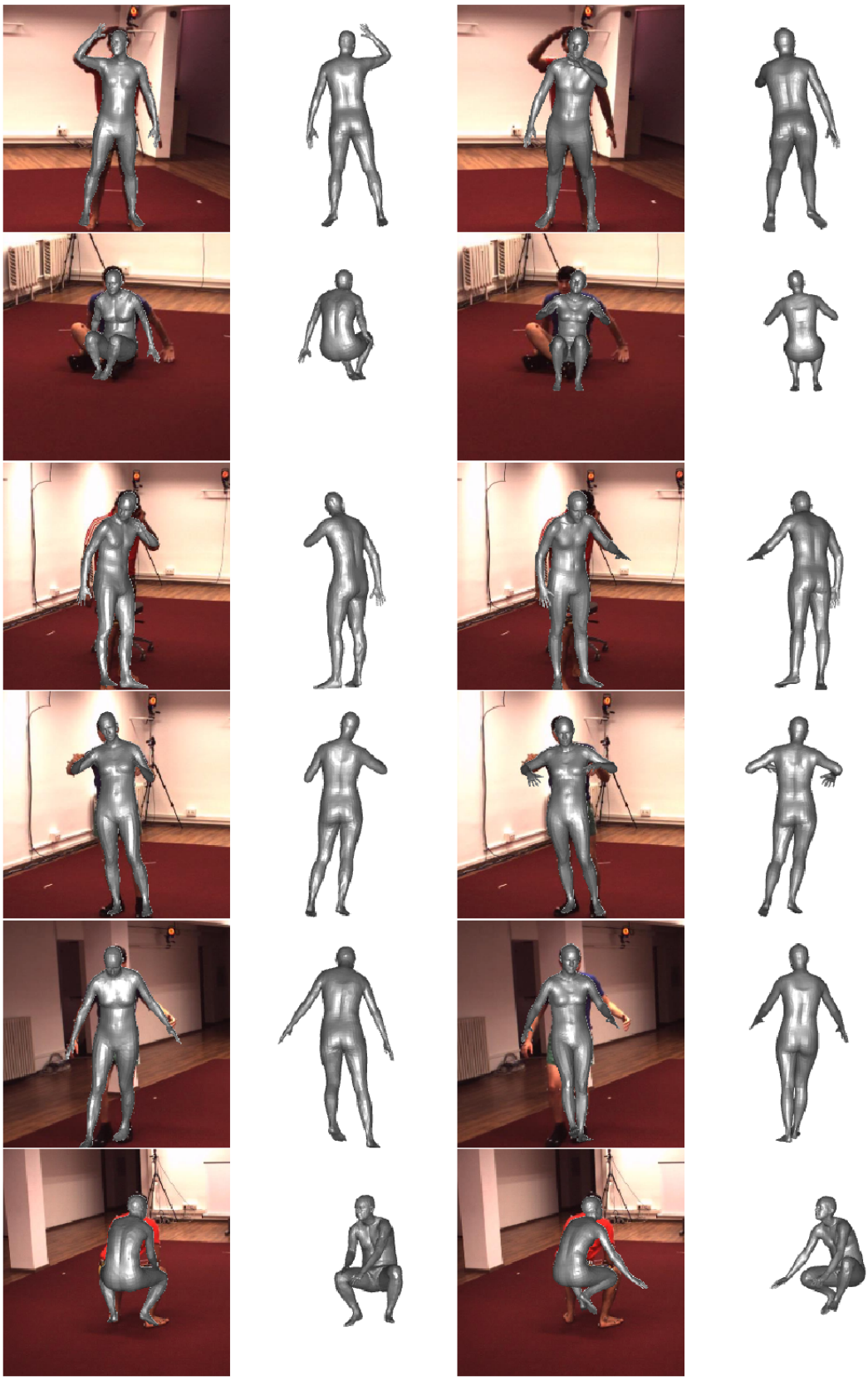


Figure 4.5: Left two columns: results from network D (our image to mesh network). Right two columns: results on the same image from network A (SMPL baseline).

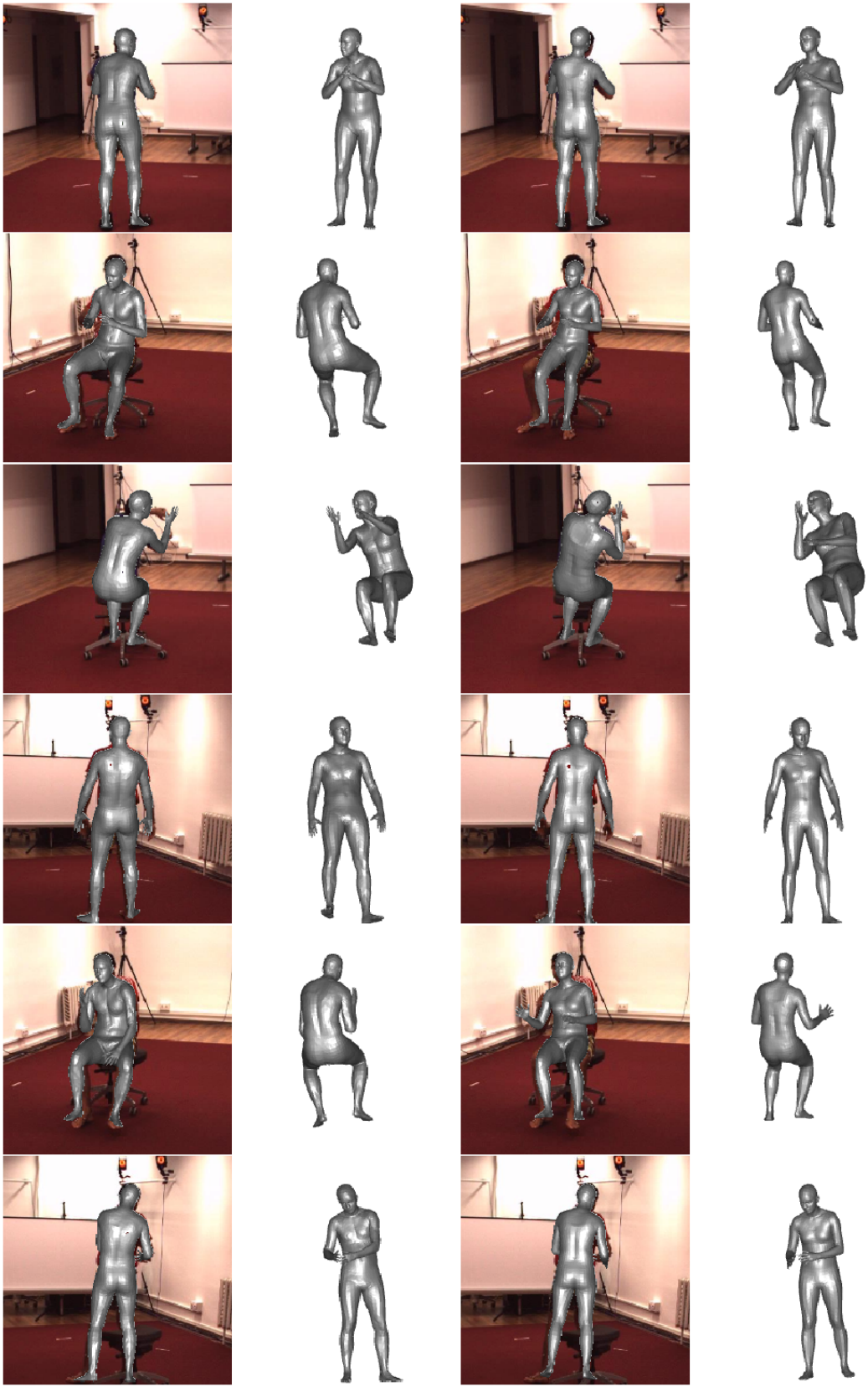


Figure 4.6: Left two columns: results from network F (our proposed approach). Right two columns: results on the same image from network B (HMR).

4.5.4 Results on in the wild images

For experiments on in the wild images, we train our network on the UP-3D [86] dataset. In order to compare the performance with various state-of-the-art approaches which report performance on the Human3.6M dataset, we also include the Human3.6M training set during training. In Table 4.4, we report per-vertex mean reconstruction error on UP-3D dataset, and per-joint mean reconstruction error on Human3.6M dataset (Protocol 2). We also show example reconstructions sampled from different error quartiles on the testsets (Fig. 4.7).

We observe that our approach outperforms optimisation based method by a large margin. This is also the case when comparing to learning based methods that use RGB images as input and regress SMPL model parameters (*SMPL param. reg. [88] and Pavlakos [87]), this suggests that the mesh convolutional decoder is a more suitable structure compared to linear blend skinning based models in deep neural networks in the task of mesh recovery from images. On Human3.6M dataset, Neural Body Fitting (NBF) [88] and HMR [10] achieved better performance than us, however NBF also use a proxy network to produce part segmentation maps as input to the pose estimation network, while HMR was trained on a large number of images with 2D keypoint supervisions and additionally 3D MoCap datasets. These network structure and training techniques can also be used to improve the performance of our proposed network, however we leave this for future work.

The most closely related work that we compare to is Convolutional Mesh Regression (CMR) [28], where they use the same image encoder to compute image features. The image features are then attached to the downsampled template mesh graph, and processed by a graph convolutional network and upsampled again to regress the 3D vertex locations. Compared to their network, our mesh is decoded from a latent vector through a fully connected layer, and the mesh is upsampled multiple times in a deeper mesh convolutional network. Although CMR achieves better joint reconstruction error on Human3.6M dataset, we observe that our network outputs high fidelity meshes, while the output of CMR is noisy and lack of details.

UP-3D	Vertex Error (mm)
Lassner et al. [86]	169.8
Pavlakos et al. [87]	117.7
* Tan et al. [105]	105
* SMPL param. reg. [88]	98.5
Ours	**80.2**

Human3.6M (Protocol 2)	Joint Error (mm)
Bogo et al. [66]	82.3
Lassner et al. [86]	80.7
Pavlakos et al. [87]	75.9
Ours	70.3
Kanazawa et al. [10]	56.8
Omran et al. [88]	59.9
Kolotouros et al. [28]	**50.1**

Table 4.4: Comparison to state-of-the-art on UP-3D dataset and Human3.6M dataset (Protocol 2). Errors are measured in millimetres. (*) indicates that the vertex loss is measured on a sparse set of points (landmark or keypoint loss).

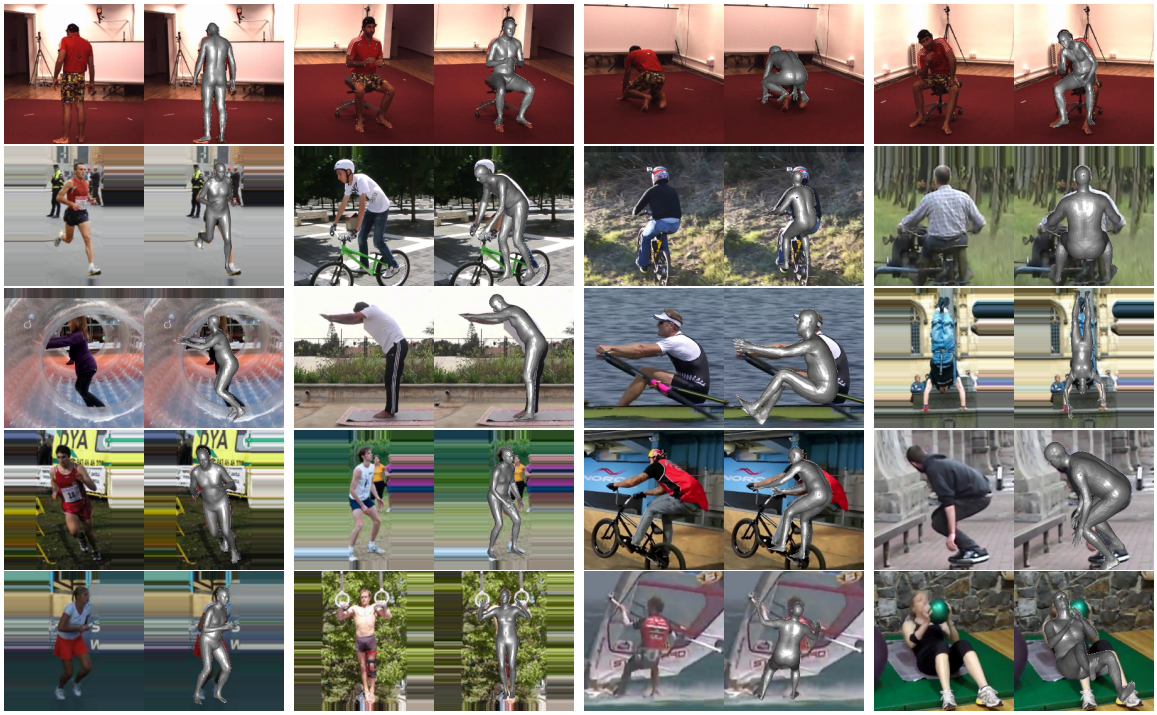


Figure 4.7: Qualitative results on Human3.6M Protocol 1 testset (1st row) and UP-3D testset by error quartile in terms of per-vertex mean reconstruction error. The columns show examples from different error quartiles, from left to right: 0-25%, 25-50%, 50-75%, 75-100%.

4.6 Conclusion

In this chapter we proposed a network structure that utilise a powerful deep mesh convolutional decoder to reconstruct 3D human body meshes from single RGB image input. We learn the shape and pose variabilities of the 3D human body space by pre-training a mesh convolutional autoencoder. The decoder part is then combined with

an image encoder and trained end-to-end to minimise the 3D vertex reconstruction loss. We also proposed to use the pretrained mesh autoencoder as a mesh prior to enforce plausible reconstructions.

We performed evaluations on Human3.6M and UP-3D dataset on the task of 3D pose estimation and compared our proposed approach to several baseline methods. Our network outperforms the baseline method that regresses SMPL model parameters, with considerable improvement while using our mesh autoencoder based discriminator in the training.

Our major contribution in this chapter is to demonstrate that mesh convolutions can outperform linear blend skinning based models in the task of human mesh recovery from images. The merit of mesh convolutional decoder is that it can be trained and fine-tuned end-to-end together with the image encoder, while the most commonly used linear blend skinning models are typically fixed when used in a deep neural network, restricting the representational power of the network to the model space.

The performance of our network can be further improved by training on more data in weakly supervised manner or with synthetic data, as well as by using a more complicated image encoding network structure that uses intermediate image representations. With more 3D ground truth data, our method can also be used to reconstruct more detailed meshes with different facial expressions and hand poses.

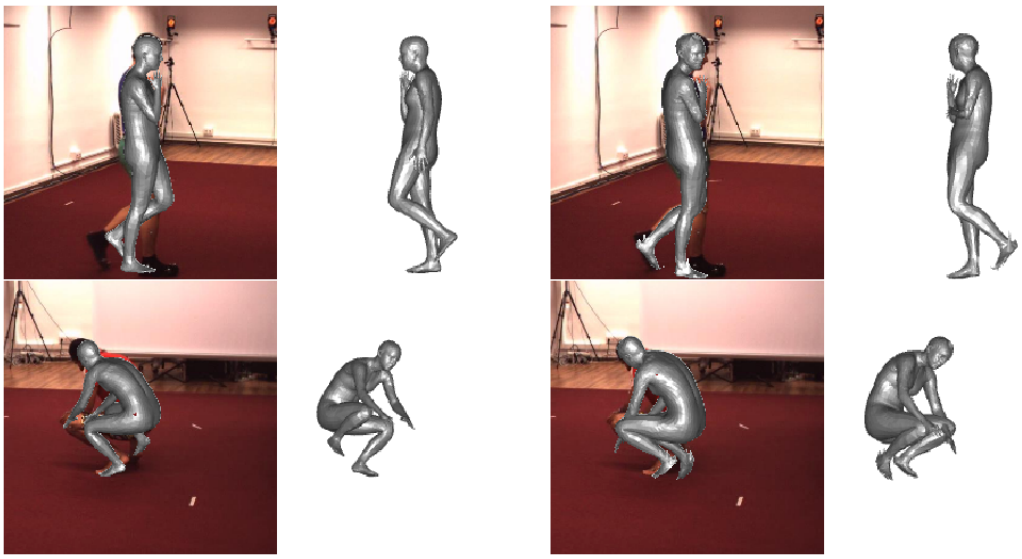


Figure 4.8: Qualitative results from network Config F (with mesh decoder pretraining) and Config E (training from scratch). From left to right: results of Config F network overlaid on the input image, Config F result from a different angle, results of Config E network, Config E result from a different angle.

Chapter 5

BLSM: A Bone-Level Skinned Model of the Human Mesh

In this chapter we introduce BLSM, a bone-level skinned model of the human body mesh where bone scales are set prior to template synthesis, rather than the common, inverse practice. BLSM first sets bone lengths and joint angles to specify the skeleton, then specifies identity-specific surface variation, and finally bundles them together through linear blend skinning. We design these steps by constraining the joint angles to respect the kinematic constraints of the human body and by using accurate mesh convolution-based networks to capture identity-specific surface variation.

We provide quantitative results on the problem of reconstructing a collection of 3D human scans, and show that we obtain improvements in reconstruction accuracy when comparing to a SMPL-type [3] baseline. Our decoupled bone and shape representation also allows for out-of-box integration with standard graphics packages like Unity, facilitating full-body Augmented Reality (AR) effects and image-driven character animation.

5.1 Introduction

Mesh-level representations of the human body form a bridge between computer graphics and computer vision, facilitating a broad array of applications in motion capture, monocular 3D reconstruction, human synthesis, character animation, and augmented reality. The articulated human body deformations can be captured by rigged modelling where a skeleton animates a template shape; this is used in all graphics packages for human modelling and animation, and also in state-of-the-art statistical models such

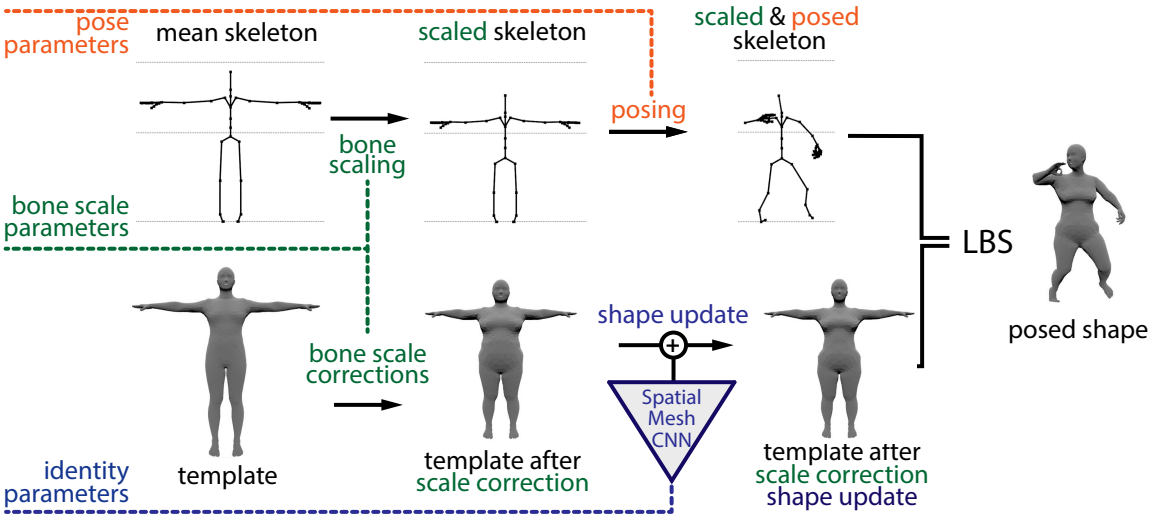


Figure 5.1: Overview of our Bone-Level Skinned Model (BLSM): The top row shows skeleton synthesis: starting from a canonical, bind pose, we first scale the bone lengths and then apply an articulated transformation. The bottom row shows shape control: the canonical mesh template is affected by the bone scaling transform through Bone-Scaling Blend Shapes, and then further updated to capture identity-specific shape variation. The skeleton drives the deformation of the resulting template through Linear Blend Skinning, yielding the posed shape.

as SMPL or SCAPE [3, 30].

Our work aims at increasing the accuracy of data-driven rigged mesh representations. Our major contribution consists in revisiting the template synthesis process prior to rigging. Current models, such as SMPL, first synthesize the template mesh in a canonical pose through an expansion on a linear basis. The skeleton joints are then estimated post-hoc by regressing from the synthesized mesh to the joints. Our approach instead disentangles bone length variability from acquired body traits dependent e.g. on exercise or dietary habits.

Based on this, we first model bone length-driven mesh variability in isolation, and then combine it with identity-specific updates to represent the full distribution of bodies. As we show experimentally, this disentangled representation results in more compact models, allowing us to obtain highly-accurate reconstructions with a low parameter count.

Beyond this intuitive motivation, decoupling bone lengths from identity-specific variation is important when either is fixed; e.g. when re-targeting an outfit to a person we

can scale the rigged outfit's lengths to match those of the person, while preserving the bone length-independent part of the outfit shape. In particular, we model the mesh synthesis as the sequential specification of identity-specific bone length, pose-specific joint angles, and identity-specific surface variation, bundled together through linear blend skinning.

We further control and strengthen the individual components of this process: Firstly, we constrain joint angles to respect the kinematic constraints of human body, reducing body motion to 47 pose atoms, amounting to joint rotations around a single axis. Alternative techniques require either restricting the form of the regressor [96] or penalizing wrong estimates through adversarial training [10].

Secondly, we introduce accurate mesh convolution-based networks to capture identity-specific surface variation. We show that these largely outperform their linear basis counterparts, demonstrating for the first time the merit of mesh convolutions in rigged full-body modelling (earlier works [106] were applied to the setup of the face mesh).

We provide quantitative results on the problem of reconstructing a collection of 3D human scans, and show that we obtain systematic gains in average vertex reconstruction accuracy when comparing to a SMPL-type baseline. We note that this is true even though we do not use the pose-corrective blendshapes of [3]; these can be easily integrated, but we leave this for future work.

Beyond quantitative evaluation, we also show that our decoupled bone and shape representation facilitates accurate character animation in-the-wild. Our model formulation allows for out-of-box integration with standard graphics packages like Unity, leading to full-body augmented reality experiences.

The rest of the chapter is structured as follows: we first provide a brief literature survey about 3D human mesh models and mesh convolutional networks. We then describe in details our proposed model formulation, followed by technical approach on model training, including the registration method we used to process our training data. Finally we provides quantitative and qualitative evaluations on the task of 3D scan reconstructions and character animation in the wild.

5.2 Related Work

3D Human Body Modelling

Linear Blend Skinning (LBS) is widely used to model 3D human bodies due to its ability to represent articulated motions. Some early works have focused on synthesizing realistic 3D humans by modifying the LBS formulation. Pose Space Deformation (PSD) [107] defines deformations as a function of articulated pose. [108] use the PSD approach learned from 3D scans of real human bodies. Other authors have focused on learning parametric model of human body shapes independently from the pose [41, 33, 109]. Following these works, [30, 110, 23, 31, 111] model both body shape and pose changes with triangle deformations. These work has been extended to also model dynamic soft-tissue motion [112].

Closely related to [113], SMPL [3] propose an LBS-based statistical model of the human mesh, working directly on a vertex coordinate space: T-posed shapes are first generated from a PCA-based basis, and then posed after updating joint locations. More recent works have focused on improving the representational power of the model by combining part models, e.g. for face and hands [114, 115], without however modifying the body model. One further contribution of [3] consists in handling artefacts caused by LBS around the joints when posing the template through the use of pose-corrective blend shapes [3]. Our formulation can be easily extended to incorporate these, but in this work we focus on our main contribution which is modelling of the shape at the bone level.

Graph Convolutions for 3D Human Bodies

Different approaches have been proposed to extend convolutional neural networks to non-euclidean data such as graphs and manifolds [51, 100, 26, 7, 27, 6]. Among these, [100, 26, 27, 7] have attempted to model and reconstruct 3D human bodies using convolutional operators defined on meshes. While these methods achieve good performance on shape reconstruction and learning correspondences, their generalisation is not comparable to LBS based methods. Furthermore, the process of synthesising new articulated bodies using mesh convolutional networks is not easy to control since the latent vector typically encodes both shape and pose information.

5.3 Bone-Level Skinned Model

We start with a high-level overview, before presenting in detail the components of our approach. As shown in Fig. 5.1, when seen as a system, our model takes as input bone scales, joint angles, and shape coefficients and returns an array of 3-D vertex locations. In particular, BLSM operates along two streams, whose results are combined in the last stage. The upper stream, detailed in Sec. 5.3.1, determines the internal skeleton by first setting the bone scales through bone scaling coefficients $\mathbf{c}_b$, delivering a bind pose. This is in turn converted to a new pose by specifying joint angles $\boldsymbol{\theta}$, yielding the final skeleton $T(\mathbf{c}_b, \boldsymbol{\theta})$.

The bottom stream, detailed in Sec. 5.3.2, models the person-specific template synthesis process: Starting from a mesh corresponding to an average body type, $\bar{\mathbf{V}}$, we first absorb the impact of bone scaling by adding a shape correction term, $\mathbf{V}_b$. This is in turn augmented by an identity-specific shape update $\mathbf{V}_s$, modelled by a mesh-convolutional network. The person template is obtained as

$$\mathbf{V} = \bar{\mathbf{V}} + \mathbf{V}_b + \mathbf{V}_s \,. \tag{5.1}$$

Finally, we bundle the results of these two streams using Linear Blend Skinning, as described in Sec. 5.3.3, delivering the posed template $\hat{\mathbf{V}}$:

$$\hat{\mathbf{V}} = LBS(\mathbf{V}, T(\mathbf{c}_b, \boldsymbol{\theta})) \,. \tag{5.2}$$

5.3.1 Skeleton Modeling

Kinematic Model

Our starting point for human mesh modelling is the skeleton. As is common in graphics, the skeleton is determined by a tree-structured graph that ties together human bones through joint connections.

Starting with a single bone, its 'bind pose' is expressed by a template rotation matrix $\mathbf{R}^t$ and translation vector $\mathbf{O}^t$ that indicate the displacement and rotation between the coordinate systems at the two bone joints. We model the transformation with respect to the bind pose through a rotation matrix $\mathbf{R}$ and a scaling factor s, bundled together in a 4×4 matrix $\mathbf{T}$:

$$\mathbf{T} = \underbrace{\begin{bmatrix} sI & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & 0 \\ 0 & 1 \end{bmatrix}}_{\text{deformation}} \underbrace{\begin{bmatrix} \mathbf{R}^t & \mathbf{O}^t \\ 0 & 1 \end{bmatrix}}_{\text{resting bone}}. \tag{5.3}$$

We note that common models for character modelling use $s = 1$ and only allow for limb rotation. Any change in object scale, or bone length is modelled by modifying the displacement at the bind pose, $\mathbf{O}$. This is done only implicitly, by regressing the bind pose joints from a 3D synthesized shape. By contrast, our approach gives us a handle on the scale of a limb through the parameter s, making the synthesis of the human skeleton explicitly controllable.

The full skeleton is constructed recursively, propagating from the root node to the leaf nodes along a kinematic chain. Every bone transformation encodes a displacement, rotation, and scaling between two adjacent bones, i and j, where i is the parent and j is the child node. To simplify notation, we will describe the modelling along a single kinematic chain, meaning $j = i + 1$, and denote the local transformation of a bone by $\mathbf{T}^i$.

The global transformation $\mathbf{T}_j$ from the local coordinates of bone j to world coordinates is given by: $\mathbf{T}_j = \prod_{i \leq j} \mathbf{T}^i$, where we compose the transformations for every bone on the path from the root to the j-th node. This product accumulates the effects of consecutive transformations: for instance a change in the scale of a bone will incur the same scaling for all of its descendants. These descendants can in turn have their own scale parameters, which are combined with those of their ancestors. The 3D position of each bone j can be read from the last column of $\mathbf{T}_j$, while the upper-left 3×3 part of $\mathbf{T}_j$ provides the scaling and orientation of its coordinate system.

Parametric Bone Scaling

We model human proportions by explicitly scaling each bone. For this we perform PCA on bone lengths, as detailed in Sec. 5.4.2 and use the resulting principal components to express individual bone scales as:

$$\mathbf{b} = \bar{\mathbf{b}} + \mathbf{c}_b \mathbf{P}_b \tag{5.4}$$

where $\mathbf{c}_b$ are the bone scaling coefficients, $\mathbf{P}_b$ is the bone-scaling matrix, and $\bar{\mathbf{b}}$ is the mean bone scale.

From Eq. 5.4 we obtain individual bone scales. However, the bone scales s that appear in Eq. 5.3 are meant to be used through the kinematic chain recursion, meaning that the product of parent scales delivers the actual bone scale, $\mathbf{b}_j = \prod_{i \leq j} s_i$; this can be used to transform the predictions of Eq. 5.4 into a form that can be used in Eq. 5.3:

$$s_i = \begin{cases} \mathbf{b}_i / \mathbf{b}_{i-1}, & i > 0 \\ 1 & i = 0 \end{cases} \tag{5.5}$$

Kinematically Feasible Posing

We refine our modeling of joint angles to account for the kinematic constraints of the human body. For instance the knee has one degree of freedom, the wrist has two, and the neck has three. For each joint we set the invalid degrees of freedom to be identically equal to zero, and constrain the remaining angles to be in a plausble range (e.g. ± 45 degrees for an elbow). In Fig. 5.2 we show sample meshes synthesized by posing a template along one valid degree of freedom.

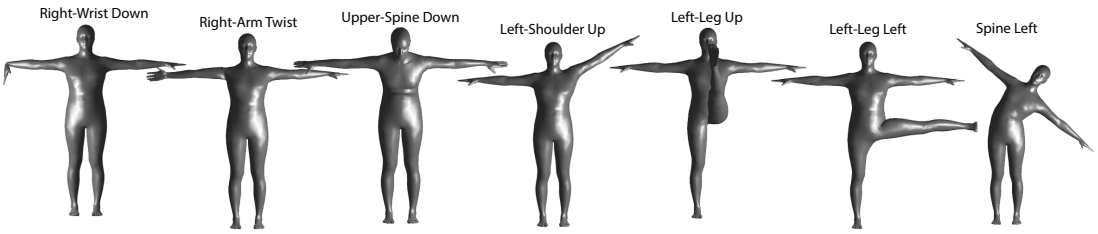


Figure 5.2: 7 out of the 47 degrees of freedom corresponding to kinematically feasible joint rotations for our skeleton.

For this, for each such degree of freedom we use an unconstrained variable $x \in R$ and map it to a valid Euler angle $\theta \in [\theta_{min}, \theta_{max}]$ by using a hyperbolic tangent unit:

$$\theta = \frac{\theta_{max} - \theta_{min}}{2} \tanh(x) + \frac{\theta_{min} + \theta_{max}}{2} \tag{5.6}$$

This allows us to perform unconstrained optimization when fitting our model to data, while delivering kinematically feasible poses. The resulting per-joint Euler angles are converted into a rotation matrix, delivering the matrix R in Eq. 5.3.

Using Eq. 5.6 alleviates the need for restricting the regressor form [96] or adversarial training [10], while at the same time providing us with a compact, interpretable

dictionary of 47 body motions.

5.3.2 Template Synthesis

Having detailed skeleton posing, we now turn to template synthesis. We start by modeling the effect of bone length on body shape, and then turn to modelling identity-specific variability.

Bone-Dependent Shape Variations

Bone length can be used to account for a substantial part of body shape variability. For example, longer bones correlate with a male body-shape, while limb proportions can correlate with ectomorph, endomorph and mesomorph body-type variability. We represent the bone-length dependent deformation of the template surface through a linear update:

$$\mathbf{V}_b = \mathbf{c}_b \mathbf{P}_{bc} \tag{5.7}$$

where $\mathbf{P}_{bc}$ is the matrix of bone-corrective blendshapes.

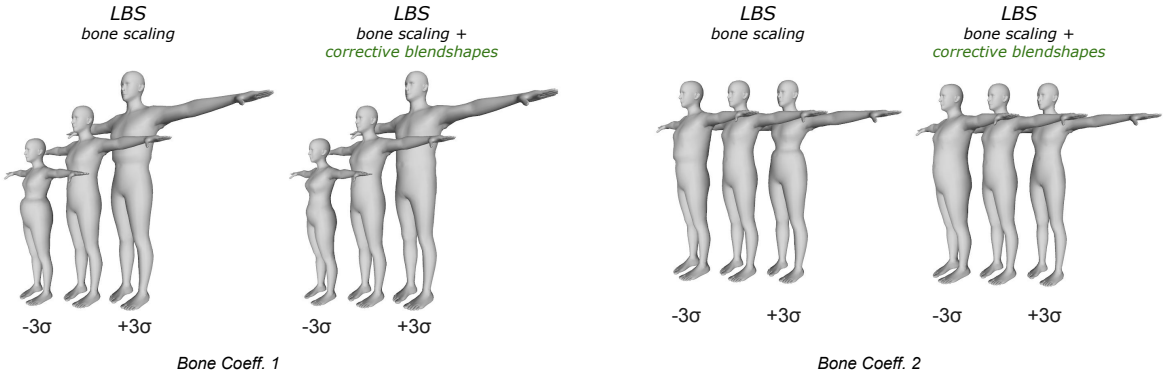


Figure 5.3: Impact of bone length variation on the template. Plain linear blend skinning results in artifacts. The linear, bone-corrective blendshapes eliminate these artifacts, and capture correlations of bone lengths with gender and body type.

Graph Convolutional Shape Modelling

Having accounted for the bone length-dependent part of shape variability, we turn to the remainder of the person-specific variability. The simplest approach is to use a linear update:

$$\mathbf{V}_s = \mathbf{c}_s \mathbf{P}_s \tag{5.8}$$

where $\mathbf{c}_s$ are the shape coefficients, and $\mathbf{P}_s$ is the matrix of shape components; we refer to this baseline as the linear model. By contrast, we propose a more powerful, mesh-convolutional update. For this we use multi-layer mesh convolution decoder that precisely models the nonlinear manifold of plausible shapes in its output space.
We represent the triangular mesh as a graph $(\mathbf{V}, \mathcal{E})$ with vertices $\mathbf{V}$ and edges $\mathcal{E}$ and denote the convolution operator on a mesh as:

$$(f \star g)_x = \sum_{l=1}^{L} g_l f(x_l) \tag{5.9}$$

where g_l denotes the filter weights, and $f(x_l)$ denotes the feature of the l-th neighbour of vertex x. The neighbours are ordered consistently across all meshes, allowing us to construct a one-to-one mapping between the neighbouring features and the filter weights. Here we adapt the setting of [7], where the ordering is defined by a spiral starting from the vertex x, followed by the d-ring of the vertex, i.e. for a vertex x, x_l is defined by the ordered sequence:

$$S(x) = \{x, R_1^1(x), R_2^1(x), ..., R_{\|R^h\|}^h\}, \tag{5.10}$$

where h is the patch radius and $R_j^d(x)$ is the j-th element in the d-ring.

We use a convolutional mesh decoder to model the normalised deformations from the bone-updated shape. The network consists of blocks of convolution-upsampling layers similar to [6]. We pre-compute the decimated version of the template shape with quadratic edge collapse decimation to obtain the upsampling matrix. Given the latent vector $\mathbf{z}$, shape variation is represented as

$$\mathbf{V}_s = \mathcal{D}(\mathbf{z}) \tag{5.11}$$

where $\mathcal{D}$ is the learned mesh convolutional decoder.

5.3.3 Linear Blend Skinning

Having detailed the skeleton and template synthesis processes, we now turn to posing the synthesized template based on the skeleton. We use Linear Blend Skinning (LBS), where the deformation of a template mesh $\mathbf{V}$ is determined by the transformations of the skeleton. We consider that the bind pose of the skeleton is described by the matrices $\hat{\mathbf{T}}_j$, where the 3D mesh vertices take their canonical values $\mathbf{v}_i \in \mathbf{V}$, while the

target pose is described by $\mathbf{T}_j$.

According to LBS, each vertex is influenced by every bone j according to a weight w_{ij}; the positions of the vertices $\hat{\mathbf{v}}_i$ at the target pose are given by:

$$\hat{\mathbf{v}}_k = \sum_j w_{ij} \mathbf{T}_j \hat{\mathbf{T}}_j^{-1} \mathbf{v}_k. \tag{5.12}$$

In the special case where $\mathbf{T} = \hat{\mathbf{T}}$, we recover the template shape, while in the general case, Eq. 5.12 can be understood as first charting every point $\mathbf{v}_k$ with respect to the bind bone (by multiplying it with $\hat{\mathbf{T}}_j^{-1}$), and then transporting to the target bone (by multiplying with $\mathbf{T}_j$).

5.4 Model training

Having specified BLSM, we now turn to learning its parameters from data. For this we use the Civilian American and European Surface Anthropometry Resource (CAESAR) dataset [12] to train the shape model, which contains high resolution 3D scans of 4400 subjects wearing tight clothing. This minimal complexity due to extraneous factors has made CAESAR appropriate for the estimation of statistical body models, such as SMPL. For training skinning weights we use D-FAUST [11] dataset. Our training process consists in minimising the reconstruction error of CAESAR and D-FAUST through BLSM.

Since BLSM is implemented as a multi-layer network in pytorch, one could try to directly minimize the reconstruction loss with respect to the model parameters using any standard solver. Unfortunately however, this is a nonlinear optimisation problem with multiple local minima; we therefore use a carefully engineered pipeline that solves successively demanding optimization problems, as detailed below, and use automatic differentiation to efficiently compute any derivatives required during optimization.

5.4.1 Unconstrained Landmark-based Alignment

Each CAESAR scan $\mathbf{S}^n$ is associated with 73 anatomical landmarks, $\mathbf{X}_{lm}^n$ that have been localised in 3D. We start by fitting our template to these landmarks by gradient descent on the joint angles $\boldsymbol{\theta}^n$ and bone scales $\mathbf{s}^n$, so as to minimize the 3D distances between the landmark positions and the respective template vertices. More specifically,

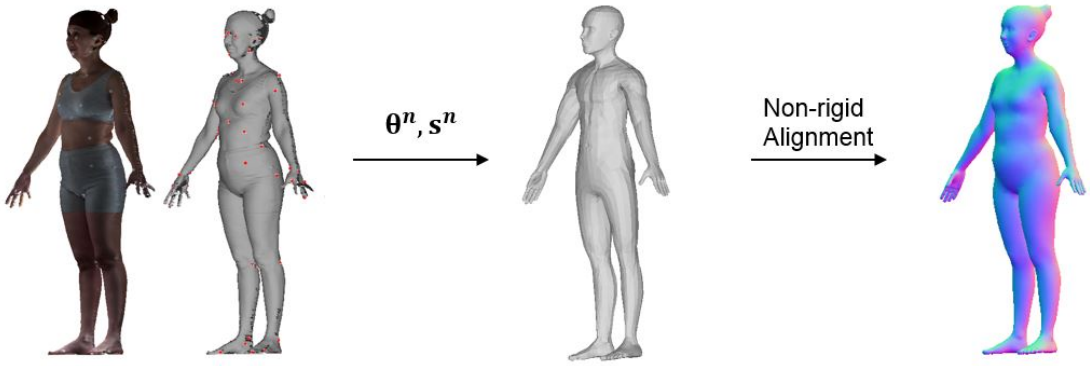


Figure 5.4: Example of one registered CAESAR instance. We first fit our rigged template to the landmarks on the scan surface by optimising over the joint angles and bone scales, then non-rigidly deform the vertices freely to align the scan surface.

the following optimization problem is solved:

$$\boldsymbol{\theta}^n, \mathbf{s}^n = \text{argmin}_{\boldsymbol{\theta},\mathbf{s}} \|\mathbf{A}_{lm}\, LBS(\mathbf{V}_T, T(\mathbf{s}, \boldsymbol{\theta})) - \mathbf{X}_{lm}^n\|^2 \tag{5.13}$$

where $\mathbf{A}_{lm}$ selects the subset of landmarks from the template.

This delivers an initial fitting which we further refine by registering our BLSM-based prediction $\hat{\mathbf{S}}_T^n = LBS(\mathbf{V}_T, T(\mathbf{s}^n, \boldsymbol{\theta}^n))$ to each scan $\mathbf{S}^n$. The final registration $\hat{\mathbf{S}}^n = LBS(\mathbf{V}_T + \mathbf{V}_D^n, T(\mathbf{s}^n, \boldsymbol{\theta}^n))$ is obtained by iteratively solving for $\mathbf{V}_D^n$ that minimises the following error:

$$E = E_{data} + E_{smoothness} + E_{reg} + E_{rigid} \tag{5.14}$$

where E_{data} is the data term, $E_{smoothness}$ enforces the surface of the registered mesh to be smooth, E_{reg} is the regularization term, and E_{rigid} enforces some parts of $\hat{\mathbf{S}}_{BLSM}^n$ to deform rigidly.

Data Loss The data loss encourages the surface of $\hat{\mathbf{S}}^n$ to be as close to $\mathbf{S}^n$ as possible. At each iteration, we sample 50k points $\mathbf{S}_p^n$ on the surface of $\mathbf{S}^n$, and 20k points $\hat{\mathbf{S}}_p^n$ on the surface of $\hat{\mathbf{S}}^n$. The data loss is then defined as:

$$E_{data} = \lambda_{mnn} \sum_{(\mathbf{v}_r, \mathbf{v}_s) \in M(\hat{\mathbf{S}}_p^n, \mathbf{S}_p^n)} \|\mathbf{v}_r - \mathbf{v}_s\|^2 + \lambda_{lm} \|\mathbf{A}\hat{\mathbf{S}}^n - \mathbf{X}_{lm}^n\|^2 \tag{5.15}$$

where $M(\hat{\mathbf{S}}_p^n, \mathbf{S}_p^n)$ denotes the set of mutual nearest neighbours (MNNs) between the two sampled pointcloud:

$$M(\hat{\mathbf{S}}_p^n, \mathbf{S}_p^n) = \{(\mathbf{v}_r, \mathbf{v}_s) \in (\hat{\mathbf{S}}_p^n, \mathbf{S}_p^n) | NN(\mathbf{v}_s, \hat{\mathbf{S}}_p^n) == \mathbf{v}_r \wedge NN(\mathbf{v}_r, \mathbf{S}_p^n) == \mathbf{v}_s\} \quad (5.16)$$

and $NN(\mathbf{v}, \mathbf{S})$ return the nearest neighbour of $\mathbf{v}$ in a set of points $\mathbf{S}$. Mutual nearest neighbours are used here since it is robust to holes on the scan compared to chamfer loss. (Fig. 5.5) illustrates an example. MNNs use sparse yet more reliable correspondences, and since we use densely sampled points on the surface, MNNs are sufficient to drive the template vertices towards the scan surface. To account for potential missing correspondences for the template vertices, we introduce several regularisation terms, which are explained in the following paragraphs.

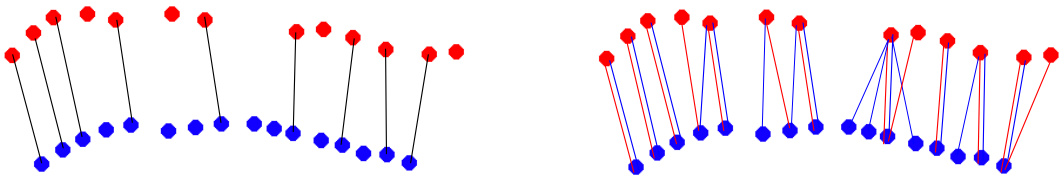


Figure 5.5: Comparison between mutual nearest neighbour (MNN) loss and chamfer loss. Red represents points on the scan, and blue for points on the model. Left: MNN loss. Right: chamfer loss. With chamfer loss the blue points would collapse around the edge of holes, while MNN loss only considers more confident correspondences.

Smoothness Loss The smoothness term $E_{smoothness}$ penalize non-smooth surfaces. Here we use mesh Laplacian smoothing. Denote each registration $\hat{\mathbf{S}}^n = (\mathbf{V}, \mathcal{E})$ where $\mathbf{V} \in \mathbb{R}^{N \times 3}$ is the set of vertices and $\mathcal{E}$ is the set of edges, the Laplacian δ_i for vertex $\mathbf{v}_i$ is defined as [116]:

$$\delta_i = \sum_{(i,j) \in \mathcal{E}} w_{ij}(\mathbf{v}_j - \mathbf{v}_i) \quad (5.17)$$

where

$$w_{ij} = \frac{\omega_{ij}}{\sum_{(i,k) \in \mathcal{E}} \omega_{ik}} \quad (5.18)$$

Here we use 'uniform laplacian' where $\omega_{ij} = 1$. The Laplacian can then be written in the matrix form:

$$\Delta(\hat{\mathbf{S}}^n) = \mathbf{LV} \quad (5.19)$$

where $\mathbf{L}$ is the $N \times N$ matrix:

$$\mathbf{L}_{ij} = \begin{cases} -1, & i = j \\ w_{ij}, & (i,j) \in \mathcal{E} \\ 0, & otherwise \end{cases} \tag{5.20}$$

The mesh smoothness loss is then defined as:

$$E_{smoothness} = \lambda_{smoothness} \|\Delta(\hat{\mathbf{S}}^n)\| \tag{5.21}$$

Regularization Loss The regularization loss is defined as:

$$E_{reg} = \lambda_{EV} \|EV(\hat{\mathbf{S}}^n) - EV(\hat{\mathbf{S}}_T^n)\|^2 + \lambda_{EL} \|EL(\hat{\mathbf{S}}^n) - EL(\hat{\mathbf{S}}_T^n)\|^2 \tag{5.22}$$

where $EV(\mathbf{S})$ is the function that computes the edges for a given mesh $\mathbf{S}$ with edges $\mathcal{E} = (i1, j1), ...(ie, je)$:

$$EV(\mathbf{S}) = [\mathbf{v}_{i1} - \mathbf{v}_{j1}, ..., \mathbf{v}_{ie} - \mathbf{v}_{je}]^T \tag{5.23}$$

This term enforces neighbouring vertices that share the same edge to have similar deformations. And $EL(\mathbf{S})$ returns the lengths of all such edges:

$$EL(\mathbf{S}) = [\|\mathbf{v}_{i1} - \mathbf{v}_{j1}\|^2, ..., \|\mathbf{v}_{ie} - \mathbf{v}_{je}\|^2]^T \tag{5.24}$$

which preserves the topology of the template.

Rigid Loss The CAESAR meshes do not have landmarks for fingers, therefore the landmark based BLSM alignment do not capture pose of the subjects' hands correctly, and subsequently the registered hands will be noisy. In order to avoid these artefacts, we enforce the hands to deform rigidly. This is done by adding extra weight for the regularization loss described in the previous paragraph for the edges that belongs to the hands:

$$E_{rigid} = \lambda_{rigid}(\|EV_H(\hat{\mathbf{S}}^n) - EV_H(\hat{\mathbf{S}}_T^n)\|^2 + \|EL_H(\hat{\mathbf{S}}^n) - EL_H(\hat{\mathbf{S}}_T^n)\|^2) \tag{5.25}$$

where EV_H and EL_H denotes the edge losses masked for the hand edges only.

In our experiment we use $\lambda_{mnn} = \lambda_{lm} = 1$, $\lambda_{smoothness} = 0.007$, $\lambda_{EV} = 0.003$, $\lambda_{EL} = 0.02$ and $\lambda_{rigid} = 0.01$. The registration process is implemented with Py-

torch3D [117] and thus benefits from efficient optimisation on GPU. We use the Adam optimizer with initial learning rate of 0.005 to optimize $\mathbf{V}_D^n$, and multiply the learning rate by 0.9 on plateau.

Note that this alignment stage does not use yet a statistical model to constrain the parameter estimates, and as such can be error-prone; the following steps recover shapes that are more regularized, but the present result acts like a proxy to the scan that is in correspondence with the template vertices.

5.4.2 Bone Basis and Bone-corrective Blendshapes

We start learning our model by estimating a linear basis for bone scales. For each shape $\hat{\mathbf{S}}^{\mathbf{n}}$ we estimate the lengths of the bones obtained during the optimization process described in the previous section.

We perform PCA on the full set of CAESAR subjects and observe that linear bases capture 97% of bone length variability on the first three eigenvectors. We convert the PCA-based mean vector and basis results from bone lengths into the mean bone scaling factor $\bar{\mathbf{b}}$ and bone scaling basis $\mathbf{P}_b$ used in Eq. 5.4 by dividing them by the mean length of the respective bone along each dimension.

Having set the bone scaling basis, we use it as a regularizer to re-estimate the pose $\boldsymbol{\theta}^n$ and bone scale coefficients $\mathbf{c}_b^n$ used to match our template $\mathbf{V}_T$ to each registration $\hat{\mathbf{S}}^n$ by solving the following optimisation problem:

$$\boldsymbol{\theta}^n, \mathbf{c}_b^n = \text{argmin}_{\boldsymbol{\theta}, \mathbf{c}_b} \| LBS(\mathbf{V}_T, T(\mathbf{c}_b, \boldsymbol{\theta})) - \hat{\mathbf{S}}^n \|^2 \tag{5.26}$$

Finally we optimize over the bone-corrective basis $\mathbf{P}_b$ and mean shape $\bar{\mathbf{V}}$:

$$\mathbf{P}_{bc}^*, \bar{\mathbf{V}}^* = \text{argmin}_{\mathbf{P}_b, \bar{\mathbf{V}}} \sum_{n=1}^{N} \| LBS(\bar{\mathbf{V}} + \mathbf{c}_b^n \mathbf{P}_{bc}, T(\mathbf{c}_b^n, \boldsymbol{\theta}^n)) - \hat{\mathbf{S}}^n \|^2 \tag{5.27}$$

Given that $\mathbf{V}_T$ and $\hat{\mathbf{S}}^n$ are in one-to-one correspondence, we no longer need ICP to optimize Eq. 5.26 and Eq. 5.27, allowing us instead to exploit automatic differentiation and GPU computation for gradient descent-based optimization.

5.4.3 Shape Blendshapes

Once bone-corrected blendshapes have been used to improve the fit of our model to the registered shape $\hat{\mathbf{S}}^n$, the residual in the reconstruction is attributed only to identity-specific shape variability. We model these residuals as vertex displacements ${\mathbf{V_D}}^n$ and estimate them for each registration $\hat{\mathbf{S}}^n$ by setting:

$$LBS(\bar{\mathbf{V}} + \mathbf{V}_b^n + {\mathbf{V_D}}^n, T(\mathbf{c}_b^n, \boldsymbol{\theta}^n)) = \hat{\mathbf{S}}^n \tag{5.28}$$

to ensure that the residual is defined in the T-pose coordinate system.

For the linear alternative described in Sec. 5.3.2 the shape basis $\mathbf{P}_s$ is computed by performing PCA analysis of $\{{\mathbf{V_D}}^n\}$. To train the graph convolutional system described in Sec. 5.3.2, we learn the parameters of the spiral mesh convolutional decoder $\mathcal{D}$ and the latent vectors $\mathbf{z}^n$ that minimize the following loss:

$$\text{argmin}_{\mathcal{D},\mathbf{z}} \sum_{n=1}^{N} \|{\mathbf{V_D}}^n - \mathcal{D}(\mathbf{z}^n)\| \tag{5.29}$$

5.4.4 Blending Weights

So far the blending weights of our LBS formulation are manually initialised, which can be further improved from the data. For this purpose we use the D-FAUST dataset [11], which contains registrations of a variety of identity and poses. For each registration $\mathbf{S}^n$ in the dataset, we first estimate the parameters of our model, namely $\mathbf{c}_b^n$, $\mathbf{c}_s^n$, $\boldsymbol{\theta}^n$, as well as the residual $\hat{\mathbf{V}}_D^n$ which is the error on the T-pose coordinate system after taking into account the shape blendshapes. Then we optimize instead the blending weights to minimize the following error:

$$\text{argmin}_{\mathbf{W}} \sum_{n=1}^{N} \|\mathbf{S}_n - LBS_{\mathbf{W}}(\bar{\mathbf{V}} + \mathbf{V}_b^n + \mathbf{V}_s^n + \hat{\mathbf{V}}_{\mathbf{D}}^n, T(\mathbf{c}_b^n, \boldsymbol{\theta}^n))\|^2 \tag{5.30}$$

where we use the mapping:

$$\mathbf{W} = \frac{f(\mathbf{W}')}{\sum_j f(\mathbf{W}')_{ij}} \quad \text{with} \quad f(\mathbf{X}) = \sqrt{\mathbf{X}^2 + \varepsilon} \tag{5.31}$$

to optimize freely $\mathbf{W}'$ while ensuring the output weights $\mathbf{W}$ satisfy the LBS blending weights constraints: $\sum_j \mathbf{W}_{ij} = \mathbf{1}$, and $\mathbf{W}_{ij} \geq 0$.

5.5 Evaluation

5.5.1 Implementation Details

Baseline Implementation

The publicly available SMPL model [3] has 10 shape bases, a mesh topology that is different to that of our model, and pose-corrective blendshapes, making any direct comparison to our model inconclusive. In order to have directly comparable results across multiple shape coefficient dimensionalities we train a SMPL-like model (referred to as SMPL-reimpl) using the mesh topology, kinematic structure, and blending weight implementation of our model, and SMPL's PCA-based modeling of shape variability in the T-pose. We further remove any pose-corrective blendshape functionality, allowing us to directly assess the impact of our disentangled, bone-driven modeling of mesh variability against a baseline that does not use it.

In order to train SMPL-reimpl, we first define manually the joint regressor required by [3] by taking the mean of the ring of vertex that lies around a certain joint; we then train the blending weights and joint regressor on the D-FAUST dataset, as described in [3]. The shape blendshapes are then trained with the CAESAR dataset using the same method described in [3].

Mesh Convolutional Networks

For graph convolutional shape modelling, we train networks with 4 convolutional layers, with (48, 32, 32, 16) filters for each layer, respectively. Convolutional layers are followed by batch normalisation and upsampling layers with factors (2, 2, 2, 4) respectively. For the convolutional layers, we use ELU as the activation function. Finally, the output layer is a convolutional layer with filter size 3 and linear activation, which outputs the normalised vertex displacements. We train our network with an Adam optimiser, with a learning rate of 1e-3 and weight decay of 5e-5 for the network parameters, and learning rate of 0.1 and weight decay 1e-7 for the latent vectors. The learning rates are multiplied by a factor of 0.99 after each epoch.

5.5.2 Quantitative Evaluation

We evaluate the representation power of our proposed BLSM model on the CAESAR dataset and compare its generalisation ability against the SMPL-type baseline on D-FAUST dataset and our in-house testset. D-FAUST contains 10 subjects, each doing

14 different motions. We further expand the testset with our in-house dataset. Captured with a custom-built multi-camera active stereo system (3dMD LLC, Atlanta, GA), our in-house testset consists of 4D sequences at 30 FPS of 20 individuals spanning different body types and poses. Each instance contains around 50K vertices. These scans are registered to our template as described in Sec. 5.4.1, while using also temporal consistency constraints.

The models that we compare are aligned to the registered meshes by minimising the L2 distance between each vertex. We use an Adam optimiser with learning rate 0.1 to optimise parameters for all models, and reduce the learning rate by a factor of 0.9 on plateau. To avoid local minima, we use a multi-stage optimization approach as in [66]. We first fit the vertices on the torso (defined by the blending weights of the torso bones on our template) by optimising over the shape coefficients and the joint angles of the torso bones. Then for second and third stage, upper-limbs and lower-limbs are added respectively. In the last stage, all the vertices are used to fine-tune the fitted parameters. In the following, unless specified, we report the mean absolute vertex errors (MABS) of gender neutral models.

CAESAR

In Fig. 5.6 we plot the fitting errors on the CAESAR dataset as a function of the number of shape coefficients, namely shape blendshapes for SMPL-reimpl, bone blendshapes and shape blendshapes for BLSM-linear and latent space dimension for BLSM-spiral.

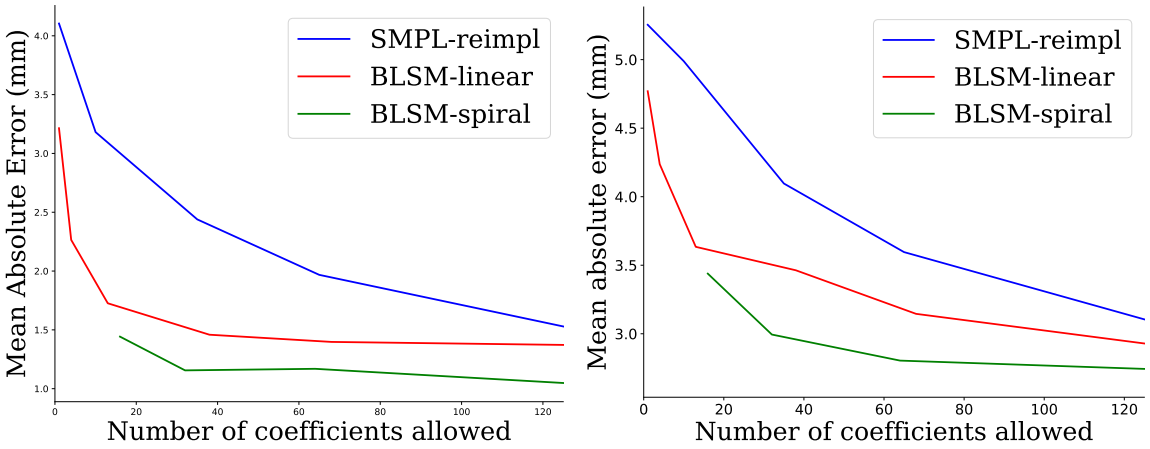


Figure 5.6: Mean absolute vertex error on the CAESAR dataset (left) and our in-house testset (right) against number of shape coefficients.

We observe that our BLSM-linear model attains lower reconstruction error compared to the SMPL-reimpl baseline. The sharpest decrease happens for the first three coefficients, corresponding to bone-level variability modelling. Starting from the fourth coefficient the error decreases more slowly for the linear model, but the BLSM-spiral variant further reduces errors. These results suggest that our BLSM method captures more of the shape variation with fewer coefficients compared to the SMPL-reimpl baseline.

Reconstruction Performance Analysis on CAESAR We also evaluate the reconstruction performance of our proposed linear and graph convolutional model on different demographic groups in the CAESAR dataset. We observe that the bias in our proposed model reflects the biases in the dataset, specifically related to body type variations. For example in the training set, only 0.9% of the subjects has BMI over 40, while our models have significantly higher reconstruction error on this group of subjects. For other attributes (e.g. gender) that are less biased, our model performs similar on all the groups while slightly better on the more common group (53.7% female vs 46.3% male).

Model / Group	BLSM-linear-125 error (mm)	BLSM-spiral-128 error (mm)
BMI ≤ 40	1.224 ± 0.337	1.071 ± 0.426
BMI > 40	2.385 ± 0.429	1.369 ± 0.240
Height ≤ 200 cm	1.221 ± 0.338	1.064 ± 0.419
Height > 200 cm	2.160 ± 0.384	1.642 ± 0.399
Weight ≤ 120 kg	1.217 ± 0.329	1.065 ± 0.422
Weight > 120 kg	2.355 ± 0.377	1.534 ± 0.341
Female	1.283 ± 0.346	0.982 ± 0.385
Male	1.399 ± 0.332	1.204 ± 0.445

Table 5.1: Reconstruction error on groups of subjects with different BMI value, height, weight and gender

D-FAUST

For D-FAUST, we select one male and one female subject (50009 and 50021) for evaluation, and the rest for training blending weights for both models and joint regressors for SMPL-reimpl. We evaluate first shape generalisation error by fitting the models to all sequences of the test subjects (Fig. 5.7 left). We observe that our BLSM-linear

model obtain lower generalisation error compared to SMPL-reimpl baseline, and the result is improved further with BLSM-spiral.

We also evaluate the pose generalisation error of our models (Fig. 5.7 right). The errors are obtained by first fitting the models to one random frame of each subject, then fit the pose parameter to rest of the frames while keeping the shape coefficients fixed. This metric suggests how well a fitted shape generalise to new poses. We observe that both of our linear and spiral models generalises better than our SMPL-reimpl baseline. We argue that by introducing bone scales to the model, the fitted poses are well regularized, thus during training it is more straight forward to decouple the shape and pose variations in the dataset, while avoiding the need to learn subject specific shapes and joints as in SMPL.

In-house Testset

In Fig. 5.6, we also report the average MABS across all sequences in our in-house testset as a function of the number of shape coefficients used. We observe that our proposed models are able to generalise better than the SMPL-reimpl model on our testset. Our proposed models are compact and able to represent variations in our testset with a smaller number of shape coefficients than SMPL-reimpl.

In Fig. 5.8, we show the mean per-vertex error heatmaps on all sequences and on some example registrations in our testset. Compared to SMPL-reimpl, our proposed models are able to fit closely across the full body, while the SMPL-reimpl model produces larger error on some of the vertices. The result suggests that our proposed model

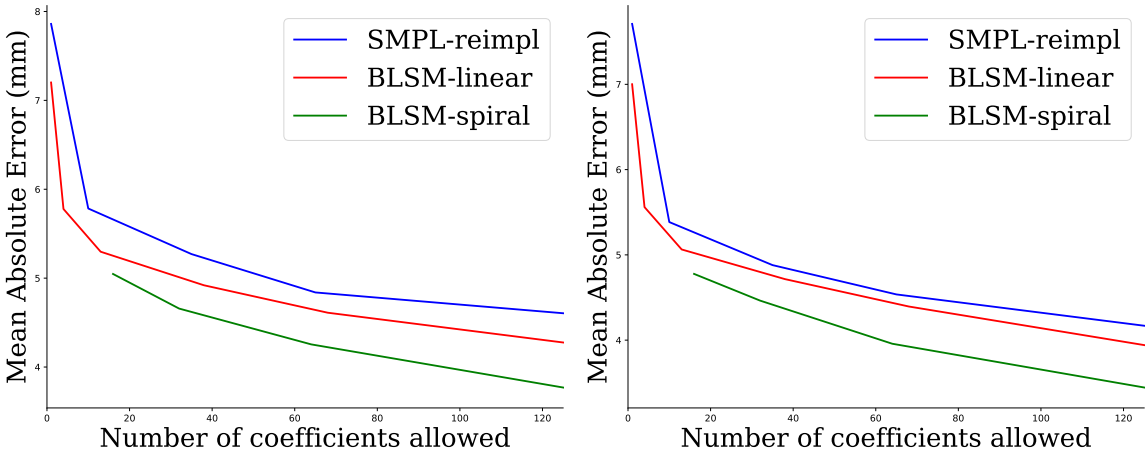


Figure 5.7: Shape generalisation error (left) and pose generalisation error (right) on D-FAUST dataset against number of shape coefficients.

generalise better on surface details than the SMPL-reimpl baseline model.

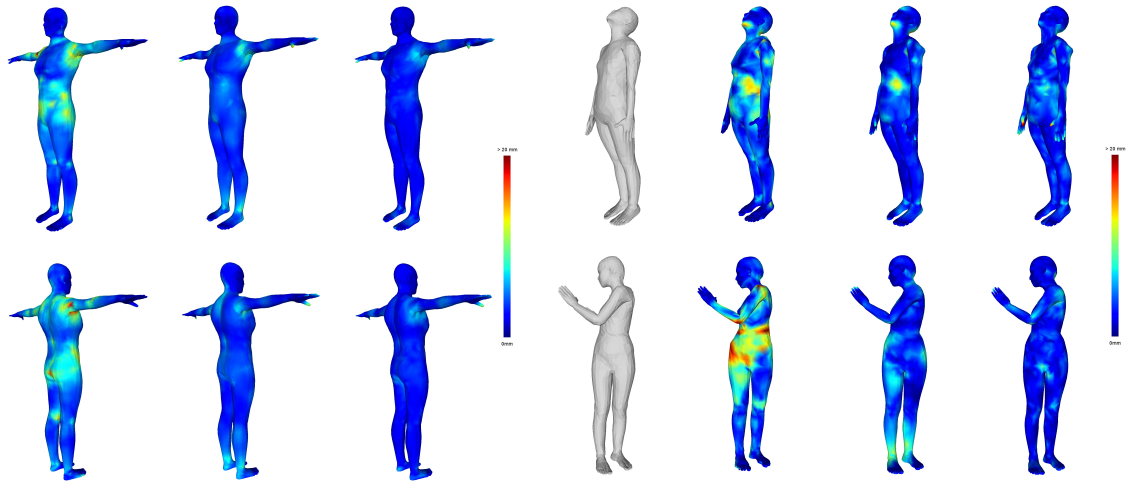


Figure 5.8: Mean absolute vertex error and example of reconstructions on the testset. Left to right: SMPL-reimpl, BLSM-Linear, BLSM-Spiral. For linear models we show result with 125 coefficients allowed. For BLSM-spiral the latent size is 128.

Comparison to original SMPL implementation Here we compare to the publicly available version of SMPL up to 10 shape bases. In order to evaluate the model on our registrations, we transferred the SMPL learnt parameters, namely the template, shape blendshapes, joint regressors and blending weights to our template topology with barycentric correspondences between our template and the SMPL template, while keeping the original SMPL kinematic structures. Poseblend shapes are excluded in this comparison. Table 5.2 shows the errors when 10 and 32 coefficients are used, as well as the area under curve for the cumulative per-vertex error distribution. We observe that SMPL has slightly higher errors than our SMPL-reimpl, we believe that this is due to the topological difference and registration method difference presented in the SMPL training set and our training set. Other than this difference, our SMPL-reimpl performs in par with the publicly available SMPL.

No. Bases / Method	10		32	
	error (mm)	AUC	error (mm)	AUC
SMPL	5.45 ± 3.51	0.695	-	-
SMPL-reimpl	4.98 ± 3.64	0.686	4.09 ± 3.24	0.769
BLSM-linear	$\mathbf{3.63 \pm 2.33}$	**0.794**	3.46 ± 2.86	0.803
BLSM-spiral	-	-	$\mathbf{2.74 \pm 2.03}$	**0.819**

Table 5.2: Generalisation error and AUC for cumulative error distribution on our in-house testset.

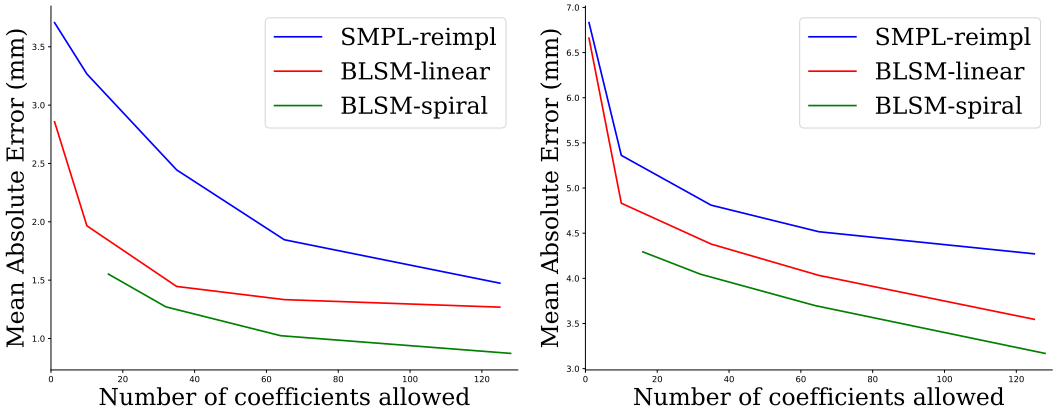


Figure 5.9: Mean absolute vertex error of gender specific models on the CAESAR dataset (left) and DFAUST dataset (right) against number of shape coefficients.

Gender-specific Models

Here we show results of the gender specific models. We plot the mean absolute vertex reconstruction errors on CAESAR and generalisation error on DFAUST. The results shows gender specific models obtain lower error compare to the gender neutral models.

5.5.3 Qualitative Evaluation

In Fig. 5.13 we show samples from our linear model by varying the bone bases, as well as identity-specific shape coefficients from -3σ to +3σ. We observe that our model captures a variety of body shapes and the method successfully decouples bone length-dependent variations and identities specific shape variations.



Figure 5.11: Image-driven character animation: we rig two characters from [9] using our model's bone structure. This allows us to transform any person into these characters, while preserving the pose and body type of the person in the image.

This decoupling allows us to perform simple and accurate character animation driven by persons in unconstrained environments as shown in Fig. 5.11 and Fig. 5.12. In an

offline stage, we rig several characters from [9] to our model's skeleton. In Fig. 5.11, given an image of a person, we first fit our model to it using a method similar to [10]. For Fig. 5.12 we use the ground truth annotations of [118]. We then apply the estimated bone transformations (scales and rotations) to the rigged characters. This allows accurate image-driven character animation within any standard graphics package like Unity. Alternative methods require either solving a deformation transfer problem [2][119], fixed shape assumptions, or approximations to a constant skeleton, while our approach can exactly recover the estimated skeleton position as it is part of the mesh construction.

Please note that many recent works that predict model parameters for image alignment are applicable to our model [10][96][67]; in this work we focus on showing the merit of our model once the alignment is obtained.

We also assess the representational power of our mesh convolutional networks by examining the samples from each dimension of the latent space (Fig. 5.13). We observe that while capturing large deformations such as gender and body type, the network also captures details such as different body fat distributions.

5.6 Conclusion and Future Work

In this chapter we propose BLSM, a bone-level skinned model of the 3D human body mesh where bone modelling and identity-specific variations are decoupled. We introduce a data-driven approach for learning skeleton, skeleton-conditioned shape variations and identity-specific variations. Our formulation facilitates the use of mesh convolutional networks to capture identity specific variations, while explicitly modeling the range of articulated motion through built-in constraints.

We provide quantitative results showing that our model outperforms existing SMPL-like baseline on the 3D reconstruction problem. Qualitatively, we also show that by virtue of being bone-level our formulation allows us to perform accurate character retargeting in-the-wild.

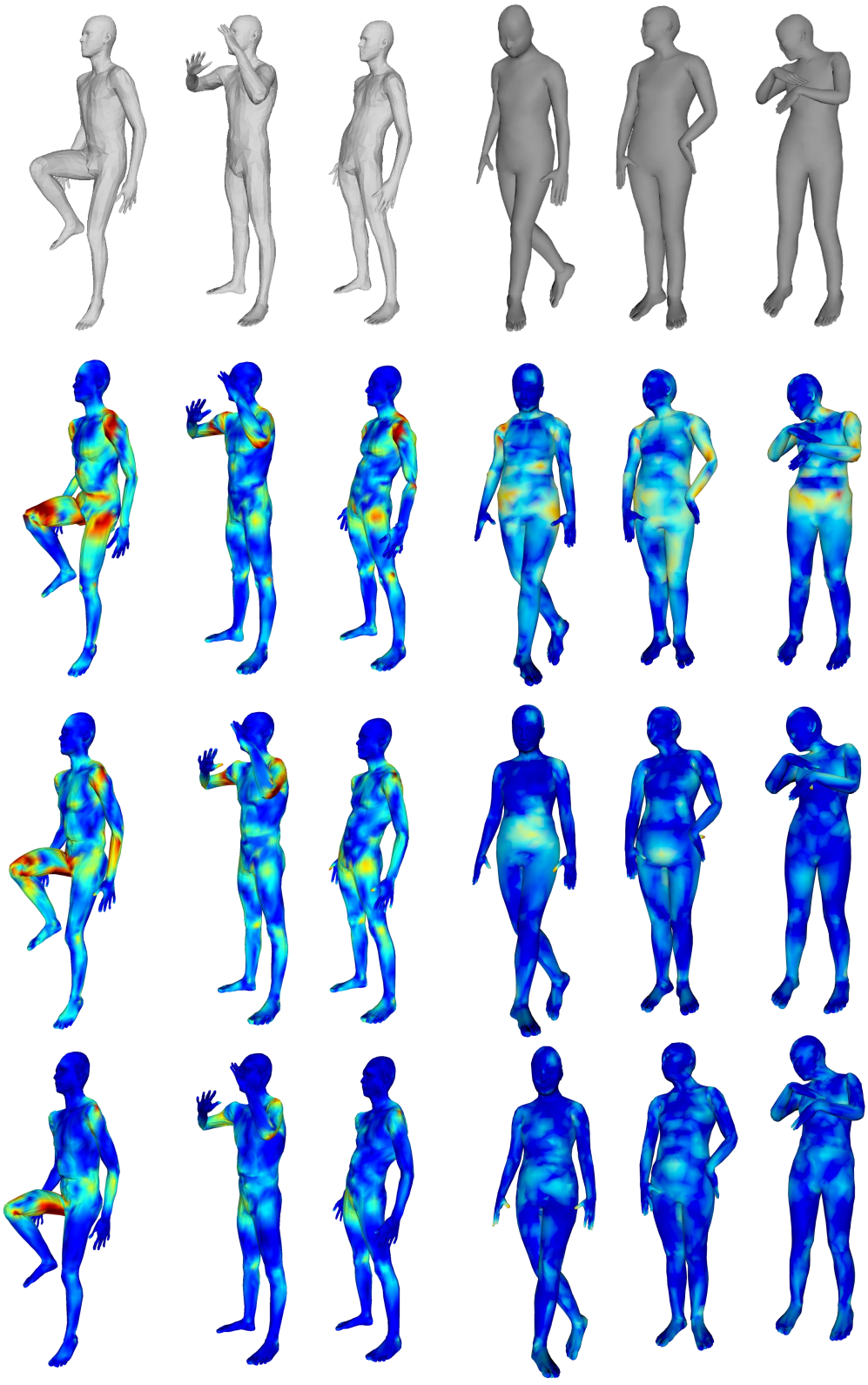


Figure 5.10: Samples from reconstructions of D-Faust and our testset. Top to bottom: ground truth, SMPL-reimpl (125), BLSM-lienar (125), BLSM-spiral (128)



Figure 5.12: Pixel-accurate, image-driven character animation in-the-wild.

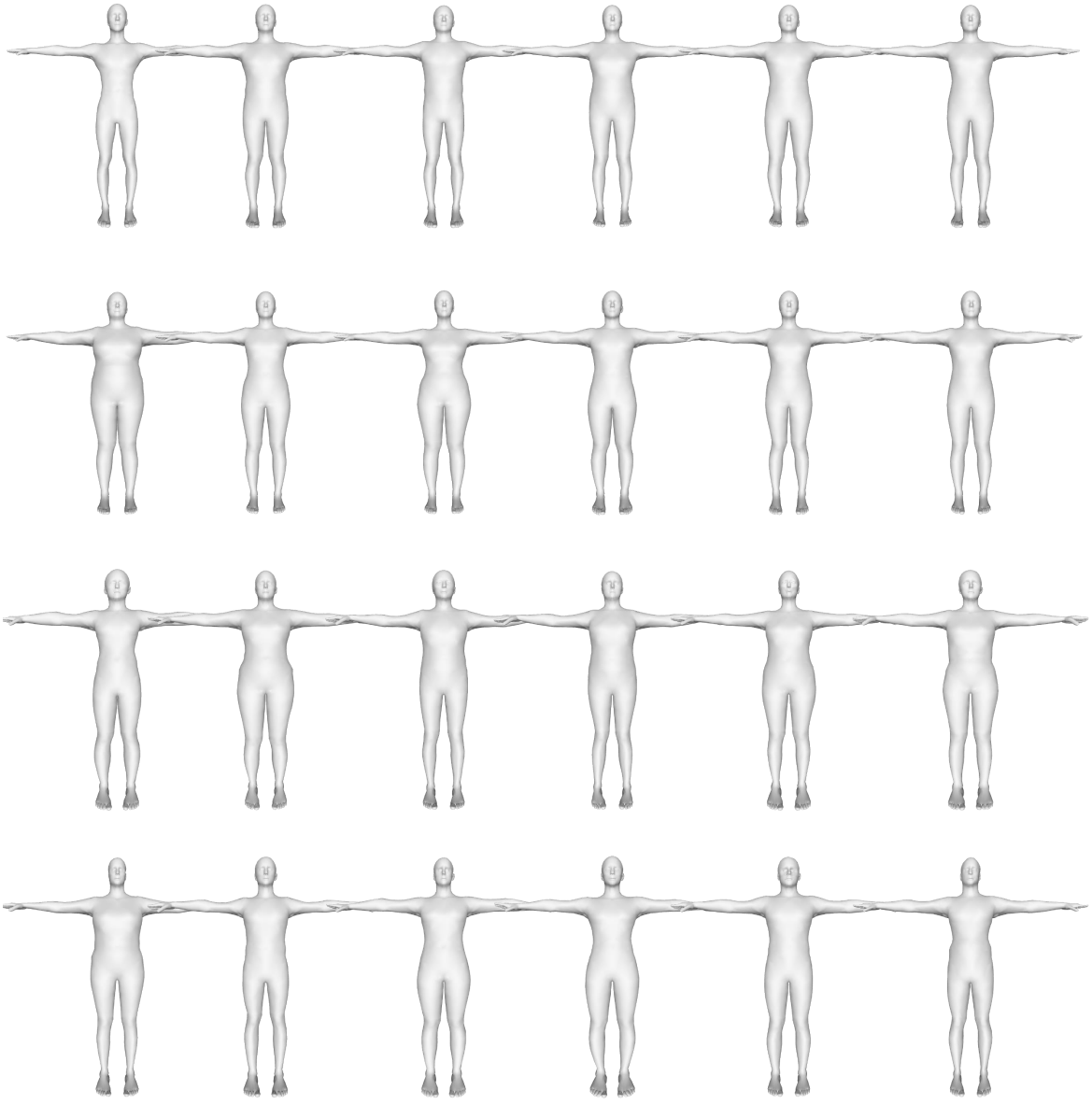


Figure 5.13: Linear (top two rows) vs. graph convolutional (bottom two rows) modeling of shape variation.

Chapter 6

MeDigital: A Large Scale 4D Dataset Of Human Body



Figure 6.1: Sample scans from our large scale 4D scan dataset. The dataset contains over 1.3 million 3D scans of human body with high resolution textures, capturing with over 4200 identities and 7500 dynamic sequences, the dataset surpasses most existing scan datasets in terms of subject and pose varieties, as well as texture quality.

3D human body modelling facilitates a wide range of applications in computer vision and human computer interaction. Current data-driven human body models are restricted to model human body shapes under clothes due to the lack of high quality 3D scan data of clothed human bodies. In this chapter we present the largest to date 4D dataset of clothed human bodies. Our dataset contains over 1.3 million 3D scans of

human body with high resolution textures, surpassing existing scan datasets both in terms of subject and pose varieties, as well as texture quality.

We propose a registration pipeline for registering the large number of high resolution 3D scans. Our registration approach guarantees accurate and realistic registrations even in the case of noisy scans. We evaluate the registration pipeline qualitatively and quantitatively in two folds of our dataset, and perform detailed analysis on the performance in extremely challenging cases. We then demonstrate two use cases of our dataset, namely building parametric models of clothes human body, and monocular 3D reconstruction with synthetic training.

6.1 Introduction

Modelling 3D human body has numerous applications in the context of monocular 3D reconstruction, human body synthesis, motion capture and AR/VR applications such as virtual try-on and character animation. Currently the most widely used human body models capture the variety of human body geometries by learning from 3D scan data [3][30][111]. Building a compact data-driven 3D model requires large number of scan data of varied body type and pose. Among existing large scale datasets, CAESAR [12] and D-FAUST [11] are most commonly used for learning body shape and pose deformations respectively. However in both datasets the subjects are captured in minimum clothing with additional markers or stamps attached to the body, resulting in models of near naked 3D body meshes. While applied to tasks such as monocular 3D reconstruction of in the wild images, the result lacks realism without details such as clothing and hair even after further processing from image cues.

Motivated by the aforementioned problem, we present a large scale 4D dataset of clothed human bodies. The dataset contains 7566 4D sequences of 4205 subjects captured in their daily clothes, resulting in over 1.3 million high resolution textured 3D scans. To the best of our knowledge, this is the largest dataset of 3D human bodies to date, while existing datasets either contains small number of static or dynamic pose variations, small number of subjects, no texture information or low resolution synthetic meshes (Table 6.2).

We design a fast and robust multi-stage scan registration pipeline. We first detect 3D joints of each scan by taking advantage of an accurate 2D keypoint localisation network and perform linear triangulation of the keypoints from rendered 2D image

from multiple views. We then fit a parametric models to a sparse set of points sampled from the scan surface to roughly estimate the subject's body shape and pose. The estimation is optionally refined with a multiview DensePose [95] projection based optimisation pipeline for better body shape fitting. The parametric model estimation is then used as an initialisation to register the set of dense surface points to provide accurate surface alignment.

We carefully regularise the registration process to deal with missing and noisy data in the captured scans. Holes and missing parts in the raw scans are completed in our registration process, resulting in accurate yet realistic meshes. Since the registration process is based on the use of a parametric model, the resulting meshes are parameterised by the model parameters, which also provides markerless motion capture and automatic scan rigging results apart from the dense surface correspondences. Moreover, all the steps of this pipeline are formulated and implemented as batch optimisation problems for each 4D sequence in pytorch, apart from being able to exploit the temporal information in the scan sequences, this implementation also guarantees fast computation while the fitting and alignment process for each 4D sequence can be done with a standard solver in 5 minutes on a single GPU.

We evaluate our proposed registration pipeline quantitatively and qualitatively on two subsets of our dataset, and show that our approach provides high quality results even when the data is extremely noisy. We provide detailed analysis for the registration performance on some extreme cases and demonstrate the merit of our proposed method.

We then perform two more experiments to demonstrate the merit of our dataset. We first build a parametric model from our registered meshes, and show qualitatively that the model captures a wide variety of human body shapes, demonstrating the diversity of our dataset. We also show that our dataset facilitates attribute driven 3D human synthesis. We then propose a 3 stream synthetic image and annotation generation pipeline for monocular motion capture and 3D reconstruction. We train a human mesh recovery system with our parametric model and the synthetic images, and show that training with our synthetic image dataset improves the reconstruction results both qualitatively and quantitatively on in the wild images.

6.2 Dataset Overview

6.2.1 Data Acquisition

The dataset was captured with a custom-built multi-camera active stereo system (3dMD LLC, Atlanta, GA). The system has 14 camera units, each consists of a pair of stereo cameras, a RGB camera and a speckle projector (Fig. 6.2). Speckle patterns are projected to the subject for geometry acquisition with the stereo cameras. Textures are captured with the RGB camera with a lighting system of 12 LED panels which provides uniform lighting. Sequences are captured at 30 frames per-second (FPS), while the system iterates between speckle pattern projection and texture acquisition with a delay of milliseconds. Each reconstructed frame contains a 3D mesh of 50K - 200K vertices depending on the reconstruction quality, together with a high quality texture map of resolution between 6000 to 8000 pixels. Fig. 6.3 shows an example frame of a subject.



Figure 6.2: The data capturing system consists of 14 Modular Camera Units of 42 cameras synchronised with a lighting system of 12 LED panels.

The dataset contains 4205 individuals spanning different ages, body types and ethnic groups. The subjects were captured in their daily clothes. Each subject was asked to provide metadata about themselves, including age, height, weight, and ethnic group. During recording, the subjects were asked to perform a sequence of free motion of their choice for approximately 10 seconds, and a motion from our designed protocols. The protocols cover a wide range of motions for all the human body joints that a person could perform on a daily basis. Table 6.1 lists all the protocols that we have used.

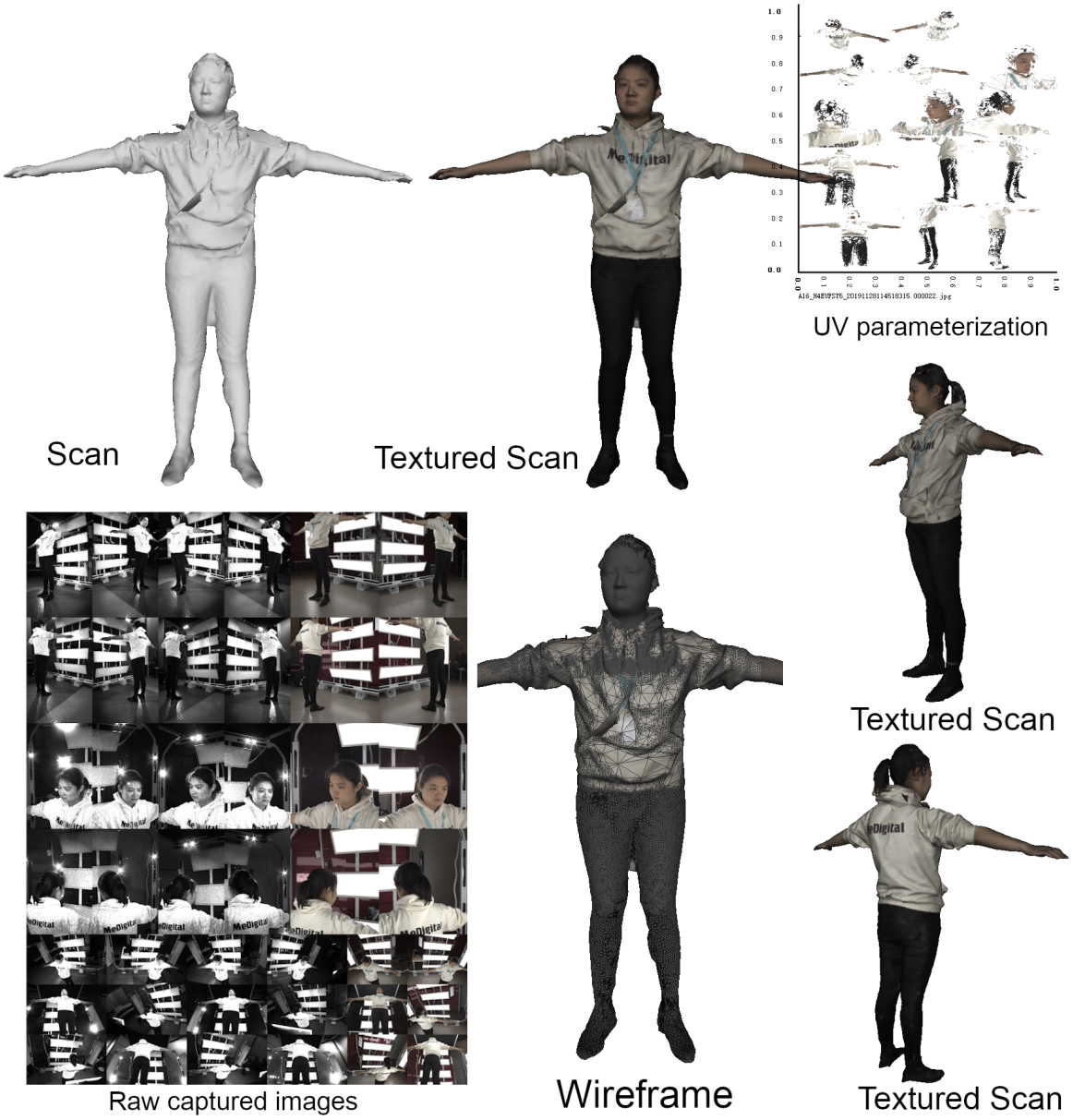


Figure 6.3: An example scan frame from our dataset. At the top row we show the geometry of the scan, the textured scan and the UV parameterization. At the bottom row, we show the collage of raw captured images of this frame, and wireframe of the scan.

7566 sequences were recorded in total with 4205 sequences containing free motions and 3361 sequences containing our protocol motions.

Protocol ID	Description	No. Sequences
01	Neck exercise	96
02	Spine exercise	157
03	Knee exercise	178
04	Squat with arms up	169
05	Macarena dance	238
06	Elbow exercise	201
07	Playing drums	9
08	Scratch back	240
09	Touch ankles	235
10	Pick up object from floor	204
11	Check under feet	196
12	Cross legs (standing)	227
13	Pick up object from shelf, place on the floor	176
14	Cover eyes, mouth, ears	277
15	Sit down and cross legs	265
16	Crunch	393
21	Two person greet	99
22	Two person hug	42
23	Two person interaction	15

Table 6.1: List of motion protocols and number of sequences captured for each protocol. The protocols were designed to capture most of the feasible human joint motions.

	Subjects	Poses / Motions	Frames / duration	Dynamic	Texture	Raw scan available	No. vertices	Real people	Daily clothes	Demographic Info	Note
Ours	4205	7566	1316719 (12h+)	✓	✓	✓	50K - 200K	✓	✓	✓	4205 free motion and 20+ designed pose
CAESAR [12]	4400	3	13200	✗	✓	✓	150K	✓	✗	✓	
Human3.6M [103]	11	17	5h	✓	✗	✗	-	✓	✗	✗	mocap motions, static scan per actor
HUMBI [120]	772	-	230k	✓	✓	✗	4129	✓	✓	✓	
GHUM [49]	48	55	-	✓	✗	-	10,168	✓	✗	✗	
3DPW [118]	7	60	51000	✓	✗	✗	6890	✓	✓	✗	SMPL shape w. mocap motions
AMASS [121]	344	11265	40h	✓	✗	✗	6890	✗	✗	✗	SMPL shape w. mocap motions
3DBodyTex [13]	200	35	400	✗	✓	✓	300K	✓	✗	✗	
SCAPE [30]	1	70	70	✗	✗	✗	12500	✓	✗	✗	
SPRING [122]	3000	1	3000	✗	✗	✗	12500	✓	✗	✗	Registered CAESAR mesh
MPI [31]	114	35	520	✗	✗	✓	180K - 450K	✓	✗	✗	
MPII [33]	4300	1	4300	✗	✗	✗	6449	✓	✗	✗	Registered CAESAR mesh
FAUST [56]	10	30	300	✗	✗	✓	180K	✓	✗	✗	
D-FAUST [11]	10	14	40000	✓	✗	✗	6890	✓	✗	✗	
K3D-hub [123]	50	5	250	✗	✗	✓	150K	✓	✗	✗	
TOSCA [124]	3	20	39	✗	✗	✗	90K (tris)	✗	✗	✗	

Table 6.2: Overview of existing 3D/4D human body datasets

6.2.2 Comparison to Existing Datasets

In Table 6.2 we provide an overview of existing 3D and 4D human body datasets compared to ours. We observe that most of the datasets do not capture a large number of subjects and motions. While for CAESAR [12], over 4000 subjects were captured, however the dataset is not dynamic and contains only 3 different poses. Moreover, the CAESAR subjects were captured with minimum clothing, as such, the geometry and texture information do not represent human bodies in the form they appear in the commonly seen daily scenarios.

We further demonstrate the merit of our dataset by comparing the subjects age, gen-

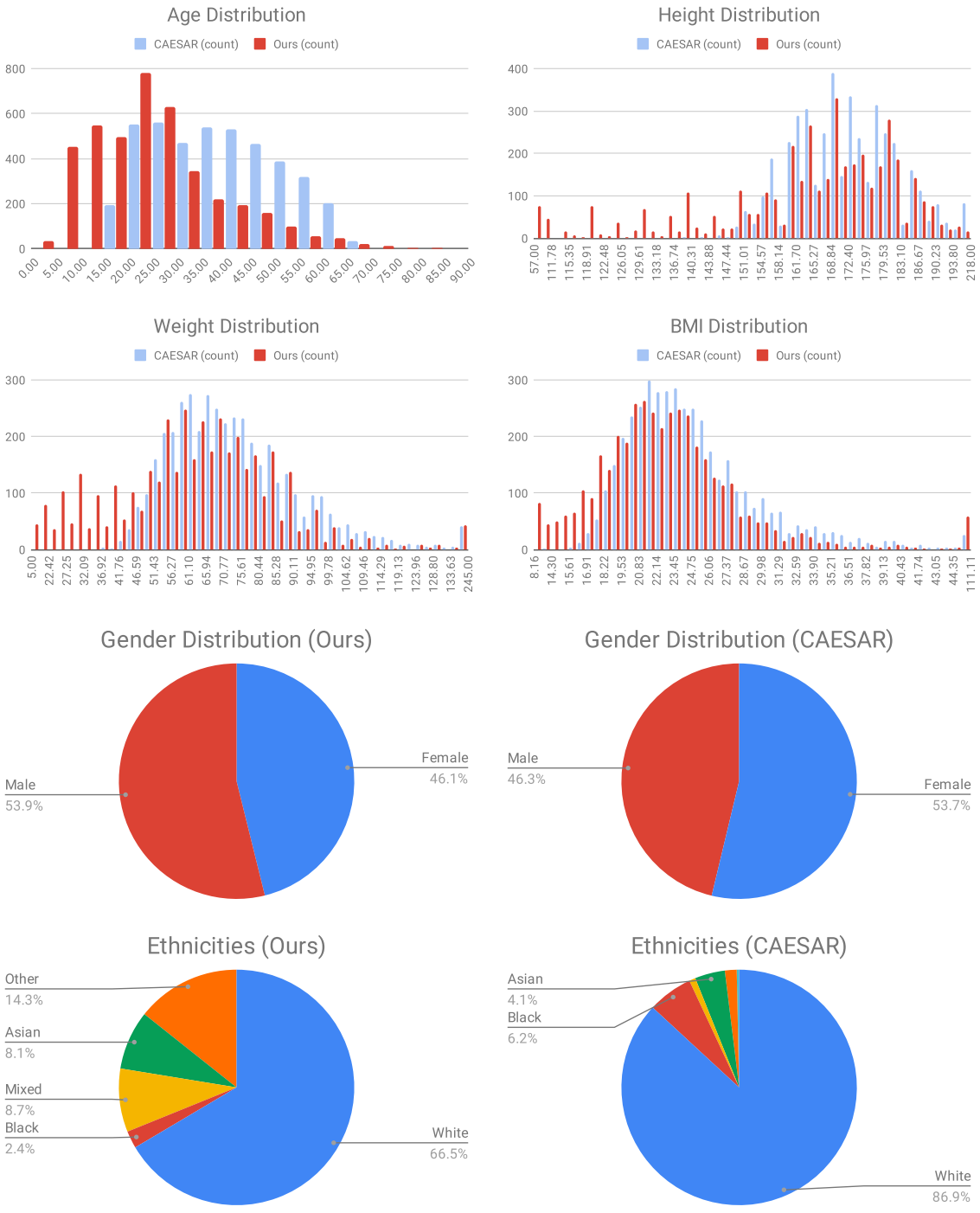


Figure 6.4: Subjects age, gender, height, weight, and ethinic group distribution of our dataset and CAESAR.

der, height, weight, and ethinic group distributions to the CAESAR subjects (Fig. 6.4). We include subjects that are under 18 years old, which are not present in CAESAR. In addition, our dataset covers a wider range of ethnicity, and the distribution is less biased compare to CAESAR.

6.3 Registration

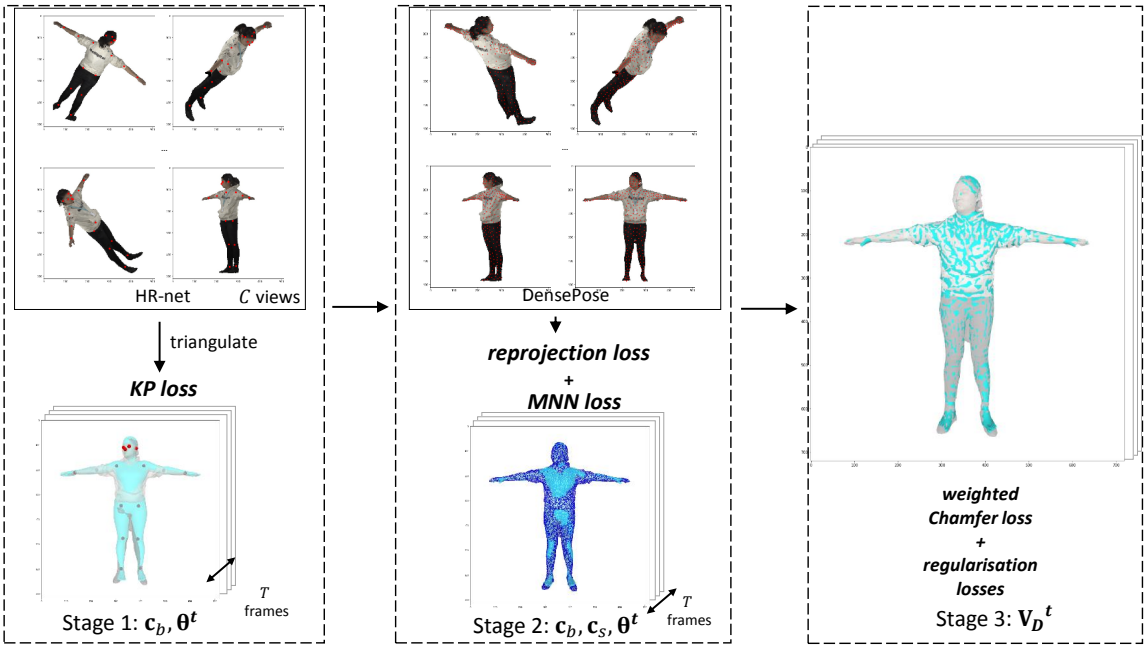


Figure 6.5: Our proposed registration pipeline.

We use a three stage method to register our BLSM model to each sequence in the dataset. At the first stage, we detect 3D keypoints for each frame, and fit the BLSM bone parameters to these keypoints. At the second stage, we include the BLSM shape parameters to fit to pointclouds that are sparsely sampled from the scan surface. The shape is then further refined with multiview image based dense correspondences. And finally, we deform the fitted BLSM model vertices freely to capture the clothes deformations. In the following sections we denote the BLSM model output as:

$$BLSM(\mathbf{c}_b, \mathbf{c}_s; \mathbf{V}_D; \boldsymbol{\theta}) \tag{6.1}$$

which is equivalent to:

$$LBS(\bar{\mathbf{V}} + \mathbf{c}_b\mathbf{P}_{bc} + \mathbf{c}_s\mathbf{P}_s + \mathbf{V_D}, T(\mathbf{c}_b, \boldsymbol{\theta})) \tag{6.2}$$

For simplicity, BLSM parameters that are not optimised in the corresponding registration stages are omitted from Eq. 6.1. We also assume the subject's shape (described by $\mathbf{c}_b$ and $\mathbf{c}_s$) do not change between frames, only pose $\boldsymbol{\theta}$ and vertex displacement $\mathbf{V}_D$ describing the clothes deformations would vary over time.

6.3.1 Stage 1: 3D Keypoints Based Alignment

Detecting 3D keypoints directly from the scan could be computationally expensive given the high resolution nature of the scans and the large amount of data that need to be processed. Moreover, the detected keypoints may not be robust to the noisy scan surfaces — objects other than human body could be present in the scan, such as chairs, markers on the floor, and stairs for young subjects to step on so that their faces are visible in all pairs of cameras — and subsequently affect performance of the whole registration pipeline. Therefore we take advantage of the robustness and compactness of 2D image CNNs.

We render each scan from multiple views where the cameras are sampled uniformly from a sphere. Then we run a 2D human body keypoint detector [125] for each rendered image. With the known camera parameters for each rendered view, we can then perform linear triangulation to find the 3D coordinates for the keypoints. Suppose we have C cameras, where the projection matrix for camera i is $\mathbf{P}_i$, the mesh at frame t is rendered from this camera as $\mathbf{I}_i^t$, and 2D keypoints detected from $\mathbf{I}_i^t$ are denoted as $\hat{\mathbf{L}}_i^t$. The 3D coordinates of keypoints at frame t $\mathbf{L}_t$ can be computed by solving the following linear equation system:

$$\hat{\mathbf{L}}_i^t = \mathbf{P}_i \mathbf{L}_t, \quad \forall i \in \{1, ..., C\} \tag{6.3}$$

After each estimate, for each keypoint we remove 25% of the views with the largest reprojection error, and re-estimate $\mathbf{L}_t$. This is repeated for 3 iterations before we obtain the final estimate of $\mathbf{L}_t$. The confidence score $\mathbf{cf}_i^t$ of the survived keypoints are then averaged to obtain confidence $\mathbf{cf}^t$ for frame t.

Once keypoints for all frames are computed, the BLSM bone parameters $\mathbf{c}_b$ for the subject in this sequence and $\boldsymbol{\theta}^t$ at each frame can be estimated as:

$$\boldsymbol{\theta}^t, \mathbf{c}_b = \text{argmin}_{\boldsymbol{\theta}, \mathbf{c}_b^*} \sum_{t=1}^{T} (\mathbf{cf}^t \| \mathbf{A}_{kp} BLSM(\mathbf{c}_b^*; \boldsymbol{\theta}) - \mathbf{L}_t \|^2 + S(\boldsymbol{\theta})) \tag{6.4}$$

where $\mathbf{A}_{kp}$ is a linear matrix that regresses keypoints from a BLSM mesh, and S is the temporal smoothing function that encourages smooth and slow changes of pose between frames.

6.3.2 Stage 2: Shape Initialisation with Sparse Correspondences

The keypoints based stage provides a good initialisation for bone dependent shape fittings. We then optimise over non-bone dependent shape parameters. For each frame we sparsely sample 10K points $\mathbf{S}_t^s$ evenly from the scan vertices, then the BLSM shape parameter $\mathbf{c}_s$ of this subject is estimated by minimising the following loss:

$$\sum_{t=1}^{T}(\lambda_{kp}\mathbf{cf}^t\|\mathbf{A}_{kp}BLSM(\mathbf{c}_b,\mathbf{c}_s;\boldsymbol{\theta}^t)-\mathbf{L}_t\|^2+\sum_{(\mathbf{v}_r,\mathbf{v}_s)\in MNN}\|\mathbf{v}_r-\mathbf{v}_s\|^2+S(\boldsymbol{\theta}^t)) \quad (6.5)$$

with λ_{kp} is the keypoint term weight and

$$MNN = M(BLSM(\mathbf{c}_b,\mathbf{c}_s;\boldsymbol{\theta}^t),\mathbf{S}_t^s) \quad (6.6)$$

where function M is the mutual nearest neighbour operation defined in Eq. 5.16. In practice we initialise $\mathbf{c}_b$ and $\boldsymbol{\theta}^t$ with results from the previous stage, and jointly optimise them with $\mathbf{c}_s$. We use $\lambda_{kp} = 0.1$ to regularise the shape fitting with the keypoints while emphasising more on the surface loss.

6.3.3 Shape Refinement Based on Multiview Dense Correspondences

The previous two stages guarantee fast and robust fittings that can act as a prior for the following registration steps. However the fitting may not be accurate enough due to noise in the scans. In particular, some body parts, such as bottom of the feet, that are not visible in most of the camera views are sparsely represented in the reconstructed 3D scans due to lack of correspondence evidence in the raw captured images. As such, these vertices contribute less to the loss in Eq. 6.5, and the optimisation process will be purely driven by the estimated keypoints which is also prune to errors. Moreover, in some cases, with the mutual nearest neighbour approach, wrong correspondences could be established due to bad initialisation from stage 1 (Fig. 6.6).

We observe that even though the vertices might be missing for certain body parts, the

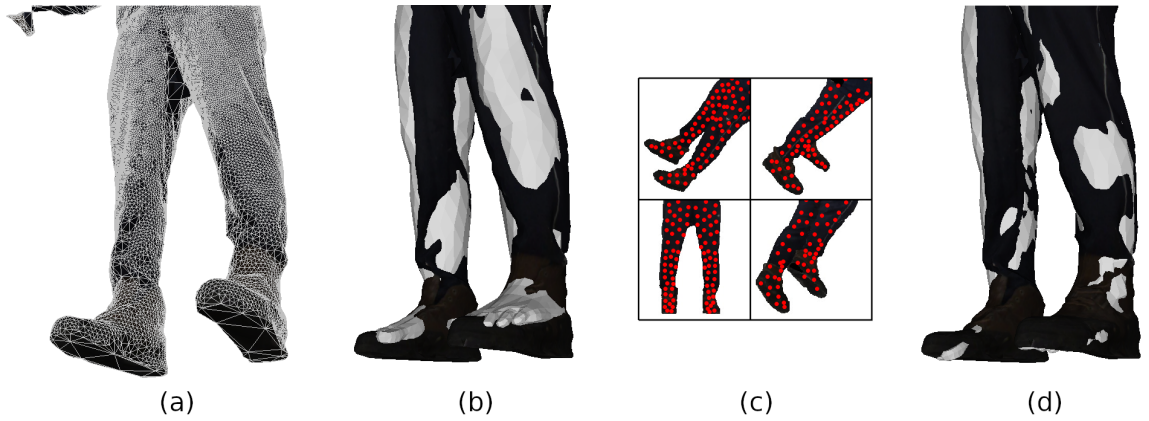


Figure 6.6: (a) Example of a scan instance where vertices at bottom of the feet are missing. (b) BLSM fitting result at stage 2. Wrong correspondences are established due to bad initialisation from stage 1. (c) DensePose result from multiple views. (d) BLSM fitting result with DensePose reprojection loss.

textures are still present in the sparsely connected faces, and could be used as evidence to refine our shape estimations. Therefore we propose to obtain dense correspondences from the previously rendered multiview images. For this purpose, we run DensePose [95] to estimate 2D coordinates of the BLSM vertices $DP(\mathbf{I}_i^t)$ in each rendered image $\mathbf{I}_i^t$, then add the following reprojection loss to Eq. 6.5:

$$\sum_{t=1}^{T}\sum_{i=1}^{C} \|\mathbf{P_i} BLSM(\mathbf{c}_b, \mathbf{c}_s; \boldsymbol{\theta}^t) - DP(\mathbf{I}_i^t)\|^2 \tag{6.7}$$

6.3.4 Stage 3: Refining Surface Details

Finally, we optimise over vertex displacements $\mathbf{V}_D^t$ to capture details such as clothes deformations and shape variability from subject groups that are not present in the BLSM model built from the CAESAR dataset. The loss function used here is similar to what we have used for registering the CAESAR scans in Sec. 5.4.1:

$$E = \sum_{t=1}^{T}(E_{data} + E_{smoothness} + E_{reg} + E_{rigid}) \tag{6.8}$$

We use the same smoothness, regularisation and rigidity term for each frame as in Sec. 5.4.1, while using instead the result from stage 3 for initialisation and replacing



Figure 6.7: Registration result of one example frame. (a) Triangles with area larger than a certain threshold is omitted to deal with missing parts and noise in the scans. (b) Registration result with pointcloud evenly sampled from the scan surface. (c) Registration result with pointcloud sampled from triangles smaller than $5 \times 10^{-4}\ m^3$

the data term with the following loss:

$$E_{data} = \lambda_{scan} \sum_{\mathbf{v}_r \in \mathbf{S}^{ds}_{BLSM_t}} \sum_{\mathbf{v}_s \in NN(\mathbf{v_r}, \mathbf{S^{ds}_t})} \|\mathbf{v}_r - \mathbf{v}_s\|^2 + \lambda_{model} \sum_{\mathbf{v}_s \in \mathbf{S}^{ds}_t} \sum_{\mathbf{v}_r \in NN(\mathbf{v_s}, \mathbf{S^{ds}_{BLSM_t}})} \|\mathbf{v}_r - \mathbf{v}_s\|^2 \tag{6.9}$$

where $\mathbf{S}_{BLSM_t} = BLSM(\mathbf{c}_b, \mathbf{c}_s; \boldsymbol{\theta}^t; \mathbf{V}^t_D)$ and the superscription **ds** represents densely sampled surface points. Here we use weighted chamfer loss instead of the mutual nearest neighbour loss so as to capture the surface details resulted from clothes deformations. At each frame, both the BLSM model surface and scan surface are densely sampled, resulting in 20K and 50K points respectively. In order to be robust to missing parts, for the scan surface, we only sample from triangles whose area is smaller than a certain threshold (Fig. 6.7).

6.4 Evaluation

6.4.1 Registration Quality

In this section we evaluate the quality of the registration results provided by our proposed pipeline. Since there is no ground truth registration available for our dataset, we evaluate in terms of model to scan distance. More specifically, for each vertex in our registered mesh, we compute its distance to the closest vertex in the raw scan. For this experiment, we split the dataset into 2 groups. One group is the **multi-subject** set that contains the first frame from the free motion sequence of all subjects. This

is to evaluate how well our registration pipeline captures the variety in the subjects' body types. Another group is the **multi-pose** set, for which we evaluate a subset of all the recorded sequences downsampled to 10 fps. In the rest of this experiment, we report mean per-vertex euclidean errors.

Protocols	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**Multi-subject**
Mean vert. err. (mm)	8.46	15.81	9.41	8.91	10.50	12.91	9.95	12.19	13.13	10.74	9.65
Protocols	**Free**	**11**	**12**	**13**	**14**	**15**	**16**	**21***	**22***	**23***	**Multi-pose**
Mean vert. err (mm).	13.62	9.15	16.24	13.78	10.30	21.13	18.03	42.02	31.69	45.14	14.17

Table 6.3: Mean vertex to scan registration error in (mm) on the multi-subject and multi-pose subset of the dataset. Protocols (01-23) are as listed in Table 6.1. Here (*) indicates that sequences of this protocol contains two persons.

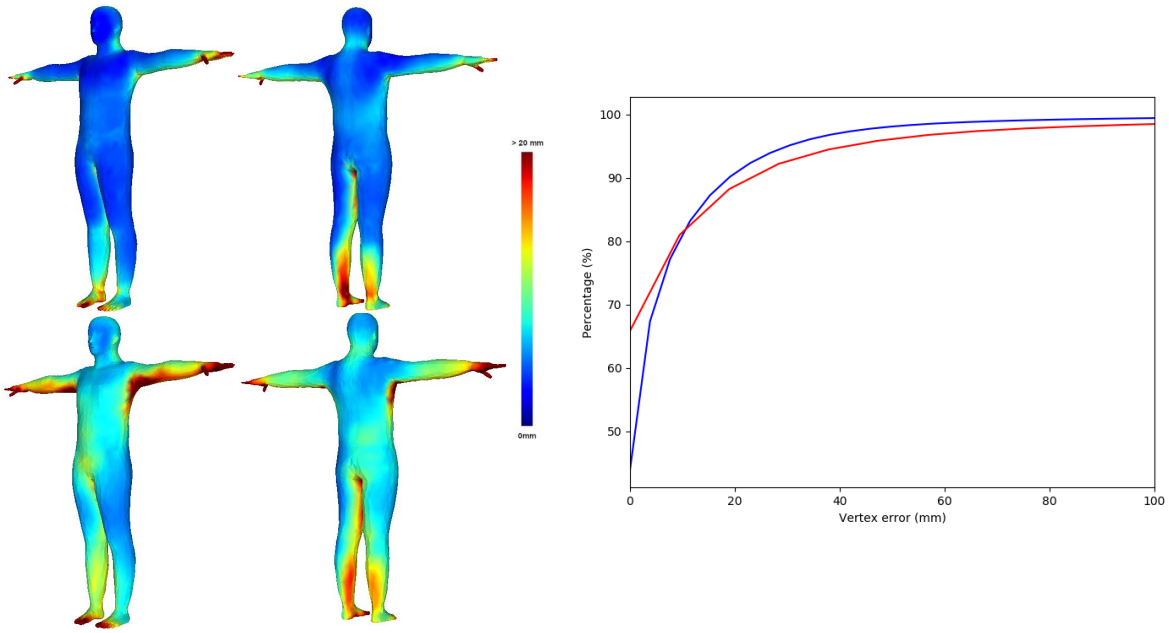


Figure 6.8: Left: Error heatmaps on the multi-subject (top row) and multi-pose (bottom row) subset visualised on the mean BLSM template. Colors are shown in millimetres. Right: cumulative per-vertex error plot. Blue line: multi-subject subset, red line: multi-pose subset.

In Table 6.3 we report the mean vertex to scan registration error on the multi-subject set and multi-pose set, as well as detailed result for each motion protocols. The per-vertex errors are visualised as error heatmaps in Fig. 6.8, as well as the cumulative error for the multi-subject set (blue line) and multi-pose set (red line). For the multi-subject set, the mean model vertex to scan vertex error is 9.65 mm. For this set, largest errors occur on the fingers and feet, as well as between the legs. These regions typically have sparse points in the raw scan, and since we compute the errors from

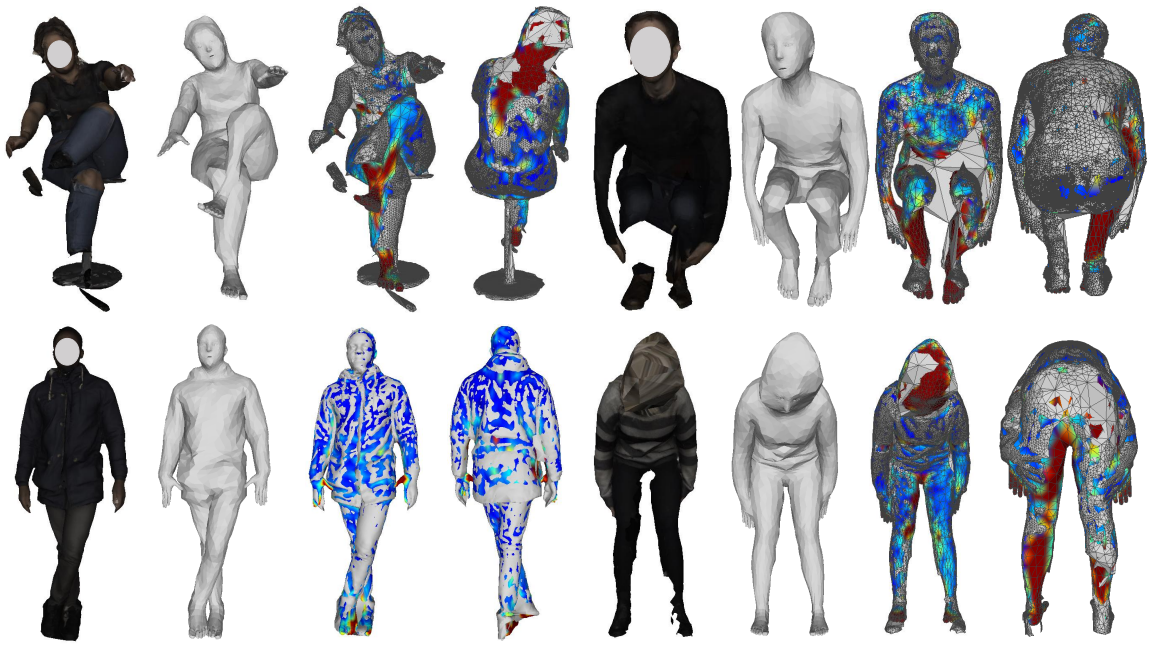


Figure 6.9: Visualisation of some example frames from the multi-pose subset where the registration error is large. Samples are from protocol 15, 16, 12, 02 respectively. For each frame we show: raw scan with texture (with face of the subject occluded), registration result, error heatmap visualised on the registration overlaid with the raw scan from front and back view.

model vertex to scan points instead of the scan surface, it is expected that the errors will be large for these parts. Same reason applies for the large error on the under arm regions in the multi-pose set.

For the multi-pose set, we obtain the error of 14.17 mm. Largest errors occur for protocols 21, 22 and 23, which contains scene of more than one person. Since the keypoint detector we used in stage 1 are trained for single person, wrong keypoints are detected for the multiperson cases, resulting in bad initialisation for sequences of these protocol. Moreover, the data capturing system are designed for capturing one person, while multiple persons are present in the scene, mutual occlusions between the subjects results in large missing parts and noise in the raw scan, which also cause large registration errors.

For protocols with one person, the largest errors are from sequences of Protocol 15 (21.13 mm), 16 (18.03 mm), 12 (16.24 mm) and 02 (15.81 mm). These protocols correspond to the motion of sit down and cross legs, crunch, cross legs (standing), and spine exercise. In Fig. 6.9 we visualise example frames from these protocols where the registration error is large. Large motions from these protocols cause self occlusions

of the subjects, result in poor reconstruction quality in the raw scan of the occluded parts. In extreme cases, e.g. the first example shown in Fig. 6.9, half of the limb is missing and our keypoint detector from registration stage 1 provides false detections, consequently harms the registration results in the upcoming stages. Self occlusion of the subjects also causes sparse reconstruction in some parts (back of neck in first example, torso of second example, head and legs of fourth example in Fig. 6.9), however our pipeline provides plausible and more realistic registrations despite the large model to scan points error that we evaluate and visualise.

6.4.2 Attribute-driven 3D Mesh Synthesis

When modelling meshes of clothed human bodies, decoupling shape and bone modelling is particularly useful. Prior work either model the garments as a standalone mesh [23], or as vertex offsets on top of the shape model [24][22]. This requires estimating the body shape under the clothes, which is a difficult problem itself. The BLSM formulation we proposed in the previous chapter is more suitable for modelling clothed human bodies, since the bones are modelled separately and are not influenced by the clothes deformations.

We build BLSM on the resulting registrations following the pipeline described previously. For each subject, we take the first frame from one of the recorded sequences, which is the T-pose frame of the subject as instructed during the recording. In figure Fig. 6.10 we visualise the bone and shape bases varying from -3σ to $+3\sigma$.

We could attempt to interpret the semantic meaning of each BLSM bases from this visualisation. For example, when sampling along the first bone bases from $+3\sigma$ to -3σ, the height of the person increases, and we can see that the corresponding bone corrective blendshape varies from children to adult, then female to male. However there is no true semantic meanings associated to any of the BLSM parameters. In this experiment, we attempt to close the gap between semantically meaningful parameters, which in our case, are the demographical information we collected from the subjects (age, height, weight and gender), and the BLSM parameters.

More specifically, our goal is to build a mapping to obtain BLSM parameters that could result in a shape which could be interpreted by the input attribute values. In the experiment, we evaluate several different approaches and show qualitative results. Note that there are prior works [31][122] that attempts to edit meshes based on detailed

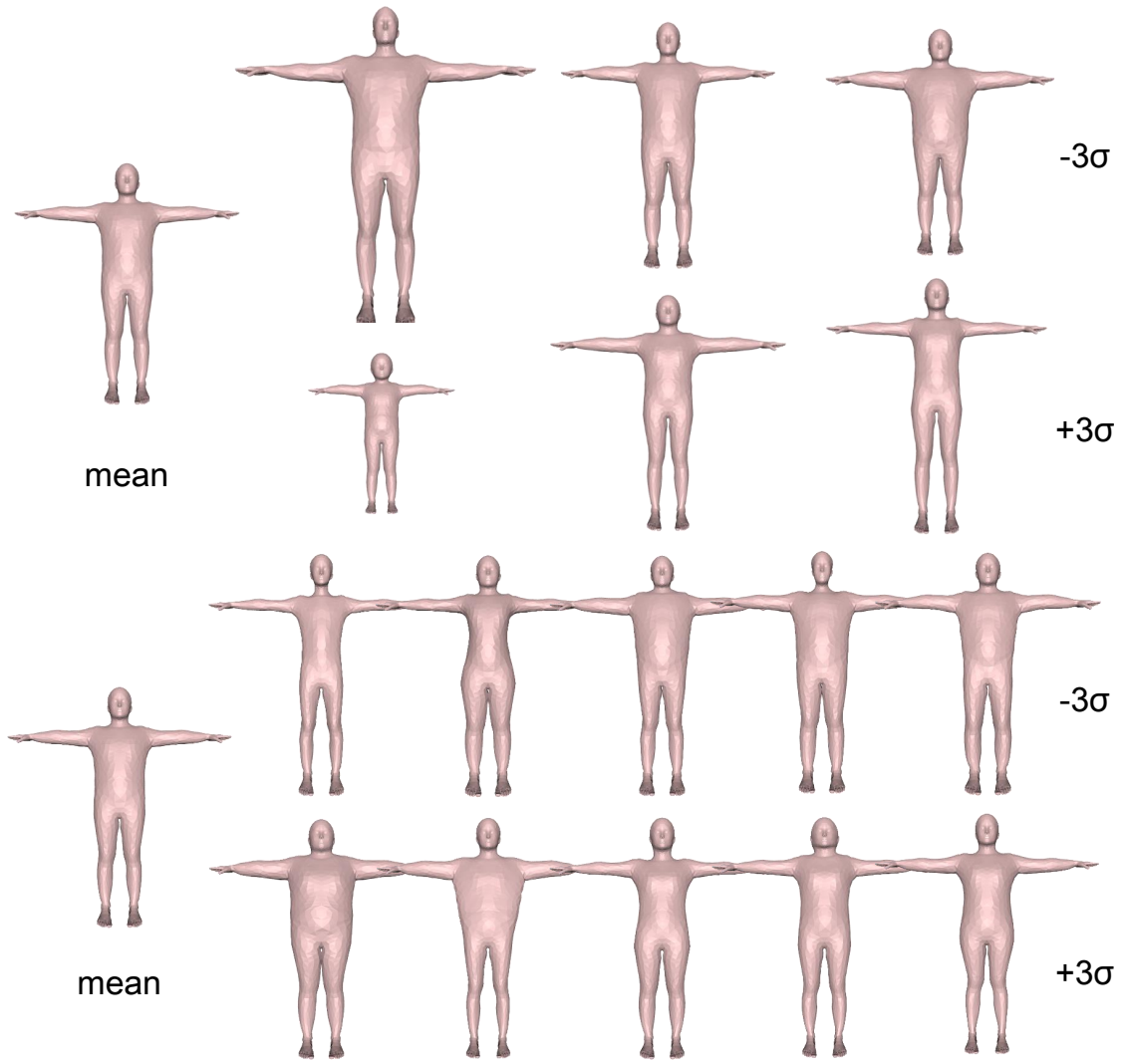


Figure 6.10: First 3 bone bases and first 5 shape bases from -3σ to $+3\sigma$

semantic parameters (e.g. limb lengths, circumference of each body parts), here we focus instead on the more generic and interpretable semantic parameters, and leave the detailed semantic mesh editing for future work.

Regressing BLSM parameters from attribute values

The most straightforward method to obtain a mapping from attribute values to BLSM parameters is regression as proposed in [41]. We first obtain the BLSM shape parameter vector we intend to regress $[\mathbf{c}_b^n, \mathbf{c}_s^n]$ and the pose parameter $\boldsymbol{\theta}^n$ for all of the registered meshes $\mathbf{S}^n$ by minimising the following loss:

$$\|BLSM(\mathbf{c}_b^n, \mathbf{c}_s^n; \boldsymbol{\theta}^n) - \mathbf{S}^n\|^2 \tag{6.10}$$

Then the mapping is obtained by training a 3-layer fully connected network N_r with 512 unit for each layer. Denote the normalised attribute values as $\mathbf{x}^n \in \mathbb{R}^4$, the network is trained by minimising the following loss:

$$\sum_{n=1}^{N} \|N_r(\mathbf{x}^n) - [\mathbf{c}_b^n, \mathbf{c}_s^n]\|^2 + \lambda_{decay}\|N_r\|^2 \tag{6.11}$$

where λ_{decay} is the weight for the L2-norm term.

In Fig. 6.14 we visualise meshes generated from the attribute values as indicated for each example in the figure. We use [5, 15, 30, 50, 80] for age, [80, 130, 160, 180, 200] (cm) for height, [20, 50, 80, 100, 120] (kg) for weight and [female, male] for gender. We observe that the network do not generalise well to values that are not presented in the dataset (for example, 5 years old kid that is 165 cm tall), and the network has overfit to the correlations in the input attribute values.

Mapping PCA subspaces

Inspired by [31], we have attempted another approach which is to map the PCA subspace of the attribute values and the BLSM shape parameters. More specifically, we perform PCA analysis on the input attribute values, and regress the projected PCA coefficents to obtain the BLSM shape parameters. We use the same network architecture as before, and the results are shown in Fig. 6.15. Compared to the direct regression approach, we observe that the results are improved. However invalid body shapes can still occur when the input value is extreme.

Generating BLSM parameters with a conditional GAN

Having observed the artefacts that could occur when using the previously described methods, we now turn to a method that can generate valid shapes while being able to model the mapping between the attribute values and the BLSM shape parameters. For this purpose we propose to use a conditional GAN with reconstruction losses. Given a vector of attribute values $\mathbf{x}$ and the target BLSM parameters $\mathbf{c} = [\mathbf{c}_b, \mathbf{c}_s]$, a discriminator network D and a generator network G are trained by minimising the following loss:

$$E_D = \sum_{n=1}^{N}((D(\mathbf{c}^n, \mathbf{x}^n) - 1) + D(G(\mathbf{z}, \mathbf{x}^n), \mathbf{x}^n)) \tag{6.12}$$

$$E_G = \sum_{n=1}^{N}((D(G(\mathbf{z}, \mathbf{x}^n), \mathbf{x}^n) - 1) + \|G(\mathbf{z}, \mathbf{x}^n) - \mathbf{c}^n\|^2) \tag{6.13}$$

where $\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$ is random noise sampled from the noise distribution.

In Fig. 6.16 we show random samples from our trained generator conditioned on the attribute values as indicated in the figure. We observe that all the generated shapes are valid shapes for human bodies. In addition, the shapes exhibits some properties of the input attribute values. In the case where the combination of input values are extreme (e.g. the first instance in the first row), the network is able to generate a valid interpolate between the shapes that exhibits both of the properties.

6.4.3 Single Image Mesh Recovery with Synthetic Training

In this experiment we demonstrate an application of our dataset for the task of human mesh recovery from single image. One challenge in learning based single image mesh recovery is the lack of ground truth 3D annotation, which can be tackled by using synthetic images. Prior works such as [126] uses MoCap data to generate meshes with a parametric model. In this case, the synthesised images have limited realism since the parametric model only captures surface variations of minimum clothed human bodies. Moreover, meshes synthesised with parametric models have low resolution, which cannot be used for training systems that outputs 3D reconstructions with fine details.

Synthetic Image Generation Our dataset can be used to improve the quality of synthetic images. In Fig. 6.11 we show 3 different synthetic image generation pipelines that can be applied given our dataset. One possibility is to directly render the raw 3D scans into background images (Fig. 6.11 black). The raw 3D scans provide high resolution 3D ground truth meshes, which can be useful for systems such as [29] that requires ground truth surface with fine details, and does not require the vertices to be in one-to-one correspondence.

Similarly, the raw 3D scan set can be expanded by modifying the pose of each scans, resulting in exponential growth of the dataset size (Fig. 6.11 blue). Our registration pipeline can act as an automatic rigging system, where the blending weights of each vertex in the raw scan is painted by finding its closest vertex in the BLSM registration. We then solve for the high-resolution subject specific T-pose template by fixing the pose parameter and optimising over the template's vertex locations. The high-

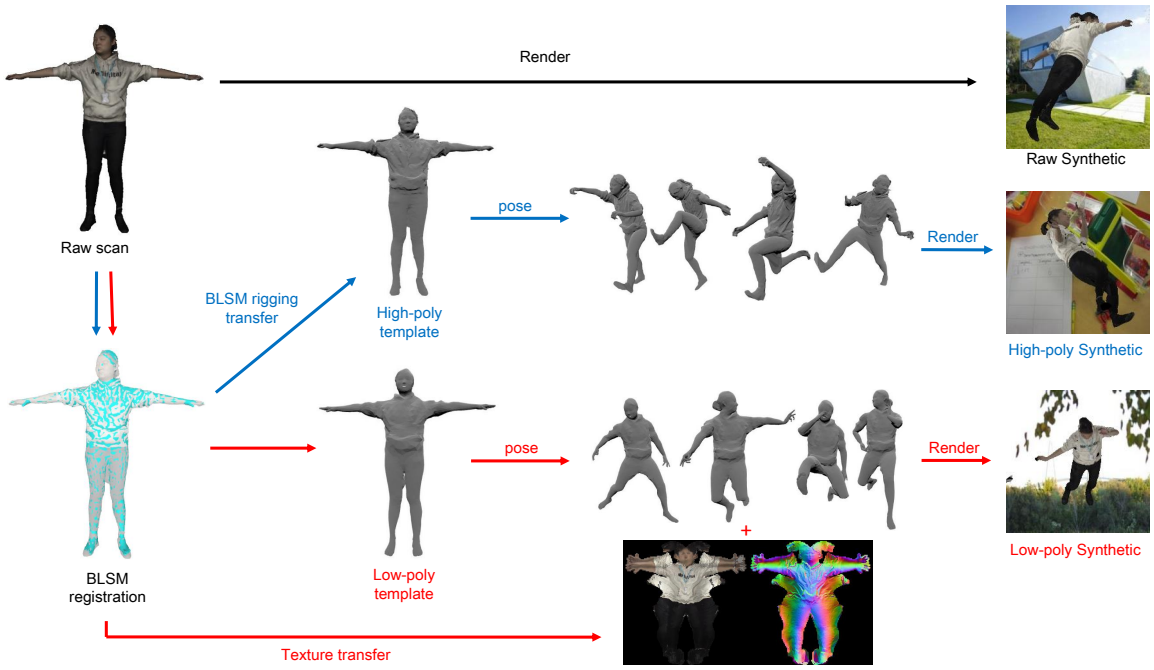


Figure 6.11: Variations of synthetic image generation pipelines using our scan dataset.

poly subject specific BLSM model can then be posed with MoCap data from other sequences in our dataset or from other datasets.

The major artefact in the high-poly synthetic images is the missing parts in the raw scans. In order to obtain noise free synthetic images, the registered BLSM mesh can be used (Fig. 6.11 red). Since our registration pipeline was implemented in the BLSM model space, the pose of the registered mesh can be changed simply by modifying the BLSM pose parameter. Texture and normal maps from the raw scans are transferred to the BLSM UV space using correspondences obtained from the registered mesh to improve rendering quality in the synthetic image. While providing noise free synthetic images, this approach also provides one to one correspondence of the image pixels to the BLSM template mesh, facilitating parametric model based mesh recovery methods. Resolution of the 3D annotations can be improved by subdividing the BLSM model surface.

Experimental Setup For this experiment we generate synthetic images using the raw scans only. Fig. 6.12 shows samples from the synthetic image dataset we used for training the human mesh recovery system. We reuse the rendered raw scans obtained in Section 6.3.1 and place the rendered subjects onto background images from the

Place365 dataset [127]. Each sequence is downsampled to 5 fps to reduce redundancy. We also exclude subjects whose raw scan reconstruction quality is poor (e.g. whole limb missing).



Figure 6.12: Samples from our synthetic image dataset.

The mesh recovery architecture we used in this experiment is HMR [10]. Since the synthetic images are annotated in the BLSM model space, we replaced the SMPL mesh generation head of HMR with BLSM. Each pose atom of BLSM is by design in a kinematically plausible range, therefore the pose embedding and individual joint rotation discriminator network are not needed in our experiment. More specifically, our pose discriminator only consists of one network with 2 layers of fully connected

	UP-3D	LSP			
		Fg vs Bg		Parts	
Training Data	Joint Reconst. Err. (mm)	Acc (%)	F1	Acc (%)	F1
Real only	72.44	85.65	0.79	82.86	0.49
Real + Synthetic	68.67	86.17	0.79	83.95	0.52

Table 6.4: Joint reconstruction results on UP-3D dataset and segmentation results on LSP dataset. Segmentation results are evaluated for both foreground vs background segmentation as well as part segmentations (6 parts + background). Network trained on synthetic + real image outperforms the network trained only with real images on all tasks.

layers of 1024 neurons. The network takes the 47D BLSM pose parameter as input, and outputs 1 binary value.

Results We train the network on the synthetic images with 3D joints, 2D joints, pose and shape parameter supervision. The network is then fine-tuned on UP-3D dataset to incorporate real in the wild images. In the following experiments, we compare the performance of this network (**real + synthetic**) to the same network trained only on the UP-3D dataset (**real only**).

We evaluate each network on the UP-3D testset and LSP dataset [86]. For UP-3D testset, we compare the 3D joint reconstruction error in millimetres. For LSP dataset, we evaluate foreground vs background segmentation results as well as 6 body parts segmentations. The results are shown in Table 6.4. The network trained with real + synthetic images outperforms network trained only with real images on all the tasks. This is expected since the UP-3D training set only contains 8K images in total, which is not sufficient for the challenging task of mesh recovery from raw RGB images. Our synthetic dataset expanded the training set by over 130K images, thus guarantees better performance.

In Fig. 6.13, we show example reconstructions from both networks. The network pretrained on the synthetic images outperforms the other network in most of the cases. For cases such as challenging pose or up side down person, the network trained with real images typically fails, while the network pretrained on the synthetic dataset successfully reconstructs the person. The synthetic dataset is particularly useful for these cases as we can synthesis as many images with hard poses and rare camera setups as needed.

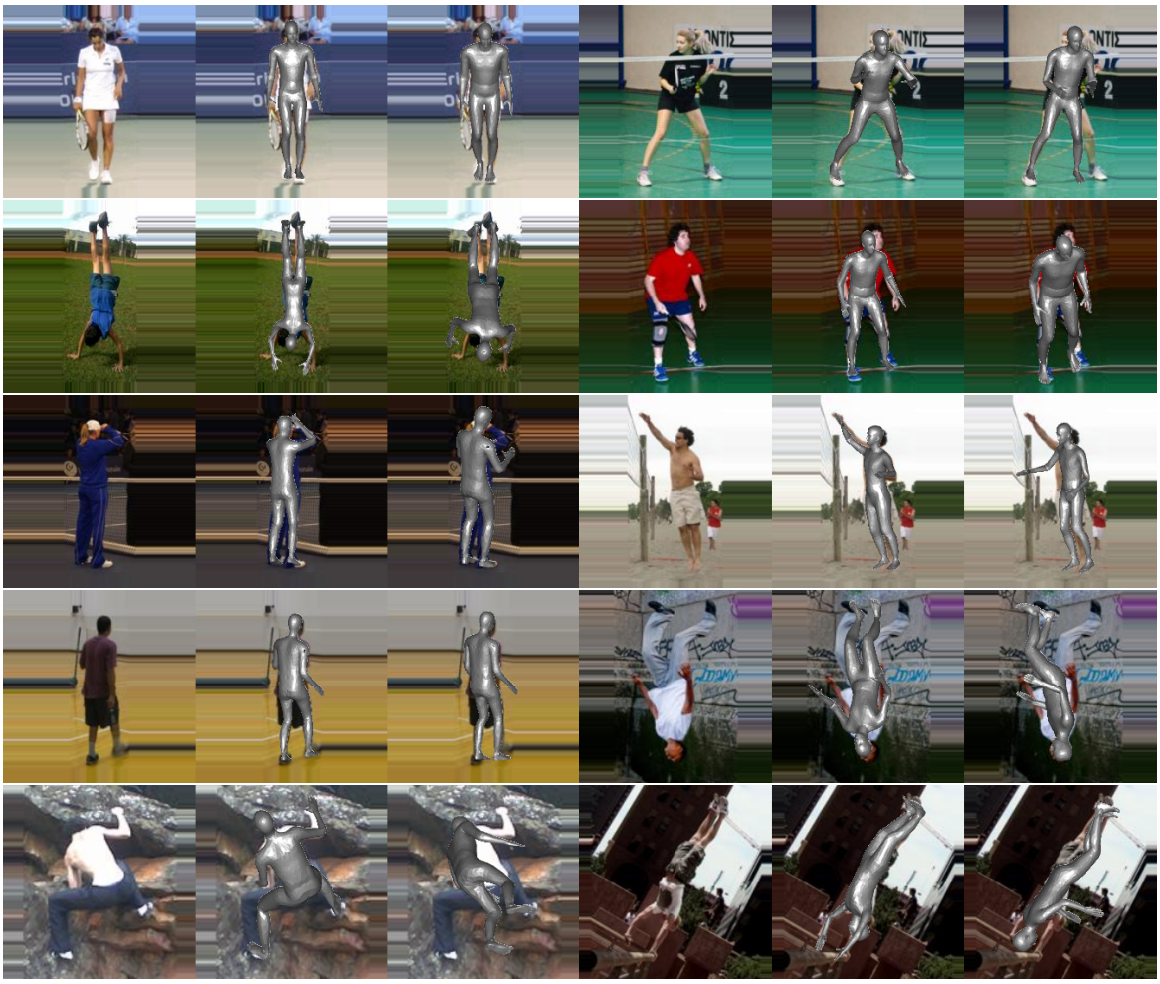


Figure 6.13: Qualitative results on LSP dataset. For each example in the figure: left: input image, middle: result from **real + synthetic** images trained network, right: result from **real** images trained network. Qualitatively the network trained with real + synthetic images outperforms the network trained with only real images.

6.5 Conclusion

In this chapter we presented the largest to date 4D dataset of clothed human bodies. The dataset contains over 1.3 million 3D scans of human body with high resolution textures. 7566 dynamic sequences of 3D meshes were captured in 30 FPS from 4205 subjects, resulting in over 12 hours of recordings in total.

We presented a robust and fast registration pipeline for registering the large number of high resolution 3D scans. Our registration approach exploits both 3D geometry information and texture information through 2D multiview rendered images which facilitates the use of deep CNNs for sparse keypoints and dense landmark localisation.

We evaluated the registration pipeline qualitatively and quantitatively and performed detailed analysis on the performance in extremely challenging cases. We showed that our proposed registration method guarantees accurate and realistic registrations even in the case of noisy scans.

We then demonstrated two use cases of our dataset in the context of 3D mesh synthesis and reconstruction. Our dataset covers a wide range of subjects in terms of age and ethnicity, facilitates the task of attribute driven mesh synthesis. We further exploit our parametric model based registration pipeline as a markerless motion capture and automatic rigging method for synthetic image and annotation generation, improving the performance of model based 3D human mesh recovery in monocular images.

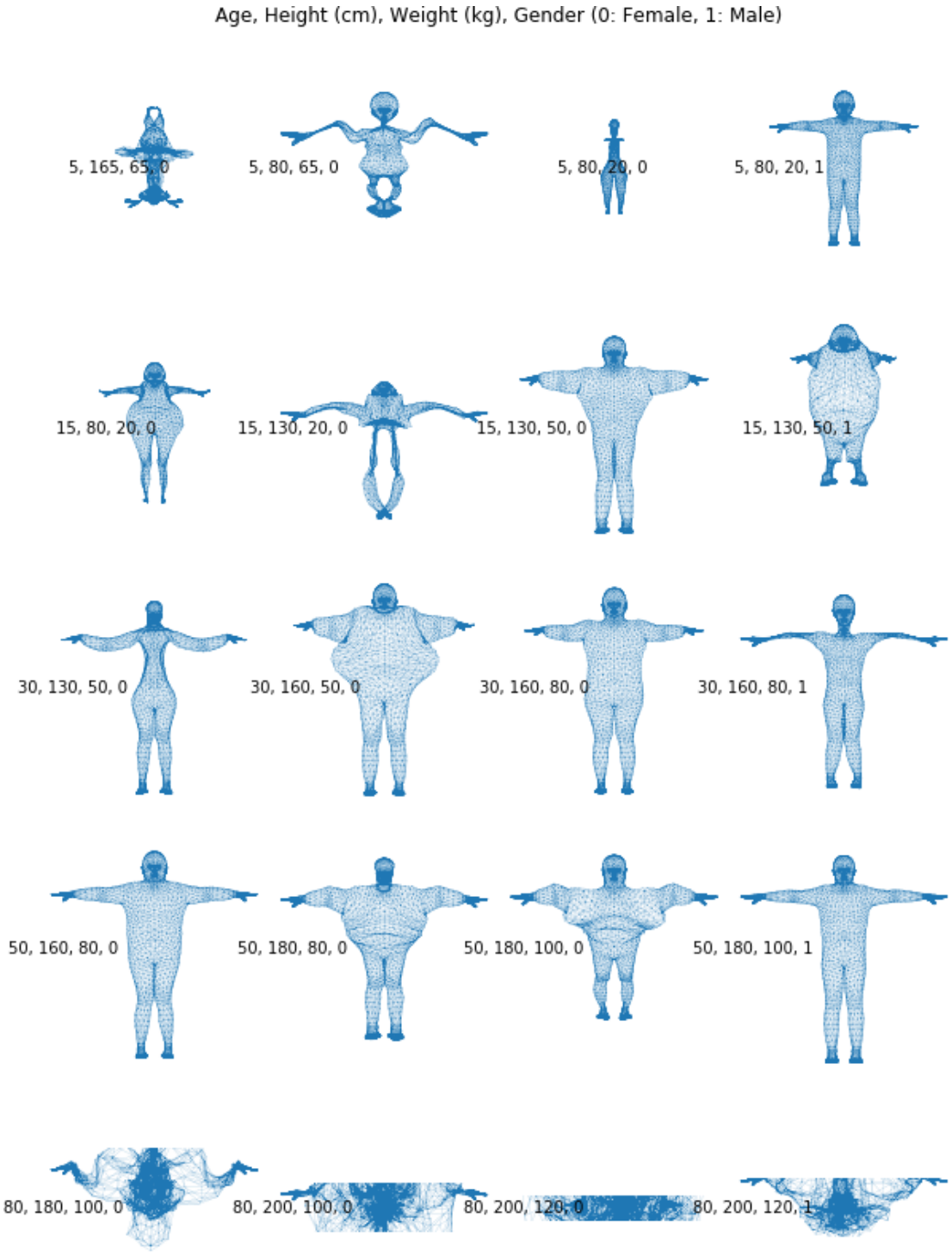


Figure 6.14: Shapes synthesised by regressing from the input attribute values to BLSM shape parameters.

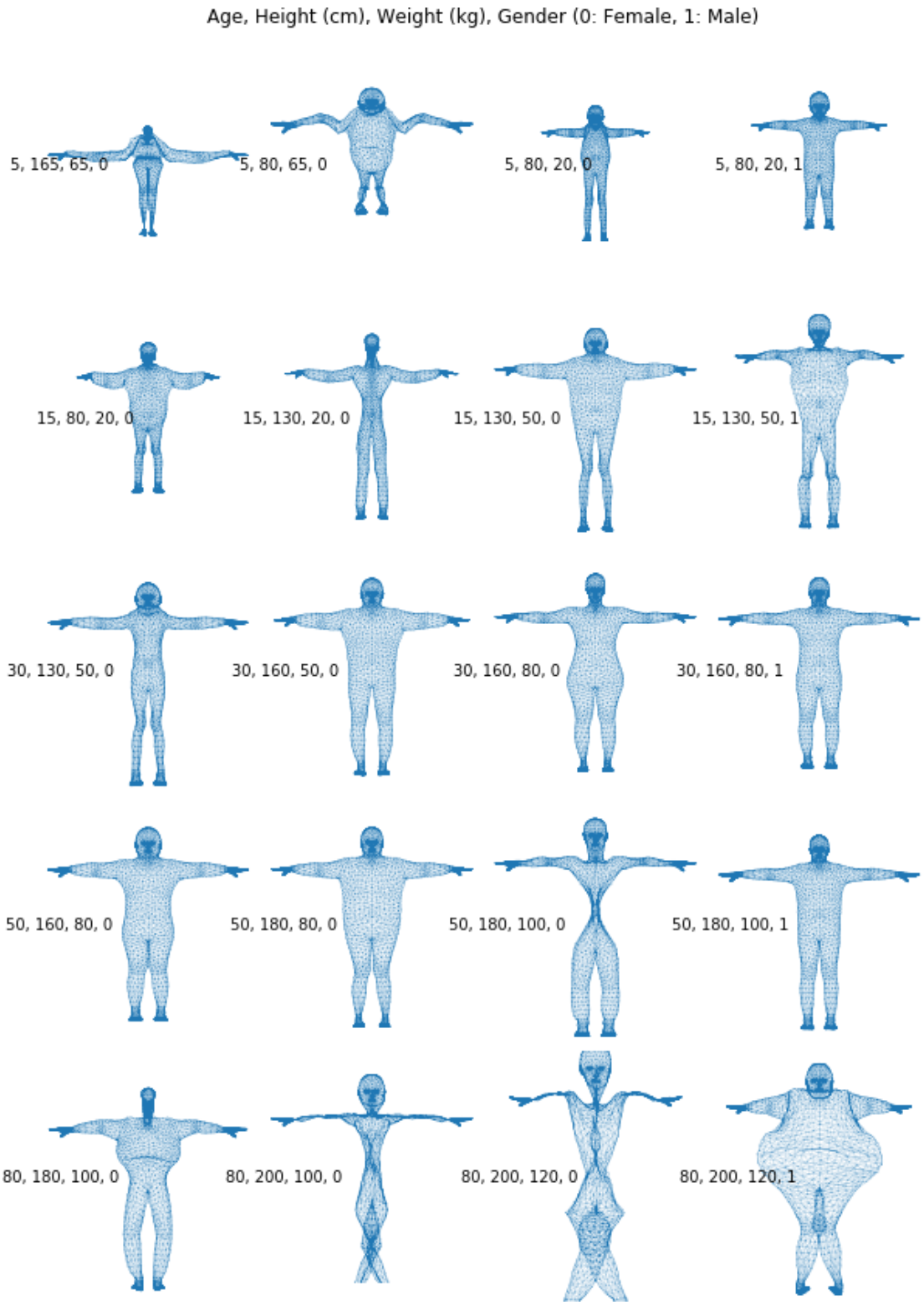


Figure 6.15: Shapes synthesised by regressing the PCA coefficents of the input attribute values to the BLSM shape parameters.

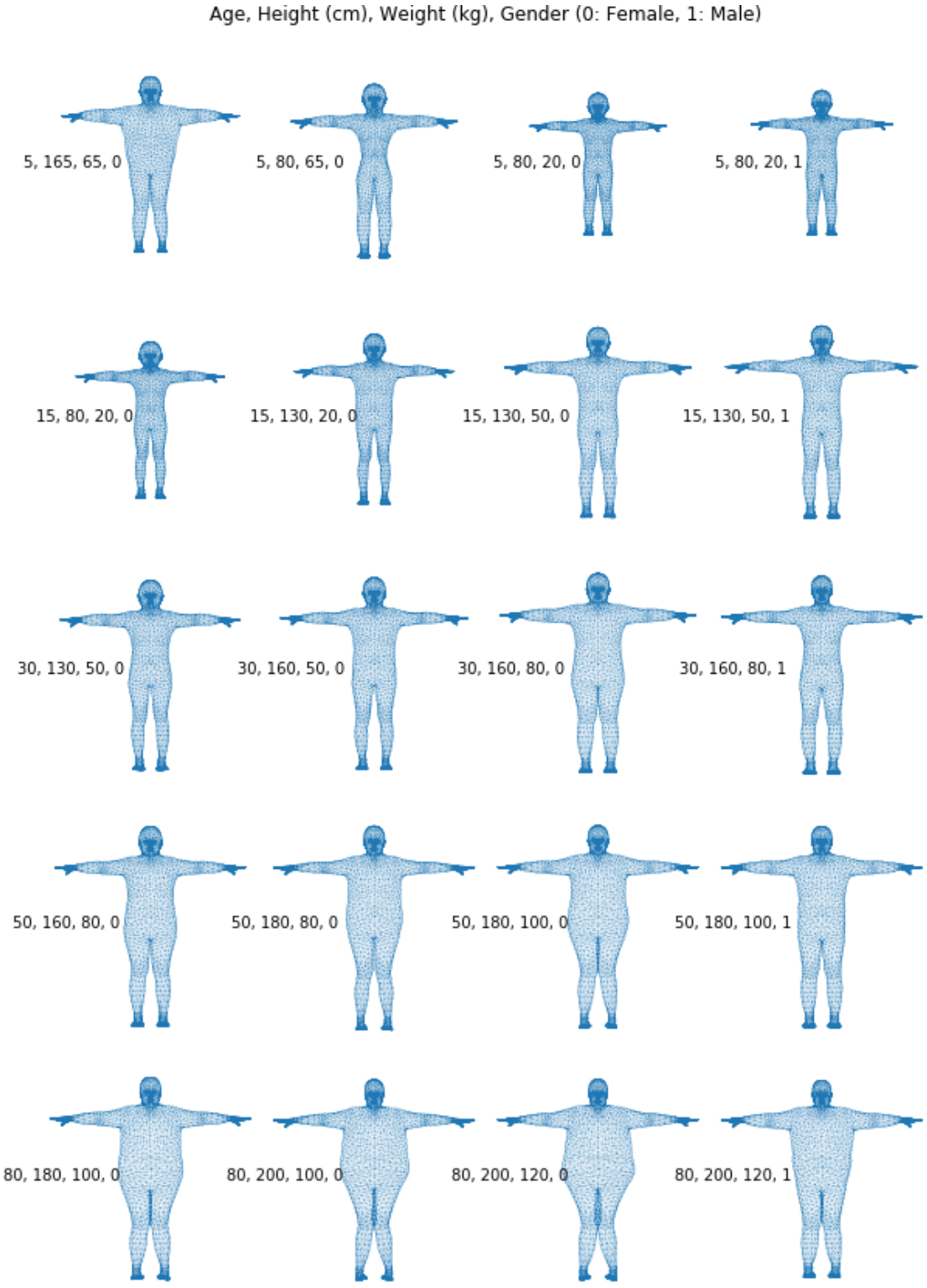


Figure 6.16: Shapes synthesised with a conditional GAN.

Chapter 7

Conclusion

7.1 Summary

In this thesis we aim at increasing the accuracy and exploring the applications of 3D human body modelling with deep deformable models.

Firstly a method for reparameterising 3D statistical shape models given a new template is proposed. Given a linear statistical shape model and a template of a different topology, we exploited the probabilistic nature of statistical shape models to compute a model in the new topology space without information loss. Throughout the thesis, this method provides the foundation for training downsteam methods with datasets annotated by parametric models with different topology and for comparing between models without the need of rebuilding the model with the original data.

We then presented a deep mesh convolutional network based parametric model for 3D human mesh recovery from monocular images. While most of the deep neural network based approach uses a linear blend skinning based head for mesh generation, we proposed to use a light weighted mesh convolutional decoder that directly operates on the 3D vertex locations, instead of the common practise that relies on regressing parametric model parameters.

We performed qualitative evaluation and demonstrated that the network outperforms the baseline method that regresses SMPL model parameters in the task of 3D pose estimation and 3D mesh reconstruction. We further improved the robustness of our network by using a mesh autoencoder based discriminator for dense adversarial training. The proposed network is also more flexible compared to the classic parametric

models in the sense that they can be trained end-to-end and fine-tuned given 3D mesh annotations, while linear blend skinning model based method requires pre-building a parametric model from 3D data and can only be fixed while used in deep neural networks.

However the downside of mesh convolutional based parametric model is that synthesising new meshes is not explicitly controllable. With the decoder we proposed, pose and shape variations of the human body are modelled in a single latent code space. Moreover, in order to train a network that is able to reconstruct fine details on extreme human poses, a large number of training data is needed.

To address this issue while exploting the advantage of mesh convolutional networks, we proposed a bone-level skinned model (BLSM). The key contribution of BLSM is the formulation where bone modelling and identity-specific variations are decoupled. Such formulation facilitates the use of mesh convolutional networks for capturing detailed identity specific variations, while explicitly controlling and modelling the pose variations through linear blend skinning with built-in motion constraints. Apart from being more accurate in the task of reconstructing 3D scans, we also demonstrated that the bone-level formulation which is compatible with any standard graphic packages allows for accurate in the wild character animation and retargetting.

So far the models we built were restricted to model human bodies in minimum clothing due to the limitations in the datasets. This leads to issues when applied to real world applications such as mesh recovery from images, as the reconstruction cannot capture details such as hair and clothing. To resolve this issue we presented the MeDigital dataset which contains over 1.3 million 3D scans of human body in daily clothing with high resolution textures. The dataset contains over 12 hours of 4D recordings at 30 FPS, consisting of 7566 dynamic sequences of 3D meshes from 4205 subjects. We proposed a fast and accurate sequence registration pipeline where the BLSM model is aligned to every frame of the dataset. While representing the raw scans in a low-poly common template and completing the noisy parts in the original data, our registration pipeline also facilitates markerless motion capture and automatic rigging of the raw scans, leading to automatic large scale synthetic image and annotation generation.

7.2 Future Work

The work in this thesis leads to many open questions and directions for future work, among which the most attracting direction could be representing, modelling and reconstructing high fidelity 3D human body meshes.

7.2.1 High Fidelity Clothed Human Modelling

Currently most of the parametric models of human body has limited resolution of 6K to 10K vertices. One consideration is the fact that the storage space required for linear blend skinning models grows linearly as the resolution of the template increases, consequently the inference time also increases. Apart from this, having a more detailed template has little gains in downstream applications as the models are built on scans of minimum clothed human bodies that lack of high frequency details. Following our work on capturing high resolution scans of clothed human bodies, the natural question to ask is then are the existing modelling methods sufficient and suitable for modelling clothed human bodies?

In the research community many approaches have been proposed to address the problem of clothes modelling. Unlike physics based clothes simulation that is commonly used in graphics community, the computer vision community has attempted to learn the clothes deformations in a data driven manner.

One line of work models the clothed human body as one mesh in a non-parametric manner [128][29]. Such methods could lead to high resolution reconstructions, however the result is not animatable and do not captures the relationship between the clothes deformation and the underlying human body.

Another line of work models the clothes and underlying human body as one mesh, however in which the clothes are represented as vertex displacements on top of a parametric model of human body [22][24][129]. In this case the clothes deformation can be represented as a blendshape controlled by the model's shape and pose parameter. Meshes generated from such models could suffer from the linear blend skinning artefacts as well as overly smoothed details.

Approaches such as [130][23] obtain realistic clothes that is modelled as a separate mesh, however such method could be difficult to apply to downstream applications

as the clothes need to be combined with the underlying body mesh, potentially by solving a complex optimisation problem.

In the mean time progress has been made on modelling and reconstructing high fidelity human faces. Mesh convolutional neural networks have been applied for generating high resolution 3D faces due to its light weight nauture [52][131]. Apart from mesh representations, representing shapes in UV space can also lead to high resolution shape generation by exploiting powerful image GANs [132][133].

Similar representations has been used to generate normal maps of clothes for more realistic rendering [130], but not on the whole body. Given the articulated nature of the human body, it could be difficult to model the shapes purely from mesh convolutional networks or shape UV images, however one interesting direction could be to combine linear blend skinning, which offers controllable articulated object modelling, with clothes deformation modelling through mesh convolutions or UV GANs for high resolution detail generation.

7.2.2 Robust Registration of Scans

One issue that could lead to problems while modelling clothed human bodies is the fact that reliable registrations of clothed human body scans are difficult to obtain. Some researchers attempted to place markers on the clothes while capturing clothes human body scans to guide the registration process [130], however such method lead to unrealistic textures. Segmentations could be performed on the raw scan to separate the clothes from the body region [134], however this requires learning from ground truth labels and potentially could not generalise well on all clothes types.

The registration pipeline we proposed in this thesis is good for capturing the shape and pose of the scan, however the registrations might suffer from vertex sliding and lead to noisy or overly smoothed meshes while trying to build high resolution models or synthesising clothes deformations. One thing that we did not attempt is to explicitly include a texture alignment term in the losses as this could significantly slow down the registration process. However this could be used as an extra refinement step on top of the geometry based results by exploiting the temporal information in the raw scan data. How to deal with noisy and flickering or missing patch in the texture maps remains a challenge for future research.

7.2.3 Monocular Human Mesh Recovery

What follows after modelling high fidelity clothed human body is to be able to reconstruct this from monocular images. First of all, monocular human mesh recovery performance could be boosted by exploiting the texture information in our presented dataset either by including a texture model in the training process or using differentiable renderers for detail refinment.

Another interesting problem to tackle is to generate realistic synthetic images from the 3D data and models. The synthetic data generation pipeline presented in this thesis do not take into account appropriate lighting, camera position and background. As a result, there exist domain gaps between the synthetic and real in the wild images, thus the performance gains from synthetic training is not considerable. A more reliable approach would be to synthesise images with camera approximation from real in the wild images. As such, the images could also be synthesised to simulate the real images that target the specific use case of the application. Alternatively, the synthetic image generation pipeline could focus on challenging cases that the current system fails to predict to compensate for the lack of hard training samples.

Bibliography

[1] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D Shapenets: A Deep Representation for Volumetric Shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 6, 21

[2] Robert W Sumner and Jovan Popović. Deformation Transfer for Triangle Meshes. *ACM Transactions on graphics (TOG)*, 23(3):399–405, 2004. 6, 24, 90

[3] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A Skinned Multi-person Linear Model. *ACM transactions on graphics (TOG)*, 34(6):248, 2015. 6, 24, 25, 26, 36, 37, 39, 40, 47, 69, 70, 71, 72, 84, 95

[4] Chao Zhang, Behrend Heeren, Martin Rumpf, and William AP Smith. Shell PCA: Statistical Shape Modelling in Shell Space. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1671–1679, 2015. 6, 26, 27, 37

[5] Marcel Lüthi, Thomas Gerig, Christoph Jud, and Thomas Vetter. Gaussian Process Morphable Models. *IEEE transactions on pattern analysis and machine intelligence*, 2017. 6, 27, 28, 32

[6] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3D Faces Using Convolutional Mesh Autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 704–720, 2018. 7, 29, 53, 72, 77

[7] Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Michael Bronstein, and Stefanos Zafeiriou. Neural 3D Morphable Models: Spiral Convolutional Networks for 3D Shape Representation Learning and Generation. In *Proceedings of the*

IEEE International Conference on Computer Vision, pages 7213–7222, 2019. 7, 29, 51, 52, 72, 77

[8] Zhiqin Chen and Hao Zhang. Learning Implicit Fields for Generative Shape Modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 7, 22, 30, 31

[9] Mixamo. `https://www.mixamo.com`, 2019. 9, 89, 90

[10] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end Recovery of Human Shape and Pose. In *Computer Vision and Pattern Regognition (CVPR)*, 2018. 12, 30, 33, 35, 48, 50, 55, 57, 59, 60, 61, 65, 66, 71, 75, 90, 113

[11] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J Black. Dynamic FAUST: Registering Human Bodies in Motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6233–6242, 2017. 16, 78, 83, 95, 99

[12] Kathleen M Robinette, Sherri Blackwell, Hein Daanen, Mark Boehmer, and Scott Fleming. Civilian American and European Surface Anthropometry Resource (CAESAR), Final Report. Volume 1. Summary. Technical report, SYTRONICS INC DAYTON OH, 2002. 16, 78, 95, 99

[13] Alexandre Saint, Eman Ahmed, Kseniya Cherenkova, Gleb Gusev, Djamila Aouada, Bjorn Ottersten, et al. 3dbodytex: Textured 3D Body Dataset. In *2018 International Conference on 3D Vision (3DV)*, pages 495–504. IEEE, 2018. 16, 99

[14] Daniel Maturana and Sebastian Scherer. Voxnet: A 3D Convolutional Neural Network for Real-time Object Recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 922–928. IEEE, 2015. 21

[15] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree Generating Networks: Efficient Convolutional Architectures for High-Resolution 3D Outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2088–2096, 2017. 21

[16] Ismail Khalid Kazmi, Lihua You, and Jian Jun Zhang. A Survey of 2D and 3D Shape Descriptors. In *2013 10th International Conference Computer Graphics, Imaging and Visualization*, pages 1–10. IEEE, 2013. 21

[17] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local Deep Implicit Functions for 3D Shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020. 22, 30

[18] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 22, 30

[19] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 22, 30

[20] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. DISN: Deep Implicit Surface Network for High-quality Single-view 3D Reconstruction. In *Advances in Neural Information Processing Systems*, pages 492–502, 2019. 22

[21] Alan Brunton, Augusto Salazar, Timo Bolkart, and Stefanie Wuhrer. Review of Statistical Shape Spaces for 3D Data with Comparative Analysis for Human Faces. *Computer Vision and Image Understanding*, 128:1–17, 2014. 23

[22] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video Based Reconstruction of 3D People Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8387–8397, 2018. 34, 35, 108, 122

[23] Peng Guan, Loretta Reiss, David A Hirshberg, Alexander Weiss, and Michael J Black. DRAPE: DRessing Any PErson. 34, 35, 72, 108, 122

[24] Qianli Ma, Siyu Tang, Sergi Pujades, Gerard Pons-Moll, Anurag Ranjan, and Michael J Black. Dressing 3D Humans using a Conditional Mesh-VAE-GAN. 29, 34, 35, 108, 122

[25] Qingyang Tan, Lin Gao, Yu-Kun Lai, and Shihong Xia. Variational Autoencoders for Deforming 3D Mesh Models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5841–5850, 2018. 28, 35

[26] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Feastnet: Feature-steered Graph Convolutions for 3D Shape Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2598–2606, 2018. 29, 35, 72

[27] Isaak Lim, Alexander Dielen, Marcel Campen, and Leif Kobbelt. A Simple Approach to Intrinsic Correspondence Learning on Unstructured 3D Meshes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 29, 35, 51, 72

[28] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional Mesh Regression for Single-image Human Shape Reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4501–4510, 2019. 29, 33, 35, 50, 65, 66

[29] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned Implicit Function for High-resolution Clothed Human Digitization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2304–2314, 2019. 30, 35, 111, 122

[30] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape Completion and Animation of People. In *ACM transactions on graphics (TOG)*, volume 24, pages 408–416. ACM, 2005. 23, 26, 70, 72, 95, 99

[31] Nils Hasler, Carsten Stoll, Martin Sunkel, Bodo Rosenhahn, and H-P Seidel. A Statistical Model of Human Pose and Body Shape. In *Computer graphics forum*, volume 28, pages 337–346. Wiley Online Library, 2009. 23, 72, 99, 108, 110

[32] Michael Kazhdan and Hugues Hoppe. Screened Poisson Surface Reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):29, 2013. 23

[33] Leonid Pishchulin, Stefanie Wuhrer, Thomas Helten, Christian Theobalt, and Bernt Schiele. Building Statistical Shape Spaces for 3D Human Modeling. *Pattern Recognition*, 67:276–286, 2017. 24, 26, 32, 72, 99

[34] Stefanie Wuhrer, Chang Shu, and Pengcheng Xi. Posture-Invariant Statistical Shape Analysis using Laplace Operator. *Computers & Graphics*, 36(5):410–416, 2012. 25

[35] Alexandros Neophytou and Adrian Hilton. Shape and Pose Space Deformation for Subject Specific Animation. In *3D Vision-3DV 2013, 2013 International Conference on*, pages 334–341. IEEE, 2013. 25

[36] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active Appearance Models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001. 26

[37] Volker Blanz and Thomas Vetter. Face Recognition Based on Fitting a 3D Morphable Model. *IEEE Transactions on pattern analysis and machine intelligence*, 25(9):1063–1074, 2003. 26, 36, 37

[38] James Booth, Epameinondas Antonakos, Stylianos Ploumpis, George Trigeorgis, Yannis Panagakis, Stefanos Zafeiriou, et al. 3D Face Morphable Models "In-The-Wild". In *Proceedings of the IEEE Conference on ComputerVision and Pattern Recognition*, 2017. 26, 36, 37

[39] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. A 3D Morphable Model Learnt from 10,000 Faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5543–5552, 2016. 26

[40] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning A Model of Facial Shape and Expression From 4D Scans. *ACM Transactions on Graphics (TOG)*, 36(6):194, 2017. 26, 32, 37

[41] Brett Allen, Brian Curless, Brian Curless, and Zoran Popović. The Space of Human Body Shapes: Reconstruction and Parameterization From Range Scans. In *ACM transactions on graphics (TOG)*, volume 22, pages 587–594. ACM, 2003. 26, 72, 109

[42] Hyewon Seo, Young In Yeo, and Kwangyun Wohn. 3D Body Reconstruction from Photos Based on Range Scan. In *International Conference on Technologies for E-Learning and Digital Entertainment*, pages 849–860. Springer, 2006. 26

[43] Yu Chen and Roberto Cipolla. Learning Shape Priors for Single View Reconstruction. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1425–1432. IEEE, 2009. 26

[44] Jonathan Boisvert, Chang Shu, Stefanie Wuhrer, and Pengcheng Xi. Three-Dimensional Human Shape Inference from Silhouettes: Reconstruction and Validation. *Machine vision and applications*, 24(1):145–157, 2013. 26

[45] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics (TOG)*, 36(6):245, 2017. 26, 32

[46] Martin Rumpf and Benedikt Wirth. An Elasticity-Based Covariance Analysis of Shapes. *International Journal of Computer Vision*, 92(3):281–295, 2011. 27

[47] Marcel Lüthi, Thomas Gerig, Christoph Jud, and Thomas Vetter. Gaussian Process Morphable Models. *IEEE transactions on pattern analysis and machine intelligence*, 2017. 27, 38

[48] Hang Dai, Nick Pears, and William Smith. Non-Rigid 3D Shape Registration using an Adaptive Template. *arXiv preprint arXiv:1803.07973*, 2018. 27, 32

[49] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. GHUM & GHUML: Generative 3D Human Shape and Articulated Pose Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6184–6193, 2020. 28, 99

[50] Qingyang Tan, Lin Gao, Yu-Kun Lai, Jie Yang, and Shihong Xia. Mesh-based Autoencoders for Localized Deformation Component Analysis. *arXiv preprint arXiv:1709.04304*, 2017. 29

[51] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016. 29, 51, 72

[52] Shiyang Cheng, Michael Bronstein, Yuxiang Zhou, Irene Kotsia, Maja Pantic, and Stefanos Zafeiriou. Meshgan: Non-linear 3D Morphable Models of Faces. *arXiv preprint arXiv:1903.10384*, 2019. 30, 51, 123

[53] Stan Sclaroff and Alex Pentland. Generalized Implicit Functions for Computer Graphics. *ACM Siggraph Computer Graphics*, 25(4):247–250, 1991. 30

[54] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal Step Nonrigid ICP Algorithms for Surface Registration. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 31, 32

[55] Tamas Varady, Ralph R Martin, and Jordan Cox. Reverse Engineering of Geometric Models — An Introduction. *Computer-aided design*, 29(4):255–268, 1997. 32

[56] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. FAUST: Dataset and Evaluation for 3D Mesh Registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014. 32, 99

[57] Paul J Besl and Neil D McKay. Method for Registration of 3D Shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992. 32

[58] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. FAUST: Dataset and Evaluation for 3D Mesh Registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014. 32

[59] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J Black. Dynamic FAUST: Registering Human Bodies in Motion. In *Proc. the Conference on Computer Vision and Pattern Recognition*, 2017. 32

[60] Olga Sorkine. Laplacian Mesh Processing. In *Eurographics (STARs)*, pages 53–70, 2005. 32

[61] Thomas W Sederberg and Scott R Parry. Free-form Deformation of Solid Geometric Models. *ACM SIGGRAPH computer graphics*, 20(4):151–160, 1986. 32

[62] John C Gower. Generalized Procrustes Analysis. *Psychometrika*, 40(1):33–51, 1975. 33

[63] Gary KL Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C Langbein, Yonghuai Liu, David Marshall, Ralph R Martin, Xian-Fang Sun, and Paul L Rosin. Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid. *IEEE transactions on visualization and computer graphics*, 19(7):1199–1217, 2013. 33

[64] Ankur Agarwal and Bill Triggs. Recovering 3D Human Pose From Monocular Images. *IEEE transactions on pattern analysis and machine intelligence*, 28(1):44–58, 2006. 33

[65] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-Time 3D Human Pose Estimation with a Single RGB Camera. *ACM Transactions on Graphics (TOG)*, 36(4):44, 2017. 33

[66] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape From A Single Image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016. 33, 48, 49, 50, 66, 85

[67] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to Reconstruct 3D Human Pose and Shape via Model-fitting in the Loop. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2252–2261, 2019. 33, 50, 90

[68] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. FrankMocap: Fast Monocular 3D Hand and Body Motion Capture by Regression and Integration. *arXiv preprint arXiv:2008.08324*, 2020. 33

[69] Verica Lazova, Eldar Insafutdinov, and Gerard Pons-Moll. 360-degree Textures of People in Clothing from a Single Image. In *2019 International Conference on 3D Vision (3DV)*, pages 643–653. IEEE, 2019. 34

[70] Oran Gafni, Lior Wolf, and Yaniv Taigman. Vid2game: Controllable Characters Extracted from Real-world Videos. *arXiv preprint arXiv:1904.08379*, 2019. 34

[71] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody Dance Now. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5933–5942, 2019. 34

[72] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Photo Wake-Up: 3D Character Animation From a Single Photo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 34

[73] Shizhe Zhou, Hongbo Fu, Ligang Liu, Daniel Cohen-Or, and Xiaoguang Han. Parametric Reshaping of Human Bodies in Images. *ACM transactions on graphics (TOG)*, 29(4):1–10, 2010. 34

[74] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-Garment Net: Learning to Dress 3D People From Images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 34

[75] Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. TailorNet: Predicting Clothing in 3D as a Function of Human Pose, Shape and Garment Style.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 34

[76] James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. Large Scale 3D Morphable Models. *International Journal of Computer Vision*, 126(2-4):233–254, 2018. 36, 37, 40

[77] Brett Allen, Brian Curless, and Zoran Popović. The Space of Human Body Shapes: Reconstruction and Parameterization From Range Scans. In *ACM transactions on graphics (TOG)*, volume 22, pages 587–594. ACM, 2003. 36, 37

[78] Hyewon Seo, Young In Yeo, and Kwangyun Wohn. 3D Body Reconstruction From Photos Based on Range Scan. In *International Conference on Technologies for E-Learning and Digital Entertainment*, pages 849–860. Springer, 2006. 36, 37

[79] Jonathan Boisvert, Chang Shu, Stefanie Wuhrer, and Pengcheng Xi. Three-Dimensional Human Shape Inference From Silhouettes: Reconstruction and Validation. *Machine vision and applications*, 24(1):145–157, 2013. 36, 37

[80] Tobias Heimann and Hans-Peter Meinzer. Statistical Shape Models for 3D Medical Image Segmentation: A Review. *Medical image analysis*, 13(4):543–563, 2009. 36

[81] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3D Face Model for Pose and Illumination Invariant Face Recognition. In *Advanced video and signal based surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 296–301. Ieee, 2009. 37

[82] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics (TOG)*, 36(6):245, 2017. 37

[83] Leonid Pishchulin, Stefanie Wuhrer, Thomas Helten, Christian Theobalt, and Bernt Schiele. Building Statistical Shape Spaces for 3D Human Modeling. *Pattern Recognition*, 67:276–286, 2017. 40

[84] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal Step Nonrigid ICP Algorithms for Surface Registration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 40

[85] Martin A Styner, Kumar T Rajamani, Lutz-Peter Nolte, Gabriel Zsemlye, Gábor Székely, Christopher J Taylor, and Rhodri H Davies. Evaluation of 3D Correspondence Methods for Model Building. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 63–75. Springer, 2003. 41

[86] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the People: Closing the Loop Between 3D and 2D Human Representations. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 3, 2017. 43, 49, 65, 66, 114

[87] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to Estimate 3D Human Pose and Shape from a Single Color Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 459–468, 2018. 48, 65, 66

[88] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural Body Fitting: Unifying Deep Learning and Model Based Human Pose and Shape Estimation. In *2018 international conference on 3D vision (3DV)*, pages 484–494. IEEE, 2018. 48, 50, 55, 65, 66

[89] Yu Chen, Tae-Kyun Kim, and Roberto Cipolla. Inferring 3D Shapes and Deformations from Single Views. In *European Conference on Computer Vision*, pages 300–313. Springer, 2010. 49

[90] Peng Guan, Alexander Weiss, Alexandru O Balan, and Michael J Black. Estimating Human Shape and Pose from a Single Image. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1381–1388. IEEE, 2009. 49

[91] Nils Hasler, Hanno Ackermann, Bodo Rosenhahn, Thorsten Thormählen, and Hans-Peter Seidel. Multilinear Pose and Body Shape Estimation of Dressed Subjects from Image Sets. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1823–1830. IEEE, 2010. 49

[92] Matthew M Loper and Michael J Black. OpenDR: An Approximate Differentiable Renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014. 49

[93] Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric Inference of 3D Human

Body Shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 20–36, 2018. 50

[94] Aaron S Jackson, Chris Manafas, and Georgios Tzimiropoulos. 3D Human Body Reconstruction from a Single Image via Volumetric Regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 50

[95] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense Human Pose Estimation In The Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018. 50, 96, 104

[96] Riza Alp Guler and Iasonas Kokkinos. Holopose: Holistic 3D Human Reconstruction In-The-Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10884–10894, 2019. 50, 60, 71, 75, 90

[97] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral Networks and Locally Connected Networks on Graphs. *arXiv preprint arXiv:1312.6203*, 2013. 51

[98] Anurag Ranjan, Timo Bolkart, and Michael J Black. Convolutional Mesh Autoencoders for 3D Face Representation. 2018. 51

[99] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic Convolutional Neural Networks on Riemannian Manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015. 51

[100] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning Shape Correspondence with Anisotropic Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, pages 3189–3197, 2016. 51, 72

[101] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric Deep Learning on Graphs and Manifolds using Mixture Model CNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017. 51

[102] David Berthelot, Thomas Schumm, and Luke Metz. BEGAN: Boundary Equilibrium Generative Adversarial Networks. *arXiv preprint arXiv:1703.10717*, 2017. 57

[103] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014. 58, 99

[104] Matthew Loper, Naureen Mahmood, and Michael J Black. MoSh: Motion and Shape Capture from Sparse Markers. *ACM Transactions on Graphics (TOG)*, 33(6):1–13, 2014. 58

[105] Jun Kai Vince Tan, Ignas Budvytis, and Roberto Cipolla. Indirect Deep Structured Learning for 3D Human Body Shape and Pose Prediction. 2017. 66

[106] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a Model of Facial Shape and Expression From 4D Scans. *ACM Transactions on Graphics (TOG)*, 36(6):194, 2017. 71

[107] John P Lewis, Matt Cordner, and Nickson Fong. Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-driven Deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 165–172. ACM Press/Addison-Wesley Publishing Co., 2000. 72

[108] Brett Allen, Brian Curless, and Zoran Popović. Articulated Body Deformation From Range Scan Data. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 612–619. ACM, 2002. 72

[109] Hyewon Seo, Frederic Cordier, and Nadia Magnenat-Thalmann. Synthesizing Animatable Body Models with Parameterized Shape Modifications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 120–125. Eurographics Association, 2003. 72

[110] Yinpeng Chen, Zicheng Liu, and Zhengyou Zhang. Tensor-Based Human Body Modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 105–112, 2013. 72

[111] David A Hirshberg, Matthew Loper, Eric Rachlin, and Michael J Black. Coregistration: Simultaneous Alignment and Modeling of Articulated 3D Shape. In *European Conference on Computer Vision*, pages 242–255. Springer, 2012. 72, 95

[112] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J Black. Dyna: A Model of Dynamic Human Shape in Motion. *ACM Transactions on Graphics (TOG)*, 34(4):120, 2015. 72

[113] Brett Allen, Brian Curless, Zoran Popović, and Aaron Hertzmann. Learning A Correlated Model of Identity and Pose-Dependent Body Shape Variation for Real-Time Synthesis. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 147–156. Eurographics Association, 2006. 72

[114] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total Capture: A 3D Deformation Model for Tracking Faces, Hands, and Bodies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8320–8329, 2018. 72

[115] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics (TOG)*, 36(6):245, 2017. 72

[116] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Laplacian Mesh Optimization. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 381–389, 2006. 80

[117] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3D Deep Learning with PyTorch3D. *arXiv:2007.08501*, 2020. 82

[118] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering Accurate 3D Human Pose in The Wild Using IMUs and a Moving Camera. In *European Conference on Computer Vision (ECCV)*, sep 2018. 90, 99

[119] Jiashun Wang, Chao Wen, Yanwei Fu, Haitao Lin, Tianyun Zou, Xiangyang Xue, and Yinda Zhang. Neural Pose Transfer by Spatially Adaptive Instance Normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5831–5839, 2020. 90

[120] Zhixuan Yu, Jae Shin Yoon, In Kyu Lee, Prashanth Venkatesh, Jaesik Park, Jihun Yu, and Hyun Soo Park. HUMBI: A Large Multiview Dataset of Human Body Expression. June 2020. 99

[121] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of Motion Capture as Surface Shapes. In *International Conference on Computer Vision*, pages 5442–5451, October 2019. 99

[122] Yipin Yang, Yao Yu, Yu Zhou, Sidan Du, James Davis, and Ruigang Yang. Semantic Parametric Reshaping of Human Body Models. In *2014 2nd International Conference on 3D Vision*, volume 2, pages 41–48. IEEE, 2014. 99, 108

[123] Zongyi Xu, Qianni Zhang, and Shiyang Cheng. Multilevel Active Registration for Kinect Human Body Scans: From Low Quality to High Quality. *Multimedia Systems*, 24(3):257–270, 2018. 99

[124] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer Science & Business Media, 2008. 99

[125] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation. In *CVPR*, 2020. 102

[126] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from Synthetic Humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2017. 111

[127] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 Million Image Database for Scene Recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 113

[128] Ryota Natsume, Shunsuke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeo Morishima. SiCloPe: Silhouette-based Clothed People. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4480–4490, 2019. 122

[129] Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. MonoPerfCap: Human Performance Capture from Monocular Video. *ACM Transactions on Graphics (ToG)*, 37(2):1–15, 2018. 122

[130] Zorah Lahner, Daniel Cremers, and Tony Tung. DeepWrinkles: Accurate and Realistic Clothing Modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 667–684, 2018. 122, 123

[131] Yuxiang Zhou, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Dense 3D Face Decoding Over 2500FPS: Joint Texture and Shape Convolutional Mesh Decoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 123

[132] Baris Gecer, Alexandros Lattas, Stylianos Ploumpis, Jiankang Deng, Athanasios Papaioannou, Stylianos Moschoglou, and Stefanos Zafeiriou. Synthesizing Coupled 3D Face Modalities by Trunk-branch Generative Adversarial Networks. In *European Conference on Computer Vision*, pages 415–433. Springer, 2020. 123

[133] Stylianos Moschoglou, Stylianos Ploumpis, Mihalis Nicolaou, Athanasios Papaioannou, and Stefanos Zafeiriou. 3DFaceGAN: Adversarial Nets for 3D Face Representation, Generation, and Translation. 123

[134] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment Net: Learning to Dress 3D People from Images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5420–5430, 2019. 123