3D Human Body Mesh Generation from 2D Images Using Body Silhouette, Bone Orientation, and Joints Triangulation

by

Jordy AJANOHOUN

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT OF A MASTER'S DEGREE
WITH THESIS IN INFORMATION TECHNOLOGY ENGINEERING
M.A.Sc.

MONTREAL, APRIL 21, 2021

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Carlos Vázquez, supervisor
Department of Software Engineering and Information Technology, École de technologie supérieure

Mr. Eric Paquette, co-supervisor
Department of Software Engineering and Information Technology, École de technologie supérieure

Mr. David Labbé, president of the board of examiners
Department of Software Engineering and Information Technology, École de technologie supérieure

Mr. Simon Drouin, external examiner
Department of Software Engineering and Information Technology, École de technologie supérieure

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON APRIL 16, 2021

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

First of all, I would like to thank and express my sincere gratitude to my supervisors Carlos VÁZQUEZ and Eric PAQUETTE, for this incredible opportunity they gave to me, their ongoing support, and their trust. I have grown and learned so much by their side regarding research and academic skills. Always with kindness, without forgetting to push me further, to a higher level.

I would also like to acknowledge and show my gratefulness to the industrial partner of this master's thesis, BodyForm3D (directed by Antonin BÉRUBÉ), which helped to delimit the research and made available its technology. By the side of BodyForm3D, I have gained experience in teamwork and learned skills relevant to industry.

Besides, I would like to extend my deep gratitude to Polytech Sorbonne-University, which made possible my double degree with École de technologie supérieure to conduct this thesis. Please, keep your high standards and continue to give birth to talented engineers with your high-quality education. I could not have accomplished this thesis and my achievements without all the prerequisite knowledge I learned from the institution.

Last but not least, I would like to thank my family from the deep down of my heart for their love, encouragement, and wisdom. You are always there for me. Also, I could not have completed this work without the support of my friends. Special thanks to Nestor MARTINEZ and Alberto MARTINEZ for their interest in my work and their reflections, Constance JACQUES for her patience and affection, Inès LANKRI and Inès LABIADH for always make me laugh and enjoy life.

Génération d'avatar 3D du corps humain à partir d'images 2D en utilisant la silhouette du corps, l'orientation des os et la triangulation des articulations

Jordy AJANOHOUN

RÉSUMÉ

Dans ce mémoire, nous abordons le problème de l'estimation de la pose et de la morphologie 3D d'une personne à partir d'images multi-vues. Comme d'autres méthodes adressant ce problème, nous utilisons le modèle corporel paramétrique nommé Skinned Multi-Person Linear model (SMPL). L'objectif est alors de trouver les paramètres du modèle SMPL qui correspondent le mieux à la morphologie et à la pose 3D de l'individu sur les images. Le principal défi réside dans l'estimation précise de ces paramètres. Pour y parvenir, nous estimons dans un premier temps la localisation 2D des articulations de l'individu sur l'image. Ensuite, au moyen d'une triangulation algébrique linéaire, nous estimons la localisation 3D de ces articulations à partir des localisations 2D estimées. Cela permet d'obtenir une estimation de la localisation 3D des articulations plus fiable. Par la suite, nous faisons correspondre les articulations 3D et la silhouette 2D du modèle 3D du corps avec celles estimées de l'individu. Pour ce faire, nous introduisons un nouveau processus d'optimisation en deux étapes avec une nouvelle fonction objectif. Cette dernière permet d'obtenir une meilleure initialisation pour l'optimisation finale et simultanée de la pose et la morphologie. Enfin, nous mettons en lumière que la position sémantique des articulations dans le modèle paramétrique et dans les bases de données d'évaluation n'est pas identique. Pour tenir compte de cette divergence, nous introduisons, pour chaque articulation, un vecteur de recalage calculé dans le repère local de l'articulation. Notre approche entièrement automatique est évaluée sur les bases de données Human3.6M et HumanEva, montrant des résultats supérieurs aux méthodes de l'état de l'art.

Mots-clés: reconstruction 3D, estimation de pose et de morphologie humaine, modèle du corps, multi-vues, vision par ordinateur

3D Human Body Mesh Generation from 2D Images Using Body Silhouette, Bone Orientation, and Joints Triangulation

Jordy AJANOHOUN

ABSTRACT

In this dissertation, we address the problem of 3D human pose and shape estimation from multi-view images. Like similar methods, we make use of the Skinned Multi-Person Linear (SMPL) parametric body model, and try to regress the model parameters that best fit the shape and pose of the individual on the images. The main challenge lies in accurately inferring these parameters. To solve this problem, we first estimate 2D joints. Then, we use a linear algebraic triangulation to lift estimated 2D joints to 3D, resulting in a joint estimation with fewer errors. Next, we fit the 3D parametric body model to the 3D joints while imposing silhouette and bone orientation consistency between the 3D model and the detected individual in the images. We do so by minimizing a new set of objective functions through a two-step optimization process that provides a good initialization for the refinement of the shape and pose parameters. Finally, we demonstrate that the semantic position of joints in the body model and in the validation data sets do not exactly match. To account for this discrepancy we introduce, for each joint, a shift vector computed in the joint's local space. Our fully automatic approach is evaluated on the widely used benchmarks Human3.6M and HumanEva, showing superior results with respect to state-of-the-art methods.

Keywords: 3D reconstruction, human shape and pose estimation, body model, multi-view, computer vision

TABLE OF CONTENTS

Page

INTRODUCTION 1

CHAPTER 1 RELATED WORK 5
1.1 3D Human Body Modeling 5
1.1.1 SMPL Model 8
1.2 Human Pose Estimation 14
1.2.1 2D pose estimation 14
1.2.2 3D pose estimation 18
1.3 3D Human Body Reconstruction from Images 20
1.3.1 CNN-based methods 20
1.3.2 Optimization-based methods 25

CHAPTER 2 MULTI-VIEW 3D BODY RECONSTRUCTION 31
2.1 2D Pose and Body Silhouette 32
2.2 3D Pose Triangulation 33
2.3 Two-Step Optimization Process 34

CHAPTER 3 EXPERIMENTAL RESULTS AND DISCUSSION 39
3.1 Validation on HumanEva-I 40
3.2 Generalization on Human3.6M 41
3.3 Qualitative Evaluation 43
3.4 Discussion 43

CONCLUSION AND RECOMMENDATIONS 47

BIBLIOGRAPHY 50

LIST OF FIGURES

Page

Figure 0.1 Input and desired output 1

Figure 1.1 Example of 3D meshes generated with SMPL. Taken from Loper, Mahmood, Romero, Pons-Moll & Black (2015) 8

Figure 1.2 SMPL overview. Taken from Loper *et al.* (2015) 9

Figure 1.3 Example of SMPL mesh with an impossible pose 13

Figure 1.4 Human pose estimation pipeline. Taken from Liu, Zhu, Bu & Chen (2015) 15

Figure 1.5 **Top:** 2D poses. **Bottom:** PAF for the right forearm. Taken from Cao, Martinez, Simon, Wei & Sheikh (2021) 16

Figure 1.6 OpenPose's CNNs architecture. Taken from Cao *et al.* (2021) 16

Figure 1.7 Left elbow and left shoulder heatmaps. Taken from Cao *et al.* (2021) 18

Figure 1.8 Human Mesh Recovery overview. Taken from Kanazawa, Black, Jacobs & Malik (2018) 21

Figure 1.9 Overview of Pavlakos, Zhu, Zhou & Daniilidis (2018). Taken from Pavlakos *et al.* (2018) 22

Figure 1.10 Neural Body Fitting overview. Taken from Omran, Lassner, Pons-Moll, Gehler & Schiele (2018) 23

Figure 1.11 Convolutional Mesh Regression overview. Taken from Kolotouros, Pavlakos & Daniilidis (2019b) 24

Figure 1.12 SMPLify overview. Taken from Huang, Bogo, Lassner, Kanazawa, Gehler, Romero, Akhter & Black (2017) 26

Figure 1.13 SMPL mesh approximation with capsules. Taken from Bogo, Kanazawa, Lassner, Gehler, Romero & Black (2016) 28

Figure 2.1 Overview of the proposed approach 32

Figure 2.2 Example of joints correction thanks to the triangulation 34

Figure 2.3 Example using our two-step optimization process 37

Figure 3.1 Shift between the ground-truth Human3.6M and the SMPL joints 40

Figure 3.2 Qualitative results on Human3.6M subject 9 44

Figure 3.3 Qualitative results on Human3.6M subject 11 45

LIST OF ABBREVIATIONS

BOC	Bone Orientation Constraint
CDCL	Cross-Domain Complementary Learning
CNN	Convolutional Neural Network
DQBS	Dual-Quaternion Blend Skinning
GAN	Generative Adversarial Network
HPE	Human Pose Estimation
LBS	Linear Blend Skinning
MPJPE	Mean Per Joint Position Error
MuVS	Multi-View SMPLify (Huang *et al.*, 2017)
PA	Procrustes Analysis
PAF	Part Affinity Field
PCA	Principal Components Analysis
SMPL	Skinned Multi-Person Linear model (Loper *et al.*, 2015)

INTRODUCTION

In areas such as virtual/augmented reality, healthcare, virtual try-on, and video games, it is important to be able to accurately model, in 3D, the body of an individual. It means to generate an accurate 3D mesh of the body. This problem is known as 3D human body reconstruction. It benefited from a lot of attention in recent years since it profits to various applications. For example, tailors could instantly extract measurements from the mesh, surgeons could plan various operations using the mesh, car crash-test simulations could be more and more realistic, clients could try-on clothes virtually online with their own specific avatar, and so on. In such contexts, only the shape and the pose of the subject are essential; we are not looking for hairiness, facial detail, or texture. To achieve so, related work relied mainly on three types of inputs: 3D point clouds, anthropometric measurements, and 2D images. This thesis investigates the generation of an individual's accurate 3D body mesh from 2D images (Fig. 0.1).

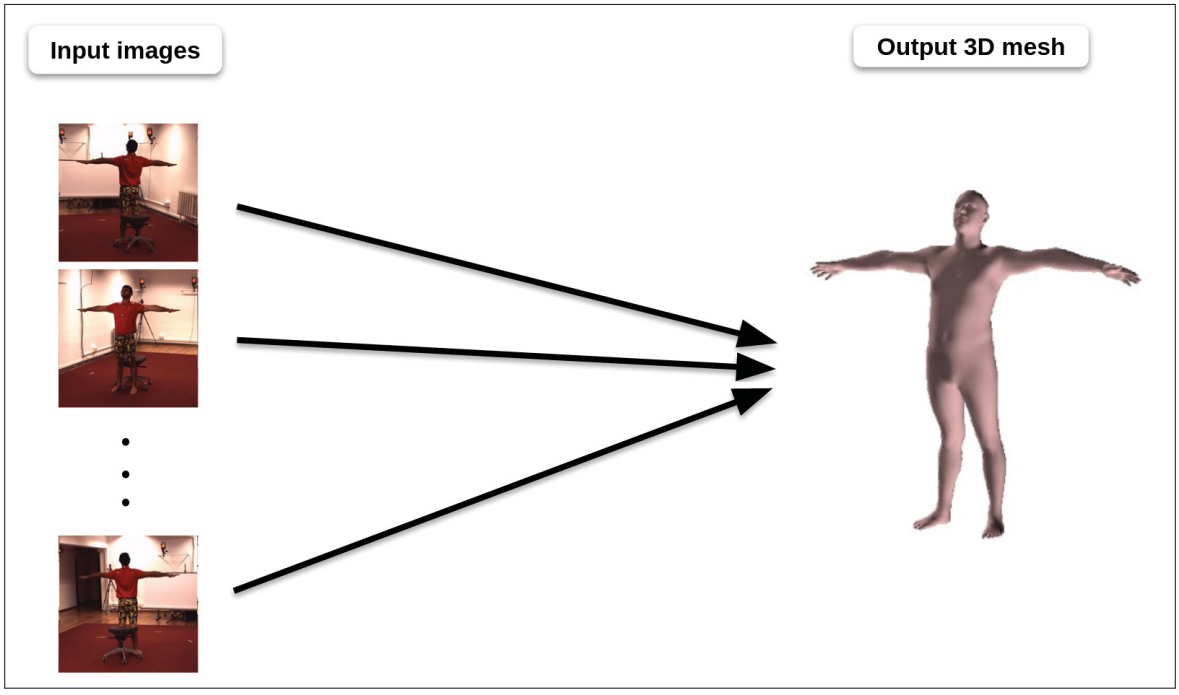


Figure 0.1 Input and desired output

We are interested in cases where the input is several images of a person taken at the same time. The subject can be in any pose and located anywhere on the image. Occlusion of some body parts is allowed but not of the whole body. The only constraints we have is that the whole body should appear on each images, and only the subject should be on these images. No other people should be visible. The output is an accurate full-body 3D mesh having the shape and the pose of the individual.

There are several challenges in this task. First, the human body is complex and there are as many shapes as humans. Besides, inferring 3D from 2D is ambiguous by nature. Occlusion makes the problem even harder since some body parts could be hidden.

Current methods use parametric human body models, such as the Skinned Multi-Person Linear model (SMPL), and try to regress the parameters that best fit the individual on the images. Some methods train a neural network to make this task, others design complex objective functions and optimize them looking for the accurate parameters. Training a neural network in order to accomplish this task is hard because there is not enough end-to-end annotated data to feed them (databases with images and the corresponding accurate 3D mesh or model parameters for each of them). Consequently, they have to use several tricks and other databases with 2D silhouettes and 2D poses to bypass this limitation. However, there is still an impact on the accuracy they can reach since the ideal data is lacking. Furthermore, most of them use only one view and suffer from the ambiguity of inferring 3D from 2D. The neural network is supposed to implicitly learn to deal with this ambiguity through the training, but often fails because the examples given to it do not associate 2D images to 3D mesh or 3D features. The examples map 2D images to other 2D features such as 2D pose and 2D silhouette. Having difficulties with overcoming these limitations, other efforts follow another path by setting up an optimization problem (no learning needed) with hand-designed constraints to deal with the ambiguity. These optimization-based

methods, contrary to efforts with neural networks, explicitly define the depth ambiguity problem through formalization with mathematical constraints.

Methods designing objective functions are more robust but heavily rely on the quality of the objective functions. 2D pose and 2D silhouette estimates are integrated in the objective functions. Therefore, the quality of these functions depends on the accuracy of the estimates besides the intrinsic quality of the terms and constraints they are made of. There is a need for the improvement of these estimates and the way they are integrated and used in the optimization process. The closest work to ours is Multi-View SMPLify (MuVS) by Huang *et al.* (2017). It is a multi-view and optimization-based method whose objective function is constructed from the estimated 2D pose and 2D silhouette on each view. MuVS aggregates the 2D pose and silhouette estimations from all the views into a single objective function. It helps to reduce the depth ambiguity but the objective function suffers from numerous local optima due to the way the aggregation is done (a sum through all the views). Consequently, MuVS only works well when initialized close to the real solution. Furthermore, MuVS uses the SMPL model (Loper *et al.*, 2015) and simultaneously optimizes the pose and shape parameters. Although this strategy works, it is not optimal because of the tight relationship and interdependence between the pose and the shape parameters in SMPL. This complex relationship makes the optimization problem harder and the initialisation even more crucial. Given the current limitations of the state of the art, our objective is to propose a more precise and robust method to generate a 3D mesh of a subject from images. In achieving our objective, we designed key contributions that can be summarized as follows:

- A bone orientation constraint (BOC) to recover the pose parameter independently from the shape parameter;
- A more precise initialization for the simultaneous optimization of pose and shape parameters thanks to the BOC;
- A two-step optimization process that improves the accuracy of the pose and shape estimations.

The remainder of this dissertation is organized as follows. First, the literature on human body models as well as pose and shape estimation from images are reviewed. Then, we describe our approach to accurately estimate 3D human pose and shape from multi-view images. Finally, experimental results are presented and discussed, before concluding with potential future works.

CHAPTER 1

RELATED WORK

The goal in 3D human body reconstruction is to generate an accurate 3D mesh of the subject's body. Therefore, the 3D mesh is central in this problem. In the related work, instead of computing each vertex location and how they are arranged in triangles to form the mesh, parametric human body models are used. They allow to generate a 3D body mesh from a specified set of parameters such as its shape and its pose. Parametric body models therefore considerably reduce the problem complexity, since only the value of the parameters have to be computed from the images, and not the complete mesh from scratch. Consequently, 3D body modeling is an essential part of the problem and the body model choice matters. Due to the key role of the 3D mesh, in this chapter, we first review the literature on 3D human body modeling. Then, since human pose estimation is systematically used and key to infer the model parameters, we present the related work on this topic. We conclude this chapter by reviewing in detail how current methods use the body model and the pose estimation to generate a subject's full-body 3D mesh from images. We take this occasion to also discuss the dissatisfaction with current methods.

1.1 3D Human Body Modeling

In 3D animation software and video games, to animate the characters modelled by a surface mesh, we equip them with bones and joints. We can see it as the process used to animate puppets or simply as the human body is made. The set of bones forms the skeleton and the mesh models the skin. Each vertex of the mesh is then linked (via a coefficient) to one or more joints in such a way that it follows the motion of the joints coherently. This technique is called blend skinning (Jacobson, Deng, Kavan & Lewis, 2014). The closer a vertex is to a particular joint, the higher the coefficient that binds the two will be. Therefore, the movement of the joint will have an important impact on the displacement of the vertex. On the other hand, the lower the coefficient is, the lower will be the influence. For example, the vertices that make up the right tibia will have a high coefficient with the right knee joint, but null with the left elbow joint. It

is intuitive, bending the right knee immediately changes the position of the right tibia. In the opposite, rotating the left forearm has no consequence on the position of the tibia.

There are different ways to calculate the response of a vertex to joint motion. Meaning its position after motion. One technique used for this is the linear blend skinning (LBS) (Lewis, Cordner & Fong, 2000) which, as its name indicates, simply uses a linear relationship taking into account the binding coefficients. Other approaches are not linear like the dual-quaternion blend skinning (DQBS) (Kavan, Collins, Žára & O'Sullivan, 2007). Each solution has its advantages and disadvantages. But generally, these blend skinning techniques have common well-known flaws such as taffy and bowtie effects (Jacobson *et al.*, 2014; Loper *et al.*, 2015).

This whole process with joints, bones, and vertices imitates the human body. It describes how the vertices of the mesh are connected to the bone structure and the joints. The difficulty is to determine which vertices to associate with which joints and with which weight (coefficient), but also which equation to use to calculate the new position of the vertices after a motion. Of course this requires a judicious and adequate choice of skeleton and joints beforehand. The goal is to achieve a modeling of the human body as realistic as possible and that remains so regardless of the poses.

Most research (Bogo *et al.*, 2016; Huang *et al.*, 2017; Kolotouros *et al.*, 2019b; Madadi, Bertiche & Escalera, 2020; Omran *et al.*, 2018; Pavlakos *et al.*, 2018) relies on body models based on statistical data on the human body to determine how to deform a mesh according to the pose and the shape. Indeed, the body of a 10-year-old child does not deform exactly the same way as the body of a 50-year-old adult. The use of a statistical approach makes it possible to take into account the different shapes. Of course, this is if the training data for the statistical model contains observations on a wide variety of human bodies. The data must be representative of the population.

Many efforts (Allen, Curless & Popović, 2003; Allen, Curless, Popović & Hertzmann, 2006; Chen, Liu & Zhang, 2013; Freifeld & Black, 2012; Hasler, Stoll, Sunkel, Rosenhahn & Seidel, 2009; Loper *et al.*, 2015) focus on the learning of a realistic model for the human body from

statistical data. These models consider that any human body can be modeled starting from a mesh of the average human body, which is then deformed to map the targeted shape. Deformation in this context means the displacement of some vertices or triangles of the mesh. This concept is fundamental and its formulation comes from Lewis *et al.* (2000). It is the way of defining, formalizing, and learning these deformations that varies across the models. For example, Allen *et al.* (2003) are only interested in the shape; their model does not take into account how the body deforms according to the pose. They use a principal component analysis (PCA) to characterize the space of human body shapes. Shape-based deformations then become a linear combinations of basic deformations. These basic deformations are the principal components retained from the PCA. Allen *et al.* (2006) focused on pose besides shape. They also used a PCA for the shape space. For the pose, their idea was to reduce the space of the poses to a few key poses, in the same spirit as the PCA. For each of these key poses, they calculated the displacement of each vertex relative to the starting average human body mesh. They thus obtained a base of vertex displacements that depend on the pose. The final pose-dependent deformations are a linear combination of the key pose displacements. Another example of body model focusing on pose besides shape is the BlendSCAPE model (Hirshberg, Loper, Rachlin & Black, 2012), an improvement and the successor of the SCAPE model (Anguelov, Srinivasan, Koller, Thrun, Rodgers & Davis, 2005). The idea in BlendSCAPE and SCAPE is similar to Allen *et al.* (2006) but a major difference is that the deformations are triangle-based in BlendSCAPE and SCAPE, instead of vertex-based for Allen *et al.* (2006). One of the main disadvantages of SCAPE is that it learns the shape-dependent and pose-dependent deformations independently, which neglects the correlations between the body shape and pose. BlendSCAPE improves SCAPE by taking into account these correlations. More recently, the SMPL model (Loper *et al.*, 2015) has been proposed. In SMPL, the pose-dependent and shape-dependent deformations are vertex-based and the correlations between both are integrated. Contrary to SCAPE, BlendSCAPE, and the model of Allen *et al.* (2006), SMPL models the deformations in a linear way which greatly improves the efficiency.

According to Loper *et al.* (2015), triangle-based models, even though generally providing the most realistic results, hardly integrate into the graphic pipeline and graphic engines. This is because they may not be based on a blend skinning technique contrary to vertex-based models. Comparing to other models, SMPL is the most accurate model with lowest computation complexity, still according to Loper *et al.* (2015). Since SMPL is the leading model in 3D body reconstruction from images (Bogo *et al.*, 2016; Huang *et al.*, 2017; Kolotouros *et al.*, 2019b; Madadi *et al.*, 2020; Omran *et al.*, 2018; Pavlakos *et al.*, 2018), we review below how it works and how it is built.

1.1.1 SMPL Model

SMPL allows to generate a realistic 3D human body mesh (Fig. 1.1) given a shape $\vec{\beta}$ and a pose $\vec{\theta}$ parameters specified by the user. This section presents an overview of the SMPL (Loper *et al.*, 2015) model. We refer the reader to the SMPL paper for a full description.

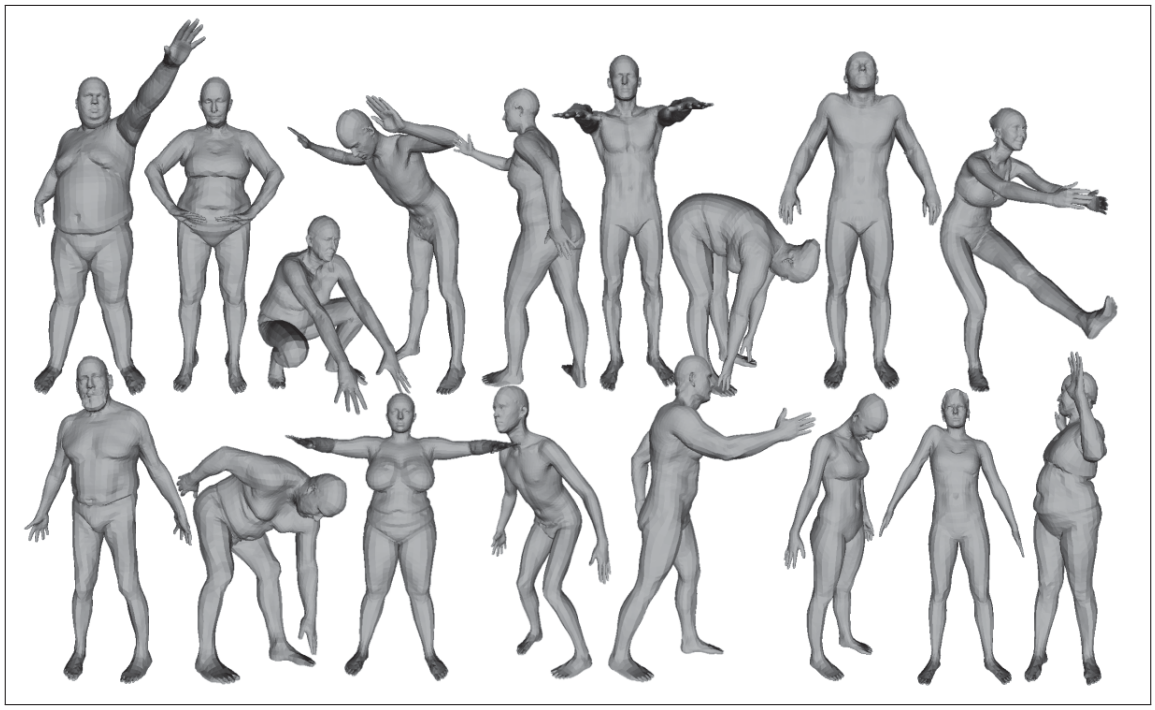


Figure 1.1 Example of 3D meshes generated with SMPL.
Taken from Loper *et al.* (2015)

To generate the mesh having the shape specified with $\vec{\beta}$ in the pose specified with $\vec{\theta}$, the SMPL procedure always begins with a mesh (called initial mesh) that models the average human body in "T" pose (Fig. 1.2a). This (T-shaped) pose is called the base pose or null pose and is denoted $\vec{\theta}^*$. The mesh consists of $N = 6890$ vertices and $K = 23$ joints. It can be written as a concatenation vector of the vertices: $\bar{\mathbf{T}} \in \mathbb{R}^{3N}$. It is also equipped with a skeleton, and each vertex is bound to each joint via a blend skinning coefficient (which can be zero). All these weights are stored in the vector $\mathcal{W} \in \mathbb{R}^{N \times K}$. No matter the value of $\vec{\beta}$ and $\vec{\theta}$, the SMPL procedure always begins with this initial mesh. Then, three steps (Fig. 1.2), are executed to deform this initial mesh towards the output mesh having the specified shape $\vec{\beta}$ and in the specified pose $\vec{\theta}$. The whole procedure is fully automatic, the user has only to provide the shape and the pose. The initial mesh has exactly the same number of vertices, faces, and joints than the output mesh.

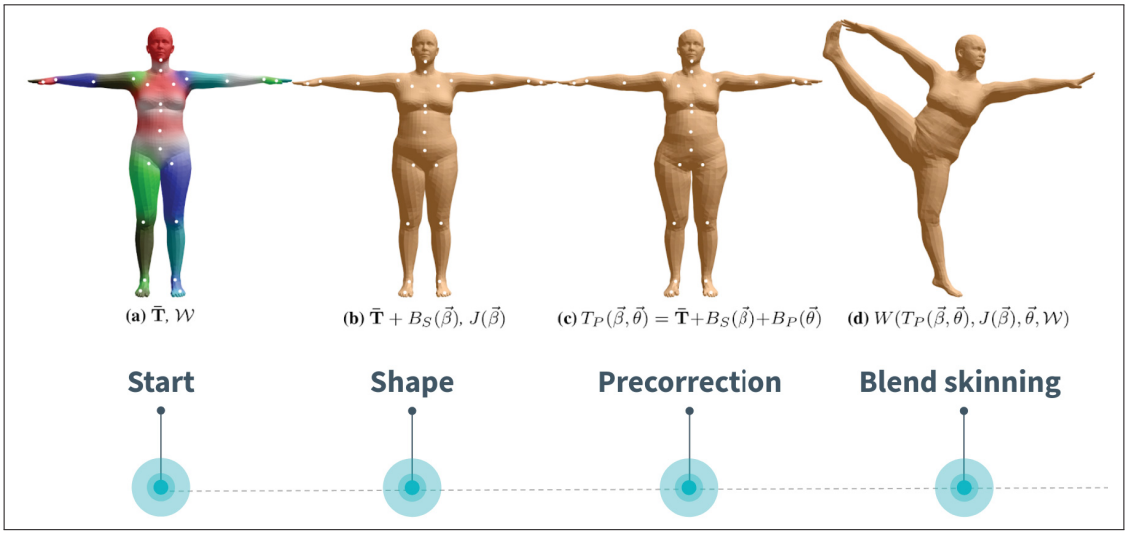


Figure 1.2 SMPL overview. Taken from Loper *et al.* (2015)

The first step consists in modifying the initial mesh so that it no longer models the mean human body shape but the shape specified with $\vec{\beta}$, still in the null pose. The idea is to translate the vertices of the initial mesh in order to change its shape to obtain the shape corresponding to $\vec{\beta}$. These translations will allow, for example, to enlarge the initial mesh, reduce the volume of its arms, etc. They are a function of $\vec{\beta}$. The translation vector to be applied to each vertex

is computed and concatenated into the vector $B_S(\vec{\beta}) \in \mathbb{R}^{3N}$. At the end of the first step, the resulting mesh is then $\bar{\mathbf{T}} + B_S(\vec{\beta})$ with

$$B_S(\vec{\beta}; \mathcal{S}) = \sum_{n=1}^{|\vec{\beta}|} \beta_n \mathbf{S}_n \tag{1.1}$$

where the matrix $\mathcal{S} = [\mathbf{S}_1, ..., \mathbf{S}_{|\vec{\beta}|}] \in \mathbb{R}^{3N \times |\vec{\beta}|}$ and the vector $\vec{\beta} = [\beta_1, ..., \beta_{|\vec{\beta}|}]^T \in \mathbb{R}^{|\vec{\beta}|}$. The translations to be performed are a linear combination of a translation base. This base $\mathcal{S}$ has been previously learned from a database of 3D scans by using a PCA and it is fixed in the model (Loper *et al.*, 2015). The number of selected principal components is $|\vec{\beta}| = 10$ and it is up to the user to specify $\vec{\beta}$ which contains the coefficients of the linear combination. The PCA has been performed on 3D scans from the CAESAR database (Robinette, Blackwell, Daanen, Boehmer & Fleming, 2002). For the notation, the parameters after ";" are the model parameters that have been learnt and fixed during the learning phase. While the parameters before are those specified by the users when using the model. The learning phase is detailed in the next section.

Since the initial mesh is modified to change its shape to $\vec{\beta}$, the location of the joints must also be readjusted to be able to animate the mesh correctly. The position of the joints are computed from $\vec{\beta}$. The function $J(\vec{\beta}) : \mathbb{R}^{|\vec{\beta}|} \mapsto \mathbb{R}^{3K}$ is the function that determines these positions. The function is defined as follows:

$$J(\vec{\beta}; \mathcal{J}, \bar{\mathbf{T}}, \mathcal{S}) = \mathcal{J} \cdot (\bar{\mathbf{T}} + B_S(\vec{\beta}; \mathcal{S})) \tag{1.2}$$

with $\mathcal{J} \in \mathbb{R}^{3K \times 3N}$ a matrix which, when multiplied by the vector of the mesh vertices in the null pose, gives the correct location of its joints. This matrix $\mathcal{J}$ contains somehow the information of which vertices are important, and how to combine them, to compute the position of the joints.

The goal of the second step is to pre-correct errors due to blend skinning. If a standard technique of blend skinning (LBS or DQBS for example) is directly applied to the current mesh, $\bar{\mathbf{T}} + B_S(\vec{\beta})$, the result will not necessarily be satisfactory because of the defects of these techniques. SMPL's trick to overcome this problem is to modify the mesh before the blend skinning in order to

anticipate the errors due to blend skinning (Fig. 1.2c). In SMPL, this pre-correction is noted $B_P(\vec{\theta}) \in \mathbb{R}^{3N}$ and depends on the pose parameter $\vec{\theta}$ (specified by the user). The pose in SMPL is described by the rotation of each of the skeleton joints relative to its parent in the kinematic tree, with the null pose as reference. When all the rotations are null it is equivalent to the "T" pose $\vec{\theta}^*$. The rotations must be specified in axis-angle. There are $K = 23$ joints, hence $|\vec{\theta}| = 3 \times 23 + 3 = 72$ coefficients (3 for each joint and 3 for the global orientation of the whole body).

Just like B_S, B_P is a set of translations to modify the mesh (one translation per vertex). The model defines

$$B_P(\vec{\theta}; \mathcal{P}) = \sum_{n=1}^{9K} (R_n(\vec{\theta}) - R_n(\vec{\theta}^*))\mathbf{P}_n \tag{1.3}$$

with $R : \mathbb{R}^{|\vec{\theta}|} \mapsto \mathbb{R}^{9K}$ the function that maps $\vec{\theta}$ to the matrices (concatenated into a single vector) of the rotation to be performed for each joint to pose the mesh. These matrices are computed from the equation of the blend skinning technique used. $R_n(\vec{\theta})$ is the n-th term of $R(\vec{\theta})$. $\mathbf{P}_n \in \mathbb{R}^{3N}$ are vectors of vertex displacements and $\mathcal{P} = [\mathbf{P}_1, ..., \mathbf{P}_{9K}] \in \mathbb{R}^{3N \times 9K}$ is a matrix of all $9K$ translation vectors determined and fixed during the learning phase. $R(\vec{\theta}^*)$ is subtracted so that the pre-correction is null if $\vec{\theta}$ is the null pose because, in such case, $\bar{\mathbf{T}} + B_S(\vec{\beta})$ is the final output mesh. If $\vec{\theta}$ is different from the null pose, then the mesh resulting from this second step is $T_P(\vec{\beta}, \vec{\theta}) = \bar{\mathbf{T}} + B_S(\vec{\beta}) + B_P(\vec{\theta})$ where $B_P(\vec{\theta})$ is the pre-correction.

Now that the mesh is pre-corrected, the blend skinning technique to go from the "T" pose to $\vec{\theta}$ (Fig. 1.2d) can be applied. It can be LBS, DQBS or another one, as long as the parameters of the SMPL model have been learned using this technique. The final mesh is

$$M(\vec{\beta}, \vec{\theta}) = W(T_P(\vec{\beta}, \vec{\theta}), J(\vec{\beta}), \vec{\theta}, \mathcal{W}) \tag{1.4}$$

with

$$T_P(\vec{\beta}, \vec{\theta}) = \bar{\mathbf{T}} + B_S(\vec{\beta}) + B_P(\vec{\theta}) \tag{1.5}$$

where W corresponds to the blend skinning technique. It takes as input the mesh $T_P(\vec{\beta}, \vec{\theta})$, the position of its joints $J(\vec{\beta}) \in \mathbb{R}^{3K}$, the desired pose parameter $\vec{\theta}$, and the matrix of the blend skinning weights $\mathcal{W}$. From these parameters, by applying the blend skinning, the new position of the vertices (in the desired pose) is obtained.

Learning phase

The set of learned parameters of the model is $\phi = \{\bar{\mathbf{T}}, \mathcal{W}, \mathcal{S}, \mathcal{J}, \mathcal{P}\}$. SMPL distinguishes these parameters in two groups: those related to the shape $\{\bar{\mathbf{T}}, \mathcal{S}\}$ and those related to the pose $\{\mathcal{J}, \mathcal{W}, \mathcal{P}\}$.

The deformation basis $\mathcal{S}$ is obtained through a PCA as discussed before. The mesh of the mean human body shape $\bar{\mathbf{T}}$, which is the starting point as explained previously, is also an outcome of this PCA. In an nutshell, this PCA allows to extract the mesh of the mean human body shape, and to learn how to deform it to obtain any other shape $\vec{\beta}$ (through the shape-dependent deformation base $\mathcal{S}$).

For $\{\mathcal{J}, \mathcal{W}, \mathcal{P}\}$, the learning is also done from 3D scans. For each scan (3D point cloud) a mesh is built. To do so, they align their initial mesh $\bar{\mathbf{T}}$ with the scan using the method of Bogo, Romero, Loper & Black (2014). The learning is then done on these meshes. There are several meshes of the same individual in different poses. For each individual, starting from his mesh in the null pose, the last two steps of the model (pre-correction and blend skinning) are executed in order to obtain, with minimal errors, his mesh in the other poses. The goal is to find the value of the parameters $\{\mathcal{J}, \mathcal{W}, \mathcal{P}\}$ that minimizes the pose reconstruction error. This error is defined as the square of the euclidean distance between the vertices of the registered mesh and the one reconstructed with the SMPL model.

A distinction is made between men and women. There is a model ϕ_m for men, trained using only men 3D scans, and ϕ_f the equivalent for women.

The modeling of the body is central in 3D human body reconstruction. The generated mesh must be deformable to be able to put it in any pose while maintaining its realism. The SMPL model seems central because it has been used on many occasions. Compared to the other models mentioned (Allen *et al.*, 2003,0; Anguelov *et al.*, 2005; Hirshberg *et al.*, 2012) SMPL is more suitable since it models both the shape and the pose, takes into account the correlations between both, is vertex-based, and is linear. SMPL is the most accurate with lowest computation complexity (Loper *et al.*, 2015). Moreover, one of the main advantages in using SMPL, besides being realistic, is that Eq. 1.4 is fully differentiable with respect to pose and shape parameters. It means that we can easily optimize $\vec{\beta}$ and $\vec{\theta}$ given an objective function involving $M(\vec{\beta}, \vec{\theta})$. However, the main drawback with SMPL is that it can generate meshes in poses that are impossible for humans (Fig. 1.3). The rotations in the pose parameter $\vec{\theta}$ are not limited and there are no constraints in the mathematical model to limit these rotations. It is up to the user to make sure that the given pose $\vec{\theta}$ is really achievable for a human.

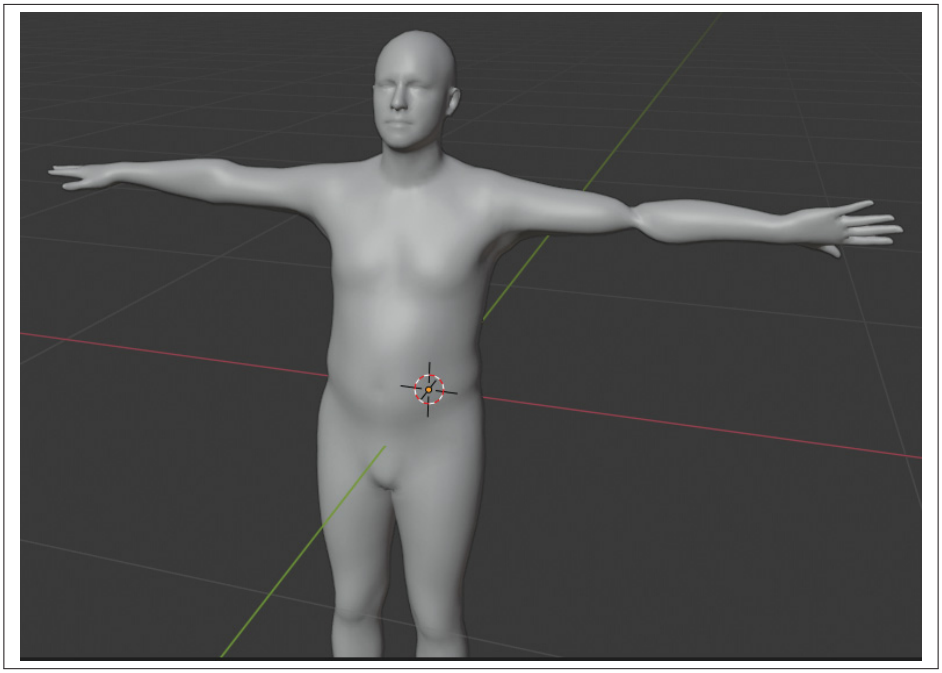


Figure 1.3 Example of SMPL mesh with an impossible pose

1.2 Human Pose Estimation

In the process of 3D body reconstruction from images, human pose estimation is systematically used (Bogo *et al.*, 2016; Huang *et al.*, 2017; Kolotouros *et al.*, 2019b; Madadi *et al.*, 2020; Omran *et al.*, 2018; Pavlakos *et al.*, 2018) to guide and drive the estimation of the body model parameters. Human pose estimation is defined as the problem of localizing the human joints (also known as keypoints) in images. It can be separated into two categories: 2D pose estimation and 3D pose estimation. 2D pose estimation consists in estimating a 2D (x,y) coordinates for each joint. Each 2D coordinates corresponds to a pixel of the image. Conversely, 3D pose estimation consists in estimating a 3D (x,y,z) coordinates for each joint. Each 3D coordinate corresponds to a location in a 3D space. In both cases, the problem is difficult due to joint occlusions, small and barely visible joints, clothing, and because several persons can be present in the images.

In the context of 3D body reconstruction, we are interested in the 3D pose. However, since 3D pose estimation methods are built on top, or from, 2D pose estimation methods, we first review 2D pose estimation in this section. The building blocs and the key ideas of human pose estimation as a domain lie in 2D pose estimation. We then review how these principles are extended for 3D pose estimation. In both cases, we focus on the most recent efforts relevant to our work.

1.2.1 2D pose estimation

2D pose (as well as 3D pose) estimation process can be broken down into two stages (Liu *et al.*, 2015): a pre-processing stage, followed by a stage of anatomical parts detection. Figure 1.4 illustrates the full and detailed process. The steps framed with dotted lines are those that are not always present in the methods. The pre-processing stage includes camera calibration, feature extraction from images, and human bodies detection. The remaining steps correspond to the anatomical parts detection stage.

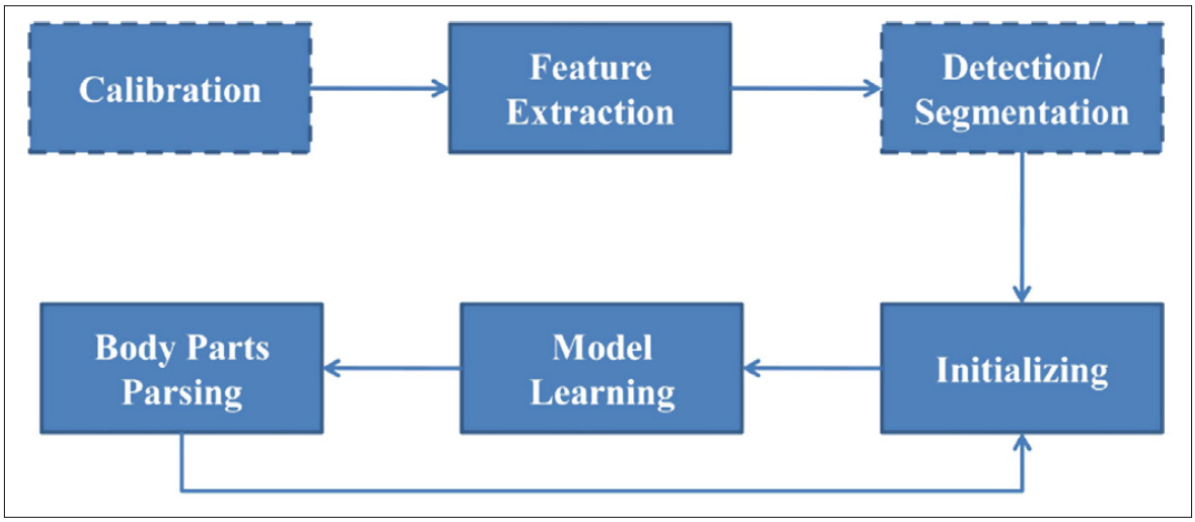


Figure 1.4 Human pose estimation pipeline. Taken from Liu *et al.* (2015)

Successful methods for 2D pose estimation make use of deep neural networks for these stages (Cao *et al.*, 2021; Fang, Xie, Tai & Lu, 2017; Newell, Yang & Deng, 2016; Pishchulin, Insafutdinov, Tang, Andres, Andriluka, Gehler & Schiele, 2016; Wei, Ramakrishna, Kanade & Sheikh, 2016). Each method adopts a different network architecture, leverages different data for the learning, and uses various strategies to learn better and faster. One of the most recent and accurate (Cao *et al.*, 2021), OpenPose, is reviewed in detail below since it is further used in our approach.

OpenPose

OpenPose (Cao *et al.*, 2021) is a tool and a method for detecting the 2D pose of all the persons present in an image. It uses two convolutional neural networks (CNN): a first one to obtain part affinity fields (PAFs), and a second one to compute the joint locations from these PAFs. The PAFs are 2D vector fields that indicate where limbs of the human body are on the image (Fig. 1.5). The output of the first network serves as an input to the second (Fig. 1.6). Below we describe how OpenPose works at inference time.

First, the input image of dimensions $w \times h$ is passed through an external neural network, composed of the first 10 layers of VGG-19 (Simonyan & Zisserman, 2015), for features extraction. It

Figure 1.5 **Top:** 2D poses. **Bottom:** PAF for the right forearm. Taken from Cao *et al.* (2021)

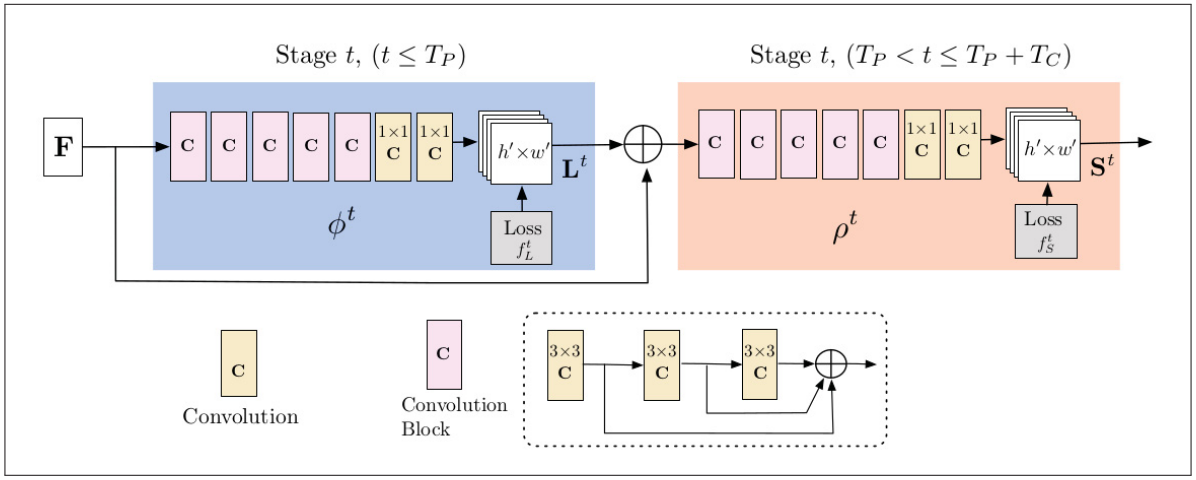


Figure 1.6 OpenPose's CNNs architecture. Taken from Cao *et al.* (2021)

generates a set of features $\mathbf{F}$ and these features are the input for the first CNN of OpenPose. This first CNN computes the set of PAFs $\mathbf{L} = (\mathbf{L}_1, \mathbf{L}_2, ..., \mathbf{L}_C)$ with $\mathbf{L}_c \in \mathbb{R}^{w \times h \times 2}$ $(c \in \{1, ..., C\})$ a PAF and C the number of limbs. There is one PAF per limb. A PAF (Fig. 1.5) is a 2D grid of dimensions $w \times h$ where each cell corresponds to a pixel of the input image and where a 2D vector is computed. For a given limb c, this vector is null if the pixel is not a pixel of the limb c in the image. Otherwise, it is a vector oriented in the direction of the limb. A limb in OpenPose is defined as a pair of joints, although not all pairs of joints define a limb of the human body anatomically speaking. For example, if the limb c is the right forearm, the vectors of $\mathbf{L}_c$ will be null for the pixels that do not belong to any right forearm on the image. On the other hand, if a pixel belongs to a right forearm, the associated vector will be oriented in the forearm direction (from elbow to wrist). $\mathbf{L}$ is computed iteratively (several passes through the network). Each iteration refines the estimation. In total, T_P iterations are made (Fig 1.6). The last estimate of $\mathbf{L}$ (after the T_P iterations) is named $\mathbf{L}^{T_P}$ and is sent to the second CNN of OpenPose together with $\mathbf{F}$.

The second CNN is in charge of computing the set $\mathbf{S} = (\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_J)$ where $\mathbf{S}_j \in \mathbb{R}^{w \times h}$ is associated to a particular joint $j \in \{1, ..., J\}$ with J the number of joints defined by the model. Each $\mathbf{S}_j$ is a 2D grid of dimensions $w \times h$ where each cell corresponds to a pixel of the input image. In each cell is computed the probability that the corresponding pixel belongs to the pixels of the joint j (Fig. 1.7). For example, if j is the right shoulder, $\mathbf{S}_j$ contains the probability for each pixel to be a pixel of a right shoulder on the image. This is also known as a heatmap and is common to many 2D pose estimation methods. The network architecture and the way used to compute these heatmaps vary across the methods. In OpenPose, only one pixel per joint has the probability 1 (maximum probability). It can be seen as the center of the joint. The coordinates of this pixel correspond to the 2D location of the joint. The other pixels of the joint have a high but not the maximum probability. The closer to the pixel having the maximum probability, the higher their probability is. $\mathbf{S}$ is also estimated iteratively: T_C passes through the second CNN are made. Each iteration refines the estimation.

Figure 1.7 Left elbow and left shoulder heatmaps. Taken from Cao *et al.* (2021)

A final step consists in using the information of **S** and **L** to compute the 2D skeleton of each person on the image. This means determining which joints of **S** belong to the same person with the help of **L** (PAFs). This problem is known as the matching problem and is NP-Hard (West, 2001). OpenPose uses a greedy approach to solve it. In our work, one of the constraints is that only one person (the subject) should be on the images, therefore the solution to the matching problem is obvious and straightforward. OpenPose supports various models of skeleton. We refer the reader to the OpenPose paper for a full description of the tool and method.

To conclude with 2D pose estimation, OpenPose as well as the other methods (Cao *et al.*, 2021; Fang *et al.*, 2017; Newell *et al.*, 2016; Pishchulin *et al.*, 2016; Wei *et al.*, 2016) have well-known issues such as sometimes confusing left and right sides, besides having troubles with occluded joints and complex poses (Huang *et al.*, 2017). However, their effectiveness is not questionable and they can reach incredible levels of accuracy on some benchmarks (Cao *et al.*, 2021).

1.2.2 3D pose estimation

There is a large body of research about 3D human pose estimation. Unlike 2D pose estimation, multi-view images can be used in 3D pose estimation. Two paradigms stand out in the literature:

the direct regression of 3D joints from images or the estimation of 2D joints followed by their lifting to 3D.

Direct 3D joint regression is mostly achieved by training a CNN (Joo, Liu, Tan, Gui, Nabbe, Matthews, Kanade, Nobuhara & Sheikh, 2015; Kocabas, Karagoz & Akbas, 2019; Pavlakos, Zhou, Derpanis & Daniilidis, 2017a,1; Rhodin, Meyer, Sporri, Muller, Constantin, Fua, Katircioglu & Salzmann, 2018; Sun, Shang, Liang & Wei, 2017) leveraging data sets like Human3.6M (Ionescu, Li & Sminchisescu, 2011; Ionescu, Papava, Olaru & Sminchisescu, 2014) and HumanEva (Sigal, Balan & Black, 2010). The concept of heatmaps discussed with 2D pose estimators is replaced by volumetric heatmaps for the 3D pose. It is the same idea but in a 3D volume instead of a 2D grid.

In two-stage methods, 2D pose estimation (Cao *et al.*, 2021; Pishchulin *et al.*, 2016; Wei *et al.*, 2016) is first performed, after which 2D estimates are lifted to 3D. To this end, various strategies have been applied, such as: exploit a dictionary of learned 3D poses (Sanzari, Ntouskos & Pirri, 2016; Tung, Harley, Seto & Fragkiadaki, 2017), take advantage of pictorial structure models (Belagiannis, Amin, Andriluka, Schiele, Navab & Ilic, 2014), triangulate 3D locations from 2D positions (Iskakov, Burkov, Lempitsky & Malkov, 2019), and develop a 3D-aware 2D pose estimator using transformers (He, Yan, Fragkiadaki & Yu, 2020).

Direct regression methods have the advantage of being faster than two-stage methods since they do not rely on 2D pose estimators (Bartol, Bojanić, Petković, D'Apuzzo & Pribanic, 2020; Desmarais, Mottet, Slangen & Montesinos, 2020). However, two-stage methods tend to be more accurate and robust (Bartol *et al.*, 2020; Desmarais *et al.*, 2020) than direct regression methods because they rely on 2D pose estimators which have been extensively studied and improved over the years. Indeed, 3D pose estimation is more recent than 2D pose estimation (Bartol *et al.*, 2020; Desmarais *et al.*, 2020) and has been able to raise mainly thanks to the huge advances in 2D pose estimation. The building blocks of the human pose estimation domain comes from the 2D pose estimation. Surprisingly, what makes two-stage methods competitive compared to direct regression methods is also their weakness. They suffer from the flaws of 2D pose

estimators mentioned in the previous section. Their accuracy heavily rely on the accuracy of the 2D joints.

1.3 3D Human Body Reconstruction from Images

Now that 3D human body modeling and human pose estimation have been discussed, we review in this section how current solutions use them and combine them to generate a subject's full-body 3D mesh from images. As stated before, SMPL (Loper *et al.*, 2015) is the leading body model used for this task. The recent and main methods can be classified as either CNN-based (Kanazawa *et al.*, 2018; Kolotouros *et al.*, 2019b; Madadi *et al.*, 2020; Omran *et al.*, 2018; Pavlakos *et al.*, 2018) or optimization-based (Bogo *et al.*, 2016; Huang *et al.*, 2017).

1.3.1 CNN-based methods

The CNN-based methods use only one view to estimate the SMPL pose and shape parameters. The idea is to use intermediate representations of the image to infer the shape and the pose. Each method uses its own intermediate representations. The main issue with CNN-based methods is that end-to-end training data is not widely available. There are very few images annotated with ground-truth SMPL pose and shape or a ground-truth 3D mesh. The only available data set, UP-3D (Lassner, Romero, Kiefel, Bogo, Black & Gehler, 2017), contains only 5703 single-view images for the training. Each image comes with its ground-truth SMPL pose and shape parameters. However, these ground-truths have been generated by running an optimization-based method presented in the next section (Bogo *et al.*, 2016). Therefore, their quality is questionable and one could argue that the accuracy of a CNN trained on these data is theoretically bounded by the accuracy of the optimization-based method. Each CNN-based method manages this lack of end-to-end training data differently. Below are reviewed, in chronological order, some key efforts and CNN architectures that have been proposed in this direction.

Human Mesh Recovery

Kanazawa *et al.* (2018) proposed a generative adversarial network (GAN) (Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville & Bengio, 2014) to solve the problem. Figure 1.8 summarizes their architecture. It consists in an encoder ResNet-50 (He, Zhang, Ren & Sun, 2016), a regression module, and a discriminator. The idea is to extract features from the image (with the encoder) and to feed the regression network with them to estimate the SMPL parameters and the camera parameters. The mesh corresponding to the estimated parameters is then generated, its 3D joints are projected on the image according to the estimated camera, and the error can be calculated. Since end-to-end training data is not widely available, 2D pose estimation data sets like Leeds Sports Pose (Johnson & Everingham, 2010) are used to assess the accuracy of the reconstruction. The error is therefore the L2 distance between the ground-truth 2D joints and the projected mesh joints on the image. Finally, the estimated SMPL parameters are sent to the discriminator. The discriminator is trained to distinguish the SMPL parameters that correspond to a realistic pose and shape. To learn that, they leverage the CAESAR database (Robinette *et al.*, 2002) together with the CMU MoCap database (University, 2007). The output of this discriminator is the probability that the SMPL parameters are not valid. They are not valid if they do not correspond to a possible shape and pose for the human body.

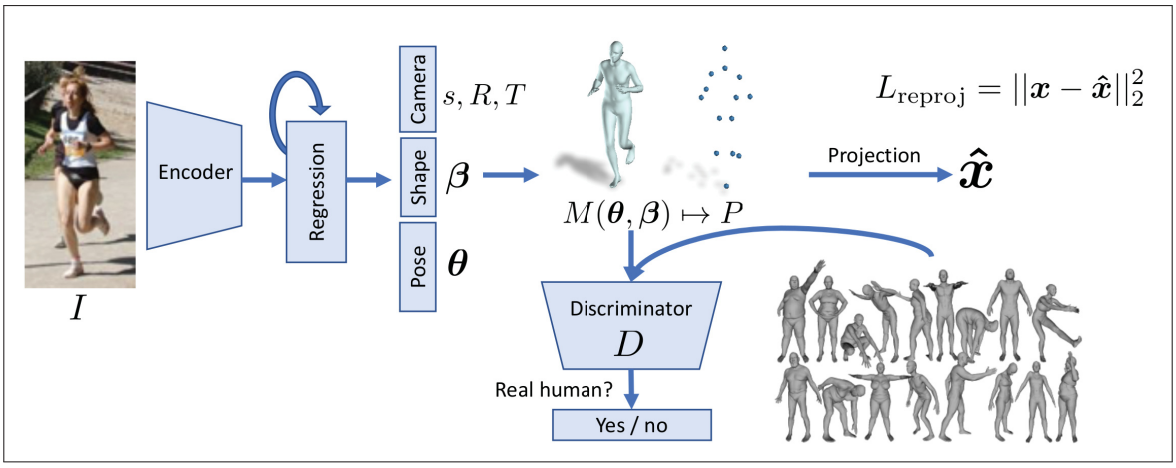


Figure 1.8 Human Mesh Recovery overview. Taken from Kanazawa *et al.* (2018)

Pavlakos *et al.* (2018)

Whereas Kanazawa *et al.* (2018) use image features extracted with ResNet-50 as intermediate representation to estimate the parameters, Pavlakos *et al.* (2018) proposed another alternative (Fig. 1.9). The first step of their method consists in estimating the 2D pose (in the form of heatmaps) and the silhouette (2D binary mask) of the individual from the image. They built and trained their own CNN for this rather than using existing models because they wanted one network to extract both at the same time. The second step is the estimation of $\vec{\theta}$ from the estimated 2D pose, and the estimation of $\vec{\beta}$ from the estimated silhouette. The PosePrior network, composed of two bi-linear units, is in charge of estimating $\vec{\theta}$. The ShapePrior network, in charge of estimating $\vec{\beta}$, has a simple architecture with five layers of 3×3 convolutions, each of them followed by a max-pooling and a bi-linear unit.

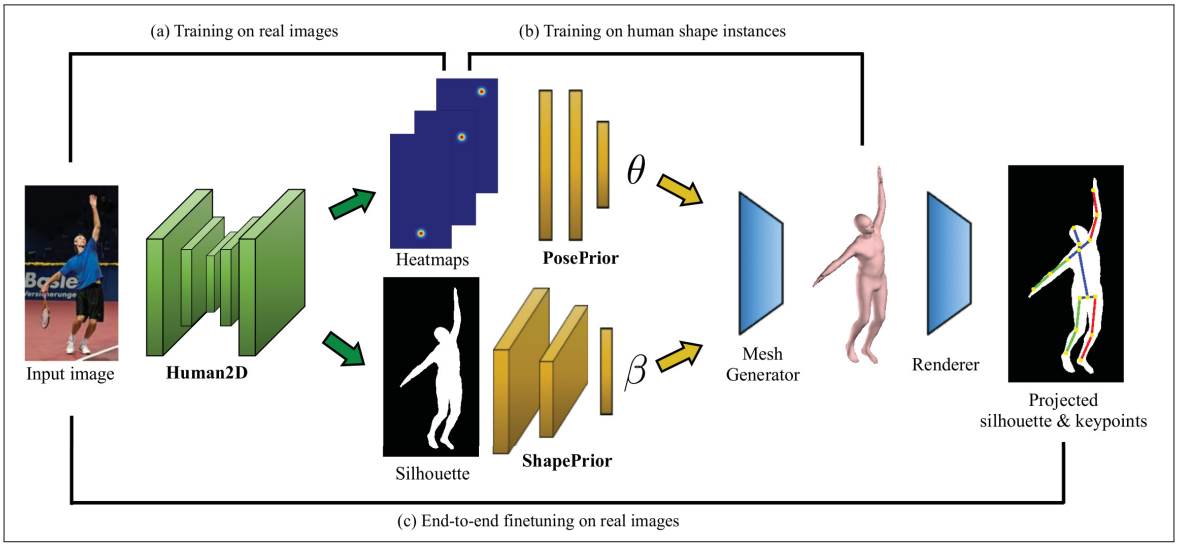


Figure 1.9 Overview of Pavlakos *et al.* (2018). Taken from Pavlakos *et al.* (2018)

To account for the lack of end-to-end training data, Pavlakos *et al.* (2018) generated SMPL meshes with pose and shape parameters that correspond to real people. Then, they rendered pictures of these meshes and from these pictures they extracted the silhouette and the 2D pose. Thus, they are in possession of silhouettes and 2D poses corresponding to SMPL meshes and

they know the corresponding SMPL parameters. This forms their own data set and ground-truths allowing to train the full pipeline.

Neural Body Fitting

Omran *et al.* (2018) use a body segmentation of the individual as an intermediate representation to compute the SMPL parameters (Fig. 1.10). First, a CNN having the same architecture as RefineNet (Lin, Milan, Shen & Reid, 2017) is used to segment the body of the person. The result of this segmentation is the input image but with different color masks, one color mask per limb. Each limb is colored in a different color on the segmented image. Then, from this segmented image, another CNN, based on the architecture of ResNet-50 (He *et al.*, 2016), is used to estimate the SMPL and camera parameters. The corresponding estimated mesh can then be generated and projected on the image to calculate the 2D joint error. Thus, like Kanazawa *et al.* (2018), they also use 2D pose estimation data sets for the training. However, unlike Kanazawa *et al.* (2018), they use the UP-3D data set (Lassner *et al.*, 2017) to supplement the learning. During the training, when an image comes from the UP-3D data set, the L2 distance between the estimated SMPL parameters and the ground-truth parameters is used as error metric to drive the learning (end-to-end training). Otherwise, the image comes from a 2D pose estimation data set and the 2D joint error metric is used.

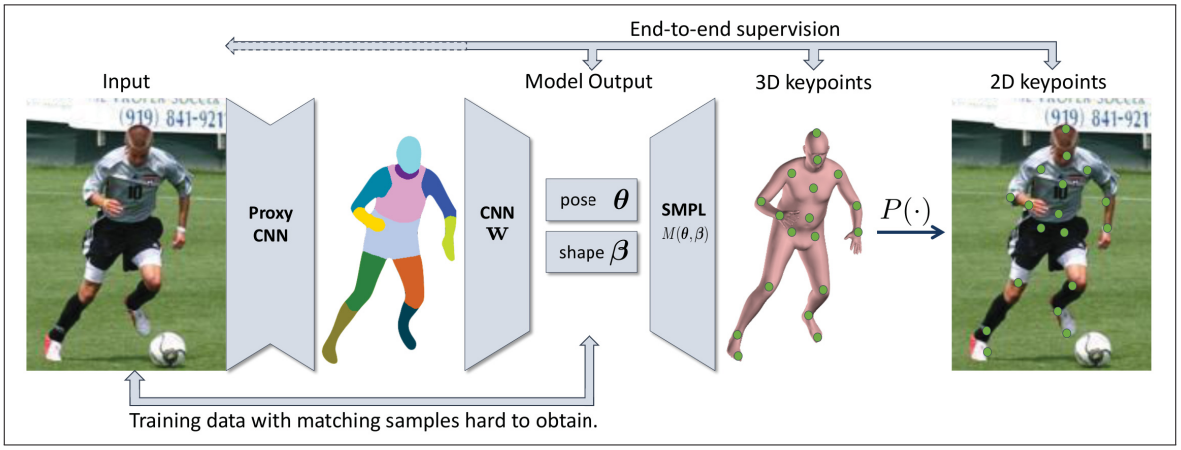


Figure 1.10 Neural Body Fitting overview. Taken from Omran *et al.* (2018)

Convolutional Mesh Regression

Kolotouros *et al.* (2019b) have a very different approach (Fig. 1.11) from the others. Indeed, they first estimate the complete SMPL mesh (the 3D position of each of the N vertices). Then, based on the mesh they have estimated, they regress the SMPL parameters. To achieve this, they use a Graph CNN (Litany, Bronstein, Bronstein & Makadia, 2018). They claim that this type of architecture allows to encode the structure of the SMPL mesh. In a way, this architecture can learn the SMPL mesh structure and be able to reproduce it. The process is as follows. The input image is encoded (features extraction with ResNet-50) and the features are attached to each vertex of the initial SMPL mesh $\bar{\mathbf{T}} \in \mathbb{R}^{3N}$ (average body mesh). Then, the vertices (together with the extracted features) are sent to the CNN graph which computes new coordinates for each vertex in order to deform the mesh. The output mesh is supposed to have the shape and the pose of the individual on the image. The CNN graph learns to deform the initial SMPL mesh to reach the shape and pose of the individual. It also estimates the camera parameters, allowing the projection of the mesh to compute the 2D joint error. Like Kanazawa *et al.* (2018), 2D pose estimation data sets are used for the training since end-to-end training data is not widely available.

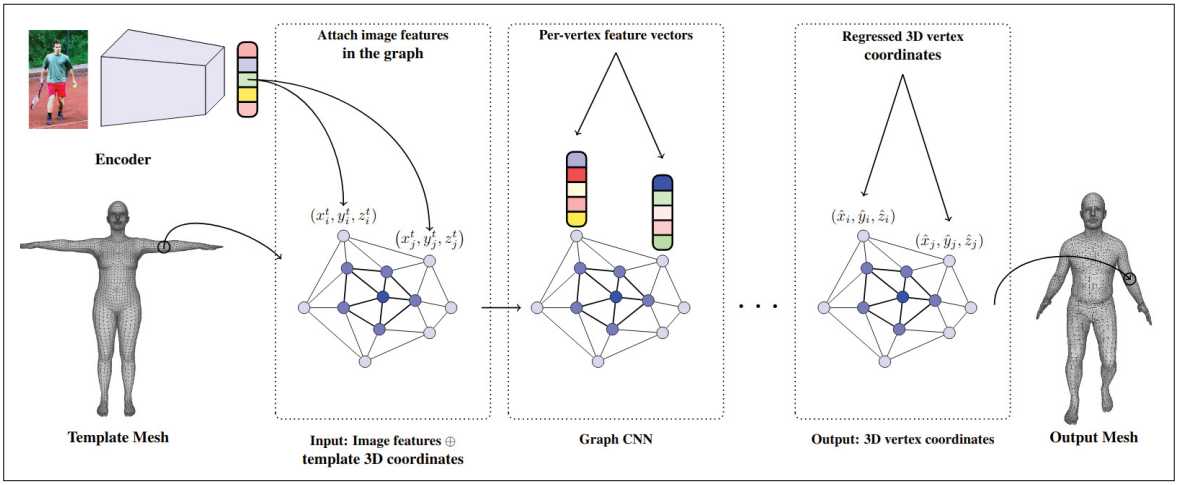


Figure 1.11 Convolutional Mesh Regression overview.
Taken from Kolotouros *et al.* (2019b)

Finally, the mesh estimated by the CNN graph is sent to a multi-layer perceptron whose purpose is to estimate $\vec{\beta}$ and $\vec{\theta}$ from the mesh. The idea is to find the SMPL parameters that best match this mesh.

The variation across the methods is the intermediate representation of the image used to estimate the SMPL parameters. Features extracted with ResNet-50 are used for Kanazawa *et al.* (2018) and Kolotouros *et al.* (2019b), while Omran *et al.* (2018) use body segmentation and Pavlakos *et al.* (2018) silhouette and 2D pose. The disadvantage with features extraction is that information is lost. Indeed, using only features of the image and not the complete image itself, some details possibly useful are set aside. CNN methods are generally less accurate and robust than optimization-based methods (Kolotouros, Pavlakos, Black & Daniilidis, 2019a; Pavlakos *et al.*, 2018) which follow. In addition, they do not generalize well either (Kolotouros *et al.*, 2019a), probably because they use only one view and because of the lack of end-to-end training data.

1.3.2 Optimization-based methods

Optimization-based solutions formalize the problem through one or several objective functions to be solved. The goal is to find the value of the SMPL parameters minimizing the objective functions. These functions require human-made priors and constraint terms to relax the objective function. Below, two key optimization-based methods are reviewed: SMPLify (Bogo *et al.*, 2016) and MuVS (Huang *et al.*, 2017).

SMPLify

The input for SMPlify is a single image (single-view method) and the output is the SMPL mesh. The problem is solved in two steps (Fig. 1.12). First, the 2D joints are estimated, J_{est}, using DeepCut (Pishchulin *et al.*, 2016). For each joint i, DeepCut also provides a confidence value, w_i, indicating the degree of confidence in the estimate. Then, the SMPL parameters are estimated from the knowledge of these 2D joints. For this, an objective function is defined. This function aims for minimizing the error between the position of the joints estimated by DeepCut

Figure 1.12 SMPLify overview. Taken from Huang *et al.* (2017)

and the projected SMPL joints. The energy function is:

$$E(\beta, \theta) = E_J(\beta, \theta; K, J_{est}) + \lambda_\theta E_\theta(\theta) + \lambda_\alpha E_\alpha(\theta) + \lambda_{sp} E_{sp}(\theta; \beta) + \lambda_\beta E_\beta(\beta) \tag{1.6}$$

where K corresponds to the camera parameters (for the projection of the SMPL 3D joints), and $\lambda_\theta, \lambda_\alpha, \lambda_{sp}, \lambda_\beta$ are scalar weights. The term $E_J(\beta, \theta; K, J_{est})$ corresponds to the error between the DeepCut joints J_{est} and the projected SMPL joints. Since the problem of inferring 3D from 2D is fundamentally ambiguous, constraint terms are added. While in CNN-based methods these constraints should be implicitly learnt by the network during the training, in optimization-based methods they have to be hand-designed. This is what is done in SMPLify with the term

$$E_\alpha(\theta) = \sum_{i \in \{elbows, knees\}} \exp(\theta_i) \tag{1.7}$$

which penalizes the poses where elbows and knees bend abnormally. In this term, the sum only concerns the elbow and knee joints. Furthermore, θ_i is the part of θ responsible for the rotation of the i-th joint. The exponential is used to penalize in a very strong way the poses where the knee and elbow rotations are too important. As a reminder, the null pose $\vec{\theta} = \vec{0}$ (i.e. $\theta_i = 0 \quad \forall i$) is the "T" pose. A pose where elbows and knees are not bent at all. The "negative" rotations in this context correspond to the natural knee and elbow rotations. Therefore, they are slightly penalized with the exponential. This is not the case for the "positive" rotations which are strongly penalized since they correspond to abnormal bending.

While $E_\alpha(\theta)$ penalizes the poses that are impossible for a human body, the pose prior $E_\theta(\theta)$ penalizes the poses that are least likely, although achievable by the human body. It is a statistical model trained on the CMU MoCap database (University, 2007) to learn the most likely poses. In the same vein, the shape prior

$$E_\beta(\beta) = \beta^T \Sigma_\beta^{-1} \beta \tag{1.8}$$

penalizes the least likely shapes. In this term, Σ_β^{-1} is the diagonal matrix of the squared singular values coming from the PCA of the SMPL model.

Finally, the term $E_{sp}(\theta;\beta)$ is used to penalize solutions containing interpenetration. For this, the SMPL mesh corresponding to the parameters β and θ is approximated with a set of capsules (Figure 1.13). Each limb of the mesh is approximated with a capsule, which facilitates the computation to determine whether there is interpenetration between the limbs or not. Indeed, making this calculation for a volume such as the human body is complex and requires a lot of computing time. On the other hand, it is very fast when dealing with convex objects such as these capsules. It is easier to check whether or not they intersect. Each capsule is defined by its height and radius. Thus, Bogo *et al.* (2016) have trained a model (Ridge regression with cross-validation) that automatically determines the radius and height of each capsule from the shape β. The term $E_{sp}(\theta;\beta)$ is the sum of the intersection volumes between the capsules that are not supposed to intersect. Indeed, we can see on the figure 1.13 that to have a good approximation of the body with the capsules, some capsules must necessarily intersect. It is important to mention that this term penalizes but does not prevent interpenetration.

The focal length of the camera with which the image was taken is assumed to be known. The optimization method used is Powell's Dogleg (Nocedal & Wright, 2006). SMPLify suffers from depth ambiguity issues since it relies on a single view. Human-made priors cannot recover all failure cases due to the depth ambiguity. This is why MuVS (Huang *et al.*, 2017) has been proposed.

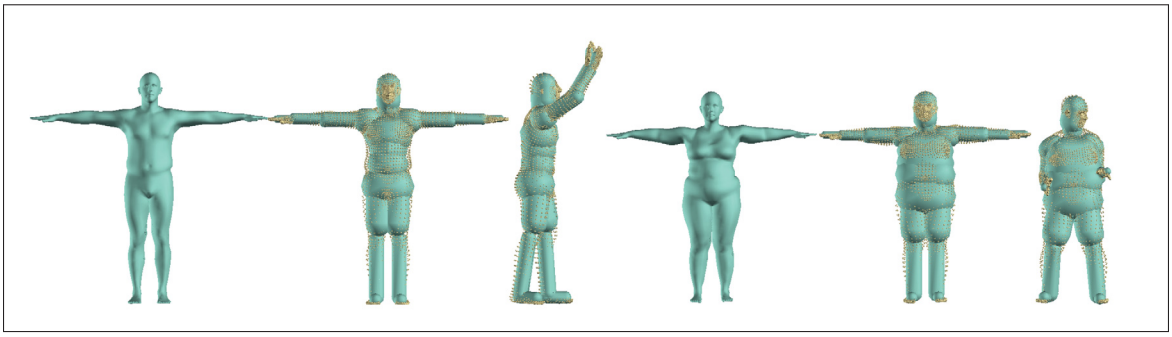


Figure 1.13 SMPL mesh approximation with capsules. Taken from Bogo *et al.* (2016)

MuVS

MuVS (Huang *et al.*, 2017) is the multi-view version of SMPLify. It is built upon SMPLify and uses a similar optimization process. The main difference is that the objective function considers 2D pose estimation through all views and also integrates 2D silhouettes. The function is defined as follows:

$$E(\beta, \theta) = \lambda_\theta E_\theta(\theta) + \lambda_\beta E_\beta(\beta) + \sum_{v=1}^{V} (E_J(\beta, \theta; K_v, J^v_{est}) + E_S(\beta, \theta; K_v, S_v)) \tag{1.9}$$

where λ_θ and λ_β are scalar weights, V is the total number of views, K_v corresponds to the camera parameters of the v-th view, J^v_{est} corresponds to the 2D joints estimated on view v, and S_v is the 2D silhouette (binary mask) estimated on view v. The pose prior E_θ, the shape prior E_β, and the 2D joint error term E_J are the same as in SMPLify. The silhouette error term E_S aims for silhouette consistency between the SMPL mesh and the individual. Huang *et al.* (2017) use the silhouette consistency term defined by Lassner *et al.* (2017). The more the two silhouettes differ on each view, the more the value of this term is high. K_v is needed in this term because the SMPL mesh silhouette is rendered according to each view v. Notice that unlike SMPLify, the terms E_{sp} and E_α are no longer in the objective function. According to Huang *et al.* (2017), they are no longer useful in a multi-view setting since the problem is more constrained. The depth ambiguity is significantly reduced thanks to the multiple views, hence abnormal bending and solutions with interpenetration are naturally avoided.

2D pose estimators have well-known issues like sometimes confusing left and right sides, besides having troubles with occluded joints and complex poses. For optimization-based methods, inaccurate 2D joint estimates lead to local minima of the energy function that can be too far from the global optimum result aimed for. MuVS (Huang *et al.*, 2017) tries to improve with respect to these shortcomings with the aid of the body's silhouette. Adding a silhouette consistency constraint between the images and the recovered shape allows to reduce the impact of incorrect 2D joint locations, and adds a direct constraint on the shape that was not previously available (SMPLify). However, even if silhouettes are well estimated, during the optimization process, the solver still makes a trade-off between optimizing the silhouette consistency term, the joint error term, and the other prior terms. Despite its deficiencies, the joint error term is crucial to the convergence of the optimization process since the silhouette alone does not constrain the location of joints within the silhouette. Another attempt of MuVS to mitigate joint inaccuracies is to use temporal smoothing when successive frames are available. A temporal smoothing stage with a specific objective function is introduced to constrain the acceleration of each 3D joint along the successive frames. The acceleration can be a manifestation of an error in the 2D estimates. Nevertheless, there is still a trade-off between the other terms in the objective function (to still constrain the SMPL parameters) and successive frames must be available.

To conclude on 3D human body reconstruction from images, CNN-based and optimization-based methods both have advantages and disadvantages. On one hand, CNN-based methods are faster at inference time but generally less accurate and robust than optimization-based methods. The 3D joint mean error on the Human3.6M database (Ionescu *et al.*, 2011,1) varies from 181 *mm* to 55 *mm* for these methods (Kanazawa *et al.*, 2018; Kolotouros *et al.*, 2019b; Madadi *et al.*, 2020; Omran *et al.*, 2018; Pavlakos *et al.*, 2018). This is explained by the lack of end-to-end training data and the use of only one view. On the other hand, optimization-based methods are slower, require hand-designed constraints, and heavily rely on the accuracy of the joint estimates, but are more robust and accurate. MuVS achieves a 3D joint mean error of 47 *mm* on Human3.6M. However, MuVS still suffers from the quality of the 2D joint estimates and optimizes shape and pose simultaneously, making the optimization harder and time consuming due to the complex

interleaving of shape and pose in SMPL. Moreover, optimization-based methods suffer from numerous local optima and only work well when initialized close to the real solution. We believe that there is a more natural road for improving optimization-based methods than CNN-based methods which require end-to-end training data.

CHAPTER 2

MULTI-VIEW 3D BODY RECONSTRUCTION

Given multi-view images of a human subject taken at the same time, together with camera parameters for each view, our goal is to generate a realistic and precise 3D body model of the subject as we saw in Figure 0.1. Like in other papers (Bogo *et al.*, 2016; Huang *et al.*, 2017; Kanazawa *et al.*, 2018; Kolotouros *et al.*, 2019b; Omran *et al.*, 2018; Pavlakos *et al.*, 2018), we use the SMPL body model (Loper *et al.*, 2015) to reach this goal. The challenge is to find out both the 3D shape and the 3D pose of the individual, from the images. As in other efforts (Huang *et al.*, 2017; Kanazawa *et al.*, 2018; Lassner *et al.*, 2017; Pavlakos *et al.*, 2018), we use the 2D pose and 2D silhouette estimations to infer accurate 3D pose and 3D shape. The proposed approach is summarized in Figure 2.1. While MuVS (Huang *et al.*, 2017) aggregates the 2D pose estimations from all the views into a single objective function, our optimization-based approach relies on two objective functions, both directly integrating 3D joint positions. We triangulate these 3D joints from 2D joint estimations by weighting the contribution of each view to the final 3D joint position. To determine the influence of each view, we rely on the 2D pose estimator's confidence values. This leads to better estimates for the 3D joints which are later injected in our shape and pose objective functions, reducing the undesirable local optima. Furthermore, we design a different optimization process with a novel objective function. This function aims to achieve bone orientation consistency between the 3D skeleton (triangulated joints) and the 3D body model. Thanks to our bone orientation constraint (BOC), we are able to closely approximate the pose parameter and take advantage of this information when conducting the final optimization stage (simultaneous shape and pose refinement). Finally, we demonstrate that the semantic position of joints in the body model and in the validation data sets do not exactly match. To account for this discrepancy we introduce, for each joint, a shift vector computed in the joint's local space. Results on widely used benchmark data sets (HumanEva and Human3.6M) show that our approach has a higher accuracy than the state-of-the-art methods.

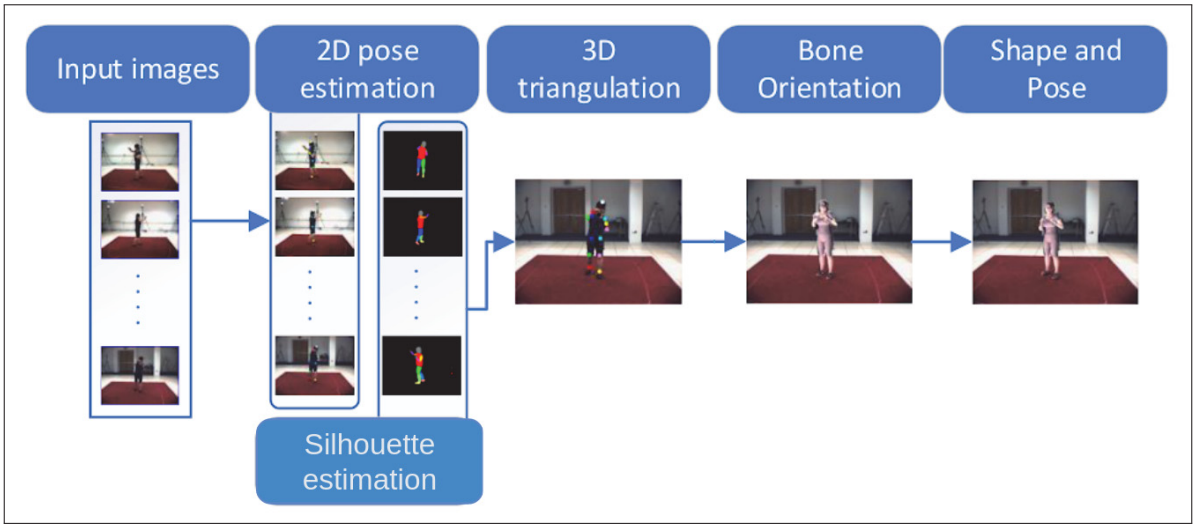


Figure 2.1 Overview of the proposed approach

The remainder of this chapter is organized as follows. Section 2.1 explains how 2D poses and body silhouettes are estimated. Then, the 3D pose triangulation is discussed in Section 2.2. Finally, our two-step optimization process is described in Section 2.3.

2.1 2D Pose and Body Silhouette

We first estimate the 2D pose and body silhouette on each view $v \in \{1, 2, 3, ..., V\}$, where V is the number of views. We use OpenPose (Cao *et al.*, 2021) to estimate the 2D pose on each image. For each view, OpenPose provides 25 joint locations and a confidence value for each joint j. The OpenPose skeleton contains 4 joints which are not part of the SMPL skeleton (left eye, right eye, left ear, and right ear). We only keep the 21 joints shared by the two skeletons.

Concerning the silhouette, Cross-Domain Complementary Learning (CDCL) (Lin, Wang, Luo, Chen, Liu & Sun, 2021) is used. For the view v, the corresponding silhouette image is denoted S_v, which consists in a binary image with pixels belonging to the silhouette having a value of 1. Our approach is versatile; other human body silhouette tools could be used instead.

2.2 3D Pose Triangulation

We use a linear algebraic triangulation (Iskakov *et al.*, 2019) combined with OpenPose's joint confidence values to lift the 2D joints to 3D. Given a joint j, its 2D estimated position on each view v, and the camera parameters (intrinsic and extrinsic) of each view, the algebraic triangulation consists in solving the following system of equations:

$$((\mathbf{w}_j \cdot \mathcal{J}) \circ A_j)\tilde{\mathbf{y}}_j = \mathbf{0} \tag{2.1}$$

where $\tilde{\mathbf{y}}_j$ is the unknown location of the 3D joint j and $A_j \in \mathbb{R}^{2V\times 4}$ is a matrix that allows to calculate, for all V views, the difference between the estimated 2D joint locations and the projected 3D joint locations. The weights $\mathbf{w}_j = (w_{1,j}, w_{1,j}, w_{2,j}, w_{2,j}, ..., w_{V,j}, w_{V,j})^\intercal \in \mathbb{R}^{2V\times 1}$ correspond to confidence values. The weight $w_{v,j} \in [0, 1]$ denotes the confidence in the estimate of joint j on view v. These weights are multiplied (matrix product) by the all-ones row vector $\mathcal{J} \in \mathbb{R}^4$ and operator $\circ$ denotes the Hadamard product. The idea behind the linear system of Eq. 2.1, is to recover the homogeneous coordinates, $\tilde{\mathbf{y}}_j \in \mathbb{R}^4$, of the joint j knowing its 2D projection on the V images. The 3D joint location $J_{3D_j} \in \mathbb{R}^3$ is then computed from the homogeneous coordinates. The system is solved independently for each joint using a differentiable singular value decomposition. We refer the reader to the original paper (Iskakov *et al.*, 2019) for full details. Iskakov *et al.* (2019) use their own 2D pose estimator (and confidence values) trained to map Human3.6M joints. We use OpenPose instead because its joints are at the same semantic positions as in the SMPL model.

The weights $\mathbf{w}_j$ are crucial since they adjust the contribution of each view in the triangulation. When a joint is likely to be occluded in one of the views, the weight for this view is low, ensuring that other views with larger confidences will drive the convergence to the right location. Thus, the emanating 3D joint contains less uncertainty than the 2D ones. For instance, in Figure 2.2, this allowed to correctly converge despite the inaccurate left elbow (pink) estimation in the view corresponding to the second row. Note the adjustment of the left elbow after the triangulation.

We automatically detect that this joint is likely to be inaccurate and thus decrease its contribution to the triangulation process, resulting in an accurate 3D joint location.

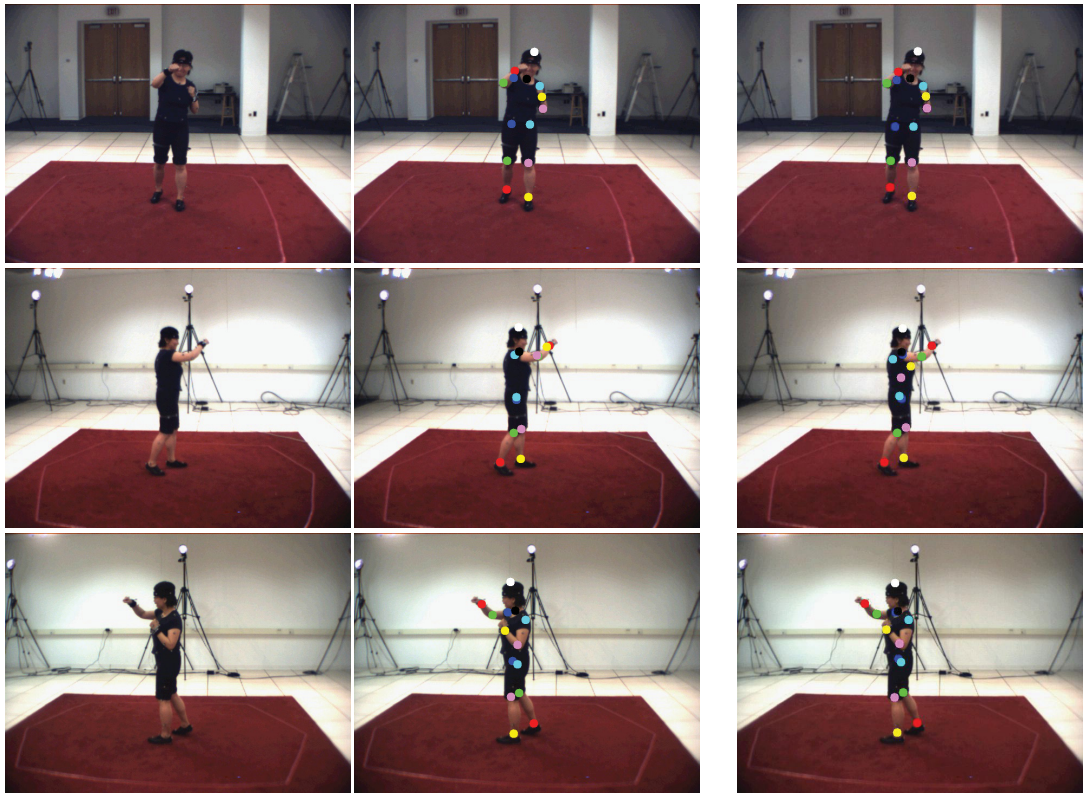


Figure 2.2 Example of joints correction thanks to the triangulation

2.3 Two-Step Optimization Process

We now describe our optimization process to infer SMPL parameters from the 3D joints and the silhouettes. SMPL does not constrain invalid pose and shape values. Therefore, given an energy function, one may converge to a *non-human* pose and/or shape if the problem is not constrained enough. Furthermore, SMPL joint locations after posing, $J(\vec{\beta}, \vec{\theta}) \in \mathbb{R}^{3\times 23}$, depend on SMPL

joint locations $J(\vec{\beta})$ which are a function of the shape. This means that each modification of the shape $\vec{\beta}$ necessarily leads to a change in $J(\vec{\beta}, \vec{\theta})$, even if the pose $\vec{\theta}$ remains unchanged.

In SMPLify (Bogo *et al.*, 2016) and MuVS (Huang *et al.*, 2017), both the shape and the pose are estimated simultaneously. As a consequence, the cost functions are complex and result in multiple local optima. That is the reason why optimization-based methods are sensitive to the initialization point.

In our approach, we overcome these obstacles in a novel fashion. The triangulated 3D joints allow us to provide a robust initialization for $\vec{\theta}$. The proposed optimization process is decomposed into two steps: SMPL mesh bone orientation, followed by simultaneous posing and shaping. First, $\vec{\beta}$ is initialized to the mean shape and $\vec{\theta}$ to an initial pose (Fig. 2.3a). The first step of the optimization process (bone orientation constraint) consists in estimating only the pose parameter (Fig. 2.3b). We want the 3D mesh, which is currently in the initial pose, to be posed as in the multiple views. To that end, we designed a new objective function. Let B be the set of bones of the triangulated 3D skeleton. A bone $b \in B$ is defined by two consecutive joints (child-parent) in the skeleton's kinematic tree. We name these joints child(b) and parent(b). Given a bone b and a 3D pose J (3D joint locations), the function $\Phi(J, b)$ returns the normalized orientation vector of the bone b in J:

$$\Phi(J, b) = \frac{J_{\text{child}(b)} - J_{\text{parent}(b)}}{||J_{\text{child}(b)} - J_{\text{parent}(b)}||_2} \tag{2.2}$$

Then, our objective function is:

$$E_{\text{pose}}(\vec{\theta}) = \lambda_\theta E_\theta(\vec{\theta}) + \lambda_{\text{bone}} \sum_{b \in B} ||\Phi(J(\vec{\beta}, \vec{\theta}), b) - \Phi(J_{3D}, b)||_2^2 \tag{2.3}$$

where J_{3D} denotes the triangulated 3D joints, $J(\vec{\beta}, \vec{\theta})$ denotes the SMPL mesh 3D joints, $\lambda_\theta = 1$ and $\lambda_{\text{bone}} = 100$ are weights, and $E_\theta(\vec{\theta})$ is the same pose prior (learned from the CMU data set) used by Huang *et al.* (2017) in MuVS and Bogo *et al.* (2016) in SMPLify. The pose prior role is to prevent convergence to *non-human* poses. During this optimization, $\vec{\beta}$ is kept fixed to the mean shape.

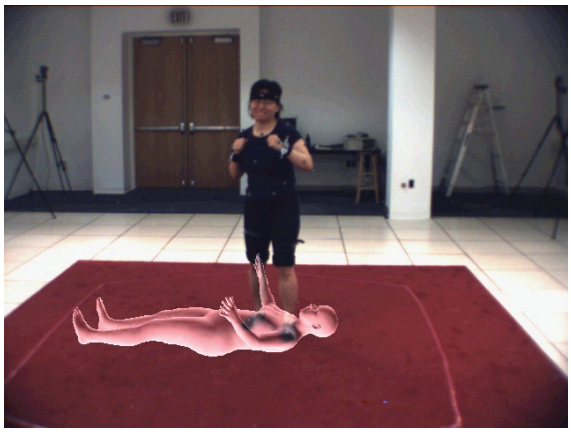
With Eq. 2.3, we are constraining the bones to have orientations consistent with the triangulated 3D joint positions. Normalizing the limb orientation vectors in the equation is crucial because the non-normalized vectors are a function of shape (limb lengths) besides pose. We are able to get a close approximation of $\vec{\theta}$ alone, without caring about the shape $\vec{\beta}$, because we get rid of the bone lengths by normalizing. Whatever the individual's shape, by minimizing Eq. 2.3, we are able to obtain accurate bone orientations (*e.g.* $\vec{\theta}$). This strategy resolves concerns arising from the simultaneous optimization of shape and pose used in previous papers (Bogo *et al.*, 2016; Huang *et al.*, 2017). One important advantage with our technique is that we can then use this estimation of $\vec{\theta}$ to initialize the final optimization step.

In the final step (Fig. 2.3c), we also want to recover the shape $\vec{\beta}$, therefore, the bone lengths matter. Our energy function is then:

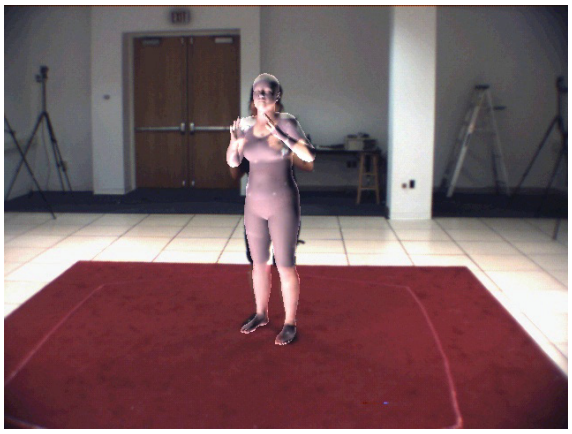
$$E_{\text{final}}(\vec{\beta}, \vec{\theta}, \vec{\gamma}) = \lambda_\theta E_\theta(\vec{\theta}) + \lambda_\beta E_\beta(\vec{\beta}) + \lambda_J ||J_{3D} - J(\vec{\beta}, \vec{\theta}) + \vec{\gamma}||_2^2 + \lambda_S \sum_{v=1}^{V} E_S(\vec{\beta}, \vec{\theta}; K_v, S_v) \quad (2.4)$$

where $\vec{\gamma} \in \mathbb{R}^3$ is the SMPL mesh global translation, $\lambda_\theta = 5,\ \lambda_\beta = 300,\ \lambda_J = 1,$ and $\lambda_S = \frac{1}{10}$ are weights, and $E_\beta(\vec{\beta})$ is the shape prior learnt from the SMPL body shape training data (Loper *et al.*, 2015). K_v corresponds to the camera parameters of the v-th view and $E_S(\vec{\beta}, \vec{\theta}; K_v, S_v)$ is the silhouette consistency term defined by Lassner *et al.* (2017). Contrary to MuVS, we use 3D rather than 2D joints data in this last stage. As a consequence, our objective function does not contain joint projection operations, and has fewer terms. In our case, the number of views comes up first when triangulating 3D joints. However, the triangulation is solved for each joint independently, and through a singular value decomposition, which is simpler than having more terms to deal with during the optimization.

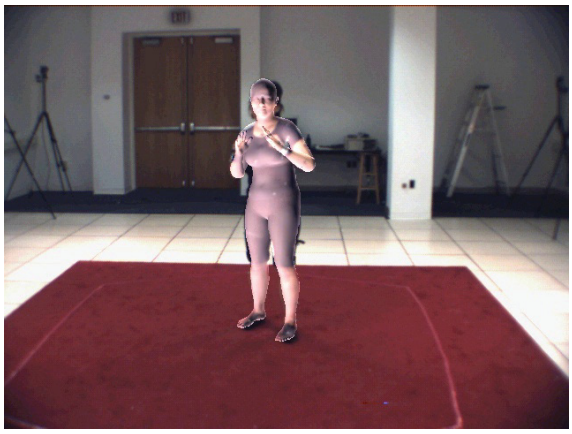
Like SMPLify and MuVS, we use the differentiable renderer OpenDR (Loper & Black, 2014) to optimize the objective functions (Eq. 2.3 and 2.4) using Powell's Dogleg method (Nocedal & Wright, 2006).



a) Initialisation



b) BOC



c) Simultaneous posing and shaping

Figure 2.3 Example using our two-step optimization process

CHAPTER 3

EXPERIMENTAL RESULTS AND DISCUSSION

We evaluate our approach on two widely used multi-view images data sets: HumanEva-I (Sigal *et al.*, 2010) and Human3.6M (Ionescu *et al.*, 2011,1). They both contain ground-truth 3D joint locations recovered from motion capture. Since HumanEva-I is a significantly smaller data set than Human3.6M, as other work (Bogo *et al.*, 2016; Huang *et al.*, 2017), we use HumanEva-I to make design choices and validate our approach, whereas Human3.6M serves to gauge the solution's generalization. We first present quantitative results followed by qualitative results. We measure the performance with the commonly used Mean Per Joint Position Error (MPJPE) metric and compare our approach to the state-of-the-art alternatives. Neither OpenPose nor CDCL training data sets overlap with our test sets. As is common practice (Bogo *et al.*, 2016; Loper *et al.*, 2015), we separate the models between male and female. We also manually fine tune all cost function weights on the training data set of HumanEva.

HumanEva-I and Human3.6M joints differ from the SMPL joints as showed in Figure 3.1 for Human3.6M. Despite the fact that the SMPL mesh and its silhouette (green contour on the left image) match the individual on the image, there is a shift between the ground-truth Human3.6M joints (blue squares) and the SMPL joints (green squares). To account for this discrepancy, we compute one shift vector (in the local bone coordinate) for each of the SMPL joints. Among the first 1000 frames of each video for Human3.6M, we took every 100th frame and computed the shift between the result of our optimization and the ground-truth. We then apply the mean of the shift vectors before computing the MPJPE for all of the other frames. For HumanEva-I we took every 20th frame among the first 300 frames. Shift vectors are only applied when computing the MPJPE for our approach since for the other methods (in Tables 3.1 and 3.2), the MPJPEs report the values found in the respective papers.

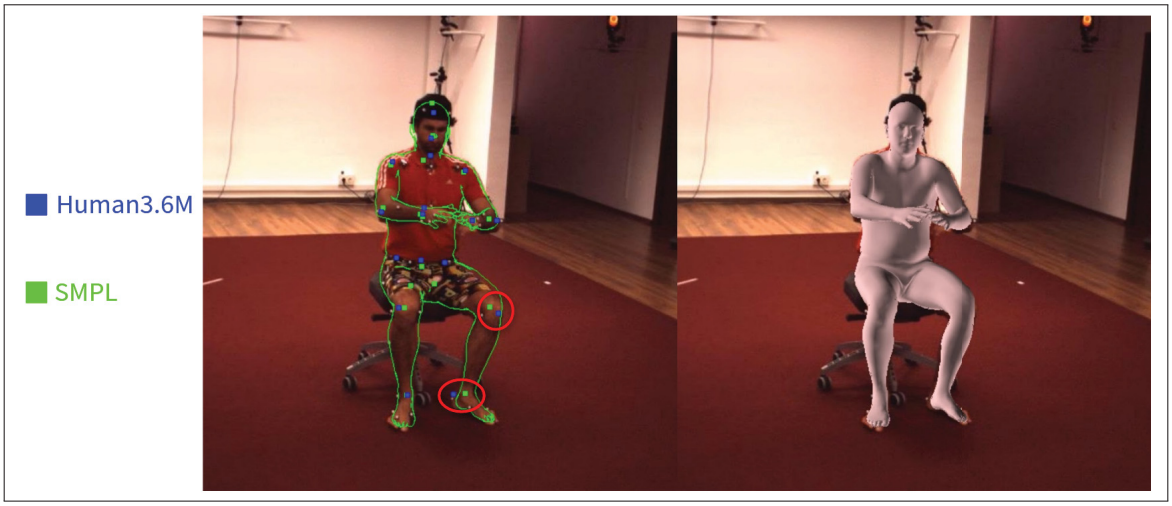


Figure 3.1 Shift between the ground-truth Human3.6M and the SMPL joints

3.1 Validation on HumanEva-I

We carry out a first validation on the HumanEva-I data set. On the order of 50, 000 images are available in HumanEva-I. There are six predefined actions, each performed by four subjects. Following a common practice (Bogo *et al.*, 2016; Huang *et al.*, 2017), we report results for subjects S1, S2, and S3 on the "Walking" and "Box" actions of the validation set. We use all three views and the ground-truth camera parameters.

Table 3.1 MPJPE (*mm*) comparison on HumanEva-I (the smaller the better)

Method	Walking	Box	Mean
MuVS	65.92	75.46	70.69
MuVS^{S}	58.32	68.41	63.37
MuVS$^{S,\,T}$	56.68	67.79	62.23
Ours	48.59	61.45	55.02
Ours^{SV}	47.22	59.88	53.55
Ours^{BOC}	42.63	53.75	48.19
Ours$^{BOC,\,SV}$	**41.96**	**51.12**	**46.54**
Ours$^{BOC,\,SV,\,S}$	42.13	53.03	47.58

Table 3.1 compares our approach with MuVS. The first row ("MuVS") refers to the MuVS optimization process using strictly 2D joint error terms as well as shape and pose priors (without using temporal information or silhouettes). Superscript S means adding the silhouette consistency term and superscript T the temporal information as described by Huang *et al.* (2017). "Ours" corresponds to our approach without the BOC step in the optimization process and without using the silhouettes. We notice that introducing 3D joints triangulated with OpenPose's confidence values ("Ours") significantly improves the MPJPE as compared to using 2D joints computed with Deepcut (Pishchulin *et al.*, 2016) (MuVS). We empirically observed that most of the time, Deepcut's confidence values are all around 99% which is not convenient to identify and reduce the weight of incorrectly detected joints. Moreover, OpenPose is generally more accurate than Deepcut. Adding the shift vectors when computing the MPJPE (superscript SV) for our approach leads to a slight improvement. "Ours^{BOC}" illustrates the effectiveness of our BOC in further decreasing the error. Unlike MuVS, incorporating the silhouette consistency term into our final energy function does not reduce further the MPJPE. Maybe because the joint errors left after the triangulation cannot be further decreased with the help of the silhouettes, compared to the joint errors left in MuVS (before the optimization). However, our approach outperforms MuVS without taking advantage of temporal nor silhouette information.

3.2 Generalization on Human3.6M

Human3.6M is a multi-view images data set composed of around 3.6 million images. Human3.6M poses are more challenging than HumanEva-I because of asymmetric and other complex poses. As in other studies (Huang *et al.*, 2017; Pavlakos *et al.*, 2017b; Trumble, Gilbert, Hilton & Collomosse, 2018), we use subjects 9 and 11 to evaluate our approach. We use all four views and the ground-truth camera parameters.

Table 3.2 compares our approach with other state-of-the-art methods (He *et al.*, 2020; Huang *et al.*, 2017; Iskakov *et al.*, 2019; Kanazawa *et al.*, 2018; Kolotouros *et al.*, 2019a; Pavlakos *et al.*, 2017b; Trumble *et al.*, 2018). In the table, "Shape" indicates if the method estimates the shape besides the pose, "PA" indicates if Procrustes analysis is applied before computing the

Table 3.2 MPJPE (*mm*) comparison on Human3.6M

Method	Shape	PA	MV	MPJPE
Kanazawa *et al.* (2018)	Yes	Yes	No	66.65
Trumble *et al.* (2018)	No	No	Yes	62.50
Kolotouros *et al.* (2019a)	Yes	Yes	No	62.00
Pavlakos *et al.* (2017b)	No	No	Yes	56.89
MuVS$^{S, T}$	Yes	Yes	Yes	47.09
Ours	Yes	Yes	Yes	54.86
Ours^{SV}	Yes	Yes	Yes	39.56
Ours^{BOC}	Yes	Yes	Yes	46.37
Ours$^{BOC, SV, S}$	Yes	Yes	Yes	33.07
Ours$^{BOC, SV}$	Yes	Yes	Yes	30.13
Iskakov *et al.* (2019)	No	Yes	Yes	20.80
He *et al.* (2020)	No	Yes	Yes	19.00

MPJPE, and "MV" states if the method uses multiple views. See Section 3.1 for the meaning of the subscripts. All the methods in the table are multi-view methods except Kanazawa *et al.* (2018) and Kolotouros *et al.* (2019a), which are single-view. We notice that almost all the multi-view methods perform better than the single-view ones, highlighting the fact that multiple views significantly improve the accuracy. Note that among the multi-view methods, only MuVS and our approach return a complete 3D human body mesh. The other methods optimize only for joint locations, which is a more constrained problem than simultaneously optimizing for shape and joint location. As such, Iskakov *et al.* (2019) and He *et al.* (2020) outperform our approach, as they do not optimize for a body mesh. Unlike these methods, we compute the parameters for a full data-driven body shape, which incurs a trade-off between the accuracy of the pose and the body shape. Even so, our approach outperforms most of the methods that compute only joint locations. Moreover, on Human3.6M our approach significantly outperforms the temporal version of MuVS. Finally, the shift vectors significantly reduce the MPJPE on Human3.6M (Table 3.2) contrary to HumanEva-I (Table 3.1).

3.3 Qualitative Evaluation

We evaluated the visual quality of the reconstructed 3D body mesh with our approach on Human3.6M. Figures 3.2 and 3.3 show some examples of the application of our approach to non-trivial poses and two body shapes (respectively subject 9 and 11). The green contour corresponds to the reconstructed body mesh silhouette and the mesh itself is in pink. We can see that our approach is effective and accurately recovers both the shape and the pose even in some challenging situations from Human3.6M.

3.4 Discussion

Using the same tools (Python2.7 and OpenDR) on the same machine, the execution time of MuVS when using temporal smoothness ($\text{MuVS}^{\text{S, T}}$) is around 13 minutes for each frame. For our approach without the silhouette consistency term ($\text{Ours}^{\text{BOC, SV}}$), the execution time is around 4 minutes per frame. Our approach is faster and more accurate than MuVS, and the use of 3D joints plays a role. The 3D joints simplify the objective functions since they allow to reduce the number of terms. The complexity of having multiple views for the joints estimation is moved to the triangulation in our case. We think this is good because the triangulation is solved with a learned differentiable singular value decomposition, as opposed to having more terms to deal with in the optimization process. The more terms there are in the objective functions, the more the solver struggles and must do a trade-off between all the terms. However, it is not the only reason why we are faster, we also have a different optimization process with a better initialization of the pose. We try to pull apart pose and shape in order to better estimate each one (divide and conquer). The motivation behind is that when interleaving the estimation of shape and pose, the objective function is more complex to solve because, given a 3D pose (3D joint locations) and shape, each modification of the shape parameter needs a re-computation of the pose parameter to keep the 3D joint locations unchanged.

We use the shift vectors because the semantic position of joints in the SMPL model and in the validation data sets do not exactly match. Our final joint estimates are the SMPL joints but

Figure 3.2 Qualitative results on Human3.6M subject 9

Figure 3.3 Qualitative results on Human3.6M subject 11

we compute the MPJPE relative to Human3.6M and HumanEva joints. Therefore, to account for this discrepancy, we have introduced the shift vectors. In Table 3.2, the methods (He *et al.*, 2020; Iskakov *et al.*, 2019; Pavlakos *et al.*, 2017b; Trumble *et al.*, 2018) that only compute joint locations do not use SMPL. They train neural networks to regress joint locations from images. These neural networks are trained on the Human3.6M training set, thus learning to regress Human3.6M joints. Consequently, applying shift vectors to them does not make sense. For the other methods (Huang *et al.*, 2017; Kanazawa *et al.*, 2018; Kolotouros *et al.*, 2019a) which, like us, compute shape and pose using SMPL, it would be interesting to assess the effect of the shift vectors. We did not conduct this comparison since official full implementations of these methods are not publicly available. This is why, for fairness comparison, we also reported the MPJPE for our approach without using the shift vectors. Even without using the shift vectors, our approach outperforms all methods (Huang *et al.*, 2017; Kanazawa *et al.*, 2018; Kolotouros *et al.*, 2019a) computing shape besides pose.

One drawback of the proposed approach is its dependency on OpenPose because we cannot rely on other joint detectors unless they comply with SMPL's joint positions. There is not a consensus for the joint definitions across the data sets and the pose estimators. Even if OpenPose and SMPL joint positions look similar, there is also a discrepancy between them. We do not measure this discrepancy. Although it is small, at the precision we achieve, it could still be significant. This could also partially explain why Iskakov *et al.* (2019) and He *et al.* (2020) perform better than our approach.

CONCLUSION AND RECOMMENDATIONS

In this dissertation, we have presented an approach to accurately estimate 3D human shape and pose in the multi-view setting. The literature review on human shape and pose estimation from images pointed out the usefulness of 3D human body models for this task. All of the reviewed methods use a parametric body model, most of the time the SMPL model (Loper *et al.*, 2015), and try to infer the model parameters that best fit the 3D shape and pose of the subject on the images. Two categories raised from the previous efforts in that direction: CNN-based methods and optimization-based methods. In both cases, intermediate clues such as 2D pose and 2D silhouette are used to estimate the model parameters (3D pose and 3D shape). CNN-based methods build and train deep neural networks to infer these parameters from the images using intermediate representations such as 2D pose, 2D silhouette, body segmentation, and so fourth. Although faster at inference time, the literature review allowed to conclude that these methods are usually less accurate and robust than optimization-based methods. Mainly because end-to-end training data is not widely available to train the neural networks. In contrast, optimization-based methods rely on the crafting of objective functions that are then optimized at inference time to compute the model parameters. These objective functions include 2D pose and 2D silhouette terms to drive the optimization. In this way, no training and no training data are required. However, the major disadvantages of optimization-based methods are their slowness, their sensibility regarding the initialisation, and the fact that they heavily rely on the quality of the pose and silhouette estimations. MuVS (Huang *et al.*, 2017) is one of these optimization-based methods and is the closest work to ours. While one of the most accurate method for human shape and pose estimation, MuVS is not exempt from these optimization-based method flaws. In this work, we have improved MuVS in several ways. First, we use 3D joints instead of 2D positions to infer the SMPL pose and shape parameters. We achieve this by triangulating 3D joint locations from 2D locations with a weighted algebraic triangulation. Second, we designed a new optimization process from the 3D joints to regress the SMPL parameters. This optimization

process incorporates a new bone orientation constraint (BOC) step which consists in solving a novel objective function to recover the SMPL pose parameter, independently from the shape. This allows us to decouple the pose parameter estimation from the final mutual shape and pose refinement. But also to accurately approximate the exact pose parameter and initialize the final optimization stage with it. All this leads to a significantly better (3D) joint estimation and therefore to a better final mesh. Finally, we demonstrated that the semantic position of joints in the SMPL model and in the validation data sets do not exactly match. To account for this discrepancy we introduced, for each joint, a shift vector computed in the joint's local space. Evaluation on widely used benchmarks demonstrated the effectiveness of our approach in comparison to the state-of-the-art methods.

Future work could investigate ways of reducing the execution time. The 2D pose and 2D silhouette estimations are very fast nowadays, OpenPose (Cao *et al.*, 2021) supports real-time pose estimation for example. The performance bottleneck in our approach is rather the optimization process. We solve the optimization problems using the differentiable renderer OpenDR (Loper & Black, 2014) which allows to derive the silhouette of the SMPL mesh, as well as its joint locations, according to the SMPL parameters. OpenDR is only available in Python and is quite slow. Improvements regarding the execution time could be achieved using another faster differentiable renderer such as the one recently introduced in TensorFlow 2.0, or by implementing the solution in C++. Other optimization methods than the Powell's Dogleg can also be investigated to speedup the solving.

Other than the execution time, an interesting future work is to look for other clues and intermediate representations than 2D pose and 2D silhouette to drive the inference of the 3D pose and 3D shape. Maybe there are other clues more relevant, or other intermediate representations which could supplement 2D pose and 2D silhouette. Find one allowing to infer the shape parameters alone (independently from the pose), as we succeed to do with the pose parameters, will

simplify the problem and its resolution following the divide and conquer principle. Indeed, when interleaving the estimation of shape and pose, the objective function is more complex to solve because, given a 3D pose (3D joint locations) and shape, each modification of the shape parameter needs a re-computation of the pose parameter to keep the 3D joint locations unchanged. The SMPL model is built such that given 3D joint locations to reach with the model's mesh, the pose parameter value to achieve so depends on the shape parameter value.

Moreover, there is a need for the development of a database allowing to assess the accuracy of the shape. As said before, end-to-end training data is not widely available. The only data set with such data, UP-3D (Lassner *et al.*, 2017), is a single-view data set where the ground truth has been obtained running the optimization-based method SMPLify (Bogo *et al.*, 2016). This is the reason why the majority of the reviewed papers do not assess the shape accuracy but only the 3D pose accuracy. The only few papers assessing the shape do it using the UP-3D data set or report an error about the silhouette. However, the silhouette is not the ideal data for measuring the accuracy of the 3D shape. It can be relevant when there is a large number of views, otherwise the silhouette error is not necessarily representative of the 3D shape error. The logical next step with our approach would be to evaluate the shape accuracy and compare it with the one of the other methods. It would be interesting to better evaluate the trade-off between joint accuracy and shape accuracy. Indeed, we have seen that we loose accuracy for the joints when adding the silhouette consistency term in the optimization, but we do not know quantitatively how much the shape improves or not.

Finally, we believe that our approach is extensible and that its accuracy could be improved by testing with future and more sophisticated body models. To better evaluate this accuracy, it is also worth investigating and propose a more general solution to discrepancies between joint definitions across the different body models, evaluation data sets, and pose estimators. We hope our shift vectors will inspire others in future work.

REFERENCES

Allen, B., Curless, B. & Popović, Z. (2003). The Space of Human Body Shapes: Reconstruction and Parameterization from Range Scans. *ACM Transactions on Graphics*, 22(3), 587–594.

Allen, B., Curless, B., Popović, Z. & Hertzmann, A. (2006). Learning a Correlated Model of Identity and Pose-dependent Body Shape Variation for Real-time Synthesis. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 147–156.

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J. & Davis, J. (2005). SCAPE: Shape Completion and Animation of People. *ACM Transactions on Graphics*, 24(3), 408–416.

Bartol, K., Bojanić, D., Petković, T., D'Apuzzo, N. & Pribanic, T. (2020). A Review of 3D Human Pose Estimation from 2D Images. *International Conference and Exhibition on 3D Body Scanning and Processing Technologies*, 12.

Belagiannis, V., Amin, S., Andriluka, M., Schiele, B., Navab, N. & Ilic, S. (2014, June). 3D Pictorial Structures for Multiple Human Pose Estimation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1669–1676.

Bogo, F., Romero, J., Loper, M. & Black, M. J. (2014, June). FAUST: Dataset and Evaluation for 3D Mesh Registration. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3794-3801.

Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J. & Black, M. J. (2016). Keep It SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. In Leibe, B., Matas, J., Sebe, N. & Welling, M. (Eds.), *European Conference on Computer Vision (ECCV)* (pp. 561–578). Cham: Springer International Publishing.

Cao, Z., Martinez, G. H., Simon, T., Wei, S. & Sheikh, Y. A. (2021). OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1), 172-186.

Chen, Y., Liu, Z. & Zhang, Z. (2013). Tensor-Based Human Body Modeling. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 105–112.

Desmarais, Y., Mottet, D., Slangen, P. & Montesinos, P. (2020). *A Review of 3D Human Pose Estimation Algorithms for Markerless Motion Capture*. Consulted at https://arxiv.org/abs/2010.06449?context=cs.

Fang, H.-S., Xie, S., Tai, Y.-W. & Lu, C. (2017). RMPE: Regional Multi-person Pose Estimation. *IEEE International Conference on Computer Vision (ICCV)*, pp. 2353-2362.

Freifeld, O. & Black, M. J. (2012). Lie Bodies: A Manifold Representation of 3D Human Shape. *European Conference on Computer Vision (ECCV)*, 1–14.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014). Generative Adversarial Nets. *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2, 2672–2680.

Hasler, N., Stoll, C., Sunkel, M., Rosenhahn, B. & Seidel, H.-P. (2009). A Statistical Model of Human Pose and Body Shape. *Computer Graphics Forum*, 28(2), 337-346.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778.

He, Y., Yan, R., Fragkiadaki, K. & Yu, S.-I. (2020). Epipolar Transformers. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7779–7788.

Hirshberg, D. A., Loper, M., Rachlin, E. & Black, M. J. (2012). Coregistration: Simultaneous Alignment and Modeling of Articulated 3D Shape. *European Conference on Computer Vision (ECCV)*, pp. 242–255.

Huang, Y., Bogo, F., Lassner, C., Kanazawa, A., Gehler, P. V., Romero, J., Akhter, I. & Black, M. J. (2017). Towards Accurate Marker-Less Human Shape and Pose Estimation over Time. *International Conference on 3D Vision (3DV)*, pp. 421-430.

Ionescu, C., Li, F. & Sminchisescu, C. (2011). Latent Structured Models for Human Pose Estimation. *IEEE International Conference on Computer Vision (ICCV)*, pp. 2220-2227.

Ionescu, C., Papava, D., Olaru, V. & Sminchisescu, C. (2014). Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7), 1325-1339.

Iskakov, K., Burkov, E., Lempitsky, V. & Malkov, Y. (2019, October). Learnable Triangulation of Human Pose. *IEEE International Conference on Computer Vision (ICCV)*, pp. 7717-7726.

Jacobson, A., Deng, Z., Kavan, L. & Lewis, J. (2014). Skinning: Real-time Shape Deformation. *ACM SIGGRAPH 2014 Courses*.

Johnson, S. & Everingham, M. (2010). Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 12.1–12.11.

Joo, H., Liu, H., Tan, L., Gui, L., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S. & Sheikh, Y. (2015, December). Panoptic Studio: A Massively Multiview System for Social Motion

Capture. *IEEE International Conference on Computer Vision (ICCV)*, pp. 3334–3342.

Kanazawa, A., Black, M. J., Jacobs, D. W. & Malik, J. (2018). End-to-end Recovery of Human Shape and Pose. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7122–7131.

Kavan, L., Collins, S., Žára, J. & O'Sullivan, C. (2007). Skinning with Dual Quaternions. *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, pp. 39–46.

Kocabas, M., Karagoz, S. & Akbas, E. (2019, June). Self-Supervised Learning of 3D Human Pose using Multi-view Geometry. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1077–1086.

Kolotouros, N., Pavlakos, G., Black, M. J. & Daniilidis, K. (2019a, October). Learning to Reconstruct 3D Human Pose and Shape via Model-fitting in the Loop. *IEEE International Conference on Computer Vision (ICCV)*, pp. 2252–2261.

Kolotouros, N., Pavlakos, G. & Daniilidis, K. (2019b). Convolutional Mesh Regression for Single-Image Human Shape Reconstruction. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4496-4505.

Lassner, C., Romero, J., Kiefel, M., Bogo, F., Black, M. J. & Gehler, P. V. (2017, July). Unite the People: Closing the Loop Between 3D and 2D Human Representations. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4704-4713.

Lewis, J., Cordner, M. & Fong, N. (2000). Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation. *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 165–172.

Lin, G., Milan, A., Shen, C. & Reid, I. (2017, July). RefineNet: Multi-path Refinement Networks for High-Resolution Semantic Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5168-5177.

Lin, K., Wang, L., Luo, K., Chen, Y., Liu, Z. & Sun, M.-T. (2021). Cross-Domain Complementary Learning Using Pose for Multi-Person Part Segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(3), 1066-1078.

Litany, O., Bronstein, M. A., Bronstein, M. M. & Makadia, A. (2018, June). Deformable Shape Completion with Graph Convolutional Autoencoders. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1886-1895.

Liu, Z., Zhu, J., Bu, J. & Chen, C. (2015). A survey of human pose estimation: The body parts parsing based methods. *Journal of Visual Communication and Image Representation*, 32,

10 - 19.

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G. & Black, M. J. (2015). SMPL: A Skinned Multi-Person Linear Model. *ACM Transactions on Graphics*, 34(6), 248:1–248:16.

Loper, M. M. & Black, M. J. (2014). OpenDR: An Approximate Differentiable Renderer. *European Conference on Computer Vision (ECCV)*, 8695, 154–169.

Madadi, M., Bertiche, H. & Escalera, S. (2020). SMPLR: Deep learning based SMPL reverse for 3D human pose and shape recovery. *Pattern Recognition*, 106, 107472.

Newell, A., Yang, K. & Deng, J. (2016). Stacked Hourglass Networks for Human Pose Estimation. *European Conference on Computer Vision (ECCV)*, pp. 483–499.

Nocedal, J. & Wright, S. J. (2006). *Numerical optimization* (ed. 2). Springer.

Omran, M., Lassner, C., Pons-Moll, G., Gehler, P. V. & Schiele, B. (2018). Neural Body Fitting: Unifying Deep Learning and Model-Based Human Pose and Shape Estimation. *International Conference on 3D Vision (3DV)*, pp. 484–494.

Pavlakos, G., Zhou, X., Derpanis, K. G. & Daniilidis, K. (2017a). Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1263-1272.

Pavlakos, G., Zhou, X., Derpanis, K. G. & Daniilidis, K. (2017b, July). Harvesting Multiple Views for Marker-Less 3D Human Pose Annotations. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1253-1262.

Pavlakos, G., Zhu, L., Zhou, X. & Daniilidis, K. (2018). Learning to Estimate 3D Human Pose and Shape from a Single Color Image. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 459–468.

Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P. & Schiele, B. (2016). DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4929-4937.

Rhodin, H., Meyer, F., Sporri, J., Muller, E., Constantin, V., Fua, P., Katircioglu, I. & Salzmann, M. (2018, June). Learning Monocular 3D Human Pose Estimation from Multi-view Images. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8437–8446.

Robinette, K., Blackwell, S., Daanen, H., Boehmer, M. & Fleming, S. (2002). Civilian American and European Surface Anthropometry Resource (CAESAR). 1, 74.

Sanzari, M., Ntouskos, V. & Pirri, F. (2016). Bayesian Image Based 3D Pose Estimation. *European Conference on Computer Vision (ECCV)*, pp. 566–582.

Sigal, L., Balan, A. O. & Black, M. J. (2010). HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion. *International Journal of Computer Vision*, 87(1), 4–27.

Simonyan, K. & Zisserman, A. (2015, May). Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICLR)*.

Sun, X., Shang, J., Liang, S. & Wei, Y. (2017, October). Compositional Human Pose Regression. *IEEE International Conference on Computer Vision (ICCV)*, pp. 2621-2630.

Trumble, M., Gilbert, A., Hilton, A. & Collomosse, J. (2018, September). Deep Autoencoder for Combined Human Pose Estimation and Body Model Upscaling. *European Conference on Computer Vision (ECCV)*, pp. 800–816.

Tung, H.-Y. F., Harley, A. W., Seto, W. & Fragkiadaki, K. (2017, October). Adversarial Inverse Graphics Networks: Learning 2D-to-3D Lifting and Image-to-Image Translation from Unpaired Supervision. *IEEE International Conference on Computer Vision (ICCV)*, pp. 4364–4372.

University, C. M. (2007). CMU MoCap [Online dataset]. Consulted at http://mocap.cs.cmu.edu/.

Wei, S.-E., Ramakrishna, V., Kanade, T. & Sheikh, Y. (2016). Convolutional Pose Machines. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724-4732.

West, D. B. (2001). *Introduction to graph theory* (ed. 2). Prentice hall Upper Saddle River.