3D Reconstruction of Human Body via Machine Learning

by

Qi He

B.S., Tsinghua University, China (2018)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2020

Author . . . .
Department of Mechanical Engineering
May 13, 2020

Certified by . . . .
Ju Li
Professor
Thesis Supervisor

Accepted by . . . .
Nicolas G. Hadjiconstantinou
Chairman, Committee on Graduate Students

3D Reconstruction of Human Body via Machine Learning

by

Qi He

Submitted to the Department of Mechanical Engineering
on May 13, 2020, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

Three-dimensional (3D) reconstruction and modeling of the human body and garments from images is a central open problem in computer vision, yet remains a challenge using machine learning techniques. We proposed a framework to generate the realistic 3D human from a single RGB image via machine learning. The framework is composed of an end-to-end 3D reconstruction neural net with a skinned multi-person linear model (SMPL) model by the generative adversarial networks (GANs). The 3D facial reconstruction used the morphable facial model by principal component analysis (PCA) and the LS3D-W database. The 3D garments are reconstructed by the multi-garment net (MGN) to generate UV-mapping and remapped into the human model with motion transferred by archive of motion capture as surface shapes (AMASS) dataset. The clothes simulated by the extended position based dynamics (XPBD) algorithm realized fast and realistic modeling.

Thesis Supervisor: Ju Li
Title: Professor

Acknowledgments

My graduate career has been completed with the support of many people.

I would like to first express my sincere gratitude to Professor Ju Li for his academic guidance and financial support during the past year. On the one hand, we collaborated to generate several profound ideas and results. On the other hand, during the years at MIT, the myriad discussions that I had with Prof. Li sharpened my research philosophy. It has been my highest honor to work with Prof. Li, and I am proud of having such an outstanding advisor at MIT.

I would like to thank Professor Xuanhe Zhao, for his invaluable suggestions and guidance throughout my research and the financial support for my first year at MIT. I would also like to thank Professor David Parks. He encouraged me to aspire for my research interests without fear and served as the thesis readers. A great thank to Dr. Yunwei Mao, who has provided all of the necessary support throughout my years to ensure that I could focus on the best research. Dr. Mao imparted me with great tools for researching in computer graphics and machine learning and taught me patiently on how to disentangle the various aspects of research. I would also like to thank Leslie Regan for her support. Your kindness and patience have made me what I am today.

I would like to thank my parents, Mr. Bin He and Mrs. Mo Chen, who have devoted their boundless love to me in every aspect, and work hard to support the family. Special thanks to my girlfriend Jianqiao Cui, who has shown me unwavering support throughout the year.

Thank you all for making it happen.

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

1.1 Motivation

Since the first-ever online sale happened in August 11, 1994, the booming of e-Commerce already caused the upheaval to society. Nowadays people prefer to shop online at Amazon, Bestbuy, eBay, etc. However, the most faced problem is that the customers do not know the size and quality of the garments in the shopping cart. Only several photos of the garments in the showcase could be misleading. Sometimes the received items are different when we see it in real life. Though some e-Commerce platforms display the clothes dressed by the real models, the garments can still be the wrong size for the customers. To alleviate the anxiety of customers, almost all top fashion e-Commerce provide the free return service for customers. The costly business of retail returns is a $62.4 billion 'ticking time bomb', according to a CNBC report [41] .

Generally, to enhance the e-Commerce shopping experience, this thesis attempts to answer the following questions:

1. How to generate the 3D avatars for the customers themselves?
2. How to generate the 3D garments to fit in the avatars?
3. How to do fast clothes simulation and photorealistic rendering?

This thesis is comprised of two major parts to deal with the questions mentioned above.

- 3D human reconstruction
    - Body reconstruction
    - Face reconstruction
- 3D garments reconstruction
    - Garments reconstruction
    - Cloth simulation
    - Rendering system

In the 3D human reconstruction part, the human body reconstruction was discussed in the first place. The general SMPL model [29] was introduced to represent different human body shapes. The rest pose, blend weights, and blend shape of the SMPL model was learned from thousands of labeled 3D scan human body data. Thereafter, an end-to-end reconstruction method from a single image was built with convolutional neural networks (CNNs) and generative adversarial networks (GANs). With the computer-generated parameters of pose and shape, the human body's avatar could be animated into different motions with the AMASS database [31].

Secondly, the face reconstruction method was introduced after the 3D human body reconstruction. Initially, we introduced the 3D facial landmarks detection method from a single face image. A morphable model for 3D faces was generated with principal component analysis (PCA). The prediction of parameters and textures of the reconstructed face was mapped with the help of 3D detection landmarks.

In the 3D garments reconstruction part, the computer-generated garments were built from the multi-garment net (MGN). It predicted the garment geometry from images and layered on the top of the SMPL model. The cloth simulation was conducted by the extended position-based dynamics (XPBD), which is an iterative method to solve complex contains based on Gauss's principle of least constraint. Afterward, the optical-tracing rendering was conducted by the Blender cycle engine.

This thesis provides a workflow to construct an online dressing system with the help of several general open-source toolboxes. The 3D human reconstruction module enables the users to reconstruct the computer-generated human body and face via a single full-shot portrait. The 3D garment reconstruction enables the users to dress in the 3D garments, which were generated from images. The physical engine and rendering system realized the fast, realistic cloth simulation and photorealistic rendering in virtual indoor and outdoor environments.

1.2 Background

1.2.1 Machine learning and GANs

Machine learning (ML) is an application of artificial intelligence (AI). It enables computers to automatically complete several complex tasks without explicit coding, such as face recognition, data mining, recommendation system, etc. Deep learning is one of the most popular and influential subfields in the machine learning. It attempts to intimate the human brain and neural networks to process the data and learning skills. The typical bio-inspired architecture of deep learning consists of the multiple layers built with artificial neural networks, made from hardware, e.g., GPU units, other than biological tissues. Currently, there are two popular frameworks in the deep learning field, TensorFlow (Google) and PyTorch (Facebook), both of which are adopted in this thesis.

Generative adversarial networks (GANs) is a recently developed machine learning framework proposed to creatively generate complex outputs, such as fake faces, speeches, and videos. It is comprised of two competing deep neuron networks, a generative network and a discriminate network [18]. GANs were adopted in this thesis to enhance the quality of models in end-to-end reconstruction of the human body.

1.2.2 3D human computer-generated imagery

The first 2D computer-generated imagery (CGI) was adopted in the movie in 1973's Westworld. The first usage of 3D computer-generated human hand and faces happened in its sequel, Futureworld (1973) according to Wikipedia. One of the most famous 3D human CGI film is the Avatar (2009) by American director, James Cameron.

The traditional way in the film industry involves tremendous hand rigging of mesh and manually sculpting [29]. The great manual effort in the generation of realistic 3D human models was made to correct the problems of models.

The traditional ways to create a new realistic human model could be summarized here. Initially, the real human was scanned with multiple RBG-D (depth) cameras in different angles and distances. The images were combined with the help of computer vision/graphics techniques. After that, the 3D human model was created manually by artists from a large database. After generating the mesh of the human body, the 3D rigging was required to produce the skeletal animation. The rigging refers to generate the bone structures to manipulate each part of the mesh. The bone structures work together with the weight painting, which determines the movement of the mesh section with the corresponding joint, i.e. the control point.

The research community majorly focused on the statistic bodies representation, which is not compatible with the current film and game industries. Skinned Multi-Person Linear model (SMPL) [29] was presented recently to describe a wide variety of body shapes. It is a simple linear formulation learned from a large human-pose database. It is compatible with the standard industry pipeline and rendering requirements. This major part of this thesis is based on the SMPL model.

Chapter 2

3D human reconstruction

2.1 Introduction

The creation of realistic humans is crucial in computer-generated imagery (CGI) in films, animations and games. Image-based 3D human reconstruction is an important topic in virtual dressing [39], VR/AR tech [10], image and video editing [21]. It's a hot topic starting from 2D pose detection [21, 9, 38] , 3D pose detection [33, 48, 43] and model-based full reconstruction [47]. However, due to the ambiguity of the 3D information, it is still challenging to recover an accurate human model from a single RGB image. Even worse, multiple variations in in-the-wild images, including human body shapes, clothes, environment, and viewpoints, gives this inverse problem multiple solutions.

The optimal representation of the 3D object remains the open question in the research field. Generally, there are two categories of research methods to reconstruct 3D objects from in-the-wild images, the volumetric way, and the parametric way. Recent work in the volumetric representation explores the voxel [34, 17], octree [46] and point cloud [16] to recognize, segment or reconstruct the 3D objects. However, the highly nonlinear mapping from 2D positions to the corresponding 3D positions makes the learning process difficult to proceed. The output model can be problematic and far away from the original object. Specifically, the estimation of the undressed 3D human body in volumetric representation sometimes have broken body parts due to the bad viewpoint or occlusion in the input image.

The model-based parametric representation enables the researchers to fully reconstruct the body shape even with several visible parts of the body. The unreasonable artifacts that happened in volumetric representation could be avoided with the skeleton regression method. Besides, the embedded auto rigging algorithm in model-based parametric representation and blend weight make the animation of the human model more feasible.

2.2 Body reconstruction

2.2.1 SMPL model

Skinned Multi-Person Linear Model (SMPL) is a generalized animated human body model representing different shapes and poses [29]. It proposed a parametric human body model with parameters on decoupled identity-dependent shape and pose-dependent shape. The significant advantage of SMPL model is that it can

- represent different body shapes
- naturally deform with different motion
- be easily formed and rendered by the existing graphics pipelines

The SMPL model mesh includes $N = 6890$ vertices on the surface and $K = 23$ joints to control vertices. The training dataset is composed of 1786 high-resolution 3D scan models, and the loss function is the Euclidean distance of the each vertices between the SMPL-generated models and the registered models. Before diving into the detailed mathematical description, we defined the crucial model parameters and functions here:

- N concatenated vertices $\overline{\mathbf{T}} \in \mathbb{R}^{3\mathrm{N}}$, here $\mathrm{N} = 6890$
- zero pose $\vec{\theta}^{*}$
- shape parameter $\vec{\beta}$
- pose parameter $\vec{\theta}$

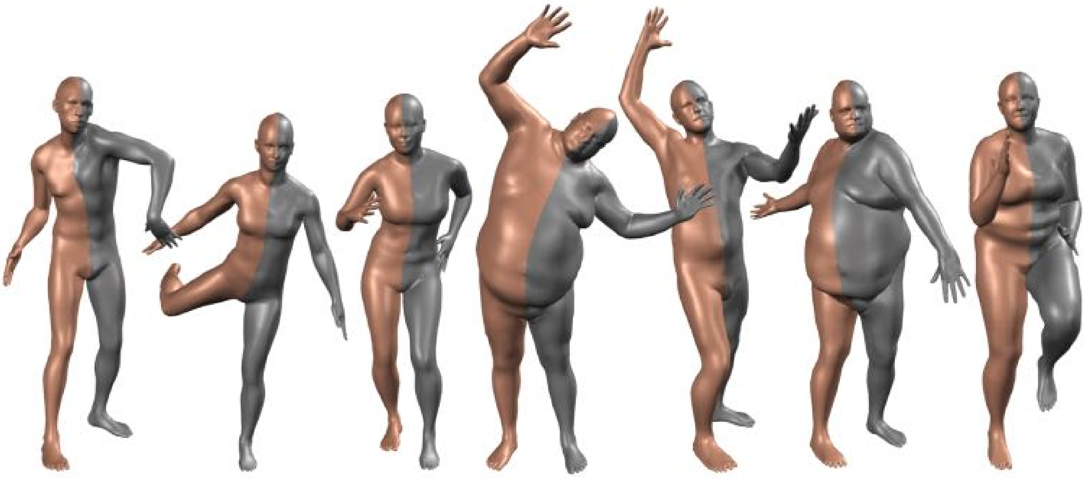


Figure 2-1: Skinned Multi-Person Linear Model (SMPL) [29] model to fit in the 3D meshes.

- blend shape function $B_{\mathrm{S}}(\vec{\beta}) : \mathbb{R}^{|\vec{\beta}|} \mapsto \mathbb{R}^{3\mathrm{N}}$
- prediction of K joint locations $J(\vec{\beta}) : \mathbb{R}^{|\vec{\beta}|} \mapsto \mathbb{R}^{3\mathrm{K}}$
- pose-dependent blend shape function $B_P(\tilde{\theta}) : \mathbb{R}^{|\tilde{\theta}|} \mapsto \mathbb{R}^{3\mathrm{N}}$
- blend weight $\mathcal{W} \in \mathbb{R}^{\mathrm{N}\times\mathrm{K}}$

The blend shape $B_{\mathrm{S}}(\vec{\beta})$ outputs the vertices location of rest human body $\vec{\theta} = \vec{\theta}^{*}$. For different body shape, we use the principal component analysis (PCA) and retrieve the first ten coefficient $\vec{\beta} = [\beta_1, \ldots, \beta_{10}]$ for simplicity, and $S_{\mathrm{n}} \in \mathbb{R}^{3\mathrm{N}}$ is the orthogonal principal components of shape displacement:

$$B_{\mathrm{S}}(\vec{\beta}; \mathcal{S}) = \sum_{n=1}^{10} \beta_{\mathrm{n}} S_{\mathrm{n}} \tag{2.1}$$

The joint location function $J(\vec{\beta})$ outputs the locations of K joints, since different people have different skeletal systems and the specific skeletal systems are independent with the pose.

The pose-dependent blend shape $B_{\mathrm{p}}(\vec{\theta})$ outputs the vertices location of human body in different pose. The two blend shape $B_{\mathrm{p}}(\vec{\beta})$ and $B_{\mathrm{p}}(\vec{\theta})$ can be linearly combined.

The body model utilizes the standard skeletal rig, which has $K = 23$ joints. Assuming that the local rotation angle concerning its parent in the kinematic tree is

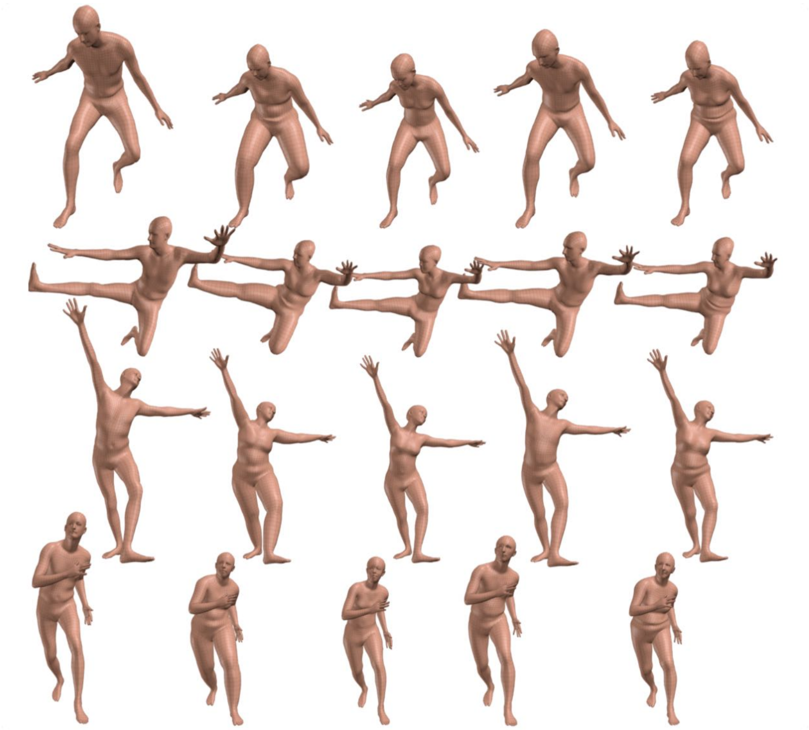


Figure 2-2: SMPL [29] sample human body with decomposed pose and shape. Pose parameters $\vec{\theta}$ vary from top to bottom and shape parameter $\vec{\beta}$ vary from left to right.

$\vec{\omega}_{\mathrm{k}} \in \mathbb{R}^3$, the pose parameter is

$$\vec{\theta} = \left[\vec{\omega}_0^{\mathrm{T}}, \ldots, \vec{\omega}_{\mathrm{K}}^{\mathrm{T}}\right]^{\mathrm{T}} \tag{2.2}$$

The number of pose parameters is $3K + 3 = 72$; i.e., 3 for each part plus 3 for the root orientation. Each set of pose parameters represents a set of poses, and it is independent of body shape.

For each joint j, the rotation matrix can be retrieved by the Rodriguez formula:

$$\exp\left(\vec{w}_{\mathrm{j}}\right) = \mathcal{I} + \hat{\bar{w}}_{\mathrm{j}} \sin\left(\|\bar{w}_{\mathrm{j}}\|\right) + \hat{\bar{w}}_{\mathrm{j}}^2 \cos\left(\|\bar{w}_{\mathrm{j}}\|\right) \tag{2.3}$$

So, the standard linear blend skinning function is:

$$W(\bar{T}, J, \vec{\theta}, \mathcal{W}) : \mathbb{R}^{3\mathrm{N} \times 3\mathrm{K} \times |\tilde{\theta}| \times |\mathcal{W}|} \mapsto \mathbb{R}^{3N} \tag{2.4}$$

In the traditional rendering and animation pipeline, the maximum number of entries in each column of the weight matrix W is 4. In other words, every vertex on the body surface can be affected by a maximum of four joints. The weight matrix is sparse. The vertices i in $\vec{T}$ can be formulated as:

$$\bar{t}'_{\mathrm{i}} = \sum_{k=1}^{K} w_{\mathrm{k,i}} G'_{\mathrm{k}}(\vec{\theta}, J(\vec{\beta})) \left(\bar{t}_{\mathrm{i}} + b_{\mathrm{S,i}}(\vec{\beta}) + b_{\mathrm{P,i}}(\vec{\theta}) \right) \tag{2.5}$$

$$G'_{\mathrm{k}}(\vec{\theta}, J) = G_{\mathrm{k}}(\vec{\theta}, J) G_{\mathrm{k}} \left(\vec{\theta}^{*}, J \right)^{-1} \tag{2.6}$$

$$G_{\mathrm{k}}(\vec{\theta}, J) = \prod_{j \in A(k)} \left[\begin{array}{cc} \exp\left(\vec{w}_{\mathrm{j}}\right) & j_{\mathrm{j}} \\ \bar{0} & 1 \end{array} \right] \tag{2.7}$$

Here $w_{k,i}$ represents the items in blend weight matrix W. It means the weight of the vertices I from the k joint. $G_{\mathrm{k}}(\vec{\theta}, J)$ is the global transfer matrix of the k joint. $A(k)$ denotes the ordered set of joint ancestors of joint k. j_{j} is the location of joint j in joint location matrix J. $b_{\mathrm{S,i}}(\vec{\beta})$ represents the displacement in shape blend and $b_{\mathrm{P,i}}(\vec{\theta})$ represents the one in the pose blend.

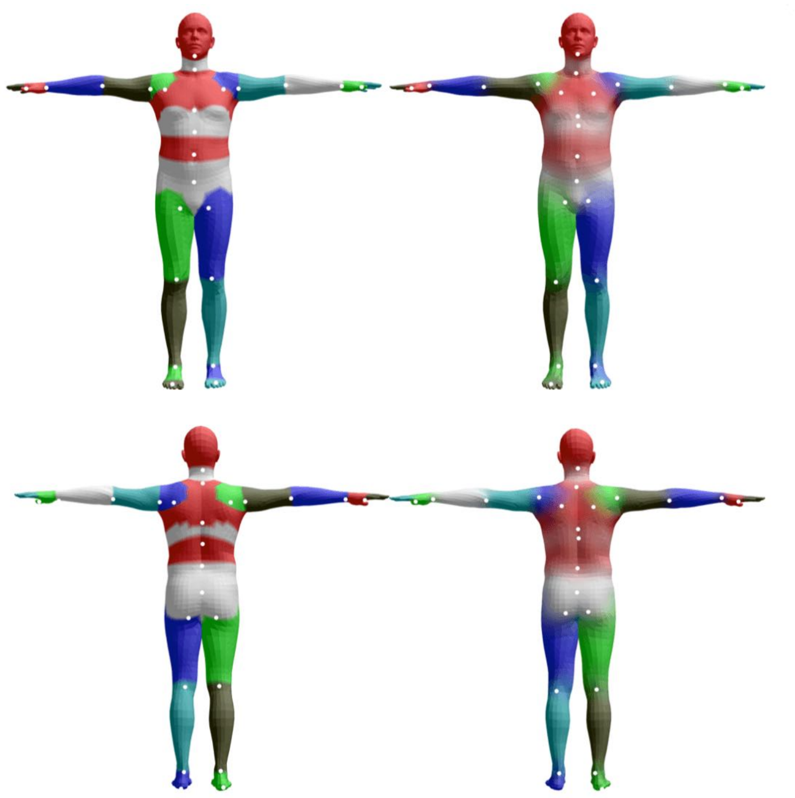
After the SMPL model description, the optimization process could be divided into the pose part and shape part. The first part was optimized on the multi-pose database, which contained the 40 people and 1786 registration data (891 registrations spanning 20 females and 895 registrations spanning 20 males) [7]. The second part optimized on multi-shape data of CAESA (1700 registrations for males and 2100 for females) [42].

In the optimization process, we need to obtain the optimal parameters of $\Phi = \{\bar{T}, \mathcal{W}, \mathcal{S}, \mathcal{J}, \mathcal{P}\}$. We first optimize the $\{\mathcal{J}, \mathcal{W}, \mathcal{P}\}$ in multi-pose dataset and then $\{\bar{T}, \mathcal{S}\}$ in the multi-shape dataset.

In the pose parameter optimization, we minimize an objective function consisting of a data term E_D and several regularization term $\{E_{\mathrm{J}}, E_{\mathrm{Y}}, E_{\mathrm{P}}, E_{\mathrm{W}}\}$.

$$E\left(\hat{T}^{\mathrm{P}}, \hat{J}^{\mathrm{P}}, \Theta, \mathcal{W}, \mathcal{P}\right) = E_{\mathrm{D}} + \lambda_{\mathrm{Y}} E_{\mathrm{Y}} + \lambda_{\mathrm{J}} E_{\mathrm{J}} + \lambda_{\mathrm{P}} E_{\mathrm{P}} + E_{\mathrm{W}} \tag{2.8}$$

Here E_{D} is the squared Euclidean distance between registration vertices and model vertices. E_{Y} is the symmetry regularization to penalize the left-right symmetry. The model was manually segmented into 24 parts, and E_{J} penalize the difference of the vertices center of each segment and the joints. To prevent the overfitting in the optimization process, E_{P} and E_{W} are the regularization of $\mathcal{W}, \mathcal{P}$.



(a) *Segmentation* **(b)** *Initialization* $\mathcal{W}_I$

Figure 2-3: Initialization of joints (white dots), segmentations (a) and blend weights (b).

In the shape parameters optimization, we need to utilize the pre-trained parameters in pose optimization $\{\mathcal{J}, \mathcal{W}, \mathcal{P}\}$ to initialize the models in the dataset. It ensures the pose blend and shape blend could not affect each other. For each registration V_j^S we need to predict the pose that minimizes the difference between the transformed and the original one.

$$\vec{\theta}_{\mathrm{j}} = \arg\min_{\vec{\theta}} \sum_{e} \left\| W_{\mathrm{e}}\left(\hat{\mathbf{T}}^{\mathrm{P}}_{\mu} + B_{\mathrm{P}}(\vec{\theta};\mathcal{P}), \hat{\mathbf{J}}^{\mathrm{P}}_{\mu}, \vec{\theta}, \mathcal{W}\right) - \mathbf{V}^{\mathrm{S}}_{\mathrm{j,e}} \right\|^2 \tag{2.9}$$

Here $\hat{T}^{\mathrm{P}}_{\mu}$ is the mean pose in the multi-pose dataset and $\hat{J}^{\mathrm{P}}_{\mu}$ is the mean joint location in the multi-pose dataset.

The rest pose $\overrightarrow{\theta^*}$ registration $\hat{T}_{\mathrm{j}}^{\mathrm{S}}$ can be predicted as:

$$\hat{T}_{\mathrm{j}}^{\mathrm{S}} = \arg\min_{\vec{T}} \left\| W\left(\hat{T} + B_{\mathrm{p}}\left(\vec{\theta}_{\mathrm{j}}; \mathcal{P}\right), \mathcal{J}\hat{T}, \vec{\theta}, \mathcal{W}\right) - V_{\mathrm{j}}^{\mathrm{S}} \right\|^2 \tag{2.10}$$

After that, the principal component analysis was conducted on the reconstructed zero pose θ^* mesh to evaluate the $\{\bar{T}, \mathcal{S}\}$.

2.2.2 End-to-end reconstruction from a single image

The common way to estimate the 3D human shape from a single RGB image can be composed of two stages. The first step relies on the 2D key joints detection from images, and then 3D joints location estimation of the 2D joints [6, 25, 44]. After that, the researchers could construct the whole human model with 3D joints information. This multi-stage process loses information step by step and makes the output model unrealistic. Besides, occlusion and truncation make the 2D detection unreliable, and the 3D joints mapping requires explicit constraints of the joint angle limits. A direct end-to-end reconstruction from a single image is preferred with the usage of convolutional neural networks (CNNs) and generative adversarial network (GANs) [24].

The standard way to reconstruct the human body model from 3D joint rotation is not robust. On the one hand, 3D joint location alone does not constrain the full DoF at each joint; on the other hand, joints are sparse, whereas a surface defines the human body in 3D space [24]. This end-to-end method also deals with the problems of data in the previous framework.

1. Lack of 3D in-the-wild ground-truth dataset. A lot of data were captured in the lab environment with multiple RGB-D high-resolution cameras.
2. Multi-mapping problem of the 3D shape and the corresponding 2D image.

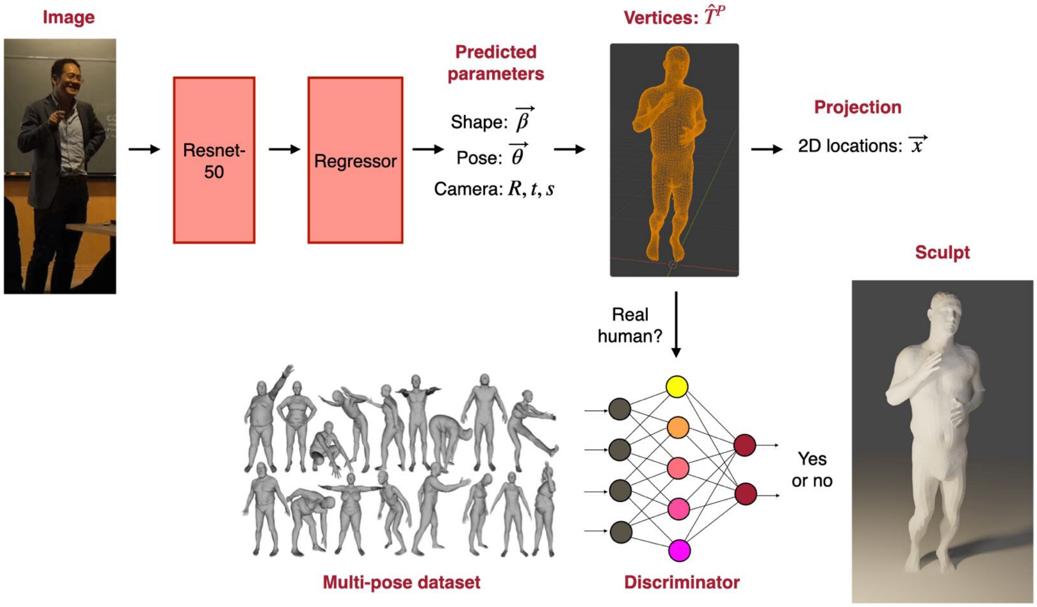


Figure 2-4: Overview of the end-to-end framework. A single image is the input for the convolutional encoder ResNet-50. The regressor transferss the output of Resnet-50 into the predicted parameters $\vec{\beta}, \vec{\theta}, R, t, s$. The parameters are used to reconstructed vertices by the SMPL model. The 3D model has been projected into the 2D locations. In the meanwhile, the pre-trained discriminator is used to identify the problematic computer-generated models.

The discriminator is to deal with the problematic 3D model. It embeds a fast check algorithm for constraints of the joint rotation matrix. Since the significant target of the discriminator is to ensure that SMPL parameters reasonable, there is no need to use a 2D image corresponding to 3D ground-truth shape dataset for training. To fully make use of the SMPL model, we could use the separate pose $\vec{\theta}$ discriminator and shape $\vec{\beta}$ discriminator. Furthermore, the pose discriminator could be decomposed into each of $K = 23$ joint discriminators and one global pose discriminator.

The loss function of the network use here is:

$$L = \lambda \left(L_{\text{reproj}} + \delta L_{\text{3D}} \right) + L_{\text{adv}} \tag{2.11}$$

Here, λ is used to control the weight of each loss function. δ is set to be one if there is the corresponding 3D shape of the input 2D images, or 0 if no corresponding 3D shape.

The L_{reproj} is the penalty to minimize the difference of the computer-generated 3D joints and its corresponding 2D joints.

$$L_{\mathrm{reproj}} = \sum \| v_{\mathrm{i}} \left(x_{\mathrm{i}} - \hat{x}_{\mathrm{i}} \right) \|_1 \tag{2.12}$$

Here, use the projection function $\hat{x}_{\mathrm{i}} = s\Pi(RX(\theta, \beta)) + t_{\mathrm{c}}$. Π represents the orthogonal projection.

$$\begin{aligned} L_{\mathrm{3D}} &= L_{\mathrm{3Djoints}} + L_{\mathrm{3Dsmpl}} && (2.13) \\ L_{\mathrm{joints}} &= \left\| \left(X_{\mathrm{i}} - \hat{X}_{\mathrm{i}} \right) \right\|_2^2 && (2.14) \\ L_{\mathrm{smpl}} &= \left\| [\beta_{\mathrm{i}}, \theta_{\mathrm{i}}] - \left[\hat{\beta}_{\mathrm{i}}, \hat{\theta}_{\mathrm{i}} \right] \right\|_2^2 && (2.15) \end{aligned}$$

In the GANs training process, mode collapse did not happen because the network need not only to deceive the discriminator but also minimize the loss function of 3D shape projection. The adversarial loss function for the encoder is:

$$\min L_{\mathrm{abv}}(E) = \sum_i \mathbb{E}_{\Theta \mathrm{pE}} \left[\left(D_{\mathrm{i}}(E(I) - 1)^2 \right] \tag{2.16}$$

and the objective for each discriminator is:

$$\min L \left(D_{\mathrm{i}} \right) = \mathbb{E}_{\Theta \mathrm{pdata}} \left[(D_{\mathrm{i}}(\Theta) - 1)^2 \right] + \mathbb{E}_{\Theta \mathrm{pE}} \left[D_{\mathrm{i}} \left(E(I)^2 \right] \tag{2.17}$$

Dataset

The in-the-wild image datasets annotated with 2D keypoints that we use is LSP, LSP-extended [23] , MPII [2] and MS COCO [27]. For the 3D shape datasets we use Human 3.6M [22] and MPI-INF-3DHP [35]. For the Human 3.6M [22] we obtain the SMPL parameters using the MoSh [28] from 3D markers.

Architecture

Initially, the network needs the input of a single RGB full-shot image of the target person. The encoder for the image is a pre-trained ResNet-50 network on the ImageNet classification.

The ResNet-50 is the popular residual networks used as the backbone for multiple computer vision tasks. The ResNet-50 is the smaller version of ResNet-152. Deep neural networks sometimes are hard to train because of the notorious vanishing problem. When the stacked layers go more in-depth, the performance on the training data gets saturated. The strength of the ResNet is to skip the connection. This skip connection ensures the model to learn identity function so that they could be as good the original smaller one.

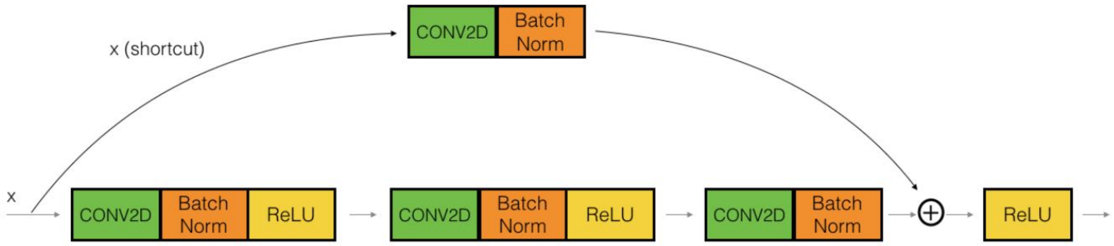
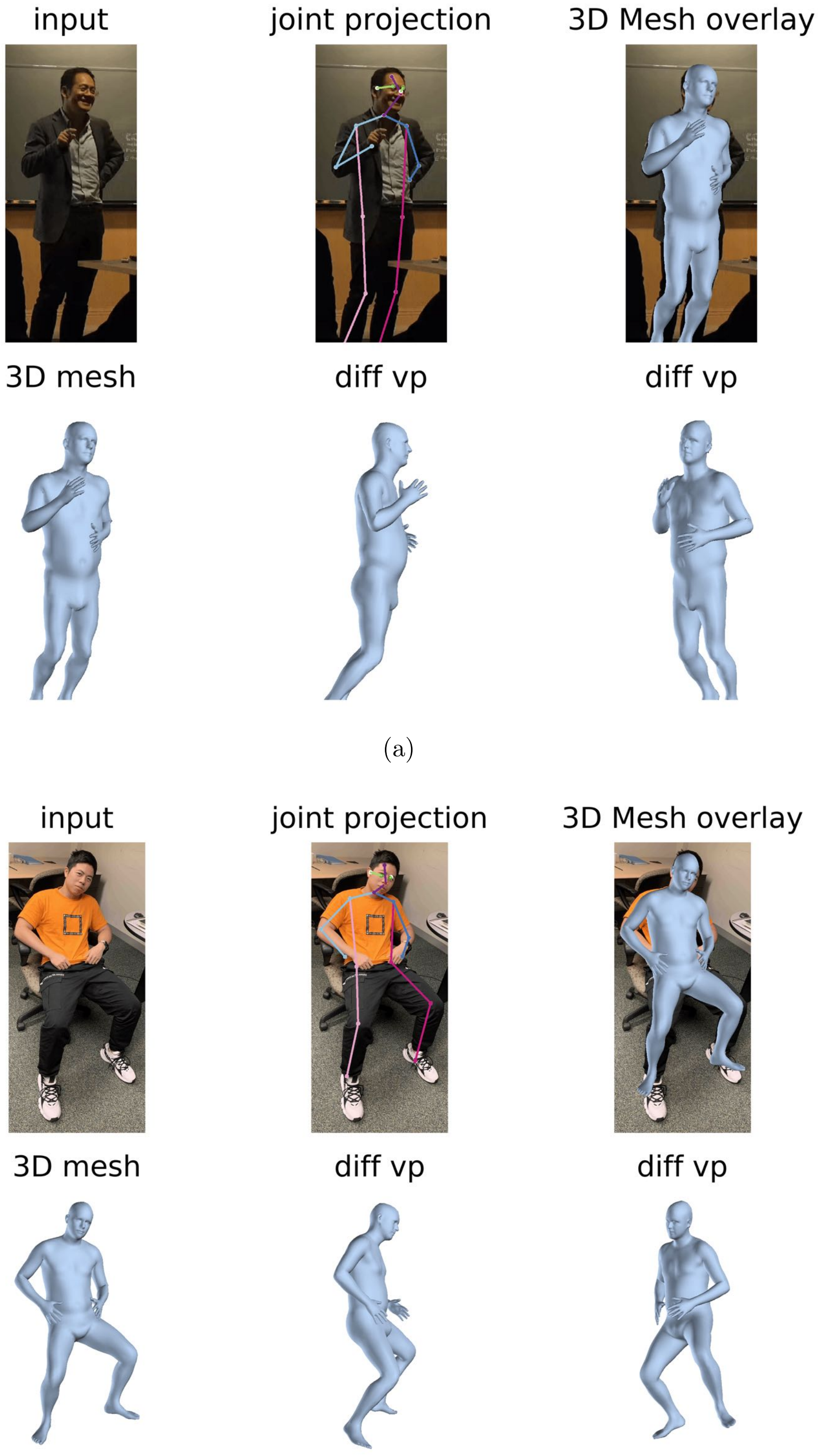


Figure 2-5: The structure of the ResNet-50.

The encoder (Resnet-50) output a feature $\in \mathbb{R}^{2048}$, after three iterations in the regressor (Three layers, $2048D \rightarrow 1024D \rightarrow 1024D \rightarrow 85D$) the network generates the camera parameter as global rotation $R \in \mathbb{R}^{3\times3}$ in the axis-angle representation, translation $t \in \mathbb{R}^2$ and the scale $s \in \mathbb{R}$; the parameters of the SMPL model as shape $\vec{\beta}$ and pose $\vec{\theta}$. The discriminator is two fully-connected layers with 10, 5, 1 neurons.

Evaluation

Here we used image samples to evaluate the quality of the computer-generated models. The 3D mesh overlaid with the original image, and the joints projection to 2D was also shown in the figure.



(a)

(b)

Figure 2-6: Comparison of the computer-generated 3D overlapped with the 2D image. The upper left is the input image, and the 2D joints detection overlapped with the image is shown in the upper center. The overlapping 3D mesh is in the upper right part. The down part is the corresponding 3D mesh and its different viewpoints.

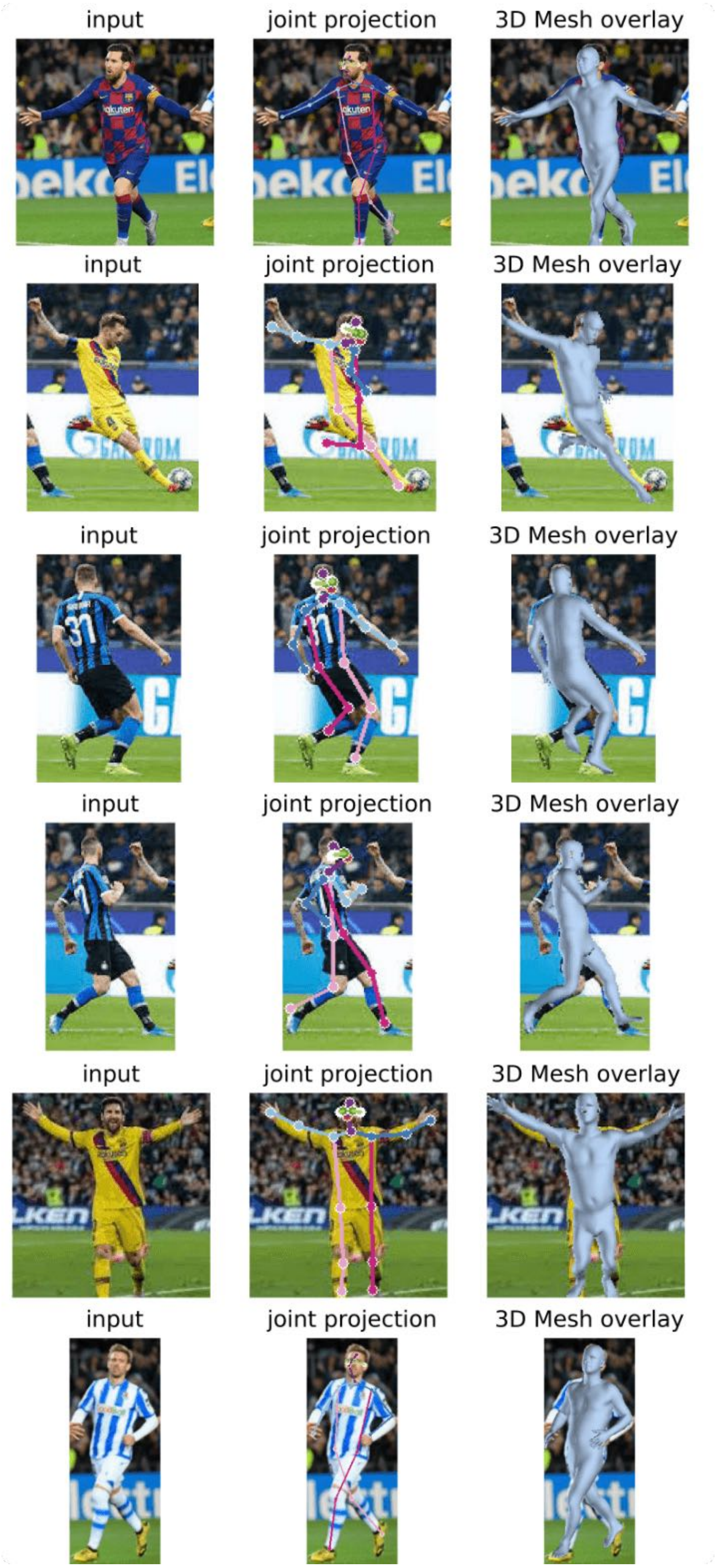


Figure 2-7: More 3D models samples to evaluate the end-to-end framework.

2.2.3 Animated motion

We utilize the archive of motion capture as surface shapes (AMASS) dataset [31] to test the generated animation of the image. AMASS is a large and varied dataset of human motion that unified the 15 different marker-based mocap datasets. The MoSh++ was introduced here to generated the 3D human model mesh from mocap data. AMASS has 42 hours of mocap, 346 subjects, and 11451 motions.

The original MoSh method relied on the SCAPE model, which is not compatible with the current industry standard. MoSh++ utilize the SMPL (2.2.1) model. It captures the body shape, pose, and soft-tissue dynamics. It also provides the rigged skeleton in animation.

Here is the table of datasets contained in the AMASS.

	Markers	Subjects	Motions	Minutes
ACCAD	82	20	258	27.22
BioMotion	41	111	3130	541.82
CMU	41	97	2030	559.18
EKUT	46	4	349	30.74
Eyes Japan	37	12	795	385.42
HumanEva	39	3	28	8.48
KIT	50	55	4233	662.04
MPI HDM05	41	4	219	147.63
MPI limits	53	3	40	24.14
MPI MoSh	87	20	78	16.65
SFU	53	7	44	15.23
SSM	86	3	30	1.87
TCD Hand	91	1	62	8.05
TotalCapture	53	5	40	43.71
Transitions	53	1	115	15.84
Total	-	**346**	**11451**	**2,488.01**

Table 2.1: Datasets contained in AMASS [31] .

The large group of the marker-based optical human mocap datasets

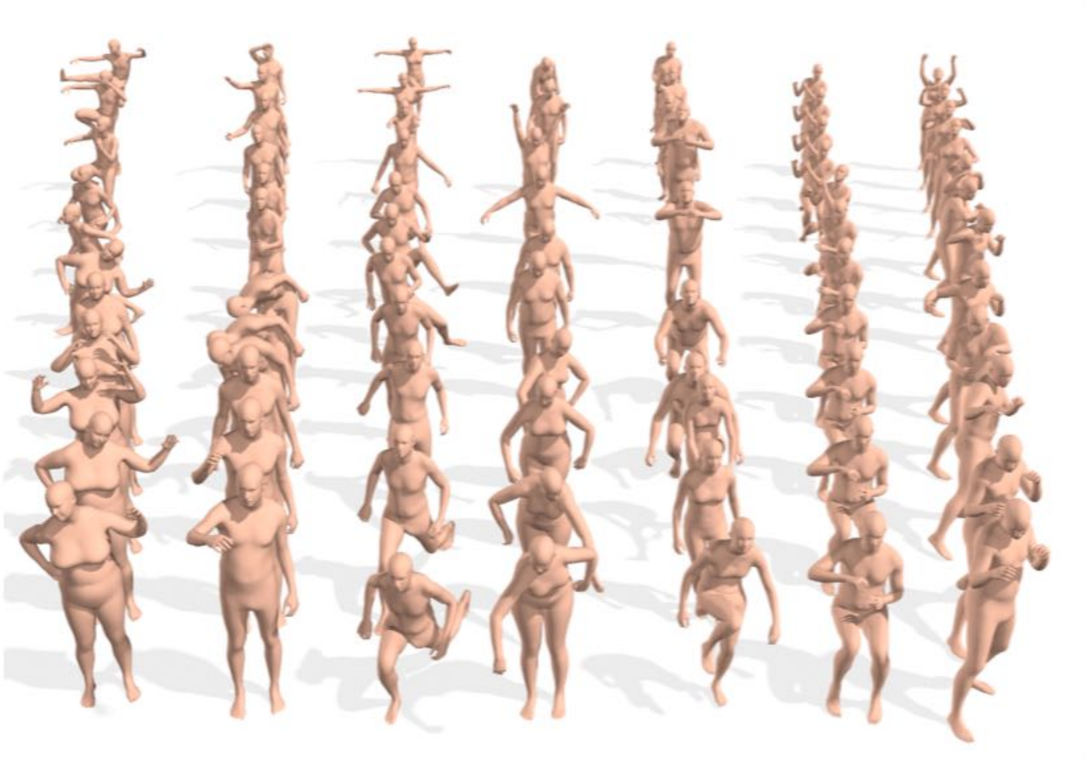


Figure 2-8: Archival of the mocap datasets [31]. From left to right CMU [13], MPI-HDM05 [36, 37], MPI- Pose Limits [1], KIT [32], BioMotion Lab [45], TCD [20] and ACCAD [26] datasets.

The pose data $\vec{\theta}$ could be transferred to the current SMPL model to visualize the animation. Here we used the sample of the volunteers' 3D human model to explain it. The shape data $\vec{\beta}$ was extracted from the end-to-end reconstruction was preserved, and the pose data $\vec{\theta}$ was adapted on the SMPL model to animate 3D human body.

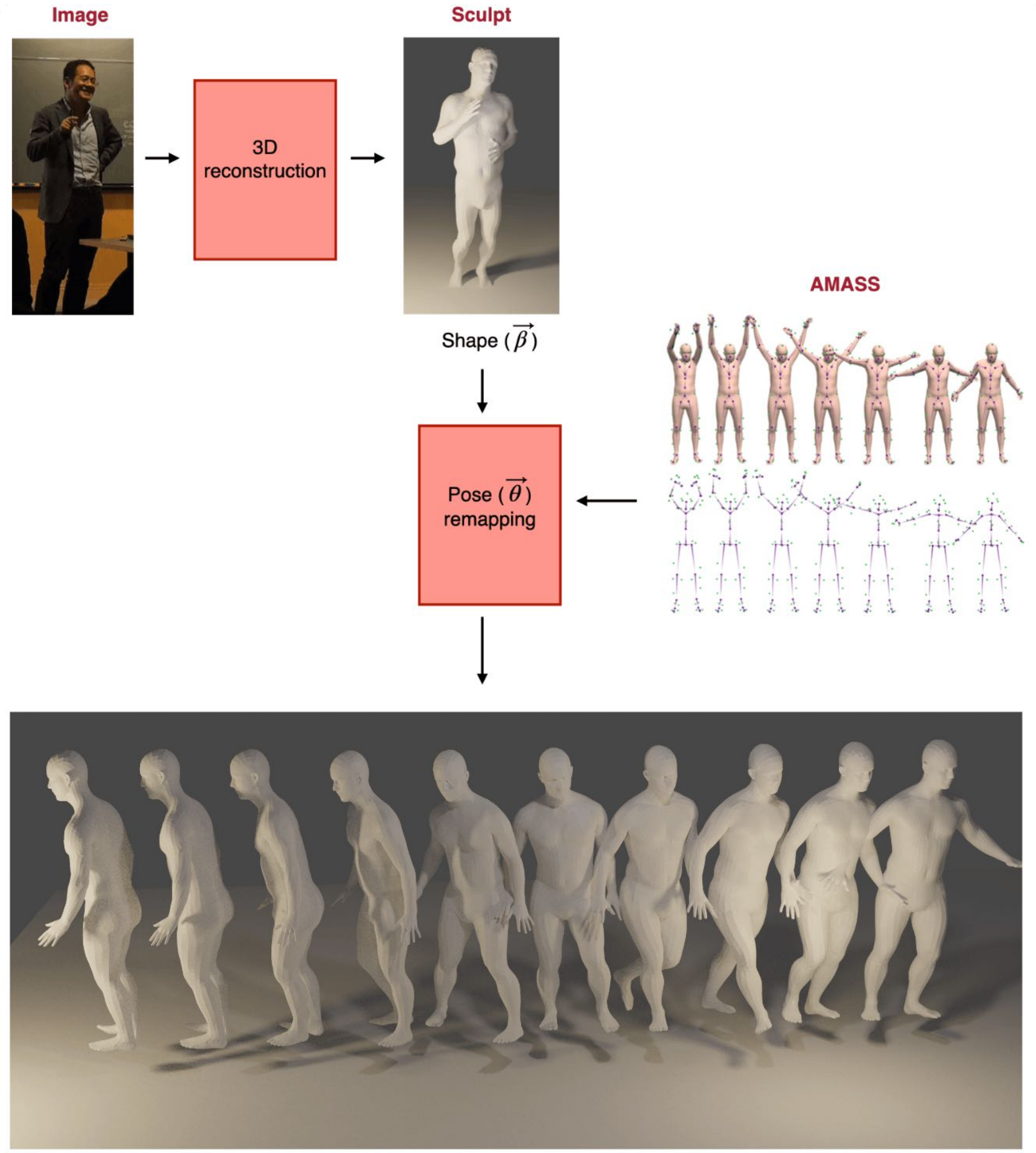


Image
3D reconstruction
Sculpt
Shape ($\overrightarrow{\beta}$)
AMASS
Pose ($\overrightarrow{\theta}$) remapping

Figure 2-9: Workflow to animate the corresponding model. The input image was reconstructed by the end-to-end method and outputted the SMPL model. The shape parameters remained the same, and the pose parameters were captured in the AMASS database. The generated animated model was displayed in time series.

2.3 Face reconstruction

The 3D face reconstruction is another popular topic in the computer graphics research field. It is a fundamental problem with extra difficulty. It requires the realistic 3D mesh and the texture on the surface, i.e., RGB color, roughness, normal vectors field. Besides, compared to the $K = 23$ joints in the SMPL body model, the facial landmarks detection requires more control points to animate complex facial expression, e.g., smile, laugh, sorrow. We could easily identify the problematic computer-generated face since human eyes are susceptible to details, especially for the facial animation.

Here we combined the methods of facial landmarks 3D detection and the morphable facial model to reconstruct 3D face with texture from 2D images. It could also be used in the face recognization field.

2.3.1 Detect facial landmarks

Much recent research focus on the 2D landmark detection from a single for facial recognization. LS3D-W database [8] used the most state-of-art landmark localization and residual block to build a baseline for a 3D landmark detection task. It was constructed by annotating the images from AFLW, 300VW, 300W, and FDDB with 69 key points. Based on a massive 2D facial landmark dataset, it transferred the 2D information into the 3D information with convolutional neural networks. The database is in total 230,000 images with 3D annotations.

In the two-stage 3D detection method, the 2D-to-3D face-alignment net (FAN) [8] first predict the 2D face alignment with four hourglass-like neural networks. All bottleneck blocks, i.e. the layer with reduced number of channels, are replaced by hierarchical, parallel, and multi-scale block. The input is the combination of the images and 2D landmarks, layered by an hourglass and ResNet 152, and the output is the 3D landmarks.

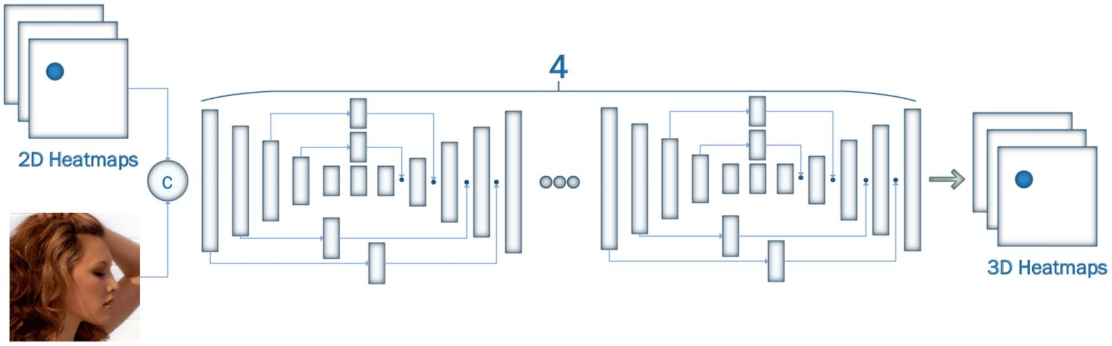


Figure 2-10: 2D-to-3D FAN used as the converter of the LS3D-W database [8].

Evaluation of the 2D-to-3D face-alignment net

FAN model is applicable to reconstruct the 3D landmarks. With the 3D information predicted from the image, we could utilize the morphable model to generate the facial mesh and texture.

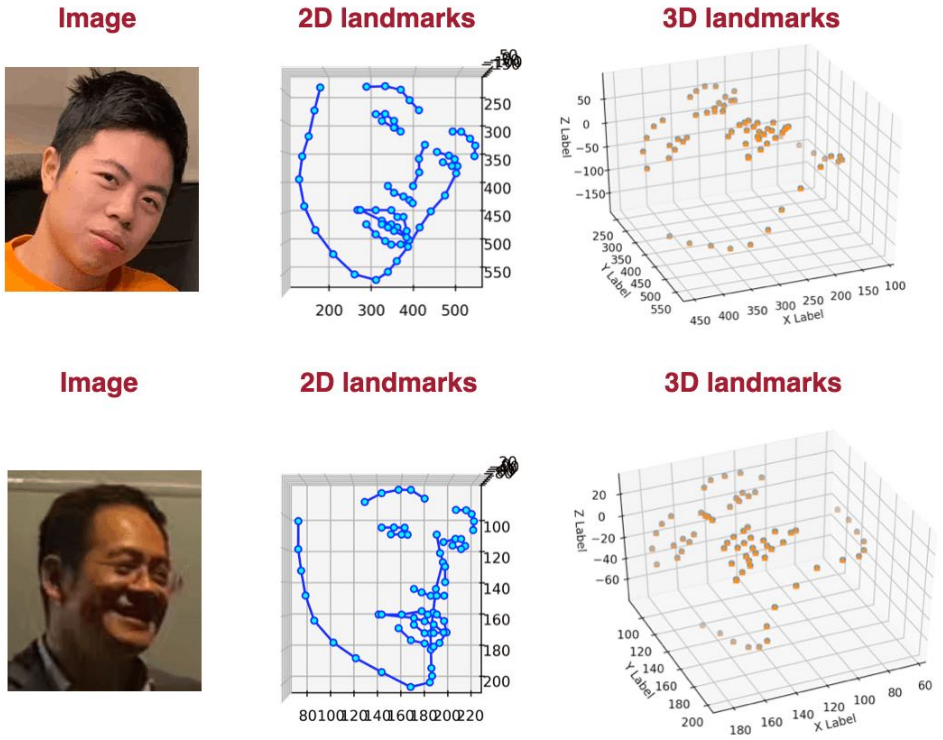


Figure 2-11: Image, the corresponding 2D landmarks and 3D landmarks.

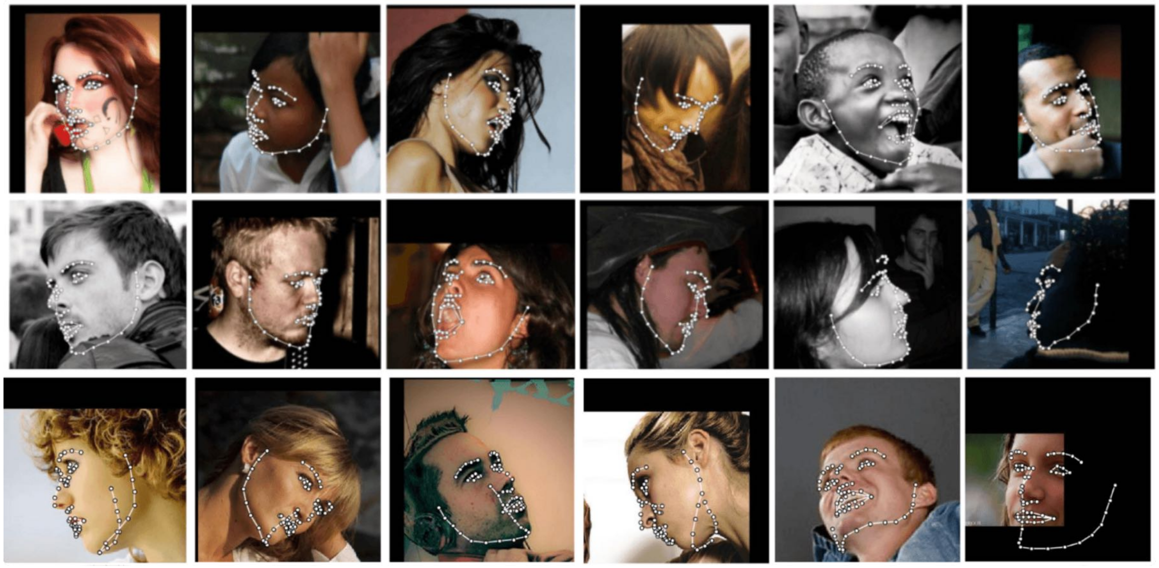


Figure 2-12: More samples for the FAN 2D landmarks detection. Images from the LS3D-W database [8].

2.3.2 Morphable model for 3D faces

3D morphable facial model is a genaralized model-based method [4]. The shape information can be represented by shape vector S = $(X_1, Y_1, Z_1, X_2 \ldots ..Z_\mathrm{n})^T$, and the texture information (RGB color value) can be represented by T = $(R_1, G_1, B_1, R_2 \ldots ..B_\mathrm{n})^T$. Here X, Y, Z is the 3D location of the corresponding vertices and the R (red) , G (green) , B (blue) is the color of the vertices.

We could use principal component analysis (PCA) to retrieve orthogonal components to generate arbitrary face, and each of them are the combination of the face models.

$$S_\mathrm{model} = \bar{S} + \sum_{i=1}^{m-1} \alpha_\mathrm{i} s_\mathrm{i} \tag{2.18}$$

$$T_\mathrm{model} = \bar{T} + \sum_{i=1}^{m-1} \beta_\mathrm{i} t_\mathrm{i} \tag{2.19}$$

Here $\bar{S}$ and $\bar{T}$ are the mesh and texture of the standard face. s_i and t_i are the eigenvectors of the covariance matrix.

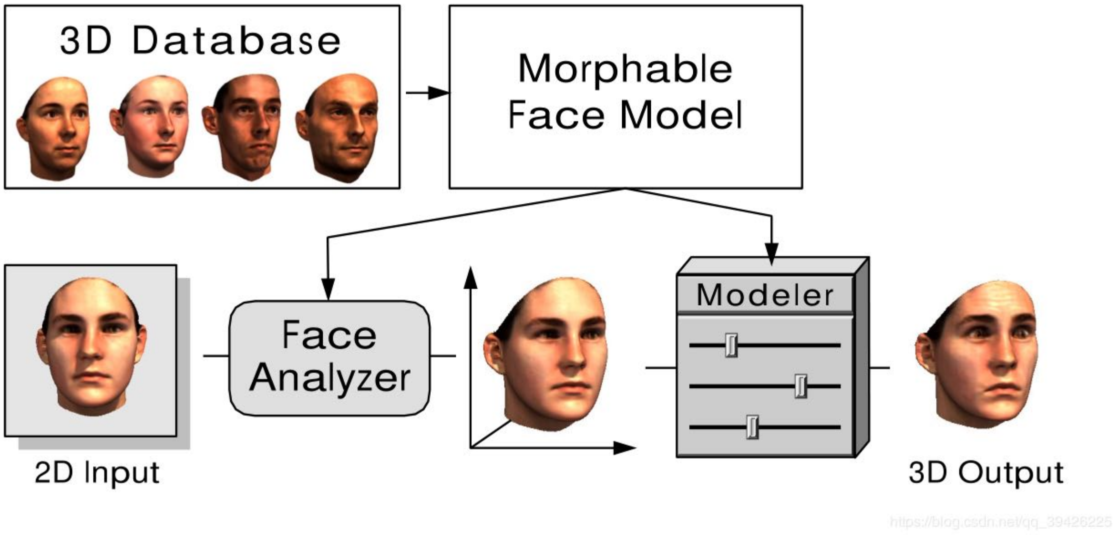


Figure 2-13: Morphable facial model from a dataset of prototypical 3D scans of faces [4]. The 3D face can be derived from a novel image. The shape and texture could be modified in natural way.

To match the target face with the mophable model, the loss function is the euclidean distance between the $I_{\text{model}}(x, y)$ and $I_{\text{input}}(x, y)$:

$$E_{\text{I}} = \sum \|I_{\text{input}}(x, y) - I_{\text{model}}(x, y)\|^2 \tag{2.20}$$

With the help of Blender KeenTool [15] plugin and the 3D joints locations extracted from the FAN, we could generate the corresponding 3D models from 2D facial images.

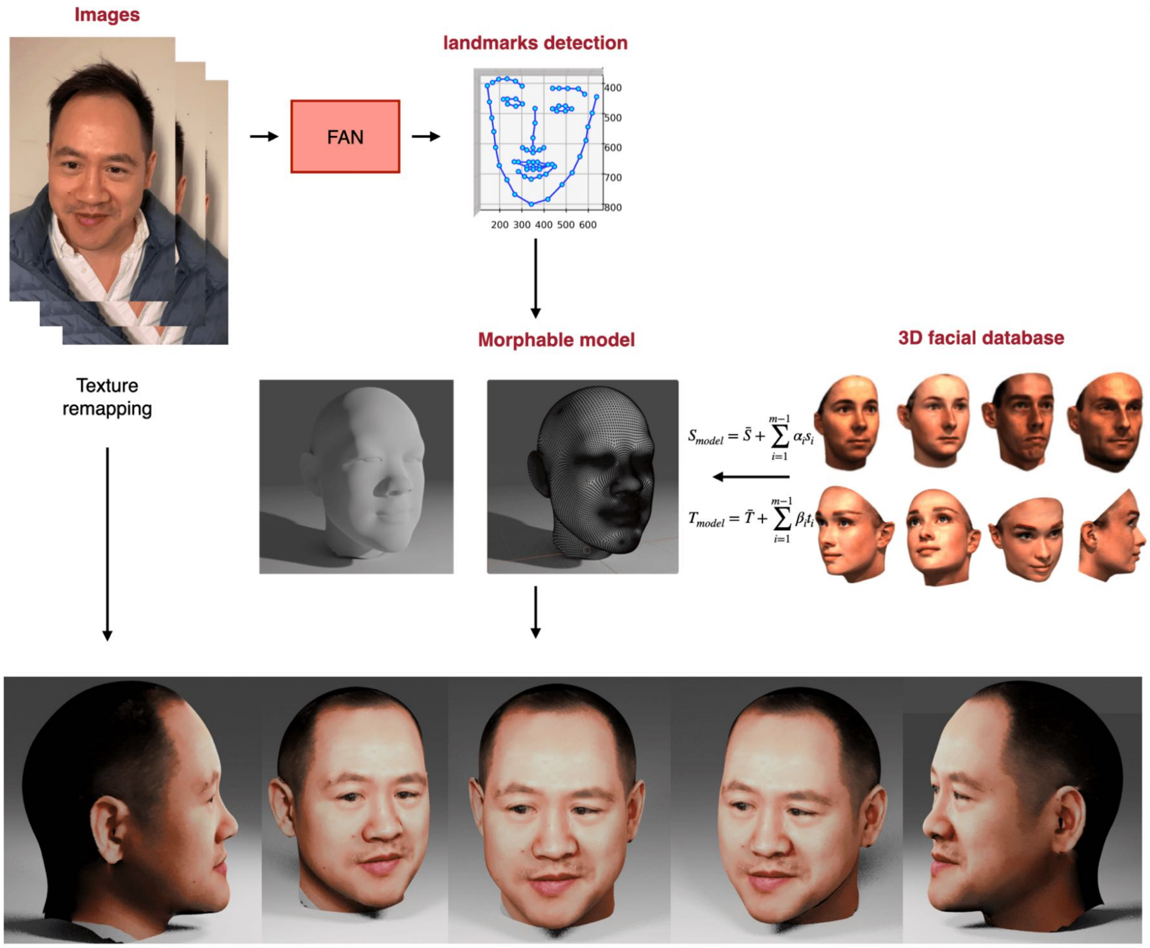


Figure 2-14: Face reconstruction with the morphable facial models. The detected landmarks had been sent to the pretrained morphable model, and the textures from original images had been merged into the computer-generated facial model.

Chapter 3

3D garments reconstruction

3.1 Introduction

The end-to-end 3D reconstruction of the human body via the SMPL model only infer the shape under the garments without texture. However, garments are essential in the realistic rendering in special effects in films, CG movies, animations. The 3D garments reconstruction is crucial in the e-Commence fashion industry.

The previous research has a major limitation because they use a single layer to represent the whole mesh, i.e., including the human mesh and the garments mesh. Estimates of the body shape and clothing from images have been attempted in [19, 11], but the body shape was not separated from the clothing. The generated cloth model cannot be transferred into another body model. With RGB-D cameras, researchers could generate similar looking synthetic clothing templates [12].

3.2 Garment reconstruction

The problem of garment reconstruction from RGB images into separated human body shape and clothing can be partially addressed by the multi-garment net (MGN) [3]. The multi-garment net could predict the body shape with the SMPL model and the clothing it covers from several images. Besides, this model could be transferred to different people with different poses. To train the multi-garment net, we proposed a digital wardrobe containing 712 digital garments.

The garments are separated into five templates as:

- Pants;
- ShortPants;
- ShirtNoCoat;
- TShirtNoCoat;
- LongCoat

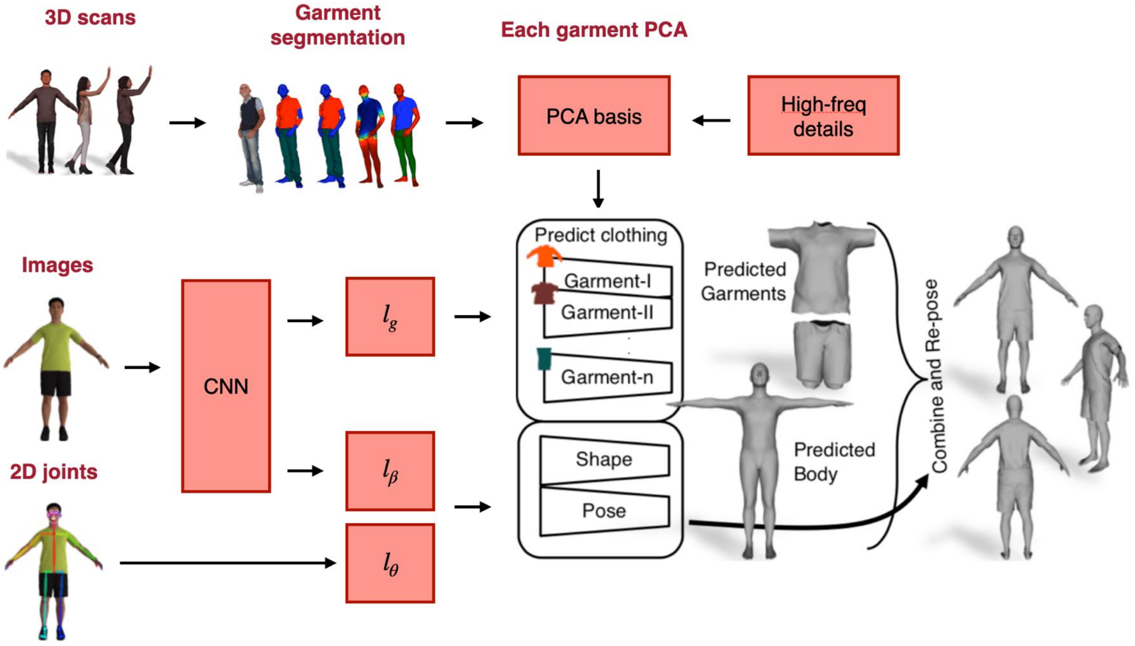


Figure 3-1: Detailed architecture of MGN [3]. CNN is used to encode image and 2D joint information. The garment network decoded the garment parameters to predict the garments parameters with PCA and added high-frequency details to the garment mesh.

Within each template, different clothes still possess diverse 3D shape. We need a linear system to minimize the distance between the template and the 3D scanning, and keep the laplacian on the surface of the template. In the registration process, we could get the vertex-based PCA for each garment. MGN was trained with multiple images, body pose and shape, PCA components of each garment. This method is better compared with the silhouette matching.

Data pre-processing

It requires the segmentation of registration of the 3D scan data. The body-aware scan segmentation will separate the skin, upper outer garment, and lower outer garment. All 3D scans will be annotated. After the non-rigid alignment, we could solve the Markov Random Field (MRF) on the UV mapping of the SMPL model to do the scan segmentation. To measure the garment prior, we could define the labels $\mathbf{l}_{\mathrm{g}}^{\mathrm{i}} \in \{0, 1\}$ indicting the vertices $\mathbf{v}_{\mathrm{i}} \in \mathcal{S}$ on the SMPL surface to overlap with the garment inner surface. Besides, we define the loss function increasing with the geodesic distance from the garment region boundary [3].

For each garments categories g, e.g., Pants, ShirtNoCoat, etc. , we could define a zero-pose template mesh G^{g}. Here we use the I^{g} as an indicator matrix to compute the correlation between the garment g vertex i and the body shape vertex j. If they are associated, let $I_{\mathrm{i,j}}^{\mathrm{g}} = 1$. As a result, I^{g} is a matrix representing the vertex on the SMPL model that overlapped with the garments.

The distance of the garments mesh and the SMPL model can be computed as:

$$\mathbf{D}^{\mathrm{g}} = \mathbf{G}^{\mathrm{g}} - \mathbf{I}^{\mathrm{g}} T\left(\beta^{\mathrm{g}}, \mathbf{0}_{\theta}, \mathbf{0}_{\mathbf{D}}\right) \tag{3.1}$$

Here $\mathbf{0}_{\theta}$ is the zero pose, β^{g} is the SMPL body shape. To compute the unposed clothing model T^{g} with new SMPL model with shape β and pose θ, we can get

$$T^{\mathrm{g}}\left(\beta, \theta, \mathbf{D}^{\mathrm{g}}\right) = \mathbf{I}^{\mathrm{g}} T(\beta, \theta, \mathbf{0}) + \mathbf{D}^{\mathrm{g}} \tag{3.2}$$

The skinning function W was used to compute the posed garment model:

$$G\left(\beta, \theta, \mathbf{D}^{\mathrm{g}}\right) = W\left(T^{\mathrm{g}}\left(\beta, \theta, \mathbf{D}^{\mathrm{g}}\right), J(\beta), \theta, \mathbf{W}\right) \tag{3.3}$$

Garment Registration

We used multi-part alignment on the segmented scans to non-rigidly match the body mesh and the garments templates to the scans. To deal with the problems of massive shape differences in each garment, we need to initialize each garment with the SMPL model. The deformed vertices $G_{\mathrm{init}}^{\mathrm{g}}$ could be used to dress other SMPL models.

After the registration, the generated pairs of images and bodies were the training dataset as well as the 3D garment pairs. The input of the multi-garment net was the segmented images of the corresponding 2D landmarks prediction. The underlying code l_p could be computed frame by frame as:

$$l_{\mathcal{P}} = f_{\mathrm{w}}^{\theta}(\mathcal{I}, \mathcal{J}) \tag{3.4}$$

The body shape l_p and l_g were computed from the F frames average latent code

$$\boldsymbol{l}_{\boldsymbol{\beta}}, \boldsymbol{l}_{\mathcal{G}} = \frac{1}{F}\sum_{f=0}^{F-1} f_{\mathrm{w}}^{\boldsymbol{\beta},\mathcal{G}}\left(\mathbf{I}_{\mathrm{f}}, \mathbf{J}_{\mathrm{f}}\right) \tag{3.5}$$

For each category of garments, the $M_{\omega}^{\mathrm{g}}(.)$ was individually trained through latent code $l_{\mathcal{G}}$. The output as the un-posed garment G^{g} was computed through the major components of PCA plus the high-frequency deviation $D^{\mathrm{hf,g}}$

$$M_{\mathrm{w}}^{\mathrm{g}}\left(l_{\mathcal{G}}, \mathbf{B}^{\mathrm{g}}\right) = \mathbf{G}^{\mathrm{g}} = \mathbf{B}^{\mathrm{g}}\boldsymbol{z}^{\mathrm{g}} + \mathbf{D}^{\mathrm{hf},\mathbf{g}} \tag{3.6}$$

The shape and pose underlying code l_{β} was computed in a fully-connect layer. The deviation D^{g} could be calculated as

$$\mathbf{D}^{\mathrm{g}} = M_{\mathrm{w}}^{\mathrm{g}}\left(\boldsymbol{l}_{\mathcal{G}}, \mathbf{B}^{\mathrm{g}}\right) - \mathbf{I}^{\mathrm{g}} T\left(\boldsymbol{\beta}, \mathbf{0}_{\theta}, \mathbf{0}_{\mathbf{D}}\right) \tag{3.7}$$

The final predict 3D vertex could be computed from $C\left(\beta, \theta_{\mathrm{f}}, D\right)$. The 2D segmented masks $\mathbf{R}_{\mathrm{f}}$ is

$$\mathbf{R}_{\mathrm{f}} = R\left(C\left(\boldsymbol{\beta}, \boldsymbol{\theta}_{\mathrm{f}}, \mathbf{D}\right), c\right) \tag{3.8}$$

Loss function

The loss function is the summary of 3D and 2D loss. The 3D vertex loss in canonical T-pose ($\theta = 0_\theta$)

$$\mathcal{L}_{0_\theta}^{3\mathrm{D}} = \left\| C\left(\boldsymbol{\beta}, \mathbf{0}_{\boldsymbol{\theta}}, \mathbf{D}\right) - C\left(\hat{\boldsymbol{\beta}}, \mathbf{0}_{\boldsymbol{\theta}}, \hat{\mathbf{D}}\right) \right\|^2 \tag{3.9}$$

The 3D vertex loss in posed space is defined as

$$\mathcal{L}_{\mathcal{P}}^{3\mathrm{D}} = \sum_{f=0}^{F-1} \left\| C\left(\boldsymbol{\beta}, \boldsymbol{\theta}_{\mathrm{f}}, \mathbf{D}\right) - C\left(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}}_{\mathrm{f}}, \hat{\mathbf{D}}\right) \right\|^2 \tag{3.10}$$

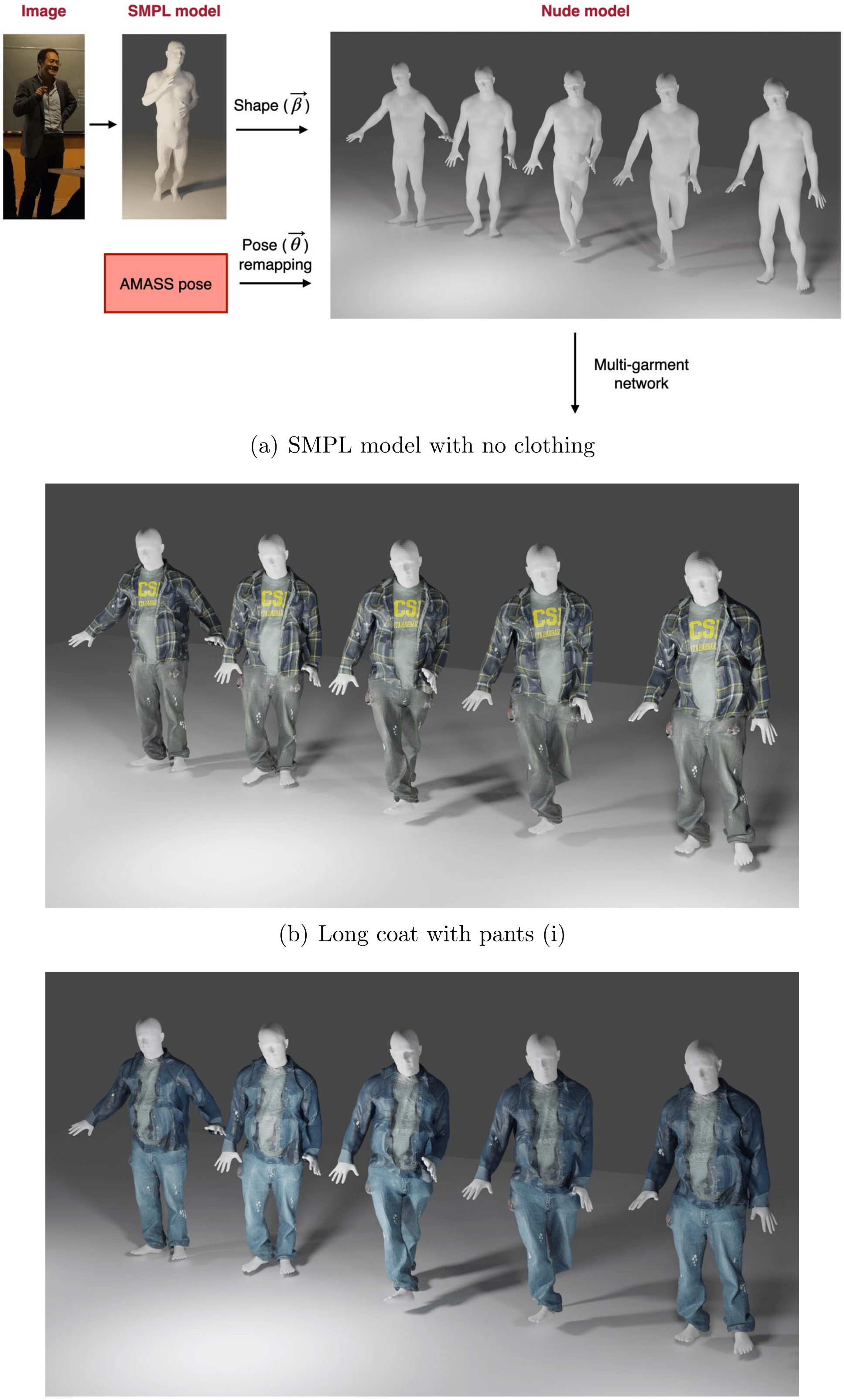
The 2D segmentation loss is not optimized with silhouette overlap but the projected segmentation mask against the input segmentation.

$$\mathcal{L}_{\mathrm{seg}}^{2\mathrm{D}} = \sum_{f=0}^{F-1} \left\| \mathbf{R}_{\mathrm{f}} - \mathbf{I}_{\mathrm{f}} \right\|^2 \tag{3.11}$$

The intermediate losses was also imposed on the pose, shape and garment parameter predictions to stabilize learning: $\boldsymbol{\mathcal{L}}_{\boldsymbol{\theta}} = \sum_{f=0}^{F-1} \left\| \hat{\boldsymbol{\theta}}_{\mathrm{f}} - \boldsymbol{\theta}_{\mathrm{f}} \right\|^2, \boldsymbol{\mathcal{L}}_{\boldsymbol{\beta}} = \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^2, \boldsymbol{\mathcal{L}}_{\boldsymbol{z}} = \sum_{g=0}^{L-1} \|\hat{z}^{\mathrm{g}} - z^{\mathrm{g}}\|^2$. Here $\hat{z}$ are the ground truth PCA garment parameters.

The base network worked as the CNN to map the dataset into the body shape, pose, and garment latent spaces. Each category of the garments could be trained in separate garment networks. Two branch was contained in the garment network. The first one predicted the mesh shape, and the second work added the high-frequency details.

Evaluation of the remapped garments of the digital wardrobe into different human body shape and poses

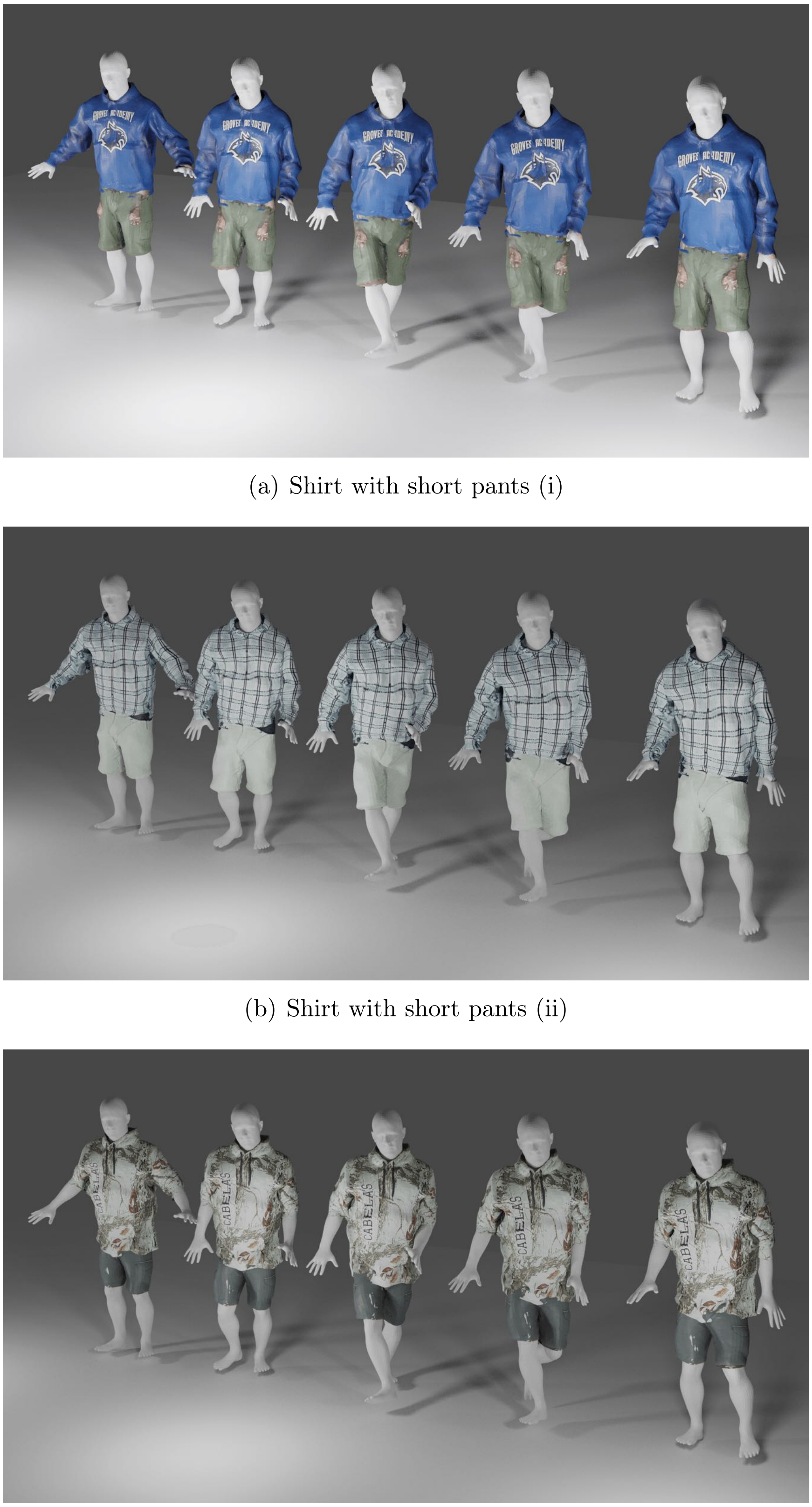


(a) SMPL model with no clothing

(b) Long coat with pants (i)

(c) Long coat with pants (ii)

Figure 3-2: Multi-garment networks samples (I). Garments from digital wardrobes remapped into SMPL models.



(a) Shirt with short pants (i)

(b) Shirt with short pants (ii)

(c) T-shirt with short pants

Figure 3-3: Multi-garment networks samples (II). Garments from digital wardrobes remapped into SMPL models.

3.3 Cloth simulation

A physical engine is the computer software that provides a realistic simulation of certain physical systems, e.g., rigid body dynamics, clothes, soft tissues, fluid dynamics, etc. The simulation in computer graphics is usually different from the one in engineering. The latter always requires extraordinary high accuracy, and the algorithm needs to be convergent in the finer mesh. However, the physical engine here does not need to achieve the best accuracy. However, the real-time speed is required, especially in the application of video games.

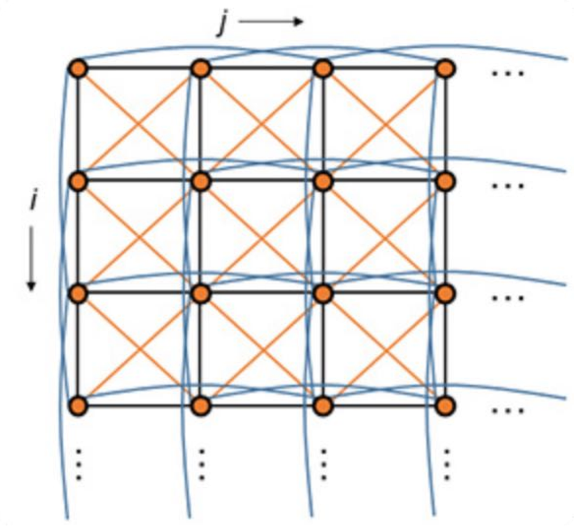


Figure 3-4: Illustration of the mass-spring system in the cloth simulation.

In the clothing simulation, we could simplify the meshes of cloth into the simple spring-mass system [40]. The cloth could be considered as a collection of particles interconnected with three types of springs:

- **Structural spring**: each particle $[i, j]$ is connected to four particles via structural connections: $[i, j+1], [i, j-1], [i+1, j], [i-1, j]$;
- **Shear spring**: each particle $[i, j]$ is connected to four particles via shear connections: $[i+1, j+1], [i+1, j-1], [i-1, j-1], [i-1, j+1]$;
- **Flexion spring**: each particle [i,j] is connected to four particles via flexion connections: $[i, j+1], [i, j-2], [i+2, j], [i-2, j]$

The force can be classified into types in the cloth simulation:

- **Spring force**: constrain the distance of each particle in the structural mesh;
- **Gravity force**: the major force to actively drag the cloth;
- **Damping force**: constrain the infinitesimal vibration of the mass particles;
- **Collision force**: constrain the self-penetration of the mesh and the penetration of the human body

To effectively animate the movement of the clothing, we utilize the extended position-based dynamics (XPBD) [30] method. The difference between the XPBD method and the traditional one is that there is no explicit contact force in the calculation. The constraints of position determine the trajectory of the particles.

Gauss' principle of least constraint

The principle of least constraint was enunciated by Carl Friedrich Gauss in 1829. It is a least-squares principle stating that the actual acceleration of a mechanical system of n masses is the minimum of the quantity

$$Z \overset{\text{def}}{=} \sum_{\mathrm{j}=1}^{n} m_{\mathrm{j}} \cdot \left| \ddot{\mathbf{r}}_{\mathrm{j}} - \frac{\mathbf{F}_{\mathrm{j}}}{m_{\mathrm{j}}} \right|^2 \tag{3.12}$$

where the jth particle hass mass m_{j}, position vector $\mathbf{r}_j$, and the non-constraint force $\mathbf{F}_{\mathrm{j}}$.

In the position-based dynamics method, let us assume the $\mathbf{p}^{\mathrm{t}}$ and $\mathbf{v}^{\mathrm{t}}$ is the location and velocity of the particle in time t, and Δt is a time step. In the next time, the location of this particle is

$$\mathbf{p}^{\mathrm{t}+\Delta\mathrm{t}} = \mathbf{p}^{\mathrm{t}} + \Delta t \left(\mathbf{v}^{\mathrm{t}} + \Delta t \mathbf{g}\right) + \Delta \mathbf{p} \tag{3.13}$$

and the velocity of this particle is

$$\mathbf{v}^{\mathrm{t}+\Delta\mathrm{t}} = \left(\mathbf{p}^{\mathrm{t}+\Delta\mathrm{t}} - \mathbf{p}^{\mathrm{t}}\right) / \Delta t = \mathbf{v}^{\mathrm{t}} + \Delta t \mathbf{g} + \Delta \mathbf{p} / \Delta t \tag{3.14}$$

As a result, the acceleration of this particle can be calculated as

$$\ddot{\mathbf{p}} = \left(\mathbf{v}^{\mathrm{t}+\Delta\mathrm{t}} - \mathbf{v}^{\mathrm{t}}\right)/\Delta t = \Delta\mathbf{p}/\Delta t^2 + \mathbf{g} \tag{3.15}$$

Let's use the Gauss's principle of least constraint to solve for the $\Delta\mathbf{p}$.

$$\arg\min_{\Delta\mathbf{p}} \sum_{\mathrm{a}\in\mathrm{A}} m_{\mathrm{a}} \left|\Delta\mathbf{p}_{\mathrm{a}}\right|^2 = \arg\min_{\Delta\mathbf{p}} \Delta\mathbf{p}^{\mathrm{T}}\mathbf{M}\Delta\mathbf{p} = \frac{1}{2}\arg\min_{\Delta\mathbf{p}} \Delta\mathbf{p}^{\mathrm{T}}\mathbf{M}\Delta\mathbf{p} \tag{3.16}$$

$$\text{subject to } \mathbf{C}(\mathbf{p}+\Delta\mathbf{p}) = 0 \tag{3.17}$$

This is a quadratic minimization problem, and the Lagrange multiplier could solve it. Let us assume there are M constraints, and the Lagrange multiplier is $\lambda \in \mathbb{R}^{\mathrm{M}}$, and the non-constrained function is

$$L(\Delta\mathbf{p}, \lambda) = \sum_{\mathrm{a}\in\mathrm{A}} m_{\mathrm{a}} \left|\Delta\mathbf{p}_{\mathrm{a}}\right|^2 + \lambda^T\mathbf{C} \tag{3.18}$$

To minimize the $L(\Delta\mathbf{p}, \lambda)$, we could get the derivative of L with $\Delta\mathbf{p}$ and λ.

$$\Delta\mathbf{p} = -\mathbf{M}^{-1}\nabla\mathbf{C}\lambda \tag{3.19}$$

Constraints.

The constraint $\mathbf{C}$ varies in different cases. In the cloth simulation, we used stretch constraint with the constraint function as

$$C_{\text{stretch}}\left(\mathbf{p}_1, \mathbf{p}_2\right) = \left|\mathbf{p}_1 - \mathbf{p}_2\right| - l_0 \tag{3.20}$$

For the bending constraint

$$\begin{aligned} &C_{\text{bend}}\left(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\right) = \\ &\operatorname{acos}\left(\frac{(\mathbf{p}_2-\mathbf{p}_1)\times(\mathbf{p}_3-\mathbf{p}_1)}{|(\mathbf{p}_2-\mathbf{p}_1)\times(\mathbf{p}_3-\mathbf{p}_1)|}\cdot\frac{(\mathbf{p}_2-\mathbf{p}_1)\times(\mathbf{p}_4-\mathbf{p}_1)}{|(\mathbf{p}_2-\mathbf{p}_1)\times(\mathbf{p}_4-\mathbf{p}_1)|}\right) - \varphi_0 \end{aligned} \tag{3.21}$$

Here φ_0 represents the initial dihedral angle between the two triangles.

We use the spatial hashing to find vertex triangle collisions [30]. If a vertex $\mathbf{q}$ penetrates the triangle $\mathbf{p_1}$, $\mathbf{p_2}$, $\mathbf{p_3}$, the self-collision constraint function is

$$C\left(\mathbf{q}, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\right) = \left(\mathbf{q} - \mathbf{p}_1\right) \cdot \frac{\left(\mathbf{p}_2 - \mathbf{p}_1\right) \times \left(\mathbf{p}_3 - \mathbf{p}_1\right)}{\left|\left(\mathbf{p}_2 - \mathbf{p}_1\right) \times \left(\mathbf{p}_3 - \mathbf{p}_1\right)\right|} - h \tag{3.22}$$

Here h is the cloth thickness.

In the position-based dynamics, we use the Taylor series expansion to simplify the nonlinear constraints. The position dynamics position method can be solved by the Sequential Quadratic Programming (SQP):

$$\min \frac{1}{2} \Delta \mathbf{x}^T \mathbf{M} \Delta \mathbf{x} \tag{3.23}$$

$$\text{subject to } \mathbf{J} \Delta \mathbf{x} = \mathbf{b} \tag{3.24}$$

Here $\mathbf{J} = \nabla C(x), \mathbf{b} = \left[-C_1, C_2, \cdots, -C_{\mathrm{m}}\right]^T$

As a result, the Lagrange multiplier can be calculated as

$$\left[\mathbf{J} \mathbf{M}^{-1} \mathbf{J}^{\mathrm{T}}\right] \lambda = \mathbf{b} \tag{3.25}$$

The solver used the Gauss-Seidel method to solve the equation iteratively. We could separately solve the Lagrange multiplier for each constraint of C_i and get the position deviation $\Delta \mathbf{p}$.

However, the traditional position-based dynamics solve the system in a quasi-static way without considering the kinetic energy. Besides, the material's stiffness is dependent on the time step, which is fatal in the simulation.

The control equation in the backward Euler method can be formulated as

$$\mathbf{M}\left(\frac{\mathbf{x}^{\mathrm{n}+1} - 2\mathbf{x}^n + \mathbf{x}^{n-1}}{\Delta t^2}\right) = -\nabla U^{\mathrm{T}}\left(\mathbf{x}^{n+1}\right) \tag{3.26}$$

The ∇U^{T} was used by the constrain $\mathbf{C} = \left[C_1(\mathbf{x}), C_2(\mathbf{x}), \cdots, C_{\mathrm{m}}(\mathbf{x})\right]^{\mathrm{T}}$

$$U(\mathbf{x}) = \frac{1}{2} \mathbf{C}(\mathbf{x})^{\mathrm{T}} \alpha^{-1} \mathbf{C}(\mathbf{x}) \tag{3.27}$$

Here α is the block diagonal compliance matrix. The force can be represented as

$$\mathbf{f}_{\text{elastic}} = -\nabla_{\mathbf{x}} U^{\text{T}} = -\nabla \mathbf{C}^{\text{T}} \alpha^{-1} \mathbf{C} \tag{3.28}$$

The original equation can be represented as

$$\begin{aligned} \mathbf{M}\left(\mathbf{x}^{n+1} - 2\mathbf{x}^{n} + \mathbf{x}^{n-1}\right) &= \Delta t^2 \mathbf{f}_{\text{elastic}} \qquad (3.29) \\ &= -\nabla \mathbf{C}^{\text{T}} \left(\frac{\alpha}{\Delta t^2}\right)^{-1} \mathbf{C} \qquad (3.30) \\ &= \nabla \mathbf{C}^{\text{T}} \lambda_{\text{elastic}} \qquad (3.31) \end{aligned}$$

Here $\lambda_{\text{elastic}} = -\tilde{\boldsymbol{\alpha}}^{-1}\mathbf{C}(\mathbf{x})$ is the Lagrange multiplier. The original equations are equivalent as

$$\begin{aligned} \mathbf{M}\left(\mathbf{x}^{n+1} - \tilde{\mathbf{x}}\right) - \nabla \mathbf{C}^{T}\left(\mathbf{x}^{n+1}\right)\lambda^{n+1} &= 0 \qquad (3.32) \\ \mathbf{C}\left(\mathbf{x}^{n+1}\right) + \tilde{\alpha}\lambda^{n+1} &= 0 \qquad (3.33) \end{aligned}$$

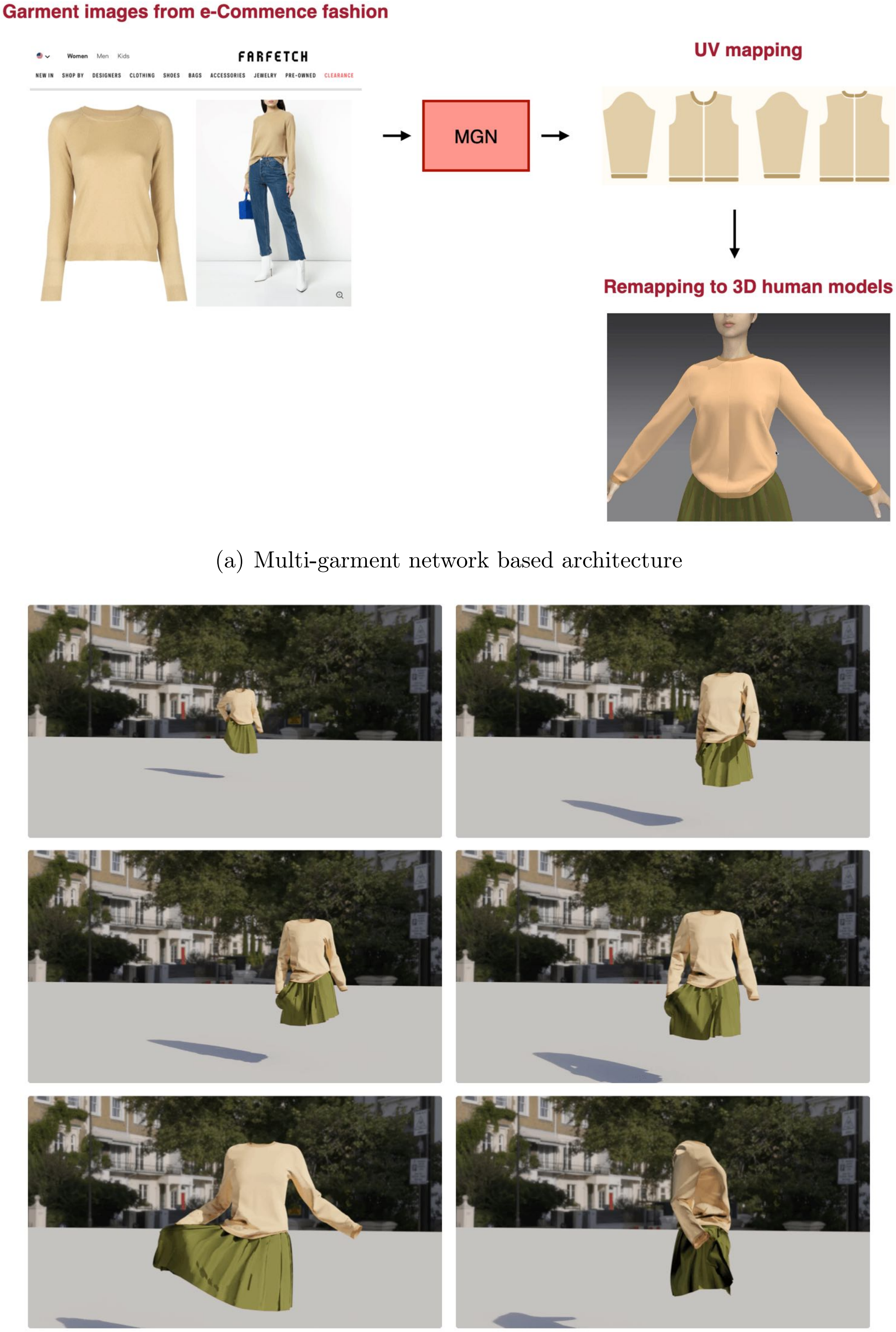
The Lagrange multiplier can be solved by the Schur complement of the mass matrix $\mathbf{M}$

$$\left[-\nabla \mathbf{C}\left(\mathbf{x}_{\text{i}}\right)\mathbf{M}^{-1} - \nabla \mathbf{C}^{\text{T}}\left(\mathbf{x}_{\text{i}}\right) + \tilde{\alpha}\right]\Delta\lambda = -\mathbf{C}\left(\mathbf{x}_{\text{i}}\right) - \tilde{\alpha}\lambda_{\text{i}} \tag{3.34}$$

and the position deviation $\Delta\mathbf{x}$ from the constraints is

$$\Delta\mathbf{x} = \mathbf{M}^{-1} - \nabla \mathbf{C}^{\text{T}}\left(\mathbf{x}_{\text{i}}\right)\Delta\lambda \tag{3.35}$$

The evaluation of clothing simulation utilizes the Marvelous Designer [14] to match the patterns and Blender cycle [5] for realistic rendering.


Garment images from e-Commence fashion
Women Men Kids
FARFETCH
NEW IN SHOP BY DESIGNERS CLOTHING SHOES BAGS ACCESSORIES JEWELRY PRE-OWNED CLEARANCE
MGN
UV mapping
Remapping to 3D human models

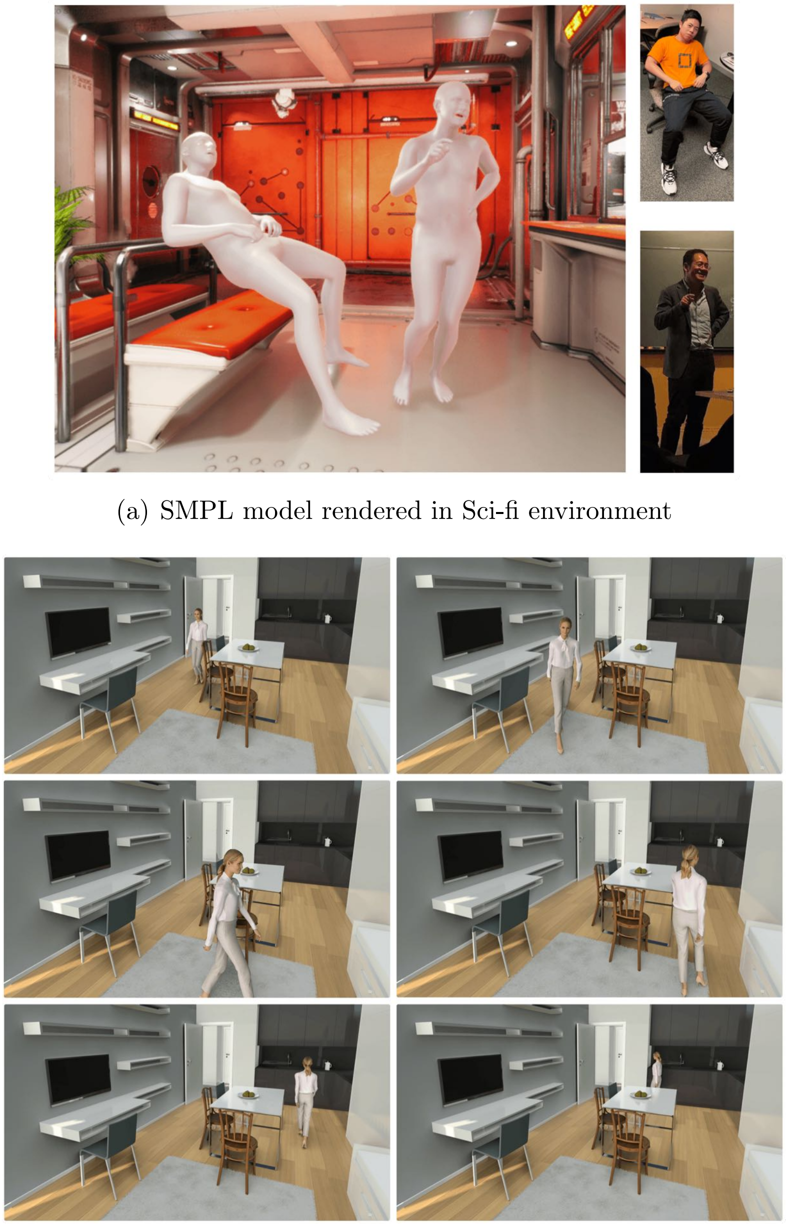
(a) Multi-garment network based architecture

(b) Cloth modeling with moving models inside(not visible)

Figure 3-5: The multi-garment network generates the UV mapping from the garments images on fashion e-Commence and remapped it into the 3D human model. The clothing was modeled as a mass-spring system.

3.4 Rendering system

Blender cycle [5] is used as the general rendering system in the thesis. It is an open-source, physically-based renderer module. The usage of Python API is conveniently to get flexible control of the environment, light setting, camera parameters, and rendering quality. Evaluation of different environments rendered with Blender cycle.



(a) SMPL model rendered in Sci-fi environment

(b) Human 3D model walks in virtual living room

Figure 3-6: Blender cycle rendering evaluation for different enviroments.

Chapter 4

Conclusion

3D reconstruction and modeling of humans from images is a central open problem in computer vision and graphics, yet remains a challenge using machine learning techniques. In this thesis, we propose a framework to generate a realistic 3D human with a single RGB image via machine learning. To conclude, we briefly summarize the main topic of each part of this thesis:

Skinned Multi-Person Linear Model(SMPL) is a generalized animated human body model to represent different shapes and poses. Usage of the end-to-end framework could input an image for the convolutional encoder ResNet-50. The regressor transfer the output of Resnet-50 into the predicted parameters $\vec{\beta}, \vec{\theta}, R, t, s$. The parameters are used to reconstructed the vertices by the SMPL model. The shape data $\vec{\beta}$ extracted from the end-to-end reconstruction was preserved, and the pose data $\vec{\theta}$ from the AMASS database was adapt on the SMPL model to animate 3D human body. The detected landmarks from facial images have been sent to the pre-trained morphable model, and the textures from original images had been merged into the computer-generated facial model.

The multi-garment net preprocessed the 3D scan data and registered the garments. The garments databases are categorized into five classes. For each category of garments, the $M_{\omega}^{\mathrm{g}}(.)$ was individually trained through latent code $l_{\mathcal{G}}$. The output as the un-posed garment G^{g} was computed through the major components of PCA plug the high-frequency deviation $D^{\mathrm{hf,g}}$. The digital garment could be added to the

SMPL human model with arbitrary shapes and poses. The clothing was treated as a mass-spring system in physical simulation. The extended position based dynamics algorithm was used to realize fast and realistic modeling.

Bibliography

[1] Ijaz Akhter and Michael J. Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1446–1455, Boston, MA, USA, June 2015. IEEE.

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693, Columbus, OH, USA, June 2014. IEEE.

[3] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-Garment Net: Learning to Dress 3D People from Images. *arXiv:1908.06903 [cs]*, August 2019.

[4] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 187–194, USA, July 1999. ACM Press/Addison-Wesley Publishing Co.

[5] blender. Cycles. https://www.cycles-renderer.org/.

[6] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. July 2016.

[7] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3794 –3801, Columbus, Ohio, USA, June 2014.

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

[9] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. November 2016.

[10] Young-Woon Cha, True Price, Zhen Wei, Xinran Lu, Nicholas Rewkowski, Rohan Chabra, Zihe Qin, Hyounghun Kim, Zhaoqi Su, Yebin Liu, Adrian Ilie, Andrei State, Zhenlin Xu, Jan-Michael Frahm, and Henry Fuchs. Towards Fully Mobile 3D Face, Body, and Environment Capture Using Only Head-worn Cameras. *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2993–3004, November 2018.

[11] Xiaowu Chen, Yu Guo, Bin Zhou, and Qinping Zhao. Deformable model for estimating clothed and naked human shapes from a single image. *The Visual Computer*, 29(11):1187–1196, November 2013.

[12] Xiaowu Chen, Bin Zhou, Feixiang Lu, Lin Wang, Lang Bi, and Ping Tan. Garment modeling with a depth camera. *ACM Transactions on Graphics*, 34(6):203:1–203:12, October 2015.

[13] Fernando De la Torre, Jessica Hodgins, Adam Bargteil, Xavier Martin, Justin Macey, Alex Collado, and Pep Beltran. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. 2009.

[14] Marvelous Designer. Marvelous Designer. https://www.marvelousdesigner.com/.

[15] FaceBuilder. FaceBuilder | KeenTools. https://keentools.io/facebuilder.

[16] Haoqiang Fan, Hao Su, and Leonidas Guibas. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. December 2016.

[17] Rohit Girdhar, David F. Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a Predictable and Generative Vector Representation for Objects. March 2016.

[18] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. June 2014.

[19] Yu Guo, Xiaowu Chen, Bin Zhou, and Qinping Zhao. Clothed and naked human shapes estimation from a single image. In *Proceedings of the First international conference on Computational Visual Media*, CVM'12, pages 43–50, Beijing, China, November 2012. Springer-Verlag.

[20] Ludovic Hoyet, Kenneth Ryall, Rachel McDonnell, and Carol O'Sullivan. Sleight of hand: perception of finger motion from reduced marker sets. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games - I3D '12*, page 79, Costa Mesa, California, 2012. ACM Press.

[21] Peng Huang, Margara Tejera, John Collomosse, and Adrian Hilton. Hybrid Skeletal-Surface Motion Graphs for Character Animation from 4D Performance Capture. *ACM Transactions on Graphics*, 34(2):1–14, March 2015.

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

[23] Sam Johnson and Mark Everingham. *JOHNSON, EVERINGHAM: CLUSTERED MODELS FOR HUMAN POSE ESTIMATION 1 Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation.*

[24] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end Recovery of Human Shape and Pose. *arXiv:1712.06584 [cs]*, June 2018.

[25] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J. Black, and Peter V. Gehler. Unite the People: Closing the Loop Between 3D and 2D Human Representations. January 2017.

[26] Matthew Lewis and Richard Parent. An Implicit Surface Prototype for Evolving Human Figure Geometry. page 10.

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr DollÃą̧r. Microsoft COCO: Common Objects in Context. May 2014.

[28] Matthew Loper, Naureen Mahmood, and Michael J. Black. MoSh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics (TOG)*, 33(6):1–13, 2014. Publisher: ACM New York, NY, USA.

[29] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):1–16, November 2015.

[30] Miles Macklin, Matthias MÃijller, and Nuttapong Chentanez. XPBD: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games - MIG '16*, pages 49–54, Burlingame, California, 2016. ACM Press.

[31] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of Motion Capture as Surface Shapes. April 2019.

[32] Christian Mandery, Omer Terlemez, Martin Do, Nikolaus Vahrenkamp, and Tamim Asfour. The KIT whole-body human motion database. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 329–336, Istanbul, Turkey, July 2015. IEEE.

[33] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. A simple yet effective baseline for 3d human pose estimation. May 2017.

[34] Daniel Maturana and Sebastian Scherer. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, Hamburg, Germany, September 2015. IEEE.

[35] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. May 2017.

[36] Meinard MÃijller, Andreas Baak, and Hans-Peter Seidel. Efficient and robust annotation of motion capture data. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 17–26, 2009.

[37] Meinard MÃijller, Tido RÃűder, Michael Clausen, Bernhard Eberhardt, BjÃűrn KrÃijger, and Andreas Weber. Documentation mocap database hdm05. 2007.

[38] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation. November 2015.

[39] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J. Black. ClothCap: seamless 4D clothing capture and retargeting. *ACM Transactions on Graphics*, 36(4):1–15, July 2017.

[40] Xavier Provot. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface*, pages 147–147. Canadian Information Processing Society, 1995.

[41] Courtney Reagan. A $260 billion 'ticking time bomb': The costly business of retail returns, December 2016. Library Catalog: www.cnbc.com Section: Holiday Central.

[42] Kathleen M. Robinette and Hein Daanen. Lessons Learned from Caesar: A 3-D Anthropometric Survey:. Technical report, Defense Technical Information Center, Fort Belvoir, VA, January 2003.

[43] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net: Localization-Classification-Regression for Human Pose. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1216–1224, Honolulu, HI, July 2017. IEEE.

[44] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.

[45] Nikolaus F. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of vision*, 2(5):2–2, 2002.

[46] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis. December 2017.

[47] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. DeepHuman: 3D Human Reconstruction from a Single Image. March 2019.

[48] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3D Human Pose Estimation in the Wild: a Weakly-supervised Approach. April 2017.