# 3D Animation of a Human Body Reconstructed from a Single Photograph

by

Yezhe Ding

Thesis submitted to the University of Ottawa
In partial fulfillment of the requirements
For the M.A.Sc degree in
Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

# Abstract

3D modelling is a technology in massive demand now and can potentially become a key factor for enabling subsequent technological evolutions such as metaverses, digital twins, and virtual reality. Current 3D modellings include high-precision 3D human body modelling and rapid modelling through single or multiple monocular photos. However, some problems persist in both modellings. The modelling based on high-precision equipment has low practicability, few applicable scenarios, and high cost. Modelling through monocular photos, on the other hand, has low accuracy and is sensitive to noisy data. And both modellings generate static 3D models. Therefore, to realize the model's dynamic effect in various fields while retaining fast modelling, we propose a system that recovers a 3D model from a single photo to fuse skeleton animation extracted from videos, for a realization of the Digital Twin (DT). DT is defined as “digital replications of living as well as non-living entities that enable data to be seamlessly transmitted between the physical and virtual worlds”.

Rigging is setting up the skeleton-based animation to combine the 3D model and skeleton animation. Traditional rigging method is time-consuming and non-reusable, since rigging is often done manually or semi-automatically. In this thesis, we propose an automatic rigging method to achieve a loose coupling fusion of one-to-many or many-to-one 3D models and skeletal animations. Our rigging method is fast and efficient, and only needs a single photo as input.

# Acknowledgments

As my three years of study in Canada come to an end, I have not only gained a deep understanding of my major but also met many brilliant people.

First, I would like to express my deepest gratitude to Professor Abdulmotaleb El Saddik, a rigorous, professional, enlightened and patient life coach. During the COVID-19 pandemic, when I was at my most challenging time, he provided me with the most significant support, guided my project and thesis remotely, and inspired me never to give up. Furthermore, he encouraged me to drill and discuss with my labmates to keep me on track and to make advancement of the project.

Second, I thank Professor Fedwa for guiding me to adapt into Canadian lifestyle and the lab environment. She encouraged me to be dedicated and perseverant and ready to rise to any challenge lying ahead of me. Her teaching style also inspired me to be more receptive to new knowledge and thus raising my confidence to the next level, particularly when communicating with people.

Next, I would like to thank Haopeng Wang and Xiaocong Ma sincerely. When I encountered a bottleneck in my project, they provided swift support which helped me in adjusting the direction of my project in time. I am grateful to have these hard-working and talented partners to accomplish this dissertation together.

Finally, I would like to thank my family for their support and financial help to my studies. They motivated me to keep going forward during my study abroad.

# Table of Contents

# List of Tables



# List of Figures

# List of Equations

# List of Abbreviations and Acronyms

| *Abbreviation* | *Definition* |
|---|---|
| **DT** | Digital Twin |
| **CAM** | Computer Aided Manufacturing |
| **CAD** | Computer Aided Design |
| **CT** | Computed Tomography |
| **TFLite** | TensorFlow Lite |
| **FPS** | Frame per second |
| **GPU** | Graphic Processing Unit |
| **FNN** | Feedforward neural network |
| **BVH** | BioVision Hierarchy |
| **SMPL** | Skinned Multi-Person Linear |
| **PCA** | Principal Component Analysis |
| **HMR** | Human Body Mesh Recovery |
| **PIFu** | Pixel-Aligned Implicit Function |
| **PIFuHD** | Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization |
| **INR** | Implicit Neural Representation |
| **CPU** | Central Processing Unit |
| **IMU** | Inertial Measurement Unit |
| **JSON** | JavaScript Object Notation |
| **CD** | Chamfer distance |
| **P2S** | Point-to-surface distance |
| **NSRM** | Normalized Signed Rotation Matric |
| **IK** | Inverse Kinematics |
| **FK** | Forward Kinematics |
| **API** | Application Programming Interface |
| **RMSE** | Root Mean Square Error |

# Chapter 1. Introduction

## 1.1. Background, Problem Statement and Motivation

The concept of meta-universe materialized, the 5G era officially arrived, and COVID-19 is still not over, all of which means an amount of information transmission dependent on the Internet Technology. As cutting-edge technologies such as artificial intelligence and VR technology take hold, the volume of data and the transmission power of information are enhanced as never before. Character reconstruction technologies will change people's original 2D video streaming communication through devices such as computers and cell phones to 3D streaming communication that can be interacted with in real-time. The Digital Twin (DT) in Figure 1.1, is defined as a digital replica of living and non-living beings that enables data to be transferred seamlessly through the physical and virtual world, allowing real-time 3D communication and multidimensional communication to be established in between[1]. The current study reconstructs and dynamizes a 3D model of the human body from a single photograph, which represents a rapid modelling approach for the DT. This approach is an integrated multidisciplinary crossover result involving integrated disciplines such as image processing, deep learning, neural networks and computer vision and thus has significant practical commercial and research values.

Figure 1.1: Digital Twin[1]

[1] El Saddik, A. (2018). Digital Twins: The Convergence of Multimedia Technologies. IEEE MultiMedia, 25(2), 87-92

In terms of commercial value, reconstructing the 3D human body structure can potentially bring revolutionary changes in various industries. Figure 1.2 presents that 3D reconstruction is widely applied in typical areas like games, virtual fitting, medical and others. For example, researchers quickly obtain 3D data models and size data of a human body through human body 3D scanning. The acquired data is then imported into professional Computer Aided Manufacturing (CAM) and Computer Aided Design (CAD) apparel design and layout software, which simulates body shape adjustment, model selection, colour matching, size modification and fitting[2]. This process digitizes well-fitting, comfortable, and fashionable garments which are customizable online to meet customers' personal preferences. In addition, human body 3D reconstruction technology is also widely available in video gaming industry. Instead of creating digitized apparel, researchers in this category focus on reconstructing various digitized human body poses by utilizing scanned and pre-imported data[3]. The acquired data is transferred into professional animation design tools for enabling further 3D graphical editing, creation, and design in the film and television industry, particularly when reconstructing animated virtual characters[4]. More importantly, 3D reconstruction can reduce patients' risks in precision medicine. The current Computed Tomography (CT) 3D reconstruction increases the accuracy of cranial, oral and cardiovascular disease diagnosis by improving disease localization and qualitative diagnosis[5]. Specifically, the method in 3D reconstruction of the human body proposed in this thesis has the potential to realize remote diagnosis, even for communicating complex transnational diseases.


Figure 1.2: 3D Reconstruction Application

In terms of research value, recent work has made significant progress on estimating the major joints and rough pose in 3D directly from a single 2D image. However, we need the entire 3D surface of the body, hands, and face to analyze the human behaviour. The traditional methods like Kinect (seen in Figure 1.3) with complex flags and stickers are inconvenient and not portable. Moreover, the accuracy of the results obtained from these devices also depend heavily on the surrounding environment. On top of these, for a device with depth sensors like Kinect, data collection usually costs a large amount of time. Thus, in order to achieve a cost-effective solution, recovering a 3D model accurately and quickly from a single image is essential. On the other hand, only a few recent systems can provide reasonable solutions due to several significant challenges, including the lack of appropriate 3D models and 3D training data. In this thesis, we use existing techniques such as PIFuHD[6] and MocapNET[7] to obtain a 3D model of a human body and a human motion sequence, and then fuse the two together to generate a dynamic 3D human body.

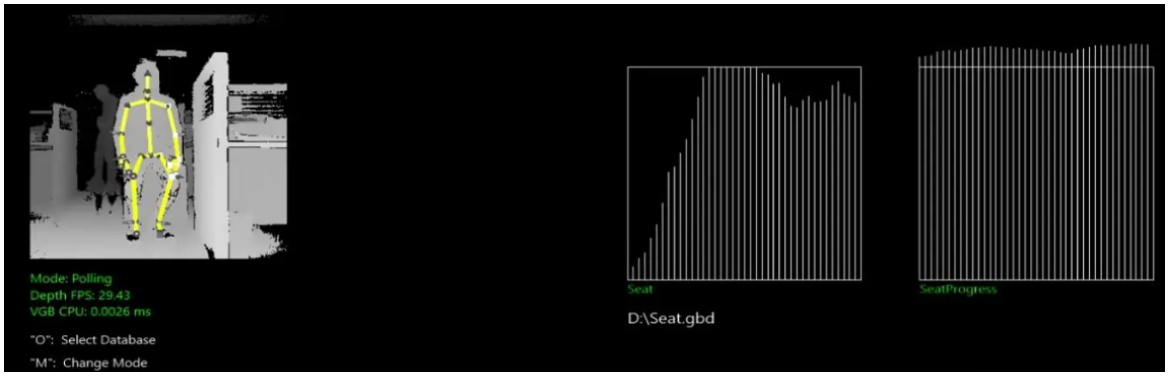


Figure 1.3: Kinect[2] Example

To fuse the results of 3D reconstruction with human motion, we use the method of rigging. Traditional manual rigging method has high precision and strong applicability, but requires the operator to have certain proficiency and experience[8]. Manual rigging is time-consuming and non-reusable. In this case, a manual rigging process is required for each newly extracted 3D human model and a new set of motions. In this research, it's impractical due to many combinations of 3D human models and skeleton animation. Therefore, we design an automatic rigging algorithm to achieve fast and highly adaptable 3D animation

[2] https://blog.csdn.net/zdt_zdh/article/details/90474114

output. In addition, during the process of 3D recovery, we improve the accuracy of 3D human by adding image matting to PIFuHD[6].

## 1.2. Challenge

Extracting a 3D model from a single photo has some certain limitations. For example, it is imperative but difficult to choose a suitable method to restore the 3D model of the human body from a single photo. This is because the quality of the results obtained from a single photo can heritably affects the subsequent experimental results.

For different methods, more research and comparison need to be conducted, including the comparison and experimentation of the training set data, the observation and comparison of the results, and the analysis of the quantitative effect. However, in terms of dataset, the training dataset used in this thesis contains many natural human poses. Therefore, when 3D restoration of the human body is performed on the standard T-pose photos, the result is inferior and missing 3D points when imported. Nevertheless, for some natural poses, it can restore more clothing details, which is better than other methods. In this study, we adopt A-pose as a compromise strategy, which will be descript in detail in Section 3.2.2.

As we try to further process the extracted 3D model, we encountered more challenges when choosing a right processing tool for the 3D file. Currently, there are several popular 3D processing software on the market, such as Blender[9], Maya[10], Cinema 4D[11], 3ds Max[12], ZBrush[13], Houdini[14], Rhino[15], and Modo[16]. All require a very long learning process. When rigging 3D models, the results of automatic rigging are inferior, and many 3D models need to be bound manually. This work is very labor-intensive and time-consuming, and manual operations are prone to errors, which affect the effect of 3D animation. When designing the algorithm for automatic rigging, it is also challenging to define the parent and child bone coordinate system together with the surface vertex class and then calculate the motion in the next frame using bone matrix transform.

In the early days of this research, the original design of our system has an intention to be portable by integrating the whole system on a mobile device. However, due to the

hardware restrictions, a mobile version of our solution is not likely to be achievable at the current stage. Specifically, when testing the human 3D recovery methods, the time to recover a 3D human from a small image with a resolution of 300 × 300 is at least 90 seconds. The biggest problem is that the model produces more than 5000 space points with three coordinates, which takes too long to calculate. And this is a necessary process that cannot be simplified. In addition, when converting the pre-trained model to a mobile device model in TensorFlow Lite (TFLite)[17] format, the large sized models exceed the maximum capacity of 'TFLite' models on mobile devices. Therefore, this thesis mainly focuses on designing a system of dynamized 3D human body to realize the dynamic effect of the model in various domains while maintaining fast modelling.

## 1.3. Thesis Objectives

This research aims to achieve fast, accurate and multi-detailed simultaneous DT modelling in the context of a metaverse. Mainly, we demonstrate our objectives in following steps.

1. **To use Background Matting to achieve noise cleaning:** The result of extracting the foreground human body from still image species is the first step of our study that the rest of the research depends on. This study promptly uses Background Matting, a real-time, high-resolution background elimination technique that is compatible to run at 4K resolution with both 30 and 60 frames per second (fps) settings on contemporary Graphic Processing Units (GPUs)[18]. This compatibility has performance advantages, particularly when the background of the human body in various scenes in realistic environments is relatively complex, and traditional extinction methods cannot be captured at high resolution in the real-time and thus manual input is usually required for completing the extraction process. Previously, this situation can lead to difficulties foregrounding human extraction and inaccurate results[19]. Due to this reason, our study utilizes a pre-processing of the original data at high resolution to aid the recovery of the static model of the human body and the reconstruction of the human motion pose at later stages. By applying

Background Matting for the noise removal, both the quality of resolution and the speed of processing can be significantly improved.

2. **To reconstruct a static 3D human model from a foreground human body using a PIFuHD's pretrained model**: Because of the regulations and restrictions imposed by the COVID-19 pandemic, it is difficult to carry out multi-camera high-precision 3D modeling of the human body such as Kinect in our laboratory environment. Hence, this thesis is initially aims to extract a static 3D model from a single photo instead. Compared to the reconstruction utilizing laboratory devices, the accuracy of this approach is low, but the data collection process is convenient and fast. The reconstruction of a high-precision static 3D human structure requires attention to details on the subject pose, particularly when processing the multidimensional recognition on the head and limb end pose. Among the same type of 3D reconstruction methods, PIFuHD is capable of high-detail 3D reconstruction with accurate recognition of detailed information such as fingers, facial features, and clothing folds[6].
3. **To extract motion pose sequences using MocapNET**: The prerequisite for accomplishing pose sequence recognition is motion capture, measuring, tracking, and recording the motion trajectory of an object in 3D space. To meet this demand of our target system, we utilize the MocapNET which trains a feedforward neural network (FNN) that regresses directly from two-bit joint rotations[7]. It also proposes Normalized Signed Distance Matrices that combine 3D pose with 2D joint point marker correspondence to achieve 3D pose reconstruction based on 2D human limb detection[20]. As a result, compared to traditional pose recognition methods, MocapNET performs faster and is the first to provide direct BioVision Hierarchy (BVH[3]) output of 2D points in an end-to-end neural network[7]. On top of that, the reconstructed motion pose sequences are output as skeletal animations in the form of BVH which can be easily integrated with 3D static human bodies.
4. **To design an auto-rigging method to generate 3D animation**: In order to make a static 3D model character moveable, we need to set up bones for this model. This process is called rigging, which is a crucial link between a static 3D mannequin and

[3] https://research.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html

a dynamic 3D mannequin. This research uses masking as an essential driving link to create continuous movements on the mannequin, such as walking, running and dancing. It is noteworthy that, the accuracy of this work has a direct impact on the next step of motion production. In this study, we apply Blender, an open-source lightweight 3D animation software, to complete the dynamic transformation of the static model.

## 1.4. Thesis Contributions

The major contributions of the research are listed as the following:

- Improvement of accuracy of 3D reconstruction from a single image using image matting.
- Design and development of a system integrated from multiple processing stages to obtain 3D animated model from a single 2D image and a human motion video.
- Design and development of an algorithm based on automatic rigging to apply human motion sequences to the static 3D human model and obtain 3D animations of the model.

## 1.5. Practical Application

As the 3D human dynamic reconstruction is further refined, it plays a vital role in many complex real and virtual-world scenarios, such as the applications of the DT. The present research results build a framework for the DT, which is the basis for various applications bridging the real and virtual-worlds. On a deeper level, dynamic reconstruction can increase interactive communication online through the transformations of the users' body movements. It can also increase the sense of interaction in the metaverse era, where the spatial distance between each other is felt through the grid faces on the screen, not just a mere conference call. Besides, dynamic reconstruction can bring the real emotions of life into virtual gaming world, and thus, creating an immersive gaming experience by interacting virtually with the surrounding environment or other users.

## 1.6. Thesis Organization

The content of this thesis is organized as the follows:

- In Chapter 1, we provide an introduction that explains the background, motivation, objective, and contribution of this research.
- In Chapter 2, we conduct a literature review on the previous works that are related within the scope of our research.
- In Chapter 3, we present an overview on the overall system architecture and we present system preprocessing stage.
- In Chapter 4, we present the 3D human reconstruction and motion capture.
- In Chapter 5, we present the auto rigging method and animation copy.
- A conclusion and a final discussion on the potential of our work are included in Chapter 6.

## 1.7. Scholarly Achievements

In the process of completing this work, the following publications have been submitted, accepted or published:

1. Laamarti F, Badawi H F, Ding Y, et al. An ISO/IEEE 11073 standardized digital twin framework for health and well-being in smart cities[J]. IEEE Access, 2020, 8: 105950-105961.
2. Gámez Díaz R, Yu Q, Ding Y, et al. Digital twin coaching for physical activities: A survey[J]. Sensors, 2020, 20(20): 5936.
3. Ding Y, Laamarti F, Wang H, El Saddik A. Auto-Rigging and 3D Animation of a Human Body Reconstructed from a Single Photograph. In progress.

# Chapter 2. Related Work

In recent academic research, recovering 3D models of the human body based on 2D monocular photographs are not a new concept. Compared with 2D colour images, 3D mannequins can vividly show the state of a person from all angles. Based on the theory of 3D human model reconstruction from 2D colour images and the Skinned Multi-Person Linear (SMPL) model, this thesis surveys various studies on 3D human structure reconstruction from photographs in recent years. In order to obtain more comprehensive and high-precision reconstruction results, this study borrows a pre-training PIFuHD model. In the pose recognition phase, according to recent advances in monocular 2D and 3D human pose estimation, we survey the single human pose recognition from a video composed of a pose encoder and a pose decoder. The former extracts the backbone of the target person, while the latter generates 2D positions of key points based on regression or detection techniques. In addition, this thesis realizes 3D animation fundamentals by designing an auto-rigging method to fuse static 3D models with skeletal animations to produce new 3D animations.

## 2.1. Work on Image Matting

Matting is a method for removing arbitrarily formed foreground elements from a photograph. Through the development of digital matting technology, it is possible to recover high-quality and detailed foreground in a picture, and thus, enhancing the virtual reality system's realism by generating an intense sensation. Deep learning-based matting is separated into two primary areas.

Specifically, the deep learning-based matting can be separated into two primary areas.

1. Supplementary input - in addition to the original picture and the annotated image, the prediction requires inputting multi-dimensional data. Trimap, a dimensionality reduction method that uses triplet constraints to form a low-

dimensional embedding of a set of points[21], is frequently used to separate a picture into three sections — foreground, background, and surplus area. On top of this, additional background and interaction information are also commonly utilized as further auxiliary information. Anat Levin et al.[22] introduced a technique for unsupervised computation of fuzzy matting components that simplifies user input with matting options and mouse guidance. Eduardo S. L. Gastal et al. [23] proposed a method that shares similar attributes of adjacent pixels to improve the speed of processing pixel Trimap. In contrast to the pixel optimization approach, Jian Sun et al. [24] operated directly on the gradient of the mask by interactively using filtering to reduce the error caused by the misclassification of colour samples. However, these previous methods all require manual input and thus cannot be run in real-time at high resolution[18].

2. Alpha prediction - this is achieved directly without relying on any auxiliary information. Bingke Zhu et al. [25] designed a system for real-time video matting on cell phones with accurate adaptive matting capability on the head and neck parts. However, it could not distinguish the details in the hair region. Normally, a key challenge of this type of background-distinguishing problem is the lack of accurate ground-truth data. Soumyadip Sengupta et al.[26] used an adversarial network to recover alpha masks and foreground colours to address this problem.

In the process of the static model recovery and motion posture reconstruction on a human body, it is essential that our research can obtain high-resolution results from the beginning. Shanchuan Lin et al.[18] introduced a real-time unified matting architecture that produces sound quality results on 4K video at 30fps and HD video at 60fps. This method would give an ideal performance on noise removal, and thus become our choice for the pre-processing stage.

## 2.2. Work on Reconstruction of the 3D Human Body

In today's era of information transmission explosion, two-dimensional image display has certain limitations on displayable presentation, while the 3D form and shape of the human body can include more vivid information about a person when participating in the

real-time interaction. The current methods of 3D human reconstruction can be summarized into two categories based on their methodologies used. One is to use the existing human data model to recover the human 3D model directly from a single RGB picture or video, which is called model matching[27]. The other is to use depth sensors to collect depth information directly and then construct a complete model by stitching, which is called fusion way[28]. This thesis extracts the human body structure from a single photo; therefore, we mainly adopt the model matching method.

### 2.2.1. The Ideological Basis of 3D Reconstruction

The mainstream method is to use a standard parametric human model. The parameters of the 3D human model are first obtained from 2D colour images. Then the parameters are input to a standard 3D human template to compare the changes with default values for calculating a 3D human structure with a specific pose and body shape[29]. Nikos Kolotouros et al.[30] proposed two main paradigms for model-based human pose recognition. One is based on an iterative approach, obtaining more accurate but slower initialized images[30]. The other is a regression approach, which does not pursue accurate pixel results but tends to provide reasonable results[30]. In other words, both iterative fitting optimization and regression approaches based on these two ideas have advantages and disadvantages, which motivates us to use them in a way for achieving an optimal combination.

Iterative fitting optimization mainly refers to extracting the joint point information of the desired human structure from a single photograph based on changing the pose and body shape of the template. By continuously adjusting the parameters of the model so that the joint point information in the projection of the two-dimensional plane has the slightest error with the information in the photograph, a relatively accurate three-dimensional human structure can be obtained [31]. On the other hand, regression refers to the direct regression of parametric information and the matching of body shape and pose. Once pose information of a 3D human structure is obtained from a single photograph, it is gradually applied to a standard 3D human model template for modifying its pose and body shape. The goal is to reach the best body pose consistency between the original photograph and

the modified 3D template[32]. As shown in Figure 2.1, Christoph Lassner et al. relied on 91 predicted landmarks in 2D for the fitting process of the 3D human pose estimation, providing a good hint for estimating the human body[33]. Also, they proposed a regression tree model to predict 3D human structure directly from 2D key points, which is an approach that runs orders of magnitude faster than optimization-based approaches[33]. In addition, Hayato Onizuka et al.[34] proposed a new CNN-PCN-based hourglass network for end-to-end 3D human shape regression from monochrome images, which can better reconstruct people wearing loose clothing in a single RGB image.

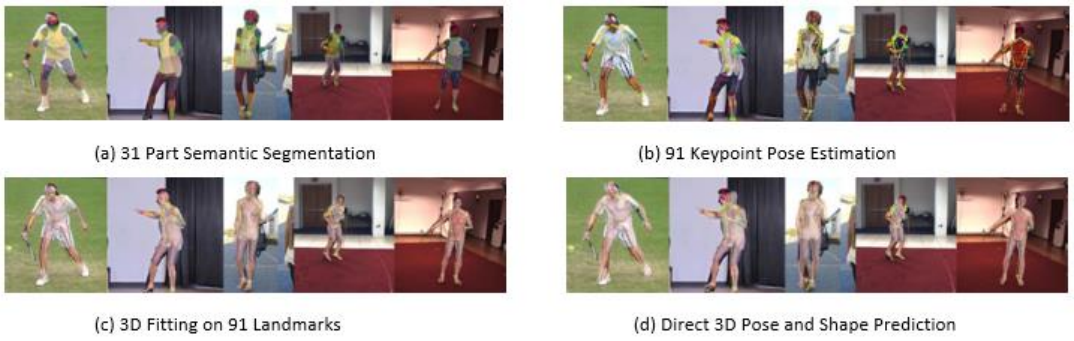


Figure 2.1: Results from Various Methods Generated from the UP-3D Dataset[33]

### 2.2.2. SMPL Related Models

Skinned Multi-Person Linear stands for SMPL, which was proposed by Matthew Loper et al.[35] to comprehensively depict the human body structure for achieving an accurate animation foundation. The primary method used here is to comprehend the human body as a base model then to form total deformations on top of it. Next, the Principal Component Analysis (PCA)[36] is conducted based on the deformation to determine the parameters used for carving the shape. And PCA uses motion tree to represent the rotational relationship between each joint node and its parent node, which can be expressed as a three-dimensional vector. More specifically, the SMPL employs a vertex-based skinning approach with N=6890 vertices and K=23 joints; its fundamental idea is shown in Figure 2.2.

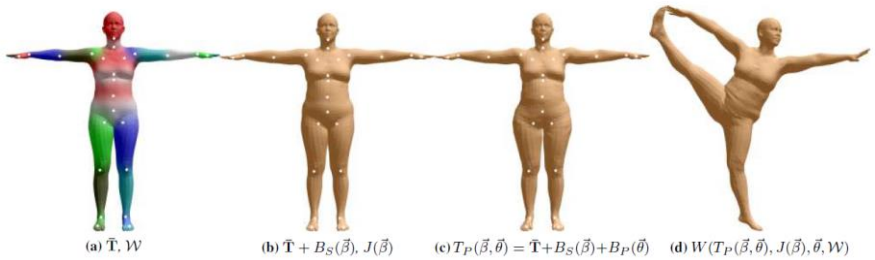


Figure 2.2: SMPL Model[35]

(a) Template mesh with colored blend weights and white-outlined joints. (b) Vertex and joint positions are linear in the shape vector; only identity-driven blend shape contribution is used. (c) In order to prepare for the split posture, pose blend shapes are added; see how the hips have expanded. (d) Dual quaternion skinning was used to reposition deformed vertices for the split posture.

Based on the SMPL model, Georgios Pavlakos et al. [37] presented the SMPL-X model for a more realistic human body structure. It may depict various forms and postures of the 3D human body by merging the status parameters for body shape and stance. Due to the open-source nature of SMPL and SMPL-X models and the widespread use of 3D reconstruction, these models are fast gaining popularity quickly in academia and have reached a number of additional milestones in research. Based on the SMPL model, Angjoo Kanazawa et al. [32] presented a new Human body Mesh Recovery (HMR) method, a framework which is capable of reconstructing a full 3D mesh of the human body from a single RGB photograph. This work uses the HMR to compare reconstructed 3D models, as described in Chapter 4. Nikos Kolotouros et al. [30] suggested SPIN (SMPL oPtimization IN the loop) employing a deep network to regress the SMPL parameters as a self-improving approach when training neural networks for estimating the 3D human position and form. In addition, Muhammed Kocabas et al. presented Video Inference for Body Posture and Shape Estimation (VIBE)[38], which outputs a collection of pose and shape parameters in SMPL body model format. It employs a large-scale motion capture dataset called AMASS[39] with unpaired 2D key point field annotations. In other words, it is an adversarial network that employs AMASS to differentiate between human motion and motion created by our temporal pose and shape regression network [38].

### 2.2.3. PIFu and PIFuHD

Real-life photographs of human bodies often depict them in varied settings, with various attire and hairstyles. Shunsuke Saito et al. developed the Pixel-aligned Implicit Function (PIFu) to digitize clothed humans and derive 3D surfaces and textures from a single picture. The PIFu is an Implicit Neural Representation(INR) capable of completely convolving to align the pixels of a 2D picture with the global context of the matching 3D object [40]. The concept is to train an encoder to learn distinct feature vectors for each picture pixel, taking the global context relative to its position into account. The feature vectors spatially connect the global 3D surface form to its pixel, preserving the local features in the input picture while inferring reasonable details in the unseen areas.

With current technology, the PIFu can only process shallow fractional input pictures (resolution < 512 × 512), limiting the accuracy of the output model with a constrained level of details. This work uses high fractional rate photos (resolution > 1024 × 1024) to examine the correctness of the outcomes. Therefore, we applied PIFuHD [6], which uses picture samples to feed a PIFu base layer with fewer details. Then, a new independent network is introduced to provide the entire resolution surface with fine detail. These procedures are carried out in two primary phases. Initially, the model is trained at a lower resolution to concentrate on global inference, which may encompass a broader area of the image's contextual context. Next, using this information, the model then calculates precise geometry information of a person based on the picture and the high resolution of the first output. By downsampling the picture and sending it into the PIFu model, the coarse layer is able to capture the overall 3D structure. Iteratively, the initial 3D outputs are used as high-resolution inputs in a comparable lightweight PIFu network for adding high-resolution details. As seen in Figure 2.3 [6], multilayer PIFu produces high-resolution 3D models, while single-layer PIFu enables the rapid development of precise models. The PIFuHD has become more relevant to real-time streaming in recent years. Si-han Xu utilized the PIFuHD to transform 2D characters to 3D, conducted single-camera motion capture, and applied artificial intelligence algorithms to a virtual YouTuber [41].

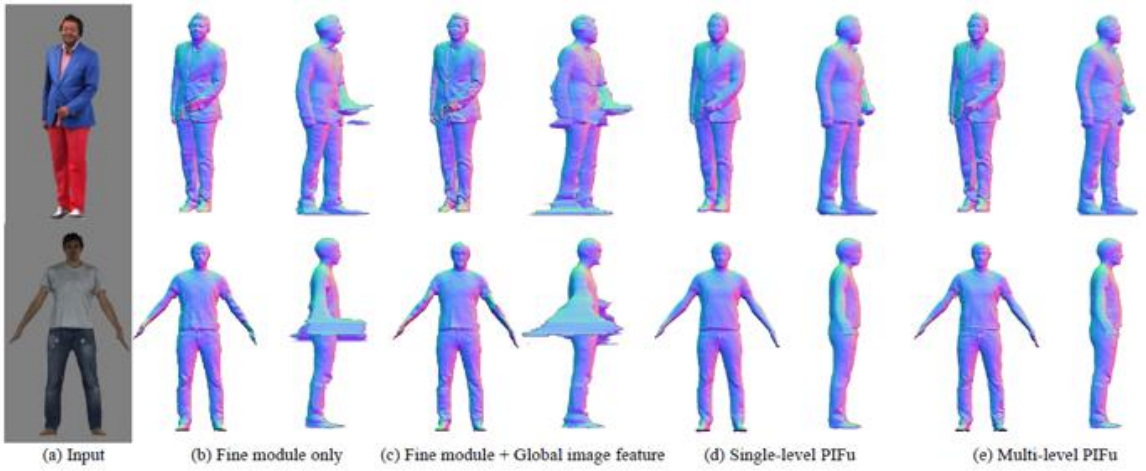


Figure 2.3: Qualitative Evaluation from RenderPeople and BUFF[6]

## 2.3. Work on Pose Estimation and Motion Capture

To extract motion trajectories accurately, comprehensively and naturally, this thesis combines both posture recognition and motion capture technologies. Pose estimation refers to the process of recognizing and analyzing human body poses in a static image or video at a certain moment[42]. Meanwhile, motion capture aims to track the motion trajectory of an object or human body, and also quantifies and records its motion parameters such as angle, speed, acceleration, etc.[43].

Diaz et al.[44] tracks the pose from a 2D video. It generates keypoints from the trainee-coach videos to measure, monitor, and record trainee physical exercises. Yu et al.[45] propose a multitask system to estimate human pose, identify physical activities and count repeated motions via 2D pose estimation.

Based on 2D pose estimation, extra depth information is predicted in the work of 3D pose estimation. Because of contextual complexity rows and fiber variations in true realistic circumstances, it is still challenging to properly discern the placement of human limbs from RGB-only images. Currently, 3D human pose estimation can be divided into two approaches: model-free methods and model-based methods. Model-based methods typically use a parametric human model or template to estimate the human pose and shape from an image[46]. In Tan et al.'s approach[47], the decoder was trained to use SMPL

parameters as input for predicting body silhouettes. Subsequently, the decoder was kept fixed during training on real images and their corresponding silhouettes. In addition, using adversarial learning, Kanazawa et al. employed a generator to predict SMPL parameters and a discriminator to distinguish between the predicted SMPL model and the real one[32].

Model-free methods do not rely on a human model as either the predicted output or as an intermediary signal[46]. Li and Chan [48] used deep convolutional neural networks to learn a pose joint regressor and a sliding-window body part detector, directly regressing 3D joint coordinates. Tekin et al. [49] considered the interdependence between human joints and pre-trained an unsupervised auto-encoder to learn a high-dimensional latent pose, which was then used to predict the structure of the human body. The MocapNET directly encodes 2D poses into 3D BVH. It is integrated with OpenPose[50], enabling real-time estimation and rendering of 3D human poses using only Central Processing Unit (CPU) processing[7].

Motion capture base on Inertial Measurement Unit (IMU) is a measurement of human movement and posture using sensors, with a focus on initial alignment to improve the precision of system measurements[51]. Glen Cooper et al. [52]utilized external camera devices for initial alignment, installing markers on key parts of the human body and using external cameras to capture the positions of these points and complete the initial alignment. X L Meng et al. [53]used a standing still static posture for initial alignment, completing the alignment through measuring the data of the gyro and acceleration at the initial pose. Compared to the motion capture methods based on IMU, MocapNET does not require a large amount of data for training and is able to quickly and accurately capture human motion[7]. It is a "deep virtual sensor" technology that converts RGB images into information similar to depth sensors, thereby capturing more accurate human motion trajectories.

## 2.4. Work on Rigging

This research investigates the fundamentals of 3D animation, with a focus on rigging to complete the shift from static 3D model to dynamic 3D animation.

As Figure 2.4 shows, rigging (also known as skeleton binding) process is setting up the skeleton-based animation to combine the 3D model and skeleton animation, to generate animated 3D animation. The principle of rigging is the use of virtual controllers to bind at the character's joint bones or at the appropriate position on the character. It can be programmed to do actions such as reaching, running, and leaping. Automatic rigging is a method for creating and assigning weights to skeletons without human interaction.

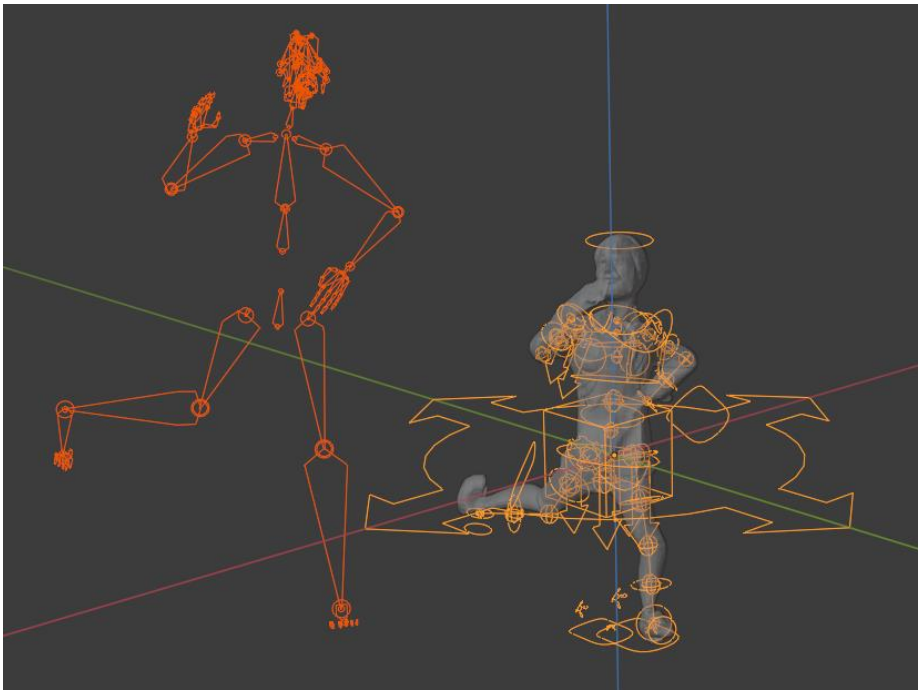

Figure 2.4: Rigging Function

Pinocchio that is presented by Ilya Baran et al. is a child-friendly animation system [8]. It is a skeleton-adaptive character technique that animates the character based on skeletal motion data. In addition, if the cumbersome limbs were too thin, the produced rig might cause the connection between the legs to break, resulting in a decrease in performance. If the knees and elbows of the target human body are misaligned, there is no way to correct the issue. Pan et al. [54] established geometric entities and developed a 3D Silhouette method. During extraction, this approach does not alter the connectivity of the mesh to retain the original model's topology. 3D Silhouette is able to cope with the majority of characters. But it has trouble recognizing asymmetric forms, which may result in frames

being placed outside of the center of the castings. Binh Huy Le et al. [55] provide a robust approach for creating high-quality armatures, sufficient to decrease the film and gaming industries cost. But it can only prune a small number of bones at a time, and thus its iterative rigging may need to be repeated many times leading to long running time. Andrew Feng et al. [56]demonstrate a method to automatically rig a 3D mesh by matching a set of morphable models against the 3D scan. But their results are limited by the shape space spanned by models in the database. The model fitting may be inaccurate if the scanned human is wearing excessive clothing. Zeeshan Bhati et al. [57] introduce widgets to achieve automatic rigging for quadruped characters with custom controls and manipulators for animation. The biggest limitation of their work is the need of manual work including adjusting the widgets according to the size and shape of its quadruped character.

In this research, software for 3D processing is employed for rigging. Blender[9], Maya[10], Cinema 4D[11], 3ds Max[12], ZBrush[13], Houdini[14], Rhino[15], and Modo[16] are some of the most popular 3D processing software on the market today. Blender (1994) and Autodesk Maya (1998) now offer their own programs and tool extensions, and Rigify is one of the fast and accurate plugins that will be explained in detail in Chapter 5.

The last step after rigging is to conceal the frames. Skinning indicates that the vertices of the mesh are connected to the bones, and that each vertex can be controlled by several bones. So that vertices at joints can remove fissures by changing position as they are dragged by both parent and child bones. Skin information controls how vertices are attached to bones. For each bone, a BoneOffsetMatrix[4] is required to translate the vertex from mesh space to bone space.

[4] https://learn.microsoft.com/en-us/windows/win32/direct3d9/id3dxskininfo--setboneoffsetmatrix

# Chapter 3. System Overview and Preprocessing

This chapter gives detailed information on how multiple stages of the entire data processing are integrated together. Moreover, the functionality of each module and the input and output are explained. This gives a step-by-step overview of how the framework of the system is constructed and operated. In addition, this chapter demonstrates the preprocessing stage.

## 3.1. System Architecture Overview

Our system architecture is based on an integration of three primary approaches: To reconstruct human 3D models from a single picture using the PIFuHD pretrained model, to capture motions from sports videos using the MocapNET model, and to do the auto rigging and animation copy. As indicated below, the entire system architecture consists of five major modules called Image Acquisition Module, Image Preprocessing Module, 3D Human Reconstruction Module, Motion Capture Module, and Rigging and Animation Copy Module. Figure 3.1 shows how the five modules are coupled and integrated.

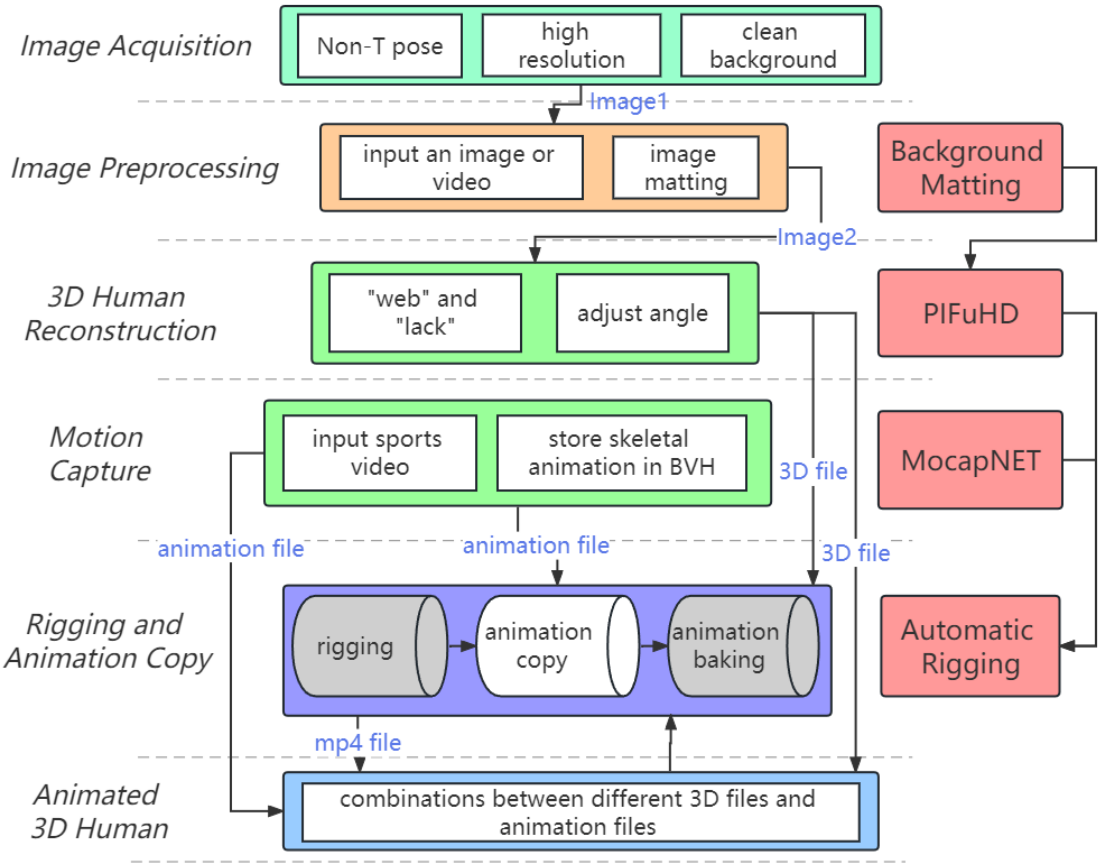


Figure 3.1: General System Architecture

The functionalities of each module are described as the following:

1. Image Acquisition Module

This module is dedicated for extracting 3D models. Due to the impact of COVID-19, using the Kinect multi-camera method to output high-precision 3D models presents major challenges due to newly implemented regulations in the university laboratory. As an alternative solution, this research chooses the method that reconstructs the 3D model of the human body from a single monocular photo for our case instead. As the first step in our entire system, the 3D model of the human body created later on is entirely dependent on the monocular photo output by this module. This module puts forward strict requirements for the acquisition of this monocular photo, to ensure that the extracted 3D model is correct and complete, with as many details as possible of the human body and clothing attached.

2. Image Preprocessing Module

The images acquired from the previous module require further processing before they can be used to enhance the accuracy of the reconstructed 3D model. As the second step in our system, this preprocessing process utilized by this module is mainly to apply several denoise techniques on monocular photos, including background removal, human body stroke, and occlusion removal. The final output is a complete body photo with its background removed.

3. 3D Human Reconstruction Module

As the 3rd step in the system, the function of this module is to convert a monocular photo input into a 3D file output. The input monocular photo comes from the human photo obtained by the previous image preprocessing module, and the output 3D file is a reconstructed human 3D model based on the corresponding monocular photo. This module uses the pre-trained model of the PIFuHD, which can be run locally or on the Google Colab.

4. Motion Capture Module

In the 4th step, this module is capable to perform gesture recognition and skeletal action extraction from videos captured by mobile phones or webcams. The video sources of the gesture recognition can be recorded locally, or downloaded online. This module utilizes the MocapNET for pose recognition and the generation of a skeletal animation file, which can be used for rigging with 3D human body in Blender later to mimic the recorded action presented in the original video.

5. Rigging and Animation Copy Module

The module in this last step is designed to fuse the 3D files and animation files output by modules 3 and 4 in the 3D file processing software-Blender. The 3D human body is bound and skinned in the Blender, and the bound bones are associated with the input skeletal animation to replicate similar actions. We design an auto

rigging method to avoid manual operations, like adjusting the bone's position and setting up the weight, to ensure a final animation is generated efficiently.

In our work, the system is evolvable and extensible.

1. The modules in the framework are loosely coupled with each other and each module is relatively independent. So, the algorithm in the corresponding module in this framework can be updated. Different 3D reconstruction algorithms, motion capture algorithms, and rigging algorithms have a huge impact on the final result.
2. In terms of extensibility, new modules can be added to this framework. For example, 3D rendering (adding color to the character), motion segmentation and compositing, etc. The addition of new modules will have no impact on the existing modules. The system user only needs to control the input and output of each module.

## 3.2. Image Acquisition Module

This section describes the preprocessing before restoring the 3D human body from a monocular image. The quality of the reconstruction depends significantly on the obtained quality of the monocular photo. Before the preprocessing, we proposed several requirements for the photo, such as the resolution of the photo, and the acceptable human body poses.

### 3.2.1. Resolution Effects

Results of high-resolution reconstruction enable the DT to acquire a higher value in a variety of disciplines. For instance, in virtual fitting, rich garment details and realistic limb morphology used to communicate data between the computer and the human body give clients with a high-quality, commercially valuable fitting experience[2]. In this thesis, OpenPose[50] and PIFuHD are used for 3D reconstruction. OpenPose generates the

keypoints' position file, which provides input parameters for the reconstruction procedure and has no influence on the output resolution. While PIFuHD is capable of outputting results with various resolutions, for instance in Figure 3.2. In the output findings, the identical single picture with a resolution of 128 × 128 exhibits blurred face and finger features. As the resolution increases, the accuracy of the human information improves.

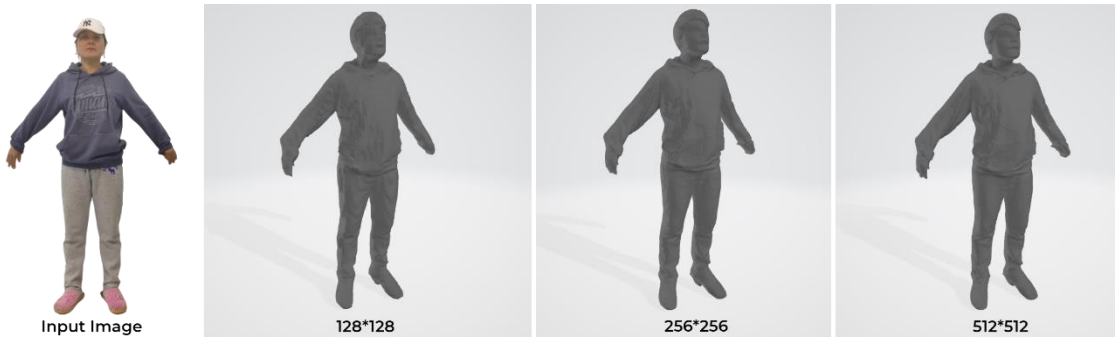


Figure 3.2: The Influence of Different Resolution to 3D Reconstruction Results

### 3.2.2. Requirements for Taking Photos on Static Poses

The algorithm "PIFuHD" used in this thesis to reconstruct the 3D human body produces better results on high-resolution photos than on low-resolution photos.

The resolution of the photo needs to be higher than 1024 × 1024. When the resolution of the photo is too low, many 3D points become hard to identify from the reconstructed 3D model, especially around the limbs.

Figure 3.3 shows A-pose, T-pose and natural pose. In the training set data of the PIFuHD, most photos are taken when participants are in a natural stance. As a result, the trained model would give a much better performance when input data in a natural stance is introduced. And standard T-pose will cause errors in reconstruction results which will be presented in Section 4.2.1.

Additionally, the input poses of the human body should not be too random, or more manual labor will be required to select, adjust, and bind, or more manual labor will be required to select, adjust, and bind the human bones for the reconstructed 3D human body in the Blender later. To clearly distinguish the gaps between bones and body in the Blender

when importing an action, one has to make sure that an identifiable gap between the body and limbs of a participant is well presented. This is a preventative measure for separating the clothes from the human body in the 3D model during the import of an action.

Based on the above requirements, we decide to adopt a standing posture with straight and slightly downward arms, and stand apart feet in this research. Notably, the gap between the limbs of the human body needs to be large enough, and photos with participants wearing thick clothes and pants are not recommended. In conclusion, we mainly apply photo with A-pose as Figure 3.3 shows.

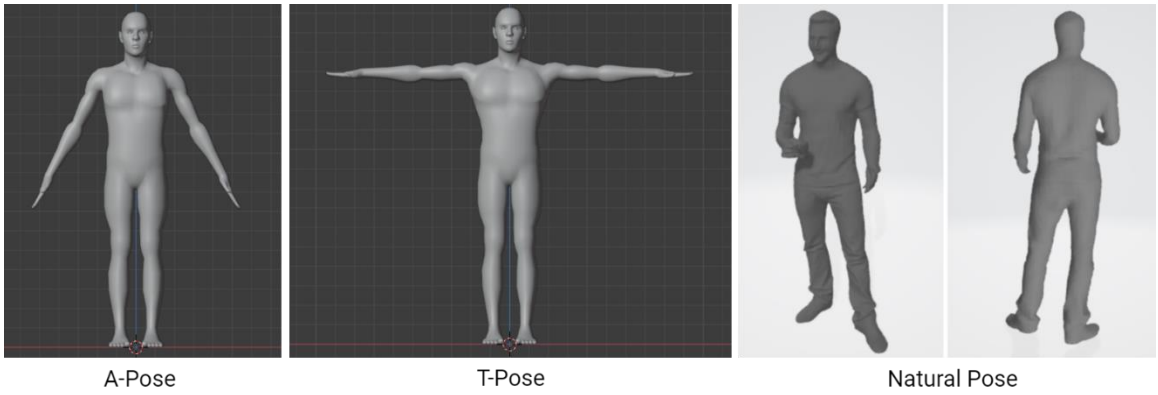


Figure 3.3: A-Pose, T-Pose and Natural Pose

## 3.3. Image Preprocessing

Background Matting[18], created by Shanchuan Lin et al., is used for preprocessing in this study. It is currently extensively cited in other video conferencing systems, including Zoom and Microsoft Teams. Users may preserve their privacy by removing the backdrop from their location. This research primarily used this technique to get a more precise 3D human body. Figure 3.4 demonstrates that even with a pretty clean backdrop and without Background Matting, the head portion of the 3D human body results include artifacts and missing data. When reconstructing a human body from the same picture using matting followed by reconstruction, the face features are more distinct than using the former method.


Figure 3.4: Result without Matting and Result with Matting

Numerous applications currently offer background replacement functionality. Marco Forte et al.[58] proposed F, B, Alpha Matting (FBA) adapting a low-cost modification to alpha matting networks. Compared to Background Matting, most photos with FBA still have artifacts after processing, as Figure 3.5 shows. The processing result of FBA is obvious and competitive based on the preceding photos, but he requires thorough human annotation of Trimap. Hence, we choose the comparably fast and straightforward output. In addition, Background Matting is very adaptable since it may use a variety of dynamic movies and 2D photos of intricate scenes.

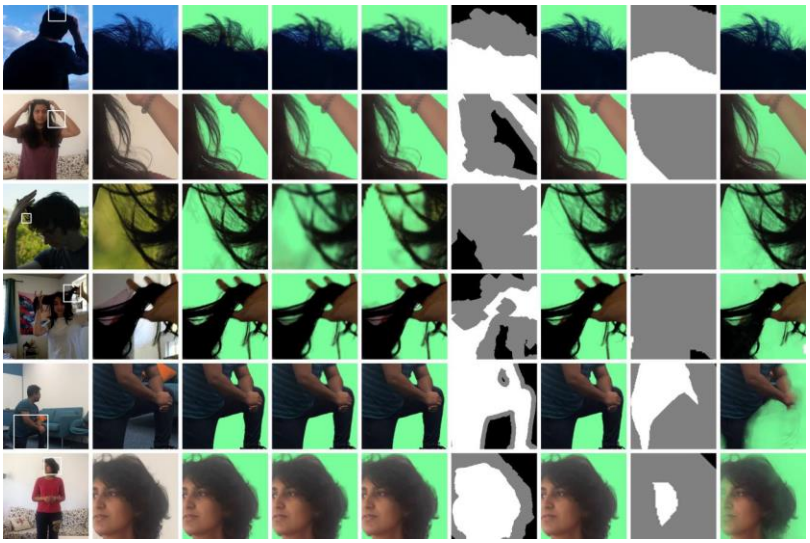

Figure 3.5: Artifacts[18]

## 3.4. Discussion

This chapter introduces the project's structure and explains each step's procedure in detail. Next, the pre-processing steps, including picture acquisition and matting, are presented to prepare the job for the subsequent project. The available open-source matting techniques need to be more transparent regarding hair processing. Moreover, when the subject throws many shadows on the backdrop or is the same color as the background (at the top) and when the background has much texture (at the bottom), the results of Background Matting are not suitable.

# Chapter 4. System Functions Development

This chapter elaborated on the two functions of the system: 3D human body reconstruction and motion capture.

3D human reconstruction consists of two major modules: the skeletal information extraction module and the reconstruction module. The skeletal information extraction module uses OpenPose[50], an open-source library created by CMU (Carnegie Mellon University), based on convolutional neural networks, supervised learning, and the Caffe framework. OpenPose employs a non-parametric model to link body parts with individuals in photographs. This strategy begins with a top-down system to obtain high accuracy and real-time performance, with little or no linkage to the individual in the 2D picture. In this article, pre-processed 2D photos are fed into OpenPose to generate JavaScript Object Notation (JSON) files, including crucial bone positioning information for use as input parameters for 3D reconstruction work. PIFuHD then utilizes these characteristics to generate the 3D structure of the human body of interest.

The static 3D structure of the human body reconstructed through PIFuHD does not have any movement, while a dynamic 3D human body requires the combination of motion capture technology. In the motion capture module, we used MocapNET to extract skeletal motion data of individuals from different videos and saved them as BVH files. These dynamic data will be combined with the static 3D model to complete the conversion to a dynamic 3D model. The animation conversion process will be elaborated in detail in Chapter 5.

## 4.1. PIFuHD Introduction

Our research focuses on generating highly detailed 3D models of human structures from a single 1k picture utilizing PIFuHD, which does not need depth sensors or motion capture equipment. As previously explained, PIFuHD relies on coarse and fine layers for

PIFu functionality. To maintain local picture features, PIFu first incorporates a pixel-aligned implicit function represented by a convolutional image encoder G and a multilayer perceptron[40]:

$$f(F(x), z(X)) = s : s \in R[6] \quad (4.1)$$

For a 3D point X, x = π (X) is its 2D projection, z(X) is the depth value in the camera coordinate space, F(x) = g(I(x)) is the image feature at x

As illustrated in Figure 4.1, PIFuHD employs coarse-to-fine two-level pipeline for fine surface reconstruction. This research uses the pre-trained PIFuHD model to generate the result containing the desired 3D human body, which is the fundamental step of the animation.

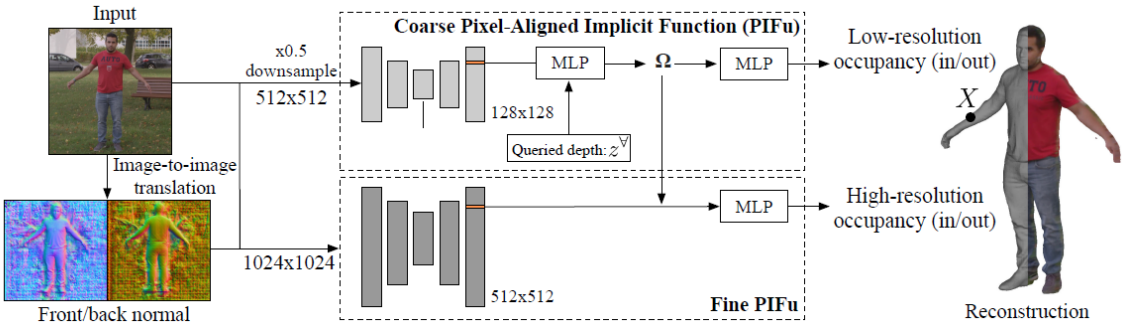


Figure 4.1: Diagram of the Operational Framework for Surface Reconstruction[6]

## 4.2. Experiment Results and Evaluation

### 4.2.1. The Impact of Dress and Arm-to-Body Angle

Based on the T-pose, we investigated various arm-body clamping and wearing angles. During reconstruction, we discovered that some relative arm-to-body postures cause gaps and webs. Figure 4.2 demonstrates that when the angle between the arm and body of the human body is too tiny, adhesion of the body and arm in the 3D results occurs, which we

refer to as a "web." The web produces experimental faults in binding weights and pivots control when the 3D human body is utilized for rigging. Figure 4.3 demonstrates that when the angle between the arm and the body is too great, it will result in a lack of 3D points at the end of the arm, which we refer to as a “lack”.

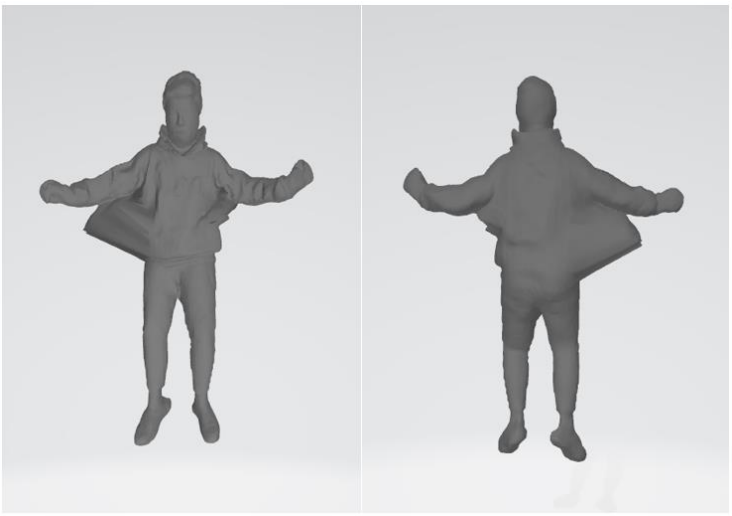

Figure 4.2: “web”

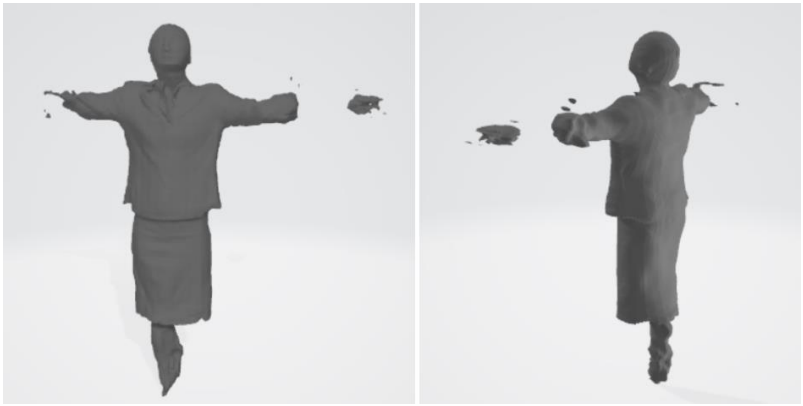

Figure 4.3: “lack”

The PIFuHD model was tested comprehensively with two variables: arm-to-body (A2B) angle, and hand-to-body(H2B) distance. As Figure 4.4 shows, the human body wearing light clothing reveals its structure more clearly than when wearing bulky clothing, which facilitates the measurement of A2B angle and comparison of experimental data. At

the same time, it is simpler to generate a “web” of a person wearing bulky clothing. In this research, we utilize the 2D picture of the human body wearing lightweight clothing.

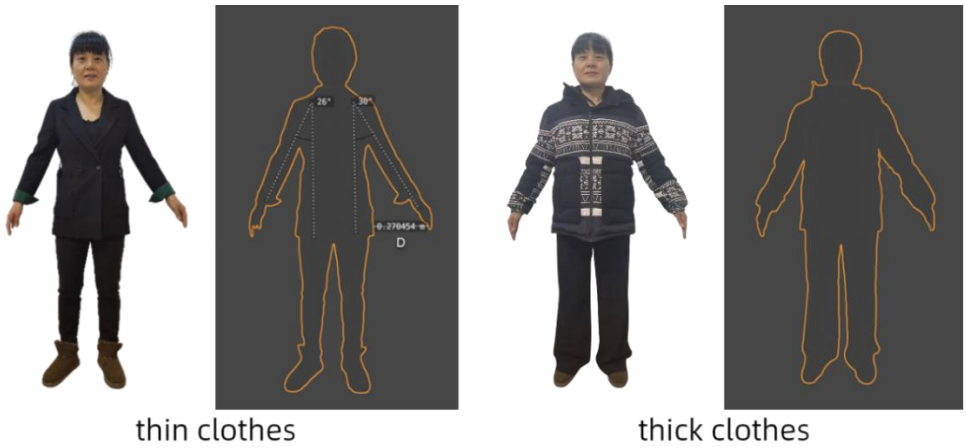


Figure 4.4: Clothes Outline

During the experiment, the A2B angle and H2B distance are calculated. The A2B angle varies between 0 and 90 degrees while the H2B distance is measured according to the A2B angle. We tested the results for 3 participants as Figure 4.5 shows (Participant 1 with height of 1.32m, participant 2 with 1.66m, and participant 3 with 1.79m). The A2B angle starts at 0° and increasing by 5° to a maximum of 90°. The measurements were collected using the Clinometer in Figure 4.6, with the two straight edges aligned with the ipsilateral arm and trunk, and the central circular segment displaying angles. And we recorded the results in Table 4.1.

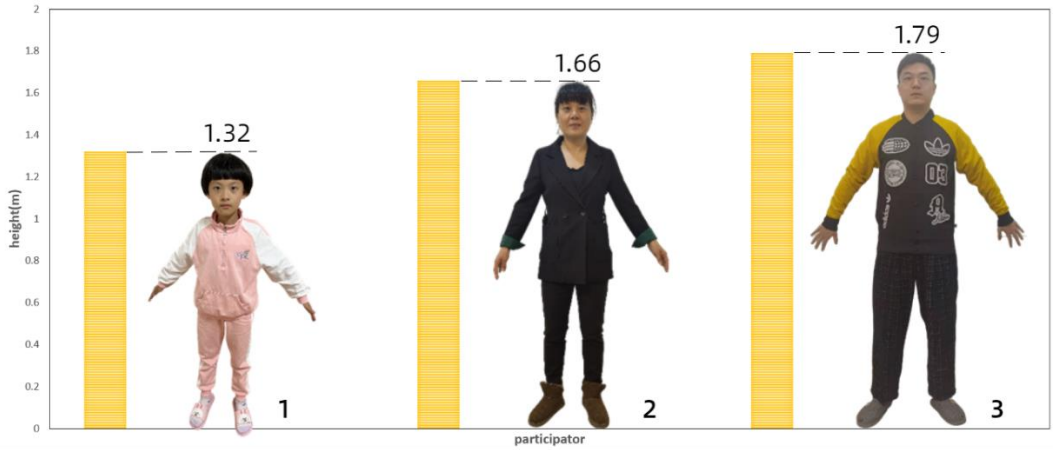


Figure 4.5: Three Participators in Angle Test


Figure 4.6: Clinometer

Table 4.1: Angle Test Result

| Participator | Height(m) | A2B Angle(°) | H2B Distance (m) | If "web"(Y for yes, N for no) | If "lack"(Y for yes, N for no) |
|---|---|---|---|---|---|
| 1 | 1.324 | <25 | <0.195 | Y | N |
| **1** | **1.324** | **25-75** | **0.195-0.498** | **N** | **N** |
| 1 | 1.324 | >75 | >0.498 | N | Y |
| 2 | 1.661 | <20 | 0.227 | Y | N |
| **2** | **1.661** | **20-70** | **0.227-0.630** | **N** | **N** |
| 2 | 1.661 | >70 | >0.630 | N | Y |
| 3 | 1.786 | <20 | <0.249 | Y | N |
| **3** | **1.786** | **20-70** | **0.249-0.713** | **N** | **N** |
| 3 | 1.786 | >70 | >0.713 | N | Y |

From Table 4.1, despite the various heights, "web" is readily formed when the angle is less than 25°, but "lack" is easily generated when the angle is more significant than 70°. Since a wider angle is more advantageous for rigging, we obtain optimal experimental results when the angle is approximately 60 degrees. Moreover, we compared the height and H2B distance data of the three participants, and the experiment outcomes were not necessarily connected. Therefore, the experimental results mainly rely on the A2B angle. And we apply 60° as the standard A2B angle in the future steps.

### 4.2.2. Comparison with Other Models

The chamfer distance (CD) is primarily used to quantify the degree of fit between two-point stacks and the effect of 3D reconstruction[59]. The CD serve primarily as a comparison parameter for various reconstruction methods in this work. The PSG point cloud reconstruction method initially suggested the CD to quantify the distance between the rebuilt point cloud and the ground truth point cloud. It has played a significant role in point cloud reconstruction[60]. Chamfer Distance (CD) is an evaluation metric for two-point clouds. It takes the distance of each point into account. For each point in each cloud, CD finds the nearest point in the other point set and sums the square of distance up. It is utilized in Shapenet's shape reconstruction challenge[61].

Chamfer Distance defines the distance between the genuine mesh and each point on the predicted mesh's nearest neighbor. This index also includes a completeness index, which calculates the nearest neighbor distance from every point on the genuine mesh to each point on the projected mesh. The formula is shown below[62]:

$$\mathrm{CD}(S_1, S_2) = \frac{1}{|S_1|}\sum_{x \in S_1} \min_{y \in S_2} \| x - y \|_2^2 + \frac{1}{|S_2|}\sum_{y \in S_2} \min_{x \in S_1} \| x - y \|_2^2 \quad (4.2)$$

The first term of CD represents the total distances between every point x in the scene point set S1 and the closest point in the model point set S2. The second term is the total of the distances between every point y in the model point set S2 and the closest point in the scene point set S1. A lower CD indicates that the reconstructed model corresponds better to the actual scene. A higher CD indicates that the reconstructed model is far from the original scene and that the predicted target parameters must be more accurate. CD measures the precision of model reconstruction and parameter prediction at a 3D point cloud level.

However, due to the lack of datasets containing frontal photos of human bodies and 3D human body ground truth, we cannot calculate the CD value in this research. As an alternative evaluation method, we designed a survey form that was randomly distributed to

different testers. Testers are from a group of Chinese observers aged from 6 to 74. The survey form recorded the degree of restoration perception of the subjects to 3D reconstruction work. By providing different restoration results of the same picture to all testers (including the restoration result of HMR, the restoration result of PIFuHD, and the restoration result of PIFuHD with matting), their intuitive observation was recorded. Similarity perception is evaluated visually on a scale of 0 to 10 (0 being completely different, and 10 being completely identical) to evaluate the accuracy of the three algorithms. The survey results are recorded in Table 4.2 below. It can be seen that the overall restoration accuracy is highest in PIFuHD with matting and lowest in HMR.

Table 4.2: Survey Form for 3D Reconstruction Restoration Perception

| | HMR | PIFuHD without matting | **PIFuHD with matting** |
|---|---|---|---|
| Participant 1 | 3 | 6 | **8** |
| Participant 2 | 5 | 8 | **9** |
| Participant 3 | 1 | 5 | **5.5** |
| Participant 4 | 2 | 6 | **8** |
| Participant 5 | 4 | 7 | **7.5** |
| Participant 6 | 3 | 6 | **7** |
| Participant 7 | 4 | 7.5 | **8** |
| Participant 8 | 7 | 8 | **9** |
| Median | 3.5 | 7.5 | **8** |

## 4.3. Motion Capture

The approach to complete the 3D human pose is divided into two main phases: Extracting 2D human joints; Upgrading the 2D joints to 3D. We used MocapNET with breakneck estimation speed and direct BVH output in this project. It models the reciprocal links directly instead of using exhaustive joint-to-joint relationships for neural networks. Two-dimensional human poses are first extracted from two-dimensional image species using OpenPose[50], which is converted into two normalized signed rotation matrices (NSRM) encoding the upper and lower body halves. MocapNET converts the NSRMs into BVH poses by selecting integrations trained specifically for classification orientation.

Finally, the BVH pose is improved by inverse kinematics (IK) with optionally known limb sizes and camera configurations.

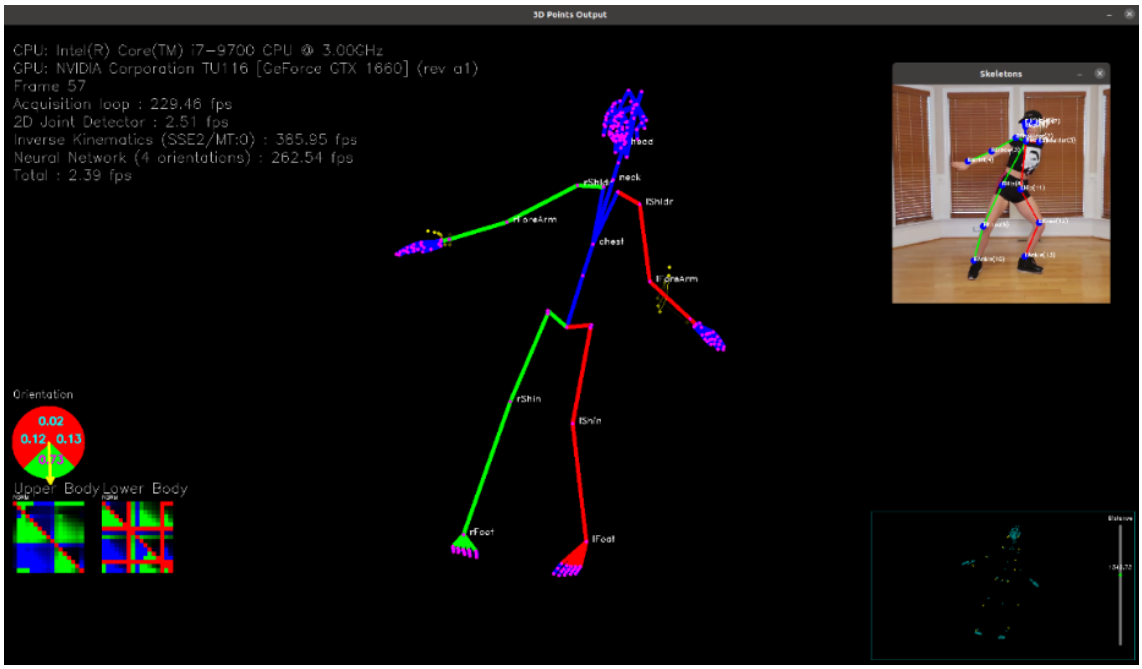


Figure 4.7: 3D Motion Capture via MocapNET

Figure 4.7 presents the running interface of the referenced model MocapNET. In our research, the MocapNet process is running under the following hardware conditions: i7-9700 CPU, NVIDIA GTX 1660 GPU, and the process speed on the video is 2.39 fps. MocapNET supports real-time pose estimation. This thesis does not apply real-time motion capture, but uses recorded motion videos to extract motion. Figure 4.8 presents the output of MocapNET. All the skeletal animation is saved in BVH file and will be further processing in Chapter 5.

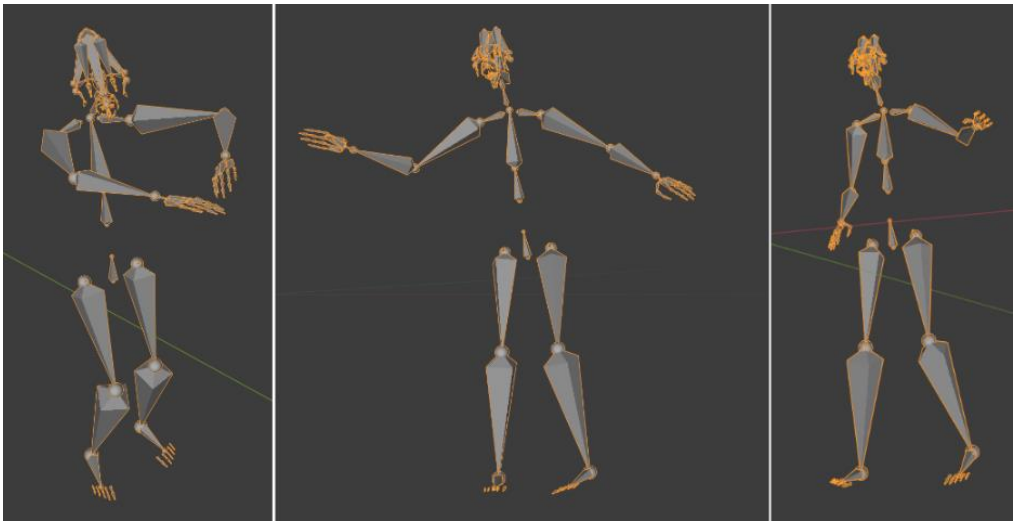

Figure 4.8: The Display of Skeletal Animation Saved as BVH file in Blender

## 4.4. Discussion

Compared to HMR and PIFuHD without matting, PIFuHD with matting produced more accurate 3D reconstructions of the target human body in this investigation. Matting followed by PIFuHD reconstruction can correctly identify the target human body. Additionally, we need the initial photos with a person standing in A-pose and at an appropriate A2B angle, to avoid “web” and “lack”. When additional rigging is conducted, the reconstruction of the human stance in A-pose is more accurate compared to other poses.

In addition, PIFuHD can produce output with different resolution We examined the output with resolutions of 128×128, 256×256, 512×512, and 1024×1024 and discovered that greater resolution linking to more precise 3D reconstruction results. Nevertheless, the better the resolution, the longer the runtime and the higher the hardware requirement, such as the GPU and RAM. Besides, when comparing the outputs of 512×512 and 1024×1024, there is no discernible difference in clarity and precision. However, the resolution of 512×512 has a less runtime and lower hardware requirements, which is closer to the actual modeling requirements. We choose the 512×512 outcomes of 3D reconstruction for the automatic rigging process module.

For motion capture part, there are still limitations in using MocapNET to go for a 3D human pose. After capturing the motion, MocapNET copies the motion to the same armature without creating a new armature based on characters of different sizes. The built-in armature of MocapNET is a body shape average skeleton. In the case that the body shape of the detected person is not around the average, the application of motion capture may produce errors. Using the same skeleton, without changing the skeleton size and dimensions, cannot be used to measure various human body data. Only the same skeleton is used to save the motion state.

# Chapter 5. Rigging and Animation Copy

This chapter uses Blender to reconstruct a 3D human body (OBJ[5] format) and skeletal action file (BVH format) for fusion to achieve 3D human animation. The specific operations include rigging and action duplication. This study automates the script with guaranteed accuracy.

## 5.1. Rigging Principle

Rigging is to simulate human moving joints, create virtual bones at the corresponding locations in the animated character image, add motion constraints, and bind at the character joint bones or the desired locations of the character using virtual controllers [63]. We mainly use the controller to manipulate the character, where simple human body movements such as hand clapping, walking, running, and stretching can be performed.

In this project, the rigging is done using Blender software, where each object is isolated by default, and isolated objects do not affect each other [64]. Figure 5.1 shows the parent-child relationship in 3D animation. A parent-child relationship can be set for both objects to make one of them can follow the other to move or rotate [64]. The parent will control the movement of all child nodes below it, and changes in the child level do not affect the parent [65]. The parent-child relationship is nestable; a child level can have only one parent, and a parent level can have multiple children [66].

[5] https://docs.fileformat.com/3d/obj/

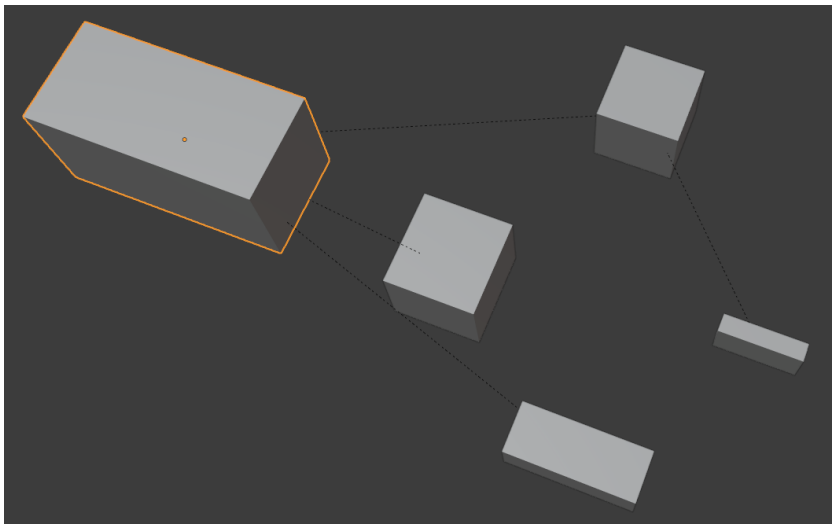

Figure 5.1: Parent-Child Relationship

A mesh is a pile of points (vertex) used to manipulate the human body model, an object with all the vertices and faces in the same mesh. The essence of local control of such models is to move some of the points in the mesh. Precise local control cannot rely solely on the parent-child level but requires the introduction of bones. A skeleton is a virtual object in Blender, not a mesh, used to influence the model's vertices. As shown in Figure 5.2, a skeleton is a vertex's parent level, directing specific vertices' movement on the mesh.

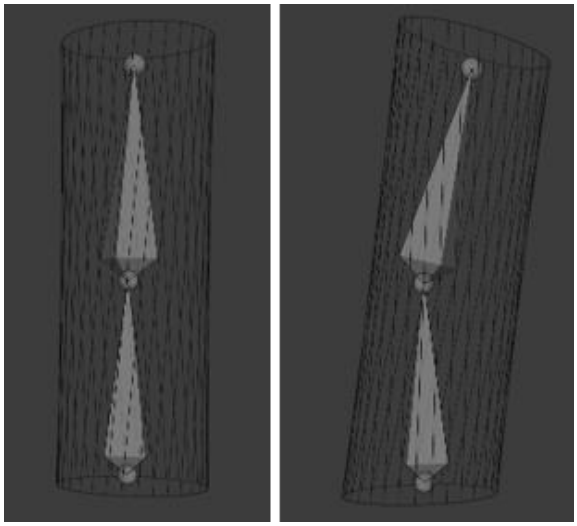

Figure 5.2: Skeleton Leads Vertices' Movements

Blender can connect bones and vertices by grouping points in a mesh. Generally, each bone can control a vertex group of its surrounding vertices. The same vertex can exist inside multiple vertex groups at the same time. In Blender, weights are used to indicate the amount of influence a bone has on a vertex. The weight is between 0 and 1, indicating the degree of influence, with a weight of 1 indicating 100% influence and 0 indicating no influence.

### 5.1.1. Kinematics in Animation

In Blender, different systems are used for different actions, which makes the results more realistic and natural, and the operation more flexible. We used the Inverse Kinematics (IK) and Forward Kinematics (FK) systems in this project. In the IK system, when the final node moves, the parameters of each node in the connected objects (kinetic chain) are automatically calculated to reach the desired position [67]. IK refers to how the parent node will be tugged in its displacement and rotation when the child node moves. For dragging, tugging, and supporting these actions, it is more accessible and speedier to accomplish them using the IK system.

However, active force actions, such as punching in boxing, still rely on the FK system, which refers to the displacement and rotation of a part of the joints in the model at a specified time. In other words, FK means how the parent node affects the child nodes when it moves or rotates[67]. The punch is a force transmission process, which emanates from the waist, rotates through the upper body, drives the shoulders, passes to the elbows, and finally, the hands deliver the force. This action is more in line with the FK system.

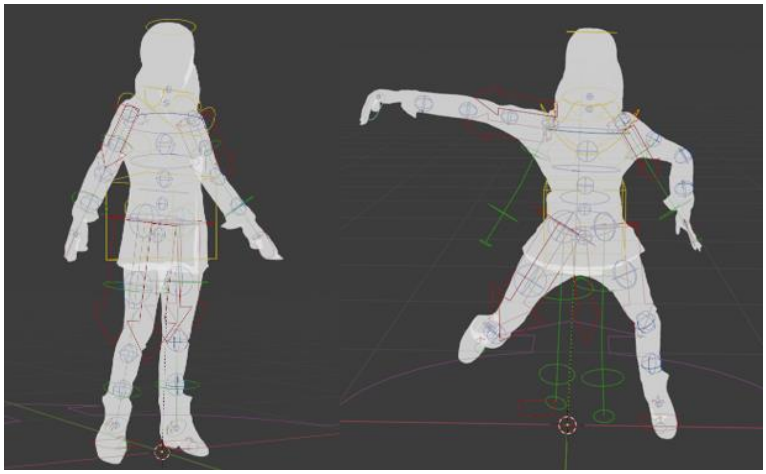

Figure 5.3: The Controller of IK and FK

As Figure 5.3 shows, we use Rigify to generate controllers for the whole human body automatically. The controller is associated with the IK/FK system and has the corresponding visibility. Under the IK system, only the IK controller is visible, and under the FK system, only the FK controller is visible. When the controller is in IK mode, the weight value of IK in the driven panel is 1, and the weight value of FK is 0. When the controller is in FK mode, the weight value of FK in the driven panel is 1, and the weight value of IK is 0. In Blender's action mode, it is possible to detect the action pattern of each body part under the influence of multiple controllers. In order to meet different movement needs, the weight and position of each part of the controller can be adjusted, and new controllers can be customized.

## 5.2. Rigging in Blender

Blender is an integrated open-source 3D software. The mainstream can be completed in three ways of rigging. And we indicate the optimized auto rigging method in this section.

### 5.2.1. Introduction to Three Methods of Rigging

We utilized Rigify, a plug-in of Blender, to quickly generate skeleton sets with Inverse Kinematics (IK) budgets that conform to the laws of motion on ordinary skeletons. To control the 3D human model, meta-rigs[6] are required to fit the 3D model's key bones. A meta-rig is an assembly of bone chains and every bone is able to control the motion and rotation of surrounding 3D points. Rigify's function is providing several practical human armatures, here called meta-rigs, for rigging process. We applied a basic built-in meta-rig of Rigify as Figure 5.4 shows, which represents the head and hands with simple bones. After rigging, a rig with controllers will be generated as Figure 5.4 shows. Users can control the movement and rotation via the controllers set in the human armature.

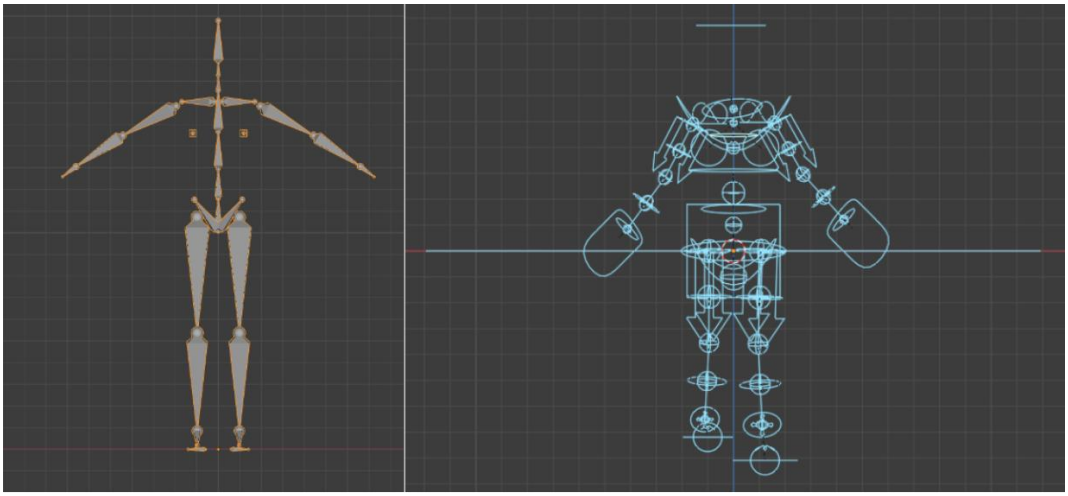


Figure 5.4: Metarig and Rig with Controllers in Rigify

The second is a paid plug-in for blender called Auto Rig Pro. for humanoid characters. Auto-Rig Pro is an all-in-one solution to rig characters, retarget animations, and provide 3D export for Unity and Unreal Engine. It provides functions that specify a few body joints to rig the skeleton automatically. This method speeds up the editing process

[6] https://docs.blender.org/manual/en/latest/addons/rigging/rigify/metarigs.html

but could be more accurate. For non-humanoid characters, such as dogs and horses, it remains to set all the bones into the mesh points manually.

The last one, Mixamo, is an animation library from Adobe that provides a unique API as a plug-in interface for blende [68]. Mixamo's proprietary dataset contains about 2400 unique motion sequences for 71 characters (armature), which can satisfy various actions of the basic human body [69]. After uploading a static character model in Mixamo, the character binds with the character on Mixamo and will act according to a character's action. This model is downloaded as an FBX[7] format file and rendered in blender by its auto control rig.

### 5.2.2. Optimize Rigging for Automation

We optimize Rigify's basic manual rigging function to an automate rigging function. This approach accelerates the process of rigging while ensuring the accuracy.

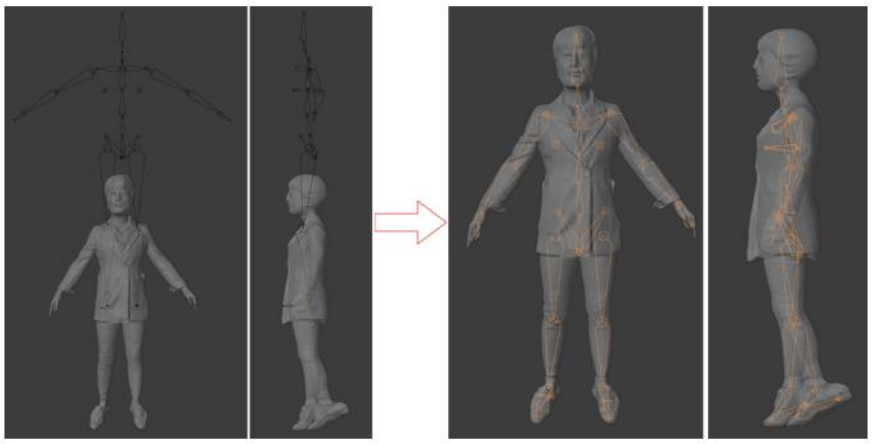


Figure 5.5: Rigify's Initial Metarig Position

As Figure 5.5 shows, the default initial position of the bones added by Rigify is above the imported model. Generally, it is necessary to manually adjust the position of

[7] https://www.autodesk.com/products/fbx/overview

each bone in the Rigify armature to fit it inside the human model. But manual rigging is complex according to Figure 5.6.

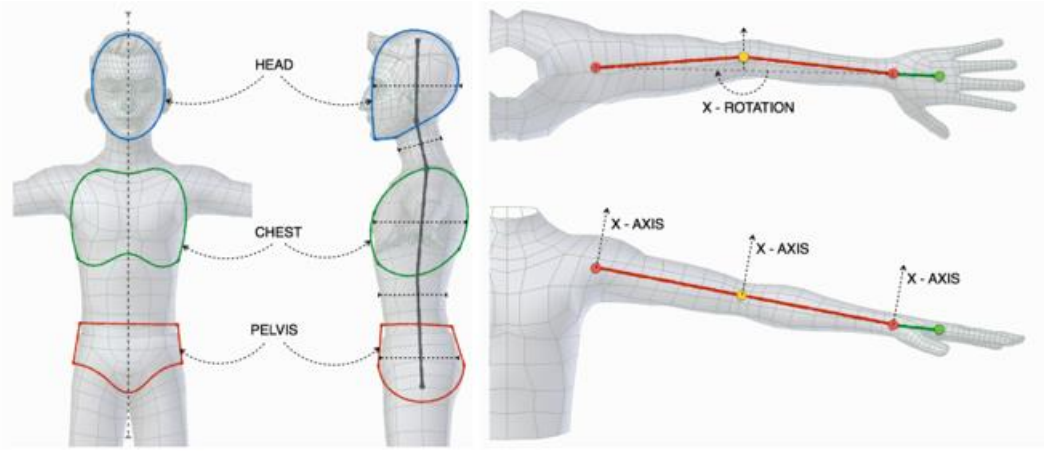


Figure 5.6: Manual Rigging Standards[8]

As shown in the Figure 5.6, animators need to manually adjust the position, length and angle of metarig according to the standard position of human bones in Blender's official document to make metarig fully fit into the 3D human model. Although this manual method is highly accurate, it is time-consuming, labor-intensive, and non-reusable.

We introduced Python scripts on Rigify's source files. The first step is to adjust the initial position of each skeleton in the metarig to adaptive human coordinates. Each control in the Blender UI comes with a corresponding Python Application Programming Interface (API). In theory, any manual operation can be executed using Python scripts.

This experiment consulted a large number of related works on manual rigging. Manual bone rigging entails specific technical requirements for the operator and necessitates numerous rebinding operations for different 3D shells. Although there may be more intricate details in certain areas, such as fingers and faces, the time cost of manual rigging is high, and it lacks universality. Thus, in this project, we utilized two algorithms and two scripts to accomplish automatic rigging adapted to the 3D human shell of this project.

[8] https://docs.blender.org/bone_positioning.html

The 3D file obtained from PIFuHD is recognized in Blender as a surface point 3D shell. To avoid manually modifying the armature, we need to automatically adjust the coordinates of each armature bone to fit the 3D shell in Blender.

We first used pose estimation to obtain the coordinates of each key point in the image. The coordinates of the critical points in the picture obtained by OpenPose[50] are in picture pixels, and the upper left corner of the picture is set to zero. After obtaining the picture pixels, we needed to find the real-world coordinates of these points in Blender. To achieve this, we approximated the coordinates of the corresponding vital points in Blender using a range. Consequently, there were two sets of discrete points: one set was composed of the key point coordinates of the 2D image obtained by OpenPose, while the other set was composed of points corresponding to the body key points in the actual coordinate range of the Blender world coordinate system. To determine the equation for the automatic transformation of the coordinate system and find the one-to-one correspondence between the two sets of discrete points, we used the least squares method. This method is mainly used to fit the most approximate curve of discrete points, and in this experiment, the transformation of the coordinate system was a straight line. In order to achieve the highest accuracy closest to reality, we calculated the fitting curve of the x coordinate and the fitting curve of the y coordinate separately. By fitting the point distance according to the corresponding relationship determined by the first set of discrete points and the second set of range points, we determined the distance of the straight line, with several error points removed, to determine the final reliable approximate correspondence. The formula of the least squares method is shown below[70].

$$\frac{\partial J}{\partial w} = \frac{\partial(\frac{1}{2m}\sum_{i=1}^{m} (y_i - wx_i - b)^2)}{\partial w} = \frac{1}{m}\sum_{i=1}^{m}(y_i - wx_i - b)(-x_i) \quad (5.1)$$

Formula (5.1) represents for solving model parameters using the least squares method in linear regression. Here, $J$ represents the loss function, $w$ represents the coefficient of the independent variable, $b$ represents the intercept, $y_i$ represents the dependent variable of the i-th observation, xi represents the corresponding independent

variable, and m represents the sample size[70]. The fitting results of the least squares method for x and z coordinates are showed in the following Figure 5.7.

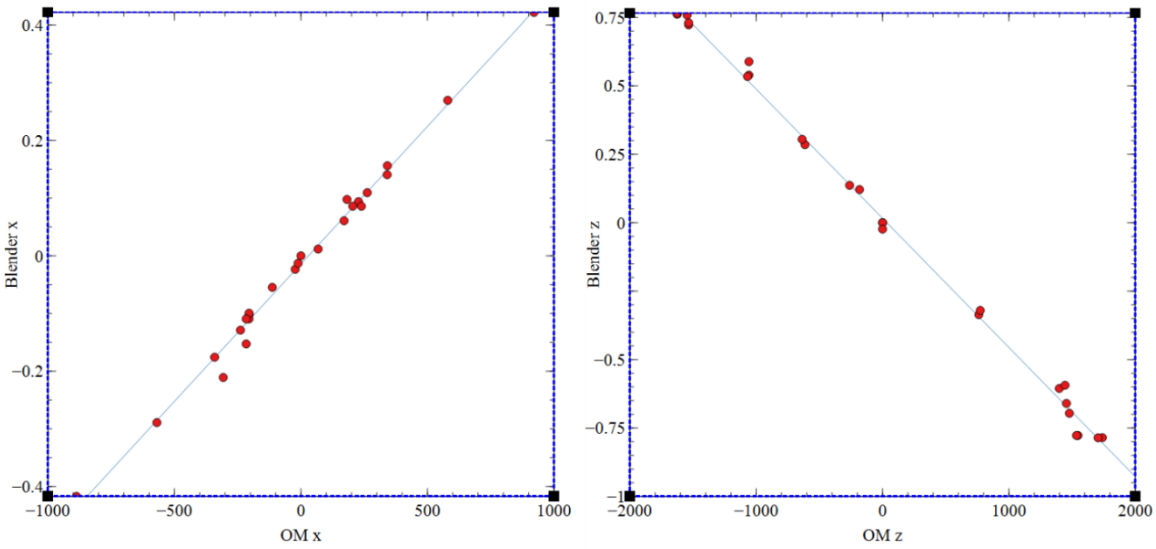


Figure 5.7: The Fitting of the Least Squares Method for x and z Coordinates

After obtaining the most closely corresponding curve using the least squares method, we can calculate the approximate coordinates of the key point in the Blender world coordinate system by using the 2D key point coordinates obtained by OpenPose. During rigging, each bone has two nodes: the head and the tail. Therefore, we use a specific equation to convert the key point coordinates to the coordinates of the head and the tail of the bone at the respective key point positions. To approximate the head and tail coordinates, we use a set of offsets, which we determine based on various data tests. After conducting numerous data tests, we determine the offsets to estimate the coordinates of the head and the tail of each bone. By adding and subtracting the offsets from the key point coordinates, we can approximate the coordinates of the head and the tail. For example, the shoulder point (X, Y) refers to the key point in the middle of the shoulder bone. Since the shoulder bone is horizontal, we obtain the head coordinates of the shoulder bone as (X-0.1, Y-0.2) and the tail coordinates as (X+0.1, Y+0.2) based on the established offset. The leg bones are longitudinal, so we consider their characteristics when determining the head and the tail nodes. For instance, when the key point of the leg bone is (x2.y2), we set the head node of the leg bone as (x2+0.2, y2+0.8) and the tail node as (x-0.2, y2-0.8). We implemented a

lot of experiments and produced the following Table 5.1 recording the optimal coordinate offset to get the best conversion effect.

Table 5.1: Optimal Offsets for the Correspondence between Key Points and Bones

| | | | x | y | z |
|---|---|---|---|---|---|
| 0 | spine | head | 0 | 0.0552 | 0-(Lhip x - Rhip x)/2 |
| | | tail | 0 | 0.0172 | 0+(Lhip x - Rhip x)/2 |
| 1 | spine.001 | head | 0 | 0.0172 | 0+(Lhip x - Rhip x)/2 |
| | | tail | 0 | 0.0004 | 0+(Lhip x - Rhip x)/2+(Lshoulder z - MidHip z)/4 |
| 2 | pelvis.L | head | 0 | 0.0552 | 0-(Lhip x - Rhip x)/2 |
| | | tail | Lhip x | -0.0451 | 0+(Lhip x - Rhip x)/2 |
| 3 | pelvis.R | head | 0 | 0.0552 | 0-(Lhip x - Rhip x)/2 |
| | | tail | Rhip x | -0.0451 | 0+(Lhip x - Rhip x)/2 |
| 4 | thigh.L | head | Lhip x | 0.0124 | MipHip z |
| | | tail | Lknee x | -0.0286 | Lknee z |
| 5 | thigh.R | head | Rhip x | 0.0124 | MipHip z |
| | | tail | Rknee x | -0.0286 | Rknee z |
| 6 | spine.002 | head | 0 | 0.0004 | 0+(Lhip x - Rhip x)/2+(Lshoulder z - MidHip z)/4 |
| | | tail | 0 | 0.0059 | 0+(Lhip x - Rhip x)/2+(Lshoulder z - MidHip z)/5*3 |
| 7 | shin.L | head | Lknee x | -0.0286 | Lknee z |
| | | tail | Lankle x | 0.0162 | Lankle z |
| 8 | shin.R | head | Rknee x | -0.0286 | Rknee z |
| | | tail | Rankle x | 0.0162 | Rankle z |
| 9 | spine.003 | head | 0 | 0.0059 | 0+(Lhip x - Rhip x)/2+(Lshoulder z - MidHip z)/5*3 |
| | | tail | 0 | 0.0114 | Lshoulder z+(neck z - Lshoulder z)/3 |
| 10 | foot.L | head | Lankle x | 0.0162 | Lankle z |
| | | tail | Lankle x | -0.0934 | LBigToe z |
| 11 | foot.R | head | Rankle x | 0.0162 | Rankle z |
| | | tail | Rankle x | -0.0934 | RBigToe z |
| 12 | spine.004 | head | 0 | 0.0114 | Lshoulder z+(neck z - Lshoulder z)/3 |
| | | tail | 0 | -0.0130 | neck z |
| 13 | shoulder.L | head | 0+(Lhip x - Rhip x)/8 | -0.0684 | Lshoulder z |
| | | tail | Lshoulder x | 0.0205 | Lshoulder z |
| 14 | shoulder.R | head | 0-(Lhip x - Rhip x)/8 | -0.0684 | Rshoulder z |
| | | tail | Rshoulder x | 0.0205 | Rshoulder z |
| 15 | breast.L | head | Lhip x | 0.0485 | 0+(Lhip x - Rhip x)/2+(Lshoulder z - MidHip z)/5*3 |
| | | tail | Lhip x | -0.0907 | 0+(Lhip x - Rhip x)/2+(Lshoulder z - MidHip z)/5*3 |

| | | | | | |
|---|---|---|---|---|---|
| 16 | breast.R | head | Rhip x | 0.0485 | 0+(Lhip x - Rhip x)/2+(Lshoulder z - MidHip z)/5*3 |
| | | tail | Rhip x | -0.0907 | 0+(Lhip x - Rhip x)/2+(Lshoulder z - MidHip z)/5*3 |
| 17 | toe.L | head | Lankle x | -0.0934 | LBigToe z |
| | | tail | Lankle x | -0.1606 | LBigToe z |
| 18 | heel.02.L | head | LBigToe x | 0.0459 | LBigToe z |
| | | tail | LSmallToe x | 0.0459 | LBigToe z |
| 19 | toe.R | head | Rankle x | -0.0934 | RBigToe z |
| | | tail | Rankle x | -0.1606 | RBigToe z |
| 20 | heel.02.R | head | RBigToe x | 0.0459 | RBigToe z |
| | | tail | RSmallToe x | 0.0459 | RBigToe z |
| 21 | spine.005 | head | 0 | -0.0130 | neck z |
| | | tail | nose x | -0.0247 | neck z + (nose z - neck z)/4 |
| 22 | upper_arm.L | head | Lshoulder x+0.001 | 0.0267 | Lshoulder z-0.003 |
| | | tail | Lelbow x | 0.0885 | Lelbow z |
| 23 | upper_arm.R | head | Rshoulder x-0.001 | 0.0267 | Rshoulder z-0.003 |
| | | tail | Relbow x | 0.0885 | Relbow z |
| 24 | spine.006 | head | nose x | -0.0247 | neck z + (nose z - neck z)/4 |
| | | tail | nose x | -0.0247 | nose z + (nose z - neck z)/2 |
| 25 | forearm.L | head | Lelbow x | 0.0885 | Lelbow z |
| | | tail | Lwrist x | 0.0492 | Lwrist z |
| 26 | forearm.R | head | Relbow x | 0.0885 | Relbow z |
| | | tail | Rwrist x | 0.0492 | Rwrist z |
| 27 | hand.L | head | Lwrist x | 0.0492 | Lwrist z |
| | | tail | Lwrist x + 0.03 | 0.0412 | Lwrist z - 0.03 |
| 28 | hand.R | head | Rwrist x | 0.0492 | Rwrist z |
| | | tail | Rwrist x - 0.03 | 0.0412 | Rwrist z - 0.03 |

After obtaining all of the bone head and tail node coordinates of the Rigify base armature, we utilized a script to input this information into the Rigify source file. This allowed us to modify the initial coordinates of the Rigify armature to obtain matching bones. Additionally, we used Blender's internal python to automate rigging operations within the script.

As the focus of this project was on achieving the final human action effects, the automatic rigging process only targeted a rough approximation of the human body without considering detailed manipulation of fingers and facial expressions. Consequently, there

may be some errors in the final rigging result, which predominantly stem from three sources.

First, the key points recognized by OpenPose are inconsistent with the key points of the 3D human body reconstructed by PIFuHD. This project has utilized least squares to approximate the range points and remove high error points, in order to reduce errors. However, errors are inevitable due to the fact that the recognition of key points comes from two different systems, and OpenPose is a 2D recognition system that does not include the z-axis. In contrast, PIFuHD builds a 3D human body and saves it as a three-dimensional space point.

Second, there is data error when the key point coordinates in Blender are converted to the corresponding bone coordinates. Generally, rigging uses manual methods to fine-tune the position of each bone to fit the human body. However, this project extracts general formulas for automatic transformation of key points. Nonetheless, specific bone offset will still exist due to differences in human body postures, bone sizes, and positions. To address this issue, we use a correction parameter "I" to approximate the correction error. The parameter "I" is based on the height and arm span of the human body to control the generated metarig's specifications.

The last aspect is the weight distribution error of the internal algorithm of rigging. Since skeletal animation has FK (forward kinematics) and IK (inverse kinematics) narrative methods, different bones use different methods under different actions. The bone rigging algorithm inside Rigify establishes the general movement mode of all bones, making it unsuitable for all bone movements. As a result, certain errors will occur under certain specific action sequences.

## 5.3. Animation Copy

After completing the rigging process, we get a 3D model with the controller. We can add motions to this 3D model. At this stage, we use motion replication to achieve motion synchronization between the 3D model and the skeleton animation from MocapNET. Because the skeletons' names of the 3D model are inconsistent with the skeletons' names of the skeleton animation, we designed a correspondence table to realize the

correspondence of each bone. In the process of animation copying, we only need to find the corresponding target bone of the source bone to completely copy the animation of each bone.

Table 5.2 shows the bone mapping relationship to animate a static 3D human body. The source of the motion is the BVH skeletal animation file extracted from the video by MocapNET. The target of the motion is a 3D static human body with automatic rigging.

Table 5.2: Bone Mapping Table of Motion Copy

| Rigify Rig | BVH Rig from MocapNET |
|---|---|
| head | head |
| neck | neck |
| upper_arm_fk.L | lShldr |
| forearm_fk.L | lForeArm |
| hand_fk.R | rHand |
| spine_fk.003 | chest |
| spine_fk.002 | abdomen |
| spine_fk | hip |
| thigh_fk.L | lThigh |
| shin_fk.L | lShin |
| foot_fk.L | lFoot |
| toe.L | toe2-2.L |
| thigh_fk.R | rThigh |
| shin_fk.R | rShin |
| foot_fk.R | rFoot |
| toe.R | toe2-2.R |
| upper_arm_fk.R | rShldr |
| forearm_fk.R | rForeArm |
| hand_fk.L | lHand |

As Figure 5.8 shows, the 3D human can move according to skeletal animation after rigging and animation copy.

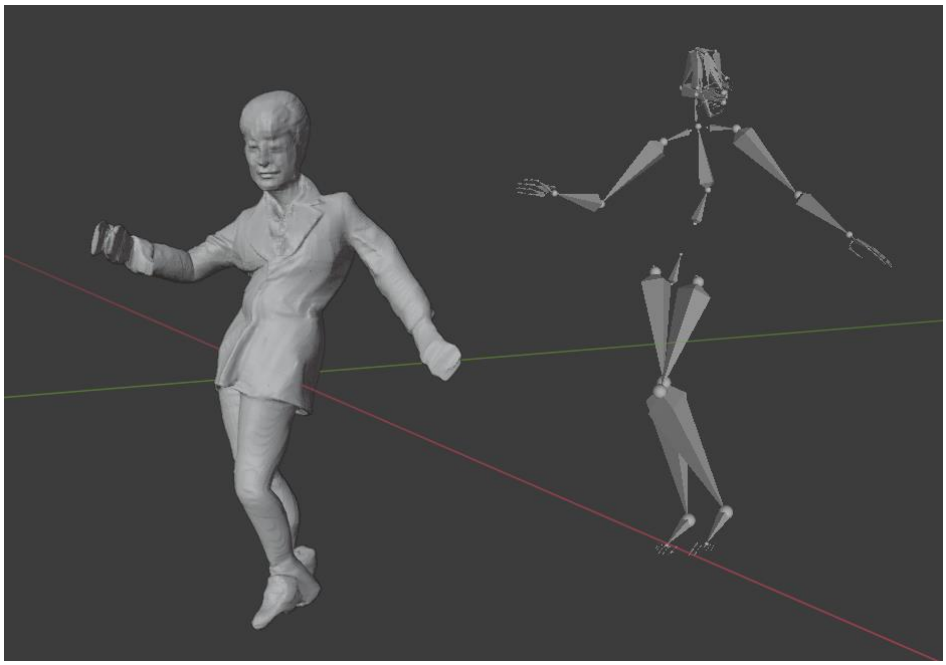

Figure 5.8: Dynamic 3D Human

## 5.4. Experiment Results and Evaluation

In this thesis, we adopted the least squares method to identify the head and tail coordinates of the Rigify skeleton and determine their positions. Subsequently, we employed Python scripts in Blender to bind the basic human body skeleton to the corresponding 3D model, achieving automated rigging. To evaluate the performance of this automatic rigging method, multiple tests were conducted.

### 5.4.1. Comparison with Other Models

The primary goal of our work is to quickly integrate BVH file and 3D humans reconstructed from a single photograph. Therefore, the rigging method employed in this paper is compared against several current automatic rigging methods mainly based on time efficiency.

The Pinocchio system, combined with a collection of motion data for several skeletons, provides an animation system that is easy to use for beginners[8]. The team of Peter Boorssan[71] introduced RigMesh that unifies the process of modeling and rigging in the 3D character animation pipeline. Hao Li and his team[72] transfer control meanings and expression dynamics from a general template to the target blend shape model, while also achieving an ideal replication of the training poses. Binh Huy Le and Zhigang Deng[73] present an optimization approach to solve a linear blend skinning model using a skeleton, which includes joint constraints and weight smoothness regularization.

In terms of time, we collected data on the number of vertices and completion time of the rigging process, and compared it to the data from the above papers. As for the above Papers[8] [71] [72] [73]: Paper[8] tested 16 different models, with three representative models representing a simple human body structure, a complex facial human body structure, and a complexly dressed human body structure. Paper[71] mainly focuses on simple mesh rigging and calculates the time of every step of the rigging process. Table 5.3 presents the time results of Paper[8], Paper[71], and our paper. Both Paper [72] and our paper used the least squares method to perform vertices position transformation. But the former method mainly concentrates on facial rigging and takes significantly longer time for automatic rigging compared to ours. Paper [73] applies multiple iterations for rigging precision. Thus, compared to our method, it has no obvious time advantage.

Table 5.3: Comparison with Pinocchio: The metrics with * denote the result of this thesis while the metrics without * represent the result of Pinocchio[8] and the result of RigMesh[71]

| Method | 3D Model | Number of Vertices | Total Time(s) | **Time/Number of Vertices(ms)** |
|---|---|---|---|---|
| Our work | Adult Male 1* | 182963 | 43.6 | **0.24** |
| | Adult Female 1* | 178806 | 41.4 | **0.23** |
| | Child 1* | 219112 | 46.8 | **0.21** |
| | Mean* | 193627 | 43.9 | **0.23** |
| Pinocchio[8] | Model 3 | 19001 | 12.6 | 0.66 |
| | Model 10 | 34339 | 56.8 | 1.65 |
| | Model 11 | 56856 | 77.1 | 1.36 |
| | Mean | 33224 | 31.3 | 0.94 |

| | | | | |
|---|---|---|---|---|
| RigMesh[71] | Step 1 | 1685 | 0.102 | 0.06 |
| | Step 2 | 1567 | 0.115 | 0.07 |
| | Step 3 | 1771 | 0.142 | 0.08 |
| | Step 4 | 1639 | 0.151 | 0.09 |
| | Step 5 | 937 | 0.096 | 0.10 |
| | Step 6 | 352 | 0.029 | 0.08 |
| | Total | 7951 | 0.635 | 0.08 |

In Table 5.3, the number of vertices in the human body in our paper is much greater than that in Pinocchio, with an average vertex count of about 5.8 times that of the latter. The larger the number of vertices, the greater the workload of rigging and weight allocation. In addition, although Pinocchio does not provide data on edges and faces, the number of points can determine the order of magnitude of edges and faces. Therefore, the difficulty of rigging in this paper is greater than that of Pinocchio. Finally, the total time spent on rigging is almost the same, although our method takes an average of 12.6 seconds more than Pinocchio, it is relatively stable and controlled at around 40 seconds. On human body files with a large number of vertices, our method takes less time than Pinocchio, as can be seen from the Time/Number of Vertices values. And for RigMesh, it has a good performance on handling simple 3D models with a very small number of vertices. Although it has least rigging time, it is not able to rig the complex 3D models of dressed human body in this research.

### 5.4.2. Accuracy Analysis

In terms of accuracy, we used Root Mean Square Error (RMSE) as the metrics to assess the discrepancy between two models. RMSE is a statistical measure that quantifies the differences between predicted and actual values, representing the average magnitude of prediction errors. RMSE is widely used for evaluating various predictive models, including regression analysis, time series analysis, and machine learning.

RMSE[74] is a statistical measure that quantifies the differences between predicted and actual values, representing the average magnitude of prediction errors. In rigging

testing, it primarily examines whether the bone model accurately reproduces the characteristics of human motion and evaluates the performance of the rigging algorithm. A smaller RMSE value indicates a higher degree of agreement between the model and data, indicating better performance of the model. For the measurement length of RMSE, the smaller the better, but the specific target depends on the application. In terms of the measurement angle of RMSE, an RMSE of less than 2 degrees is considered excellent, and an RMSE of less than 5 degrees is acceptable[75]. The formula for RMSE is as follows[74].

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n} (Y_i - f(x_i))^2} \quad (5.2)$$

N represents the sample data, where the actual value of the i-th sample is $Y_i$ and the predicted value of the model is $f(x_i)$.

The following Table 5.4 presents the RMSE values of three models in measuring angles and lengths. We tested this for adult male and female from a Chinese group aged between 21 and 87 for example.

Table 5.4: RMSE of Length and Rotation

| | RMSE-Length(m) | RMSE-Rotation(°) |
|---|---|---|
| Adult Male1 | 0.142 | 8.4 |
| Adult Male2 | 0.118 | 7.3 |
| Adult Male3 | 0.153 | 8.6 |
| Adult Male4 | 0.127 | 6.2 |
| Adult Male 5 | 0.139 | 5.3 |
| Adult Female1 | 0.126 | 5.2 |
| Adult Female2 | 0.132 | 6.1 |
| Adult Female3 | 0.163 | 9.2 |
| Adult Female4 | 0.085 | 6.8 |
| Adult Female5 | 0.142 | 7.1 |
| Mean | 0.133 | 7.0 |

As Table 5.4 shows, the average value of RMSE is 0.133 meters, indicating that there may be significant errors in the range of motion of some joints. In the field of robot control, further improvement in accuracy is required, but it is acceptable in the field of virtual reality. The main reason for the large RMSE value is that the human body coordinate points were not captured when using PIFuHD to reconstruct the 3D human body structure. Therefore, OpenPose was used to re-determine the skeleton coordinates and generate the skeleton. In other words, two different sources of mesh and skeleton lines were used for rigging in Blender. Secondly, when converting 2D coordinates into 3D coordinates, we artificially added the Y-axis to estimate the depth of the human body. These steps all have certain errors. However, the Rigify basic skeleton used does not include detailed hand and facial bone data, and small deviations in the skeleton do not have a significant impact on the rigging effect. The animation follow-up effect after rigging is within an acceptable range.

### 5.4.3. Limitation

In the results analysis, due to the impact of COVID-19, this study only tested the basic models on children, adult males, and adult females, and did not individuals with different body types and clothing. Therefore, the lack of data may result in errors and limitations in the accuracy and comprehensiveness of the test results. Hence, it is necessary to expand the testing sample in future research, covering more diverse age, gender, and feature groups to enhance the reliability and generalizability of the test results.

Regarding the time tests, firstly, the models used in this study and Pinocchio were not running on the same system, hence their data were obtained based on different models, and cannot be directly compared. Secondly, the data sources for the two models are different, and the structure of these character files also differs, resulting in potential biases in the test results. In addition, hardware conditions can also affect the rigging time, as better computer performance can accelerate the process.

During the precision test, due to the lack of ground truth for bones in the 3D human model files used in this study, manual adjustment of the bones was necessary to obtain an approximate ground truth skeleton position. When calculating the RMSE, the actual data

used was based on this skeleton generated through such manual adjustment, which may lead to certain errors. To minimize such errors, more fine-tuned manual adjustment methods or other reliable bone localization tools can be used to obtain a more accurate ground truth skeleton and improve the reliability and precision of the test results.

# Chapter 6. Conclusion and Future Works

## 6.1. Conclusion

In integrating digital media and metaverse concepts, we implemented a fast-modeling system for DTs. This system integrates 3D reconstruction, human motion sequence recognition, rigging, and 3D animation: 1. To quickly obtain the user's 3D human surface, we refer to and improve the technique of recovering the 3D human body from a single photograph. We used matting to improve the recognition accuracy of the recovery network in the pre-processing phase of the photos.2. For 3D pose recognition, we refer to the transformation of human motion sequence files.3. We adopted and implemented three manual rigging methods. Moreover, we propose a Rigify-based automatic skeleton method through the Blender Python module. Through rigging, we combined the human 3D model obtained in step 1 and the skeleton motion file obtained in step 2 to generate the 3D animation.4. The 3D model shell and human motion sequence files in this study can come from different participants, film and TV works, and online media. So, this study achieves a low coupling association between human 3D surface and human motion.

Under the influence of COVID-19, more and more human activities that rely on close contact cannot be carried out correctly. Our system can avoid this impact by remote and fast modeling. Also, because the individual modules are low-coupled, our proposed framework can progress with the improvement of the internal technology of the modules of the framework. For example, the accuracy and speed improvement of the single photo 3D recovery technique, the accuracy and speed improvement of 3D pose recognition, and the weight algorithm update of rigging can be considered later. Our project will have higher usability and scalability in the 5G era.

## 6.2. Future Works

There are still plenty of directions in which our system can be expanded. Our research is mainly about integration and implementation on pc. It will be precious for project porting and application development on mobiles such as Android and Apple systems:

1. The existing 3D deep learning models are relatively large in volume. The current hardware capability of cell phones is significantly weaker than that of PCs, so it is imperative to simplify the model when porting the pre-trained model.
2. Take Android as an example; unlike TensowFlow[9] model and PyTorch[10] model, the available model for Android is TensowFlow lite model. The study of the lossless transformation of different model types is also of great interest.
3. Application development for mobile requires additional learning costs and technical support.

Research in this area is fundamental to mobile development.

The dataset with ground truth is not used in this study. In single photo recovery 3D human technology, ground truth can be approximated as body scans from high-precision hardware devices such as Kinect and XTOM. Using more excellent datasets for training the model is a valuable improvement.

The automatic rigging algorithm proposed in this study still needs to be corrected in the final, rigging results. Furthermore, we can combine 3D pose detection and 3D body recovery by building in human key point annotation before the work of 3D body recovery. Besides, Improving the rigging weight assignment algorithm is also desirable.

[9] https://www.tensorflow.org/
[10] https://pytorch.org/

# References


[1] A. El Saddik, “Digital Twins: The Convergence of Multimedia Technologies,” *IEEE Multimed.*, vol. 25, no. 2, pp. 87–92, Apr. 2018, doi: 10.1109/MMUL.2018.023121167.

[2] A. Porterfield and T. A. M. Lamar, “Examining the effectiveness of virtual fitting with 3D garment simulation,” *Int. J. Fash. Des. Technol. Educ.*, vol. 10, no. 3, pp. 320–330, Sep. 2017, doi: 10.1080/17543266.2016.1250290.

[3] T. Bebie and H. Bieri, “A Video-Based 3D-Reconstruction of Soccer Games,” *Comput. Graph. Forum*, vol. 19, no. 3, pp. 391–400, 2000, doi: 10.1111/1467-8659.00431.

[4] S.-R. Ke, L. Zhu, J.-N. Hwang, H.-I. Pai, K.-M. Lan, and C.-P. Liao, “Real-Time 3D Human Pose Estimation from Monocular View with Applications to Event Detection and Video Gaming,” in *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, Aug. 2010, pp. 489–496. doi: 10.1109/AVSS.2010.80.

[5] J. Carr, “Surface reconstruction in 3D medical imaging,” 1996, doi: 10.26021/2870.

[6] S. Saito, T. Simon, J. Saragih, and H. Joo, “Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 84–93.

[7] A. Qammaz and A. A. Argyros, “MocapNET: Ensemble of SNN Encoders for 3D Human Pose Estimation in RGB Images.,” in *BMVC*, 2019, p. 46.

[8] I. Baran and J. Popović, “Automatic rigging and animation of 3D characters,” *ACM Trans. Graph.*, vol. 26, no. 3, pp. 72-es, Jul. 2007, doi: 10.1145/1276377.1276467.

[9] B. Foundation, “blender.org - Home of the Blender project - Free and Open 3D Creation Software,” *blender.org*. https://www.blender.org/ (accessed Mar. 21, 2023).

[10] “Maya Software | Get Prices and Buy Maya 2023 | Autodesk.” https://www.autodesk.com/products/maya/overview (accessed Mar. 21, 2023).

[11] “3D Modeling, Texturing, Lighting, Animation and Simulation Software |…,” *Maxon*. https://www.maxon.net/en/cinema-4d (accessed Mar. 21, 2023).

[12] “3ds Max Software | Get Prices & Buy Official 3ds Max 2023 | Autodesk.” https://www.autodesk.com/products/3ds-max/overview (accessed Mar. 21, 2023).

[13] “ZBrush - The all-in-one-digital sculpting solution.” https://pixologic.com/ (accessed Mar. 21, 2023).

[14] “Houdini - 3D modeling, animation, VFX, look development, lighting and rendering | SideFX.” https://www.sidefx.com/ (accessed Mar. 21, 2023).

[15] “Rhino - Rhinoceros 3D.” https://www.rhino3d.com/en/ (accessed Mar. 21, 2023).

[16] “Modo | Creative 3D modeling, animation, texturing and rendering tools.” https://www.foundry.com/products/modo (accessed Mar. 21, 2023).

[17] P. Warden and D. Situnayake, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O’Reilly Media, Inc., 2019.

[18] S. Lin, A. Ryabtsev, S. Sengupta, B. L. Curless, S. M. Seitz, and I. Kemelmacher-Shlizerman, “Real-time high-resolution background matting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8762–8771.

[19] K. S. Arun, “Transactions on Pattern Analysis and Machine Intelligence,” *IEEE Vol PAMI-9*, no. 5, pp. 698–770, 1987.
[20] J. A. Díaz-García and F. J. Caro-Lopera, “Estimation of mean form and mean form difference under elliptical laws,” *Electron. J. Stat.*, vol. 11, no. 1, pp. 2424–2460, 2017.
[21] E. Amid and M. K. Warmuth, “TriMap: Large-scale Dimensionality Reduction Using Triplets.” arXiv, Mar. 25, 2022. doi: 10.48550/arXiv.1910.00204.
[22] A. Levin, A. Rav-Acha, and D. Lischinski, “Spectral Matting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1699–1712, Oct. 2008, doi: 10.1109/TPAMI.2008.168.
[23] E. S. Gastal and M. M. Oliveira, “Shared sampling for real-time alpha matting,” in *Computer Graphics Forum*, Wiley Online Library, 2010, pp. 575–584.
[24] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum, “Poisson matting,” in *ACM SIGGRAPH 2004 Papers*, 2004, pp. 315–321.
[25] B. Zhu, Y. Chen, J. Wang, S. Liu, B. Zhang, and M. Tang, “Fast deep matting for portrait animation on mobile phone,” in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 297–305.
[26] S. Sengupta, V. Jayaram, B. Curless, S. M. Seitz, and I. Kemelmacher-Shlizerman, “Background Matting: The World Is Your Green Screen,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 2288–2297. doi: 10.1109/CVPR42600.2020.00236.
[27] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, “Keep It SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image,” in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 561–578. doi: 10.1007/978-3-319-46454-1_34.
[28] D. S. Alexiadis, D. Zarpalas, and P. Daras, “Real-Time, Full 3-D Reconstruction of Moving Foreground Objects From Multiple Consumer Depth Cameras,” *IEEE Trans. Multimed.*, vol. 15, no. 2, pp. 339–358, Feb. 2013, doi: 10.1109/TMM.2012.2229264.
[29] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, “Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach,” presented at the Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 398–407. Accessed: Mar. 19, 2023. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2017/html/Zhou_Towards_3D_Human_ICCV_2017_paper.html
[30] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, “Learning to Reconstruct 3D Human Pose and Shape via Model-Fitting in the Loop,” presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2252–2261. Accessed: Sep. 21, 2022. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2019/html/Kolotouros_Learning_to_Reconstruct_3D_Human_Pose_and_Shape_via_Model-Fitting_ICCV_2019_paper.html
[31] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.
[32] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, “End-to-end recovery of human shape and pose,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7122–7131.

[33] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler, "Unite the People: Closing the Loop Between 3D and 2D Human Representations," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6050–6059. Accessed: Sep. 21, 2022. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Lassner_Unite_the_People_CVPR_2017_paper.html

[34] H. Onizuka, Z. Hayirci, D. Thomas, A. Sugimoto, H. Uchiyama, and R. Taniguchi, "TetraTSDF: 3D human reconstruction from a single image with a tetrahedral outer shell," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6011–6020.

[35] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph. TOG*, vol. 34, no. 6, pp. 1–16, 2015.

[36] "Principal component analysis: a review and recent developments | Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences." https://royalsocietypublishing.org/doi/full/10.1098/rsta.2015.0202 (accessed Mar. 19, 2023).

[37] G. Pavlakos *et al.*, "Expressive Body Capture: 3D Hands, Face, and Body From a Single Image," presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10975–10985. Accessed: Sep. 27, 2022. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2019/html/Pavlakos_Expressive_Body_Capture_3D_Hands_Face_and_Body_From_a_CVPR_2019_paper.html

[38] M. Kocabas, N. Athanasiou, and M. J. Black, "VIBE: Video Inference for Human Body Pose and Shape Estimation," presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5253–5263. Accessed: Sep. 27, 2022. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Kocabas_VIBE_Video_Inference_for_Human_Body_Pose_and_Shape_Estimation_CVPR_2020_paper.html

[39] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "AMASS: Archive of Motion Capture As Surface Shapes," presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 5442–5451. Accessed: Mar. 19, 2023. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2019/html/Mahmood_AMASS_Archive_of_Motion_Capture_As_Surface_Shapes_ICCV_2019_paper.html

[40] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization," presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2304–2314. Accessed: Sep. 27, 2022. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2019/html/Saito_PIFu_Pixel-Aligned_Implicit_Function_for_High-Resolution_Clothed_Human_Digitization_ICCV_2019_paper.html

[41] S. Xu, "The Research on Applying Artificial Intelligence Technology to Virtual YouTuber," in *2021 IEEE International Conference on Robotics, Automation and Artificial Intelligence (RAAI)*, Apr. 2021, pp. 10–14. doi: 10.1109/RAAI52226.2021.9507778.

[42] A. Toshev and C. Szegedy, “DeepPose: Human Pose Estimation via Deep Neural Networks,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1653–1660. Accessed: Apr. 23, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2014/html/Toshev_DeepPose_Human_Pose_2014_CVPR_paper.html

[43] J. K. Aggarwal and L. Xia, “Human activity recognition from 3D data: A review,” *Pattern Recognit. Lett.*, vol. 48, pp. 70–80, Oct. 2014, doi: 10.1016/j.patrec.2014.04.011.

[44] R. G. Díaz, F. Laamarti, and A. El Saddik, “DTCoach: Your Digital Twin Coach on the Edge During COVID-19 and Beyond,” *IEEE Instrum. Meas. Mag.*, vol. 24, no. 6, pp. 22–28, Sep. 2021, doi: 10.1109/MIM.2021.9513635.

[45] “MTI | Free Full-Text | Deep Learning-Enabled Multitask System for Exercise Recognition and Counting.” https://www.mdpi.com/2414-4088/5/9/55 (accessed Jun. 17, 2023).

[46] Y. Chen, Y. Tian, and M. He, “Monocular human pose estimation: A survey of deep learning-based methods,” *Comput. Vis. Image Underst.*, vol. 192, p. 102897, 2020.

[47] “Indirect deep structured learning for 3D human body shape and pose prediction.” https://www.repository.cam.ac.uk/handle/1810/274296 (accessed Apr. 23, 2023).

[48] S. Li, Z.-Q. Liu, and A. B. Chan, “Heterogeneous Multi-task Learning for Human Pose Estimation with Deep Convolutional Neural Network,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 482–489. Accessed: Apr. 23, 2023. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_workshops_2014/W15/html/LI_Heterogeneous_Multi-task_Learning_2014_CVPR_paper.html

[49] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua, “Structured Prediction of 3D Human Pose with Deep Neural Networks.” arXiv, May 17, 2016. doi: 10.48550/arXiv.1605.05180.

[50] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields.” arXiv, May 30, 2019. doi: 10.48550/arXiv.1812.08008.

[51] I. A. Faisal, T. W. Purboyo, and A. S. R. Ansori, “A Review of accelerometer sensor and gyroscope sensor in IMU sensors on motion capture,” *J Eng Appl Sci*, vol. 15, no. 3, pp. 826–829, 2019.

[52] G. Cooper *et al.*, “Inertial sensor-based knee flexion/extension angle estimation,” *J. Biomech.*, vol. 42, no. 16, pp. 2678–2685, 2009.

[53] X. L. Meng, Z. Q. Zhang, S. Y. Sun, J. K. Wu, and W. C. Wong, “Biomechanical model-based displacement estimation in micro-sensor motion capture,” *Meas. Sci. Technol.*, vol. 23, no. 5, p. 055101, 2012.

[54] “Automatic rigging for animation characters with 3D silhouette - Pan - 2009 - Computer Animation and Virtual Worlds - Wiley Online Library.” https://onlinelibrary.wiley.com/doi/abs/10.1002/cav.284 (accessed Sep. 28, 2022).

[55] "Robust and accurate skeletal rigging from mesh sequences | ACM Transactions on Graphics." https://dl.acm.org/doi/abs/10.1145/2601097.2601161 (accessed Sep. 28, 2022).
[56] A. Feng, D. Casas, and A. Shapiro, "Avatar Reshaping and Automatic Rigging Using a Deformable Model," May 2015. doi: 10.1145/2822013.2822017.
[57] Z. Bhatti, A. Shah, A. Waqas, and N. Mahmood, "Analysis of Design Principles and Requirements for Procedural Rigging of Bipeds and Quadrupeds Characters with Custom Manipulators for Animation," *Int. J. Comput. Graph. Animat.*, vol. 5, no. 1, pp. 47–67, Jan. 2015, doi: 10.5121/ijcga.2015.5104.
[58] M. Forte and F. Pitié, "$F$, $B$, Alpha Matting." arXiv, Mar. 17, 2020. Accessed: Mar. 24, 2023. [Online]. Available: http://arxiv.org/abs/2003.07711
[59] G. Borgefors, "Distance transformations in digital images," *Comput. Vis. Graph. Image Process.*, vol. 34, no. 3, pp. 344–371, Jun. 1986, doi: 10.1016/S0734-189X(86)80047-0.
[60] M. Akmal Butt and P. Maragos, "Optimum design of chamfer distance transforms," *IEEE Trans. Image Process.*, vol. 7, no. 10, pp. 1477–1484, Oct. 1998, doi: 10.1109/83.718487.
[61] A. Rosenfeld and J. L. Pfaltz, "Distance functions on digital pictures," *Pattern Recognit.*, vol. 1, no. 1, pp. 33–61, Jul. 1968, doi: 10.1016/0031-3203(68)90013-7.
[62] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa, "Fast directional chamfer matching," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2010, pp. 1696–1703. doi: 10.1109/CVPR.2010.5539837.
[63] M. R. Arshad, K. Yoon, and A. Manaf, "Physical Rigging Procedures Based on Character Type and Design in 3D Animation," vol. 8, p. 4138, Oct. 2019, doi: 10.35940/ijrte.C5484.098319.
[64] "Blender Foundations | The Essential Guide to Learning Blender 2.5 | Ro." https://www.taylorfrancis.com/books/mono/10.4324/9780240814315/blender-foundations-roland-hess (accessed Jan. 01, 2023).
[65] "Open source rigging in Blender: A modular approach - ProQuest." https://www.proquest.com/openview/c91b02848aef077bbe4f56c846b6d06e/1?pq-origsite=gscholar&cbl=18750 (accessed Jan. 01, 2023).
[66] T. Mullen, *Mastering Blender*. John Wiley & Sons, 2011.
[67] P. Raju, "Bones RiggingBones rigging," in *Character Rigging and Advanced Animation : Bring Your Character to Life Using Autodesk 3ds Max*, P. Raju, Ed., Berkeley, CA: Apress, 2019, pp. 77–118. doi: 10.1007/978-1-4842-5037-2_4.
[68] S. Blackman, "Rigging with Mixamo," in *Unity for Absolute Beginners*, S. Blackman, Ed., Berkeley, CA: Apress, 2014, pp. 565–573. doi: 10.1007/978-1-4302-6778-2_12.
[69] "Mixamo." https://www.mixamo.com/#/ (accessed Jan. 05, 2023).
[70] Å. Björck, "Least squares methods," in *Handbook of Numerical Analysis*, Elsevier, 1990, pp. 465–652. doi: 10.1016/S1570-8659(05)80036-5.
[71] P. Borosán, M. Jin, D. DeCarlo, Y. Gingold, and A. Nealen, "RigMesh: automatic rigging for part-based shape modeling and deformation," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 1–9, Nov. 2012, doi: 10.1145/2366145.2366217.

[72] H. Li, T. Weise, and M. Pauly, “Example-based facial rigging,” *ACM Trans. Graph.*, vol. 29, no. 4, p. 32:1-32:6, Jul. 2010, doi: 10.1145/1778765.1778769.
[73] B. H. Le and Z. Deng, “Robust and accurate skeletal rigging from mesh sequences,” *ACM Trans. Graph.*, vol. 33, no. 4, pp. 1–10, Jul. 2014, doi: 10.1145/2601097.2601161.
[74] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature,” *Geosci. Model Dev.*, vol. 7, no. 3, pp. 1247–1250, Jun. 2014, doi: 10.5194/gmd-7-1247-2014.
[75] P. Slade, A. Habib, J. L. Hicks, and S. L. Delp, “An Open-Source and Wearable System for Measuring 3D Human Motion in Real-Time,” *IEEE Trans. Biomed. Eng.*, vol. 69, no. 2, pp. 678–688, Feb. 2022, doi: 10.1109/TBME.2021.3103201.