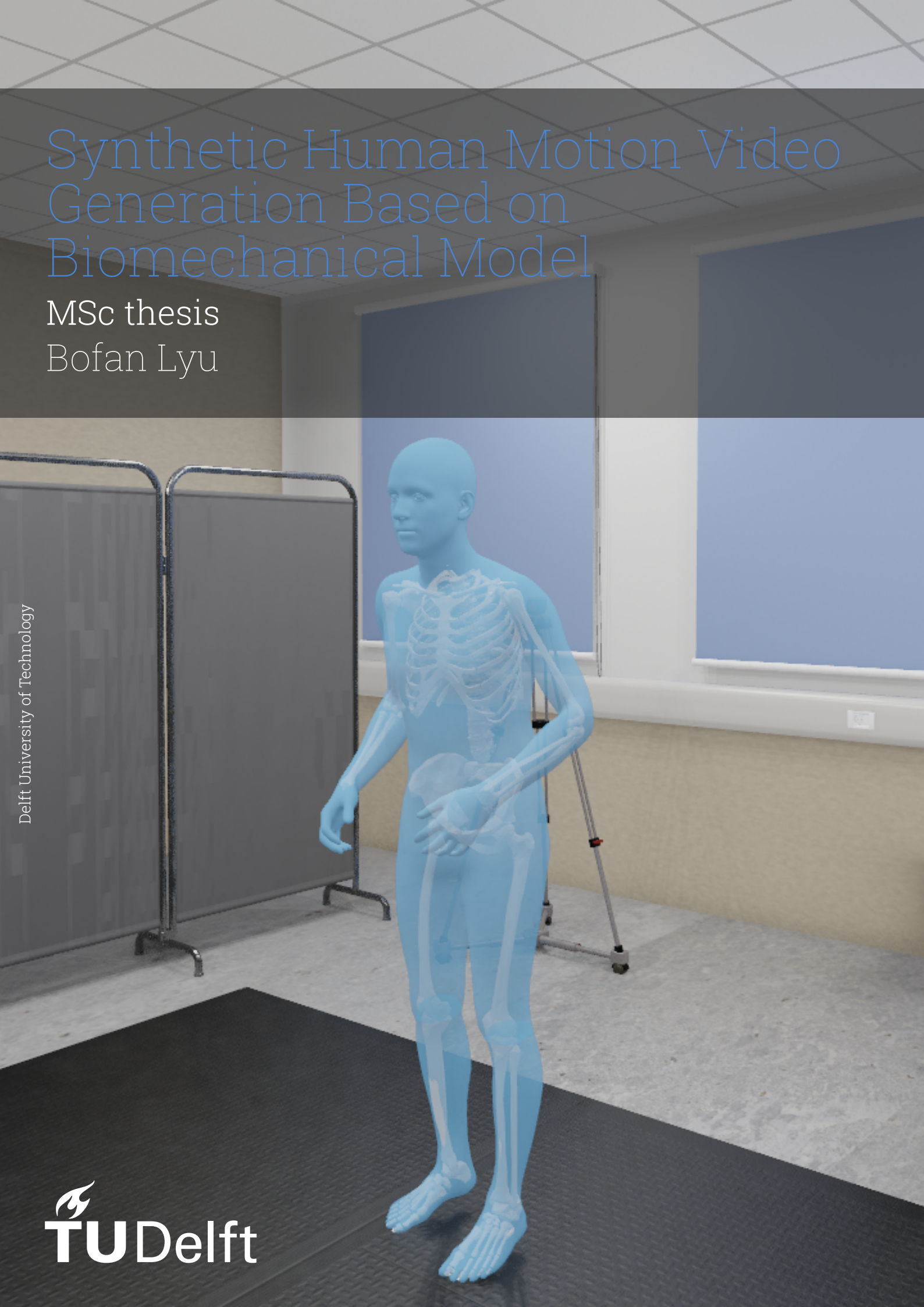
Synthetic Human Motion Video Generation Based on Biomechanical Model

MSc thesis

Bofan Lyu



Delft University of Technology

TU Delft

Synthetic Human Motion Video Generation Based on Biomechanical Model

by

Bofan Lyu

At the:
Department of Biomechanical Engineering,
Delft University of Technology
to be defended publicly on Thursday December 21, 2023

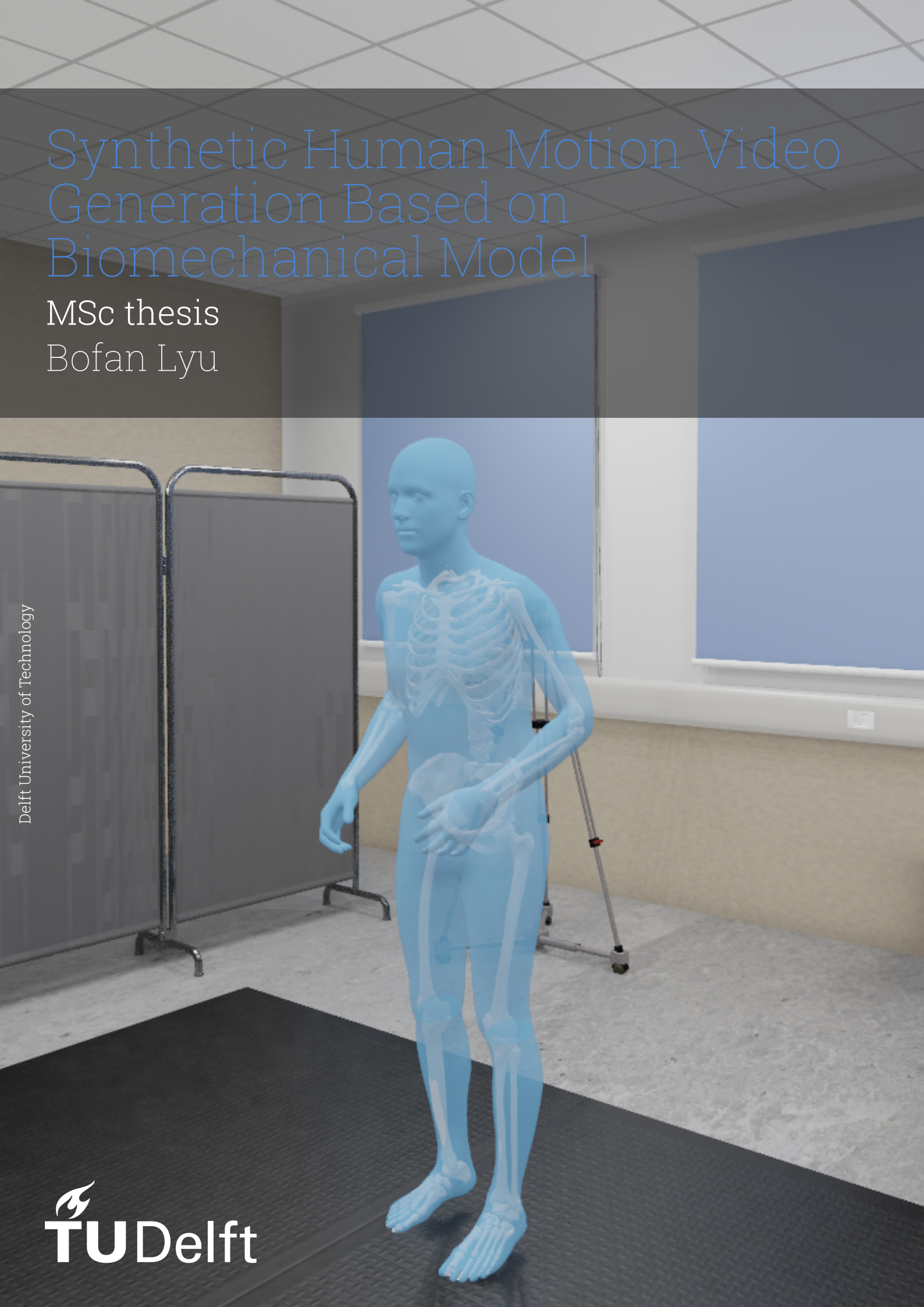
Student number:	5519268
Supervisor:	Dr. A. Seth
Thesis committee:	Dr. E. van der Kruk
	Dr. X. Zhang
	Prof. dr. F. C. T. van der Helm
Project Duration:	Sep, 2022 - Dec, 2023

TU Delft

Synthetic Human Motion Video Generation Based on Biomechanical Model

MSc thesis

Bofan Lyu



Delft University of Technology

TU Delft

Synthetic Human Motion Video Generation Based on Biomechanical Model

by

Bofan Lyu

At the:
Department of Biomechanical Engineering,
Delft University of Technology
to be defended publicly on Thursday December 21, 2023

Student number:	5519268
Supervisor:	Dr. A. Seth
Thesis committee:	Dr. E. van der Kruk
	Dr. X. Zhang
	Prof. dr. F. C. T. van der Helm
Project Duration:	Sep, 2022 - Dec, 2023

TU Delft

Contents

1 Introduction **1**

2 Method **3**

2.1 Pipeline description 3

2.1.1 Overview 3

2.1.2 Biomechanical model 3

2.1.3 Human mesh model 4

2.1.4 MoSh++ 4

2.1.5 Joint regressor 5

2.1.6 OpenSim model based fitting 7

2.1.7 Rendering setup 7

2.2 Pipeline validation 8

2.2.1 Datasets 10

3 Results **10**

3.1 Visual Evaluation 10

3.2 Quantitative Evaluation 10

4 Discussion, Limitation and Future Work **12**

Abstract

Biomechanics studies the underlying mechanisms between body movements and forces. Accurate motion data is crucial for the biomechanics. Currently, marker-based motion capture systems are often used by researchers to record motion data. Marker-based motion capture systems are not widely adopted due to its drawbacks in terms of financial and time costs, portability, etc. Video-based motion capture systems can record motions using videos collected by webcams, cameras, and smartphones as the input and then estimate human motions from those videos. A simpler setup makes video-based motion capture technology more accessible for widespread use. However, existing motion capture datasets commonly lack of biomechanically accurate annotation, resulting in a deficiency in the biomechanical accuracy of exsisting video-based motion capture methods. In the biomechanics community, there are a lot of validated and biomechanically accurate models and motion data; however, corresponding video data is lacking. Therefore, we aim to construct human-like appearance based on these data and generate a synthetic human motion video dataset using 3D graphic software. In this thesis, we purposed a pipeline that can generate synthetic human motion videos. The pipeline takes a subject-specific OpenSim model and motion as input and uses SMPL-X model to generate human-like appearance. We validated the synthetic data generated by our pipeline and demonstrated the biomechanical reliability of the pipeline. Using this pipeline, we created synthetic dataset ODAH with biomechanically accurate annotations for neural network training.

Keywords: Motion capture, biomechanical model, OpenSim, synthetic video, SMPL-X model

1 Introduction

Biomechanics is the study of forces acting on the body and their effect. Researchers study human movement to understand its underlying mechanism. Biomechanics influenced applications in rehabilitations, sports, clinical diagnosis, et al. To analyse human motions, it is necessary to capture human motions from reality. Currently, marker-based motion capture systems can achieve very high accuracy, and they are often regarded as the gold standard in motion capture [1]. In marker-based motion capture systems, a set of markers are attached on the subject. To avoid the influence of clothing deformation, the subject need to wear a tight-fitting suit, and markers are usually placed at specific bony landmarkers where bones are very close to the skin. Light either emitted or reflected from the markers is picked up by high speed motion capture cameras. Thus marker-based motion capture is usually conducted indoor [1] where lighting conditions are ideal. Then the markers' positions in 3D space are calculated using triangulation. In specific motion capture data processing software, the collected marker position data are derived to the motions of a skeleton template using inverse kinematics. Although marker-based motion capture systems can provide accurate results, they haven't been adopted widely in biomechanical applications. It is partly due to its complex and costly hardware setup, and time consuming marker placement. Another reason is that marker-based motion capture is not suitable for all tasks. It is reported that subjects may not perform natural movements due to feeling uncomfortable having markers attached to the body and being recorded in an unfamiliar lab environment [2]. In some certain scenarios, such as capturing the motion of an athlete in a sports competition, it is not feasible to attach markers on the subject. In general, marker-based motion capture systems show shortcomings in cost, portability, and ease of use. Therefore, it is difficult to widely adopt marker-based motion capture systems in clinical and personal applications, which hinders researchers in biomechanics from collecting human movement data [2, 3].

There is a need for motion capture methods that are easier to use and less costly. Video-based motion capture systems provide a more feasible solution. Video-based motion capture systems take input from smartphone, webcam, or internet video to record motions, and often leveraging deep learning-based algorithm to estimate body scale and motion [4]. Once videos are recorded, pose estimation algorithms are then utilized to estimate 3D human joints. Current pose estimation algorithms mostly rely on deep learning networks [4]. These methods are 'trained' on large scale datasets that provide video input from single view or multi-views and annotated 3D poses for each time frame. During training, the pose estimation algorithms are optimized to minimize the difference between estimation and ground truth annotation. Video-based motion estimation methods have showed their potential in biomechanic applications, such as gait analysis [5], rehabilitation [6], sports training [7–9].

Deep learning algorithms can learn the annotation of its training dataset includes bias and errors. A lot of motion capture datasets are recorded by marker-based methods [10–12]. Although marker-based methods are often considered as the "gold standard", the results provided by these methods also have a certain margin of error. Markers placed on subjects are assumed to be fixed to bones. In reality, due to soft tissue artifact, markers can move relative to bones, which leads to error in motions. For example, soft tissue artifact can result in an error of 5.8 degrees in hip internal rotation estimation [13]. And marker-based methods are relatively inaccurate in predicting rotation outside the sagittal plane because markers are placed closer to the axis of rotation [14]. The process of transferring marker data to motions also introduces errors. Due to these reasons, marker-based motion capture methods cannot perfectly record human motions. Neural networks are trained based on a training set, which means the limit of their accuracy is restricted to the accuracy of the training set. It actually obstructs the potential of video-based motion capture methods.

To solve the issue of pseudo ground truth in existing motion capture datasets, researchers

purposed to train networks on synthetic datasets which can provide "perfect" ground truth annotations. Many synthetic datasets have been created for neural network training. Surreal [15] is a large scale synthetic human motion dataset. Methods trained on Surreal achieved good accuracy in human depth estimation, human part segmentation [15] and human pose estimation [16] on real RGB images. Michael et al. created BEDLAM [17]. They trained a neural network solely on synthetic data and achieved state-of-the-art on human pose and shape estimation from real RGB images. These studies suggest that synthetic datasets have the potential to replace existing datasets as training sets for neural networks.

A common issue for both current open-source real world motion capture dataset and synthetic dataset is that these datasets are not designed for biomechanical applications [4]. In some datasets [18, 19], joint centers are manually labeled by people who have no anatomical knowledge. Some datasets employ marker-based systems to record motion data, while not using biomechanical human model [11, 20] when transferring marker data to human motions. Synthetic databases also face the same issue. Although they can reconstruct human-like external appearance, the underlying skeleton cannot represent an actual human skeleton [3] and thus these synthetic datasets and data generation pipelines cannot provide biomechanically accurate ground truth.

Having an appropriate model is crucial for biomechanical research. Musculoskeletal models are commonly used in biomechanical simulation. A musculoskeletal model includes: a well defined skeletal structure, muscle actuators, anatomical joints [21]. Such model can be implemented directly into the work flow of biomechanical research. Attempts have been made to combine video-based motion capture systems with musculoskeletal models and joint angles. Theia3D [22] is a commercially available AI-based 3D markerless motion capture solution for biomechanics. Their system requires at least 6 cameras to capture human motions. Unfortunately, its data source and algorithm are not publicly available. OpenCap [23] is an open-source online platform that can estimate the biomechanical model and its kinematics and dynamics from a setup with at least two cameras. OpenCap first derives 3D keypoint locations using deep learning model and then obtain joint angles and model scaling using OpenSim Scale tool and Inverse Kinematic tool. Such multi-step approaches may introduce variability and inconsistency. D3KE [24] is an end-to-end method that can predict biomechanical model and its kinematics from monocular videos. In general, end-to-end methods are more preferable due to higher reliability and accuracy [24].

As mentioned above, existing motion capture datasets lack accurate biomechanical annotations. Current open-source datasets cannot provide a large scale of data to train a neural network for biomechanical pose estimation. It is also very time consuming to build a large scale biomechanical motion capture dataset from the beginning. Existing massive open-source biomechanical motion data should be used to build such dataset. Therefore, creating synthetic dataset based on biomechanical models and motions can be a potential solution. Schleicher et al. [25] purposed Biomechanical Animated Skinned Human (BASH) that can rig a human skin mesh based on an OpenSim model. However, BASH was designed to demonstrate muscle activation instead of generating realistic synthetic data for network training. The biomechanical accuracy of their method is questionable.

We proposed a method that can generate synthetic human motion videos based on a biomechanical skeleton. In comparison to existing motion capture datasets, our synthetic data generation pipeline offers numerous advantages. We utilized biomechanical models and motions as inputs, providing trustworthy annotation for biomechanical research. Our pipeline allows the easy generation of a large amount of data based on existing dataset, and users are free to set backgrounds, lighting condition, camera parameters, etc. in 3D graphics software to create data based on their needs.

The main contributions of our study are:

- A pipepline to generate synthetic video given a motion of a biomechanical (OpenSim) model.
- A synthetic video dataset with ground truth joint angle trajectories for biomechanical models, varied subject appearance, motions, and scene settings.

2 Method

2.1 Pipeline description

2.1.1 Overview

Human-like mesh and biomechanical skeletal model are two of the primary components in our pipeline. We employed SMPL-X [26] mesh model and OpenSim fullbody skeletal model [27] to achieve our goal. We developed a synthetic data generation pipeline to rig an SMPL-X [26] model against a full-body OpenSim skeletal model [27] and its associated joint angles that produce a variety of human motions (See Figure 1). The OpenSim skeletal model and SMPL-X model used in our pipeline are introduced in section 2.1.2 and 2.1.3 respectively. The OpenSim skeletal model and the joint angles serve as the inputs to the pipeline. We used MoSh++ [28] to provide initial guess for our data generation pipeline, which is introduced in section 2.1.4. The subject-specific SMPL-X models were fit to the joint and marker locations extracted from the OpenSim skeletal model. In section 2.1.5, we trained a joint regressor that can indicate OpenSim skeletal structure from SMPL-X mesh. Then, our OpenSim based fitting process is described in section 2.1.6. Our rendering setup is described in section 2.1.7

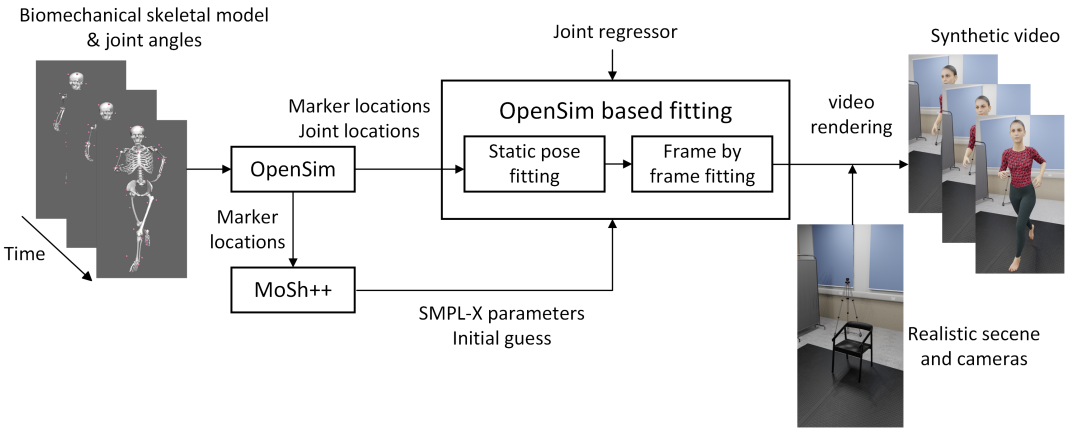


Figure 1: An overview of our synthetic data generation pipeline. We first trained a joint regressor that represents the mapping from the SMPL-X mesh to the OpenSim skeleton. For given OpenSim model, we then optimized the body shape and motion to achieve a best fit to subject-specific OpenSim skeletal model and joint angles. Finally, we built a realistic environment in 3D graphic software with scene and cameras to render the synthetic video data.

2.1.2 Biomechanical model

An OpenSim model is composed of rigid bodies (bones) that are connected by joints. Joints connect two reference frames: one on the parent and one on the child body of the joint, which

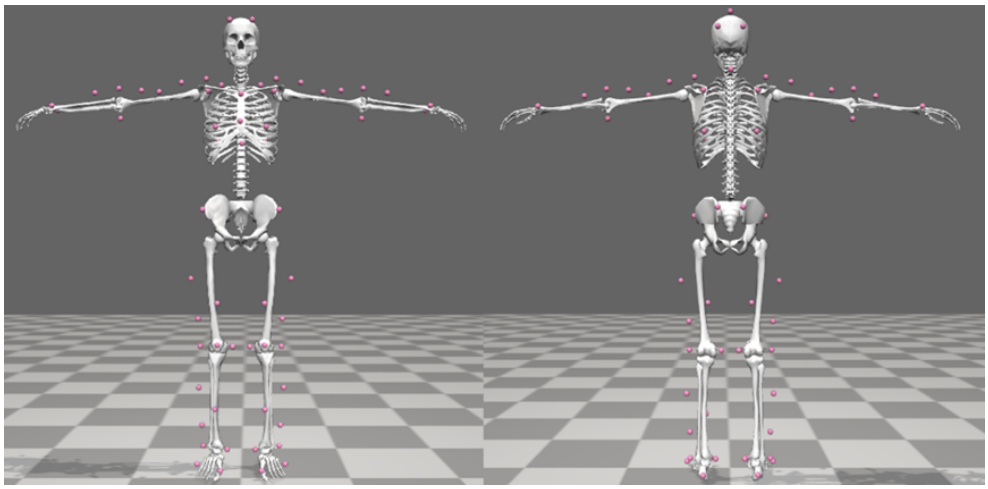


Figure 2: The OpenSim skeletal model used in our pipeline. The model has 22 rigid bodies in total. There are 13 bodies in the lower body (pelvis, right and left femur, patella, tibia and fibula, talus, calcaneus, toes), and 9 in the upper body (torso and head, right and left humerus, ulna, radius, hand). The total number of DOFs is 37 (20 in the lower body and 17 in the upper body). This model is created and validated by Rajagopal et al. [27]

.

coincide in space at the joint center. The fullbody model [27] applied in the pipeline is designed for gait simulation. Figure 2 shows the front and back view of the model. It has 22 rigid bodies and 37 degrees of freedom. To assist 3D human skin generation, a set of markers is attached to the model. These markers are place on bony landmarks where bones are very close to the skin.

2.1.3 Human mesh model

We used Skinned Multi-Person Linear expressive (SMPL-X) model [26] as our 3D human mesh model. SMPL-X is a skinned vertex-based model that can produce various natural human body shapes in different poses. The mesh consists of 10,475 vertices and 20,908 faces, driven by an internal armature with 54 joints. A SMPL-X mesh can be defined by 2 parameters: shape β and pose θ. The SMPL-X mesh is based on a template mesh. Given a shape parameter β, the SMPL-X model calculates the displacement for each vertex based on a shape displacement matrix. The model utilizes the SMPL-X joint regressor to obtain the internal armature. Vertices are then driven by the internal armature to obtain the human meshes at given pose. Figure 2.1.3 shows some example SMPL-X meshes. The SMPL-X model can generate various human-like appearances. By optimizing these two parameters, we can find an optimal SMPL-X mesh that best fits the given subject-specific musculoskeletal model and joint angles.

2.1.4 MoSh++

We utilized MoSh++ (Motion and Shape capture) [28] to create the initial human mesh. Given marker trajectories and a marker layout on the mesh, MoSh++ generated SMPL-X mesh sequences. The marker layout was defined manually by identifying corresponding SMPL-X vertices as OpenSim markers. From MoSh++, we obtained the initial SMPL-X shape β and pose θ for the given skeletal model and its motion.

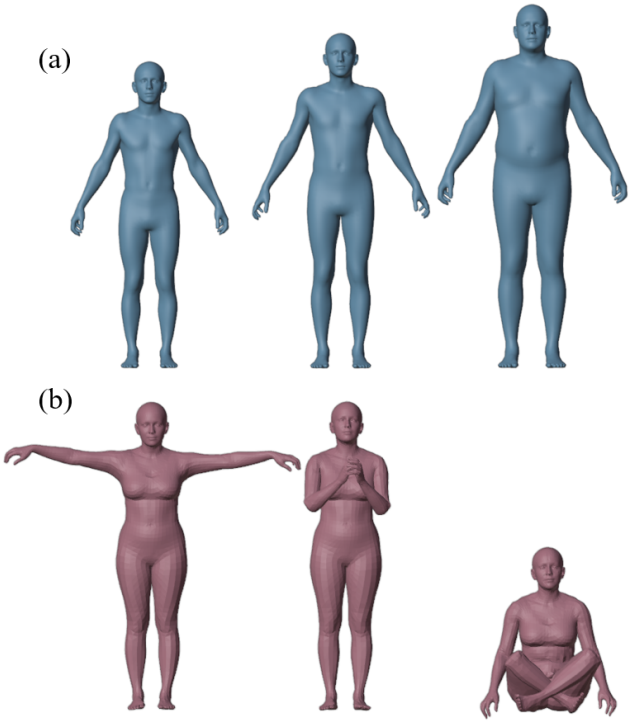


Figure 3: Some example SMPL-X meshes. (a) Meshes with various body shape can be obtained by changing the shape parameter β. (b) SMPL-X model can generate human body meshes in various poses by changing the pose parameter θ.

2.1.5 Joint regressor

SMPL-X and OpenSim model do not share the same joint definition. The OpenSim skeletal model is based on human anatomical structures, whereas the armature of the SMPL-X model is based on matching the mesh to surface scans [29] instead of skeletal anatomy. To remove this discrepancy, we trained a joint regressor for the SMPL-X model that identifies biomechanical joint centers as joint keypoints based on the position of the SMPL-X vertices. This set of joint keypoints has anatomical-based structure same as the structure in the OpenSim skeletal model. The location of each joint keypoint are represented by a linear combination of several adjacent vertices. It allows us to align the mesh and skeletal models using these joint keypoints as reference.

To obtain our joint regressor, we manually posed and scaled several OpenSim skeletons to match SMPL-X meshes, aligning them with the T-pose (see Figure 4). To avoid overfitting, it is necessary to include a variety of different poses during the process of obtaining our joint regressor. For such a massive data requirement, it is impractical to create a database entirely by manual fitting. Hence, we assumed that the offset between an OpenSim joint and its corresponding SMPL-X joint is fixed in the respective OpenSim parent body frame. (e.g., the offset of hip joints is fixed in the pelvis frame, the offset of knee joint is fixed in the femur frame).

The offset is subject dependent, meaning that different body shape may correspond to different offset values. The offset for each subjects is obtained via our manually static pose fitting. We used the output of MoSh++ as our initial guess. Under that assumption, we utilized the Adam optimizer [30] to optimize SMPL-X shape parameter β and pose parameter θ, aiming

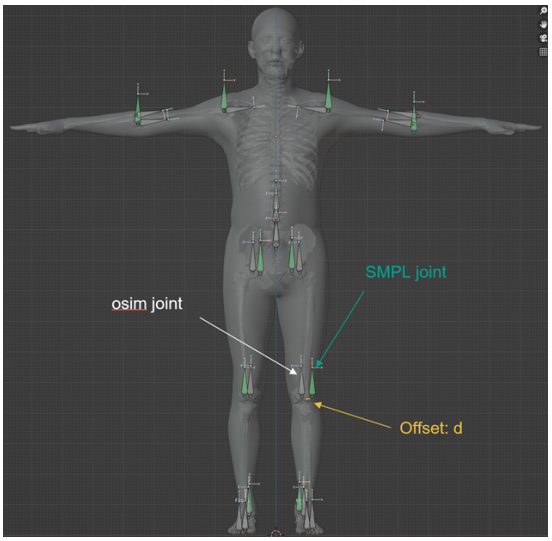


Figure 4: An example of our manually fitted skeleton and mesh. The bottom of each gray dart-shaped object corresponds to a joint center of the SMPL-X armature. The bottom of each green dart-shaped object corresponds to a joint center of the OpenSim skeletal model. We scaled and posed the OpenSim skeletal model to fit the SMPL-X mesh at static T-pose. We then assumed that the offset between each OpenSim joint center and SMPL-X joint center is fixed in the parent frame of the OpenSim joint.

to minimized the following loss:

$$\mathcal{L}_{joint} = \frac{1}{N_j N_f} \sum_{f=0}^{N_f-1} \sum_{j=0}^{N_j-1} \|\boldsymbol{p_{tgt}}_{j,f} - \boldsymbol{p}_{j,f}\|_1 \tag{1}$$

where $\boldsymbol{p}_{j,f}$ is the joint location of fitted SMPL-X mesh, and $\boldsymbol{p_{tgt}}_{j,f}$ is the target location of corresponding SMPL-X joint. N_j and N_f are number of joint and number of frames respectively.

Given OpenSim joint location, the target SMPL-X joint locations can be calculated by the following formula:

$$\boldsymbol{p_{tgt}}_{j,f} = \boldsymbol{p_{osim}}_{j,f} + \boldsymbol{R}^{\boldsymbol{world}}_{\boldsymbol{parent}} \boldsymbol{d_j} \tag{2}$$

where $\boldsymbol{p_{osim}}_{j,f}$ and $\boldsymbol{R}^{\boldsymbol{world}}_{\boldsymbol{parent}}$ are OpenSim joint location and rotation matrix of corresponding parent frame respectively. They can be obtained via OpenSim. $\boldsymbol{d_j}$ is our predefined offset.

After previous fitting steps, we obtained a series of SMPL-X vertices along with their corresponding OpenSim joint locations. We utilized an Adam optimizer [30] to obtain our joint regressor by minimizing:

$$\mathcal{L}_{JR} = \frac{1}{N_f} \sum_{f=0}^{N_f-1} \|\mathcal{J}\hat{\boldsymbol{V}}_f - \boldsymbol{P_{osim}}_f\|^2 \tag{3}$$

where $\hat{\boldsymbol{V}} \in \mathbb{R}^{N_v \times 3}$ represent the SMPL-X vertices of each frame, and $\boldsymbol{P_{osim}}_f \in \mathbb{R}^{N_j \times 3}$ represents OpenSim joints locations of each frame. $\mathcal{J} \in \mathbb{R}^{N_j \times N_v}$ (N_v, number of vertices) represents our joint regressor, which is our optimization objective.

2.1.6 OpenSim model based fitting

We used the output of MoSh++ as our initial guess, and improved the fitting by introducing additional joint based correction. The fitting process consists of two steps, static shape optimization and frame by frame pose optimization.

Static shape optimization. Given an OpenSim model, we first determine a subject-specific shape through a static trial. In this step, we optimize both the SMPL-X shape and pose parameters. The objective is to obtain the subject-specific SMPL-X shape parameter β. The Loss function is given in equation 8. Additionally, our pipeline also allows transferring the SMPL-X mesh to STAR [31] mesh which has an independent BMI related shape term, then we can modify the overall size (fat or thin) of the synthetic subject.

Frame by frame pose optimization. Then, we optimize the SMPL-X pose parameter frame by frame. The loss function consist of marker keypoint loss and joint keypoint loss, which is given as follow:

$$\mathcal{L}_{joint} = \frac{1}{N_j} \sum_{j=0}^{N_j-1} \boldsymbol{\lambda_{joint}}_j \|\hat{\boldsymbol{p}}_{\boldsymbol{Jsmpl}\,f,j} - \boldsymbol{p}_{\boldsymbol{Josim}\,f,j}\|_1 \tag{4}$$

$$\boldsymbol{p}_{\boldsymbol{Josim}\,f,j} = (\mathcal{J}\hat{\boldsymbol{V}}_{\boldsymbol{smpl}\,f})_j \tag{5}$$

$$\mathcal{L}_{marker} = \frac{1}{N_m} \sum_{m=0}^{N_m-1} \boldsymbol{\lambda_{marker}}_m \|\hat{\boldsymbol{p}}_{\boldsymbol{Msmpl}\,f,m} - \boldsymbol{p}_{\boldsymbol{Mosim}\,f,m}\|_1 \tag{6}$$

$$\mathcal{L}_{pose} = \boldsymbol{\lambda_{joint}}_j \|\hat{\boldsymbol{\theta}}_{\boldsymbol{f}} - \boldsymbol{\theta}_{\boldsymbol{mosh}\,\boldsymbol{f}}\|_1 \tag{7}$$

$$\mathcal{L} = \mathcal{L}_{joint} + \mathcal{L}_{marker} + \mathcal{L}_{pose} \tag{8}$$

where $\mathcal{L}_{motion}$ is the total loss. $\mathcal{L}_{joint}$, $\mathcal{L}_{marker}$ and $\mathcal{L}_{pose}$ are joint loss, marker loss and pose loss respectively. $\boldsymbol{\lambda_{joint}}_j$ and $\boldsymbol{\lambda_{marker}}_m$ are the weight for each joint and marker respectively. N_j is the number of OpenSim joint. $\boldsymbol{p}_{\boldsymbol{Josim}\,f,j}$ is the position of each OpenSim joint obtained via OpenSim. $\hat{\boldsymbol{p}}_{\boldsymbol{Jsmpl}\,f,j}$ is the position of the joint keypoint calculated following equation 5. $\mathcal{J}$ is a matrix that represents our joint regressor. $\hat{\boldsymbol{V}}_f \in \mathbb{R}^{N_v \times 3}$ is the positions of fitted SMPL-X vertices at current frame. N_m is the number of OpenSim marker. $\boldsymbol{p}_{\boldsymbol{Msmpl}\,f,j}$ is the virtual SMPL-X marker position for current frame. $\hat{\boldsymbol{p}}_{\boldsymbol{Mosim}\,f,j}$ is the OpenSim marker position for current frame. $\hat{\boldsymbol{\theta}}_{\boldsymbol{f}}$ is the SMPL-X pose parameter for our fitted mesh at current frame. $\boldsymbol{\theta}_{\boldsymbol{mosh}\,\boldsymbol{f}}$ is the initial guess of the pose parameter obtained via MoSh++. MoSh++ can provide a natural pose that is close to the optimal result. The purpose of the pose loss is to prevent twisting and unrealistic pose. For each frame, we optimized the pose parameter θ of SMPL-X model by minimizing the joint loss (4 and marker loss (6). Our static shape optimization and frame by frame pose optimization are based on the same loss function. However, only the SMPL-X pose parameter θ are optimized in the frame by frame pose optimization step.

2.1.7 Rendering setup

Scene settings. To augment appearances, we used four types of upper body clothing, ranging from vests to long-sleeved shirts, and four types of lower body clothing, from shorts to trousers.

Each type of the clothing is randomly combined with 5 different textures in each trial. We used multiple area light sources evenly distributed on the ceiling.

Cameras settings We employed two static cameras for video rendering that were positioned at a height of 1.1 ± 0.1 meters. One camera captured the frontal view, while the other one captured the sagittal view. To enhance diversity, the positions of the cameras are randomly perturbed within a small range. Both cameras had a fixed focal length of 33 mm. The sensor fit of the cameras was set to horizontal with the sensor width set to 36 mm.

Rendering The videos were rendered using the BLENDER_EEVEE [32] engine in Blender 3.5. The video resolution was set to 1080 by 720 and collected at a frame rate of 100 fps. Motion blur effects are disabled. The videos are encoded in the H264 format with a medium-quality configuration.

2.2 Pipeline validation

We aim to validate our pipeline from two aspects: whether it can generate realistic appearances for different subjects and actions, and whether the synthetic data accurately reflects the realistic skeletal-skin relationship. To evaluate the performance of our pipeline quantitatively, we introduced the following two categories of accuracy metrics:

Mesh to mesh error: We introduced **mesh-to-mesh RMS error** and **distance to mesh** to evaluate the body shape reconstruction ability of our pipeline. The ground truth mesh consists of vertices V_{GT} and faces F_{GT}. The fitted SMPL-X mesh consists of vertices V_{fit} and faces F_{fit}. We calculated the summation of the mean distance of selected vertices on SMPL-X mesh to ground truth 3D scan and the mean distance of all vertices on the ground truth mesh to SMPL-X mesh. We used triangle mesh to distance function [33] to calculate the distance. Figure 5 is a flowchart of this error. The mesh-to-mesh RMS error is calculated following:

$$\mathcal{E}_{meshRMS} = \sqrt{\frac{\sum_{i=0}^{N_{vsmpl}-1} \boldsymbol{d_{smpl}}_i^2}{N_{vsmpl}}} + \sqrt{\frac{\sum_{i=0}^{N_{vGT}-1} \boldsymbol{d_{GT}}_i^2}{N_{vGT}}} \tag{9}$$

where N_{vsmpl} and N_{vscan} are the number of vertices on the synthetic mesh and ground mesh respectively. $\boldsymbol{d_{smpl}}$ is the distance from a vertex on the synthetic mesh to the ground mesh. $\boldsymbol{d_{GT}}$ is the distance from a vertex on the ground mesh to the synthetic mesh. This error are calculated on one frame only. For a series of frames, we calculate the error for each frame and then take the average.

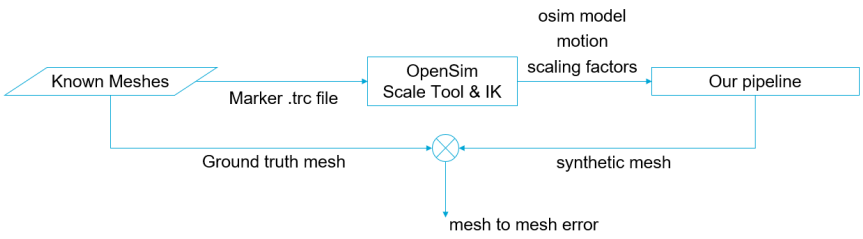


Figure 5: A flowchart of mesh to mesh error

Marker error: In the field of biomechanics, researchers utilize marker-based methods for motion capture, and assess the accuracy of the collected motion data by analyzing marker errors. Similarly, we aim to use the marker error the assess whether our generated synthetic mesh

accurately reproduce the movement of the skeletal model from a biomechanical perspective. We introduced **mean marker spacial error** to evaluate the motion following quality of the synthetic meshes generated by our pipeline. We selected several vertices on SMPL-X mesh to represent OpenSim virtual markers. These virtual markers correspond to bony landmarks crucial for estimating bone location and orientation. This error evaluates how well our fitted mesh can track the input OpenSim motions. A flowchart of this error is shown in figure 6. This error is calculated by the following formula:

$$\mathcal{E}_{marker3D} = \frac{1}{N_m N_f} \sum_{f=0}^{N_f-1} \sum_{j=0}^{N_j-1} \|\hat{\boldsymbol{p}}_{\boldsymbol{M}\,m,f} - \boldsymbol{p}_{\boldsymbol{M}\,m,f}\|_2 \tag{10}$$

, where N_m and N_f are number of markers and number of frames respectively. $\hat{p}_M$ is the position of a virtual marker on fitted SMPL-X mesh, and $\hat{p}_M$ is the position of a virtual marker on opensim model.

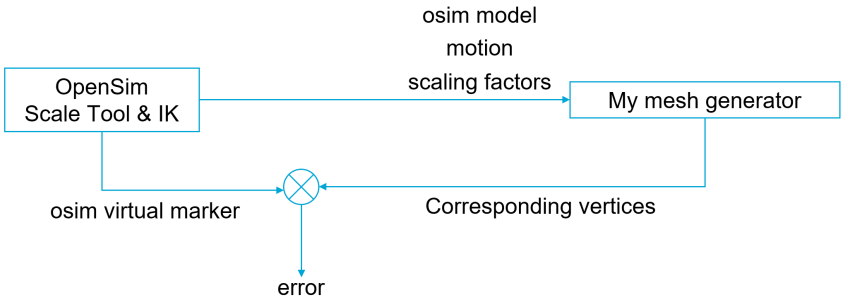


Figure 6: Mean marker spacial error

To assess the motion reconstruction ability of our pipeline in the output synthetic videos, we introduced the **mean marker pixel error** (See Figure 7). In our synthetic video generaion pipeline,the camera's intrinsic and extrinsic matrices are known. We projected OpenSim virtual markers and SMPL-X virtual markers into the image pixel coordinate system. The marker mean pixel error is calculated by the following formula:

$$\mathcal{E}_{marker2D} = \frac{1}{N_m N_f} \sum_{f=0}^{N_f-1} \sum_{j=0}^{N_j-1} \|\hat{\boldsymbol{x}}_{\boldsymbol{M}\,m,f} - \boldsymbol{x}_{\boldsymbol{M}\,m,f}\|_2 \tag{11}$$

, where $\hat{x}_M$ and x_M represent opensim virtual marker position and SMPL-X vritual marker position in image coordinate system.

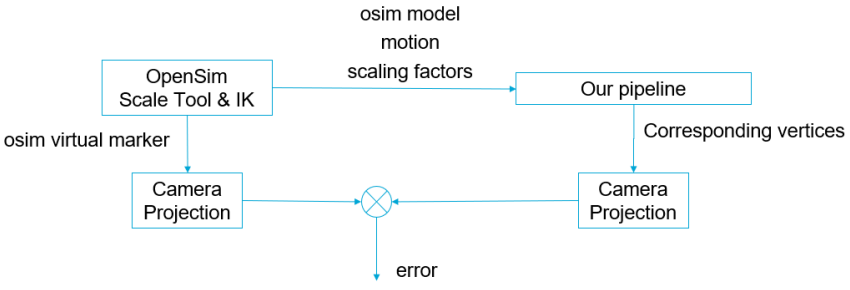


Figure 7: Mean marker pixel error

2.2.1 Datasets

To include variety of motions, we derived OpenSim models and motions from AMASS dataset [28]. We defined virtual markers on AMASS mesh vertices to obtain marker trajectories. The joint angle sequences were computed from the marker trajectories by scaling and performing inverse kinematics in OpenSim. 56 subjects in AMASS BMLmovi subset were selected. Each subject has around 19 motions. The motions include running, jogging, jumping, sideways, scratching head, throwing and catching, hand clapping, walking, checking watch, sitting down, hand waving, crossing arms, stretching, kicking, phone talking, taking photos, pointing, vertical jumping, crawling, crossing legs while sitting, and freestyle. These motions are used as synthetic dataset for neural network training. This dataset is named as OpenSim Driven Animated Human (ODAH). We validated our pipeline by assessing the quality of the meshes in ODAH.

The AMASS meshes are synthetic, and the motion capture markers and mesh obtained from them may not necessarily accurately reflect real-world situations. To assess our synthetic video ganeration pipeline, we added 2 real-world subjects for validation: subject 1 with a static pose 3D scan and corresponding motion capture marker data, and subject 2 with a treadmill running motion. Subject 1 is used to assess mesh to mesh error, and subject 2 is used to evaluate 3D and 2D marker error.

3 Results

3.1 Visual Evaluation

The ODAH dataset has diverse body shape, motions, appearances, etc. Our pipeline generated realistic appearances for skeletal model with various body sizes. The deformation of the mesh is reasonable when when performing different motions. And, the relative positional relationship is biomechnically accurate visually.

3.2 Quantitative Evaluation

We measured the mesh-to-mesh RMS error on several subjects selected from the ODAH dataset. The subjects and motions in ODAH dataset is derived from meshes in AMASS dataset. In this case, the original AMASS meshes are the ground truth. We measured the mesh-to-mesh RMS error on 7 subjects (index in ODAH: 1, 2, 3, 4, 6, 7, 19). Each subject has 19 motion trails. We calculated the mesh-to-mesh RMS error for each motions, and then calculated the mean error and standard error across all motions for each subject individually. Table 9 lists our result. The mean mesh-to-mesh error for each subject ranges from 1.15 to 1.48cm. The result is comparable to the accuracy of other methods such as MoSh [34]. This result indicates that our pipeline is capable of generating reliable meshes for various body shapes and motions.

AMASS is a synthetic dataset using SMPL-X human representation. The AMASS meshes have realistic appearance, but they cannot represent a real-world individual. To validate our pipeline on real-world human, we compared our synthetic mesh with the 3D scan of the subject in a static pose. We measured the mesh-to-mesh RMS error and max distance to ground truth mesh in fullbody and several interested regions. Table 1 show the results. Figure 11 shows the body segmentation and distance from our synthetic mesh to 3D scan. The average mesh to mesh error of the fullbody is 0.95cm. The shoulder area exhibits both the largest mesh-to-mesh RMS error and the max distance to the ground truth mesh. We overlaid the 3D scan with the OpenSim skeleton and noticed that the skeleton protrudes several centimeters above the

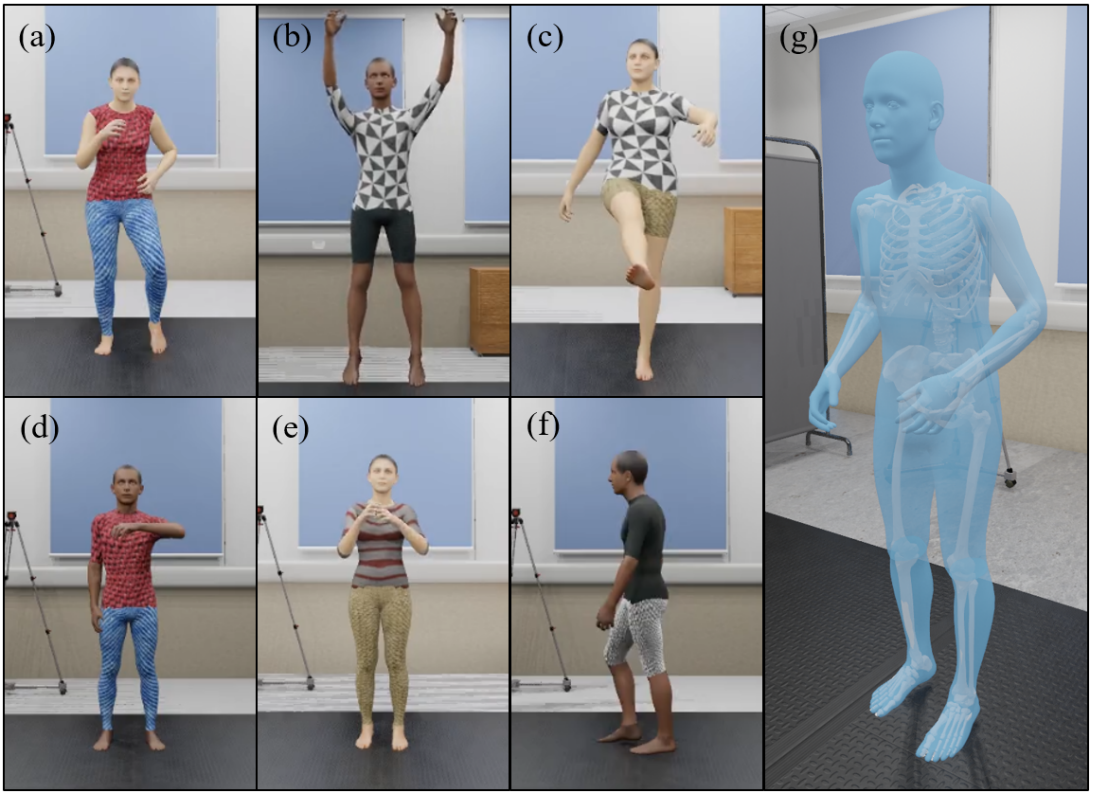


Figure 8: Examples of ODAH dataset. As an example, here are several different characters in various poses: (a) Jogging. (b) Vertical jumping. (c) Kicking. (d) Checking watch. (e) Clapping. (f) Walking. In (g), the SMPL-X mesh is fitted well against the Opensim skeletal model visually.

skin (See Figure 10). Our fitted mesh is generated to fit the skeletal model, leading to larger error in shoulder area. The skeletal model is scaled and posed using OpenSim Scale Tool and Inverse Kinematic Tool to "best match" experimental marker data. Also, in our synthetic data generation pipeline, we selected a set of vertices on our synthetic mesh as virtual markers to assist fitting. In SMPL-X, vertices are discrete, and we can only choose the nearest vertex as the virtual marker. The marker layout in SMPL-X mesh cannot perfectly reflect the actual marker placement in real world. The marker placement error in OpenSim skeletal model and SMPL-X mesh model could result in a misalignment between the 3D scan and the skeletal model. The asymmetry noticed in the heatmap can also be explained by the asymmetric marker placement in OpenSim skeletal model and SMPL-X mesh.

In figure 11, we also observed a relatively larger error in waist and thigh areas. The body shape of these areas are not reconstructed very well compared with 3D scan. In real human body, these areas contains more soft tissues, such as fat and muscles, resulting in a lack of bony landmarks that can constrain body shape reconstruction. In the case of this example subject, our synthetic mesh has larger volume in these area, which can be the cause of larger distance to ground truth in these area. a larger distance to ground truth is observed in the area of right knee. This indicates that there may be an inaccurate alignment between the synthetic mesh and the skeletal model at the right knee joint, suggesting that we need a more accurate joint regressor to represent the mapping between the OpenSim skeletal model and the SMPL-X mesh.

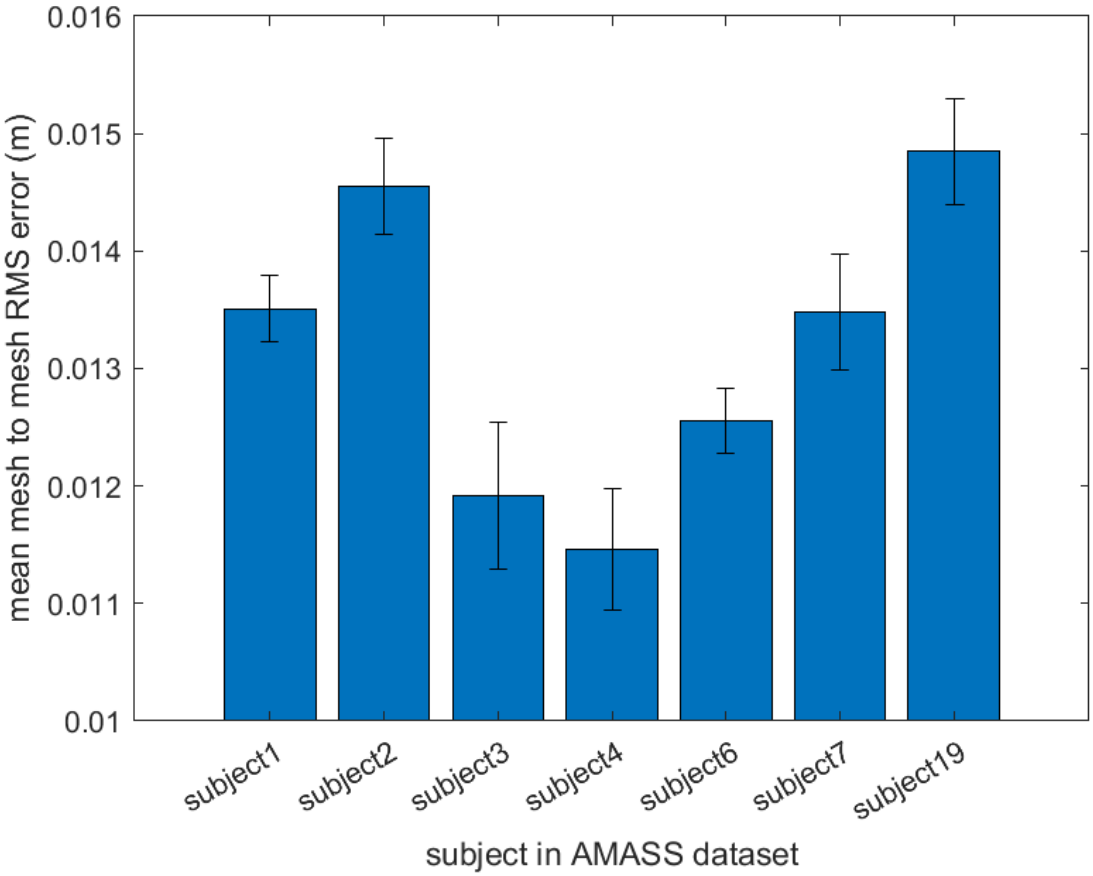


Figure 9: AMASS full body mesh to mesh error. For different subjects and motion, the mesh to mesh RMS error is acceptable. We also plot the error bar for each subject. The results indicate that the our pipeline can generate suitable mesh for different body shape and motions.

We calculated the mean marker spacial error and mean marker pixel error on a running motion following equations 11 and 10 respectively. The subject-specific model and the motion are recorded and validated by Rajagopal et al. [27]. The results are shown in table 2. The mean marker spacial error is 1.38cm. The mean marker pixel error is 2.32 pixel. The resolution of the video is 1080*720, and the mean marker pixel error in the image coordinate represent roughly an error of 1.0cm in 3D space. The result meets the biomechanical accuracy requirement for motion capture [35]. The result of mean marker pixel error in two views are shown in figure 13.

4 Discussion, Limitation and Future Work

Based on our results, we showed that our synthetic data generation pipeline can generate realistic human appearance, and the skeleton-mesh registration is also realistic. The synthetic dataset was used as the training set of a human pose estimation neural network [36], and the network performs well in both pose estimation on real world dataset and synthetic dataset. It further proves the practical value of our pipeline.

Our pipeline allows researchers to build synthetic video dataset from opensim models and

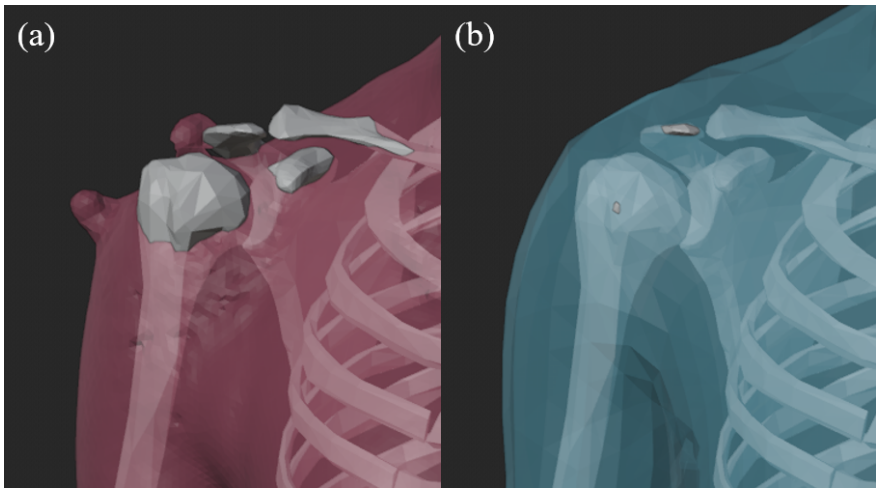


Figure 10: **Errors in shoulder area** (a) The skeletal model protrudes the 3D scan of the subject. (b) The skeletal is covered by the synthetic mesh. It explains the why the distance to ground truth is larger in shoulder area.

Region	mesh to mesh error	max distance
fullbody	0.95cm	3.81cm
wrists	1.10cm	2.72cm
forearms	0.94cm	2.72cm
elbows	0.77cm	1.49cm
upper arms	1.04cm	3.81cm
shoulders	1.34cm	3.81cm
torso	0.91cm	3.13cm
pelvis	0.91cm	2.57cm
hips	0.94cm	2.19cm
thighs	0.88cm	2.46cm
knees	0.86cm	2.07cm
shanks	0.79cm	1.3cm
ankles	0.92cm	1.73cm
feet	0.95cm	2.95cm

Table 1: mesh reconstruction error

motions. In OpenSim models, the connections of various body parts and the angle of each joint are defined in accordance with prior biomechanical knowledge [27]. The knowledge is based on decades of research, and is validated by comparing simulation results with experimental measures [21]. The model shows higher biomechanical credibility compared to other models. Therefore, our ground truth annotations are considered more trustworthy from a biomechanical perspective.

Our pipeline also allows researchers to create a synthetic video database using existing motion capture data. Existing experimental marker data can be transformed into an OpenSim models and motions using the OpenSim Inverse Kinematics and scale tool. Synthetic human motion video can be generated by inputting OpenSim models and motions into our pipeline. It is very convenient to modify background scene, lighting condition, and camera parameters within the pipeline to make the video data more diverse. People can rapidly expand existing OpenSim datasets. The data generation process is entirely on computer and requires minimal human operation. It is both time and money friendly, comparing to creating a video dataset from scratch. They can then train task-specific neural network based on their existing data.

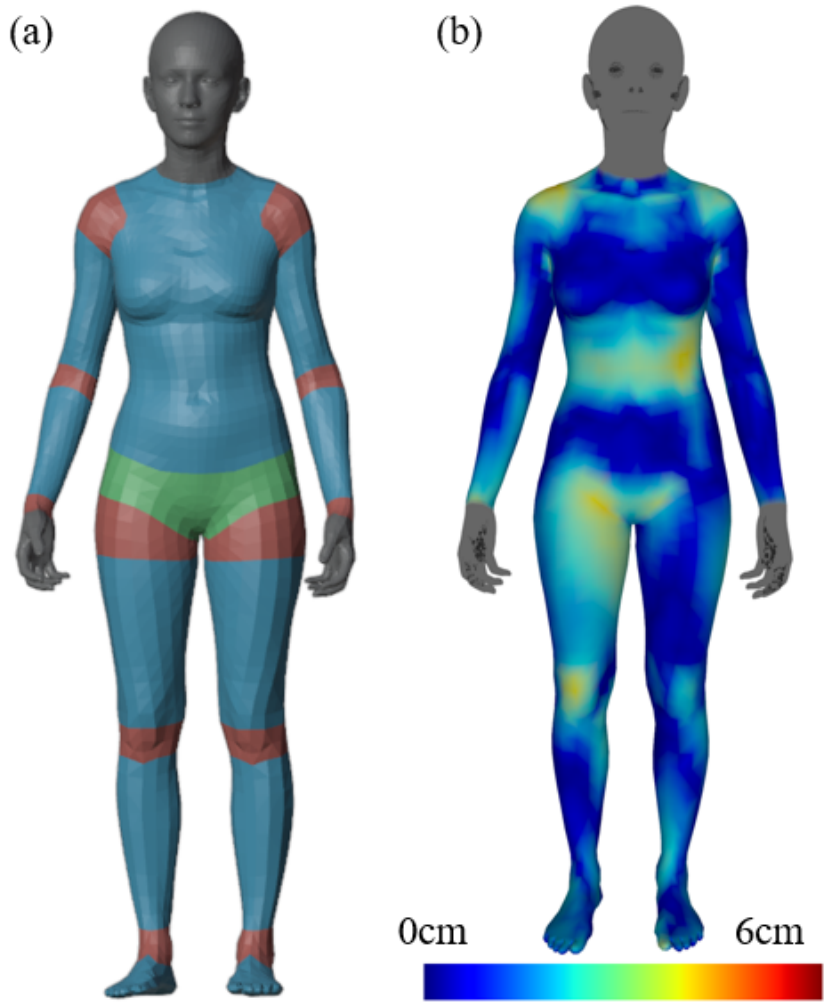


Figure 11: (a) Body segmentation. We manually segmented the human mehs into 13 areas (wrists, forearms, elbows, upper arms, shoulders, torso, pelvis, hips, thighs, knees, shanks, ankles, and feet) Note that hands and head are not included due to a lack of accurate model. (b) A heat map of the distance from each vertex on SMPL-X mesh to the 3D scan.

Our pipeline has some limitations. The joint regressor used in our pipeline is trained based on our manually fitted templates. This step lacks repeatability. The credibility of these templates needs to be verified. Using fluoroscopy [37], X-ray [38], and MRI can help people find out the accurate mapping between human skin and skeleton. Then we can get an accurate joint regressor. Another limition of our pipeline, is that the SMPL-X model is still driven by its internal armature, although we rigged it against the given OpenSim model. Vertices on the mesh still rotate around SMPL-X joint centers instead of joints centers in the OpenSim skeletal model. The difference in joint centers introduces errors in our synthetic mesh, especially when a joint is located at the distal end of the body and has large joint angles. This can lead to unrealistic relative position between the mesh and the OpenSim skeleton, such as bones protruding through the surface of the mesh. A solution to this issues is to create a human mesh model that has an internal biomechanical skeleton, which is our ultimate goal. It requires a human body 3D scan dataset with a diverse range of poses for body shapes, along with accurate mapping between the human skin and internal skeleton. There are several existing human 3D scan datasets [29,39,40], but the mapping between skin and skeleton remains unclear. Finding this relationship will be the key element to achieving our ultimate goal.

mean marker spacial error	mean marker pixel error
1.38cm	2.32 pixel

Table 2: The mean marker spatial error and the mean marker pixel error for a subject performing a running motion.

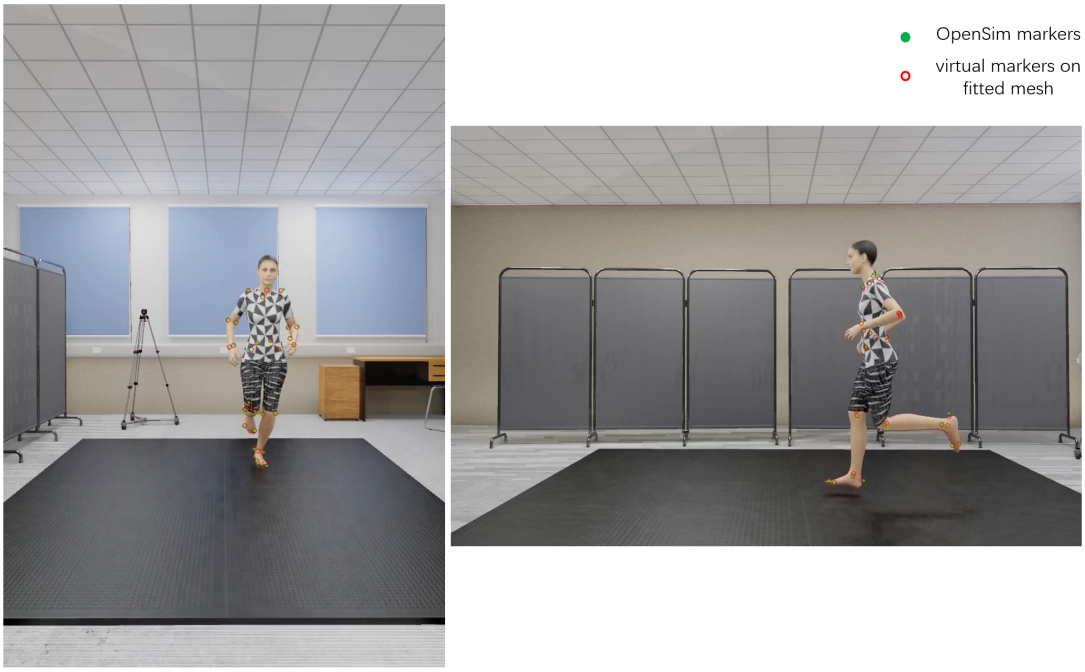


Figure 12: The mean marker pixel error in front and side views. Virtual markers on the OpenSim model are displayed in green. Virtual markers on the fitted SMPL-X mesh are displayed in red. Visually, the 2D marker error is very minimal in the output video.

Our synthetic dataset and pipeline has been proven effective for neural network training. However, it still remains unclear whether training solely on synthetic datasets can achieve results comparable to training solely on real world datasets. We aim to create a real world dataset and a synthetic dataset based on the same groups of subjects and motions, and train two identical neural networks. Comparing their training results can further validate the concept of utilizing synthetic datasets for neural network training.

Another question worthy of consideration is what elements in synthetic dataset contribute more to neural network training? Is it realistic motions, texture, clothing, lighting or others factors? We can create various synthetic datasets with different setups to find out the answer to this question.

Figure 13: An example of bad fitting. The calcaneus may protruding through the surface of the mesh. The mesh and the OpenSim skeletal model are driven by different models. We fitted the shape at the static pose, but we cannot be sure that the mesh and the skeleton align perfectly for all poses.

References

[1] E. Van der Kruk and M. M. Reijne, "Accuracy of human motion capture systems for sport applications; state-of-the-art review," *European journal of sport science*, vol. 18, no. 6, pp. 806–819, 2018.

[2] B. Scott, M. Seyres, F. Philp, E. K. Chadwick, and D. Blana, "Healthcare applications of single camera markerless motion capture: a scoping review," *PeerJ*, vol. 10, p. e13517, 2022.

[3] S. L. Colyer, M. Evans, D. P. Cosker, and A. I. Salo, "A review of the evolution of vision-based motion analysis and the integration of advanced computer vision methods towards developing a markerless system," *Sports medicine-open*, vol. 4, no. 1, pp. 1–15, 2018.

[4] L. Wade, L. Needham, P. McGuigan, and J. Bilzon, "Applications and limitations of current markerless motion capture methods for clinical gait biomechanics," *PeerJ*, vol. 10, p. e12995, 2022.

[5] Y. Guo, F. Deligianni, X. Gu, and G.-Z. Yang, "3-d canonical pose estimation and abnormal gait recognition with a single rgb-d camera," *IEEE Robotics and Automation letters*, vol. 4, no. 4, pp. 3617–3624, 2019.

[6] S. K. Natarajan, X. Wang, M. Spranger, and A. Gräser, "Reha@ home-a vision based markerless gait analysis system for rehabilitation at home," in *2017 13th IASTED international conference on biomedical engineering (BioMed)*. IEEE, 2017, pp. 32–41.

[7] R. Šajina and M. Ivašić-Kos, "3d pose estimation and tracking in handball actions using a monocular camera," *Journal of Imaging*, vol. 8, no. 11, p. 308, 2022.

[8] M. Ostrek, H. Rhodin, P. Fua, E. Müller, and J. Spörri, "Are existing monocular computer vision-based 3d motion capture approaches ready for deployment? a methodological study on the example of alpine skiing," *Sensors*, vol. 19, no. 19, p. 4323, 2019.

[9] B. K. Lahkar, A. Muller, R. Dumas, L. Reveret, and T. Robert, "Accuracy of a markerless motion capture system in estimating upper extremity kinematics during boxing," *Frontiers in Sports and Active Living*, vol. 4, p. 939980, 2022.

[10] S. Ghorbani, K. Mahdaviani, A. Thaler, K. Kording, D. J. Cook, G. Blohm, and N. F. Troje, "Movi: A large multi-purpose human motion and video dataset," *Plos one*, vol. 16, no. 6, p. e0253157, 2021.

[11] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2013.

[12] "Carnegie mellon university - cmu graphics lab - motion capture library." [Online]. Available: http://mocap.cs.cmu.edu/

[13] N. M. Fiorentino, P. R. Atkins, M. J. Kutschke, J. M. Goebel, K. B. Foreman, and A. E. Anderson, "Soft tissue artifact causes significant errors in the calculation of joint angles and range of motion at the hip," *Gait & posture*, vol. 55, pp. 184–190, 2017.

[14] S. E. Kessler, M. J. Rainbow, G. A. Lichtwark, A. G. Cresswell, S. E. D'Andrea, N. Konow, and L. A. Kelly, "A direct comparison of biplanar videoradiography and optical motion capture for foot and ankle kinematics," *Frontiers in bioengineering and biotechnology*, vol. 7, p. 199, 2019.

[15] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, "Learning from synthetic humans," in *CVPR*, 2017.

[16] H. Y. F. Tung, H. W. Tung, E. Yumer, and K. Fragkiadaki, "Self-supervised learning of motion capture," in *31st Annual Conference on Neural Information Processing Systems (NIPS)*, ser. Advances in Neural Information Processing Systems, vol. 30. Neural Information Processing Systems (Nips), 2017.

[17] M. J. Black, P. Patel, J. Tesch, and J. Yang, "BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion," in *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 8726–8737.

[18] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, 2014, pp. 3686–3693.

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[20] L. Sigal, A. O. Balan, and M. J. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International journal of computer vision*, vol. 87, no. 1-2, p. 4, 2010.

[21] A. Seth, J. L. Hicks, T. K. Uchida, A. Habib, C. L. Dembia, J. J. Dunne, C. F. Ong, M. S. DeMers, A. Rajagopal, M. Millard *et al.*, "Opensim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement," *PLoS computational biology*, vol. 14, no. 7, p. e1006223, 2018.

[22] Aug 2022. [Online]. Available: https://www.theiamarkerless.ca/

[23] S. D. Uhlrich, A. Falisse, Ł. Kidziński, J. Muccini, M. Ko, A. S. Chaudhari, J. L. Hicks, and S. L. Delp, "Opencap: 3d human movement dynamics from smartphone videos," *bioRxiv*, pp. 2022–07, 2022.

[24] M. Bittner, W.-T. Yang, X. Zhang, A. Seth, J. van Gemert, and F. C. van der Helm, "Towards single camera human 3d-kinematics," *Sensors*, vol. 23, no. 1, p. 341, 2022.

[25] R. Schleicher, M. Nitschke, J. Martschinke, M. Stamminger, B. M. Eskofier, J. Klucken, and A. D. Koelewijn, "Bash: Biomechanical animated skinned human for visualization of kinematics and muscle activity." in *VISIGRAPP (1: GRAPP)*, 2021, pp. 25–36.

[26] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3D hands, face, and body from a single image," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 975–10 985.

[27] A. Rajagopal, C. L. Dembia, M. S. DeMers, D. D. Delp, J. L. Hicks, and S. L. Delp, "Full-body musculoskeletal model for muscle-driven simulation of human gait," *IEEE transactions on biomedical engineering*, vol. 63, no. 10, pp. 2068–2079, 2016.

[28] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "AMASS: Archive of motion capture as surface shapes," in *International Conference on Computer Vision*, Oct. 2019, pp. 5442–5451.

[29] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] A. A. A. Osman, T. Bolkart, and M. J. Black, "STAR: A sparse trained articulated human body regressor," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 598–613. [Online]. Available: https://star.is.tue.mpg.de

[32] [Online]. Available: https://docs.blender.org/manual/en/latest/render/eevee/introduction.html

[33] A. Yu, "sdf," https://github.com/sxyu/sdf, 2020.

[34] M. Loper, N. Mahmood, and M. J. Black, "Mosh: motion and shape capture from sparse markers." *ACM Trans. Graph.*, vol. 33, no. 6, pp. 220–1, 2014.

[35] T. K. Uchida and A. Seth, "Conclusion or illusion: Quantifying uncertainty in inverse analyses from marker-based motion capture due to errors in marker registration and model scaling," *Frontiers in Bioengineering and Biotechnology*, vol. 10, p. 874725, 2022.

[36] Z.-Y. Lin, "3d kinematics estimation with biomechanics model," Master's thesis, Delft University of Technology, 2023.

[37] R. Stagni, S. Fantozzi, A. Cappello, and A. Leardini, "Quantification of soft tissue artefact in motion analysis by combining 3d fluoroscopy and stereophotogrammetry: a study on two subjects," *Clinical biomechanics*, vol. 20, no. 3, pp. 320–329, 2005.

[38] M. Keller, S. Zuffi, M. J. Black, and S. Pujades, "Osso: Obtaining skeletal shape from outside," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 492–20 501.

[39] F. Bogo, J. Romero, M. Loper, and M. J. Black, "Faust: Dataset and evaluation for 3d mesh registration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3794–3801.

[40] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel, "A statistical model of human pose and body shape," in *Computer graphics forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 337–346.

Appendix

Code

The code will be publicly availibel via: `https://github.com/blyu413/Synthetic-human-motion-video-generation`

Rendering

Figure 14 shows our scene and camera setup from top view.

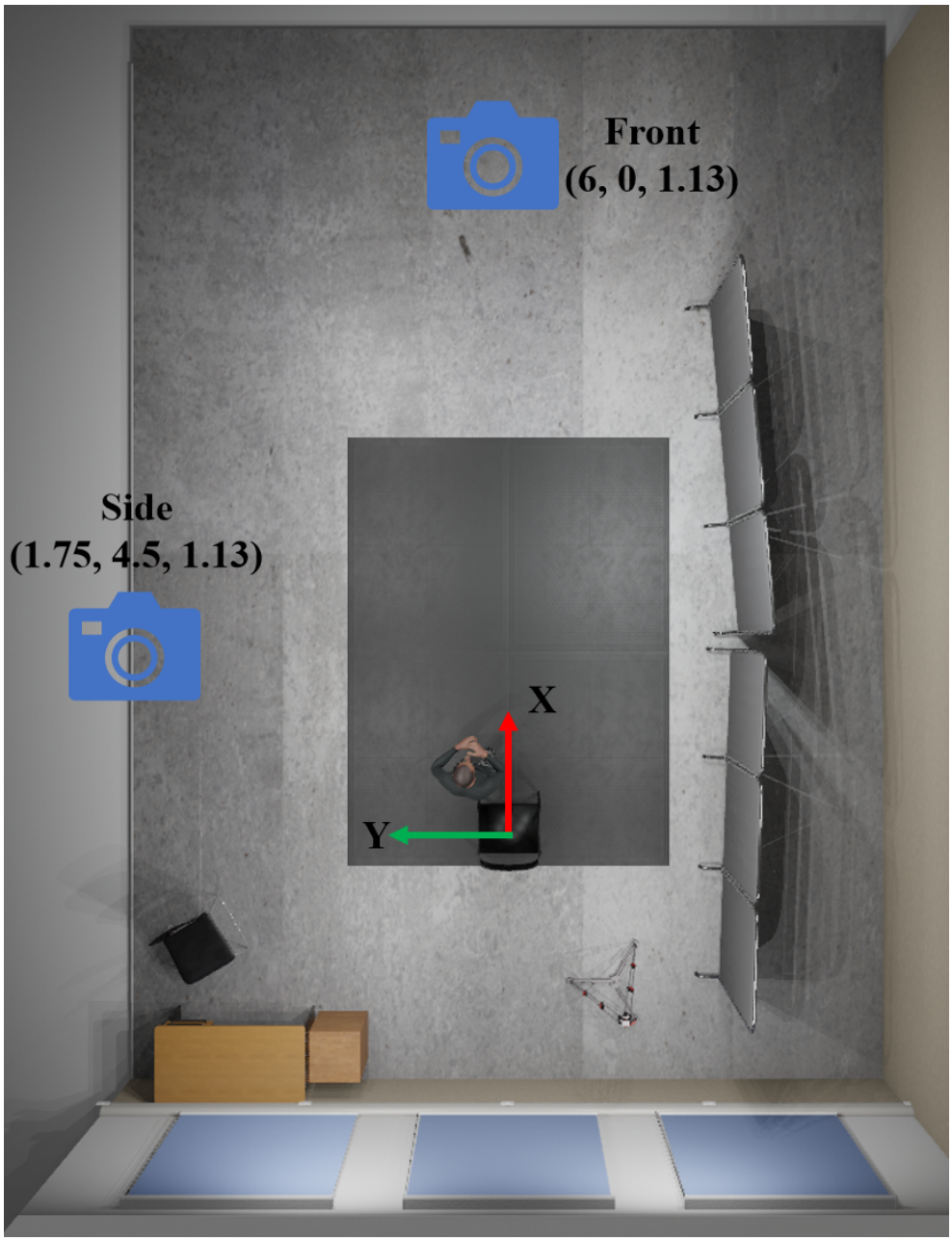


Figure 14: Scene and camera setup from top view.

Mesh Error

Table 3 lists the mesh to mesh RMS error for all subjects and motions.

Marker Error

The mean and max spacial marker error for each marker during running is listed in table 4. And figure 15 the marker placement

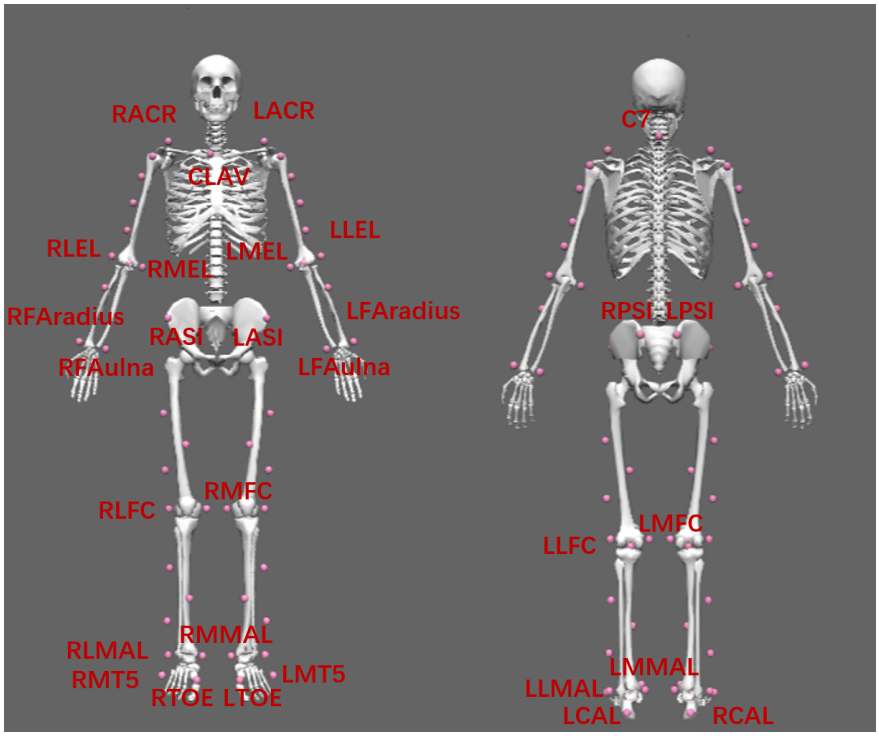


Figure 15: Marker placement in running subject.

Motion name	**Subject 1**	**Subject 2**	**Subject 3**	**Subject 4**	**Subject 6**	**Subject 7**	**Subject 19**
checking watch	1.36	1.38	0.99	1.01	1.12	1.50	1.33
crossarms	1.22	1.30	0.89	0.99	1.24	1.32	1.46
dancing	1.26	1.48	1.36	1.19	1.32	0.00	1.55
hand clapping	1.21	1.25	0.89	0.91	1.08	1.44	1.29
hand waving	1.28	1.41	0.99	1.01	1.24	1.39	1.34
jogging	1.28	1.37	1.16	1.03	1.25	1.42	1.33
jumping	1.49	1.52	1.35	1.36	1.34	1.41	1.45
kicking	1.26	1.37	1.14	0.99	1.28	1.37	1.69
phone talking	1.28	1.40	1.13	1.07	1.05	1.45	1.27
pointing	1.37	1.33	1.07	1.15	1.22	1.34	1.43
running in spot	1.19	1.37	1.00	0.97	1.25	1.62	1.39
scratching head	1.39	1.46	1.11	1.13	1.27	1.35	1.32
sideways	1.33	1.38	1.02	1.06	1.27	1.53	1.66
sitting down	1.24	1.45	1.05	1.06	1.15	1.32	1.51
stretching	1.56	1.65	1.63	1.31	1.35	1.27	1.47
taking photo	1.28	1.41	1.31	0.98	1.18	1.30	1.27
throw/catch	1.48	1.49	1.22	1.22	1.32	1.36	1.29
vertical jumping	1.30	1.38	1.00	1.14	1.26	1.45	1.65
walking	1.34	1.39	1.06	1.10	1.18	1.41	1.55

Table 3: The mesh to mesh RMS error for all subjects and motions. The error is expressed in centimeter

Marker Name	**mean spacial error (cm)**	**max spacial error (cm)**
RACR	0.013022	0.018833
LACR	0.027348	0.031274
C7	0.025297	0.029584
CLAV	0.009441	0.014297
RLEL	0.012412	0.014601
RMEL	0.00121	0.003734
RFAradius	0.007793	0.013361
RFAulna	0.012357	0.014803
LLEL	0.007322	0.01092
LMEL	0.011302	0.017017
LFAradius	0.018406	0.023181
LFAulna	0.005241	0.007867
RASI	0.01512	0.026761
LASI	0.021646	0.031225
RPSI	0.008419	0.012703
LPSI	0.019457	0.029106
RLFC	0.016339	0.023291
RMFC	0.011515	0.018374
RLMAL	0.016734	0.020439
RMMAL	0.019052	0.022482
RCAL	0.01653	0.020117
RTOE	0.00198	0.004797
RMT5	0.013954	0.018352
LLFC	0.004438	0.009054
LMFC	0.013488	0.029218
LLMAL	0.014397	0.020014
LMMAL	0.015748	0.022259
LCAL	0.023116	0.026178
LTOE	0.001832	0.007516
LMT5	0.027817	0.029886

Table 4: Mean and max marker spacial error for each marker.