Modeling Dynamic Clothing for Data-Driven Photorealistic Avatars

Donglai Xiang

CMU-RI-TR-23-75

Sept 2023

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Thesis Committee:
Jessica K. Hodgins (Chair)
Fernando De la Torre
Matthew P. O'Toole
Chenglei Wu (Google)
Niloy J. Mitra (University College London, Adobe)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics.

Abstract

In this thesis, we aim to build photorealistic animatable avatars of humans wearing complex clothing in a data-driven manner. Such avatars will be a critical technology to enable future applications such as immersive telepresence in Virtual Reality (VR) and Augmented Reality (AR). Existing full-body avatars that jointly model geometry and view-dependent texture using Variational Autoencoders (VAEs) can be efficiently animated and rendered from arbitrary viewpoints, but they only work with tight garments such as T-shirts and pants. Loose-fitting clothing, however, does not closely follow the body motion and has a much larger deformation space. Most styles of clothing pose a significant challenge in various aspects of the existing frameworks for avatar modeling, including tracking, animation and rendering.

This thesis builds a systematic solution for modeling dynamic clothing for data-driven photorealistic avatars. As opposed to the single-layer representation of existing full-body avatars, where the clothing is treated as a residual deformation on top of the human body, we utilize a separate representation of clothing that allows modeling at a finer granularity. We address the challenge by unifying three components of avatar modeling: a model-based statistical prior from pre-captured data, a physics-based prior from simulation, and measurement from sparse sensor input.

In the first work, we introduce a separate two-layer representation for body and clothing in animatable full-body avatars. This separation allows us to disentangle the dynamics between the pose-driven body part and temporally-dependent clothing part, which leads to much higher overall quality results. In addition, this formulation also enables photorealistic editing of clothing color and texture in a temporally coherent manner. In the second work, we further combine physics-based cloth simulation with the photorealistic avatars, which can generate rich and natural dynamics even for loose clothing, such as a skirt or a dress. We develop a physics-inspired neural rendering model to bridge the generalization gap between the training data from registered captured clothing and the test data from simulated clothing. This approach further allows the animation of a captured garment on the body of a novel avatar without the need for the person to wear the clothing in reality, thus opening up the possibility of photorealistic virtual try-on. After that, we go beyond pose-driven animation of clothing, and incorporate denser sensor input to achieve more faithful telepresence including clothing. We first investigate a simpler setting, where we build a linear clothing model to capture clothing deformation in a temporally coherent manner from a monocular RGB video input. Finally, we develop a two-stage method to faithfully drive photorealistic avatars with loose clothing using several RGB-D cameras. We first coarsely track the clothing surface online to produce texel-aligned unwrapped image and geometric features in the UV space. This sensor-based conditioning input is then fed to the avatars to reproduce the clothing appearance from an arbitrary viewpoint. We demonstrate that such avatars can be driven not only in the original capture environment, but also a novel environment with different illumination and background with several RGB-D cameras.

Acknowledgement

I still remember the mixture of excitement and nervousness for my first meeting with Jessica Hodgins in the Bakery Square office with glass walls in the summer of 2019. I started as an intern at FAIR and ended up being advised by her for four years at CMU. There are three things that I particularly appreciate: the opportunities that she helped to create for students such as the AI mentorship program at Meta where I conducted most of my thesis research, the encouragement by her to pursue wild ideas in areas that I had no previous experiences in, and the top priority she always puts on the well-being of students. I feel very fortunate to have had a relatively smooth journey during my Ph.D. career, which would not have been possible without her efforts and support.

I would also like to thank my other thesis committee members, Fernando De la Torre, Matthew O'Toole, Chenglei Wu and Niloy Mitra, for their valuable time and detailed, constructive feedback on my thesis. Their expertise in different areas of graphics and vision will be a goal for me to pursue for quite a long time in my career. I am particularly grateful to the help from Chenglei, who was effectively my mentor at Meta. He co-authored every work in this thesis and offered insights that were extremely helpful in setting the overall direction of this thesis at critical times.

During the past six years of graduate studies, I have been extremely fortunate to work with a number of great researchers at CMU and Meta. I was first introduced to the area of modeling digital humans, or even serious research as a whole, as a master student in Yaser Sheikh's lab. I was grateful to receive hands-on guidance from Hanbyul Joo, who carefully set the stage for me to get on track in my first project, and patiently answered all my newbie questions. I also received a lot of kind advice from Aayush Bansal and Minh Vo, and had fun working with Ginés Hidalgo and Yaadhav Raaj on projects related to Panoptic Studio as well as OpenPose. During my Ph.D., I was fortunate to be on the same boat with Yanzhe Yang, Emily Kim, Ziyan Wang and Jiashun Wang in Jessica's lab at CMU.

By the time I finished my master degree, the booming Pittsburgh FRL directed by Yaser had showed the first demo of Codec Avatars and attracted a great number of talents, many of whom later became my collaborators during my PhD. As an intern and visiting researcher, I benefited substantially from the insight and help of so many great researchers across different teams at Meta, including but not limited to Fabian Prada, Timur Bagautdinov, Weipeng Xu, Tuur Stuyck, Takaaki Shiratori, Shunsuke Saito, Zhe Cao, Kaiwen Guo, Javier Romero, Breannan Smith, Yuan Dong and He Wen. I remember how they went above and beyond to assist me with my projects, especially in times of hardship. I also had fun with many fellow interns, notably Oshri Halimi and Jingfan Guo whom I had the opportunity to work with. There were also many valuable memories with Ziyan Wang and Zhengyi Luo; we shared so many unique difficulties working for a long term as visiting reseachers with international student status.

Beyond research, I was grateful for the friendship that made my life much more delightful. I remember spending the past lunar new years missing home together with friends such as Gengshan Yang, Sha Yi, Xianyi Cheng, Shuyan Zhou, Wenxuan Zhou, etc. I recall the parties nights with Jason Zhang, Helen Jiang, Sudeep Dasari, Shikhar Bahl and others. They are many people who I didn't get the chance to work together but I had a lot fun chatting with: Dinesh Reddy, Chaoyang Wang, Martin Li, Kangle Deng, Soyong Shin, Zhiqiu Lin, Jinkun Cao, Ruihan Gao, Jeff Tan etc. Six years is long enough for so many people to come and leave, which makes the list much longer than I could note down. I also cherish the memory of the concert performance with the All University Orchestra, the tennis games at

Schenley Park, and the late hours in the UC gym.

Lastly, I would like to thank my parents, Qian Yang and Yong Xiang, for supporting my pursuit of studies with freedom from want and for their not-so-successful attempts to understand my research. I am grateful to my grandparents and other family members for always having faith on me. I would like to appreciate my partner, Yufei Ye, for being the best ally on this journey, since many years ago and for more years to come.

Contents

Chapter 1

Introduction

Photorealistic digital humans are a key capability for enabling social telepresence, which is one of the key applications for Virtual Reality (VR) or Augmented Reality (AR). Such a technology would allow people wearing VR/AR devices to communicate and interact with friends and colleagues in an immersive way that is natural and compelling, because their partners are represented in a way that is indistinguishable from reality. If we are successful in developing this technology, it will open up a new way for people around the world to remain connected without geographic constraints.

One key question is how to create high-fidelity digital avatars that are photorealistic and resemble the appearance of individual subjects. One milestone is the development of the Codec Avatar [117], where the geometry and photorealistic appearance of human heads can be compressed into a low-dimensional latent space and then decoded for display efficiently by a Variational Autoencoder (VAE) [92] trained with captured human imagery. The absence of other body parts such as torso and hands was one of the major limitations of the first iteration of this technology. Recently, photorealistic full-body avatars [7,67,115] have been developed so that the communication signals conveyed by body and hands can also be represented. The central idea behind these avatars is to model large, skeleton-level deformation with skinning techniques to allow control through body joint angles.

We argue that this quest for completeness in photorealistic avatars should not be limited to the human body, but should include the clothing as well. Capturing and animating clothing for full-body avatars is the central topic in this thesis. Why is clothing so important? Clothing is an integral part of everyday human appearance, and visual artifacts in clothing will inevitably jeopardize the immersive experiences in VR/AR. Clothing is also an essential form of self-expression. Without the ability to model clothing, an avatar cannot establish the full identity of a subject, which is vital for replicating interaction in social telepresence.

The wide variety of clothing, however, poses a significant challenge in the modeling of geometry and appearance. The root of this challenge is the huge deformation space and rapid dynamics of clothing as it is driven by the underlying human body. Loose-fitting garments present particular challenges because they do not tightly follow the motion of the underlying body, and their deformation can go far beyond what the skeleton-level transformation can describe. Clothing does not contain universal identifiable keypoints to assist tracking of deformation, such as those commonly used for tracking the face [194,213] and body [16,242].

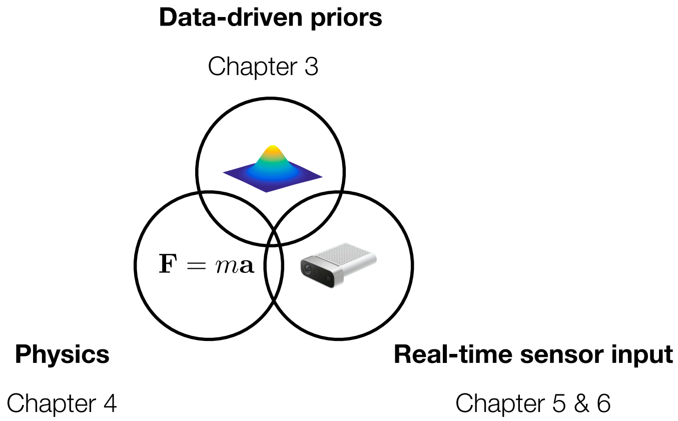

Figure 1.1: An overview of this thesis work. The three components we investigate are data-driven priors, physics and real-time sensor input.

The challenge is manifest in various aspects of the modeling of clothing, including tracking, animation, and rendering, which are all required to enable high-fidelity clothing for photorealistic avatars. The efficient learning of dynamic appearance in NN-based avatars often requires *tracking*[1] of the clothing geometry, which is inherently difficult due to its rapid motion and abundance of folds and wrinkles. Existing data-driven approaches also struggle to generate high-quality *animation* of clothing, because the mapping from body pose to the dynamics of clothing is a complex, nonlinear and history-dependent function. Furthermore, the sophisticated deformation of clothing also leads to complicated illumination and shadowing effects. Taken together with the wide variety of possible textures on the clothing, the *rendering* of photorealistic appearance of the clothing is also a challenge.

1.1 Thesis Overview

In this thesis, we aim to build photorealistic full-body avatars with dynamic clothing that are useful for social telepresence. In order to address the aforementioned challenges and to achieve the highest possible fidelity, we believe that the system should include the three components: *data-driven priors*, *physics*, and *real-time sensor input*.

Data-driven priors. Modern machine learning algorithms provide a powerful way to build statistical models from a large amount of data. We would like to explore how such data-driven priors can be applied to dynamic clothing. In the first work (Chapter 3), we develop a method to build an animatable clothed body avatar with an explicit representation of the clothing on the upper body from multi-view captured videos. We use a two-layer mesh representation to register each 3D scan separately with the body and clothing templates. In order to improve the photometric correspondence across different frames, texture alignment is then performed through inverse rendering of the clothing geometry and texture predicted by a variational autoencoder. We then train a new two-layer codec avatar with separate modeling of the upper clothing and the inner body layer. To learn the interaction

[1] Also interchangeably called "registration" throughout this thesis.

between the body dynamics and clothing states, we use a temporal convolution network to predict the clothing latent code based on a sequence of input skeletal poses. We show photorealistic animation output for three different actors, and demonstrate the advantage of our clothed-body avatars over the single-layer avatars used in previous work. We also show the benefit of an explicit clothing model that allows the clothing texture to be edited in the animation output. This work was published as Xiang, Donglai, et al. ”Modeling Clothing as a Separate Layer for an Animatable Human Avatar.” ACM Transactions on Graphics (SIGGRAPH Asia) 2021 [218].

Physics. Physics governs the motion of dynamic clothing on top of the human body, and cloth simulation can generate clothing animation with rich and natural dynamics. An interesting question is whether and how physics-based cloth simulation can serve as a complement for the learning-based approach, which faces difficulties modeling complicated dynamics of loose clothing. In the second work (Chapter 4), we introduce pose-driven avatars with explicit modeling of clothing that exhibit both photorealistic appearance learned from real-world data and realistic clothing dynamics. The key idea is to introduce a neural clothing appearance model that operates on top of explicit geometry: at training time we use high-fidelity tracking, whereas at animation time we rely on physically simulated geometry. Our core contribution is a physically-inspired appearance network, capable of generating photorealistic appearance with view-dependent and dynamic shadowing effects even for unseen body-clothing configurations. We conduct a thorough evaluation of our model and demonstrate diverse animation results on several subjects and different types of clothing. Unlike previous work on photorealistic full-body avatars, our approach can produce much richer dynamics and more realistic deformations even for many examples of loose clothing. We also demonstrate that our formulation naturally allows clothing to be used with avatars of different people while staying fully animatable, thus enabling, for the first time, photorealistic avatars with novel clothing. This work was published as Xiang, Donglai, et al. ”Dressing Avatars: Deep Photorealistic Appearance for Physically Simulated Clothing.” ACM Transactions on Graphics (SIGGRAPH Asia) 2022 [216].

Real-time sensor input. For certain applications, we may want the rendered clothing to look not only realistic, but also match its motion in the real world. The methods developed above can generate clothing animation that looks plausible but is not necessarily consistent with the real-world clothing states. This goal may require system to extract more information than just skeleton poses [7,67] from the sparse sensor input, e.g. one or few RGB(-D) cameras. In other words, we would like to infer clothing states and dynamics from the sparse driving views, and use the data-driven priors to fill in the missing information.

As an initial step in this direction, in Chapter 5, we present a method to capture temporally coherent dynamic clothing deformation from a monocular RGB video input. In contrast to the existing literature, our method does not require a pre-scanned personalized mesh template, and thus can be applied to in-the-wild videos. To constrain the output to a valid deformation space, we build statistical deformation models for three types of clothing: T-shirt, short pants and long pants. A differentiable renderer is utilized to align our captured shapes to the input frames by minimizing the difference in both silhouette, segmentation, and texture. We develop a UV texture growing method which expands the visible texture region of the clothing sequentially in order to minimize drift in deformation tracking. We also extract fine-grained wrinkle detail from the input videos by fitting the clothed surface to normal maps estimated by a convolutional neural network. Our method produces temporally coherent reconstruction of body and clothing from monocular video. We demonstrate successful clothing capture results from a variety of challenging videos. Extensive quantitative experiments demonstrate the effectiveness of our method on metrics including body pose error and surface reconstruction error of the clothing. This work is published as Xiang, Donglai, et al. ”Monoclothcap: Towards Temporally Coherent Clothing Capture from Monocular RGB Video.” International Conference on 3D Vision (3DV) 2020 [220].

For the last work in this thesis, in Chapter 6, we present avatars with dynamically moving loose clothing that can be faithfully driven by sparse RGB-D inputs as well as body and face motion. We propose a Neural Iterative Closest Point (N-ICP) algorithm that can efficiently track the coarse garment shape given sparse depth input. Given the coarse tracking results, the input RGB-D images are then remapped to texel-aligned features, which are fed into the drivable avatar models to faithfully reconstruct appearance details. We evaluate our method against recent image-driven synthesis baselines, and conduct a comprehensive analysis of the N-ICP algorithm. We demonstrate that our method can generalize to a novel testing environment, while preserving the ability to produce high-fidelity and faithful clothing dynamics and appearance. This work is published as Xiang, Donglai, et al. ”Drivable Avatar Clothing: Faithful Full-Body Telepresence with Dynamic Clothing Driven by Sparse RGB-D Input.” SIGGRAPH Asia 2023 (Conference Track) [219].

1.2 Summary of Contributions

The contributions of this thesis are as follows:

- We present an animatable two-layer full-body avatar model for photorealistic full-body telepresence; our proposed avatar can produce animation that is more temporally coherent, has sharper boundaries and less ghosting artifacts when compared to a single-layer avatar; it also enables editing of the clothing texture that is hard to achieve with previous single-layer model.

- We present animatable clothed human avatars with data-driven photorealistic appearance and physically realistic clothing dynamics from simulation; we develop a deep clothing appearance model to produce photorealistic clothing appearance that bridges the generalization gap between the tracked clothing geometry for training and the simulated clothing geometry at test time; this animation system further enables transferring clothing between different subjects as well as editing of garment sizes.

- We present the first approach for temporally coherent clothing capture from a monocular RGB video without using a pre-scanned template of the subject; we develop a novel method to capture clothing deformation by fitting statistical clothing models to image measurements including silhouette, segmentation, texture and surface normal with a differentiable renderer.

- We develop photorealistic full-body avatars with dynamic clothing that faithfully reproduces the original states of subject’s appearance and geometry, which are driven by sparse (up to 3) RGB-D inputs and enable free-viewpoint rendering; as an important component of our system, we introduce a Neural Iterative Closest Point algorithm that learns to iteratively find the most effective parameter update to track the input point cloud efficiently with a deformation model; our avatars can directly generalize to a novel testing environment with a different background and illumination, while capturing the complex clothing dynamics and preserving the original high-quality appearance from the training data. We also provide an option to finetune the model to adapt to the new appearance in the novel environment.

Chapter 2

Related Work

In this chapter, we examine existing literature related to this thesis. In accordance with the three major components of our system as described in Chapter 1, we also divide related work into three categories. We start by reviewing related work where data-driven statistical priors are utilized to model humans in Section 2.1. Then, in Section 2.2, we move to the topic of clothing animation, where the central idea is to explore physics-based approaches, or to learn from physics. Lastly, we review the literature on clothing capture, including reconstruction and tracking of clothing, in both multi-view and monocular settings in Section 2.3. Related work specific to each project is detailed in the corresponding chapter.

2.1 Data-Driven Human Modeling

Models for dynamic full-body humans have been widely studied because of their applications in the movie and gaming industries, as well as virtual and augmented reality. In this thesis, we also refer to these models as avatars. Some human models focus on the geometry or shape of the human body, with or without clothing; others include appearance as well.

2.1.1 Human Shape Models

Human avatars are usually built to be animatable with joint angles and skinning to allow easy control of skeleton-level deformation. The most commonly used skinning method is Linear Blend Skinning (LBS). A seminal work, SCAPE [6], learns a parametrized human body shape model with LBS deformation from a large-scale dataset of 3D scans. Many methods have been developed in order to reduce the unnatural skinning artifacts that occur with LBS, e.g., [89, 90]. However, a fundamental disadvantage of these approaches is that high-frequency deformations of skin and clothing, such as muscle bulging, folds, and wrinkles, cannot be precisely modeled. In order to solve this problem, pose-dependent blend shapes [102] have been proposed to reduce skinning artifacts. These blend shapes are corrective shapes that can be interpolated with respect to the pose and added to the skinned mesh. A variation of SCAPE that integrates the linear pose-dependent blend shapes, SMPL [120], has been widely used for human modeling and pose estimation.

The models mentioned above can only represent a human body dressed in skin-tight clothing. Several methods have been developed to model complex non-rigid deformation of looser clothing. DeepWrinkle [100] consists of two modules that learn the global cloth deformation in a PCA subspace as well as high-frequency details, such as finer wrinkles, on a normal texture. Ma and colleagues learn a pose-dependent clothing shape from 4D scans with different geometric representations, including mesh-based graph convolution [128], surface elements [126, 127, 129] and implicit functions [169]. Chen and colleagues further propose a generative model of humans with clothing across categories [39]. These methods mostly focus on modeling the clothing geometry, with less effort on creating photorealistic rendering of clothing appearance.

2.1.2 Photorealistic Avatars

Some human models have been developed to not only represent the body geometry, but also the appearance, which is important for photorealistic rendering. The classical approach for modeling full-body appearance is to use a fixed texture associated with the template mesh [29,44,182], which enables the rendering of the avatars by various graphics techniques such as direct shading or ray tracing. However, it is impossible for simple rendering to account for the complicated appearance change that occurs under variations of viewpoint, deformation, material, and lighting. Improved realism has been achieved in classic methods with techniques related to image-based warping [31,51,79,199], but they do not enable body reposing. To achieve photorealistic rendering of animatable full-body human appearance using classical techniques requires heavy computation and the work of digital artists.

In recent years there has been a lot of interest in building photorealistic human avatars directly from captured images in a data-driven manner. In these approaches, a deep neural network takes as input a specific body configuration (for example pose or mesh vertices), and outputs a predicted image supervised by the ground truth capture from the same viewpoint [67, 158, 178, 251]. Data-driven approaches based on image translation networks have also attacked this problem purely from a monocular 2D perspective, using rendered skeletons [32] or 2D body correspondences [205, 206] as input. Some work further utilizes recent breakthroughs in 3D neural rendering, such as Neural Radiance Fields [97,115,135,141,150,151,186,223,249] and Deferred Neural Rendering [62,160,195]. Most of these approaches consider clothing as rigidly attached to the human body, and the results are limited to tight clothing such as T-shirts and pants.

Notably, Codec Avatars [117] have been proposed as a promising approach to achieving photorealistic telepresence. The key idea is to build statistical models of human geometry and texture with Variational Autoencoders (VAE), which can be efficiently encoded for transmission and decoded for rendering. Some follow-up work [118, 119] in this direction further incorporates techniques of volumetric rendering to improve the appearance quality while maintaining similar efficiency for the purpose of real-time telepresence. Although the earliest Codec Avatar work [117] is designed only for the head region, more recent full-body avatars [7,161] unifies face, body and hands in a single model. These avatars can well model large skeleton-level deformation of body and hands by leveraging traditional skinning methods, but struggle at dealing with complex deformation of loose clothing such as temporally dependent dynamics and relative sliding on top of the skin. In this thesis, we build upon the framework of full-body Codec Avatars and address the challenges posed by loose clothing. We model clothing for photorealistic avatars at a fine-grained level from different aspects, including tracking, animation, and rendering, and thus achieve high-fidelity full-body telepresence with loose clothing.

2.2 Clothing Animation

Clothing is an integral part of human appearance, and thus the animation of clothing has long been studied as an important component in character animation. By the term 'clothing animation', we refer to the problem of synthesizing the deformation of a particular garment set given a sequence of underlying body motion as input. Compared with clothing models in Section 2.1, clothing animation focuses on modeling temporal dynamics of garments that cannot be fully determined by the body pose of a single time instant. Compared with clothing capture in Section 2.3, it does not attempt to recover the shape of clothing from captured imagery, but allows for synthesizing clothing motion for unseen body configurations.

As the classical approach for clothing animation, *physics-based simulation* of cloth has been established as a standard tool in the animation and gaming industry. Various aspects of cloth simulation, including speed, robustness, and collision detection and response, have been thoroughly studied [9,19,23,104,106,125,130,136]. In order to further improve animation efficiency, there have been efforts to combine physics-based animation with *data-driven* approaches [91, 202]. A notable paradigm is to use simulation to provide training data to supervise machine learning models that predict clothing dynamics given body pose and shape inputs [11, 40, 64, 72, 84, 146, 152, 170, 172, 198, 246]. A notable paradigm is to use simulation to provide training data to supervise machine learning models that predict clothing dynamics given body pose and shape inputs [11, 40, 64, 72, 84, 146, 152, 170, 172, 198, 246]. Additionally, the use of a self-supervised physics-based loss [10,171] has been explored for the generation of 3D garment deformations.

Although the aforementioned work can produce reasonable clothing shape deformation with dynamics, photorealistic appearance modeling of the animation output is a nontrivial problem, especially for telepresence applications. Classic methods develop specialized BRDF models [235], offline models for knitware like the lumislice [225], or spatially-varying BRDFs which use pattern geometry arrays to improve realism like [43]. More recently, classic graphic methods evolved to generate cloth fibers procedurally on the GPU in real-time, improving rendering quality [215] although still not reaching photorealism. A more complete review of fabric rendering can be found in [30].

2.3 Clothed Body Capture

The capture of human body with clothing has been explored as a source of geometry for garment modeling. Previous work on this topic can be divided into multi-view approaches and single-view approaches.

2.3.1 Multi-View Clothing Capture

Multi-view approaches use a multi-camera system or a 3D scanner to capture and reconstruct the shape of the clothing. In order for the captured clothing shape to be ready for use in downstream tasks, much effort has been devoted to registration, i.e. the representation of the garment geometry by a consistent mesh topology in a temporal sequence [13,22,73,155]. Some work [38, 242] also estimates the underlying body pose and shape from the clothed human reconstruction. In this thesis, the technique of multi-view clothing capture provides the data to train the clothed avatars in Section 3 and Section 4.

Pattern-based clothing registration makes use of identifiable patterns on clothing to explicitly encode correspondences on the captured surface and makes the registration problem easier. Early work [157, 173, 212] utilizes classical computer vision techniques such as corner detection and multi-view geometry to reconstruct and identify printed markers on the garments. A recent work [71] extends the color-coded pattern approach [173] to achieve denser detection. Although this line of work can achieve high-quality registration, it requires clothing with specially designed patterns which are generally not available in the telepresence scenario, and thus is not applicable in this thesis.

2.3.2 Monocular Clothed Human Capture

The early work in this area focuses on estimating human body keypoints from monocular images in 2D [27, 28, 210] or 3D [149, 188, 252]. Because estimating 3D pose from single images is highly ill-posed, deformable human models including SMPL [120], SMPL-X [147] and Adam [85] are used to help with the problem by fitting the models to images [16, 147, 217]. These models not only provide a strong prior for body pose, but also enable estimation of 3D body shape from single images. Deformable human models can be further integrated in deep neural network architectures [87,142,148,226]. These networks are usually trained in a weakly-supervised manner without full 3D supervision.

Because classical deformable human models (SMPL, SMPL-X and Adam) are not able to express clothing shape, the work mentioned above only estimates body shapes with minimal clothing. Recent progress in computer vision makes it possible to reconstruct clothing shape from a single-view input, which allows a much simpler capture setup than the multi-view case. One line of work treats the problem as a monocular 3D shape regression task with explicit [4,250] or implicit [24,75,108,167,168,222] shape representation. Alternatively, optimization-based [68,224] or learning-based approaches [69,70] are used to deform a personalized template shape to match the input image. Most of these methods only focus on estimating clothed body shape without photorealistic texture (except [108]). In addition, due to the fundamental difficulty of estimating depth from monocular RGB images, the output accuracy of these approaches is deficient for high-quality telepresence. Higher depth accuracy can be achieved in a monocular setup by using depth cameras [37, 238], although the quality of the results still does not match that of multiview systems.

A notable line of work aims to capture clothing with the assistance of cloth simulation, both in the cases of monocular [233, 234, 239] and multiview input [65, 111, 183]. However, these methods are still limited in terms of output accuracy. Estimating precise physical parameters solely from visual input is still a difficult problem.

Chapter 3

Modeling Clothing as a Separate Layer

3.1 Introduction

In this work, we seek to build photorealistic full-body clothed avatars that can be animated with driving signals that can be easily accessed, for example, 3D body pose and facial keypoints. Simultaneously modeling both geometry and texture with a deep generative model, like Variational Autoencoders (VAE), has been demonstrated to be an effective way to create photorealistic face avatars [117]. Recently, Bagautdinov and colleagues [7] extend this approach to model full-body avatars with VAE, conditioned on body pose and facial keypoints. Because these conditional signals cannot uniquely describe the states for the clothing, hair and gaze, the VAE latent code is used to distinguish between these different states. In addition, it is essential to disentangle the effects of driving signals and the latent code, in order to reduce the spurious correlations between them.

Despite the progress in previous work [7], challenges still remain in building high-fidelity animatable full-body avatars, and we identify the modeling of clothing as one major difficulty. Artifacts include the imperfect correlation between body pose and clothing state, ghosting effects along the boundary between clothing and skin, as well as loss of wrinkle details and dynamics in the clothing. These artifacts become more noticeable when the captured clothing is loose and the performer moves more dynamically. On the one hand, due to registration error, the network may underfit the data, making it unable to reproduce high-frequency clothing detail; on the other hand, in spite of the disentanglement, the network may still overfit, capturing unwanted chance correlation between the driving signal and the clothing state.

In this work, we explicitly represent the body and clothing as separate layers of meshes in a codec avatar. The separation leads to several benefits. First, it allows us to accurately register both body and clothing, especially with our newly developed photometric tracking approach that uses inverse rendering to align clothing texture to a reference. Second, modeling the body and clothing in separate layers alleviates the aforementioned problem of chance correlation for a single-layer avatar, as the separate layers are naturally disentangled from each other. With our two-layer VAE, a single frame of joint angles can well describe the body state, while the clothing dynamics can be inferred from the sequences of poses

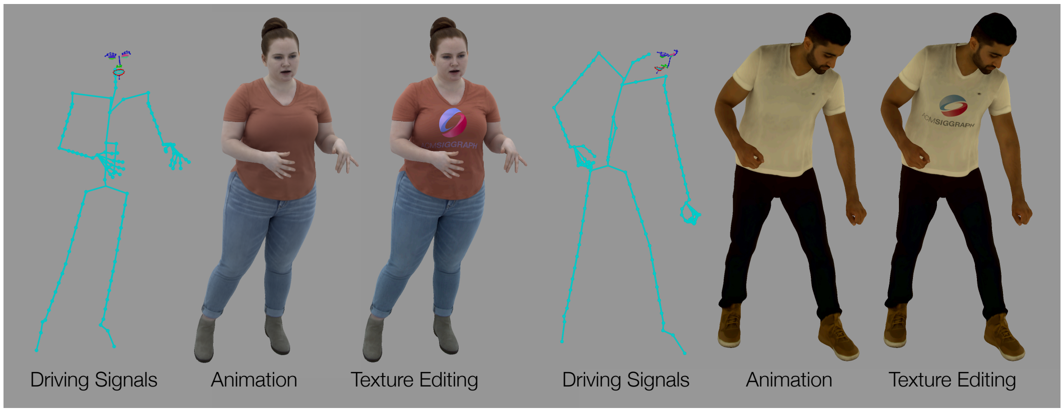

Figure 3.1: Given a novel sequence of skeletal poses and facial keypoints as input, our proposed two-layer codec avatars produce photorealistic animation output, where the clothing texture can be consistently edited. From left to right, we show driving signals, animation output and editing results for two subjects.

with a Temporal Convolutional Network (TCN), which evolves the clothing state in a way that is consistent with the body motion. Third, thanks to the explicit modeling of clothing, the animation output can be further edited by changing the clothing texture.

Our contributions are as follows:

- We present an animatable two-layer codec avatar model for photorealistic full-body telepresence; our proposed avatar can produce more temporally coherent animation with sharper boundaries and fewer ghosting artifacts compared to a single-layer avatar;
- Inverse rendering with our proposed two-layer codec avatar allows a photometric tracking algorithm that aligns the salient clothing texture, significantly improving correspondence in the registered clothing meshes;
- We demonstrate an application of our two-layer codec avatar for editing of the clothing texture that is hard to achieve with the single-layer model used in previous work.

We evaluate the proposed pipeline on the captured sequences of three different actors. We demonstrate the effectiveness of our proposed method against alternative approaches. We show that our model, with only a sequence of poses and facial keypoints as input, achieves high-quality body animation and rendering with photorealistic clothing that can be viewed from arbitrary viewpoints.

This work is published in ACM Transaction on Graphics (SIGGRAPH Asia) 2021.

3.2 Related Work

Our goal in this work is to build a realistic virtual avatar of a human that can be animated by driving signals of skeletal poses and facial keypoints to create a telepresence experience. Most of the related work in this area has been reviewed in Section 2.1. Here we discuss the difference between our work and the most relevent work [7,67]. Bagautdinov and colleagues [7] extend deep appearance models [117] to full bodies. This method can create

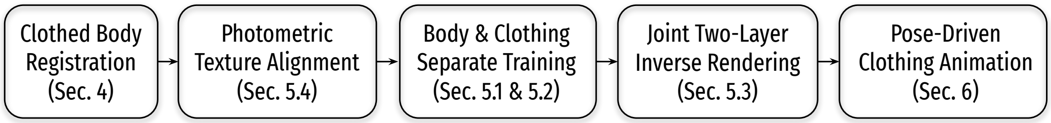
Clothed Body Registration (Sec. 4)
Photometric Texture Alignment (Sec. 5.4)
Body & Clothing Separate Training (Sec. 5.1 & 5.2)
Joint Two-Layer Inverse Rendering (Sec. 5.3)
Pose-Driven Clothing Animation (Sec. 6)

Figure 3.2: An overview of our proposed method in procedural order.

photorealistic full-body avatars that are animatable by joint angles. In this approach, the clothing is modelled only implicitly in the same layer as the human body. Therefore, this method works well when the clothing tightly follows the body motion, but may struggle in settings where clothing is loose or exhibits significant dynamics. Habermann and colleagues [67] address a similar problem of creating a dynamic free-view point rendering of a specific subject given skeleton motion as input. It uses a neural network to regress the clothed body shape represented by an embedded graph plus additional deformation and a dynamic texture. Compared with this work, our method uses a two-layer formulation for both registration and modeling that enables high-quality animation output.

3.3 Method Overview

Our goal in this work is to build full-body clothed digital avatars that enable photorealistic rendering from any viewpoint. To make the avatars useful, they should be animatable given some driving signals that can be obtained at modest cost. We choose 3D skeletal joint angles and facial keypoints as the input conditioning, similar to previous work [7]. For example, these driving signals can be obtained by multi-view triangulation and inverse kinematics from a sparse set of cameras.

The central idea of our method is to explicitly represent body and clothing as two separate layers. We take this approach for three reasons. First, we notice that the deformation of the body and the clothing follow different movement patterns because of their different dynamics. A single frame of joint angles in the driving signal can largely determine the body state through Linear Blending Skinning (LBS) and pose-dependent deformation. In contrast, the dynamics of clothing can vary too much to be described only by current body pose without considering temporal information. Thus the body and clothing layers need to be controlled by different input conditioning. Second, in the single-layer registration of the body with the clothing, a specific vertex along the clothing boundary can belong to either the body region or the clothing region in different frames due to the sliding motion of the clothing relative to the body, which violates the single layer assumption. A codec avatar trained with such data often has a color between the clothing and skin colors in such a region, leading to ghosting effects around the sleeves and neck of the garment. Although disentanglement could alleviate this kind of artifact, it cannot eliminate it due to limited training data capturing the complex interaction between clothing and the body. In our work, with the registration of body and clothing in separate layers, such artifacts can be avoided because each vertex is either part of body or the clothing across all frames. Third, separate layers for body and clothing open up opportunities for further changing the appearance of the avatar, such as temporally consistent editing of the clothing texture without interfering with the body appearance. This capability will also make it possible to alter the clothing style through physical simulation in Chapter 4.

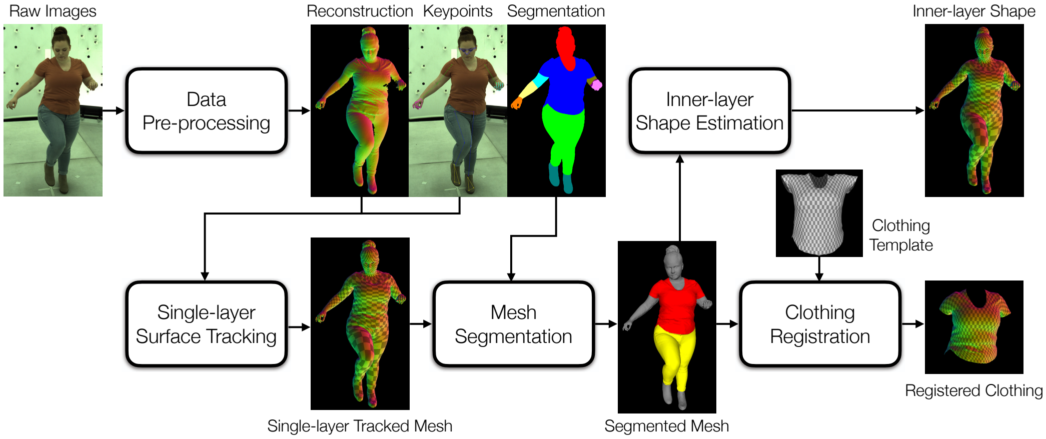

Figure 3.3: The clothed body registration pipeline that we use to generate training data for our two-layer codec avatars.

In this work, we assume that the subject to be modeled wears a T-shirt and pants. We only model the T-shirt in the second, outer layer because it exhibits most of the dynamics and variations in geometry and texture. In the inner layer, we model the body region covered by the outer layer (torso and upper arms) and the rest of human surface, including the head, arms, pants[1] and shoes.

In Section 3.4, we briefly describe our two-layer geometry-based surface registration method to generate the necessary training data for the codec avatars. In Section 3.5, we present our two-layer codec avatars. We describe the architecture of the body branch in Section 3.5.1 and clothing branch in Section 3.5.2, as well as the joint training of both branches through inverse rendering in Section 3.5.3. In Section 3.5.4, we propose a method for texture alignment to improve the photometric correspondences between registered clothing meshes across different frames. In Section 3.6, we present the temporal model used to animate our clothed avatars using a sequence of joint angles as the driving signal. A visualization of the method is shown in Figure 3.2.

3.4 Clothed Body Registration

The pipeline to generate the data for training our two-layer codec avatars is illustrated in Figure 3.3. Our goal is to register the body and clothing geometry in two separate layers. A more detailed description of this pipeline can be found in Section 3.8.1.

Data preprocessing. The input to our pipeline is a sequence of RGB images of the subject captured by a synchronized multi-camera system. The raw RGB images are used to create a dense 3D reconstruction of the human surface with a multi-view Patchmatch reconstruction algorithm [56]. An example of the reconstructed mesh can be seen in Figure 3.3. In addition, we obtain a part segmentation of different body and clothing regions for each captured image. We also run 2D keypoint detection for the body, face and hands, which are

[1] The pants of the captured subjects in this work are tight and thus not worth the effort of modeling as a separate layer. We demonstrate in the results that the advantage of clothing modeling as a separate layer is obvious when the garment is loose.

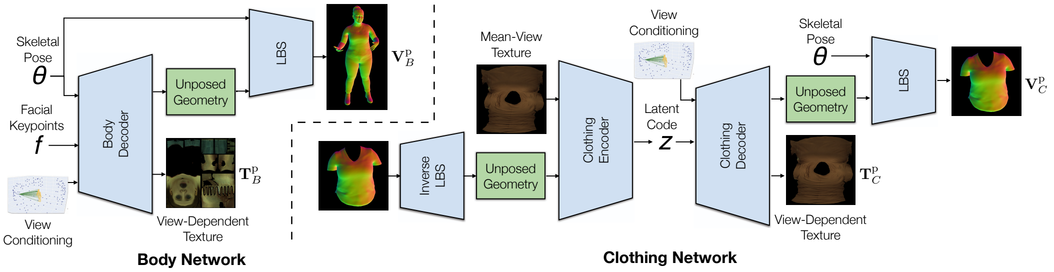

Figure 3.4: Network architecture of our two-layer full-body codec avatar. We show the body network on the left and the clothing network on the right, including the input and output of each network.

triangulated to obtain 3D keypoints.

Single-Layer Surface Tracking. We non-rigidly register the reconstructed meshes with a kinematic body model, similar to [242] and [201]. We estimate a personalized rest-state shape and a set of of joint angles for each frame by minimizing the difference between the LBS output and the reconstructed surface, as well as the 3D keypoints in the previous step. We further perform free-form Iterative Closest Points (ICP) registration using the skinned kinematic model as initialization.

Mesh Segmentation. In this step, we segment the single-layer tracked meshes into separate body and clothing parts. We unproject the image segmentation labels onto the mesh and for each vertex take the majority of votes across different views. Similar to [155], we also run the Markov Random Field (MRF) to remove noisy segmentation labels.

Clothing Registration. Our clothing registration step is similar to [155]. We manually create a template clothing mesh and use it to register the clothing region of the single-layer tracked mesh for each frame. Essentially we run a non-rigid ICP algorithm that aligns the template and target clothing region. To provide good initialization for the optimization, we find it useful to apply Biharmonic Feformation Fields [77] which generate a deformed template mesh whose boundary is directly aligned with the target clothing boundary with the lowest possible interior distortion.

Inner-Layer Shape Estimation. The inner-layer geometry consists of two parts: the invisible body region covered by the clothing in the upper body, which we estimate using the method in [242], and the visible region of the human surface, which can be directly obtained by matching with the single-layer tracking results. Unlike [242], we only need to estimate the underlying body shape of the upper body, because the pants are treated as part of the inner layer in this work.

3.5 Clothed Body Modeling

We now present our two-layer codec avatars with explicit clothing modeling. Similar to [117] and [7], we employ a Variational Autoencoder (VAE) as our generative model. In our two-layer formation, we train a separate network to learn the deformation space for body and clothing, while the correlation between body and clothing can be learned afterwards with a temporal model for animation. To this end, we train a body decoder which takes the skeletal pose as input, and predicts geometry and view-conditioned texture for the inner body layer, as shown on the left of Figure 3.4. Similarly, we train a clothing decoder with a VAE, as shown on the right of Figure 3.4. Similar to existing approaches to body modeling [120, 143], we only learn the geometry in the canonical pose space for both the body layer and the clothing layer by applying an inverse LBS transform. This technique reduces the deformation space that needs to be learned. In the following sections, we introduce the detailed structure for the body and clothing networks, and explain how we train them. Implementation details including loss weights and network architecture can be found in Section 3.8.2 and Section 3.8.3.

3.5.1 Body Decoder

As shown on the left of Figure 3.4, our body network is similar to the decoder structure in [7], without the encoder. Once the clothing is decoupled from the body, the skeletal pose and facial keypoints contain sufficient information to describe the body state (including pants that are relatively tight). We do not use a latent code as conditioning for the body network to avoid the difficult problem of disentanglement between the latent space and the driving signal, as described in [7]. Our body decoder takes in the skeletal pose, facial keypoints and view-conditioning as input, produces unposed geometry in a UV positional map and view-dependent texture for the body as output. A LBS transformation is then applied to the unposed mesh restored from the UV map to produce the final output mesh.

The loss function to train the body network is defined as:

$$\begin{aligned} E^{B}_{\text{train}} = \lambda_g \|\mathbf{V}^{\text{p}}_B - \mathbf{V}^{\text{r}}_B\|^2 + \lambda_{lap} \|\text{L}(\mathbf{V}^{\text{p}}_B) - \text{L}(\mathbf{V}^{\text{r}}_B)\|^2 \\ + \lambda_t \|(\mathbf{T}^{\text{p}}_B - \mathbf{T}^{\text{t}}_B) \odot M^{\text{V}}_B\|^2, \end{aligned} \tag{3.1}$$

where $\mathbf{V}^{\text{p}}_B$ is the vertex position interpolated from the predicted position map in UV, and $\mathbf{V}^{\text{r}}_B$ is the vertex from inner layer registration from Section 3.4, $L(\cdot)$ is the Laplacian operator, $\mathbf{T}^{\text{p}}_B$ is the predicted texture, $\mathbf{T}^{\text{t}}_B$ is the reconstructed texture per-view, and M^{V}_B is the mask indicating the valid UV region.

3.5.2 Clothing Network

As shown on the right of Figure 3.4, we model the clothing appearance with a Conditional Variational Autoencoder (cVAE). The encoder takes as input the unposed clothing geometry and mean-view texture, and produces parameters of a Gaussian distribution, from which a latent code $\mathbf{z}$ is sampled. Besides the latent code, the decoder also takes spatial-varying view conditioning as input, and predicts geometry and texture for the clothing. Then, the training loss is described as:

$$\begin{aligned} E^{C}_{\text{train}} = \lambda_g \|\mathbf{V}^{\text{p}}_C - \mathbf{V}^{\text{r}}_C\|^2 + \lambda_{lap} \|\text{L}(\mathbf{V}^{\text{p}}_C) - \text{L}(\mathbf{V}^{\text{r}}_C)\|^2 \\ + \lambda_t \|(\mathbf{T}^{\text{p}}_C - \mathbf{T}^{\text{t}}_C) \odot M^{\text{V}}_C\|^2 + \lambda_{\text{kl}} E_{\text{kl}}, \end{aligned} \tag{3.2}$$

where $\mathbf{V}^{\text{p}}_C$, $\mathbf{V}^{\text{t}}_C$, $\mathbf{T}^{\text{p}}_B$, $\mathbf{T}^{\text{t}}_B$, and M^{V}_C are all defined similarly to the parameters in the body decoder but with respect to clothing, E_{kl} is a conventional KL divergence loss.

3.5.3 Inverse Rendering with Two-layer Representation

The ICP-based clothing registration algorithm in Section 3.4 and previous work [155] aims to align the boundary of the clothing template with the target area, while there is no explicit constraint for the interior correspondences except for the mesh regularization. Therefore, the registered meshes from Section 3.4 may suffer from correspondence errors in the interior (see the first column of Figure 3.8), which significantly influences the decoder quality, especially for dynamic clothing. In order to correct the correspondences in the training stage, we need to link the predicted geometry and texture to the input multi-view images in a differentiable way.

To this end, after the body and clothing networks are separately trained as described in Section 3.5.1 and 3.5.2, we jointly train the body and clothing networks by rendering the output with a differentiable renderer. We use the following loss functions:

$$\begin{aligned} E^{\text{inv}}_{\text{train}} = \lambda_i \|\mathbf{I}^{\text{R}} - \mathbf{I}^{\text{C}}\| + \lambda_m \|\mathbf{M}^{\text{R}} - \mathbf{M}^{\text{C}}\| \\ + \lambda_v E_{\text{softvisi}} + \lambda_{lap} E_{\text{lap}}, \end{aligned} \tag{3.3}$$

where $\mathbf{I}^{\text{R}}$ and $\mathbf{I}^{\text{C}}$ are the rendered image and the captured image, $\mathbf{M}^{\text{R}}$ and $\mathbf{M}^{\text{C}}$ are the rendered foreground mask and the captured foreground mask, and E_{lap} is the Laplacian geometry loss similar to that defined in Eq. 3.1 and 3.2. E_{softvisi} is a soft visibility loss, similar to [116], that is specifically designed to handle the depth reasoning between the body and clothing so that the gradient can be back-propagated through if the depth order is wrong. In detail, we define the soft visibility for a specific pixel as

$$S = \sigma\left(\frac{D^{\text{C}} - D^{\text{B}}}{c}\right), \tag{3.4}$$

where $\sigma(\cdot)$ is the sigmoid function, D^{C} and D^{B} are the depth rendered from the current viewpoint for the clothing and body layer, and c is a scaling constant. Then the soft visibility loss is defined as:

$$E_{\text{softvisi}} = S^2, \tag{3.5}$$

when $S > 0.5$ and the current pixel is assigned to be clothing according to the 2D cloth segmentation. Otherwise, E_{softvisi} is set to 0. If the pixel is labeled as clothing but the body layer is on top of the clothing layer from this viewpoint, the soft visibility loss will back-propagate the information to update the surfaces until the correct depth order is achieved.

Following [7] in this inverse rendering stage, we also use a shadow network that computes quasi-shadow maps for body and clothing given the ambient occlusion maps. In contrast to the approach of [7] where the ambient occlusion is approximated with the body template after the LBS transformation, we compute the exact ambient occlusion using the output geometry from the body and clothing decoders because we aim to model a more detailed clothing deformation than can be produced by the LBS transformation. The quasi-shadow map is then multiplied with the view-dependent texture before applying the differentiable renderer.

3.5.4 Texture Alignment with Inverse Rendering

The inverse rendering method mentioned in Section 3.5.3 already has the capability to improve photometric correspondences to some extent, because the network tends to predict texture with less variance across frames, along with deformed geometry to align the rendering output with the ground truth images. Ideally we only need to train the two decoders

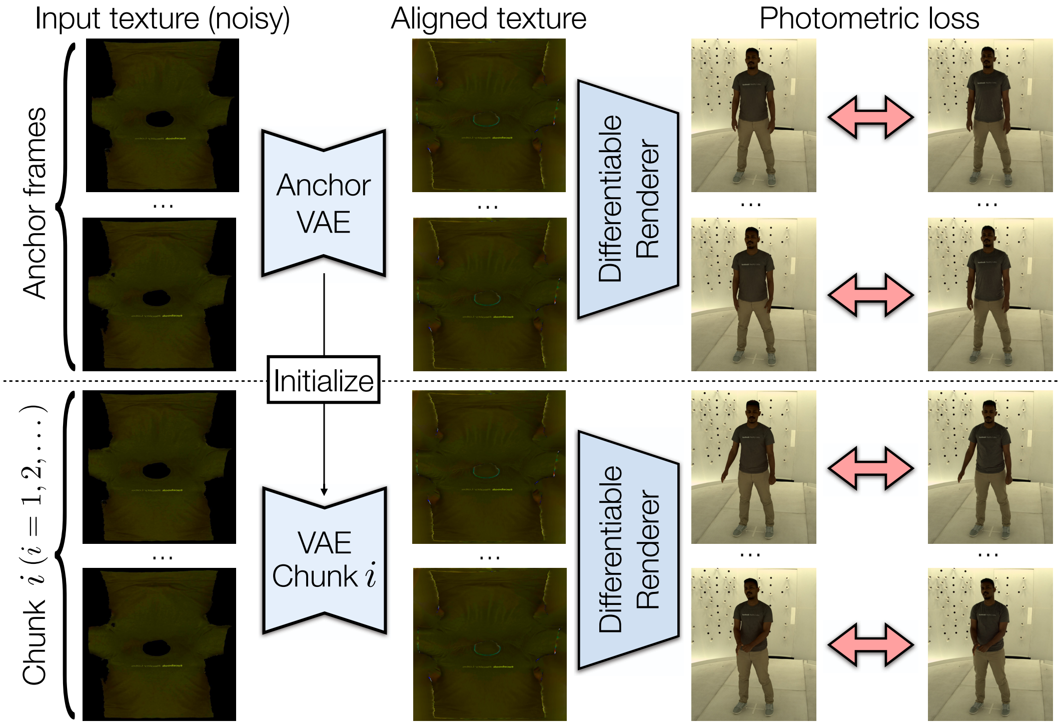

Figure 3.5: Our inverse-rendering-based photometric texture alignment method (Section 3.5.4). First, the anchor frames are used to train the anchor VAE with photometric loss applied to the differentiable rendering output. Then, a separate VAE for each chunk of frames is initialized independently from the anchor VAE and trained using the same loss function. Here we only show the texture and omit the geometry in the VAE input and output for clarity.

simultaneously with the inverse rendering loss to correct the correspondences while creating the generative model for driving the animation. However, we find that this alone would not correct all the correspondence errors. The model might not find a good minimum for two reasons. First, the variation in photometric correspondences in our initial registration may be too large for the network to fix. Secondly, our VAE model with view conditioning may allow the decoder to cheat with the view-dependent texture rather than moving the geometry.

These problems motivate us to propose a new way to use inverse rendering for correspondence improvement. First, we separate the registered meshes into chunks of 50 neighboring frames. Then, we select the first chunk as the anchor frames, and train an anchor network for this chunk using the inverse rendering model described in Section 3.5.3. After convergence, we use the trained network parameters to initialize the training of other chunks. To make sure that the alignment of the other chunks does not drift from the anchor frames, we set a small learning rate (1e-4 for the AdamW optimizer), and mix the anchor frames with each other chunk during training. We remove the view conditioning from the texture branch of our decoder in Section 3.5.3, and use a single texture prediction for inverse rendering in all the camera views. The output geometry predicted by the

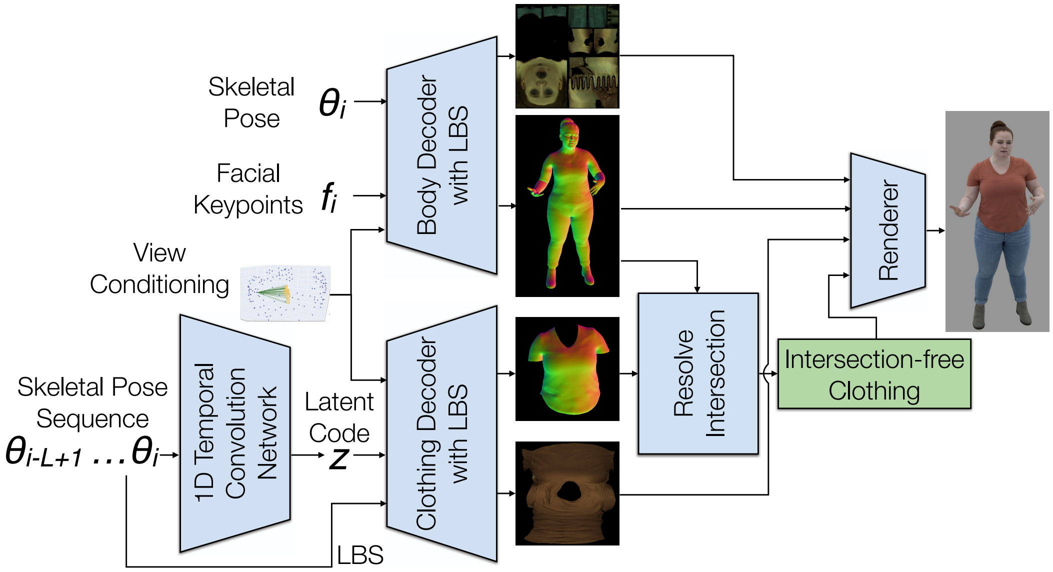

Figure 3.6: The clothed body animation pipeline.

network of each chunk after training has more consistent correspondences across frames compared with the input, which is manifested by the consistent projected texture pattern in the UV space shown in Figure 3.8. A visual illustration of this process is provided in Figure 3.5. This method has a similar spirit to previous UV-template-based texture alignment approaches [17,59], but naturally extends the idea to a neural-network formulation under the framework of codec avatars.

The method described here is applied after the two-layer registration is obtained in Section 3.4, as shown in Figure 3.2. For each frame, we use the output geometry predicted by the network as a new registered mesh with the improved correspondences. We use these data to train the body and the clothing networks, as described in Section 3.5.1-3.5.3.

3.6 Temporal Modeling for Pose-Driven Clothing Animation

In our two-layer codec avatars, the body output is conditioned on a single frame of skeletal pose and facial keypoints, while the clothing state is determined by the latent code. In order to animate the clothing from the driving signal, we use a Temporal Convolution Network (TCN) to learn the correlation between body dynamics and clothing deformation. Our TCN takes in the sequence of previous and current skeletal pose and infers the latent clothing state.

An illustration of our animation pipeline is shown in Figure 3.6. The temporal convolution network takes as input the joint angles in a window of L frames up to the target frame, and passes through several 1D temporal convolution layers to predict the clothing latent code for the current frame $\mathbf{z}$. To train the TCN, we minimize the following loss function:

$$E^{TCN}_{\text{train}} = \|\mathbf{z} - \mathbf{z}^{\text{c}}\|^2, \tag{3.6}$$

where $\mathbf{z}^{\text{c}}$ is the ground truth latent code obtained from the trained clothing VAE.

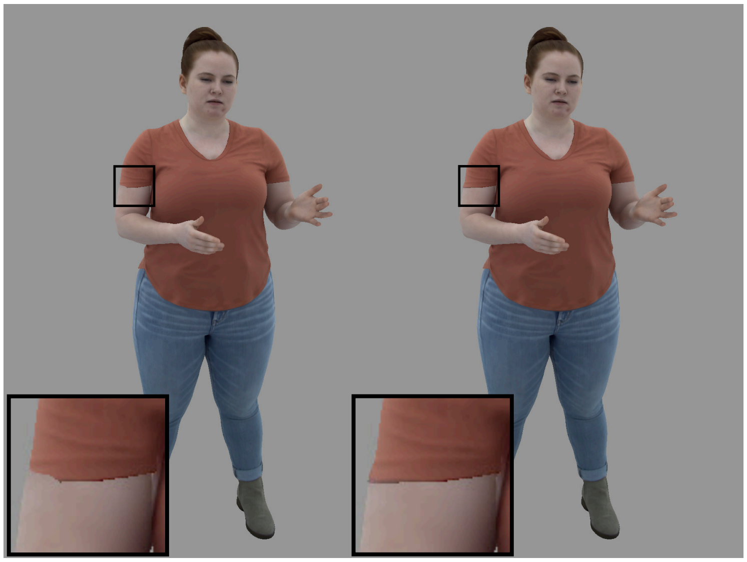

Figure 3.7: An example of resolving intersection. The intersecting area is highlighted by the zoomed boxes.

An alternative formulation would be to condition the prediction on not just previous body states, but also previous clothing states. This formulation is inspired by cloth simulation, where the clothing vertex position and velocity in the previous frame are needed to compute the current clothing state. However, in our data-driven setting, we find that such an auto-regressive model that takes in previous clothing states is hard to train and does not outperform the non-autoregressive model given the limited amount of data (25 min). Therefore, the input to our TCN is a temporal window of skeletal poses, not including the previous clothing states.

Resolving Intersection. One solution is to add a training loss for TCN to make sure that the predicted clothing does not intersect with the body. However, even without a loss to penalize intersection, the clothing states predicted by our TCN model already match the body shape well, resulting in only minimal intersection. Thus we only need to resolve intersection as a post processing step. We project the intersecting clothing back onto the body surface with an additional margin in the body normal direction. This operation will solve most intersections and make sure that the clothing and body are in the right depth order for rendering. An example of these results can be seen in Figure 3.7.

3.7 Results

In this section, we first introduce our capture system and captured data. Then we show the results of our photometric texture alignment method to demonstrate its effectiveness in achieving better photometric correspondence in the UV space. After that, we show the animation output of our two-layer codec avatars with explicit clothing modeling. In particular, we demonstrate the advantage of our two-layer formulation against the single-layer model

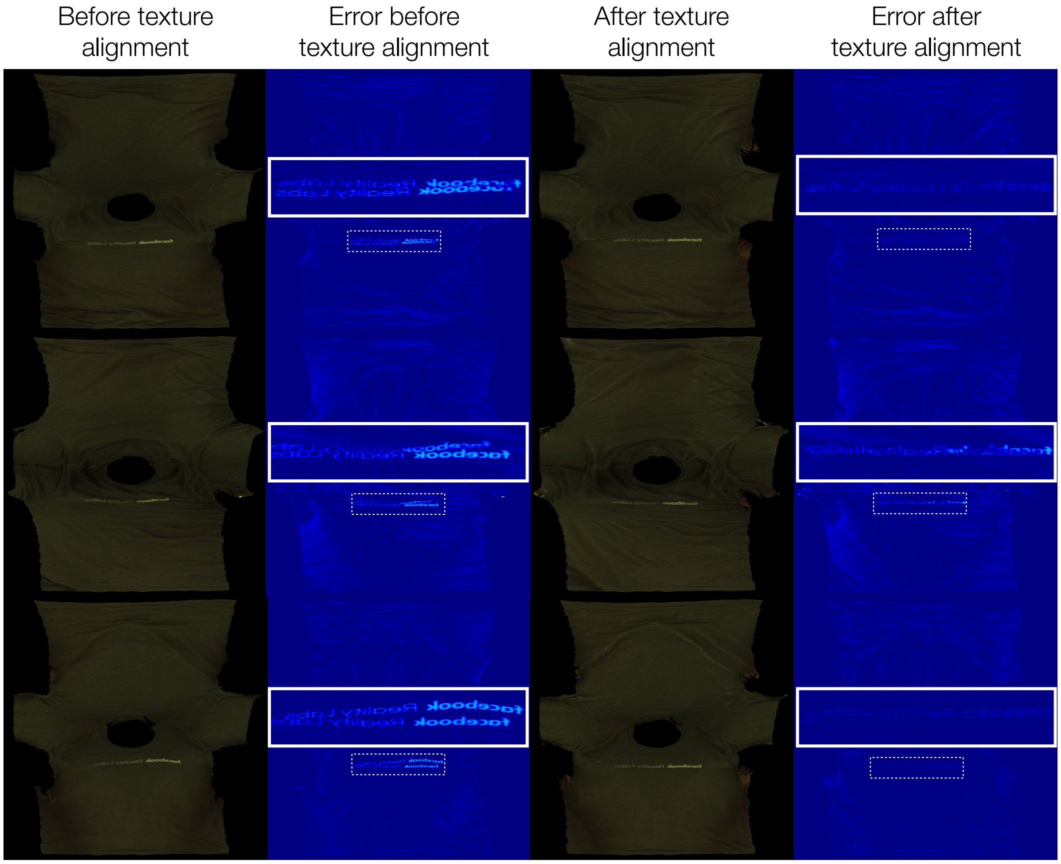

Figure 3.8: Inverse-rendering-based texture alignment results. From left to right, we show (1) projected texture on the clothing mesh before texture alignment, (2) error map between the first column and the mean texture of anchor frames, (3) projected clothing texture after texture alignment, and (4) the difference between the third column and the mean texture of anchor frames. The error maps are visualized with the Jet colormap; lighter color represents larger error. We also show a zoomed-in version of the text region to highlight the difference.

in previous work. We close by demonstrating clothing texture editing for animation.

3.7.1 Data Capture

The training data for our codec avatars are captured by a multi-view capture system consisting of around 140 cameras that are distributed uniformly on a half dome above the ground. All the cameras run with hardware synchronization, capturing at the resolution of 4096×2668 and 30 fps. Three identities, one female (*Subject 1*) and two males (*Subject 2* and *Subject 3*), are captured with a pre-defined acting script. The script is designed to capture peak poses with the activation going through all body joints, followed by a 10-minute conversation to capture social behavior. For each subject, we collect sequences of 40k-50k frames in total and intentionally leave out approximately 4-5k contiguous frames for testing.

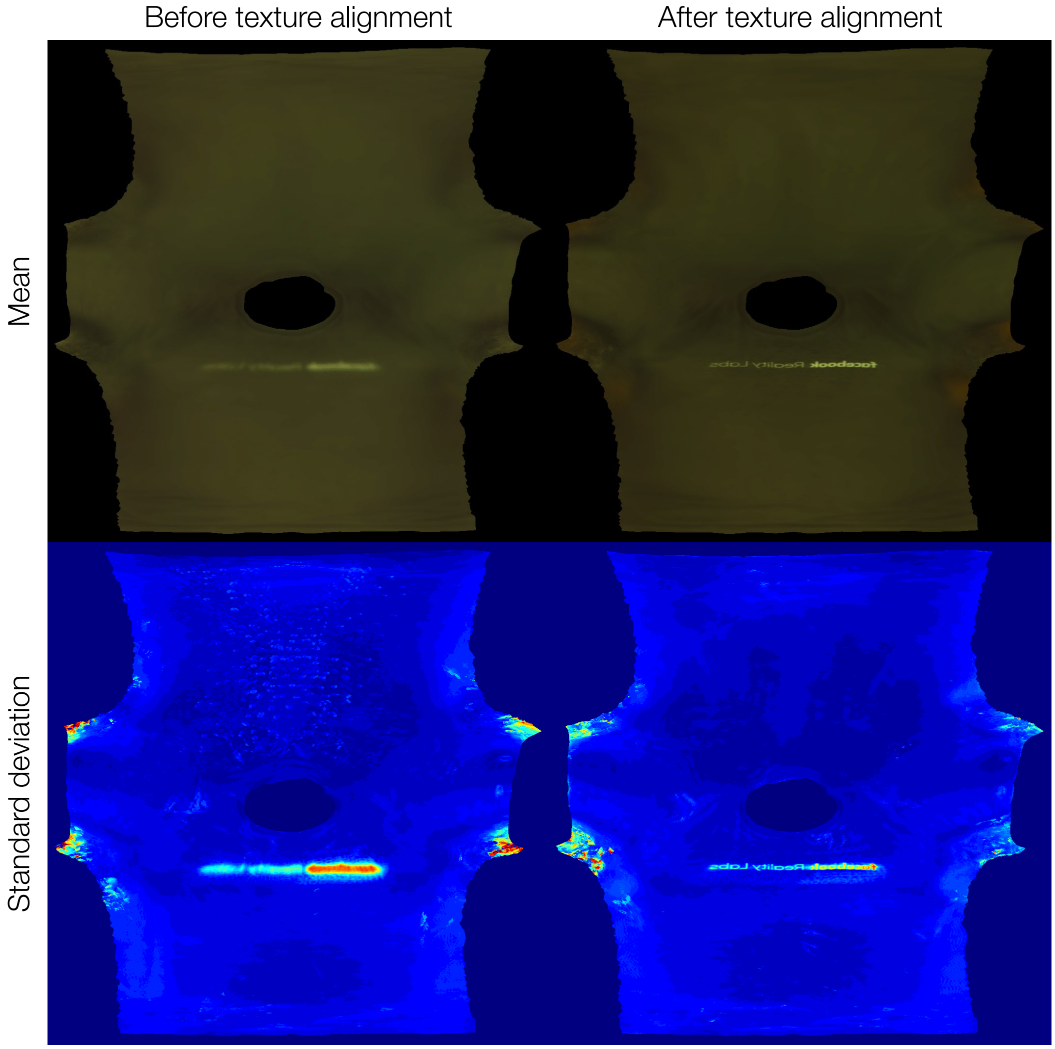

Figure 3.9: Mean (top row) and standard deviation (bottom row, converted to jet colormap) of unwrapped texture before (left column) and after (right column) texture alignment on the sequence of *Subject 2*.

3.7.2 Texture Alignment with Inverse Rendering

In this section, we show the results of texture alignment based on inverse rendering (Section 3.5.4) on the sequence of *Subject 2*. Textures are projected from the raw captured images to the registered meshes before and after the texture alignment procedure, and then unwrapped into the UV space for comparison. Example results for several frames are shown in the first and third column of Figure 3.8. To assess the quality of alignment, we compare the mean UV texture of the anchor frames with the unwrapped texture of each individual frame. The error map is then visualized by the Jet colormap, shown in the second and fourth column of Figure 3.8 respectively.

The visible pattern in the heatmap before texture alignment (the second column) verifies the lack of accurate interior correspondences in the registered clothing meshes from the ICP algorithm (Section 3.4). After the texture alignment (the fourth column), the error between the UV texture of those frames and the mean of anchor frames is significantly reduced. This result suggests that the correspondences in the mesh interior are improved in the inverse rendering process, and demonstrates the effectiveness of our texture alignment method.

To statistically evaluate the quality of photometric correspondence in the UV space, we compute the mean and standard deviation of the unwrapped texture across different frames, as visualized in Figure 3.9. Comparing the mean texture images, we observe a much sharper text pattern after texture alignment than before. Similarly, the standard deviation after texture alignment becomes smaller and more concentrated in the spatial domain. This result also verifies the improvement of photometric correspondence thanks to our texture alignment approach.

3.7.3 Pose-Driven Animation

In this section, we present animation results produced by our two-layer codec avatars driven by the 3D skeletal pose and facial keypoints. In our animation pipeline, the body decoder is directly driven by skeletal pose and facial keypoints of the current frame; on the other hand, the clothing decoder is driven by latent clothing code generated by the temporal clothing model in Section 3.6, which takes a temporal window of history and current poses as input. We compare the quality of our animation with previous work [7] that uses a single-layer codec avatar. We follow the method described in [7] to animate the single-layer codec avatar: we randomly sample the unit Gaussian distribution, and use the resulting noise values for imputation of the latent code. The sampled latent code, the skeletal pose and facial keypoints are fed together into decoder network. We present qualitative animation results on all three testing sequences, shown in Figure 3.10, 3.11, and 3.12. Our animation results are better seen in the supplementary video[2].

Our two-layer formulation helps remove the severe artifacts in the clothing regions in the animation output of [7], especially around the clothing boundary of Figure 3.10, and 3.12. Indeed, as the body and clothing are modeled together, the single-layer avatars rely on the latent code to describe the many possible clothing states corresponding to the same body pose. During animation, however, the absence of a ground truth latent code leads to degradation of the output, despite the efforts in [7] to disentangle the latent space from the driving signal. In contrast, our animation model achieves better animation quality by separating body and clothing into different modules: the body decoder can determine the body states given the driving signal of the current frame; the temporal model learns to infer the most plausible clothing states from body dynamics for a longer period; the clothing VAE ensures a reasonable clothing output given its learned smooth latent manifold. In addition, our two-layer avatars show results with a sharper clothing boundary and clearer wrinkle patterns in these images.

We also quantitatively compare the animation output of our two-layer codec avatars with the baseline method [7] by evaluating the output images against the captured ground truth images. We report the evaluation metrics of Mean Square Error (MSE) and Structural Similarity Index Measure (SSIM) over the foreground pixels. The results are shown

[2] https://drive.google.com/file/d/1MZhx5VGrfYEDQCpTREtOVGBPkf2WuafU/view?usp=share_link

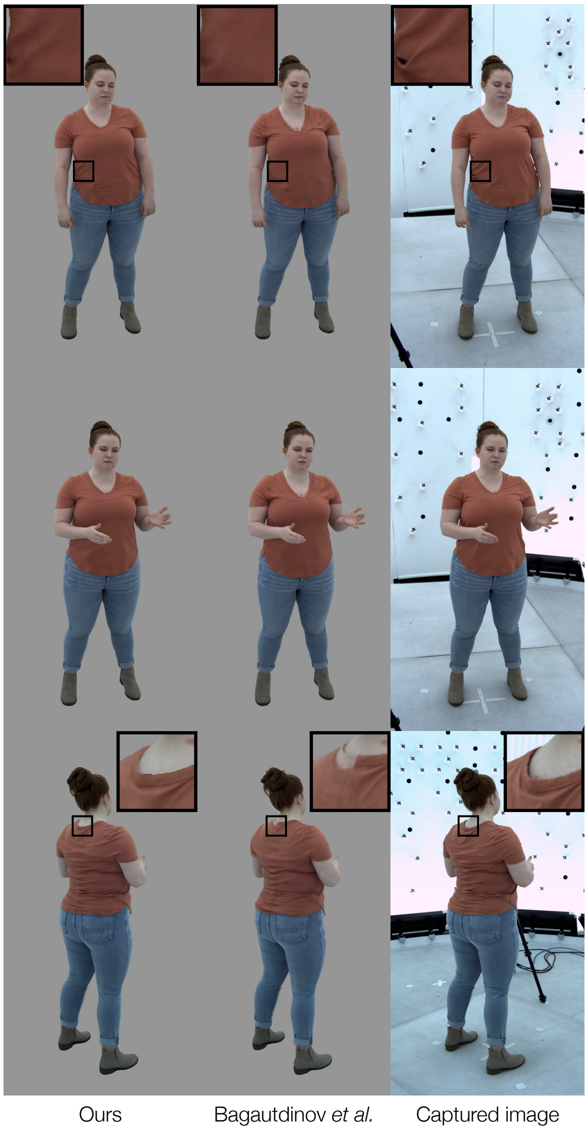

Figure 3.10: Comparison of animation output between our proposed method and baseline [7] on the *Subject 1* sequence.

in Table 3.1. Our method consistently outperforms [7] on all three sequences and both evaluation metrics. In particular, it is worth noting that our advantage on MSE is most obvious on the sequence of *Subject 3*, who is wearing a loose T-shirt that is hard to model with the single-layer avatar. This result agrees with our qualitative observation of the images as well.

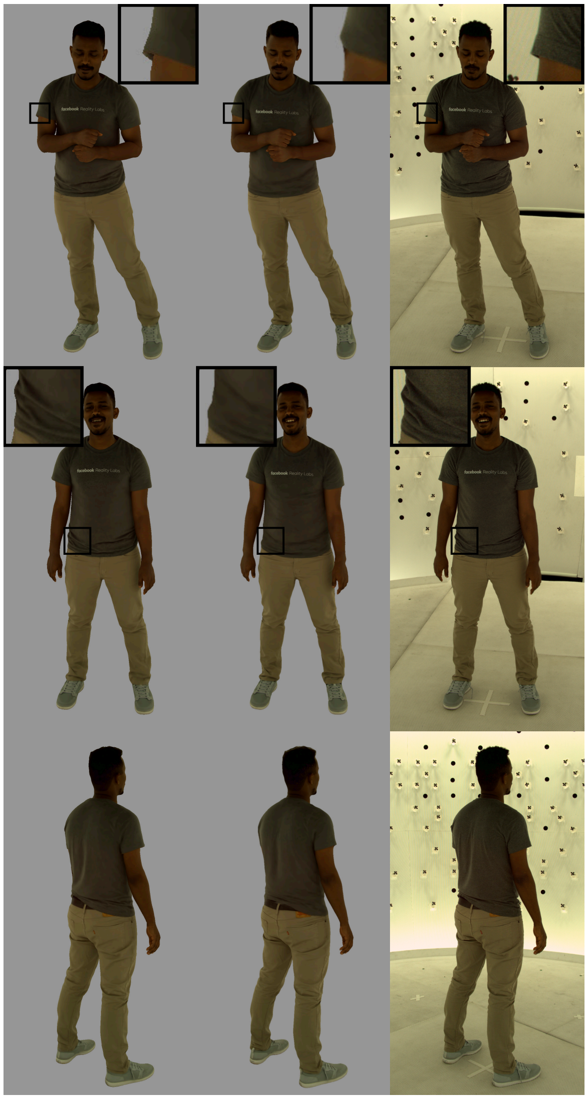

Figure 3.11: Comparison of animation output between our proposed method and baseline [7] on the *Subject 2* sequence.

3.7.4 Ablation Analysis

In this section, we present an ablation analysis on several different components in the design choice of our system. The results are shown in Figure 3.13.

First, we analyze our design of VAE (Section 3.5.2) + temporal modeling (Section 3.6) for clothing animation. One alternative for this design is to combine the functionality of these two networks into one: to train a decoder that takes a sequence of skeleton poses as input and predicts clothing geometry and texture as output. The result of this compari-

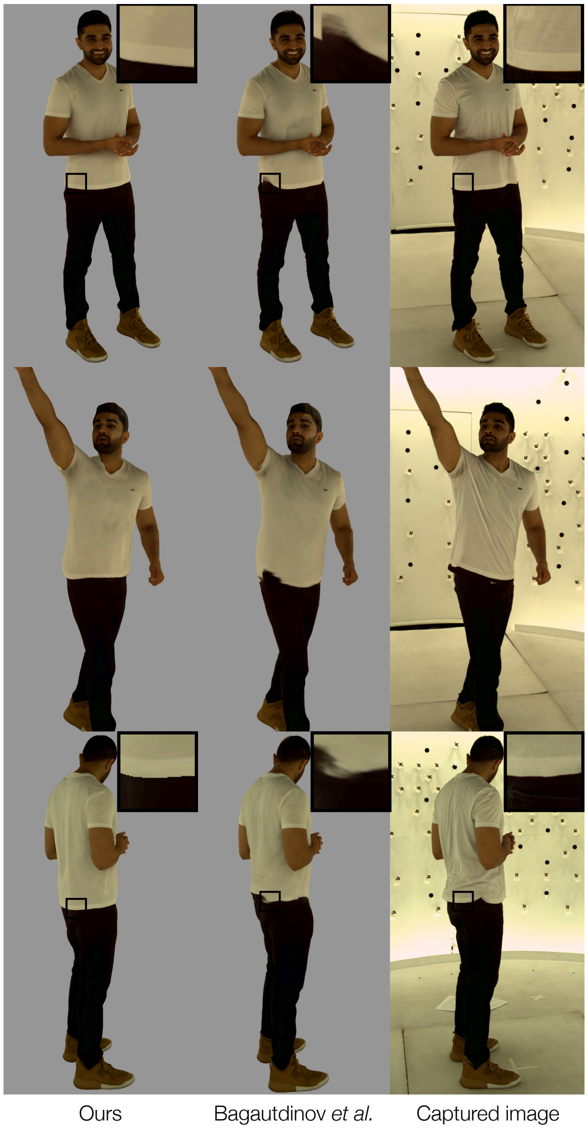

Figure 3.12: Comparison of animation output between our proposed method and baseline [7] on the *Subject 3* sequence.

son is shown on the left of Figure 3.13. Here, the baseline model produces blurry output around the logo on the T-shirt. Even a sequence of skeleton poses does not contain enough information to fully determine the clothing state. Therefore, similar to the analysis in [7], directly training a regressor from the information-deficient input to final clothing output leads to underfitting to the data by the model. In contrast, in our proposed system, the VAE network can model different clothing states in detail with a generative latent space, while the temporal modeling network infers the most probable clothing state. In this way, our

Table 3.1: Quantitative comparison between our proposed method and the previous work. We report Mean Square Error (lower better) and the Structural Similarity Index Measure (higher better) on all three testing sequences.

Sequence	[7]		Ours	
	MSE↓	SSIM↑	MSE↓	SSIM↑
Subject 1	100.57	0.8720	**74.73**	**0.8816**
Subject 2	81.95	0.8804	**58.14**	**0.8917**
Subject 3	456.20	0.8159	**356.52**	**0.8230**

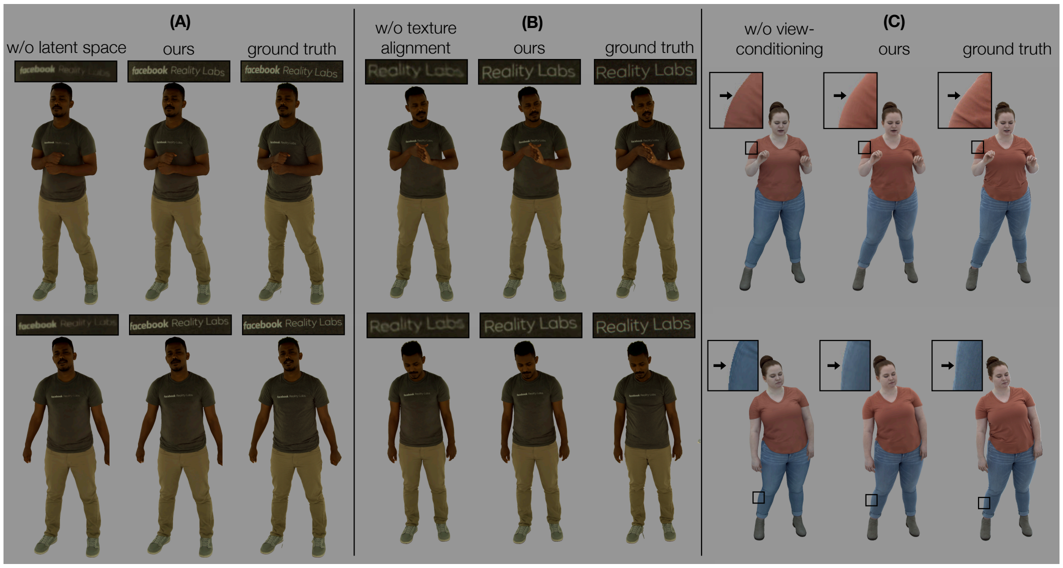

Figure 3.13: Ablation analysis of system components. In (A) we compare our results with a model without clothing VAE latent space for clothing, instead directly regressing clothing geometry and texture from a sequence of skeleton poses as input. In (B) our output is compared with the model trained using data without the texture alignment step. In both (A) and (B) our method shows sharper logo pattern. In (C), we show results with (ours) and without view-conditioning effects. Notice the strong reflectance of lighting near the silhouette of subject captured by our view-conditioning modeling.

method can produce high-quality animation output with sharp detail.

Next, we demonstrate the influence of photometric texture alignment (Section 3.5.4) on the final animation output. We compare the results generated by our full model, which is trained on registered body and clothing data with texture alignment, against a baseline model trained on data without texture alignment (output of Section 3.4). The result is shown in the middle of Figure 3.13. We see that photometric texture alignment also helps to produce sharper detail in the animation output, as the better texture alignment makes the data easier for the network to model.

In addition, we also validate the ability of our network to generate view-dependent effects. We compare our full model with a baseline model where the body and clothing networks do not take view conditioning as input. The results are shown on the right of Figure 3.13. Our output with view-dependent effects is visually more similar to the ground truth than the baseline model without view conditioning. The most obvious difference is observed near the silhouette of the subject, where the view-dependent output is brighter due to Fresnel reflectance when the incidence angle gets close to 90° [99], an important factor that makes the view-dependent output more photo-realistic.

In the supplementary video, we show a comparison of animation results using different lengths of temporal window L as input to our TCN (Section 3.6), including $1, 3, 8, 15, 30, 60$. We observe that using a small temporal window length (for example $L = 1, 3, 8$) leads to unnatural jittering in the animation output. Our analysis is that, as in the situation with [7], similar poses (or short pose sequences) in the training set may correspond to drastically different clothing states, and the network can overfit to the data by trying to distinguish between the nuances in pose variation, thus predicting highly different clothing states in consecutive frames. By contrast, using a larger temporal window length allows the network to take a longer history information of body motion into consideration, and thus becomes less prone to the overfitting problem. In addition, the temporal convolution architecture itself tends to predict temporally smooth output, and can also help avoid the jittering. Obviously, using a too long temporal window is also bad for model efficiency. Thus we empirically choose $L = 15$ or 30 in our model configuration.

3.7.5 Application: Clothing Texture Editing

In this section, we demonstrate editing for the clothing texture. On top of our photorealistic animation output, we further edit the clothing pattern in four different styles. First, we multiply the RGB channels of the clothing UV texture with different scaling factors to modify the color of the clothing. Second, we apply a checkerboard pattern on our clothing layer. Third, we ask an artist to create a stylistic pattern and then apply it to our clothing animation output. Fourth, we add the ACM SIGGRAPH Logo and text to the front side of the clothing. The results are shown in Figure 3.14. Once the desired pattern is determined, our model can produce animation with the edited texture for any motion sequence similar to those shown in Section 3.7.3.

Compared with the single-layer model, our two-layer structure naturally allows us to easily manipulate the clothing texture in the UV space without interfering with the inner layer in a temporally coherent manner. For comparison, we apply the same blue color transformation to the single-layer output. For this purpose, we manually segment out the clothing region for the first frame in the sequence in the UV space, and apply the color transformation in the segmented region to all the following frames. This approach produces reasonable results for the first frame (shown on the first column of Figure 3.15); for the following frames, however, applying the color transformation in the same UV region will suffer from misalignment of the edited area and actual clothing region, as shown in the right two columns of Figure 3.15. The visual artifact caused by this misalignment is highlighted in the zoomed-in boxes in the figure.

3.8 Implementation Detail

3.8.1 Clothed Body Registration

In this section, we give a detailed description of our clothed body registration pipeline in Section 3.4.

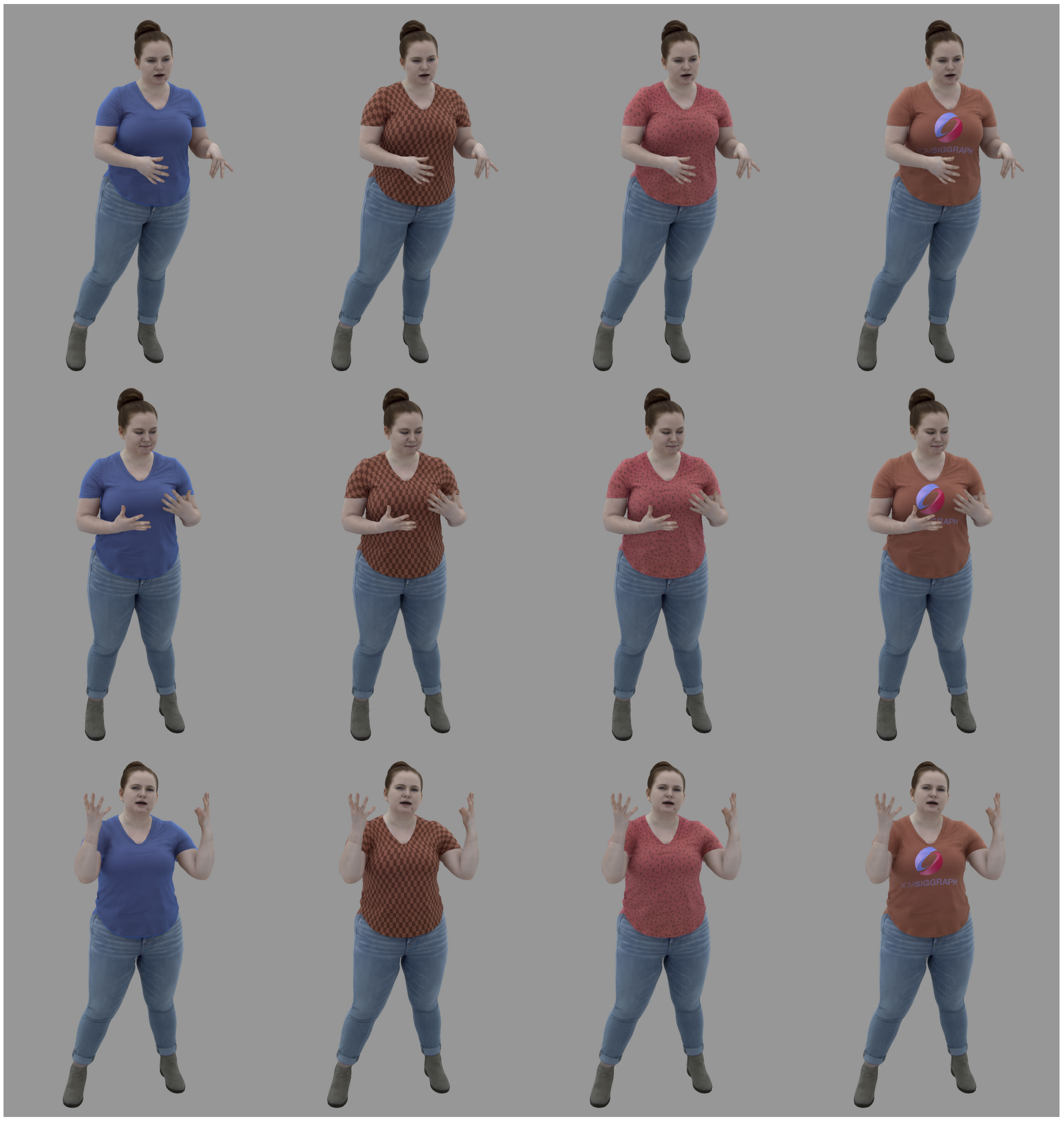

Figure 3.14: Texture editing results of our two-layer codec avatars. From left to right, we show application of color transformation, checkerboard pattern, random artist-created pattern, and an ACM SIGGRAPH logo, respectively, for three different frames.

Data Preprocessing.

The input to our pipeline is a sequence of RGB images of the subject captured by a synchronized multi-camera system. The raw RGB images are used to create a dense 3D reconstruction of the human surface with a multi-view Patchmatch reconstruction algorithm [56]. One example of reconstructed mesh can be seen in Figure 3.3.

Then, we run body part segmentation for the input images. For the network architecture we use PointRend [93], which takes raw RGB images as input and outputs segmentation

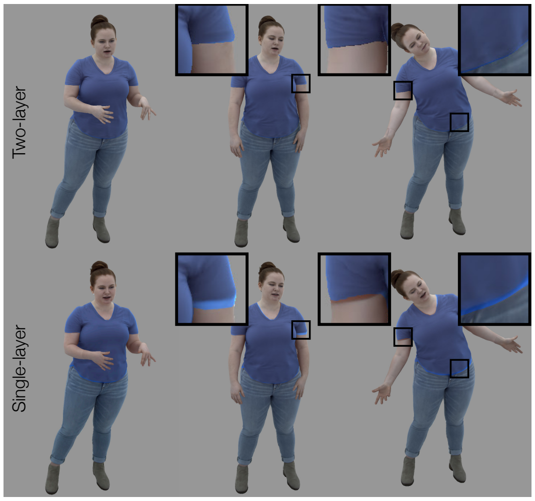

Figure 3.15: Comparison between the single-layer model (bottom row) and the two-layer model (top row) on texture editing in three different frames. The first column shows the frame where we manually segment out the upper clothing region in the UV space for the single-layer model.

masks of the same size. We also run keypoint detection for body and hand joints. We use a multi-stage 2D pose estimation network similar to [191] to first detect 2D keypoints from raw RGB images, then perform multi-view triangulation to obtain 3D keypoints. Both the segmentation and keypoint detection models are trained on our in-house datasets consisting of images captured in the aforementioned multi-camera system. Additionally the 2D pose estimation model is pretrained on the MS COCO dataset [114]. An example of part segmentation and the detected keypoints can be seen in Figure 3.3.

We aim to separately register body and clothing templates to the above observations, including images, scan, segmentation masks and keypoints. First, we run single-layer surface registration to fit to the scan. Then, we segment the single-layer registered meshes into different body and clothing regions based on the multi-view 2D segmentation masks. After that, we register the clothing template to the segmented surface from the single-layer registration with an explicit boundary-aware loss. Finally, we estimate the underlying body shape in the inner layer.

Single-Layer Surface Tracking.

We non-rigidly register the reconstructed meshes with a kinematic body model, similar to [242] and [201]. We use a kinematic body model with $N_j = 159$ joints, $N_v = 614,118$ vertices and pre-defined LBS skinning weights for all the vertices. Let $\mathcal{W}(\cdot,\cdot)$ be the LBS function that takes rest-pose vertices and joint angles as input, and outputs the target-pose vertices. First, we estimate a personalized model by computing the rest-state shape $\mathbf{V}_i \in \mathbb{R}^{N_v \times 3}$ that best fit a collection of manually selected peak poses. Then, for each frame i, we estimate a set of joint angles $\boldsymbol{\theta}_i$, such that the skinned mesh $\hat{\mathbf{V}}_i = \mathcal{W}(\mathbf{V}_i, \boldsymbol{\theta}_i)$ best matches the 3D reconstruction and detected keypoints introduced in the data preprocessing step. Finally, we introduce additional per-frame vertex offsets on top of the skinned kinematic model. We minimize the distance between the deformed surface and the 3D reconstruction with Laplacian regularization. At this point, we have obtained registered meshes representing the human surface as a whole, which provides a prior for solving the two layer registration problem.

Clothing Registration.

Our clothing registration step produces aligned geometry for the outer layer, and is similar in spirit to [155]. We briefly explain the process for completeness and focus on the differences. The process uses the single-layer surface tracking results and clothing part segmentation in all camera views as input. In detail, it consists of the following steps.

Mesh segmentation. Here we aim to label each vertex in the tracked surface mesh as either 'outer' or 'inner'. Because we only model the clothing on the upper body in the outer layer, we label only the vertices in the upper body clothing region as 'outer', and all other regions, including exposed skin and pants, as 'inner'. To identify the vertices belonging to the upper body clothing, we project the mesh to each camera view and check the clothing part segmentation in the projected pixel location. The majority vote of segmentation labels among different camera views gives us the initial vertex segmentation labels. We also filter out vertices visible in less than three camera views and leave them as undetermined, which happens frequently in the region below the armpit. Similar to [155], we then use Markov Random Fields (MRF) to refine the initial vertex segmentation results. This approach allows us to fill in the undetermined region and remove noisy labelling in the initial segmentation results. The output of MRF gives us the binary inner/outer per-vertex segmentation results.

Clothing template creation. We manually select one frame of single-layer tracking results, and use the upper clothing region identified in the mesh segmentation step as our clothing template. We ask an artist to create a UV map for the template, which is used to render the clothing and to represent the clothing geometry in the UV space in the codec avatar (see Figure 3.17). Each vertex in the clothing template is associated with a vertex in the whole-body model $\mathbf{V}_i$, and can be skinned using the same kinematic joints and LBS weights. We also reuse the triangulation in the whole-body model to create a topology for the clothing template. In the following steps, we will use this template to register the clothing shape for all other frames in the sequence.

Deformation initialization. For each frame in the sequence, we use the mesh segmentation results to identify the target region of upper clothing. We also initialize a clothing mesh using the template topology created in the previous step. The goal is to deform the template mesh to match the target clothing mesh. In order to provide better initialization for the deformation, we apply biharmonic deformation fields [77] to find a per-vertex deformation that aligns the boundary of the template mesh to the target mesh boundary while keeping the interior distortion as low as possible. We observe that this allows the template shape to converge to a better minimum in the following iterative optimization step.

Boundary-aware non-rigid ICP. Given the deformation initialization results from the last step, we then run non-rigid Iterative Closest Points (ICP) algorithm to register the template

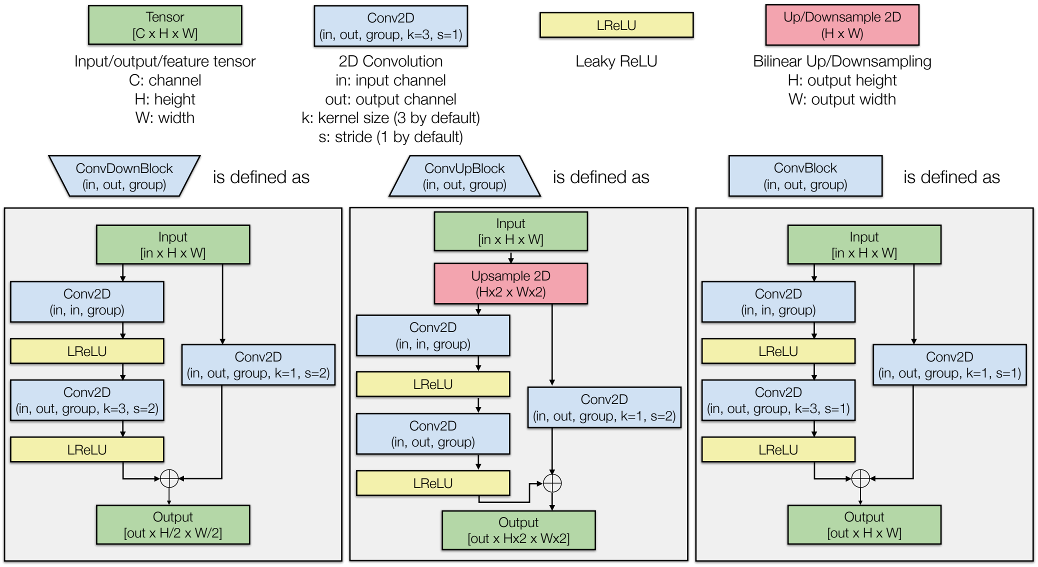

Figure 3.16: Definition of network elements that will be used in the following figures for network architecture. We represent data tensor in green, convolution operations in blue, leaky ReLU in yellow, and up-sampling/down-sampling in red. We also define three types of residual convolution blocks with different up-sampling or down-sampling effects.

mesh to the target clothing mesh, with a special focus on the boundary alignment. Similar to Section 5.3.1 in [155], we optimize a weighted sum of the ICP term, the Laplacian term for mesh regularization and a boundary term that penalizes the distance between the boundary vertices of the template mesh and the target mesh.

After these steps, we obtain a separate mesh registered to the upper clothing of the subject in the same template topology for each frame in the sequence, which we later use to train the clothing branch of our codec avatar.

Per-frame inner-layer body shape estimation. Given the estimated inner-layer template in the previous step, now we individually estimate the inner-layer body shape for every frame in the sequence. For each frame, the estimated inner-layer shape combined with registered upper clothing mesh (Section 3.8.1) should resemble the whole-body surface $\hat{\mathbf{V}}_i$ when observed from outside, and allow us to render the full-body appearance of the person. For this purpose it is important that the estimated inner-layer shape is completely under the upper clothing mesh in the upper body region without intersection between the two layers.

For each frame i, we estimate an inner-layer shape $\mathbf{V}_i^{\text{In}} \in \mathbb{R}^{N_v \times 3}$ in the rest pose. We use the same LBS function $\mathcal{W}(\cdot,\cdot)$ as in Section 3.8.1 to transform $\mathbf{V}_i^{\text{In}}$ into the target pose $\hat{\mathbf{V}}_i^{\text{In}} = \mathcal{W}(\mathbf{V}_i^{\text{In}}, \boldsymbol{\theta}_i)$. We solve the following optimization problem:

$$\min_{\mathbf{V}_i^{\text{In}}} E^I = w_{\text{out}}^I E_{\text{out}}^I + w_{\text{vis}}^I E_{\text{vis}}^I + w_{\text{cpl}}^I E_{\text{cpl}}^I. \tag{3.7}$$

Our two-layer formulation requires that the estimated inner-layer shape stays strictly inside

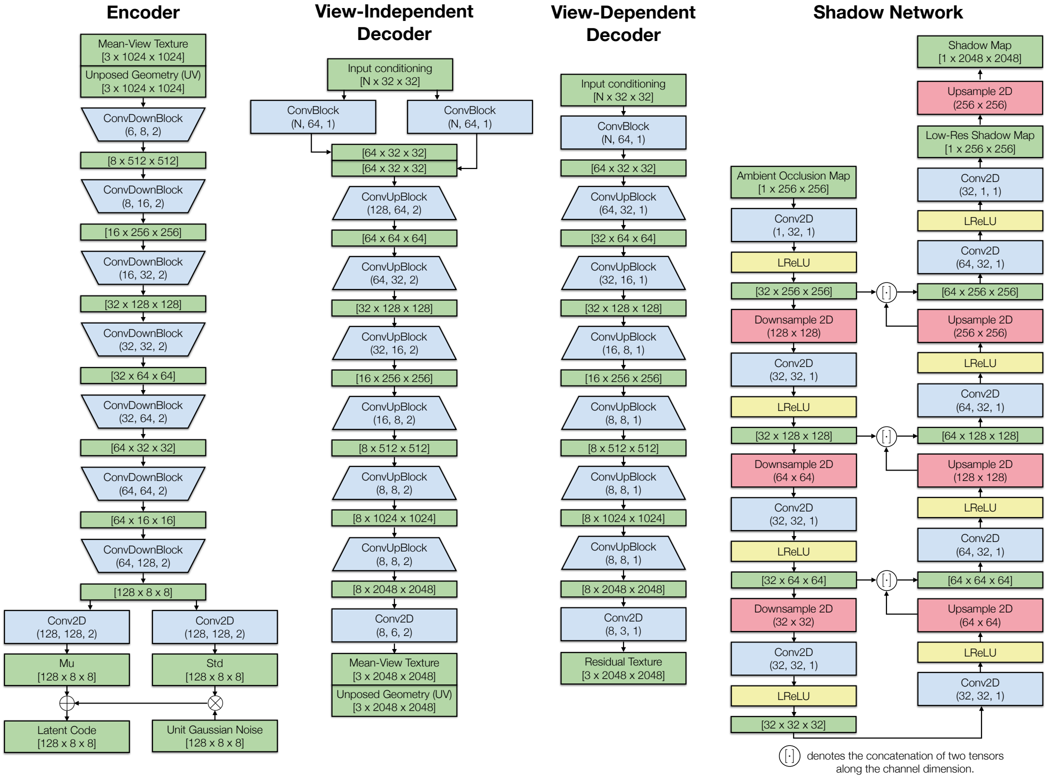

Figure 3.17: Architecture of the encoder, decoder, and shadow network used in the body and clothing networks (Figure 3.18). The encoder takes as input mean-view texture and unposed geometry embedded in the UV map, and outputs a latent code. The decoder consists of the two parts: the view-independent part which outputs the mean-view texture as well as geometry UV map, and the view-dependent part which outputs the per-view residual texture to be added to the mean-view texture. The shadow network has a U-Net [164] architecture. It converts the Ambient Occlusion (AO) map into a shadow map.

the upper clothing. Therefore, we introduce a minimum distance of $\varepsilon = 10$ millimeter that any vertex in the upper clothing should keep away from the inner-layer shape, and use

$$E^{I}_{\text{out}} = \sum_{\mathbf{v}_j \in \hat{\mathbf{V}}_i} s_j \min\{0, d(\mathbf{v}_j, \mathcal{W}(\mathbf{V}^{\text{In}}_i, \boldsymbol{\theta}_i)) + \varepsilon\}^2, \tag{3.8}$$

where, with a slight abuse of notation, s_j denotes the segmentation results for vertex $\mathbf{v}_j$ in the mesh $\hat{\mathbf{V}}_i$, with the value of 1 for a vertex in the upper clothing and 0 otherwise. Similarly, for directly visible regions in the inner-layer we have

$$E^{I}_{\text{vis}} = \sum_{\mathbf{v}_j \in \hat{\mathbf{V}}_i} (1 - s_j) d(\mathbf{v}_j, \mathcal{W}(\mathbf{V}^{\text{In}}_i, \boldsymbol{\theta}_i))^2. \tag{3.9}$$

We also couple the frame-specific rest-pose shape with the cross-frame inner-layer template

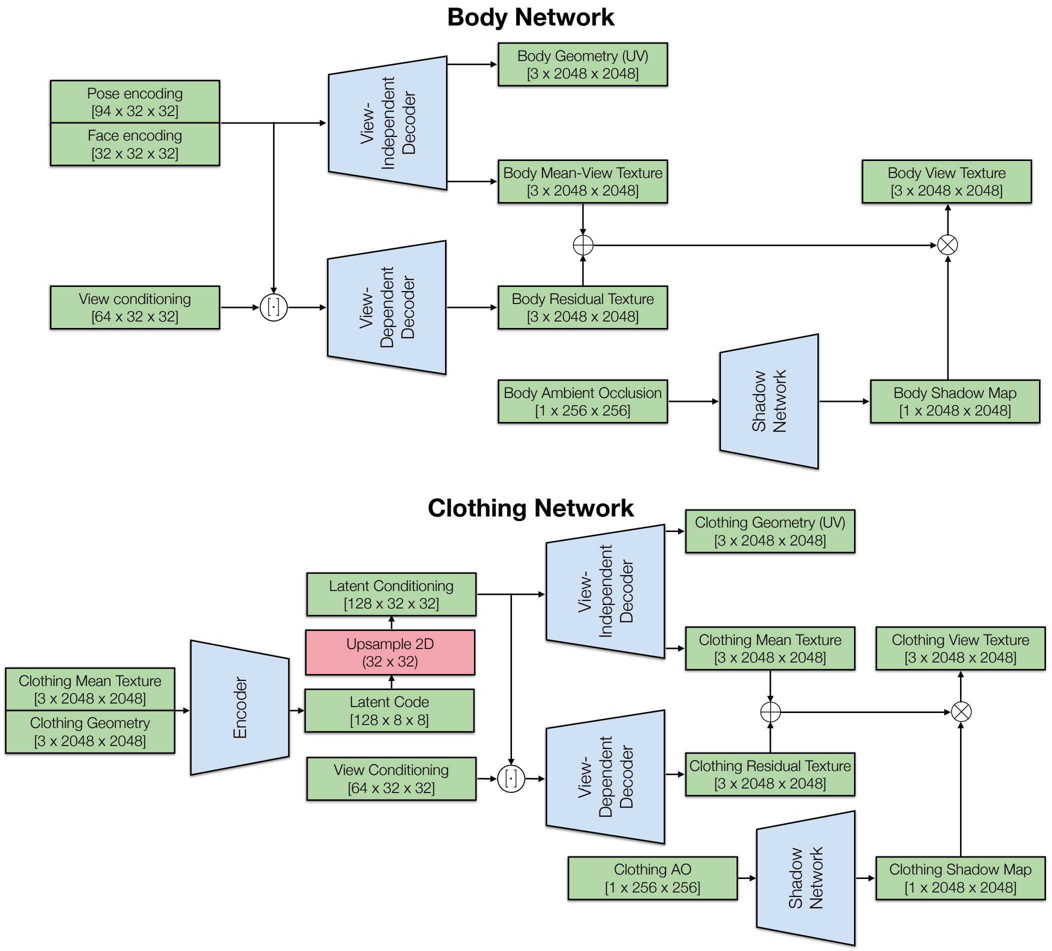

Figure 3.18: The architecture of body and clothing networks. Notice that LBS and inverse LBS are omitted in the figure. All geometry involved is in unposed space, with mean value subtracted. The Ambient Occlusion (AO) for body and clothing is computed from the reconstructed geometry of body and clothing together.

to make use of the strong prior encoded in the template:

$$E^I_{\text{cpl}} = \|\mathbf{V}^{\text{In}}_{i,e} - \mathbf{V}^t_e\|^2, \tag{3.10}$$

where, similar to Equation (5) in [242], the subscript e denotes that the coupling is performed on the edges of the two meshes. In our experiment, we use the following loss weights: $w^I_{\text{out}} = 1.0, w^I_{\text{vis}} = 1.0, w^I_{\text{cpl}} = 500.0$.

Solving Equation (3.7) gives us an estimation of inner-layer shape in a registered topology for each frame in the sequence. The inner-layer meshes and the outer-layer meshes obtained in Section 3.8.1 are both essential for our two-layer codec avatars.

3.8.2 Loss Weights

The loss weights below are provided with any involved length in the unit of millimeter (the Laplacian term in Equation (3.1-3.3) and depth falloff scale in Equation (3.4)).

Section 3.5.1 Body Decoder Equation (3.1):

$$\lambda_g = 0.5, \quad \lambda_{lap} = 50.0, \quad \lambda_t = 5.0.$$

Section 3.5.2 Clothing Network Equation (3.2):

$$\lambda_g = 0.5, \quad \lambda_{lap} = 50.0, \quad \lambda_t = 5.0, \quad \lambda_{kl} = 1.0.$$

We train the above two networks for 40k iterations with a batch size of 8, implemented together in group convolution for computation efficiency.

Section 3.5.3 Inverse Rendering with Two-layer Representation Equation (3.3):

$$\lambda_i = 10.0, \quad \lambda_m = 1000.0, \quad \lambda_v = 0.001, \quad \lambda_{lap} = 100.0.$$

We train the network for 100k iterations with a batch size of 8.

Equation (3.4):

$$c = 10.0.$$

For all the networks above, we use the AdamW optimizer with parameters $\alpha = 1 \times 10^{-3}$ (learning rate), $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1 \times 10^{-8}$, except when we fine-tune the anchor VAE to individual chunks for photometric texture alignment (Section 3.5.4), where we use $\alpha = 1 \times 10^{-4}$.

3.8.3 Network Architecture

Body and Clothing Networks We describe the network architecture for our body and clothing in Figure 3.18, corresponding to the networks in Figure 3.4. The detail of each module, including encoder, decoder and shadow network, is shown in Figure 3.17, and each element in Figure 3.17 is further defined in Figure 3.16.

Geometry. Following previous work [7], the input and output geometry in both networks are defined in the unposed space (converted from the world space by inverse LBS with known joint angles) and with mean value subtracted, represented by a 3-channel UV map.

Texture. We also follow the practice in previous work [7] to separate the final view-conditioned texture into three components: a view-independent texture across different views, an additive residual texture to encode per-view variation, as well as a multiplicative shadow map to encoder long-range shadowing effects. The shadow map is predicted by a shadow network conditioned on an ambient occlusion map that is computed from the reconstructed body and clothing geometry. This separation is only formulated as an inductive bias in the network, and supervision is only applied to the final view-dependent texture.

Input conditioning. As explained in Section 3.5, our body and clothing decoders take in different input conditioning.

- The pose encoding and face encoding are computed by tiling the pose vector and facial keypoint vector respectively along the spatial UV dimension to a 32 × 32 feature map. The feature map is then processed by a spatially localized masking operation and then a 2D residual convolution block. For the detail of this process, see Section 3.3 of [7]).

- The view conditioning is computed as the ray direction from the camera center to the reconstructed geometry in each location of the UV map, with the global orientation factored out. This 3-dim ray direction is then converted to a 64-dim feature independently for each UV location by a fully-connected layer and non-linearity (Leaky ReLU).

- The latent encoding is directly up-sampled from the latent code predicted by the encoder to keep the spatial dimension consistent with other input conditioning.

Temporal Convolution Network for Clothing Animation For the temporal convolution network used in Section 3.6, we use a total of 6 'ConvDownBlock's defined in Figure 3.16, only with the 'Conv2D' operation replaced by 'Conv1D' along the temporal dimension. The input channel dimension is 94 corresponding to the pose vector, and the channel number after each 1D convolution block is 128, 128, 256, 512, 1024, 8192 respectively. The final output is reshaped to match the dimension of the clothing VAE latent code.

3.9 Conclusion

We have proposed a two-layer mesh representation for building an animatable avatar for clothed body. The results demonstrate that the explicit clothing modeling not only improves the rendered clothing quality in animation, but also enables the editability of the clothing texture, opening up new possibilities for codec avatars. The two-layer avatar models cannot be obtained without the success of two-layer registration of the clothed body. We thus have presented a new clothed body registration method along with a texture alignment method to improve the photometric correspondences using inverse rendering.

Chapter 4

Deep Photorealistic Appearance for Simulated Clothing

4.1 Introduction

Existing work on avatars with animatable clothing can be categorized into two main streams. Cloth simulation creates realistic clothing deformations with dynamics [19,23,125,130,136], but only focuses on modeling geometry. The other line of the work leverages real-world captures to build neural representations of clothing geometry [10] and may include appearance (see [69,115] and Chapter 3 [218]). However, these systems usually damp the clothing dynamics, struggle at generalizing to unseen poses and cannot handle collisions well. Our key insight is that these two lines of work are actually complementary to each other, and combining them can help achieve the best of both worlds.

In this work, we propose to integrate physics-based cloth simulation into avatar modeling, so that the clothing on the avatar can be animated photorealistically with the body, while achieving high-quality dynamics, collision handling and the capability to animate and render avatars with novel clothing. Our work builds upon full-body Codec Avatars (see [7] and Chapter 3 [218]), which leverage a Variational Autoencoder (VAE) to model the geometry and appearance of a human body. In particular, we follow the multi-layer formulation of Chapter 3, but redesign the clothing layer to integrate a physically-based simulator. Namely, at the training stage, we learn the clothing appearance model using real-world data, by processing raw captures with our dynamic clothing registration pipeline. At test time, we simulate the clothing geometry on top of the underlying body model with appropriate material parameters, and then apply the learned appearance model to synthesize the final output.

Unfortunately, there are two major issues with a naive implementation of this pipeline. First, there exists a gap between the simulator output and the tracking obtained from the real data. Estimating the full set of physical parameters for body and clothing to faithfully reproduce the clothed body configuration remains an unsolved problem, despite some progress in controlled settings [133] or in estimating only the body parameters [65]. There are inevitable differences between the test-time simulation output with manually selected parameters and the real-world clothing geometry used for training. Second, tracking clothing and underlying body geometry at high accuracy is still a challenging problem, espe-

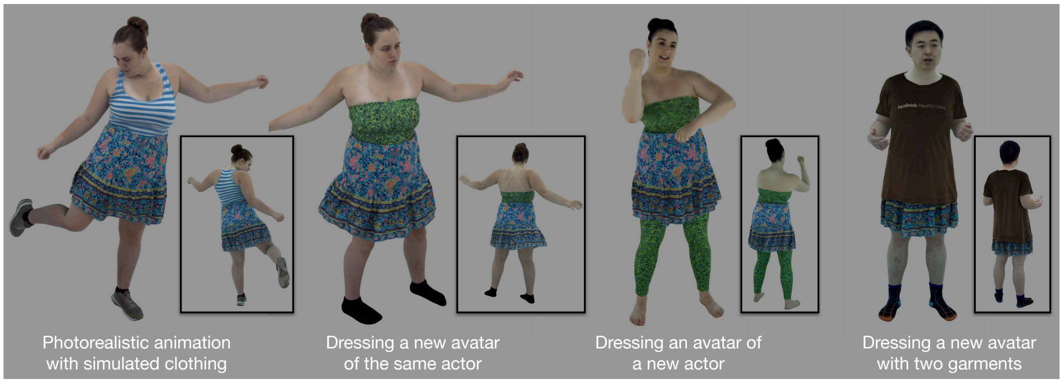

Figure 4.1: We develop pose-driven full-body avatars with photorealistic clothing by applying neural rendering to physically simulated garments. On the left, we show a skirt animation together with the body avatar built from the same captured sequence. We further retarget the skirt to a novel sequence with the same actor and two new actors. On the right, we animate the skirt and a T-shirt together.

cially for loose clothing such as skirts and dresses. Both of these issues, inconsistency between training and test scenarios and unreliable tracking, make learning a generalizable appearance model more challenging. Thus, a good design of the appearance model should avoid learning chance correlations between degenerated tracked cloth geometry and specific appearance. To this end, we design the model to be localized in terms of both architecture (U-Net [164]) and input representation (normals). We also take inspiration from physically-based rendering and decompose appearance into local diffuse components, view-dependent and global illumination effects such as shadowing[1]. In particular, we rely on an unsupervised shadow network conditioned on the ambient occlusion map explicitly computed from the body and clothing geometry, so that the dynamic shadowing can be effectively modeled even for a different underlying body model at test time.

Our approach generates physically realistic dynamics and photorealistic appearance that are robust to diverse body motion with complex body-clothing interactions. In addition, our formulation allows the transfer of clothing between different individuals' body avatars as shown in Figure 4.1, as well as virtual garment resizing in Figure 4.9. Our method opens up the possibility to dress photorealistic avatars with novel garments. Our contributions are as follows:

- We present animatable clothed human avatars with data-driven photorealistic appearance and physically realistic clothing dynamics from simulation;
- We develop a deep clothing appearance model to produce photorealistic clothing appearance that bridges the generalization gap between the tracked clothing geometry for training and the simulated clothing geometry at test time;

[1] Global illumination is a broad term that is used to refer both to rendering algorithms as well as to the way that light is reflected around the scene, perhaps through multiple bounces before reaching the camera. In this thesis, although we do not explicitly model indirect illumination, we still adopt this term for the approximate shadowing effect based on ambient occlusion. We consider ambient occlusion as a global illumination technique both because it is determined by the other geometry in the scene, and because it is a scalar coefficient for ambient lighting for a very crude approximation of indirect illumination [83].

- Our animation system further enables transferring clothing between different subjects as well as editing of garment size.

In our experiments, we evaluate the effectiveness of our approach by animating multiple different identities and clothing types, and provide a comprehensive qualitative and quantitative comparison to existing techniques.

This work is published in ACM Transaction on Graphics (SIGGRAPH Asia) 2022.

4.2 Related Work

Aside from the existing literature on clothing animation already reviewed in Section 2.2, here we discuss a particularly relevant work [245]. Zhang and colleagues [245] propose a neural rendering approach that produces output images from coarse garments generated by a temporal model. As an advantage, this method can animate a garment by simulating their simplified versions and then adding detailed structures and appearance by neural rendering. However, this work has several limitations. First, this work shows results mostly in the synthetic domain. The output rendering is detailed but limited in photorealism, because generating highly photorealistic training images for the neural renderer is non-trivial. Second, the method assumes input pairs of body-only images as background and clothed body images as ground truth. In the real-world setting, it is not clear how to obtain such paired data, especially the body-only images. Third, the screen-space garment renderer can have difficulty inferring the depth order between the body in the background layer and the clothing in the foreground layer. By contrast, our method builds photorealistic avatars with highly dynamic clothing using captured data in the real world. We track body and clothing geometry and photometric correspondences at high quality in the captured data, which facilitates the effective learning of the appearance function. Our physically inspired clothing appearance model operates in 3D space, and thus is free from the depth ordering issue even with complex body-clothing interactions.

4.3 Method Overview

Our goal is to build pose-driven full-body avatars with dynamic clothing and photorealistic appearance. Following [7] and Chapter 3 [218], we train our avatars on multi-view capture sequences of each subject wearing the clothing of interest. At test time, the avatars are animated by a sparse driving signal of skeleton motion (including facial keypoints if available), and can be rendered in a novel camera viewpoint.

We aim to achieve high-fidelity animation that looks realistic both in terms of appearance of the human subject and temporal clothing dynamics. For this purpose, we develop an animation pipeline consisting of three modules: an underlying body avatar model, physics-based cloth simulation and a clothing appearance model. The underlying body avatar takes as input the skeleton pose (including face conditioning if animating expression) and outputs the body geometry. Given a sequence of body geometry, we then use cloth simulation [184] to generate clothing geometry with natural and rich dynamics, physically consistent with the motion of the underlying body. Finally, we apply the clothing appearance model to the simulated geometry and generate photorealistic texture, which takes into account not only the clothing geometry but also shadows caused by the occlusion of the avatar

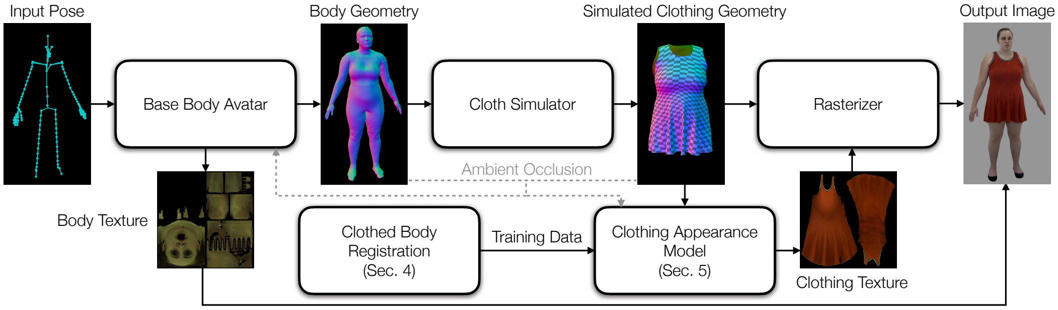

Figure 4.2: Our animation pipeline includes three major modules: the base body avatar that predicts body geometry and texture given pose as input, the cloth simulator that generates clothing deformation on top of the body geometry, and the clothing appearance model that predicts photorealistic clothing texture. The appearance model is trained using real captured data with registered body and clothing geometry. The body avatar and clothing appearance model also takes in ambient occlusion between the body and clothing geometry for dynamic shadowing effects. The geometry and texture pairs are then rasterized together to produce the final output image.

body. The shadow cast by the clothing on the body is also modeled similarly in the underlying body avatar. Figure 4.2 illustrates our overall pipeline.

Photorealistic full-body Codec Avatars [7,218] and physics-based cloth simulation [184] have been extensively explored in the existing literature. For these two modules we mostly follow previous work and provide implementation detail in the Section 4.7.1[2]. We find efficient photorealistic appearance modeling of clothing to be a critical missing component in such a pipeline, and we tackle two major technical challenges in order to build such a system. On the one hand, as explained in Section 4.5, we develop a deep clothing appearance model that can efficiently generate highly photorealistic clothing texture with dynamic view-dependent and shadowing effects. For the design of the model, we focus on the generalization of the produced appearance, because the input geometry from the cloth simulator can be different from the tracked clothing geometry used to train the appearance model. On the other hand, in order to generate training data for the appearance model, we extend the previous clothing registration algorithm [155, 218] to handle highly dynamic clothing types including a skirt and a dress. In addition, we track photometric correspondences on the garments by matching salient features, which are essential for the modeling of highly textured clothing appearance. The clothing registration step is described in Section 4.4.

4.4 Clothed Body Registration

In this section, we introduce our data processing pipeline to obtain training data for the clothing appearance model described in Section 4.5. Because clothing registration is not the core contribution of this work, we focus on challenges posed by large clothing dynamics and rich texture, which must be addressed to achieve high-fidelity animation.

[2] Unless otherwise stated, we use a GPU-based XPBD [130] cloth simulation of a mass-spring system for its superior runtime performance, but our method is not restricted to a specific choice of model or integration technique. Examples generated by a different simulator [125] are provided in the Section 4.6.6.

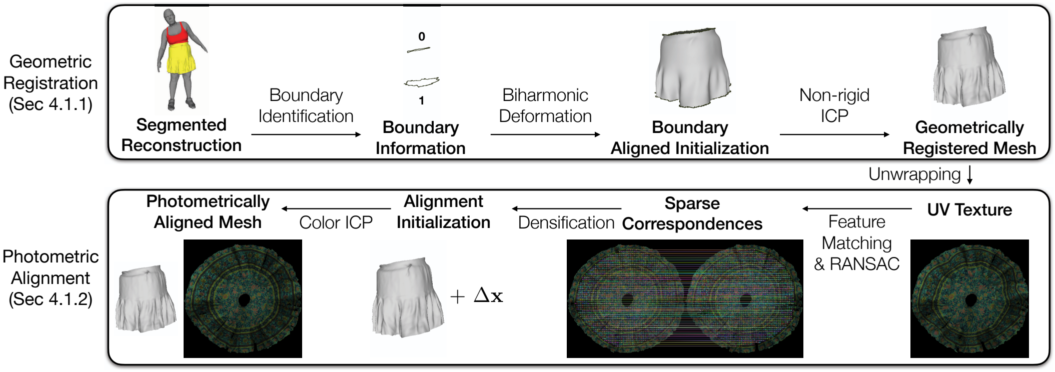

Figure 4.3: Our full clothing registration method consists of two major steps: geometric registration and photometric alignment. The geometric registration step aligns the garment surface of the segmented reconstruction with a template mesh. We rely on the boundary information for initialization and then minimize the surface error using non-rigid ICP. In the photometric alignment step, we establish sparse correspondences between the salient region of the unwrapped texture and a template texture. The sparse correspondences are then densified as a displacement for each vertex. This step is followed by color ICP which outputs both a geometrically and photometrically aligned mesh in the template topology.

Our data capture setup is similar to Chapter 3. The pipeline takes multi-view image sequences of a subject as input, and outputs registered meshes of the garment and the underlying body in two separate layers. We follow Chapter 3 for raw geometry reconstruction and mesh segmentation into body and clothing regions. We similarly estimate the kinematic body pose and inner-layer body surface.

4.4.1 Dynamic Clothing Registration

The goal of clothing registration is to represent the clothing geometry in a single mesh topology with consistent correspondences. Our clothing registration method consists of two major steps, geometric registration and photometric alignment. Figure 4.3 illustrates the overview of the clothing registration method.

Geometric Registration

In this step, we fit a clothing template to the segmented clothing region of the reconstructed mesh using the non-rigid Iterative Closest Point (ICP) algorithm. The non-rigid ICP algorithm is similar to previous work [155, 218] and thus we omit the details here.

In order to track loose and dynamic clothing types (e.g. skirt and dress), it is important to provide good initialization for non-rigid ICP. We observe that the clothing boundaries provide useful information about the overall orientation and deformation of the garment. Therefore, we start by estimating coarse boundary correspondences. Utilizing the fact that each garment has a fixed number of boundaries (for example, two for the skirt and four for the dress), we associate each point on the the target clothing boundary with the template mesh boundary by querying the nearest vertex in the tracked inner-body surface. Given this coarse boundary correspondence, we use Biharmonic Deformation Fields [77] to solve for per-vertex deformations that satisfy the boundary alignment constraints while minimizing the interior distortion. We use the output of the Biharmonic Deformation Fields as the initialization for the non-rigid ICP algorithm.

Photometric Alignment

The geometric registration method in [155,218] aligns the clothing geometry with a single template topology by minimizing the surface distance, but does not explicitly solve for interior correspondences. In order to effectively model clothing appearance, it is necessary to make sure that each vertex in the template consistently tracks the same color (essentially reflectance), which we call photometric correspondences in this thesis. We observe that the chunk-based inverse rendering algorithm in Chapter 3 can correct small deviations of the photometric correspondences in the geometric registration step but cannot recover from large errors in the initialization.

Here, we explicitly solve for photometric correspondences by matching salient features in highly textured regions of the garments. For each frame in the sequence, we first unwrap a mean texture from multi-view images to the UV space using the geometrically aligned mesh from the previous step. Then, following [18], we use DeepMatching [162] to establish sparse correspondence pairs between the unwrapped texture and the template texture. We also use RANSAC to prune erroneous correspondences. The sparse correspondences are then densified to each vertex by solving a Laplace's equation, similar to Biharmonic Deformation Fields in the previous step. Finally, we run color ICP [144] (our own implementation) between the template and the target mesh to photometrically align all the vertices.

4.5 Deep Dynamic Clothing Appearance Model

In this section, we introduce our clothing appearance model which is the key technical component that enables our photorealistic clothing animation system. Given clothing geometry as input, the goal of the clothing appearance model is to generate clothing texture that can be used for rasterization together with the input geometry to produce photorealistic appearance.

We build a data-driven clothing appearance model in order to learn complex photorealistic appearance from real captured image sequences. Several factors need to be taken into account when designing such a model. First, the appearance model is trained on tracked geometry from the previous registration step, but at test time, it takes simulated clothing geometry as input. Therefore, it is essential for the model to bridge the generalization gap between the training and testing data. Second, the generated texture should include various aspects of photorealistic appearance, such as view-dependent effects and dynamic shadowing. Third, for the sake of efficiency, the model should only involve basic quantities that can be easily derived from the clothing geometry, without computationally expensive operations such as multi-bounce Monte-Carlo ray tracing.

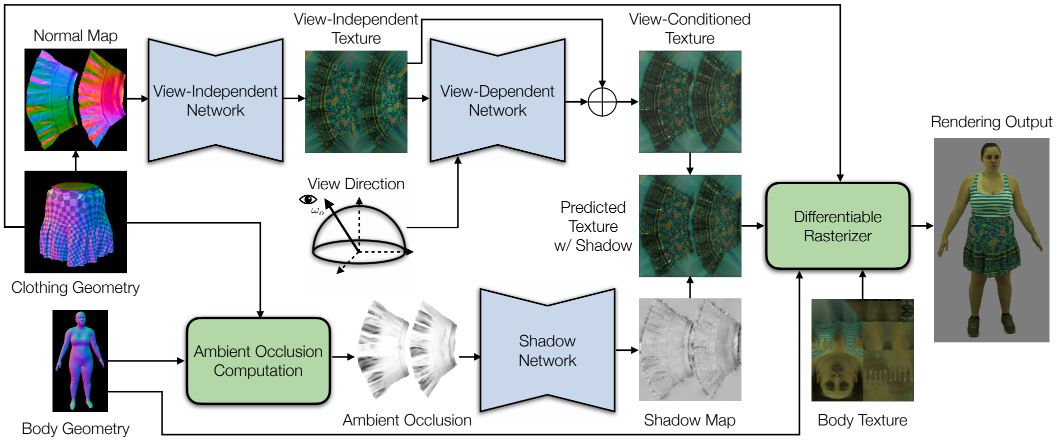

Figure 4.4: The architecture of our deep clothing appearance model. We model local diffuse appearance with a view-independent network conditioned on surface normals. The view-dependent network additionally takes in view direction and produces a view-conditioned offset. We explicitly model shadowing using a shadow network that predicts a multiplicative shadow map given ambient occlusion between the body and clothing as input.

4.5.1 Background: Rendering Equation

For a particular point $\mathbf{x}$ in the scene, the classical rendering equation [86] is written as

$$L_o(\mathbf{x}, \omega_o) = \int_{\Omega(\mathbf{n})} f(\mathbf{x}, \omega_i, \omega_o) L_i(\mathbf{x}, \omega_i)(\omega_i \cdot \mathbf{n}) d\omega_i, \tag{4.1}$$

where ω_i, ω_o denotes the incident and outgoing direction, L_i, L_o denotes the incident and output radiance, f denotes the BRDF function, and Ω denotes the hemisphere where the integral is computed, determined by the normal direction $\mathbf{n}$. Here the emission term is omitted because clothing usually does not emit radiance.

A common strategy in physics-based rendering is to decompose the appearance into a diffuse component and a specular component. For the diffuse component, the BRDF function f is assumed to be a constant per point. Then the diffuse radiance can be written as

$$L_d(\mathbf{x}) = f_d(\mathbf{x}) \int_{\Omega(\mathbf{n})} L_i(\omega_i)(\omega_i \cdot \mathbf{n}) d\omega_i. \tag{4.2}$$

When assuming distant lighting and ignoring cast shadows, L_i can be considered as an environment lighting map which represents the consistent illumination condition for the whole captured sequence. Under this circumstance, the integral term is only determined by the normal direction $\mathbf{n}$ at each point, and thus

$$L_d(\mathbf{x}) = f_d(\mathbf{x}) E(\mathbf{n}), \quad E(\mathbf{n}) = \int_{\Omega(\mathbf{n})} L_i(\omega_i)(\omega_i \cdot \mathbf{n}) d\omega_i. \tag{4.3}$$

Besides the local diffuse component, we need to additionally account for specular (view-dependent) effects and global illumination effects especially cast shadows. The decomposition of appearance into these three components has direct implication for the design of network architecture described in the following section.

4.5.2 Model Formulation

From our capture setup we obtain a sequence of registered clothing meshes $\{V_i\}$ and underlying body meshes $\{B_i\}$, paired with multi-view images I_i^c, where i denotes the frame number and c denotes the camera ID. We learn an appearance function F from geometry V_i, B_i and viewpoint v^c to a photorealistic UV texture. Figure 4.4 shows our clothing appearance model.

From the analysis in the previous section, the function F should be able to model local diffuse (view-independent) appearance, view-dependent effects and cast shadows. Under the assumption in Equation (4.3), the diffuse component at each location is determined by the normal direction and diffuse albedo. Therefore, the view-independent part of F is conditioned on the normal direction $\mathbf{n}$ to encode the illumination direction. We use 2D convolution with untied biases to encode the spatially varying reflectance over the clothing. In our experiments, we show that the normal direction $\mathbf{n}$, as a local geometric property, is more effective for generalization when used as input conditioning than the absolute vertex positions V_i.

The view-dependent part of the network additionally takes the viewpoint information v_c as input, and predicts an additive view-dependent appearance offset. The viewpoint is represented by the viewing direction vector in the local Tangent-Binormal-Normal (TBN) coordinate at each position on the mesh.

For dynamic shadowing effects, previous work [7,218] demonstrates that shadow maps predicted by a shadow network from ambient occlusion can achieve a good tradeoff between output quality and computation efficiency. Therefore, we follow previous work to use a shadow network to predict a quasi-shadow map, which is multiplied with the view-dependent texture to obtain the final predicted texture. The input ambient occlusion is computed by ray-mesh intersection with both body and clothing geometry.

We use a 2D convolutional neural network (CNN) for the architecture of the appearance model F. All the inputs to the network are converted to 2D feature maps according to a fixed UV mapping. Instead of the autoencoder structure of Codec Avatars [7,218], we adopt the U-Net architecture [164] in each of the three parts. This architecture can focus on local information around each input location to learn a generalizable appearance function.

Loss function. The appearance model produces a view-conditioned photorealistic texture for the clothing given the input clothing and underlying body geometry (used for ambient occlusion computation), denoted by $F(V_i, B_i, v_c)$. A differentiable rasterizer R is then used to render body and clothing, denoted by $R(V_i, F(V_i, B_i, v_c))$. We penalize the difference between the output rendering and the raw captured image in the clothing region with the following differentiable rendering loss

$$l = \sum_{i,c} \|(R(V_i, F(V_i, B_i, v_c)) - I_i^c) \odot M_i^c\|_1, \tag{4.4}$$

where M_i^c denotes the mask of clothing region in I_i^c obtained from image segmentation, and $\odot$ denotes element-wise multiplication. We also rasterize the underlying tracked body geometry with a mean unwrapped body texture together with the clothing for correct body-clothing occlusion, which is omitted in Equation (4.4) for simplicity.

4.6 Results

In this section we present experimental results. We first introduce the data capture setup in Section 4.6.1. Then we evaluate the clothing appearance model presented in Section 4.6.2, followed by full pose-driven animation results including body and clothing in Section 4.6.3. In Section 4.6.4, we show an application of dressing photorealistic avatars, i.e. the animation of clothing on top of novel subjects. Finally, in Section 4.6.5, we present a runtime analysis of our method.

4.6.1 Data Capture Setup

We capture and process a total of four garments: two T-shirts worn by two different male subjects, and a skirt and a dress worn by a female subject. The four garments span different types of texture patterns, including uniform color, a logo, and tiled floral texture, and thus provide a good testbed for our deep clothing appearance model. Following Chapter 3, we treat other clothing worn along with the four garments of interest in the body layer, including pants paired with the T-shirts and the tank top paired with the skirt. These garments move closely together with the body, and can be handled well in the same layer.

For the purpose of dressing avatars, we additionally capture three base body avatars with minimal clothing (tight outfits with greenish color in Figure 4.8), including the subject from the skirt and dress capture and two new subjects. In Section 4.6.4, we generate synthetic clothing animation on these three base body avatars.

4.6.2 Evaluation of the Clothing Appearance Model

In this section, we evaluate the deep clothing appearance model, including ablation studies and comparisons with previous works. We leave out a segment of the captured skirt sequence as the test set, and apply the clothing appearance model on the tracked clothing geometry. In this way, we can directly compute the difference between the rendered clothing output and the original captured images as ground truth. We report the error in the clothing area of the images indicated by the part segmentation masks.

Ablation studies: network components

In the first group of experiments in Table 4.1, we validate the effectiveness of different modules in our clothing appearance model. The simplest version of the appearance model is to directly rasterize a single mean unwrapped texture for all the frames. This extremely cheap rendering method can retain the pattern on the skirt, but no dynamic appearance effects from illumination and shadowing are modeled, which is exemplified by the unnatural baked-in shadows in Row (A) of Figure 4.5. We further add view-independent, view-dependent and shadow networks, which correspond to the three dynamic effects we attempt to model. In Row (B) of Figure 4.5, we compare the full method with the network without view-dependent effects, which corresponds to the "view-independent + shadow" case in Table 4.1. The most obvious difference is the brightness pixel intensity near the silhouette of the clothing in the full method, due to Fresnel reflection when the incident angle is close to $90°$. The results in Table 4.1 verify that the full model with all three modules performs the best.

Method	MSE↓	SSIM↑
Network components: mean texture	316.48	0.67
Network components: mean texture + shadow	260.44	0.68
Network components: view-independent + shadow	189.29	0.73
Network components: view-independent + view-dependent	177.27	0.75
Our full model (network components: view-independent + view-dependent + shadow)	**155.31**	**0.76**
Geometry input conditioning: raw vertices	206.10	0.72
Geometry input conditioning: unposed normal	160.38	**0.76**
Geometry input conditioning: unposed vertices	161.08	**0.76**
Our full model (geometry input conditioning: normal)	**155.31**	**0.76**
W/o photometric alignment for training data (test on data w/o photometric alignment)	597.71	0.44
W/o photometric alignment for training data (test on data w/ photometric alignment)	483.73	0.46
Our full model (w/ photometric alignment for training data)	**155.31**	**0.76**
Previous method: Dynamic Neural Garments [245]	326.58	0.57
Chapter 3 [218]: Clothing Codec Avatars (texture only)	261.28	0.64
Chapter 3 [218]: Clothing Codec Avatars (geometry + texture)	379.87	0.58
Our full model	**155.31**	**0.76**

Table 4.1: Quantitative evaluation of the clothing appearance model applied on tracked clothing geometry. Mean Squared Error (MSE, the lower the better) and Structural Similarity Index Measure (SSIM, the higher the better) are reported. We conduct ablation studies on the effectiveness of each network component, type of input conditioning, and photometric alignment for the training data. We also compare with similar modules in the previous work.

Ablation studies: input conditioning

In Section 4.5.2, our appearance model is conditioned on surface normals as geometry input. In the second group of experiments in Table 4.1, we compare this formulation with the one from [218] (Chapter 3), where the network takes the clothing vertex location after inverse Linear Blend Skinning (LBS) as input, also called "unposed vertices". We additionally consider the other possibilities of using "raw vertices" (without inverse LBS) and the surface normals of unposed vertices ("unposed normal"). The quantitative results demonstrate the superiority of our proposed formulation. We believe there are two reasons for this performance improvement. First, the surface normals are a localized representation, and thus allow better generalization of the model than the absolute positions of vertices. Second, given the analysis in Section 4.5.1, we know that the surface normals encode the direction of incident illumination, thus following the principles of physics-based rendering.

Ablation studies: photometric alignment

In the third group of experiments, we verify the importance of photometric alignment for highly textured clothing. As a comparison, we also train an appearance model using clothing meshes generated by geometric registration without photometric alignment. Both the quantitative results in Table 4.1 and the qualitative evaluation in Row (C) of Figure 4.5

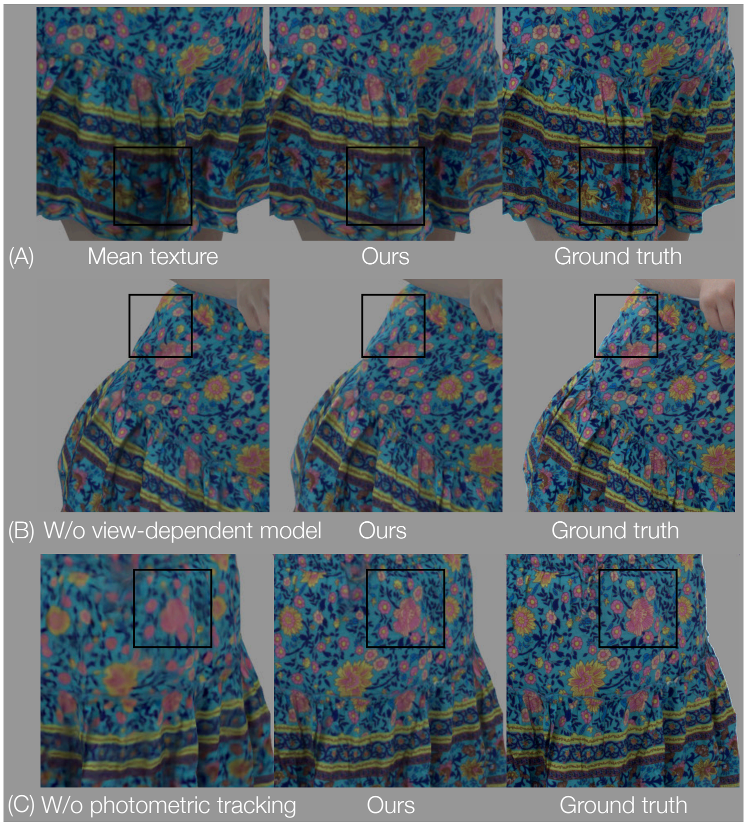

Figure 4.5: Illustration for ablation studies of the deep clothing appearance model. We compare our full model with representative baselines. Regions of particular interest are marked with black squares. (A) The mean-texture rendering has baked-in shadows. (B) Without the view-dependent component, brightness near the sihouette caused by Frensnel reflection cannot be modeled. (C) The model trained on data without photometric alignment produces blurry results.

show significant degradation when photometric alignment is removed. The output images are very blurred when the model is trained on such data. Different positions in the textured regions of the clothing have a different reflectance (BRDF), so the tracked photometric correspondences must be consistently aligned with the captured images in order to learn the correct appearance function.

Comparison with previous methods

We compare the clothing appearance model with similar modules from two previous methods: Clothing Codec Avatars (Chapter 3 [218]) and Dynamic Neural Garments [245]. For these experiments, we use the same training data as our full model and only test the influence of model choice.

Clothing Codec Avatars (Chapter 3 [218]) are deep neural networks with an autoencoder architecture. The encoder takes unposed clothing geometry as input and outputs a

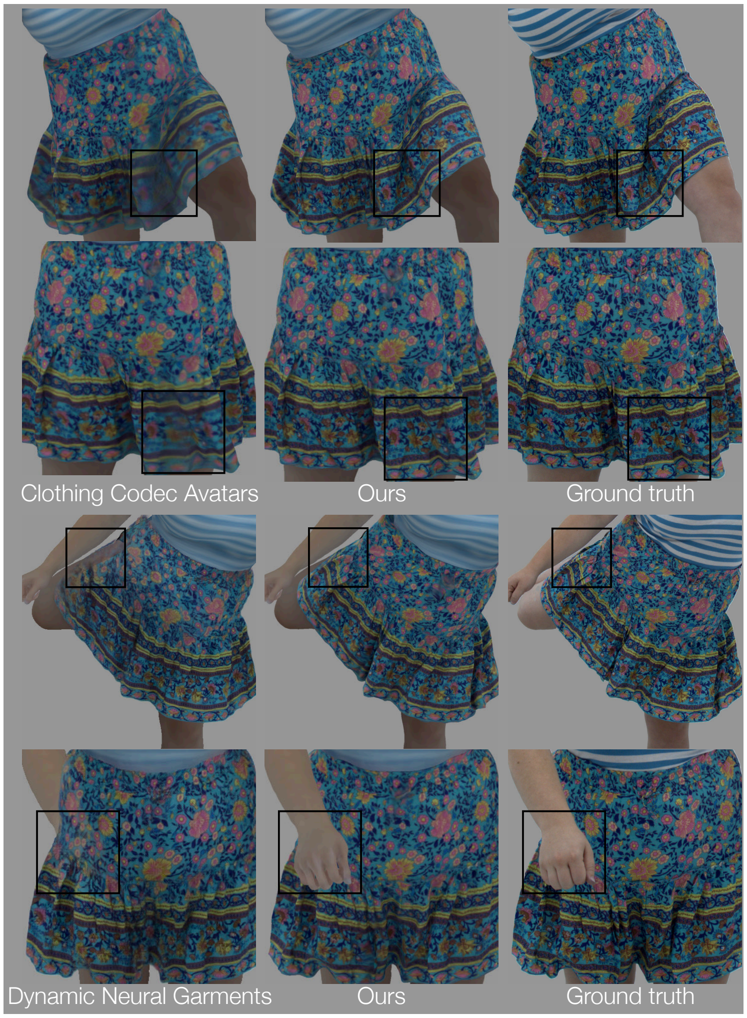

Figure 4.6: Comparison between our clothing appearance model and similar modules in previous methods. The Clothing Codec Avatars (Chapter 3 [218]) have difficulties modeling the highly dynamic region of the skirt. A major issue with the screen-space garment renderer in Dynamic Neural Garments [245] is the incorrect depth ordering between body in the background layer and clothing in the foreground layer.

latent code which is fed to the decoder to generate both geometry and photorealistic view-dependent texture, so the network can be used as clothing appearance model as well[3]. We test the model in two ways: (1) rendering tracked geometry (identical to input) with decoded texture (2) rendering decoded geometry with decoded texture. As shown in the last group of results in Table 4.1 and in Figure 4.6, our deep appearance model achieves the lowest rendering errors, which we attribute to three factors. First, the bottleneck structure

[3] In the original work, the encoder takes both geometry and texture as input. We adapt the encoder input for this experiment. The Clothing Codec Avatars have the same shadow network that takes ambient occlusion as input.

of Clothing Codec Avatars is essential for controlling the avatar with a low-dimensional latent code but disadvantageous for learning detailed appearance compared with the U-Net architecture of our appearance model. This analysis is further supported by comparing Clothing Codec Avatars with the appearance model with the same input conditioning ("unposed vertices") but a U-Net architecture among the second group of experiments in Table 4.1. Second, Clothing Codec Avatars are trained using a differentiable rendering loss on decoded geometry and texture. However, the high frequency dynamics of the skirt itself is hard for the network to learn, further jeopardizing the modeling of texture, manifested by the blurry regions at the bottom of the skirt in Figure 4.6. Third, in the ablation studies, we verified that the input conditioning of unposed vertices is not as good as the surface normal used by our appearance model. All these factors demonstrate that for our goal, it is advantageous to adopt the novel formulation of the appearance model instead of that of Clothing Codec Avatars.

Dynamic Neural Garments (DNG) [245] models garment appearance with an improved version of Deferred Neural Rendering [195] that is more temporally coherent. The code for the garment renderer is released (not including the garment geometry prediction part), so we train the renderer using the same data as our appearance model for comparison. We generate the background images by rasterizing tracked body geometry with the fixed mean body texture, and composite it with the clothing region of the original image as ground truth. The results are shown in Table 4.1 and Figure 4.6. As a screen-space method, the garment renderer in DNG has difficulty reasoning about the depth order between the body and clothing layer when there are complex interactions between the arms or legs and the garment. This limitation is acknowledged in the original paper [245] and contributes to the high quantitative error in Table 4.1 when skin color appears in the clothing region of ground truth images.

4.6.3 Pose-Driven Animation

In this section, we present pose-driven animation results for the four garments on top of the original body avatars that are captured together with the clothing. These results are generated by the complete animation pipeline shown in Figure 4.2. We make modest efforts to select the physical parameters for the simulation to roughly resemble the material behavior in the training sequence.

We compare the results with the output of the full method of Clothing Codec Avatars (Chapter 3 [218]). Some examples are shown in Figure 4.7 and full animation results are shown in the supplementary video[4]. The most obvious advantage of our method is the temporal dynamics, especially for the loose clothing that does not exactly follow the body motion. Although clothing is modeled explicitly in a separate layer in Chapter 3 [218], the mapping from body motion to clothing dynamics is a highly complex, temporally dependent function that is not easy to learn well. Our approach further separates the problem into dynamics and appearance to best capture both these aspects.

4.6.4 Application: Dressing Avatars

Our formulation naturally enables garment animations on top of base body avatars that are different from the original captured sequence. Because we model the clothing separately from the body layer, we simply replace the base body avatar and apply the physics-based

[4] https://drive.google.com/file/d/10gmbpQ0rqYPSEZ1oFq5fSp_n5Dqmhjr8/view?usp=share_link

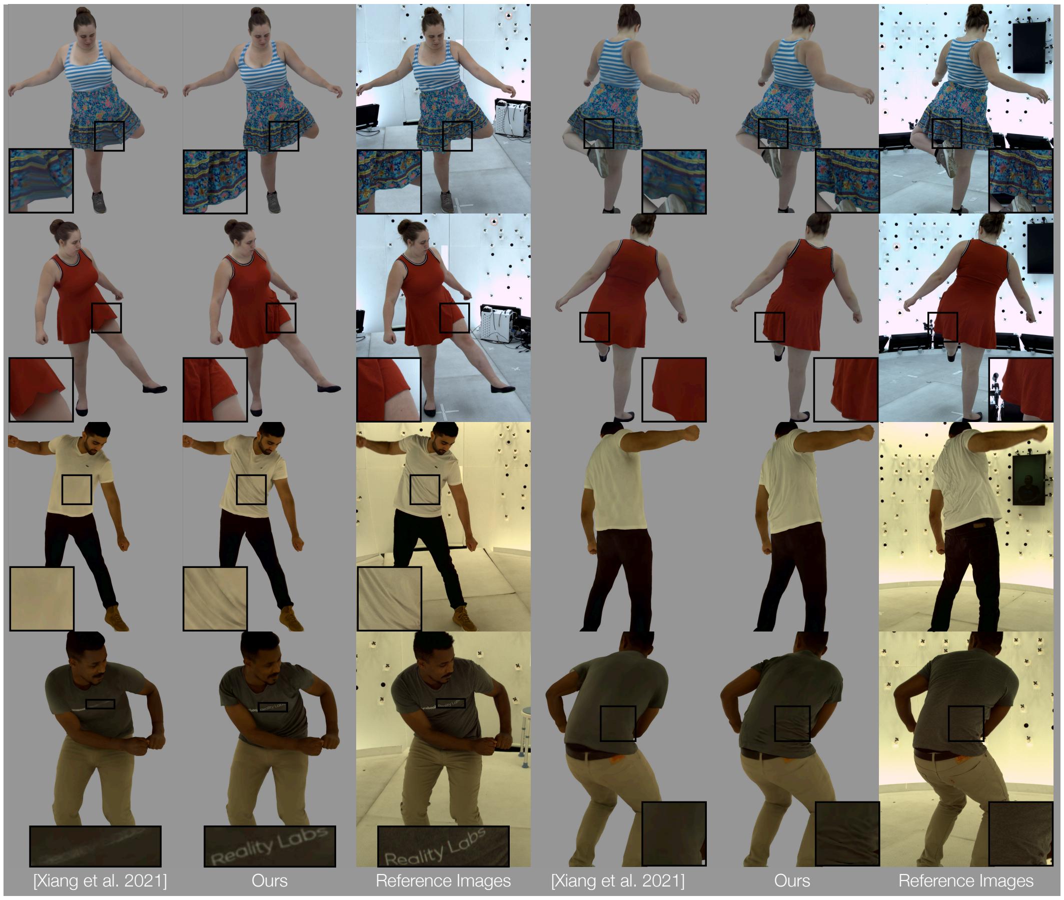

Figure 4.7: Pose-driven animation results in comparison with Chapter 3 [218]. In each row we show one result generated by both methods and a held out captured image for reference from two different views.

simulation (with scaling if necessary) and clothing appearance model in exactly the same way as the default pipeline. We show dressed avatars of the same identity (wearing a tight capture suit) and different identities. The animation results are shown in Figure 4.8 and in the supplementary video.

Application: Editing garment size

We edit the garment size by changing the rest length of the garment template in the cloth simulator and keeping the same set of physical parameters. We maintain the same mesh topology, so that our appearance model can be directly applied to the edited garment. The animation results after size editing are shown in Figure 4.9. This manipulation is only possible because both the physics-based simulator and our appearance model generalize well to unseen clothing configurations.

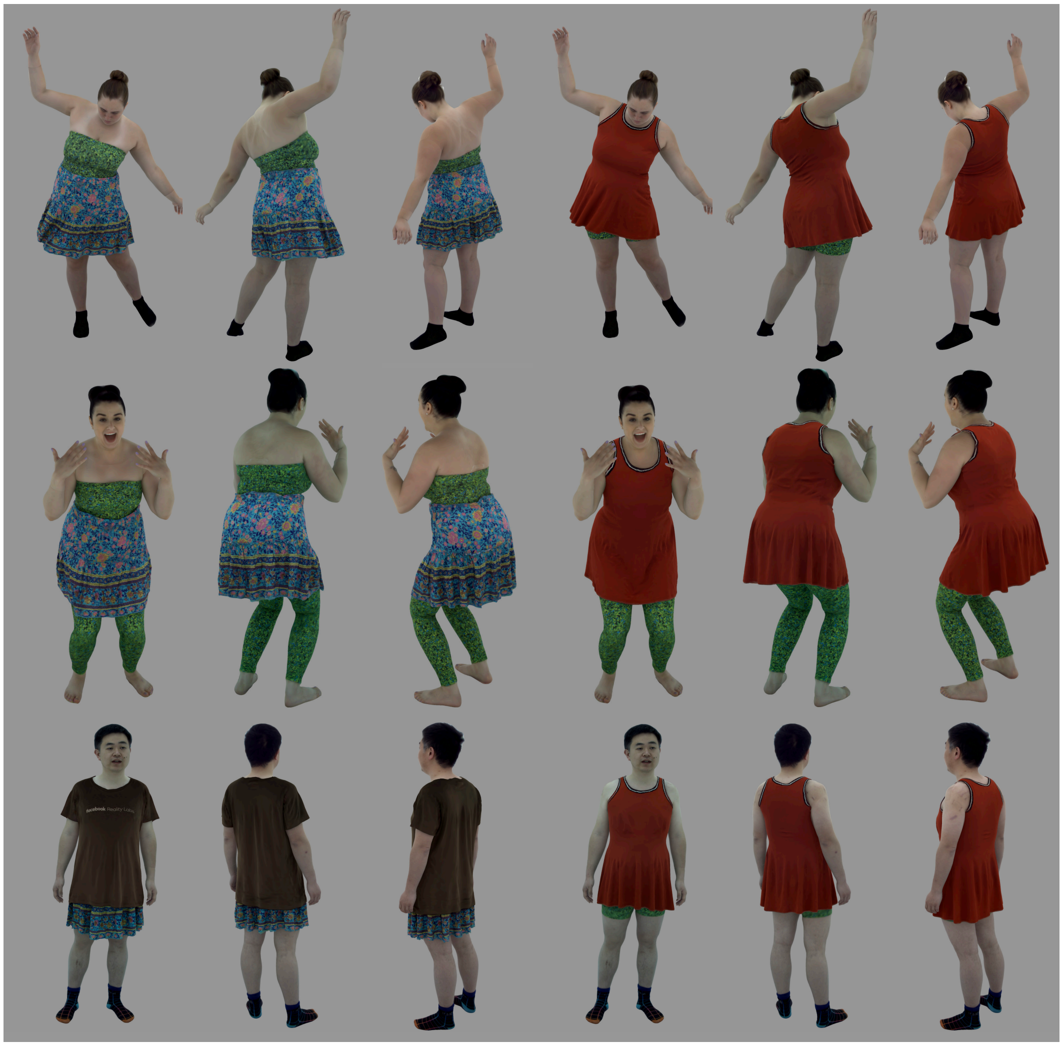

Figure 4.8: Dressing novel avatars. In the top row, a minimally clothed body avatar of the same actor as the original capture is shown in a skirt and dress. The middle and bottom rows show dressed avatars created from other actors. The bottom left results are dressed with two pieces of clothing together, a skirt and T-shirt.

4.6.5 Runtime Analysis

In this section we report the runtime for the key modules of our algorithm: the physics-based cloth simulator, the base body avatar and the deep clothing appearance model.

We use a GPU-based cloth simulator with the eXtended Position Based Dynamics (XPBD) formulation [130]. When simulating a garment of 35k vertices on top of a body sequence of 9k vertices, the average runtime per frame for the simulation solver is between 8 ms $\sim$ 10 ms when taking 20 steps per frame on a Nvidia Tesla V100 GPU.

The base body avatar and the clothing appearance models are implemented using the

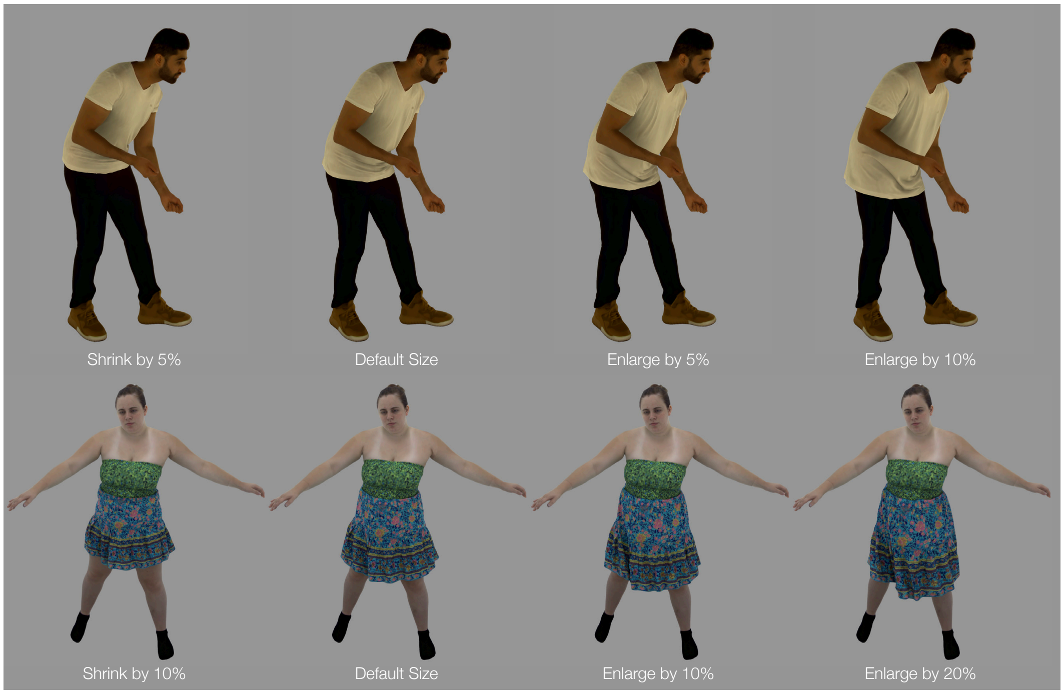

Figure 4.9: We edit garment sizes by scaling the template shape used in the physics-based simulation. Although the clothing appearance models are trained only using captured data of a particular size, they generalize well to the edited garments of different sizes.

PyTorch deep learning framework [145]. For deployment, we use PyTorch Just-In-Time (JIT) compilation to improve performance. The average runtime for those two modules combined together is 78 ms on a Nvidia RTX 3090 GPU on a Lambda workstation. We use multiple GPUs to parallelize the inference. For example, we can achieve $>$ 30 fps with 3 GPUs in parallel. In the supplementary video, we show a demo of viewing an animated sequence in a VR headset, where the cloth simulation is precomputed and the body avatars and clothing appearance are rendered in real-time. Although we currently run the cloth simulator and the neural networks separately, given the modest computation for each of the three modules, our pipeline has the potential to be integrated into a complete real-time online system.

4.6.6 The Effect of Cloth Simulation

In this section, we further investigate the influence of cloth simulation on the overall pipeline. We first show results of our method with different simulation parameters. Then we test our method with a different simulator from the default XPBD simulator.

Discussion: Simulation Parameters

In Figure 4.10, we show example results generated by our system using different material parameters in the cloth simulation. Two critical parameters in the simulation are bending

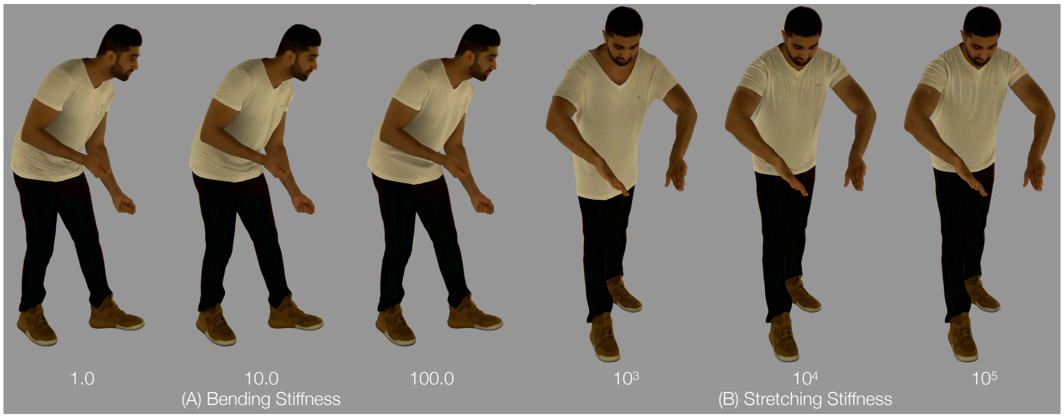

Figure 4.10: We show results of different physical parameters used in the cloth simulation. We adjust the scale of bending stiffness on the left (A), and stretching stiffness on the right (B). Notice that our rendered results are reasonable despite the difference in simulation parameters.

stiffness and stretching stiffness.

- (A) The bending stiffness controls the level of wrinkles. The larger the bending stiffness is, the fewer wrinkles remain in the output results.
- (B) The stretching stiffness influences the level of stretching of clothing at equilibrium. The larger the stretching stiffness is, the closer the output sticks to the rest length of the clothing template.

For the results in this chapter, we experiment with different physical parameters and select the output that is visually most similar to the captured images. The process is well illustrated by Figure 4.10. In addition, it can be observed in Figure 4.10 that the performance of our appearance model is not sensitive to the parameters. Therefore, we only need to devote modest efforts into the selection of parameters. More sophisticated approaches can also be adopted, including using measured material data [133,203], or building a perceptual control space for simulation [179], which are beyond the scope of this thesis.

Discussion: A Different Simulator

In this section, we demonstrate the possibility of using a different simulator from the our default XPBD simulator in our system. For this purpose, we adopt an open-source cloth simulator[5] based on Projective Dynamics with frictional contact modeling [125]. We compare the results generated by this simulator (named 'Projective Friction') with those from XPBD simulator in Figure 4.11. The body configurations and the clothing appearance model are kept the same for the comparison.

As shown in Figure 4.11, the XPBD simulator and Projective Friction simulator produce different clothing geometry due to the discrepancies in their formulations. However, our clothing appearance model can generate proper appearance with reasonable detail of wrinkles and shadow that agree with the corresponding geometry from both simulators. These

[5] https://gitlab.inria.fr/elan-public-code/projectivefriction

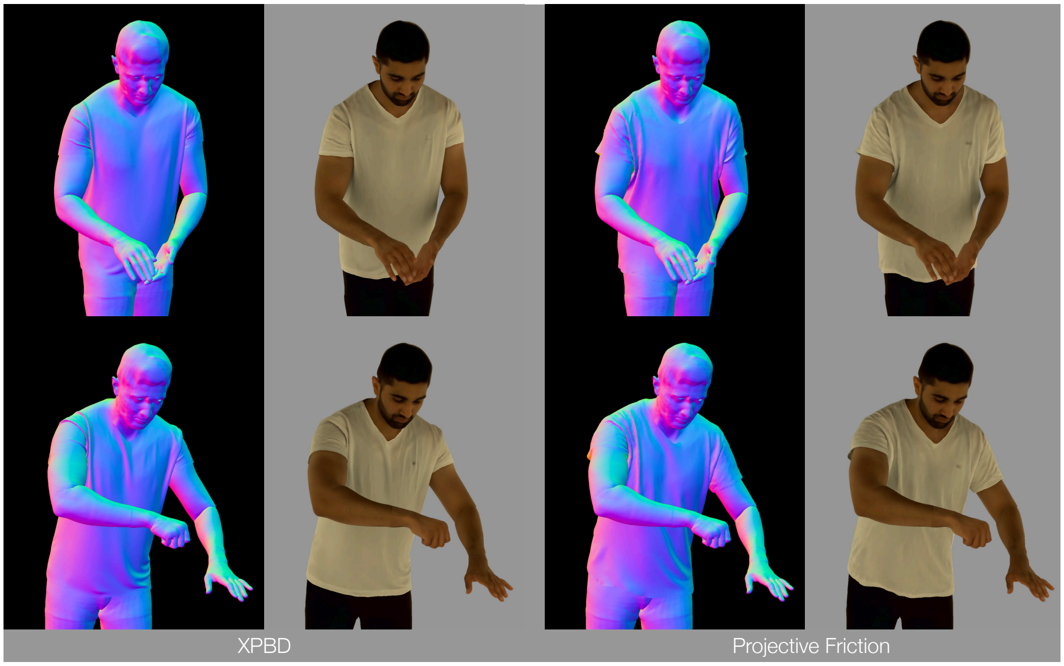

Figure 4.11: We compare animation results using our default XPBD simulator with Projective Friction [125]. We show normal rendering of the simulated geometry together with the underlying body on the left, and the results of our photorealistic clothing appearance model on the right.

results suggest that the clothing appearance model is not tied to a specific simulator. The animation framework that we present in this chapter has the potential to generalize to different implementation of physics-based cloth simulation.

4.7 Implementation Detail

4.7.1 Cloth Simulator

Our real-time cloth simulator implements eXtended Position Based Dynamics [55, 130]. XPBD is a constraint-based simulation model that often obtains much better performance compared to expensive non-linear solvers. It uses an iterative Gauss-Seidel solution for the linearized equations of motion. The method can be easily parallelized [55] and implemented on hardware such as multi-core CPUs and GPUs, enabling interactive or real-time simulations on common modern hardware.

The method aims to solve Newton's equations of motion

$$\mathbf{M}\ddot{\mathbf{x}} = -\nabla U(\mathbf{x}), \tag{4.5}$$

where $\mathbf{x} \in \mathbb{R}^{3n}$ encodes n vertex positions (of the cloth mesh in this case) and $\mathbf{M}$ is the mass matrix computed from element volumes and constant material density ρ.

The energy potential $U(\mathbf{x})$ needs to be specified in terms of a vector of constraint functions $\mathbf{C}(\mathbf{x}) = [C_1(\mathbf{x}), C_2(\mathbf{x}), \cdots, C_m(\mathbf{x})]^\top$ as

$$U(\mathbf{x}) = \frac{1}{2}\mathbf{C}(\mathbf{x})^\top \boldsymbol{\alpha}^{-1}\mathbf{C}(\mathbf{x}), \tag{4.6}$$

where $\boldsymbol{\alpha}$ is a block diagonal compliance matrix. Any energy that can be written this way is suitable for XPBD. Using implicit Euler time integration, the XPBD algorithm reduces to solving for the constraint multiplier updates $\Delta\boldsymbol{\lambda}$ with

$$(\nabla\mathbf{C}(\mathbf{x}_i)^\top \mathbf{M}^{-1}\nabla\mathbf{C}(\mathbf{x}_i) + \tilde{\boldsymbol{\alpha}})\Delta\boldsymbol{\lambda} = -\mathbf{C}(\mathbf{x_i}) - \tilde{\boldsymbol{\alpha}}\boldsymbol{\lambda}_i, \tag{4.7}$$

where $\mathbf{x}_i$ and $\boldsymbol{\lambda}_i$ are the values of $\mathbf{x}$ and $\boldsymbol{\lambda}$ at iteration i, and $\tilde{\boldsymbol{\alpha}} = \frac{\boldsymbol{\alpha}}{\Delta t^2}$. Then the position is updated by

$$\Delta\mathbf{x} = \mathbf{M}^{-1}\nabla\mathbf{C}(\mathbf{x}_i)\Delta\boldsymbol{\lambda}. \tag{4.8}$$

The system in Equation (4.7) is typically solved using Gauss-Seidel- or Jacobi-style updates. Stretching and shearing of the fabric is modeled as a mass-spring system whereas the bending is modeled as a zero angle constraint for dihedral elements. The underlying body is modeled as a triangle mesh and it is directly used for collision handling. Our solver is implemented using CUDA kernels and runs on the GPU.

Simulation Template

Our XPBD simulator supports using a 3D clothing mesh as the rest shape, from which the the reference triangle sizes for the stretching and shearing energy terms are computed. For the bending energy, we still assume zero rest angles for dihedral elements. This formulation allows us to directly use the same template mesh for clothing registration as the rest shape for the simulation.

The open-source simulator [125] requires a 2D template as the rest shape. We follow previous work [8] to create a 2D template from the 3D registration template. The basic idea is to cut the 3D template mesh into several pieces, flatten them with minimal distortion, and enforce boundary smoothness requirements.

It should be also possible to adopt a newer method [153] to create a 2D template. Another alternative is to modify the implementation of 'Projective Friction' [125] to support 3D template shapes.

4.7.2 Network Architecture

Base Body Avatars

In this work, we use two types of base body avatars, which we call minimally clothed avatars and underlying body avatars.

The minimally clothed avatars are built from captured sequences where the subjects wear only a green tight suit. To build these types of avatars, we use the same procedure as in previous work [7]. Because the capture suit tightly follow the body motion, the single-layer full-body avatar is able to model the full appearance. The avatars adopt a convolutional Variational Autoencoder (cVAE) architecture, conditioned on body pose, facial conditioning (if applicable), a latent code from the encoder, and ambient occlusion as input. At test (animation) time, we follow previous work [7] and use a fixed latent code (all zeros in our case) for all the body motion in the test sequence.

The underlying body avatars are built from normally clothed body capture as in Chapter 3. To train these types of avatars, we register the clothing and body in two separate layers. The clothing registration method is described in Section 4.4. In order to track body under loose clothing, we utilize the minimally clothed body data of the same subject as a prior. When tracking the skeleton poses using the clothed body reconstruction in Section 3.8.1, we exclude the highly dynamic clothing region (bottom of the dress and the whole skirt) in the surface distance loss. To estimate the underlying body surface, we couple the invisible region of the body shape with the minimally clothed LBS model, and penalize collisions with the clothing surface similar to [242]. With the tracked body data, we train the body-layer avatars with the same network architecture as described in Figure 3.18, but without the clothing branch. The training process is similar to the description in Section 3.5.1 and 3.5.3.

Clothing Appearance Model

In this section we provide more implementation detail on the clothing appearance model. We use the same architecture for both the view-independent and view-dependent networks, which is described in Figure 4.12. For the shadow network, the architecture is the same as Chapter 3. The appearance model is trained end-to-end in PyTorch using the AdamW optimizer with an initial learning rate of 1×10^{-4}. The training goes on for 100k iterations with the batch size of 2.

4.8 Discussion

In this work, we present efficient full-body clothed avatars with physically plausible dynamics and photorealistic appearance. Our method achieves high-fidelity registration of dynamic cloth geometry and learns a deep clothing appearance model based on the registered geometry for training. At test time, the clothing is deformed through physics-based simulation and is rendered by the appearance model. To bridge the gap between tracked cloth geometry at training time and simulated clothing geometry at test time, we design an input representation and an architecture that consists of three modules inspired by the rendering equation. Once a garment is modeled, our system can use it to *dress* a novel avatar.

Limitations.

Clothing has a very large space of geometric and appearance variations. Although our method handles single-layer loose clothing much better than prior work, it may have difficulty in dealing with large folds of extremely loose clothing such as a Kimono and it may have difficulty in dealing with large folds of extremely loose clothing such as a Kimono or multi-layered clothing such as a coat and dress where the inner layers are significantly occluded. The photometric registration may fail when the feature matching is inaccurate due to bad initialization and/or repeated patterns. Some possibilities for improvement include using simulated clothing data with high-quality offline rendering to train the appearance model, or to utilize clothing with printed markers for registration [71,212].

In this work we manually adjust the physical parameters of the simulated garments. Optimization-based parameter estimation may improve the accuracy by leveraging recent

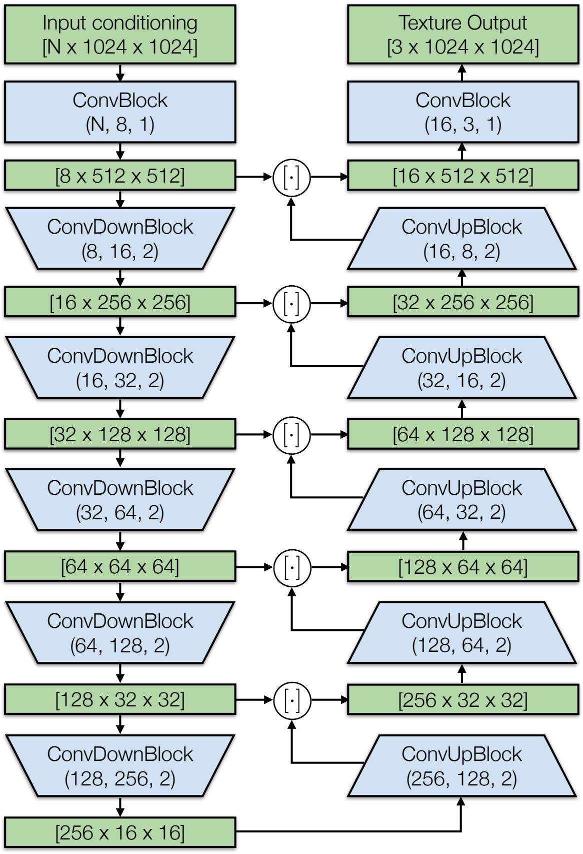

Figure 4.12: The detailed architecture of view-independent and view-dependent networks used in the clothing appearance models. We adopt the UNet [164] structure with skip connections. Data tensors are shown in green where the numbers represent '[channels, height, width]'. Network modules are shown in blue, where the numbers represent '(input channels, output channels, stride)'. For the exact detail of each block, please refer to the Figure 3.16.

advancements in differentiable cloth simulators [110, 112]. In addition, the physics-based cloth simulation may face challenges in handling highly complicated hand-cloth interactions such as dragging and pinching, due to the real-time constraint imposed on the simulation.

Although the clothing appearance model is trained using the data captured for only one subject, the shadow branch shows reasonable generalization ability when used to dress novel avatars because it is influenced by the body shape only indirectly through the occlusion of rays. Nevertheless, how the model works for extreme body shapes remains untested due to the limited availability of minimally clothed data. A systematic investigation of shadowing across different garments and identities is an interesting future direction but beyond the scope of this thesis. Furthermore, we do not consider complicated interreflection of lighting between clothing and body and between the folds of clothing. We also assume consistent illumination conditions when dressing novel avatars. When this assumption does not hold, slight inconsistency can be observed, for example, in the two-cloth animation results (T-shirt and skirt). For future work beyond this thesis, we may relax this constraint by incorporating a relighting capability [14] to our avatar and clothing models with an updated capture setup.

Chapter 5

Temporally Coherent Clothing Capture from Monocular RGB Video

5.1 Introduction

Capturing a temporally coherent shape for clothing from monocular RGB imagery is an extremely challenging task, due to the fundamental ambiguity of single-view 3D reconstruction and the large deformation space of clothing. Previous work [68, 70, 224] utilizes a 3D personalized actor model as a shape prior to track the dynamic clothing deformation. This model is acquired by multi-view reconstruction on an additional video of the same actor wearing the same clothing and rotating in a T-pose. However, such a model is generally unavailable for in-the-wild videos. The need for a pre-scanned template model limits the applicability of these approaches.

With the development of deep neural networks, other efforts have been made to regress a clothed human shape directly from a single input image with supervised learning [4, 137, 167, 168, 193, 197, 250]. These methods produce plausible results for individual input images of common human poses. However, it is difficult to extend them to capture temporally coherent dynamic clothing deformation from monocular videos for the following reasons. First, these methods are not robust to the variety of human motion due to the limited diversity of training data. They can easily produce incomplete geometry that is difficult to fix via post-processing. Second, it is non-trivial to estimate the temporal correspondence from the output of individual frames due to the data representation used (voxel [197,250], depth map [193] or implicit function [4, 137, 167, 168, 193, 197, 250]). This limits the application of these methods in scenarios that require correspondence, such as clothing retargeting or image editing.

In this work, we present a novel method to capture dynamic clothing deformation from a monocular RGB video in a *temporally coherent* manner, as illustrated by Figure 5.1. To the best of our knowledge, it is the first attempt to solve this challenging problem without the prerequisite of a pre-scanned personalized template [68, 70, 224].

Our method is based on the following observations. First, a deformation model of the clothing that provides a statistical shape prior is key to solving the problem. It not only re-

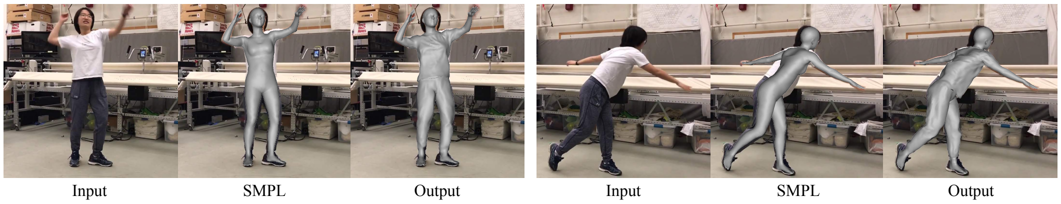

Figure 5.1: Given a monocular RGB video as input, our proposed method captures temporally coherent dynamic clothing deformation that cannot be explained by a bare human body model. From left to right, we show the input frame, body capture results using the SMPL model [120], and our clothing capture results.

duces the ambiguity of single-view 3D reconstruction, but also helps to estimate temporal correspondence across frames. While clothing models have been investigated in the existing literature [128,232] for the purpose of clothing shape generation, our work is the first study that fully demonstrates the value of a clothing model for RGB-based clothing capture[1]. Second, to solve the clothing capture problem, we make use of human appearance information including silhouette, segmentation, texture and surface normal. We present a novel method to integrate all those image measurements using a differentiable renderer [36,116]. Our method captures the realistic dynamic of clothing in a temporally coherent manner including fine-grained wrinkle details from various videos.

Our Contributions. (1) We present the first approach for temporally coherent clothing capture from a monocular RGB video without using a pre-scanned template of the subject. (2) We propose a novel method to capture clothing deformation by fitting statistical clothing models to image measurements including silhouette, segmentation, texture and surface normal with a differentiable renderer.

This work is published at International Conference on 3D Vision (3DV) 2020 and received the Best Paper Honorable Mention Award.

5.2 Related Work

In this work, we build a linear clothing model in order to constrain the garment deformation space for capture. An existing approach [232] also builds a linear clothing model. However, similar to most clothed human models reviewed in Section 2.1, this model is primarily used for shape generation, while we use the model to track clothing deformation from a monocular video.

One line of work also reconstructs clothing shape from a monocular video or several images by allowing per-vertex deformation on top of the SMPL body model. Alldieck and colleagues [2, 3] build clothed human avatar from videos of a person slowly rotating in A-pose. This is further improved to use only images of several different views [1, 13] or even a single image [4]. However, these methods reconstruct clothing as near-static objects without considering the temporal dynamics. By contrast, in this work, we address the challenging problem of capturing clothing dynamics from a monocular video.

Human performance capture refer to the capture of space-time coherent sequences of

[1] Due to the limitation in types of available clothing data to train our model, in this chapter, we assume that the subject to be captured wears a T-shirt on the upper body and shorts or pants on the lower body.

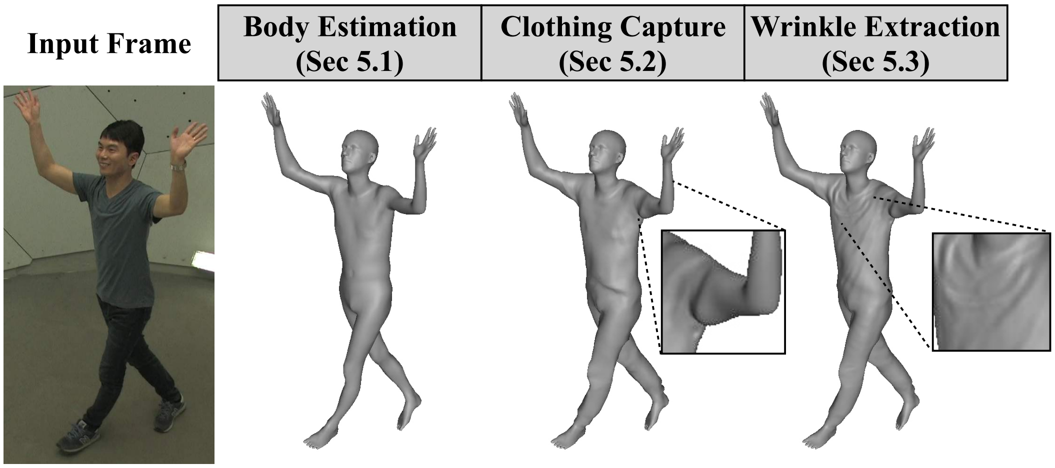
Input Frame
Body Estimation (Sec 5.1)
Clothing Capture (Sec 5.2)
Wrinkle Extraction (Sec 5.3)

Figure 5.2: An overview of our clothing capture pipeline.

surface geometry respectively. Most relevant to our work are performance capture methods from monocular RGB videos [68,224]. These methods, however, require a pre-scanned mesh template of the subject, which restricts the applications where they can be used. Habermann *et al.* [70] further proposes to train a deep neural network to deform a pre-scanned mesh template to match the surface deformation in the video. This method, requires the mesh template and multi-view images of the subject for network training. Our method relaxes the constraint to scenarios such as in-the-wild videos where neither pre-scanned templates nor multi-view images are available.

5.3 Method Overview

In this section, we present an overview of our approach. Our goal is to capture the dynamic deformation of three types of garments, T-shirt, shorts and pants, along with the underlying body shape from a monocular video. Our method takes as input a sequence of images, denoted as $\{\mathbf{I}_i\}_{i=1}^{F}$, where F is the number of frames in the sequence. The subject is assumed to be wearing a T-shirt for the upper body. The clothing for the lower body is manually identified as either short pants or long pants. Our method outputs a sequence of mesh pairs $\{\mathbf{M}_i^b, \mathbf{M}_i^c\}_{i=1}^{F}$, where $\mathbf{M}_i^b$ denotes the body mesh and $\mathbf{M}_i^c$ denotes the clothed mesh. $\{\mathbf{M}_i^b\}_{i=1}^{F}$ and $\{\mathbf{M}_i^c\}_{i=1}^{F}$ are both temporally coherent with fixed topology across time. $\mathbf{M}_i^b$ and $\mathbf{M}_i^c$ share the same vertex positions except for the clothing region.

Our method makes use of linear clothing deformation models defined in the canonical pose. We briefly describe our model formulation and model building procedure in Section 5.4. Our pipeline to capture clothing from a monocular video consists of four stages, explained in Section 5.5. First, we estimate the underlying body pose and shape of subject (Section 5.5.1). Then, we run sequential tracking of the clothing using our linear clothing models. This step is followed by a batch optimization stage including all the frames to produce temporally coherent dynamic clothing deformation (Section 5.5.2). In the final stage, we add fine-grained wrinkle detail to our results (Section 5.5.3). A visualization of this pipeline is shown in Figure 5.2.

5.4 Statistical Clothing Deformation Model

Statistical models of clothing have been investigated for clothing shape generation in the previous literature [128, 232], but have yet to be exploited for capturing clothing from a monocular video. In this section we give the mathematical formulation of our clothing deformation models and briefly describe the procedure to learn these models from data.

5.4.1 Model Formulation

Our clothing models are built on top of the SMPL body model [120]. SMPL is controlled by a set of model parameters $(\boldsymbol{\beta}, \boldsymbol{\theta})$, where $\boldsymbol{\beta} \in \mathbb{R}^{10}$ is the shape coefficients and $\boldsymbol{\theta} \in \mathbb{R}^{72}$ is the joint angles that control body pose. We denote the set of $n_v = 6890$ output vertices by $M(\boldsymbol{\beta}, \boldsymbol{\theta})$. Then formally,

$$M(\boldsymbol{\beta}, \boldsymbol{\theta}) = W(T(\boldsymbol{\beta}, \boldsymbol{\theta}), J(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathcal{W}), \tag{5.1}$$

where W is the Linear Blend Skinning (LBS) function; $T(\boldsymbol{\beta}, \boldsymbol{\theta})$ is the rest pose body shape; $J(\boldsymbol{\beta})$ is the locations of 24 kinematic joints; $\mathcal{W}$ is the blend weights. In particular, the unposed shape $T(\boldsymbol{\beta}, \boldsymbol{\theta})$ is defined as the sum of template shape $\overline{T}$, shape dependent deformation $B^S(\boldsymbol{\beta})$ and pose dependent deformation $B^P(\boldsymbol{\theta})$,

$$T(\boldsymbol{\beta}, \boldsymbol{\theta}) = \overline{T} + B^S(\boldsymbol{\beta}) + B^P(\boldsymbol{\theta}) \tag{5.2}$$

On top of the SMPL model, an extra additive offset field D is introduced to account for clothing deformation in rest pose, i.e.,

$$T^c = T(\boldsymbol{\beta}, \boldsymbol{\theta}) + D. \tag{5.3}$$

D includes a number of n_v per-vertex offsets, each denoted by $D_j \in \mathbb{R}^3$, where $1 \leq j \leq n_v$. Here we decompose D into offsets from the upper clothing D^u and offsets from the lower clothing D^l. D^u and D^l share the same dimensionality as D. They take non-zero values if the respective garments cover body vertex j; for an exposed skin vertex we have $D^u_j = D^l_j = 0$. Notice that some body vertices might be covered by both upper and lower clothing, for example, around the waist. To account for this phenomenon, we merge D^u and D^l into a single offset field D by

$$D_j = \begin{cases} D^u_j & \text{if } \|D^u_j\| \geq \|D^l_j\|, \\ D^l_j & \text{otherwise.} \end{cases} \tag{5.4}$$

The dimensions of D^u and D^l are very high $(3n_v)$, so we use PCA dimension reduction to enable control with low-dimensional parameters $\mathbf{z}^u, \mathbf{z}^l \in \mathbb{R}^{n_z}$. Formally,

$$D^k(\mathbf{z}^k) = \mathbf{A}^k \mathbf{z}^k + \overline{\mathbf{d}^k},\ k \in \{u, l\} \tag{5.5}$$

where $\mathbf{A}^k \in \mathbb{R}^{3n_v \times n_z}$ is the matrix of PCA bases and $\overline{\mathbf{d}^k}$ is the mean value vector. We use the skinning function W of SMPL to transform the clothed shape from rest pose to target pose. Finally, our clothing model is formulated as

$$M^c(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{z}) = W(T^c(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{z}), J(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathcal{W}), \tag{5.6}$$

$$T^c(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{z}) = T(\boldsymbol{\beta}, \boldsymbol{\theta}) + D(\mathbf{z}), \tag{5.7}$$

where $\mathbf{z} = \{\mathbf{z}^u, \mathbf{z}^l\}$ is the collection of clothing parameters. A visual illustration of our clothing model formulation is shown in Figure 5.3.

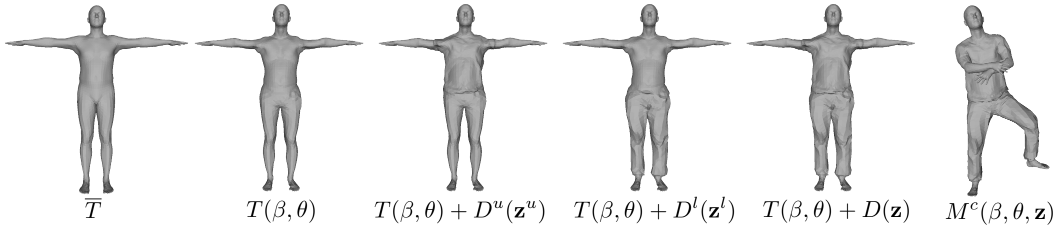
$\overline{T}$
$T(\beta, \theta)$
$T(\beta, \theta) + D^u(\mathbf{z}^u)$
$T(\beta, \theta) + D^l(\mathbf{z}^l)$
$T(\beta, \theta) + D(\mathbf{z})$
$M^c(\beta, \theta, \mathbf{z})$

Figure 5.3: A visualization of our clothing model formulation. From left to right, we show (1) the mean SMPL template, (2) personalized body shape with pose dependent deformation, (3) upper clothing offsets, (4) lower clothing offsets, (5) combined clothing offsets, and (6) posed clothing output.

5.4.2 Model Building

We build our models from the BUFF dataset [242], a collection of high-resolution 4D people scan. We build a model for each of the three garment types in the dataset, T-shirts, shorts and pants. For each garment type k, we need to train the model parameters $\{\mathbf{A}^k, \overline{\mathbf{d}^k}\}$ from a collection of clothing offsets, denoted by $\mathbf{X}^k \in \mathbb{R}^{3n_v \times n_k}$, where n_k is the number of samples for garment type k in the dataset. To obtain each sample in the collection, we follow [155, 242] to register the raw scan with SMPL model. This operation not only brings the raw scan data into the same topology, but also "unposes" the human shape with clothing into the rest pose, denoted by $\mathbf{X}^c$. We also follow [242] to estimate the underlying body shape of the subject in rest pose, denoted by $\mathbf{X}^b$. In addition, we obtain a per-vertex binary mask $\boldsymbol{\sigma}^k$ that has value 1 for the region of garment type k and 0 for any other regions (skin and other clothing types) by rendering the meshes to images and applying a state-of-the-art clothing segmentation algorithm [61]. Then we obtain the clothing offset data $\mathbf{X}^k$ for garment type k by

$$\mathbf{X}^k = (\mathbf{X}^c - \mathbf{X}^b) \odot \boldsymbol{\sigma}^k, \tag{5.8}$$

where $\odot$ denotes the element-wise multiplication. We use a standard PCA training algorithm based on Singular Value Decomposition (SVD), leaving $n_z = 50$ bases in our model. We refer readers to the original papers [155,242] for details on scan registration and underlying body shape estimation.

5.5 Monocular Clothing Capture

Given the pre-trained clothing models, we now present our approach for temporally coherent clothing capture from only a monocular video.

5.5.1 Body Motion Estimation

In the first stage, we estimate underlying body motion in 3D with the SMPL body model. We estimate per-frame SMPL pose parameters $\boldsymbol{\theta}_i$ and global translation $\mathbf{t}_i \in \mathbb{R}^3$, together with SMPL shape parameters $\boldsymbol{\beta}$ across the whole sequence. Meanwhile, we estimate the camera intrinsics $\mathbf{K}$ of a full perspective projection model for all the frames. In order to achieve good robustness under different in-the-wild scenarios, we integrate a variety of different image measurements into an energy optimization problem. Formally, we solve the following minimization problem:

$$\min_{\boldsymbol{\beta},\{\boldsymbol{\theta}_i,\mathbf{t}_i\}_{i=1}^F,\mathbf{K}} E^b = E^b_{\text{2d}} + E^b_{\text{dp}} + E^b_{\text{sil}} + E^b_{\text{pof}} + E^b_{\text{reg}}. \tag{5.9}$$

In particular, E^b_{2d} is the squared L_2 error between projected SMPL joints and 2D keypoint detection from OpenPose [27,28,210]. E^b_{dp}, also used in Guler *et al.* [63], is an energy term for dense correspondence estimation from DensePose [5]. Specifically, for any pixel $\mathbf{p}$ in the image with DensePose prediction, we identify the corresponding SMPL vertex index $j(\mathbf{p})$ and optimize an energy term defined as

$$E^b_{\text{dp}} = \frac{1}{F}\sum_{i=1}^{F}\sum_{\mathbf{p}} \|\Pi(M_{j(\mathbf{p})}(\boldsymbol{\beta},\boldsymbol{\theta}_i) + \mathbf{t}_i; \mathbf{K}) - \mathbf{p}\|^2, \tag{5.10}$$

where Π denotes the projection function determined by the camera intrinsics $\mathbf{K}$. E^b_{sil} is the silhouette matching term. We extract silhouettes $\mathbf{S}_i$ of our SMPL body mesh with a differentiable renderer [116], and obtain the target silhouette $\hat{\mathbf{S}}_i$ from a clothing segmentation algorithm [61]. We use an Intersection-over-Union error [116]

$$E^b_{\text{sil}} = \frac{1}{F}\sum_{i=1}^{F}\left(1 - \frac{\|\mathbf{S}_i \odot \hat{\mathbf{S}}_i\|_1}{\|\mathbf{S}_i + \hat{\mathbf{S}}_i - \mathbf{S}_i \odot \hat{\mathbf{S}}_i\|_1}\right). \tag{5.11}$$

E^b_{pof} is an error term based on 3D orientation between adjacent joints in the body skeleton hierarchy. We match the spatial orientation of SMPL body joints to the prediction of Part Orientation Field (POF) similar to [217]. We refer readers to the original papers [123,217] for details. We also apply regularization on our estimation, denoted by E^b_{reg}, which consists of a Mixture of Gaussian prior for body pose $\{\boldsymbol{\theta}_i\}_{i=1}^F$ [16], L_2 regularization on the shape parameters $\boldsymbol{\beta}$, and temporal smoothness terms to reduce motion jitters.

After solving the energy optimization, we obtain a temporally consistent body mesh for every frame by $\mathbf{M}^b_i = M(\boldsymbol{\beta},\boldsymbol{\theta}_i)$. We fix the SMPL parameters $\boldsymbol{\beta}, \{\boldsymbol{\theta}_i, \mathbf{t}_i\}_{i=1}^F$ and camera parameters $\mathbf{K}$ during later stages of our pipeline. The estimated body meshes provide a strong guidance for the subsequent estimation of clothing deformation.

5.5.2 Clothing Deformation Capture

We now illustrate our proposed method to capture clothing deformation. Compared to previous work [68, 70, 224] where a pre-scanned template of the subject is assumed, this problem is significantly more challenging due to the lack of strong shape prior to resolve the single-view 3D ambiguity, and the lack of a pre-defined personalized texture that provides correspondence for surface tracking. To solve this problem, we (1) exploit the deformation space learned in our clothing models and (2) progressively extract a personalized texture from the input image sequence to enable surface tracking across time and reduce drifting.

We perform clothing capture in a sequential manner. For each frame i, we estimate per-frame clothing parameters for clothing on the upper and lower body $\mathbf{z}_i = \{\mathbf{z}^u_i, \mathbf{z}^l_i\}$ given the input image $\mathbf{I}_i$, initializing from the result of the previous frame $\mathbf{z}_{i-1}$. We formulate the task as solving an energy optimization problem, formally,

$$\min_{\mathbf{z}_i} E^c = E^c_{\text{sil}} + E^c_{\text{seg}} + E^c_{\text{photo}} + E^c_{\text{reg}}. \tag{5.12}$$

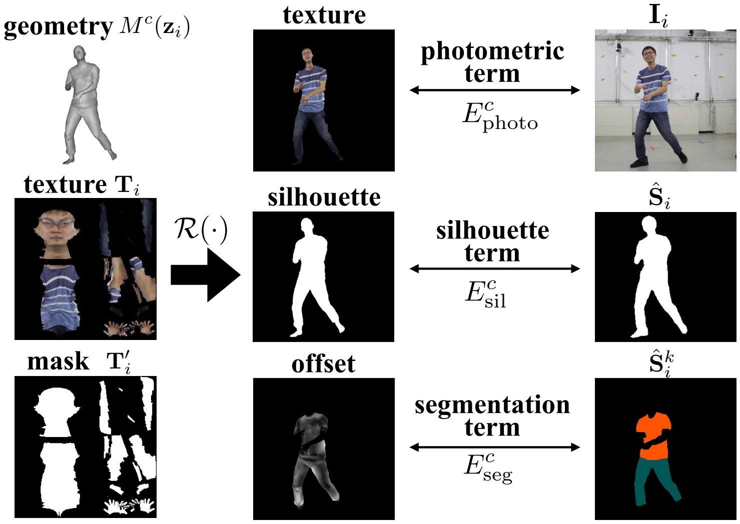

Figure 5.4: Explanation of the energy terms used for clothing capture. We obtain the rendered texture, silhouette and clothing offset with a differentiable renderer, which are compared with target images using different energy terms.

Now we explain each cost term individually. An illustration of the different cost terms is shown in Figure 5.4.

Silhouette matching term E^c_{sil}: Similar to E^b_{sil} in the first stage, we use a differentiable renderer to match the silhouette of our rendering output with a target silhouette extracted from the original image. Differently, here what we compare with target silhoutte is the silhoutte of human shape with clothing $M^c(\boldsymbol{\beta}, \boldsymbol{\theta}_i, \mathbf{z}_i)$, instead of bare body shape $M(\boldsymbol{\beta}, \boldsymbol{\theta}_i)$ in the previous stage.

Clothing segmentation term E^c_{seg}: Clothing segmentation [61] provides not only the overall silhouette of the person, but also the boundary between different garment regions and exposed skin in the image. We utilize this information by penalizing the clothing offset on vertices whose projection falls outside the segmentation region. Concretely, the differentiable renderer is used to render the per-vertex offset fields $D^u(\mathbf{z}^u_i), D^l(\mathbf{z}^l_i)$ on the clothed mesh $M^c(\mathbf{z}_i)$[2]; we denote the output by $\mathcal{R}\left(D^k(\mathbf{z}^k_i)\right)$, where k represents either u for upper clothing or l for lower clothing. In addition, we denote the segmentation masks for the clothing region $k \in \{u, l\}$ by $\hat{\mathbf{S}}^k_i$. Then we have

$$E^c_{\text{seg}} = \sum_{k \in \{u,l\}} \sum_{\mathbf{p}} (1 - \hat{\mathbf{S}}^k_i) \odot \left\| \mathcal{R}\left(D^k(\mathbf{z}^k_i)\right) \right\|^2, \tag{5.13}$$

where $\mathbf{p}$ iterates over all pixels in the image. Effectively, for each clothing type we penalize the clothing offset outside the corresponding clothing region, where $\hat{\mathbf{S}}^k_i$ is 0. The gradient in the image domain is propagated to the mesh by the differentiable renderer $\mathcal{R}$.

Photometric tracking term E^c_{photo}: This term is introduced to estimate temporal correspondence more accurately, especially when the garments we capture have high-contrast texture.

[2] SMPL body parameters $\boldsymbol{\beta}, \boldsymbol{\theta}_i$ and $\mathbf{t}_i$ are fixed in this stage thus omitted here.

We progressively build a personalized RGB texture image $\mathbf{T}_i$ in a pre-defined UV space of SMPL model, along with a binary mask $\mathbf{T}'_i$ that indicates texels where RGB values in $\mathbf{T}_i$ have been identified. The photometric tracking term is defined to compare the rendered output of our clothing models using $\mathbf{T}_i$ with the input image. For this purpose we use a differential renderer that works with UV texture [36,80], and denotes the rendered output as $\mathcal{R}(\mathbf{T}_i)$. We also render the mesh with $\mathbf{T}'_i$ to indicate pixels where texture from $\mathbf{T}_i$ is available. Formally, we have

$$E^c_{\text{photo}} = \sum_{\mathbf{p}} \|\mathcal{R}(\mathbf{T}_i) - \mathbf{I}_i\|^2 \odot \mathcal{R}(\mathbf{T}'_i), \tag{5.14}$$

where the summation is taken over the pixels in the image. After the optimization in Equation (5.12) is solved for frame i, we update $\mathbf{T}_i, \mathbf{T}'_i$ to obtain $\mathbf{T}_{i+1}, \mathbf{T}'_{i+1}$, which will be used for solving optimization for frame $i+1$. To achieve this, we project $\mathbf{I}_i$ to the mesh surface and fill in new RGB values to UV texels in $\mathbf{T}_i$ where no previous values have been identified, indicated by 0s in $\mathbf{T}'_i$. Corresponding texels in $\mathbf{T}'_i$ are also set to 1 to obtain $\mathbf{T}'_{i+1}$. This process is initialized by setting $\mathbf{T}_1$ and $\mathbf{T}'_1$ to 0; in other words, the photometric tracking term takes no effect for the first frame in the sequence, since no texture has been extracted.

Regularization term E^c_{reg}: Our clothing deformation models are PCA-based linear models. They may produce unreasonable shapes when the parameters **z** are large. Therefore we apply regularization on the cloth parameters using an adaptive cost function ρ that penalizes large input values:

$$E^c_{\text{reg}} = \rho\left(\|\mathbf{z}_i\|^2\right). \tag{5.15}$$

The sequential tracking stage is then followed by a batch optimization stage that optimize for all F frames in the sequence together. The energy function we use is the same as the previous stage (Equation 5.12) with an additional term that penalizes too drastic temporal change of clothing parameters, which helps to produce temporally stable results. This term is defined as

$$E^c_{\text{temp}} = \frac{1}{F-1}\sum_{i=1}^{F-1} \|\mathbf{z}_{i+1} - \mathbf{z}_i\|^2. \tag{5.16}$$

The output of batch optimization stage is a sequence of capture results with clothing $\{\mathbf{M}^c_i = M^c(\boldsymbol{\beta}, \boldsymbol{\theta}_i, \mathbf{z}_i)\}_{i=1}^F$.

5.5.3 Wrinkle Detail Extraction

Up to the batch optimization stage, we can capture large clothing deformation. However, the results are limited by low mesh resolution and unable to capture the fine-grained wrinkles on the clothing. Therefore, the last stage of our approach is to extract wrinkle details from the original images and apply them to our coarsely tracked meshes.

Traditionally, such wrinkle details are captured with Shape from Shading (SfS) [2,214]. For in-the-wild monocular clothing capture, we empirically find it difficult to extract wrinkles reliably by SfS due to complex garment albedo, large variation of lighting conditions and self-shadowing. Recently, we observed the success of learning-based approaches in estimating accurate surface normal for human appearance using neural networks [168,193]. The estimated surface normal provides strong and direct clues on how the wrinkles should be added to our clothing capture results in order to match the original images.

Formally, let us denote the output of a surface normal estimation network for frame i to be $\mathbf{I}_i^n$, a 3-channel normal map for each pixel in the original image. We first subdivide the mesh $\mathbf{M}_i^c$ with Loop subdivision to increase the spatial resolution, with the subdivided mesh denoted by $\mathbf{M}_i^s$. Then, we solve for a deformed mesh $\mathbf{O}_i$ whose rendered normal map matches the estimated normal map $\mathbf{I}_i^n$ in the garment region. We denote the rendered normal output by $\mathcal{R}^n(\mathbf{O}_i)$, where $\mathcal{R}^n$ is the differential renderer. We solve the following optimization problem for each frame i individually:

$$\min_{\mathbf{O}_i} E^w = E^w_{\text{normal}} + E^w_{\text{reg}} + E^w_{\text{lpl}}, \tag{5.17}$$

$$E^w_{\text{normal}} = \sum_{\mathbf{p} \in \hat{\mathbf{S}}_i^c} \|\nabla \mathcal{R}^n(\mathbf{O}_i) - \nabla \mathbf{I}_i^n\|^2, \tag{5.18}$$

$$E^w_{\text{reg}} = \|\mathbf{O}_i - \mathbf{M}_i^s\|^2, \; E^w_{\text{lpl}} = \|\mathbf{L}\mathbf{O}_i\|^2 . \tag{5.19}$$

where ∇ denotes the image gradient operator, and $\mathbf{L}$ denotes the mesh Laplacian operator, $\hat{\mathbf{S}}_i^c = \hat{\mathbf{S}}_i^u \bigcup \hat{\mathbf{S}}_i^l$ is the union of all pixels in the clothing segmentation masks. Here we penalize the difference between normal maps in the image gradient domain to be more tolerant to error in absolute normal direction from the neural network. We also restrict the deformation of $\mathbf{O}_i$ from $\mathbf{M}_i^s$ be in the direction of the camera rays. Here, we use the implementation of differentiable renderer in [36,80] and surface normal network in [168]. The final results are the deformed meshes $\{\mathbf{O}_i\}_{i=1}^F$.

5.6 Quantitative Evaluation

In this section, we present the results of quantitative evaluation. We use a benchmark sequence from the MonoPerfCap dataset [224] and video sequences rendered from BUFF dataset [242] to test the performance of our method.

5.6.1 Evaluation on MonoPerfCap Dataset

Experiment Setting. We follow previous work [224] to use the *Pablo* sequence in their dataset to perform quantitative comparison. Surface meshes and 3D joints obtained by a multi-view performance capture method [163] are provided as the ground truth in the dataset. We compare our method with a state-of-the-art template-based performance capture method [224][3] and single-image human reconstruction methods [4,167,168,250,253]. Body pose is not estimated in [4], so we apply our estimated body pose to their T-pose results.

Evaluation of Clothing Surface Reconstruction. We first evaluate our method using a surface reconstruction metric. Due to the intrinsic depth-scale ambiguity of single-view reconstruction, we compute a global scaling factor from our result to the ground truth, which is applied to our result before comparison. Following [224] we align our results to the ground truth with a translation to eliminate the global depth offsets. We compute the average point-to-surface distance from all the ground truth vertices in the clothing region to the output mesh as the evaluation metric. The clothing region (the T-shirt and shorts) is

[3] Monocular capture results are provided by the author.

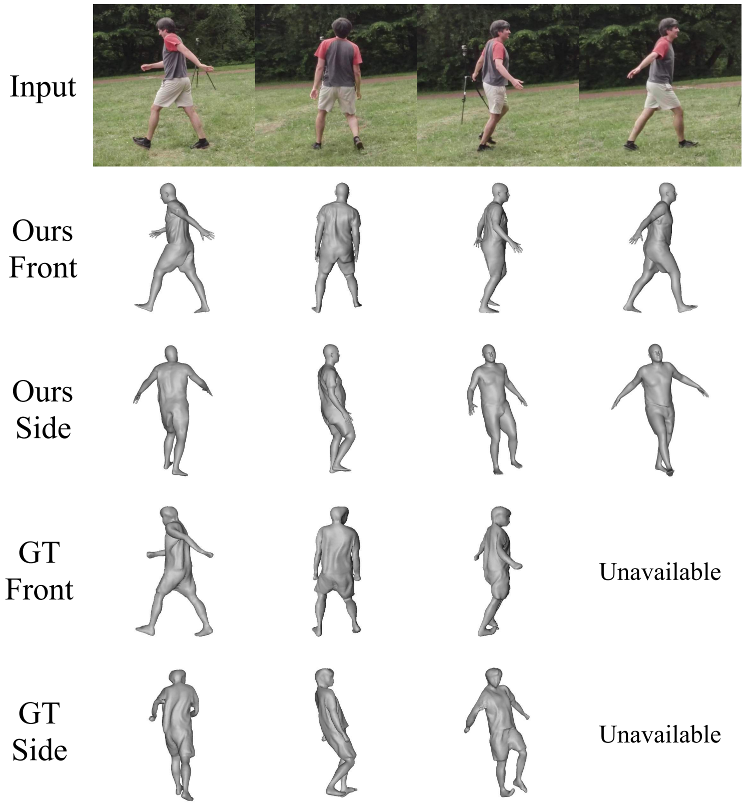

Figure 5.5: Visualization of experiment results on the *Pablo* sequence. From top to bottom we show original images, our results from the front and side views, and ground truth from the front and side views. The ground truth mesh for the last frame is not provided in the dataset.

obtained by manual segmentation on the ground truth surface mesh. The same procedure is applied to all the methods under evaluation. The mean results for all methods are reported in Table 5.1 (left) and the per-frame results are plotted in Figure 5.6. A visualization of our results is shown in Figure 5.5.

Evaluation of 3D Pose Estimation. Although body pose estimation is not a focus of this work, we follow [224] to validate our method on the metric of 3D joint error on the *Pablo* sequence. Average per-joint 3D position error after alignment with translation is reported in Table 5.1 (right), and per-frame results are reported in Figure 5.7. Our method achieves an error of 77.3 mm, significantly lower than 118.7 mm in [224]. This verifies the effectiveness of our body pose initialization that utilizes various image measurements including 2D joints, dense correspondences, silhouette, etc.

We also provide a quantitative comparison with recent state-of-the-art approaches that estimate 3D body pose with SMPL model from a monocular view on the *Pablo* sequence. The evaluation results are shown in Table 5.2. As a part of our pipeline, our estimation of 3D body pose is highly accurate even when compared with recent state-of-the-art approaches

Methods	Surface Error	Joint Error
MonoPerfCap* [224]	**14.6**	118.7
HMD [253]	31.9	-
Tex2Shape [4]	27.7	-
DeepHuman [250]	24.2	-
PIFu [167]	30.5	-
PIFuHD [168]	26.5	-
Ours	**17.9**	**77.3**

Table 5.1: Quantitative comparison with previous work on *Pablo* sequence using mean point-to-surface error and mean joint error across frames. All numbers are in mm. The method annotated with '*' uses a pre-scanned personalized template that provides a strong shape prior.

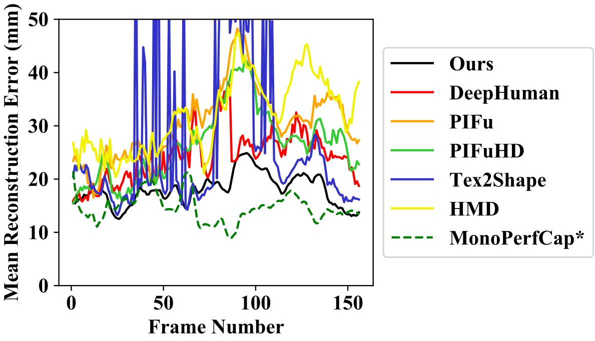

Figure 5.6: Per-frame results of the quantitative comparison with previous work on *Pablo* sequence using mean point-to-surface error. Notice that the method annotated with '*' uses a pre-scanned personalized template that provides strong shape prior.

that focus on 3D body pose only. This lays a solid foundation for the following clothing capture stages.

5.6.2 Evaluation on BUFF Dataset

Experiment Setting. BUFF [242] is a dataset of high-resolution 4D textured scan sequences of five people. In this experiment, we sample a test sequence from the BUFF dataset (00096-shortlong_hips, first 200 frames) and train a pair of upper and lower clothing models with the data of four other people. We render the sequence from three views: front, left and front-left, as visualized in Figure 5.8. We evaluate our method in four stages: body initialization, sequential tracking, batch optimization and wrinkle extraction The evaluation protocol is the same as Section 5.6.1: we rigidly align the estimated and ground truth meshes with a global scaling and translation, and compute average distance from ground truth clothing vertices to our results.

Results. The quantitative results are shown in Table 5.3 and Figure 5.9. First, from all three viewpoints, results with clothing consistently achieve lower reconstruction error than body only. This verifies that our method captures clothing shape that cannot be explained by the

Ours	Temporal HMR [88]	SPIN [96]	VIBE [94]
77.3	94.7	89.5	87.2

Table 5.2: Quantitative comparison with recent SMPL-based 3D body pose estimation approaches on the *Pablo* sequence. All numbers are in mm.

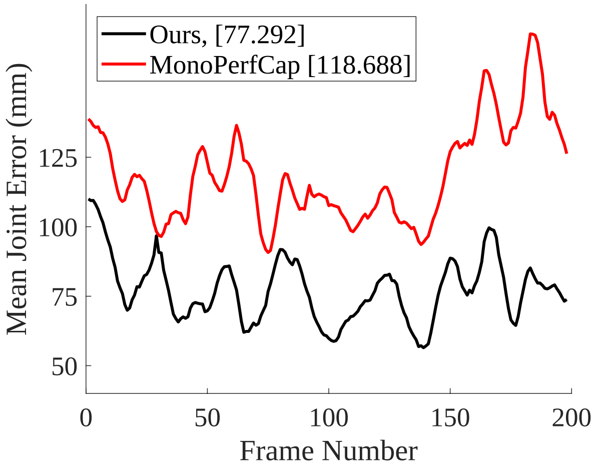

Figure 5.7: Per-frame results of the quantitative comparison with previous work on *Pablo* sequence using mean joint error.

SMPL body shape space. Second, we can see that temporal smoothing and wrinkle extraction, which improve the visual quality as shown in qualitative results, have little influence on the reconstruction error. Third, our results show similar range of error in the clothing region across different views, implying that our method is not very sensitive to the viewpoint variation.

5.7 Qualitative Evaluation

We qualitatively evaluate our method on various videos including public benchmark and in-the-wild videos where no pre-scanned template is available. Example results are shown in Figure 5.10. Please see our supplementary video[2] for full results and qualitative comparison with other work.

As shown in the supplementary video, our result not only demonstrates better temporal robustness than the single-image 3D human reconstruction methods in terms of reconstructed surfaces, but also provides 3D temporal correspondences effectively shown by the re-rendering of our output mesh with a consistent texture map. This is hard to obtain by methods that regress 3D shape in voxels [197, 250], depth maps [193] or implicit functions [137,167,168]. Template-based monocular performance capture methods [68,224] rely heavily on non-rigid surface regularization such as As-Rigid-As-Possible (ARAP), which

[2] https://drive.google.com/file/d/1Ne1GOcasMTypgFvAXi8fwjP5KYtjQ3JW/view?usp=share_link

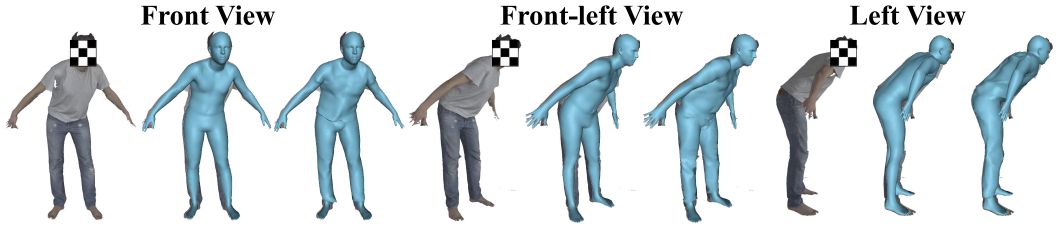

Figure 5.8: Visualization of three viewpoints in the BUFF evaluation. In each view we show the input image, body estimation result, and clothing capture result.

	Front	Front-left	Left
Body only	29.4	30.3	29.2
Clothed w/o batch	26.8	25.5	**24.7**
Clothed w/ batch	**26.7**	**25.3**	**24.7**
Clothed w/ wrinkle	26.8	25.5	24.9

Table 5.3: Quantitative ablation study of different stages of our method on the rendered BUFF dataset using mean point-to-surface error. All numbers are in mm.

often prevents those methods from capturing natural dynamics of the clothing deformation. In comparison, our method is able to capture more realistic dynamics of the garment with regularization provided by the clothing models.

5.7.1 Ablation Studies

In this section, we conduct more ablation studies on various loss terms we use in the energy optimization for clothing capture and body shape estimation.

Loss Terms for Clothing Capture

We first study the loss terms used for clothing capture in Section 5.5.2 (Equation 5.12). In the experiments below, we compare the results of the batch optimization stage with different loss terms, initialized from the same body capture and sequential tracking results.

Clothing segmentation term (Equation 5.13). In order to study the effect of the clothing segmentation term, we run an ablative experiment where the weight for the segmentation term is set to 0, while all other terms remain the same. To better visualize the effect, we render the output meshes in three colors: grey for skin, yellow for upper clothing and green for lower clothing. We consider a vertex j as a skin vertex if the length of the clothing offset for this vertex is below a certain threshold ε, or

$$\|D_j\| < \varepsilon,$$

where D_j is defined in Equation (5.4). We consider a vertex as belonging to the upper clothing if

$$\|D_j^u\| \geq \|D_j^l\| \quad \text{and} \quad \|D_j\| \geq \varepsilon,$$

or, similarly, as belong to the lower clothing if

$$\|D_j^l\| > \|D_j^u\| \quad \text{and} \quad \|D_j\| \geq \varepsilon.$$

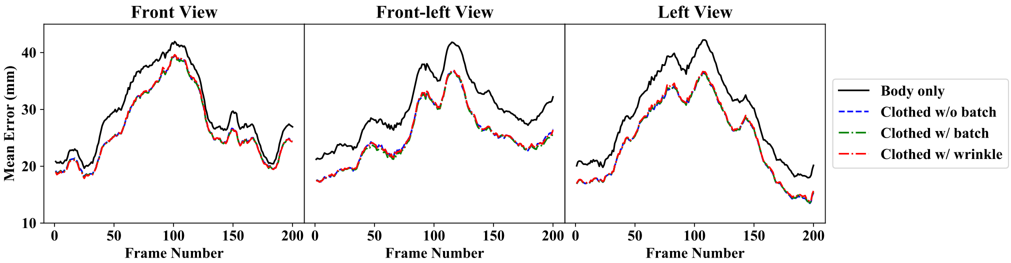

Figure 5.9: Per-frame results of the quantitative ablation study for different stages of our method on rendered BUFF dataset using mean point-to-surface error.

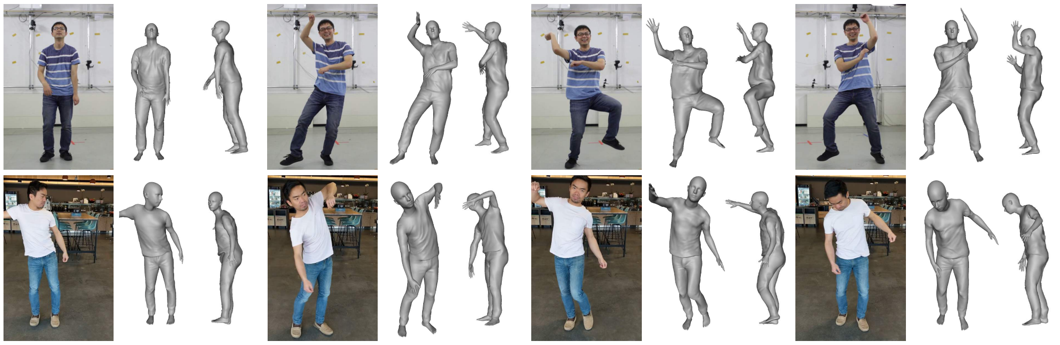

Figure 5.10: Examples of our clothing capture results. In each example, we show the input image, capture results from the front view and the side view. Please see our supplementary video for complete results.

The result of this experiment is shown in Figure 5.11. In each frame, we observe that the boundary between the upper and lower clothing is *more consistent* with the original image in the result with segmentation term than the result without segmentation term. Our method adopts a combination of upper clothing and lower clothing models, which might both have non-zero offsets around the body waist. It is important for our method to produce both offsets with correct relative length to realistically reconstruct the spatial arrangement of the T-shirt and trousers in the original images. This result proves the effectiveness and necessity of the clothing segmentation term.

Photometric tracking term (Equation 5.13). Similarly, we run an ablative experiment where the weight for the photometric tracking term is set to 0 and other terms remain the same. To visualize its effect, we render the output tracked mesh with the final texture extracted in the sequential tracking stage (see Section 5.5.2 for detail), and compare the results with and without the photometric tracking term with the original images.

The result of this experiment is shown in Figure 5.12. Notice that the same final texture image is used to render all the results. In order to assist visual comparison of the rendered pattern, we draw several auxiliary horizontal dashed lines in red. We can observe that the results with photometric tracking term is more consistent with the original image than the result without photometric tracking term, in terms of the location of the white strip on the T-shirt and the boundary between the T-shirt and trousers. This demonstrates that our

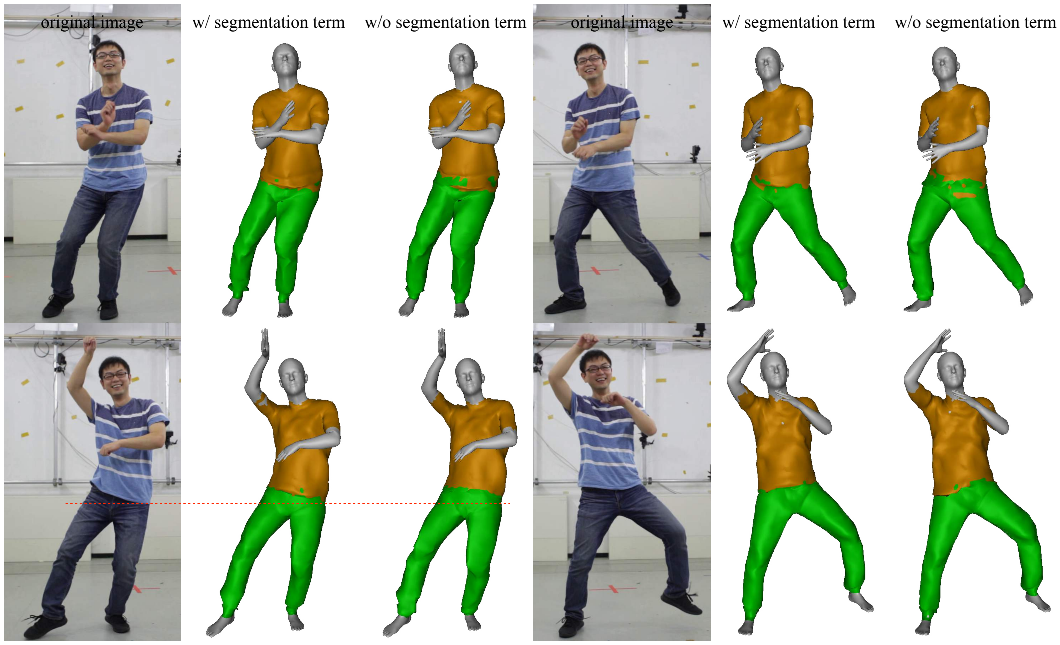

Figure 5.11: Comparison between results with and without clothing segmentation loss. The vertices for skin, upper clothing and lower clothing are rendered in grey, yellow and green respectively. A horizontal red dashed line is drawn in the bottom left example to help visually check the location of the boundary between upper and lower clothing.

photometric tracking loss can help to obtain better temporal correspondence across different frames in the video.

Silhouette matching term. We now compare the results with and without the silhouette matching term. We render both results and align them with the original images to visualize how well the silhouette matches.

The result of this experiment is shown in Figure 5.13. We observe that the result with silhouette matching term achieves a better alignment of silhouette with the original image. This suggests that the silhouette matching term can help to reconstruct the accurate shape of the clothing in the video.

Losses Terms for Body Shape Estimation

Although body pose and shape estimation is not a focus of this work, we conduct ablative studies on the loss terms used in body shape estimation in Section 5.5.1 (Equation 5.9). In each of the experiment in this section, the weight for the loss term under study is set to 0, and all other terms stay the same as the full results. We render the estimated body shapes and compare them with the full results.

Silhouette term. The result of this experiment is shown in Figure 5.14. We can observe in the result that silhouette provides critical information for the estimation of body shape and pose in the following two ways. First, the projection of human body should always lie in the interior of the overall silhouette in the image, which includes the region of body

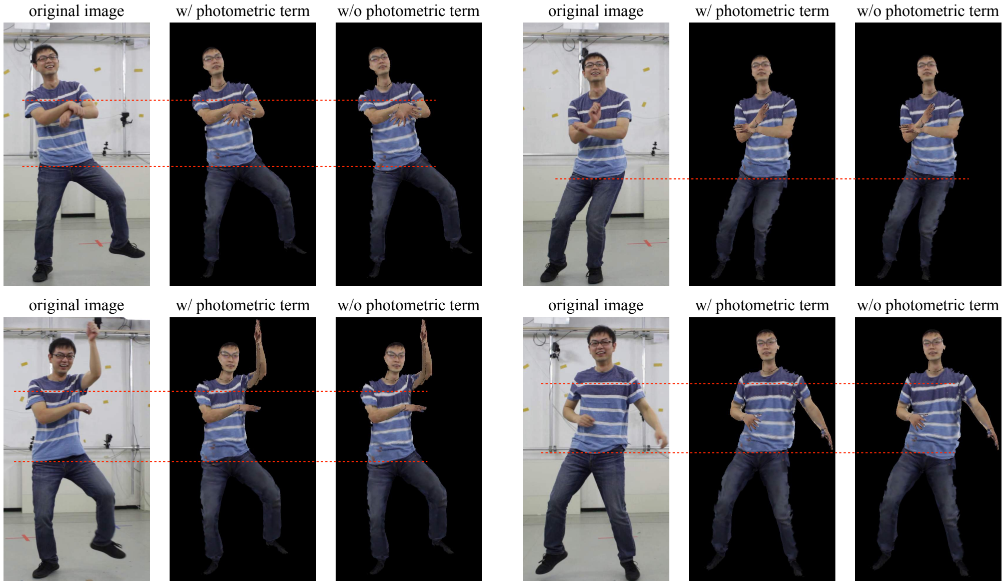

Figure 5.12: Comparison between results with and without photometric tracking loss. Horizontal dashed lines are drawn in red to help visually compare the location of rendered texture pattern.

Table 5.4: Average per-frame runtime of each stage in our pipeline. The numbers are in seconds.

Stage	Runtime (s)
Body Estimation (Section 5.5.1)	6
Sequential Tracking (Section 5.5.2)	62
Batch Optimization (Section 5.5.2)	27
Wrinkle Extraction (Section 5.5.3)	232
Total	327

and clothes. Second, in the top-right and bottom-left examples, an arm of the subject is occluded by the torso. There is no available information to reason about the location of the arm from the 2D keypoints or DensePose results. In this situation, only the silhouette can constrain the position of the arm to be behind the torso in the camera view. This proves the importance of the silhouette term for accurate estimation of human body and shape.

DensePose term. The result of this experiment is shown in Figure 5.15. The use of DensePose together with SMPL model for accurate body estimation was first proposed in [63]. In our work, we find that the DensePose term helps to estimate the hand orientation more accurately, as fingers are usually not included in the hierarchy of 2D body pose output.

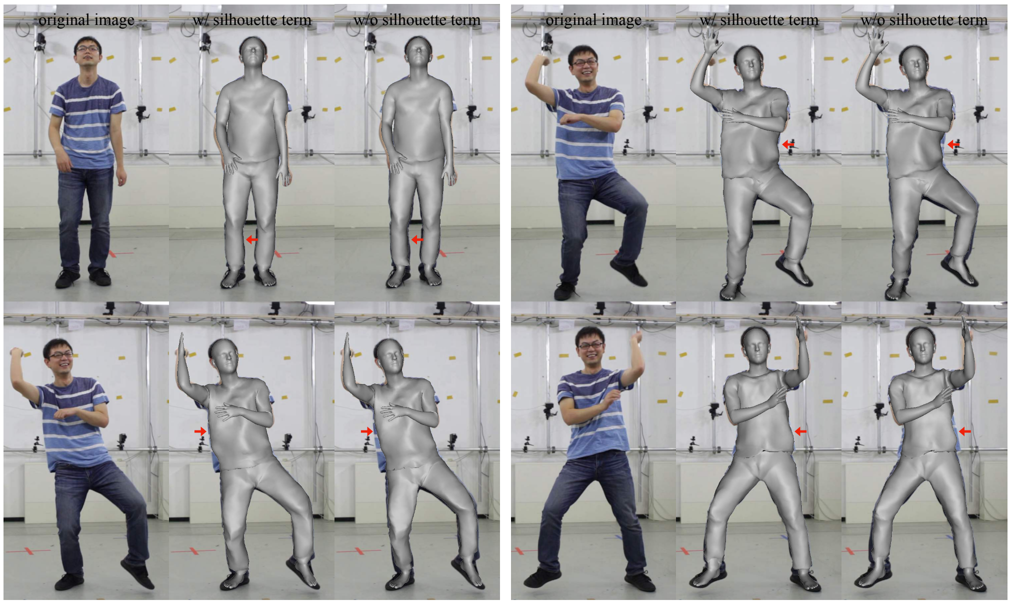

Figure 5.13: Comparison between results with and without silhouette matching loss. In each example, we show the original image, the result with and without silhouette matching loss from left to right.

5.7.2 Runtime Analysis

In this section, we present the runtime information of our approach. Our method runs on a Linux server with 40 CPU cores and 4 GTX TITAN X GPUs. Our approach requires the memory of 4 GPUs in order to run the batch optimization on a video of around 250 frames together. For optimization, we use the L-BFGS solver implemented in PyTorch [145]. We measure the average time consumed for each frame in every stage, and the results are shown in Table 5.4.

5.8 Conclusion and Future Work

In this chapter, we have presented a method to capture temporally coherent dynamic deformation of clothing from a monocular video. To the best of our knowledge, we have shown the first result of temporally coherent clothing capture from a monocular RGB video without using a pre-scanned template. Our results on various in-the-wild videos endorse the effectiveness and robustness of our method.

Our method is limited by the types of garments in the available training data. We have demonstrated results on several types of tight clothing. Treatment of free-flowing garments like skirts requires collection of more data and additional design of the clothing model. Our method is constrained in the ability to capture drastically changing deformations due to the limited expressiveness of our models, which may be addressed by using higher-capacity models like a deep neural network. We also would like to further incorporate physics into the clothing models to enable more physically realistic clothing capture.

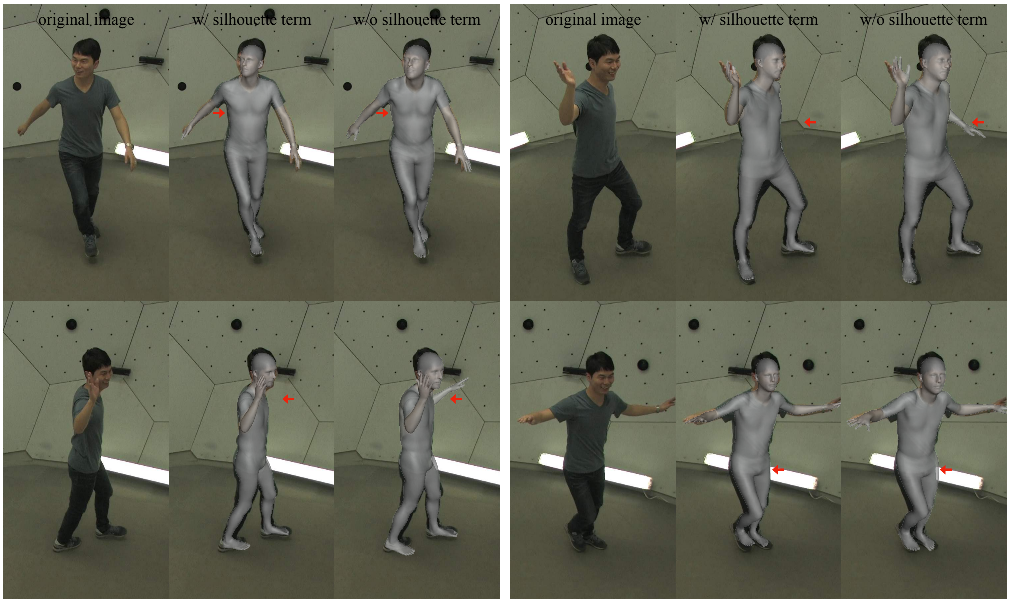

Figure 5.14: Comparison between results with and without silhouette loss. In each example, we show the original image, the result with and without silhouette loss from left to right.

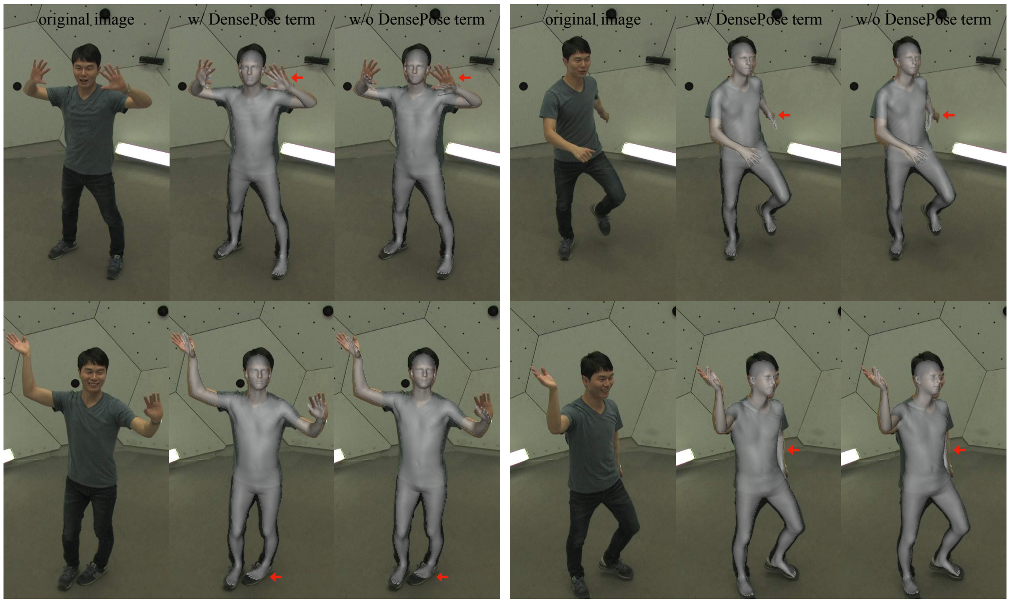

Figure 5.15: Comparison between results with and without DensePose loss. In each example, we show the original image, the result with and without DensePose loss from left to right.

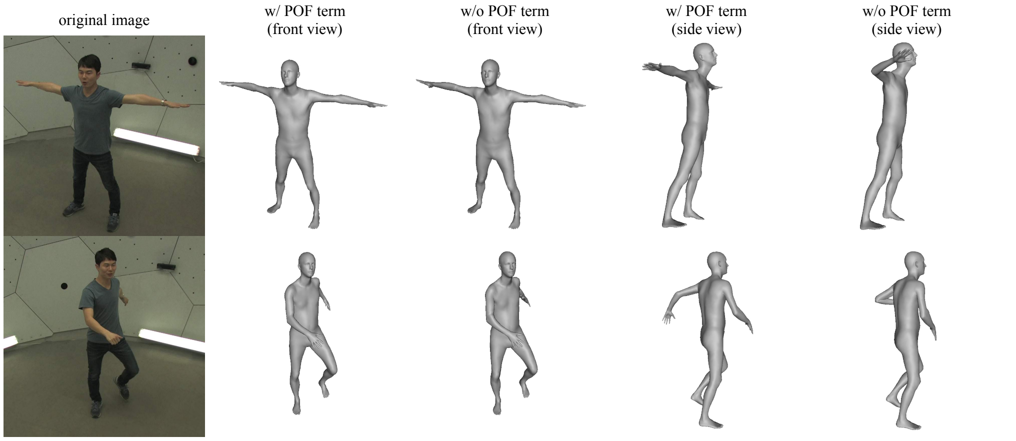

Figure 5.16: Comparison between results with and without POF loss. In each example, we show the original image, the result with and without POF loss from both the front view and the side view.

Chapter 6

Faithful Clothing Telepresence Driven by Sparse RGB-D Input

6.1 Introduction

Photorealistic avatars are important for enabling truly immersive and believable telepresence experiences. An ideal telepresence application should not only produce plausible-looking results, but also be complete and accurate: all salient aspects of human appearance should be captured and resynthesized to fully match the real-world states of the subject. However, these properties are particularly challenging to achieve with clothing, an integral part of human appearance, due to its complex movements on human body.

To deal with this challenge, some previous methods go beyond treating clothing effectively as a part of the human body [7, 115] and perform explicit modeling of clothing as a separate layer on top of the underlying body (Chapter 3 and 4). These methods can work well for pose-driven animation, i.e., synthesizing plausible clothing deformation and photo-realistic appearance that are perceptually compatible with the input pose signal. However, there is no guarantee that the animation output will faithfully reproduce the actual states of clothing (Figure 6.8), and potentially distorting the conveyed social signals. In fact, the dynamics of clothing cannot be fully explained by the body pose of the current frame or a few previous frames. Given two distinct initial states of clothing, the same body motion can result in completely different trajectories of clothing deformations, especially for loose garments like skirts or dresses. Therefore, it is impossible to infer accurate clothing states given such incomplete input signals.

An alternate approach for telepresence relies heavily on the availability of sensory inputs without a strong human prior. For example, volumetric fusion methods [49, 50, 138] produce a complete geometrical representation of a scene by tracking and fusing observations from sparse RGB-D cameras. Neural implicit functions can also be utilized to reconstruct a dynamically moving human surface from sparse camera inputs [174, 237], or even to directly model the radiance field of clothed human appearance [113, 175]. In theory, these methods are flexible enough to be able to reconstruct arbitrary shape from the given input streams. However, due to a lack of model constraints, it is generally more challenging for these methods to achieve high-fidelity temporal coherency especially with noisy or incomplete input, and the output quality is heavily tied with the sensory input. For example, it

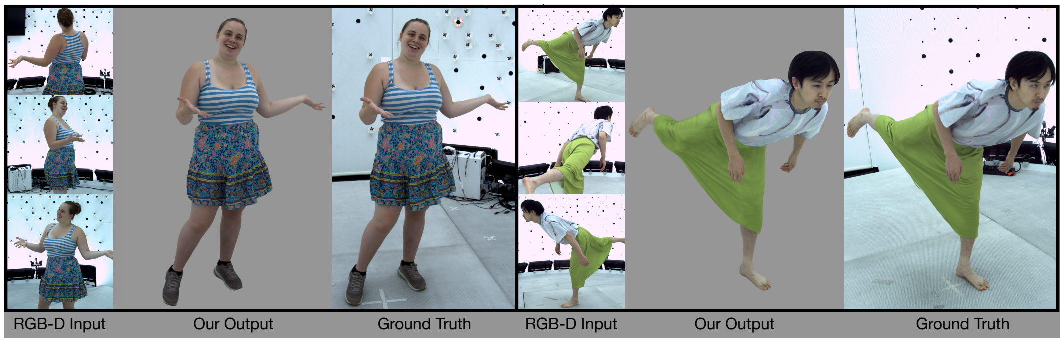

Figure 6.1: We present photorealistic full-body avatars that can be driven by sparse RGB-D views (along with body pose and facial keypoints) and faithfully reproduce the appearance and dynamics of challenging loose clothing from the input views. We show the input views, our output and the ground truth reference images in each group of results.

is hard to produce the sharp and subtle detail in hands [175] with the limited resolution and noise, and the observed and unobserved regions from the input can have obvious difference in the output quality [174]. Human priors have been introduced to regularize the predictions [97, 98], but the ability to reliably handle loose clothing has not been clearly demonstrated.

To leverage the benefits of both families of approaches, we can rely on explicit avatar models as a prior, but expand the driving signal to include the denser input in addition to the body pose. We build avatars with dynamic clothing that can be driven from a sparse set of RGB-D cameras (usually three unless otherwise stated). This formulation allows for more faithful resynthesis of the human appearance, including clothing details. We build on top of DVA [161], which proposes the texel-conditioned avatar, an encoder-decoder model that takes in UV-aligned driving features and predicts geometry and appearance for rendering. However, DVA only works well for tight clothing that closely follows underlying body, due to the limitation in relying on a generic body shape prior.

To better handle loose clothing, our insight is to introduce a tracking stage that coarsely aligns the loose clothing surface with the input depth. More specifically, we propose a simple-yet-effective Neural Iterative Closest Points (N-ICP) algorithm to iteratively update a clothing deformation model given the feedback from surface error in a data-driven manner. N-ICP combines the flexibility of the classical ICP methods which allows us to handle large clothing deformations, while relying on learning for more efficient inference and reliable geometry estimates. In contrast to DVA, which uses coarse body geometry to extract features, the N-ICP tracking allows us to extract more accurate and meaningful texel-aligned features. It also eases the burden on the encoder-decoder model, since large deformations and misalignments are handled by the coarse tracking, and ultimately leads to better quality and generalization. In addition, several technical components have been leveraged to further improve the texel-conditioned avatars. To aid geometry prediction, we augment texel-aligned features with geometry features computed from depth and coarsely tracked geometry. To improve appearance, we adopt a specific perceptual loss to encourage high-frequency texture detail on the predicted clothing.

Our contributions can be summarized as follows:

- We develop photorealistic full-body avatars with dynamic clothing that faithfully reproduces the original states of subject's appearance and geometry. These avatars are driven by sparse (up to 3) RGB-D inputs and enable free-viewpoint rendering.

- As an important component of our system, we introduce a Neural Iterative Closest Point algorithm that learns to iteratively find the most effective parameter update to track the input point cloud efficiently with a deformation model.
- Our avatars can directly generalize to a novel testing environment with a different background and illumination, while capturing the complex clothing dynamics and preserving the original high-quality appearance from the training data. We also provide an option to finetune the model to adapt to the new appearance in the novel environment.

This work is published as a conference paper at SIGGRAPH Asia 2023.

6.2 Related Work

Photorealistic Clothed Avatars. Besides the work on full-body clothed avatars already listed in Section 2.1.2, Drivable Volumetric Avatars (DVA) [161] is a special one, which is additionally conditioned on a dynamic texture unwrapped from sparse driving views to incorporate the clothing information. However, it is limited to tight clothing due to a constraint in body representation. Our method can further track and render dynamically moving loose clothing realistically.

Sensing-Based Telepresence Approaches. Another category of approaches for telepresence rely more heavily on the sensing input for surface reconstruction, usually from one or several RGB(-D) cameras. Volumetric fusion [49, 50, 138] and multiview-conditioned implicit functions [174, 189, 237] have been utilized to reconstruct scene geometry from the sensor input. Early work only reconstructs the geometry [138], but later work also predicts color by warping and blending RGB information from input views [50, 101], possibly assisted by deep neural networks [174, 189, 237]. Neural rendering has also been introduced [131, 139] to compensate for artifacts in the reconstructed geometry. These sensing-based methods enjoy the flexibility in handling varying topology, but are generally more sensitive to noisy or missing input than the model-based approaches described in the previous section.

Image-Conditioned Novel-View Synthesis. Our task can also be regarded as Novel-view Synthesis (NVS) based on sparse input images. Generalizable NeRF is a group of methods that extend the Neural Radiance Field (NeRF) [135] and allow the reconstruction of a scene with sparse images as input without per-scene optimization [34, 113, 175, 204, 236]. This formulation is thus more suitable for telepresence than the original NeRF, but tends to perform less well when the target view is far away from the sparse input views due to the inherent 3D ambiguity. To alleviate this problem, some methods incorporate prior knowledge of the human body to achieve better quality. Neural Body [151] utilizes the SMPL model to aggregate the temporal information over the multiview videos. It still does not allow direct use of novel images as input, but this limitation is addressed in Neural Human Performer [97] and GP-NeRF [35]. Several other methods [57, 58, 98, 247] predict radiance fields in a canonical space with the help of forward and inverse skinning transformations.

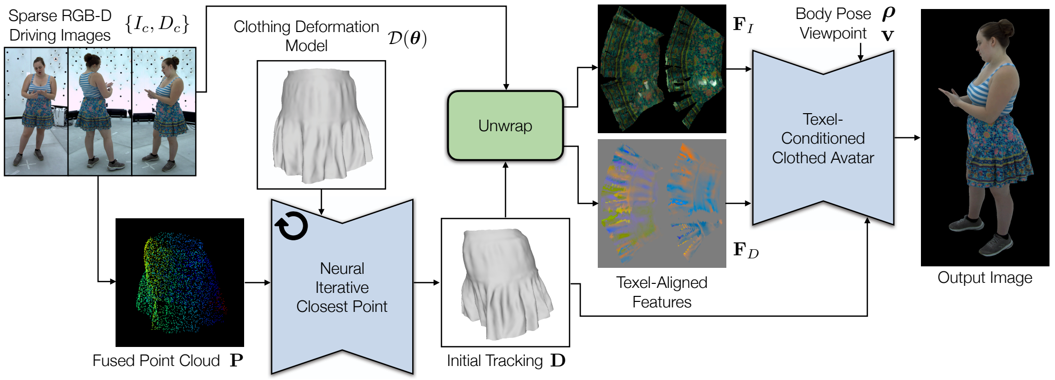
Sparse RGB-D Driving Images $\{I_c, D_c\}$
Clothing Deformation Model $\mathcal{D}(\boldsymbol{\theta})$
Unwrap
$\mathbf{F}_I$
Body Pose $\boldsymbol{\rho}$
Viewpoint $\mathbf{v}$
Texel-Conditioned Clothed Avatar
Neural Iterative Closest Point
$\mathbf{F}_D$
Texel-Aligned Features
Output Image
Fused Point Cloud $\mathbf{P}$
Initial Tracking $\mathbf{D}$

Figure 6.2: An overview of our method. The Neural Iterative Closest Point module efficiently tracks the clothing surface from the input point cloud $\mathbf{P}$ with a clothing deformation model $\mathcal{D}(\boldsymbol{\theta})$; the initial tracking result $\mathbf{D}$ is then used to unwrap the driving images I_c and depth maps D_c into texel-aligned features $\mathbf{F}_I, \mathbf{F}_D$, which are then fed into the texel-conditioned avatar, together with body pose $\boldsymbol{\rho}$, facial keypoints and viewpoint $\mathbf{v}$, to predict the output image.

These methods additionally allow generalization across identities, which is not the focus of our work. KeypointNeRF [134] utilizes 3D body keypoints to encode the spatial information in the rendered volume. These methods have shown limited capability to handle dynamic loose clothing due to the body representation.

Learning to Optimize for Non-Rigid Tracking. Many non-rigid tracking and reconstruction problems are traditionally formulated as optimization. Examples include SMPL model fitting for 3D human pose estimation [16], and non-rigid ICP for deformable surface tracking [66]. To improve the speed and robustness of tracking in the presence of noisy data, Supervised Descent Method [221] and Discriminatory Optimization [200] have been proposed to generate iterative parameter update from a learned linear transformation of handcrafted features. In recent years, researchers attempted to incorporate deep neural networks into these problems [12, 20, 21, 109]. Our N-ICP formulation essentially treats the neural network as an optimization solver that iteratively generates a parameter update. Some work [41, 60, 180, 240] explores a similar idea for monocular human pose reconstruction in a supervised setting. RMA-Net [53] also addresses the non-rigid registration problem with recurrent update. In addition to the difference in deformation model and loss function, our method integrates learning into optimization more deeply by explicit feeding the error and gradient into the network for update prediction, which is shown to be the key to effectiveness in the ablation study.

6.3 Method Overview

Our method takes as input RGB-D images from sparse (up to 3) views, as well as 3D body pose in the form of joint angles (and facial keypoints if available), and generates photorealistic rendering of the subject from an arbitrary viewpoint. The model is trained on images of the subject captured in a dense camera system. We adopt the two-layer representation that has proven effective in previous work on loose clothing (Chapter 3 [218] and Chapter 4 [216]). Our method consists of two major modules. First, in the Neural Iterative Closest Point module, we coarsely track the loose clothing surface given the fused point cloud from the input depth maps using a deformation graph model. Second, we convert the sparse driving RGB-D images into texel-aligned features and feed them into texel-conditioned clothed avatars to infer detailed geometry and view-dependent texture, which are then rasterized to form the output image. The overall pipeline is illustrated in Figure 6.2.

6.4 Neural Iterative Closest Point

As the first step of our approach, we introduce a Neural Iterative Closest Point (N-ICP) algorithm to coarsely track the dynamic clothing surface using a deformation graph representation of the clothing geometry. Such a module is needed for two reasons. First, coarse tracking provides rough correspondences on the clothing surface across different frames. Previous work (Chapter 3 [218] and Chapter 4 [216]) shows that such canonicalization reduces the variance in appearance that needs to be modelled by the downstream module and leads to improved quality. Second, compared with skeleton-level tracking [161], the deformation graph enjoys the flexibility to track the surface at a higher accuracy from the input depth, so that the following stage (Section 6.5) only needs to predict a small geometry correction. This concept of coarse-to-fine modeling has also proven useful in previous work, e.g. [67].

Non-Rigid ICP. The classical approach to track a deforming surface is the non-rigid Iterative Closest Point (ICP) algorithm [103]. Given a deformation model $\mathcal{D}$ controlled by deformation parameters $\boldsymbol{\theta}$, the non-rigid ICP algorithm tracks an input point cloud[1] $\mathbf{P}$ by solving the following minimization problem

$$\min_{\boldsymbol{\theta}} L_{\text{ICP}}(\boldsymbol{\theta}; \mathbf{P}) = \min_{\boldsymbol{\theta}} \sum_{\mathbf{p} \in \mathbf{P}} \|\mathcal{C}(\mathbf{p}, \mathcal{D}(\boldsymbol{\theta})) - \mathbf{p}\|^2, \tag{6.1}$$

where $\mathcal{C}$ queries the closest point on the deformed mesh $\mathcal{D}(\boldsymbol{\theta})$ from a point $\mathbf{p}$ in the point cloud based on Euclidean or projective distance. To keep the deformed mesh in a smooth shape, some regularization terms are often used together with Equation (6.1) to penalize extreme distortion. This non-linear optimization problem is usually solved by the Gauss-Newton method, especially its Levenberg–Marquardt variant. Concretely, in each iteration of optimization, the algorithm evaluates the residual vector $\mathbf{r}$, which consists of offset $\mathbf{r}(\mathbf{p}) = \mathcal{C}(\mathbf{p}, \mathcal{D}(\boldsymbol{\theta})) - \mathbf{p}$ for each point $\mathbf{p}$ in the point cloud, and the Jacobian $\mathbf{J}$ of each residual term with respect to the parameters. Then the algorithm searches for the update $\Delta\boldsymbol{\theta}$ by solving the linear system $(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})\Delta\boldsymbol{\theta} = -\mathbf{J}^T\mathbf{r}$, where $\lambda > 0$ is a damping coefficient. The parameters are updated by $\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} + \Delta\boldsymbol{\theta}$, where i denotes the index of iteration.

N-ICP. The problem of driving avatars for online telepresence poses a challenge in terms of both robustness and speed for the non-rigid tracking algorithm. The nonlinear minimization problem in Equation (6.1) relies heavily on good initialization, so classical method

[1] At test time, the point cloud comes from fused driving depth images. We only use the points from the regions of dynamic clothing identified by image segmentation.

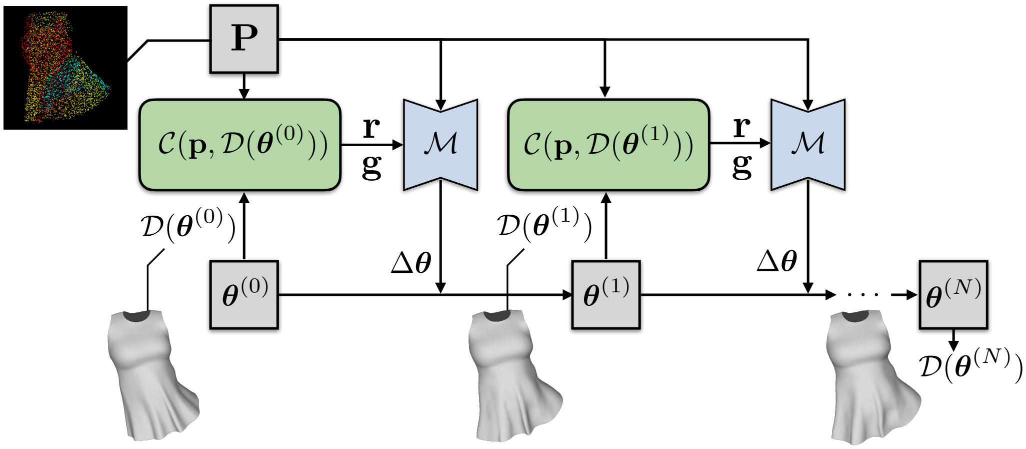
$\mathbf{P}$
$\mathcal{C}(\mathbf{p}, \mathcal{D}(\boldsymbol{\theta}^{(0)}))$
$\mathbf{r}$
$\mathbf{g}$
$\mathcal{M}$
$\mathcal{C}(\mathbf{p}, \mathcal{D}(\boldsymbol{\theta}^{(1)}))$
$\mathbf{r}$
$\mathbf{g}$
$\mathcal{M}$
$\mathcal{D}(\boldsymbol{\theta}^{(0)})$
$\Delta\boldsymbol{\theta}$
$\mathcal{D}(\boldsymbol{\theta}^{(1)})$
$\Delta\boldsymbol{\theta}$
$\boldsymbol{\theta}^{(0)}$
$\boldsymbol{\theta}^{(1)}$
$\cdots$
$\boldsymbol{\theta}^{(N)}$
$\mathcal{D}(\boldsymbol{\theta}^{(N)})$

Figure 6.3: The Neural Iterative Closest Point algorithm. In each iteration, we perform the closest point query between the input point cloud $\mathbf{P}$ and deformed clothing model $\mathcal{D}(\boldsymbol{\theta}^{(i)})$. The residual $\mathbf{r}$ and gradient $\mathbf{g}$ is passed to a neural network $\mathcal{M}$, which finds the best update $\Delta\boldsymbol{\theta}$ to the parameters. This process is repeated for N iterations.

usually requires sequential tracking that is hard to recover from failure and technically demanding GPU implementation to meet the runtime constraint [138, 255].

This challenge motivates us to introduce a Neural Iterative Closest Point (N-ICP) algorithm, where the goal is to leverage the prior learned by a deep neural network to make an efficient and robust prediction of the update direction $\Delta\boldsymbol{\theta}$. We utilize a PointNet [159] architecture to operate on the input point cloud $\mathbf{P}$. In addition to the coordinates of each point $\mathbf{p}$, we also use the offset $\mathbf{r}(\mathbf{p}) = \mathcal{C}(\mathbf{p}, \mathcal{D}(\boldsymbol{\theta})) - \mathbf{p}$ as a feature, which includes the essential closest point information for solving the non-rigid ICP problem. Inspired by the classical optimization paradigm, we also a provide first-order derivative to the network. For the ease of computation, we use the gradient $\mathbf{g} = \mathbf{J}^T\mathbf{r}$ of total loss with respect to the parameters, which can be automatically derived in most modern neural network libraries [145], or manually derived for better computation efficiency. The formulation of the network $\mathcal{M}$ can be written as $\Delta\boldsymbol{\theta} = \mathcal{M}(\mathbf{P}, \mathbf{r}, \mathbf{g})$. The update $\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} + \Delta\boldsymbol{\theta}$ is then performed for $N = 3$ times.

Weakly-Supervised Learning. We train the network in a *weakly-supervised* manner without requiring the ground truth deformation parameters $\boldsymbol{\theta}$. Instead, we only use the clothing geometry $\overline{\mathbf{P}}$ reconstructed by Multi-View Stereo (MVS). Compared with $\mathbf{P}$ which has missing parts due to occlusion from the sparse depth input, $\overline{\mathbf{P}}$ includes the complete clothing geometry reconstructed from all cameras in the capture studio, and is only used at training time for supervision. Our loss function for training the network is written as $L_{\text{N-ICP}} = \frac{1}{N}\sum_{i=1}^{N} L_{\text{ICP}}(\boldsymbol{\theta}^{(i)}, \overline{\mathbf{P}})$ plus regularization (see Section 6.7.2). In other words, the network is trained to find an update to the parameters so that the deformed mesh can most efficiently track the clothing surface for every ICP iteration.

Clothing Deformation Model. ICP-style methods require good initialization to converge to the right local minimum. We observe that the underlying body pose can provide coarse information about body orientation and articulation as a good starting point. Thus we adopt a hierarchical deformation model for dynamic loose clothing: an outer layer of Linear Blend Skinning (LBS) $\mathcal{W}$ with respect to body pose, and an inner layer of deformation graphs $\mathcal{E}$ [187]: $\mathcal{D}(\boldsymbol{\theta}) = \mathcal{W}(\mathcal{E}(\mathbf{M}, \boldsymbol{\theta}), \boldsymbol{\rho})$, where $\mathbf{M}$ is the template shape of the clothing defined in the canonical space of LBS. The underlying body pose $\boldsymbol{\rho}$ can be obtained from sparse RGB-D input by vision-based keypoint detection followed by inverse kinematics [132], and it remains fixed during the N-ICP process. In this formulation, $\boldsymbol{\theta}$ is defined as the rotation and translation of deformation graph nodes, and we set its initialization to be the rest pose $\boldsymbol{\theta}^{(0)} = \mathbf{0}$. Thanks to the global transformation and body articulation encoded in $\boldsymbol{\rho}$, $\mathcal{D}(\boldsymbol{\theta}^{(0)}) = \mathcal{W}(\mathbf{M}, \boldsymbol{\rho})$ is close enough to the target $\mathbf{P}$ for N-ICP to converge nicely. This formulation also makes it efficient to perform per-frame tracking, preventing the failure caused by error accumulation in sequential tracking. The complete N-ICP process is illustrated in Figure 6.3.

6.5 Texel-Conditioned Clothed Avatars

The N-ICP algorithm in Section 6.4 can track large clothing dynamics, providing a good starting point for rendering the clothed body appearance. However, the underlying deformation model is designed to only capture large geometry deformations, and does not model fine geometrical detail or appearance. Therefore, in the next step, we develop a clothed avatar that can produce high-fidelity geometry and appearance when conditioned on both sparse RGB-D views as well as the output from the previous N-ICP stage. The critical question here is how to *faithfully* reconstruct the appearance detail from the sparse driving views for dynamic clothing.

To achieve this goal, we build upon the texel-conditioned avatars from Drivable Volumetric Avatars (DVA) [161]. DVA takes in driving signals of several RGB images mapped to texel-aligned features, as opposed to conditioned primarily on pose [7], and predicts geometry and view-dependent appearance that can reproduce the full-body appearance. Formally, in DVA, the input feature $\mathbf{F}_I^b$ is the mean unwrapped image from multiple RGB driving views based on the skinned mesh $\mathbf{W}_{\boldsymbol{\rho}}^b = \mathcal{W}(\mathbf{M}^b, \boldsymbol{\rho})$ of the body template $\mathbf{M}^b$. This unwrapping process can be written as

$$\mathbf{F}_I^b = \mathcal{U}(\{I_c\}, \mathbf{W}_{\boldsymbol{\rho}}^b), \tag{6.2}$$

where $\mathcal{U}$ denotes the unwrapping operation and $\{I_c\}$ denotes the set of driving images. The neural avatar $\mathcal{A}^b$ is a convolutional encoder-decoder that takes in $\mathbf{F}_I^b$, viewpoint $\mathbf{v}$ and body pose $\boldsymbol{\rho}$, and predicts the geometry corrective $\delta\mathbf{G}^b$ and appearance $\mathbf{T}^b$ with $[\delta\mathbf{G}^b, \mathbf{T}^b] = \mathcal{A}^b(\mathbf{F}_I^b, \mathbf{v}, \boldsymbol{\rho})$. The final geometry $\mathbf{G}^b$ is obtained by a function $\mathcal{G}$ that applies $\delta\mathbf{G}^b$ on top of the LBS mesh $\mathbf{W}_{\boldsymbol{\rho}}^b$ with a pre-defined coordinate transformation

$$\mathbf{G}^b = \mathcal{G}(\mathbf{W}_{\boldsymbol{\rho}}^b, \delta\mathbf{G}^b), \tag{6.3}$$

which is then used to render[2] the output image together with the view-conditioned appearance $\mathbf{T}^b$.

[2] The geometry $\mathbf{G}$ and appearance $\mathbf{T}$ may take different forms depending on whether the Mixture of Volumetric Primitive (MVP) [119] or mesh-texture formulation [117] is adopted as the rendering model, but the high-level concepts are similar and described here in a unified manner. The original DVA method [161] uses MVP, while we use mesh and texture.

Texel-Conditioned Clothing Avatars. The DVA baseline, however, struggles to handle the large deformation of dynamic clothing. The root of the problem is that the LBS mesh $\mathbf{W}^b_{\boldsymbol{\rho}}$, which encodes only the skeleton-level motion, is too coarse to serve as the base geometry for dynamic clothing. The large deviation of the LBS mesh $\mathbf{W}^b_{\boldsymbol{\rho}}$ from the true clothing surface has two consequences: first, the unwrapping operation in Equation (6.2) cannot effectively capture the appearance detail; second, it places a heavy burden for the network $\mathcal{A}^b$ to predict a large offset $\delta\mathbf{G}^b$ to update the geometry in Equation (6.3).

One of the key ideas of our clothed avatar model is to use the non-rigid tracking result $\mathbf{D} = \mathcal{D}(\boldsymbol{\theta}^{(N)})$ from the final N-ICP iteration as the starting point. Because $\mathbf{D}$ is already well-aligned with the clothing surface, we can obtain better texel-aligned features with more appearance detail from the unwrapping operation $\mathbf{F}_I = \mathcal{U}(\{I_c\}, \mathbf{D})$. In order to further guide the estimation of the geometry corrective using the driving depth images, we also provide a "depth offset" feature $\mathbf{F}_D$ as input. For each pixel $[x, y]$ in camera c with depth value D_c, we associate it with the rendered depth d_c from the tracked geometry $\mathbf{D}$ at the same pixel location and compute the offset as

$$O_c[x, y] = \mathbf{R}_c(d_c[x, y] - D_c[x, y])\mathbf{K}_c^{-1} \begin{bmatrix} x & y & 1 \end{bmatrix}^T + \mathbf{t}_c, \tag{6.4}$$

where $\mathbf{K}_c$ is the camera projection matrix, and $\{\mathbf{R}_c, \mathbf{t}_c\}$ are the rigid transformation from each camera frame to a unified body root coordinate frame. The depth offset feature is then a 3-channel average tensor obtained by unwrapping $\{O_c\}$ for each driving camera c, $\mathbf{F}_D = \mathcal{U}(\{O_c\}, \mathbf{D})$. Thus, $\mathbf{F}_D$ can be regarded as the offset to be corrected on top of $\mathbf{D}$ in order to match the sensor depth for each UV location. Our clothing model takes in the texel-aligned features $\mathbf{F}_I$ and $\mathbf{F}_D$ as well as the viewpoint $\mathbf{v}$, and predicts the geometry corrective $\delta\mathbf{G}$ and texture $\mathbf{T}$ with $[\delta\mathbf{G}, \mathbf{T}] = \mathcal{A}(\mathbf{F}_I, \mathbf{F}_D, \mathbf{v})$. Finally, the geometry $\mathbf{G}$ and texture $\mathbf{T}$ are used to render the output image. Following Chapter 3 [218], the clothing is modelled as a separate layer from the underlying body avatar. The body avatar can be conditioned on body pose $\boldsymbol{\rho}$ or additionally on texel-aligned features [161]. To train the model, besides the standard image reconstruction loss and mesh regularization, we use a part segmentation loss and an ID-MRF perceptual loss [207], which are detailed in Section 6.7.2.

6.6 Results

For best quality please see our supplementary video[3].

6.6.1 Capture Setup and Detail

We capture a total of three sets of garments: (1) a red dress with a short full skirt; (2) a flared, short skirt[4] in floral pattern with a bottom ruffle; (3) a loose T-shirt and a long skirt. The training data are captured in a multi-view capture studio equipped with roughly 150 RGB cameras. This dense capture setup enables us to use Multi-View Stereo (MVS) to reconstruct geometry, which can be rasterized to depth and used as input to drive our avatars. We split out a small segment of each sequence for testing purpose and use the multi-view captured images as ground truth for evaluation.

To demonstrate the application of our method for telepresence, we additionally capture the subjects with the same garments in a novel environment with different background and

[3] https://drive.google.com/file/d/1D7J0nEc0zA2vMwxDARiP6NHeQVxvH929/view?usp=sharing

[4] The paired upper-body garments are tight and modelled in the body layer.

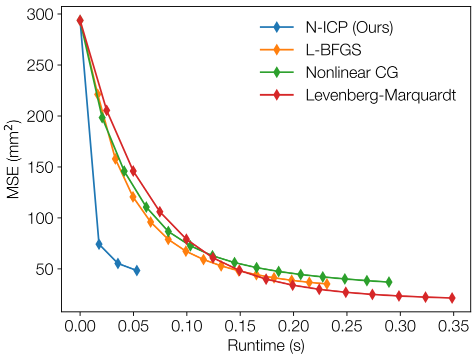

Figure 6.4: Comparison between our method and classical non-rigid ICP with different types of optimization solvers. For each method, we plot the runtime in seconds vs. the Mean Squared Surface Error (MSE) in mm^2 for the surface tracking results. The square markers on the curves denote individual steps in the optimization.

illumination. It is designed to be a simpler capture setup, where nine Kinect RGB-D cameras are deployed, and we select three cameras as driving input to our model. We also split the data in the novel environment for fine-tuning and testing.

6.6.2 Evaluation of Neural Iterative Closest Point

We conduct an evaluation on the N-ICP algorithm using the dress sequence. We use the ground truth from offline registration by non-rigid ICP from Chapter 4 [216]. For this evaluation, we adopt the same input as our full method: a point cloud fusing the depth from three driving views. We report the evaluation metric of the mean squared point-to-triangle distance for both directions from prediction to ground-truth and from ground truth to prediction.

We compare N-ICP with classical optimization solvers, including L-BFGS and nonlinear Conjugate Gradient with strong Wolfe linear search implemented in PyTorch-Minimize [52], and Levenberg-Marquart implemented in Theseus [154]. All the methods are implemented in CUDA PyTorch with analytically computed derivatives. The results are shown in Figure 6.4. Our method converges at a faster rate than the baseline methods, including both the gradient-based methods (L-BFGS, CG and ours) and the more complicated Jacobian-based Levenberg-Marquardt method.

We also conduct ablation studies on our design of the N-ICP algorithm. The results are shown in Table 6.1. The most naive baseline is to simply use the point cloud as the input feature, shown on the first row of the table. On the second row, we add the closest point residual to the input feature, which provides useful information for surface alignment and enables an iterative update of the deformation parameters. The following rows suggest that the energy gradient derived from the residuals can provide more effective guidance, similar to its critical role in traditional nonlinear optimization. The last two rows verify the benefit of iterative parameter update compared with a one-shot prediction by the network.

Table 6.1: Ablation studies on different types of input for N-ICP. The evaluation metric is the Mean Squared Error (MSE) in mm^2 between surfaces. $\mathbf{P}, \mathbf{r}$ and $\mathbf{g}$ refer to the point cloud, residual and gradient defined in Section 6.4. N denotes the number of update iterations. When $N = 1$, the network makes a one-shot prediction. Our full method is shown in the last row.

Input Type	MSE (mm^2)
$\mathbf{P}$ ($N = 1$)	101.19
$\mathbf{P}, \mathbf{r}$ ($N = 3$)	82.72
$\mathbf{P}, \mathbf{g}$ ($N = 3$)	49.60
$\mathbf{P}, \mathbf{r}, \mathbf{g}$ ($N = 1$)	72.30
$\mathbf{P}, \mathbf{r}, \mathbf{g}$ ($N = 3$, full)	**48.47**

Table 6.2: Quantitative comparison with various baselines and ablation studies using the loose T-shirt and long skirt sequence captured in the multi-view studio. The metrics are computed on the whole rendered images with a plain background from two different views. We compare with DVA [161] and its two-layer variant, ENeRF [113], KeypointNeRF [134], and sensing-based baselines. Ablation studies are also included.

Methods	$L_1 \downarrow$	PSNR↑	SSIM↑	LPIPS↓
DVA	2.89	27.96	0.9218	0.0932
DVA (Two-layer)	2.90	28.03	0.9219	0.0912
ENeRF (Masked)	3.07	27.21	0.9130	0.0772
KeypointNeRF	3.14	27.42	0.9111	0.1123
Depth-based warping	3.22	26.94	0.9171	0.1111
+ LookinGood UNet	2.27	29.91	0.9382	0.0865
Ours w/o N-ICP	2.29	29.04	0.9299	0.0755
Ours w/o depth offset	2.19	29.35	0.9327	0.0728
Ours w/o part loss	2.11	29.63	0.9354	0.0724
Ours full	**1.93**	**30.25**	**0.9404**	**0.0686**

6.6.3 Full Method Evaluation

In this section, we present evaluation and comparison using high-quality data from the dense multi-view capture studio. For comparison with most methods, we report quantitative results on the the challenging loose T-shirt and long skirt sequence in in Table 6.2, except for pose-driven clothed avatars because the tucked T-shirt is extremely challenging to simulate using the method in Chapter 4 [216]. Therefore for this category we report results on the skirt with floral pattern shown in Table 6.3.

Comparison w/ DVA [161]. Although DVA adopts the Mixture of Volumetric Primitive (MVP) [119] formulation that can render any structure in theory, the spatial arrangement of the primitives relies on the guidance of base body mesh as initialization. Both the original method and its variant with body and clothing layers are guided by LBS transformation,

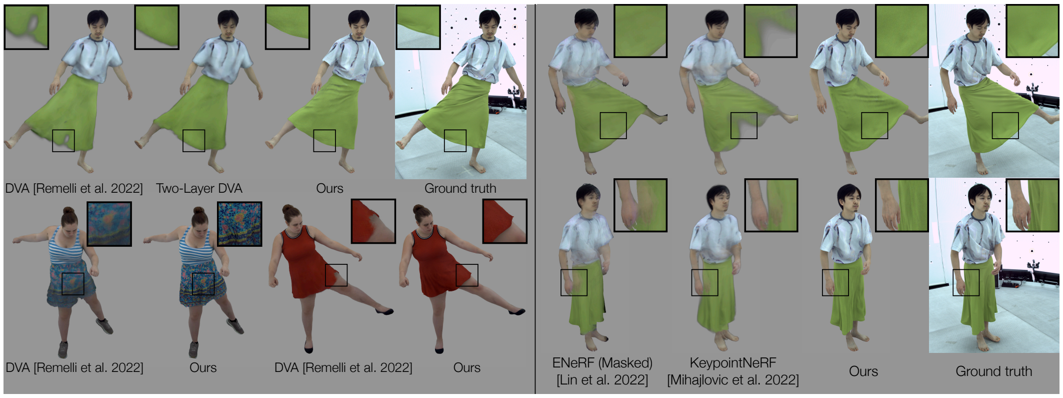

Figure 6.5: **Left**: comparison with Drivable Volumetric Avatars (DVA) [161] and its two-layer variant; **right**: comparison with NeRF-based methods [113, 134] against the ground truth. For the results of ENeRF [113], we mask out its black background with ground truth mask.

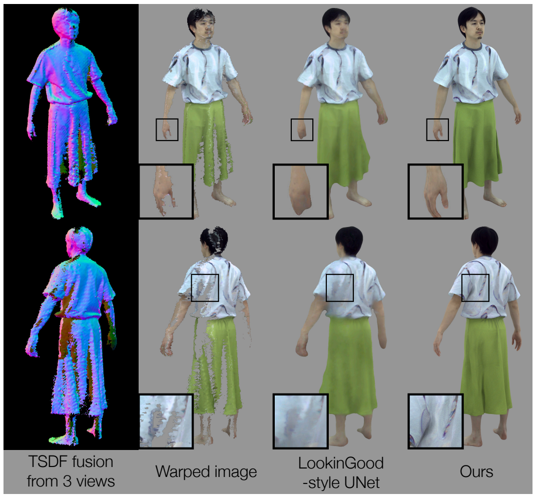

Figure 6.6: Comparison with basic sensing-based baselines. Given the depth input from 3 views, we run TSDF fusion [42, 47] to obtain a proxy geometry (first column), and then warp the driving RGB images to the target view (second column). We train a UNet following LookinGood [131] to inpaint the missing regions (third column).

which is too coarse to capture the motion of loose clothing. Qualitative comparison is shown on the left of Figure 6.5.

Comparison w/ sensing-based baselines. Most sensing-based approaches [49, 131, 174, 237] are proprietary and tightly integrated with their capture system without available open-source implementations. Therefore, we rather compare our method with baselines that can be implemented with moderate effort, listed in the third group of results in Table 6.2, including image warping based on TSDF fusion and further applying a U-Net follwing

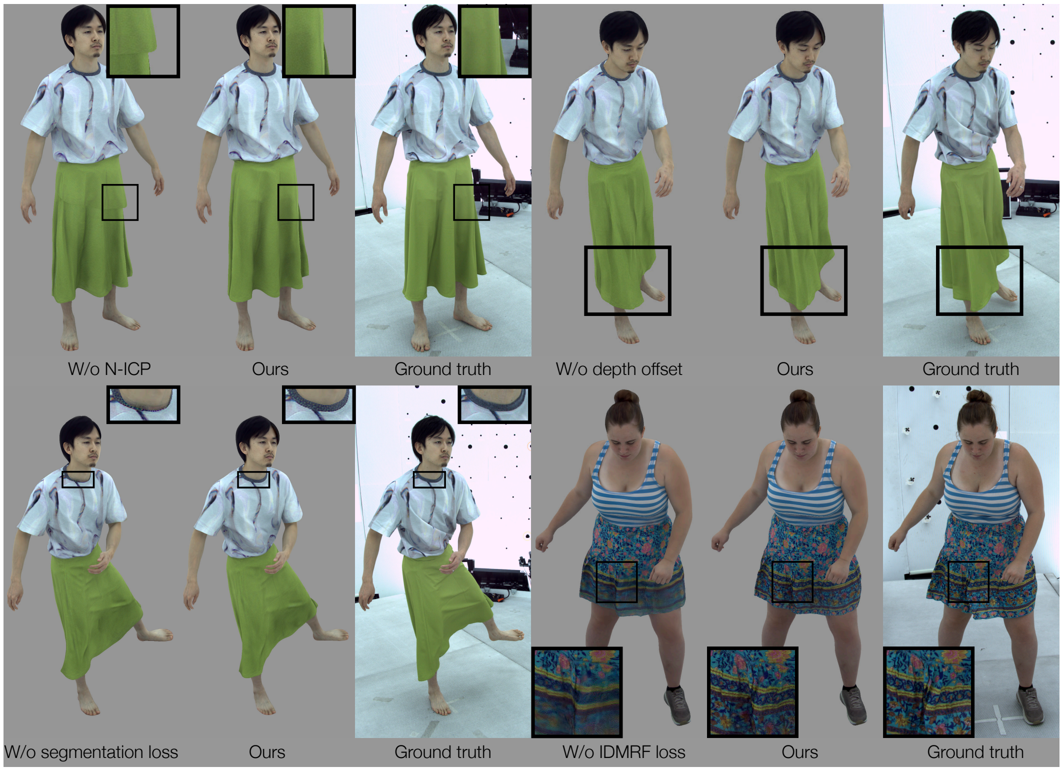

Figure 6.7: Ablation studies on components in our framework, including N-ICP, depth offset as input to the encoder-decoder, part segmentation loss, and ID-MRF loss. Each result is shown in comparison to our full output and the ground truth.

the idea of LookinGood [131]. We first fuse the sparse input depth maps into a single Truncated Signed Distance Field (TSDF) volume [42, 47], and then extract from it an explicit mesh representation. Using the fused geometry, we can then warp the input RGB images from the source views to any target view. However, the warped image is usually imperfect because the fused geometry is often incomplete and noisy. Therefore, we follow the idea of LookinGood [131] and train a U-Net to complete the warped image. This baseline essentially learns to inpaint complete human appearance from partial input only in the screen space, and struggles to achieve 3D-aware temporal consistency in the output. This experiment is not intended to be a full-scale comparison against state-of-the-art sensing-based approaches, but to better understand our method in comparison to a modest baseline along this line of work given similar input.

Ablation studies. We show ablation studies on several components of our framework and the results are shown in Figure 6.7 and the fourth group of Table 6.2. First, the initial tracking by N-ICP provides a basis for the whole framework to estimate the correct clothing geometry and appearance. Without this component, the initialization by only body pose is too coarse and leads to obvious artifacts. Second, we demonstrate that the additional depth offset input to the encoder-decoder allows our method to predict more accurate over-

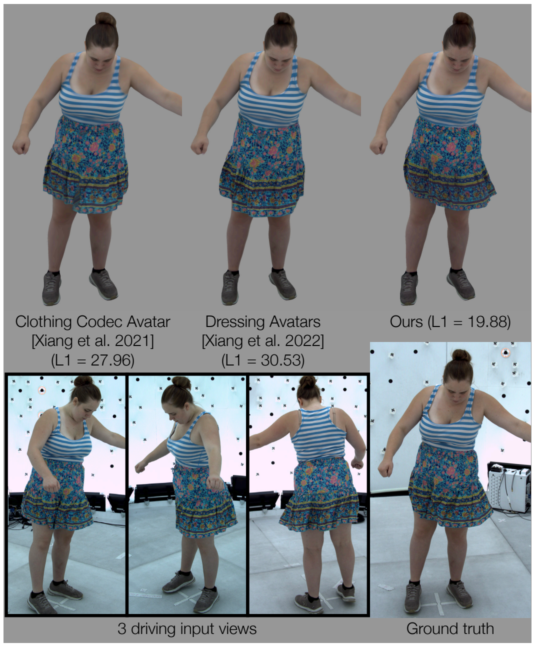

Figure 6.8: Qualitative comparison with Clothing Codec Avatars [218] (Chapter 3) and Dressing Avatars [216] (Chapter 4). The results are shown on the top row; the input views and ground truth are shown on the bottom. The L_1 error in the skirt region is reported beside the names of the methods.

Table 6.3: Quantitative comparison with Clothing Codec Avatars (Chapter 3 [218]) and Dressing Avatars (Chapter 4 [216]) on the skirt sequence. The metrics are computed in the skirt region.

Methods	L_1↓	PSNR↑	SSIM↑	LPIPS↓
Clothing CAs	25.46	17.81	0.2431	0.507
Dressing Avatars	27.95	17.01	0.1818	0.486
Ours	**15.67**	**21.99**	**0.6527**	**0.203**

all clothing shape. Third, we verify that the part segmentation loss helps to produce correct body-clothing boundary. Last, we compare the results with and without ID-MRF loss, which preserves the high-frequency texture detail in our output when the predicted geometry is not completely photometrically aligned with the driving images.

Comparison w/ pose-driven avatars. A major motivation of our framework is that the underlying body motion does not contain enough information to fully determine the loose clothing states. We validate this intuition by comparison with Clothing Codec Avatars (Chapter 3 [218]) and Dressing Avatars (Chapter 4 [216]) shown in Figure 6.8 and Table 6.3. Clothing Codec Avatars struggle to learn the mapping from a sequence of body pose to large clothing dynamics. Dressing Avatars produce clothing motion that looks *realistic*

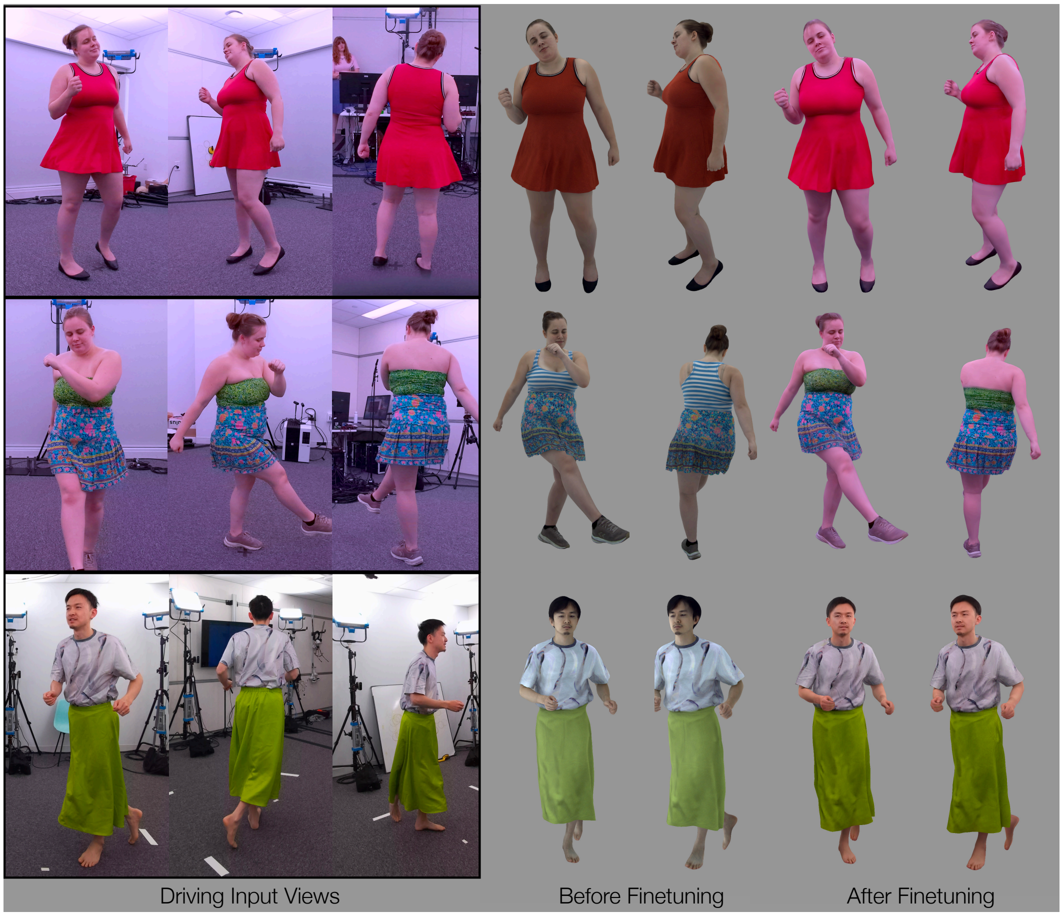

Figure 6.9: Testing results in the novel capture environment. On the left we show in the input RGB-images. We show our output from 2 different viewpoints both before and after fine-tuning. These frames are not seen during fine-tuning. The subject on the second row wears different upper-body garments between the captures in the original and new environment. The tank top is preserved in the body layer before fine-tuning and adapted after that.

with the help of physics-based simulation, but its output is not *faithful* to the actual motion because of the lack of an efficient approach for estimating underlying physical parameters for simulation. Our method utilizes the driving signal from sparse RGB-D input to achieve faithful clothing telepresence, which is verified by the low error on the evaluation metrics in Table 6.3.

6.6.4 Results in the Novel Capture Environment

For the novel capture environment, we test our model in two different scenarios: without and with fine-tuning, both shown in Figure 6.9. The first scenario refers to a direct application of the model trained from the dense capture studio to the same subject but in the novel

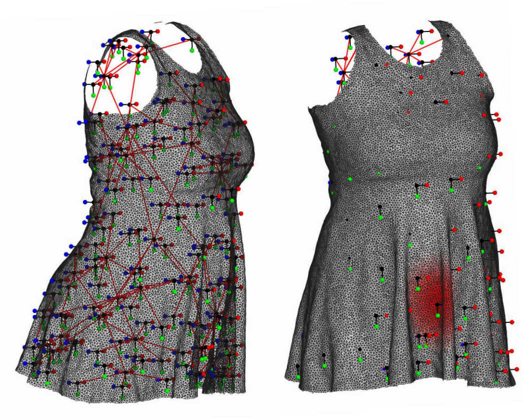

Figure 6.10: A visualization of the deformation graph $\mathcal{E}$ used in the dress example. On the left side, we show the coordinate frame at each graph node and their connectivity by the red lines. On the right side, the region of influence by a node located in the center is shown in red.

environment. For this scenario, our model needs to handle the difference in the input RGB-D images between the training and testing environments, such as illumination and sensor setup. Note that our model is trained to be robust to these variations in the input and preserve the appearance style from the original training data in our output (see supplementary document), but with the unseen body and clothing motion captured in the novel environment. This experiment demonstrates the ability of our method to directly generalize to a novel environment.

In the second scenario, we test our model after fine-tuning it with the training data captured in the new environment. Note that we use the same three Kinects cameras for both driving input and ground truth supervision. During fine-tuning, the output of our model adapts to the new appearance caused by the illumination and sensor specification in the environment, as well as the change in body over time such as hair style. We test our model with a even more aggressive change in the body-layer appearance from the tank-top to the green capture suit in the second row of Figure 6.9. The fine-tuning step can be viewed as an option to further boost the model output quality if time and computation budgets permit.

6.7 Implementation Detail

In this section, we provide the implementation detail for better reproducibility.

6.7.1 Clothing Deformation Graph

In Figure 6.10, we provide a visual illustration of the deformation graph $\mathcal{E}$ in the inner layer of the clothing deformation model $\mathcal{D}$ (Section 6.4) for the dress example. The parameters for the deformation graph include the rotation and translation for each node:

$$\boldsymbol{\theta} = \{\mathbf{r}_k, \mathbf{t}_k\}_{k=1}^{K},\ \mathbf{r}_k, \mathbf{t}_k \in \mathbb{R}^3, \tag{6.5}$$

where $\mathbf{r}_k$ is the axis-angle representation of a 3D rotation. We use a total of $K = 125$ nodes for each example.

6.7.2 Training Setup

N-ICP.

When training the N-ICP module, we adopt a regularization term for deformation graph that compares the difference in transformation between adjacent nodes:

$$L_{\text{DG-Reg}} = \frac{1}{K(K-1)} \sum_{1 \leq j \neq k \leq K} \|T_j \mathbf{m}_{jk} - T_k \mathbf{m}_{jk}\|^2, \tag{6.6}$$

where T_j and T_k denote the SE(3) transformation for the $j-$th and $k-$th nodes respectively, and $\mathbf{m}_{ij}$ denotes the middle point between the rest positions of the $j-$th and $k-$th nodes. Then the total loss function for training N-ICP is written as

$$L_{\text{N-ICP}} = \frac{1}{N} \sum_{i=1}^{N} L_{\text{ICP}}(\boldsymbol{\theta}^{(i)}, \overline{\mathbf{P}}) + \lambda_{\text{DG-Reg}} L_{\text{DG-Reg}}, \tag{6.7}$$

where the balancing weight is set to $\lambda_{\text{DG-Reg}} = 1 \times 10^{-3}$. The trainable parameters in N-ICP are those in PointNet $\mathcal{M}$. The input and output of the PointNet $\mathcal{M}$ are converted to the root body coordinate of the subject given the tracked body pose $\boldsymbol{\rho}$ to be invariant to the global orientation and translation. We use the AdamW optimizer with an initial learning rate of 1×10^{-5}.

Initialization. We find it crucial to initialize the parameters in the last layer of the PointNet with values close to zero, so that $\boldsymbol{\theta}^{(i)} = \Delta\boldsymbol{\theta}^{(i)} \approx \mathbf{0}$ for $i = 1, \ldots, N$ at the first training iteration, with $\boldsymbol{\theta}^{(0)}$ set to $\mathbf{0}$. In this way, thanks to the two-layer clothing deformation model (Section 6.4), $\mathcal{D}(\boldsymbol{\theta}^{(i)})$ is close enough to the ICP target to generate meaningful gradient at the beginning of the training process, and gradually converges to the desired minimum. In practice, we initialize the last layer of the network by random sampling from a uniform distribution $U[-\varepsilon, \varepsilon]$ where $\varepsilon = 1 \times 10^{-6}$.

Discussion on supervision. N-ICP is trained in a self-supervised manner, because the loss function $L_{\text{N-ICP}}$ does not involve the "ground truth" deformation parameters. The reasons are two-fold. First, it takes extra processing time efforts obtain the ground truth. Second, the problem of estimating reliable "ground truth" deformation parameters is challenging by itself. Unless the garment under capture has been specially designed to encode correspondences in a printed pattern [71], otherwise, the principal approach is to run offline ICP between the deformation model and MVS geometry. In this way, the "ground truth" essentially offers no more information than directly supervising N-ICP by MVS. The self-supervised formulation, instead, allows solving a global optimization by sharing the information across all frames.

Texel-Conditioned Clothed Avatars.

We use the following loss functions to train the texel-conditioned clothed avatars (Section 6.5)

$$L_{\text{avatars}} = \sum_{i} \lambda_{\text{i}} L_{\text{i}}, \; i \in \{\text{RGB}, \text{mask}, \text{reg}, \text{part}, \text{ID-MRF}\}. \tag{6.8}$$

L_{RGB} and L_{mask} are the standard L_1 losses for RGB and mask respectively; L_{reg} is the Laplacian regularization terms for body and clothing meshes. L_{part} is similar to L_{mask} but identify background, body and clothing in three different categories. Following [54], we use the ID-MRF loss [207], a stronger form of perceptual loss to encourage sharpness for high-frequency texture in the clothing region. We use $\lambda_{\text{RGB}} = 0.2, \lambda_{\text{mask}} = \lambda_{\text{part}} = 500.0, \lambda_{\text{reg}} = 100.0, \lambda_{\text{ID-MRF}} = 1.0$. The gradient of loss functions defined in the image space (RGB, mask, part and ID-MRF) with respect to the network parameters are back-propagated through a differentiable rasterizer. We use the AdamW optimizer with a learning rate of 1×10^{-3}.

Color Augmentation. In order to deal with the domain gap in illumination and color when the directly applying the avatars to the novel capture environment (Section 6.6.4), we apply a random color augmentation to texel-aligned RGB features $\mathbf{F}_I$ using the 'ColorJitter' function in TorchVision[5] at training time. Notice that we leave the ground truth images used for supervision in L_{avatars} unchanged, so that the network always preserves the original appearance in the *output*, despite a different color mode in the *input* feature $\mathbf{F}_I$ when we direct apply the model to the novel environment. The output appearance only changes after fine-tuning with ground truth images in the novel environment.

6.7.3 Preprocessing and Postprocessing

Input Preprocessing.

Our method takes RGB and depth images as input. When training and testing using data from the dense capture studio, we run image-based part segmentation and transfer the result to the MVS mesh by projection and visibility check. This operation allows us to extract the clothing region. The MVS mesh may include floating noise, which we remove by checking the mesh connectivity and setting a threshold on the minimal number of vertices in a connected component. Then, we rasterize the segmented mesh to RGB views to "simulate" a depth image.

When training and testing in the novel environment, we use the RGB-D images from calibrated Kinect sensors as input. We also run image-based part segmentation to extract the clothing regions. Then we use TSDF fusion [42] and Marching Cubes [121] to form a mesh from the extracted depth images, which allows us to perform similar connectivity check as above to remove noise from the depth sensors.

Temporal Smoothing.

Due to the unstructured point cloud input, the output of N-ICP may have undesirable jittering. We apply temporal smoothing to the output of N-ICP by taking the average on the vertex positions in a small temporal window, which is feasiable because the N-ICP output shares a consistent registered topology across all the frames. The filtered meshes are then used to unwrap texel-aligned features and as input to the texel-conditioned avatars as shown in Figure 6.2. We find no need to apply additional smoothing on the final output of texel-conditioned avatars if the provided initial tracking is temporally stable and the floating depth noise has been removed in the preprocessing step.

[5] `https://github.com/pytorch/vision`. We use the following parameters: brightness=0.5, contrast=0.5, saturation=0.5, hue=0.2.

Collision.

To resolve the collision between the body and clothing layers, which is usually slight in the results, we follow Clothing Codec Avatars in Section 3.6 [218] to project the clothing vertices in collision beyond the nearest body points by a slight margin. More sophisticated ways to handle collision based on geometry or learning [192] may be incorporated, which we leave for future work.

6.7.4 Network Architecture

N-ICP.

N-ICP takes an unstructured point cloud as input, so we adopt the PointNet++ [159] architecture. To specify the architecture, we reuse the notation of Set Abstraction function from [159]:

$$\text{SA}(K, r, [l_1, \dots, l_d]),$$

where K denotes the number of grouping centers, r denotes the radius of the grouping regions, and l_i denotes the output size of a fully connected layer in the Multi-Layer Perceptron (MLP). We also denote a standalone MLP as $\text{FC}([l_1, \dots, l_d])$. Then the architecture of the network $\mathcal{M}$ can be described as

$$\begin{gathered}[\mathbf{p}, \mathbf{r}] \rightarrow \text{SA}(32, 0.1, [16, 16, 32]) \rightarrow \text{SA}(32, 0.2, [64, 64, 128]) \rightarrow \\ \text{SA}(32, 0.4, [256, 256, 256]) \rightarrow \text{SA}(32, 0.8, [256, 256, 512]) \rightarrow \\ \text{MaxPool} \rightarrow \bigoplus \mathbf{g} \rightarrow \text{FC}([512, 512, 512, 750]) \rightarrow \Delta\boldsymbol{\theta},\end{gathered}$$

where $\mathbf{p}$ and $\mathbf{r}$ denote the point coordinate and residual as defined in Section 6.4, and $\bigoplus \mathbf{g}$ denotes the operation to concatenate the result from the previous step with gradient input $\mathbf{g}$.

Texel-Conditioned Clothed Avatars.

The overall architecture of the texel-conditioned avatar models is shown in Figure 6.11. Given the texel-aligned features $\mathbf{F}_I, \mathbf{F}_D$ unwrapped from the initial tracking results $\mathbf{D}$ as input, the encoder produces a feature map that is spatially aligned with the input. The encoded feature maps are then decoder into a vertex offset map $\delta\mathbf{G}$, from which the offsets are extracted and then applied on top of the initial tracking to obtain the output geometry $\mathbf{G}$. The geometry $\mathbf{G}$ and the viewpoint $\mathbf{v}$ are used together to compute the view-conditioning, including the viewing vector expressed in the local Tangent-Bitangent-Normal (TBN) frame (Section 4.5 [216]) as well as its reflected direction. The view-dependent U-Net takes in the view conditioning and the view-independent texture to produce an additive view-dependent offset. With the final geometry $\mathbf{G}$, we also compute the ambient occlusion, which is fed into the shadow U-Net to produce a multiplicative shadow map. The view-dependent texture is then upsampled to 2k resolution by a upsampling network.

To specify the architecture of the individual networks above, we define the blocks shown in Figure 6.12.

(1) *Convolutional encoder* consists of the network blocks in the following table. Following DVA [161], we find that using a U-Net at 64×64 resolution instead of a bottleneck structure helps to preserve the UV-space detail in the output.

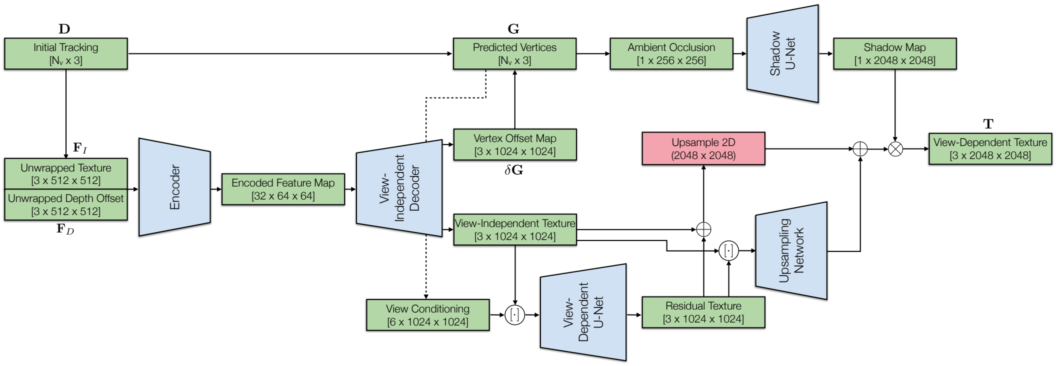

Figure 6.11: The network architecture for the texel-conditioned clothed avatars. It consists of the following five components: (1) a convolutional encoder that encodes the texel-aligned input, (2) a view-independent decoder that outputs vertex and texture maps, (3) a view-dependent U-Net that regresses view-dependent variation in the texture, (4) a shadow network that takes in ambient occlusion and computes a multiplicative shadow map, and (5) an upsampling network that predicts the residual after increasing the spatial resolution from 1024 to 2048.

Block	Output Size $(C \times H \times W)$
ConvBlock(6, 16, 1)	$16 \times 512 \times 512$
ConvDownBlock(16, 32, 1)	$32 \times 256 \times 256$
ConvDownBlock(32, 64, 1)	$64 \times 128 \times 128$
ConvDownBlock(64, 64, 1)	$64 \times 64 \times 64$
U-Net(64, 64, 32)	$32 \times 64 \times 64$

(2) *View-independent decoder* consists of the network blocks in the following table. Here, the "RepeatChannels" operation repeats the channels of the input feature for the geometry and texture branches. The following "ConvUpBlocks" processing them separately in different groups. The output is then evenly split into a vertex offset map and a texture map.

Block	Output Size $(C \times H \times W)$
ConvBlock(32, 32, 1)	$32 \times 64 \times 64$
RepeatChannels	$64 \times 64 \times 64$
ConvUpBlock(64, 32, 2)	$32 \times 128 \times 128$
ConvUpBlock(32, 16, 2)	$16 \times 256 \times 256$
ConvUpBlock(16, 8, 2)	$8 \times 512 \times 512$
ConvUpBlock(8, 8, 2)	$8 \times 1024 \times 1024$
Conv2D(8, 6, 2, k=1)	$6 \times 1024 \times 1024$
SplitChannels	$(2\ \times)\ 3 \times 1024 \times 1024$

(3) *View-dependent U-Net* is a single block "U-Net(9, 4, 3)" defined in Figure 6.12.

(4) *Shadow U-Net* is an upsampling operation on the input ambient occlusion map from 256 resolution to 2048, followed by a block "U-Net(1, 2, 1)".

(5) *Upsampling network* is defined in the following table. Here the "PixelShuffle(r)" is an operation that rearranges a tensor from shape $(C \times r^2) \times H \times W$ to $C \times (H \times r) \times (W \times r)$.

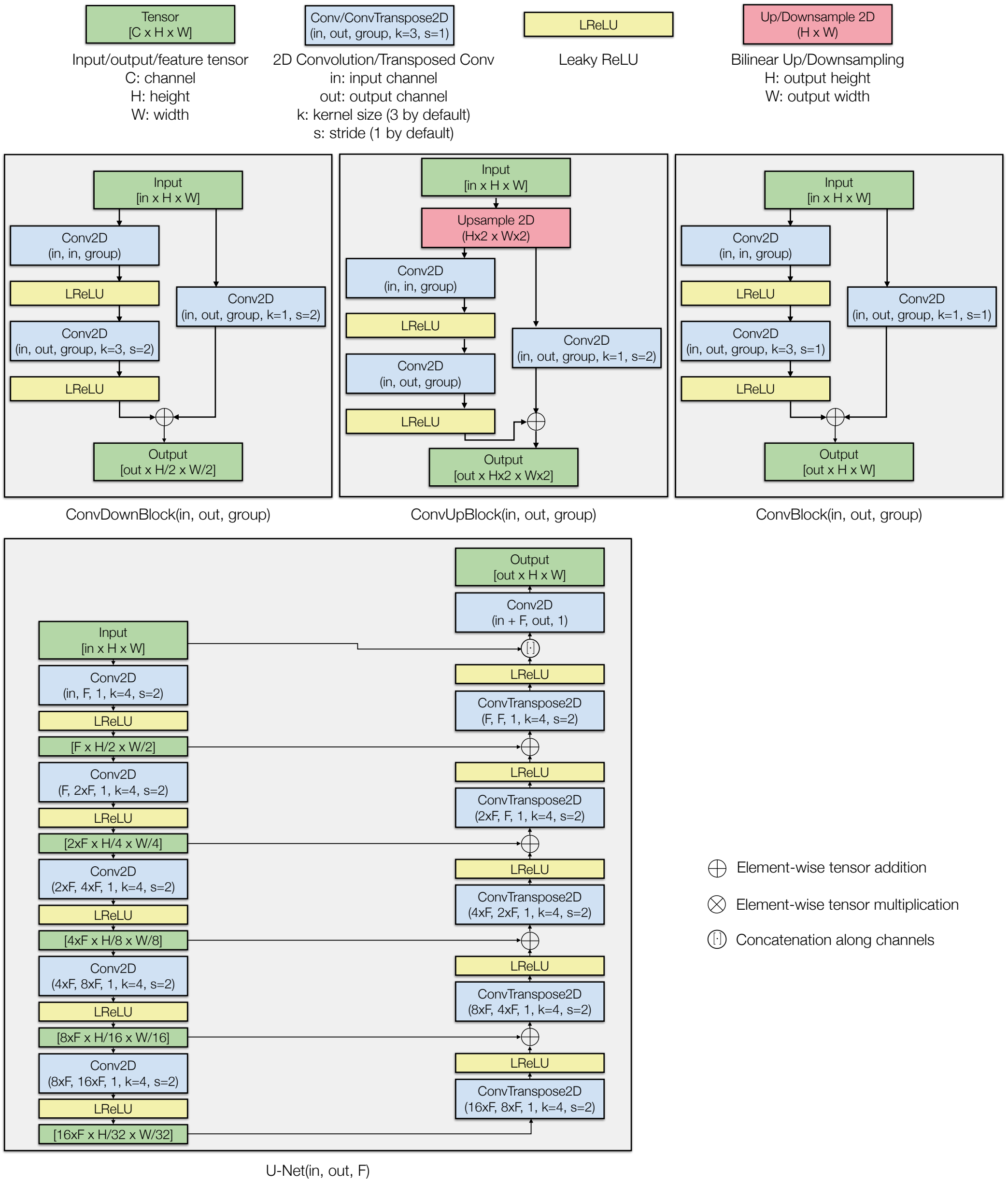

Figure 6.12: Network blocks used in the architecture of texel-aligned avatars.

Block	Output Size $(C \times H \times W)$
Conv2D$(6, 2, 1)$	$2 \times 1024 \times 1024$
LReLU(0.2)	$2 \times 1024 \times 1024$
Conv2D$(2, 12, 1)$	$12 \times 1024 \times 1024$
PixelShuffle(2)	$3 \times 2048 \times 2048$

6.7.5 Training Data Preparation

In this section, we describe how we prepare the assets required to train the avatars. Given the multi-view images captured by more than one hundred synchronized cameras, we run 2D keypoint detection, part segmentation, and Multi-View Stereo (MVS). The 2D body keypoints are triangulated to estimate 3D keypoints. For each vertex in the MVS output, we aggregate its category label from each camera view by checking its projection in the image segmentation, and then perform a majority voting, followed by a Markov Random Field (MRF) to ensure spatial smoothness. We also use the method in [242] to estimate an underlying body template and the body pose for each frame given the 3D keypoints and segmented MVS mesh. The whole process is similar to the one in Chapter 3, except that we do not perform clothing registration offline in the style of ClothCap [155]. Instead, we define a deformation model $\mathcal{D}(\boldsymbol{\theta})$, and train the N-ICP network to track the clothing in a self-supervised manner. The clothing template is created from the segmented clothing region in the MVS mesh in a rest-pose frame with some manual cleanup and remeshing.

6.7.6 Comments on Runtime

The two major computationally intensive modules in this method are the N-ICP and texel-conditioned clothed avatars. As shown in Figure 6.3, the N-ICP module takes less than 50 milliseconds per frame when running three iterations in PyTorch without any optimization of runtime performance. The two-layer architecture of our texel-conditioned avatars is similar to Drivable Volumetric Avatars [161], which are measured to run at 85 fps in their supplementary document. With better engineering effort, especially the adoption of deep learning inference library such as TensorRT[6], our method can be potentially integrated into a real-time system for telepresence.

6.8 Conclusion

We presented a framework for building photorealistic full-body avatars that can be driven by sparse RGB-D inputs and faithfully reproduce the motion of loose clothing. Our method accurately reconstructs challenging clothing appearance of the subjects, thus tackling a major drawback of existing pose-driven avatars. As per limitations, our model is still person- and garment-specific and cannot handle clothing motion that falls far outside of the deformation space of the deformation graph model, such as topology change. Interesting future directions would be to extend our method to multi-identity setting, and develop a formulation that can handle more generic garment categories, e.g. with implicit representations.

[6] `https://developer.nvidia.com/tensorrt`

Chapter 7

Conclusion

We have presented a comprehensive framework to build high-quality full-body photorealistic avatars with challenging dynamic loose clothing. Except for the MonoClothCap work in Chapter 5 where the problem setting is different, the three other pieces of work in Chapter 3, 4 and 6 dive deeply into a series of problems sharing the following characteristics:

- **Dense personalized capture:** an individual wearing a set of garments is captured in a multi-view system with more than one hundred synchronized cameras to train a personalized model;

- **Sparse driving signal:** at test time, we assume very sparse input signals, e.g., skeleton motion or few RGB-D sensors, to drive the models, which produce appearance consistent with the training data and compatible with the driving signal;

- **Photorealistic appearance:** our output includes not only the geometry, but also can be rendered in a photorealistic manner from arbitrary viewpoints;

- **Dynamic clothing:** we focus on scenarios where the clothing poses a significant challenge to the appearance modeling because of its motion that is separate from the underlying body.

Under such a setting, the work in this thesis has made significant progress as compared to baseline methods such as the Full-Body Codec Avatars [7]. Each chapter explores this problem from a different perspective, and applies tools in specific domains to achieve the best possible quality, including data-driven priors in Chapter 3, physics in Chapter 4 and additional sensing in Chapter 6.

In Table 7.1, we summarize the connection and differences between these pieces of work: Clothing Codec Avatars (CCA) in Chapter 3 [218], Dressing Avatars (DA) in Chapter 4 [216], and Drivable Avatar Clothing (DAC) in Chapter 6 [219]. The idea central to all three methods is to model clothing as a separate layer on top of the human body. DA and DAC can generate richer and more realistic dynamics for loose clothing than CCA, but DA requires a proprietary implementation of real-time cloth simulation. In terms of driving signal, CCA and DA take body and face motion as input, while DAC additionally requires sparse RGB-D views. CCA and DA utilize ground truth clothing registration to train their models, while DAC does not require such preprocessing and can be trained in a self-supervised way. Finally, with the additional sensing information, the output of DAC is more faithful to the

Table 7.1: Comparison between Clothing Codec Avatars (CCA) in Chapter 3 [218], Dressing Avatars (DA) in Chapter 4 [216] and Drivable Avatar Clothing (DAC) in Chapter 6 [219].

	CCA	DA	DAC
Clothing as a separate layer	✓	✓	✓
Loose clothing dynamics		✓	✓
Physical simulation		✓	
RGB-D driving input			✓
Ground truth registration	✓	✓	
Faithful output			✓

actual clothing motion than CCA and DA due to the intrinsic ambiguity of clothing states given only body motion as input. We believe that these three methods are actually complementary to each other depending on the actual use cases, and that an ultimate system might be a flexible combination of these components.

7.1 Limitations

Here we discuss the major limitations shared by the methods in this thesis.

Multi-layer clothing. In this thesis, a common assumption is that the appearance of the subject can be modeled by a body layer and a clothing layer, and the deformation in the clothing layer can be explained by a relatively "isometric" (distance-preserving) transformation of a surface template. This framework can already handle multiple pieces of clothing to some extent. For example, the pants in Figure 3.10 and the tank top in Figure 6.8 follow the body motion quite closely and are thus modeled as part of the inner body layer. As another example, the T-shirt and the green skirt are jointly modeled in the clothing layer in the last row of Figure 6.9.

However, multi-layer clothing can violate this assumption if different layers are disconnected and can move separately from each other, as well as separately from the underlying body. Imagine that the T-shirt is untucked from under the green skirt in the example at the bottom of Figure 6.9. Then it will deform in a way that is separate from the skirt. The *topology* of the clothing layer will change and cannot be explained by the deformation model we use in Chapter 6. In addition, the bottom end of the T-shirt will be exposed and then cover part of the green skirt, leading to a *missing* part from the template as well as a *redundant* part that cannot be explained by the template. More complicated types of clothing such as Kimonos may suffer more seriously from such problems. Extending our framework to model clothing with additional layers may be a possible solution, but several issues need to be addressed, such as how to handle the interaction between different layers and impute the information for the occluded layers.

More complex appearance effects. For most of the thesis, we assume a fixed illumination environment from the multi-view capture studio, which is baked into the learned appearance model for the subjects. We take a preliminary step to test our method in a different illumination setup in Section 6.6.4, but treat the problem in a simple way, by either keeping the original illumination condition or fine-tuning the output to adapt to the new condition. One way to formally address this problem is to introduce illumination as an explicit input variable to our models, and train them with data captured with controlled varying lighting, i.e. in light stages [15, 46]. The caveat here is the increasing scale of data required to cover the joint distribution of clothing deformation and illumination variation.

Our appearance models treat view direction as an explicit input and are able to model some view-dependent effects such as Fresnel reflection in Figure 3.13 and 4.5. However, we have not tested our methods on highly specular surfaces which may appear in clothing made of certain materials, or some components such as zippers. While we approximate the mutual shadows between body and clothing by ambient occlusion and shadow networks, the models do not explicitly account for higher-order appearance effects such as interreflection. The artifacts caused by this simplification is obvious when we put the red dress on the body avatars wearing the green capture suit.

Reliance on data and capture system. As a data-driven framework, there is little surprise that the methods in this thesis cannot model what is not included in the training data. Besides clothing, there are other body parts that do not closely follow the skeleton motion such as loose hair and accessories that may change on a daily basis such as glasses. These components share some similarities with clothing but may require different modeling strategies. For example, meshes are often not a good representation for loose hair [165, 177, 208, 209]; compared with clothing, specular reflection plays a much heavier role in the modeling of glasses [105]. From an engineering perspective, it may be possible to directly put these components together if the cost is permissible; however, it remains an intriguing research question whether there will be an approach that can unify all these components *without* sacrificing too much quality in appearance, dynamics, and drivability.

The models we build in this thesis are essentially person-specific and garment-specific, and they rely on the training data for individual subjects from the dense capture studio. We focus on modeling the intra-class variation of particular subjects and outfits under a wide range of possible states of deformation. On the one hand, our methods demonstrate what quality is achievable for telepresence when sufficient information is given about a subject with an outfit; on the other hand, we hope that the techniques involved in the system will lay a foundation and be useful, though certainly not sufficient, for solving the inter-class problems across different identities and garments. With such consideration, we acknowledge that our formulation is not easily affordable at the consumer level. At the corporate business level, however, these methods may inspire some solutions such as hosting a virtual wardrobe consisting of a certain number of high-quality clothing instances, which can be chosen by users to drive their own avatars.

7.2 Future Work

Here we discuss some interesting future directions that extend the scope of this thesis.

Universal models for clothed humans. This thesis focuses on person-specific and garment-specific photorealistic avatars, and in the future, we are interested in building universal models that span the space of a large number of personal identities and, in particular, multiple instances of garments. For the underlying body avatars, following the path of universal face avatars [25] may be a direct and viable solution, because the variance of underlying body shape and appearance across different people is relatively small. For a fully clothed model, we need to answer several central questions, especially representation and data.

Representation. This thesis primarily uses mesh as the geometric representation of clothing. On the positive side, it allows us to efficiently and easily apply priors derived from traditional graphics literature in many aspects, such as shape (e.g. Laplacian and ARAP [181] energy), physics (e.g. simulation [130] and collision), and appearance modeling (e.g. ambient occlusion). However, in order to model multiple clothing categories in a single model, the mesh representation has some obvious disadvantages such as the reliance on a predefined topology. We need to consider other options, including (signed/unsigned) distance fields [39,45], radiance fields (including tri-plane tensors) [76,176], points [127,129], and shape primitives [122,161]. Each representation has its own advantages and disadvantages in terms of flexibility, rendering quality, and speed. We believe that it is important to identify the most suitable ones for individual subproblems such as generation, animation, and relighting, and integrate them properly in the whole system.

Data. For future work, we would like to go beyond using the data from the multi-view capture system in this thesis. While high-quality data from the studio, including individual 3D scans, can be useful for building certain prior models such as dynamics, shading, and surface normals, the major problem is that they do not scale up as easily as in-the-wild data. We are interested in utilizing large-scale datasets of images to cover a wide range of identities and garments [48,244], including paired image-and-text data for the descriptive power of language [26,95]. We believe that monocular videos of humans have the potential to provide a particularly good source of supervision because richer information about shape and appearance can be potentially extracted by dynamic reconstruction that leverages priors such as temporal smoothness, flow, and correspondences, which has been demonstrated in Chapter 5 and more recent work [78,190,211,228–231].

Personalization and reconstruction from sparse sensing. Personalization refers to the process of adapting a generic model to the configuration of a particular subject and outfit. In this thesis, personalization is equal to the network training and preprocessing steps such as the creation of mesh templates from the multi-view capture. In the future, we also would like to relax the requirement on multi-view capture for personalization, and ideally achieve this goal from very sparse sensing input, because it seems a necessary condition to make the avatars accessible to the general public. In our work, the fine-tuning process in the novel capture environment using three RGB-D views in Section 6.6.4 can be regarded as an attempt towards adapting to the new illumination and appearance change over time. The creation of personalized avatars from sparse input such as an RGB(-D) video has been investigated in some previous work, but they either focus on head-only avatar [25,248,254], or are limited in the level of detail or drivability for clothing [3,54,81,82]. Text-conditioned diffusion models have recently demonstrated an impressive power of generative modeling, and the personalization of diffusion models based on DreamBooth [166] and Score Distillation Sampling [156] has also received a lot of attention [74,241].

It is also an interesting direction to continue the work on faithful reconstruction from sparse sensing beyond our Drivable Avatar Clothing work in Chapter 6, especially under the guidance of a universal prior model. A highly desirable ability of such methods is to work with only a forward process without (many) iterations of optimization for the consideration of performance (our method in Chapter 6 uses only 3 iterations). One successful example comes from the recent work on generative models for the human face in EG3D [33], which has been exploited to reconstruct a photorealistic 3D talking head from a monocular video input [196]. Some work has also taken steps in this direction for full-body avatars using generalizable NeRF [57, 98, 134], which faces a greater challenge due to the large variance of pose, shape and appearance in clothed humans. We believe that much larger datasets are needed to fully solve this problem than what is currently available, such as ZJU-Mocap [151] and RDDC [67].

Inverse physics for dynamic humans. In Chapter 4, we made an attempt to integrate the physics of clothing dynamics into the avatar modeling framework. Physics are the first principle that determines how human bodies move in the 3D world, how clothing deforms in reaction to the underlying body, and how light interacts with their appearance to form an image. In order to fundamentally understand human behavior and appearance, we would like to fully invert the physics behind an observation of human performance, for example, a sequence of 4D scans or a monocular video. We are interested in such a problem with two goals in mind. First, we will have full control to *synthesize* or render completely realistic human under arbitrary body motion in any environment without worrying about generalization to out-out-distribution input; second, it will allow us to *analyze* or reconstruct human behavior and appearance given limited input such as a monocular video, where missing information can be filled in by physics.

Fortunately, in recent years, a lot of progress has been made in differentiable physics[1]. First, there has been work that tries to synthesize physically plausible skeleton motion from visual input [124,227]. Second, several versions of differentiable cloth simulation have been developed, such as those based on Projective Dynamics [110] and XPBD [185]. In the meantime, some technical challenges in simulating deformable clothing have been addressed, such as collision handling [107]. Lastly, differentiable physics-based rendering has become increasingly mature over the past few years [140, 243]. Such progress makes the goal of inverting every aspect of digital humans seem more achievable.

Nevertheless, inverse physics is an inherently high-dimensional problem that is hard to solve because of the requirement for good initialization and noise in gradient estimation. We believe this is where data-driven methods and vision-based techniques can come into play. For example, the reconstructed clothing shape from a monocular video input (Chapter 5) can serve as the initial value to be further optimized via a differentiable simulator and renderer. As another example, the registration output of a T-shirt captured with a printed pattern [71] can provide a stronger form of supervision with correspondences than using only the chamfer distance between a simulated garment and multi-view reconstruction. For these reasons, we believe it is beneficial to incorporate both techniques to leverage their advantages.

[1] This term usually refers to the simulation of mechanics, but I am abusing the term to include the simulation of optics (rendering) as well.

Bibliography

[1] T. Alldieck, M. Magnor, B. L. Bhatnagar, C. Theobalt, and G. Pons-Moll. Learning to reconstruct people in clothing from a single rgb camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[2] T. Alldieck, M. Magnor, W. Xu, C. Theobalt, and G. Pons-Moll. Detailed human avatars from monocular video. In *2018 International Conference on 3D Vision (3DV)*, 2018.

[3] T. Alldieck, M. Magnor, W. Xu, C. Theobalt, and G. Pons-Moll. Video based reconstruction of 3d people models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[4] T. Alldieck, G. Pons-Moll, C. Theobalt, and M. Magnor. Tex2shape: Detailed full human body geometry from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[5] R. Alp Güler, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[6] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *ACM Transactions on Graphics (TOG)*, 24(3), 2005.

[7] T. Bagautdinov, C. Wu, T. Simon, F. Prada, T. Shiratori, S.-E. Wei, W. Xu, Y. Sheikh, and J. Saragih. Driving-signal aware full-body avatars. *ACM Transactions on Graphics (TOG)*, 40(4), 2021.

[8] S. Bang, M. Korosteleva, and S.-H. Lee. Estimating garment patterns from static scan data. In *Computer Graphics Forum*, volume 40, 2021.

[9] D. Baraff and A. Witkin. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998.

[10] H. Bertiche, M. Madadi, and S. Escalera. Pbns: physically based neural simulation for unsupervised garment pose space deformation. *ACM Transactions on Graphics (TOG)*, 40(6), 2021.

[11] H. Bertiche, M. Madadi, E. Tylson, and S. Escalera. Deepsd: Automatic deep skinning and pose space deformation for 3d garment animation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[12] B. L. Bhatnagar, C. Sminchisescu, C. Theobalt, and G. Pons-Moll. Loopreg: Self-supervised learning of implicit surface correspondences, pose and shape for 3d human mesh registration. *Advances in Neural Information Processing Systems*, 33, 2020.

[13] B. L. Bhatnagar, G. Tiwari, C. Theobalt, and G. Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[14] S. Bi, S. Lombardi, S. Saito, T. Simon, S.-E. Wei, K. Mcphail, R. Ramamoorthi, Y. Sheikh, and J. Saragih. Deep relightable appearance models for animatable faces. *ACM Transactions on Graphics (TOG)*, 40(4), 2021.

[15] S. Bi, S. Lombardi, S. Saito, T. Simon, S.-E. Wei, K. Mcphail, R. Ramamoorthi, Y. Sheikh, and J. Saragih. Deep relightable appearance models for animatable faces. *ACM Transactions on Graphics (TOG)*, 40(4), 2021.

[16] 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 *European Conference on Computer Vision*, 2016.

[17] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black. Dynamic faust: Registering human bodies in motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[18] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black. Dynamic faust: Registering human bodies in motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[19] S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly. Projective dynamics: Fusing constraint projections for fast simulation. *ACM Transactions on Graphics (TOG)*, 33(4), 2014.

[20] A. Bozic, P. Palafox, M. Zollhöfer, A. Dai, J. Thies, and M. Nießner. Neural non-rigid tracking. *Advances in Neural Information Processing Systems*, 33, 2020.

[21] A. Bozic, P. Palafox, M. Zollhofer, J. Thies, A. Dai, and M. Nießner. Neural deformation graphs for globally-consistent non-rigid reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[22] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur. Markerless garment capture. *ACM Transactions on Graphics (TOG)*, 27(3), 2008.

[23] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. In *SIGGRAPH 2002 Conference Papers*, 2002.

[24] A. Burov, M. Nießner, and J. Thies. Dynamic surface function networks for clothed human bodies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[25] C. Cao, T. Simon, J. K. Kim, G. Schwartz, M. Zollhoefer, S.-S. Saito, S. Lombardi, S.-E. Wei, D. Belko, S.-I. Yu, et al. Authentic volumetric avatars from a phone scan. *ACM Transactions on Graphics (TOG)*, 41(4), 2022.

[26] Y. Cao, Y.-P. Cao, K. Han, Y. Shan, and K.-Y. K. Wong. Dreamavatar: Text-and-shape guided 3d human avatar generation via diffusion models. *arXiv preprint arXiv:2304.00916*, 2023.

[27] Z. Cao, G. H. Martinez, T. Simon, S.-E. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[28] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[29] J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. *ACM Transactions on Graphics (TOG)*, 22(3), 2003.

[30] C. Castillo, J. López-Moreno, and C. Aliaga. Recent advances in fabric appearance reproduction. *Computers & Graphics*, 84, 2019.

[31] C.-F. Chabert, P. Einarsson, A. Jones, B. Lamond, W.-C. Ma, S. Sylwan, T. Hawkins, and P. Debevec. Relighting human locomotion with flowed reflectance fields. In *Eurographics Symposium on Rendering*, 2006.

[32] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros. Everybody dance now. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[33] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. De Mello, O. Gallo, L. J. Guibas, J. Tremblay, S. Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[34] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[35] M. Chen, J. Zhang, X. Xu, L. Liu, Y. Cai, J. Feng, and S. Yan. Geometry-guided progressive nerf for generalizable and efficient neural human rendering. In *European Conference on Computer Vision*, 2022.

[36] W. Chen, H. Ling, J. Gao, E. Smith, J. Lehtinen, A. Jacobson, and S. Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[37] X. Chen, B. Zhou, F.-X. Lu, L. Wang, L. Bi, and P. Tan. Garment modeling with a depth camera. *ACM Transactions on Graphics (TOG)*, 34(6), 2015.

[38] X. Chen, A. Pang, W. Yang, P. Wang, L. Xu, and J. Yu. Tightcap: 3d human shape capture with clothing tightness field. *ACM Transactions on Graphics (TOG)*, 41(1), 2021.

[39] X. Chen, T. Jiang, J. Song, J. Yang, M. J. Black, A. Geiger, and O. Hilliges. gdna: Towards generative detailed neural avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[40] N. Chentanez, M. Macklin, M. Müller, S. Jeschke, and T.-Y. Kim. Cloth and skin deformation with a triangle mesh based convolutional neural network. In *Computer Graphics Forum*, volume 39, 2020.

[41] E. Corona, G. Pons-Moll, G. Alenyà, and F. Moreno-Noguer. Learned vertex descent: a new direction for 3d human model fitting. In *European Conference on Computer Vision*, 2022.

[42] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH 1996 Conference Papers*, 1996.

[43] K. Daubert, H. Lensch, W. Heidrich, and H.-P. Seidel. Efficient cloth modeling and rendering. In *Eurographics Workshop on Rendering Techniques*, 2001.

[44] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM Transactions on Graphics (TOG)*, 27(3), 2008.

[45] L. De Luigi, R. Li, B. Guillard, M. Salzmann, and P. Fua. Drapenet: Garment generation and self-supervised draping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[46] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH 1996 Conference Papers*, 1996.

[47] W. Dong, Y. Lao, M. Kaess, and V. Koltun. Ash: A modern framework for parallel spatial hashing in 3d perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[48] Z. Dong, X. Chen, J. Yang, M. J. Black, O. Hilliges, and A. Geiger. Ag3d: Learning to generate 3d avatars from 2d image collections. *arXiv preprint arXiv:2305.02312*, 2023.

[49] M. Dou, P. Davidson, S. R. Fanello, S. Khamis, A. Kowdle, C. Rhemann, V. Tankovich, and S. Izadi. Motion2fusion: Real-time volumetric performance capture. *ACM Transactions on Graphics (TOG)*, 36(6), 2017.

[50] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (TOG)*, 35(4), 2016.

[51] M. Eisemann, B. De Decker, M. Magnor, P. Bekaert, E. De Aguiar, N. Ahmed, C. Theobalt, and A. Sellent. Floating textures. In *Computer Graphics Forum*, volume 27, 2008.

[52] R. Feinman. Pytorch-minimize: a library for numerical optimization with autograd, 2021.

[53] W. Feng, J. Zhang, H. Cai, H. Xu, J. Hou, and H. Bao. Recurrent multi-view alignment network for unsupervised surface registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[54] Y. Feng, J. Yang, M. Pollefeys, M. J. Black, and T. Bolkart. Capturing and animation of body and clothing from monocular video. In *SIGGRAPH Asia 2022 Conference Papers*, 2022.

[55] M. Fratarcangeli, V. Tibaldo, and F. Pellacini. Vivace: A practical gauss-seidel method for stable soft body dynamics. *ACM Transactions on Graphics (TOG)*, 35(6), 2016.

[56] S. Galliani, K. Lasinger, and K. Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[57] Q. Gao, Y. Wang, L. Liu, L. Liu, C. Theobalt, and B. Chen. Neural novel actor: Learning a generalized animatable neural representation for human actors. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[58] X. Gao, J. Yang, J. Kim, S. Peng, Z. Liu, and X. Tong. Mps-nerf: Generalizable 3d human rendering from multiview images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[59] P. Garrido, L. Valgaert, C. Wu, and C. Theobalt. Reconstructing detailed dynamic face geometry from monocular video. *ACM Transactions on Graphics (TOG)*, 32(6), 2013.

[60] E. Gärtner, L. Metz, M. Andriluka, C. D. Freeman, and C. Sminchisescu. Transformer-based learned optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[61] K. Gong, Y. Gao, X. Liang, X. Shen, M. Wang, and L. Lin. Graphonomy: Universal human parsing via graph transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[62] A. Grigorev, K. Iskakov, A. Ianina, R. Bashirov, I. Zakharkin, A. Vakhitov, and V. Lempitsky. Stylepeople: A generative model of fullbody human avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[63] R. A. Guler and I. Kokkinos. Holopose: Holistic 3d human reconstruction in-the-wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[64] E. Gundogdu, V. Constantin, A. Seifoddini, M. Dang, M. Salzmann, and P. Fua. Garnet: A two-stream network for fast and accurate 3d cloth draping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[65] J. Guo, J. Li, R. Narain, and H. S. Park. Inverse simulation: Reconstructing dynamic geometry of clothed humans via optimal control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[66] K. Guo, F. Xu, Y. Wang, Y. Liu, and Q. Dai. Robust non-rigid motion tracking and surface reconstruction using l0 regularization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[67] M. Habermann, L. Liu, W. Xu, M. Zollhoefer, G. Pons-Moll, and C. Theobalt. Real-time deep dynamic characters. *ACM Transactions on Graphics (TOG)*, 40(4), 2021.

[68] M. Habermann, W. Xu, M. Zollhoefer, G. Pons-Moll, and C. Theobalt. Livecap: Real-time human performance capture from monocular video. *ACM Transactions On Graphics (TOG)*, 38(2), 2019.

[69] M. Habermann, W. Xu, M. Zollhoefer, G. Pons-Moll, and C. Theobalt. A deeper look into deepcap. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[70] M. Habermann, W. Xu, M. Zollhofer, G. Pons-Moll, and C. Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[71] O. Halimi, T. Stuyck, D. Xiang, T. Bagautdinov, H. Wen, R. Kimmel, T. Shiratori, C. Wu, Y. Sheikh, and F. Prada. Pattern-based cloth registration and sparse-view animation. *ACM Transactions on Graphics (TOG)*, 41(6), 2022.

[72] D. Holden, B. C. Duong, S. Datta, and D. Nowrouzezahrai. Subspace neural physics: Fast data-driven interactive simulation. In *Proceedings of the 18th annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2019.

[73] F. Hong, L. Pan, Z. Cai, and Z. Liu. Garment4d: Garment reconstruction from point cloud sequences. *Advances in Neural Information Processing Systems*, 34, 2021.

[74] Y. Huang, H. Yi, Y. Xiu, T. Liao, J. Tang, D. Cai, and J. Thies. Tech: Text-guided reconstruction of lifelike clothed humans. *arXiv preprint arXiv:2308.08545*, 2023.

[75] Z. Huang, Y. Xu, C. Lassner, H. Li, and T. Tung. Arch: Animatable reconstruction of clothed humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[76] M. Işık, M. Rünz, M. Georgopoulos, T. Khakhulin, J. Starck, L. Agapito, and M. Nießner. Humanrf: High-fidelity neural radiance fields for humans in motion. *ACM Transactions on Graphics (TOG)*, 42(4), 2023.

[77] A. Jacobson, E. Tosun, O. Sorkine, and D. Zorin. Mixed finite elements for variational surface modeling. In *Computer Graphics Forum*, volume 29, 2010.

[78] Y. Jafarian and H. S. Park. Learning high fidelity depths of dressed humans by watching social media dance videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[79] A. Jain, T. Thormählen, H.-P. Seidel, and C. Theobalt. Moviereshape: Tracking and reshaping of humans in videos. *ACM Transactions on Graphics (TOG)*, 29(6), 2010.

[80] K. M. Jatavallabhula, E. Smith, J.-F. Lafleche, C. F. Tsang, A. Rozantsev, W. Chen, and T. Xiang. Kaolin: A pytorch library for accelerating 3d deep learning research. *arXiv preprint arXiv:1911.05063*, 2019.

[81] B. Jiang, Y. Hong, H. Bao, and J. Zhang. Selfrecon: Self reconstruction your digital avatar from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[82] T. Jiang, X. Chen, J. Song, and O. Hilliges. Instantavatar: Learning avatars from monocular video in 60 seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[83] J. Jiménez, X. Wu, A. Pesce, and A. Jarabo. Practical real-time strategies for accurate indirect occlusion. *SIGGRAPH 2016 Courses: Physically Based Shading in Theory and Practice*, 2016.

[84] N. Jin, Y. Zhu, Z. Geng, and R. Fedkiw. A pixel-based framework for data-driven clothing. In *Computer Graphics Forum*, volume 39, 2020.

[85] H. Joo, T. Simon, and Y. Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[86] J. T. Kajiya. The rendering equation. In *SIGGRAPH 1986 Conference Papers*, 1986.

[87] 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.

[88] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik. Learning 3d human dynamics from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[89] L. Kavan, S. Collins, J. Žára, and C. O'Sullivan. Geometric skinning with approximate dual quaternion blending. *ACM Transactions on Graphics (TOG)*, 27(4), 2008.

[90] L. Kavan and J. Žára. Spherical blend skinning: a real-time deformation of articulated models. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games*, 2005.

[91] D. Kim, W. Koh, R. Narain, K. Fatahalian, A. Treuille, and J. F. O'Brien. Near-exhaustive precomputation of secondary cloth effects. *ACM Transactions on Graphics (TOG)*, 32(4), 2013.

[92] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[93] A. Kirillov, Y. Wu, K. He, and R. Girshick. Pointrend: Image segmentation as rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[94] M. Kocabas, N. Athanasiou, and M. J. Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[95] N. Kolotouros, T. Alldieck, A. Zanfir, E. G. Bazavan, M. Fieraru, and C. Sminchisescu. Dreamhuman: Animatable 3d avatars from text. *arXiv preprint arXiv:2306.09329*, 2023.

[96] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[97] Y. Kwon, D. Kim, D. Ceylan, and H. Fuchs. Neural human performer: Learning generalizable radiance fields for human performance rendering. *Advances in Neural Information Processing Systems*, 34, 2021.

[98] Y. Kwon, D. Kim, D. Ceylan, and H. Fuchs. Neural image-based avatars: Generalizable radiance fields for human avatar modeling. In *International Conference on Learning Representations*, 2023.

[99] E. P. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg. Non-linear approximation of reflectance functions. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997.

[100] Z. Lahner, D. Cremers, and T. Tung. Deepwrinkles: Accurate and realistic clothing modeling. In *European Conference on Computer Vision*, 2018.

[101] J. Lawrence, D. Goldman, S. Achar, G. M. Blascovich, J. G. Desloge, T. Fortes, E. M. Gomez, S. Häberling, H. Hoppe, A. Huibers, et al. Project starline: a high-fidelity telepresence system. *ACM Transactions on Graphics (TOG)*, 40(6), 2021.

[102] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH 2000 Conference Papers*, 2000.

[103] H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Computer graphics forum*, volume 27, 2008.

[104] J. Li, G. Daviet, R. Narain, F. Bertails-Descoubes, M. Overby, G. E. Brown, and L. Boissieux. An implicit frictional contact solver for adaptive cloth simulation. *ACM Transactions on Graphics (TOG)*, 37(4), 2018.

[105] J. Li, S. Saito, T. Simon, S. Lombardi, H. Li, and J. Saragih. Megane: Morphable eyeglass and avatar network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[106] M. Li, D. M. Kaufman, and C. Jiang. Codimensional incremental potential contact. *ACM Transactions on Graphics (TOG)*, 40(4), 2021.

[107] M. Li, D. M. Kaufman, and C. Jiang. Codimensional incremental potential contact. *ACM Transactions on Graphics (TOG)*, 40(4), 2021.

[108] R. Li, Y. Xiu, S. Saito, Z. Huang, K. Olszewski, and H. Li. Monocular real-time volumetric performance capture. In *European Conference on Computer Vision*, 2020.

[109] Y. Li, A. Bozic, T. Zhang, Y. Ji, T. Harada, and M. Nießner. Learning to optimize non-rigid tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[110] Y. Li, T. Du, K. Wu, J. Xu, and W. Matusik. Diffcloth: Differentiable cloth simulation with dry frictional contact. *ACM Transactions on Graphics (TOG)*, 2022.

[111] Y. Li, M. Habermann, B. Thomaszewski, S. Coros, T. Beeler, and C. Theobalt. Deep physics-aware inference of cloth deformation for monocular human performance capture. In *2021 International Conference on 3D Vision (3DV)*, 2021.

[112] J. Liang, M. Lin, and V. Koltun. Differentiable cloth simulation for inverse problems. *Advances in Neural Information Processing Systems*, 32, 2019.

[113] H. Lin, S. Peng, Z. Xu, Y. Yan, Q. Shuai, H. Bao, and X. Zhou. Efficient neural radiance fields for interactive free-viewpoint video. In *SIGGRAPH Asia 2022 Conference Papers*, 2022.

[114] 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 *European Conference on Computer Vision*, 2014.

[115] L. Liu, M. Habermann, V. Rudnev, K. Sarkar, J. Gu, and C. Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM Transactions on Graphics (TOG)*, 40(6), 2021.

[116] S. Liu, T. Li, W. Chen, and H. Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[117] S. Lombardi, J. Saragih, T. Simon, and Y. Sheikh. Deep appearance models for face rendering. *ACM Transactions on Graphics (TOG)*, 37(4), 2018.

[118] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh. Neural volumes: learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG)*, 38(4), 2019.

[119] S. Lombardi, T. Simon, G. Schwartz, M. Zollhoefer, Y. Sheikh, and J. Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (TOG)*, 40(4), 2021.

[120] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6), 2015.

[121] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH 1987 Conference Papers*. 1987.

[122] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.

[123] C. Luo, X. Chu, and A. Yuille. Orinet: A fully convolutional network for 3d human pose estimation. In *BMVC*, 2018.

[124] Z. Luo, J. Cao, A. Winkler, K. Kitani, and W. Xu. Perpetual humanoid control for real-time simulated avatars. *arXiv preprint arXiv:2305.06456*, 2023.

[125] M. Ly, J. Jouve, L. Boissieux, and F. Bertails-Descoubes. Projective dynamics with dry frictional contact. *ACM Transactions on Graphics (TOG)*, 39(4), 2020.

[126] Q. Ma, S. Saito, J. Yang, S. Tang, and M. J. Black. Scale: Modeling clothed humans with a surface codec of articulated local elements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[127] Q. Ma, J. Yang, M. J. Black, and S. Tang. Neural point-based shape modeling of humans in challenging clothing. In *2022 International Conference on 3D Vision (3DV)*, 2022.

[128] Q. Ma, J. Yang, A. Ranjan, S. Pujades, G. Pons-Moll, S. Tang, and M. J. Black. Learning to dress 3d people in generative clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[129] Q. Ma, J. Yang, S. Tang, and M. J. Black. The power of points for modeling humans in clothing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[130] M. Macklin, M. Müller, and N. Chentanez. Xpbd: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*, 2016.

[131] R. Martin-Brualla, R. Pandey, S. Yang, P. Pidlypenskyi, J. Taylor, J. Valentin, S. Khamis, P. Davidson, A. Tkach, P. Lincoln, et al. Lookingood: enhancing performance capture with real-time neural re-rendering. *ACM Transactions on Graphics (TOG)*, 37(6), 2018.

[132] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4), 2017.

[133] E. Miguel, D. Bradley, B. Thomaszewski, B. Bickel, W. Matusik, M. A. Otaduy, and S. Marschner. Data-driven estimation of cloth simulation models. In *Computer Graphics Forum*, volume 31, 2012.

[134] M. Mihajlovic, A. Bansal, M. Zollhoefer, S. Tang, and S. Saito. Keypointnerf: Generalizing image-based volumetric avatars using relative spatial encoding of keypoints. In *European Conference on Computer Vision*, 2022.

[135] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020.

[136] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2), 2007.

[137] R. Natsume, S. Saito, Z. Huang, W. Chen, C. Ma, H. Li, and S. Morishima. Siclope: Silhouette-based clothed people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[138] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[139] P. Nguyen-Ha, N. Sarafianos, C. Lassner, J. Heikkilä, and T. Tung. Free-viewpoint rgb-d human performance capture and rendering. In *European Conference on Computer Vision*, 2022.

[140] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (TOG)*, 38(6), 2019.

[141] A. Noguchi, X. Sun, S. Lin, and T. Harada. Neural articulated radiance field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[142] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *2018 International Conference on 3D Vision (3DV)*, 2018.

[143] A. A. Osman, T. Bolkart, and M. J. Black. Star: Sparse trained articulated human body regressor. In *European Conference on Computer Vision*, 2020.

[144] J. Park, Q.-Y. Zhou, and V. Koltun. Colored point cloud registration revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[145] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

[146] C. Patel, Z. Liao, and G. Pons-Moll. Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[147] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[148] G. Pavlakos, N. Kolotouros, and K. Daniilidis. Texturepose: Supervising human mesh estimation with texture consistency. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[149] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[150] S. Peng, J. Dong, Q. Wang, S. Zhang, Q. Shuai, X. Zhou, and H. Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[151] S. Peng, Y. Zhang, Y. Xu, Q. Wang, Q. Shuai, H. Bao, and X. Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[152] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021.

[153] N. Pietroni, C. Dumery, R. Falque, M. Liu, T. Vidal-Calleja, and O. Sorkine-Hornung. Computational pattern making from 3d garment models. *ACM Transactions on Graphics (TOG)*, 41(4), 2022.

[154] L. Pineda, T. Fan, M. Monge, S. Venkataraman, P. Sodhi, R. T. Chen, J. Ortiz, D. DeTone, A. Wang, S. Anderson, et al. Theseus: A library for differentiable nonlinear optimization. *Advances in Neural Information Processing Systems*, 35, 2022.

[155] G. Pons-Moll, S. Pujades, S. Hu, and M. J. Black. Clothcap: Seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics (TOG)*, 36(4), 2017.

[156] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2022.

[157] D. Pritchard and W. Heidrich. Cloth motion capture. In *Computer Graphics Forum*, volume 22, 2003.

[158] S. Prokudin, M. J. Black, and J. Romero. Smplpix: Neural avatars from 3d human models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[159] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017.

[160] A. Raj, J. Tanke, J. Hays, M. Vo, C. Stoll, and C. Lassner. Anr: Articulated neural rendering for virtual avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[161] E. Remelli, T. Bagautdinov, S. Saito, C. Wu, T. Simon, S.-E. Wei, K. Guo, Z. Cao, F. Prada, J. Saragih, et al. Drivable volumetric avatars using texel-aligned features. In *SIGGRAPH 2022 Conference Papers*, 2022.

[162] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Deepmatching: Hierarchical deformable dense matching. *International Journal of Computer Vision*, 120(3), 2016.

[163] N. Robertini, D. Casas, H. Rhodin, H.-P. Seidel, and C. Theobalt. Model-based outdoor performance capture. In *2016 International Conference on 3D Vision (3DV)*, 2016.

[164] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[165] R. A. Rosu, S. Saito, Z. Wang, C. Wu, S. Behnke, and G. Nam. Neural strands: Learning hair geometry and appearance from multi-view images. In *European Conference on Computer Vision*. Springer, 2022.

[166] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[167] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[168] 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.

[169] S. Saito, J. Yang, Q. Ma, and M. J. Black. Scanimate: Weakly supervised learning of skinned clothed avatar networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[170] I. Santesteban, M. A. Otaduy, and D. Casas. Learning-based animation of clothing for virtual try-on. In *Computer Graphics Forum*, volume 38, 2019.

[171] I. Santesteban, M. A. Otaduy, and D. Casas. Snug: Self-supervised neural dynamic garments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[172] I. Santesteban, N. Thuerey, M. A. Otaduy, and D. Casas. Self-supervised collision handling via generative 3d garment models for virtual try-on. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[173] V. Scholz, T. Stich, M. Keckeisen, M. Wacker, and M. Magnor. Garment motion capture using color-coded patterns. In *Computer Graphics Forum*, volume 24, 2005.

[174] R. Shao, L. Chen, Z. Zheng, H. Zhang, Y. Zhang, H. Huang, Y. Guo, and Y. Liu. Floren: Real-time high-quality human performance rendering via appearance flow using sparse rgb cameras. In *SIGGRAPH Asia 2022 Conference Papers*, 2022.

[175] R. Shao, H. Zhang, H. Zhang, M. Chen, Y.-P. Cao, T. Yu, and Y. Liu. Doublefield: Bridging the neural surface and radiance fields for high-fidelity human reconstruction and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[176] R. Shao, Z. Zheng, H. Tu, B. Liu, H. Zhang, and Y. Liu. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[177] Y. Shen, S. Saito, Z. Wang, O. Maury, C. Wu, J. Hodgins, Y. Zheng, and G. Nam. Ct2hair: High-fidelity 3d hair modeling using computed tomography. *ACM Transactions on Graphics (TOG)*, 42(4), 2023.

[178] A. Shysheya, E. Zakharov, K.-A. Aliev, R. Bashirov, E. Burkov, K. Iskakov, A. Ivakhnenko, Y. Malkov, I. Pasechnik, D. Ulyanov, et al. Textured neural avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[179] L. Sigal, M. Mahler, S. Diaz, K. McIntosh, E. Carter, T. Richards, and J. Hodgins. A perceptual control space for garment simulation. *ACM Transactions on Graphics (TOG)*, 34(4), 2015.

[180] J. Song, X. Chen, and O. Hilliges. Human body model fitting by learned gradient descent. In *European Conference on Computer Vision*, 2020.

[181] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing*, 2007.

[182] J. Starck and A. Hilton. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, 27(3), 2007.

[183] C. Stoll, J. Gall, E. De Aguiar, S. Thrun, and C. Theobalt. Video-based reconstruction of animatable human characters. *ACM Transactions on Graphics (TOG)*, 29(6), 2010.

[184] T. Stuyck. Cloth simulation for computer graphics. *Synthesis Lectures on Visual Computing: Computer Graphics, Animation, Computational Photography, and Imaging*, 10(3):1–121, 2018.

[185] T. Stuyck. Diffxpbd: Differentiable position-based simulation of compliant constraint dynamics. *arXiv preprint arXiv:2301.01396*, 2023.

[186] S.-Y. Su, F. Yu, M. Zollhöfer, and H. Rhodin. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. *Advances in Neural Information Processing Systems*, 34, 2021.

[187] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. *ACM Transactions on Graphics (TOG)*, 26(3), 2007.

[188] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei. Integral human pose regression. In *European Conference on Computer Vision*, 2018.

[189] X. Suo, Y. Jiang, P. Lin, Y. Zhang, M. Wu, K. Guo, and L. Xu. Neuralhumanfvv: Real-time neural volumetric human performance rendering using rgb cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021.

[190] J. Tan, G. Yang, and D. Ramanan. Distilling neural fields for real-time articulated shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[191] M. Tan, R. Pang, and Q. V. Le. Efficientdet: Scalable and efficient object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[192] Q. Tan, Y. Zhou, T. Wang, D. Ceylan, X. Sun, and D. Manocha. A repulsive force unit for garment collision handling in neural networks. In *European Conference on Computer Vision*, 2022.

[193] S. Tang, F. Tan, K. Cheng, Z. Li, S. Zhu, and P. Tan. A neural network for detailed human depth estimation from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[194] A. Tewari, M. Zollhofer, H. Kim, P. Garrido, F. Bernard, P. Perez, and C. Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[195] J. Thies, M. Zollhöfer, and M. Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4), 2019.

[196] A. Trevithick, M. Chan, M. Stengel, E. Chan, C. Liu, Z. Yu, S. Khamis, M. Chandraker, R. Ramamoorthi, and K. Nagano. Real-time radiance fields for single-image portrait view synthesis. *ACM Transactions on Graphics (TOG)*, 42(4), 2023.

[197] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *European Conference on Computer Vision*, 2018.

[198] R. Vidaurre, I. Santesteban, E. Garces, and D. Casas. Fully convolutional graph neural networks for parametric virtual try-on. In *Computer Graphics Forum*, volume 39, 2020.

[199] M. Volino, D. Casas, J. P. Collomosse, and A. Hilton. Optimal representation of multi-view video. In *Proceedings of British Machine Vision Conference*, 2014.

[200] J. Vongkulbhisal, F. De la Torre, and J. P. Costeira. Discriminative optimization: Theory and applications to computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[201] A. Walsman, W. Wan, T. Schmidt, and D. Fox. Dynamic high resolution deformable articulated tracking. In *2017 International Conference on 3D Vision (3DV)*, 2017.

[202] H. Wang, F. Hecht, R. Ramamoorthi, and J. F. O'Brien. Example-based wrinkle synthesis for clothing animation. *ACM Transactions on Graphics (TOG)*, 29(4), 2010.

[203] H. Wang, J. F. O'Brien, and R. Ramamoorthi. Data-driven elastic models for cloth: modeling and measurement. *ACM Transactions on Graphics (TOG)*, 30(4), 2011.

[204] Q. Wang, Z. Wang, K. Genova, P. P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[205] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro. Video-to-video synthesis. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[206] T. Y. Wang, D. Ceylan, K. K. Singh, and N. J. Mitra. Dance in the wild: Monocular human animation with neural dynamic appearance synthesis. In *2021 International Conference on 3D Vision (3DV)*, 2021.

[207] Y. Wang, X. Tao, X. Qi, X. Shen, and J. Jia. Image inpainting via generative multi-column convolutional neural networks. *Advances in neural information processing systems*, 31, 2018.

[208] Z. Wang, G. Nam, T. Stuyck, S. Lombardi, C. Cao, J. Saragih, M. Zollhöfer, J. Hodgins, and C. Lassner. Neuwigs: A neural dynamic model for volumetric hair capture and animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[209] Z. Wang, G. Nam, T. Stuyck, S. Lombardi, M. Zollhöfer, J. Hodgins, and C. Lassner. Hvh: Learning a hybrid neural volumetric representation for dynamic hair performance capture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[210] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[211] C.-Y. Weng, B. Curless, P. P. Srinivasan, J. T. Barron, and I. Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern Recognition*, 2022.

[212] R. White, K. Crane, and D. A. Forsyth. Capturing and animating occluded cloth. *ACM Transactions on Graphics (TOG)*, 26(3), 2007.

[213] C. Wu, T. Shiratori, and Y. Sheikh. Deep incremental learning for efficient high-fidelity face tracking. *ACM Transactions on Graphics (TOG)*, 37(6), 2018.

[214] C. Wu, C. Stoll, L. Valgaerts, and C. Theobalt. On-set performance capture of multiple actors with a stereo camera. *ACM Transactions on Graphics (TOG)*, 32(6), 2013.

[215] K. Wu and C. Yuksel. Real-time cloth rendering with fiber-level detail. *IEEE Transactions on Visualization and Computer Graphics*, 25(2), 2017.

[216] D. Xiang, T. Bagautdinov, T. Stuyck, F. Prada, J. Romero, W. Xu, S. Saito, J. Guo, B. Smith, T. Shiratori, et al. Dressing avatars: Deep photorealistic appearance for physically simulated clothing. *ACM Transactions on Graphics (TOG)*, 41(6), 2022.

[217] D. Xiang, H. Joo, and Y. Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[218] D. Xiang, F. Prada, T. Bagautdinov, W. Xu, Y. Dong, H. Wen, J. Hodgins, and C. Wu. Modeling clothing as a separate layer for an animatable human avatar. *ACM Transactions on Graphics (TOG)*, 40(6), 2021.

[219] D. Xiang, F. Prada, Z. Cao, K. Guo, C. Wu, J. Hodgins, and T. Bagautdinov. Drivable avatar clothing: Faithful full-body telepresence with dynamic clothing driven by sparse rgb-d input. In *SIGGRAPH Asia 2023 Conference Papers*, 2023.

[220] D. Xiang, F. Prada, C. Wu, and J. Hodgins. Monoclothcap: Towards temporally coherent clothing capture from monocular rgb video. In *2020 International Conference on 3D Vision (3DV)*, 2020.

[221] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[222] Y. Xiu, J. Yang, D. Tzionas, and M. J. Black. Icon: Implicit clothed humans obtained from normals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[223] H. Xu, T. Alldieck, and C. Sminchisescu. H-nerf: Neural radiance fields for rendering and temporal reconstruction of humans in motion. *Advances in Neural Information Processing Systems*, 34, 2021.

[224] W. Xu, A. Chatterjee, M. Zollhöfer, H. Rhodin, D. Mehta, H.-P. Seidel, and C. Theobalt. Monoperfcap: Human performance capture from monocular video. *ACM Transactions on Graphics (TOG)*, 37(2), 2018.

[225] Y.-Q. Xu, Y. Chen, S. Lin, H. Zhong, E. Wu, B. Guo, and H.-Y. Shum. Photorealistic rendering of knitwear using the lumislice. In *SIGGRAPH 2001 Conference Papers*, 2001.

[226] Y. Xu, S.-C. Zhu, and T. Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[227] G. Yang, Y. Shuo, Z. Zhang, Z. Manchester, and D. Ramanan. Ppr: Physically plausible reconstruction from monocular videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[228] G. Yang, D. Sun, V. Jampani, D. Vlasic, F. Cole, H. Chang, D. Ramanan, W. T. Freeman, and C. Liu. Lasr: Learning articulated shape reconstruction from a monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[229] G. Yang, D. Sun, V. Jampani, D. Vlasic, F. Cole, C. Liu, and D. Ramanan. Viser: Video-specific surface embeddings for articulated 3d shape reconstruction. *Advances in Neural Information Processing Systems*, 34, 2021.

[230] G. Yang, M. Vo, N. Neverova, D. Ramanan, A. Vedaldi, and H. Joo. Banmo: Building animatable 3d neural models from many casual videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[231] G. Yang, C. Wang, N. D. Reddy, and D. Ramanan. Reconstructing animatable categories from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[232] J. Yang, J.-S. Franco, F. Hétroy-Wheeler, and S. Wuhrer. Analyzing clothing layer deformation statistics of 3d human motions. In *European Conference on Computer Vision*, 2018.

[233] S. Yang, J. Liang, and M. C. Lin. Learning-based cloth material recovery from video. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[234] S. Yang, Z. Pan, T. Amert, K. Wang, L. Yu, T. Berg, and M. C. Lin. Physics-inspired garment recovery from a single-view image. *ACM Transactions on Graphics (TOG)*, 37(5), 2018.

[235] T. Yasuda, S. Yokoi, and J.-i. Toriwaki. A shading model for cloth objects. *IEEE Computer Graphics and Applications*, 12(6), 1992.

[236] A. Yu, V. Ye, M. Tancik, and A. Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[237] T. Yu, Z. Zheng, K. Guo, P. Liu, Q. Dai, and Y. Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[238] T. Yu, Z. Zheng, K. Guo, J. Zhao, Q. Dai, H. Li, G. Pons-Moll, and Y. Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[239] T. Yu, Z. Zheng, Y. Zhong, J. Zhao, Q. Dai, G. Pons-Moll, and Y. Liu. Simulcap: Single-view human performance capture with cloth simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[240] A. Zanfir, E. G. Bazavan, M. Zanfir, W. T. Freeman, R. Sukthankar, and C. Sminchisescu. Neural descent for visual 3d human pose and shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[241] Y. Zeng, Y. Lu, X. Ji, Y. Yao, H. Zhu, and X. Cao. Avatarbooth: High-quality and customizable 3d human avatar generation. *arXiv preprint arXiv:2306.09864*, 2023.

[242] C. Zhang, S. Pujades, M. J. Black, and G. Pons-Moll. Detailed, accurate, human shape estimation from clothed 3d scan sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[243] C. Zhang, B. Miller, K. Yan, I. Gkioulekas, and S. Zhao. Path-space differentiable rendering. *ACM Transactions on Graphics (TOG)*, 39(4), 2020.

[244] J. Zhang, Z. Jiang, D. Yang, H. Xu, Y. Shi, G. Song, Z. Xu, X. Wang, and J. Feng. Avatargen: A 3d generative model for animatable human avatars. *arXiv preprint arXiv:2211.14589*, 2022.

[245] M. Zhang, D. Ceylan, T. Y. Wang, and N. J. Mitra. Dynamic neural garments. *ACM Transactions on Graphics (TOG)*, 40(6), 2021.

[246] M. Zhang, T. Wang, D. Ceylan, and N. J. Mitra. Deep detail enhancement for any garment. In *Computer Graphics Forum*, volume 40, 2021.

[247] F. Zhao, W. Yang, J. Zhang, P. Lin, Y. Zhang, J. Yu, and L. Xu. Humannerf: Efficiently generated human radiance field from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[248] Y. Zheng, V. F. Abrevaya, M. C. Bühler, X. Chen, M. J. Black, and O. Hilliges. Im avatar: Implicit morphable head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[249] Z. Zheng, H. Huang, T. Yu, H. Zhang, Y. Guo, and Y. Liu. Structured local radiance fields for human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[250] Z. Zheng, T. Yu, Y. Wei, Q. Dai, and Y. Liu. Deephuman: 3d human reconstruction from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[251] T. Zhi, C. Lassner, T. Tung, C. Stoll, S. G. Narasimhan, and M. Vo. Texmesh: Reconstructing detailed human texture and geometry from rgb-d video. In *European Conference on Computer Vision*, 2020.

[252] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[253] H. Zhu, X. Zuo, S. Wang, X. Cao, and R. Yang. Detailed human shape estimation from a single image by hierarchical mesh deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[254] W. Zielonka, T. Bolkart, and J. Thies. Instant volumetric head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[255] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, et al. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Transactions on Graphics (TOG)*, 33(4), 2014.