SAPIENZA
UNIVERSITÀ DI ROMA

Towards 3D Virtual Try-On: Clothing Reconstruction and Retargeting with Gaussian Splatting

Facoltà di ingegneria dell'informazione,informatica e statistica

Corso di Laurea Magistrale in Computer Science

Candidate

Andrea Sanchietti

ID number 1883210

Thesis Advisor

Prof. Emanuele Rodola

Co-Advisor

Dr. Riccardo Marin

Academic Year 2023/2024

Thesis defended on 22/10/2024
in front of a Board of Examiners composed by:

Prof. Velardi Paola (chairman)
Prof. Cinque Luigi
Prof. Friolo Daniele
Prof. Galasso Fabio
Prof. Marini Marco Raoul
Prof. Nemmi Eugenio
Prof. Pontarelli Salvatore

Towards 3D Virtual Try-On: Clothing Reconstruction and Retargeting with Gaussian Splatting
Master's thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: sanchietti.1883210@studenti.uniroma1.it

Acknowledgments

Questa tesi rappresenta il culmine di un viaggio straordinario iniziato cinque anni fa, un percorso fatto di impegno, crescita e scoperta, che mi ha condotto attraverso sfide accademiche e personali, incontri significativi e momenti di profonda riflessione. Ogni passo compiuto ha contribuito a formare non solo il mio bagaglio di conoscenze, ma anche la persona che sono oggi, ricordandomi costantemente l'importanza della passione e della tenacia. Nel corso di questo percorso, sono grato a chi mi ha accompagnato e sostenuto, rendendo possibile ogni traguardo e trasformando le difficoltà in occasioni di crescita. In primo luogo desidero esprimere la mia profonda gratitudine al Professor Rodola, per la sua costante disponibilità e per avermi dato l'opportunità di sviluppare questo lavoro. Un sentito ringraziamento va anche a Riccardo Marin, che mi ha seguito con attenzione durante questi mesi, guidandomi, incoraggiandomi, e insegnandomi come portare a termine un complesso lavoro con metodo e costanza. Ringrazio di cuore i miei genitori, che mi hanno sempre sostenuto incondizionatamente e che, con grande pazienza, hanno sopportato le mie preoccupazioni per gli esami. Fin da piccolo, mi hanno cresciuto insegnandomi a essere curioso e a dare sempre il massimo. Se oggi sono qui, è proprio grazie a questi valori che mi avete trasmesso, e che mi hanno reso la persona che sono ora. Grazie anche a mia sorella, che ha perso preziose ore di sonno a causa delle mie chiacchierate notturne. Un pensiero speciale va alla mia famiglia, in particolare a mia nonna Gabriella, zia Paola, nonna Antonella, zia Paola, zio Piero e tutti i miei cari, per il sostegno e per il tempo passato insieme. Un ringraziamento speciale va ad Azzurra, per aver condiviso con me non solo i momenti belli, ma anche quelli difficili. La sua costante fiducia in me e il suo incoraggiamento mi hanno dato la forza di affrontare ogni sfida, permettendomi così di superarle e crescere. Nel corso di questo viaggio, ho avuto il privilegio di incontrare persone straordinarie che mi hanno arricchito e reso questo percorso ancora più speciale. Ringrazio gli Homework Heroes - Luca (Scut), Tommaso (Tom.exe), Alessandro (Gigachad), Juan (El Chico Malo), Diego (Skills), Luca (Strank), Costanza, Alessia, Giulia e Martina - con cui ho condiviso momenti indimenticabili sia in Università che fuori. Un grazie di cuore anche a Riccardo, Lucrezia, Shuya, Lorenzo e Hazem per le ore trascorse insieme in aula studio e per i caffè che ci hanno tenuti svegli e motivati. A Francesco, con cui ho condiviso gli ultimi due anni di studi, va la mia gratitudine per un'amicizia che so durerà per sempre, alimentata dai sogni che condividiamo e dalle ambizioni che ci spingono a guardare al futuro con entusiasmo. Infine, non posso dimenticare i miei amici di sempre - Davide, Gianluca, Carlo, Francesco, Luca, Alberto, Linda e Flavia - che da ormai dieci anni condivido serate in compagnia di una birra e un'immancabile dose di spensieratezza. A tutti voi, grazie di cuore.

Abstract

The rapid growth of online shopping, particularly in the fashion industry, has intensified the demand for innovative solutions that enhance the virtual shopping experience. Traditional methods of purchasing clothing online often fall short due to the challenges of accurately considering fit and appearance without physically trying on garments. This thesis presents a novel 3D Virtual Try-On (3D VTON) framework that addresses these limitations by leveraging Gaussian Splatting for high-quality avatar and garment reconstruction, as well as a specialized approach for clothes retargeting that brings posed avatars into canonical space. Our method enables the seamless reconstruction of realistic 3D avatars and garments from multiview images, overcoming the need for costly scanning equipment and manual work. Additionally, the computational efficiency of our approach allows for real-time interactions, providing users with immediate feedback on their virtual outfits. Through extensive evaluations, we showcase the effectiveness of our method in avatar reconstruction and clothing retargeting, highlighting the advancements it introduces to virtual fitting technologies.

Contents

Chapter 1

Introduction

In the last two decades, online shopping has gradually become a part of our daily lives. Today, almost anything can be bought online, from furniture and vehicles to tickets and electronic devices. One industry that has benefited tremendously from online shopping is the fashion industry. The online sales of clothing have seen a dramatic rise, reaching over 600 billion dollars in revenue and expected to exceed 1.6 trillion by the end of 2030[1]*. However, a fundamental problem has emerged within this context, and, to date, there are no widely available commercial solutions that can be considered definitive. Previously, the only way to purchase clothing was by visiting a physical store, where one could try on clothes of different sizes to find the best fit. With the convenience of online shopping, people have lost the ability to know before buying how a garment looks and feels in real life. E-commerce websites often display pictures of clothing, along with size charts listing measurements for each size, but these tools are often insufficient for making a confident purchase decision.*

With the advent of deep learning, techniques have been developed to overlay an image of clothing onto an image uploaded by the user, allowing for a virtual garment fitting experience. The process of generating a realistic image from a photo of a garment and a subject is called 2D Virtual Try-On (2D VTON). Recently, this field has attracted significant attention from both the research community and commercial applications. The latest works ([28], [50], [148], [67]) have demonstrated the ability to produce realistic results, with execution times of under 10 seconds on high-end graphics cards. While 2D Virtual Try-On has made significant steps towards offering users a view of how a garment might look on them, it remains limited in its ability to fully capture the complexities of real-world interactions. One major limitation of 2D VTON is that it cannot handle changes in viewpoint, posture, or body movements, and users are restricted to a single, static picture, which cannot simulate the dynamic behavior of the garment. Another limitation that these works show is the struggle to maintain the identity of the person. Due to the nature of image-generation techniques, output images often contain artifacts and are influenced by biases in the training data, resulting in body shape alterations. These limitations have led researchers to explore 3D Virtual Try-On (3D VTON), which offers a more immersive and interactive experience by allowing users to view clothing from multiple angles and see it in motion.

The concept of dressing 3D avatars has a long history, especially in the entertainment industry, where character modeling for films and video games has long required detailed and realistic clothing simulation. However, these traditional 3D methods typically rely on artists who sculpt avatars with complex 3D modeling software or

[1] https://www.statista.com/statistics/1298198/market-value-fashion-ecommerce-global/

use expensive equipment to digitalize people, making the process time-consuming and costly. Only recently, with advancements in computer vision and machine learning, researchers began to explore ways to automatically generate and reconstruct avatars and clothing directly from images. This shift aims to provide a more accessible and scalable solution to 3D VTON, enabling realistic and dynamic virtual try-on experiences without the need for handcrafted models or specialized capture hardware.

Despite the great promises of 3D Virtual Try-On, achieving realistic results in real-time brings several key challenges. A robust 3D VTON system must accurately reconstruct both the avatar and garments, capturing details like fabric draping and fit, and then it should be able to quickly retarget garments on any possible human avatar. A key component of 3D VTON is Avatar Reconstruction, as traditional scanning techniques are impractical for everyday users. Recent approaches aim to generate avatars from accessible inputs, such as images or videos. Garment Reconstruction adds further complexity, requiring high-detail 3D models that reflect fabric interaction with the body. Current methods use representations like meshes, sewing patterns, Neural Fields, and Gaussian splatting, but capturing fine details while correctly simulating the garment's dynamic remains an open challenge. Lastly, Clothes Retargeting is a critical task that adapts garments to different bodies and poses. Classic cloth simulation approaches are too slow for real-time applications, thus we need more rapid deformation techniques that keep garment alignment without artifacts, which is hard to accomplish, especially for loose and complex clothes.

Our work addresses these challenges. We propose a 3D Virtual Try-On framework based on Gaussian Splatting which supports realistic avatar and garment reconstruction. Also, we deploy garments retargeting based on avatar transformation in a canonical space. Our method aims to achieve a quick and realistic try-on experience that supports body and garment diversity. Our evaluation showed the potential that our method has to offer a high-quality 3D VTON experience.

The following chapters of this thesis are structured as follows: Chapter 2 provides a comprehensive overview of existing research relevant to avatar reconstruction and virtual try-on technologies. In Chapter 3, we discuss technical frameworks used in our pipeline. Chapter 4: the proposed method is presented. We define our research's objectives and scope, then proceed with a detailed description of our approach. In Chapter 5: we evaluate our method through qualitative and quantitative evaluations. Finally, Chapter 6 summarizes our findings and contributions to the field of 3D Virtual Try-On, discussing potential developments for future research.

Chapter 2

Related Works

The reconstruction of human avatars and clothing from RGB images or videos has garnered significant attention in recent years due to its wide range of applications, including virtual try-on, augmented reality, motion capture, animation, and modeling. However, capturing accurate data for these tasks remains challenging. Currently, two main categories of datasets exist: monocular and multiview captures. Monocular inputs pose significant challenges due to their highly underdetermined nature, making it difficult to accurately model a 3D animatable avatar. Issues such as inaccurate body motion estimation and complex wrinkle deformations further complicate the process. In contrast, multiview captures, while more robust, are costly and tedious to produce, often relying on subject-specific templates and precise registration of input frames, which are difficult to obtain in practice. Thanks to recent advancements in computer vision, new methods are being explored to automate avatar reconstruction without the need for pre-scanning. Current approaches can be categorized by the type of input data (monocular or multiview images/videos), the representation used (implicit, explicit, or hybrid), and whether they aim to reconstruct both the body and clothes in the same representation or separate. This chapter provides an overview of recent advancements in virtual human reconstruction and clothing retargeting, offering a comprehensive review of the current state-of-the-art.

2.1 Avatar Reconstruction

2.1.1 Monocular Reconstruction

Monocular Image *Reconstructing an animatable avatar from a single image is a challenging task due to the lack of depth information and the difficulty of estimating the correct pose and occluded body parts. In recent years, numerous methods have been proposed to tackle these challenges, evolving from simple body pose estimations to more complex full-body reconstructions, including clothing and texture details. Existing methods can be broadly divided into two categories: template-based ([17], [65], [31], [192]) and template-free ([153], [193], [139], [140], [173]) reconstruction. Template-based methods rely on pre-defined body models, which represent the human body, and often fail to capture loose or complex clothing due to their reliance on a fixed body template. On the other hand, Template-free methods do not depend on a predefined body model and are more flexible in representing diverse clothing and body shapes. However, these methods typically suffer from producing less detailed or blurry textures, especially on the unseen or occluded back side of the avatar. Given the lack of data for image-to-avatar reconstruction, a standard method that has been*

used in recent years is to use scans, monocular or multiview video datasets ([2], [63], [100], [161], [12], [182]), and extract single frames from them to evaluate the models.

Early methods in avatar reconstruction primarily focused on recovering 2D joints, but groundbreaking approaches like SMPLify by Bogo et al. [17] and Kanazawa et al. [65] shifted the field towards 3D shape and pose estimation. However, these methods were limited to body and pose parameters, without addressing clothing. To overcome this, Varol et al. introduced BodyNet [153], which was the first to incorporate clothing appearance, using volumetric 3D loss and multi-view re-projection, marking a significant step towards full-body avatar reconstruction. DeepHuman by Zheng et al. [193] introduced the THuman dataset and formulated the reconstruction problem as an image-guided volume-to-volume task, which helped refine volumetric approaches to body and clothing reconstruction.

Methods based on implicit functions recently demonstrated promising results. PIFu [139] presented the concept of learning an implicit function over the 3D space using pixel-aligned image features. This approach was later enhanced by PIFuHD [140], which utilized a multi-level architecture to improve the resolution and accuracy of the reconstructions. In parallel, SMPLicit [31] addressed the problem of clothes representation by learning a latent representation based on an Unsigned Distance Function (UDF). At inference time, it predicted the occupancy volume of clothes on a T-posed SMPL body, extracted the mesh using Marching Cubes, and then posed the avatar using a skinning function. This method also demonstrated early retargeting capabilities, although the output quality remained limited, and proposed a method for representing clothes via latent codes. Further advances came with PAMIR [192], which employed two encoders—one for the input image and one for the associated SMPL parameters—to obtain two feature vectors. These were decoded via an MLP to produce an implicit field from which a mesh was extracted using Marching Cubes. This work focused on improving accuracy by conditioning implicit functions on the parametric body model. To generalize for unseen poses in real-world scenarios, ICON [169] proposed a regression approach that queries shapes from locally extracted features, while ECON [168] extended this by inferring features from 2D images to parametric human bodies via normal estimation and shape completion.

Recently, the use of transformers and diffusion models has introduced new possibilities for high-quality reconstruction. These methods ([186], [187], [173]) try to find a way to reconstruct unseen parts of the avatar, and generate a result that is as much consistent as possible with the input image. Zhang et al. [186] leveraged Vision Transformers to extract global image features and extract tri-plane features using a decoder, improving the performance and scalability of 3D avatar reconstruction pipelines. SIFU [187] introduced a side-view decoupling transformer to process features from the input image and side-view normals of the SMPL-X model. These features were then used to reconstruct the textures and mesh via an MLP, followed by a diffusion-based refinement step to improve the quality of the final avatar. Finally, Human 3Diffusion [173] proposed a fully multi-view diffusion and Gaussian Splatting-based approach, which significantly improved the realism and consistency of avatar reconstruction from multiple viewpoints, pushing the boundaries of what can be achieved from a single image.

Despite the significant progress made in avatar reconstruction from a single image, these methods are still constrained by the limited information available from a single viewpoint. The reliance on minimal input data limits the accuracy and realism of the reconstructions, underscoring the need for further advancements in integrating more robust cues such as multi-view information, richer priors, and improved texture inference techniques.

Monocular Video *When shifting to videos as input, there is more possibility to capture temporal data and more details from a simple RGB video. This is fundamental as captures from expensive scanners do not scale for real-world applications. Alternatively, multiview passive reconstruction from a dense set of static body pose images can be used. However, it is hard for people to stand still for a long time, and so this process is time-consuming and error-prone. Also, consumer RGB-D cameras can be used to scan 3D body models, but these specialized sensors are not as widely available as video.*

In [2] the problem of reconstructing a 3D body model from a monocular video was first faced. This work was based on silhouette projection onto a canonical template and was able to reconstruct an animatable avatar. [165], [126] and [164] explored the application of neural representation to generate novel views of the video subject. Such pipelines either rely on pose-dependent representations or fall short of motion coherency due to frame-independent optimization, making it difficult to generalize to unseen pose sequences realistically. In MonoHuman [183], disentanglement of backward and forward deformation was introduced to separate non-rigid motions from rigid ones, thus generalizing better for unseen poses.

Recent methods [132], [133], [57], [56], [190] implied a pipeline based on initialization of 3D Gaussian Splatting onto the canonical body model, and LBS plus an enhancing module to learn non-rigid deformations. Similarly, [163] and [141] bounded Gaussians to a triangle mesh, and jointly optimized both at the same time. Despite the promising results, all these methods rely on offsets from an underlying body model to represent clothes, which is not a good representation, especially for loose clothes. Additionally, these methods do not enable retargeting as they learn to displace Gaussians from a template body model via neural networks and using that same neural network for a different body shape would give incorrect clothes fit. Additionally, Gaussian Splatting learns implicit properties of materials jointly, making it harder to relight the models. In [158], a completely different approach based on Monte Carlo ray tracing was used to retrieve the intrinsic properties of clothed human avatars including geometry, albedo, material, and environment lighting from only monocular videos, but this process still needs too much time to be applicable in a consumer setting.

2.1.2 Multiview Reconstruction

Multiview Avatar Reconstruction *Multiview reconstruction addresses the challenge of creating an avatar from multiple images or videos ([161], [193], [191], [59], [194], [126], [36], [6]), providing a more comprehensive representation compared to single-view methods. One of the most common 3D representations in this context is the polygon mesh, valued for its compatibility with traditional rendering pipelines. Early works such as Stoll et al. [146] and Guan et al. [45] used textured meshes to reconstruct avatars, animating them via physical simulation. More recently, neural networks have been introduced to simulate deformations and dynamic textures [7], [167], [166], [48]. HDHumans [47] proposed a hybrid approach that combines meshes with Neural Radiance Fields (NeRF) to produce accurate and temporally coherent 3D deforming surfaces. Although mesh-based avatars remain popular, newer methods based on alternative representations have demonstrated superior reconstruction quality, especially for clothes.*

Free-viewpoint video systems, an important part of multiview reconstruction, traditionally required either a dense array of cameras for image-based novel view synthesis [42], [51], or depth sensors for high-quality 3D reconstruction [35], [29]. However, such systems are expensive and only viable in controlled environments due

to their hardware complexity. The recent exploration of implicit neural representations ([106], [117], [144]), has allowed for high-quality novel view synthesis, though these methods generally struggle to model motion. To overcome this limitation, various methods have projected human features from posed space onto canonical space and learned deformation fields to handle non-rigid deformations. For example, in Peng et al. [125] and Animatable NeRF [124] SMPL [98] and NeRF are combined to capture body dynamics by conditioning neural fields with latent codes, while AvatarReX [194] improves reconstruction quality using structured local implicit fields and dynamic feature patches. However, these methods require SMPL registration, which isn't always feasible, and the NeRF-based approaches suffer from long training times.

To eliminate the dependency on a parametric body model, TAVA [85] uses implicit representations to reconstruct shape, appearance, and skinning weights in canonical space, enabling animation and rendering even in out-of-distribution poses, and the representation of animals as well as humans. Other methods, such as Xu et al. [172], have addressed the issue of implicit representations not learning material properties, proposing techniques to extract normals, light visibility, albedo, roughness, and specular information for relightable neural avatars. HumanRF [59] introduced AvatarHQ, a high-quality multiview dataset of humans, and proposed a method for 4D novel synthesis from dynamic scenes, though neural field methods are still not fast enough for real-time rendering due to the time required to march rays through the scene.

Recently, Gaussian Splatting approaches have emerged as a faster and more animation-friendly alternative to neural methods [90], [89], [198], [81], [109]. These methods explicitly learn the avatar in canonical space, which is then deformed based on input pose through Linear Blend Skinning (LBS). Gaussian Splatting techniques benefit from faster rasterization compared to implicit representations, making them more suitable for animations and real-time rendering. For instance, in Animatable Gaussians [90], a character-specific template is first reconstructed from multi-view images. Pose-dependent Gaussian maps are then predicted using StyleUNet [157], with the final avatar synthesized through LBS and differentiable rasterization. This approach was extended in Animatable Relight [89] to include material properties. GaussianBody initializes a set of Gaussians on the SMPL body model, applying LBS to deform the Gaussians and using observations to fine-tune them. Like most of the previously NeRF-based methods, these methods remain limited by their dependence on the SMPL surface, restricting the overall expressiveness of the Gaussians during training.

While video-based reconstruction has gained significant popularity in recent years, avatar reconstruction from sparse multiview images has been less explored. Most works that use sparse image inputs focus on the more general task of reconstructing objects or scenes ([116], [115], [61], [33], [68]. [97], [176]). One of the few works about human reconstruction from few shot images is Octopus [1], whose core lies in a network that learns to encode the images of the person into pose-invariant latent codes, and reconstruct a T-posed avatar. A more recent approach named HaveFun [177] introduced a pipeline specifically designed to reconstruct human avatars from sparse, unconstrained views—where the same subject appears in different poses across images. Their method is built upon DMTet [143], a hybrid representation that combines Signed Distance Functions (SDF) with a tetrahedral grid to model geometry. Given a gallery of images along with SMPL parameters for each, HaveFun queries a canonical model from DMTet, then applies Linear Blend Skinning (LBS) to warp the queried points into the posed space. The model is optimized through a reconstruction loss based on texture, normals, depth, and segmentation masks. To compensate for the lack of views and improve the reconstruction quality, they leverage Zero123 [96],

which generates novel views of the subject, helping to capture previously unseen parts. However, this approach assumes the correct pose and shape parameters are known for each image, and it relies on depth and normal maps to achieve high-quality results.

2.1.3 Garment reconstruction

Clothes reconstruction has become a crucial component of avatar modeling, with various approaches proposed to address the challenge of capturing complex garment shapes, textures, and how they interact with the human body. Broadly, these methods can be classified into three categories: template-based methods ([62], [46], [103], [195], [40], [87]), sewing pattern-based reconstruction ([131], [22], [94], [50], [77], [74], [73]), and neural-based techniques ([83], [128], [32], [79], [91], [24]). The success of these methods often depends on the availability of high-quality datasets. In general, there are two main types of datasets used for training clothing reconstruction models: scans ([52], [161]) and synthetic ([130], [12], [199], [121]) garments.

Template Methods *Template-based methods rely on predefined garment models that are adapted to fit the subject. These approaches typically start with a standard template of garments, such as shirts, pants, or dresses, and then deform or warp them to match the target subject's shape and pose. One of the key challenges in these methods is accurately capturing the variation in garment appearance while ensuring the clothing deforms naturally in response to the body's movements. BCNet [62] utilizes predefined templates and proposes a blend of body and cloth deformation to handle tight-fitting clothing and ensure smooth transitions between the two, by just modifying the pose of the avatar and the non-rigid deformation of garments. Works like [103], [195], and [40] instead proposed methods to generate clothes from 2d images based on deforming a template to match a set of features of the target image. DiffAvatar [88] instead first reconstruct a mesh of the target, and then use a physically driven simulation to estimate parameters of a template dress to match the previously extracted mesh. Despite their effectiveness, template-based methods do not generalize well as they are bound to a predefined set of templates. They often struggle with extremely loose or highly detailed garments, as their templates are typically designed for more standard clothing types.*

Sewing Pattern Methods *Another line of work aims to reconstruct garments by simulating sewing patterns—the 2D shapes that are typically stitched together to create clothing. These approaches often model the physical properties of fabric, such as stretch and fold, to generate realistic garments that can be "worn" by avatars. Works like [131] and [74] try to infer 2D sewing patterns from point clouds, while [22] proposed a PCA based embedding space to encode sewing patterns, and a neural network to reconstruct 3D clothes from it. Liu et al. [94] proposed SewFactory, a dataset and method for creating training data based on the previous method [73], and then formulate their solution, called SewFormer, to tackle sewing pattern reconstruction from single images by predicting offsets for a base set of panel and edges templates. Finally in DressCode [50], sewing patterns are used in a 3D garment generation framework. Despite being a powerful tool for expressing clothes, these methods can be problematic due to the inherent complexity of sewing patterns, particularly when dealing with garments that have intricate designs, irregular shapes, or non-standard fabrics. Moreover, when clothing is warped due to body movement, extracting accurate sewing patterns can become problematic, as the fabric may appear*

distorted—stretched or longer on one side—resulting in patterns that don't reflect the garment's true shape in a neutral pose.

Neural Methods *Neural methods have emerged as powerful approaches for clothing reconstruction by leveraging neural network techniques to extract garments from various inputs like images, meshes, and implicit representations. These approaches often utilize advanced representations such as Neural Fields, Signed Distance Functions (SDFs), and Unsigned Distance Functions (UDFs) to model the shape and behavior of clothing. Some methods rely on neural networks to estimate clothing geometry from image inputs, while others reconstruct garments by conditioning on mesh or body models. In [84], a neural network estimates garment shapes and deformations based on deformation prior knowledge. ClothCap [128] instead tracks and optimizes the garment surfaces from meshes, decoupling them from the body. In DrapNet [32], a fully differentiable garment generative network and garment draping network are used to both generate and reconstruct clothes. Recently, given the high variety that garments can offer, methods that use implicit or hybrid representations for clothes demonstrated to have superior visual reconstruction power than mesh-based methods. In DrapNet [32], a UDF-based fully differentiable garment generative network and garment draping network are used to both generate and reconstruct clothes. LayGa instead [91] builds on top of Animatable Gaussians [90] by reconstructing clothes and body separately. To reconstruct garments from an image, [24] proposed a framework that combines a pretrained sparse view generator and a volumetric SDF representation-based network for 3D garment modeling. Another Interesting approach comes from [39] and his extension [38], where a template-based model is used to represent the body of the avatar, while NeRF is used to represent clothes and hairs. This representation helps separating body from clothings, and also exploits neural radiance fields, which are better suited to capturing the large variety in shape and appearance present in clothing. One of the main limitations of techniques that rely on representations different from the classical explicit ones (like meshes or point clouds) is animating loose-fitting clothes as simulation-based techniques are hard to apply to implicit representations. AniDress [21] recently proposed a method for loose clothes rigging based on LBS, but their method falls short when applied to complex dynamic animations.*

In recent years, significant research has focused on finding more efficient ways to represent clothing, primarily through implicit or hybrid approaches, as well as sewing pattern-based methods. This shift arises from the limitations of traditional mesh-based methods, which struggle to capture intricate garment details and dynamic deformations, especially with complex clothing types and motion. Implicit or hybrid representations offer greater flexibility and visual fidelity for realistic clothing reconstruction and retargeting, while sewing pattern-based approaches aim to provide physically accurate garment representations by modeling how clothes are constructed in the real world.

2.2 Virtual Try On

2.2.1 2D Virtual Try On

2D Virtual Try-On (VTON) involves transferring an input garment onto a target person, preserving their body structure and pose. Early attempts at image synthesis using GANs, such as [174], [78], [178], and [197], laid the groundwork but lacked

the precision required for accurate clothing replacement. For example, FashionGAN [197] could replace garments based on text descriptions, but not in a visually accurate manner.

The first method to formalize the 2D VTON task was VITON [49]. VITON introduced a pipeline where, given an image I of a clothed person and a target clothing item c, the objective was to generate a new image $\hat{I}$, transferring c onto the person while preserving their pose and body parts. The method used Thin Plate Spline (TPS) to warp the input clothes and blend them with the person.

Building on VITON's framework, subsequent works improved various aspects of this process, focusing on refining garment fusion, preserving person-specific features, and leveraging additional priors. Methods like [154], [175], and [34] proposed ways to better integrate the clothing with the person's body, while others ([114], [60], [55], [181]) focused on enhancing the quality and realism of the synthesized images.

To address the limitations of warping in 2D space, UVTON [76] introduced a method that aligns source and target clothing in UV space using DensePose [136], allowing for more precise garment placement. Other approaches, such as [107] and [188], explored combining 2D and 3D techniques to capture the advantages of both modalities. Despite these advances, most methods remained limited to specific clothing types, mainly focusing on upper-body garments, and struggled with generalization. A significant step forward was done by Dress Code [110], which introduced a large dataset that incorporates various complex garments (tops, bottoms, and full-body clothing). The proposed method also improved garment warping by using pixel-level feature extraction to guide the process, increasing the overall accuracy and variety of clothing options.

More recently, diffusion-based approaches like [185], [196], [67], and [148] have demonstrated impressive results in 2D VTON. These methods leverage the generalization capabilities of pre-trained diffusion models, enabling better adaptability to diverse inputs. For instance, methods like StableVITON [67] and OutfitAnyone [148] independently process the person and garment images, then merge the garment features into the diffusion loop of the target person to generate the final output. However, despite their visual quality, diffusion-based methods face challenges like input misalignment, where the model incorrectly interprets input images, resulting in garment leaking or body shape distortions. For example, applying a typically gender-specific garment (like a dress) on a male body might result in unintended alterations to body features, such as the addition of breasts. Errors in segmentation maps further complicate this issue and pose estimations during preprocessing, which can lead to artifacts and misrepresentations of the garment. Additionally, these methods do not explicitly account for cloth physics, relying instead on the diffusion model's prior knowledge, which can sometimes lead to unrealistic garment behavior.

2.2.2 3D Virtual Try On

Dressing 3D avatars has been a fundamental problem in Computer Graphics and Vision for decades [102], often referred to as 3D Virtual Try-On (VTON). The task involves applying an input garment to a virtual human model referred to as retargeting. While this sounds straightforward, the challenge lies in how the garments and the body are represented and how they are aligned and fit together realistically. The various approaches to solving this problem differ primarily in the way they represent both garments and avatars, as well as the nature of the inputs and outputs involved.

One common approach is to use a body model to represent the physiognomy of the target and the clothing jointly. These methods typically model clothes as offsets

from an underlying body mesh, such as the SMPL model, to capture the shape and movement of the clothes in relation to the body. However, this offset-based approach has inherent limitations, as transferring the offsets to different body shapes often leads to artifacts and unrealistic clothing behavior. For example, Multi-Garment Net (MGN) [15] learns to predict both the body shape and clothing using a dataset of 712 digital garments layered on top of SMPL. However, MGN is constrained by its reliance on SMPL offsets, which limits its applicability to tight-fitting clothing and fails to generalize well to loose or multi-layered garments. In Pix2Surf [108] the proposed approach consists of transferring clothes images onto the SMPL model by learning dense correspondences between 2D garment silhouettes and UV maps of 3D garment surfaces.

In contrast, image-to-mesh methods attempt to reconstruct a 3D mesh of the garment from an input image. These methods take an image of the garment and a target body model as inputs and then generate a 3D mesh where the garment is applied to the body. For instance, M3D-VTON [189] uses an image of a garment and an image of a target person to produce a 3D mesh of the person wearing the garment. However, like other works in this category ([179]), it struggles to generalize across different clothing types due to the complexity involved in reconstructing volumetric garments from 2D images.

Another approach relies on sewing patterns, where garments are modeled as 2D panels that are stitched together to form a 3D mesh. These methods 2.1.3 aim to predict a garment's 2D sewing pattern and warp it onto a 3D body model. While this approach offers a precise way to represent garment geometry, it faces challenges when trying to generalize to complex, multi-layered, or occluded garments. The dimensionality difference between 2D patterns and 3D volumes introduces difficulties in capturing realistic deformations. Works like those presented in [199], [94], [], [72] and [75] utilize datasets that simulate physics to model 2D patterns, but these methods often falter when applied to real-world clothing scenarios ([9], [94]), where the diversity and complexity of garments exceed the capabilities of the models.

More recent methods explore the use of neural fields and volumetric representations such as point clouds, which allow for a more flexible and accurate representation of complex garments (like some of the methods shown in 2.1.3 and 2.1.3). These methods can represent clothing as continuous volumes, which makes them particularly suited for capturing garments from images and videos. Neural fields offer the potential to overcome some of the limitations of mesh-based and sewing-pattern approaches by representing garments as implicit functions that can easily adapt to changes in the body shape. SCARF and DELTA [39], [38], for example, use a hybrid approach where the body is represented as a mesh, and the clothes are encoded using neural fields. This method allows for easier retargeting of garments between different avatars, although it works best when the body shapes of the source and target avatars are similar, limiting its generalization.

Gaussian Splatting has recently been explored as a promising representation for modeling clothes in 3D avatar reconstruction. While many methods using Gaussian splatting focus on video-to-avatar reconstruction, they often fall short in handling the complexities of retargeting due to the lack of structured surfaces for accurate garment tracking. To overcome these limitations, Layered Gaussian Avatars (LayGA) introduces a new representation that formulates body and clothing as separate layers for photorealistic animatable clothing transfer from multi-view videos. LayGA builds on the Gaussian map's ability to capture detailed garments but enhances it by proposing a two-stage training process to improve garment tracking and handle collisions between body and clothing. However, LayGA is limited to upper-body garments, reconstructs the underlying body in an unrealistic manner, and fails to

ensure accurate retargeting, as the transferred clothing does not conform well to the body shape of the target avatar.

A recent method named GALA [69] works instead on meshes, and attempts to disentangle clothing layers from an input avatar mesh. This approach relies on high-quality mesh inputs, which are expensive to obtain, and it reconstructs both underlying body and clothing using Score Distillation Sampling loss (SDS) [129]. As a consequence, its result suffers from blurry artifacts and unrealistic body shapes due to the intrinsic and widely known limitations of the SDS loss.

Despite these advances, significant challenges remain in 3D VTON. One major issue is cloth-to-body alignment, especially in mesh-based methods where transferring garment offsets to different body models often results in unrealistic draping. Dimensionality gaps also pose a problem for sewing-pattern methods, as translating between 2D pattern representations and 3D avatar volumes is difficult. Additionally, the quality of input data is a significant bottleneck. High-resolution garment digitization requires expensive equipment and setup, and while datasets like DeepFashion3D [52] provide a collection of garment point clouds, they often lack the resolution and texture needed for high-quality results. Labeling clothes from scans is also a hard task, and there are only a few datasets, such as CloSe [5], 4d-Dress [161] and Sizer [152], to have them as other datasets either use physically simulated data ([111], [12], [16]), or do not have garment segmentation masks ([101], [63], [100], [2], [182]). Even state-of-the-art approaches struggle with generalization, as most methods are tailored to specific clothing types (e.g., tight-fitting garments) and fail to perform well with loose-fitting or multi-layered garments. Methods like MGN work well with tight clothes but lack the ability to handle a wide range of garment styles.

Furthermore, physically accurate garment behavior remains a challenge for most methods, especially when trying to simulate realistic interactions between clothing and the body during motion. Volumetric methods and neural fields offer promising ways to represent garment geometry, but it is hard to apply geometrical transformations, and they often rely on pre-existing knowledge about cloth behavior, leading to potential artifacts, unrealistic deformations, and incorrect garment layering.

In conclusion, while significant progress has been made in the field of 3D VTON, current approaches are limited by their reliance on specific representations, data quality constraints, and the complexity of simulating realistic cloth-body interactions. Future research must address these limitations to improve generalization, accuracy, and scalability, particularly in real-world applications where the diversity of clothing and body shapes presents a significant challenge.

Chapter 3

Background

3.1 Differentiable Representation Techniques

When dealing with 3D data, a suitable representation of the scene is required. Point clouds and meshes are the most common representations used in graphics as they are well-suited for GPU-based rasterization. These representations are classified as explicit methods as they are characterized by vector-valued parametrization functions $f : \Omega \to \mathcal{S}$, that maps a 2D domain $\Omega \subset \mathbb{R}$ to the surface $\mathcal{S} = f(\Omega) \subset \mathbb{R}^3$, and, which usually represent a piecewise approximation. In contrast, implicit representations, such as Unsigned Distance Functions and Signed Distance Functions, are defined by a function $F : \mathbb{R}^3 \to \mathbb{R}$ that classifies each point in the space to lie either inside, outside, or on the surface of the object[1].

3.1.1 NeRF

One of the latest advancements in the field of representation learning are Neural Fields ([120], [26], [106]) which are a subset of implicit representations that use Neural Networks to express the volume occupancy. Among these new representations, Neural Radiance Fields (NeRF [106]) recently became more popular as they enabled the representation of a scene as a combination of color and density instead of just surface occupancy. In NERF, the 3D scene is represented as a continuous 5D function implemented as an MLP that takes as input a 3D position (x, y, z) and two angles (θ and ϕ) for a total of 5 input parameters, and outputs the radiance, i.e., the emitted color $c = (r, g, b)$ in (x, y, z), and a density σ, i.e., the probability that a ray hits a particle in (x, y, z), of the scene for the direction (θ, ϕ) at the point (x, y, z). The process of rendering an image from a NeRF consists of marching camera rays through the scene to generate a set of points, evaluating those points using the network, and finally using classical volumetric rendering to accumulate colors and densities in an image. Given a Neural Radiance Field, and a ray $r(t) = o + rd$, where t is a value between a near and a far bound t_n and t_f, the expected color can be calculated as:

$$C(r) = \int_{t_n}^{t_f} T(t)\sigma(r(t))c(r(t), d)dt, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(r(s))ds\right) \quad (3.1)$$

where the function $T(t)$ denotes the accumulated transmittance along the ray from t_n up to t, i.e., the probability that the ray travels from t_n to t without hitting any other particle, and s is the integration variable bounded by the interval $[t_n, t]$. Rendering a

[1] M. Botsch et.al., Polygon Mesh Processing, 2010, pp. 1-20 [20]

view from a continuous neural radiance field requires estimating $C(r)$ for a camera ray traced through each pixel of the desired virtual camera. It is possible to discretize the color function C by sampling points across the input ray r, and summing up their color contribution to the pixel:

$$C(r) = \sum_{i=1}^{N} T_i(1 - exp(-\sigma_i \delta_i))c_i \tag{3.2}$$

where samples of density σ, transmittance T and color c are taken along the ray with interval δ_i. During this discretization process, NeRF exploits different techniques to better approximate the color function by dividing the ray into buckets and then sampling a point inside each one of them. This technique improves the final approximation of the original continuous function 3.1 via discrete operations, as having fixed intervals δ_i could worsen the approximation of the target continuous function. To further improve optimization, positional encoding, and a coarse network are implied to improve high-frequency details and sampling process. Finally, NeRF is trained using the total squared error between renders and true pixel colors for both coarse and fine renderings. The current state of the art builds on the original NeRF paper by implying voxels [[95], [147]], hash grids ([113]), and points ([171]), or by modifying parts of the model [[10], [11], [41], [25], [151], [135]]. These methods' main focus is faster optimization and better convergence, but the required computational cost is still high, and some of them suffer from noise. The current state of the art is represented by Mip-NeRF [11], whose main contributions are the integrated positional encoder, which consists of a positional encoder function that encodes a Gaussian region of space around a point, rather than an infinitesimal point, allowing the network to reason about sampling and aliasing, and an algorithm that casts cones instead of rays inside the scene. Despite its outstanding rendering capability, Mip-NeRF's training remains too high for real-time applications.

3.1.2 Gaussian Splatting

To merge real-time rendering and optimization, Kerbl et al. introduced Gaussian Splatting [66]. This method is based on 3D Gaussians, which are represented by a position $p = (x, y, z)$, a scaling vector $s = (s_x, s_y, s_z)$, a rotation expressed as a quaternion $q = (q_r, q_i, q_j, q_k)$, where q_r is the real part, and q_i, q_j, q_k are the imaginary parts, an opacity value α, and Spherical Harmonics to represent the color of the Gaussians.

Optimizable Gaussians

The process of rendering a 3D Gaussian consists of projecting it to the 2D image space. This can be done by calculating the formula provided in [201] by Zwicker et al., where, given a view transformation, the covariance matrix Σ' in camera coordinates is calculated as:

$$\Sigma' = JW\Sigma W^T J^T \tag{3.3}$$

where J is the Jacobian of the affine approximation of the projective transformation. However, a covariance matrix is meaningful only when it is positive semi-definite, so directly optimizing it is not a good practice. Instead, a covariance matrix can be calculated from the scale S and rotation matrices R as:

$$\Sigma = RSS^T R^T \tag{3.4}$$

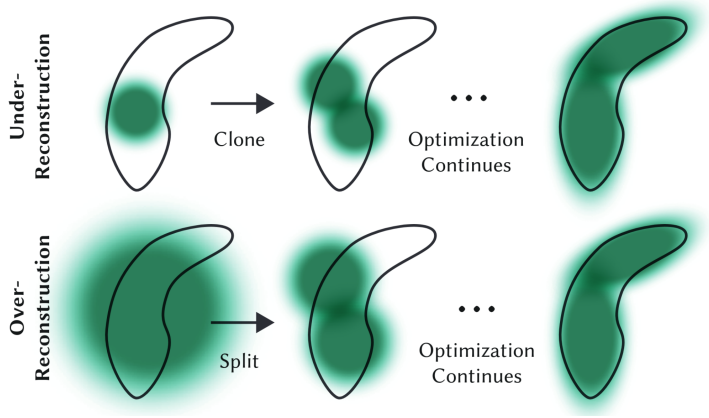

Figure 3.1. Adaptive densification scheme from Gaussian Splatting [66](Fig. 4). Top row (under-reconstruction): When small-scale geometry (black outline) is insufficiently covered, the respective Gaussian are cloned. Bottom row (over-reconstruction): If small-scale geometry is represented by one large splat, it is split in two smaller ones.

Both the scaling and rotation matrices can be trivially calculated from s and q as:

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}) \tag{3.5}$$

$$R = \begin{bmatrix} 1-2(q_j^2+q_k^2) & 2(q_iq_j-q_kq_r) & 2(q_iq_k+q_jq_r) \\ 2(q_iq_j+q_kq_r) & 1-2(q_i^2+q_k^2) & 2(q_jq_k-q_iq_r) \\ 2(q_iq_k-q_jq_r) & 2(q_jq_k+q_iq_r) & 1-2(q_i^2+q_j^2) \end{bmatrix} \tag{3.6}$$

These equations give us the basic differentiable mathematical tools for splatting a single 3D Gaussian onto our image plane. Now, we need a differentiable algorithm to manage the rasterization process of multiple Gaussians and thus create an image.

Rendering Process

The image formation model for Gaussian Splatting is the same as the one used by NeRF and reported in Formula 3.2, while the rasterization algorithm differs. At the start of the rendering process, the image is divided into 16×16 *pixels, and Gaussians are assigned to a tile. This assignment process consists of checking for every Gaussian in which tiles the* 99% *confidence interval falls. Then Gaussians are ordered based on their viewspace depth, and their tile ID via Radix sort. Finally, the image formation formula is used to accumulate color per pixel via Gaussian sampling.*

Optimization

For optimization, Stochastic Gradient Descent (SGD) is implied with the following loss:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SSIM} \tag{3.7}$$

where $\mathcal{L}_1$ is the L1 loss and $\mathcal{L}_{D-SSIM}$ is the DSSIM term proposed by [8]. During optimization, some Gaussians may aggregate and form over-represented areas or may do the opposite, creating gaps, or leading to an under-populated area. To overcome

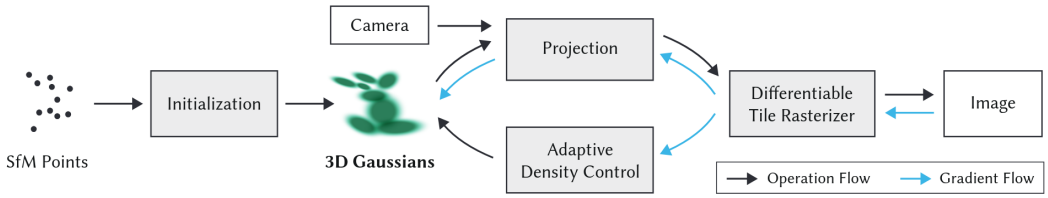

Figure 3.2. Full pipeline from Gaussian Splatting [66](Fig. 2). 3D Gaussians are initialized by a sparse Structure from Motion (SfM) pointcloud. During the forward pass, Gaussians are projected on the camera plane and rasterized to form an image. The backward pass brings then back the gradient to the 3D Gaussians, and is also used during the Adaptive Density Control to choose whether to split, clone, or leave Gaussians.

these problems, adaptive control of Gaussians (Figure 3.1) is implemented to split them into over-represented areas, and clone when they are needed. The choice of cloning or splitting Gaussians is done based on the positional gradients accumulated as a large value indicates that the model is trying to move those Gaussians to fix the area. Since during training also an opacity α is learned, Gaussians whose value is under a certain threshold are removed to avoid keeping them inside the representation. The full Gaussian Splatting learning process is shown in Figure 3.2.

3.2 SMPL: Skinned Multi-Person Linear Model

Many models have been proposed during the course of Computer Science history, with the sole scope of representing the human body ([3], [23], [127], [200], [170], [119], [4], [98]). Despite the existence of high-detailed body models such as STAR [119] and GHUML [170], during the last few years, the SMPL [98] body model has gained so much attention and popularity because of its simplicity, low number of parameters, differentiability and ease to be adapted to rendering pipelines, that is now one of the most used by the research community for expressing metahumans. The goal of SMPL is to have realistic poseable models that cover the space of human shape variation and consider how the body shape changes with pose, by exploiting a vertex displacement approach. Many works have then extended the SMPL model to enhance its expressive power by adding parameters to model hands ([137], [138]), face ([86], [122]), and was used as a base to recover body shape from images ([17], [58], [65], [123], [118], [184], [71], [64], [150]), clothed humans ([168], [169], [37], [2], [87], [15], [198], [92], [128], [56], [90], [132], [190]). By modeling the human body with high accuracy and flexibility, SMPL has become a critical building block in a wide range of computer vision and graphics applications. Its ability to account for both body shape variations and pose-related deformations has made it a go-to solution for researchers aiming to create realistic digital humans. As a result, SMPL has not only contributed to advances in metahuman representation but has also enabled new directions in fields such as virtual try-on, motion capture, and human-computer interaction. In the following subsections, we will focus deeper into the technical structure of SMPL, explaining its key components.

3.2.1 SMPL Body Model

The core idea of the SMPL model is that people can be expressed as a combination of pose and identity (or shape) and that it is possible to learn a function $M(\vec{\beta}, \vec{\theta})$

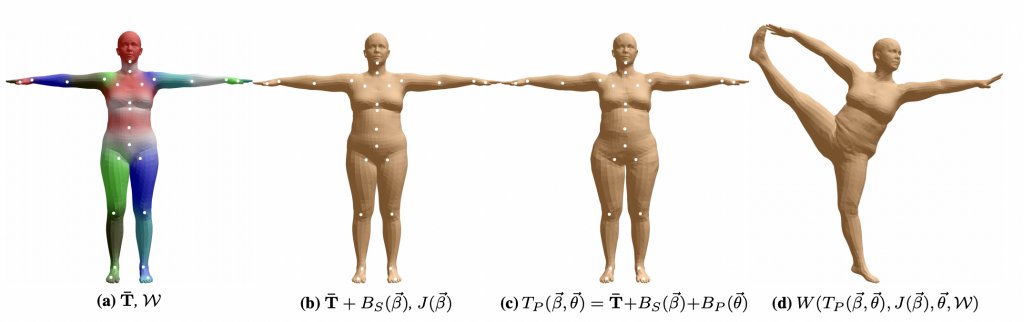

Figure 3.3. SMPL model image from the original paper [98]. (a) Template mesh with blend weights indicated by color and joints shown in white. (b) With identity-driven blendshape contribution only; vertex and joint locations are linear in shape vector $\overrightarrow{\beta}$. (c) With the addition of pose blend shapes in preparation for the split pose; note the expansion of the hips. (d) Deformed vertices reposed by dual quaternion skinning for the split pose.

3.12 that takes a shape $\overrightarrow{\beta}$ and a pose $\overrightarrow{\theta}$ parameters as input and applies a series of transformations to a T-posed template model $\bar{\boldsymbol{T}}$ to generate a posed, realistic human 3.3. The next sections will cover the SMPL body model by first defining how the template model, the joint parameters, and how the latter are applied to the former via Linear Blend Skinning 3.2.1, and then following with a description of all the key components of the SMPL model (Shape Blend Shape 3.2.1, Pose Blend Shape 3.2.1, Joint Locations 3.2.1 and the final SMPL Model Mormulation 3.2.1

Template and Joints Parametrization

The template avatar $\bar{\boldsymbol{T}} \in \mathcal{R}^{N\times 3}$ used by SMPL consists of a mean shape model with $N = 6890$ vertices in zero pose $\overrightarrow{\theta}^*$. The pose of the body is defined by a standard skeletal rig, composed of $K = 23$ joints plus a global orientation. This defines the pose parameter $\overrightarrow{\theta} = [\omega_1^T, ..., \omega_k^T]^T$, and a number of parameters $|\overrightarrow{\theta}| = 3 \times 23 + 3 = 72$. The rotation angle for each joint is transformed to a rotation matrix through the rodriguez formula:

$$exp(\overrightarrow{\omega}_j) = I + \hat{\bar{\omega}}_j sin(||\overrightarrow{\omega}_j||) + \hat{\bar{\omega}}_j^2 cos(||\overrightarrow{w}_j||) \tag{3.8}$$

where I is a $3x3$ identity matrix, $\bar{\omega}_j$ is the unit norm axis of rotation for the j-th angles and $\hat{\bar{\omega}}_j$ is the skew symmetric matrix of the 3-vector $\bar{\omega}$. Given a skinning weight matrix $\mathcal{W} \in \mathcal{R}^{N\times K}$ that determines how much each vertex of the template body is influenced by the bending of each joint, and the 3D position of the joints $\boldsymbol{J}$, we can now define a linear blending skinning function $W(\bar{\boldsymbol{T}}, \boldsymbol{J}, \overrightarrow{\theta}, \mathcal{W})$ that can be applied to a vertex $\bar{t}_i$ of the canonical template, to get the posed vertex $\bar{t}'_i$:

$$\bar{t}'_i = \sum_{k=1}^{K} w_{k,i} G'_k(\overrightarrow{\theta}, \boldsymbol{J}) \bar{t}_i \tag{3.9}$$

$$G'_k(\overrightarrow{\theta}, \boldsymbol{J}) = G_k(\overrightarrow{\theta}, \boldsymbol{J}) G_k(\overrightarrow{\theta}*, \boldsymbol{J})^{-1} \tag{3.10}$$

$$G_k(\overrightarrow{\theta}, \boldsymbol{J}) = \prod_{j\in A(k)} \begin{bmatrix} exp(\hat{\omega}_j) & \boldsymbol{j}_i \\ \overrightarrow{0} & 1 \end{bmatrix} \tag{3.11}$$

where $w_{k,i}$ is an element of the blending weight matrix $\mathcal{W}$, representing how much the rotation of part k effects the vertex i, $exp(\vec{\theta}_j)$ is the local 3×3 rotation matrix corresponding to the j-th join, $G_k(\vec{\theta}, \boldsymbol{J})$ is the world transformation of joint k, and $G'_k(\vec{\theta}, \boldsymbol{J})$ is the trasformation after removing the rest pose transformation parametrized by $\vec{\theta}*$. $A(x)$ denotes the ordered set of joint ancestors of joint k. To increase the expressive power of the model, and to avoid modifying the skinning function presented in 3.9, SMPL modify the template $\bar{\boldsymbol{T}}$ in an additive way. The full model is then expressed as:

$$M(\vec{\beta}, \vec{\theta}) = W(T_P(\vec{\beta}, \vec{\theta}), J(\vec{\beta}), \vec{\theta}, \mathcal{W}) \tag{3.12}$$

$$T_P(\vec{\beta}, \vec{\theta}) = \bar{\boldsymbol{T}} + B_S(\vec{\beta}) + B_P(\vec{\theta}) \tag{3.13}$$

where $B_S(\vec{\beta})$ and $B_P(\vec{\theta})$ are vectors of vertices representing offsets from the template. Authors from the SMPL paper refer to these vectors as shape and pose blend shapes respectively.

Shape Blend Shapes

The shape blend shapes describe how different body shapes can be represented by a set of linear coefficients that define various body morphologies. This relationship is expressed as:

$$B_S(\vec{\beta}; \mathcal{S}) = \sum_{n=1}^{|\vec{\beta}|} \beta_n \mathbf{S}_n \tag{3.14}$$

where $\vec{\beta} = [\beta_1, \ldots, \beta_{|\vec{\beta}|}]^T$ is the vector of shape coefficients, $|\vec{\beta}|$ is the number of shape coefficients, $\mathbf{S}_n \in \mathbb{R}^{3N}$ represents the orthonormal principal components of shape displacements, and $\mathcal{S} = [\mathbf{S}_1, \ldots, \mathbf{S}_{|\vec{\beta}|}] \in \mathbb{R}^{3N \times |\vec{\beta}|}$ is the matrix of all shape displacements.. This can be written in matrix form as:

$$B_S(\vec{\beta}; \mathcal{S}) = \mathcal{S}\vec{\beta} \tag{3.15}$$

The matrix $\mathcal{S}$ is learned from registered training meshes, as explained in Section 4 of the SMPL paper [98].

Pose Blend Shapes

The pose blend shapes model the deformations of the body due to changes in pose. SMPL authors define a function $R : \mathbb{R}^{|\vec{\theta}|} \to \mathbb{R}^{9K}$, which maps a pose vector $\vec{\theta}$ to a vector of concatenated relative part rotation matrices. Since the model rig has 23 joints, the length of the vector $R(\vec{\theta})$ is $23 \times 9 = 207$. Each element of $R(\vec{\theta})$ is a function of the sines and cosines of the joint angles. SMPL authors ensure that the effect of the pose blend shapes is linear in: $R^*(\vec{\theta}) = R(\vec{\theta}) - R(\vec{\theta}^*)$, where $\vec{\theta}^*$ represents the rest pose. The vertex deviations from the rest template are given by:

$$B_P(\vec{\theta}; \mathcal{P}) = \sum_{n=1}^{9K} (R_n(\vec{\theta}) - R_n(\vec{\theta}^*))\mathbf{P}_n \tag{3.16}$$

where $\mathbf{P}_n \in \mathbb{R}^{3N}$ *are vectors representing vertex displacements. In matrix form, this can be expressed as:*

$$B_P(\vec{\theta};\mathcal{P}) = \mathcal{P} \cdot (R(\vec{\theta}) - R(\vec{\theta}^*)) \tag{3.17}$$

where $\mathcal{P} = [\mathbf{P}_1, \ldots, \mathbf{P}_{9K}] \in \mathbb{R}^{3N \times 9K}$ *is the matrix of all 207 pose blend shapes, which are learned during training.*

Joint Locations

The joint locations depend on the body shape, as different shapes result in different joint positions. Each joint is represented by its 3D location in the rest pose. To model this, SMPL authors define the joints as a function of the body shape $\vec{\beta}$*:*

$$J(\vec{\beta};\mathcal{J},\bar{\boldsymbol{T}},\mathcal{S}) = \mathcal{J}(\bar{\boldsymbol{T}} + B_S(\vec{\beta};\mathcal{S})) \tag{3.18}$$

where $\mathcal{J}$ *is a regression matrix that transforms the rest vertices into rest joint locations.*

The matrix $\mathcal{J}$ *is learned from examples of different people in various poses and determines which mesh vertices are important for estimating the joint locations and how to combine them.*

SMPL Model Formulation

We can finally define the SMPL model $M(\vec{\beta}, \vec{\theta};\phi)$*, based on the parameters* $\phi = \left\{\bar{\boldsymbol{T}}, \mathcal{W}, \mathcal{S}, \mathcal{J}, \mathcal{P}\right\}$ *that were defined in the previous subsections:*

$$W(T_P(\vec{\beta}, \vec{\theta};\bar{\boldsymbol{T}},\mathcal{S},\mathcal{P}), \mathcal{J}(\vec{\beta},\mathcal{J},\bar{\boldsymbol{T}},\mathcal{S}), \vec{\theta}, \mathcal{W}) \tag{3.19}$$

and hence the transformation that brings a vertex of the template $\bar{\boldsymbol{T}}$ *from canonical space to posed space is defined as:*

$$t'_i = \sum_{k=1}^{K} w_{k,i} G'(\vec{\theta}, J(\vec{\beta};\mathcal{J},\bar{\boldsymbol{T}},\mathcal{S}))\boldsymbol{t}_{P,i}(\vec{\beta}, \vec{\theta};\bar{\boldsymbol{T}},\mathcal{S},\mathcal{P}) \tag{3.20}$$

where $t_{P,i}(\vec{\beta}, \vec{\theta};\bar{\boldsymbol{T}},\mathcal{S},\mathcal{P})$ *is the function that applies the blend shapes to the vertex* i *and is defined as:*

$$t_{P,i}(\vec{\beta}, \vec{\theta};\bar{\boldsymbol{T}},\mathcal{S},\mathcal{P}) = \bar{\boldsymbol{t}}_i + \sum_{m=1}^{|\vec{\beta}} \beta_m s_{m,i} + \sum_{n=1}^{9K} (R_n(\vec{\theta}) - R_n(\vec{\theta}^*))\mathbf{p}_n \tag{3.21}$$

in which $s_{m,i}, \boldsymbol{p}_{n,i} \in \mathbb{R}^3$ *are the elements of the shape and pose blend shapes corresponding to template vertex* $\bar{\boldsymbol{t}}_i$*. With the final SMPL body model, it is possible to sample from the pose and shape space to synthesize different human models as shown in Figure 3.4 and exploits its differentiability and simplicity in downstream tasks like those discussed in 2 and 3.3.2.*

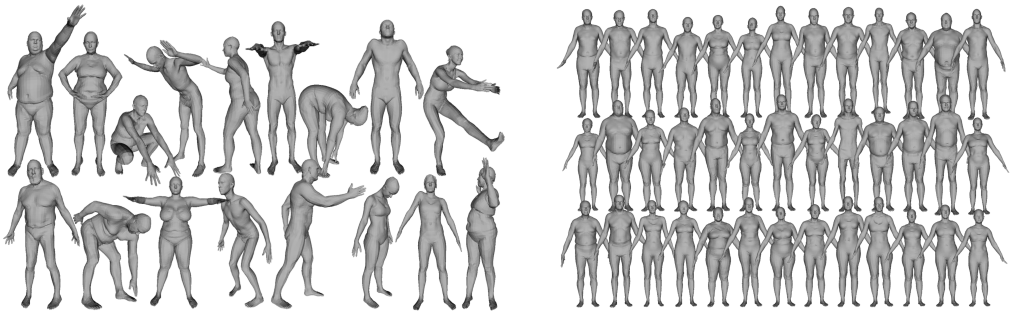

Figure 3.4. Set of pose+shape (left) and shape (right) samples from [98]

3.3 3D Body Registration

One of the oldest and studied crucial problems of Computer Vision is shape registration, whose main goal is to recover the correspondence between two shapes, and is implied in many sub-tasks such as shape analysis, pattern discovery, and statistical model training. 3D body registration is a subset of shape registration that focuses on human body retrieval and is crucial in many tasks such as virtual try-on, animation, and mixed reality.

3.3.1 Shape Registration

Problem Formulation

In the simplest scenario for shape registration, we assume that both the input and target datasets contain the same number of points N. Under this assumption, the registration problem can be formulated as minimizing the following energy function:

$$E = \sum_{(i,j)\in C} ||x_i - y_j||^2 \tag{3.22}$$

here $C = \{(x_i, y_i)\}^N$ is a set of pairs that denote the correspondences between points of the input and target point clouds. If we further assume C to be known, a naive approach could consist of moving each point of the input point cloud towards its corresponding target point. Yet, this approach is prone to shape collapses and unwanted artifacts, and regularization is needed to run a correct rigid registration.

Procrusters analysis *A closed-form solution for the rigid shape registration problem is given by the Procrusters analysis, whose solution is based on finding an optimal scale s, rotation R, and translation t by optimizing:*

$$E = \sum_i ||f(x_i) - y_i||^2 \tag{3.23}$$

where

$$f(x_i) = sRx_i + t \tag{3.24}$$

The solution for this problem is based on Singular Value Decomposition (SVD), a technique that decomposes a matrix $M \in \mathbb{R}^{n\times m}$ into a multiplication of three matrices $M = U\Sigma V^T$, with $U \in \mathbb{R}^{n\times n}$ an orthogonal matrix made by vectors represent directions in the row space of M (also known as left singular vectors), $\Sigma \in \mathbb{R}^{n\times m}$

diagonal matrix with non-negative real numbers (known as singular values), and $V^T \in \mathbb{R}^{m\times m}$ is another orthogonal matrix containing vectors represent directions in the column space of M (also known as right singular vectors). Accordingly to the geometric interpretation of SVD, U and V^T can be seen as two rotations, while Σ is a diagonal matrix that represents a scaling.

Procruster analysis finds the optimal rigid trandformation between two sets of points x and y by:

- *Performing SVD decomposition on the cross-covariance $x^T y = U\Sigma V^T$, and deriving the optimal rotation $R = UV^T$;*
- *Finding the optimal translation as a difference of centroids $t = sR\bar{x} - \bar{y}$;*
- *Calculating the optimal scale as a quotient of eigenvalue sums $s = \frac{tr(x^T y)}{||x||^2}$*

However, in real-world applications, the correspondences between points are usually unknown, and the cardinality of the two sets, x and y, may differ. In such cases, more advanced techniques, such as the Iterative Closest Point (ICP) algorithm, are required to iteratively find both the correspondences and the optimal transformation.

Iterative Closest Point *Given a set of input points $X = [x_1, x_2, \ldots, x_n]$ and a set of target points $Y = [y_1, y_2, \ldots, y_n]$, target function for a general rigid shape registration problem is:*

$$E = \sum_i ||f(x_i) - y_i||^2 \tag{3.25}$$

$$f(x_i) = sRx_i + t \tag{3.26}$$

where s is a scale, R is a rotation matrix and t is a translation. To minimize this target function, we need to assign each input point x_i to a target point y_i to minimize the error. We have to optimize over f and the set of correspondences $C = \{(x_i, y_i)\}_i^N$:

$$E(C, f) = \sum_i \min_{x\in X} ||f(x) - y_i||^2 \tag{3.27}$$

Solving this equation is not trivial, and directly optimizing for 3.27 is not an efficient approach as it would require optimizing a permutation of the elements of X, whose cost is exponential with respect to the number of input and target points N, making this naive approach unfeasible for a large set of points.

ICP *One of the most famous and widely used algorithms for rigid shape registration is ICP [13], which proposes an iterative optimization approach divided into two steps. The first one consists of fixing f and optimizing for correspondences:*

$$x_i^{t+1} = \operatorname*{argmin}_{x\in X} ||f^j(x) - y_i||^2 \tag{3.28}$$

This step associates each element of the input set X to an element of the target set Y, giving us a sub-optimal pairing C. These pairings are then used during the second step, where we optimize for f:

$$f^{j+1} = \operatorname*{argmin}_f \sum_i ||f(x_i^{j+1}) - y_i||^2 = \operatorname*{argmin}_{r,S,t} \sum_i ||sRx_i^{j+1} + t) - y_i||^2 \tag{3.29}$$

this second step involves optimizing for a scale s, a rotation R and a translation t, which can be don via Procrusters algorithm. By initializing $R = I$, $s = 1$, $t = \frac{\sum y_i}{i} - \frac{\sum x_i}{i}$, and repeating steps 3.28-3.29 until convergence, the algorithm will reach a local minimum.

Limitations *ICP remains a simple approach with many limitations as it is sensible to outliers, does not consider surfaces, is limited to rigid transformations, and converges to a local minimum. Many works have been proposed to address the limitations of the ICP algorithm by addressing noise and sparsity ([145], [27]) and by adding deep learning approaches during the correspondence estimation ([180], [99], [162]). To address non-rigid deformations, methods based on probabilistic formulations have been proposed such as the Coherent Point Drift algorithm [112], and his followups ([53], [54]). Subsequently, deep learning based approaches tried to learn the deformation from data ([80], [156]), or to predict offsets based on reduced representations ([149]).*

3.3.2 3D Parametric Body Registration

Given its parametric formulation, the SMPL model has been widely used in literature to fit human scans. [43] used an autoencoder network to predict the correspondences between a template shape and an input human shape. [155] proposed a spatial-temporal attention convolution to directly regress the vertex coordinates of a fitted mesh model. [159] used the occupancy function to represent the canonical space, and used a piecewise transformation field to deform the source point to the canonical shape. [18] proposed the FAUST dataset for 3D mesh registration, and [19] extended it by adding movement and proposing a method for 4D registration based on dense texture match computation between temporally subsequent registration and 3D model un-synchronized fitting. [44] introduced Shape Deformation Networks to jointly encode 3D shapes and correspondences. [105] proposed an approach based on estimating functional maps between a fixed template $\mathcal{M}$, and an observed data $\mathcal{N}$. Recently, works based on implicit functions and Neural Fields ([30], [104], [14], [160]) demonstrated impressive results for 3D body registration from point clouds. In Learned Verted Descent (LVD) [30], a neural field was trained to learn the direction towards which each sample of the target point cloud should be brought to match the input data (whether it is 2D or 3D). This idea was then improved in NSR [104], where a Neural Iterative Closes Point (NICP) was used during the template fitting, and a local version of LVD (called LoVD) was used to enhance the optimization quality.

NSR, NICP and LoVD

As anticipated in 3.3.2, NSR builds on top of Learned Vertex Descent (LVD). The key idea of LVD is to train a neural network $F^\theta = \mathbb{R}^3 \rightarrow \mathbb{R}^{m\times 3}$ tha, for any point in $\mathbb{R}^3$ outputs the offsets towards the m vertices of the target deformed template $\hat{\boldsymbol{X}}$. During inference, the network generates the Neural Field for the target, and every vertex v_i of the SMPL template shape is driven towards one of the $i-th$ points predicted by the network:

$$\hat{x}_i = x_i + F^\theta(x_i)_i \tag{3.30}$$

the inference step is run multiple times on the template until it converges toward the target point cloud.

To enhance the NF predictions, authors proposes Localized Vertex Descent (LoVD), a variation of LVD that, instead of having a single MLP that predicts the Neural Field for every SMPL vertex, divides the template into l regions via spectral clustering, and then learns a local MLP for each region. During inference, the NICP consists of iterating between correspondence and registration steps until convergence (resembling the ICP 3.3.1 approach), and after, a refinement stage is used to yield a better result and capture details.

Correspondence *This step consists of sampling points y_k over the target shape and querying the NF to pair points of the target data with points of the template by choosing the minimum predicted offset:*

$$\tilde{i_k} = \underset{i}{\operatorname{argmin}} ||F^{\tilde{\theta}}(\boldsymbol{y}_k)_i||_2^2 \tag{3.31}$$

The intuition is that points on the target shape should be close to the desired registered SMPL vertices locations. Hence, for every point of the target, the network is expected to predict at least one offset with a norm close to zero. Considering the smallest offset predicted by the network, the NICP correspondence step pairs the target points with the SMPL point, relying on the network data-prior.

Registration *The registration step consists of minimizing the sum of minimum offsets $F^{\theta}(\boldsymbol{y}_k)_{\tilde{i_k}}$ for all the query points $\boldsymbol{y_k}$ by backpropagating:*

$$\tilde{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{k=1}^{n} ||F^{\theta}(\boldsymbol{y}_k)_{\tilde{i_k}}||_2^2 \tag{3.32}$$

Refinement *After running the previous steps multiple times, the predicted template parameters are refined with a Chamfer distance optimization. Given two cloud points $A, B \in \mathbb{R}^d$, the Chamfer distance from A to B is a measure of similarity defined as:*

$$d_{CH}(A, B) = \sum_{a \in A} \min_{b \in B} d(a, b) + \sum_{b \in B} \min_{a \in A} d(a, b) \tag{3.33}$$

where $d(a, b)$ is a distance measure (usually Euclidean). Intuitively, the Chamfer distance quantifies how close each point in one set is to its nearest point in the other set and vice versa. Finally, if high resolution is required, SMPL+D is implied to register the fine details of the target point cloud.

Chapter 4

Method

In this work, we present a method aimed at retargeting clothes in 3D. Transferring garments between two different body shapes, even when posed similarly, presents significant challenges. Traditional approaches, such as applying offsets from a T-posed SMPL model to a target, have demonstrated their limitations, especially when dealing with loose or complex garments, as shown in related works 2.

Alternatively, one could build an approach based on finding correspondencies via K-Neaerst Neighbors. However, as shown with our baseline in 4.1, these method introduces numerous artifacts and fails to provide the necessary regularization, making it impractical for real-world applications. The baseline that we built consists of a transfer function that takes a source Gaussians Splatting P_S, a source SMPL registration M_S, a target 3D Gaussian point cloud P_T, and a target SMPL registration M_T, and retarget clothes based on establishing correspondencies between the body registration to directly transfer clothes. It starts by selecting garment points from the source point cloud, creating a subset denoted as P_S^g. For each point P_{Si}^g in this subset, the method assigns K neighbors from the source SMPL mesh M_S using the KNN algorithm. The nearest neighbors are expressed as

$$\mathcal{N}_i = KNN(P_{Si}^g; M_S)$$

Then, for each point P_{Si}^g, a weighted average of offsets from the SMPL points is calculated as

$$O_i = \sum_{j=0}^{k} w_J (P_{Si}^g - \mathcal{N}ij)$$

where $\mathcal{N}ij$ represents the j-th SMPL neighbor of P_{Si}^g, w_j is a normalization weight given by

$$w_j = \frac{d(P_{Si}^g, \mathcal{N}ij)}{\sum_k d(P_{Si}^g, \mathcal{N}_{ik})}$$

and $d(\cdot, \cdot)$ is the Euclidean distance. Since the correspondencies between M_S and M_T are known, the function transfers all the Gaussians in P_S^g on the corresponding vertices of M_T, resulting in a roughly posed target set of garment Gaussians P_T^g. At this stage, however, offsets are expressed with respect to the source avatar surface M_S, and Gaussians have not been rotated. We can express O with respect to the target avatar by calculating the relative rotations between surface points on M_S and M_T, and then by applying those rotations to the offsets O (and to the Gaussians P_T^g). Finally, the retargeting process ends by adding the offsets to the target clothes

$$P'^g_S = P_T^g + O$$

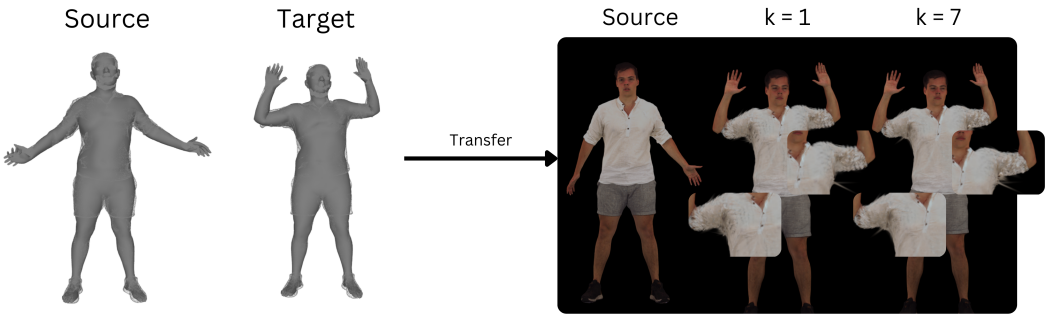

Figure 4.1. The figure shows the results of applying K-Nearest Neighbors to find a correspondence between a source and a target avatar. On the left side, the inputs for the baseline: source SMPL registration M_S, source set of Gaussian points P_S that represent the avatar (first grey mesh and set of points from the left), target SMPL registration M_T, and target set of Gaussian points that represent the avatar (second grey mesh and set of points from the left). On the right side, we show a gallery that comprehends the source avatar (first column), the results for $K = 1$ (second column) and $K = 7$ (third column).

The results on the right side of Figure 4.1 reveal several artifacts that arise from using the naive KNN-based approach for retargeting. In particular, the retargeted Gaussian points on the arms appear blurred, with a noticeable loss of detail such as the folds and sharp features of the shirt. This blurring is especially pronounced in areas where the body undergoes more significant rotations, such as the arms. Similarly, baselines built on free-form registration methods like Coherent Point Drift fall short due to the absence of strong constraints and regularization.

Our method is designed to address these issues by normalizing the source garments into a canonical pose and shape before fitting them to the target avatar. By operating in a canonical space, we ensure easier and more reliable transfer of garments, with pose and shape adjustments applied subsequently to animate the clothing on the target body.

We aim to develop a solution that is less computationally intensive than existing video-based methods, making it more suitable for real-time applications. To achieve this, our method avoids any strong dependency on an initial SMPL registrations, which can often be unreliable, while still supporting efficient linear blend skinning (LBS) and rendering.

There are two core objectives of our approach:

1. *to represent garments in a flexible, manipulable representation;*

2. *to facilitate garment transfer between avatars by working in a canonical pose and shape space, allowing for straightforward animation of the clothing once the pose and shape are reapplied.*

Given the limitations of representing clothes as offsets from a body model and the computational intensity of neural fields, we opted for Gaussian Splatting [66] as our representation method. This approach has demonstrated promising results in modeling garments ([92], [56], [90], [57]) and, due to its point cloud-based structure, allows for efficient transformations.

One key issue in prior works is their dependence on accurate SMPL registration for modeling avatars, typically involving the projection between canonical and posed spaces. By leveraging the inherent point-cloud nature of Gaussian Splatting, we can reconstruct the avatar-based solely on appearance data and subsequently address

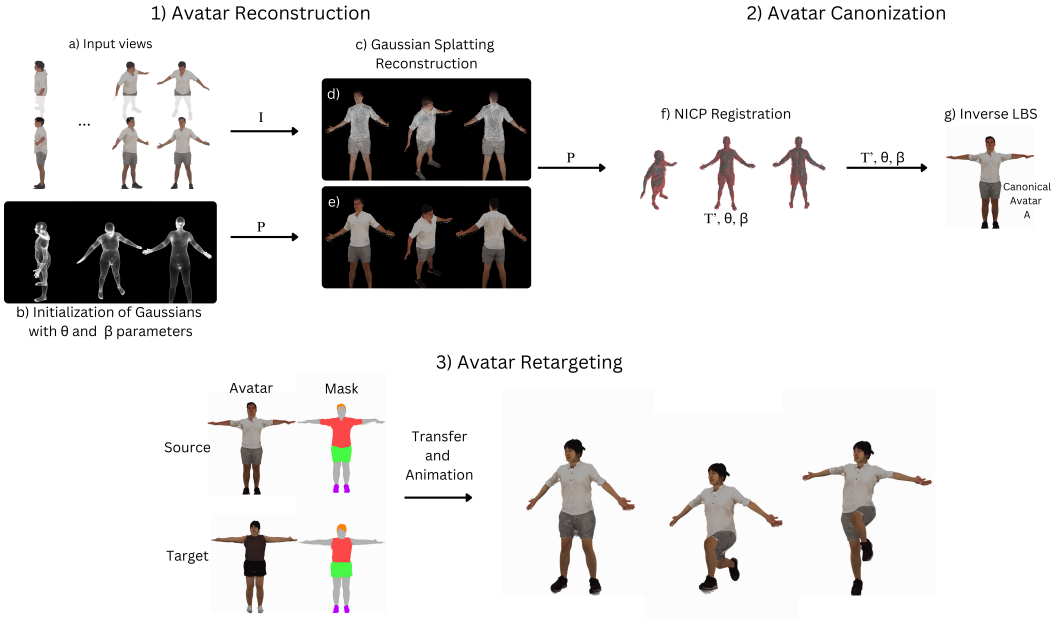

Figure 4.2. Representation of the three main steps of our pipeline: 1) Avatar Reconstruction, 2) Avatar Canonization, and 3) Avatar Retargeting.

the problem of retargeting by registering SMPL on the reconstructed point cloud, facilitating the canonical pose and shape normalization process.

4.1 Problem Formulation

Retargeting in 3D typically involves transferring garments from a source avatar X to a target one Y. This requires an appropriate representation of both the source and target avatars as input. Various prior works have tackled this problem using different representations and methodologies. For example, some approaches have utilized monocular images of the source garment combined with a 3D model for the target avatar ([188], [108]), while others have used 3D garment reconstructions alongside the SMPL parameters of the target avatar to warp garments around the target ([69], [93]). Additionally, certain reconstruction pipelines have been shown to enable clothing retargeting as a secondary outcome of the reconstruction process ([39], [38], [128]).

For our approach, we adopt a multiview image setup for both the source and target avatars. Specifically, we are given a set of N sparse-view images $I_X = \{I_X^1, I_X^2, \ldots, I_X^N\}$ of the source person, as well as a set of M sparse-view images $Y = \{I_Y^1, I_I^2, \ldots, I_Y^M\}$ of the target person. The goal is to produce an animatable avatar of the target person A_Y dressed in the clothes of the source person $A'_Y = f(X_G, A_Y)$, where f is the transfer function and X_G are source avatar's clothes. The choice of multiview images as input is motivated by their ability to capture more comprehensive geometric and appearance information than a single image while being computationally more feasible than a multiview video setting. Furthermore, with recent advances in multiview diffusion techniques, our method can be easily extended to handle single-image inputs, making it scalable and applicable to real-world scenarios.

Our pipeline is divided into three main stages:

1. ***Avatar Reconstruction*** *4.2: reconstructing a visual representation of source and target avatars from multiview images;*

2. ***Avatar Canonization*** *4.3: making the reconstructed avatars animatable, and bringing them in canonical space;*

3. ***Avatar Retargeting*** *4.4: swapping the clothing between the avatars in canonical space, and reapplying pose and shape transformations to generate the final dressed avatar.*

Figure (4.2) shows the proposed approach. In the subsequent sections, we provide a detailed description of our pipeline and discuss the experiments and design choices that led to the development of the final model. For clarity, we will first explain the initial two steps of the pipeline (reconstruction and canonicalization) using a general avatar, whose input will be denoted as $I = \{I^1, I^2, \ldots, I^N\}$*. In our actual implementation, these two steps are applied separately to both the source and target avatars.*

4.2 Avatar Reconstruction

4.2.1 Gaussian Representation

The first step of our pipeline involves reconstructing the subject using Gaussian Splatting (GS). This step is straightforward, as GS can easily reconstruct 3D scenes from multiview images I*. The process follows the differentiable rasterization technique described in Section 3.1, where images are rendered, compared with the training views* I*, and the reconstruction error is backpropagated. Adaptive control then adjusts the density of the Gaussians where necessary.*

However, following this process, two primary artifacts emerge:

- *Some Gaussians tend to disappear by learning a low opacity* δ*, causing them to fade out.*

- *Gaussians tend to stretch excessively, which can result in spikes during animations when clothes are moved and joints are bent.*

4.2.2 Modifications to GS

To address these issues, we modified Gaussian Splatting in two ways. First, we forced the opacity of all Gaussians to be always 1*. This prevents Gaussians from fading, which we observed to be a frequent issue due to the initial discrepancy between the Gaussian cloud and the reference images. With this modification, rather than disappearing, the Gaussians are encouraged to reposition themselves correctly within the scene.*

Second, we mitigated the issue of stretched Gaussians, which led to spiking artifacts during animation. We explored two solutions. The first, inspired by [56], involved using Isotropic Gaussians, where a single scale factor is learned, effectively reducing the 3D Gaussians to spheres. The second solution involved restricting the scale of the Gaussians (referred to as Small Anisotropic Gaussians or SAG), limiting their growth in any direction. After some experiments, we found that Small Anisotropic Gaussians yield a qualitatively greater scene reconstruction than Isotropic Gaussians, and reported a comparison in Figure 4.3. It is clear that the reconstruction made by Isotropic Gaussians results in blurred areas on the face, trousers, and shirt

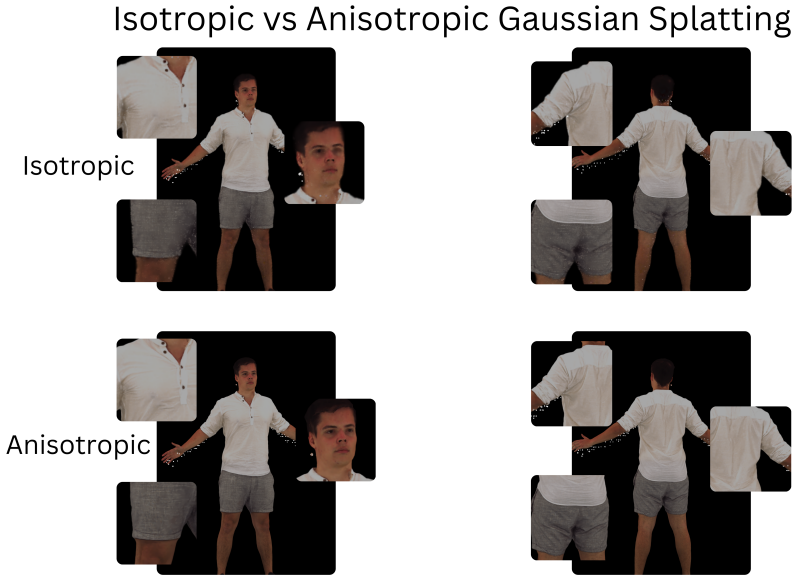

Figure 4.3. A comparison between Isotropic Gaussians (top) and Small Anisotropic Gaussians (bottom) demonstrates how the latter better preserves finer details and offers superior texture quality.

folds, whereas Small Anisotropic Gaussians capture sharper features. Therefore, we adopted SAG as the base representation of our approach.

Additionally, during scene initialization, the Gaussian Splatting algorithm initializes Gaussians uniformly within a spherical region. We found that initializing Gaussians on the surface of the input SMPL registration significantly improved reconstruction quality. While our method does not require SMPL registration, when provided (even an approximate one), it accelerates the learning process.

At the end of this reconstruction step, we obtain a point cloud of Small Anisotropic Gaussians P that visually represents the avatar (Figure 4.4), with clothing modeled independently from any underlying body structure. This enhances the flexibility of the SAG representation and avoids the limitations imposed by body-bound clothing representations.

4.3 Avatar Canonization

4.3.1 Template Fitting

The next step in our pipeline is to make the point cloud P animatable, resulting in an avatar A, and to bring it into canonical space. This step is essential for our retargeting model, which relies on canonizing both the source and target avatars. By expressing both models in the same space, we enable straightforward retargeting of the clothing between them.

For a posed SMPL mesh T', canonization involves bringing each vertex t'_i to its corresponding canonical position $\bar{t}_i$ by inverting Equations 3.20 and 3.21.However, to estimate and subsequently invert $G'(\vec{\theta}, J(\vec{\beta}; \mathcal{J}, \bar{\boldsymbol{T}}, \mathcal{S}))$, $B_S(\vec{\beta}, \mathcal{S})$, and $B_P(\vec{\theta}, \mathcal{P})$, the body shape β and pose θ parameters are required.

Due to our initial decision to use Gaussian Splatting as the underlying representation, we can leverage registration techniques to fit an SMPL mesh T' to the Gaussian

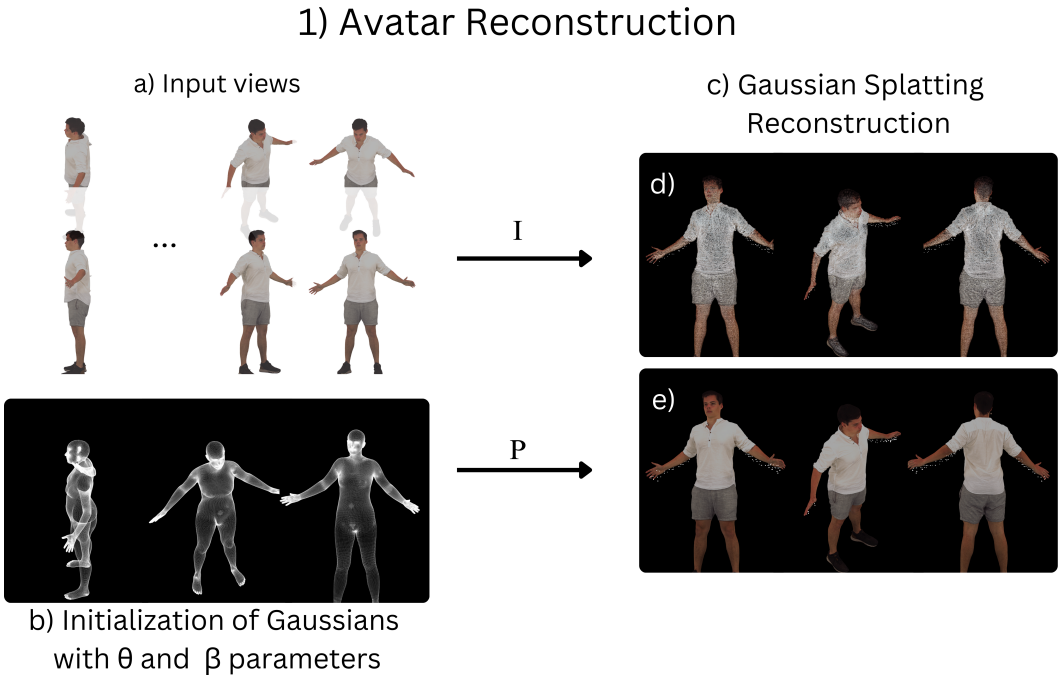

Figure 4.4. To reconstruct the appearance of the avatar, we a) get the input views, b) use the input β and θ to initialize a set of Gaussians on SMPL vertices and faces, and c) run the learning process of Gaussian Splatting. The figure also shows d) renders of the Gaussians with the scale set to e^{-10} and e) renders of the Gaussians with the learned scale.

point cloud P. This fitting allows us to estimate the transformations necessary to bring the avatar into its canonical pose. We apply NICP 3.3.2 to register a posed SMPL mesh T' on point cloud P, thus estimating the shape parameters β and pose parameters θ. The use of NICP is motivated by its robustness to noise, which is present in our learned Gaussian point cloud.

4.3.2 SMPL Modifications

NICP uses the standard SMPL model, which contains a mesh with $6,890$ vertices. To achieve finer granularity and improve results during the inverse skinning of Gaussians, we extended the SMPL model to support $27,554$ vertices by subdividing the original template mesh. This increased vertex resolution enables a denser representation of the human body, which enhances the accuracy of Linear Blend Skinning when applied to points that do not lie on the template surface.

To ensure that the forward Linear Blend Skinning (LBS) and other SMPL functionalities continue to work with the denser mesh, we updated several key components: shape blendshape matrix $\mathcal{S}$, pose blendshape matrix $\mathcal{P}$, joint regressor matrix $\mathcal{J}$, and skinning weights matrix $\mathcal{W}$. For each new vertex v_{new}, we computed the corresponding values by interpolating the matrix rows corresponding to its neighboring vertices $N(v_{new})$ in the original template mesh.

Specifically, for each of these key components $X \in \{\mathcal{S}, \mathcal{P}, \mathcal{J}, \mathcal{W}\}$, an entry x_{new} is added for each new vertex v_{new}:

$$x_{new} = \sum_{v_i \in N(v_{new})} w(v_i) \cdot x_i \tag{4.1}$$

where x_i is the value from the component associated to the i-th neighbor v_i, and $w(v_i)$ are weights determined by the distances between v_{new} and its neighbors v_i:

$$w(v_i) = 1 - \frac{d(v_{new}, v_i)}{\sum_{v_j \in N(v_{new})} d(v_{new}, v_j)} \tag{4.2}$$

where $d(v_{new}, v_\cdot)$ is the Euclidean distance between the new vertex and the neighbor $v_\cdot$.

This approach ensures that the higher-resolution SMPL model maintains consistent behavior with the original while benefiting from increased granularity, improving the quality of deformation and retargeting, particularly when dealing with noisy point clouds.

4.3.3 Point Cloud Pose Canonization

Once we have the pose parameters θ and shape parameters β, we can compute the transformation matrices $G'(\overrightarrow{\theta}, J(\overrightarrow{\beta}; \mathcal{J}, \bar{\boldsymbol{T}}, \mathcal{S}))$, $B_S(\overrightarrow{\beta}, \mathcal{S})$ and $B_P(\overrightarrow{\theta}, \mathcal{P})$ by following the forward linear blend skinning procedure outlined in 3.2.1, and then invert the transformations to obtain $G'^{-1}(\overrightarrow{\theta}, J(\overrightarrow{\beta}; \mathcal{J}, \bar{\boldsymbol{T}}, \mathcal{S}))$, $B_S(\overrightarrow{\beta}, \mathcal{S})^{-1}$ and $B_P(\overrightarrow{\theta}, \mathcal{P})^{-1}$.

With all the necessary parameters to invert the posed SMPL model, we now need to apply these transformations to the Gaussian point cloud P. It is important to note that linear blend skinning is traditionally defined on the template mesh surface, thus we have to find a way to apply the transformations to our point cloud P.

To achieve this, we use the mesh T' registered by NICP with the K-Nearest Neighbors (KNN) algorithm. For each Gaussian in our representation, we assign a list of K nearest vertices from the mesh T'. In our experiments, we evaluated several

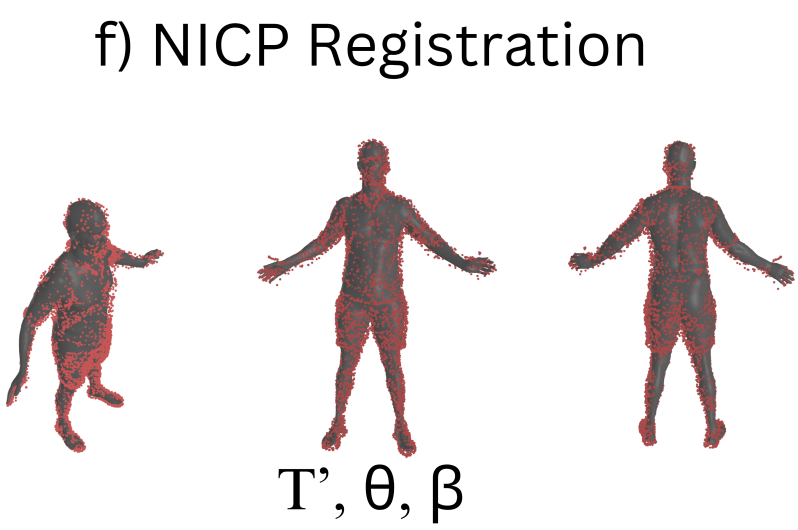

Figure 4.5. NICP registration results. The red point cloud is a downsampled set of points learned during the Gaussian Splatting step, while the grey mesh corresponds to the SMPL model fitted by NICP.

values for $K : 1, 3, 5, 7,$ *and* 11*, and found that better results were obtained with* $K = 5$*. A higher value of* K *resulted in artifacts, particularly in regions near the armpits and between the legs, while a lower value led to an inadequate distribution of Gaussians close to the joints.*

To retrieve the transformation to be applied to each point $p'_i \in P$ *and bring it into canonical space* $\bar{p}_i$*, we calculate a weighted average of the transformations applied to the neighboring vertices using the formula established in [39]:*

$$\bar{p}_i = \sum_{t_i \in \mathcal{N}(\boldsymbol{p}'_i)} \frac{\omega_i(\boldsymbol{p}'_i)}{\omega(\boldsymbol{p}'_i)} G'(\overrightarrow{\theta}, J(\overrightarrow{\beta}; \mathcal{J}, \bar{\boldsymbol{T}}, \mathcal{S}))^{-1} \boldsymbol{p}'_i \tag{4.3}$$

where $\mathcal{N}(\boldsymbol{p}'_i)$ *is the set of neighbors* t_i *associated with* $\boldsymbol{p}'_i$*, and the weighting functions are defined as follows:*

$$\omega_i(\boldsymbol{p}'_i) = exp\left(\frac{||\boldsymbol{p}'_i - t_i||_2 ||w_{nn(\boldsymbol{p}'_i)} - w_i||_2}{2\sigma^2}\right) \tag{4.4}$$

$$\omega(\boldsymbol{p}'_i) = \sum_{t_i \in \mathbb{N}(\boldsymbol{p}'_i)} \omega_i(\boldsymbol{p}'_i) \tag{4.5}$$

These weighting functions assign greater importance to closer neighbors while still considering more distant ones. In these equations, $nn(\boldsymbol{p}'_i)$ *is the index of the nearest neighbor vertex of* $\boldsymbol{p}'_i$ *in the mesh* T' *fitted by NICP,* σ *is a constant that determines the influence of distance, and* $w_i \in \mathbb{R}^{nk}$ *are the blending weights corresponding to* t_i*.*

4.3.4 Point Cloud Shape Canonization

After applying the inverse transformation described in 4.3 to every Gaussian, we obtain the canonical Gaussians $\bar{P}$ *and the animatable avatar* A *is represented*

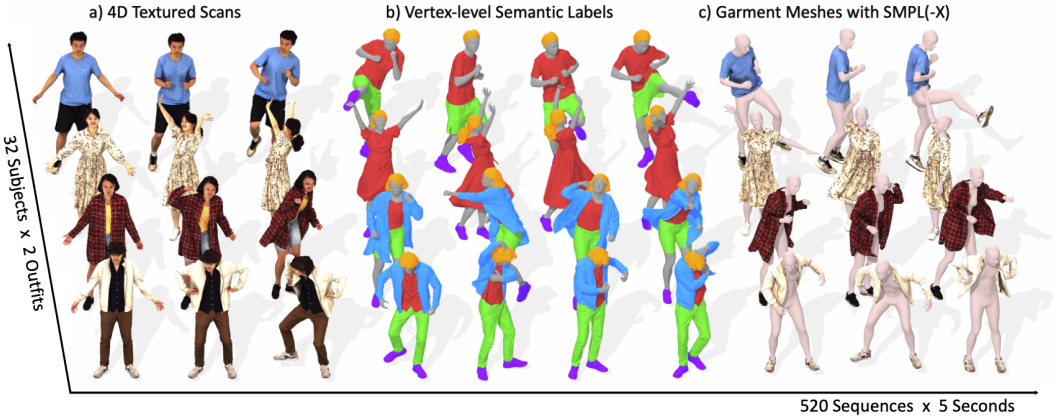

Figure 4.6. Image from 4D-Dress dataset [161]. a) High-Resolution temporal Textured Scans; b) Vertex-level Semantic Labels: skin (grey), hair (orange), shoes (purple), upper clothing (red), lower clothing (green), and outer garments (blue); c) SMPL(-X) registrations.

by $\bar{P}$ and the T-Posed SMPL mesh registered by NICP T. At this stage, we can employ linear blend skinning to animate our avatar. However, we cannot directly swap clothing between a source and a target avatar, as they may differ in body shape (e.g. the target could be taller, skinnier, or shorter than the source). To address this, we subtract the pose shapes and blend shapes from the T-posed avatars, inverting the process described in Formula 3.21. This step ensures that both the source and target avatars are deformed to match the template avatar $\bar{\boldsymbol{T}}'$, thereby standardizing their physiognomy.

4.4 Avatar Retargeting

4.4.1 Clothes Mask

At this point, we need to transfer clothes between the avatars. To achieve this, we require a mask that segments the Gaussians and classifies them either as clothing or skin. This mask is represented as a vector $\mathbf{M} \in \mathbb{N}^J$, where $J = |\bar{P}|$ is the total number of Gaussians in our representation. Each element of this vector is an integer indicating the type of garment. For our method, we define six garment types, each associated with a specific integer: 0 for skin, 1 for hair, 2 for shoes, 3 for upper clothing, 4 for lower clothing, and 5 for outer garments. For a better understanding of clothing masks and outer garments, we refer to Figure 4.6, where in b) some examples are reported. Our method can retarget any set of Gaussians between source and target if a mask is provided. The choice to use the described clothing segmentation is bound to the labels offered by the dataset used for our experiments (4D-Dress [161]).

Our pipeline does not actively segment clothing; instead, it expects the mask $\mathbf{M}$ as input. Various techniques can automate the segmentation process. One approach could involve running a segmentation algorithm on the input images, such as Segment Anithing ([134], [70]), and then projecting the resulting labels onto the scene ([142], [82]). Other methods, such as [5], and [152], work directly in 3D, taking meshes or point clouds to perform segmentation or recover garments. However, we assume the mask is provided, as estimating it falls outside the scope of our pipeline. Our primary aim is to explore and develop advanced techniques for transferring clothing

between avatars by aligning them within a common space, although this aspect is not the main objective of the thesis.

4.4.2 Clothes Transfer

Once we have the clothes mask $\mathbf{M}$*, we can remove the Gaussians corresponding to those garments from the target avatar and transfer the Gaussians corresponding to the same clothing type from the source avatar. This transfer occurs in canonical pose and shape space, ensuring that the two avatars remain aligned.*

Subsequently, we can use linear blend skinning (LBS) to animate the target avatar with the retargeted clothing. Some parts of the skin on the target avatar may be obscured by the source clothing, necessitating their removal to avoid artifacts. If a body part becomes exposed after the retargeting process, we could reconstruct it by leveraging the underlying SMPL model. However, in practice, we leave it as it is, without reconstructing the missing body part.

4.4.3 Transformation Optimization

To optimize the avatar's animation, rather than rotating each Gaussian directly, we first add pose shape and blend shape deformations into the Gaussian positions, and subsequently apply the process utilized by Gaussian Splatting (GS) during covariance estimation (Equation 3.4), to apply the skinning transformation:

$$\Sigma = G'(\overrightarrow{\theta}, J(\overrightarrow{\beta}; \mathcal{J}, \bar{\boldsymbol{T}}, \mathcal{S}))^{-1} R S S^T R^T (G'(\overrightarrow{\theta}, J(\overrightarrow{\beta}; \mathcal{J}, \bar{\boldsymbol{T}}, \mathcal{S}))^{-1})^T \tag{4.6}$$

By following this method, we can apply LBS easily and efficiently during the rendering process of Gaussian Splatting.

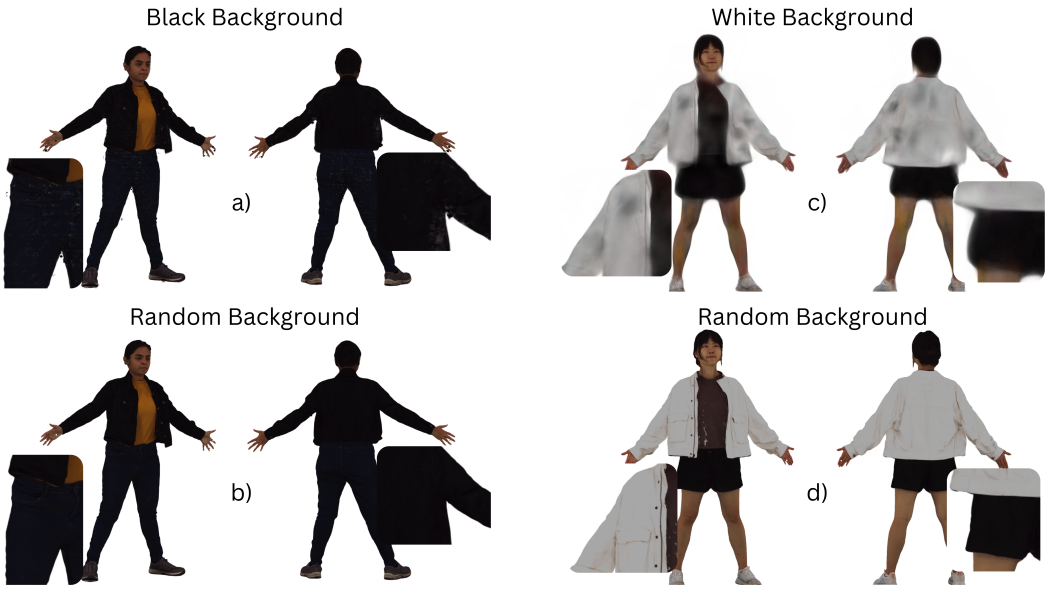

Figure 4.7. Effect of a random background during Gaussian Optimization. Illustrated are a) an avatar wearing dark clothes and trained on a black background; b) an avatar wearing dark clothes and trained on a random background; c) an avatar wearing light clothes and trained on a white background; d) an avatar wearing light clothes and trained on a random background.

4.5 Implementation Details

We run a 2000-step optimization for the initial Gaussian Splatting training, on a list of 20 rotating cameras disposed on 0 and 40 degrees of elevation. The initial and final positional learning rates are set to respectively $1.6e^{-4}$ *and* $1.6e^{-7}$*. We disabled the opacity reset implemented by the original Gaussian Splatting paper, and ran densification intervals every* 100 *step, starting from step* 200*. To have small Anisotropic Gaussians, we lowered the identifying threshold to* $2e^{-5}$*, the size threshold to* 0.25*, the camera extent was multiplied by* 0.5*, and at each step, we randomly selected a background, either white or black, to prevent dark or light clothes to blend with a static background and resulting in blurry silhouettes (Figure 4.7. For the NICP fitting, we left the default parameters set by the authors in their official implementation and disabled the SMPL+D step as it is not useful for our task. The whole Avatar Reconstruction and Canonization steps take about 3 minutes, while the retargeting of clothes in canonical space requires about 10 seconds. Our method can then animate and render the retargeted avatar at 20 fps, allowing a real-time interaction.*

Chapter 5

Evaluation

For our experiments, we require multiview images of both the source and target avatars to reconstruct their respective representations and perform retargeting. However, there are no publicly available datasets specifically designed for our retargeting task. As a result, we chose to validate our pipeline in two parts: by separating the validation of the reconstruction phase from the retargeting phase. This approach leverages the extensive body of work on reconstruction, where well-established metrics allow us to perform meaningful comparisons.

We selected the 4D-Dress dataset due to its comprehensive offering of 32 subjects, each with two different outfits and over 520 motion sequences. The dataset includes SMPL registrations and clothing labels, and it features sequences with and without an outer garment, such as jackets, coats, or t-shirts. This provides an excellent resource for both reconstruction and preliminary validation of clothing retargeting.

5.1 Reconstruction Evaluation

In our setup, we use as input multiview images $I = \{I_0, I_1, \ldots, I_N\}$ of a person posed with parameters θ and β, along with a target pose θ'. The objective is to output the avatar posed in θ'. Since the 4D-Dress dataset [161] provides high-quality temporal scans, we can select two scans taken at different timesteps, using one as the source and the other as the source for our ground-truth label images $Y = \{Y_0, Y_1, \ldots, Y_N\}$ to evaluate the reconstruction. This partial validation is particularly useful as it allows us to assess how the model handles clothing motion, giving us insight into how garments are retargeted onto the same individual in different poses.

We compared our method with HaveFun [177], as it represents one of the most recent approaches for the task of generating avatars from multiview images. To ensure a fair comparison, we disabled HaveFun's expression modeling, since our work focuses on the accurate reconstruction of clothing. Consequently, modeling hands and facial expressions is outside the scope of our research.

5.1.1 Qualitative Evaluation

We evaluated the models using 10 different scenes and $N = 20$ orbital cameras, with qualitative results shown in Figures 5.1, 5.2, and 5.3. These images demonstrate that our method achieves better visual representations of clothing, capturing more details than HaveFun. Our model handles garments that are farther from body registration more effectively, while HaveFun tends to warp them excessively. Additionally, the visual quality of our textures is superior, with our method capturing fine details

Figure 5.1. Qualitative comparisons between our method and HaveFun. The figure shows experiments on subjects 123 inner take1, 123 outer take11, and 147 inner take10.

Figure 5.2. Qualitative comparisons between our method and HaveFun on subjects 122 inner take1, 123 outer take13, and 147 outer take17.

such as buttons on the shirt in Figure 5.2 row 1, and white stripes on the t-shirt in Figure 5.3 rows 2/3, whereas HaveFun's output appears blurred. Our approach also performs better when the source and target poses differ significantly, particularly when the source pose has highly bent arms, where HaveFun struggles to recreate a correct avatar.

However, our method does not reconstruct occluded areas (such as armpits), which leads to sparse Gaussian coverage in certain regions, as seen in Figure 5.1 row 1.

5.1.2 Quantitative Evaluation

For quantitative evaluation, we employed metrics commonly used in avatar reconstruction, specifically $SSIM$ and $PSNR$. Given a render of a posed avatar x

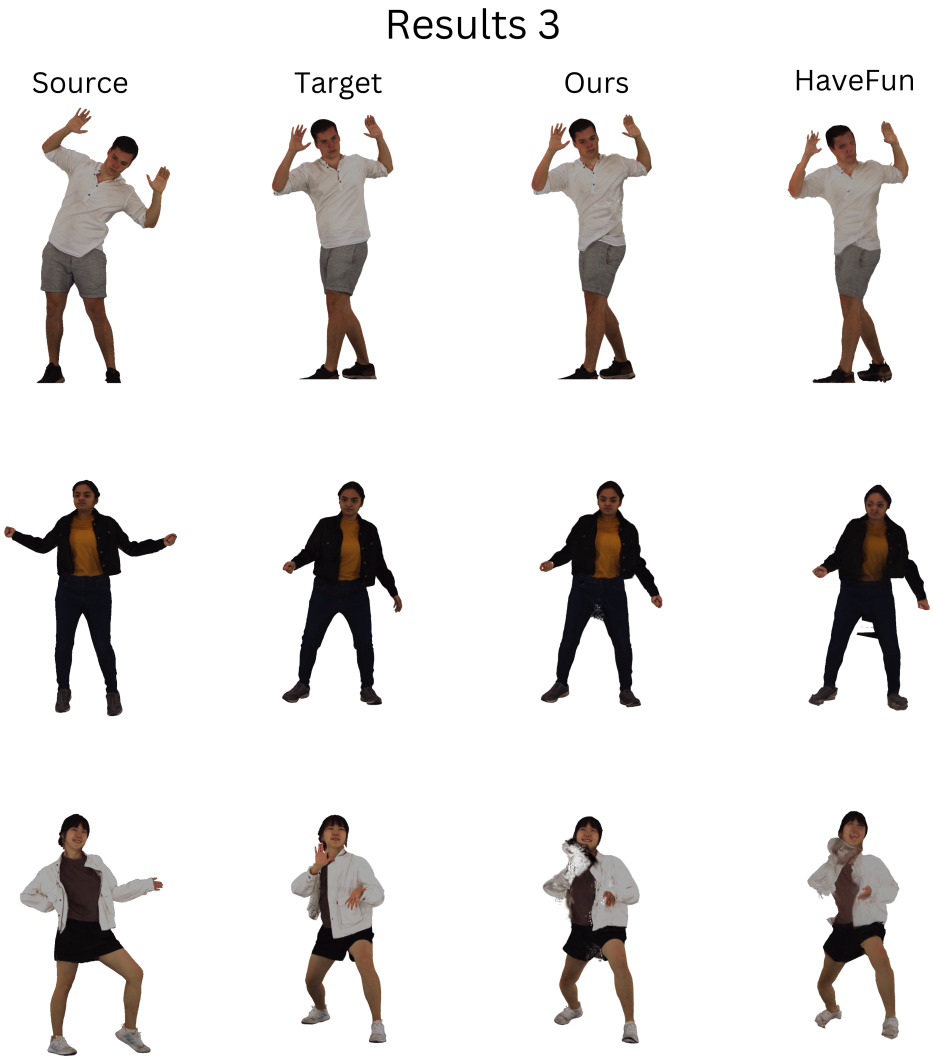

Figure 5.3. Qualitative comparisons between our method and HaveFun on subjects 127 outer take16, 127 outer take9, and 123 outer take9.

Metric	Ours	HaveFun
SSIM ↑	**0.923**	0.922
PSNR ↑	**21.96**	19.56
RUNTIME(min.) ↓	**3.5**	60

Table 5.1. Quantitative comparison between our method and HaveFun. The table reports the SSIM, PSNR, and training time for our method and HaveFun. These values were obtained by running an evaluation over 10 samples randomly extracted from the dataset.

and the expected output y, the $SSIM$ is defined as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{5.1}$$

where μ_x and μ_y represent the pixel sample means of x and y, σ_x^2 and σ_y^2 are the variances of x and y, σ_{xy} is the covariance of x and y, and c_1, c_2 are constants used to stabilize the calculation in cases of weak denominators. The $PSNR$ is defined as:

$$PSNR(x,y) = 20 \cdot log_{10}\left(\frac{MAX\{x\}}{\sqrt{MSE(x,y)}}\right) \tag{5.2}$$

where the mean squared error (MSE) is:

$$MSE(x,y) = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}||x_{i,j} - y_{i,j}||^2 \tag{5.3}$$

and $MAX\{x\}$ is the maximum value of all pixels in the image x.

The results of our experiments are shown in Table 5.1.

These results indicate that our method generally outperforms HaveFun. Although both methods achieve a similar SSIM score, our method yields a significantly higher PSNR (an increase of nearly 2.5 points), consuming only 5% of the training time of HaveFun. This demonstrates the effectiveness of our approach for avatar reconstruction and highlights its ability to retarget clothing on the same person in different poses with high accuracy.

5.2 Retargeting Evaluation

To evaluate the retargeting capabilities of our method, we conducted two qualitative experiments designed to assess its performance in different scenarios.

5.2.1 Inter-Person Clothing Transfer

In the first experiment, we took multiview images of a source person and a target person and applied the source's clothing onto the target (Inter-Person clothing transfer experiment). After retargeting the clothes, we animated the target avatar using LBS to assess how well the model moves the Gaussians. Results experiments in Figure 5.4 demonstrated that our method can effectively retarget clothes and follow the physiognomy of the target avatar. Results also show that tight clothes help with the NICP registration for the target avatar, thus producing a better result after the retargeting process. We did not reconstruct missing body parts (e.g., occluded areas), and some artifacts appeared on the hands due to inaccurate NICP registration. Overall, the pipeline performed well, yielding coherent and realistic results.

Figure 5.4. Results for the Inter-Person clothing transfer experiment. The first column has the source avatar, the second column has the target avatar, and the last three columns show three different poses of the target avatar after the clothing transfer.

5.2.2 Intra-Person Garment Transfer

In the second experiment, we leveraged 4D-Dress sequences of the same person with and without an outer garment and transferred it from the source to the target avatar (Intra-Person clothing transfer experiment). The results in Figure 5.5 revealed that our method struggles with garment parts that are close to hidden regions, such as the armpits, if these regions were not visible in the source scan. Additionally, for loose garments, Gaussians occasionally detached from the body, creating visible gaps or floating artifacts. Despite these challenges, our method successfully handled the majority of the retargeting task, demonstrating its robustness in preserving most of the clothing structure and details even in these complex scenarios.

Figure 5.5. Results for the Intra-Person clothing transfer experiment. The first column has the source avatar, the second column has the target avatar, and the last three columns show three different poses of the target avatar after the clothing transfer.

Chapter 6

Conclusions

In this thesis, we have explored the challenging task of retargeting clothes between avatars, focusing on transferring garments from one subject to another while preserving key details and garment characteristics. Our primary goal was to address the lack of methods capable of handling this task in a way that is both effective and efficient, particularly for 3D representations that require real-time rendering and animation.

The core problem we sought to solve is how to accurately transfer clothing between avatars, especially when the source and target avatars may differ significantly in size, shape, or pose. Traditional methods tend to fall short when dealing with garments that are far from body registration or when there are drastic differences in poses between the source and target. Our approach addresses garments retargeting by warping clothing into a common space where the garments can be transferred more naturally, without compromising their structure or detail.

The proposed method leverages Gaussian Splatting for avatar reconstruction and clothing retargeting, focusing on capturing sharp details such as fabric folds and textures. We validated our approach through experiments that demonstrated its competitive performance in avatar reconstruction, particularly when compared to a state-of-the-art method called HaveFun. Our model produced sharper textures and more accurate clothing details, even in challenging cases where garments were distant from the body. Additionally, we showcased the ability of our method to retarget clothes by transferring garments between different individuals and across different poses of the same person, successfully preserving garment structure and detail in both scenarios, even when there were significant differences in the shape or pose of the source and target avatars.

One of the challenges we encountered was the absence of specific datasets designed for garment retargeting. To overcome this, we built our own data by rendering multiview images from scans provided by the 4D-Dress dataset. Although this dataset allowed us to conduct our experiments, it highlighted the need for more comprehensive benchmarks that specifically address the task of retargeting.

Our quantitative evaluation demonstrated the effectiveness of our approach, with our method achieving higher scores than a state-of-the-art method in Avatar Reconstruction. Qualitatively, our method captured finer clothing details, particularly in cases where garments moved significantly between poses. These results highlighted the potential of our method for practical applications in virtual try-on systems and digital fashion.

While this work focuses primarily on the task of retargeting, the insights gained and the techniques developed lay the foundation for future research in related fields, such as automated garment segmentation for noisy point clouds and real-time clothing

synthesis. Future work could explore expanding the dataset for more diverse clothing types, optimizing performance for real-time applications, expanding the method to single-view reconstructions, and integrating the retargeting process with more advanced neural garment simulation to further enhance realism.

Bibliography

[1] *Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single RGB camera. In* IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2019.*

[2] *Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In* IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8387–8397, Jun 2018. CVPR Spotlight Paper.*

[3] *Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people.* ACM Trans. Graph.*, 24(3):408–416, July 2005.*

[4] *Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis.* SCAPE: Shape Completion and Animation of People*. Association for Computing Machinery, New York, NY, USA, 1 edition, 2023.*

[5] *Dimitrije Antić, Garvita Tiwari, Batuhan Ozcomlekci, Riccardo Marin, and Gerard Pons-Moll. CloSe: A 3D clothing segmentation dataset and model. In* International Conference on 3D Vision (3DV)*, March 2024.*

[6] *Matthieu Armando, Laurence Boissieux, Edmond Boyer, Jean-Sebastien Franco, Martin Humenberger, Christophe Legras, Vincent Leroy, Mathieu Marsot, Julien Pansiot, Sergi Pujades, Rim Rekik, Gregory Rogez, Anilkumar Swamy, and Stefanie Wuhrer. 4dhumanoutfit: a multi-subject 4d dataset of human motion sequences in varying outfits exhibiting large displacements.* Computer Vision and Image Understanding*, 2023.*

[7] *Timur Bagautdinov, Chenglei Wu, Tomas Simon, Fabián Prada, Takaaki Shiratori, Shih-En Wei, Weipeng Xu, Yaser Sheikh, and Jason Saragih. Driving-signal aware full-body avatars.* ACM Trans. Graph.*, 40(4), July 2021.*

[8] *Allison H. Baker, Alexander Pinard, and Dorit M. Hammerling. Dssim: a structural similarity index for floating-point data, 2023.*

[9] *Seungbae Bang, Maria Korosteleva, and Sung-Hee Lee. Estimating garment patterns from static scan data.* Computer Graphics Forum*, 40(6):273–287, 2021.*

[10] *Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields.* ICCV*, 2021.*

[11] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. CVPR*, 2022.*

[12] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Cloth3d: clothed 3d humans. In European Conference on Computer Vision*, pages 344–359. Springer, 2020.*

[13] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.*

[14] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3d human reconstruction. In European Conference on Computer Vision (ECCV)*. Springer, August 2020.*

[15] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019.*

[16] Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8726–8737, June 2023.*

[17] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image, 2016.

[18] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. Faust: Dataset and evaluation for 3d mesh registration. In 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014.*

[19] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic faust: Registering human bodies in motion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.*

[20] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Levy. Polygon Mesh Processing*. Ak Peters Series. Taylor & Francis, 2010.*

[21] Beijia Chen, Yuefan Shen, Qing Shuai, Xiaowei Zhou, Kun Zhou, and Youyi Zheng. Anidress: Animatable loose-dressed avatar from sparse views using garment rigging model. arXiv preprint arXiv:2401.15348*, 2024.*

[22] Xipeng Chen, Guangrun Wang, Dizhong Zhu, Xiaodan Liang, Philip H. S. Torr, and Liang Lin. Structure-preserving 3d garment modeling with neural sewing machines. In Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc.*

[23] Yinpeng Chen, Zicheng Liu, and Zhengyou Zhang. Tensor-based human body modeling. In 2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 105–112, 2013.*

[24] Yizheng Chen, Rengan Xie, Sen Yang, Linchen Dai, Hongchun Sun, Yuchi Huo, and Rong Li. Single-view 3d garment reconstruction using neural volumetric rendering. IEEE Access*, 12:49682–49693, 2024.*

[25] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In The Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.*

[26] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.*

[27] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The trimmed iterative closest point algorithm. In 2002 International Conference on Pattern Recognition*, volume 3, pages 545–548 vol.3, 2002.*

[28] Seunghwan Choi, Sunghyun Park, Minsoo Lee, and Jaegul Choo. Viton-hd: High-resolution virtual try-on via misalignment-aware normalization. In Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2021.*

[29] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. ACM Trans. Graph.*, 34(4), July 2015.*

[30] Enric Corona, Gerard Pons-Moll, Guillem Alenyà, and Francesc Moreno-Noguer. Learned vertex descent: A new direction for 3d human model fitting, 2022.

[31] Enric Corona, Albert Pumarola, Guillem Alenyà, Gerard Pons-Moll, and Francesc Moreno-Noguer. Smplicit: Topology-aware generative model for clothed people. In CVPR*, 2021.*

[32] Luca De Luigi, Ren Li, Benoit Guillard, Mathieu Salzmann, and Pascal Fua. DrapeNet: Garment Generation and Self-Supervised Draping. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.*

[33] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.*

[34] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bochao Wang, Hanjiang Lai, Jia Zhu, Zhiting Hu, and Jian Yin. Towards multi-pose guided virtual try-on network. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.*

[35] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, Pushmeet Kohli, Vladimir Tankovich, and Shahram Izadi. Fusion4d: real-time performance capture of challenging scenes. ACM Trans. Graph.*, 35(4), July 2016.*

[36] Qi Fang, Qing Shuai, Junting Dong, Hujun Bao, and Xiaowei Zhou. Reconstructing 3d human pose by watching humans in the mirror. In CVPR*, 2021.*

[37] Qiao Feng, Yebin Liu, Yu-Kun Lai, Jingyu Yang, and Kun Li. Fof: Learning fourier occupancy field for monocular real-time human reconstruction, 2023.

[38] Yao Feng, Weiyang Liu, Timo Bolkart, Jinlong Yang, Marc Pollefeys, and Michael J. Black. Learning disentangled avatars with hybrid 3d representations. arXiv*, 2023.*

[39] Yao Feng, Jinlong Yang, Marc Pollefeys, Michael J. Black, and Timo Bolkart. Capturing and animation of body and clothing from monocular video. In SIGGRAPH Asia 2022 Conference Papers*, SA '22, 2022.*

[40] Daiheng Gao, Xu Chen, Xindi Zhang, Qi Wang, Ke Sun, Bang Zhang, Liefeng Bo, and Qixing Huang. Cloth2tex: A customized cloth texture generation pipeline for 3d virtual try-on, 2023.

[41] Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps, 2021.

[42] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 43–54, New York, NY, USA, 1996. Association for Computing Machinery.*

[43] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In Proceedings of the european conference on computer vision (ECCV)*, pages 230–246, 2018.*

[44] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.*

[45] Peng Guan, Loretta Reiss, David A. Hirshberg, Alexander Weiss, and Michael J. Black. Drape: Dressing any person. ACM Trans. Graph.*, 31(4), July 2012.*

[46] Erhan Gundogdu, Victor Constantin, Amrollah Seifoddini, Minh Dang, Mathieu Salzmann, and Pascal Fua. Garnet: A two-stream network for fast and accurate 3d cloth draping. In IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019.*

[47] Marc Habermann, Lingjie Liu, Weipeng Xu, Gerard Pons-Moll, Michael Zollhoefer, and Christian Theobalt. Hdhumans: A hybrid approach for high-fidelity digital humans. Proc. ACM Comput. Graph. Interact. Tech.*, 6(3), aug 2023.*

[48] Oshri Halimi, Tuur Stuyck, Donglai Xiang, Timur Bagautdinov, He Wen, Ron Kimmel, Takaaki Shiratori, Chenglei Wu, Yaser Sheikh, and Fabian Prada. Pattern-based cloth registration and sparse-view animation. ACM Trans. Graph.*, 41(6), November 2022.*

[49] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. Viton: An image-based virtual try-on network. In CVPR*, 2018.*

[50] Kai He, Kaixin Yao, Qixuan Zhang, Jingyi Yu, Lingjie Liu, and Lan Xu. Dresscode: Autoregressively sewing and generating garments from text guidance. ACM Transactions on Graphics (TOG)*, 43(4):1–13, 2024.*

[51] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. ACM Trans. Graph.*, 37(6), December 2018.*

[52] Zhu Heming, Cao Yu, Jin Hang, Chen Weikai, Du Dong, Wang Zhangye, Cui Shuguang, and Han Xiaoguang. Deep fashion3d: A dataset and benchmark for 3d garment reconstruction from single images. In Computer Vision – ECCV 2020*, pages 512–530. Springer International Publishing, 2020.*

[53] Osamu Hirose. A bayesian formulation of coherent point drift. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7):2269–2286, 2021.*

[54] Osamu Hirose. Geodesic-based bayesian coherent point drift. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5816–5832, 2023.*

[55] Chia-Wei Hsieh, Chieh-Yun Chen, Chien-Lung Chou, Hong-Han Shuai, Jiaying Liu, and Wen-Huang Cheng. Fashionon: Semantic-guided image-based virtual try-on with detailed human and clothing information. In Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 275–283, New York, NY, USA, 2019. Association for Computing Machinery.*

[56] Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians, 2024.

[57] Shoukang Hu and Ziwei Liu. Gauhuman: Articulated gaussian splatting from monocular human videos, 2023.

[58] Yinghao Huang, Federica Bogo, Christoph Lassner, Angjoo Kanazawa, Peter V. Gehler, Ijaz Akhter, and Michael J. Black. Towards accurate markerless human shape and pose estimation over time, 2018.

[59] Mustafa Işık, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. Humanrf: High-fidelity neural radiance fields for humans in motion. ACM Transactions on Graphics (TOG)*, 42(4):1–12, 2023.*

[60] Thibaut Issenhuth, Jérémie Mary, and Clément Calauzènes. Do not mask what you do not need to mask: A parser-free virtual try-on. In Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX*, page 619–635, Berlin, Heidelberg, 2020. Springer-Verlag.*

[61] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5885–5894, October 2021.*

[62] Boyi Jiang, Juyong Zhang, Yang Hong, Jinhao Luo, Ligang Liu, and Hujun Bao. Bcnet: Learning body and cloth shape from a single image. In European Conference on Computer Vision*. Springer, 2020.*

[63] Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. Neuman: Neural human radiance field from a single video. In Proceedings of the European conference on computer vision (ECCV)*, 2022.*

[64] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation, 2021.

[65] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose, 2018.

[66] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics*, 42(4), July 2023.*

[67] Jeongho Kim, Guojung Gu, Minho Park, Sunghyun Park, and Jaegul Choo. Stableviton: Learning semantic correspondence with latent diffusion model for virtual try-on. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8176–8185, June 2024.*

[68] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In CVPR*, 2022.*

[69] Taeksoo Kim, Byungjun Kim, Shunsuke Saito, and Hanbyul Joo. Gala: Generating animatable layered assets from a single scan, 2024.

[70] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. arXiv:2304.02643*, 2023.*

[71] Muhammed Kocabas, Chun-Hao P. Huang, Otmar Hilliges, and Michael J. Black. Pare: Part attention regressor for 3d human body estimation, 2021.

[72] Maria Korosteleva, Timur Levent Kesdogan, Fabian Kemper, Stephan Wenninger, Jasmin Koller, Yuhan Zhang, Mario Botsch, and Olga Sorkine-Hornung. GarmentCodeData: A dataset of 3D made-to-measure garments with sewing patterns. In Computer Vision – ECCV 2024*, 2024.*

[73] Maria Korosteleva and Sung-Hee Lee. Generating datasets of 3d garments with sewing patterns. arXiv preprint arXiv:2109.05633*, 2021.*

[74] Maria Korosteleva and Sung-Hee Lee. Neuraltailor: reconstructing sewing pattern structures from 3d point clouds of garments. ACM Trans. Graph.*, 41(4), July 2022.*

[75] Maria Korosteleva and Olga Sorkine-Hornung. GarmentCode: Programming parametric sewing patterns. ACM Transaction on Graphics*, 42(6), 2023. SIGGRAPH ASIA 2023 issue.*

[76] Shizuma Kubo, Yusuke Iwasawa, Masahiro Suzuki, and Yutaka Matsuo. Uvton: Uv mapping to consider the 3d structure of a human in image-based virtual try-on network. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.*

[77] Hyun-Song Kwon and Sung-Hee Lee. Deepiron: Predicting unwarped garment texture from a single image, 2023.

[78] Christoph Lassner, Gerard Pons-Moll, and Peter V. Gehler. A generative model for people in clothing. In Proceedings of the IEEE International Conference on Computer Vision*, 2017.*

[79] Jeonghaeng Lee, Duc Nguyen, Jongyoo Kim, Jiwoo Kang, and Sanghoon Lee. Double reverse diffusion for realistic garment reconstruction from images. Eng. Appl. Artif. Intell.*, 127(PB), February 2024.*

[80] Jialian Li, Jingyi Zhang, Zhiyong Wang, Siqi Shen, Chenglu Wen, Yuexin Ma, Lan Xu, Jingyi Yu, and Cheng Wang. Lidarcap: Long-range marker-less 3d human motion capture with lidar point clouds, 2022.

[81] Mengtian Li, Shengxiang Yao, Zhifeng Xie, and Keyu Chen. Gaussianbody: Clothed human reconstruction via 3d gaussian splatting, 2024.

[82] Mingrui Li, Shuhong Liu, and Heng Zhou. Sgs-slam: Semantic gaussian splatting for neural dense slam. arXiv preprint arXiv:2402.03246*, 2024.*

[83] Ren Li, Corentin Dumery, Benoit Guillard, and Pascal Fua. Garment Recovery with Shape and Deformation Priors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.*

[84] Ren Li, Benoit Guillard, and Pascal Fua. ISP: Multi-Layered Garment Draping with Implicit Sewing Patterns. In Advances in Neural Information Processing Systems*, 2023.*

[85] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhofer, Jurgen Gall, Angjoo Kanazawa, and Christoph Lassner. Tava: Template-free animatable volumetric actors. 2022.

[86] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017.*

[87] Xiongzheng Li, Jinsong Zhang, Yu-Kun Lai, Jingyu Yang, and Kun Li. High-quality animatable dynamic garment reconstruction from monocular videos, 2023.

[88] Yifei Li, Hsiao-yu Chen, Egor Larionov, Nikolaos Sarafianos, Wojciech Matusik, and Tuur Stuyck. DiffAvatar: Simulation-ready garment optimization with differentiable simulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024.*

[89] Zhe Li, Yipengjing Sun, Zerong Zheng, Lizhen Wang, Shengping Zhang, and Yebin Liu. Animatable and relightable gaussians for high-fidelity human avatar modeling. arXiv preprint arXiv:2311.16096v4*, 2024.*

[90] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.*

[91] Siyou Lin, Zhe Li, Zhaoqi Su, Zerong Zheng, Hongwen Zhang, and Yebin Liu. Layga: Layered gaussian avatars for animatable clothing transfer. In ACM SIGGRAPH 2024 Conference Papers*, SIGGRAPH '24, New York, NY, USA, 2024. Association for Computing Machinery.*

[92] Siyou Lin, Zhe Li, Zhaoqi Su, Zerong Zheng, Hongwen Zhang, and Yebin Liu. Layga: Layered gaussian avatars for animatable clothing transfer, 2024.

[93] Siyou Lin, Boyao Zhou, Zerong Zheng, Hongwen Zhang, and Yebin Liu. Leveraging intrinsic properties for non-rigid garment alignment. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14485–14496, October 2023.*

[94] Lijuan Liu, Xiangyu Xu, Zhijie Lin, Jiabin Liang, and Shuicheng Yan. Towards garment sewing pattern reconstruction from a single image. ACM Transactions on Graphics (SIGGRAPH Asia)*, 2023.*

[95] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. NeurIPS*, 2020.*

[96] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023.

[97] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In European Conference on Computer Vision*, pages 210–227. Springer, 2022.*

[98] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. ACM Trans. Graph.*, 34(6), oct 2015.*

[99] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. Deepvcp: An end-to-end deep neural network for point cloud registration. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, October 2019.*

[100] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. Learning to Dress 3D People in Generative Clothing. In Computer Vision and Pattern Recognition (CVPR)*, June 2020.*

[101] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. Learning to dress 3d people in generative clothing. In Computer Vision and Pattern Recognition (CVPR)*, June 2020.*

[102] Nadia Magnenat-Thalmann and Daniel Thalmann. Handbook of virtual humans*. John Wiley and Sons, 2005.*

[103] Sahib Majithia, Sandeep N. Parameswaran, Sadbhavana Babar, Vikram Garg, Astitva Srivastava, and Avinash Sharma. Robust 3d garment digitization from monocular 2d images for 3d virtual try-on systems. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3428–3438, January 2022.*

[104] Riccardo Marin, Enric Corona, and Gerard Pons-Moll. Nicp: Neural icp for 3d human registration at scale, 2024.

[105] Riccardo Marin, Simone Melzi, Emanuele Rodola, and Umberto Castellani. Farm: Functional automatic registration method for 3d human bodies. In Computer Graphics Forum*, volume 39, pages 160–173. Wiley Online Library, 2020.*

[106] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In ECCV*, 2020.*

[107] Matiur Rahman Minar and Heejune Ahn. Cloth-vton: Clothing three-dimensional reconstruction for hybrid image-based virtual try-on. In Asian Conference on Computer Vision (ACCV)*, 2020.*

[108] Aymen Mir, Thiemo Alldieck, and Gerard Pons-Moll. Learning to transfer texture from clothing images to 3d humans. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020.*

[109] Arthur Moreau, Jifei Song, Helisa Dhamo, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. Human gaussian splatting: Real-time rendering of animatable avatars. In CVPR*, 2024.*

[110] Davide Morelli, Matteo Fincato, Marcella Cornia, Federico Landi, Fabio Cesari, and Rita Cucchiara. Dress Code: High-Resolution Multi-Category Virtual Try-On. In Proceedings of the European Conference on Computer Vision*, 2022.*

[111] Pietro Musoni, Simone Melzi, and Umberto Castellani. Gim3d: A 3d dataset for garment segmentation. In Daniela Cabiddu, Teseo Schneider, Dario Allegra, Chiara Eva Catalano, Gianmarco Cherchi, and Riccardo Scateni, editors, Smart Tools and Applications in Graphics - Eurographics Italian Chapter Conference*. The Eurographics Association, 2022.*

[112] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010.*

[113] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics*, 41(4):1–15, July 2022.*

[114] Assaf Neuberger, Eran Borenstein, Bar Hilleli, Eduard Oks, and Sharon Alpert. Image based virtual try-on network from unpaired data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.*

[115] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs, 2021.

[116] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.*

[117] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020)*, pages 3501 – 3512, Piscataway, NJ, 2020. IEEE.*

[118] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model-based human pose and shape estimation, 2018.

[119] Ahmed A A Osman, Timo Bolkart, and Michael J. Black. STAR: A sparse trained articulated human body regressor. In European Conference on Computer Vision (ECCV)*, pages 598–613, 2020.*

[120] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019.*

[121] Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020.*

[122] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.*

[123] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image, 2018.

[124] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14294–14303, 2021.*

[125] Sida Peng, Chen Geng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Implicit neural representations with structured latent codes for human body modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.*

[126] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In CVPR*, 2021.*

[127] Frank Perbet, Sam Johnson, Minh-Tri Pham, and Bjorn Stenger. Human body shape estimation using a multi-resolution manifold forest. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.*

[128] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J. Black. Clothcap: seamless 4d clothing capture and retargeting. ACM Trans. Graph.*, 36(4), jul 2017.*

[129] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. arXiv*, 2022.*

[130] Albert Pumarola, Jordi Sanchez, Gary Choi, Alberto Sanfeliu, and Francesc Moreno-Noguer. 3DPeople: Modeling the Geometry of Dressed Humans. In International Conference on Computer Vision (ICCV)*, 2019.*

[131] Anran Qi, Sauradip Nag, Xiatian Zhu, and Ariel Shamir. Personaltailor: Personalizing 2d pattern design from 3d garment point clouds. ArXiv*, abs/2303.09695, 2023.*

[132] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. 2024.

[133] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting, 2024.

[134] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714*, 2024.*

[135] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In International Conference on Computer Vision (ICCV)*, 2021.*

[136] Iasonas Kokkinos Riza Alp Guler, Natalia Neverova. Densepose: Dense human pose estimation in the wild. 2018.

[137] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), November 2017.*

[138] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: modeling and capturing hands and bodies together. ACM Transactions on Graphics*, 36(6):1–17, November 2017.*

[139] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization, 2019.

[140] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization, 2020.

[141] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting, 2024.

[142] Qiuhong Shen, Xingyi Yang, and Xinchao Wang. Flashsplat: 2d to 3d gaussian splatting segmentation solved optimally, 2024.

[143] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In Advances in Neural Information Processing Systems (NeurIPS)*, 2021.*

[144] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In Advances in Neural Information Processing Systems*, 2019.*

[145] Mark Pauly Sofien Bouaziz, Andrea Tagliasacchi. Sparse iterative closest point. Computer Graphics Forum*, 2013.*

[146] Carsten Stoll, Juergen Gall, Edilson de Aguiar, Sebastian Thrun, and Christian Theobalt. Video-based reconstruction of animatable human characters. ACM Trans. Graph.*, 29(6), December 2010.*

[147] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction, 2022.

[148] Ke Sun, Jian Cao, Qi Wang, Linrui Tian, Xindi Zhang, Lian Zhuo, Bang Zhang, Liefeng Bo, Wenbo Zhou, Weiming Zhang, and Daiheng Gao. Outfitanyone: Ultra-high quality virtual try-on for any clothing and any person. arXiv preprint arXiv:2407.16224*, 2024.*

[149] Ramana Sundararaman, Riccardo Marin, Emanuele Rodola, and Maks Ovsjanikov. Reduced representation of deformation fields for effective non-rigid shape matching, 2022.

[150] István Sárándi and Gerard Pons-Moll. Neural localizer fields for continuous 3d human pose and shape estimation, 2024.

[151] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. arXiv*, 2022.*

[152] Garvita Tiwari, Bharat Lal Bhatnagar, Tony Tung, and Gerard Pons-Moll. Sizer: A dataset and model for parsing 3d clothing and learning size sensitive 3d clothing. In European Conference on Computer Vision (ECCV)*. Springer, August 2020.*

[153] Gül Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes, 2018.

[154] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, Liang Lin, and Meng Yang. Toward characteristic-preserving image-based virtual try-on network. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, Computer Vision – ECCV 2018*, pages 607–623, Cham, 2018. Springer International Publishing.*

[155] Kangkan Wang, Jin Xie, Guofeng Zhang, Lei Liu, and Jian Yang. Sequential 3d human pose and shape estimation from point clouds. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7275–7284, 2020.*

[156] Lingjing Wang, Jianchun Chen, Xiang Li, and Yi Fang. Non-rigid point set registration networks, 2019.

[157] Lizhen Wang, Xiaochen Zhao, Jingxiang Sun, Yuxiang Zhang, Hongwen Zhang, Tao Yu, and Yebin Liu. Styleavatar: Real-time photo-realistic portrait avatar from a single video. In ACM SIGGRAPH 2023 Conference Proceedings*, 2023.*

[158] Shaofei Wang, Božidar Antić, Andreas Geiger, and Siyu Tang. Intrinsicavatar: Physically based inverse rendering of dynamic humans from monocular videos via explicit ray tracing. In Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.*

[159] Shaofei Wang, Andreas Geiger, and Siyu Tang. Locally aware piecewise transformation fields for 3d human mesh registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7639–7648, 2021.*

[160] Shaofei Wang, Andreas Geiger, and Siyu Tang. Locally aware piecewise transformation fields for 3d human mesh registration. In Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.*

[161] Wenbo Wang, Hsuan-I Ho, Chen Guo, Boxiang Rong, Artur Grigorev, Jie Song, Juan Jose Zarate, and Otmar Hilliges. 4d-dress: A 4d dataset of real-world human clothing with semantic annotations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.*

[162] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration, 2019.

[163] Jing Wen, Xiaoming Zhao, Zhongzheng Ren, Alexander G. Schwing, and Shenlong Wang. Gomavatar: Efficient animatable human modeling from monocular video using gaussians-on-mesh, 2024.

[164] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video, 2022.

[165] Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. Multi-view neural human rendering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1682–1691, 2020.*

[166] Donglai Xiang, Timur Bagautdinov, Tuur Stuyck, Fabian Prada, Javier Romero, Weipeng Xu, Shunsuke Saito, Jingfan Guo, Breannan Smith, Takaaki Shiratori, Yaser Sheikh, Jessica Hodgins, and Chenglei Wu. Dressing avatars: Deep photorealistic appearance for physically simulated clothing. ACM Trans. Graph.*, 41(6), November 2022.*

[167] Donglai Xiang, Fabian Prada, Timur Bagautdinov, Weipeng Xu, Yuan Dong, He Wen, Jessica Hodgins, and Chenglei Wu. Modeling clothing as a separate layer for an animatable human avatar. ACM Trans. Graph.*, 40(6), December 2021.*

[168] Yuliang Xiu, Jinlong Yang, Xu Cao, Dimitrios Tzionas, and Michael J. Black. Econ: Explicit clothed humans optimized via normal integration, 2023.

[169] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J. Black. Icon: Implicit clothed humans obtained from normals, 2022.

[170] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T. Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Ghum & ghuml: Generative 3d human shape and articulated pose models. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6183–6192, 2020.*

[171] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields, 2023.

[172] Zhen Xu, Sida Peng, Chen Geng, Linzhan Mou, Zihan Yan, Jiaming Sun, Hujun Bao, and Xiaowei Zhou. Relightable and animatable neural avatar from sparse-view video. In CVPR*, 2024.*

[173] Yuxuan Xue, Xianghui Xie, Riccardo Marin, and Gerard Pons-Moll. Human 3diffusion: Realistic avatar creation via explicit 3d consistent diffusion models, 2024.

[174] Kota Yamaguchi, M. Hadi Kiapour, and Tamara L. Berg. Paper doll parsing: Retrieving similar styles to parse clothing items. In 2013 IEEE International Conference on Computer Vision*, pages 3519–3526, 2013.*

[175] Han Yang, Ruimao Zhang, Xiaobao Guo, Wei Liu, Wangmeng Zuo, and Ping Luo. Towards photo-realistic virtual try-on by adaptively generating-preserving image content. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.*

[176] Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K. Wong. Ps-nerf: Neural inverse rendering for multi-view photometric stereo. In European Conference on Computer Vision (ECCV)*, 2022.*

[177] Xihe Yang, Xingyu Chen, Daiheng Gao, Shaohui Wang, Xiaoguang Han, and Baoyuan Wang. Have-fun: Human avatar reconstruction from few-shot unconstrained images, 2024.

[178] Donggeun Yoo, Namil Kim, Sunggyun Park, Anthony S. Paek, and In-So Kweon. Pixel-level domain transfer. In European Conference on Computer Vision*, 2016.*

[179] Jae Shin Yoon, Kihwan Kim, Jan Kautz, and Hyun Soo Park. Neural 3d clothes retargeting from a single image. arXiv preprint arXiv:2102.00062*, 2021.*

[180] Hao Yu, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, and Slobodan Ilic. Rotation-invariant transformer for point cloud matching, 2024.

[181] Ruiyun Yu, Xiaoqi Wang, and Xiaohui Xie. Vtnfp: An image-based virtual try-on network with body and clothing feature preservation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.*

[182] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR2021)*, June 2021.*

[183] Zhengming Yu, Wei Cheng, Xian Liu, Wayne Wu, and Kwan-Yee Lin. Monohuman: Animatable human neural field from monocular video, 2023.

[184] Hongwen Zhang, Yating Tian, Xinchi Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop, 2021.

[185] Xujie Zhang, Ente Lin, Xiu Li, Yuxuan Luo, Michael Kampffmeyer, Xin Dong, and Xiaodan Liang. Mmtryon: Multi-modal multi-reference control for high-quality fashion generation. arXiv preprint arXiv:2405.00448*, 2024.*

[186] Zechuan Zhang, Li Sun, Zongxin Yang, Ling Chen, and Yi Yang. Global-correlated 3d-decoupling transformer for clothed avatar reconstruction, 2023.

[187] Zechuan Zhang, Zongxin Yang, and Yi Yang. Sifu: Side-view conditioned implicit function for real-world usable clothed human reconstruction, 2024.

[188] Fuwei Zhao, Zhenyu Xie, Michael Kampffmeyer, Haoye Dong, Songfang Han, Tianxiang Zheng, Tao Zhang, and Xiaodan Liang. M3d-vton: A monocular-to-3d virtual try-on network. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13239–13249, October 2021.*

[189] Fuwei Zhao, Zhenyu Xie, Michael Kampffmeyer, Haoye Dong, Songfang Han, Tianxiang Zheng, Tao Zhang, and Xiaodan Liang. M3d-vton: A monocular-to-3d virtual try-on network. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13239–13249, October 2021.*

[190] Haoyu Zhao, Chen Yang, Hao Wang, Xingyue Zhao, and Wei Shen. Sg-gs: Photo-realistic animatable human avatars with semantically-guided gaussian splatting, 2024.

[191] Zerong Zheng, Han Huang, Tao Yu, Hongwen Zhang, Yandong Guo, and Yebin Liu. Structured local radiance fields for human avatar modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.*

[192] Zerong Zheng, Tao Yu, Yebin Liu, and Qionghai Dai. Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction, 2020.

[193] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. Deephuman: 3d human reconstruction from a single image, 2019.

[194] Zerong Zheng, Xiaochen Zhao, Hongwen Zhang, Boning Liu, and Yebin Liu. Avatarrex: Real-time expressive full-body avatars. ACM Transactions on Graphics (TOG)*, 42(4), 2023.*

[195] Heming Zhu, Lingteng Qiu, Yuda Qiu, and Xiaoguang Han. Registering explicit to implicit: Towards high-fidelity garment mesh reconstruction from single images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3845–3854, June 2022.*

[196] Luyang Zhu, Dawei Yang, Tyler Zhu, Fitsum Reda, William Chan, Chitwan Saharia, Mohammad Norouzi, and Ira Kemelmacher-Shlizerman. Tryondiffusion: A tale of two unets. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4606–4615, June 2023.*

[197] Shizhan Zhu, Sanja Fidler, Raquel Urtasun, Dahua Lin, and Change Loy Chen. Be your own prada: Fashion synthesis with structural coherence. In Proceedings of the IEEE Conference on International Conference on Computer Vision*, 2017.*

[198] Wojciech Zielonka, Timur Bagautdinov, Shunsuke Saito, Michael Zollhöfer, Justus Thies, and Javier Romero. Drivable 3d gaussian avatars. 2023.

[199] Xingxing Zou, Xintong Han, and Waikeung Wong. Cloth4d: A dataset for clothed human reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12847–12857, 2023.*

[200] Silvia Zuffi and Michael J. Black. The stitched puppet: A graphical model of 3d human shape and pose. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.*

[201] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Ewa volume splatting. In Proceedings Visualization, 2001. VIS '01.*, pages 29–538, 2001.*