3D Shape Analysis: Reconstruction and Classification

Thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Engineering

by

Jinka Sai Sagar
201550857
jinka.sagar@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
May 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled " 3D Shape Analysis: Reconstruction and Classification " by Jinka Sai Sagar, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Dr. Avinash Sharma

To my family and Lord Ayyappa...

Acknowledgments

The work in this thesis has been made possible by my family, my colleagues, friends and mainly my advisor Dr. Avinash Sharma. I have met several people in this professional journey who have left an everlasting impact on me.

I would like to start this section by thanking my family. I remember in 2015, when I initially shared my desire to do PhD with my parents and family. While some of them suggested not to opt as I was earning handsomely, my parents asked me if I would be happy doing PhD. I said yes. The support my parents gave me from that day to till date cannot be described in words. The discipline they taught me always made me a better person. I have two wonderful younger siblings who always cheer for my achievements. Me and my brother share a unique bond and I am sure no brothers in our family shares similar bond as ours :). My sister is like an elder daughter to me and teasing her is sometimes a bigger relief than anything else. I would like to say that I am there for both of you in every circumstances in your life. Next, I would like to thank my wife who has been a constant pillar of my support and strength. PhD is a long journey where I fought with many challenges, depression, problems etc. She is there in every struggle of mine and helped me to face all these demons successfully. I would also like to thank my mother-in-law and brother-in-law for their constant support. Last but not least, my daughter Indrakshi and son Ved Rudhransh are wonderful kids. Holding them in my arms for the first time are easily the most celebrated moments in my life. Playing with them is the ultimate stress buster that I can ask for. This thesis is dedicated to all of my family members.

I would like to thank my advisor Dr. Avinash Sharma for all his efforts which made me a person who I am today. He has provided me immense support and confidence during my PhD journey which eased my path in doing research. He really understands students personally and keeps in all efforts to address their concerns. Sir, I am sure there are not many professors like you who backs their students as you do. His philosophical talks are very informative and he has been a mentor to me in all sense who always gives examples on how to lead life during these talks. On a lighter note, writing paper with him is really fun and tiring :). Thanks sir for making me a better researcher and person. I am also privileged to work with Prof. P.J.Narayanan who inspired me further to pursue better research.

Without my wonderful colleagues, it is really difficult to carry the work forward. The canteen talks with these people proved effective than reading many papers. I would like to thank Astitva Srivastava, Chandradeep, Rohan Chacko, Snehith, Abbhinav Venkat and Snehith for being co-authors in my

research works. You guys are awesome. I would also like to thank Saurabh Saini sir who helped as mentor for me in many of my academic problems.

A friend in need is friend indeed. The final part of this section is dedicated to all my friends without whom this journey would not be completed. To my friends Bhanu Prakash, Sreeram, Anudeep and Yashwanth, you guys deserve a special mention. They helped me to cheer up in one of the toughest situations in my life. My stay with them in Green Valley apartment is the most memorable and joyous period. They keep motivating me every time. I would also like to thank my cousins and friends Girish and Guru for having good times in Hyderabad. To my cheddi buddies Akhilesh, Viswateja and Rahamthulla, cheers to these guys. I always meet them when I went to home. They always bring back smiles and awesome memories we had. I always look forward to meet all these guys. To a lifetime of happiness and fun together.

Abstract

The reconstruction and analysis of 3D objects by computational systems has been an intensive and long-lasting research problem in the graphics and computer vision scientific communities. Traditional acquisition systems are largely restricted to studio environment setup which requires multiple synchronized and calibrated cameras. With the advent of active depth sensors like time-of-flight sensors, structured lighting sensors made 3D acquisition feasible. This advancement of technology has paved way to many research problems like 3D object localization, recognition, classification, reconstruction which demand innovating sophisticated/elegant solutions to match their ever growing applications. 3D human body reconstruction, in particular, has wider applications like virtual mirror, gait analysis, etc. Lately, with the advent of deep learning, 3D reconstruction from monocular images garnered significant interest among the research community as it can be applied to in-the-wild settings.

Initially we started exploration of classification of 3D rigid objects due to availabilty of ShapeNet datasets. In this thesis, we propose an efficient characterization of 3D rigid objects which take local geometry features into consideration while constructing global features in the deep learning setup. We introduce learnable B-Spline surfaces in order to sense complex geometrical structures (large curvature variations). The locations of these surfaces are initialized over the voxel space and are learned during training phase leading to efficient classification performance. Later on, we primarily focus on rather challenging problem of non-rigid 3D human body reconstruction from monocular images. In this context, this thesis presents three principle approaches to address 3D reconstruction problem. Firstly, we propose a disentangled solution where we recover shape and texture of the 3D shape predicted using two different networks. We recover the volumetric shape of non-rigid human body shapes given a single view RGB image followed by orthographic texture view synthesis using the respective depth projection of the reconstructed (volumetric) shape and input RGB image.

Secondly, we propose PeeledHuman - a novel shape representation of the human body that is robust to self-occlusions. PeeledHuman encodes the human body as a set of Peeled Depth and RGB maps in 2D, obtained by performing ray-tracing on the 3D body model and extending each ray beyond its first intersection. We learn these Peeled maps in an end-to-end generative adversarial fashion using our novel framework - PeelGAN. The PeelGAN enables us to predict shape and color of the 3D human in an end-to-end fashion at significantly low inference rates.

Finally, we further improve PeelGAN by introducing a shape prior while reconstructing from monocular images. We propose a sparse and efficient fusion strategy to combine parametric body prior with

a non-parametric PeeledHuman representation. The parametric body prior enforces geometrical consistency on the body shape and pose, while the non-parametric representation models loose clothing and handles self-occlusions as well. We also leverage the sparseness of the non-parametric representation for faster training of our network while using losses on 2D maps.

We evaluate our proposed methods extensively on a number of datasets. In this thesis, we also introduce 3DHumans dataset, which is a 3D life-like dataset of human body scans with rich geometrical and textural details. We cover a wide variety of clothing styles ranging from loose robed clothing like saree to relatively tight-fitting shirt and trousers. The dataset consists of around 150 male and 50 unique female subjects. Total male scans are about 180 and female scans are around 70. In terms of regional diversity, for the first time, we capture body shape, appearance and clothing styles for the South-Asian population. This dataset will be released for research purposes.

Contents

List of Figures

Chapter 1

Introduction

3D computer vision is an interesting and an active research domain which has wider applications spanning across various domains like robotics, augmented/mixed reality, entertainment etc. Traditionally, 3D data has largely been restricted to CAD models created by artists in [1] AutoCAD like softwares. However, the recent technical advancement in last decade has enabled us to efficiently capture, process and visualize 3D data at large scale. This advancement made 3D data more accessible and cheaper. Many of the mobile phones available now-a-days has depth sensor installed enabling 3DFace unlock using depth features. Recently, iPad is equipped with LiDAR sensor to scan large scale objects like trees, buildings, caves [2] etc,. Additionally, the cameras are able to click pictures with very high resolution (upto 100 MegaPixels). This will lead to exponential increase in availability of high quality 3D data in the near future [117]. Thus, the emergence of 3D data generation capabilities along with computational advancement in terms of GPUs, research problems like 3D object localization, recognition, classification, reconstruction demand innovating sophisticated/elegant solutions to match their ever growing applications. 3D data can be broadly classified into three classes of objects namely rigid (which undergo rigid transformation e.g. chair, sofa etc.), non-rigid (undergoes non-rigid deformation e.g. human bodies) and elastic (exhibits elastic deformation e.g. rubber). The representation of 3D data (outlined in section 2.1) is a crucial choice which largely determines the efficiency of proposed algorithms to analyze the data.

In this thesis, we initially attempted to explore classification of rigid objects and then primarily focused on reconstruction of non-rigid shapes, in particular human bodies with loose clothing. Recovering shape, pose and motion of clothed human body from the commercially available depth and RGB sensors can play a crucial role in analysing their interaction with environment. Such understanding may open doors to wide variety of applications which may require answers to questions like ”How does the person look in other dressing outfits?”, ”How does the person look from other side?”, ”What is the pose of the person in 3D?”, etc. The problem related to 3D human body estimation has a long history in computer vision community. The accurate estimation of 3D human body reconstruction is largely

[1] http://www.autodesk.in

[2] https://www.tetongravity.com/story/gear-tech/digitalisation-of-caves-proceeds-using-lidar-on-new-iphones

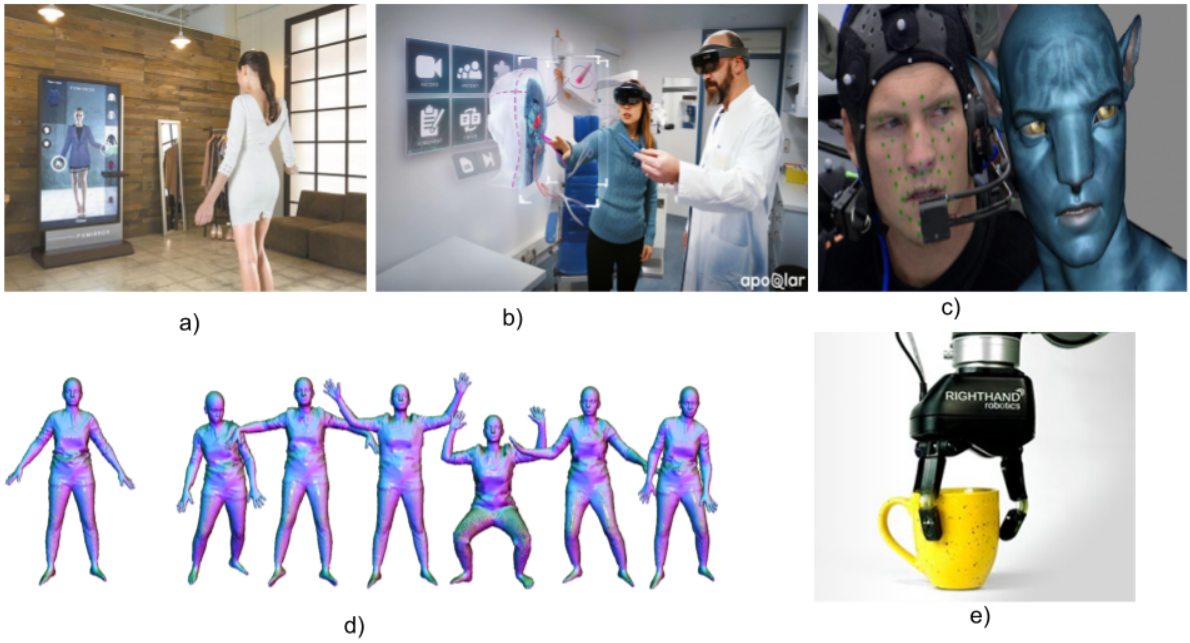

Figure 1.1 Applications of 3D shape analysis. (a) Virtual Try-on (adopted from), (b) Virtual reality for training health-care professionals (adopted from), (c) Motion transfer adopted from, (d) Animatable human bodies [93] and (e) Robot-object interaction

limited to expensive capture studio involving multiple synchronized calibrated cameras. However, real world scenario is quite different and pose many challenges. We cannot expect multiple images of a same person at the same time stamp to make 3D reconstruction plausible. Hence, estimating 3D humans from monocular image, which is an ill-posed problem, has garnered wider interest in the research community. However, the estimation/hallucination of humans from different viewpoint as compared to input image is a mammoth task for computers while humans can easily hallucinate the occluded regions. With the advent of deep learning, estimating 3D human body from monocular images became feasible. Nevertheless, many of the solutions still far from obtaining a credible reconstruction. Moreover, these solutions are expensive in time. This can be attributed to the underlying representation of 3D data which greatly affects the performance. The goal of this thesis is to design and propose efficient representation and algorithms to make computer understand 3D data, specifically human body.

1.1 Motivation

Enabling visual perception for computers a.k.a. computer vision had started in late 1960s with a small scale project aiming to extract shapes of 3D objects from 2D images [119]. The invention of commodity camera in late 1980s facilitated the research. However, with very limited computational

power then, research was largely limited to processing images. With the advent of GPUs, the computer vision research again started to focus on its original objective, i.e., understand 3D world from images.

Understanding 3D world around us enables us to interact and visualize the world from different viewpoints. The pivotal task for understanding 3D world is to analyze its constituent 3D elements. These elements vary from rigid-objects to non-rigid humans. Image is a powerful tool available with us which captures a moment of time of the 3D world. The goals of 3D modeling and reconstruction are to replicate the 3D world around us as faithfully as possible from the images. The field has made rapid progress in the last decade due to advancements in robust algorithms for depth estimation and the availability of consumer hardware for 3D sensing and capture. Successful reconstruction of 3D objects paves way to many interesting applications in 3D computer vision. Some of them include understanding 3D rigid objects enables robot to interact with these elements, virtual reality can be a new norm for training health care professionals, motion transfer which is widely used in entertainment industry, animating reconstructed human body and virtual try-on which is widely used in fashion industry, to name a few, as shown in Figure 1.1.

There are many research problems to probe in order to understand humans interaction with 3D world. These problems include human pose estimation in 3D, clothed 3D body reconstruction from images, temporal reconstruction of human in action (4D video), etc. Reconstruction of 3D human bodies is the primary problem of interest in this thesis as it provides accurate estimates of shape and pose.

Over the decades, the research problem of reconstructing humans from images has been solved incrementally with various computer vision techniques. Traditionally, active 3D sensors are employed to capture the shape of human bodies. These scanners emit radiation i.e. laser rays, structured light and collect the depth information of the surfaces. The complete shape can be captured by fusing multiple partial scans of the body which poses limitation that it is hard to reconstruct the shape which is deforming temporally. Also, the cost of these sensors is typically high. To address these limitations, interest had garnered around computer vision algorithms to infer object shape. Images captured from multiple synchronized calibrated cameras are used to maximize the 3D volume of the object with the constraint which enforces the projection of this 3D volume to each camera view completely falls inside the silhouette of that view. These methods are named Shape-from-X [57, 114] where X can be silhouettes, photometric stereo, depth, etc. However, these methods need studio setup with green screen mat. Additionally, multiple cameras (typically 200) are to be calibrated and synchronized. These requirements make the capture process expensive and not scalable to real-world scenarios.

Although severely ill-posed, performing monocular reconstruction makes it non-intrusive, affordable, and scalable to real-world environments. Computer-vision based methods may not produce reconstruction from monocular images as the data is not available from other views. Recent advancement in Machine Learning has helped to develop algorithms in deep learning networks which felicitated the research along this direction. In order to generalize these models on real-world images, we need to train the networks/models with appropriate high quality 3D data similar to that of real humans. Additionally, the performance of these deep models are determined by the representation of 3D data that we will be

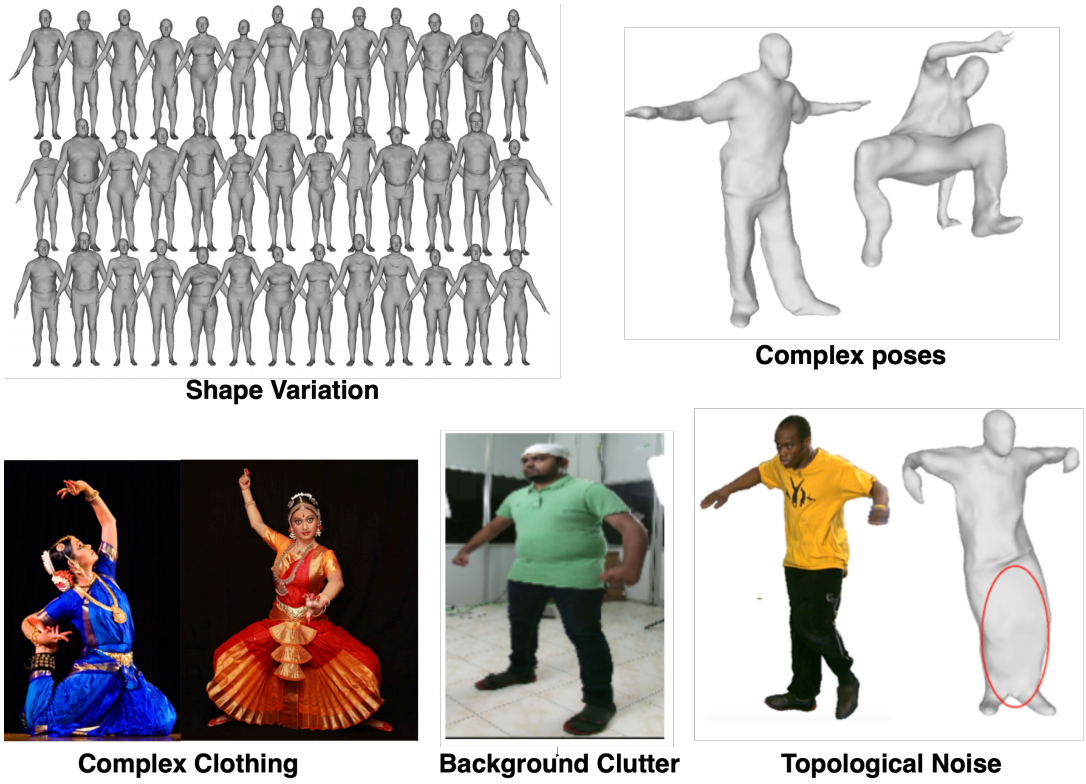

Figure 1.2 Some of the challenges include complex clothing (link), topological noise, wide variety of articulated poses, background clutter, variation of shapes [6].

using. Hence, keeping all the elements together, we aim to represent and reconstruct accurate estimates 3D of human body from monocular images using deep learning models effectively.

1.2 3D Human body reconstruction: challenges

An image is a projection of the 3D world to a 2D image plane, during which the depth dimension is not retained. Specifically, the imaging of 3D humans onto 2D image is the result of blended effects of various factors and the inverse mapping i.e., 3D reconstruction of human body models from image is a very challenging task as we need to unfold the blended effects. This requires the proposed model to implicitly or explicitly understand what the shape and appearance of a typical human is, which helps the model to hallucinate unseen parts. Moreover, the proposed model needs to deal with challenges that are ambient in monocular images.

These challenges include:

- Humans exhibit significant diversity in body shape variation. Any solution to recover the 3D human body must be robust to change in shape and gender. An illustration is shown in Figure 1.2.

- Human body is non-rigid objects can exhibit a large number of complex articulated body poses, with significant subjective pose variation across individuals. Some of the poses can be as complex as hand stand.
- Clothing partially or fully occludes the underlying body resulting in uncertainty of shape and pose as shown in Figure 1.2. Moreover, the cloth can also undergo through complex surface deformations. In general, there are many varieties of clothes ranging from tight clothing (wore by athletes) to very loose clothing (wore by artists).
- Human body undergoes non-rigid deformation and geometry can also evolve over time while interacting with objects (yielding a large space of complex body poses as well as shape variations owing to soft-tissue deformations).
- Self occlusions are natural to human bodies due to articulated poses where body parts like limbs occlude the other body parts which severely affects both pose and shape estimation.
- In-the-wild images are captured with complex environments with varied backgrounds. Extracting 3D human body in these settings is a challenging task.
- Human body has specific body structure consisting of two hands, legs etc. Monocular image can sometimes be misleading! When there is no sufficient information to distinguish between body & cloth, reconstructions might result in inaccurate estimation of topology of the body parts as visualized in Figure 1.2.
- Estimation of shape and pose from images captured at various skewed camera viewpoints which results in occluded body parts. Moreover, solutions to the problem should ensure the consistency across all multiple views.
- Severe change in illumination can cause the change in the appearance of the underlying human body while imaging.
- The availability of 3D life-like datasets is crucial to train deep learning models. Lack of real-world life-like 3D scans of humans inhibits the accuracy of models while deploying on in-the-wild images.

1.3 Research landscape

The problem of understanding 3D human body has manifested into the four sub problems such as, (i)Marker Based capture, (ii) Marker-less Multi-view 3D capture, (iii) Reconstruction from sensors and (iv) Monocular 3D human body reconstruction. We review the prior art of these sub-problems in this section.

Figure 1.3 Drawbacks of (a) marker-based capture and (b)multi-view marker-less capture. Image adopted from [34].

Marker based capture: These methods reconstruct humans by placing markers or sensors [27, 52, 106, 107, 89] on the body and has been one of the most common ways to estimate 3D skeletal motion. The visible 3D markers on the body surface are used to infer the shape and articulated pose from a proxy-3D human body model. The main drawbacks of these techniques is that the actor needs to wear markers and with minimal cloth as shown in Figure 1.3. Hence, to address these issues research has progressed towards marker-less capture.

Marker-less multi-view Capture: This class of 3D capture employ traditional computer vision algorithms to extract 3D human body from images. Multiple images are needed to extract 3D using these algorithms because of self-occlusions caused by body parts and clothing deformations. Early attempts along this direction employ visual hull based methods [30, 39, 40, 97, 153] due to its efficiency and robustness to approximate underlying 3D geometry. Basic visual hull algorithm can be outlined as: (i) Each input image is segmented to obtain a silhouette of an object. (ii) using known camera parameters, a silhouette of an object is projected to 3D space thereby creating a visual cone. (iii) a visual hull is obtained as the intersection of all the visual cones generated for different points of view. The advantage of this method is that it requires neither constant object appearance nor the presence of textured regions. However, visual hull based algorithms cannot handle concave regions nor produce fine-scale details especially when the number of input views is limited.

To achieve fine-scale details in the recovered geometry we need to consider RGB image features. Multi-view stereo methods [125, 135, 156, 178] consider these RGB features to produce fine-scale geometrical and textural details. On similar lines, temporal reconstruction of human bodies in action i.e. human performance capture is also attempted using multi-view images. Extremely high-quality reconstruction results have also been demonstrated with tens or even hundreds of cameras [34]. However, these systems are expensive and expect capture studio and some times green screen background as

shown in Figure 1.3(b). With the advent of Microsoft Kinect RGBD sensors, a low cost and consumer friendly 3D reconstruction of human body is plausible. 3D human reconstruction from consumer multiple Kinect sensors is initially proposed in [139]. The algorithm is as follows, initially each pair of raw scans are registered pairwise. After pairwise registration, the first and last frame does not well match as error accumulates. Global deformation registration is used to deal with these problems. Despite advances in technology, multi-view methods often produce noise in the final reconstructions. Nevertheless, we need multiple views in a studio setup and these limitations serve as a bottleneck for 3D acquisition in in-the-wild setup.

Reconstruction from sensors: Range or depth maps give estimates of distance measurements from a known reference coordinate system to surface points on the object to be reconstructed. In many real-world scenarios, high quality 3D models of a static human body are obtained using commercial scanners (range systems), that are generally expensive, but accurate. Other common sensors include LIDAR and structured light sensors Figure 1.3. LIDAR is typically used for capturing 3D scans of outdoor scenes. Structured light 3D scanner like in is the ideal choice for making a quick, textured and accurate 3D model of medium sized objects such as a human body, an alloy wheel, or a motorcycle exhaust system. It scans quickly, capturing precise measurements in high resolution. However, these scanners are expensive and thus inhibits 3D human body reconstruction in-the-wild images.

Figure 1.4 a. Sample LIDAR scan of a house, Image adopted from cite b. Structured light Artec Eva sensor and a sample scan

Monocular 3D human body reconstruction: One potential remedy to the aforementioned problems is to estimate 3D from monocular images captured from mobile phones. The solution to this problem can make 3D human body reconstruction makes a wide reach. Recent advancements in deep learning, which has capability to halucinate, accelerated the research along this direction. Deep learning based

solutions from monocular image-based 3D reconstruction can be broadly classified into (i) Model-based approaches and (ii) Model-free approaches. Below we review these two primary paradigms:

Model-based Approaches: 3D body estimation from monocular images has attracted substantial interest because of emergence of statistical body models like SCAPE[6]. One of the early works in this direction is optimization-based approach [46] where SCAPE model parameters are inferred from monocular images. Other methods which estimate SCAPE parameters are [49, 28]. SCAPE model is triangle-based defomation which poses artifacts and is not compatible with rendering engines. Hence, SMPL [90] model is proposed which is vertex-based deformation parametric model. SMPL is discussed in detail in section 2.4. SMPLify [17] the first method to automatically estimate the 3D pose of the human body as well as its 3D shape from a single unconstrained image. The objective function is formulated to optimize pose and shape directly so that the projected joints of the 3D model are close to the 2D joints estimated by the CNN. Several approaches have been proposed to infer the shape and pose parameters from the input images using CNNs. One of the predominant works along this direction is [63] which proposes to regress SMPL shape and pose parameters from the input image directly by minimizing the re-projection loss of keypoints, which allows the model to be trained using in-the-wild images that only have ground truth 2D annotations. Since only re-projection loss is under constrained, a discriminator is trained to classify if the parameters are real/fake. Other cues like segmentation are used to refine the parameters by various works [75, 74, 104, 85, 165]. One of major drawbacks of model-based approaches is that the mesh obtained from parametric methods is smooth and naked. Although we can extract shape and pose accurately, we miss out the person-specific details like hair, clothes etc. To address these issues, some approaches [1, 14, 108, 4, 77] estimate offsets on these SMPL vertices. Nevertheless, the offset estimation can handle only relatively tight clothing scenarios and fail on reconstruction loose clothing.

Another section of works deform a scanned template model of the actor in canonical pose to fit to input monocular RGB videos. These works aim marker-less approach for temporally coherent 3D performance capture of a human with general clothing. Optimization based estimation is proposed in [164]. However, this approach requires expensive computation making hard to estimate in real-time. [47] proposes an efficient solution which can be computed in real-time. Multi-view based self-supervision while predicting skeletal pose and non-rigid deformation proposed in [48] enabled better reconstruction accuracy. Physical constraints have been employed in [163, 129] while recovering motion from monocular videos.

Model-free Approaches: These kind of approaches pose no body model constraints. Hence, we can recover loose and arbitrary clothing using model-free approaches. The underlying representation is a crucial choice in designing the neural network for these approaches. As discussed in section 2.1, 3D data volumetric, implicit, point cloud and mesh representations. Early efforts along this direction proposed to use volumetric convolution [148, 143] as it is direct extension to Euclidean convolution proposed on pixels in 2D image. However, volumetric approaches poses a serious computational disadvantage as

there is redundancy/wastage in computation of empty voxels outside the body surface or inside the body surface. Moreover, these approaches are limited by voxel resolution and texture cannot be recovered in end-to-end fashion. Another interesting deep learning solution [102] uses silhouettes as cue, however, it requires multiple images.

Recently,implicit function approaches has gained importance as neural network can used as classifier which can represent arbitrary functions. In particular, the recent implicit function learning models, PIFu [122] and PIFuHD [123] estimate voxel occupancy by utilizing pixel-aligned RGB image features computed by projecting 3D points onto the input image. When multiple view images are available, these methods refine the reconstruction by fusing features from these views. However, the pixel-aligned features suffer from depth ambiguity as multiple 3D points are projected to the same pixel. 3D human body reconstruction has also been attempted in the same vein by predicting front and back depth maps in [42, 134]. However, they fail to handle self-occlusions by body parts. Nevertheless, these approaches do not seek to enforce global consistency on the body shape/pose to encourage physically plausible shapes and poses of human body parts. Introducing shape prior while reconstructing the shapes using the aforementioned non-parametric approaches addresses the issue as proposed in [55, 51, 173, 174].

1.4 Goal of this thesis

This thesis addresses the problem of 3D human body reconstruction using monocular images. In this context, we propose various solutions for associated sub-problems, and show ways of addressing the problem. The major goals of this thesis are two-fold, (i) design effective representation/encoding for 3D data, especially human body. The state-of-the-art 3D human body reconstruction paradigms suffer from computational disadvantages. Our thesis proposes a new representation which is fast, sparse and robust encoding of 3D human body shapes, (ii) deep learning solutions for effectively recovering 3D human body from monocular images.

Humans, in general, wear wide variety of clothes ranging from tight clothing to very loose clothing like sarees. Many of the SOTA methods fail to perform on very loose clothing. Our thesis aims to propose reconstruction approaches which can handle any types of clothing. Also, texture is another important aspect in 3D human body reconstruction. Many applications in current scenarios, rely on plausible texture. However, there are hardly few works which recover shape and texture of 3D body in an end-to-end fashion. In this thesis, we aim to recover texture and shape using a single deep learning network. Moreover, since this area is recently grown, many of the available datasets for the problem are either synthetic or not very challenging for real scenario, i.e. they have relatively tight clothing. Hence, as part of this thesis, we also introduce and benchmark 3D humans dataset.

In addition to 3D human body, this thesis aims at proposing efficient network for 3D shape analysis, in particular classification, of rigid objects.

1.5 Contributions of this thesis:

1. **Efficient 3D shape classification [60]:** Majority of the existed deep learning networks for 3D shape classification [161] are proposed over volumetric convolution. We propose a novel, fast and robust characterization of 3D shapes that accounts for local geometric variations as well as global structure. We built up on the learning scheme of [83] by introducing sets of B-spline surfaces whose positions are learnable. The input to this network is 3D shapes represented in distance transform which is more efficient than volumetric convolution.

2. **Volumetric reconstruction [148]:** We exploit volumetric representation of 3D bodies and propose a novel deep learning module. Using this module we recover shape of the body and texture is reconstructed using a separate novel network. To our knowledge, our proposed textured reconstruction of 3D human body models is the first solution along this direction. Our method generalizes to arbitrary topology of 3D shapes and textures.

3. **PeeledHuman representation [59]:** Existing representation of 3D data poses disadvantages in terms of computation and ability to represent textures. In this thesis, we present PeeledHuman, a novel representation for encoding 3D human body shapes. PeeledHuman encodes the human body as a set of Peeled Depth and RGB maps in 2D, obtained by performing ray-tracing on the 3D body model and extending each ray beyond its first intersection. This formulation allows us to handle self-occlusions efficiently compared to other representations.

4. **Generative model for shape reconstruction (PeelGAN) [59]:** Given a monocular RGB image, we learn these Peeled maps in an end-to-end generative adversarial fashion using our novel framework - PeelGAN. We train PeelGAN using a 3D Chamfer loss and other 2D losses to generate multiple depth values per-pixel and a corresponding RGB field per-vertex in a dual-branch setup. Unlike other SOTA methods, PeelGAN predicts shape and colors in a single forward pass.

5. **SHARP: Shape-Aware Reconstruction of People in Loose Clothing [61]:** We further attempted to improve PeelGAN by introducing body shape prior in the form of SMPL. To this end, we propose SHARP which uses a sparse and efficient fusion strategy to combine parametric body prior (SMPL) with a non-parametric 2D representation (PeeledHuman) of clothed humans. The parametric body prior enforces geometrical consistency on the body shape and pose, while the non-parametric representation models loose clothing and handles self-occlusions as well. We train our network with only 2D losses unlike adversarial and chamfer losses used in PeelGAN making the network significantly faster to train. Our SHARP achieves significantly faster inference rate when compared to many of the SOTA methods.

6. **Dataset:** Many state-of-the-art methods for reconstructing 3D human bodies train their models on expensive commercial datasets which are not publicly available for research. These datasets have 3D human body scans which resemble real humans. This data helps the learning-based

models to generalize well on unseen real-world scenarios. Existing datasets [14, 174, 11, 138] are either synthetic or have minimal clothing. To bridge these gaps, we collected **3DHumans**, a dataset of 3D human body scans with wide variety of clothing styles and varied poses. We are able to retain high frequency geometrical and texture details using a commercial structured light sensor (accurate up to 0.5mm). We also collected a real dataset of textured 3D human body models in action and their corresponding multi-view RGBD using commercially available Kinect V2 sensors.

1.6 Thesis roadmap

In chapter 2, we provide the necessary background for this thesis and briefly summarize the aspects of various way to represent 3D data, ways to capture image from 3D data i.e. ray tracing and rasterization followed by camera projection models. We also discuss about parametric human SMPL model which is directly relevant to the work that follows.

In chapter 3, we address the problem of rigid 3D shape classification using deep learning network we proposed. In this chapter, we show results on ModelNet dataset and compare our performance of our method with previously published methods.

Remaining chapters focus on challenging non-rigid shape i.e. human body reconstruction from monocular images. Chapter 4 proposes the disentangled solution where shape and texture are recovered with separate networks. In this chapter, we evaluate our method on MPI, MIT's articulated mesh animation datasets. We also perform a rigorous analysis of all steps in our approach and analyse the results and show superior performance to SOTA methods then.

In chapter 5, we present PeeledHuman, a novel representation for encoding 3D human body shapes. We also propose a PeelGAN - a novel deep learning network to predict the representation from monocular image. We train our method with MonoPerfCap dataset and evaluate on BUFF method where we outperform existing methods. In chapter 6, we improve the PeelGAN by including body shape prior while reconstructing the 3D shape. In this work, we work on CLOTH3D, THUman and our datasets. In chapter 7, we propose state-of-the art 3D human body data with very high resolution details. We also present multi-camera calibrated dataset. Finally, chapter 8 provides the summary of our work, impact of this thesis. Here, we also discuss the future directions that span out of this thesis.

Chapter 2

Background

In this chapter, we initially discuss different representations of 3D data. We later outline image rendering, different camera models we use in our work and followed by SMPL, a parametric model for human body shapes. Many of the methods in this thesis are inspired by and developed over these background concepts.

2.1 3D Representations

2.1.1 Polygonal meshes

Polygonal meshes are the widely used geometric representation used in computer graphics. These meshes are built upon basic primitives called polygon or face. A face can have minimum of 3 vertices which is then a *triangular mesh*. A face with four vertices is called a quad. .obj is one of the common file formats used to store polygonal meshes. 3D objects represented as polygonal meshes in .obj format stores different types of elements:

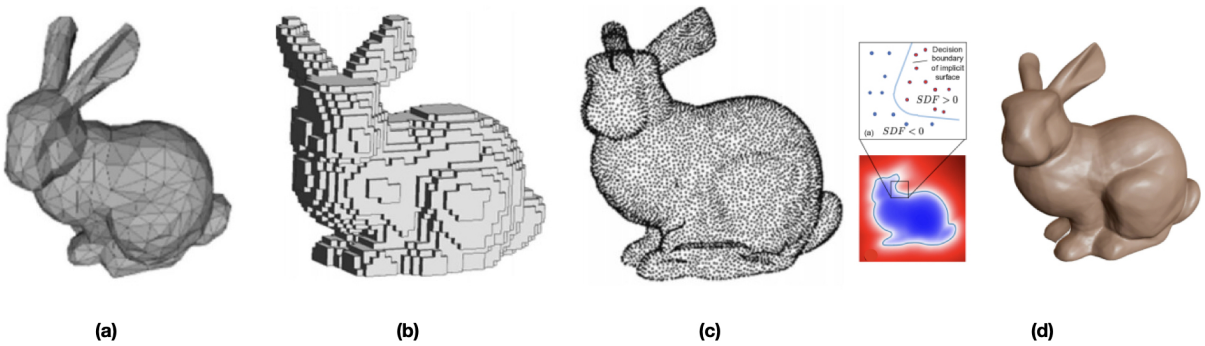

Figure 2.1 (a) Polygonal Mesh, (b) Voxel/volumetric, (c) Point cloud and (d) Implicit representations, figure adopted from [105].

- Vertex: x,y,z coordinates of the point. We can also vertex color of this point. Each vertex in .obj file starts with "v" e.g. v -0.500000 -0.500000 0.500000 255 255 255)
- Vertex Normals: The normal (n_x, n_y, n_z) are represented as vn 0.000000 0.000000 1.000000.
- Texture coordinates: Each vertex can be assigned a texture coordinate for texture mapping. Each coordinate is two dimensional and is represented by vt 0.375000 0.000000.
- Face: A sample face of triangular mesh is represented by "f 7/13/21 1/1/22 3/3/23". f defines a face declaration which is followed by 3 groups of three numbers separated by a '/'. The first number defines the index of the vertex in the vertex array (the v's). The second number defines the index of the vertex texture coordinates in the texture coordinates array (the vt's). The third number defines the index of the vertex normal in the normal array (the vn's). In the OBJ file format, arrays are 1-based (the first element in the array has index 1)

The usage of meshes has become more intensive as modern computers are optimized to handle polygons. Unlike other representations, meshes are easier to visualize, import into many of the existing graphics tools, render images from the scene. Since meshes are created from primitives as discussed earlier, they are easy to animate and deform. On the flip side, mesh is limited by its resolution i.e. with less number of vertices, it becomes extremely difficult to represent high frequency details like wrinkles in the cloth.

2.1.2 Voxels/Volumetric representation

Voxels are analogous to pixel in images. Pixel can be defined as sample of an original image and more samples provide a better estimate of the original image. A voxel is similarly a regular grid in a 3 dimensional space. Unlike polygon meshes, which represent the surface as polygons, volumetric/voxel meshes discretize the interior structure of the object which can be represented as:

$$V(x, y, z) = \begin{cases} 1, & \text{if inside surface.} \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

where x, y, z represents the 3D location of the voxel. Voxels are described by the resolution of this 3D grid. Higher the resolution better the representation. However, higher resolution occupies significantly large memory. Additionally, the current computer hardware is highly optimized for rendering polygonal meshes, and we don't have specialized hardware to efficiently render high-resolution voxels. Initial deep learning methods on 3D shape analysis predominantly used voxel representation as we can directly extend 2D euclidean convolution to 3D voxels in trivial manner.

2.1.3 Point cloud

A point cloud is a set of data points in space. It is a collection of data points defined by a given coordinates system. In a 3D coordinates system, a point cloud may define the shape of some real

or created physical system. Point clouds are used to create 3D meshes and other models used in 3D modeling. Unlike voxels and meshes point cloud is unordered i.e. these are set of points without any order. To be more specific, consider N points, any combination of N points (of total N!) results in same 3D structure. All the points in the set are from a space with distance metric (Euclidean generally). Many scanners (e.g. LIDAR) for 3D sensing outputs their sensory data in point cloud format. We can convert point cloud to mesh using Poisson Surface Reconstruction [69], delaunay triangulation etc.

2.1.4 Implicit surface

The basic concept of implicit surface representations for geometric models is to characterize the whole embedding space by classifying each 3D point to lie either inside, outside, or exactly on the surface $\mathcal{S}$ that bounds a solid object. The surface $\mathcal{S}$ is defined by the zero-level set of a scalar function F : $\mathbb{R}^3 \rightarrow [-1, 1]$. The function F takes 3D point p location as input and outputs implicit functional value at the p. The usual convention is negative values of F depicts the points inside the surface and positive points outside the surface. The zero-level set consists of all points that are on the surface. Recent deep learning based approaches based on implicit surface representation consider F : $\mathbb{R}^3 \rightarrow 0, 1$ making it a binary classification problem.

Deforming implicit surfaces can be done by decreasing (= growing) or increasing (= shrinking) the function values of F locally. Since the structure of F (e.g., the voxel grid) is independent from the topology of the level-set surface, we can easily change the surface topology and connectivity. The implicit function F for a given surface $\mathcal{S}$ is not uniquely determined since, e.g., any scalar multiple λF yields the same zero-set. Another most common implicit representation is *signed distance function(SDF)* which maps the 3D point x to its signed distance $d(x)$ from the surface $\mathcal{S}$. The absolute value $|d(x)|$ represent the distance of x and the sign of the functional value indicates whether the point is inside or outside of the 3D object.

On the flip side, generating sample points on an implicit surface, finding geodesic neighborhoods, and even just rendering the surface is relatively difficult. Implicit functions can be converted into mesh representation using marching cubes algorithm.

2.2 Image rendering

An image is a 2D dimensional representation of a 3D scene when viewed from a specific viewpoint. The process of generating image is known as image rendering and is a function of object geometry, lighting in the scene (we can't see any object if there is no light) and interactions between various objects in the scene. In computer graphics, Ray tracing and rasterization techniques are used to create images. Rasterization is a rendering approach which operates in the screen space, and achieves high frame rates on modern hardware. However, rasterization is an approximation and does not produce physically accurate images. On the other hand, ray-tracing simulate the flow of light through a scene

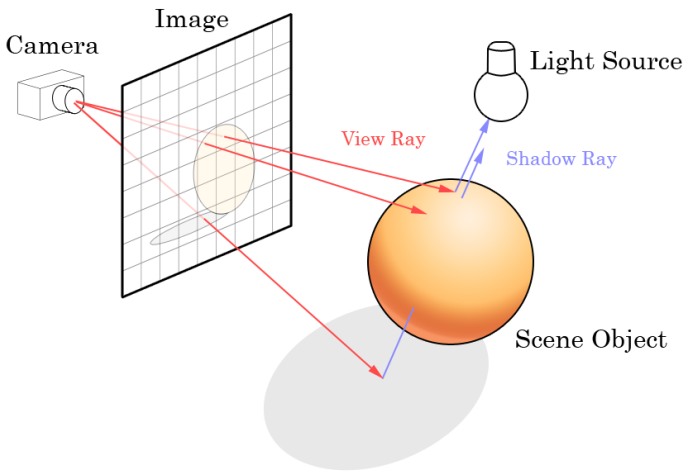

Figure 2.2 Ray tracing deals with computing ray-object and object-light interactions. Figure adopted from [91].

and produce photorealistic images of the scene. Ray-tracing produces physically accurate and photo-realistic images, and thus has been widely adopted for visual effects in movies. With the advent of GPUs like NVidia RTX and optix, ray-tracing has become a possibility.
Below we outline these two widely adopted algorithms:

2.2.1 Ray tracing

Ray tracing is all about rays. From a camera, we shoot multiple rays into the scene for every pixel of the image plane. The direction of the ray is computed by tracing the line from camera origin to pixel center. We then check which object in the scene these rays intersect. To be specific, for each ray, we compute the first intersection with the object and add the light due to external light sources at the intersected point. We also consider material properties of that point while integrating the lighting effect on the point. This process is done recursively, where at the intersection point, a secondary ray can be shot in any direction and a similar calculation can be performed at it's intersection. This recursive process implicitly takes into account *indirect* lighting effects (e.g. inter-object interactions), typically referred to as *global illumination*, a critical component for photo-realism. By setting up the pixel color with the color at the intersected point of each ray passing through the centre of each pixel, then we can form an image of the scene as seen from a particular viewpoint. We can summarize the basic implementation of ray-tracing in three steps as:

- Rays: Rays are cast from camera center to every pixel in the image plane.
- Object interactions: For each ray, test if it intersects any of the objects in the scene by looping over all the objects for each cast ray and compute the nearest point that is intersected.

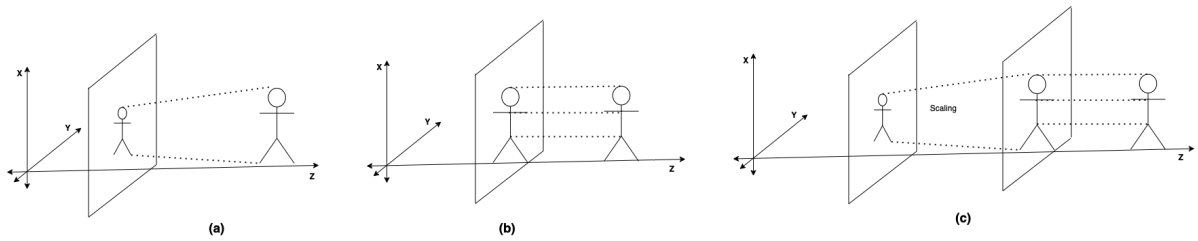

Figure 2.3 (a)Perspective projection (b) Orthographic projection and (c) Weak perspective projection.

- Shading: Compute the net effect of all lighting and inter-object interactions.

Ray equation: Let us assume a camera with intrinsic parameters (f_u, f_v, u_0, v_0) and the image pixel (u, v). Then, the ray originating from the camera centre travelling from pixel can be given by:

$$\left(\frac{(u-u_0)}{f_u}, \frac{(v-v_0)}{f_v}, 1\right) \tag{2.2}$$

direction (u-u0 , v-v0 , 1) in the camera coordinate frame. fu fv Given the camera extrinsics, the origin and direction of the ray r can also be inferred in the world frame.

2.2.2 Rasterization

Rendering process can be decomposed into two tasks: visibility and shading. Both ray tracing and rasterization provide solutions to visibility problem. Ray tracing is essentially image centric because we shoot rays from the camera to the scene. The other way around is the approach used in rasterization. To be more specific, rasterization solves the visibility problem by projecting (using camera projection) all the triangles onto the image plane. This approach is object-centric as we start from geometry and move to image. Each pixel is assigned to the nearest triangle that is projected on to the pixel. Its color is determined by the barycentric interpolation of vertex colors of the triangle. Rasterization algorithm stores depth buffer to determine the depth order of triangles. To summarize, rasterization algorithm is very well suited for the GPU and is actually the rendering technique applied by GPUs to generate images of 3D objects and it can also easily be run in parallel.

2.3 Camera projection

In the previous section, we discussed about shooting camera rays in case of ray tracing and projection of triangles onto screen in rasterization. Projection is nothing more than 4x4 matrices, which are designed so that when multiplied with a 3D point in camera space, we end up with a new point which is the projected version of the original 3D point onto the image plane. In this section, we discuss about types of camera projection that we use predominantly in this thesis.

2.3.1 Perspective projection

Perspective projection or perspective transformation is a linear projection where three dimensional objects are projected on a picture plane. This has the effect that distant objects appear smaller than nearer objects i.e. the distance to an object is inversely proportional to its image size. Please refer Figure 2.3. Another important property of perspective projection is that lines which are parallel in nature (that is, meet at the point at infinity) appear to intersect in the projected image. For instance if railways are pictured with perspective projection, they appear to converge towards a single point, called the *vanishing point*. Photographic lenses and the human eye work in the same way, therefore perspective projection looks most realistic. The perspective camera model can be expressed mathematically as:

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.3}$$

2.3.2 Orthographic projection

It is the projection of a 3D object onto a plane by a set of parallel rays orthogonal to the image plane. This type of projection does leave parallel lines parallel, and it preserves relative distance between objects. A 3D scene is assumed to be at infinite distance from camera. Because of parallel projection, the x and y coordinates does not change. An example of this model would be if one holds an object above the ground at noon on a sunny day (so that the sun is directly overhead) and views the shadow of the object on the ground as the image of the object. Since the sun is so far away from us, all of the light rays hitting the object are effectively parallel, resulting in the described effect. The orthographic camera model can be expressed as:

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.4}$$

2.3.3 Weak-perspective projection

We can notice that the orthographic camera does not have any deformations to the object e.g. zooming. That is, we flatten out space through orthogonal projection, but we don't dilate, or scale the image. The weak perspective camera is nothing more than an orthographic camera, followed by a scaling of the resulting image. Please refer Figure 2.3(c). The main application of weak perspective camera model is that we can approximate the perspective camera projection with orthographic projection followed by scaling of the image. However, for such an approximation to hold, we need the objects that to be reasonably far away from the camera. Specifically, we need the differences in depth of the objects to be small, compared to the average depth of all of the objects, Z_{ave}. With orthographic projection an object

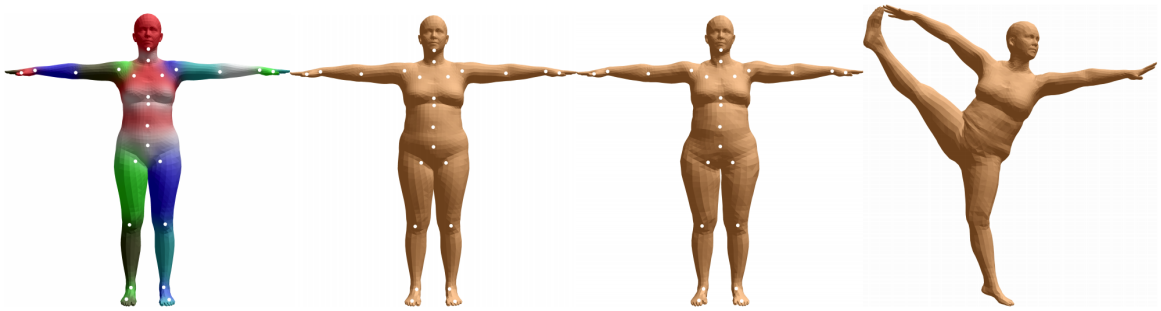

Figure 2.4 (a) Template mesh with blend weights indicated by color and joints shown in white. (b) With identity-driven blendshape contribution only (c) With the addition of of pose blend shapes in preparation for the split pose; note the expansion of the hips. (d) Deformed vertices reposed by dual quaternion skinning for the split pose. Figure adopted from [90]

of unit size (regardless of how far away it is) will have an image of unit size in the image plane. Thus, to make weak-perspective camera approximate the pinhole camera, we need to scale both axes of the orthographic image by $\alpha = \beta = \frac{f}{Z_{ave}}$. In this case, the camera projection matrix is:

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{Z_{ave}}{f} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.5}$$

Note that all the points in the projection undergo same scaling. For this to be valid, all points need to be at constant depth(Z_{ave}).

2.4 Parametric model: Skinned Multi-Person Linear (SMPL) [90]

SMPL is a realistic learned model of human body shape and pose that is compatible with existing rendering engines designed and developed for convenient animation of human bodies. SMPL deforms a template vertex by an additive blending of pose, shape, and soft tissue dynamics. Each SMPL mesh consists of N = 6890 vertices and K = 23 joints. The mesh has the same topology for men and women, spatially varying resolution, a clean quad structure, a segmentation into parts, initial blend weights, and a skeletal rig. The pose of the body is defined by vector $\theta = [\theta_0, \theta_1, ..., \theta_{23}]$ where $\theta_i \in \mathcal{R}^3$ denotes the axis-angle representation of the relative rotation of part i with respect to its parent in the kinematic tree. The deformation model of a template mesh represented by a vector of N concatenated vertices $T \in \mathcal{R}^{3N}$ can be represented using:

- Blend weights which are used for skinning $\mathcal{W} \in \mathcal{R}^{N \times K}$

- Blend Shape function $B_S(\beta) : \mathcal{R}^{|\beta|} \rightarrow \mathcal{R}^{3N}$ is a function that takes shape parameters β and outputs a blend shape sculpting the subject identity.
- A function to predict K joint locations $J(\beta) : \mathcal{R}^{|\beta|} \rightarrow \mathcal{R}^{3K}$ from the shape parameters β.
- A pose dependent blend shape function $B_P(\theta) : \mathcal{R}^{\theta} \rightarrow \mathcal{R}^{3N}$ that takes input vector of pose parameters, θ and outputs locations of updated vertices corresponding to pose deformation.

The Deformation process is outlined in Figure 2.4. Template T and blend weights W are plotted in (a). Then template vertices are deformed in (b) w.r.t shape blending function $B_S(\beta)$. The new joint locations after shape deformation are $J(\beta)$. The resultant mesh is deformed according to pose in (c). Finally, vertices are further deformed according to skinning methods. Each vertex's new displacement can be written mathematically as

$$t_i^{'} = \sum_{k=1}^{K} w_{k,i} G_k^{'}(\theta, J(\beta))(t_i + b_{S,i}(\beta) + b_{P,i}(\theta)) \tag{2.6}$$

where $G_k(\theta, J)$ is the world transformation of joint k and $w_{k,i}$ is the weight associated with vertex i to joint K. $b_{S,i}(\beta)$ and $b_{P,i}(\theta)$ is vertex i in $B_S(\beta)$ and $B_P(\theta)$ respectively.

Shape parameters: The deformations in shape can be represented as eigenfunctions of shape displacement matrix. SMPL uses 10 parameters to cover a wide range of shape variations. SMPL model doesn't capture hand and face movements. [111, 120] extends SMPL to hands and face movements. Recent upgrade to SMPL i.e. SMPL-X also provides 300 shape parameters offering more control over body shape simulation.

Chapter 3

3D Shape Analysis Using Distance Transform

In this chapter, we propose SplineNet, a deep network consisting of B-spline surfaces for classification of input 3D data represented in volumetric grid. We propose a novel, fast and robust characterization of 3D shapes that accounts for local geometric variations as well as global structure. We introduce learnable B-Spline surfaces in order to sense complex geometrical structures (large curvature variations). The locations of these surfaces are initialized over the voxel space and are learned during training phase. We derive analytical solutions for updates of B-spline surfaces during back propagation. We show results on publicly available dataset and achieve superior performance as compared to state-of-the-art method.

3.1 Introduction

3D shape acquisition and analysis is an active research area in both computer vision and graphics. Advancement in the field of 3D capture, owing to use of consumer depth sensors, has reinvigorated the research interest for scalable shape classification and recognition algorithms. Recently, deep neural networks have emerged as key learning framework for various computer vision tasks. Majority of existing works on deep learning on 3D data are proposed in volumetric representation where shapes are represented as occupancy grid which is analogous to pixels in the image, thereby directly extending concept of 2D convolution to 3D domain [8, 96, 159].

Nevertheless, the volumetric representation poses a serious computational disadvantage as most of the voxel grids are empty and results in redundant computation. Moreover, a 3D shape is determined by its surface and hence performing convolutions on the voxels inside the shape is sheer wastage of computation. This issue has been recently addressed in [83] by introducing field probing filters which effectively sense informative locations in the 3D space. This enables intelligent and sparse sampling in the grid space. However, the filters proposed in [83] are point-based, which evaluate functional value at a given point without accounting for geometrical information over the neighborhood. Hence this approach captures only global representation of voxelized 3D data for shape classification. As shown in Figure 5.1, the object has regions with flat as well as significantly varying curvature. In case of flat structures, sampling anywhere for estimating functional value (e.g., distance transform) is acceptable. However,

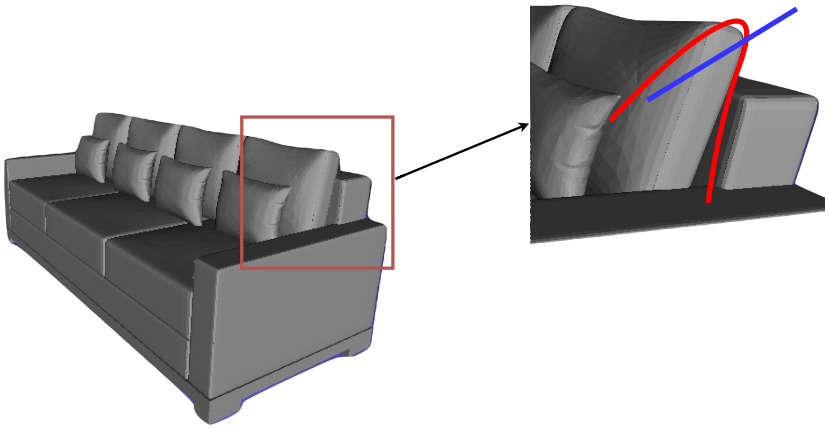

Figure 3.1 In case of planar surfaces, sampling anywhere on the surface results in the same vector field. To capture non-planar surfaces, estimating field value along the red curve captures the local variations. Points on the blue line will not capture the local topology

for regions with complex geometrical variations, point based sampling may not be sufficient. Hence, we introduce higher order *B*-spline surfaces to capture complex geometrical variations in the data. Our novel characterization of 3D shapes accounts for local information as well as global geometry. We built up on the learning scheme of [83] by introducing sets of B-spline surfaces instead of point filters, in order to sense complex geometrical structures (large curvature variations). The locations of these surfaces are initialized randomly over the voxel space and are learned over training phase. We modify the dot product layer of [83] to aggregate local sensing and provide the global characterization of the input data. The key contributions of this chapter are listed as follows.

- We propose SplineNet, a deep network consist of B-spline surfaces for classification of input 3D data represented in volumetric fields. To the best of our knowledge, parametric curves and surfaces are not proposed in a learning setup in deep neural network for classification applications.
- We derive analytical solutions for updates of B-spline surfaces during back propagation.
- Our algorithm generates local-geometry aware global characterization of 3D shape using neural network.
- We show results on ModelNet[161] dataset and achieve superior performance to state-of-the-art method.

The remainder of this chapter is organized as follows. In Section 3.2, we present brief survey of the most relevant works on rigid 3D shape analysis, in particular, classification. Section 3.3 provide background details of B-spline curves and surfaces. Subsequently, in Section 3.4 we outline our proposed B-spline neural network followed by details of experiments and results in Section 3.5.

3.2 Related work

Here we present solutions for 3D shape analysis using traditional hand-crafted features and recent learning representations from data via deep neural networks.

3.2.1 Shape descriptors

3D feature description using global and local analysis has drawn its inspiration from 2D images algorithms where features are represented using the sparse or dense set of local feature , e.g. SIFT [92]. The existing local feature descriptors are broadly categorized into extrinsic and intrinsic based on how they evaluate local geometry around a feature point. Extrinsic descriptors capture the local Euclidean geometry. Surface normals is one such descriptor used in many applications including 3D shape reconstruction, plane extraction, and point set registration [29, 53]. *Point descriptors* [33, 166] encode local features on the surface mesh by defining relative local surface normal at a sample point with respect to a superimposed plane or line segment at the sample points. Local surface normal vectors are computed at discrete points on the surface mesh to capture the local surface features in [43]. Other popular extrinsic descriptors are [62, 76]. Intrinsic descriptors capture pose invariant intrinsic geometry of the underlying manifold. However, these descriptors are confined to articulated 3D shapes. Another class of local shape descriptors are *ring-based* [100, 113] which are based on local sampling of a predefined metric over the discrete 3D surface mesh.

Global shape descriptors are quite popular for shape retrieval tasks where a single representation is used for shape retrieval. The Laplacian-Beltrami operator [121] is proposed to compute the diffusion-based shape descriptors. Heat Kernel Signature(HKS) uses eigen spectrum of the Laplacian operator to extract intrinsic properties by evaluating heat distribution on vertices of a mesh. The Wave Kernel Signature (WKS) [9] is another popular category of global descriptors that employ principles of quantum mechanics instead of heat diffusion on eigen spectrum to characterize the shape. Similarly, [133] proposed to characterize global representation of a shape which is robust against isometric deformations. This is achieved by computing the geodesic distances between sample points on the 3D surface mesh.

3.2.2 Deep learning on 3D data

Majority of deep learning works on 3D data are based on the idea of partitioning the 3D space into regular voxels and extending 2D CNNs to voxels. A deep belief network is trained for classification of ModelNet dataset in [161]. Voxel based variational auto-encoder is trained in [24] for shape modeling and object classification tasks. 3D object is recognized[124] by predicting the pose of the object in addition to the class label as a parallel task. However, these methods cannot be scaled to high resolutions due to inherent increase in computational complexity. The issue has been addressed by [83] defining field probing scheme. Nevertheless, only global representation of the object is learned.

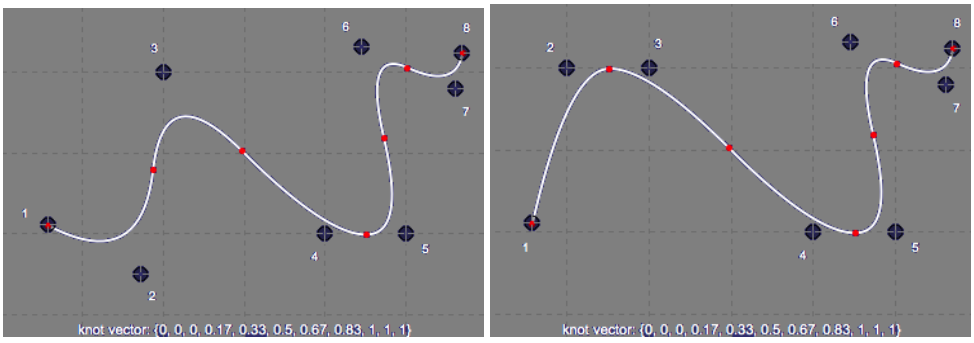

Figure 3.2 The curve at left is deformed at position 2. The resulting curve at right is deformed around 2nd position. Similar illustration on B-Spline surfaces can be found in link.

Extensive literature on 2D CNNs has prompted many works to render images of a 3D object and use these images for feature description through CNNs. Each 3D shape is converted into a panoramic view and recognized in [128]. Information from multiple views of the object [136] is combined through novel view-pooling. [68] is also proposed on similar lines where they treat viewpoints as latent variables. 3D object is generated from a single 2D image [158] by generating images of surface normals, depth from various camera view points.

Other section of works operate directly on point cloud. PointNet [115] is a pioneering work in this direction that is proposed for raw point cloud as input and generate a permutation-invariant representation of the object. PointNet++ [116] proposed to use hierarchical neural network where PointNet is applied recursively on a nested partitioning of input point set. This approach addresses the drawback of local structure sensing of PointNet. A new architecture is proposed in [72] performs multiplicative transformations and shares parameters of these transformations according to the subdivisions of the point clouds imposed onto them by kdtrees. Similarly, OCNN [155] is proposed which is built upon the octree representation of 3D shapes. Challenges in unsupervised learning on point clouds is addressed by [168] by training an auto-encoder where decoder deforms a canonical 2D grid onto the underlying 3D object surface of a point cloud.

Graph-based approaches characterizes point clouds as graphs. A 3D point cloud can be represented as a polygon mesh or connectivity graph which is converted to the spectral representation and apply convolution in spectral domain employing analogy between the classical Fourier transforms and projections onto the eigen basis of the graph Laplacian operator [26]. Recurrent Chebyshev polynomials to circumvent the problem of computation of the Laplacian eigenvectors is proposed in [36]. The first work in this approach is Geodesic CNN [94] where local patches represented in geodesic polar coordinates were applied with filters. Anisotropic heat kernels were used as an alternative way of extracting intrinsic patches on manifolds [23]. [95] provides a good overview of recent advancement in the field of graph deep learning for non-Euclidean data.

3.3 Background: B-Spline surfaces

In this section, we discuss various properties of B-spline surfaces that are exploited for efficient feature representation using neural network. Parametric curves and surfaces are most commonly used in computer graphics for generation of 3D objects. A parametric surface in 3D is defined by three bivariate functions as

$$\alpha(u,v) = (\alpha_x(u,v), \alpha_y(u,v), \alpha_z(u,v)) \tag{3.1}$$

B-spline surfaces share similar properties to that of B-spline curves. Since curves are easier to visualize, we discuss about B-spline curve, a parametric polynomial curve which is defined as

$$\alpha(u) = \sum_{i=0}^{n} N_{i,k}(u) x_i \tag{3.2}$$

$$0 \leq u \leq n - k + 2$$

where k is the order of the curve and we have $n+1$ control points and $N_{i,k}$ are termed as $SplineBasis$. The curve/surface is obtained by the blending of its control points and the blending functions are provided by spline basis. The most important properties of B-spline curves and surfaces include:

- The curve can be defined using arbitrarily large number of points without increasing the degree of the curve.
- The curve is a piecewise curve with each component a curve of degree $k-1$.
- A B-spline curve is enclosed in the convex hull of its control polyline. Specifically, if u is in knot span $[u_i, u_{i+1})$, then $\alpha(u)$ is in the convex hull of control points $x_{i-k}, x_{i-k+1}, ..., x_i$.
- The continuity of the curve/surface is $k-2$ in each parametric dimension and hence is differentiable and derivatives can be computed analytically.
- Local Modification: By changing the position of control point x_i, only affects the curve $\alpha(u)$ on interval $[u_i, u_{i+k+1})$ as shown in Figure 3.2. This property is primarily exploited in our method for the calculation of local surface information.

3.3.0.1 B-spline basis calculation

$N_{i,k}$ in Equation 2 can be computed recursively as

$$N_{i,k}(u) = \frac{(u - t_i) N_{i,k-1}(u)}{t_{i+k-1} - t_i} + \frac{(t_{i+k} - u) N_{i+1,k+1}(u)}{t_{i+k} - t_{i+1}} \tag{3.3}$$

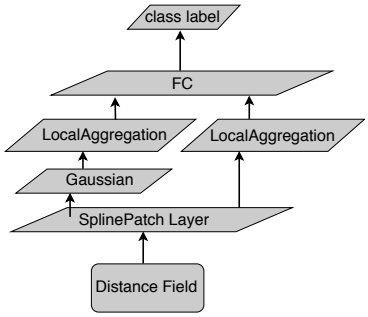

Figure 3.3 Overview of our SplineNet architecture. Input shapes represented in volumetric fields are fed to SplinePatch layer for effective local sensing which is then optionally passed to Gaussian layer to retain values near surface boundaries. LocalAggregation layer accumulates local sensing to give local geometry aware global characterization of input shapes. Resulting characterization is fed to Fully Connected(FC) layers from which class label is predicted.

where t_i are knot values. The number of knot values is equal to sum of number of points and degree of the curve. They are computed as follows

$$t_i = \begin{cases} 0 & \text{if } i < k \\ i - k + 1 & \text{if } k \leq i \leq n \\ n - k + 2 & \text{if } i > n \end{cases}$$

$$N_{i,k}(u) = \begin{cases} 1 & \text{if } t_i \leq u \leq t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

3.4 Our method

We design a novel learning scheme for classification of rigid 3D objects which is learned using deep neural network. Figure 3.3 outlines the architecture of the proposed SplineNet. The input to our network is a 3D distance field or any differentiable vector field. Subsequently, we process this vector field input with novel SplinePatch layer for capturing local geometric variations. Later on, we pass the output of this layer to either Gaussian layer or directly to LocalAggregration layer. As explained in [83], Gaussian layer ensures that function values around object surface are retained. LocalAggregation layer accumulates sensing done by spline surfaces to recover the global characterization of object structure. Finally, the output is fed to a FC (Fully Connected) layer to be able to learn and predict final classifications labels.

3.4.1 SplinePatch layer

This layer is accountable to sense the local information and pass it to LocalAggregation layer. It consists of N sets of surfaces. Each set is initialized with a line in the 3D grid space as described in Section 3.5.1. On each line, we randomly sample P points. At every sampled point, a B-spline patch is initialized with $(m+1) \times (n+1)$ control points along $U \times V$ directions. Essentially, this layers contains $N \times P$ patches. Each control point is three dimensional and the total number of parameters in this layer are $N \times P \times ((m+1) \times (n+1)) \times 3$.

Each point on the surface is expressed as

$$\alpha(u,v)(x) = \sum_{i=0}^{m+1} \sum_{j=0}^{n+1} N_{i,k}(u) * N_{j,l}(v) * x_{i,j} \tag{3.4}$$

$$0 \leq u \leq m-k+2$$

$$0 \leq v \leq n-l+2$$

The parametric space of U, V is divided into $D = M \times N$ divisions i.e. if 3 and 2 divisions, the parametric space is $(0-0.33; 0.33-0.66; 0.66-1.0)$ and $(0-0.5; 0.5-1)$ along U and V directions respectively. It is illustrated on a spline curve for better visualization in in Fig 3.5. The curve is generated by 12 control points represented in black dots. The red dots represents knot values. B-spline curve is piece-wise continuous in every consecutive knot interval. For instance, each knot interval can be assumed as a division of parametric space. We randomly evaluate the functional values in each segment and the minimum value $f(u_i)$ is sent to further layers. We perform similar operations in surface which is a natural extension of the curve In each division, we randomly sample s sampling points. We evaluate the differentiable functional value at these sampling points $f(\alpha(u,v))$, for instance, distance transform. Notice that the functional value at sampled points depend on the control points $x_{i,j}$. Hence, during back-propagation, the functional value affects the location of control points. The gradient allows these locations to drift for effective sensing. The analytical solution for updates is discussed in back propagation section.

These sampling points provide local sensing. We introduce min-pooling in each division for distance transform function which is forwarded to LocalAggregation layer. Since the sensing is done in all divisions of a patch, the network tries to approximate the surface over the region.

$$f(\alpha(u,v))_{d_i,p,n} = min(f(\alpha(u_s,v_s)))\forall s \in d_i \tag{3.5}$$

where d_i is i^{th} division of p^{th} patch in $n^{th} set$.

3.4.2 Gaussian layer

Points or locations that are distant from object surface has larger distance value from the distance field and thus, contribute less information about object. Feeding this information to local aggregation layer

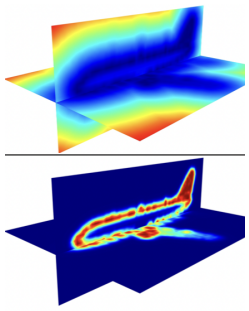

Figure 3.4 Affect of Gaussian layer on distance field. Figure adopted from [83]

is redundant and the sample locations will not converge. To emphasize the importance of samples in the vicinity of the object surface, we apply a Gaussian transform (inverse exponential) on the distances so that regions approaching the zero surface have larger weights while distant regions matter less as shown in figure Figure 3.4.

3.4.3 LocalAggregation layer

The LocalAggregation layer attempts to perform a two-level aggregation. Firstly, this layer aggregates the functional values sensed from each of the divisions of a patch (output of SplinePatch layer). This operation helps the network to analyze the local geometry. The input to this layer is the output of SplinePatch/Gaussian layer which is of dimension $N \times P \times D \times C$ where C is the number of channels in the function and concatenates the local information of all divisions around each patch. Essentially, the feature of a patch is D dimensional. We have also used various other operations such as average of the functional values of all the divisions. However, concatenation results in better performance.

Second level aggregation attempts to generate global characterization which is obtained by performing dot product operation across the set.

$$f_{net,n} = \sum_{p=0}^{P} \sum_{d_i=0}^{D} \sum_{c=0}^{C} f(\alpha(u,v))_{d_i,p,c} \times \beta_{d_i,p,c} \tag{3.6}$$

where f_{net} is the net global and local contribution of the n^{th} set and $\beta_{d_i,p,c}$ is the weight of d_i^{th} division of p^{th} patch and c^{th} channel. The output of this layer is of dimension N and is connected to Fully connected (FC) layers. The final FC layer is connected loss layer to predict the label of the input shape.

3.4.4 Back-propagation

For making SplineNet trainable, it is necessary to compute gradients with respect to the location of control points and the weights in the LocalAggregation layer. Let E be the error function. Analytical

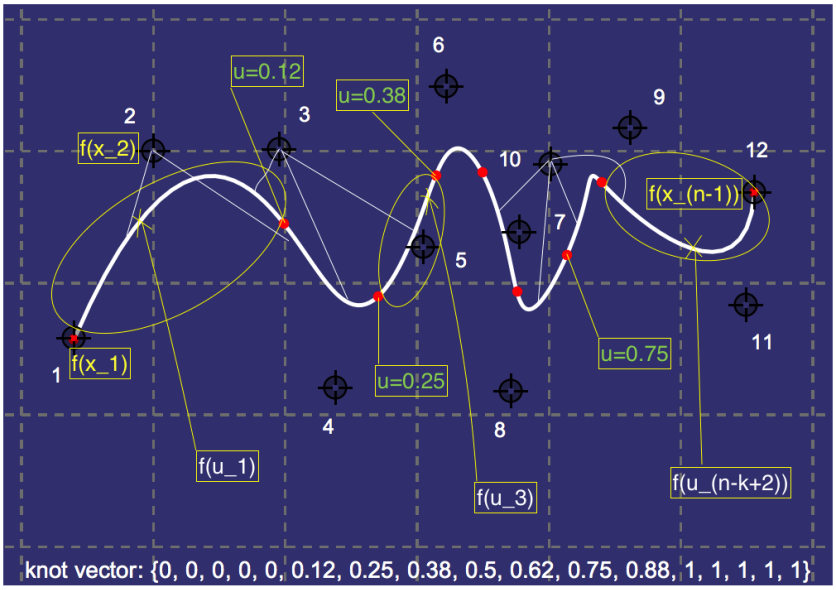

Figure 3.5 Quartic Curve generated by 12 control points. Knot values are shown. Parametric divisions are visualized.

solutions can be derived as

$$\frac{\partial E}{\partial f_{net,n}} = \begin{bmatrix} \frac{\partial E}{\partial f(\alpha(u,v))_{d_i,p,c}} \\ \frac{\partial E}{\partial \beta_{d_i,p,c}} \end{bmatrix} = \begin{bmatrix} \beta_{d_i,p,c} \\ f(\alpha(u,v))_{d_i,p,c} \end{bmatrix} \tag{3.7}$$

The update of control points can be derived as

$$\frac{\partial E}{\partial x_{i,j}} = \sum_{d_i=0}^{D} \sum_{c=0}^{C} \left(\frac{\partial E}{\partial f(\alpha(u_m, v_m))_{d_i,p,c}} * \frac{\partial f(\alpha(u_m, v_m))_{d_i,p,c}}{\partial \alpha(u_m, v_m)_{d_i,,p,c}} * \frac{\partial \alpha(u_m, v_m)_{d_i,p,c}}{\partial x_{i,j}} \right) \tag{3.8}$$

$$= \sum_{d_i=0}^{D} \sum_{c=0}^{C} (\beta_{d_i,p,c} * f^{'}(\alpha(u_m, v_m))_{d_i,p,c}) * N_{i,k}(u_m) * N_{j,l}(v_m) \tag{3.9}$$

From Eq. 3.4,

$$\frac{\partial \alpha(u_m, v_m)_{d_i,p,c}}{\partial x_{i,j}} = N_{i,k}(u_m) * N_{j,l}(v_m) \tag{3.10}$$

$$(u_m, v_m)_{d_i,p,c} = _{u_s,v_s} \ (f(\alpha(u_s, v_s)))_{d_i,p,c} \tag{3.11}$$

Similarly, $\frac{\partial E}{\partial y_{i,j}}$ and $\frac{\partial E}{\partial z_{i,j}}$ are updated. During training, while computing the functional value of a division, the indices (u_m, v_m) of min pooling operation are stored. The detailed algorithm is provided below in Algorithm 1.

3.5 Experiments and results

We used Nvidia's GTX 1080Ti, with 11 GB of VRAM to train our network. The point clouds are converted into distance fields. We used a batch size of 32, learning rate of 0.01 with SGD solver and momentum 0.75 and weight decay of 10^{-5}.

Baseline: We show our results on ModelNet40 [161] dataset which has 40 classes of rigid 3D CAD models. As mentioned in Section 3.2, there are several works for classification of ModelNet datasets. However, the input formats are different for each of these works. We will compare our results with volumetric input, in particular, the FPNN [83].

ALGORITHM 1: The learning scheme

Result: Updated locations of control points of each surface.

Initialize: Number of sets of surfaces N, number of candidates in each set P, number of sampling points S, randomly initialize locations of control points of each surface $x_{i,j}, y_{i,j}, z_{i,j}$, Degree of the curve along both parametric directions K, number of divisions in parametric space D, number of iterations T;

while $iterations \leq T$ **do**

  Forward Pass: for each set *n* in *N*

  for *p := 0 to P* **do**

    for *d := 0 to D* **do**

      for *s := 0 to S* **do**

        evaluate $\alpha(u_s, v_s)$ using Eq 3.4

        $f(\alpha(u_s, v_s))_{d_i,p} \leftarrow min(f(\alpha(u_s, v_s)))\forall s \in S\ u_m, v_m \leftarrow_{u_s,v_s} f(\alpha(u_s, v_s))$

    $f_{net,n} = concat(f(\alpha(u_s, v_s))_{d_i,p}))\forall d_i \in D$

  Calculate global value by Eq 3.6

  Backward Pass:

  for *p := 0 to P* **do**

    for *d := 0 to D* **do**

      Compute $\frac{\partial E}{\partial x_{i,j}}, \frac{\partial E}{\partial y_{i,j}}$ and $\frac{\partial E}{\partial z_{i,j}}$

      by using Eq 3.7 and 3.8 with u_m, v_m

We train our SplineNet with varied parameters and settings and compare the results with FPNN. To argue that constructing local geometry aware global characterization greatly enhances the classification

Method	without updating locations	Updated locations
FPNN [83]	79.1	85.0
OURS	**82.94**	**86.8**

Table 3.1 A comparison of accuracy on ModelNet40 [161] in 1FC setting on 64 resolution input.

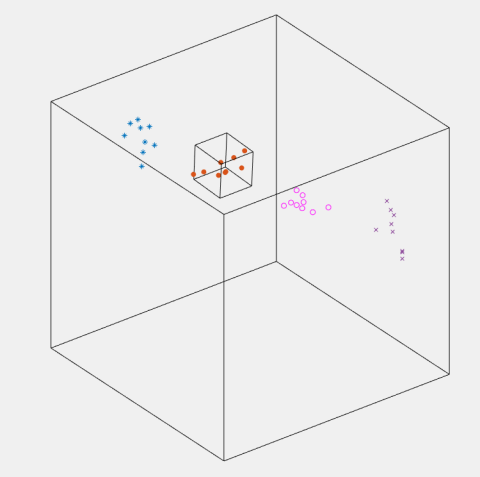

Figure 3.6 Initialization of control points of each surface. A single set of four surfaces i.e. their control points are visualized. Please refer Section 3.5.1 for more details.

performance on ModelNet40 dataset, we perform experiments where the locations of the control points of each surface are updated and not updated. For this evaluation, we have used 1024 sets of surfaces wherein each set has 8 surfaces and each surface has 9 control points. The number of divisions in each surface are 6. In each division, the sampling points are 5. The order of the polynomial is fixed to 4 along each of the parametric direction. We show the quantitative results of both the experiments in Table 3.1.

It is evident that local sensing is adding more essential information and the performance is increased from 79.1% to 82.94%. FPNN constructs a robust global representation which can be improved by adding local geometry to achieve a better performance as shown in Table 1 on 64 resolution data.

3.5.1 Initialization of surfaces

A surface is defined by the locations of its control points. For initialization of these control points, we assume the volumetric grid to be unit dimensional. We initialize a line of random length l and orientation varying from $0.1 - 0.9$. On the line, we randomly chose 8 points. With the chosen points as center, we initialize cuboid which has random length, breadth, height and orientation with not exceeding $0.4 * l$.

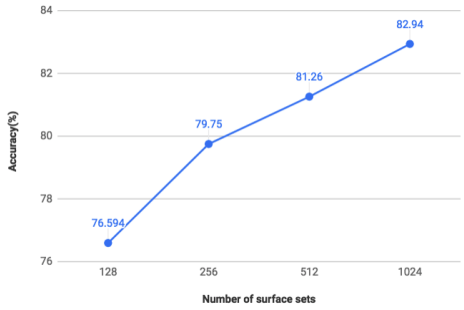

Figure 3.7 Accuracy of SplineNet with varied number of surface sets

order	(3,3)	(4,4)
accuracy	82.17	82.94

Table 3.2 Evaluation of our approach by varying the order of the curve. We keep the order of curve same along U,V directions

This procedure of initialization ensures that each set of surfaces has a different span in the volumetric grid and the range of sensing is maintained. Within each cuboid we initialize uniformly sampled control points for each of the surface. We demonstrate only four candidates in Figure 3.6 for better visualization. However, in experiments we used 8 surfaces.

3.5.2 Hyper-parameter estimation

The network has many hyper parameters which include number of surface sets to be initialized, order of the surfaces, number of divisions of the parametric space etc. In order to estimate these parameters, we train the SplineNet under different settings without updating the locations of the control points. We performed experiments with varying number of sets of b-spline surfaces initialized. The quantitative results are shown in Figure 3.7. It is to be observed that 256 sets of surfaces with 8 surfaces in each set is required to match the performance of FPNN which has 1024 filters and 8 points in a filter.

The smoothness of the surface is dependent on the order of the curve. If the order i.e. (degree+1) of the curve is 2, we get piecewise continuous planes. Increasing in the order of the surface results in a much smoother surface. In all our experiments, we use the same order along U,V directions. Table 2 show the results of experiments. To sense the surface information locally, we divide each patch into several divisions in parametric space. We choose number of divisions emperically which is reported in Table 3.2. From each division, we evaluate the functional value at $n = 5$ random locations. We take minimum of these values and forward to LocalAggregation layer. Increase in the number of divisions results in dense sampling on the surface. Hence, the performance is increased with the increase of

Divisions	4	6	9
accuracy	82.3	82.94	82.96

Table 3.3 Performance of number of parametric divisions on 128 resolution data

Resolution	32	64	128
accuracy	84.8	86.8	87.4

Table 3.4 Classification accuracy of our method on varying input resolutions

number of divisions as reported in Table 3.3. We train our network with number of divisions set to 6 further in all experiments.

To study the importance of various functions on local geometry of 3D shapes, we tested with several functions in the LocalAggregation layer apart from concatenation. From each of the patch initialized, we take the mean of the functional values of randomly sampled points instead of concatenating. We consider this mean value as the contributed functional value for of the patch which is fed to further layers. We have achieved an under-par accuracy of 78.3 accuracy when the locations are not updated. Similarly, we also used standard deviation in the functional values as the net contribution and learned that concatenation greatly influences the performance.

We also observe that our method performs better when the resolution of the input is high as shown in Table 3.4. This is essentially because of the fact that in forward propagation, we evaluate the functional value at a point by performing tri-linear interpolation. Increasing the resolution results in effective sensing in local neighborhood which the surface attempts to exploit. Finally, we compare our method with the then SOTA in Table 3.5 where our method outperform the existing approaches.

Method	Accuracy
Aravind et al. [8]	86.5
3D-CapsNets [79]	82.73
VSL [87]	84.5
Ours	87.4

Table 3.5 Classification accuracy (%) on ModelNet40 comparison of our model and other recent volumetric methods

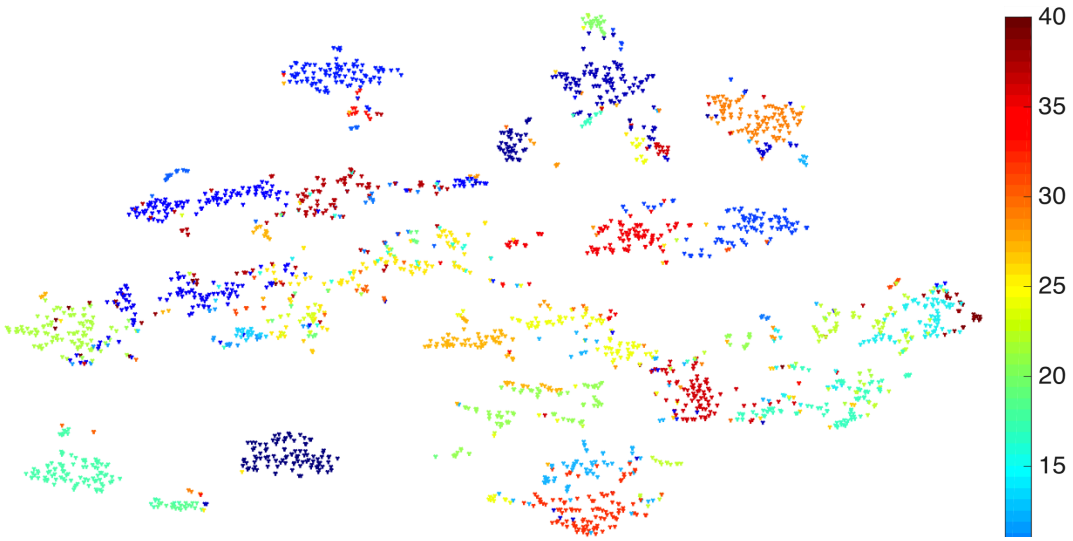

Figure 3.8 tSNE feature visualization

3.5.3 Visualization of SplineNet features

We show tSNE features of the fully connected layer of our SplineNet in Figure 3.8. It is easy to infer that SplineNet is able to efficiently embed similar class candidates in one neighborhood. This embedding suggests that the learned features are generalizable for retrieval tasks whereas the network is trained for classification.

3.6 Summary

In this chapter, we proposed SplineNet, a novel learning paradigm to address the challenging issues of efficient and effective 3D volumetric data classification. In particular, to account for local geometric variations while generating global representation of 3D data, we introduce B-spline surfaces in SplineNet. The locations of these surfaces are learned from data and analytical solutions to perform back propagation are derived. We show results on publicly available dataset and show superior over state-of-the-method. We also demonstrate the robustness of features learned by SplineNet.

Chapter 4

3D Reconstruction of Human Bodies Using Volumetric Convolution

In previous chapter, we started with shape analysis of rigid objects. In coming chapters, we focus on a more challenging problem of 3D human body (non-rigid objects) reconstruction. In this chapter, we propose a deep learning based solution for textured 3D reconstruction of human body shapes from a single view RGB input. This is achieved by first recovering the volumetric shape of non-rigid human body shapes given a single view RGB image followed by orthographic texture view synthesis using the respective depth projection of the reconstructed (volumetric) shape and input RGB image. We propose to co-learn the depth information readily available with affordable RGBD sensors (e.g., Kinect) and showing multiple views of the same object while training the network. We show superior reconstruction performance in terms of quantitative and qualitative results, on both, publicly available datasets (by simulating the depth channel with virtual Kinect) as well as real RGBD data collected with our calibrated multi Kinect setup.

4.1 Introduction

Recovering the textured 3D model of non-rigid human shapes from images is of high practical importance in the entertainment industry, e-commerce, health care (physiotherapy), mobile based AR/VR platforms, etc. This is a challenging task as the object geometry of non-rigid human shapes evolve over time, yielding a large space of complex body poses as well as shape variations. In addition to this, there are several other challenges such as self-occlusions by body parts, obstructions due to free form clothing, background clutter (in non-studio setup), sparse set of cameras with non-overlapping fields of views, sensor noise, etc. Model based reconstruction techniques attempt to overcome some of these limitations. However, accurate geometrical information over the shape surface is not retained and are typically applicable only for tight clothing scenarios [7, 16, 21].

Traditionally, calibrated multi-camera setups have been employed to recover textured 3D models through triangulation or voxel carving [152, 21]. However, these techniques yield reconstructions with severe topological noise [127]. Recent attempts replace/augment the capture setup with high resolution depth sensors [179, 38] making the setup more accurate but less affordable. Nevertheless, the funda-

mental limitation of these techniques is the requirement of a calibrated multi-camera/sensors that restrict their applicability to studio environments. We aim to achieve 3D reconstruction of textured human body models using single/multiple affordable RGB/D sensor(s) in calibration-free setup. Model based reconstruction techniques attempt to overcome some of these limitations. However, the accurate geometrical information over the shape surface is not retained [7, 16, 21]. Another approach involves performing non-rigid registration over point clouds (coming from RGBD sensors) to recover textured surface reconstructions [179, 38]. Nevertheless, even such methods suffer from challenges shown in Figure 1.2. More importantly, majority of approaches belonging to the three shape reconstruction paradigms listed above expect a high-resolution calibrated multi-camera setup which is expensive and limited in their ubiquitousness. In a calibration-free setting such as ours, motion capture of dynamic scenes is difficult, thus, making both reconstruction and texture recovery not trivial.

In this chapter, we propose a deep learning based solution for textured 3D reconstruction of human body shapes given an input RGB image, in a calibration-free environment. Given a single view RGB image, both reconstruction and texture generation are ill-posed problems. Thus, we proposed to co-learn the depth cues (using depth image obtained from affordable sensors like Kinect) with RGB images while training the network. This helps deep network to learn the space of complex body poses, which otherwise is difficult with just 2D content in RGB images. Although we propose to learn the reconstruction network with multi-view RGB and depth images (shown one at a time during training), co-learning them with shared filters enabled us to recover 3D volumetric shapes using just single RGB image at test time. Apart from the challenge of non-rigid poses, the depth information also helps addressing the challenges caused by cluttered background, shape variations and free form clothing. Our texture recovery network uses variational auto-encoder to generate orthographic textured images of reconstructed body models that are subsequently backprojected to recover a texture 3D mesh model. We show quantitative and qualitative results on three publicly available datasets (by simulating the RGB and D whenever unavailable) as well as real RGBD data collected with calibrated multi Kinect setup. The key contributions of this work are:

- First, we introduce a novel deep learning pipeline to obtain the textured 3D models of non-rigid human body shapes from a single image. (Section 4.2) and show reconstruction results obtained using single view RGB input image. To the best of our knowledge, both, non-rigid human body reconstruction as well as texture recovery of human body models has not been attempted in a calibration-free environment using deep learning.

- Second, we demonstrate the importance of depth cues (*used only at train time*) for the task of non-rigid reconstruction. This is achieved by our novel training methodology of alternating RGB and D in order to capture the large space of pose and shape deformation.

- Third, we show that our model can partially handle non-rigid deformations induced by free form clothing, as we do not impose any model constraint while training the volumetric reconstruction network.

- Fourth, we proposed to use depth cues for texture recovery in the variational auto-encoder setup which are later utilized by Generative Adversial Network to produce relatively high quality textures.

- Finally, we collected a real dataset (that shall be publicly released) of textured 3D human body models and their corresponding multi-view RGBD, that can be used in solving a variety of other problems such as human tracking, segmentation etc.

4.1.1 Related work

Shape reconstruction can be broadly categorized into model-free and model-based approaches. Model-free techniques attempt to register multiple depth frames captured by sensors from different viewpoints to obtain a complete 3D scan. KinectFusion [57] create 3D of rigid objects by moving RGBD sensor. Reconstructions of non-rigid surface deformations from high-resolution monocular depth scans by using a smooth template as a geometric prior is proposed in [82]. Model-based approaches parametrize a shape by estimating low dimensional parameters for pose and shape independently. These techniques estimate parameters in order to recover complete 3D models from partial data. Zhang et al. [171] train a personalized body model by using the registrations obtained by registering several Kinect scans of a subject in multiple poses. A mapping is learned from depth images to initial body shape and pose parameters as proposed in Perbet et al[112]. Coarse to fine processing is employed in [15] for a detailed 3D reconstruction of freely moving humans from monocular RGB-D Kinect sequences. However, these methods work only for tight clothing scenarios.

Recent advancement in deep networks has enabled learning class specific 3D structure of a set of objects (e.g. cars, chairs, rooms) using large scale dataset of synthetic 3D models for deep network training [162, 160, 32, 167, 142]. ShapeNet[162] proposed a deep network representation with a convolutional deep belief network to give a probabilistic representation of the voxel grid. Along similar lines, 3D Generative Adversarial Networks (GAN's) were proposed to learn a probabilistic latent space of rigid objects (such as chairs, tables) in [160]. [167] proposed an encoder-decoder network that utilizes observations from the 2D space, and without supervision from 3D, performs reconstruction for a few classes of object. This relationship between 2D and 3D is further exploited in [142] where they define a loss function in terms of ray consistency and train for single view reconstruction. However, these methods have been employed only for rigid object reconstruction.

In regard to non-rigid reconstruction, [132] proposed to directly achieve the surface reconstruction by learning a mapping between 3D shapes and the geometry image representation (introduced in [131]). One of the key limitations of their method is that it is only suitable for genus-0 meshes. This constraint is frequently violated in real-world human body shapes due to topological noise [127] induced by complex poses, clothing, etc. Another very recent work proposed in [157] use multi-view silhouette images to obtain reconstructions of free form blob-like objects/sculptures. However, the use of silhouettes limits the application to scenarios where background subtraction is assumed to be trivial. All these initial

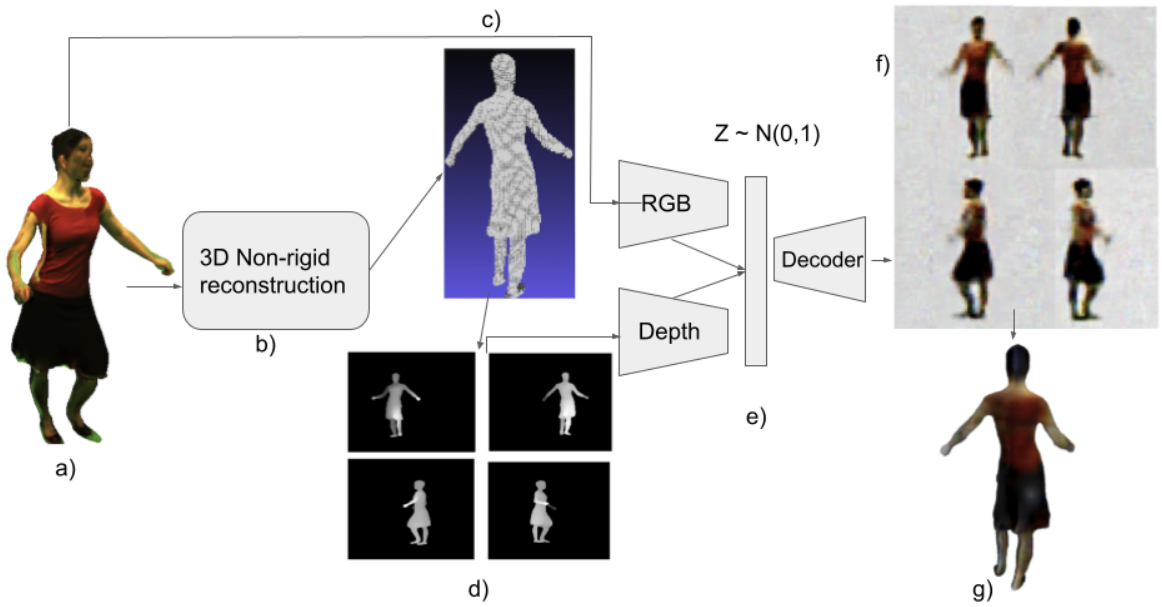

Figure 4.1 Proposed pipeline for reconstructing textured non-rigid 3D human body models using single view perspective RGB image during inference.

efforts are focused on texture less 3D reconstruction and do not seem to directly extendable to non-rigid human body shapes.

In regard to texture generation, recent work on multi-view synthesis [172, 175] propose to generate an image from alternate view-point given an input image. Recent work proposed in [66] attempts texture recovery for category specific 3D mesh models. However, the texture map is predicted only for the mean shape in the UV space and shown on reconstruction of birds. Nevertheless, this is not easily generalizable to human body shapes as accurate warping is non-trivial due to large space of shape and pose variations associated with human body. Very recently, a model based learning paradigm outlined in [3] proposed to generate human body models by estimating their deviation from nearest SMPL model. Nevertheless, to recover the texture, the person is assumed to be seen from all sides in a video sequence. Additionally such model based methods would fail to deal with large geometric deformations induced by free-form clothing (e.g., rob or long skirt) scenarios.

4.2 Our method

Figure 4.1 shows the *test time* flow of the proposed end-to-end pipeline for reconstruction of textured non-rigid 3D shapes where in (a), we perform a voxelized 3D reconstruction (c) using the reconstruction

network (b). Then, to add texture to the generated 3D model, we first convert the voxels to a mesh representation using Poisson's surface reconstruction algorithm [70] and capture its four orthographic depth maps (d). These are fed as an input to the texture recovery network (e), along with the perspective RGB views used for reconstruction (a). The texture recovery network produces orthographic RGB images (f), that are back-projected onto the reconstructed model, to obtain the textured 3D model (g). A detailed outline of each stage is given below.

4.2.1 Non-rigid reconstruction

We train the network with multiple views (typically 3) of the input actor. While during inference, we propose to use single RGB image to recover both shape and color of the predicted model. In this regard, we propose to use encoder-decoder network with two methods for combining multi-view information - i) a 3D GRU (like in [32]) and ii) max pooling of the CNN feature vectors of the views (like in [157]). The above two settings are considered to show that irrespective of the method of combining multi-view information for non-rigid human body reconstruction, depth information immensely help to capture the vast range of complex pose and shape variations while enhancing the reconstruction accuracy. Moreover, this provides the added advantage of obtaining 3D reconstruction from either only RGB or only D at test time, as discussed below.

Network architecture.
a) *Encoder* - ResNet-18 is used here that takes images of size 256x256 in one of the input modes (see below) and produces a 1024 dimensional feature vector. Each view produces one such feature vector, which is combined in the mulit-view module. We use Leaky ReLU as the activation function in both the encoder and decoder.
b) *Multi-view Module* - Multi-view information is combined using either a max-pooling of the 1D CNN feature vectors, or using a 3D-GRU. The outputs are resized to 4^3 and fed to the decoder
c) *Decoder* - Deconvolutional ResNet-18 is used here that up-samples the output of the multi-view module to 128^3

Input modes. In order to capture the large space of complex pose and shape deformations of humans, we experiment with four input modes:
a) *RGB* - This setup is commonly used in rigid body reconstructions [142, 32]. However, we qualitatively and quantitatively show in Section 4.3.2 that this setup in inadequate for reconstructing non-rigid shapes.
b) *D* - The premise behind this mode is that depth-maps give us information about the geometry of the object, which as seen in Section 4.3.2, help in significantly enhancing the reconstruction quality.
c) *RGBD* - In order to exploit both the depth and color channels, we augment RGB with D in a 4 channel input setup.
d) *RGB/D* - Lastly, we propose a unique training methodology that gives us superior performance than

the above 3 modes. Here, at train time, each mesh is reconstructed from either only multi-view RGB or multi-view depth. Thus, while we train with depth information, we can test with only RGB as well, which is a major advantage. Intuitively, this strategy is equivalent to sharing weights between the RGB and D spaces, in order to exploit the coherence between RGB and D; thus combining the advantages from both the spaces.

Loss function. We use Voxel-wise Cross Entropy to train the reconstruction models. It is the sum of the cross-entropies for each pair of predicted and target voxel values. Let 'p' be the predicted value at voxel (i, j, k) and 'y' the corresponding expected value. Then, the loss is defined as :

$$L(p, y) = \sum_{i,j,k} y_{(i,j,k)} \log(p_{(i,j,k)}) + (1 - y_{(i,j,k)}) \log(1 - p_{(i,j,k)}) \tag{4.1}$$

4.2.2 Texture recovery

We obtain a set of four orthographic depth maps (excluding top and bottom views) of the 3D human mesh M generated by the reconstruction network and represent them as $D = \{\pi^o_{D,i}\}$. Here $\pi^o_{D,i}$ corresponds to the orthographic depth projection of M to the i^{th} face of the cube. The 3D model M can either be the ground truth 3D model or the reconstructed one as stated in the earlier section. We also have a set of perspective RGB images used in reconstruction $P = \pi^P_{RGB,j}$ from which we choose a random image. Given each of the orthographic depth image and perspective image, texture recovery aims to estimate the color of each pixel in the depth map.

In order to achieve this, we propose a simple texture recovery approach that detaches the requirement of calibrated setup by employing a variational auto-encoder (VAE). The VAE is trained to learn the distribution $p(\pi^O_{RGB,i}|\pi^O_{D,i}, \pi^P_{RGB,j})$ which models the color of provided orthographic depth. The data is modeled by normalizing out Z from the joint probability distribution $p(\pi^O_{RGB,i}, z)$. $p(z)$ is inferred using $p_\theta(z|\pi^O_{D,i}, \pi^P_{RGB,j})$ which is modeled by encoder parameterized by θ. Variational distribution q_ϕ is introduced to approximate the unknown true posterior p_θ. Variational auto-encoder is trained by maximizing the log likelihood $log(p_\theta(\pi^O_{RGB,i}|\pi^O_{D,i}, \pi^P_{RGB,j})$ which is equivalent to minimizing variational lower bound:

$$\begin{aligned} V(\pi^O_{RGB,i}, \pi^O_{D,i}, \pi^P_{RGB,j}; \theta, \phi) = &-KL(q_\phi(z|\pi^O_{D,i}, \pi^P_{RGB,j})||p_\theta(z)) \\ &+ E_{q_\phi}[logp_\theta(\pi^O_{RGB,i})|\pi^O_{D,i}, \pi^P_{RGB,j}, z] \end{aligned} \tag{4.2}$$

The second term in equation 2 is log likelihood of samples which is simply L2 reconstruction loss which encourages consistency between encoder and decoder. The KL divergence reduces the distance between the variational distribution and prior distribution. $q_\phi(z|x)$ is associated with a prior distribution over the latent variables for which we follow multivariate Gaussian with unit variance $\mathcal{N}(0, I)$. To the best of our knowledge, this is the first attempt to use depth cues for novel view synthesis.

Network architecture. As shown in Figure 4.1 encoder consists of two symmetric branches one each

for orthographic depth and perspective RGB image as input. However, the weights are not shared between the two encoders. These networks consist of convolutional layers which have 64,128,256,256,512 and 1024 channels with filters of size 5,5,3,3,3 and 4 respectively followed by a fully connected layer with 1024 neurons. The representations from two branches combined to form 1024 dimensional latent variable. The decoder network consist of a fully connected layer with $256 \times 8 \times 8$ neurons. Deconvolution layers are then followed with 2×2 upsampling and has 256, 256, 256, 128, 64 and 3 channels. The filter sizes are 5×5 for all layers. ReLU is used as activation. The output size is set to 64×64

We have used architecture of SRGAN [80]. The network trained on MS COCO dataset is finetuned with the output of variational auto-encoder. The size of high resolution image is set to 256×256.

4.3 Experiments & results

4.3.1 Datasets

MPI dataset: First, we use the parametric SMPL model [16, 145] to generate 100 mesh sequences, each containing 300 frames, i.e., 3000 meshes, consisting of an equal number of male and female models. Additionally, we use FAUST data [19] which consists of 300 high resolution human meshes.There are a total of 10 different subjects in 30 different poses. The 300 meshes come divided in 2 sequences, one having complete meshes and the other having broken/incomplete parts - the former used for training, and the latter for testing. In addition, we simulated a virtual Kinect setup to capture aligned RGBD. Since both these datasets had correspondences, a few meshes were manually textured, and this texture was transferred across each of the datasets.

MIT's articulated mesh animation [152]: This dataset consists of 5 mesh sequences (approx. 175 to 250 frames each). It provides RGB images from 8 views corresponding 3D meshes for each frame. The total number of meshes used from this dataset for training is 1,525.

Our data: A total of 5 mesh sequences, each containing 200 to 300 frames with significant pose and shape variation were captured using a calibrated multi-Kinect setup. The colored point clouds were re-meshed using Poisson's surface reconstruction algorithm, after pre-processing them for noise removal. The processed RGBD and corresponding textured models were used for training the pipeline.

4.3.2 Non-rigid reconstruction

Network's Training: We used Nvidia's GTX 1080Ti, with 11GB of VRAM to train our models. A batch size of 5 with the ADAM optimizer having an initial learning rate of 10^{-4}, and a weight decay of 10^{-5} is used to get optimal performance. Further, a standard $80 : 20$ split between the training and testing datasets is adhered to. In order to ensure that reconstruction is feasible from single as well as multiple views, we choose random number of views from available views for training a mesh in each iteration. Using this randomization in training, we are providing sufficient view information to the network so that it can learn the space of body pose and shape variations and hence able to achieve single

Figure 4.2 3D Shapes obtained with our reconstruction network (top row) compared to ground truth models (bottom row) obtained with MVG.

Dataset	Multi-View Module	RGB (Baseline)	D	RGBD	RGB/D
MPI-SMPL [145]	3D-GRU	0.6903	0.7709	0.7541	**0.8040**
	Max Pool	0.7144	0.7633	0.7550	**0.7816**
MIT [152]	3D-GRU	0.0103	0.7403	-	**0.7547**
	Max Pool	0.0081	0.7205	-	**0.7480**
MPI-FAUST [21]	3D-GRU	0.8113	0.8629	0.8356	**0.8644**
	Max Pool	0.8150	**0.8661**	0.8366	0.8521
OUR DATA	3D-GRU	0.6816	0.7963	0.8114	**0.8241**
	Max Pool	0.6883	0.7844	0.8017	**0.8066**

Table 4.1 A comparison of IoU values tested using a single view on datasets [152, 145, 21], under the various input modes, when trained with two different view modules.

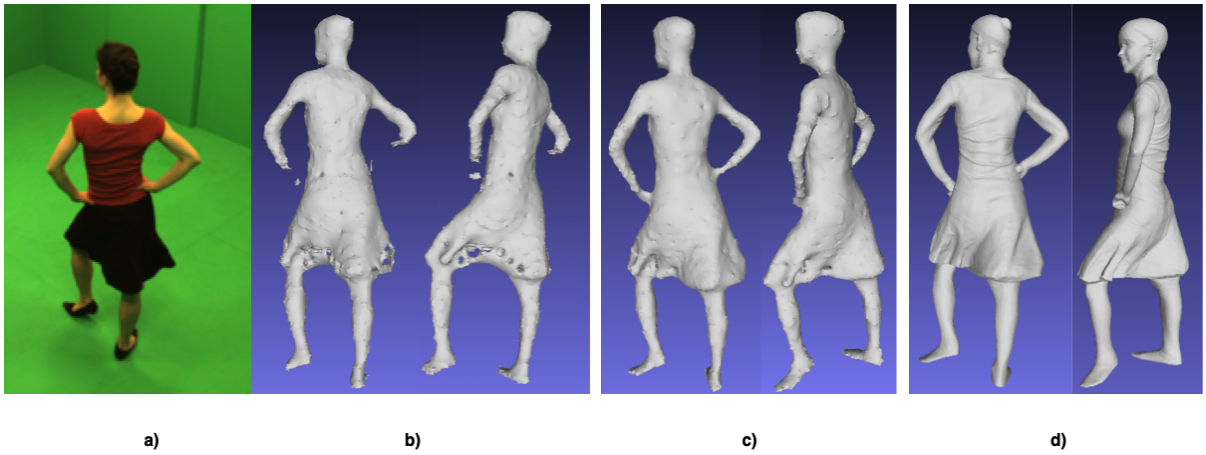

Figure 4.3 Comparison of shape reconstruction (two views shown) (a) Input Image (b) VRN (c) Ours (d) Ground truth mesh

view reconstruction at test time.

Evaluation metric: The primary metric used to evaluate our performance is the Intersection over Union (IoU), which is a comparison between the area of overlap and the total area encompassing both the objects. Larger its value, the better the quality of reconstruction.

Let 'p' be the predicted value at voxel (i, j, k) and 'y' the corresponding expected value. 'I' is an indicator function which gives gives a value of 1 if the expression it is evaluating is true, if not, it gives 0. 't' is an empirically decided threshold of 0.5 above which the cell is considered as filled.

$$IoU = \frac{\sum_{i,j,k}[I(p(i,j,k) > t)I(y(i,j,k))]}{\sum_{i,j,k}[I(p(i,j,k) > t) + I(y(i,j,k))]} \tag{4.3}$$

4.4 Comparison

Results & discussion: Quantitative results (IOU metric) in Table 4.1 suggests that for variety of datasets of varying complexity and irrespective of the method of combining multiple views, the depth information is very critical for accurate reconstruction of human models. It is interesting to notice that the difference in IoU values between RGB and RGB/D widens under two scenarios - a) when the dataset has very complicated poses (such as the handstand sequence in MIT) and b) when the background becomes more complicated. An intuition behind the working of this training paradigm is that the co-learning of the shared filter weights of the two modalities act as a regularization for one another, thus enhancing the information seen by the network. In Figure 4.3, we show the performance of our method

with voxel regression network (VRN) [32]. It can be observed that our network predicts plausible body parts and clothing deformation.

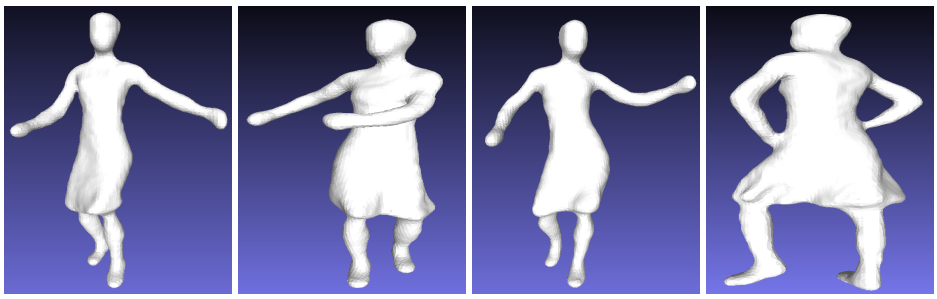

Figure 4.4 Clothing induced deformations captured by our proposed method on [152].

Figure 4.2 shows the robustness of the learned model performing a vast range of actions. As a result of not imposing any body model constraint, we were able to partially handle non-rigid deformations induced by free form clothing as shown is Figure 4.4.

4.4.1 Texture recovery

The textured model is obtained by synthesizing a texture orthographic view image corresponding to respective orthographic depth image taken from volumetric model predicted by reconstruction module. Figure 4.6 shows the generated orthographic texture image and corresponding textured model obtained after their back-projection on reconstructed 3D model on samples from various datasets. One can infer that our single view textured reconstruction performs relatively well given the ill posed setup of recovering both texture and 3D from single image. Nevertheless, further post-processing like super-resolution networks can be used to improve the resolution of synthesized texture images as described below in subsection 4.4.2 for high quality texture recovery.

4.4.2 Upsampling

Variational auto-encoder (VAE) generates a low resolution 64*64 image, $\pi^{O}_{RGB,i,LR}$ of the corresponding ground truth colored orthographic image $I^{O}_{RGB,i}$. This is because VAE cannot generate for high resolution images. Hence, we resort to another network which upsamples the output produced by VAE. We train another generator in GAN setup which maps the low resolution image to high resolution. Following SRGAN [80], generator G is trained to minimize $L2$ loss along with adversarial loss and the discriminator tries to counterfeit the generator by reducing adversarial loss. In Figure 4.5, we show the effect of applying bilinearly upsampling the VAE output (a) to 256×256 resolution (b) and GAN

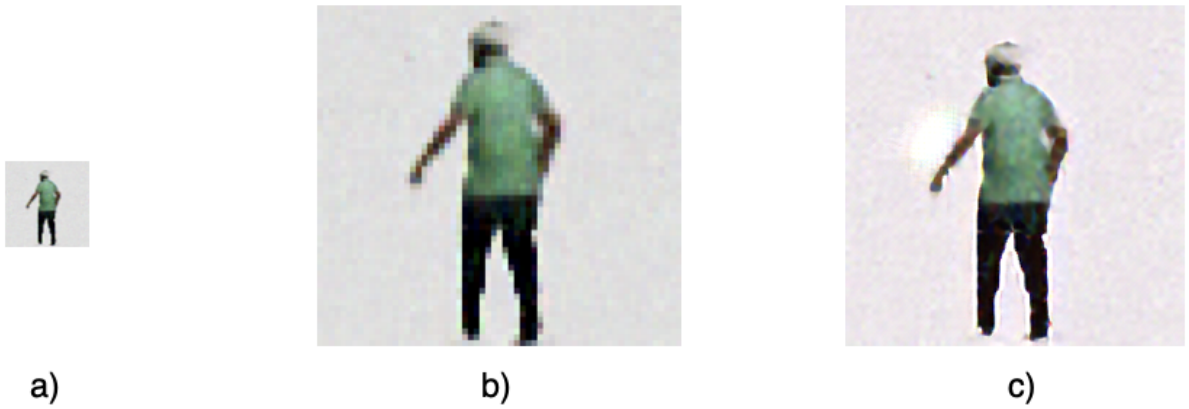

Figure 4.5 (a) Output of VAE, (b) Bilinearly Upsampled Image (c) GAN generated Image

upsampled image (c). We can observe that GAN has restored significant values.

$$\begin{aligned} L_G &= E_{z\ p(z)}[log(1 - D(I^O_{RGB,i}), G(\pi^O_{RGB,i,LR}))] \\ &\quad + \beta \| I^O_{RGB,i} - \pi^O_{RGB,i,HR} \|_2 \\ L_D &= E_{z\ p(z)}[log(1 - D(\pi^O_{RGB,i,HR}))] \\ &\quad + E_{I^O_{RGB,i}}[log(D(I^O_{RGB,i}))] \end{aligned} \tag{4.4}$$

4.5 Summary

We proposed a novel deep learning pipeline for reconstructing textured 3D models of non-rigid human body shapes using single view RGB images (during test time). This is a severely ill posed problem due to self-occlusions caused by complex body poses and shapes, clothing obstructions, lack of surface texture, background clutter, sparse set of cameras with non-overlapping fields of view, etc. We showed superior reconstruction performance using the proposed method in terms of quantitative and qualitative results on both publicly available datasets (by simulating the depth channel with virtual Kinect) as well as real RGBD data collected with calibrated multi Kinect setup.

The approach presented in this chapter disentangles 3D body reconstruction and texture using two separate networks. A part of this chapter especially non-rigid reconstruction is also claimed by my colleague in [147] which is a joint work with equal contribution. For extracting surface, we utilized volumetric representation. However, volumetric representation poses severe computational disadvantage. This problem arises because deep neural network probes every location in volumetric grid which results in wastage of computation. Additionally, 3D convolution is also done in inside the surface which leads to redundant computation. Moreover, texture from VAE produces 64×64 resolution which is later

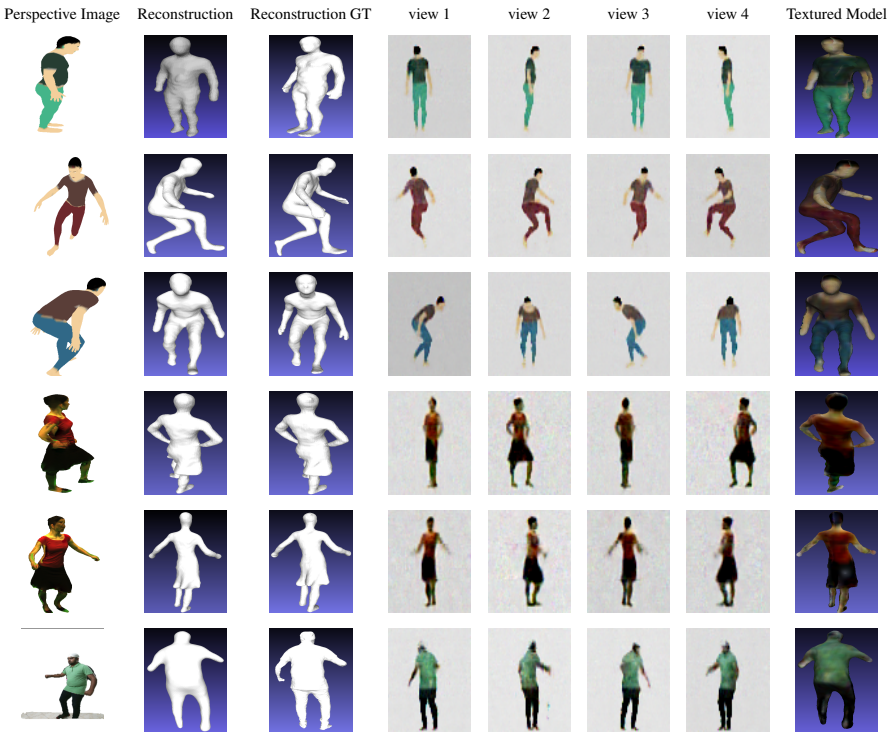

Figure 4.6 Results on reconstruction and texture recovery on [152, 16] and on our real data.

upsampled using SRGAN. Thus, to achieve textured 3D model we need three networks. Hence, a new representation which encodes surface geometry and texture efficiently is desired.

Chapter 5

PeeledHuman: Robust Shape Representation for Textured 3D Human Body Reconstruction

In previous chapters, we discussed the drawbacks of existing representations and networks to predict these representations. In this chapter, we introduce PeeledHuman - a novel shape representation of the human body that is robust to self-occlusions. PeeledHuman encodes the human body as a set of Peeled Depth and RGB maps in 2D, obtained by performing ray-tracing on the 3D body model and extending each ray beyond its first intersection. This formulation allows us to handle self-occlusions efficiently compared to other representations. Given a monocular RGB image, we learn these Peeled maps in an end-to-end generative adversarial fashion using our novel framework - PeelGAN. We train PeelGAN using a 3D Chamfer loss and other 2D losses to generate multiple depth values per-pixel and a corresponding RGB field per-vertex in a dual-branch setup. In our simple non-parametric solution, the generated Peeled Depth maps are back-projected to 3D space to obtain a complete textured 3D shape. The corresponding RGB maps provide vertex-level texture details. We compare our method with current parametric and non-parametric methods in 3D reconstruction and find that we achieve state-of-the-art-results. We demonstrate the effectiveness of our representation on publicly available BUFF and MonoPerfCap datasets as well as loose clothing data collected by our calibrated multi-Kinect setup.

5.1 Introduction

Recent advancements in deep learning have renewed interest with the focus on a more challenging variant of the problem: a fast and robust monocular 3D reconstruction. Existing deep-learning solutions for *monocular* 3D human reconstruction can be broadly categorized into two classes. The first class of model-based approaches (e.g., [64, 104]) attempt to fit a parametric body representation, like the SMPL [90, 110], to recover the 3D surface model. Such model-based methods efficiently approximate the shape and pose of the underlying naked body but fail to reconstruct fine surface texture details of the body and the wrapped clothing. Parametric SMPL models have been extended to include clothing

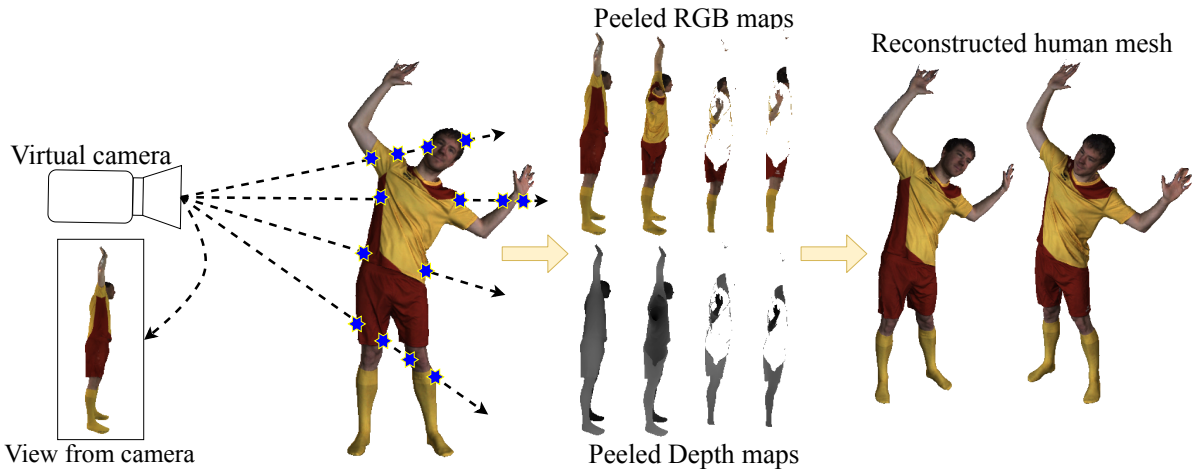

Figure 5.1 PeeledHuman. Our proposed representation encodes a human body as a set of *Peeled Depth & RGB maps* from a given view. These maps are back-projected to 3D space in the camera coordinate frame to recover the 3D human body.

details like in [109, 4]. Another approach by [5] predicts a UV map for every foreground pixel to generate texture over a SMPL model. However, it does not account for large clothing deformations.

The second class of model-free approaches does not assume any parametric model of the body. One set of model-free approaches employ volumetric regression, a natural extension of 2D convolutions, for human body recovery from a monocular image [144, 149]. However, volumetric regression is known to be memory intensive and computationally inefficient as it involves redundant 3D convolutions on empty voxels. Additionally, this memory-intensive behavior restricts the ability to learn detailed surface geometry.

The recent works in this direction include MouldingNet [41], PIFu [122] and its follow-up work PIFuHD [123]. PIFu proposes a deep network that learns an implicit function to recover 3D human models under loose clothing. More precisely, they compute local per-pixel feature vectors on an inference image and a specified z-depth along the outgoing camera ray from each pixel to learn an implicit function that can classify whether a 3D point corresponding to this z-depth is inside or outside the body surface. However, this requires sampling multiple 3D points from the canonical 3D volume and testing for each of them independently. Such sampling adds redundancy at inference time as a large number of points inside as well as outside the 3D body surface are tested. Instead, identifying the 3D points on the surface is more efficient for recovering the surface geometry. On the other hand, MouldingNet [41] proposes to recover 3D body models by performing a pixel-wise regression of two *independent* depth maps (visible and hidden). This is similar to generating depth maps captured by two RGBD virtual cameras separated by 180° along z-axis. Although such pixel-wise regression is computationally more efficient as compared to PIFu and can model arbitrary surface topology, it still fails to handle self-occlusions.

To summarize, model-based methods cannot reconstruct highly textured clothed subjects with arbitrary shape topologies. On the other hand, existing model-free approaches are either computationally intensive or unable to handle large self-occlusions.

In this chapter, we tackle the problem of textured 3D human reconstruction from a single RGB image by introducing a novel shape representation, shown in Figure 5.1. Our proposed solution derives inspiration from the classical ray tracing approach in computer graphics. We estimate a fixed number of ray intersection points with the human body surface in the canonical view volume for every pixel in an image, yielding a multi-layered shape representation called *PeeledHuman*. PeeledHuman encodes a 3D shape as a set of depth maps called hereinafter as *Peeled Depth maps*. We further extend this layered representation to recover texture by capturing a discrete sampling of the continuous surface texture called hereinafter as *Peeled RGB maps*. Such a layered representation of the body shape addresses severe self-occlusions caused by complex body poses and viewpoint variations. Our representation is similar to *depth peeling* used in computer graphics for order-independent transparency. The proposed shape representation allows us to recover multiple 3D points that project to the same pixel in 2D image plane (see Figure 5.1), thereby overcoming the limitation handling self-occlusions in MouldingNet. This solution is also more efficient than PIFu at both training and inference time as it simultaneously (globally) predicts and regresses to a fixed set of Peeled Depth & RGB maps for an input monocular image. It is important to note that our representation is not restricted only for human body models but can generalize well to any 3D shapes/scenes, given specific training data prior.

Thus, we reformulate the solution to the monocular textured 3D body reconstruction task as predicting a set of Peeled Depth & RGB maps. To achieve this dual-prediction task, we propose PeelGAN, a dual-task generative adversarial network that generates a set of depth and RGB maps in two different branches of the network, as shown in Figure 5.2. These predicted peeled maps are then back-projected to 3D space to obtain a point cloud. Similar to [154], we propose to include Chamfer loss over the reconstructed point cloud in the camera coordinate frame. This loss implicitly imposes a 3D body shape regularization during training. Our model is able to hallucinate plausible parts of the body that are self-occluded in the image. As compared to PIFu and MouldingNet, PeelGAN has the advantage of being computationally efficient while handling severe self-occlusions and arbitrary surface topology deformations caused by loose clothing. Our proposed representation enables an end-to-end, non-parametric and differentiable solution for textured 3D body reconstruction.

We evaluate our method with prior work on public datasets such as BUFF [170] and MonoPerfCap [164]. MonoPerfCap consists of articulated skeletal motions and medium-scale non-rigid surface deformations by deforming a template mesh. Hence, loose clothing and large scale non-rigid deformations are not included. On the other hand, BUFF sequences are noisy with limited variations in shape and clothing. To compensate for the lack of realistic 3D datasets with large variations in shape and clothing, we present a challenging 3D dataset captured from our calibrated multi-Kinect setup. It consists of 8 subjects with large variations in loose clothing and shape (see section 7.2). We evaluate our method on

all three datasets and report superior quantitative and qualitative results to other state-of-the-art methods. To summarize our contributions in this paper:

- We introduce PeeledHuman - a novel shape representation of the human body encoded as a set of Peeled Depth and RGB maps, that is robust to severe self-occlusions.
- Our proposed representation is efficient in terms of both encoding 3D shapes as well as feed-forward time yielding superior quality of reconstructions with faster inference rates.
- We propose PeelGAN - a complete end-to-end pipeline to reconstruct a textured 3D human body from a single RGB image using an adversarial approach.
- We introduce a challenging 3D dataset consisting of multiple human action sequences with variations in shape and pose, draped in loose clothing. We intend to release this data along with our code for academic use.

5.2 Related work

Traditionally, voxel carving and triangulation methods were employed for recovering a 3D human body from calibrated multi-camera setups [37, 22]. Majority of existing deep learning methods to recover 3D shapes from monocular RGB images use parametric SMPL [90] model. HMR [64] proposes to regress SMPL parameters while minimizing re-projection loss. Segmentation masks [146] were used to further improve the fitting of the 3D model to the available 2D image. However, these parametric body estimation methods yield a smooth naked mesh missing out on surface geometry details. Additionally, researchers have explored to incorporate tight clothing details over SMPL model by estimating displacements of each vertex [14, 2]. Very recently, clothing deformation is predicted as a function of garment size [138] and in [150] estimate vertex displacements by regressing to SMPL vertices. These techniques fail for complex clothing topologies such as skirts and dresses.

On the other hand, model-free approaches do not use any parametric model. Volumetric regression [144, 149, 54] uses a voxel grid, i.e., a binary occupancy map to recover the human body from a single RGB image. Volumetric representations pose a serious computational disadvantage due to the sparsity of the voxel grid and surface quality is limited to the voxel grid resolution. Deformation based approaches have been proposed over parametric models which incorporate these details to an extent. The constraints from body joints, silhouettes, and per-pixel shading information are utilized in [176] to produce per-vertex movements away from the SMPL model. However, only the visible pixels are modelled in this approach.

To address the aforementioned issues during reconstruction of 3D human bodies, interest has garnered around non-parametric approaches recently. Deep generative models have been proposed in [102] taking inspiration from the visual hull algorithm to synthesize 2D silhouettes that are back-projected from inferred 3D joints. The silhouettes are back-projected to obtain clothed models with different

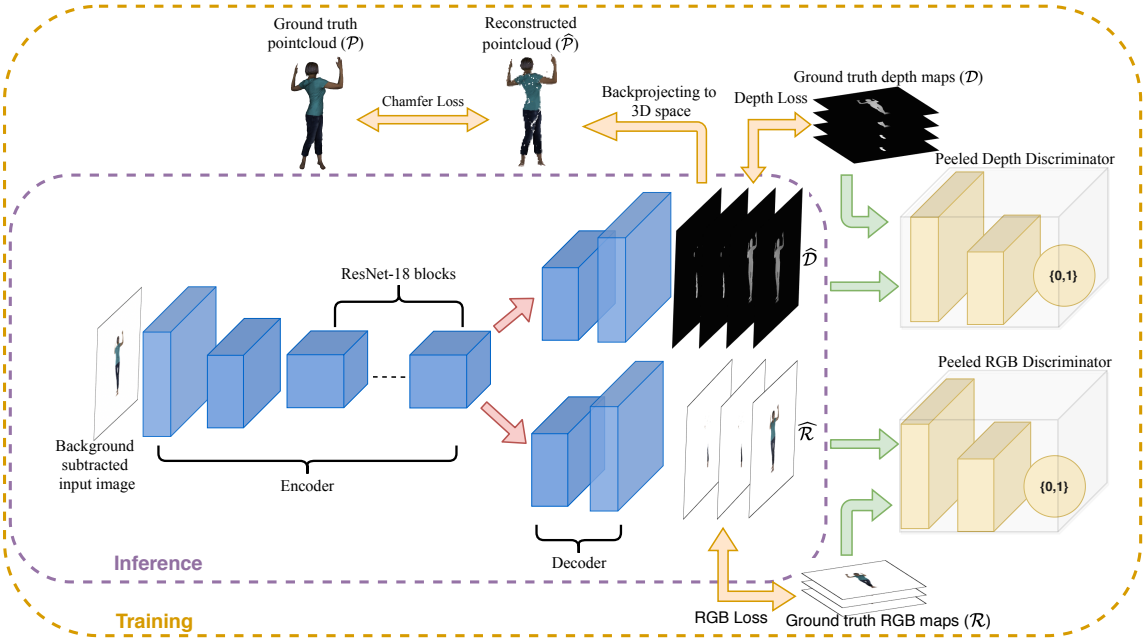

Figure 5.2 PeelGAN overview: The dual-branch network generates Peeled Depth ($\widehat{\mathcal{D}}$) and RGB ($\widehat{\mathcal{R}}$) maps from an input image. The generated maps are each fed to a discriminator: one for RGB and one for Depth maps. The generated maps are back-projected to obtain the 3D human body represented as a point cloud ($\widehat{\mathcal{P}}$) in the camera coordinate frame. We employ a Chamfer loss between the reconstructed point cloud and the ground-truth point cloud ($\mathcal{P}$) along with several other 2D losses on the Peeled maps, as listed in subsection 5.3.3.

shape complexities. Implicit representations of 3D objects have been employed for deep learning based approaches in [98, 122, 123, 84, 12, 51, 31] which represent the 3D surface as the continuous decision boundary of a deep neural network classifier. PIFu has been extended to animate implicit representation in [55]. Unsupervised estimation of implicit functions have been addressed in [88, 103]. Authors in [41] represent the human body as a mould and recover visible and hidden depth maps. Self-occlusions are not handled by these approaches as they do not impose any human body shape prior.

Similar to our peeled representation, multi-layer approaches have been used for 3D scene understanding. Layered Depth Images were proposed in [126] for efficient rendering applications. Layer-structured 3D scene representation was proposed in [141] which performs view synthesis as a proxy task. Recently, transformer networks were proposed in [130] to transfer features to a novel view to better recover 3D scene geometry. Nested shape layer representation was introduced in [118] to encode a 3D object efficiently.

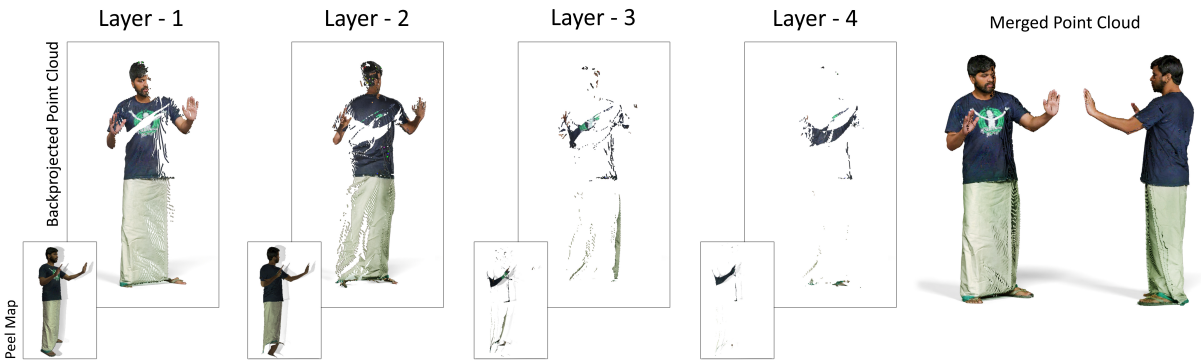

Figure 5.3 Back-projecting peel maps to generate point cloud.

5.3 Proposed method

5.3.1 Peeled representation

We encode a 3D human body model as a set of Peeled Depth & RGB maps as follows. We assume the human body to be a non-convex object placed in a virtual scene. Given a virtual camera, a set of rays originating from the camera center are traced through each pixel to the 3D world. The set of first ray-intersections with the 3D body are recorded as depth map d_1 and RGB map r_1, capturing visible surface details that are nearest to the camera. Subsequently, we *peel* away the occlusion and extend the rays beyond the first bounce to hit the next intersecting surface. We successively record the corresponding depth and RGB values of the next layer as d_i and r_i, respectively. We consider 4 intersections of each ray , 4 Peeled Depth & RGB maps to faithfully reconstruct a human body assuming this can handle self-occlusions caused by the most frequent body poses. The Peeled depth and RGB maps are back projected to obtain colored point cloud.

5.3.2 Back-projection

Here, we explain the term ”back-projection”. The PeeledHuman representation encodes human mesh into multi-layered depth and RGB peel maps. These peel maps are then back-projected to get the point cloud for each layer, i.e., each pixel in the depth peel map is converted to a 3D point by multiplying it with the inverse of the camera matrix. The estimated 3D point is then assigned a color from the corresponding pixel in the RGB peel maps. This operation is performed for all the pixels to get the point cloud per layer. The final merged point cloud is the union of these per-layer point clouds, as shown in Figure 5.3. A point cloud can be constructed from these maps using classical camera projection methods. If the camera intrinsics, , the focal length of camera $f = [f_x, f_y]$ and its center of axes $C = [C_x, C_y]$ are

known, then the ray direction in the camera coordinate frame corresponding to pixel $[X, Y]$ is given as

$$ray[X, Y] = \left(\frac{X - C_x}{f_x}, \frac{Y - C_y}{f_y}, 1 \right). \tag{5.1}$$

For a pixel $[X, Y]$ with depth d_1^{XY} in the first depth map, its 3D location in the camera coordinate frame is given by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{X_{norm} \cdot d_1^{XY}}{fx} \\ \frac{Y_{norm} \cdot d_1^{XY}}{fy} \\ d_1^{XY} \end{bmatrix}, \tag{5.2}$$

where $X_{norm} = X - h/2$ and $Y_{norm} = Y - w/2$. Here, we assume $[h/2, w/2]$ is the center of the image.

Problem formulation Given an RGB image r_1 of resolution $(h \times w \times 3)$ captured from an arbitrary viewpoint, our goal is to reconstruct a textured 3D body model from n Peeled Depth maps ($\widehat{\mathcal{D}}$) and $n-1$ Peeled RGB maps ($\widehat{\mathcal{R}}$) where $\widehat{\mathcal{D}} = \{\hat{d}_1, \hat{d}_2, \cdots, \hat{d}_n\}$ and $\widehat{\mathcal{R}} = \{\hat{r}_2, \hat{r}_3, \cdots, \hat{r}_{n-1}\}$ respectively. The ground-truth maps are denoted as $\mathcal{D} = \{d_1, d_2, \cdots, d_n\}$ and $\mathcal{R} = \{r_1, r_2, \cdots, r_n\}$. A reconstructed point cloud $\widehat{\mathcal{P}}$ is obtained using Equation 5.2. Background pixels have zero depth values and are not part of $\widehat{\mathcal{P}}$. The ground-truth point cloud $\mathcal{P}$ is used as 3D supervision in Equation 5.7. We do not generate $\hat{r}_1$ as the input image r_1 can be considered as the first generated RGB map.

5.3.3 PeelGAN

To generate Peeled maps from an input image, we propose a conditional GAN, named PeelGAN, as depicted in Figure 5.2. PeelGAN takes a single RGB image as its input and generates Peeled Depth maps $\widehat{\mathcal{D}}$ and corresponding RGB maps $\widehat{\mathcal{R}}$ (refer to subsection 5.3.1). The input RGB image is first fed to an encoder network (similar to [56]) consisting of a few convolutional layers for recovering $128 \times 128 \times 256$ feature maps and is subsequently fed to a series of 18 ResNet [50] blocks. The network uses ReLU as its activation function. We propose to decode the Peeled Depth and RGB maps in two separate branches since they are sampled from different distributions. The network produces 3 Peeled RGB maps and 4 Peeled Depth maps which are then separately fed to two different discriminators, one for each RGB and Depth maps. We use PatchGAN discriminator as proposed in [56]. We denote our generator as G, the Peeled RGB map discriminator as D_r and the Peeled depth map discriminator as D_d. We train our network with the following loss function:

$$L_{peel} = L_{gan} + \lambda_{depth} L_{depth} + \lambda_{rgb} L_{rgb} + \lambda_{cham} L_{cham} + \lambda_{smooth} L_{smooth}, \tag{5.3}$$

where λ_{depth}, λ_{rgb}, λ_{cham}, λ_{smooth} are weights for depth loss(L_{depth}), RGB loss(L_{rgb}), Chamfer loss(L_{cham}) and smoothness loss(L_{smooth}) respectively. Each loss term is explained in detail below.

GAN loss (L_{gan}) We follow the usual GAN objective for the generated $\widehat{\mathcal{R}}$ and $\widehat{\mathcal{D}}$ maps conditioned on the input image r_0 as

$$L_{gan} = E_{r_0,\mathcal{R}}[logD_r(r_0,\mathcal{R})]+E_{r_0,\mathcal{D}}[logD_d(r_0,\mathcal{D})]+E_{r_0}[log(1-D_r(r_0,\widehat{\mathcal{R}}))]+E_{r_0}[log(1-D_d(r_0,\widehat{\mathcal{D}}))]. \tag{5.4}$$

Depth loss (L_{depth}) We minimize the masked L1 loss over ground-truth and generated peeled depth maps as

$$L_{depth} = \sum_{i=1}^{4} \left\| m_i \cdot (d_i - \hat{d}_i) \right\|_1, \tag{5.5}$$

where $m_i = \gamma$ for occluded pixels and $m_i = 1$ otherwise. Occluded pixels are the ones which have multiple non-zero depth values across the four Peeled Depth maps.

RGB loss (L_{rgb}) The generator minimizes L1 loss between the ground-truth and generated peeled RGB maps.

$$L_{rgb} = \sum_{i=2}^{4} \left\| (r_i - \hat{r}_i) \right\|_1. \tag{5.6}$$

Chamfer loss (L_{cham}) To enable the network to capture the underlying 3D structure of the generated depth maps, we minimize Chamfer distance between the reconstructed point cloud ($\widehat{\mathcal{P}}$) and the ground-truth point cloud ($\mathcal{P}$),

$$L_{cham}(\widehat{\mathcal{P}},\mathcal{P}) = \sum_{\vec{p_i}\in\widehat{\mathcal{P}}} \min_{\vec{q_j}\in\mathcal{P}} \|\vec{p_i} - \vec{q_j}\|_2^2 + \sum_{\vec{q_j}\in\mathcal{P}} \min_{\vec{p_i}\in\widehat{\mathcal{P}}} \|\vec{q_j} - \vec{p_i}\|_2^2. \tag{5.7}$$

Chamfer loss induces 3D supervision by fusing multiple independent 2.5D generated peel depth maps. Refer subsubsection 5.4.4.1 for evaluation of Chamfer loss.

Smoothness loss (L_{smooth}) There is additional need to enforce smoothness in depth variations over the surface (except for the boundary regions). Thus, motivated by [137], we enforce the first derivative of generated Peeled Depth maps to be close to that of the ground-truth Peeled Depth maps as

$$L_{smooth} = \sum_{i=1}^{4} \left\| \bigtriangledown d_i - \bigtriangledown \hat{d}_i \right\|_1 \tag{5.8}$$

5.4 Experiments

We implement our proposed pipeline in PyTorch using 4 Nvidia GTX 1080 Ti GPUs with 11GB RAM trained for 45 epochs. A batch size of 12 is used for 512×512 images. Ground-truth Peeled maps are captured using trimesh [1]. We use the Adam optimizer with a learning rate of $1.5e$-4 and γ, λ_{dep}, λ_{cham}, λ_{rgb} and λ_{smooth} as 10, 100, 500, 500, 500, respectively. The predicted point cloud contains 30000 3D body surface points on average.

5.4.1 Datasets and pre-processing

We perform qualitative and quantitative evaluation on three datasets, namely (i) BUFF [170] (ii) MonoPerfCap [164] (iii) Our new dataset. We scale each 3D body model to a unit-box and compute 4 Peeled Depth and RGB maps from 4 different camera angles each: 0° (canonical view), 45°, 60°, 90°.

BUFF dataset consists of 5 subjects with tight and loose clothing performing complex motions. The dataset consists of 11,054 3D human body models in total. We use this completely for testing our method.

MonoPerfCap dataset consists of 13 daily human motion sequences in tight and loose clothing styles. It has approximately 40,000 3D human body models with subjects in indoor and outdoor settings. We use two sequences for inference and six sequences for training. One sequence is divided equally between training and inference.

Our data We introduce a 3D dataset consisting 2,000 human body models from 8 human action sequences including marching and swinging limbs using a calibrated setup of 4 Kinect sensors. The RGBD data is back-projected to obtain a point cloud and post-processed using Poisson surface reconstruction to obtain the corresponding meshes. As our data is independently reconstructed in each frame without any template constraint, we were able to capture realistic large scale deformations. The dataset contains significant variations in shape and clothing consisting of both loose and tight clothing as shown in Figure 5.4. We use six sequences for training and two sequences for inference. The dataset will be released for academic purposes to spur further research in this field.

5.4.2 Qualitative results

We demonstrate single-view/monocular reconstruction results on all 3 datasets in Figure 5.5 and Figure 5.6. Our method is able to accurately recover the 3D human shape from previously unseen views. Due to the nature of our encoding, our method is able to recover the self-occluded body parts reasonably well for severely occluded views.

[1] www.trimesh.org

Figure 5.4 Our Dataset captured from our calibrated Kinect setup with variations in clothing, shape and pose.

5.4.3 Comparison with prior work

We perform qualitative comparison of our proposed representation with other commonly used representations for single-view 3D human reconstruction. In particular, first we compare our method with parametric body model regression (meshes) and implicit function learning methods in Figure 5.7 as well as, with voxel regression and point cloud regression method in **??**. We retrain PIFu [122] using MonoPerfCap and our dataset. We also evaluate PIFu after finetuning the model provided by authors with MonoPerfCap and our dataset. We compare with HMR [64] as a parametric model regression (mesh-based) method. To compare against MouldingNet [41] in Figure 5.8, we train PeelGAN with two depth maps and our own specifications as neither code nor data was made public by the authors. For voxel-based method, we train our PeelGAN model and DeepHuman [174] (predicts only textureless models) using the released THUman dataset [174] shown in Figure 5.9.

As demonstrated in Figure 5.7, our proposed method consistently recovers the underlying shape and texture. When trained from scratch, PIFu fails to recover shape but finetuning the pre-trained model (trained on commercial high resolution meshes) results in lesser artifacts. This emphasizes the necessity of high resolution data to train implicit function approaches. Moreover, PIFu is not end-to-end trainable since it requires training shape and color components separately. HMR produces a smooth naked body mesh missing surface texture details. MouldingNet fails to recover body shape when there is significant self-occlusion in the input image, as shown in Figure 5.8. Figure 5.9 shows that our method is able to recover plausible human shapes even when it is challenging to distinguish body parts from a single-view (here hand is indistinguishable from torso due to texture-less dark shade clothing). Quantitative evaluation of our method using Chamfer distance against PIFu, BodyNet [144], SiCloPe [102] and VRN [58] is shown in Table 5.1. Here we report results on both 256 resolution input along with 512

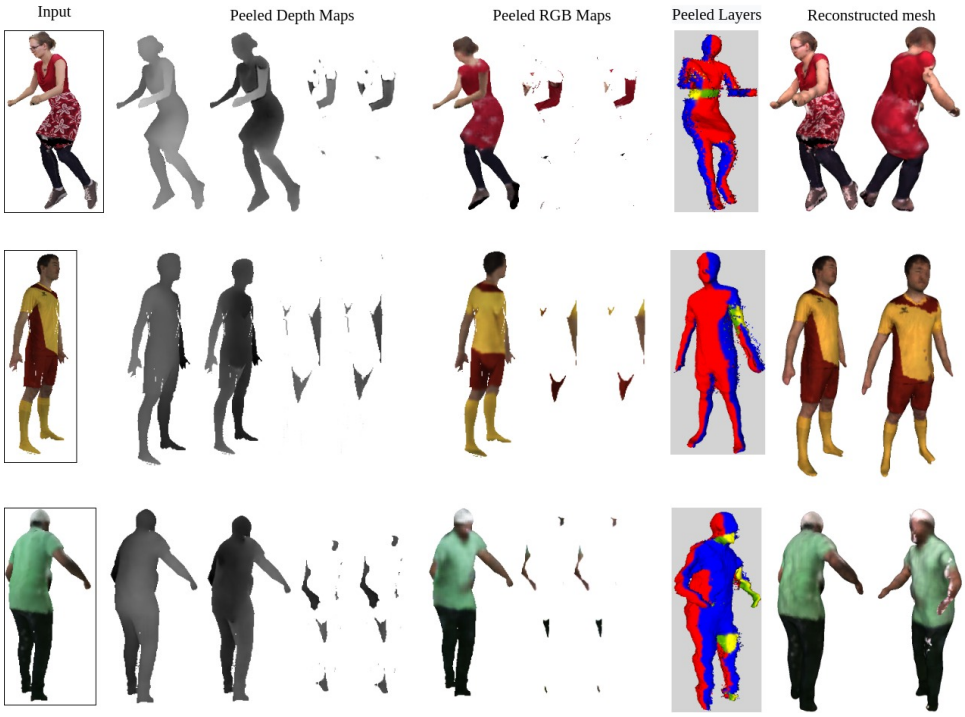

Figure 5.5 Qualitative results on MonoPerfCap (Top row), BUFF (Middle row) and Our Dataset (Bottom row). For each subject, we show (from left to right) input image, 4 Peeled Depth and RGB maps, backprojected Peeled layers (colored according to their depth order : red, blue, green and yellow, respectively), reconstructed textured mesh. Please refer to the supplementary material for extended set of results.

Method	Chamfer Distance ↓	Image Resolution
BodyNet [144]	4.52	256
SiCloPe [102]	4.02	256
VRN [58]	2.48	256
PIFu [122]	1.14	512
Ours	1.283	256
Ours	**0.9254**	**512**

Table 5.1 Quantitative comparison with other methods. Our method achieves the lowest Chamfer score for single-view reconstruction, indicating the robustness of our representation.

Figure 5.6 Qualitative textured reconstruction results on MonoPerfCap and BUFF datasets. For each subject, we show the input image and multiple views of the reconstructed mesh (after performing Poisson surface reconstruction on the reconstructed point cloud). Our proposed PeeledHuman representation efficiently reconstructs the occluded parts of the body from a single view.

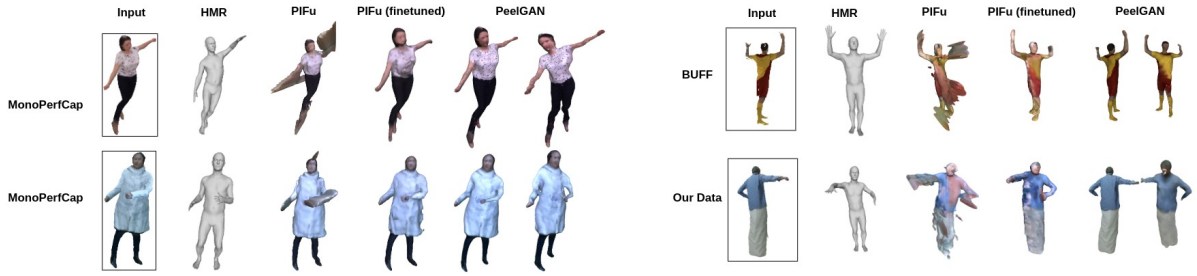

Figure 5.7 Qualitative comparison of HMR and PIFu with PeelGAN for MonoPerfCap, BUFF and Our Dataset. Our method is able to reconstruct plausible shapes efficiently even under severe self-occlusions.

resolution input image so as to have a fair comparison with other methods. We can conclude that our method achieves significantly lower Chamfer distance values as compare to other existing methods.

5.4.4 Discussion

5.4.4.1 Ablation study

We perform a few ablative studies to demonstrate the effect of Chamfer and smoothness losses on the reconstruction quality of our method. Firstly, we train our network without Chamfer loss (Equation 5.7)

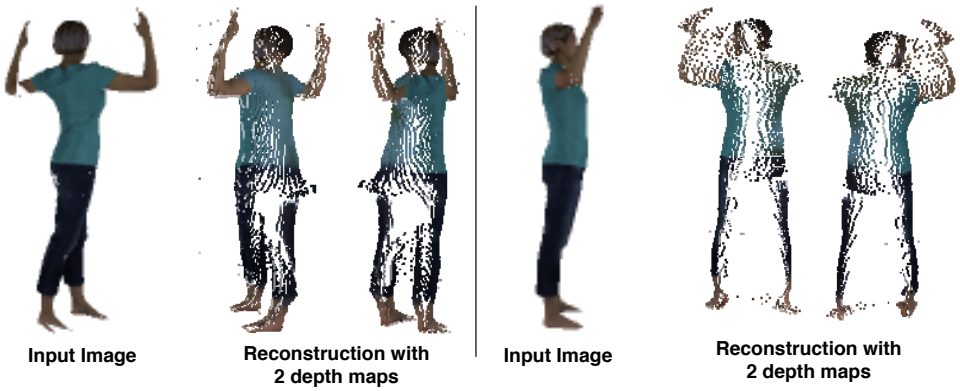

Figure 5.8 Qualitative comparison with (a) Moulding Humans [41] (trained on MonoPerfCap and our dataset)

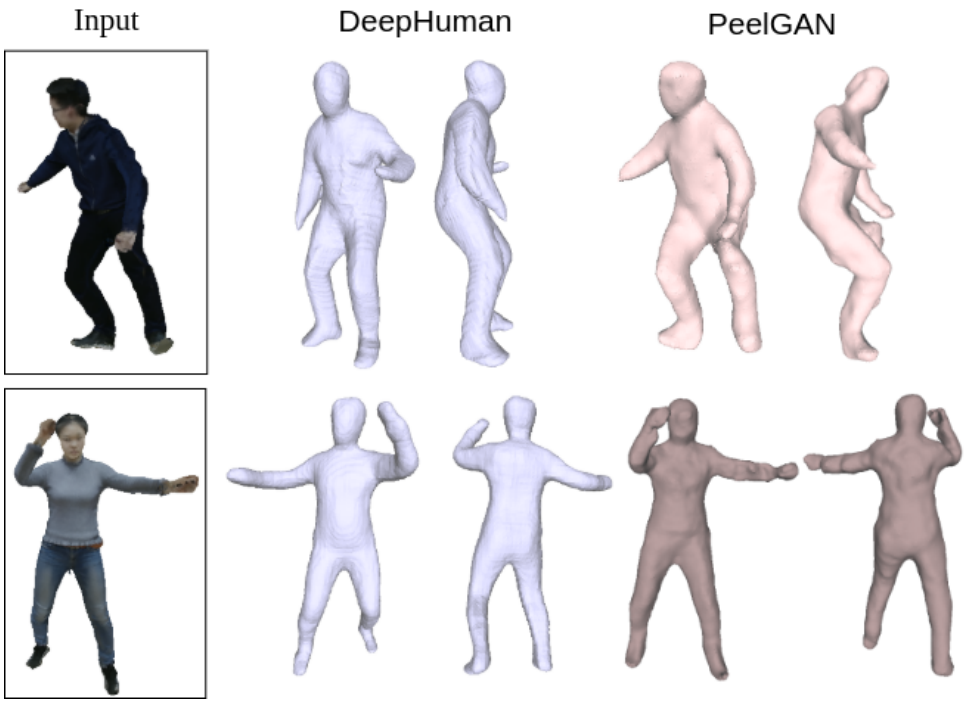

Figure 5.9 DeepHuman [174] (trained on THUman dataset). Both methods fail to recover the shape and surface texture accurately.

as shown in Figure 5.10. The network is not able to hallucinate the presence of occluded parts in the 3^{rd} and 4^{th} depth maps and are hence, missing in Figure 5.10. We can also observe that absence of Chamfer loss produces significant noise in reconstructions (red color points). This can be attributed to independent predictions of individual depth maps using L1 loss.

We also study the effect of smoothness loss (Equation 6.12). This helps the network to produce smoother

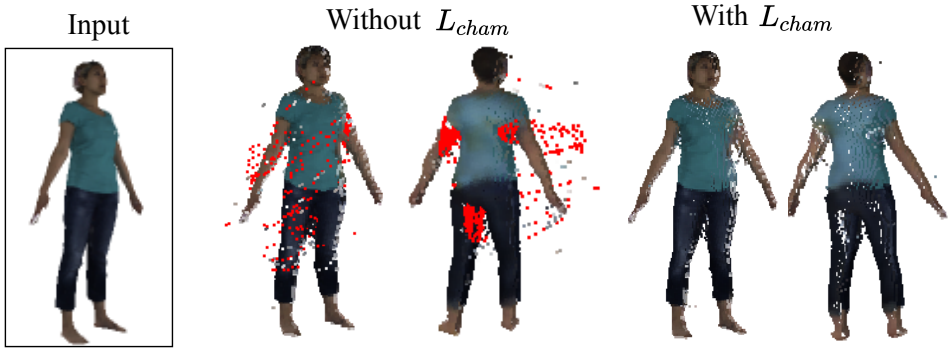

Figure 5.10 Reconstruction without and with Chamfer loss. Red points indicate both noise and occluded regions that were not predicted by the network.

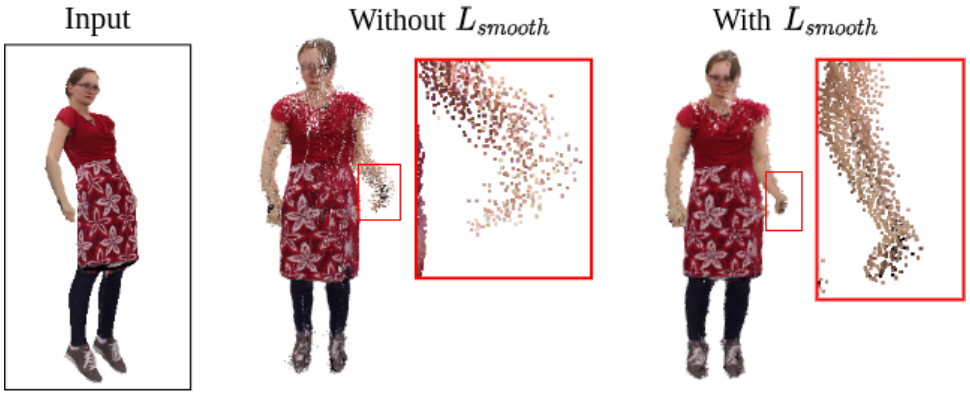

Figure 5.11 Training with smoothness loss improves the quality of Peeled Depth maps.

depth values in layers as shown in Figure 5.11. Thus, Chamfer loss forces the network to predict plausible shapes, that are often noisy, for the occluded parts. Smoothness loss helps the network to smooth out these noisy depth predictions.

5.4.4.2 Effect of Adversarial loss:

In Figure 5.12, we demonstrate the affect of adversarial loss (depth discriminator) on the overall performance of depth peel maps. We train the network with only chamfer, L1 loss where we observe noise in the depth maps (Figure 5.12(b)). For better visualization, we back projected the point cloud. When trained with chamfer, L1 and adversarial loss, we observe the reduction of noise significantly as in Figure 5.12(c). If we train the network without RGB discriminator, we consistently observed that layer 3 and 4 were missing in RGB peel map prediction.

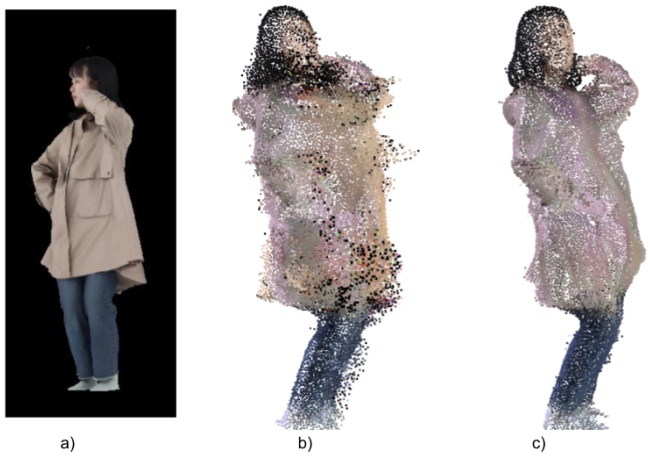

Figure 5.12 Affect of adversarial loss on peel maps: (a) Input image (b) only L1 loss (c) L1 loss and Adversarial loss

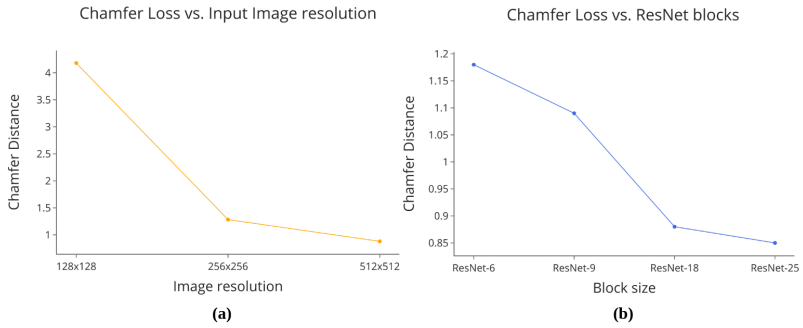

Figure 5.13 (a) Chamfer loss vs. Input image resolution (b) Chamfer loss vs. ResNet blocks

5.4.4.3 In-the-wild images

We also showcase results in Figure 5.14 on an in-the-wild image not present in any dataset. We segment the input image using [44] before feeding it to our model. The predicted Peeled Depth and RGB maps are visualised in (c) and final textured reconstruction in (d). This shows that our method can handle wide varieties in shape, pose and texture.

5.4.4.4 Effect of input Resolution and ResNet blocks

We demonstrate the effect of ResNet blocks and input image resolutions on the performance of PeelGAN in Figure 5.13. As we can observe, Chamfer loss decreases with increase in input image resolution. Similar trend is observed with increasing the number of ResNet blocks. Since the improvement in Chamfer loss from ResNet-18 to ResNet-25 is not significant, we stick to using ResNet-18.

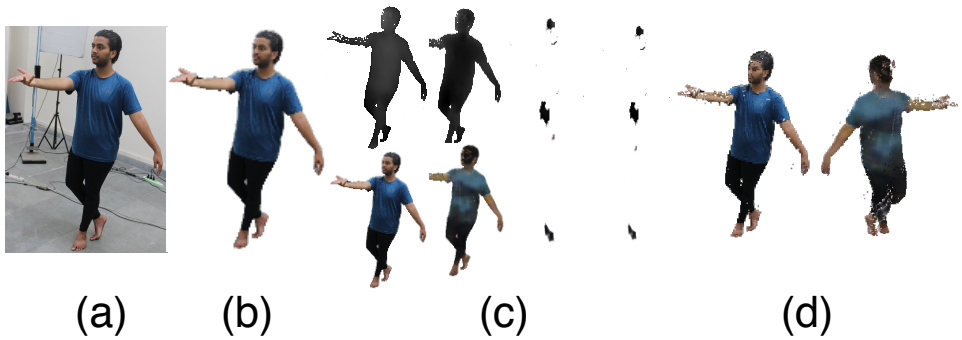

Figure 5.14 Performance of our method on in-the-wild images.

5.5 Summary

We presented a novel representation to reconstruct textured human model from a single RGB image using Peeled Depth and RGB maps. Such an encoding is robust to severe self-occlusions while being accurate and efficient at learning & inference time. Our end-to-end framework i.e. PeelGAN predicts these Peeled maps and has low inference time and recovers accurate 3D human body models from a single view. Our peeled representation suffer from a limitation that it cannot recover surface triangles that are tangential to the viewpoint of the input image. However, this limitation can be addressed with minimal post-processing when recovering corresponding meshes of the reconstructed point clouds. Additionally, PeelGAN might result in distorted body parts as there is no body shape regularization provided to the network.

Chapter 6

SHARP: Shape-Aware Reconstruction of People in Loose Clothing

In this chapter, we propose SHARP (**SH**ape **A**ware **R**econstruction of **P**eople in loose clothing), a novel end-to-end trainable network that accurately recovers the 3D geometry and appearance of humans in loose clothing from a monocular image. SHARP uses a sparse and efficient fusion strategy to combine parametric body prior with a non-parametric PeeledHuman representation introduced in previous chapter. The parametric body prior enforces geometrical consistency on the body shape and pose, while the non-parametric representation models loose clothing and handles self-occlusions as well. We also leverage the sparseness of the non-parametric representation for faster training of our network while using losses on 2D maps. We also introduce 3DHumans dataset, which is a 3D life-like dataset of human body scans with rich geometrical and textural details. We evaluate SHARP on 3DHumans and other publicly available datasets, and show superior qualitative and quantitative performance than existing state-of-the-art methods.

6.1 Introduction

Existing deep learning solutions attempts to fit a parametric body model like SMPL [90] to a monocular input image by learning from image features [65, 5, 104, 86, 75]. However, such parametric SMPL mesh does not capture geometrical details owing to person-specific appearance and clothing. The other class of non-parametric reconstruction techniques pose no such body prior constraints [122, 123, 102, 143, 13, 148] and hence can potentially handle loose clothing scenarios.

In particular, the recent implicit function learning models, like PIFu [122] and PIFuHD [123], estimate voxel occupancy by utilizing pixel-aligned RGB image features computed by projecting 3D points onto the input image. However, the pixel-aligned features suffer from depth ambiguity as multiple 3D points are projected to the same pixel. Another interesting work, Geo-PIFu [51] attempted to refine implicit function estimation by combining volumetric features and pixel-aligned features together to resolve local feature ambiguity. As an alternate representation for 3D objects/scenes, some of the recent works model scenes as multiple (depth) plane images (MPIs) [140] in camera frustum. 3D human body

Figure 6.1 Results of our method on in-the-wild images. Point cloud, uncolored and colored mesh is shown in (a), (b) & (c), respectively.

reconstruction has also been attempted in the same vein by predicting front and back depth maps in [42, 134]. However, the front-back modeling fails to handle self-occlusions caused by body parts.

In the previous chapter, we introduced *PeeledHuman*; a novel non-parametric shape representation of the human body to address the self-occlusion problem. PeeledHuman representation encodes the 3D human body shape as a set of depth and RGB peel maps. Depth (and RGB) peeling is performed by ray-tracing on the 3D body mesh and extending each ray beyond its first intersection to obtain the peel maps. This provides an elegant, sparse 2D encoding of body shape, which inherently addresses the self-occlusion problem. However, the non-parametric approaches do not explicitly seek to impose global body shape consistency and hence, produces implausible body shape and pose.

The aforementioned problems can be addressed by introducing a body shape prior while reconstructing humans in loose clothing. The volume-to-volume translation network proposed in DeepHuman [174] attempts to combine image features with the SMPL prior in a volumetric representation. ARCH [55] proposed to induce a human body prior by sampling points around a template SMPL mesh before evaluating occupancy labels for each point. However, sampling around the canonical SMPL surface is insufficient to reconstruct humans with articulated poses in loose clothing. Similarly, PaMIR [173] proposes to voxelize SMPL body and feed it as an input to the network, which conditions the implicit function around the SMPL feature volume. However, volumetric feature estimation is still computationally expensive and is limited by the resolution. Moreover, in PaMIR, texture and geometry cannot

be inferred in an end-to-end fashion and require two separate networks. Additionally, all these existing SMPL prior-based methods do not effectively exploit the rich surface representation as they either voxelize or sample points around the SMPL surface.

The continuous surface representation provided by SMPL prior is valuable as it models the natural curvature of body parts which cannot be easily recovered with non-parametric methods. Some of the existing methods have been successfully shown to deform SMPL surfaces locally to accommodate relatively tight clothing scenarios [1, 14, 108, 4, 78, 177]. Nevertheless, they fail to handle loose clothing scenarios as the surface of garments can also have complex geometrical structures that are only partially dependent on the underlying body shape and pose, where non-parametric methods have mainly been successful. Interestingly, we can retain the best of these two approaches by deforming SMPL surface locally while reconstructing the remaining surface details (loose clothing) with no body prior constraints. More specifically, one can decouple the reconstruction of 3D clothed body surface into two complementary partial reconstruction tasks: (a) to recover the person-specific body surface details by locally deforming the SMPL prior, (b) to recover the remaining surface geometrical details of the loose clothing that cannot be recovered by just deforming the SMPL prior.

In regard to the representation of the 3D surface, while implementing the above two tasks, PeeledHuman representation seems to be a good choice owing to its sparse encoding of 3D surface into 2D maps. More importantly, such representation also enables seamless fusion of the two partial reconstructions due to the spatially aligned nature of these maps.

Thus, in this chapter we propose SHARP, a novel 3D body reconstruction method that can successfully handle significantly loose clothing, self-occlusions and arbitrary viewpoints. SHARP takes SMPL body encoded in PeeledHuman representation aligned to the input image as a prior to the reconstruction framework. The *SMPL prior peel maps*, along with the monocular RGB image, is fed as an input to our framework, which initially predicts the *residual peel maps* and *auxiliary peel maps*, along with *RGB peel maps*. Here, the residual peel maps represent the pixel-wise depth offsets from SMPL prior peel maps in the view direction. On the other hand, auxiliary peel maps model the complementary geometrical details of the surface, which are not handled by residual peel maps. Subsequently, predicted residual and auxiliary peel maps are fused to obtain *fused peel maps*, representing the geometry of the unified clothed body. The final fused peel maps, along with predicted RGB peel maps are back-projected to obtain the colored point cloud. We finally recover the mesh after minimal post-processing of the corresponding point cloud followed by meshification using Poisson Surface Reconstruction[69]. The fused peel maps can model arbitrarily loose clothing and can handle accessories (e.g., bags) as well, as shown in Figure 6.1. Unlike other existing methods that use adversarial loss and 3D Chamfer loss, the proposed problem formulation enables our network to learn only with L_1 losses on 2D maps, which reduces the training time. Since, the network predicts only 2D maps, the inference time is also significantly reduced.

Additionally, many state-of-the-art methods for reconstructing 3D human bodies [122, 123, 173, 102, 55] train their models on expensive commercial datasets which are not publicly available. These datasets have 3D human body scans which resemble real humans. This data helps the learning-based models to

generalize well on unseen real-world images. Unfortunately, the majority of existing datasets available in the public domain [14, 174, 11, 138] either consist of 3D body models in relatively tighter clothing, lack high-frequency geometrical & texture details, or are synthetic in nature. Recently, THUman2.0 [169] dataset released in the public domain has high-quality 3D body scans captured using a dense DSLR rig. Although, they provide human scans with relatively loose clothing styles, their data lacks significantly loose garment types which occlude the lower body completely, e.g., long-skirt/tunic/saree. Moreover, the dataset is reconstructed with the multi-camera setup which has its known limitations. To bridge these gaps, we collected **3DHumans**, a dataset of 3D human body scans with a wide variety of clothing styles and varied poses using a commercial structured-light sensor (accurate up to 0.5mm). We are able to retain high-frequency geometrical and textural details, as shown in Figure 6.5. We also benchmark some of the State-of-the-Art (SOTA) methods on this dataset and report superior results of our method. To summarize, our contributions are:

1. We propose SHARP, a novel approach to fuse parametric and non-parametric shape representation using losses on 2D maps for reconstructing 3D body model from an input monocular (RGB) image.

2. Our proposed end-to-end learnable encoder-decoder framework infers color and geometrical details of body shape in a single forward pass at lower inference time as compare to SOTA methods.

3. We collected **3DHumans**, a dataset that has a wide variety of clothing and body poses with high-frequency details of texture and geometry. The dataset will be released in the public domain to further accelerate the research.

6.2 Related work

Parametric body fitting. Estimating the 3D parametric human body models, like SMPL [90], SMPL-X [110], SCAPE [6] etc., using deep learning methods [18, 65] has achieved a great success with robust performance. These methods estimate SMPL parameters from a single image. In particular, HMR [65] proposes to regress SMPL parameters while minimizing re-projection loss with the known 2D joints. Different priors have been used to refine the parametric estimates as in [146, 104, 73, 67, 75, 86]. Despite these approaches being computationally efficient, they lack realistic human appearance and clothing details. Methods for modelling details like hair/cloth/skin by estimating offsets from SMPL vertex have been proposed, but they work on very tight clothing and can not model the loose clothing deformation arising from pose. [14, 151, 74].

Non-parametric body reconstruction: Recovering 3D human body from multi-camera setup requires traditional techniques like voxel carving, triangulation, multi-view stereo, shape-from-X [10, 37, 22, 101]. Stereo cameras and consumer RGBD sensors are highly susceptible to noise. In the domain of deep learning, initially, voxel methods gained popularity as 3D voxels are a natural extension to 2D

pixels [148, 143, 174]. SiCloPe [102] estimates human body silhouettes in novel views to recover underlying 3D shape from 2D contours. Recently, implicit function learning methods for human body reconstruction became popular, which use pixel-aligned features to learn neural implicit function over a discrete occupancy grid. [122, 123]. However, these methods suffer from sampling redundancy as they have to sample points in a grid to infer the surface, majority of which do not lie on the actual surface. They also suffer from depth ambiguity as multiple 3D points map to the same pixel-aligned feature. In our recent work[59], we proposed a sparse 2D representation of 3D surface by estimating and storing the intersection of the surface with ray.[99] where it samples points along the camera ray to evaluate RGBσ on these samples.

Prior-based non-parametric body reconstruction: ARCH [55] learns a deep implicit function by sampling points around the 3D clothed body in the canonical space. But, the transformation of the clothed mesh from canonical space to arbitrary space is done by learning SMPL-based skinning weights which can not handle the deformation of the loose clothing. Geo-PIFu [51] utilizes structure-aware latent voxel features, along with pixel-aligned features to learn a neural implicit function. PaMIR [173] learns a deep implicit function conditioned on the features which are a combination of 2D features obtained from image and 3D features obtained from the SMPL body volume. However, voxel features are computationally expensive and of low resolution. DeepHuman [174] leverages dense semantic representations from SMPL as an additional input. Nevertheless, similar to Geo-PIFu, DeepHuman is also a volumetric-regression based approach and hence, incurs a high computational cost. Moreover, similar to PIFu, these deep implicit methods require separate networks for learning geometry and texture.

3D Human body datasets: Deep-Learning based 3D human body reconstruction solutions rely on the data available at hand. Not only the shear amount of samples, but the quality of geometry and texture is also important in order to drive the learning. Many 3D human body datasets have been proposed, some of which only contain body-shape information, while some also include clothing details on top of it. TOSCA [25] dataset contains synthetic meshes of fixed topology with artist-defined deformations. SHREC [81] and FAUST [20] provide meshes and deformation models created by an artist that cannot reproduce what we find in the real world. BUFF [170] contains 3D scans with relatively richer geometry details, but the number of subjects, poses and clothing style is very limited and not sufficient to generalize deep learning models. Another synthetic dataset CLOTH3D [11] incorporates loose clothing by draping 3D modeled garments on SMPL in Blender. It has a wide variety of clothing styles, but due to the nature of SMPL body model, details like hair and skin are absent. THUman1.0 [174] dataset provides a large number of human meshes with varied poses and subjects. However, the texture quality is low and cannot mimic real-world subjects. SIZER [138] dataset provides real scans of 100 subjects, wearing garments in 4 different sizes of 10 fixed garments classes. But all the scans are in A-pose which is insufficient for a deep learning model to generalize to different poses. THUman2.0 [169] dataset provides a large number of high-quality textured meshes of different subjects in various poses. It also

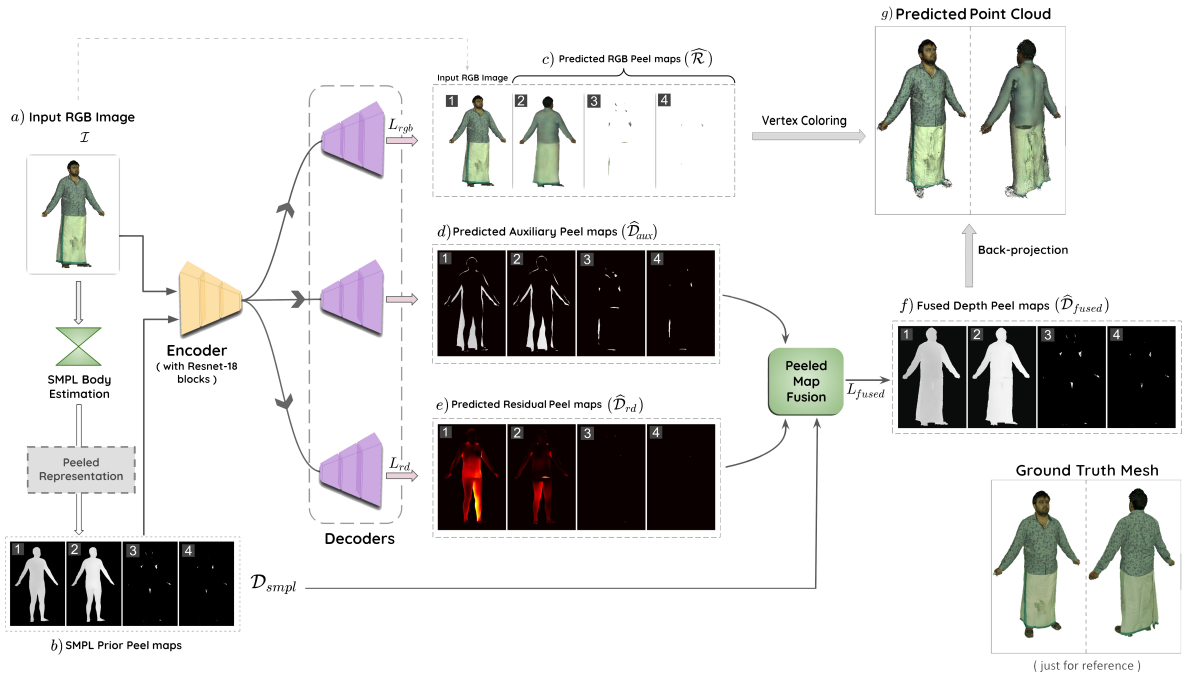

Figure 6.2 Pipeline: We use an off-the-shelf method to estimate SMPL prior from the input image $\mathcal{I}$, and encode it into peeled representation ($\mathcal{D}_{smpl}$). This, along with image $\mathcal{I}$, is fed to an encoder. Subsequently, three separate decoders branches predict RGB peel maps ($\widehat{\mathcal{R}}$), auxiliary peel maps ($\widehat{\mathcal{D}}_{aux}$) and residual peel maps ($\widehat{\mathcal{D}}_{RD}$), respectively. Finally, a layer-wise fusion of $\widehat{\mathcal{D}}_{aux}$, $\widehat{\mathcal{D}}_{rd}$ and $\mathcal{D}_{smpl}$ is performed to obtain fused peel maps $\widehat{\mathcal{D}}_{fused}$, which is then back-projected along with $\widehat{\mathcal{R}}$ to obtain a vertex colored point-cloud. (The ground truth mesh is shown for comparison only.)

incorporates varied clothing styles and high-frequency geometrical details like hair and wrinkles etc. However, loose wrapped clothing styles, which completely occlude the full body, are still absent.

6.3 Method

In this section, we outline the details of SHARP. We aim to reconstruct a 3D textured human body model of a person in arbitrary pose and clothing from a given monocular input image $\mathcal{I}$, as shown in Figure 6.2.

Here, we discuss the steps involved in our proposed method.

1. SMPL shape and pose parameters (i.e., $\beta \in \mathbb{R}^{10}$, $\theta \in \mathbb{R}^{72}$) along with parameters of weak perspective camera (s, t_x, t_y) are estimated from ProHMR [75]. We convert the estimated SMPL to depth peel maps which acts as a shape prior $\mathcal{D}_{smpl}$ (Figure 6.2) as outlined in subsubsection 6.3.1.1.

2. Later, input image $\mathcal{I}$ (with background removed) is concatenated with $\mathcal{D}_{smpl}$ and is fed as an input to the shared encoder in our network.

3. Subsequently, three decoders predict different outputs through separate branches, namely, RGB peel maps $\widehat{\mathcal{R}}$, auxiliary peel maps $\widehat{\mathcal{D}}_{aux}$ and residual peel maps $\widehat{\mathcal{D}}_{rd}$, as shown in Figure 6.2 (c)-(e). The topmost decoder branch predicts only three RGB peel maps as the input $\mathcal{I}$ naturally acts as the first RGB peel map.

4. The predicted auxiliary peel maps $\widehat{\mathcal{D}}_{aux}$ and residual peel maps $\widehat{\mathcal{D}}_{rd}$ are further combined using SMPL mask Γ_i (estimated using Equation 6.2) to obtain the final fused peel maps $\widehat{\mathcal{D}}_{fuse}$.

5. Finally, a colored point-cloud is obtained by back-projecting $\widehat{\mathcal{D}}_{fuse}$ and $\widehat{\mathcal{R}}$ to camera coordinate system, as shown in Figure 6.2 (g). This point-cloud is further post-processed, and then meshified using Poisson Surface Reconstruction (PSR) [69].

To illustrate the importance of a shape prior in the prediction of peel maps, we compare SHARP with PeelGAN [59]. The PeelGAN network predicts inconsistent body parts as shown in Figure 6.4 (a). This is because of the fact that there are no geometrical constraints imposed on the structure of predicted body parts. The introduction of prior enables SHARP to reconstruct the human body with plausible body parts and accurate pose as shown in Figure 6.4 (b).

6.3.1 Pipeline details

Here, we discuss in detail about our pipeline, which involves peeled shape prior, residual & auxiliary peel maps and finally, peel map fusion. We also explain in detail the loss functions used to train SHARP.

6.3.1.1 Peeled shape (SMPL) Prior

We initially use [75] to estimate the SMPL pose and shape parameters (β, θ), along with weak-perspective camera parameters (s, t_x, t_y). The SMPL mesh is brought into the camera coordinate system using (s, t_x, t_y), and then encoded into depth peel maps by passing camera rays through each pixel, as explained in **??**, i.e., for every pixel p in layer i, depth value of the point intersected by the camera ray is stored:

$$\mathcal{D}_{smpl} = \{(d_p^i) : \forall p \in \mathcal{I}, i \in \{1, 2, 3, 4\}, d \in \mathbb{R}\} \tag{6.1}$$

We initialize a layer-wise SMPL mask Γ_i by applying binary thresholding on SMPL prior peel maps. Additionally, we condition this mask on a pre-estimated binary foreground mask $\mathcal{F}$. The foreground mask $\mathcal{F}$ covers only the clothed human in the input image and can be obtained using off-the-shelf background segementation methods e.g.PGN [45] . We use $\mathcal{D}_{smpl}^i$ and $\mathcal{F}$ to estimate the SMPL maks

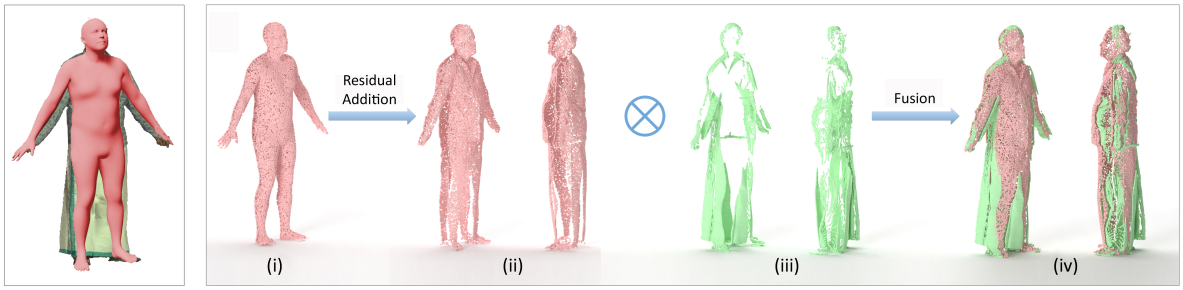

Figure 6.3 (a) SMPL prior overlayed on the input image: The residual peel maps recover depth along the pixels over which SMPL prior is present across all the layers. For the remaining pixels, auxiliary peel maps are used to recover depth. **(b) 3D representation of fusion:** (i) Point cloud obtained from $\mathcal{D}_{smpl}$ shown in red. (ii) Point cloud obtained from $\widehat{\mathcal{D}}_{rd}$ shown from two views in red. (iii) Point cloud obtained from $\widehat{\mathcal{D}}_{aux}$ shown from two views in green. (iv) Final point cloud from fused depth peel maps $\widehat{\mathcal{D}}_{fuse}$.

Γ_i as :

$$\Gamma_i = \begin{cases} 1, & \text{if } \mathcal{D}^i_{smpl} \odot \mathcal{F} > 0 \text{ and} \\ 0, & \text{otherwise.} \end{cases} \tag{6.2}$$

In essence, Γ_i for each layer is estimated by retaining only the overlapping regions in corresponding SMPL prior peel map and the foreground mask. This helps refine the SMPL mask Γ_i by eliminating parts of the SMPL prior peel maps that goes outside the human body & clothing silhouette, thereby enabling our method to partially overcome the misalignment of SMPL prior with the input image. Γ_i mask is subsequently used for peel map fusion in Equation 6.3.1.3.

6.3.1.2 Residual and Auxiliary peel maps

To estimate view specific deformations from the SMPL prior input, we propose to predict residual peel maps $\widehat{\mathcal{D}}_{rd}$ by computing additive pixel-wise offsets from the input SMPL depth peel maps $\mathcal{D}_{smpl}$. For every pixel p in layer i of peeled SMPL prior, we predict offset along z-axis [1]:

$$\widehat{\mathcal{D}}_{rd} = \{(\widehat{\delta d}^{\,i}_p) : \forall p \in \mathcal{I}, i \in \{1,2,3,4\}, \widehat{\delta d} \in \mathbb{R}\} \tag{6.3}$$

On pixels depicting the projection of bare body parts, network predicts minimal offsets ($\widehat{\mathcal{D}}_{rd}$), thereby capturing the person-specific appearance features like hairline and facial feature while preserving overall structure of the body parts.

[1]The camera is placed at (0,0,10), Y-axis is up and -Z axis is forward, while meshes are placed at origin

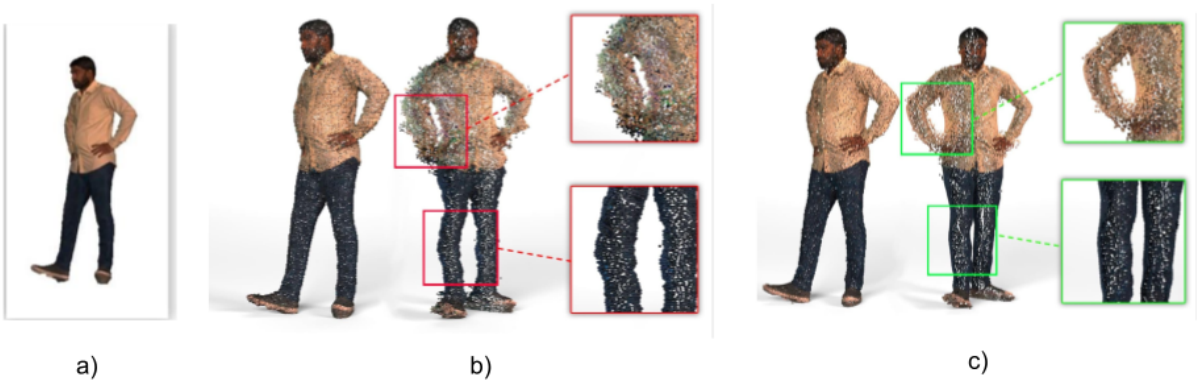

Figure 6.4 (a) Input image, (b) Distorted body parts in the prediction from PeelGAN [59]. (c) Reconstruction obtained from SHARP.

Thus, each layer of the residual peel maps provides pixel-wise displacements of the corresponding layer of peeled SMPL prior maps along the view-direction (z-axis). These residual deformations only cover the pixels in the input image for which SMPL prior is present. For the remaining pixels of clothed body, we propose to learn their depth values using a separate branch in the form of auxiliary peel maps.

$$\widehat{\mathcal{D}}_{aux} = \{(\widehat{d}^{i}_{aux}) : \forall p \in \mathcal{I}, i \in \{1,2,3,4\}, \widehat{d} \in \mathbb{R}\} \tag{6.4}$$

Figure 6.3 provide 3D visualization of $\widehat{\mathcal{D}}_{rd}$ and $\widehat{\mathcal{D}}_{aux}$ by back-projecting respective partial depth peel maps. We can observe that these two maps capture the complementary geometrical details of 3D body and clothing.

6.3.1.3 Peel map fusion

The predicted residual and auxiliary peel maps independently capture complimentary surface details and are subsequently fused to obtain the geometry of unified clothed body. We propose to obtain final fused peel depth maps by layer-wise fusion of $\mathcal{D}_{smpl}$, $\widehat{\mathcal{D}}_{rd}$ and $\widehat{\mathcal{D}}_{aux}$ expressed as:

$$\widehat{\mathcal{D}}_{fuse} = (\mathcal{D}_{smpl} + \widehat{\mathcal{D}}_{rd}) \otimes \widehat{\mathcal{D}}_{aux} \tag{6.5}$$

where $\otimes$ is the proposed layer-wise fusion operator as explained below. Here,

$$\widehat{\mathcal{D}}^{i}_{fuse} = \Gamma_i \odot (\widehat{\mathcal{D}}^{i}_{rd} + \mathcal{D}^{i}_{smpl}) + (1 - \Gamma_i) \odot \widehat{\mathcal{D}}^{i}_{aux} \tag{6.6}$$

where $\odot$ is element-wise multiplication and for each i^{th} layer $\widehat{\mathcal{D}}^{i}_{aux} \in \widehat{\mathcal{D}}_{aux}$, $\widehat{\mathcal{D}}^{i}_{rd} \in \widehat{\mathcal{D}}_{rd}$ and $\mathcal{D}^{i}_{smpl} \in \mathcal{D}_{smpl}$.

In summary, we have decoupled the task of recovering the clothed 3D human body surface into predicting residual and auxiliary peel maps. We later fused these partial reconstructions into a single unified 3D surface. Our approach ensures geometrically consistent body parts as the residual peel maps

predict minimal offsets on the pixels belonging to the bare body where there is no clothing, thereby retaining body-specific geometry.

6.3.2 Loss functions

We use encoder-decoder architecture for our predictions in SHARP. We train our network with losses on 2D map predictions only. Our final learning objective is defined as:

$$L = L_{fuse} + \lambda_{rd} L_{rd} + \lambda_{rgb} L_{rgb} + \lambda_{sm} L_{sm} \tag{6.7}$$

where λ_{rd}, λ_{rgb} and λ_{sm} are regularization parameters for L_{rd}, L_{rgb}, L_{sm}, respectively. We provide the formulation for the individual loss terms below.

$$L_{fuse} = \sum_{i=1}^{4} \left\| \widehat{\mathcal{D}}^i_{fuse} - \mathcal{D}^i_{fuse} \right\|_1 \tag{6.8}$$

L_{fuse} is the sum of L_1 norm between ground truth depth peel maps $\mathcal{D}_{fuse}$ and predicted fused peel maps $\widehat{\mathcal{D}}_{fuse}$ for each of the i^{th} peeled map layer.

$$L_{rd} = \sum_{i=1}^{4} \left\| \widehat{\mathcal{D}}^i_{rd} - \mathcal{D}^i_{rd} \right\|_1 \tag{6.9}$$

L_{rd} constraints the residual peel map offsets predictions to that of ground truth offsets. Note that we are training auxiliary peel maps branch without any explicit loss on $\widehat{\mathcal{D}}_{aux}$. The gradient to auxiliary peel map branch back-propagates using L_{rd} and L_{fuse}.

We also enforce per layer first order gradient smoothness of the predicted $(\widehat{\mathcal{D}}^i_{rd} + \mathcal{D}^i_{smpl})$ and ground truth $(\mathcal{D}^i_{rd} + \mathcal{D}^i_{smpl})$ as well as between ground truth and predicted $\widehat{\mathcal{D}}_{fuse}$ maps. L^{fuse}_{sm} ensures smoothness between the two surfaces we predict.

$$L_{sm} = L^{rd}_{sm} + L^{fuse}_{sm} \tag{6.10}$$

where,

$$\begin{aligned} L^{rd}_{sm} &= \sum_{i=1}^{4} \left\| \bigtriangledown(\mathcal{D}^i_{rd} + \mathcal{D}^i_{smpl}) - \bigtriangledown(\widehat{\mathcal{D}}^i_{rd} + \mathcal{D}^i_{smpl}) \right\|_1 \\ L^{fuse}_{sm} &= \sum_{i=1}^{4} \left\| \bigtriangledown\mathcal{D}^i_{fuse} - \bigtriangledown\widehat{\mathcal{D}}^i_{fuse} \right\|_1 \end{aligned} \tag{6.11}$$

Additionally, We also train our network with L_1 loss between predicted and ground truth RGB peel maps (L_{rgb}).

6.4 Experiments & results

In this section, we present the experimental details, datasets and training protocol for SHARP. We also show qualitative and quantitative comparisons with current state-of-the-art methods.

6.4.1 Implementation details

We employ a multi-branch encoder-decoder network for SHARP, which is in an end-to-end fashion. The network takes input image concatenated with SMPL peel maps in 512×512 resolution. The shared encoder consists of a convolutional layer and 2 downsampling layers which have $64, 128, 256$ kernels of size 7×7, 3×3 and 3×3, respectively. This is followed by ResNet blocks which take downsampled feature maps of size 128×128×256. The decoders for predicting $\widehat{\mathcal{D}}_{fuse}$ and $\widehat{\mathcal{D}}_{rd}$ and $\widehat{\mathcal{R}}$ consist of two upsampling layers followed by a convolutional layer which have similar kernel sizes as in encoders. Sigmoid activation is used in $\widehat{\mathcal{D}}_{fuse}$ and $\widehat{\mathcal{D}}_{rd}$ decoder branches, while a tanh activation is used for the $\widehat{\mathcal{R}}$ decoder branch. The $\widehat{\mathcal{D}}_{rd}$ output values are scaled to a $[-1, 0.5]$ range which is found empirically.

We use the Adam optimizer with an exponentially decaying learning rate starting from 5×10^{-4}. Our network takes around 18 hrs to train for 20 epochs on 4 Nvidia GTX 1080Ti GPUs with a batch size of 8 and λ_{rd}, λ_{fuse}, λ_{rgb} and λ_{sm} are set to $1, 1, 0.1$ and 0.001, respectively, found empirically. We use trimesh [35] library for rendering the peel maps.

6.5 Datasets

6.5.1 3DHumans

We present 3DHumans, a dataset of around 250 scans containing people in diverse body shapes in various garments styles and sizes with high fidelity geometry and textural details Figure 6.5. We cover a wide variety of clothing styles ranging from loose robed clothing like saree to relatively tight-fitting shirt and trousers, as shown in Figure 7.3. The dataset consists of around 150 male and 50 unique female subjects. Total male scans are about 180 and female scans are around 70. We will release this data in the public domain for academic use.[2]. For additional details regarding distribution of clothes, poses and capture information of the dataset refer section 7.2.

6.5.2 Other datasets

In addition to our 3dHumans dataset (section 7.2), we perform both qualitative and quantitative evaluations on the following publicly available datasets. **CLOTH3D** [11] is a collection of 6500 synthetic sequences of SMPL meshes with garments draped onto them, simulated with MoCap data. Each frame of a sequence contains garment and corresponding SMPL body. The garment styles range from skirts

[2]http://cvit.iiit.ac.in/research/projects/cvit-projects/sharp-3dhumans-a-rich-3d-dataset-of-scanned-humans

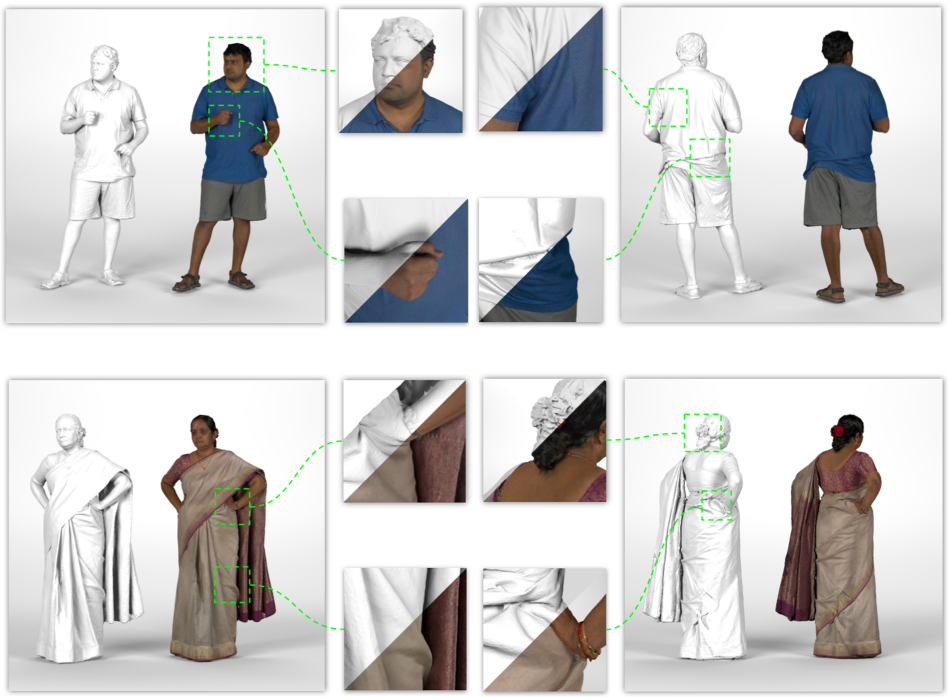

Figure 6.5 High-frequency geometrical and textural details present in our 3DHumans dataset.

to very loose robes. We augment this data by capturing SMPL texture maps with minimal clothing to simulate realistic body textures using [1]. For each sequence, five frames are randomly sampled. Please refer supplementary for the pre-processing of these meshes.

THUman1.0 [174] consists of 6800 human meshes registered with SMPL body in varying poses and garments. The dataset was obtained using consumer RGBD sensors. Although the dataset has diverse poses and shapes, it has relatively tight clothing examples with low-quality textures. Please refer supplementary for results on this dataset. Note that the dataset is originally called the **THUman** dataset, we refer it to as **THUman1.0** to avoid the confusion.

THUman2.0 [169] is a collection of 500 high quality 3D scans captured using dense DSLR rig. The dataset offers wide variety of poses. However, very loose clothing styles like robed skirts are still lacking. Each mesh in the provided dataset is in different scale. We have brought all the meshes in the same scale by registering SMPL to the scans and performed our experiments.

6.5.3 Evaluation metrics

To quantitatively evaluate performance of SHARP, we use the following evaluation metrics:

	Our Dataset			THUman2.0 Dataset		
Method	CD ($\times 10^{-5}$) $\downarrow$	P2S $\downarrow$	Normal $\downarrow$	CD ($\times 10^{-5}$) $\downarrow$	P2S $\downarrow$	Normal $\downarrow$
PIFu	20.79	0.00826	0.054	23.72	0.0091	0.036
Geo-PIFu	15.73	0.0092	0.058	17.01	0.0092	0.041
PaMIR	12.54	0.00714	0.054	6.05	**0.0049**	0.038
PeeledHuman	20.88	0.0094	0.061	23.34	0.0094	0.054
Ours	**7.718**	**0.0051**	**0.045**	**6.044**	0.00529	**0.034**

Table 6.1 Quantitative comparison on 3DHumans and THUman2.0 datasets.

Method	CD $\downarrow$	P2S $\downarrow$
JumpSuit	0.00031	0.00872
Dress	0.0012	0.021
Top+Trousers	0.00057	0.0118

Table 6.2 Performance of our method on clothing styles of CLOTH3D dataset.

Point-to-Surface (P2S) Distance: Given a set of points and a surface, P2S measures the average L2 distance between each point and the nearest point to it on the given surface. We use P2S to measure the deviation of the point cloud (back-projected from predicted fused peel maps) from the ground truth mesh.

Chamfer Distance (CD): Given two sets of points S_1 and S_2, Chamfer distance measures the discrepancy between them as follows:

$$d_{CD}(S_1, S_2) = \sum_{x \epsilon S_1} min_{y \epsilon S_2} \|x - y\|_2^2 + \sum_{y \epsilon S_2} min_{x \epsilon S_1} \|x - y\|_2^2 \tag{6.12}$$

Normal Re-projection Loss: To evaluate the fineness of reconstructed quality, we compute normal reprojection loss introduced in saito2019pifu. We render the predicted and ground truth normal maps in the image space from the input viewpoint. We then calculate the L2 error between these two normal maps.

6.5.4 Quantitative evaluation

We evaluate the aforementioned metrics on 3DHumans, THUman2.0 datasets on PIFu [122], PaMIR [173], Geo-PIFu [51] and PeeledHuman [59]. We retrained the respective models on these datasets under the same train/test split. We transform all the predicted models from different methods to the

Figure 6.6 Results on 3DHumans and THUman2.0 datasets.

canonical coordinates of the ground truth mesh and report metrices in Table 6.1. The quantitative comparison concludes that our method outperforms the SOTA methods. Unlike PeelGAN [59] that uses a generative network, we use a simple encoder-decoder architecture. We trained PaMIR with approximately thrice the amount of data (SHARP is trained on 70 views per mesh, while for PaMIR, 180 views per mesh are used). Geo-PIFu needs to be trained for coarse and query networks separately and complete training takes three days to train on 3DHumans in our setup.

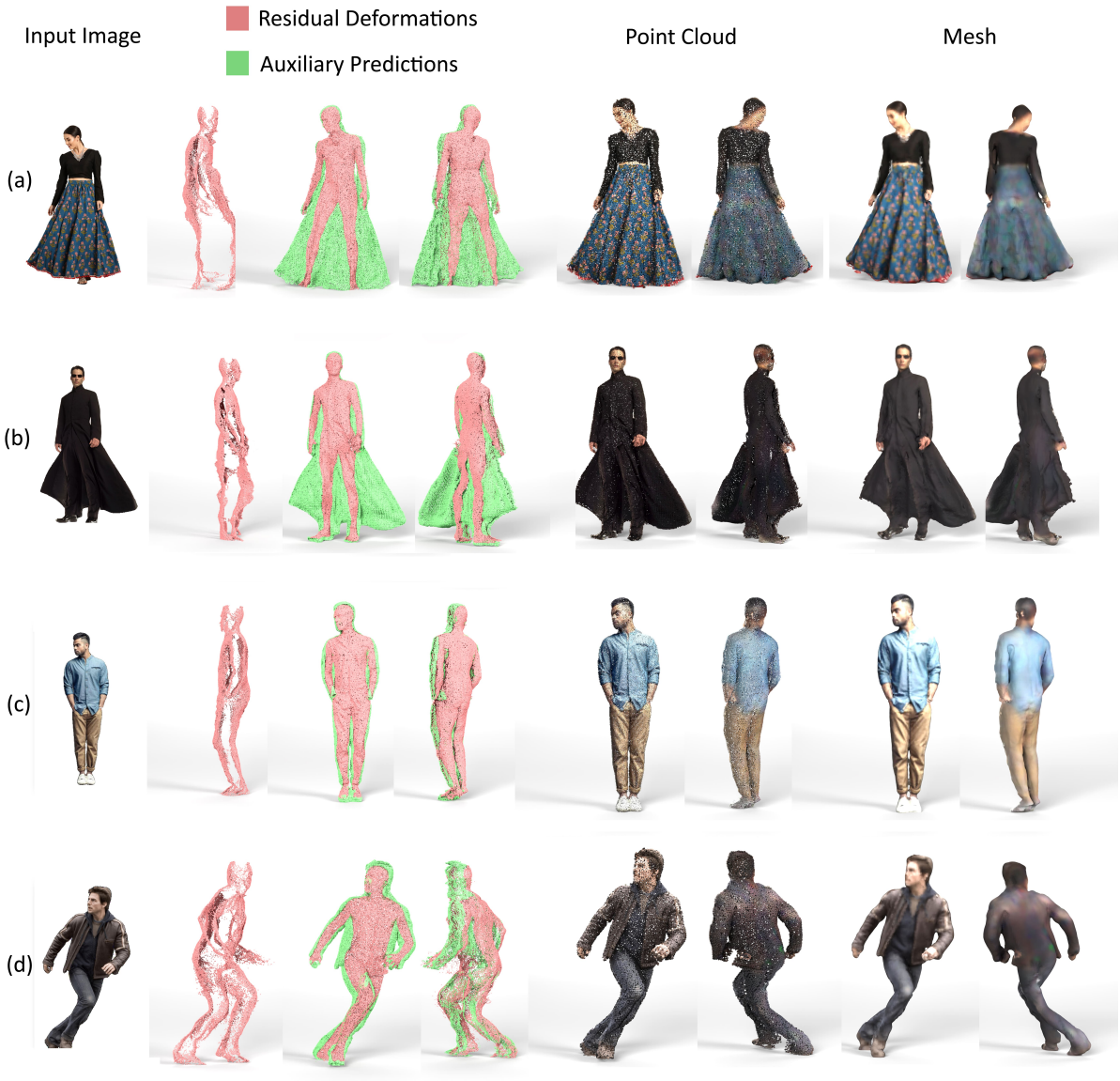

Figure 6.7 Results on in-the-wild images.

Additionally, Table 6.2 summarizes quantitative analysis on the CLOTH3D dataset where we evaluate CD and P2S metrics on different styles of clothing to indicate the generalization of our method across various clothing styles. We also provide comparisons with THUman1.0 and CLOTH3D datasets in the supplementary.

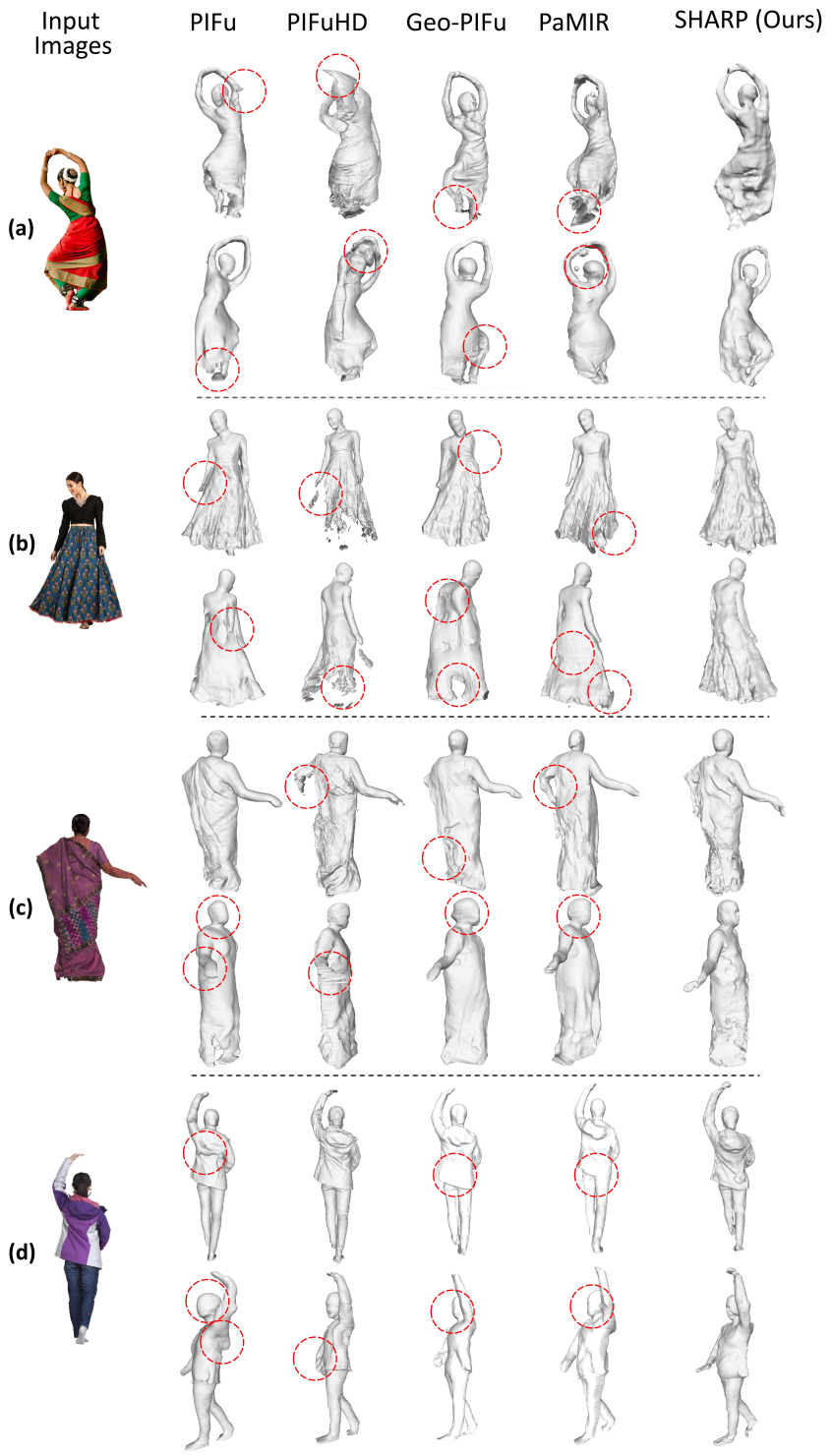

Figure 6.8 Qualitative comparison of SOTA methods. (a) and (b) are from random internet images, (c) and (d) are from 3DHumans and THUman2.0 datasets respectively, shown in two views each.

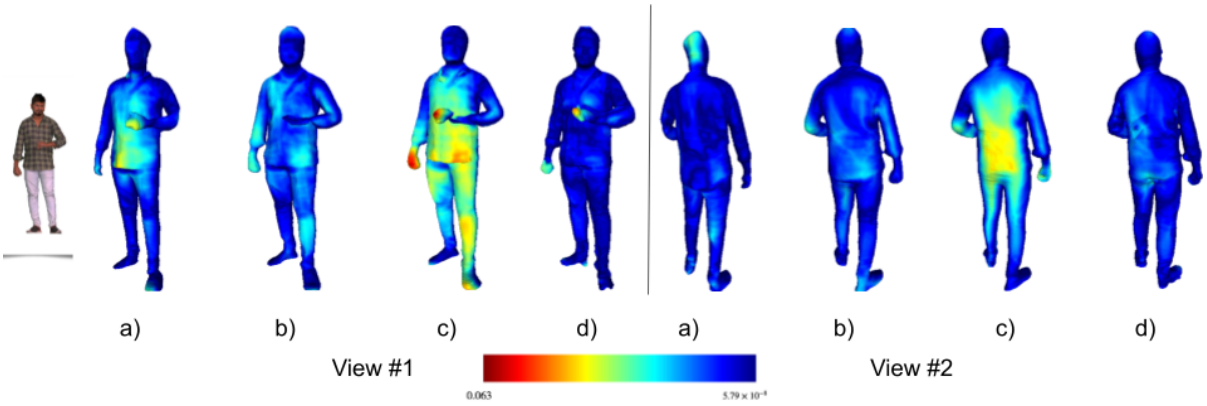

Figure 6.9 P2S Plot: Point-to-surface plots on the reconstructed outputs from (a) PaMIR, (b) Geo-PIFu, (c) PIFu and (d) Ours.

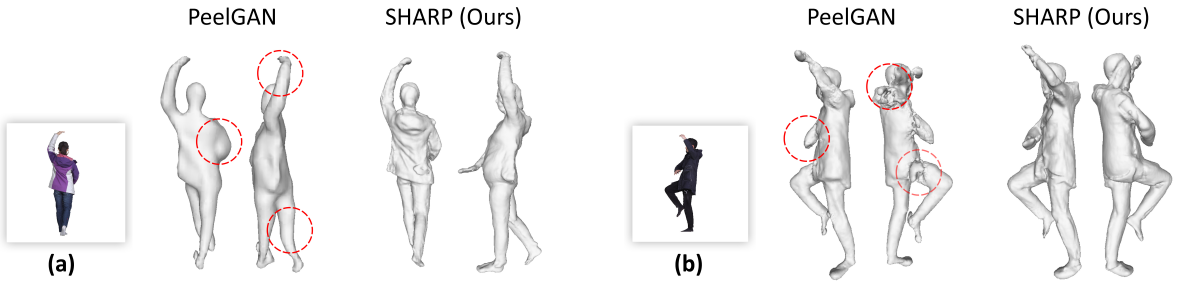

Figure 6.10 Qualitative comparison of PeelGAN and SHARP.

6.5.5 Qualitative evaluation

We show the reconstructions obtained by our method using THUman2.0 and 3DHumans datasets (Figure 6.6), where we also show point clouds obtained by back-projecting residual and auxiliary peel maps. Figure 6.6 (a) and (b) are samples from our dataset, (c) and (d) are from THUman2.0 dataset. Please refer supplementary for additional results on CLOTH3D dataset. One can observe that our model can handle various styles of clothing (including *tunic*) covering the lower body parts and with a wide variety of poses. Residual and auxiliary peel maps captured the complimentary surface details as visualized in red and green in Figure 6.6. In order to test the generalizability of our method on unseen in-the-wild images, we show results on random internet images using our method in Figure 6.7. Similar to PIFu [122], we use an off-the-shelf method to remove the background from these images before passing them to our network. It can be noted that our method is able to reconstruct the human body with self-occlusions and tackle a wide variety of clothing styles, ranging from tight to loose clothing with diverse poses. Notably, our method is able to generalize well on unseen, very loose clothing styles present in Figure 6.7 (a) & (b).

In Figure 6.8, we show qualitative comparison with SOTA methods. PIFu and PIFuHD do not use body prior, which leads to missing and distorted body parts. Geo-PIFu predicts a volumetric prior before performing implicit reconstruction. On the other hand, PaMIR uses SMPL prior as input. Hence, both methods tends to produce smoother geometry as they use voxelized representation, which is known to smooth out the geometrical details. It can be noted that our method retains high-frequency surface details as shown in Figure 6.8. Additionally, we also show comparison with our previous work PeelGAN [59] in Figure 6.10. We observed that our formulation yielded superior results over PeelGAN which also uses the PeeledHuman representation sans SMPL prior.

All the aforementioned methods have been trained on our 3DHumans dataset except for PIFuHD. Since the training code for PIFuHD is not yet available, we use the model provided by the authors. In order to fairly compare with other methods, we selected a body with tight clothing and generated plots of P2S error of all methods trained on our dataset as visualized in Figure 6.9. One can infer from these plots that our approach yields superior performance in terms of distribution of P2S error over the reconstructed surface.

6.5.6 Network complexity

Method	No. of parameters	Execution Time
PaMIR(Geo+Tex)	40M(27M+13M)	4.03s(3.9s+0.13s)
Geo-PIFu(coarse+fine)	30.6M (14.9M+15.7M)	16.32s(0.32s+16s)
Ours	**22M**	**0.09s**

Table 6.3 Comparison of complexity analysis.

We report a detailed analysis of the execution time of SOTA methods in Table 6.3. All the numbers are computed on a single NVIDIA GTX 1080Ti GPU with a single input image. PaMIR needs feed-forward of two networks to infer shape and geometry. On the other hand, Geo-PIFu needs to infer coarse volumetric shape followed by fine shape. We calculate the feed-forward execution time for the complete forward pass of Geo-PIFu and PaMIR as these methods need multiple forward passes while inferring. Note that ours is an end-to-end inference model which predicts both shape and color in a single forward pass efficiently with 0.09 seconds, which is significantly faster when compared to the aforementioned methods. Additionally, our network is lightweight, consisting of 22 million parameters, while PaMIR and Geo-PIFu has total 40 and 30.6 million parameters, respectively.

6.5.7 Ablation Study: Architectural Choices

Impact of loss functions: In Table 6.4, we demonstrate the impact of various loss functions on the output point cloud. First, we evaluate SHARP without smoothness loss (L_{sm}) and observe that it leads

Method	CD ↓	P2S ↓
Ours w.o. L_{sm}	8.3652	0.0053
Ours w.o. fusion	9.98	0.0054
Ours	**7.718**	**0.0051**

Table 6.4 Ablation Study: Effect of loss functions.

to an increase in Chamfer distance and P2S error, which is caused by the noise in prediction of fused peel maps.
Secondly, to evaluate the importance of the peel map fusion, we train our network without fusion. In this setting, we used only two decoder branches, one for predicting RGB peel maps and the other for predicting depth peel maps. This lead to smooth, predictions, which misses out body-specific geometrical details, further increasing CD and P2S values.

Impact of different components: In Table 6.7, we demonstrate the impact of various components in our method. We train the network without any depth prior ($\mathcal{D}_{smpl}$) and predict depth peel maps without fusion and we observe the performance of the network is worser than PeeledHuman which has adversarial setup. However, if depth prior is provided as additional input to this network, the performance is significantly improved indicating the impact of SMPL depth prior. If we model the final 3D body as only residual deformation ($(\mathcal{D}_{smpl} + \widehat{\mathcal{D}}_{rd})$), we observe the deterioration in Chamfer score as loose clothing details are completely neglected. Note that point-to-surface distance is not deteriorated as significantly as Chamfer score as it computes the distance from predicted point cloud to ground truth mesh and vice-versa is not computed. Finally, when we fuse ($\mathcal{D}_{fused}$) both residual and auxiliary peel maps the performance is improved further indicating the importance of our proposed fusion.

Impact of various backbone networks: We evaluate the performance of SHARP on various backbone network architectures. In particular, we used U-Net ronneberger2015u and stacked hourglass network newell2016stacked as backbone networks along with residual networks. All the backbone networks are trained with same loss functions as described in **??**. We report the performance of these networks in Table 6.8. Residual network outperforms both Unet and hourglass networks in this multi-branch prediction task. We also observed that hourglass network was not able to predict four layer RGB peel maps.

Impact of number of ResNet blocks: We also evaluate the performance of SHARP by varying the number of ResNet blocks as shown in Table 6.10. We train our network on 3DHumans with 6, 9 and 18 blocks. Using only 6 ResNet blocks, which is almost one-third of the original network, SHARP is able to achieve similar performance as PIFu (please refer Table 6.1). Using 9 ResNet blocks, we are able to achieve closer numbers to majority of existing SOTA methods. We observed that the further increase in

the number of ResNet blocks did not yield any significant improvement.

Fusion Strategies We analyse the performance of SHARP with various fusion strategies of peel maps. In this experiment, we perform feature level fusion instead of auxiliary and residual depth peel map fusion. We train this fusion network in a coarse-to-fine strategy where initially we replace the auxiliary peel map branch with predicting complete depth peel maps $\widehat{D}_{peel}$. We train this network with losses L_{rd}, L_{sm} and L_1 loss on predicted and ground truth peel maps. We then, take this network as initialization to train fusion module where we take intermediate features of $\widehat{D}_{peel}$ and $\widehat{D}_{rd}$ branches respectively. Refer supplementary (Figure 4) for the architecture diagram. We fuse them using three strategies (a) addition, (b) average and (c) concatenation. These fused features are then passed to upsampling and convolutional layers to predict final fused depth peel maps. Here, we freeze the weights of the network except the layers after the feature fusion. We call it as *Late Fusion* as it requires pre-trained network.

We report the performance in Table 6.9 and learn that late fusion with concatenation results in better performance. However, we note the training for late fusion is not end-to-end as described and we adopt end-to-end trainable network with fusion proposed in Equation 6.5 as our final choice.

6.6 Experiments on additional datasets

We also conduct experiments on two other publicly available datasets, namely CLOTH3D [11] and THUman1.0 [174] (described in **section 5.2** of the main draft). CLOTH3D is a synthetic dataset, which lacks a realistic human appearance (Figure 7.6 **(a)**). Though it contains samples with loose clothing styles, the geometry of the clothes is highly smooth, and the texture is also limited. Due to its synthetic nature, the CLOTH3D dataset can not provide sufficient generalizability to deep neural networks to perform well in in-the-wild settings. On the other hand, THUman1.0 is a real-world dataset that provides meshes of clothed humans obtained using RGBD sensors. However, the samples in THUman1.0 have distorted body structures, and contain artifacts (Figure 7.6 **(b)**). The clothing style is also relatively tighter and high-frequency geometrical and textural details are missing. Thus, we opted to benchmark on the other recent datasets,THUman2.0 [169] and our 3DHumans dataset in the main draft, as samples from these datasets are rich in geometry and texture (Figure 7.6 **(b)** & **(d)**). Nonetheless, we present minimal benchmarking comparison for a subset of SOTA methods on these datasets for completeness of our experimental evaluation. Additionally, we also show the reconstruction results of SHARP on unseen (test set) images from these datasets.

6.6.1 Evaluation on CLOTH3D dataset

Data Preparation

CLOTH3D provides meshes for clothes and human body (SMPL [90]) separately for each frame in a sequence. In order to merge the cloth and body into a single surface, we first cast rays from each face

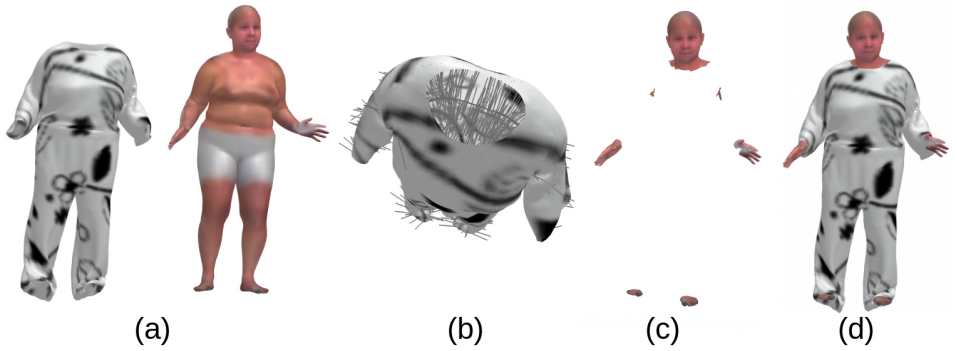

Figure 6.11 Data Preparation for CLOTH3D: (a) Clothed mesh and SMPL body. (b) Casting rays inside clothing volume. (c) Faces of the body which intersect with the rays are removed. (d) Mesh with cloth and body merged.

Method	CD ↓	P2S ↓
Geo-PIFu	0.0016	0.0291
PeelGAN	0.0012	0.0256
Ours	**0.00058**	**0.011**

Table 6.5 Quantitative comparison on CLOTH3D dataset.

of the cloth mesh along the negative direction of face normals. We then compute the intersection of these rays with the underlying SMPL mesh and remove the faces from the SMPL mesh which intersect with these rays. This process removes the faces that are occluded by clothes, retaining the visible body parts which are merged with the cloth mesh (see Figure 6.11). Finally, we apply hole-filling to fill the gaps where boundaries of the cloth meet SMPL body. Since the CLOTH3D data is synthetic in nature, it does not have a realistic human appearance. In order to simulate human realism, we took images of people in minimal clothing from multiple views and used Octopus [1] to generate realistic skin texture maps.

Quantitative and qualitative evaluation

We perform quantitative comparison of SHARP with PeelGAN [59] and Geo-PIFu [51] on CLOTH3D dataset using the CD and P2S metric (introduced in section 5.3 of main draft), as reported in Table 6.5. Our proposed SHARP method outperforms the listed SOTA methods. However, it is important to note that none of these SOTA methods uses explicit SMPL prior along with input monocular image. Figure 6.12 shows the visualization of reconstruction results of our method on CLOTH3D dataset, where

our method is able to recover occluded body parts, along with plausible geometry and textures for unseen loose clothing styles.

6.6.2 Evaluation on THUman1.0 dataset

Quantitative Evaluation:

Table 6.6 shows the quantitative comparison between DeepHuman [174], PIFu [122] and PeelGAN [59] on THUman1.0 dataset. Unlike PIFu and Geo-PIFu, our method and DeepHuman uses explicit input SMPL prior. Our method yield superior performance than PIFu and DeepHuman. Geo-PIFu [51] seems to perform on par with our SHARP. This is probably due to the fact that Geo-PIFu tends to give smooth surface geometry (owing to it's voxel based prior prediction), thereby missing out on high-frequency details. Since THUman1.0 datasets also lack high-frequency details (see Figure 7.6 **(b)**), the average CD and P2S error in prediction from Geo-PIFu is lower.

Qualitative Results: Figure 6.13 shows the reconstruction results of our method on THUman1.0 dataset, where our method is able to recover clothed humans in various poses.

Figure 6.12 Results of SHARP on CLOTH3D dataset.

Method	CD ↓	P2S ↓
DeepHuman	0.00119	0.00112
PIFu	0.00026	0.0004
Geo-PIFu	0.00017	**0.00019**
Ours	**0.00016**	**0.00019**

Table 6.6 Quantitative comparison on THUman1.0 dataset.

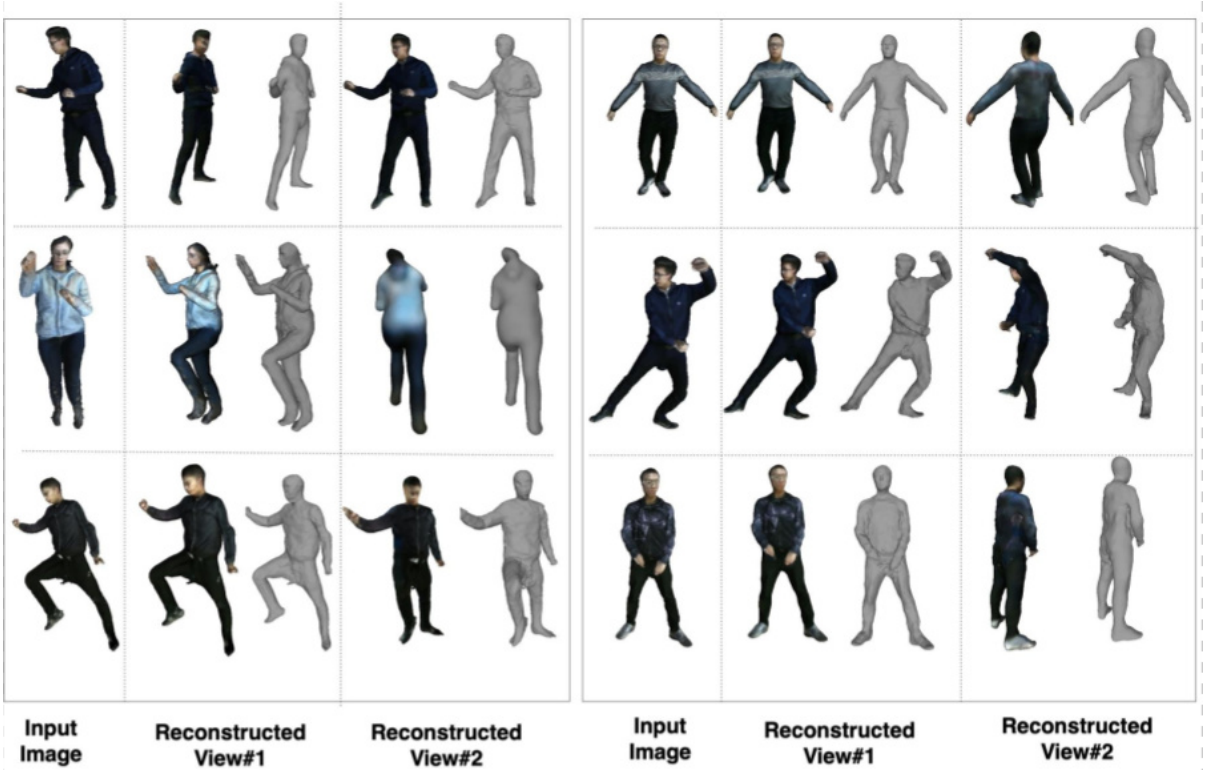

Figure 6.13 Results of SHARP on THUman1.0 dataset.

6.7 Discussion

In this section, we discuss about refining SMPL, provide an ablation on loss functions and number of ResNet blocks. We also discuss in detail about the post-processing steps, along with the limitations and failure cases of our method.

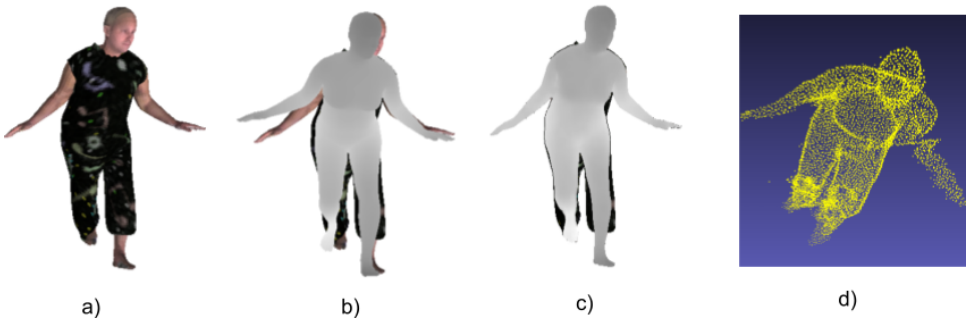

Figure 6.14 (a) Input Image, (b) Error in SMPL, (c) Error corrected by network and (d) noisy point cloud of corrected SMPL

6.7.1 SMPL refinement module:

The motivation behind this module is that the predicted SMPL may not be perfectly aligned with the input image. By getting consistent SMPL prediction, a more accurate surface inference can be obtained. Thus, when there is this misalignment, we propose to refine SMPL estimates in two steps. i) SMPL refinement network and ii) optimization framework. Note that our refinement module is a standalone approach and can be used to refine any off-the-shelf SMPL predictions.

6.7.1.1 SMPL refinement network:

The input to the refinement network is input image ($\mathcal{I}$) Figure 6.14 (a) concatenated with initial SMPL prior represented in peeled depth maps ($\mathcal{D}^{init}_{smpl}$). The misalignement is shown in Figure 6.14 (b). SMPL refinement network is trained to predict the intermediate depth peel maps $\widehat{\mathcal{D}}^{inter}_{smpl}$ as shown in Figure 6.14 (c) which are aligned to the silhouette of input image. The network is expected to solve for the misalignment between $\mathcal{I}$ and $\mathcal{D}^{init}_{smpl}$. We train our refinement network with $\mathcal{L}_ref$ between predicted and ground truth peel maps where

$$L_{ref} = \sum_{i=1}^{4} \left\| \widehat{\mathcal{D}}^{inter,i}_{smpl} - \mathcal{D}^{inter,i}_{smpl} \right\|_1 \tag{6.13}$$

where $\widehat{\mathcal{D}}^{inter,i}_{smpl} \in \widehat{\mathcal{D}}^{inter}_{smpl}$ and i is the i^{th} layer in peel maps.

6.7.1.2 Optimization framework:

The predicted $\widehat{\mathcal{D}}^{inter}_{smpl}$ is back projected to obtain point cloud $\widehat{P}$ using the camera parameters s, t_x, t_y. The point cloud is aligned to silhouette of input RGB image. Nevertheless, it does not contain plausible body shapes as shown in Figure 6.14(d). Hence, we propose to fit SMPL mesh P to this point cloud to produce final SMPL body parameters $(\beta_{final}, \theta_{final})$. Specifically, we minimize the following energy

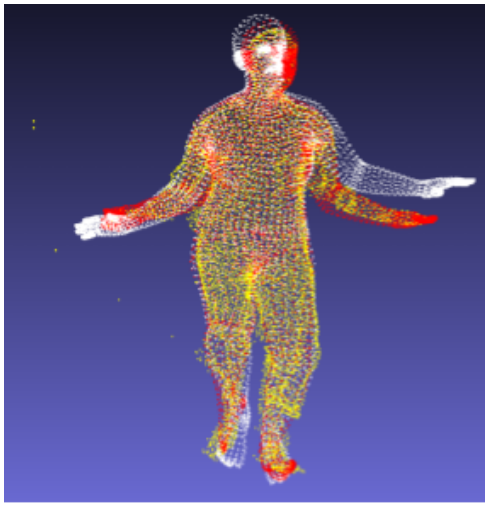

Figure 6.15 Effect of Optimization: Point clouds of initial SMPL, output of SMPL refinement network and final optimized SMPL are shown in white, yellow and red respectively.

function:

$$(\beta_{final}, \theta_{final}) =_{\beta,\theta} \mathcal{L}_{cd} + \lambda \mathcal{L}_{reg} \tag{6.14}$$

where $\mathcal{L}_{cd}$ is the chamfer distance between $\widehat{P}$ and SMPL mesh P. This loss enforces the SMPL mesh to be constrained by the point cloud whose silhouette is aligned to the input image.

$$\mathcal{L}_{cd} = \sum_{\vec{p_i} \in \widehat{\mathcal{P}}} \min_{\vec{q_j} \in \mathcal{P}} \|\vec{p_i} - \vec{q_j}\|_2^2 + \sum_{\vec{q_j} \in \mathcal{P}} \min_{\vec{p_i} \in \widehat{\mathcal{P}}} \|\vec{q_j} - \vec{p_i}\|_2^2 \tag{6.15}$$

Similar to [173], we regularize the initial predicted $\beta_{init}, \beta_{pred}$ which ensures that SMPL parameters are not deviated by large amount w.r.t input parameters.

$$\mathcal{L}_{reg} = \|\beta - \beta_{init}\|_2^2 + \|\theta - \theta_{init}\|_2^2 \tag{6.16}$$

Thus, we obtain SMPL mesh corresponding to the SMPL parameters $(\beta_{final}, \theta_{final})$ which is aligned to input image. Please refer Figure 6.15.

However, when loose clothing is tested, our SMPL refinement network fails to predict the exact shape of the body inside clothing as visualized in Figure 6.17. This results in erroneous estimate of SMPL after optimization.

6.7.2 Ablation Study: Architectural Choices

Impact of loss functions: In Table 6.4, we demonstrate the impact of various loss functions on the output point cloud. First, we evaluate SHARP without smoothness loss (L_{sm}) and observe that it leads to an increase in Chamfer distance and P2S error, which is caused by the noise in prediction of fused

Depth Prior	Residual	Fusion	CD ↓	P2S ↓
✘	✘	✘	24.22	0.0098
✓	✘	✘	9.98	0.0054
✓	✓	✘	14.11	0.0052
✓	✓	✓	**7.718**	**0.0051**

Table 6.7 Ablation Study: Importance of various components

peel maps.
Secondly, to evaluate the importance of the peel map fusion, we train our network without fusion. In this setting, we used only two decoder branches, one for predicting RGB peel maps and the other for predicting depth peel maps. This lead to smooth, predictions, which misses out body-specific geometrical details, further increasing CD and P2S values.

Impact of different components: In Table 6.7, we demonstrate the impact of various components in our method. We train the network without any depth prior ($\mathcal{D}_{smpl}$) and predict depth peel maps without fusion and we observe the performance of the network is worser than PeeledHuman which has adversarial setup. However, if depth prior is provided as additional input to this network, the performance is significantly improved indicating the impact of SMPL depth prior. If we model the final 3D body as only residual deformation ($(\mathcal{D}_{smpl} + \widehat{\mathcal{D}}_{rd})$), we observe the deterioration in Chamfer score as loose clothing details are completely neglected. Note that point-to-surface distance is not deteriorated as significantly as Chamfer score as it computes the distance from predicted point cloud to ground truth mesh and vice-versa is not computed. Finally, when we fuse ($\mathcal{D}_{fused}$) both residual and auxiliary peel maps the performance is improved further indicating the importance of our proposed fusion.

Impact of various backbone networks: We evaluate the performance of SHARP on various backbone network architectures. In particular, we used U-Net ronneberger2015u and stacked hourglass network newell2016stacked as backbone networks along with residual networks. All the backbone networks are trained with same loss functions as described in **??**. We report the performance of these networks in Table 6.8. Residual network outperforms both Unet and hourglass networks in this multi-branch prediction task. We also observed that hourglass network was not able to predict four layer RGB peel maps.

Impact of number of ResNet blocks: We also evaluate the performance of SHARP by varying the number of ResNet blocks as shown in Table 6.10. We train our network on 3DHumans with 6, 9 and 18 blocks. Using only 6 ResNet blocks, which is almost one-third of the original network, SHARP is able to achieve similar performance as PIFu (please refer Table 6.1). Using 9 ResNet blocks, we are able to

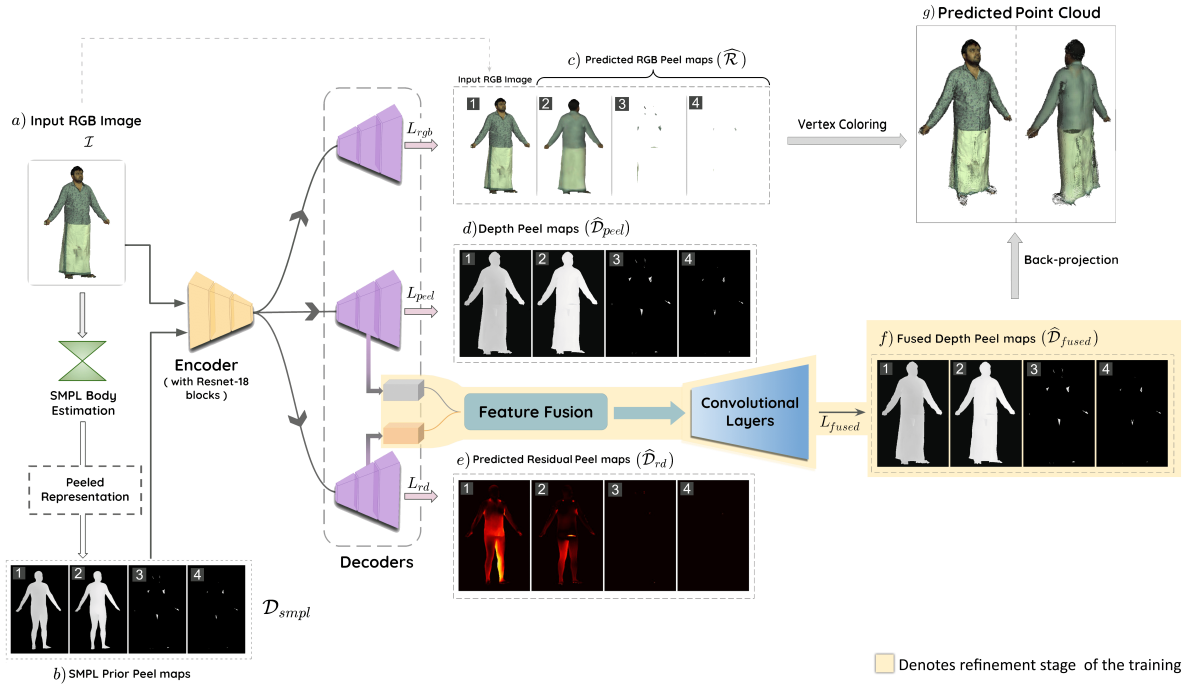

Figure 6.16 Architecture for feature fusion module.

achieve closer numbers to majority of existing SOTA methods. We observed that the further increase in the number of ResNet blocks did not yield any significant improvement.

Fusion Strategies We analyse the performance of SHARP with various fusion strategies of peel maps. In this experiment, we perform feature level fusion instead of auxiliary and residual depth peel map fusion. We train this fusion network in a coarse-to-fine strategy where initially we replace the auxiliary peel map branch with predicting complete depth peel maps $\widehat{D}_{peel}$. We train this network with losses L_{rd}, L_{sm} and L_1 loss on predicted and ground truth peel maps. We then, take this network as initialization to train fusion module where we take intermediate features of $\widehat{D}_{peel}$ and $\widehat{D}_{rd}$ branches respectively as shown in Figure 6.16. We fuse them using three strategies (a) addition, (b) average and (c) concatenation. These fused features are then passed to upsampling and convolutional layers to predict final fused depth peel maps. Here, we freeze the weights of the network except the layers after the feature fusion. We call it as *Late Fusion* as it requires pre-trained network.

We report the performance in Table 6.9 and learn that late fusion with concatenation results in better performance. However, we note the training for late fusion is not end-to-end as described and we adopt end-to-end trainable network with fusion proposed in Equation 6.5 as our final choice.

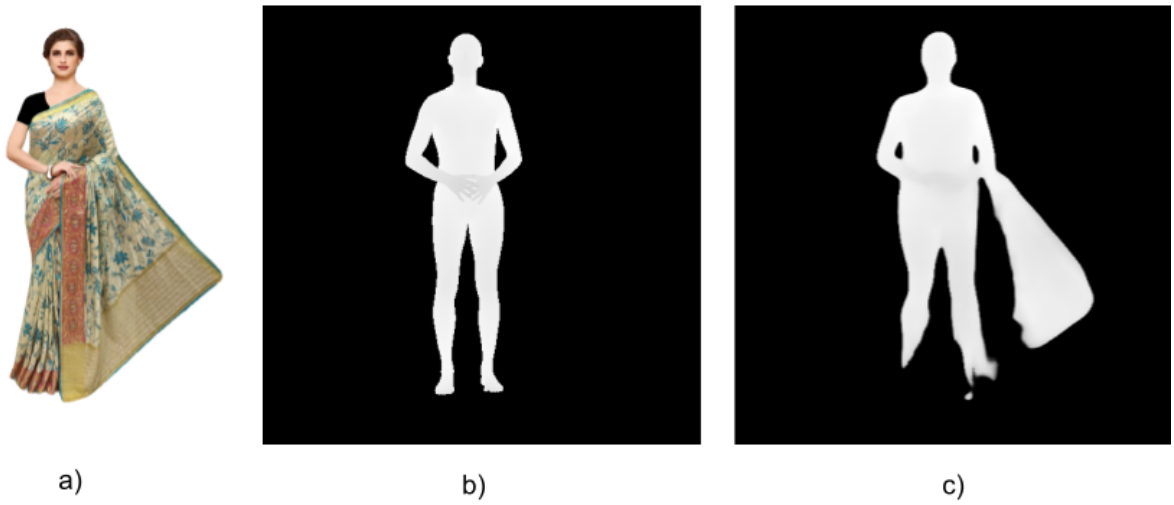

Figure 6.17 Failure case: (a) Input Image, (b)Initial SMPL, (c) Predicted by SMPL refinement network

Network	CD ↓	P2S ↓
U-Net	8.417	0.0052
Hourglass	15.6	0.0068
ResNet(ours)	**7.71**	**0.0051**

Table 6.8 Effect of different architectures.

Network	CD ↓	P2S ↓
Addition	8.24	0.0052
Average	8.82	0.0058
Concat	**7.57**	**0.0049**
Ours*	7.71	0.0051

Table 6.9 Comparison of various Fusion Strategies. Ours is only end-to-end trainable mechanism as opposed to Addition, Average and Concat fusion.

6.7.3 Robustness to noise in SMPL

Thus, our method had an advantage as we used the ground-truth SMPL prior. Nevertheless, to test the sensitivity of SHARP on the accuracy of SMPL prior, we induce additive random Gaussian noise with zero mean and three variances $(0.005, 0.01, 0.1)$ in ground-truth SMPL pose parameters, as shown in Figure 6.18. We compute and plot the P2S distance between the predicted point-clouds with noisy input SMPL versus predicted point-cloud without noise, as shown in Figure 6.18. As one can observe, our method can recover from Gaussian noise in SMPL prior. Also, note that the regions near legs are largely unaffected as the noise in SMPL prior is compensated by auxiliary peel maps in the clothed region during fusion.

Blocks	parameters	CD ↓	P2S ↓
6	8.26M	22.81	0.0073
9	12.17M	8.9	0.0053
18	22M	**7.71**	**0.0051**

Table 6.10 Effect of ResNet blocks.

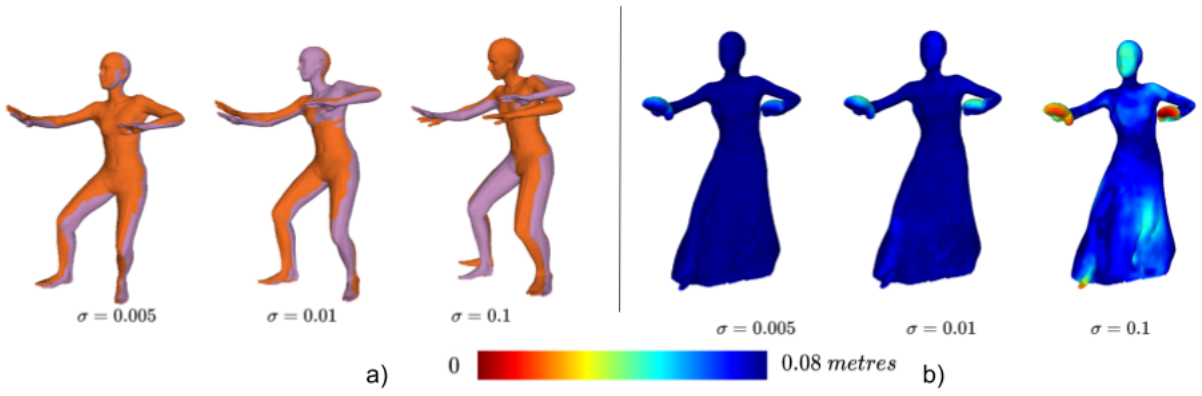

Figure 6.18 Sensitivity to noise in SMPL prior.

6.7.4 Handling noisy shape prior

The shape prior based reconstruction methods are susceptible to noisy initialization from incorrect prior. Generally, this leads to incorrect pose conditioning, which further deteriorates the final reconstruction. Our method can partially handle such noisy prior as we use refined per-layer SMPL mask Γ_i (introduced in subsubsection 6.3.1.1, Equation 6.2) to mask out the regions of the SMPL prior peel maps which fall outside the input image human silhouette. Thus, residual deformation $\widehat{\mathcal{D}}_{rd}$ predicted for the misaligned regions of the SMPL prior is not considered during fusion, and we are able to avoid the errors in reconstruction due to such misalignment. Figure 6.19 depicts a noisy SMPL prior examples and final reconstruction results where our method is able to recover from incorrect prior in leg region.

6.7.5 Post-processing

The output of our network is prone to slight noise in the predicted peel maps, resulting in sparse outliers in the back-projected point cloud, as shown in Figure 6.21 (a). These outliers are removed by density-based filtering, where we fit spheres with 16 neighbours on each point. The points, which are inside the spheres having a radius greater than the threshold (0.01 in our case), are removed to obtain a clean point cloud, as shown in Figure 6.21 (b). Finally, the filtered point cloud might have some

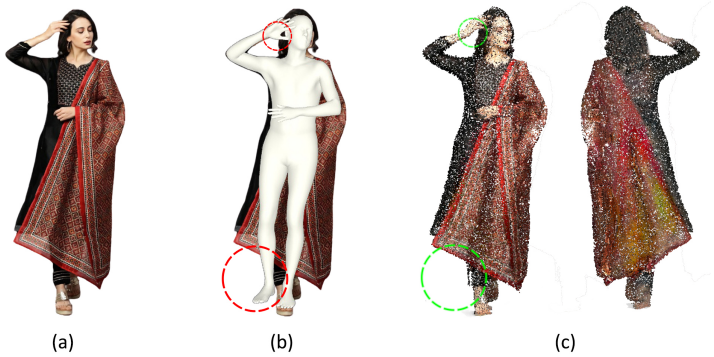

Figure 6.19 Handling noisy shape prior: (a) Input image, (b) SMPL prior misaligned with the input image, (c) Point cloud output from SHARP.

small holes which are subsequently filled by meshification using Poisson Surface Reconstruction (PSR) Figure 6.21 (c).

6.7.6 Limitations

Ambiguity due to textural edges: 3D reconstruction from a monocular RGB image, being an ill-posed problem, is susceptible to interpreting the textural edges as geometrical details. In Figure 6.20, we show reconstructions from our method and PaMIR, where both the methods incorrectly interpret textural details of flat clothing surface as geometrical details and hallucinate geometrical structures, which are non-existent.

Peeled representation layers: In this work, we use 4 layers of peeled representation for human body recovery. While a majority of the poses in real-world scenarios can be modeled using 4 layers, there can be few poses/viewpoints which require more than 4 layers to model. Although, our formulation can be extended to any number of layers, training a network to predict more than 4 peel maps would require significantly larger training data containing such rare and complex poses to generalize well.

Failure cases: One of the key challenges faced by majority of existing prior-based methods is self-intersection of body parts in the prior, mainly due to challenging pose. In Figure 6.22, a failure case of our approach is shown where the network reconstructs the occluded regions well, but fails to recover from interpenetrating body parts, present in the input SMPL prior (hands intersecting with the legs).

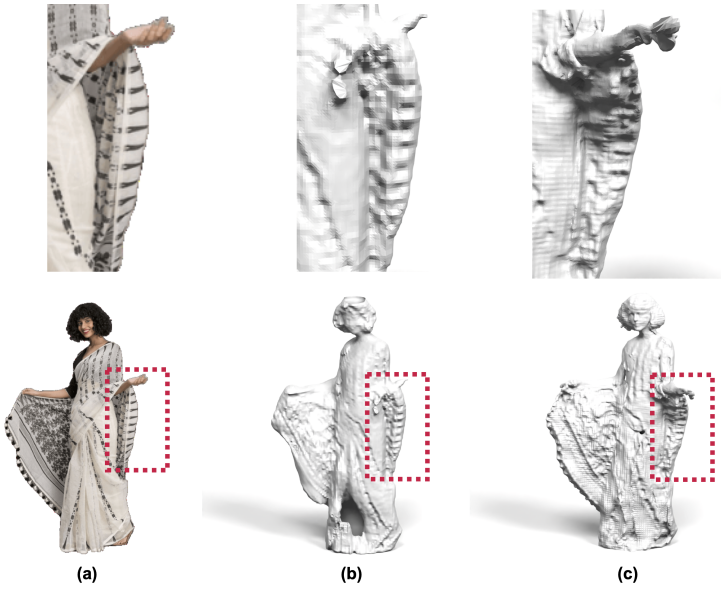

Figure 6.20 Texture-Geometry Ambiguity: High-frequency textural details can be interpreted as geometrical details by monocular deep reconstruction techniques. (a) Input image, (b) PaMIR, (c) SHARP.

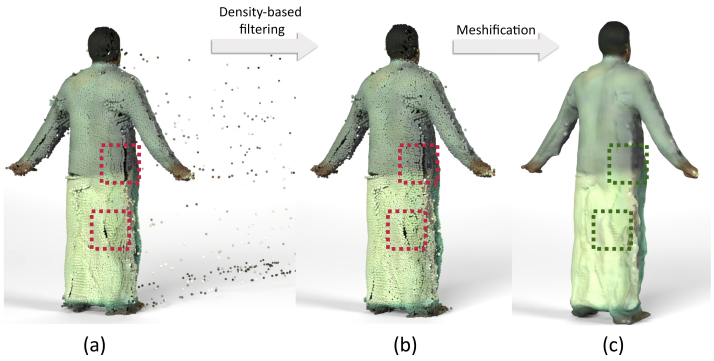

Figure 6.21 Effect of post-processing.

6.8 Summary

Reconstructing 3D clothed human bodies using monocular RGB images is an extremely ill-posed problem due to skewed viewpoints, depth ambiguities, complex poses and arbitrary clothing styles. Although many solutions exist which can recover clothed human body, they fail to generalize on in-the-wild loose clothing scenarios. To this end, we have contributed a novel end-to-end trainable deep learning framework, SHARP, which uses a sparse and efficient fusion of parametric body prior with non-parametric PeeledHuman representation, and is able to reconstruct human body in arbitrarily loose clothing.

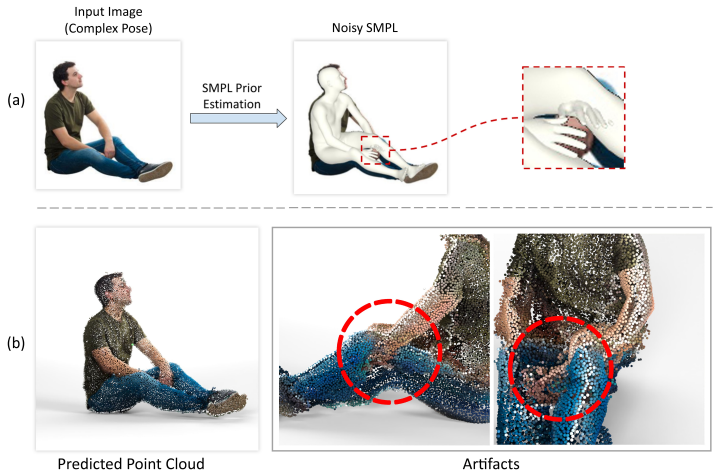

Figure 6.22 Failure Case : **(a)** Noisy SMPL estimation (hands are intersecting with the legs) due to highly complex pose. **(b)** Artifacts in the predicted point cloud.

In more general perspective, we built on our sparse non-parametric 2D shape representation and proposed an efficient strategy to fuse it with parametric shape prior using a set of simple 2D loss functions while reconstructing complex 3D geometry from a monocular image. The proposed formulation is sparse in terms of representation, with low compact network size and low inference time. We evaluated our framework on various publicly available datasets and reported superior qualitative and quantitative performance as compared to state-of-the-art methods. Since, data is a key bottleneck in the field of 3D human body reconstruction, we contributed 3DHumans dataset in public domain to accelerate the further research. Our dataset contains 3D human body scans of high-frequency textural and geometrical details with a wide variety of the body shapes in various garment styles.

Although per-frame reconstruction of SHARP yields reasonable intra-frame consistency without any explicit temporal conditioning (as shown in the supplementary video), it will be interesting to explore extension of our method to learn over video sequences where it is difficult to get high quality ground-truth data. Another interesting direction is to incorporate learning from multi-view images for better reconstruction results. Additionally, performance of our method can be further improved by addressing the texture-geometry ambiguity and recovering from challenging scenarios such as self-intersecting body parts.

Chapter 7

Datasets

The success of 3D human body reconstruction from monocular images is highly dependent of the data on which the deep learning models are trained. The data plays a crucial role in generalizability of these models to unseen real world images. To this extent, as part of this thesis, we propose two datasets of 3D human body scans. Many of the existing datasets are either synthetic or with relatively tight clothing. In this thesis, we create datasets consisting of 3D scans which are as close as to real life humans.

7.1 Multi-camera calibrated dataset

We setup a multiple synchronized calibrated Kinect V2 [57] cameras to capture human bodies in action as shown in Figure 7.1. In this setup, a total of 15 mesh sequences, each containing 200 to 300 frames with significant pose and shape variation were captured. We also captured with wide variety of clothes ranging from tight clothing to loose clothing as shown in Figure 7.2. Below we provide the details of the sensor we used and later explain the capture setup.

7.1.1 Kinect

Kinect device introduced by Microsoft provides RGB and depth cues. Kinect V2 uses a wide-angle time-of-flight camera, and processes 2 gigabits of data per second. A time-of-flight camera (ToF camera) is a range imaging camera system employing time-of-flight techniques to resolve distance between the camera and the subject for each point of the image, by measuring the round trip time of an artificial light signal provided by a light source. The Kinect V2 has greater accuracy with three times the fidelity over its predecessor and can track without visible light by using an active IR sensor. The color camera captures 1920×1080 video at 30 fps that can be displayed in the same resolution as the viewing screen, allowing for a broad range of scenarios. The infrared depth camera captures depth at 512×424 which can be warped to RGB camera using factory calibration parameters.

Figure 7.1 Our capture setup

7.1.2 Our Setup

We setup a calibrated multi-camera 3D Capture System with 5 v2 Microsoft Kinect cameras vertically mounted on stands, all connected to a workstation grade server to store and process the captured data in real time, from each of the devices. The setup and working of our system can be encapsulated in the 4 stages -

- *Synchronization:* We have implemented a software based trigger to manually synchronize these five devices. After this synchronization, all devices capture RGBD frames at same time stamp. Note that the current version of Kinect i.e. Azure kinect devices are provided with hardware based synchronization by the manufacturer which increases the reliability in multi-camera setup.

- *Calibration:* We use checkerboard based calibration to find the extrinsic matrices of these devices. We calibrate each pair of kinect cameras individually. We used implementation provided by OpenCV [1] where we first detect corners of the checkerboard and solves the camera equation using homography and 3D conics.

- *Data Capture:* Each Kinect camera captures the RGB and Depth image in synchronization. The Depth image provides the point cloud in camera frame using the camera's intrinsic parameters. The transformation matrices between each pair of cameras obtained in the calibration process are used to merge the view-specific partial point clouds to get a complete colored 3D point cloud.

- *Mesh Generation:* The 3D point clouds obtained in the previous step are often noisy due to errors in calibration, sensor noise, etc. We clean these point clouds by fitting a spherical ball with 30 neighbours. We remove the points with radius of this ball greater than threshold. This cleaned 3D point cloud is converted to a mesh using surface reconstruction algorithms such as Poisson's Algorithm to obtained a coloured 3D mesh.

[1] https://opencv.org

Figure 7.2 Our sample reconstructions

7.1.3 Caveats

Kinect V2 has very low resolution of depth when compared to RGB resolution. The depth resolution is not sufficient to capture intricate details of face and hand geometry which leads to treat them as just blobs. Additionally, due to absence of hardware based synchronization, registration issues are predominant leading to misalignment between frames of the cameras. Also, we observed significant noise in the depth sensed.

7.2 3DHumans dataset

As mentioned in section 6.1, one of the key bottlenecks that hinder progress in the field of 3D human body reconstruction is the lack of real-world datasets that contains high-frequency texture and geometrical details. To address the aforementioned issues of dataset captured using Kinect V2 sensors, we created high resolution scans of people in loose clothing with varied poses and shape. We captured this dataset using Artec Eva structured light sensor.

7.2.1 Dataset Details

We present 3DHumans, a dataset of around 250 scans containing people in diverse body shapes in various garments styles and sizes. We cover a wide variety of clothing styles ranging from loose robed clothing like saree (a typical South-Asian dress) to relatively tight-fitting shirt and trousers, as shown in Figure 7.3. The dataset consists of around 150 male and 50 unique female subjects. Total male scans are about 180 and female scans are around 70. In terms of regional diversity, for the first time, we capture body shape, appearance and clothing styles for the South-Asian population. We will release this data in the public domain for academic use.[2]

[2]http://cvit.iiit.ac.in/research/projects/cvit-projects/sharp-3dhumans-a-rich-3d-dataset-of-scanned-humans

Figure 7.3 Our dataset: Meshes visualized with (a) texture, (b) geometry

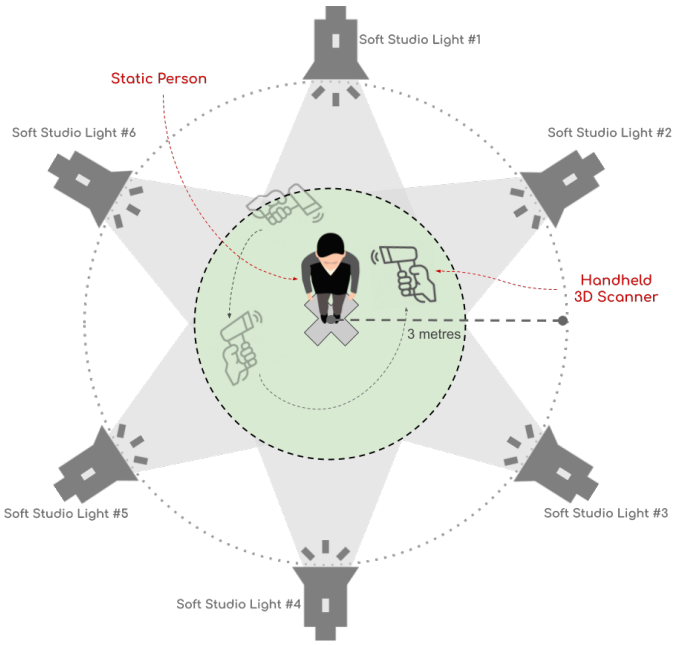

Figure 7.4 Our data capture setup

Category	Number of scans
Moderate	161
Loose	40
Robed	44

Table 7.1 Distribution of clothes

7.2.2 Data Capture

Figure 7.4 outlines our capture setup for 3DHumans dataset where we place the subject at the center of circle with diameter 1.5 meters. We scan the subject using the hand held scanner from the circle covering 360° (including top and bottom of the subject). To uniformly lit the the subject, we place lights on a circle which is 3 meters in diameter as shown in Figure 7.4. The data captured by scanner is transferred to PC. We then use the commercial software [3] for post-processing the scans which includes the following steps. Initially, we clean the noise in scans including the base removal. Further, multiple partial scans of the same subject are aligned followed by registration to recover single complete mesh of the subject. We later fill holes if any to obtain the final mesh. Finally, we obtain the texture map of the mesh at 2048×2048 resolution. The sensor for capture acquisition and post-processing we applied is further detailed in section 7.3 Each of the mesh obtained using the post-processing above is oriented differently. We align all the meshes using "Align Tool" in meshlab. For each 3D human scan, we also provide the SMPL body aligned to it, computed using [173, 110].

7.2.3 Clothing style and pose details

In Table 7.1, we present the number of scans categorized according to the clothing style. Clothing styles like shirts, trousers, shorts etc., which are moderately loose are included in "Moderately loose clothing". Clothing styles which cover portion of body at least till knees such as kurtha, pyjamas, dresses are included in "Loose clothing" category. Very loose clothing such as sarees [4], lungi [5], long dresses etc. which covers entire body are included in "Robed clothes" category.

We have wide variety of poses of standing people in the dataset. Specifically, we have 71 scans where the hands are open as in A-pose and T-pose and remaining scans are with different varying poses.

[3] https://www.artec3d.com
[4] https://en.wikipedia.org/wiki/Sari
[5] https://en.wikipedia.org/wiki/Lungi

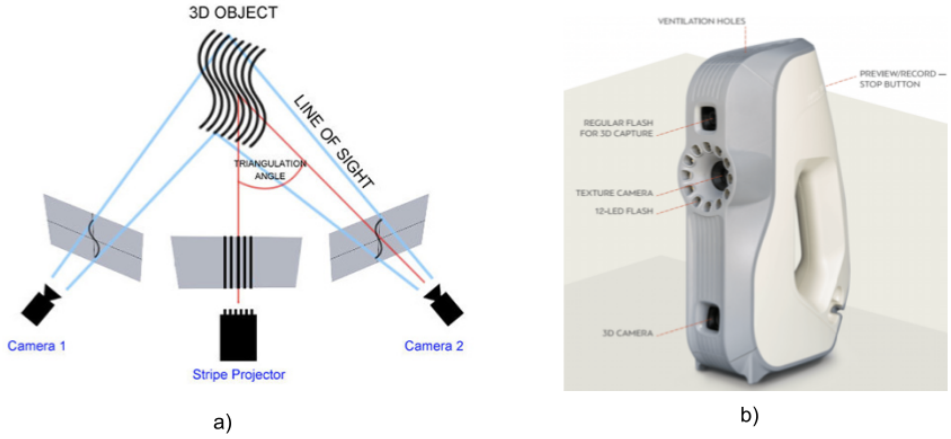

Figure 7.5 (a) A typical structured light sensor (Figure adopted from 3dnatives), (b)Artec Eva scanner (Figure adopted from artec)

7.3 Structured light sensor

These sensors use trigonometric triangulation of a light pattern projected onto the object to scan. The pattern is projected on the object using an LCD projector or some other source of stable light. Multiple cameras typically 2 which are situated with slight offset from the projector. These cameras look at the shape of the pattern of light and calculate the distance of every point in the field of view. The structured light used in the scanning process can be white or blue and the pattern of light usually consists of a series of stripes, but can also consist of a matrix of dots or other shapes. With finely calibrated stripes and accurate cameras, it's possible to measure the dimensions of very small details even the minute variations on face. Unlike laser scanners, structured light sensors are not disrupted by reflective surfaces.

7.3.1 Capture with Artec Scanner

We use Artec 3D Eva lite handheld scanner equipped with texture camera. The scanner has a 3D point accuracy of up to 0.1mm and 3D resolution is 0.5mm. It provides hybrid geometry and texture based tracking. The texture resolution captured at each frame is 1.3 megapixel and the 3D reconstruction rate is upto 16 fps. The scanner is connected to external computational server which processes the raw data using Artec Studio 15 software. Below we outline the steps involved to process the data:

- *Alignment:* Although the sensor allows continuous scanning of entire 3D human body, there may be cases where we need multiple scans of the same subject. To assemble all scans into a single whole, we must align all the data to a single coordinate system. The texture provides additional cue which eases the alignment process. It uses texture-image characteristics of scanned objects and greatly decreases the possibility of incorrect alignment.

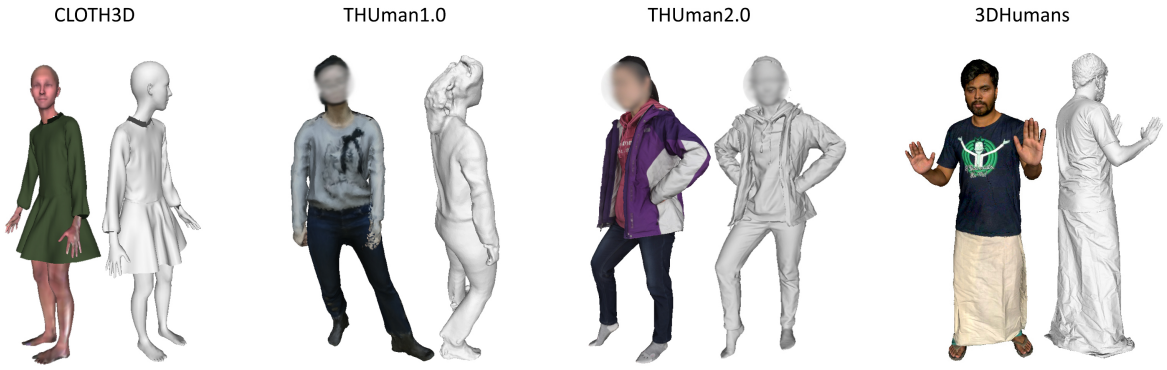

Figure 7.6 A sample from each dataset.

- *Global Registration:* After alignment of multiple scans, global registration is performed which brings all single frame surfaces to a single coordinate system using information on the mutual position of each surface pair. To do so, it selects a set of special geometry points on each frame, followed by a search for pair matches between points on different frames.
- *Fusion:* This stage creates the polygonal mesh model from the output of global registration phase. Artec Studio provides fast, smooth and sharp fusion mechanisms. We use sharp fusion as it preserves fine details and sharp edges to maximum extent. However, this fusion is expensive in time (5 minutes) per scan. We also apply hole filling of the meshes obtained to remove any small holes further.
- *Texture:* After the mesh is obtained, texturing process projects textures from the individual frames onto the fused mesh. For each scan, we output 4K resolution texture map which is very high resolution and preserves fine textural details.

For each 3D human scan, we also provide the SMPL body aligned to it, registered using [173]. In Figure 7.6, we show the samples provided by other datasets and visualize the sample from our dataset. Cloth3D [11] provides synthetic clothed models which lack realism. THUman1.0 [174] provides models scanned from Kinect which looses many fine details of both texture and geometry. THUman2.0 [169] provides data with multiple calibrated cameras which are prone to errors. On the other hand, our dataset provides meshes with fine details of geometry captured from a sensor which is of high resolution.

Chapter 8

Conclusion and future work

In this thesis, we conclude this thesis by discussing the contributions, impact and comparisons of our proposed approaches with contemporary methods. Finally, we also provide the future directions of this dissertation at the end of this chapter.

8.1 Discussion

This thesis attempted to propose classification of rigid objects and then primarily focused on reconstruction of non-rigid shapes, in particular human bodies with loose clothing. In particular, we attempt to design and implement algorithms which enhances the capability of machines to interpret 3D human poses and shapes from a monocular image such that it has wide-scale applicability. In doing so, we worked with different representation of 3D data and understood the drawbacks of existing representations. We proposed a new representation which addresses these drawbacks.

Initially, we proposed SplineNet (chapter 3), a novel learning paradigm to address the challenging issues of efficient and effective 3D volumetric data classification. We proposed an efficient descriptor for 3D shape classification using neural network. In particular, to account for local geometric variations while generating global representation of 3D data, we introduce B-spline surfaces in SplineNet. The locations of these surfaces are learned from data and analytical solutions to perform back propagation are derived. The B-spline layer introduced in the network extracts local geometry aware global representation of 3D shapes using differentiable vector field.

The problem of 3D human body reconstruction from monocular images is quite challenging with several challenges to address as mentioned in chapter 1. We presented three effective solutions for this problem. Firstly, we disentangle 3D body reconstruction and texture using two separate networks (chapter 4). For extracting surface, we utilized volumetric representation. Given an input image, we predict voxel grid. However, volumetric representation poses severe computational disadvantage. This problem arises because deep neural network probes every location in volumetric grid which results in wastage of computation. Additionally, 3D convolution is also done in inside the surface which leads to redundant computation. Also, the output of shape depends on the resolution of voxel grid. However, more the

voxel resolution, more the memory consumption and networks find difficult to converge. Additionally, texture from VAE produces 64×64 resolution which is later upsampled using SRGAN. Thus, to achieve textured 3D model we need three networks.

Secondly, we addressed the drawbacks of earlier approach by proposing a novel PeeledHuman representation (chapter 5) to reconstruct textured human model from a single RGB image. PeeledHuman encodes the human body as a set of Peeled Depth and RGB maps in 2D, obtained by performing ray-tracing on the 3D body model and extending each ray beyond its first intersection. Such an encoding is robust to severe self-occlusions while being accurate and efficient at learning & inference time. Our end-to-end framework PeelGAN predicts these Peeled maps and has low inference time and recovers accurate 3D human body models from a single view. On the flip side, PeelGAN might result in distorted body parts as there is no body shape regularization provided to the network.

We addressed the distorted body parts issue by introducing body shape prior while reconstructing peel maps in SHARP (chapter 6). We propose a sparse and efficient fusion of parametric SMPL body prior with non-parametric PeeledHuman representation, and is able to reconstruct human body in arbitrarily loose clothing. The proposed formulation is sparse in terms of representation, with low compact network size and low inference time. Finally, we contributed 3DHumans dataset (chapter 7) in public domain to accelerate the further research. Our dataset contains 3D human body scans of high-frequency textural and geometrical details with a wide variety of the body shapes in various garment styles.

Impact of this thesis: This thesis advances the field of 3D human body reconstruction significantly. The proposed methods achieve noticable improvement on 3D reconstruction accuracies and performance on multiple datasets. Many of the methods proposed in this thesis assume minimal assumptions. These methods enable 3D human body reconstruction in calibration-free and in-the-wild images. 3D human body reconstruction is a very recent field and many of the existing datasets available in public domain are either synthetic or does not mimic the real world clothing. As part of this thesis, we have introduced couple of datasets which helps in accelerating the research in this domain further. This thesis paves way for many more interesting problems in this area which are discussed below.

8.2 Future directions

In this thesis, we focused on 3D reconstruction of human body with loose clothing. Animating these loose clothed models is a mammoth task. Learning the dynamics of cloth as function of rigid body movements and external factors like environmental and physical constraints, type of fabric etc., might result in a realistic animation of the recovered models. Recent surge in AR/VR devices has open doors to many interesting problems virtual setup. Apart from human body reconstruction from monocular images, understanding human-object and human-scene interaction play a vital role in enhancing virtual reality experiences. Next, in this thesis, we focused on complete 3D human body estimating with less focus on face and hands. However, reconstructing face and hands from monocular images and further

animating them is a challenging research problem in itself with potential applications. Further, with the advent of diffusion models [71], image generation is possible from multi-modal linguistic models. Similar generative modelling can be thought of generating human body meshes where user can create 3D meshes which can be further used in many different applications.

Specific to this thesis, promising avenues for future research are listed as follows:

- **Temporal reconstruction of human bodies:** We can extend our work to incorporate temporal consistency of 3D human bodies. Existing works along this direction assume a template 3D model which they deform using image evidence. However, deforming a template model can lead to unrealistic deformation. Our PeeledHuman representation can be an ideal representation in this context as we can model unconstrained transformation.

- **Multiple images integration:** As discussed in chapter 5, PeeledHuman misses out triangles which are tangential to the surface. However, this missing information is available in another view. Integrating multiple views can enhance the final reconstruction.

- **Multiple humans in a scene:** One interesting direction is to reconstruct a scene with multiple humans. This is a significant challenging scenario as there can occlusion within one subject and across subjects as well. As PeeledHuman is robust to self-occlusions within subject, it can be a good starting point for research in this direction.

- **Temporal Data Capture:** Till date, there are only a few 4D reconstruction datasets which satisfies all the needs. Installing and capturing the 4D template-free dataset is really valuable to community which can significantly accelerate the research in AR/VR problem setups.

Related publications

- **Sai Sagar Jinka** and Avinash Sharma, "SplineNet: B-spline neural network for efficient classification of 3D data", **Indian Conference on Computer Vision, Graphics and Image Processing, 2018 (Long oral)**.
- Abbhinav Venkat, **Sai Sagar Jinka** and Avinash Sharma, "Deep Textured 3D Reconstruction of Human Bodies", **British Machine Vision Conference, 2018**
- **Sai Sagar Jinka**, Rohan Chacko, Avinash Sharma and P.J.Narayan, "PeeledHuman: Robust Shape Representation for Textured 3D Human Body Reconstruction", **International Conference on 3D Vision, 2020 (Spotlight)**
- **Sai Sagar Jinka**, Astitva Srivastava, Chandradeep Pokariya, Avinash Sharma and P.J.Narayan, "SHARP: Shape-Aware Reconstruction of People in Loose Clothing", **International Journal of Computer Vision, 2022**

Other conference/workshop publications that are not part of this thesis:

- Snehith Routhu, **Sai Sagar Jinka** and Avinash Sharma,"Coarse-to-fine 3D Clothed Human Reconstruction using Peeled Semantic Segmentation Context", **Indian Conference on Computer Vision, Graphics and Image Processing, 2021 (oral)**
- Snehith Routhu, **Sai Sagar Jinka** and Avinash Sharma,"REF-SHARP: REFined face and geometry reconstruction of people in loose clothing", **Indian Conference on Computer Vision, Graphics and Image Processing, 2022**
- Astitva Srivastava, Chandradeep Pokhariya, **Sai Sagar Jinka** and Avinash Sharma, "xCloth: Extracting Template-free Textured 3D Clothes from a Monocular Image", **ACM MultiMedia 2022**
- Surabhi Guptha, **Sai Sagar Jinka**, Avinash Sharma and Anoop Namboodiri. "Supervision by Landmarks: An Enhanced Facial De-occlusion Network for VR-based Applications", **European Conference on Computer Vision (ECCVW) Workshop and Challenge on People Analysis:**

Bibliography

[1] T. Alldieck, M. Magnor, B. L. Bhatnagar, C. Theobalt, and G. Pons-Moll. Learning to reconstruct people in clothing from a single RGB camera. In *CVPR*, 2019.

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

[3] T. Alldieck, M. A. Magnor, W. Xu, C. Theobalt, and G. Pons-Moll. Video based reconstruction of 3d people models. *CoRR*, abs/1803.04758, 2018.

[4] T. Alldieck, G. Pons-Moll, C. Theobalt, and M. Magnor. Tex2Shape: Detailed full human body geometry from a single image. In *ICCV*, 2019.

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

[6] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005.

[7] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *ACM Transaction on Graphics*, 24:408–416, 2005.

[8] V. Arvind, A. Costa, M. Badgeley, S. Cho, and E. Oermann. Wide and deep volumetric residual networks for volumetric image classification. *arXiv preprint arXiv:1710.01217*, 2017.

[9] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1626–1633. IEEE, 2011.

[10] T. C. Azevedo, J. M. R. Tavares, and M. A. Vaz. *3D Object Reconstruction from Uncalibrated Images Using an Off-the-Shelf Camera*, pages 117–136. 2009.

[11] H. Bertiche, M. Madadi, and S. Escalera. Cloth3d: Clothed 3d humans. In *European Conference on Computer Vision*, pages 344–359. Springer, 2020.

[12] B. L. Bhatnagar, C. Sminchisescu, C. Theobalt, and G. Pons-Moll. Combining implicit function learning and parametric models for 3D human reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[13] B. L. Bhatnagar, C. Sminchisescu, C. Theobalt, and G. Pons-Moll. LoopReg: Self-supervised learning of implicit surface correspondences, pose and shape for 3D human mesh registration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[14] B. L. Bhatnagar, G. Tiwari, C. Theobalt, and G. Pons-Moll. Multi-Garment Net: Learning to dress 3D people from images. In *ICCV*, 2019.

[15] F. Bogo, M. J. Black, M. Loper, and J. Romero. Detailed full-body reconstructions of moving people from monocular rgb-d sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2300–2308, 2015.

[16] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision (ECCV)*, 2016.

[17] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016.

[18] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[19] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA, June 2014. IEEE.

[20] F. Bogo, J. Romero, M. Loper, and M. J. Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3794–3801, 2014.

[21] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black. Dynamic FAUST: Registering human bodies in motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

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

[23] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 3189–3197, 2016.

[24] A. Brock, T. Lim, J. M. Ritchie, and N. Weston. Generative and discriminative voxel modeling with convolutional neural networks. corr abs/1608.04236 (2016), 2016.

[25] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008.

[26] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *CoRR*, abs/1312.6203, 2013.

[27] C. Canton-Ferrer, J. R. Casas, and M. Pardas. Marker-based human motion capture in multiview sequences. *EURASIP Journal on Advances in Signal Processing*, 2010:1–11, 2010.

[28] Y. Chen, T.-K. Kim, and R. Cipolla. Inferring 3d shapes and deformations from single views. In *European Conference on Computer Vision*, pages 300–313. Springer, 2010.

[29] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.

[30] G. K. Cheung, S. Baker, and T. Kanade. Visual hull alignment and refinement across time: A 3d reconstruction algorithm combining shape-from-silhouette with stereo. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–375. IEEE, 2003.

[31] J. Chibane, T. Alldieck, and G. Pons-Moll. Implicit functions in feature space for 3D shape reconstruction and completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[32] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *European Conference on Computer Vision (ECCV)*, pages 628–644, 2016.

[33] C. S. Chua and R. Jarvis. Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision*, 25(1):63–85, 1997.

[34] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (ToG)*, 34(4):1–13, 2015.

[35] Dawson-Haggerty et al. Trimesh library, 2019.

[36] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *CoRR*, abs/1606.09375, 2016.

[37] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, et al. Fusion4D: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (ToG)*, 2016.

[38] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, and S. Izadi. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transaction on Graphics*, 35(4):114:1–114:13, July 2016.

[39] J.-S. Franco, M. Lapierre, and E. Boyer. Visual shapes of silhouette sets. In *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, pages 397–404. IEEE, 2006.

[40] Y. Furukawa and J. Ponce. Carved visual hulls for image-based modeling. In *European Conference on Computer Vision*, pages 564–577. Springer, 2006.

[41] V. Gabeur, J.-S. Franco, X. Martin, C. Schmid, and G. Rogez. Moulding humans: Non-parametric 3D human shape estimation from single images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[42] V. Gabeur, J.-S. Franco, X. Martin, C. Schmid, and G. Rogez. Moulding humans: Non-parametric 3d human shape estimation from single images. In *ICCV*, 2019.

[43] T. Gatzke, C. Grimm, M. Garland, and S. Zelinka. Curvature maps for local shape comparison. In *Shape Modeling and Applications, 2005 International Conference*, pages 244–253. IEEE, 2005.

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

[45] K. Gong, X. Liang, Y. Li, Y. Chen, M. Yang, and L. Lin. Instance-level human parsing via part grouping network, 2018.

[46] P. Guan, A. Weiss, A. O. Balan, and M. J. Black. Estimating human shape and pose from a single image. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1381–1388. IEEE, 2009.

[47] M. Habermann, W. Xu, M. Zollhoefer, G. Pons-Moll, and C. Theobalt. Livecap: Real-time human performance capture from monocular video. *ACM Transactions On Graphics (TOG)*, 38(2):1–17, 2019.

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

[49] N. Hasler, H. Ackermann, B. Rosenhahn, T. Thormählen, and H.-P. Seidel. Multilinear pose and body shape estimation of dressed subjects from image sets. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1823–1830. IEEE, 2010.

[50] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[51] T. He, J. Collomosse, H. Jin, and S. Soatto. Geo-PIFu: Geometry and pixel aligned implicit functions for single-view human reconstruction. In *Advances in Neural Information Processing Systems*, 2020.

[52] L. Herda, P. Fua, R. Plankers, R. Boulic, and D. Thalmann. Skeleton-based motion capture for robust reconstruction of human motion. In *Proceedings Computer Animation 2000*, pages 77–83. IEEE, 2000.

[53] D. Holz and S. Behnke. Fast range image segmentation and smoothing using approximate surface reconstruction and region growing. In *Intelligent autonomous systems 12*, pages 61–73. Springer, 2013.

[54] Z. Huang, T. Li, W. Chen, Y. Zhao, J. Xing, C. LeGendre, L. Luo, C. Ma, and H. Li. Deep volumetric video from very sparse multi-view performance capture. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[55] Z. Huang, Y. Xu, C. Lassner, H. Li, and T. Tung. ARCH: Animatable reconstruction of clothed humans. In *CVPR*, 2020.

[56] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[57] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011.

[58] A. S. Jackson, C. Manafas, and G. Tzimiropoulos. 3D human body reconstruction from a single image via volumetric regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[59] S. S. Jinka, R. Chacko, A. Sharma, and P. Narayanan. PeeledHuman: Robust shape representation for textured 3D human body reconstruction. In *Proceedings of the IEEE Conference on 3D Vision (3DV)*, 2020.

[60] S. S. Jinka and A. Sharma. Splinenet: B-spline neural network for efficient classification of 3d data. In *Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing*, ICVGIP 2018, New York, NY, USA, 2020. Association for Computing Machinery.

[61] S. S. Jinka, A. Srivastava, C. Pokhariya, A. Sharma, and P. Narayanan. Sharp: Shape-aware reconstruction of people in loose clothing. *International Journal of Computer Vision*, 131(4):918–937, 2023.

[62] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (5):433–449, 1999.

[63] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[64] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[65] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[66] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik. Learning category-specific mesh reconstruction from image collections. *arXiv preprint arXiv:1803.07549*, 2018.

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

[68] A. Kanezaki, Y. Matsushita, and Y. Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. *arXiv preprint arXiv:1603.06208*, 2016.

[69] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

[70] M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(3):29, 2013.

[71] D. Kingma, T. Salimans, B. Poole, and J. Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

[72] R. Klokov and V. Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 863–872. IEEE, 2017.

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

[74] N. Kolotouros, G. Pavlakos, and K. Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4501–4510, 2019.

[75] N. Kolotouros, G. Pavlakos, D. Jayaraman, and K. Daniilidis. Probabilistic modeling for human mesh recovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11605–11614, 2021.

[76] M. Körtgen, G.-J. Park, M. Novotni, and R. Klein. 3d shape matching with 3d shape contexts. In *The 7th central European seminar on computer graphics*, volume 3, pages 5–17. Budmerice, 2003.

[77] Z. Lahner, D. Cremers, and T. Tung. Deepwrinkles: Accurate and realistic clothing modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 667–684, 2018.

[78] Z. Lahner, D. Cremers, and T. Tung. Deepwrinkles: Accurate and realistic clothing modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 667–684, 2018.

[79] R. Lambert. Capsule nets for content based 3d model retrieval, 2018.

[80] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint*, 2017.

[81] B. Li, A. Godil, M. Aono, X. Bai, T. Furuya, L. Li, R. J. López-Sastre, H. Johan, R. Ohbuchi, C. Redondo-Cabrera, et al. Shrec'12 track: Generic 3d shape retrieval. In *Proceedings of Eurographics Workshop on 3D Object Retrieval (3DOR)*, pages 119–126, 2012.

[82] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. In *ACM Transactions on Graphics (TOG)*, volume 28, page 175. ACM, 2009.

[83] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas. Fpnn: Field probing neural networks for 3d data. *Advances in Neural Information Processing Systems*, 29:307–315, 2016.

[84] Z. Li, T. Yu, C. Pan, Z. Zheng, and Y. Liu. Robust 3D self-portraits in seconds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[85] J. Liang and M. C. Lin. Shape-aware human pose and shape reconstruction using multi-view images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4352–4362, 2019.

[86] K. Lin, L. Wang, and Z. Liu. Mesh graphormer. *arXiv preprint arXiv:2104.00272*, 2021.

[87] S. Liu, L. Giles, and A. Ororbia. Learning a hierarchical latent-variable model of 3d shapes. In *2018 International Conference on 3D Vision (3DV)*, pages 542–551. IEEE, 2018.

[88] S. Liu, S. Saito, W. Chen, and H. Li. Learning to infer implicit surfaces without 3D supervision. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2019.

[89] M. Loper, N. Mahmood, and M. J. Black. Mosh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics (TOG)*, 33(6):1–13, 2014.

[90] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015.

[91] J. Louw and A. Rix. Irradiance modelling for bi-facial pv modules using the ray tracing technique. In *2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)*, pages 383–388. IEEE, 2019.

[92] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[93] Q. Ma, J. Yang, S. Tang, and M. J. Black. The power of points for modeling humans in clothing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10974–10984, 2021.

[94] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015.

[95] J. Masci, E. Rodolà, D. Boscaini, M. M. Bronstein, and H. Li. Geometric deep learning. In *SIGGRAPH ASIA 2016 Courses*, page 1. ACM, 2016.

[96] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 922–928. IEEE, 2015.

[97] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 369–374, 2000.

[98] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

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

[100] M. Mortara, G. Patané, M. Spagnuolo, B. Falcidieno, and J. Rossignac. Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. *Algorithmica*, 38(1):227–248, 2004.

[101] A. Y. Mulayim, U. Yilmaz, and V. Atalay. Silhouette-based 3-D model reconstruction from multiple images. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(4):582–591, 2003.

[102] R. Natsume, S. Saito, Z. Huang, W. Chen, C. Ma, H. Li, and S. Morishima. SiCloPe: Silhouette-based clothed people. In *CVPR*, 2019.

[103] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[104] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *2018 international conference on 3D vision (3DV)*, pages 484–494. IEEE, 2018.

[105] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[106] S. I. Park and J. K. Hodgins. Capturing and animating skin deformation in human motion. *ACM Transactions on Graphics (TOG)*, 25(3):881–889, 2006.

[107] S. I. Park and J. K. Hodgins. Data-driven modeling of skin and muscle deformation. In *ACM SIGGRAPH 2008 papers*, pages 1–6. 2008.

[108] C. Patel, Z. Liao, and G. Pons-Moll. TailorNet: Predicting clothing in 3D as a function of human pose, shape and garment style. In *CVPR*, 2020.

[109] C. Patel, Z. Liao, and G. Pons-Moll. TailorNet: Predicting clothing in 3D as a function of human pose, shape and garment style. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

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

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

[112] F. Perbet, S. Johnson, M.-T. Pham, and B. Stenger. Human body shape estimation using a multi-resolution manifold forest. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 668–675, 2014.

[113] H. Pottmann, J. Wallner, Q.-X. Huang, and Y.-L. Yang. Integral invariants for robust geometry processing. *Computer Aided Geometric Design*, 26(1):37–60, 2009.

[114] E. Prados and O. Faugeras. Shape from shading. In *Handbook of mathematical models in computer vision*, pages 375–388. Springer, 2006.

[115] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CoRR*, abs/1612.00593, 2016.

[116] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *CoRR*, abs/1706.02413, 2017.

[117] J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut, and D. Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021.

[118] S. R. Richter and S. Roth. Matryoshka networks: Predicting 3D geometry via nested shape layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[119] L. G. Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[120] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017.

[121] R. M. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 225–233. Eurographics Association, 2007.

[122] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, 2019.

[123] S. Saito, T. Simon, J. Saragih, and H. Joo. PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3D human digitization. In *CVPR*, 2020.

[124] N. Sedaghat, M. Zolfaghari, and T. Brox. Orientation-boosted voxel nets for 3d object recognition. *CoRR*, abs/1604.03351, 2016.

[125] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006.

[126] J. Shade, S. Gortler, L.-W. He, and R. Szeliski. Layered depth images. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, 1998.

[127] A. Sharma, R. Horaud, J. Cech, and E. Boyer. Topologically-robust 3d shape matching based on diffusion geometry and seed growing. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2481–2488, 2011.

[128] B. Shi, S. Bai, Z. Zhou, and X. Bai. Deeppano: Deep panoramic representation for 3-d shape recognition. *IEEE Signal Processing Letters*, 22(12):2339–2343, 2015.

[129] S. Shimada, V. Golyanik, W. Xu, and C. Theobalt. Physcap: Physically plausible monocular 3d motion capture in real time. *ACM Transactions on Graphics (ToG)*, 39(6):1–16, 2020.

[130] D. Shin, Z. Ren, E. B. Sudderth, and C. C. Fowlkes. Multi-layer depth and epipolar feature transformers for 3D scene reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR-W)*, 2019.

[131] A. Sinha, J. Bai, and K. Ramani. Deep learning 3d shape surfaces using geometry images. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *European Conference on Computer Vision (ECCV)*, pages 223–240, 2016.

[132] A. Sinha, A. Unmesh, Q. Huang, and K. Ramani. Surfnet: Generating 3d shape surfaces using deep residual networks. *CoRR*, abs/1703.04079, 2017.

[133] D. Smeets, J. Hermans, D. Vandermeulen, and P. Suetens. Isometric deformation invariant 3d shape recognition. *Pattern Recognition*, 45(7):2817–2831, 2012.

[134] D. Smith, M. Loper, X. Hu, P. Mavroidis, and J. Romero. Facsimile: Fast and accurate scans from an image in less than a second. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5330–5339, 2019.

[135] J. Starck and A. Hilton. Surface capture for performance-based animation. *IEEE computer graphics and applications*, 27(3):21–31, 2007.

[136] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[137] F. Tan, H. Zhu, Z. Cui, S. Zhu, M. Pollefeys, and P. Tan. Self-supervised human depth estimation from monocular videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[138] G. Tiwari, B. L. Bhatnagar, T. Tung, and G. Pons-Moll. Sizer: A dataset and model for parsing 3d clothing and learning size sensitive 3d clothing. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 1–18. Springer, 2020.

[139] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan. Scanning 3d full human bodies using kinects. *IEEE transactions on visualization and computer graphics*, 18(4):643–650, 2012.

[140] R. Tucker and N. Snavely. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[141] S. Tulsiani, R. Tucker, and N. Snavely. Layer-structured 3D scene inference via view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[142] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Computer Vision and Pattern Recognition (CVPR)*, pages 209–217, 2017.

[143] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid. BodyNet: Volumetric inference of 3D human body shapes. In *ECCV*, 2018.

[144] G. Varol, D. Ceylan, B. C. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid. BodyNet: Volumetric inference of 3D human body shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[145] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from Synthetic Humans. In *CVPR*, 2017.

[146] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[147] A. VENKAT. *MONOCULAR 3D HUMAN BODY RECONSTRUCTION*. PhD thesis, International Institute of Information Technology Hyderabad, 2020.

[148] A. Venkat, S. S. Jinka, and A. Sharma. Deep textured 3D reconstruction of human bodies. In *BMVC*, 2018.

[149] A. Venkat, S. S. Jinka, and A. Sharma. Deep textured 3D reconstruction of human bodies. In *British Machine Vision Conference (BMVC)*, 2018.

[150] A. Venkat, C. Patel, Y. Agrawal, and A. Sharma. HumanMeshNet: Polygonal mesh recovery of humans. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV-W)*, 2019.

[151] A. Venkat, C. Patel, Y. Agrawal, and A. Sharma. HumanMeshNet: Polygonal mesh recovery of humans. In *Proceedings of the IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2019.

[152] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH*, 2008.

[153] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH 2008 papers*, pages 1–9. 2008.

[154] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2mesh: Generating 3D mesh models from single RGB images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[155] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (TOG)*, 36(4):72, 2017.

[156] M. Waschbüsch, S. Würmlin, D. Cotting, F. Sadlo, and M. Gross. Scalable 3d video of dynamic scenes. *The Visual Computer*, 21(8):629–638, 2005.

[157] O. Wiles and A. Zisserman. Silnet : Single- and multi-view reconstruction by learning from silhouettes. In *British Machine Vision Conference*, 2017.

[158] J. Wu, Y. Wang, T. Xue, X. Sun, B. Freeman, and J. Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. In *Advances in neural information processing systems*, pages 540–550, 2017.

[159] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

[160] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Neural Information Processing Systems (NIPS)*, pages 82–90, 2016.

[161] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[162] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015.

[163] K. Xie, T. Wang, U. Iqbal, Y. Guo, S. Fidler, and F. Shkurti. Physics-based human motion estimation and synthesis from videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11532–11541, 2021.

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

[165] Y. Xu, S.-C. Zhu, and T. Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7760–7770, 2019.

[166] S. M. Yamany and A. A. Farag. Free-form surface registration using surface signatures. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1098–1104. IEEE, 1999.

[167] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Neural Information Processing Systems (NIPS)*, pages 1696–1704, 2016.

[168] Y. Yang, C. Feng, Y. Shen, and D. Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 3, 2018.

[169] T. Yu, Z. Zheng, K. Guo, P. Liu, Q. Dai, and Y. Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5746–5756, 2021.

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

[171] Q. Zhang, B. Fu, M. Ye, and R. Yang. Quality dynamic human body modeling using a single low-cost depth camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 676–683, 2014.

[172] B. Zhao, X. Wu, Z. Cheng, H. Liu, and J. Feng. Multi-view image generation from a single-view. *CoRR*, abs/1704.04886, 2017.

[173] Z. Zheng, T. Yu, Y. Liu, and Q. Dai. Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[174] Z. Zheng, T. Yu, Y. Wei, Q. Dai, and Y. Liu. Deephuman: 3d human reconstruction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7739–7749, 2019.

[175] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. *CoRR*, abs/1605.03557, 2016.

[176] H. Zhu, X. Zuo, S. Wang, X. Cao, and R. Yang. Detailed human shape estimation from a single image by hierarchical mesh deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[177] H. Zhu, X. Zuo, S. Wang, X. Cao, and R. Yang. Detailed human shape estimation from a single image by hierarchical mesh deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4491–4500, 2019.

[178] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM transactions on graphics (TOG)*, 23(3):600–608, 2004.

[179] M. Zollhofer, M. Niessner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, and M. Stamminger. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Transaction on Graphics*, 33(4):156:1–156:12, 2014.