Development and Tracking of *Consensus Mesh* for Monocular Depth Sequences.

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

Master Of Science
in
Computer Science and Engineering By Research

by

Gaurav Mishra
201202057
gaurav.mishra@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2019

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Development and Tracking of *Consensus Mesh* for Monocular Depth Sequences.**" by **Gaurav Mishra**, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. P. J. Narayanan

To My Parents

Acknowledgements

I would like to thank all the members, especially the esteemed professors, at CVIT lab for fostering a rich and conducive environment that has not only made research possible but enjoyable. I thank, Prof. P.J. Narayanan and Dr. Kiran Varanasi for guiding me not just for my work but in shaping an all round outlook towards research. I thank them for their invaluable insights, discussions and guidance on issues ranging from trivial doubts to those that eventually made this work possible.

I am grateful to Saurabh Saini for always helping me with issues both within and outside the scope of this work and to be with me for the entire journey. I would also like to thank Parikshit Sakurikar and Rajvi Shah for the constant help and support that they provided with great patience.

I would also like to thank Mayank, Manas, Vikram, Vrushank, Sudarsh, Aabhas, Arjit, Ambika, Navya, Samrudhdhi and Shashikala for helping me create the dataset I needed for my work.

John Heywood once said: *Rome wasn't built in a day, but they were laying bricks every hour*. The proverb is universal and applies to many aspects of life. The take-away that one should keep in mind is that important work takes time. We first think of a bigger problem which is there in the world and try to solve it step by step, fitting one cog at a time. Research at times is not the most pleasant thing to be stuck with, and I thank all my friends and family members for making this journey possible.

"Life is like playing a violin in public and learning the instrument as one goes on."

– Samuel Butler

Abstract

Human body tracking typically requires specialized capture set-ups. Although pose tracking is available in consumer devices like Microsoft Kinect, it is restricted to stick figures visualizing body part detection. In this thesis, we propose a method for full 3D human body shape and motion capture of arbitrary movements from the depth channel of a single Kinect, when the subject wears casual clothes. We do not use the RGB channel or an initialization procedure that requires the subject to move around in front of the camera. This makes our method applicable for arbitrary clothing textures and lighting environments, with minimal subject intervention. Our method consists of 3D surface feature detection and articulated motion tracking, which is regularized by a statistical human body model [40]. We also propose the idea of a *Consensus Mesh (CMesh)* which is the 3D template of a person created from a single view point. We demonstrate tracking results on challenging poses and argue that using *CMesh* along with statistical body models can improve tracking accuracies. Quantitative evaluation of our dense body tracking shows that our method has very little drift which is improved by the usage of *CMesh*.

We explore the possibility of improving the quality of *CMesh* using RGB images in a post processing step. For this we propose a pipeline involving Generative Adversarial Networks. We show that *CMesh* can be improved from RGB images of the original person by learning corresponding relative normal maps (N^R_{map}). These N^R_{map} have the potential to encode the nuances in the *CMesh* with respect to ground truth object. We explore such method in a synthetic setting for static human like objects. We demonstrate quantitatively that details which are learned from such a pipeline are invariant to lighting and texture changes. In future the generated N^R_{map} can be used to improve the quality of *CMesh*.

Contents

Chapter Page

Abstract . . . vii

1 Introduction . . . 1
1.1 Motivation . . . 2
1.2 Area of Human Shape and Motion Capture . . . 3
1.3 Our Contributions . . . 5
1.4 Thesis Organization . . . 5

2 Related Work . . . 6
2.1 Human Body Models/Datasets . . . 7
2.2 Multi-view Systems . . . 8
2.3 Monocular Systems - Dense Surface Reconstruction . . . 8
2.4 Monocular Systems - Shape and Motion Capture . . . 9
2.5 Background . . . 10
2.5.1 The MPI-SMPL Model . . . 10
2.5.2 Non Rigid ICP . . . 11
2.5.3 Inverse Kinematics . . . 12

3 Dataset . . . 13
3.1 Dataset Acquisition . . . 13
3.2 Camera Calibration . . . 13

4 Tracking SMPL Body Model on Monocular Depth Sequences . . . 17
4.1 Pose refinement . . . 18
4.2 Tracking Framework . . . 18
4.2.1 Depth Map Triangulation ($\mathbf{F_t} \rightarrow \mathbf{M_t}$): . . . 19
4.2.2 Initialization ($\mathbf{S_0} \rightarrow \mathbf{S_1}$): . . . 19
4.2.3 Mesh alignment ($\mathbf{M_t} \rightarrow \mathbf{M'_t}$): . . . 20
4.2.4 SMPL registration ($\mathbf{S_t} \rightarrow \mathbf{S_{t+1}}$): . . . 20
4.3 Qualitative Results . . . 20
4.4 Summary . . . 21

5 Developing The *Consensus Mesh* . . . 22
5.1 Generating subject specific Consensus Mesh . . . 23
5.2 Importance of Consensus Mesh . . . 24
5.3 Experiments . . . 25

5.3.1 Qualitative Results . . . 25
5.3.2 Synthetic Dataset . . . 25
5.3.3 Quantitative Results . . . 26
5.4 Summary . . . 28

6 Towards Fine Registration of Consensus Mesh . . . 29
6.1 Modeling Nuances . . . 29
6.2 Problem Definition . . . 30
6.3 Dataset Generation . . . 30
6.3.1 Proxy Object . . . 31
6.3.2 Texture Maps . . . 31
6.3.3 Configurations . . . 32
6.3.4 Normal maps . . . 32
6.4 Network Architecture . . . 32
6.4.1 Network objective: . . . 33
6.5 Network Training . . . 33
6.6 Results . . . 34
6.7 Discussion . . . 36
6.8 Summary . . . 36

7 Conclusions and Future Work . . . 37

Related Publications . . . 38

Bibliography . . . 39

List of Figures

Figure Page

1.1 Left: Input to our system (only depth maps are used), Right: Output of our system, where first three meshes show SMPL [40] model tracked over time and last three meshes show Consensus meshes, reposed using SMPL. 2
1.2 A Data-Driven Approach for Real-Time Full Body Pose Reconstruction from a Depth Camera Baak et al. [8]. 3
1.3 Results of Microsoft's Fusion 4D system Dou et al. [28]. 4

2.1 Overview of Anguelov et al. [7]: Animation of a motion capture sequence taken for a subject. The muscle deformations are synthesized automatically from the space of pose and body shape deformations. 6
2.2 SMPL model. (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. Loper et al. [40]. 7
2.3 Top: A sequence of poses captured from eight video recordings of a capoeira turn kick [24]. Bottom: From left to right, (1) An example 3D texture scan. (2) Multi-part aligned mesh model (4) The body made fater and dressed in the same clothing. (5) This new body shape posed in a new, never seen, pose Zhang et al. [63]. 8
2.4 Real-time reconstructions of a moving scene with DynamicFusion; both the person and the camera are moving. The initially noisy and incomplete model is progressively denoised and completed over time (left to right) Newcombe et al. [43]. 9
2.5 Overview of [59]: (a) Four views of the body in different poses are captured from a single Kinect. (b) 3D point cloud and segmented 3D point cloud with ground plane for four frames (one shown). (c) Recovered pose and shape (4 frames). (d) Recovered shape in new pose. 9
2.6 From a monocular RGB-D sequence (background), a low-dimensional parametric model of body shape (left), detailed 3D shape (middle), and a high-resolution texture map (right) [11]. 10
2.7 Left: Various stages of Non-Rigid ICP with $\mathbf{M_{t+1}}$ (Brown). Right: Inverse Kinematics, small points depict end effectors e_i (Blue) and e_f (Red) section 4.2. 11

3.1 Equipments used in our work. Top row shows Kinect sensor and Canon EOS 70D DSLR. Bottom row shows room ready for capture and camera placement on tripods. . 14
3.2 Frames from our dataset captured by Kinect,showing all 8 subjects. 15

3.4 Calibration of Kinect RGB and IR sensor. Top row shows images captured by Kinect RGB and IR sensors respectively. 15
3.3 Calibration of Kinect RGB sensor and Canon camera. Top row shows images captured by Kinect and Canon respectively. 16
3.5 Figure showing differences in the detail of RGB images captured by Kinect (Left) and Canon (Right). Kinect images become blurry while capturing fast actions. 16

4.1 Block Diagram of our proposed method. First we process depth maps to generate triangulated meshes. We use these along with SMPL [40] model for manual initialization for first frame. After that we iterate between the *Mesh Alignment* and *SMPL Registration* stages for each frame. The final tracked, pose aligned and shape adjusted SMPL mesh S_{t+1} is generated as output. We also generate consensus mesh of the person from a 360° sequence which is then reposed using ARAP. 17
4.2 An example initialization. Top left: Depth image from Kinect. Top right: Markers placed on SMPL mesh and point cloud. Bottom left: IK solution. Bottom right: Fine tuned estimate. 19
4.3 Qualitative results of our algorithm. Left: Segmented point clouds, color coded as Blue (Near) and Red (Far). Right: Tracked SMPL model. 21

5.1 Consensus mesh results. Note how the generated meshes are more faithful to the true geometry compared to the SMPL mesh (Best viewed in color). 22
5.2 Left: RGB Image in starting pose. Middle: $\mathbf{M_1}$ generated using subsection 4.2.1. Right: Front surface approximation using Kinect Fusion ($\mathbf{P_f}$). Notice that $\mathbf{P_f}$ has more details (e.g. wrinkles) compared to $\mathbf{M_1}$. 23
5.3 Left to Right: $\mathbf{P_f}$, $\mathbf{P_b}$, filling of side profile views, $\mathbf{C_p}$ 24
5.4 Left: RGB image, Middle: SMPL model, Right: Generated *CMesh*. Notice that the details captured by *CMesh* are better as compared to SMPL. 25
5.5 Qualitative results of our framework on our dataset. Each figure shows the input point cloud (color coded), pose refined SMPL mesh and reposed CMesh. In inset figure we also show corresponding RGB image. Please refer to the supplementary video [4] to notice the various challenging poses, actions, hairstyles and clothing worn by the subjects (Best viewed in color). 26
5.6 Quantitative results on De Aguiar et al. [24] and Vlasic et al. [55]. Top row shows graphs for mean drift error (ϵ_t) for various frames in the sequence (S31 [24] and I_squat [55]). The color coded CMesh and SMPL mesh represents average (ϵ_t) per vertex over all frames. Bottom row shows percentile based Hausdorff distance (d_l) per frame (for various $l = 95\%$ (red)), 75% (blue), 50% (magenta)). In the graph S stands for SMPL (dotted line) and C stands for CMesh (solid lines) based errors. Color coded ground truth mesh based on nearest neighbor found with respect to SMPL and Cmesh is also shown. (Best viewed in colors on screen) 27

6.1 Training phase of our pipeline, Note that the input to the generator **(G)** is of dimension $256 \times 256 \times 6$ which is formed by stacking RGB image and corresponding Canny edges. 30
6.2 Dataset used in our experiments, Left: Top row shows texture images **T1, T2** and **T3** for Dwarf dataset and Bottom row has both models shown in **T1, T2, T3** respectively. Right: Input pair (RGB and N_{map}) for the network and corresponding canny edges. . . 31

6.3 Testing phase of our pipeline, Note here we only need RGB images as an input, 3D meshes are not required. 33
6.4 Qualitative result of our experiments, each row shows RGB image, N_{map}^{R} and GAN result from left to right. Notice how the network is able to generate convincing and similar results for different textures. 34
6.5 Results showing $\mathbf{T3's}$ invariance to lighting conditions. From left to right: RGB, N_{map}^{R}, output of networks learned on $\mathbf{T1}$, $\mathbf{T2}$ and $\mathbf{T3}$ respectively. Notice the consistency in $\mathbf{T3's}$ result. 35

List of Tables

Table Page

3.1 Clothing details 14
3.2 8 actions common to each person. 14

5.1 Percentage error reduction of repositioned CMesh with respect to SMPL tracking computed over (ϵ_t^i and d_l^i). Top two sequences are from [24] and bottom four from [55] 27

6.1 ϵ' for Dwarf Dataset. 35
6.2 ϵ' for Human Dataset. 35

Chapter 1

Introduction

The problem of *'Full Body Motion Transfer'* which refers to transferring action of one person to another has been a topic of interest for researchers in the past decade. The problem on its own is the result of the realization that people try to mimic actions of their favorite performers like Rita Hayworth, Michael Jackson etc. Novice performers always dream to dance like them. The problem aims at creating a system which can take a near similar video of some performance and correct it automatically. Aiming for such a scenario where a person uploads his/her dance video and all the mistakes in dance steps are corrected automatically is interesting but hard to solve. Anguelov et al. [7] address the problem of 'Deformation Transfer' which is limited to transferring deformations to a human model using input data. Thies et al. [52] address the complexity of the problem and show excellent results on a subproblem, which is facial expression transfer.

A simpler version of the problem that one can attempt to solve is to identify nuances in the performance of novice actor with respect to original actor. For 2D videos, one can think of image morphing solutions based on optical flows or correspondence matching to tackle the same. However dense correspondence finding algorithms for RGB images are not accurate at the pixel level when the poses are far apart. Even if we get accurate correspondences, the identified nuances will be limited to 2D domain, and may not be useful for a performer.

Going in 3D can help simplify the problem as 3D is how we perceive the world and understanding where one is lacking becomes easier. In 3D due to additional depth channel information available, many tasks like occlusion handling can be done with ease as compared to 2D videos. The problem of finding robust correspondences across different frames with varying poses is also difficult in 3D. Algorithms based on 3D surface attributes like Heat Kernel Signatures [16] do exist but they give good accuracies which are suitable for body part matching in general.

Figure 1.1: Left: Input to our system (only depth maps are used), Right: Output of our system, where first three meshes show SMPL [40] model tracked over time and last three meshes show Consensus meshes, reposed using SMPL.

1.1 Motivation

Going through the vast literature of the field we stumble upon interesting works like De Aguiar et al. [24] and Vlasic et al. [55] where a person is laser scanned to create a 3D template mesh, which can then be tracked using videos captured by multiple cameras. The approach in a way fits a digital model of the person on each and every frame of the video, which gives us the ability to view the person in 3D and look for where the nuances are. Since the template mesh is common, projected pixels are in correspondence across frames.

Such approaches have limitations of their own. The biggest one is that these solutions are not affordable to a common man as one has to have access to a laser scanner and a multi camera setup or special body markers to record the performance. 3D is definitely good but the solution in 3D should be affordable. Several RGB-D approaches are there which fit skeletons on monocular videos. But a skeletal output is not suitable for the problem as we want to visualize the performance in full 3D with skin level accuracy.

Techniques by Bogo et al. [11], which fit 3D human body models to monocular RGB-D sequences are interesting, as they fit a common statistical model on every frame, leaving the body parts of the person across frames in perfect correspondence. Such models are able to give correspondence across different body shapes and size as well. However works like Bogo et al. [11] require the videos to be captured with tight fitting clothes.

These various observations led us to define the following problem which we are attempting to address in this thesis. A proper technical definition of the problem could be:

How do we fit a statistical human body model on monocular RGB-D sequences, when the subject wears casual clothes. How do we enhance it by adding clothing details and bringing it closer to the ground truth (Figure 1.1).

The problem aligns with the area of Human shape and motion capture which has been a largely studied topic in the field of computer vision. Going in such a direction is helpful for our end goal which is to guide the performer on how to correct themselves. Such a guidance system will be hard to build since videos of any actor, artist that are available to the general public are in the form of RGB videos. Statistical body model fitting help us tackle this problem to a certain level by inferring the models directly from RGB images [12]. Now we discuss in brief various works that are related to the area and how our work contributes in the field.

1.2 Area of Human Shape and Motion Capture

Recently, computer vision systems achieved 2D and 3D human body tracking from a simple capture setup e.g. convolutional neural network (CNN) models can detect body parts in RGB images [58]. Baak et al. [8] use a generative-discriminative hybrid framework, in which they combine inferences from skinned kinematic chain model and retrieved pose from a curated dataset to decide the final pose in each frame as shown in Figure 1.2. Their solution is data driven and limited by the number of poses in the dataset. However, these methods are not yet applicable for full body shape and motion visualization. Many applications in today's scenario like bio-mechanical analysis, medical rehabilitation etc. require motion tracking that is broadly accurate not only with respect to the position of the bone joints, but also on the surface of the skin.

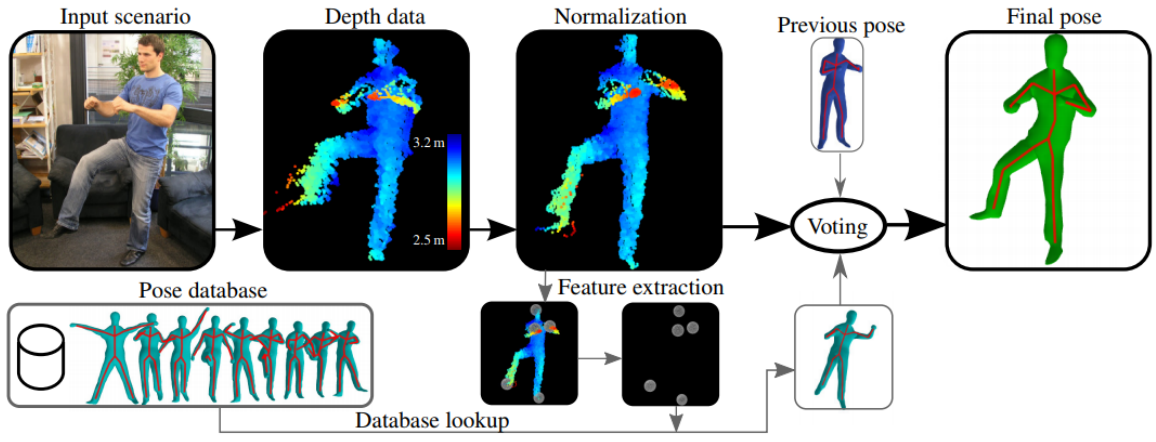

Figure 1.2: A Data-Driven Approach for Real-Time Full Body Pose Reconstruction from a Depth Camera Baak et al. [8].

Tracking results can be improved if we do not limit ourselves to monocular videos. Systems like [24, 63, 27], give results of excellent quality and are necessary for the progress of research in monocular field as well. Algorithms that take advantage of multi-camera setup often produce results which are of

good quality (Figure 1.3) as the data itself is of high quality. In this thesis we try to explore what is possible when the data is monocular.

Recently Bogo et al. [11] showed results on monocular RGB-D Kinect sequences of freely moving subjects to construct a detailed 3D reconstruction by doing coarse-to-fine processing. Such methods work with subject wearing tight fitting clothes. Models like Loper et al. [40] and Anguelov et al. [7] have been created from a large pool of real-world 3D scans of people to address the problem of human shape and motion capture. These models are accurate but they can only be fitted to a person wearing tight clothes. It is challenging to track people in everyday clothing.

To this end we propose a novel pipeline for tracking people in everyday clothing. Unlike prior methods, we take only the depth channel as input, for the sake of simplicity and independence to illumination conditions and clothing texture. An important distinctive element of our method is that we do not rely on 3D body part detection e.g. using the Microsoft Kinect API or a deep neural-network model trained for this purpose. This makes our method a useful baseline method, which can be improved by such features or from alternative information channels such as RGB data.

Figure 1.3: Results of Microsoft's Fusion 4D system Dou et al. [28].

In addition to above we also propose a novel *Consensus Mesh (CMesh)* generation pipeline which refers to creating a 3D template mesh of a subject from 360° sequences captured using a single Kinect. Such a mesh will capture the topology of the subject in consideration with clothing details. Along with qualitative evaluation, we show quantitative evaluation by treating output sequences of De Aguiar et al. [24] and Vlasic et al. [55] as our ground truth. We show that we achieve comparable tracking from just a monocular depth input, with very little drift. We have also shown that using the concept of a *CMesh* we can greatly improve the tracking accuracy with respect to naked body models based tracking, simply because of the fact that *CMesh* has a better adherence to topology of the person.

We explore the idea of making this *CMesh* of high quality by bringing it closer to the ground truth using RGB images of the person. We have tested the idea using relative normal maps for synthetic data. The approach shows immense potential as it is invariant to lighting and texture conditions and captures 3D properties well. Whole work of this thesis provides us with an essential subsystem for solving the bigger problem which aims at creating a low cost, easily deployable guidance system for emerging performers.

1.3 Our Contributions

This thesis comprises of several novel contributions :

1. We propose a novel tracking pipeline which consists of 3D HKS (heat-kernel signature) driven non-rigid ICP, articulated skeleton tracking and regularization by a statistical human body model, for fitting a 3D mesh template to a point cloud sequence.

2. We propose a body pose refinement step that uses statistical human body model for correspondence computation, and produces smooth trajectories over time.

3. We believe that we are the first to do temporal tracking on monocular depth input for subjects wearing casual clothes, using a statistical body model. We show tracking results for a variety of clothing styles and challenging poses.

4. We contribute a dataset of 88 sequences involving various clothing types. It consists of 8 people performing 11 actions ranging from simple to complex motions. Such dataset will help the community to progress further in this area.

5. We propose a pipeline to create a *Consensus mesh (CMesh)* using a single Kinect. Repositioning of the *CMesh* is shown to reduce quantitative drift and geometric errors in clothed models.

6. We show that generative models like Generative Adversarial Networks have the potential to further improve the *CMesh* by learning relative normal maps N^R_{map}. Such a technique is invariant to lighting and texture conditions to a great extent.

1.4 Thesis Organization

This thesis is organized as follows: chapter 2 discusses the works that are related to the problem and give some technical background on important algorithms we use. Our major contributions start from chapter 3 which talks about the captured dataset. chapter 4 focuses on tracking SMPL body model on monocular depth input. chapter 5 shows how to improve upon naked SMPL model by adding clothing details and chapter 6 gives us insight on how Generative Adversarial Networks can help us in improving the model further. We conclude this thesis in chapter 7 mentioning future scope of this work.

Chapter 2

Related Work

As our goal is towards building a framework enabling a low-cost, easily-deployable pose tracking system without any restriction on clothing, we will focus only on marker-less methods. We also want to estimate unrestricted body motion, so we exclude methods that recognize specific movements or track specific motion cycles from our review. Based on the complexity of data acquisition process, we can split human body motion capture methods into two groups. The first group consists of methods which require complex 3D data capturing setup such as laser scanners, inertial sensors or several multi-view, stereo pairs, depth cameras etc. [24, 29, 51, 63, 56, 27]. The second group consists of methods which use only a monocular RGB camera or a single depth sensor [59, 8, 23, 47, 37, 11]. In this case, as the data is limited and from one perspective only, most of the methods restrict themselves by defining system specific input constraints and priors. The output of this latter group is generally inferior compared to the former but the cost of the system, ease of capturing and setup, makes them applicable for the general public. An important factor in enabling the success of monocular methods is the availability of statistical human body models and datasets. So we first review them.

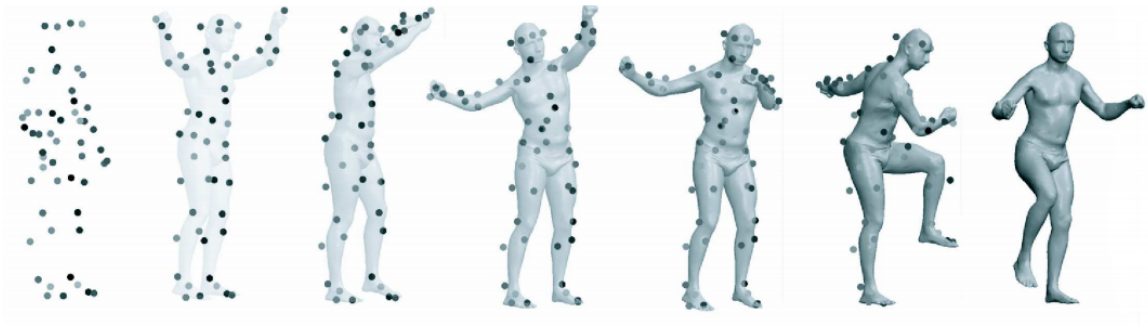

Figure 2.1: Overview of Anguelov et al. [7]: Animation of a motion capture sequence taken for a subject. The muscle deformations are synthesized automatically from the space of pose and body shape deformations.

2.1 Human Body Models/Datasets

Several methods used for human body shape modeling, pose estimation and motion tracking use learned parametric human body models for regularization. The data for learning such models is obtained from 3D scans of humans in varying body shapes and poses, that are captured with a high quality multi-camera set-up and registered with each other to a common mesh template. These parametric models can generate new plausible body shapes and poses by interpolating between the data. The SCAPE model was introduced by Anguelov et al. [7] and was learned using several registered scans. It required shape and pose transformation to be applied separately on mesh triangles which sometimes lead to inaccuracies near joints (Figure 2.1).

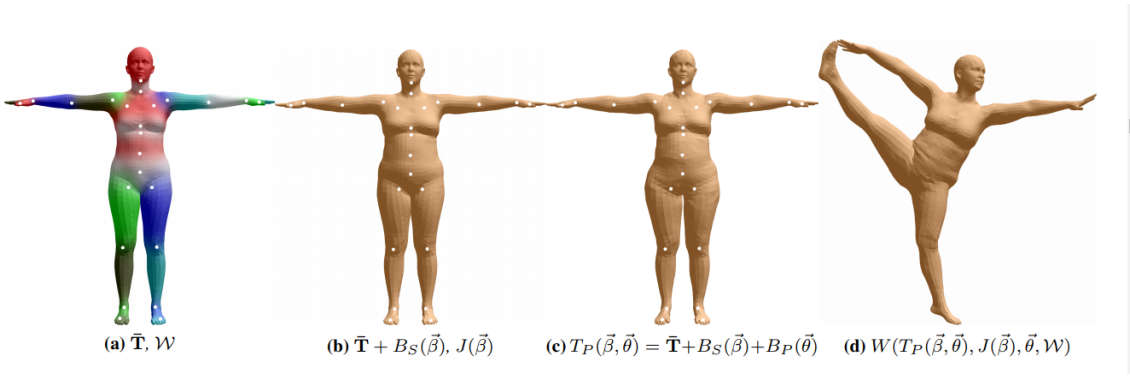

Figure 2.2: SMPL model. (a) Template mesh with blend weights indicated by color and joints shown in white. (b) With identity-driven blendshape contribution only (c) With the addition of of pose blend shapes in preparation for the split pose; note the expansion of the hips. (d) Deformed vertices reposed by dual quaternion skinning for the split pose. Loper et al. [40].

BlendSCAPE by Hirshberg et al. [30] addressed this issue by approximating triangle rotations as a linear combination of parts' rotations weighted using *blend weights*. Unlike the previous two models, Skinned Multi Person Linear (SMPL) [40] is a *vertex* based model of body shape and pose-dependent shape variations, which uses joint locations of kinematic chain of body parts. Although other complex models which can capture dynamic soft-tissue deformations and give textured outputs also exist (e.g. [45, 22, 11]), we choose to use publicly available SMPL for our framework (Figure 2.2) as our main goal is not highly accurate shape reconstruction but motion tracking in complex clothing feasible for a common user. Several datasets associated with the models mentioned above exists for learning and benchmarking. Most of these are static [7, 14, 15, 10] but recently a few large dynamic datasets have also been introduced with captured motion [45, 13]. Although the recent datasets like [45, 10, 13] are better at emulating challenges of the real-world than synthetic datasets like [14, 15], they can not yet be used to represent the output from an inexpensive commodity depth sensor like Kinect which has relatively high noise and low resolution.

2.2 Multi-view Systems

Some earlier methods rely on static contours or silhouettes for estimating the topology of the shape but they either assume multi-view acquisition [54, 24] or process binary silhouettes as inputs [38, 26, 48]. Methods like [24, 29, 41, 61, 63, 27] produce good results as shown in Figure 2.3, but they depend on specialized data acquisition stage and are inaccessible for a common user. Tong et al. [53] generate good quality 3D meshes but use three kinects, and require the person to stand on a rotating turntable while holding the pose. We differ from all of them as our system requires simple set-up of just one Kinect.

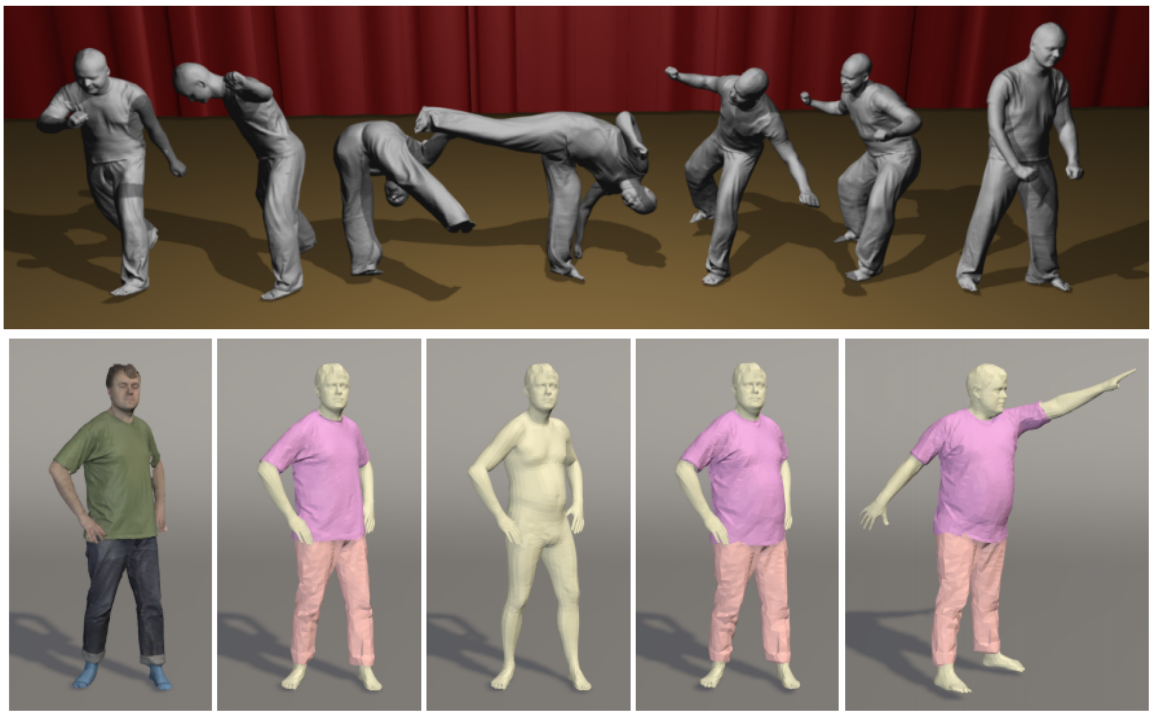

Figure 2.3: Top: A sequence of poses captured from eight video recordings of a capoeira turn kick [24]. Bottom: From left to right, (1) An example 3D texture scan. (2) Multi-part aligned mesh model (4) The body made fater and dressed in the same clothing. (5) This new body shape posed in a new, never seen, pose Zhang et al. [63].

2.3 Monocular Systems - Dense Surface Reconstruction

Monocular pose estimation, owing to its under-constrained nature, is generally solved using strong priors and multi-stage optimization frameworks. A class of such methods aim specifically at building detailed user model assuming *static* or little motion in the input sequence [33, 41]. In [37], authors present a method for building watertight models of static scenes using only a single Kinect aimed for 3D printing, which requires the subject to rotate around while roughly holding the pose. Recent methods

are able to achieve dynamic 3D surface reconstruction, also in real-time for virtual reality and teleconferencing applications [66]. These methods typically require the surface topology to be preserved during motion, such that shape regularization can be applied. Newcombe et al. [43] is able to reason about the canonical shape topology (Figure 2.4) while reconstructing dense motion. Though their method result in a high fidelity surface reconstruction of arbitrary shapes, they show results on slow moving subjects and are not concerned with pose of the person or connecting them with a human body shape. 4D surface reconstruction of complex real world clothing with fast motion remains a challenging problem.

Figure 2.4: Real-time reconstructions of a moving scene with DynamicFusion; both the person and the camera are moving. The initially noisy and incomplete model is progressively denoised and completed over time (left to right) Newcombe et al. [43].

2.4 Monocular Systems - Shape and Motion Capture

If we do not require full surface reconstruction, but only human body shape estimation, certain additional assumptions can be placed. Weiss et al. [59] use a single Kinect and SCAPE body model [7] to recover human shape and pose in different configurations (Figure 2.5). They show results on minimally clothed people and do not attempt tracking.

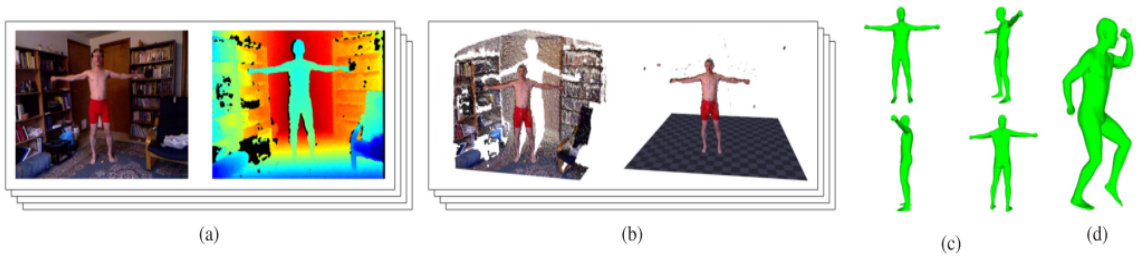

Figure 2.5: Overview of [59]: (a) Four views of the body in different poses are captured from a single Kinect. (b) 3D point cloud and segmented 3D point cloud with ground plane for four frames (one shown). (c) Recovered pose and shape (4 frames). (d) Recovered shape in new pose.

In [60] authors focus on fitting a minimally clothed shape (MCS) under complex clothing using motion cues. Cui et al. [23] build a full 3D human model using a single Kinect for scanning but require the user to maintain a specific pose. In [47] authors focus on virtual *avatar* creation of a person in any pose using four static images from a commodity depth sensor but they do not explicitly model human

motion. Efficient human body shape estimation and tracking is demonstrated by [64, 62] on minimally clothed subjects.

Figure 2.6: From a monocular RGB-D sequence (background), a low-dimensional parametric model of body shape (left), detailed 3D shape (middle), and a high-resolution texture map (right) [11].

Bogo et al. [11] is most similar to ours in motion as they use *dynamic* monocular RGB-D Kinect sequences of a freely moving subject to construct a detailed 3D reconstruction, as shown in Figure 2.6. They do coarse-to-fine processing involving several optimization stages and introduce a new multi-resolution body model called Delta which is based on SCAPE [7]. Although their results have fine details, they do not tackle clothed subjects. Our method differs from theirs in that we do not use the color channel, and rely entirely on non-rigid surface matching on point clouds which is regularized by the statistical human body model. We show good model fitting irrespective of clothing type, additionally we have shown that tracking accuracy can be improved by using a subject specific *consensus mesh* along with statistical human body model.

2.5 Background

In this section we discuss certain models and algorithms which are essential to understand and develop our tracking pipeline. We have modified some of these algorithms according to our need but the underlying structure of the algorithm remain the same.

2.5.1 The MPI-SMPL Model

Skinned Multi-Person Linear or SMPL [40] is a statistical human body model learned over a large 4D dataset. We use the standard SMPL model for regularizing shape in each frame. The model consists of 6890 vertices and its underlying skeleton contains 24 joints. The parameters for this model are 24×3 joint angles ($\theta_\tau, \tau \in \{1, 2, \ldots 72\}$); 10 shape parameters ($\eta \in \{1, 2, \ldots 10\}$) and additional 3 translation parameter of the root ($\Lambda \in \{1, 2, 3\}$). We concatenate all the parameters to build a 85 dimensional vector Θ. Using these we can generate SMPL mesh $S(\Theta)$ at any pose for a particular shape. The 3D mesh which is generated by the model takes into account shape and pose dependent body deformations.

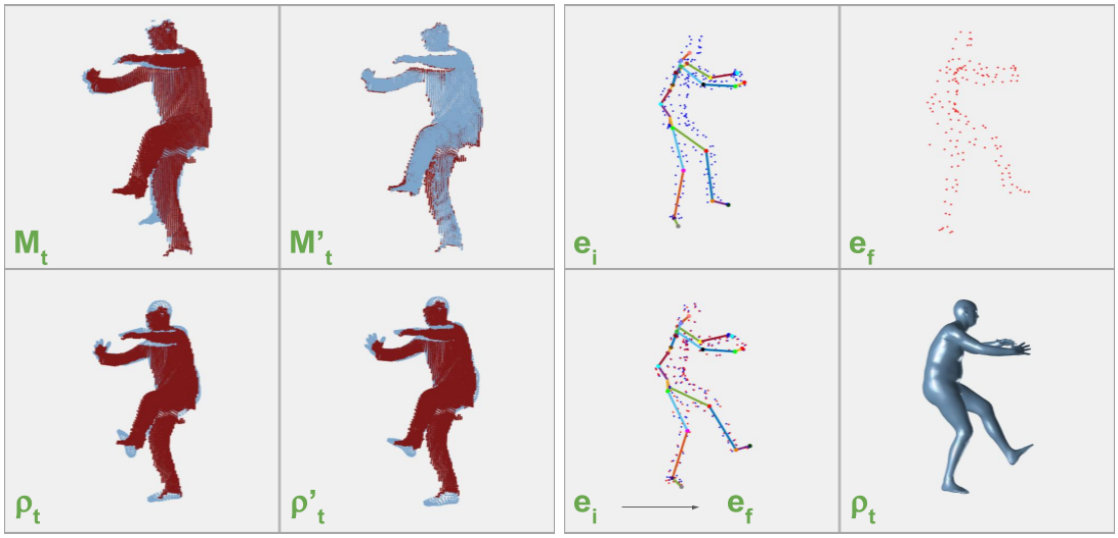

Figure 2.7: Left: Various stages of Non-Rigid ICP with $\mathbf{M_{t+1}}$ (Brown). Right: Inverse Kinematics, small points depict end effectors e_i (Blue) and e_f (Red) section 4.2.

2.5.2 Non Rigid ICP

We adapt the registration algorithm by Amberg et al. [6] for non-rigid alignment in our framework by estimating landmarks using Scale Invariant Heat Kernel Signatures (SIHKS [16]). We register meshes as shown in Figure 2.7, by computing the full mesh transformation matrix $X_{4n\times 3}$ (where n = number of vertices in input mesh). X is formed by vertically concatenating per vertex 4×3 transformation matrices and is computed by minimizing the following equation:

$$E(X) = E_d(X) + \alpha E_s(X) + \beta E_l(X) \tag{2.1}$$

Here E_d, E_s, E_l stand for the distance, stiffness and landmark energies respectively, regulated by α, β weight parameters. Both E_d and E_l are of the form $E(X) = \sum_i w_i \, ||X_i u_i - v_i||^2$, where (u_i, v_i) represent initial correspondence pair. For E_d, v_i is computed using nearest neighbor for all vertices of input mesh. For E_l, (u_i, v_i) represents sparse SIHKS correspondences.

Here w_i is the indicator function, which is 0 for invalid correspondences. We use two constraints to check for the validity of u_i, v_i pairs. First constraint is that the angle between normals at u_i and v_i should be less than 45° and second constraint states that u_i must be visible to the camera. We decrease the contribution of the landmark energy term E_l by varying β from 1 to 0 as the algorithm proceeds. This is to capture the increasing confidence of ICP based correspondences compared to SIHKS, over the course of iterations.

Similar to Amberg et al. [6] we define E_s based on differences between transformation matrices assigned to neighboring vertices. To this end we build a node-arc incident matrix (Dekker [25]) by converting the input mesh into a directed graph and following the same process as in Amberg et al. [6]. Just like β we gradually decrease the stiffness factor α from 10 to 0.1 over the course of non-rigid ICP

iterations. This models motion over various stiffness scales and hence capture the overall body part movement better.

2.5.3 Inverse Kinematics

Inverse Kinematics (IK) generates angular updates ($\Delta\theta$) of a kinematic chain given the parameterized initial (e_i) and the final position (e_f) of the end-effectors. Mathematically this can be written in matrix form involving Jacobian, $J(\theta)$ of joint angles [34]:

$$\Delta\theta = J'(JJ' + \lambda^2 I)^{-1}\bar{e}$$

Here $\bar{e} = \bar{e}_f - \bar{e}_i$ and λ is the damping constant. We use *Pseudo-Inverse Damped Least Squares* (PI-DLS [17]) for solving this as it provides well behaved solutions near singularities.

It is known that for an IK problem there are multiple solutions possible when the number of end effectors are sparse e.g. a few sensors attached to the free end of a robotic arm. In our case there are multiple such 'end-effectors' which are a few set of points attached to every bone of SMPL's skeleton Figure 2.7 ($\sim$ 7-8 markers per bone). This introduces additional constraints in an otherwise under constrained system thereby restricting the search space of incorrect and trivial solutions. We run this algorithm for 50 iterations, ignoring a solution whenever upper (θ_τ^{max}) or lower (θ_τ^{min}) angular bounds of a joint (τ) are breached. These bounds are chosen for each joint in order to restrict the solutions to naturally feasible joint angles. e.g. sideways head rotation (along the vertical Y axis) $\theta_\tau^{min} = -90^\circ$ and $\theta_\tau^{max} = 90^\circ$.

Chapter 3

Dataset

For performance evaluation of our pipeline which fits SMPL on monocular depth sequences of clothed subjects, we decided to capture our own dataset using a calibrated setup of one Kinect V2 sensor and one Canon EOS 70D DSLR. We use only the depth frames returned by Kinect for all our work in this thesis, but capturing RGB images was necessary for the dataset to be more useful. The actions we capture involve fast movements and due to Kinect limitations RGB images captured are sometimes blurry. We use Canon to overcome this and capture high quality RGB images side by side. There are datasets which are publicly available but we do not use them because most datasets are captured by Kinect V1 which has a lower depth resolution compared to the recent Kinect V2 and majority of them [18] are for object recognition [9], human detection [50], etc. Our dataset has good depth and RGB estimates on a variety of clothing styles, even for fast actions and hence is more suitable for our purpose.

3.1 Dataset Acquisition

We use a normal room as shown in Figure 3.1 with no special background for recording purpose. In order to have an evenly distributed source of light we use two Harrison Digipro 300 lamps. We kept depth resolution of Kinect at 512×424 and both RGB images were captured at 1920×1080 respectively. Note that for SMPL tracking and CMesh generation we have not used any RGB images, they are used only for visualization and will be used for fine registration in future. For present work we have captured actions for 4 male and 4 female subjects. They wore challenging everyday clothes like Hoodie, Jeans, T-shirt, Loose top , different hairstyles (Figure 3.2) etc. We record 11 sequences for each subject (7 common and 4 different actions) details of which are shown in Table 3.2.

3.2 Camera Calibration

We use standard checkerboard pattern and 'stereo camera calibrator' application of Matlab for calibrating the cameras. Application uses Zhang [65] inherently for calibration, which is a widely used algorithm giving sub-pixel errors for calibration. Note that since in Kinect V2 the depth camera coin-

Figure 3.1: Equipments used in our work. Top row shows Kinect sensor and Canon EOS 70D DSLR. Bottom row shows room ready for capture and camera placement on tripods.

Id	Gender	Clothing style
P1	Male	Tshirt, Jeans
P2	Female	Loose Top, Jeans, Hoodie
P3	Female	Full sweater, Jeans
P4	Female	Loose Top, Jeans
P5	Male	Tshirt, Jeans, Hoodie
P6	Male	Tshirt, Jeans
P7	Female	Kurta, Salwaar
P8	Male	Kurta, Jeans

Table 3.1: Clothing details

Id	Sequence Information
S1	360 rotation in T pose
S2	Bending on all four sides
S3	Standing toe touches
S4	Bending and cross toe touches
S5	Trikonasana (Yoga pose)
S6	Bowling
S7	Karate style front kick
S8	Waving hands and feets

Table 3.2: 8 actions common to each person.

cides with the IR camera we use IR images for calibration (Figure 3.3). Though RGB cameras are kept far apart (Figure 3.1) the calibration strategy gave us good results (Figure 3.4).

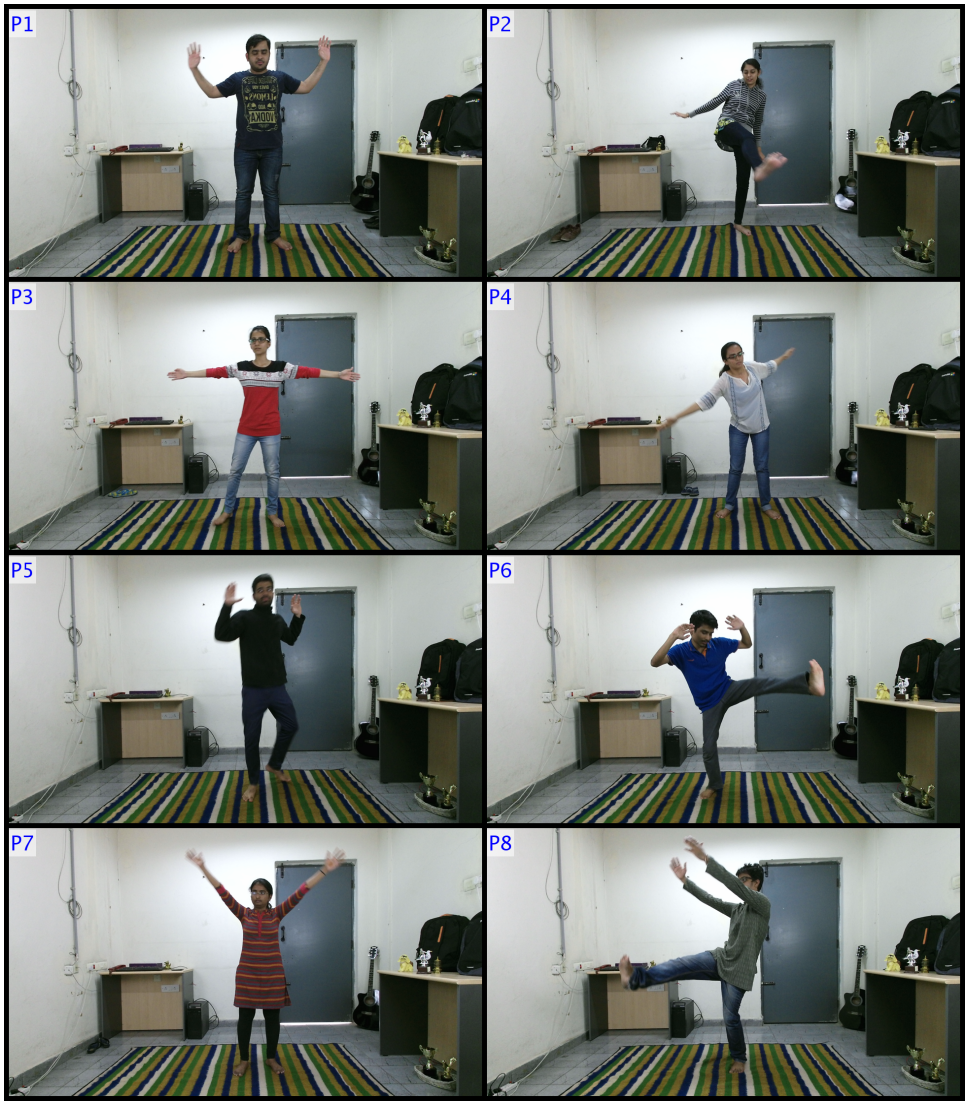

Figure 3.2: Frames from our dataset captured by Kinect,showing all 8 subjects.

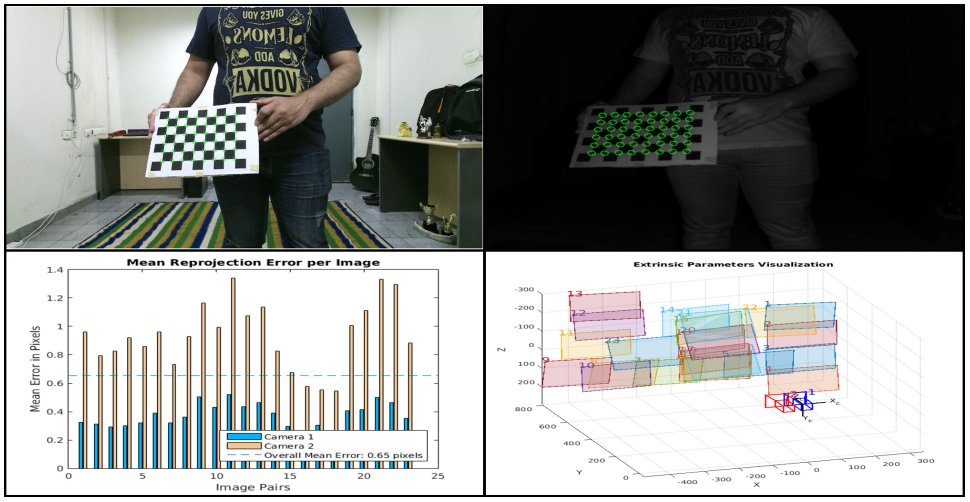

Figure 3.4: Calibration of Kinect RGB and IR sensor. Top row shows images captured by Kinect RGB and IR sensors respectively.

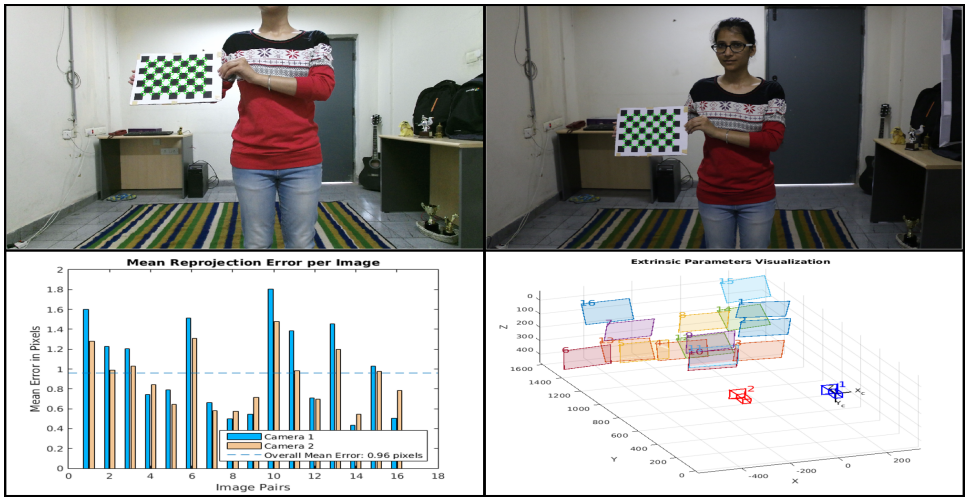

Figure 3.3: Calibration of Kinect RGB sensor and Canon camera. Top row shows images captured by Kinect and Canon respectively.

Figure 3.5: Figure showing differences in the detail of RGB images captured by Kinect (Left) and Canon (Right). Kinect images become blurry while capturing fast actions.

We use 'Canon Camera Connect' application available on Google Play-Store [1], to trigger the camera in a wireless manner. While capturing we set Exposure time as $1 \,/\, 160$ seconds, Aperture as $f \,/\, 3.5$ and ISO speed as 3200. These settings enable us to capture good images when the person is performing a fast action (Figure 3.5), which will be useful in future for doing fine registration of current *CMesh* results.

Chapter 4

Tracking SMPL Body Model on Monocular Depth Sequences

This chapter focuses on developing a tracking algorithm for SMPL on monocular depth sequences. The algorithm consists of various subsystems like 3D surface feature (SIHKS) driven Non-rigid ICP, Inverse Kinematics (section 2.5) and pose refinement based on local energy minimization. Pose refinement is a very important part of our pipeline. Since it is formulated as a local energy minimization problem it expects a good initialization. All other subsystems help us achieve a good initialization for the same. Block diagram of our complete pipeline is shown in Figure 4.1, please refer to the same whenever necessary.

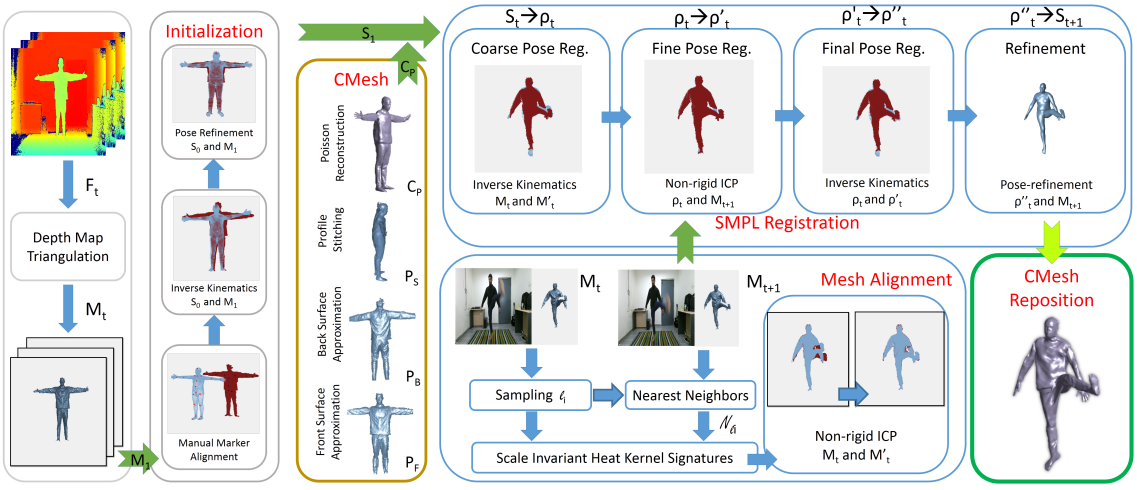

Figure 4.1: Block Diagram of our proposed method. First we process depth maps to generate triangulated meshes. We use these along with SMPL [40] model for manual initialization for first frame. After that we iterate between the *Mesh Alignment* and *SMPL Registration* stages for each frame. The final tracked, pose aligned and shape adjusted SMPL mesh S_{t+1} is generated as output. We also generate consensus mesh of the person from a 360° sequence which is then reposed using ARAP.

4.1 Pose refinement

This is a crucial and novel part of our proposed framework. It enables us to fine-tune a coarse SMPL estimate by minimizing a local energy term (E) defined as :

$$E = \alpha E_1 + \beta E_2 + \gamma E_3 \tag{4.1}$$

The intuition behind the various energy terms is explained below. E_1 penalizes the difference in the visible SMPL mesh vertices and the target point cloud. Let V_i be vertices of point cloud and V_f be its nearest neighbor in visible subset of SMPL mesh, then : $E_1 = \sum_i ||V_f - V_i||^2$.

Using E_1 alone can lead to unnatural human poses. We resolve this issue by defining E_2 using θ^{min} and θ^{max} for each θ in SMPL model s.t. $E_2 = \sum_j ||\theta_j - f(\theta_j)||^2$ where

$$f(\theta) = \begin{cases} \theta^{min}, & \theta < \theta^{min} \\ \theta, & \theta^{min} \leq \theta \leq \theta^{max} \\ \theta^{max}, & \theta > \theta^{max} \end{cases}$$

For temporal smoothing we add E_3 which restricts the current solution θ_{k_t} to be in close vicinity to the solution $\theta_{k_{t-1}}$ from the previous frame. $E_3 = \sum_k \left|\left|\theta_{k_t} - \theta_{k_{t-1}}\right|\right|^2$. It also helps in penalizing abrupt movements and limits jerky perturbations in the results. Hence the final energy can be defined by rewriting Equation 4.1 as a sum of L_2 terms :

$$E(\Theta) = \alpha \sum_i ||V_f - V_i||^2 + \beta \sum_j ||\theta_j - f(\theta_j)||^2 + \gamma \sum_k \left|\left|\theta_{k_t} - \theta_{k_{t-1}}\right|\right|^2 \tag{4.2}$$

For minimizing E we use quasi newton gradient descent algorithm (BFGS). During implementation we have used auto differentiation toolbox [39] for computing gradients.

4.2 Tracking Framework

Here we discuss step-by-step details for our iterative coarse-to-fine tracking framework as shown in Figure 4.1. Note again that we are only using segmented depth maps from Kinect as input. Human segmentation on depth stream was done using Kinect SDK 2.0. An important point to note over here is that the segmentation relies solely on depth maps. These depth maps are generated using infrared lights emitted from Kinect sensor. Due to this fact Kinect can capture depth maps in complete darkness.

Before proceeding further, we define some common mathematical notations: Subscript t refers to a time instance in the sequence from 1 to number of frames in the sequence. Each depth map is denoted by F_t. The corresponding triangulated mesh is denoted as $\mathbf{M_t}$ and the SMPL mesh as $\mathbf{S_t}$. The neighbors of a vertex v_i in 3D space are denoted by $\mathcal{N}_{v_i}$ which are estimated using approximate nearest neighbor algorithm.

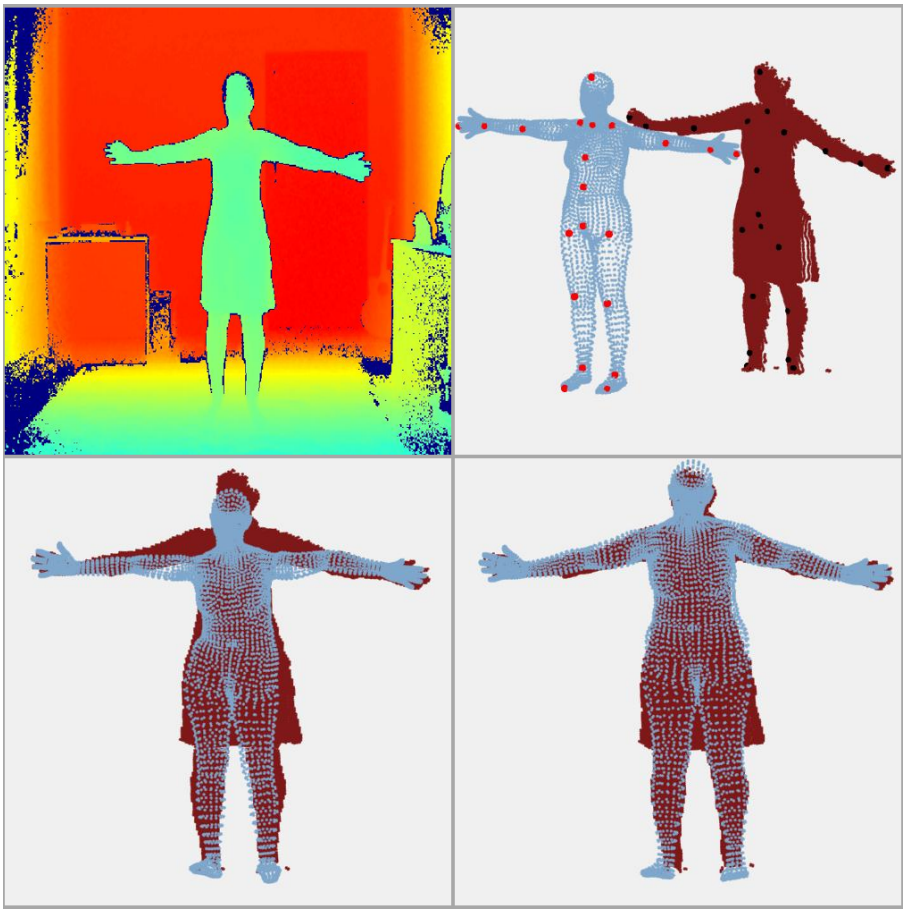

Figure 4.2: An example initialization. Top left: Depth image from Kinect. Top right: Markers placed on SMPL mesh and point cloud. Bottom left: IK solution. Bottom right: Fine tuned estimate.

4.2.1 Depth Map Triangulation ($\mathbf{F_t} \rightarrow \mathbf{M_t}$):

Given segmented depth maps, we convert them into triangulated meshes. For this we iterate row wise over the depth maps and connect a pixel to its right and bottom neighbors if the edge length between them in point cloud space is less then a certain threshold (< 5cm). This generates a good initialization mesh suitable for our purpose.

4.2.2 Initialization ($\mathbf{S_0} \rightarrow \mathbf{S_1}$):

For initialization (refer Figure 4.2) we manually associate 24 markers on the default SMPL mesh S_0 with the corresponding points on $\mathbf{M_1}$. We apply IK to give us a coarse alignment between S_0 and $\mathbf{M_1}$ (subsection 2.5.3). To further refine the initialization we fine-tune this coarse alignment using Equation 4.2 which yields the final MCS denoted by S_1. This is the only manual step in our entire framework and needs to be performed only once for a sequence. Although there are no strict restriction regarding the starting pose of the subject but we use a common '**T**' pose for our experiments. For a

random shape and pose, automatic initialization is a hard problem but can be solved to a certain extent using techniques mentioned in Bogo et al. [11]. However such an initialization is not the focus of our current work.

4.2.3 Mesh alignment ($\mathbf{M_t} \rightarrow \mathbf{M'_t}$):

For mesh alignment between consecutive frames we use non-rigid ICP (subsection 2.5.2). We define landmarks as randomly sampled vertices near each joint in $\mathbf{M_t}$. We use SIHKS as mesh features which are quite robust but cannot differentiate between symmetric body parts. Furthermore in our case topological difference between $\mathbf{M_t}$ and $\mathbf{M_{t+1}}$ (which might arise due to unrestricted motion and loose clothing) also aggravate the problem. In order to deal with this, we match the landmarks $l_i \subset \mathbf{M_t}$ within a small neighborhood $\mathcal{N}_{l_i} \subset \mathcal{M}_{t+1}$, which yields less noisy correspondences. We denote the resultant aligned mesh as $\mathbf{M'_t}$.

4.2.4 SMPL registration ($\mathbf{S_t} \rightarrow \mathbf{S_{t+1}}$):

We register the SMPL mesh $\mathbf{S_t}$ with the point cloud mesh $\mathbf{M_{t+1}}$ using a coarse-fine alignment strategy described below :

1. **Coarse pose registration ($\mathbf{S_t} \rightarrow \rho_\mathbf{t}$):** We apply IK on $\mathbf{S_t}$ using initial end-effectors from $\mathbf{M_t}$ and final from $\mathbf{M'_t}$ using sampling strategy defined in (subsection 2.5.3).

2. **Fine pose registration ($\rho_\mathbf{t} \rightarrow \rho'_\mathbf{t}$):** As the meshes $\rho_\mathbf{t}$ and $\mathbf{M_{t+1}}$ are relatively close, we use non-rigid ICP without landmarks for fine-tuning the alignment between these two meshes.

3. **Final pose registration ($\rho_\mathbf{t} \rightarrow \rho''_\mathbf{t}$):** The result from the last step is pose correct but shape deformed. We perform IK again by choosing initial end effectors from $\rho_\mathbf{t}$ and final from $\rho'_\mathbf{t}$ which gives $\rho''_\mathbf{t}$ which is a refined estimate.

4. **Shape and pose refinement ($\rho''_\mathbf{t} \rightarrow \mathbf{S_{t+1}}$):** Finally we apply pose-refinement Equation 4.2 for fine-tuning the alignment of $\rho''_\mathbf{t}$ to give the pose and shape corrected $\mathbf{S_{t+1}}$.

We perform our automatic mesh alignment and SMPL registration steps for all pairs of consecutive frames. We require approximately 3-4 minutes per frame on a 3^{rd} generation Intel processor with 8 GB memory. As we are not aiming for a real time scenario, our framework is currently implemented in Matlab and C++ as a prototype code, which can be improved significantly for computational efficiency.

4.3 Qualitative Results

We show qualitative performance of our system in Figure 4.3. We show color coded point cloud (blue = near, red = far) and corresponding tracked SMPL mesh for a few key-frames for some of the captured sequences. Even with a very minimal input, our system is able to tackle challenging cases

involving loose clothing and complex poses. Note how our results show correct shape and pose for the following cases : *(i)* significant self-occlusion (1a, 1b, 4c, 5a, 5c). *(ii)* complex and fast motion (1b subject turning around, 2a, 2b kicking) *(iii)* challenging hairstyles (1c, 3c long tied hair, 5a, 5c pony tail)

Figure 4.3: Qualitative results of our algorithm. Left: Segmented point clouds, color coded as Blue (Near) and Red (Far). Right: Tracked SMPL model.

4.4 Summary

In this chapter, we focus on developing a method for full 3D human body shape and motion capture for subjects wearing everyday clothes. We use publicly available SMPL body model for accomplishing the same. Our method has the simple capture set-up of just one depth camera. We show effective articulated motion tracking, by iterating between computation of surface features and performing inverse kinematics fine tuned by a pose refinement algorithm. Despite the simplicity of the method, qualitative results show that it can track challenging poses. In the next chapter we try to improve upon SMPL body model by adding clothing details on top of it and quantitatively evaluating its performance.

Chapter 5

Developing The *Consensus Mesh*

In chapter 4 we have discussed about tracking SMPL body model on depth sequences captured by Kinect. As shown in the results (Figure 4.3) we can track the SMPL model well, but still it seems that something is missing. SMPL is a statistical model, having the ability to change the body shape and size, but it can not look like the person in question without any further post processing. Since our subjects are wearing clothes, adding clothing details becomes a priority. Such a template is analogous to output of a laser scanner. This chapter explains the second major contribution of our work, which is creating a subject specific template mesh ($\mathbf{C_p}$) from a monocular 360° sequence of the person, to assist tracking. Note that such a *CMesh* is good as compared to SMPL model as it provides better adherence to the topology of the 3D data. We now explain our pipeline in detail.

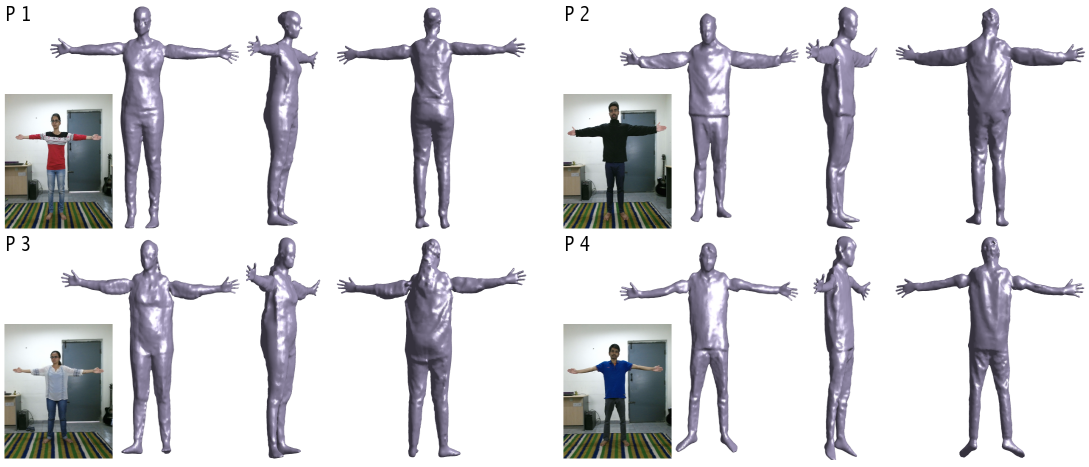

Figure 5.1: Consensus mesh results. Note how the generated meshes are more faithful to the true geometry compared to the SMPL mesh (Best viewed in color).

5.1 Generating subject specific Consensus Mesh

Learning shape and pose based clothing deformations is a hard task. Works like [63] discuss about such challenges in detail, they also propose a new model which learns clothing deformations. In order to accomplish that a setup of multiple cameras is required. The probem becomes difficult when we have input data from a singular point of view. But if we want a template mesh of a person which looks good, alternate approaches can be tried. Here we aim at developing such a *CMesh* which is a 3D mesh but has an underlying SMPL model. We show that such a model can improve the results both qualitatively and quantitatively. We now discuss step by step process of our pipeline.

Retrieving candidate frames:
We run tracking framework (section 4.2) on the sequence to get SMPL model parameters (Θ_t). We treat $\Theta_f = \Theta_1$ as our front canonical frame. Rotating the root of Θ_f by 180° we get Θ_b, which gives us back canonical frame. We retrieve $\mu_{\mathbf{f}}$ and $\mu_{\mathbf{b}}$ as the set of closest matching frames based on 3D positions of skeletal joints of Θ_f and Θ_b using nearest neighbor search. During implementation we have kept $|\mu_{\mathbf{f}}| = |\mu_{\mathbf{b}}| = 5$.

Pose alignment:
In order to align meshes in $\mu_{\mathbf{f}}$ and $\mu_{\mathbf{b}}$ with their respective canonical frames Θ_f and Θ_b we do pose cancellation with respect to frame zero. For this we perform a reverse transformation for each point cloud mesh $\mathbf{M_i} \in \mu_{\mathbf{f}}$ to $\mathbf{M_0}$ (which represents the virtual point cloud mesh corresponding to rest pose of SMPL model ($\mathbf{S_0}$)). We then repose $\mathbf{M_0}$ to $\mathbf{M_f}$ using Θ_f. We perform same operations on $\mathbf{M_i} \in \mu_{\mathbf{b}}$. This gives us pose aligned nearest neighbor set $\mu_{\mathbf{f}'}$ and $\mu_{\mathbf{b}'}$.

Figure 5.2: Left: RGB Image in starting pose. Middle: $\mathbf{M_1}$ generated using subsection 4.2.1. Right: Front surface approximation using Kinect Fusion ($\mathbf{P_f}$). Notice that $\mathbf{P_f}$ has more details (e.g. wrinkles) compared to $\mathbf{M_1}$.

Approximating front and back surfaces:
Using $\mu_{\mathbf{f}'}$ and $\mu_{\mathbf{b}'}$ as static sequence inputs we execute Kinect Fusion [42] to obtain $\mathbf{P_f}$ and $\mathbf{P_b}$ as an approximation of front and back surfaces of the person. This step helps us in removing structured and

real world noise due to Kinect, while simultaneously enriching the topology Figure 5.2. Pose alignment in the previous step was necessary to remove noise caused due to movement of the person, which is essential for Kinect fusion. We substitute hands and feet from SMPL mesh (S_f) owing to the low resolution of Kinect depth maps in these regions.

Stitching everything together:
In order to merge all estimated surfaces, we repose $\mathbf{P_b}$ to $\mathbf{P_f}$ using Θ_f and fill in the missing regions on the left and right profiles (Figure 5.3) by interpolating vertices of SMPL mesh (S_f). Finally we run Poisson reconstruction ([35]) in Meshlab ([20]) to generate the final *consensus mesh* ($\mathbf{C_p}$) which are shown in Figure 5.1.

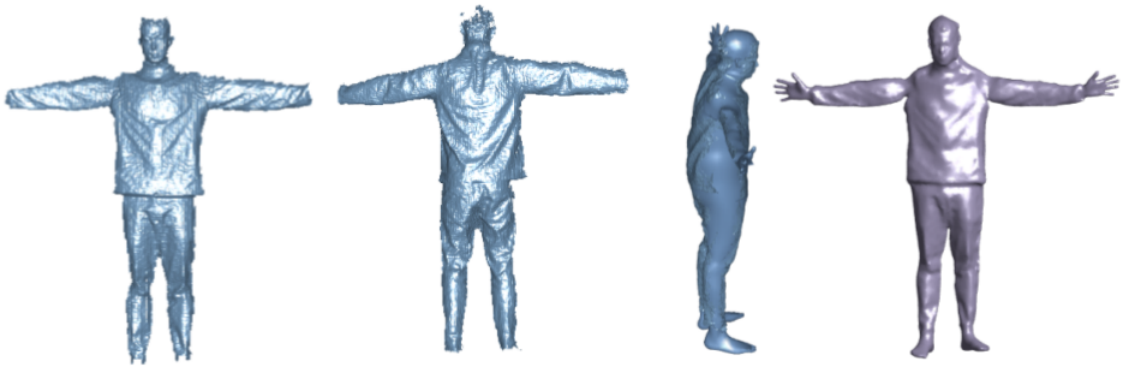

Figure 5.3: Left to Right: $\mathbf{P_f}$, $\mathbf{P_b}$, filling of side profile views, $\mathbf{C_p}$

Repositioning of Consensus Mesh:
In order to repose the Consensus mesh in each frame according to the tracking, we use a set of highly aligned (as per a certain threshold) vertices between the SMPL (S_f) and Consensus mesh (C_p) and perform As Rigid As Possible based surface deformation Sorkine and Alexa [49]. This animates the movements using the Consensus mesh and reduces the tracking error due to better adherence of *CMesh* to the true geometry.

5.2 Importance of Consensus Mesh

As explained above, the CMesh ($\mathbf{C_p}$) is a clothed 3D mesh of a person, with a corresponding parametric SMPL model. $\mathbf{C_p}$ adheres better to the topology of the loosely-clothed person (Figure 5.4) and the underlying SMPL allows it to be animated in plausible ways as shown before. The combination of CMesh and SMPL can be used for better human tracking, learning body shapes and cloth segmentation. The pair can also be used for pose related cloth deformations to re-target body shapes or virtual avatars, captured in more realistic setting.

Figure 5.4: Left: RGB image, Middle: SMPL model, Right: Generated *CMesh*. Notice that the details captured by *CMesh* are better as compared to SMPL.

5.3 Experiments

For the purpose of evaluating the performance of our algorithm we captured several RGB-D sequences (refer chapter 3). Our subjects included 4 males and 4 females. Subjects wore challenging everyday clothes like Hoodie, Jeans, T-shirt, Loose top , different hairstyles etc. We recorded 11 sequences per subject (7 common and 4 different actions). Recorded actions included simple exercises, athletic action, Yoga poses etc. To emulate real world setting all sequences were recorded without any special background or body markers. We will be releasing our implementation and the entire dataset to help research in this area. For the purpose of quantitative evaluation we use dataset released by De Aguiar et al. [24] and Vlasic et al. [55].

5.3.1 Qualitative Results

We show qualitative performance of our system in Figure 5.5. We show color coded point cloud (blue = near, red = far), corresponding tracked SMPL mesh and reposed *consensus mesh* for a few keyframes for some of the captured sequences. Even with a very minimal input, our system is able to tackle challenging cases. Note how our results show correct tracked shape and pose for the following cases : *(i)* complex and fast motion (2c subject turning around, 3c kicking) *(ii)* challenging hairstyles (1b long tied hair, 3a pony tail) *(iii)* loose clothing and complex poses (1c, 2c and 3b Hoodie, 1b and 2b loose top) *(iv)* significant self-occlusion (2c, 3a). This highlights the robustness and generality of our framework.

5.3.2 Synthetic Dataset

Contemporary methods that are based on monocular input do not tackle 'temporal tracking' with 'statistical body model' fitting specifically for subjects in 'casual clothes'. Lack of implementation resources (codes, complete datasets etc.) make comparisons hard to do. Hence to show objective effec-

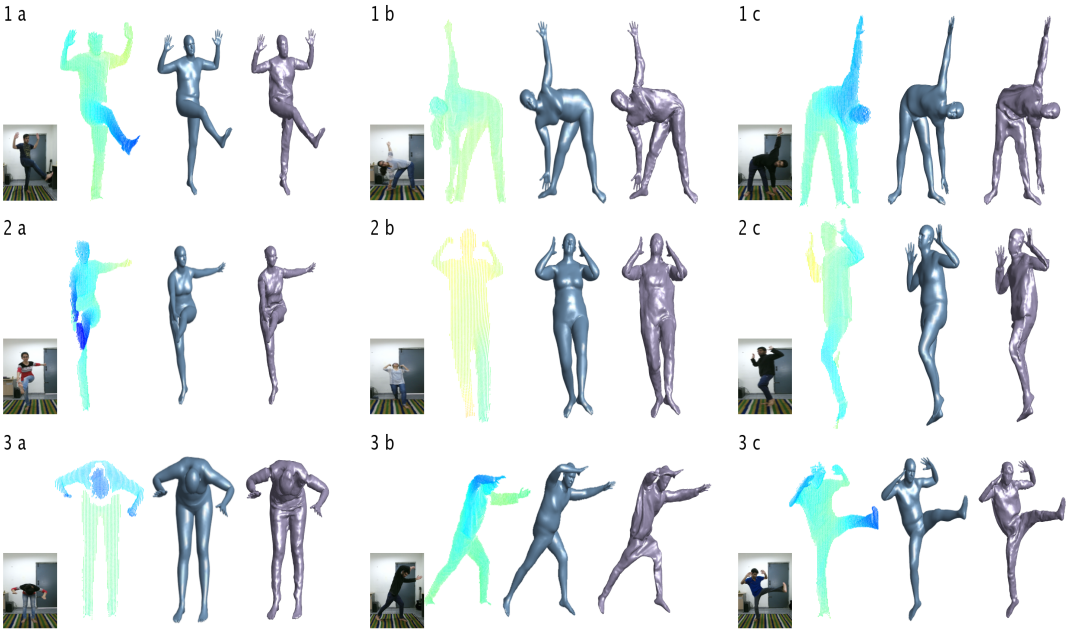

Figure 5.5: Qualitative results of our framework on our dataset. Each figure shows the input point cloud (color coded), pose refined SMPL mesh and reposed CMesh. In inset figure we also show corresponding RGB image. Please refer to the supplementary video [4] to notice the various challenging poses, actions, hairstyles and clothing worn by the subjects (Best viewed in color).

tiveness of our system we ran our algorithm on the dataset by De Aguiar et al. [24] and Vlasic et al. [55]. We use their results as our ground truth. To test our system we generate synthetic depth maps from a singular point of view using OpenGL. We additionally created a virtual 360° sequence of the person by rotating few ground truth meshes. We created *consensus mesh* using this sequence.

5.3.3 Quantitative Results

For quantifying the error of our SMPL tracking with respect to ground truth meshes (G_t) we compute mean absolute drift error ϵ_t, as follows: We find correspondences between vertices $a \in S_1$ and $b \in G_1$ by nearest neighbor search to get an ordered set $(a, b) \in C$. Consider a time step $F_t \rightarrow F_{t+1}$. during which $(a_t, b_t) \rightarrow (a_{t+1}, b_{t+1})$. For this transition we define ϵ_t as : $\epsilon_t = d(a_{t+1}, b_{t+1}) - d(a_t, b_t)$. Here $d(x, y)$ is the euclidean distance. ϵ_t measures error with respect to motion over time but does not tell us anything about how close the geometry of our solution is to the ground truth mesh.

Hausdorff distance 'd_H' is one way to measure the same, but since it is sensitive to outliers we compute percentile based Hausdorff distance 'd_l' as:

$$d_l(P, Q) = \max \left\{ \max_j^{1\%} \min_i \left|\left| y_i^p - y_j^q \right|\right|, \max_i^{1\%} \min_j \left|\left| y_i^p - y_j^q \right|\right| \right\}, \tag{5.1}$$

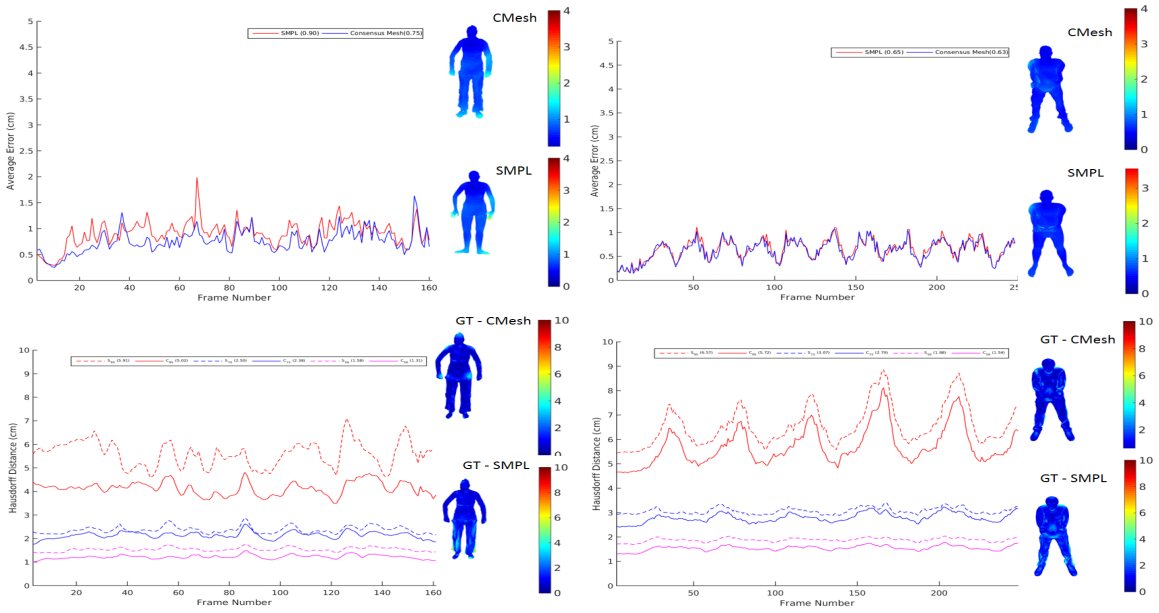

Figure 5.6: Quantitative results on De Aguiar et al. [24] and Vlasic et al. [55]. Top row shows graphs for mean drift error (ϵ_t) for various frames in the sequence (S31 [24] and I_squat [55]). The color coded CMesh and SMPL mesh represents average (ϵ_t) per vertex over all frames. Bottom row shows percentile based Hausdorff distance (d_l) per frame (for various $l = 95\%$ (red)), 75% (blue), 50% (magenta)). In the graph S stands for SMPL (dotted line) and C stands for CMesh (solid lines) based errors. Color coded ground truth mesh based on nearest neighbor found with respect to SMPL and Cmesh is also shown. (Best viewed in colors on screen)

e.g. when $l = 50\%$ we are taking max over medians. We computed the same error matrices for CMesh with respect to ground truth Figure 5.6. Notice how errors for *CMesh* are low as compared to SMPL even for $l = 95\%$. Notice the significant percentage error reduction of repositioned *CMesh* with respect to SMPL tracking computed on average of ϵ_t and d_l for all frames ($\epsilon_t^{'}$ and $d_l^{'}$) in Table 5.1.

Sequence	$\epsilon_t^{'}$	$d_{95}^{'}$	$d_{75}^{'}$	$d_{50}^{'}$
S08	2.40%	5.25%	10.58%	19.55%
S31	16.67%	14.96%	4.56%	17.06%
I_crane	15.45%	9.21%	12.50%	23.78%
I_jumping	12.88%	5.33%	8.04%	14.18%
I_march	16.08%	5.51%	10.25%	20.44%
I_squat	3.07%	12.97%	9.13%	17.97%

Table 5.1: Percentage error reduction of repositioned CMesh with respect to SMPL tracking computed over ($\epsilon_t^{'}$ and $d_l^{'}$). Top two sequences are from [24] and bottom four from [55]

5.4 Summary

In this chapter we have tried to improve upon naked SMPL body model by creating a *Consensus Mesh*, which is a 3D mesh template of the person created from a 360° monocular sequence of the person. The *CMesh* is created by first approximating front and back surfaces of the person using Kinect Fusion. Using such technique removes structured noise from the Kinect sensor and improves the topology of the final surface. We have shown that animating *CMesh* using SMPL gives good qualitative results, and reduces error during tracking. In the next chapter we explore the possibility of improving *CMesh* using RGB images.

Chapter 6

Towards Fine Registration of Consensus Mesh

This chapter aims at doing fine registration of *CMesh* presented in chapter 5. This step is necessary as *CMesh* generated till now though has same topology as ground truth (refer Figure 5.1), still lacks finer detail. *CMesh* can be treated as a rough estimate or a proxy object $\mathbf{M}'$ which exists in the same space and pose as the original object $\mathbf{M}$. There are some errors or nuances in $\mathbf{M}'$ with respect to $\mathbf{M}$, and we need to correct these nuances as part of fine registration.

Zhang et al. [63] tells us that when it comes to the area of improving 3D details of a surface in cases where we already have some rough estimate (like SMPL body model or *CMesh*), best way to remove those nuances is to have estimates in 3D itself. We agree with the theory and it is definitely suitable and a necessity for developing new parameterized body models. But such systems have heavy cost of setting the infrastructure itself. Hence we focus on improving *CMesh* using RGB images, which is beneficial in cases like ours where we have good RGB images but extremely noisy 3D data.

6.1 Modeling Nuances

One way to model these nuances is by using normals, as normal define fine details of a 3D surface. With the recent development in deep learning techniques (CNN, VAEs, hierarchical CRFs) researchers have attempted to learn normal maps directly from RGB. Recently [57], [36] have shown good results on challenging datasets. This idea is more common and works well for general case. But we argue that learning normals is a hard problem compared to learning relative normal maps, because of the fact that normals are rotation variant which makes it harder for a network to learn it from similar looking RGB patches. Problem becomes even more harder when you have same lighting conditions from many sides such that RGB patches are exactly the same and normals are different. Decoupling texture and light becomes even harder for the network in these cases.

Such problems are not there when it comes to normal differences. Approaches where the problem was made simpler deliberately to improve the results are common these days Nguyen-Phuoc et al. [44] show excellent results in rendering directly from 3D shapes. Here the authors explicitly provide

camera pose, and light information as input to the network in order to make the learning simpler. CNNs though popular are best suitable for discriminative tasks Isola et al. [32], We wanted to explore Generative Adversarial Networks here as it is a fairly new generative model with lot of potential.

6.2 Problem Definition

As explained earlier our aim is to learn nuances in proxy object $\mathbf{M}'$ via RGB images of $\mathbf{M}$. Full pipeline of our approach can be seen in Figure 6.1. This section explains independent subsystems of the pipeline in detail.

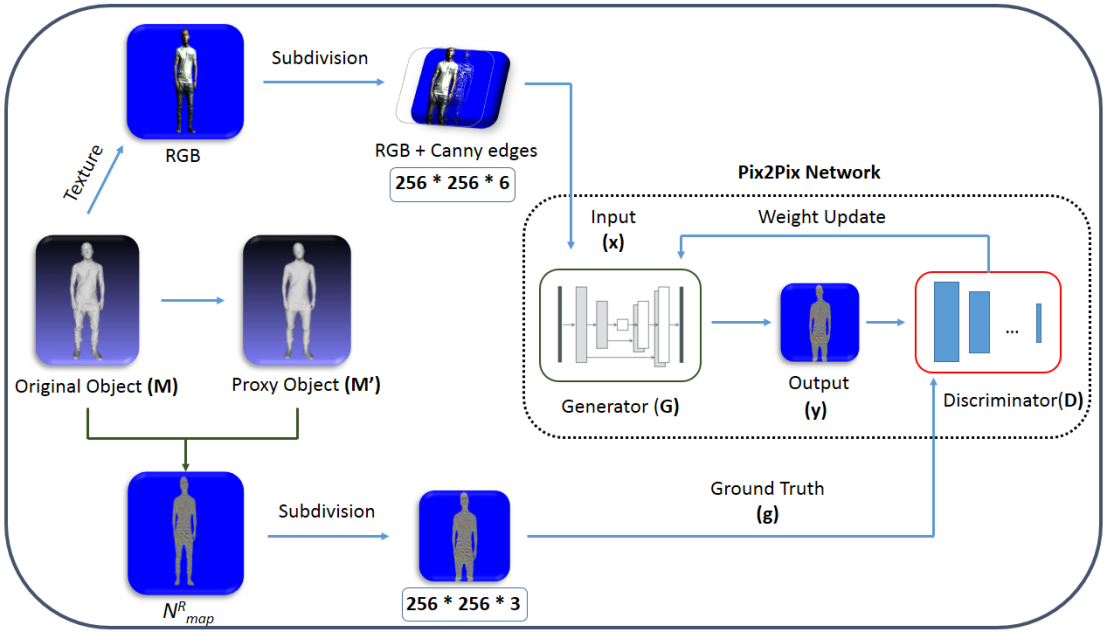

Figure 6.1: Training phase of our pipeline, Note that the input to the generator **(G)** is of dimension $256 \times 256 \times 6$ which is formed by stacking RGB image and corresponding Canny edges.

6.3 Dataset Generation

For the purpose of our experiments we downloaded textured 3D objects $(\mathbf{M})$ from [5]. Although there are many publicly available datasets [19], most of them have objects with highly planar structure. Such objects are not suitable for our work, as we want the objects to have bumps and wrinkles in 3D. Though our method is general and does not limit object shape, for present work we limit ourselves to human looking objects. We perform all our experiments on Human [3] and Dwarf [2] objects. We now explain how we use them to generate complete dataset (Figure 6.2) which we further use.

Figure 6.2: Dataset used in our experiments, Left: Top row shows texture images **T1, T2** and **T3** for Dwarf dataset and Bottom row has both models shown in **T1, T2, T3** respectively. Right: Input pair (RGB and N_{map}) for the network and corresponding canny edges.

6.3.1 Proxy Object

We generate a low polygon, smoothed version of $(\mathbf{M})$ using Quadric Edge Collapse Decimation and Laplacian smoothing filters available in Cignoni et al. [21]. This gives us a proxy $(\mathbf{M}')$ for the original object $(\mathbf{M})$. Note that there are various approaches to do the same but we want our proxy object to have considerable nuances with respect to $\mathbf{M}$ and Laplacian smoothing works well in this area. As can be seen in Figure 6.1 $\mathbf{M}'$ has a very smooth topology and all the significant details are in $\mathbf{M}$.

6.3.2 Texture Maps

To show that cGANs can decouple texture and shading information, and can learn the 3D surface underneath we performed our experiments on three different textures. These texture maps will also help us in showing texture invariance of our approach in inverse rendering problems. Let the original texture image of the object as given on [5] be $\mathbf{T1}$, we generate $\mathbf{T2}$ as:

$$H_{T_2} = 1 - H_{T_1} \tag{6.1}$$

Where H is Hue component of the image. $\mathbf{T2}$ will help us understand how the network is performing when color of the images are different but brightness conditions are same. $\mathbf{T3}$ was generated as a random blocky image with blocks of size $128 * 128$ (Figure 6.2). As this texture has no correspondence to how the object should look like and simply attaches random colors to $\mathbf{M}$, learning from this texture must be harder. For all the experiments the UV coordinates of $\mathbf{M}$ remain the same.

6.3.3 Configurations

We first center $(\mathbf{M})$ at $(0, 0, 0)$ and normalize it to fit in a unit cube. We fix a point source of white light at $(5, 5, 5)$. Taking Euler angles x, y, z as $0°$, $30°$, $60°$ ... $330°$ we compute rotation matrix R as:

$$R = R_x R_y R_z \tag{6.2}$$

We multiply R with model matrix in OpenGL to generate different configurations of the object. This is same as keeping the model static and moving the camera. It gives us 1728 images which are generated by different camera angles under same lighting conditions. All images are rendered using Phong shading model. We followed $75\%, 15\%$ and 15% split to generate Training, Validation and Test datasets.

6.3.4 Normal maps

To model nuances one can go for any relative 3D attributes like depth, normals etc. We use normals in our algorithm as they give details at a finer level. To show that relative normals can be learned with less error with respect to normals we perform experiments with both normal map N_{map} for $\mathbf{M}$ and relative normal map N^R_{map} for $\mathbf{M'}$. Using N_{map} in a subsequent pipeline one can compute N^R_{map} by taking image differences, but here we try to learn such N^R_{map} from RGB itself. For computing N^R_{map} we modify the vertex normals of $\mathbf{M'}$ as:

$$N'_{v'} = N'_{v_0} - N'_{v'} \tag{6.3}$$

where $N'_{v'}$ is the vertex normal of $v' \in \mathbf{M'}$ and v_0 is the approximate nearest neighbor of v' in $\mathbf{M}$. We then render the relative normal map (N^R_{map}) using OpenGL by treating these $N'_{v'}$ as true vertex normals of $\mathbf{M'}$.

6.4 Network Architecture

We use the GAN architecture proposed in Isola et al. [32], who have explored GANs in the conditional setting (cGANs), which allows the network to learn a conditional generative model. One major advantage of using cGANs is that they give good results on a variety of dataset. For the generator, author use a "U-net"- based architecture and for the discriminator a convolutional "PatchGAN" classifier, which penalizes structure at the scale of image patches.

Both generator **G** and discriminator **D** use modules of the form convolution - BatchNorm - ReLu [31]. Following the general shape of a "U-Net" Ronneberger et al. [46], generator was modified to have skip connections between each layer i and layer n - i, where n is the total number of layers. Each skip connection simply concatenates all channels at layer i with those at layer n - i.

This architecture is based on the fact that though the input and output to the network differ in surface appearance, both are renderings of the same underlying structure and hence are roughly aligned. Since this is true for our problem as well we chose to use the same architecture for our experiments.

6.4.1 Network objective:

Conditional GANs learn a mapping from observed image x, random noise vector z to output image y i.e. $G : x, z \rightarrow y$. The generator **G** is trained to produce realistic outputs which cannot be distinguished from "real" with the help of a discriminator **D**, which is simultaneously trained to detect the Generators "fake". The objective of a conditional GAN can be expressed as:

$$\mathcal{L}_{cGAN}(G, D) = E_{x,y\sim p_{data}(x,y)}[\log D(x, y)]+ \\ E_{x\sim p_{data}(x),z\sim p_z(z)}[\log(1 - D(x, G(x, z))]. \quad (6.4)$$

where **G** tries to minimize this objective against an adversarial **D** that tries to maximize it, i.e.

$$G^* = \underset{G}{\operatorname{argmin}} \max_{D} \mathcal{L}_{cGAN}(G, D)$$

Isola et al. [32] also explored the option where **G** is tasked to not only fool **D** but also to be near the ground truth output in an **L1** sense. As **L1** encourages less blurring:

$$\mathcal{L}_{L1}(G) = E_{x,y\sim p_{data}(x,y),z\sim p_z(z)}[||y - G(x, z)||_1].$$

Hence the final objective becomes:

$$G^* = \underset{G}{\operatorname{argmin}} \max_{D} \mathcal{L}_{cGAN}(G, D) + \lambda\mathcal{L}_{L1}(G).$$

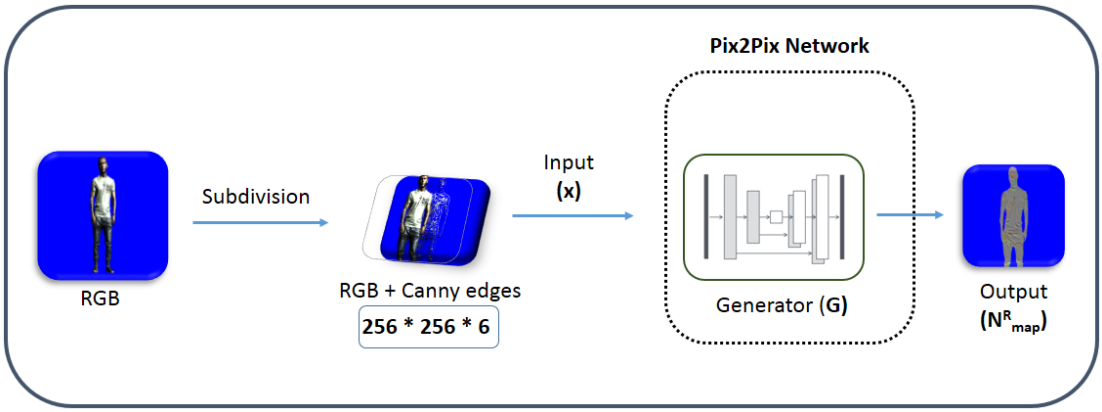

Figure 6.3: Testing phase of our pipeline, Note here we only need RGB images as an input, 3D meshes are not required.

6.5 Network Training

The input to the cGAN is RGB of $\mathbf{M}$ and corresponding output is N^R_{map} of $\mathbf{M}'$. Since the images were larger we subdivided them (Figure 6.2) into non-overlapping blocks of size 256×256 excluding

the blocks which contain 98% background pixels. RGB images are concatenated with their canny edges and this combination is given as input to the network. The reason for doing so is to direct the network towards learning bumps and wrinkles in the images more accurately. We trained the network for 1000 iterations by varying network parameters, details of which are in next chapter.

6.6 Results

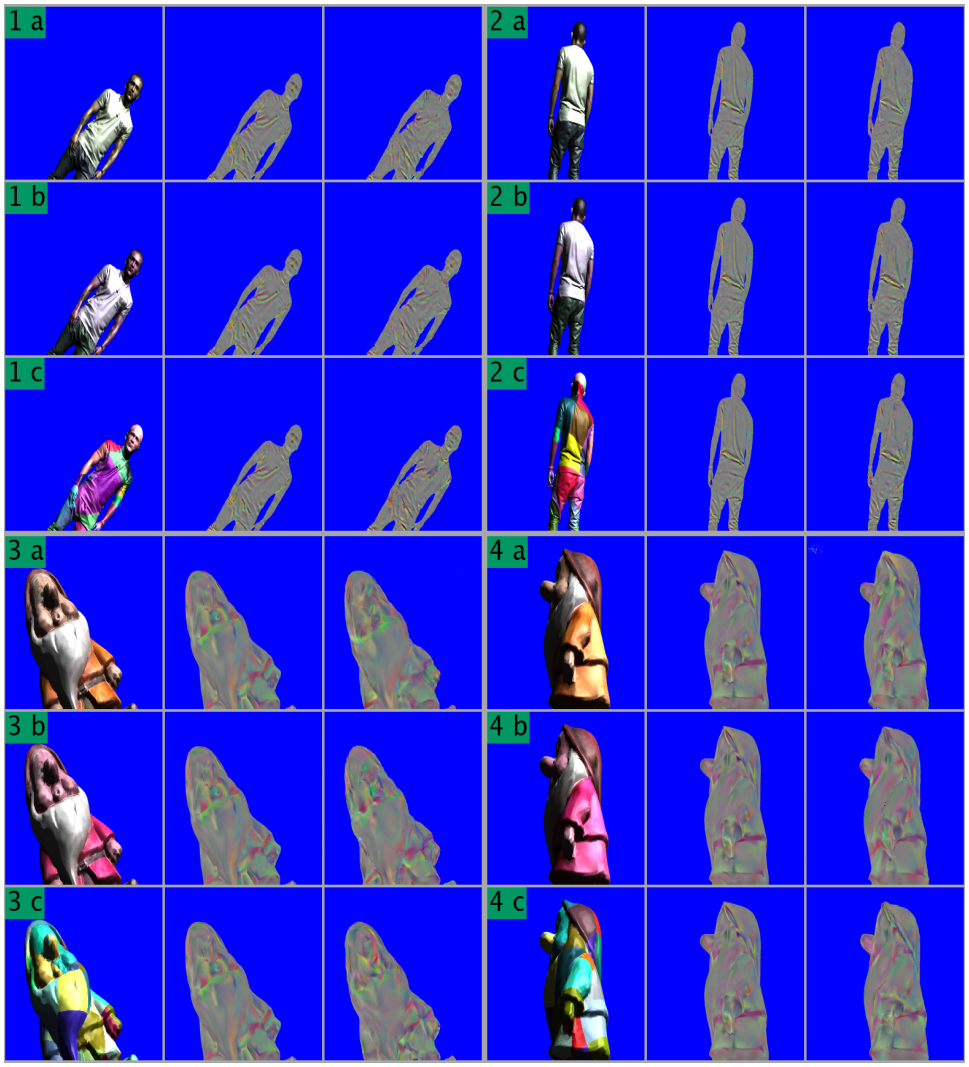

Figure 6.4: Qualitative result of our experiments, each row shows RGB image, N_{map}^{R} and GAN result from left to right. Notice how the network is able to generate convincing and similar results for different textures.

For 'Dwarf' and 'Human' object [5], we created 6 discrete datasets (refer Figure 6.2) by varying their texture (refer 6.3.2). For all the experiments the training and testing split was kept common among

all textures. Network was trained separately for each texture under identical conditions. Qualitative results of the experiments are shown in Figure 6.4.

Trained on ↓	Tested on →		
	T1	**T2**	**T3**
T1	4.65%	4.69%	5.46%
T2	4.97%	4.70%	5.58%
T3	4.97%	4.87%	4.76%

Table 6.1: ϵ' for Dwarf Dataset.

Trained on ↓	Tested on →		
	T1	**T2**	**T3**
T1	5.76%	6.11%	9.87%
T2	5.75%	5.77%	6.94%
T3	5.92%	5.94%	5.85%

Table 6.2: ϵ' for Human Dataset.

We found that the network learn roughly the same structure even though the textures were different. Fine details like wrinkles, bumps etc. are also visible in the generated result. Previous work that are based on cGAN's [32] do not bother about closeness of y to g (ground truth), as the target there is to generate results that are convincing to humans. Here we wanted to quantify such error, hence we compute ϵ for an image as:

$$\varepsilon = 100 \times \frac{\Sigma||g_i - y_i||}{\sigma}.$$

Where $G : x, z \rightarrow y$ and g is the corresponding ground truth image, Since a lot of pixels are comprising of background we do not consider them and hence σ is the count of all non background pixels, and the index i iterates over all of them. In the end we report ϵ' which is average ϵ for the dataset. Note R,G, and B are treated as separate pixels and images were normalized to $[0, 1]$ range.

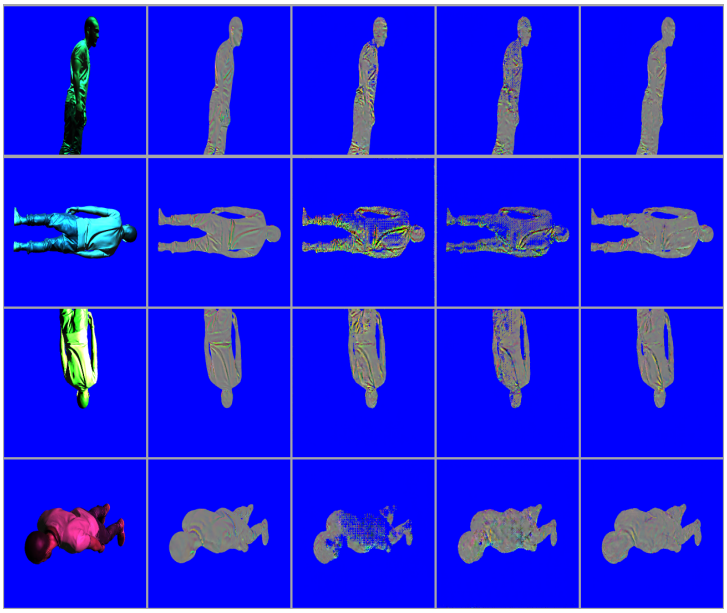

Figure 6.5: Results showing $\mathbf{T3}'\mathbf{s}$ invariance to lighting conditions. From left to right: RGB, N^R_{map}, output of networks learned on $\mathbf{T1}$, $\mathbf{T2}$ and $\mathbf{T3}$ respectively. Notice the consistency in $\mathbf{T3}'\mathbf{s}$ result.

6.7 Discussion

1. An interesting inference that can be drawn from Table 6.1 and Table 6.2 is that when the model is trained on **T3**, the network gave good results on **T1** and **T2** as well, we think this is happening because **T3** is a complex texture to learn, hence the network is forced to look beyond the texture and decouple shading information more accurately.

2. Using N^R_{map} instead of N_{map} gave 2% increase on an average, which shows learning relative attributes is easier.

3. Not only network learned on $\mathbf{T3}$ gave good results on $\mathbf{T1}$, $\mathbf{T2}$. The network handles lighting changes very well Figure 6.5, for showing this we kept the Light position and color of the test dataset as random. Errors for network learned on $\mathbf{T1}$, $\mathbf{T2}$ and $\mathbf{T3}$ were 7.89%, 6.98% and 6.49%.

6.8 Summary

In this chapter we propose an algorithm for doing fine registration of *CMesh*. We show that using N^R_{map} and a cGAN architecture, one can learn fine details of a 3D surface from RGB images. We have shown convincing results on synthetic dataset we created for the purpose. It is also evident that using a random texture map while learning (**T3**) help the network generalize the 3D shape well. Using such a strategy helps cGAN decouple texture and shading information well and produce an understanding of the shape which is invariant to texture and lighting conditions. The work is in initial stage and require more experimental validation.

Chapter 7

Conclusions and Future Work

In this work, we demonstrate a method for full 3D human body shape and motion capture for subjects wearing everyday clothes. Our method has the simple capture set-up of just one depth camera. We show effective articulated motion tracking, by iterating between computation of surface features and performing inverse kinematics regularized by a statistical human body model. Despite the simplicity of the method, our evaluation shows that it can track challenging poses. We also propose a method for creating *Consensus mesh* of a person which can assist in tracking. In our current work we have shown that animating such a *CMesh* using tracked SMPL models improves tracking accuracy. In future we would like to use *CMesh* in our tracking pipeline itself to improve tracking further.

As we do not explicitly restrict the range of possible human poses, our system sometimes generate unnatural poses. Although our system is capable of handling fairly fast actions, it faces issues in highly challenging cases e.g. when fast actions are in conjunction with prominent self-occlusion or profile view. We have also observed that such cases can be corrected if the rate of capturing is fast. We are able to handle large range of casual clothing styles (Figure 5.5) but our method can face issues in extremely challenging cases (e.g. Wedding dresses, Saree, Kimono, etc.), which might require explicit cloth modeling.

We explore the possibility of making the *CMesh* of high quality by modeling the nuances using N^R_{map}. We show good results in learning N^R_{map} using conditional Generative Adversarial Network for synthetic data. To some extent the approach is invariant to lighting and texture changes and has a great potential for future work.

Related Publications

Gaurav Mishra, Saurabh Saini, Kiran Varanasi, and P.J. Narayanan. Human Shape Capture and Tracking at Home. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pages $390 - 399$. IEEE, 2018.

A paper is under preparation based on work of Chapter 6 of this thesis.

Bibliography

[1] Canon camera connect. https://play.google.com/store/apps/details?id=jp.co.canon.ic.cameraconnect&hl=en_IN.

[2] Dwarf object. https://www.turbosquid.com/3d-models/dwarf-2-res-max-free/671354.

[3] Human in casual clothes. https://www.turbosquid.com/3d-models/water-park-slides-3d-max/1093267.

[4] Human shape capture and tracking at home (wacv 2018) - supplementary video. https://www.youtube.com/watch?v=DNqn1O12ICI.

[5] Turbosquid website. https://www.turbosquid.com/.

[6] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[7] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: Shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, July 2005. ISSN 0730-0301.

[8] A. Baak, M. Mller, G. Bharaj, H. P. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *ICCV*, pages 1092–1099, Nov 2011.

[9] L. Bo, X. Ren, and D. Fox. Unsupervised feature learning for rgb-d based object recognition. In *Experimental Robotics*, pages 387–402. Springer, 2013.

[10] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *CVPR*, pages 3794 –3801, Columbus, Ohio, USA, June 2014.

[11] F. Bogo, M. J. Black, M. Loper, and J. Romero. Detailed full-body reconstructions of moving people from monocular rgb-d sequences. In *ICCV*, ICCV '15, pages 2300–2308, Washington, DC, USA, 2015. IEEE Computer Society. ISBN 978-1-4673-8391-2.

[12] 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.

[13] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black. Dynamic FAUST: Registering human bodies in motion. In *CVPR*, July 2017.

[14] A. Bronstein, M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer Publishing Company, Incorporated, 1 edition, 2008.

[15] A. M. Bronstein, M. M. Bronstein, U. Castellani, A. Dubrovina, L. J. Guibas, R. P. Horaud, R. Kimmel, D. Knossow, E. von Lavante, D. Mateus, M. Ovsjanikov, and A. Sharma. Shrec 2010: robust correspondence benchmark. 2010.

[16] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *CVPR*, pages 1704–1711, June 2010.

[17] S. R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. Technical report, IEEE Journal of Robotics and Automation, 2004.

[18] Z. Cai, J. Han, L. Liu, and L. Shao. Rgb-d datasets using microsoft kinect or similar sensors: a survey. *Multimedia Tools and Applications*, 76(3):4313–4355, 2017.

[19] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[20] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In V. Scarano, R. D. Chiara, and U. Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008.

[21] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In V. Scarano, R. D. Chiara, and U. Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008.

[22] J. Cownie, J. DelSignore, B. R. de Supinski, and K. Warren. Dmpl: An openmp dll debugging interface. In *Proceedings of the OpenMP Applications and Tools 2003 International Conference on OpenMP Shared Memory Parallel Programming*, WOMPAT'03, pages 137–146, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 3-540-40435-X.

[23] Y. Cui, W. Chang, T. Nöll, and D. Stricker. Kinectavatar: Fully automatic body capture using a single kinect. In *Proceedings of the 11th International Conference on Computer Vision - Volume 2*, ACCV'12, pages 133–147, Berlin, Heidelberg, 2013. Springer-Verlag.

[24] E. De Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. In *ACM Transactions on Graphics (TOG)*, volume 27, page 98. ACM, 2008.

[25] M. Dekker. Mathematical programming. *CRC, May*, 1986.

[26] E. Dibra, A. C. Öztireli, R. Ziegler, and M. H. Gross. Shape from selfies: Human body shape estimation using cca regression forests. In *ECCV*, 2016.

[27] 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 Trans. Graph.*, 35(4):114:1–114:13, July 2016.

[28] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (TOG)*, 35(4):114, 2016.

[29] J. Gall, C. Stoll, E. D. Aguiar, C. Theobalt, B. Rosenhahn, and H. peter Seidel. Motion capture using joint skeleton tracking and surface estimation. In *CVPR*, 2009.

[30] D. A. Hirshberg, M. Loper, E. Rachlin, and M. J. Black. *Coregistration: Simultaneous Alignment and Modeling of Articulated 3D Shape*, pages 242–255. Springer Berlin Heidelberg, 2012.

[31] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[32] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[33] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. 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*, UIST '11, pages 559–568, New York, NY, USA, 2011. ACM.

[34] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi. *Introduction to humanoid robotics*, volume 101. Springer, 2014.

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

[36] B. Li, C. Shen, Y. Dai, A. Van Den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1119–1127, 2015.

[37] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev. 3d self-portraits. *ACM Trans. Graph.*, 32(6):187:1–187:9, Nov. 2013. ISSN 0730-0301.

[38] Y. Liu, C. Stoll, J. Gall, H. P. Seidel, and C. Theobalt. Markerless motion capture of interacting characters using multi-view image segmentation. In *CVPR*, pages 1249–1256, June 2011.

[39] M. Loper. Chumpy library. `https://pypi.python.org/pypi/chumpy`, 2017.

[40] 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.

[41] A. Neophytou and A. Hilton. A layered model of human body and garment deformation. In *Proceedings of the 2014 2Nd International Conference on 3D Vision - Volume 01*, 3DV '14, pages 171–178, Washington, DC, USA, 2014. IEEE Computer Society.

[42] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[43] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*, pages 343–352, June 2015.

[44] T. Nguyen-Phuoc, C. Li, S. Balaban, and Y. Yang. Rendernet: A deep convolutional network for differentiable rendering from 3d shapes. *arXiv preprint arXiv:1806.06575*, 2018.

[45] G. Pons-Moll, J. Romero, N. Mahmood, and M. J. Black. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, 34(4):120:1–120:14, Aug. 2015.

[46] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[47] A. Shapiro, A. Feng, R. Wang, H. Li, M. Bolas, G. Medioni, and E. Suma. Rapid avatar capture and simulation using commodity depth sensors. *Comput. Animat. Virtual Worlds*, 25(3-4):201–211, May 2014. ISSN 1546-4261.

[48] D. Song, R. Tong, J. Chang, X. Yang, M. Tang, and J. J. Zhang. 3d body shapes estimation from dressed-human silhouettes. In *Proceedings of the 24th Pacific Conference on Computer Graphics and Applications*, PG '16, pages 147–156, 2016.

[49] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007.

[50] L. Spinello and K. O. Arras. People detection in rgb-d data. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3838–3843. IEEE, 2011.

[51] C. Stoll, N. Hasler, J. Gall, H. P. Seidel, and C. Theobalt. Fast articulated motion tracking using a sums of gaussians body model. In *ICCV*, pages 951–958, Nov 2011.

[52] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. Real-time expression transfer for facial reenactment. *ACM Transactions on Graphics (TOG)*, 34(6), 2015.

[53] 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.

[54] K. Varanasi, A. Zaharescu, E. Boyer, and R. Horaud. *Temporal Surface Tracking Using Mesh Evolution*, pages 30–43. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[55] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *ACM Transactions on Graphics (TOG)*, volume 27, page 97. ACM, 2008.

[56] T. von Marcard, B. Rosenhahn, M. Black, and G. Pons-Moll. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. *Computer Graphics Forum 36(2), Proceedings of the 38th Annual Conference of the European Association for Computer Graphics (Eurographics)*, 2017.

[57] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 539–547, 2015.

[58] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[59] A. Weiss, D. Hirshberg, and M. J. Black. Home 3d body scans from noisy image and range data. In *ICCV*, pages 1951–1958, Nov 2011.

[60] S. Wuhrer, L. Pishchulin, A. Brunton, C. Shu, and J. Lang. Estimation of human body shape and posture under clothing. *Comput. Vis. Image Underst.*, 127:31–42, Oct. 2014.

[61] J. Yang, J. S. Franco, F. Hetroy-Wheeler, and S. Wuhrer. Estimation of human body shape in motion with wide clothing. In *ECCV*, 2016.

[62] M. Ye and R. Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *CVPR*, pages 2345–2352, 2014.

[63] C. Zhang, S. Pujades, M. Black, and G. Pons-Moll. Detailed, accurate, human shape estimation from clothed 3d scan sequences. In *CVPR*, 2017.

[64] Q. Zhang, B. Fu, M. Ye, and R. Yang. Quality dynamic human body modeling using a single low-cost depth camera. In *CVPR*, pages 676–683, 2014.

[65] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 2000.

[66] M. Zollhöfer, 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 Trans. Graph.*, 33(4):156:1–156:12, July 2014.