Chapter 2: Shape-Aware Human Reconstruction Using Multi-View Images

2.1 Introduction

In order to accurately reconstruct and synthesize garments in virtual try-on systems, it is necessary to obtain a precise estimation of the 3D user body mesh first. Human body reconstruction, including both pose and shape estimation, is an essential building block for realistic virtual try-on systems. Despite its importance in a large number of applications, human body reconstruction still remains a challenging and popular topic of interest. While direct 3D body scanning can provide excellent and sufficiently accurate results, its adoption is somewhat limited by the required specialized hardware. I propose a practical method that can estimate body pose and shape directly from a small set of images (typically 3 to 4) taken at several different view angles, which can be adopted in many applications, such as Virtual Try-On. Compared to existing scanning-based reconstruction, my proposed approach is much easier to use. Compared to previous image-based estimation methods, my method offers a higher degree of accuracy in shape estimation when the input human body is not within a normal range of body-mass index (BMI) and/or when the body is wearing loose clothing. Furthermore, my framework is flexible in the number of images used and this feature considerably extends its applicability.

In contrast to many existing methods, I use "multi-view" images as input, referring to photos taken of the same person with *similar* poses from different viewing angles. They can be taken using specialized multi-view cameras, but it is not necessary (Sec. 2.6.4). Single-view images often lack the necessary and complete information to infer the pose and shape of a human body, due to the nature of projection transformation. By obtaining information from multiple view angles, the ambiguity from projection can be considerably reduced, and the body shape under loose garments can also be more accurately reconstructed.

Previous work on pose and shape estimation of a human body (see Sec. 2.2) mostly rely on optimization. One of the most important metrics used in these methods is the difference between the original and the estimated silhouette. As a result, these methods cannot be directly applied to images where the human wears loose garments, e.g. long coat, evening gown. The key insight of my method is as follows. When estimating a person's shape, how the human body interacts with the cloth, *e.g.* how a t-shirt is shaped due to the push by the stomach or the chest, provides more information than the silhouette of the person. So image features, especially those on clothes, play an important role in the shape estimation. With recent advances in deep learning, it is widely believed that the deep Convolutional Neural Network (CNN) structure can effectively capture these subtle visual details as activation values. I propose a multi-view multi-stage network structure to effectively capture visual features on garments from different view angles to more accurately infer pose and shape information.

Given a limited number of images, I incorporate prior knowledge about the human body shape to be reconstructed. Specifically, I propose to use the Skinned Multi-Person Linear (SMPL) model [30], which uses Principal Component Analysis (PCA) coefficients to represent human body shapes and poses. In order to train the model to accurately output the coefficients for the SMPL model, a sufficient amount of data containing ground-truth information is required. However, to the best of my knowledge, no such dataset exists to provide multiple views of a loosely clothed body with its ground-truth shape parameters (i.e. raw mesh). Previous learning-based methods do not address the shape (geometry) recovery problem [31] or only output one approximation close to the standard mean shape of the human body [32], which is insufficient when recovering human bodies with largely varying shapes. Taking advantage of physically-based simulation, I design a system pipeline to generate a large number of multi-view human motion sequences with different poses, shapes, and clothes. By training on the synthetic dataset with ground-truth shape data, my model is "shape-aware", as it captures the statistical correlation between visual features of garments and human body shapes. I demonstrate in the experiments that the neural network trained using additional simulation data can considerably enhance the accuracy of shape recovery.

To sum up, the key contributions of my work include:

- A learning-based *shape-aware* human body mesh reconstruction using SMPL parameters for both pose and shape estimation that is supervised directly on shape parameters.

- A scalable, end-to-end, multi-view multi-stage learning framework to account for the ambiguity of the 3D human body (geometry) reconstruction problem from 2D images, achieving improved estimation results.
- A large simulated dataset, including *clothed* human bodies and the corresponding ground-truth parameters, to enhance the reconstruction accuracy, especially in shape estimation, where no ground-truth or supervision is provided in the real-world dataset.
- Accurate *shape* recovery *under occlusion of garments* by (a) providing the corresponding supervision and (b) deepening the model using the multi-view framework.

2.2 Related Work

In this section, I survey recent works on human body pose and shape estimation, neural network techniques, and other related work that make use of synthetic data.

2.2.1 Human Body Pose and Shape Recovering

Human body recovery has gained substantial interest due to its importance in a large variety of applications, such as virtual environments, computer animation, and garment modeling. Previous works reduce the ambiguity from occlusion using different assumptions and input data. They consist of four main categories: pose from images, pose and shape from images under tight clothing, scanned meshes, and images with loose clothing.

Pose From Images. Inferring 2D or 3D poses in images of one or more people is a popular topic in Computer Vision and has been extensively studied [33, 34, 35, 36, 37]. I refer to a recent work, VNect by Mehta *et al.* [31] that is able to identify human 3D poses from RGB images in real time using a CNN. By comparison, my method estimates the pose and shape parameters at the same time, recovering the entire human body mesh rather than only the skeleton.

Pose and Shape From Images under Tight Clothing. Previous work [38, 39, 40, 41, 42, 43] use the silhouette as the main feature or optimization function to recover the shape parameters. As a result, these methods can only be used when the person is wearing tight clothes, as shown in examples [44, 45]. By training on images with humans under various garments both in real and synthetic data, my method can learn to capture the underlying human pose and shape based on image features.

Pose and Shape From Scanned Meshes. One major challenge of recovering human body from scanned meshes is to remove the cloth mesh from the scanned human body wearing clothes [22]. Hasler *et al.* [46] used an iterative approach. They first apply a Laplacian deformation to the initial guess, before regularizing it based on a statistical human model. Wuhrer *et al.* [47] used landmarks of the scanned input throughout the key-frames of the sequences to optimize the body pose, while recovering the shape based on the 'interior distance' that helps constrain the mesh to stay under the clothes, with temporal consistency from neighboring frames. Yang *et al.* [48] applies a landmark tracking algorithm to prevent excessive human labor. Zhang *et al.* [49] took more advantages of the temporal information to detect the skin and cloth region. As mentioned before, methods based on scanned meshes are limited: the scanning equipment is expensive and not commonly used. My method uses RGB images that are more common and thus much more widely applicable.

Pose and Shape from Images under Clothing. Bălan *et al.* [50] are the first to explicitly estimate pose and shape from images of clothed humans. They relaxed the loss on clothed regions and used a simple color-based skin detector as an optimization constraint. The performance of this method can be easily degraded when the skin detector is not helpful, *e.g.* when people have different skin colors or wear long sleeves. However, my method is trained on a large number of images, which does not require this constraint. Bogo *et al.* [51] used 2D pose machines to obtain joint positions and optimizes the pose and shape parameters based on joint differences and inter-penetration error. Lassner *et al.* [52] created a semi-automatic annotated dataset by incorporating a silhouette energy term on SMPLify [51]. They trained a Decision Forest to regress the parameter based on a much more dense landmark set provided by the SMPL model [30] during the optimization. Constraining the silhouette energy effect to a human body parameter subspace can reduce the negative impact from loose clothing, but their annotated data are from the optimization of SMPLify [51], which has introduced errors inherently. In contrast, I generate a large number of human body meshes wearing clothes, with the pose and shape ground-truth, which can then train the neural network to be "*shape-aware*".

2.2.2 Learning-Based Pose/Shape Estimations

Recently a number of methods have been proposed to improve the 3D pose estimation with calibrated multi-view input, either using LSTM [53, 54], auto-encoder [55, 56] or heat map refinement [57, 58]. They mainly focus on 3D joint positions without parameterization, thus not able to articulate and animate. Choy *et al.* [59] proposed an LSTM-based shape recovery network for general objects. Varol *et al.* [4] proposed a 2-step estimation on human pose and shape. However, both methods are largely limited by the resolution due to the voxel representation. In contrast, my method outputs the entire body mesh with parameterization, thus is articulated with a high-resolution mesh quality. Also, my method does not need the calibration of the camera, which is more applicable to in-the-wild images. Kanazawa *et al.* [32] used an iterative correction framework and regularized the model using a learned discriminator. Since they do not employ any supervision other than joint positions, the shape estimation can be inaccurate, especially, when the person is relatively over-weighted. In contrast, my model is more shape-aware due to the extra supervision from my synthetic dataset. Recent works [60, 61, 62] tackle the human body estimation problem using various approaches; my method offers better performance in either single- or multi-view inputs by comparison.

2.2.3 Use of Synthetic Dataset

Since it is often time- and labor-intensive to gather a dataset large enough for training a deep neural network, an increasing amount of attention is drawn to synthetic dataset generation. Recent studies [21, 63] have shown that using a synthetic dataset, if sufficiently close to the real-world data, is helpful in training neural networks for real tasks. Varol *et al.* [64] built up a dataset (SURREAL) which contains human motion sequences with clothing using the SMPL model and CMU MoCap data [65]. While the SURREAL dataset is large enough and is very close to my needs, it is still insufficient in that (a) the clothing of the human is only a set of texture points on the body mesh, meaning that it is a tight clothing, (b) the body shape is drawn from the CAESAR dataset [66], where the uneven distribution of the shape parameters can serve as a "prior bias" to the neural network, and (c) the data only consists of single view images, which is not sufficient for my training. Different from [63, 64], my data generation pipeline is based on physical simulation rather than pasting textures on the human body, enabling the model to learn from more realistic images where the human is wearing looser garments. Recent works [67, 68] also generate synthetic data to assist training, but their datasets have only very limited variance on pose, shape, and textures to prevent from overfitting. In contrast, my dataset consists of a large variety of different poses, shapes, and clothing textures.

2.3 Overview

In this section, I give an overview of my approach. First, I define the problem formally. Then, I introduce the basic idea of my approach.

Problem Statement: Given a set of multi-view images, $\mathbf{I}_1 \ldots \mathbf{I}_n$, taken for the same person with the same pose, recover the underlying human body pose and shape.

In the training phase, I set $n = 4$, *i.e.* by default I take four views of the person: front, back, left and right, although the precise viewing angles and their orders are not required, as shown in Sec. 2.4.3. To extend my framework to be compatible with single view images, I copy the input image four times as the input. For more detail about image ordering and extensions to other multi-view input, please refer to Sec. 2.4.3. I employ the widely-used SMPL model [30] as my mesh representation, for its ability to express various human bodies using low dimensional parametric structures.

As mentioned before, this problem suffers from ambiguity issues because of the occlusions and the camera projection. Directly training on one CNN as the regressor can easily lead to the model getting stuck in local minima, and it cannot be adapted to an arbitrary number of input images. Inspired by the residual network structure [69], I propose a multi-view multi-stage framework (Sec. 2.4) to address this problem. Since real-world datasets suffer from limited foreground/background textures and ground-truth pose and shape parameters, I make use of synthetic data as additional training samples (Sec. 2.5) so that the model can be trained to be more shape-aware.

2.4 Model Architecture

In this section, I describe the configuration of my network model. As shown in Fig. 2.1, I iteratively run my model for several stages of error correction. Inside

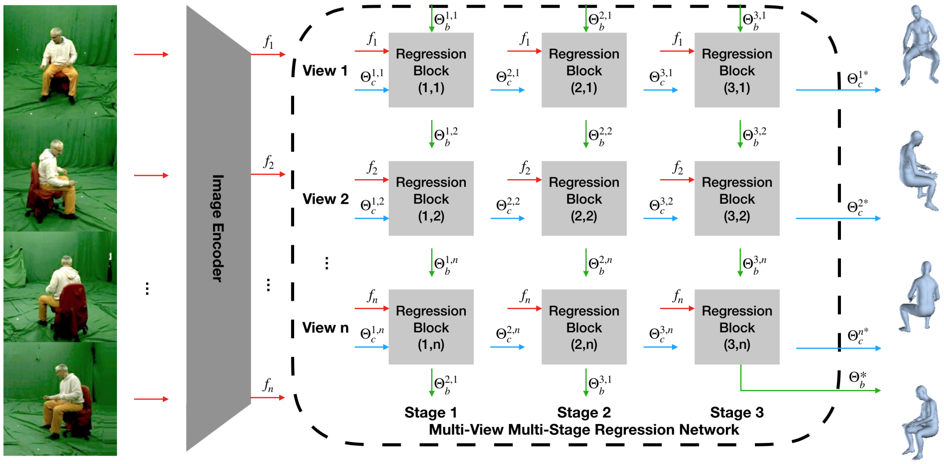


Figure 2.1: The network structure. Multi-view images are first passed through an image encoder to get feature vectors $f_1, ..., f_n$. With initial guesses of the camera parameters $\Theta_c^{1,i}$ and the human body parameters $\Theta_b^{1,1}$, the network starts to estimate the parameters stage by stage and view by view. Each regression block at the i^{th} stage and the j^{th} view regresses the corrective values from image feature f_j (red) and previous guesses $\Theta_c^{i,j}$ (blue) and $\Theta_b^{i,j}$ (green). The results will be added up to the input values and passed to future blocks. While the new human body parameters (green) can be passed to the next regression block, the view-specific camera parameters (blue) can only be passed to the next stage of the same view. Finally, the predictions of the n views in the last stage are outputted to generate the prediction.

each stage, the multi-view image input is passed on one at a time. At each step, the shared-parameter prediction block computes the correction based on the image feature and the input guesses. I estimate the camera and the human body parameters at the same time, projecting the predicted 3D joints back to 2D for loss computation. The estimated pose and shape parameters are shared among all views, while each view maintains its camera calibration and the global rotation. The loss at each step is the sum of the joint loss and the human body parameter loss:

$$L_i = \lambda_0 L_{2Djoint} + \lambda_1 L_{3Djoint} + L_{SMPL} \tag{2.1}$$

where λ_0 and λ_1 scale the units and control the importance of each term. I use L1 loss on 2D joints and L2 loss on others. L_{SMPL} is omitted if there is no ground-truth.

2.4.1 3D Body Representation

I use the Skinned Multi-Person Linear (SMPL) model [30] as my human body representation. It is a generative model trained from human mesh data. The pose parameters are the rotations of 23 joints inside the body, and the shape parameters are extracted from PCA. Given the pose and shape parameter, the SMPL model can then generate a human body mesh consisting of 6980 vertices:

$$\mathbf{X}(\theta, \beta) = \mathbf{W}\mathbf{G}(\theta)(\mathbf{X}_0 + \mathbf{S}\beta + \mathbf{P}\mathbf{R}(\theta)) \tag{2.2}$$

where $\mathbf{X} \in \mathbb{R}^{6980} \times \mathbb{R}^3$ is the computed vertices, $\theta \in \mathbb{R}^{72}$ are the rotations of each joint plus the global rotation, $\beta \in \mathbb{R}^{10}$ are the PCA coefficients, $\mathbf{W}, \mathbf{S}$ and $\mathbf{P}$ are trained matrices, $\mathbf{G}(\theta)$ is the global transformation, $\mathbf{X_0}$ are the mean body vertices, and $\mathbf{R}(\theta)$ is the relative rotation matrix.

For the camera model, I use orthogonal projection since it has very few parameters and is a close approximation to real-world cameras when the subject is sufficiently far away, which is mostly the case. I project the computed 3D body back to 2D for loss computation:

$$\mathbf{x} = s\mathbf{X}(\theta, \beta)\mathbf{R}^T + \mathbf{t} \tag{2.3}$$

where $\mathbf{R} \in \mathbb{R}^2 \times \mathbb{R}^3$ is the orthogonal projection matrix, s and $\mathbf{t}$ are the scale and the translation, respectively.

2.4.2 Scalable Multi-View Framework

My proposed framework uses a recurrent structure, making it a universal model applicable to the input of any number of views. At the same time, it couples the shareable information across different views so that the human body pose and shape can be optimized using image features from all views. As shown in Fig. 2.1, I use a multi-view multi-stage framework to couple multiple image inputs, with shared parameters across all regression blocks. Since the information from multiple views can interact with each other multiple times, the regression needs to run for several iterative stages. I choose to explicitly express this shared information as the predicted human body parameter since it is meaningful and also contains all of the information of the human body. Therefore the input of a regression block is the corresponding image feature vector and the predicted camera and human body parameters from the previous block. Inspired by the residual networks [69], I predict the corrective values instead of the updated parameters at each regression block to prevent gradient vanishing.

I have n blocks at each stage, where n is the number of views. Since all the input images contain the same human body with the same pose, these n blocks should output the same human-specific parameters but possibly different camera matrices. Thus I share the human parameter output across different views and the camera transformation across different stages of the same view. More specifically, the regression block at the i^{th} stage and the j^{th} view takes an input of $(f_j, \Theta_c^{i,j}, \Theta_b^{i,j})$, and outputs the correction $\Delta\Theta_c^{i,j}, \Delta\Theta_b^{i,j}$, where f_j denotes the j^{th} image feature vector, $\Theta_c^{i,j}$ is the camera matrices and $\Theta_b^{i,j}$ is the human parameters. After that, I pass $\Theta_c^{i+1,j} = \Theta_c^{i,j} + \Delta\Theta_c^{i,j}$ to the next **stage** of the block at the same view, while I pass $\Theta_b^{i,j+1} = \Theta_b^{i,j} + \Delta\Theta_b^{i,j}$ to the next **block** of the chain (Fig. 2.1). At last, I compute the total loss as the average of the prediction of all n views in the final stage. Different from static multi-view CNNs which have to fix the number of inputs, I make use of the RNN-like structure in a cyclic form to accept any number of views, and avoid the gradient vanishing by using the error correction framework.

2.4.3 Training and Inferring

Intuitively I use $n = 4$ in my training process, since providing front, back, left, and right views can often give sufficient information about the human body. I choose a random starting view from the input images to account for the potential correlation between the first view and the initial guess. A specific order of the input views is not required since (a) the network parameters of each regression block are identical, and (b) none of the camera rotation information are shared among different views. To make use of large public single-view datasets, I copy each instance to 4 identical images as my input.

During inference, my framework can adapt to images with any number of views n as shown below. If $n \leq 4$, I use the same structure as used for training. I can pad any of the input images to fill up the remaining views. As each view is independent in terms of global rotation, the choice of which view to pad does not matter. If $n > 4$, I extend my network to n views. Since this is an error-correction structure, the exceeded values introduced by extra steps can be corrected back. Note that the number of camera parameter corrections of each view always remains the same, which is the number of stages.

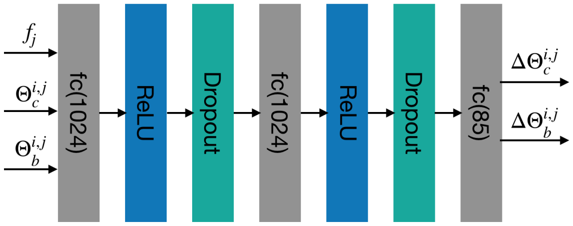


Figure 2.2: Detailed network structure of the regression block at the i^{th} stage and the j^{th} view. f_j denotes the image feature of the j^{th} view, $\Theta_c^{i,j}$ denotes the camera parameters, and $\Theta_b^{i,j}$ denotes the human body parameters.

2.4.4 Implementation Details

During training, besides my synthetic dataset for enhancing the shape estimation (detailed discussion in Sec. 2.5), I train on MS-COCO [70], MPI_INF_3DHP [71] and Human3.6M [72] datasets. Each mini-batch consists of half single view and half multi-view samples. Different from HMR [32], I do not use the discriminator. This is because (a) I initialized my parameters as the trained model of HMR [32], (b) the ground-truth given by my dataset serves as the regularization to prevent unnatural pose not captured by joint positions (*e.g.* foot orientations), and most importantly, (c) the ground-truth SMPL parameters from their training dataset does not have sufficient shape variety. Enforcing the discriminator to mean-shape biased dataset will prevent the model to predict extreme shapes. I use 50-layer ResNet-v2 [73] for image feature extraction. The detailed structure inside the regression block is shown in Fig. 2.2. I fix the number of stages as 3 throughout the entire training and all testing experiments. The learning rate is set to 10^{-5}, and the training lasts for 20 epochs. Training on a GeForce GTX 1080 Ti GPU takes about one day.

2.5 Data Preparation

To the best of my knowledge, there is no public real-world dataset that captures motion sequences of human bodies, annotated with pose and shape (either using a parametric model or raw meshes), with considerable shape variation and loose garments. This lack of data, in turn, forces most of the previous human body estimations to focus only on joints. The most recent work [32] that recovers both pose and shape of human body does not impose an explicit shape-related loss function, so their model is not aware of varying human body shapes. In order to make my model shape-aware under clothing, I need data with ground-truth human body shapes where the garments should be dressed rather than pasted on the skin. A large amount of data is needed for training; sampling real-world data that captures the ground-truth shape parameters is both challenging and time-consuming. I choose an alternate method — using synthesized data. In this section, I propose an automatic pipeline to generate shape-aware training data, to enhance the shape estimation performance.

2.5.1 Parameter Space Sampling

I employ the SMPL model [30], which contains pose and shape parameters for human body. Pose parameters are rotation angles of joints. To sample meaningful human motion sequences in daily life, I use the CMU MoCap dataset [65] as my pose subspace. The shape parameters are principle component weights. It is not ideal to sample the shape parameters using Gaussian distribution; otherwise there will be many more mean-shape values than extreme ones, resulting in an unbalanced training data. To force the model to be more shape-aware, I choose to uniformly sample values at $[\mu - 3\sigma, \mu + 3\sigma]$ instead, where μ and σ represent the mean value and standard deviation of the shape parameters.

2.5.2 Human Body Motion Synthesis

After combining CMU MoCap pose data with the sampled shape parameters, it is likely that the human mesh generated by the SMPL model has inter-penetration due to the shape difference. Since inter-penetration is problematic for cloth simulation, I design an optimization scheme to avoid it in a geometric sense:

$$\min \|\mathbf{x} - \mathbf{x_0}\| \quad s.t. \quad g(\mathbf{x}) + \epsilon \leq 0 \tag{2.4}$$

where $\mathbf{x}$ and $\mathbf{x}_0$ stand for the vertex positions, $g(\mathbf{x})$ is the penetration depth, and ϵ is designed to reserve space for the garment. The main idea here is to avoid inter-penetrations by popping vertices out of the body, but at the same time keeping the adjusted distance as small as possible, so that the body shape does not change much. This practical method works sufficiently well in most of the cases.

2.5.3 Cloth Registration and Simulation

Before I can start to simulate the cloth on each body generated, I first need to register them to the initial pose of the body. To account for the shape variance of different bodies, I first manually register the cloth to one of the body meshes. I mark the relative rigid transformation T of the cloth. For other body meshes, I compute and apply the global transformation, including both the transformation T and the scaling between two meshes. At last, I use the similar optimization scheme described in Sec. 2.5.2 to avoid any remaining collisions since it can be assumed that the amount of penetration after the transformation is small.

I use ArcSim [74] as the cloth simulator. I do not change the material parameters during the data generation. However, I do randomly sample the tightness of the cloth. I generally want both tight and loose garments in my training data.

2.5.4 Multi-View Rendering

I randomly apply different background and cloth textures in different sets of images. I keep the same cloth textures but apply different background across different views. I use the four most common views (front, back, left, and right), which are defined w.r.t. the initial human body orientation and fixed during the rendering. I sample 100 random shapes and randomly apply them to 5 pose sequences in

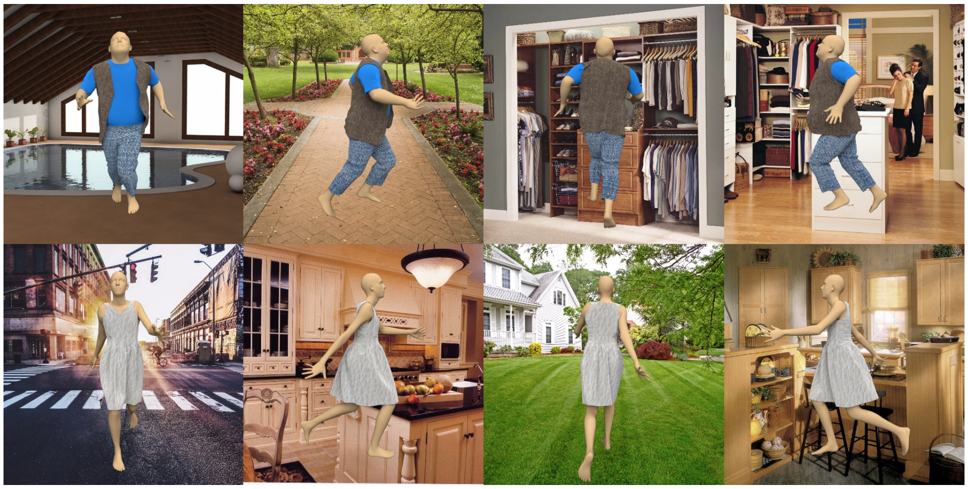


Figure 2.3: Examples of rendered synthetic images. I use a large number of real-world backgrounds and cloth textures so that the rendered images are realistic and diverse.

the CMU MoCap dataset (slow and fast walking, running, dancing, and jumping). After resolving collisions described in Sec. 2.5.3, I register two sets of clothes on it, one with a dress and the other with a t-shirt, pants, and jacket (Fig. 2.3). The pose and garment variety is arguably sufficient because (a) they provide most commonly seen poses and occlusions, and (b) it is an auxiliary dataset providing shape ground-truth which is jointly trained with real-world datasets that have richer pose ground-truth. I render two instances of each of the simulated frames, with randomly picked background and cloth textures. Given an average of 80 frames per sequence, I have generated 32,000 instances, with a total number of 128,000 images. I set the first 90 shapes as the training set and the last 10 as the test set. I ensure the generalizability across pose and clothing by coupling my dataset with other datasets with joint annotations (Sec. 2.4.4).

2.6 Results

I use the standard test set in Human3.6M and the validation set of MPI_INF_3DHP to show the performance gain by introducing multi-view input. Since no publicly available dataset has ground-truth shape parameters or mesh data, or data contains significantly different shapes from those within the normal range of BMI (*e.g.* overweight or underweight bodies), I test my model against prior work (as the baseline) using the synthetic test set. Also, I test on real-world images to show that my model is more *shape-aware* than the baseline method – qualitatively using online images and quantitatively using photographs taken with hand-held cameras.

My method does not assume prior knowledge of the camera calibration so the prediction may have a scale difference compared to the ground-truth. There is also extra translation and rotation due to image cropping. To make a fair comparison against other methods, I report the metrics after a rigid alignment, following [32].

2.6.1 Ablation Study

I conduct an ablation study to show the effectiveness of my model and the synthetic dataset. In the experiments, HMR [32] is fine-tuned with the same learning setting.

2.6.1.1 Pose Estimation

I tested my model on datasets using multi-view images to demonstrate the strength of my framework. I use *Mean Per Joint Position Error* (MPJPE) of the

14 joints of the body, as well as *Percentage of Correct Keypoints* (PCK) at the threshold of 150mm along with *Area Under the Curve* (AUC) with threshold range 0-150mm [75] as my metrics. PCK gives the fraction of keypoints within an error threshold, while AUC computes the area under the PCK curve, presenting a more detailed accuracy within the threshold.

I use the validation set of MPI_INF_3DHP [32] as an additional test dataset since it provides multi-view input. It is not used for validation during my training. I also evaluated the original test set, which consists of single-view images.

Comparison: As shown in Table 2.1 and 2.2, under the same training condition, my model in single-view has similar, if not better, results in all experiments. Meanwhile, my model in multi-view achieves much higher accuracy.

Method	MPJPE w/ syn. training	MPJPE w/o syn. training
HMR	60.14	58.1
Mine (single)	58.55	59.09
Mine (multi)	**45.13**	**44.4**

Table 2.1: Comparison results on Human3.6M using MPJPE. Smaller errors implies higher accuracy.

Method	PCK/AUC/MPJPE w/ syn. training	PCK/AUC/MPJPE w/o syn. training
HMR	86/49/89	88/52/83
Mine (single)	88/52/84	87/52/85
Mine (multi)	**95/63/62**	**95/65/59**

Table 2.2: Comparison results on MPI_INF_3DHP in PCK/AUC/MPJPE. Better results have higher PCK/AUC and lower MPJPE.

2.6.1.2 Shape Estimation

To the best of my knowledge, there is no publicly available dataset that provides images with the captured human body mesh or other representation among a sufficiently diverse set of human shapes. Since most of the images-based datasets are designed for joint estimation, I decide to use my synthetic test dataset for large-scale statistical evaluation, and later compare with [32] using real-world images.

Other than MPJPE for joint accuracy, I use the Hausdorff distance between two meshes to capture the shape difference to the ground-truth. The Hausdorff distance is the maximum shortest distance of any point in a set to the other set, defined as follows:

$$d(V_1, V_2) = \max(\hat{d}(V_1, V_2), \hat{d}(V_2, V_1)) \tag{2.5}$$

$$\hat{d}(V_1, V_2) = \max_{\mathbf{u} \in V_1} \min_{\mathbf{v} \in V_2} \|\mathbf{u} - \mathbf{v}\|^2 \tag{2.6}$$

where V_1 and V_2 are the vertex set of two meshes in the same ground-truth pose, in order to negate the impact of different poses. Intuitively a Hausdorff distance of d means that by moving each vertex of one mesh by no more than d away, two meshes will be exactly the same.

Method	MPJPE/HD w/ syn. training	MPJPE/HD w/o syn. training
HMR	42/83	89/208
Mine (single)	44/65	102/283
Mine (multi)	**27**/**53**	**84**/273

Table 2.3: Comparison results on my synthetic dataset in MPJPE/Hausdorff Distance(HD). Better results have lower values.

As shown in Table 2.3, my model with multi-view input achieves the smallest error values, when compared to two other baselines. After joint-training with synthetic data, all models perform better in shape estimation, while maintaining similar results using other metrics (Table 2.1 and 2.2), i.e. they do not overfit. The joint errors of the HMR [32] are fairly good, so they can still recognize the synthesized human in the image. However, a larger Hausdorff distance indicates that they lose precision on the shape recovery.

Adding my synthetic datasets for training can effectively address this issue and thereby provide better shape estimation. I achieved a much smaller Hausdorff distance (with syn. training) even only using single view. This is because my refinement framework is effectively deeper, aiming at not only the pose but also the shape estimation, which is much more challenging than the pose-only estimation. With the same method, multi-view inputs can further improve the accuracy of shape recovery compared to results using only one single-view image.

2.6.2 Comparisons with Multi-View Methods

Since other multi-view methods only estimate human poses but not the entire body mesh, I compare the pose estimation results to them in Human3.6M. As shown in Table 2.4, I achieved state-of-the-art performance even when camera calibration is unknown and no temporal information is provided. As stated in Sec. 2.6, unknown camera parameters result in a scaling difference to the ground-truth, so the joint error would be worse than what it actually is. After the Procrustes alignment that accounts for this effect, my method achieves the best MPJPE compared to other methods. Another potential source of the error is that my solution is constrained in a parametric subspace, while other methods output joint positions directly. In contrast, my method computes the entire human mesh in addition to joints and the result can be articulated and animated directly.

Method	MPJPE	Known Camera?	Run Time	Temporal Opt?	Articulated?	Shape?
Rhodin *et al.* [76]	-	Yes	0.025fps	Yes	No	Mix-Gaussian
Rhodin *et al.* [55]	98.2	Yes	-	Yes	No	No
Pavlakos *et al.* [57]	56.89	Yes	-	No	No	No
Trumble *et al.* [53]	87.3	Yes	25fps	Yes	No	No
Trumble *et al.* [56]	62.5	Yes	3.19fps	Yes	No	Volumetric
Núñez *et al.* [54]	54.21	Yes	8.33fps	Yes	No	No
Tome *et al.* [58]	52.8	Yes	-	No	No	No
Mine	79.85	No	33fps	No	Yes	Parametric
Mine (PA)	**45.13**					

Table 2.4: Comparison on Human3.6M with other multi-view methods. My method has comparable performance with previous work even without the assistance of camera calibration or temporal information. PA stands for Procrustes Aligned results.

2.6.3 Real-World Evaluations

Method	Standing	Sitting
HMR [32]	7.72%	7.29%
BodyNet [4]	13.72%	29.30%
Mine (single)	6.58%	10.18%
Mine (multi)	**6.23%**	**5.26%**

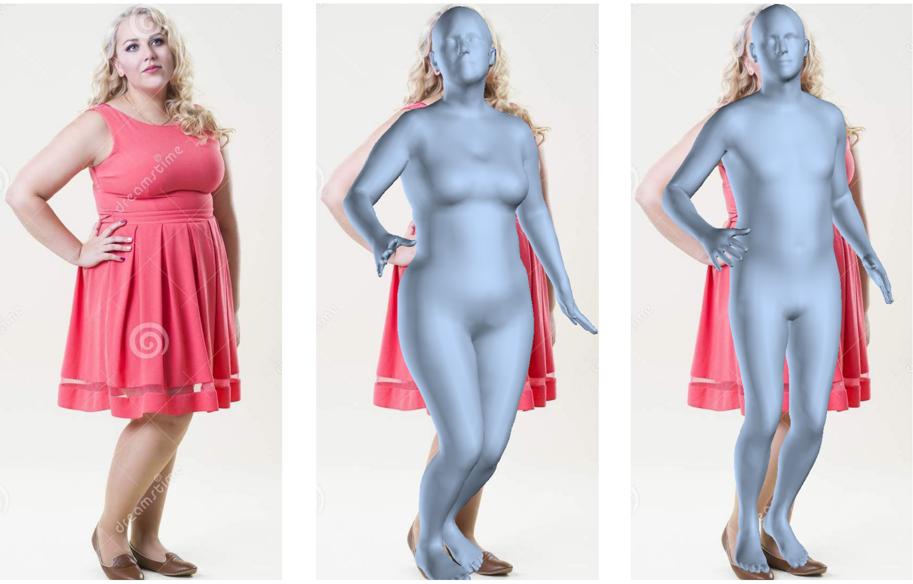
Table 2.5: Comparison results on tape-measured data using average relative errors (lower the better).

I first conduct a study on how my method performs differently with either single- or multi-view inputs under various conditions. My test subjects have two poses: standing and sitting, and the model is additionally tested on two sets of variants from the images. One is slightly dimmed, and the other has a large black occlusion at the center of the first image. I use the percentage of errors from common body measurements used by tailors (*i.e.* lengths of neck, arm, leg, chest, waist, and hip), which is obtained using direct tape measurements on the subjects. I report the average relative error in Table 2.5. It is observed that single-view results are affected by the "occluded sitting" case, while the multi-view input can largely reduce the error. The reason why HMR is not impacted is that they uniformly output average human shapes for all input images. I also report results from BodyNet [4]. BodyNet outputs voxelized mesh and needs a time-consuming optimization to output the SMPL parameters. Its accuracy largely depends on the initial guess. Therefore, it resulted in a large amount of errors on the "sitting" case.

I also tested my model on other online images, where no such measurement can be done. As shown in Fig. 2.4, HMR [32] can predict the body pose but fails on inferring the person's shape. On the contrary, my model not only refines the relative leg orientations but also largely respects and recovers the original shape of the body.

2.6.4 Multi-View Input in Daily Life

It is often difficult to have multiple cameras from different view angles capturing a subject simultaneously. My model has the added benefit of not requiring the multi-view input be taken with the exact same pose. As the model has an error correction structure, it can be applied as long as the poses of the four views are not significantly different. I do not impose any assumptions on the background, so



(a) The input image. (b) My result. (c) HMR.

Figure 2.4: Prediction results compared to HMR. My model can better capture the shape of the human body. The recovered legs and chest are closer to the person in the image.

the images can be even taken with a fixed camera and a “rotating” human subject, which is the typically case when the method is used in applications like virtual try-on.

2.6.5 Extra Test Results

Table 2.6 and 2.8 shows the test results before Procrustes Alignment in MPI_INF_3DHP validation set and Human3.6M, respectively. The same conclusion about over-fitting and multi-view improvement can also be drawn from these data.

Table 2.7 shows the result in MPI_INF_3DHP test dataset. Since there is only one view fed into the model, the results are similar.

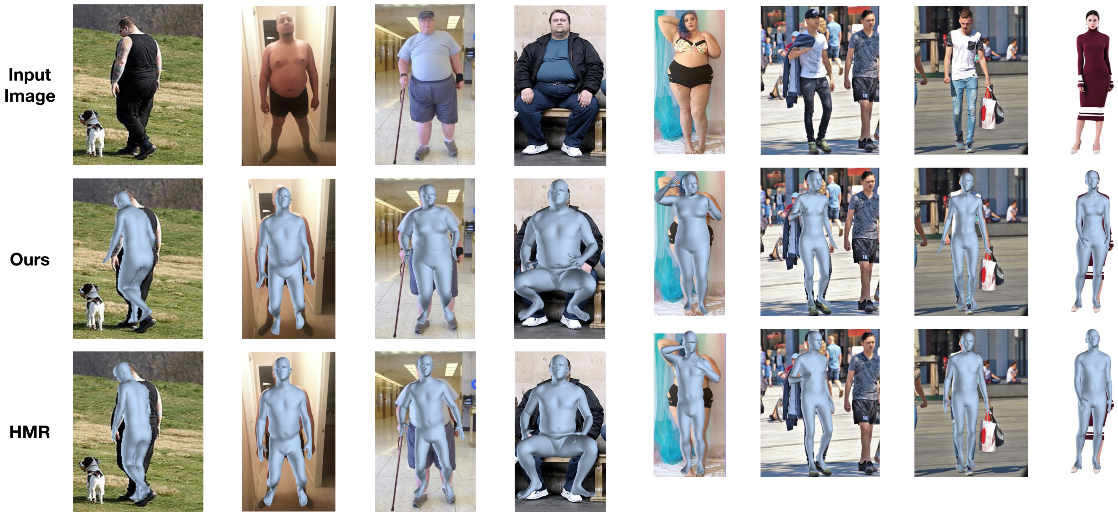


Figure 2.5: Results on images with varying pose and shape. The top row is the input image. The middle row shows my recovery results, and the bottom row shows the results from HMR [32]. Mine achieves better shape recovery results.

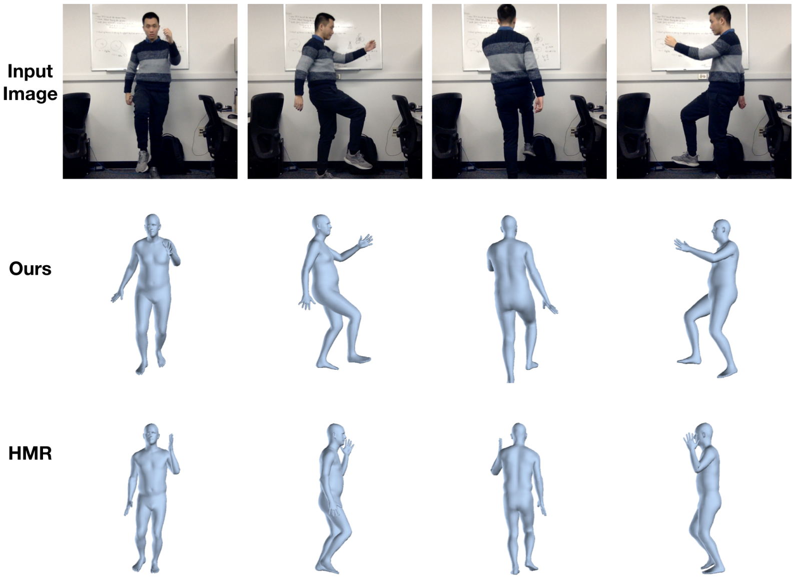


Figure 2.6: Results on real-world multi-view images. The top row is the input image. The middle row shows my recovery results, and the bottom row shows the results from HMR [32]. HMR is only given the front view as input. Mine achieves better pose recovery results due to more view angles.

2.6.6 Additional Results on Real-World Images

As shown in Fig. 2.5, given similar joint estimation results, my model captures more image features that indicate the shape of the human body and thereby gives

Method	PCK/AUC/MPJPE w/ syn. training	PCK/AUC/MPJPE w/o syn. training
HMR [32]	66/33/141	71/36/129
Mine (single)	69/32/139	68/33/138
Mine (multi)	72/34/128	72/35/126

Table 2.6: Results on MPI_INF_3DHP, validation set, before Procrustes aligment.

Method	PCK/AUC/MPJPE w/ syn. training	PCK/AUC/MPJPE w/o syn. training
HMR [32]	65/30/139	65/29/137
HMR (PA)	84/47/91	85/48/89
Mine	65/29/142	66/29/137
Mine (PA)	85/49/89	86/49/89

Table 2.7: Results on MPI_INF_3DHP, test set. The results of [32] are tested on cropped images by Mask-RCNN [77] so the values have minor difference than their reported ones. Only single view is available in this dataset.

much better results in terms of human shape. My method can distinguish between fat (Column 1-5) and slim (Column 6-8) persons, and between male and female. On the other hand, the output shapes from HMR are almost the same, which is around the mean shape value. By incorporating the shape-aware synthetic dataset, my method largely improves the recovery when the input human body does not have an average shape. I also tested with real-world multi-view images vs. single-view HMR. I feed the front view of the subject to HMR but input all views into my model. As shown in Fig. 2.6, the front view does not provide complete information of the subject pose, resulting in large pose errors on the limbs. By sharing information from more views (most importantly side views in this case), my model can effectively reduce the ambiguity from the camera projection and thereby provide good pose estimations across all views.

2.6.7 Comparison on Human3.6M with Single-View Methods

Table 2.8 shows the comparison with single-view results. As mentioned previously, the reason I don't have much better accuracy before rigid alignment is that:

- My method does not assume known camera, resulting in an unknown scaling difference to the real-world coordinates. After the Procrustes alignment, I achieved similar (and better with multi-view) performance.

- My solution is constrained in a subspace. Other methods output joint positions directly so they have more DOF and can be more accurate. However, my output is more comprehensive, as it contains the entire human mesh in addition to joints and the result can be articulated and animated directly.

Compared to Kolotouros *et al.* [62], my model is trained on a much more diverse dataset (*e.g.* MS-COCO), which means that the accuracy may not be minimized on the specific subset (Human 3.6M).

2.6.8 Results Without Training on Synthetic Data

I further tested another variant of my model, which is trained without synthetic data (Fig. 2.7). It achieves better joint estimation, but the recovered human body does not seem to be visually correct, especially at the end-effectors. This is because the joint-only supervision does not impose any constraints on the orientations of the end-effectors, resulting in an arbitrary guess. The HMR model [32] avoids this by adding a discriminator, which however could have negative impact on shape

Method	MPJPE	PA-MPJPE
Tome *et al.* [35]	88.39	-
Rogez *et al.* [78]	87.7	71.6
Mehta *et al.* [31]	80.5	-
Pavlakos *et al.* [33]	71.9	**51.23**
Mehta *et al.* [75]	68.6	-
Sun *et al.* [79]	**59.1**	-
Zhou *et al.* [37]	107.26	-
Debra *et al.* [80]	55.5	-
*Kolotouros *et al.* [62]	**74.7**	51.9
*Omran *et al.* [60]	-	59.9
*Pavlakos *et al.* [61]	-	75.9
*HMR [32]	87.97	58.1
*Mine (single-view)	88.34	58.55
*Mine (multi-view)	79.85	**45.13**

Table 2.8: Results on Human3.6M. My method results in smaller reconstruction errors compared to HMR [32]. * indicates methods that output both 3D joints *and* shapes.

estimations, as discussed in Sec. 2.4.4. My synthetic dataset provides a supervision to not only the joint positions but also the rotations, hence the model will learn a prior at the end-effectors, demonstrating more natural results.

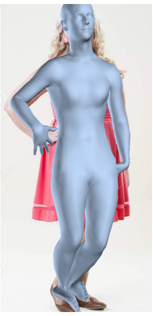


Figure 2.7: My model trained without synthetic data.

2.6.9 Detailed Errors on Real World Evaluation

The error percentages of each measure are shown in Table 2.9. Since the length of the arm and leg can be seen clearly in the front view, both inputs provide a reasonably good estimation. However, given more views, my model can significantly reduce the error on other measurements, especially on those of chest, waist, and hip. I found that image illuminance has a negligible effect on the recovery result, which is due to the translation invariance of the convolutional layers. Occlusion has a notable impact on the recovery using only a single-view image, given only one view of the human body. However, by incorporating more views using my network model, the estimation can be considerably improved, indicating that the model using multi-view images is more robust to occlusion than with a single-view image as input.

error %	Regular				Dimmed				Partly Occluded			
input	Standing		Sitting		Standing		Sitting		Standing		Sitting	
# of views	Single	Multi	Single	Multi	Single	Multi	Single	Multi	Single	Multi	Single	Multi
neck	1.12	12.19	0.048	3.53	0.58	11.31	0.39	2.55	0.45	11.28	22.11	6.11
arm	4.76	4.22	8.03	7.33	6.21	4.95	8.10	6.89	5.20	3.82	7.20	6.70
leg	6.65	4.66	2.94	3.46	5.18	3.92	2.83	3.64	2.53	3.54	4.94	4.24
chest	4.59	7.72	8.40	3.1	6.20	7.20	8.13	3.19	19.80	1.57	30.04	13.72
waist	2.42	12.80	5.46	0.70	3.73	11.98	5.01	0.0084	13.78	8.52	30.05	10.61
hip	8.88	0.62	11.88	5.83	11.36	0.12	11.78	5.50	15.08	1.65	15.95	7.54
error %	Regular				Dimmed				Partly Occluded			
input	Standing		Sitting		Standing		Sitting		Standing		Sitting	
method	HMR	BodyNet	HMR	BodyNet	HMR	BodyNet	HMR	BodyNet	HMR	BodyNet	HMR	BodyNet
neck	10.4	2.9	4.8	26.3	8.4	1.6	4.6	26.2	9.2	3.9	5.7	6.8
arm	6.1	21.3	9.8	25.6	8.6	22.8	9.7	23.6	8.1	19.5	9.7	9.6
leg	7.9	6.3	1.8	4.4	4.3	6.6	1.8	3.3	5.1	6.2	2.1	3.0
chest	11.2	26.3	11.7	51.9	11.7	24.9	11.6	41.3	11.9	24.9	11.6	21.3
waist	9.4	9.0	8.7	42.7	9.4	7.7	8.5	33.7	9.7	8.3	8.4	11.4
hip	1.25	19.2	7.8	79.8	3.5	18.8	7.7	80	2.9	17	5.5	36.9

Table 2.9: Percentages of errors in common measurements of the human body under various lighting conditions using single-view vs. multi-view images. The multi-view model performs significantly better in estimating measurements of chest, waist, and hip, and is more robust, given variations in lighting and partial occlusion.

2.6.10 Evaluation on *3D People in the Wild*.

I have conducted the evaluation on *3D People in the Wild* dataset. As shown in Table 2.10, although the dataset consists of single view images of only a few subjects with nearly standard shapes, my model achieved better accuracy over HMR, while Alldieck *et al*. did not generalize well. The metric I used is mean joint error for pose, and mean vertex error with ground-truth pose for shape.

2.6.11 Running Time

The previous work [32] trained 55 epochs for 5 days, while mine trained 20 epochs for 1 day. I list the training time here for reference, but it is actually not comparable since the batch size, epoch size and GPU type are not the same. In my environment, the inference time of HMR [32] is 2 microseconds while mine takes 7.5 (per view). This is because my network has a deeper structure to account for multiple views.

Method	Mean Joint Err.	Mean Vertex Err. (GT Pose)
HMR	93.77	21.71
Alldieck *et al.* [68]	169.61	47.07
Mine	96.86	20.96

Table 2.10: Evaluation on an unseen single-view dataset: *3D People in the Wild.* Values are mean joint error for pose and mean vertex error with ground-truth pose. My method has smaller errors than Alldieck *et al.*

2.7 Conclusion and Future Work

I proposed a novel multi-view multi-stage framework for pose and shape estimation. The framework is trained on datasets with at most 4 views but can be naturally extended to an arbitrary number of views. Moreover, I introduced a physically-based synthetic data generation pipeline to enrich the training data, which is very helpful for shape estimation and regularization of end effectors that traditional datasets do not capture. Experiments have shown that my trained model can provide equally good pose estimation as state-of-the-art using single-view images, while providing considerable improvement on pose estimation using multi-view inputs and a better shape estimation across all datasets.

While synthetic data improves the diversity of human bodies with ground-truth parameters, a more convenient cloth design and registration are needed to minimize the performance gap between real-world images and synthetic data. In addition, other variables such as hair, skin color, and 3D backgrounds are subtle elements that can influence the perceived realism of the synthetic data at the higher expense of a more complex data generation pipeline. With the recent progress in image style transfer using GAN [81], a promising direction is to transfer the synthetic result to more realistic images to further improve the learning result.

This work has been published in the proceedings of the International Conference on Computer Vision (ICCV) 2019.