DeepDraper: Fast and Accurate 3D Garment Draping over a 3D Human Body

Lokender Tiwari Brojeshwar Bhowmick
TCS Research, India
lokender.tiwari@tcs.com

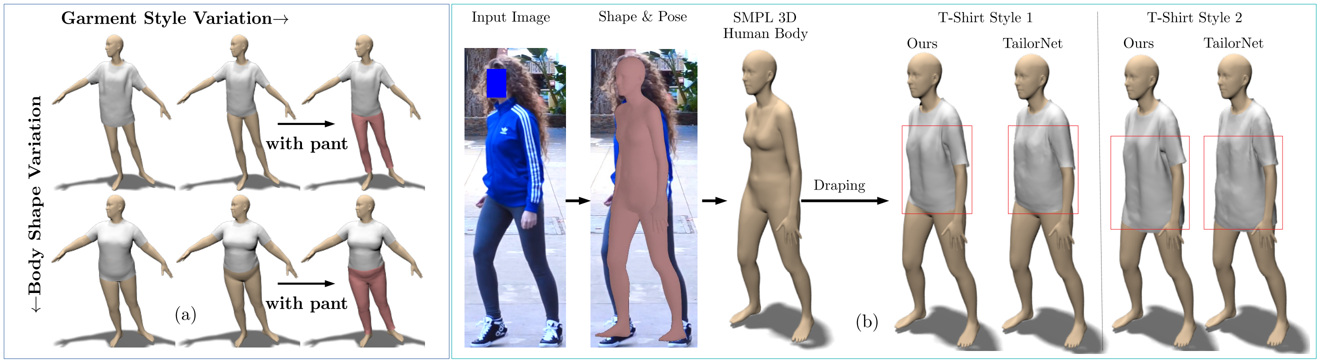


Figure 1: **DeepDraper**: We propose a 3D garment draping method which generalizes to (a) different garment styles and body shape variations, and (b) can drape 3D garment on 3D human body of an arbitrary shape, pose and garment styles. In (b), the SMPL body shape and pose parameters are estimated using ViBE [20]. A bulge around stomach can be seen in the case of TailorNet, while DeepDraper predicts accurate folds and wrinkles. Our model is ∼ 10× smaller in memory size, and ∼ 23× faster than the closest state-of-the-art TailorNet [32] method. Refer supplementary for more qualitative results.

Abstract

Draping a 3D human mesh has garnered broad interest due to its wide applicability in virtual try-on, animations, etc. The 3D garment deformations produced by the existing methods are often inconsistent with the body shape, pose, and measurements. This paper proposes a single unified learning-based framework (DeepDraper) to predict garment deformation as a function of body shape, pose, measurements, and garment styles. We train the DeepDraper with coupled geometric and multi-view perceptual losses. Unlike existing methods, we additionally model garment deformations as a function of standard body measurements, which generally a buyer or a designer uses to buy or design perfect fit clothes. As a result, DeepDraper significantly outperforms the state-of-the-art deep network-based approaches in terms of fitness and realism and generalizes well to the unseen style of the garments. In addition to that, DeepDraper is ∼ 10 *times smaller in size and* ∼ 23 *times faster than the closest state-of-the-art method (TailorNet), which favors its use in real-time applications with less computational power. Despite being trained on the static poses of the TailorNet [32] dataset, DeepDraper generalizes well to unseen body shapes, poses, and garment styles and produces temporally coherent garment deformations on the pose sequences even from the unseen AMASS [25] dataset.*

1. Introduction

Dressing digital humans in 3D [16, 24, 32] have gained much attention due to its use in online shopping, virtual try-on, gaming, 3D content generation etc. Online shopping of clothes provide consumers the comfort of the home, and they get access to a wide range of the latest products without going to the physical stores. However, it has one major limitation: it does not enable buyers to try clothes physically, which results intoa high return/exchange rate due to the cloth fitting issues [29]. The concept of virtual try-on helps to resolve that limitation. It allows buyers to visualize any garment on its 3D avatar as if they are wearing it. The two important factors that a buyer considers while deciding to purchase a particular garment are its *fitting* and *appearance*. In a virtual try-on setup, a person can infer a particular garment's fitness by looking at the *folds* and *wrinkles* in various poses and the gap between the body and the garment in the rendered image or video.

Physics-Based Simulation (PBS) [36, 35, 54] has always been the first choice for generating accurate and realistic cloth draping over a human body. The PBS considers many factors while simulating garments over a human body, making it computationally expensive and non-ideal for real-time /web-based applications. Additionally, a PBS-based garment simulation pipeline requires expert knowledge to design the garment and tune the parameters to get the desired results. The involvement of an expert further increases the cost, and, therefore, it is not scalable.

In contrast to the computational expensive PBS-based methods, the learning-based methods have gained much attention due to their speed and less manual intervention. These methods learn to predict the garment deformation and draping using PBS-based ground-truth data. We can model deformation/animation of a garment on a human body as a function of three important factors: the human *body shape*, the human *body pose* and the *garment style* (e.g., long T-shirt, short T-shirt). Several methods [40, 17, 12, 47, 32] learn garment deformation as a function of one or two of the above factors. While the method in [49] focus on predicting garment style keeping the pose fixed, the method in [40] and the GarNet [17] models the garment deformation as a function of body shape and its pose. The DeepWrinkles [21], and the method in [12] drapes clothes on a fixed body shape. Moreover, the pose retargeting of DeepWrinkles [21] works when new poses are similar to ones included in the training dataset. Apart from body pose and shape, a garment also varies a lot in its style, e.g., a t-shirt can have different variations along its length or sleeve length. Due to these variations, different garment styles deform differently on different body shapes and poses. Therefore, the models trained on single garment style [21, 40, 15] have restricted use. Furthermore, the methods [15, 17] that do not consider different garment styles in their modeling process tend to produce over-smooth results.

To alleviate these problems, a recent method TailorNet [32] learns the garment deformation as a function of body shape, pose, and garment style. The realistic draping results from TailorNet compared to the method in [40] show the importance of considering the garment style in the garment modeling. However, Tailornet has several limitations. Firstly, TailorNet learns one garment deformation model to predict one smooth low-frequency geometry, 20 shape-style specific garment deformation models to capture high-frequency geometry and a RBF kernel which mixes these 20 models to produce final high frequency component of the garment. The fixed number (20) of shape-style specific mixtures makes the method sensitive towards the number of such components (see Fig. 5). Moreover, since each of the 20 shape-style specific models learned separately, it defeats the purpose of jointly modeling all the variations in style, body shape, and poses. Secondly, Tailornet fails to generalize beyond the range of body shape and garment style parameters available in the dataset and the training (see Fig. 2). Finally, Tailornet requires training a new shape-style specific high-frequency predictor and shape-style specific mixture weight predictor with every new shape and style that are not in the range of parameters in the dataset used. Adding a new shape-style specific predictor will further increase the overall size and the inference time of the TailorNet, which is not suitable for portable devices with limited computing power.

In this paper, we overcome these limitations and present a single unified model to learn the garment deformation as a function of body shape, pose, measurements, and garment style. Unlike mixture based approach, we use multi-view perceptual losses to capture the high-frequency geometry of garment deformation. The perceptual loss has been shown effective in image synthesis tasks [30, 50]. We leverage the shading of the folds and wrinkles captured in the multi-view rendered images to guide the network to learn the high-frequency geometry. We use the standard normal map representation of the t-shirt for rendering multi-view images as shown in Fig. 3. Our method differs from method like [21] which again uses un-wrap normal UV-mapping of low resolution (LR) normal maps obtained from blend shape, and high resolution (HR) normal maps obtained from the scans. Instead of the separate LR/HR UV-map, we drape the normal mapped t-shirt on the 3D human body in an arbitrary pose and render images from multi-view to capture shading due to lighting. Finally, we propose to use the standard body measurements such as the size of bust, hip waist, sleeve, etc. Fig. 4(c). Generally, a buyer uses these measurements to buy perfect fit clothes, or a designer uses them to design perfect fit clothes. In our approach, we automatically compute these measurements from the SMPL [22] and use them as an additional factor of variations, which helps improve the fitting of the final predicted garment while maintaining the realism. In Fig. 5 we show a comparison with the TailorNet in the context of garment fitting.

Our main contributions in this paper are:

- A unified deep neural network that learns a garment deformation as a function of body shape, measurements, pose and garment style.
- Our method couples the geometric and the perceptual constraints to efficiently learn the high frequency components (wrinkles and folds) of garment deformation.
- We demonstrate the impact of considering the standard body measurements in modeling garment deformation on the fitting of the final predicted garment.

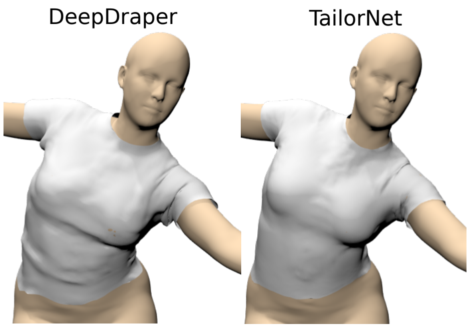
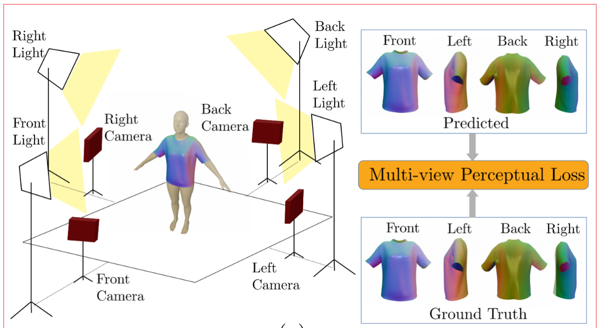


Figure 2: Comparison between DeepDraper and TailorNet on an unseen body pose from AMASS dataset and garment style. Notice the difference in the wrinkles and folds. The garment style parameter we use in this example are $\gamma = [-2.75, 1.0, 0.0, 0.0]$. These parameters are outside the range of the parameters in the simulated TailorNet dataset. While we train DeepDraper on the same dataset as TailorNet. The DeepDraper generalizes well on the garment styles beyond the dataset. Refer, Sec. 5 for detail.



(a)

Figure 3: Multi-view perceptual loss computed between the generated and ground-truth multi-view rendered images of the draped T-shirt on a 3D human with arbitrary pose under multiple lightening. Refer, Sec. 3 and 4.1 for detail.

- Our method generalization well beyond the range of body shape, pose and garment style parameters, where other methods like TailorNet fails or require retraining for the new garment styles of body shape.
- Our model is $\sim 10\times$ smaller and $\sim 23\times$ faster method compared to the closest state-of-the-art TailorNet [32].

2. Related Work

2.1. Garment Reconstruction and Modeling

This category of methods reconstruct the clothed human body from the inputs.

Non-Parametric models: Recovering the whole body of a dressed person from dynamic sequences has been studied in [43, 13, 48]. Recently, several learning-based approaches [27, 38, 45, 56, 39] have demonstrated the reconstruction of combined human body and garments from input images. The final clothed human body predicted by these methods is not parametric, and hence the body pose and shape of the reconstruction cannot be controlled. Although the methods in [10, 10] have demonstrated the re-animation of captured performances, the digital manipulation of the garment alone is still a challenge.

Parametric models: Neophytou and Hilton[28] learns a layered garment model on top of SCAPE[4] 3D human body model from dynamic sequences. Yang et al.[51] train a model to regress a PCA-based representation of clothing. Another set of methods[1, 2, 3, 2, 7, 57] models the garment geometry as an offset over the SMPL[22] 3D human body mesh. These methods can change the final reconstruction's pose and shape using the deformation model of the underlying SMPL body. Recently generative approaches to model the garments have also been proposed in [24, 11]. The reconstructed or generated garments can be used to dress a new subject or train a data-driven model.

2.2. Garment Animation

This class of methods animate a given garment based on a desired body pose or shape or both. Our work belongs to this category.

Physics-Based Simulation (PBS) Approaches: The predominant approach for cloth animation is still the physics-based simulation [5, 46, 36, 35, 41]. At the back-end, it solves multiple-objectives to achieve realistic animation, which makes it computationally expensive. Furthermore, a typical PBS approach requires manual interventions, including garment designing, placing the garment over the 3D body, and fine-tune the parameters to get the desired results.

Re-Targeting Based Approaches: In order to achieve accurate realism in rendering, several approaches re-target the clothing captured from the scans [28, 34], RGBD videos [52, 53], or directly from the images [1, 7] to the human of different body shapes, keeping the body pose fixed. Another set of approaches use pose aware garment deformations [21, 24, 51]. However, one major limitation of these approaches [21, 24, 51] is they can generalize to the body poses similar to the training data.

Learning using Offline PBS data: The methods in this category leverage the ground truth data, which is generated offline by using the physics-based simulation and then learn the garment deformations/animations. Recently, several data-driven approaches have been proposed [32, 16, 47, 40] which uses PBS data. These approaches vary based on the factors considered while modeling the garment deformations. Generally, a combination of body shape, pose, and garment style has been considered (see Table 1). The methods in [15, 40, 51] use fixed garment style. Garnet [17] proposes a two-stream network to process body and garment inputs followed by a fusion strategy to predict the final 3D clothing. However, it tends to produce over-smooth results. Parametric Virtual-Try-on [47] method predicts 3D drape as a function of body shape, garment style, and material but works for a fixed body pose. SizerNet [44] learns an encoder-decoder network to predict 3D clothing as a function of human body shape and garment size parameters. However, it is limited to a human body in A-pose. DeepWrinkles [21] learns the pose dependent variations using normal map, while [18] learns it as a function of displacement map in the UV-space. Some methods [40, 51] learns the pose and shape-dependent variations using multi-layer perceptions and recurrent neural networks. All of these methods either tend to produce over-smooth results [17], or works for a fixed [44] or learned body pose [21] and garment style. To overcome these limitations, a recent method TailorNet [32] has considered the three factors of variations, i.e., body shape, pose, and garment style, in modeling the garment deformation. It decomposes the garment deformation into 1 low, and 20 high-frequency geometric components and learns them separately. Then they combine them using an RBF kernel to predict the final 3D draping. Since the TailorNet has considered the three factors of variations, its results outperform the previous approaches [40]. However, Tailornet fails to generalize beyond the range of body shape and garment style parameters used in training (see Fig. 2) and is also limited by the number of shape-style specific mixture components (20).

Method	S/D	Pose Variations	Shape Variations	Style Variations	Body Measurements
DeepWrinkles[21]	D	✓	✓	✗	✗
Santesteban[40]	D	✓	✓	✗	✗
DRAPE[15]	D	✓	✓	✗	✗
Wang[49]	S	✗	✓	✓	✗
GarNet[17]	S	✓	✓	✗	✗
Parametric[47]	S	✗	✓	✓	✗
TailorNet[32]*	S	✓	✓	✓	✗
Ours	S	✓	✓	✓	✓

Table 1: Comparison of our method with the others in the context of modeling garment deformation as a function of various body and garment variations. Our method is the first method to model the garment deformation as a function of body measurements in addition to body pose, shape and garment styles. Static/Dynamic (S/D). * denote the inference code is publicly available [33].

3. Garment Modeling and Data Specifications

Notations: We use bold capital letter $\mathbf{N}$ to represent a matrix and the respective small bold letter $\mathbf{n}$ to refer its row. We refer the i^{th} row of the matrix $\mathbf{N}$ as $\mathbf{n}_i$ and its elements as $\mathbf{n}_i = [\mathrm{n}_{i1}, \mathrm{n}_{i2}, ..., \mathrm{n}_{ij}]$. The overhat counterpart of any row or matrix (e.g., $\widehat{\mathbf{N}}$ of a matrix $\mathbf{N}$) represents the corresponding is predicted from the neural network. The $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\theta}$, and $\boldsymbol{\gamma}$ are the vectors.

This section explains the garment model and other data specifications that we use in this work. Following the work in [7, 34, 32], we consider our garment model is aligned with the SMPL [22] 3D human model. The SMPL model represents the 3D human body as a function of body pose and shape. Mathematically, the SMPL body model [22] is defined as follows:

$$SMPL(\boldsymbol{\theta}, \boldsymbol{\beta}) = \mathcal{W}(\mathcal{T}_b(\boldsymbol{\beta}, \boldsymbol{\theta}), \mathcal{J}(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathbf{W}) \quad (1)$$

$$\mathcal{T}_b(\boldsymbol{\theta}, \boldsymbol{\beta}) = \mathbf{V} + \mathcal{B}_s(\boldsymbol{\beta}) + \mathcal{B}_p(\boldsymbol{\theta}) \quad (2)$$

Where, $\mathcal{W}(\cdot)$ is a skinning function, $\mathcal{T}_b(\boldsymbol{\beta}, \boldsymbol{\theta})$ is a linear function, $\mathcal{J}(\cdot)$ is the skeleton joint prediction function and $\mathbf{W}$ is the blend weights of skeleton $\mathcal{J}(\cdot)$. The function $\mathcal{T}_b(\cdot)$ adds the pose and shape dependent deformations $\mathcal{B}_p(\cdot)$ and $\mathcal{B}_s(\cdot)$ respectively to the base template mesh vertices $\mathbf{V} \in \mathbb{R}^{n \times 3}$ in a T-pose. The final SMPL model is obtained by applying the skinning to the updated mesh vertices.

The garment mesh vertices are defined as the subset of the SMPL mesh vertices. Let $\mathbf{I}$ be an indicator matrix, whose element $i_{mn} = 1$, indicates that the m^{th} garment vertex is associated with the n^{th} SMPL body vertex . A particular garment style draped over an un-posed SMPL body $\mathcal{T}_b(\boldsymbol{\theta}, \boldsymbol{\beta})$ is encoded as the vertex offsets $\mathbf{O}$ as shown in Eq. 3. Since the garment model is aligned with the SMPL body model, for a fix $\mathbf{O}$, following others in [7, 34, 32], we also make a simplifying assumption that clothing deforms similarly to the underlying body Eq. 4.

$$\mathcal{T}_g(\boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{O}) = \mathbf{I}\, \mathcal{T}_b(\boldsymbol{\theta}, \boldsymbol{\beta}) + \mathbf{O} \quad (3)$$

$$\mathcal{G}(\boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{O}) = \mathcal{W}(\mathcal{T}_g(\boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{O}), \mathcal{J}(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathbf{W}) \quad (4)$$

Let the rows of the matrix $\mathbf{G} = \mathcal{G}(\boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{O})$ represents the vertices of the ground truth simulated garment. We use publicly available TailorNet dataset [32], and augment it with the additional ground-truth data: *standard body measurements* and normal mapped *multi-view rendered garments*.

Body Measurements: We measure the various body measurements shown in Fig. 4(c) of an SMPL 3D body model in the T-pose. These are standard body measurements that directly affect a particular garment's fitting on the body and are commonly available on various online shopping websites [8, 26].

Multi-view Rendering: We associate a texture to each garment vertex in $\mathbf{G}$. The RGB value of the texture of a garment vertex is the function of the (x, y, z) components of its unit normal vector. Let $\mathbf{T}$ be the texture matrix, where each row $\mathbf{t}_i$ represent the texture of the garment vertex $\mathbf{g}_i$. Let $\Phi(\cdot)$ represents the rendering function, it is composed of a mesh rasterizer and a shader. The function $\Phi(\cdot)$ takes the garment vertices $\mathbf{G}$, the texture $\mathbf{T}$, garment mesh faces $\mathbf{F}$, camera location $\mathbf{C}$ and the light location $\mathbf{L}$ as input. The

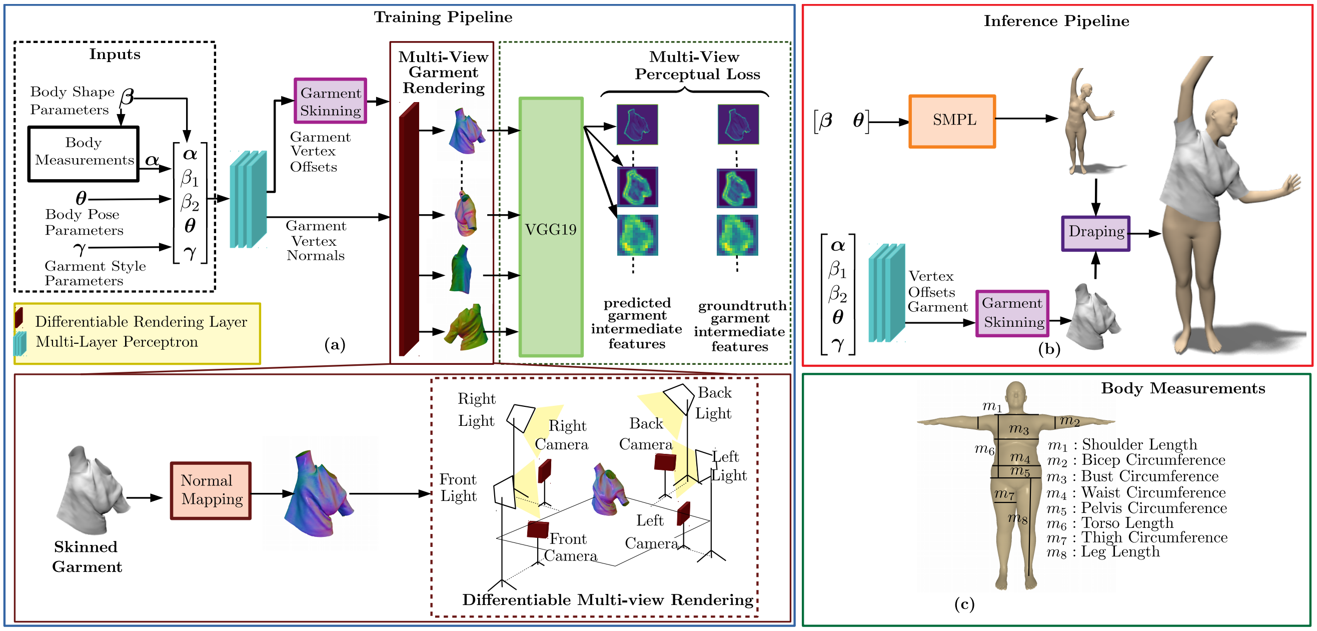


Figure 4: (a)-(b) DeepDraper Training and Inference Pipeline, Refer text in Sec. 4.. (c) Standard body measurements.

output of the function $\Phi(\cdot)$ is a rendered image. Some sample multi-view ($front$, $back$, $left$ and $right$) rendered images of a t-shirt draped over a A-posed 3D human body is shown in Fig. 3. The lights and the camera setup for multi-view rendering is shown in Fig. 4. We implement the function $\Phi(\cdot)$ using PyTorch3D renderer [37], using Phong shading and point lights. **Note:** all the components of the function $\Phi(\cdot)$ (e.g., Phong shading) are fully differentiable, therefore, we use the same function $\Phi(\cdot)$ for rendering the multi-view images of the predicted garments in the Sec. 4.1.

4. DeepDraper Network

The DeepDraper training and inference pipeline is shown in Fig. 4. *Training Pipeline:* Our method takes the SMPL PCA-shape coefficient $\boldsymbol{\beta} \in \mathbb{R}^{10}$ and compute the body measurements $\boldsymbol{\alpha}$. For the t-shirt, we consider the body measurements $\boldsymbol{\alpha} = [m_1, m_2, m_3, m_4, m_5, m_6]$ that affects the t-shirt fitting only (for pants see Sec. 5). We construct the input $\mathbf{X}$ for the DeepDraper network $\mathcal{N}(\cdot)$ by stacking the body measurements $\boldsymbol{\alpha}$, first two coefficients $[\beta_1, \beta_2]$ of the SMPL body shape, body pose $\boldsymbol{\theta}$ and the garment style coefficients $\boldsymbol{\gamma}$, $\mathbf{X} = [\boldsymbol{\alpha}, \beta_1, \beta_2, \boldsymbol{\theta}, \boldsymbol{\gamma}]$. The network predicts the garment vertices offsets $\widehat{\mathbf{O}}$ and their associated unit normals $\widehat{\mathbf{N}}$. The 3D garment $\widehat{\mathbf{G}}$ is constructed by applying the skinning function as following

$$\widehat{\mathbf{G}} = \mathcal{G}(\boldsymbol{\theta}, \boldsymbol{\beta}, \widehat{\mathbf{O}}) = \mathcal{W}(\mathcal{T}_g(\boldsymbol{\theta}, \boldsymbol{\beta}, \widehat{\mathbf{O}}), \mathcal{J}(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathbf{W}) \tag{5}$$

We assign a texture to the skinned garment $\widehat{\mathbf{G}}$ and render multi-view images. As mentioned in the Sec. 3, we assign texture to each predicted garment vertex $\widehat{\mathbf{t}}_i$ as a function of its unit normal $\widehat{\mathbf{n}}_i$. We compute the perceptual similarity loss by comparing the intermediate visual features of rendered and generated multi-view images. In addition to the perceptual loss, we additionally use geometric losses to train the DeepDraper network. *Inference Pipeline:* During inference, we pass the stacked input $\mathbf{X}$ to the DeepDraper network to get the garment vertex offsets. We obtain the garment vertices using the predicted vertex offsets. We apply garment skinning to the predicted garment and drape it over the SMPL body. We can use methods like ViBE [20], MEVA[23], SMPLify[9] to estimate the SMPL body shape and pose parameters from an RGB image. Examples under this setting are shown in Fig. 1 and 8. Next, we describe the losses we use to train the DeepDraper network.

4.1. Training Losses

Geometric Losses: The geometric losses consist of L1-loss $\mathcal{L}_{OL} = \frac{1}{\kappa}\sum_{i=1}^{\kappa} ||\widehat{\mathbf{o}}_i - \mathbf{o}_i||_1$ on predicted garment vertex offset with the ground-truth offset, and the cosine similarity loss $\mathcal{L}_{NL} = \frac{1}{\kappa}\sum_{i=1}^{\kappa}(1 - \frac{\widehat{\mathbf{n}}_i \cdot \mathbf{n}_i}{||\widehat{\mathbf{n}}_i||||\mathbf{n}_i||})$ on predicted vertex normal with the ground-truth vertex normal. Here, κ is the total number of garment vertices.

Let $\Psi(\cdot)$ be a function that takes the predicted garment mesh vertices $\widehat{\mathbf{G}} = \mathcal{G}(\boldsymbol{\theta}, \boldsymbol{\beta}, \widehat{\mathbf{O}})$ and ground-truth mesh faces $\mathbf{F}$ and output the garment mesh vertices normals $\overline{\mathbf{N}}$. Note that, the normals in $\overline{\mathbf{N}} = \Psi(\mathcal{G}(\boldsymbol{\theta}, \boldsymbol{\beta}, \widehat{\mathbf{O}}), \mathbf{F})$ are computed using the predicted garment derived from the predicted offsets, therefore it is different from the normals directly predicted by the network $\widehat{\mathbf{N}}$. We compute a regularization loss between $\overline{\mathbf{N}}$ and $\widehat{\mathbf{N}}$ as follows: $\mathcal{L}_{NReg} = \frac{1}{\kappa}\sum_{i=1}^{\kappa} ||\widehat{\mathbf{n}}_i - \overline{\mathbf{n}}_i||_2$. The regularization loss enforce the network to directly predict the vertex normals ($\widehat{\mathbf{N}}$) that are consistent with vertex normals conditioned on the predicted garment vertex offsets ($\overline{\mathbf{N}}$). An alternate approach could be to compute the loss directly on the predicted garment normals. However, in our experiments, we found our regularization choice is more effective than computing loss directly on the predicted garment. The rationale behind this choice is that vertex normals are directly related to the respective vertices (offsets). Hence directly predicting vertex normals as an auxiliary task helps the network learn vertex offsets accurately. Since the goal is to predict the offsets accurately, we predict normal directly from the network only during the training phase. Once trained, we remove the corresponding part of the network and predict only the offsets during inference.

Body-Garment Collision Loss: To ensure the predicted deformations are free from the body-garment collisions, we use a body-garment collision penalty. Specifically, for each predicted garment vertex say i^{th} vertex $\widehat{\mathbf{g}}_i$, we find the nearest 3D body vertex (say j^{th} vertex) $\mathbf{v}_j$ and its associated normal $\mathbf{n}_j$. The body-garment collision loss is computed as follows

$$\mathcal{L}_{coll} = \frac{1}{\kappa}\sum_{i=1}^{\kappa} \max(-\mathbf{n}_j(\mathbf{v}_j - \widehat{\mathbf{g}}_i), \delta) \tag{6}$$

Similar body-garment collision loss has been used in [17, 47] and found effective in reducing the majority of collisions during training.

Perceptual Losses: As mentioned in the Sec. 3, the $\Phi(\cdot)$ is a differentiable rendering function. It takes the predicted garment vertices $\widehat{\mathbf{G}}$, the texture $\widehat{\mathbf{T}}$, the garment mesh faces $\mathbf{F}$, the camera $\mathbf{C}$, the light $\mathbf{L}$ and the view as input and output the rendered image. Next, we explain the process of computing perceptual loss (PL) for the $front$ view rendered image of a deformed garment. Let $I_{\widehat{\mathbf{G}},\widehat{\mathbf{T}},f}$ denote the $front(f)$ view rendered image obtained using the rendering function $\Phi(\cdot)$ as shown in Eq. 7.

$$I_{\widehat{\mathbf{G}},\widehat{\mathbf{T}},f} = \Phi(\widehat{\mathbf{G}}, \widehat{\mathbf{T}}, \mathbf{F}, \mathbf{C}_f, \mathbf{L}_f) \tag{7}$$

Deep neural networks have been shown effective in [55] to compare the perceptual similarity between the two images. The perceptual losses have been proven successful in GAN based image synthesis process [30], and in end-to-end 3D view, synthesis [50]. Inspired by perceptual metric usage in the image synthesis works [30, 50], we use it as a loss to capture the perceptual similarity between ground-truth and predicted multi-view rendered images. We use the VGG19 [42] trained on ImageNet [14] dataset for the image classification task. We forward pass the predicted rendered images to the VGG19 network and extract the intermediate features maps (Γ) from the CNN layers ($S = [1, 3, 5, 9, 13]$). The perceptual similarity $PL(\cdot)$ between the two images is the weighted L1-loss between their intermediate feature maps.

$$PL(I_{\widehat{\mathbf{G}},\mathbf{T},f}, I_{\widehat{\mathbf{G}},\widehat{\mathbf{T}},f}) = \sum_{l\in S} \lambda_l ||\Gamma^l_{\widehat{\mathbf{G}},\mathbf{T},f} - \Gamma^l_{\widehat{\mathbf{G}},\widehat{\mathbf{T}},f}||_1 \tag{8}$$

Where, Γ^l denote the l^{th} layer feature map and λ_l is the weight of the l^{th} layer. The total perceptual loss for the $front$ view rendered image is given as

$$\ell^f = PL\big(I_{\widehat{\mathbf{G}},\mathbf{T},f},\ I_{\widehat{\mathbf{G}},\widehat{\mathbf{T}},f}\big) + PL\big(I_{\mathbf{G},\mathbf{T},f},\ I_{\mathbf{G},\widehat{\mathbf{T}},f}\big) \tag{9}$$

The perceptual loss in Eq. 9 forces the network to predict the garment vertex texture (predicted vertex normals via normal mapping) to be consistent with the ground truth vertex texture in the images space. The rasterization and the shading components of the rendering layer are differentiable. Therefore the loss in the Eq. 9 is fully differentiable. We collect the perceptual loss for multi-views i.e., $front(f)$, $back(b)$, $left(l)$, $right(r)$, $top(t)$ as $\mathcal{L}_p = \sum_{i\in\{f,b,l,r,t\}} \ell^i$. We can also compute $PL\big(I_{\widehat{\mathbf{G}},\widehat{\mathbf{T}},f},\ I_{\mathbf{G},\widehat{\mathbf{T}},f}\big)$ or $PL\big(I_{\widehat{\mathbf{G}},\mathbf{T},f},\ I_{\mathbf{G},\mathbf{T},f}\big)$, however, in our experiments, we found the addition of these losses to the Eq. 9 does not affect much but increases the computational overhead due to additional renderings. However, if there is no computational constraint, one may include them in the final loss. Note, we render both the ground-truth and the predicted garments under the same light and camera setup.

Content Losses: In addition to the perceptual loss, we also compute the image content loss as the average L1 distance between the ground-truth and respective predicted rendered multi-view images. The content loss ℓ^f_{con} for the front view rendered image is computed as in Eq. 10. We collect the total content loss in $\mathcal{L}_{con}$ as $\mathcal{L}_{con} = \sum_{i\in\{f,b,l,r,t\}} \ell^i_{con}$.

$$\ell^f_{con} = ||I_{\widehat{\mathbf{G}},\mathbf{T},f} - I_{\widehat{\mathbf{G}},\widehat{\mathbf{T}},f}||_1 + ||I_{\mathbf{G},\mathbf{T},f} - I_{\mathbf{G},\widehat{\mathbf{T}},f}||_1 \tag{10}$$

Total Loss: The total loss $\mathcal{L}_{total}$ to train the DeepDraper network is the combination of the geometric, perceptual, and content losses. The Υ in the Eq. 11 denote the weightage of the respective loss components.

$$\begin{aligned}\mathcal{L}_{total} = {} & \Upsilon_{OL}\mathcal{L}_{OL} + \Upsilon_{NL}\mathcal{L}_{NL} + \Upsilon_{NReg}\mathcal{L}_{NReg} \\ & + \Upsilon_p\mathcal{L}_p + \Upsilon_{coll}\mathcal{L}_{coll} + \Upsilon_{con}\mathcal{L}_{con}\end{aligned} \tag{11}$$

5. Experiments

In this section, we evaluate our method and compare it qualitatively and quantitatively with the closest state-of-the-art method TailorNet [32]. Since the inference code and the trained model of TailorNet are publicly available at [33], we train our DeepDraper method on the TailorNet dataset[33] instead of Sizer [44] or CLOTH3D [6] datasets for the fair comparisons. Similar to TailorNet, we post-process the output to remove the garment intersections with the body. In what follows, we compare DeepDraper with the TailorNet.

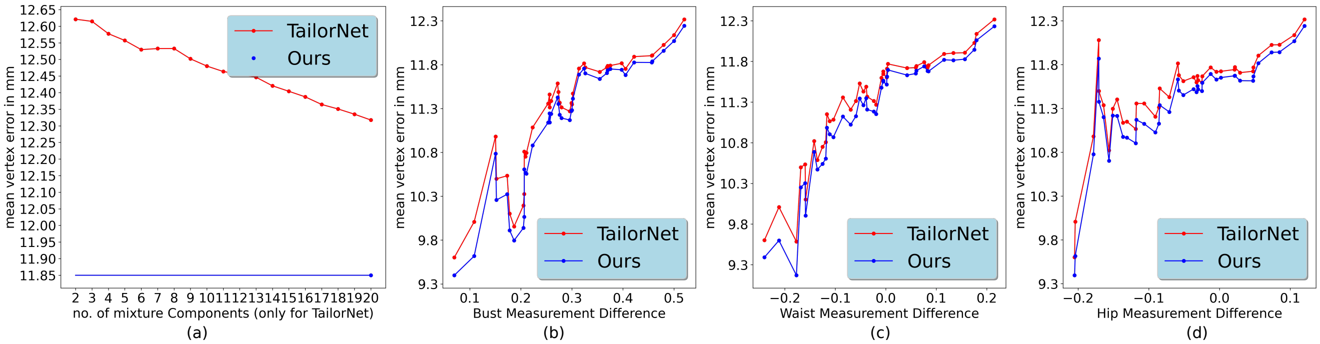


Figure 5: (a). TailorNet [32] is sensitive to the number of mixture components, refer text for detail. (b-d) Garment fitment analysis, refer fitment analysis section for the details.

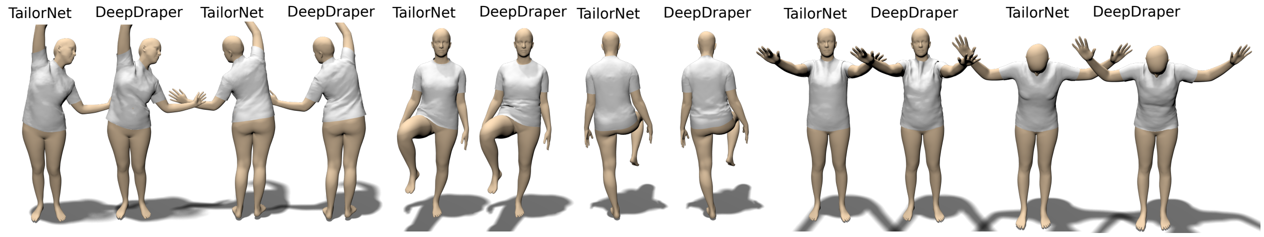


Figure 6: Qualitative comparison of DeepDraper with TailorNet on a new garment style $\gamma = [2.75, 2.6, 0.0, 0.0]$. DeepDraper generalize well on new garment styles and predicts accurate folds and wrinkles compared to the TailorNet.

5.1. Single vs Mixture Model

TailorNet's performance depends on the shape-style specific high-frequency components. In Fig.5(a), we show reducing the number of mixture components increases its per-vertex error. Furthermore, to accurately predict the garment deformation for the new garment style, TailorNet would require training new shape-style specific components, which will increase the size and inference time of the TailorNet. In contrast to TailorNet, DeepDraper does not depend on any style or shape-specific components, hence its performance is consistent, and generalizes well beyond the unseen garment styles, see Fig. 6.

5.2. Fitment Analysis

We compare our approach with the TailorNet in the context of garment fitting. Similar to the body measurements shown in Fig. 4(c), we compute the ground truth garments' bust, waist, and hip circumference measurements. The result in Fig. 5(b-d) shows the mean per-vertex error against the difference in the respective measurements of the garment and the body. The negative value on the x-axis indicates the garment is tight (e.g., garment measurement on the bust is smaller than the bust measurement of the naked body) at the respective body parts (bust, waist, or hip). Our method's mean per-vertex error is consistently low compared to TailorNet for both loose and tight-fitting clothes. Furthermore, DeepDraper predicts garment deformations consistent with the varying body height and overall body fatness, see Fig. 7.

5.3. Performance Metrics

DeepDraper takes $\sim 10\times$ lesser memory space and run $\sim 23\times$ faster on GPU and $\sim 11\times$ faster on CPU, than the TailorNet [32]. We evaluate the run-time of both ours and TailorNet on a laptop with Intel i7 CPU and Nvidia GeForce RTX 2070 GPU. We want to highlight the reported time, i.e., 1-2 ms in the TailorNet paper, was the run-time of a batch of 21 samples [33][1]. DeepDraper further reduce the mean per-vertex error (in mm) by ($\sim 4\%$) to **11.85**, compared to the TailorNet **12.32**.

5.4. Generalization

To demonstrate the generalization ability of DeepDraper methods to other garments, we train it for female pants from the TailorNet dataset [33]. The body measurements $\boldsymbol{\alpha}$ that we considered are $\boldsymbol{\alpha} = [m_4, m_5, m_7, m_8]$. Only these measurements affect the deformation of the pants. All the settings remain the same as it was in the t-shirt garment. DeepDraper reduces the mean per-vertex error to **4.2** ($\sim 12\%$), compared to the TailorNet **4.8**. We further compare the gen-

[1] This is clarified by the author in their official code [33] repository and confirmed over a personal communication with the authors

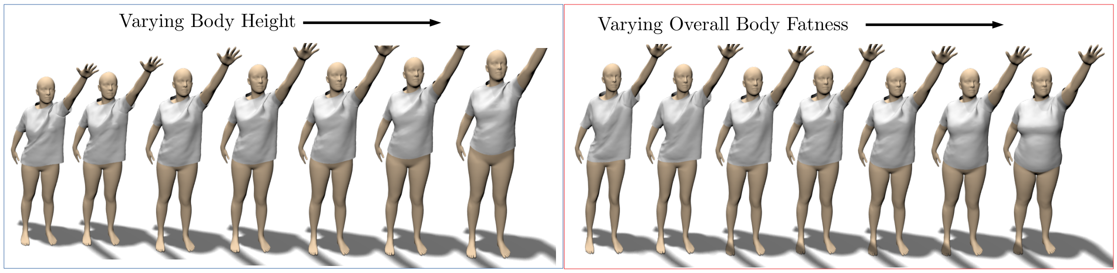


Figure 7: DeepDraper predicts the accurate wrinkles, folds and overall fitment with respect to the body height and fatness.

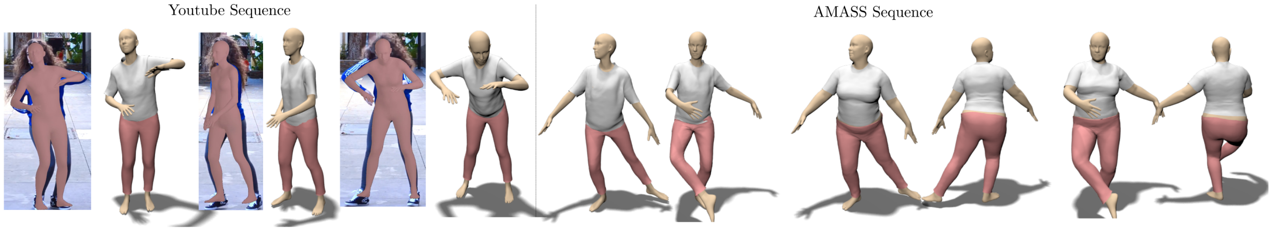


Figure 8: DeepDraper's results on a Youtube video sequence, and on unseen pose sequence from the AMASS[25] dataset.

eralization of the TailorNet and DeepDraper on unseen garment styles in Fig. 6. Also, DeepDraper generalizes well on the unseen pose sequences and both unseen pose and body shape. In Fig.8 we show qualitative results on unseen human body shape and poses. These visually consistent results for unseen body shape, pose, and garment styles of multiple garments like t-shirt and pants show the effectiveness of the DeepDraper compared to the TailorNet.

5.5. Ablation Study

We study the effect of different loss components in Eq. 11 in the fitting and rendering of garments. Table 2 shows the result of the ablation study. The usage of both the geometric and perceptual constraints and the conditioning on the body measurements yield the best results.

Losses (refer Eq. 11)	Mean per vertex error
$\mathcal{L}_{OL}$	12.55
$\mathcal{L}_{OL} + \mathcal{L}_{NL}$	12.38
$\mathcal{L}_{OL} + \mathcal{L}_{NL} + \mathcal{L}_{NReg}$	12.21
$\mathcal{L}_{OL} + \mathcal{L}_{NL} + \mathcal{L}_{NReg} + \mathcal{L}_p$ + (w/o body mmts)	12.10
$\mathcal{L}_{OL} + \mathcal{L}_{NL} + \mathcal{L}_{NReg} + \mathcal{L}_p$ + (w body mmts)	**11.85**

Table 2: Ablation Study

5.6. Implementation Details

We have implemented our pipeline using PyTorch [31], and PyTorch3D [37], and train the DeepDraper network with batch size 32 and use Adam optimizer[19]. Our MLP has three hidden layers. We use a pre-trained VGG19 network, trained for an image classification task on ImageNet dataset [14] and freeze its weights. The weights of the VGG19 feature maps for computing the perceptual loss in Eq. 9 are as follows $[\lambda_1, \lambda_3, \lambda_5, \lambda_9, \lambda_{13}] = [1.0/32, 1.0/16, 1.0/8, 1.0/4, 1.0]$. These values are similar to those used in image synthesis works [30, 50]. We set $\delta = 1e - 4$ in Eq. 6. We use the PyTorch3D [37] differentiable renderer and set the rasterizer image size to 64×64, blur radius to 0.0 and the faces per pixel to 1. We use the soft Phong shader with point lights. We set the initial learning rate to $1e-4$, and reduce it by a factor of 0.1 after every 100 epochs up-to $1e - 7$. We train the DeepDraper model for a total of 500 epochs. We set the weightage of the loss component $\Upsilon_{OL} = 1e3$, and of the remaining components in the Eq. 11 to 1. Refer supplementary for other details.

6. Conclusion and Future Directions

We have presented a novel strategy for draping 3D garments over a 3D human body using a single unified garment deformation model that learns the shared space of variations in body shape, pose, and garment style, yielding realistic rendering in terms of wrinkles and folds on the output garment. Unlike the existing methods we use standard body measurement to produces better fitment. Our method generalizes well for unseen garment style, unseen human pose sequences, and significantly improved final draping of garments compared to state-of-the-art methods. Moreover, our method is $\sim 10\times$ *smaller* and $\sim 23\times$ *faster* than the closet existing method TailorNet.
Limitations: Our method learns one garment at a time and does not consider fabric properties. Learning a single model for multiple garments with fabric properties will be our future research direction.

References

[1] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single rgb camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1175–1186, 2019. 3

[2] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Detailed human avatars from monocular video. In *2018 International Conference on 3D Vision (3DV)*, pages 98–109. IEEE, 2018. 3

[3] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8387–8397, 2018. 3

[4] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005. 3

[5] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54, 1998. 3

[6] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Cloth3d: Clothed 3d humans. In *European Conference on Computer Vision*, pages 344–359. Springer, 2020. 6

[7] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5420–5430, 2019. 3, 4

[8] Birdsnest. Birdsnest: Online shopping portal for women, 2021. 4

[9] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European conference on computer vision*, pages 561–578. Springer, 2016. 5

[10] Dan Casas, Marco Volino, John Collomosse, and Adrian Hilton. 4d video textures for interactive character appearance. In *Computer Graphics Forum*, volume 33, pages 371–380. Wiley Online Library, 2014. 3

[11] Enric Corona, Albert Pumarola, Guillem Alenyà, Gerard Pons-Moll, and Francesc Moreno-Noguer. Smplicit: Topology-aware generative model for clothed people. *arXiv preprint arXiv:2103.06871*, 2021. 3

[12] Edilson De Aguiar, Leonid Sigal, Adrien Treuille, and Jessica K Hodgins. Stable spaces for real-time clothing. *ACM Transactions on Graphics (TOG)*, 29(4):1–9, 2010. 2

[13] Edilson De Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. In *ACM SIGGRAPH 2008 papers*, pages 1–10. 2008. 3

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6, 8

[15] Peng Guan, Loretta Reiss, David A Hirshberg, Alexander Weiss, and Michael J Black. Drape: Dressing any person. *ACM Transactions on Graphics (TOG)*, 31(4):1–10, 2012. 2, 3, 4

[16] Erhan Gundogdu, Victor Constantin, Shaifali Parashar, Amrollah Seifoddini Banadkooki, Minh Dang, Mathieu Salzmann, and Pascal Fua. Garnet++: Improving fast and accurate static 3d cloth draping by curvature loss. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 3

[17] Erhan Gundogdu, Victor Constantin, Amrollah Seifoddini, Minh Dang, Mathieu Salzmann, and Pascal Fua. Garnet: A two-stream network for fast and accurate 3d cloth draping. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8739–8748, 2019. 2, 3, 4, 6

[18] Ning Jin, Yilin Zhu, Zhenglin Geng, and Ronald Fedkiw. A pixel-based framework for data-driven clothing. In *Computer Graphics Forum*, volume 39, pages 135–144. Wiley Online Library, 2020. 4

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. 8

[20] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5253–5263, 2020. 1, 5

[21] Zorah Lahner, Daniel Cremers, and Tony Tung. Deepwrinkles: Accurate and realistic clothing modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 667–684, 2018. 2, 3, 4

[22] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 2, 3, 4

[23] Zhengyi Luo, S Alireza Golestaneh, and Kris M Kitani. 3d human motion estimation via motion compression and refinement. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 5

[24] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. Learning to dress 3d people in generative clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6469–6478, 2020. 1, 3

[25] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5442–5451, 2019. 1, 8

[26] Myntra. Myntra: Online shopping portal, 2021. 4

[27] Ryota Natsume, Shunsuke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeo Morishima. Siclope: Silhouette-based clothed people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4480–4490, 2019. 3

[28] Alexandros Neophytou and Adrian Hilton. A layered model of human body and garment deformation. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 171–178. IEEE, 2014. 3

[29] Rose Otieno, Gina Pisut, and Lenda Jo Connell. Fit preferences of female consumers in the usa. *Journal of Fashion Marketing and Management: An International Journal*, 2007. 1

[30] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 2, 6, 8

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 8

[32] Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7365–7375, 2020. 1, 2, 3, 4, 6, 7

[33] Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. Tailornet Code, Jan. 2021. 4, 6, 7

[34] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J Black. Clothcap: Seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics (TOG)*, 36(4):1–15, 2017. 3, 4

[35] Xavier Provot. Collision and self-collision handling in cloth model dedicated to design garments. In *Computer Animation and Simulation'97*, pages 177–189. Springer, 1997. 2, 3

[36] Xavier Provot et al. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface*, pages 147–147. Canadian Information Processing Society, 1995. 2, 3

[37] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 5, 8

[38] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2304–2314, 2019. 3

[39] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 84–93, 2020. 3

[40] Igor Santesteban, Miguel A Otaduy, and Dan Casas. Learning-based animation of clothing for virtual try-on. In *Computer Graphics Forum*, volume 38, pages 355–366. Wiley Online Library, 2019. 2, 3, 4

[41] Andrew Selle, Jonathan Su, Geoffrey Irving, and Ronald Fedkiw. Robust high-resolution cloth using parallelism, history-based collisions, and accurate friction. *IEEE transactions on visualization and computer graphics*, 15(2):339–350, 2008. 3

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

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

[44] Garvita Tiwari, Bharat Lal Bhatnagar, Tony Tung, and Gerard Pons-Moll. Sizer: A dataset and model for parsing 3d clothing and learning size sensitive 3d clothing. In *16th European Conference on Computer Vision*. Springer, 2020. 4, 6

[45] Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 20–36, 2018. 3

[46] Tzvetomir Vassilev, Bernhard Spanlang, and Yiorgos Chrysanthou. Fast cloth animation on walking avatars. In *Computer Graphics Forum*, volume 20, pages 260–267. Wiley Online Library, 2001. 3

[47] Raquel Vidaurre, Igor Santesteban, Elena Garces, and Dan Casas. Fully convolutional graph neural networks for parametric virtual try-on. In *Computer Graphics Forum*, volume 39, pages 145–156. Wiley Online Library, 2020. 2, 3, 4, 6

[48] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH 2008 papers*, pages 1–9. 2008. 3

[49] Tuanfeng Y Wang, Duygu Ceylan, Jovan Popović, and Niloy J Mitra. Learning a shared shape space for multimodal garment design. *ACM Transactions on Graphics (TOG)*, 37(6):203, 2019. 2, 4

[50] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7467–7477, 2020. 2, 6, 8

[51] Jinlong Yang, Jean-Sébastien Franco, Franck Hétroy-Wheeler, and Stefanie Wuhrer. Analyzing clothing layer deformation statistics of 3d human motions. In *Proceedings of the European conference on computer vision (ECCV)*, pages 237–253, 2018. 3, 4

[52] Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7287–7296, 2018. 3

[53] Tao Yu, Zerong Zheng, Yuan Zhong, Jianhui Zhao, Qionghai Dai, Gerard Pons-Moll, and Yebin Liu. Simulcap: Single-view human performance capture with cloth simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5504–5514, 2019. 3

[54] Cyril Zeller. Cloth simulation on the gpu. In *ACM SIGGRAPH 2005 Sketches*, pages 39–es. 2005. 2

[55] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[56] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. Deephuman: 3d human reconstruction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7739–7749, 2019. 3

[57] Hao Zhu, Xinxin Zuo, Sen Wang, Xun Cao, and Ruigang Yang. Detailed human shape estimation from a single image by hierarchical mesh deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4491–4500, 2019. 3