EUROGRAPHICS 2017 / L. Barthe and B. Benes
(Guest Editors)

Volume 36 (2017), Number 2

DeepGarment : 3D Garment Shape Estimation from a Single Image

R. Daněřek[1,2,⋆], E. Dibra[1,2,⋆], C. Öztireli[1], R. Ziegler[2], M. Gross[1]

[1] Computer Graphics Laboratory, ETH Zürich
[2] Vizrt Switzerland
[⋆] These authors contributed equally

Figure 1: *Garment 3D shape estimation using our CNN model and a single-view. From left to right: real-life images capturing a person wearing a T-shirt, segmented and cut-out garments and 3D estimations of the shape.*

Abstract

3D garment capture is an important component for various applications such as free-view point video, virtual avatars, online shopping, and virtual cloth fitting. Due to the complexity of the deformations, capturing 3D garment shapes requires controlled and specialized setups. A viable alternative is image-based garment capture. Capturing 3D garment shapes from a single image, however, is a challenging problem and the current solutions come with assumptions on the lighting, camera calibration, complexity of human or mannequin poses considered, and more importantly a stable physical state for the garment and the underlying human body. In addition, most of the works require manual interaction and exhibit high run-times. We propose a new technique that overcomes these limitations, making garment shape estimation from an image a practical approach for dynamic garment capture. Starting from synthetic garment shape data generated through physically based simulations from various human bodies in complex poses obtained through Mocap sequences, and rendered under varying camera positions and lighting conditions, our novel method learns a mapping from rendered garment images to the underlying 3D garment model. This is achieved by training Convolutional Neural Networks (CNN-s) to estimate 3D vertex displacements from a template mesh with a specialized loss function. We illustrate that this technique is able to recover the global shape of dynamic 3D garments from a single image under varying factors such as challenging human poses, self occlusions, various camera poses and lighting conditions, at interactive rates. Improvement is shown if more than one view is integrated. Additionally, we show applications of our method to videos.

1. Introduction

Clothing is an important part of virtual human modeling. Capturing and modeling garments are fundamental steps for many applications ranging from online retail to virtual character and avatar creation. There exist many options to model or capture garments with professional tools used by talented artists, digitalized traditional garment sewing patterns, 3D meshes and expensive physically based simulations, or 3D capture with advanced hardware in controlled setups. However, such tools and setups are not available to most content generators and users. Instead, a practical approach is allowing the users to utilize commodity cameras and capture clothing from a single image or video. Such simple and practical capture systems have recently been developed for human faces [CBZB15], hair [HMLL15], eyes [BBGB16], body shapes [DOZG16, DJO*16], or hands [TPT16]. Cloth capturing from dynamic scenes with monocular imagery remains a challenge due to the complex deformations.

A successful approach to solve such ill-posed capture problems is utilizing data-driven priors. With the recent advances in machine learning techniques, it has been demonstrated that accurate reconstructions of various types and classes of objects can be obtained even from a single image [WSK*15, TDB16, DJO*16]. This requires constructing a database that covers the subspace of possible data points while staying practical in terms of its size, and a careful modeling of the input/output spaces and the associated learning algorithm.

In this paper, we present a data-driven technique that can recover the 3D shape of a garment from a single image by utilizing a database of purely synthesized clothing. We construct our database by simulating garments on virtual characters for different poses and shapes, and under different lighting conditions and camera views. We then propose a convolutional neural network based architecture that is trained with our data set. The key idea is to learn the deformation (simply represented as the vertex displacement) from a reference mesh (either a garment mesh or a body mesh, depending on the application) with respect to image observations using a CNN. As the data contains physically simulated garments, our technique is able to capture dynamic clothes in motion for various scene conditions and poses. Our goal is to obtain the correct global 3D shape, possibly with plausible high-frequency deformations (such as wrinkles and folds), ready to be used in applications such as virtual avatars. This is a challenging problem for a single view due to the occlusions and loss of 3D information in real images. We illustrate that even very challenging cases can be handled with the proposed technique with garment specific data-driven priors.

In summary, we have the following main contributions in this paper:

- An end-to-end 3D garment shape estimation algorithm. The algorithm automatically extracts 3D shape from a single image captured with an uncontrolled setup that depicts a dynamic state of a garment at interactive rates.
- A regressor based on convolutional neural networks (CNN-s) combined with statistical priors and a specialized loss function for garment shape estimation. We further present experiments with several architectures including those for single and multi-view setups.

2. Related Work

Following the growing interest in online apparel shopping, virtual reality, and virtual cloth fitting for avatar creation, a wide variety of approaches have been presented that tackle the problem of 3D cloth estimation and modeling. With respect to the input expected, they could be divided into pose-based [HTC*14], pose and shape based [GRH*12], single RGB image based [ZCF*13], [YAP*16], single silhouette based [JHK15], multiple RGB images based [PZB*09] or RGB and Depth image based [CZL*15]. In terms of the estimation techniques utilized, the methods can be classified as follows. Some of them are based on optimization routines that deform a cloth model to fit to image-space information (e.g. contour [YAP*16]), others find a mapping to cloth panels or measurements that in turn are used to reconstruct the meshes with Physically Based Simulations (PBS) [JHK15], or directly find a mapping to 3D shape or shape deformations [GRH*12].

Our method takes a single RGB image as the input, and estimates 3D vertex deformations. The current single image based methods come with various limitations such as the need for manual interaction and assumptions on the camera poses and lighting conditions [YAP*16, JHK15, ZCF*13], restriction on the complexity of human poses [JHK15, ZCF*13], symmetry assumptions for the back of the cloth [ZCF*13], inability to handle self occlusions [ZCF*13], high run-time [BPS*08, YAP*16, ZCF*13], and the assumption of a statically stable physical state on the cloth and underlying human body [YAP*16, JHK15] that prohibits the estimation of clothes under dynamic scenes. Our method aims at overcoming these limitations, making single-image 3D garment capture practical.

The cloth shape estimation techniques can be further split into several categories based on the general approach utilized as follows.

Structure-from-motion-based techniques modify and extend the standard SfM setup to estimate the shape of the garment. Some of the techniques rely on special markers depicted on the garment to make the process easier, with early work focusing on reconstruction of just single sheets of cloth or smaller pieces of garments from single images [PH03, SM04, GKB03]. The first work [SSK*05] to solve the reconstruction problem for the entire garment assumes special markers that are easily detectable and localizable in 3D via a standard multi-view camera setup. White et al. [WCF07] optimized the quality of the results further with a smarter marker selection and a new hole filling strategy producing high quality results with a speed of several minutes per frame. Bradley et al. [BPS*08] utilized anchor points set to special garment locations that can be easily detected (e.g. sleeves or the neckline) in a controlled setup, eliminating the need of special markers. In a follow up work [PZB*09], the final garment shape is further augmented utilizing edge detection to deform areas using a handcrafted non-rigid local transformation that can reconstruct higher frequency plausible wrinkles. Unlike these works, we target a single image-based setting with a minimally controlled setup.

Shape-from-shading-based techniques Zhou et al. [ZCF*13] propose garment modeling from a single image, utilizing statistical human body models and having the user outline the silhouette of the garment and set the body pose. The initial shape is then estimated by constructing oriented facets for each bone [RMSC11], and assuming symmetry in order to model the garment from the back as well. Then, shape-from-shading is used to recover higher frequency folds, achieving comparable results to White et al. [WCF07], however with considerable user interaction, run-time in order of minutes, and the inability to handle self occlusions of the character.

Data-driven techniques Most data-driven works have focused on estimating the naked human body shape from images, mainly utilizing statistical human body shape priors learned from 3D naked human body scans with techniques similar to SCAPE [ASK*05]. Balan et al. [BB08] utilized such a model to infer human body under clothing with a multi-camera setup. Other works estimate

the human body from a single image by mapping silhouette features with random forests [SBB07], regression forest with canonical correlation analysis [DOZG16], or Gaussian process latent variables [CKC10].

Earlier data-driven techniques estimate 2D garment shapes based on computed 2D silhouette cloth descriptors and difference from naked body silhouettes [GFB10]. Applying this idea to 3D cloth modeling, a generative model (DRAPE [GRH*12]) was proposed that allows to dress any person in a given shape and pose, by learning a linear mapping from SCAPE [ASK*05] parameters to DRAPE cloth parameters. A similar approach was taken by Neophytou et al. [AN14], utilizing a more powerful technique for modeling human shapes, and a clothing model that is treated as a transformation factor from a reference body shape to the final clothed shape. This is in contrast to DRAPE that learns separate models for every type of clothing. Hahn et al. [HTC*14] take a different approach, where instead of modeling clothing as an 'offset' from the naked body shape, they approximate physically based cloth simulators by clustering in the pose space and performing PCA on the simulated garment per cluster to reduce the complexity and speed up the process. Similarly, we model a garment as a deformation from a body or from a template garment shape. However, unlike these methods, we tackle the problem of 3D garment estimation from images.

Other works aim at estimating 3D garment shape from a single image. Some of these methods assume depth is known [SSP*14, CZL*15], while others work for restricted mannequin poses and given cloth panels [JHK15], or assume considerable manual interaction and statically stable physical state of the garment and the underlying human body to map wrinkle patterns and segmented garments to cloth parameters and materials through fitting, taking several hours to compute [YAP*16]. In contrast, our method can estimate dynamic garment shapes that are not in steady-state, minimizes user interaction, and runs at interactive rates.

Deep learning In recent years, there has been a massive uptake of deep learning in all of the applied machine learning fields thanks to advances in parallel computing on GPUs and the concept of stacking multiple layers to learn richer representations of data. CNN-s have been proven to outperform state-of-the art techniques in computer vision applications such as image classification [KSH12, SLJ*15, SZ14], feature detection (Overfeat [SEZ*13]) and description (LIFT [YTLF16]), optical flow estimation [FDI*15], 2D pose estimation [TS13], denoising, segmentation etc. Since the creation of AlexNet [KSH12], deeper architectures have been developed, such as the Deep Residual Net [HZRS15] which introduced the "shortcut connections" to achieve state-of-the-art, along with smaller architectures like SqueezeNet [IMA*16] that achieves AlexNet performance utilizing 50× less parameters. While there has been recent works on 2D cloth recognition and retrieval [LLQ*16], or 3D shape classification, retrieval and representation such as [SMKL15], [WSK*15] and [WKL15] targeted to rigid 3D objects, except for [DJO*16] that infer 3D human body shape from silhouettes, to the best of our knowledge, there has been no previous work that attempts to infer 3D garment shape from images, as in here.

3. Data Generation

We require to have a database of pairs of renderings and the corresponding 3D garment shapes.

Unfortunately there exist no such datasets, mostly due to the difficulty of capturing garments under various settings. Hence, a very important step in our technique is synthesizing data that capture garment deformations. Fig.2 shows our data generation pipeline consisting of the following steps:

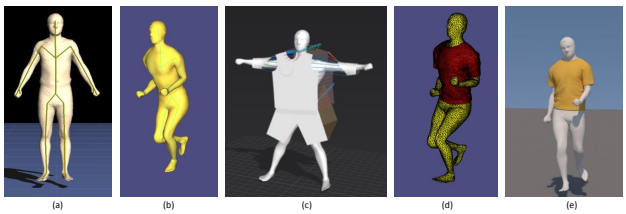

Figure 2: *Data generation pipeline. A human model is automatically rigged (a) and animated (b). Then a desired garment is designed (c), simulated under some motion (d), and realistically rendered (e). A mask of the rendered garment is used as a training input.*

Human Model Creation and Animation The first step in obtaining an accurate garment geometry is to create a dataset of naked human meshes and animate them in a natural manner. We picked 10 meshes, from a dataset of 1500 male meshes [PWH*15] (Figure 2 (a)), generated by a statistical human shape model [ASK*05], covering major variations in body types. For animation we utilize varying motions such as walking, running, jumping, turning, dancing, and boxing sequences, represented as 3D skeletons and extracted from an available motion capture dataset [Cmu], adding up to 20 minutes of motions. We attach the skeletons to the human shapes with an automatic method [BP07] that computes skinning weights (Figure 2 (a)), by augmenting its implementation with our motion capture skeleton. Each motion pose is then represented as a transformation relative to a T-pose, scaled to the size of the corresponding auto-rigged bones and mesh. The meshes are animated applying Dual Quaternion Skinning [KCvO08], as in Figure 2 (b).

Garment Design In order to design the clothing and then dress the character with it, we use Marvelous Designer [MD], a commercial software that allows to select clothing type, material properties, and set tightness of the cloth onto a normalized body posture (T-pose), as in Fig.2 (c). This is a tedious manual process, and without loss of generality we design men's t-shirts, as well as a woman's dress, representing semi-tight and loose clothings.

Garment Simulation We animate the characters dressed in the designed garments with the motion capture dataset and simulate cloth material behavior utilizing ARCSim [NSO12, NPO13] for physically based simulation. This software has the advantage of cluster deployment, due to the extensive use of OpenMP, which benefits our data generation process. After extending it to support non-rigid object animations and deformations, we run our simulations at 30 FPS, which results in approximately 15000 shapes per character and per garment (with the resolution of 6500 vertices, and 12800 triangles, Figure 2 (d)). In order to align the generated

meshes, we remove the global translation and rotation, as computed by the translation and rotation element of the root joint from the articulated skeleton of the corresponding animation frame.

Rendering In order to realistically render the simulated geometry accounting for phenomena such as soft shadows, global illumination, or realistic light sources, we utilize Mitsuba [Jak10], which is also easily deployable to a cluster. We create a scene with a simple planar diffuse surface serving as the ground with a gray albedo as the scene. The garment material is set to be diffuse due to Lambertian assumptions and the color is randomly sampled. The whole scene is lit by a realistic sun-sky emitter varying its position, to approximate natural lighting as accurately as possible. The camera pose is also varied to capture view-point changes. We show an example rendering in Fig.2 (e). We also render a mask, which is then cropped with a padding of 5 pixels around the boundaries of the masked area while setting the non-garment pixels to zero. Then, the image is downscaled to the size of 64x64 pixels, as in Figure 3. In our work we utilize single and two-view models. Therefore, we render views from the front and back. The camera is placed on the normal direction of the pelvis, with a variation of +-30 degrees, as also shown in Figures 3, 7, 8. Hence, at test time, the system can cope with large variations in view and pose.

4. 3D Garment Shape Estimation

Our method aims at estimating 3D garment shape or shape deformations from a single image capturing dynamic motion. Below we explain each step and the neural network architectures we developed to tackle this problem. Given an input image, our system masks the garment, and feeds it as an input to a specialized CNN, trained end-to-end to regress to 3D garment vertex deformations or offsets from a template mesh or human body. The method accurately captures global (low-frequency) deformations and for data similar to the training set it is even capable of recovering high frequency details. A better recovery of higher frequency details (such as wrinkles or folds) can be further enforced using a specialized loss layer computed over vertex positions and normals simultaneously. As a final step, in order to avoid interpenetration between the estimated garment and the body, we minimize an energy term on the vertex displacements.

4.1. Preprocessing

As our system is trained to regress with CNN-s, including background information would add noise, and in order not to bias the regressor towards backgrounds, one would have to generate a variety of them for the same training sequences, increasing the training time and data space drastically. Hence, we assume to have a mask for segmenting out the garment as input to our technique.

An accurate segmentation can be obtained by assuming a background of uniform color, utilizing Gaussian Mixture Models to learn a background model, and finally segmenting with graph-cuts [BFL06]. We could also apply learning based techniques like state-of-the-art background subtraction CNN-s [BKC15], or cloth specific segmentation methods [YHKB13] similar to previous works [YAP*16]. The masks used for segmentation can also be propagated if a video is used as the input. When assuming a background of uniform color, we use an automatic method as elaborated above [BFL06]. Otherwise, we have an interactive segmentation pipeline based on scribbles as illustrated in Figure 4.

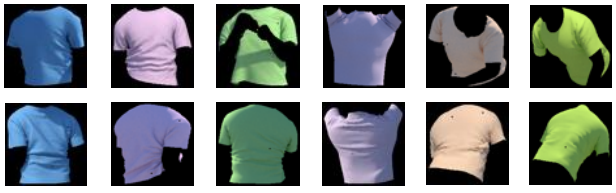

Figure 3: *Samples of masked and downscaled renderings of a garment for a front (top) and back (bottom) view.*

4.2. Mesh Deformation Representation

We have two different representations for the garment deformation. Both are based on the idea of vertex displacements, with different reference meshes. We will see in the next sections that each representation has its own advantages for different applications of our technique.

Garment-from-Garment Shape Representation The output of our method is a 3D mesh represented as follows : Let $\mathcal{S}_{ref} = (\mathbf{V}_{ref}, \mathbf{F}_{ref})$ be a reference garment mesh which is dressed on a character in a T-pose, with $\mathbf{V} \in \mathbb{R}^{n\times 3}$ as the matrix storing the 3D coordinates of each vertex in each row, and $\mathbf{F}_{i,j} \in \{1,\dots,n\}$ containing vertex indices, with each row defining one triangle of the mesh. We encode deformations of $\mathcal{S}_{ref}$ with difference vectors from the reference vertices. We thus encode a mesh $\mathcal{S}_k$ that was created by deforming $\mathcal{S}_{ref}$ with the matrix $\mathbf{V}'_k = \mathbf{V}_k - \mathbf{V}_{ref}$. In order to organize the dataset more conveniently, $\mathbf{V}'_k$ is flattened into $\mathbf{v}'_k$, where $\mathbf{v}'_k \in \mathbb{R}^{3n}$, and these vectors for all meshes in the database are then stacked into a large matrix $\mathbf{Y} \in \mathbb{R}^{M\times 3n}$, where M is the number of deformed shapes (or samples) in the database. Not all of the $3n$ degrees of freedom are necessary to represent our shape deformation set. We compress the matrix $\mathbf{Y}$ by performing a principal component analysis (PCA) to get $\mathbf{U} \in \mathbb{R}^{(N\times l)}, l << 3n$, where $\mathbf{U} = PCA_l(\mathbf{Y})$, and $l = 1000$, still achieving almost perfect reconstruction while reducing the dimensionality by a factor of 20. We thus set $\mathbf{Y} = \mathbf{U}$ for this case.

Garment-from-Body Shape Representation Depending on the intended application, an alternative representation can be opted.

Figure 4: *Video segmentation pipeline with our software. From left to right : First video frame, foreground and background scribbles, segmentation result on the frame, segmentation automatically propagated to another frame.*

The above representation does not guarantee that any estimated garment shape will fit the body mesh of our choice. Hence, if the intended application is to dress human meshes, we can represent the garment mesh as an offset from a body. For a given pose, we thus first associate each vertex of the garment mesh to its closest vertex on the body mesh, and compute the difference between those to get $\mathbf{V}'_k$ as above. The advantage of this alternative is that one can vary the body mesh and the garment dressed on that body will vary accordingly, avoiding major interpenetration artifacts. The downside is that without a specific body shape, the garment shape cannot be reconstructed. Hence the selection of the representation depends on the choice of the application, which is either garment shape estimation or body dressing estimation.

4.3. Single-View Architecture

We formulate the 3D garment shape estimation as an end-to-end regression performed by a Convolutional Neural Network (CNN), from an image depicting a person wearing the garment to the 3D shape of the garment. The network learns a representation space of image features that are able to encode the visual appearance of the possibly wrinkled clothing pattern.

Our dataset can be described with the input and output pairs $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$. Let $\mathcal{X}$ be the set of our observations and $x_i \in \mathcal{X}$ a sample from our observation set, where $i \in \{1 \dots N\}$, and N is the number of samples in the dataset. The input x_i may consist of one or more images corresponding to different views. In our experiments, we only use one or two camera views, specifically frontal view and/or back view. Our images are masked and resized to 64×64 pixels hence the full input becomes $64 \times 64 \times 3$ dimensional. The $y_i \in \mathcal{Y}$ is the ground truth output data, which is obtained with either of the 3D garment mesh representations as described above, corresponding to the observed input x_i. The $\mathcal{Y}$ can be either the PCA-reduced output denoted as $\mathcal{Y}_{PCA}$, or the full-space dataset denoted as $\mathcal{Y}_{full}$ (Section 4.2).

The regression can then be written as the map $y = CNN(x)$, where CNN is the convolutional neural network model we consider. We have experimented with various CNN architectures including the most recent and advanced ones explained below and in the supplementary, whose macro-architectural pattern is inspired by Alexnet [KSH12]. The input of the network is a 64x64-shaped RGB images. The convolutional part of the networks can either contain a sequence of simple convolutional layers or other more advanced convolutional architectural patterns. The convolutional part is followed by a flattening layer, after which one or more dense layers are employed. The final dense layer yields the output of the regression. The activation function we use is always the rectified linear unit (ReLU). To avoid overfitting, dropout is added after the convolutional part of the net. We have considered the following architecture:

SqueezeNet, introduced by Iandola et al. [IMA*16], achieves AlexNet performance but needs significantly less parameters and therefore is much faster to train, thanks to its novel "Fire" Layers. The benefit of this architecture is its short training time while maintaining a high degree of quality, which makes it a great candidate for heavy experimentation.

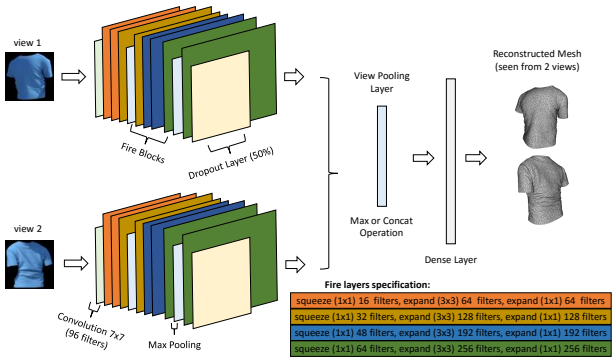

Figure 5: *Our SqueezeNet incarnation for the two-view case. The single-view is similar, except that only one convolutional block is utilized and there is no view-pooling layer. The input is one or two images of a masked garment, and the output is the garment mesh. For a very detailed description of the networks and a further discussion about the architecture please see the supplementary material.*

We base our network architecture on SqueezeNet, and adapt it to our problem. Figure 5 demonstrates the two view architecture described in Section 4.5. For the single view case, the network consists of only one convolutional block and no view-pooling layer.

4.4. Loss Layer

The choice of the loss function plays an important role in the training of neural networks. The ideal way to measure the performance of our neural net model would be to first reconstruct the garment based on the network output and then use it to render an image with the same configuration as the input image. A pixelwise distance between the rendered and the groundtruth image would suffice to measure the performance and backpropagate the loss. However, this is impractical due to the high rendering times that would significantly slow down the learning process. Therefore, we recur to a loss function that measures the error between the parameters of the estimated and the ground truth meshes. One possible way to do that would be to compute the mean squared error over vertex positions

$$L_{full}(\mathbf{Y}^P, \mathbf{Y}^{GT}) = \frac{1}{n} \sum_{i=1}^{n} \left\| \mathbf{Y}_i^P - \mathbf{Y}_i^{GT} \right\|^2, \tag{1}$$

where $\mathbf{Y}^P$ is the predicted output, and $\mathbf{Y}^{GT}$ is the corresponding ground truth, and $\mathbf{Y}_i$ denotes the i-th row of $\mathbf{Y}$. If we regress to PCA component coefficients instead of vertices, we use following weighted mean squared error function:

$$L_{PCA}(\mathbf{Y}^P, \mathbf{Y}^{GT}) = \frac{1}{l} \sum_{i=1}^{l} w_i \left| \mathbf{Y}_i^P - \mathbf{Y}_i^{GT} \right|, \tag{2}$$

Here, w_i is the PCA variance ratio corresponding to the i-th principal component, and l is the number of components. In order to capture the curvature better and in turn the folds and wrinkles, we extend Eq.1 by integrating normal estimations through an additional

term in the loss. At each training iteration we compute the normals of the estimated vertices and compare them to the ground truth normals computed on the ground truth garment meshes. The final loss becomes:

$$L^*(\mathbf{Y}^P, \mathbf{Y}^{GT}) = \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{Y}_i^P - \mathbf{Y}_i^{GT}\right\|^2 - \lambda\left[k\left(\mathbf{N}_i^P\right)^T \mathbf{N}_i^{GT}\right]^3, \quad (3)$$

where the matrices $\mathbf{N}$ are the normals of the corresponding vertices, λ is a weighting term set by the user (throughout our experiments set to 1000) that controls the influence of the normals as opposed to the vertex positions, and k a stretching term (set to 3) of the dot product, which when combined with the cubic exponential, gives more weight to the penalization for estimated normals that form a large angle with respect to the ground truth. This new loss function not only fixes some of the global deformation rotations, but also stresses the high frequency wrinkle patters, as demonstrated in Sec. 5.

4.5. Two-View Architecture

We additionally tackle the problem of simultaneously predicting the garment mesh when more evidence is included through a second view (e.g. a front and back view image of the garment). It turns out that simply concatenating the images along the channel dimension and then passing them through an architecture similar to the single-view one described above performs worse than the networks trained on single-view input only. One reason for this would be that including a complementary view at the early stages of the network, for the same amount of training data, might inflict noise in the system as also observed in a recent work [DJO*16]. Hence, we decided to combine information coming from multiple views at a later stage by separately training two similar CNN-s on each view, and then concatenating the outputs of the last convolutional layer of each CNN through a view-pooling layer that performs either a max or a concatenation operation, as shown in Fig. 5. This architecture is capable of using the additional information from the multi-view input to produce more accurate results. The disadvantage of this architecture is that is has almost twice as many parameters and therefore doubles the training time and the memory needed.

4.6. Interpenetration Handling

Despite the fact that the garment shapes estimated from the CNN give small training and testing error, it can still happen that the estimated mesh does not fit the body perfectly but some vertices may be placed inside it, especially in cases where the input pose or body shape is very different from the shapes and poses that we consider during our training stage. Therefore, we employ a least squares energy minimization similar to [GRH*12] to push the interpenetrating vertices out of the body mesh. The energy (Eq.4) consists of multiple terms :

$$E_{\mathcal{B}}(\mathcal{Y}) = p_{\mathcal{C}}(\mathcal{Y}) + \lambda_s s(\mathcal{Y}) + \lambda_d d(\mathcal{Y}), \quad (4)$$

where $p_{\mathcal{C}}(\mathcal{Y})$ stands for the interpenetration term, $s(\mathcal{Y})$ for the smoothness term and $d(\mathcal{Y})$ for the damping term. Parameters λ_s and λ_d are used to weight the importance of the individual terms.

Garment-body interpenetration is the most important term of the objective function. It takes care of pushing the interpenetrating vertices out of the body mesh. Let $\mathcal{C}$ be a set of correspondences between each garment vertex $\vec{v}_i$ and its closest body mesh vertex $\vec{b}_j$. Let $\mathcal{P}$ be a set of vertices that are currently located inside the body. A garment vertex $\vec{v}_i$ is located inside the body if $\vec{n_{b_j}}^T\left(\vec{v}_i - \vec{b}_j\right) < 0$, where $\vec{n_{b_j}}$ is the normal of the body vertex $\vec{b}_j$. Hence we have

$$p_{\mathcal{C}}(\mathcal{Y}) = \sum_{(i,j)\in\mathcal{C}\wedge i\in\mathcal{P}}\left\|\varepsilon + \vec{n_{b_j}}^T\left(\vec{v}_i - \vec{b}_j\right)\right\|^2 \quad (5)$$

where ε is set to a small negative number to ensure that the garment vertices are moved safely out of the body. This equation is underdetermined and has infinitely many solutions, therefore two additional terms are added to regularize the system.

The Smoothness term is added to make sure that the vertices are being moved smoothly with respect to their neighbors. This prevents the final solution from having undesirable spikes in place of the interpenetrating vertices which are being moved out of the body.

$$s(\mathcal{Y}) = \sum_{i\in\mathbf{V}}\left\|\left(\vec{v}_i - \widetilde{\vec{v}_i}\right) - \frac{1}{|\mathbf{B}_i|}\sum_{j\in\mathbf{B}_i}\left(\vec{v}_j - \widetilde{\vec{v}_j}\right)\right\|^2 \quad (6)$$

where $\widetilde{\vec{v}_i}$ is the current position of vertex i, $\mathbf{V}$ the set of vertices and $\mathbf{B}_i$ the list of neighboring vertices of vertex $\vec{v}_i$.

The Damping term is added to favor solutions in which the positions of the vertices have not changed very much from the input mesh.

$$d(\mathcal{Y}) = \sum_{i\in\mathbf{V}}\left\|\left(\vec{v}_i - \widetilde{\vec{v}_i}\right)\right\|^2, \quad (7)$$

where λ_s and λ_d are tunable parameters we can set to control the impact of individual terms. In our experiments, we set $\lambda_s = 1.5$ and $\lambda_d = 0.8$.

Interpenetration algorithm The mere solution of the objective function minimization might not guarantee the removal of interpenetration for all vertices at once. Therefore we iterate over the described process multiple times to get rid of the interpenetration entirely, as described in Algorithm 1.

5. Experiments and Results

In order to assess the performance of our method, we evaluate it on synthetic and real images and videos, both qualitatively and quantitatively. The experiments consist of tight and loose clothing of male and female models with t-shirts and dresses simulated using physically based simulations on mocap data, and rendered under varying camera poses and lighting conditions. We demonstrate garment capture results on single images, two-view images and videos (supplementary).

Data: body mesh $\mathcal{B}$ and garment mesh $\mathcal{Y}_0$
Result: unpenetrating garment mesh $\mathcal{Y}$
iter=0;
while *iter* < *maxIter* **do**
  Find garment to body vertex correspondences $\mathcal{C}$;
  Find penetrating vertices
  $\mathcal{P} = \left\{ \vec{v}_i \;\middle|\; \forall i : \vec{n_{b_j}}^T \left(\vec{v}_i - \vec{b}_j \right) < 0 \right\}$;
  if $empty(\mathcal{P})$ **then**
    return $\mathcal{Y}_{iter}$;
  end
  Solve for: $\mathcal{Y}_{iter+1} = \underset{v_i \in \mathbf{V}_\mathcal{Y}}{arg\,min}\, E_\mathcal{B}(\mathcal{Y}_{iter})$;
  iter = iter + 1;
end
return $\mathcal{Y}_{iter-1}$;

Algorithm 1: Interpenetration removal algorithm

5.1. Datasets

Utilizing the pipeline described in Sec.3, we simulated around 100,000 T-shirt meshes on 7 male bodies of various shapes creating a geometry dataset (Fig.6 left). Likewise, we simulated around 15,000 dress meshes on a female character (Fig.6 right). We then construct the final dataset consisting of geometry and corresponding images under different lighting conditions and from front and back views, as explained in Sec.3. We separate the samples into a training dataset, containing 90% of the images and the corresponding geometries, and a testing dataset consisting of the rest. We would like to stress that our dataset consists of purely synthetic images, hence the training has never seen a real image, but is still able to capture plausible low-frequency deformations on real data.

Figure 6: *The garment meshes used for simulation*

5.2. Qualitative Evaluation

We firstly assess the estimation quality from the visual perspective and we encourage the reader to view this section electronically.

Synthetic Data We show results for our "Garment-from-Body" mesh representation on the T-shirt dataset in Fig.7 and the dress dataset in Fig.8, achieving accurate reconstructions. Note that the captured wrinkles lack some of the fine details not present in the input images to the network, since they are relatively small with a resolution of 64×64, as shown in the figures. However, we get realistic deformations with dynamic details at different scales preserved for all cases. The algorithm can recover the overall shape and deformation of the garments, as well as finer wrinkles and folds. One main advantage of our single image-based geometry estimation method is that we can capture deformations due to a dynamic motion, as opposed to methods that would simulate the garment assuming a known body shape and pose in a statically stable physical state, which is illustrated in the figures, and can be more clearly seen in Fig.8.

As we mention in Sec.3, our generation conditions contain multiple degrees of freedom (DOF), such as camera position, illumination and body pose change. In Fig. 9, 10 and 11, we illustrate that we get consistent estimations under different poses, lighting changes, and views, respectively. Incorporating these degrees of freedom into the database thus provides robust results under such changes, which is essential for a practical garment capture system.

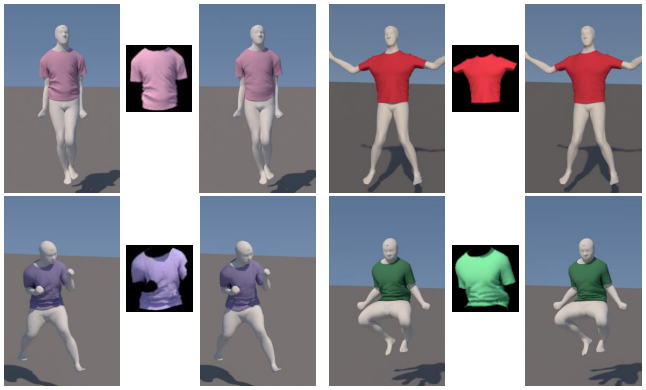

Figure 7: *Recovered garment shapes with the "Garment-from-Body" representation. From left to right: initial rendering, segmented T-shirt, and rendering of the same scene using the estimated mesh.*

Figure 8: *Recovered garment shapes with the "Garment-from-Body" representation. From left to right: initial rendering, segmented dress and rendering of the same scene using the estimated mesh.*

Real Data We evaluate the models trained on the "T-shirt" dataset on real data we captured in an uncontrolled environment with a cell-phone camera. Fig.12 shows the estimation on single-view inputs and Fig.13 on two-view inputs, utilizing the respective

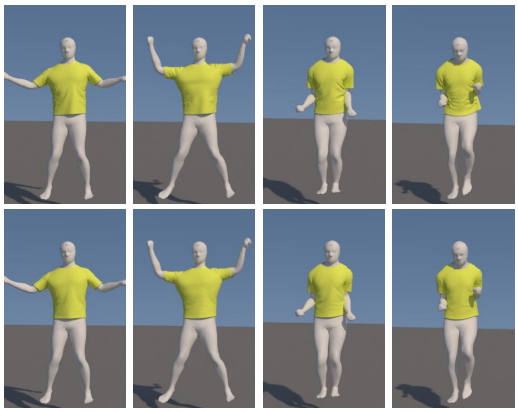

Figure 9: *Pose changes: input images (first row) and the estimations (second row).*

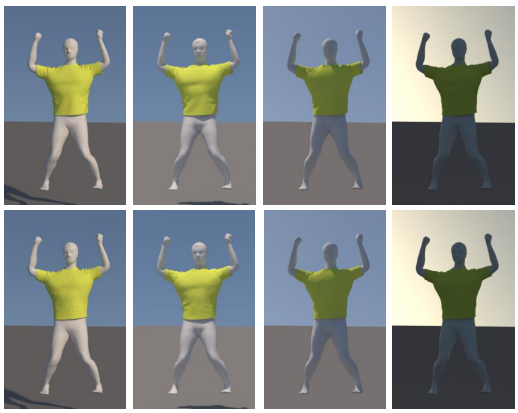

Figure 10: *Illumination changes: input images (first row) and the estimations (second row).*

Figure 11: *View changes: input images (first row) and the estimations (second row).*

CNN architectures as explained in Sec.4. The major deformations in shape and pose are captured accurately and look plausible. This is despite the fact that the material of the captured garment is quite different than the one we have in the database, and the input images to the network are very small. Hence, although we cannot capture small-scale wrinkle details, we still get quite faithful garment shapes. This generality also allows us to use images depicting textured garments as we show in Fig.13.

Furthermore, we evaluate our dress models on our "dress" dataset (Figure 14). Please note, that this is a much more challenging problem as dresses usually have much more variety in both intrinsic shape and material. Despite that, our technique can still recover the global shape.

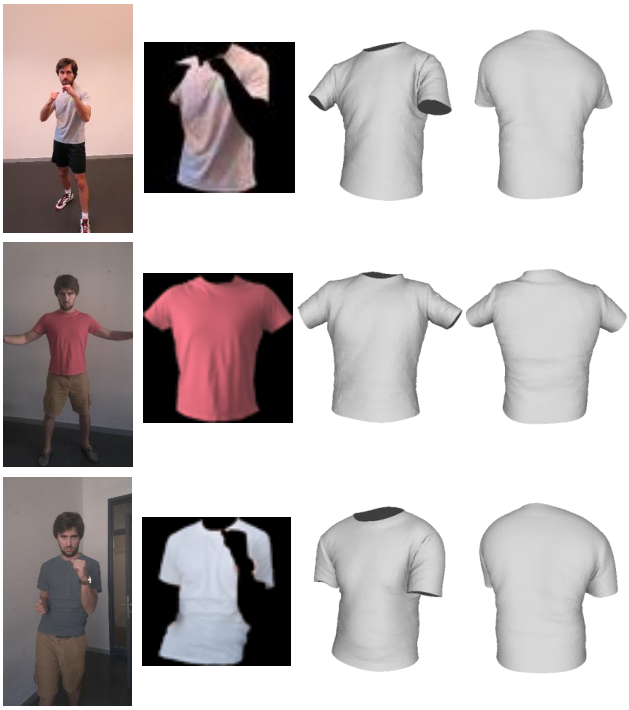

Figure 12: *Estimated T-shirt shapes from captured images. From left to right: original image, image after segmentation (input image), view of the estimated mesh from the front and back.*

5.3. Quantitative Evaluation

Due to the fact that there is no real-life garment image dataset that would contain accurate ground truth geometry, we quantitatively evaluate our method on synthetic datasets. For every quantitative experiment we report the average of the mean squared error of the vertex positions from the ground truth mesh over the entire training set, and the mean cosine similarity of the face normals of the estimated and ground truth meshes given by $\mathbf{n}^T\mathbf{n}'$ for the ground truth $\mathbf{n}$ and estimated $\mathbf{n}'$ normals.

The results are reported in Tables 1 and 2. While the absolute errors for the vertex positions per-se do not explicitly inform us on the quality of the individual reconstructions, they serve as a mean to assess and compare the generalization error over the various experiments that we consider, as we elaborate below.

Learned Shape Representation As mentioned in Sec.4, we

Figure 13: *Estimated garment shapes from two views. From left to right: original image from the front and back, image after segmentation (input image) from the front and back, view of the estimated meshes from the front and back.*

consider two mesh representation formulations. One outputs the PCA coefficients, which can be used to obtain the per-vertex deformations and the other performs the estimation on the full space, directly outputting the displacement for every vertex. The experiments have shown, that the models using PCA get outperformed significantly, having greater mean squared error and standard deviation. This points to the fact that the deep neural nets do a much better job in creating an internal representation of the data from the training samples than simple PCA. One potentially big disadvantage of the non-PCA model is its size. Because the output layer has over 18,000 units, the size of the model grows quickly. For instance, while the PCA model of SqueezeNet takes about 42MB, the full non-PCA model takes about 660MB.

Garment-from-Garment vs Garment-from-Body In this experiment, we compare the performance of the two formulations of the regression that we introduced earlier. The "Garment-from-Body" formulation achieves lower reconstruction errors as reported in the tables. This happens because the offsets from the body tend to be much smaller than those from the reference garment mesh in the T-pose. The scale of the estimated values is smaller and therefore the scale of the error is also smaller. This representation might, however, create a problem if we want to use the estimated garment to dress a body mesh, as the displacement is often too big for the interpenetration solver to work properly without distorting the mesh too much. For this reason the "Garment-from-Body" formulation is better for dressing characters and "Garment-from-Garment" is more suitable for reconstructing the garment only.

The Importance of Silhouettes A 2D silhouette of an object is one of the most important visual cues as it restricts the space of shapes the object could possible have in 3D. For this reason, the following experiment has been conducted. We have trained two models of our SqueezeNet incarnations. One was trained on the image dataset we have, the other was trained only on the silhouettes of the garments, loosing shading and color information. As it

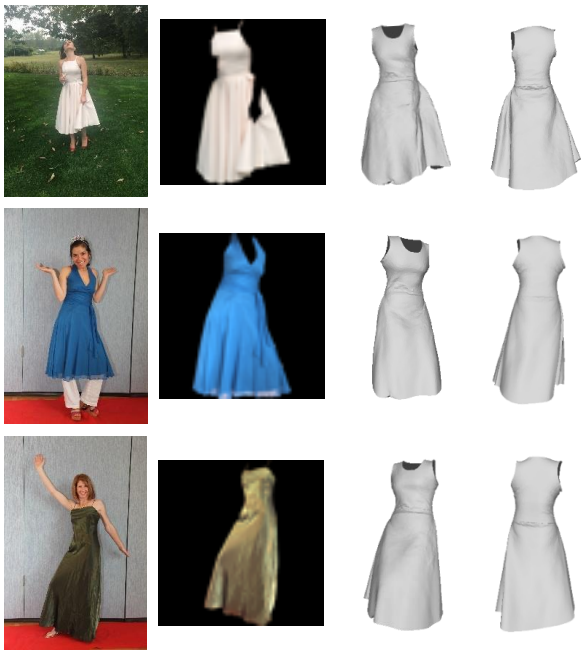

Figure 14: *Dress shapes estimated from real images. From left to right: original image, image after segmentation (input image), view of the estimated mesh from the front and back. Please note that none of the dresses match exactly our test dress in neither shape nor material stiffness or reflectance. Despite that, we are able to capture the overall shape even for more challenging images (such as the first image, where the actress grabs the side of her dress).*

model	MSE	NCS
GfG-PCA space frontal view	507.140± 781.390	0.903± 0.049
GfG-MSE full space frontal view	342.164± 522.742	0.906± 0.04
GfG-MSE full space frontal view silhouettes	496.308± 765.161	0.901± 0.047
GfG-MSE+normals full space frontal view	331.327± 557.942	0.916± 0.044
GfG-MSE+normalsExp full space frontal view	345.163± 607.051	0.921± 0.046
GfG-MSE+normals-viewMaxPool full space two views	323.168± 472.058	0.917± 0.041
GfB-MSE full space frontal view	81.037± 205.640	0.908± 0.048
GfB-MSE+normals full space frontal view	95.299± 194.844	0.900± 0.052

Table 1: *The performance of models trained on the "T-shirt" dataset. "GfG" stands for the "Garment from Garment" representation. "GfB" stands for the "Garment from Body" representation, each entry also contains information on which architecture and loss function was used. MSE stands for vertex mean squared error and NCS for mean cosine similarity of the face normals.*

can be observed in Tab.1, the silhouette model performs well, but is outperformed by the model trained on RGB images. This is an important result as it proves that CNNs can in fact learn the shading cues due to wrinkling patterns to further enhance the quality of the estimation. The silhouette though still remains an important cue that the network automatically learns.

Comparison of Single-View and Multi-View Nets In this experiment, we compare the performance of single and multi-view

model	MSE	NCS
GfG-MSE full space frontal view	294.487± 303.214	0.937± 0.043
GfG-MSE full space frontal view silhouettes	376.824± 387.903	0.925± 0.052
GfG-MSE+normals full space frontal view	297.833± 318.009	0.946± 0.040
GfG-MSE+normals-viewConcat full space two views	185.926± 222.316	0.965± 0.026

Table 2: *The performance of models trained on the "Dress" dataset. "GfG" stands for the "Garment-from-Garment" representation, and "GfB" for the "Garment-from-Body" representation. Each entry also contains information on which architecture and loss function was used. MSE stands for vertex mean squared error and NCS stands for mean cosine similarity of the face normals.*

CNN-s. The multi-view architectures achieve superior performance in comparison to the single view models on both datasets. Hence, our method can benefit from multi-view input to achieve a more accurate estimation. The difference is particularly visible on the dress dataset as loose clothing has naturally more ambiguities of shape in the occluded parts. Please refer to Fig.15 for a visual comparison.

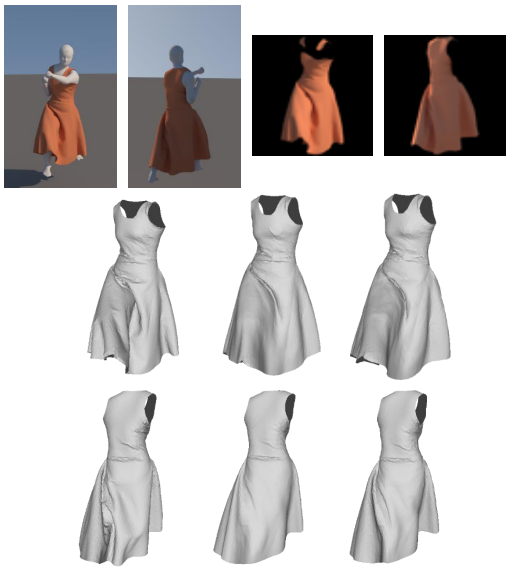

Figure 15: *Demonstration of superiority of multi-view models over single-view models. (top) From left to right: initial rendering frontal and back view, the segmented and rescaled garment, frontal and back viewObserve that the mesh estimated by the multi-view model is much more accurate than that estimated by the single-view model, especially on the back side. Front (middle) and back (bottom) views of (from left to right): ground truth geometry, geometry estimated by the single-view SqueezeNet model trained on frontal views, and geometry estimated by the multi-view SqueezeNet model.*

Loss Function with Normals We compare the performance of the models which are optimized for MSE and the models which try to account for the curvature of the surface of the estimated mesh by using the customized loss function in (3) as described in Section 4. The results in Tables 1 and 2 show that these models usually have slightly worse MSE, but their normals are aligned more accurately.

Failure Cases In Figure 16 we present interesting failure cases for garments with different kind of shape or pose, which was not present in the training set. Similarly, sometimes the input image is imperfect and contains noise (such as segmentation artifacts).

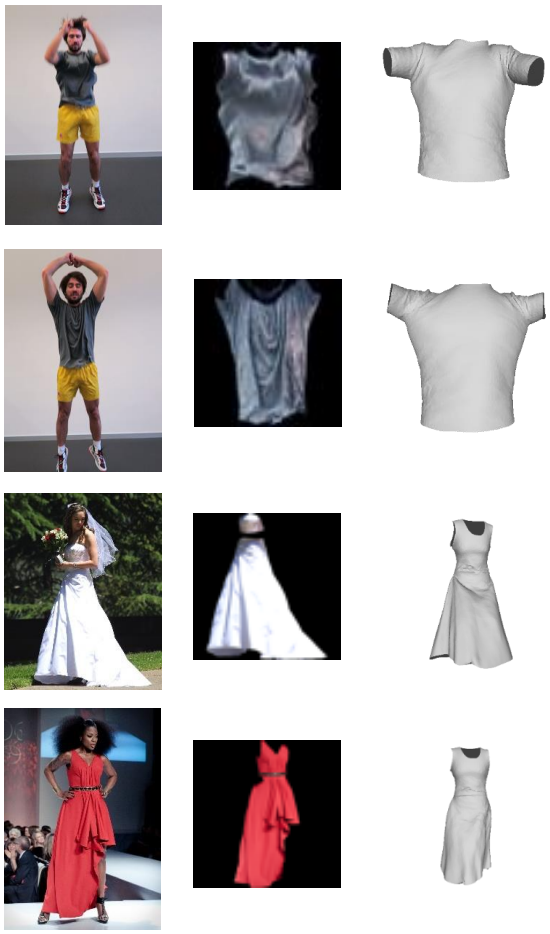

Figure 16: *Failure cases. From left to right: original image, image after segmentation (input image), view of the estimated mesh from the front.*
First row: *A very hard image. The similar silhouette is not present in the training set and the wrinkling patterns are also completely new. Such cases can cause the estimation to fail.*
Second row: *Here the estimation is hindered by the imperfect segmentation at the top of the image as well as unseen wrinkling patterns.*
Third row: *The specific dress captured slightly from the side is also not part of the training set. However, please observe how our technique tried to generate a very close shape (the wrinkles on the lower left part of the dress).*
Fourth row: *The "intrinsic" shape of this dress is dramatically different. Please observe how our model compensated for lack of clothing on the right with a more plausible inwards deformation.*

5.4. Performance

Training Time Deep neural nets are known to take a long time to train. However, this is not the case for our architectures. Thanks to the compressed nature of the SqueezeNet architecture, we are able to train our incarnations in less than 8 hours. This corresponds to less than 100 epochs over 100,000 samples till convergence. Hence, our method scales well to bigger garment databases as well.

Testing Time Since the overall estimation is a neural net inference followed by estimation of the offsets, the total test time taken is a few miliseconds. This makes our method practical even for real-time applications, in contrast to many other 3D shape estimation techniques, or garment simulations.

Data Generation Our data generation pipeline is not optimized for speed. The generation of all the data that we used for training using the pipeline described in Sec.3 took approximately 10 days on a cluster. However, please note that both simulation and rendering is done on a CPU and the process could most certainly be optimized to run orders of magnitude faster.

6. Conclusion and Limitations

We rely on a data generation pipeline where the deformation of the human body shape and thus the garment is given by a standard skinning model with automatically computed weights and simple blending of bone transformations (see Section 3). We also use an off-the-shelf physical simulator for the cloth deformation, and default parameters. These steps introduce certain artifacts. Better and tailored databases can be obtained by improving these steps, and considering accurate cloth parameters.

We illustrate results on videos in the supplementary material. Although the results are already temporally smooth overall, they can still exhibit certain artifacts or discontinuities, since we do not enforce any temporal constraints. This can be achieved by considering multiple frames, and incorporating a recurrent network architecture, which we leave as an interesting future direction.

Our method relies on segmented garments from an image, although it tolerates a certain amount of noise and inaccuracy as illustrated in Figure 4, and moderately textured garments. By training on more data covering a larger variety of cases, the method can be extended to handle complex textures and unsegmented images, for non-garment deforming objects as well.

For practical purposes, such as hardware or time constraints, the networks are trained with relatively small input images of 64×64 pixels. This prevents our method from capturing some of the wrinkles. With the current progress in the compactness of CNN representations, one could envisage higher-frequency details to be captured as well.

One of the limitations of our technique is the need to train a new model for every new clothing type since the reference garment mesh may vary dramatically in resolution due to the shape complexity. Additionally, the performance of our data-driven approach can only be as good as the training dataset. The more realistic and general the dataset becomes the more details can be captured.

We refer the reader to our supplementary material, where we expand further our discussion of the potential deep architectures that can be used. Furthermore, we propose a potential application of our technique to speed-up physically based clothing simulation. Finally, we include a more detailed discussion of the limitations of our technique.

In summary, we proposed an end-to-end 3D garment shape estimation algorithm, that captures plausible garment deformations at interactive rates, from an uncontrolled single view setup depicting a dynamic garment state. Additionally, we showed how to achieve this by training a CNN based regressor with statistical priors and specialized loss function, that utilizes purely synthetically generated data for training, demonstrating its scalability for such a supervised learning task. Finally, we showed how to leverage information simultaneously present in multi-view setups in order to increase the predictive performance of the network and in turn the accuracy of the estimations.

Acknowledgement This work was funded by the KTI-grant 15599.1. We thank the reviewers for their valuable inputs.

References

[AN14] ALEXANDROS NEOPHYTOU A. H.: A layered model of human body and garment deformation. 3

[ASK*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: Scape: Shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), SIGGRAPH '05, ACM, pp. 408–416. 2, 3

[BB08] BĂLAN A. O., BLACK M. J.: The naked truth: Estimating body shape under clothing. In *Proceedings of the 10th European Conference on Computer Vision: Part II* (Berlin, Heidelberg, 2008), ECCV '08, Springer-Verlag, pp. 15–29. 2

[BBGB16] BÉRARD P., BRADLEY D., GROSS M., BEELER T.: Lightweight eye capture using a parametric model. *ACM Trans. Graph. 35*, 4 (2016). 1

[BFL06] BOYKOV Y., FUNKA-LEA G.: Graph cuts and efficient n-d image segmentation. *Int. J. Comput. Vision 70*, 2 (Nov. 2006), 109–131. 4

[BKC15] BADRINARAYANAN V., KENDALL A., CIPOLLA R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561* (2015). 4

[BP07] BARAN I., POPOVIĆ J.: Automatic rigging and animation of 3d characters. In *ACM SIGGRAPH 2007 Papers* (New York, NY, USA, 2007), SIGGRAPH '07, ACM. 3

[BPS*08] BRADLEY D., POPA T., SHEFFER A., HEIDRICH W., BOUBEKEUR T.: Markerless garment capture. *ACM Trans. Graphics (Proc. SIGGRAPH) 27*, 3 (2008), 99. 2

[CBZB15] CAO C., BRADLEY D., ZHOU K., BEELER T.: Real-time high-fidelity facial performance capture. *ACM Trans. Graph. 34*, 4 (July 2015), 46:1–46:9. 1

[CKC10] CHEN Y., KIM T.-K., CIPOLLA R.: Inferring 3d shapes and deformations from single views. In *ECCV (3)* (2010), Daniilidis K., Maragos P., Paragios N., (Eds.), vol. 6313 of *Lecture Notes in Computer Science*, Springer, pp. 300–313. 3

[Cmu] CMU: Carnegie-Mellon Mocap Database. 3

[CZL*15] CHEN X., ZHOU B., LU F., WANG L., BI L., TAN P.: Garment modeling with a depth camera. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 203:1–203:12. 2, 3

[DJO*16] DIBRA E., JAIN H., OZTIRELI C., ZIEGLER R., GROSS M.: Hs-nets : Estimating human body shape from silhouettes with convolutional neural networks. In *Int. Conf. on 3D Vision* (October 2016). 1, 2, 3, 6

[DOZG16] DIBRA E., OZTIRELI C., ZIEGLER R., GROSS M.: Shape from selfies : Human body shape estimation using cca regression forests. In *Proceedings of the 14th European Conference on Computer Vision: Part IV* (2016), ECCV '16. 1, 3

[FDI*15] FISCHER P., DOSOVITSKIY A., ILG E., HÄUSSER P., HAZIRBAS C., GOLKOV V., VAN DER SMAGT P., CREMERS D., BROX T.: Flownet: Learning optical flow with convolutional networks. *CoRR abs/1504.06852* (2015). 3

[GFB10] GUAN P., FREIFELD O., BLACK M. J.: A 2d human body model dressed in eigen clothing. In *Proceedings of the 11th European Conference on Computer Vision: Part I* (Berlin, Heidelberg, 2010), ECCV'10, Springer-Verlag, pp. 285–298. 3

[GKB03] GUSKOV I., KLIBANOV S., BRYANT B.: Trackable surfaces. In *SCA* (2003). 2

[GRH*12] GUAN P., REISS L., HIRSHBERG D. A., WEISS A., BLACK M. J.: Drape: Dressing any person. *ACM Trans. Graph. 31*, 4 (July 2012), 35:1–35:10. 2, 3, 6

[HMLL15] HU L., MA C., LUO L., LI H.: Single-view hair modeling using a hairstyle database. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2015) 34*, 4 (July 2015). 1

[HTC*14] HAHN F., THOMASZEWSKI B., COROS S., SUMNER R. W., COLE F., MEYER M., DEROSE T., GROSS M.: Subspace clothing simulation using adaptive bases. *ACM Trans. Graph. 33*, 4 (July 2014), 105:1–105:9. 2, 3

[HZRS15] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. *CoRR abs/1512.03385* (2015). 3

[IMA*16] IANDOLA F. N., MOSKEWICZ M. W., ASHRAF K., HAN S., DALLY W. J., KEUTZER K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR abs/1602.07360* (2016). 3, 5

[Jak10] JAKOB W.: Mitsuba renderer, 2010. http://www.mitsuba-renderer.org. 4

[JHK15] JEONG M.-H., HAN D.-H., KO H.-S.: Garment capture from a photograph. *Comput. Animat. Virtual Worlds 26*, 3-4 (May 2015), 291–300. 2, 3

[KCvO08] KAVAN L., COLLINS S., ŽÁRA J., O'SULLIVAN C.: Geometric skinning with approximate dual quaternion blending. *ACM Trans. Graph. 27*, 4 (Nov. 2008), 105:1–105:23. 3

[KSH12] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.* (2012), pp. 1106–1114. 3, 5

[LLQ*16] LIU Z., LUO P., QIU S., WANG X., TANG X.: Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). 3

[MD] MD: Marvelous designer. 3

[NPO13] NARAIN R., PFAFF T., O'BRIEN J. F.: Folding and crumpling adaptive sheets. *ACM Transactions on Graphics 32*, 4 (July 2013), 51:1–8. Proceedings of ACM SIGGRAPH 2013, Anaheim. 3

[NSO12] NARAIN R., SAMII A., O'BRIEN J. F.: Adaptive anisotropic remeshing for cloth simulation. *ACM Transactions on Graphics 31*, 6 (Nov. 2012), 147:1–10. Proceedings of ACM SIGGRAPH Asia 2012, Singapore. 3

[PH03] PRITCHARD D., HEIDRICH W.: Cloth motion capture, 2003. 2

[PWH*15] PISHCHULIN L., WUHRER S., HELTEN T., THEOBALT C., SCHIELE B.: Building statistical shape spaces for 3d human modeling. *CoRR abs/1503.05860* (2015). 3

[PZB*09] POPA T., ZHOU Q., BRADLEY D., KRAEVOY V., FU H., SHEFFER A., HEIDRICH W.: Wrinkling captured garments using space-time data-driven deformation. *Computer Graphics Forum (Proc. Eurographics) 28*, 2 (2009), 427–435. 2

[RMSC11] ROBSON C., MAHARIK R., SHEFFER A., CARR N.: Context-aware garment modeling from sketches. *Computers and Graphics (Proc. SMI 2011)* (2011), 604–613. 2

[SBB07] SIGAL L., BALAN A. O., BLACK M. J.: Combined discriminative and generative articulated pose and non-rigid shape estimation. In *NIPS* (2007), Platt J. C., Koller D., Singer Y., Roweis S. T., (Eds.), Curran Associates, Inc. 3

[SEZ*13] SERMANET P., EIGEN D., ZHANG X., MATHIEU M., FERGUS R., LECUN Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR abs/1312.6229* (2013). 3

[SLJ*15] SZEGEDY C., LIU W., JIA Y., SERMANET P., REED S., ANGUELOV D., ERHAN D., VANHOUCKE V., RABINOVICH A.: Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)* (2015). 3

[SM04] SCHOLZ V., MAGNOR M. A.: Cloth motion from optical flow. In *VMV* (2004). 2

[SMKL15] SU H., MAJI S., KALOGERAKIS E., LEARNED-MILLER E. G.: Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV* (2015). 3

[SSK*05] SCHOLZ V., STICH T., KECKEISEN M., WACKER M., MAGNOR M.: Garment motion capture using color-coded patterns. *Computer Graphics forum 24*, 3 (September 2005), 439–448. Conference Issue: 26th annual Conference Eurographics 2005, Dublin, Ireland, August 29th - September 2nd, 2005. 2

[SSP*14] SEKINE M., SUGITA K., PERBET F., STENGER B., NISHIYAMA M.: Virtual fitting by single-shot body shape estimation. In *Int. Conf. on 3D Body Scanning Technologies* (October 2014), pp. 406–413. 3

[SZ14] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. *CoRR abs/1409.1556* (2014). 3

[TDB16] TATARCHENKO M., DOSOVITSKIY A., BROX T.: Multi-view 3d models from single images with a convolutional network. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII* (2016), pp. 322–337. 2

[TPT16] TKACH A., PAULY M., TAGLIASACCHI A.: Sphere-meshes for real-time hand modeling and tracking. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* (2016). 1

[TS13] TOSHEV A., SZEGEDY C.: Deeppose: Human pose estimation via deep neural networks. *CoRR abs/1312.4659* (2013). 3

[WCF07] WHITE R., CRANE K., FORSYTH D.: Capturing and animating occluded cloth. In *ACM Transactions on Graphics (SIGGRAPH)* (2007). 2

[WKL15] WANG F., KANG L., LI Y.: Sketch-based 3d shape retrieval using convolutional neural networks. *CoRR abs/1504.03504* (2015). 3

[WSK*15] WU Z., SONG S., KHOSLA A., YU F., ZHANG L., TANG X., XIAO J.: 3d shapenets: A deep representation for volumetric shapes. In *CVPR* (2015), IEEE Computer Society, pp. 1912–1920. 2, 3

[YAP*16] YANG S., AMBERT T., PAN Z., WANG K., YU L., BERG T. L., LIN M. C.: Detailed garment recovery from a single-view image. *CoRR abs/1608.01250* (2016). 2, 3, 4

[YHKB13] YAMAGUCHI K., HADI KIAPOUR M., BERG T. L.: Paper doll parsing: Retrieving similar styles to parse clothing items. In *The IEEE International Conference on Computer Vision (ICCV)* (December 2013). 4

[YTLF16] YI K. M., TRULLS E., LEPETIT V., FUA P.: LIFT: learned invariant feature transform. *CoRR abs/1603.09114* (2016). 3

[ZCF*13] ZHOU B., CHEN X., FU Q., GUO K., TAN P.: Garment modeling from a single image. *Pacific Graphics* (2013). 2