ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2020
J. Bender and T. Popa
(Guest Editors)

Volume 39 (2020), Number 8

A Pixel-Based Framework for Data-Driven Clothing

N. Jin[†1], Y. Zhu[2,3], Z. Geng[2], and R. Fedkiw[2,3]

[1]Calico Labs, USA [2]Stanford University, USA [3]Epic Games, USA

Abstract

We propose a novel approach to learning cloth deformation as a function of body pose, recasting the graph-like triangle mesh data structure into image-based data in order to leverage popular and well-developed convolutional neural networks (CNNs) in a two-dimensional Euclidean domain. Then, a three-dimensional animation of clothing is equivalent to a sequence of two-dimensional RGB images driven/choreographed by time dependent joint angles. In order to reduce nonlinearity demands on the neural network, we utilize procedural skinning of the body surface to capture much of the rotation/deformation so that the RGB images only contain textures of displacement offsets from skin to clothing. Notably, we illustrate that our approach does not require accurate unclothed body shapes or robust skinning techniques. Additionally, we discuss how standard image based techniques such as image partitioning for higher resolution can readily be incorporated into our framework.

CCS Concepts

• ***Computing methodologies*** → ***Animation; Neural networks; Computer vision representations;***

1. Introduction

Virtual clothing has already seen widespread adoption in the entertainment industry including feature films (e.g., Yoda [BFA02], Dobby [BMF03], Monsters, Inc. [BWK03]), video games (e.g., [MHHR07, MC10, dASTH10, KGBS11, KCMF12, KKN*13]), and VR/AR and other real-time applications (e.g., [MTCSP04, HE09, WHRO10, XUC*14]). However, its potential use in e-commerce for online shopping and virtual try-on ([HWW*18, SOC19, PLPM20]) is likely to far surpass its use in the entertainment industry especially given that clothing and textiles is a three trillion dollar industry (`https://fashionunited.com/global-fashion-industry-statistics`). Whereas games and real-time applications can use lower quality cloth and films have the luxury of a large amount of time and manual efforts to achieve more realistic cloth, successful e-commerce clothing applications demand high quality predictive clothing with fast turnaround, low computational resource usage, and good scalability.

Although there have been many advances in cloth simulation, the ability to match real cloth of a specific material, especially with highly detailed wrinkling, hysteresis, etc. is rather limited. Moreover, contact and collision approaches typically lack physical accuracy due to unknown parameters dependent on a multitude of factors even including body hair density and garment thread friction. Thus, while embracing simulation and geometric techniques wherever possible, we pursue a new paradigm approaching clothing on humans in a fashion primarily driven by data at every scale. This is rather timely as 3D cloth capture technology is starting to seem very promising [RDAHT15, CZL*15, PMPHB17].

Motivated by a number of recent works that view cloth deformations as offsets from the underlying body [GRH*12, NH14, PMPHB17, YFHWW18] as well as the recent phenomenal impact of convolutional neural networks for image processing [KSH12, SZ14, HZRS15, RPB15, LSD15, RHGS17], we recast cloth deformation as an image space problem. That is, we shrink wrap the cloth mesh onto the underlying body shape, viewing the shrink-wrapped vertex locations as pixels with RGB values representing displacements from body to cloth. By factoring out the complex nonlinear skinning as a pre-process and focusing only on smoother residual displacements from skin to cloth, we reduce demands on the neural network; not only does the network require less nonlinearity, but it is also easier to differentiate and train. These cloth pixels are barycentrically embedded into the triangle mesh of the body, and as the body deforms the pixels move along with it; however, they remain at fixed locations in the pattern space of the cloth just like standard pixels on film. Thus, cloth animation is equivalent to playing an RGB movie on the film in pattern space, facilitating a straightforward application of CNNs. Each cloth shape is an image, and time dependent joint angles choreograph image sequences that specify the deforming cloth.

Although we leverage body skinning [**?**, MTLT88, ASK*05, Kv05, KCvO07, LMR*15] to move the cloth pixels around in world space, we are not constrained by a need to ascertain the unclothed body shape accurately as other authors aim to [NH14, PMPHB17]. Of course, an accurate unclothed body shape might reduce variability in the cloth RGB image to some degree, but it is likely that CNN

† Work done while at Stanford University

network efficacy will advance faster than the technology required to obtain and subsequently accurately pose unclothed body shapes. Even if consumers were willing to provide more accurate unclothed body data or inferences of their unclothed body forms improve, it is still difficult to subsequently pose such bodies to create accurate shapes governed by animation parameters such as joint angles. In contrast, we demonstrate that CNNs can learn the desired clothing shapes even when unclothed body shapes are intentionally modified to be incorrect, thus providing some immunity to problematic skinning artifacts (e.g., candy wrapper twisting [Kv05,KCvO07,JS11]).

We demonstrate the efficacy of our approach on a T-shirt and its generality on a necktie. Data and code are available at https://physbam.stanford.edu/~njin19/pixel_based_cloth/. Furthermore, researchers have already begun to utilize our approach to advance the state-of-the-art for predicting cloth shape from body pose. [GJF19] incorporates physical priors on mesh deformation energies and trains the model end-to-end, differentiating though either a second order cone program or a quasi-static physics simulation before our network, and thus is greatly enabled by our well-conditioned approach that removes nonlinearities into a skinning pre-process. In addition, [WJG*20] takes a hierarchal approach applying our framework once as we do and then sequentially a second time in order to learn a corrective perturbation to the texture coordinates.

2. Related Work

Skinning: Linear blend skinning (LBS) [Lan98, MTLT88] is perhaps the most popular skinning scheme used in animation software and game engines. Although fast and computationally inexpensive, LBS suffers from well-known artifacts such as candy wrapper twisting, elbow collapse, etc., and many works have attempted to alleviate these issues, e.g., spherical blend skinning (SBS) [Kv05], dual-quaternion skinning (DQS) [KCvO07], stretchable and twistable bones skinning (STBS) [JS11], optimizied centers of rotations [LH16], etc. Another widely used geometric skinning approach is the Delta Mush (DM) [MDRW14]. Notably, similar to our cloth pixels that store displacements from body to cloth, DM stores the per-vertex difference ("delta") between the original and smoothed version of the rest state in local surface coordinates, and applies these "delta"s to the smoothed deformed surface. Another line of works explicitly model pose specific skin deformation from sculpted or captured example poses. For example, pose space deformation (PSD) [LCF00] uses radial basis functions to interpolate between artist-sculpted surface deformations, [KM04] extends PSD to weighted PSD, and [ACP02] uses k-nearest neighbor interpolation. EigenSkin [KJP02] constructs compact eigenbases to capture corrections to LBS learned from examples. The SCAPE model [ASK*05] decomposes pose deformation of each mesh triangle into a rigid rotation R from its body part and a non-rigid deformation Q and learns Q as a function of nearby joints, and BlendSCAPE [HLRB12] extends this expressing each triangle's rigid rotation as a linear blend of rotations from multiple parts. [LMR*15] learns a statistical body model SMPL that skins the body surface from linear pose blendshapes along with identity blendshapes. More recently, [BODO18] uses neural networks to approximate the non-linear component of surface mesh deformations from complex character rigs to achieve real-time deformation evaluation for film productions. Still, skinning remains one of the most challenging problems in the animation of virtual characters; thus, we illustrate that our approach has the capability to overcome some errors in the skinning process.

Cloth Skinning and Capture: A number of authors have made a library of cloth versus pose built primarily on simulation results and pursued ways of skinning the cloth for poses not in the library. [WHRO10] looks up a separate wrinkle mesh for each joint and blends them, and similarly [XUC*14] queries nearby examples for each body region and devises a sensitivity-optimized rigging scheme to deform each example before blending them. [KKN*13] incrementally constructs a secondary cloth motion graph. [dASTH10] learns a linear function for the principal component coefficients of the cloth shape, and [HTC*14] runs subspace simulation using a set of adaptive bases learned from full space simulation data. Extending the SCAPE model to cloth, [GRH*12] decomposes per-triangle cloth deformation into body shape induced deformation D, rigid rotation R, and non-rigid pose induced deformation Q, and applies PCA on D and Q to reduce dimensionality. Whereas [GRH*12] treats the cloth as a separate mesh, [NH14] models cloth as an additional deformation of the body mesh and learns a layered model. More recently [PMPHB17] builds a dataset of captured 4D sequences and retargets cloth deformations to new body shapes by transfering offsets from body surfaces. The aforementioned approaches would all likely achieve more realistic results using real-world cloth capture as in [PMPHB17, LCT18] as opposed to physical simulations.

Networks: Some of the aforementioned skinning type approaches to cloth and bodies learn from examples and therefore have procedural formulas and weights which often require optimization in order to learn, but here we focus primarily on methods that use neural networks in a more data-driven as opposed to procedural fashion. While we utilize procedural methods for skinning the body mesh and subsequently finding our cloth pixel locations, we use data-driven networks to define the cloth deformations; errors in the procedural skinning are simply incorporated into the offset function used to subsequently reach the data. Several recent works used neural networks for learning 3D surface deformations for character rigs [BODO18] and cloth shapes [DDO*17, LCT18, YFHWW18, WCPM18, SOC19, GCS*19]. In particular, [BODO18, YFHWW18] input pose parameters and output nonlinear shape deformations of the skin/cloth, both using a fully connected network with a few hidden layers to predict PCA coefficients. [DDO*17] takes input images from single or multiple views and uses a convolutional network to predict 1000 PCA coefficients. [LCT18] takes a hybrid approach combining a statistical model for pose-based global deformation with a conditional generative adversarial network (CGAN) for adding details on normal maps to produce finer wrinkles.

2D Representation of 3D Surfaces: There has been a rich body of works that leverage 2D representations of 3D meshes [SPR07] for a variety of tasks. For instance, 2D texture maps [Hec86] can be conveniently adapted to store properties of 3D surfaces. [FWS*18] stores per-vertex positions of the full 3D face shape as RGB values of the face texture map and trains a network to regress it

from a single face image. [LCT18] stores per-vertex normal vectors as RGB values of the garment texture map and learns to upsample it. In contrast to these works, our cloth pixels store offsets from their skinned positions as RGB values in the texture space. Our approach could thus be considered an Arbitrary Lagrangian-Eulerian (ALE [Mar97]) method where the computational domain follows the material partially but not fully, i.e., our cloth pixels follow only the deformation captured by body skinning. In addition to texture maps, geometry images [GGH02] present another option for flattening 3D shape surfaces. [SBR16] uses CNNs to learn on geometry images created via spherical authalic parametrization, and [SUHR17] learns to predict geometry images for generating genus-0 surfaces. Unlike [SBR16, SUHR17], we do not strive for a general method that handles both rigid and deformable objects, but rather a specialized method for the specific problem of 3D clothing.

3. Pixel-Based Cloth

We assign UV texture coordinates to each vertex and transform the cloth mesh to this two-dimensional space, as shown for the front side of a T-shirt mesh in Figure 1a. Each vertex stores a vector-valued function of displacements $\mathbf{dx}(u,v) = (\Delta u, \Delta v, \Delta n)$ representing perturbations in the texture coordinate and normal directions. This can be visualized by moving each vertex by $\mathbf{dx}$ as shown in Figure 1b. These displacements can be converted to per-vertex RGB colors as shown in Figure 1c; thus, we refer to these vertices as *cloth pixels*. Note that the RGB colors may contain values outside the visible range using HD image formats, floating point representations, etc. This framework allows us to leverage standard texture mapping [BN76, Cat74, Hec86] as well as other common approaches, e.g. bump maps [Bli78] to perturb normal directions and displacement maps [Coo84] to alter vertex positions; these techniques have been well-established over the years and have efficient implementations on graphics hardware enabling us to take advantage of the GPU-supported pipeline for optimized performance.

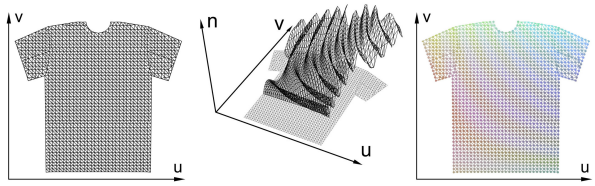

Figure 1: Left: Triangle mesh depicted in texture space using the vertices' UV coordinates. Middle: depiction of the displacement via $(u,v,0) + \mathbf{dx}$ for each vertex. Right: visualization of the displacement field $\mathbf{dx}$ converted into RGB values normalized to the visible $[0,255]$ range.

4. Cloth Images

The vertex data is connected via a triangle mesh topology that could be addressed using graph learning techniques [BZSL14, HBL15, MBBV15, DBV16, BBL*16], see in particular [TGLX18]. Alternatively, since our cloth pixels have fixed UV coordinates independent of the deformation, we may readily interpolate to a uniform background Cartesian grid of pixels using triangle rasterization ([FvDFH96]) while adding some padding at the boundaries to ensure smoothness (see Figure 2), thus facilitating an efficient application of standard CNNs especially via GPUs.

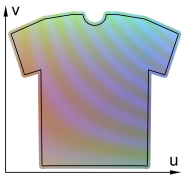

Figure 2: Standard uniform Cartesian grid of pixels for our cloth image. We add some padding to ensure smoothness on the boundaries for convolutional filters.

Note that we convert all the training data into pixel-based cloth images and train on those images directly, so the network learns to predict 2D images, not 3D cloth shapes. If one wanted to connect body pose parameters to cloth vertex positions in a fully end-to-end manner, then the interpolatory approach back and forth between the triangle mesh vertices and the Cartesian grid pixels might require further scrutiny. For example, the fluid dynamics community takes great care in addressing the copying back and forth of data between particle-based data structures (similar to our cloth pixels in Figure 1c) and background grid degrees of freedom (similar to our cloth image in Figure 2). Most notable are the discussions on PIC/FLIP, see e.g. [JSS*15].

4.1. Cage and Patch Based Cloth

Quite often one needs to down-sample images, which creates problems for learning high frequency details. Instead, we use a support "cage" to divide the cloth mesh into smaller patches to aid the learning process, see Figure 3. This notion of a cage and patch based cloth is quite powerful and is useful for capture, design, simulation, blendshape systems, etc. (see supplementary C for more discussions). While cloth already exhibits spatially invariant physical properties making it suitable for convolutional filters and other spatially coherent approaches, further dividing it into semantically coherent individual patches allows a network to enjoy a higher level of specialization and performance, as shown

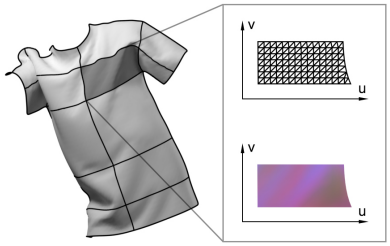

Figure 3: Left: front side of a T-shirt mesh divided into patches by a "cage" (depicted as black edges). Right: the triangulated cloth pixels and corresponding RGB cloth image for the highlighted patch.

in our experiments in Section 8.3. The only caveat is that one needs to take care to maintain smoothness and consistency across patch boundaries, but this can be achieved using a variety of techniques such as GANs [GPAM*14, LLYY17], image inpainting [BSCB00, YLY*18], PCA filtering, etc.

5. Skinning Cloth Pixels

While the cloth pixels have fixed UV locations in their 2D pattern space, their real-world 3D positions change as the body moves. We generate real-world positions for the cloth pixels by barycentrically embedding each of them into a triangle of the body mesh. Then as the body mesh deforms, the real-world locations of the cloth pixels move along with the triangles they were embedded into. Figure 4a shows the pixel RGB values from Figure 1c embedded into the rest pose and a different pose. Applying the **dx** offsets depicted in Figure 1b to the real-world pixel locations in Figure 4a yields the cloth shapes shown in Figure 4b.

(a) The cloth pixels are shown embedded into body triangles with RGB values copied over from Figure 1c in the rest pose and a different pose.

(b) The final cloth shapes obtained by adding displacements **dx** depicted in Figure 1b to the cloth pixel locations in (a).

Figure 4: Skinned cloth pixels and corresponding 3D cloth shapes in the rest pose and a different pose.

In Figure 5, we show the process reversed where the cloth shape shown in Figure 5 left is recorded as **dx** displacements and stored as RGB values on the cloth pixels embedded in the body mesh, see Figure 5 middle. These pixel RGB values in turn correspond to a cloth image in the pattern space, see Figure 5 right.

Figure 5: Left: part of a 3D cloth shape. Middle: cloth pixels embedded on the body mesh storing displacements **dx** as RGB values. Right: corresponding cloth image in the two-dimensional pattern space.

In order to obtain barycentric embeddings of the cloth pixels to the triangles of the body mesh, we start in a rest pose (T-shape) and uniformly shrink the edges of the cloth mesh making it skin-tight on the body. This is achieved by leveraging the cage structure defined on the body and the T-shirt as correspondence for a morphing operation followed by collision detection and pushout. Since this preprocessing step is only done once, and moreover can be accomplished on a template mesh, we take some care in order to achieve a good sampling distribution of the body deformations that drive our cloth image. Specifically, we run a fine-tuning simulation to reduce mesh distortion while constraining the T-shirt to be skin-tight. See supplementary material D.5 for more details. Note that our formulation readily allows for more complex clothing (such as shirts/jacket collars) to be embedded on the body with overlapping folds in a non-one-to-one manner, i.e., the inverse mapping from the body texture coordinates to the cloth texture coordinates does not need to exist (see supplementary material A).

One might alternatively skin the cloth as discussed in Section 2 to obtain a candidate cloth shape, and embed our cloth pixels into that skinned cloth, learning offsets from the skinned cloth to the simulated or captured cloth. The difficulty with such an approach is that example-based cloth can behave unpredictably making it difficult for a network to learn the offsets. Thus, we prefer to embed our pixels into the better behaved and more predictable skin of the body geometry, allowing us to leverage the numerous efforts and successes of that community (as opposed to the small number of works on cloth skinning).

6. Dataset

6.1. Pre-processing

We use a commercial solution [Art] to scan a person in the T-pose and manually rig the body using Blender [Ble18]. The *rest state* of our garment mesh is carefully constructed to be faithful to real-world garments in order to reduce distortion and ease the learning task. We employ a reverse engineering approach where we cut up a T-shirt along its seam lines, scan in the 2D pieces, and then digitally stitch them back together (see supplementary D.4). The T-shirt mesh is 67 cm long and contains 3K vertices. The texture map is then derived from this rest state mesh.

6.2. Pose Sampling

We generate poses for the upper body by independently sampling rotation angles along each axis for 10 joints in the upper body from a uniformly random distribution in their natural range of motion, and then applying a simple pruning procedure to remove invalid poses, e.g., with severe nonphysical self-penetrations. For more details, see supplementary D.3.

6.3. Cloth Simulation

For each sampled pose, we first skin the cloth onto the body. Then we take this skinned cloth as a collision-free starting point and simulate the garment mesh in a physics engine [Phy] with gravity, elastic and damping forces, and collision, contact, and friction forces until static equilibrium is reached. Any meshes that exhibit simulation artifacts (large area distortion) are subsequently discarded to reduce noise. See supplementary D.6 for more details.

The T-shirt dataset contains 20,011 samples and is divided into an 80% training set (16,009 samples), a 10% regularization set (2,001 samples to prevent the network from overfitting), and a 10%

test set (2,001 samples that the optimization never sees during training). The test set is used for model comparisons in terms of loss functions and network architectures, and serves as a proxy for generalization error. See Figure 6 for some examples.

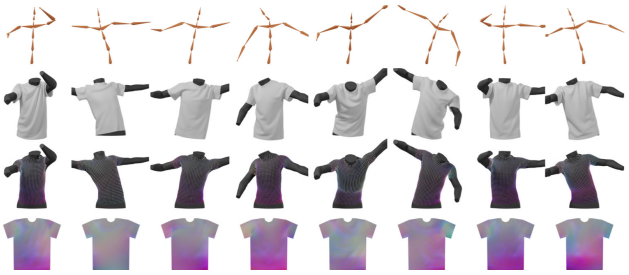

Figure 6: Dataset samples. Top to bottom: skeletal poses, simulated T-shirts, skinned cloth pixels, and cloth images (front side).

7. Learning Cloth Images from Skeletal Poses

Given input pose parameters, we train a CNN to predict cloth images, see Figure 8. These images represent offsets **dx** in local geodesic coordinates u and v and the normal direction n in order to enable the representation of complex surfaces via simpler functions (e.g., see Figure 7); even small perturbations in offset directions can lead to interesting structures.

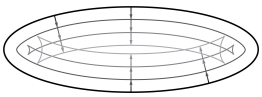

Figure 7: An ellipse with constant offsets in the normal direction results in the well-known swallowtail structure (see for example page 21 of [Set99]).

Although fully connected networks have been a common choice for generating dense per-vertex 3D predictions such as in [BODO18, YFHWW18], coalescing a 2D triangulated surface into a 1D vector forgos potentially important spatial adjacency information and may lead to a bigger network size as pointed out in [FWS*18]. A commonly employed remedy resorts to linear dimensionality reduction methods such as PCA to recover some amount of spatial coherency and smoothness in the output, as the regularized network predicts a small number of PCA coefficients instead of the full degrees of freedom. Alternatively, our pixel-based cloth framework leverages convolutional networks that are particularly well-suited for and have demonstrated promising results in tasks in the image domain where the filters can share weights and exploit spatial coherency.

Our convolutional decoder network takes in $1 \times 1 \times 90$ dimensional input rotation matrices, and applies transpose convolution, batch normalization, and ReLU activation until the target output size of $256 \times 256 \times 6$ is reached (see Figure 8 bottom left), where 3 output channels represent offset values for the front side of the T-shirt and 3 channels represent those of the back.

For training the model, the base loss is defined on the standard

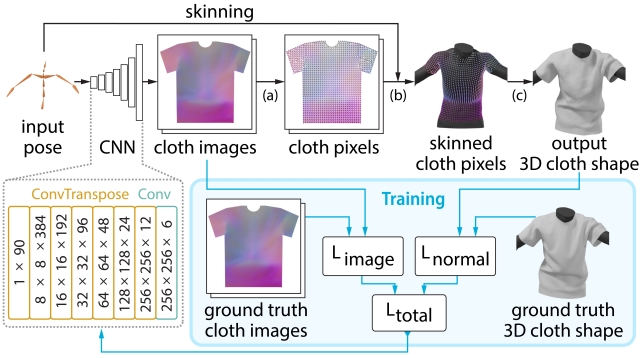

Figure 8: Overview of our learning system. **Top:** The CNN takes input poses and outputs cloth images. **(a)** The network predicted cloth images are interpolated back to cloth pixels in the texture space. **(b)** Cloth pixels are skinned onto the posed body using precomputed embedding weights. **(c)** The displacements **dx** encoded in the cloth pixels are added to their skinned locations to obtain the 3D cloth shape. **Bottom left:** CNN architecture. **Bottom right:** During training, loss is computed on the cloth images as well as the normal vectors of the 3D cloth shapes.

Cartesian grid pixels of the cloth images, weighted by a Boolean mask of the padded UV map:

$$\mathcal{L}_{\text{image}} = \sum_k \left(\frac{\sum_{i,j} W_k(i,j) ||\mathbf{I}_k^{pd}(i,j) - \mathbf{I}_k^{gt}(i,j)||}{\sum_{i,j} W_k(i,j)} \right), \quad (1)$$

where $\mathbf{I}^{gt}$ denotes ground truth grid pixel values, $\mathbf{I}^{pd}$ denotes predicted grid pixel values, and W denotes the Boolean padded mask of the UV map. i, j are indices into the image width and height dimensions, and k is index to the garment piece (e.g., front or back of the T-shirt). One can use different norms for this loss, and empirically we find that while L_1 leads to slightly better quantitative metrics than L_2, their visual qualities are roughly similar. Noting that normal vectors are important in capturing surface details, we include an additional loss term on the per-vertex normals:

$$\mathcal{L}_{\text{normal}} = \frac{1}{N_v} \sum_v \left(1 - \mathbf{n}_v^{pd}(\mathbf{I}^{pd}) \cdot \mathbf{n}_v^{gt} \right), \quad (2)$$

where we compute a predicted unit normal vector $\mathbf{n}_v^{pd}$ on each vertex v using the predicted grid pixel values $\mathbf{I}^{pd}$, and use the cosine distance to the ground truth unit normal vector $\mathbf{n}_v^{gt}$ as the loss metric. N_v is the number of vertices. The total loss is thus

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{image}} + \lambda \mathcal{L}_{\text{normal}}, \quad (3)$$

where $\lambda = 0.01$ is used when $\mathcal{L}_{\text{normal}}$ is enabled. The models are implemented in PyTorch [PGC*17] and trained using the Adam optimizer [KB14] with 10^{-3} learning rate.

8. Experiments

8.1. Whole T-shirts

The best visual results we obtained were from models that used additional losses on the normals, see Figure 9. Figure 10 shows more examples in various poses from both the training and the test set

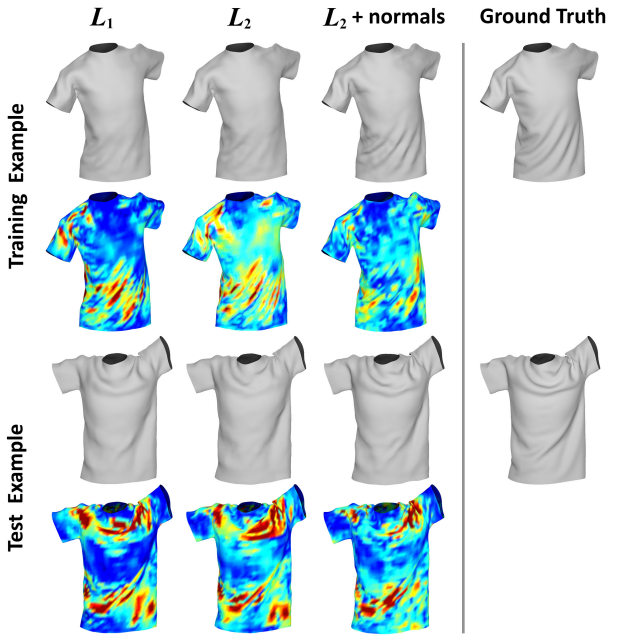

Figure 9: Network predictions/errors (blue = 0, red $\geqslant$ 1 cm) from models trained with different loss functions. While L_1 and L_2 loss on the pixels behave similarly, adding a loss term on the normals yields better visual quality. Left to right: L_1 on the pixels; L_2 on the pixels; L_2 on the pixels and cosine on the normals; ground truth.

using our best loss model. Figure 11 shows the average per cloth pixel model prediction errors on the training and test set. Unsurprisingly, the biggest errors occur near the sleeve seams and around the waist, where many wrinkles and folds form as one lifts their arms or bends. Finally, to see how well our model generalizes to new input data, we evaluated it on a motion capture sequence from [cmu], see Figure 12 and supplementary video.

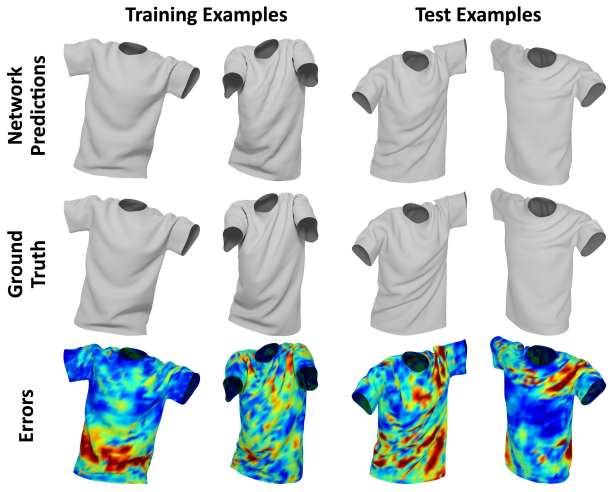

Figure 10: Network predictions and errors on training set and test set examples using our best loss model.

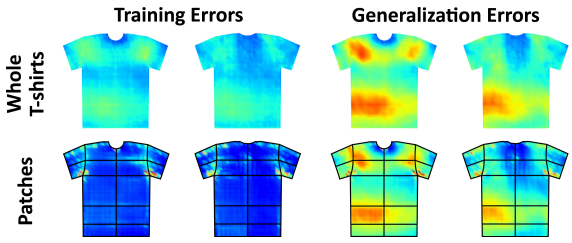

Figure 11: Dataset average per cloth pixel errors on the front/back side of the T-shirt. Top row: model trained on whole T-shirts (training/generalization error is 0.37 cm/0.51 cm). Bottom row: models trained on patches (training/generalization error is 0.20 cm/0.46 cm).

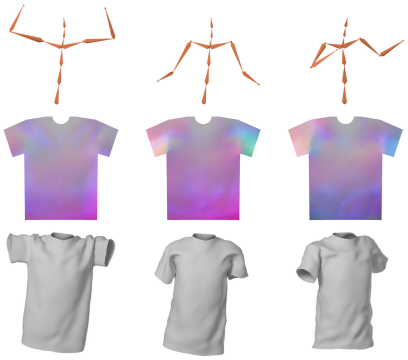

Figure 12: Evaluation on motion capture. Top: skeletal poses. Middle: predicted cloth images. Bottom: predicted cloth shapes.

8.2. Modified Body Shapes & Skinning Artifacts

The inability to obtain accurate unclothed body shapes is often seen as a real-world impediment to e-commerce clothing applications. However, our approach only uses an estimate of the unclothed form in order to pre-factor some nonlinearities out of the function to be learned. The network will learn whatever displacements are required to map from the body shape to the cloth, even if the body shape is erroneous. In order to demonstrate this, we modified our unclothed body shape making it too thick/thin in order to represent inaccuracies in the assumed body shape of the user. For each modified body shape, we use the same training data for cloth shapes noting that this merely changes the values of **dx** and thus the cloth image stored for each pose. As compared to the high variance in **dx** caused by folds and wrinkles, changing the body shape makes lower frequency modifications that are not too difficult for the network to learn. Surprisingly, the erroneously modified too thick/thin body shapes had almost no effect on the network's prediction ability indicating that our approach is robust to inaccuracies in the unclothed body shape. See Figure 13.

Whether using an accurate unclothed body shape or not, body skinning is not a solved problem; thus, we modified our skinning scheme to intentionally create artifacts using erroneous bone weights. Then, we trained the CNN as before noting that the cloth training images will be automatically modified whenever skinning artifacts appear. The erroneous skinning artifacts had almost no ef-

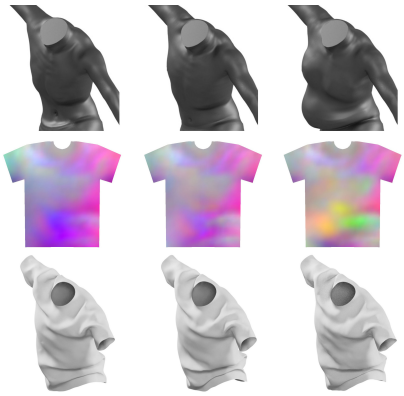

Figure 13: Training the network on unclothed body shapes that are too thin (left column) or too thick (right column) does not hinder its ability to predict cloth shapes, as compared to the ground truth (middle column). The cloth images (middle row) readily compensate for the incorrect unclothed body shape assumptions leading to similar cloth shapes (bottom row) in all three cases.

fect on the network's prediction ability indicating that our approach is robust to inaccuracies in the body skinning. See Figure 14.

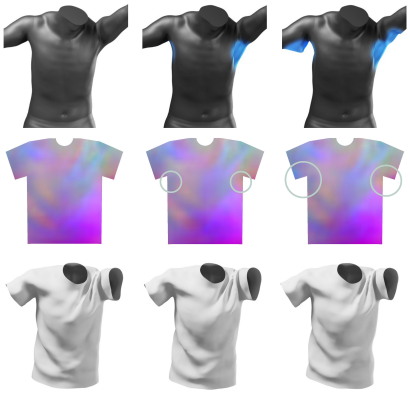

Figure 14: Training the network using a body skinning method that contains artifacts (shown in blue) does not hinder its ability to predict cloth shapes as compared to the ground truth (left column). The cloth images (middle row) readily compensate (see circled regions) for the skinning artifacts leading to similar cloth shapes (bottom row).

8.3. T-shirt Patches

As mentioned in Section 4.1, we can segment the cloth mesh into smaller semantically coherent pieces, and then train separate networks on these individual patches to achieve better results. Note that we also pad each patch to encourage smoothness on the seams and boundaries, similar to the whole T-shirt shown in Figure 2. Figure 11 shows that the models trained on the patches yield lower errors. See Figure 15 for visual comparison.

One can use a variety of methods to achieve visually continuous and smooth results across the patch boundaries. For example,

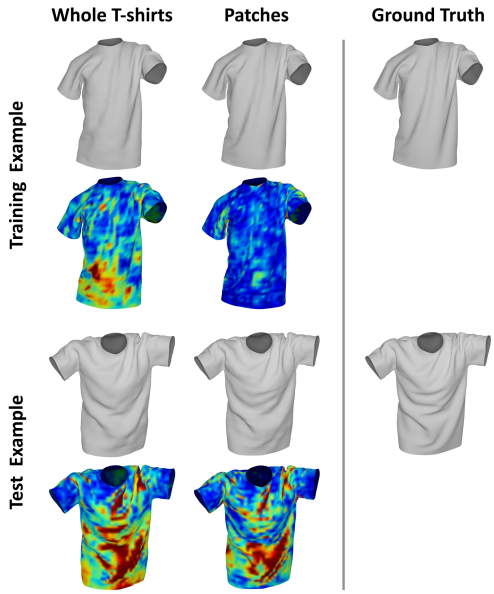

Figure 15: Comparison of network predictions/errors from model trained on whole T-shirts versus models trained on patches. The latter can better capture folds and wrinkles.

one can precompute the PCA bases of the whole mesh on the training samples, and then project the stitched mesh onto a subset of those bases. Since the simulation/captured data do not have kinks at patch boundaries, the PCA bases also will not have kinks at boundaries unless one gets into ultra-high frequency modes that represent noise; thus, reconstructing the network predicted results using a not too high number of PCA bases acts as a filter to remove discontinuities at patch boundaries. In our experiments, using 2048 components leads to the best filtering results, see Figure 16.

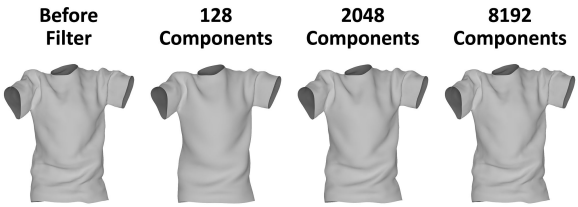

Figure 16: PCA filtering on a stitched mesh from predicted patches (an example from the test set).

8.4. Necktie

For generality, we also show a necktie example. Unlike T-shirts which tend to stay relatively close to a person's body, neckties are loose and can drift far from the body (similar to skirts/dresses). They exhibit much larger deformation as the body moves; the maximum per-vertex offset value can be over 50 centimeters in our dataset. See Figure 17, and supplementary F.

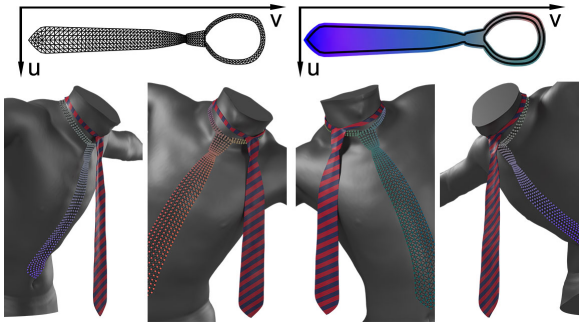

Figure 17: Top left: triangle mesh of necktie in pattern space. Top right: a necktie image. Bottom: network predictions of neckties in different poses (also, necktie pixels are shown embedded on the skinned body mesh).

9. Conclusion and Future Work

We proposed an approach to inferring cloth shape from body pose utilizing a procedural skinning pre-process to factor out significant nonlinearities due to joint rotation and skin deformation thus reducing demands on the neural network. Subsequently, we recast the prediction of vertex positions into the prediction of images in order to directly use well-studied CNNs. Importantly, we showed that our approach removes the need to ascertain accurate unclothed body shapes and is robust enough to compensate for inaccuracies in the prediction of body deformations.

For future work, we would like to leverage real-world captured cloth data and generalize our approach to a larger variety of garment types and materials as well as body types. We would also like to explore alternative network architectures, loss functions, and training schemes to enhance the visual quality of the predictions, and reach the level of details achieved in the state-of-the-art traditional physics-based 3D cloth simulation and capture methods. In addition, while our evaluation on the motion capture sequence already appears quite smooth in time, we would like to experiment with techniques such as 3D CNNs and recurrent neural networks to achieve better temporal coherency in dynamic sequences (e.g., swinging a dress).

Acknowledgements

Research supported in part by ONR N000014-13-1-0346, ONR N00014-17-1-2174, ARL AHPCRC W911NF-07-0027, and generous gifts from Amazon and Toyota. In addition, we would like to thank Radek Grzeszczuk for initiating conversations with Amazon and those interested in cloth there, Andrew Ng for many fruitful discussions on cloth for e-commerce, and both Reza and Behzad at ONR for supporting our efforts into machine learning. Also, we greatly appreciate the remarkable things that Jen-Hsun Huang (Nvidia) has done for both computer graphics and machine learning; this paper in particular was motivated by and enabled by a combination of the two (and inspirations from chatting with him personally). NJ was supported by a Stanford Graduate Fellowship, YZ was supported by a Stanford School of Engineering Fellowship, and ZG was supported by a VMWare Fellowship. NJ would also like to personally thank a number of people who helped contribute to our broader efforts on data-driven cloth, including Davis Rempe, Haotian Zhang, Lucy Hua, Zhengping Zhou, Daniel Do, and Alice Zhao.

References

[ACP02] ALLEN B., CURLESS B., POPOVIĆ Z.: Articulated body deformation from range scan data. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques* (2002), SIGGRAPH '02, ACM, pp. 612–619. 2

[Art] ARTEC: Artec 3D. https://www.artec3d.com/. 4

[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. 1, 2

[BBL*16] BRONSTEIN M. M., BRUNA J., LECUN Y., SZLAM A., VANDERGHEYNST P.: Geometric deep learning: going beyond euclidean data. *arXiv:1611.08097* (2016). 3

[BFA02] BRIDSON R., FEDKIW R., ANDERSON J.: Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. 21*, 3 (July 2002), 594–603. 1

[Ble18] BLENDER ONLINE COMMUNITY: *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2018. URL: http://www.blender.org. 4

[Bli78] BLINN J. F.: Simulation of wrinkled surfaces. In *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1978), SIGGRAPH '78, ACM, pp. 286–292. 3

[BMF03] BRIDSON R., MARINO S., FEDKIW R.: Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2003), SCA '03, Eurographics Association, pp. 28–36. 1

[BN76] BLINN J. F., NEWELL M. E.: Texture and reflection in computer generated images. *Commun. ACM 19*, 10 (Oct. 1976), 542–547. 3

[BODO18] BAILEY S. W., OTTE D., DILORENZO P., O'BRIEN J. F.: Fast and deep deformation approximations. *ACM Trans. Graph. 37*, 4 (July 2018), 119:1–119:12. 2, 5

[BSCB00] BERTALMIO M., SAPIRO G., CASELLES V., BALLESTER C.: Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2000), SIGGRAPH '00, pp. 417–424. 4

[BWK03] BARAFF D., WITKIN A., KASS M.: Untangling cloth. In *ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), SIGGRAPH '03, ACM, pp. 862–870. 1

[BZSL14] BRUNA J., ZAREMBA W., SZLAM A., LECUN Y.: Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014* (2014). 3

[Cat74] CATMULL E. E.: *A Subdivision Algorithm for Computer Display of Curved Surfaces.* PhD thesis, 1974. AAI7504786. 3

[cmu] Cmu graphics lab motion capture database. http://mocap.cs.cmu.edu/. 6

[Coo84] COOK R. L.: Shade trees. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1984), SIGGRAPH '84, ACM, pp. 223–231. 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. 1

[dASTH10] DE AGUIAR E., SIGAL L., TREUILLE A., HODGINS J. K.: Stable spaces for real-time clothing. In *ACM SIGGRAPH 2010 Papers* (New York, NY, USA, 2010), SIGGRAPH '10, ACM, pp. 106:1–106:9. 1, 2

[DBV16] DEFFERRARD M., BRESSON X., VANDERGHEYNST P.: Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (USA, 2016), NIPS'16, pp. 3844–3852. 3

[DDO*17] DANECEK R., DIBRA E., ÖZTIRELI A. C., ZIEGLER R., GROSS M.: Deepgarment : 3d garment shape estimation from a single image. *Computer Graphics Forum (Proc. Eurographics)*, 2 (2017). 2

[FvDFH96] FOLEY J. D., VAN DAM A., FEINER S. K., HUGHES J. F.: *Computer Graphics (2Nd Ed. In C): Principles and Practice*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1996. 3

[FWS*18] FENG Y., WU F., SHAO X., WANG Y., ZHOU X.: Joint 3d face reconstruction and dense alignment with position map regression network. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV* (2018), pp. 557–574. 2, 5

[GCS*19] GUNDOGDU E., CONSTANTIN V., SEIFODDINI A., DANG M., SALZMANN M., FUA P.: Garnet: A two-stream network for fast and accurate 3d cloth draping. In *IEEE International Conference on Computer Vision (ICCV)* (oct 2019), IEEE. 2

[GGH02] GU X., GORTLER S. J., HOPPE H.: Geometry images. SIGGRAPH '02, pp. 355–361. 3

[GJF19] GENG Z., JOHNSON D., FEDKIW R.: Coercing machine learning to output physically accurate results. *Journal of Computational Physics* (2019), 109099. 2

[GPAM*14] GOODFELLOW I., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 2672–2680. 4

[GRH*12] GUAN P., REISS L., HIRSHBERG D., WEISS A., BLACK M. J.: DRAPE: DRessing Any PErson. *ACM Trans. on Graphics (Proc. SIGGRAPH) 31*, 4 (July 2012), 35:1–35:10. 1, 2

[HBL15] HENAFF M., BRUNA J., LECUN Y.: Deep convolutional networks on graph-structured data. *arXiv:1506.05163* (2015). 3

[HE09] HILSMANN A., EISERT P.: Tracking and retexturing cloth for real-time virtual clothing applications. In *Computer Vision/Computer Graphics Collaboration Techniques* (2009), Springer Berlin Heidelberg, pp. 94–105. 1

[Hec86] HECKBERT P. S.: Survey of texture mapping. *IEEE Computer Graphics and Applications 6*, 11 (Nov 1986), 56–67. 2, 3

[HLRB12] HIRSHBERG D. A., LOPER M., RACHLIN E., BLACK M. J.: Coregistration: Simultaneous alignment and modeling of articulated 3d shape. In *Computer Vision – ECCV 2012* (2012), Springer Berlin Heidelberg, pp. 242–255. 2

[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

[HWW*18] HAN X., WU Z., WU Z., YU R., DAVIS L. S.: Viton: An image-based virtual try-on network. In *CVPR* (2018). 1

[HZRS15] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. *arXiv:1512.03385* (2015). 1

[JS11] JACOBSON A., SORKINE O.: Stretchable and twistable bones for skeletal shape deformation. In *Proceedings of the 2011 SIGGRAPH Asia Conference* (2011), SA '11, ACM, pp. 165:1–165:8. 2

[JSS*15] JIANG C., SCHROEDER C., SELLE A., TERAN J., STOMAKHIN A.: The affine particle-in-cell method. *ACM Trans. Graph. 34*, 4 (July 2015), 51:1–51:10. 3

[KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *CoRR* (2014). 5

[KCMF12] KIM T.-Y., CHENTANEZ N., MÜLLER-FISCHER M.: Long range attachments - a method to simulate inextensible clothing in computer games. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2012), SCA '12, Eurographics Association, pp. 305–310. 1

[KCvO07] KAVAN L., COLLINS S., ŽÁRA J., O'SULLIVAN C.: Skinning with dual quaternions. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games* (2007), I3D '07, ACM, pp. 39–46. 1, 2

[KGBS11] KAVAN L., GERSZEWSKI D., BARGTEIL A. W., SLOAN P.-P.: Physics-inspired upsampling for cloth simulation in games. In *ACM SIGGRAPH 2011 Papers* (New York, NY, USA, 2011), SIGGRAPH '11, ACM, pp. 93:1–93:10. 1

[KJP02] KRY P. G., JAMES D. L., PAI D. K.: Eigenskin: Real time large deformation character skinning in hardware. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2002), SCA '02, ACM, pp. 153–159. 2

[KKN*13] KIM D., KOH W., NARAIN R., FATAHALIAN K., TREUILLE A., O'BRIEN J. F.: Near-exhaustive precomputation of secondary cloth effects. *ACM Trans. Graph. 32*, 4 (July 2013), 87:1–87:8. 1, 2

[KM04] KURIHARA T., MIYATA N.: Modeling deformable human hands from medical images. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), SCA '04, Eurographics Association, pp. 355–363. 2

[KSH12] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 1097–1105. 1

[Kv05] KAVAN L., ŽÁRA J.: Spherical blend skinning: A real-time deformation of articulated models. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games* (2005), I3D '05, ACM, pp. 9–16. 1, 2

[Lan98] LANDER J.: Skin them bones: Game programming for the web generation. *Game Developer Magazine* (May 1998). 2

[LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (2000), SIGGRAPH '00, pp. 165–172. 2

[LCT18] LÄHNER Z., CREMERS D., TUNG T.: Deepwrinkles: Accurate and realistic clothing modeling. In *Computer Vision – ECCV 2018* (2018), pp. 698–715. 2, 3

[LH16] LE B. H., HODGINS J. K.: Real-time skeletal skinning with optimized centers of rotation. *ACM Trans. Graph. 35*, 4 (July 2016), 37:1–37:10. 2

[LLYY17] LI Y., LIU S., YANG J., YANG M.-H.: Generative face completion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017). 4

[LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 34*, 6 (Oct. 2015), 248:1–248:16. 1, 2

[LSD15] LONG J., SHELHAMER E., DARRELL T.: Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015). 1

[Mar97] MARGOLIN L.: Introduction to "an arbitrary lagrangian-eulerian computing method for all flow speeds". *J. Comput. Phys. 135*, 2 (Aug. 1997), 198–202. 3

[MBBV15] MASCI J., BOSCAINI D., BRONSTEIN M. M., VANDERGHEYNST P.: Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop (ICCVW)* (2015), ICCVW '15, IEEE Computer Society, pp. 832–840. 3

[MC10] MÜLLER M., CHENTANEZ N.: Wrinkle meshes. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar Germany, Germany, 2010), SCA '10, Eurographics Association, pp. 85–92. 1

[MDRW14] MANCEWICZ J., DERKSEN M., RIJPKEMA H., WILSON C.: Delta mush: Smoothing deformations while preserving detail. pp. 7–11. 2

[MHHR07] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position based dynamics. *J. Vis. Comun. Image Represent. 18*, 2 (Apr. 2007), 109–118. 1

[MTCSP04] MAGNENAT-THALMANN N., CORDIER F., SEO H., PAPAGIANAKIS G.: Modeling of bodies and clothes for virtual environments. In *2004 International Conference on Cyberworlds* (Nov 2004), pp. 201–208. 1

[MTLT88] MAGNENAT-THALMANN N., LAPERRIÈRE R., THALMANN D.: Joint-dependent local deformations for hand animation and object grasping. In *Proceedings on Graphics Interface '88* (1988), pp. 26–33. 1, 2

[NH14] NEOPHYTOU A., HILTON A.: A layered model of human body and garment deformation. In *Proceedings of the 2014 2Nd International Conference on 3D Vision - Volume 01* (Washington, DC, USA, 2014), 3DV '14, IEEE Computer Society, pp. 171–178. 1, 2

[PGC*17] PASZKE A., GROSS S., CHINTALA S., CHANAN G., YANG E., DEVITO Z., LIN Z., DESMAISON A., ANTIGA L., LERER A.: Automatic differentiation in pytorch. 5

[Phy] PHYSBAM: Physbam: physically based animation. http://physbam.stanford.edu/. 4

[PLPM20] PATEL C., LIAO Z., PONS-MOLL G.: Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (jun 2020), IEEE. 1

[PMPHB17] PONS-MOLL G., PUJADES S., HU S., BLACK M. J.: Clothcap: Seamless 4d clothing capture and retargeting. *ACM Trans. Graph. 36*, 4 (July 2017), 73:1–73:15. 1, 2

[RDAHT15] ROBERTINI N., DE AGUIAR E., HELTEN T., THEOBALT C.: Efficient multi-view performance capture of fine-scale surface detail. In *Proceedings - 2014 International Conference on 3D Vision, 3DV 2014* (02 2015), pp. 5–12. 1

[RHGS17] REN S., HE K., GIRSHICK R., SUN J.: Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence 39*, 6 (June 2017), 1137–1149. 1

[RPB15] RONNEBERGER O., P.FISCHER, BROX T.: U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2015), vol. 9351 of *LNCS*, Springer, pp. 234–241. 1

[SBR16] SINHA A., BAI J., RAMANI K.: Deep learning 3d shape surfaces using geometry images. In *ECCV* (2016). 3

[Set99] SETHIAN J.: *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 1999. 5

[SOC19] SANTESTEBAN I., OTADUY M. A., CASAS D.: Learning-based animation of clothing for virtual try-on. *Comput. Graph. Forum 38* (2019), 355–366. 1, 2

[SPR07] SHEFFER A., PRAUN E., ROSE K.: Mesh parameterization methods and their applications. *Foundations and Trends in Computer Graphics and Vision 2*, 2 (2007), 105–171. 2

[SUHR17] SINHA A., UNMESH A., HUANG Q.-X., RAMANI K.: Surfnet: Generating 3d shape surfaces using deep residual networks. In *CVPR* (2017), pp. 791–800. 3

[SZ14] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. In *arXiv 1409.1556*. 09 2014. 1

[TGLX18] TAN Q., GAO L., LAI Y.-K., XIA S.: Variational autoencoders for deforming 3d mesh models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018). 3

[WCPM18] WANG T. Y., CEYLAN D., POPOVIC J., MITRA N. J.: Learning a shared shape space for multimodal garment design. *ACM Trans. Graph. 37*, 6 (2018), 1:1–1:14. 2

[WHRO10] WANG H., HECHT F., RAMAMOORTHI R., O'BRIEN J. F.: Example-based wrinkle synthesis for clothing animation. In *ACM SIGGRAPH 2010 Papers* (New York, NY, USA, 2010), SIGGRAPH '10, ACM, pp. 107:1–107:8. 1, 2

[WJG*20] WU J., JIN Y., GENG Z., ZHOU H., FEDKIW R.: Recovering geometric information with learned texture perturbations. *arXiv:2001.07253* (2020). 2

[XUC*14] XU W., UMENTANI N., CHAO Q., MAO J., JIN X., TONG X.: Sensitivity-optimized rigging for example-based real-time clothing synthesis. *ACM Trans. Graph. 33*, 4 (July 2014), 107:1–107:11. 1, 2

[YFHWW18] YANG J., FRANCO J.-S., HETROY-WHEELER F., WUHRER S.: Analyzing clothing layer deformation statistics of 3d human motions. In *The European Conference on Computer Vision (ECCV)* (September 2018). 1, 2, 5

[YLY*18] YU J., LIN Z., YANG J., SHEN X., LU X., HUANG T. S.: Generative image inpainting with contextual attention. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018). 4