Point-Based Modeling of Human Clothing

Ilya Zakharkin [1,2*], Kirill Mazur [1*], Artur Grigorev [1], Victor Lempitsky [1,2]

[1] Samsung AI Center, Moscow
[2] Skolkovo Institute of Science and Technology, Moscow

Figure 1: Our approach models the geometry of diverse clothing outfits using point clouds (bottom row; random point colors). The point clouds are obtained by passing the SMPL meshes (shown in grey) and latent outfit code vectors through a pretrained deep network. Additionally, our approach can model clothing appearance using *neural point-based graphics* (top row). The outfit appearance can be captured from a video sequence, while a single frame is sufficient for point-based geometric modeling.

Abstract

We propose a new approach to human clothing modeling based on point clouds. Within this approach, we learn a deep model that can predict point clouds of various outfits, for various human poses and for various human body shapes. Notably, outfits of various types and topologies can be handled by the same model. Using the learned model, we can infer geometry of new outfits from as little as a singe image, and perform outfit retargeting to new bodies in new poses. We complement our geometric model with appearance modeling that uses the point cloud geometry as a geometric scaffolding, and employs neural point-based graphics to capture outfit appearance from videos and to re-render the captured outfits. We validate both geometric modeling and appearance modeling aspects of the proposed approach against recently proposed methods, and establish the viability of point-based clothing modeling.

1. Introduction

Modeling realistic clothing worn by humans is a big part of the overarching task of realistic modeling of humans

in 3D. Its immediate practical applications include virtual clothing try-on as well as enhancing the realism of human avatars for telepresence systems. Modeling clothing is difficult since outfits have wide variations in geometry (including topological changes) and in appearance (including wide variability of textile patterns, prints, as well as complex cloth reflectance). In particular, modeling interaction between clothing outfits and human bodies is an especially daunting task.

In this work, we propose a new approach to modeling clothing (Figure 1). In our approach, the clothing geometry is modeled with a point-cloud. Using a recently introduced synthetic dataset [6] of simulated clothing, we learn a joint geometric model of diverse human clothing outfits. The model describes a particular outfit with a latent code vector (the *outfit code*). For a given outfit code and a given human body geometry (for which we use the most popular SMPL format [33]), a deep neural network (the *draping network*) then predicts the point cloud that approximates the outfit geometry draped over the body.

The key advantage of our model is its ability to reproduce diverse outfits with varying topology using a single latent space of outfit codes and a single draping network. This is made possible because of the choice of the point cloud representation and the use of topology-independent, point cloud-specific losses during the learning of the joint model. After learning, the model is capable of generalizing to new outfits, capturing their geometry from data, and to drape the acquired outfits over bodies of varying shapes and in new poses. With our model, acquiring the outfit geometry can be done from as little as a single image.

We extend our approach beyond geometry acquisition to include the appearance modeling. Here, we use the ideas of differentiable rendering [35, 49, 30] and neural point-based graphics [1, 38, 58]. Given a video sequence of an outfit worn by a person, we capture the photometric properties of the outfit using neural descriptors attached to points in the point cloud, and the parameters of a rendering (decoder) network. The fitting of the neural point descriptors and the rendering network (which capture the photometric properties) is performed jointly with the estimation of the outfit code (which captures the outfit geometry) within the same optimization process. After fitting, the outfit can be transferred and re-rendered in a realistic way over new bodies and in new poses.

In the experiments, we evaluate the ability of our geometric model to capture the deformable geometry of new outfits using point clouds. We further test the capability of our full approach to capture both outfit geometry and appearance from videos and to re-render the learned outfits to new targets. The experimental comparisons show the viability of the point-based approach to clothing modeling.

2. Related work on clothing modeling

Modeling clothing geometry. Many existing methods model clothing geometry using one or several pre-defined garment templates of fixed topology. DRAPE [13], which is one of the earlier works, learns from Physic-based simulation (PBS) and allows for pose and shape variation for each learned garment mesh. Newer works usually represent garment templates in the form of offsets (displacements) to SMPL [34] mesh. ClothCap [46] employs such a technique and captures more fine-grained details learned from the new dataset of 4D scans. DeepWrinkles [29] also addresses the problem of fine-grained wrinkles modeling with the use of normal maps generated by a conditional GAN. GarNet [15] incorporates two-stream architecture and makes it possible to simulate garment meshes at the level of realism that almost matches PBS, while being two orders of magnitude faster. TailorNet [44] follows the same SMPL-based template approach as [46, 7] but models the garment deformations as a function of pose, shape and style simultaneously (unlike the previous work). It also shows greater inference speed than [15]. The CAPE system [36] uses graph ConvNet-based generative shape model that enables to condition, sample, and preserve fine shape detail in 3D meshes.

Several other works recover clothing geometry simultaneously with the full body mesh from image data. BodyNet [55] and DeepHuman [64] are voxel-based methods that directly infer the volumetric dressed body shape from a single image. In SiCloPe [42] the authors use similar approach, but synthesize the silhouettes of the subjects in order to recover more details. HMR [25] utilizes SMPL body model to estimate pose and shape from an input image. Some approaches such as PIFu [51] and ARCH [19] employ end-to-end implicit functions for clothed human 3D reconstruction and are able to generalise to complex clothing and hair topology, while PIFuHD [52] recovers higher resolution 3D surface by using two-level architecture. MouldingHumans [11] predicts the final surface from estimated "visible" and "hidden" depth maps. MonoClothCap [59] demonstrates promising results in video-based temporally coherent dynamic clothing deformation modeling. Most recently, Yoon et al. [62] design relatively simple yet effective pipeline for template-based garment mesh retargeting.

Our geometric modeling differs from previous works through the use of a different representations (point clouds), which gives our approach topological flexibility, the ability to model clothing separately from the body, while also providing the geometric scaffold for appearance modeling with neural rendering.

Modeling clothing appearance. A large number of work focus on direct image-to-image transfer of clothing bypassing 3D modeling. Thus, [23, 16, 56, 60, 21] address the task

of transferring a desired clothing item onto the corresponding region of a person given their images. CAGAN [23] is one of the first works that proposed to utilize image-to-image conditional GAN to tackle this task. VITON [16] follows the idea of image generation and uses non-parametric geometric transform which makes all the procedure two-stage, similar to SwapNet [48] with the difference in the task statement and training data. CP-VTON [56] further improves upon [16] by incorporating a full learnable thin-plate spline transformation, followed by CP-VTON+ [40], LA-VITON [22], Ayush et al. [5] and ACGPN [60]. While the above-mentioned works rely on pre-trained human parsers and pose estimators, the recent work of Issenhuth er al. [21] achieves competitive image quality and significant speed-up by employing a teacher-student setting to distill the standard virtual try-on pipeline. The resulting student network does not invoke an expensive human parsing network at inference time. Very recently introduced VOGUE [31] train a pose-conditioned StyleGAN2 [27] and find the optimal combination of latent codes to produce high-quality try-on images.

Some methods make use of both 2D and 3D information for model training and inference. Cloth-VTON [39] employs 3D-based warping to realistically retarget a 2D clothing template. Pix2Surf [41] allows to digitally map the texture of online retail store clothing images to the 3D surface of virtual garment items enabling 3D virtual try-on in real-time. Other relevant research extend the scenario of single template cloth retargeting to multi-garment dressing with unpaired data [43], generating high-resolution fashion model images wearing custom outfits [61], or editing the style of a person in the input image [17].

In contrast to the referenced approaches to clothing appearance retargeting, ours uses explicit 3D geometric models, while not relying on individual templates of fixed topology. On the downside, our appearance modeling part requires video sequence, while some of the referenced works use one or few images.

Joint modeling of geometry and appearance. Octopus [2] and Multi-Garment Net (MGN) [7] recover the textured clothed body mesh based on the SMPL+D model. The latter method treats clothing meshes separately from the body mesh, which gives it the ability to transfer the outfit to another subject. Tex2Shape [4] proposes an interesting framework that turns the shape regression task into an image-to-image translation problem. In [53], a learning-based parametric generative model is introduced that can support any type of garment material, body shape, and most garment topologies. Very recently, StylePeople [20] approach integrates polygonal body mesh modeling with neural rendering, so that both clothing geometry and the texture are encoded in the neural texture [54]. Similarly to [20] our approach to appearance modeling also relies on neural rendering, however our handling of geometry is more explicit. In the experiments, we compare to [20] and observe the advantage of a more explicit geometric modeling, especially for loose clothing.

3. Method

We first discuss the point cloud draping model. The goal of this model is to capture the geometry of diverse human outfits draped over human bodies with diverse shapes and poses using point clouds. We propose a latent model for such point clouds that can be fitted to a single image or to more complete data. We then describe the combination of the point cloud draping with neural rendering that allows us to capture the appearance of outfits from videos.

3.1. Point cloud draping

Learning the model. We learn the model using generative latent optimization (GLO) [9]. We assume that the training set has a set of N outfits, and associate each outfit with d-dimensional vector z (the *outfit code*). We thus randomly initialize $\{z_1, \dots, z_N\}$, where $z_i \in Z \subseteq \mathbb{R}^d$ for all $i = 1, \dots, N$.

During training, for each outfit, we observe its shape for a diverse set of human poses. The target shapes are given by a set of geometries. In our case, we use synthetic CLOTH3D dataset [6] that provides shapes in the form of meshes of varying topology. In this dataset, each subject is wearing an outfit and performs a sequence of movements. For each outfit i for each frame j in the corresponding sequence, we sample points from the mesh of this outfit and obtain the point cloud $x_i^j \in X$, where X denotes the space of point clouds of a fixed size (8192 is used in our experiments). We denote the length of the training sequence of the i-th outfit as P_i. We also assume that the body mesh $s_i^j \in S$ is given, and in our experiments we work with the SMPL [33] mesh format (thus S denotes the space of SMPL meshes for varying body shape parameters and body pose parameters). Putting it all together, we obtain the dataset $\{(z_i, s_i^j, x_i^j)\}_{i=1..N,\, j=1..P_i}$ of outfit codes, SMPL meshes, and clothing point clouds.

As our goal is to learn to predict the geometry in new poses and for new body shapes, we introduce the *draping function* $G_\theta : Z \times S \to X$ that maps the latent code and the SMPL mesh (characterizing the naked body) to the outfit point cloud. Here, θ denotes the learnable parameters of the function. We then perform learning by the optimization of the following objective:

$$\min_{\substack{\theta \in \Theta \\ \{z_1, \dots, z_N\}}} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{P_i} \sum_{j=1}^{P_i} L_{\text{3D}} \left(G_\theta(z_i, s_i^j),\, x_i^j \right) \qquad (1)$$

In (1), the objective is the mean reconstruction loss for the

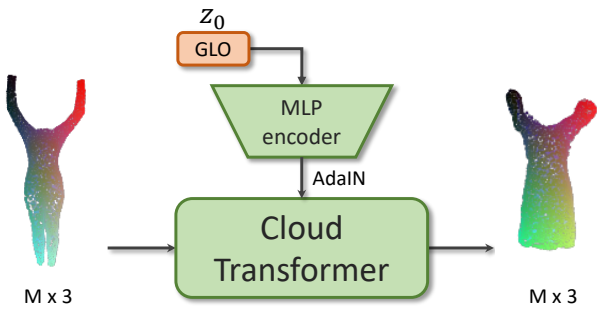

Figure 2: Our draping networks morphs the body point cloud (left) and the outfit code (top) into the outfit point cloud that is adapted to the body pose and the body shape.

training point clouds over the training set. The loss L_{3D} is thus the 3D reconstruction loss. In our experiments, we use the approximate Earth Mover's Distance [32]. Note, that as this loss measures the distance between point clouds and ignores all topological properties, our learning formulation is naturally suitable for learning outfits of diverse topology.

We perform optimization jointly over the parameters of our draping function G_θ and over the latent outfit code z_i for all $i = 1, \ldots, N$. Following [9], to regularize the process, we clip the outfit codes to the unit ball during optimization. The optimization process thus establishes the outfit latent code space and the parameters of the draping function.

Draping network. We implement the draping function $G_\theta(z, s)$ as a neural network that takes the SMPL mesh s and transforms this point cloud into the outfit point cloud. Over the last years, point clouds have become (almost) first-class citizens in the deep learning world, as a number of architectures that can input and/or output point clouds and operate on them have been proposed. In our work, we use the recently introduced *Cloud Transformer* architecture [37] due to its excellent results across a range of diverse tasks.

The cloud transformer comprises of blocks, each of which sequentially rasterizes, convolves, and de-rasterizes the point cloud at the learned data-dependent positions. The cloud transformer thus deforms the input point cloud (derived from the SMPL mesh as discussed below) into the output point cloud x over a number of blocks. We use a simplified version of the cloud transformer with single-headed blocks to reduce the computational complexity and memory requirements. Otherwise, we follow the architecture of the generator suggested in [37] for image-based shape reconstruction, which in their case takes the point cloud (sampled from the unit sphere) and a vector (computed by the image encoding network) as an input and outputs the point cloud of the shape depicted in the image.

In our case, the input point cloud and the vector are different and correspond to the SMPL mesh and the outfit code respectively. More specifically, to input the SMPL mesh into the cloud transformer architecture, we first remove the parts of the mesh corresponding to the head, the feet and the hands. We then consider the remaining vertices as a point cloud. To densify this point cloud, we also add the midpoints of the SMPL mesh edges to this point cloud. The resulting point cloud (which is shaped by the SMPL mesh and reflects the change of pose and shape) is input into the cloud transformer.

Following [37], the latent outfit code z is input into the cloud transformer through AdaIn connections [18] that modulate the convolutional maps inside the rasterization-derasterization blocks. The particular weights and biases for each AdaIn connection are predicted from the latent code z via a perceptron, as is common for style-based generators [26]. We note that while we have obtained good results using the (simplified) cloud transformer architecture, other deep learning architectures that operate on point clouds (e.g. PointNet [47]) can be employed.

We also note that the morphing implemented by the draping network is strongly non-local (i.e. our model does not simply compute local vertex displacements), and is consistent across outfits and poses (Figure 3).

Estimating the outfit code. Once the draping network is pre-trained on a large synthetic dataset [6], we are able to model the geometry of a previously unseen outfit. The fitting can be done to a single or multiple images. For a single image, we optimize the outfit code z^* to match the segmentation mask of the outfit in the image.

In more detail, we predict the binary outfit mask by passing given RGB image through Graphonomy network [12] and combining all semantic masks that correspond to clothing. We also fit the SMPL mesh to the person in the image using the SMPLify approach [8]. We then minimize the 2D Chamfer loss between the outfit segmentation mask and the projection of the predicted point-cloud onto the image. The projection takes into account the occlusions of the outfit by the SMPL mesh (e.g. the back part of the outfit when seen from the front). In this case, the optimization is performed over the outfit code z^* while the parameters of the draping network remain fixed to avoid overfitting to a single image.

For complex outfits we observed instability in the optimization process, which often results in undesired local minima. To find a better optimum, we start from several random initializations $\{z_1^*, \ldots z_T^*\}$ independently (in our experiments, $T = 4$). After several optimization steps we take the average outfit vector $\bar{z} = \frac{1}{T}\sum_{t=1}^{T} z_t^*$ and then continue the optimization from $\bar{z}$ until convergence. We observed that this simple technique provides consistently accurate outfit codes. Typically we make 100 training steps while optimizing T hypothesis. After the averaging the op-

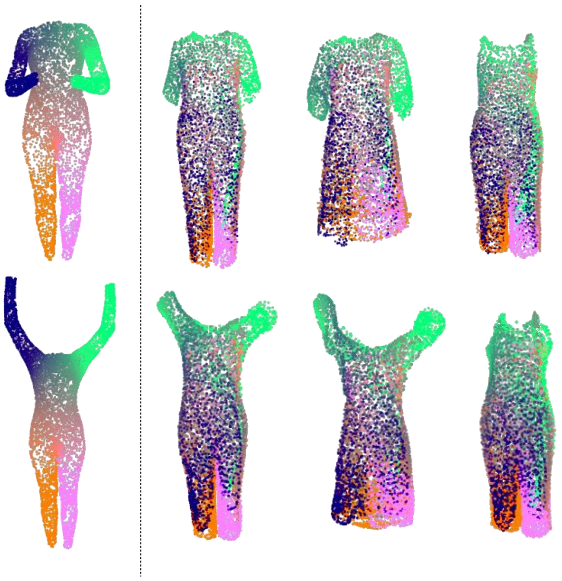

Figure 3: More color-coded results of the draping networks. Each row corresponds to a pose. The leftmost image shows the input to the draping network. The remaining columns correspond to three outfit codes. Color coding corresponds to spectral coordinates on the SMPL mesh surface. Color coding reveals that the draping transformation is noticeably non-local (i.e. the draping network does not simply compute local displacements). Also, color coding reveals correspondences between analogous parts of outfit point clouds across the draping network outputs.

timization takes $50-400$ steps depending on the complexity of the outfit's geometry.

3.2. Appearance modeling

Point-based rendering. Most applications of clothing modeling go beyond geometric modeling and require to model the appearance as well. Recently, it has been shown that point clouds provide good geometric scaffolds for neural rendering [1, 58, 38]. We follow the neural point-based graphics (NPBG) modeling approach [**?**] to add appearance modeling to our system (Figure 4).

Thus, when modeling the appearance of a certain outfit with the outfit code z, we attach p-dimensional latent appearance vectors $T = \{t[1], \dots, t[M]\}$ to each of the M points in the point cloud that models its geometry. We also introduce the rendering network R_ψ with learnable parameters ψ. To obtain the realistic rendering of the outfit given the body pose s and the camera pose C, we then first compute the point cloud $G_\theta(z, s)$, and then rasterize the point cloud over the image grid of resolution $W \times H$ using the camera parameters and the neural descriptor $t[m]$ as a pseudo-color of the m-th point. We concatenate the result of the rasterization, which is a p-channeled image,

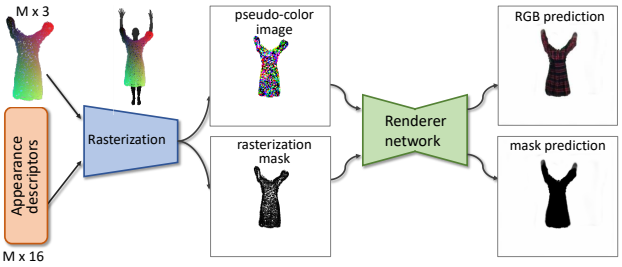

Figure 4: We use neural point-based graphics to model the appearance of an outfit. We thus learn the set of neural appearance descriptors and the renderer network that allow to translate the rasterization of the outfit point cloud into its realistic masked image (right).

with the rasterization masks, which indicates non-zero pixels, and then process (translate) them into the outfit RGB color image and the outfit mask (i.e. a four-channel image) using the rendering network R_ψ with learnable parameters ψ.

During the rasterization, we also take into account the SMPL mesh of the body and do not rasterize the points occluded by the body. For the rendering network we use a lightweight U-net network [50].

Video-based appearance capture. Our approach allows to capture the appearance of the outfit from video. To do that we perform two-stage optimization. In the first stage, the outfit code is optimized, minimizing the Chamfer loss between the point cloud projections and the segmentation masks, as described in the previous section. Then, we jointly optimize latent appearance vectors T, and the parameters of the rendering network ψ. For the second stage we use (1) the perceptual loss [24] between the masked video frame and the RGB image rendered by our model, and (2) the Dice loss between the segmentation mask and the rendering mask predicted by the rendering network.

Appearance optimization requires a video of a a person with whole surface of their body visible in at least one frame. In our experiments training sequences consist of 600 to 2800 frames for each person. The whole process takes roughly 10 hours on NVIDIA Tesla P40 GPU.

After the optimization, the acquired outfit model can be rendered for arbitrarily posed SMPL body shapes, providing RGB images and segmentation masks.

4. Experiments

We evaluate the geometric modeling and the appearance modeling within our approach and compare it to prior art.

Datasets. We evaluate both stages using two datasets of human videos. The *PeopleSnapsot* presented in [3] contains

24 videos of people in diverse clothes rotating in A-pose. In terms of clothing, it lacks examples of people wearing skirts and thus does not reveal the full advantage of our method. We also evaluate on a subset from *AzurePeople* dataset introduced in [20]. This subset contains videos of eight people in outfits of diverse complexity shot from 5 RGBD Kinect cameras. For both datasets we generate cloth segmentation masks with Graphonomy method [12] and SMPL meshes using SMPLify [8]. To run all approaches in our comparison, we also predict Openpose [10] keypoints, DensePose [14] UVI renders and SMPL-X [45] meshes.

In addition to the evaluation dataset described above, we also use the Cloth3D [6] dataset to train our geometric meta-model. The Cloth3D dataset has 11.3K garment elements of diverse geometry modeled as meshes draped over 8.5K SMPL bodies undergoing pose changes. The fitting uses physics-based simulation.

4.1. Details of the draping network

To build a solid geometric prior on clothing, our draping function G_θ is pre-trained on synthetic Cloth3D dataset.

We split it into train and validation parts, resulting in $N = 6475$ training video sequences. Since most of the consequent frames share common pose/clothes geometry, only each 10-th frame is considered for the training. As described in Sec. 3.1, we randomly initialize $\{z_1, \dots, z_N\}$, where $z_i \in Z \subseteq \mathbb{R}^d$ for each identity i in the dataset. In our experiments, we set the latent code dimensionality relatively low to d=8, in order to avoid overfitting during subsequent single-image shape fitting (as described in Sec. 3.1).

We feed the outfit codes z_i to an MLP encoder consisting of 5 fully-connected layers to obtain a 512-dimensional latent representation. Then it is passed to the AdaIn branch of the Cloud Transformer network. As of pose and body information, we feed an SMPL point cloud with hands, feet and head vertices removed, see Figure 1. The draping network outputs three-dimensional point clouds with 8.192 points in all our experiments. We choose approximate Earth Mover's Distance [32] as the loss function and optimize each GLO-vector and the draping network simultaneously using Adam [28].

While our pre-traning provides highly expressive priors on dresses and skirts, the diversity on tighter outfits is somewhat limited. Our hypothesis that this effect is mainly caused by a high bias towards jumpsuits in the Cloth3D tight clothing categories.

4.2. Recovering outfit geometry

In this series of experiments, we evaluate the ability of our method to recover the outfit geometry from a single photograph. We compare the result the following three methods:

	Ours v Tex2Shape	Ours v MGN	Ours v Octopus
PeopleSnapshot	38.1% vs 61.9%	50.9% vs 49.1%	47.8% vs 52.2%
AzurePeople	65.6% vs 34.3%	74.5% vs 25.5%	73.7% vs 26.3%

Table 1: Results of user study, in which the users compared the quality of 3D clothing geometry recovery (fitted to a single image). Our method is preferred on the AzurePeople dataset with looser clothing, while the previously proposed methods work better for tighter clothing of fixed topology.

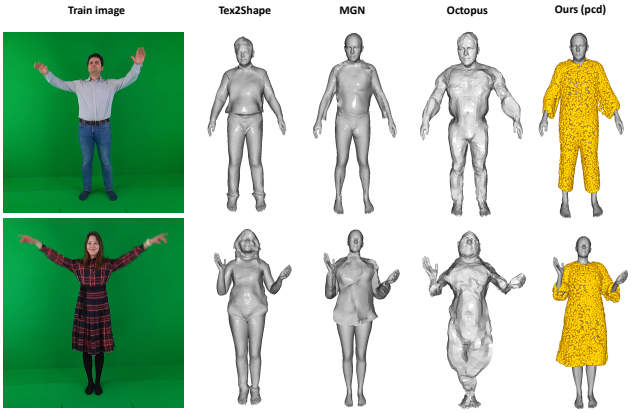

Figure 5: We show the predicted geometries in the validation poses fitted to a single frame (left). For our method (right) the geometry is defined by a point cloud (shown in yellow), while for Tex2Shape and MultiGarmentNet (MGN) the outputs are mesh based. Our method is able to reconstruct the dress, while other methods fail (bottom row). Note, our method is able to reconstruct a tighter outfit too (top row), though Tex2Shape with its displacement-based approach achieves a better result in this case.

1. The *Tex2Shape* method [4] that predicts offsets for vertices of SMPL mesh in texture space. It is ideally suited for the people Snapshot dataset, while less suitable to AzurePeople sequences with skirts and dresses.

2. The *Multi-garment net* approach [7] predicts clothing layered on top of SMPL body models. It proposes a virtual wardrobe of pre-fitted garments, and is also able to fit new outfits from single image.

3. Our point-based approach (*Ours*) that predicts point cloud garment geometry.

We note that the compared systems use different formats to recover clothing (point cloud, vertex offsets, meshes). Furthermore, they are actually solving slightly different problems, as our method and Multi-garment net recover clothing, while Tex2Shape recovers meshes that comprise clothing, body, and hair. All three systems, however, support retargeting to new poses. We therefore decided to evaluate

the relative performance of the three methods through a user study that assesses the realism of clothing retargeting.

We present the users with triplets of images, where the middle image shows the source photograph, while the side images show the results of two compared methods (in the form of shaded mesh renders for the same new pose). The result of such pairwise comparisons (user preferences) aggregated over $\sim$1.5k user comparisons are shown in Table 1. Our method is strongly preferred by the users in the case of AzurePeople dataset that contains skirts and dresses, while Tex2Shape and MGN are prefered on PeopleSnapshot dataset that has tighter clothing with fixed topology. Figure 5 shows typical cases, while the supplementary material provides more extensive qualitative comparisons. Note, in user study we paint our points in gray to exclude the coloring factor in user's choice.

For completeness, in the supplementary material, we provide additional comparisons of our method with the Octopus system [2], which is not ideally suited for reconstruction based on a single photograph.

4.3. Appearance modeling

We evaluate our appearance modeling pipeline against the *StylePeople* system [20] (the multiframe variant) that is the closest to ours in many ways. StylePeople fits a neural texture of the SMPL-X mesh alongside the rendering network using a video of a person using backpropagation. For comparison purposes we modify StylePeople to generate clothing masks along with rgb images and foreground segmentations. Both approaches are trained separately on each person from *AzurePeople* and *PeopleSnapshot* dataset. We then compare outfit images generated for holdout views in terms of three metrics that measure visual similarity to ground truth images, namely learned perceptual similarity (*LPIPS*) [63] distance, structural similarity (*SSIM*) [57] and its multiscale version (*MS-SSIM*).

The results of the comparison are shown in Table 2, while qualitative comparison is shown in Figure 7. In Figure 1, we show additional results for our methods. A number of clothing outfits of varying topology and type retargeted to new poses from both test datasets are shown. Finally, in Figure 6, we show examples of retargeting of outfit geometry and appearance to new body shapes within our approach.

5. Summary and Limitations

We have proposed a new approach to human clothing modeling based on point clouds. We have thus built a generative model for outfits of various shape and topology that allows us to capture the geometry of previously unseen outfits and to retarget it to new poses and body shapes. The topology-free property of our geometric representation (point clouds) is particularly suitable for modeling clothing due to wide variability of shapes and composition of outfits in real life. In addition to geometric modeling, we use the ideas of neural point-based graphics to capture clothing appearance, and to re-render full outfit models (geometry + appearance) in new poses on new bodies.

Our current approach to appearance modeling requires a video sequence in order to capture outfit appearance, which can be potentially addressed by expanding the generative modeling to the neural descriptors in a way similar to generative neural texture model from [20]. Also, the capabilities of our model to model cloth dynamics are severely limited, and to extend our model in that direction some integration of our approach with physics-based modeling (finite elements) could be useful. Finally, our model is limited to outfits similar to those represented in the Cloth3D dataset. Garments

Figure 6: Our approach can also retarget the geometry and the appearance to new body shapes. The appearance retargeting works well for uniformly colored clothes, though detailed prints (e.g. chest region in the bottom row) can get distorted.

	LPIPS↓	SSIM↑	MS-SSIM↑
	PeopleSnapshot		
Ours	**0.031**	**0.950**	**0.976**
StylePeople	0.0569	0.938	0.972
	AzurePeople		
Ours	**0.066**	**0.925**	0.937
StylePeople	0.0693	0.923	**0.946**

Table 2: Quantitative comparisons with the StylePeople system on the two test datasets using common image metrics. Our approach outperforms StylePeople in most metrics thanks to more accurate geometry modeling within our approach. This advantage is validated by visual inspection of quantitative results (Figure 7).

Figure 7: We compare the appearance retargeting results of our method to new poses unseen during fitting between our method and the StylePeople system (multi-shot variant), which uses the SMPL mesh as the underlying geometry and relies on neural rendering alone to "grow" loose clothes in renders. As expected, our system produces sharper results for looser clothes due to the use of more accurate geometric scaffolding. *Zoom-in is highly recommended.*

not present in the dataset (e.g. hats) can not be captured by our method.

References

[1] K. Aliev, A. Sevastopolsky, M. Kolos, D. Ulyanov, and V. S. Lempitsky. Neural point-based graphics. In A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, editors, *Proc. ECCV*, volume 12367 of *Lecture Notes in Computer Science*, pages 696–712. Springer, 2020. 2, 5

[2] T. Alldieck, M. Magnor, B. L. Bhatnagar, C. Theobalt, and G. Pons-Moll. Learning to reconstruct people in clothing from a single rgb camera. In *Proc. CVPR*, 2019. 3, 7

[3] T. Alldieck, M. Magnor, W. Xu, C. Theobalt, and G. Pons-Moll. Video based reconstruction of 3d people models. In *Proc. CVPR*, pages 8387–8397, 2018. 5

[4] T. Alldieck, G. Pons-Moll, C. Theobalt, and M. Magnor. Tex2shape: Detailed full human body geometry from a single image. In *Proc. 3DV*, 2019. 3, 6

[5] K. Ayush, S. Jandial, A. Chopra, and B. Krishnamurthy. Powering virtual try-on via auxiliary human segmentation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 3

[6] H. Bertiche, M. Madadi, and S. Escalera. Cloth3d: Clothed 3d humans. In *Proc. ECCV*, 2020. 2, 3, 4, 6

[7] B. L. Bhatnagar, G. Tiwari, C. Theobalt, and G. Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *Proc. ICCV*. IEEE, oct 2019. 2, 3, 6

[8] F. Bogo, A. Kanazawa, C. Lassner, P. V. Gehler, J. Romero, and M. J. Black. Keep it SMPL: automatic estimation of 3d human pose and shape from a single image. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Proc. ECCV*, volume 9909 of *Lecture Notes in Computer Science*, pages 561–578. Springer, 2016. 4, 6

[9] P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam. Optimizing the latent space of generative networks. In *Proc. ICML*, 2019. 3, 4

[10] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019. 6

[11] V. Gabeur, J.-S. Franco, X. Martin, C. Schmid, and G. Rogez. Moulding humans: Non-parametric 3d human shape estimation from single images. 2019. 2

[12] K. Gong, Y. Gao, X. Liang, X. Shen, M. Wang, and L. Lin. Graphonomy: Universal human parsing via graph transfer learning. In *Proc. CVPR*, 2019. 4, 6

[13] P. Guan, L. Reiss, D. Hirshberg, A. Weiss, and M. J. Black. DRAPE: DRessing Any PErson. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 31(4):35:1–35:10, July 2012. 2

[14] R. A. Güler, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proc. CVPR*, pages 7297–7306, 2018. 6

[15] E. Gundogdu, V. Constantin, A. Seifoddini, M. Dang, M. Salzmann, and P. Fua. Garnet: A two-stream network for fast and accurate 3d cloth draping. In *Proc. ICCV*, 2019. 2

[16] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis. Viton: An image-based virtual try-on network. In *Proc. CVPR*, 2018. 2, 3

[17] W.-L. Hsiao, I. Katsman, C.-Y. Wu, D. Parikh, and K. Grauman. Fashion++: Minimal edits for outfit improvement. In *Proc. ICCV*, 2019. 3

[18] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. ICCV*, 2017. 4

[19] Z. Huang, Y. Xu, C. Lassner, H. Li, and T. Tung. Arch: Animatable reconstruction of clothed humans. In *Proc. CVPR*, 2020. 2

[20] K. Iskakov, A. Grigorev, A. Ianina, R. Bashirov, I. Zakharkin, A. Vakhitov, and V. Lempitsky. Stylepeople: A generative model of fullbody human avatars. In *Proc. CVPR*, 2021. 3, 6, 7

[21] T. Issenhuth, J. Mary, and C. Calauzènes. Do not mask what you do not need to mask: a parser-free virtual try-on. In *Proc. ECCV*, 2020. 2, 3

[22] H. Jae Lee, R. Lee, M. Kang, M. Cho, and G. Park. Laviton: A network for looking-attractive virtual try-on. In *Proc. ICCV*, Oct 2019. 3

[23] N. Jetchev and U. Bergmann. The conditional analogy gan: Swapping fashion articles on people images. In *Proc. ICCV*, 2017. 2, 3

[24] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*, 2016. 5

[25] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *Proc. CVPR*, 2018. 2

[26] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, pages 4401–4410. Computer Vision Foundation / IEEE, 2019. 4

[27] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *Proc. CVPR*, 2020. 3

[28] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017. 6

[29] Z. Laehner, D. Cremers, and T. Tung. Deepwrinkles: Accurate and realistic clothing modeling. In *Proc. ECCV*, 2018. 2

[30] S. Laine, J. Hellsten, T. Karras, Y. Seol, J. Lehtinen, and T. Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020. 2

[31] K. M. Lewis, S. Varadharajan, and I. Kemelmacher-Shlizerman. Vogue: Try-on by stylegan interpolation optimization, 2021. 3

[32] M. Liu, L. Sheng, S. Yang, J. Shao, and S.-M. Hu. Morphing and sampling network for dense point cloud completion. *arXiv preprint arXiv:1912.00280*, 2019. 4, 6

[33] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 2, 3

[34] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 2

[35] M. M. Loper and M. J. Black. OpenDR: An approximate differentiable renderer. In *Proc. ECCV*, volume 8695 of *Lecture Notes in Computer Science*, pages 154–169. Springer International Publishing, Sept. 2014. 2

[36] Q. Ma, J. Yang, A. Ranjan, S. Pujades, G. Pons-Moll, S. Tang, and M. J. Black. Learning to Dress 3D People in Generative Clothing. In *Proc. CVPR*. 2

[37] K. Mazur and V. Lempitsky. Cloud transformers, 2020. 4

[38] M. Meshry, D. B. Goldman, S. Khamis, H. Hoppe, R. Pandey, N. Snavely, and R. Martin-Brualla. Neural rerendering in the wild. In *Proc. CVPR*, June 2019. 2, 5

[39] M. R. Minar and H. Ahn. Cloth-vton: Clothing three-dimensional reconstruction for hybrid image-based virtual try-on. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020. 3

[40] M. R. Minar, T. T. Tuan, H. Ahn, P. Rosin, and Y.-K. Lai. Cp-vton+: Clothing shape and texture preserving image-based virtual try-on. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 3

[41] A. Mir, T. Alldieck, and G. Pons-Moll. Learning to transfer texture from clothing images to 3d humans. In *Proc. CVPR*. IEEE, June 2020. 3

[42] R. Natsume, S. Saito, Z. Huang, W. Chen, C. Ma, H. Li, and S. Morishima. Siclope: Silhouette-based clothed people. In *Proc. CVPR*, 2019. 2

[43] A. Neuberger, E. Borenstein, B. Hilleli, E. Oks, and S. Alpert. Image based virtual try-on network from unpaired data. In *Proc. CVPR*, June 2020. 3

[44] C. Patel, Z. Liao, and G. Pons-Moll. Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In *Proc. CVPR*, 2020. 2

[45] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proc. CVPR*, pages 10975–10985, 2019. 6

[46] G. Pons-Moll, S. Pujades, S. Hu, and M. Black. Clothcap: Seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, 36(4), 2017. Two first authors contributed equally. 2

[47] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. CVPR*, pages 77–85. IEEE Computer Society, 2017. 4

[48] A. Raj, P. Sangkloy, H. Chang, J. Hays, D. Ceylan, and J. Lu. Swapnet: Image based garment transfer. *Proc. ECCV*, pages 679–695, 2018. 3

[49] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 2

[50] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]). 5

[51] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proc. ICCV*, 2019. 2

[52] S. Saito, T. Simon, J. Saragih, and H. Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proc. ICCV*, 2020. 2

[53] Y. Shen, J. Liang, and M. C. Lin. Gan-based garment generation using sewing pattern images. In *Proc. ECCV*, 2020. 3

[54] J. Thies, M. Zollhöfer, and M. Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 2019. 3

[55] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid. BodyNet: Volumetric inference of 3D human body shapes. In *Proc. ECCV*, 2018. 2

[56] B. Wang, H. Zheng, X. Liang, Y. Chen, L. Lin, and M. Yang. Toward characteristic-preserving image-based virtual try-on network. In *Proc. ECCV*, 2018. 2, 3

[57] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 7

[58] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson. SynSin: End-to-end view synthesis from a single image. In *Proc. CVPR*, 2020. 2, 5

[59] D. Xiang, F. Prada, C. Wu, and J. Hodgins. Monoclothcap: Towards temporally coherent clothing capture from monocular rgb video. In *Proc. 3DV*, 2020. 2

[60] H. Yang, R. Zhang, X. Guo, W. Liu, W. Zuo, and P. Luo. Towards photo-realistic virtual try-on by adaptively generating↔preserving image content. In *Proc. CVPR*, 2020. 2, 3

[61] G. Yildirim, N. Jetchev, R. Vollgraf, and U. Bergmann. Generating high-resolution fashion model images wearing custom outfits. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 3

[62] J. S. Yoon, K. Kim, J. Kautz, and H. S. Park. Neural 3d clothes retargeting from a single image, 2021. 2

[63] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. 7

[64] Z. Zheng, T. Yu, Y. Wei, Q. Dai, and Y. Liu. Deephuman: 3d human reconstruction from a single image. In *Proc. ICCV*, October 2019. 2