xCloth: Extracting Template-free Textured 3D Clothes from a Monocular Image

Astitva Srivastava
astitva.srivastava@research.iiit.ac.in
Center for Visual Information Technology, IIIT Hyderabad
Hyderabad, Telangana, India

Chandradeep Pokhariya
chandradeep.pokhariya@research.iiit.ac.in
Center for Visual Information Technology, IIIT Hyderabad
Hyderabad, Telangana, India

Sai Sagar Jinka
jinka.sagar@research.iiit.ac.in
Center for Visual Information Technology, IIIT Hyderabad
Hyderabad, Telangana, India

Avinash Sharma
asharma@iiit.ac.in
Center for Visual Information Technology, IIIT Hyderabad
Hyderabad, Telangana, India

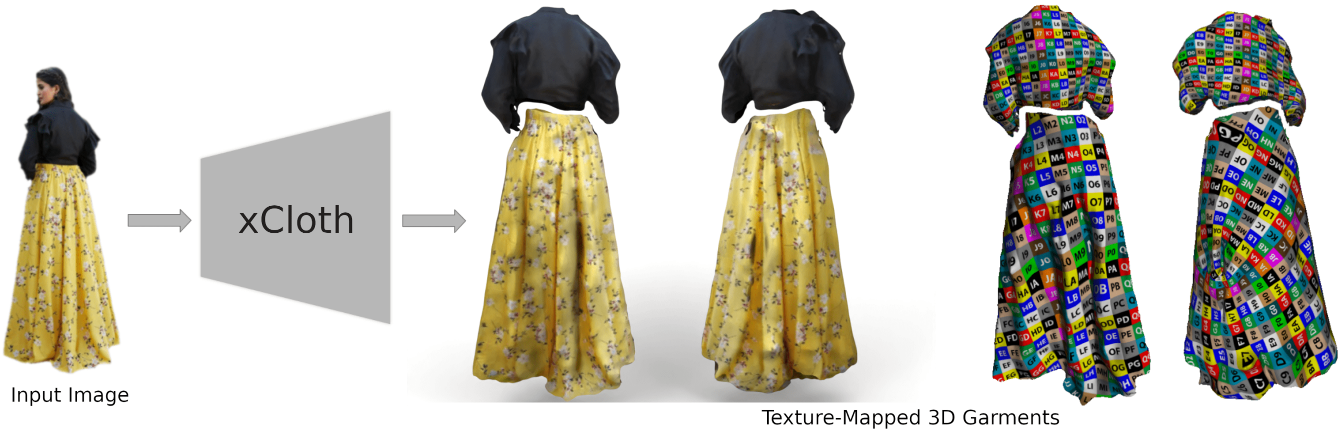

Figure 1: Proposed xCloth framework extracts high-fidelity template-free textured 3D garments from a monocular image.

ABSTRACT

Existing approaches for 3D garment reconstruction either assume a predefined template for the garment geometry (restricting them to fixed clothing styles) or yield vertex colored meshes (lacking high-frequency textural details). Our novel framework co-learns geometric and semantic information of garment surface from the input monocular image for template-free textured 3D garment digitization. More specifically, we propose to extend PeeledHuman representation to predict the pixel-aligned, layered depth and semantic maps to extract 3D garments. The layered representation is further exploited to UV parametrize the arbitrary surface of the extracted garment without any human intervention to form a UV atlas. The texture is then imparted on the UV atlas in a hybrid fashion by first projecting pixels from the input image to UV space for the visible region, followed by inpainting the occluded regions. Thus, we are able to digitize arbitrarily loose clothing styles while retaining high-frequency textural details from a monocular image. We achieve high-fidelity 3D garment reconstruction results on three publicly available datasets and generalization on internet images.

CCS CONCEPTS

• **Computing methodologies** → *Shape inference*; *Reconstruction*.

KEYWORDS

monocular, PeeledHuman, parametrization, texture maps, virtual try-on

ACM Reference Format:
Astitva Srivastava, Chandradeep Pokhariya, Sai Sagar Jinka, and Avinash Sharma. 2022. xCloth: Extracting Template-free Textured 3D Clothes from a Monocular Image. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548419

1 INTRODUCTION

The high-fidelity digitization of 3D garment(s) from a 2D image(s) is essential in achieving photorealism in a wide range of applications in computer vision, e.g., 3D virtual try-on, human digitization for AR/VR, realistic virtual avatar animation, etc. The goal

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548419

of the 3D digitization is to recover 3D surface geometry as well as high-frequency textural details of the garments with arbitrary styles/designs. Textured 3D digitization of garments remains notoriously tricky, as designing clothing is a work of art where a lot of creativity is involved at the designer's end. The lack of standardization in designing vocabulary (e.g., cloth panels) further makes the task challenging. Traditionally, expensive multi-view capture/scan setups were used for digitizing garments, but they can't scale up in fast-fashion scenarios [3, 7], owing to their cost, scan latency and volume of efforts.

The majority of the existing learning-based garment digitization methods [15, 17, 24] rely on the availability of predefined 3D template mesh for a specific clothing style, generally taken from popular parametric body models (e.g., SMPL [14]) or designed by an artist [6]. Specifically, one class of methods (e.g., [15, 17]) propose to transfer the texture from the RGB image to the UV map of the predefined template while retaining the fixed template geometry. The other class of methods (e.g., [4, 11, 26]) propose to deform the predefined template using cues from input RGB image in a learnable fashion but do not attempt to recover texture. MGN [4] proposed a hybrid approach to achieve the best of these two by learning to locally deform the template mesh using SMPL+D for recovering geometry and transfer texture from multiple images to SMPL UV atlas using [1] which is fixed for any garment style. However, these methods restrict the clothing styles to a set of predefined templates, which are very simplistic in nature and cannot model arbitrary clothing styles with high-frequency geometrical details (e.g., folds in long skirts). A recent attempt in [26] propose to modify the SMPL template by adaptable template using handle-based deformation and register it to non-parametric mesh recovered from OccNet [16] to accommodate loose clothing. Another very recent attempt [27] further extends this idea to incorporate learnable segmentation & garment-boundary field to improve the reconstruction quality. However, these methods do not attempt to recover textured or even vertex-colored meshes. A similar scenario exists in the related domain of clothed human body reconstruction, where all existing methods either recover 3D mesh surface sans texture or achieve a fix (SMPL mesh) texture map based reconstruction, which imposes tight clothing limitations.

A textured mesh not only retains high-quality appearance details that are crucial for a 3D virtual-try-on kind of application, but it can also be exploited for extended applications such as texture map super-resolution for enhanced detailing, appearance manipulation by texture swapping, etc. In regard to computational efficiency, texture representation is also memory efficient as high-frequency appearance details can be retained by rendering low-quality meshes (with less number of faces), as shown in Fig. 2. Existing template-based garment digitization methods assumes a predefined UV parametrized template for texture mapping. Various parameterization techniques are available (e.g., LSCM [13]) that can be used for template-free garment reconstruction. However, they need to find optimal seams (for partitioning the mesh), which often require either manual intervention or do not provide any control over the placements of seams.

In this paper, we propose a novel framework for template-free, textured 3D digitization of arbitrary garment styles from a monocular image. The proposed framework consists of three modules; where first, we predict the 3D geometry of the garment in the form of a sparse, non-parametric peeled representation (to handle self-occlusions) along with semantic information and surface normals in the form of semantic and normal peelmaps. The semantic labels provide supervision for extracting the cloths separately and also help in dealing with complex garment geometry and pose, while the normal maps help in recovering the high-frequency surface details. This yields a dense point-cloud representing the garment. Subsequently, we refine this point cloud to recover a mesh representation of the garment. Finally, we exploited the peeled representation in a novel way to automatically UV parametrize the extracted 3D garment mesh. More specifically, the peeled representation provides natural view-specific partitions of the 3D garment mesh; we propose to automatically infer seams by exploiting these partitions for UV parametrization. We recover the associated texture map by appropriately projecting the RGB peelmaps to the corresponding partitions. However, in many cases where the garment has high-frequency textural details, generative methods tend to yield blurred predictions and hence, as a remedy, we propose to use an existing planar structure-based inpainting method for the texture maps. We thoroughly evaluated the proposed framework, reported qualitative and quantitative results on the publicly available datasets as well as internet images and reported superior performance as compared to existing SOTA methods. We intend to release the proposed model in the public domain.

Please refer to the supplementary material for a detailed literature survey.

2 PROPOSED FRAMEWORK

The proposed xCloth framework is divided into three key modules. First, the input image is passed to the "3D Reconstruction" module, which extracts the 3D garment in the form of a point cloud. This point cloud contains missing regions, especially along the tangential boundaries, owing to the inherent limitation of peeled representation. The noisy point cloud is passed to the "Geometry Refinement" module, which deals with the missing regions and extracts a dense mesh out of it. Finally, the mesh is passed to the "Peeled Texture Mapping & In-painting" module, where the mesh is automatically (UV-) parametrized, followed by texture in-painting for the occluded regions producing a high-fidelity textured mapped 3D garment. We discuss each of these modules in detail below.

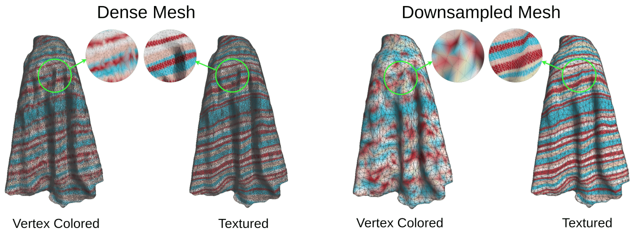

Figure 2: Advantage of textured over vertex-colored mesh.

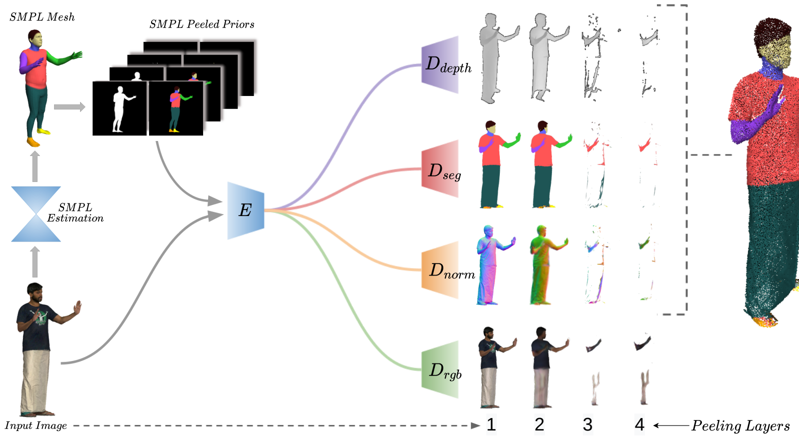

Figure 3: Outline of the proposed 3D Reconstruction Module. SMPL mesh is estimated from the input image which is then segmented and peeled to generate SMPL peeled priors. These priors are passed to the shared encoder E. The output of encoder is passed to the decoders D_{depth}, D_{seg}, D_{norm} & D_{rgb}, which produce depth, segmentation, normal and RGB peelmaps. The semantic point cloud is generated from depth, segmentation and normal peelmaps.

2.1 3D Reconstruction

The first task is to recover the 3D geometry of the garment from a monocular image. To represent the 3D geometry, we use PeeledHuman [12] representation which is a non-parametric, multi-layered encoding of 3D shapes, stored in the form of sparse images called peelmaps. To generate ground truth peelmaps, the mesh is placed in a virtual scene and a set of rays are emanated from the camera center through each pixel towards the mesh. The first set of ray-intersections with the mesh are recorded as the first layer depth peelmap and RGB peelmap, capturing visible surface information nearest to the camera. Subsequently, the rays are extended beyond the first intersection point (piercing through the intersecting surface) to hit the surface behind it. The corresponding depth and RGB values are recorded in the next layer peelmaps. We use total four layers of peeled representation for the 3D reconstruction module. We adopt the architecture proposed in [21], which is comprised of a shared encoder and predicts only RGB and Depth peelmaps. We extend the proposed architecture as follows. The shared encoder encodes the input image along with peeled SMPL (depth and body part segmentation) priors. This common encoding is fed to four decoder branches, namely, D_{depth}, D_{seg}, D_{norm} & D_{rgb}, which predicts depth peelmaps $\widehat{\mathcal{P}}^i_{depth}$, segmentation peelmaps $\widehat{\mathcal{P}}^i_{seg}$, normal peelmaps $\widehat{\mathcal{P}}^i_{norm}$ & RGB peelmaps $\widehat{\mathcal{P}}^i_{rgb}$, respectively.

We compute L1 loss on $\widehat{\mathcal{P}}^i_{depth}$ & $\widehat{\mathcal{P}}^i_{rgb}$, while L2 loss on $\widehat{\mathcal{P}}^i_{norm}$ w.r.t corresponding ground truth peelmaps $\mathcal{P}$, represented by following equations:

$$L_{depth} = \sum_{i=1}^{4} \left\| \widehat{\mathcal{P}}^i_{depth} - \mathcal{P}^i_{depth} \right\| \tag{1}$$

$$L_{rgb} = \sum_{i=1}^{4} \left\| \widehat{\mathcal{P}}^i_{rgb} - \mathcal{P}^i_{rgb} \right\| \tag{2}$$

$$L_{norm} = \sum_{i=1}^{4} (\widehat{\mathcal{P}}^i_{norm} - \mathcal{P}^i_{norm})^2 \tag{3}$$

In the case of segmentation peelmaps, we minimize the multi-class cross-entropy loss over the segmentation label classes, represented by the following equation:

$$L_{seg} = \sum_{c=1}^{N} \mathcal{P}^c_{seg} \log(\widehat{\mathcal{P}}^c_{seg}) \tag{4}$$

Here, N refers to the total number of semantic segmentation labels, which correspond with the segmentation classes present in the CIHP[8] dataset. Please refer to supplementary material for the ground truth semantic data generation process. We train the entire network in an end-to-end fashion, minimizing the following objective function:

$$L = \lambda_{depth} L_{depth} + \lambda_{seg} L_{seg} + \lambda_{norm} L_{norm} + \lambda_{rgb} L_{rgb} \tag{5}$$

The output of the 3D reconstruction module is the point cloud estimated from back-projected depth peelmaps, where each 3D point is assigned a segmentation label from the predicted semantic peelmaps, and point normals are assigned from the predicted normal peelmaps. The semantic labelling information is further used to extract each garment separately in the form of a point cloud. The predicted RGB peelmaps are later used during the texture-mapping.

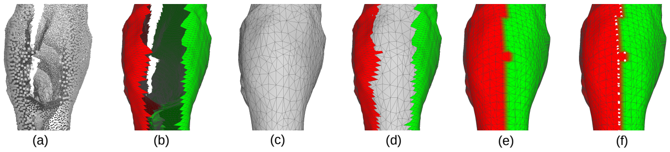

Figure 4: Geometry refinement & automatic seam estimation.

2.2 Geometry Refinement

The reconstruction module yields a dense point cloud representing the garment that has missing regions due to the inherent drawback of PeeledHuman representation, as shown in Figure 4(a). These missing regions are typically the regions that are tangential to the peeling camera rays (w.r.t. input view) and hence need to be filled to form a complete dense 3D surface of the garment. One common solution is to run the Poisson Surface Reconstruction (PSR) on top of the point cloud, which gives a watertight and hole-filled mesh. However, PSR also smoothens the geometrical details present in the back-projected depth peelmaps. Instead of naively running PSR, we first induce independent partial mesh (Figure 4(b)) for each of the depth peelmaps by exploiting the image grid structure. Additionally, for the missing region, we induce another partial mesh by sampling 3D points on the Poisson surface reconstruction of the point cloud. Finally, we merge these partial meshes to obtain a single refined mesh as in Figure 4(c). This helps in retaining the fine-grained surface details present in the predicted depth maps while filling holes & missing regions. An additional advantage of independent meshification of each peelmaps is that we can retain the association of each vertex in the final mesh with the respective peeling layer. This association information is subsequently used for automatic seam estimation for texture mapping in the next module.

2.3 Peeled Texture Mapping & In-painting

The proposed peeled texture mapping approach has three steps:

1. Automated Seam Estimation: Traditionally, seams are used to partition the mesh for obtaining non-overlapping UV parametrization. We propose a novel method to automatically estimate seams (without any manual intervention) by exploiting the partitioning provided by depth peelmaps. The previous module assigns each vertex of the refined mesh to a specific peeling layer, except for the vertices belonging to filled regions (Figure 4(d)). We assign these vertices to a peeling layer using nearest neighbor extrapolation. Thus, every vertex on the refined mesh is assigned to a peeling layer, as can be seen in Figure 4(e). Among them, all the boundary vertices define seams (Figure 4(f)) that are used to split the refined mesh into multiple partitions. The seam vertices are replicated across the adjacent partitions in order to avoid the artifacts near the seam boundary during rendering. Finally, each of the partition is UV parametrized separately.

2. Peeled Parametrization: We employ the Boundary-First Flattening (BFF) [23] parametrization technique separately on each individual partition to avoid any overlap. This gives us a UV parametrization for each partition with very minimal distortion. Instead of mapping all the vertices of the mesh to the same UV space, we now map the vertices of each partition of the mesh to the individual UV maps, thereby constructing a UV Atlas.

3. Texture Filling & In-painting: We use the information from RGB peelmaps for filling textures in the constructed UV atlas. Every parametrized partition has an associated RGB peelmap pixel aligned with it. In order to fill the UV map associated with this partition, we assign the RGB value for each pixel by projecting them from the UV space to the corresponding RGB peelmap. This enables filling a large part of the UV map. However, the pixels associated with the tangential regions are not visible in the corresponding RGB peelmap and remain unfilled. We propose to employ an off-the-shelf structure-based image inpainting solution [10] and fill the texture for these missing pixels independently for each UV map.

Additionally, we empirically observed that predicting high-quality texture for unseen areas is a tedious task for any generative model and tends to cause artifacts and over-smoothening, which also applies to the predicted RGB peelmaps. Thus, if the image has high-frequency textures, we propose to employ the Structure-Completion inpainting for the second or subsequent RGB peelmaps by directly extending the first UV map or extending a known patch either taken from the visible region of the first peel map (input image) or from the externally provided image of the underlying cloth. This results in a completely filled UV atlas.

3 EXPERIMENTS & RESULTS

3.1 Implementation Details

We employ a multi-branch encoder-decoder network as a part of our framework, which is trained in an end-to-end fashion. The network takes the input image concatenated with SMPL peeled priors in 512 × 512 resolution. The shared encoder consists of a convolutional layer and 2 downsampling layers which have 64, 128, and 256 kernels of size 7×7, 3×3 and 3×3, respectively. This is followed by ResNet blocks which take downsampled feature maps of size 128×128×256. The decoders consist of two upsampling layers followed by a convolutional layer, having the same kernel sizes as the shared encoder. For D_{depth} and D_{norm}, *sigmoid* activation function is used while for D_{rgb}, *tanh* activation function is used. We use the Adam optimizer with an exponentially decaying learning rate starting from 5×10^{-4}. The training takes around 18 hrs (for 20 epochs) on four Nvidia GTX 1080Ti GPUs with a batch size of 4 and hyperparameters λ_{depth}, λ_{seg}, λ_{norm} & λ_{rgb} are empirically set to 1.0, 0.1, 1.0 & 0.05.

3.2 Evaluation Metrics

Point-to-Surface (P2S) Distance: P2S measures the average L2 distance between each point to the surface for a given pair of point clouds and the surface. Thus, a lower value of P2S indicates higher fidelity geometric reconstruction w.r.t. ground truth surface.

Intersection Over Union (IOU): It is the area of overlap between the predicted and the ground truth segmentation labels divided by the area of union between them. Thus, this metric ranges from 0 to 1, and values closer to 1 are desired, indicating a higher overlap between predicted and ground truth labels.

Normal Reprojection Error (NRE)[22]: NRE is computed by estimating L2 error between rendered normal maps (from input view) for reconstructed and ground truth surfaces. Hence, a lower value is desired for NRE, indicating higher fidelity of reconstructed surface normals.

3.3 Datasets

We use following datasets for training & evaluation.

Digital Wardrobe (DW) [4]: The dataset consist of 356 real body scans with textured meshes in arbitrary, yet simple poses and include only five categories of tight clothing styles. The dataset also includes underlying SMPL body and segmentation labels. However, out of 365 meshes, only 96 are made publicly available. We train

	Topwear			Bottomwear		
Dataset	P2S ↓	IOU ↑	NRE ↓	P2S ↓	IOU ↑	NRE ↓
3DHumans	0.0087	0.82	0.089	0.0077	0.83	0.081
THUman2.0	0.0079	0.80	0.094	0.0072	0.74	0.085
DW	0.0091	0.91	0.088	0.0083	0.95	0.073

Table 1: Quantitative evaluation of our framework.

	Topwear		Bottomwear	
Dataset	P2S ↓	NRE ↓	P2S ↓	NRE ↓
3DHumans	0.0091	0.118	0.0082	0.122
THUman2.0	0.0088	0.182	0.0078	0.177
DW	0.0096	0.107	0.0089	0.097

Table 2: Ablation without normal peelmaps.

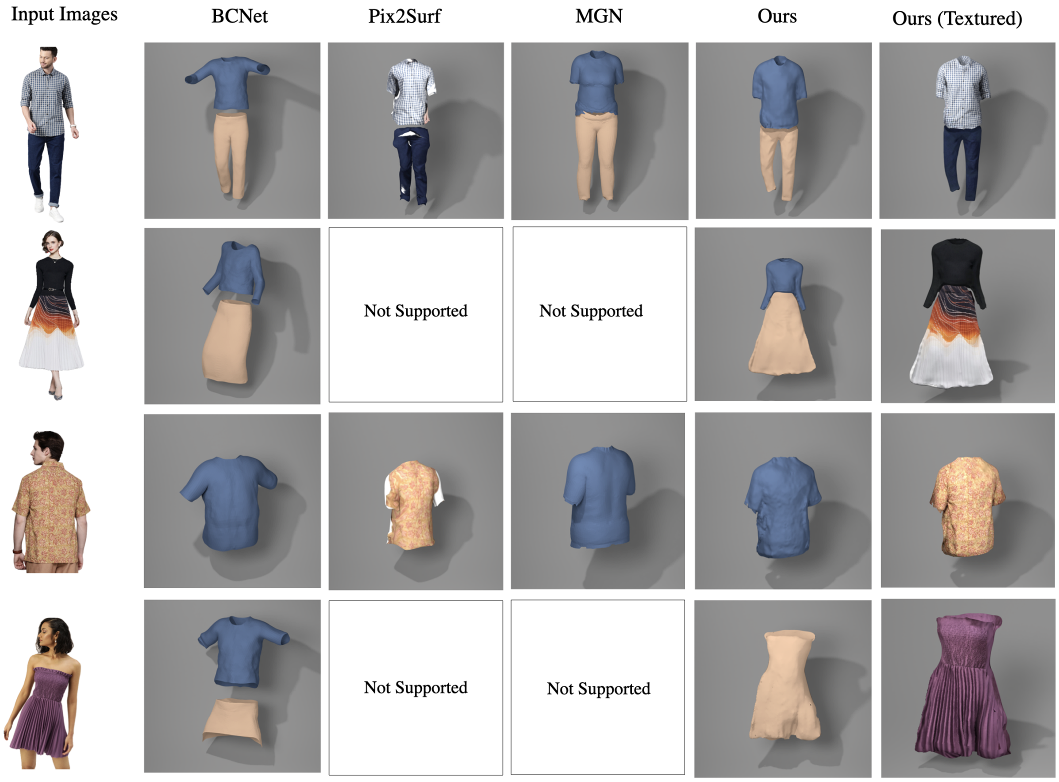

Figure 5: Qualitative comparison of xCloth (ours) with the existing SOTA methods on in-the-wild images.

our 3D reconstruction module on 80 out of 96 meshes and report evaluation metrics on the remaining 16 meshes.

THUman2.0 [25]: This has around 500 real body scans with arbitrary clothing and diverse poses. We register the SMPL to the provided scans and also curate segmentation labels using our scan-segmentation pipeline. We train our 3D reconstruction module on 400 meshes and test on remaining 100 meshes.

3DHumans [5, 21]: This recently introduced dataset contains 200 high-frequency textured scans of the South-Asian population in diverse poses with garments of arbitrary loose styles. We post-process this dataset in a similar fashion as THUman2.0. Around 170 meshes are used for training and 30 for testing.

We intend to release the curated segmentation labels for THUman2.0 and 3DHumans dataset with our scan-segmentation pipeline.

3.4 Quantitative Evaluation

We evaluate xCloth framework separately for two classes of garments (topwear & bottomwear). To evaluate the reconstruction error, we uniformly sample points on the extracted 3D garment mesh and compute P2S distance with the ground truth garment mesh. For evaluating the output of the segmentation branch, we report per-class IOU (i.e. topwear and bottomwear) for all segmentation peelmaps. Finally, to evaluate the quality of the extracted garment surface, we compute NRE separately for each class (ignoring the background). We report the quantitative results in Table 1. Here, we can infer that the DW dataset has lower NRE as opposed to the other two datasets. This can be attributed to the fact that it has meshes that largely lack fine-grained geometrical details in comparison to THUman2.0 and 3DHumans. On the other hand we observe that P2S distance reduces with larger training data size and thus our framework obtain higher P2S on the DW dataset as compare to other two datasets. We take a random test sample from

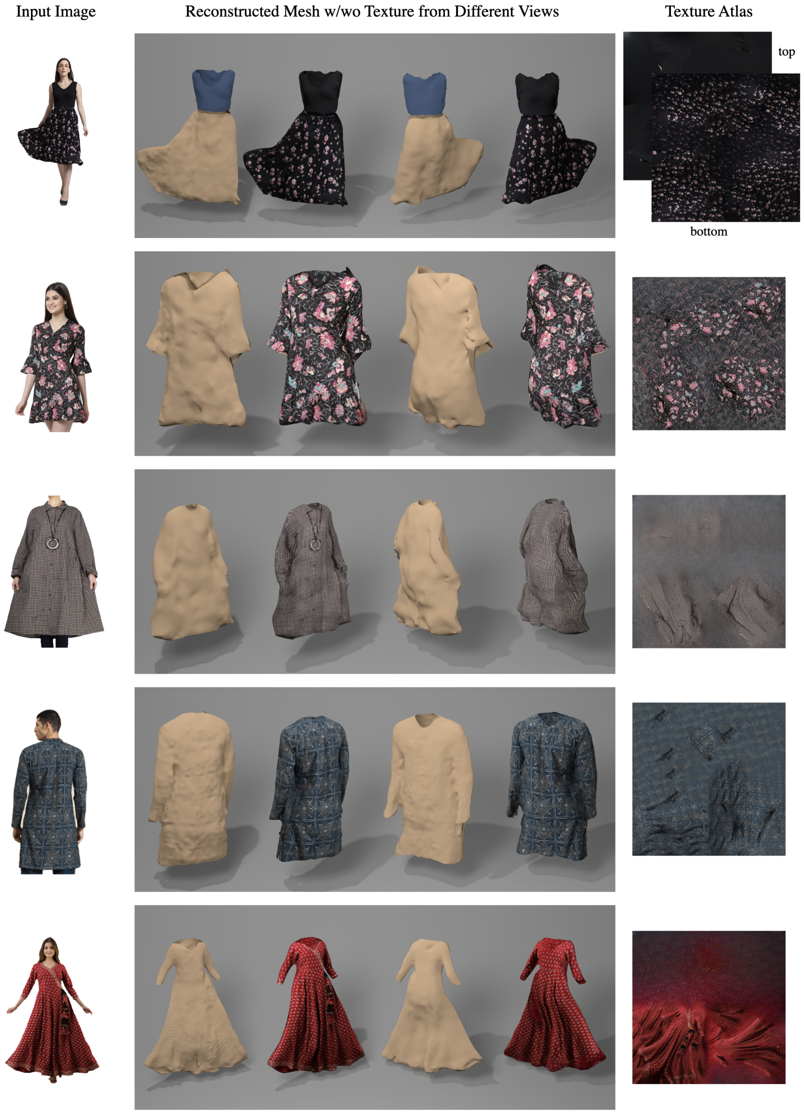

Figure 6: Results generated by xCloth for in-the-wild internet images. For each row, the input image is followed by the reconstruction in a different view (including geometry and textured mesh) and respective texture maps.

the 3DHumans dataset and visualize the P2S distance (w.r.t., corresponding ground truth mesh) on the self-occluded (back side) of the garment in Figure 7. As we can infer that reconstruction error is slightly higher in the region where clothing is loose and far from the body surface. This is due to the inherently ill-posed nature of the problem, even though the final reconstructed garment mesh looks plausible.

In regard to comparison with SOTA methods, the training code for BCNet [11] & MGN [4]) is not publicly available, and Pix2Surf [17] does not model geometry of the garment and hence we skip comparing with them. DeepFashion3D [26] is a closely related work to xCloth in terms of geometry estimation as it can model arbitrary geometry up to some extent; however, the authors do not provide training/inference code. Since OccNet [16] is a major part of DeepFashion3D and the authors compare the method with OccNet in the paper, we train OccNet [16] on 3DHumans and THUman2.0 datasets and quantitatively compare it with xCloth in terms of P2S. In OccNet, neither there is any provision for semantic segmentation for each garment class, nor the final reconstruction is pixel-aligned with the input image; therefore, we skip IOU and NRE during quantitative comparison. We report the quantitative comparison with OccNet in Table 3 where our framework significantly outperforms OccNet for both Topwear and Bottomwear. This can be attributed to the fact that OccNet relies on global features while our framework exploits pixel-aligned features.

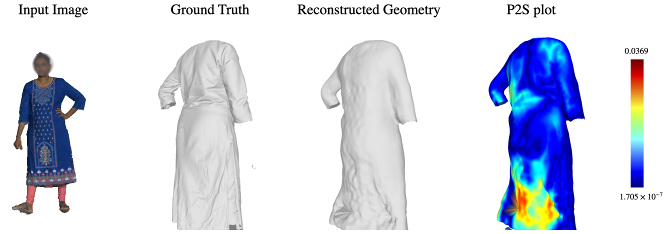

Figure 7: Visualization of P2S distance.

Dataset	Method	P2S ↓ (Topwear)	P2S ↓ (Bottomwear)
3DHumans	OccNet	0.0632	0.0363
	Ours	**0.0087**	**0.0077**
THUman2.0	OccNet	0.0609	0.0312
	Ours	**0.0079**	**0.0072**

Table 3: Quantitative comparison with OccNet [16].

3.5 Qualitative Evaluation

We show the qualitative results of xCloth on the test samples taken from three datasets in the Figure 9. As we can infer, we are able to model arbitrary style loose clothing as shown in Figure 9(i) and handle self-occlusions as shown in Figure 9(i). Figure 9(iii) and (iv) demonstrate the ability of our framework to deal with arbitrary poses. Additionally, we show the ability of xCloth to reconstruct 3D garments from in-the-wild/catalogue images taken from the internet with arbitrary loose clothing styles in Figure 6. We also compare our framework with the existing state of the art methods (BCNet [11], MGN [4], Pix2Surf [17] , OccNet [16]) on in-the-wild images taken from the internet. Compared to the other methods, our framework is superior in recovering high quality textured meshes, as can be seen in Figure 5. Pix2Surf [17] can transfer texture map for specific cloth categories from two input images (front and back), but it fails to generalize if the input cloth image is not in canonical form. Similarly, [4] only recovers the geometry of specific categories (Shirt, Tshirt, Shorts, Pants & Long Coat) as the final output but the texture stitching pipeline of [2] is used for transferring texture which requires multi-view images for a decent texture recovery, so we only qualitatively compare only the geometry under monocular settings. We train OccNet [16] on the 3DHumans [5] and THUman2.0 [25] datasets and show qualitative comparison in Figure 8. Here ”Not Suppported” is listed when respective garment template is unavailable.

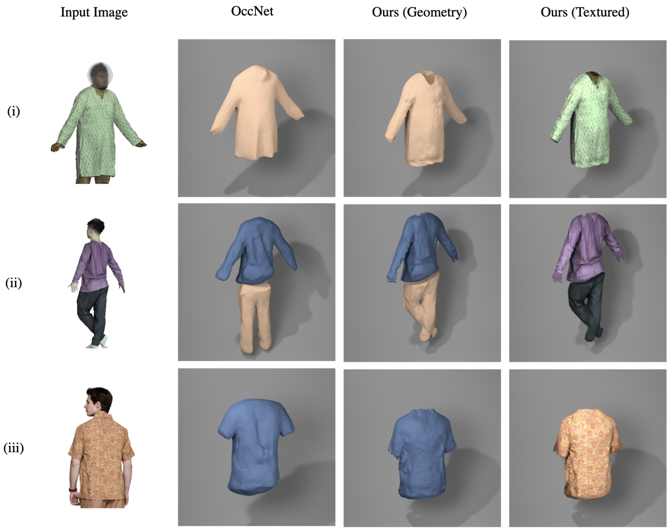

Figure 8: Qualitative comparison with OccNet [16].

It is important to note that, unlike other methods, our framework can recover both fine-grained geometrical details as well as textured representation. Further, our framework can generalize to loose and diverse garments with a complex style, unlike other template-based approaches.

4 DISCUSSION

4.1 Ablation Study

Effect of Normal Loss: Here, we discuss the importance of adding a decoder branch for predicting normal peelmaps. The depth estimation helps recover the global structure of the surface. However, it might not recover high-frequency geometrical details (like wrinkles) on the surface. The surface normals aptly capture such details, and hence predicting normal peel maps (while minimizing average loss) helps in recovering fine geometrical details of the garments. Additionally, it also helps in regularising the depth map prediction, and yielding lowers values for P2S and NRE. Thus, when we exclude the Normal Loss (L_{norm}) and associated loss, the surface quality deteriorates, leading to a drop in P2S and NRE metrics, as reported in Table 2.

Choice of Architecture: Instead of directly adopting the ResNet architecture proposed in [21], we perform an ablative study on the architecture choice. We experiment with U-Net [20] & Stacked-Hourglass Network [19] apart from ResNet [9] and compute the P2S error on the final reconstruction while training on the 3DHumans dataset. The corresponding P2S error values for UNet, ResNet and Stacked-Hourglass Network are 0.0062, 0.0074 and **0.0057**, respectively. Thus, we retain ResNet as the underlying architecture and build on top of it.
Please refer to the supplementary material for further ablative experiments.

4.2 Analysis & Insights

We adopted the non-parametric peeled representation to model garments, thereby removing the need for predefined garment templates which can model only a fixed number of garment styles (mostly tight clothing). The peeled representation is inherently sparse and also deals with self-occlusion. Additionally, the peelmaps (depth, segmentation, normal & RGB) are pixel-aligned and hence offer an easy way to extract each garment separately from the reconstructed clothed body. SMPL peeled depth prior provided to the common encoder deals with both pose and depth ambiguity in the monocular setup. The proposed novel SMPL body segmentation prior helps the segmentation decoder to localise the body parts

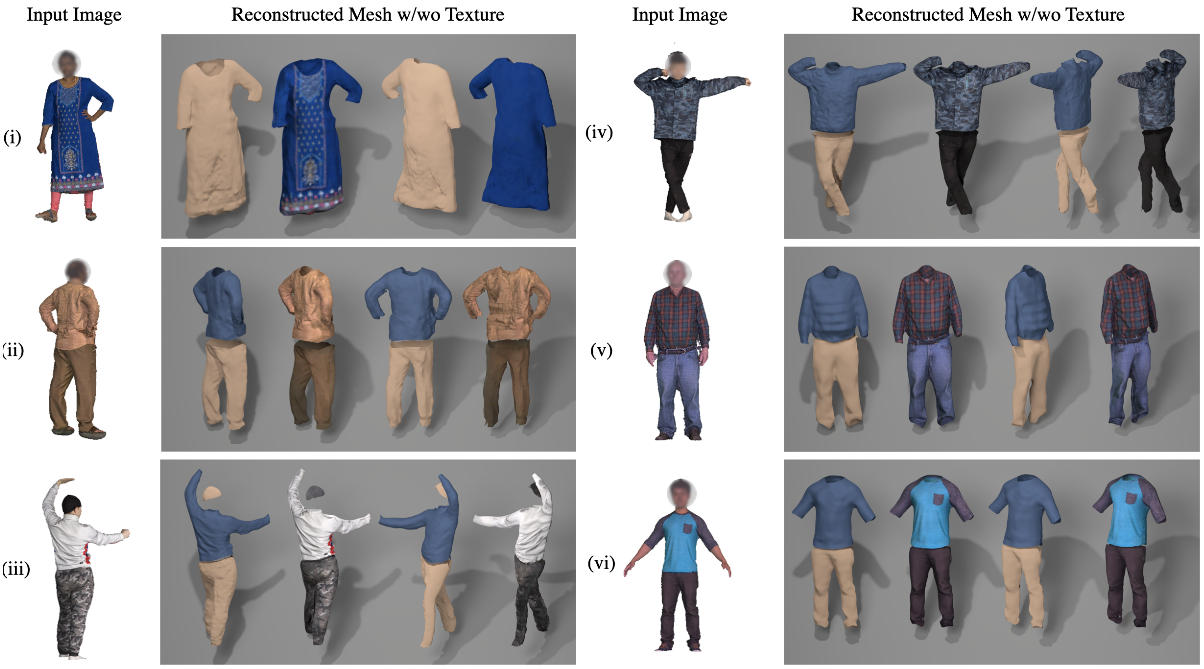
Input Image
Reconstructed Mesh w/wo Texture
Input Image
Reconstructed Mesh w/wo Texture
(i)
(ii)
(iii)
(iv)
(v)
(vi)

Figure 9: Qualitative results of xCloth on test samples from different datasets.

while predicting garment segmentation labels (e.g. beanie extracted in Figure 9(iii)). However, it is important to note that our proposed non-parametric framework predicts garment specific segmentation peelmaps for significantly loose clothing styles that are not in direct correspondence to SMPL body segmentation prior. Apart from recovering the geometry, peeled representation also guides automatic seams estimation, enabling UV parametrization of any arbitrary garment mesh. Another useful insight comes from employing a common encoder with multiple decoder branches. All the different decoder branches provide each other just the relevant information by propagating the gradients via the common encoder while avoiding interfering directly with each other, resulting in more flexible learning. This is evident from the ablation study of normal loss, where the addition of D_{norm} branch helps to reduce the P2S distance values in the predictions of D_{depth}.

For inpainting of the texture atlas, we explored existing learning-based SOTAs. However, almost all inpainting solutions employ generative models, which tend to produce blurry output. Since garment textures come in all kinds of patterns, colors and styles, the unavailability of such diverse datasets to enable generalized learning is another key reason for the failure of learning-based methods. Here, structure completion in inpainting performs effortlessly better than learning-based methods as the former takes cues coming from the structures and repeated patterns present in the image and tries to replicate them periodically. Please refer to supplementary material for the comparative study.

4.3 Limitations & Future Work

Our framework does not enforce any spatial relationship between similar parts (e.g. sleeves) of the different garment styles (e.g.shirt and long coat) in the UV space. One possible option is to explore continuous surface embeddings (e.g., CSE [18]) to establish some meaningful relationship across an independent reconstruction of different garment categories. Additionally, in the case of monocular video, our framework is not guaranteed to extract consistent texture for each frame. In future, it would be interesting to explore solutions to predict consistent texture across all the frames. Our framework can be further extended to predict the underlying body with accurate shape and pose (instead of taking from SMPL prior) and also to deal with layered clothing, which is extremely difficult to model due to the absence of such real-world datasets.

5 CONCLUSION

We propose a novel method for template-free digitization of 3D garments with automated texture-mapping. We adopt a non-parametric representation and exploit it in a novel way to recover textured 3D garment meshes that can be rendered with a high-fidelity appearance. We compare our framework both qualitatively and quantitatively, where we outperform existing SOTA methods and demonstrate the generalization of our framework to in-the-wild images. We provide an ablative study and also discuss insights gathered from the analysis. Finally, we discuss the limitations & future directions of our proposed framework.

Acknowledgement: We thank Dhawal Sirikonda for helping us in the GPU based implementation of texture filling module.

REFERENCES

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

[2] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. 2019. Learning to Reconstruct People in Clothing from a Single RGB Camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2019.00127

[3] Vertica Bhardwaj and Ann Fairhurst. 2010. Fast fashion: response to changes in the fashion industry. *The international review of retail, distribution and consumer research* 20, 1 (2010), 165–173.

[4] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. 2019. Multi-Garment Net: Learning to Dress 3D People from Images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/ICCV.2019.00552

[5] CVIT. 2021. 3DHumans: A Rich 3D Dataset of Scanned Humans. http://cvit.iiit.ac.in/research/projects/cvit-projects/sharp-3dhumans-a-rich-3d-dataset-of-scanned-humans

[6] J de Joya, R Narain, J O' Brien, A Samii, and V Zordan. 2012. Berkeley Garment Library. (2012). http://graphics.berkeley.edu/resources/GarmentLibrary/index.html

[7] Patrizia Gazzola, Enrica Pavione, Roberta Pezzetti, and Daniele Grechi. 2020. Trends in the fashion industry. The perception of sustainability and circular economy: A gender/generation quantitative approach. *Sustainability* 12, 7 (2020), 2809.

[8] Ke Gong, Xiaodan Liang, Yicheng Li, Yimin Chen, Ming Yang, and Liang Lin. 2018. Instance-level Human Parsing via Part Grouping Network. https://doi.org/10.1007/978-3-030-01225-0_47 arXiv:1808.00157 [cs.CV]

[9] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 770–778.

[10] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. 2014. Image completion using planar structure guidance. *ACM Transactions on graphics (TOG)* 33, 4 (2014), 1–10.

[11] Boyi Jiang, Juyong Zhang, Yang Hong, JinHao Luo, Ligang Liu, and Hujun Bao. 2020. BCNet: Learning Body and Cloth Shape from A Single Image. *ArXiv* abs/2004.00214 (2020).

[12] Sai Sagar Jinka, Rohan Chacko, Avinash Sharma, and PJ Narayanan. 2020. PeeledHuman: Robust Shape Representation for Textured 3D Human Body Reconstruction. In *Proceedings of the IEEE Conference on 3D Vision (3DV)*. https://doi.org/10.1109/3DV50981.2020.00098

[13] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. 2002. Least squares conformal maps for automatic texture atlas generation. *ACM transactions on graphics (TOG)* 21, 3 (2002), 362–371.

[14] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: a skinned multi-person linear model. *ACM Trans. Graph.* 34 (2015), 248:1–248:16.

[15] Sahib Majithia, Sandeep N Parameswaran, Sadbhavana Babar, Vikram Garg, Astitva Srivastava, and Avinash Sharma. 2022. Robust 3D Garment Digitization from Monocular 2D Images for 3D Virtual Try-On Systems. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 3428–3438.

[16] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2019.00459

[17] Aymen Mir, Thiemo Alldieck, and Gerard Pons-Moll. 2020. Learning to Transfer Texture from Clothing Images to 3D Humans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[18] Natalia Neverova, David Novotný, Vasil Khalidov, Marc Szafraniec, Patrick Labatut, and Andrea Vedaldi. 2020. Continuous Surface Embeddings. *ArXiv* abs/2011.12438 (2020).

[19] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked Hourglass Networks for Human Pose Estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. https://doi.org/10.1007/978-3-319-46484-8_29

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*.

[21] Sai Sagar Jinka, Astitva Srivastava, Chandradeep Pokhariya, Avinash Sharma, and P. J. Narayanan. 2022. SHARP: Shape-Aware Reconstruction of People in Loose Clothing. *arXiv e-prints*, Article arXiv:2205.11948 (May 2022), arXiv:2205.11948 pages. arXiv:2205.11948 [cs.CV]

[22] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. 2019. PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/ICCV.2019.00239

[23] Rohan Sawhney and Keenan Crane. 2017. Boundary first flattening. *ACM Transactions on Graphics (ToG)* 37, 1 (2017), 1–14.

[24] Shan Yang, Zherong Pan, Tanya Amert, Ke Wang, Licheng Yu, Tamara Berg, and Ming C Lin. 2018. Physics-inspired garment recovery from a single-view image. *ACM Transactions on Graphics (TOG)* 37, 5 (2018), 1–14.

[25] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. 2021. Function4D: Real-time Human Volumetric Capture from Very Sparse Consumer RGBD Sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5746–5756. https://doi.org/10.1109/CVPR46437.2021.00569

[26] Heming Zhu, Yu Cao, Hang Jin, Weikai Chen, Dong Du, Zhangye Wang, Shuguang Cui, and Xiaoguang Han. 2020. Deep Fashion3D: A Dataset and Benchmark for 3D Garment Reconstruction from Single Images. *ArXiv* abs/2003.12753 (2020).

[27] Heming Zhu, Lingteng Qiu, Yuda Qiu, and Xiaoguang Han. 2022. Registering Explicit to Implicit: Towards High-Fidelity Garment mesh Reconstruction from Single Images. https://doi.org/10.48550/ARXIV.2203.15007