DeepIron: Predicting Unwarped Garment Texture from a Single Image

Hyun-Song Kwon and Sung-Hee Lee

Korea Advanced Institute of Science and Technology, Republic of Korea

Figure 1: *We introduce DeepIron, which reconstructs the texture of 3D garments by inferring the unwarped original texture (right, middle, and bottom) from the input garment image (right, top). The inferred unwarped textured images allow to create realistic appearance of 3D garments when deformed to fit new poses (left).*

Abstract

Realistic reconstruction of 3D clothing from an image has wide applications, such as avatar creation and virtual try-on. This paper presents a novel framework that reconstructs the texture map for 3D garments from a single garment image with pose. Since 3D garments are effectively modeled by stitching 2D garment sewing patterns, our specific goal is to generate a texture image for the sewing patterns. A key component of our framework, the Texture Unwarper, infers the original texture image from the input garment image, which exhibits warping and occlusion of the garment due to the user's body shape and pose. This is effectively achieved by translating between the input and output images by mapping the latent spaces of the two images. By inferring the unwarped original texture of the input garment, our method helps reconstruct 3D garment models that can show high-quality texture images realistically deformed for new poses. We validate the effectiveness of our approach through a comparison with other methods and ablation studies.

CCS Concepts

• ***Computing methodologies*** → ***Texturing;***

1. Introduction

Reconstructing garment textures from images presents a challenging task because garments are heavily deformed by the body shape and pose, and are occluded by other body parts and wrinkles. Researchers have developed techniques to restore garment textures from input images. Mir et al. [MAP20] proposed a novel method to establish correspondences between garment image silhouettes and a 2D UV map of the 3D garment surface. Majithia et al. [MPB*22] used parametric mesh models for some garment types (e.g., T-shirts, trousers) to map high-quality textures from a fashion catalog image to UV map panels for the garment models. Cloth2Tex utilizes neural rendering to establish dense correspondences between 2D catalog images and 3D clothing meshes to extract textures, which are then enhanced with high-fidelity texture inpainting [GCZ*23].

Despite these significant advancements, existing research has not yet taken into account the texture distortion resulting from prominent curves on the human body surface. Texture for the occluded part of the garment can hardly be reconstructed as well. In addition, some approaches focusing on reproducing accurate appearances for

the given pose (e.g., simply copying the garment texture in the input image) may struggle to generate realistic garment appearances for new poses. To address this limitation, we take the approach of reconstructing garments in terms of their original sewing patterns and distortion-free texture map. These are subsequently stitched and draped by a physical simulator to generate realistic deformation and appearance for arbitrary poses.

In this work, we focus on generating texture maps for the sewing patterns from input images. Central to our method is the Texture Unwarper, which effectively transforms the distorted and occluded garment image to its original texture image. Our main idea to obtain a high-quality texture image is to separately obtain latent spaces for the input garment image and the output texture image, and learn to map between the two latent spaces. By sampling only from the texture image latent space, our method guarantees to produce clean undistorted textures.

As a second contribution, we propose a strategy to separately train each module of the Texture Unwarper. We show that this sequential training increases the quality of the output texture image over the end-to-end training by effectively training each module. We validate the effectiveness of our approach through a comparison with other methods and ablation studies.

2. Method

2.1. System Overview

Figure 2 shows the overall pipeline to reconstruct a 3D garment from a single image. The framework takes as input an RGB image (800×800) of a clothed person and its normal map estimated from the RGB image by using [SSSJ20]. We assume that a human model and appropriate sewing patterns for the garment in the input image are estimated by external modules (e.g., [ZCL*20] and [BKL21]). For our experiment, we use a ground truth single female body and a single sewing pattern for each garment type and only focus on predicting texture.

Given the input RGB and normal map images, the Garment Segmentation stage first segments the garment area by using [JSS*20]. For T-shirt, we extract only the torso region under the assumption that the torso area contains sufficient information for the texture image for the whole area. The Texture Unwarper then generates an unwarped image (256×256) for the front side of the T-shirt or the pants within the entire texture map. To obtain the entire texture map, the Texture Map Generator places the unwarped texture image in a predefined region and fills the remaining areas with a symmetry operation and in-painting by using [SLM*22]. Lastly, the generated texture map is applied to the sewing pattern. The sewing pattern is stitched and draped on the posed body by the garment simulator to obtain the final output of a posed character.

2.2. Texture Unwarper

The Texture Unwarper is tasked to predict the original texture image from distorted and occluded garment image. It consists of three main components: encoders, Distortion Corrector, and texture generator (Figure 3).

Our encoder and texture generator are based on Variational Auto-Encoder (VAE) to deal with the indeterminacy of the solution (Figure 3, top). To transform between images, we need to change both the shape of the distorted patterns and overall pixel values in the input image, which motivated us to adopt StyleGAN [KLA19] that disentangles content and style components and modifies individual components. Note that only the encoder of input images and the decoder (texture generator) of the texture images are used for the Texture Unwarper. The Garment Encoder generates the content code $z_{content} \in \mathbb{R}^{N_1 \times N_1 \times D_1}$ that describes the location and shape of patterns in a texture, and the style code $z_{style} \in \mathbb{R}^{D_2}$ that describes the overall distribution of RGB pixel values. The Normal Map Encoder, following the encoder structure of [ALY*21], extracts a latent code $z_{normal} \in \mathbb{R}^{N_2 \times N_2 \times D_2}$ that reflects the distortion of the garment in the input image.

The Distortion Corrector (Figure 3, bottom) takes these three latent vectors (z_{normal}, $z_{content}$, z_{style}) as inputs and fuse them by matching their sizes, concatenating them, and then passing them through two 3×3 convolution layers. The fused result passes through a convolution layer for content or a linear layer for style to predict μ and σ to sample new content code $z'_{content} \in \mathbb{R}^{N_1 \times N_1 \times D_1}$ and style code $z'_{style} \in \mathbb{R}^{D_2}$ to be fed to the Texture Generator. The idea of translating between disentangled latent codes was inspired by [KPTF21], but a large modification was added to match our purpose, including the addition of a normal map latent vector as input.

Modeled based on StyleGAN, the Texture Generator receives $z'_{content}$ and z'_{style} as input and generates an unwarped texture image. StyleGAN employs adaptive instance normalization (AdaIN) [DSK16] layers, which are positioned after each convolutional layer in its generator, to exert precise control over the visual attributes of the generated images.

Training of Texture Unwarper

Figure 3 shows the training scheme of the Texture Unwarper. Since the data distributions for the input images and the output images are distinct, the latent spaces need to be learned separately. The Distortion Corrector is then learned to translate between the two latent spaces. First, the encoder-generator structure is used to reconstruct the segmented RGB image and the distortion-corrected image (GT), respectively. The loss function used in this step is designed based on β-VAE and StyleGAN as follows:

$$\begin{aligned} L_{Enc-Gen} &= L_{StyleGAN} + L_{VAE} \\ L_{VAE} &= -E_{z\sim q(z|x)}[\log(p(x|z))] + \beta KL(q(z|x) \parallel p(z)) \end{aligned} \tag{1}$$

where $L_{StyleGAN}$ denotes the adversarial losses of StyleGAN. The first term of L_{VAE} measures the reconstruction quality of the image x in terms of the perceptual distance. The second term encourages the latent space z to follow the prior distribution $p(z)$, modeled as the standard normal distribution, with weight controlled by β. Subsequently, we train the Normal Map Encoder and Distortion Corrector with parameters for the Garment Encoder and Texture Generators fixed. The loss function used in this step is as follows:

$$L_{disCorr} = L_{VAE} + L_{StyleGAN} + \|\widehat{I} - I\|_1 \tag{2}$$

where we add an L_1 loss to reduce the difference in pixel values between the predicted $\widehat{I}$ and GT I images.

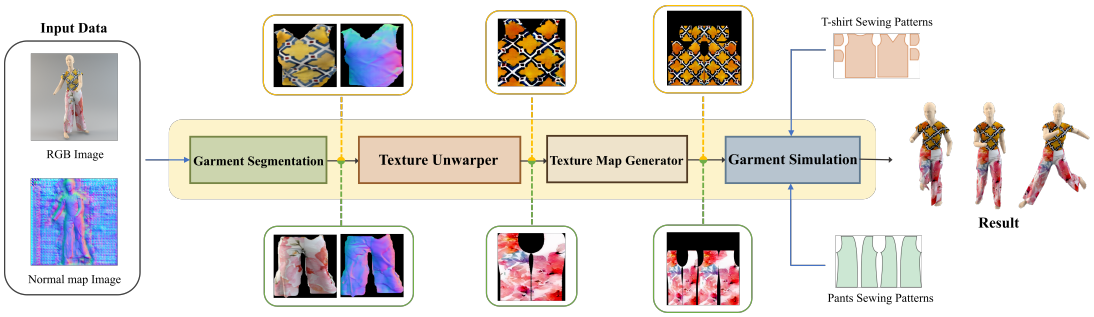

Figure 2: *The overall pipeline to reconstruct a 3D garment model from a single RGB image of a posed person and its estimated normal map. This paper focuses on the Texture Unwarper module, which predicts the original undistorted texture image from the input images. Reconstructing upper and lower garments is conducted separately.*

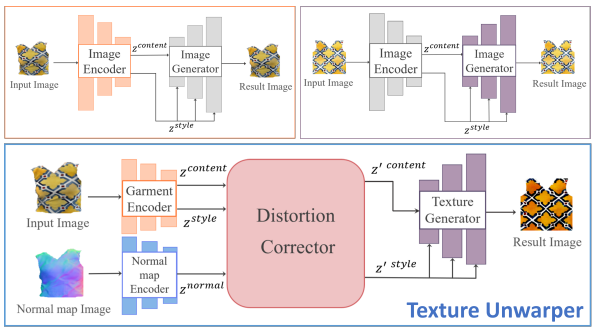

Figure 3: *Training scheme of the Texture Unwarper. Garment Encoder and Texture Generator are trained separately first, followed by the concurrent training of the Distortion Corrector and Normal Map Encoder.*

Figure 4: *Images generated by the Texture Unwarper. Input image (left), generated result (middle), and GT (right) are shown for T-shirt and pants.*

Compared with a single end-to-end training scheme, this sequential training scheme helps each encoder learn its own data distribution more accurately. In addition, by storing the latent codes for the images and using them for training the Distortion Corrector, the overall training time could be significantly reduced.

To train our model, we need a pair of an original texture map and its corresponding dressed image. For this purpose, we generated texture maps for the sewing patterns provided by [KL21]. Our dataset includes a variety of texture images, including diagonal lines, vertical stripes, and flower patterns, sourced from Vecteezy, an online photo marketplace. The training dataset included 20K images for input data (segmented RGB and normal map images) and 7362 for GT. The validation dataset included 3922 and 3000, respectively. Titan Xp GPU was used for training.

3. Results and Experiments

3.1. Qualitative Comparison

We conducted qualitative comparisons with Pix2Surf [MAP20]. Since the garment models in the Pix2Surf dataset do not exhibit significant warping due to body shape, we used our own test dataset for comparison. As Pix2Surf has been pretrained with its own dataset, this is not a strictly fair comparison. Therefore, instead of performing a quantitative comparison, we focus on a qualitative assessment of the overall texture quality, specifically looking for the presence of distortion or wrinkles in the texture map.

Figure 5 shows the results from Pix2Surf and our method. It is evident that the texture maps created by Pix2Surf still contain wrinkles from the original image. This artifact is somewhat unavoidable in approaches, including Pix2Surf, that attempt to map the input image to a texture map without completely addressing warping and wrinkles in the reconstructed garment geometry. In contrast, our method infers the original texture pattern from the image using a texture generative model, which can learn and generate distortion-free textures.

3.2. Ablation studies

	SSIM (↑)	LPIPS (↓)	FID (↓)
End-to-end trained	0.17	0.91	394.98
Without Distortion Corrector	0.29	0.62	136.86
Ours	**0.31**	**0.61**	**114.57**

Table 1: *Ablation studies. Top: Texture Unwarper trained end-to-end. Middle: Distortion Corrector network removed.*

We examined the benefit of training each sub-module of the Texture Unwarper separately by comparing with training the entire

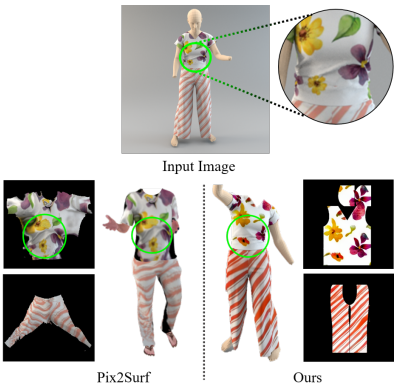

Figure 5: *Comparison with Pix2Surf and ours.*

module end-to-end. In addition, we evaluated the contribution of the Distortion Corrector by removing it from the Texture Unwarper. Table 1 shows the results of the ablation study. The result shows that the end-to-end training has the worst performance across all metrics, highlighting the benefit of our sequential training scheme. Furthermore, the model trained without the Distortion Corrector also exhibited lower performance than our model. This suggests that fixing two latent spaces and connecting them by using the normal map information is more effective than retraining the networks.

4. Conclusion and Future Work

In this paper, we presented a novel framework for realistic reconstruction of 3D clothing from a single image. Our framework addresses the challenge of inferring the texture map for 3D garments. A key component of our framework is the Texture Unwarper, which effectively transforms the input clothing image, accounting for warping and occlusion of texture caused by the user's body shape and pose. By mapping the latent spaces of the input and output images, the Texture Unwarper infers the unwarped original texture of the input garment. This allows us to reconstruct 3D garment models capable of realistically deforming high-quality texture images for new poses.

As our Texture Unwarper learns the distribution of image data, the quality of the results tends to decrease when the input texture deviates from the learned distribution. This issue could potentially be addressed by incorporating additional texture data. In this study, our focus was primarily on T-shirts and pants. Extending our method to accommodate a wider range of clothing types and sizes, and incorporating a sewing pattern prediction model, remains an important future direction. Furthermore, developing a robust model for diverse real-world variables, such as body shape, lighting, etc., is also essential for research going forward (Figure 6).

Acknowledgement

This work was supported by NRF, Korea (2022R1A4A5033689), by IITP, MSIT, Korea (2022-0-00566), and by MSIT and Gwangju Metropolitan City, Korea (AI industrial convergence cluster development project).

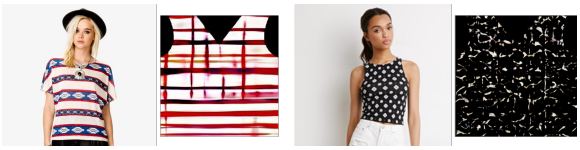

Figure 6: *Testing with real-world data (left: input image, right: Texture Unwarper output image).*

References

[ALY*21] ALBAHAR, BADOUR, LU, JINGWAN, YANG, JIMEI, et al. "Pose with Style: Detail-preserving pose-guided image synthesis with conditional stylegan". *ACM Transactions on Graphics (TOG)* 40.6 (2021), 1–11 2.

[BKL21] BANG, SEUNGBAE, KOROSTELEVA, MARIA, and LEE, SUNG-HEE. "Estimating Garment Patterns from Static Scan Data". *Computer Graphics Forum* 40.6 (2021), 273–287. DOI: https://doi.org/10.1111/cgf.14272 2.

[DSK16] DUMOULIN, VINCENT, SHLENS, JONATHON, and KUDLUR, MANJUNATH. "A learned representation for artistic style". *arXiv preprint arXiv:1610.07629* (2016) 2.

[GCZ*23] GAO, DAIHENG, CHEN, XU, ZHANG, XINDI, et al. "Cloth2Tex: A Customized Cloth Texture Generation Pipeline for 3D Virtual Try-On". *arXiv preprint arXiv:2308.04288* (2023) 1.

[JSS*20] JIA, MENGLIN, SHI, MENGYUN, SIROTENKO, MIKHAIL, et al. "Fashionpedia: Ontology, segmentation, and an attribute localization dataset". *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer. 2020, 316–332 2.

[KL21] KOROSTELEVA, MARIA and LEE, SUNG-HEE. "Generating datasets of 3d garments with sewing patterns". *arXiv preprint arXiv:2109.05633* (2021) 3.

[KLA19] KARRAS, TERO, LAINE, SAMULI, and AILA, TIMO. "A style-based generator architecture for generative adversarial networks". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, 4401–4410 2.

[KPTF21] KIM, SEUNG WOOK, PHILION, JONAH, TORRALBA, ANTONIO, and FIDLER, SANJA. "Drivegan: Towards a controllable high-quality neural simulation". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 5820–5829 2.

[MAP20] MIR, AYMEN, ALLDIECK, THIEMO, and PONS-MOLL, GERARD. "Learning to transfer texture from clothing images to 3d humans". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 7023–7034 1, 3.

[MPB*22] MAJITHIA, SAHIB, PARAMESWARAN, SANDEEP N, BABAR, SADBHAVANA, et al. "Robust 3D Garment Digitization from Monocular 2D Images for 3D Virtual Try-On Systems". *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, 3428–3438 1.

[SLM*22] SUVOROV, ROMAN, LOGACHEVA, ELIZAVETA, MASHIKHIN, ANTON, et al. "Resolution-robust large mask inpainting with fourier convolutions". *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2022, 2149–2159 2.

[SSSJ20] SAITO, SHUNSUKE, SIMON, TOMAS, SARAGIH, JASON, and JOO, HANBYUL. "Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 84–93 2.

[ZCL*20] ZHANG, HONGWEN, CAO, JIE, LU, GUO, et al. "Learning 3d human shape and pose from dense body parts". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.5 (2020), 2610–2627 2.