FabricDiffusion: High-Fidelity Texture Transfer for 3D Garments Generation from In-The-Wild Clothing Images (Supplementary Materials)

CHENG ZHANG*, Carnegie Mellon University, United States of America
YUANHAO WANG*, Carnegie Mellon University, United States of America
FRANCISCO VICENTE CARRASCO, Carnegie Mellon University, United States of America
CHENGLEI WU, Google Inc., United States of America
JINLONG YANG, Google Inc., Switzerland
THABO BEELER, Google Inc., Switzerland
FERNANDO DE LA TORRE, Carnegie Mellon University, United States of America

ACM Reference Format:
Cheng Zhang, Yuanhao Wang, Francisco Vicente Carrasco, Chenglei Wu, Jinlong Yang, Thabo Beeler, and Fernando De la Torre. 2024. FabricDiffusion: High-Fidelity Texture Transfer for 3D Garments Generation from In-The-Wild Clothing Images (Supplementary Materials). In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 3–6, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3680528.3687637

In this supplementary material, we provide details and results omitted in the main text.

- Section 1: Key advantages of FabricDiffusion.
- Section 2: Additional details on dataset construction.
- Section 3: Additional Implementation Details.
- Section 4: Additional results and analyses.

1 KEY ADVANTAGES OF FABRICDIFFUSION

Normalized texture representation. Unlike existing image-to-3D texture transfer methods, FabricDiffusion generates normalized textures that can be used in the 2D UV space. We highlight two outputs: (1) High-quality, distortion-free, and tileable texture maps from a non-rigid garment surface. (2) Seamless integration with SVBRDF material estimation pipelines, which usually build upon the first output — standard close-up views of the materials as input.

Sim-to-real generalizability. The conditional diffusion model, trained entirely using *synthetic* rendering images, proves highly effective in generating normalized texture maps from *real-world* images. We attribute this success to: (1) Our model bridging the domain gap between real and rendered textures by conditioning on the real input texture. (2) Synthetic data offering controllable supervision and diverse geometric, illumination, and occlusion variations.

Data and computational efficiency. During training, our method of creating pseudo-BRDF material is effective in scaling up the training examples. During inference, our model performs feed-forward sampling from Gaussian noise, which takes approximately less than 5 sec on a single NVIDIA A6000 GPU. In contrast, existing texture transfer methods often rely on costly per-example optimization.

2 DETAILS ON DATASET CONSTRUCTION

Fabric BRDF and textile dataset. To curate textures and their BRDF materials, we use several public libraries (AmbientCG[1], ShareTextures[2], 3D Textures[3]) under the CC0 license and supplement them with additional assets purchased from artists. The real BRDF dataset we collected comprises 3.8k assets, encompassing a broad spectrum of fabric materials. The pseudo-BRDF dataset contain 100k fabric textures with only RGB color images. We reserved 200 materials from the real BRDF dataset for testing our BRDF generator, and 800 materials from the pseudo BRDF dataset (combined with the previous 200 materials) for testing the texture flattening module.

Our textile images are collected from online sources including Openverse[4], PublicDomainPictures[5], and ARTX[6] under CC0 or royalty-free license.

3D garment mesh dataset. We collect 22 raw 3D garment meshes for training and 5 garment meshes for testing. That is, during the testing with synthetic data, the model has not seen the geometry from the 5 testing meshes. With the method described in Section 3.2 of the main paper, we construct approximately 220k flat and warped texture pairs for training and 5k pairs for testing.

Logos and prints dataset. We collect a dataset of 7k prints and logos in PNG format with CC0 license. Their corresponding pseudo-BRDF materials are generated by assigning a uniform roughness value sampled from $\mathcal{U}(0.4, 0.7)$, a uniform metallic value sampled from $\mathcal{U}(0, 0.3)$, and a default flat normal map. In cases where a print was uniformly black, we converted it to white if the background texture was also dark. By compositing the logo prints onto the 3D

[*] Equal contribution. Website: https://humansensinglab.github.io/fabric-diffusion.

[1] https://ambientcg.com/
[2] https://www.sharetextures.com/
[3] https://3dtextures.me/
[4] https://openverse.org/
[5] https://publicdomainpictures.net/en/
[6] https://architextures.org/

SA Conference Papers '24, December 3–6, 2024, Tokyo, Japan
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1131-2/24/12.
https://doi.org/10.1145/3680528.3687637

garments, we obtain a total of 82k warped print images, following the method outlined in Section 3.2 of the main paper.

3 ADDITIONAL DETAILS OF OUR METHOD

Details on physics-based rendering. During rendering, each image pixel value at a specific viewing direction can be computed using the following reflectance equation:

$$L(p,\omega_o)=\int_{\Omega} f_r(p,\omega_i,\omega_o)L_i(p,\omega_i)(\omega_i\cdot n_p)\mathrm{d}\omega_i, \tag{1}$$

where L is the rendered pixel color along the direction ω_o from the surface point p, $\Omega=\{\omega_i:\omega_i\cdot n_p\geq 0\}$ denotes a hemisphere with the incident direction ω_i and surface normal n_p at point p, L_i is the incident light that is represented by the environment map, and f_r is known as the BRDF that scales or weighs the incoming radiance based the material parameters (k_d,k_n,k_r,k_m) of the garment surface. By aggregating the rendered pixel colors along the direction ω_o (i.e., camera pose), we are able to obtain the rendered the image of the input patch (image x in Equation (1) of the main paper).

Classifier-free guidance for conditional image generation We leverage Classifier-Free Guidance (CFG) [Ho and Salimans 2022] during the training for trading off the quality and diversity of samples generated by our FabricDiffusion model. The implementation of CFG involves jointly training the diffusion model for conditional and unconditional denoising, and combining the two score estimates (the ℓ_2 loss of the noise term in Equation (3) of the main paper) at inference time. Training for unconditional denoising is done by simply setting the conditioning to a fixed null value $\mathcal{E}(x)=\varnothing$ at some frequency during training. At inference time, with a guidance scale $s\geq 1$, the modified score estimate $\tilde{e_\theta}(x_t,\mathcal{E}(x))$ is extrapolated in the direction toward the conditional $e_\theta(x_t,\mathcal{E}(x))$ and away from the unconditional $e_\theta(x_t,\varnothing)$:

$$\tilde{e_\theta}(x_t,\mathcal{E}(x))=e_\theta(x_t,\varnothing)+s\cdot(e_\theta(x_t,\mathcal{E}(x))-e_\theta(x_t,\varnothing)). \tag{2}$$

CFG enhances the visual quality of generated texture maps and ensures that the sampled images more accurately correspond to the input texture in terms of color, pattern, and scale.

Strategy for determining tiling scales. After extracting PBR material maps from an image exemplar, we tile them in the garment UV space for realistic rendering. The key question is how to determine the scale for tiling? We investigate two specific strategies: (1) Proportion-aware tiling. We use image segmentation to calculate the proportion of the captured region relative to the segmented clothing, maintaining the same ratio when tiling the generated texture onto the sewing pattern. (2) User-guided tiling. We emphasize that an end-to-end automatic tilling method may not be optimal, as user involvement is often necessary to resolve ambiguities and provide flexibility in fashion industries.

Implementation details. We use pre-trained Stable Diffusion v1.5 as the backbone of the normalized texture map generation and fine-tune it on our texture and print datasets, respectively. Both the input and output scales are set as 256×256px. We use a batch size of 512 and a learning rate of 5×10^{-5}. It takes roughly 2 days (20k iterations) to train on four NVIDIA A6000 GPUs. For PBR materials estimation, we fine-tuned the pre-trained MatFusion model for roughly 1 hour with our 3.8k BRDF materials training data.

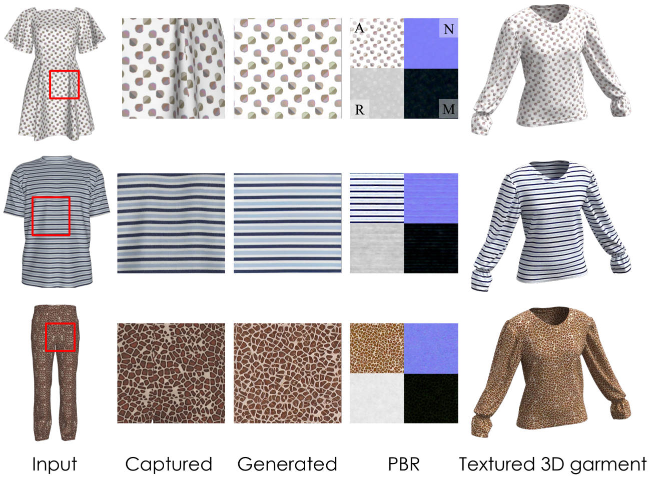

Fig. 1. **Results on texture transfer on synthetic data.** Given the input image of the 3D garment and a captured patch, our method generates a normalized texture map that is flat and tileable, along with the corresponding PBR materials. The PBR materials maps can be applied to the target 3D garment with different geometry for reliable rendering. Our model is capable of removing shadows (1st row), disentangling distortions (1st & 2nd row), and capturing physical properties (3rd row) from the input fabric texture. Note that, both the input 3D garment meshes and textures in this figure were not used for model training. See Table 1 of the main paper for qualitative results.

Table 1. **Quantitative comparison on texture images extraction from 3D garments.** Results are evaluated on synthetic testing data. The ground-truths are normalized texture images that are flat and with a unified lighting condition. Our method outperforms Material Palette [Lopes et al. 2024] across different evaluation metrics.

	LPIPS↓	SSIM↑	MS-SSIM↑	DIST↓	CLIP-s↑
Material Palette	0.66	0.27	0.31	0.45	0.89
FabricDiffusion (ours)	**0.53**	**0.32**	**0.32**	**0.32**	**0.91**

4 ADDITIONAL RESULTS

Texture transfer on synthetic data. We first validate our method using synthetic data and show the qualitative results in Figure 1. We test on textured garments with ground-truth BRDF materials, enabling controlled evaluation of geometric distortions and illumination variations. Our method reliably generates normalized textures and PBR materials. As our focus is on clothing fabrics with minimal metallic properties, we omit metallic map results for simplicity in the following experiments. Quantitative results are shown in Table 1 of the main paper.

Additional results on textures extraction. Generating a normalized texture image plays a crucial intermediate step to ensure reliable texture transfer. Figure 7 (in the main paper) shows some cases of the generated normalized textures. In Table 1, we provide a quantitative analysis using synthetic data, for which we have ground-truth textures, and compare our method with state-of-the-are methods. As we observe, our method consistently outperforms Material Palette [Lopes et al. 2024] across various evaluation metrics. As discussed in Section 2 and Section 4.1 of the main paper, personalization-based methods struggle at capturing fine-grained texture details, or disentangling the effects of distortion.

REFERENCES

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).

Ivan Lopes, Fabio Pizzati, and Raoul de Charette. 2024. Material Palette: Extraction of Materials from a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.