Supplementary materials: Woven Fabric Capture from a Single Photo

Wenhua Jin
Nanjing University of Science and Technology
China
jwh@njust.edu.cn

Beibei Wang*
Nanjing University of Science and Technology, Nankai University
China
beibei.wang@nankai.edu.cn

Miloš Hašan
Adobe Research
USA
milos.hasan@gmail.com

Yu Guo
Tencent America
USA
tflsguoyu@gmail.com

Steve Marschner
Cornell University
USA
srm@cs.cornell.edu

Ling-Qi Yan
University of California, Santa Barbara
USA
lingqi@cs.ucsb.edu

Table 1: Notation used for our BRDF model.

ω_i/ω_o	incident / outgoing direction
ω_m	macroscopic surface normal
ω_n	yarn normal
ω_t	yarn orientation
h	half vector between ω_i and ω_o
$D(\omega)$	directional distribution of microflakes
$G(\omega_i, \omega_o)$	shadowing-masking term
$\sigma(\omega)$	projected area of microflakes
ρ	microflake density
T	layer thickness (different from $\top$)
$\top$	matrix transpose (different from T)
k_s	microflake reflectance
S	3×3 matrix for microflake distribution
α	microflake roughness
$f_r(\omega_i, \omega_o)$	BRDF
$f_r^d(\omega_i, \omega_o)$	diffuse term of BRDF
$f_r^s(\omega_i, \omega_o)$	specular term of BRDF

1 WANG'S MICROFLAKE BRDF

Wang's BRDF [Wang et al. 2022] can be written as:

$$f_r(\omega_i, \omega_o) = \frac{k_s\, D(h)\, G(\omega_i, \omega_o)}{4 \cos \omega_i \cdot \cos \omega_o}, \tag{1}$$

where

$$G(\omega_i, \omega_o) = \frac{1 - e^{-T\rho(\Lambda(\omega_i)+\Lambda(\omega_o))}}{\Lambda(\omega_i) + \Lambda(\omega_o)}, \tag{2}$$

where ω_i and ω_o represent the incident and outgoing directions, respectively, h is the half vector between incident and outgoing directions, T is the thickness of the homogeneous microflake layer, and ρ represents the microflake density. We define $\Lambda(\omega) = \sigma(\omega)/\cos\omega$, and $\sigma(\omega)$ is defined below. k_s represents the reflectance of a microflake. Finally,

$$D(\omega) = \frac{1}{\pi\alpha q^2}, \text{ where } q = \omega^\top S^{-1}\omega, \text{ and } \sigma(\omega) = \sqrt{\omega^\top S \omega}, \tag{3}$$

where α is a roughness parameter, and S is a symmetric, positive definite 3×3 matrix. We use the definition $S = \mathrm{diag}(1, 1, \alpha^2)$ for a microflake oriented along the z axis, with roughness α. Other orientations are easily achieved by defining $S' = R^\top S R$ for any 3×3 rotation matrix R. We always rotate the microflake to be aligned with the predicted fiber direction.

*Corresponding author.

Table 2: Distributions used to sample the parameter space of our model. The third column notes whether the parameter has separate versions for weft and warp. $\mathcal{U}(x, y)$ represents a continuous uniform distribution in the interval (x, y). $\mathcal{V}(X)$ is a discrete uniform random variable on a finite set X. Yarn density is defined in yarns per inch, and converted internally to actual yarn size.

Parameter	Sampling Function	weft / warp
yarn pattern	$W = \mathcal{V}(\{0, 1, 2, 3, 4\})$	No
yarn density	$y = \mathcal{U}(45, 335)$	Yes
roughness	$\alpha = \mathcal{U}(0.1, 1)^2$	Yes
diffuse albedo	$k_d = \mathcal{U}(0, 1)$	Yes
specular albedo	$k_s = (k_d)^m, m = \mathcal{U}(0, 1)$	Yes
scaling factor	$\beta = \mathcal{U}(0.1, 2)$	Yes
blending weight	$w = \mathcal{U}(0, 1)$	No
noise level	$Q = \mathcal{V}(\{0 - 10\})$	No

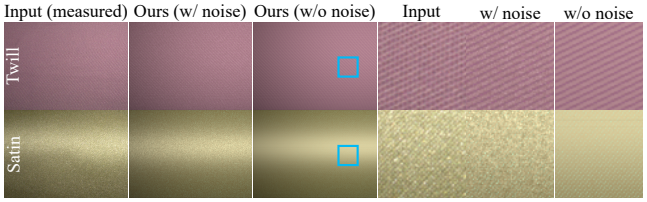

Figure 1: Comparison with and without added noise for highlight imperfections.

2 OUR FULL MODEL

Our fabric shading model includes a specular and a diffuse term: $f_r(\omega_i, \omega_o) = f_r^d(\omega_i, \omega_o) + f_r^s(\omega_i, \omega_o)$, where the specular term is defined in Eqn. (1) and the diffuse term is defined in Eqn. (1) (main

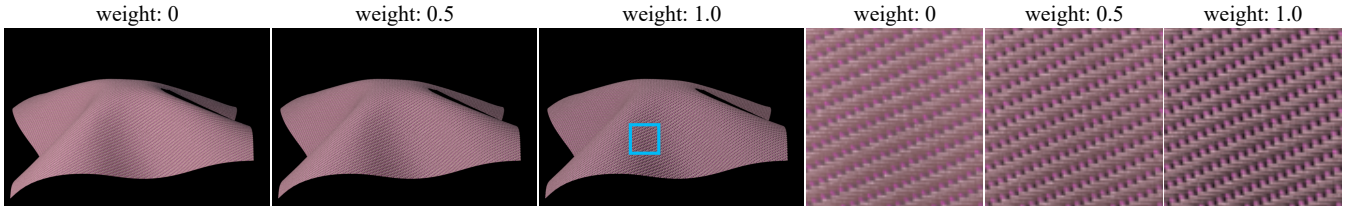

Figure 2: Comparison of rendered results when setting weights to different values.

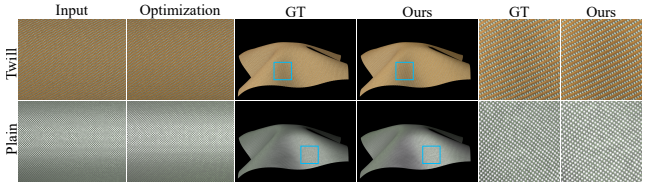

Figure 3: Our model can handle fabrics with different warp and weft colors.

paper), resulting in the full model:

$$f_r(\omega_i, \omega_o) = U_s(\xi)\frac{k_s\, D(h)\, G(\omega_i, \omega_o)}{4\cos\omega_i \cdot \cos\omega_o} + w\frac{k_d\,\langle\omega_i \cdot \omega_m\rangle}{\pi\,\langle\omega_i \cdot \omega_n\rangle} + (1-w)\frac{k_d}{\pi}, \tag{4}$$

where $U_s(\xi)$ is a unit mean exponential random variable defined on $[0, +\infty)$ to introduce imperfections in the specuar, and is defined as:

$$U_s(\xi) = -\log(R(\xi)), \tag{5}$$

where R is a pseudo random number generator (e.g., Tiny Encryption Algorithm) to generate random numbers defined on $(0, 1]$ with seed ξ. The seed ξ is set as the product of the yarn segment index and the noise level Q. Note that the range of $U_s(\xi)$ is $[0, +\infty)$ and the mean value is 1 to keep the specular term consistent.

We introduce a heightfield scaling factor β to adjust the heightfield of yarns, which will affect both the normal (n_x, n_y, n_z) and orientation (t_x, t_y, t_z). We also introduce some imperfections on the normal using a random value $U_n(\xi)$. The disturbed normal is defined as:

$$\begin{aligned}\omega_n &= (\frac{\beta U_n(\xi) n_x}{n_z}, \frac{\beta U_n(\xi) n_y}{n_z}, 1),\\ \omega_n &= \frac{\omega_n}{||\omega_n||}.\end{aligned} \tag{6}$$

The definition of $U_n(\xi)$ is different from $U_s(\xi)$ and is defined as:

$$U_n(\xi) = \frac{\beta - 0.1Q + 0.15QR(CP(\xi))}{\beta}. \tag{7}$$

Q is the noise level and set as $[0, 10]$. A larger value of Q indicates more noise. R is the same as in Eqn. (5). P is a one-dimensional Perlin noise function and calculated on the weft and warp separately. C is a constant value to scale the Perlin noise and set as 100 in our implementation. The seed ξ is set as the yarn segment index.

To ensure the scaled normal and orientation are still perpendicular, the orientation after scaling is defined as:

$$\begin{aligned}\beta_o &= \frac{-\omega_n^z \omega_t^z}{\omega_t^x \omega_n^x + \omega_t^y \omega_n^y},\\ \omega_t &= (\beta_o \omega_t^x, \beta_o \omega_t^y, \omega_t^z),\\ \omega_t &= \frac{\omega_t}{||\omega_t||}.\end{aligned} \tag{8}$$

The ω_t is used to compute the matrix S in Eqn. (3).

In Fig. 1, we compare our results (with noise and without noise) with photos. By comparison, we find that the noisy highlights are more realistic.

We show a comparison between the rendered results when setting weights in the diffuse term to different values in Fig. 2.

3 MORE RESULTS

3.1 Results of our inverse model

Synthetic data. In Fig. 3, we show the results of our inverse model on synthetic data. Our model is able to support fabrics with different warp and weft colors.

Real data. In Fig. 4, we provide more and larger results on measured data than shown in the main paper. Since there are no ground truth parameters for the measured data, we compare the visual match between the input image and the rendered image with the estimated parameters. Our inverse model, including the FabricNet and the optimization, produces closely matching results to the input images. The renderings with the draped cloth mesh also show a plausible appearance.

Novel view. To verify the correctness of our method, we recaptured the fabrics at novel view and rendered them in the same configuration. The comparison results are shown in Fig. 5.

3.2 Comparison with other works

We compare our method with MATch [Shi et al. 2020] and Henzler et al. [2021]. The input images for both methods were taken with the same camera (iPhone12) as our approach. The fabric samples are placed on a flat surface, since these methods only support flat surfaces. We use the flash as the light source in both methods. The camera distance for our method and competitor methods is similar, following a rule: a 10cm × 10cm fabric sample covers the camera view for all captures.

Comparison with other procedural models. More results to compare our method and MATch [Shi et al. 2020] are provided in Fig. 6. We also provide all the intermediate results or inputs in Fig. 6, which

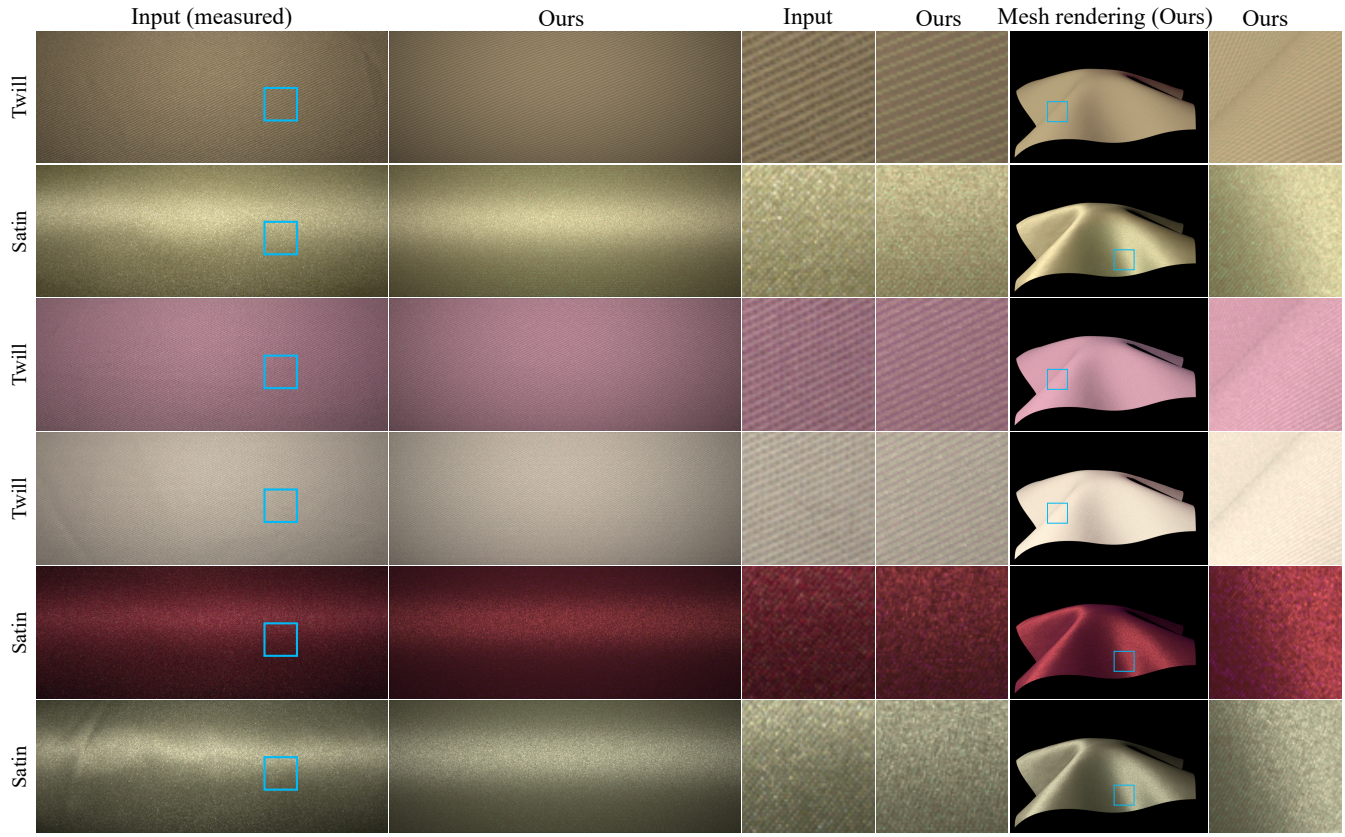

Figure 4: Given an input image captured with our measurement configuration, our inverse model is able to produce closely matching results. The rendered results on the draped cloth mesh also show a natural appearance.

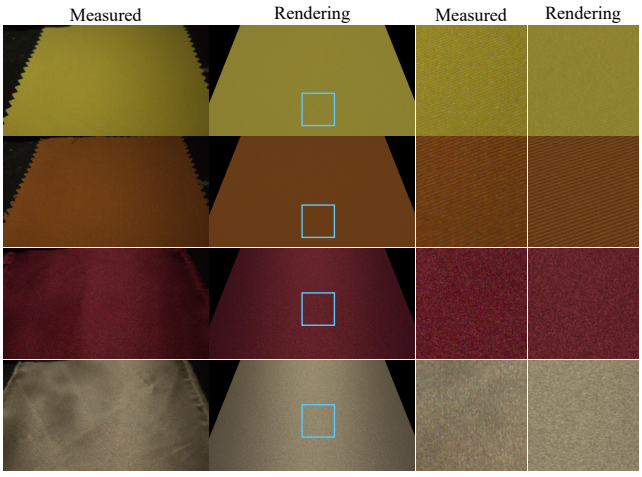

Figure 5: We captured the fabrics from a novel view and compared them with our reconstructed results rendered from the same view, achieving a similar appearance at a local scale (we do not attempt to model the imperfect macroscopic shape of the real cloth).

are not in the main paper. We show the SVBRDF maps and the two Substance procedural graphs designed for fabrics (fabric suit and smooth silk), besides the rendered results.

Comparison with a general SVBRDF model. We provide more results to compare our model and Henzler et al. [2021] in Fig. 7. We also provide the SVBRDF maps besides the rendered results, which are not in the main paper.

3.3 Ablation study

Cylinder vs. Plane. In Fig. 8, we show the results using our configuration and using the plane as the setup, we provide structure similarity index measure (SSIM) to measure the effect of both methods. Better results can be obtained using our method.

4 DISCUSSION AND LIMITATIONS

Generality. Since we only use local shading in a fixed scene with no occlusion for inverse rendering, we implemented our differentiable renderer (like the rest of the solver) in PyTorch. Other frameworks (e.g., Mitsuba 2 [Nimier-David et al. 2019]) would be feasible as well.

Scope. Our model only focuses on woven fabrics. However, we believe that it will inspire more types of fabric capture in the future

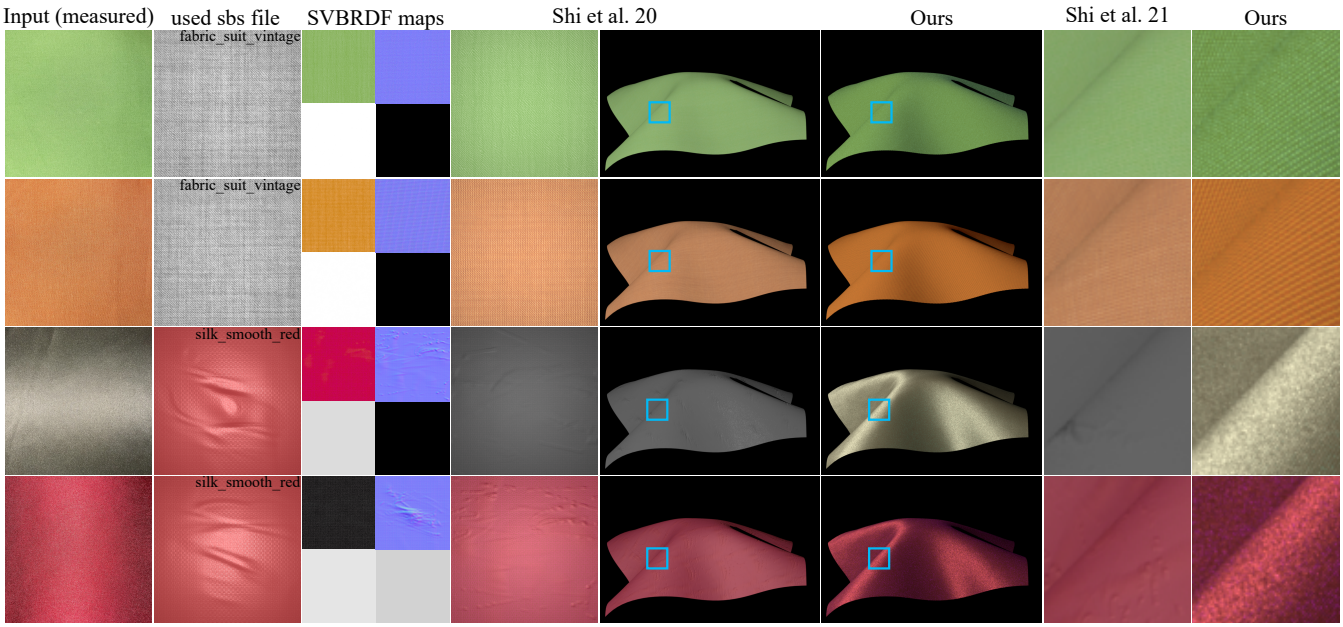

Figure 6: Comparison between our method and MATch [Shi et al. 2020].

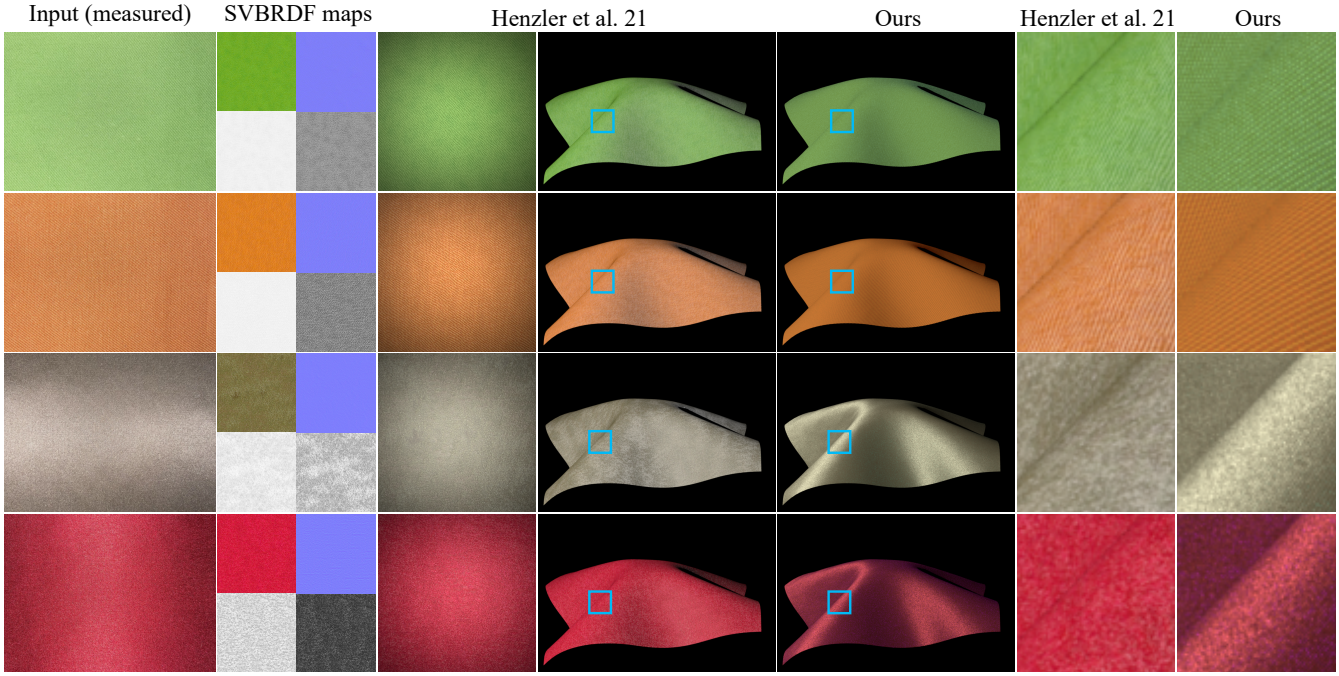

Figure 7: Comparison between our method and a SVBRDF recovery model [Henzler et al. 2021].

and our techniques (geometry and shading component, differentiable rendering, neural initialization) will transfer to these cases.

Expressiveness of the forward model. Our current model does not consider gaps between yarns, nor yarns with varying thickness, which are common in linen fabrics as shown in the example in Fig. 9 (Top). These features would be fairly easy to add to our model, and we leave it for future work. Our model also does not implement one feature present in Irawan and Marschner's model [2012]: their κ parameter can be negative, giving yarns negative curvature and

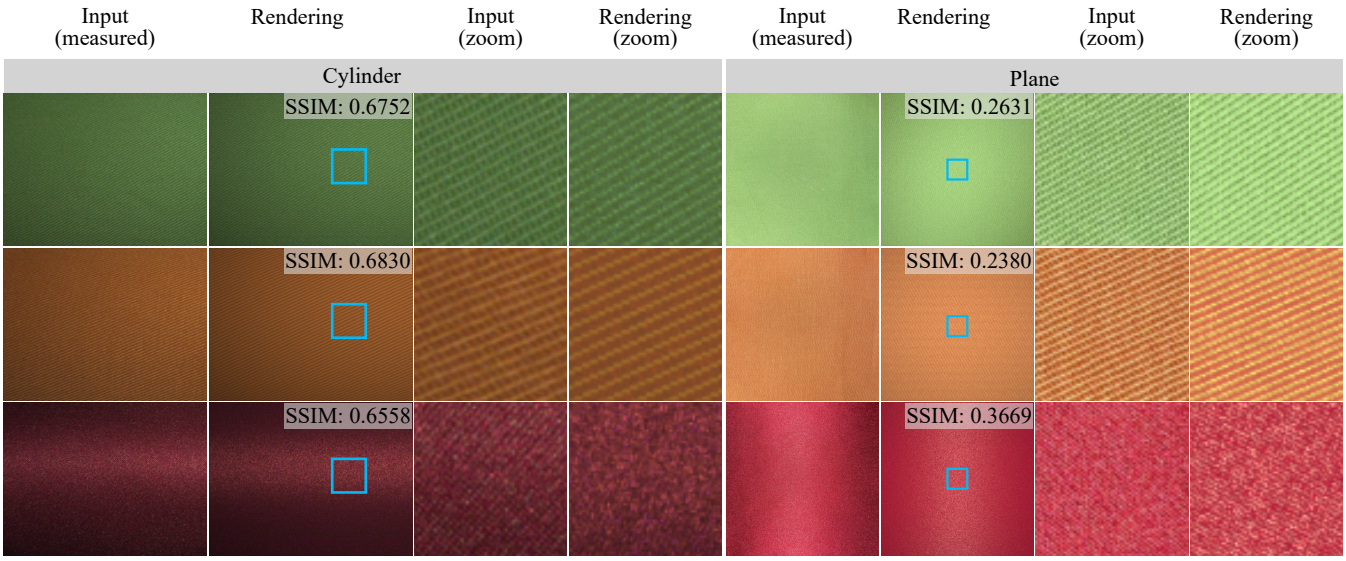

Figure 8: Using our cylinder configuration gives better results than the flat plane configuration.

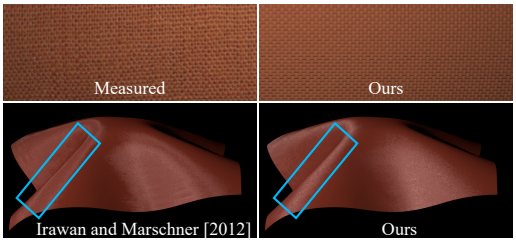

Figure 9: Limitations. Our model does not yet handle fabrics that have large gaps (top) or with double-highlights (bottom).

Figure 10: We rotate different angles when placing the fabric on the cylinder, and calculate the corresponding Gram matrix loss, the quality of the recovered fabrics will decrease as the rotation angle increases.

causing the double-highlight appearance, as shown in Fig. 9 (bottom). This feature could also be added in the future. Our model is at the yarn level, and does not consider flyaway fibers, which could be modeled as explicit strands.

Axis-aligned patterns and pattern variety. In our geometric model, we only consider axis-aligned yarns. Our approach will not work well with input measurements rotated at arbitrary angles. In Fig. 10 in the supplementary material, we show an error curve with varying degrees of warp misalignment in the sample. We trained our model only on three patterns (five with rotations), which are very common in practice, though other patterns exist and our model could support them in the future with minor effort.

Domain gap. There is a domain gap between synthetic and real data. Some domain gap exists partly due to the forward model capabilities, and partly due to the sampling distribution for synthetic data. The latter issue is addressed by the optimization step.

Lack of yarn-level displacement. Our current model does not implement yarn-level displacement, which has a strong effect in thicker fabrics (e.g., jacquard), nor non-regular thick yarns (e.g., linen) where yarn-to-yarn occlusion plays a significant role at foreshortening angles. More advanced differentiable rendering could support these effects.

Lack of transmission. Our model does not consider transmission effects for now. Since the forward model can be easily extended to handle transmission, we believe our model will be extensible to these effects, but will need a modified setup to capture transmission images.

REFERENCES

Philipp Henzler, Valentin Deschaintre, Niloy J Mitra, and Tobias Ritschel. 2021. Generative Modelling of BRDF Textures from Flash Images. *ACM Trans. Graph.* 40, 6 (2021), 1–13.

Piti Irawan and Steve Marschner. 2012. Specular Reflection from Woven Cloth. *ACM Trans. Graph.* 31, 1 (2012), 1–20.

Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A Retargetable Forward and Inverse Renderer. *ACM Trans. Graph.* 38, 6 (2019), 1–17.

Liang Shi, Beichen Li, Miloš Hašan, Kalyan Sunkavalli, Tamy Boubekeur, Radomir Mech, and Wojciech Matusik. 2020. MATch: Differentiable Material Graphs for Procedural Material Capture. *ACM Trans. Graph.* 39, 6 (2020), 1–15.

Beibei Wang, Wenhua Jin, Miloš Hašan, and Ling-Qi Yan. 2022. SpongeCake: A Layered Microflake Surface Appearance Model. *ACM Trans. Graph.* (2022), 1–15.