

# Woven Fabric Capture from a Single Photo

Wenhua Jin

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

Yu Guo

Tencent America  
USA  
tfsguoyu@gmail.com

Beibei Wang\*

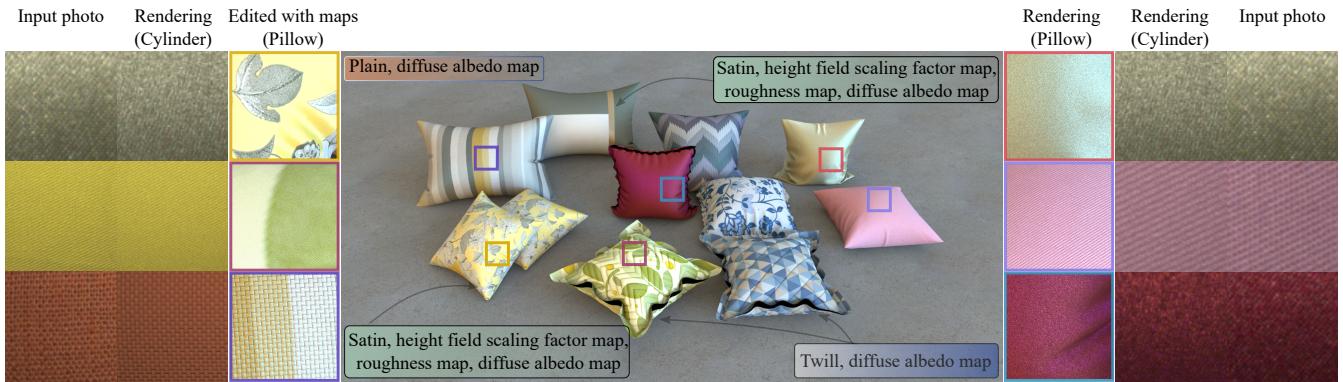
Nanjing University of Science and  
Technology and Nankai University  
China  
beibei.wang@nankai.edu.cn

Miloš Hašan

Adobe Research  
USA  
milos.hasan@gmail.com

Ling-Qi Yan

University of California, Santa  
Barbara  
USA  
lingqi@cs.ucsb.edu



**Figure 1:** Given an input photo of a woven fabric sample, our approach estimates the parameters for our woven fabric material model using an initial neural network estimate, further refined by a differentiable rendering optimization. Re-rendered results with estimated parameters closely match the input photos (columns on the left and right). The resulting fabric parameters can be used for rendering directly, or can be further edited to control the final appearance, as shown in the rendered scene.

## ABSTRACT

Digitally reproducing the appearance of woven fabrics is important in many applications of realistic rendering, from interior scenes to virtual characters. However, designing realistic shading models and capturing real fabric samples are both challenging tasks. Previous work ranges from applying generic shading models not meant for fabrics, to data-driven approaches scanning fabrics requiring expensive setups and large data. In this paper, we propose a woven fabric material model and a parameter estimation approach for it. Our lightweight forward shading model treats yarns as bent and twisted cylinders, shading these using a microflake-based bidirectional reflectance distribution function (BRDF) model. We propose

a simple fabric capture configuration, wrapping the fabric sample on a cylinder of known radius and capturing a single image under known camera and light positions. Our inverse rendering pipeline consists of a neural network to estimate initial fabric parameters and an optimization based on differentiable rendering to refine the results. Our fabric parameter estimation achieves high-quality recovery of measured woven fabric samples, which can be used for efficient rendering and further edited.

## CCS CONCEPTS

- Computing methodologies → Rendering; Reflectance modeling.

## KEYWORDS

microflake, fabric rendering

## ACM Reference Format:

Wenhua Jin, Beibei Wang, Miloš Hašan, Yu Guo, Steve Marschner, and Ling-Qi Yan. 2022. Woven Fabric Capture from a Single Photo. In *SIGGRAPH Asia 2022 Conference Papers (SA '22 Conference Papers)*, December 6–9, 2022, Daegu, Republic of Korea. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3550469.3555380>

\*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SA '22 Conference Papers, December 6–9, 2022, Daegu, Republic of Korea

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9470-3/22/12...\$15.00

<https://doi.org/10.1145/3550469.3555380>

## 1 INTRODUCTION

Woven fabrics are an important material in rendering applications from video games to interior design. Accurate fabrics affect realism significantly, and a common indicator of an image being rendered (as opposed to a photograph) is that fabrics appear unrealistic. However, the capture and rendering of woven fabrics are challenging tasks, due to the very specific internal structure of the material (repeated weave patterns consisting of yarns, which themselves consist of fibers), and due to the fact that microfacet-based shading models commonly used in the rendering industry [Burley 2012; Walther et al. 2007] are not a close fit for representing fabric reflectance.

Many previous methods target fabric rendering and capture, but they generally require complex devices or pipelines. Some methods capture the appearance through data-driven approaches such as bidirectional texture functions (BTFs) [Kautz et al. 2007]. Other methods use micro-CT scanning approaches to capture the micro-structures of the fabrics at the fiber level, which can produce highly-detailed renderings [Zhao et al. 2011], but is expensive to capture and render, and does not provide the optical properties of the fibers (only their geometry). Another option is applying generic shading models not meant for fabrics, as single-image capture methods exist for such models [Deschaintre et al. 2018; Guo et al. 2021], but these produce less realistic results than using a fabric-focused model.

In this paper, we propose a procedural fabric parameter estimation approach. We introduce a lightweight forward rendering model for fabrics, consisting of a geometric component that represents the yarns of the fabric using curved cylinders, building upon the work of Irawan and Marschner [2012], and a shading component based on a custom microflake BRDF extending recent work from Wang et al. [2022]. Our forward model produces realistic renderings on a large class of woven fabrics and is straightforward to adapt for differentiable rendering.

Next, we propose an inverse rendering framework to estimate the parameters of our forward model from a single photograph of a material sample taken using a simple and inexpensive fabric capture setup with a cylindrical surface and a single light and camera (a phone camera in our results). We first estimate the fabric parameters from the input image using a neural network trained on a large synthetic dataset generated with our forward shading model.

These parameters are then further refined via differentiable rendering to produce even more accurate results (see Fig. 1). To summarize, our contributions include:

- A forward rendering model for woven fabrics that produces realistic renderings and is simple enough to allow for straightforward differentiable rendering,
- a simple inexpensive setup to capture real fabric data,
- a neural network to predict initial fabric parameters from a single image, and
- an optimization strategy using differentiable rendering to refine the estimate.

The resulting texture maps produced by our method can be further edited based on artistic needs (Fig. 1). We believe our method can be used in practical 3D content creation workflows and further push the realism of virtual characters and environments.

## 2 RELATED WORK

*SVBRDF capture.* Different neural network structures and training strategies have been proposed for single-image per-pixel spatial varying bidirectional reflectance distribution function (SVBRDF) acquisition, including convolutional neural network (CNN) on limited labeled SVBRDF training pairs [Li et al. 2017] or unlabeled data [Ye et al. 2018], a combination of U-Net and a fully-connected network for better feature extraction [Deschaintre et al. 2018; Li et al. 2018], fine-tuning for large planar surfaces under ambient lighting [Deschaintre et al. 2020] or highlight-aware convolution to deal with highlights regions [Guo et al. 2021].

Another group of previous works targets stationary textured materials, using a texture descriptor [Aittala et al. 2016] for synthesis, a generative adversarial network (GAN) for both SVBRDF recovery and synthesis [Zhao et al. 2020], or a latent space-based recovery and editing [Henzler et al. 2021].

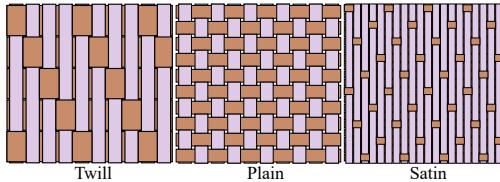
Unlike these SVBRDF capture methods, our approach is procedural: instead of computing the texels of material texture maps, we estimate a small parameter vector used to build these maps. Our approach avoids the artifacts common in SVBRDF estimation, at the cost of not supporting arbitrary textures that the procedural model cannot describe. In the context of our woven fabric application, our procedural model already covers an extensive range of appearances. Another key difference is that we propose a shading model appropriate for fabrics, while all previous single-image methods estimate the parameters maps of a generic shading model (albedo, normal, roughness, etc.), which is not optimal for fabric rendering.

*Procedural material parameter estimation.* Several approaches were proposed for predicting procedural material parameters. Hu et al. [2019] introduced an inverse procedural modeling framework to automatically select procedural models and estimate parameters from an input image. The latter part is implemented as a neural network that learns a mapping from images to material parameters from synthetic data. This approach is taken by all subsequent inverse procedural material methods, including ours.

Guo et al. [2020] proposed a procedural material parameter estimation approach from photographs using a Bayesian framework. Shi et al. [2020] apply a similar approach to capture materials into production-grade procedural material models, based on a differentiable version of Adobe Substance material graphs. Both Guo et al. and Shi et al. use a style transfer loss based on the Gram matrices of VGG features. We also use this loss, which is by now standard for single-image material capture; we also combine it with an  $L_1$  loss between down-sized images.

The above three methods are designed for general materials and are not optimal for fabrics. We follow a similar framework, but with a number of improvements designed for high-accuracy woven fabric reconstruction: a different geometric model treating yarns as bent cylinders, a different shading model based on a microflake layer, and a different capture setup.

*Microflake models.* Jakob et al. [2010] first propose the microflake framework to define anisotropic participating media. The microflake model can represent fabrics [Zhao et al. 2011], foliage [Loubet and Neyret 2018], and special pigments [Guillén et al. 2020].



**Figure 2: Several fabric patterns studied in this paper.**

Heitz et al. [2015] introduced symmetric GGX (SGGX), a compact representation of microflake distributions using  $3 \times 3$  positive-definite matrices. Recently, Wang et al. [2022] proposed the Sponge-Cake model, which defines each layer as a volumetric medium described by a microflake distribution. Our shading model is a subset of the SpongeCake model, with further additions to enhance fabric realism. Our version of the model allows for efficient data generation and inverse rendering.

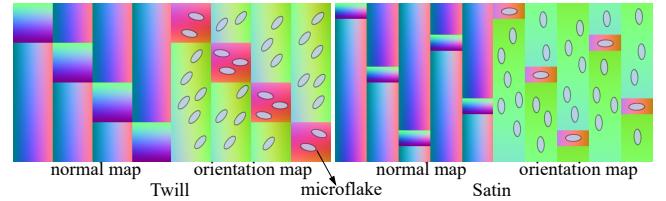
*Surface fabric models.* Fabrics have been modeled with specialized surface BRDFs (e.g., [Adabala et al. 2003] [Irawan and Marschner 2012] [Sadeghi et al. 2013] on a smooth surface, and they are capable of reproducing a wide range of appearances. This type of model is simpler than volumetric models [Zhao et al. 2011] or those requiring detailed geometric modeling [Montazeri et al. 2020], although the results might be less realistic. Our forward model is also a surface-based BRDF model but with some key differences. Our forward fabric model is differentiable, allowing for inverse rendering.

*Single-image Fabrics recovery.* Schröder et al. [2015] propose a pipeline to reverse-engineer cloth and estimate a parameterized cloth model from a single image. Their model is able to achieve fiber-level detail and produces visually plausible results. However, their model relies on manual selection of model parameters, while our model is automatic. Later, Guarnera et al. [2017] estimated the yarn parameters in the spatial and frequency domain, resulting in high-quality yarn-level recovery. They do not propose a novel forward fabric model and use generic ones, while our forward model is specific to woven fabrics. Furthermore, our inverse pipeline is flexible, allowing modifications to the forward model and relying on differentiable rendering for parameter estimation. Recently, Wu et al. [2019] estimated large-scale yarn geometry by yarn layout extraction and the fine-scale fiber details. Their model can capture fiber-level detail, but the time cost is high. Please refer to the survey by Castillo et al. [2019] for more work in this area. Similar to all these approaches, we focus on woven fabric recovery, rather than knitted fabrics [Kaspar et al. 2019; Trunz et al. 2019]. Unlike the previous methods, both our capture setup and the inverse model are lightweight.

Rodríguez-Pardo et al. [2019] propose to recover the macroscopic color pattern textures for woven fabrics rather than the fabric parameters. Our model does not consider these patterns for now, but could be combined with their approach.

### 3 FORWARD MODEL

Fabrics are constructed from yarns via manufacturing technologies like weaving and knitting. Yarns are complex across several scales,



**Figure 3: Visualization of the normal maps and orientation maps for twill and satin patterns. The microflakes are aligned with the orientations.**

and generally consist of micron-diameter fibers. We focus on woven fabrics, which are manufactured by a process that starts with stretched parallel *warp yarns*, through which *weft yarns* are woven, passing over or under the warp yarns in a repeating pattern. Three weaving patterns are shown in Fig. 2; the warp is vertical and the weft is horizontal. In this paper, we focus on these three patterns, plus their 90-degree rotations, though others could easily be added.

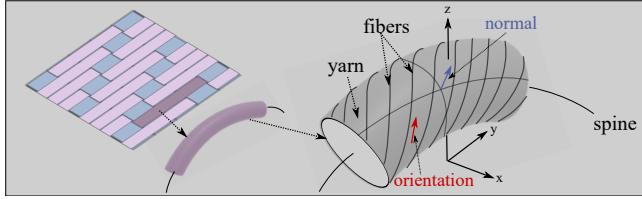
In our paper, we model woven fabrics at the yarn level. We do not resolve specific fibers within the yarns, though we still model the (optionally twisted) fiber direction. More precisely, we approximate the yarn geometry with smooth bent cylinders, which yields normal vectors, tangents and other information, used as input to a reflectance model. Our procedural model thus has two components: a *geometric* model that generates normal and fiber orientation maps based on the weave pattern and additional parameters, and a *shading* model that assigns a reflectance function to each surface point, using the normal and orientation information local to that point and the optical properties of the yarns. Both components are controlled by a number of parameters, and our inverse rendering solution estimates these parameters from material sample photographs.

#### 3.1 Geometric model based on curved cylinders

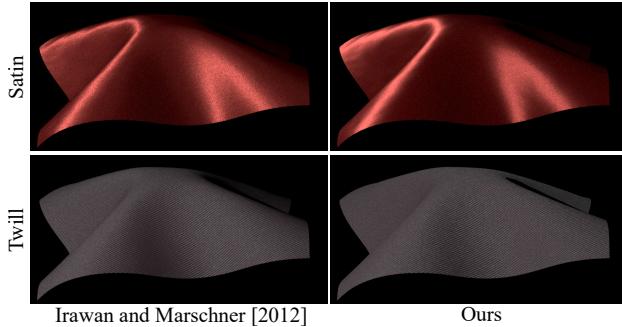
Our geometric model extends the work of Irawan and Marschner [2012]. Their model is based on an analysis of specular scattering from fibers that are spun into yarns and then woven into fabric based on a given weaving pattern. Their method is based on the curved cylinder model: each visible yarn segment is approximated as a bent cylinder, whose width, length and curvature depend on the pattern and other parameters.

We make several modifications to their model. Most importantly, in our method the resulting array of curved cylinders gives explicit height, normal and orientation maps, used for subsequent pointwise shading, while in Irawan and Marschner's approach, explicit construction of these maps is bypassed and their shading model works differently. Specifically, their specular shading is designed such that it only depends on the local yarn curvature. We find it more flexible to explicitly define the local yarn geometry.

We model each yarn as a cylinder constructed along a circular or elliptical arc, as shown in Fig. 4. Given this yarn model, the normal and orientation vectors can be computed analytically (we could also compute the heightfield values, though they are not currently used). In practice, we generate a  $2048 \times 2048$  map for both normals and orientations, though this resolution is configurable. In Fig. 3, we visualize the normal and orientation maps of several yarns. The



**Figure 4:** The twist angle  $\psi$  and a maximum inclination angle  $u_{\max}$  define a curved cylinder, from which the orientation map and normal map can be generated.



**Figure 5:** Comparison between Irawan and Marschner [2012] and our forward model on several fabrics. Our model produces close results to their model.

microflakes are aligned with the orientations and perpendicular to the normals. Since our method models yarns with curved cylinders, the angle at which a weft dives under warp at yarn crossings is controllable (and optimizable). We also introduce some imperfections to our orientation map and normal map by using a random value  $U(\xi)$  to scale the height field.

We compare our model with Irawan and Marschner [2012] on twill and satin patterns in Fig. 5. We find that our model can produce realistic fabrics from a far-field view, similar to theirs.

### 3.2 BRDF model based on SGGX microflakes

We propose to combine the above procedural woven fabric geometry model with a custom BRDF, instead of using the shading model given by Irawan and Marschner. We use the microflake framework, which is easily differentiable and gives some additional flexibility.

We found that a good model for the light reflection from a yarn can be obtained by treating it locally as a homogeneous volumetric medium, whose scattering is described with the SGGX microflake phase function [Heitz et al. 2015]. This formulation models fibers as long, skinny ellipsoids; the ellipsoids are aligned with the fiber orientation, which in our case is always orthogonal to the shading normal. The microflakes are randomly oriented with normals taken from the ellipsoid normal distribution.

Wang et al. [2022] derived the closed-form single-layer BRDF for single scattering under the SGGX microflake framework. They also show that the corresponding multiple scattering BRDF, while not closed-form, has a shape similar to the single-scattering BRDF, only with different parameters. This suggests that we can reasonably

model the scattering from a yarn (single and multiple) using Wang's single-scattering closed-form BRDF, as long as its parameters are chosen well (which is addressed by our inverse rendering).

Our full model is based on Wang's BRDF [2022] (detailed in the supplementary material), with several extensions to enhance the capabilities for representing fabrics. Our model replaces the multiple-scattering lobe with weighted Lambertian lobes for original and modified normal, which makes the model simpler and easier to fit.

For a given surface location on the fabric, let  $\omega_i$  and  $\omega_o$  be the incident (light) and outgoing (camera) directions in world space. Let  $\omega_m$  be the smooth macroscopic surface normal (e.g. interpolated from mesh vertex normals) and let  $\omega_n$  be the final yarn normal predicted by our geometric model.

Our fabric shading model includes a specular and a diffuse term:  $f_r(\omega_i, \omega_o) = f_r^d(\omega_i, \omega_o) + f_r^s(\omega_i, \omega_o)$ . The diffuse term is defined as:

$$f_r^d(\omega_i, \omega_o) = w \frac{k_d \langle \omega_i \cdot \omega_m \rangle}{\pi \langle \omega_i \cdot \omega_n \rangle} + (1-w) \frac{k_d}{\pi}, \quad (1)$$

where  $k_d$  is the diffuse albedo. The diffuse term is defined as a weighted sum of the Lambertian term under the original surface shading normal and the Lambertian term using the yarn normal. The goal of this weighted blending is to approximate the softened shading caused by subsurface scattering in the yarns; using only the yarn normal gives too much contrast to the inter-yarn shading, making them appear completely opaque, while using only the original surface normal gives an unrealistic look with little detail. A weight  $w$  is used for the blending, and will be estimated by inverse rendering along with the other parameters. We show a comparison between the rendered results when setting weights to different values in Fig. 2 in the supplementary material. Note that the diffuse term is non-reciprocal; however, the use of mapped normals distinct from the geometric normal is known to cause non-reciprocity in any case, so this cannot easily be avoided.

Our specular term  $f_r^s(\omega_i, \omega_o)$  is based on Wang's BRDF, where we set the value  $T\rho = 2$ ; we could also estimate this value using our inverse rendering, but we found no benefits compared to using a constant value. We orient the fiber-like microflakes to align with the local fiber orientation (which is always orthogonal to the yarn normal). We also multiply the specular term by a unit mean exponential random variable  $U(\xi)$  defined on  $[0, +\infty)$ , which introduces imperfection in the highlights to enhance realism.

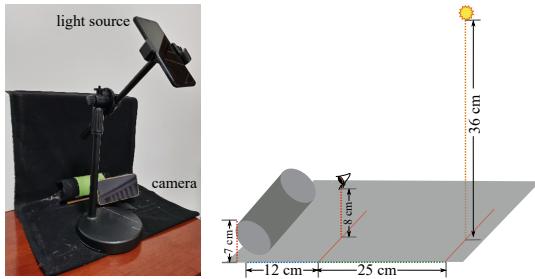
To summarize, our woven fabric shading model consists of the following parameters (see Table 1): a discrete weave pattern, diffuse/specular albedo for both weft and warp yarns, roughness for both weft and warp yarns, yarn size for both weft and warp yarns, a height field scaling factor for both weft and warp yarns, a twist angle of the fiber, Lambertian term normal map blending weight, randomness on the specular term, randomness on the orientation and normal maps, and a noise level performed on the height field scaling factor to control the intensity of the orientation / normal map randomness.

## 4 INVERSE PARAMETER ESTIMATION

In this section, we present our inverse approach to estimate, given a sample photograph, the set of fabric parameters controlling the forward model explained in the previous section. First, we describe

**Table 1: Parameters in our BRDF model.** Warp yarns are denoted with  $v$ -subscripts (vertical) and weft yarns with  $h$ -subscripts (horizontal). Top two parameters affect yarn geometry, the rest affect reflectance.

|                      |  |
|----------------------|--|
| $s^h, s^v$           | yarn size for weft and warp yarns                  |
| $\beta^h, \beta^v$   | heightfield scaling factor for weft and warp yarns |
| $k_d^h, k_d^v$       | diffuse albedo for weft and warp yarns             |
| $k_s^h, k_s^v$       | specular albedo for weft and warp yarns            |
| $\alpha^h, \alpha^v$ | roughness for weft and warp yarns                  |
| $\psi$               | fiber twist angle                                  |
| $u_{\max}$           | maximum inclination angle                          |
| $w$                  | weight for the Lambertian term blending            |
| $U_s(\xi)$           | randomness on the specular term                    |
| $U_n(\xi)$           | randomness on the normal and orientation           |
| $Q$                  | normal / orientation randomness level              |



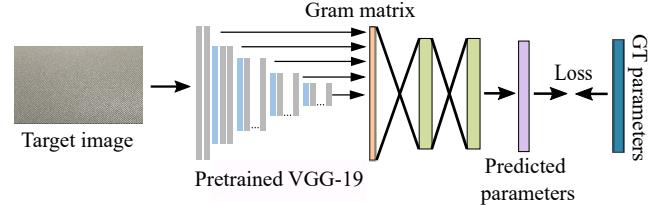
**Figure 6: The configuration to measure the real fabric data.** We use a cell phone flash as a light source, though any point light source can be used. The fabric sample is wrapped onto a cylinder. The distances between elements are measured, as well as the cylinder diameter and camera field of view, so we can reconstruct the same setup in synthetic renderings.

our single-shot capture setup. Next, we propose a neural network trained to estimate fabric parameters given an input image. Finally, we propose an optimization process via differentiable rendering to refine the results further.

We find that predicting the parameters of our procedural forward model is preferable to predicting spatially-varying textures of local shading parameters, as the latter option is very under-constrained with a single image, and suffers from ambiguity issues between the lighting and the materials, leading to polluted SVBRDF maps. Furthermore, our resulting textures are inherently seamless, making it easy to cover a large surface. Thus, we always produce valid and seamless fabric materials.

#### 4.1 Measurement setup for fabrics

We propose a simple configuration to capture real fabric samples with a single photograph, as shown in Fig. 6. We use a cell phone camera and a point light source (in this case we use the flash of a second cell phone as the point light, though any white point source could be used). The captured raw images have a 4K resolution, and we crop and downsample them to a resolution of  $300 \times 800$ .



**Figure 7: Our network architecture.**

The fabric samples are put on top of a cylinder-shaped holder. We find that this setup makes it easier to get a clear view of the specular highlight shape across many material parameters, compared to overhead views of flat samples and/or using a collocated light and camera. The main reason is that fiber highlights often appear at angles different from surface-like materials.

We measure the relative locations of the cylinder, camera and light source, so we can synthetically render images in the same setup. The diameter of the cylinder and the field of view of the camera are known. We calibrate the light brightness using a target with known gray level. We capture linear images with a single exposure; occasional pixel clipping is not a problem as it is applied to our synthetic renders as well.

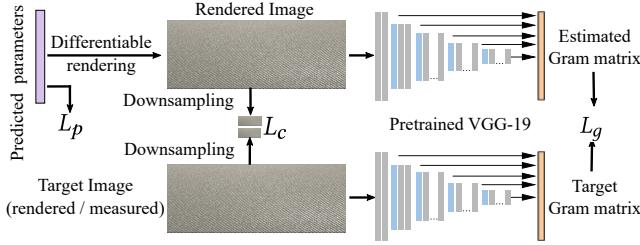
We notice that our photographs have a smooth falloff from the image center towards the corners, due to lens vignetting and directional falloff of the light source. We simulate this effect by multiplying our corresponding rendered images with a Gaussian falloff located at the image center with  $\sigma = 450$  pixels, which empirically fits the effect well in our setup (though other cameras/lights will need adjustment to  $\sigma$ ).

#### 4.2 Neural network for fabric parameter prediction

In the next step of our inverse pipeline, we train a neural network, FabricNet, which maps the measurement images into approximate parameter vectors. Our forward shading model can generate a large amount of synthetic data mapping images to parameters, which we use for training.

*Architecture.* The architecture of our network is shown in Fig. 7. With an image as input, we first use a pretrained VGG-19 network, and select five layers after the pooling operations and obtain their Gram matrices (flattened into a vector of size 610304). This represents the input image's features, but is global and shift-invariant (it essentially approximates the means and covariances of the VGG feature activations at five selected layers). The flattened Gram matrix is fed into a fully connected (FC) module, which includes two intermediate layers (256 nodes per layer) with LeakyReLU activation function. The final FC layer outputs the predicted parameters (21 channels for our forward model). A sigmoid is used to output the final parameters in the  $[0, 1]$  range, after which they are remapped into their respective ranges.

*Dataset generation.* We use three weave patterns (twill, satin and plain weave), together with 90 degree rotations of the twill and satin, giving five patterns in total. We generate 2,000 images for each pattern with our shading model (Sec. 3). We sample the fabric parameter space to generate these images, as shown in Table S2



**Figure 8:** We perform an optimization step, initialized with the network-predicted parameters, using three losses: a Gram matrix loss  $L_g$ , a pixel loss  $L_c$  and a prior loss  $L_p$ , to improve the quality further.

(Table 2 in the supplementary material). We sample the parameter space with some priors, for example, the yarn size and roughness have different ranges for satin and twill. Instead of independently sampling specular albedo, we derive it as a random power of the diffuse albedo, which has the effect of shifting the hue towards white. This produces more realistic training image distributions (avoiding e.g. pink fabric with green highlights).

Note that although our forward model also supports varying microflake layer thickness, we currently do not vary it, and simply use typical values. We currently set the thickness (product  $T\rho$ ) to 2 and set the twist angle to -30 degrees for the twill pattern during data generation, and no twist for other patterns.

*Training.* The loss function for network training is the  $\mathcal{L}_1$  difference between the ground truth parameters and the predicted parameters from the network. Our network is implemented in the PyTorch framework. We apply the Adam solver, where the learning rate is set as 0.0001. The training samples are fed into the network in a batch size of 32. Only FC weights are updated during training (VGG weights are frozen). Training took eleven hours on a single NVIDIA 2080Ti GPU.

### 4.3 Differentiable rendering and optimization

The network-predicted fabric parameters could be used directly, but there are common issues, like color bias or imperfect highlight shape match. To further improve the prediction quality, we propose an optimization step to refine the parameters via differentiable rendering of our shading model. We get a better prediction after the optimization, as shown in Fig. 12. In practice, we find the pretrained model benefits the optimization significantly, instead of biasing it.

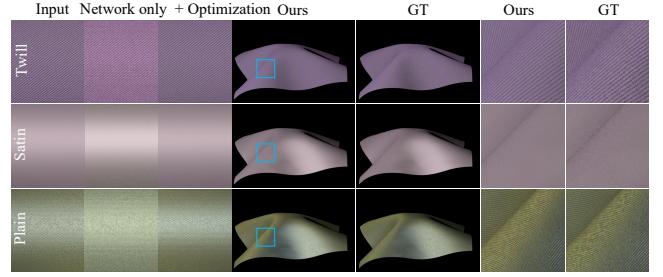
We perform a differentiable rendering of our configuration from Fig. 6 (right). To measure the difference between the rendered image and the input image, we consider three components: a VGG-19 Gram matrix loss  $L_g$ , a prior loss  $L_p$  on the scaling factor  $\beta$ , and a pixel loss  $L_c$  between down-sampled images with resolution  $8 \times 16$ . Our final loss is defined as

$$L_{\text{opt}} = L_g + w_1 L_p + w_2 L_c, \quad (2)$$

$$L_g = \mathcal{L}_1(\text{Gram}(I), \text{Gram}(R)), \quad (3)$$

$$L_p = -\log \left( \exp \left( -\frac{(\beta - \mu_\beta)^2}{2\sigma_\beta^2} \right) \right), \quad (4)$$

$$L_c = \mathcal{L}_1(I_{\text{down}}, R_{\text{down}}). \quad (5)$$



**Figure 9:** Given synthetic input images, our neural network estimation can predict parameters that approach the appearance of the input. Using the optimization further improves the accuracy. Our results on the draped cloth mesh match the ground truth closely.

where  $w_1$  and  $w_2$  are set as 0.0005 and 0.2 respectively,  $\mu_\beta$  and  $\sigma_\beta$  are the mean and the variance of the Gaussian prior on the scaling factor  $\beta$ , respectively. They are set as (1.0, 0.5) for the twill, (0.1, 0.5) for the satin and (1.0, 1.0) for the plain. Note that  $L_p$  simplifies to a quadratic term. We find that  $L_p$  improves optimization robustness and  $L_c$  improves color and highlight shape reproduction. This joint loss will drive the back-propagation to optimize the predicted parameters. The optimization process is shown in Fig. 8.

*Discrete parameters.* During optimization, we treat four parameters as discrete: the yarn density of the weft and warp, the noise level, and the twist angle. This is because we do not currently implement their gradients. With some effort, these gradients could be added, but our discrete solution gives good results. We update these four variables in the following way every five iterations. For the noise level, we randomly perturb plus one or minus one, and accept the perturbation if it gives lower loss than before. Yarn densities are perturbed as follows: +/- 10 yarns per inch before 100 iterations; +/- 5 yarns per inch from 100 to 150 iterations, and +/- 2 yarns per inch from 150 to 200 iterations. For the twist angle, it is perturbed as follows: +/- 5 degrees before 100 iterations; +/- 2.5 degrees from 100 to 150 iterations, and +/- 1 degree from 150 to 200 iterations.

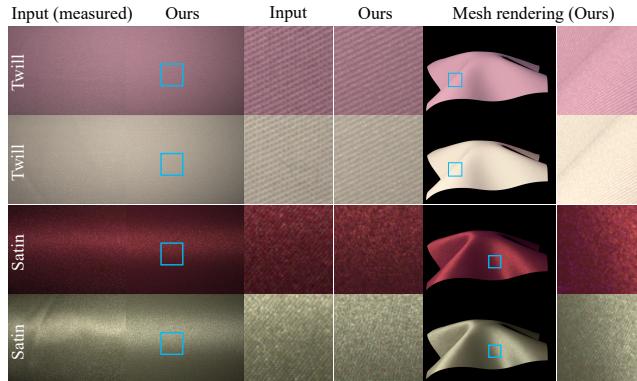
*Details.* We use the Adam optimizer with learning rate 0.01 for 200 iterations. This costs three minutes on an NVIDIA 2080Ti GPU.

## 5 RESULTS

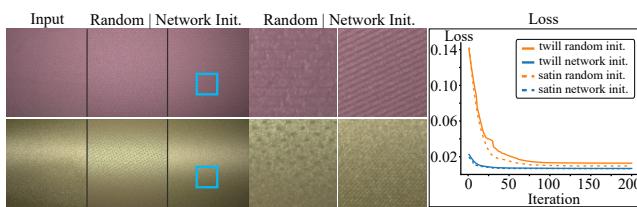
We first show the results of our procedural parameter estimation model on both synthetic data and real data. Next, we show the impact of some critical components (neural network, optimization, and priors) in our inverse model. Lastly, we compare our inverse model with a general procedural method [Shi et al. 2020] and a stationary SVBRDF recovery method [Henzler et al. 2021]. The capture of their inputs is shown in the supplementary material. More discussions and limitations are shown in the supplementary.

### 5.1 Results of our inverse model

*Synthetic data.* In Fig. 9, we show the results of our inverse model on synthetic data, on several different kinds of fabrics. Using synthetic data allows us to establish a precise ground truth, and evaluate the ability of our method to match it. Given the input images rendered



**Figure 10:** 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 11:** Comparison between a random initialization and our network outputs as an initialization.

with our forward shading model, our FabricNet estimates the fabric parameters, which can recover the overall appearance, but with some inaccuracy such as color and roughness bias. Performing an optimization step improves the recovery accuracy, to the point where the re-rendered images and the input images show a near-perfect match. We also show the rendered results on the draped cloth mesh and compare them with the ground truth. Our model can handle fabrics with different warp and weft colors; see more results in Fig. 3 of the supplementary material.

*Real data.* In Fig. 10, we perform the parameter estimation on measured data. 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, can produce closely matching results to the input images. The renderings with the draped cloth mesh also show a plausible appearance.

More renderings using the estimated parameters are shown in Fig. 1. In this scene, we further edit the parameters of our model, using spatially varying diffuse color maps, heightfield scaling factor maps and roughness maps to enrich the final appearance.

## 5.2 Ablation study

*Impact of the FabricNet.* We compare the results with and without using FabricNet for initialization in Fig. 11. The parameters optimized with FabricNet match the inputs much better than optimization with random initialization, visually and quantitatively. After both

loss curves converge, random initialization remains at a larger error than initialization with our FabricNet.

*Impact of the components in optimization.* Optimization has a significant impact on the predicted results, as shown in Fig. 12. We further validate the influence of each component in the optimization, including the Gram matrix, the Gaussian falloff mask, and the pixel loss. Starting from the network predicted parameters, the rendered difference from the input image is noticeable. Including the Gram matrix reduces the color bias for both patterns. The prior loss makes the parameters optimize towards the correct range. The pixel loss enables the highlights at the center for the satin pattern and reduces the color bias for the twill pattern. Finally, the Gaussian falloff mask further helps precisely matching the measurement.

## 5.3 Comparison with other works

*Comparison with other procedural models.* MATch [Shi et al. 2020] is designed for general procedural materials in Substance format. We choose two Substance procedural graphs designed for fabrics (fabric suit and smooth silk) and optimize their parameters to match our samples. To use their implementation, we capture the fabric data on a plane with a single cell phone camera with flash. By comparison, our model has a better recovery of the fabric patterns. While MATch succeeds at matching color and preserves a fabric-like appearance, our results are much closer to the physical samples. Under new view/lighting on the draped cloth mesh, our model produces a more plausible appearance, especially in highlights.

*Comparison with a general SVBRDF model.* In Fig. 13, we compare our method with a recent per-pixel stationary SVBRDF recovery model [Henzler et al. 2021]. As expected, their model loses the fabric appearance under a new view and lighting, as it is not specific to fabrics and simply produces pleasing stationary variation in all parameter maps. On the contrary, our model retains a fabric appearance and avoids artifacts, also in the rendered results of the draped cloth.

## 6 CONCLUSION

In this paper, we have presented a forward fabric shading model and an inverse procedural framework for estimating the model parameters, including a neural network and an optimization based on differentiable rendering. Our forward model can generate high-quality renderings at the yarn level, while remaining simple to implement, supporting efficient differentiable rendering and synthetic data generation. Our inverse framework can estimate fabric parameters that match the ground truth for the synthetic data, and remain close to the input images for measured data.

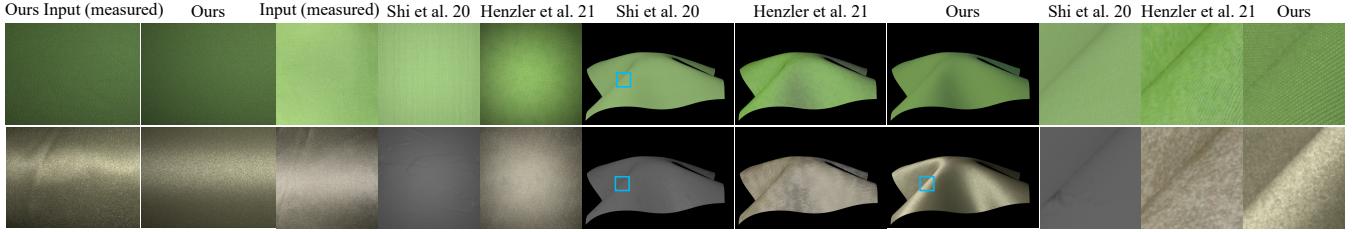
We believe that our inverse framework will act as a foundation for future practical and accurate fabric capture methods. In the future, we plan to extend the capabilities of our forward and inverse models to more types of yarns, wider range of patterns, and more advanced effects such as transmission and knitted fabrics.

## ACKNOWLEDGMENTS

We thank the reviewers for their valuable suggestions. This work has been partially supported by the National Natural Science Foundation of China under grant No. 62172220. Ling-Qi Yan is supported by gift funds from Adobe, Dimension 5 and XVerse.

|       | Input (measured) | Network only | + Gram loss | + Pixel loss | + Gaussian falloff | + Prior loss |
|-------|------------------|--------------|-------------|--------------|--------------------|--------------|
| Twill |                  | SSIM: 0.3699 |             | SSIM: 0.6381 |                    | SSIM: 0.6562 |
| Satin |                  | SSIM: 0.3555 |             | SSIM: 0.4595 |                    | SSIM: 0.5299 |

**Figure 12: The impact of our pipeline components (the Gram matrix loss, the prior loss, the pixel loss, and the Gaussian falloff mask). The structure similarity index measure (SSIM) shows the difference between the images and the ground truth.**



**Figure 13: Comparison of our method to Henzler et al. [2021] and Shi et al. [2020].**

## REFERENCES

- Neeharika Adabala, Nadia Magnenat-Thalmann, and Guangzheng Fei. 2003. Real-Time Rendering of Woven Clothes. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST)*. ACM, New York, NY, USA, 41–47.
- Miika Aittala, Timo Aila, and Jaakko Lehtinen. 2016. Reflectance Modeling by Neural Texture Synthesis. *ACM Trans. Graph.* 35, 4 (2016), 1–13.
- Brent Burley. 2012. Practical Physically-based Shading in Film and Game Production - Physically Based Shading at Disney. <http://blog.selfshadow.com/publications/s2012-shading-course/>.
- Carlos Castillo, Jorge Lopez-Moreno, and Carlos Aliaga. 2019. Recent Advances in Fabric Appearance Reproduction. *Computers & Graphics* 84 (2019), 103–121.
- Valentin Deschaintre, Miika Aittala, Frédéric Durand, George Drettakis, and Adrien Bousseau. 2018. Single-Image SVBRDF Capture with a Rendering-aware Deep Network. *ACM Trans. Graph.* 37, 4 (2018), 1–15.
- Valentin Deschaintre, G. Drettakis, and A. Bousseau. 2020. Guided Fine-Tuning for Large-Scale Material Transfer. *Computer Graphics Forum* 39, 4 (2020), 91–105.
- Giuseppe Guarnera, Peter Hall, Alain Chesaïnais, and Mashhudha Glencross. 2017. Woven Fabric Model Creation from a Single Image. *ACM Trans. Graph.* 36, 5 (2017), 1–13.
- Íbón Guillén, Julio Marco, Diego Gutierrez, Wenzel Jakob, and Adrian Jarabo. 2020. A General Framework for Pearlescent Materials. *ACM Trans. Graph.* 39, 6 (2020), 1–15.
- Jie Guo, Shuichang Lai, Chengzhi Tao, Yuelong Cai, Lei Wang, Yanwen Guo, and Ling-Qi Yan. 2021. Highlight-aware Two-stream Network for Single-image SVBRDF Acquisition. *ACM Trans. Graph.* 40, 4 (2021), 1–14.
- Yu Guo, Miloš Hašan, Lingqi Yan, and Shuang Zhao. 2020. A Bayesian Inference Framework for Procedural Material Parameter Estimation. *Computer Graphics Forum* 39, 7 (2020), 255–266.
- Eric Heitz, Jonathan Dupuy, Cyril Crassin, and Carsten Dachsbacher. 2015. The SGGX Microflake Distribution. *ACM Trans. Graph.* 34, 4 (2015), 1–11.
- 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.
- Yiwei Hu, Julie Dorsey, and Holly Rushmeier. 2019. A Novel Framework for Inverse Procedural Texture Modeling. *ACM Trans. Graph.* 38, 6 (2019), 1–14.
- Piti Irawan and Steve Marschner. 2012. Specular Reflection from Woven Cloth. *ACM Trans. Graph.* 31, 1 (2012), 1–20.
- Wenzel Jakob, Adam Arbree, Jonathan T. Moon, Kavita Bala, and Steve Marschner. 2010. A Radiative Transfer Framework for Rendering Materials with Anisotropic Structure. *ACM Trans. Graph.* 29, 4 (2010), 1–13.
- Alexandre Kaspar, Tae-Hyun Oh, Liane Makatura, Petr Kellnhofer, and Wojciech Matusik. 2019. Neural Inverse Knitting: From Images to Manufacturing Instructions. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, Vol. 97. PMLR, 3272–3281.
- Jan Kautz, Solomon Boulos, and Frédéric Durand. 2007. Interactive Editing and Modeling of Bidirectional Texture Functions. *ACM Trans. Graph.* 26, 3 (2007), 53–es.
- Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. 2017. Modeling Surface Appearance from a Single Photograph using Self-augmented Convolutional Neural Networks. *ACM Trans. Graph.* 36, 4 (2017), 1–11.
- Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. 2018. Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, Cham, 74–90.
- Guillaume Loubet and Frédéric Neyret. 2018. A New Microflake Model with Microscopic Self-shadowing for Accurate Volume Downsampling. *Computer Graphics Forum* 37, 2 (2018), 111–121.
- Zahra Montazeri, Søren B. Gammelmark, Shuang Zhao, and Henrik Wann Jensen. 2020. A Practical Ply-Based Appearance Model of Woven Fabrics. *ACM Trans. Graph.* 39, 6 (2020), 1–13.
- Carlos Rodríguez Pardo, Sergio Suja, David Pascual, Jorge Lopez-Moreno, and Elena Garces. 2019. Automatic Extraction and Synthesis of Regular Repeatable Patterns. *Computers & Graphics* 83 (2019), 33–41.
- Iman Sadeghi, Oleg Bisker, Joachim De Deken, and Henrik Wann Jensen. 2013. A Practical Microcylinder Appearance Model for Cloth Rendering. *ACM Trans. Graph.* 32, 2 (2013), 1–12.
- Kai Schröder, Arno Zinke, and Reinhard Klein. 2015. Image-Based Reverse Engineering and Visual Prototyping of Woven Cloth. *IEEE Transactions on Visualization and Computer Graphics* 21, 2 (2015), 188–200.
- Liang Shi, Beichen Li, Miloš Hašan, Kalyan Sunkavalli, Tammy Boubekeur, Radomir Mech, and Wojciech Matusik. 2020. MATCh: Differentiable Material Graphs for Procedural Material Capture. *ACM Trans. Graph.* 39, 6 (2020), 1–15.
- Elena Trunz, Sebastian Merzbach, Jonathon Klein, Thomas Schulze, Michael Weinmann, and Reinhard Klein. 2019. Inverse Procedural Modeling of Knitwear. In *Proceedings of CVPR*, 8622–8631.
- Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction through Rough Surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*. Goslar, DEU, 195–206.
- Beibei Wang, Wenhua Jin, Miloš Hašan, and Ling-Qi Yan. 2022. SpongeCake: A Layered Microflake Surface Appearance Model. *ACM Trans. Graph.* (2022), 1–15.
- Hong-yu Wu, Xiao-wu Chen, Chen-xu Zhang, Bin Zhou, and Qin-ping Zhao. 2019. Modeling Yarn-level Geometry from a Single Micro-image. *Frontiers of Information Technology & Electronic Engineering* 20 (2019), 1165–1174.
- Wenjie Ye, Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. 2018. Single Image Surface Appearance Modeling with Self-augmented CNNs and Inexact Supervision. *Computer Graphics Forum* 37, 7 (2018), 201–211.
- Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. 2011. Building Volumetric Appearance Models of Fabric Using Micro CT Imaging. *ACM Trans. Graph.* 30, 4 (2011), 98–105.
- Yezi Zhao, Beibei Wang, Yanning Xu, Zheng Zeng, Lu Wang, and Nicolas Holzschuch. 2020. Joint SVBRDF Recovery and Synthesis From a Single Image using an Unsupervised Generative Adversarial Network. In *EGSR*. DL, 53–66.