Woven Fabric Capture with a Reflection-Transmission Photo Pair

Yingjie Tang
Nankai University
Tianjin, China
lxtyin.ac@gmail.com

Zixuan Li
Nankai University
Tianjin, China
zixuan.li_2001@outlook.com

Miloš Hašan
Adobe Research
San Jose, USA
milos.hasan@gmail.com

Jian Yang
Nanjing University of Science and
Technology
Nanjing, China
csjyang@njust.edu.cn

Beibei Wang*
School of Intelligence Science and
Technology, Nanjing University
Suzhou, China
beibei.wang@nju.edu.cn

| Inputs | Rendering (plane) | Rendering (lamp) | Inputs | Rendering (plane) | Rendering(cloth) |
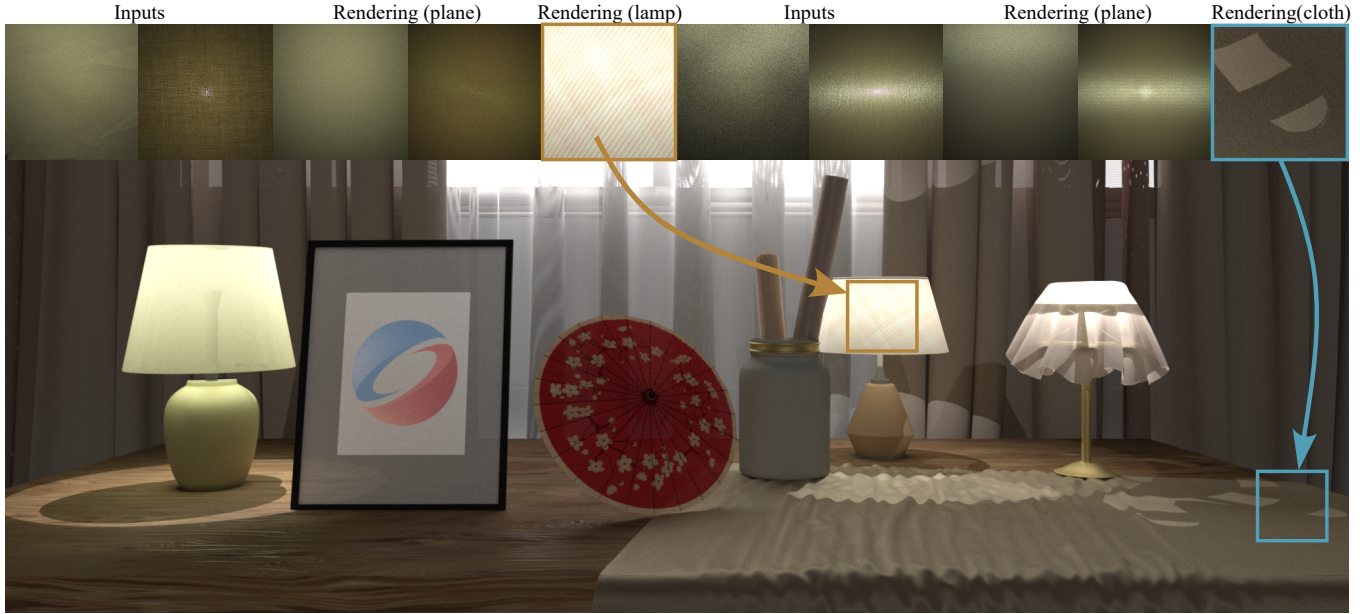
Figure 1: Given two photos of a woven fabric sample (front-lit and back-lit), our approach estimates the parameters of our proposed woven fabric material model. Re-rendered results with estimated parameters closely match the input photos on the top. The resulting fabric parameters can be used in a rendered scene, either directly or after further editing, by using spatially-varying diffuse color maps.

ABSTRACT

Digitizing woven fabrics would be valuable for many applications, from digital humans to interior design. Previous work introduces a lightweight woven fabric acquisition approach by capturing a single reflection image and estimating the fabric parameters with a differentiable geometric and shading model. The renderings of the estimated fabric parameters can closely match the photo; however, the captured reflection image is insufficient to fully characterize the fabric sample reflectance. For instance, fabrics with different thicknesses might have similar reflection images but lead to significantly different transmission. We propose to recover the woven fabric parameters from *two* captured images: reflection and transmission. At the core of our method is a differentiable bidirectional scattering distribution function (BSDF) model, handling reflection and transmission, including single and multiple scattering. We propose a two-layer model, where the single scattering uses an SGGX phase function as in previous work, and multiple scattering uses a new azimuthally-invariant microflake definition, which we term ASGGX. This new fabric BSDF model closely matches real woven fabrics in both reflection and transmission. We use a simple setup

---

*Corresponding author.

for capturing reflection and transmission photos with a cell phone camera and two point lights, and estimate the fabric parameters via a lightweight network, together with a differentiable optimization. We also model the out-of-focus effects explicitly with a simple solution to match the thin-lens camera better. As a result, the renderings of the estimated parameters can agree with the input images on both reflection and transmission for the first time. The code for this paper is at https://github.com/lxtyin/FabricBTDF-Recovery.

CCS CONCEPTS

• **Computing methodologies** → **Rendering**; **Reflectance modeling**.

KEYWORDS

fabric capture, microflake, BTDF

1 INTRODUCTION

Rendering fabrics is valuable for many applications, such as interior visualization, fabric design, virtual reality, digital humans, etc. However, creating high-quality digital fabric assets requires extensive work, even for experienced artists. The alternative is to capture the fabric materials from the real world. Historically, capture required specialized devices and complex pipelines; recently, lightweight fabric capture has become an active research topic. In this paper, we focus on lightweight capture of woven fabrics: a common and important subset.

Recent work by Jin et al. [2022] introduced a lightweight woven fabric capture approach by taking a single photo in a simple setup. At the core of their method is a differentiable procedural model and a bidirectional reflectance distribution function (BRDF) specialized for woven fabrics, used for parameter recovery through inverse rendering. The recovered parameters can match the captured photos very well, in terms of both highlights and structure. However, their method only models the reflection and recovers the parameters under the supervision of only the reflection image. Fabrics with different thicknesses might have similar reflection images, but very different backlighting (light transmission) behavior. Transmission is of critical importance if the fabric is used as a window or lamp shade, but can also be important when used as clothing. This effect can no longer be estimated from a single image; both the reflection and transmission images are needed to recover the complete set of fabric parameters.

In this paper, we propose a simple configuration to capture two images of a flat fabric sample from the same camera but with two light positions, front and back, for lightweight parameter recovery. Our method recovers the parameters of woven fabrics by a combination of a small neural network and differentiable optimization, following a reconstruction pipeline similar to the work of Jin et al. [2022], matching the captured reflection and transmission images. Fabrics with the recovered parameters can be used in a final rendering engine.

A key component of our method is a new bidirectional scattering distribution function (BSDF) model for woven fabrics, modeling both single and multiple scattering for both reflection and transmission. Previous works models the transmission of fabrics using the SpongeCake model [Wang et al. 2022; Zhu et al. 2023a], which consists of volumetric layers with fiber-like microflake phase functions. They represent multiple scattering with a single scattering lobe but with modified parameters. Unfortunately, these models in their current form cannot fully explain the behavior of multiple scattering from woven fabrics in the transmission image, as shown in Fig. 4. To this end, we extend SpongeCake to a new two-layer BSDF model for woven fabrics, whose novel component uses a specialized empirical phase function for aggregated microflakes to model the multiple scattering from fibers. Furthermore, we also model gaps between yarns and the out-of-focus appearance of the lightsource to better match the transmission photographs. As a consequence, our recovered parameters can faithfully match the captured images on both synthetic and real data for several typical woven fabric patterns. To summarize, our main contributions include:

- a new phase function for aggregated microflakes to better characterize the appearance of multiple scattering from fibers,
- a new two-layer BSDF model with several key components to match real woven fabric reflection and transmission,
- a lightweight capture configuration that only requires two photos of a fabric sample under front and back point illumination.

2 RELATED WORK

Fabric models. We briefly review woven fabric models, consisting of geometry and appearance models. The geometry of woven fabrics can be represented in different ways, including volume, curve, and surface-based models. The volume representation usually couples with the microflake model [Heitz et al. 2015; Jakob et al. 2010] to define fiber-like participating media. Curve-based fabric models usually use bidirectional curve scattering distribution functions (BCSDFs) [Chiang et al. 2015; Marschner et al. 2003; Montazeri et al. 2020; Zhu et al. 2023b] to define the optical properties of each fiber or ply. Surface models rely on macroscopic fabric geometry, together with a BSDF as the surface shading model [Irawan and Marschner 2012; Jin et al. 2022; Sadeghi et al. 2013; Zhu et al. 2023a].

Among these three groups, the volume and curve-based models can bring high realism at extreme close-up views at the cost of memory and computation. In contrast, surface-based models are lightweight and can achieve high-fidelity results at the macroscopic scale. We focus on the latter type of model in our paper.

Surface fabric models. Surface fabric models act as BRDFs (e.g., [Adabala et al. 2003] [Irawan and Marschner 2012] [Sadeghi et al. 2013]). They model the fabric structures with normals and tangents, which are used in the reflectance model. Recently, Jin et al. [2022] propose a lightweight procedural geometric and reflectance model based on the SpongeCake model [Wang et al. 2022]. Their method is simple and differentiable, allowing for inverse rendering. Zhu et

Table 1: Parameters in our BSDF model. The top three parameters affect the yarn geometry, and the rest affect reflectance. The * indicates that this variable allows different values for warp and weft yarns.

s	yarn size
β *	heightfield scaling factor
ξ *	gap scaling
k_s^s, k_s^m *	specular albedo for single / multiple
k_d^r, k_d^t	diffuse albedo for reflection / transmission
α^s, α^m *	roughness for single / multiple
ψ *	fiber twist angle
u_{max}	maximum inclination angle
T^s, T^m *	thickness of the fabric for single / multiple
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

al. [2023a] use the SpongeCake forward model to enable shadowing-masking from the yarns at the cost of a more complex formulation, and do not consider capture.

Single-image fabric recovery. Several methods have been proposed to recover fabrics with a single image as input at different levels. Schröder et al. [2015] and Wu et al. [2019] can achieve fiber-level detail, but rely on manual selection of model parameters or require expensive time cost. Guarnera et al. [2017] estimate the yarn parameters in the spatial and frequency domain at the yarn level at the cost of a complex pipeline. More related work can be found in the survey by Castillo et al. [2019]. Unlike the above work, our method targets a simple setup and lightweight pipeline. Rodríguez Pardo et al. [2019] aims at recovering the macroscopic color pattern textures for woven fabrics rather than the fabric parameters, which is out of our scope.

A closely related work to ours is by Jin et al. [2022], which estimates woven fabric parameters from a single photo with differentiable rendering. However, it only considers the reflection image, which we will show to be insufficient; our method utilizes both the reflection and transmission images for fabric parameter recovery.

Procedural material parameter estimation. Besides the above works specialized for fabrics, some approaches have been proposed for predicting general procedural material parameters by learning the mapping from the input image to the parameters with a neural network [Garces et al. 2023; Hu et al. 2019; Rodriguez-Pardo et al. 2023; Shi et al. 2020] or Bayesian framework [Guo et al. 2020]. These methods are designed for general materials and are not optimal for fabrics, but our method shares some common components with these approaches, such as neural parameter initialization and differentiable optimization with perceptual losses.

3 BACKGROUND AND MOTIVATION

Woven fabrics are manufactured by interweaving warp and weft yarns. Here we define the the warp as vertical and the weft as horizontal, as shown in Fig. 7. We focus on several typical weave patterns, though the system is easily extensible to other patterns.

Background on reflection capture. Jin et al. [2022] propose a geometric and appearance model for woven fabrics as a spatially-varying BRDF defined on a macroscopic fabric surface, instead of using volumes [Zhao et al. 2011] or curves [Zhu et al. 2023b]. They model yarn geometry as smooth bent cylinders, which do not have to be explicitly constructed, and instead yield normal vectors, tangents and other information used in their reflectance model. The reflectance model includes a specular and a diffuse term; the former is based on the SpongeCake model [Wang et al. 2022] with a fiber-like microflake phase function, and the latter is a double diffuse term considering both the macro-surface normal and the yarn normal. Their model can represent accurate woven fabric reflection, both in terms of highlight shapes and spatial texture. They recover the fabric parameters from a single captured reflection image with a neural parameter prediction network followed by optimization via differentiable rendering.

Motivation for adding transmission capture. Recovering the fabric parameters solely from reflection cannot provide enough information to reproduce all fabric parameters, most obviously the thickness, which is not sensitive to reflection but strongly affects transmission. Most other parameters also affect transmission and can be estimated more accurately by matching the transmission image as well. Our solution is to use reflection and transmission together for fabric parameter recovery.

## 4 FABRIC BIDIRECTIONAL SCATTERING DISTRIBUTION FUNCTION

Jin et al. [2022] provide an appearance model for woven fabric reflection. Their model is based on SpongeCake [Wang et al. 2022], which supports transmission automatically. The straightforward way is to simply enable transmission in Jin et al.'s model. Unfortunately, we find that using the transmission as-is cannot accurately match multiple scattering in the transmission image, as shown in Fig. 4. The problem turns out to be that the approximation of reusing the single-scattering lobe based on SGGX microflakes for multiple scattering, while generally reasonable, produces the largest error where the incoming and outgoing directions are pointing opposite each other, which happens to be in the center of the back-lit transmission image. Therefore, a key problem is to design a better multiple-scattering lobe, which can match real fabric transmission photographs closely. We propose a new BSDF (Sec. 4.2) to model fabric reflection and transmission, whose main new component is a better representation for the multiple scattering of microflake media (Sec. 4.1).

4.1 An azimuthally-invariant phase function for multiple scattering

As shown in Fig. 2, the multiple scattering distribution quickly becomes azimuthally uniform due to the diffusion of the multiple bounces among the microflakes. This behavior cannot be achieved by a single-scattering SpongeCake lobe using standard SGGX microflakes, no matter how the parameters are set. Hence, we design a specialized azimuthally-invariant microflake phase function, which we term ASGGX, to represent the multiple scattering distribution.
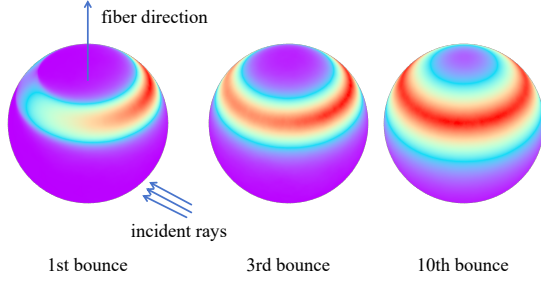
Figure 2: Given an incoming ray, we simulate the scattering among the fibers with different bounces using the SGGX phase function and visualize the distribution of the outgoing ray. The outgoing distribution becomes uniform along the azimuth angle. Representing such an azimuthally invariant distribution is beyond the capability of the SGGX phase function.
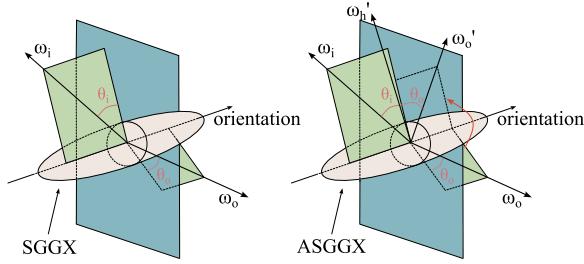

Figure 3: Configurations of the SGGX (left) and our ASGGX (right). In our ASGGX, ω_o is rotated to the same longitudinal plane as ω_i, leading to a new vector ω_o'. Then, ω_i and ω_o' form a half-vector ω_h', which is used to look up the microflake density.
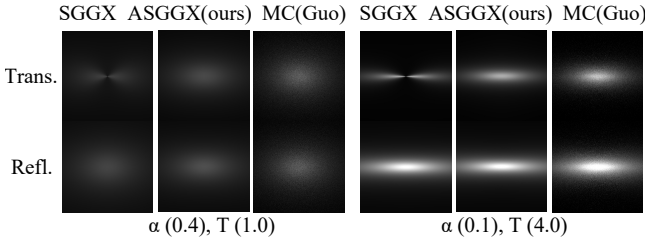
Figure 4: Multiple scattering comparison among SGGX, ASGGX (ours), and Monte-Carlo simulation [Guo et al. 2018], which serves as a synthetic reference on two different sets of parameters (roughness α and thickness T). Our method can closely match the reference on both reflection and transmission, while the results by SGGX match the GT on reflection only.

This new phase function can be plugged into the SpongeCake formulation to derive the corresponding BSDF.

Intuitively, we can think of an aggregation of microflakes as a single scattering event. Note that the overall fiber direction is given (as in standard SGGX), and we will assume a coordinate frame aligned
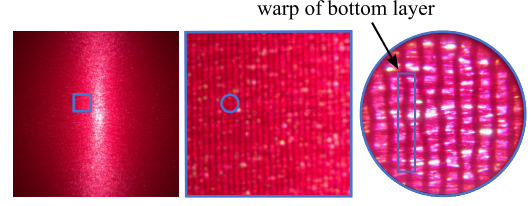
Figure 5: Macro appearance and its zoom-in. Yarns of the bottom layer are also visible, which have a significant effect on the macro appearance.

with this direction. We propose an azimuthally-invariant phase function as follows: given an incoming direction ω_i and outgoing direction ω_o, we rotate them into the same arbitrary longitudinal plane, so that their azimuth angles become identical. We define the new phase function using the half-vector ω_h' computed from the *modified* directions. For this definition to be valid, the longitudinal plane chosen above should not matter for the final result, which will be true in our case, because the microflake density is rotationally invariant around the fiber direction. For convenience we choose the plane that contains ω_i for Fig. 3 and the discussion below.

More precisely, our azimuthally-invariant phase function $f_p(\omega_i \rightarrow \omega_o)$ has very similar formulation to SGGX, and reuses its density $D(\omega)$, but the half vector is computed differently:

$$f_p(\omega_i \rightarrow \omega_o) = \frac{D(\omega_h')}{2\sigma(\omega_i)}. \tag{1}$$

Here, ω_h' is the half-vector between the modified (rotated) ω_o' and ω_i, as shown in Fig. 3. The 2 in the denominator comes from the transformation $d\omega_o' = 2|\omega_o' \cdot \omega_h'|d\omega_h'$, where ω_o' and ω_h' are treated as 2D unit vectors restricted to the longitudinal plane. This is different from the standard half-vector reflection Jacobian, $d\omega_o = 4|\omega_o \cdot \omega_h|d\omega_h$. The functions D and σ are the same as for SGGX [Heitz et al. 2015], and are rotationally invariant around the fiber direction. The proposed azimuthally-invariant phase function satisfies both energy conservation and reciprocity.

After establishing the phase function, we use the single scattering of this *aggregated microflake* to represent the multiple scattering of the original microflakes. For that, we need to modify the relative thickness of the aggregated flake. The final multiple scattering is computed by considering the attenuation:

$$f_m(\omega_i, \omega_o) = \frac{k_s^m D_m(\omega_{h'})G_m(\omega_i, \omega_o)}{2cos(\omega_i)cos(\omega_o)} \tag{2}$$

where D_m and G_m is specific for multiple scattering, detailed in the supplementary. They are the same form as SpongeCake but using modified thickness and roughness.

In Fig. 4, we compare the rendered results of our model, the SpongeCake model and the reference which is rendered with Guo et al. [2018]. Here, the parameters are found by optimization.

4.2 Two-layer microflake model

Now we can model single scattering by a SpongeCake layer using the original SGGX microflake and the multiple scattering term by a layer using the new ASGGX microflake, which we observe to match the fabric photos better. However, we have several additional
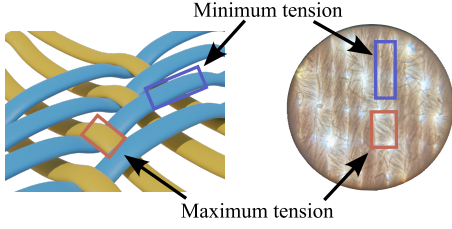
Figure 6: The yarn becomes thinner at the intersection of two yarns because of the increased tension.
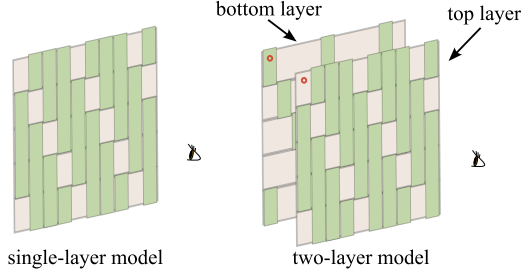
Figure 7: Instead of using a single-layer model (left), we introduce a two-layer model (right), where both the weft and warp yarns are considered at each location (red dot).

observations (Fig. 5) from the captured transmission image, which will inform our final BSDF model.

First, in a transmission image, both the weft and warp yarns have an effect at a single pixel, which cannot be represented by the single-layer model that was sufficient in the work of Jin et al. [2022] and Irawan and Marschner [2012]. Second, yarn thickness varies due to different tension at different points within a weave pattern. As the thickness decides the fraction of the light passing through, it significantly impacts the transmission, while it could be ignored in previous work considering only reflection. These observations are not modeled in previous work [Jin et al. 2022; Zhu et al. 2023a]. Therefore, we introduce a two-layer yarn model, and propose a tension-aware thickness modulation function.

Two-layer model. We use two SpongeCake layers to represent, at each point on the fabric, the yarn on the top (closer to camera) and at the bottom (away from camera) respectively, as shown in Fig. 7. In this model, each layer represents a yarn and its parameters depend on whether it is a weft or warp. Each of them has their own yarn parameter set, including diffuse albedo and roughness. All the parameters are summarized in Table. 1.

Tension-aware thickness function. To characterize the appearance from the varying thickness, we propose a thickness modulation function for each yarn, depending on the location on the yarn. The thickness is defined by considering the scaling factor:

$$T = T \times (S_{\min} + \mu \times (1 - S_{\min})), \tag{3}$$

where $S_{\min}$ is the minimum thickness scale, set as 0.5 for satin and twill, and 1.0 for plain in practice. μ is the tension level, defined as 0.0 at the center of a yarn's long part, 1.0 at the center of the yarn's compressed part, and linearly interpolated between the two.

4.3 Final BSDF model

We now define our full bidirectional scattering distribution function (BSDF), which includes three terms: single scattering term for two layers using the SGGX microflake phase function, low-order multiple scattering term for two layers using the new ASGGX phase function, and an additional modified diffuse term for high order scattering. As for the diffuse term, we use a blended formulation similar to Jin et al. [2022], considering both the macro surface normal and the micro-yarn normals on both sides. Each term is defined for both reflection and transmission.

For a given surface location on the fabric, let ω_i and ω_o be the incident (light) and outgoing (camera) directions in world space. Let ω_m be the smooth macroscopic surface normal (e.g. interpolated from vertex normals). Our fabric shading model has three terms:

$$f(\omega_i, \omega_o) = f_s(\omega_i, \omega_o) + f_m(\omega_i, \omega_o) + f_d^{r,t}(\omega_i, \omega_o). \tag{4}$$

Here $f_s(\omega_i, \omega_o)$ represents single scattering and is a two-layer SpongeCake lobe using SGGX, $f_m(\omega_i, \omega_o)$ represents lower-order multiple scattering and is a two-layer SpongeCake lobe using ASGGX, and the diffuse term represents higher-order scattering:

$$f_d^r(\omega_i, \omega_o) = w \frac{k_d^r \langle \omega_i \cdot \omega_{n1} \rangle}{\pi \langle \omega_i \cdot \omega_m \rangle} + (1 - w) \frac{k_d^r}{\pi}, \tag{5}$$

$$f_d^t(\omega_i, \omega_o) = w \frac{k_d^t \langle \omega_i \cdot \omega_{n1} \rangle \cdot \langle \omega_i \cdot \omega_{n2} \rangle}{\pi \langle \omega_i \cdot \omega_m \rangle} + (1 - w) \frac{k_d^t}{\pi}, \tag{6}$$

where ω_{n1}, ω_{n2} are the yarn normal of first layer and second layer respectively, and k_d^r, k_d^r denote the diffuse albedo for the reflection and transmission, respectively. For reflection, we only consider the normal at the first layer, while for transmission, we use a product of cosine terms, since we empirically observe that both normals affect the diffuse component.

Our single scattering term $f_r^s(\omega_i, \omega_o)$ is the same as the single scattering of a two-layer SpongeCake model with the SGGX phase function [Wang et al. 2022], including both reflection and transmission, where the thickness is scaled by the thickness scaling function from Eqn. (3). The detailed formulation is shown in the supplementary. Finally, our multiple scattering term is also formulated as a two-layer SpongeCake model with the ASGGX phase function and its own parameters, as shown in Eqn. (2).

To summarize, our woven fabric shading model consists of the following parameters (see Table 1): a discrete weave pattern, single/multiple scattering albedo for weft/warp, diffuse albedo for reflection and transmission, roughness/thickness for weft/warp and single/multiple scattering respectively, yarn size for weft/warp, a height field scaling factor for weft/warp, a gap scaling factor for weft/warp (shown in Fig. 1 in the supplementary), a twist angle for weft/warp, blending weight of the diffuse term, randomness on the specular term, and a noise level performed on the height field scaling factor to control the intensity of the orientation / normal map randomness.
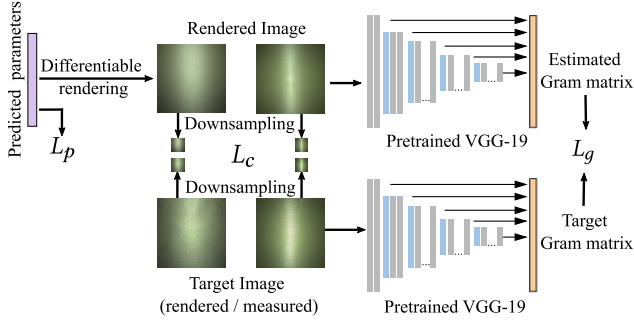
5 FABRIC PARAMETER ESTIMATION

Based on the proposed fabric BSDF, we estimate the woven fabric parameters. We first introduce a simple setup for fabric sample capture (Sec. 5.1), and then introduce the pipeline for the parameter estimation (Sec. 5.2 and 5.3).
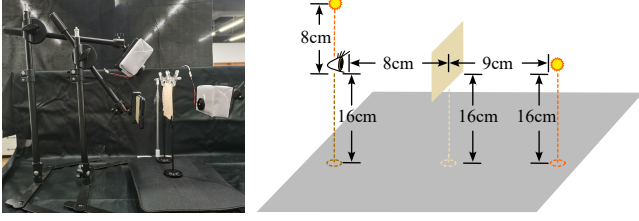
Figure 8: The configuration to measure the real fabric data. We use one cell phone as camera and two small light sources to capture the reflection and transmission image respectively. The fabric sample is fixed by a holder. We also measure the distances in this configuration to reconstruct the same setup in synthetic renderings.
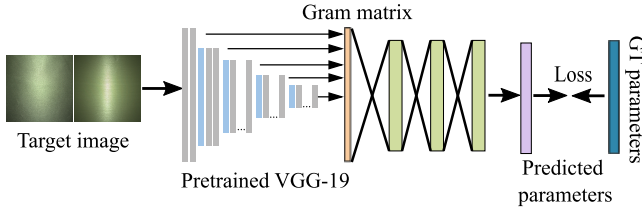
Figure 9: Our network architecture.

Figure 10: Our optimization pipeline.

5.1 Measurement setup for fabrics

We propose to capture a reflection-transmission photo pair for fabric parameter reconstruction, as shown in Fig. 8. We use one cell phone as camera and two point lights for illumination and put the fabric sample in-between two light sources with a holder. The captured raw images have a $4K$ resolution, and we crop and downsample them to a resolution of 512×512. We calibrate the light brightness and apply falloff due to lens vignetting, similar to Jin et al. [2022].

5.2 Neural network for fabric parameter prediction

Architecture. Given the input reflection-transmission pair, we use a simple neural network (Fig. 9) to predict approximated parameter vectors and the pattern type. We feed the two images into

a pretrained VGG-19 network separately, and compute Gram matrices, resulting in two features, each a vector of size 610304. We concatenate the features from the two images and feed them into a fully connected (FC) module, which includes three intermediate layers (256 nodes per layer) with LeakyReLU activation function. The final FC layer outputs the predicted parameters (34 channels for our forward model). Note that this network is similar to the one by Jin et al. [2022], except we use two images, one more layer, and different output channels.

Dataset generation. We generate a rendered dataset of five weave patterns (twill, satin, plain, and 90-degree rotations of the twill and satin), The twist angle is set at -30 degrees for the twill and no twist for other patterns. We generate 1,280 images for each pattern with our shading model (Sec. 4), by sampling the fabric parameter space, detailed in Table *S2* (Table 2 in the supplementary material).

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 with the Adam solver, where the learning rate is set as 0.0001 and the batch size is set as 32. Only FC weights are updated during training (VGG / Gram matrix computation is frozen). Training took four hours on a single A40 GPU.

5.3 Optimization with differentiable rendering

We use the network-predicted fabric parameters as an initialization and perform optimization (Fig. 10) with differentiable rendering, which further improve the match, e.g. in color.

We render the reflection and transmission images in a differentiable PyTorch function. To better match the out-of-focus light appearance in transmission images caused by the camera focusing on the fabric plane, we project the point light to the rendered image, and generate a Gaussian around the projected center. The gaps between the yarns look up this Gaussian, simulating direct view of the light source. This way, our rendered result can match an out-of-focus effect, despite using a pin-hole camera model in the simulation. In practice, we set the Gaussian scaling as 8 and the variance as 20 pixels, but this depends on our specific camera and light, and could be calibrated for more generality.

We compute the difference between the rendered images and the target images to drive the optimization. We use several losses to measure the difference between the target images and the rendered images, including a VGG-19 Gram matrix loss L_g, a prior loss L_p on the scaling factor β and gap scaling factor ξ for optimization robustness, and a pixel loss L_c between down-sampled images with resolution 16×16 to improve color.

Our final loss is defined as

$$L_{\text{opt}} = L_g + w_1 L_p + w_2 L_c, \tag{7}$$

$$L_g = \mathcal{L}_1(\text{Gram}(I), \text{Gram}(R)), \tag{8}$$

$$L_p = -\log\left(\exp\left(-\frac{(\beta - \mu_\beta)^2}{2\sigma_\beta^2}\right)\right) - \log\left(\exp\left(-\frac{(\xi - \mu_\xi)^2}{2\sigma_\xi^2}\right)\right) \tag{9}$$

$$L_c = \mathcal{L}_1(I_{\text{down}}, R_{\text{down}}). \tag{10}$$

where w_1 and w_2 are set as 0.001 and 0.1 respectively, μ_β and σ_β are the mean and the variance of the Gaussian prior on the scaling

factor β, 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. The Gaussian prior on gap scaling factor ξ is set as (0.9, 0.05) for satin and twill, (0.75, 0.1) for plain. We also optimize several discrete parameters, including the yarn density of the weft and warp, the twist angle and gap scaling, detailed in the supplementary. We use the Adam optimizer with learning rate 0.01 for 300 iterations, which takes about four minutes on an NVIDIA 4060 GPU.

Discussion. Note that the tension-aware thickness is not optimized and treated as a fixed per-pattern property, as it is determined by the proportion of the yarn's long and compressed parts.

During the optimization, we perform differentiable rendering with Eqn. (4), without considering the yarn-level shadowing-masking term [Zhu et al. 2023a], as the shadowing-masking effect in the current capture configuration is not obvious, and its formulation is not differentiable. Instead, we apply this term for the final mesh renderings (Figs. 13 and 16) after estimating the parameters.

6 RESULTS

We first show the results of our procedural parameter estimation model on both synthetic data and real data. Afterwards, we perform ablation studies on several key components.

### 6.1 Results of our inverse model

Synthetic data. In Fig. 16, we validate our method on synthetic data using five kinds of fabrics. Given two images as inputs, our network first predicts coarse parameters whose renderings roughly match the inputs but with some color bias and highlight mismatches. Then, the differentiable optimization addresses the above issues, producing a closer match between input images and renderings, which is also confirmed by the rendering of the draped cloth mesh. For further validation, we provide the estimated parameters (roughness and thickness) in Table 3 (supplementary), showing that the difference between the predicted and the ground-truth parameters is acceptable.

Real data. In Fig. 13, we perform parameter estimation on real measured data and compare our method with Jin et al. [2022]. Since no ground truth parameters exist for the measured data, we compare the visual match between the input and rendered images with the estimated parameters. The renderings with the draped cloth mesh also show a plausible appearance for both reflection and transmission. For Jin et al. [2022], we capture their input image in their suggested setup, wrapping the fabric around a cylinder. Their method can reproduce reflection results matching the real photo. In theory, these estimated parameters can be used to render transmission as well. However, we find that their transmission prediction differs significantly from the captured photo due to the missing of some critical parameters (e.g., thickness) and the suboptimal handling of multiple scattering in the original SpongeCake model. More results are shown in Fig. 2 (supplementary).

We further validate our method by capturing the fabric samples from a novel view (by rotating them) and rendering them under the same configuration. The rendering of estimated parameters can match the real fabrics at the novel view, as shown in Fig. 3 (supplementary).
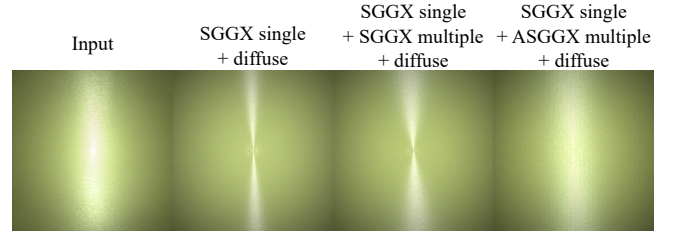
Figure 11: The impact of our ASGGX on fabric recovery, by comparing different shading models on a satin fabric. Our ASGGX outperforms the others and can match the highlights in the input transmission image.
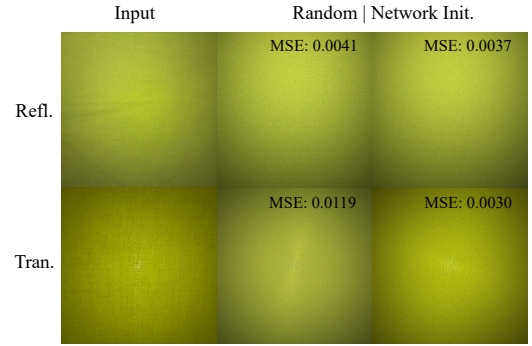
Figure 12: Comparison between a random initialization and our network initialization, where the latter shows a lower error.

We apply the estimated parameters from real data into a complex scene and further edit the parameters with spatially varying diffuse color maps to demonstrate the various appearances in Fig. 1.

6.2 Ablation studies

Impact of the ASGGX phase function. Our ASGGX for multiple scattering is critical for transmission representation and recovery. We validate its impact by comparing fabric recovery with three different transmission shading models: 1) a diffuse term, 2) SGGX + diffuse, and 3) ASGGX + diffuse, where all of them have another SGGX for single scattering. We use the network for prediction in these three results and then perform the optimization with these shading models under the same settings (iterations and learning rate). As shown in Fig. 11, a single diffuse, or together with the SGGX, mismatch the highlights with the input images due to the characteristics of SGGX, while our solution (ASGGX + diffuse) produces a better match with the input.

Impact of the two-layer model. Fabrics have vertical continuous yarns in the real capture, particularly for the satin, as shown in Fig. 5. Our two-layer model captures this appearance. To show its influence, we compare the renderings of estimated parameters optimized by our two-layer and single-layer models on the satin example in Fig. 14. The single layer cannot produce the vertical continuous yarns, as the weft yarns always cut the warp yarns. In contrast, in the two-layer model, the light can pass through the

weft and reach the warp even if the weft yarn is on the top, leading to a continuous vertical yarn.

Impact of the tension-aware thickness. We validate the influence of tension-aware thickness in Fig. 15, by comparing with a constant thickness. By comparison, we find that the overall diagonal structure cannot be captured with a constant thickness due to the overlap between the two layers. This issue is addressed by our tension-aware thickness, which enhances this overall diagonal structure, leading to a better match with the real data.

Impact of the network prediction. Similar to Jin et al. [2022], our method uses a network for initialization. We show the impact of the network in Fig. 12 by comparing the results with and without using the network for initialization. The results with network initialization show higher quality than a random initialization.

Impact of the loss function. We use three loss terms (the pixel loss, the Gram matrix loss, and the prior loss) in the optimization step. We validate their influence in Fig. 4 (supplementary). By comparison, the pixel loss and the Gram matrix loss reduce color bias, while the prior loss improves robustness. More detailed discussions are shown in the supplementary material.

6.3 Discussion and limitations

Missing yarn variations and global features. Our shading model does not consider the variations in yarn diameter, yarn sliding, or global features (e.g., wrinkles and flyaway fibers), leading to some mismatches in the estimated results. Our forward model can be extended to support all these features by introducing more complex procedural spatial variation; However, estimating many more parameters controlling these features may make the optimization more difficult with only two images as inputs.

Unseen weave patterns. Our network is trained on several typical patterns, similar to Jin et al. [2022]. Regarding the other weave patterns, the network needs retraining.

7 CONCLUSION

In this paper, we presented a woven fabric parameter estimation pipeline using a captured reflection-transmission photo pair. The key component of the pipeline is our new fabric BSDF using an azimuth-invariant phase function to fit the multiple scattering of real back-lit fabrics better. Our full BSDF model has several components to match real woven fabric reflection and transmission. Our inverse framework allows a close match of both reflection and transmission to the input images. Our lightweight capture can achieve high-fidelity recovery of woven fabrics at a distant view. However, we could further model yarn-level details and imperfections, and consider extensions to other types of fabrics, like knitted fabrics.

ACKNOWLEDGMENTS

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.

Carlos Castillo, Jorge Lopez-Moreno, and Carlos Aliaga. 2019. Recent Advances in Fabric Appearance Reproduction. *Computers & Graphics* 84 (2019), 103–121.

Matt Jen-Yuan Chiang, Benedikt Bitterli, Chuck Tappan, and Brent Burley. 2015. A Practical and Controllable Hair and Fur Model for Production Path Tracing. In *ACM SIGGRAPH 2015 Talks (SIGGRAPH '15)*. Association for Computing Machinery, New York, NY, USA, Article 23, 1 pages. https://doi.org/10.1145/2775280.2792559

Elena Garces, Victor Arellano, Carlos Rodriguez-Pardo, David Pascual-Hernandez, Sergio Suja, and Jorge Lopez-Moreno. 2023. Towards Material Digitization with a Dual-scale Optical System. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–13.

Giuseppe Guarnera, Peter Hall, Alain Chesnais, and Mashhuda Glencross. 2017. Woven Fabric Model Creation from a Single Image. *ACM Trans. Graph.* 36, 5 (2017), 1–13.

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.

Yu Guo, Miloš Hašan, and Shuang Zhao. 2018. Position-Free Monte Carlo Simulation for Arbitrary Layered BSDFs. *ACM Trans. Graph.* 37, 6, Article 279 (Dec. 2018), 14 pages.

Eric Heitz, Jonathan Dupuy, Cyril Crassin, and Carsten Dachsbacher. 2015. The SGGX Microflake Distribution. *ACM Trans. Graph.* 34, 4 (2015), 1–11.

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.

Wenhua Jin, Beibei Wang, Milos Hasan, Yu Guo, Steve Marschner, and Ling-Qi Yan. 2022. Woven Fabric Capture from a Single Photo. In *SIGGRAPH Asia 2022 Conference Papers (SA '22)*. Association for Computing Machinery, New York, NY, USA, Article 33, 8 pages. https://doi.org/10.1145/3550469.3555380

Stephen R. Marschner, Henrik Wann Jensen, Mike Cammarano, Steve Worley, and Pat Hanrahan. 2003. Light Scattering from Human Hair Fibers. *ACM Trans. Graph.* 22, 3 (jul 2003), 780–791. https://doi.org/10.1145/882262.882345

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 Rodriguez-Pardo, Henar Dominguez-Elvira, David Pascual-Hernandez, and Elena Garces. 2023. Umat: Uncertainty-aware single image high resolution material capture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5764–5774.

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, 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.

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.

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.

Junqiu Zhu, Adrian Jarabo, Carlos Aliaga, Ling-Qi Yan, and Matt Jen-Yuan Chiang. 2023a. A Realistic Surface-Based Cloth Rendering Model. In *ACM SIGGRAPH 2023 Conference Proceedings (SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 5, 9 pages. https://doi.org/10.1145/3588432.3591554

Junqiu Zhu, Zahra Montazeri, Jean-Marie Aubry, Ling-Qi Yan, and Andrea Weidlich. 2023b. A Practical and Hierarchical Yarn-based Shading Model for Cloth. *Computer Graphics Forum* (2023). https://doi.org/10.1111/cgf.14894
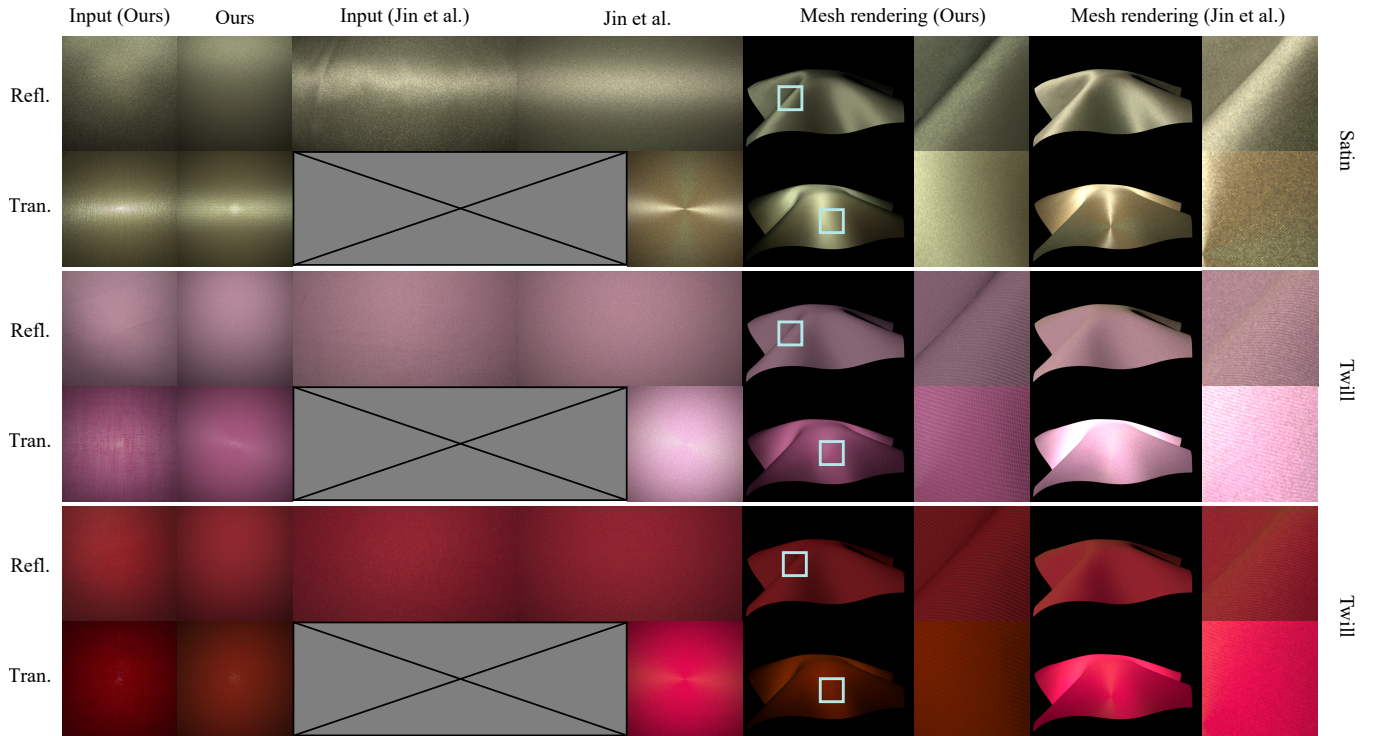
Figure 13: Given an input image captured with our measurement configuration, our inverse model is able to produce closely matching results. The rendered results on the draped cloth mesh also show a natural appearance. Note that the shadowing-masking effects are included for the mesh renderings.
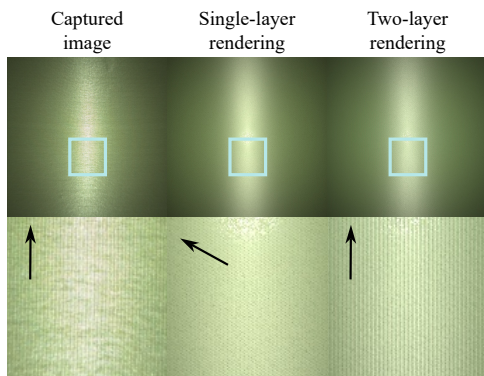
Figure 14: The impact of the two-layer shading model. By comparing with the single-layer shading model, we find that our two-layer model can reproduce the continuous vertical yarns shown in the captured image. In contrast, the single-layer model shows oblique stripes (as the arrow indicates), which is inconsistent with the captured image.
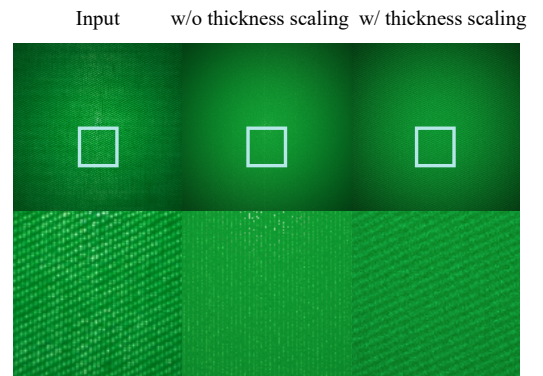
Figure 15: Comparison between the rendered results with and without our tension-aware thickness scaling. With the tension-aware thickness scaling, the overall diagonal structure can be represented, which is missing in a constant thickness.

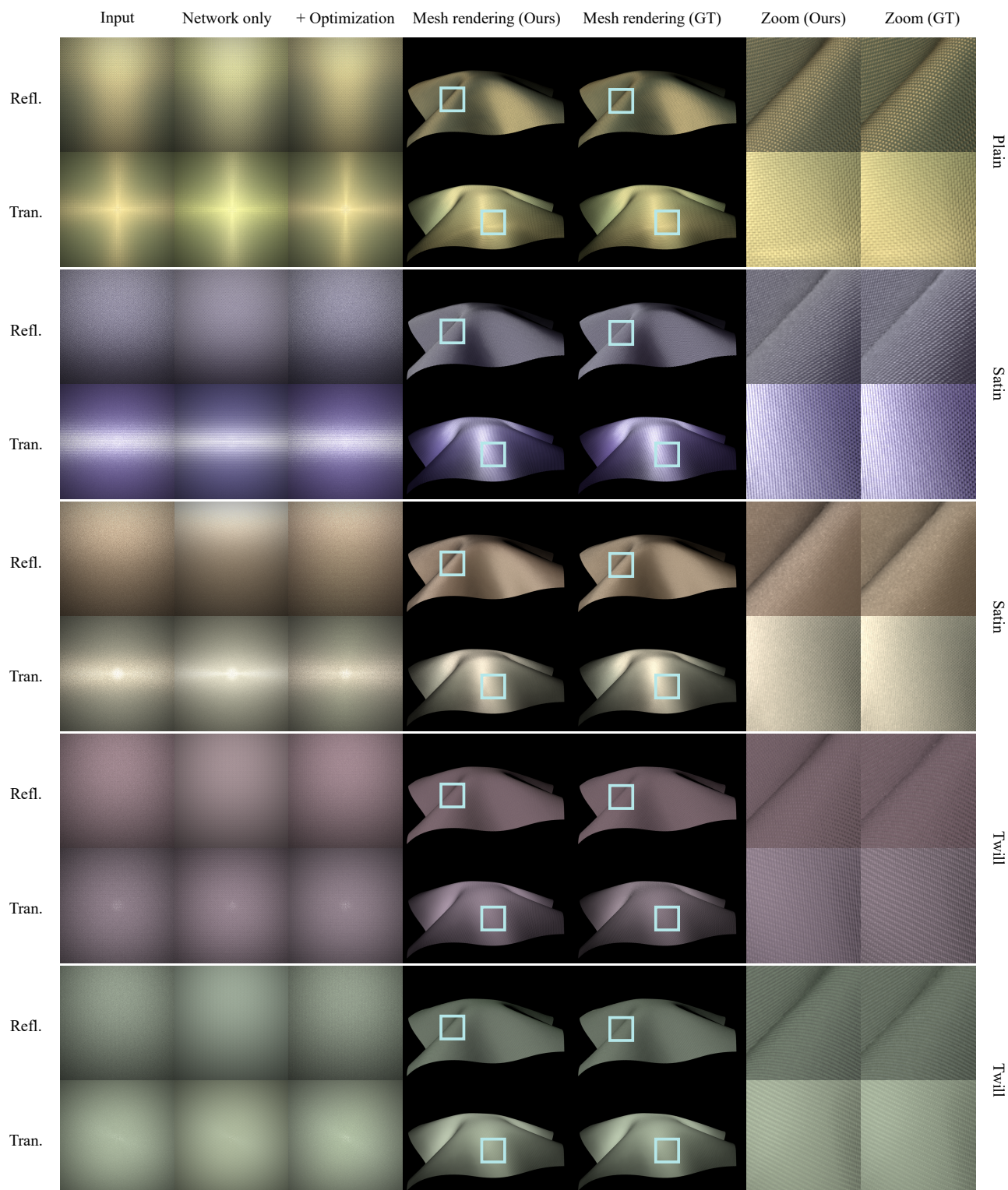Yingjie Tang, Zixuan Li, Miloš Hašan, Jian Yang, and Beibei Wang

Figure 16: 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. Note that the shadowing-masking effects are included for the mesh renderings.