

Automatic Reconstruction of Woven Cloth from a Single Close-up Image

C. Wu¹ , A. Khattar¹ , J. Zhu³ , S. Pettifer¹ , L. Yan²  and Z. Montazeri¹ 

¹University of Manchester, United Kingdom

²Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE.

³Shandong University, China

Abstract

Digital replication of woven fabrics presents significant challenges across a variety of sectors, from online retail to entertainment industries. To address this, we introduce an inverse rendering pipeline designed to estimate pattern, geometry, and appearance parameters of woven fabrics given a single close-up image as input. Our work is capable of simultaneously optimizing both discrete and continuous parameters without manual interventions. It outputs a wide array of parameters, encompassing discrete elements like weave patterns, ply and fiber number, using Simulated Annealing. It also recovers continuous parameters such as reflection and transmission components, aligning them with the target appearance through differentiable rendering. For irregularities caused by deformation and flyaways, we use 2D Gaussians to approximate them as a post-processing step. Our work does not pursue perfect matching of all fine details, it targets an automatic and end-to-end reconstruction pipeline that is robust to slight camera rotations and room light conditions within an acceptable time (15 minutes on CPU), unlike previous works which are either expensive, require manual intervention, assume given pattern, geometry or appearance, or strictly control camera and light conditions.

CCS Concepts

• **Computing methodologies** → **Computer graphics; Rendering;**

1 Introduction

Woven fabrics play an important role across various domains, ranging from online retail to entertainment industries, such as games and movies. The digital replication of fabric samples, a process essential in these fields, presents significant challenges due to their complex microstructure. This structure, characterized by repeated weave patterns composed in turn of yarns, plies, and fibers, greatly influences the overall appearance, despite not being explicitly visible.

Previous approaches employ generic shading models for inverse rendering [DAD*18; GSH*20], which often result in suboptimal outcomes for fabrics with special micro-geometry. Other methods that leverage surface-based shading models specialized for cloth often either overlook the transmission and microscopic details in close up [JWH*22], or require manual interventions [WCZ*19]. Alternatively, image-based methods [SZK15; GHCG17] can recover fabrics with fine-grained details in the form of texture, but can not generalize to different camera views and light conditions. However, the Bidirectional Texture Function (BTF) as an intensive data-driven method [KBD07] is rather expensive and impractical to capture and store. Similarly, complex capture setups such as CT

scans have also been employed to reproduce the microgeometry of fabrics [ZJMB11], which are still expensive to capture.

More recently, a method using a surface-based shading model has been proposed [TLH*24] to adapt transmission, but it requires two images captured under two different light settings and still only handles far-view images without including weave patterns estimation, and strictly fixed camera and light setup, which constrains its practical application. Although they claim that their method can be used after applying pattern estimation methods, we find it suboptimal to divide the full problem into two parts, separately estimating pattern and appearance. We elaborate on this in our ablation study in § 7.5.

In response to these limitations, we propose an inverse rendering pipeline aimed at procedural cloth parameter estimation using one captured image in close-up with a mobile phone. Our method encompasses the estimation of pattern, geometry, and appearance parameters, ranging from discrete parameters such as the number of plies and fibers, to continuous parameters like reflection and transmission components. Building on a realistic yarn-based shading model [KZP*24], our approach integrates explicit modeling of yarn, ply, and fiber geometries using B-spline curves defined by control points. This detailed modeling better fits the light reflection and transmission in the captured images. To pursue global op-

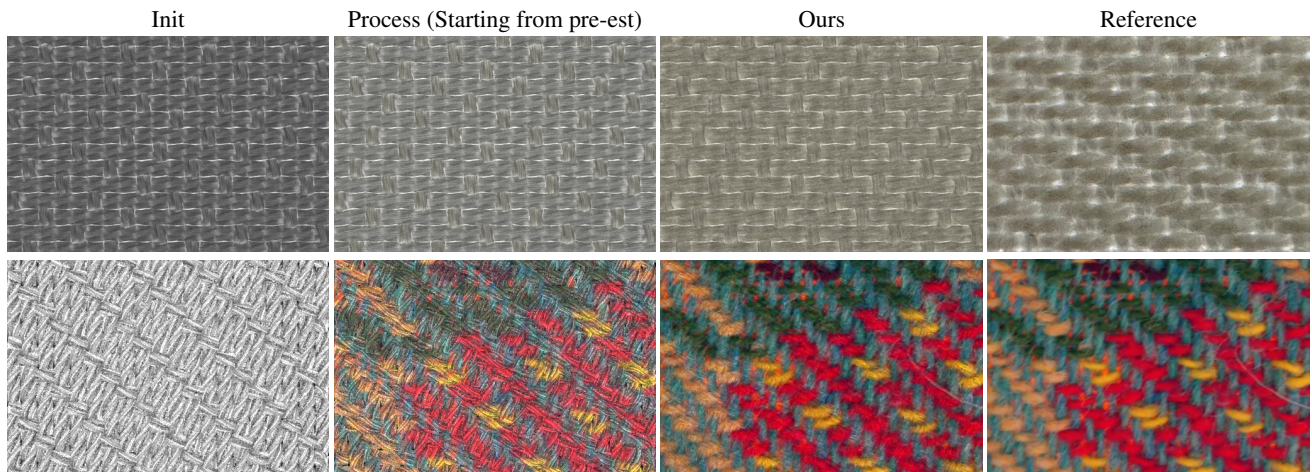


Figure 1: From a single input image capturing a woven fabric on a close-up, our method estimates a comprehensive set of parameters for digital cloth replication. It identifies weave patterns, geometry details such as ply and fiber count, alongside key appearance components such as reflection. This is achieved through a joint discrete-continuous optimization approach, with Simulated Annealing and Gradient Descent, to ensure a possible match to the reference image without prior assumptions or manual effort. Please play the animation in the second column or the video in the supplementary materials, showcasing the process of joint optimization of pattern, geometry, and appearance. Note that the animation shows the pre-estimation in the beginning and the post-estimation at the end.

tima under various ambiguities, we combine differentiable rendering with simulated annealing [KGV83] to address both differentiable and non-differentiable parameters. In summary, our work's contributions include:

- an end-to-end inverse rendering pipeline that recovers both the pattern and the appearance parameters of woven fabrics without any manual interventions and fixed camera and light settings.
- a neural network to provide initial parameters for optimization.
- a joint optimization strategy for discrete and continuous parameters combining Simulated Annealing and Gradient Descent.
- a post-processing step of approximating noise and irregularities with a set of 2D Gaussians.

2 Related Works

2.1 Cloth Pattern

The creation of woven patterns involves interlacing horizontal and vertical yarns known as warps and wefts, which are themselves composed of plies that in turn made up of hundreds of fibers [IM12]. Due to the alternative ups and downs of warps and wefts, a weave pattern can be efficiently represented by a binary matrix. The estimation of patterns from an image has two steps, detecting the minimum size, and estimating a binary matrix representing weft and warp.

Previous methods [ANB12; ABN13] used the superposition of distance matching functions (DMFs) of all rows and columns from an image and its forward difference to get the size of the smallest pattern. An alternative approach [RSP*19] extracts candidate patches from CNN activations of an input image and tiles them to compare the perceptual loss with the input to confirm the size of the

minimum pattern. While these methods work for general repeated patterns, but do not estimate the binary matrix of woven patterns as its special nature.

Some studies estimate the binary matrix by binary classification based on image features. Such as the histograms of image gradient [XNZL14], gray means, variances, and level co-occurrence matrix (GLCM) [JXL15], multiscale wavelet features [HC18] and used fuzzy c-means clustering (FCM) to classify these features into weft and warp. Besides to FCM, other classification methods are also proposed, like Support Vector Machine (SVM) [GHCG17], which is used to classify yarn segments from region growing. In addition, a deformable part model (DPM) trained using latent SVM [WCZ*19] is proposed to detect the yarn bounding box. However, it requires manual adjustments, which take about 3-5 minutes for a trained user in addition to the 15 minutes of the run time. Jin et al. [JWH*22] trained a neural network on a synthetic dataset to estimate patterns, but due to the limited dataset, it only covers three types of patterns and their 90-degree rotations. While these methods work for general cases, they treat the weave pattern as an independent part of the woven cloth appearance. Once the estimated pattern is incorrect, it hinders the following geometry and appearance estimations, and thus makes an end-to-end recovery pipeline without manual corrections impossible.

2.2 Cloth Geometry

The study of cloth geometry heavily emphasizes the microstructure of the material. One notable approach to understanding cloth microstructure involves the use of CT scan data. This technique, explored in depth by [ZJMB12] provides a detailed volumetric analysis of cloth. However, the CT scan method is particularly expensive, and the resultant data is challenging to edit.

Additionally, Schroder et al. [SZK15] made significant strides in this field by presenting a procedural yarn model. This model,

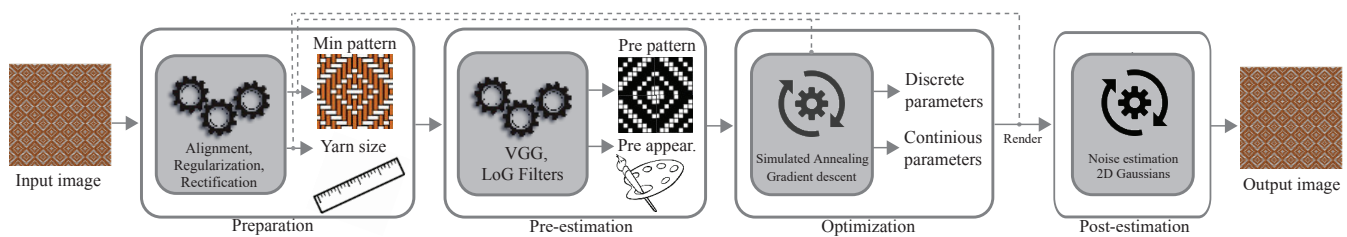


Figure 2: Overview of our Pipeline. The process begins with a captured image as input and aims to produce a digital replica of the fabric. Initially, we align and rectify the image to mitigate the influence of capture settings, and regularize the image for pre-estimation. Then, we extract the minimum repeating weave pattern from the regularized image and determine the yarn size and spacing. Using this information, we pre-estimate the weave pattern and appearance parameters of the weft and warp, setting the initial conditions for optimization. In the following stage, we jointly optimize both discrete and continuous parameters through a combination of simulated annealing and gradient descent, resulting in an accurate estimation of pattern, geometry, and appearance aspects of the fabric. Finally, we estimate the noise and irregularities of captured images with a set of 2D Gaussians and apply it to rendering results to provide better realism.

while effective in capturing the intricate details of yarn structures, required manual selection of model parameters, posing challenges in terms of usability and flexibility. Building on this, Zhao et al. [ZLB16] further developed this concept by fitting the model at a fiber level, which enhanced precision but was resource-intensive in terms of storage and rendering, thus restricting practicality. Notably, a fiber-based model, comprising hundreds of fibers per yarn, typically demands at least ten times the rendering time and a hundred times more storage compared to a yarn-based model.

Further, Guarnera et al. [GHCG17] focused on estimating yarn parameters within both the spatial and frequency domains for woven fabrics. Their methodology facilitated high-quality yarn-level recovery, marking a substantial improvement in the accuracy of cloth simulations. Despite the advancements in geometry recovery, they utilize generic appearance models and are not optimal for fabrics, specifically.

Wu et al. [WCZ*19] developed a method to estimate fiber orientations and discretize them into sticks. By tracing the sticks, they obtain the fiber geometry. However, this technique is based on a surface-based model by Irawan and Marschner [IM12] and shows a particular limitation when dealing with transparent yarns.

Most recently, Trunz et al. [TKM*24] proposed a neural inverse procedural modeling to capture detailed fiber and yarn geometry parameters. However, this relies on a single loose yarn and disregards the appearance parameters and pattern. Please refer to the survey for more work in this area by Castillo et al. [CLA19].

2.3 Cloth Appearance

A key approach in cloth appearance is the capture of Spatially Varying Bidirectional Reflectance Distribution Functions (SVBRDF). Some studies like [LDPT17; DAD*18; GSH*20] primarily utilized single-image methods to estimate the parameter maps of a generic shading model, encompassing attributes like albedo, normal, and roughness. However, all these methods use maps and are non-procedural. Besides, they are not specifically optimized for fabric rendering, leading to potential inaccuracies in capturing the unique visual characteristics of fabrics.

Hu et al. [HDR19] introduced a method for predicting procedural material parameters from input images. However, it was also

designed for general material types and did not specifically target cloth materials. As a result, while effective in a broader context, this method is not optimal for the unique challenges of cloth appearance.

The development of micro-appearance models for cloth is mainly derived from the individual strands of fabric structure, similar to hair [MJC*03]. Irawan and Marschner [IM12] and later several yarn-level appearance models [SBDJ13; ZMA*23; SM23] built upon the individual hair model and proposed forward surface-based appearance models for fabrics. These models lack the fiber-level details and they do not offer an inverse framework for parameter fitting.

In a departure from traditional surface-based models, Khungurn et al. [KSZ*15] computes cloth appearance using a curve-based model at the fiber level. This method represents high-fidelity renderings of fabrics, capturing the intricate details that are often missed in previous models. However, its practical application is hindered by the significant resources required for both storage and rendering time.

To tackle these challenges, a ply-based approach was proposed as an encapsulated cloth model that enhances the performance by aggregating the light interaction in the ply level for woven ([MGZJ20]) and knitted ([MGJZ21]) fabrics. However, their inverse rendering framework is basic and does not yield the most accurate results. Later, Khattar et al. [KZP*24; KZYM25] adapted the model to function at the yarn level, thereby enhancing performance even further while maintaining the fiber detail fidelity. In our research, we incorporate this model into our own inverse framework.

Recently, Wang et al. [WJHY22] proposed the Sponge-Cake model, defining each layer of the fabric as a volumetric medium. A key feature is its use of microflake distribution to describe these layers. Following this, Jin et al. [JWH*22] introduced the inverse-SpongeCake model for cloth, but with a notable limitation of three types of weave patterns without considering transmission, and the assumption of knowing the camera and light. The following work [TLH*24] extends it to adapt transmission by requiring one more image with only back-lit on to pair with the one captured with front-

lit, but still has the limitation of only considering a few types of patterns, and the same assumption of knowing the camera and light.

3 Background

3.1 Cloth Structure

In textile modeling, particularly for woven fabrics, a fundamental characteristic is their construction from repeated weave patterns. Instead of requiring a comprehensive analysis of the entire fabric, our approach focuses on identifying and estimating the parameters of the smallest unit of repetition, referred to as the "minimum repeating pattern". Once the geometry and appearance of the weave pattern are determined, the yarns will be tiled to reconstruct the entire fabric by copying the minimum repeating pattern along the surface.

3.2 Forward Model

As the study of the cloth forward model develops, many advanced techniques have shown up. While our framework can utilize any forward model, we chose to build upon the state-of-the-art yarn-based model [KZP*24] with explicit representation of yarn curves using B-spline curves. The plies and fiber details are added procedurally using texturing techniques, allowing for a practical yet relatively accurate geometric model. Our forward shading model employs their approach, capturing the intricate interactions of light with the yarn using the Bidirectional Yarn Scattering Distribution Function (BYSDF). It encompasses four components, including specular reflection and transmission, accounting for the light that directly escapes from the fabric's front and back surfaces, respectively. Similarly, the model also considers body reflection and transmission, which represent the light that scatters as it interacts with the yarn, emerging from both the front and back sides of the fabric. Each of these four components is governed by a set of parameters, all of which are included in Tab. 1 and discussed in § 3.3. The main advantage of adopting this model as our forward rendering tool, compared to more conventional models [JM12], is due to its capability to accurately simulate light reflection and transmission. This feature, often overlooked in prior research, aligns more closely with the requirements of our practical inverse-rendering approach. We will further elaborate on it in Fig. 9 and § 6.4.

3.3 BYSDF

The BYSDF model consists of forward and backward components to capture the part of the light that bounces back and the part that travels forward. The backward component captures both the immediate reflection, which is part of the specular property f_x^S , and the scattered light that exits the medium from the same side as the incident ray, referred to as the body property f_x^B . Then, using the transmission component of f_x^S , we sample a point y on the yarn surface as the exit point, following the GGX distribution. Lastly, the forward component represents both the specularly transmitted light f_y^S and the scattered light f_x^B that continues in the forward direction.

Specular Component: The specular components f_x^S and f_y^S represent the prominent highlights on the fabric surface when light reflects immediately or transmits through the fabric without being scattered. The BYSDF model utilizes a rough dielectric BSDF with

Table 1: Shading model parameters optimized in our experiments. Warp yarns are marked with a subscripts and weft yarns with e subscripts. The top three parameters are discrete, and the remaining are continuous parameters. The right column shows the empirical range of parameters for the dataset generation, VGG prediction, and regularization during joint optimization.

	Name	Meaning	Type	Range
Discrete	P	Weave pattern	Array	{0,1}
	n_p	Number of ply	Int	[1,7]
	n_f	Number of fiber	Int	[20,50]
Continuous	m_r	Specularity coefficient	Float	[0.2,0.8]
	R^S	Specular reflection	Color	[0,1]
	T^S	Specular transmission	Color	[0,1]
	T_y^S	Specular transmission at y	Color	[0,1]
	α^S	Specular roughness	Float	[0,1]
	α_y^S	Specular roughness at y	Float	[0,1]
	η^S	Specular IOR	Float	[1.4,1.6]
	η_y^S	Specular IOR at y	Float	[1.4,1.6]
	A^S	Specular attenuation	Float	[9,11]
	A^B	Body attenuation	Float	[0.1,0.7]
	R_e^B	Body reflection / Texture (weft)	Color/Array	[0,1]
	R_a^B	Body reflection / Texture (warp)	Color/Array	[0,1]
	S_e^B	Body subsurface reflection (weft)	Color	[0,1]
	S_a^B	Body subsurface reflection (warp)	Color	[0,1]
	r_e	Radius of weft	Float	[1,5]
	r_a	Radius of warp	Float	[1,5]
	G_p	Ply-shadowing function	Float	[0,1]
	G_f	Fiber-shadowing function	Float	[0,1]

a GGX distribution.

$$f_x^S(\omega_i, \omega_o) = \begin{cases} R^S \cdot \frac{F_s G_2(\omega_i, \omega_o, \omega_h) D(\omega_h; \alpha^S)}{4|\omega_i \cdot n(x)| |\omega_o \cdot n(x)|}, & (\omega_i \cdot \omega_o > 0) \\ T^S \cdot \frac{|\omega_i \cdot \omega_o| |\omega_o \cdot \omega_i|}{|\omega_i \cdot n(x)| |\omega_o \cdot n(x)|} \cdot \frac{(\eta^S)^2 (1 - F_s) G_2(\omega_i, \omega_o, \omega_h) D(\omega_h; \alpha^S)}{[(\omega_i \cdot \omega_i) + \eta^S (\omega_o \cdot \omega_i)]^2}, & (\omega_i \cdot \omega_o < 0) \end{cases} \quad (1)$$

$$f_y^S(\omega_i, \omega_o) = \begin{cases} 0, & (\omega_i \cdot \omega_o > 0) \\ T_y^S \cdot \tau(x, y, A^S)^{N_i} \cdot \frac{|\omega_i \cdot \omega_o| |\omega_o \cdot \omega_i|}{|\omega_i \cdot n(y)| |\omega_o \cdot n(y)|} \cdot \frac{(1 - F_y) G_2(\omega_i, \omega_o, \omega_h) D(\omega_h; \alpha_y^S)}{[\eta_y^S (\omega_i \cdot \omega_i) + (\omega_o \cdot \omega_i)]^2}, & (\omega_i \cdot \omega_o < 0) \end{cases} \quad (2)$$

where $\tau(x, y, A^S)^{N_i}$ is an attenuation between x and y using the Beer-Lambert law [Swi62].

Body Component: To capture the scattering behavior of the bundle of fibers as a whole and account for multiple scattering components, we utilize a diffuse-like distribution to approximate the sub-yarn scattering events [MGZJ20; YJR17]. At point y , the body component f_y^B is represented by a Lambertian term. On the other hand, f_x^B additionally considers a Lommel-Seeliger (LS) term [HK93; JMLH01] as a mix to ensure energy conservation. The expressions for f_x^B and f_y^B are as below:

$$f_x^B(\omega_i, \omega_o) = m_r G_p(x, \omega_i) G_f(x) R^B \left[LS(\omega_i, \omega_o) + \frac{1}{\pi} \right] \quad (3)$$

where $LS(\omega_i, \omega_o)$ refers to a Lommel-Seeliger term [HK93].

$$f_y^B(\omega_i, \omega_o) = (1 - m_r) G_p(x, \omega_i) G_f(x) \frac{S^B}{\pi} \quad (4)$$

In our experiments, due to the alternative and repetitive nature of weave patterns, we deploy two sets of body components for weft yarns R_e^B and warp yarns R_a^B respectively, and one set of specular components for both. However, it can be extended to yarn-wise for knitted cloth, or even sample point-wise for customized cloth. For example, to handle spatially varying colors, we extend the shading model by replacing the Body reflectance R_e^B and R_a^B with a texture map in our experiment § 7.2.

Self-shadowing Component: To address the limitations of existing yarn-based shading models, BYSDF introduces an additional shadowing component that considers the occlusion caused by one ply on another (G_p) and by one fiber on another (G_f). The multiplication of these two shadowing terms approximates the overall amount of shadowing, compensating for the absence of explicit ply and fiber geometries.

Yarn Gaps/Delta Transmission: As we only consider recovering a pattern with a fixed camera setting, the gap between yarns can be determined by the radius of yarn r_e and r_a . Even if the ground truth of the camera setting is unavailable, the cloth recovery remains proportional to the target and can be transformed to the desired scenes and re-rendered.

3.4 Discrete Parameters

Discrete parameters, such as weave patterns and the number of plies and fibers, are either considered as a presumption at user-level, which requires manual effort [WCZ*19] or greedily searched due to its non-differentiability, which fails to promise global optima. However, as discrete parameters contribute a great sense of realism, suboptimal estimation of them will hinder the final recovery quality.

Weave Pattern: Independent weave pattern detection has been well studied with various vision and learning techniques [ANB12; JXL15; MPG*21]. However, no methods promise 100% success due to the degree of freedom of irregularities in captured images and the loss of information in their image processing. Sometimes, even manual identification is not easy due to the distraction of color, such as the samples in Fig. 12. Previous methods [GHCG17; WCZ*19; SZK15] apply pattern detection before appearance estimation, which highly relies on correct pattern estimations. Incorrect patterns will mislead the appearance reconstruction and hinder the final cloth recovery results, as shown in Fig. 6.

Number of Ply/Fiber: The number of ply and fiber have large effects on cloth appearance both in far view and close view. However, they are difficult to estimate due to occlusions and irregularities of the plies and fibers. Previous image-based methods do not consider such parameters [SZK15; GHCG17], other methods either apply a greedy strategy in the optimization [JWH*22] or adjust them manually [WCZ*19].

Therefore, we consider discrete parameters to share the same optimization space with continuous parameters and have non-trivial influence on the convexity. We elaborate on the details of our method in § 6.

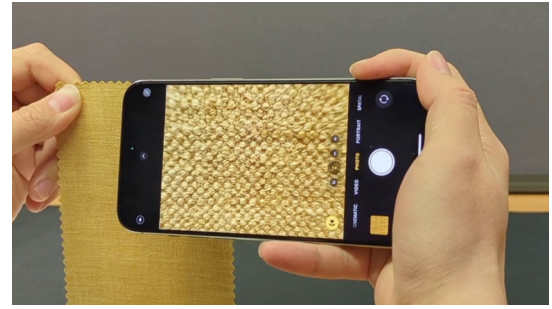


Figure 3: This shows our capture setup to obtain the input image using a handheld mobile phone, under uncontrolled room light conditions, allowing both front and back illumination by lifting the fabric sample in the air. Note that this image is captured by a different device; thus, the fabric sample's color looks different from the actual capture shown in Fig. 8 (e).

4 Preparation

The input to our system is a single captured image of a piece of cloth, taken by a handheld mobile phone from a close-up view where the yarns are distinctly visible. The lighting condition during capture is a room environment lighting that allows both reflection and transmission when lifting the fabric sample in the air, as shown in Fig. 3. We observed that when capturing at a very close distance, strict camera and light controls are less necessary, as slight camera shifts caused by handheld movement can be corrected through rectification, which will be introduced in § 4.1, and the room light is approximately soft and uniform.

The overview of our pipeline is listed in Fig. 2. In the first step, we implement a preparation phase designed to refine the input image, setting the stage for the most effective optimization. While this preparation step can be omitted for synthetic data, which is typically clean and well-defined, we have found it to be useful for processing real-world samples that may include a range of anomalies.

4.1 Rectification

One of the common challenges with real-world image capture is the bias, particularly in the orientation of the yarns, which may not align perfectly with the x and y axes of the image. To align this, we employ the Radon Transform [Gri86], a technique for feature extraction in image processing. By applying this transform, we can find the predominant angle of the yarns in the image, denoted as the angle of peak in the transform's output. This angle subsequently guides the rotation of the image, ensuring that the yarns align with the axes, thus standardizing the orientation for all input images. To eliminate the angular feature of highlight potentially introduced by the direction of light reflections, we first apply a Laplacian filter on the original image to ensure the edges of weft and warp yarns are being rectified instead of the edges of highlights.

Besides, the misalignment of the input image can also be caused by non-flat cloth swatches and camera perspective distortion, like involuntary rotations in pitch and yaw. We apply a RANSAC-based method [CDI14] to eliminate the other slight distortions and en-

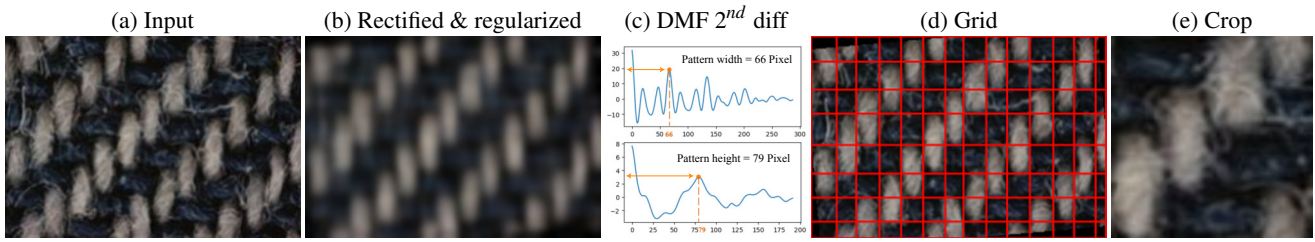


Figure 4: (a) Given a captured image as input, which is not aligned and blurry, (b) we first rectify and regularize it. (c) Then we analyze the second difference of DMFs to determine the pattern size in pixels. (d) The grid segmentation (fourth image) helps detect yarn density, (e) leading to the cropped minimum repeating pattern.

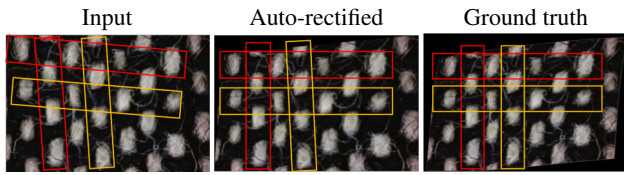


Figure 5: Given a distorted image as input, we can get the auto-rectified result close to the manually adjusted ground truth.

hance the robustness to non-professional capture settings as shown in Fig. 5.

4.2 Regularization

The objective of this step is to refine the input images into their "ideal" weave pattern representations. This involves the elimination of noise and irregularities that could potentially skew the optimization results. Our approach utilizes a Gaussian filter to reduce the noise while preserving the main features. In practice, we set the standard deviation of Gaussian to $1/16$ of yarn's size in pixels as a yarn usually contains 10-20 fibers in the vertical cross-section. Following this, we apply Kernel Principal Component Analysis (KPCA), a method for feature extraction and dimensionality reduction [SSM97]. It allows us to focus on the significant structural elements of the weave pattern by discarding fine details and irregularities that do not contribute meaningfully to the overall pattern.

4.3 Minimum Pattern Size Detection

This phase is for identifying the minimum repeating pattern size in the fabric's weave. To accomplish this, we utilize a common technique that uses the superposition of distance matching functions (DMFs) across the rows and columns of the image [ANB12] to detect the same copies of the grid. Analyzing the maximum second forward differences of these functions reveals the length and width of the minimum pattern, measured in pixels. As illustrated in Fig. 4 (c), the orange mark shows the minimum pattern size in pixels.

It is noteworthy that our approach maintains its effectiveness even for complex, custom-designed fabrics, where the number of repeats could be as few as one (no peaks in the forward difference of DMF).

4.4 Yarn Width Detection

To determine the yarn width, a spatial integral projection is applied in both horizontal and vertical orientations on the regularized

grey-scale image. This procedure yields two distinct graphs, within which the valleys serve as indicators of the yarn intersections. By measuring the average distance between these valleys, we can determine the yarn's size in pixels.

We divide the image into smaller cells of varying widths, transforming the weave into a grid-like visualization as shown in Fig. 4 (d). Such segmentation is needed for accurately reasoning the number of wefts and warps comprising the minimum weave pattern.

Our preparation step, while enhancing practicality, is not fixed and can be substituted with alternative methods. For instance, a user interface could be used, allowing users to manually draw the minimum pattern. Also, besides the clean and well-defined synthetic data, when the input image is not analyzable, this step can be bypassed entirely by the user, as an example shown in Fig. 16. Nevertheless, it is crucial to note that our automated approach yields precise and robust results, offering practicality and accuracy with no need for supervision.

5 Pre-estimation

5.1 Pattern Pre-estimation

An ideal weave pattern, characterized by its wefts and warps, can be effectively represented as a binary matrix. The distinguishing factors between weft and warp are their directions and their ups and downs. This forms the basis of our objective function, which is defined as the Mean Squared Error (MSE) between the rendered images split into cells and the corresponding cells in the regularized reference images.

To achieve this, we consider the aligned structure of a typical weave pattern and segment the weave grid into smaller sub-spaces based on their cross-sections, acquired in the previous step. The gridded input is shown in the first column in Fig. 6. For each cell, it is simplified into a binary state: either 0 or 1. Given the binary weave matrix, we generate the B-spline curves to align the pattern by adjusting the height of the control points to mimic the ups and downs. The maximum height of a control point is on the peak of the top yarn, set to the yarn width. For each cell, we set 12 control points by default, and the heights are sinusoidally smoothed using a Gaussian filter with a size of 16 and a standard deviation of 1. The generation pseudocode of B-spline curves is provided in Appendix A. To determine the optimal state for each cell, we employ the Simulated Annealing algorithm [KGV83] since the parameter space is discrete. In our implementation, the temperature is set to 0,

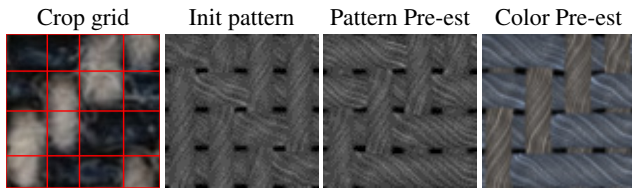


Figure 6: We pre-estimate the weave pattern, diffuse and specular color analytically from twill and grey by default. The output of pre-estimation will be used as initial parameters for the following joint optimization in § 6. Note that the pattern and color are not optimum since they are pre-estimated separately, the mismatch (at the bottom-right) of the pattern at the bottom right leads to a brighter color of wefts and a darker color of warps.

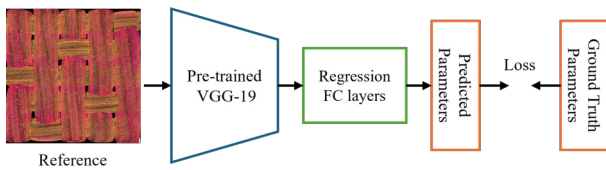


Figure 7: We pre-estimate the other unanalyzable parameters with a neural network that serves as a good initialization in § 6, similar to Fig. 6. The neural network can not promise optimal recovery directly either, because of the limitation of training data.

which is equivalent to looping over both states for all the cells and picking the best configuration. An example is shown in the third column of Fig. 6, given the init pattern.

5.2 Color Pre-estimation

Once the weave pattern is pre-estimated, the next step is determining a rough estimation to initialize the color of the weft and warp. This allows us to assign the diffuse colors for the weft and warp, derived from the rectified input image, averaged over the cells while ignoring the edge 10% from each side. We average the pixels of weft and warp that have top 10% intensity to assign the specular color. Mapping the colors to the corresponding cell provides our optimization setup with a plausible initialized state as shown in the last column of Fig. 6.

5.3 Parameters Pre-estimation

Once the weave pattern is pre-estimated and a rough estimate for color is determined, the next step is to determine a rough estimation to initialize the rest parameters of weft and warp. For these parameters that are difficult to analyze, we propose a compact network to pre-estimate the parameters of woven fabric. This provides a plausible initialized state to be used during our subsequent optimization.

As shown in Fig. 7, this process begins with the use of a pre-trained VGG network to capture the features of the input image. Then, we employ two fully connected (FC) layers activated by LeakyReLU to predict a given parameter. A Sigmoid function, adjusted with specific weight and bias, is then applied to calibrate the output range. This adjustment enables prior knowledge of each pa-

rameter's estimation. For instance, the Index of Refraction (IOR) is restricted within the range of [1.4, 1.6] for feasible outputs. The ranges of other parameters are shown in Table 1. For each parameter, we train new FC layers while keeping the pre-trained VGG fixed. For the training data, we generate 2000 images by randomly sampling the parameter space as listed in Table 1. The generation of explicit yarn geometry for different patterns in our dataset is provided in Appendix A.

When input images are unanalyzable or never learned by the network, the pre-estimation step can also be bypassed by the user, which makes the joint optimization longer to converge. An example is shown in Fig. 16. In the ablation studies in § 7.5, the effect of using pre-estimation as an initial state for our optimization is elaborated in comparison to the randomized initialization.

6 Optimization Method

With the weave pattern and parameters pre-estimated, we achieve a solid starting point for initiating joint rendering. This process allows us to optimize both discrete and continuous parameters. The loss function employed is the Mean Squared Error (MSE) between the rendered and reference images. The complete list of parameters that are optimized in our study is detailed in Table 1.

6.1 Continuous Parameters

Continuous parameters exhibit a direct correlation with changes in the rendering, where changes can be tracked through derivatives, hence directly differentiable. These parameters follow the formulations of the yarn-based model capturing four components as stated in § 3.3 to represent forward/backward reflection and body components. However, continuous parameters need to be constrained within a reasonable range during optimization; thus, we add a barrier regularization similar to clamp:

$$L_b = \lambda \cdot (\max(0, x - a)^2 + \max(0, b - x)^2) \quad (5)$$

where x is the continuous parameter, a and b refer to the corresponding lower and upper bounds of the data ranges in the Tab. 1. In our experiments, λ is set to 0.01.

6.2 Discrete Parameters

Discrete parameters, such as weave patterns (including types like twill, satin, or custom designs like floral patterns) and the number of plies and fibers, do not correspond to a continuous change in the rendering output when changes so not directly differentiable. The optimization process presents a unique challenge when discrete parameters mix with continuous ones, potentially disrupting the convexity of the objective space. To navigate this, we optimize discrete parameters simultaneously with continuous ones. Instead of greedy searching, we employ simulated annealing as a probabilistic strategy to approach global optima for discrete parameters such as the number of plies, the number of fibers, and the weave pattern.

In the context of optimizing the number of plies and fibers, we initiate simulated annealing with an initial temperature of 100 and gradually decay at a rate of 0.98. The temperature is set to this value so that a particular percentage of uphill proposals is accepted at the beginning. This specific approach is chosen because these

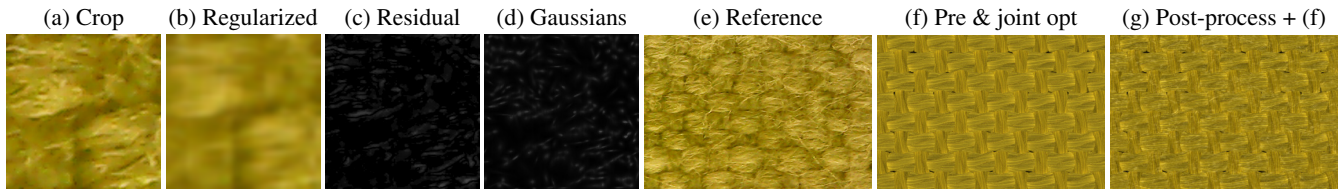


Figure 8: (a) is a crop of minimum pattern from the input reference image; (b) refers to the regularized cropped image; (c) illustrates the residual, which is the difference between reference and regularized image. (d) demonstrates the fitting of residuals with a set of 2D Gaussians. (e) shows reference input image and then (f), shows the result of our joint optimization. Finally, in (g), we add the Gaussians to provide more sense of reality.

parameters significantly influence the finer details of the cloth image, thus increasing the likelihood of encountering multiple local minima within these dimensions.

6.3 Noise Estimation

Inspired by previous work [SZK15; GHCG17], we also estimate the image noise caused by flyaways, highlights, and other irregularities as a post-estimation process. We use a set of 2D Gaussians to estimate the frequency spectrum of these fine details. In the synthesis of tiling, we sample the frequency spectrum from the estimated Gaussians and transform it back to the time domain for the new patch. After joint optimization, we add the estimated noise on the rendering as post-processing to provide more sense of reality, as shown in Fig. 8.

6.4 Light Setting

Our method does not target the recovery of lighting but only the properties of cloth under common room light conditions. As our shading model considers both the reflection and transmission components, our method does not necessarily require a controlled light for the captured input, but allows natural room light environments, such as that is shown in Fig. 3. In Fig. 9, we compare our recovery results with the references in three light conditions: with only front, back, and both lit. The match of backlit appearance in Fig. 9 (b) and (c) elaborates our shading model's ability to handle the transmission component. Note that the three references are from the same fabric swatch. In our experiments where the ground truth of the light setting is unavailable, such as the public real data in Fig 10 (c), we assume that the input is captured in a natural room light condition and deploy an ambient light with four directional lights around the cloth in the optimization by default. For the synthetic data in Fig 10 (a), where the light setting is available, we faithfully follow it.

7 Results

7.1 Main Results

Fig. 10 demonstrates the robustness and precision of our inverse rendering framework, evaluated across both synthetic and real data. For all the results through the paper, we used the path-traced differentiable renderer Mitsuba 3 [JSR*22], using an Intel i7 CPU. We trained the FC layers in the neural network with a single NVIDIA RTX 3080 GPU. The total time cost for the pipeline is about 15 minutes, with 3% allocated for preparation § 4, 17% for determining the initialization § 5, and the remainder dedicated to the actual

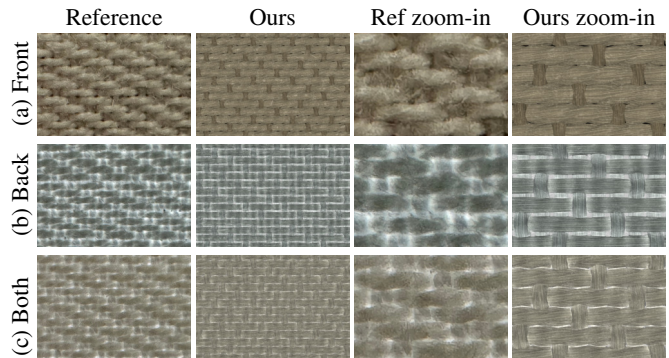


Figure 9: Our reconstructed results of the same fabric illuminated with only front, back, and both lit.

optimization process § 6. If not otherwise specified, the light setting in the experiments of this section is an ambient light with four directional lights from the front, back, and two sides that mimic common real-life environments.

Synthetic Data Evaluation: In our experiments with synthetic data, shown in Fig. 10 (a), we tested our model on different fabric types. The use of synthetic data enabled us to establish an exact ground truth, shaded by our forward model so we could assess the effectiveness of our method. Initially, our pre-estimation state yielded a rough alignment in the overall fabric pattern and appearance. Then, subsequent optimization steps improved the accuracy of the parameter recovery by jointly optimizing a pool of discrete and continuous parameters. Additionally, our model can manage fabrics with varying warp and weft appearances for any arbitrary weave pattern. Our model offers an advancement over previous methods like [WCZ*19] and [JWH*22], providing the ability to recover transparency and support multiple plies, unlike their reflectance-only and single-ply approach.

Real Data Application: We also applied our pipeline to our actual measured fabric data with a handheld mobile phone and public source fabric data[†] as illustrated in Fig. 10 (b) and (c). Given the absence of ground truth parameters for these real samples, our evaluation focused on the visual similarity between the input images and the images rendered using the estimated parameters. Renderings performed on a non-flat mesh in the last column further show the plausible appearance achieved by our model.

[†] The reference images we use are available in <https://exteta.it/en/materials/fabrics/>

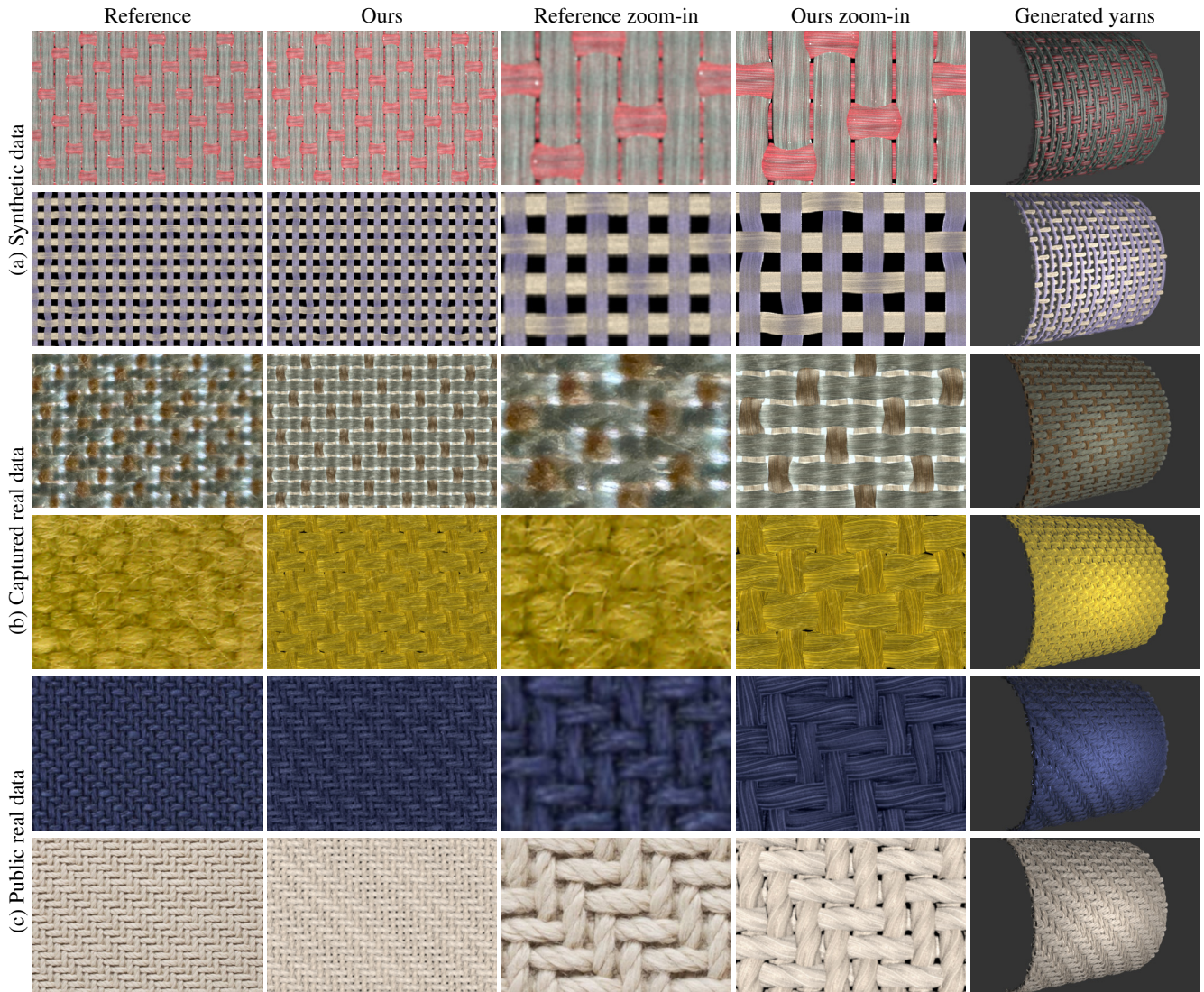


Figure 10: The results of our method given synthetic (first two rows), our mobile phone captured real pictures (middle two rows), and public source real data (last two rows) as input. The first two columns show captured or generated images as references and our corresponding recovery. In the third and fourth columns, we show a zoomed-in reference cropped from the first column and our reconstruction re-rendered with the camera moved closer. In the last column, we show the generated yarns in a cylinder shape with the recovered parameters under a point light condition.

Parameter Error Evaluation: We also evaluate the recovery of parameters. As the ground truth parameters of real data are unavailable, we only plot the parameter errors of synthetic data in Fig. 10 (a). As shown in Fig. 11, the high-dimensional joint optimization has a very unsmooth convergence. Several local minima can be observed at around the 20th and 55th iterations; nevertheless, after the 55th iteration, most parameters come to convergence. The notable turning points are largely related to the matching of pattern P , which greatly affects the distribution of appearance in the image space. For example, in the case of synthetic data 1 (blue curve), the error curve of the pattern shows a slight increase from the 25th to 55th iterations, which correlates to many other parameters such as m_r , A^B , S_e^B , and S_d^B . After the pattern P converges to

the global optimum, most parameters reach it as well. This shows that our proposed joint optimization can, to a certain extent, mitigate the optimization challenges arising from ambiguity between discrete and continuous parameters. Note that the parameter from pre-estimation calculates the initial error.

7.2 Spatially Varying Color

Our method can support spatially varying colors as well. In this experiment, we replace the body reflection component of warp and weft (R_e^B and R_d^B) with a texture map instead of a single color. The initial state of the texture map is set to a matrix initialized with ones by default. The results are shown in Fig. 12. The reference images

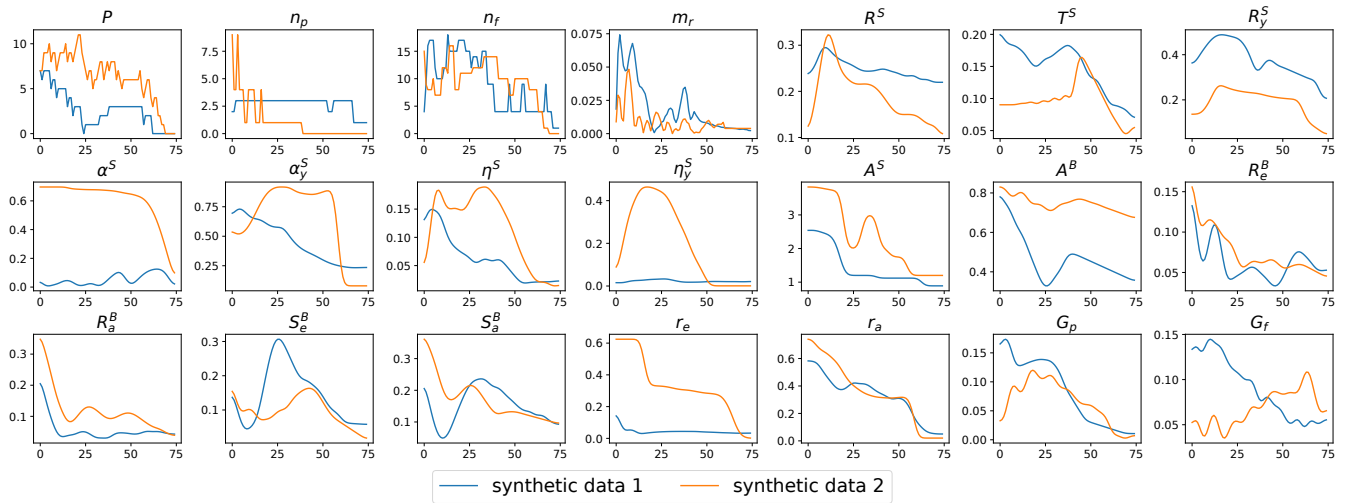


Figure 11: The parameter error plot of our synthetic data in Fig. 10 (a).

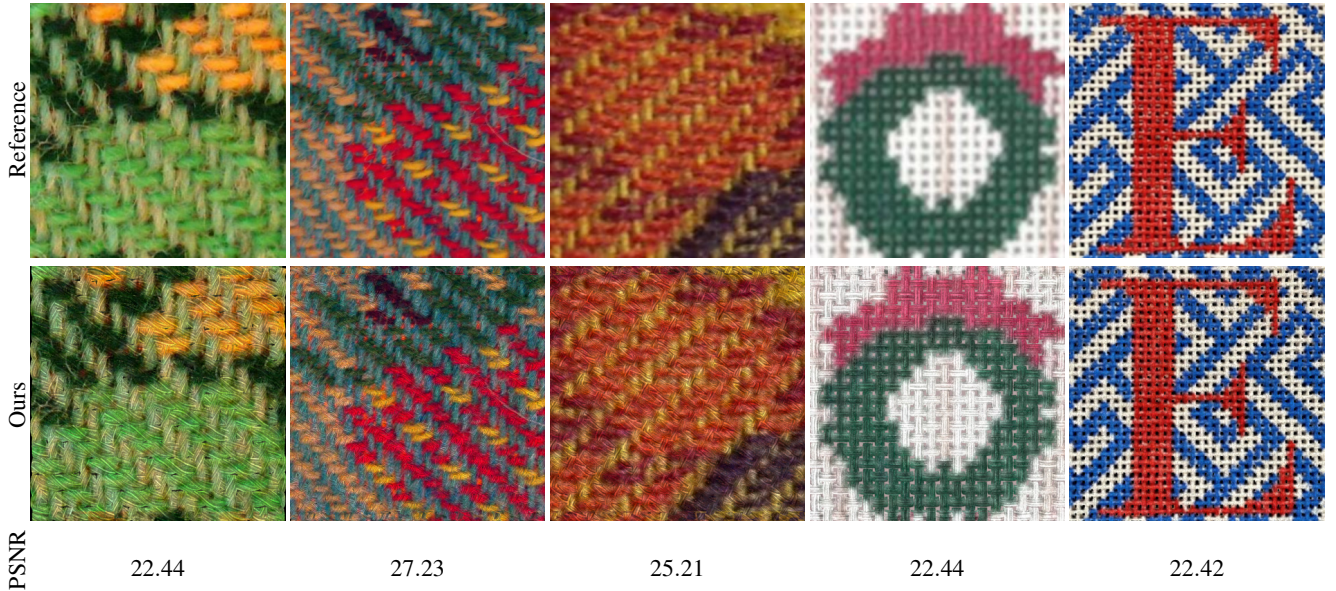


Figure 12: Our recovery of woven fabrics with spatially varying color and the PSNR to reflect the visual quality of recovery results.

are from public sources[‡]. Note that as the texture map covers the noise and irregularities as part of the spatially varying color, the post-estimation step has less improvement in this case.

7.3 Customized Pattern

Besides, our method also has no limit to the types of weave patterns because we explicitly model yarn curves to constitute a pattern. Combining simulated annealing, our method can recover complex customized patterns along with appearance, even those that do not exist in the real world, without any priors or assumptions, and still reach global optima. This advantage allows users to customize the

[‡] The left three columns are from <https://tapadesi.com/>, which are also captured by a personal mobile phone. The right two columns are from <https://www.oldworlddesigns.com/>

fabric by capturing any image with a mobile phone, while previous work [WCZ*19; JWH*22; TLH*24] cannot. In this experiment, as the reference has no features of woven fabrics (wefts and warps) and has never been learned by the network, we bypassed the preparation and pre-estimation steps and directly joint-optimized it from a default initialization, which still converged while taking more iterations. The results are shown in Fig. 16, and the joint optimization animation is in the supplementary materials.

7.4 Comparison with Previous Works

Our framework's capabilities have been benchmarked against two previous works in the field. It is important to acknowledge that, to the best of our knowledge, there is no prior end-to-end pipeline that matches all of the pattern, geometry, and appearance aspects

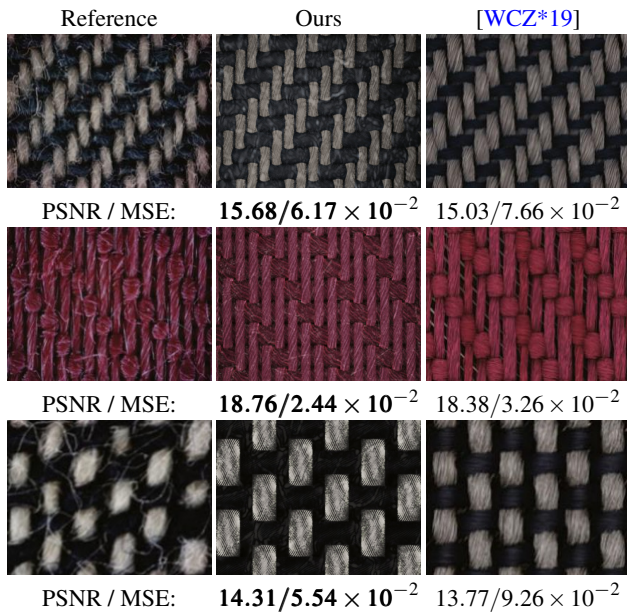


Figure 13: The comparison of our method with [WCZ*19], which recovers the geometry but manually tweaks the yarn layout using a yarn texture model, while ours recovers pattern, geometry, and appearance without any manual intervention.

of a captured fabric image. For a fair comparison, we adapted our pipeline to focus on aspects optimized by these earlier methods.

In our comparison with [JWH*22], which assumes weave patterns only among twill, plain, or satin using a surface-based shading model, we matched their conditions by fixing the geometry alignment and focusing on appearance parameters. As shown in Fig. 14, our results demonstrate a reasonable match with the reference and [JWH*22], reflected both qualitatively and quantitatively. The third row has slightly worse quantitative results compared to [JWH*22], because our advantage lies in the recovery of yarn-level details, but the input reference is taken at a distance without clear yarn details. However, in the overall view, our results are still comparable. Besides, unlike their model, which only relies on the network to estimate the pattern and is suitable mainly for distant renderings, our method can optimize patterns for close-up views, offering high-fidelity details. Moreover, our adaptive and fully procedural approach contrasts with their training requirement for neural networks, which takes 11 hours, yet is still limited by datasets.

Additionally, we compared our results with [WCZ*19], as shown in Fig. 13, who estimated geometry by tracking fibers and used a yarn texture model for shading, which lacks transmission. Our model outperforms this by using a specific cloth shading model and explicitly defining yarn curves, resulting in a more detailed appearance compared to their method. While their system requires micro-images, our model is adaptable to any close-up images where yarns are visible. Our method is more comprehensive, completing the process automatically without any manual adjustments needed, as opposed to [WCZ*19]'s method with manual intervention by an expert user to adjust the yarn layout. Given these irregular and

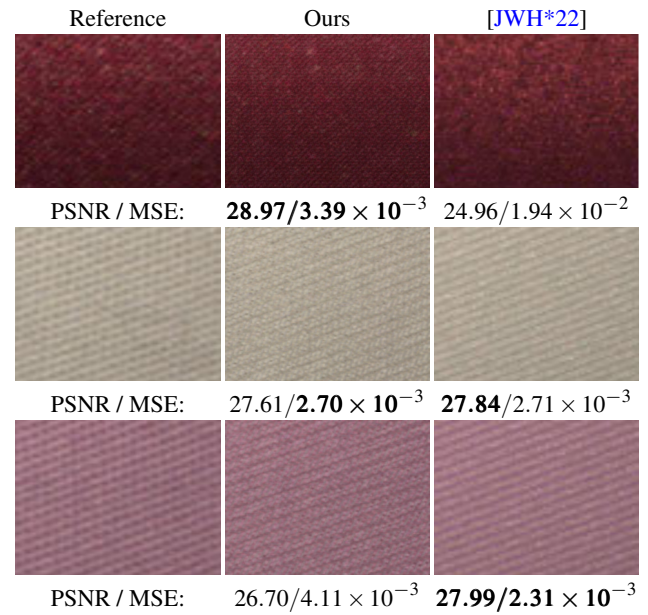


Figure 14: The comparison of our method with [JWH*22], which is limited to five given patterns, while ours includes pattern estimation in the pipeline. Our yarn-based method can achieve comparable results both qualitatively and quantitatively to their surface-based method, even given relatively blurry far-view input photos. For the first row, please zoom in to compare the diagonal details.

fuzzy inputs, even though previous work explicitly traces the fibers, our method still achieves better quantitative results.

In these two comparisons, there is no source code available; thus, we can only use the images from their paper. Besides, we do not have the ground truth light and camera settings for their captured image, while they do, therefore, we can only set up the light source according to their report to simulate their light condition as much as possible.

7.5 Ablation Study

Our ablation study, as depicted in Fig. 15, examines the three steps of our pipeline. First, Fig. 15 (b) highlights the need for our pre-estimation step to serve as better initialization. This step involves initializing the optimization process with a rough estimation of the pattern and color, as opposed to the random initialization used in general joint optimization. We find that sometimes joint optimization without pre-estimation can eventually approach an accurate appearance from a higher initial loss (orange line in the second row), but is significantly slower with the risk of getting trapped in local minima. Our pre-estimation method, although not always perfectly precise, ensures faster convergence and consistent alignment with the global minima.

Second, it examines the difference in optimization approaches. A separate optimization is commonly used in the previous work [SZK15; GHCG17; WCZ*19; JWH*22], which firstly analyzes patterns, and fixes them in the following optimization for the remaining parameters, like the reflectance component. This method

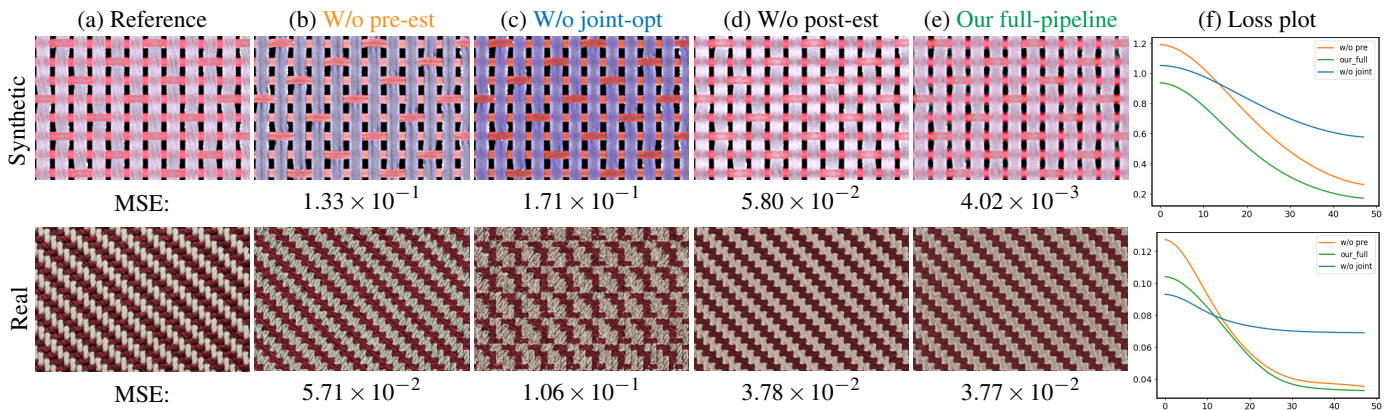


Figure 15: Ablation study of our pipeline; (b) refers to the results without pre-estimation described in § 5; (c) illustrates a naive combination strategy of prior works, first estimating pattern and then estimating appearance parameters with pattern fixed. (d) demonstrates the results without post-estimation described in § 6.3. Then, (e) shows the results of our full pipeline. (f) plots the loss of (b), (c), and (e) in corresponding colors. Note that the loss value corresponds to the images of the minimum pattern during optimization instead of the tiled results in (b)–(e).

often falls short, especially in terms of accurately aligning the weave pattern, as shown in the last row of Fig. 15 (c). The corresponding loss plot (blue line) shows that even a separately pre-estimated pattern provides a lower initial loss, but not jointly being optimized in the following step limits its optimal convergence.

Last, the post-estimation of noise provides more sense of reality, especially when recovering fuzzy woven fabric with a close-up capture as demonstrated in Fig. 8. Note that the post-estimation does not have a corresponding loss plot.

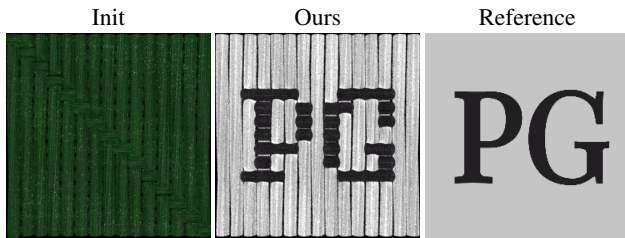


Figure 16: Matching a PG logo. The joint optimization starts from a default initialization. Even though the optimized pattern does not exist in the real world, as the warps without interlacing tension cannot hold, it still converged to match the reference.

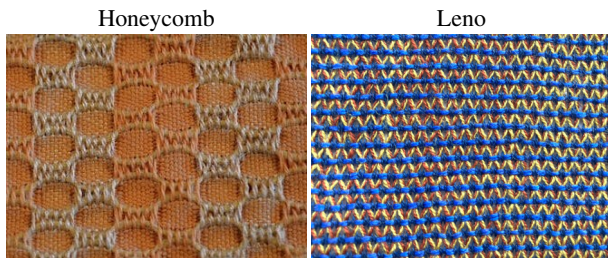


Figure 17: Examples of Honeycomb and Leno weave[§].

8 Conclusion and Discussions

In this study, we introduced an inverse rendering framework that enhances digital replication of woven fabrics. Our method estimates pattern, geometry, and appearance parameters from a single

close-up image, integrating discrete and continuous parameter optimization. The core of our approach lies in the explicit modeling of yarn geometries using B-spline curves and the ply and fiber details shaded by a yarn-based shading model. The joint optimization of discrete and continuous parameters in our system, executed through a combination of simulated annealing and differentiable rendering, represents a novel methodology in the area of inverse rendering. This framework has demonstrated its effectiveness in accurately replicating fabrics with various weave patterns and multiple plies, showcasing an improvement over existing methods.

However, our method still faces challenges with certain real-world samples. Specifically, our approach does not perform optimally when dealing with images that are extremely fuzzy due to clutter from flyaways or excessive irregularities, including sharp highlights, extensive occlusions, and severe capture distortion. Besides, our model assumes that the warps and wefts are grid-like straight, and does not perform optimally for irregular yarn layout, even though our results slightly perform better than manual determination, the PSNRs are still relatively low. In addition, some non-grid patterns where wefts are not orthogonal to warps, such as Honeycomb weave and Leno weave, will also require special rules to define the yarn layout as shown in Fig. 17. These could be areas for future exploration and improvement by a better curve optimization approach and a more advanced shading model.

Our current implementation of the forward shading model does not consider spatially varying parameters, like yarn radius, ply twist angle, and roughness, which could be an interesting direction to extend in the future. Additionally, as the forward shading model is based on the yarn level, it struggles with images captured from a far distance where the yarns are not explicitly visible. For slight irregularities, our post-estimation is limited in 2D space. In the future, we could extend it to 3D Gaussians to represent flyaway fibers.

[§] The left image is from <https://plainweave.net/2021/02/07/honeycomb-by-any-other-name-is-just-as-sweet/> and the right image is from <https://www.crafthub.eu/material/leno-weave-with-multicolor-wool/>.

Acknowledgement

Open Access funding enabled and organized by the University of Manchester Department of Computer Science.

References

- [ABN13] ASHA, V., BHAIJANTRI, NAGAPPA, and NAGABHUSHAN, P. “Periodicity Extraction using Superposition of Distance Matching Function and One-dimensional Haar Wavelet Transform”. (Nov. 2013) 2.
- [ANB12] ASHA, V., NAGABHUSHAN, P., and BHAIJANTRI, N.U. “Automatic extraction of texture-periodicity using superposition of distance matching functions and their forward differences”. *Pattern Recognition Letters* 33.5 (2012), 629–640. ISSN: 0167-8655. DOI: <https://doi.org/10.1016/j.patrec.2011.11.027>. URL: <https://www.sciencedirect.com/science/article/pii/S016786551100417X> 2, 5, 6.
- [CDI14] CHAUDHURY, KRISHNENDU, DIVERDI, STEPHEN, and IOFFE, SERGEY. “Auto-rectification of user photos”. *2014 IEEE International Conference on Image Processing (ICIP)*. 2014, 3479–3483. DOI: 10.1109/ICIP.2014.7025706 5.
- [CLA19] CASTILLO, CARLOS, LÓPEZ-MORENO, JORGE, and ALIAGA, CARLOS. “Recent Advances in Fabric Appearance Reproduction”. *Comput. Graph.* 84.C (Nov. 2019), 103–121. ISSN: 0097-8493. DOI: 10.1016/j.cag.2019.07.007. URL: <https://doi.org/10.1016/j.cag.2019.07.007> 3.
- [DAD*18] DESCHAMPTRE, VALENTIN, AITTALA, MIKA, DURAND, FREDO, et al. “Single-image SVBRDF capture with a rendering-aware deep network”. *ACM Trans. Graph.* 37.4 (July 2018). ISSN: 0730-0301. DOI: 10.1145/3197517.3201378. URL: <https://doi.org/10.1145/3197517.3201378> 1, 3.
- [GHCG17] GUARNERA, GIUSEPPE CLAUDIO, HALL, PETER, CHESNAIS, ALAIN, and GLENCROSS, MASHHADA. “Woven Fabric Model Creation from a Single Image”. *ACM Trans. Graph.* 36.5 (Oct. 2017). ISSN: 0730-0301. DOI: 10.1145/3132187. URL: <https://doi.org/10.1145/3132187> 1–3, 5, 8, 11.
- [Gri86] GRINBERG, ERIC L. “Radon transforms on higher Grassmannians”. *Journal of Differential Geometry* 24.1 (1986), 53–68. DOI: 10.4310/jdg/1214440257. URL: <https://doi.org/10.4310/jdg/1214440257> 5.
- [GSH*20] GUO, YU, SMITH, CAMERON, HAŠAN, MILOŠ, et al. “MaterialGAN: reflectance capture using a generative SVBRDF model”. *ACM Trans. Graph.* 39.6 (Nov. 2020). ISSN: 0730-0301. DOI: 10.1145/3414685.3417779. URL: <https://doi.org/10.1145/3414685.3417779> 1, 3.
- [HC18] HC, SULOCHAN. “Fabric Weave Pattern Detection Based on Fuzzy Clustering and Texture Orientation Features in Wavelet Domain”. *Journal of Textile Science & Engineering* 08 (Jan. 2018). DOI: 10.4172/2165-8064.1000383 2.
- [HDR19] HU, YIWEI, DORSEY, JULIE, and RUSHMEIER, HOLLY. “A novel framework for inverse procedural texture modeling”. *ACM Trans. Graph.* 38.6 (Nov. 2019). ISSN: 0730-0301. DOI: 10.1145/3355089.3356516. URL: <https://doi.org/10.1145/3355089.3356516> 3.
- [HK93] HANRAHAN, PAT and KRUEGER, WOLFGANG. “Reflection from Layered Surfaces due to Subsurface Scattering”. (1993) 4, 5.
- [IM12] IRAWAN, PITI and MARSCHNER, STEVE. “Specular reflection from woven cloth”. *ACM Transactions on Graphics (TOG)* 31.1 (2012), 1–20 2–4.
- [JMLH01] JENSEN, HENRIK WANN, MARSCHNER, STEPHEN R, LEVOY, MARC, and HANRAHAN, PAT. “A Practical Model for Subsurface Light Transport”. (2001) 4.
- [JSR*22] JAKOB, WENZEL, SPEIERER, SÉBASTIEN, ROUSSEL, NICOLAS, et al. *Mitsuba 3 renderer*. Version 3.1.1. <https://mitsuba-renderer.org>. 2022 8.
- [JWH*22] JIN, WENHUA, WANG, BEIBEI, HASAN, MILOS, et al. “Woven Fabric Capture from a Single Photo”. *SIGGRAPH Asia 2022 Conference Papers*. 2022, 1–8 1–3, 5, 8, 10, 11.
- [JXL15] JING, JUNFENG, XU, MENGMEI, and LI, PENGFEI. “Automatic recognition of weave pattern and repeat for yarn-dyed fabric based on KFCM and IDMF”. *Optik* 126.21 (2015), 2876–2883. ISSN: 0030-4026. DOI: <https://doi.org/10.1016/j.ijleo.2015.07.025>. URL: <https://www.sciencedirect.com/science/article/pii/S003040261500577X> 2, 5.
- [KBD07] KAUTZ, JAN, BOULOS, SOLOMON, and DURAND, FRÉDO. “Interactive editing and modeling of bidirectional texture functions”. *ACM SIGGRAPH 2007 Papers*. SIGGRAPH '07. San Diego, California: Association for Computing Machinery, 2007, 53–es. ISBN: 9781450378369. DOI: 10.1145/1275808.1276443. URL: <https://doi.org/10.1145/1275808.1276443> 1.
- [KGV83] KIRKPATRICK, SCOTT, GELATT, C. DANIEL, and VECCHI, MARIO P. “Optimization by Simulated Annealing”. *Science* 220.4598 (1983), 671–680 2, 6.
- [KSZ*15] KHUNGURN, PRAMOOK, SCHROEDER, DANIEL, ZHAO, SHUANG, et al. “Matching Real Fabrics with Micro-Appearance Models”. *ACM Trans. Graph.* 35.1 (2015), 1–13.
- [KZP*24] KHATTAR, APOORV, ZHU, JUNQI, PADOVANI, EMILIANO, et al. *A Multi-scale Yarn Appearance Model with Fiber Details*. 2024. arXiv: 2401.12724 [cs.GR] 1, 3, 4.
- [KZYM25] KHATTAR, APOORV, ZHU, JUNQI, YAN, LING-QI, and MONTAZERI, ZAHRA. “A Texture-Free Practical Model for Realistic Surface-Based Rendering of Woven Fabrics”. *Computer Graphics Forum* 44 (Feb. 2025). DOI: 10.1111/cgf.15283 3.
- [LDPT17] LI, XIAO, DONG, YUE, PEERS, PIETER, and TONG, XIN. “Modeling surface appearance from a single photograph using self-augmented convolutional neural networks”. *ACM Trans. Graph.* 36.4 (July 2017). ISSN: 0730-0301. DOI: 10.1145/3072959.3073641. URL: <https://doi.org/10.1145/3072959.3073641> 3.
- [MGJZ21] MONTAZERI, ZAHRA, GAMMELMARK, SOREN, JENSEN, HENRIK W, and ZHAO, SHUANG. “A Practical Ply-Based Appearance Modeling for Knitted Fabrics”. *arXiv preprint arXiv:2105.02475* (2021) 3.
- [MGJZ20] MONTAZERI, ZAHRA, GAMMELMARK, SØREN B, ZHAO, SHUANG, and JENSEN, HENRIK WANN. “A practical ply-based appearance model of woven fabrics”. *ACM Transactions on Graphics (TOG)* 39.6 (2020), 1–13 3, 4.
- [MJC*03] MARSCHNER, STEPHEN R, JENSEN, HENRIK WANN, CAMMARANO, MIKE, et al. “Light scattering from human hair fibers”. *ACM Transactions on Graphics (TOG)* 22.3 (2003), 780–791 3.
- [MPG*21] MENG, SHUO, PAN, RURU, GAO, WEIDONG, et al. “A multi-task and multi-scale convolutional neural network for automatic recognition of woven fabric pattern”. *J. Intell. Manuf.* 32.4 (Apr. 2021), 1147–1161. ISSN: 0956-5515. DOI: 10.1007/s10845-020-01607-9. URL: <https://doi.org/10.1007/s10845-020-01607-9> 5.
- [RSP*19] RODRIGUEZ-PARDO, CARLOS, SUJA, SERGIO, PASCUAL, DAVID, et al. “Automatic extraction and synthesis of regular repeatable patterns”. *Comput. Graph.* 83.C (Oct. 2019), 33–41. ISSN: 0097-8493. DOI: 10.1016/j.cag.2019.06.010. URL: <https://doi.org/10.1016/j.cag.2019.06.010> 2.
- [SBDJ13] SADEGHI, IMAN, BISKER, OLEG, DE DEKEN, JOACHIM, and JENSEN, HENRIK WANN. “A practical microcylinder appearance model for cloth rendering”. *ACM Transactions on Graphics (TOG)* 32.2 (2013), 1–12 3.
- [SM23] SOH, GUAN YU and MONTAZERI, ZAHRA. *Neural Yarn-Level Appearance Model for Cloth Rendering*. 2023. arXiv: 2311.04061 [cs.GR] 3.

- [SSM97] SCHÖLKOPF, BERNHARD, SMOLA, ALEXANDER, and MÜLLER, KLAUS-ROBERT. “Kernel principal component analysis”. *Artificial Neural Networks-ICANN’97*. Springer, 1997, 583–588. URL: <http://link.springer.com/chapter/10.1007/BFb00202176>.
- [Swi62] SWINEHART, D. F. “The Beer-Lambert Law”. *Journal of Chemical Education* 39 (1962), 333. URL: <https://api.semanticscholar.org/CorpusID:973797914>.
- [SZK15] SCHRODER, KAI, ZINKE, ARNO, and KLEIN, REINHARD. “Image-Based Reverse Engineering and Visual Prototyping of Woven Cloth”. *IEEE Transactions on Visualization and Computer Graphics* 21.2 (Feb. 2015), 188–200. ISSN: 1077-2626. DOI: [10.1109/TVCG.2014.2339831](https://doi.org/10.1109/TVCG.2014.2339831). URL: <https://doi.org/10.1109/TVCG.2014.2339831> 1, 2, 5, 8, 11.
- [TKM*24] TRUNZ, ELENA, KLEIN, JONATHAN, MÜLLER, JAN, et al. “Neural inverse procedural modeling of knitting yarns from images”. *Computers & Graphics* 118 (2024), 161–172. ISSN: 0097-8493. DOI: <https://doi.org/10.1016/j.cag.2023.12.013>. URL: <https://www.sciencedirect.com/science/article/pii/S00978493230030723>.
- [TLH*24] TANG, YINGJIE, LI, ZIXUAN, HASAN, MILOS, et al. “Woven Fabric Capture with a Reflection-Transmission Photo Pair”. *ACM SIGGRAPH 2024 Conference Papers*. SIGGRAPH ’24. Denver, CO, USA: Association for Computing Machinery, 2024. ISBN: 9798400705250. DOI: [10.1145/3641519.3657410](https://doi.org/10.1145/3641519.3657410). URL: <https://doi.org/10.1145/3641519.3657410> 1, 3, 10.
- [WCZ*19] WU, HONGYU, CHEN, XIAO-WU, ZHANG, CHEN-XU, et al. “Modeling yarn-level geometry from a single micro-image”. *Frontiers of Information Technology and Electronic Engineering* 20.9, 1165 (2019), 1165. DOI: [10.1631/FITEE.1800693](https://doi.org/10.1631/FITEE.1800693) 1–3, 5, 8, 10, 11.
- [WJHY22] WANG, BEIBEI, JIN, WENHUA, HASAN, MILOŠ, and YAN, LING-QI. “SpongeCake: A Layered Microflake Surface Appearance Model”. *ACM Transactions on Graphics* 42.1 (2022) 3.
- [XNZL14] XIAO, ZHITAO, NIE, XINXIN, ZHANG, FANG, and LEI, GENG. “Recognition for woven fabric pattern based on gradient histogram”. *Journal of the Textile Institute* 105 (Jan. 2014). DOI: [10.1080/00405000.2013.8475422](https://doi.org/10.1080/00405000.2013.8475422).
- [YJR17] YAN, LING-QI, JENSEN, HENRIK WANN, and RAMAMOORTHY, RAVI. “An Efficient and Practical Near and Far Field Fur Reflectance Model”. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017)* 36.4 (2017) 4.
- [ZJMB11] ZHAO, SHUANG, JAKOB, WENZEL, MARSCHNER, STEVE, and BALA, KAVITA. “Building volumetric appearance models of fabric using micro CT imaging”. *ACM Transactions on Graphics (TOG)* 30.4 (2011), 1–10 1.
- [ZJMB12] ZHAO, SHUANG, JAKOB, WENZEL, MARSCHNER, STEVE, and BALA, KAVITA. “Structure-aware synthesis for predictive woven fabric appearance”. *ACM Transactions on Graphics (TOG)* 31.4 (2012), 1–10 2.
- [ZLB16] ZHAO, SHUANG, LUAN, FUJUN, and BALA, KAVITA. “Fitting Procedural Yarn Models for Realistic Cloth Rendering”. *ACM Trans. Graph.* 35.4 (July 2016). ISSN: 0730-0301. DOI: [10.1145/2897824.2925932](https://doi.org/10.1145/2897824.2925932). URL: <https://doi.org/10.1145/2897824.2925932> 3.
- [ZMA*23] ZHU, JUNQIU, MONTAZERI, ZAHRA, AUBRY, JEAN-MARIE, et al. “A Practical and Hierarchical Yarn-based Shading Model for Cloth”. *Computer Graphics Forum* 42.4 (2023). Ed. by RITSCHEL TobiasWeidlich, ANDREA, 11 pages. DOI: [10.1111/cgfm.14894](https://doi.org/10.1111/cgfm.14894) 3.

A Explicit yarn geometry generation for our synthetic pattern

As spline curves define the yarn, we generate 3D control points of wefts and warps c_e, c_a to represent the smallest pattern, given a binary matrix P , the radius of weft and warp r_e, r_a , and the heights h at the crossover.

ALGORITHM 1: Explicit yarn geometry generation

Input: P, r_e, r_a, h
Output: c_e, c_a

```

1  $c_e, c_a \leftarrow []$ ;
2  $n_e, n_a \leftarrow P.shape$ ;
3  $y_e \leftarrow [-(n_e - 1) \cdot r_a, (n_e - 1) \cdot r_a]$ ;
4  $x'_e \leftarrow arange[0.5, n_e, 1]$ ;
5  $x_e \leftarrow hstack[flip(-x'_e) \cdot r_a, x'_e \cdot r_a]$ ;
6  $x_a \leftarrow [-(n_a - 1) \cdot r_e, (n_a - 1) \cdot r_e]$ ;
7  $y'_a \leftarrow arange[0.5, n_a, 1]$ ;
8  $y_a \leftarrow hstack[flip(-y'_a) \cdot r_e, y'_a \cdot r_e]$ ;
9 for  $i \leftarrow 0$  to  $n_e$  do
10    $c \leftarrow ones(2 \cdot n_a, 3)$ ;
11    $c[:, 0] \leftarrow x_e$ ;
12    $c[:, 1] \leftarrow y_e[i]$ ;
13    $c[:, 2] \leftarrow Gauss(P[i]) \cdot h$ ;
14    $c_e.append(c)$ ;
15 end
16 for  $i \leftarrow 0$  to  $n_a$  do
17    $c \leftarrow ones(2 \cdot n_e, 3)$ ;
18    $c[:, 0] \leftarrow x_a[i]$ ;
19    $c[:, 1] \leftarrow y_a$ ;
20    $c[:, 2] \leftarrow Gauss(1 - P^T[i]) \cdot h$ ;
21    $c_a.append(c)$ ;
22 end
23 return  $c_e, c_a$ ;

```

An example of generated control points for a 5×5 satin pattern is shown in Fig. 18.

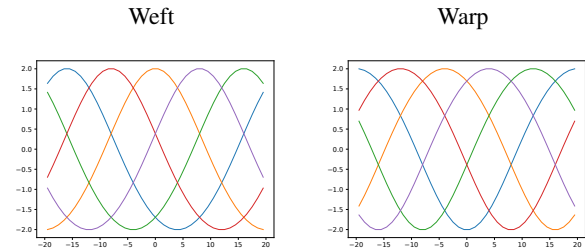


Figure 18: An example of generated control points for a 5×5 satin pattern.