**Computational Visual Media**
**https://doi.org/10.1007/s41095-0xx-xxxx-x**

**Research Article**

# A Multi-scale Yarn Appearance Model with Fiber Details

**Apoorv Khattar[1], Junqiu Zhu[2], Jean-Marie Aubry[3], Emiliano Padovani[3], Marc Droske[3], Ling-Qi Yan[2], Zahra Montazeri[1] (Corresponding Author)**



**Abstract** Rendering realistic cloth has always been a challenge due to its intricate structure. Cloth is made up of fibers, plies, and yarns, and previous curved-based models, while detailed, were computationally expensive and inflexible for large cloth. To address this, we propose a simplified approach. We introduce a geometric aggregation technique that reduces ray-tracing computation by using fewer curves, focusing only on yarn curves. Our model generates ply and fiber shapes implicitly, compensating for the lack of explicit geometry with a novel shadowing component. We also present a shading model that simplifies light interactions among fibers by categorizing them into four components, accurately capturing specular and scattered light in both forward and backward directions. To render large cloth efficiently, we propose a multi-scale solution based on pixel coverage. Our yarn shading model achieves 3-5 times faster rendering speed with less memory in near-field views compared to fiber-based models. Additionally, our multi-scale solution offers a 20% speed boost for distant cloth observation.

**Keywords** Cloth Rendering, Yarn Shading, Bidirectional Scattering Distribution Function (BSDF)

## 1 Introduction

Fabrics are an integral part of our everyday lives, serving a wide range of purposes from clothing to functional textiles such as curtains, furniture, and table sheets. Accurately modeling the appearance of cloth in a physically faithful manner has extensive applications in various fields, including design, online retail, and entertainment. However, they present a challenge due to their complex geometry and optics, with a hierarchical structure consisting of yarns, plies, and individual fibers.

There are mainly two predominant approaches for cloth rendering: surface-based and curve-based methods. Surface-based models represent cloth geometry using 2D sheets, often in the form of polygonal meshes [2–5]. While these models offer lightweight and editable representations suitable for macroscopic scale, they lack the fine-grained details required to convincingly render cloth in close-up views.

On the other hand, curve-based models focus on capturing intricate cloth details by representing the geometry at a microscale [6, 7]. These fiber-based models represent the structure of individual fibers and utilize volumetric or fiber-based light scattering models to simulate light interactions. While these approaches can achieve highly realistic renderings with exceptional fidelity, they suffer from computational expense, making them slow and challenging to manipulate.

To address these limitations, Montazeri et al. [1] introduced a ply-based approach that replaces the explicit representation of individual fibers with ply curves, implicitly incorporating fiber details. However, the number of plies in a yarn can vary (typically from 3 to 12) which makes modeling fabrics at the ply-level challenging and it still remains more computationally expensive due to multiple light bounces between plies. Thus, we desire an even more simplified model; we aim to represent the cloth solely with yarn curves. Building upon this ply-based approach, we pose the question: Can we completely conceal the hierarchical structure as well as the light transport within the yarns, treating the yarn as a whole aggregated model that embeds all sub-yarn geometries and light transport? A yarn-based model is notably more challenging compared to a ply-based model because simplifying the geometry yields to a more complicated appearance model. It requires modeling

1 The University of Manchester, Manchester, M13 9PL, United Kingdom. E-mail: A. Khattar apoorv.khattar@manchester.ac.uk; Z. Montazeri: zahra.montazeri@manchester.ac.uk (Corresponding Author)

2 University of California Santa Barbara, California, United States of America. Email: J. Zhu: zhujunqiu@mail.sdu.edu.cn; L. Yan: lingqi@cs.ucsb.edu

3 WETA FX, Wellington, New Zealand. Email: J.M. Aubry: jaubry@wetafx.co.nz; E. Padovani: emilianop@wetafx.co.nz; M. Droske: mdroske@wetafx.co.nz

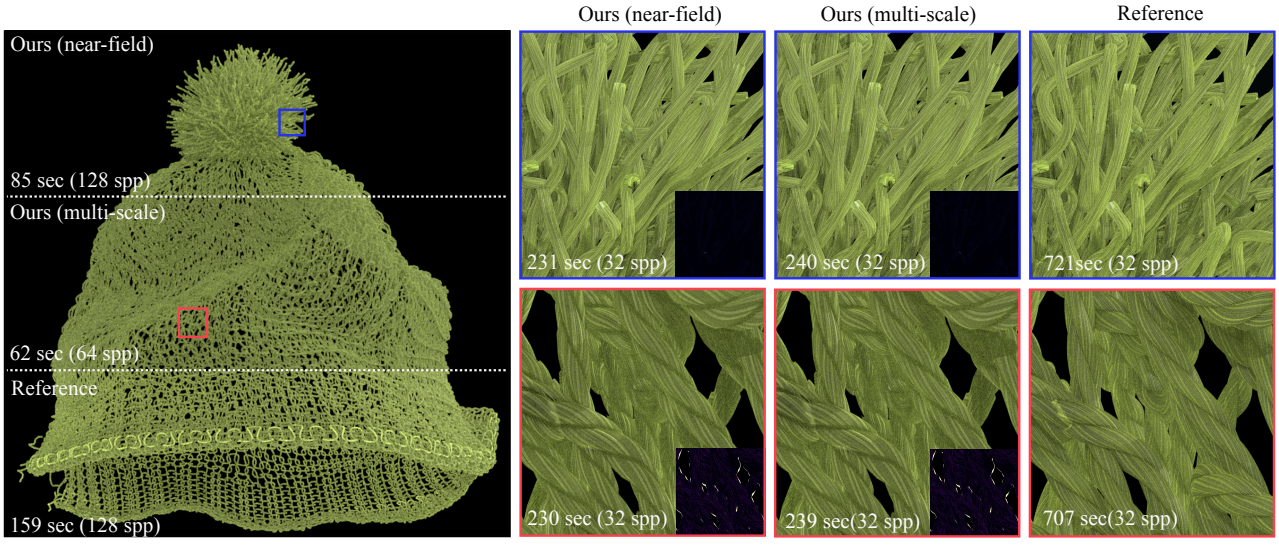



**Fig. 1** In this scene we compare the rendering results of our BYSDF model (near-field and multi-scale) to the reference model [1] for a knitted beanie, with the error map provided in the insets. Our model achieves close-matched results to the reference, both in far-away and zoomed-in views, with accurate soft shadows and geometry of plies and fibers as seen in Row I and Row II close-up, respectively. Notably, our approach achieves these results while utilizing only 20% of the memory and over 2.5 times acceleration compared to the reference under the same rendering quality (noise-level). Our multi-scale result offers additional performance gain by leveraging a level-of-detail strategy as shown in zoomed-out results. The scene has been lit using two area lights and one constant environment lighting.

a ray that interacts with a yarn, aggregating light interactions among plies as well as fibers, and ensuring both efficiency and accuracy without introducing obvious trade-offs.

In this paper, we make the following contributions:

**Accurate geometric representation with fiber details:** We propose using yarn curves as input geometry and dynamically computing ply and fiber details during rendering (Section 3). This approach allows us to achieve significant efficiency improvements while maintaining fiber-level details, outperforming previous models such as the ply-based model [1] and fiber-based ones [6].

**An aggregated yarn-based appearance representation:** We introduce a Bidirectional Yarn Scattering Distribution Function (BYSDF), a shading model that computes the appearance of the yarn as a whole using four components, following similar notations as Montazeri et al. [1] but now at the yarn-level representation (Section 4.1). These components include reflected and transmitted specular components that capture highlights bouncing off the surface or transmitted through the cloth without much scattering, as well as reflected and transmitted body components that approximate scattered light within the yarn medium. Additionally, we propose a novel consideration for self-shadowing to compensate for the lack of sub-yarn geometry (Section 4.1).

**Multi-scale solution:** Our model begins with a near-field solution to accurately depict the appearance of yarns and seamlessly transitions between near-field and far-field rendering using an efficient integration technique based on pixel coverage (Section 4.2). Inspired by fur and hair models [8, 9], our approach addresses the issue of resolving individual highlights within yarns when viewed from a distance, which traditionally required an inefficient ray sampling process. Our multi-scale rendering significantly reduces variance and enables faster far-view renderings with substantially fewer samples per pixel, while preserving the same level of quality as our near-field model.

**Significant speed-up and efficient memory usage:** By employing our aggregated yarn-based appearance model, on-the-fly geometry realization, and efficient integration for multi-scale rendering, we achieve equal quality with 3-5 times reduction in rendering time and memory usage compared to the fiber-based model ([6]) and 2 times improvement over the ply-based model ([1]). Furthermore, our multi-scale model delivers an additional 20% speed-up for far-away renderings, maintaining equal quality, as shown in Fig. 1. Consult the numbers about render times, memory usage and number of bounces presented in the figures and for a comprehensive comparison, refer to Table 1. We have made the code publicly available and submitted as a supplementary material to support future work.

## 2 Related Work

**Surface-based Models:** Early cloth rendering used 2D mesh surfaces with BRDFs [2–4] or BTFs [10] for fast

pipelines, but lacked faithful cloth geometry and fiber details essential for close-up and overall appearance in distance. These methods also overlooked light transmission, crucial for realism. Microfacet BRDF analysis by Ngan et al. [11] revealed limitations of BRDF-based approaches and specifically examined the anisotropic nature of velvet fibers. In contrast, our method provides fine details for near-field views and accurate light transmission, addressing these shortcomings.

**Volumetric Models:** Volumetric models in cloth rendering focus on capturing the geometry of individual fibers. Inspired by the pioneering work of Kajiya et al. [12], these models employ micro-imaging techniques like CT scans to obtain precise fiber geometry. The Radiative Transport Equation (RTE) is then used to simulate light interactions within the cloth.

RTE was extended by Jakob et al. [13] for anisotropic cloth by introducing direction-dependent scattering and attenuation to simulate light interactions through anisotropic media such as cloth. Further advancements were made proposing micro-appearance models [6, 7, 14] which relied on the anisotropic RTE and CT images for highly accurate renderings, considering fiber-level interactions. However, volumetric models are slower, memory-intensive compared to surface-based models, and challenging for non-static cases due to detailed fiber geometry and light interactions and the use of a practical resolution of volumetric grids. A procedural on-the-fly approach addressed the need for pre-storing curves [15] as well as fast renderings under restricted lighting [16]. Yet, acquiring data for volumetric representation techniques remains challenging. Our model offers comparable fidelity but is faster, treating the yarn as a whole and analytically aggregating the expensive fiber interactions.

**Curve-based Models:** Curve-based models depict cloth surfaces using fiber-modeled curves, utilizing Bi-directional Curve Scattering Distribution Functions (BCSDFs), originally introduced for hair strands by Marschner et al. [17]. This approach has gained popularity for simulating the appearance of fur, hair, and cloth. Subsequent research has improved it by considering scattering events [5, 9, 18].

An alternative model, proposed by Irawan et al. [4], is a yarn-based model, leveraging geometric information within yarns to represent cloth appearance. Jin et al. [19] propose a differentiable fabric built upon the work of [4] in which use a deep neural network to estimate the model parameters with a real photograph as reference. Fiber-based models [6], were later introduced to capture detailed cloth appearance considering the scattering of light off individual fibers and later extended exploring the correlation between mechanical simulations and fiber appearance [20]. However, these models can be inefficient due to the slow construction and traversal of the hierarchy of a large number of fibers and the increased computational complexity of simulating light interactions between fibers. They have been further enhanced to include real-time capabilities through core fiber aggregation [21] but this technique does not support ray tracing, so it cannot handle realistic lighting simulations. Lastly, a neural-based approach has been recently proposed [22] that also fails to capture the fiber details in close-up views.

To strike a balance, Montazeri et al. [1] proposed a ply-based model for woven cloth which was later extended to knitted cloth [23]. In this model, individual plies are represented as curves and 1D textures are used to add fiber-level details. It successfully achieves a detailed appearance, ensuring energy conservation during scattering events [5] without the need for a large volume hierarchy for fiber curves.

**Aggregated-based Models:** We argue that the ply-based model, as discussed in Section 1 inefficiently represents yarns, due to the use of explicit curves for individual plies. Yarns typically consist of 3 to 12 plies, with 5-ply yarns being the most common [14]. While offering high quality in near-field views, it exhibits high variance in far-field due to inherent micro-structure details. To address the ply dependency issue, a new yarn model was proposed that offers ply-level details [24]. However, apart from the lack of fiber details, this model can be problematic in both geometry and shading: wrong ply geometry can be produced, especially from grazing angle; Sharp silhouette of plies and significant color mismatch may appear due to incorrect transparency in a dual-scattering [25] style shading. In contrast, our multi-scale yarn-based model offers a comprehensive solution, effectively managing varied ply counts with fiber details without impacting performance and accuracy across different viewing distances (Section 6).

## 3 Modeling Yarn Geometry

**Preliminaries:** While the geometric representation of cloth can be complex, its hierarchical structure allows for faithful representation at different scales. Cloth is made up of long strands of *yarn*, consisting of multiple intertwined *plies* and hundreds of micro-diameter twisted *fibers*.

### 3.1 Overview

In our model, we focus on accurately representing cloth by explicitly generating yarn curves while implicitly computing ply and fiber details. This approach achieves an efficient yet accurate depiction, particularly well-suited for near-field viewing. Previous models either lack sub-yarn details [3],

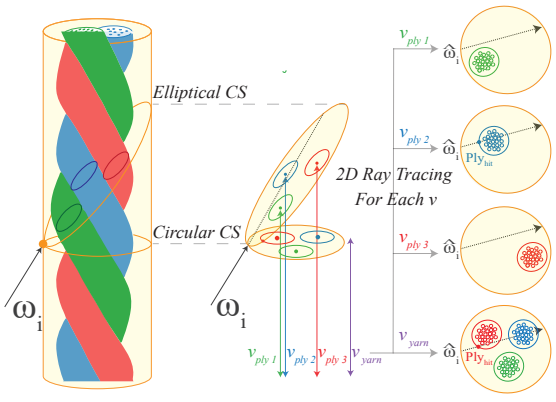



**Fig. 2 Implicit tracing in the elliptical yarn cross-section (CS) v/s circular CS.** The ray first intersects the yarn surface and cuts it into an ellipse. Our iterative approach (as discussed in Section 3.3) calculates the intersection of ply-helices with the elliptical plane and then returns the closest ply by employing 2D ray tracing with a different $v_{ply}$.

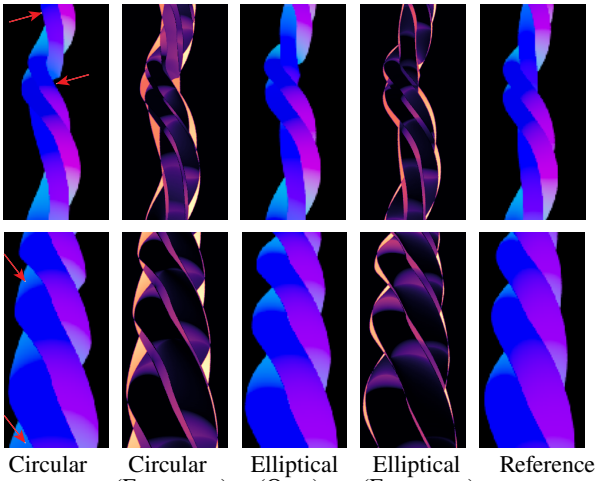



**Fig. 3 Implicit plies using elliptical cross-section (CS)** Circular cross-sections cause stretching or shortening in highly curved yarns (first row), leading to tangents differing significantly from the reference. In a slightly oblique view at 60 degrees (second row), the circular approximation introduces inaccuracies, causing an offset in overall shape and tangents compared to the reference, as marked with red arrows.

employ explicit fiber structures that are computationally intensive [6], or rely on explicit ply structures that are inefficient for multi-ply yarns [1, 23]. In what follows, we describe our geometric representation approach for cloth and the implicit ray tracing technique we employ to accurately compute the ray intersections with the cloth sample.

Our simplified geometric method takes a ray and a curved cylinder representing the yarn geometry as inputs. The outputs include the intersection point with the ply and the fiber's canonical frame (normal and tangent) at that point as if the fiber has been intersected explicitly. In case the ray doesn't find the intersection with the plies after hitting the yarn, a no-hit case is reported as an output.

To achieve the ply hit point, we propose an implicit ray tracing by focusing on the elliptical cross-section of the yarn instead of traditional circular cross-sections, as compared in Fig. 2. To this end, we assume plies and fibers as curved cylindrical helices. Thus, a perpendicular cut through the yarn results in a circular and an angled cut yields an elliptical cross-section, which consists of conceptual ply and fiber circles. By implicitly tracing the ray and analyzing light interactions within the elliptical cross-section formed by the ray, we can extract relevant information about the ply and fibers without the need for complex ray tracing computations involving explicit ply and fiber curves. It is important to note that our novel implicit ray tracing on the elliptical cross-section is more accurate than the traditional circular cross-section perpendicular to the yarn axis [24, 26] especially when viewed at grazing angles, as demonstrated in Fig. 3.

## 3.2 Explicit Yarns

We represent the yarns as B-Spline Curves. The hit point computed for a curved cylinder has a UV map for the yarn, where $u_{yarn}$ represents the rotated angle between the yarn normal and a reference yarn normal at the root, referred to as *azimuthal phase*, and $v_{yarn}$ represents the length of the hit point from the root of the curve, referred to as *longitudinal length*. Additionally, we generate reference normals which are a static directions determined at every control point of the yarn by creating rotation minimization frames [27], depicted as a double line in Fig. 4. This fixed direction along the yarn curve is needed as the reference to ensure the azimuthal angles are consistent through the animation.

## 3.3 Implicit Plies

Given the yarn hit point from Section 3.2, we employ a 2D ray tracing technique along the elliptical cross-section of the yarn cylinder to model the ply geometry. The previous works [24] always rely on the circular cross-section immediately at the yarn hit to compute the ply intersection. However, the distance between this cross-section using the projected incident ray $\hat{\omega}_i$ and the actual ply intersections with the ellipse plane formed along the incident ray $\omega_i$ is not negligible, especially at the grazing angle. Therefore, we utilize the more accurate elliptical cross-section strategy and visualize the comparison between elliptical and circular cross-sections by generating error maps of tangents for implicit plies (Fig. 3). Tangent

maps, based on the ply-centric model [1], illustrate the impact on yarn geometry when the incident ray is at an oblique angle, especially for curved yarns shown in the first row of Fig. 3.

The elliptical approach is challenging because as the ray travels through the yarn volume, the position of the plies also change, with each ply following a helix. We use a lightweight Newtonian method to find the intersection of the ply center line (assumed as a helix) with the ellipse plane using only 3-5 steps, iteratively. The initial step is the immediate cross-section at the yarn hit and we update the cross-section and closer to the final solution as illustrated in Fig. 2. Similar to the yarn scenario (Section 3.2), the intersected point on the ply is identified by the azimuthal phase $\mathrm{u}_{ply}$ and longitudinal length $\mathrm{v}_{ply}$. This is done for each ply separately until we find the intersection of the ply helices with the ellipse plane.

In tracing for each step, we propose a heuristic by projecting the incident ray $\hat{\omega}_\mathrm{i}$ and finding the ply center at that specific circular cross-section. Then we intersect the computed ply center with the ellipse plane using a line following the yarn tangent direction. At the new position with a different $\mathrm{v}_{yarn}$ we repeat the circular processing in the updated cross-section to find the ply center closer to the final solution. Once the ply center converges and no change in two consecutive iterations, the final solution of the helix and ellipse is returned as the hit-ply. This offers a fast tracing and please note the converged circular cross-section is different from the perpendicular circular cross-section at the yarn hit.

Once the appropriate positions of the ply centers are identified, we determine the ply closest to the yarn hit point. Once the appropriate ply hit is identified, we update its normal ($n_p = n(\mathrm{u}_{yarn})$) and tangent ($t_p = t(\mathrm{u}_{yarn})$) to incorporate the necessary ply-level geometry. At the hit point, the local ply tangent and ply normal are computed to form the ply geometry. The tangent vector is calculated using the first derivative of the 3D helix formula at the hit point, and the local ply normal is the vector pointing away from the ply center in the circular cross-section. If the ray does not intersect any plies, the ray is allowed to pass through the yarn cylinder.

### 3.4 Implicit Fibers

To model the fiber geometry, we adopt a precomputation-based approach first introduced by Montazeri et al. [1]. In this method, the ply cross-section is conceptualized as a collection of individual fiber centers, as illustrated in Fig. 4. Within this cross-section, we precompute fiber-level details, including normals and tangents, for the outermost fibers. These values are represented as a 1D texture map that wraps around the circumference of the ply and performs as a 1D height-map

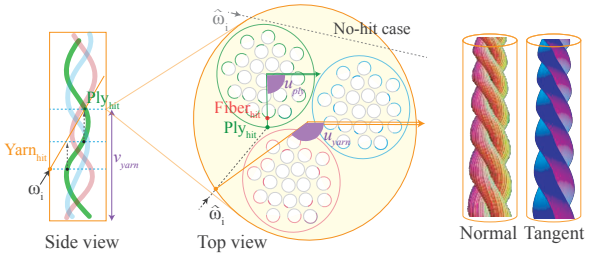
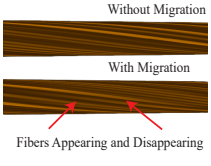



**Fig. 4 Schematic overview of our iterative approach to find the ellipse-helix intersection.** The ray intersects the yarn surface, forming an ellipse in the side view. A Newtonian approach iteratively finds the helix-ellipse intersection, determining the ply hit and ply center, obtaining azimuthal phase $\mathrm{u}_{ply}$ and longitudinal length $\mathrm{v}_{ply}$ which is used to add fiber texture from 1D texture maps.

to capture the visible fibers when viewed uniformly from the ply boundary.

While yarns typically consist of a few plies, each ply is usually composed of tens of fibers, resulting in a tightly packed representation. Therefore, the intersection point at the ply level serves as a reasonable approximation to the nearest fiber, and the necessary fiber-level details can be accessed directly from the texture without the need for additional implicit ray tracing. The resulting normal and tangent directions at the intersection points of the visible fibers are stored in a 1D texture that covers the outermost part of the cross-section. These precomputed values can be queried later using $\mathrm{u}_{ply}$. Finally, by combining the values obtained from Section 3.3 and rotating the ply frame further based on the fiber frame, the final normal and tangent in global space can be computed as, $n_f = n'(R(n_p))$ and $t_f = t'(R(t_p))$, respectively. where $n'$ and $t'$ are 1D perturbations from the precomputed textures, $R$ is the rotation of fibers.

**Adding fiber migration:** In Section 3.4, initially, fibers are assumed to follow a regular helical configuration around ply centerlines with a constant radius. However, in reality, fibers exhibit radius variations, known as migration, often characterized by sinusoidal functions [28] that sometimes move out and form a loop or disappear as get closer to the center. Our implicit ray tracing method, using exact 1D textures from the same fiber distribution in a circular cross-section, introduces an unwanted regular pattern lacking migration irregularities. To tackle these irregularities, we adopted an approach inspired by the ply-based model. Specifically, we periodically and randomly switch between different 1D textures, formed by distinct fiber alignments in the cross-section. Interpolating between these textures mimics fiber disappearance and



introduces irregularities, breaking the continuity.

## 4 Modeling Yarn Appearance

In this section, we discuss our approach to modeling the appearance of yarn, which complements the simplified geometry described in §3. Our yarn-based shading model extends the aggregated model first introduced in the ply-based model [1] to capture the appearance of a bundle of fibers.

### 4.1 BYSDF

The ply-based model provided a plausible shading model for individual plies, which consists of four lobes to capture specular and body components in both forward and backward directions. To adapt this model for yarns with implicit geometry (Section 3), three components require modification to accommodate interactions between plies. Only the immediate reflection component remains the same as the ply-based model.

Following the same notation from the literature, when an incident ray arrives at the surface of the yarn at point $\boldsymbol{x}$, it is divided into forward and backward portions. The backward component captures both the immediate reflection, which is part of the *specular* property ($f_{\boldsymbol{x}}^{\mathrm{S}}$), and the scattered light that exits the medium from the same side as the incident ray, referred to as the *body* property ($f_{\boldsymbol{x}}^{\mathrm{B}}$). Then, using the transmission component of $f_{\boldsymbol{x}}^{\mathrm{S}}$, we sample a point $\boldsymbol{y}$ on the yarn surface as the exit point, following the GGX distribution. Lastly, the forward component represents both the specularly transmitted light ($f_{\boldsymbol{y}}^{\mathrm{S}}$) and the scattered light ($f_{\boldsymbol{y}}^{\mathrm{B}}$) that continues in the forward direction. The illustration of our yarn-based appearance model is shown in Fig.5.

Assuming that we hit the yarn via a path starting from the camera at point $\boldsymbol{x}$. $f_{\boldsymbol{x}}$ is simply the average of two lobes to capture specular and body but $f_{\boldsymbol{y}}$ has to collect all contributions from the back side on different $\boldsymbol{y}$-s because the ray will refract towards different directions into a yarn. Therefore, it can be expressed as an integral with a kernel defined by $\rho$ over $\Omega_{\boldsymbol{y}}$ which is the part of surfaces that the refracted rays may cover at point $\boldsymbol{y}$. They can be formulated as follows, where $\hat{\boldsymbol{y}}$ is one refracted ray sampled following the GGX distribution:

$$f_{\boldsymbol{x}} = f_{\boldsymbol{x}}^{\mathrm{S}} + f_{\boldsymbol{x}}^{\mathrm{B}}, \quad f_{\boldsymbol{y}} = \int_{\Omega_{\boldsymbol{y}}} \left(f_{\hat{\boldsymbol{y}}}^{\mathrm{S}} + f_{\hat{\boldsymbol{y}}}^{\mathrm{B}}\right) \rho_y(\hat{\boldsymbol{y}})\, d\hat{\boldsymbol{y}} \tag{1}$$

**Specular components:** The specular components $f_{\boldsymbol{x}}^{\mathrm{S}}$ and $f_{\boldsymbol{y}}^{\mathrm{S}}$ represent the prominent highlights on the fabric surface when light reflects immediately or transmits through the fabric without being scattered. To model these specular lobes, we utilize a rough dielectric BSDF [29, 30] with a GGX distribution. The formulations for the specular components at points $\boldsymbol{x}$ and $\boldsymbol{y}$ are as follows, based on $C = \boldsymbol{\omega}_{\mathrm{i}} \cdot \boldsymbol{\omega}_{\mathrm{o}}$:

$$f_{\boldsymbol{x}}^{\mathrm{S}}(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}}) = k_{\boldsymbol{x}}^{\mathrm{S}} \cdot \tag{2}$$
$$\begin{cases} \frac{F_{\boldsymbol{x}}\, G_2(\boldsymbol{\omega}_{\mathrm{i}},\boldsymbol{\omega}_{\mathrm{o}},\boldsymbol{\omega}_{\mathrm{h}})\, D(\boldsymbol{\omega}_{\mathrm{h}};\boldsymbol{\beta}_{\boldsymbol{x}})}{4\,|\boldsymbol{\omega}_{\mathrm{i}}\cdot\boldsymbol{n}(\boldsymbol{x})|\,|\boldsymbol{\omega}_{\mathrm{o}}\cdot\boldsymbol{n}(\boldsymbol{x})|}, & (C>0) \\ \frac{|\boldsymbol{\omega}_{\mathrm{i}}\cdot\boldsymbol{\omega}_{\mathrm{t}}|\,|\boldsymbol{\omega}_{\mathrm{o}}\cdot\boldsymbol{\omega}_{\mathrm{t}}|}{|\boldsymbol{\omega}_{\mathrm{i}}\cdot\boldsymbol{n}(\boldsymbol{x})|\,|\boldsymbol{\omega}_{\mathrm{o}}\cdot\boldsymbol{n}(\boldsymbol{x})|} \frac{\eta^2\,(1-F_{\boldsymbol{x}})\, G_2(\boldsymbol{\omega}_{\mathrm{i}},\boldsymbol{\omega}_{\mathrm{o}},\boldsymbol{\omega}_{\mathrm{t}})\, D(\boldsymbol{\omega}_{\mathrm{t}};\boldsymbol{\beta}_{\boldsymbol{x}})}{[(\boldsymbol{\omega}_{\mathrm{i}}\cdot\boldsymbol{\omega}_{\mathrm{t}})+\eta\,(\boldsymbol{\omega}_{\mathrm{o}}\cdot\boldsymbol{\omega}_{\mathrm{t}})]^2}, & (C<0) \end{cases}$$

$$f_{\boldsymbol{y}}^{\mathrm{S}}(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}}) = k_{\boldsymbol{y}}^{\mathrm{S}} \cdot \tau(\boldsymbol{x}, \boldsymbol{y}, \sigma_{\mathrm{t}})^{N_i} \cdot \tag{3}$$
$$\begin{cases} 0, & (C>0) \\ \frac{|\boldsymbol{\omega}_{\mathrm{i}}\cdot\boldsymbol{\omega}_{\mathrm{t}}|\,|\boldsymbol{\omega}_{\mathrm{o}}\cdot\boldsymbol{\omega}_{\mathrm{t}}|}{|\boldsymbol{\omega}_{\mathrm{i}}\cdot\boldsymbol{n}(\boldsymbol{y})|\,|\boldsymbol{\omega}_{\mathrm{o}}\cdot\boldsymbol{n}(\boldsymbol{y})|} \frac{(1-F_{\boldsymbol{y}})\, G_2(\boldsymbol{\omega}_{\mathrm{i}},\boldsymbol{\omega}_{\mathrm{o}},\boldsymbol{\omega}_{\mathrm{t}})\, D(\boldsymbol{\omega}_{\mathrm{t}};\boldsymbol{\beta}_{\boldsymbol{y}})}{[\eta\,(\boldsymbol{\omega}_{\mathrm{i}}\cdot\boldsymbol{\omega}_{\mathrm{t}})+(\boldsymbol{\omega}_{\mathrm{o}}\cdot\boldsymbol{\omega}_{\mathrm{t}})]^2}, & (C<0) \end{cases}$$

where $\boldsymbol{n}(\boldsymbol{x})$ is the surface normal function, which is computed using the ply geometry at a given point $\boldsymbol{x}$ and similarly for $\boldsymbol{y}$. The normals are transformed based on the fiber alignments as described in Section 3.3. We denote this transformation as $\boldsymbol{n} = \boldsymbol{n}_f(\boldsymbol{n}_p)$. $\eta$ represents the refractive index, $F$ is the Fresnel reflection coefficient, $\boldsymbol{\omega}_{\mathrm{h}}$ and $\boldsymbol{\omega}_{\mathrm{t}}$ are the normalized half vectors for two different cases, and $G_2$ is the Smith uncorrelated masking-shadowing function. The term $D(\cdot;\boldsymbol{\beta})$ represents the normal distribution function (NDF) with $\boldsymbol{\beta} \in \mathbb{R}^2$ as the (anisotropic) roughness, and $k^{\mathrm{S}}$ is the specular albedo.

Notably, a key distinction from the ply-based model [1] is the updated attenuation term $\tau$, marked in gray. The term $\tau(\boldsymbol{x}, \boldsymbol{y}, \sigma_{\mathrm{t}})$ considers the attenuation of light between $\boldsymbol{x}$ and $\boldsymbol{y}$ using the Beer-Lambert law [31] where $\sigma_{\mathrm{t}}$ is the material's extinction coefficient:

$$\tau(\boldsymbol{x}, \boldsymbol{y}) := \exp(-\sigma_{\mathrm{t}}\, \|\boldsymbol{x} - \boldsymbol{y}\|). \tag{4}$$

In the ply-based model, explicit plies' appearance is studied, and ray tracing multiple scattering events between plies is computationally intensive, particularly for yarns with numerous plies. In contrast, our yarn-based model focuses on modeling the overall yarn appearance, inherently incorporating interactions between plies more efficiently. This is accomplished by computing a new $\tau$ based on $\boldsymbol{N}_i$, the number of intersected plies along the elliptical cross-section for

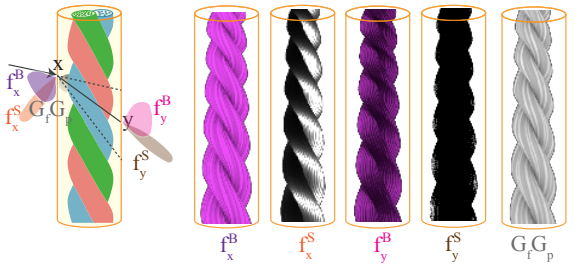



**Fig. 5 The components of BYSDF.** Illustration of the four lobes and their individual contributions as well as the shadowing component as a product of fiber and ply shadowing queried from the precomputed 1D textures.

the ray. Thus, as a ray passes through the yarn cylinders, we update the attenuation term as $\tau(\boldsymbol{x}, \boldsymbol{y}, \sigma_{\mathrm{t}})^{\boldsymbol{N}_i}$ to account for the attenuation by plies.

**Body components:** 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 [1, 9]. At point $\boldsymbol{y}$, the body component $f_{\boldsymbol{y}}^{\mathrm{B}}$ is represented by a Lambertian term. On the other hand, $f_{\boldsymbol{x}}^{\mathrm{B}}$ additionally considers a Lommel-Seeliger (LS) term [32, 33] as a mix, to ensure the energy conservation. The expressions for $f_{\boldsymbol{y}}^{\mathrm{B}}$ and $f_{\boldsymbol{x}}^{\mathrm{B}}$ are as follows:

$$f_{\boldsymbol{x}}^{\mathrm{B}}(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}}) = r\, B\, \mathrm{G}_{\mathrm{p}}(x, \omega_{\mathrm{i}})\, \mathrm{G}_{\mathrm{f}}(\boldsymbol{x})\, k_{\boldsymbol{x}}^{\mathrm{B}} \left[ LS(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}}) + \frac{1}{\pi} \right], \tag{5}$$

$$f_{\boldsymbol{y}}^{\mathrm{B}}(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}}) = (1 - r)\, B\, \mathrm{G}_{\mathrm{p}}(x, \omega_{\mathrm{i}})\, \mathrm{G}_{\mathrm{f}}(\boldsymbol{x})\, \frac{k_{\boldsymbol{y}}^{\mathrm{B}}}{\pi}, \tag{6}$$

where $k^{\mathrm{B}}$ is the body albedo and $r$ represents the reflected portion of the body energy $B$ and determined in correlation with the refracted albedo of $f_{\boldsymbol{x}}^{\mathrm{S}}$. The terms $\mathrm{G}_{\mathrm{p}}(., \boldsymbol{\omega}_{\mathrm{i}})$ and $\mathrm{G}_{\mathrm{f}}(.)$ correspond to the ply-shadowing and fiber-shadowing functions, respectively. These functions account for the occlusion and shadowing effects caused by the plies and fibers and are queried from 1D textures as described next.

**Self-Shadowing Components:** To address the limitations of existing yarn-based shading models [3, 4], we introduce an additional shadowing component that considers the occlusion caused by one ply on another ($\mathrm{G}_{\mathrm{p}}$). In the ply-based method, the ply shadows are handled by the path tracer, thus it only has the fiber shadowing function. In our approach using implicit plies, we approximate the inter-ply shadows since they are part of our yarn appearance. This component is computed by calculating the occlusion ratio at the outermost hit points on the ply surface and storing them in 1D a texture map. This texture map behaves like a horizon map that wraps around the yarn cross-section circumference. The ply-shadowing module $\mathrm{G}_{\mathrm{p}}$ is pre-computed based on the procedural model of the ply structure described in Section 3.3 in the absence of fibers, and it provides full coverage of the yarn surface by sweeping along the yarn centerline.

In addition to the ply-shadowing term, we also consider a fiber-shadowing term $\mathrm{G}_{\mathrm{f}}$ to account for self-shadowing caused by adjacent fibers as elaborated in Section 3.4. The multiplication of these two shadowing terms approximates the overall shadowing amount, compensating for the absence of explicit ply and fiber geometries. Although this shadowing component is computed in a 2D cross-section and does not capture the full 3D occlusion, our experiments have shown that it provides a reasonable estimation of the shadowed regions due to the regular procedural geometric representation of the yarn.

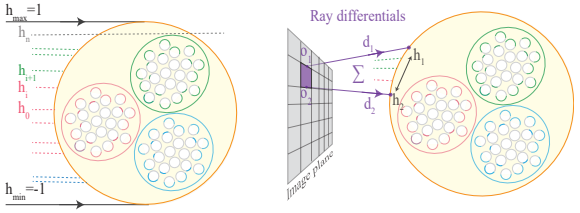




**Fig. 6 Multi-scale feature.** Illustration of a) our far-field integration when the entire azimuthal section of yarn is covered in one pixel and b) our multi-scale model that uses ray differentials to compute the pixel coverage (marked purple) on the surface of yarns and compute the integration accordingly to allow a smooth transition from near-field to far-field.

**Importance Sampling:** Importance sampling optimizes light interaction in rendering. To manage energy conservation, we define parameter ($B$), to normalize the energy ratio between specularity and body scattering of cloth. This parameter determines specularity energy, with the remaining energy allocated to the body component. The body component is further divided into reflection and transmission using another parameter ($r$), computed based on the probability of the Fresnel term. After selecting the lobe based on energy allocation, importance sampling employs appropriate distributions. The GGX distribution is used for specular lobes, ensuring plausible modeling of their behavior. For diffuse-like components, a cosine-weighted distribution facilitates importance sampling. Details about energy conservation testing can be found in Section 4.3.

## 4.2 Multi-scale Feature

The described shading model is efficient for close-up rendering, capturing fiber details in the near field. However, when the camera moves away, the pixel coverage increases, covering multiple fibers or the entire yarn, causing variations in normal distributions and necessitating supersampling to mitigate noise. To address this, we introduce a multi-scale model inspired by Yan et al. [9], adapting shading computation based on pixel coverage for improved efficiency.

For close-up views covering individual fibers, the original near-field model $f_{\mathrm{near}}$ (described in Section 4.1) is suitable, considering sub-yarn details. When the camera is distant, and the pixel footage spans the entire yarn, we employ a far-field model $f_{\mathrm{far}}$. In this case, we numerically integrate the BYSDF over the full azimuthal offset range from $h_{\max} = 1$ to $h_{\min} = -1$. This integration captures the overall yarn

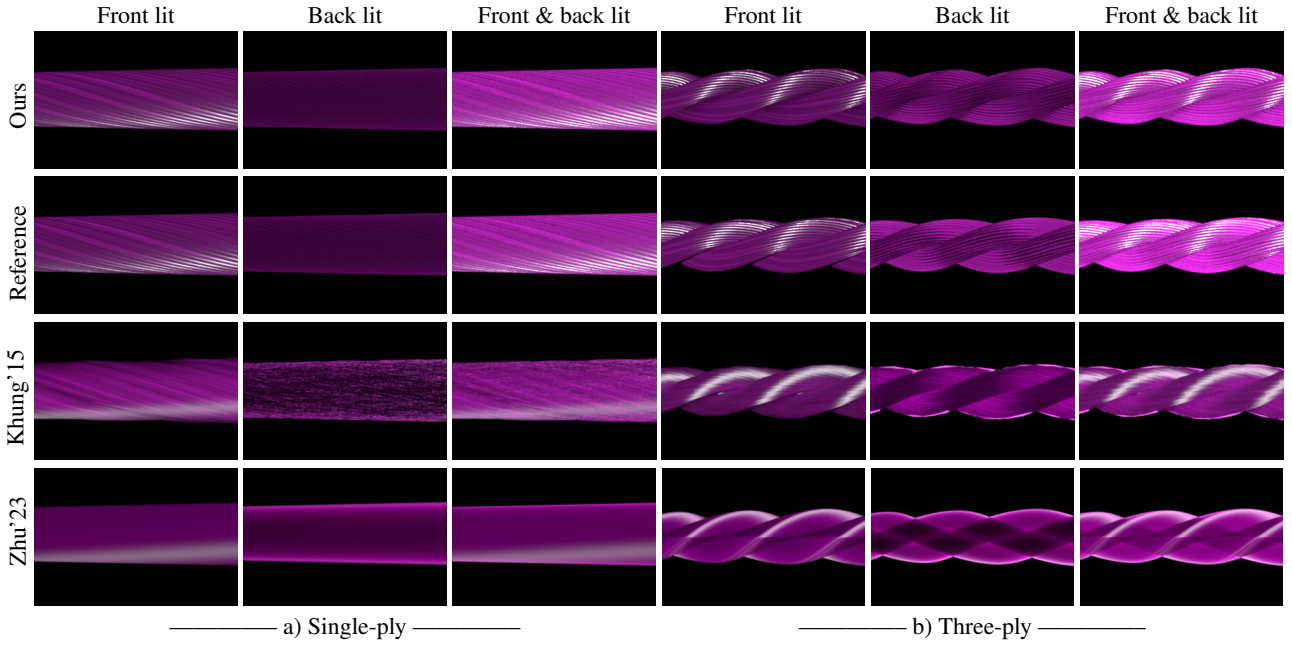



**Fig. 7** **Single yarn comparison** Our approach is compared with the reference ply-based model [1], fiber-based model [6] and recent yarn-based model [24] for a single yarn to observe the micro-scale events. The performance is listed in Table 1.

appearance in the far field, significantly reducing the required ray samples compared to the near-field model. By numerically solving the integrals instead of relying on the path-tracer, we allow more efficient rendering using fewer light samples.

As illustrated in Fig. 6, the computation of the far-field model involves stratified Monte Carlo sampling of the range of $h$ and performing the integration accordingly. This approach ensures that the far-field model accounts for the global appearance of the yarn, and can be written as follows with normalization factor $C = \int_{-1}^{1} W(h)\,\mathrm{d}h$,

$$f_{\mathrm{far}}(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}}) = \frac{1}{C}\int_{-1}^{1} W(h)\, f_{\mathrm{near}}(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}}, h)\,\mathrm{d}h. \quad (7)$$

To achieve a smooth transition between the near-field model ($f_{\mathrm{near}}$) and the far-field model ($f_{\mathrm{far}}$), we introduce a multi-scale BYSDF that adapts based on the pixel coverage. The computation of the multi-scale BYSDF relies on ray differentials $h_1$ and $h_2$, which define the azimuthal offset range covered by a specific pixel. The pixel coverage, represented by the range $(h_1, h_2) \subseteq (h_{\min}, h_{\max})$, determines the scale of the shading model. When the difference between $h_1$ and $h_2$ is small, indicating a close alignment, the multi-scale model performs similarly to the near-field model. These ray differentials are explained in detail in the Section 5 and the final formulation of our multi-scale appearance model combines the near-field and far-field components as follows:

$$f_{\mathrm{multi}}(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}}) = \frac{1}{\hat{C}}\int_{h_1}^{h_2} W(h)\, f_{\mathrm{near}}(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}}, h)\,\mathrm{d}h. \quad (8)$$

with $\hat{C}$ being the normalization over $[h_1, h_2]$. To approximate the multi-scale BYSDF ($f_{\mathrm{multi}}$), again we use a Monte Carlo algorithm that involves sampling discrete values of $h$ within the queried azimuthal range $h \in (h_1, h_2)$. These samples are selected based on their distance to the reference azimuthal offset $h_0$ and are used to numerically compute the integrated geometry (Section 3) and the shading model (Section 4).

Specifically, note that the methods we use to compute $f_{\mathrm{multi}}$ and $f_{\mathrm{far}}$ are still numerical. However, this approach is much more efficient than relying solely on a ray tracer with only the near-field model. We demonstrate this in Fig. 6.

To determine the aggregated geometry spanning this range, we compute the weighted average of the normal and tangent vectors of the visible fibers, which are precomputed as 1D texture maps. The weight ($W$) for each $h$ represents the distance to $h_0$, considering the contribution of the samples based on their distribution. The aggregated normal ($\boldsymbol{n}_{\mathrm{multi}}$) and tangent ($\boldsymbol{t}_{\mathrm{multi}}$) vectors are then used in equations (2) and (5) to compute $f_{\mathrm{multi}}$. The computation of $\boldsymbol{n}_{\mathrm{multi}}$ and $\boldsymbol{t}_{\mathrm{multi}}$ can be expressed as follows:

$$\boldsymbol{n}_{\mathrm{multi}} = \frac{1}{\hat{C}}\int_{h_1}^{h_2} W(h)\boldsymbol{n}(h)\mathrm{d}h, \quad (9)$$

$$\boldsymbol{t}_{\text{multi}} = \frac{1}{\hat{C}} \int_{h_1}^{h_2} W(h)\boldsymbol{t}(h)\mathrm{d}h, \tag{10}$$

Similarly, to find the aggregated shadowing component covering the queried azimuthal range, we use the weighted average of the fiber-shadowing terms queried from the pre-computed 1D texture map. Aggregated shadowed value $G_{\text{multi}}$ accounts for the energy loss in the original near-field model due to the presence of the microstructure. Ignoring this term would lead to unwanted brightness, similar to the known issue in microfacet materials. Additionally, the roughness variable of the specular lobe must be also adjusted. Aggregated roughness $\beta_{\text{multi}}$ is mapped linearly as a function of the azimuthal offset range to compensate for the absence of micro-geometries and the number of samples queried in that range. The updated $\beta_{\text{multi}}$ is experimentally given by, $\beta_{\text{multi}} = \beta + \frac{(h_2 - h_1)}{\hat{C}}$, which naturally contributes to a rougher appearance.

$$G_{\text{multi}} = \frac{1}{\hat{C}} \int_{h_1}^{h_2} W(h)\, \mathrm{G_f}(h)\, \mathrm{d}h. \tag{11}$$

Additional implementation details for the multi-scale feature are explained in Section 5.2.

### 4.3 Energy Conservation Test

Our model guarantees energy conservation because the sampling weight to select between the four components of our model as well as each component individually is normalized. In the theoretical case that the albedos were set to 1 and $\sigma_{\mathrm{t}}$ equals 0, the sum of $f_{\boldsymbol{x}}$ and $f_{\boldsymbol{y}}$ satisfies the criteria for energy conservation, effectively passing the white-furnace test, by construction. When we examine $f_{\boldsymbol{x}}^{\mathrm{S}} + f_{\boldsymbol{y}}^{\mathrm{S}}$, we observe that it is nearly energy conserving. There is a slight loss of energy due to (a) approximations of multiple scattering within $f_{\boldsymbol{x}}^{\mathrm{B}}$ and (b) the absence of internal reflections within $f_{\boldsymbol{y}}^{\mathrm{S}}$, which is set to 0, following the limitation of previous works. To ensure the entire BYSDF maintains energy conservation, the energy attributed to the missing internal reflection should be equal to $f_{\boldsymbol{x}}^{\mathrm{B}} + f_{\boldsymbol{y}}^{\mathrm{B}}$. Consequently, there exists a connection between the total scattering coefficient $\sigma_{\mathrm{t}}$ and the weight of the body components. If $\sigma_{\mathrm{t}}$ approaches 0, we can expect the body components to tend towards 0 as well, as they primarily result from internal volumetric scattering.

## 5 Implementation

Our multi-scale BYSDF is implemented as a custom path-tracer in Mitsuba 3 [34], which supports both environment and local lighting. To efficiently represent the yarn geometry, we utilize the B-Spline Curve shape module and compute the mapping of the ply and fiber level geometries on-the-fly.

### 5.1 Parameter Fitting

Our shading model requires a similar set of parameters to derive the realistic appearance compared to previous ply-based [1] and fiber-based [6] models. However, our decoupled approach, which separately handles reflection and transmission, provides an advantage in reproducing the target appearance under complex lighting configurations. This decoupling significantly enhances the practicality of BYSDF because the parameters can fit more easily. To achieve the best possible match with a target reference and avoid the mundane manual effort of parameter tweaking, we have developed an automatic parameter tweaking method using differentiable rendering. This method allows us to automatically adjust the parameters of our shading model to closely match the appearance of any target reference.

To perform automatic parameter tweaking, we treat the process as an inverse-rendering problem leveraging the differentiable setup in Mitsuba 3, which allows us to compute gradients with respect to the model parameters. Using these gradients, we iteratively adjust and fit the parameters to minimize the pixel-wise mean squared error (MSE) between our rendered image and the target reference. This optimization process ensures that our model captures the desired appearance as closely as possible. We optimize the parameters of the body components i.e. $r$, the diffuse color $k_{\boldsymbol{x}}^{\mathrm{B}}$ and $k_{\boldsymbol{y}}^{\mathrm{B}}$, the attenuation coefficient $\sigma_t$. Since optimization is sensitive to initial values, the initial value are set same as the ones to generate the ply-based reference. We do not optimize the parameters of the specular component, setting them same as the reference parameters since they are dependent on the geometry which is handled by the geometric component.

While many parameters can be adjusted independently, implicit relationships exist among parameters like B, $\mathrm{G_p}(\boldsymbol{x}, \boldsymbol{\omega}_{\mathrm{i}})$, $\mathrm{G_f}(\boldsymbol{x})$, etc. This provides artistic control but can lead to physically implausible choices. Inverse rendering helps find sensible parameters and implicit coupling can constrain optimization to more physically plausible solutions, reducing issues related to ill-posedness and non-uniqueness.

### 5.2 Multi-Scale Implementation

To enable the multi-scale feature for BYSDF, we made modifications to the ray-sampler in Mitsuba 3. Instead of generating standard rays, we generate samples of the RayDifferential structure. The RayDifferential consists of the actual ray, along with two offset rays used for pixel coverage measurement. These offset rays provide different perspectives and allow us to capture a range of azimuthal offsets for the multi-scale feature. It is important to note that classical differentials

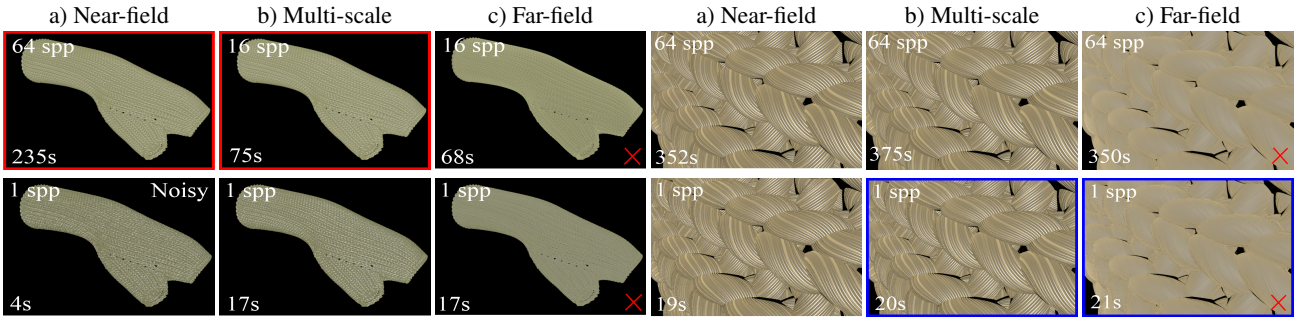



**Fig. 8** **Multi-scale comparison.** Near-field mode captures fine details but becomes inefficient as the camera moves away. In the Multi-scale mode, we smoothly transition between near-field and far-field renderings using an adaptive scheme based on pixel coverage, yielding fewer samples. Far-field mode efficiently captures macro-scale appearance when yarn width is smaller than pixel coverage. Comparing equal quality rendering (EQ) for zoom-out (left 3 columns), our multi-scale model is around 2.4× faster than the near-field model. Similarly, in the close-up (right 3 columns), our multi-scale model offers fiber details compared to the far-field model at a similar time. Note in the third column the yarns are still larger than one pixel so we should not use Far-field and this is for comparison only, hence marked by a red cross.

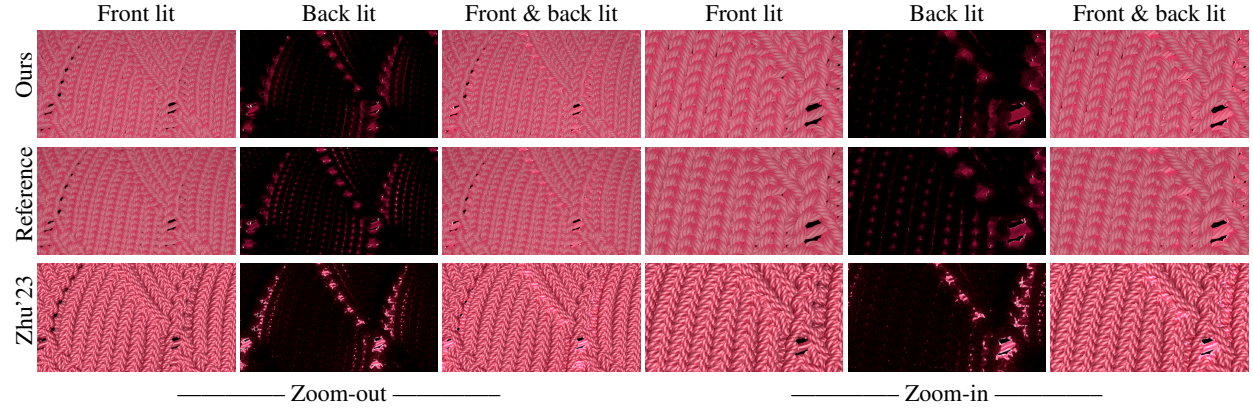



**Fig. 9** **Comparison with previous yarn-based model** Our model provides a close match to the reference ply-based model [1], offering faster rendering speed and reduced memory usage for this 6-ply knit sample. The performance metrics are listed in Table 1. Besides, in comparison to the yarn-based model [24], ours enjoys 1.2 times faster performance as well as presenting the fiber-level detail in close-up views. Please note since their model can not accurately handle the geometric information of the multi-ply yarns, the appearance looks more rigid and lacks soft shadows with inaccurate highlights.

correspond to the pixel footprint. In our case, we compute the intersected point when landing on a yarn and hence they might not be aligned with the azimuthal plane. To address this, we implemented custom ray differentials that ensure the range [$h_1$-$h_2$] corresponds to the actual azimuthal plane range within the pixel. This allows us to sample within this range adaptively based on the trajectory of pixels on the yarn geometries.

To determine the azimuthal offset range ($h_1$ and $h_2$), we utilize the ray-to-surface hit distance. By calculating the distances between the hit points after intersecting the ray differentials, we can define the range of azimuthal offset. $h_1$ corresponds to the distance between the origin hit point at $h_0$ and its offset point in one direction, while $h_2$ corresponds to the distance between $h_0$ and its offset point in the opposite direction. In this scenario, the range of $h_1$ and $h_2$ is dependent on the yarn radius, and for simplicity, we normalized them to ensure the range of (-1,1). These values use the origin $(o, o_1, o_2)$ and direction $(d, d_1, d_2)$ parameters as visualized in Fig. 6.

## 6 Results

In the following section, we present the rendering results obtained using our practical BYSDF and conduct a comparative analysis with existing approaches. The reference model in this paper is the ply-based model proposed by Montazeri et al. [1], which was evaluated against real photographs in the original paper. Consequently, we omit direct comparisons with actual photographs in this paper. We use the differentiable setup in Mitsuba 3 to match the set of parameters of our model against the reference. The details regarding parameter fitting can be found in the supplementary. We also compare the BYSDF with the model by Khungurn et al. [6] which to this end is

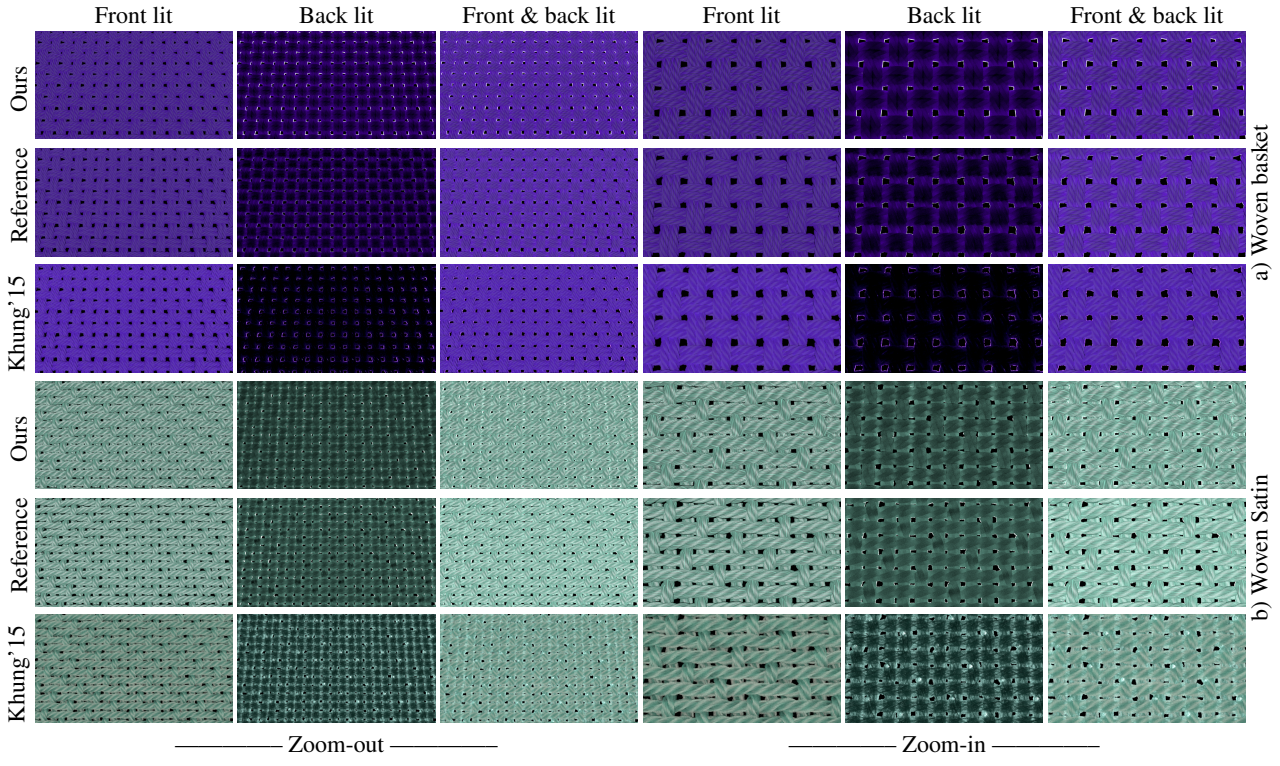



**Fig. 10 Comparison with previous fiber-based model** Our model provides a close match to the reference ply-based model [1], more efficiently as listed. The performance comparison can be found in Table 1. Please see the supplementary video for further comparisons. For far-away renderings, our multi-scale solution further enhances efficiency. In comparison, the fiber-based model [6] can achieve a relative match to the reference, but at a significantly higher computational cost. The woven samples are four- and five-plies yarns, respectively.

known as the fiber-based reference as well as Zhu et al. [24] as the recent yarn-based model. Our implementation of the geometric and appearance representation is integrated into the Mitsuba renderer [34] as a custom integrator and material plugin. All scenes shown in the paper were rendered using an AMD Ryzen 9 7950X 16-core Processor 4.50 GHz.

**Single yarns:** In the first set of results (Fig. 7), we compare our yarn-based method to the ply-based reference as well as the previous models under three different lighting configurations: front lighting only, back lighting only, and both front and back lighting. Our yarn-based model, similar to the reference, handles reflection and transmission separately, which allows it to capture the complex appearance of the yarn more accurately compared to the fiber-based model that considers all individual fiber interactions [6]. We observed that the fiber-based model can match the appearance of the reference model reasonably well under front lighting conditions, but it struggles to match the appearance when the lighting is changed to the back or when both front and back lighting are present, highlighting the limitations of its complex geometry representation. Furthermore, our model offers more accurate back-lighting due to the proper transmission sampling, while [24] showcases an unwanted sharp silhouette from the plies in the back visible due to a clear transparency. This artifact arises because the refracted ray in their approach is employed without any sampling or perturbation; it is directly utilized as transmission, resulting in this visual discrepancy. Finally, our model demonstrates improved efficiency compared to the reference ply-based model [1], especially when dealing with yarns that have multiple plies.

Additionally, we generate results using real fiber data provided Zhao et al [28] (Fig. 11). Notice that in the real data, the plies stretch and compress along the yarn (especially evident in 2-ply) while our method represents the plies implicitly using a fixed radius.

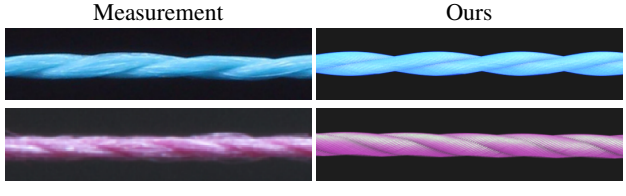



**Fig. 11 Comparison with real samples** Rayon 2-ply and Silk 3-ply taken from the data provided by [28]. Our model produces a close match in appearance to the measurement lit by an area and constant light.

**Table 1 Performance statistics. All rendering times are counted at equal quality (EQ).** We compare the time, storage memory required and number of bounces of Ours, Reference ([1]) Khungurn et al. (written as [6]) and Zhu et al. (written as [24]). In this table, #ply means the number of plies in a yarn. Our method on average is able to perform 1.7 times faster than the Reference, 1.3 times faster Zhu et al. and about 6 times faster than Khungurn et al. It is important to note that our method takes 200 kB more memory than Zhu et al. as we additionally store ply-level shadow maps and fiber-level shadow, local tangent and local normal maps for yarn cross-section.

| | | Time (min) | | | Memory (MB) | | | #Bounce | | |
|---|---|---|---|---|---|---|---|---|---|---|
| scene | #ply | Ours | Reference | [6]/[24] | Ours | Reference | [6]/[24] | Ours | Reference | [6]/[24] |
| Fig. 7.a | 1 | **0.1** | 0.1 | 2.5/0.2 | 0.06 | 0.06 | 17.3/**0.05** | **1** | 1 | 7.6/1 |
| Fig. 7.b | 3 | **0.1** | 0.3 | 2.2/0.2 | 0.06 | 0.18 | 31.5/**0.05** | **1** | 1.6 | 6.7/1 |
| Fig. 10 zoom-out | 6 | **7.4** | 12.2 | -/10.2 | 9.1 | 53.2 | -/**8.9** | **3.6** | 5.1 | -/3.6 |
| Fig. 10 zoom-in | 6 | **7.5** | 12.4 | -/10.3 | 9.1 | 53.2 | -/**8.9** | **3.7** | 5.3 | -/3.7 |
| Fig. 11.a zoom-out | 5 | **5.2** | 10.4 | 37.6/- | **73.0** | 364.5 | 1520/- | **1.9** | 4 | 24.0/- |
| Fig. 11.a zoom-in | 5 | **5.7** | 10.5 | 38.1/- | **73.0** | 364.5 | 1520/- | **1.7** | 4 | 22.7/- |
| Fig. 11.b zoom-out | 4 | **5.1** | 8.9 | 25.3/- | **14.4** | 57.4 | 719/- | **1.6** | 3.9 | 14.2/- |
| Fig. 11.b zoom-in | 4 | **5.8** | 9.1 | 24.9/- | **14.4** | 57.4 | 719/- | **1.6** | 3.8 | 15.6/- |

**Woven and knitted samples:** Furthermore, we conducted a comparison between our rendering results and the reference for woven and knit samples to demonstrate that our yarn-based approach is capable of accurately reproducing the appearance of fabrics regardless of the manufacturing structures on a macro scale and reasonably matching with the ply-based reference. Fig. 9 showcases the comparison between Our BYSDF and the recent yarn-based model [24]. While our model captures the transmission light accurately, their model has difficulty with back lighting because of the lack of sampling as explained in Fig. 7. Besides, the overall appearance of their result is hard and cannot reproduce the softness of the reference cloth. In Fig. 10, the fiber-based model reasonably matches the reference only if we enforced the parameters for each of the three setups separately. Our model, however, is only fitted for front- and back-lighting setups and automatically matches the front-and-back lighting scene with the reference due to the reflection and transmission parameters being decoupled. Our model boasts about 6-time performance gain due to the simplified geometry in comparison to the fiber-based model. The yarn geometries of all three samples are taken from the dataset by Leaf et al. [35] and the fiber curves are generated procedurally [28] and shaded accordingly [6]. Please refer to the Table 1 for a quantitative comparison.

**Multi-scale results:** In Fig. 8, we illustrate the three different modes of our model: a) In the near-field mode, the model is optimized for rendering when the camera is close to the fabric, and it accurately captures the fine details of the fibers and ply-level geometry. However, as the camera moves away, rendering using the near-field model becomes noisy and inefficient. b) To address this, we introduce the multi-scale mode, which offers a smooth transition between the near-field and far-field rendering. This model is able to adaptively match the appearance of far-away renderings using a significantly smaller number of samples per pixel (spp) compared to the near-field mode. c) In the far-field mode, the model is integrated over the entire yarn, capturing the macro-scale appearance of the fabric with no fiber detail. This mode is suitable when the camera is far away from the fabric and the pixel coverage spans more than the yarn width entirely. The geometry of the glove scene is taken from the yarn dataset by Yuksel et al. [36].

Please note that the far-field appearance in both cases is incorrect because, even when zoomed far away (left), a yarn is still much thicker than a pixel. As a result, the actual range of $h$ is much smaller than the range of [-1, 1] used for far-field rendering. Therefore, the far-field appearance should not be expected to match the other modes. However, the far-field mode still enables a significant reduction in the number of ray samples compared to the near-field mode. In essence, when the fabric is viewed from close-up (right), our multi-scale method should provide rendering performance comparable to the near-field mode. This is achieved by dynamically determining the number of ray samples proportional to the azimuthal offsets range ($h_2 - h_1$). On the other hand, when the fabric is viewed from far away, our method should be significantly more efficient than the near-field mode, as it allows for a substantial reduction in the number of required ray samples.

**Additional Result:** We rendered the sweater model, taken from the yarn dataset by [36] for varying geometry and appearance parameters under environment, area and point light.

We also rendered a simple lamp with satin and twill patterns under constant and point light to show the anisotropic appearance of fabrics. The twill pattern exhibits a round highlight with appearance whereas the satin exhibits a rectangular

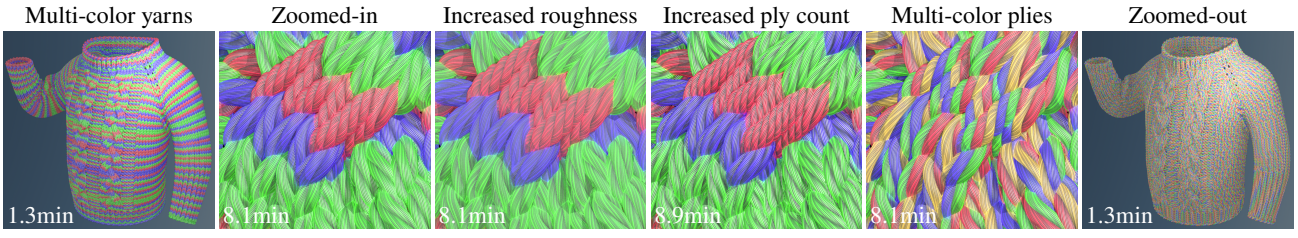



**Fig. 12 Additional results.** Sweater model taken from the yarn dataset by Yuksel et al. [36] shaded using BYSDF at 128 spp. In the first column, we show a zoomed-out image with multi-color four-ply yarns followed by the zoomed-in image. The third column displays higher roughness, while the fourth column depicts an increase in the number of plies to six. In the fifth column, we use different color for each ply followed by the zoomed-out image. Please refer to the supplementary video showcasing an animated multi-color sweater.

highlight as shown in previous works [1].

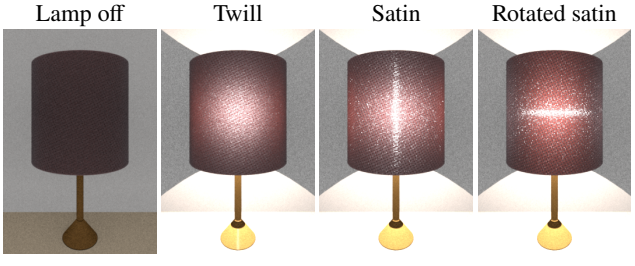



**Fig. 13 Additional results.** Lamp model for twill and satin pattern under constant and point light.

## 7 Discussion and Conclusion

**Limitations and Future Works:** Our model, while efficiently adding fiber texture using 1D texture maps, has room for improvement in seamlessly integrating fiber details directly into the yarn cross-section, eliminating the need for texture maps. Besides, the loss of energy due to the approximation of the multiple scattering follows the same limitation as previous works. Additionally, capturing the appearance of flyaways, a common feature in real fabrics, is a potential area for enhancement in our framework. For future work, we also aim to model fabrics like velvet, which presents unique challenges due to their protruding fiber bundles.

**Conclusion:** Our research introduces BYSDF, an effective yarn-based model that excels in rendering the intricate appearance of cloth, particularly in close-up views. It simplifies geometry, thereby reducing ray tracing costs, and avoids complex light transport between plies and fibers, resulting in fewer rays needed. The model successfully transitions from near-field to far-field renderings, integrating geometry and appearance seamlessly as the camera distance changes.

BYSDF builds upon the ply-based model, aggregating shading at the yarn level to capture the implicit geometries of ply and fiber. We compensate for the lack of explicit geometry with realistic shadowing effects, enhancing performance and fidelity. Our model achieves faster rendering speeds and lower memory usage for near-field renderings, and our multi-scale solution further optimizes far-away renderings.

## Acknowledgements

We would like to thank all the reviewers as well as Wētā FX rendering team for their valuable inputs. This research was partially funded by The University of Manchester's Dean's Award.

## Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

## References

[1] Montazeri Z, Gammelmark SB, Zhao S, Jensen HW. A practical ply-based appearance model of woven fabrics.*ACM Transactions on Graphics (TOG)*, 202039(6): 1–13.

[2] Adabala N, Magnenat-Thalmann N, Fei G. Real-time rendering of woven clothesIn *Proceedings of the ACM symposium on Virtual reality software and technology*200341–47.

[3] Sadeghi I, Bisker O, De Deken J, Jensen HW. A practical microcylinder appearance model for cloth rendering*ACM Transactions on Graphics (TOG)*, 201332(2): 1–12.

[4] Irawan P, Marschner S. Specular reflection from woven cloth*ACM Transactions on Graphics (TOG)*, 201231(1): 1–20.

[5] Zhu J, Zhao S, Wang L, Xu Y, Yan LQ. Practical level-of-detail aggregation of fur appearance*ACM Transactions on Graphics (TOG)*, 202241(4): 1–17.

[6] Khungurn P, Schroeder D, Zhao S, Bala K, Marschner S. Matching Real Fabrics with Micro-Appearance Models.*ACM Trans. Graph.*, 201535(1): 1–1.

[7] Zhao S, Jakob W, Marschner S, Bala K. Building volumetric appearance models of fabric using micro CT imaging*ACM Transactions on Graphics (TOG)*, 201130(4): 1–10.

[8] Chiang MJY, Bitterli B, Tappan C, Burley B. A Practical and Controllable Hair and Fur Model for Production Path TracingIn *ACM SIGGRAPH 2015 Talks*SIGGRAPH '15, New York,

NY, USA: Association for Computing Machinery20151doi: 10.1145/2775280.2792559.

[9] Yan LQ, Jensen HW, Ramamoorthi R. An efficient and practical near and far field fur reflectance model*ACM Transactions on Graphics (TOG)*, 201736(4): 1–13.

[10] Dana KJ, van Ginneken B, Nayar SK, Koenderink JJ. Reflectance and Texture of Real-World Surfaces*ACM Trans. Graph.*, 199918(1): 1–34doi:10.1145/300776.300778.

[11] Ngan A, Durand F, Matusik W. Experimental Analysis of BRDF Models.*Rendering Techniques*, 20052005(16th): 2.

[12] Kajiya JT, Kay TL. Rendering Fur with Three Dimensional TexturesIn *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques*SIGGRAPH '89, New York, NY, USA: Association for Computing Machinery1989271–280doi:10.1145/74333.74361.

[13] Jakob W, Arbree A, Moon JT, Bala K, Marschner S. A radiative transfer framework for rendering materials with anisotropic structureIn *ACM SIGGRAPH 2010 papers*, ACM New York, NY, USA20101–13.

[14] Zhao S, Jakob W, Marschner S, Bala K. Structure-aware synthesis for predictive woven fabric appearance*ACM Transactions on Graphics (TOG)*, 201231(4): 1–10.

[15] Luan F, Zhao S, Bala K. Fiber-Level On-the-Fly Procedural Textiles*Comput. Graph. Forum*, 201736(4): 123–135doi:10.1111/cgf.13230.

[16] Khungurn P, Wu R, Noeckel J, Marschner S, Bala K. Fast rendering of fabric micro-appearance models under directional and spherical gaussian lights*ACM Trans. Graph.*, 201736(6)doi: 10.1145/3130800.3130829.

[17] Marschner SR, Jensen HW, Cammarano M, Worley S, Hanrahan P. Light scattering from human hair fibers*ACM Transactions on Graphics (TOG)*, 200322(3): 780–791.

[18] Yan LQ, Tseng CW, Jensen HW, Ramamoorthi R. Physically-accurate fur reflectance: Modeling, measurement and rendering*ACM Transactions on Graphics (TOG)*, 201534(6): 1–13.

[19] Jin W, Wang B, Hasan M, Guo Y, Marschner S, Yan LQ. Woven Fabric Capture from a Single PhotoIn *SIGGRAPH Asia 2022 Conference Papers*20221–8.

[20] Montazeri Z, Xiao C, Fei Y, Zheng C, Zhao S. Mechanics-Aware Modeling of Cloth Appearance*IEEE Transactions on Visualization and Computer Graphics*, 2021: 137 – 150doi: 10.1109/TVCG.2019.2937301.

[21] Wu K, Yuksel C. Real-Time Fiber-Level Cloth RenderingIn *Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*I3D '17, New York, NY, USA: Association for Computing Machinery20171–8doi:10.1145/3023368.3023372.

[22] Soh GY, Montazeri Z. Neural Yarn-Level Appearance Model for Cloth Rendering, 2023.

[23] Montazeri Z, Gammelmark S, Jensen HW, Zhao S. A Practical Ply-Based Appearance Modeling for Knitted Fabrics*arXiv preprint arXiv:2105.02475*, 2021.

[24] Zhu J, Montazeri Z, Aubry JM, Yan LQ, Weidlich A. A Practical and Hierarchical Yarn-based Shading Model for Cloth. *Computer Graphics Forum*, 202342(4): 11 pagesdoi: 10.1111/cgf.14894.

[25] Zinke A, Yuksel C, Weber A, Keyser J. Dual scattering approximation for fast multiple scattering in hairIn *ACM SIGGRAPH 2008 Papers*SIGGRAPH '08, New York, NY, USA: Association for Computing Machinery20081–10doi: 10.1145/1399504.1360631.

[26] Yan LQ, Sun W, Jensen HW, Ramamoorthi R. A BSSRDF Model for Efficient Rendering of Fur with Global Illumination*ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2017)*, 201736(6).

[27] Wang W, Jüttler B, Zheng D, Liu Y. Computation of rotation minimizing frames*ACM Transactions on Graphics (TOG)*, 200827(1): 1–18.

[28] Zhao S, Luan F, Bala K. Fitting Procedural Yarn Models for Realistic Cloth Rendering*ACM Trans. Graph.*, 201635(4)doi: 10.1145/2897824.2925932.

[29] Heitz E. Sampling the GGX Distribution of Visible Normals*Journal of Computer Graphics Techniques (JCGT)*, 20187(4): 1–13.

[30] Walter B, Marschner SR, Li H, Torrance KE. Microfacet Models for Refraction through Rough SurfacesIn *Proceedings of the 18th Eurographics Conference on Rendering Techniques*EGSR'07, Goslar, DEU: Eurographics Association2007195–206.

[31] Swinehart DF. The Beer-Lambert Law*Journal of Chemical Education*, 196239(7): 333doi:10.1021/ed039p333.

[32] Hanrahan P, Krueger W. Reflection from Layered Surfaces Due to Subsurface ScatteringIn *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*SIGGRAPH '93, New York, NY, USA: Association for Computing Machinery1993165–174doi:10.1145/166117.166139.

[33] Jensen HW, Marschner SR, Levoy M, Hanrahan P. A Practical Model for Subsurface Light TransportIn *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*SIGGRAPH '01, New York, NY, USA: Association for Computing Machinery2001511–518doi:10.1145/383259.383319.

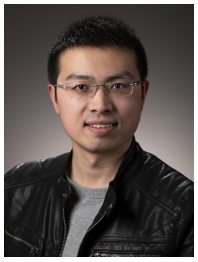
[34] Jakob W, Speierer S, Roussel N, Nimier-David M, Vicini D, Zeltner T, Nicolet B, Crespo M, Leroy V, Zhang Z. Mitsuba 3 renderer, 2022Https://mitsuba-renderer.org.

[35] Leaf J, Wu R, Schweickart E, James DL, Marschner S. Interactive Design of Yarn-Level Cloth Patterns*ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2018)*, 201837(6)doi:https://doi.org/10.1145/3272127.3275105.

[36] Yuksel C. Yarn-level Cloth Modelshttp://www.cemyuksel.com/research/yarnmodels, 2020.

## Author biography

**Apoorv Khattar** is currently pursuing his PhD at the University of Manchester under the supervision of Dr. Zahra Montazeri and is the recipient of the Dean's Doctoral Scholarship Award. His research interest is realistic cloth appearance modelling and is admitted for a research internship at Weta Digital in 2024. He holds a bachelor's degree from Indraprastha Institute of Information Technology (IIIT), Delhi in 2020.



**Ling-Qi Yan** is an Assistant Professor of Computer Science at UC Santa Barbara, a co-director of the MIRAGE Lab, and affiliated faculty in the Four Eyes Lab. Lingqi strives to achieve photo-realistic rendering. His research focuses on physically-based rendering in both micro and macro ends, encompassing areas such as appearance modeling and representation, physical light transport theory, and real-time ray tracing practices. Lingqi's outstanding contributions have been recognized through various accolades, including the SIGGRAPH 2019 Outstanding Doctoral Dissertation Award, a SIGGRAPH 2022 Best Paper Honorable Mention and an EGSR 2023 Best Paper Award.

**Zahra Montazeri** is a lecturer (Assistant Professor) at the University of Manchester in the Department of Computer Science. Her field of research is rendering and appearance modeling for complex materials such as cloth. She worked as a research consultant at Disney Research and received Star Wars movie credit for The Mandalorian and her research paper was used in the production of Avatar, The Way of Water. Before joining academia, she was a R&D at Industrial Light & Magic (ILM) and interned at Pixar Animation Studios, DreamWorks Animation and Luxion. She holds a PhD from University of California, Irvine, and her B.Sc. from the Sharif University of Technology in 2015, Iran.

## Electronic Supplementary Material

We have provided a video showing our results. The video comprises of the following scenes:

(1) 5-ply Woven Basket comparison with reference with directional lighting and camera transition between close-up and far-view.
(2) 1-ply Glove comparison with reference with a moving light.
(3) 1-ply Glove comparison with reference with a rotating geometry.
(4) 1-ply Glove multi-scale comparison with reference with directional lighting and camera transition between close-up and far-view.
(5) 3-ply multi-color sweater with environment, area and sharp point lighting and camera transition between close-up and far-view.

We have also made our code publicly available which can be found here: https://github.com/apoorvkhattar/yarn_model_mitsuba3/tree/master.