A Multi-scale Yarn Appearance Model with Fiber Details

Citation for published version (APA):

Citing this paper
Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights
Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy
If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [http://man.ac.uk/04Y6Bo] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.

Download date: 23. Jun. 2024
A Multi-scale Yarn Appearance Model with Fiber Details

APOORV KHATTAR, University of Manchester, United Kingdom
JUNQUI ZHU, University of California, Santa Barbara, UK
EMILIANO PADOVANI, Weta FX, NZ
JEAN-MARIE AURBY, Weta FX, NZ
MARC DROSKE, Weta FX, NZ
LING-QI YAN, University of California, Santa Barbara, USA
ZAHRA MONTAZERI, University of Manchester, USA

Fig. 1. In this scene we compare the rendering results of our BYSDF model (near-field and multi-scale) to the reference model [Montazeri et al. 2020] for a green knitted beanie. Our model achieves close-matched results to the reference, both in far-away and zoomed-in views. 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, most notably in zoomed-out results. The scene has been lit using two area lights and one constant environment lighting.

Rendering realistic cloth has always been a challenge due to its intricate structure. Cloth is made up of fibers, plies, and yarns, and previous curve-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 outperforms previous methods, achieving rendering speeds 3-5 times faster with less memory in near-field views. Additionally, our multi-scale solution offers a 20% speed boost for distant cloth observation.

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.

In the realm of computer graphics, researchers have explored predominantly two primary 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 [Adabala et al. 2003; Irawan and Marschner 2012; Sadeghi et al. 2013; Zhu et al. 2022]. 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 [Khungurn et al. 2015; Zhao et al. 2011]. These models delve into 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. [2020] introduced a novel approach that replaces the explicit representation of individual fibers with ply curves, implicitly incorporating fiber details. However, 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? This question presents a notably higher level of difficulty compared to the ply-based model. It requires simplifying the geometry when a ray 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 (§3). This approach allows us to achieve significant efficiency improvements while maintaining the same level of fidelity, surpassing previous models such as the ply-based model [Montazeri et al. 2020] and fiber-based ones [Khungurn et al. 2015].

**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. [2020] but now at the yarn-level representation (§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 (§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 (§4.2). Inspired by fur and hair models [Chiang et al. 2015; Yan et al. 2017a], 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 a remarkable 3-5 times reduction in rendering time and memory usage. This is compared to the work of Montazeri et al. [2020]—referred to as the ply-based model throughout the paper—while generating results of equal quality in close-up scenarios. Furthermore, our multi-scale model delivers an additional 20% speed-up for far-away renderings, maintaining equal quality. For a comprehensive comparison, please refer to Tab. ??.

### 2 RELATED WORK

#### Surface-based Models

Early cloth rendering used 2D mesh surfaces with BRDFs [Adabala et al. 2003; Irawan and Marschner 2012; Sadeghi et al. 2013] or BTFs [Dana et al. 1999] 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. [2005] 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 and Kay [1989], 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. [2010] 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 [Khungurn et al. 2015; Zhao et al. 2011, 2012] 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 [Luan et al. 2017]. 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. 2003]. This approach has gained popularity for simulating the appearance of fur, hair, and cloth. Subsequent research has improved it by considering scattering events [Yan et al. 2017a, 2015; Zhu et al. 2022].

An alternative model, proposed by Irawan and Marschner [2012], is a yarn-based model, leveraging geometric information within yarns to represent cloth appearance. Fiber-based models [Khungurn et al. 2015], were later introduced to capture detailed cloth appearance considering the scattering of light off individual fibers. They have been further enhanced to include real-time capabilities through core fiber aggregation [Wu and Yuksel 2017] and exploring the correlation between mechanical simulations and fiber appearance [Montazeri et al. 2021b]. 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.

To strike a balance, Montazeri et al. [2020] proposed a ply-based model for woven cloth which was later extended to knitted cloth [Montazeri et al. 2021a]. 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 [Zhu et al. 2022] without the need for a large volume hierarchy for fiber curves.

In Sec. 1, we contend that the ply-based model inadequately represents yarns, focusing on individual plies as explicit curves. Yarns typically consist of 3 to 12 plies, with 5-ply yarns being the most common [Zhao et al. 2012]. While offering high quality in near-field views, it exhibits high variance in far-field due to inherent microstructure details. To address the ply dependency issue, a new yarn model was proposed that offers ply-level details [Zhu et al. 2023]. 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 [Yuksel 2020] 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 (§5).

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 [Sadeghi et al. 2013], employ explicit fiber structures that are computationally intensive [Khungurn et al. 2015], or rely on explicit ply structures that are inefficient for multi-ply yarns [Montazeri et al. 2021a, 2020]. 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. 4. 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 [Yan et al. 2017b; Zhu et al. 2023] especially when viewed at grazing angles, as demonstrated in Fig. 3.

3.2 Explicit Yarns

To determine the intersection between the ray and the yarn geometry, we utilize the B-Spline Curve shape module in Mitsuba 3 [Jakob et al. 2022]. The hit point computed for a curved cylinder has a UV map for the yarn, where \( u_{\text{yarn}} \) represents the rotated angle between the yarn normal and a reference yarn normal at the root, referred to as azimuthal phase, and \( v_{\text{yarn}} \) represents the length of the hit point from the root of the curve, referred to as longitudinal length. Additionally, we create a module for generating reference normals which are a static directions determined at every control point of the yarn by creating rotation minimization frames [Wang et al. 2008], depicted as a double line in Fig. 2. This fixed direction along the yarn curve is needed as the reference point to determine the azimuthal angles consistent through the animation.

Fig. 2. 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 angle \( \omega_i \) and longitudinal length \( v_{\text{ply}} \) which is used to fiber texture from 1D texture maps.

Fig. 3. Implicit Plies using Elliptical Cross-section (CS) Circular cross-section 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 highlighted in red boxes.
3.3 Implicit Plies

Given the yarn hit point from §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 [Zhu et al. 2023] 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 ω for the actual ply intersections with the ellipse plane formed along the incident ray ω 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. Tangent maps, based on the ply-centric model [Montazeri et al. 2020], illustrate the impact on yarn geometry when the incident ray is at an oblique angle and when the yarn is highly curved in Fig. 3.

The elliptical approach is challenging because as the ray advances through the yarn volume, the plies also change in position each 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. 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. 4. Similar to the yarn scenario (§3.2), the intersected point on the ply is identified by the azimuthal phase \( \pi \) and longitudinal length \( v_{\text{ply}} \). 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 smart trick by projecting the incident ray \( \omega_i \) 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 \( v_{\text{yarn}} \) 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 \( (n_p = n(u_{\text{yarn}})) \) and tangent \( (t_p = t(u_{\text{yarn}})) \) 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. [2020]. In this method, the ply cross-section is conceptualized as a collection of individual fiber centers, as illustrated in Fig. 2. 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 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 \( v_{\text{ply}} \).

Finally, by combining the values obtained from §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 §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 [Zhao et al. 2016]. 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 address this, we periodically and randomly switch between different 1D textures, formed by different 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 [Montazeri et al. 2020] 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 (§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 \( x \), it is divided into forward and backward portions. The backward component captures both the immediate reflection, which is part of the specular property \((f^B_x)\) (the first case in Eqn. 2), and the scattered light that exits the medium from the same side as the incident ray, referred to as the body property \((f^B_y)\). Then, using the transmission component of \( f^S_x \) (the second case in Eqn. 2), 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 \((f^S_y)\) and the scattered light \((f^B_y)\) 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 \( x \); \( f_x \) is simply the average of two lobes to capture specular and body but \( f_y \) has to collect all contributions from the back side on different \( y \)-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 \( \Omega \) which is the part of surfaces that the refracted rays may cover. They can be formulated as follows:

\[
f_x = f^S_x + f^B_x, \quad f_y = \int_{\Omega_y} \left( f^S_y + f^B_y \right) \rho_y(y) \, dy
\]  

(1)

**Specular components:** The specular components \( f^S_x \) and \( f^S_y \) 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 [Heitz 2018; Walter et al. 2007] with a GGX distribution. The formulations for the specular components at points \( x \) and \( y \) are as follows:

\[
f^S_x(\omega_x, \omega_o) = k_x^S \cdot G(x, y, \sigma_i)^N_i.
\]

\[
\begin{cases}
\frac{1}{4} G_x(\omega_x, \omega_o, \omega_n) D(\omega_x, \beta_x) & (\omega_x \cdot \omega_o > 0), \\
[\omega_o \cdot n(x)] [\omega_o \cdot n(x)] & (\omega_x \cdot \omega_o < 0). 
\end{cases}
\]

(2)

\[
f^S_y(\omega_o, \omega_y) = k_y^S \cdot r(x, y, \sigma_i)^N_i.
\]

\[
\begin{cases}
\frac{1}{4} G_y(\omega_o, \omega_y, \omega_n) D(\omega_o, \beta_y) & (\omega_o \cdot \omega_y > 0), \\
[\omega_o \cdot n(y)] [\omega_o \cdot n(y)] & (\omega_o \cdot \omega_y < 0). 
\end{cases}
\]

(3)

where \( n \) is the surface normal function, which is computed using the ply geometry at a given point and then further transformed based on the fiber alignments as described in §3. We denote this transformation as \( n_f(\beta_p) \). \( \eta \) represents the refractive index, \( F \) is the Fresnel reflection coefficient, \( \omega_o \) and \( \omega_y \) are the normalized half vectors for two different cases, and \( G_2 \) is the Smith uncorrelated masking-shadowing function. The term \( D(\beta, \beta_p) \) represents the normal distribution function (NDF) with \( \beta \in \mathbb{R}^2 \) as the (anisotropic) roughness, and \( k_3^S \) is the specular albedo.

Notably, a key distinction from the ply-based model [Montazeri et al. 2020] is the updated attenuation term \( \tau \), marked in gray. The term \( r(x, y, \sigma_i) \) considers the attenuation of light between \( x \) and \( y \) using the Beer-Lambert law [Swinehart 1962]. 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 \( N_i \), the number of intersected plies along the elliptical cross-section for the projected ray. Thus, as a ray passes through the yarn cylinders, we update the attenuation term as \( r(x, y, \sigma_i)^N_i \) 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 [Montazeri et al. 2020; Yan et al. 2017a]. At point \( y \), the body component \( f^B_y \) is represented by a Lambertian term. On the other hand, \( f^B_x \) additionally considers a Lommel-Seeliger (LS) term [Hanrahan and Krueger 1993; Jensen et al. 2001] as a mix, to ensure the energy conservation. The expressions for \( f^B_y \) and \( f^B_x \) are as follows:

\[
f^B_x(\omega_x, \omega_o) = r B G_p(x, \omega_o) G_f(x)^B_x \int_{\Omega_x} [\text{LS}(\omega_o, \omega_o) + \frac{1}{\pi}] \, d\Omega_x.
\]

(4)

\[
f^B_y(\omega_o, \omega_y) = (1-r) B G_p(x, \omega_o) G_f(x) \frac{f^B_y}{\pi}.
\]

(5)

where \( k^B \) is the body albedo and \( r \) represents the reflected portion of the body energy \( B \) and determined in correlation with the refracted albedo of \( f^S_x \). The terms \( G_p(\cdot, \omega_o) \) and \( G_f(\cdot) \) 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 [Irawan and Marschner 2012; Sadeghi et al. 2013], we introduce an additional shadowing component that considers the occlusion caused by one ply on another (\( G_p \)). In 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 estimate 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...
details. When the camera is distant, and the pixel footage spans the entire yarn, we employ a far-field model $f_{\text{far}}$. In this case, we numerically integrate the BYSDF over the full azimuthal offset range from $h_{\text{max}} = 1$ to $h_{\text{min}} = -1$. This integration captures the overall yarn 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 smaller 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 $C = \int_{-1}^{1} W(h) \, dh$.

$$f_{\text{far}}(\omega_i, \omega_o) = \frac{1}{C} \int_{-1}^{1} W(h) f_{\text{near}}(\omega_i, \omega_o, h) \, dh. \tag{6}$$

To achieve a smooth transition between the near-field model ($f_{\text{near}}$) and the far-field model ($f_{\text{far}}$), 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 $(h_1, h_2) \subseteq (h_{\text{min}}, h_{\text{max}})$, 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 supplementary and the final formulation of our multi-scale appearance model combines the near-field and far-field components as follows:

$$f_{\text{multi}}(\omega_i, \omega_o) = \frac{1}{C} \int_{h_1}^{h_2} W(h) f_{\text{near}}(\omega_i, \omega_o, h) \, dh. \tag{7}$$

with $C$ being the normalization over $[h_1, h_2]$. To approximate the multi-scale BYSDF ($f_{\text{multi}}$), 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 (§3) and the shading model (§4).

Specifically, note that the methods we use to compute $f_{\text{multi}}$ and $f_{\text{far}}$ 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 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 ($n_{\text{multi}}$) and tangent ($t_{\text{multi}}$) vectors are then used in equations (2) and (4) to compute $f_{\text{multi}}$. The computation of $n_{\text{multi}}$ and $t_{\text{multi}}$ can be expressed as follows:

$$n_{\text{multi}} = \frac{1}{C} \int_{h_1}^{h_2} W(h) n(h) \, dh, \quad t_{\text{multi}} = \frac{1}{C} \int_{h_1}^{h_2} W(h) t(h) \, dh. \tag{8}$$

Similarly, to find the aggregated shading component covering the queried azimuthal range, we use the weighted average of the fiber-shadowing terms queried from the precomputed 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, \( \beta_{\text{multi}} = \beta + \frac{(h_2 - h_1)}{C} \), which naturally contributes to a rougher appearance.

\[
G_{\text{multi}} = \frac{1}{C} \int_{h_1}^{h_2} W(h) G_f(h) \, dh. \tag{9}
\]

Additional implementation details for the multi-scale feature are added in the supplementary.

5 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. [2020], which was evaluated against real photographs in the original paper. We use the differentiable setup in Mitsuba 3 to match the set of parameters of our model against the reference. The set of parameters and details regarding parameter fitting can be found in the supplementary. We also compare the BYSDF with the model by Khungurn et al. [2015] which to this end is known as the fiber-based reference as well as Zhu et al. [2023] as the recent yarn-based model. Our implementation of the geometric and appearance representation is integrated into the Mitsuba renderer [Jakob et al. 2022] 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 [Khungurn et al. 2015]. 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 backlighting due to the proper transmission sampling, while Zhu et al. [2023] showcases an unwanted sharp transparent silhouette from the plies in the back. 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 [Montazeri et al. 2020], especially when dealing with yarns that have multiple plies.

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 [Zhu et al. 2023]. 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. [2018] and the fiber curves are generated procedurally [Zhao et al. 2016] and shaded accordingly [Khungurn et al. 2015]. Please refer to Table ?? for a detailed comparison.

Multi-scale results: In Fig. 6, 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. [Yuksel 2020].

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.

Please see the supplementary for the additional results.

6 DISCUSSION AND CONCLUSION

Limitations and future works: Our model adds fiber texture using 1D texture maps following the ply-based model [Montazeri et al. 2020]. Ideally the implicit ray tracing along the elliptical yarn cross-section, employed for incorporating ply geometry, can be extended for a comprehensive solution to seamlessly integrate fiber details without the need for texture maps. Additionally, the appearance of fabrics can include loose fibers sticking out, referred to as "flyaways," which are not integrated into our current framework. However, flyaways can be incorporated as future extensions by
introducing random distributions and considering the lifespan of fibers.

Our model works for woven and knitted patterns, it would be interesting to explore an accurate model for fabrics featuring a bundle of protruding fibers, as seen in materials like velvet.

**Conclusion:** We have introduced BYSDF, a practical yarn-based model that efficiently captures the appearance of cloth. First, by reducing the complexity of geometry so the cost of ray tracing is lower, second, we avoid complex light transport between plies and fibers so the number of rays needed is smaller. Our model excels in rendering close-up views of cloth samples, known as the near field. We have also proposed an efficient solution for seamlessly integrating the geometry and appearance of cloth as the camera moves away from the object, enabling a smooth transition from the near field to the far field.

Our BYSDF model extends the reference ply-based model by computing an aggregated shading model at the yarn level, accurately reproducing the appearance of cloth and capturing the implicit geometries of ply and fiber. We have addressed the absence of explicit geometry by incorporating realistic shadowing effects. Compared to existing methods, our model offers a significant improvement in performance, providing high-fidelity results with faster rendering speed and reduced memory usage for near-field renderings. Additionally, our multi-scale solution allows for a substantial reduction in samples per pixel and offers enhanced efficiency for far-away renderings. We believe that our research will benefit a variety of offline graphics production for visual effects and feature animation, such as Weta Digital.

**REFERENCES**


A Multi-scale Yarn Appearance Model with Fiber Details

Fig. 7. Single yarn comparison. Our approach is compared with the reference ply-based model [Montazeri et al. 2020], fiber-based model [Khungurn et al. 2015] and recent yarn-based model [Zhu et al. 2023] for a single yarn to observe the micro-scale events. The material parameters follow the measured values of silk [Khungurn et al. 2015] and the rendering time is shown on the top left, memory on the bottom left, and the number of bounces on the top right.

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 noise-level rendering for zoom-out (left 3 columns), our multi-scale model (95s) is around 2.4× faster than the near-field model (235s). Note that in c 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. Our multi-scale result is correct and efficient for both close-up (right 3 columns) and faraway views. Refer to the supplementary video for a seamless transition demonstration.
Fig. 9. **Comparison with previous yarn-based model** Our model provides a close match to the reference ply-based model [Montazeri et al. 2020], offering faster rendering speed and reduced memory usage for this 6-ply knit sample. The rendering time is shown on the top left, memory on the bottom left, and the number of bounces on the top right. Besides, in comparison to the yarn-based model [Zhu et al. 2023], ours enjoys 1.2 times faster performance as well as presenting the fiber-level detail in close-up views. Please note since their model cannot accurately handle the geometric information of the multi-ply yarns, the appearance looks more rigid and lacks soft shadows with inaccurate highlights.

Fig. 10. **Comparison with previous fiber-based model** Our model provides a close match to the reference ply-based model [Montazeri et al. 2020], more efficiently as listed. The rendering time is shown on the top left, memory on the bottom left, and the number of bounces on the top right. 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 [Khungurn et al. 2015] can achieve a relative match to the reference, but at a significantly higher computational cost. The woven samples are four- and five-ply yarns, respectively.