DOI: 10.1111/cgf.15283

COMPUTER GRAPHICS *forum*
Volume 0 (2025), number 0, e15283

A Texture-Free Practical Model for Realistic Surface-Based Rendering of Woven Fabrics

Apoorv Khattar,[1] Junqiu Zhu,[2] Ling-Qi Yan[2] and Zahra Montazeri[1]

[1]University of Manchester, Manchester, UK
{apoorv.khattar, zahra.montazeri}@manchester.ac.uk
[2]University of California Santa Barbara, Santa Barbara, USA
zhujunqiu@mail.sdu.edu.cn, lingqi@cs.ucsb.edu

Abstract
Rendering woven fabrics is challenging due to the complex micro geometry and anisotropy appearance. Conventional solutions either fully model every yarn/ply/fibre for high fidelity at a high computational cost, or ignore details, that produce non-realistic close-up renderings. In this paper, we introduce a model that shares the advantages of both. Our model requires only binary patterns as input yet offers all the necessary micro-level details by adding the yarn/ply/fibre implicitly. Moreover, we design a double-layer representation to handle light transmission accurately and use a constant timed ($O(1)$) approach to accurately and efficiently depict parallax and shadowing-masking effects in a tandem way. We compare our model with curve-based and surface-based, on different patterns, under different lighting and evaluate with photographs to ensure capturing the aforementioned realistic effects.

Keywords: rendering, ray tracing, reflectance and shading models

CCS Concepts: • Computing methodologies → Reflectance modelling; Ray Tracing; Rendering;

1. Introduction

Woven fabrics play an important role in numerous applications such as textile design, online retail and digital entertainment, including games and films. Rendering these fabrics with high fidelity is crucial, yet it remains one of the most resource-intensive tasks in computer graphics. Fabrics exhibit a complex hierarchical structure, beginning with *yarns*, each composed of numerous *plies*, which in turn are made up of hundreds of twisted *fibres*. The complexity of this structure, leads to even more complex light transport, necessitating a sophisticated appearance model to accurately simulate their properties.

Existing models can be broadly categorized into two types: curve-based and surface-based. Curve-based models represent each fibre [ZJMB11], ply [MGZJ20] or yarn [KZP*24], explicitly and offering high-level detail but at a high computational and storage cost to handle the curves explicitly. These methods, involving complex geometry, are not efficient for large-scale fabric samples. On the other hand, surface-based models, which utilize simpler mesh geometries, often lack the depth and details [IM12, SBDDJ13] or require pre-computation, only handling limited known weave patterns and overlook parallax effects [ZJA*23]. This results in a trade-off between geometric simplification and the sophistication of the appearance model. Currently, no efficient surface-based model adequately provides fibre-level details with a faithful appearance suitable for a close-up view.

This paper proposes a surface-based appearance model for woven fabrics that addresses these issues by offering an efficient geometric intersection with fibre details and no requirement for pre-computation allowing the model to adapt to arbitrary weave patterns. Additionally, it incorporates complex visual features including a double-layer model with accurate light transmission, parallax effects and shadowing. All these features are typically lost in existing mesh applications due to geometric simplification.

To conclude, we highlight our key contributions to geometric representation:

- **Micro-level detail representation**: We include individual yarns, plies and fibres, thereby outperforming the capabilities of previous surface-based models ([IM12, SBDDJ13]).

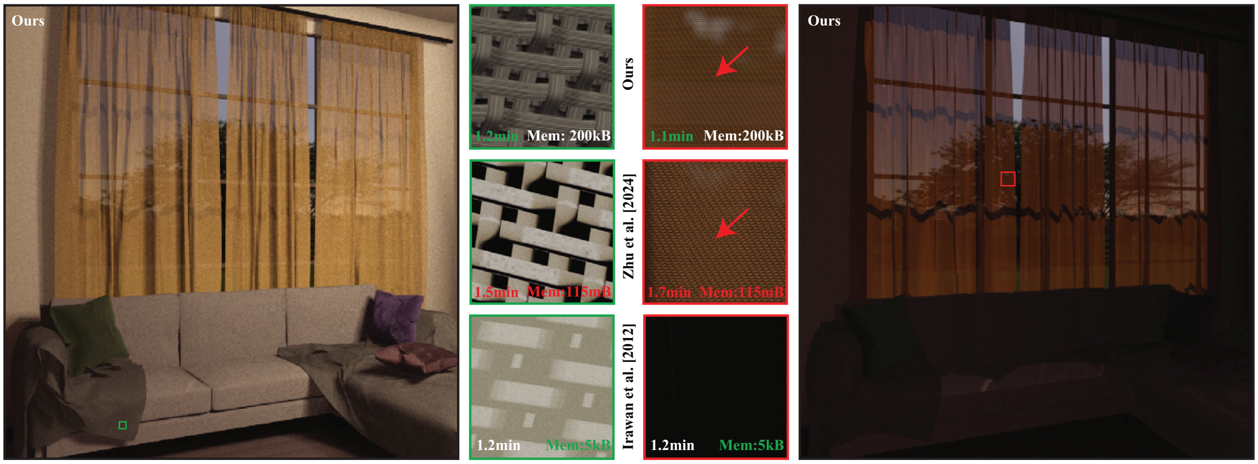


Figure 1: *In this indoor scene, the fabric surfaces show distinct weave patterns: thin twill for the curtains, thick plain for the sofa, thick twill for the cushions and satin for the blankets. Two lighting conditions are shown: exterior light for curtain transmission (right) and combined interior/exterior light for simultaneous reflection and transmission (left). We compare our model against previous surface-based models by Irawan et al. [IM12] and Zhu et al. [ZHB*24]. At the first zoom level (red), both our model and Zhu's support light transmission, unlike Irawan's. Our model improves on Zhu's by updating fabric mesh geometry to account for yarn curvature, revealing the window bar through yarn gaps (red arrow). At the second zoom level (green), Zhu's and Irawan's models lack parallax, shadowing and fibre detail in close-ups. Zhu's method also needs pre-computation maps, while ours uses only 1D yarn cross-section texture maps and a binary pattern, reducing memory by 99.8% and render time (pre-computation+tracing) by 27.6%.*

- **Texture-free definition**: Our implementation eliminates the need for input-specific pre-computation, unlike previous works [ZJA*23, ZHB*24], enabling the accommodation of any arbitrary weave pattern.

As well as on appearance representation:

- **Double-layer model**: We introduce a novel model that simulates realistic light transmission and attenuation through fabrics where the yarns overlap, effectively mimicking interactions within a virtual volumetric medium.
- **Handling parallax and shadowing**: Our approach combines parallax effects and shadowing through a novel Newtonian-based iterative approach that requires a fixed number of iterations (two steps) by relying on the geometric priors of yarns in a weave pattern. Thus avoiding the computational costs associated with traditional ray tracing in curve-based models and eliminating the storage costs associated with texture mapping in previous surface-based models.

We validate our model against real photographs (Figure 11) and compare it with existing surface-based models, as illustrated in Figure 1. Our model matches the efficiency of the light models like Irawan et al. [IM12], while achieving higher fidelity than Zhu et al. [ZHB*24], without requiring input-specific pre-computed textures. We also validate against the existing curve-based models to demonstrate that our model can achieve a close match in a close-up view without modelling explicit curves, which previous surface-based models cannot. Furthermore, we demonstrate its capability to capture backlighting effects for light transmission in far-view and depth-like appearance in close-up views. We plan to release the source code upon acceptance of this paper to support further research and development.

2. Related Work

Cloth rendering models can vary based on the method of geometric representation and the appearance model used for matching. This is due to the complex geometric features of cloth, which require different approaches to accurately capture their shape and appearance. Fibres form plies which in turn are combined into yarns, and these yarns are then woven or knitted into fabrics. Zhao et al. [ZJMB11] analyzed the geometric characteristics of cloth models. Such methods detailed model the fibre-level details and are considered curve/volumetric-based, while other methods, which only require surface geometry, are surface-based. We will discuss the advantages and disadvantages of these two approaches in detail.

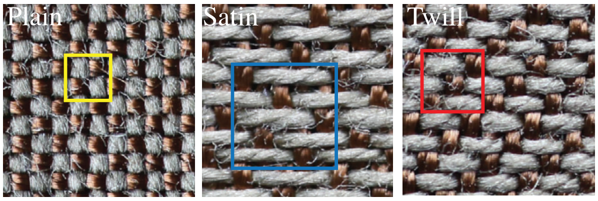
Curve/volumetric-based models: A large number of works define the cloth as an assembly of fibres or plies. Volumetric models rely on micro-imaging techniques to extract the precise fibre distribution and represent the cloth as a heterogeneous volumetric representation. The light interactions are then formulated using the anisotropic radiative transport equation (RTE) [JAM*10]. Further advances [ZJMB11, ZJMB12, KSZ*15] have been made to accurately capture the *micro-level* appearance while representing cloth as a heterogeneous volume. However, these methods are slow, memory intensive and acquiring data is challenging.

The challenges of representing cloth as a heterogeneous volume are addressed by curve-based models. The main advantage of curve-based cloth models is the accurate representation of fibre-level geometric details at a microscopic scale as curved cylinders. This alleviates the need for solving the anisotropic RTE and allows the incorporation of fast ray tracing techniques. Khungurn et al. [KSZ*15] introduced a fibre-based model considering the light interactions and scatterings between fibres for cloth appearance.

However, the storage requirements for 3D details in curve-based methods are substantial, and the complexity of multiple bounces in modelling curves makes efficient rendering challenging. To address some of the practical concerns, Zhao et al. [ZLB16] proposed a procedural model to account for fibre geometry using CT data. Recently, Montazeri et al. [MGZJ20, MGJZ21] reduced the modelling of geometry to only the level of plies, implicitly modelling the underlying fibres' contribution with approximate single and multiple scattering. Zhu et al. [ZJA*23] further proposed a method that models only yarn curves, hiding ply and fibre geometries into a multiple scattering appearance model. Khattar et al. [KZP*24] also propose a yarn-based model that takes into account the elliptical shape of plies addressing the shortcomings of the previous curve-based model and improving performance. However, these methods are still curve-based, only to different levels of simplification and complex light transport between plies/yarns still requires significant storage and extensive ray tracing. Furthermore, Soh et al. [SM24] proposed a fully neural approach to aggregate yarn appearance model, however, this method requires extensive material-specific training and encounters similar storage challenges associated with maintaining yarn curve data.

Surface-based models: To further improve practicality, as the name indicates, cloth is defined as 2D thin sheets, usually on the surface of a 3D mesh. In this way, the appearance of cloth can be easily represented by bidirectional reflectance distribution functions (BRDFs) [AMTF03, IM12, SBDDJ13, JWH*22]. The surface-based models are popular for their fast performance; however, these models often lack realism due to missing yarn/ply/fibre details (often compensated in an ad-hoc texture mapping process), energy transmission (often misunderstood as transparency/alpha) and shadowing-masking (either missing or homogeneous regardless of local structures). Recently, Zhu et al. [ZJA*23] proposed a realistic surface-based cloth shading method. This method is able to reproduce important visual features, including surface highlights, transmittance and delta transmittance, providing a realistic representation of cloth appearance. Zhu et al. [ZHB*24] build upon this to incorporate parallax and sheen effects for far-view renderings. This method requires pre-computed texture maps for normals, tangents and height of the yarns and summed-area table (SAT) which increases the storage requirements to more than 100 MBs. Chen et al. [CWW24] propose a neural network-based approach where the network takes as input texture maps, same as [ZHB*24], a spatial query and the incident+outgoing direction to output the four components of their appearance model. Their model focuses on real-time far-view renderings and is thus excluded from comparison. We incorporate fibre-level details and propose a data-free parallax mapping technique to ensure a more realistic representation of cloth in close-up without additional storage overhead.

Parallax effect: Defining fabric solely by surface geometry is insufficient, as the yarn/ply geometry of the fabric creates significant occlusion and parallax effects, which are especially important at grazing angles. Typically, techniques such as displacement mapping [BAC*18, CFS*18, FHL*18], shell mapping [PBFJ05, Rit06] and parallax mapping [KTI*01] are used. However, due to the high computational cost of displacement mapping and shell mapping, we believe parallax mapping is more suitable due to its speed.



$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Figure 2: *Sample weave patterns. Three common patterns are taken from Leaf et al. [LWS*18b]. The minimum weave patterns are marked with squares, showing the corresponding binary matrices. Our model, however, is not limited to any particular pattern and allows arbitrary custom-designed weave patterns.*

Parallax mapping [KTI*01] is a classic technique that improves bump and normal mapping by adding an approximate parallax effect. Parallax is the effect where the position or direction of an object appears to differ when viewed from different positions. In the context of cloth rendering, parallax mapping traces the surface geometry but virtually intersects the heightfield, avoiding the heavy computation of displacement. Other parallax-aware methods [OP05, WTS*05] can also enhance the geometry details in rendering. parallax occlusion mapping (POM) [Tat06] uses an iterative approach to find the virtual intersection by using heightfield textures for each type of cloth and offsetting the UV-map of the surface. As a consequence, POM is dependent on the step size and the resolution of the heightfield texture. [Kuz21] proposed a neural offset method, but they need to train a neural network for each type of cloth. Compared to the previous state-of-the-art methods, our Newtonian approach is efficient (thanks to the simple definition of weave patterns), accurate and general (it requires no pre-computed textures and works for any arbitrary patterns). This is discussed further in Section 7.3.

3. Motivation

Before introducing our proposed model, it is important to analyze the motivation and discuss the visual and geometric features of real-world fabrics, which underpin our model's development. Woven fabrics exhibit distinct characteristics due to their hierarchical structure. At the macroscopic level, these fabrics appear relatively flat and consist of two orthogonal types of yarns: warps and wefts. These yarns interlace to form repetitive weave patterns, which can be abstracted into a binary matrix representation where "1" indicates a warp overlapping a weft, and "0" signifies a warp lying beneath a weft. Figure 2 illustrates some common weave patterns along with their binary representations.

At a more detailed level, each yarn comprises numerous plies, which are in turn made up of hundreds of twisted fibres. The complexity of this structure leads to unique visual effects, such as varied silhouettes and perturbed normals and tangents. Therefore, rendering simplistic representations (such as curved cylinders) is inadequate. In close-ups, individual plies and fibres become visible, especially with backlighting that highlights the translucent nature of the fabric. Additionally, the curvature of yarns is observable in close-up contrasting with their flat appearance in the far-view. From a distance, the repetitive nature of weave patterns produces anisotropic highlights—a natural phenomenon that presents significant challenges to reproduce virtually.

Our aim is a highly detailed yet texture-free approach that eliminates the need for storing input-specific pre-computation texture maps, thereby saving storage space and being adaptable to new weave patterns, unlike previous works. Our method introduces new features, including **data-free** parallax effects and shadowing, without any precomputation. While parallax and shadowing have been addressed in previous work ([ZHB*24]), their approach requires heightfield textures as input and involves heavy precomputation. In contrast, our method incorporates ply- and fibre-level details for close-up renderings. These differences are highlighted in Figure 18. Our method utilizes a cloth mesh with an aligned UV layout and a binary weave pattern. Given this weave pattern, we overlay yarn cylinders onto the mesh taking into account the parallax mapping to avoid the flat appearance of textured yarns in close-ups. Parallax mapping adjusts the perceived position of an object based on the viewer's angle, providing a more dynamic and realistic texture. We employ a Newtonian approach to estimate these parallax mappings effectively, applying heightfield corrections to the UV map for accurate shadowing and texture depth between yarns without the need for explicit heightfield texture maps.

Given the input, the initial phase of our method involves resolving ray-mesh intersections, which then informs the mapping of the binary weave pattern onto the UV-mapped surface, identifying the warp and weft positions. The mapping of yarn/ply/fibre geometries is outlined in Section 4. Subsequently, the second phase of our model simulates the interaction of light with these geometries through an approximation of the double-layer model for BYSDF at the local shading level. This incorporates global effects such as the parallax effect as well as shadowing-masking interactions between fibres, plies and yarns, as elaborated in Section 5.

4. Modelling the Geometry

As outlined in Section 3, our model simplifies the hierarchical geometry of fabrics into a base mesh, while implicitly adding the details of yarns, plies and fibres to compensate for the simplified geometry. The binary weave pattern used in our model indicates the orthogonal arrangement of warp and weft yarns, specifying which yarn overlaps the other. This section describes how the warps and wefts are mapped onto the fabric mesh from a binary pattern as implicit yarns (Section 4.1), and subsequently how ply and fibre-level details are mapped onto these implicit yarns (Section 4.2).

4.1. Pattern mapping

Following the pattern definition from Irawan et al. [IM12], our binary matrix defines how warps and wefts overlap: a "1" indicates a warp overlapping a weft, and a "0" indicates a warp lying below a weft. Our model can accept any arbitrary binary matrix as input, not limited to typical weave patterns. Figure 2 displays various weave patterns alongside their binary representations.

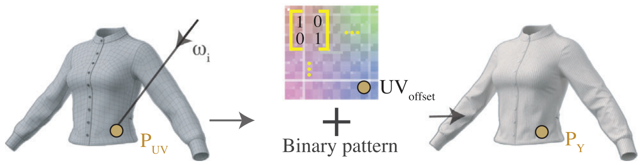
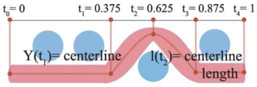


Figure 3: *Pattern mapping. When the ray hits the surface mesh, the corresponding point in UV space is obtained. Using this location and the provided binary pattern, the warp and weft intersections are defined by aligning and tiling the pattern with the UV map.*

Yarns are represented as curved cylinders, with the centerline of each cylinder defined by $Y(t)$. Given the simplified axis-aligned structure of woven patterns, we can model each yarn centerline using a combination of sinusoidal and constant functions to represent the transitions between upper and lower positions. This approach ensures C1 continuity at the points where segments meet and allows $2R$ distance between the top and lower curve, where R is the yarn radius. When a ray intersects with the mesh, a secondary intersection calculation is performed locally, between the ray and the yarn cylinders with radius R. This intersection is determined analytically and executed in constant time, thereby enhancing the efficiency of our rendering process.



For instance, for the binary representation $0, 0, 1, 0$, as illustrated in the adjoining figure, the warp centerline $Y(t)$ is defined as:

$$Y(t) = \begin{cases} -1 & t \in [0, 0.125] \\ \sin(4\pi\ t) & t \in [0.375, 0.875] \\ -1 & t \in [0.875, 1]. \end{cases} \tag{1}$$

Mapping the centerlines of warp and weft yarns onto the mesh surface is achieved using the UV map, which positions them perpendicularly along the U and V directions, respectively. The local displacement of the yarns, $Y(UV)$, is computed at the point where a light ray intersects the mesh. The yarns are modelled as curved cylinders by adjusting their depth in the third dimension. To fully capture the cylinder geometry on the fly, we identify the intersected UV point and its correspondence yarn geometry by determining its tangent and surface normal. The illustration of the pattern mapping is shown in Figure 3.

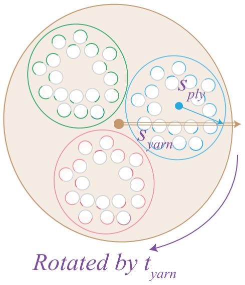
Building on the cylindrical yarn centerlines, we further define the geometric frame (tangent and normal) of these centerlines to obtain the full geometry per intersected point. The tangents are given by calculating the derivative of the actual yarn curve ($Y'_{warp}(U)$) in the local frame and are transformed to the mesh space using the mesh tangent, normal and binormal. The reference normals are calculated by removing the component along $Y'_{warp}(U)$ from the mesh normal. For warps, we use the V-values to determine the azimuthal phase, s_{warp}, which defines the angular distance from the edges of the yarn. Based on the s_{warp} values, we rotate the reference normals along the tangent to obtain the surface normals of the curved warp cylinder. Additionally we calculate the length of the curved cylinders ($l_{warp}(U)$) using the sinusoid equation of the yarn centerline $Y_{warp}(U)$.

The same process is repeated for wefts but with U and V interchanged to obtain the geometric properties of the weft-curved cylinders. Finally, the mesh surface geometry is replaced by that of either the warp or the weft and the yarn type is determined by comparing the local displacements $Y_{warp}(U)$ and $Y_{weft}(V)$ at a given UV coordinate. Please note this process is executed dynamically at each intersected point on the mesh, ensuring that the model operates on-the-fly without the necessity for input-specific pre-computation.

In Figure 4, we visualize a comparison of the geometry and normals obtained by pattern mapping against normal maps of explicit yarn cylinders for two different patterns. We also provide MSE error maps to show the proposed pattern mapping can accurately extract geometric information of the yarns using only the binary pattern.

4.2. Adding ply and fibre details

Once the yarn geometries are established, a secondary mapping is required to incorporate sub-yarn details onto the surface of these cylinders. For the implicit addition of ply and fibre details, we utilize methods introduced by Montazeri et al. [MGZJ20] for embedding fibre details into plies. The integration of this method into our surface-based model once the yarn geometries are realized, becomes simplified and elaborated in the subsequent paragraph.



We adapt this approach to the surface level, enabling the implicit addition of ply and fibre geometry details. Drawing on the procedural model by Zhao et al. [ZLB16], a single 2D slice of yarn is sufficient to represent the yarn comprehensively, as it captures the distribution of plies and fibres (shown in the adjoining figure). The yarn cross-section is a collection of individual ply centres in a yarn and the ply cross-section is a collection of individual fibre centres in a ply. Within each cross-section, we compute the individual ply and fibre details that is the tangent and normals which are stored as a 1D texture that covers the outermost part of each cross-section. Using the s-values discussed earlier, we first access the 1D texture for the yarn cross-section to add ply geometry and then add the fibre geometry using the ply cross-section. We use the l-values (length of yarn) to access the 1D textures in a round-robin manner to incorporate the twisted geometry of plies and fibres, similar to [MGZJ20].

Using 1D textures from a single yarn cross-section can produce a regularized appearance. A naive approach to introduce variation would be to create 1D textures for multiple yarn cross-sections with different fibre orientations and assign a unique set for each yarn. Since our focus is on reducing the pre-computation time and memory, we only use one yarn cross-section and instead offset the s-values for yarns. This ensures that different orientations of the same yarn cross-section are utilized, introducing the desired irregularities.

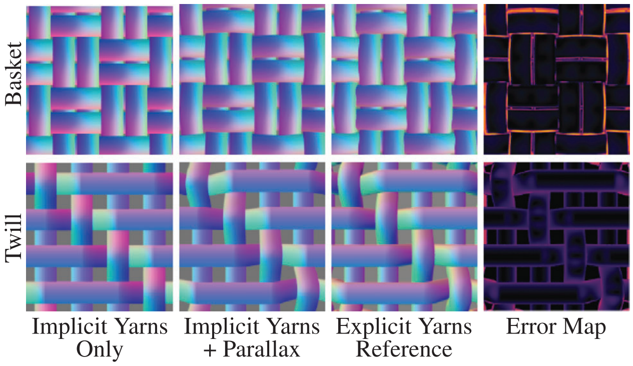


Figure 4: *Normal map visualizations. We visualize the normal map obtained after pattern mapping for two weave patterns: basket and twill. We compare them against normals of explicit yarns generated for each pattern and show the error maps in Column III. Using only the pattern mapping generates a flat appearance (Column I) but with the parallax offset UV Maps (Column II), we can achieve a close match to the geometry of explicitly modelled yarns. Note the curved shape of yarns which is absent from previous surface-based models except [ZHB*24] that requires on-time and memory intensive pre-computation.*

5. Appearance Model

In Section 4, we detailed how the geometric representation of yarns, plies and fibres can be mapped onto the mesh surface on-the-fly. After mapping, the implicit yarn geometry closely resembles that of explicit yarn models. This allows us to adapt any yarn-based shading model (such as [ZMA*23, SM24, KZP*24]). Specifically, we base our model on the bi-directional yarn surface distribution function (BYSDF) introduced by Khattar et al. [KZP*24]. The overview of our BYSDF is outlined in Section 5.1.

The naive approach would be to trace rays inside and between the yarns found by the input binary pattern for transmission and inter-yarn shadows, which would compromise the efficiency gained by a surface-based approach. Our goal is to replicate how light accurately interacts with yarn-based models while preserving the simplicity and efficiency of surface-based modelling. We address the transmission by examining light transport within the yarn layers, employing a *double-layer shading model* to simulate light entering and exiting the yarn medium. This approach is detailed in Section 5.2.

To compensate for the loss of visual depth resulting from our model's geometrical simplification, we integrate *parallax mapping* as detailed in Section 5.3, which helps to restore the depth and micro-structure often compromised in surface representation. Moreover, to account for the lost shadow effects in between yarns, we refine our shadow mapping techniques as outlined in Section 5.3.

5.1. BYSDF overview

The BYSDF model we employ is divided into forward and backward components to capture the interaction of light with the yarn medium at the point x and exit point y. The backward component accounts for both the immediate reflection, manifesting as the

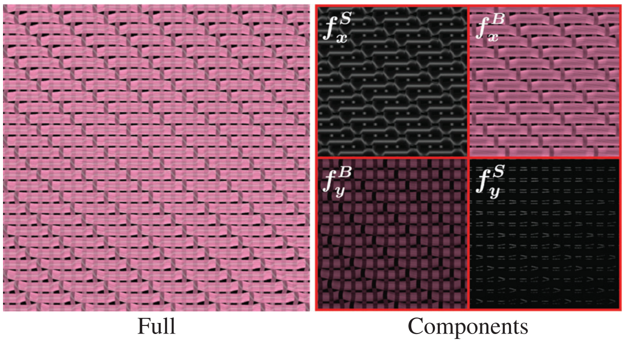


Figure 5: *Visualization of all four components. Illustration of the four lobes: f_x^{S} (top left), f_x^{B} (top right), f_y^{B} (bottom left) and f_y^{S} (bottom right) of the appearance model and their individual contributions.*

specular (f_x^{S}) and the scattered light exiting from the same side as the incident ray, known as the *body* property (f_x^{B}). Similarly, the forward component, handles light that transmits through the yarn, capturing both the specularly transmitted (f_y^{S}) and scattered light (f_y^{B}) continuing in the forward direction. These four components are illustrated in Figure 5 and formulated as follows:

$$f_x = f_x^{\mathrm{S}} + f_x^{\mathrm{B}}, \quad f_y = f_y^{\mathrm{S}} + f_y^{\mathrm{B}}. \tag{2}$$

They are further detailed in the supplementary document with the modification explained in Section 5.2.

5.2. Double-layer model

When a ray intersects the fabric mesh, the backward components of the BYSDF can be applied directly for reflections, given the close resemblance of our mesh to explicit yarns. However, for light transmission, the ray in a yarn-based BYSDF model would typically traverse the yarn medium and emerge at a different point. For a fabric mesh, the ray exits directly from the other side of the mesh without interacting with an explicit yarn medium. To address this discrepancy due to the simplified geometry, we propose a modified forward component, termed the *double-layer model*. This enhances the accuracy of light transport through our implicit yarn representation by simulating two layers of yarn. We do not consider explicit transmission with different entry and exit points but instead approximate the distance a ray travels through the double-layer leveraging the geometric information obtained from pattern mapping (Section 4.1).

The attenuation factor plays a crucial role in our surfaced-based model, particularly in simulating the volumetric appearance which is otherwise lacking. This factor is key in enhancing the visual impact at the edges of the yarns, making them appear brighter as the distance between entry and exit points, $\boldsymbol{x}$ and $\boldsymbol{y}$ decreases. This adjustment creates a perception of volume rather than a flat texture, essential for realistic yarn rendering. In explicit yarn models, the original attenuation factor τ for the yarn cylinder, accounting for material extinction coefficient σ_{t}, is calculated as follows:

$$\tau(\boldsymbol{x}, \boldsymbol{y}) := \exp(-\sigma_{\mathrm{t}} \|\boldsymbol{x} - \boldsymbol{y}\|). \tag{3}$$

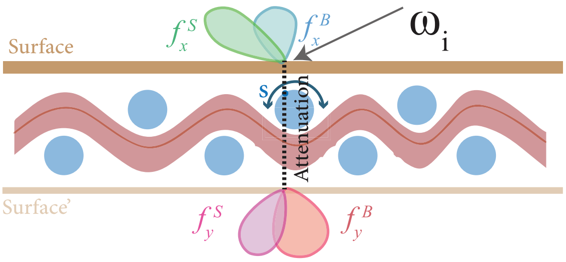


Figure 6: *Double-layer model. Once the incoming ray hits the surface, the backward components follow the original BYSDF but the forward components are adjusted based on the updated attenuation term. The distance a ray travels through a virtual warp and weft is approximated before it exits from the second layer.*

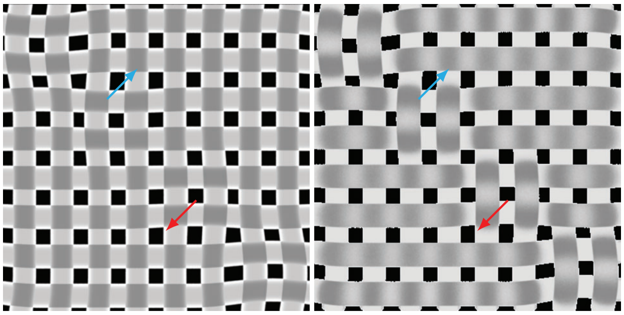
The points $\boldsymbol{x}$ and $\boldsymbol{y}$ are determined using the path tracer by sampling the transmitted ray and intersecting it with the explicit yarns. This process is computationally expensive due to the need for multiple ray bounces. To mitigate the lack of explicit yarns, we simulate the translucent appearance using our double-layer model. We observe that the distance travelled by a ray through the yarn medium correlates to the azimuthal phase, s such that the attenuation increases as s → 0. Thus, we modify the attenuation term relying on the s-values, which is defined as the positional offset between the intersected point on the yarn surface and the central axis of the yarn. The s-values vary from -1 to 1, a value of 0 indicates that the hit point is at the yarn's central axis, and it shifts as the hit point moves towards the edges of the yarn. The modified attenuation penalizes the rays more for s → 0, thereby making the edges appear brighter.

The modified attenuation taking into account the s value is given by,

$$\tau(\boldsymbol{x}, \boldsymbol{y}) := \exp\left(-\sigma_{\mathrm{t}}\, R\, G(s)\right), \tag{4}$$

while we approximate $\|\boldsymbol{x} - \boldsymbol{y}\|$ as $R\,G(s)$. R is the radius of the implicit yarns and $G(s) = \log(2 - |s|)$ is a non-linear smoothing of s from 0 to 1 to make attenuate the rays making the appearance brighter on the yarn edges. This is also illustrated in Figure 6.

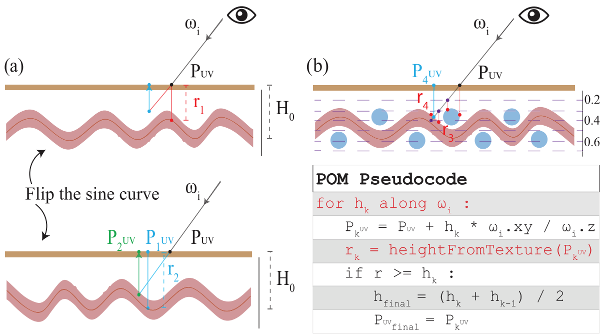
To validate the double-layer model, we set the yarn surface colour to 1 and visualize the attenuation experienced by rays as they travel through the yarn medium for a custom 8 × 8 weave pattern, as shown in Figure 7. Where the yarns overlap, increased attenuation results in a darker appearance (indicated by a blue arrow), while at the yarn edges, lower attenuation creates bright spots (indicated by a red arrow). The double-layer model provides a good approximation for simulating the volumetric appearance but it is not a perfect match because the transmitted ray is not explicitly sampled. This model offers a trade-off between performance by reducing the number of ray bounces (four for explicit yarns, only one for our model) and the softer appearance achieved through explicit sampling of the transmitted ray in the yarn-based model [KZP*24]. We perform a qualitative and quantitative comparison to justify the use of a double-layer model in Section 7.1.



Double-Layer Approximation

Explicit Refraction With BYSDF

Figure 7: *Visualization of the attenuation for rays travelling through the yarn medium. We compare the attenuation of rays refracted through fabrics using our double-layer model for a custom weave pattern mapped to a flat mesh with that of rays refracted through explicit yarns generated for the weave pattern. In regions where the yarns overlap, higher attenuation results in a darker appearance, while lower attenuation near the edges creates a brighter appearance.*



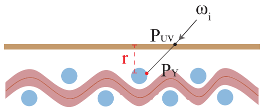
```
POM Pseudocode
for $h_k$ along $\omega_i$ :
    $P_{k^{UV}}$ = $P_{UV}$ + $h_k$ * $\omega_i$.xy / $\omega_i$.z
    $r_k$ = heightFromTexture($P_{k^{UV}}$)
    if r >= $h_k$ :
        $h_{final}$ = ($h_k$ + $h_{k-1}$) / 2
        $P_{UV_{final}}$ = $P_{k^{UV}}$
```

Figure 8: *Parallax mapping comparison. (a) Illustrates the proposed Newtonian parallax mapping. In step I, we obtain the local yarn displacement r_1 following the yarn-axis of the sinusoid function according to the binary pattern. In step II, we flip the sinusoid function to obtain the second local displacement r_2 and again offset the UV map to get the final position P_2UV. (b) Illustrates commonly used POM. The process is summarized by inline code.*

5.3. Parallax mapping

From a distance, yarns in a woven fabric appear relatively flat. However, this assumption fails in close-ups and at grazing angles, where the curvature and texture details of the yarns become prominent. Previous surface-based models often represent yarns as flat textures, neglecting the three-dimensional appearance that parallax effects can provide [ZJA*23, IM12]. [ZHB*24] incorporate parallax using pre-computed textures but storing textures for weave patterns increases the storage requirements and pre-computation time. Parallax mapping is crucial in simulating the curved appearance of yarn cylinders on what is essentially a flat mesh by manipulating the heightfield of the micro-geometry, as illustrated in Figure 8.

The primary challenge in traditional parallax mapping is determining the actual hit point on the heightfield rather than simply on a flat texture map. Inspired by previous work [Kuz21], our approach modifies the traditional UV mapping by applying an *offset*, which redefines the depth as if the ray initially hit this new point. By offsetting the UV map using the heightfield value r, we enhance the geometric fidelity of surface-based models.



When a local incident ray, ω_i, hits the fabric at an oblique angle, the point of intersection on the texture, PY, differs from where the texture is sampled at PUV. To correct this, the UV coordinates are adjusted according to the following equation:

$$(U_{\text{offset}}, V_{\text{offset}}) = (U, V) + \frac{r}{\omega_i.z}(\omega_i.x, \omega_i.y), \tag{5}$$

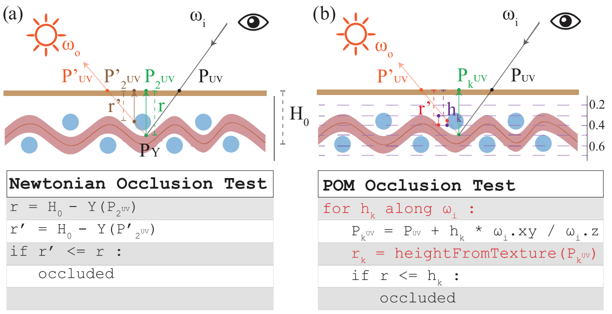
where r represents the heightfield value at the intersection point.

To simplify the computational process and reduce overhead, we do not rely on individual heightfield textures for each weave pattern. Taking inspiration from the Newtonian approach used for implicit ply intersection by [KZP*24], we propose an iterative approach for estimating the heightfield using the geometric information retrieved by pattern mapping (Section 4.1). This approach eliminates the need for storing individual heightfields for different weave patterns which are needed for POM [Tat06].

Parallax mapping not only enhances the curvature of yarns in close-up but also changes the inter-yarn spacing for a given weave pattern. This can be observed in Figure 4 where the parallax-mapped implicit yarns provide a closer match to the geometry of explicit yarns. This effect also allows objects behind the fabrics to become visible through the gaps, referred to as delta transmission [ZMA*23]. Our Newtonian approach allows easy and fast updates on the fabric mesh, providing more accurate delta transmission as can be seen in Figure 1. More information about the implementation details can be found in Section 6.2.

Similar to how POM is used for shadowing our insight is that the parallax mapping method outlined in Section 5.3 can also help for effective shadowing because recognizing the visibility towards the light at any point is linked with parallax occlusion effects. We repurpose the parallax mapping technique to also handle shadowing in between yarns, thereby integrating these two visual effects through a single process.

Essentially the *UV* offset is used to introduce a better *masking* viewed from the observer by sampling a ray towards the emitter from the displaced position of the implicit yarn cylinders. We infer the heightfield (r) by performing parallax and pattern mapping for the incident camera ray and displace the mesh hit point. From the displaced position we sample a ray towards the emitter. The sample emitter ray again hits the mesh surface and we perform parallax + pattern mapping at the new hit point. If the heightfield (r') for the new hit point is less than the r then it is marked as occluded. POM also samples a ray towards the emitter from the displaced position, however it checks for each sample along this ray if the sampled height position h_k is more than than height returned from the heightfield texture r'_k to mark it as occluded. Figure 9 illustrates a



Newtonian Occlusion Test

```
r  = H₀ - Y(P₂ᵤᵥ)
r' = H₀ - Y(P'₂ᵤᵥ)
if r' <= r :
    occluded
```

POM Occlusion Test

```
for hₖ along ωᵢ :
    Pₖᵤᵥ = Pᵤᵥ + hₖ * ωᵢ.xy / ωᵢ.z
    rₖ = heightFromTexture(Pₖᵤᵥ)
    if r <= hₖ :
        occluded
```

Figure 9: *Shadowing. (a) Illustrates our proposed method while (b) illustrates the commonly used technique using POM. The processes are summarized by the following inline codes.*

side-by-side comparison between the two. Same as parallax mapping, our shadowing masking function takes four steps while for POM, the worst case time complexity is $O(N)$, where N is the maximum number of iterations. More details about how we implement the shadowing masking function can be found in Section 6.3.

6. Implementation

Our method is implemented as a custom BSDF in Mitsuba 3 [JSR*22]. We additionally modify the path replay backpropogation integrator (PRB) [VSJ21] to account for the change in shape of the fabric mesh due to spacing between yarns in weave patterns while still supporting environment and delta lighting.

6.1. Importance sampling

Importance sampling enhances the efficiency of a model, particularly under complex lighting conditions. It operates by selectively sampling the more significant parts of a distribution to reduce variance in the final rendering. We adopt the importance sampling as supported by the literature [MGZJ20, KZP*24]. The importance sampling is integrated into our modified BYSDF (Section 5.1) using a specific parameter that optimally allocates energy between the body and specular components of the material, ensuring the energy conservation.

This parameter not only normalizes the energy ratio among the four components of the BYSDF—specular reflection, specular transmission, body reflection and body transmission—but also dictates the allocation of sampling ratio to these components. Once a component is selected based on its relative importance, a per-component importance sampling function is applied.

6.2. Newtonian parallax mapping

POM [Tat06] iteratively takes fixed step size and requires a heightfield texture to find the hit point for the underlying yarns. Our Newtonian approach, on the other hand takes variable length steps offering faster convergence and does not require the heightfield texture saving on memory.

The process starts with separately mapping warps and wefts given the input binary pattern. For warps the initial estimate for the heightfield is given by: $r = H_0 - Y_{warp}(U)$, where H_0 is a user-controlled parameter defining the distance of the base mesh from the yarns. This heightfield is used to offset the UV map. Following Equation 5, the mesh UV map is offsetted and we repeat the process by reversing the warp curve i.e. flipping the sign of the sinusoidal functions. The UV map is offsetted again with the estimated heightfield of the negative sinusoid function to obtain the parallax-corrected warps. We empirically found that flipping the sinusoid function gives a close approximation for the final UV offset as an alternative to taking small fixed-sized steps by repeatedly offsetting the UV map and calculating the new r. The same two steps are repeated for wefts and finally we obtain two offsetted UV maps for warps and wefts respectively. We perform the pattern mapping (Section 4.1) one more time using the two UV maps and the warps and weft are interlaced by comparing their local displacements. The parallax-enabled pattern mapping allows easy updates on the fabric mesh for marking the spaces between the yarns as empty and allowing the ray to pass through seamlessly.

In practice, we found the two steps to provide a reasonable trade-off between accuracy and speed but the number of iterations can be increased for grazing angles.

6.3. Handling inter-yarn shadows

Our shadowing-masking function offers a faster approach to determining if the implicit yarns are occluded. In Figure 9.a, we demonstrate how it works with algorithmic code. For the mesh hit point P_{UV}, the parallax + pattern mapping is used to obtain the implicit height r. We displace the mesh using r to get the approximate point where ω_i intersects the underlying geometry P_Y. From P_Y, we sample a ray ω_o towards the light source. ω_o hits the mesh surface at a new point P'_{UV} and we again parallax + pattern mapping for a second time with new inputs P'_{UV} and ω_o to get r'. If the heightfield, r' at P'_{UV} is smaller than r then the P_{UV} is marked as occluded, otherwise not.

In Figure 9.b, we illustrate the traditional use of a per-pixel visibility function to assess whether the underlying weave pattern texture is visible. Given mesh hit point P_{UV} for incident direction ω_i, POM iterates until it estimates the intersection point with the underlying geometry, P_Y and heightfield r. A ray is sampled towards the light source from P_Y, ω_o. Again POM samples several points at fixed increments along ω_o, and the heightfield is queried by offsetting the UV (Equation 5). If the queried heightfield at any position (r'_k) is less than sample point height (h_k), it suggests that another yarn segment is obstructing the path, thus marking P_{UV} as occluded. The process can be summarized by the following inline code for all sampled points along ω_o.

In the worst-case scenario, POM has to check for every sample before concluding whether an intersection point is occluded and has a time complexity of $O(N)$ where N is the maximum of iterations. Our shadowing-masking function on the other hand, only requires four steps to estimate whether an intersection point is occluded or not.

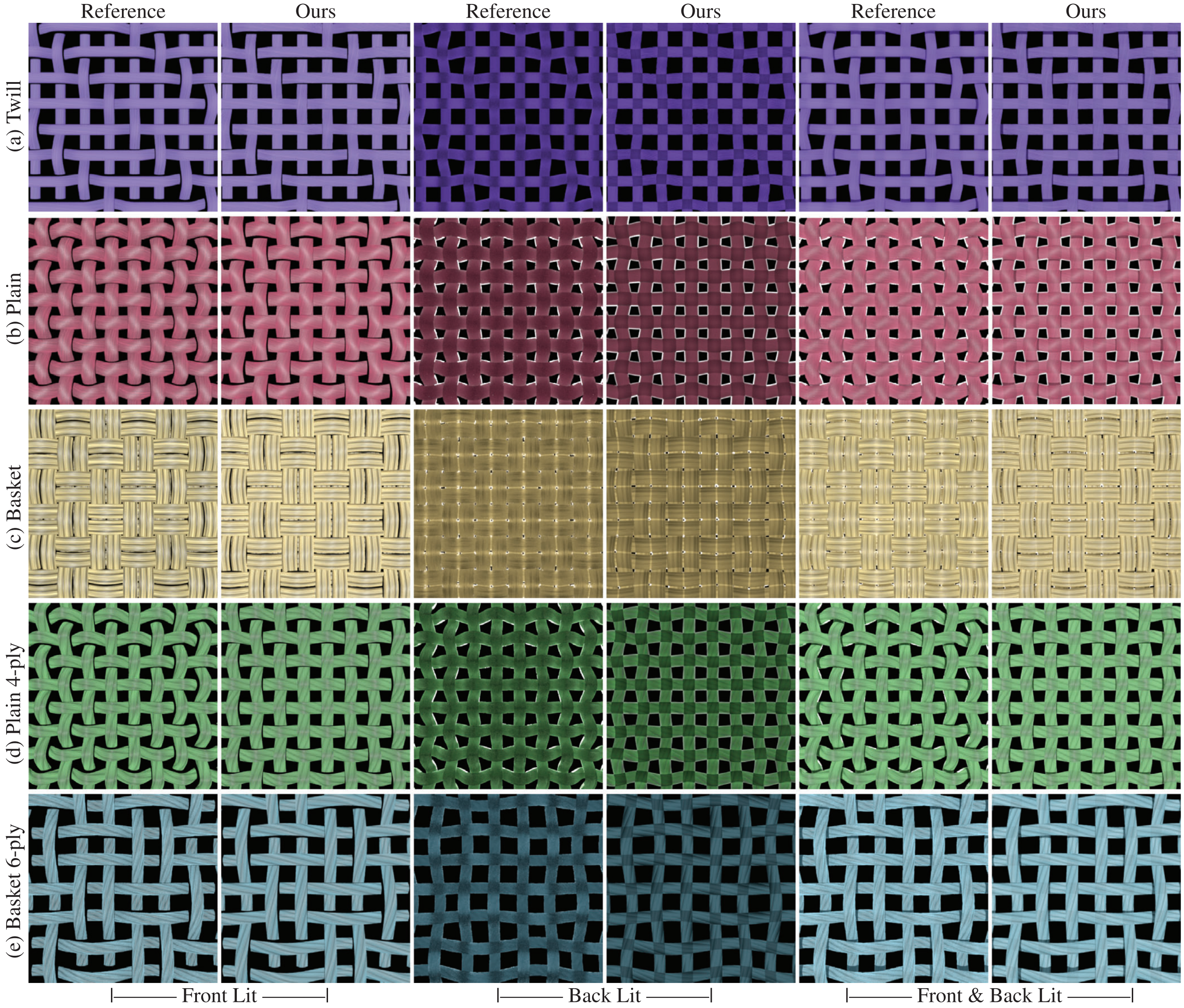


Figure 10: *Comparison to the reference(yarn-based model). Our model provides a close match to the reference yarn-based model, more efficiently. A quantitative performance comparison can be found in Table 1.*

A detailed analysis of the parallax mapping and the shadowing mask can be found in Section 7.3.

7. Results

In this section, we present renderings of our surface-based model across different scales and evaluate it against existing approaches as well as real photographs. At the microscale, we compare our model with the yarn-based model by Khattar et al. [KZP*24] (referred to as Khat'24). At the macroscale, we compare with the ply-based model by Montazeri et al. [MGZJ20] (abbreviated as Monta'20) and the surface-based model by Zhu et al. [ZJA*23] (referred to as Zhu'23). Additionally, we present an ablation study to analyze the impact of parallax mapping and shadows at the microscale. All scenes in the paper were rendered using an Intel Core i7-10700 8-core processor.

7.1. Close-up comparison with yarn-based model

In Figure 10, we show the ability of our model to take into account the curvature of the yarns and the underlying micro-geometry. We compare against the yarn-based model for three different weave patterns: basket, twill and plain. The yarn geometries for three samples were taken from Leaf et al. [LWS*18] to generate the reference. Our model only requires the binary pattern as input to produce a close match to the reference.

Table 1: *Performance statistics for Figure 10 at equal quality (EQ).*

		Render time (min)		Storage (MB)	
Scene	#Yarn	Ours	Khat'24	Ours	Khat'24
Figure 10.a	16	***0.13***	1.03	***0.02***	1.52
Figure 10.b	16	***0.13***	1.03	***0.02***	1.52
Figure 10.c	24	***0.15***	1.53	***0.02***	2.22
Figure 10.d	16	***0.13***	1.03	***0.02***	1.52
Figure 10.e	14	***0.11***	0.98	***0.02***	1.34

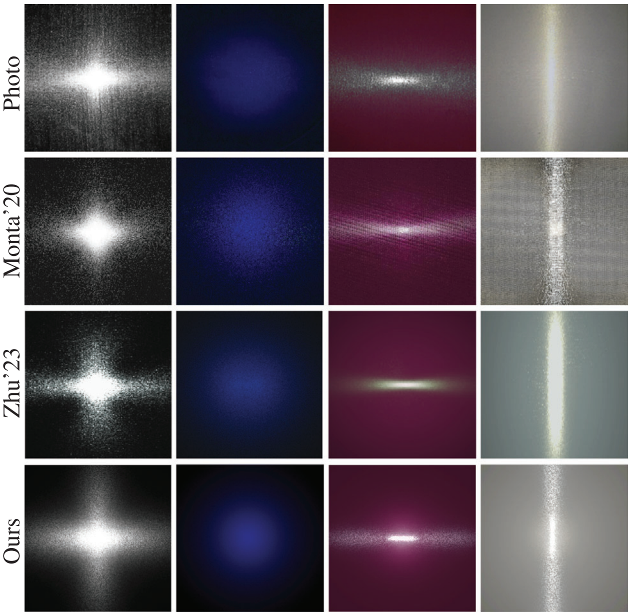
Note: We compare the performance of Ours and yarn-based model [KZP*24] referred to as Khat'24. Our method on average performs 8.7 times faster and saves 98.7% times on storage compared to yarn-based model. #Yarn means the number of explicit yarns interlaced for each pattern to generate the reference.

Our model is able to capture the change in appearance due to the different curvature of yarns in the three woven patterns for arbitrary yarn radii. It also produces a close match in terms of diffuse colour and specular highlights while taking into the inter-yarn shadows. The 1D textures are used to add ply and fibre-level details. For back lighting, the yarn-based model performs proper transmission sampling and attenuation leading to a soft appearance. Our double surface transmission model, on the other hand assumes the attenuation to be inversely proportional to the yarn radius (4) leading to more hard appearance. This is discussed further in Section 8. Our model is 8.5 times faster and requires 98.7% less memory as the input is only a flat mesh and small 1D textures and it requires only a single ray bounce. While the yarn-model requires the explicit yarns and 2–4 light bounces to exit the yarn medium. Please refer to Table 1 for a quantitative comparison of the performance.

Our model currently uses only the binary pattern as input, but it can be extended to directly incorporate weave patterns from Leaf et al. dataset. Since our work focuses on a data-free approach, additional information on using the explicit yarns for pattern mapping is provided in the supplementary document.

7.2. Far-view comparison with photograph

In Figure 11, we showcase the ability of the double-layer forward model to approximate light transmission in the forward direction in the case of backlighting. We do not pursue and expect a perfect match with the reference but aim to demonstrate that our model can effectively reproduce these complex anisotropic effects of various weave patterns. We present comparisons of our model with the ply-based model [MGZJ20] and the surface-based model by Zhu et al. [ZJA*23] (photograph and results of previous works taken from [MGZJ20] and [ZXW19]). We intentionally exclude Irawan et al. [IM12] from this comparison as their model only handles reflections and is not designed to manage backlighting scenarios, for light transmission.



(a) Black Plain (b) Blue Twill (c) Pink Satin (d) Ivory Satin

Figure 11: *Comparison with back-lit photograph at macro scale. The ply-based model is referred to Monta'20 [MGZJ20], and the surface-based model is refer to Zhu'23 [ZJA*23]. Our double-layer transmission model serves as a good approximation for anisotropic highlights for different weave patterns.*

Table 2: *Performance statistics for Figure 11 at equal quality (EQ).*

	Render time (min)			Storage (MB)		
Scene	Ours	Monta'20	Zhu'23	Ours	Monta'20	Zhu'23
Figure 11.a	***1.1***	12.73	1.7	***0.02***	139.8	16
Figure 11.b	***1.2***	4.1	1.7	***0.02***	346.6	15
Figure 11.c	***1.2***	6.3	1.7	***0.02***	100.3	16
Figure 11.d	***1.2***	5.5	1.8	***0.02***	142.3	16

Note: We compare the performance of Ours, ply-based model [MGZJ20] referred to as Monta'20 and surface-based model [ZJA*23], written as Zhu'23 for abbreviation. Our method on average is able to perform 4–10 times faster than Monta'20 model, 1.2–1.5 times faster than Zhu'23.

The ply-based model requires the generation and storage of explicit ply curves (ranges from 100.3 to 346.6 MB in this figure), in comparison our model only requires storing a flat mesh and the small 1D textures of yarn cross-section to store ply and fibre distributions (constant 0.02 MB). Our model is 4–20 times faster and saves 99.8% on storage in comparison. The surface-based model, however, requires the generation of pre-computation maps, unlike ours for additional 20 s on average. Please refer to Table 2 for a quantitative comparison of the performance.

7.3. Ablation of parallax and inter-yarn shadows

We perform an ablation study to showcase the significance of the parallax mapping and inter-yarn shadows.

7.3.1. *Close-up comparison with actual yarns*

In the first experiment as shown in Figure 12, we compare the close-up appearance with and without the two components against the ex-

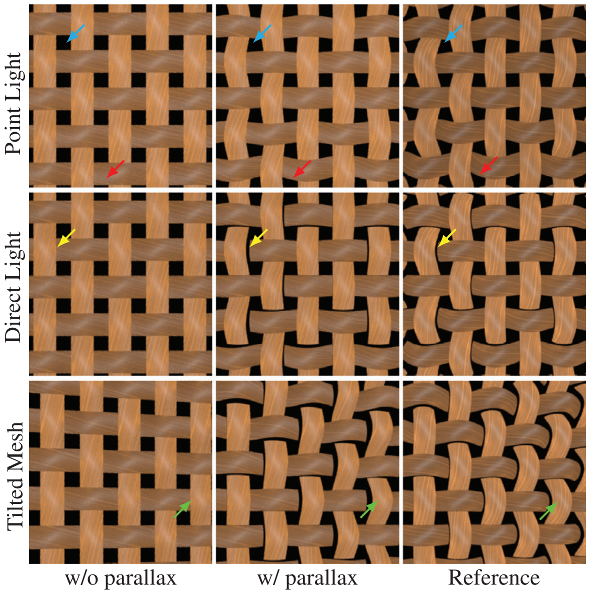


Figure 12: *With and without handling the parallax and shadowing. We show how the parallax and inter-yarn shadows affect the close-up appearance in our surface-based model under different settings. Reference is when the yarns are explicitly modelled [KZP*24].*

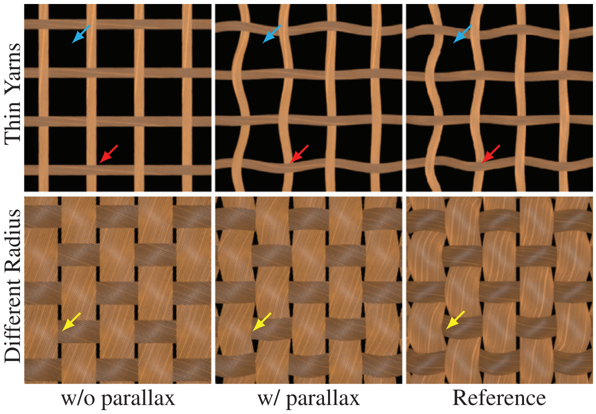


Figure 13: *Intra-pattern flexibility of parallax mapping. We show how our parallax mapping can be seamlessly adapted to yarns of the same pattern having different geometries. In Row I, we show thin yarns which make the curvature more exaggerated. In Row II, we show warps and wefts having different radii. Reference is when the yarns are explicitly modelled [KZP*24].*

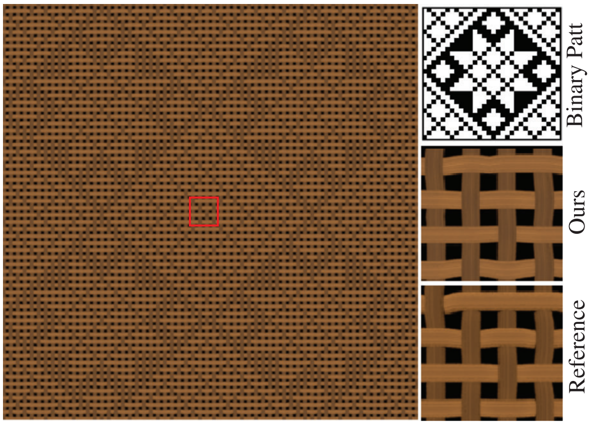


Figure 14: *Parallax mapping on custom weave pattern. We show tiled and close-up renderings for a custom weave pattern as input and compare it against reference yarns that are explicitly modelled.*

plicitly modelled yarns. We choose the plain pattern due to the high curvature of yarns to clearly bring out these differences.

We look at three settings. First, we use a point light placed alongside the camera. In the absence of shadows for the explicit yarns, not using the parallax mapping produces a flat appearance while parallax mapping ensures that the curvature of the yarns is preserved (shown with red arrows). Additionally the spacing between the yarns is warped which is captured by our Newtonian approach (shown with blue arrows). Second, we use a sharp direct light to show the inter-yarn shadows being captured accurately by our model (shown with yellow arrows). Finally, we tilt the mesh such that the rays approach grazing angles. In this setting, the parallax mapping is able to capture the change in the geometry of yarns matching closely with the reference while also taking into account the inter-yarn shadows (shown with green arrows), showcasing the capability to adapt to changing viewing angle. Note: the shadows for explicit yarns are curved more because our Newtonian parallax mapping is done against the sinusoid axis and the shadows for explicit yarns are traced against the curved cylinders. We mitigate this issue by setting the heightfield to $r = H_0 - (\,Y(UV) + R\,N_Y(UV)\,)$, where R is the radius set for the warp/weft yarns and $N_Y(UV)$ is the normal obtained from pattern mapping.

7.3.2. *Generalizabilty of the newtonian approach*

In the second experiment, we show the generalizabilty of our Newtonian approach. For a given weave pattern, the warps and wefts can have different geometries. POM and other state-of-the-art parallax mapping techniques require separate heightfield textures for each configuration however, our approach is only dependent on the binary pattern. This makes the process of modelling yarn appearance on fabric mesh easier. In Figure 13, we show the flexibility of our method when the warps and wefts have different radii. For both thin and thick yarns, the curvature is exaggerated (as shown with red and blue arrows) and our parallax mapping approach faithfully captures the yarn geometry and appearance.

Weave patterns are not limited to the samples we have previously discussed. An arbitrary pattern can be designed to depict more complex structures such as flowers, logos, etc. For an input image, we create a custom-designed binary matrix to be tiled on the fabric mesh. Importantly, the warps and wefts are arranged to ensure that the pattern remains weavable and does not fall apart. We compare the close-up renderings of our approach to reference yarn-based model that requires explicit yarns for the custom pattern in Figure 14. It can be observed that our method is able to adapt to any complex, unseen weave pattern—a distinction often overlooked in

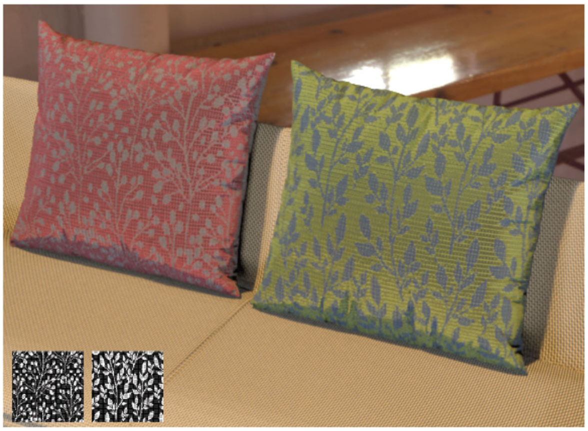


Figure 15: *Arbitrary weave pattern. Our method can support arbitrary weave pattern. In this figure, we show two complex pattern with* 1024 × 1024 *binary matrices mapped to cushions demonstrating the ability of our model to adapt to unseen weave patterns. The sofa showcases a classic plain pattern.*

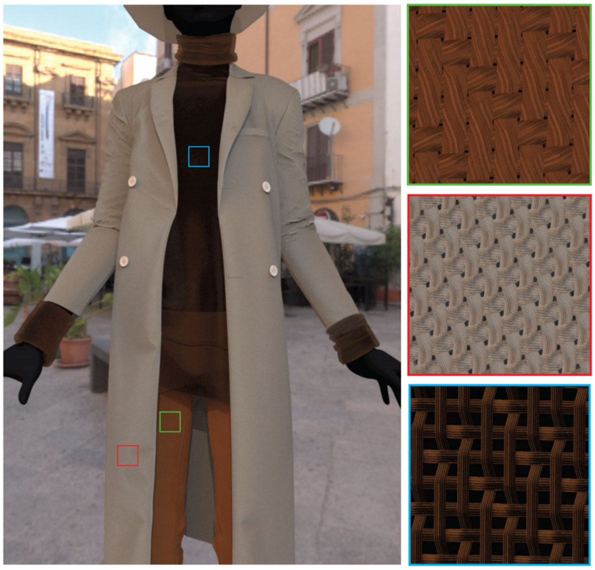


Figure 16: *Day to day scene. Additional rendering comprising of different types of cloth: shiny thin satin sweater, rough twill pants and thick plain jacket in far-view and close-up. The parallax mapping is able to the curvature of yarns in close-up as can be seen in close-up of satin sweater. Please refer to the supplementary video for a zoom-in of the scene.*

previous models. Furthermore, Figure 15 displays two cushions featuring large custom binary patterns, each with a binary matrix size 1024 × 1024 in a complex scene with environment light.

7.4. Day to day scenes

In Figure 16, we show an example of renderings in close-up and far-view of different types of cloth in the presence of environment and point light, composed of a shiny thin satin sweater, rough twill pattern pants and thick plain jacket. The parallax mapping is able to faithfully capture the curvature of yarns in close-up rendering as evident in the plain jacket and satin sweater. Please refer to the supplemental video for a smooth transition, demonstrating the renderings at multiple scales as well as rotating environment light.

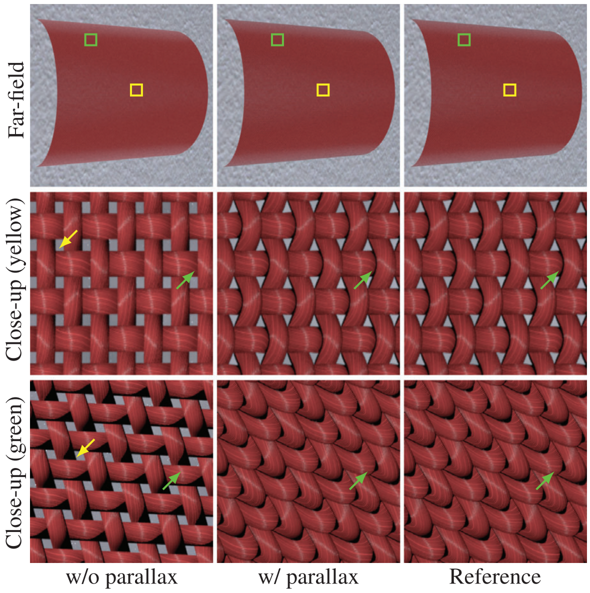


Figure 17: *Comparison with explicit yarns generated for a curved surface. We show how the parallax and inter-yarn shadows affect the close-up and far-field appearance. In close-up, not using parallax produces a flat appearance with missing (shown with green arrow) / incorrect (shown with yellow arrow) inter-yarn shadows. In far-view, the parallax is not noticeable however the missing inter-yarns shadows produces brighter appearance. Reference is when the yarns are explicitly modelled along the cylinder surface [KZP*24].*

7.5. Additional rendering

Figure 17 presents a comparison with explicit yarn curves generated along a cylinder. Our Newtonian approach produces a close match to the reference in both close-up and far-view renderings without the need to generate explicit yarn curves. In close-up, it takes into account the curvature of the yarn and the inter-yarn shadows, as discussed previously in Section 7.3, whereas not using the parallax mapping produces a flat appearance with incorrect shadowing. In far-view, fabrics appear relatively flat, however when the parallax mapping is not done then the incorrect yarn shadowing leads to a brighter appearance.

In Figure 18, we show a comparison with Zhu et al. [ZHB*24] in close-up and far-view for the basket pattern in environment and point light. Their requires tilable texture maps for normals, tangents and heightfield of the yarns in the weave pattern and SAT tables for parallax but is unable to maintain the details in close-up. Ours, on

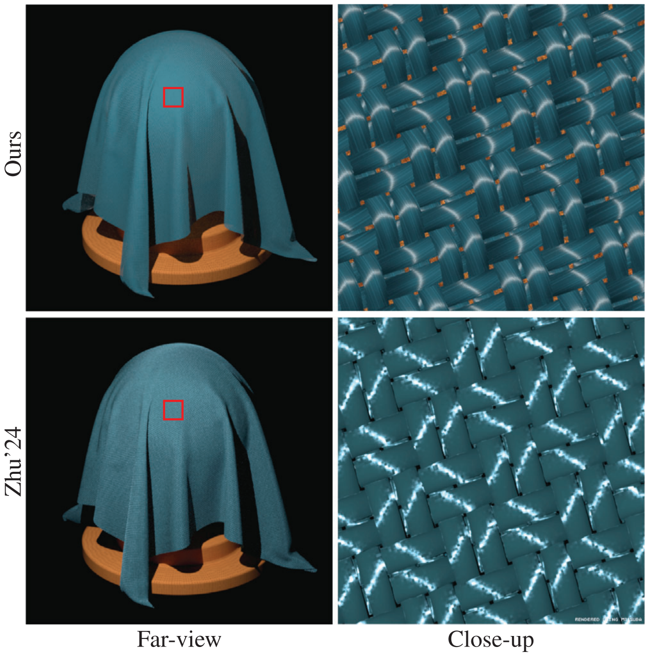


Figure 18: *Additional rendering. Comparison with the parallax mapping method proposed by [ZHB*24] for the basket pattern. Our approach is texture-free, whereas their model relies on high-resolution texture maps for the weave pattern and SAT to handle parallax. In close-up views, our method shows distinct advantages: (i) it better captures the high curvature of yarns, enhancing highlights and making underlying yarns visible, and (ii) it provides superior delta transmission, allowing the wooden sphere to be seen through the gaps between yarns.*

the other hand is texture-free and is able to incorporate fibre-level details and represent the close-up appearance more faithfully.

8. Discussions and Conclusion

Far-field integration. Our model currently depends on local shading details, including weave pattern geometry, inter-yarn interactions, yarn curvature, and fibre normals and tangents. Future work could explore efficient far-field integration methods, as seen in [KZP*24, ZHB*24], to aggregate appearance over a pixel footprint, $\mathcal{P}$. This approach would enable faster rendering in distant views while incorporating effects like parallax, inter-yarn shadowing and yarn-ply-fibre geometry.

Irregularities. Our model currently produces a very structured appearance in close-ups. To enhance realism and reduce the visibility of repetitive artefacts, we plan to introduce controlled noise to simulate irregularities and imperfections. This adjustment is left as future work where the yarns can vary in direction and thickness.

Fly-away fibres. Although our model supports fibre-level details, it currently excludes fly-away fibres, which contribute significantly to the fabric's sheen by creating a fuzzy appearance. While these fibres have traditionally been modelled explicitly, integrating them efficiently within a surface-based framework remains a challenge. An interesting direction for future research is to develop methods for incorporating fly-away fibres into our model efficiently.

Multi-resolution. Our model allows users to toggle parallax and shadowing effects based on scene requirements, which can add small computational overhead when activated. To address this, future work could focus on developing a multi-resolution approach that adapts the level of detail dynamically based on the pixel footprint. This would enable a smoother transition between levels of details and optimizing rendering times while maintaining visual fidelity in close-ups and avoiding aliasing in far view.

Conclusion. This paper introduced an appearance model for woven fabrics that balances high detail and computational efficiency by moving to the surface level. Using a Newtonian approach, our model dynamically generates parallax and shadowing effects from simple binary matrix inputs, eliminating the need for pre-computed textures and reducing storage demands. Our approach provides detailed visual features, including fibre details and incorporates a double-layer model to allow faithful light transmission. Validated against actual fabric samples, our model demonstrates accuracy in rendering fabrics from both distant and close-up perspectives while outperforming previous models.

Acknowledgements

We thank the reviewers for their valuable inputs. This research was partially funded by the University of Manchester Dean's Doctoral Scholarship Award.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[AMTF03] Adabala N., Magnenat-Thalmann N., Fei G.: Real-time rendering of woven clothes. In *Proceedings of the ACM symposium on Virtual reality software and technology (VRST '03)* (New York, NY, 2003) Association for Computing Machinery, pp. 41–47. ISBN: 1581135696. DOI: https://doi.org/10.1145/1008653.1008663.

[BAC*18] Burley B., Adler D., Chiang M. J.-Y., et al.: The design and evolution of disney's hyperion renderer. *ACM Transactions on Graphics (TOG) 37*, 3 (2018), 1–22.

[CFS*18] Christensen P., Fong J., Shade J., et al.: Renderman: An advanced path-tracing architecture for movie rendering. *ACM Transactions on Graphics (TOG) 37*, 3 (2018), 1–21.

[CWW24] Chen X., Wang L., Wang B.: Real-time neural woven fabric rendering. In *ACM SIGGRAPH 2024 Conference Papers* (New York, NY, 2024), Association for Computing Machinery, pp. 1–10.

[FHL*18] Fascione L., Hanika J., Leone M., et al.: Manuka: A batch-shading architecture for spectral path tracing in movie production. *ACM Transactions on Graphics (TOG) 37*, 3 (2018), 1–18.

[IM12] Irawan P., Marschner S.: Specular reflection from woven cloth. *ACM Transactions on Graphics 31*, 1 (2012), 1–20.

[JAM*10] Jakob W., Arbree A., Moon J. T., et al.: A radiative transfer framework for rendering materials with anisotropic structure. *ACM Transactions on Graphics 29*, 4 (2010), 1–13.

[JSR*22] Jakob W., Speierer S., Roussel N., et al.: Mitsuba 3 renderer, Version 3.1.1. https://mitsuba-renderer.org. 2022.

[JWH*22] Jin W., Wang B., Hasan M., et al.: Woven fabric capture from a single photo. In *SIGGRAPH Asia 2022 Conference Papers* (New York, NY, 2022), Association for Computing Machinery, pp. 1–8.

[KSZ*15] Khungurn P., Schroeder D., Zhao S., et al.: Matching real fabrics with micro-appearance models. *ACM Transactions on Graphics 35*, 1 (2015), 1–26.

[KTI*01] Kaneko T., Takahei T., Inami M., et al.: Detailed shape representation with parallax mapping. In *Proceedings of ICAT* (2001), vol. *2001*, pp. 205–208.

[Kuz21] Kuznetsov A.: Neumip: Multi-resolution neural materials. *ACM Transactions on Graphics (TOG) 40*, 4 (2021).

[KZP*24] Khattar A., Zhu J., Padovani E., et al.: A multi-scale yarn appearance model with fiber details. *arXiv preprint arXiv:2401.12724* (2024).

[LWS*18] Leaf J., Wu R., Schweickart E., et al.: Interactive design of yarn-level cloth patterns. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2018) 37*, 6 (November 2018), 1–15. DOI: https://doi.org/10.1145/3272127.3275105.

[MGJZ21] Montazeri Z., Gammelmark S. B., Jensen H. W., Zhao S.: A practical ply-based appearance modeling for knitted fabrics. In *Proceedings of Eurographics Symposium on Rendering 2021* (2021).

[MGZJ20] Montazeri Z., Gammelmark S. B., Zhao S., Jensen H. W.: A practical ply-based appearance model of woven fabrics. *ACM Transactions on Graphics 39*, 6 (2020), 1–13.

[OP05] Oliveira M. M., Policarpo F.: An efficient representation for surface details. *Instituto de Informatica UFRGS* (2005).

[PBFJ05] Porumbescu S. D., Budge B., Feng L., Joy K. I.: Shell maps. *ACM Transactions on Graphics (TOG) 24*, 3 (2005), 626–633.

[Rit06] Ritsche N.: Real-time shell space rendering of volumetric geometry. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia* (New York, NY, 2006), Association for Computing Machinery, pp. 265–274.

[SBDDJ13] Sadeghi I., Bisker O., De Deken J., Jensen H. W.: A practical microcylinder appearance model for cloth rendering. *ACM Transactions on Graphics 32*, 2 (2013), 1–12.

[SM24] Soh G. Y., Montazeri Z.: Neural appearance model for cloth rendering. *Computer Graphics Forum 43*, 4 (2024), e15156.

[Tat06] Tatarchuk N.: Dynamic parallax occlusion mapping with approximate soft shadows. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games* (New York, NY, 2006), Association for Computing Machinery, pp. 63–69.

[VSJ21] Vicini D., Speierer S., Jakob W.: Path replay backpropagation: Differentiating light paths using constant memory and linear time. *ACM Transactions on Graphics (TOG) 40*, 4 (2021), 1–14.

[WTS*05] Wang J., Tong X., Snyder J., et al.: Capturing and rendering geometry details for BTF-mapped surfaces. *The Visual Computer 21* (2005), 559–568.

[ZHB*24] Zhu J., Hery C., Bode L., et al.: A realistic multi-scale surface-based cloth appearance model. In *ACM SIGGRAPH 2024 Conference Papers* (New York, NY, 2024), Association for Computing Machinery, pp. 1–10.

[ZJA*23] Zhu J., Jarabo A., Aliaga C., et al.: A realistic surface-based cloth rendering model. In *ACM SIGGRAPH 2023 Conference Proceedings* (New York, NY, 2023), Association for Computing Machinery, pp. 1–9.

[ZJMB11] Zhao S., Jakob W., Marschner S., Bala K.: Building volumetric appearance models of fabric using micro CT imaging. *ACM Transactions on Graphics 30*, 4 (2011), 1–10.

[ZJMB12] Zhao S., Jakob W., Marschner S., Bala K.: Structure-aware synthesis for predictive woven fabric appearance. *ACM Transactions on Graphics 31*, 4 (2012), 1–10.

[ZLB16] Zhao S., Luan F., Bala K.: Fitting procedural yarn models for realistic cloth rendering. *ACM Transactions on Graphics 35*, 4 (2016), 1–11.

[ZMA*23] Zhu J., Montazeri Z., Aubry J., et al.: A practical and hierarchical yarn-based shading model for cloth. *Computer Graphics Forum 42*, 4 (2023), e14894.

[ZXW19] Zhu J., Xu Y., Wang L.: A stationary SVBRDF material modeling method based on discrete microsurface. *Computer Graphics Forum 38*, 7 (2019), 745–754.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supporting Information

Supplemental Video 1