DOI: 10.1111/cgf.70356

EUROGRAPHICS 2026 / B. Masia and J. Thies
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 2

A Texture-Free Multi-Scale Model for Surface-Based Rendering of Knitted Fabrics

Apoorv Khattar[1], Jean-Marie Aubry[2], Ling-Qi Yan[3] and Zahra Montazeri[1]

[1]University of Manchester, United Kingdom
{apoorv.khattar, zahra.montazeri} @manchester.ac.uk
[2]WETAFX
jaubry@wetafx.co.nz
[3]Mohamed bin Zayed University of Artificial Intelligence
lingqi.yan@mbzuai.ac.ae

Abstract

Knitted fabrics present unique challenges for realistic rendering due to their complicated structure and scale-dependent appearance. Existing methods typically rely on explicit yarn geometry, which is computationally complex, or texture-based representations that require heavy storage and precomputed maps. In this paper, we introduce the first texture-free, surface-based appearance model for knitted fabrics, in which stitches are represented parametrically as thick curves and mapped directly onto fabric meshes. This avoids explicit yarn or fiber geometry, yet preserves the characteristic 3D look of yarn-based models. Unlike prior surface-based approaches, our method produces realistic volumetric effects such as depth, parallax, and silhouette preservation. To achieve this, we propose a curvature-aware parallax mapping technique that ensures coherent appearance at grazing angles. Furthermore, we extend the appearance model to a multi-scale formulation that aggregates geometry and visibility over texture footprints and adjusts roughness parameters for stable far-field rendering. Our model combines the efficiency and simplicity of surface-based methods with the volumetric realism of fiber-based models, reproducing characteristic knit effects such as 3D stitch structure in a multi-scale manner without the complexity or storage cost of texture-based approaches.

Keywords: rendering, ray tracing, reflectance & shading models

CCS Concepts

• **Computing methodologies** → *Reflectance modeling; Ray Tracing; Rendering;*

1. Introduction

Fabrics play a central role in creating realistic digital content, spanning applications in visual effects, fashion design, gaming, and virtual reality. Their rich appearance arises from complex structures across multiple scales, where the interplay of yarns and fibers produces subtle phenomena such as fuzziness, sheen, and translucency. While woven fabrics have been well studied and modeled using surface-based methods, knitted fabrics remain significantly more challenging. This is due to their highly curved, interlocking stitch geometry and their often substantial thickness, which complicates both modeling and rendering.

The characteristics of knitted cloth are strongly tied to visual expectations: viewers rarely accept knitted fabrics rendered using simple surface-based models, which flatten all loops into a plane regardless of shading quality. Conversely, explicit yarn- or stitch-level models [YKJM12; HSW*25; MGJZ21] can reproduce fine detail but suffer from costly curve intersections, high memory overhead, and limited scalability for production rendering. Texture-based methods alleviate some of these costs but require precomputed maps, introducing storage overhead and reducing generality. More recent texture-free models for woven fabrics [ZJA*23; KZYM25] offer efficient parametric surface-based representations using sinusoidal functions, but their formulations do not naturally extend to the looped geometry and discontinuous heightfields characteristic of knitted fabrics.

A key difficulty lies in balancing efficiency with realism. On the one hand, surface-based approaches are lightweight, but when applied to knitted fabrics they flatten loops into a plane, producing results that viewers often perceive as unconvincing. On the other hand, explicitly modeling every yarn as a 3D curve delivers realism but at prohibitive computational cost, particularly for large fabric swatches or animations.

This gap motivates our work. We aim to develop a *texture-free, surface-based model for knitted fabrics* that is efficient, general, and capable of reproducing realistic appearance both at close range and from a distance. Our model inherits the advantages

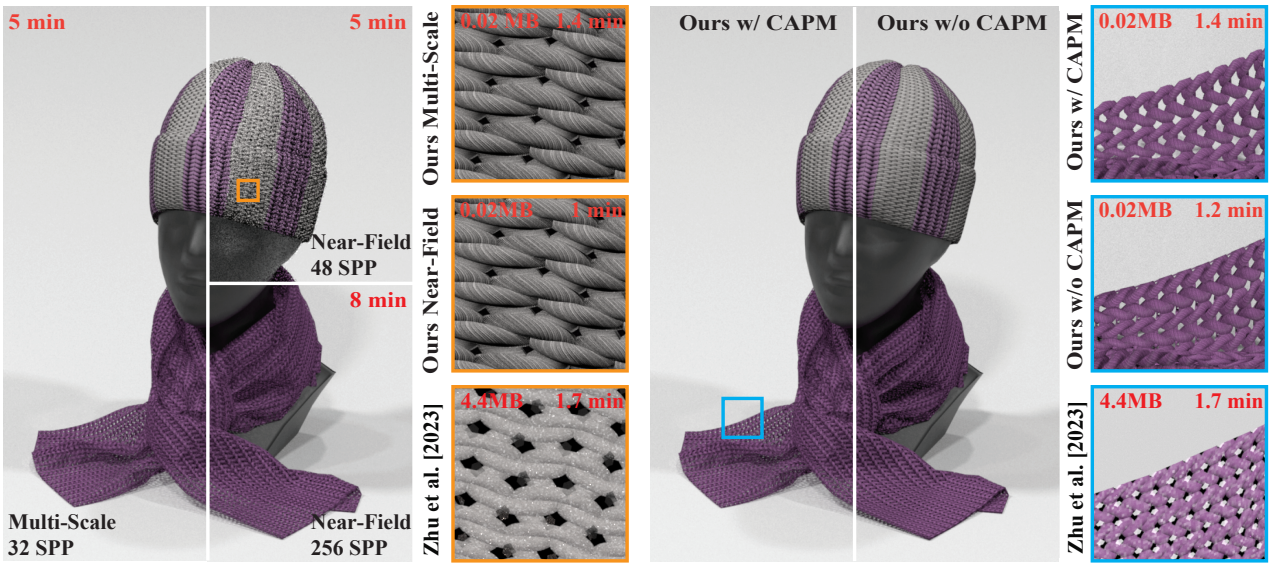

Figure 1: *In this scene, we have two fabrics surfaces: a scarf with knit yarn stitch and a hat with 4-alternating knits and purls. On the left we compare our multi-scale model against our near-field model. On the right we have our model without our Curvature-Aware Parallax Mapping (CAPM) to show how parallax mapping improves the delta transmission by changing the inter-yarn spacing and also makes shadows smoother. We further compare our model against previous surface-based model by Zhu et al. [ZMA*23] at two different zoom levels. In the blue box, our model accounts for the curvature of the mesh to update the silhouette unlike Zhu's. In the orange box, we show the efficiency of our multi-scale in preserving the micro-geometry details of yarns. Zhu's method also requires precomputation of the normal+tangent maps and visibility while ours only requires a matrix representation of the knit pattern and just 0.02MB memory for the 1D textures of yarn micro-geometries, saving 99.5% of memory time, offering 1.2 to 1.4 times faster rendering while preserving the micro-geometry details.*

of surface-based approaches—simplicity and efficiency—while avoiding heavy texture storage and achieving a volumetric, yarn-like 3D appearance comparable to fiber-based models. Concretely, we extend the idea of parametric yarn representations from woven fabrics to knitted stitches through a flexible stitch mapping scheme based on sine–cosine functions. To address the challenges of discontinuous heightfields and silhouette preservation in knitted loops, we introduce a silhouette-aware parallax mapping technique. Finally, to ensure consistent rendering quality across scales, we propose a multi-scale appearance model that aggregates geometry and visibility over texture footprints and adapts roughness parameters accordingly.

Our contributions are as follows:

- A **texture-free parametric representation** of knitted stitches and patterns, mapped directly onto fabric surfaces without explicit yarn geometry or precomputed textures.
- A **curvature-aware parallax mapping (CAPM) method** that resolves heightfield discontinuities and preserves the volumetric appearance of knitted loops at grazing angles.
- A **multi-scale appearance model** that aggregates geometry and visibility for stable far-field rendering, while adaptively refining roughness for accurate near-field detail.

2. Related Work

2.1. Curve-/Volume-Based Models

Fiber-level and yarn-level methods explicitly model the microgeometry of textiles, delivering high fidelity in close-up views at the cost of substantial computation and memory. Fiber-scale fitting pipelines [KSZ*15] and procedural yarn-fitting methods based on CT data [ZLB16] both aim to capture detailed micro-appearance, but remain computationally demanding. They were later extended to a mechanics-aware model [MXF*21]. Structure-aware volumetric synthesis for woven cloth predicts appearance from design specifications, but remains expensive at render time [ZJMB11; SZZ12]. Lopez et al. [LMCO17] address some of these challenges in their recent work by proposing a fast voxelization pipeline to model fabrics at sub-voxel level. At the other end, GPU rasterization at fiber-level achieves real-time rates by aggressive approximations, trading physical accuracy for speed [WY17; HSW*25]. For knitted fabrics specifically, yarn-explicit modeling provides geometric realism but requires costly ray–curve intersections and complex BVHs, which scales poorly to production scenes.

2.2. Aggregated and Data-Driven Models

To reduce complexity while retaining fidelity, aggregated models group microgeometry into higher-level primitives. Ply-based appearance models collapse hundreds of fibers into twisted plies, re-

producing fuzz, sheen, and body scattering with far lower cost than fiber-explicit methods [MGZJ20]; Montazeri et al. extend this idea from woven to knitted fabrics [MGJZ21]. Follow-up works were later proposed aggregation up to the yarn level [ZMA*23; SM24], achieving efficient shading without resolving sub-ply detail. More recently, multi-scale yarn appearance models synthesize fiber detail into compact analytic forms, yet still rely on explicit yarn layout [KZP*24].

Data-driven approaches such as BTFs encode view- and light-dependent texture appearance directly from measurements or simulations [DvGNK99]. A suitable alternative is to compress the BTF function using neural architectures. Rainer et al. [RJGW19] introduced an autoencoder-based approach that compresses BTF data on a per-texel basis and extends it to support multiple materials within a shared latent space. NeuMIP [Kuz21] [KWM*22] represents BTFs as a neural texture pyramid, enabling multi-scale rendering while preserving texture silhouettes. However, this approach is material-specific and still requires large BTF datasets for training. To address this limitation, Rodrigues et al. [RKLG23] propose extrapolating neural BTF representations from sparse input data. While faithful, BTFs and Neural BTFs require precomputation, and offer limited editability, making them less suitable for interactive pipelines.

2.3. Surface-Based Models

Surface-based cloth models replace explicit microgeometry with analytic shading on the base surface mesh. The widely used microcylinder model [SBDJ13] derives an empirical thread shading from measurements and assembles cloth appearance at the weave level, but only accounts for woven fabric. For woven fabrics, recent parametric and texture-free methods map sinusoidal yarn profiles onto surfaces and introduce analytic attenuation/parallax components; these achieve strong realism at low storage but again have been limited to woven, not knitted, structures [KZYM25]. Texture-based surface models can also attain good visual quality but typically require precomputed normal/tangent maps and visibility, constraining generality and editability [ZJA*23; ZHB*24].

3. Background & Motivation

3.1. Background

Fabrics are broadly categorized into two families: woven and knitted. Woven fabrics consist of a large number of yarns running in perpendicular directions, known as warps and wefts. Their patterns are often represented as binary matrices, where 1 indicates warp-over-weft and 0 indicates weft-over-warp. Knitted fabrics, by contrast, are constructed from only a few yarns looped together in series. The horizontal direction of loops is called the *course*, and the vertical stacking direction is called the *wale*. Because of their interconnected loops, knitted fabrics are inherently more stretchable and flexible. Prior works (such as [LWS*18; YKJM12]) have formalized knit patterns as rectangular text matrices, similar to binary encodings for woven patterns. These string encodings can also be extended to numerical encodings, as illustrated in Table 1. At the microscopic level, yarns exhibit a hierarchical structure of plies and fibers. Fibers twist to form plies, and multiple plies twist to form a yarn. This twisted construction produces non-cylindrical yarn shapes, introducing variations in normals and tangents that generate complex specular highlights. Additionally, stray "flyaway" fibers create fuzziness in close-up views and sharp grazing-angle highlights known as the *sheen* effect in far-field rendering.

Table 1: ***Numerical encoding of different stitches present in knitted fabrics taken from Leaf et al. [LWS*18]****. Each stitch can be represented as a number to create the desired knit pattern represented as a rectangular matrix*

Stitch Type	Encoding	Visualization
Knit	0	
Purl	1	
Front Stitch	2	
Back Stitch	3	

Building on this foundation, our work draws inspiration from the recent texture-free, surface-based model for woven fabrics proposed by Khattar et al. [KZYM25]. In this approach, woven patterns are represented as parametric sinusoidal curves corresponding to warp and weft yarns, mapped directly onto fabric meshes without explicit texture maps. Curvature of yarns is captured using parallax offsetting, while reflection and transmission are modeled with high accuracy through a double-layer attenuation function that accounts for overlapping yarns, enabling realistic delta transmission and inter-yarn shadowing.

Although highly effective for woven fabrics, this framework faces several limitations. First, it is restricted to woven patterns and does not extend to knitted loops. Second, it remains a near-field model, requiring a large number of samples per pixel to converge in far-view renderings. Finally, while the parallax formulation adjusts appearance according to yarn curvature and the underlying mesh, it does not capture silhouette changes introduced by yarn geometry. To set the stage for our method, we next summarize the key components of this near-field model.

3.2. Near-Field Model

This model [KZYM25], represents the warp and weft yarns as parametric sinusoids running along the U and V directions, respectively. The underlying ply and fiber geometries are represented as 1D textures wrapped around the implicit yarn cylinder. The appearance of the yarns as a result of the reflective and transmissive properties of fabrics is separated into two independent terms. Reflection, at the entry point x, is expressed as a combination of a specular term, represented as a rough dielectric, f_x^S, and a diffuse term, represented as a Lommel-Seeliger (LS) [JMLH01]. The transmission, at exit point y, is also represented using a rough dielectric, f_y^S and a Lambertian diffuse term. The overlapping nature of warp and weft creates two layers, thus they further propose a *double-layer attenuation* term to

account for the loss of energy when passing through areas where the warp and weft overlap. Finally they define 1D textures for ply and fiber geometries that are wrapped around the yarn cylinders to make the micro-geometry details present within the yarn, affecting the specular highlights and shadowing.

The complete appearance model can be expressed as,

$$f(\omega_i,\omega_o) = \begin{cases} f_x^S(\omega_i,\omega_o) + G_{p,f}(x)\; r\; B\; \mathrm{LS}(\omega_i,\omega_o)\; k_x^B & (\omega_i.\omega_o > 0) \\ \tau(x,y)\left(f_y^S(\omega_i,\omega_o) + G_{p,f}(y)(1-r)\frac{B}{\pi}k_y^B\right) & (\omega_i.\omega_o < 0), \end{cases} \quad (1)$$

where $\tau(x,y)$ is the double-layer attenuation accounting for the loss of through single and overlapping yarns, B is the energy assigned to the light that undergoes multiple scattering, r is the reflection portion of the body energy, $G_{p,f}(.)$ are the inter-ply and inter-fiber shadows fetched from the 1D textures and k_x^B and k_y^B are the diffuse reflection and transmission color, respectively.

3.3. Motivation

Several surface-based models exist for woven fabrics [ZZW*22; ZJA*23; KZYM25], but knitted fabrics remain difficult to model at the surface level. Recent works [MGJZ21; HSW*25] address this by explicitly constructing curves along the fabric surface and computing ray–curve intersections, which provide realism but at very high computational cost. Our goal is a *texture-free, surface-based model for knitted fabrics* that avoids explicit yarn/ply/fiber geometry or heavy texture maps, while still capturing volumetric characteristic appearance across scales.

Our approach proceeds in three steps.

First, explicitly modeling every yarn or fiber is infeasible, as even modest-scale knitted garments contain millions of individual fibers. On the other hand, purely surface-based representations cannot capture the mutual occlusions of yarns and the variations in their local geometry. We therefore represent yarns in a parametric and implicit form, providing a compact, storage-free description of the knit structure. Following prior works [LWS*18], we map looped yarns directly to the fabric's UV layout, identifying the course and wale directions and the corresponding stitch types. Then, sub-yarn details such as ply and fiber geometries are embedded as 1D profiles, and can be incorporated using existing techniques [MGZJ20; MGJZ21]. Section 4.1 details how knitted yarns are expressed parametrically as sinusoidal functions and tiled across the surface.

Second, a purely surface-based mapping lacks volumetric depth, making knitted loops appear flat. To recover a 3D look, our insight is that a surface-based representation can act as a virtual "shell" above the yarns, capturing volumetric structure without explicit geometry. We introduce a silhouette-aware parallax mapping technique that shifts texture coordinates according to local view direction, producing coherent silhouettes and volumetric appearance even at grazing angles. In this formulation, rays intersect only the base mesh, while local intersections are resolved against a lightweight implicit representation of the micro-geometry. This yields a fast, intersection-friendly approach that avoids the costly dynamic realization of fiber models [LZB17] or explicit ray–curve intersections [MGZJ20]. Unlike displacement mapping, our method (i) is not restricted to a single-valued heightfield, since knitted loops exhibit multiple overlapping heights, and (ii) represents yarns implicitly rather than through explicit displacement. Section 4.2 provides details on how this technique accounts for the geometry and silhouette variations introduced by interlooping knitted yarns.

Finally, rendering ultimately resolves shading at the pixel level, where rays through a single footprint may hit widely varying loop positions. Instead of leaving this integration entirely to the renderer, we aggregate geometry and visibility analytically across the texture footprint, following principles from previous multi-scale appearance models. This stabilizes far-field rendering and adaptively adjusts roughness parameters, ensuring both efficiency and accuracy across scales. In Section 5.1, we extend the near-field appearance model into a multi-scale formulation that converges with fewer ray samples per pixel.

4. Modelling the Geometry

As described in Section 3.3, our method begins with ray–mesh intersection on the base fabric surface. Using the matrix representation of the knit pattern, we implicitly define yarn placement within the mesh's UV layout, while micro-geometry details such as plies and fibers are added using 1D textures, following the work of Montazeri et al. [MGZJ20].

In Section 4.1, we describe how the 3D yarn geometry is defined on the surface plane: specifically, how the matrix representation of a knit pattern is tiled across the mesh UV domain, and how parametric functions are used to represent individual stitches.

In Section 4.2, we address the second challenge: performing intersections against the ray in the presence of layered heightfields. Here we introduce our curvature-aware parallax mapping technique, which enables consistent local intersections with the implicit yarn geometry while preserving silhouettes and volumetric appearance.

4.1. Knit Pattern Mapping

Mapping the knit pattern using the UV-values of the fabric mesh can be broken down into two steps: (i) given the input knit pattern, define a tiling schematic that allows pattern to be repeated over the fabric surface (ii) interpreting the yarn type associated with each cell in the pattern and map the individual stitches to add implicit yarn geometry without explicit generation of the knit curves.

Defining the Knit Pattern. The consecutive rows of the matrix representation of the knit patterns indicate the type of knit yarn that is interconnected with each other. Therefore, we can split the alternating rows of the pattern. Dealing with a single type of knitted curve in each cell simplifies the mapping of the curve function without worrying about which one of the two curves is looping inward or outward. At the end, the alternating rows can be merged and since the curves have the correct depth, they stack over one another seamlessly. Given this insight, we split the pattern matrix

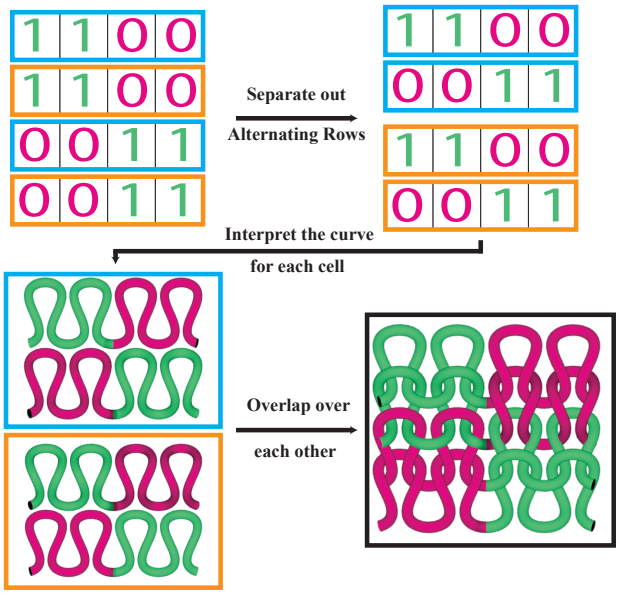

Figure 2: ***Illustrative example of our knit pattern mapping approach***. *For a given input pattern, we separately map the alternating rows and then overlap them to form the interconnected loops of knit fabrics.*

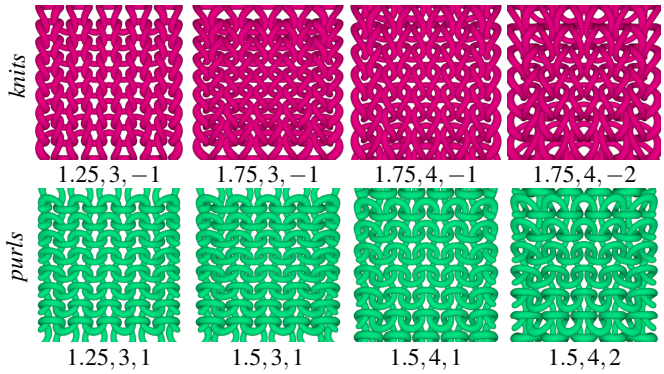

Figure 3: ***Varying parameters*** a,h,d ***for parametric knitted curves***. *Using the parametric equation, we can easily adjust the configuration of the knitted curves over the surface. Increasing* a *(as shown along column I and II), makes the yarns more round, changing* h *affects the vertical height of each individual yarns and changing* d *affects how the yarns loop in and out of the XY plane.*

into two sub-matrices. Each comprising of the alternating rows and perform pattern mapping separately for the two halves. Finally, we overlap them to create the interconnected loops with yarns going inwards/outwards based on their depth. This idea is illustrated in Figure 2 for a 4×4 pattern which comprises *knits* and *purls* represented as 0 and 1, respectively.

Interpreting Pattern Cell. Having separated the alternating rows, we now shift our focus to each cell within the sub-matrix to map the individual stitches using the UV layout. Building on Khattar et al. [KZYM25] that models yarns for woven fabrics as sine curves, we parameterize knit stitches using a combination of sine and cosine curves. In the following section, we consider mapping of *knit* stitch on to a UV cell but our approach can be generalized to other types of stitches as well.

As shown by Crane et al. [Cra23], the common *knit* or *purl* stitch can be represented parametrically as,

$$Y(t) = \begin{cases} x(t) = t + a\sin(2t) \\ y(t) = h\cos(t) \\ z(t) = d\cos(2t), \end{cases} \tag{2}$$

where $t \in [-\pi, \pi]$ and a,h,d are user-controlled parameters, $a > 0$ controls the roundness of the yarns, $h > 0$ controls the height and $d \neq 0$ controls the yarns bend. The yarn curve can be added assuming $x(t)$ varies along the U values, $y(t)$ varies along the V values and $z(t)$ is the depth / height of the parametric curve. Figure 3 visualizes the explicit yarn curves for varying values of a,h,d to model different geometric characteristics of knitted fabrics.

The mapping of the stitch geometry can be seen as intersecting a ray, $R(s)$ with origin as $(u,v,|d|)$ and direction along the mesh normal with $Y(t)$. To avoid explicit generation of the stitch yarn and creation of a bounding volume hierarchy, we make use of the parametric equation of the stitch yarn to solve the ray-curve intersection analytically in $O(1)$ time. We solve an analytic equation to find the point on the ray that is closest to the curve $Y(t)$ and for all points with distance $\leq R$ (where R is the radius of the *knit* stitch) we define the geometry of cylindrical yarns. The parametric equation is given by,

$$\begin{aligned} t,s &= \underset{t,s}{\operatorname{argmin}}\ D(t,s) = \|Y(t) - R(s)\| \\ &= \underset{t,s}{\operatorname{argmin}}\ \|(x_Y(t), y_Y(t), z_Y(t)) - (x_R(s), y_R(s), z_R(s))\| \end{aligned} \tag{3}$$

where (t,s) can be found by determining the roots of the first-order derivative of $D(t,s)$, which requires $O(1)$ time, given by,

$$\begin{aligned} \frac{\delta D'(t,s)}{\delta s} &= 0 \\ &= (x_Y(t) - x_R(s))\, x'_Y(s) \\ &\quad + (y_Y(t) - y_R(s))\, y'_Y(s) \\ &\quad + (z_Y(t) - z_R(s))\, z'_Y(s) \\ \frac{\delta D'(t,s)}{\delta t} &= 0 \\ &= (x_Y(t) - x_R(s))\, x'_R(s) \\ &\quad + (y_Y(t) - y_R(s))\, y'_R(s) \\ &\quad + (z_Y(t) - z_R(s))\, z'_R(s) \end{aligned} \tag{4}$$

and finally $R(s)$ can be used as the implicit height of the curve.

Substituting the cosines and sinusoids associated with $Y(t)$ will make Equation 4 hard to solve analytically. Thus, we approximate them as quadratic functions. In the domain $[0,\ \pi]$, the sine curve

can be represented as,

$$\sin(t) \approx \frac{4t}{\pi^2}(\pi - t) \qquad \frac{4t}{\pi} - \frac{4t^2}{\pi}. \tag{5}$$

For $sin(2t)$, we split the domain into two parts $[0, \frac{\pi}{2})$ and $[\frac{\pi}{2}, \pi]$. t remains unchanged in the first domain and for the second domain, we substitute $t = t - \frac{\pi}{2}$ and take the negative of the final output, as shown below,

$$\sin(2t) \approx \begin{cases} \frac{4t}{\pi} - \frac{4t^2}{\pi^2} & 0 \le t < \frac{\pi}{2} \\ \frac{4t^2}{\pi^2} - \frac{12t}{\pi} + 8 & \frac{\pi}{2} \le t \le pi. \end{cases} \tag{6}$$

Similarly, $cos(t) = sin(\frac{\pi}{2} - t)$, thus the cosine function can be expressed as,

$$\cos(t) \approx \begin{cases} \frac{4t^2}{\pi^2} - \frac{8t}{\pi} + 3 & 0 \le t < \frac{\pi}{2} \\ 1 - \frac{4t^2}{\pi^2} & \frac{\pi}{2} \le t \le \pi. \end{cases} \tag{7}$$

Finally, this can be extended to approximate $cos(2t)$ as,

$$\cos(2t) \approx \begin{cases} \frac{16t^2}{\pi^2} + 1 & 0 \le t < \frac{\pi}{4} \\ \frac{16t^2}{\pi^2} - \frac{16t}{\pi} + 3 & \frac{\pi}{4} \le t < \frac{\pi}{2} \\ \frac{16t^2}{\pi^2} - \frac{16t}{\pi} + 3 & \frac{\pi}{2} \le t < \frac{3\pi}{4} \\ \frac{16t^2}{\pi^2} - \frac{32t}{\pi} - 15 & \frac{3\pi}{4} \le t \le \pi \end{cases} \tag{8}$$

By substituting the quadratic approximations of $Y(t)$, Equation 4 reduces to the problem of computing the intersection between a ray and a quadratic curve. Setting $\frac{\delta D'(t,s)}{\delta s} = 0$ allows s to be expressed in terms of t which can be substituted in $\frac{\delta D'(t,s)}{\delta t} = 0$, to return a cubic equation in t. The roots of a cubic equation can be found analytically using Cardano's Formula [Nic93]. This enables ray–knitted-curve intersections to be computed in $O(1)$ time, avoiding the need to explicitly generate sample points or construct a bounding volume hierarchy.

4.2. Curvature-Aware Parallax Mapping (CAPM)

Mapping yarn geometry along the mesh normal creates an overall flat appearance even though the parametric yarns are not flat. The curvature of the yarns affects the shape, shadowing and inter-yarn spacing for knit fabrics which affects the resulting appearance in close-up and far-view. Additionally, the knitted yarns affect silhouette of the mesh, making it more bumpy, rather than smooth. To this end, we propose Curvature-Aware Parallax Mapping (CAPM) to account for the changing shape of the yarns and mesh silhouette.

Traditional parallax mapping [Tat06] aims to determine the intersection point of the incoming ray with the heightfield texture of the micro-geometry and use the offset UV position for mapping the textures relating to the geometry and appearance of the underlying micro-geometry. However, it has two major drawbacks: (i) parallax mapping requires a continuous heightfield texture. For a stitch yarn, this assumption is not true due to overlapping nature of yarns which creates two separate levels of height but the heightfield texture only looks at the top level. (ii) parallax mapping assumes that the mesh is locally flat which is not true for meshes with high curvature. Due to these assumption the silhouette of the base mesh remains unchanged. These drawbacks are illustrated in Figure 4. This is often resolved by either creating heightfield and normal texture map for each layer separately [PO06], which is increases the storage requirements or using an alternative approach called shell mapping that creates a prism-like geometry, [PBFJ05; XCL*25] but is more computationally expensive.

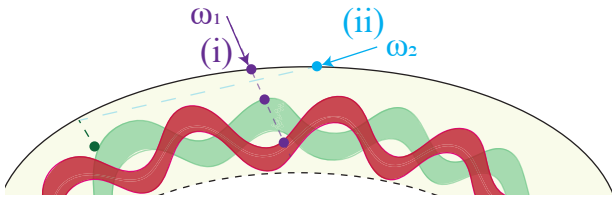

Figure 4: ***Failure cases using traditional parallax mapping techniques****. (i) For ray sample ω_1, parallax mapping accounts only for the uppermost layer of the heightfield and neglects the underlying multi-layer structure of the texture. This limitation is particularly noticeable in knitted fabrics, where inter-looping yarns form two distinct height layers. (ii) As shown for ω_2, on a curved surface, it is possible for the ray to hit the mesh but never pass through the underlying micro-geometry. Parallax mapping assumes the surface to be locally flat thus silhouette of the base mesh remains unchanged.*

Taking inspiration from the Newtonian parallax mapping approach proposed by Khattar et al. [KZYM25] for woven fabrics, we design our approach to work for layered heightfields associated with knit fabrics. By solving the analytic equation 3, we obtain the implicit height, s for all UV-values and using that as the offset value, we update the UV-map similar to theirs. However, unlike woven yarns which are simple sinusoids, the stitch yarns require a more precise estimate of the actual depth, thus after offsetting the UV-map once, we take small iterative steps along the ray, offsetting the UV-map with each step. This resolves the changing shape of the yarns but does not account change in silhouette.

Chen et al. [CC08] proposed an alternative to shell mapping, that involves bending the ray following the curvature of the mesh. We extend this technique to our problem to change the silhouette of the mesh while tiling the knit pattern. For any given mesh, we first generate the normal and tangent maps only for the base mesh. Please note: our model only requires the normal and tangent maps of the base mesh and the underlying geometry is still texture-free, represented parametrically, only depending on the input pattern matrix. At each step of the parallax offset, we first retrieve the mesh normals and tangents for the offset P^i_{UV} and transform the ray ω_i into the local space ω_{local} defined by the new normals and tangents. If $\omega_{local}.z$ is less than zero i.e. the ray is curving away then the hit-

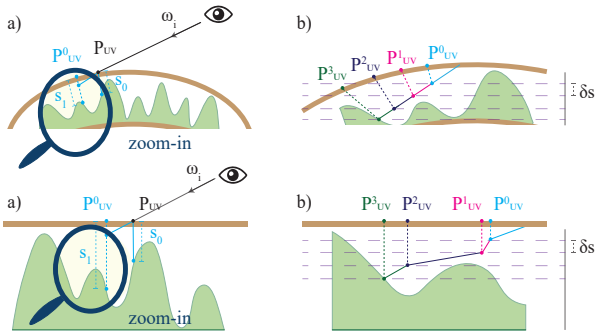

Figure 5: ***Traditional parallax mapping v/s curvature-aware parallax mapping***. *Parallax mapping assumes the surface to be locally planar; as a result, it advances along the ray by offsetting the UV coordinates using the object frame defined at the initial hit point, P_{UV}, without considering the curvature of the underlying mesh. In contrast, CAPM samples the object normal and tangent maps using a different object frame at each step, bending the ray to follow the mesh curvature while keeping the base surface flat, and does so without additional computational cost. Refer to Algorithm 1 for more details.*

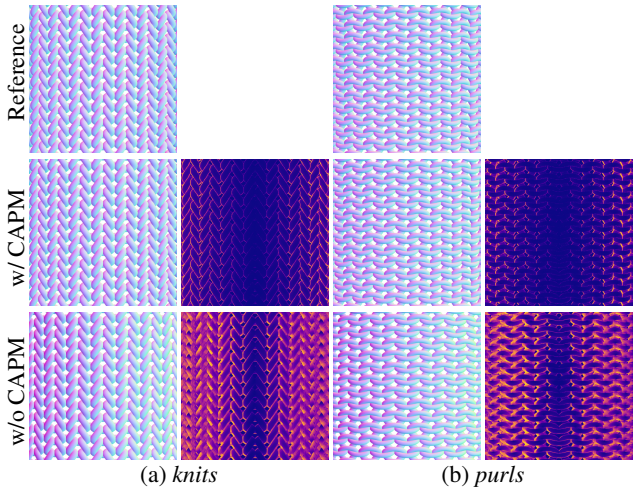

Figure 6: ***Normal map comparison over a cylinder surface***. *We compare the normals obtained after pattern mapping with and without CAPM against explicit curve normals as reference obtained by generating the curve along a cylinder surface for knits and purls . CAPM achieves a close match to the explicit yarn geometry while taking into account the changing curvature.*

point is declared invalid. Otherwise, P_{UV}^{i} is offset and as discussed above we solve the analytic equation until we find the intersection with the parametric knitted yarns. This is further illustrated in Figure 5.

Figure 6 provides a comparison between normals obtained after complete pattern mapping with and without parallax correction against normal maps of explicit knit yarn generated along the cylinder surface. The changing curvature of the yarns alongside the base surface, affects the shape of the geometry of the yarns and the inter-yarn spacing.

More information about the implementation details can be found in Section 6.1

5. Modelling the Appearance

After mapping the parametric knit curves, the updated mesh geometry closely resembles the geometry of the explicitly generated curves. Thus, the near-field proposed by Khattar et al. [KZYM25], as explained in Section 3.1, can be used to model the appearance of the fabric. This near-field appearance model is suitable for close-up renderings. However, as the camera moves further back, each ray can encompass not only the micro-structures present within a yarn but also multiple yarns or multiple blocks of the repeating pattern. [ZJA*23; ZHB*24] rely on texture-based aggregation techniques that present two problems: the need for explicitly generated texture maps for the given pattern and repeated querying of the texture to reduce the noise.

Our focus in this work is on a texture-free model and to this end, we propose a new multi-scale appearance model that leverages our pattern mapping approach (Section 4.1) to aggregate the underlying yarn, ply and fiber geometry over the ray footprint and then updates the roughness parameters to account for accurate shape and size of specular highlights.

5.1. Multi-scale Model

Having determined the coverage of a given ray sample in the UV-space [Ige99], also known as the texture footprint $\mathcal{T}$, we perform aggregation of the underlying geometry over $\mathcal{T}$ by generating implicit rays originating from a point sampled along $\mathcal{T}$ on the fly. However, as shown by Khattar et al. [KZP*24], simple geometric aggregation can lead to incorrect appearance as the some appearance parameters like roughness need to be updated. To account for the correct roughness associated with aggregated geometry, we rely on LEAN Mapping [OB10] to find the change in roughness due to the geometric aggregation while keeping the "texture" of our yarns completely implicit. Another key aspect of cloth presented due to the looping of yarns in and out is that the light can simply pass through tiny holes in the fabric, also known as delta transmission [ZJA*23].

Aggregation of Geometry and Visibility. The texture footprint defines the coverage in the UV-space by estimating $\frac{\partial u}{\partial x}$ and $\frac{\partial v}{\partial y}$, which are the changes in U-values and V-values for a small offset along the screen directions.

Similar to aggregating geometry, we also keep track of the visibility of each sample i.e. whether the sample is lying on a yarn or is simply passing through. We compute the average of these values to get a probabilistic visibility, p_{delta} for a given hit point on the fabric mesh. This map is used in conjunction with the sampled direction and sampled weight for the outgoing ray from the near-field model (Section 3.2). When $p_{delta} \to 0$, the sampled direction is used and

attenuated accordingly and when $p_{delta} \rightarrow 1$, the ray direction is unchanged to allow it to pass through without attenuation.

Updating the Appearance Parameters. As discussed in LEAN Mapping [OB10], simply aggregating the geometry / normals over $\mathcal{T}$, gives the correct position of where the specular highlight should appear but the shape of it is controlled by the underlying normal distribution function. Beckmann Distribution is commonly used to express microfacet normals and is represented as,

$$\frac{1}{2\pi\sqrt{|\Sigma|}}\exp\left(-1/2\ (\tilde{h_n}-\tilde{b_n})^T\ \Sigma^{-1}\ (\tilde{h_n}-\tilde{b_n})\right), \tag{9}$$

where $\tilde{h_n}$ and $\tilde{b_n}$ are the half vector and bump normal projected onto the $z=1$ plane. The challenge is to recover the co-variance matrix defined over $\mathcal{T}$ which can be done as follows,

$$\Sigma = \begin{bmatrix} \frac{1}{n}\Sigma(\tilde{b_n}.x^2) - (\Sigma\tilde{b_n}.x)^2 & \frac{1}{n}\Sigma(\tilde{b_n}.x\ \tilde{b_n}.y) - \Sigma\tilde{b_n}.x\ \Sigma\tilde{b_n}.y \\ \frac{1}{n}\Sigma(\tilde{b_n}.x\ \tilde{b_n}.y) - \Sigma\tilde{b_n}.x\ \Sigma\tilde{b_n}.y & \frac{1}{n}\Sigma(\tilde{b_n}.y^2) - (\Sigma\tilde{b_n}.y)^2 \end{bmatrix}, \tag{10}$$

and for a surface with roughness coefficients $\rho_x = \frac{1}{s_x}$ and $\rho_y = \frac{1}{s_y}$, the co-variance matrix can be expressed as,

$$\Sigma_{rough} = \Sigma + \begin{bmatrix} \frac{1}{s_x} & 0 \\ 0 & \frac{1}{s_y} \end{bmatrix}. \tag{11}$$

6. Implementation Details

Our proposed method is implemented as a custom BSDF plugin in Mitsuba 3 [JSR*22] and we additionally update the Path Replay Backpropagation Integrator to account for delta transmission.

6.1. Details on CAPM

Newtonian parallax mapping [KZP*24] takes two steps for wefts and warps i.e. a total of four steps to offset the UV-map, our proposed parallax mapping solution adds an additional overhead due to refinement step. However, it is still considerably faster in comparison to traditional parallax mapping approaches. Additionally unlike previous parallax mapping approaches, our approach further updates for the silhouette of the mesh with little to no effect on performance.

The individual steps for our approach is summarized in Algorithm 1. In the first step, we offset the UV map by querying the underlying height of the parametric knit curves, s_0. If the ray overshoots and ends up below the curves then we trace back towards the ray-origin in small increments of δs. With each new offset UV position, we first query the tangent and normal maps of the base mesh and transform the ray into the local space defined by the queried vectors. This step bends the ray ensuring that the curvature of the mesh is taken into account. If at any stage, the ray bends away from the mesh, then it is allowed to pass-through the mesh otherwise we continue to offset the UV map until an intersection with the parametric curves is found. In case the ray undershoots after step one, i.e. the offset position is still above the parametric curves then we continue moving by δs along the ray direction.

In practice, we found that our approach takes at most ten steps to converge and a detailed analysis and comparison of our approach can be found in Section 7.2.

Algorithm 1: CAPM: Curvature-Aware Parallax Mapping

```
/* P_UV is the texture co-ordinate of the
   intersection point on the base mesh */
/* ω_i is the incident ray direction */
/* a, h, d, r are the parameters for the
   knit curve */
/* δs is the step-size for refinement */
Input: P_UV, ω_i, a, h, d, r, δs
Output: P_UV^hit
/* Big newtonian step with the height at
   point P_UV */
P_UV.z = 1 ;
isHitCurve = false ;
b, t, n = fetchFrame( P_UV.x, P_UV.y ) ;
s_0 = getHeight( P_UV.x, P_UV.y, a, h, d, r ) ;
ω_local = b.ω_i + t.ω_i + n.ω_i ;
P_UV^0 = P_UV + s_0 ω_local ;
s_1 = getHeight( P_UV^0.x, P_UV^0.y, a, h, d, r ) ;
if s_1 ≤ P_UV^0.z then
    δs = −δs ;
end
/* Refinement step with δs */
while !isHitCurve && ω_local.z ≥ 0 do
    b, t, n = fetchFrame( P_UV^i.x, P_UV^i.y ) ;
    ω_local = b.ω_i + t.ω_i + n.ω_i ;
    P_UV^{i+1} = P_UV^i + δs ω_local ;
    s_{i+1} = getHeight( P_UV^{i+1}.x, P_UV^{i+1}.y, a, h, d, r ) ;
    if s_{i+1} ≤ P_UV^{i+1}.z && δs ≥ 0 then
        isHitCurve = true ;
    end
    if s_{i+1} ≥ P_UV^{i+1}.z && δs ≤ 0 then
        isHitCurve = true ;
    end
end
P_UV^hit = P_UV^i ;
return P_UV^hit ;
```

6.2. Details on the Multi-Scale Model

Our multi-scale model relies on aggregation of the pattern geometry over the texture footprint, $\mathcal{T}$. The coverage over $\mathcal{T}$, defines a 4D space in $(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y})$ and thus we require a fast approach to generate the samples that will serve as the originating point of our implicit rays for the aggregation. To this end, we use Sobol sequences [Sob90], which are quasi-random sampling patterns that produce evenly distributed (low-discrepancy) samples. We use the generated samples to offset the UV values which serve as the new origin for the implicit rays and perform repeated pattern mapping to aggregate the normal, and visibility over the texture footprint $\mathcal{T}$.

The visibility map determined is no longer binary thus for sampling the next ray, we take a weighted average, based on p_{delta} of the incoming ray direction and the sampled direction from one

of the four BSDF components. This ensures that when $p_{delta} \to 0$, more importance is given to the chosen BSDF component and when $p_{delta} \to 1$ then the ray simply passes through the base mesh surface.

7. Results

In this section, we present the results of our surface-based model in different settings. We first c ompare o ur w ork t o p revious state-of-the-art ray-tracing models and real photographs, we use the yarn-based model [KZP*24] as the reference comparison in both the macro and micro-scale and compare against the surface-based model proposed by Zhu et al. [ZJA*23], referred to as Zhu'23. We do not consider Zhu et al. [ZHB*24] in our analysis, as their work follows an approach that is perpendicular to ours. We exclude rasterization-based methods [WY17; HSW*25] because as their execution model and performance characteristics differ fundamentally from ours. In particular, these methods rely on the rasterization pipeline rather than ray-based evaluation, making their performance profiles a nd t rade-offs n ot d irectly c omparable t o the approach presented in this work. In addition, we analyze the different components of our model and show its flexibility i n different scenes with varying knitted patterns. All our renderings are generated on an 8-core Intel i7.

7.1. Comparison with Previous Works

In Figure 7, we demonstrate the ability of our model to achieve a close match to the explicitly generated yarns along a half-cylinder under both front and back lighting. The yarn geometries are generated along a cylinder surface for *knits* and *purls* for the reference and Zhu'23 requires additional normals and tangent maps. Our model only requires the input pattern representation to closely match the reference.

In far-view, our parallax mapping takes into account the inter-looping nature of knitted yarns that affects the delta transmission by completely closing up the holes along the edges of the cylinder. Yarn-based reference explicitly requires the yarn curves and has to construct a complex bounding volume hierarchy increasing the render times in far-view. Our model is 3-5 times faster than the yarn-based reference and only requires 0.02MB of memory for the 1D textures of plies and fibers, s aving 9 9.8% o f m emory. Zhu'23 requires a small texture patch of normal and tangent maps however, since it wraps the texture along the surface normals the delta transmission is inaccurate along the azimuthal width of the cylinder (as shown with red and green boxes). In close-up, our method takes into account the micro-geometry present within a yarn, preserving the ply and fiber-level d etails a n d t h e v olumetric l o ok unlike Zhu'24 that produces a flat a p pearance a n d a d ds r a ndom perturbations on the yarns. Additionally, because of the front and back lighting, the double-layer approximation makes the yarns darker in the region where the two yarns overlap and brighter where they don't, which is once again missing in the Zhu'23.

In Figure 8, we also compare our model with a real photograph against Zhu'23 under a single spot light. Our parallax mapping offers a closer match to the changing geometry of knits along the azimuthal width of the cylinder. Additionally, Zhu'23 requires pre-fitting Anisotripic Spherical Gaussians (ASG) to determine the shadowing and masking while ours is pre-computation free, and ensures soft shadows towards the bottom half of the cylinder.

Table 2: ***Performance statistics for Figure 7 at equal quality (EQ).*** *We compare the performance of Ours against surface-based model [ZJA*23] referred as Zhu'23 and yarn-based model [KZP*24] referred to as Khat'24. Our method on average performs 3-5 times faster than Khat'24 and 1.4 times faster than Zhu'23.*

		Render Time (min)			Storage (MB)		
Scene	#Yarn	Ours	Zhu'23	Khat'24	Ours	Zhu'23	Khat'24
Fig. 7.a.i	200	**1.1**	1.7	5	**0.02**	4.4	20
Fig. 7.b.i	200	**1.1**	1.7	5	**0.02**	4.4	20
Fig. 7.a.ii	200	**1.1**	1.7	3.2	**0.02**	4.4	20
Fig. 7.b.ii	200	**1.1**	1.7	3.2	**0.02**	4.4	20

7.2. Ablation study

Effect of parallax-mapping in close-up and far-view. Our parallax mapping changes the shape of the yarns, which affects the inter-yarn spacing leading and the silhouette of the mesh. In Figure 9, we render *knit* curves on a cylinder surface. The parallax mapping effects the inter-yarn spacing, making some holes close-up or become more wide and the overall silhouette of the mesh also changes with a ribbed boundary appearing for the cylinder as the curves loop in and out of the surface. We further test our approach on arbitrary meshes in Figure 10. We analyze how it affects the overall appearance of fabrics in both close-up and far-view for two different fabric meshes. For the tablecloth (row I), parallax mapping changes the inter-yarn spacing which perturbs the holes present in the fabric. This effects the delta transmission. For the scarf (row II), parallax mapping makes the bumps of the parametric knit curves more prominent in far-view, the overall result is brighter due to more accurate delta transmission and additional changes to the silhouette of the mesh.

Support for arbitrary knit patterns. With the matrix representation of knit patterns as discussed in Section 3.3, our approach can be seamlessly extended to support any arbitrary knitted patterns unlike previous approaches that require explicitly generated height-field and normal textures. An arbitrary pattern can be designed to depict more complex structures such as pyramids or logos by combining *knits* and *purls* of varying colors. In Figure 11, we show the renderings of two custom patterns, taken from the dataset provided by [LWS*18] on a rectangular flat mesh.

Pattern mapping exhibits a flat appearance on a rectangular mesh. To show the flexibility of our model, we map a 16×16 pyramid and 48×48 block pattern to two cushions as shown in Figure 12. We further add flyaway fibers, commonly present in knitted fabrics to enhance the realism and show that our model is compatible to pre-existing rendering techniques, commonly used in production.

Multi-scale evaluation. In Figure 1 and 13, we show render-

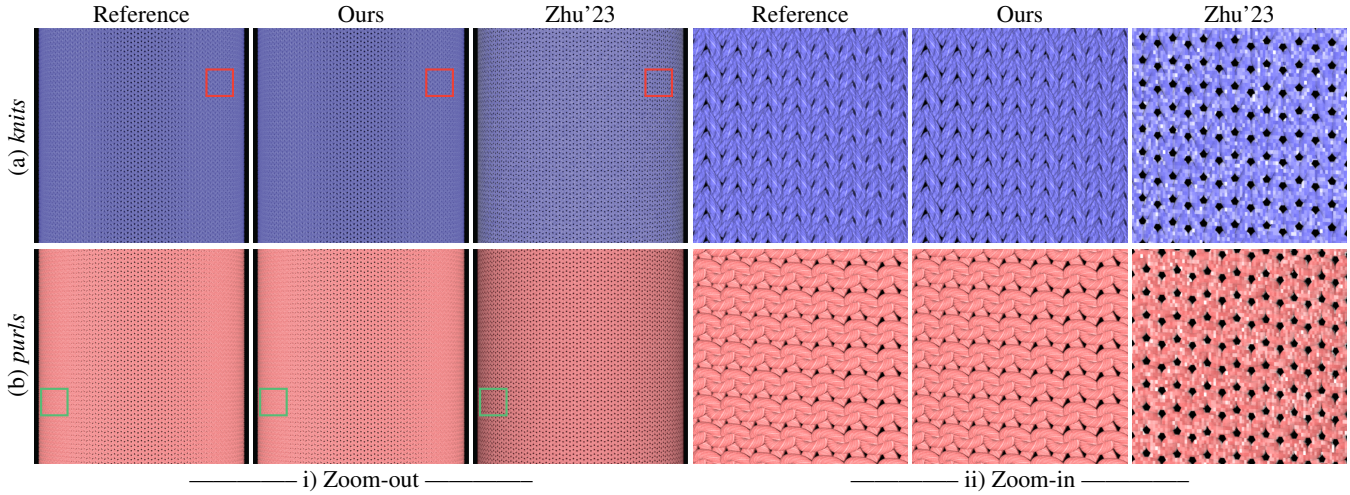

Figure 7: ***Comparison with previous yarn-based and surface-based models on a half cylinder at different viewing distances with both front and back directional lighting.*** *In far-view, our parallax mapping affects the inter-yarn spacing and yarn curvature unlike Zhu'23 which tiles flat yarns over the cylinder surface. Furthermore, our model adds the ply and fiber details in close-up that are missing for Zhu'23 and accounts for the loss when light transmits through overlapping regions of the yarns. A quantitative comparison can be found in Table 2*

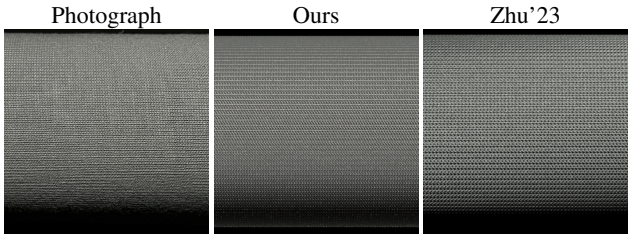

Figure 8: ***Comparison with photograph in far-view under the same lighting setup as Zhu et al. [ZJA*23].*** *We compare against the cotton jersey knit to show that our parallax accounts for the changing inter-yarn spacing and accurate shadows unlike Zhu'23 that requires a pre-fitted Anisotripic Spherical Gaussians (ASG) for shadowing and has difficulty on the soft shadows toward the bottom half.*

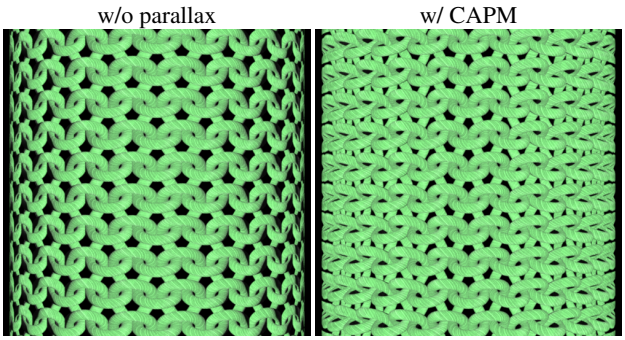

Figure 9: ***Effect of parallax mapping on yarn shape and mesh silhouette***

. We visualize a cylinder surface with and without CAPM to show how much it can affect the inter-yarn spacing and silhouette for loose *purls* .

ings in close-up and far-view to analyze our multi-scale model. In far-view, we perform comparison in Equal Quality (EQ) and Equal Time (ET) to show the efficiency offered by our model and also look at the overhead from pattern aggregation over the pixel patch. The multi-scale performs 1.5-1.7 faster than the near-field model to converge because it requires fewer samples per pixel. In close-up, the multi-scale model converges to the same result as the near-field however since we do not change number of sampled positions for implicit pattern aggregation, it does an additional overhead. In essence, when a fabric is viewed in far-view, our multi-scale model provides much more efficient performance and in close-up, the number of sampled positions for implicit pattern aggregation can be reduced to achieve the same result as the near-field model in equal time.

8. Discussions & Conclusion

Faster multi-scale integration: We propose a sampling-based multi-scale which adds an overhead in close-up in comparison to near-field model. To address this, future work can focus on multi-scale approach that can ensure faster aggregation of the appearance over the texture footprint $\mathcal{T}$ to perform better.

Support for all knit stitch types: Leaf et al. [LWS*18] describe several additional knit stitch types beyond the four presented in Table 1, many of which produce more pronounced volumetric effects. Future work could investigate how these stitches may be represented as parametric curves, enabling the generation and support of a broader range of knitted patterns.

Flyaway fibers: As shown in Figure 12, flyaway fibers enhance

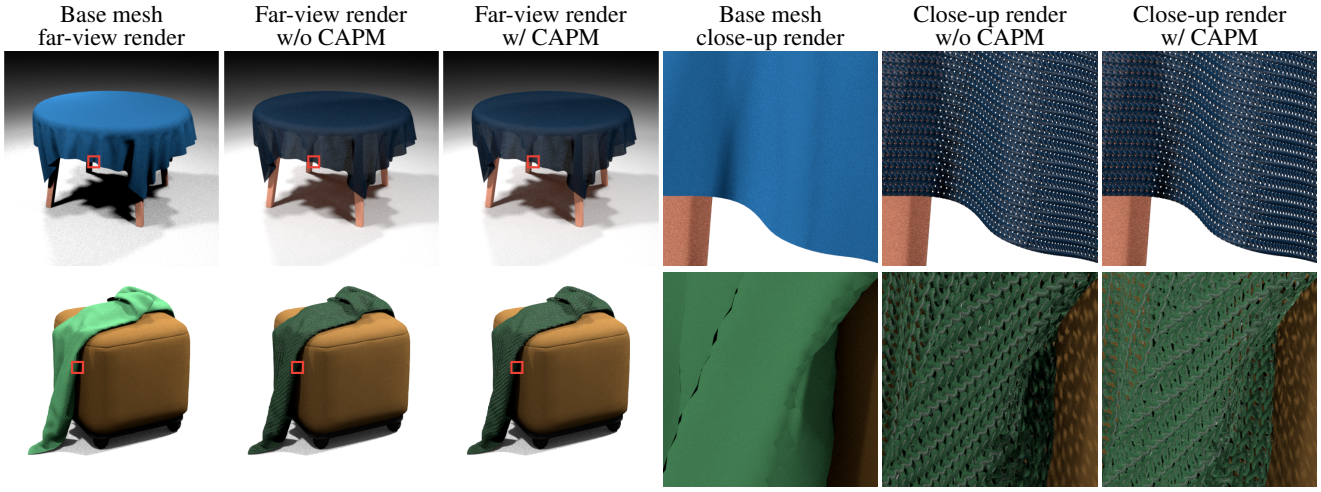

Figure 10: ***Rendering knitted fabrics at different scales with CAPM on/off.*** *We render two different fabric meshes under four area lights to showcase the efficiency of our silhouette-aware parallax mapping. For the tablecloth (row I), CAPM improves the delta transmission through the tiny holes. For the scarf (row II), the change in the silhouette and bumps of knits becomes pronounced.*

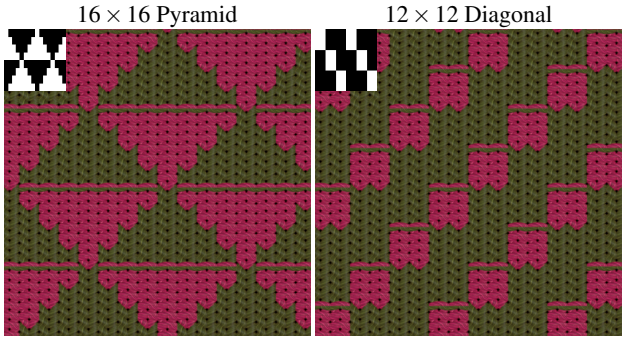

Figure 11: ***Mapping of arbitrary knit pattern****. We visualize the parametric cylinders obtained by mapping two pre-existing patterns to show how our model can easily handle complex interlooping nature of yarns.*

Figure 12: ***Additional results with flyaways.*** *We show two complex patterns:* 16 *pyramid and* 48 × 48 *block pattern knitted cushions with flyaway fibers modeled as explicit curves. The sofa showcases a classic plain weave pattern. The scene is rendered in environment lighting with large number of samples per pixel to model individual flyaways noise-free, which are independent of our approach.*

the realism of knitted fabrics with fuzziness and sheen effect. However these fibers need to be modelled explicitly as curves and developing a complete surface-based representation that can model the individual fibers remains a challenge.

Curvature-Aware Parallax Mapping (CAPM): Similar to traditional parallax mapping approaches that offset the *UV* sphere iteratively, CAPM alleviates some of the issues such as being texture-free or requiring less than steps but one interested problem to explore in the future is to determine if the parallax correction can be done in $O(1)$ time similar to the Newtonian method proposed by Khattar et al. [KZYM25] for woven fabrics.

Conclusion: This paper proposes a surface-based model for knitted fabrics, that allows efficient a n d d e tailed r e nderings i n both close-up and far-view. Unlike other surface-based model which require high resolution texture maps of knitted fabrics, our model is "texture-free", requiring only 1D textures for ply and fiber-level details and the matrix representation of the knit pattern to account for the yarn mapping. Taking into account the mesh curvature, our parallax mapping technique updates the silhouette of the mesh with no additional overhead in comparison to previous state of the art parallax mapping methods. We validate our model against real photograph and previous state of the methods commonly used for rendering knitted fabrics to showcase our model abilities to capture the intricate details of knitted fabrics in both close-up and far-view.

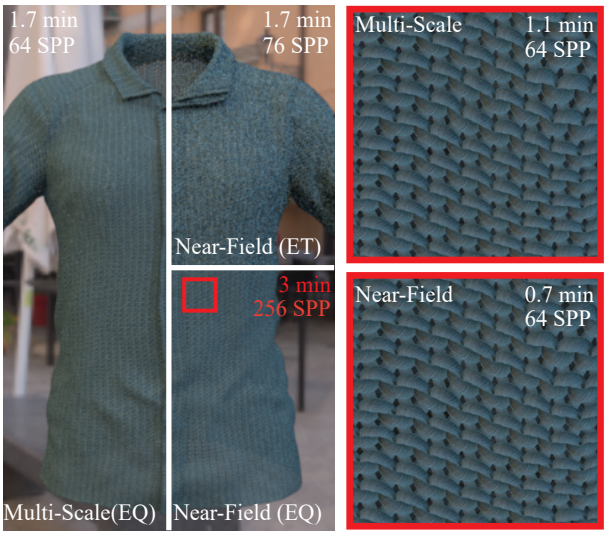

Figure 13: ***Multi-scale comparison in Equal Quality (EQ) and Equal Time (ET) under environment lighting.*** *In far-view the near-field model requires more ray samples per pixel to converge to a noise-free, unlike ours that performs pattern aggregation over the pixel, performing 1.5-1.7 times faster. In close-up, the aggregation adds an overhead but overall offers seamless transition from close-up to far-view.*

Acknowledgments

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

References

[CC08] CHEN, YING-CHIEH and CHANG, CHUN-FA. "A Prism-Free Method for Silhouette Rendering in Inverse Displacement Mapping". *Computer Graphics Forum*. Vol. 27. 7. Wiley Online Library. 2008, 1929–1936 6.

[Cra23] CRANE, KEENAN. *A Simple Parametric Model of Plain-Knit Yarns*. 2023 5.

[DvGNK99] DANA, KRISTIN J., van GINNEKEN, BRAM, NAYAR, SHREE K., and KOENDERINK, JAN J. "Reflectance and Texture of Real-World Surfaces". *ACM Trans. Graph.* 18.1 (Jan. 1999), 1–34. ISSN: 0730-0301. DOI: 10.1145/300776.300778. URL: https://doi.org/10.1145/300776.300778 3.

[HSW*25] HUANG, TAO, SHI, HAOYANG, WANG, MENGDI, et al. "Real-Time Knit Deformation and Rendering". *ACM Transactions on Graphics (TOG)* 44.4 (2025), 1–12 1, 2, 4, 9.

[Ige99] IGEHY, HOMAN. "Tracing ray differentials". *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 1999, 179–186 7.

[JMLH01] JENSEN, HENRIK WANN, MARSCHNER, STEPHEN R., LEVOY, MARC, and HANRAHAN, PAT. "A Practical Model for Subsurface Light Transport". *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '01. New York, NY, USA: Association for Computing Machinery, 2001, 511–518. ISBN: 158113374X. DOI: 10.1145/383259.383319. URL: https://doi.org/10.1145/383259.383319 3.

[JSR*22] JAKOB, WENZEL, SPEIERER, SÉBASTIEN, ROUSSEL, NICOLAS, et al. *Mitsuba 3 renderer*. Version 3.1.1. https://mitsuba-renderer.org. 2022 8.

[KSZ*15] KHUNGURN, PRAMOOK, SCHROEDER, DANIEL, ZHAO, SHUANG, et al. "Matching Real Fabrics with Micro-Appearance Models." *ACM Trans. Graph.* 35.1 (2015), 1–1 2.

[Kuz21] KUZNETSOV, ALEXANDR. "NeuMIP: Multi-resolution neural materials". *ACM Transactions on Graphics (TOG)* 40.4 (2021) 3.

[KWM*22] KUZNETSOV, ALEXANDR, WANG, XUEZHENG, MULLIA, KRISHNA, et al. "Rendering neural materials on curved surfaces". *ACM SIGGRAPH 2022 conference proceedings*. 2022, 1–9 3.

[KZP*24] KHATTAR, APOORV, ZHU, JUNQUI, PADOVANI, EMILIANO, et al. "A Multi-scale Yarn Appearance Model with Fiber Details". *arXiv preprint arXiv:2401.12724* (2024) 3, 7–9.

[KZYM25] KHATTAR, APOORV, ZHU, JUNQIU, YAN, LING-QI, and MONTAZERI, ZAHRA. "A Texture-Free Practical Model for Realistic Surface-Based Rendering of Woven Fabrics". *Computer Graphics Forum*. Wiley Online Library. 2025, e15283 1, 3–7, 11.

[LMCO17] LOPEZ-MORENO, JORGE, MIRAUT, DAVID, CIRIO, GABRIEL, and OTADUY, MIGUEL A. "Sparse GPU Voxelization of Yarn-Level Cloth". *Computer Graphics Forum*. Vol. 36. 1. Wiley Online Library. 2017, 22–34 2.

[LWS*18] LEAF, JONATHAN, WU, RUNDONG, SCHWEICKART, ESTON, et al. "Interactive design of periodic yarn-level cloth patterns". *ACM Transactions on Graphics (TOG)* 37.6 (2018), 1–15 3, 4, 9, 10.

[LZB17] LUAN, FUJUN, ZHAO, SHUANG, and BALA, KAVITA. "Fiber-Level On-the-Fly Procedural Textiles". *Computer Graphics Forum*. Vol. 36. 4. Wiley Online Library. 2017, 123–135 4.

[MGJZ21] MONTAZERI, ZAHRA, GAMMELMARK, SOREN B, JENSEN, HENRIK WANN, and ZHAO, SHUANG. "A Practical Ply-Based Appearance Modeling for Knitted Fabrics". *Proceedings of Eurographics Symposium on Rendering 2021*. 2021 1, 3, 4.

[MGZJ20] MONTAZERI, ZAHRA, GAMMELMARK, SØREN B, ZHAO, SHUANG, and JENSEN, HENRIK WANN. "A practical ply-based appearance model of woven fabrics". *ACM Trans. Graph.* 39.6 (2020), 1–13 3, 4.

[MXF*21] MONTAZERI, ZAHRA, XIAO, CHANG, FEI, YUN, et al. "Mechanics-Aware Modeling of Cloth Appearance". *IEEE Transactions on Visualization and Computer Graphics* 27.1 (Jan. 2021), 137–150. ISSN: 1077-2626. DOI: 10.1109/TVCG.2019.2937301. URL: https://doi.org/10.1109/TVCG.2019.2937301 2.

[Nic93] NICKALLS, RICHARD WD. "A new approach to solving the cubic: Cardan's solution revealed". *The Mathematical Gazette* 77.480 (1993), 354–359 6.

[OB10] OLANO, MARC and BAKER, DAN. "Lean mapping". *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*. 2010, 181–188 7, 8.

[PBFJ05] PORUMBESCU, SERBAN D, BUDGE, BRIAN, FENG, LOUIS, and JOY, KENNETH I. "Shell maps". *ACM Transactions on Graphics (TOG)* 24.3 (2005), 626–633 6.

[PO06] POLICARPO, FABIO and OLIVEIRA, MANUEL M. "Relief mapping of non-height-field surface details". *Proceedings of the 2006 symposium on Interactive 3D graphics and games*. 2006, 55–62 6.

[RJGW19] RAINER, GILLES, JAKOB, WENZEL, GHOSH, ABHIJEET, and WEYRICH, TIM. "Neural BTF compression and interpolation". *Computer Graphics Forum*. Vol. 38. 2. Wiley Online Library. 2019, 235–244 3.

[RKLG23] RODRIGUEZ-PARDO, CARLOS, KAZATZIS, KONSTANTINOS, LOPEZ-MORENO, JORGE, and GARCES, ELENA. "NeuBTF: Neural fields for BTF encoding and transfer". *Computers & Graphics* 114 (2023), 239–246 3.

[SBDJ13] SADEGHI, IMAN, BISKER, OLEG, DE DEKEN, JOACHIM, and JENSEN, HENRIK WANN. "A practical microcylinder appearance model for cloth rendering". *ACM Trans. Graph.* 32.2 (2013), 1–12 3.

[SM24] SOH, GUAN YU and MONTAZERI, ZAHRA. "Neural Appearance Model for Cloth Rendering". *Computer Graphics Forum*. Vol. 43. 4. Wiley Online Library. 2024, e15156 3.

[Sob90] SOBOĹ, IM. "Quasi-monte carlo methods". *Progress in Nuclear Energy* 24.1-3 (1990), 55–61 8.

[SZZ12] SCHRÖDER, KAI, ZHAO, SHUANG, and ZINKE, ARNO. "Recent advances in physically-based appearance modeling of cloth". *SIGGRAPH Asia 2012 Courses* (2012), 1–52 2.

[Tat06] TATARCHUK, NATALYA. "Dynamic parallax occlusion mapping with approximate soft shadows". *Proceedings of the 2006 symposium on Interactive 3D graphics and games*. 2006, 63–69 6.

[WY17] WU, KUI and YUKSEL, CEM. "Real-time Fiber-level Cloth Rendering". *Proceedings of I3D*. 2017. ISBN: 978-1-4503-4886-7/17/03. DOI: 10.1145/3023368.3023372. URL: http://doi.acm.org/10.1145/3023368.3023372 2, 9.

[XCL*25] XU, ZILIN, CHEN, XIANG, LIU, CHEN, et al. "Towards Comprehensive Neural Materials: Dynamic Structure-Preserving Synthesis with Accurate Silhouette at Instant Inference Speed". *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. 2025, 1–11 6.

[YKJM12] YUKSEL, CEM, KALDOR, JONATHAN M, JAMES, DOUG L, and MARSCHNER, STEVE. "Stitch meshes for modeling knitted clothing with yarn-level detail". *ACM Transactions on Graphics (TOG)* 31.4 (2012), 1–12 1, 3.

[ZHB*24] ZHU, JUNQIU, HERY, CHRISTOPHE, BODE, LUKAS, et al. "A Realistic Multi-scale Surface-based Cloth Appearance Model". *ACM SIGGRAPH 2024 Conference Papers*. 2024, 1–10 3, 7, 9.

[ZJA*23] ZHU, JUNQIU, JARABO, ADRIAN, ALIAGA, CARLOS, et al. "A Realistic Surface-based Cloth Rendering Model". *ACM SIGGRAPH 2023 Conference Proceedings*. 2023, 1–9 1, 3, 4, 7, 9, 10.

[ZJMB11] ZHAO, SHUANG, JAKOB, WENZEL, MARSCHNER, STEVE, and BALA, KAVITA. "Building volumetric appearance models of fabric using micro CT imaging". *ACM Trans. Graph.* 30.4 (2011), 1–10 2.

[ZLB16] ZHAO, SHUANG, LUAN, FUJUN, and BALA, KAVITA. "Fitting procedural yarn models for realistic cloth rendering". *ACM Trans. Graph.* 35.4 (2016), 1–11 2.

[ZMA*23] ZHU, JUNQUI, MONTAZERI, ZAHRA, AUBRY, J, et al. "A Practical and Hierarchical Yarn-based Shading Model for Cloth". *Computer Graphics Forum*. Vol. 42. 4. Wiley Online Library. 2023, e14894 2, 3.

[ZZW*22] ZHU, JUNQIU, ZHAO, SIZHE, WANG, LU, et al. "Practical level-of-detail aggregation of fur appearance". *ACM Transactions on Graphics (TOG)* 41.4 (2022), 1–17 4.