A Realistic Multi-scale Surface-based Cloth Appearance Model

Junqiu Zhu
University of California, Santa Barbara
USA
junqiuzhu@ucsb.edu

Christophe Hery
Meta Reality Labs Research
USA
chery@meta.com

Lukas Bode
Meta Reality Labs Research
Switzerland
lbode@meta.com

Carlos Aliaga
Meta Reality Labs Research
USA
caliaga@meta.com

Adrian Jarabo
Meta Reality Labs Research
Spain
ajarabo@meta.com

Ling-Qi Yan
University of California, Santa Barbara
USA
lingqi@cs.ucsb.edu

Matt Jen-Yuan Chiang
Meta Reality Labs Research
USA
mattchiang@meta.com

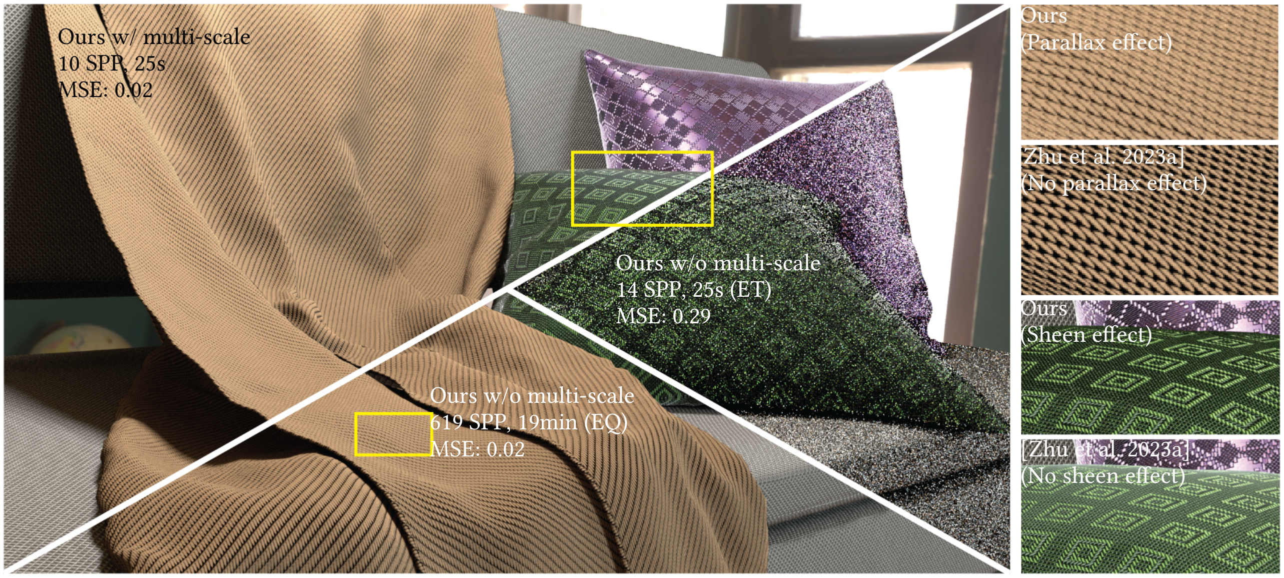

Figure 1: We demonstrate the effectiveness of our new BSDF model and multi-scale shading techniques in a scene featuring various fabrics: twill (blanket), plain weave (couch), and fancy diamond and argyle patterns for satins of the green and purple pillows, respectively. Through comparisons in equal time (ET) and equal quality (EQ) to brute force Monte Carlo integration within each pixel footprint (ours w/o multi-scale), our multi-scale scheme achieves significant improvements in both realism and performance, with a 33x speedup in this example. The impact of the new sheen component and parallax offset is more apparent in the close-up views on the right.

ABSTRACT

Surface-based cloth appearance models have been rapidly advancing, shifting from detail-less BRDFs to modern per-point shading models with accurate spatially-varying reflection, transmission, and so on. However, the increased complexity has brought about realism-performance trade-offs: from closeup, rendered cloth can be highly inaccurate due to the missing, unaffordable parallax effects; from far away, significant amount of noise will show up since every point can be shaded differently inside a pixel's footprint. In this paper, we aim at eliminating the trade-off with a realistic multi-scale surface-based cloth appearance model. We propose a comprehensive micro-scale model focusing on correct parallax effects, and a practical meso-scale integration scheme, emphasizing efficiency while losslessly preserving accurate highlights and self-shadowing. We further improve its performance using our novel Clustered Control Variates (CCV) and Summed-Area Table (SAT) integration scheme, and its practicality using an efficient Clustered Principal Component Analysis (C-PCA) compression method. As a result, our multi-scale model achieves a 30× acceleration compared to the state-of-the-art, is able to represent a variety of realistic cloth appearance, and can be potentially applied in real-time applications.

CCS CONCEPTS

• **Computing methodologies → Rendering**; **Reflectance modeling**.

KEYWORDS

cloth, parallax, sheen, multi-scale

ACM Reference Format:

Junqiu Zhu, Christophe Hery, Lukas Bode, Carlos Aliaga, Adrian Jarabo, Ling-Qi Yan, and Matt Jen-Yuan Chiang. 2024. A Realistic Multi-scale Surface-based Cloth Appearance Model. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24 (SIGGRAPH Conference Papers '24), July 27-August 1, 2024, Denver, CO, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3641519.3657426

1 INTRODUCTION

Cloth rendering has been a focal topic in the field of computer graphics, chiefly due to its ubiquity as a common element in virtual scenes, encompassing online electronic sales, movies, games, and virtual reality. The realism of cloth rendering is vital to the overall realism in renderings. However, accurately modeling cloth remains a formidable challenge; it is complex in both structure and optical properties, leading to important variations in its appearance.

Numerous explorations have been undertaken for reproducing the appearance of cloth, with particular attention given to surface-based cloth rendering methods. These methods garner significant research and application attention due to their typically faster rendering compared to more resource-intensive volumetric or curve-based methods, and their direct applicability to geometries described by 3D meshes.

Albeit their advantages, surface-based models are often criticized for struggling to balance performance and realism. From traditional [Irawan and Marschner 2012] to the state-of-the-art [Zhu et al. 2023a], surface-based models have been introducing abundant details with reflection, transmission, shadowing-masking, etc., using only a few small tileable feature maps as input. On the one hand, these details greatly enhance the realism of rendered cloth. On the other hand, however, they impose significant challenges at both far- and near-field scales. At the far-field, shading a pixel requires accounting for every texel within the pixel footprint, hampering rendering efficiency and indicating a lack of multi-scale[1] rendering scheme. At the near-field, they manifest incorrect parallax and thus unnatural black areas. This is because they cannot afford hierarchy-based parallax/displacement map rendering techniques, which are already introducing per point rendering overhead, let alone making far-field rendering even more difficult.

Therefore, we find surface-based cloth rendering in an awkward situation: from closeup, it is complex [Xu et al. 2019] and difficult to render every point correctly and realistically, unless details can be completely decoupled from cloth appearance [Sadeghi et al. 2013]; from far away, there is no efficient multi-scale solution whose performance is independent of the size of pixel footprints. Motivated by these observations, we propose a realistic and efficient multi-scale cloth appearance model. We base our model upon Zhu et al. [2023a]. However, we aim at addressing these general, long-standing issues in surface-based cloth rendering with the following components:

- a comprehensive micro-scale (near-field) model focusing on correct parallax/masking effect, as well as introducing the previously overlooked sheen effect; and
- a practical meso-scale (mid- to far-field) integration scheme of our near-field model, emphasizing efficiency while losslessly preserving accuracy.

More specifically, to counteract the incorrect self-occlusions resulting from flat geometry assumptions, in Sec. 3, we propose to address parallax by computing a direction-dependent offset, inspired by NeuMIP [Kuznetsov 2021]. This allows for direct querying on the fly without the need for additional tracing. We also additionally incorporate a sheen layer into our model, to better capture the appearance around grazing angle. Following our micro-scale model, in Sec. 4, we propose an multi-scale solution that effectively integrates cloth appearance within a pixel. Our insight is to leverage advances in Control Variates (CV) for rendering [Novák et al. 2018; Crespo et al. 2021], splitting the integral into a low-dimensional *integral radiance*, as well as a generally dim *residual* – the former solved using Summed Area Tables (SATs) in $O(1)$, with our efficient compression scheme reducing storage to a mere 0.8% of the original size, and the latter sampled with an extremely low budget (usually 2 samples per pixel), thanks to the Clustered Control Variates that we propose to further enhance efficiency.

As a result, we are able to achieve over 30× acceleration comparing to previous state-of-the-art cloth rendering model [Zhu et al. 2023a]. In Sec. 6, for offline applications, we showcase the realistic parallax and sheen effects with fast convergence, along with preliminary real-time results. Therefore, we believe that our multi-scale surface-based cloth appearance model is comprehensive to represent a variety of realistic cloth appearances.

2 RELATED WORK

2.1 Cloth Rendering

Cloth rendering can primarily be categorized into curve-based, volume-based, and surface-based methods.

Curve/Volume-based Methods. A large number of works define the cloth as an assembly of fibers or plies, either through explicit geometry [Zhao et al. 2016] or as a heterogeneous volumetric representation [Zhao et al. 2011; Schroder et al. 2011]. The main advantage of curve-based cloth models is the accurate representation of fiber-level geometric details at a microscopic scale. However, the storage requirements for 3D details in curve-based methods are substantial, and the complexity of multiple bounces in modeling curves makes efficient rendering challenging. To address practical concerns, recently, Montazeri et al. [2020; 2021] reduced the modeling of geometry to only the level of plies, implicitly modeling the underlying fibers' contribution with approximate single and multiple scattering. Zhu et al. [2023a] further proposed a method that models only yarn curves, hiding ply and fiber geometries into

[1] We avoid the wording "Level of Detail (LoD)" since it is often connected to geometry simplification, and "pre-filtering" since it usually indicates quality loss.

a multiple scattering appearance model. However, these methods are still curve-based, only to different levels of simplification, complex light transport between plies/yarns are still required by heavy ray-tracing.

Surface-based Models have been used for further 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) [Adabala et al. 2003; Irawan and Marschner 2012; Sadeghi et al. 2013; Jin et al. 2022]. The surface-based models are popular for their fast performance, however, these models often lack realism, due to missing yarn/ply/fiber 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. [2023b] proposed a realistic surface-based cloth shading method. This method is able to reproduce the important visual features, with its appearance model derived from only a few texture maps. Our model is based on this method, enhancing its realism by adding the effect of sheen. Meanwhile, as analyzed in Sections 3 and 4, we challenge the two common issues from surface-based cloth appearance models, aiming at a correct microscale model with a fast parallax effect, as well as an efficient multi-scale solution applicable for both offline and real-time rendering.

2.2 Appearance Aggregation

Appearance aggregation is the process of condensing or summarizing the intricate details of the appearance into a simplified representation. This process can enhance rendering efficiency and performance without sacrificing visual quality. Appearance aggregation methods, such as LEAN [Olano and Baker 2010], LEADR [Dupuy et al. 2013], Bump2Roughness [Hery et al. 2014], and more [Han et al. 2007; Wu et al. 2009], simplify the complex surface details in the geometric structure and adjust the generated BRDFs to maintain the overall appearance. They often take advantage of frequency analysis or optimization to balance the positional and angular resolution, thus are approximate. A more complex appearance aggregation is the "glinty" effect. Yan et al. [2014; 2016] rendered glints by accurately modeling the distribution of microfacets using 4D Gaussian primitives. They utilized a hierarchical structure to accelerate the calculation of contributions of microfacets within each pixel footprint. To further accelerate rendering, several methods have been developed to improve efficiency using SAT-like structures. Gamboa et al. [2018] directly employed the SAT structure to efficiently calculate pixel contributions in $O(1)$ time. However, the SAT structure requires significant storage. Addressing this, Atanasov et al. [2018] proposed an Inverse Bin Mapping (IBM) structure, which builds an efficient 2D grid plus 2D forest hierarchy. This method mitigates the storage issue; however, its performance, though arguably fast, can still not be comparable to $O(1)$ due to the traversal. These methods essentially solve for fast and accurate appearance, just like our goal. However, neither can be directly applied to our cloth appearance model, because we have much more complex "primitives"–the BRDFs from each texel are far more complex than a microfacet. Besides, we need to consider accurate shadowing-masking, which further raises the dimension of the problem. We will elaborate the analysis in Sec. 4.

2.3 Control Variates in Rendering

Control Variates is a general method for variance reduction in solving an integral, by splitting the integral into a relatively easy-to-solve part "base" term and a "residual" term. The residual will still be sampled but can result in much less noise provided that the base is close to the actual integrand. There are many works [Crespo et al. 2021; Novák et al. 2018] that apply the Control Variates to rendering. In the context of production path tracing for animated films, Control Variates has been used to reduce variance in the direct illumination integral, including visibility [Hery 2018]. Some methods have applied Control Variates to direct illumination or glossy reflections [Szécsi et al. 2004], average hemispherical visibility [Clarberg and Akenine-Möller 2008]. Müller et al. [2020] proposed a neural control variate method to light transport. In our rendering, we find the shadowing-masking term and randomness of fiber directions difficult to handle. So we take inspiration from Control Variates, taking these terms out and resulting in a relatively simple part that resembles glints rendering. We further cluster the visibilities to reduce the residual term for better efficiency as will be described in Sec. 4.2.

3 AN IMPROVED BSDF MODEL FOR CLOTH

Here we introduce an improved surface-based model for rendering cloth, which builds upon the recent model from Zhu et al. [2023a]. Zhu's model is a comprehensive model that supports reflection, transmission, and delta transmission, as well as non-local effects such as shadowing and masking. The model has been proved very effective for reproducing both knitted and woven cloth up to ply level. However, it has a few limitations that result in a poor representation of cloth at grazing angles, in particular by neglecting the effect of parallax, and by not accounting for the sheen effect resulting from light scattering with flyaway fibers (see Figure 8). The improvement of realism can also benefit inverse rendering. In the following, we briefly introduce the model from Zhu et al., and then explain our solution for these two effects, and how they fit into our improved model.

3.1 Background

Zhu et al. [2023a] modeled the meso-scale scattering of light from direction $\mathbf{i}$ to direction $\mathbf{o}$ of a point x on a piece of cloth as

$$\begin{aligned} f(x,\mathbf{i},\mathbf{o}) = {} & V(x,\mathbf{i},\mathbf{o})(I(x)\left(f^{r,s}(x,\mathbf{i},\mathbf{o}) + f^{r,d}(x,\mathbf{i},\mathbf{o})\right)) \\ & + V(x,\mathbf{i},\mathbf{o})(I(x)\left(f^{t,s}(x,\mathbf{i},\mathbf{o}) + f^{t,d}(x,\mathbf{i},\mathbf{o})\right)) \\ & + V(x,\mathbf{i},\mathbf{o})(1 - I(x))f^{\delta}(\mathbf{i},\mathbf{o}), \end{aligned} \tag{1}$$

with $V(x,\mathbf{i},\mathbf{o}) = V(x,\mathbf{i})V(x,\mathbf{o})$ and $V(x,\mathbf{i})$ and $V(x,\mathbf{o})$ the visibility functions from $\mathbf{i}$ and $\mathbf{o}$ respectively; $f^{r,s}$ and $f^{t,s}$ the specular reflection and transmission respectively, modeled using the SpongeCake model [Wang et al. 2022]; $f^{r,d}$ and $f^{t,d}$ the diffuse reflection and transmission modeling higher-order bounces; and f^{δ} the delta transmission handled via the indicator function $I(x)$ which returns 0 if x is in a gap and 1 otherwise (see the supplemental for details on

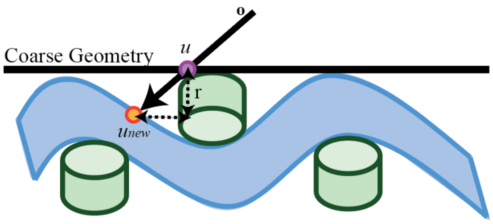

Figure 2: Schematic illustration of the parallax offset. Green cylinders represent the warp yarns, blue cylinder represents the weft yarn.

these terms). The model is parametrized using a small set of textures representing a minimally-tileable pattern, including fiber tangent direction, normal, height map, and an horizon map per texel computing visibility encoded as an anisotropic spherical Gaussian [Xu et al. 2013].

3.2 Handling parallax

A common problem for most surface-based cloth models [Sadeghi et al. 2013; Zhu et al. 2023a] is the assumption of flat surfaces, that breaks at grazing angles due to non-planar features in the cloth, specially in knits with big yarns. In these cases, parallax plays a significant role: As exampled in Figure 2, when observing a point u at direction $\mathbf{o}$, it is occluded from the geometry and the first visible point is at u_{new}, effectively requiring a view-dependent offset when querying the texture, following

$$u_{new} = u + \frac{r}{\mathbf{o}.z}(\mathbf{o}.x, \mathbf{o}.y), \tag{2}$$

where $\mathbf{o}.z$ is the z dimension in vector $\mathbf{o}$ in local coordinates (and analogously the rest of dimensions), r is the ray depth, computed by ray-tracing the height field. Several techniques exist handling such parallax, including explicit displacement [Wang et al. 2004] or parallax mapping [Kaneko et al. 2001] which solves Equation (2) in render time. Unfortunately, these methods introduce overhead, and make very complicated to pre-filter the aggregated reflectance within the pixel footprint. Inspired by NeuMIP [Kuznetsov 2021], we instead precompute the directionally-dependent parallax offset r as a 2D×2D table, which we query in run-time. More specifically, for each texel in the texture u we store the parameter r for a grid of 32^2 grid of directions uniformly distributed in solid angle. To avoid an infinite r in grazing angles, we only consider $\mathbf{o}.z > 0.6$.

3.3 Sheen

Fly-away fibers in textiles generate a strong forward and backward reflectance at grazing angles, generally called sheen. This is particularly noticeable in fabrics made of staple natural fibers, such as cotton and wool. A natural approach to handle this effect in our model would just require to stack an additional Spongecake microflake-like layer on top of the base cloth model. However, we found that the sheen model proposed by Zeltner et al.[2022] results in a more natural look, since it takes into account multiple scattering and it is extremely cheap to evaluate: Just a few evaluations of the linearly transformed cosine distribution [Heitz et al. 2016] are required to compute the sheen scattering function f^{sheen}, as well as the directionally-dependent sheen transmittance T^{sheen}.

3.4 Our Full BSDF

Putting all pieces together, we define our full local BSDF as

$$f(x, \mathbf{i}, \mathbf{o}) = V(x', \mathbf{i})\, \widehat{f}(x', \mathbf{i}, \mathbf{o}), \tag{3}$$

With x' the point obtained after applying the parallax transformation to x following Equation (2), and $\widehat{f}(x', \mathbf{i}, \mathbf{o})$ the local BSDF defined as

$$\begin{aligned}\widehat{f}(x', \mathbf{i}, \mathbf{o}) = {} & f^{sheen}(x', \mathbf{i}, \mathbf{o}) \\ & + T^{sheen} I(x') \left(f^{r,s}(x', \mathbf{i}, \mathbf{o}) + f^{r,d}(x', \mathbf{i}, \mathbf{o})\right) \\ & + T^{sheen} I(x') \left(f^{t,s}(x', \mathbf{i}, \mathbf{o}) + f^{t,d}(x', \mathbf{i}, \mathbf{o})\right) \\ & + T^{sheen}(1 - I(x')) f^{\delta}(\mathbf{i}, \mathbf{o}).\end{aligned} \tag{4}$$

Note that the outgoing visibility $V(x, \mathbf{o})$ is now omitted, since it is implicit after applying the parallax transformation.

4 MULTI-SCALE SHADING

4.1 Problem Analysis and Overview

In the following, we describe how to efficiently render cloth at any viewing distance using the improved BSDF model. Taking parallax effect into account, we now arrive at our meso-scale integration formula. The goal is to integrate the appearance of a piece of cloth within a pixel footprint $\mathcal{P}$ as

$$F_{\mathcal{P}}(\mathbf{i}, \mathbf{o}) = \int_{\mathcal{P}} \widehat{f}(x(p), \mathbf{i}, \mathbf{o}) \cdot V(x(p), \mathbf{i}) \cdot \langle \mathbf{n_s} \cdot \mathbf{i} \rangle \, \mathrm{d}p, \tag{5}$$

where our local BSDF $\widehat{f}$ (Eqn. 4) and the visibility V are unique for each point $p \in \mathcal{P}$, n_s is surface normal and $\langle . \cdot . \rangle$ is the absolute value of the dot product operator. This is to aggregate a 4D directional function over 2D positions. Such high dimensional integral is particularly complex in the case of shiny fabrics or intricate woven or knit structure patterns. Previous techniques [Zhu et al. 2023a] rely on the *effective BSDF* formulation to aggregate reflectance over micro-geometry (please refer to the original paper for details). However, such a solution still requires a large amount of samples per pixel to be noise free, especially at far distances, which results in long render time, as illustrated in Fig. 1.

Instead, inspired by the work of Hery [2018], we leverage Control Variates (CV) to make the problem tractable: The key idea is that, if there is a term in Equation 5 that is easy to compute, we can split the integral to a *base* term and a *residual* term that corrects for such approximation. This can be expressed as:

$$\int_{\mathcal{P}} f(p) \cdot v(p)\, \mathrm{d}p = \int_{\mathcal{P}} \alpha \cdot f(p)\, \mathrm{d}p + \int_{\mathcal{P}} f(p) \cdot (v(p) - \alpha)\, \mathrm{d}p \tag{6}$$

$$= \alpha \cdot F + \int_{\mathcal{P}} f(p) \cdot (v(p) - \alpha)\, \mathrm{d}p, \tag{7}$$

with $f(p) = f^j(p, \mathbf{i}, \mathbf{o})$ and $v(p) = V(p, \mathbf{i})$, $f^j(p, \mathbf{i}, \mathbf{o})$ a scattering component in Eqn. 4, α a constant, and $F = \int_{\mathcal{P}} f(p) \mathrm{d}p$. Note that for brevity we omit the directional dependence on each factor and the foreshortening term in Eqn. 5. We now arrive at a base term that is potentially lower dimension and easy to compute, and a residual term that remains a high dimensional integral which we compute using Monte Carlo.

In the following sections, we will first introduce a technique called *Clustered Control Variates* to further limit the use of brute-force Monte Carlo integration for the residual term in Sec. 4.2. Then, drawing inspiration from *Integral Histogram* commonly used for glints rendering, in Sec. 4.3 we will introduce a technique called *Integral Radiance* to solve our base term efficiently.

4.2 Clustered Control Variates

The residual term in Eqn. 7 is of the original high dimension and it is typically computed by brute force Monte Carlo sampling. However, we now have a free constant α to fine tune the blending between base and residual terms. Obviously, to limit variance from Monte Carlo sampling, we would like to favor the base term if possible, while lowering the energy in the residual.

Following the approach of Hery [2018], we set α to the average $\bar{v}$ of the visibility $v(p)$ function, as long as this computation stays a constant in regards to the integral and is also fairly cheap to obtain. Unfortunately, in our case computing $\bar{v}$ would require to average the visibility encoded as ASGs over all texels in the pixel footprint, which effectively would fall back to computing the integral numerically. However, by exploiting the repeated structure of fabrics, where fibers are oriented in a limited number of directions, mostly depending on the pattern type and the level of fiber twisting, we can design a better heuristic.

Instead of computing an average visibility for the whole pixel footprint in run-time, we segment the texture map into several clusters, each containing texels with similar visibility, using K-means. As a similarity criteria for two different ASGs, we compute the L2 difference between their set of parameters, namely orthogonal axis, rotation angle, and bandwidth, which have intuitive physical meaning. Then, we calculate the average visibility $\bar{v}_c$ for each cluster, and incorporate in the Control Variates scheme following

$$\int_{\mathcal{P}} f \cdot v \, \mathrm{d}p = \sum_{c \in 1,\dots,N_c} \bar{v}_c F_c + \int_{\mathcal{P}_c} f \cdot (v - \bar{v}_c) \, \mathrm{d}p, \tag{8}$$

with $\mathcal{P}_c$ the texels of cluster c in the pixel footprint $\mathcal{P}$. During rendering, we accumulate the contribution of each cluster, denoted as F_c, along with its corresponding average visibility, $\bar{v}_c$. We perform Monte Carlo sampling on the residual term, taking into account the average visibility of the cluster corresponding to this particular sample, represented by $\bar{v}_c$. We find $N_c = 8$ to be a good balance of variance reduction and high performance (see Fig. 11) through a small number of samples. In practice, 2 SPP is often sufficient to sample the residual term.

4.3 Integral Radiance

In this section, we discuss how to compute $F(\mathbf{i}, \mathbf{o})$ over the pixel footprint $\mathcal{P}$ efficiently. Unfortunately, this remains a high-dimensional problem. However, we observe that several terms of the scattering function $f(x(p), \mathbf{i}, \mathbf{o})$ vary slowly along $\mathcal{P}_c$, which allows us to readily reduce the dimensionality of the problem. Taking the specular reflection lobe $f^{r,s}(x, \mathbf{i}, \mathbf{o})$ as an example, we observe that the fiber-like SGGX normal distribution function $D(x, h)$, a function of the half vector h, is responsible for most of its behavior and that both the albedo $k^{r,s}(p)$ and the rest of the terms are reasonably smooth. This allows us to rewrite the Control Variates estimator in Eqn. 7 as

$$\int_{\mathcal{P}} f(p) \cdot v(p) \, \mathrm{d}p \approx \bar{k} \, \bar{G} \left(\bar{v} \cdot F + \int_{\mathcal{P}} D(p) \cdot (v(p) - \bar{v}) \, \mathrm{d}p \right), \tag{9}$$

with $\bar{k} = \int_{\mathcal{P}} k^{r,s}(p) \mathrm{d}p$, computed using mip-mapping and

$$\bar{G} = \frac{G^r(\mathbf{i}, \mathbf{o}, \bar{p})}{4 \langle \mathbf{i}, \mathbf{n_s} \rangle \langle \mathbf{o}, \mathbf{n_s} \rangle}, \tag{10}$$

with G^r the attenuation and $\bar{p}$ the center of the pixel footprint, and most importantly

$$F = \int_{\mathcal{P}} D(x(p), h) \mathrm{d}p, \tag{11}$$

which reduces the problem of efficiently computing the base term to a 2D problem on half-vector space.

This makes our problem similar to glints rendering. Different to existing works on glints, where the aggregation is done over perfect specular facets, cloth presents a harder challenge, since fabric BSDFs are complex radiance distributions in the directional space, instead of delta functions. Besides, while each texel is typically associated with a surface-like microfacet normal for glints rendering in materials like metal or glass, for cloth, each texel is associated with *many* (or rather an infinite number of) fiber-like microflake normals. Thus, we cannot directly use glints rendering methods to solve F, as discussed in Sec. 2.

Instead, we propose a method called *Integral Radiance*, taking inspiration from *Integral Histogram* (IH), used by previous glints rendering work [Gamboa et al. 2018; Atanasov et al. 2021]. In contrast to storing histograms, as is done for glints rendering, we employ a Summed-Area Table (SAT)-like structure at the spatial domain to store radiance values. By doing so, we can determine the integral of the radiance over any arbitrary range of pixel footprint coverage using only four constant cost queries. Specifically, as a 4D position-angular radiance distribution, F is organized as a 2D×2D grid table. In the directional (or half-vector) space, we create a 32^2 grid of 2D directional bins that are uniformly distributed in solid angle. Meanwhile, in the positional (or texel) space, we store SAT tables with same resolution as our texture map, which is 256^2. We selected this directional resolution based on the scattering distribution of typical garments. In practice, as Zhu et al. [2023a] noted, low roughness values can be observed in certain silk satin fabrics, corresponding to a SGGX normal distribution with roughness value of 0.04. Following the 2-sigma rule, the directional bin resolution of 32^2 is adequate to handle this degree of roughness.

As a precomputation step, we populate the SAT tables by calculating the radiance (more specifically, value of $D(p, h)$) for each grid with corresponding half vector h and texel position p. During rendering, we first identify the corresponding direction bin based on the half vector. Upon retrieving the associated SAT table, we compute F by querying the four corners of the corresponding pixel footprint, achieving constant time complexity. To eliminate banding artifacts, we additionally bilinearly interpolate the results from the four nearest directional bins. The details of precomputing and rendering the other lobes are provided in the supplementary materials.

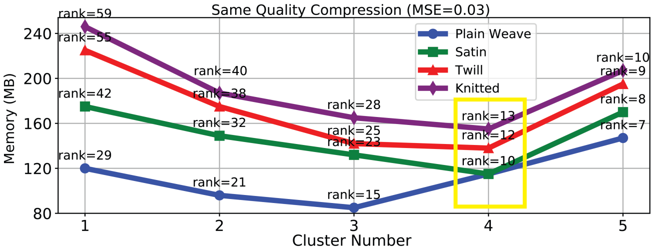

Figure 3: *C-PCA* Rank and storage by the number of clusters for the same quality compression for four of our fabrics, each original fabric taking up to 13 GB of storage. The graph illustrates the optimal combination of cluster number (4) and size of the SAT (rank 10) that results in the smallest memory footprint (115 MB).

4.4 Compression

Similar to other IH-based approaches, our method also has large memory requirements. It can amount to several gigabytes (GB) of storage for each type of cloth (for 8-clustered SAT, the storage can reach 13 GB), posing a challenge to its practicality, especially when various kinds of cloth are present in a scene. However, directly compressing SAT data presents a challenge due to its inherent disorder. To overcome this, we first normalize the SAT data and then use Clustered Principal Component Analysis (C-PCA) [Liu et al. 2004] for compression.

Normalization for SAT involves two steps. The first step addresses the cumulative nature of SATs, resulting from the incremental summing process. This cumulative characteristic often hampers the efficiency of compression, as illustrated in Fig. 4. To alleviate this, we calculate the average value of the original dataset and construct an SAT corresponding to this average. By taking the difference between the original SAT with this average SAT, we normalize the data in a way that emphasizes deviations from the average. Following this step, we then map the data onto the interval $[0, 1]$. Fig. 4 demonstrates the importance of normalization in achieving superior compression quality.

Then we utilized C-PCA, an effective rank reduction tool to compress the normalized data. Optimal number of clusters depends on the specific dataset. Increasing the number of clusters typically reduces the noise, while introducing additional storage. We experimented with several garments and found that, with a fixed target compression quality (MSE=0.03), 4 clusters typically takes the least amount of storage for our normalized SAT data, as illustrated in Fig. 3. We rendered with all our results with this configuration, achieving a compression ratio of 113:1, demonstrating the efficiency of our compression method while preserving data fidelity. Besides, our evaluation has shown that only an additional 2% of computation is necessary for data decompression during run time for offline rendering.

5 IMPLEMENTATION

5.1 Importance Sampling

To enhance the efficiency of our model under complex lighting conditions, such as environment light, we have developed an importance sampling strategy. Our cloth model is based on a BSDF framework and includes several components like sheen, delta transmission, surface reflection, surface transmission, and a blended diffuse term.

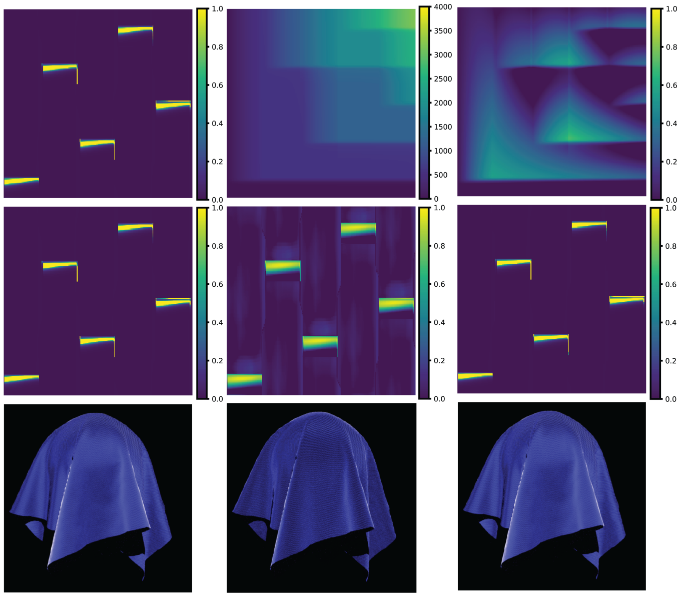

Figure 4: *SAT normalization.* Comparisons of the original radiance and resulting rendering (first column) to the SAT and the normalized SAT (second and third columns). From top to bottom: SAT visualizations, reconstructed 4D tables, and renderings. Both SAT and normalized SAT achieve very high compression ratios (115 MB from 13GB uncompressed), the later better matching the original radiance.

It can be summarized in the following stages:

- A given outgoing direction, ω_r.
- Compute attenuations, $A(p)$, for each lobe p.
- Select a lobe in proportion to the specular attenuations $A(p)$. In this step, calculate the pdf as $p_a = \frac{A(p)}{\sum_{p=1}^{P}(p)}$.
- Perform importance sampling on the selected lobe to obtain an incident direction ω_i. Calculate the lobe’s value $f_p(\omega_i, \omega_r, x)$, and in this step, we determine the pdf of sampling the lobe, p_l.
- Return a sample weight calculated as $w = \frac{f_p(\omega_i, \omega_r, x)}{p_a \cdot p_l}$.

As shown in Fig. 5, our importance sampling strategy significantly enhances efficiency under complex environmental lighting conditions.

5.2 Parallax Transition

The parallax offset proposed in Sec.3.2 is effective when a pixel footprint is smaller than a texel. However, when there are multiple texels within a pixel, the directional per-texel offsets can vary significantly. Given the low pixel sample count nature of our multi-scale method, we opt to use the averaged offset within each pixel. We adopt a similar strategy to Integral Radiance, as described in Sec.4.3, by precomputing per-texel offsets and storing them in SATs. This allows for efficient querying and calculation of the averaged per-pixel offset during render time. This enables us to render patches

Table 1: Statistics of all scenes and performance break-down.

Scene	Pattern	MSE	SPP (w/ | w/o CV)	Render Time (w/ | w/o CV)	Storage (uncompressed | compressed)
Couch (Fig. 1)	twill/plain/diamond/argyle	0.02	10 SPP | 619 SPP	25s | 19.0min	53 GB | 460 mb
Blouse (Fig. 5)	twill/satin	0.05	17 SPP | 1109 SPP	16s | 9.2min	26 GB | 230 mb
Evening gown (Fig. 10)	plain/satin	0.02	32 SPP | 1760 SPP	15s | 9min	26 GB | 230 mb
Cloth Cylinder (Fig. 8)	twill	0.04	2 SPP | 65 SPP	5s | 2min	13 GB | 115 mb
Draping Fabric (Fig. 11)	satin	0.03	2 SPP | 179 SPP	4s | 3min	13 GB | 115 mb

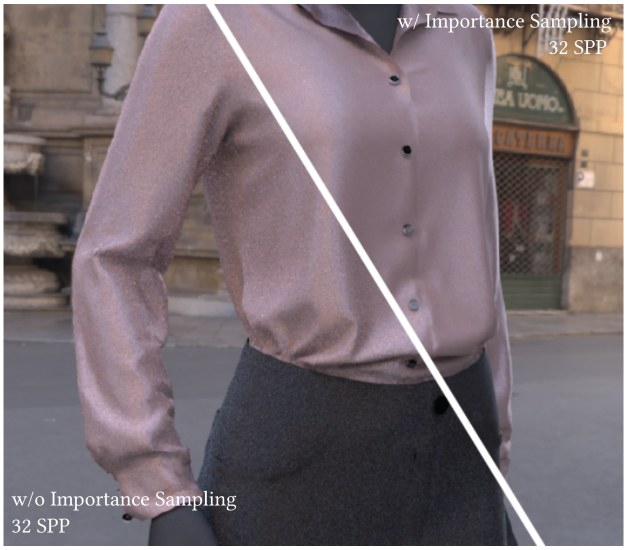

Figure 5: *Importance sampling* produces almost noise-free rendering with only 32 SPP under environment lighting. Both parts use our multi-scale model.

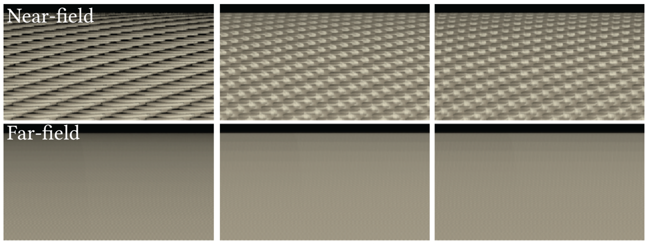

Figure 6: *Parallax effect.* Close up and far away renderings at grazing angles of the cloth cylinder, comparing the model with and without parallax offset to the ray traced ground truth. Notice how accounting for parallax better matches the reference at near view, which fixes the overly dark top boundary of the cylinder at far distance.

with accurate parallax effects while still using only a few samples per pixel and unifies the way we handle parallax for near- and far-field cases. As shown in Fig. 7, this approach eliminates undesired dark spots at close up range and produces results that are indistinguishable from brute force rendering at far distances.

6 RESULTS

We show the results of our method over different scenes to demonstrate the comprehensive nature of our complete single-point model and effectiveness of our multi-scale approach. We implemented our model in Mitsuba physically-based renderer [Jakob 2010]. Offline renderings shown in the paper are all using path tracing on an AMD Ryzen 9 7950X 16-Core (4.50 GHz) machine. In all cases, the feature maps have a resolution of 256×256. Unless otherwise specified, for each fabric, we precompute a set of 8-clustered SAT tables with a resolution of 256×256 for each of the 32×32 directional bins. The data is then compressed to a total size of 115 MB with 4 C-PCA clusters before rendering. A summary of reconstruction quality, performance and storage requirements is displayed in Table 1.

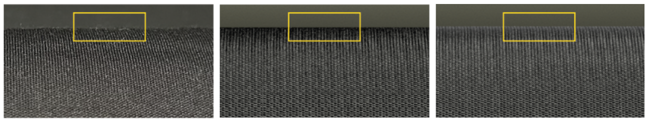

Figure 7: *Sheen effect.* Comparison of sheen effects on a black fabric. By contrasting with an actual photo of cloth (measurement), the sheen significantly enhances the realism of the rendering.

Fabric draping over a sphere. For the validation images (Figs. 4 and 11), we decide to drape a piece of fabric over a sphere, lit under point light. This is purposely designed to enhance the visual details or features of the material so that subtle modifications, like compression rates, can be easily assessed.

Cloth cylinder. We rely on the controlled setup captures from Zhu et al. [2023a] to compare our model against a photograph of areal fabric wrapping a cylinder in Figure 8. Notice how the new model faithfully matches the reference, specially improving the results from recent work at grazing angles due to the new added sheen term and the parallax offset.

Day-to-day scenes. Figures 1 and 10 showcase our model in realistic scenes. In the couch scene, we demonstrate the improvements in the appearance of the fabrics by the enhanced BSDF model, particularly important at grazing angles, and efficiently preserved at different viewing distances. Thanks to our multi scale shading technique, we produce noise free images of complex fabrics with up to 30× speedup in rendering time. In Figure 10, we present a detailed rendering of a two-layer dress at a life-like viewing distance, made of fabrics with complex anisotropic highlights, transparency, and multiple scattering effects through the cloth layers. The scene employs an additional set of SAT tables specifically for the delta transmission lobe of the purple sheer fabric.

Real-time Rendering. Our multi-scale method can be easily incorporated into a real-time engine. In Figure 9, we showcase this by integrating our cloth shader into a real-time path tracer using the Falcor rendering framework [Kallweit et al. 2022]. As a proof of concept, we implemented the specular reflection lobe of the base term defined in Sec. 4.3, utilizing the same SAT configuration as in other results. For simplicity, we used a single CCV cluster and

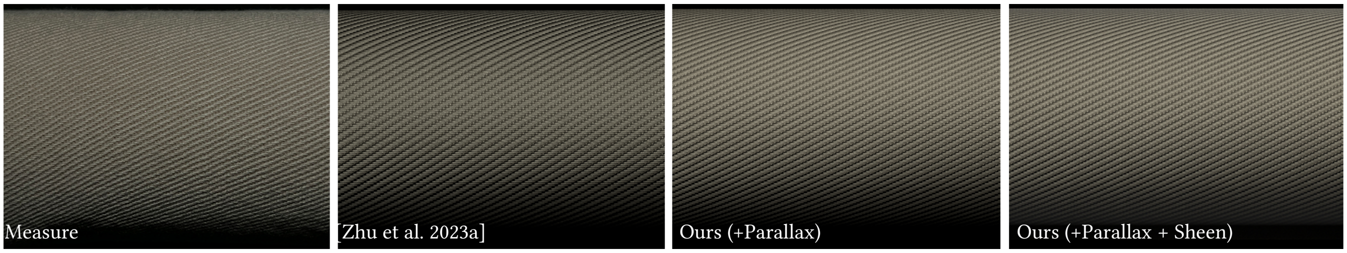

Figure 8: *Reproducing reflectance of a real cloth sample.* Comparison against a photograph to showcase the improvement of our model also in close up scenarios thanks to the inclusion of parallax offset and sheen, specially noticeable at grazing angles, now brighter and correct. Please zoom in to avoid display alias.

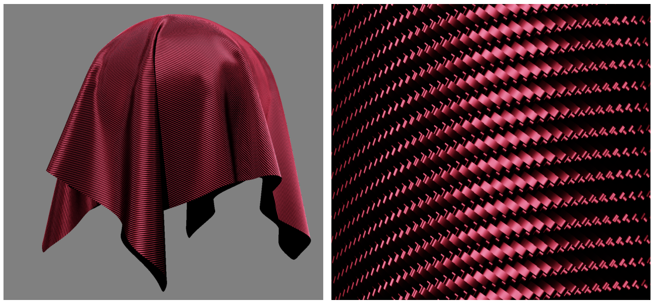

Figure 9: Real-time rendering of an approximation of our model. The rendering is completed in 8.3 milliseconds for both images.

uncompressed SAT data. The supplemental video shows a smooth and noise-free transition from varying viewing distances rendered with 2 samples per pixel (SPP) at approximately 120 frames per second on an Nvidia RTX 4090 GPU. Our goal is to showcase the potential of our method in a real-time application. We expect that by including the Monte Carlo sampled residual term and its spatial temporal denoising, we will achieve a complete real-time solution. However, this is left to future work.

7 CONCLUSION AND LIMITATIONS

We presented a multi-scale surface-based cloth appearance model that alleviates the trade-off between realism and performance. Our model consists of two main components–a comprehensive BSDF model for accurate parallax effects and light transport at grazing angles, and an efficient multi-scale integration scheme to losslessly preserve accurate highlights and self-shadowing. Our novel Clustered Control Variates (CCV) scheme and Summed-Area Table (SAT) compression method improve practicality and performance, achieving a 30× acceleration compared to the state-of-the-art. We demonstrated its realism and efficiency on a variety of cloth appearance, as well as its potential for real-time applications and inverse tasks.

Our method still has some limitations that are left for future work. First, our model requires pre-clustering of the cloth's visibility, followed by the generation and compression of several SATs for each fabric pattern. Therefore, any change of parameters necessitates a pre-computation time of several minutes, making arbitrary on-the-fly edits difficult for artists. However, it is often the case that albedos change dynamically in practice, whereas fabric patterns or roughness remain relatively constant, and our method already supports this dynamic behavior of albedos. Second, our method lacks fiber-level geometric information, which becomes particularly evident in extremely close-up shots, especially for fly-away fibers. Third, our approach is defined on the surface, suitable for representing woven cloth and thin knitted cloth, such as t-shirts. However, it cannot capture the silhouette effects of thicker knitted items (like sweaters) at grazing angles, a limitation that is shared with other surface-based methods that do not have explicit surface displacements.

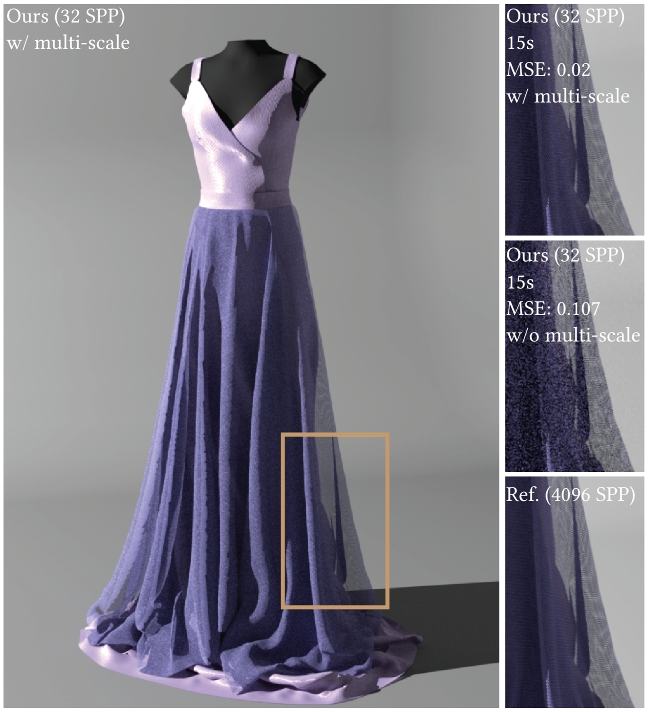

Figure 10: Evening gown made of an overlay of two fabrics: a pink satin and a purple sheer. This demonstrates the ability of our technique to handle transparency and multiple scattering through the cloth layers.

ACKNOWLEDGMENTS

We thank Olivier Maury and Naty Hoffman for their valuable input. We also thank Natasha Devaud for her contribution in authoring procedural tools, generating ply-level feature maps, as well as her guidance on the garment selection process. This project is solely sponsored by Meta. And Ling-Qi Yan is also supported by gift funds from Adobe, Intel, Linctex, Meta and XVerse.

REFERENCES

Neeharika Adabala, Nadia Magnenat-Thalmann, and Guangzheng Fei. 2003. Real-Time Rendering of Woven Clothes *(VRST '03)*. Association for Computing Machinery, New York, NY, USA, 41–47. https://doi.org/10.1145/1008653.1008663

Asen Atanasov, Alexander Wilkie, Vladimir Koylazov, and Jaroslav Křivánek. 2021. A multiscale microfacet model based on inverse Bin mapping. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 103–113.

Petrik Clarberg and Tomas Akenine-Möller. 2008. Exploiting visibility correlation in direct illumination. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1125–1136.

Miguel Crespo, Adrian Jarabo, and Adolfo Muñoz. 2021. Primary-space adaptive control variates using piecewise-polynomial approximations. *ACM Transactions on Graphics (TOG)* 40, 3 (2021), 1–15.

Jonathan Dupuy, Eric Heitz, Jean-Claude Iehl, Pierre Poulin, Fabrice Neyret, and Victor Ostromoukhov. 2013. Linear Efficient Antialiased Displacement and Reflectance Mapping. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 32, 6 (2013), 211:1–211:11.

Luis E Gamboa, Jean-Philippe Guertin, and Derek Nowrouzezahrai. 2018. Scalable appearance filtering for complex lighting effects. *ACM Trans. Graph.* 37, 6 (2018), 277–1.

Charles Han, Bo Sun, Ravi Ramamoorthi, and Eitan Grinspun. 2007. Frequency Domain Normal Map Filtering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 26, 3 (2007), 28:1–28:12.

Eric Heitz, Jonathan Dupuy, Stephen Hill, and David Neubelt. 2016. Real-time polygonal-light shading with linearly transformed cosines. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–8.

Christophe Hery. 2018. Illumination 101. In *MCQMC 2018*. https://graphics.pixar.com/library/MCQMC2018/paper.pdf

Christophe Hery, Michael Kass, and Junyi ling. 2014. Geometry into Shading. https://graphics.pixar.com/library/BumpRoughness/paper.pdf.

Piti Irawan and Steve Marschner. 2012. Specular reflection from woven cloth. *ACM Trans. Graph.* 31, 1 (2012), 1–20.

Wenzel Jakob. 2010. Mitsuba renderer. http://www.mitsuba-renderer.org.

Wenhua Jin, Beibei Wang, Milos Hasan, Yu Guo, Steve Marschner, and Ling-Qi Yan. 2022. Woven Fabric Capture from a Single Photo. In *SIGGRAPH Asia 2022 Conference Papers*. 1–8.

Simon Kallweit, Petrik Clarberg, Craig Kolb, Tom'aš Davidovič, Kai-Hwa Yao, Theresa Foley, Yong He, Lifan Wu, Lucy Chen, Tomas Akenine-Möller, Chris Wyman, Cyril Crassin, and Nir Benty. 2022. The Falcor Rendering Framework. https://github.com/NVIDIAGameWorks/Falcor https://github.com/NVIDIAGameWorks/Falcor.

Tomomichi Kaneko, Toshiyuki Takahei, Masahiko Inami, Naoki Kawakami, Yasuyuki Yanagida, Taro Maeda, and Susumu Tachi. 2001. Detailed shape representation with parallax mapping. In *Proceedings of ICAT*, Vol. 2001. 205–208.

Alexandr Kuznetsov. 2021. NeuMIP: Multi-resolution neural materials. *ACM Transactions on Graphics (TOG)* 40, 4 (2021).

Xinguo Liu, Peter-Pike J Sloan, Heung-Yeung Shum, and John Snyder. 2004. All-Frequency Precomputed Radiance Transfer for Glossy Objects. *Rendering Techniques* 2004 (2004).

Zahra Montazeri, Soren B Gammelmark, Henrik Wann Jensen, and Shuang Zhao. 2021. A Practical Ply-Based Appearance Modeling for Knitted Fabrics. In *Proceedings of Eurographics Symposium on Rendering 2021*.

Zahra Montazeri, Søren B Gammelmark, Shuang Zhao, and Henrik Wann Jensen. 2020. A practical ply-based appearance model of woven fabrics. *ACM Trans. Graph.* 39, 6 (2020), 1–13.

Thomas Müller, Fabrice Rousselle, Alexander Keller, and Jan Novák. 2020. Neural control variates. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–19.

Jan Novák, Iliyan Georgiev, Johannes Hanika, and Wojciech Jarosz. 2018. Monte Carlo Methods for Volumetric Light Transport Simulation. *Computer Graphics Forum (Proceedings of Eurographics - State of the Art Reports)* 37, 2 (2018).

Marc Olano and Dan Baker. 2010. LEAN Mapping. *Proceedings of the Symposium on Interactive 3D Graphics and Games* (2010), 181–188.

Iman Sadeghi, Oleg Bisker, Joachim De Deken, and Henrik Wann Jensen. 2013. A practical microcylinder appearance model for cloth rendering. *ACM Trans. Graph.* 32, 2 (2013), 1–12.

Kai Schroder, Reinhard Klein, and Arno Zinke. 2011. A volumetric approach to predictive rendering of fabrics. *Computer Graphics Forum* 30, 4 (2011), 1277–1286.

László Szécsi, Mateu Sbert, and László Szirmay-Kalos. 2004. Combined correlated and importance sampling in direct light source computation and environment mapping. In *Computer Graphics Forum*, Vol. 23. Wiley Online Library, 585–593.

Beibei Wang, Wenhua Jin, Miloš Hašan, and Ling-Qi Yan. 2022. SpongeCake: A Layered Microflake Surface Appearance Model. *ACM Trans. Graph.* 42, 1 (2022), 1–16.

Xi Wang, Xin Tong, Stephen Lin, Shimin Hu, Baining Guo, and Heung-Yeung Shum. 2004. Generalized displacement maps. In *Eurographics Symposium on Rendering (EGSR)*. 227–233.

Hongzhi Wu, Julie Dorsey, and Holly Rushmeier. 2009. Characteristic Point Maps. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)* 28, 4 (2009), 1227–1236.

Chao Xu, Rui Wang, Shuang Zhao, and Hujun Bao. 2019. Multi-scale hybrid micro-appearance modeling and realtime rendering of thin fabrics. *IEEE transactions on visualization and computer graphics* 27, 4 (2019), 2409–2420.

Kun Xu, Wei-Lun Sun, Zhao Dong, Dan-Yong Zhao, Run-Dong Wu, and Shi-Min Hu. 2013. Anisotropic Spherical Gaussians. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 32, 6 (2013), 209:1–209:11.

Ling-Qi Yan, Miloš Hašan, Steve Marschner, and Ravi Ramamoorthi. 2016. Position-Normal Distributions for Efficient Rendering of Specular Microstructure. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 35, 4 (2016), 56:1–56:9.

Ling-Qi Yan, Miloš Hašan, Wenzel Jakob, Jason Lawrence, Steve Marschner, and Ravi Ramamoorthi. 2014. Rendering Glints on High-Resolution Normal-Mapped Specular Surfaces. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 33, 4 (2014), 116:1–116:9.

Tizian Zeltner, Brent Burley, and Matt Jen-Yuan Chiang. 2022. Practical Multiple-Scattering Sheen Using Linearly Transformed Cosines. In *ACM SIGGRAPH 2022 Talks*. 1–2.

Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. 2011. Building volumetric appearance models of fabric using micro CT imaging. *ACM Trans. Graph.* 30, 4 (2011), 1–10.

Shuang Zhao, Fujun Luan, and Kavita Bala. 2016. Fitting procedural yarn models for realistic cloth rendering. *ACM Trans. Graph.* 35, 4 (2016), 1–11.

Junqiu Zhu, Adrian Jarabo, Carlos Aliaga, Ling-Qi Yan, and Matt Jen-Yuan Chiang. 2023a. A Realistic Surface-based Cloth Rendering Model. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–9.

Junqui Zhu, Zahra Montazeri, J Aubry, Lingqi Yan, and Andrea Weidlich. 2023b. A Practical and Hierarchical Yarn-based Shading Model for Cloth. In *Computer Graphics Forum*, Vol. 42. Wiley Online Library, e14894.

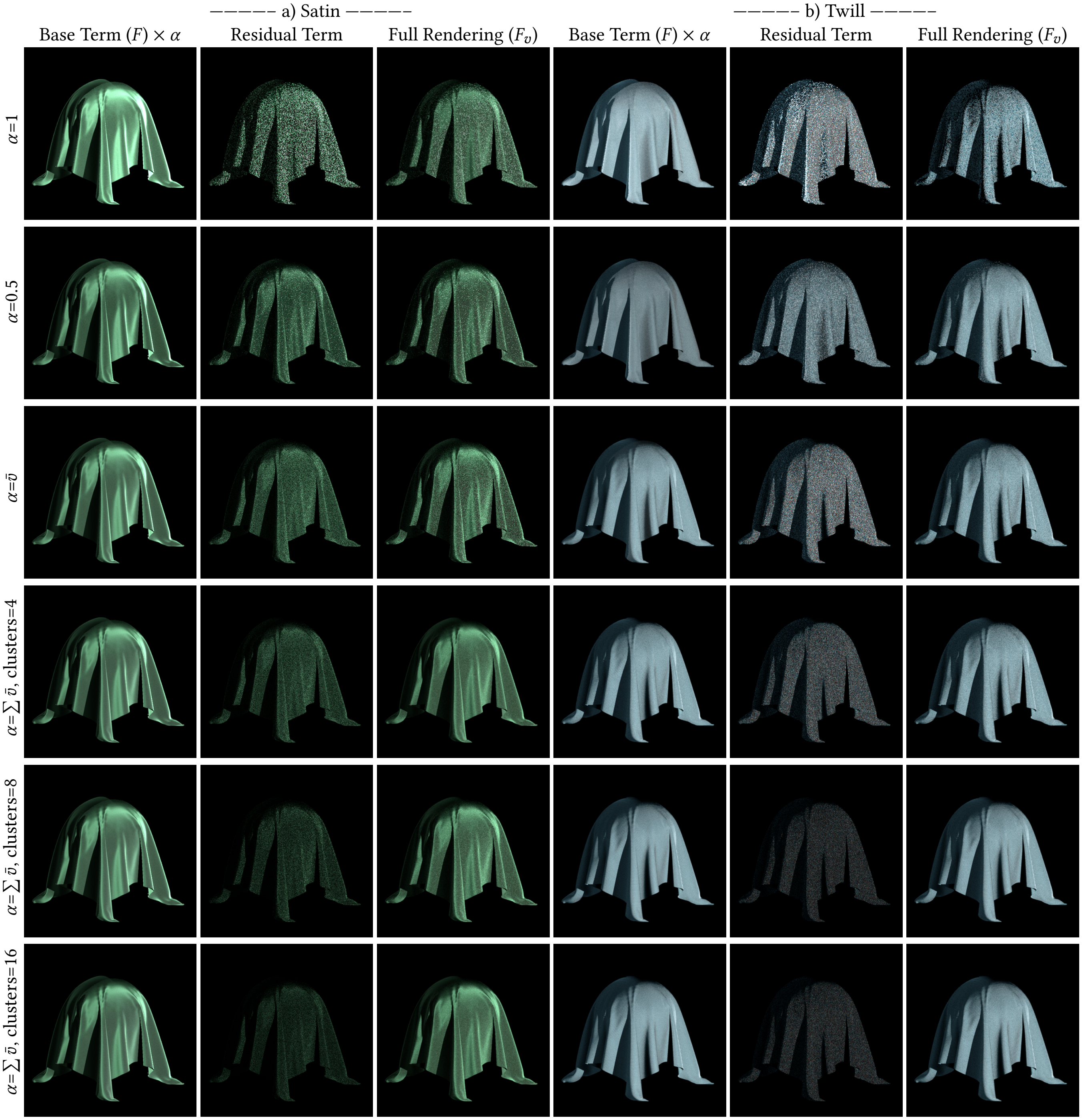
a) Satin
b) Twill
Base Term (F) × α
Residual Term
Full Rendering (F_v)
Base Term (F) × α
Residual Term
Full Rendering (F_v)
α=1
α=0.5
$\alpha=\bar{v}$
$\alpha=\sum\bar{v}$, clusters=4
$\alpha=\sum\bar{v}$, clusters=8
$\alpha=\sum\bar{v}$, clusters=16

Figure 11: We compare the control variates using different values of α on satin and twill, and we present the renderings of the base term $F \times \alpha$, the residual term, and the full rendering using 2 SPP. The Base Term is expected to be brighter than the Full Rendering for large values of α, since the residual can become negative.