Multi-Scale Hybrid Micro-Appearance Modeling and Realtime Rendering of Thin Fabrics

Chao Xu, Rui Wang, Shuang Zhao, and Hujun Bao*

Abstract—Micro-appearance models offer state-of-the-art quality for cloth renderings. Unfortunately, they usually rely on 3D volumes or fiber meshes that are not only data-intensive but also expensive to render. Traditional surface-based models, on the other hand, are light-weight and fast to render but normally lack the fidelity and details important for design and prototyping applications. We introduce a multi-scale, hybrid model to bridge this gap for thin fabrics. Our model enjoys both the compactness and speedy rendering offered by traditional surface-based models and the rich details provided by the micro-appearance models. Further, we propose a new algorithm to convert state-of-the-art micro-appearance models into our representation while qualitatively preserving the detailed appearance. We demonstrate the effectiveness of our technique by integrating it into a real-time rendering system.

Index Terms—rendering, micro-appearance modeling, textiles.

1 Introduction

Cloth is an important material to render at high fidelity because it is essential to our daily lives and very often important in computer graphics scenes. Further, accurate cloth appearance is crucial for applications in the textile design and manufacturing industries.

Traditionally, cloth appearances are modeled with surface-based representations such as specialized BRDFs [1], [2]. These models are usually light-weight, fast to render, and provide convincing results when a fabric is viewed from a distance with individual yarns barely visible. Unfortunately, these models lack fine-grained details needed for generating realistic results at close-up views, which are important for many applications in design and virtual reality.

Micro-appearance cloth models [3], [4], [5], [6], [7] offer superior quality by explicitly capturing a fabric's detailed structures down to the fiber-level. These micro-geometries are generally described using high-resolution volumes or fiber meshes. Thanks to the presence of fiber-level details, these models remain highly realistic even when viewed very closely. In contrast to the surface-based models, the micro-appearance models are usually data-intensive and expensive to render. Even with heavy precomputation, rendering a single image using these models can take minutes on the CPU [8] and tens of seconds on the GPU [9].

In this paper, we bridge the gap between the two families of techniques by introducing a multi-scale hybrid micro-appearance model for thin fabrics. Our model enjoys the light-weight and fast rendering offered by traditional surface-based models and the high fidelity and details provided by the micro-appearance models. Our method provides the following advantages over existing techniques. First, compared to traditional surface-based methods, our model captures fiber-level details and thus remains highly realistic even when viewed extremely closely. Second, compared to previous micro-appearance models, our model is light-weight and can be rendered and edited in realtime. Further, our model can be efficiently prefiltered, resulting in a multi-scale representation that can be rendered at greatly varying resolutions without heavy super-sampling.

Technically, our contributions include:

- We present a new multi-scale micro-appearance model to describe a thin fabric's fiber-level geometric and optical properties (§4). Our model involves a hybrid representation by coupling a thin volume expressing flyaway fibers and a normal-mapped height field capturing the fabric's main surface comprised of regular fibers.
- We propose a new model fitting algorithm that fits our representation to existing micro-appearance models (§5). We also show how our models can be prefiltered for multi-scale rendering (§5.3.5).
- We develop an end-to-end pipeline capable of rendering our hybrid models at multiple scales in real-time (§6).

2 Related Work

Surface-based cloth models. Cloth rendering has been an active research topic in graphics for decades since fabrics are important to many virtual scenes. Traditionally, fabrics are treated as smooth 2D surfaces described with polygonal meshes, and the light-cloth interaction is captured by surface-based reflectance profiles (e.g., [1], [2]). These models are light-weight and have produced high-quality results. On the other hand, the surface-based models usually lack small-scale details that are crucial for creating realistic

- *C. Xu is with the Department of Computer Science, Zhejiang University. E-mail: xskyer@zju.edu.cn*
- *R. Wang is with the Department of Computer Science, Zhejiang University. E-mail: rwang@cad.zju.edu.cn*
- *S. Zhao is with the Department of Computer Science, University of California, Irvine. E-mail: shz@ics.uci.edu*
- *H. Bao is with the Department of Computer Science, Zhejiang University. E-mail: bao@cad.zju.edu.cn. He is the corresponding author*

renderings at the near field. This level of fidelity and details is important for applications in textile design and virtual prototyping.

Micro-appearance cloth models. A family of *micro-appearance modeling* techniques have been developed recently [3], [4], [7], [10], [11]. These methods explicitly describe fabrics' small-scale geometries down to the fiber level and have reached a new level of fidelity and details. However, due to the high complexity of these models, rendering them have been very challenging. To address this problem, several precomputation based techniques have been developed. Zhao et al. [8] introduced a technique that exploits the modularity of large models by precomputing light transport on a small collection of basic building blocks and efficiently combining such information on the fly. Wu and Yuksel [12] developed a GPU-based technique that interactively renders fabrics with procedurally described fiber-level details by leveraging 'core fibers': combined bundles of regular fibers with baked in shadowing and ambient occlusion information. Khungurn et al. [9] extended precomputed radiance transfer [13] to handle micro-appearance cloth models. By decoupling single and multiple scattering and approximating the latter using precomputed visibility and indirect radiance information, very good performance can be achieved with low accuracy loss.

Unfortunately, these previous methods either lack interactive performance [8], [9] or physical accuracy [12] for predictively rendering and editing micro-appearance cloth models at realtime.

Data-driven models. Cloth appearance can also be modeled in data-driven ways, such as using Bidirectional Texture Functions (BTFs) [14]. However, these representations are cumbersome and difficult to manipulate. Further, they are normally too data-intensive to capture the level of details offered by micro-appearance models.

Fiber scattering models. The reflectance profile used by our model is inspired by various fiber scattering models [4], [15], [16], [17]. These models utilize bidirectional curve scattering distribution functions (BCSDFs) to describe how light scatters off individual fibers. These functions generally involve a few modes capturing light-fiber interactions with varying numbers of reflections and transmissions. The fiber scattering models were originally developed for rendering hair and fur and adopted by Khungurn et al. [4] for cloth rendering.

Model creation methods. Previously, a few techniques have been developed to reverse engineer a fabric's weave pattern and optical properties using single images as input (e.g., [6], [18])svn. These methods focus on analyzing fabric properties based on real photographs and output traditional surfaced-based or micro-appearance models.

3 Background

We now provide a brief recap on volumetric micro-appearance models, which will be used as input in our model fitting pipeline (§5).

Anisotropic radiative transfer. Jakob et al. [19] has introduced a generalization to the standard radiative transfer framework [20] for better handling anisotropic media with directionally varying structures. Under this framework, the extinction σ_t and scattering coefficient σ_s are both directionally dependent, and the phase function f_p takes its general 4D form (instead of depends only on the angle between incoming and outgoing directions).

Modeling fabrics as anisotropic media. Volumetric micro-appearance cloth models describe fabrics as anisotropic media. To conveniently specify the required radiative transfer parameters (i.e., σ_t, σ_s, and f_p), the *micro-flake model* is introduced. This model uses its own set of spatially varying parameters to determine σ_t, σ_s, and f_p including (i) flake density ρ; (ii) distribution D of flake normals; and (iii) albedo α.

At the core of the micro-flake framework is the flake normal distribution D which plays a similar role as the normal distribution function (NDF) to micro-facet reflectance models. In practice, this distribution can be specified using a number of function basis including spherical harmonics [19], Gaussians [3], and SGGX [21]. In this paper, we use the Gaussian representation (although our technique is not limited to this particular formulation). In this case, D becomes a spherical Gaussian with user specified mean $\boldsymbol{\omega}_f$ and standard deviation σ.

4 Our Hybrid Model

Our technique takes as input volumetric micro-appearance fabric models [3], [4], [21] that capture a fabric's fine-grained structures down to the fiber level. Unlike previous work that directly renders these models by explicitly simulating anisotropic radiative transfer [19], we convert them into a *hybrid representation* that involves:

- A normal-mapped height field capturing a fabric's main surface comprised of tightly packed regular fibers and a thin volume capturing the flyaway fibers which cannot be accurately described using the height field (§4.1);
- Light-weight light scattering models expressing how light interacts with the fabric's main surface as well as flyaway fibers (§4.2).

Our hybrid model can be rendered at realtime, enabling many applications in design, retail, and entertainment. In the rest of this section, we provide more details on our hybrid model description.

4.1 Geometric Description

Real-world cloth fibers are usually classified into two categories: regular and flyaway [6], [7]. Regular fibers are generally tightly packed to form the main body of a fabric. Flyaways, in contrast, go beyond the main body and contribute significantly to a fabric's hairy appearance.

Based on this understanding, our hybrid model leverages a height field to capture a fabric's top surface formed with tightly packed regular fibers and a thin volume to describe the flyaways (Figure 1).

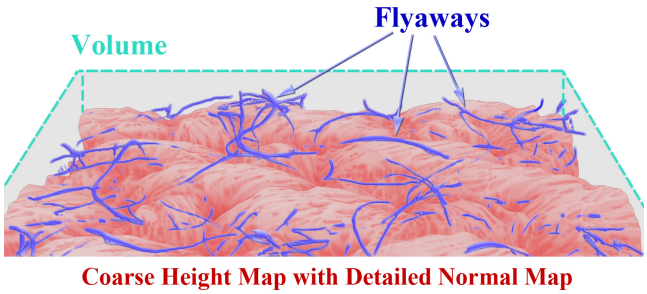


Fig. 1: Our hybrid model involves a normal-mapped height field and a thin volume capturing a fabric's main surface and flyaway fibers, respectively.

Unfortunately, detailed height fields can be challenging to render since they can cause severe aliasing when rendered from afar. In addition, they cannot be easily pre-filtered since doing so changes the model's intrinsic shadowing and masking structures, leading to complex overall appearance changes that are challenging to fix [22]. To address this problem, we decompose the detailed height field into a smoother version and a normal map. The normal map, combined with our light scattering model, is further prefiltered to enable efficient multi-scale rendering of our model (§4.2). In practice, we apply a low-pass filtering to the height map so that only yarn-level structures are preserved while fiber-level details are removed. The effectiveness of this step is further demonstrated in §7.1.

4.2 Light Scattering Model

The other major component of our technique is a new light scattering model inspired by previous work [2], [4]. At each surface point $\boldsymbol{x}$, assuming the tangent $\boldsymbol{t}(\boldsymbol{x})$ is aligned with local fiber direction, given incoming and outgoing directions $\boldsymbol{\omega}_i, \boldsymbol{\omega}_o$, we use a BRDF given by

$$f_r(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = k_d + F(\eta, \theta_i)\, \cos(\varphi_i)^a\, f_{r,s}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o), \quad (1)$$

where k_d denotes the diffuse coefficient, F is the Fresnel term, $\cos(\varphi_i)^a$ captures the shadowing effect [2], and $f_{r,s}$ is the multi-lobe specular term given by

$$f_{r,s}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \sum_j \frac{k_j}{\sqrt{2\beta_j \pi}} \exp\left(-\frac{(\theta_i + \theta_o)^2}{2\tilde{\beta}_j^2}\right), \quad (2)$$

where j is the lobe index[1], and

$$\tilde{\beta}_j = \exp\left(-\frac{(\varphi_i + \varphi_o)^2}{2\gamma_j^2}\right). \quad (3)$$

In Eqs. 1 and 2, θ_i and θ_o are respectively the angles from $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$ to the normal plane, φ_i and φ_o are given by the projections of $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$ in the normal plane (see Figure 2).

Eq. 2 describes specular reflectance and involves a number of Gaussian-like functions where the dependency on $(\theta_i + \theta_o)$ is inspired by the R mode of the bidirectional curve scattering distribution function (BCSDF) used previous for rendering fibrous structures [4], [15]. We did not include higher-order modes (e.g., TRT) of those BCSDFs since our model relies on the diffuse term to capture multiple scattering. We further introduce a Gaussian $\tilde{\beta}_j$ on $(\varphi_i + \varphi_o)$ to better handle the change of reflectance with φ (see Figure 6).

Note that our BRDF expressed in Eq. 1 is shared by both the base surface h and the thin volume V. For the latter, the BRDF is applied to the implicit surface derived from the volume (see §6 for more details).

Our model can be pre-filtered, leading to a multi-scale representation that can be efficiently rendered without heavy super-sampling. In the following section, we introduce a novel technique to convert input micro-appearance models to our hybrid representation at varying scales (§5).

5 Fitting Model Parameters

We now introduce an end-to-end pipeline that fits our hybrid representation to input micro-appearance models described in §3. Our fitted models can be rendered in realtime and produces appearances closely resembling the input. An overview of our technique is shown in Figure 3.

5.1 Input

The input to our model fitting pipeline is a volumetric representation that describes a flat fabric sample using the following spatially varying properties:

- material density ρ,
- single-scattering albedo α,
- material orientation $\boldsymbol{\omega}_f$.

The material orientation, when coupled with a roughness parameter γ (which is assumed to be spatially invariant), in turn determines the material's phase function via microflake [3], SGGX [21], or fiber-driven [4] models, which are all compatible with our method. When rendered using the anisotropic radiative transfer framework introduced by Jakob et al. [19], the volumetric models can offer similar quality compared to fiber-based representations [7], [12], [23], another common approach to model fabric appearance. This has been demonstrated by Khungurn et al. [4].

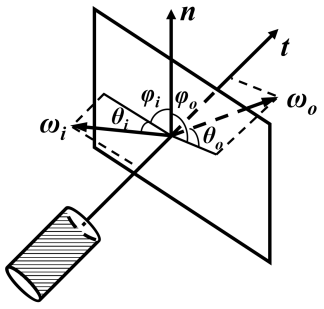


Fig. 2: Definitions of the angles used by our BRDF model 1. Under this configuration, both θ and φ angles vary from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$. When $\theta = 0$, the corresponding angle $\boldsymbol{\omega}$ lies within the normal plane; when $\varphi = 0$, the projection of $\boldsymbol{\omega}$ normalizes to $\boldsymbol{n}$.

1. In our implementation, we use two lobes at the finest LOD level and more at higher levels (see §5.3.5).

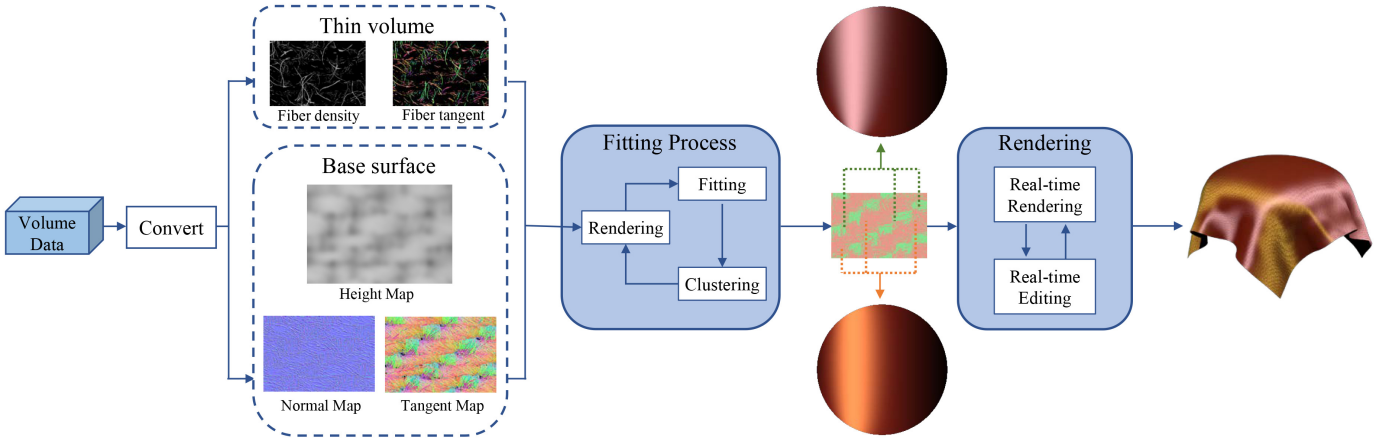


Fig. 3: Our model fitting pipeline: taking volumetric models as input, our technique starts from extracting sub yarn-level geometric information represented as height, normal, and tangent maps. Then, our model fitting algorithm (Algorithm 1) computes spatially varying reflectance profile. Our resulting model closely resembles the input in appearance, and is much lighter weighted and can be rendered in real-time.

In practice, all the three spatially varying quantities (i.e., ρ, α, and $\boldsymbol{\omega}_f$) are represented as regular 3D grids with identical resolutions $s_x \times s_y \times s_z$. Further, we assume Z+ to be the "up" direction. In other words, s_z captures the sample's thickness, and $z = s_z$ corresponds to its front surface.

Given the input data, we then construct our hybrid model by fitting model geometry (§5.2) and light scattering properties (§5.3).

5.2 Fitting Model Geometry

As stated in §4, our model uses a hybrid representation that leverages (i) a smooth height field, denoted as h, for describing a fabric's main surface at the yarn level; (ii) a detailed normal map $\boldsymbol{n}$ capturing the fabric's fiber-level details; and (iii) a thin volume, denoted as V, for capturing the flyaways. In practice, we express h and $\boldsymbol{n}$ as 2D arrays with resolution $s_x \times s_y$ in which each pixel $h[x, y]$ and $\boldsymbol{n}[x, y]$ describes the height and normal values at (x, y), respectively. The thin volume V is expressed using a regular 3D grid with resolution $s_x \times s_y \times d$ for some $d \ll s_z$ in which each voxel stores local material density and orientation information, similar to the input representation.

To extract h, $\boldsymbol{n}$ and V from the input model, we start with computing a detailed height map h_0 as well as the thin volume V. To this end, we take for each (x, y) a stacks of voxels $\rho[x, y, :]$ from the input density field. Starting with $z = s_z$, we iteratively decrease z until reaching a nonempty voxel (i.e., one with $\rho[x, y, z] > 0$). We then check the number of consecutive voxels that are nonempty. If the voxel count is greater than the size of a single fiber, we set $h_0[x, y] = z$; otherwise, we skip to the next empty voxel and continue decreasing z. After the height field h_0 is constructed, the volume V simply contains the regions of ρ and $\boldsymbol{\omega}_f$ that are not captured by h_0.

After obtaining h_0, we further decompose it into the smooth height filed h and detailed normal map $\boldsymbol{n}$. Since our goal is to have h to capture mostly yarn-level geometry of a fabric, we apply anisotropic Gaussian filtering to h_0, resulting in a smooth version h in which details smaller than a single yarn are removed. Lastly, we compute the normal map $\boldsymbol{n}$ that captures fiber-level details based on the difference between h and h_0 (i.e., $h_0 - h$).

For input models with well-defined patterns, we determine the filter size automatically using a technique developed in textile research [24]. Models with disorganized fibers (e.g., felt), on the other hand, are largely random and insensitive to the exact amount of smoothing. Thus, we use one predetermined filter size in this case. Given the detailed height map h, we compute the following two vectors $\boldsymbol{C}_0$ and $\boldsymbol{C}_1$ for all x and y:

$$\boldsymbol{C}_0[x] := \sum_{i,j} h[i, j]\, h[i-x, j], \quad \boldsymbol{C}_1[y] := \sum_{i,j} h[i, j]\, h[i, j-y], \tag{4}$$

where h is assumed to be infinite and periodic. As demonstrated by Kang et al. [24], when h is patterned, $\boldsymbol{C}_0$ and $\boldsymbol{C}_1$ will peak at values that are multiples of the size of the pattern's smallest repetitive structure, which usually corresponds to individual yarns for cloth. Thus, we pick the first peak location for both $\boldsymbol{C}_0$ and $\boldsymbol{C}_1$, and set our filter size to the smaller of the two. See Figure 4 for an example.

5.3 BRDF Fitting

With the height field h and normal map $\boldsymbol{n}$ extracted, we then fit the BRDF parameters used in Eq. 1 based on a set of training renderings generated using the input model. To avoid double counting the flyaways, we remove them from the input model by taking the complement of V. BRDF fitting for the flyaways will be discussed in §5.3.4.

Due to the large number of texels (in millions), performing BRDF fitting individually for each texel in the height field is generally impractical. To address this problem, we partition the texels into multiple clusters (§5.3.1) and apply parameter fitting for each cluster center (§5.3.2).

5.3.1 *Texel Clustering*

We cluster the texels based on a few training renderings with sparsely sampled viewing and lighting direc-

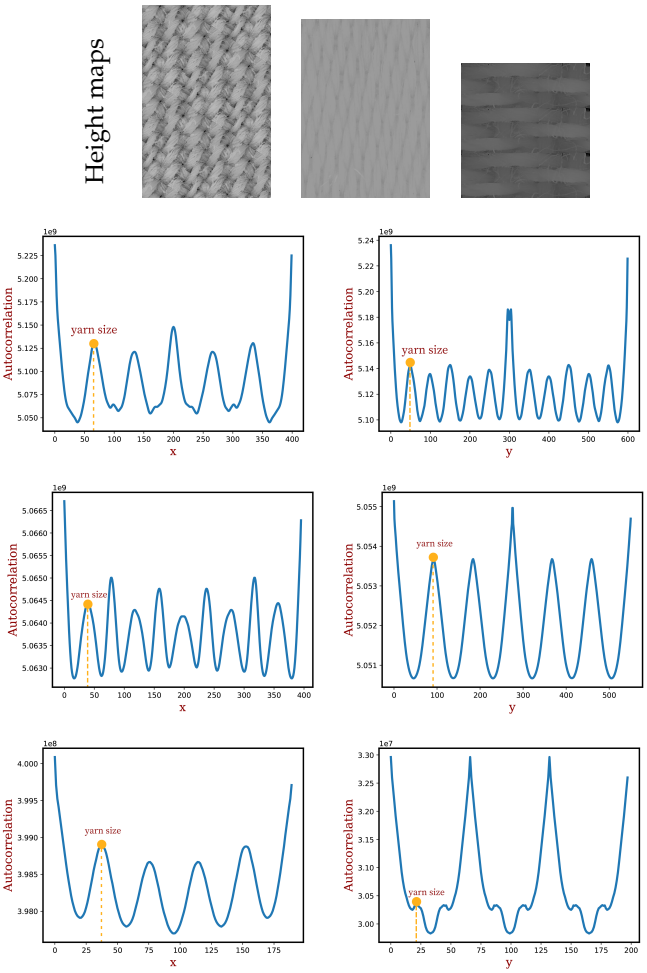


Fig. 4: Automatic selection of the size of our height-map-smoothing kernel: we select filter size automatically for fabrics with strong patterns by analyzing the height map's horizontal and vertical autocorrelations via Eq. 4. In this example, each row of plots corresponds to a height map shown on the top.

tions. Precisely, we render the input model from a predetermined set of m viewing and lighting directions $(\boldsymbol{\omega}_i^{(1)}, \boldsymbol{\omega}_o^{(1)}), \ldots, (\boldsymbol{\omega}_i^{(m)}, \boldsymbol{\omega}_o^{(m)})$. For each texel j, this gives a number of evaluations of the local reflectance in the form of $(\hat{\boldsymbol{\omega}}_i^{(j,k)}, \hat{\boldsymbol{\omega}}_o^{(j,k)}, f_r^{(j,k)})$ where $\hat{\boldsymbol{\omega}}_i^{(j,k)}$ and $\hat{\boldsymbol{\omega}}_o^{(j,k)}$ respectively equal $\boldsymbol{\omega}_i^{(k)}$ and $\boldsymbol{\omega}_o^{(k)}$ transformed into texel j's local shading frame, and $f_r^{(j,k)}$ is the corresponding BRDF value directly taken from the rendered image.

Then, we construct a feature vector for each texel j as

$$\boldsymbol{y}_j := [\boldsymbol{n}_j,\ \boldsymbol{t}_j,\ w\boldsymbol{f}_j], \tag{5}$$

where $\boldsymbol{n}_j$ and $\boldsymbol{t}_j$ are respectively the normal and tangent at texel j, $w \in \mathbb{R}_+$ is a global weight, and $\boldsymbol{f}_j$ is a vector of BRDF values taken from a set of incoming and outgoing directions from the local shading frame:

$$\boldsymbol{f}_j := [f_r^{(j)}(\hat{\boldsymbol{\omega}}_i^{(1)}, \hat{\boldsymbol{\omega}}_o^{(1)}),\ f_r^{(j)}(\hat{\boldsymbol{\omega}}_i^{(2)}, \hat{\boldsymbol{\omega}}_o^{(2)}),\ \ldots]. \tag{6}$$

Note that the local directions $(\hat{\boldsymbol{\omega}}_i^{(k)}, \hat{\boldsymbol{\omega}}_o^{(k)})$ for $k = 1, 2, \ldots$ are specified globally and generally do not equal $\hat{\boldsymbol{\omega}}_i^{(j,k)}$ and

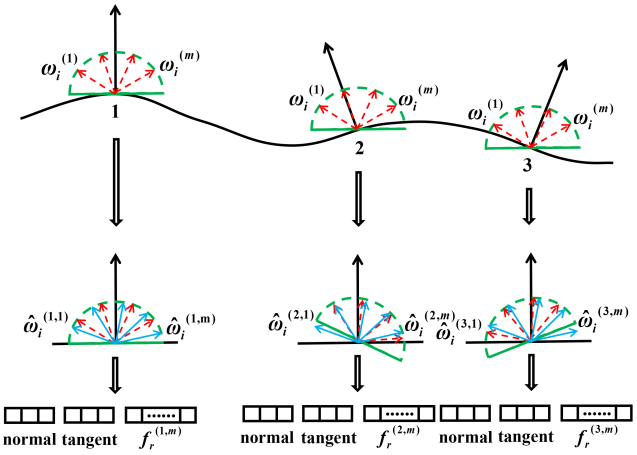


Fig. 5: Feature vectors: we construct a feature vector $\boldsymbol{y}_j$ for each texel j. We start with rendering the full input model under a fixed set of m viewing and lighting directions $(\boldsymbol{\omega}_i^{(k)}, \boldsymbol{\omega}_o^{(k)})$ $(k = 1, 2, \ldots, m)$ in the world space (drawn in red). Then, for each texel j, we transform $\boldsymbol{\omega}_i^{(k)}$ and $\boldsymbol{\omega}_o^{(k)}$ to $\hat{\boldsymbol{\omega}}_i^{(k)}$ and $\hat{\boldsymbol{\omega}}_o^{(k)}$ in the texel's local shading frame. Then, the training renderings effectively provide the BRDF values at each texel j with directions $\hat{\boldsymbol{\omega}}_i^{(k)}$ and $\hat{\boldsymbol{\omega}}_o^{(k)}$. Finally, we resample these BRDF values (using linear interpolation) at a fixed set of local viewing and lighting directions $(\hat{\boldsymbol{\omega}}_i^{(k)}, \hat{\boldsymbol{\omega}}_o^{(k)})$ (drawn in blue) to fill the feature vector.

$\hat{\boldsymbol{\omega}}_o^{(j,k)}$ associated with particular texel j. To estimate $f_r^{(j)}$ in Eq. 6, we linearly interpolate the measured data

$$\left\{(\hat{\boldsymbol{\omega}}_i^{(j,k)}, \hat{\boldsymbol{\omega}}_o^{(j,k)}, f_r^{(j,k)}) \mid k = 1, 2, \ldots, m\right\}.$$

Figure 5 shows a 2D example on this process. Lastly, we use w as a normalization factor by setting

$$w := \frac{N}{\sum_{j=1}^{N} \|\boldsymbol{f}_j\|_1}, \tag{7}$$

where $N := s_x \times s_y$ is the total number of texels.

With the feature vector $\boldsymbol{f}_j$ obtained for each texel j, we use K-means to cluster all texels into K groups and use the one closest to each cluster center as the representative. We will describe how K is determined in §5.3.3.

5.3.2 *Parameter Fitting*

With the texels clustered and representatives selected, we then perform a second batch of training renderings. This time, we only render the input model with small viewports corresponding to individual representative texels under a much higher angular sampling rate (see Figure 6-a). The outcome of this process is a set of M BRDF evaluations $(\hat{\boldsymbol{\omega}}_i^{(j,k)},\ \hat{\boldsymbol{\omega}}_o^{(j,k)},\ f_r^{(j,k)})$ for each representative texel j and $k = 1, 2, \ldots, M$. To seek for proper parameter values used by our BRDF model (1), denoted as $\boldsymbol{\theta}$, we minimize the L2 difference between measured and fitted BRDFs:

$$E(\boldsymbol{\theta}) := \sum_{k=1}^{M} \left[f_r(\hat{\boldsymbol{\omega}}_i^{(j,k)}, \hat{\boldsymbol{\omega}}_o^{(j,k)}) - f_r^{(j,k)}\right]^2, \tag{8}$$

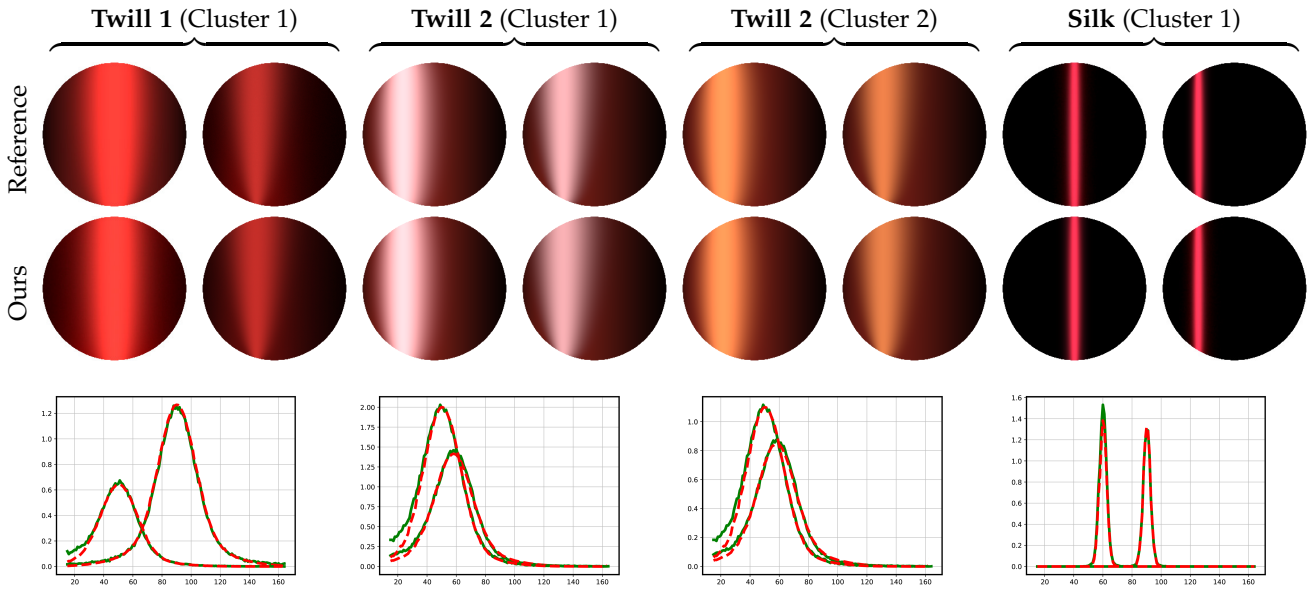


Fig. 6: BRDF fitting: (top) Reference BRDF slices obtained from our training renderings; (middle) Our fitted BRDF slices; (bottom) Luminance plots of the center rows of the BRDF slice images with the references in green and ours in red.

where $f_r(\hat{\boldsymbol{\omega}}_i^{(j,k)}, \hat{\boldsymbol{\omega}}_o^{(j,k)})$ is evaluated using Eq. (1) and $f_r^{(j,k)}$ comes from the training renderings. In practice, we solve this optimization using the Nelder-Mead simplex method [25]. Figure 6-b contains visualizations of our fitted BRDFs.

After finishing the fitting process for all representative texels, we simply assign each texel on the height map the BRDF from the corresponding representative.

5.3.3 Determining the Number of Clusters

Our clustering algorithm described in §5.3.1 requires the number of clusters K to be given as an input. In practice, however, the optimal K is generally unknown. If K is picked too low, the resulting accuracy will be limited. On the other hand, if K is too high, more training renderings have to be performed, and more parameters need to be fitted, stored, and evaluated, yielding lower performance. Our goal is to determine the proper value of K automatically and efficiently.

Our basic idea is to iteratively double K and perform our clustering (§5.3.1) and fitting (§5.3.2) steps until the approximation error becomes low enough. To define such error, we create validation renderings using our model and framework (described in §6) of the full input model using the same sparse set of m viewing and lighting configurations. If the difference between the validation renderings and previously created training ones is already below some pre-determined threshold or has stopped decreasing, the entire process terminates. Otherwise, we multiply K by two and re-perform the clustering (§5.3.1) and fitting (§5.3.2) steps. See Figure 10 in §7 to see how the choice of K affects final model accuracy.

Our full height map BRDF fitting pipeline is summarized in Algorithm 1.

Algorithm 1 Our reflectance profile fitting algorithm.

Generate m training renderings $I_1, I_2, \ldots, I_m$ using the full input model
Create a feature vector $\boldsymbol{f}_j$ for each texel j via Eq. (5)
$K \leftarrow K_{\min}$
while $K < K_{\max}$ **do**
  Apply k-means to group the texels into K clusters
  Generate M training renderings for each rep. texel and fit BRDF parameters by minimizing Eq. (8)
  Generate m validation renderings $I'_1, I'_2, \ldots, I'_m$ using our fitted model
  if $\sum_{t=1}^{m} \|I_t - I'_t\|_2^2 < \delta$ **then**
    break
  else
    $K \leftarrow 2K$
  end if
end while

5.3.4 Handling Flyaways

We use a similar approach to fit BRDFs for the flyaway fibers represented by the voxels in the thin volume V. We choose to fit BRDFs separately for the height map and the thin volume since it provides better numerical stability in our experiments.

We characterize these voxels based on their tangent directions as well as the diameter of the corresponding flyway fibers measured with the number of voxels. Then, we cluster the voxels and fit one BRDF to each representative using Algorithm 1. We generate training and validation renderings for fitting flyaway fiber parameters, we include the full model (i.e., height field and the thin volume) so that interreflection between a fabric's main surface and the flyaway fibers can be properly captured. To speed up the parameter fitting process, we fix β_1 and β_2 for the

flyaway fibers to be identical to the corresponding height map BRDF parameters. This is because the flyaway fibers are generally made of the same material as the regular ones, so they exhibit similar roughnesses. The other parameters such as the specular reflection coefficients k_1 and k_2, on the other hand, need to be fitted separately since they capture interreflections caused by a fabric's micro-geometry and can differ considerably between regular and flyaway fibers.

5.3.5 Fitting Multi-Scale Representation

Due to the high complexity of our hybrid models, rendering them from the far field can lead to severe aliasing without heavy super-sampling. We address this problem by generating multi-resolution representations of our model.

To this end, we prefilter our normal and tangent maps as well as the spatially varying BRDF parameters used in Eqs. 1 and 2. We leave the height map unchanged since (i) filtering a height map can change its intrinsic shadowing and masking structures which in turn affects its overall appearance; and (ii) we have already moved the high-frequency contents from the height map into the normal map (§5.2), and the height map itself is smooth and thus not prone to aliasing.

We jointly prefilter the normal and tangent maps as well as the BRDF parameters. For each pixel at some reduced resolution, we take all BRDF lobes from the original resolution and transform them into the world space using the corresponding normal map. We then fit a multi-lobe representation based on all the input lobes using an optimization that minimizes an error function similar to Eq. (8). The only difference is that, besides the BRDF parameters, we also fit a normal and tangent direction for each lobe. We limit the maximal number of lobes at each pixel to eight to balance performance and accuracy. Compared to the previous fitting steps, this step is much cheaper as it requires no training renderings.

Lastly, we prefilter our flyaway fiber volume in a similar manner by fitting multi-lobe BRDFs for each downsampled voxel. Since this volume is usually very sparse, we also compute the transparency (via the alpha channel) of each downsampled voxels. This value is given by the fraction of original voxels (inside the downsampled one) that are non-empty.

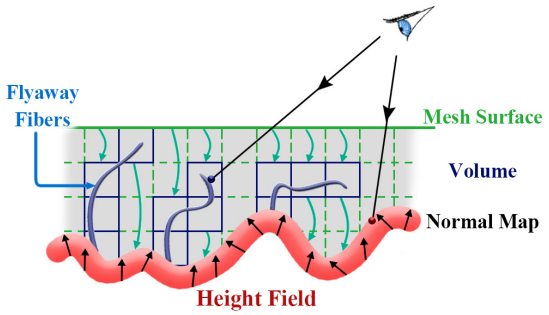


Fig. 7: Our realtime rendering pipeline: At render time, we use a polygonal mesh to represent the top surface of a fabric. When shading a pixel on the mesh, we apply GPU based ray marching over the thin volume V and underlying height field h jointly. When an intersection is found, we then calculate light reflected at that location using our BRDF (1).

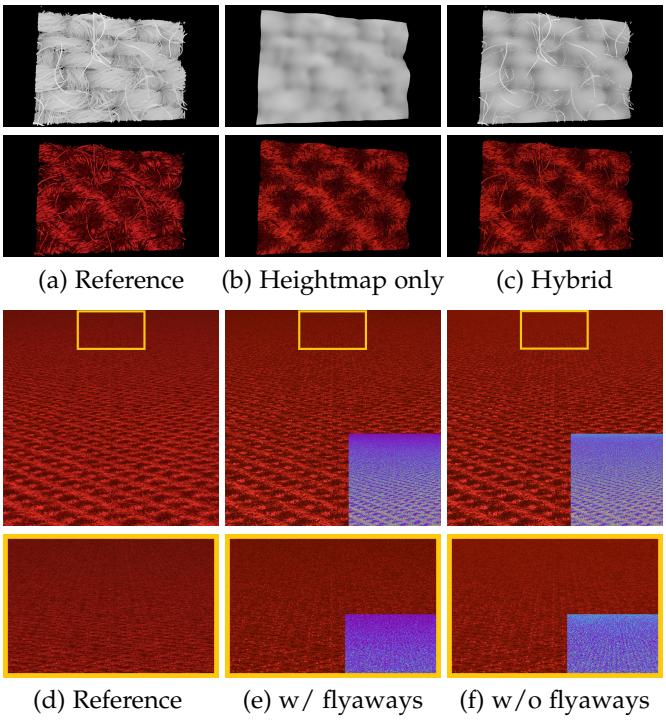


Fig. 8: The effectiveness of our hybrid representation: (a) Reference generated using the input volumetric model. (b) A purely surface-based representation has difficulties in capturing the flyaways fibers due to their strong 3D structures. (c) Our hybrid representation accurately captures the flyaway fibers. (d–f) Renderings of our models with and without flyaway fibers. The inclusion of flyaways yields notable changes in overall brightness of the fabric. The relative color maps in (e) and (f) use the same color scheme as Figure 10. Rendering the flyaway fibers introduces a 7% computational overhead.

6 Real-time Rendering of Our Models

We develop a new system capable of rendering our hybrid models (§4) at realtime. This system is also used to generate the validation renderings during BRDF parameter fitting (Line 7 of Algorithm 1).

As illustrated in Figure 7, our GPU-based system uses the top surface of each fabric as their geometric proxies. When shading a pixel on this surface, we perform GPU ray marching [26] over the thin volume V and the height map h jointly. This is achieved via a sparse data structure: for each texel (x, y) on the height field, we store a linked list representing the nonempty voxels above $h[x, y]$ in the thin volume V. Since V captures the flyaway fibers and is usually sparse, so is our linked list. With the linked lists available at each texel, we can then compute the intersection between the eye ray and the thin volume V or the height field h. When an intersection is found, we retrieve the local normal, tangent, and BRDF parameters accordingly, which are in turn used to shade the pixel via Eq. 1.

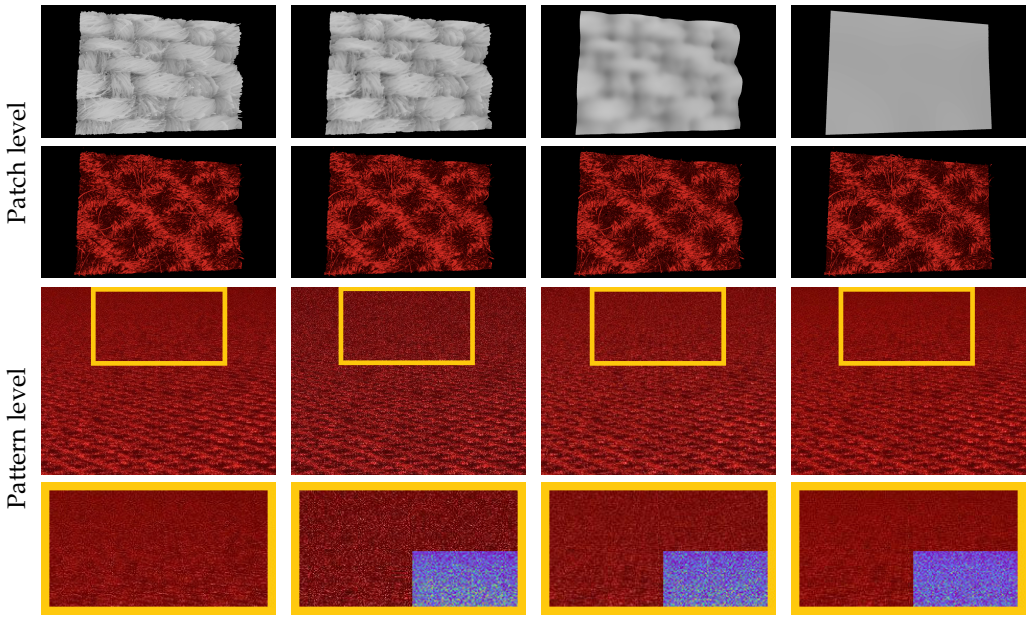


Fig. 9: We use low-pass filtering properly to balance result quality at near and far fields. Without prefiltering (b), the model suffers from severe aliasing when viewed from afar (without supersampling). The filtered models (c, d) produce much cleaner results. The bottom row contains zoomed-in versions of the row above.

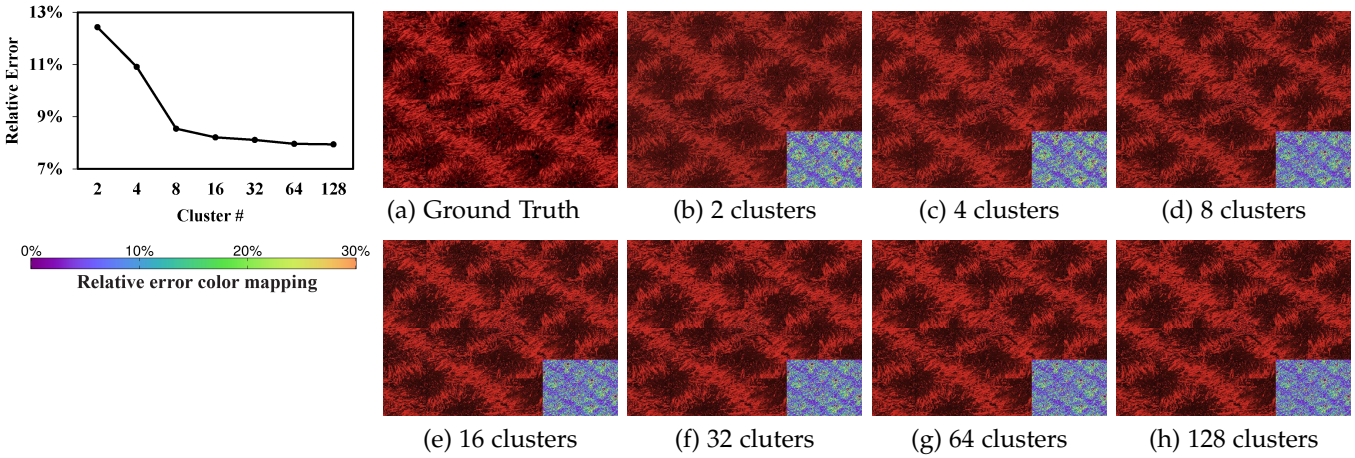


Fig. 10: Texel clustering and model fitting: Our algorithm iteratively doubles the number of clusters. This process terminates when the relative error between training and validation images drops below a user-specified threshold or stops to decrease.

Note that our realtime rendering system does not have to simulate inter-reflection as such information has already been baked in by our model fitting process (§5).

7 Results

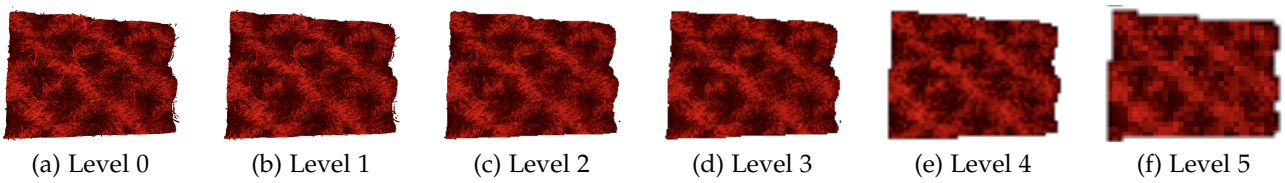
We now demonstrate the effectiveness of our technique via experimental results. In §7.1, we validate our main fitting algorithm by showing how it works in practice. Then, §7.2 contains rendered results generated with our models.

7.1 Validation

Hybrid representation.. We demonstrate the effectiveness of our hybrid representation in Figure 8. Capturing flyaway fibers is not only crucial for realistic close-up renderings (c) but also important for accurately resembling macro-scale shadowing effects of the original model (d–f).

Height map decomposition. Figure 9 demonstrates the effectiveness of our height map decomposition step. We choose the filter size properly so that the resulting model preserves the original height map's yarn-level geometry and shadowing while removing the fiber-level details to reduce aliasing.

Model fitting. We now show experimental results of our model fitting technique (Algorithm 1). Taking a volumetric model of a red twill fabric from prior work [22] as input, Figure 10-a contains a training rendering of this model.



(a) Level 0 (b) Level 1 (c) Level 2 (d) Level 3 (e) Level 4 (f) Level 5

Fig. 11: Multi-resolution representation: Prefiltered versions of our model (at reduced resolutions) accurately preserve the overall appearance of the original and enable efficient aliasing-free rendering at multiple scales.

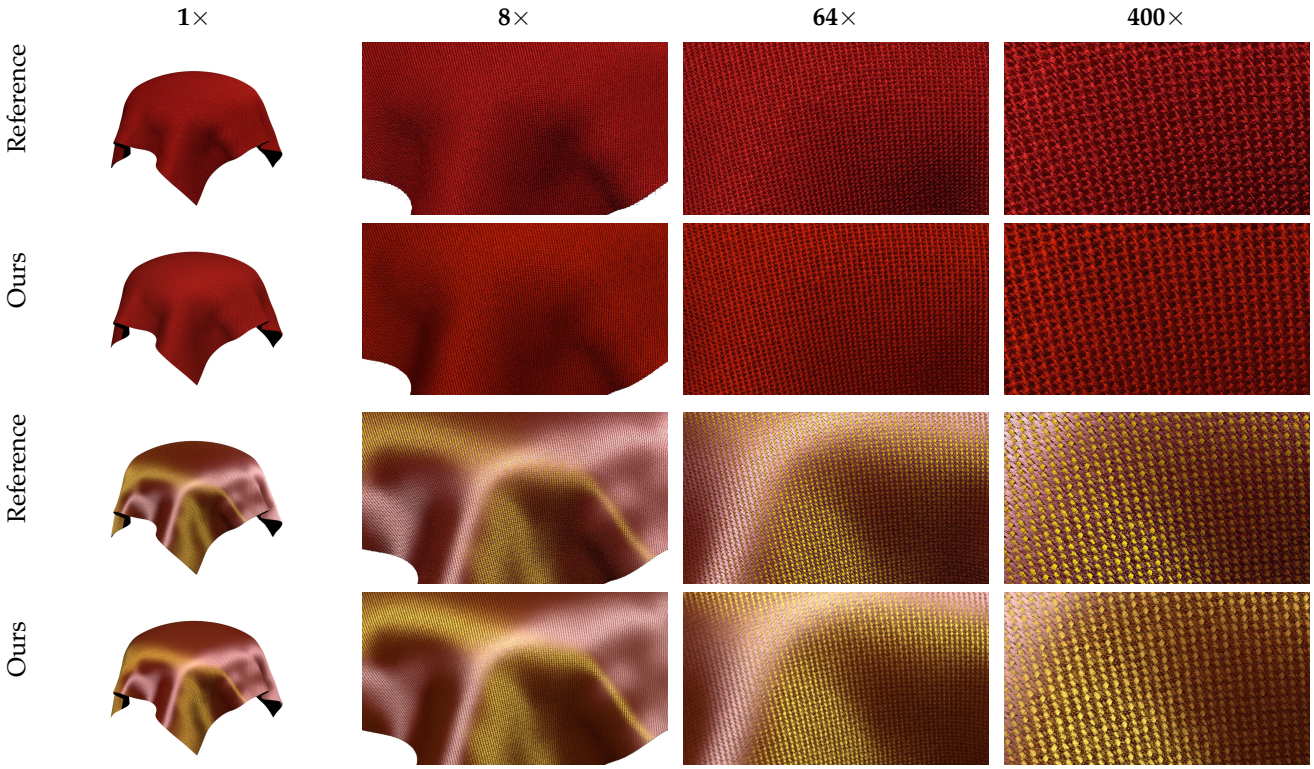


Fig. 12: Main results: Twill. Rendered images generated using the input volumetric models and our approach. Please see Table 1 for data size and render time statistics.

TABLE 1: Performance and render time information for all our results. Numbers in parentheses for training renderings indicate the amount of images rendered.

Model	Training 1 (for clustering)	Training 2 (for fitting)	K	BRDF fitting	Filter size (x / y)	Model size (input)	Model size (ours)	Render time (input)	Render time (ours)
Silk	16 h (324)	4 h (129 K)	8	3 h	18 / 37	1.9 GB	25 MB	72 min	24 ms
Twill 1	13 h (324)	2 h (65 K)	16	7 h	26 / 21	1.1 GB	31 MB	54 min	25 ms
Twill 2	23 h (648)	2 h (65 K)	16	7 h	26 / 21	0.5 GB	29 MB	52 min	27 ms
Satins	29 h (648)	3 h (65 K)	32	12 h	18 / 12	1.5 GB	41 MB	106 min	28 ms
Felt	21 h (648)	2 h (65 K)	32	12 h	28 / 33	1.8 GB	37 MB	94 min	31 ms

Since this model has two perpendicular set of yarns, we use two clusters (i.e., $K_{\min} = 2$) as the initial configuration. Unfortunately, using only two clusters yields high approximation error (over 10%), as one can see from the validation renderings in Figure 10-b. Thus, our algorithm iteratively doubles the number of clusters to 4, 8, and 16, leading to significantly better accuracy (Figure 10-cde). When further increase K to 32 and beyond, however, the improvement is quite limited (Figure 10-fgh). Our fitting algorithm then terminates with $K = 16$.

Multi-resolution representations. We demonstrate the effectiveness of our multi-resolution representations (§5.3.5) in Figure 11 where we show renderings generated using our original fitted model (level 0) and prefiltered ones with reduced resolution (level 1 upward). In addtition, we compare results filtered by different filter sizes in Figure 9. Our prefiltered models accurately capture the overall appearance of the original model at patch level and allow aliasing-free renderings at multiple resolutions at pattern level.

7.2 Main Results

We apply our model fitting technique (Algorithm 1) to four volumetric micro-appearance cloth models published by

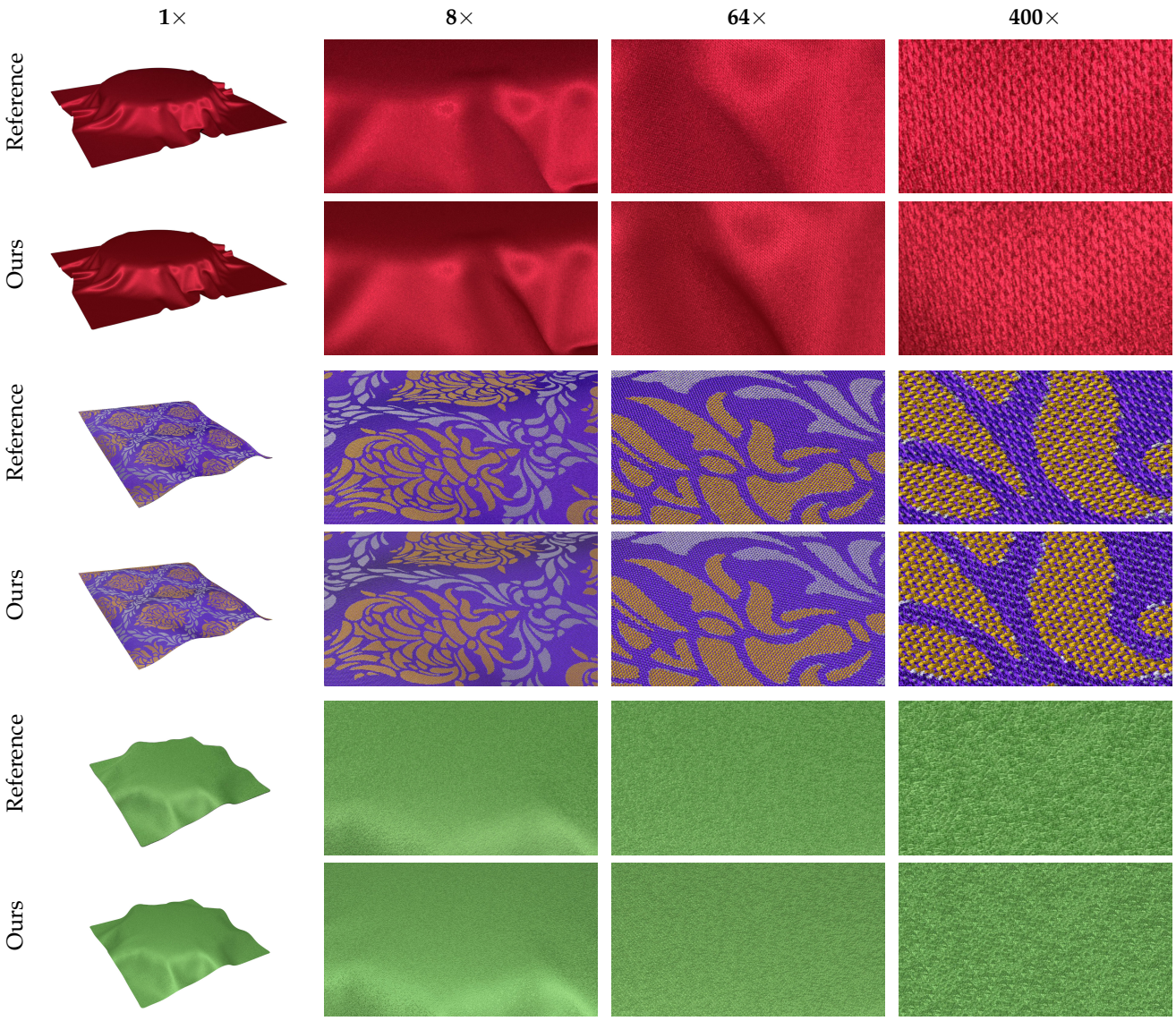


Fig. 13: Main results: Rendered images of silk charmeuse (top), satin (middle), and felt (bottom) generated using the input volumetric models and our approach. Please see Table 1 for data-size and render-time statistics.

previous work [22] (including the one used in Figure 6).

Figure 12 and 13 contains rendered images generated both from the original volumetric micro-appearance models and our fitted hybrid representations.

The first result in Figure 12 uses the same red twill material as in Figure 6. Our model successfully reproduces the diffuse yet strongly textured appearance of this material while reduces the rendering time from 54 minutes (using E5 2683V3 CPU) to 28 ms (with an nVIDIA GTX 1080 Ti GPU). The second result in Figure 12 uses a modified version of the first model with heterogeneous albedo and lower roughness, yielding distinctive colored highlights across its surface. Our fitted model accurately captures this anisotropic effect and renders at over 30 frames per second, compared with 52 minutes per frame when using the input model and volume path tracing.

Figure 13 shows rendered results with a variety of materials. In the first row, we show a piece of silk exhibiting very strong and anisotropic highlights. Some reference and fitted BRDF slices of this material have been visualized in Figure 6. Our model accurately captures this characteristic reflectance even under extremely close-up views while reducing the render time from 72 minutes to 26 ms. The second row shows a satin fabric with three types of yarn colors. Our results closely resemble the input appearances and offer a speedup of five orders of magnitude (from 106 minutes to 31 ms per frame). The last row of Figure 13 displays a piece of felt, a nonwoven fabric comprised of a thick layer of matted fibers. Due to this material's high hairiness, it has been very hard to describe using traditional models. Our hybrid model successfully captures the visual characteristics of this material while offering a speedup of 160000×.

Edited results. Our hybrid models are light-weight and can be easily edited. This is demonstrated in Figure 14. On the top row, we show two edited silk results with the left example having lower and the right one having higher roughnesses. In addition, we modified the tangent

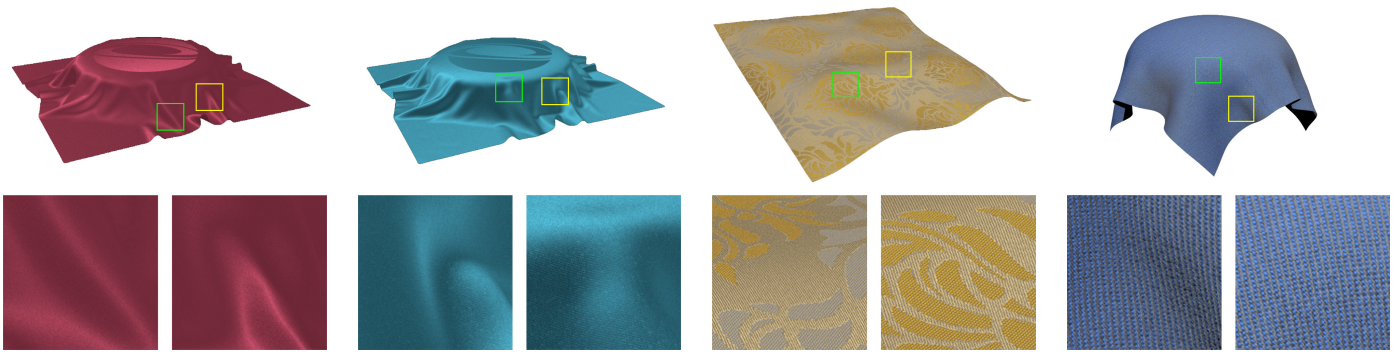
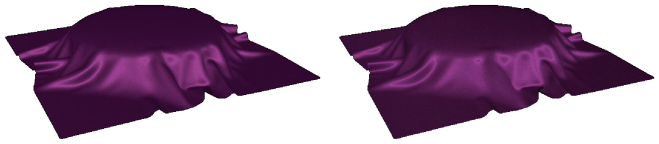


Fig. 14: Edited results: Our models can be easily edited and applied to different base geometries, greatly benefiting design and prototyping applications.



(a) Ours (edited) (b) Vol. (color-matched)

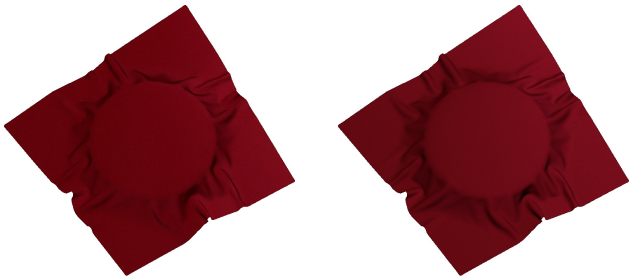
Fig. 15: Physical plausibility of our edited model: although our models are fitted with interreflection baked in, a simple editing of the reflectance albedo still yields plausible results.

Fig. 16: Extra result: Our method can be extended to handle environmental lighting.

directions on part of both samples, leading to logos formed with anisotropic reflections. On the bottom row, we show edited satin and twill fabrics with modified roughness and yarn colors. Please see the accompanying video for more material editing results.

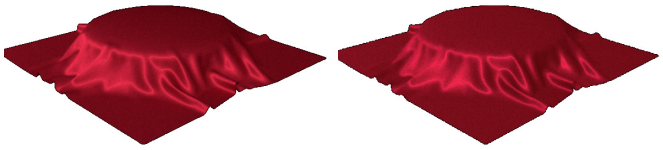
Since our fitting process effectively bakes shadowing and interreflection effects into the BRDF, linearly changing reflectance albedo is not fully accurate. Fortunately, as the overall appearance is usually dominated by direct illumination which is linear to the albedo, our edited results remain physically plausible. We demonstrate this in Figure 15. In this example, we edit the reflectance albedo of our model drastically (a) and color-match the volumetric version (b) by searching for single-scattering albedo that minimizes image differences. Despite small difference in contrast (caused by the aforementioned nonlinearity), our model is capable of reproducing the overall appearance of the volumetric one.

Extensions. Although we have focused on simple lighting (e.g., point and directional) and light reflectance, our technique can be extended to environmental lighting and light transmission. As shown in Figure 16, by fitting our BRDFs



Reference Ours

Fig. 17: Extra result: Our method can be extended to handle light transmission. In this example, we show renderings of a piece of cloth with a point light source located on the other side. Our model well captures the scattered and transmitted light from one side of cloth to the other side and closely matches the reference.



Reference Ours

Fig. 18: Limitation: the accuracy of our model regrades at grazing angles (see the top of this model as well as the silhouette edges).

to mixtures of spherical Gaussians [27], [28], our models can be rendered in realtime using prefiltered environment maps. In addition, our model can be extended to full BSDF by also fitting the transmission component. An example is shown in Figure 17 where a point light source is located on the other side of the fabric. Our model when fitted as a full BSDF well captures the scattered and transmitted light from one side of cloth to the other side. When handling BSDFs (as opposed to BRDFs), our fitting process becomes roughly two times slower (due to doubled training renderings) while the rendering performance remains identical.

8 Discussion and Conclusion

Limitations and future work. A main limitation of our technique is from the underlying geometric representation: relying on a height field to describe a fabric's main surface, our method cannot handle fabrics that are too fuzzy or too loose to have well-defined main surfaces. Also, as most surface-based representations, the physical accuracy of our model reduces at grazing angles (see Figure 18). Since our method relies heavily on texel clustering, input models with smoothly changing colors will require many clusters and thus be very expensive to fit to. Lastly, the micro-geometries of our models cannot be easily edited as they are extracted directly from voxelized input models.

In the future, more sophisticated clustering and fitting algorithms are worth exploring for better efficiency. Additionally, generalizing our light scattering model to support fibers with non-circular cross sections (e.g., based on the fiber scattering model introduced by Aliaga et al. [17]) is an interesting future topic.

Conclusion. Traditional surface-based cloth models are light-weight, easy to render, but lack sub-yarn-level details. Volumetric and fiber-based micro-appearance models, on the other hand, are extremely detailed but too heavy-weight and expensive to render. We introduce a hybrid micro-appearance cloth representation that enjoys the strength from both sides: our models are compact and easy to render and edit while offering rich details. Further, we develop a novel technique to fit our models to volumetric micro-appearance representations. The resulting models qualitatively match the input in appearance and can be rendered orders of magnitude faster. This unique combination of fidelity and performance offered by our technique can greatly benefit many applications in textile design and virtual prototyping.

Acknowledgments

We thank the anonymous reviewers for their constructive suggestions and comments. This work was supported in part by National Key R&D Program of China (No. 2017YFB1002605), NSFC (No. 61872319), Key R&D Program of Zhejiang Province (2018C01090), Zhejiang Provincial NSFC (No. LR18F020002) and NSF grant 1813553.

References

[1] P. Irawan and S. Marschner, "Specular reflection from woven cloth," *ACM Trans. Graph.*, vol. 31, no. 1, pp. 11:1–11:20, 2012.
[2] I. Sadeghi, O. Bisker, J. De Deken, and H. W. Jensen, "A practical microcylinder appearance model for cloth rendering," *ACM Trans. Graph.*, vol. 32, no. 2, pp. 14:1–14:12, 2013.
[3] S. Zhao, W. Jakob, S. Marschner, and K. Bala, "Building volumetric appearance models of fabric using micro CT imaging," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 44:1–44:10, 2011.
[4] P. Khungurn, D. Schroeder, S. Zhao, K. Bala, and S. Marschner, "Matching real fabrics with micro-appearance models," *ACM Trans. Graph.*, vol. 35, no. 1, pp. 1:1–1:26, 2015.
[5] S. Zhao, W. Jakob, S. Marschner, and K. Bala, "Structure-aware synthesis for predictive woven fabric appearance," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 75:1–75:10, 2012.
[6] K. Schröder, A. Zinke, and R. Klein, "Image-based reverse engineering and visual prototyping of woven cloth," *IEEE transactions on visualization and computer graphics*, vol. 21, no. 2, pp. 188–200, 2015.
[7] S. Zhao, F. Luan, and K. Bala, "Fitting procedural yarn models for realistic cloth rendering," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 51:1–51:11, 2016.
[8] S. Zhao, M. Hašan, R. Ramamoorthi, and K. Bala, "Modular flux transfer: Efficient rendering of high-resolution volumes with repeated structures," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 131:1–131:12, 2013.
[9] P. Khungurn, R. Wu, J. Noeckel, S. Marschner, and K. Bala, "Fast rendering of fabric micro-appearance models under directional and spherical gaussian lights," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 232:1–232:15, 2017.
[10] K. Schroder, R. Klein, and A. Zinke, "A volumetric approach to predictive rendering of fabrics," in *Computer Graphics Forum*, vol. 30, no. 4, 2011, pp. 1277–1286.
[11] G. Loubet and F. Neyret, "A new microflake model with microscopic self-shadowing for accurate volume downsampling," in *Computer Graphics Forum*, vol. 37, no. 2, 2018, pp. 111–121.
[12] K. Wu and C. Yuksel, "Real-time cloth rendering with fiber-level detail," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 2, pp. 1297–1308, 2017.
[13] P.-P. Sloan, J. Kautz, and J. Snyder, "Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 527–536, 2002.
[14] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM Trans. Graph.*, vol. 18, no. 1, pp. 1–34, 1999.
[15] S. R. Marschner, H. W. Jensen, M. Cammarano, S. Worley, and P. Hanrahan, "Light scattering from human hair fibers," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 780–791, 2003.
[16] A. Zinke and A. Weber, "Light scattering from filaments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 2, pp. 342–356, 2007.
[17] C. Aliaga, C. Castillo, D. Gutierrez, M. A. Otaduy, J. Lopez-Moreno, and A. Jarabo, "An appearance model for textile fibers," in *Computer Graphics Forum*, vol. 36, no. 4, 2017, pp. 35–45.
[18] G. C. Guarnera, P. Hall, A. Chesnais, and M. Glencross, "Woven fabric model creation from a single image," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 5, p. 165, 2017.
[19] W. Jakob, A. Arbree, J. T. Moon, K. Bala, and S. Marschner, "A radiative transfer framework for rendering materials with anisotropic structure," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 53:1–53:13, 2010.
[20] S. Chandrasekhar, *Radiative transfer*. Courier Corporation, 1960.
[21] E. Heitz, J. Dupuy, C. Crassin, and C. Dachsbacher, "The sggx microflake distribution," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 48:1–48:11, 2015.
[22] S. Zhao, L. Wu, F. Durand, and R. Ramamoorthi, "Downsampling scattering parameters for rendering anisotropic media," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 166:1–166:11, 2016.
[23] F. Luan, S. Zhao, and K. Bala, "Fiber-level on-the-fly procedural textiles," *Comput. Graph. Forum*, vol. 36, no. 4, pp. 123–135, 2017.
[24] T. J. Kang, C. H. Kim, and K. W. Oh, "Automatic recognition of fabric weave patterns by digital image analysis," *Textile Research Journal*, vol. 69, no. 2, pp. 77–83, 1999.
[25] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the nelder–mead simplex method in low dimensions," *SIAM Journal on optimization*, vol. 9, no. 1, pp. 112–147, 1998.
[26] N. Tatarchuk, "Practical parallax occlusion mapping with approximate soft shadows for detailed surface rendering," in *ACM SIGGRAPH 2006 Courses*, 2006, pp. 81–112.
[27] J. Wang, P. Ren, M. Gong, J. Snyder, and B. Guo, "All-frequency rendering of dynamic, spatially-varying reflectance," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 133:1–133:10, 2009.
[28] K. Xu, W.-L. Sun, Z. Dong, D.-Y. Zhao, R.-D. Wu, and S.-M. Hu, "Anisotropic spherical gaussians," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 209:1–209:11, 2013.