Practical Level-of-Detail Aggregation of Fur Appearance

JUNQIU ZHU, Shandong University, China
SIZHE ZHAO, Shandong University, China
LU WANG, Shandong University, China
YANNING XU, Shandong University, China
LING-QI YAN, University of California, Santa Barbara, USA

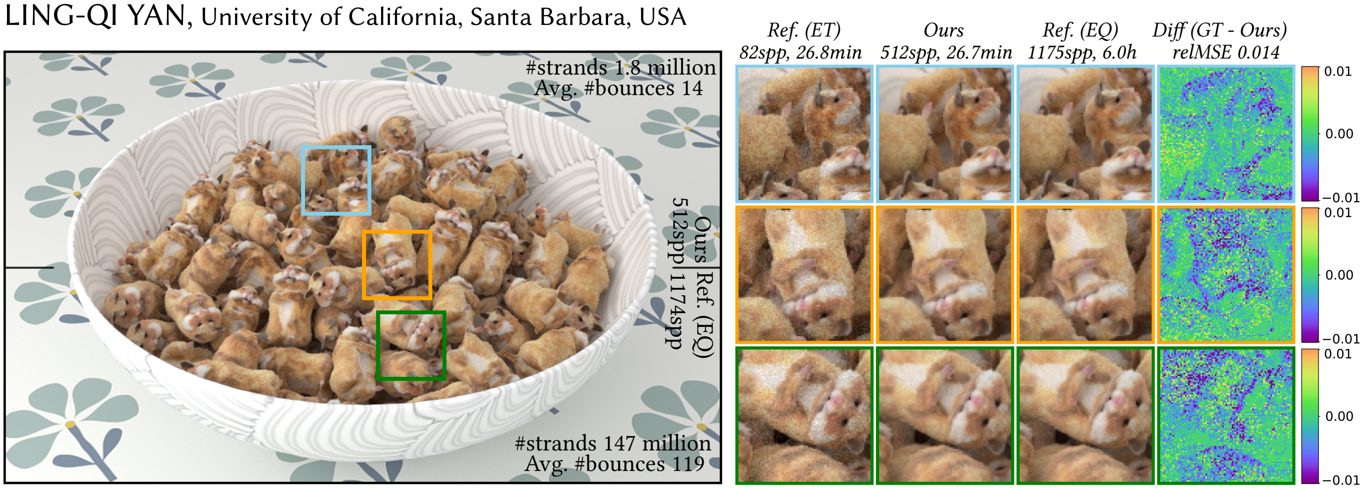


Fig. 1. A bowl of hamsters rendered using our aggregated fur appearance model. This scene originally contains about 147 million strands of fur fibers, but we only use 1.8 million strands of aggregated fur fibers. We use 512 samples per pixel (spp), and compare with the reference (Ref.) [Yan et al. 2017a] at equal time (ET) and equal quality (EQ). Our method not only converges 13.5× faster than the reference at EQ (because each light path in our approach introduces a smaller variance), but also traces more spp at ET (because the average number of bounces along a light path in Ref. is 8× more than ours). On the right, we show the difference images between GT (converged Ref. w/ 16384 spp) and our result (still 512 spp). Note the extremely small scale marked on the color bar.

Fur appearance rendering is crucial for the realism of computer generated imagery, but is also a challenge in computer graphics for many years. Much effort has been made to accurately simulate the multiple-scattered light transport among fur fibers, but the computation cost is still very high, since the number of fur fibers is usually extremely large. In this paper, we aim at reducing the number of fur fibers while preserving realistic fur appearance. We present an aggregated fur appearance model, using one thick cylinder to accurately describe the aggregated optical behavior of a bunch of fur fibers, including the multiple scattering of light among them. Then, to acquire the parameters of our aggregated model, we use a lightweight neural network to map individual fur fiber's optical properties to those in our aggregated model. Finally, we come up with a practical heuristic that guides the simplification process of fur dynamically at different bounces of the light, leading to a practical level-of-detail rendering scheme. Our method achieves nearly the same results as the ground truth, but performs 3.8×-13.5× faster.

CCS Concepts: • **Computing methodologies → Rendering**.

Additional Key Words and Phrases: physically based rendering, fur, level-of-detail

1 INTRODUCTION

Computer generated animals are pervasively seen in all kinds of art forms from animations and movies generated by offline rendering techniques to real-time applications such as video games and virtual/augmented reality. The fuzziness and the saturated appearances of animal furs greatly enhance the realism of animal characters as well as the entire artistic work.

However, rendering the appearance of fur is never an easy task. The main reason, as one might immediately think of, is that the number of individual fur fibers is enormous, easily passing the order of millions. The huge number of fur fibers imposes heavy computation cost on ray-scene intersections, shading, as well as the entire rendering process. Moreover, in order to render the realistic appearance of an entire fur pelt, a renderer is supposed to simulate multiple (usually from 30 to 100) bounces of light among fur fibers. What is worse, in modern computer generated imagery (CGI), it is common to see hundreds of animals in one scene, making it impractical to simply leave the complex light transport problem to, e.g., brute force path tracing.

Much effort has been devoted to accelerating animal fur rendering. Some previous work focuses on simplifying the complex multiple

Authors' addresses: Junqiu Zhu, Shandong University, Jinan, Shandong, China, 250100, junqiuzhu27@gmail.com; Sizhe Zhao, Shandong University, Jinan, Shandong, China, 250100, jeff23101719@gmail.com; Lu Wang, Shandong University, Jinan, Shandong, China, 250100, luwang_hcivr@sdu.edu.cn; Yanning Xu, Shandong University, Jinan, Shandong, China, 250100, xyn@sdu.edu.cn; Ling-Qi Yan, University of California, Santa Barbara, Santa Barbara, CA, USA, 93106, lingqi@cs.ucsb.edu.

0730-0301/2022/7-ART1
https://doi.org/10.1145/3528223.3530105

scattering of light within the hair/fur volume. For example, Zinke et al. [2008] proposed the technique of dual scattering to approximate multiple scattering without tracing rays. Some other work turns the problem of hair/fur rendering into a relatively better-studied problem of participating media rendering. For example, Moon et al. [2008] convert human hair geometry to a volume of participating media to approximate low-frequency multiple scattering. Yan et al. [2017a] use subsurface scattering, specifically designed for animal fur, to simplify and reduce the appearance of fur from multiple bounces of light to a simple BSSRDF (Bidirectional Surface Scattering Reflectance Distribution Function) model.

Despite the success of these methods, we notice that approximating light transport is still not the most straightforward way to speed up fur appearance rendering. Instead, sharply reducing the number of fur fibers in the sense of geometry is more intuitive and potentially more efficient for the purpose of acceleration. One naïve but pervasively used method by the industry is to reduce the number of fur fibers while making each of the remaining fiber a thicker cylinder. However, this approach results in overly hard/bold/dry appearances with unrealistically biased brightness/hue that fails to match the original. Therefore, we ask the question: can we do better than the naïve method?

In this paper, we analyze the possibility of reducing the number of fur fibers without visually noticeable compromise to the rendering quality. We seek a solution that is both accurate and practical. The key idea of our method is to use one thick fiber to represent the *aggregated appearance* of a bunch of fur fibers, including the multiple bounces of light among these fibers. And we show that the aggregated appearance can be well captured using a single fur fiber's optical properties with some extensions. For example, we assign our aggregated fur fiber different optical properties according to different incident directions. Then we refer to a data-driven approach, using a neural network, lightweight enough to be implemented inline in shaders, to evaluate our model practically. We further come up with a level-of-detail scheme that dynamically simplifies the fiber geometry based on the viewing distance and different number of bounces along one light path, achieving reliable appearance aggregation together with controllable geometry simplification.

We demonstrate that our results are almost indistinguishable from the ground truth, but are 3.8×-13.5× faster. Moreover, since our method aims at accelerating fur rendering in the level of geometry and appearance, other methods that approximate light transport as mentioned earlier, can still be applied in addition to our method. Therefore, our work can benefit a variety of applications, from those who can only afford a small amount of fur fibers, possibly due to the constraint on computational power, to those requiring high precision but still needing better performance. Therefore, we believe that our method has made an important contribution to the long-standing research problem of geometry/appearance prefiltering.

2 RELATED WORK

Hair/fur models for single fiber. Hair and fur fibers share similar types of structure form outside to inside: cuticle that reflects light, cortex that absorbs light, and medulla that scatters light. The medullas in human hair fibers are usually small, therefore, Marschner et al. [2003] proposed the initial physically-based human hair reflectance model, approximating hair fibers as rough dielectric cylinders and presenting a longitudinal-azimuthal decomposed parametric appearance model (Fig. 2 (a)). Their model has three types of reflectance: R, TT and TRT, where R and T stand for reflection and transmission, respectively. The Marschner model assumes that the azimuthal sections (Fig. 2 (b)) of hair fibers are perfectly smooth, which is not strictly physically correct. Therefore, d'Eon et al. [2011] extended the Marschner model to account for azimuthal roughness. However, their computational cost is significant, because the evaluation containing azimuthal roughness relies on Gaussian quadrature and Taylor expansion. Chiang et al. [2016] adopted a near-field formulation by considering accurate incident positions azimuthally. Their model uses an exaggerated azimuthal roughness to mimic the scattering effects from the inner structure of hair/fur fibers, which is not physically based, but achieves good visual effects. Khungurn and Marschner [2017] focused explicitly on elliptical hair fibers, revealing the different properties compared to those with circular azimuthal sections. Xia et al. [2020] proposed a hair reflectance model based on the wave optics theory. This paper focuses on the fur reflectance under the geometric optics framework.

Compared to human hair, a fur fiber usually has a non-negligible medulla volume in the center part, which scatters transmitting light. Kayjiya and Kay [1989] introduced an empirical fur shading model with a diffuse lobe and a specular lobe, similar to the Phong reflectance model. Yan et al. [2015] proposed a physically accurate fur model, known as the double cylinder model, in which the cuticle, the cortex and the medulla are all involved. Yan et al. [2017a] further simplified the light scattering types, resulting in a model that only appends two additional lobes TTs and TRTs based on the Marschner hair reflectance model (Fig. 2 (c)). In this paper, we use the model by Yan et al. [2017a] as a base model that accounts single fur fiber appearance and propose an extended model to account for a bunch of aggregated fur fibers.

Hair/fur multiple scattering methods. Accurate simulation of multiple scattering is very costly because it requires tracing light bouncing between hair fibers. Unlike the case when the light bounces between surfaces, where $3 \sim 5$ bounces is usually good enough, the number of light bouncing between hair/fur fibers can easily reach $30 \sim 100$, which is very costly to simulate. Some methods accelerate multiple scattering by using better importance sampling. Hery et al. [2012] and d'Eon et al. [2013] proposed different importance sampling schemes for hair fibers to accelerate the convergence of path tracing global illumination. These methods made it possible to ray-trace the hair geometry, thus are widely used. We adopt the importance sampling scheme by d'Eon et al. [2013].

A line of work focuses on the similarity between the multiple scattering between hair fibers and in participating media. Moon et al. [2006] adopted the photon mapping method to distribute photons within the hair volume as light bounces inside. But the photon mapping method is still inefficient, especially when the importance sampling approaches emerge. Moon et al. [2008] further provided a volumetric representation to replace the actual hair fiber geometry, and precomputed a 3D grid of spherical harmonic coefficients that stores the directional distribution of scattered radiance in the hair volume. However, those volumetric approaches cannot capture fine details, thus are usually used in combination–hair fibers for primary hits, and volumes for further bounces–which impose even heavier storage and memory consumption. Moreover, since these methods were designed for human hair, they cannot easily match fur appearance, as will be shown in Sec. 6. To our knowledge, the work by Yan et al. [2017a] is the only work that approximates the multiple scattering of light between animal fur fibers. In that work, a conversion scheme was proposed to switch the problem of rendering multiple scattering from fur fibers to rendering Bidirectional Surface Scattering Reflectance Distribution Functions (BSSRDFs). This method does not need extra storage, but it requires a preprocess stage (5-7 minutes for each scene) to each static pose/model, and usually introduces bias/overblur in the rendering result.

The dual scattering approximation [Zinke et al. 2008] is a widely-used method that fakes multiple scattering in the hair volume. It assumes that the scattering events always happen along the main path–the light hits a hair fiber and penetrates through the hair volume in a straight line. Globally, the light reaches the shading point by going through the hair volume. And locally, the light scatters forward and back, assuming all hair fibers are the same near the shading point. The dual scattering method is successfully used in real-time rendering. However, the bold simplifications (main path and local similarity) make it difficult for the results to match the ground truth. Moreover, the results will be inconsistently brighter or darker, since the brightness is determined by a couple of empirical parameters up to artists.

Appearance prefiltering. Appearance prefiltering includes surface-based prefiltering and volume-based prefiltering. Surface-based approaches such as LEAN [Olano and Baker 2010], LEADR [Dupuy et al. 2013], Han et al. [2007] and Wu et al. [2019] simplify the complex surface details in geometry, and adjust the resulting BRDFs to keep the overall appearance unchanged. Yan et al. [2014; 2016] rendered high-frequency materials by accurately prefiltering the microfacets' distribution as a sum of contribution from 4D Gaussian primitives.

Zhao et al. [2016] proposed a volumetric micro-appearance prefilering method. They prefilter phase functions to accurately downsample heterogeneous and anisotropic media. Vicini et al. [2021] presented an empirical method that adds surface-like correlation to the volume representation. However, for hair and fur, as well as general fibers, the difference between geometric and volumetric representations are still obvious [Khungurn et al. 2015]. It is not yet clear how to convert hair/fur fibers into volumes, while accurately accounting for their orientations, lengths and the correlation between different fibers.

Prefiltering also happens among sub-components of the entire appearance. Granular materials are composed of numerous dielectric grains, and their aggregate behaviour determines the overall appearance. In order to accelerate rendering, a line of research [Lee and O'Sullivan 2007; Meng et al. 2015; Moon et al. 2007; Müller et al. 2016] use precomputed teleport functions or BSSRDFs to "jump over" blocks of grains approximately. Nevertheless, we cannot directly use those methods to speed up fur appearance because fur fibers are cylinders and have more complex appearance.

Neural aided rendering. Neural networks have been successfully introduced to solve the complex and high-dimensional problems in the field of physically-based rendering. Since rendering is sensitive to performance, we mainly introduce those designed to facilitate the rendering process, bypassing those requiring large-scale deep networks.

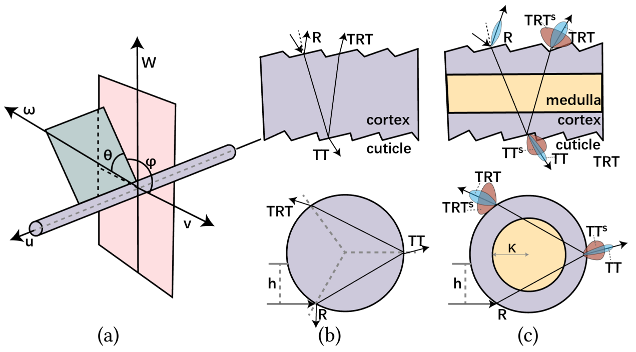


Fig. 2. (a) Longitudinal-azimuthal parameterization for hair/fur fibers. Any direction ω can be parameterized into θ in the plane spanned by ω and the cylinder axis u, and ϕ orthogonal to the plane. (b) Longitudinal and azimuthal lobes of the Marschner hair model. (c) Lobes of the fur model by Yan et al. [2017a]

Early work used small multilayer perceptrons (MLPs) to estimate global illumination [Ren et al. 2015, 2013] and to denoise the Monte Carlo path tracing [Kalantari et al. 2015]. Lightweight networks have also been applied as compressed representations by overfitting to specific assets that can be efficiently stored and evaluated [Davies et al. 2020; Zhu et al. 2021]. Yan et al. [2017b] proposed a lightweight MLP to convert fur fiber properties to participating media scattering properties. We also use a lightweight MLP to provide the actual parameters that describe an aggregated fiber's optical properties.

Industrial approaches. In the industry, Unreal Engine 4 (UE4) [Epic Games 2019] and Frostbite Engine [EA DICE 2006] focus on real-time rendering model for hair, and use the same model for fur. They both use a simplified version of the Marschner model [Marschner et al. 2003] for single scattering. UE4 uses an ad-hoc model for multiple scattering, which generates fake normal for the hair volume and UE4 approximates the multiple scattering with a Lambertian model. The Frostbite Engine uses simplified dual scattering [Zinke et al. 2008] to calculate the multiple scattering. Neither method is physically-based on multiple scattering, and neither can afford a large amount of hair/fur fibers (note specifically that the Nanite technique in UE5 can only simplify triangular geometry).

3 BACKGROUND AND MOTIVATION

In this section, we first briefly recap some background knowledge of hair/fur reflectance models for individual fibers. Then we provide observations on facts and existing methods that motivate our approach.

Hair/fur fibers look like cylinders from outside, but have complex internal structures. As Fig. 2 illustrates, from outside to inside, a hair/fur fiber has two common types of layers: cuticle that is covered with tilted scales, cortex that contains pigments thus absorbing light, and the medulla that scatters light like participating media.

Table 1. The parameters used to describe our aggregated model. **Input**: the first 10 parameters are adopted from Yan et al. [2017a], defining a single fur fiber's properties. ρ, $\mathcal{N}$ and θ_i are the parameters specifying the aggregation statistics. **Output**: additional parameters that we use to evaluate our aggregated BCSDF model.

Parameter	Definition
Input	
κ	medulla index (rel. radius length)
η	refractive index of cortex and medulla
α	scale tilt for cuticle
β_m	longitudinal roughness of cuticle (stdev.)
β_n	azimuthal roughness of cuticle (stdev.)
$\sigma_{c,a}$	absorption coefficient in cortex
$\sigma_{m,a}$	absorption coefficient in medulla
$\sigma_{m,s}$	scattering coefficient in medulla
g	anisotropy factor of scattering in medulla
l	layers of cuticle
ρ	(positionally-varying) density of aggregated fiber
$\mathcal{N}$	number of single fibers in an aggregated fiber
θ_i	longitudinal incident angle
Output	
μ	the scaling parameter
θ_t	center of the longitudinal top lobe
θ_m	center of the longitudinal middle lobe
θ_b	center of the longitudinal bottom lobe
β_t	roughness of the longitudinal top lobe
β_m	roughness of the longitudinal middle lobe
β_b	roughness of the longitudinal bottom lobe
A_t	attenuation term of the azimuthal top lobe
A_m	attenuation term of the azimuthal middle lobe
A_b	attenuation term of the azimuthal bottom lobe
$y_{0\ldots4}$	the interpolating spline's control points

3.1 Hair/fur appearance model

Researchers model the hair/fur fibers as cylinders and use the BCSDF (Bidirectional Curve Scattering Distribution Function) [Zinke and Weber 2007] to model how hair/fur fibers scatter light:

$$L_r(\omega_r) = \int L_i(\omega_i)\, S(\omega_i, \omega_r) \cos\theta_i \,\mathrm{d}\omega_i, \tag{1}$$

where S is the BCSDF, L_i and L_r are the incoming radiance from direction ω_i, and the outgoing radiance to direction ω_r, respectively.

Hair BCSDF model. For simplicity, the BCSDF model is generally analyzed using a longitudinal-azimuthal (θ, ϕ) parameterization. As shown in Fig. 2 (a), Marschner et al. [2003] modeled the hair BCSDF as a product of M and N profiles which represent the longitudinal and azimuthal light-cylinder interactions separately:

$$L_r(\theta_r, \phi_r) = \int_{-\pi}^{\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} L_i(\theta_i, \phi_i)\, S(\theta_i, \theta_r, \phi_i, \phi_r) \cos^2\theta_i \,\mathrm{d}\theta_i \,\mathrm{d}\phi_i \tag{2}$$

where the single cosine term becomes squared because the solid angle $\mathrm{d}\omega_i = \cos\theta_i \,\mathrm{d}\theta_i \,\mathrm{d}\phi_i$ in this parameterization. And the BCSDF S can be further specified as

$$\begin{aligned} S(\theta_i, \theta_r, \phi_i, \phi_r) &= \sum_p S_p(\theta_i, \theta_r, \phi_i, \phi_r) / \cos^2\theta_d \\ &= \sum_p M_p(\theta_i, \theta_r) \cdot N_p(\phi; \eta') / \cos^2\theta_d, \end{aligned} \tag{3}$$

where $\theta_h = (\theta_r + \theta_i)/2$ is the longitudinal half angle, $\theta_d = (\theta_r - \theta_i)/2$ is the longitudinal difference angle, and $\phi = \phi_r - \phi_i$ is the relative azimuthal angle. The η' can be written as $\eta' = \sqrt{\eta^2 - \sin^2\theta_d}/\cos\theta_d$ which is the cortex's virtual index of refraction, accounting for inclined longitudinal incident directions. As shown in Fig. 2 (b), this model takes three types of light paths/lobes $p \in R, TT, TRT$ into consideration, where R stands for reflection and T for transmission.

Fur BCSDF model. Yan et al. [2015] proposed a double cylinder model for fur fibers. The outer cylinder represents the cuticle and the inner cylinder represents the medulla, which scatters light. And between these two cylinders is the cortex, which simply absorbs light. Yan et al. [2017a] summarize the double cylinder model into five lobes, including the classic unscattered R, TT and TRT lobes from the Marschner model and two additional scattered lobes TT^s and TRT^s to describe the scatter properties when light passes through the medulla. Fig.2 (c) illustrates the structure of the double cylinder model and the five lobes. Fur BCSDF model can still be defined as Eqn. 3 with additional TT^s and TRT^s paths in p.

Unscattered lobes: R, TT and TRT are unscattered lobes formed by light paths that do not go through the medulla or are not scattered through the medulla. The longitudinal lobes M_p are normalized, while the azimuthal lobes are further separated as $N_p = A_p \cdot D_p$, where A_p is the attenuation term specifying the amount of energy loss, and D_p is the distribution term that describes how the attenuated energy distributes, which is also normalized. The unscattered lobes are generalization and extension of the those from the hair BCSDF model.

Scattered lobes (TT^s, TRT^s): The scattered lobes TT^s and TRT^s are formed by the light paths going through the medulla and being scattered. Both scattered lobes are still parameterized longitudinally and azimuthally. Since the scattering behavior can be complex, Yan et al. [2017a] precomputed the scattering profiles of the medulla for different parameters. Then an analytical representation of the longitudinal distribution as well as the azimuthal distribution and attentuation can be presented similar to the unscattered lobes.

Near field and far field models: Near field scattering specifies the offset h azimuthally as the incoming position (Fig. 2 (c)). For far field approximation, parallel light is assumed, covering a fiber's width. Thus, far field approximation yields the azimuthal scattering function N_p by integrating over all possible offsets h:

$$N_p(\phi; \eta') = \frac{1}{2} \int_{-1}^{1} N_p(h, \phi; \eta')\, dh. \tag{4}$$

The visual difference between near field and far field models is significant when viewed from closeup: near field models render a hair/fur fiber like a cylinder, while far field models generate flat and ribbon-like appearance, due to ignoring of different azimuthal offset h. Nevertheless, when viewed from sufficiently far away, i.e. when a hair/fur fiber is narrower than a pixel, near field and far field models produce exactly the same results. However, far field models in this case will be far more efficient to render, since near field models are essentially leaving the integration numerically to the renderer.

Since we focus on the reduction of the number of fur fibers, it directly implies that we should always use far field models.

Table 2. Statistics of various types of animal fur including density (number of fur fibers/cm^2), average skin area (cm^2), estimated number of fibers in total, and radius for a single fur fiber. We mark the maximum of each property as red and the minimum as blue.

Species	**fibers/cm^2**	**skin area (cm^2)**	**Est. #fibers**	**radius (μm)**
Gorilla	48	98283	4.4M	80
Badger	320	16875	5.4M	*N/A*
Monkey	501	10443	5.23M	60
Vole	3000	1150	3.4M	5
Antelope	780	177504	138.4M	45
Kangaroo	1960	76514	149.9M	*N/A*
Mink	6387.5	7120	45.4M	65
Hamster	9519	200	1.90M	15
Rabbit	5699.5	3725	21.2M	35
Cat	3572.3	10720	38.2M	40
Dog	1640	4866	8.0M	45
Fox	3780	32830	124.1M	35
Squirrel	10425	2954	30.8M	*N/A*
Yak	1616	111494	180.17M	20
Horse	6351	96480	608.2M	*N/A*

3.2 Motivation

Number of fur fibers. When we talk about fur, one immediate property to consider is the huge number of fur fibers. Sandel [2013] collected the statistics of fur fibers from a wide range of animal species, listed in Tab. 2. As we can immediately observe, the total number of fur fibers on an animal can already be extremely large even for offline applications, let alone it is quite likely that multiple animals can appear together in one viewport. Therefore, several problems can arise at once.

- Performance. In offline applications such as animations, the rendering process requires tracing hundreds of bounces along each light path through the fur volume. And the number of bounces scales up with the number of fibers. Classic trade-offs, such as Russian Roulette, will introduce significant noise to the rendering result. Meanwhile, rasterizing this amount of fibers is nearly impossible for real-time applications.
- Memory consumption. The fibers have to be fully stored in video memory for real-time applications. And an acceleration structure for the fibers, such as a Bounding Volume Hierarchy (BVH), must be built in main memory in offline applications. In either case, the memory consumption scales at least linearly with the number of fur fibers, which can easily reach the level of dozens of gigabytes. Moreover, a complex BVH also slows down the performance significantly. In fact, when we render the hamster model (Fig. 3) with only one million fur fibers, the BVH traversal time already takes up about 10% of the total rendering time.
- Energy conservation. Any undesired energy gain or loss (e.g. ±1% due to improper numerical cut-offs or other issues in the implementation) at each bounce will be accumulated and magnified as illustrated in Fig. 4. This is a severe problem that did not draw enough attention from the academia, but pervasively encountered in the industry. More fibers leads to more bounces, thus much larger possibility that the results will not be energy conserving, even if they are guaranteed/designed to be energy conserving theoretically.

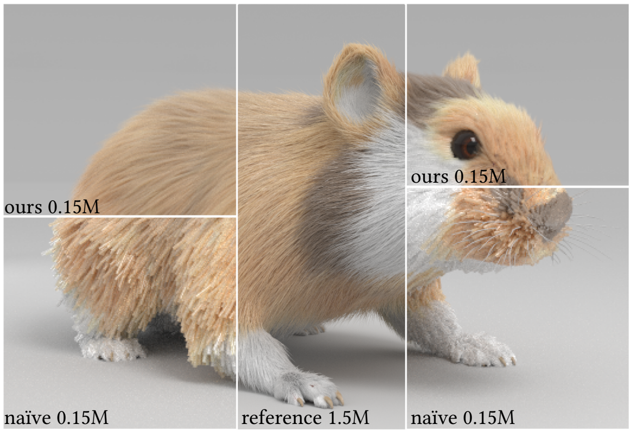


Fig. 3. We reduce the number of the hamster's fur fibers to 10%. Then we render it with our aggregated BCSDF model, and compare with the naïve approach that only simplifies but does not aggregate. The naïve method results in a harder and brighter appearance. Note specifically that since this is an extremely close-up view, using 10% fur fibers is a drastic over-simplification according to our proposed heuristics (Sec. 4.3). However, even with such small amount of fur fibers, our method is still able to plausibly recover the original appearance with minor detail loss.

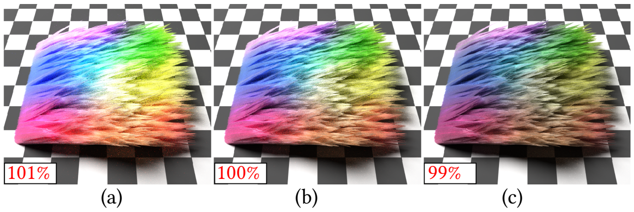


Fig. 4. The *Pelt* scene rendered using the fur BCSDF model [Yan et al. 2017a] with manually scaled total energy of each fur fiber to (a) 101%, (b) 100% and (c) 99%. After multiple scattering, (a) is 30% brighter than (b), and (c) is 20% dimmer than (b). The average bounces of a light path in this scene is only about 32. In a real scene with a lot more fibers, slight energy gain or loss due to implementation will be magnified even more significantly.

Since the number of fur fibers is an issue, in the industry, a commonly used simplification method is to directly reduce the number of fibers. Meanwhile, in order for the result not to appear more sparse, each of the remaining fibers will be simply thicken. Indeed, reducing the number of fibers this way could alleviate the aforementioned issues. However, as Fig. 3 shows, the rendered appearance is almost completely different to the original.

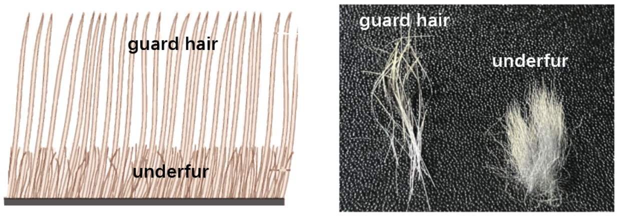


Fig. 5. **Left**: An illustration of the guard hair and underfur. **Right**: A photo of actual rabbit guard hair and underfur. The guard hair is straight, long and thick while the underfur is frizzier, shorter and thinner. For rabbit, the guard hair only occupies about 10% amount of total fur fibers.

We identify this approach as a typical mistake, as pointed out by various appearance prefiltering work [Han et al. 2007; Yan et al. 2014], where the core idea is that geometry and appearance are mutually convertible. The simplified geometry is not gone. Instead, it becomes more complex appearance. Think about the microfacet BRDF as an example, the microfacets are simplified so the macro surface is perfectly flat, but the macro surface must aggregate the mirror-like normals from the microfacets into a normal distribution that is no longer mirror-like.

With the above analysis, we seek a solution to reduce the number of fur fibers by asking the following question: what is the appearance of a fiber, if it is used to represent an aggregation of a bunch of fur fibers? We model the complex appearance of the aggregated fiber in Sec. 4.1.

Guard hair and underfur. For most types of animals, their fur consists of a combination of guard hair on top and underfur beneath. The guard hair is relatively sparse, taking up $\sim$ 10% of the total (as verified in Fig. 5) number of fibers, and is often oily and thick. The underfur is usually not seen directly, and is much thinner but much more, forming a thick layer of cover close to the skin of animals.

The biological functionality of guard hair and underfur is out of the scope of this paper, but the existence of the underfur immediately signals us that they *can be* safely simplified since they are not usually seen directly. And we find that in practice we can also simplify the guard hair together. So, in this paper, we won't distinguish the difference between guard hair and under fur and we simplify fur fibers all together.

The similarity between single fiber and aggregated fiber. Although the BCSDF will complexify after the simplification of geometry, we observe that there is still some interesting similarity between an aggregated fiber's and a single fiber's appearance, as will be elaborated in the next section. Based on the observation, we derive an analytical model for aggregated fur fibers based on the double cylinder model. In the next section, we define this model, and explain how to acquire the parameters for it.

4 OUR METHOD

In this section, we introduce our aggregated BCSDF model mainly concern the multiple scattering effects (Sec. 4.1). Because it is difficult to derive analytical solutions to represent each distribution, we propose a lightweight parameter conversion network to evaluate some scattering terms of our fur model (Sec. 4.2). Finally, we describe how to apply a level of detail representation based on our aggregated fur model to handle complex scenes with many fur characters in the same view frustum (Sec. 4.3).

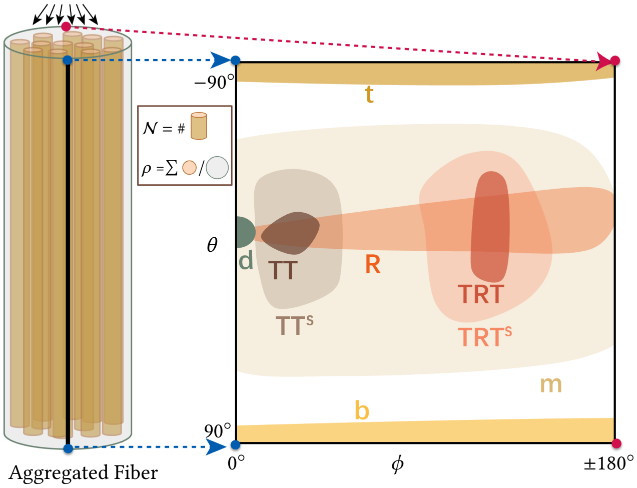


Fig. 6. (Left) An aggregated fur fiber. (Right) Illustration of common positions and shapes of all the lobes in its RDM.

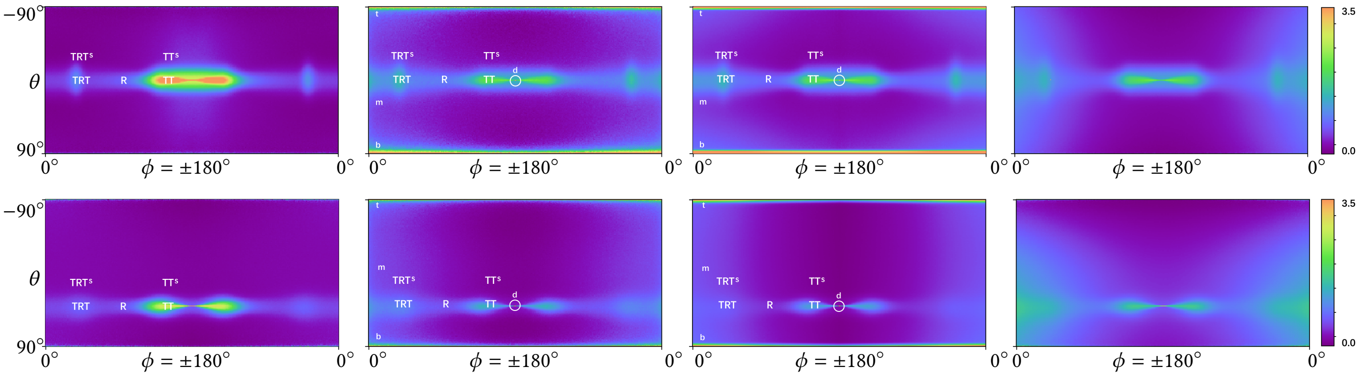
4.1 Aggregated BCSDF Model

We define the aggregated fur fiber as a thick bounding cylinder of a bunch of fibers. And we would like to derive an aggregated BCSDF model in the same form as that described as Eqn. 3. That is, our model is still the sum of distinct lobes, where each lobe is a product of M (longitudinal) and N (azimuthal) profiles. The difference is that we introduce new types of lobes to describe the multiple scattered light bounces inside our aggregated fur fiber.

Before we dive into the components of our BCSDF model, we first explain its input parameters. Based on the parameters of original five-lobe fur BCSDF [Yan et al. 2017a], we further introduce three additional parameters $\mathcal{N}$, ρ and θ_i in our aggregated BCSDF model. The first two parameters are naturally added: $\mathcal{N}$ is the number of fibers inside an aggregated fiber, and the density parameter ρ is ratio between the the total area covered by the $\mathcal{N}$ fibers and the area of the thick cylinder in the cross-sectional plane. Different from the idea that derive an analytical form for the aggregated fiber's behavior from the distributions of inside fibers, we focus on a statistical average method and introduce the longitudinal incident angle θ_i as a parameter.

To describe the aggregated behavior of fur fibers, we start from the observation and analysis of the Radiance Distribution Map (RDM). As illustrated in Fig. 6, an RDM of a fiber, either single or aggregated, is a recording of the exiting energy towards all directions in the θ-ϕ spherical coordinates, given an incident light hitting this fiber from a certain direction, and assuming far field. We generate ground truth RDMs using Monte Carlo random walk until convergence.

By comparing the RDMs (Fig. 7), we first notice that the radiance distribution of an aggregated fiber is similar to those of a single fiber. However, the single fiber's BCSDF is still not capable of replacing the RDM of the aggregated fiber. The main differences between a single fiber and an aggregated fiber are:



(a) single fiber (b) a bunch of fibers (c) Our aggregated model (d) dual scattering

Fig. 7. A comparison of the RDM of (a) single fiber, (b) a bunch of fibers with $\mathcal{N} = 100$ and $\rho = 0.3$, (c) our aggregated model and (d) dual scattering. The top row uses the mouse's parameters and the longitudinal incident angle is $\theta_i = 0°$. The bottom row uses the raccoon's parameters and the longitudinal incident angle is $\theta_i = -45°$. Our aggregated model can closely fit the distinct nine lobes on the RDM. In contrast, dual scattering predicts brighter R and wrong backward scattering, and fails to simulate the bright regions on the top t and bottom b of the RDM.

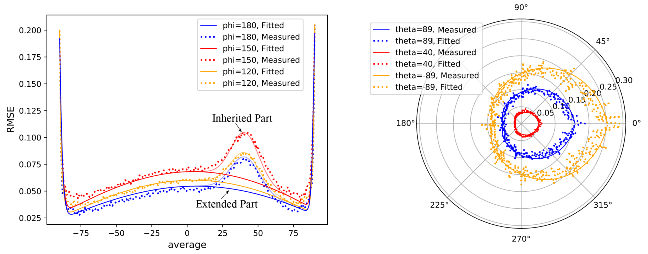


Fig. 8. The 1D profiles extracted from the RDM of an aggregated raccoon fiber ($\mathcal{N} = 100$, $\rho = 0.3$, dots) with a longitudinal incident angle $\theta_i = -45°$, and our fitted profiles (lines). (Left) the longitudinal profiles, where we can clearly observe three lobes located near the angle $\theta_r = -90°$, $0°$ and $90°$. Note specifically that the lobe near $\theta_r = 45°$ is a mixture of inherited lobes (R only for $\phi = 180°$ and $150°$, and R, TT, TT^s for $\phi = 120°$), and we show our fitted profiles w/ and w/o the inherited part. (Right) the azimuthal profiles, where we find that the three lobes share a very similar shape, albeit that they are extracted from different longitudinal angles. Therefore, we use the same spline (with different attenuation) to fit the azimuthal profiles.

(1) Compared to the RDM of a single fiber, the aggregated RDM still has five distinct lobes but dimmer. This is in accordance with the observation by Zinke et al. [2008]–after being multiple scattered, the BCSDF will keep a similar distribution as the single fiber.
(2) The aggregated RDM has noticeable distributions resulted from multiple scattering at the top, middle and bottom regions, marked as t, m and b, respectively. The middle part is mostly backward-scattering, around $\phi = 0$. Due to the Fresnel effect in the longitudinal cross section, part of the energy will keep on deviating from the incident direction and tends to aggregate along the fibers in the end, and finally forms the bright bands on the top and bottom parts in a RDM. This effect is in essence similar to how optical fibers conduct the light running through, just happening on the outside.
(3) Part of the light can directly pass through the aggregated fiber and form a bright point opposite to incident direction. Note again that once aggregated, the original fibers will be gone and only statistics are left. Therefore, this part of energy is not binary (either occluded by a fiber or not), but always exists.

Based on the observations, we classify the radiance distributions of an aggregated fiber into three parts:

Inherited part. (R, TT, TRT, TT^s, TRT^s). Since there is a part of radiance distribution similar to that of a single fur fiber (Fig. 7), we first keep the five lobes the same as single fur fiber BCSDF model (Sec. 3.1) as an inherited part. However, since the inherited part's energy is lower than that of a single fiber, to control the energy of this part, we define an extra scaling parameter $\mu \in (0, 1)$. How to determine the value of μ will be stated in Sec. 4.2.

Extended part. (t, m and b). Since multiple scattering mainly produces three additional regions, top, middle and bottom in the RDM, in practice, we find that using three lobes (t, m and b) is adequate to simulate this entire part of radiance distribution.

In Fig. 8, we show 1D profiles extracted from the RDM longitudinally and azimuthally. As we can see, the three lobes in this extended part also exhibit interesting patterns along the two directions. Similar to any other lobes in the standard BCSDF definition, we study each lobe in this extended part as the product of a longitudinal term M and an azimuthal term N. Note that in this figure we show the fitted inherited lobes and extended lobes separately, and our aggregated could fit those separate components well.

From Fig. 8 (left), we immediately find that the longitudinal lobes of all t, m and b can be faithfully represented as Gaussians:

$$M_{t,m,b}\left(\theta_i, \theta_r\right) = G\left(\theta_{t,m,b}; \theta_r, \beta_{t,m,b}\right), \tag{5}$$

where $\theta_{t,m,b}$ is the center of each lobe, $\beta_{t,m,b}$ is the roughness respectively, and $G(x; \mu, \sigma)$ means the Gaussian function with mean μ and standard deviation σ. How to determine the value of $\theta_{t,m,b}$ and $\beta_{t,m,b}$ will be introduced in Sec. 4.2. Generally, the roughness of the m lobe is relatively large, while the other two are small. Note here we do not describe the amplitude of a lobe longitudinally, following previous approaches.

As shown in Fig. 8 right, the profiles of azimuthal terms are more complex than the longitudinal terms. However, we observe that the three lobes share a very similar shape azimuthally, all gradually decreasing from the back ($\phi = 0$) to the front ($\phi = \pm\pi$). So we unify the distribution of the three azimuthal terms. Also, because this distribution is symmetric in $\phi \in (-\pi, 0)$ and $\phi \in (0, \pi)$, we only need to define the distribution on the range of $\phi \in (-\pi, 0)$. We rely on a cubic interpolated spline controlled by five points to represent the azimuthal distributions:

$$N_{t,m,b}(\phi) = A_{t,m,b} \cdot \mathrm{spline}_{y_{0\ldots4}}(\phi), \tag{6}$$

where the five control points are $(-\pi, y_0)$, $(-\frac{3}{4}\pi, y_1)$, $(-\frac{1}{2}\pi, y_2)$, $(-\frac{1}{4}\pi, y_3)$ and $(0, y_4)$. The $A_{t,m,b}$ is the attenuation term that represents the total energy of each lobe. And the determination of $y_{0\ldots4}$ and $A_{t,m,b}$ will be discussed in Sec. 4.2.

Direct transport (d). The direct transport part is simpler. It is a Dirac delta function opposite to the incident direction:

$$S_d = \delta(\theta_i, -\theta_r) \cdot A_d \cdot \delta(\phi_i, \pi + \phi_r). \tag{7}$$

In practice, we treat the direct transport as a Gaussian with a tiny intrinsic roughness $\beta_d = 0.05$ [Yan et al. 2016], both longitudinally and azimuthally, simply to facilitate renderer integration:

$$S_d = G(\theta_i; -\theta_r, \beta_d) \cdot A_d \cdot G(\phi = \phi_r - \phi_i; \pi, \beta_d). \tag{8}$$

The attenuation term A_d is determined by the statistics $\mathcal{N}$ and ρ of aggregation, since they directly control how frequently multiple scattering can happen, and we find other factors barely contribute to this term. While it is straightforward to assume that A_d exhibits an exponential decay, in practice, we find that using a simple exponential function is unable to represent A_d well.[1] Since our method is data-driven, we directly derive an empirical formula using a symbolic regression software TuringBot [Software 2020],

$$A_d = (1.216 + 0.331 \cdot \mathcal{N} \cdot \rho)^{-0.623} - 0.038, \tag{9}$$

producing a good match throughout our simulated RDM database. In Fig. 9, we verify that the result of our fitted A_d is accurate compared with simulated data.

A schematic diagram of the lobes in our aggregated fur fiber is marked on the RDM in Fig. 6 (right). In Fig. 10, we compare the rendering results using our aggregated model to those rendered with an actual bunch of fibers using the single fur fiber model [Yan et al. 2017a]. In Fig. 11, we compare the rendering results using our aggregated model against those using simulated RDM (regarded as ground truth) in the fur block scene. Our aggregated model is also able to simulate a wide range of simulated RDMs, as shown in Fig. 13, with the corresponding best-fit parameters acquired in the next subsection.

4.2 Lightweight parameter predicting network

Though we now have an aggregated fur BCSDF model to simplify complex light transport inside, it is still far from complete. One immediate question is, how do we find the parameters of the aggregated BCSDF model (Tab. 1, **Output**), if we only know individual fur fibers' parameters and the statistical parameters (Tab. 1, **Input**)? The most straightforward way is to do optimization every time when a set of input parameters is given. However, it is impractical to perform optimization per shading event, since it can happen on every bounce along every light path. Optimization is a good way to provide correct data offline, but we need immediate online queries of the mapped output from any given input.

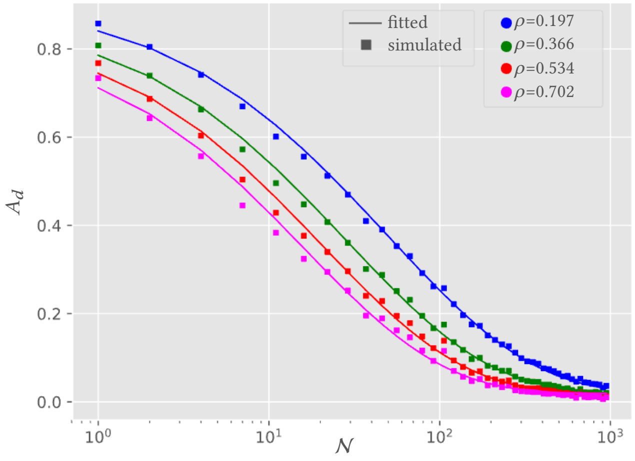


Fig. 9. Comparison of simulated (square points) and fitted (soiled lines) direct transport attenuation term. Using our fitted curve, the profiles have good matches with the simulated data over a wide range of parameters.

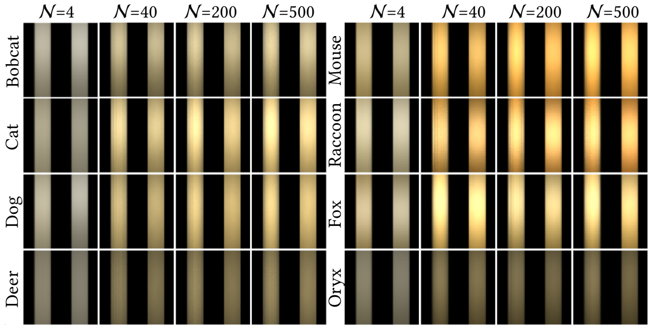


Fig. 10. The actual rendering results of a bunch of fur fibers (left of each insert) and the corresponding aggregated fur fiber (right of each insert). Overall appearance of our aggregated model is very close to the reference. The actual rendering results look like cylinders and our results look like ribbons because our aggregated BCSDF model is far-field.

We propose a lightweight neural network to perform this task, with the aforementioned input and output. The neural network keeps taking input-output pairs as training data, generated using optimization. This step is completely irrelevant to any actual scenes and any species of animals, and needs to be performed only once. During rendering, the network will be inferred to begin the shading process.

Network architecture. The network architecture is shown in Fig. 12. It takes 13 parameters as input include 10 parameters from the single fur fiber BCSDF and 3 statistical parameters as input (Tab.

[1] A similar phenomenon is observed and discussed in the supplementary document in Yan et al. [2017b]. This reason is very likely to be the correlation between fur fibers' positions in the azimuthal section. While a line of research focuses on non-exponential participating media, we keep our method data-driven and do not extend further discussion on the physical cause.

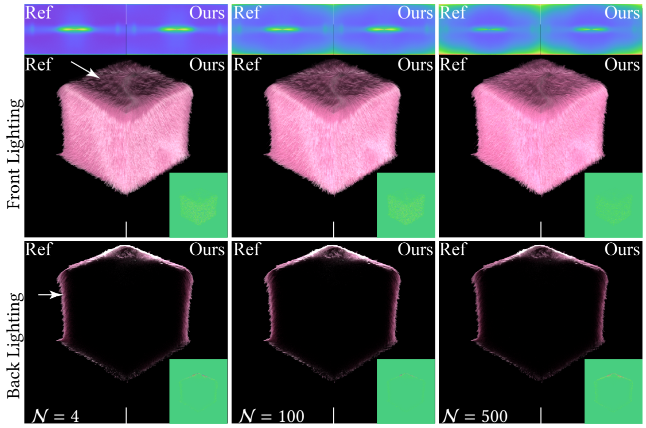


Fig. 11. In this figure, we show the rendering results of the fur block which keeps the same aggregated fiber geometry but with varying $\mathcal{N} = 4, 100, 500$ for each aggregated fiber. We compare the rendering results using our aggregated model against the ground truth (GT) using the simulated RDM. (Top) The RDM of our method and the ground truth. (Middle) Rendering results with a directional light from front to back. (Bottom) Rendering results with a directional light from back to front. With $\mathcal{N}$ increasing, we can observe that and the forward scattering becomes weaker (the highlight on the fur mainly from R lobe) and the backward scattering becomes stronger. In our accompanying video, we show a complete sequence with $\mathcal{N}$ changing continuously from 2 to 1000.

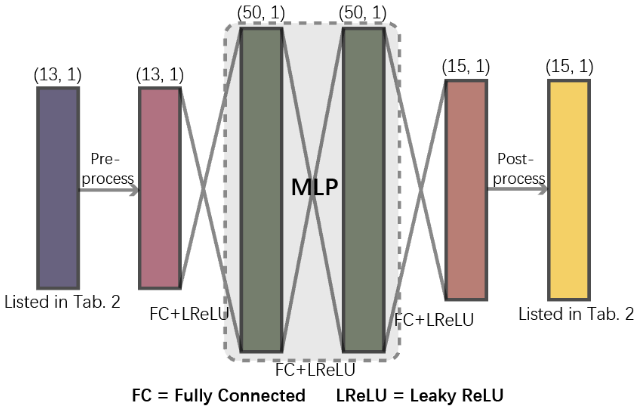


Fig. 12. Our lightweight parameter predicting network architecture is an MLP with two 50-node fully connected hidden layers using Leaky ReLU activation function.

1, **Input**), and outputs 15 parameters to evaluate the aggregated BCSDF (Tab. 1, **Output**). For the input parameters and output parameters, we apply pre/postprocess to map them into proper ranges. Specific pre/postprocess functions for each parameter are listed in Tab. 3. The processed input and output are connected with a multilayer perceptron (MLP) of 2 hidden layers, each with 50 node. How to prepare the training data and train the network will be detailed in Sec. 5.

Table 3. This table shows how to pre/post-process input/out parameters for training the neural network.

Parameter	**Pre / Postprocess**
Input	
κ	$I_0 = 2 \cdot \kappa - 1, \kappa \in (0.65, 0.91)$
η	$I_1 = \eta - 1, \kappa \in (1.19, 1.69)$
α	$I_2 = 0.2 \cdot \alpha, \alpha \in (0.55, 3.48)$
β_m	$I_3 = 4 \cdot \beta_m - 3, \kappa \in (0.098, 0.2)$
β_n	$I_4 = 2 \cdot \beta_n - 1, \beta_n \in (0.083, 0.31)$
$\sigma_{c,a}$	$I_5 = 0.2 \cdot \log(\sigma_{c,a} + 1), \sigma_{c,a} \in (0.04, 1.39)$
$\sigma_{m,a}$	$I_6 = 0.2 \cdot \log(\sigma_{m,a} + 1), \sigma_{m,a} \in (0.1, 0.31)$
$\sigma_{m,s}$	$I_7 = 0.5 \cdot \log(\sigma_{m,s} + 1), \sigma_{m,s} \in (0.65, 0.91)$
g	$I_8 = g, g \in (0.18, 0.79)$
l	$I_9 = 0.5 \cdot l, l \in (0.44, 1.96)$
ρ	$I_{10} = \rho, \rho \in (0.05, 0.9)$
$\mathcal{N}$	$I_{11} = 0.02 \cdot \mathcal{N}^{\frac{1}{2}}, \mathcal{N}^{\frac{1}{2}} \in (1, 31.62)$
θ_i	$I_{12} = 0.2 \cdot \theta_i + 0.5, \theta_i \in (-\frac{1}{2}\pi, \frac{1}{2}\pi)$
Output	
μ	$\mu = O_0$
θ_t	$\theta_t = 5 \cdot (O_1 + 1)$
θ_m	$\theta_m = 10 \cdot O_2$
θ_b	$\theta_b = 5 \cdot (O_3 - 1)$
β_t	$\beta_t = 10 \cdot O_4$
β_m	$\beta_m = 0.5 \cdot O_5 + 1$
β_b	$\beta_b = 10 \cdot O_6$
A_t	$A_t = 0.5 \cdot \log(O_7 + 1)$
A_m	$A_m = 0.1 \cdot \log(O_8 + 1)$
A_b	$A_b = 0.5 \cdot \log(O_9 + 1)$
$y_{0...4}$	$y_n = O_{10+n}$

Essentially, our neural network only serves to compress the high dimensional (input+output) data. We choose a neural network over other methods, such as tensor decomposition, because the dimensions of the input and output spaces are high, but the mapping can be naturally smooth. And decompressing the data is also very efficient running through such simple architecture. Therefore, we also differentiate our method from deep learning methods, because (1) we only need overfitting; (2) there is no need to go deeper, since our simple MLP is already sufficient to map the parameters accurately; and (3) our focus is on simplicity and performance. In practice, our neural network's total inference time only takes 2% of the total rendering time, with an inline CPU-based implementation that does not require any changes to the modern rendering pipeline. And once trained, our neural network can be applied to any scene directly. In Fig. 10 and Fig. 13, we validate that our network produces the right parameters for our aggregated BCSDF model to closely fit the simulated appearance/RDM. And we use our neural network throughout the rendering of all our scenes.

4.3 Level-of-detail simplification

So far, we have a complete introduction of our aggregated BCSDF model, with which we can already simplify the light transport within a bunch of fibers. The next problem is its usage, i.e., how to apply our aggregated model to a furry object practically, and how much should the object be simplified. Cook et al. [2007] introduced a stochastic simplification approach for the aggregated geometry. They proposed

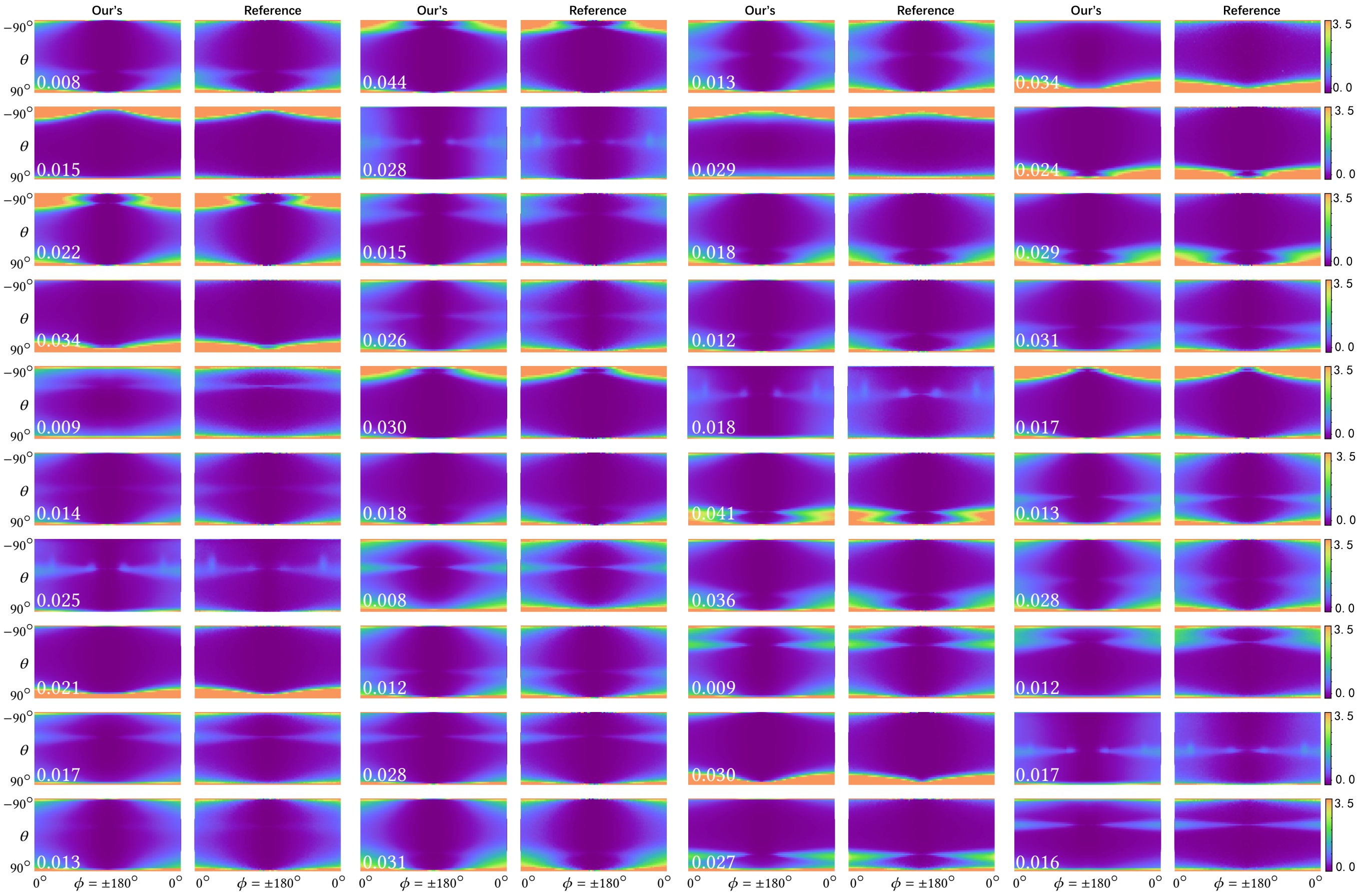


Fig. 13. We compare the RDMs generated using the output parameters from our neural network against the simulated reference. We randomly select the input parameters for all examples. All RDMs are scaled and displayed in logarithmic space for perceptually plausible brightness, and we also list the relMSE for each pair of comparison. The direct transport part is not visualized.

a unified set of requirements for aggregated geometry simplification and we follow these requirements: (1) the appearance should be nearly identical to the reference, (2) the transition should be smooth, e.g. during zooming in/out, or with animated geometry, and (3) the level of simplification should be dynamically adjustable in different scenarios. We propose two different simplification heuristics to meet these simplification requirements, one for the first (primary) bounce of camera paths and the other for subsequent (secondary) bounces. Further, we use a pseudo-random approach to strengthen the temporal coherence.

Simplification for the primary bounce. To simplify fur fibers for the first bounce, we propose a screen-space heuristic that guarantees good rendering result with enough details, since the result from the primary bounce will be directly seen. To ensure a consistently good quality, regardless of different view distances, we specify the same average number of fibers ξ_a covered in each pixel. Then, the total preserved number of fur fibers can be written as

$$\xi_1 = \mathrm{num}_p \cdot \xi_a, \tag{10}$$

where num_p is the approximate number of pixels occupied by the model to be simplified.

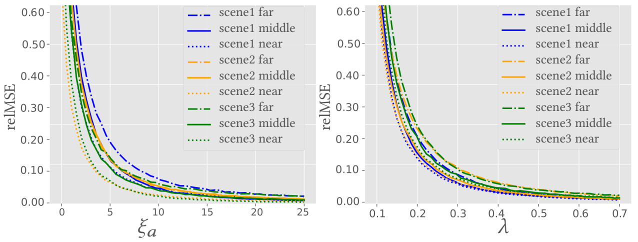


Fig. 14. Left: for the simplification of the first bounce, we plot the relMSE of the rendering results using different ξ_a, for different scenes at different distances to the camera. We find that $\xi_a \in (10, 20)$ is a proper range. Right: for subsequent bounces, we chose different λ and compare the error, still for different scenes at different distances. We find that the rendering results using $\lambda \in (0.3, 0.6)$ can be both efficient and accurate. Scene 1 is the standing wolf, scene 2 is the straight hair, and scene 3 is the crawling hamster.

We could accurately accumulate num_p by casting primary rays towards the model and counting the number of intersections as a pre-rendering process, but this can be too costly. Instead, as illustrated in Fig. 15 (left), we use a simple approximation by bounding a sphere

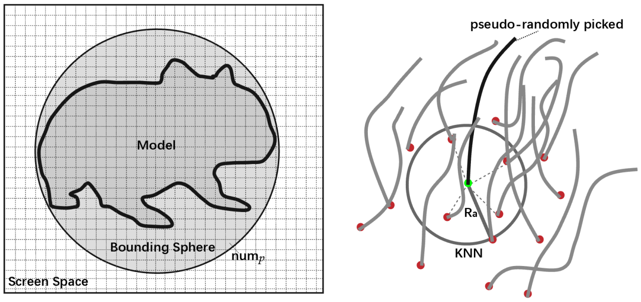


Fig. 15. Illustration of our heuristic simplification on (Left) how to calculate the number of pixels (num_p) occupied by a model, and (Right) how to determine the radius of an aggregated fiber (R_a) using KNN.

tightly around the model and then calculating the number of pixels the bounding sphere occupies in screen space:

$$\text{num}_p = \text{num}_{\text{film}} \cdot \frac{\pi \cdot R_s^2}{4 \cdot \|O - C\|^2 \cdot \tan^2 \theta_{\text{fov}}}, \tag{11}$$

where num_{film} is the screen resolution. C is the center of the bounding sphere, whose radius is R_s. O is the position of the camera and the θ_{fov} is the field of view (fov) of the camera.

Simplification for secondary bounces. Since the multiple scattering tends to be smoother and is not directly seen, we reduce the number of fibers for secondary bounces more aggressively, by further introducing a fixed-percentage reduction:

$$\xi_n = \xi_1 \cdot \lambda^{\max\{n-1,4\}} \quad (n \geq 2), \tag{12}$$

according to the number of bounces n a light path has undergone, where λ is the additional reduction ratio, which is again specified by users. Therefore, ξ_n is the total number of preserved fur fibers for the n-th bounce.

For a given model, we pre-calculate its different levels of simplification according to Eqs. 10 to 12, but we only need to build one BVH according to the unsimplified fur fibers, shared among all different levels. During rendering, we determine the desired level of simplification according to our simplification heuristics.

Determining user-specified parameters. An important practical concern is to find reasonable values for ξ_a and λ. Referring to Fig. 14, we can reach a conclusion that the $\xi_a \in (10, 20)$ and the $\lambda \in (0.3, 0.6)$ are reasonable ranges for our heuristic simplification. We emphasize that ξ_a is not the percentage of fur fibers preserved. Even with a fixed ξ_a, the level of simplification can be different per different views, resolutions, objects and the original number of fur fibers. The power of ξ_a for dynamic simplification can be seen throughout all our results. In Fig. 16, we show the effects of using different λ for secondary bounces.

Selection of simplified fibers. To actually reduce the number of fibers, we randomly select the fibers to keep (or discard). Similar to the pervasively used method in procedural appearance modeling [Jakob et al. 2014], we assign the same low-discrepancy sequence to the same object, which avoids clumping, and ensures stratification of the selected fibers. Moreover, we fix the the random sequence to a same object which ensures the temporal stability.

Determining aggregated statistics. We need to determine the parameters changed during simplification, including the statistical parameters ($\mathcal{N}$, ρ, $\sigma_{c,a}$ and $\sigma_{m,a}$) for our BCSDF model and the radius of the aggregated fiber (R_a).

We assume that the $\mathcal{N}$ of the each aggregated fiber is uniform in one model. Then it can be immediately written as $\mathcal{N} = \xi_0/\xi_n$, where ξ_0 is the total number of fibers originally.

If we want to keep the same appearance as the original fur model, we must ensure that an aggregated fiber closely represents the simplified fur fibers. So, when we determine the radius of an aggregated fiber, we apply the k-nearest neighbor (KNN) to find the $\mathcal{N}$ nearest roots of single fibers around the center of the aggregated fiber's root. And R_a is equal to the maximum radius attained from KNN (Fig. 15 right).

The density ρ equals to the sum of the cross-sectional area of single fibers over the cross-sectional area of the aggregated fiber. However, ρ is not a constant even for one aggregated fiber, because the radius of a single fur fiber is usually variable: the radius around the root of a fiber is thick, while the radius around the tip can be very thin. This is an important effect named taper in the industry, which also happens in reality, making the appearance look furry. In this paper, we also consider the taper effect, not by making the radius of an aggregated fiber thinner around its tip (therefore, it stays to be a cylinder), but by simply scaling the density ρ according to single fur fibers' taper values (a number between 0 and 1) from root to tip.

For other aggregated BCSDF parameters, we assume that an object contains the same species of fur fibers, thus only the two absorption parameters $\sigma_{c,a}$ and $\sigma_{m,a}$ are variable, which affect the color of the fur appearance. We average $\sigma_{c,a}$ and $\sigma_{m,a}$ of single fibers in one aggregated fiber to determine the $\sigma_{c,a}$ and $\sigma_{m,a}$ for the aggregated fiber.

5 IMPLEMENTATION DETAILS

5.1 Data preparation for training

As mentioned in section 4.2, our lightweight neural network is designed to compress the parameter conversion and can infer output parameters very efficiently in the rendering stage. But the data preparation process for network training is not trivial and we will explain the details in this section.

Simulating the RDM. We produce one RDM for an aggregated fiber with a group of input parameters. We first generate the geometry of the aggregated fiber which contains $\mathcal{N} \in (1, 1000)$ single fibers with the Poisson distribution. Then we simulate the light bouncing process which is similar as the photon mapping method [Jensen 1996]. One photon initially carries the energy of 1 and is shot into the aggregated fur fiber from a certain longitudinal direction (θ_i), and we record the direction where the photon leaves and the energy it carries. Finally, we classify the outgoing directions into 180×360 bins and calculate the radiance of each bin. In our implementation, we emit 40 million photons for an aggregated fiber and get a RDM with the resolution of 180×360. The implementation of single fur BCSDF in the photon mapping stage can refer to [Yan et al. 2017a] and their measurement data.

Generating the RDMs. We generate 20,000 RDM maps in the whole parameter space for one aggregated fiber. Firstly, We generate 1000

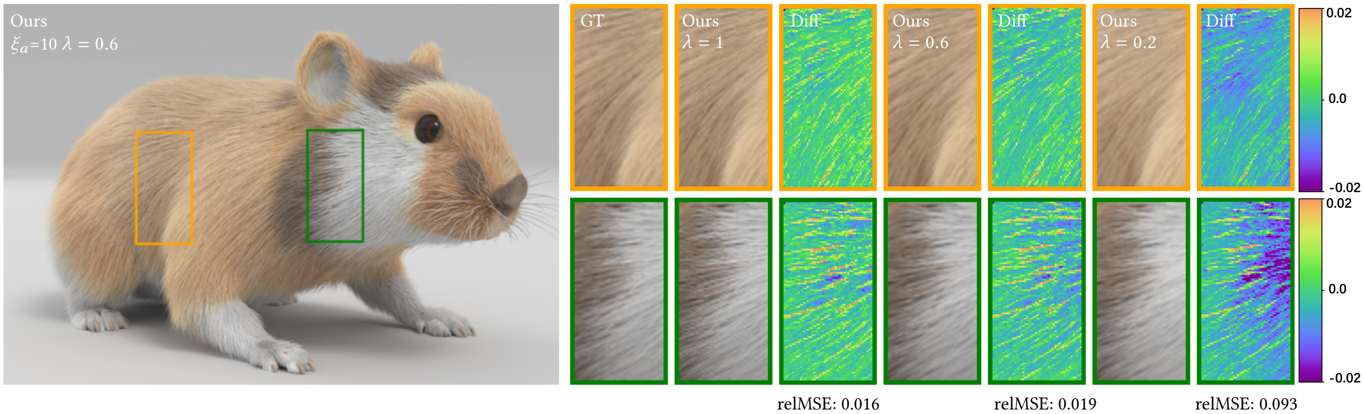


Fig. 16. This figure shows the validation of the secondary bounces heuristic simplification. We set the $\xi_a = 10$, about half of the fibers are simplified for the first bounce. We compare the rendering results to the ground truth and show the difference maps for $\lambda = 1.0$, $\lambda = 0.6$, $\lambda = 0.2$. In this figure, all the results are converged.

aggregated fibers' geometries with different ($\mathcal{N} \in (1, 1000)$). Then we randomly choose the input values in the remaining 12 input parameters' space. With the RDM simulation method mentioned above, we normally will spend about 59h to generate all the RDMs for one kind of fur. Good things are that the generic parameter conversion network needs to be trained only once and we do not need the RDMs in the rendering stage.

Fitting the output parameters. Before training, we first fit each RDM and get the corresponding output parameters by using Ceres Solver. The fittings are performed in logarithmic space, with the cost function defined as the sum of all per-pixel differences of the normalized log-measured and log-fitted values. The initial values of all parameters are manually set.

We train our network in the PyTorch framework and the training process spends 20 minutes with 120 epochs.

5.2 Renderer integration of the neural network

Our neural network can be integrated into a Monte Carlo rendering pipeline and we implement our method on the Mitusba rendering engine [Jakob 2010]. Because our network is lightweight, we execute the inferring process on CPU to avoid transmission delay. We directly transcribe the network as a matrix and infer our network for direct evaluation and importance sampling. Inferring the network once with single thread only takes 0.7 microseconds, and the total inferring time only takes about 2% of the whole rendering process, which is almost negligible.

5.3 Importance sampling

Our importance sampling is similar to d'Eon et al. [2013] and can be summarized in the following three stages:

- Choosing a lobe p to sample. The lobes are weighted according to the energy they carry. Since the longitudinal term M_p and the azimuthal distribution term D_p are normalized, the energy that lobe p carries depends on its attenuation term A_p. We calculate A_p for each of nine lobes and the sum of all A_p. The probability that we choose a lobe p is determined by A_p over the sum of all A_p.
- Sampling an outgoing direction. Similar to d'Eon et al. [2013], we separately sample the selected lobe p in longitudinal and azimuthal following their distributions. Since lobes R, TT, TRT, t, m and b are in Gaussian distributions in longitudinal, it is easy to sample the longitudinal outgoing direction θ_r. For TT^s and TRT^s, since they're smooth, we use cosine sampling function. Since lobes R, TT, TRT and d are also in Gaussian distributions in azimuthal, we apply Gaussian sampling according to the lobe's distribution. The outgoing azimuthal direction is $\phi_r = \phi_i + \phi$, where ϕ is the relative outgoing azimuthal angle. For lobes TT^s, TRT^s, t, m and b, we uniformly sample them over the azimuthal directions.
- Calculating PDF and sampling weight. The final probability density function (PDF) is the product of the selected lobe p's azimuthal PDF and longitudinal PDF followed by a conversion from (θ, ϕ) space to solid angle space. The final sampling weight is the selected p's BCSDF value divided by its chosen probability.

6 RESULTS, VALIDATION AND COMPARISON

In this section, we show the rendering results of our aggregated BCSDF model. We also compare our method to Yan et al. [2017a] for equal time (ET) and equal quality (EQ) and use relative mean square error (relMSE) to evaluate the difference between ours and the ground truth. The ground truth is the converged result rendered by Yan et al. [2017a]. If not specified explicitly, we set $\xi_{\text{average}} = 10$ and $\lambda = 0.6$ for heuristic simplification and compare the rendering time of our method to that of the EQ method. All scenes are rendered using path tracing on an Intel 20-core i9-10900K machine. We list test scenes' configurations and performances in Tab. 4 and will release our source code and trained neural network upon publication. Please also be sure to check out our accompanying video, where we provide more comparisons on animations.

A bowl of hamsters. This scene includes 120 hamsters and has 147 million fur fibers in total. In Fig. 1, we show the rendering results of ours, equal time (ET), equal quality (EQ), and the relMSE. Our method achieves about 13.5× speed up with 0.014 as relMSE. There are a lot of hamsters in this scene and all of them are far from

Table 4. Statistics of all the scenes and performance. The references are rendered using Yan et al. [2017a]. We list and compare the number of fur fibers, the average number of bounces along a light path, samples per pixel, rendering time (range for the video and blue text for the figures) and the error between our method, equal time (ET) reference, equal quality (EQ) reference and the ground truth.

Scenes	**#Fibers**		**#Bounces**		**#Samples**			**Time** (min)			**relMSE**		
	Ref.	Ours	Ref.	Ours	ET	EQ	Ours	ET	EQ	Ours	ET	EQ	Ours
A bowl of hamsters	147M	1.8M	119	14	82	1175	512	26.7	360	26.8	0.131	0.014	0.014
Crawling hamster	1.5M	0.085M-1.5M/0.7M	48	8	239	855	512	12.5	74.2	0.47-14.2/12.5	0.077	0.021	0.021
Living room	105M	6.5M-57.2M/7.5M	95	16	103	1344	512	38.0	466	35.3-41.7/38.0	0.142	0.008	0.008
Chameleon array	256M	8.7M-11.6M/9.3M	76	19	144	984	512	68.9	492	64.5-69.8/68.7	0.117	0.010	0.010
Straight hair	15K	6K	24	6	206	710	512	17.1	65.2	17.1	0.105	0.013	0.013
Standing wolf	12M	2.1M	105	27	192	1700	512	62.2	530	62.5	0.194	0.042	0.042

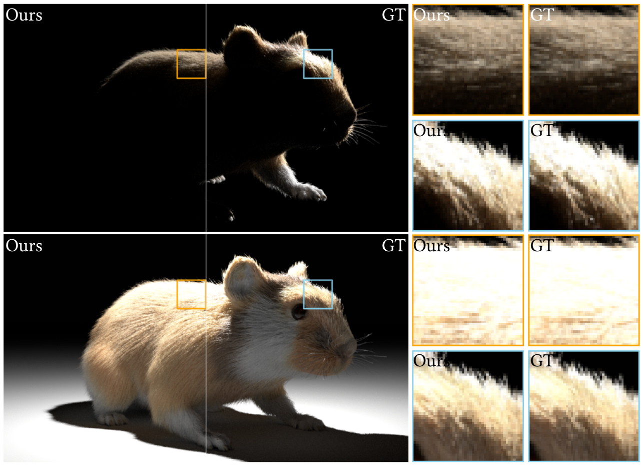


Fig. 17. This figure shows the rendering results of the crawling hamster under sharp lighting. In this scene, our aggregated model keeps about half of the fibers for the first bounce. (Top) Rendering results with a directional light from back to front. (Bottom) Rendering results with a point light in front. Our method matches the ground truth well without introducing obvious bias.

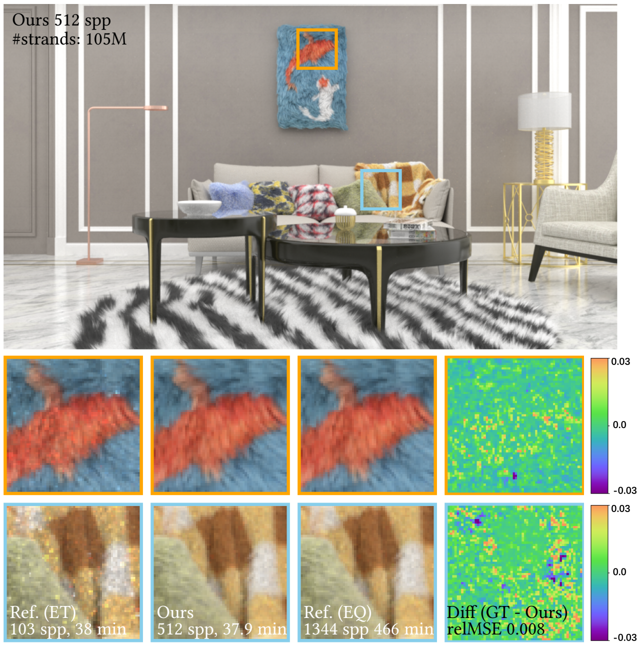


Fig. 18. A fuzzy living room with fur decorative picture, pillows, blanket and carpet. We randomly set the single fiber parameters for each fur object, and we compare the equal time (ET) and equal quality (EQ) rendering results with the reference (Ref.) [Yan et al. 2017a] to our method. The relMSE of our method is only 0.008 and achieves 12.3× speedup.

the camera, so we achieve a drastic simplification and only use 1.8 million fur fibers in total.

Crawling hamster. Fig. 16 shows the rendering results of the crawling hamster with different secondary bounces simplification degrees ($\lambda = 1$, $\lambda = 0.6$ and $\lambda = 0.2$). The rendering results match the curve in Fig. 14. In practice, we choose $\lambda = 0.6$ for secondary bounces' simplification and achieve an additional 2.6× speedup compared to the case of $\lambda = 1$. The relMSE value is 0.019 which is also reasonable. In the video, we show a zooming in/out sequence to verify the temporal stability of our method with different levels of dynamic simplification per frame. Fig. 17 shows the rendering results of the crawling hamster under sharp lighting (intense directional light and point light). Our method still maintains the highlight distribution on the fur.

Living room. In this scene, there are many different furry objects with 105M fur fibers in total. We randomly assign parameters in the whole parameter space for each fur object. Fig. 18 shows the equal time (ET) and equal quality (EQ) comparison viewed from far away. The EQ's rendering time is 12.3× longer than ours with an relMSE value of 0.008. Our method could closely match the ground truth although we only use 7.5M fur fibers in this scene. A zoom in/out sequence is also available in our video.

Chameleon array. There are 64 chameleons in this dynamic scene (Fig. 19), with 256M fur fibers in total. Each column uses a different kind of measured fur fiber parameters referring to Yan et al. [2017a]. Our method drastically reduces the number of fur fibers to 9M-11M, and is able to handle the aggregation of fur fibers with heterogeneous colors defined with a texture. Moreover, our method preserves temporal stability as the scene animates. Our method is 7.2× faster than the EQ reference, and the relMSE is 0.010. We also visualize the different simplification levels.

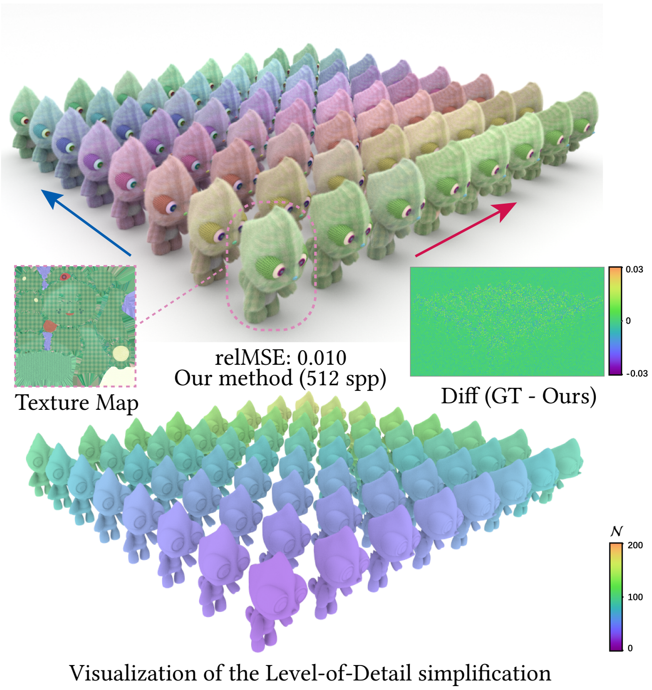


Fig. 19. We render an array of 64 furry chameleon dolls with 512 spp and the relMSE of our rendering result is 0.01. We show the rendering result, difference map and the visualization of the heuristic simplification. Every chameleon has a different set of fur parameters. Following the blue arrow from front to back, we use different texture maps that define spatially-varying colors, and we visualize one of the texture maps. Following the red arrow from left to right, we set the fur parameters as those from bobcat, cat, deer, fox, raccoon, mouse, dog and oryx.

Standing wolf. The standing wolf has 12 million fur fibers and we use 2.1 million fur fibers in our method. In Fig. 20, we compare the existing hair/fur multiple scattering approximation methods including the BSSRDF method [Yan et al. 2017b], the volume method [Moon et al. 2008] and the dual scattering method [Zinke et al. 2008] to our aggregated BCSDF model. We show the equal time (ET) rendering results and compare several methods' relMSE values. Ours relMSE is 0.042. Compared with 0.129 of the BSSRDF method, 0.228 of the volume method, and 0.682 of the dual scattering method, Our method is more accurate than other multiple scattering approximation methods.

Straight hair. In Fig. 21, we show the results of hair/fur multiple scattering approximation methods and our aggregated BCSDF model rendering with equal time. Our result closely matches the ground truth while other methods have different kinds of inaccuracies. This scene also proves that even in close up view with extremely few human hair fibers(that our fur appearance model is not designed to work with), our method is still able to halve the number of hair fibers without introducing obvious artifacts.

7 DISCUSSION AND LIMITATIONS

In this section, we extend the discussion to a deeper understanding of our method.

Orthogonality w.r.t. other methods. Though we compare with different methods, all aimed at more efficient multiple scattering of light, we would like to emphasize again on one of the properties of our method: orthogonality. Our aggregated BCSDF model is essentially a new single scattering model, just with fewer fur strands. Therefore, previous methods such as dual scattering and volumetric/BSSRDF conversion can still be used in combination with our model. This is a unique advantage of our method.

Design without simplification. Following the knowledge that our aggregated BCSDF model is essentially a single scattering model, we would like to provide another way to understand it. Since our model takes in only additional statistical parameters such as $\mathcal{N}$, it does not need to know the original unsimplified geometry. Therefore, it is possible to directly generate a desired number of fur fibers, and use our model with user-defined statistical parameters, bypassing the simplification process. We believe this is of immediate benefit to improve those applications currently using the naïve simplification method (e.g., in most video games where the number of fur fibers is already small), just by replacing the single scattering model with ours and provide a reasonable $\mathcal{N}$ and so on.

Next, though our method has been demonstrated effective in a variety of situations, there are limitations that we would like to point out for the concerns of practical use and future research.

Aggregation and far-field BCSDF. Ideally, during aggregation, all detailed appearance from individual elements should be completely preserved. However, we do make simplifications for practicality. As Fig. 23 (a) shows, we can no longer observe original fur fibers inside an aggregated fiber, which is partly due to using the far-field model, and partly because we simulate and fit the average RDM of different realizations of fibers inside. We raise two related questions for future study. First, is it possible to make the aggregated fur appearance even closer to the original, with all the details and variations, but without making the model unnecessarily complex to evaluate? Second and more important, since the use of aggregation is mostly tied with the context of reasonable approximation, is there an evidence that a more accurate aggregation is required?

Fly-away fibers and fiber misalignment. Since our model aggregates *similar* fur fibers into one, an immediate issue of concern is on the fly-away fibers. This issue is the most severe when fur fibers are extremely long and sparse, which arguably does not happen frequently on animals, but rather often on human hair, as Fig. 23 (b) shows. In this figure, we can also observe that the curly hair fibers are not parallel, but the overall appearance is still close to the reference. We further show the impact of the fiber misalignment in Fig. 22 where we found that the aggregated BCSDF RDM won't significantly change if the standard deviation of misalignment is less than 12°.

Spatially-varying simplification. Currently, our simplification heuristic is designed for an entire object, which already demonstrates the practicality of our aggregated model. For future improvement, it is useful to design more complex heuristics that performs spatially-varying simplification. For example, those hidden fur fibers from the view point can be simplified more drastically, while the reduction of the number of fly-away fibers can be less aggressive. In the

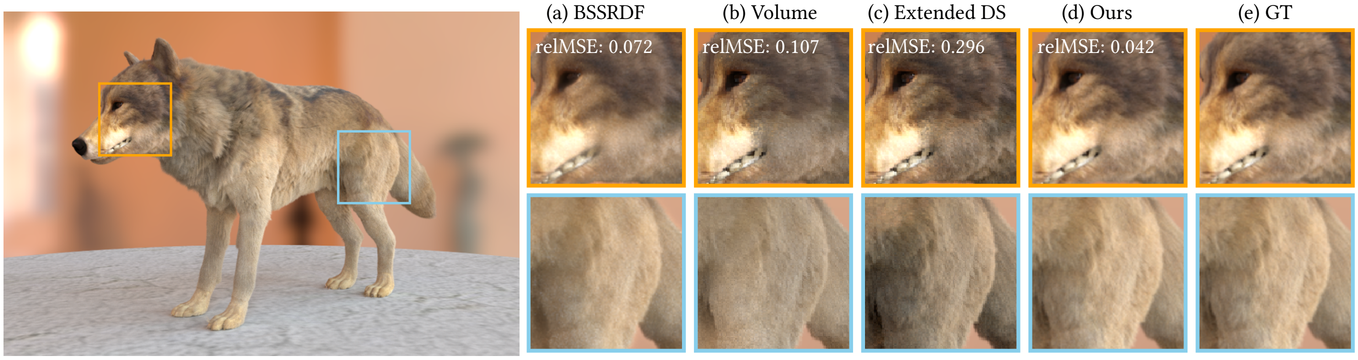


Fig. 20. Rendering of the standing wolf with different methods. The BSSRDF rendering method [Yan et al. 2017b] matches the reference well, but the result is over-blurred. The volume rendering method [Moon et al. 2008] looks flat and faded because the pre-computed Spherical Harmonics (SH) fails to represent complex distributions of fur fibers accurately. The rendering result of the extended dual scattering [Zinke et al. 2008] is apparently darker and dryer than the reference, since this method completely bypasses the multiple scattering, and uses an ad-hoc guess to determine the multiple-scattering energy, which could be higher or lower than the reference. Our method produces the least amount of bias perceptually, without noticeable overblur.

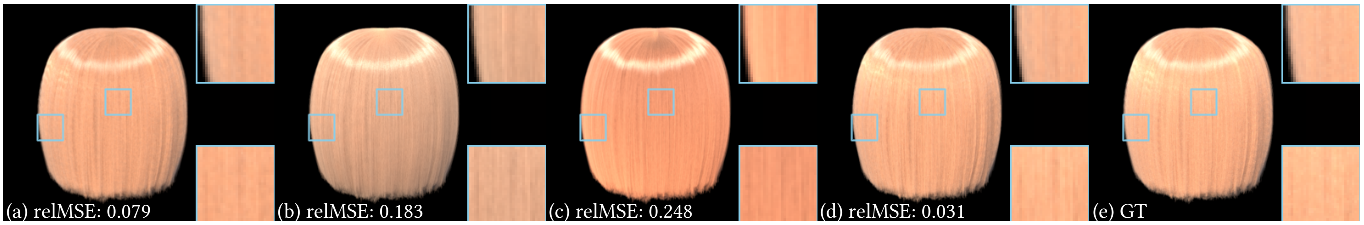
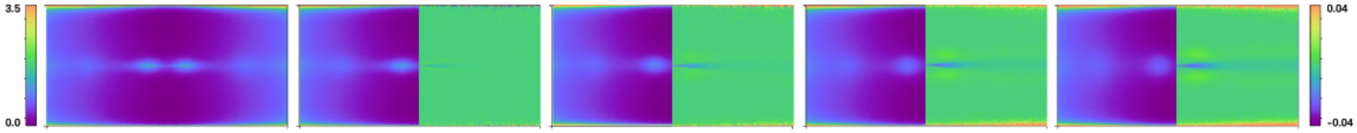
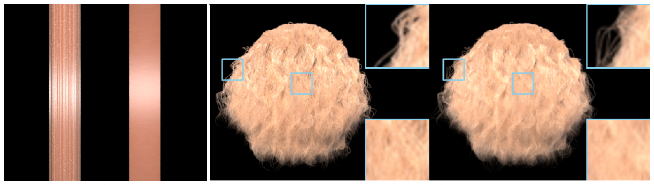


Fig. 21. We compare different rendering methods on the straight hair scene. (a) BSSRDF method [Yan et al. 2017b], (b) volume method [Moon et al. 2008], and (c) classic dual scattering [Zinke et al. 2008] will lead to some bias, while (d) our aggregated model closely matches (e) the reference. Even in close up view with extremely few hair fibers, our method is still able to halve the number of hair fibers without introducing obvious artifacts.



(a) no misalignment (b) $\sigma = 6^\circ$ (c) $\sigma = 12^\circ$ (d) $\sigma = 18^\circ$ (e) $\sigma = 24^\circ$

Fig. 22. We simulate the misalignment of fibers and show the RDMs with different extent of misalignment. The direction of individual fibers is modeled as a Gaussian distribution with different standard deviations σ from 0° to 24°. We also show the difference images between the RDM (a) without misalignment and other RDMs (b-e) with misalignment on the right of each subfigure.



(a) (b)

Fig. 23. (a) a bunch of fibers (left) and our aggregated fiber (right), our method ignores the details of individual fibers in a very close view. (b) curly hair with fly-away fibers, our simplification method (left) removes some fly-away fibers, which is undesired as compared to the ground truth (right).

future, we can also consider setting the level of simplification according to the exact footprint of different bounces along light paths, as proposed by Müller et al. [2021].

Reciprocity and bias. Since our method is data-driven and its parameters are also aware of the incident direction, it is difficult to carry out a theoretical analysis on the reciprocity. Moreover, the use of neural networks can introduce an uncontrollable amount of bias to our model. While theoretically difficult, we do not observe practical issues: if the original single scattering BCSDFs are reciprocal, the aggregation is close to perfect. And if the neural network has reasonably converged, our model will produce similar results to the reciprocal and unbiased reference.

8 CONCLUSION AND FUTURE WORK

We have presented a practical BCSDF model to describe the aggregated appearance from a bunch of animal fur fibers. We take advantage of the representation ability of the single scattering fur BCSDF model [Yan et al. 2017a], and extend it to take aggregation statistics into account and introduce the dependency of the incident direction. Then we refer to a lightweight neural network to evaluate the parameters of our aggregated model from individual fur fiber's BCSDFs. Finally, we come up with a practical level-of-detail heuristic that dynamically simplifies the fiber geometry based on the viewing distance and different number of bounces along a light path, achieving reliable appearance aggregation together with controllable geometry simplification. Our method produces nearly the same results as the ground truth, but performs 3.8×-13.5× faster. We believe that our method can immediately benefit the industry, and is an important contribution to the long-standing research problem of geometry/appearance prefiltering.

In the near future, one natural extension to our method is an optimized and simplified implementation of our aggregated BCSDF model specifically tailored for real-time applications. Since the added lobes in our model are based on observation, an artist-friendly interpretation is also possible to advocate a wider use of our aggregated BCSDF model. We also believe that our model can be generalized to handle cloth rendering, where the aggregated appearance from an bunch of cloth fibers, i.e., a ply or a yarn, could be developed in a similar way. A deeper study regarding fly-way fibers and spatially-varying simplification will also help to improve the practicality of our model on human hair appearance.

ACKNOWLEDGMENTS

This work has been partially supported by the National Key R&D Program of China under grant No.2020YFB1709200, the National Natural Science Foundation of China under grant No.61872223. Ling-Qi Yan is supported by gift funds from Adobe, Dimension 5 and XVerse.

REFERENCES

Matt Jen-Yuan Chiang, Benedikt Bitterli, Chuck Tappan, and Brent Burley. 2016. A practical and controllable hair and fur model for production path tracing. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 275–283.

Robert L Cook, John Halstead, Maxwell Planck, and David Ryu. 2007. Stochastic simplification of aggregate detail. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 79–es.

Thomas Davies, Derek Nowrouzezahrai, and Alec Jacobson. 2020. Overfit Neural Networks as a Compact Shape Representation. arXiv:2009.09808 [cs.GR]

Eugene d'Eon, Guillaume Francois, Martin Hill, Joe Letteri, and Jean-Marie Aubry. 2011. An energy-conserving hair reflectance model. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library, 1181–1187.

Eugene d'Eon, Steve Marschner, and Johannes Hanika. 2013. Importance sampling for physically-based hair fiber models. In *SIGGRAPH Asia 2013 Technical Briefs*. 1–4.

Jonathan Dupuy, Eric Heitz, Jean-Claude Iehl, Pierre Poulin, Fabrice Neyret, and Victor Ostromoukhov. 2013. Linear efficient antialiased displacement and reflectance mapping. *ACM Trans. Graph.* 32, 6 (2013), 1–11.

EA DICE. 2006. *Frostbite Engine.* https://www.ea.com/frostbite

Epic Games. 2019. *Unreal Engine.* https://www.unrealengine.com

Charles Han, Bo Sun, Ravi Ramamoorthi, and Eitan Grinspun. 2007. Frequency domain normal map filtering. *ACM Trans. Graph.* 26, 3 (2007), 28.

Christophe Hery and Ravi Ramamoorthi. 2012. Importance sampling of reflection from hair fibers. *Journal of Computer Graphics Techniques (JCGT)* 1, 1 (2012), 1–17.

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

Wenzel Jakob, Miloš Hašan, Ling-Qi Yan, Jason Lawrence, Ravi Ramamoorthi, and Steve Marschner. 2014. Discrete stochastic microfacet models. *ACM Trans. Graph.* 33, 4 (2014), 1–10.

Henrik Wann Jensen. 1996. Global illumination using photon maps. In *Eurographics workshop on Rendering techniques*. Springer, 21–30.

J. T. Kajiya and T. L. Kay. 1989. Rendering Fur with Three Dimensional Textures. In *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '89)*. Association for Computing Machinery, New York, NY, USA, 271–280. https://doi.org/10.1145/74333.74361

Nima Khademi Kalantari, Steve Bako, and Pradeep Sen. 2015. A Machine Learning Approach for Filtering Monte Carlo Noise. *ACM Trans. Graph.* 34, 4, Article 122 (July 2015), 12 pages.

Pramook Khungurn and Steve Marschner. 2017. Azimuthal Scattering from Elliptical Hair Fibers. *ACM Trans. Graph.* 36, 2, Article 13 (apr 2017), 23 pages. https://doi.org/10.1145/2998578

Pramook Khungurn, Daniel Schroeder, Shuang Zhao, S Marschner, and K Bala. 2015. Matching micro-appearance models to real fabrics. *ACM Trans. Graph* 3, 10.1145 (2015), 2818648.

Richard Lee and Carol O'Sullivan. 2007. Accelerated Light Propagation Through Participating Media.. In *VG@ Eurographics*. 17–23.

Stephen R Marschner, Henrik Wann Jensen, Mike Cammarano, Steve Worley, and Pat Hanrahan. 2003. Light scattering from human hair fibers. *ACM Trans. Graph.* 22, 3 (2003), 780–791.

Johannes Meng, Marios Papas, Ralf Habel, Carsten Dachsbacher, Steve Marschner, Markus H Gross, and Wojciech Jarosz. 2015. Multi-scale modeling and rendering of granular materials. *ACM Trans. Graph.* 34, 4 (2015), 49–1.

Jonathan T Moon and Stephen R Marschner. 2006. Simulating multiple scattering in hair using a photon mapping approach. *ACM Trans. Graph.* 25, 3 (2006), 1067–1074.

J. T. Moon, B. Walter, and S. Marschner. 2008. Efficient multiple scattering in hair using spherical harmonics. *ACM Transactions on Graphics* 27, 3 (2008), 1–7.

Jonathan T Moon, Bruce Walter, and Stephen R Marschner. 2007. Rendering discrete random media using precomputed scattering solutions. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*. 231–242.

Thomas Müller, Marios Papas, Markus Gross, Wojciech Jarosz, and Jan Novák. 2016. Efficient rendering of heterogeneous polydisperse granular media. *ACM Trans. Graph.* 35, 6 (2016), 1–14.

Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. 2021. Real-time neural radiance caching for path tracing. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–16.

Marc Olano and Dan Baker. 2010. LEAN Mapping. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (Washington, D.C.) *(I3D '10)*. Association for Computing Machinery, New York, NY, USA, 181–188. https://doi.org/10.1145/1730804.1730834

Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. 2015. Image based relighting using neural networks. *ACM Trans. Graph.* 34, 4 (2015), 1–12.

Peiran Ren, Jiaping Wang, Minmin Gong, Stephen Lin, Xin Tong, and Baining Guo. 2013. Global illumination with radiance regression functions. *ACM Trans. Graph.* 32, 4 (2013), 1–12.

Aaron A Sandel. 2013. Brief communication: Hair density and body mass in mammals and the evolution of human hairlessness. *American journal of physical anthropology* 152, 1 (2013), 145–150.

TuringBot Software. 2020. TuringBot. https://turingbotsoftware.com.

Delio Vicini, Wenzel Jakob, and Anton Kaplanyan. 2021. A Non-Exponential Transmittance Model for Volumetric Scene Representations. *ACM Trans. Graph.* 40, 4, Article 136 (jul 2021), 16 pages. https://doi.org/10.1145/3450626.3459815

Lifan Wu, Shuang Zhao, Ling-Qi Yan, and Ravi Ramamoorthi. 2019. Accurate appearance preserving prefiltering for rendering displacement-mapped surfaces. *ACM Trans. Graph.* 38, 4 (2019), 1–14.

Mengqi Xia, Bruce Walter, Eric Michielssen, David Bindel, and Steve Marschner. 2020. A wave optics based fiber scattering model. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–16.

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 Trans. Graph.* 33, 4 (2014), 1–9.

Ling-Qi Yan, Miloš Hašan, Steve Marschner, and Ravi Ramamoorthi. 2016. Position-normal distributions for efficient rendering of specular microstructure. *ACM Trans. Graph.* 35, 4 (2016), 56.

Ling-Qi Yan, H. W. Jensen, and R. Ramamoorthi. 2017a. An efficient and practical near and far field fur reflectance model. *ACM Trans. Graph.* 36, 4 (2017), 1–13.

Ling-Qi Yan, Weilun Sun, Henrik Wann Jensen, and Ravi Ramamoorthi. 2017b. A BSSRDF Model for Efficient Rendering of Fur with Global Illumination. *ACM Trans. Graph.* 36, 6, Article 208 (Nov. 2017), 13 pages. https://doi.org/10.1145/3130800.3130802

Ling-Qi Yan, Chi-Wei Tseng, Henrik Wann Jensen, and Ravi Ramamoorthi. 2015. Physically-accurate fur reflectance: Modeling, measurement and rendering. *ACM Trans. Graph.* 34, 6 (2015), 1–13.

Shuang Zhao, Lifan Wu, Frédo Durand, and Ravi Ramamoorthi. 2016. Downsampling scattering parameters for rendering anisotropic media. *ACM Trans. Graph.* 35, 6 (2016), 1–11.

Junqiu Zhu, Yaoyi Bai, Zilin Xu, Steve Bako, Edgar Velázquez-Armendáriz, Lu Wang, Pradeep Sen, Miloš Hašan, and Ling-Qi Yan. 2021. Neural complex luminaires: representation and rendering. *ACM Trans. Graph.* 40, 4 (2021), 1–12.

Arno Zinke and Andreas Weber. 2007. Light scattering from filaments. *IEEE Transactions on Visualization and Computer Graphics* 13, 2 (2007), 342–356.

Arno Zinke, Cem Yuksel, Andreas Weber, and John Keyser. 2008. Dual Scattering Approximation for Fast Multiple Scattering in Hair. *ACM Trans. Graph.* 27, 3 (aug 2008), 1–10. https://doi.org/10.1145/1360612.1360631