

Real-time Level-of-detail Strand-based Hair Rendering

TAO HUANG*, University of California, Santa Barbara & LightSpeed Studios, USA

YANG ZHOU, University of California, Santa Barbara, USA

DAQI LIN, NVIDIA, USA

JUNQIU ZHU, University of California, Santa Barbara, USA

LING-QI YAN, University of California, Santa Barbara, USA

KUI WU, LightSpeed Studios, USA

We present a real-time strand-based hair rendering framework that ensures seamless transitions between different levels of detail (LoD) while maintaining a consistent appearance. Our LoD framework dynamically, adaptively, and independently replaces clusters of individual hair strands with *thick hairs* based on their projected screen widths. We introduce a Bidirectional Curve Scattering Distribution Function (BCSDF) model for the thick hairs, which accurately captures both single and multiple scattering within the hair cluster. Through tests on diverse hairstyles, various hair colors, and animations, our framework closely replicates the appearance of multiple-scattered full hair geometries at various viewing distances, achieving up to a 13× speedup.

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

Additional Key Words and Phrases: Hair rendering, level-of-detail, real-time rendering

1 INTRODUCTION

Hair contributes significantly to believable characters and immersive environments in films, video games, or virtual reality. However, an average human scalp has about 100K - 150K hair follicles. So, rendering and simulating realistic hair with complex geometries at such a scale poses significant challenges to computational efficiency for real-time applications. In the gaming industry, a common technique is to use hair cards [Jiang 2016], flat, textured planes that simulate the visual appearance of hair. While this technique provides computational efficiency, the inherent coarse approximation limits the realism achievable in physics simulations and rendering.

Strand-based hair simulation and rendering have gained prominence in recent production practices for their capacity to offer a more authentic portrayal of hair behavior and appearance [Epic Games 2021; Hsu et al. 2023; Huang et al. 2023; Tafuri 2019]. Given the vast number of hairs, a level-of-detail (LoD) strategy that simplifies the model at different viewing distances is essential to optimize computational resources for real-time applications. However, existing LoD solutions for strand-based hair struggle with smooth transitions in dynamics and appearance between LoD levels. One approach is to use hair cards when the hair is at a distance. However, this introduces a notable discontinuity in dynamics and appearance during the transition from strands to cards due to their fundamentally different representations. An alternative strategy reduces the number of hair strands while increasing each remaining strand’s thickness. Regrettably, this method leads to an excessively bold and dry hair appearance compared to the realistic brightness and hue, as the thick hair meant to represent a cluster of hair strands should have distinct optical properties from a single hair strand. This

distinction becomes particularly pronounced during multiple light bounces, exacerbating the visual disparity. Recent research proposes aggregated scattering models for a collection of furs [Zhu et al. 2022] and fibers [Zhu et al. 2023] that capture the aggregated appearance, incorporating the effect of multiple scattering. However, neither of their methods can be used in real-time hair rendering due to the slow neural network inference [Zhu et al. 2022] and the inaccurate handling of multiple scattering within an aggregated ply, which is exaggerated at the model level [Zhu et al. 2023], respectively.

This work presents a novel real-time technique for rendering strand-based hairs with seamless transitions between different LoD levels while ensuring consistent hair appearance. Our approach starts with a hair hierarchy construction method, where we build hair clusters at each LoD level and fit each cluster with a thick hair to better conform to the overall shape. To shade these thick hairs, we consider the sparsity of hairs in a cluster and reframe existing aggregation scattering models [Zhu et al. 2023, 2022] to derive a Bidirectional Curve Scattering Distribution Functions (BCSDF) model, capturing both single and multiple scattering interactions between individual hairs within the hair cluster. Our multiple scattering component modifies the dual scattering [Zinke et al. 2008] approximation by redistributing the lobe energies and introducing an extra light path series. We present a real-time strand-based hair rendering framework that dynamically selects the appropriate hair LoD level, along with an on-the-fly hair width calculation method to tightly cover represented fine hair strands. We implement our real-time rendering pipeline in modern GPU rasterization pipeline and conduct tests on various hairstyles with dynamics. In our experiments, our LoD framework, combined with our new BCSDF model, facilitates smooth transitions with a consistent appearance, resulting in neglectable computational overhead (<1%) for close-up views, a speedup of 2× for middle views, and up to 13× for far views.

2 RELATED WORK

Hair BCSDF model. Kajiya and Kay [1989] introduced the first physically-based analytic fiber shading model, considering the variation of reflectance in azimuthal directions and accounting for internal absorption and caustics. This model is later extended by Marschner et al. [2003] into the well-known Bidirectional Curve Scattering Distribution Functions (BCSDF) model for hair, which remains actively used in the games and film industries. The analysis involved ray paths within dielectric cylinders, and the scattering is split into R , TT , and TRT modes. These modes are represented as separable products of azimuthal and longitudinal functions, where R and T signify one reflection and transmission inside the fiber,

*Part of this work was done when Tao Huang was an intern at LightSpeed Studios.

respectively. This model has undergone improvements by various researchers. Notable enhancements include adding a non-physical diffuse term for accurate fitting of measured data [Zinke et al. 2009], decoupling into artist-friendly lobes [Sadeghi et al. 2010], adding a *TRRT* lobe for energy conservation [d’Eon et al. 2011], and adding lobe for high-order internal reflections [Chiang et al. 2016]. Additionally, it has been extended for elliptical hairs [Khungurn and Marschner 2017], animal hairs with interior medullas [Yan et al. 2015], and microfacet-based hair scattering model [Huang et al. 2022]. In recent work, Xia et al. [2023, 2020] propose a wave optics scattering model to capture the colorful glints of hair and fur.

Multiple scattering among hairs. Multiple scattering is essential for realistic hair rendering but is computationally expensive. Moon and Marschner [2006] use photon mapping to track indirect light bounces, storing incident radiance in a 3D grid of spherical harmonic (SH) coefficients [Moon et al. 2008]. KT et al. [2023] use MLP to learn the high-order scattered radiance between hairs. Zinke et al. [2008] introduce the dual scattering technique, dividing computations into global multiple scattering and local multiple scattering. Using a deep opacity map [Yuksel and Keyser 2008] to accelerate global scattering, they achieved real-time strand-based hair rendering. Our work also builds upon the dual scattering by introducing an additional local scattering component to efficiently approximate the complex scattering effects between aggregated hairs in real-time scenarios.

Level-of-detail construction. The level-of-detail (LoD) strategy is commonly employed in real-time applications to optimize performance by using models with varying complexities based on viewer distance [Karis et al. 2021]. Various simplification techniques have been developed to create LoD meshes at different complexity levels, including element removal [Garland and Heckbert 1997; Hoppe 1996; Luebke et al. 2002], re-triangulation [Cohen-Steiner et al. 2004; Li and Nan 2021], and primitives fitting [Bauchet and Lafarge 2020; Nan and Wonka 2017]. However, traditional LoD methods do not suit filament-like shapes such as hair due to their unique geometry. Weier et al. [2023] present an adaptive neural representation for correlation-aware LoD voxels designed for general purposes. In simulation [Bertails et al. 2003] and hair editing [Ward et al. 2003], a bottom-up LoD strategy was employed to group hair strands into clusters by their scalp locations. Their approach, however, introduces noticeable appearance differences between LoD levels due to applying a single hair appearance model for a hair cluster, ignoring light interaction within the cluster. We use a top-down strategy to construct the hair strand hierarchy, which is more precise.

Surface-based hair simplification – hair cards. Koh and Huang [2001] pioneer the use of a polygon strip to represent a cluster of hair strands with alpha maps for real-time applications, a technique still prevalent in the game industry [Jiang 2016]. However, creating hair cards for different LoD levels is time-consuming for artists and can cause discontinuities in dynamics and appearance during transitions. In contrast, the LoD solution proposed in this paper only requires a model with full hair strands, eliminating the need for additional manual authoring in LoD asset creation.

Strand-based hair simplification – aggregation model. Based on the hierarchical structure of yarn geometries, the aggregation model has recently been employed to depict the optical characteristics of a cluster of fibers in woven [Montazeri et al. 2020] and knitted structures [Montazeri et al. 2021]. Their model integrates normal mapping with a specialized BSDF that combines a specular component for specular reflection and transmission and a body component for multiple scattering within the fibers bundle. Cook et al. [2007] introduce a stochastic technique by randomly selecting a subset of the geometric elements and altering those elements statistically to preserve the overall appearance. Zhu et al. [2022] proposed an aggregation BCSDf model for a bunch of fibers using neural networks. In contrast, Zhu et al. [2023] derived an analytical single and multiple light scattering model for yarns based on dual-scattering theory, eliminating the need for data collection for parameter fitting.

Note that the analytical aggregated scattering model [Zhu et al. 2023] is based on a single fiber BCSDf with three lobes, *R*, *TT*, and *D* (instead of *TRT*). Our work builds heavily on this approach with two significant improvements, and our framework is generalizable to also support the Marschner hair BCSDf [Marschner et al. 2003] with *R*, *TT*, and *TRT* lobes. Firstly, we redistribute the lobe energies and introduce an extra light path series in the aggregation scattering model for more accurate light scattering approximation within the hair cluster. Secondly, our model is generalized to support an elliptical cross-section, creating a tighter fitting bound for hair clusters.

3 BACKGROUND

This section briefly overviews the single hair shading model, the dual scattering technique used for multiple scattering approximation, and the previous aggregated shading model that serves as the foundation for our proposed method.

The general rendering equation for outgoing radiance L_o towards a direction parameterized by longitudinal-azimuthal angles (θ, ϕ) is defined as:

$$L_o(\theta_o, \phi_o) = \int_{-\pi}^{\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} L_i(\theta_i, \phi_i) f(\theta_i, \phi_i, \theta_o, \phi_o) \cos^2(\theta_i) d\theta_i d\phi_i,$$

where L_i and L_o denote incoming radiance from direction (θ_i, ϕ_i) and outgoing radiance in direction (θ_o, ϕ_o) , respectively.

3.1 Single Hair Shading Model

As suggested by Zhu et al. [2023], the interaction between light and a single hair can be decomposed into reflection *R* and transmission *TT* lobes [Marschner et al. 2003], along with an additional diffuse lobe *D* to approximate multiple scattering inside the hair (see Fig. 1a). The single hair BCSDf, denoted as f , is defined as:

$$f^{\text{single}}(\theta_i, \phi_i, \theta_o, \phi_o) = \sum_{p \in \{R, TT, D\}} A_p M_p(\theta_h) N_p(\phi), \quad (1)$$

where p denotes different types of lobes, $\phi = \phi_o - \phi_i$ is the relative azimuthal angle, $\theta_h = (\theta_i + \theta_o)/2$ is the longitudinal half angle, and A_p is the attenuation function associated with each lobe. M_p and N_p are longitudinal and azimuthal scattering functions, respectively. In this paper, all results are using this BCSDf for a single fiber.

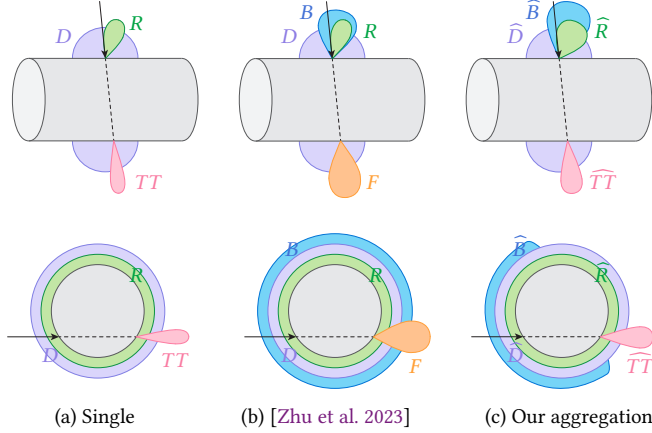


Fig. 1. Illustration of the single, aggregated [Zhu et al. 2023], and our BCSDf for longitudinal (top) and azimuthal (bottom) components. Our model replaces F lobe with \widehat{TT} lobe and distributes part of its contribution to \widehat{R} , \widehat{TT} , and \widehat{D} with larger variances. Additionally, our \widehat{B} lobe is designed to only account for the outgoing paths toward the back hemisphere.

3.2 Dual Scattering

Zinke et al. [2008] proposed dual scattering (DS) to approximate multiple scattering between millions of hair segments efficiently. The key idea is to split the multiple scattering into two additional lobes: global multiple scattering and local multiple scattering. The global scattering approximates the amount of light remaining after forward scattering through a number of hairs, while local scattering estimates the total contribution through backward scattering after multiple bounces in the neighborhood. Thus, two scattering lobes are added to the local shading (Eq. 1) as an approximation:

$$f_F^{ds}(\theta_i, \theta_o, \phi) = \frac{I_F(\phi)}{\cos^2(\theta_i)} d_F A_F \sum_{p \in \{R, TT, D\}} G(\bar{\sigma}_F^2, \theta_o + \theta_i) N_p^F, \quad (2)$$

$$f_B^{ds}(\theta_i, \theta_o, \phi) = \frac{I_B(\phi)}{\pi \cos^2(\theta_i)} d_B A_B G(\bar{\sigma}_B^2, \theta_o + \theta_i), \quad (3)$$

where $A_{F|B}$, $d_{F|B}$, $\bar{\sigma}_{F|B}$, and $I_{F|B}$ represent the forward/backward attenuations, scattering density constants, averaged spread variance, and binary backward hemisphere indicators, respectively. N_p^F is the averaged azimuthal lobe p within the front hemisphere. Forward scattering attenuation A_F can be computed by the sequential product of averaged forward attenuations that the light passes through, while $A_B = A_1 + A_3$ includes all light paths with single backward scattering A_1 and with three backward scattering A_3 . $G(\beta, \theta) = \frac{1}{\sqrt{2\pi\beta}} e^{-\theta^2/2\beta^2}$ is the traditional Gaussian distribution. We refer readers to the previous work [Zinke et al. 2008] for more details. However, dual scattering only considers R and TT and assumes infinite hair strands behind the shading point.

3.3 Aggregated Shading Model

Leveraging the dual scattering, Zhu et al. [2023] proposed an aggregated BCSDf model to approximate light scattering within a ply

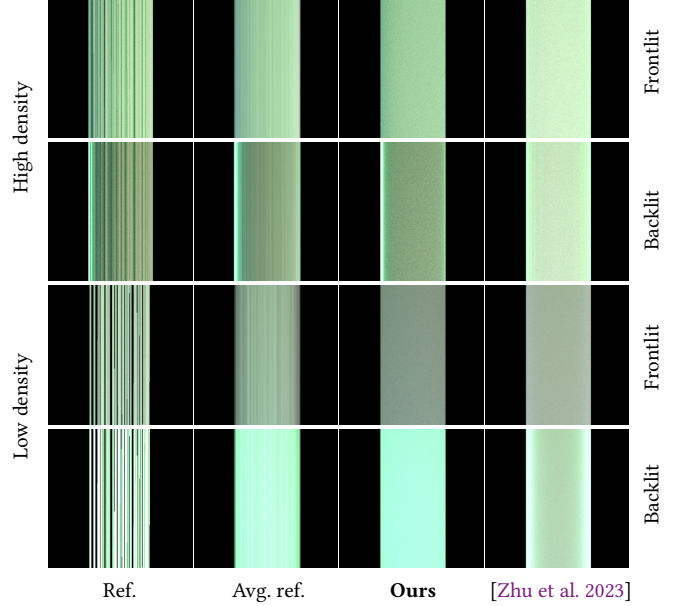


Fig. 2. Compared to prior aggregated model [Zhu et al. 2023], ours is closer to avg. ref. that averages 100 different ply instances (ref.) for both frontlit and backlit with high and low hair densities. Both avg. ref. and ref. use single hair BCSDf f^{single} .

consisting of n fibers as:

$$f^{\text{aggregated}}(\theta_i, \phi_i, \theta_o, \phi_o) = \sum_{p \in \{R, D, F, B\}} A_p M_p(\theta_h) N_p(\phi), \quad (4)$$

in which, in addition to reflection R and diffuse D lobes, they introduced forward scattering lobe F and backward scattering lobe B to capture multiple scattering components inside the aggregated ply (see Fig. 1b). The F lobe represents the distribution of remaining scattered energy that passes forward after multiple scattering inside the ply. The corresponding attenuation $A_F(\theta_d)$ is computed as:

$$A_F(\theta_d) = \prod_{i=1}^n a_F(\theta_d), \quad (5)$$

where a_F sums the corresponding outgoing radiance toward the front hemisphere, including contributions from R , TT , and D lobes of a single hair. However, as shown in Fig. 2 second column, their aggregated model does not match the reference well, especially under backlit conditions. Particularly, their F lobe represents the remaining energy after the ray passes through all the fibers in the light path. However, their R , B , and D lobes still exhibit a diffuse azimuthal distribution, which should be attenuated after multiple scattering. The double-counting leads to higher brightness. Additionally, in sparse fiber distribution, TT lobe exhibits a completely different azimuthal distribution compared to D and R lobes, and it cannot be represented by a single F lobe adequately.

4 OUR AGGREGATED SHADING MODEL

We propose an improved aggregated BCSDf model with three novel enhancements, including 1) modified lobe decomposition strategy,

2) shadowing-masking term $S(\phi, \theta_d)$, and 3) elliptical cross-section handling, to match the reference accurately. Our aggregated BCSDf is defined as:

$$f^{\text{ours}}(\theta_i, \phi_i, \theta_o, \phi_o) = S(\phi, \theta_d) \sum_{p \in \{\widehat{R}, \widehat{TT}, \widehat{D}, \widehat{B}\}} A_p M_p(\theta_h) N_p(\phi), \quad (6)$$

We consider our aggregated model to characterize lobes of the last intersected hair inside an aggregated geometry. Let n be the total number of hairs intersected by a ray that carries incident radiance. If the ray leaves from the back hemisphere of the aggregated hair, n always equals 1, meaning that \widehat{R} , \widehat{D} , and \widehat{TT} are exactly the same as their counterparts in the single hair BCSDf. Conversely, if the ray leaves from the front hemisphere, it must have traversed $n - 1$ hairs already. Following the concept of dual scattering, in our model, the original F lobe in Eq. 4 is distributed to \widehat{R} , \widehat{D} , and \widehat{TT} lobes, while \widehat{B} lobe only approximates outgoing radiance toward the back hemisphere from the shading point.

Modified \widehat{R} , \widehat{D} , and \widehat{TT} lobes. The \widehat{R} , \widehat{D} , and \widehat{TT} lobes in our model describe the scattering distribution of a hair for incident radiance that arrives at the shading point after being attenuated by $n - 1$ preceding hairs. Similar to global scattering, the aggregated attenuation of each lobe is the product of the attenuation per penetrated hair along the scattering path, expressed as:

$$A_p = a_F^{n-1} a_p, \quad p \in \{\widehat{R}, \widehat{D}, \widehat{TT}\}. \quad (7)$$

The longitudinal and azimuthal scattering functions of \widehat{R} , \widehat{D} , and \widehat{TT} lobes in Eq. 6 are defined as:

$$\begin{aligned} M_{\widehat{R}} &= G(\beta_{\widehat{R}}^M, \theta_h), & N_{\widehat{R}} &= \frac{1}{2\pi}, \\ M_{\widehat{TT}} &= G(\beta_{\widehat{TT}}^M, \theta_h), & N_{\widehat{TT}} &= G(\beta_{\widehat{TT}}^N, \phi - \pi), \\ M_{\widehat{D}} &= \frac{1}{\pi}, & N_{\widehat{D}} &= \frac{1}{2\pi}, \end{aligned} \quad (8)$$

where scattering distributions are modeled by Gaussian functions with wider variance β_p^M to account for the longitudinal spread due to front scattering. In particular, β_p^M and $\beta_{\widehat{TT}}^N$ are calculated by accumulating all variances along the light path as:

$$\begin{aligned} \beta_p^M &= (n-1)\sigma_f^2(\theta_d) + \beta_p^M, & p &\in \{\widehat{R}, \widehat{TT}\} \\ \beta_{\widehat{TT}}^N &= (n-1)\sigma_f^2(\theta_d) + \beta_{\widehat{TT}}^N. \end{aligned} \quad (9)$$

The spread variance $\sigma_f^2(\theta_d)$ is calculated by accumulating the forward azimuthal variances of n hairs. $\theta_d = \theta_i - \theta_o$ is the relative longitudinal angle.

Modified \widehat{B} lobe. In our approach, we employ the \widehat{B} lobe to approximate the contribution of multiple scattering near the shading point after the ray has reached the corresponding hair. It is essential to note that, following the assumption of dual scattering that splits multiple scattering into forward and backward components, the forward scattering is distributed among \widehat{R} , \widehat{D} , and \widehat{TT} lobes. The \widehat{B} lobe is specifically designed to account for the outgoing path toward the back hemisphere at the shading point, as illustrated in Fig. 1c.

To address the scenario that hair distribution is relatively sparse within the aggregate geometry, allowing light to easily traverse the

“first” layer of hairs and reach the hair behind (green path in Fig. 3a), we introduce an additional path series, denoted as A_{1+} , for the backscattering light, to supplement existing paths with one A_1 and three A_3 backward scattering events (denoted by B), and the paths with even number of forward scattering events (denoted by F). This series represents the scenario where the light hits the hair behind the currently shaded hair first, undergoes an even number of forward scattering events and one backward scattering event, and then hits the back side of the current hair. This path series can be described as $F^{i+1}BF^i$, where $i \geq 0$ denotes the number of hairs passed through behind the currently hit hair. The rationale behind considering the A_{1+} path series is that, when the hair density is relatively low, there is a higher probability that some hairs behind the currently hit hair get illuminated through gaps, significantly contributing to the overall appearance of the outer hair. The corresponding attenuation is computed as follows:

$$A_{1+} = d_{1+} a_B(\theta_d) \sum_{i=1}^n a_F^{2i-1}(\theta_d), \quad (10)$$

where d_{1+} accounts for the possibility of the contribution of such paths coming from the side and rear directions. In practice, we approximate it using a constant value as $d_{1+} = 0.6$.

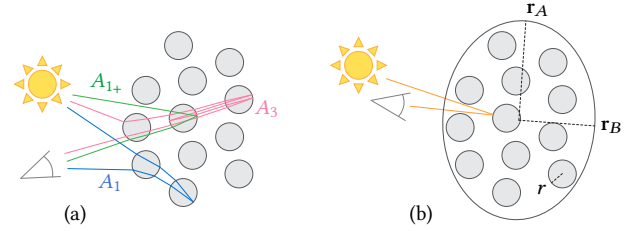


Fig. 3. Illustrations of light pass in aggregated cross-section: (a) the blue A_1 and pink A_3 paths are considered in the traditional local scattering with one and three backward scattering events by B lobe. The green path illustrates our added A_{1+} paths, which encounter two deflections before returning to the shading hair. (b) when the incident and outgoing directions are close, the observed hair along the incident direction may not be occluded by others along the outgoing direction. All hairs with radius r are contained by an elliptical cross-section, with major axis r_A and minor axis r_B .

The attenuation $A_{\widehat{B}}$, longitudinal $M_{\widehat{B}}$ and azimuthal $N_{\widehat{B}}$ distributions are as follows:

$$A_{\widehat{B}} = A_{1+} + A_1 + A_3, \quad M_{\widehat{B}} = G(\beta_{\widehat{B}}^M, \theta_h), \quad N_{\widehat{B}} = \frac{I_B(\phi)}{\pi}, \quad (11)$$

where the longitudinal variance $\beta_{\widehat{B}}^M$ is the averaged accumulated variance along each possible path, weighted by its attenuation. Since the R and D lobes of a single hair exhibit a diffuse distribution in the azimuthal direction, our aggregated $N_{\widehat{B}}$ is modeled as a uniform distribution over the back hemisphere.

Estimation of hair number n . In order to represent a collection of hairs using a thick cylinder with an elliptical cross-section, it is important to estimate the number of hairs a ray intersects inside the aggregated cylinder for computing attenuation and scattering

distribution. We define the unitless hair density ρ as the ratio of the area occupied by the hairs to the cross-sectional area of the aggregated cylinder, given by $\rho = n_{\text{total}}r^2/(\mathbf{r}_A|\mathbf{r}_B|)$, where n_{total} is the number of hairs inside, r is the single hair radius, and \mathbf{r}_A and \mathbf{r}_B are the major and minor axes of the cross-section. Given the ray travel distance l , the number of intersecting hairs n can be calculated as $n = \rho l \sqrt{\frac{n_{\text{total}}}{\pi|\mathbf{r}_A||\mathbf{r}_B|}}$, where the last term estimates number of intersecting hairs per unit distance.

Shadowing-masking $S(\phi, \theta_d)$. For a cluster of hair strands, even if ω_i and ω_o fall within the same hemisphere, it does not guarantee that the hair seen from the view direction can also be seen from the light direction. In other words, when the relative azimuthal angle $\phi \in (-\frac{\pi}{2}, \frac{\pi}{2})$, self-shadowing situations can occur within a thick aggregated geometry.

We propose a shadowing-masking term to compensate for self-shadowing for $\phi \in (-\frac{\pi}{2}, \frac{\pi}{2})$ by categorizing shadowing at the shading point into three categories: shadowing by a single hair, shadowing by multiple hairs, and no shadow. We approximate the possibility of the shadow ray from a shading point needing to pass through a single hair to reach the light source as:

$$P_s = (1 - \rho)(1 - \cos(\phi)), \quad (12)$$

where $\cos(\phi)$ is used to approximate the difference between the incident and outgoing directions in the azimuthal plane. As shown in Fig. 3b, hairs located further from the incident direction are more likely to be occluded by other strands along the light direction. The $1 - \rho$ term accounts for the fact that the shadow-masking effect becomes less prominent with denser hairs, as a thick fiber better approximates the cluster of hairs.

The shading point can also be occluded by multiple hairs along the shadow ray. We approximate the probability as

$$P_m = (1 - \rho) c_m, \quad (13)$$

where c_m is a constant value empirically determined ($c_m = 0.2$ for all our experiments). With P_s and P_m , we can determine the probability of the shading point reached directly by light with no-shadow as:

$$P_n = 1 - P_m - P_s. \quad (14)$$

Thus, the shadowing-masking $S(\phi, \theta_d)$ term is formulated as:

$$S(\phi, \theta_d) = P_s(\phi)a_F(\theta_d) + P_n\mathbb{1}, \quad (15)$$

where $\mathbb{1}$ represents a three-dimensional vector of ones. Based on the dual scattering assumption, the attenuation needs to be multiplied by a_F , as the light ray must undergo forward scattering once to reach the actual shading hair. Note that no lighting calculation is performed for P_m , given the assumption that its contribution to lighting is negligible.

5 REAL-TIME RENDERING PIPELINE

This section outlines the construction process of the hair hierarchy and describes how we utilize it in rendering.

5.1 Initialization

Given a strand-based hair model with tens of thousands of *single hairs*, we build the hair hierarchy in four stages: 1) selecting a number of guide hairs for simulation, 2) clustering hair strands into

n_L levels, 3) fitting the ellipses for thick hair cross-sections, and 4) computing interpolation weights. We denote the levels in the hierarchy by L_0 (coarsest) through L_{n_L-1} (finest).

1. Guide hair selection. Initially, a Catmull–Rom spline is fitted with n_c control points for each hair strand. Following [Wang et al. 2009], we use k-means clustering to create a user-defined number of hair clusters. The distance metric between two single hairs i and j is defined as the sum of distances between corresponding control points along the two single hairs, expressed as $D(i, j) = \sum_k^{n_c} |\mathbf{p}^{i,k} - \mathbf{p}^{j,k}|^2$, where $\mathbf{p}^{i,k}$ denotes the position of the k -th control point along the hair i . Within each cluster, the hair with the smallest average distance to all other hairs is selected as the guide hair, which serves as the basis for the physical-based simulation and is used to animate each single hair via linear skinning [Somasundaram 2015] in runtime. Note that the cluster used for guide hair selection in this stage would not be discarded afterward.

2. Hair strand clustering. After identifying guide hairs, each single hair locates three guide hairs that are closest to it. Single hairs sharing the same set of guide hairs are clustered together to form the coarsest level L_0 . The subsequent levels, from L_1 to L_{n_L-1} , are directly clustered from L_0 . While we do not maintain strict bounding relations between consecutive levels, we ensure that the cross-sectional area of the thick hairs at each level is approximately four times larger than that of the previous level.

3. Cross-section fitting. To construct a thick elliptical hair from a cluster of single hairs, we first create n_c sets of control points where each set \mathcal{P}^k contains all single hairs’ k -th control points. For each set \mathcal{P}^k , we compute the average position of all control points $\bar{\mathbf{p}}_{cs}^k$ and normalized average tangent $\bar{\mathbf{t}}^k$ to define the center of the ellipse. Principal Component Analysis (PCA) is then employed to fit an ellipsoid for \mathcal{P}^k , after which we discard the axis closest to $\bar{\mathbf{t}}^k$. Finally, we identify four *corner control points* $\mathbf{p}_{0,1,2,3}^k$ from \mathcal{P}^k that are closest to the four endpoints of the remaining two axes.

4. Interpolation weight computing. In hair skinning, three guide hairs form a triangular prism for linear skinning. We compute and record interpolation weight for every single hair control point, as well as for every thick hair cross-section center $\bar{\mathbf{p}}_{cs}^k$ and corner control points $\mathbf{p}_{0,1,2,3}^k$.

5. LoD level initialization. We compute the initial LoD levels for the starting view in rendering. Using a user-specified screen-space hair width threshold ϵ_w , we determine the LoD level for each cluster within L_0 as follows: we examine the LoD level from coarsest to the finest until we find the first level where the maximum thick hair width in the cluster is less than ϵ_w . At runtime, we adjust the LoD levels starting from the level used in the previous frame, making this process much more efficient. Note that the LoD selection is performed on a per-segment basis.

5.2 Runtime

With guide hairs dynamically simulated at runtime, our rendering framework uses the simulated control points of the guide hairs and the precomputed interpolation weights to determine the positions

and shapes of the hairs at desired LoD levels (we update quantities like \vec{p}_{cs}^k and $p_{0,1,2,3}^k$ by interpolation using these positions and weights). We dynamically compute the thick hair widths to determine the proper LoD levels and calculate density and lobes in our aggregated BCSDf.

Hair width calculation. To ensure that thick hairs selected by the width-based LoD metric adequately cover the corresponding single hairs, we dynamically compute the hair widths as follows. First, the four corner control points $p_{0,1,2,3}^k$ are projected to the plane defined by \vec{p}_{cs}^k 's position and its tangent. Then, we measure the largest screen-space distances (on the xy plane) and the largest depth differences (along the z axis) between these projected points as l_w and l_t as hair width and thickness, respectively. The corresponding density and estimated hair number are computed as $\rho = n_{total}r^2/(l_w l_t)$ for shadow-masking in Eq. 15 and $n = \rho l_w \sqrt{\frac{n_{total}}{\pi l_w l_t}}$ for all lobes in Eq. 6.

Dynamic LoD selection. During runtime, we select the LoD level for each cluster in L^0 . The goal of our LoD selection is to ensure that all thick hairs rendered have widths that are just below the screen-space hair width threshold ϵ_w . Checking through all LoD levels can be expensive at runtime. Therefore, we propose an approximately correct, deferred LoD selection that reuses information between frames. For each cluster in every frame, we render its thick hairs at the current LoD level (the LoD level for the first frame is determined during initialization) and record the maximum width, denoted as w_{cur} . We also calculate the maximum width at one coarser level, denoted as w_{cur-} . Using these two widths, we determine the LoD level L for the next frame as:

$$L = \begin{cases} L + 1, & \text{if } w_{cur} > \epsilon_w, \\ L, & \text{if } w_{cur} \leq \epsilon_w < w_{cur-}, \\ L - 1, & \text{if } w_{cur-} \leq \epsilon_w. \end{cases} \quad (16)$$

Runtime pipeline. With the geometry generated for the given LoD level, we proceed with the conventional rendering pipeline, which includes three passes:

1. *Shadow pass.* Head and hair are rendered to generate a shadow map and deep opacity map [Zinke et al. 2008], respectively.

2. *Shading pass.* The width for each thick hair is calculated dynamically in the vertex shader. The maximum hair width for each cluster is recorded in a buffer using `atomicMax`. The tessellation shader subdivides hair strands based on their distance from the camera, and the geometry shader extends them into camera-facing strips. The BCSDf described in Sec. 4 is employed in the fragment shader for shading.

3. *Compute pass.* The maximum hair width in each cluster at one coarser level is computed to determine the LoD for the next frame using the selection strategy (Eq. 16). Note that this compute pass only invokes the vertex shader.

Note that our runtime pipeline does not explicitly handle thick hairs overlapping, as hairs are rendered as camera-oriented strips for both the shading and deep opacity map passes.

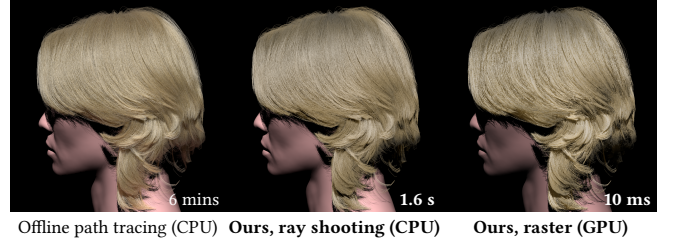


Fig. 4. Our approach (offline ray shooting and real-time OpenGL rasterizer) achieves close results to the path tracing reference, while our real-time implementation is 36000 \times faster for close view.

6 RESULTS

We conduct all experiments on a system with an AMD Ryzen ThreadRipper 3970X 32-core CPU, 256 GB of memory, and an NVIDIA RTX 3090 GPU. We implement our method (aggregated BCSDf + LoD strategy) on both a CPU renderer and a GPU renderer (Fig. 4). In contrast to employing full hair geometry in path tracing (PT) and dual scattering (DS) [Zinke et al. 2008], we utilize reduced hair geometry achieved through our hair LoD solution. Specifically, we group individual hair strands into 256 clusters and fit each cluster with thick hairs across 5-6 LoD levels, depending on the total number of hair strands. Each thick hair is represented by a B-Spline with 16 control points. For sufficient granularity, each thick hair segment is assigned an individual LoD level.

To validate the accuracy of our proposed method, we use a CPU render implemented in Embree [Wald et al. 2014] with 20 samples per pixel (spp). Similar to [Zinke et al. 2008], we use ray shooting to find all intersecting hairs along the shadow path to compute the transmittance for dual scattering. The CPU reference is generated by path tracing the full hair geometry with up to 70 ray bounces and 1024 spp. The hair BCSDf parameters are chosen generally consistent with the values provided by Marschner et al. [2003]. In particular, we use $M_R = 0.08 - 0.15$, $M_{TT} = 0.12 - 0.3$, and $N_{TT} = 0.2 - 0.4$. We further adjust attenuation to match the real-life hair color, as $A_R = (0.03, 0.03, 0.03) - (0.25, 0.25, 0.25)$, $A_{TT} = (0.25, 0.17, 0.07) - (0.38, 0.25, 0.18)$, $A_D = (0.2, 0.11, 0.1) - (0.40, 0.21, 0.08)$.

To further evaluate our real-time performance and rendering quality, we implement our framework and counterpart full hair geometry and hair cards with DS using OpenGL 4, in which the shadow computation in all methods uses deep opacity maps.

We use a LoD threshold of $\epsilon_w = 2$ pixels for all our experiments. The output image resolution for all results is 1024 \times 1024. The initialization of our method takes from 5 to 10 minutes, depending on the number of hair strands, with the majority of the time spent on hair clustering. To assess the impact of our LoD solution on rendering accuracy, we present results from near, middle, and far views with 60%, 20%, and 2% screen occupancy ratios, respectively. Note that our ray tracer produces a slightly different estimation of hair screen width than our rasterizer, which causes the hair count in the two systems to be different at the same view.

Hairstyles. In Fig. 5, we evaluate our approach on Four hairstyles, ranging from straight to curly. Two of these hairstyles are sourced from Unreal Metahumans [Epic Games 2021], and two are captured

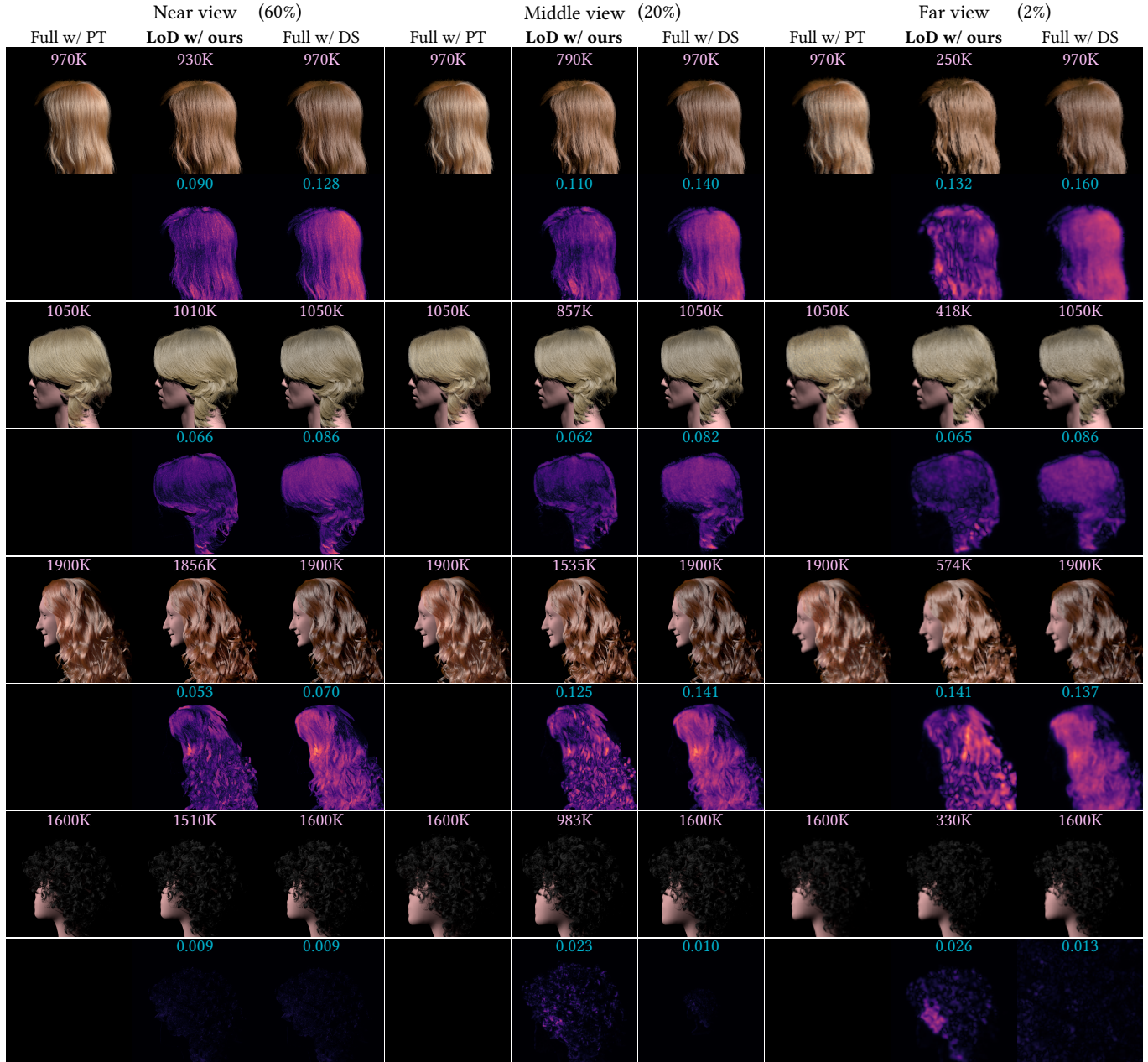


Fig. 5. Four hairstyles, ranging from straight to curly hair, under near, middle, and far views. Compared to path tracing with full hair geometry (full w/ PT), our approach (LoD w/ ours) utilizing reduced hair geometry through level-of-detail techniques can outperform existing dual scattering employing full geometry (full w/ DS) in both memory size and appearance. The FLIP error of ours increases for the far view due to the geometry simplification. Pink and cyan denote the number of segments and FLIP error of screen-space bounding box of hairs.

from the real-world examples [Derouet-Jourdan et al. 2013; Shen et al. 2023]. For the four hairstyles, we use different hair colors: brown, blonde, red, and black. Compared to using full hair geometry with DS, our method consistently delivers higher accuracy with lower FLIP errors for all close views and middle and far views across the first three hairstyles. The additional light path series A_{1+} introduced by our method significantly enhances shading accuracy.

However, as the viewing distance increases, the error of our method increases slightly due to geometric simplification.

Animated hairs. As shown in Fig. 6, our method maintains its rendering accuracy with hair animation. We deployed a strand-based hair simulator [Hsu et al. 2023] to simulate the 256 guide hairs of two hairstyles and derived all hairs via linear skinning.

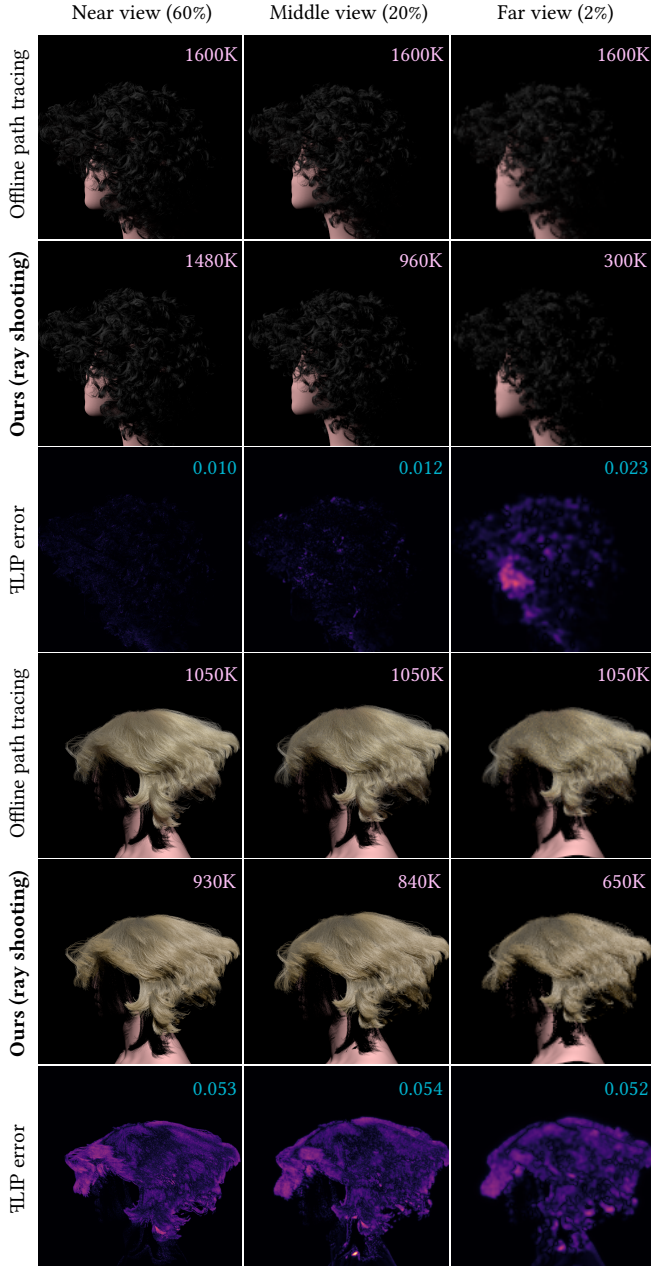


Fig. 6. Animated hairs from two simulation sequences exhibit varying shapes. *Pink* and *cyan* denote the number of segments and FLIP error of screen-space bounding box of hairs.

Please see the supplementary video for the complete simulation sequence, where our method allows a smooth transition between different LoD levels in real-time. Note that the hair LoD level is determined based on its screen space width, which means that in extreme cases of intense hair movement, the finest LoD level may be required. However, in our experiments, even during frames with the most intense motion, our method achieves a significant level of

simplification while maintaining nearly identical visual quality to that of the full geometry.

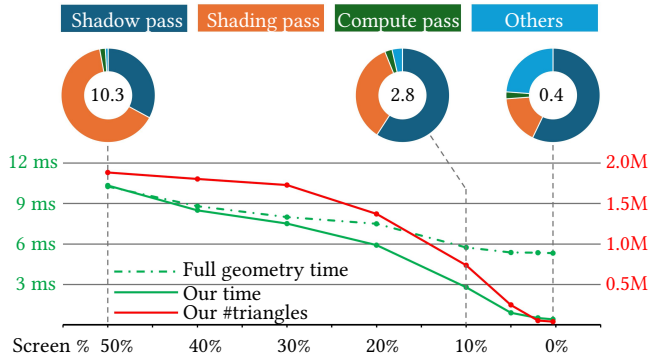


Fig. 7. A performance visualization of the scene in Fig. 8. As the screen occupation percentage decreases from 50% to 0.3%, our GPU rasterizer progressively outperforms rasterizing full hair geometry, achieving a slight overhead of $0.99\times$ (10.3ms vs. 10.2ms) to a significant improvement of $13\times$ (0.4ms vs. 5.3ms). The pie charts provide a breakdown of timings at three screen percentages of hairs.

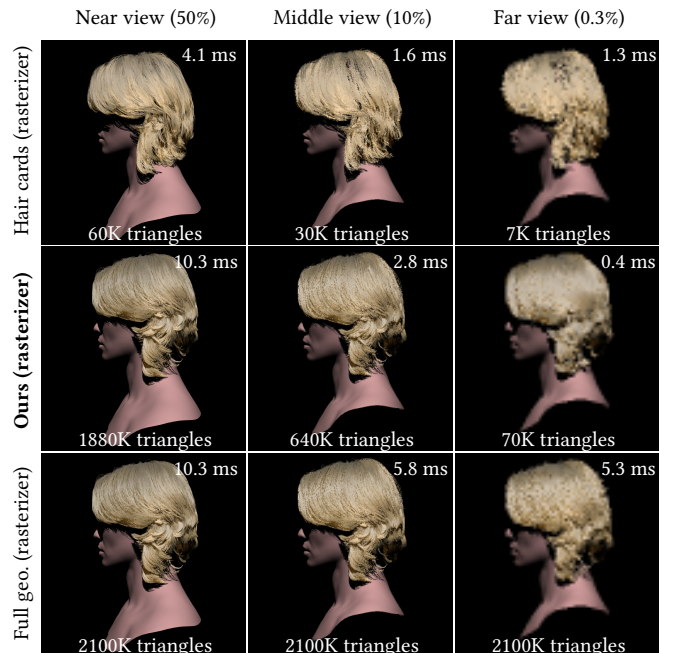


Fig. 8. Full hair strand geometry, our strand-based LoD model, and hair card model at near, middle, and far views. We mark the performance and the number of triangles for hair strips in the image.

Real-time performance. Fig. 7 illustrates the real-time performance of our GPU rasterizer pipeline. The most demanding scenario occurs with a close view (50% screen occupation ratio), where only a limited number of single hairs can be represented by thick hairs. In this case, our approach needs to render a similar number of hair segments

as the full hair geometry, along with additional compute passes for thick hair width computation. Nonetheless, our approach exhibits only a slight overhead (<1%). As the camera distance increases, hair strands gradually transition to thicker hairs, reducing the number of segments required for rendering. This leads to a noticeable speedup of 2× for the middle view (10% of screen) and up to 13× for the far view (0.3% of screen). The breakdown of single frame time indicates that the percentage of time taken by the shading pass decreases with the camera distance due to the reduction in hair geometry and fewer pixels required to shade. Additionally, our pipeline requires a constant time of about 0.1 ms for hair data organization based on the new LoD level for the next frame rendering.

Comparison to hair cards. Fig. 8 compares our strand-based model with a hair cards model. The hair cards model has 3 LoD levels, featuring 600K, 300K, and 70K triangles, respectively. We manually configure two distances for the hair cards to do the LoD switching and render them using our real-time GPU rasterizer. Despite requiring more triangles in our LoD structure, our approach facilitates a seamless transition between LoD levels and supports strand-based hair simulation. In contrast, the hair card textures at different LoD levels often exhibit inconsistencies, leading to noticeable popping artifacts during the LoD transition.

Note that we render haircards using an order-independent rasterization, meaning shading only the closest fragment without using alpha blending, for performance purposes. A threshold is set for the coverage texture, and fragments are discarded when their coverage falls below this threshold. Notably, in far-view scenarios, although hair cards have fewer triangles, they necessitate reading the coverage texture to compute accurate shadows. This requirement leads to higher computational costs compared to our method.

Additionally, we measured the smoothness of our transitions by comparing the similarity of adjacent frames using PSNR. As shown in Fig. 9, our method does not experience the drastic changes that occur with hair cards during LoD transitions. Furthermore, due to the appropriate LOD level, the flickering in our real-time rendering is less pronounced than with full geometry. Our PSNR remains consistently at the highest level among the three methods. For a more detailed comparison of LoD transitions, please refer to the supplemental video.

Comparison to [Zhu et al. 2023]. We utilize CPU path tracing to render 100 ply instances with diverse fiber geometries and average the rendering outcomes. Additionally, we illustrate the scattering intensity at the azimuthal and longitudinal planes in Fig. 10. The aggregated model proposed by Zhu et al. [2023] does not align well with the reference, particularly under backlit conditions, whereas our model provides a more accurate approximation.

Ablation study on A_{1+} . To assess the influence of A_{1+} path series, we conduct an ablation study involving the averaging of rendering results from 100 different fiber instances. Each instance is rendered with single BCSDf with full single fiber geometries using CPU path tracing. The fiber cluster forms an elliptical cross-section with a major axis to minor axis ratio of 4 : 1. Fig. 11 demonstrate that the A_{1+} term effectively compensates for dual scattering, resulting in close alignment with the average reference (avg. ref.).

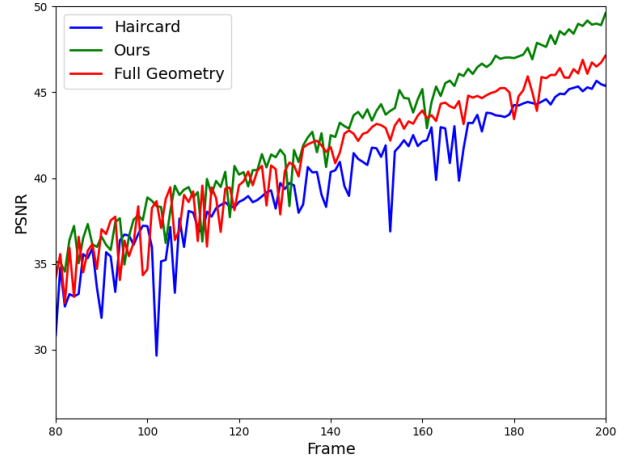


Fig. 9. Evaluation of transition smoothness. We utilize PSNR to measure the similarity of adjacent frames. Our method (Green) constantly has the highest score compared to full geometry (Red) and hair card (Blue) while avoiding the drastic changes in LoD switching.

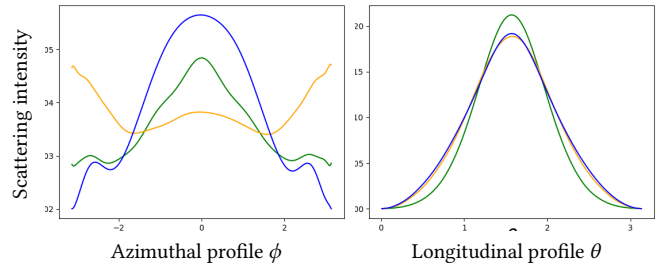


Fig. 10. The average scattering intensity (green) across 100 ply instances with distinct fiber geometries. Our method (blue) offers a more accurate approximation than that of [Zhu et al. 2023] (orange), particularly in the azimuthal profile.

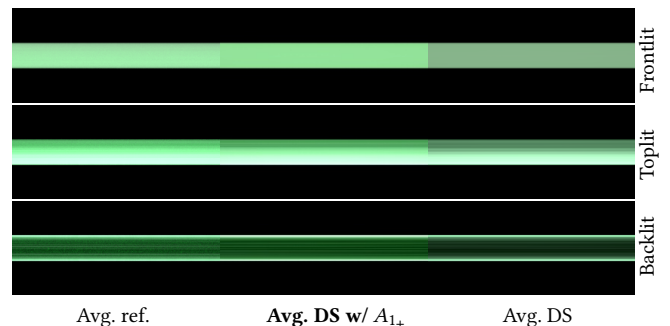


Fig. 11. Given a cluster of fibers forming an elliptical cross-section with a major axis to minor axis ratio of 4 : 1, A_{1+} term can compensate for dual scattering to closely match the average reference (avg. ref.). The major axis is pointing to the camera. Avg. indicates averaging rendering results from 100 different fiber instances.

The shadowing-masking term and aggregated BCSDf. To evaluate the impact of the shadowing-masking term in our aggregated BCSDf, we render a full hair geometry with single hair BCSDf

f^{single} as a reference and render a simplified hair geometry with our aggregated BCSDF f^{our} with and without the shadowing-masking term $S(\phi, \theta_d)$. In the simplified hair geometry, each thick hair contains around 256 single hairs. As shown in Fig. 12, omitting the shadowing-masking term results in a brighter output compared to the reference, as the occlusion inside the thick hair is ignored. Further, applying the single hair BCSDF f^{single} on the simplified hair geometry results in significant appearance bias, as expected. This bias occurs because using a single BCSDF for a cluster of hairs tends to produce brighter results, failing to account for the actual number of hairs traversed by a ray.

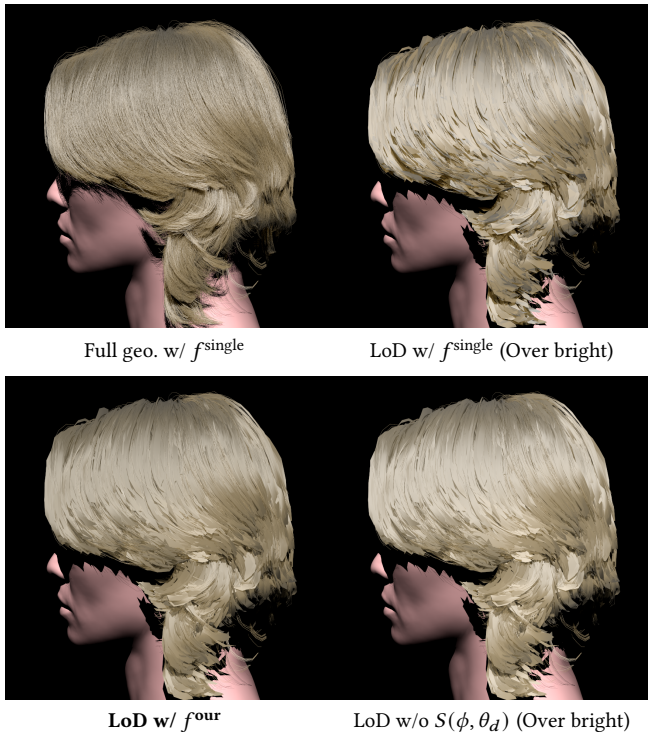


Fig. 12. Compared to rendering full hair geometry with single BCSDF f^{single} , using thick hair to reduce the hair size need our aggregated BCSDF f^{our} combined with shadow masking term $S(\phi, \theta_d)$.

Abalation study on various light directions. Fig. 16 demonstrates that our method can produce results closely resembling those obtained from offline path tracing across various light directions, including 0° (frontlit), 45° , 90° , and 180° (backlit), and from different viewing distances.

Extending to Marschner model with TRT term. While we are using the BCSDF model [Zhu et al. 2023] with R , TT , and D (instead of TRT) lobes, our framework is generalizable to support the Marschner hair BCSDF [Marschner et al. 2003] with R , TT , and TRT lobes. Fig. 17a shows an evaluation of our method using Marschner et al. [2003] with the full geometry in offline path tracing. Our method consistently and effectively reduces the number of hairs while maintaining a high level of appearance accuracy. Additionally, we demonstrate

that the BCSDF model [Zhu et al. 2023] without TRT is still capable of supporting light-colored hair, as shown in Fig. 17b.

Memory cost and initialization time. In addition to the original full-size hair strand data, we store the positions of the centerline control points and the indices of four guiding points for thickness calculation. This results in extra memory consumption of approximately three-quarters of the original model size (78% for the four models used in the paper). Our initialization time ranges from 6 to 12 minutes for hair models with strands ranging from 60,000 to 120,000. This process is performed only once, and the results can be stored as configuration.

Knit patches. Our aggregated BCSDF can also be used for fabric rendering. Our method handles twisted fibers by rotating the local frame at each shading point. We evaluate the efficacy of our aggregated BCSDF using two knit patches. Fig. 13 showcases the comparison. The reference images, denoted as ref., are generated using fiber-level geometries (9.7M and 8.9M segments for each patch, respectively) with single fiber BCSDF, while other methods render ply-level geometries with significantly fewer segments (130K and 49K segments for each patch, respectively). Our aggregate BCSDF closely resembles the reference appearance. In comparison, the absence of the shadowing-masking term contributes to the observable increase in error. Using single fiber BCSDF results in significantly brighter results. The prior aggregated BCSDF [Zhu et al. 2023] also results in increased brightness by double-counting R and D lobes.

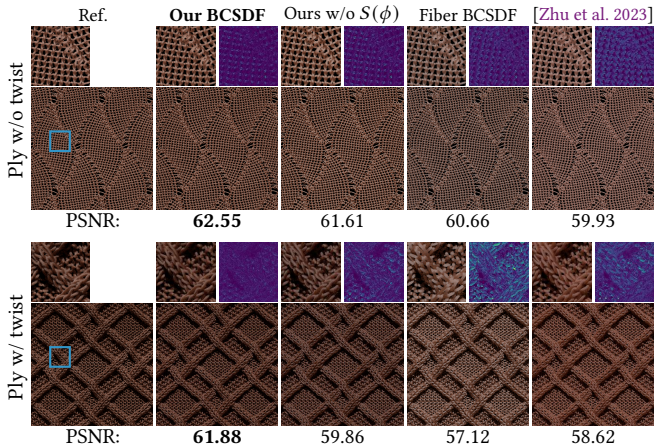


Fig. 13. Knit patches: Ref. is obtained using one fiber-level geometry instance with path tracing and single BCSDF, while others render a collection of fibers as a ply with single BCSDF and the aggregated BCSDF [Zhu et al. 2023]. All results are generated using CPU path tracing. Our approach produces the closest appearance to GT (highest PSNR). Notably, the absence of the shadowing-masking term leads to a noticeable brightness mismatch. Magnified insets (highlighted in blue) and corresponding error maps are provided.

7 CONCLUSION

We have presented a Level-of-detail framework for real-time strand-based hair rendering. Our approach incorporates a comprehensive

BCSDF that models both single and multiple scattering interactions among individual hairs within a hair cluster. Additionally, we have outlined a method for constructing a hierarchical representation of hair and a strategy for selecting appropriate levels of detail and determining the thick hair width on the fly, minimizing geometry data for efficient real-time rendering without perceptible loss in appearance. Furthermore, we have demonstrated that our proposed system can be implemented efficiently on the GPU to render various hairstyles in real time.



Fig. 14. Our method inherits limitations from dual scattering, particularly in accurately approximating the appearance of curly, light-colored hairs when illuminated from a side light direction.

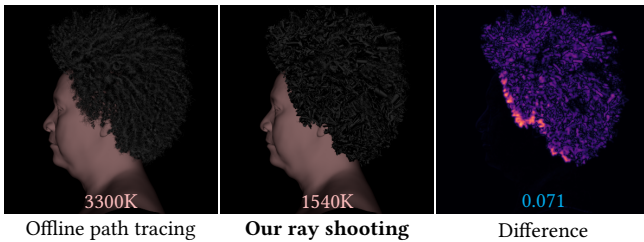


Fig. 15. For hairstyles with highly complex geometric structures, when it comes to far view and many strands of hair are replaced by a few thick fibers, a single thick fiber is insufficient to capture all the geometric properties of these hairs, leading to some inconsistency in the results.

Limitations. Our method, aligning with the concept of dual scattering, shares its limitations. In particular, the local scattering in dual scattering theory assumes that the geometric information and reflection model of the fibers surrounding the shading point is essentially consistent, which is not well-suited for hairs with various textured albedo and certain hairstyles, especially curly styles with light hair color, which would introduce biases in the results, as shown in Fig. 14. For hairstyles with highly complex geometric structures, such as the afro shown in Fig. 15, it is challenging to accurately maintain all the geometry details of a hair cluster using a single thick fiber. Also, our pipeline cannot preserve glints perfectly, as our LoD selection criteria are based on the assumption that the hair width is around 2 pixels on the screen to reduce hair geometries, while glints that appear usually need a sub-pixel (<0.5

pixels) details. Additionally, our current dynamic LoD selection ensures that rendered thick hairs remain within a certain width threshold (2 pixels), thereby minimizing the likelihood of one "thick hair" being seen through another. However, if the width threshold is increased, order-independent transparency (OIT) would be necessary to achieve correct rendering results, albeit at the cost of reduced performance. Last, we recognize the disparity between our research prototype and established commercial engines like Unreal Engine [Epic Games 2021]. While our method aligns with existing techniques utilizing spherical radial basis functions for environment lighting [Ren et al. 2010], it would be interesting to implement our LoD solution with aggregated BCSDF in real production and support environment lighting.

REFERENCES

- Jean-Philippe Bauchet and Florent Lafarge. 2020. Kinetic Shape Reconstruction. *ACM Trans. Graph.* 39, 5, Article 156 (jun 2020), 14 pages.
- F. Bertails, T.-Y. Kim, M.-P. Cani, and U. Neumann. 2003. Adaptive Wisp Tree: A Multiresolution Control Structure for Simulating Dynamic Clustering in Hair Motion. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) (SCA '03). Eurographics Association, Goslar, DEU, 207–213.
- Matt Jen-Yuan Chiang, Benedikt Bitterli, Chuck Tappan, and Brent Burley. 2016. A Practical and Controllable Hair and Fur Model for Production Path Tracing. *Comput. Graph. Forum* 35, 2 (2016), 275–283.
- David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. 2004. Variational Shape Approximation. *ACM Trans. Graph.* 23, 3 (aug 2004), 905–914.
- Robert L. Cook, John Halstead, Maxwell Planck, and David Ryu. 2007. Stochastic simplification of aggregate detail. *ACM Trans. Graph.* 26, 3 (jul 2007), 79–es.
- Eugene d'Eon, Guillaume François, Martin Hill, Joe Letteri, and Jean-Marie Aubry. 2011. An Energy-Conserving Hair Reflectance Model. *Comput. Graph. Forum* 30, 4 (2011), 1181–1187.
- Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, Gilles Daviet, and Joëlle Thollot. 2013. Inverse Dynamic Hair Modeling with Frictional Contact. *ACM Trans. Graph.* 32, 6, Article 159 (Nov. 2013), 10 pages.
- Epic Games. 2021. Unreal Engine. <https://www.unrealengine.com>
- Michael Garland and Paul S. Heckbert. 1997. Surface Simplification Using Quadric Error Metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., USA, 209–216.
- Hugues Hoppe. 1996. Progressive Meshes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. Association for Computing Machinery, New York, NY, USA, 99–108.
- Jerry Hsu, Tongtong Wang, Zherong Pan, Xifeng Gao, Cem Yuksel, and Kui Wu. 2023. Sag-Free Initialization for Strand-Based Hybrid Hair Simulation. *ACM Trans. Graph.* 42, 4, Article 74 (jul 2023), 14 pages.
- Li Huang, Fan Yang, Chendi Wei, Yu Ju (Edwin) Chen, Chun Yuan, and Ming Gao. 2023. Towards Realtime: A Hybrid Physics-Based Method for Hair Animation on GPU. *Proc. ACM Comput. Graph. Interact. Tech.* 6, 3, Article 43 (aug 2023), 18 pages.
- W. Huang, M. B. Hullin, and J. Hanika. 2022. A Microfacet-based Hair Scattering Model. *Computer Graphics Forum* 41, 4 (2022), 79–91.
- Yibing Jiang. 2016. The process of creating volumetric-based materials in uncharted 4. *Siggraph Courses: Advances in Real-Time Rendering in Games, Anaheim, CA 0*, 0 (2016), 0.
- J. T. Kajiya and T. L. Kay. 1989. Rendering Fur with Three Dimensional Textures. *SIGGRAPH Comput. Graph.* 23, 3 (jul 1989), 271–280.
- Brian Karis, Rune Stubbe, and Graham Wihlidal. 2021. A Deep Dive into Nanite Virtualized Geometry. *ACM SIGGRAPH 2021 Talks* 0, 0 (2021), 0.
- Pramook Khungurn and Steve Marschner. 2017. Azimuthal Scattering from Elliptical Hair Fibers. *ACM Trans. Graph.* 36, 2, Article 13 (apr 2017), 23 pages.
- Chuan Koon Koh and Zhiyong Huang. 2001. A Simple Physics Model to Animate Human Hair Modeled in 2D Strips in Real Time. In *Proceedings of the Eurographic Workshop on Computer Animation and Simulation* (Manchester, UK). Springer-Verlag, Berlin, Heidelberg, 127–138.
- Aakash KT, Adrian Jarabo, Carlos Aliaga, Matt Jen-Yuan Chiang, Olivier Maury, Christophe Hery, P. J. Narayanan, and Giljoon Nam. 2023. Accelerating Hair Rendering by Learning High-Order Scattered Radiance. *Computer Graphics Forum* 42, 4 (2023), e14895.
- Minglei Li and Liangliang Nan. 2021. Feature-preserving 3D mesh simplification for urban buildings. *ISPRS Journal of Photogrammetry and Remote Sensing* 173 (March 2021), 135–150.

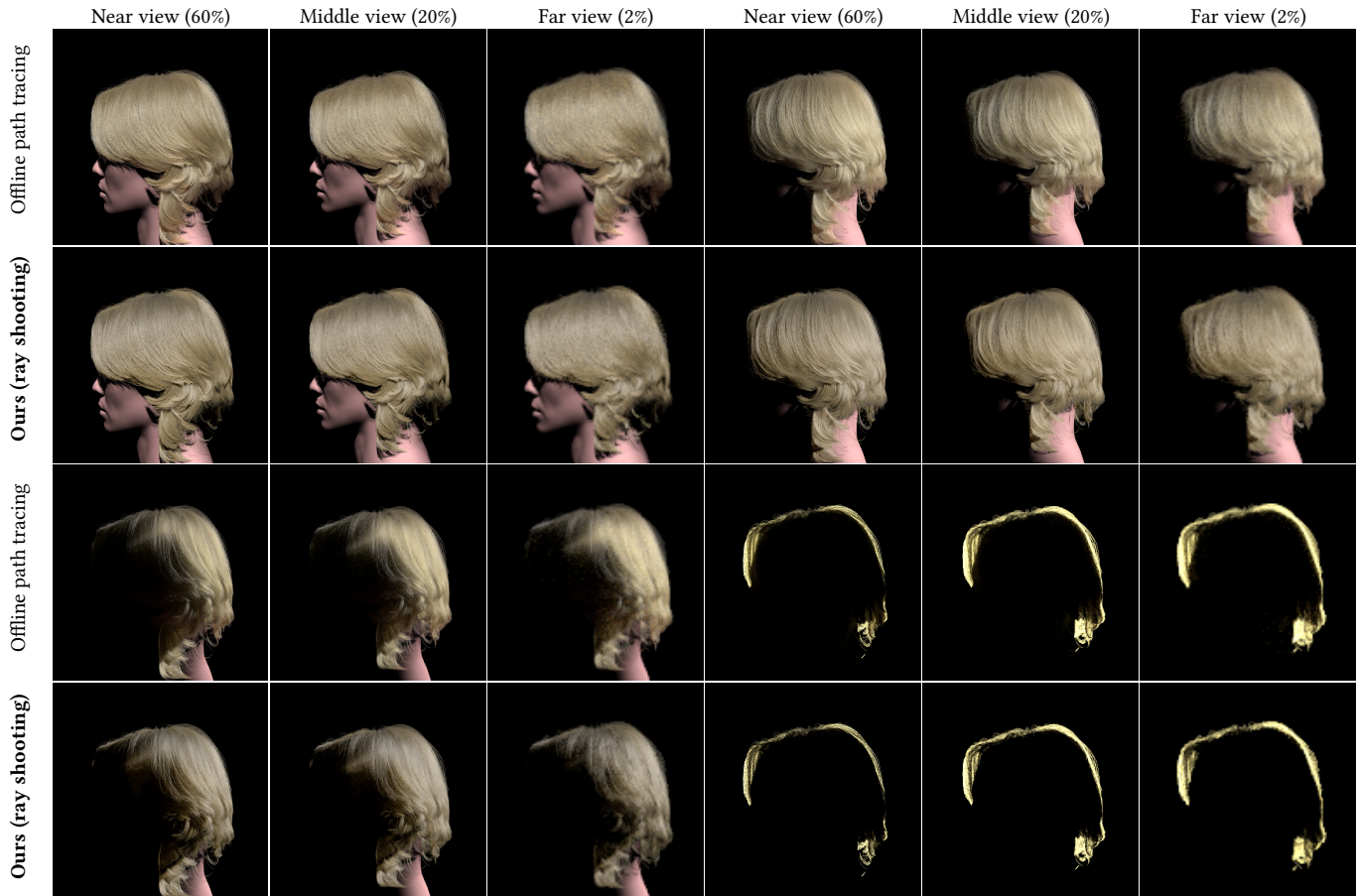


Fig. 16. Our approach can yield results closely resembling those obtained from offline path tracing when the angles between the view direction and light direction are 0° , 45° , 90° , and 180° . It's worth noting that offline path tracing employs full hair geometry with 1050K segments, while ours utilizes 1010K, 857K, and 418K segments for near, middle, and far views, respectively.

- David Luebke, Benjamin Watson, Jonathan D. Cohen, Martin Reddy, and Amitabh Varshney. 2002. *Level of Detail for 3D Graphics*. Elsevier Science Inc., USA.
- 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.
- Zahra Montazeri, Soren Gammelmark, Henrik Wann Jensen, and Shuang Zhao. 2021. Practical Ply-Based Appearance Modeling for Knitted Fabrics. *Computer Graphics Forum* 42, 4 (July 2021), 2–11.
- Zahra Montazeri, Soren B. Gammelmark, Shuang Zhao, and Henrik Wann Jensen. 2020. A Practical Ply-Based Appearance Model of Woven Fabrics. *ACM Trans. Graph.* 39, 6, Article 251 (nov 2020), 13 pages.
- 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.
- Jonathan T. Moon, Bruce Walter, and Steve Marschner. 2008. Efficient Multiple Scattering in Hair Using Spherical Harmonics. In *ACM SIGGRAPH 2008 Papers* (Los Angeles, California) (SIGGRAPH '08). Association for Computing Machinery, New York, NY, USA, Article 31, 7 pages.
- Liangliang Nan and Peter Wonka. 2017. PolyFit: Polygonal Surface Reconstruction from Point Clouds. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Venice, Italy, 2372–2380.
- Zhong Ren, Kun Zhou, Tengfei Li, Wei Hua, and Baining Guo. 2010. Interactive hair rendering under environment lighting. *ACM Trans. Graph.* 29, 4, Article 55 (jul 2010), 8 pages.
- Iman Sadeghi, Heather Pritchett, Henrik Wann Jensen, and Rasmus Tamstorf. 2010. An Artist Friendly Hair Shading System. In *ACM SIGGRAPH 2010 Papers* (Los Angeles, California) (SIGGRAPH '10). Association for Computing Machinery, New York, NY, USA, Article 56, 10 pages.
- Yuefan Shen, Shunsuke Saito, Ziyang Wang, Olivier Maury, Chenglei Wu, Jessica Hodgins, Youyi Zheng, and Giljoo Nam. 2023. CT2Hair: High-Fidelity 3D Hair Modeling Using Computed Tomography. *ACM Trans. Graph.* 42, 4, Article 75 (jul 2023), 13 pages.
- Arunachalam Somasundaram. 2015. Dynamically Controlling Hair Interpolation. In *ACM SIGGRAPH 2015 Talks* (Los Angeles, California) (SIGGRAPH '15). Association for Computing Machinery, New York, NY, USA, Article 36, 1 pages.
- Sebastian Tafuri. 2019. Strand-based Hair Rendering in Frostbite. *ACM SIGGRAPH Courses: Advances in Real-Time Rendering in Games Course 0*, 0 (2019), 0.
- Ingo Wald, Sven Woop, Carsten Benthin, Gregory S. Johnson, and Manfred Ernst. 2014. Embree: a kernel framework for efficient CPU ray tracing. *ACM Trans. Graph.* 33, 4, Article 143 (jul 2014), 8 pages.
- Lvdi Wang, Yizhou Yu, Kun Zhou, and Baining Guo. 2009. Example-Based Hair Geometry Synthesis. In *ACM SIGGRAPH 2009 Papers* (New Orleans, Louisiana) (SIGGRAPH '09). Association for Computing Machinery, New York, NY, USA, Article 56, 9 pages.
- Kelly Ward, Ming C Lin, Lee Joohee, Susan Fisher, and Dean Macri. 2003. Modeling hair using level-of-detail representations. In *Proceedings 11th IEEE International Workshop on Program Comprehension*. IEEE, New Brunswick, NJ, USA, 41–47.
- Philippe Weier, Tobias Zirr, Anton Kaplanyan, Ling-Qi Yan, and Philipp Slusallek. 2023. Neural Prefiltering for Correlation-Aware Levels of Detail. *ACM Trans. Graph.* 42, 4, Article 78 (jul 2023), 16 pages.
- Mengqi Xia, Bruce Walter, Christophe Hery, Olivier Maury, Eric Michielssen, and Steve Marschner. 2023. A Practical Wave Optics Reflection Model for Hair and Fur. *ACM Trans. Graph.* 42, 4, Article 39 (jul 2023), 15 pages.
- Mengqi (Mandy) Xia, Bruce Walter, Eric Michielssen, David Bindel, and Steve Marschner. 2020. A Wave Optics Based Fiber Scattering Model. *ACM Trans. Graph.* 39, 6, Article 252 (nov 2020), 16 pages.

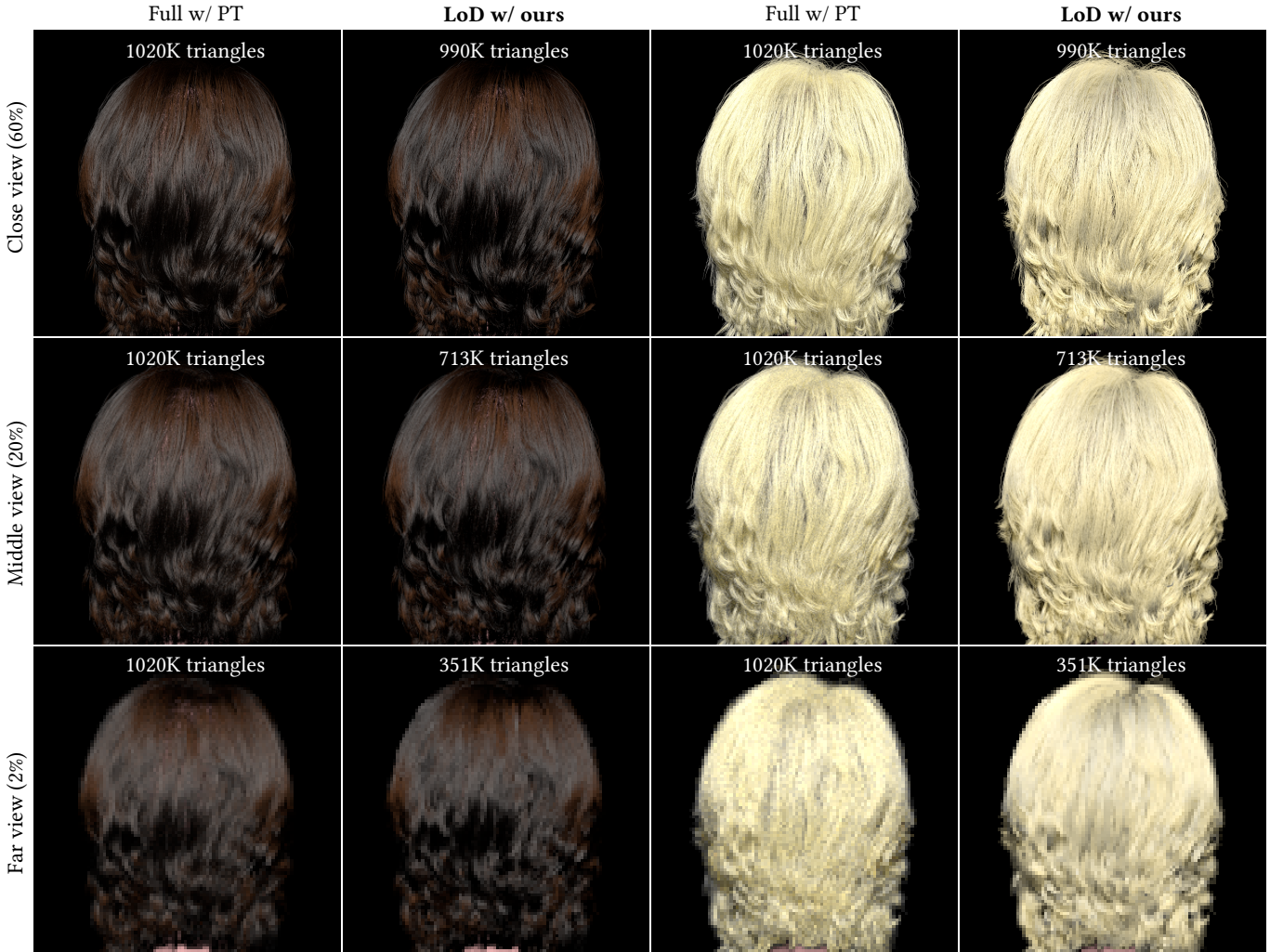
(a) BCSDf [Marschner et al. 2003] with R , TT , and TRT (b) BCSDf [Zhu et al. 2023] with R , TT , and D

Fig. 17. Additional examples: (a) Single hair BCSDf with TRT term [Marschner et al. 2003] for dark-colored hair and (b) BCSDf [Zhu et al. 2023] with R , TT , and D for light-colored hair. For both shading models, our strand-based LoD model (**LoD w/ ours**) effectively reduces the number of fibers and preserves appearance compared to path tracing with full hair geometry (**full w/ PT**). And we mark the number of segments for hair strips in the image.

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, Article 185 (nov 2015), 13 pages.

Cem Yuksel and John Keyser. 2008. Deep Opacity Maps. *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2008)* 27, 2 (2008), 675–680.

Junqiu Zhu, Zahra Montazeri, Julien Aubry, Ling-Qi Yan, and Andrea Weidlich. 2023. A Practical and Hierarchical Yarn-based Shading Model for Cloth. *Computer Graphics Forum* 42, 4 (July 2023), 2–11.

Junqiu Zhu, Sizhe Zhao, Lu Wang, Yanning Xu, and Ling-Qi Yan. 2022. Practical level-of-detail aggregation of fur appearance. *ACM Trans. Graph.* 41, 4 (2022), 47:1–47:17.

Arno Zinke, Martin Rump, Tomás Lay, Andreas Weber, Anton Andriyenko, and Reinhard Klein. 2009. A practical approach for photometric acquisition of hair color. *ACM Trans. Graph.* 28, 5 (2009), 165.

Arno Zinke, Cem Yuksel, Andreas Weber, and John Keyser. 2008. Dual scattering approximation for fast multiple scattering in hair. *ACM Trans. Graph.* 27, 3 (2008), 32.