Supplemental Material for ISHair: Importance Sampling for Hair Scattering

Jiawei Ou[1,2] Feng Xie[1] Parashar Krishnamachari[1] Fabio Pellacini[2,3]
[1]DreamWorks Animation [2]Dartmouth College [3]Sapienza University of Rome

Contents

1 Introduction

This document is the supplemental material for SIGGRAPH late breaking submission *ISHair: Importance Sampling for Hair Scattering*. Due to the limited space in the one-page abstract, we are not able to provide all the details in the sketch. Instead, we provide the complete set of derivations, rendering results and pseudocode implementation in this document. This document is organized as following: Section 2 is the full length paper submitted to EGSR 2012, which provides interested readers the motivation and background knowledge for our work, and a more complete set of results. Section 3 gives the detailed derivations of equations used in our approach. Section 4 provides rendering comparisons between our approach and [Hery and Ramamoorthi 2011]. Section 5 contains the pseudocode implementation of our approach written in Python. It can be easily translated into other programming languages.

2 Paper

The following pages are the full paper submitted to EGSR 2012 (decision pending). They are for review purpose only.

ISHair: Importance Sampling for Hair Scattering

Jiawei Ou[1,2] Feng Xie[1] Parashar Krishnamachari[1] Fabio Pellacini[2,3]
[1]DreamWorks Animation [2]Dartmouth College [3]Sapienza University of Rome

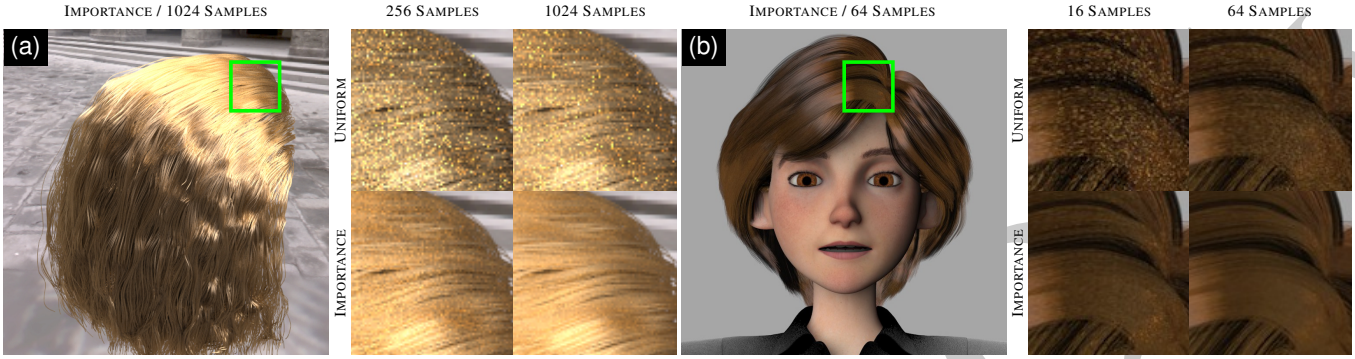

Figure 1: *Comparison between stratified uniform sampling and our importance sampling method. (a) Hair under environment lighting rendered with global illumination using path tracing. Our method efficiently samples the scattering direction in multiple bounces and converges with significanly fewer samples than uniform sampling. (b) Direct illumination of hair under environment lighting rendered with a production renderer. Our method delivers better image quality than rendering with* $4\times$ *number of uniform samples.*

Abstract

We present an importance sampling method for the bidirectional scattering distribution function (*bsdf*) of hair. Our method is based on the multi-lobe hair scattering model presented by Sadeghi et al. [2010]. We reduce noise by drawing samples from a distribution that approximates the *bsdf* well. Our algorithm is efficient and easy to implement, since the sampling process requires only the evaluation of a few analytic functions, with no significant memory overhead or need for precomputation. We tested our method in a research raytracer and a production renderer based on micropolygon rasterization. We show significant improvements for rendering direct illumination using multiple importance sampling and for rendering indirect illumination using path tracing.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture;

Keywords: hair rendering, importance sampling

1 Introduction

Hair is a ubiquitous element of human and animal characters. High-quality hair rendering is essential to provide believable appearance in digitally-created content. We are interested in rendering hair lit by area and environment lights without precomputation to support dynamic scenes. These physically based light sources have become prevalent in both visual effects and animated feature films.

Marschner et al. introduced a physically-based scattering model that captures all the nuances of hair's appearance [Marschner et al. 2003]. However, this model is computationally expensive, requiring the solution of a cubic equation derived by internal path analysis. Moreover, it is cumbersome for artists to directly control the appearance of hair by changing the model's parameters. To address these problems, Sadeghi et al. proposed an artist-friendly shading model for hair that approximates Marschner's model using only elementary functions, making it easier for artists to control than the purely physical based model [Sadeghi et al. 2010]. In this paper, we concentrate on this latter model.

Both these hair shading models have narrow peaks in their specular lobes, especially for shiny hair. This causes severe noise in Monte Carlo based rendering methods, especially when combined with large area lights and environment maps. Importance sampling is a widely used variance reduction technique for Monte Carlo numerical integration. In the context of rendering, importance sampling offers a means to reduce the variance by concentrating samples in regions with significant contribution to the illumination integral. We present an efficient importance sampling method for the hair scattering *bsdf* of [Sadeghi et al. 2010]. Our method is capable of significantly improving the quality of the rendered image, as seen in Figure 1, with negligible overhead. We reduce noise by drawing samples from a distribution that approximates well the scattering function in [Sadeghi et al. 2010]. We do so efficiently since drawing samples requires only the evaluation of a few analytic functions, with no precomputation or significant memory footprint. We found our method easy to implement both in a prototype path tracer and in a micropolygon based production renderer. In both cases, results are further improved by using importance sampling of the *bsdf* in conjunction with importance sampling of lighting, a technique commonly known as multiple importance sampling [Veach and Guibas 1995].

The main contribution of our work is to provide a sampling algorithm for hair scattering that is effective (at reducing noise), robust, simple to implement and efficient to evaluate. In the remainder of the paper, we start with a brief overview of related work, followed by the presentation of our algorithm and results, and end with the discussion and conclusion.

2 Related Work

Photorealistic Hair Rendering. There is a large body of work regarding hair modeling and shading. Here we review only the pub-

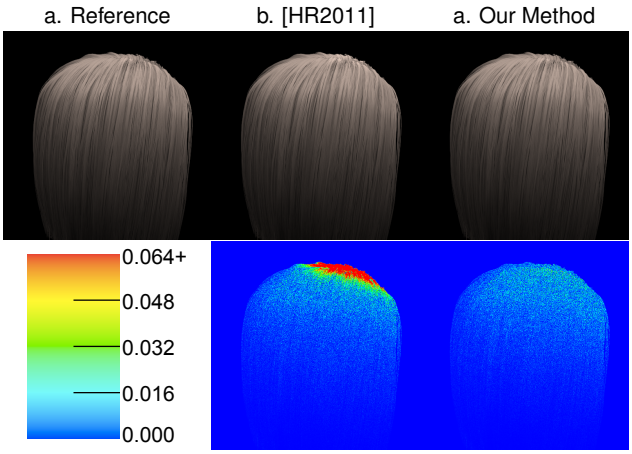

Figure 2: *Error images of [Hery and Ramamoorthi 2011] and our method. The edge cases of Box-Muller transform is not correctly handled in [Hery and Ramamoorthi 2011], resulting in incorrect energy estimation at grazing angles. Moreover, their method will not converge to correct solution as sample count increases. (The error images are computed using per-pixel L^2 difference.)*

lications most closely related to our work, referring the reader to [Ward et al. 2007] for a detailed review. Kajiya and Kay proposed the first prominant model for hair rendering where they modeled the hair *bsdf* by computing light scattering from thin cylinders [Kajiya and Kay 1989]. Marschner et al. improved upon this model by incorporating internal path analysis of hair strands [Marschner et al. 2003]. Marschner's work was the first complete physically-based hair shading model, capable of capturing the complex scattering behavior of hair. Zinke and Weber proposed a more general framework for filaments scattering [Zinke and Weber 2007]. Inspired by Marschner's model, Sadeghi et al. derived a practical hair shading model that is more efficient and easier for artist to control [Sadeghi et al. 2010]. d'Eon et al. proposed an energy conserving hair reflectance model that includes several modifications to Marschner's model to ensure energy conservation during scattering [d'Eon et al. 2011]. These models focus on providing accurate *bsdfs* for hair, but none provides an efficient method to importance sample the scattering functions. This is the focus of our work. While our method speeds up multiple scattering using Monte Carlo methods, it can also be integrated with more efficient multiple scattering solutions such as [Moon and Marschner 2006; Moon et al. 2008; Zinke et al. 2008].

Importance Sampling Surface Materials. High-quality Monte Carlo rendering requires the ability to importance sample realistic *bsdf* models. There has been extensive research on importance sampling surface *bsdfs*, as summarized in [Pharr and Humphreys 2010]. Analytic methods exist only for simple *bsdfs* such as Phong [Phong 1975], Lafortune [Lafortune et al. 1997] and Ward [Larson 1992]. For more complex *bsdfs* and measured material, approximations of varying degrees of quality are applied. A more general solution is to derive importance sampling functions using factorized representations or basis projections of *bsdfs* [Lawrence et al. 2004; Clarberg et al. 2005; Jarosz et al. 2009]. However, these methods require precomputation and have a high memory footprint, making them impractical for spatially-varying materials. Our method uses an accurate analytic approximation that does not suffer from these constraints.

Importance Sampling Hair. Moon and Marschner proposed to sample the scattering directions by tracing rays through a rough

symbol	description
$S(\theta_i, \phi_i, \theta_r, \phi_r)$	hair *bsdf*
M_{R}, M_{TT}, M_{TRT}	longitudinal scattering functions
N_{R}, N_{TT}, $N_{\text{TRT-g}}$, N_{g}	azimuthal scattering functions
ω_i	incoming direction
ω_r	reflected direction
$\mathbf{u}$	hair direction, pointing from the root to the tip
$\mathbf{v}, \mathbf{w}$	axes of the normal plane, orthogonal to $\mathbf{u}$
θ_i, θ_r	inclination of ω_i and ω_r w.r.t the normal plane where 0° is perpendicular to $\mathbf{u}$, 90° is $\mathbf{u}$, and -90° is $-\mathbf{u}$
ϕ_i, ϕ_r	azimuthal angles of ω_i and ω_r in the normal plane where $\mathbf{v}$ is 0° and $\mathbf{w}$ is 90°
ϕ	relative azimuthal angle, $\phi = \phi_r - \phi_i$
θ_d	longitudinal difference angle $\theta_d = (\theta_r - \theta_i)/2$
θ_h	longitudinal half angle $\theta_h = (\theta_r + \theta_i)/2$

Table 1: *Summary of notation.*

elliptical cylinder, instead of importance sampling the hair *bsdf* [Moon and Marschner 2006]. Moon et al. sped up this process using a precomputed lookup table [Moon et al. 2008]. These methods were either computationally expensive or required precomputation. Neulander et al. derived a practical importance sampling algorithm based on a cone-shell hair *bsdf* model, which was a variant of the Kajiya-Kay model [Neulander 2010]. Their method, however, does not support hair models that have multiple specular lobes with different widths and offsets, so it does not apply to Marschner's hair model or its variants. Hery and Ramamoorthi proposed an importance sampling method for the reflection lobe of hair *bsdf* [Hery and Ramamoorthi 2011]. Their method relied on the Box-Muller transform to sample the Gaussian distribution. This approach has several difficult edge cases and the solution presented in [Hery and Ramamoorthi 2011] has significant error and bias that can result in rendering artifacts (see Figure 2 and Appendix B).

Hair Rendering under Environment Lighting. Hair rendering under environment lighting benefits from an efficient *bsdf* importance sampling algorithm. This case is so common that algorithms have been developed specifically for it [Ren et al. 2010; Xu et al. 2011]. While these methods work well in their problem domain, they are limited to environment lighting and require a considerable amount of precomputation. Moreover, they are all derived by approximations of the illumination integral, which makes it hard to integrate shadowing from complex dynamic occluders into these methods. These constraints limit their applicability in production rendering.

3 Hair Importance Sampling

We start the presentation of our importance sampling method with a summary of the hair shading model it supports, followed by the derivation of sampling functions for all the lobes of the hair *bsdf* and a complete description of our algorithm and its implementation. We follow the notation summarized in Table 1.

3.1 Hair Shading Function

Sadeghi et al. [2010] propose an artist-friendly hair shading model, where the scattering function $S(\theta_i, \phi_i, \theta_r, \phi_r)$ of hair fibers is decomposed into four individual components: reflection (R), refractive transmission (TT), secondary reflection without glint (TRT-g) and *Glint* (g). Each component is represented as a separate lobe and further factored as the product of a longitudinal term M and an

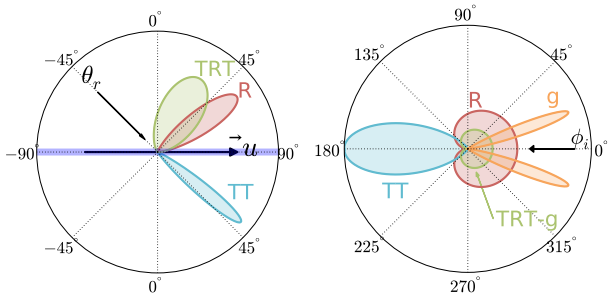

Figure 3: *Shapes of each hair lobes: Longitudinal lobes (Left). Azimuthal lobes (Right).*

azimuthal term N. The full scattering model is:

$$\begin{aligned} S(\theta_i, \phi_i, \theta_r, \phi_r) &= I_{\text{R}} M_{\text{R}}(\theta_h) N_{\text{R}}(\phi) / \cos^2 \theta_d \\ &+ I_{\text{TT}} M_{\text{TT}}(\theta_h) N_{\text{TT}}(\phi) / \cos^2 \theta_d \\ &+ I_{\text{TRT}} M_{\text{TRT}}(\theta_h) N_{\text{TRT-g}}(\phi) / \cos^2 \theta_d \\ &+ I_{\text{TRT}} M_{\text{TRT}}(\theta_h) I_{\text{g}} N_{\text{g}}(\phi) / \cos^2 \theta_d \end{aligned} \tag{1}$$

I_{R}, I_{TT} and I_{TRT} are the colored intensities of the corresponding lobe while I_{g} is the additional intensity of the *Glint* lobe.

M_{R}, M_{TT} and M_{TRT} model the longitudinal variation of each lobe. All three are Gaussian functions of the longitudinal half angle θ_h as

$$M_{\text{R}} = g(\beta^2_{\text{R}}, \alpha_{\text{R}}, \theta_h) \qquad M_{\text{TT}} = g(\beta^2_{\text{TT}}, \alpha_{\text{TT}}, \theta_h)$$
$$M_{\text{TRT}} = g(\beta^2_{\text{TRT}}, \alpha_{\text{TRT}}, \theta_h)$$

where β_{R}, β_{TT}, β_{TRT} and α_{R}, α_{TT}, α_{TRT} are the widths and means of corresponding Gaussian functions. α controls the highlight shift of each lobe, while β changes the roughness of the hair. In our notation,

$$g(\beta^2, \alpha, \theta_h) = exp\left[-\frac{(\theta_h - \alpha)^2}{2\beta^2}\right]$$

N_{R}, N_{TT}, $N_{\text{TRT-g}}$ and N_{g} model the azimuthal variation of each lobe. All azimuthal terms are functions of the relative azimuthal angle $\phi = \phi_r - \phi_i$ and are defined respectively as

$$N_{\text{R}} = \cos(\phi/2) \qquad N_{\text{TT}} = g(\gamma^2_{\text{TT}}, \pi - \phi)$$
$$N_{\text{TRT-g}} = \cos(\phi/2) \qquad N_{\text{g}} = g(\gamma^2_{\text{g}}, |\phi| - \phi_{\text{g}})$$

where γ_{TT} is a user controllable azimuthal width for N_{TT}. N_{g} has two Gaussian functions with widths γ_{g} that are symmetric about the axis $\phi = 0$, and ϕ_{g} is the half angle between its peaks.

3.2 Importance Sampling

To efficiently reduce variance in Monte Carlo integration, we seek to draw samples from a distribution whose probability distribution function (*pdf*) is proportional to the function we are integrating. In the context of hair rendering, we want to sample ω_i such that $p(\omega_i) \propto S(\theta_i, \phi_i, \theta_r, \phi_r)$.

Because the hair *bsdf* model has multiple lobes, it is impractical to sample all the lobes at the same time. So we first describe how to efficiently sample each individual lobe; then we show how to combine all the lobes by randomly selecting a lobe based on an estimate of its energy. The longitudinal terms and azimuthal terms can be sampled independently since they depend on different variables. Specifically, we sample the spherical angles θ_i and ϕ_i separately, and then convert them into the direction ω_i. The *pdf* of the sample is a product of the longitudinal *pdf* and the azimuthal *pdf* as $p(\omega_i) = p(\theta_i)p(\phi_i)$. We use the inverse cumulative distribution function (*cdf*) technique described in [Pharr and Humphreys 2010] to derive our analytic sampling functions.

Sampling Gaussians. Equation (1) uses Gaussian functions to model the variation in longitudinal and azimuthal scattering. Box-Muller transform is a general approach to draw samples from a Gaussian distribution with an infinite domain [**?**]. However, for this specific problem, we have to draw samples from a Gaussian distribution with a finite domain, e.g. $[\frac{-\pi/2+\theta_r}{2}, \frac{\pi/2+\theta_r}{2}]$. Using Box-Muller transform can result in samples outside of the finite domain (edge cases) that are difficult to handle (See appendix A).

Inverse cdf can be used to constrain the samples to fall within the finite domain, but the lack of a closed form anti-derivative for Gausisian makes this approach infeasible. Although there are numerical approximations for the *pdf* and *cdf* of the Gaussian, they require the evaluation of error functions or Taylor series [Pressa et al. 2007]. These methods are either computationally expensive or unstable at the tail of the Gaussian. To overcome these limitations, we would like to draw samples from a *pdf* that has a similar shape to the Gaussian function and a closed-form antiderivative. Observing that the Gaussian is a bell-shaped function with varying width and center, we can approximate it using another bell-shaped function.

Cauchy distribution. The Cauchy distribution is a probability distribution mainly used in physics, and it was recently used by computer graphics researchers as a sampling distribution [Kulla et al. 2011]. It is defined as:

$$f(\gamma, x - x_0) = \frac{1}{\pi}\left[\frac{\gamma}{(x - x_0)^2 + \gamma^2}\right]$$

Similar to the Gaussian, the Cauchy distribution is a bell-shaped function with offset x_0 and width γ. In contrast to the Gaussian, it has an analytic antiderivative

$$P(x) = \frac{1}{\pi}\tan^{-1}\left(\frac{x - x_0}{\gamma}\right)$$

This simple form of the antiderivative makes it possible to derive a sampling algorithm using the inverse *cdf* technique. The offset and width of a Gaussian distribution can be directly used as the offset and width of the Cauchy distribution correspondingly. Figure 4 shows the plot of a set of Gaussian and Cauchy functions with same widths and offsets. The fact that Cauchy distributions have wider tails than Gaussians guarantees that using the Cauchy distribution to approximate the Gaussian in importance sampling will not increase variance. Using this approximation, we derive our sampling method for each lobe.

3.2.1 Sampling Longitudinal Terms

Since the three longitudinal terms have the same form, we describe the approach using generic symbols M, β and α. Note that we ignore the $1/\cos^2\theta_d$ terms for simplicity, since M alone accounts for most of the variation in the longitudinal terms[1]. Substituting the Gaussian functions in the M terms with Cauchy distributions allows us to derive the sampling functions for incoming inclination

[1] $1/\cos^2\theta_d$ term has a singularity when both θ_r and θ_i approach $\pi/2$ or $-\pi/2$. However, the projection term $\cos\theta_i$ in the rendering equation cancels its effect because $\cos\theta_i/\cos^2\theta_d$ is a smooth function. Therefore the M term remains the dominant source of variance.

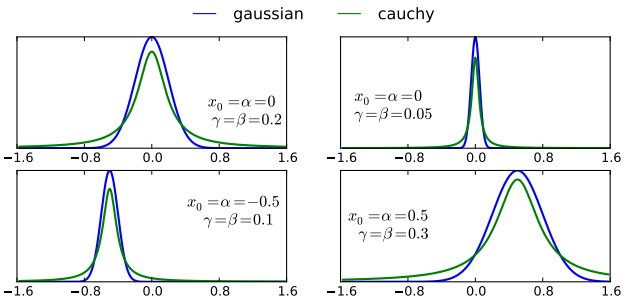

Figure 4: *Cauchy and Gaussian distributions with same widths and offsets. Both of distributions are normalized in the domain* $[-\pi/2, \pi/2]$.

θ_i. Given a random variable ξ uniformly drawn from range $[0, 1)$, we can sample θ_i as:

$$\theta_i = 2\beta \tan(\xi(A - B) + B) + 2\alpha - \theta_r$$

where $A = \tan^{-1}\left(\frac{\pi/4+\theta_r/2-\alpha}{\beta}\right)$ and $B = \tan^{-1}\left(\frac{-\pi/4+\theta_r/2-\alpha}{\beta}\right)$. The longitudinal *pdf* can be computed as

$$p(\theta_i) = \frac{1}{2\cos\theta_i(A - B)} \frac{\beta_x}{(\theta_h - \alpha)^2 + \beta^2}$$

3.2.2 Sampling Azimuthal Terms

All azimuthal terms are functions of relative azimuthal angle $\phi = \phi_r - \phi_i$. In our approach, we first sample ϕ, then compute $\phi_i = \phi_r - \phi$. The *pdf* of ϕ is the same as the *pdf* of ϕ_i since $p(\phi_i) = p(\phi)\left|d\phi_i/d\phi\right|^{-1} = p(\phi)$.

Sampling N_R. N_{R} is evaluated as $\cos(\phi/2)$. Deriving a sampling function for this term is trivial. Given a uniform random variable ξ in $[0, 1)$, we sample ϕ as

$$\phi = 2\sin^{-1}(2\xi - 1)$$

then we can compute $\phi_i = \phi_r - \phi$ and the azimuthal *pdf* $p(\phi_i) = p(\phi) = \frac{1}{4}\cos\frac{\phi}{2}$.

Sampling N_{TT}. N_{TT} is defined with a Gaussian that is positive in the range $[0, 2\pi]$. We take an approach similar to the longitudinal terms. Given a uniform random variable ξ in $[0, 1)$, we draw a sample of ϕ as

$$\phi = \gamma_{\text{TT}} \tan\left[C_{\text{TT}}\left(\xi - \frac{1}{2}\right)\right] + \pi$$

where $C_{\text{TT}} = 2\tan^{-1}\left(\frac{\pi}{\gamma_{\text{TT}}}\right)$. We then compute $\phi_i = \phi_r - \phi$ and the azimuthal *pdf* $p(\phi_i) = p(\phi) = \frac{1}{C_{\text{TT}}}\left[\frac{\gamma_{\text{TT}}}{(\phi-\pi)^2+\gamma_{\text{TT}}^2}\right]$.

Sampling N_{TRT-g}. $N_{\text{TRT-g}}$ is approximated as $\cos(\phi/2)$. Since it is the same as the N_{R} term, we follow the same approach as sampling N_{R}.

Sampling N_g. *Glint* models the lighting effect caused by the caustic light path inside hair strands. The azimuthal term of *Glint* is defined as two Gaussian functions symmetric about the $\phi = 0$ axis. Given a uniform random variable ξ from $[0, 1)$, we choose one of two glint lobes by setting the sign of ϕ and remap ξ back to the range $[0, 1)$ accordingly. Specifically, for $\xi < 1/2$, we set ϕ positive and map $\xi \leftarrow 2\xi$. For $\xi \geq 1/2$, we set ϕ negative and map $\xi \leftarrow 2(1 - \xi)$. After that, we sample ϕ using the remapped ξ as

$$\phi = \gamma_{\text{g}} \tan(\xi(C_{\text{g}} - D_{\text{g}}) + D_{\text{g}}) + \phi_{\text{g}}$$

where $C_{\text{g}} = \tan^{-1}\left(\frac{\pi/2-\phi_{\text{g}}}{\gamma_{\text{g}}}\right)$, $D_{\text{g}} = \tan^{-1}\left(\frac{-\phi_{\text{g}}}{\gamma_{\text{g}}}\right)$. Once we have ϕ, we compute $\phi_i = \phi_r \pm \phi$, and compute its *pdf* as $p(\phi_i) = \frac{1}{2}p(\phi) = \frac{1}{2(C_{\text{g}}-D_{\text{g}})}\left[\frac{\gamma_{\text{g}}}{(|\phi|-\phi_{\text{g}})^2+\gamma_{\text{g}}^2}\right]$, taking into account our remapping of the random variable.

3.2.3 Energy-based lobe selection

We have discussed how to sample each individual lobe. To sample the complete *bsdf*, we distribute samples to each lobe. A simple solution is to uniformly select a lobe. To better match the energy distribution of the *bsdf*, however, we use an energy-based lobe selection scheme. For each sample, we select a lobe with a probability proportional to an estimate of the energy of each lobe. We estimate these energies as the product of the integrals of the longitudinal and azimuthal terms. This results in the following estimates:

$$\text{E}_{\text{R}} = 4\sqrt{2\pi}\beta_{\text{R}} I_{\text{R}} \qquad \text{E}_{\text{TT}} = 2\pi\beta_{\text{TT}}\gamma_{\text{TT}} I_{\text{TT}}$$

$$\text{E}_{\text{TRT-g}} = 4\sqrt{2\pi}\beta_{\text{TRT}} I_{\text{TRT}} \qquad \text{E}_{\text{g}} = 4\pi\beta_{TRT}\gamma_{\text{g}} I_{\text{TRT}} I_{\text{g}}$$

We use the Gaussian integral in the domain $[-\infty, \infty]$ instead of $[-\pi/2, \pi/2]$ to compute the estimated energy. Although this is not accurate in general, it is easy to compute and works well as an estimation. The approximation error is less than 1% for $\beta < 30^\circ$ and $|\alpha| < 20^\circ$ and 0.003% for $\beta < 20^\circ$ and $|\alpha| < 10^\circ$.

We provide step-by-step derivations and the Python code for the sampling method in the supplemental material.

3.3 Implementation Details

Amortizing constants computation. It is important to note that A_{R}, A_{TT}, A_{TRT}, B_{R}, B_{TT}, B_{TRT}, C_{TT}, C_{g} and D_{g} in the sampling functions are constant for all the samples of the same gathering point and reflective direction ω_r. We compute these constants once and amortize the cost over all the samples.

Longitudinal grazing-angle pdf. Notice that the longitudinal *pdf* has a singularity when θ_i approaches $-\pi/2$ or $\pi/2$. The sample evaluation becomes numerically unstable at grazing angles. To avoid this problem, our implementation discards the sample if the angle between ω_i and $\mathbf{u}$ or $-\mathbf{u}$ is smaller then a predefined epsilon (10^{-5} in our case). Although in theory this may bias the result, In practice, it only rejects a small percentage of samples ($< 0.001\%$) and all discarded samples have negligible contribution with weights (< 0.0001), resulting in negligible bias.

4 Results

4.1 Sample Distribution.

Figure 5 shows the sample distributions using the described importance sampling scheme. We use the Halton quasi-random sequence to generate the samples since it is repeatable and stratified [Pharr and Humphreys 2010]. Compared to uniform sampling (Figure 5a), our importance sampling method (Figure 5b) concentrates samples in regions of high importance. Figures 5c-5f show the sample distribution of each individual lobe.

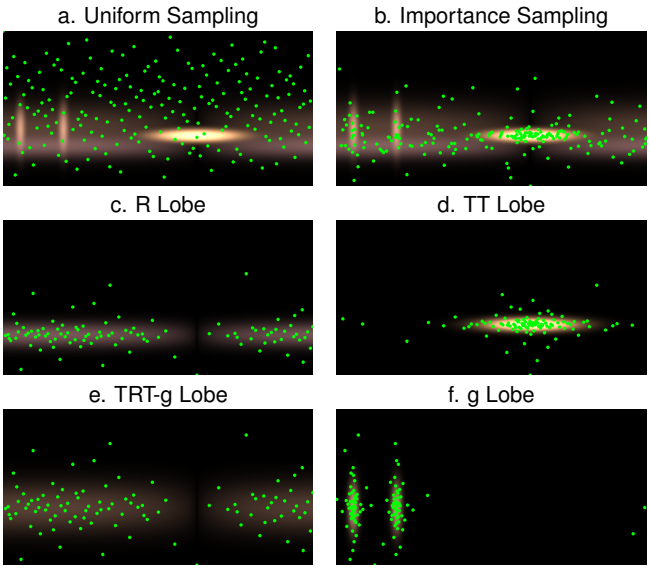

Figure 5: *Comparison of samples distributed using (a) uniform and (b) importance sampling, where we use the pseudo-random Halton sequence to generate well-distributed random numbers. We also show the sample distributions of each individual lobe using our importance sampling scheme. Notice the computed sample distribution match the energy distribution of the* bsdf.

4.2 Rendering Result

Overview. We implemented our importance sampling scheme for hair *bsdf* in a raytracing renderer written in C++. Moreover, to test our approach in a movie production environment, we also implemented our algorithm in a production renderer. Figure 6 and Figure 7 show comparisons of our method with uniform sampling. For both sampling schemes, we stratify the random numbers and render the images using multiple importance sampling (MIS) for direct illumination[2]. These are the best conditions for uniform sampling. Since we found our method to have negligible cost, we just report sample count rather than timing.

Area Lighting. Physically correct area lights have become widely adopted in production rendering. Figure 6(a) is a simple scene with a large area light above the hair geometry rendered with our raytracer. Uniform sampling exhibits significant noise at low sampling rates (32 samples), while the importance sampled result is relatively smooth. With 128 samples, the importance sampling image has no visible noise, while the uniform sampled image still has some distracting noise. Figure 7(a) is a production model lit with a large area light rendered with a production renderer. Due to the production renderer's antialiasing techniques, only a few (32) importance samples are required to generate a smooth image; while uniform sampling required over 1k samples to converge.

Environment Lighting. Environment maps are used to add realistic lighting to a scene. Figure 6(b) is a simple scene with an environment map of *Pisa Courtyard*. The illumination from this environment map is smooth with high color variation. In this case, while uniform sampling is not able to clean up the noise in the transmission and *Glint* highlights even at 256 samples, our importance sampling method is able to provide a smooth result with just a few samples (64 samples). Figure 7(b) is a production model lit with the environment map of *Ennis-Brown House*, where our importance sampling method delivers better image quality than images rendering with $4\times$ number of uniform samples.

Global Illumination. Global illumination enhances the overall realism of a scene. Of the many available algorithms, we use a path tracer since it is simple to implement and unbiased. In this case, scattering rays for indirect illumination are generated with *bsdf* sampling only, either uniform or with importance. Although resolving multiple scattering using brute force path tracing is inherently inefficient, it guarantees a physically correct result.

Figure 6(c) is a hair model lit by three area lights, and Figure 6(d) is a hair model lit by the *Grace Cathedral* environment map. In both scenes, the hair model is the only geometry. All the indirect illumination is the result of multiple scattering inside the hair geometry. Figure 6(e) is a simple scene with a hair model inside the Cornell Box, where indirect illumination comes from both the outside geometry and inside the hair geometry. For both uniform and importance sampling, we have to use a lot more samples in these tests than the previous direct lighting tests. While uniform sampling takes a long time to converge, our method converges to the correct result much faster with significantly fewer samples. Each reference image used more than 200M uniform samples per pixel and took over 60-80 hours to render.

5 Discussion and Limitations

Multiple Scattering. Our importance sampling algorithm is derived for the single scattering function. We show multiple scattering results rendered by path tracing, whose performance is drastically improved by using our importance sampling algorithm. Approximation algorithms for multiple scattering are beyond the scope of this paper. Although our sampling algorithm is not specifically designed for these algorithms, we believe that they can benefit from our work. For example, our importance sampling can be used to drive the photon shooting of [Moon and Marschner 2006] and light tracing of [Moon et al. 2008].

Extension to Marschner's Model. We believe our approach can be extended to support Marschner's model[Marschner et al. 2003]. It can be directly applied to sample the longitudinal (Gaussian) terms of the Marschner model[3]. However, applying it to the azimuthal terms is not trivial. Xu et al. proposed several approximations for fitting the azimuthal terms to Gaussian functions [Xu et al. 2011]. With these approximations, it would be possible to derive a sampling algorithm using our approach, with some precomputation and a small amount of overhead for each sample. We leave this extension to future work.

Integration with Other Sampling Techniques. Since our sampling algorithm does not require additional data structures, it can be easily integrated with other sampling techniques. We have only shown our method applied in a path tracer with multiple importance sampling, but it can also be used with other *Monte Carlo* techniques, e.g. photon mapping, bidirectional path tracing, or even more sophisticated unstructured illumination sampling techniques [Wang and Åkerlund 2009].

Replacing Gaussian with Cauchy. In this paper, we derive our *pdf* based on a *bsdf* model that uses Gaussian functions extensively.

[2] When using MIS, a sample count of 16 corresponds to 16 *bsdf* samples and 16 light samples.

[3] Marschner's model uses normalized Gaussians instead of the unnormalized ones used in [Sadeghi et al. 2010]'s model, but this will not affect the *pdf* derivation because the *pdf* is normalized by definition.

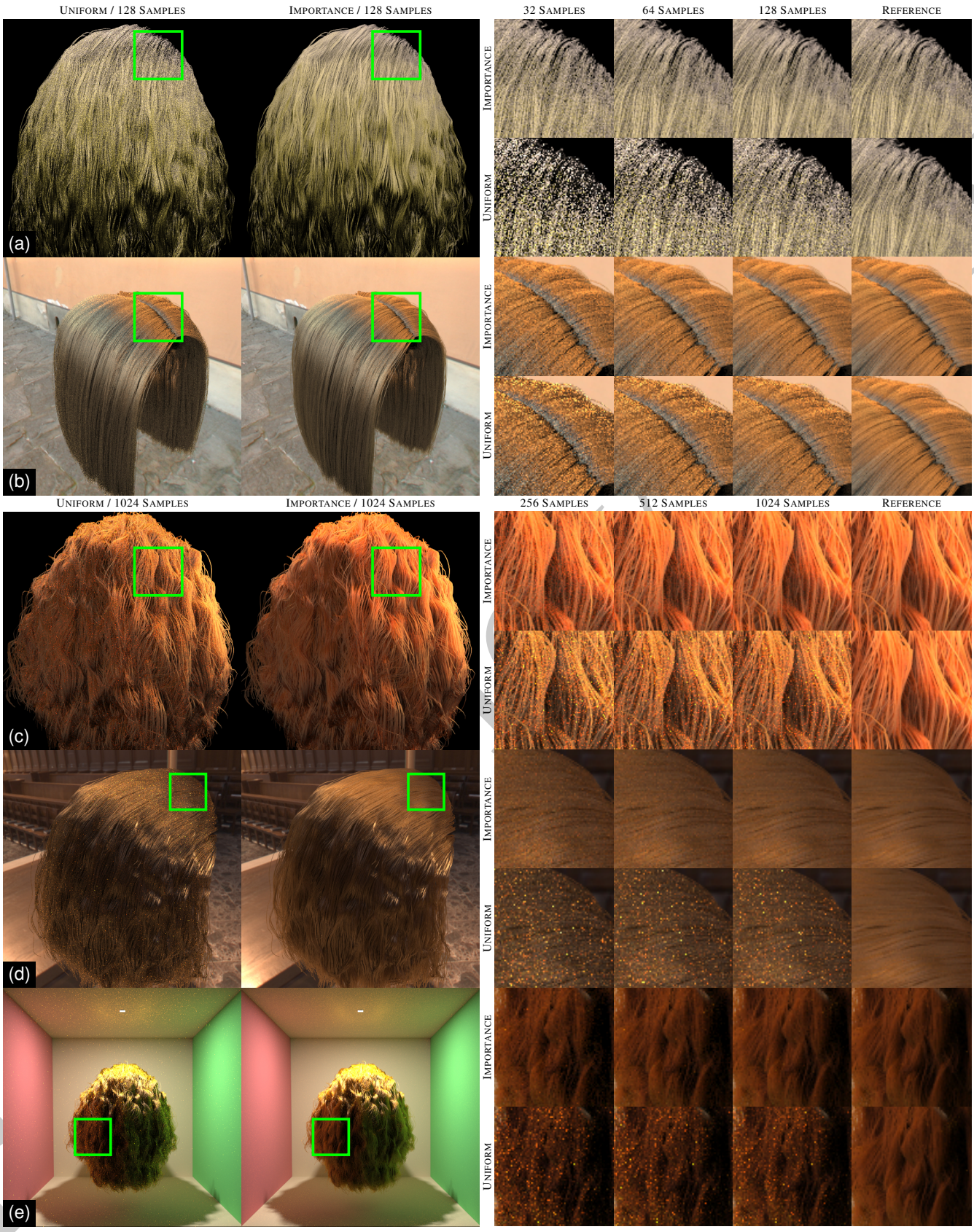

Figure 6: *Comparison of our importance sampling approach and simple uniform sampling in a ray tracing renderer. (a) Direct illumination with an area light, our method is able efficiently sample the longitudinal lobes. (b) Direct illumination with environment lighting, our method generates smooth result with significanly fewer samples than uniform sampling which fails to converge on the glint and transmission highlights. (c) Global illumination with area lighting and (d) global illumination with environment lighting show that our method is able to efficiently sample the scattering direction for multiple bounces and drastically reduce the sample number needed for convergence. (e) Global illumination in a Cornell box shows our method efficiently gathers radiance from surrounding geometry.*

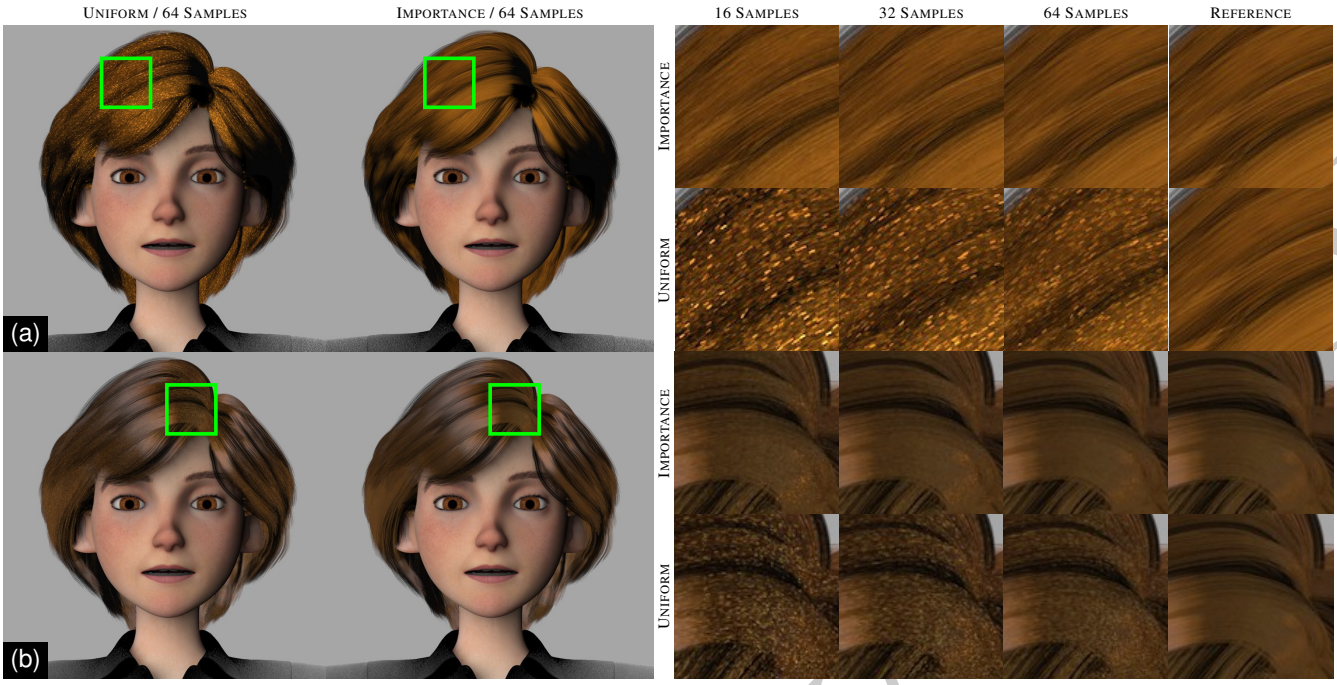

Figure 7: *Comparison of our importance sampling approach and simple uniform sampling in a production renderer: (a) Direct illumination with area lighting; (b) direct illumination with environment lighting. While uniform sampling has trouble converging to a smooth image, our method generates noise-free images with only a few samples.*

Although the Cauchy distribution can provide a good sample distribution for the Gaussian (Figure 5), the shape of these two distributions do not match exactly (see Figure 4). As an extension of our work, we propose a new hair *bsdf* by replacing all the Gaussian functions in [Sadeghi et al. 2010]'s model with Cauchy distributions. This new *bsdf* model introduces some minor visual differences, but it allows for a better match to the *pdf* during importance sampling. The comparison between this new *bsdf* and [Sadeghi et al. 2010]'s model is out of the scope of this paper. Interested readers can find comparisons using the new model in our supplemental material. All the results in this paper are generated using [Sadeghi et al. 2010]'s model.

6 Conclusions

We presented an importance sampling algorithm for the hair *bsdf*, that is simple to implement and efficient to evaluate. By approximating the Gaussian functions in the hair *bsdf* with Cauchy distribution, we were able to derive an analytic sampling algorithm with significantly reduced variance. We show results of applying our importance sampling method to render scenes with area lighting, environment lighting, indirect lighting and multiple scattering, in both a production renderer and a research raytracer.

References

Clarberg, P., Jarosz, W., Akenine-Möller, T., and Jensen, H. W. 2005. Wavelet importance sampling: efficiently evaluating products of complex functions. *ACM Transactions on Graphics 24*, 3 (Aug.), 1166–1175.

d'Eon, E., Francois, G., Hill, M., Letteri, J., and Aubry, J. 2011. An energy-conserving hair reflectance model. *Eurographics Symposium on Rendering 2011 30*, 4, 1181–1187.

Hery, C., and Ramamoorthi, R. 2011. Importance sampling of reflections from hair fibers. Tech. rep., Pixar, December.

Jarosz, W., Carr, N. A., and Jensen, H. W. 2009. Importance sampling spherical harmonics. *Computer Graphics Forum 28*, 2 (Apr.), 577–586.

Kajiya, J. T., and Kay, T. L. 1989. Rendering fur with three dimensional textures. In *Computer Graphics (Proceedings of SIGGRAPH 89)*, 271–280.

Kulla, C., Imageworks, S., and Fajardo, M. 2011. Importance sampling of area lights in participating media. In *ACM SIGGRAPH 2011 Talks*. 55.

Lafortune, E. P. F., Foo, S.-C., Torrance, K. E., and Greenberg, D. P. 1997. Non-linear approximation of reflectance functions. In *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, 117–126.

Larson, G. J. W. 1992. Measuring and modeling anisotropic reflection. In *Computer Graphics (Proceedings of SIGGRAPH 92)*, 265–272.

Lawrence, J., Rusinkiewicz, S., and Ramamoorthi, R. 2004. Efficient brdf importance sampling using a factored representation. *ACM Transactions on Graphics 23*, 3 (Aug.), 496–505.

Marschner, S. R., Jensen, H. W., Cammarano, M., Worley, S., and Hanrahan, P. 2003. Light scattering from human hair fibers. *ACM Transactions on Graphics 22*, 3 (July), 780–791.

Moon, J. T., and Marschner, S. R. 2006. Simulating multiple scattering in hair using a photon mapping approach. *ACM Transactions on Graphics 25*, 3 (July), 1067–1074.

MOON, J. T., WALTER, B., AND MARSCHNER, S. 2008. Efficient multiple scattering in hair using spherical harmonics. *ACM Transactions on Graphics 27*, 3 (Aug.), 31:1–31:7.

NEULANDER, I. 2010. Fast furry ray gathering. In *ACM SIGGRAPH 2010 Talks*, ACM, 2.

PHARR, M., AND HUMPHREYS, G. 2010. *Physically Based Rendering, Second Edition: From Theory To Implementation*, 2nd ed. Morgan Kaufmann Publishers Inc.

PHONG, B. T. 1975. Illumination for computer generated pictures. *Commun. ACM 18* (June), 311–317.

PRESSA, W., TEUKOLSKY, S., VETTERLING, W., AND FLANNERY, B. 2007. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, The Edinburgh Building, Cambridge, UK.

REN, Z., ZHOU, K., LI, T., HUA, W., AND GUO, B. 2010. Interactive hair rendering under environment lighting. *ACM Transactions on Graphics 29*, 4 (July), 55:1–55:8.

SADEGHI, I., PRITCHETT, H., JENSEN, H. W., AND TAMSTORF, R. 2010. An artist friendly hair shading system. *ACM Transactions on Graphics 29*, 4 (July), 56:1–56:10.

VEACH, E., AND GUIBAS, L. J. 1995. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of SIGGRAPH 95*, Computer Graphics Proceedings, Annual Conference Series, 419–428.

WANG, R., AND ÅKERLUND, O. 2009. Bidirectional importance sampling for unstructured direct illumination. *Computer Graphics Forum 28*, 2 (Apr.), 269–278.

WARD, K., BERTAILS, F., KIM, T.-Y., MARSCHNER, S. R., CANI, M.-P., AND LIN, M. C. 2007. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics 13*, 213–234.

XU, K., MA, L.-Q., REN, B., WANG, R., AND HU, S.-M. 2011. Interactive hair rendering and appearance editing under environment lighting. *ACM Trans. Graph. 30* (Dec.), 173:1–173:10.

ZINKE, A., AND WEBER, A. 2007. Light scattering from filaments. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (Mar./Apr.), 342–356.

ZINKE, A., YUKSEL, C., WEBER, A., AND KEYSER, J. 2008. Dual scattering approximation for fast multiple scattering in hair. *ACM Transactions on Graphics 27*, 3 (Aug.), 32:1–32:10.

A Box-Muller Transform

Although Box-Muller transform is easy to implement and works well for sampling Gaussian function in general, it has a major shortcoming for our specific problem. By definition, the incoming longitudinal angle θ_i has a valid range of $[-\pi/2, \pi/2]$. Ignoring offset α for brevity, θ_h has a valid range of $[\frac{-\pi/2+\theta_r}{2}, \frac{\pi/2+\theta_r}{2}]$ (red shaded area in Figure 8.a-b). Since Box-Muller transform generates samples from $(-\infty, \infty)$, many samples generated fall out of the valid range. Figure 8.c shows a possible θ_h sample that is out of the valid range. These invalid samples can be handled by rejection sampling but incurs the cost of wasted samples. Attempts to keep the samples are complicated and prone to error (see discussion in Section B). Moreover, Box-Muller transform does not provide a way to compute the Gaussian integral over finite intervals (required for normalizing the *pdf*).

While dealing with edge cases is inevitable and complicated using Box-Muller transform, our approach does not have this short coming because it is based on an inverse CDF technique that can sample θ_i directly. As a result, we can ensure both θ_i and θ_h always fall into valid range.

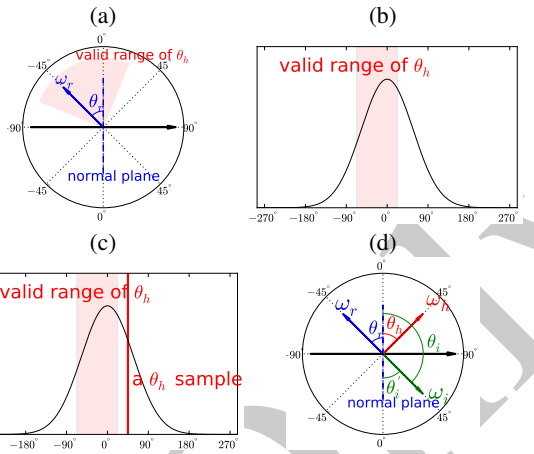

Figure 8: *(a). For case $\theta_r = -\frac{\pi}{4}$, θ_h has a valid range $[-\frac{3\pi}{8}, \frac{\pi}{8}]$ in order to make sure $\theta_i \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. (b). The valid range of θ_h only cover a portion of the entire Gaussian distribution. (c) Box-Muller transform draw samples from interval $(-\infty, \infty)$, some sample will end up outside the valid range. In this case, $\theta_h = \frac{3\pi}{4}$. (d). As a result, $\theta_i = 2\theta_h - \theta_r = \frac{3\pi}{4}$, which is not within valid range. In [Hery and Ramamoorthi 2011], this is handled by setting $\theta_i = \frac{\pi}{2} - \theta_i = \frac{\pi}{4}$. However, this also changed θ_h from $\frac{\pi}{4}$ to 0, causing inconsistency between* pdf, bsdf *value and the sampling direction ω_i.*

B Comparison to [Hery and Ramamoorthi 2011]

Hery and Ramamoorthi introduced a method to importance sample the R lobe of hair using the Box-Muller transform[Hery and Ramamoorthi 2011]. They dealt with out of range samples as follows:

1. Clamp θ_h by θ_{max};
2. If θ_i is outside of $[-\pi/2, \pi/2]$, flip θ_i about the vector $\mathbf{u}$ to ensure it falls within $[-\pi/2, \pi/2]$.

There are several problems with their approach. First of all, clamping θ_h introduces some amount of bias. More importantly, the way they handle the edge case where θ_i is out of valid range $[-\pi/2, \pi/2]$ is not mathematically correct. When θ_i is flipped about vector $\mathbf{u}$ (pseudocode page 5 line 15), the sampled direction is changed, but the *pdf* is not adjusted to reflect the altered direction. This introduces inconsistency between the sampled scattering direction ω_i and the computed *pdf* (Figure 8.d).

Furthermore, when computing the samples' *pdf* for multiple importance sampling, Hery and Ramamoorthi ignored the parts of the Gaussian outside of the valid range (pseudocode on page 12). As a result the computed *pdf* does not integrate to one. This can be verified by simple numerical tests.

We have implemented Hery and Ramamoorthi's algorithm, please see our supplemental material for rendering comparisons between our two approaches.

3 Derivations

3.1 Derivation of Longitudinal Sampling

Given a uniform random variable ξ in $[0, 1)$, we want to draw a sample of θ_i from the *pdf*

$$p(\theta_i) \propto \left[\frac{\beta}{(\frac{\theta_i+\theta_r}{2}-\alpha)^2+\beta^2}\right]\frac{1}{\cos\theta_i}$$

The $1/\cos\theta_i$ term is the correcting factor when transforming integrals over solid angle into integrals over spherical coordinates. The normalization gives

$$\begin{aligned}&\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} c\left[\frac{\beta}{(\frac{\theta_i+\theta_r}{2}-\alpha)^2+\beta^2}\right]\frac{1}{\cos\theta_i}\cos\theta_i d\theta_i\\ =\;& 2c\tan^{-1}\left(\frac{\theta_i-\alpha}{\beta}\right)\Big|_{\frac{-\pi/2+\theta_r}{2}}^{\frac{\pi/2+\theta_r}{2}} = 1\end{aligned}$$

Therefore $c = \frac{1}{2(A-B)}$, where $A = \tan^{-1}\Big(\frac{\frac{\pi/2+\theta_r}{2}-\alpha}{\beta}\Big)$ and $B = \tan^{-1}\Big(\frac{\frac{-\pi/2+\theta_r}{2}-\alpha}{\beta}\Big)$. The *pdf* of θ_i is:

$$p(\theta_i) = \frac{1}{2\cos\theta_i(A-B)}\frac{\beta}{\Big(\frac{\theta_i+\theta_r}{2}-\alpha\Big)^2+\beta^2}$$

The *cdf* can be computed by integrating the *pdf*

$$\begin{aligned}P(\theta_i) &= \int_{-\frac{\pi}{2}}^{\theta_i} c\left[\frac{\beta}{(\frac{\theta_i'+\theta_r}{2}-\alpha)^2+\beta^2}\right]\frac{1}{\cos\theta_i'}\cos\theta_i' d\theta_i'\\ &= \frac{\tan^{-1}\Big(\frac{\frac{\theta_i+\theta_r}{2}-\alpha}{\beta}\Big)-B}{A-B}\end{aligned}$$

By inverting the *cdf*, we sample θ_i, given a uniform random variable ξ from $[0, 1)$, as

$$\theta_i = 2\beta\tan(\xi(A-B)+B)+2\alpha-\theta_r$$

3.2 Derivation of N_{R} and $N_{\mathsf{TRT\text{-}g}}$ Sampling

Given a uniform random variable ξ from $[0, 1)$, we want to draw a sample of ϕ from the *pdf*

$$p(\phi) \propto \cos\frac{\phi}{2}$$

The normalization gives that

$$\int_{-\pi}^{\pi} c\cos\frac{\phi}{2}d\phi = c\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} 2\cos x dx = 2c\sin x\Big|_{-\frac{\pi}{2}}^{\frac{\pi}{2}} = 4c = 1$$

Therefore, $c = 1/4$. The *pdf* of ϕ is

$$p(\phi) = \frac{1}{4}\cos\frac{\phi}{2}$$

The *cdf* can be computed by integrating the *pdf*

$$\int_{-\pi}^{\phi}\frac{1}{4}\cos\frac{\phi'}{2}d\phi = \frac{1}{2}\sin x\Big|_{-\frac{\pi}{2}}^{\frac{\phi}{2}} = \frac{1}{2}\Big(\sin\frac{\phi}{2}+1\Big)$$

By inverting the *cdf*, we sample ϕ, given a uniform random variable ξ_2 from $[0, 1)$, as

$$\phi = 2\sin^{-1}(2\xi_2-1)$$

Then we can compute $\phi_i = \phi_r - \phi$. We have to transform the *pdf* $p(\phi)$ to $p(\phi_i)$ and it can be proved that $p(\phi_i) = \Big|\frac{d\phi_i}{d\phi}\Big|^{-1} p(\phi) = p(\phi)$

3.3 Derivation of N_{TT} Sampling

Given a uniform random variable ξ from $[0, 1)$, we want to draw a sample of ϕ from the *pdf*

$$p(\phi) \propto \frac{\gamma_{\mathrm{TT}}}{(\phi - \pi)^2 + \gamma_{\mathrm{TT}}^2}$$

The normalization gives that

$$\int_0^{2\pi} c \left[\frac{\gamma_{\mathrm{TT}}}{(\phi - \pi)^2 + \gamma_{\mathrm{TT}}^2} \right] d\phi = c \left[\tan^{-1} \left(\frac{\phi - \pi}{\gamma_{\mathrm{TT}}} \right) \right] \Bigg|_0^{2\pi} = 1$$

Therefore $c = \frac{1}{C_{\mathrm{TT}}}$ where $C_{\mathrm{TT}} = 2 \tan^{-1} (\pi / \gamma_{\mathrm{TT}})$. Then we can compute the *pdf* of ϕ

$$p(\phi) = \frac{1}{C_{\mathrm{TT}}} \left[\frac{\gamma_{\mathrm{TT}}}{(\phi - \pi)^2 + \gamma_{\mathrm{TT}}^2} \right]$$

The *cdf* can be computed by integrating the *pdf*

$$\begin{aligned} \int_0^{\phi} c \left[\frac{\gamma_{\mathrm{TT}}}{(\phi' - \pi)^2 + \gamma_{\mathrm{TT}}^2} \right] d\phi' &= \frac{1}{C_{\mathrm{TT}}} \left[\tan^{-1} \left(\frac{\phi' - \pi}{\gamma_{\mathrm{TT}}} \right) \right] \Bigg|_0^{\phi} \\ &= \frac{\tan^{-1} \left(\frac{\phi - \pi}{\gamma_{\mathrm{TT}}} \right)}{C_{\mathrm{TT}}} + \frac{1}{2} \end{aligned}$$

By inverting the *cdf*, we sample ϕ, given a uniform random variable ξ from $[0, 1)$, as

$$\phi = \gamma_{\mathrm{TT}} \tan \left[C_{\mathrm{TT}} \left(\xi - \frac{1}{2} \right) \right] + \pi$$

Then we can compute $\phi_i = \phi_r - \phi$ and $p(\phi_i) = p(\phi)$

3.4 Derivation of N_{g} Sampling

Given a uniform random variable ξ from $[0, 1)$, we want to draw samples of ϕ from the *pdf*

$$p(\phi) \propto \frac{\gamma_{\mathrm{g}}}{(|\phi| - \phi_{\mathrm{g}})^2 + \gamma_{\mathrm{g}}^2}$$

We first use ξ to randomly pick a half of the lobe and remap the random variable ξ_2 back to $[0, 1)$. Specifically, for $\xi < 1/2$, we set ϕ positive and map $\xi \leftarrow 2\xi$. For $\xi \geq 1/2$, we set ϕ negative and map $\xi \leftarrow 2(1 - \xi)$. Then we sample ϕ in the domain $[0, \pi/2)$. The normalization gives

$$\int_0^{\pi/2} c \left[\frac{\gamma_{\mathrm{g}}}{(|\phi| - \phi_{\mathrm{g}})^2 + \gamma_{\mathrm{g}}^2} \right] d\phi = c \left[\tan^{-1} \left(\frac{\phi - \phi_{\mathrm{g}}}{\gamma_{\mathrm{g}}} \right) \right] \Bigg|_0^{\pi/2} = 1$$

Therefore $c = \frac{1}{C_{\mathrm{g}} - D_{\mathrm{g}}}$ where $C_{\mathrm{g}} = \tan^{-1} \left(\frac{\pi/2 - \phi_{\mathrm{g}}}{\gamma_{\mathrm{g}}} \right)$ and $D_{\mathrm{g}} = \tan^{-1} \left(\frac{-\phi_{\mathrm{g}}}{\gamma_{\mathrm{g}}} \right)$. we can compute the *pdf* of ϕ

$$p(\phi) = \frac{1}{C_{\mathrm{g}} - D_{\mathrm{g}}} \left[\frac{\gamma_{\mathrm{g}}}{(\phi - \phi_{\mathrm{g}})^2 + \gamma_{\mathrm{g}}^2} \right]$$

The *cdf* can be computed by integrating the *pdf*

$$\begin{aligned} \int_0^{\phi} c \left[\frac{\gamma_{\mathrm{g}}}{(\phi' - \phi_{\mathrm{g}})^2 + \gamma_{\mathrm{g}}^2} \right] &= \frac{1}{C_{\mathrm{g}} - D_{\mathrm{g}}} \left[\tan^{-1} \left(\frac{\phi' - \phi_{\mathrm{g}}}{\gamma_{\mathrm{g}}} \right) \right] \Bigg|_0^{\phi} \\ &= \frac{\tan^{-1} \left(\frac{\phi - \phi_{\mathrm{g}}}{\gamma_{\mathrm{g}}} \right) - D_{\mathrm{g}}}{C_{\mathrm{g}} - D_{\mathrm{g}}} \end{aligned}$$

We sample ϕ, given a uniform random variable ξ from $[0, 1)$, as

$$\phi = \gamma_{\mathrm{g}} \tan(\xi(C_{\mathrm{g}} - D_{\mathrm{g}}) + D_{\mathrm{g}}) + \phi_{\mathrm{g}}$$

Then we can compute $\phi_i = \phi_r \pm \phi$. The sign of ϕ is determined by the value of the original random variable ξ before remapping. We also transform the *pdf* to account the remapping of the random variable.

$$p(\phi_i) = \frac{1}{2} p(|\phi|) = \frac{1}{2(C_{\mathrm{g}} - D_{\mathrm{g}})} \left[\frac{\gamma_{\mathrm{g}}}{(|\phi_r - \phi_i| - \phi_g)^2 + \gamma_{\mathrm{g}}^2} \right]$$

4 Rendering Comparison to [Hery and Ramamoorthi 2011]

In Appendix B of the paper, we discussed some protential problems of Hery and Ramamoorthi's solution for hair reflectance importance sampling [Hery and Ramamoorthi 2011]. We implemented their approach based on the pseudocode in their paper and compared it to our method using scenes in our paper. Note that [Hery and Ramamoorthi 2011] did not provide solutions for sampling lobes other than the R lobes. To make an fair comparison, we only rendered the R lobe in all the examples.

Figure 2 shows the rendering result of [Hery and Ramamoorthi 2011] and our method. The error images shows that Hery and Ramamoorthi's method generate extra energy at grazing angles. Figure 1 is the error plots of area light scene in Figure 2. As the sample count increases, our method constantly produces lower error compare to [Hery and Ramamoorthi 2011].

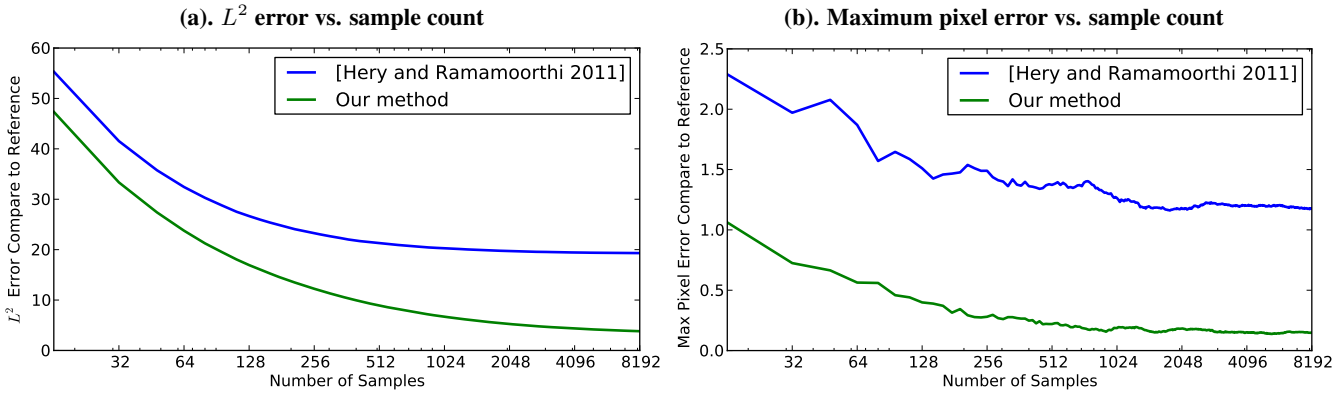

Figure 1: *L^2 error plots of area light scene in Figure 2 rendered using Hery and Ramamoorthi's method and our method. Error is measured as the differences in comparison to the reference image. As the sample count increases our method consistently achieves lower error than Hery and Ramamoorthi's method.*

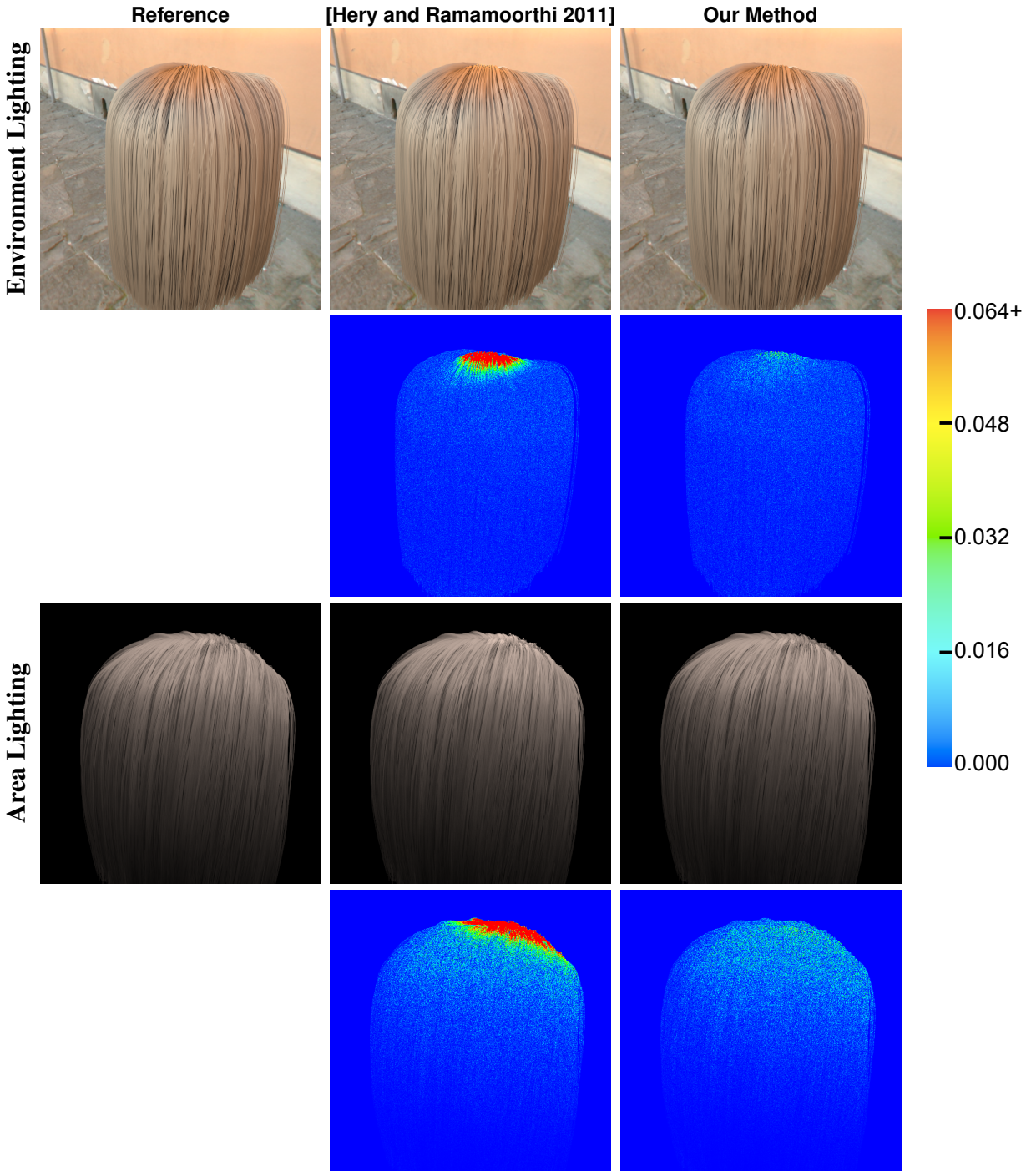

Figure 2: *Error Images of [Hery and Ramamoorthi 2011] and our method. The edge cases of Box-Muller transform are not correctly handled in [Hery and Ramamoorthi 2011], resulting in incorrect energy estimation at grazing angles. The images will not converge to correct solution as sample count increases. (The error images are computed using per-pixel L^2 difference.)*

5 Pseudo Code

Here is the *Python* pseudo code for our importance sampling algorithm. To keep the code simple, we used the simple uniform lobe selection instead of the energy-based lobe selection. Moreover, it is done without amortizing the cost of constants computation.

```
# uv - a pair of uniform random variable in [0,1]
# I  - viewing direction
# L  - light direction
# beta_R, beta_TT, beta_TRT - width of longitudinal gaussian
# alpha_R, alpha_TT, alpha_TRT - offset of longitudinal gaussian
# gamma_G - width of glint
# gamma_TT - width of transmission
# phi_g - offset of glint

pi = 3.1415926

# sample the primary lobe
def sample_R_lobe(uv, I):
    (theta_r, phi_r) = compute_angle(I)

    a_R = arctan(((pi/2 + theta_r)/2 - alpha_R) / beta_R)
    b_R = arctan(((-pi/2 + theta_r)/2 - alpha_R) / beta_R)

    t = beta_R * tan(uv[0] * (a_R - b_R) + b_R)
    theta_h = t + alpha_R
    theta_i = (2 * theta_h - theta_r)

    phi = 2 * arcsin(1 - 2 * uv[2])
    phi_i = phi_r - phi
    phi_pdf = cos(phi/2) / 4

    return compute_direction(theta_i, phi_i)

# sample the transmission lobe
def sample_TT_lobe(uv, I):
    (theta_r, phi_r) = compute_angle(I)

    a_TT = arctan(((pi/2 + theta_r)/2 - alpha_TT) / beta_TT)
    b_TT = arctan(((-pi/2 + theta_r)/2 - alpha_TT) / beta_TT)
    c_TT = 2 * arctan(pi/ 2 / gamma_TT);

    t = beta_TT * tan(uv[0] * (a_TT - b_TT) + b_TT)
    theta_h = t + alpha_TT
    theta_i = (2 * theta_h - theta_r)

    double p = gamma_TT * tan((v - 0.5) * c_TT)
    double phi = p + pi
    double phi_i = phi_r - phi

    return compute_direction(theta_i, phi_i)

# sample the secondary highlight lobe
def sample_TRT_G_lobe(uv, I):
    (theta_r, phi_r) = compute_angle(I)

    a_TRT = arctan(((pi/2 + theta_r)/2 - alpha_TRT) / beta_TRT)
    b_TRT = arctan(((-pi/2 + theta_r)/2 - alpha_TRT) / beta_TRT)

    t = beta_TRT * tan(uv[0] * (a_TRT - b_TRT) + b_TRT)
    theta_h = t + alpha_TRT
    theta_i = (2 * theta_h - theta_r)

    phi = 2 * arcsin(1 - 2 * uv[2])
    phi_i = phi_r - phi
    phi_pdf = cos(phi/2) / 4

    return compute_direction(theta_i, phi_i)
```

```
# sample the glint lobe
def sample_G_lobe(uv, I):
    (theta_r, phi_r) = compute_angle(I)

    a_TRT = arctan(((pi/2 + theta_r)/2 - alpha_TRT) / beta_TRT)
    b_TRT = arctan(((-pi/2 + theta_r)/2 - alpha_TRT) / beta_TRT)
    c_G = atan((pi/2 - phi_g) / gamma_G)
    d_G = atan(-phi_g / gamma_G)

    t = beta_TRT * tan(uv[0] * (a_TRT - b_TRT) + b_TRT)
    theta_h = t + alpha_TRT
    theta_i = (2 * theta_h - theta_r)

    if(uv[1] < 0.5):
        uv[1] = 2 * uv[1]
        sign = 1
    else:
        uv[1] = 2 * (1 - uv[1])
        sign = -1

    p = gamma_G * tan(uv[1] * (c_G - d_G) + d_G)
    phi = sign * (p + phi_g)
    phi_i = phi_r - phi

    return compute_direction(theta_i, phi_i)

# compute the pdf of primary highlight
def compute_R_pdf(L, I):
    (theta_r, phi_r) = compute_angle(I)
    (theta_i, phi_i) = compute_angle(L)

    if(pi/2 - theta_i < epsilon):
        return 0

    a_R = arctan(((pi/2 + theta_r)/2 - alpha_R) / beta_R)
    b_R = arctan(((-pi/2 + theta_r)/2 - alpha_R) / beta_R)

    theta_h = (theta_i + theta_r) / 2
    t = theta_h - alpha_R
    theta_pdf = beta_R / (t*t + beta_R*beta_R) / (2*(a_R - b_R) * cos(theta_i))

    phi = phi_r - phi_i
    phi_pdf = cos(phi/2) / 4

    return theta_pdf * phi_pdf

# compute the pdf of transmission
def compute_TT_pdf(L, I):
    (theta_r, phi_r) = compute_angle(I)
    (theta_i, phi_i) = compute_angle(L)

    if(pi/2 - theta_i < epsilon):
        return 0

    a_TT = arctan(((pi/2 + theta_r)/2 - alpha_TT) / beta_TT)
    b_TT = arctan(((-pi/2 + theta_r)/2 - alpha_TT) / beta_TT)
    c_TT = 2 * arctan(pi/ 2 / gamma_TT);

    theta_h = (theta_i + theta_r) / 2
    t = theta_h - alpha_R
    theta_pdf = beta_R / (t*t + beta_R*beta_R) / (2*(a_R - b_R) * cos(theta_i))

    phi = abs(phi_r - phi_i)
    if phi < pi/2:
        phi_pdf = 0
```

```
    else:
        p = pi - phi
        phi_pdf = (gamma_TT / (p * p + gamma_TT * gamma_TT)) / c_TT

    return theta_pdf * phi_pdf

# compute the pdf of secondary highlight without glint
def compute_TRT_G_pdf(L, I):
    (theta_r, phi_r) = compute_angle(I)
    (theta_i, phi_i) = compute_angle(L)

    if(pi/2 - theta_i < epsilon):
        return 0

    a_TRT = arctan(((pi/2 + theta_r)/2 - alpha_TRT) / beta_TRT)
    b_TRT = arctan(((-pi/2 + theta_r)/2 - alpha_TRT) / beta_TRT)

    theta_h = (theta_i + theta_r) / 2
    t = theta_h - alpha_R
    theta_pdf = beta_R / (t*t + beta_R*beta_R) / (2*(a_R - b_R) * cos(theta_i))

    phi = phi_r - phi_i
    phi_pdf = cos(phi/2) / 4

    return theta_pdf * phi_pdf

# compute the pdf of glint term
def compute_G_pdf(L, I):
    (theta_r, phi_r) = compute_angle(I)
    (theta_i, phi_i) = compute_angle(L)

    if(pi/2 - theta_i < epsilon):
        return 0

    a_TRT = arctan(((pi/2 + theta_r)/2 - alpha_TRT) / beta_TRT)
    b_TRT = arctan(((-pi/2 + theta_r)/2 - alpha_TRT) / beta_TRT)
    c_G = arctan((pi/2 - phi_g) / gamma_G)
    d_G = arctan(-phi_g / gamma_G)

    theta_h = (theta_i + theta_r) / 2
    t = theta_h - alpha_R
    theta_pdf = beta_R / (t*t + beta_R*beta_R) / (2*(a_R - b_R) * cos(theta_i))

    phi = abs(phi_r - phi_i)
    p = phi - phi_g
    phi_pdf = gamma_G / (p*p + gamma_G * gamma_G) / (2 * (c_G - d_G))

    return theta_pdf * phi_pdf

def compute_pdf(L, I):
    pdf_R = compute_R_pdf(L, I)
    pdf_TT = compute_TT_pdf(L, I)
    pdf_TRT_G = compute_TRT_G_pdf(L, I)
    pdf_G = compute_G_pdf(L, I)
    return (pdf_R + pdf_TT + pdf_TRT_G + pdf_G) / 4

def sample_brdf(uv, I):
    if uv[0] < 0.5 and uv[1] < 0.5:
        # Sample R lobe
        uv[0] = 2 * uv[0]
        uv[1] = 2 * vv[1]
        L = sample_R_lobe(uv, I)
    elif u >= 0.5 and v < 0.5:
        # Sample TT lobe
        uv[0] = 2 * (1 - uv[0])
        uv[1] = 2 * uv[1]
```

```
    L = sample_TT_lobe(uv, I)
elif u < 0.5 and v >= 0.5:
    # Sample TRT-G lobe
    uv[0] = 2 * uv[0]
    uv[1] = 2 * (1 - uv[1])
    L = sample_TRT_G_lobe(uv, I)
else:
    # Sample glint lobe
    uv[0] = 2 * (1 - uv[0])
    uv[1] = 2 * (1 - uv[1])
    L = sample_G_lobe(uv, I)
pdf = compute_pdf(L, I)
return (L, pdf)
```

References

HERY, C., AND RAMAMOORTHI, R. 2011. Importance sampling of reflections from hair fibers. Tech. rep., Pixar, December.