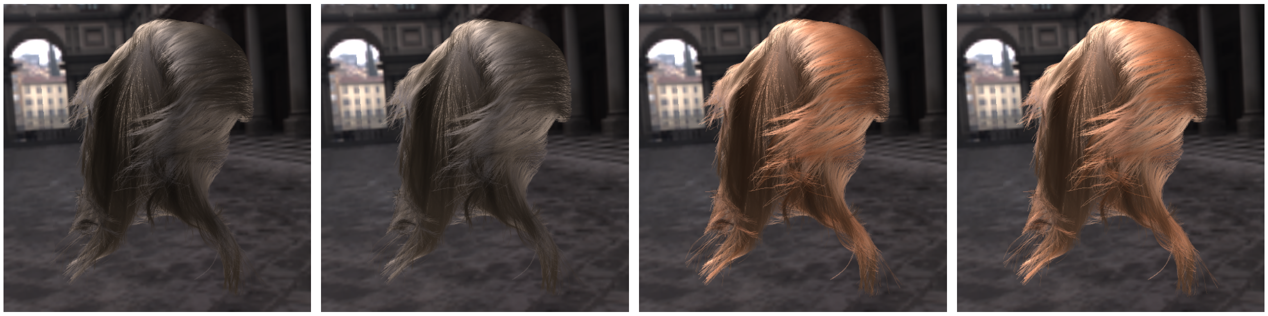
Interactive Hair Rendering Under Environment Lighting

Zhong Ren*† Kun Zhou* Tengfei Li* Wei Hua* Baining Guo†

* State Key Lab of CAD&CG, Zhejiang University † Microsoft Research Asia



(a) our single scattering result, 16.2fps (b) single scattering reference, 2.1 minutes (c) our single/multiple scattering result, 7.9fps (d) single/multiple scattering reference, 15.2 minutes

Figure 1: *Interactive hair rendering under environment lighting with both single and multiple scattering effects. The hair model consists of 10,000 fibers and 270K line segments. The reference images (b) and (d) are generated by summing up the contributions of all* $32 \times 32 \times 6$ *directional lights of the environment map, using the deep opacity map algorithm [Yuksel and Keyser 2008] and the dual scattering technique [Zinke et al. 2008], respectively. In our algorithm, the environment light is approximated with 49 SRBFs.*

Abstract

We present an algorithm for interactive hair rendering with both single and multiple scattering effects under complex environment lighting. The outgoing radiance due to single scattering is determined by the integral of the product of the environment lighting, the scattering function, and the transmittance that accounts for self-shadowing among hair fibers. We approximate the environment light by a set of spherical radial basis functions (SRBFs) and thus convert the outgoing radiance integral into the sum of radiance contributions of all SRBF lights. For each SRBF light, we factor out the effective transmittance to represent the radiance integral as the product of two terms: the transmittance and the convolution of the SRBF light and the scattering function. Observing that the convolution term is independent of the hair geometry, we precompute it for commonly-used scattering models, and reduce the run-time computation to table lookups. We further propose a technique, called the *convolution optical depth map*, to efficiently approximate the effective transmittance by filtering the optical depth maps generated at the center of the SRBF using a depth-dependent kernel. As for the multiple scattering computation, we handle SRBF lights by using similar factorization and precomputation schemes, and adopt sparse sampling and interpolation to speed up the computation. Compared to off-line algorithms, our algorithm can generate images of comparable quality, but at interactive frame rates.

Keywords: single scattering, multiple scattering, SRBF lights, convolution optical depth map, stochastic simplification

1 Introduction

Environment lighting is an effective way to model the natural illumination usually found in the real world. Over the past few years, much work has been devoted to interactive rendering under environment lighting, inspired by [Ramamoorthi and Hanrahan 2001; Sloan et al. 2002]. Despite the tremendous progress, one problem that remains unsolved is that of hair rendering under environment lighting. Existing interactive hair rendering techniques are all developed for relatively simple light sources, such as point and directional lights [Ward et al. 2007]. A naive way to handle environment lighting would be to densely sample the environment light to yield a large number of directional lights, which could then be fed to fast rendering techniques such as [Yuksel and Keyser 2008; Zinke et al. 2008]. Unfortunately, this approach does not lead to fast rendering due to the costs of handling a prohibitively large number of lights.

A fast hair rendering algorithm for environment lighting must efficiently deal with two complexities. The first is the *light integration complexity*. With environment lighting, the incident radiance varies from direction to direction. At each point to be shaded, this radiance has to be multiplied with the hair fiber's scattering function, which can be a complex function by itself, and integrated over the sphere of all directions. The second is the *light transport complexity*, which manifests in the forms of self-shadowing [Lokovic and Veach 2000] and multiple scattering [Moon et al. 2008]. All these have to be taken into account for a realistic rendering of hair.

Our goal is to efficiently handle both light integration and light transport complexities under environment lighting so as to achieve realistic hair rendering at interactive frame rates. Consider first the light integration and self-shadowing computation for single scattering. The outgoing radiance due to each single scattering event is determined by the integral of the product of the environment lighting, the scattering function, and the transmittance that accounts for self-shadowing among hair fibers. We approximate the environment light by a set of SRBFs and thus convert the outgoing radiance integral into the sum of radiance contributions of all SRBF lights [Tsai and Shih 2006]. For each SRBF light, we factor out the effective transmittance to represent the radiance integral as the product of two terms: the transmittance and the convolution of the SRBF light and the scattering function. Observing that the convolution term is independent of the hair geometry, we precompute it for the two commonly-used scattering models, one by Kajiya and Kay [Kajiya and Kay 1989] and the other by Marschner et al. [Marschner et al. 2003], and reduce the run-time computation to table lookups. We further propose a technique, called the *convolution optical depth map*, to efficiently approximate the effective transmittance by filtering the optical depth maps generated at the center of the SRBF using a depth-dependent kernel.

For multiple scattering, the state-of-art interactive technique is the dual scattering algorithm [Zinke et al. 2008]. This algorithm, however, is designed to handle simple light sources and does not directly address the light integration and transport complexities under environment lighting. We follow the same strategy used for single scattering to decouple the two complexities. This makes it possible to use precomputation schemes to reduce the run-time integration to table queries, as well as to exploit the smoothness of the multiple scattered irradiance by sparse sampling and interpolation.

We first show that the multiple scattering contribution of each SRBF light can be represented as a product of two terms: a forward scattering attenuation term and a convolution term of the multiple scattered irradiance and the scattering function. Again, for a given scattering function, the convolution term can be precomputed into a table for a set of samples in the space of all possible SRBFs, view directions and forward scattering spread values. The values of the forward scattering attenuation and spread depend on the hair geometry and need to be determined on the fly. Observing that the multiple scattered irradiance has a smooth distribution in the hair volume, we compute the forward scattering attenuation and spread of the incident radiance only at a sparse set of points, typically numbered in several thousands. These attenuation and spread values are interpolated in the view pass for computing the outgoing radiance.

Our algorithm can render realistic results at interactive frame rates under dynamic environment lighting. Our algorithm is also highly flexible for supporting quality-speed tradeoffs. For dark-colored hair, the multiple scattering component is insignificant. In this case, our algorithm allows the user to skip multiple scattering calculations and achieve real time performance. To our knowledge, ours is the first hair rendering algorithm to achieve interactive frame rates under complex environment lighting. Fig. 1 shows an example of rendering results of our algorithm.

2 Related Work

There is a large body of research on hair rendering. See the survey by Ward et al. [2007] for a comprehensive review. Here we briefly review techniques that are most relevant to our work.

Single Fiber Scattering To model the scattering of light by a single hair fiber, Kajiya and Kay [1989] proposed a simple model consisting of a diffuse term and a specular term. The model has been widely used in games and movies due to its simplicity, although it fails to correctly predict the azimuthal dependence of the scattered intensity and its diffuse term often results in a flat hair appearance. Marschner et al. [2003] proposed a more sophisticated lighting model based on measurements of light scattering from a single hair fiber. It identifies the major modes of reflection that have the greatest influence on the fiber's scattering properties and takes into account the Fresnel factor and volume absorption to give a more convincing prediction of the fiber's reflectance. This model assumes both view and light sources are distant to the hair fiber. Zinke and Webber [2007] proposed a model that generalizes to near field scattering.

Self-Shadowing and Multiple Scattering To account for the self shadowing effects among hair fibers, Lokovic and Veach [2000] extended the shadow map idea and stored for each pixel a representation of the fractional visibility at all possible depths. This deep shadow map enables high quality hair self-shadows, albeit in an off-line setup. Kim and Neuman [2001] simplified the deep shadow map by rendering slices of the hair structure and accumulating the resulting opacity in an opacity shadow map, which is then queried in the final rendering pass to retrieve the opacity between the light source and the point being shaded. Their method achieves real-time performance, but only when a relatively small number of slices are used, and it is prone to layering artifacts. Mertens et al. [2004] modeled hair geometry as 3D density fields which are sampled on the fly using rasterization. The raster fragments are then k-means clustered to get a piecewise linear approximation of the fractional visibility function at each shadow map pixel. Sintorn and Assarsson [2008] proposed a fast GPU sorting algorithm which can be used to accelerate the generation of the opacity shadow map [Kim and Neumann 2001], as well as to allow transparency in the final viewing pass. Yuksel and Keyser [2008] generated a depth map of the hair, used it as the starting point to divide the hair geometry into a few layers, and approximated the fractional visibility of each pixel as a piecewise linear function according to the layers. Since the opacity layers are shaped according to the hair geometry, the interpolation between layers is guaranteed to be limited inside the hair volume. This eliminates the layering artifacts and allows fewer layers to be used. Recently, Sintorn and Assarsson [2009] proposed a more precise approximation of the visibility function and further improve the self-shadowing quality.

Recently several approximation algorithms have been proposed to make the simulation of multiple scattering among hair fibers more tractable. These include methods based on photon mapping [Moon and Marschner 2006] or spherical harmonics approximation of multiple scattered incident radiance [Moon et al. 2008] for off-line rendering and dual scattering approximation [Zinke et al. 2008; Zinke 2008] for real-time rendering. All existing interactive methods handle only simple light sources such as point or directional lights.

Environment Lighting Representations Tsai and Shih [2006] proposed to approximate environment lighting by a set of SRBFs. We adopt this lighting representation for real-time hair rendering. Unlike spherical harmonics [Sloan et al. 2002], the support of an SRBF is controllable and can capture high frequency illumination. Unlike wavelets [Ng et al. 2003], SRBF is a parameterized representation, making it easier to precompute its convolution with the scattering function as described in Section 4. An alternative is to use the importance sampling technique [Agarwal et al. 2003] to approximate the environment light as a set of directional lights. However, this approach often requires a large number of lights in order to get a good approximation (Fig. 6).

3 Algorithm Overview

In this section we introduce the terminology to be used in the paper and provide an overview of our rendering algorithm.

Fig. 2 illustrates the scattering geometry of a hair fiber. The plane perpendicular to the fiber is referred to as the *normal plane*. The inclination of the light direction ω_i and view direction ω_o with respect to the normal plane are denoted by θ_i and θ_o. The azimuths around the fiber are denoted by ϕ_i and ϕ_o, respectively.

In the following, we separately discuss the single scattering and multiple scattering computations, as they are often handled with different techniques [Moon et al. 2008; Zinke et al. 2008].

3.1 Single Scattering

Following the definition of curved intensity in [Marschner et al. 2003], we write the single scattering integral under incident environment lighting $L(\omega_i)$ as follows:

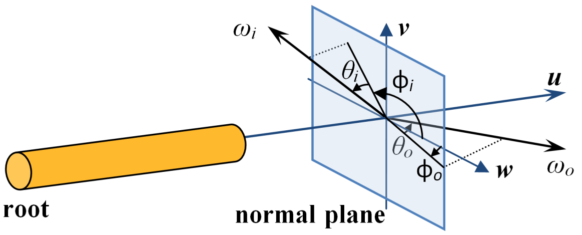


Figure 2: *Hair fiber scattering geometry.*

$$L(\omega_o) = D \int_\Omega L(\omega_i) T(\omega_i) S(\omega_i, \omega_o) \cos\theta_i \, d\omega_i, \tag{1}$$

where $L(\omega_o)$ is the outgoing curved intensity (hereafter referred to as the outgoing intensity) to the view direction and $L(\omega_i)$ is the incoming radiance. D is the diameter of the hair fiber, Ω is the unit sphere of incident directions. The bidirectional scattering function $S(\omega_i, \omega_o)$ is the counterpart of the bidirectional reflection distribution function (BRDF) in surface reflectance. It describes the infinitesimal outgoing intensity dL generated by scattering of the infinitesimal incident curved irradiance dE and is determined by the fiber's scattering properties. $T(x, \omega_i)$ is the transmittance in the incident direction generated by the hair volume, given by

$$T(x, \omega_i) = \exp\left(-\sigma_a \int_x^{\infty_{\omega_i}} \rho(x)\, dx\right),$$

where $\rho(x)$ is the density function and the integral is along the semi-infinite ray from x in the ω_i direction. The value of $\rho(x)$ is 1 if x is covered by a hair fiber and 0 otherwise.

For notational simplicity, we have dropped in Eq. (1) the dependence on spatial location x for the transmittance and outgoing intensity. These two quantities would be treated separately for each location anyway in the following.

We approximate the environment lighting by a set of SRBFs:

$$L(\omega_i) \simeq \sum_{j=1}^{N} L_j G(\omega_i; \xi_j, \lambda_j). \tag{2}$$

Here $G(\omega_i; \xi_j, \lambda_j) = \exp(-\lambda_j)\exp(\lambda_j(\omega_i \cdot \xi_j))$ is the Gaussian SRBF kernel centered at ξ_j with bandwidth parameter λ_j. In the following we will simply use $G_j(\omega)$ to denote the SRBF.

Substituting the SRBF representation of incident lighting into Eq. (1), we get

$$L(\omega_o) \simeq D \sum_{j=1}^{N} L_j \int_\Omega G_j(\omega_i) T(\omega_i) S(\omega_i, \omega_o) \cos\theta_i \, d\omega_i. \tag{3}$$

In words, the total outgoing intensity is a weighted sum of the outgoing intensities of all SRBF lights.

We can further rewrite the integral by introducing the *effective transmittance* $\tilde{T}$:

$$L(\omega_o) \simeq D \sum_{j=1}^{N} L_j \tilde{T}(\xi_j, \lambda_j) \int_\Omega G_j(\omega_i) S(\omega_i, \omega_o) \cos\theta_i \, d\omega_i. \tag{4}$$

Here the effective transmittance $\tilde{T}(\xi_j, \lambda_j)$ is the average attenuation of the SRBF lighting j. $\tilde{T}$ should be evaluated as

$$\tilde{T}(\xi_j, \lambda_j) = \frac{\int_\Omega G_j(\omega_i) T(\omega_i) S(\omega_i, \omega_o) \cos\theta_i \, d\omega_i}{\int_\Omega G_j(\omega_i) S(\omega_i, \omega_o) \cos\theta_i \, d\omega_i}. \tag{5}$$

Now we have factorized the outgoing intensity of each SRBF light into two terms: one for the convolution of a single SRBF with the scattering function and the other for the effective transmittance. The convolution term is independent of the hair geometry. Thus for a given scattering function $S(\omega_i, \omega_o)$, we can precompute the convolution term for a set of samples in the space spanned by all possible SRBFs, view directions and bandwidth parameters. At runtime, the computation of the convolution is reduced to table queries.

As for the effective transmittance, its exact evaluation according to Eq. (5) is as expensive as the original integral. We propose to approximate the effective transmittance by exponentiating the optical depth average at x over the directions covered by the SRBF light using a convolution optical depth map.

3.2 Multiple Scattering

Following the dual scattering technique [Zinke et al. 2008], the multiple scattering component of the outgoing intensity under the environment lighting, denoted as $L^D(\omega_o)$, can be written as:

$$L^D(\omega_o) \simeq D \sum_{j=1}^{N} L_j \int_\Omega L_j^G(\omega_i; \xi_j, \lambda_j) S^D(\omega_i, \omega_o) \cos\theta_i \, d\omega_i, \tag{6}$$

where $S^D(\omega_i, \omega_o)$ is the Bidirectional Curves Scattering Distribution Function (BCSDF), obtained by summing up the bidirectional scattering function S and the backward scattering function, which accounts for the multiple scattering events within the neighborhood of x:

$$S^D(\omega_i, \omega_o) = S(\omega_i, \omega_o) + d_b S_{back}(\omega_i, \omega_o),$$

where d_b is the constant density factor, typically set to a value between 0.6 and 0.8 for most human hair styles.

$L_j^G(\omega_i; \xi_j, \lambda_j)$ is the global multiple scattered irradiance corresponding to SRBF light j. Because of the wide azimuthal scattering of hair fibers, it quickly becomes near isotropic in the azimuthal direction after only a few scattering events. Similar to [Zinke et al. 2008], we can represent it by:

$$\begin{aligned} L^G(\omega_i; \xi, \lambda) &= T_f(\xi)\psi_f(\xi, \omega_i, \sigma_f(\xi, \lambda)) \\ &= T_f(\xi)\tilde{s}_f(\phi_\xi, \phi_i) g(\theta_i - \theta_\xi, \bar{\sigma}_f^2(\xi, \lambda))/\cos\theta_\xi, \end{aligned}$$

where $\psi_f(\xi, \omega_i, \sigma_f(\xi, \lambda))$ is the spread function that approximates the final angular distribution of front scattered irradiance, θ_ξ, ϕ_ξ are the inclination and azimuth of ξ, and $\tilde{s}_f(\phi_\xi, \phi_i)$ is $1/\pi$ for forward scattering directions and zero for backward scattering directions. The variance of the first scattering event depends on the SRBF bandwidth parameter λ, as does $\bar{\sigma}_f^2$. The total attenuation $T_f(\xi)$ is obtained by multiplication of the average attenuation of each forward scattering event along the shadow path, and the total spread of forward scattering $\bar{\sigma}_f^2(\xi)$ is obtained by summing up the variances due to each of such events. g is the normalized Gaussian distribution.

Eq. (6) can thus be rewritten as:

$$L^D(\omega_o) \simeq D \sum_{j=1}^{N} L_j T_f \int_\Omega \psi_f(\xi_j, \omega_i, \sigma_f) S^D(\omega_i, \omega_o) \cos\theta_i \, d\omega_i. \tag{7}$$

Note that T_f and $\bar{\sigma}_f$ are evaluated with respect to SRBF center ξ_j. For notational simplicity, we omit the argument ξ_j for both of them.

In words, the effect of multiple scattering in hair is factorized into two parts. The global forward scattering part models the attenuation and spread of the irradiance due to scattering events occurring along the shadow path. And the local backward scattering part models the scattering events within the neighborhood of x. It is described by the backward scattering function $S_{back}(\omega_i, \omega_o)$ which depends only on the fiber properties.

Unlike the convolution in the single scattering case (Eq. (4)), the convolution term in Eq. (7) depends on the hair geometry because of the forward scattering spread, $\bar{\sigma}_f$. Nonetheless, we can still precompute the convolution term for a set of samples in the space spanned by all possible SRBFs, view directions and forward scattering spread values, and turn the run-time convolution computation into table queries.

Now the remaining problem is the efficient evaluation of the forward scattering transmittance T_f and spread $\bar{\sigma}_f$. We exploit the smooth distribution of the multiple scattered irradiance in the hair volume, an assumption that has been commonly accepted in hair rendering [Moon and Marschner 2006; Moon et al. 2008], and evaluate these terms only at a small number of sample points. Then for an arbitrary shading point x, we interpolate the values of T_f and $\bar{\sigma}_f$ from these samples, and use the results to query the pre-computed integration tables and compute $L^D(\omega_o)$.

In the following, we present computation details of the convolution term (Section 4), the effective transmittance term (Section 5), and the multiple scattering (Section 6).

4 Convolving SRBF and Scattering Function

In this section we describe how to compute the convolution of a SRBF light with the scattering function:

$$I(\omega_o;\xi,\lambda) = \int_\Omega G(\omega_i;\xi,\lambda)S(\omega_i,\omega_o)\cos\theta_i \, d\omega_i. \tag{8}$$

Here we omit the SRBF index j to emphasize that this computation can be performed for an arbitrary SRBF light.

At first glance, the samples to be computed appear to lie in a 5D space, with 2D for the SRBF center ξ, 1D for the bandwidth parameter λ and 2D for the outgoing direction ω_o. Fortunately, we can leverage the symmetry of the scattering function to reduce the dimensionality of this sample space.

Marschner et al. [2003] decomposes the bidirectional scattering function into three major components, each of which is further factorized into a longitudinal function and an azimuthal function:

$$S(\omega_i,\omega_o) = \sum_t M_t(\theta_h)N_t(\theta_d,\phi), \tag{9}$$

where $t \in \{R, TT, TRT\}$ corresponds respectively to the three major modes of fiber scattering, i.e., reflection (R), transmission-transmission (TT) and transmission-reflection-transmission (TRT). The longitudinal function and azimuthal function are written in terms of the half angle $\theta_h = (\theta_i + \theta_o)/2$, difference angle $\theta_d = (\theta_i - \theta_o)/2$ and relative azimuth $\phi = \phi_o - \phi_i$ to emphasize the natural symmetry of S.

Since the integration in Eq. (8) only depends on the relative azimuth ϕ instead of the two azimuth angles, we can precompute the convolution for the Marschner model as a 4D table $\mathbf{I}_M(\cos\theta_\xi, \cos\theta_o, \cos(\phi_\xi - \phi_o), 1/\lambda)$. Here we use $1/\lambda$ instead of λ for tabulation because it is proportional to the width of the SRBF and has a more linear relationship with the convolution. The cosines are used for the table index to avoid runtime computation of inverse trigonometric functions. Fig. 3 shows typical slices of $\mathbf{I}_M$.

At runtime, $\cos\theta_\xi$, $\cos\theta_o$ and $\cos(\phi_\xi - \phi_o)$ are computed from the SRBF center ξ and outgoing direction ω_o, and combined with $1/\lambda$ to form the quadruple to retrieve the convolution values.

The Kajiya and Kay model [1989] is much simpler. It consists of a diffuse and a specular component:

$$S(\omega_i,\omega_o) = K_d + K_s\cos^p(\theta_i+\theta_o)/\cos\theta_i.$$

The diffuse component of the convolution integral

$$I_d(\omega_o;\xi,\lambda) = \int_\Omega G(\omega_i;\xi,\lambda)\cos\theta_i \, d\omega_i$$

is independent of ω_o and the azimuth angle ϕ of the SRBF center ξ, and thus can be precomputed as a 2D table $\mathbf{I}_d(\cos\theta_\xi, 1/\lambda)$.

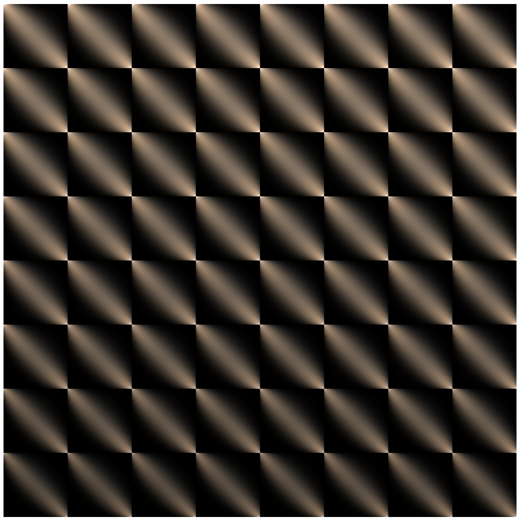


Figure 3: *A typical slice ($\phi = 120^\circ$) of the 4D convolution integral $\mathbf{I}_M$ (the Marschner model). Each little square is a 2D slice corresponding to a particular λ. Its horizontal axis corresponds to θ_i and vertical axis corresonds to θ_o. The 2D slices are tiled according to the descending order of λ, from left to right then from bottom to top. The slice for $\lambda = 64$ is at the bottom-left corner and the slice for $\lambda = 16$ is at the top-right corner.*

The specular component of the convolution integral

$$I_s(\omega_o;\xi,\lambda) = \int_\Omega G(\omega_i;\xi,\lambda)\cos^p(\theta_i+\theta_o)\, d\omega_i$$

is independent of the azimuth angles of both ξ and ω_o, and thus can be precomputed as a 3D table $\mathbf{I}_s(\cos\theta_\xi, \cos\theta_o, 1/\lambda)$. Note that the diffuse and specular coefficients are not precomputed into the tables but are multiplied at runtime. This also makes the precomputed tables monochromic and reduces the storage cost to one third.

5 Computing Effective Transmittances

The most straightforward way to approximate the effective transmittance $\tilde{T}(\xi,\lambda)$ in Eq. (5) is to use the transmittance sampled at the SRBF center, or $T(\xi)$, which can be efficiently evaluated using the deep opacity map technique [Yuksel and Keyser 2008]. This approximation is accurate if λ approaches infinity and the SRBF becomes a delta function defined on the sphere. It becomes less accurate as λ decreases and the energy of the SRBF lighting spreads in a larger solid angle. Multiplying the shading integral with the center transmittance produces a sharper shadow boundary (Fig. 5(a)) compared to the reference result (Fig. 5(b)), which exhibits soft shading variations since the occlusion changes smoothly from point to point.

This problem is analogous to the visibility averaging problem in soft shadow rendering, which can be efficiently solved by convolving the shadow map with a depth-dependent kernel [Fernando 2005; Annen et al. 2008]. Based on this observation, we introduce the convolution optical depth map to better approximate the effective transmittance by filtering the optical depth map. Our method consists of two passes: a light pass and a view pass.

Light Pass In the light pass, we first assume that light comes from the SRBF center and compute the deep opacity depth map (DODM) in a manner similar to the deep opacity map [Yuksel and Keyser 2008]. The hair geometry is rendered from the viewpoint of the light to obtain the hair's depth map z_0 as seen from the light source. The depth map is then used as the starting point for each pixel to divide the hair volume into K layers along the light direction, each of which start at depth $z_k = k(k+1)d/2, k = (1 \ldots K)$. Here d is the depth of the first layer. See Fig. 4 (left) for an illustration.

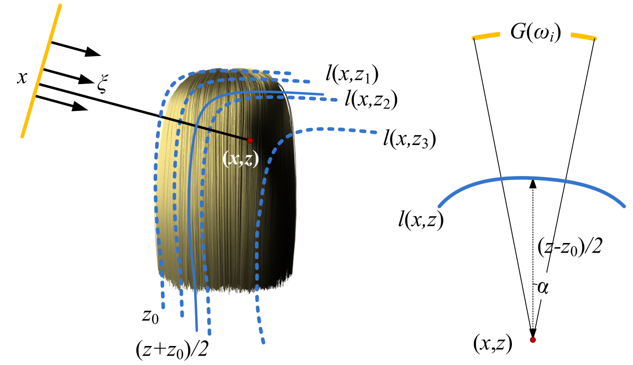


Figure 4: *Left: Approximating the effective transmittance for an SRBF light by querying the DODM generated at the center of the SRBF light. Right: Averaging the optical depth over the projection of the support of an SRBF light onto the effective opacity plane.*

The hair geometry is then rendered from the light's viewpoint again and each fragment's contribution to the optical depth, $\sigma_a D$ with σ_a being the absorption coefficient of the fiber, is accumulated to the layer k_0 to which it belongs and to all layers before layer k_0 (all layers k such that $k < k_0$). In the end, we get the DODM, a discrete sampling of the 1D optical depth function at each pixel x:

$$l(x,z_k) = \int_{z_k}^{z_0} \sigma_a \rho(x,z)\, dz, (k = 1,\ldots,K), \tag{10}$$

where x is the 2D position in the light's view space and $\rho(x,z)$ is the density function at the 3D point determined by (x,z). Note that instead of accumulating opacity as in [Yuksel and Keyser 2008] we accumulate optical depth in this step. Unlike opacity, optical depth can be directly accumulated, thus simplifying the light pass computation.

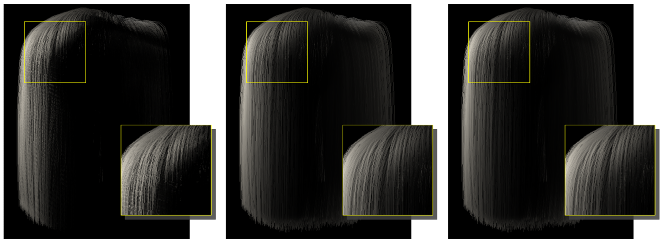
We then build a summed area table (SAT) [Hensley et al. 2005] for each layer of the DODM to facilitate computation of the average of optical depth values in the view pass. An SAT is a 2D array in which each entry holds the sum of the pixel values between the sample location and the bottom-left corner of the corresponding input image. From these SATs, we can easily obtain the average optical depth values in a square of size $2R$, denoted as $\bar{l}(x,z_k;R)$. In addition to the SATs of the DODM, we also store the starting depth z_0 in a separate texture.

View Pass In the view pass, the hair geometry is rendered in the viewer's view space. Each fragment is first projected to the light's view space. The projection depth z is used to determine an *effective occlusion plane* (indicated by the solid blue line in Fig. 4), on which the projection of the SRBF is evaluated to determine $R(z)$, the radius for the averaging, as shown in Fig. 4 (right). We then fetch the two nearest average optical depth values $\bar{l}(x,z_{k_0};R(z))$ and $\bar{l}(x,z_{k_0+1};R(z))$ from the SATs of the DODM. The average optical depth value at the fragment, $\bar{l}(x,z;R(z))$, is computed by linearly interpolating the two values according to z. The transmittance is obtained by exponentiating the negative convolution optical depth.

To determine the averaging radius on the effective occlusion plane, we observe that the radius R of the projection of the SRBF is proportional to $(z-z_0)/2 \cdot \tan\alpha$ (see Fig. 4, right), where α is the angular support of the SRBF. In practice, we set SRBFs to zero if the function value is less than 10^{-3}. Therefore α can be solved from the following equation:

$$G(\omega_i;\xi,\lambda) = \exp(-\lambda)\exp(\lambda\cos\alpha) = 10^{-3}.$$

Several approximations have been made in the above computation of the average optical depth. First, we assume the optical depth from the fragment to the starting point is generated by an optical depth function that is distributed on the effective occlusion plane. Second, by using the SAT to compute the average optical depth, we are actually using the average in an axis-aligned square instead of an average over the SRBF's projection, which should be a circle. Nevertheless, we find that this calculation significantly improves the approximation of the effective transmittance under a single SRBF light, as is shown in Fig. 5(c). It generates a visually plausible result that captures the smooth shading variations in the reference image shown in Fig. 5(b).



(a) center transmittance (b) reference (c) SAT averaging

Figure 5: *Approximation of the effective transmittance. The result using the transmittance sampled at the SRBF center is shown in (a), which does not capture the soft shading variation that is present in the reference image (b) obtained by numerically integrating over the SRBF light (2k samples). Using the convolution optical depth map significantly improves the approximation as shown in (c).*

To verify the accuracy of the approximations, we also compare our results with reference images obtained by directly integrating a large number of directional lights of environment maps. As shown in Fig. 1(a), (b) and Fig. 7, our results compare favorably with the reference images generated by directly accumulating directional lights and path tracing.

6 Multiple Scattering Computation

As mentioned in Section 3, our multiple scattering algorithm consists of two parts. During preprocessing, the convolution of global multiple scattered irradiance and the BCSDF of the fiber (Eq. (6)) is precomputed as a 4D table. At runtime, a fast approach based on sparse interpolation is employed to evaluate the forward scattering attenuation and spread according to the scattering events sampled on the shadow path, which can be used to query the precomputed 4D table and compute the dual scattering approximation.

Precomputing Convolution Table The convolution term in the multiple scattering equation (Eq. (7)) is:

$$I_G(\omega_o;\xi,\bar{\sigma}_f) = \int_\Omega \psi_f(\xi,\omega_i,\bar{\sigma}_f) S^D(\omega_i,\omega_o)\cos\theta_i\, d\omega_i. \tag{11}$$

In words, $I_G(\omega_o;\xi,\bar{\sigma}_f)$ is a convolution of the BCSDF $S^D(\omega_i,\omega_o)$ and the spread function $\psi_f(\xi,\omega_i,\bar{\sigma}_f)$, which is isotropic in the azimuthal direction and has a Gaussian distribution around the inclination of the SRBF center ξ in the longitudinal direction. We thus build a 4D table $\mathbf{I}_G(\cos\theta_\xi,\cos\theta_o,\cos(\phi_\xi-\phi_o),\bar{\sigma}_f)$.

Computing Attenuation and Spread The incident radiance generated by multiple scattering is often distributed smoothly in the space although it can have sharp angular variations. This inspires us to sample the forward scattering attenuation T_f and spread $\bar{\sigma}_f$ at only a small number of sample points via tracing the scattering events along the shadow path, and reconstruct the attenuation and spread in the entire hair volume using interpolation. The same strategy has been adopted by Moon et al. [2008] to interpolate the sampled incident radiance, represented as spherical harmonics coefficients, at a sparse set of points.

To avoid the highly expensive operation of tracing shadow paths through complex hair models, we voxelize the hair model into a volumetric representation. Within each voxel, we store three quantities that reflect the fiber distribution in the voxel, namely the average fiber direction $\bar{\omega}$, the standard deviation of fiber direction ν and the perpendicular attenuation coefficient $\sigma_t^\perp$.

Shadow rays are shot from outside the hair volume to the sample points along the direction of each SRBF center ξ_j. As we march along the shadow rays, scattering occurs with the probability

$$P = 1 - \exp(-\sigma_t^\perp \alpha(\nu, \theta_{\xi_j})\delta),$$

where δ is the marching step size. $\alpha(\nu, \theta)$ is the ratio between the actual attenuation function $\sigma_t(\omega)$ and $\sigma_t^\perp$. It is determined by the deviation ν and incident inclination θ and is precomputed as a 2D table. If scattering occurs, a fiber direction is generated according to a Gaussian distribution determined by $\bar{\omega}$ and ν, and then used to compute the average forward scattering attenuation $\bar{a}_f(\theta_{\xi_j})$ and spread $\bar{\beta}_f(\theta_{\xi_j})$ according to [Zinke et al. 2008]. The total attenuation T_f and spread $\bar{\sigma}_f$ are then updated. Note that unlike [Zinke et al. 2008], the bandwidth of the SRBF needs to be considered to determine the longitudinal variance due to the first scattering event. This can be done by querying a precomputed SRBF spread table indexed by the incident inclination $\cos\theta$ and bandwidth λ.

Once we have marched to the sample point, we have the actual forward scattering attenuation T_f and spread $\bar{\sigma}_f$. Note that this process is very similar to the light tracing process in [Moon et al. 2008], with the only difference that the rays in our case are not actually scattered - we only need to simulate the scattering events along the shadow ray for dual scattering approximation.

We sample the attenuation and spread on a coarse grid that typically contains several thousands of vertices, and pack the results as 3D textures, one for each SRBF light. In the view pass, for each shading point x, we iterate over all SRBF lights and use the position of x to query the 3D textures to get the interpolated attenuation and spread. The spread is then used together with $\cos\theta_\xi, \cos\theta_o$ and $\cos(\phi_\xi - \phi_o)$ to query the table $\mathbf{I}_G$. Finally, the queried value is multiplied with the attenuation T_f and SRBF weight L_j to yield the multiple scattering component. Due to the stochastic natural of the tracing process, we need to filter the sampled attenuation and spread to reduce noise, as Moon et al. did for the incident radiance. A box filter of one voxel width often suffices in our experiments.

Note that one straightforward approach to compute the forward scattering attenuation T_f and spread $\bar{\sigma}_f$ is to extend the deep opacity map method [Yuksel and Keyser 2008] by maintaining the attenuation, spread and direct illumination fraction values in the deep opacity map. This approach, however, requires several times the runtime video memory consumption and makes it impractical to process SRBF lights in batches – an important optimization as will be discussed in Section 7. Our approach also exploits the smoothness of the multiple scattered irradiance to reduce the number of shadow paths that need to be traced, which leads to further performance gain. Finally, the straightforward approach may suffer from the artifacts due to insufficient deep opacity map resolution and layers, which would be more visible in light colored hair where the multiple scattering component tends to be more significant.

7 Implementation Details

In this section we provide a few implementation details.

SRBF Light Approximation We use the method proposed by Tsai and Shih [Tsai and Shih 2006] to approximate an environment light with a set of SRBFs. SRBFs with very large supports may lead to large errors in approximating effective transmittances. On the other hand, the environment lighting would require an impractical number of SRBFs if the values of λ are set too high, yielding very narrow SRBFs. We found that bounding the bandwidth parameter of the SRBFs in $[16, 64]$ obtains good approximations for all environment lighting used in the paper.

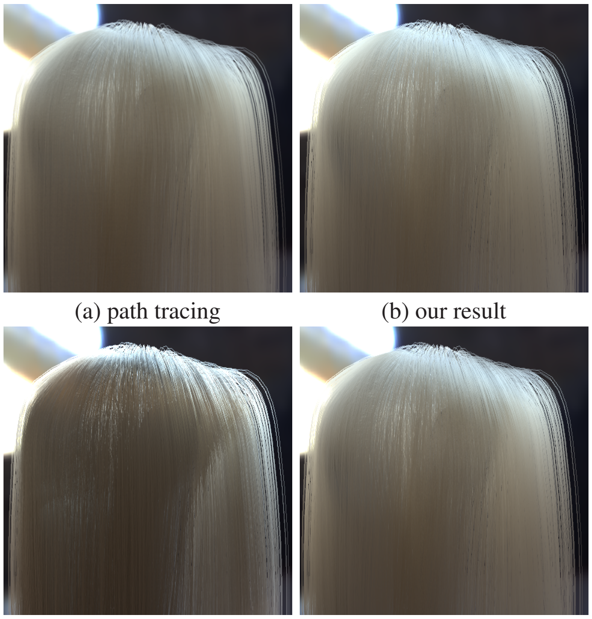
We start from a small number of SRBFs and recursively insert new SRBFs at the location of the largest fitting error when the minimizer converges. We stop the fitting when the overall error is less than a user-specified threshold. In our implementation, we start from 30 SRBFs and stop the insertion of new SRBFs if the mean squared error of the fitting is less than 15%. For all environment maps used in this paper, the resulting SRBF number is between 30 and 60.

Convolution Precomputation For the tabulation of $\mathbf{I}_d, \mathbf{I}_s$, $\mathbf{I}_M$ and $\mathbf{I}_G$ we use a resolution of 64 for the bandwidth parameter λ and $\bar{\sigma}_f$, and a resolution of 32 for $\cos\theta_\xi$, $\cos\theta_o$ and $\cos(\phi_\xi - \phi_o)$. To compute the convolution value at each table entry, we use numerical integration, and the GPU scan primitive [Harris et al. 2007] is employed to accelerate the integration computation. To compute single scattering, we only need $\mathbf{I}_M$, which is 12MB in size. For the full model, however, we need two convolution tables $\mathbf{I}_M$ and $\mathbf{I}_G$, consuming 24MB video memory in total. The S_{back} term also needs to be included in the table $\mathbf{I}_M$ in this case, according to Zinke et al.[2008].

Runtime Rendering At runtime, hair fibers are rendered as lines in both the light pass and the view pass. We iterate over all SRBF lights and accumulate the radiance values under each light into a running sum buffer. We use off-screen rendering for each light. The SAT of the DODM is first constructed in the light pass and then queried in a view pass to obtain the average transmittance as each fragment is shaded. We divide the hair volume into $K = 4$ layers and separately store the starting depth in another texture, since they cannot be SAT filtered after all. One important optimization is that multiple SRBF lights can be packed and processed in batches. This can reduce the per-pass overhead, albeit at the cost of consuming larger video memory because a large buffer is required to hold the SAT of multiple lights. In our implementation, we process 16 SRBF lights in a batch, which consumes 48MB video memory at the resolution of 512×512 and is a good tradeoff between performance and video memory consumption. The hair model voxelization and shadow path tracing are implemented using NVidia CUDA [NVIDIA 2007]. In our implementation, we always obtain the coarse grid for sampling forward scattering attenuation and spread by reducing the resolution of the grid used for hair voxelization to one fourth.

Hair Geometry Simplification When generating the DODMs (Section 5) in the light pass, we can use simplified hair geometry to obtain performance gains without visually noticeable loss of rendering quality. The basic idea is similar to [Cook et al. 2007] – only render a random subset of the hair fibers during the light pass. We first shuffle the hair fibers by sorting them according to floating point values randomly generated for each fiber. This value varies between 0 and 1. When rendering the hair geometry to construct the DODM, we exclude all fibers whose associated random value is larger than a ratio $\kappa \in [0, 1]$. The remaining fibers are then enlarged by a scale of $1/\kappa$ to preserve the overall optical depth. For all the environment maps used in this paper, we find $\kappa = 0.1$ to be a good choice, as shown in Fig. 8. While this method can reduce the light pass cost to nearly one fifth, the overall performance gain is about 1.5× and is not so significant due to the cost of the multiple scattering computation and view pass.

Transparency To handle transparency in the view pass, we implemented a recent method proposed by Sintorn and Assarsson [2009].



(a) path tracing (b) our result

(c) directly accum. 60 lights (d) directly accum. 450 lights

Figure 6: *Comparing our result (b) with the path tracing result (a), 29 hours, as well as the results obtained by directly accumulating the directional lights at a similar frame rate (c), 8.3 fps, or with similar quality (d) at 0.92 fps. Our method uses 50 SRBFs to approximate the Kitchen environment light. And the directional lights used in (c) and (d) are obtained by applying Agarwal et al.'s algorithm [2003] on the same environment light.*

This requires an occupancy map and a slab map to be built at runtime, consuming 7MB of video memory. We did not use their method for hair self shadowing mainly because of the difficulty of implementing SAT on their data structure. Enabling transparency drastically increases the number of fragments that need to be shaded, and it is important to apply early opacity culling. Prior to any shading computation, we first evaluate the order of the fragments in view space [Sintorn and Assarsson 2009]. The computation will be terminated and a black color is returned if we find the order is larger than a threshold and the current fragment has little contribution to the final pixel color.

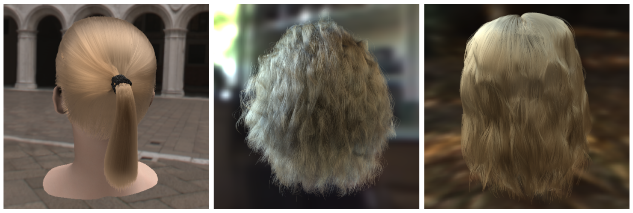
8 Results and Discussion

We have implemented the described algorithm on a PC with a 3.7 GHz CPU, 2GB memory and an NVidia GeForce GTX 285 graphics card. Images shown in the paper are all generated at a 640×480 resolution, but are sometimes cropped for better use of space.

The hair properties used in this paper, including the index of refraction and the longitudinal shifts and widths, are the same as those used in [Marschner et al. 2003]. We only modify the absorption coefficient σ_a to obtain hair with different colors.

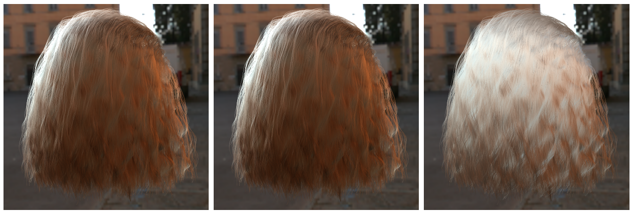
In Fig. 1, we compare our results with the reference results obtained by directly accumulating 6,144 directional lights, using the deep opacity map algorithm [Yuksel and Keyser 2008] for self shadowing and the dual scattering algorithm [Zinke et al. 2008] for multiple scattering, respectively. For the dual scattering, we use ray shooting to compute the forward scattering attenuation and spread from a dense grid of $256 \times 256 \times 256$ for the reference result. In both cases, our results compare favorably with the reference images. The self-occlusion and scattering effects generated by the hair volume are well captured by our algorithm.

In Fig. 6, we also compare our result with the results obtained by path tracing and directly accumulating the contribution of the directional lights obtained by Agarwal et al.'s algorithm [2003]. One important advantage of our algorithm is that the artifacts appear in the form of smoothing. We see obvious shadowing artifacts in the result of the alternative method, which is most severe when the hair is mainly lit by bright back lights, as shown in the comparison. In contrast, our algorithm obtains visually plausible results at similar frame rates. According to our experiment, the artifacts in Fig. 6(c) can only be eliminated with more than 400 directional lights for this environment light (d) and the frame rate is less than 1 fps.



(a) ponytail (b) curly (c) wavy

Figure 7: *Rendering different hair models under different lights.*



(a) $\kappa = 1.0$ (b) $\kappa = 0.1$ (c) $\kappa = 0.0$

Figure 8: *Rendering results with simplification ratios. At the simplification ratio* $\kappa = 0.1$*, we get a* $1.5\times$ *performance gain without noticeable quality degradation. The* $\kappa = 0.0$ *result is scaled down to avoid overexposure.*

Fig. 7 shows the rendering results of different hair models under different environment lights. We can see that the hair appearance changes significantly with respect to illumination, and the natural illumination generated by the environment lighting nicely blends the hair models with the environment.

Since the precomputation of the convolution tables is only related to the hair fiber properties and has nothing to do with the hair geometry, our algorithm can be used to render hair animations. Please see the accompanying video for rendering results of dynamic hair.

The statistics of the hair datasets used in this paper are reported in Table 1. The total cost of the runtime algorithm can be divided into four parts:

- Building the DODMs for self shadowing (Section 5);
- Computing the forward scattering attenuation and spread at the coarse grid by tracing the shadow path to each SRBF light center (Section 6, computing attenuation and spread);
- Building the occupancy and slab maps for transparency;
- Integrating the contributions of single scattering and multiple scattering due to each SRBF light in the view pass.

The last part is the most costly one for all test data in this paper, accounting for 50% -70% of the total render time. The total render time depends linearly on the SRBF number N, as well as the complexity of the hair model, or more specifically, the number of fragments generated by rasterizing the hair. For the hair animation in the accompanying video, voxelization of the hair model needs to be performed at runtime for each frame, which takes 150-200 ms.

The precomputation cost consists of two parts. Fitting each environment lighting with SRBF lights takes from 5-20 minutes. Precomputing the convolution tables for each fiber scattering function takes about 50 minutes.

9 Conclusion

We present a fast algorithm for interactive hair rendering with both single and multiple scattering effects under complex environment lighting. The environment light is approximated by a set of SRBFs. For each SRBF light, we factor out the effective transmittance to represent the outgoing radiance due to single scattering as the product of two terms: the transmittance and the convolution of the SRBF light and the scattering function. The convolution term can be precomputed as tables for commonly-used scattering models, and its run-time computation is reduced to table lookups. A convolution optical depth map technique is introduced to efficiently approximate the effective transmittance. The multiple scattering computation follows the same strategy of factorization and precomputation. Sparse sampling and interpolation are used to speed up the light transport computation. The precomputation involved in our algorithm is not related to hair geometry, and fully dynamic hair geometry can be supported.

There are a few limitations of our algorithm. First, since the scattering properties of hair fibers are baked into the convolution tabulation, our algorithm does not allow runtime changes of those properties. Second, the eccentricity of Marschner et al.'s model is omitted in the precomputed convolution table. To account for this eccentricity, another dimension would be needed in the sampling of the convolution of the SRBF lighting and the scattering function, making the precomputation and storage problematic.

Acknowledgements We would like to thank Michael Lentine, Ronald Fedkiw and Arno Zinke for sharing their hair data, and Steve Lin for proofreading this paper. Kun Zhou was partially funded by the NSFC (No. 60825201) and the 973 program of China (No.2009CB320801).

hair model	#fibers	#segments	grid size	FPS
animation (Fig. 1)	10K	270K	$54 \times 74 \times 90$	7.9/16.2
straight (Fig. 6)	10K	160K	$66 \times 60 \times 90$	8.3/17.8
ponytail (Fig. 7(a))	20K	900K	$46 \times 75 \times 75$	6.2/11.1
curly (Fig. 7(b))	50K	3.4M	$71 \times 70 \times 82$	1.3/2.30
wavy (Fig. 7(c))	10K	687K	$60 \times 55 \times 76$	6.7/12.3
natural (Fig. 8)	10K	1.6M	$72 \times 67 \times 74$	4.8/9.20

Table 1: *Statistics for test hair data used in this paper. Frame rates are given in FPS respectively for the single/multiple scattering and the single scattering only. The sizes of the grids used for hair voxelization and shadow path tracing are also provided.*

References

AGARWAL, S., RAMAMOORTHI, R., BELONGIE, S., AND JENSEN, H. W. 2003. Structured importance sampling of environment maps. *ACM Trans. Graph. 22*, 3, 605–612.

ANNEN, T., DONG, Z., MERTENS, T., BEKAERT, P., SEIDEL, H.-P., AND KAUTZ, J. 2008. Real-time, all-frequency shadows in dynamic scenes. *ACM Trans. Graph. 27*, 3, 34:1–8.

COOK, R. L., HALSTEAD, J., PLANCK, M., AND RYU, D. 2007. Stochastic simplification of aggregate detail. *ACM Trans. Graph. 26*, 3, 79.

FERNANDO, R. 2005. Percentage-closer soft shadows. In *SIGGRAPH 2005 Sketches*, 35.

HARRIS, M., SENGUPTA, S., AND OWENS, J. 2007. Parallel prefix sum (scan) in CUDA. *GPU Gems 3*, Chapter 39.

HENSLEY, J., SCHEUERMANN, T., COOMBE, G., SINGH, M., AND LASTRA, A. 2005. Fast summed-area table generation and its applications. *Computer Graphics Forum*, 547–555.

KAJIYA, J. T., AND KAY, T. L. 1989. Rendering fur with three dimensional textures. In *Proceedings of ACM SIGGRAPH9*, 271–280.

KIM, T.-Y., AND NEUMANN, U. 2001. Opacity shadow maps. In *Eurographics Workshop on Rendering*, 177–182.

LOKOVIC, T., AND VEACH, E. 2000. Deep shadow maps. In *Proceedings of ACM SIGGRAPH*, 385–392.

MARSCHNER, S. R., JENSEN, H. W., CAMMARANO, M., WORLEY, S., AND HANRAHAN, P. 2003. Light scattering from human hair fibers. *ACM Trans. Graph. 22*, 3, 780–791.

MERTENS, T., KAUTZ, J., BEKAERT, P., AND REETH, F. V. 2004. A self-shadow algorithm for dynamic hair using density clustering. In *SIGGRAPH 2004 Sketches*, 44.

MOON, J. T., AND MARSCHNER, S. R. 2006. Simulating multiple scattering in hair using a photon mapping approach. *ACM Trans. Graph. 25*, 3, 1067–1074.

MOON, J. T., WALTER, B., AND MARSCHNER, S. 2008. Efficient multiple scattering in hair using spherical harmonics. *ACM Trans. Graph. 27*, 3, 31:1–7.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2003. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph. 22*, 3, 376–381.

NVIDIA, 2007. CUDA homepage.
http://developer.nvidia.com/object/cuda.html.

RAMAMOORTHI, R., AND HANRAHAN, P. 2001. An efficient representation for irradiance environment maps. In *Proceedings of ACM SIGGRAPH*, 497–500.

SINTORN, E., AND ASSARSSON, U. 2008. Real-time approximate sorting for self shadowing and transparency in hair rendering. In *Proceedings of I3D*, 157–162.

SINTORN, E., AND ASSARSSON, U. 2009. Hair self shadowing and transparency depth ordering using occupancy maps. In *Proceedings of I3D*, 67–74.

SLOAN, P.-P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of ACM SIGGRAPH*, 527–536.

TSAI, Y.-T., AND SHIH, Z.-C. 2006. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Trans. Graph. 25*, 3, 967–976.

WARD, K., BERTAILS, F., KIM, T.-Y., MARSCHNER, S. R., CANI, M.-P., AND LIN, M. 2007. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics (TVCG) 13*, 2, 213–234.

YUKSEL, C., AND KEYSER, J. 2008. Deep opacity maps. *Computer Graphics Forum 27*, 2, 675–680.

ZINKE, A., AND WEBER, A. 2007. Light scattering from filaments. *IEEE Trans. Vis. Comp. Graph. 13*, 2, 342–356.

ZINKE, A., YUKSEL, C., WEBER, A., AND KEYSER, J. 2008. Dual scattering approximation for fast multiple scattering in hair. *ACM Trans. Graph. 27*, 3, 32:1–10.

ZINKE, A. 2008. *Photo-Realistic Rendering of Fiber Assemblies*. PhD thesis, University of Bonn.