

# On Physically Based Hair Rendering

SARAH INVERNIZZI

Master of Science,

Computer Animation and Visual Effects



August, 2013

## Abstract

The use of ray tracing and physically based rendering is a growing trend in the fields of animation and VFX. Geometric lights, in combination with high dynamic range image based lighting, can extract a more realistic response from physically based material integrators and achieve better results, usually at greater computational expense. Therefore importance sampling aims to improve efficiency of Monte Carlo based rendering methods, whilst maintaining the benefits of the physically plausible pipeline.

This thesis investigates two importance sampled hair reflectance models, ISHair: Importance Sampling for Hair Reflection (Ou et al., 2012a) and Importance Sampling of Reflection from Hair Fibers (Hery and Ramamoorthi, 2012). The former importance samples only the Marschner reflectance lobe, and the latter importance samples the three modes of light transport commonly associated with light scattering from hair: R, TT, TRT. Both shaders are implemented in RSL and images rendered using the physically plausible pipeline in RenderMan Pro Server 18.0. Hair self-shadowing remains a challenge, however the best results are achieved with deep area shadow maps. Anti-aliasing is investigated and attempts to reduce noise from a sequence of moving images show promising results, although it is not completely eliminated.

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Previous Work</b>	<b>6</b>
<b>3</b>	<b>Physically Based Rendering</b>	<b>9</b>
3.1	Motivation . . . . .	9
3.2	Theory of Physically Based Shading . . . . .	11
3.2.1	BRDF . . . . .	11
3.2.2	Microfacet Theory . . . . .	11
3.2.3	Importance Sampling . . . . .	13
<b>4</b>	<b>Importance Sampling Hair Shading Models</b>	<b>14</b>
4.1	Notation . . . . .	14
4.2	Importance Sampling the Marschner Reflection Lobe . . . . .	15
4.3	Importance Sampling a Multi-Lobe Hair Scattering Model . . . . .	17
<b>5</b>	<b>Implementation</b>	<b>20</b>
5.1	Pipeline . . . . .	20
5.2	Hair Model . . . . .	21
5.3	Shader Implementation . . . . .	23
5.3.1	plausibleMarschnerR . . . . .	25
5.3.2	plausibleIsHair . . . . .	29
5.4	Shadowing . . . . .	29
5.5	Optimisations for Hair . . . . .	30
<b>6</b>	<b>Results</b>	<b>33</b>
6.1	Environment Lighting . . . . .	33
6.2	Samples . . . . .	37
6.3	Indirect Lighting . . . . .	43
6.4	Ray Traced Shadows . . . . .	43
6.5	High Quality Renders . . . . .	45
<b>7</b>	<b>Conclusions</b>	<b>47</b>

<b>8</b>	<b>Known Issues and Further Work</b>	<b>48</b>
8.1	Known issues . . . . .	48
8.2	Further Work . . . . .	49

## List of Figures

1	The Marschner model: longitudinal and azimuthal pathways (Marschner et al., 2003) . . . . .	6
2	Diffuse sampling artefacts . . . . .	10
3	Active microfacets (Hoffman, 2012). . . . .	12
4	The shape of the specular lobes. Longitudinal lobes (left) and azimuthal lobes (right) (Ou et al., 2012a). . . . .	18
5	Pipeline flow chart for a single frame . . . . .	22
6	The Radiance Sample Construct . . . . .	25
7	The hair local coordinate system (Marschner et al., 2003) . . . . .	26
8	Shader class diagram . . . . .	27
9	RenderMan MIS Shading Pipeline . . . . .	28
10	Deep Area Shadow Map . . . . .	30
11	plausibleMarschner in Woodland Lane: 1024 lighting and 1024 materials samples. 24 minutes render time. . . . .	34
12	plausibleMarschner in Pisa: 1024 lighting and 1024 materials samples. 25 minutes render time. . . . .	34
13	plausibleMarschner in Grace: 1024 lighting and 1024 materials samples. 23 minutes render time. . . . .	35
14	plausibleIsHair in Woodland Lane: 1024 lighting and 1024 materials samples. 45 minutes render time. . . . .	35
15	plausibleIsHair in Pisa: 1024 lighting and 1024 materials samples. 46 minutes render time. . . . .	36
16	plausibleIsHair in Grace: 1024 lighting and 1024 materials samples. 44 minutes render time. . . . .	36
17	The number of light and material samples against run time . . . . .	38
18	The effect of light samples on image quality for plausibleMarschner . . . . .	39
19	The effect of material samples on image quality for plausibleMarschner . . . . .	40

20	The effect of light samples on image quality for plausibleIsHair . . . . .	41
21	The effect of material samples on image quality for plausibleIsHair . . . . .	42
22	Cornell Box Test for plausibleIsHair . . . . .	43
23	Ray Traced Shadows for plausibleIsHair: 1k light and material samples . . .	44
24	Ray Traced Shadows for plausibleIsHair: 8k light and material samples . . .	44
25	High Quality Render for plausibleIsHair: 8k light and material samples and Deep Area Shadows . . . . .	45
26	High Quality Render for plausibleMarschner: 8k light and material samples and Deep Area Shadows . . . . .	46
27	16 light samples and 64 material samples for plausibleMarschner . . . . .	55
28	16 light samples and 64 material samples for plausibleIsHair . . . . .	55

## List of Tables

1	Notation . . . . .	14
2	Hair Models (Yuksel, ca.2103b) . . . . .	23
3	The number of samples against performance. . . . .	37

# 1 Introduction

Synthesising realistic human hair is a difficult task, primarily because of the number and geometry of hair strands, but also because of the duality of hair. Its appearance depends not only on each strand (local properties) but also on the hair considered as a volume (global properties). The local properties of an individual hair strand define the way that it is illuminated. When the hair is considered as a volume, global properties such as translucency, multiple scattering and self-shadowing become especially pertinent to creating a convincing appearance.

Over the past 10 years there has been a growing trend in the use of increasingly physically accurate models to represent light scattering from human hair. This complements the progression towards Physically Based Rendering (PBR) that is currently being seen in both the CG and VFX industries (Seymour, 2103b). Although physically based illumination models have been around for some time, it is only recently that they have started to replace the common sense, practical-approach illumination models (Hoffman, 2010).

This thesis focuses on single scattering models for hair rendering; the analytic models that strive to accurately model hair-light interaction and how these models have been adapted to efficiently use Monte Carlo rendering techniques through Multiple Importance Sampling (MIS). Rendering with global illumination, physically based lights and shading models, as well as ensuring that images contain the correct visual aesthetic, such as colour bleed and self-shadowing are crucial to achieving realism.

A brief history of hair shading models is presented in Section 2; this is followed by an introduction to PBR in Section 3. The theory of each of the implemented shading models is detailed in Section 4. The shading models are implemented in RenderMan Shading Language (RSL) and used in conjunction with RenderMan Pro Server (RPS) 18.0, the implementation is described in section 5, and the results are compared and discussed in Section 6.

## 2 Previous Work

In 1989 Kajiyama and Kay introduced a methodology for rendering fur using texels and a phenomenological hair strand illumination model (Kajiyama and Kay, 1989). The model approximates the specular component as light scattered from thin cylinders, this creates the linear specular highlight that characterises hair. Due to its simplicity this model has been widely used to render hair for many years, however it does not account for important details such as the two additional modes of light transport, internal reflection and transmission, that give hair its secondary specular highlight and make light-coloured hair appear translucent.

Goldman refined the Kajiyama-Kay model (1997), he added a factor to bias the model towards either forward or backwards scattering, thus simulating translucency.

In 2003, Marschner et al. presented their highly influential, physically based light scattering model. Following experimental observations into the light scattering from human hairs, Marschner et al. successfully transferred the aesthetic nuances to an analytic model, in particular capturing the dual specular highlight. Their model separates the hair Bidirectional Scattering Distribution Function (BSDF) into functions dependent on elevation and azimuth angle, where the azimuthal terms incorporate the affects of volume absorption, Fresnel factor and caustics.

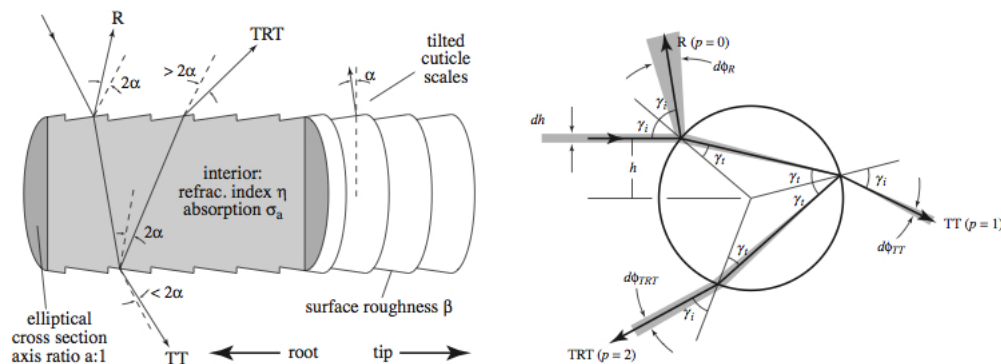


Figure 1: The Marschner model: longitudinal and azimuthal pathways (Marschner et al., 2003)

d'Eon et al. (2011) explain that the drawbacks of the Marschner model are that it is

computationally expensive, not energy conserving for all conditions, and reflectance is limited to the first three modes of light propagation: reflection (R), transmission (TT), and a single internal reflection (TRT). Nevertheless, d'Eon et al. (2011) extend and improve on the Marschner model to create one that is energy-conserving and can be extended to account for multiple orders of internal reflection.

Ou et al. (2012a) claim that another drawback of the Marschner model is that it is unintuitive for artists to control the appearance through the model's many parameters. Sadeghi et al. (2010) endeavoured to improve the artist's user experience with their artist friendly hair shading system. Inspired by the observations made by Marschner et al. (2003) they decomposed the BSDF into four specular lobes. However, at Disney the benchmark of CG hair is "that it is aesthetically pleasing and that it fits within the universe of the character" (Sadeghi et al., 2010, p. 1). Therefore, the physical accuracy of the model was compromised, for example, components of a single visual feature such as the colour and brightness of a highlight are decoupled.

The importance sampling of hair BSDFs is a relatively new area of research; Neulander first sampled a Kajiya-Kay cone-based model in 2010 (cited by Ou et al., 2012a). Hery and Ramamoorthi (2012) present a methodology for importance sampling the reflection lobe of the Marschner model using a Box-Muller transform to effectively sample the Gaussian function, this is discussed in more detail in section 4.2. Following on from the Hery model, Ou et al. (2012a) discuss its limitations and propose a solution to importance sample the artist friendly model (Sadeghi et al., 2010). They replace the Gaussian distribution with a Cauchy distribution to facilitate the use of the inverse (Cumulative Distribution Function) CDF sampling technique, see section 4.3.

One of the first effective solutions to the self-shadowing of objects with some degree of transparency, such as hair, was Pixar's Deep Shadow Maps (Lokovic and Veach, 2000). By storing a visibility function at each Pixel, instead of recording a depth, a more accurate light attenuation calculation is performed. Supplementary to this concept was the research of Kim and Neumann (2001) in Opacity Shadow Maps, and Yuksel and Keyser (2008) in Deep Opacity Maps. The former aimed at reducing the pre-computation time, and the latter improving quality and real-time performance. Most recently Pixar have introduced support for deep area shadow maps (Pixar, 2012b), these traceable maps can provide depth values over a range of directions, making it possible to use a single map with an area



light.

Multiple scattering of light between hairs is crucial for the correct appearance of low-absorbing hair, such as blonde or grey coloured hair. There is much existing research into how it can be approximated (Moon and Marschner, 2006; Moon et al., 2008; Sadeghi and Tamstorf, 2010; Zinke et al., 2008), but discussion here is outside the scope of this research.

### 3 Physically Based Rendering

Physically based rendering (PBR) means creating images that are *physically correct*; the lighting is reflected as in the real world (Pharr and Humphreys, 2010). Therefore the first consideration for PRB is to use physically based lights, and this mean that they should have size. Area lights (in a multitude of shapes), volumetric lights and Image Based Lighting (IBL) through environment lights are the main sources of geometric illumination. High Dynamic Range (HDR) images are used in IBL techniques to drive a realistic lighting response from the environment. It is strongly recommended that the unclipped (full 32-bit) HRD maps are used in order to achieve the most realistic response (FXPhd, 2013). Once area lights are used in conjunction with HDR IBL, existing shaders reach their limit and the result is renders with many sampling artefacts and unrealistic responses. Therefore a new generation of shaders is required. These shaders should be energy conserving and should have physically based Bidirectional Reflectance Distribution Functions (BRDFs) (Seymour, 2103a).

#### 3.1 Motivation

One of the main incentives to move to PBR is to reduce the complexity of shots and facilitate easier communication between lighters. Pixar’s Christophe Hery explains how, at every iteration, lighters would have to turn each light on and off to determine what effect it was having (Seymour, 2103a). Hoffman (2010) remarks that ad-hoc shading models require numerous tweaks, usually over many iterations, to get production quality images, and that this process is very laborious.

The aim with PBR is not to reduce rendering times, indeed with ray-tracing times may increase, but to improve the user’s experience and to improve shot quality (Seymour, 2103a). However, he explains that with a simpler rendering pipeline, fewer passes and reduced pass management requirements, a superior result can often be achieved in a shorter timescale. The argument for this presented by Pixar (2011) is that ”it is cheaper for a computer to produce physically plausible images automatically than it is for lighting artists to mimic the physical effects with cheaper (i.e. non ray-traced) integrators”.

Hery goes on to say ”Once you move to physically based frame work, once you do specular, your diffuse is almost for free,” he explains ”Because part of computing the specular samples

you have to do light sampling, which is what you have to do for diffuse” Seymour (2103a). In RenderMan this reuse is enabled by the radiosity cache and can it can significantly improve rendering performance, especially in shots with complex global illumination lighting Pixar (2011).

Furthermore, environment lights that use HDR IBL are commonplace in physically based rendering. One downfall of using High Dynamic Range Images (HDRI) for environment lighting is that if the image is particularly high contrast, for example the sun on a clear day or many small light sources at night, rendering artefacts can occur in the diffuse pass due to sampling (FXPhd, 2013). Figure 2 illustrates the kind of problem that can arise from the diffuse sampling of a very bright light source, or a very sharp specular reflection.

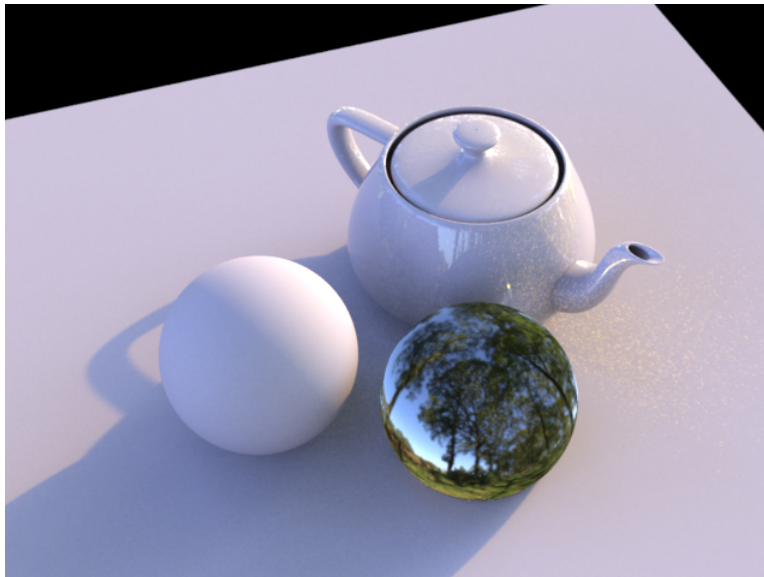


Figure 2: Diffuse sampling artefacts

Workarounds to remove artefacts include clipping the HDRI to an 8-bit map, or using a convolved environment map, which pre-blurs the map so that a single sample can be used, in conjunction with an ambient occlusion pass. However, the drawbacks are that it can result in inconsistent material settings between different lighting environments (because you end up tweaking the shader/material parameters for a single shot), and the shaders are not energy conserving.

## 3.2 Theory of Physically Based Shading

The 2012 Siggraph course on Physically Based Shading (Hoffman, 2012) provides an excellent background to the theory and physics of light, and physically based shading. However, a brief overview of the most pertinent aspects follows.

To calculate the reflected radiance at a point,  $L_o(\omega_r)$ , the *reflection equation* is used. It is a specialisation of the *rendering equation* (Hoffman, 2012) and is given as follows

$$E(\omega_r) = \int_{\Omega} f(\omega_i, \omega_r) L(\omega_i) \cos\theta d\omega_i \quad (1)$$

Where  $E(\omega_r)$  is the radiance,  $f(\omega_i, \omega_r)$  is the Bidirectional Reflectance Distribution Function (BRDF),  $L(\omega_i)$  is the incident irradiance, and the cosine factor is applied to convert the irradiance to the equivalent per projected surface area (Foley et al., 1990).

### 3.2.1 BRDF

For a BRDF to be physically plausible, Hoffman (2012) describes the two main features that it should have: *energy conservation* and *reciprocity*. In order for a BRDF to be energy conserving, a point should not reflect more light that it receives, and reciprocity refers to the BRDF producing equivalent results should the incoming and reflected vectors be swapped.

### 3.2.2 Microfacet Theory

Hoffman (2012) explains that *microfacet theory* is the foundation of the formulation of physically based specular BRDFs. It facilitates the use of physics for optically flat surfaces on a surface that is not optically flat, in order to approximate surface reflection and refraction.

An optically flat surface is "perfectly flat to the scale of hundreds of nanometers - such surfaces are called optically flat and are typically used for high-quality optical instruments such as telescopes" (Hoffman, 2012, p. 5). In rendering theory it is desirable to model

surfaces as optically flat as it provides a convenient analytical solution to Maxwell’s equations. In this special case Maxwell’s equations are better known as the Fresnel equations and are used to describe the proportion of light that is reflected and refracted at an interface. However, real-world surfaces are rarely finished to such precision and usually consist of many small irregularities that are not visible and cannot be rendered, but are greater in magnitude than the wavelength of light. Therefore, it can be assumed the surface behaves like many minuscule optically flat surfaces, each oriented in a different direction.

In shading theory it is convenient to represent the huge numbers of micro-surfaces, or microfacets, in a statistical manner. Thus, at a macrosurface level a single point can be seen to reflect and refract light in different directions. The variation in the surface orientation drives the type of reflections visible on the surface of a material; the rougher the surface, the greater the variance in the surface normals and the reflections are more blurred (Hoffman, 2012).

An active micro facet is one that has a surface normal  $m$  to be exactly halfway between the view  $v$  and light  $l$  vectors, also known as the half-vector  $h$ .

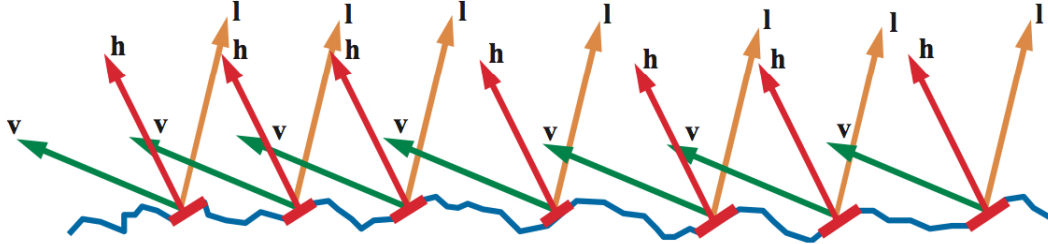


Figure 3: Active microfacets (Hoffman, 2012).

A microfacet BRDF is comprised of terms that account for Fresnel reflectance of active microfacets, the microfacet normal distribution, proportion of non-shadowed or masked microfacets, as follows:

$$f_{\mu facet}(l, v) = \frac{F(l, h)G(l, v, h)D(h)}{4(n \cdot l)(n \cdot v)} \quad (2)$$

Where  $F$  is the Fresnel reflectance,  $G$  is the shadow masking function and  $D$  is the microfacet normal.

### 3.2.3 Importance Sampling

The Monte Carlo estimator evaluates the value of an arbitrary integral, and as such can be applied to approximate the reflection equation, equation 1, as follows

$$E(\omega_r) \approx \frac{1}{N} \sum_{k=1}^N \frac{f(\omega_{ik}, \omega_r) L_i(\omega_{ik}) \cos\theta}{p(\omega_{ik}, \omega_r)} \quad (3)$$

The key for variance reduction in Monte Carlo methods is the selection of an appropriate probability density function (PDF), denoted by  $p(\omega_{ik}, \omega_r)$  in equation 3. It is easy to generate uniform random directions for rays and a uniform distribution may be suitable at times, such as for a Lambert surface BRDF, but uniformly sampling an integral with a different behaviour can introduce noise into the render. Ou et al. (2012a) state that Monte Carlo integration is more efficient when the PDF selected is proportional to the function being integrated.

Once the PDF has been selected, random samples are drawn. The inversion method is a commonly used technique to generate such samples (Hery and Ramamoorthi, 2012). Pharr and Humphreys (2010) describe the inversion method in detail in Section 13.3, however it can be summarised as follows:

1. Determine the Cumulative Distribution Function (CDF) for the selected PDF,  $P(x) = \int_0^x p(x) dx$
2. Invert the CDF,  $P^{-1}(x)$
3. Substitute a canonical uniform random number,  $\xi$ , into the inverse to calculate the sample value,  $X = P^{-1}(\xi)$

## 4 Importance Sampling Hair Shading Models

### 4.1 Notation

The notation used in the theory that follows, and in the corresponding implementation, matches that found in most literature on hair rendering, and that is the notation proposed by Marschner et al. (2003).

The hair’s local coordinate system is a right-handed orthonormal basis. The  $\mathbf{v}$ - $\mathbf{w}$  plane (or *normal plane*) is perpendicular to the hair strand and the  $\mathbf{v}$  axis runs along the major axis of its cross-section, given a cross-section with eccentricity. The hair’s tangent provides the  $\mathbf{u}$  vector, in the direction pointing from root to tip.

The direction of incoming and reflected light,  $\omega_i$  and  $\omega_r$  respectively, can be expressed in spherical coordinates, as defined in Table 1. Where elevation,  $\theta$ , lies in the range  $[-\pi/2, \pi/2]$  and azimuth,  $\phi$ , in the range  $[-\pi, \pi]$ .

Symbol	Description
$\mathbf{u}$	The hair tangent, pointing from root to tip
$\mathbf{v}, \mathbf{w}$	The two axes perpendicular to the tangent, the normal plane
$\omega_i, \omega_r$	The incident and reflected light vectors, both pointing away from the centre
$\theta_i, \theta_r$	The elevation of the incident and reflected light vectors. Measured from the normal plane where $0^\circ$ is perpendicular to $\mathbf{w}$ , $90^\circ$ is $\mathbf{u}$ and $-90^\circ$ is $-\mathbf{u}$
$\phi_i, \phi_r$	The azimuths around the hair of the incident and reflected light vectors projected onto the normal plane, where $\mathbf{v}$ is $0^\circ$ and $\mathbf{w}$ is $90^\circ$
$\theta_h$	The longitudinal half-angle: $\theta_h = (\theta_r + \theta_i)/2$
$\theta_d$	The longitudinal difference angle: $\theta_d = (\theta_r - \theta_i)/2$
$\phi$	The relative azimuth angle: $\phi = \phi_r - \phi_i$
$\beta$	The width or standard deviation of the Gaussian lobe, represents the sharpness of the specular reflection
$\alpha$	The expected value of the Gaussian lobe, represents the shift of the specular reflection away from the perfect mirror direction caused by the hair cuticle. Negative shifts towards the root, and positive towards the tip

Table 1: Notation

## 4.2 Importance Sampling the Marschner Reflection Lobe

Hery and Ramamoorthi (2012) present an implementation for importance sampling the reflection term in the Marschner et al. (2003) light scattering model; this model was used in production at Pixar for the 2013 release, *Monsters University* (Hery and Ramamoorthi, 2012).

The hair scattering function  $S(\omega_i, \omega_r)$  can be parameterised into dependence on elevation,  $\theta$ , and azimuth,  $\phi$ , in the following form

$$S(\omega_i, \omega_r) = S(\theta_i, \theta_r, \phi_i, \phi_r) = M(\theta_i, \theta_r)N(\phi_i, \phi_r) \quad (4)$$

Hery and Ramamoorthi (2012) describe the functions that represent the longitudinal dependence,  $M$ , and azimuthal dependence,  $N$ , for the reflection lobe as follows, whilst the forward and backward scattering terms,  $TT$  and  $TRT$  respectively, are not considered.

$$M = g(\beta, \theta_h - \alpha) \quad (5)$$

$$N = \cos\left(\frac{\phi}{2}\right) \quad (6)$$

Where  $g$  is a normalised Gaussian function. They go on to explain that  $N$  is a simplified version of the Marschner  $N_R$  term as derived by Sadeghi et al. (2010).

Since the integral of the Gaussian function is without solution (Vazquez-Leal et al., 2012), a Box-Muller transform is used to approximately sample  $M$ . The Box-Muller transformation takes two independent canonical uniform random variables,  $\xi_1$  and  $\xi_2$ , and produces an independent random variable,  $Z_1$ , in the standard normal distribution (Weisstein, ca.2103a). A *canonical uniform random variable* lies within the domain  $[0, 1)$ , and all values have equal probability (Pharr and Humphreys, 2010).

$$Z_1 = \sqrt{-2\ln\xi_1}\cos(2\pi\xi_2) \quad (7)$$

The sample elevation,  $\theta_s$ , is then found by multiplying  $Z_1$  by the Gaussian width to account for variance. As a Gaussian function has an unlimited domain, the samples generated could



lie outside of the desired range. Therefore, values of  $\theta_s$  are clamped to  $\theta_{smax}$ . The incident elevation angle can then be calculated as follows

$$\theta_s = \beta Z_1 \quad (8)$$

$$\theta_i = 2(\theta_s + \alpha) - \theta_r \quad (9)$$

In order to enforce values of  $\theta_i$  to lie in the range  $[-\frac{\pi}{2}, \frac{\pi}{2}]$ , Hery and Ramamoorthi (2012) flip  $\theta_i$  about the u vector. Ou et al. (2012a) highlight the errors in these assumptions, stating that clamping  $\theta_h$  introduces bias and that flipping  $\theta_i$  alters  $\theta_h$ . Since this change is not accounted for it instils an inconsistency between the BRDF, PDF and the incident light vector. For further analysis of these problems please refer to Ou et al. (2012a), Appendices A and B.

To sample the N term Hery and Ramamoorthi use the inverse CDF method described in section 3.2.3. The PDF, described by equation 6, is first normalised in accordance with the integral bounds  $[-\pi, \pi]$ , this results in the formulation shown in equation 10, the resulting CDF is shown in equation 11.

$$p_N(\phi) = \frac{1}{4}N(\phi) = \frac{\cos(\phi/2)}{4} \quad (10)$$

$$P_N(\phi) = \frac{1}{2} \left( 1 + \sin \frac{\phi}{2} \right) \quad (11)$$

$$\phi_i = \phi_r + 2\sin^{-1}(2\xi - 1) \quad (12)$$

Equation 12 is the inverse of the CDF and includes the conversion to incident azimuth angle.  $\xi$  is the canonical uniform random variable.

Given the randomly generated spherical co-ordinates,  $(\theta_i, \phi_i)$ , the sample direction can now be computed and transformed from the local hair coordinate system back to the global coordinate system, see section 5.3 for more details.

The PDF for the material can then be calculated using equation 13, a derivation is given by Hery and Ramamoorthi (2012).

$$p(\theta_i, \phi_i) = \frac{M(\theta_h)N(\phi)}{8\cos\theta_i} \quad (13)$$

Finally the Monte Carlo estimator can be evaluated in order to estimate the response of the material. Hery and Ramamoorthi (2012) claim that for sharp specular lobes, the reflected direction will be very similar to the incident direction, and therefore the cosine fraction simplifies to one. Thus, including the specular colour,  $K_s$ , the material response can be described by equation 14.

$$K_s \frac{\cos^2\theta_i}{\cos^2\theta_d} \approx K_s \quad (14)$$

### 4.3 Importance Sampling a Multi-Lobe Hair Scattering Model

Ou et al. (2012a) augment the work of Sadeghi et al. (2010) by offering an importance sampling solution for their multi-lobe hair scattering model. The decomposition of the hair scattering function into the separate components proposed by Sadeghi et al. is similar to that of Marschner et al. (2003), except that the glint component is separated from the secondary reflection component and the two components are referred to as: secondary reflection without glint, TRT-g, and glint, g.

The longitudinal components are defined as follows

$$M_R = g(\beta_R^2, \alpha_R, \theta_h) \quad M_{TT} = g(\beta_{TT}^2, \alpha_{TT}, \theta_h) \quad M_{TRT} = g(\beta_{TRT}^2, \alpha_{TRT}, \theta_h) \quad (15)$$

Where  $g$  is shown in equation 16. Note that it differs from the definition of the standard normalised Gaussian function (Weisstein, ca.2103c).

$$g(\beta^2, \alpha, \theta_h) = e^{-(\theta_h - \alpha)^2/2\beta^2} \quad (16)$$

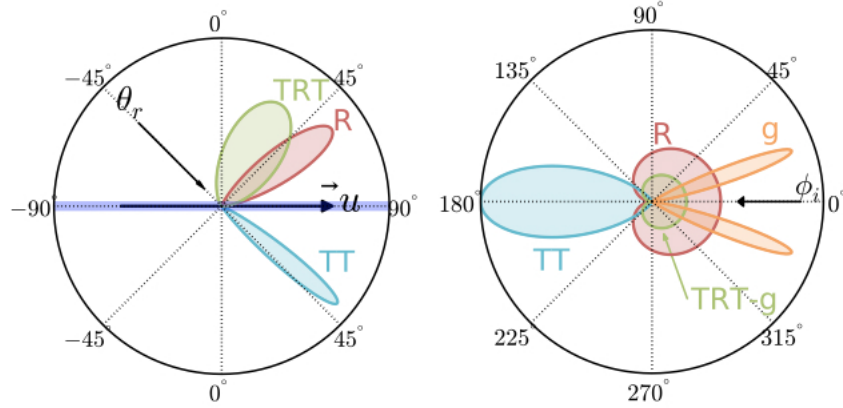


Figure 4: The shape of the specular lobes. Longitudinal lobes (left) and azimuthal lobes (right) (Ou et al., 2012a).

The azimuthal variance is captured in the following terms

$$N_R = \cos \frac{\phi}{2} \quad N_{TT} = g(\gamma_{TT}^2, \pi - \phi) \quad N_{TRT-g} = \cos \frac{\phi}{2} \quad N_g = g(\gamma_g^2, |\phi| - \phi_g) \quad (17)$$

Where  $\gamma_{TT}$  is the width of the  $N_{TT}$  lobe,  $\gamma_g$  is the width of each of the two glint lobes, and  $\phi_g$  is the half angle between them. All of these parameters are user defined.

As mentioned previously Ou et al. (2012a) criticise the use of the Box-Muller transform in the context of importance sampling, and instead opt for the antiderivative of the Cauchy distribution in conjunction with the inverse CDF technique. The sample elevation can be calculated using equation 18, and the longitudinal PDF is described by equation 20. Note that the three longitudinal lobes have the same form and so use the same equations, however the generic form of the variables has been substituted for the lobe specific variants below.

$$\theta_i = 2\beta \tan(\xi(A - B) + B) + 2\alpha - \theta_r \quad (18)$$

$$A = \tan^{-1} \left( \frac{\pi/4 + \theta_r/2 - \alpha}{\beta} \right) \quad B = \tan^{-1} \left( \frac{-\pi/4 + \theta_r/2 - \alpha}{\beta} \right) \quad (19)$$

$$p(\theta_i) = \frac{1}{2\cos\theta_i(A-B)} \frac{\beta}{(\theta_h - \alpha)^2 + \beta^2} \quad (20)$$

The sample azimuth angles for each of the four lobes can be calculated using the following equations

$$\phi_R = 2\sin^{-1}(2\xi - 1) \quad (21a)$$

$$\phi_{TT} = \gamma_{TT}\tan(C_{TT}(\xi - 1/2)) + \pi \quad (21b)$$

$$\phi_{TTR-g} = 2\sin^{-1}(2\xi - 1) \quad (21c)$$

$$\phi_g = \gamma_g\tan(\xi(C_g - D_g) + D_g) + \phi_g \quad (21d)$$

where  $C_{TT} = 2\tan^{-1}\left(\frac{\pi}{\gamma_{TT}}\right)$ ,  $C_g = \tan^{-1}\left(\frac{\pi/2 - \phi_g}{\gamma_g}\right)$ , and  $D_g = \tan^{-1}\left(\frac{-\phi_g}{\gamma_g}\right)$ .

## 5 Implementation

The foundation of this thesis was the development of two importance sampled hair shaders in RSL, and their use in RenderMan Pro Server (RPS) 18.0 as part of a physically plausible rendering framework.

### 5.1 Pipeline

The entire CG hair pipeline is complex and computationally demanding, consisting of modelling, simulation and rendering. Each topic is intrinsically complex and subject to its own set of problems. In hair rendering it is the sheer number of strands, combined with the strand geometry, that prove difficult. In reality, hair strands are cylindrical but it is common practice to use a curve primitive instead, as this can improve efficiency by up to one hundred fold (Pixar, 2012a).

In RenderMan the RiCurve primitive is equivalent to a flat ribbon, with normals that always face the camera as standard. The width of the ribbon along its length can be specified with a constant or varying width parameter. Although real human hair is constant in width, a decreasing width vector can be used to create the impression of hair getting thinner towards the tip. To complete the facade, a hair illumination model is applied to make the 2D ribbons look like thin 3D cylinders.

In this project, the pipeline is kept as simple as possible, and is picked up at the point where the modelled and simulated hair geometry finalised. RenderMan Pro Server 18.0 was the rendering software used, even though this version was released mid-way through the project, upgrading from RenderMan Pro Server 16.5 was not an onerous task. The shaders are written in RSL 2.0 and the pipeline is scripted using Python 2.6, make and bash scripts. Please refer to figure 5 for an example flow diagram when rendering a single frame.

The pipeline is configured through a configuration file, *pipeline.cfg*, its location can be specified in the command line arguments. The rendering pipeline is initiated by executing *run.py*, no command line arguments are required but there are a number of options available. Help for the command line options can be viewed by executing *run.py* with the *-h* command line argument.

For each rendering run, a time-stamped directory is created in the render directory. All renders and render artefacts, such as RIB files and the configuration file used, are written to, or subsequently copied into this directory. Each run is also profiled using python's `cProfile` functionality, and the result written to the *profile* file in render directory.

As standard, the hair geometry is read from RIB archives in a user specified directory. The user should also define the required camera and geometry transformations within two files, *camera\_transform.rib* and *geo\_transform.rib*, that also reside in the geometry directory. The transformation file paths can be specified in the configuration file, should requirements dictate it.

## 5.2 Hair Model

The hair model is critical to how the rendered hair will eventually look. Key factors such as hair density, style, colour, and texture as well as strand width and length are the physical properties that should be represented by the model. Therefore, a set of standard hair rendering models (Yuksel, ca.2103b) was used as all of these properties are encapsulated into a single file. The hair files are provided in binary format and include data structures that represent hair segments, points, thickness, transparency and colour. `cyHair` is a code release by Yuksel (ca.2103a), it supports the loading and saving of the afore mentioned hair files, and was used to create a simple C++ command line tool to export them to RIB format.

The *hairmodeller* tool exports point and width data to a single `RiCurve` primitive, and adds an index for each individual curve. To aid realism, the width of the curve was specified as a varying parameter in the RIB file. The width at the root is equal to the default width specified in the hair file, the tip width is half that, and all the values in-between are linearly interpolated from root to tip. The resulting tapered curve provides a much more refined result when rendered. When writing the data to RIB format a curve basis must be specified, both `catmull-rom` and `b-spline` gave acceptable results, however the `b-spline` was selected as it gave the smoothest rendered curve. Opacity and colour information is also available in the hair files, but it was not utilised as it could easily be injected at the shading stage. Additionally, the bounding box of the model and its centroid are calculated as this information is useful when computing the camera transformations.

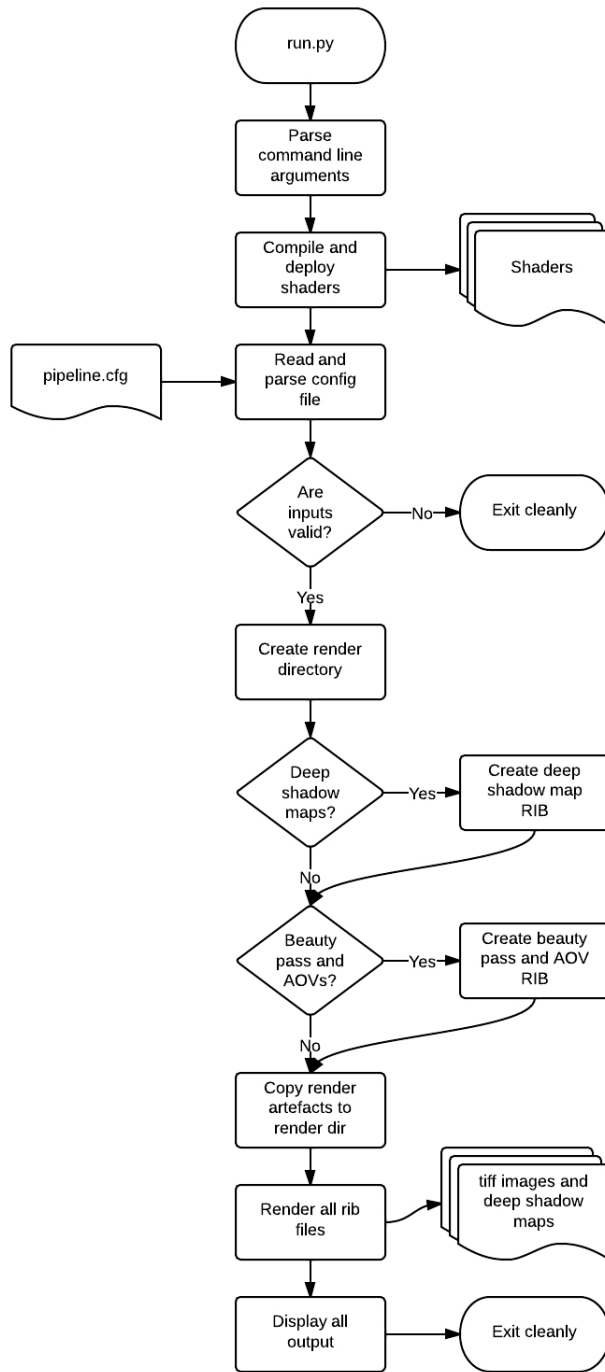


Figure 5: Pipeline flow chart for a single frame

The following hair models were exported to rib, their statistics follow in table 2.




Model	straight.hair	natural.hair	wWavyThin.hair
Reference Image			
Strands	10,000	10,000	10,000
Vertices	160,000	1,519,823	872,756
Width	0.1	0.1	0.1

Table 2: Hair Models (Yuksel, ca.2103b)

### 5.3 Shader Implementation

To compare and contrast the two most recent importance sampled hair shading models, both the Hery and Ramamoorthi (2012) model and the ISHair model (Ou et al., 2012a) were implemented in RSL. In keeping with RenderMan naming conventions, the shader names are prefixed with *plausible*: the Hery and Ramamoorthi (2012) and Ou et al. (2012a) shaders shall be referred to as *plausibleMarschner* and *plausibleIsHair* respectively. Both aim to follow the RSL 2.0 shading guidelines (Pixar, 2009), and as such they are class-based shaders and the core functionality is encapsulated into structs allowing reuse and providing performance optimisations.

The *plausibleIsHair* class is composed of a *plausibleIsHair* struct, which handles the specular sample generation and evaluation, and a *kajiyaKayDiffuse* struct that evaluates the light samples for the diffuse component. Similarly, the *plausibleMarschner* class contains a *plausibleMarschnerR* struct for the evaluation and generation of specular samples, and the same *kajiya-kay* based diffuse struct. This structure is illustrated in figure 8.

A generalised flow of the RenderMan shader methods is as follows:

```
begin()
opacity()
displacement()
```



```
prelighting()  
[lighting methods]  
postlighting()
```

However, as hair is non-displaced, the displacement method is omitted. The opacity function is also excluded from both shaders, this results in  $O_i$  being set to  $O_s$  and provides the best performance (Pixar, 2012d). The pre and post lighting methods are also not required.

In the shader classes the main lighting integrator is the *lighting* method, here all light interaction with the shading element (specular, diffuse, direct and indirect) is calculated. The lighting method is further specialised into the view-dependent (*specularlighting*) and view-independent (*diffuselighting*) methods, where the view-independent results are stored in the cache and allow the renderer to make optimisations regarding their reuse. Initially all three of the lighting methods were implemented, however low frequency noise was the result of the reuse of cached diffuse values that were not view-independent. Therefore, only the lighting method is implemented as recommended by Pixar (2012d), and the resulting method flow is simply:

```
begin()  
lighting()
```

Within this single lighting call, the computation of direct specular and diffuse lighting is handled by the *directlighting* function. The advantage of using this integrator is that the renderer deals with the details regarding importance sampling, area light sources, and it allows the sharing of certain calculations between the view dependent and independent components (Pixar, 2011). Since hair is a translucent material, the integration domain is set to "sphere".

In order for a reflectance model to be energy conserving it cannot reflect more radiance than it receives. Therefore, the proportion of reflected specular and diffuse is balanced by a user-defined factor that linearly interpolates between the two. In RPS 18.0, some of the plausible shaders use a reflectivity weight, as calculated by their Fresnel component, to balance specular and diffuse. Investigation into the effect of this should be considered in future work.

To write a shading model that can take advantage of MIS, it should be able to evaluate and generate samples, and this means implementing the *evaluateSamples* and *generateSamples* methods. Both of these methods take an array of radiance samples, and it is the responsibility of the light or material shader to fill certain properties on each sample, depending on whether the sample is being generated or evaluated. The radiance sample construct consists of the properties shown in figure 6, and MIS aware light and material shaders are required to populate the radiance sample properties shown in figure 9 during the specified function calls.

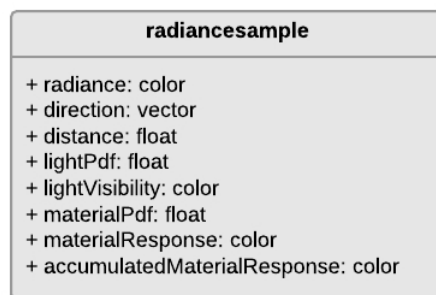


Figure 6: The Radiance Sample Construct

### 5.3.1 plausibleMarschnerR

The initialisation function is called before the direct lighting integrator performs its calculation. At this point the shading context is used to create a local coordinate system as per the notation in section 4.1. The Gram-Schmidt process is used to create an orthonormal basis (Weisstein, ca.2103b). The first basis vector is known; the shading context tangent,  $u$ . By using a hair specific index to create an arbitrary noise-based vector, each of the local coordinate systems has some variation. This arbitrary vector is projected onto the tangent following the Gram-Schmidt process to compute the second basis vector,  $v$ , and the cross-product of these two vectors completes the set,  $w$ .

The *stdrsl\_ShadingContext* is a RenderMan construct that contains information pertinent to rendering the current shading element. When evaluating or generating samples, the normalised incident ray vector and the normalised view vector are provided by the shading

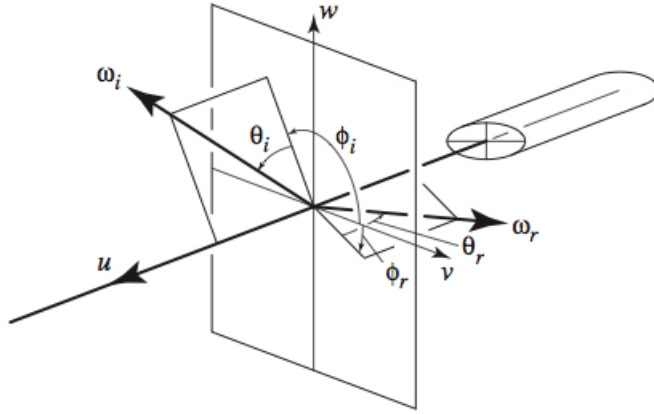


Figure 7: The hair local coordinate system (Marschner et al., 2003)

context, however they are first required to be converted into the local coordinate system, and then into spherical coordinates. Since the basis vectors are orthonormal, the coordinates of the vector with respect to new basis are  $\alpha_i = \mathbf{v} \cdot \mathbf{b}_i$  (Foley et al., 1990).

$$\mathbf{v}_{\text{local}} = (\alpha_1, \alpha_2, \alpha_3) = (\mathbf{v} \cdot \mathbf{b}_1, \mathbf{v} \cdot \mathbf{b}_2, \mathbf{v} \cdot \mathbf{b}_3) \quad (22)$$

Where the local coordinate system has basis  $B = \{b_1, b_2, b_3\}$ .

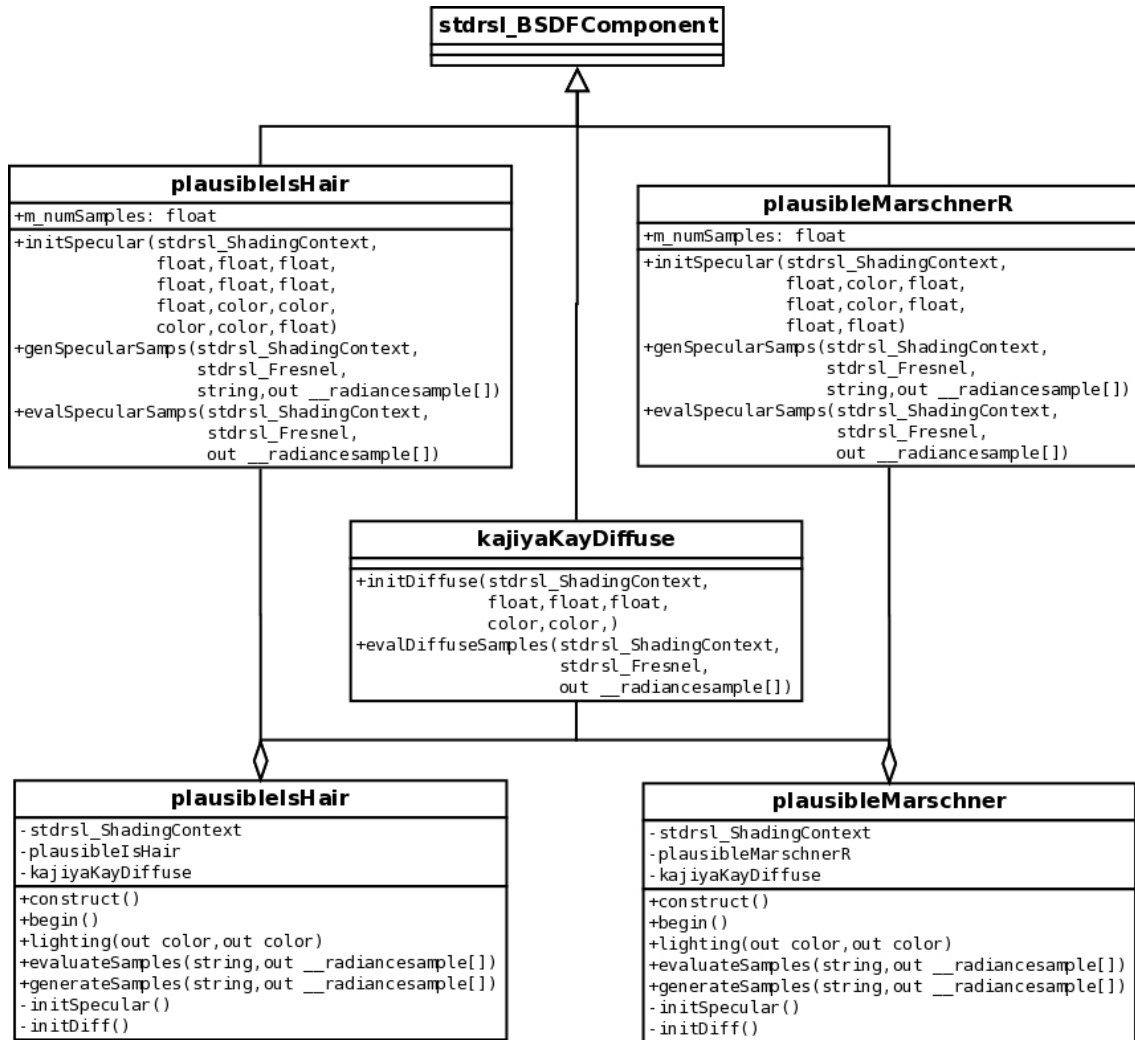


Figure 8: Shader class diagram

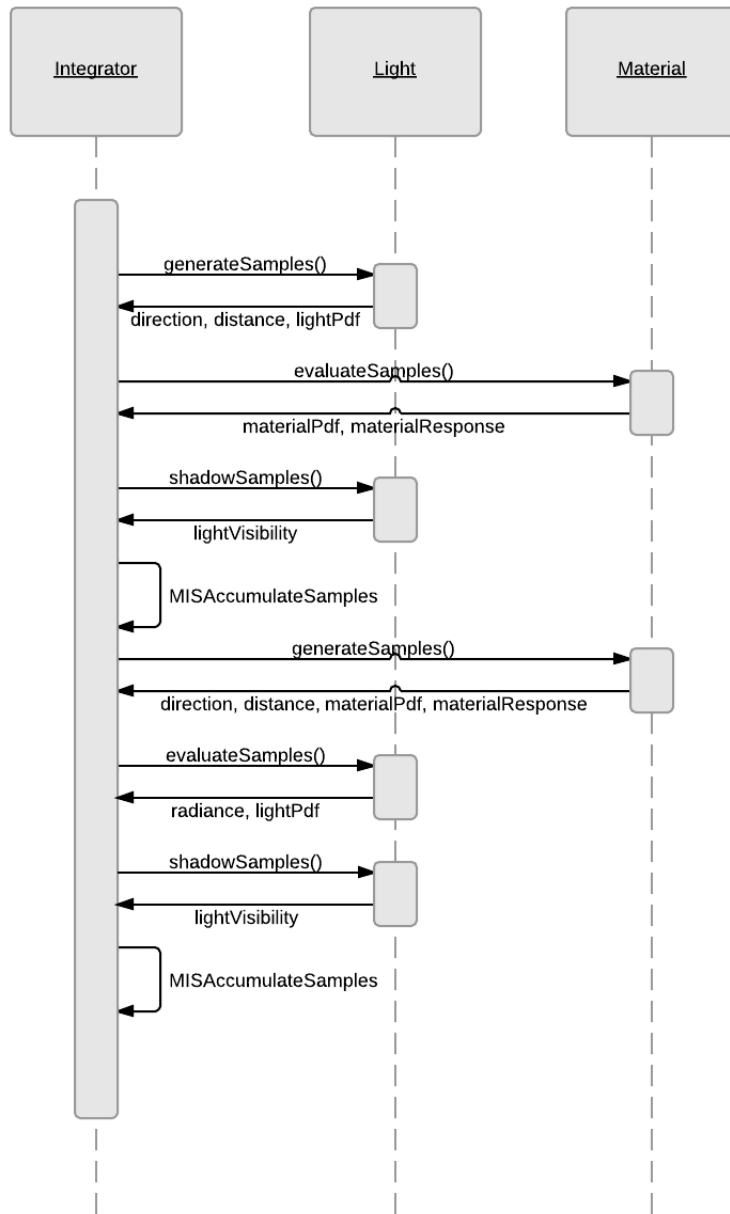


Figure 9: RenderMan MIS Shading Pipeline

To then convert into spherical coordinates, standard transformations can be applied as shown in equation 23. However, to improve performance Hery and Ramamoorthi (2012) do not always calculate the azimuth angle as such. Instead they project the vector onto the normal plane and use common trigonometric identities to reach the same result.

$$\phi = \tan^{-1}\left(\frac{w}{v}\right) \quad \theta = \frac{\pi}{2} - \cos^{-1}(u) \quad (23)$$

Once the sample direction has been calculated with respect to the local coordinate system, it should be transformed into the global space using  $\mathbf{v} = \alpha B$ .

The `generateSamples` and `evaluateSamples` are then implemented as guided by Hery and Ramamoorthi (2012), see section 4.2. Also, as per Hery and Ramamoorthi, I implemented two specular lobes that use the `plausibleMarschnerR` term. When multiple lobes are used, it is necessary to spread the sample quota over the lobes in question, and in order to ensure energy conservation they must be normalised. This can be achieved with the `normalizeMaterialResponse` function.

### 5.3.2 plausibleIsHair

The `plausibleIsHair` shader is implemented in a very similar way to `plausibleMarschner`. The local coordinate system, and transformation to and from coordinates in it, are calculated in the same way as `plausibleMarschner`, as is the conversion to and from spherical coordinates.

The four specular lobes are sampled as per the theory in section 4.3, and the samples are the uniformly distributed between the lobes (Ou et al., 2012b). Further work should include using estimate energy to distribute the samples between the lobes.

## 5.4 Shadowing

The visibility function can either be calculated directly by raytracing, or by using Pixar’s deep area shadow maps. Created in a separate pass, these maps were found to give better performance in terms of minimising noise and render times. Passed as a parameter to

the physically plausible lights, the maps are used in the visibility function and provide shadowing on both the diffuse and specular passes. The use of the deep shadow maps with the area shadow functionality was not investigated in this project, but is included in the further work in section 8.2. An example of a deep area shadow map can be seen in figure 10. The pixel window shows illustrates how the opacity of the pixel varies with increasing depth.

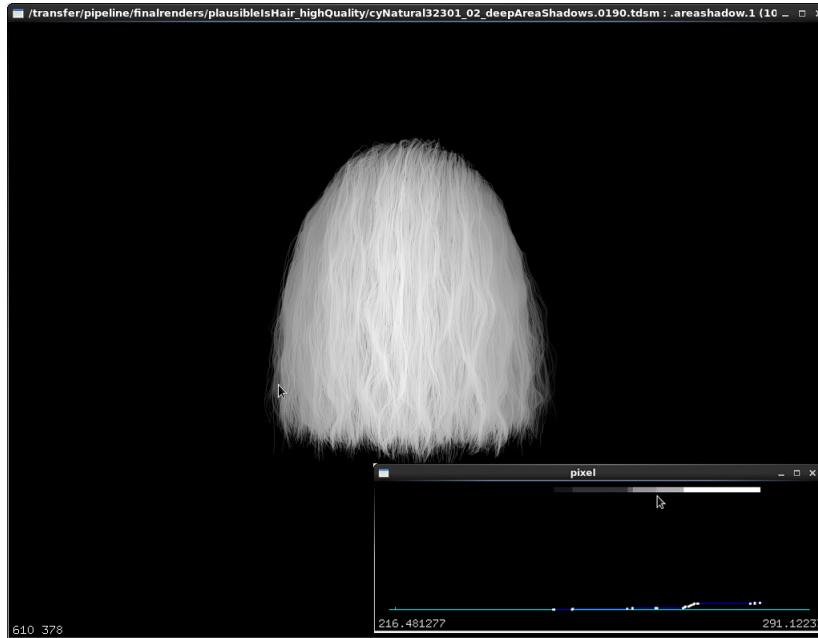


Figure 10: Deep Area Shadow Map

## 5.5 Optimisations for Hair

Pixar (2012a) provide guidance on how to ray trace RiCurves as hair in the most efficient way, and how to overcome a number of associated problems.

Hair strands should not be represented as individual RiCurve primitives, for maximum performance Pixar (2012c) recommend using approximately 10,000 curves per primitive. In addition, they advise setting the *displacement bound* to zero for non-displaced hair, as follows

```
ri.Attribute("displacementbound", {"sphere": [0]})
```

In order to create realistic occlusion for shadowing when ray tracing, the render needs to intersect rays with the 2D curves as if they were 3D geometry. RenderMan accomplishes this through the *roundcurve* and *hair dicing* attributes as demonstrated below.

```
ri.Attribute("dice", {"int roundcurve": [1], "hair": [1]})
```

Curve dicing causes derivative with respect to the curve width, *u* direction, to be zero in most cases, thus making shading curves more efficient. This assumption holds as long as curves are very thin in screen-space, such as for hair. When *roundcurve* is also applied, the curve intersects and fires rays as if it has a cylindrical surface, this can however cause performance overheads.

At the standard level-of-detail, hair strand geometry is sub-pixel in scale, this can result in aliasing artefacts. Antialiasing solutions include increasing the *pixel samples* and decreasing the *shading rate*. However, an alternative to increasing the pixel sample setting for the whole scene is to use *sigma hiding*. The default RenderMan *stochastic hider* can be adjusted to use different hiding algorithms with the *sigma* option. The alternative algorithms are optimised for use with geometry that is thin, small or translucent. The advantage of using the *sigma hider* is that the trade-off between antialiasing quality, performance and memory is harmonised (Pixar, 2013). The *sigma* option is added to the *stochastic hider* as follows, note that the *stochastic* attribute must also be used in addition to *hair dicing* for *sigma hiding* to have affect.

```
ri.Attribute("stochastic", {"int sigma": [1]})  
ri.Hider("stochastic", {"int sigma": [1], "float sigmablur": [1.0]})
```

Pixel filters can be used to help reduce noise, and when a sequence of renders was viewed there was considerable noise resulting in "sparkling". The Gaussian filter, with a three by three width was found to provide the best noise reduction.

```
ri.PixelFilter("gaussian", 3, 3)
```



As ray tracing the shadowing calculation can be very expensive, deep area shadow maps can be used to improve performance. When creating deep area shadow maps, hairs can be made at least one pixel wide and have some degree of transparency applied. This can be achieved by setting the *minwidth* option, where the minimum value is given in pixels.

```
ri.Option("hair", {"float minwidth": [1.0]})
```

To preserve optimum performance when generating deep area shadow maps, the beauty-pass roundcurve and sigma hiding optimisations should be disabled. The roundcurve attribute could cause ray bias towards the shadow camera, and sigma hiding could cause the collapse of depth data.

## 6 Results

### 6.1 Environment Lighting

The hair models were rendered under three different environment lighting conditions: indoor, outdoor (bright sunlight), outdoor (soft sunlight). The environment maps used are the Grace Cathedral (USC, ca.2103a) and Pisa (USC, ca.2103b) HDRI, available from The University of Southern California, and the Woodland Lane HDRI (Aversis, ca.2013). All environments have been turned into latitude/longitude environment maps using txmake. Shader settings have not been changed between test cases, and so all variation comes entirely from the environment map lighting.

The render times are considerably longer for `plausibleIsHair`, this may be due to the optimisations made within `plausibleMarschner` and also to its relative simplicity. The shaders respond well to the different environment maps, outdoors the lighting is generally more diffuse and the images show softer specular highlights that pick up the colour of the environment. In particular, the colour of the sky in the Woodland lane scene, and the light bouncing off the terracotta coloured wall in Pisa. As there is a bright sun source in the Woodland lane HDRI, some sharp bright primary specular highlights would have also been expected, however adjusting the shader parameters did not produce the desired effect.



Figure 11: plausibleMarschner in Woodland Lane: 1024 lighting and 1024 materials samples. 24 minutes render time.

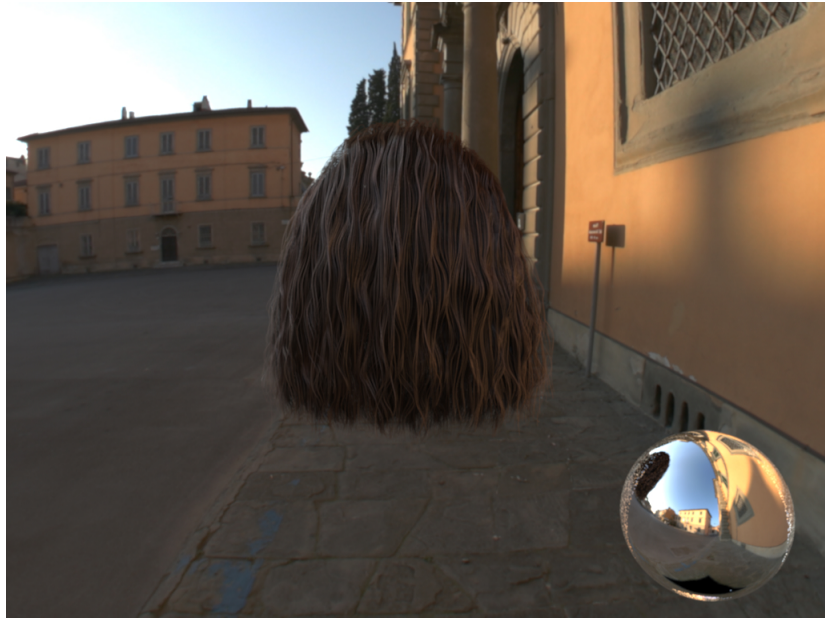


Figure 12: plausibleMarschner in Pisa: 1024 lighting and 1024 materials samples. 25 minutes render time.



Figure 13: plausibleMarschner in Grace: 1024 lighting and 1024 materials samples. 23 minutes render time.



Figure 14: plausibleIsHair in Woodland Lane: 1024 lighting and 1024 materials samples. 45 minutes render time.

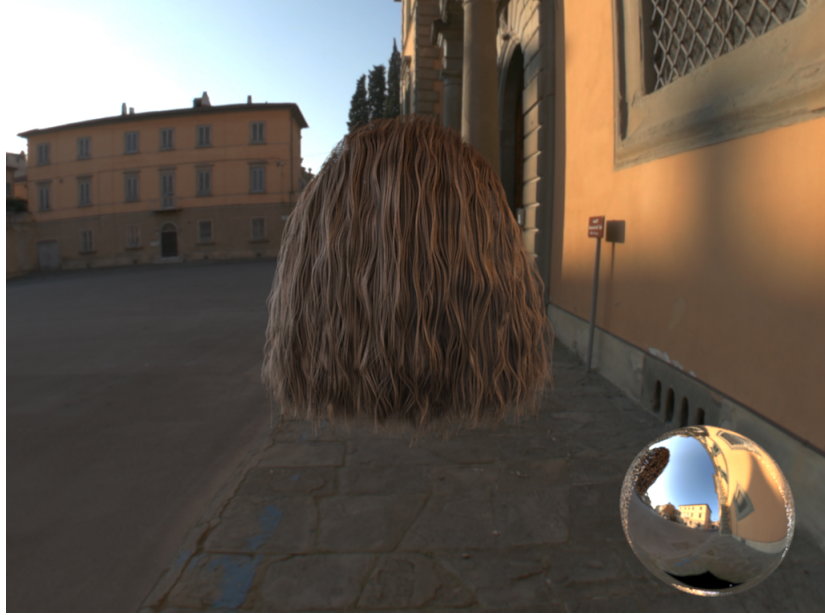


Figure 15: plausibleIsHair in Pisa: 1024 lighting and 1024 materials samples. 46 minutes render time.



Figure 16: plausibleIsHair in Grace: 1024 lighting and 1024 materials samples. 44 minutes render time.

## 6.2 Samples

To assess how the number of samples affects the image quality and performance, increasing numbers of light and then material samples were used to render images and the render performance profiled. The invariant sample number is set to 64. The number of samples against performance is compared in table 3 and in figure 17.

Light samples	Material samples	Run time (CPU seconds) plausibleMarschner	Run time (CPU seconds) plausibleIsHair
16	64	94.0	277.5
64	64	151.6	250.6 <sup>1</sup>
256	64	355.6	463.8
512	64	623.3	745.6
1024	64	1148.8	1283.3
2048	64	2238.7	2415.3
64	16	137.1	168.8
64	64	151.6	250.6
64	256	206.1	536.5
64	512	287.5	920.7
64	1024	424.8	1690.6
64	2048	756.9	3289.0

Table 3: The number of samples against performance.

The affect of increasing light samples when using the plausibleMarschner shader can be seen in figure 18, and similarly, the affect of increasing material samples can be seen in figure 19. The affect of increasing light samples when using the plausibleIsHair shader can be seen in figure 20, and similarly, the affect of increasing the material samples can be seen in figure 21. The context of the comparisons is shown in Appendix A, figures 27 and 28.

It becomes apparent that light samples affect image quality more than material samples with plausibleMarschner. A possible cause may be the effectiveness of the importance sampling scheme implemented, however it is unusual for there to be no obvious improvement in image quality with a substantial increase in samples. Compared to plausibleIsHair, where the material samples have a much more noticeable effect, this is unexpected and the cause of this inconsistency suggests further investigation is required as to its cause.

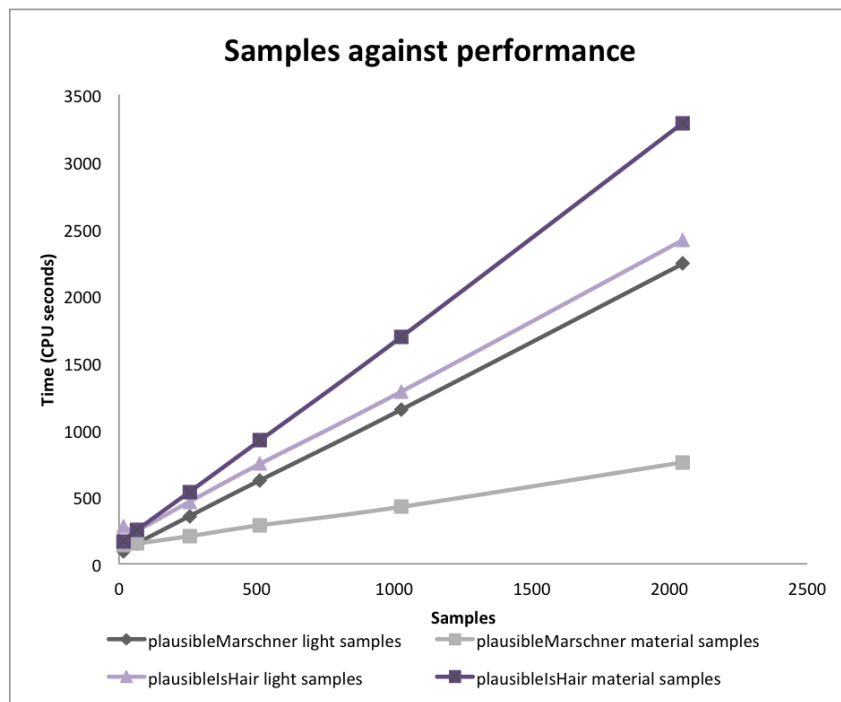


Figure 17: The number of light and material samples against run time



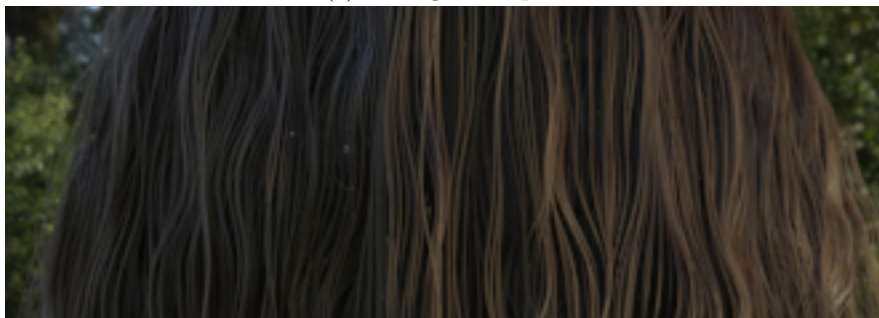
(a) 16 light samples



(b) 64 light samples



(c) 256 light samples



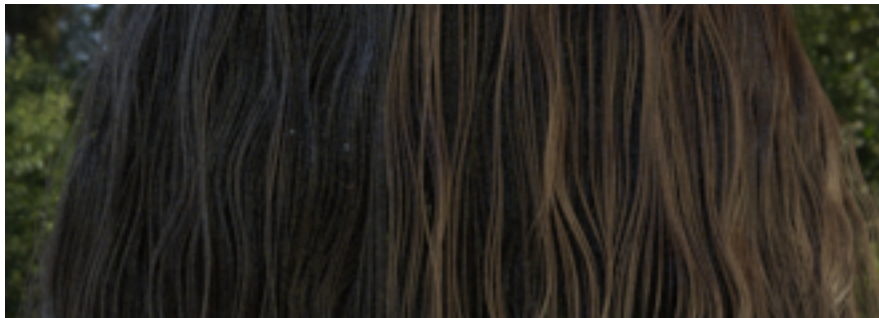
(d) 1024 light samples

Figure 18: The effect of light samples on image quality for plausibleMarschner





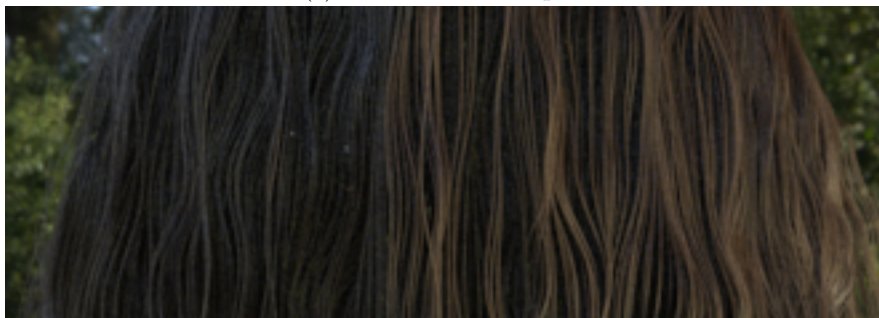
(a) 16 material samples



(b) 64 material samples



(c) 256 material samples



(d) 1024 material samples

Figure 19: The effect of material samples on image quality for plausibleMarschner



(a) 16 light samples



(b) 64 light samples



(c) 256 light samples



(d) 1024 light samples

Figure 20: The effect of light samples on image quality for plausibleIsHair



(a) 16 material samples



(b) 64 material samples



(c) 256 material samples



(d) 1024 material samples

Figure 21: The effect of material samples on image quality for plausibleIsHair

### 6.3 Indirect Lighting

Global illumination with area lighting in a Cornell Box. The effect of global illumination enhances realism, and in this example the plausibleIsHair shader demonstrates indirect lighting effects by gathering radiance from the scene. The number of indirect diffuse samples greatly increased render times.



Figure 22: Cornell Box Test for plausibleIsHair

### 6.4 Ray Traced Shadows

Ray traced shadows continue to be problematic under environment lighting. The level of noise generated is unacceptable, as shown in figure 23, and attempts to reduce it by increasing the number of samples only exacerbated the issue, see figure 24.



Figure 23: Ray Traced Shadows for plausibleIsHair: 1k light and material samples



Figure 24: Ray Traced Shadows for plausibleIsHair: 8k light and material samples

## 6.5 High Quality Renders



Figure 25: High Quality Render for plausibleIsHair: 8k light and material samples and Deep Area Shadows



Figure 26: High Quality Render for plausibleMarschner: 8k light and material samples and Deep Area Shadows

## 7 Conclusions

Investigating physically based rendering provided the opportunity to learn how to use RenderMan in a different way: class-based shaders were written in RSL 2.0, and the physically plausible shading pipeline and new pipeline methods were utilised in conjunction with importance sampling.

Upon starting this project, initial research did not discover any publicly available importance sampled hair shaders, however a plausible hair shader was released with RPS 18.0. This hair shader, `plausibleBasicHair`, ended up being a useful comparison and the entire plausible shader library provided guidance in how to get started with plausible shader writing.

At present it is difficult to provide meaningful analysis of the importance sampled shaders as there are no benchmark results to compare against. If the two shaders, `plausibleMarschner` and `plausibleIsHair`, were implemented with a uniform sampling mode, then the trade-off between quality and render times could be more closely examined. Additionally, the original Marschner shader, that was developed during the initial stages of this project, could be tested in a non-importance sampled lighting setup to gauge the benefits of optimising the hair shaders for ray tracing.

On the whole the still image quality is good and the shaders respond well to the physically based lights and test render scenes. However, there is still much to be done to achieve photo-realistic results. In particular, the shadowing needs some attention. Ray tracing the visibility function to determine self-shadowing results in sampling artefacts and the overall appearance requires more shadow depth. Increasing the minimum width option for hair doesn't improve this, however investigation into this in conjunction with other RenderMan options may yield better results. Deep area shadow maps provided greater depth to the image, however in some cases it made the hair look to separated and in some areas there appeared to be too much shadow. Experiments with the number of layers recorded in the deep area shadow map, from 10 to 100, and changing the opacity of the hair when creating the maps did not make much difference.

The specular reflection colour requires further work. At present the two shaders provide very different results despite having the same colours and intensities provided for the primary and secondary specular reflections. Also, the specular balance parameter is based



on a user input and was adjusted by eye, this could be made more physically correct by having a Fresnel-based coefficient balance the specular and diffuse components.

Rendering hair also brought up problems that had not been encountered before, in particular the aliasing artefacts of rendering very small geometry. This area required continuous investigation and tweaking of options and attributes, and there is still much room for improvement. When observing the moving image arbitrary output variables (AOVs) it is apparent that the majority of the noise comes from the shadowing functionality.

In terms of usability, `plausibleIsHair` has many parameters and whilst this may provide a seasoned artist with the flexibility they need, it required time to understand the effect of each of the parameters and the large combination of choices left a greater margin for error. Compared with the `RenderMan` `plausibleBasicHair` shader, which has fewer user inputs and provided a good result out-of-the-box, `plausibleIsHair` seems to break one of the key principles of the physically plausible paradigm, that of ease of use.

## 8 Known Issues and Further Work

The research conducted only scratches the surface of physically based hair rendering, here follows the known bugs, issues and suggested further work.

### 8.1 Known issues

As mentioned previously, the shadows require more work to create a realistic image. There is a sampling problem with ray tracing the visibility function, and deep area shadow maps create shadows that are too dark and defined. The area shadow functionality is a worthy line of investigation to improve this.

The sequence of moving images reveals a problem with noise, it is visible in both the diffuse and specular passes for both shaders. It is possible that improved shadow generation could improve this problem, and also improved understanding and further investigation of `RenderMan`'s antialiasing options could help.

There is a bug in `plausibleMarschner`, when converting to spherical coordinates the wrong coordinate index is used. This was discovered part way through the final renders and so

was not fixed at the time of discovery.

The current diffuse term is not view independent, therefore the benefits of caching the diffuse results are not received.

## 8.2 Further Work

First and foremost, comparison of the importance sampled shaders with the uniformly sampled versions, in terms of both image quality and performance, would provide further insight into how importance sampling can improve performance when ray tracing. Also creating the equivalent results with the original Marschner and non-physically plausible environment light would provide additional evidence for use in this comparison. To further examine all shaders for energy conservation the *white furnace* lighting test can be used (Hery and Ramamoorthi, 2012). Implementing this for both the plausible and non-plausible shaders would facilitate better understanding of the extent to which they are energy conserving.

During the initial stages of this project the intention was to also implement the d'Eon et al. (2011) energy conserving model. As it remains partially complete, a short-term goal is to finish it and test it against the original Marschner model.

In `plausibleIsHair` each lobe is sampled, and the samples are then uniformly distributed between lobes, however in the paper, Ou et al. (2012a) suggest distributing them based on the estimate energy produced by each of the lobes. Another of the extensions suggested by Ou et al. is to substitute the Cauchy distribution for the Gaussian distribution when calculating the BSDF.

There are some additions to be made to both of the shaders: adding support for indirect specular, and balancing specular and diffuse on the Fresnel based coefficient.

The pipeline would benefit from some robustness improvements, especially regarding the geometry specific transformations required by certain components, such as the lighting.

Finally, at the beginning of this project the research was intended to be completed in both RenderMan and Arnold Renderer, however only RenderMan was attempted. Porting the shaders to Arnold SL and implementing the pipeline with Arnold would provide a deeper understanding of the concepts and theory of physically based rendering.

## References

- Aversis. Free 50 megapixel hdri map - 01. "<http://www.aversis.be/hdri/hdri-free-50mp-01.htm>", ca.2013. Accessed: 29 July 2013.
- Eugene d'Eon, Guillaume Francois, Martin Hill, Joe Letteri, and Jean-Marie Aubry. An energy-conserving hair reflectance model. In *Proceedings of the Twenty-second Eurographics conference on Rendering*, EGSR'11, pages 1181–1187, Aire-la-Ville, Switzerland, Switzerland, 2011. Eurographics Association. doi: 10.1111/j.1467-8659.2011.01976.x. URL <http://dx.doi.org/10.1111/j.1467-8659.2011.01976.x>.
- James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer graphics: principles and practice (2nd ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990. ISBN 0-201-12110-7.
- FXPhd. Fxphd presents physically plausible shading - part 1. "<http://www.youtube.com/watch?v=b8LFBX4x4qE>", 2013. Accessed: 29 July 2013.
- Dan B. Goldman. Fake fur rendering. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 127–134, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. ISBN 0-89791-896-7. doi: 10.1145/258734.258807. URL <http://dx.doi.org/10.1145/258734.258807>.
- Christophe Hery and Ravi Ramamoorthi. Importance sampling of reflection from hair fibers. *Journal of Computer Graphics Techniques*, 2012. URL <http://graphics.berkeley.edu/papers/Hery-ISO-2012-06/>.
- N. Hoffman. Background: Physically-based shading. In *ACM SIGGRAPH 2010 Courses*, SIGGRAPH '10, New York, NY, USA, 2010. ACM. URL [http://renderwonk.com/publications/s2010-shading-course/hoffman/s2010\\_physically\\_based\\_shading\\_hoffman\\_a\\_notes.pdf](http://renderwonk.com/publications/s2010-shading-course/hoffman/s2010_physically_based_shading_hoffman_a_notes.pdf).
- N. Hoffman. Background: Physics and math of shading. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, pages 10:1–10:7, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1678-1. doi: 10.1145/2343483.2343493. URL <http://doi.acm.org/10.1145/2343483.2343493>.
- J. T. Kajiya and T. L. Kay. Rendering fur with three dimensional textures. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, SIG-

- GRAPH '89, pages 271–280, New York, NY, USA, 1989. ACM. ISBN 0-89791-312-4. doi: 10.1145/74333.74361. URL <http://doi.acm.org/10.1145/74333.74361>.
- Tae-Yong Kim and Ulrich Neumann. Opacity shadow maps. In *In Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 177–182. Springer-Verlag, 2001.
- Tom Lokovic and Eric Veach. Deep shadow maps. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 385–392, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. ISBN 1-58113-208-5. doi: 10.1145/344779.344958. URL <http://dx.doi.org/10.1145/344779.344958>.
- Stephen R. Marschner, Henrik Wann Jensen, Mike Cammarano, Steve Worley, and Pat Hanrahan. Light scattering from human hair fibers. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 780–791, New York, NY, USA, 2003. ACM. ISBN 1-58113-709-5. doi: 10.1145/1201775.882345. URL <http://doi.acm.org/10.1145/1201775.882345>.
- Jonathan T. Moon and Stephen R. Marschner. Simulating multiple scattering in hair using a photon mapping approach. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 1067–1074, New York, NY, USA, 2006. ACM. ISBN 1-59593-364-6. doi: 10.1145/1179352.1141995. URL <http://doi.acm.org/10.1145/1179352.1141995>.
- Jonathan T. Moon, Bruce Walter, and Steve Marschner. Efficient multiple scattering in hair using spherical harmonics. *ACM Trans. Graph.*, 27(3):31:1–31:7, August 2008. ISSN 0730-0301. doi: 10.1145/1360612.1360630. URL <http://doi.acm.org/10.1145/1360612.1360630>.
- Ivan Neulander. Fast furry ray gathering. In *ACM SIGGRAPH 2010 Talks*, SIGGRAPH '10, pages 2:1–2:1, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0394-1. doi: 10.1145/1837026.1837029. URL <http://doi.acm.org/10.1145/1837026.1837029>.
- Jiawei Ou, Feng Xie, Parashar Krishnamachari, and Fabio Pellacini. Ishair: Importance sampling for hair scattering. *Comp. Graph. Forum*, 31(4):1537–1545, June 2012a. ISSN 0167-7055. doi: 10.1111/j.1467-8659.2012.03150.x. URL <http://dx.doi.org/10.1111/j.1467-8659.2012.03150.x>.
- Jiawei Ou, Feng Xie, Parashar Krishnamachari, and Fabio Pellacini. Supplemental mate-

- rial for ishair. "<http://www.cs.dartmouth.edu/~ouj/papers/hairis/supplemental.pdf>", 2012b. Accessed: 15 August 2013.
- M. Pharr and G. Humphreys. *Computer graphics: principles and practice (2nd ed.)*. Morgan Kaufmann, Burlington, MA, USA, 2010. ISBN 978-0-12-375079-2.
- Pixar. Rsl 2.0 shading guidelines. "<http://renderman.pixar.com/resources/current/rps/rslBestPractices.html>", 2009. Accessed: 29 July 2013.
- Pixar. Physically plausible shading in rsl application note. "<http://renderman.pixar.com/resources/current/rps/physicallyPlausibleShadingInRSL.html>", 2011. Accessed: 29 July 2013.
- Pixar. Using the ricurves primitive. "<http://renderman.pixar.com/resources/current/rps/appnote.19.html>", 2012a. Accessed: 01 August 2013.
- Pixar. Area light and area shadowing support. "<http://renderman.pixar.com/resources/current/rps/areaShadow.html>", 2012b. Accessed: 10 August 2013.
- Pixar. Optimized rendering. "<http://renderman.pixar.com/resources/current/rps/appnote.3.html>", 2012c. Accessed: 04 August 2013.
- Pixar. Using renderman shading language pipeline methods: What, why, and when. "<http://renderman.pixar.com/resources/current/rps/shaderPipelineWhatWhyWhen.html>", 2012d. Accessed: 12 August 2013.
- Pixar. Rps 18 documentation: Options. "<https://renderman.pixar.com/resources/current/rps/options.html>", 2013. Accessed: 04 August 2013.
- Iman Sadeghi and Rasmus Tamstorf. Efficient implementation of the dual scattering model in renderman. Technical report, Walt Disney Animation Studios, 2010.
- Iman Sadeghi, Heather Pritchett, Henrik Wann Jensen, and Rasmus Tamstorf. An artist friendly hair shading system. In *ACM SIGGRAPH 2010 papers*, SIGGRAPH '10, pages 56:1–56:10, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0210-4. doi: 10.1145/1833349.1778793. URL <http://doi.acm.org/10.1145/1833349.1778793>.
- Mike Seymour. Monsters university: rendering physically based monsters. "<http://www.fxguide.com/featured/>

- monsters-university-rendering-physically-based-monsters”, 2103a. Accessed: 24 July 2013.
- Mike Seymour. The state of rendering - part 1. ”<http://www.fxguide.com/featured/the-state-of-rendering>”, 2103b. Accessed: 29 July 2013.
- USC. Grace cathedral hdr. ”<http://gl.ict.usc.edu/Data/HighResProbes/>”, ca.2103a. Accessed: 29 July 2013.
- USC. Pisa courtyard near sunset hdr. ”<http://gl.ict.usc.edu/Data/HighResProbes/>”, ca.2103b. Accessed: 29 July 2013.
- Hector Vazquez-Leal, Roberto Castaneda-Sheissa, Uriel Filobello-Nino, Arturo Sarmiento-Reyes, and Jesus Sanchez Orea. High accurate simple approximation of normal distribution integral. *Mathematical Problems in Engineering*, 2012:1 – 22, 2012. ISSN 1024123X. URL <http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=87029080&site=eds-live&scope=site>.
- Eric W. Weisstein. Box-muller transformation. from mathworld—a wolfram web resource. ”<http://mathworld.wolfram.com/Box-MullerTransformation.html>”, ca.2103a. Accessed: 4 August 2013.
- Eric W. Weisstein. Gram-schmidt orthonormalization. from mathworld—a wolfram web resource. ”<http://mathworld.wolfram.com/Gram-SchmidtOrthonormalization.html>”, ca.2103b. Accessed: 4 August 2013.
- Eric W. Weisstein. Gaussian function. from mathworld—a wolfram web resource. ”<http://mathworld.wolfram.com/GaussianFunction.html>”, ca.2103c. Accessed: 6 August 2013.
- Cem Yuksel. cyhair code release. ”<http://www.cemyuksel.com/cyCodeBase/code.html#cyHair>”, ca.2103a. Accessed: 11 August 2013.
- Cem Yuksel. Hair model files. ”[www.cemyuksel.com/research/hairmodels](http://www.cemyuksel.com/research/hairmodels)”, ca.2103b. Accessed: 11 August 2013.
- Cem Yuksel and John Keyser. Deep opacity maps. *Computer Graphics Forum*, 27(2): 675–680, 2008. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2008.01165.x. URL <http://dx.doi.org/10.1111/j.1467-8659.2008.01165.x>.

Arno Zinke, Cem Yuksel, Andreas Weber, and John Keyser. Dual scattering approximation for fast multiple scattering in hair. *ACM Trans. Graph.*, 27(3):32:1–32:10, August 2008. ISSN 0730-0301. doi: 10.1145/1360612.1360631. URL <http://doi.acm.org/10.1145/1360612.1360631>.

Appendix A



Figure 27: 16 light samples and 64 material samples for plausibleMarschner



Figure 28: 16 light samples and 64 material samples for plausibleIsHair