Fundamentals of Computer Graphics - Hair Shading Project

June 2, 2023

Donato Francesco Pio Stanco

Abstract

The aim of this project is to add support for hair shading following PBRT by refactoring it into Yocto/GL and providing the proper abstractions.

1. Introduction

This project aims to implement the hair scattering model described in the PBRT chapter. Hair is a challenging material to render realistically due to its complex geometry and intricate light interaction. Improving the hair shading model in PBRT will enable more accurate and visually appealing representations of various hair types, ranging from straight to curly, and from light to dark.

By refining the hair shading techniques in PBRT, it is possible to achieve better light transmission, absorption, and scattering effects. This will involve the implementation of advanced algorithms and methodologies that simulate the complex behaviour of light as it interacts with individual hair strands, considering properties such as hair thickness, density, and colour.

2. Related works

Physically Based Rendering A related work is the one described in this book (Pharr & Humphreys, 2010) where a method known as *literature programming* combines human-readable documentation and source code into a single reference that is specifically designed to aid comprehension and the result is a stunning achievement in graphics education.

3. Method

The implementation follows straightforwardly the one presented in the PBRT chapter but respecting the Yocto/GL convention. In the following subsections, I will describe all the problems encountered during the development and the modifications to the path tracer.

Code It is possible to view all the code and all the rendered images in the Project repository.

3.1. Scattering from Hair

First, I needed to know the main structure of the **hair BSDF model** and essentially hair and fur have three main components:

- **Cuticle**: the outer layer, which forms the boundary with air. The cuticle's surface is a nested series of scales at a slight angle to the hair surface;
- **Cortex**: the next layer inside the cuticle. The cortex generally accounts for around 90% of hair's volume but less for fur. It is typically coloured with pigments that mostly absorb light;
- **Medulla**: the centre core at the middle of the cortex. It is larger and more significant in thicker hair and fur. The medulla is also pigmented. Scattering from the medulla is much more significant than scattering from the medium in the cortex.

Incident light arriving at a hair may be scattered one more time before leaving the hair. So **p** is used to denote the number of path segments it follows inside the hair before being scattered back out to air.

So the **hair BSDF** is described as a sum over term p:

$$\sum_{p=0}^{\infty} f_p(\omega_0, \omega_i) \tag{1}$$

In the PBRT implementation, only the first few terms of the sum of equation (1) are evaluated, representing all higher-order terms with only one.

3.2. Geometry

I decided to convert `.pbrt` scenes into Yocto/GL `.json` scene format, since Yocto/GL does not support **Bézier**

Email: Donato Francesco Pio Stanco <stanco.2027523@studenti.uniroma1.it>.

Fundamentals of Computer Graphics, Sapienza University of Rome, 1st semester a.y. 2022/2023.

curves, so I approximated them with straight lines. I used two and four lines for each Bézier curve to render straight and curly hairs, respectively. I also stored curve tangents and linear interpolated widths for each vertex. To optimize rendering performances, I joined all the lines belonging to the same model into one only `.ply` shape. I tested the implementation using hair models from Benedikt Bitterli's Rendering Resources. (Bitterli, 2016).

3.3. Hair Material

I extended the `material` structure present in Yocto/GL with the parameters needed for hair shading. These parameters are:

- **sigma_a**: the absorption coefficient;
- **beta_m**: the longitudinal roughness, 0.3 by default;
- **beta_n**: the azimuthal roughness, 0.3 by default;
- **alpha**: the hair scale angle in degrees, 2 by default;
- **eta**: the index of refraction of the interior of the hair, 1.55 by default;
- **enumelian**: the enumelian concentration;
- **pheomelian**: the pheomelian concentration.

Hair **colour** can be specified in three different ways: directly with the `colour` parameter, through the absorption coefficient `sigma_a` or with the concentration of `eumelanin` and/or `pheomelanin`, which are the two pigments that determine the colour in human hair. An **eumelanin** concentration of about 8 gives black hair, 1.3 brown hair and 0.3 blonde hair. Instead **pheomelanin** is responsible for orange and red hairs.

3.4. Model Implementation

As said, the implementation follows straightforwardly the one presented in the PBRT chapter. Almost all the code is inside the `yocto_extension.h` and `yocto_extension.cpp` files and to respect the Yocto/GL convention I implemented *three* new functions:

- `eval_hair_scattering()`: evaluates hair BRDF lobes according to the input incoming and outgoing directions. As for the other functions which evaluate BRDF lobes, I folded the product by the cosine between the incoming direction and the shading normal inside this function;
- `sample_hair_scattering()`: given an outgoing direction, samples an incoming direction according to the hair BRDF;

The third function is `eval_hair_brdf()` which is responsible to evaluate the input hair material together with the v coordinate of the ray-line intersection and to return the corresponding hair BRDF lobes. Since PBRT computations are made in the BRDF coordinate system, I decided to build a frame to convert from world to local BRDF coordinate systems and vice versa. The z-axis of this frame is orientated along the shading normal, while the x-axis is orientated along the line tangent.

3.5. Modifications to the Yocto/GL Path Tracer

I also added modifications to the Yocto/GL path tracer in order to adapt it to the hair shading. More specifically I modified the following files:

- `yocto_scene.h`: here I added hair material specification in struct `material_data` and `material_point`;
- `yocto_scene.cpp`: here I added implementation for hair material in `eval_material()` function;
- `yocto_geometry.h`: here I added implementation to compute the hair normal according to PBRT;
- `yocto_trace.cpp`: here I modified some functions of the original path tracer, more specifically:
  - `eval_bsdfcos()`: it used to evaluate/sample the BRDF scaled by the cosine of the incoming direction;
  - `sample_bsdfcos()`: it is used to pick a direction based on the BRDF;
  - `sample_bsdfcos_pdf()`: it is used to compute the weight for sampling the BRDF.

4. Results

In this section, I will show the generated images obtained with the hair shading. I did several tests and comparisons for **Longitudinal Scattering**, **Absorption in fibers**, **Azimuthal Scattering** and **Scales on Hair Surfaces**. I also did a test regarding **sloth fur** to verify the differences between the hair shading implementation and the none one.

All hair images have been rendered at the resolution of **720x720** pixels with **1536** samples per pixel.

4.1. Longitudinal Scattering

The main challenge with this look was choosing a roughness v to achieve the desired look. As mentioned in PBRT there is a mapping $\beta_m \in [0, 1]$ to v where a roughness of 0 is nearly perfectly smooth and 1 is very rough. Longitudinal scattering is responsible for the highlight along the length of hair and the roughness controls the sharpness of this highlight. In Figure 1 it is possible to observe the different renders for some values of β_m. As it is possible to observe with very low roughness (Figure 1a) the hair appears too shiny, almost metallic. In Figure 1b the highlight is similar to typical human hair and in Figure 1c the hair is unrealistically flat and diffuse.

4.2. Absorption in Fibers

The A_p term describes how much of the incident light is affected by each of the scattering **modes** p. Now let's analyze the produced images:

- in Figure 2a σ_a is set to $\{3.35, 5.58, 10.96\}$ (RGB coefficients) that corresponding to black hair, almost all transmitted light is absorbed;
- in Figure 2b σ_a is set to $\{0.84, 1.39, 2.74\}$ (RGB coefficients), where $p > 1$, that corresponding to brown hair;
- in Figure 2c σ_a is set to $\{0.06, 0.10, 0.20\}$ (RGB coefficients), very low absorption coefficient, which corresponds to blonde hair.

So as I modified the absorption coefficient σ_a I obtained different colours of hairs.

4.3. Azimuthal Scattering

This is the component of scattering that depends on the **angle** ϕ and Azimuthal Scattering model is based on first computing a new azimuthal direction assuming perfect specular reflection and transmission and then defining a distribution of directions around this central direction, where increasing roughness gives a wider distribution.

In Figure 3 are reported the produced images with different values of β_n and as the azimuthal roughness increases, hair gets brighter. So for instance Figure 3a is less brighter than Figure 3c since β_n is smaller.

4.4. Scales on Hair Surface

In the end, I analyzed the Scales on Hair Surface that are responsible for the secondary coloured highlight below the white one. Now let's analyze the produced images, reported in Figure 4:

- in Figure 4a the α is set to 0 degrees;
- in Figure 4b the α is set to 2 degrees.

As is possible to see, the second one is much brighter than the second one since the value of α is higher than the first one.

4.5. Different renderings

I also made another test with and without hair shading as reported in Figure 5. As is it possible to observe the fur of the sloth in Figure 5a is defined and brighter than Figure 5b. This image has been rendered with a resolution of **1920** pixels and with **1536** samples.

5. Test Functions

In the PBRT chapter have described some tests that I decided to write in order to check the implementation and also to validate the abstractions made with the path tracer. The test functions are reported at the end of the `yocto_extension.cpp` file and the code passes successfully both the test for energy conservation and the one which validates the sampling routine. These tests have been useful to find and fix some errors I made during the implementation. The test functions are:

- `white_furnance_test()`: this function checks if hair doesn't absorb any of the light passing through it (i.e., $\sigma_a = 0$);
- `white_furnance_sampled_test()`: this function samples incident directions rather than a uniform distribution, dividing BSDF values by the PDF to compute the estimate of reflectance.

These two functions are important since if one `white_furnace_test` fails or both fail, I can have a better idea of whether the underlying problem is in the evaluation of the model or in the code that samples it.

The other two functions are:

- `sampling_weights_test()`: this function checks the variety of roughness of the hair and the random sample values;
- `sampling_consistency_test()`: this function checks if the number of samples is sufficient, and should return the same result both if is used the custom importance sampling scheme and if it is used a uniform distribution of directions over the unit sphere.

6. Conclusions

This is the implementation of hair shading that follows the PBRT ones. My implementation works as shown in the results in Figures (1, 2, 3, 4, 5) very well with human hair and also with animal fur. One possible improvement will be to add support for rendering realist-looking humans by exploiting the hair shading.

References

Bitterli, B. Rendering resources, 2016. https://benedikt-bitterli.me/resources/.

Pharr, M. and Humphreys, G. *Physically Based Rendering, Second Edition: From Theory To Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2010. ISBN 0123750792.

(a) Rendered with $\beta_m = 0.1$

(b) Rendered with $\beta_m = 0.25$

(c) Rendered with $\beta_m = 0.6$

Figure 1. Longitudinal Scattering - Rendering

(a) Black Hair with $\sigma_a = \{3.35, 5.58, 10.96\}$

(b) Brown Hair with $\sigma_a = \{0.84, 1.39, 2.74\}$

(c) Blonde Hair with $\sigma_a = \{0.06, 0.10, 0.20\}$

Figure 2. Absorption in Fibers - Rendering

(a) Rendered with $\beta_n = 0.3$

(b) Rendered with $\beta_n = 0.6$

(c) Rendered with $\beta_n = 0.9$

Figure 3. Azimuthal Scattering - Rendering

(a) Rendered with $\alpha = 0$ degrees

(b) Rendered with $\alpha = 2$ degrees

Figure 4. Scales on Hair Surfaces - Rendering

(a) Rendered with hair shading

(b) Rendered without hair shading

Figure 5. Sloth - Rendering