Worksheet 8

In previous exercises, we assumed that the reflection from a transparent surface such as glass is always 10%. We also assumed that all light is reflected from or transmitted through a transparent object. This is an idealised model. In reality, the reflectance (the amount of reflection) depends on the angle of incidence of the ray and some light is lost due to absorption. The transmittance (the amount of light that is not absorbed) depends on the distance that the ray travels through the medium. In this set of exercises, we include physically based Fresnel reflectance and absorption in the ray tracing framework.

Learning Objectives

- Implement shaders for rendering transparent and glossy objects.
- Simulate the angle-dependency of reflectance as it appears in transparent and glossy objects.
- Simulate absorption of light as a function of the distance that a ray moves through a medium.
- Use Fresnel's equations for reflection and Bouguer's law of exponential attenuation.

Fresnel Reflectance

To capture the angle-dependency of reflectance R, we need the Fresnel equations. The form of these equations which is most easily used in graphics is [**B**, Section 26.5 and 27.8.1]:

$$\begin{aligned} \tilde{r}_\perp &= \frac{n_i \cos\theta_i - n_t \cos\theta_t}{n_i \cos\theta_i + n_t \cos\theta_t} \\ \tilde{r}_\parallel &= \frac{n_t \cos\theta_i - n_i \cos\theta_t}{n_t \cos\theta_i + n_i \cos\theta_t} \\ R &= \frac{1}{2}\left(\tilde{r}_\perp^2 + \tilde{r}_\parallel^2\right) \ , \end{aligned}$$

where n_i is the refractive index of the medium from which the incident ray reaches the surface, n_t is the refractive index of the medium that the ray transmits into, θ_i is the angle of incidence, and θ_t is the angle of refraction.

- Use Fresnel reflectance R to determine the amount of reflection and refraction in transparent materials. Do this by implementing the functions in the file `fresnel.h`. Use the function `fresnel_R` in the overloaded `trace_refracted` function of the file `RayTracer.cpp`. This overloaded function includes an argument `R` for returning reflectance. Set $R = 1$ in the case of total internal reflection. Render an image of the default scene where you return the result of `Transparent::shade` from the `Glossy::shade` function and compare it to the result from when R was set to 10%.
- Use Fresnel reflectance in your glossy shader (`Glossy.cpp`) and in your photon tracing (`Particle-Tracer.cpp`). Render images of the default scene and compare them to the previous results where R was kept constant (include renderings of the caustic photon maps as white dots).

Absorption

To capture absorption, we need Bouguer's law for computing the transmittance T_r of a light ray that moves a certain distance s through a medium. Bouguer's law is [**B**, Section 27.13.1]:

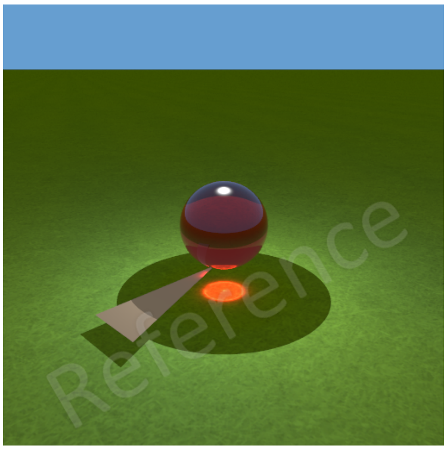
$$L = L_0 T_r = L_0 \exp(-\sigma_t s) \ ,$$

where σ_t is called the extinction coefficient. In a transparent medium, the extinction coefficient equals the absorption coefficient ($\sigma_t = \sigma_a$). To let the user specify the absorption coefficient in an intuitive way, we let her set the diffuse reflectance ρ_d, and we use the following formula to transform it into an absorption coefficient

$$\sigma_a = \frac{1}{\rho_d} - 1 \ .$$

Please note that you need to handle the discontinuity ($\rho_d = 0$). One approach is to use conditionals.

- Use Bouguer's law of exponential attenuation to capture absorption in transparent materials. Do this by implementing the functions `shade` and `get_transmittance` in the file `Volume.cpp`.
- Add absorption to the glass ball in the default scene. Do this by opening the file `default_scene-.mtl` in the `models` folder of the framework and setting the diffuse reflectance `Kd` of the `glass` material to some colour of your own choice. You also need to set the `illum` parameter to 12, as this will ensure that the framework calls the volume shader when rendering glass materials.
- Combine the shader that captures absorption with the glossy shader that includes a Phong highlight. Do this by implementing the `shade` function in the file `GlossyVolume.cpp`. Note that you may need to stitch together parts of your other shaders to exclude the Lambertian reflection that a non-zero ρ_d (or `Kd`) would normally cause.
- Implement absorption in the photon tracing. Do this by copying your implementation of the `get-_transmittance` function to the function of the same name in the file `ParticleTracer.cpp`. Then use it in the function `trace_particle` to modify the power of a photon that has travelled through an absorbing medium (`illum` > 10). Render a coloured glass ball floating over a lawn beside a cardboard triangle as the final result.



Worksheet 8 Deliverables

Renderings of the default scene (e.g. using Fresnel reflectance without highlights or caustics, caustics photon map, using Fresnel reflectance with highlights and caustics, including absorption). Compare to previous results. Include relevant code snippets. Please insert all this into your lab journal.

Reading Material

The curriculum for Worksheet 8 is (15 pages)

B Chapter 24. *Reflection Models*.