Worksheet 8

In this worksheet, we will look at colours due to dispersion. Theoretically, dispersion is a simple phenomenon. Light rays of different wavelengths are perceived as light of different colours. The index of refraction is wavelength dependent. This means that rays of different colour are refracted differently. Implementing dispersion in an RGB ray tracer is, however, a little troublesome. The problem is that we usually trace one ray for the combined RGB representation of the colour, not one per wavelength. In the following, we will disperse rays when they refract into dispersive materials to simulate colours due to dispersion.

Learning Objectives

- Simulate dispersion (the classical dispersion prism experiment, for example).
- Integrate dispersion with an RGB ray tracing and photon mapping solution.
- Transform from spectrum to RGB using the CIE RGB colour matching functions.

Photon Mapping

The classical experiment where light is dispersed by a glass prism to produce a spectrum of colours on a diffuse surface is infeasible to render using traditional path tracing. Therefore we resort to photon mapping to render this scene.

- Load a scene with a dispersion prism (`prism1.obj`) into your ray tracer. If you try to render it, you will see that this scene is not handled well by any of the previously implemented rendering methods.
- The first step is to disperse photon packets when they intersect a dispersive material. The simplest approach is to use Russian roulette to sample one of the three colour bands. Implement RGB dispersion of photon packets in your particle tracer. (In the `pathtrace` project of the framework, implement the `disperse_particle` function in `ParticleTracer.cpp`. This function is called when a material has the wavefront OBJ illumination model set to `illum 5`.)
- Render the caustics photon map. You should see some dispersion colours on the diffuse screen, but the result is probably very noisy. Tune the number of caustics photons to be traced and the number to be used in the radiance estimate (see lines 72 and 74 in `RenderEngine.cpp`). Once you get smooth caustics on the diffuse screen, switch to the complete photon mapping solution, render the scene, and store the resulting image. (In the framework, you also have to increase the maximum number of photons that the particle tracer is allowed to trace before terminating. Do this in line 71 of `RenderEngine.cpp`. Note that the tuning will increase start-up time for the program. So the numbers should be readjusted after you finish working on this set of exercises.)
- Compare the result to the photograph in the paper by Sun et al. [2000, see reference below]. The likeness is disappointing as long as traditional crown glass is used for the prism. Load a scene with a dense flint glass prism (`prism2.obj`) and render it. Store the resulting image.
- While the result is better with the dense flint glass prism, it looks a bit like a spot of blue light, a spot of green light, and a spot of red light. The next step is to do spectral dispersion. The index of refraction is given for a number of wavelengths. Importance sample a wavelength using a step function (Russian roulette) based on the CIE RGB colour matching functions. These functions are also used to transform the monochromatic sample back into RGB. Use the sampled wavelength to get an index into the array of refractive indices. Be careful when handling the start and endpoints of the visible spectrum. (In the framework, modify your implementation of the `disperse_particle` function.)

- With spectral dispersion you probably need even more caustics photons. Tune the parameters once again and store an image rendered using the complete photon mapping solution. Compare the new result to the reference photograph.

Photon Differentials (Optional)

If you experimented with photon differentials in the last worksheet, you can optionally try to use them for this prism example as well.

- Trace photon differentials alongside the photons (In the framework, implement the `disperse-_particle` function in `EyeDiffTrace.cpp`. It is almost the same as in `ParticleTracer-.cpp` except that now photon differentials need to be passed to the trace functions.)
- Adjust parameters for the photon differentials to suit this dispersion prism scene (a large number of photons need to be traced and a fairly large smoothing parameter s is probably needed). Compare the result to the result obtained with regular photon mapping. Results should be comparable to the results in the paper by Frisvad et al. [2014, see reference below].

Worksheet 8 Deliverables

Images of the classical dispersion experiment using: a crown glass prism and RGB dispersion, a dense flint glass prism and RGB dispersion, and a dense flint glass prism but spectral dispersion. The latter one both with and without density estimation techniques. Include relevant code and render log (number of samples, render time, number of photons traced and in radiance estimate, etc.). Explain the differences between the photograph in the paper by Sun et al. [2000, see reference below] and the rendering results. Answer the following question:

Why is dense flint glass better than crown glass for dispersion prisms?

Reading Material

The curriculum for Worksheet 8 is

- Sun, Y., Fracchia, F. D., and Drew, M. S. Rendering Light Dispersion with a Composite Spectral Model. In *Proceedings of the First International Conference on Color in Graphics and Image Processing (CGIP)*, 2000.
- Frisvad, J. R., Schjøth, L., Erleben, K., and Sporring, J. Photon differential splatting for rendering caustics. *Computer Graphics Forum 33*(6), pp. 252–263, September 2014.

Additional resources uploaded to CampusNet:

- Tilley, R. Colour Due to Refraction and Dispersion. In *Colour and the Optical Properties of Materials*, second edition, Chapter 2, pp. 49–90, John Wiley & Sons, 2011.