Assignment 3: Global Illumination

Due date: 1st of May 2014, 12:00

The goal of this assignment is to extend your raytracer to support indirect sources of light. **You must also turn in your work through Ilias**.

Required Features

Path Tracing (30 points)

Implement Monte Carlo path tracing as discussed in class. Use Russian roulette to terminate paths. You can use a fixed probability to terminate paths.

Testing (10 points)

Test your path tracer by rendering images of the Cornell box. Build and render the scene in your renderer. You can find scene specifications in various file formats on the web, for example here in Inventor format. Produce and save images with direct illumination only, and with global illumination using your path tracer. Generate images with different numbers of paths per pixel. Store the images and the render times so you can show them to us during the presentation.

Also make a modified box including a glossy or mirror object. Render and store images as described above.

Construct at least one interesting scene of your own to demonstrate the effect of global illumination. Show us renderings with and without indirect illumination from the path tracer. Show us images at different sampling rates to illustrate convergence.

Russian Roulette on Shadow Rays (10 points)

As an improvement to the basic path tracer from above, also apply Russian roulette to skip shooting shadow rays that make a low contribution to the final image. Tentatively compute the potential contribution of each shadow ray. If the contribution is below some threshold, apply Russian roulette to possibly skip tracing the ray. Prepare comparisons with and without this optimization and compare the image quality at (approximately) the same render time for both methods.

Bidirectional Path Tracing (40 points)

Implement bidirectional path tracing, where paths are traced from the eye and from the light sources. Extend your implementation by providing multiple importance sampling. We will discuss more details of this algorithm in class and during the exercise session on April 10.

Testing (10 points)

Use the two versions of the Cornell box scene (with and without a glossy or mirror object) as described above to test your implementation. Render images with different numbers of paths and compare uniform weighting of paths with multiple importance sampling.

Compose an additional scene with refractive objects generating caustics. Compare the results of path tracing and bidirectional path tracing at equal sample counts.

Hacker's Bonus (max. 40 points)

- Implement photon mapping. Ask us for advice if you attempt this! We will provide Java or C code for the photon kd-tree.
- Implement irradiance caching. We will discuss this in the class session on April 17th.
- Implement Perlin noise textures. See Wikipedia for more links. A Java implementation is available here.
- Add support for bitmap textures and bump mapping.
- Write an importer for scenes modeled in a 3D modeling program.