Worksheet 4

The key subject this week is Monte Carlo integration, that is, to solve integrals by sampling. With sampling comes the ability to render soft shadows and indirect illumination. We will focus on soft shadows in this worksheet.

Learning Objectives

- Apply Monte Carlo integration in rendering.
- Sample points on a triangle mesh.
- Sample directions using a cosine-weighted hemisphere.
- Render soft shadows including the effects of ambient occlusion.

Ray Tracing

Worksheet 1 included an area light, but what if the light source is more complicated. We often need to carefully consider the sampling of light sources to avoid very high variance results. We will in the following assignments work on sampling techniques.

- Load the Cornell box (`CornellBox.obj`) and the blocks inside it (`CornellBlocks.obj`) and render an image of the Cornell box where the blocks cast soft shadows. Pick a number of samples per pixel that results in smooth soft shadows. Make sure that your sampling estimates the value of the direct lighting integral properly using Monte Carlo integration theory.
  For the sampling, try sampling a position $\boldsymbol{x}_j$ on an area light with probability $\mathrm{pdf}(\boldsymbol{x}_j) = \frac{1}{N_\Delta}\frac{1}{A_\Delta}$, where N_Δ is the number of triangles and A_Δ is the area of the sampled triangle. Then ensure that you sample the area light uniformly (with probability $\mathrm{pdf}(\boldsymbol{x}_j) = \frac{1}{A}$, where A is the total surface area). (CPU framework: update the `sample` function in `AreaLight.cpp` and, if necessary, the `shade` function in `Lambertian.cpp`. GPU framework: update `sample` in `AreaLight.h` and `__closesthit__arealight` in `shaders.cu`.)
- Load the Stanford bunny (`bunny.obj`) together with the Cornell box and change its material, so that it acts as an area light (give it non-zero ambient, Ka, in the .mtl file). Compare the two different sampling techniques for this scene with two very different area lights.
- Return to the scene from Worksheet 2 (bunny and sky). Ambient occlusion is essentially the idea that the environment (the background) is an infinitely distant ambient area light. Compute ambient occlusion by tracing a ray in a direction sampled on the hemisphere over each surface point. Consider the colour returned by the sky model to be the incident illumination if the ray is not occluded. (Implement `sample_cosine_weighted` in `sampler.h`. CPU framework: implement the `shade` function in `Ambient.cpp`. Note that the ambient occlusion shader is used when you press '2' on the keyboard. GPU framework: implement the INDIRECT section of `__closesthit__directional`.)
- Use the panoramic environment map from Worksheet 2 and update the holdout shader so that it includes ambient occlusion as well as the shadow cast by the sun. Pick a number of samples that results in smooth illumination and save the image. (CPU framework: update the `shade` function in `Holdout.cpp`. GPU framework: implement the INDIRECT section of `__closesthit__holdout`.).

Worksheet 4 Deliverables

Cornell box images (with blocks that cast soft shadows) and with a bunny that is also an area light. Images of the bunny in an environment illuminated by sun and sky and in a photographed environment. Include relevant code and please give details about the number of samples per pixel.

Reading Material

The curriculum for Worksheet 4 is

P Sections 13–13.3 and 13.5–13.6.5. *Monte Carlo Integration*.
P Section 14.2–14.2.3. *Sampling Light Sources*. (Soft shadows.)
P Section 14.3. *Direct Lighting*.

Alternative literature available online or uploaded to CampusNet:

- Dutré, P. *Global Illumination Compendium*. Lecture Notes, Katholieke Universiteit Leuven, September 2003. https://people.cs.kuleuven.be/~philip.dutre/GI/
- Shirley, P., Wang, C., and Zimmerman, K. Monte Carlo techniques for direct lighting calculations. *ACM Transactions on Graphics 15*(1), pp. 1–36, 1996. https://doi.org/10.1145/226150.226151

Additional resources:

- Landis, H. Production-ready global illumination. In RenderMan in Production, ACM SIGGRAPH 2002 Course Notes, Chapter 5, pp. 87-101, 2002.
- Pharr, M., and Green, S. Ambient Occlusion. In *GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics*, Chapter 17, Addison-Wesley, 2004. https://developer.nvidia.com/gpugems/gpugems/part-iii-materials/chapter-17-ambient-occlusion