CS 667 Problem 2.

You should do this problem in pairs. If you have not taken CS665, please team up with someone who has.

This problem is about computing direct illumination using Monte Carlo integration. We'll look at the special case of irradiance on the axis-aligned square with corners at $(-1, -1, 0)$ and $(1, 1, 0)$ from a spherical light source of radiance 1.0 $\mathrm{W}/(\mathrm{m}^2\mathrm{sr})$ and radius R at position $\mathbf{c}$. You are welcome to use whatever environment you like for the computations, including Java, C++, or Matlab.

1. Suppose we use each of the following probability distributions to generate the directions for integration at a particular point:

   (a) Uniform with respect to solid angle σ over the hemisphere.

   (b) Uniform w. r. t. projected solid angle μ over the hemisphere.

   (c) Uniform w. r. t. area on the source (over the visible area only).

   (d) Uniform w. r. t. projected area on the source (over the visible area only).

   For each case, choose an integration measure and state the integral that needs to be solved, then express the pdf in that measure and give the estimator you will use.

2. Write a program to compute irradiance using these schemes. Its inputs should be $\mathbf{c}$, R, the choice of scheme, a flag for whether to use stratification, and the square root of the number of samples per pixel. The outputs should be the irradiance at $(0, 0, 0)$, the variance of the estimator, and the variance of the output (computed by repeating the computation 100 times). Verify your computation against analytical values for two cases: light source directly overhead (center $(0, 0, 1)$, radius 0.5), and light source in the plane (center $(2, 0, 0)$, radius 1.0).

3. What is the relative performance of the four schemes for small, distant sources? For large, nearby sources? When does stratification help the most?

4. Add to your program the option to include an occluding square in the x-z plane with corners at $(0, 0, -0.25)$ and $(0, 0.5, 0.25)$ and the ability to compute an irradiance map over the whole square and output it as an image. Render irradiance maps with the shadow of the occluder in them.