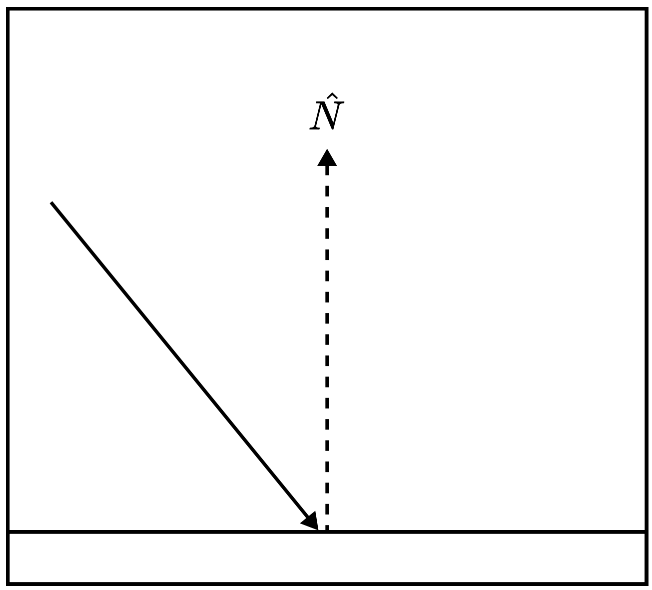
Stanford CS248A: Computer Graphics
Written Assignment 4

Understanding BRDFs

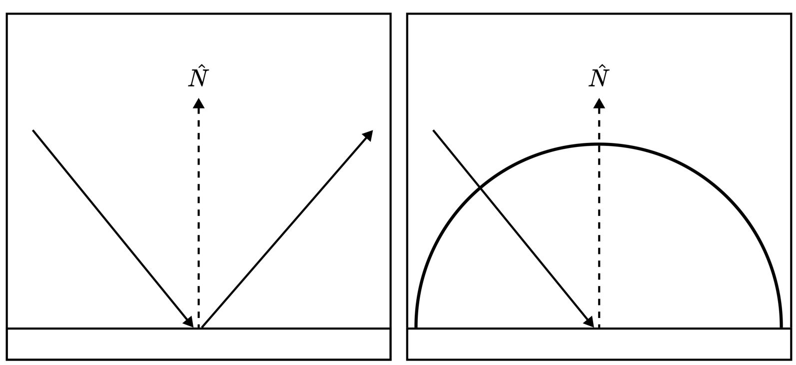
Problem 1: (Graded for Correctness - 20 pts)

A. (10 pts) As discussed in-class, an ideal diffuse surface is the special kind of surface where light that hits it is scattered **equally in all directions**. Now, picture a perfect mirror surface, or what we called an "ideal specular" surface in class. How would you explain the distribution of light that reflects off a mirror? In your answer please use terms like "reflection about the surface normal".

B. (10 pts) Sketch a BRDF diagram for a material that reflects non-zero light in all directions, but has a greater amount of light in the mirror reflection direction. Recall these diagrams plot the directional distribution of light when light hits the surface from the given angle. (See the two example diagrams at the bottom.) **After drawing your sketch, describe what a sphere surface with this BRDF looks like if you are looking at the sphere from a position where the only light source in the room is a point light source is directly behind you. (For simplicity, assume you are not casting a shadow on the sphere.)**



Hint: Here are some BRDF diagrams we've covered in-class.



Perfect Mirror BRDF **Ideal Diffuse BRDF**

Understanding Path Tracing

Problem 2: (Graded for Correctness - 15 pts)

Consider the path tracing algorithm to compute global illumination discussed in class (slide 58 of lecture 13). The code is copied below for convenience, WITH ONE MODIFICATION INDICATED IN THE CODE COMMENT BELOW: **When sampling the bounce direction, the implementation ALWAYS CHOOSES AS RANDOM RAY DIRECTION, REGARDLESS OF THE CURRENT BSDF**.

```
Spectrum PathLo(Ray ray) {
  Spectrum Lo = 0, beta = 1;
  int depth = 0;
  while (true) {
    Intersection isect = scene->Intersect(ray);
    Vector3f wo = -ray.d;
    if (depth == 0) Lo += isect.Le(wo);

    BSDF bsdf = isect.GetBSDF();
    Lo += beta * ReflFromDirectLighting(bsdf, wo);

    Spectrum fr = bsdf.Sample_f(wo, &wi, &pdf);  // ASSUME THE IMPLEMENTATION OF Sample_f
                                                 // ALWAYS CHOOSES A RANDOM RAY DIRECTION
                                                 // FROM THE HEMISPHERE ABOVE THE SURFACE
                                                 // NORMAL WITH PROBABILITY 1/2PI.

    beta *= fr * Dot(wi, isect.N) / pdf;

    float q = 0.25;
    if (randomFloat() < q) break;
    else beta /= (1-q);

    depth++;
    ray = Ray(isect.P, wi);
  }
  return Lo;
}
```

Consider a scene that contains only point light sources (lights that are infintesimally small points). Why is it the case that although the code above is an unbiased global illumination ray tracer, with extremely high probability, perfect mirror surfaces in the scene will appear black?

Diving into a Path Tracer

Problem 3: (Graded for Correctness - 15 pts)

Consider the following pseudocode for a path tracer. It is the same code shown in lecture on slide 58. It employs Russian Roulette to terminate paths with probability q.

```
// return radiance along ray 'ray'
Spectrum PathLo(Ray ray) {
  Spectrum Lo = 0, beta = 1;
  int depth = 0;
  while (true) {
    Intersection isect = scene->Intersect(ray); // find scene intersection
    Vector3f wo = -ray.d;
    if (depth == 0)         // *** QUESTION IS ABOUT THIS LINE ***
      Lo += isect.Le(wo);

    BSDF bsdf = isect.GetBSDF();

    // accumulate reflectance due to direct lighting
    Lo += beta * ReflFromDirectLighting(bsdf, wo);

    // generate new ray direction wi, and evaluate BSDF given wo and wi.
    Spectrum fr = bsdf.Sample_f(wo, &wi, &pdf);

    // update path throughput before next step along path
    beta *= fr * Dot(wi, isect.N) / pdf;

    float q = 0.25;
    if (randomFloat() < q)
       break;         // terminate path
    else
       beta /= (1-q); // update path throughput

    depth++;
    ray = Ray(isect.P, wi);
  }

  return Lo;
}
```

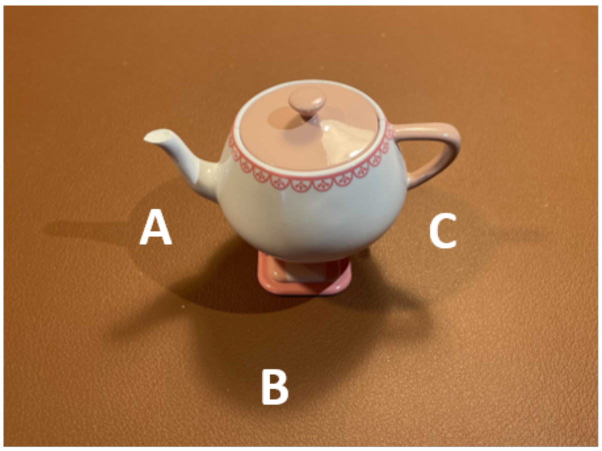
A. (7 pts) Notice the line of code with the comment `*** QUESTION IS ABOUT THIS LINE ***`. Please explain why the algorithm only accumulates surface emission into the path's output radiance if the path depth is 0. Note that path depth = 0 means this is a camera ray.

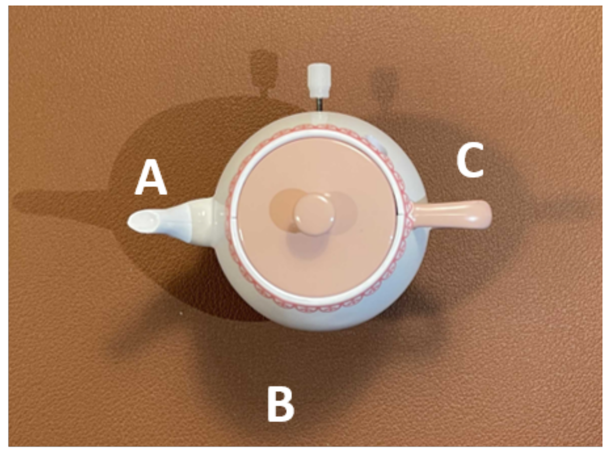
B. (8 pts) In your own words, why is it generally a good idea to special case the sampling of direct lighting (like in the code above), rather than implement the simple form of path tracing psuedocoded on slide 45?

A Photo of a Teapot

Problem 4: (Graded on Effort Only - 20 pts)

One night Kayvon decides to take a photo of a cute teapot model he has on his desk. To illuminate the teapot he uses three light sources. The first two are identical size desk lamps each with a large light emitting bulb–they are *area light* sources. But one lamp is placed closer to the teapot than the other). The third source is a tiny LED flashlight on his cellphone. Below are two photos of the teapot under these lighting conditions (a front view and top view). Which shadow (A,B,C) comes from which light source? Please explain why. Note that there is also a base level of ambient illumination in the room.

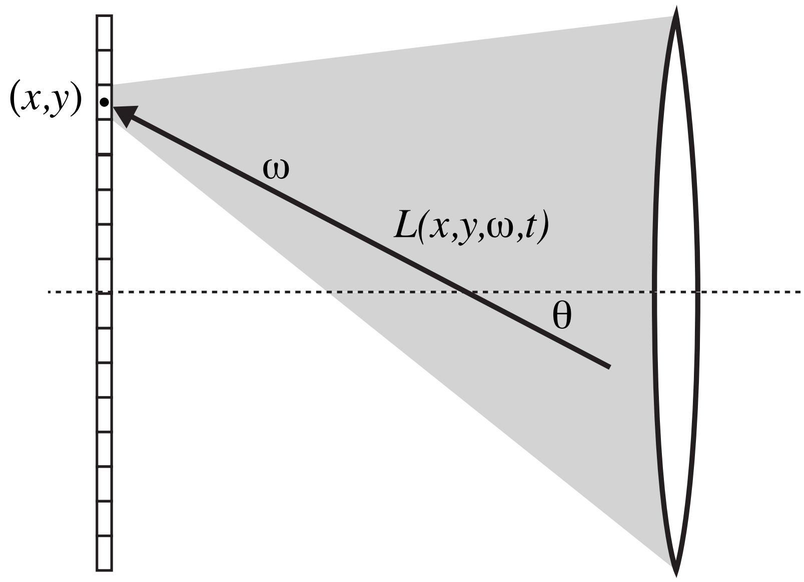




A More Realistic Camera

Problem 5: (Graded on Effort Only - 30 pts) 30

In your ray tracer assignment, you implemented an approximation to a simple pinhole camera that is exposes for an infinitesimally small amount of time. Now we are going to simulate a more realistic camera, in which a) pixels have finite area, b) the light hitting the pixel doesn't come through a pinhole, but can pass through any point on the camera's lens, and c) pixels are exposed for a finite amount of time. (That is, pixels capture incoming light energy for a total duration of T seconds.) This more realistic camera model is illustrated in the figure below. The gray region of the figure illustrates all the light rays that hit a single pixel after coming through the lens. Also on the figure is one example ray hitting the pixel at the specific point (x, y) from direction ω.



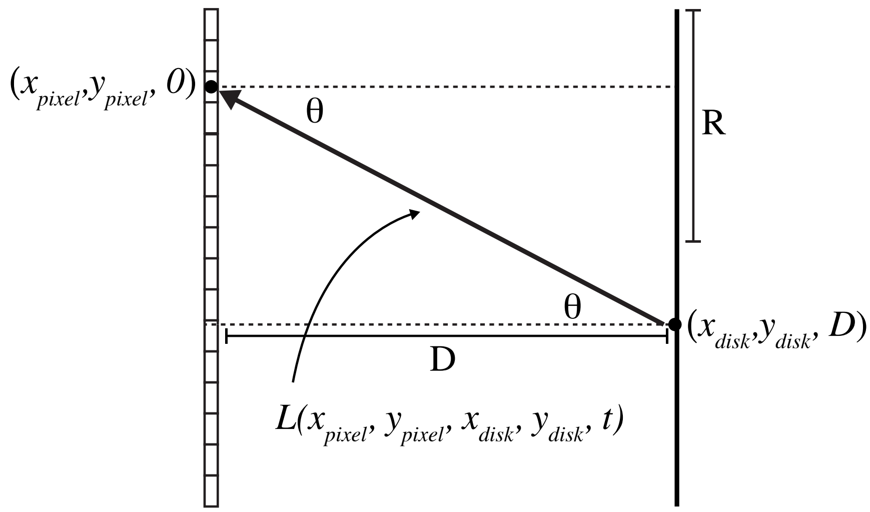
The question is on the next page...

The total **energy** hitting the pixel is given by the triple integral below. In the equation below $L(x, y, \omega, t)$ is the incident **radiance** on the sensor at point (x, y) from the direction ω at time t. Note that the radiance is a function of t, since the scene can change over the duration the exposure is taking place. The domain of the first integral is the total exposure time T. The domain of the second integral A_{pixel} is the area of a pixel. The domain of the third integral H^2 is the hemisphere of directions looking out from the sensor.

$$Q = \int_T \int_{A_{\text{pixel}}} \int_{H^2} L(x, y, \omega, t) \cos\theta \, \mathrm{d}\omega \, \mathrm{d}A_{pixel} \, \mathrm{d}t$$

A. (10 pts) Explain why this is the correct equation for computing the total energy hitting the pixel. In your answer explain the purpose of each integral and the cos term. Note that in the notation above, we're using x and y to denote the location of the current differential patch of pixel area being integrated over.

B. (10 pts) Now imagine you want to estimate this integral using Monte Carlo integration. Rather than uniformly sample the hemisphere of directions (with respect to solid angle), you decide to do something a little different: uniformly sample the area of a circular disk that is located at the back of the lens, as illustrated below. The disk has radius R and is a distance D from the sensor.



Rewriting the integral from the previous problem as an integral over the area of the disk and not the hemisphere of directions, we get:

$$
\begin{aligned}
Q &= \int_T \int_{A_{\text{pixel}}} \int_{A_{\text{disk}}} L(x_{\text{pixel}}, y_{\text{pixel}}, x_{\text{disk}}, y_{\text{disk}}, t) \cos\theta \left(\frac{\mathrm{d}A_{\text{disk}} \cos\theta}{\left(\frac{D}{\cos\theta}\right)^2} \right) \mathrm{d}A_{\text{pixel}} \, \mathrm{d}t \\
&= \int_T \int_{A_{\text{pixel}}} \int_{A_{\text{disk}}} \frac{L(x_{\text{pixel}}, y_{\text{pixel}}, x_{\text{disk}}, y_{\text{disk}}, t) \cos^4\theta}{D^2} \, \mathrm{d}A_{\text{disk}} \, \mathrm{d}A_{\text{pixel}} \, \mathrm{d}t
\end{aligned}
$$

Now, as you know, a photograph/render involves solving this integral for each pixel. You might notice that corners/edges of photographs sometimes appear darker; this effect is called vignetting. Given the above equation, please explain this phenomenon.

C. (10 pts) **THIS QUESTION IS NOT DEPENDENT ON A CORRECT ANSWER TO PARTS A AND B.** Assume that the integral you are trying to estimate is:

$$\int_T \int_{A_{\text{pixel}}} \int_{A_{\text{disk}}} f(x_{\text{pixel}}, y_{\text{pixel}}, x_{\text{disk}}, y_{\text{disk}}, t)\, \mathrm{d}A_{\text{disk}}\, \mathrm{d}A_{\text{pixel}}\, \mathrm{d}t$$

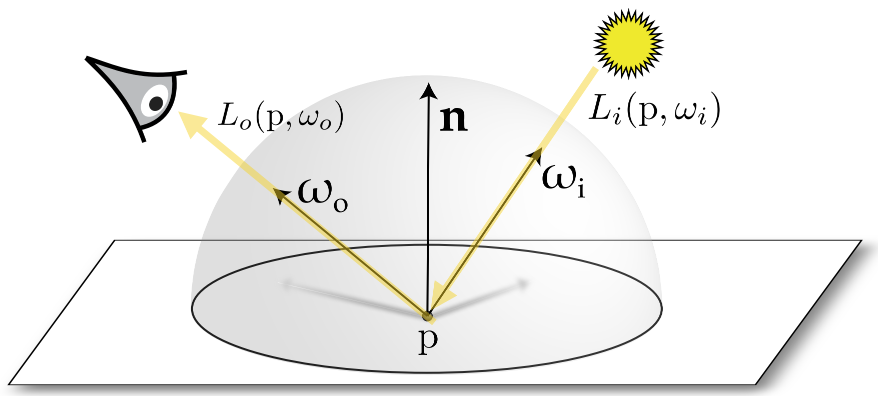
Write pseudocode to compute a Monte Carlo estimate of the integral above using N samples. Your estimate should uniformly sample time between (0...2), uniformly sample pixel area between (x,y)=(0,0) and (x,y)=(.5,.5) and uniformly sample the aperture disk that is centered at (0, 0, D) and has radius R. Your code can call the helpers functions given at the top of the code below.

```
float random1D();  // return a uniformly sampled number between 0..1
vec2 sampleDisk(); // return a 2D pt on a disk with radius 1
                   // (uniform with respect to area)
vec3 normalize(vec3 vector); // return a normalized version of vector
Spectrum f(float x1, float y1, float x2, float y2, float t); // calculate radiance
                            // at sample defined by pixel sample (x1, y1),
                            // disk sample (x2, y2), and time t

// TODO: implement this function...
Spectrum computeEstimate(int N) {

    Spectrum result = 0;
    // accumulate energy over N samples









        return result;
}
```

Describing the Reflection Equation

PRACTICE PROBLEM 1:

In your own words, please describe the terms of the reflection question, provided below. What do the parts A, B, C, D, and E represent?

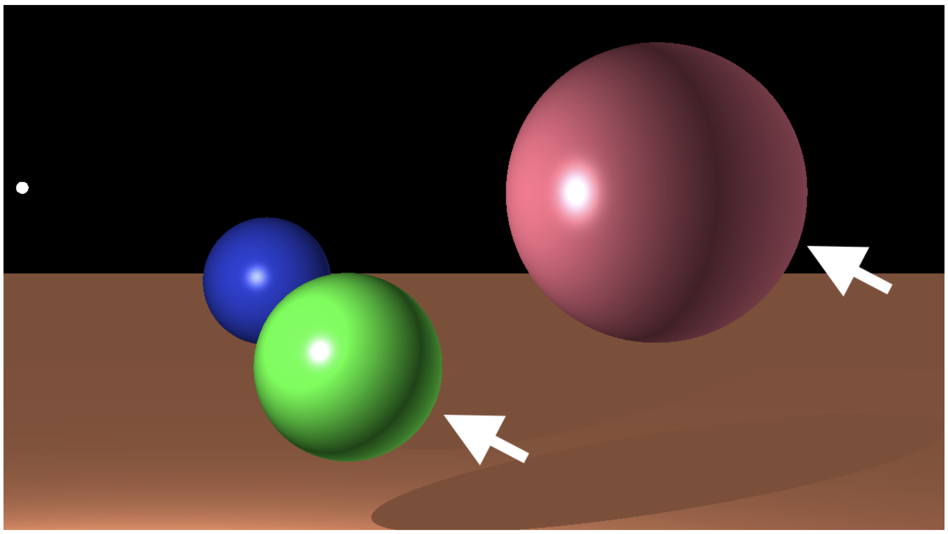


$$\underbrace{L_\mathrm{o}(\mathrm{p}, \omega_\mathrm{o})}_{\textbf{A}} = \underbrace{\int_{\Omega^2}}_{\textbf{B}} \underbrace{f_\mathrm{r}(\mathrm{p}, \omega_\mathrm{i} \rightarrow \omega_\mathrm{o})}_{\textbf{C}} \, \underbrace{L_\mathrm{i}(\mathrm{p}, \omega_\mathrm{i})}_{\textbf{D}} \, \underbrace{\cos\theta_\mathrm{i}}_{\textbf{E}} \, \underbrace{\mathrm{d}\omega_\mathrm{i}}_{\textbf{B}}$$

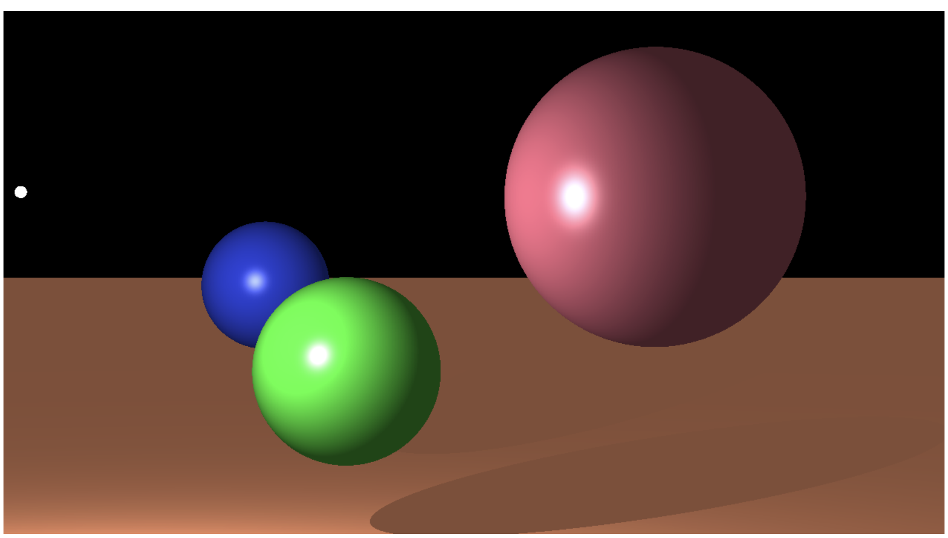
Does this Bug Sound Familiar?

PRACTICE PROBLEM 2:

You're writing a simple raytracer that supports diffuse and specular materials. Your scene contains three spheres that have a mixture of diffuse and specular components, and a single light source off to the left. When you render the scene, you get this image, but there's an odd bug that we've marked with white arrows:



You were expecting to get a render more like this one:



Check out the artifact the arrows are pointing to. Specifically, in the first image, there's a light ring on the side of the spheres not facing the light source. :(Time to debug! Your print statements reveal that your rays are successfully intersecting with the spheres and hitting light sources, and that all the objects in the scene are in their expected locations. You decide that the bug is related to how you calculate the reflected radiance at a hit point.

Question is on the next page...

Given this information, what do you think is the *cause of the bug*? Please pick one of the following, and then provide a justification for your answer.

1. For all the primary rays from the camera, you're shooting a ray into the top-left of each pixel rather than the center of each pixel.
2. Rather than clamping the dot product between the surface normal N at the hit point and the vector to the light source to be at least 0, you're taking the absolute value.
3. For all the ray bounces, you're double-counting the radiance emitted from the light source, which results in an overly-bright image.
4. None of the above. The extra light is simply a result of diffuse interreflection, caused by light reflecting off the other spheres.

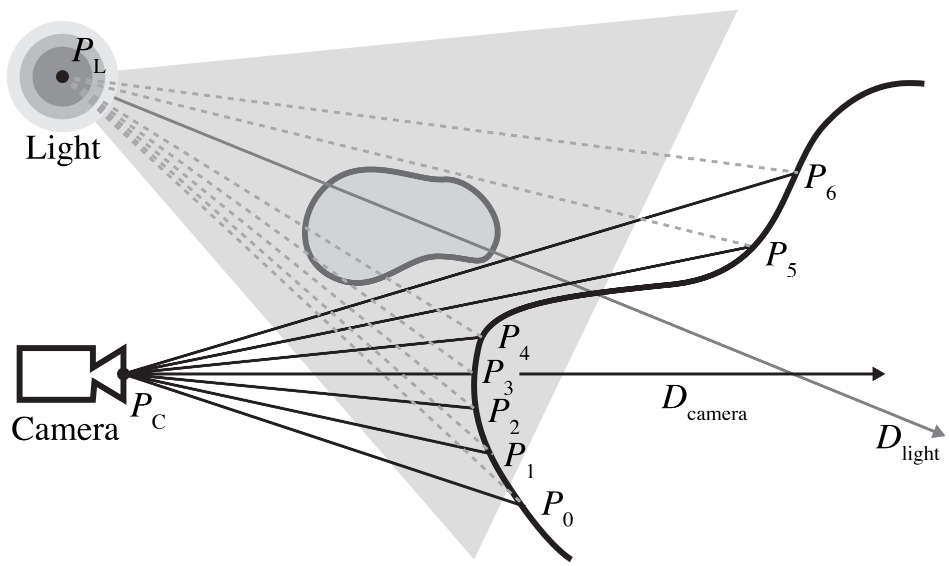
A Highly Irregular Rasterizer

PRACTICE PROBLEM 3:

Imagine that you have a special kind of rasterizer which doesn't evaluate depth/coverage at uniformly spaced screen sample points, instead it evaluates depth/coverage **at a list of arbitrary 2D screen sample points provided by the application**. An example of using this rasterizer is given below. In this problem you should assume that depths returned by `fancyRasterize` are in WORLD SPACE UNITS.

```
vector<Point2D>  myPoints;  // list of 2D coverage sample points: in [-1,1]^2
vector<Triangle> geometry;  // list of scene triangles in WORLD SPACE
Transform worldToCam;       // 4x4 world space to camera space transform
Transform worldToLight;     // 4x4 world space to light space transform
Transform perspProj;        // 4x4 perspective projection transform


// this call returns the distance to the closest scene element from the camera
// for all points in myPoints (assume infinity if no coverage)
vector<float> depths = fancyRasterize(geometry, myPoints, worldToCam, perspProj);
```

You are now going to use `FancyRasterize` to render images with shadows. Consider the setup of a camera, scene objects, and a light source as illustrated below.



A. Assume you use a traditional rasterizer to compute the depth of the closest scene element at each screen sample point. In the figure, the closest point visible under each sample when the camera is placed at position P_C and looking in the direction D_{camera} is given by P_i. All points in the figure are given in **world space!**

Assume you are given a *world space to light space* transform `worldToLight`. (Light space is the coordinate space whose origin in world space is P_L and whose -Z axis is in the direction D_{light}.) Describe an algorithm that computes, for each point P_i, if the point is in shadow from a point light source located at P_L. Your algorithm accepts as input an array of world space points P_i, world space points P_C, P_l, and has access to all variables listed in the example code. The algorithm should call `fancyRasterize` only once. (No, you are not allowed to just implement a ray tracer from scratch.)

Hint: Be careful, `fancyRasterize` wants points in 2D (represented in a space defined by the $[-1,1]^2$ "image plane") so your solution will need to describe how it converts points in world space to a list of 2D sample points in this plane. **This involves transformation, perspective projection via `perspProj`, then convert from a homogeneous 3D representation to 2D.**

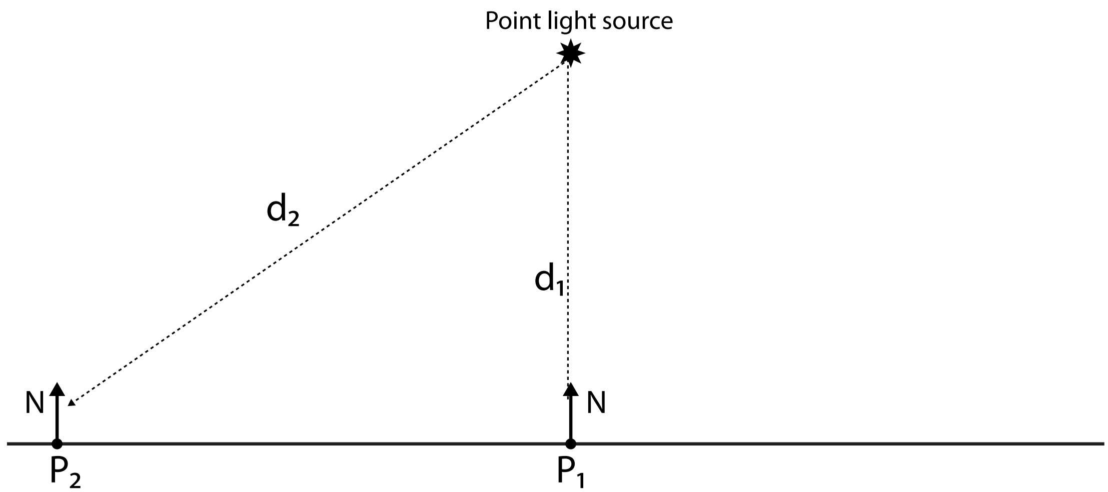
B. Does the algorithm you gave above generate "hard" or "soft" shadows? Why? (You can answer this question even if you did not correctly answer part A—just assume a solution that does what was asked in part A exists.)

C. Prof. Kayvon quickly looks at the algorithm you devised above and waves his hand dismissively. He says, "remember I told you in class that shadow mapping is such a hack", it only yields an approximation to ray traced shadows. The CAs jump in and shout, "Wait a minute here, this algorithm seems to compute the same solution a ray tracer would to me!" Who is correct? Why?

PRACTICE PROBLEM 4:

Consider a point light source that emits uniformly in all directions. Specifically, it emits **equal power per unit solid angle** ($\frac{d\Phi}{d\omega}$=C). The light is shining on the floor as shown in the figure below. Give **TWO REASONS** why the **irradience (E) incident on the floor** at point P_1, which is a distance d_1 from the light, is greater than the irradiance on the floor at point P_2, which is a distance d_2 from the light. Recall that irradience on a surface is power per unit surface area ($\frac{d\Phi}{dA}$.)

Hint: consider the definition of a differential solid angle $d\omega$ in terms of a subtended patch of surface area on a sphere with radius r. What is the power of the light source per unit surface area on the sphere? Then consider the orientation of that surface patch on the sphere compared to the orentiation of the floor.



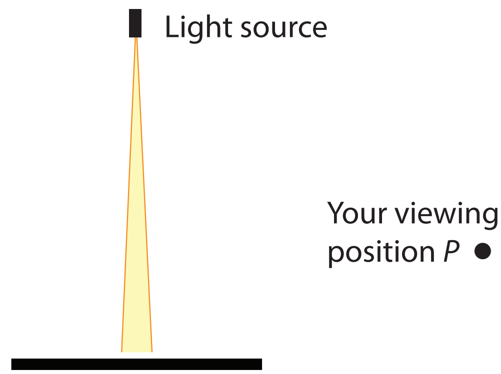
Two Material Thought Experiments

PRACTICE PROBLEM 5:

A. You are standing in a room with a floor that is completely flat, but covered with two different types of materials (denoted as 1 and 2 in the figure below). The room only has one light source, a far away point light that is directly above the floor, and shines light directly down on the floor. (You can think of it as a very, very narrow spotlight, shining straight down.) Material 1 has a **completely diffuse BRDF** (white material) and Material 2 is a **perfect mirror** that reflects all light that hits it. If you are standing on the side of the floor at point P (as shown in the figure) what do you see? In particular be precise about what the mirror tiles look like. (Can you see anything in them, yes or no?) *Remember, there is no other light source in the room other than the spotlight from above..*

2	1	2	1
1	2	1	2
2	1	2	1
1	2	1	2

Top down view of floor



Side view of floor

B.

Monte Carlo Estimator

PRACTICE PROBLEM 6:

Consider the reflection equation for a surface with a constant BRDF: $f(\omega_i, \omega_o) = C$

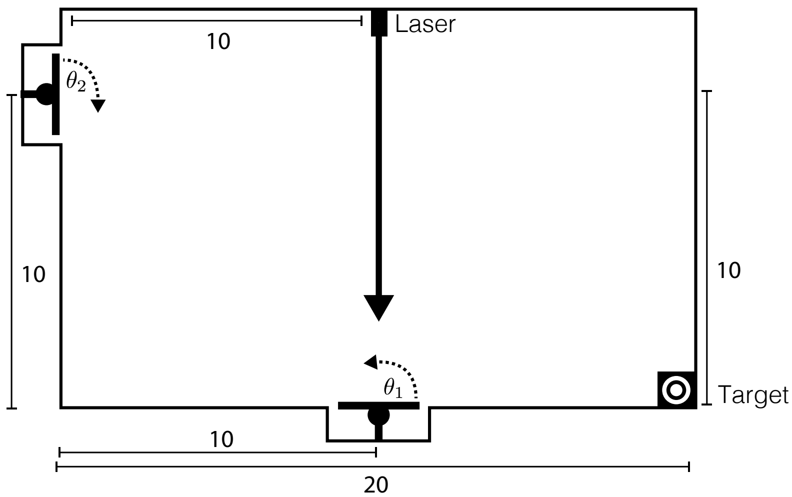
$$L_o(\omega) = \int_\Omega C\, L(\omega_i)\, \cos(\theta_i)\, \mathrm{d}\omega_i$$

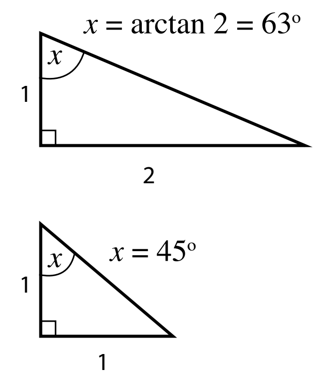
Assuming that we draw samples uniformly from the hemisphere of directions about the hemisphere, please write down an expression for the Monte Carlo estimator F_N for $L_o(w)$. Recall that N is the number of samples used for the estimate, and the expected value of F_N is equal to the integral given above. Also recall that there are 2π steradians in the hemisphere.

Everyone Loves Lasers

PRACTICE PROBLEM 7:

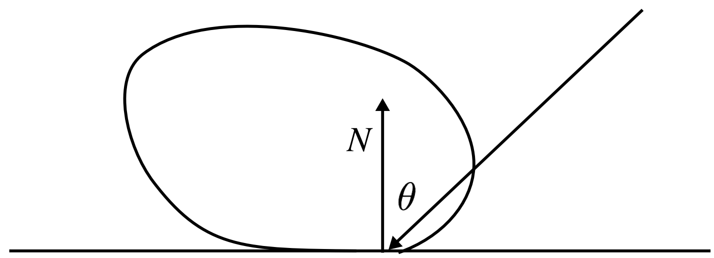
A mad scientist decides to design a fun physics experiment to amuse students in class. The goal of the experiment is to use two **perfectly reflective mirrors** to direct a laser beam, positioned downward from the top of the box, to hit a target in the bottom right corner of the box. The two mirrors can be rotated about their center point by the angles θ_1 and θ_2 as shown in the figure.





A. Please compute positive values of θ_1 and θ_2 to hit the box. Some helpful triangles are given for you, which may or may not be useful. Hint: first determine how to orient the first mirror to direct the beam to hit the second mirror. Then orient the second mirror to hit the target. Please review the slides from lecture about *perfect specular reflection.*

B. One challenge with perfect mirrors is that if you don't get them tilted just right, the laser will miss the target. One of the students, frustrated they couldn't hit the target, takes out a piece of sandpaper and scruffs up the two mirrors. The result is that the mirrors now have a BRDF that is almost fully diffuse, as given by the plot below. Note the surface reflects non-zero incoming light in all directions, but the fraction of light reflected in each of these directions is angle dependent. (More light is still reflected in the direction of perfect specular reflection.)

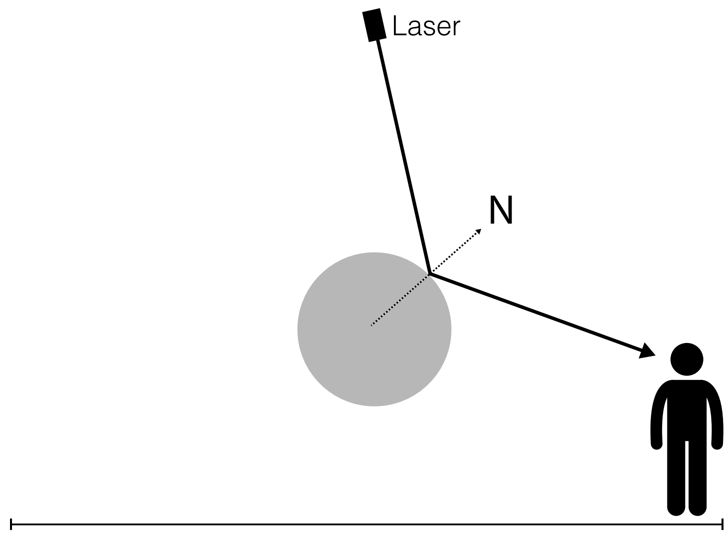


Assuming that (1) the mirrors are set so that $\theta_1 = \theta_2 = 0$ and that all the walls of the room reflect no light (they are perfectly black), does any laser light hit the target? If your answer is no, explain why not. If your answer is yes, please explain why, and also state whether target is brighter or darker compared to what it would look like in the case of well-aligned perfect mirrors from part A.

Reasoning about BRDFs

PRACTICE PROBLEM 8:

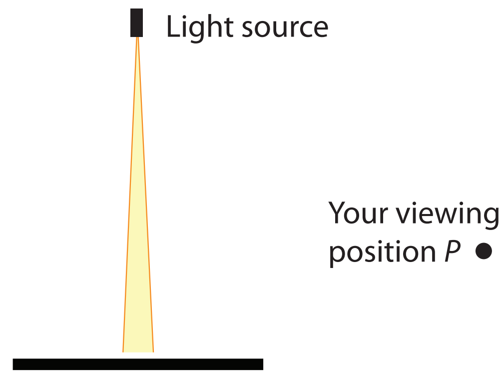
A. Consider the following setup where a red laser (a point light source emitting light in a single direction) is shining on sphere that is a **perfectly reflective mirror surface**. The light reflects off the sphere directly into the eyes of a person standing in the room. Assuming that there no other light sources in the room, and all other surfaces of the room are black, what does the person see if they lower the position of their head by sitting on the floor? Why? (Hint: a good answer will refer to the amount of light arriving at the viewer's eyes.)



B. You are standing in a room with a floor that is completely flat, but covered with two different types of materials (denoted as 1 and 2 in the figure below). The room only has one light source, a far away point light that is directly above the floor, and shines light directly down on the floor. (You can think of it as a very, very narrow spotlight, shining straight down.) Material 1 has a **completely diffuse BRDF** (white material) and Material 2 is a **perfect mirror** that reflects all light that hits it. If you are standing on the side of the floor at point P (as shown in the figure) what do you see? In particular be precise about what the mirror tiles look like. (Can you see anything in them, yes or no?) *Remember, there is no other light source in the room other than the spotlight from above..*

2	1	2	1
1	2	1	2
2	1	2	1
1	2	1	2

Top down view of floor



Side view of floor