Assignment 3: PathTracer 2

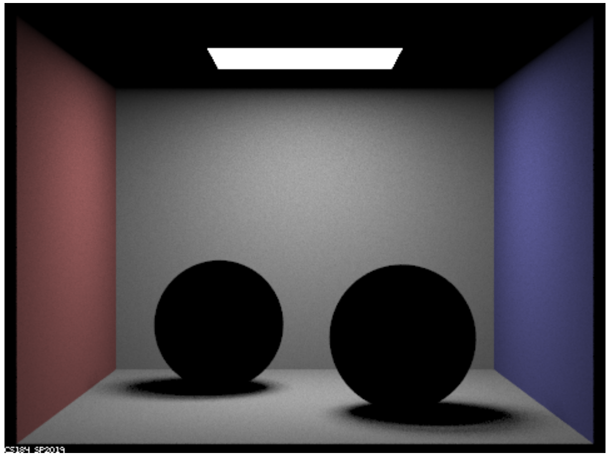
Vincent Tantra

For this project, I build on our previously implemented pathtracer project to include some more complicated functionality, including a wider range of materials, environment lighting, and depth of field effects. Let's get started!

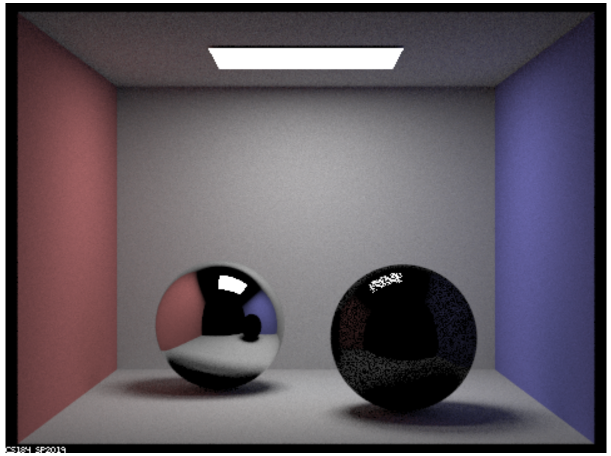
Part 1: Mirror and Glass

In this part, we implement basic reflection and refraction principles to be able to render mirror and glass materials. Perfect reflection is simple enough; the angle is mirrored perfectly, so we negate the x and y values of our local Cartesian ray direction, while keeping the z component the same. Since this delta BSDF is a special case, we set the PDF to 1.
A small note we must make: the maximum ray depth in this case must be greater than 1, because for a mirror to work, the rays of light must bounce of the mirror material! Any ray depth that is equal to 1 or less would not make sense.
When implementing refraction, we use Snell's law to determine the direction of the ray after it goes through a medium change. I make sure to account for cases where the ray both enters and exits the glass medium, as well as the case of total internal reflection, in which case the ray doesn't refract.
However, points on glass that refract also reflect at the same time. The Fresnel equations model this in full accuracy, but because they are difficult to program, I use Schlick's approximation to determine the ratio, using the approximation to give us a probability of any ray either reflecting or refracting.
Below are the results of these formulas! Note the changes due to the differences in max ray depth. The glass ball becomes more complete as rays are allowed to bounce more, and the reflection of the glass ball in the mirror ball also becomes more clear, since rays that bounce off the glass ball can bounce off the mirror ball as well.

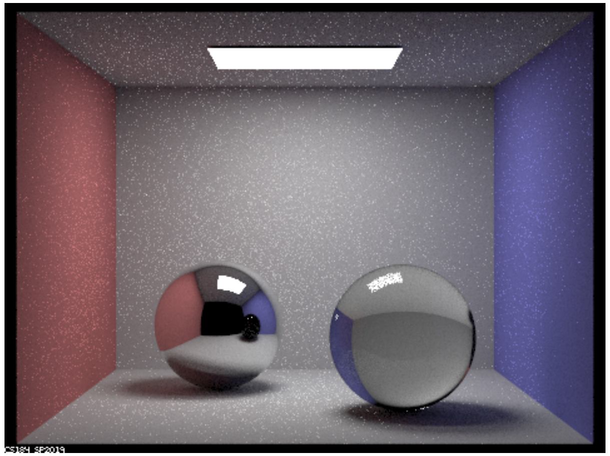
Max ray depth: 0.



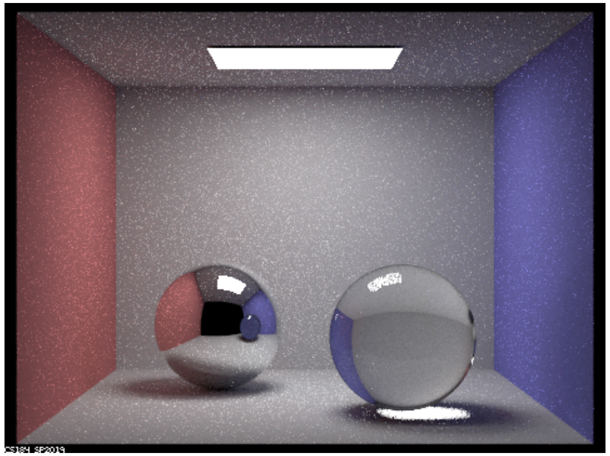
Max ray depth: 1.



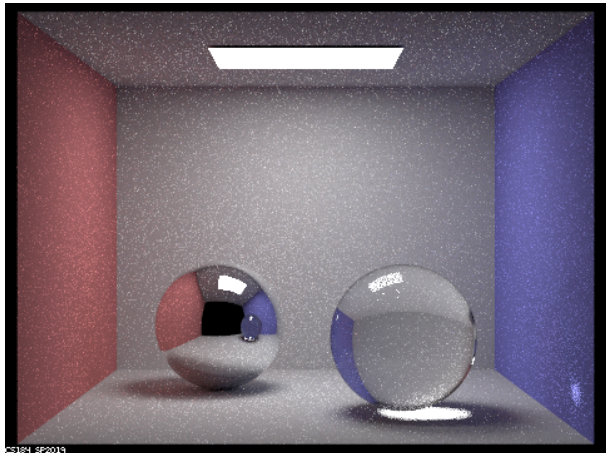
Max ray depth: 2.



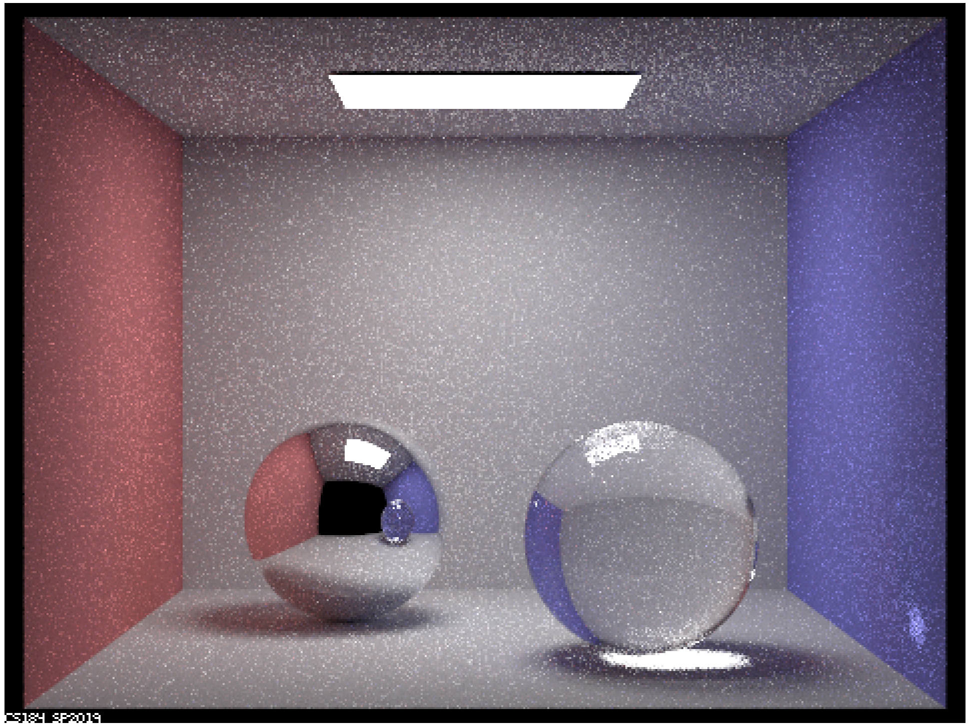
Max ray depth: 3.



Max ray depth: 4.



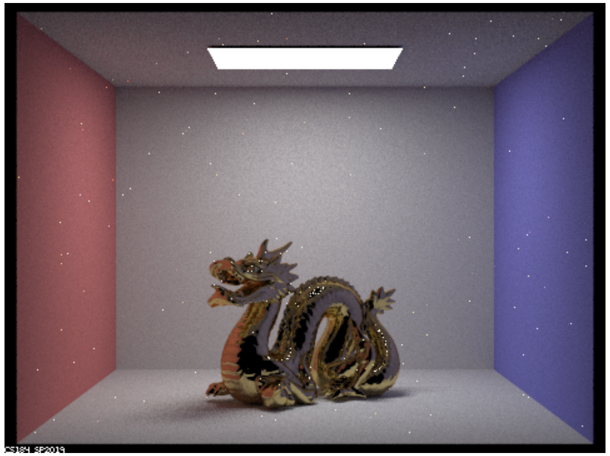
Max ray depth: 5.



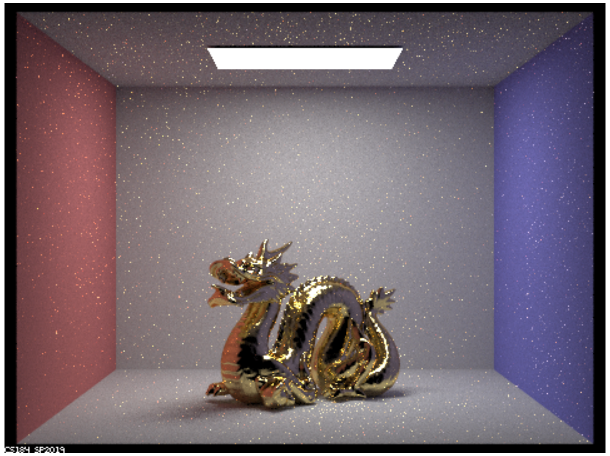
Max ray depth: 100.

Part 2: Microfacet Materials

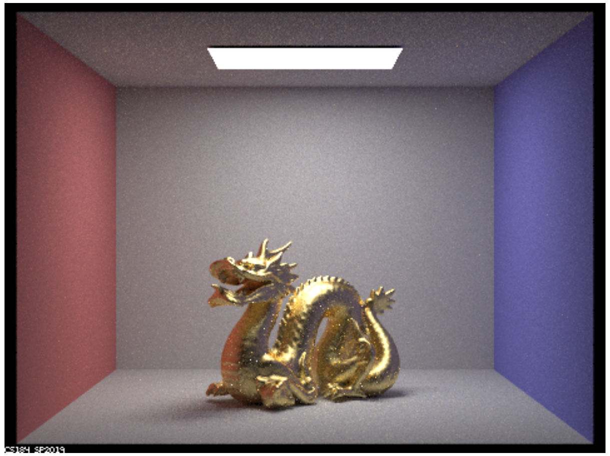
Microfacet materials are isotropic rough conductors that reflect light. These include many common metals such as gold, copper, and silver. We use specular shading and brighten spots whose reflected rays are closer to our viewpoint; specifically, microfacets are perfectly specular, so we only reflect rays whose normals lie exactly along the half vector. We distribute the microfacet's normals according to the Beckmann distribution. We also implemenet a wavelength-dependent Fresnel term calculated for the R, G, B channels (simplified for efficiency). Finally, we implement importance sampling along the Beckmann NDF for sampling. Below are rendered dragons in gold material, with different alpha values. Note the difference in the glossiness of the macro surface. A smaller roughness value (alpha) will cause the material to be smoother, while larger values will make the material more diffuse and "rough."



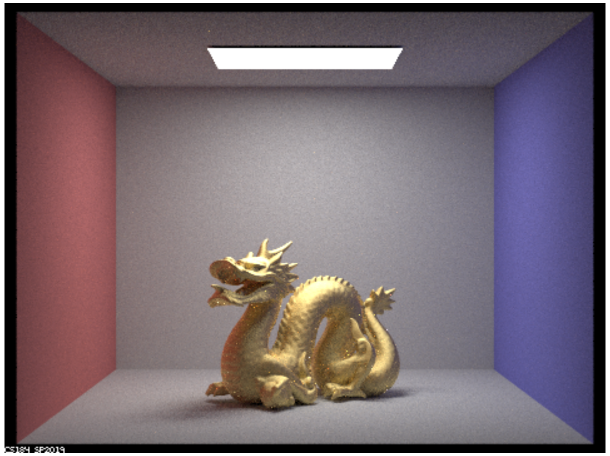
Alpha value: 0.005.



Alpha value: 0.05.

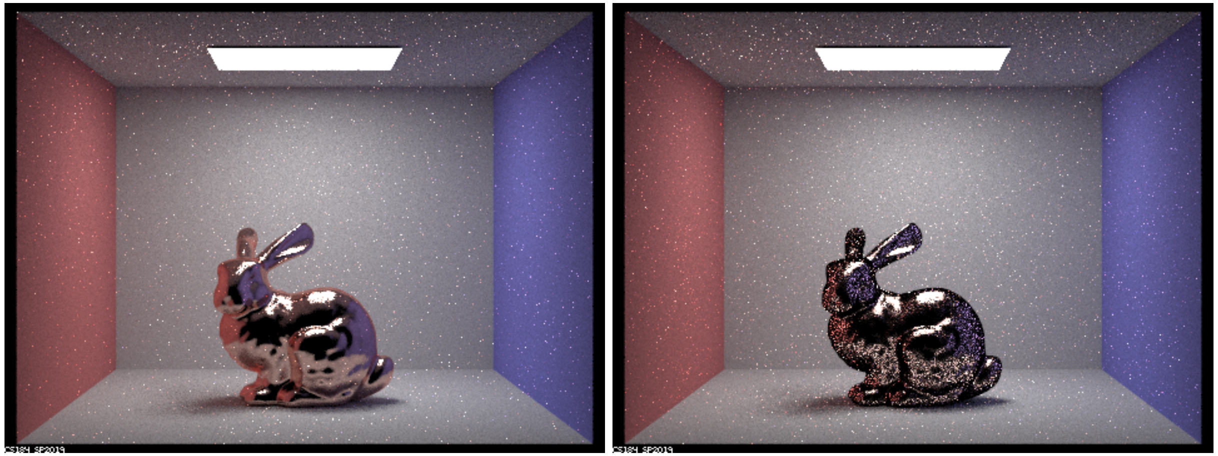


Alpha value: 0.25.



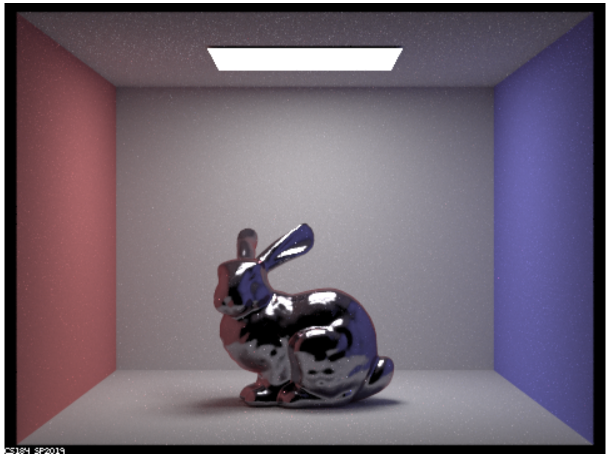
Alpha value: 0.5.

We can also compare the differences between cosine hemisphere sampling and importance sampling using the bunny image. Note that cosine hemisphere sampling is much more noisy; this is a result of not weighting samples based on the shape of the distribution.



Using importance sampling. Using cosine hemisphere sampling.

By changing the eta values, we can also render any material! As a tribute to the Valley, I chose to render a silicon material bunny (Si) using values found from a website that provides a refractive index. The eta values for the 3 fixed color channels are 3.9173, 4.0885, and 4.5251, and the k values are 0.023433, 0.041176, and 0.11358, for the RGB channels, respectively.

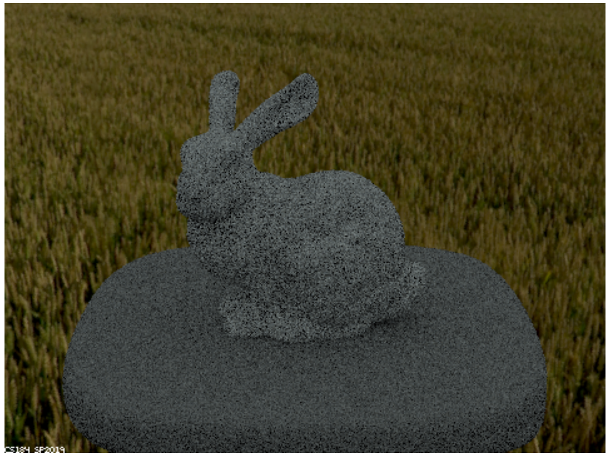


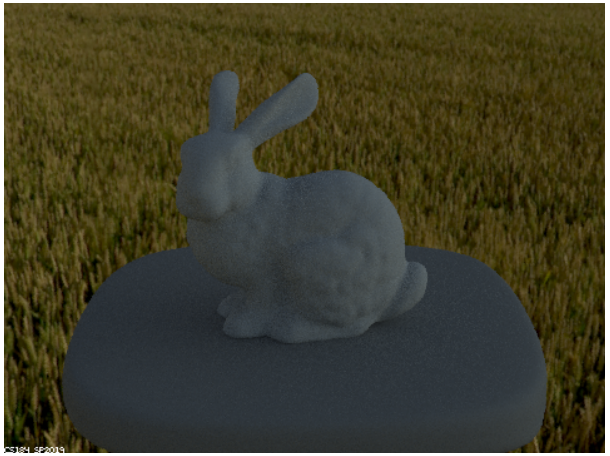
Silicon Rabbey!

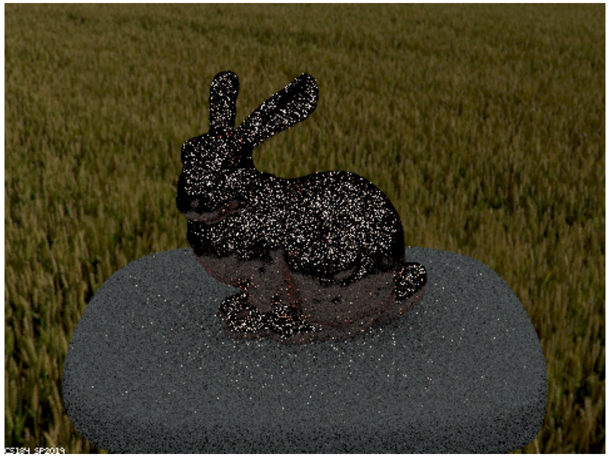
Part 3: Environment Light

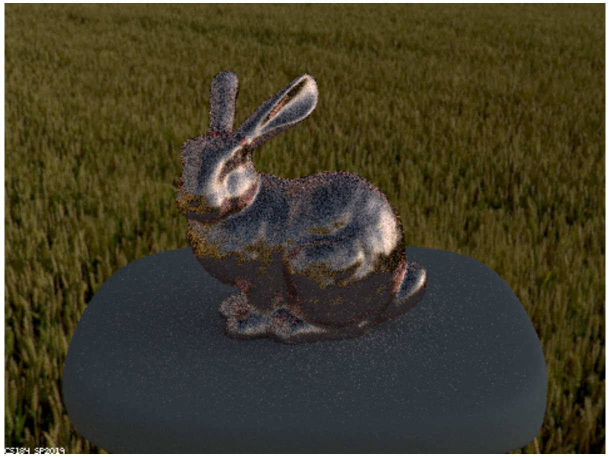
There are many cases in the real world where another kind of light source is present. These cases involve "environment light," an infinite light source that supplies incident radiance from everywhere. As such, we would like to implement this to be able to render more realistic lighting scenarios.

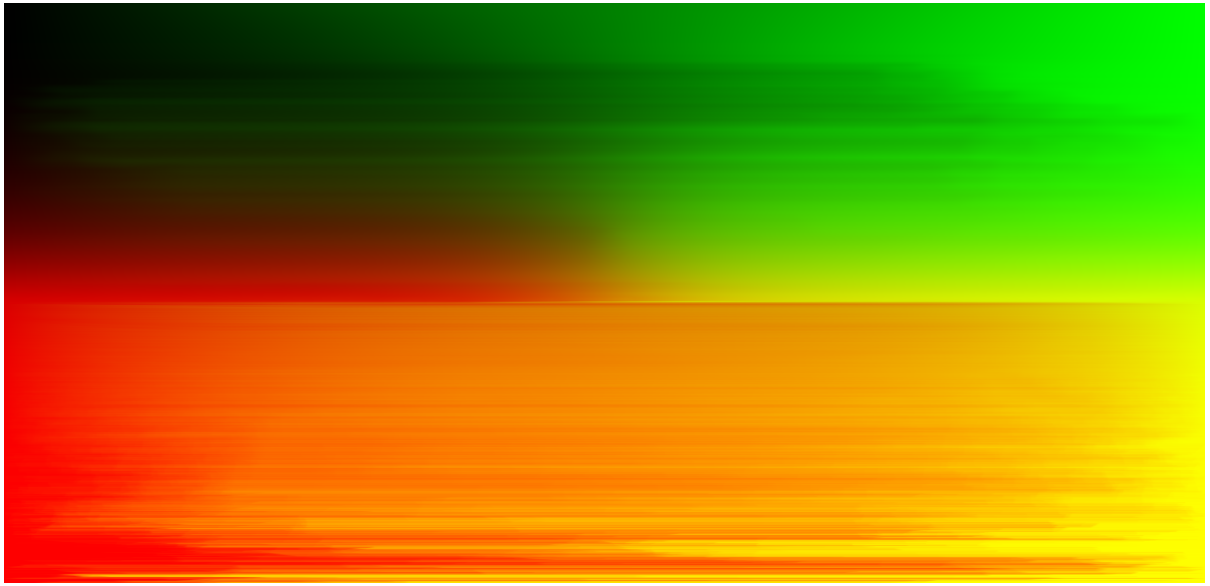
Light is represented as a texture map with theta and phi as parameters.
We take environment lights from an environment map (provided as a .exr file), and either sample rays uniformly or with importance sampling. Uniform sampling is similar in practice, where we sample in a sphere and find the appropriate radiance in the texture map. Importance sampling is a bit trickier; it requires us to compute the marginal and conditional distribution functions across our parameters, and then to sample using the inverse method on these distributions. What we get is a less noisy rendered image (much less noisy in fact, since uniform sampling takes many samples to converge to the true image).


Unlit bunny with uniform sampling.


Unlit bunny with importance sampling.


Unlit microfacet bunny with US.


Unlit microfacet bunny with IS

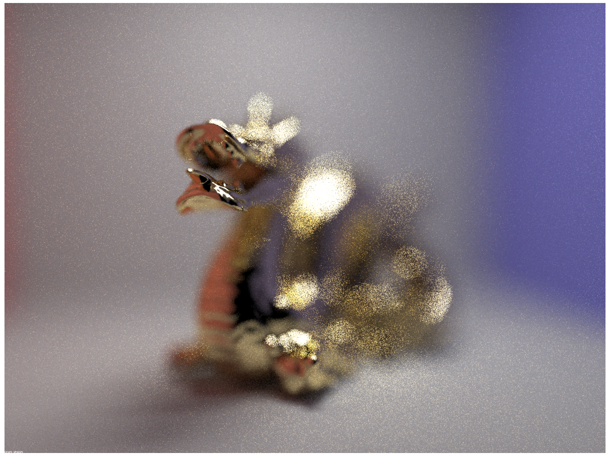


My debug image for the probability distribution of my environment.

My environment used throughout this part. The field!

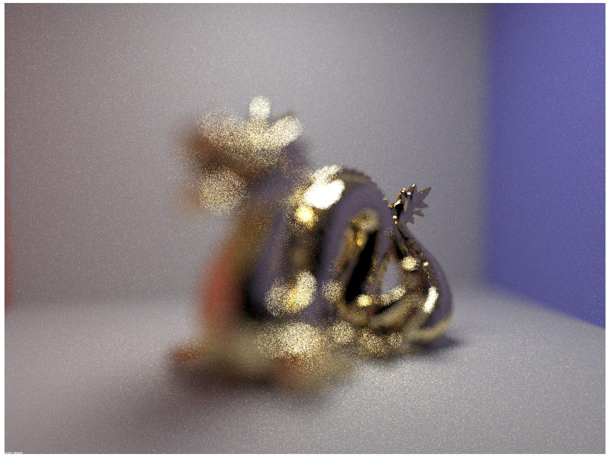
Part 4: Depth of Field

For the final part of this project, we implement depth of field, aperture, and focusing. Previously, we've been looking at our images through a "pinhole camera," where everything is in focus. However, a more accurate representation of vision is through the lens model, where cameras have finite apertures and finite planes of focus that lie at a certain focal distance away from the lense. Objects that lie in this plane are sharp, while others are fuzzy. To implement this realistic camera model, we alter the way we trace rays by sampling from a uniform thin lens, ignoring thickness, and using that to determine the focal points in an image. Below are 4 dragons rendered at different focal distances, with 16 samples per pixel, 4 light samples, and a max ray depth of 4. Note that, as focal distance increases, the part of the image that falls in focus lies farther from our camera.

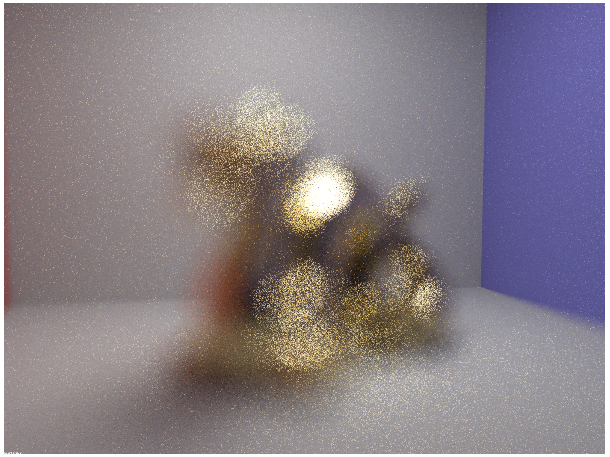


Focal distance: 1.4 (Head).

Focal distance: 1.9 (Body).

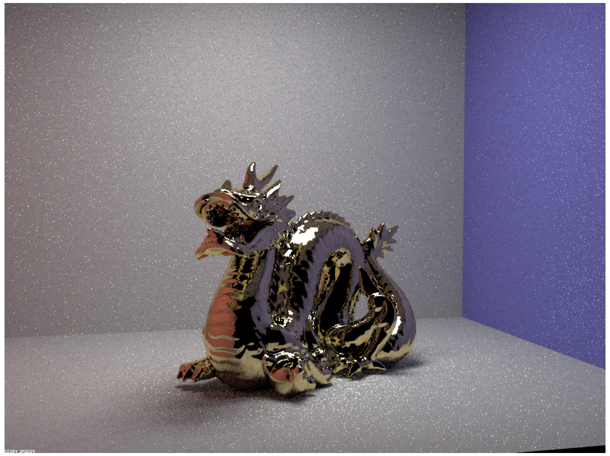


Focal distance: 2.1 (Tail).

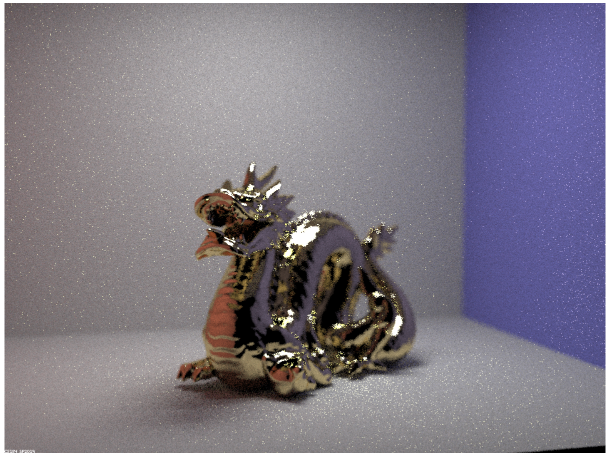


Focal distance: 3.0 (Corner).

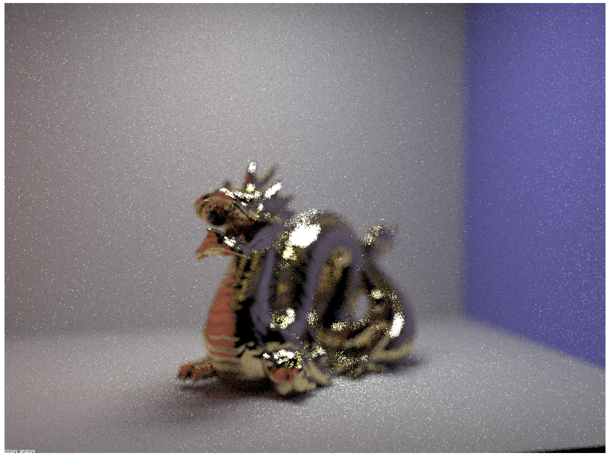
The following 4 photos also show the effect of different apertures on the same image, with the same focal distance. As we increase the aperture, the effective field of focus becomes smaller, and more of the image becomes blurry. Note how as I increase the aperture, the corner falls out of focus, then the dragon's tail, then the dragon itself. These are rendered with the 8 samples per pixel, 4 rays and a max ray depth of 3.



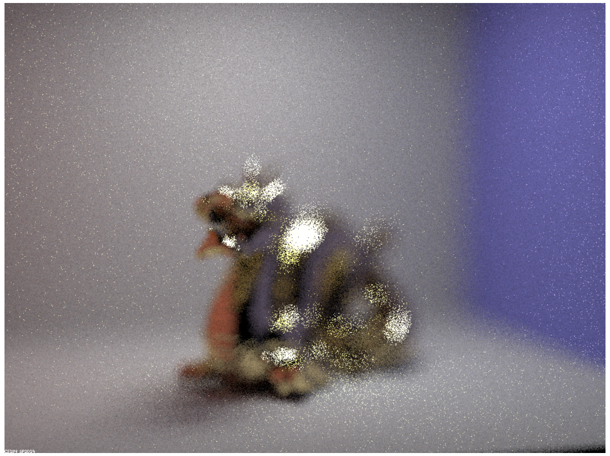
Aperture: 0.0 (pinhole camera).



Aperture: 0.03.



Aperture: 0.06.



Aperture: 0.18.