Assignment 3: PathTracer

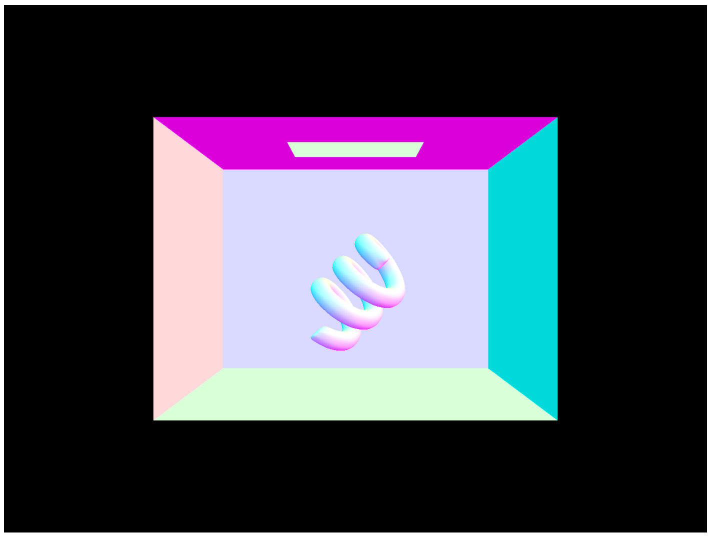
Ruixin Huang cs184-aed

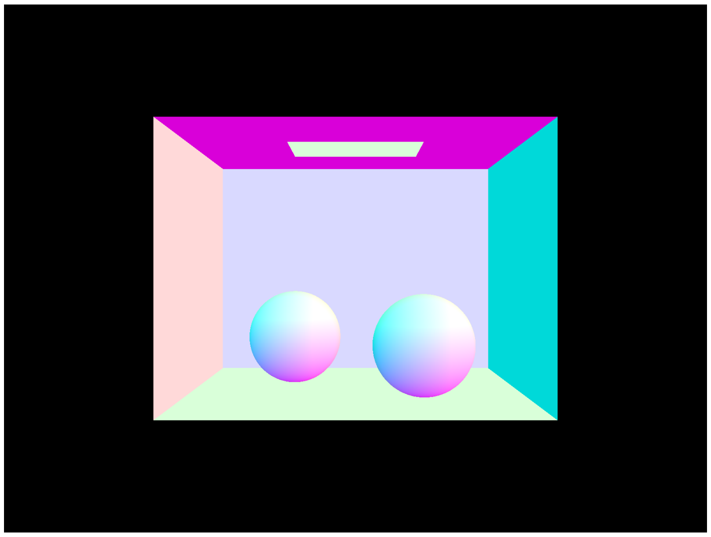
This project explores the how ray is generated and how do they interact with objects to create the rendered scene we often see. Throught this project, I explored the different way of illumination for a scene and how to make it look better and work faster.

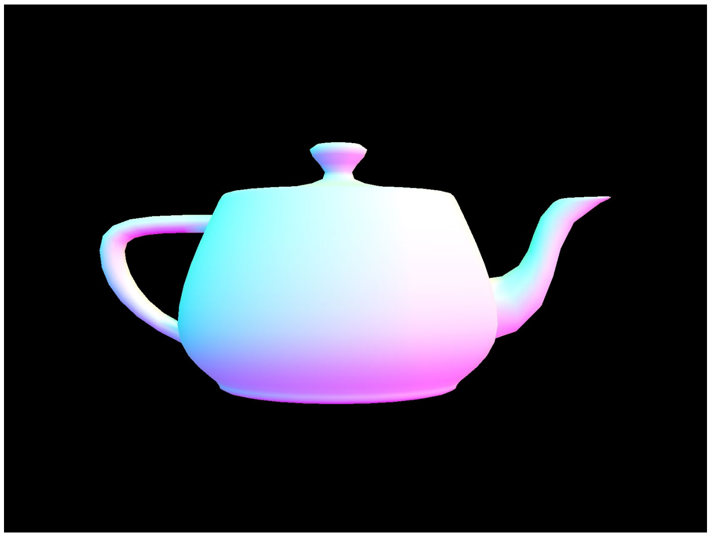
Part 1: Ray Generation and Intersection

ray generation and primitive intersection parts of the rendering pipeline: we define a sensor plane which maps (0,0) to the bottom left and (1,1) to the top right. And we use lerp as we did in the previous project to find the corresponding input points in the camera space. Then we apply c2w matrix to transform it into world space. For each pixel, we either take multiple smaples and average them or we just cast a single ray through the center of the pixel.

Explain the triangle intersection algorithm you implemented in your own words : the intersection must satisify two equations : the one of the ray and the one of the triangle. the triangle is represented with barycentric coordinates. We solved for the intersection and check if it's vlaid.


coil


spheres


teapot

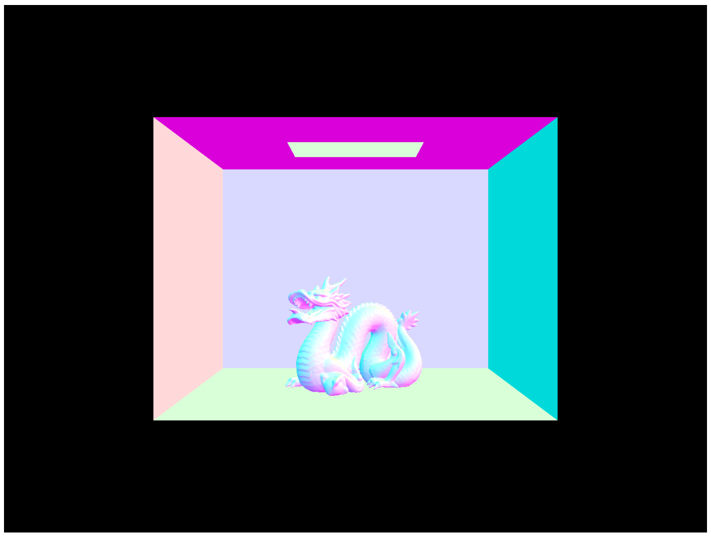
Part 2: Bounding Volume Hierarchy

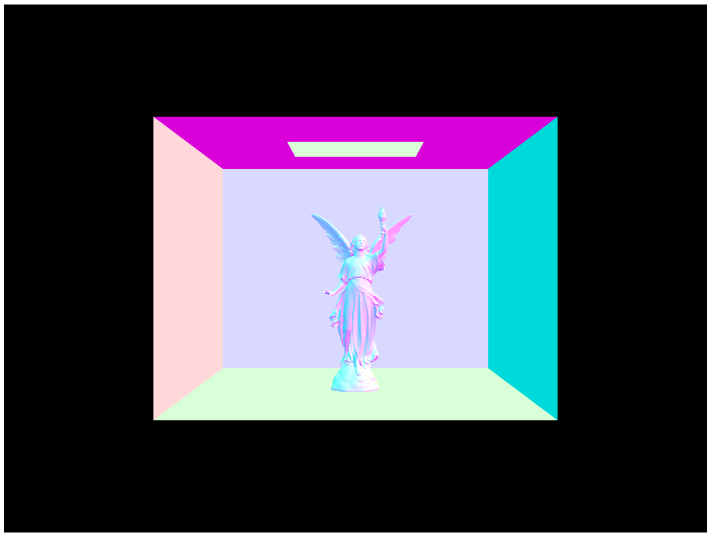
BVH construction algorithm:

First, I pick the dimension to split on by comparing x, y, z axis value and pick the largest one. I then split all the prims into two vectors, left and right, based on their centroid and current bounding box's centroid. I then build the two bounding box of left and right by using the same method recursively.

BVH intersection algorithm:

I test if a ray intersects the prims by using the bounding box. I checked to see if there's intersections between the three intervals (xmin,xmax),(ymin,ymax),(zmin,zmax). If the ray does hit a bounding box, I check if it's a leaf node. If it's a leaf node, I check all the prims the node points to, otherwise, I check all its children.


dragon.


angle.

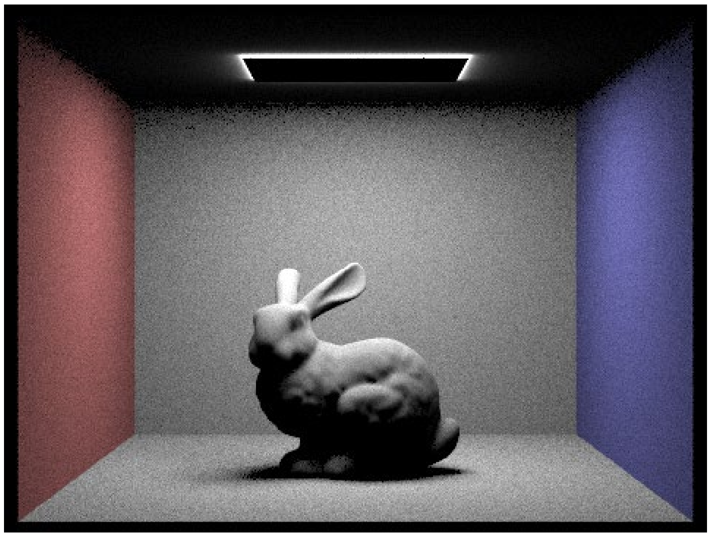
maxplanck.

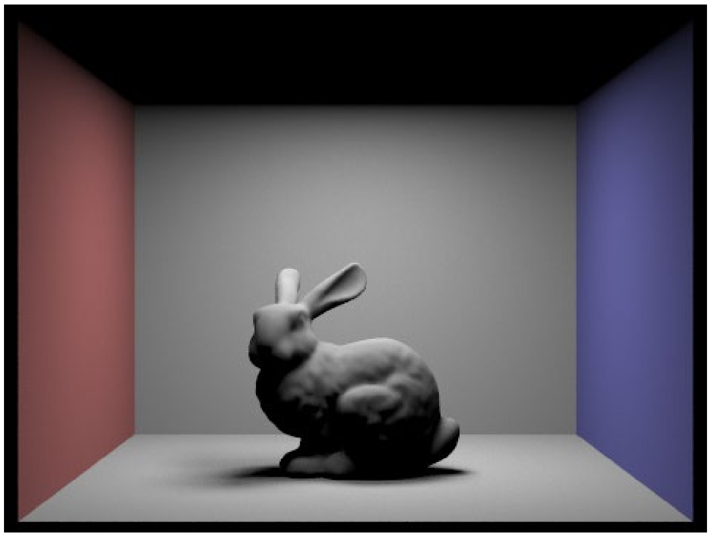
In general, I see a significant speed increase. And the increase in speed is more noticable in large files.

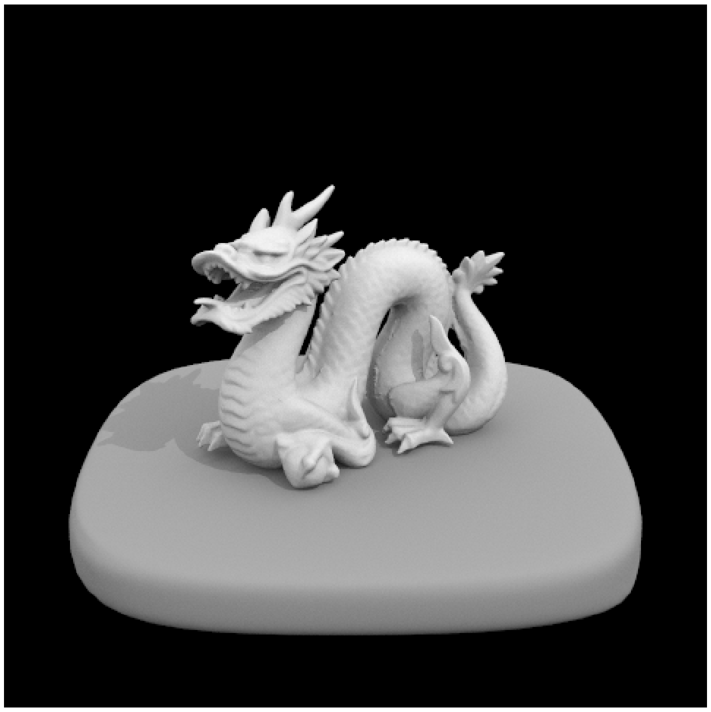
Part 3: Direct Illumination

Uniform hemisphere sampling: sample the hemisphere originated at the hit point uniformluy at random and form a ray. If there's an intersction between the ray and objects in the scene, caculate emission from the intersection point. Scale it to the right value and add it to the output.

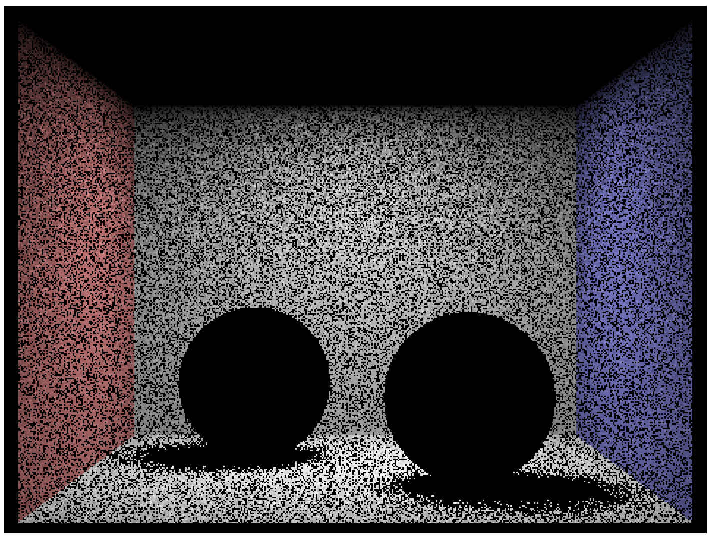
lightening sampling : Create a shadow ray originated at the hit point with a small offset, check to see if this ray intersects with any other objects in the scene before it reaches the sampled location. And check if it's a delta light to determine the number of samples. If it doesn't hit any object, we calculate the value and divide by the probablity to make it unbiased, then we the value to the output. At last, we just take the avergae of the output given the number of samples.

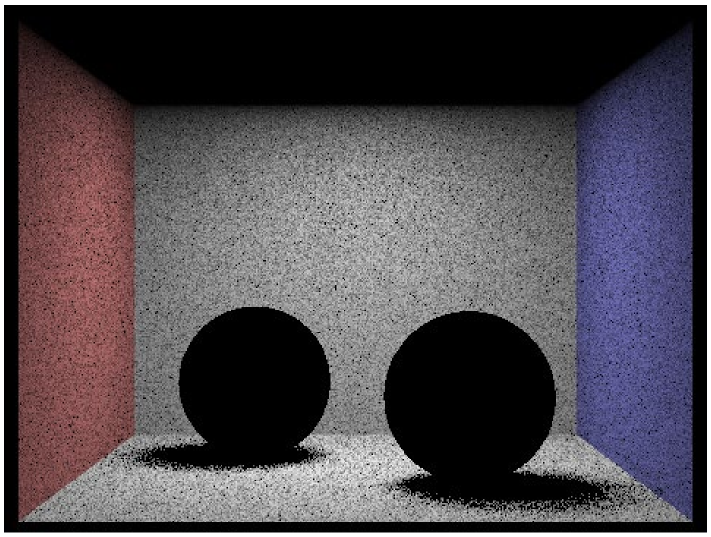

Bunny.dae, uniform hemisphere sampling.

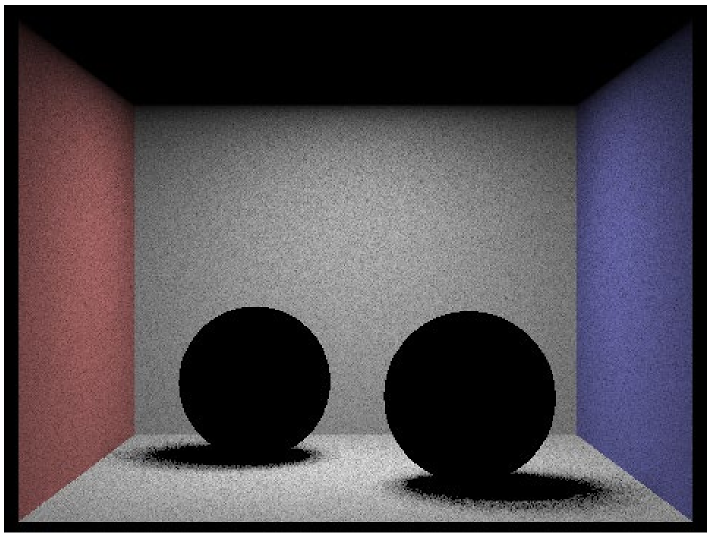

Bunny.dae, light sampling.

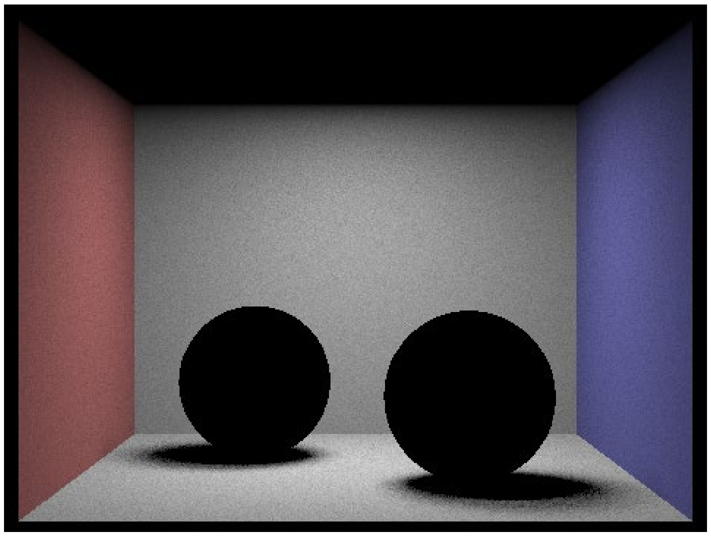

Dragon.dae,light sampling.

spheres.dae : compare the noise levels in soft shadows when rendering with 1, 4, 16, and 64 light rays (the -l flag) and 1 sample per pixel (the -s flag) using light sampling, not uniform hemisphere sampling.

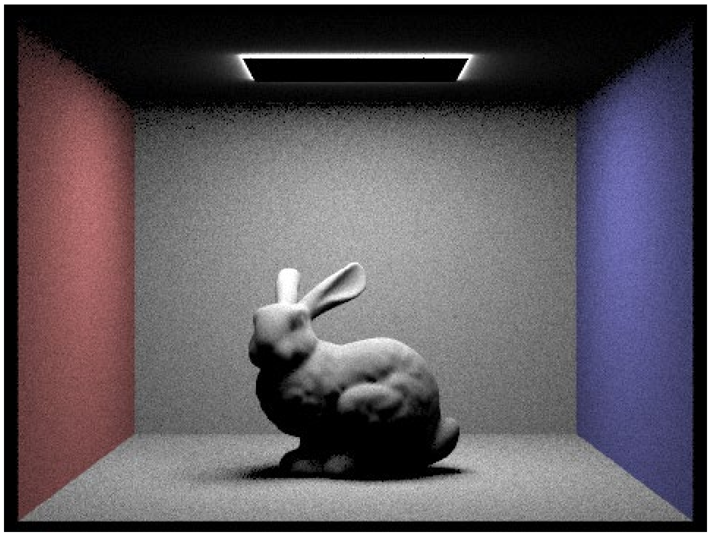

s = 1, l = 1.

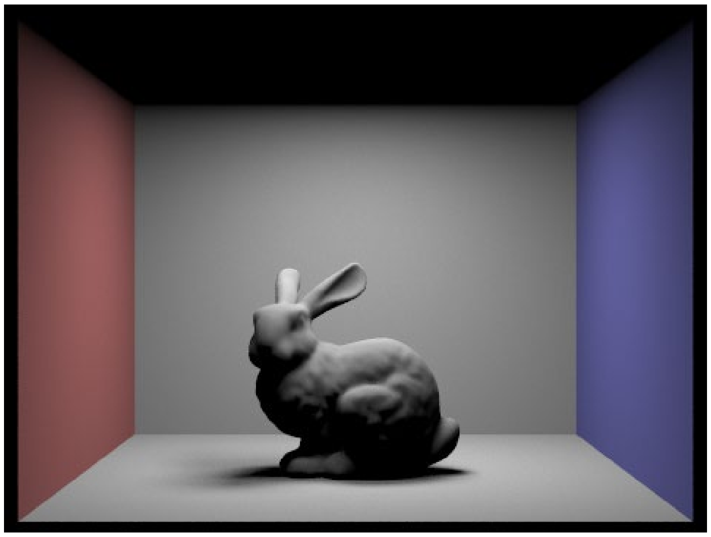

s = 1, l = 4.


s = 1, l = 16.


s = 1, l = 64.

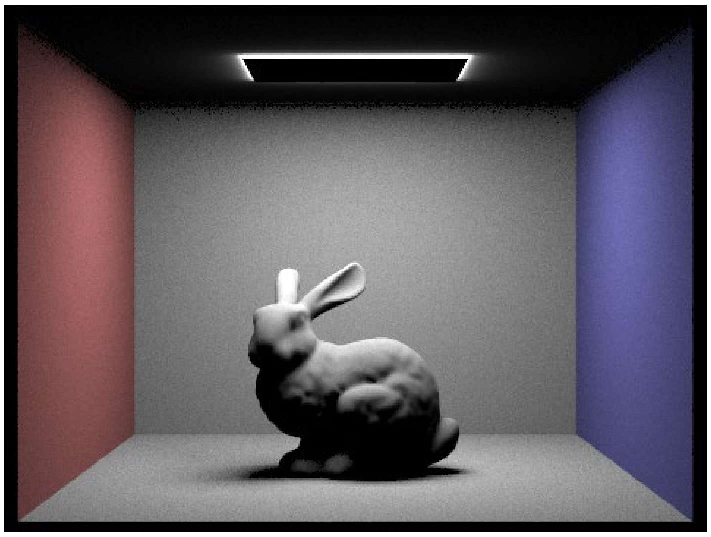
it's harder for uniform hemisphere sampling to converge, it requires a higher number of sample per pixel and ray per pixel. In general, we get better results using lightening sampling. It has less noise than uniform sampling.

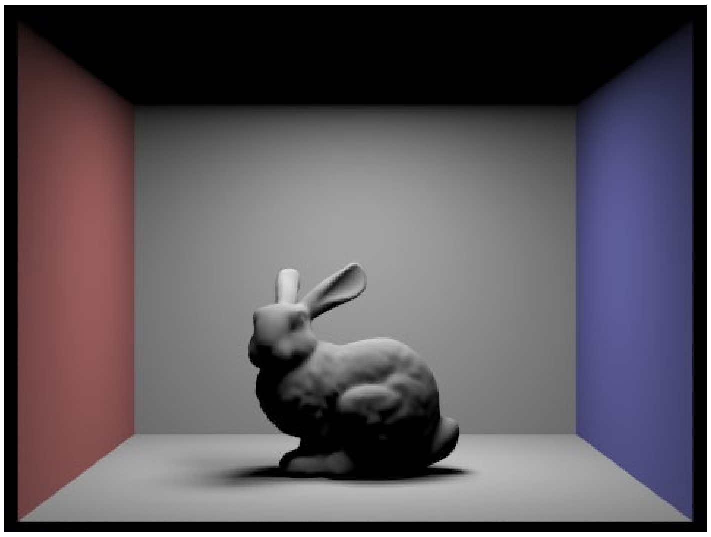




Uniform hemisphere sampling, s = 64, l = 32.

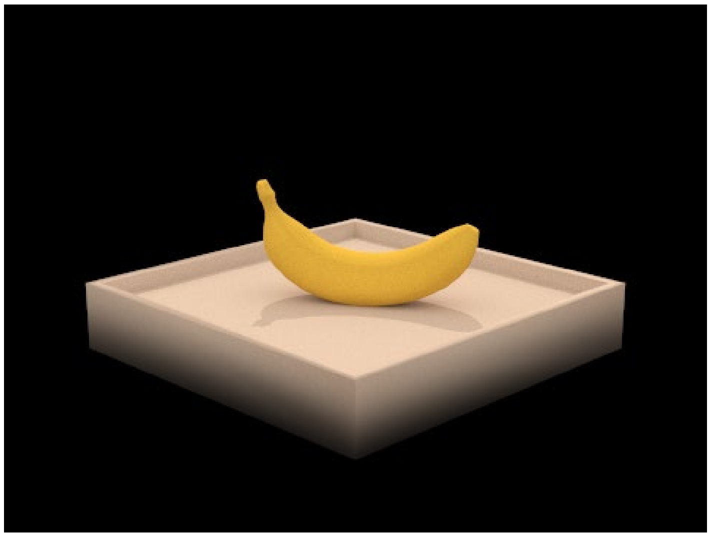
Lighting sampling, s = 64, l = 32.

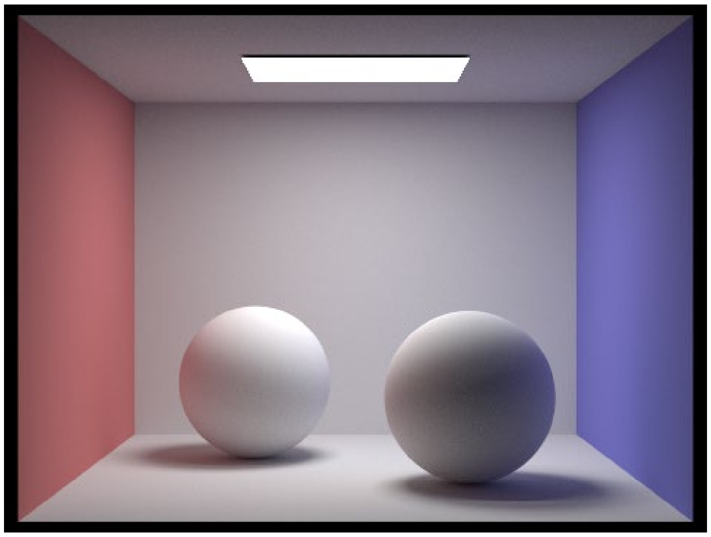

Uniform hemisphere sampling, s = 128, l = 64.

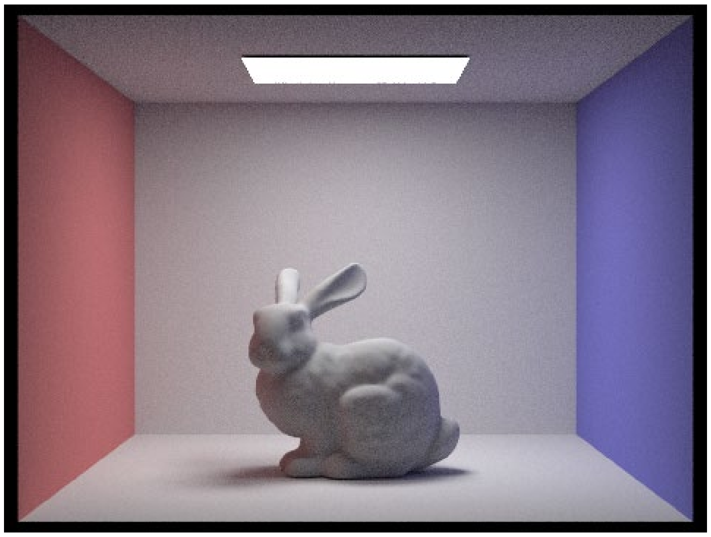

Lighting sampling, s = 128, l = 64.

Part 4: Global Illumination

The indirect lightening part calls the one_bounce_randiance function first, then based on the hit point,it samples a random direction and formes a new Ray in the direction and recursively trace that Ray. The new ray will have a new hit point, and a depth of the original ray plus one. We keep track of the depth of the ray so it won't loop forever. And we also use Russian Roulette to determine the probably of termination when the depth is the maximum of what we want. If we haven't reached the maximum depth yet, we recursively call at_least_one_bounce with probablity one. If we have reached the maximum depth, we use Russion Roulette to determine if we keep going.

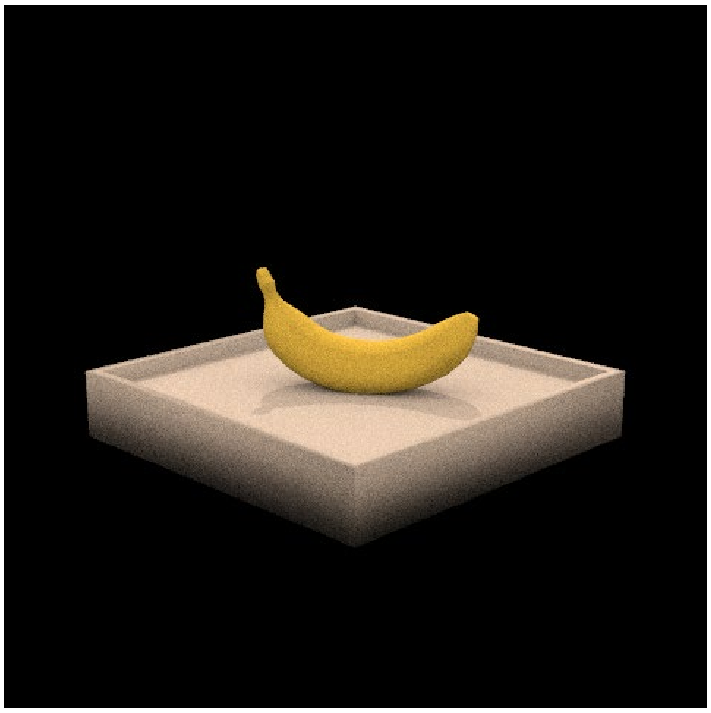

Banana.

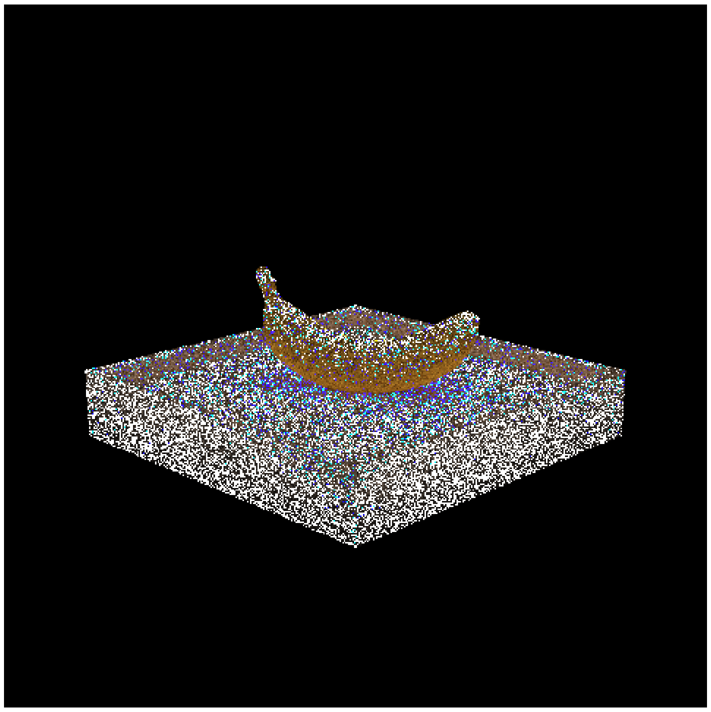

Spheres.

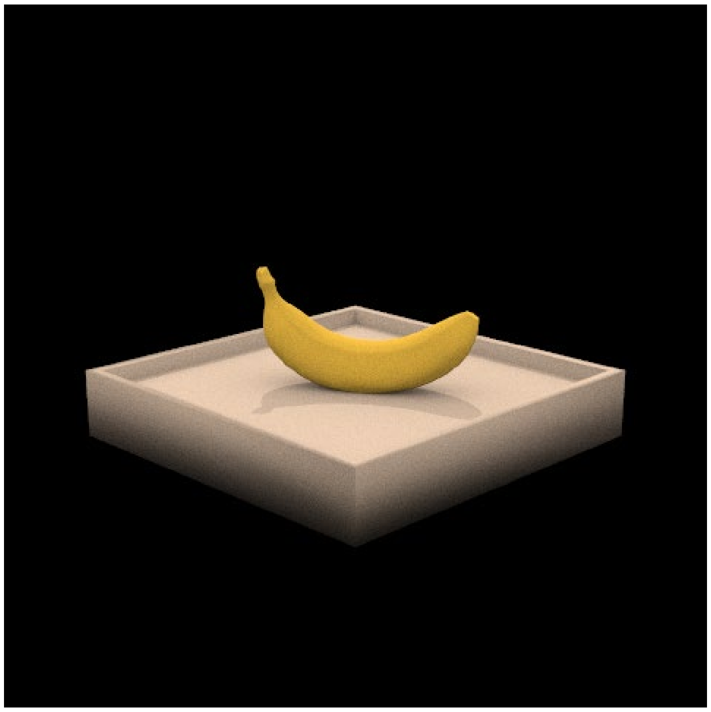

Bunny.

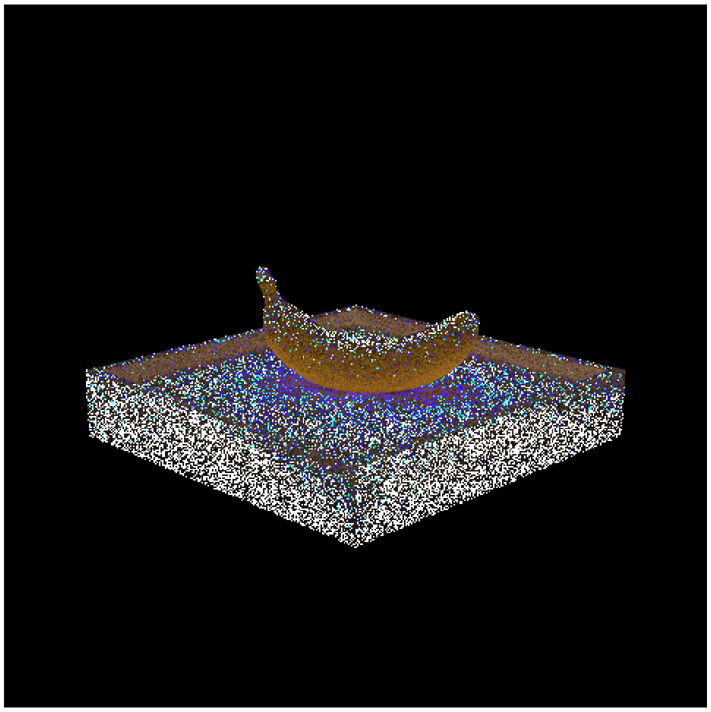
Building.

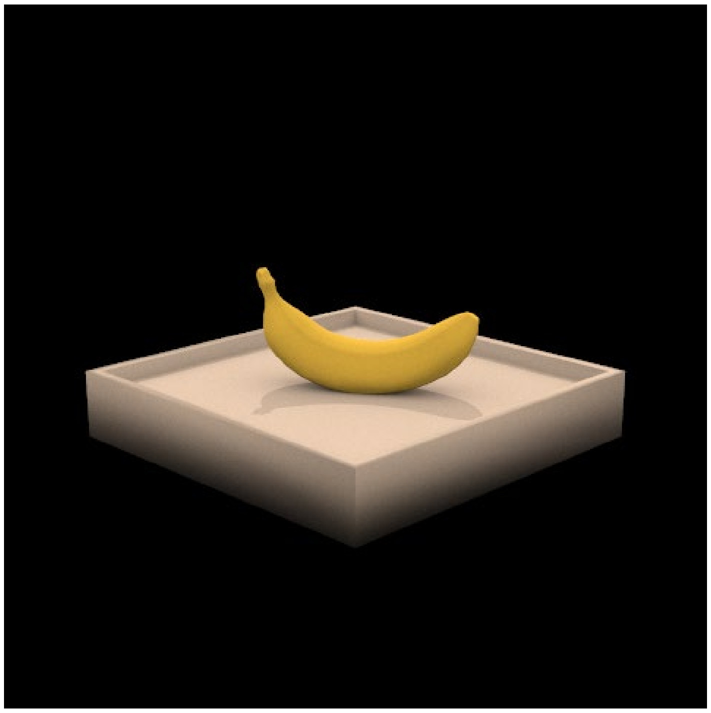
banana.dae with direct and indirect illumination.

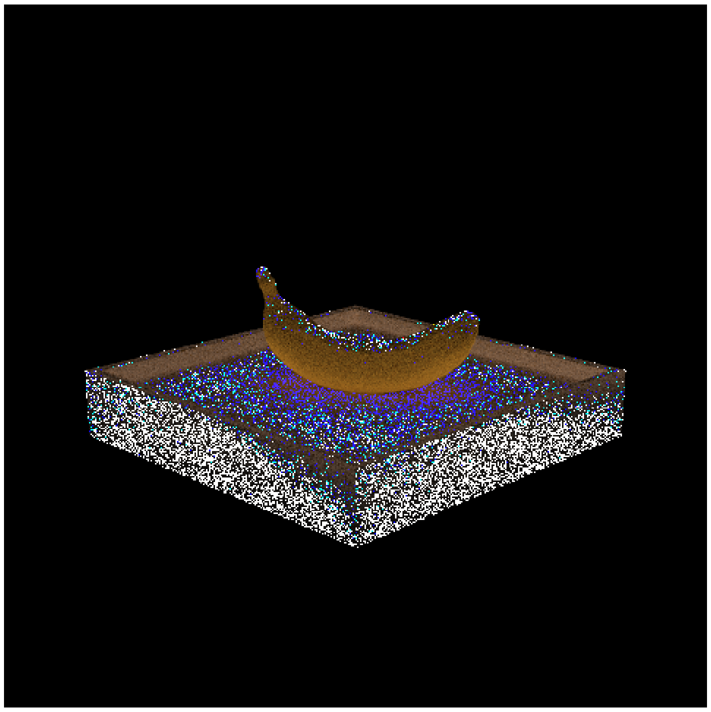

Direct illumination. s = 16, l = 8.


Indirect illumination. s = 16, l = 8.

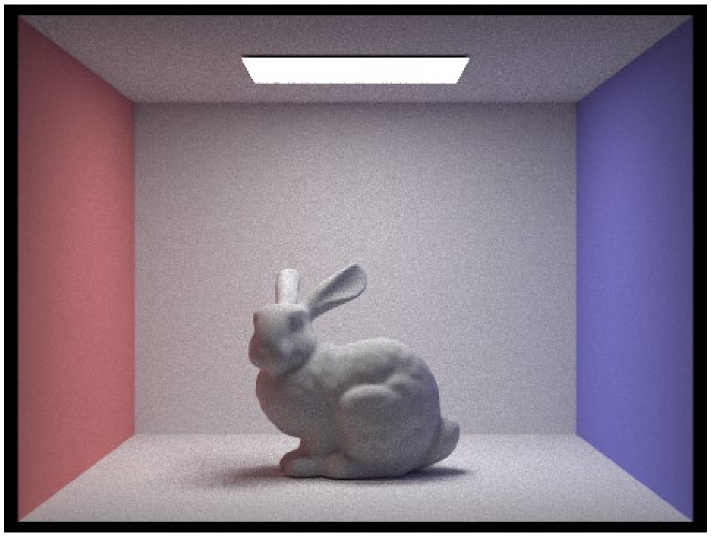

Direct illumination. s = 32, l = 16.

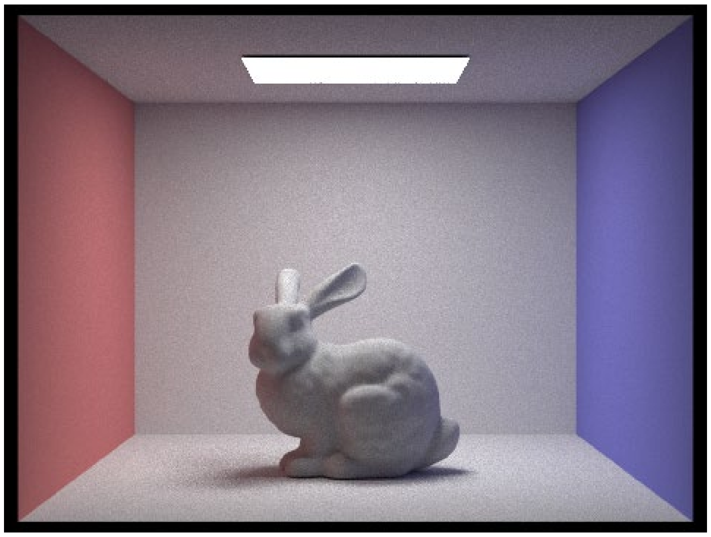

Indirect illumination. s = 32, l = 16.

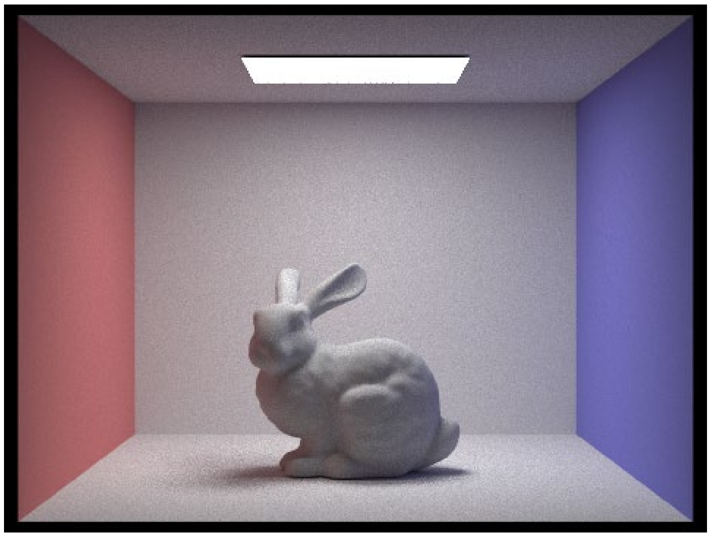

Direct illumination. s = 64, l = 32.

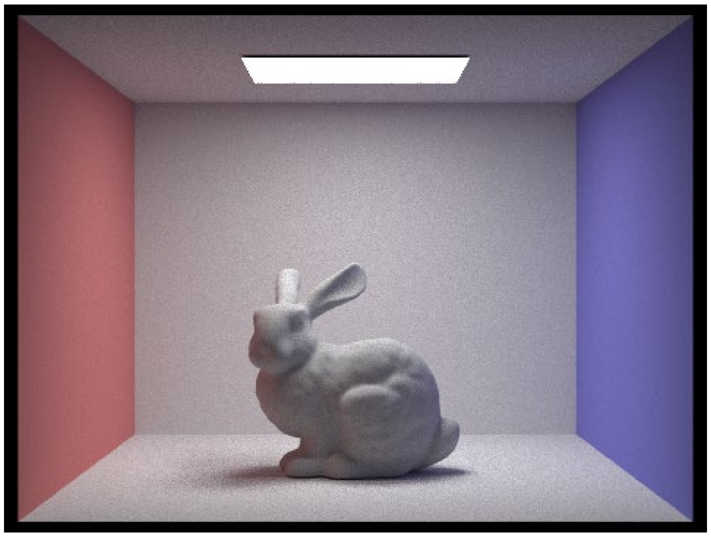

Indirect illumination. s = 64, l = 32.

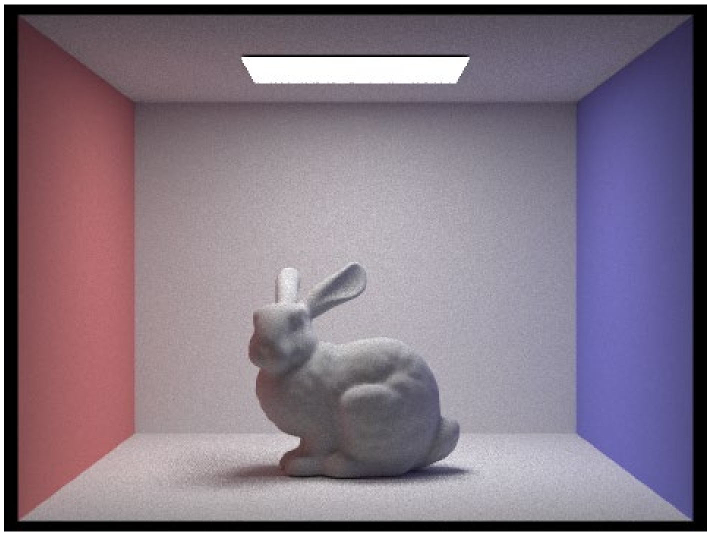
CBbunny.dae with different max depth.


max_rap_depth = 0.

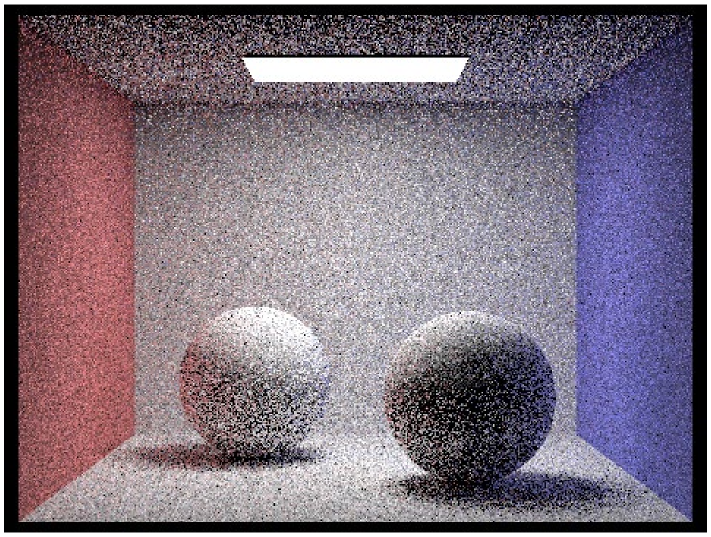

max_rap_depth = 1.

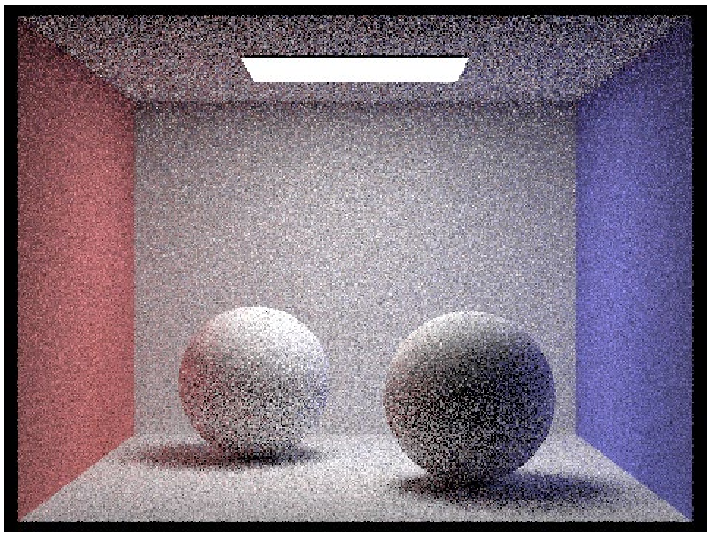

max_rap_depth = 2.

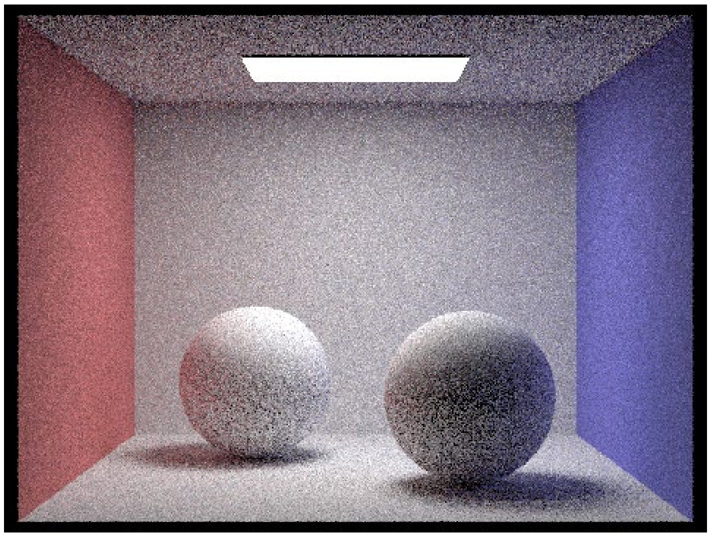

max_rap_depth = 3.

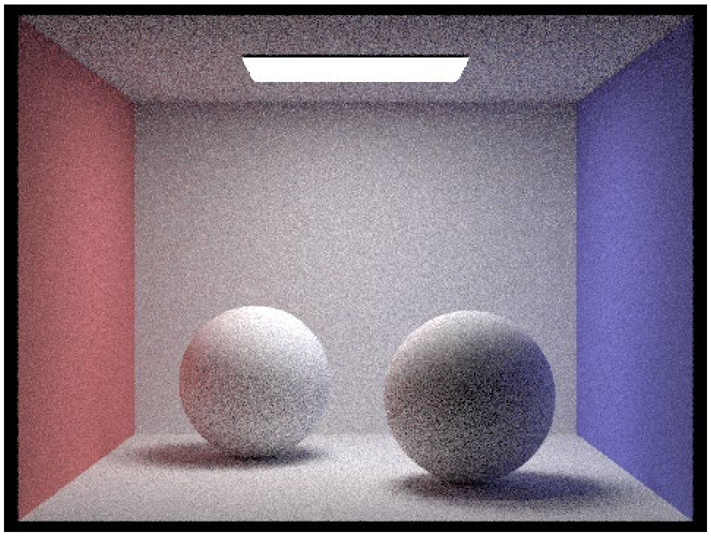

max_rap_depth = 100.

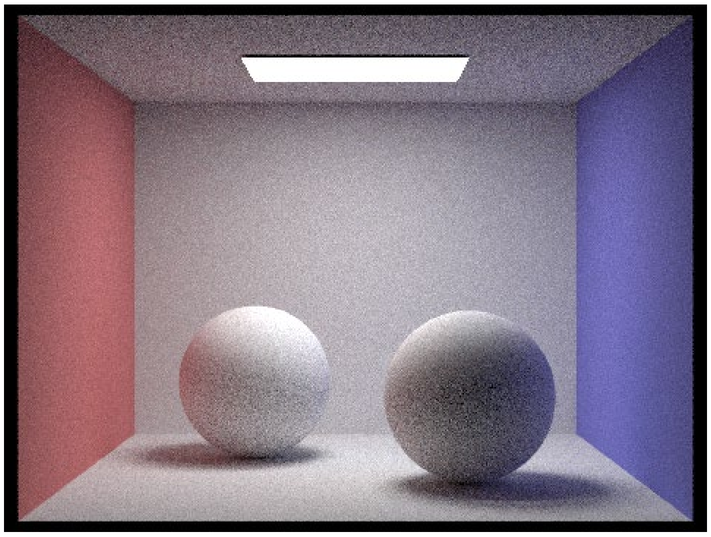
spheres.dae with diffeent sample rates..

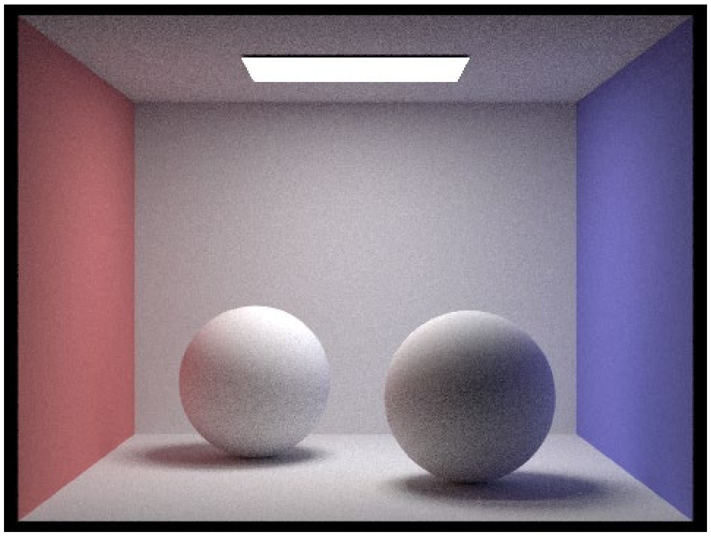

s = 1.

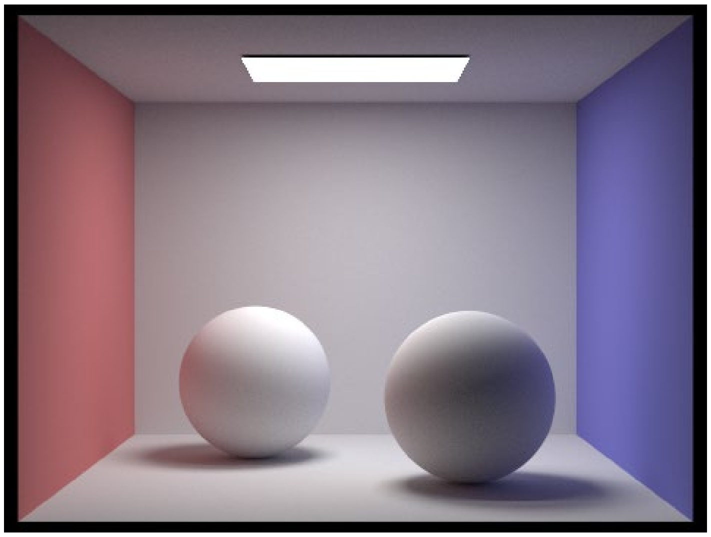

s= 2.


s = 4.


s = 8.


s= 16.


s= 64.

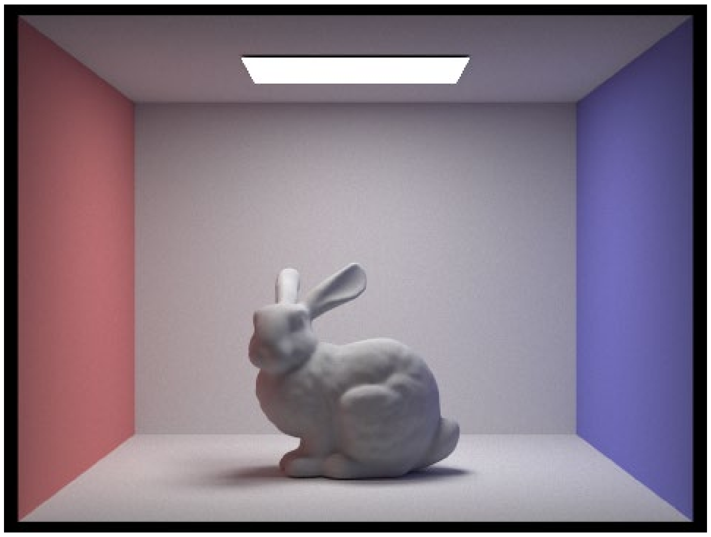

s = 1024.

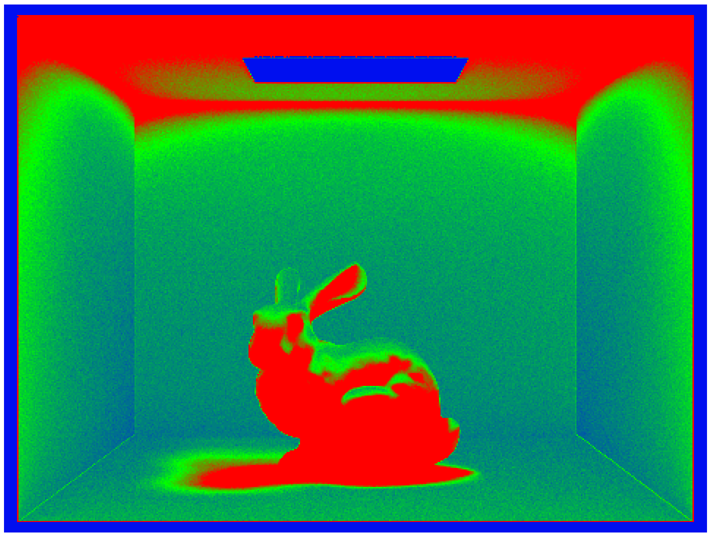
Part 5: Adaptive Sampling

implementation of the adaptive sampling:

I keep track of the variance and when the variance is small thaner than the max_tolerance*mu, I have enough confidence that the pixel value is good enough to exit the loop.

bunny.dae with the sampling rate image.


Bunny.


Bunny rate.

A Few Notes On Webpages

Here are a few problems students have encountered in the past. You will probably encounter these problems at some point, so don't wait until right before the deadline to check that everything is working. Test your website on the instructional machines early!

- Your main report page should be called index.html.
- Be sure to include and turn in all of the other files (such as images) that are linked in your report!
- Use only *relative* paths to files, such as

  ```
  "./images/image.jpg"
  ```

  Do *NOT* use absolute paths, such as

  ```
  "/Users/student/Desktop/image.jpg"
  ```

- Pay close attention to your filename extensions. Remember that on UNIX systems (such as the instructional machines), capitalization matters.

```
.png != .jpeg != .jpg != .JPG
```

- Be sure to adjust the permissions on your files so that they are world readable. For more information on this please see this tutorial: http://www.grymoire.com/Unix/Permissions.html
- And again, test your website on the instructional machines early!