Advanced Graphics Coursework 2

Samuel Budd, Oli Robinson

February 23, 2017

1 Part 1: Fresnel and Schlick

We used MatLab to produce 2 graphs of Fresnel reflectance for a dielectric material. To do this, we ranged over incident angles (θ) between 0 and 90 degrees, and for each value, we calculated the reflectance of parallel light and perpendicular light using the equations:

$$Parallel(\theta) = \left(\frac{1.0 * cos(\theta) - 1.4 * \sqrt{1 - (\frac{1.0}{1.4} sin(\theta))^2}}{1.0 * cos(\theta) + 1.4 * \sqrt{1 - (\frac{1.0}{1.4} sin(\theta))^2}} \right)^2 \tag{1}$$

$$Perpendicular(\theta) = \left(\frac{1.0 * \sqrt{1 - (\frac{1.0}{1.4} sin(\theta))^2} - 1.4 * cos(\theta)}{1.0 * \sqrt{1 - (\frac{1.0}{1.4} sin(\theta))^2} + 1.4 * cos(\theta)} \right)^2 \tag{2}$$

Where $\sqrt{1 - (\frac{1.0}{1.4} sin(\theta))^2}$ gives the angle of refraction in the material, and the values of 1.0 and 1.4 refer to the index of refraction for the two materials. The above equations calculate the reflectance of light going from air (index of 1.0) into a different material (index of 1.4). We then swapped the two indices to calculate the same values from light exiting the material rather than entering it.

Then, for light entering the material, we also calculated Schlick's approximation of the unpolarized Fresnel reflectance. We did this using the equation:

$$Schlick' sapproximation = R_0 + (1 - R_0)(1 - cos(\theta)^5 \tag{3}$$

Where R_0 if the reflectance at normal incidence, which we took as $R_0 = Parallel(0)$. The figures 1 and 2 show our results for light entering and exiting the material respectively.

We found Brewster's angle to be 54 degrees for light entering the material, and the critical angle to be 45.6 degrees for light exiting it.

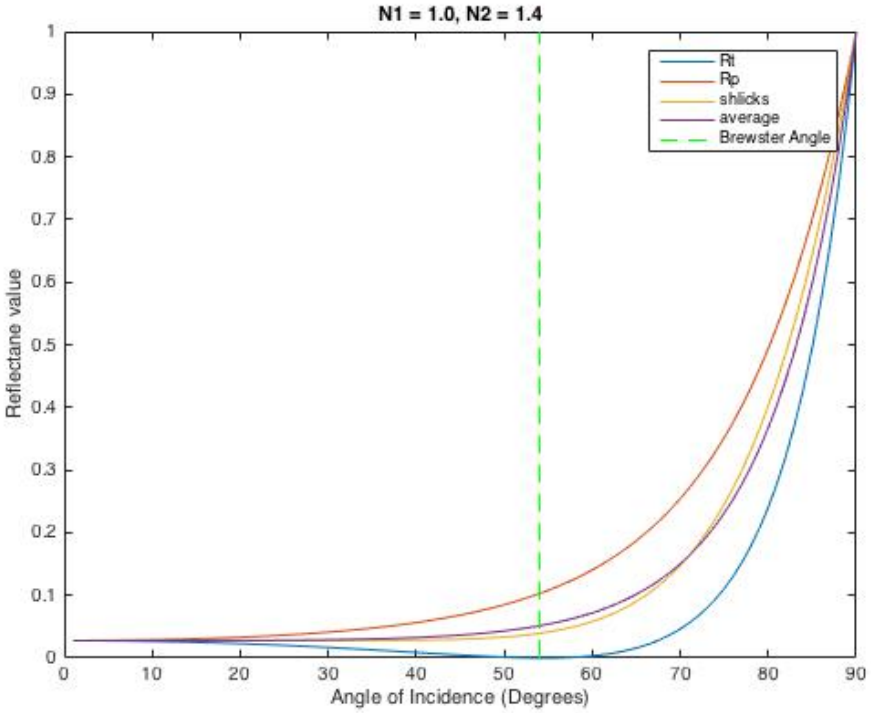


Figure 1: Figure showing Fresnel calculations and Schlick approximation of reflectance of light passing from air into an material with refraction index 1.4. Dotted line shows Brewster's angle (54)

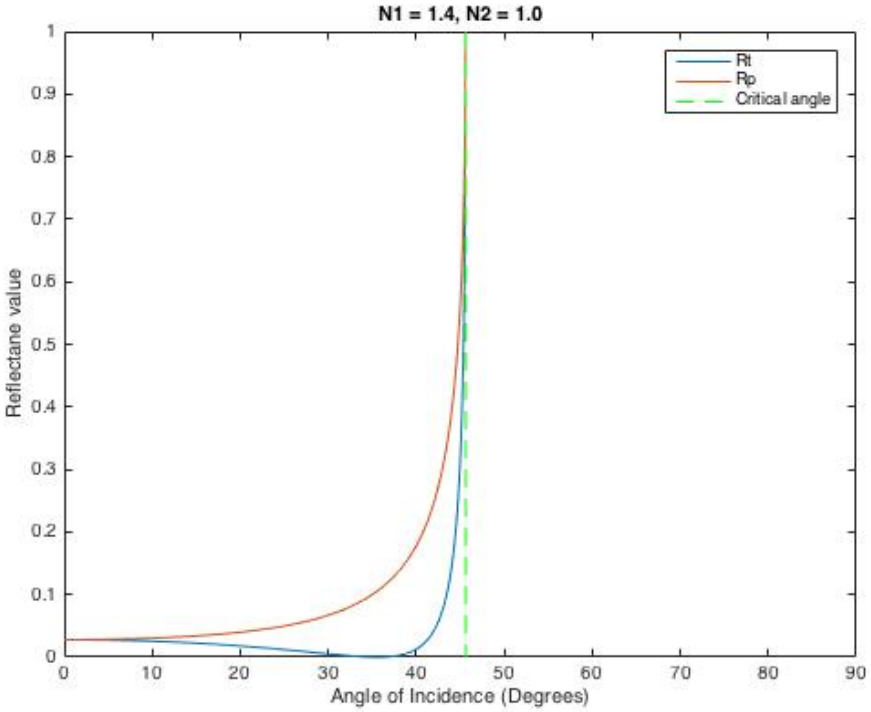


Figure 2: Figure showing Fresnel calculations of reflectance of light passing from a material with refraction index 1.4 into air. Dotted line shows critical angle (45.6)

2 Part 2: Importance Sampling

We started by using the code provided in the last assignment to load the Grace Cathedral environment map (.pfm). The next step was to re-map this image to intensity space using the formula $I = (R + G + B)/3$ for each pixel. The next step was to build our PDFs, we iterated over each row and calculated the total intensity stored in each row, making sure to scale the intensity stored in each pixel by the solid angle term $\sin(\theta)$. We calculated θ with the formula $\theta = rowIndex/mapHeight * \pi$ and then scaled the intensity of each pixel by the formula $scaledI = I * \sin(\theta)$. At this point we had the total sum of intensities for each row scaled appropriately. Next we calculated the PDF to select a row to sample from by the formula $rowProb = rowSum/totalIntensity$ where $totalIntensity$ equalled the sum of all scaled intensities in the image. At this point we could create the CDF required to select the row to sample from by using the formula by the standard formula for building a CDF from a PDF. We then created the a PDF for each row of the image by using the formula $pixelProb = pixelIntensity/rowSum$, this created a PDF for each row of the image. We then used these PDFs to create a CDF per row using the standard method for building a CDF from a PDF.

At this point we had everything we needed to begin sampling our image. To do this we generated $numberOfSamples$ random numbers in the range $[0, 1]$ and using each of these numbers selected a row to sample from by iterating through the CDF until we find a probability value $\geq$ the random number for that sample. Once we had selected the row we generated another random number in the same range to select the pixel to sample from that row, by iterating through the CDF for the row selected for that sample until we find a probability value $\geq$ our random number for that row. After doing this for the number of samples we had a set of index pairs in our map to use as our samples.

Next we wanted to display the samples on the image. We did this by changing the colour of each sampled pixel in our environment map to blue along with the pixels local 5x5 neighbourhood to make each sample more visible.

We returned the sampled indices, the choose row PDF and the per row PDF for use in Part 3.

We then performed Gamma Correction on each image generated with gamma 2.2.

The Following images were generated for 64, 256 and 1024 samples respectively:

Figure 3: Environment map with 64 samples

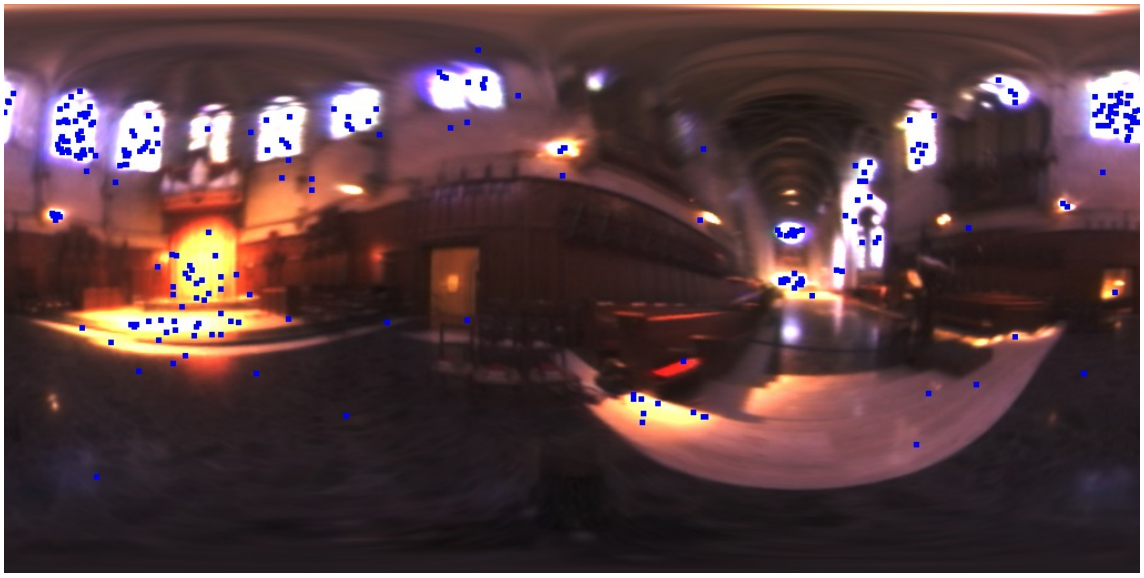


Figure 4: Environment map with 256 samples

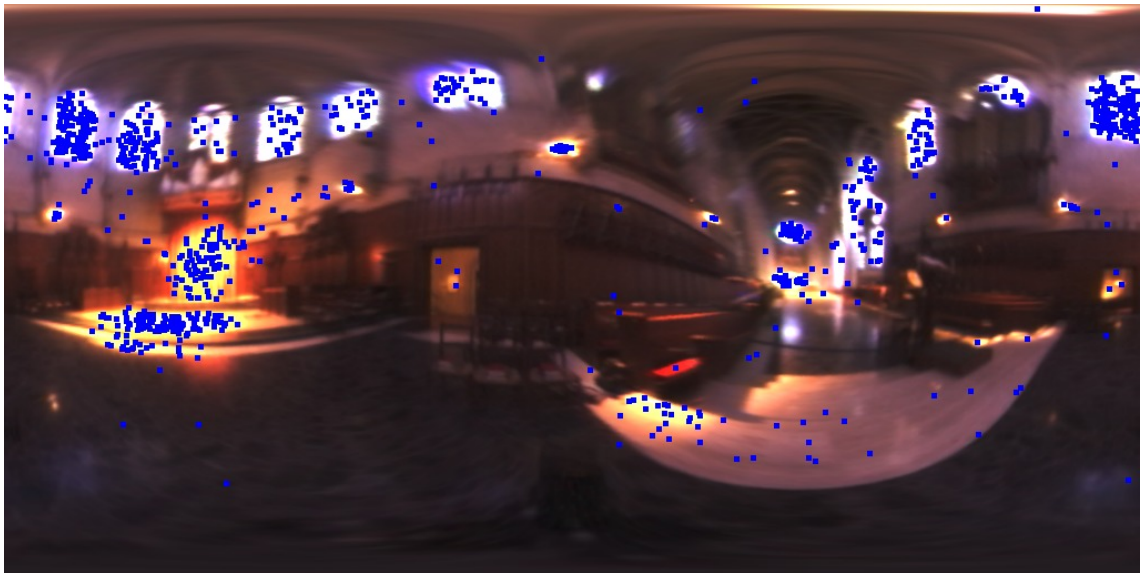


Figure 5: Environment map with 1024 samples

We can see from the images how the pixels with a higher intensity were more likely to be sampled.

3 Part 3: Relighting a sphere

To render a sphere based on the samples taken in the previous section, we began by defining a new image to hold the sphere, and calculated the sample contributions for each pixel on the sphere. To do this, we iterated over every pixel, computing the normal of the sphere, and then iterating over each sampled point from the previous part, calculating the corresponding polar and azimuthal angles for the given point:

```
#Polar (theta) and azimuthal (phi) angles
theta = (float(j)/float(height))*pi
phi = (float(i)/float(width))*pi*2
```

We then used these to calculate the lighting direction vector ω_i as such :

```
x = sin(theta)*cos(phi)*radius
y = cos(theta)*radius
z = sin(theta)*sin(phi)*radius
```

Using this and the normal, we could calculate a value for $cos(\theta)$, which we used to calculate the contribution of each sample point to the current pixel. We did this calculation in two ways to produce two different results. We first took a simple average over the samples, and took the shading of each pixel on the sphere as:

$$L_r(\omega_0) = \frac{1}{N}\left(\sum_{i=1}^{N}(\frac{\rho_d}{\pi} * cos(\theta) * L_i(\omega_i)\right) \tag{4}$$

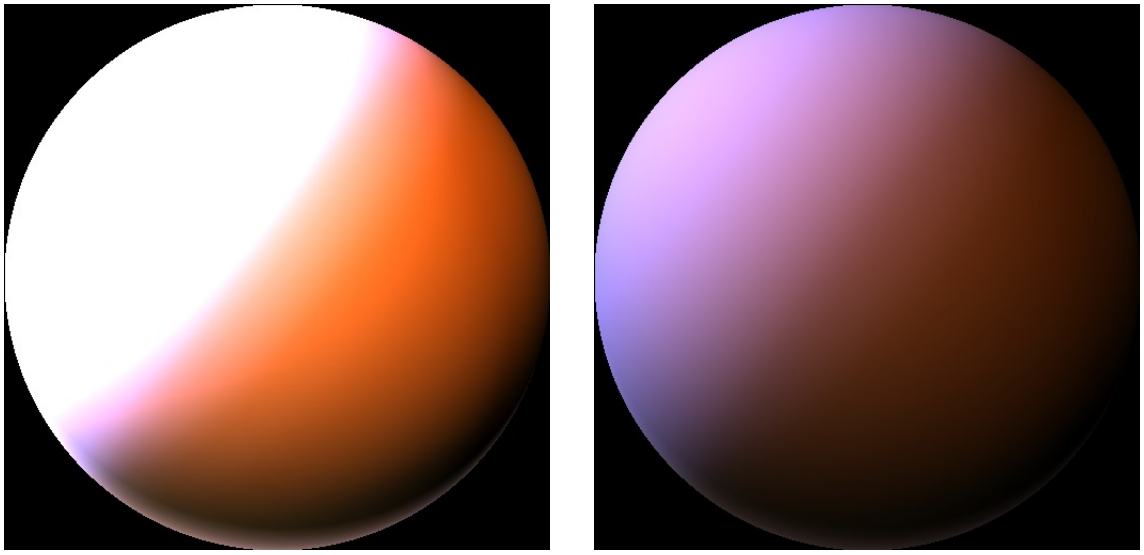
Where $N = Numberofsamplesused$, ρ_d is the reflectivity of the sphere, which is 1, and $L_i(\omega_i)$ is the RGB value of the current sample.

We then calculated the same thing, but weighting the shading value based on the probability of picking the specific lighting direction ω_i of the given sample. This was done using the equation:

$$L_r(\omega_0) = \frac{1}{N}\left(\sum_{i=1}^{N}(\frac{\rho_d}{\pi} * cos(\theta) * I * \frac{(\hat{R}, \hat{G}, \hat{B})}{P(x,y)})\right) \tag{5}$$

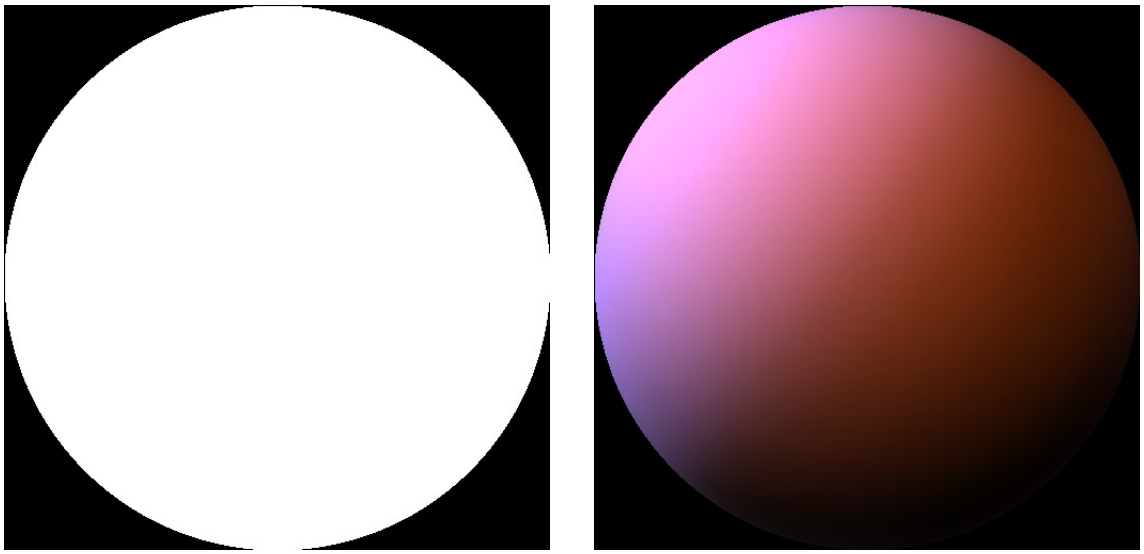
Where (R,G,B) are the colour components of the sampled pixel, and $\hat{R} = \frac{R}{\sqrt{R^2+G^2+B^2}}$ (and similarly for $\hat{G}$ and $\hat{B}$.

We did this using both these methods for sets of 64, 256, and 1024 samples drawn from the Grace Cathedral scene. We ended up with a set of spheres, all of which came out very saturated. We applied the simple linear tone mapping as created in HW1, and applied exposure correction too. The results are shown and labelled below.



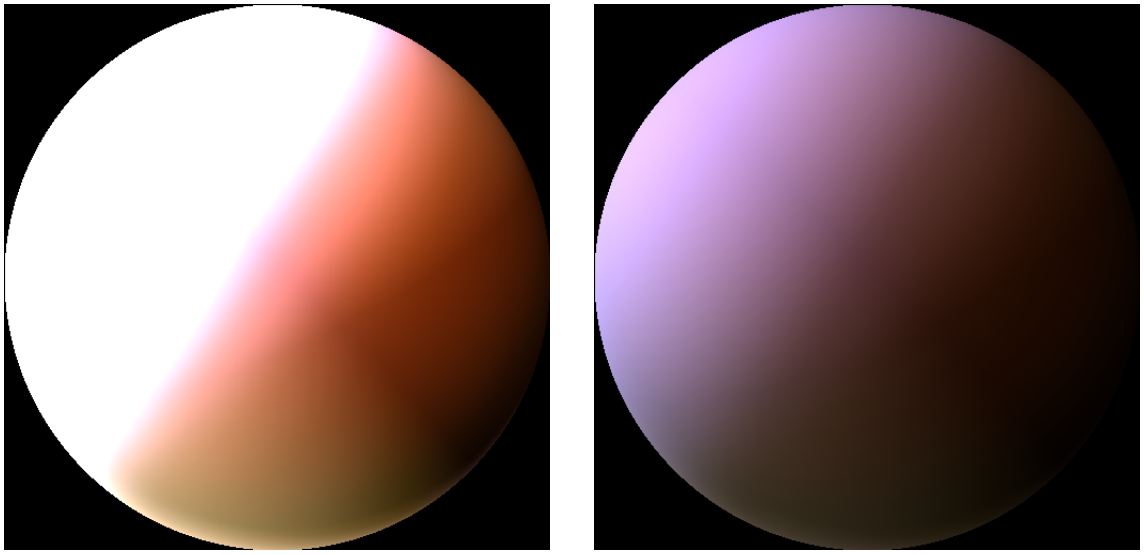
(a) Raw PPM (b) Tone mapped PPM

Figure 6: Spheres generated using equation (4) with 64 samples



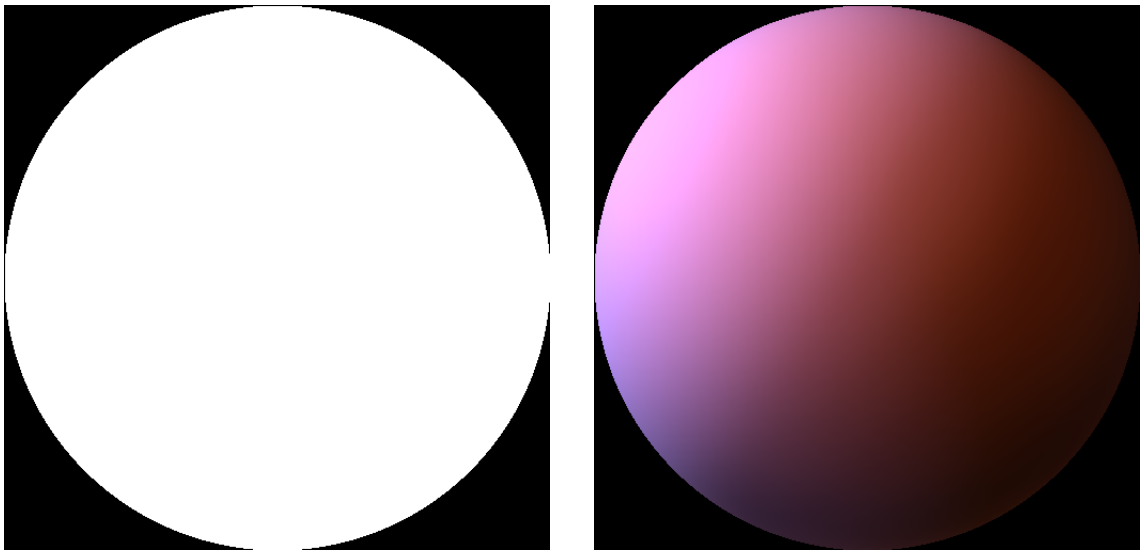
(a) Raw PPM (b) Tone mapped PPM

Figure 7: Spheres generated using equation (5) with 64 samples



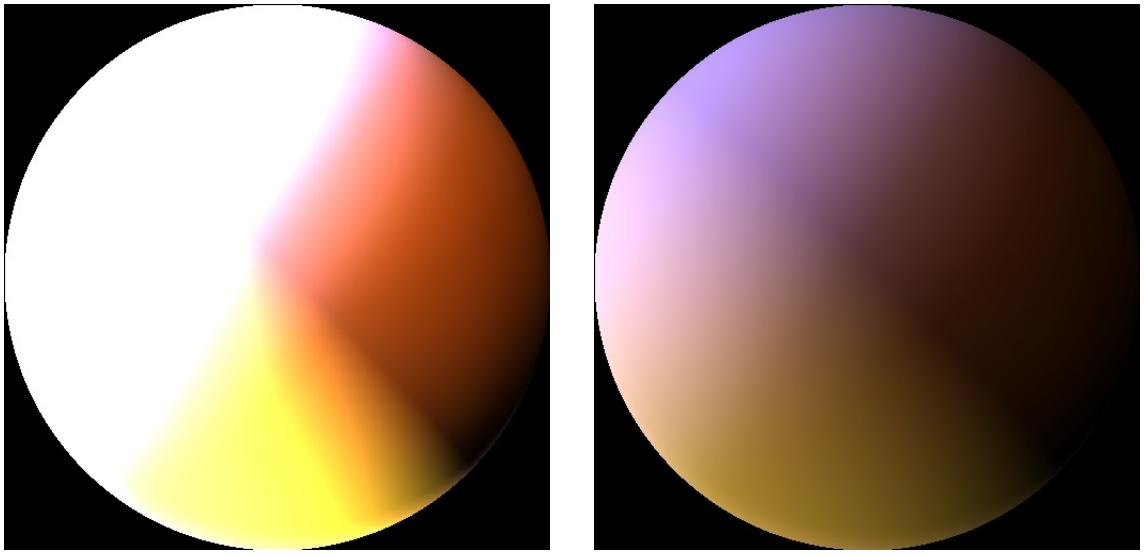
(a) Raw PPM (b) Tone mapped PPM

Figure 8: Spheres generated using equation (4) with 256 samples



(a) Raw PPM (b) Tone mapped PPM

Figure 9: Spheres generated using equation (5) with 256 samples



(a) Raw PPM (b) Tone mapped PPM

Figure 10: Spheres generated using equation (4) with 1024 samples

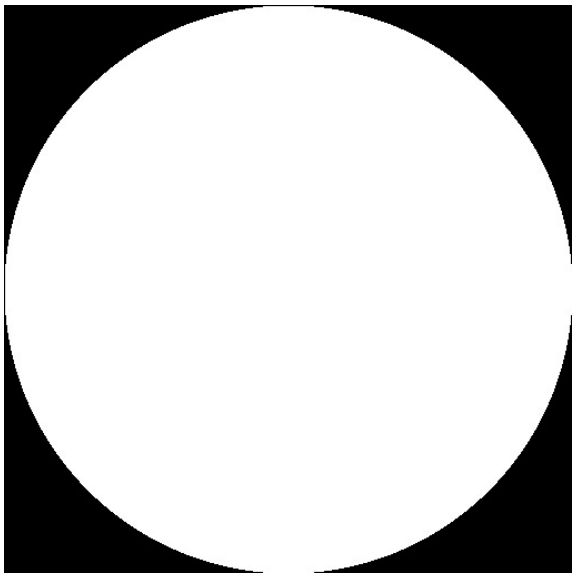


Figure 11: Sphere generated using equation (5) with 1024 samples: Raw PPM

We can see from our various spheres - especially evident in those produced using equation (4) how bias affects MC sampling techniques - running the same code multiple times produced different outputs but we think the end results all appear pleasing to the eye.

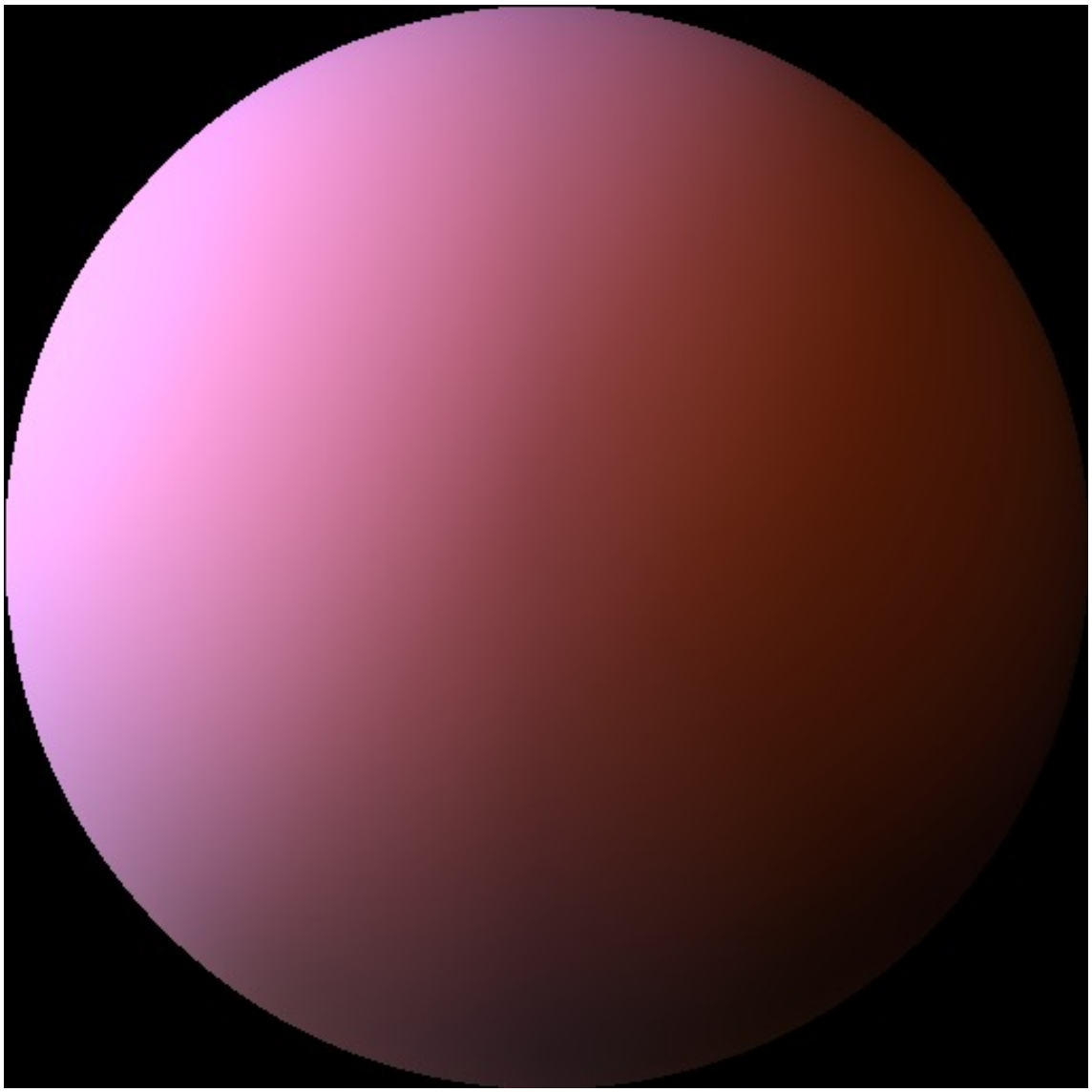


Figure 12: Sphere generated using equation (5) with 1024 samples: Tone mapped PPM

4 Part 4: PBRT

Below are our 4 diffuse spheres created by PBRT using 8, 16, 32 and 64 samples respectively. We can see from these renderings how the noise on the spheres decrease significantly as we increase the number of samples.

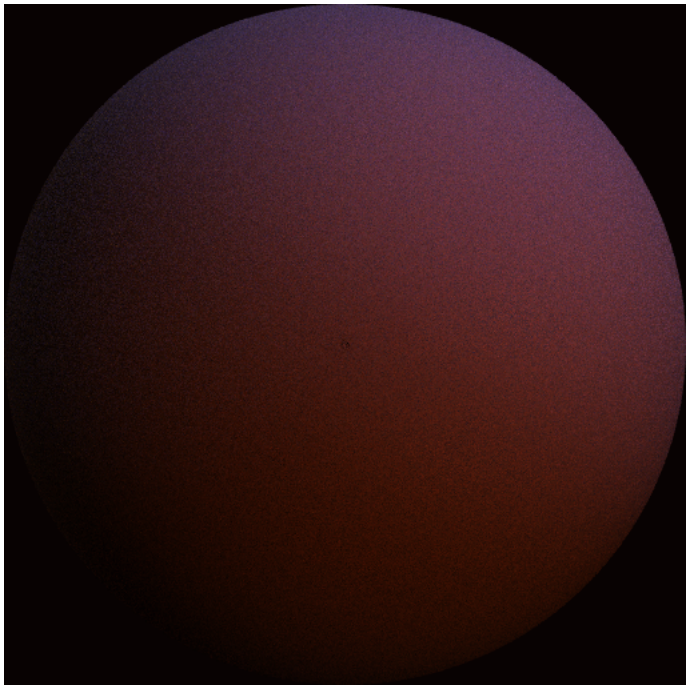


Figure 13: PBRT diffuse sphere generated with 8 samples

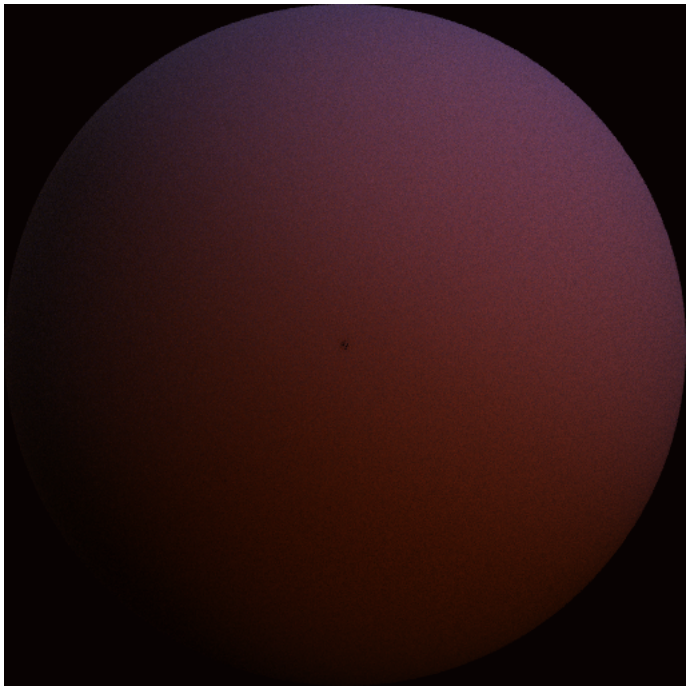


Figure 14: PBRT diffuse sphere generated with 16 samples

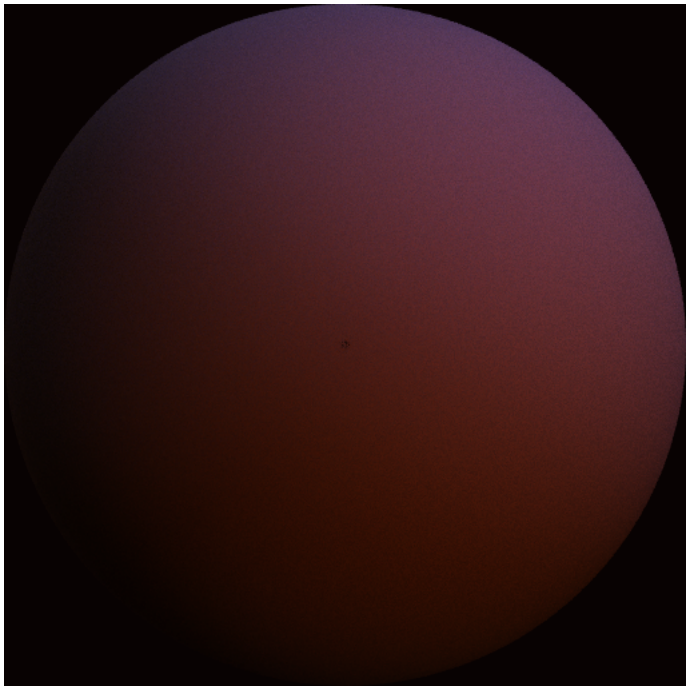


Figure 15: PBRT diffuse sphere generated with 32 samples

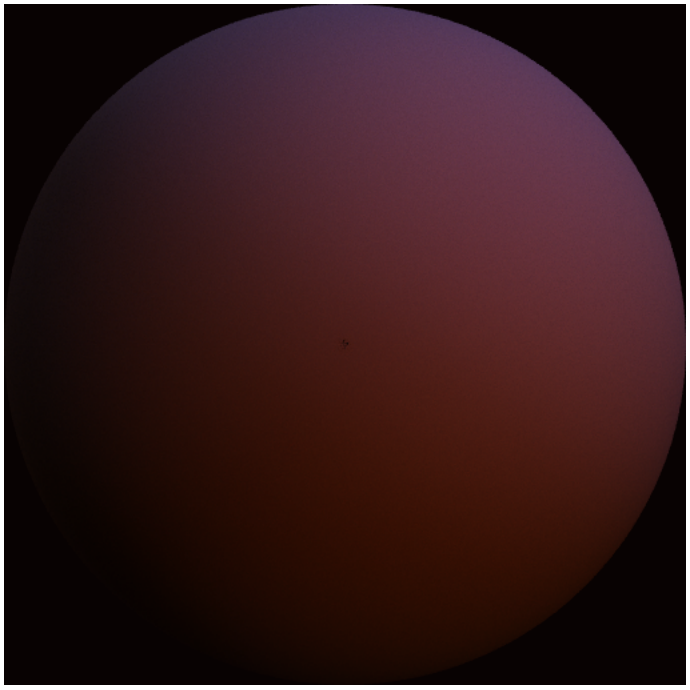


Figure 16: PBRT diffuse sphere generated with 64 samples