CSE306

elenamrdja

May 2023

1 Introduction

For this assignment, I had initially planned to create a library-like structure with cleanly separated classes and functions. However, due to unforeseen circumstances, I didn't have as much time as I wanted to organize everything in the desired manner. Thus although perfectly functional, some of my classes may seem unfinished or stylistically imperfect.

Despite these setbacks, I was still able to implement some optional features, including the Fresnel Law. This effect is particularly noticeable when light passes through transparent materials such as glass, which is the reason why I decided to leave the option to the user to render either the desired sphere from the lecture notes, or a cat, by using the object type variable as an indicator. By incorporating the Fresnel Law into my program, I was able to create more realistic renderings of scenes with reflective surfaces, which is especially noticeable on the hallow (object id = 3) and transparent sphere (object id = 4).

On the following pages are a few examples of the images created with my program:

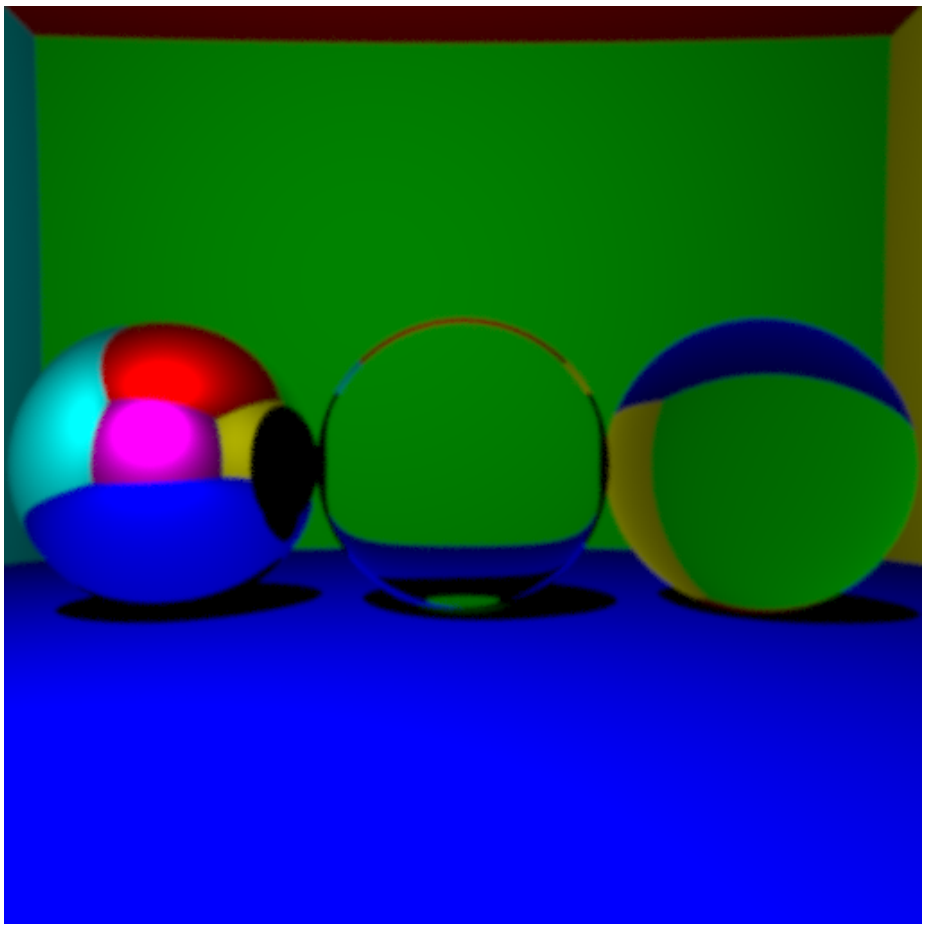

Figure 1: The computation lasts 30 seconds. This image is a 512x512 image that uses 64 rays per pixel, with a maximum ray depth of 5. The light source positioned at Vector(-10, 20, 40), light of intensity of 2e10 illuminating the scene. The spheres have a uniform albedo of Vector(0.5, 0.5, 0.5).
Image shows a mirror sphere, a hallow sphere and a transparent sphere, without with indirect lightning, and without of the effect of Fresnel law

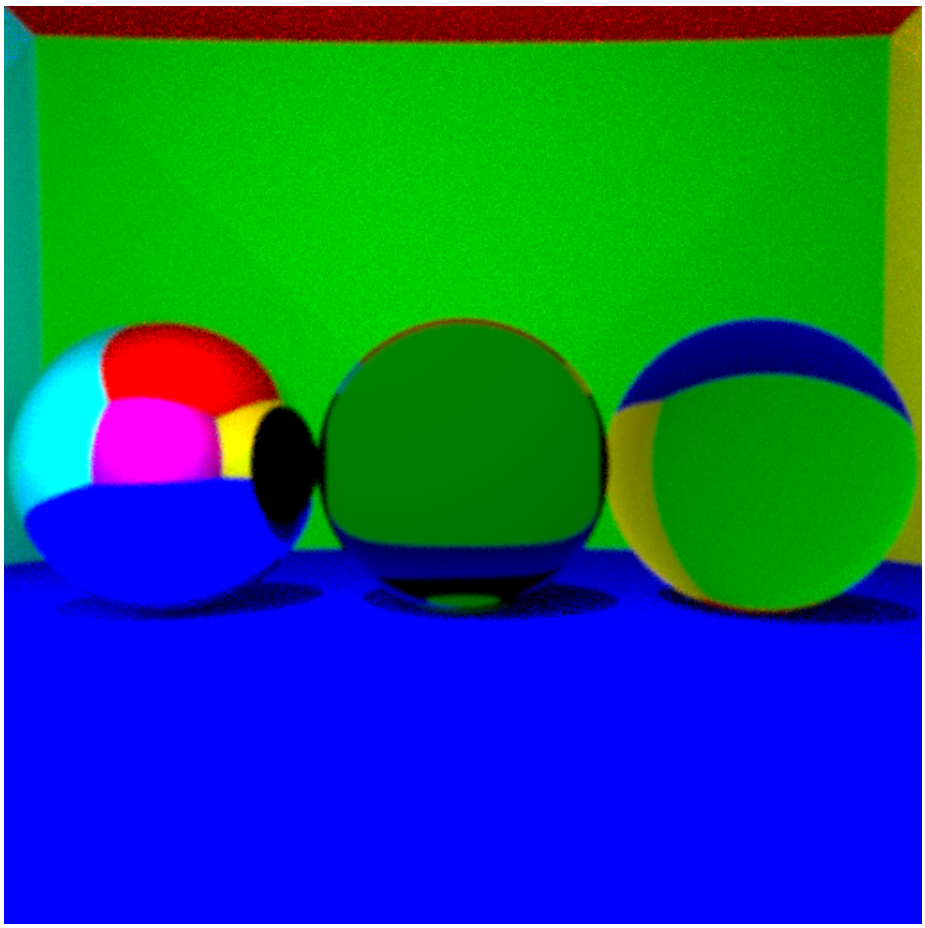

Figure 2: The computation lasts 30 seconds. This image is a 512x512 image that uses 64 rays per pixel, with a maximum ray depth of 5. The light source positioned at Vector(-10, 20, 40), light of intensity of 2e10 illuminating the scene. The spheres have a uniform albedo of Vector(0.5, 0.5, 0.5).
Image shows a mirror sphere, a hallow sphere and a transparent sphere, illuminated with indirect lightning, and without of the effect of Fresnel law

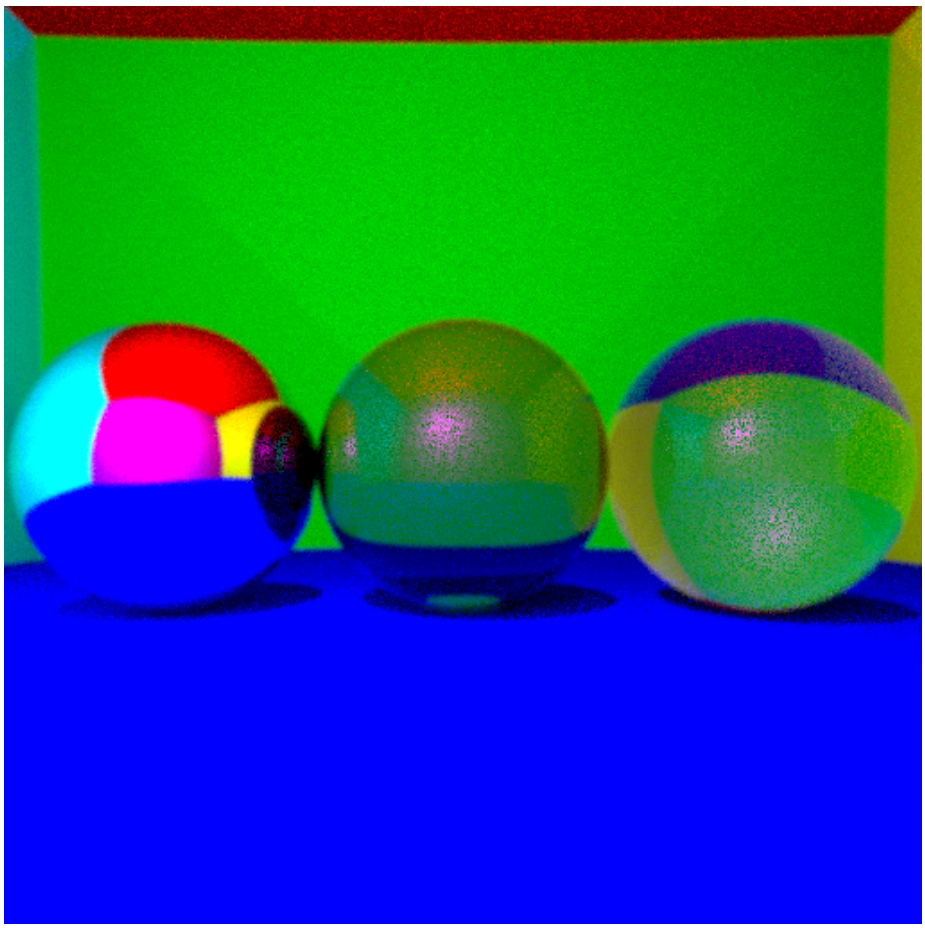

Figure 3: The computation lasts 30 seconds. This image is a 512x512 image that uses 64 rays per pixel, with a maximum ray depth of 5. The light source positioned at Vector(-10, 20, 40), light of intensity of 2e10 illuminating the scene. The spheres have a uniform albedo of Vector(0.5, 0.5, 0.5).
Image shows a mirror sphere, a hallow sphere and a transparent sphere, illuminated with indirect lightning, and with the effect of Fresnel law

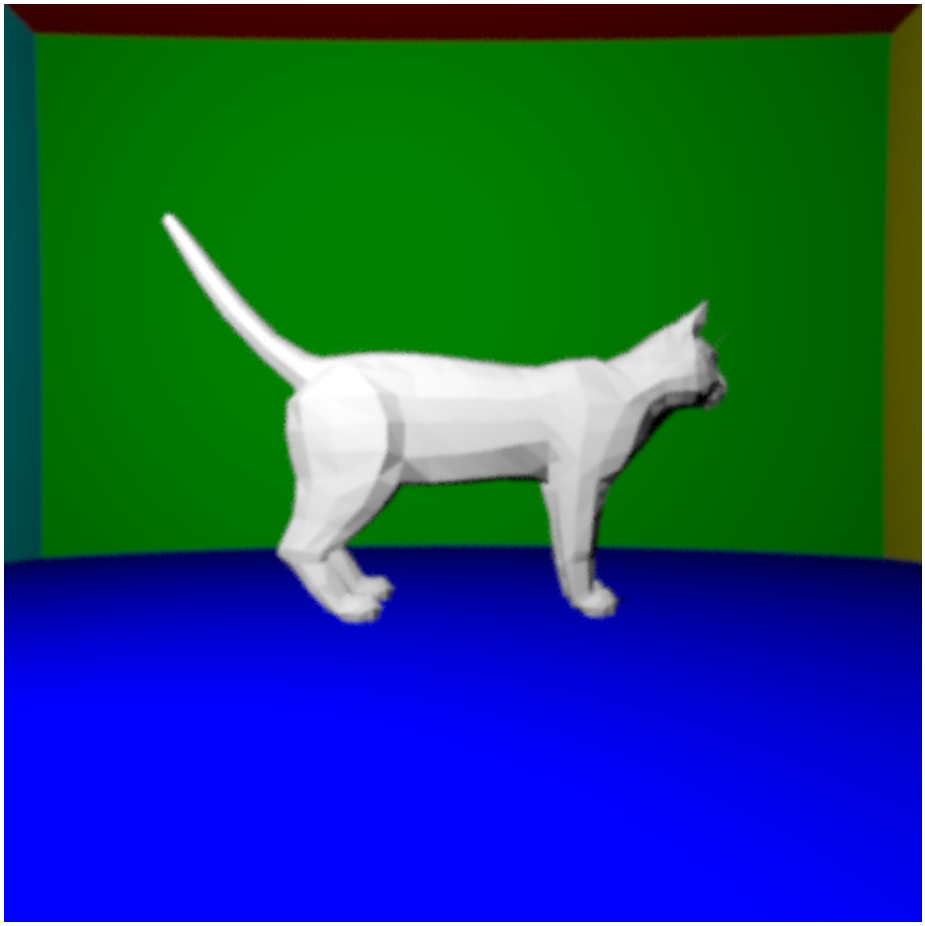

Figure 4: This image is a 512x512 image that uses 64 rays per pixel, with a maximum ray depth of 5. The light source positioned at Vector(-10, 20, 40), light of intensity of 2e10 illuminating the scene. Instead of the three spheares the image shows a mesh cat. This one does not use the Fresnel law, nor the indirect lighting, which is the process of simulating the way light bounces around a scene and interacts with different objects. Therefore, the image lacks the subtle variations in brightness and color that come with indirect lighting.

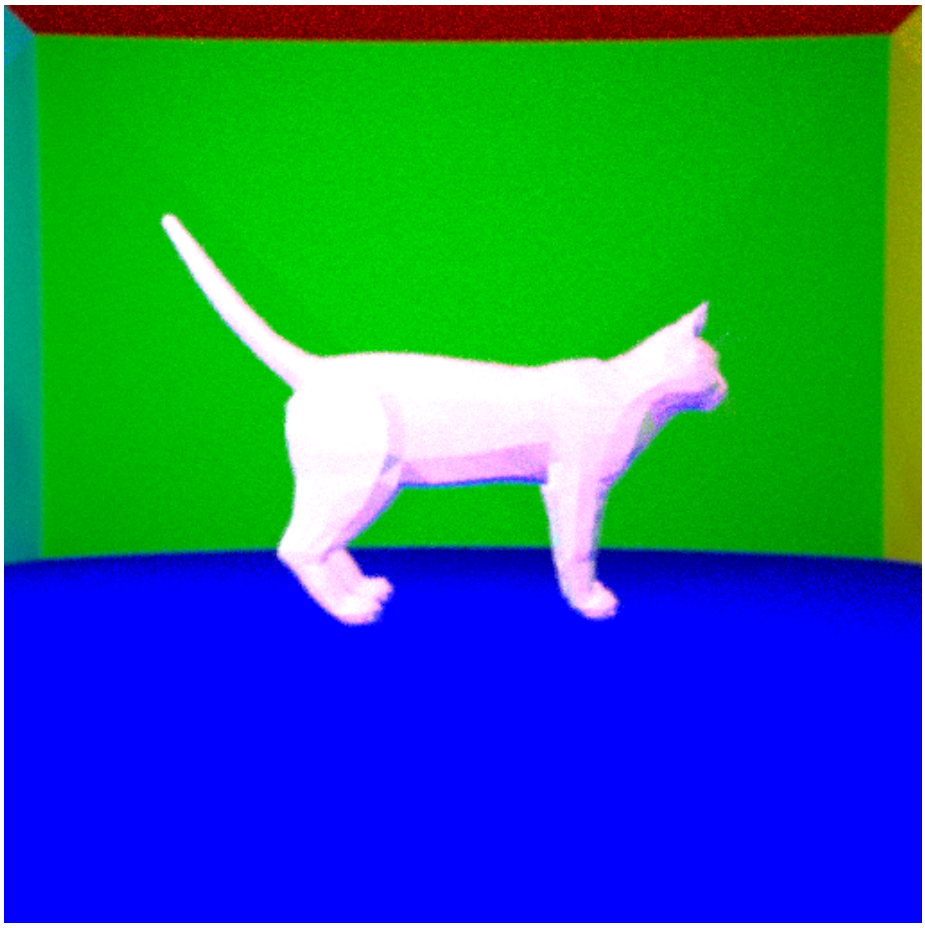

Figure 5: The computation lasts 39 seconds with parallelization. This image is a 512x512 image that uses 64 rays per pixel, with a maximum ray depth of 5. The light source positioned at Vector(-10, 20, 40), light of intensity of 2e10 illuminating the scene. The image shows a mesh cat, illuminated by indirect lighting, but does not use the fresnel law.