Project 3: Reflection

Name: Minsuk Kim(m.kim)
Instructor: Dr. Gary Herron
Class: CS500
Semester: Spring 2022
Date: 3/7/2022

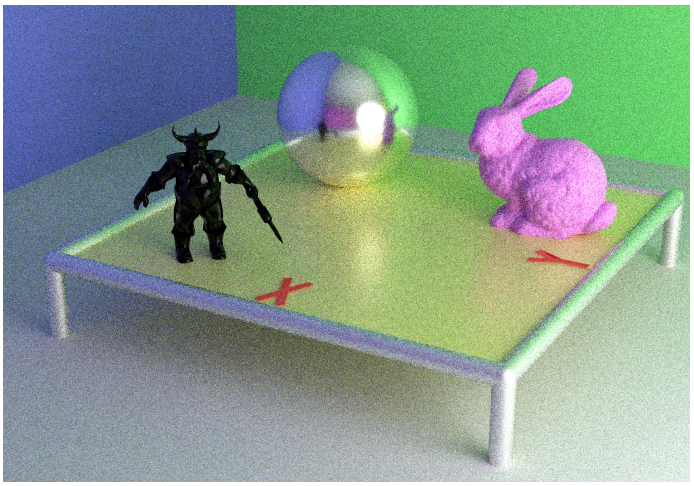


Table of Contents

Introduction

Overview

The main purpose of the project is to implement the full micro-facet BRDF with the lighting equation. The program will calculate the lighting value on each path tracing loop to evaluate the reflection and diffuse color. The generated ray stored cumulative color information by calculating the BRDF. I started the project with the previous project that was about path tracing. I also used an external library Assimp that read the object files and BVH for bounding volume hierarchy.

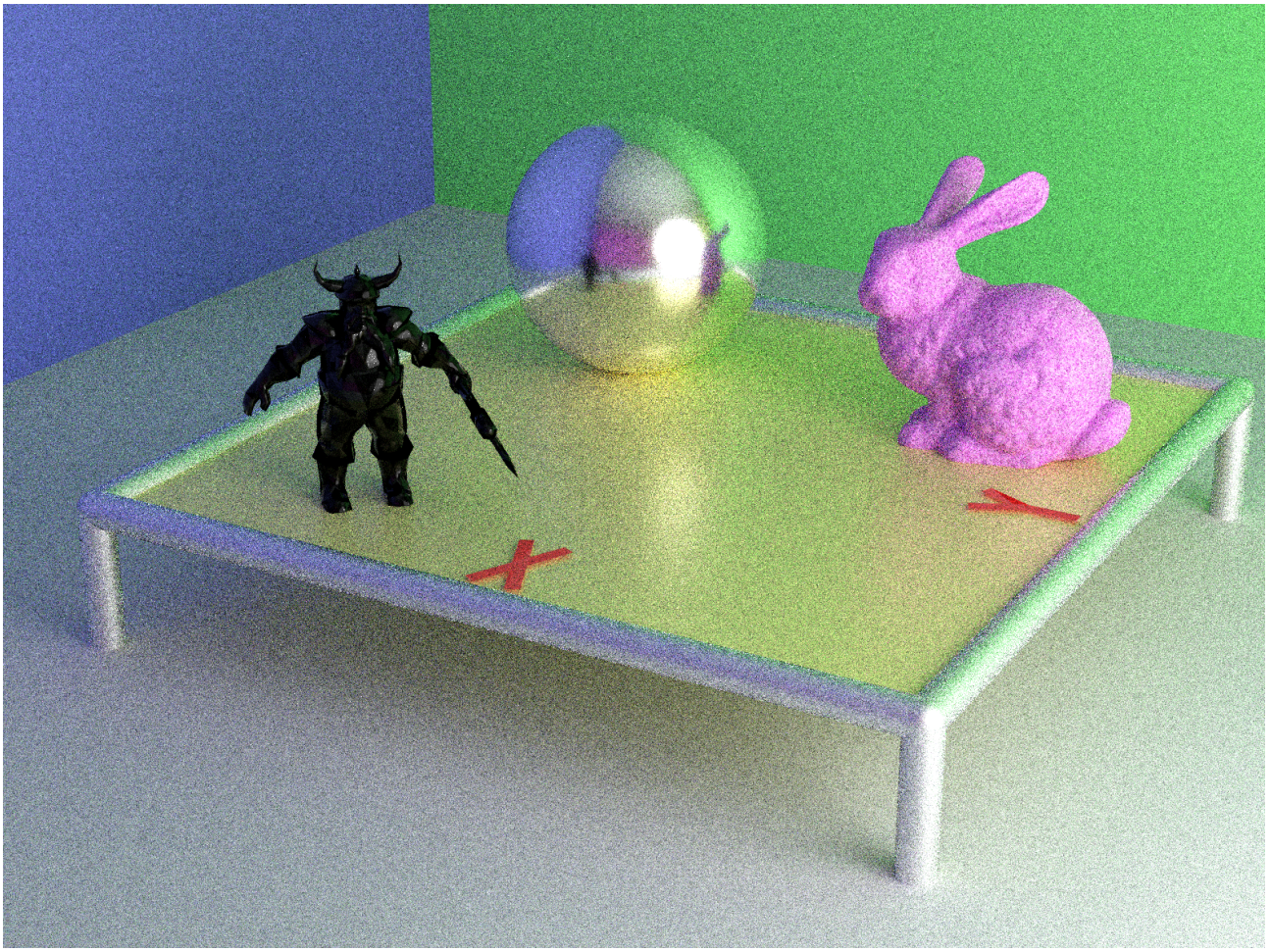
Implementation

First, the program loaded the scene file and created the shapes for each object file. The program casts ray for each pixel. When the ray collides with a certain object that is not a light object, the ray will generate another ray that towards a random direction. The random direction is distributed vector around a normal vector in the hit position. The program also uses the reflection feature that random direction is distributed vector around a reflective vector. Then, when the ray hits the light object, the program reads cumulative data that stored previous hit objects' BRDF values. The program also uses the explicit light connection that shoots another ray from the hit position to a random light object. When the ray that shoots toward the light object hits the same light object, the program adds the BRDF value to the current color. The program repeatedly generates rays to the same position for better accurate light calculation. The program also uses anti-aliasing by changing random offset to a pixel position in each pass.

Result Images

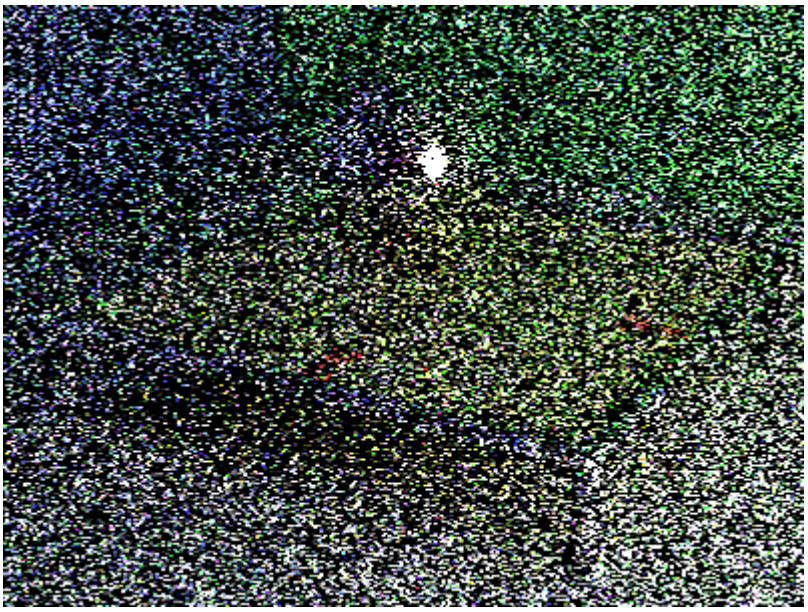
sample image

image with GGX model with 1600X1200 with various objects

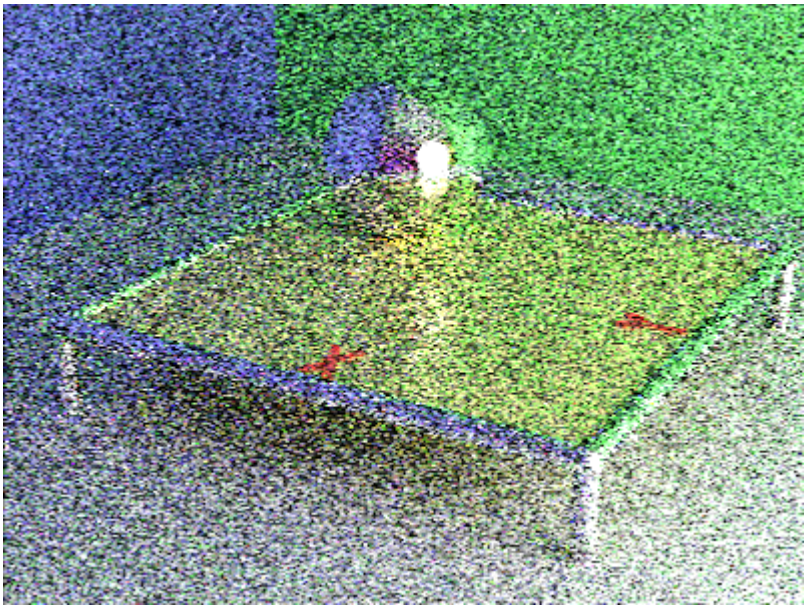


images with phong model

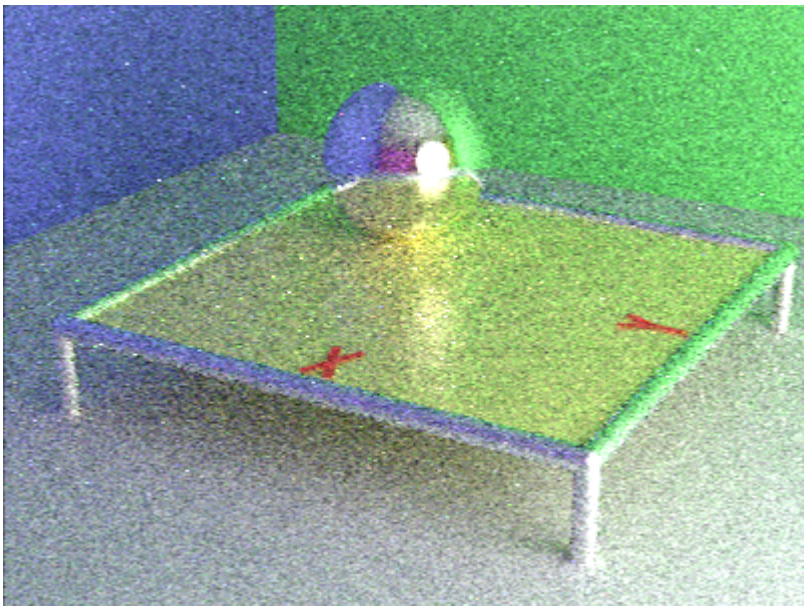
1 pass image with phong model



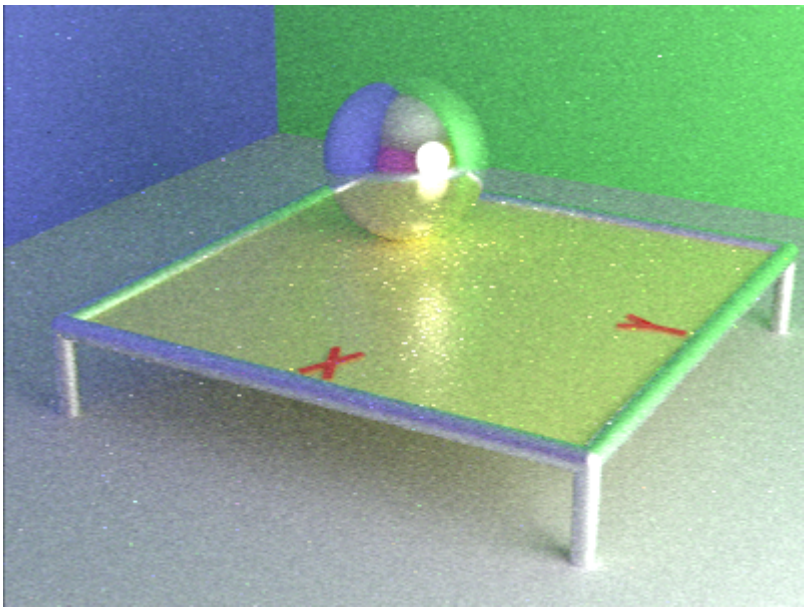
8 passes with phong model



64 passes with phong model

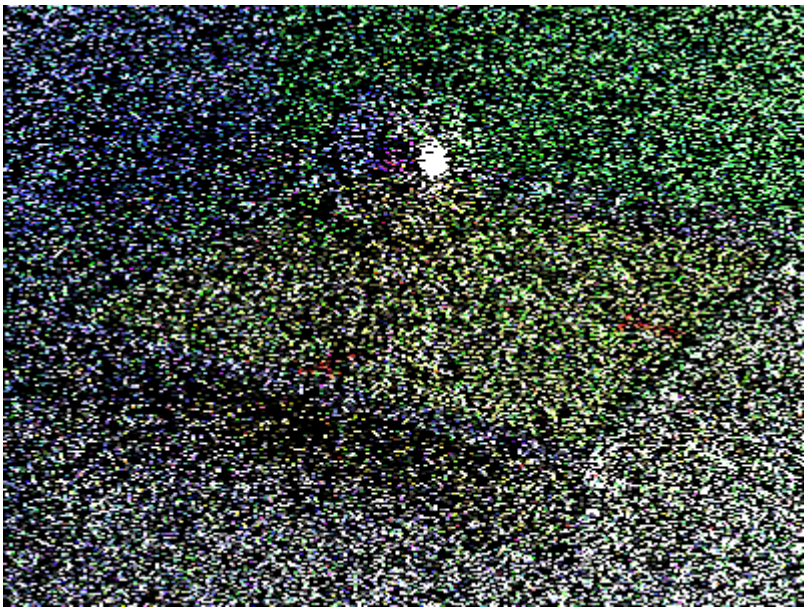


512 passes with phong model

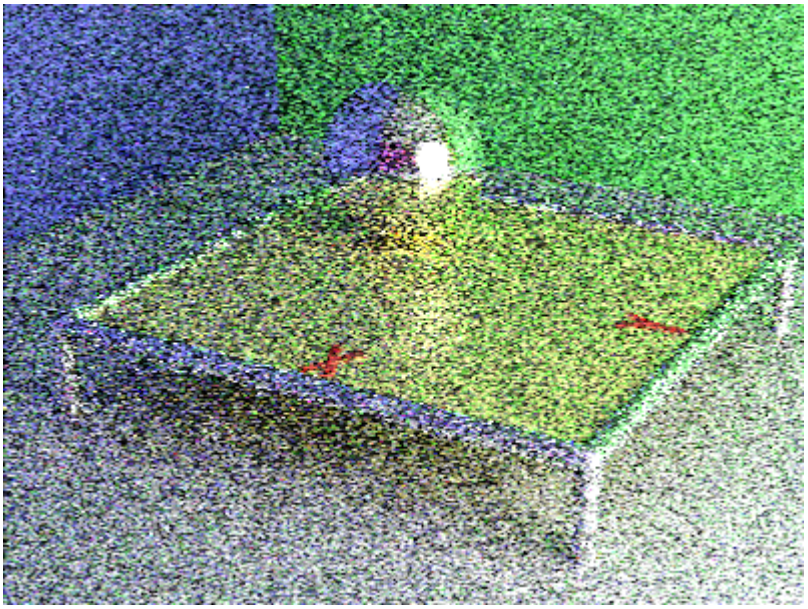


images with GGX model

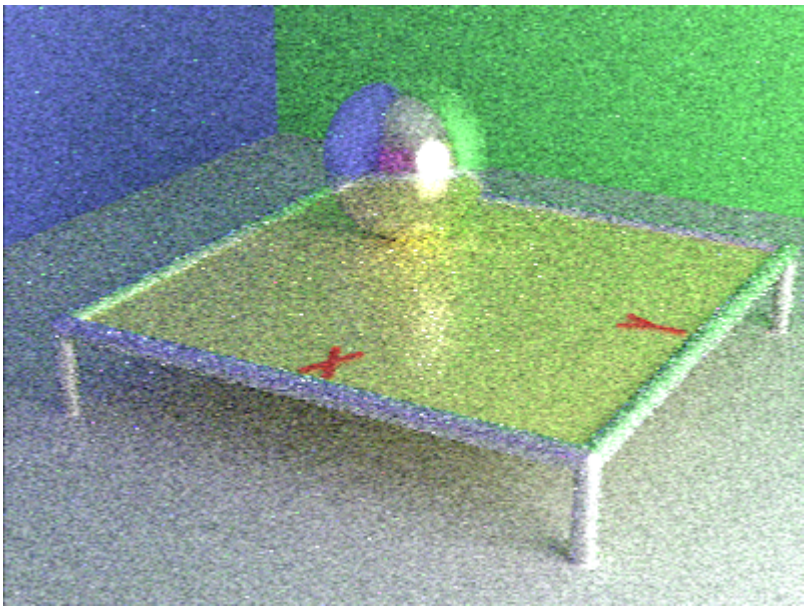
1 pass image with GGX model



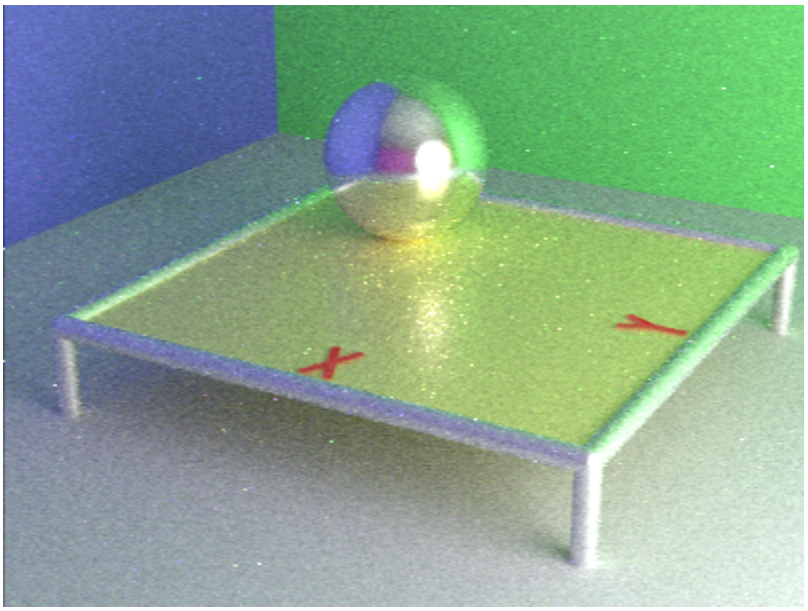
8 passes image with GGX model



64 passes image with GGX model



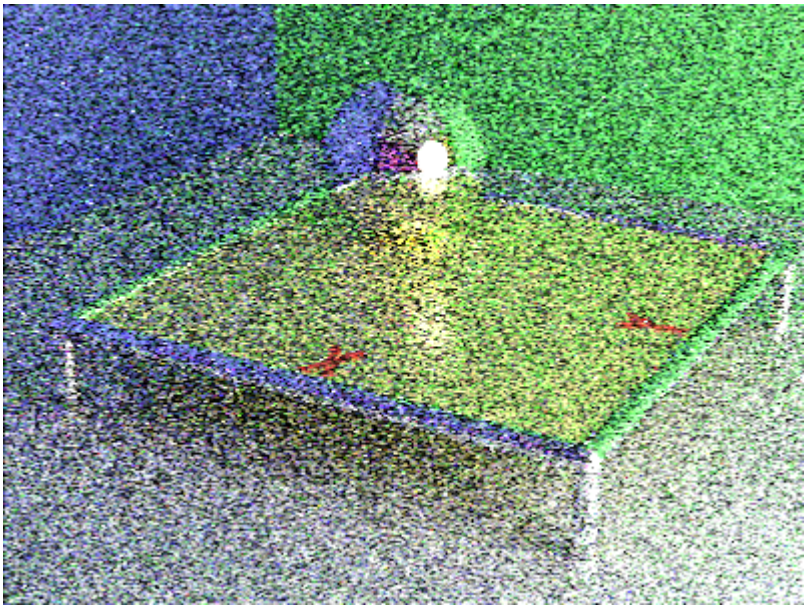
512 passes image with GGX model



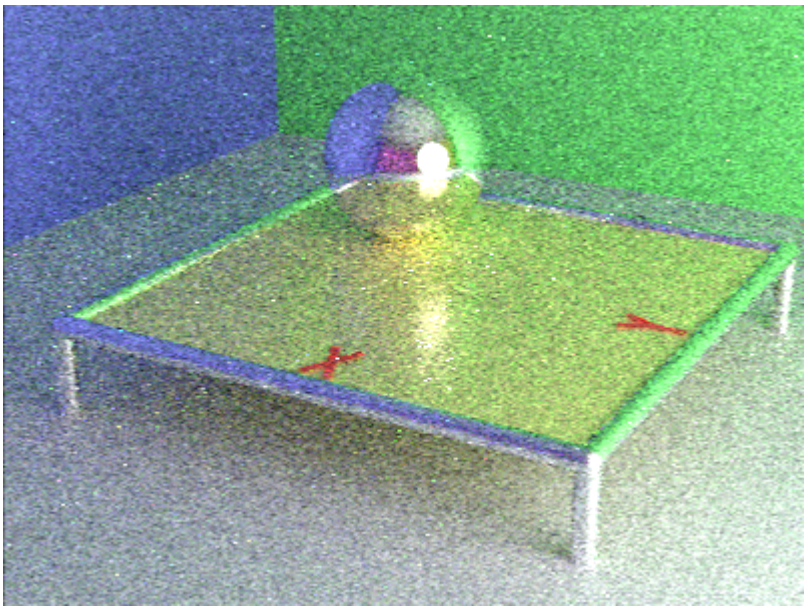
images with Beckman model

1 pass image with Beckman model

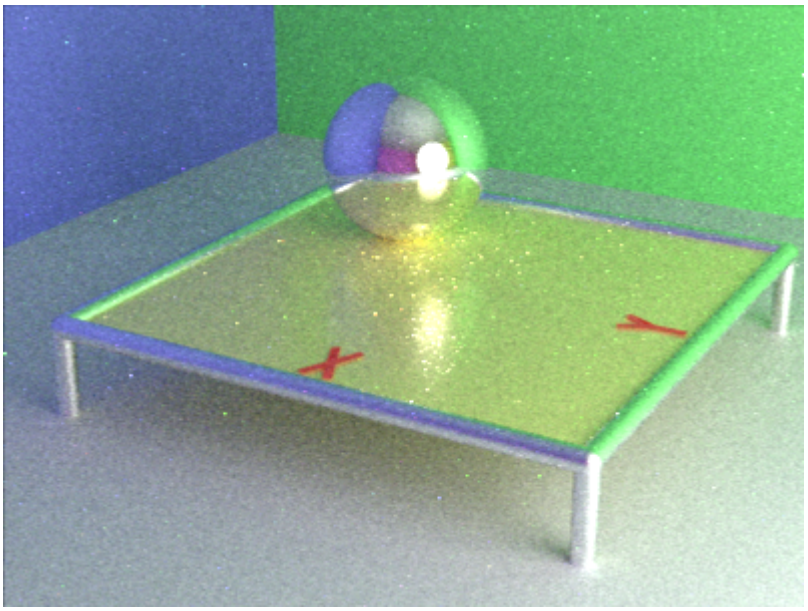
8 passes image with Beckman model



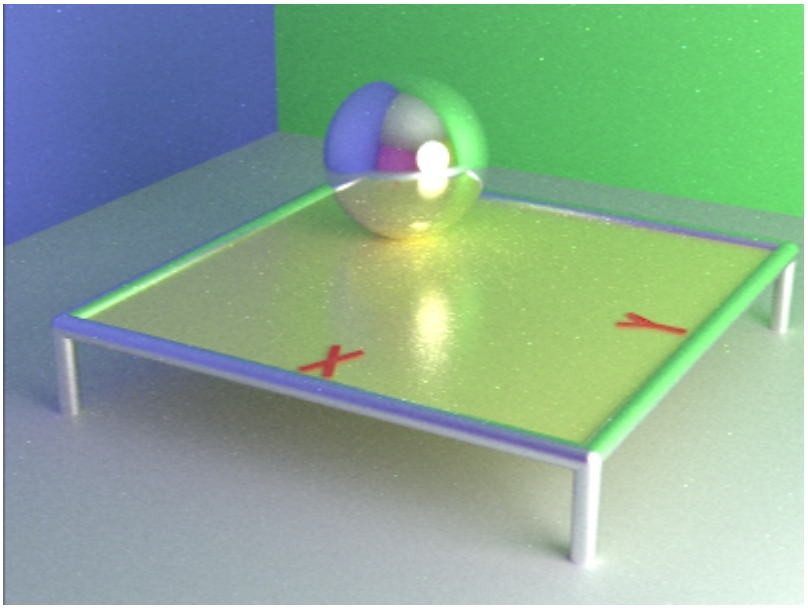
64 passes image with Beckman model



512 passes image with Beckman model



4096 passes image with Beckman model



Notes

BRDF

BRDF lighting equation is the following:

$$\frac{K_d}{\pi} + \frac{D(m)\cdot G(w_i,w_0,m)\cdot F(w_i\cdot m)}{4\cdot|w_i\cdot N|\cdot|w_0\cdot N|}$$

where, K_d is diffuse reflection color of surface and m = normalize($w_i + w_0$)

D factor

The term D is the normal distribution that can be calculated with

$$D_p(m) = \frac{a_p+2}{2\cdot\pi}(m\cdot N)^{a_p}$$

$$D_b(m) = \frac{1}{\pi a_b^{\,2}(N\cdot m)^4}e^{\frac{-tan^2\theta_m}{a_b^{\,2}}}$$

$$D_g(m) = \frac{a_g^{\,2}}{\pi(N\cdot m)^4\cdot(a_b^{\,2}+tan^2\theta_m)^2}$$

where D_p is the normal distribution for phong model

D_b is normal distribution for beckman model

D_g is the normal distribution for the GGX model

a_b and a_g is $\sqrt{\frac{2}{a_p+2}}$ with a_p is roughness value for phong model

$$tan\theta_m = \sqrt{1.0 - (m\cdot N)^2}/(m\cdot N)$$

the normal distribution will be 0 when $(m\cdot N)$goes 0

this is implemented in 501 lines in raytrace.cpp

F factor

The term F is the fresnel value that can be calculated with

$$F(d) = K_s + (1 - K_s)\cdot(1 - |d|)^5$$

the K_s is specular reflection color at L=V=N=H in this project, I used material specular color for this.

It is implemented on line 498 in raytrace.cpp

G factor

The term G is the geometry that can be calculated with

$$G(w_i, w_0, m) = G'(w_i, m) \cdot G'(w_0, m)$$

$$G_p'(w_i, m) = \frac{3.535a + 2.181a^2}{1 + 2.276a + 2.577a^2}$$ if a >= 1.6, the values will be 1

where $tan\theta_v = \sqrt{1.0 - (v \cdot N)^2}/(v \cdot N)$,

$$a = \sqrt{a_p/2 + 1}/tan\theta_v$$

$$G_b'(w_i, m) = \frac{3.535a + 2.181a^2}{1 + 2.276a + 2.577a^2}$$ if a >= 1.6, the values will be 1

where $a = 1/(a_b \cdot tan\theta_v)$

$$G_g'(w_i, m) = \frac{2}{1 + \sqrt{1 + a_g^2 + tan^2\theta_v}}$$

the geometry will be 0 when $(\frac{m \cdot v}{N \cdot v})$goes 0

this is implemented in 540 lines in raytrace.cpp

Reflection

The program will also calculate the reflection color on each object. I changed the random direction algorithm to make the program also shoots ray toward the reflective direction. The ray evaluates a new random value and determines whether it shoots toward a normal direction or reflective direction. The probability for diffuse ray selection and reflection ray selection is following:

diffuse selection = $\frac{|K_d|}{|K_D|+|K_s|}$ and reflection selection is 1 - diffuse selection.

It is implemented on line 416 in raytrace.cpp