Project 4: Transmission

Name: Minsuk Kim(m.kim)
Instructor: Dr. Gary Herron
Class: CS500
Semester: Spring 2022
Date: 3/27/2022

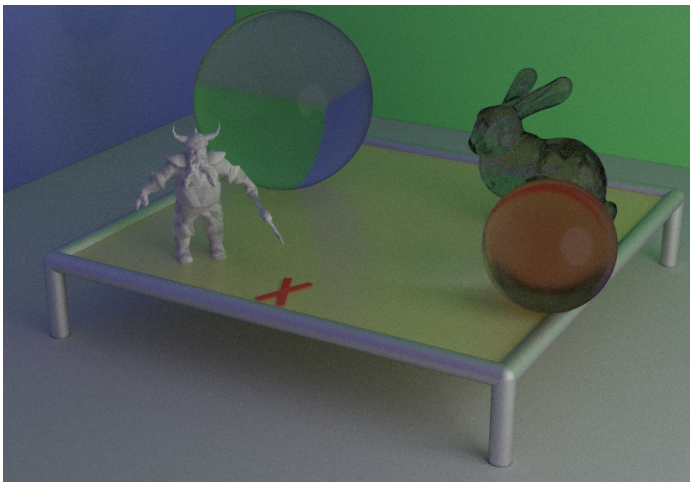


Table of Contents

Introduction

Overview

The main purpose of the project is to enhance the reflective micro-facet BRDF with the Snell's law, Beer's law and Multiple Importance Sampling. The program will calculate the lighting value on each path tracing loop to evaluate the reflection, refraction 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 BRDF. 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 rays 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 a distributed vector around a normal vector in the hit position. The program also uses the reflection feature that random direction is a distributed vector around a reflective vector. The program produces the random distributed refraction vector as well. 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 1 pass

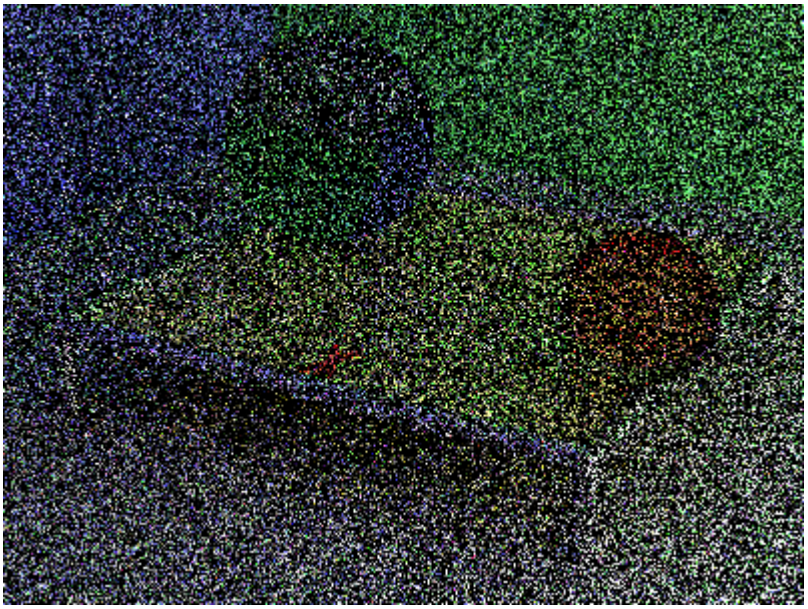


image with 8 passes

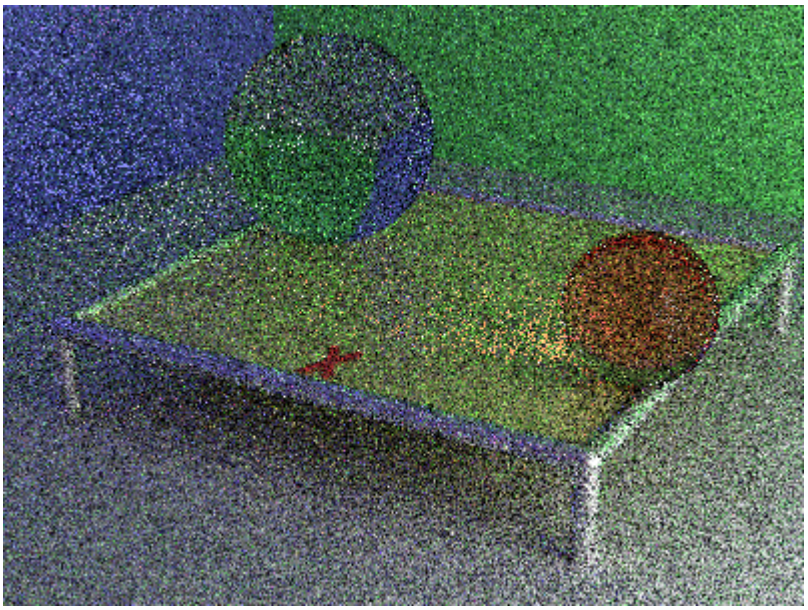


image with 64 passes

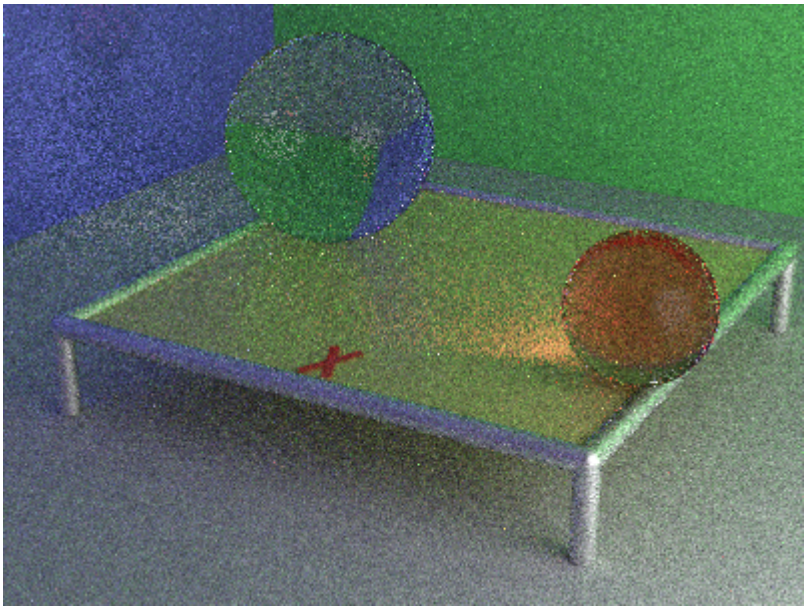


image with 512 passes

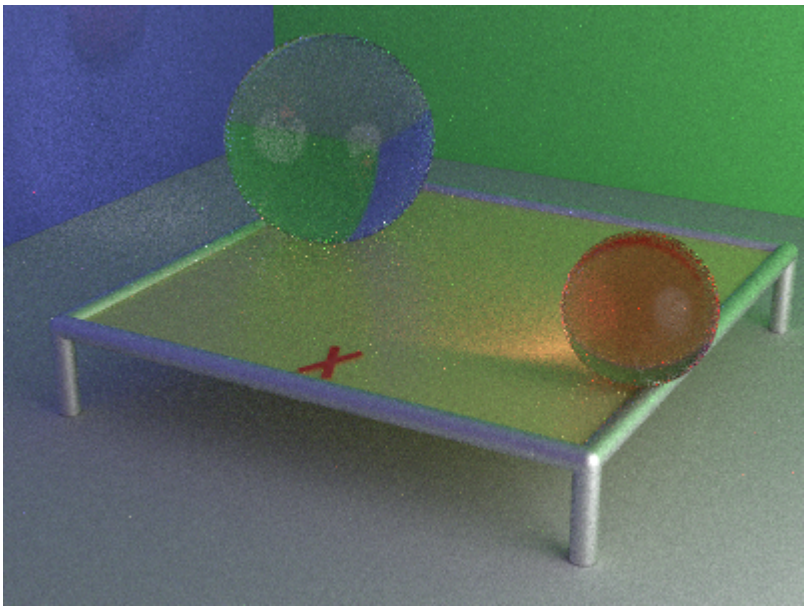
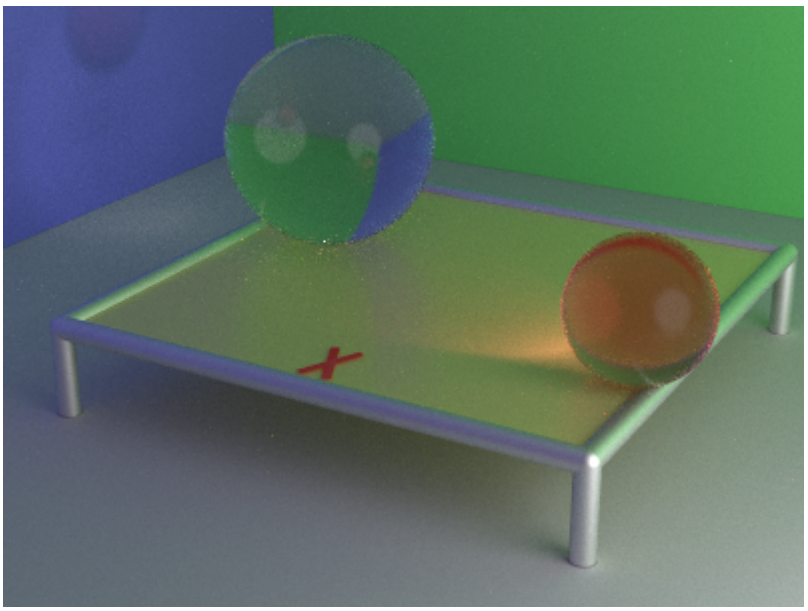


image with 4096 passes



Notes

Refraction

The ray will produce the refractive vector when it hits an object and generate the new ray. The program calculates the random value to determine whether to use diffuse or reflection or refraction. The probability for three selection is following: diffuse selection = $\frac{|K_d|}{|K_D|+|K_s|+|K_t|}$, reflection selection = $\frac{|K_d|}{|K_D|+|K_s|+|K_t|}$ and transmission(refractive vector) selection is 1 - diffuse selection - reflection selection. The vector will be calculated by :

$$w_i = (\eta(w_o \cdot m) - sign(w_o \cdot N) \cdot \sqrt{r})m - \eta \cdot w_o \text{, where}$$

$$r = 1 - \eta^2(1 - (w_o \cdot m)^2) \text{ and sign(x) is -1 or 1 that defines sign of x}$$

η is index of refraction.
It is implemented on line 482 in raytrace.cpp

In case of selecting the transmission, the probability for brdf value will be changed to

$$P = p_d P_d + p_r P_r + p_t P_t \text{, where}$$

$$p_t = \frac{|K_t|}{|K_D|+|K_s|+|K_t|},\ P_t = D(m)|m \cdot N|\frac{n_o^2|w_i \cdot m|}{(n_o(w_i \cdot m)+n_i(w_o \cdot m))^2} \text{, where}$$

$$m = -\ normalized(n_o w_i + n_i w_o)$$

The P_t will be P_r when $r = 1 - \eta^2(1 - (w_o \cdot m)^2)$ is negative.
It is implemented on line 531 in raytrace.cpp

Beer's law

When the case for transmission, the amount of light must be attenuated. The attenuation value will be calculated by :

$$A(t) = e^{tlogK_t} \text{ if } (w_o \cdot N) < 0 \text{ otherwise, A(t) = 1}$$

t is the distance of the light ray.
It is implemented on line 603 in raytrace.cpp

Transmission

The transmission value is calculated in lighting equation:

$$A\frac{D(m)\cdot G(w_i,w_0,m)\cdot(1-F(w_i\cdot m))}{4\cdot|w_i\cdot N|\cdot|w_0\cdot N|} \cdot \frac{|w_i\cdot N|\cdot|w_0\cdot N|\cdot n_o^2}{4\cdot|w_i\cdot N|\cdot|w_0\cdot N|} \text{, where}$$

A is attenuation value from Beer's law and m = -normalize($n_o w_i + n_i w_0$)

The transmission value will be same with reflective value when $r = 1 - \eta^2(1 - (w_o \cdot m)^2)$ is negative.
It is implemented on line 641 in raytrace.cpp

MIS

When the program adds light to final pixel, the program needs to adjust the light by a weight called MIS(the multiple importance sampling).

For the explicit light, the MIS will be calculated with $\frac{p*p}{p*p+q*q}$, where p is probability for light selection / angular measure between current object and selected light. q is probability for brdf selection * Russian Roulette.

For the implicit light, the MIS will be calculated with $\frac{p*p}{p*p+q*q}$, where p is probability for brdf selection * Russian Roulette and q is probability for hitted light / angular measure between current object and hitted light.

image with no MIS adjust

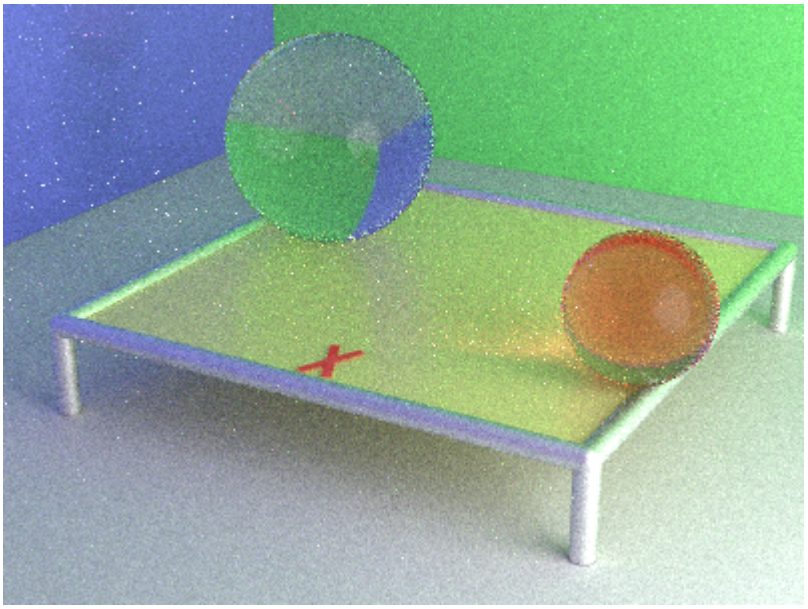
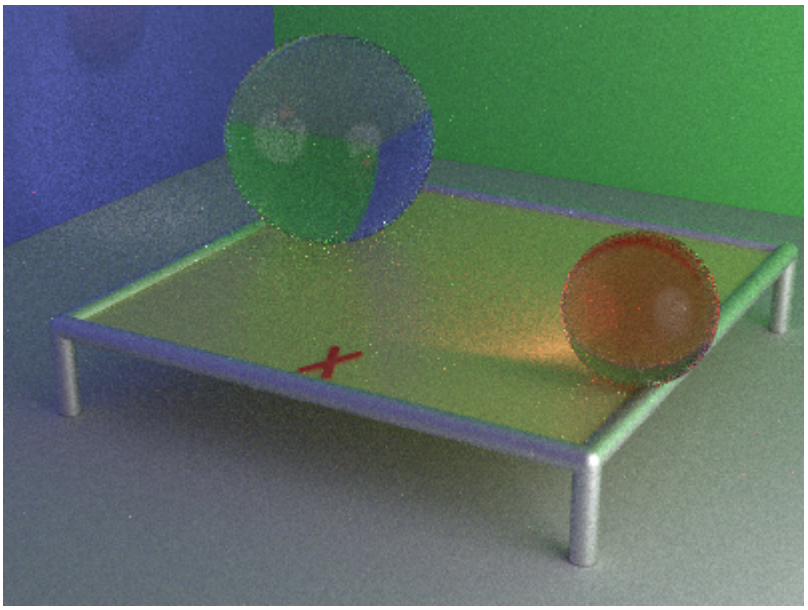


image with MIS adjust



It is implemented on line 299, 325 in raytrace.cpp