

Assignment 4:

NAME: XIAOHAN WU

STUDENT NUMBER: 2019533093

EMAIL: WUXH@SHANGHAITECH.EDU.CN

1 INTRODUCTION

In HW4, we are required to implement Monte-Carlo (single-directional) path tracing for global illumination in the ray-tracing framework. To perform a ray-mesh intersection, we are also required to build a k-d tree as the acceleration structure. The program contains 3 basic parts and 2 optional parts. The three parts are: 1.Implement ray-mesh intersection test with a k-d tree for acceleration. 2.Implement the ideal diffusion BRDF with an area light source. 3.Implement the Monte-Carlo path tracing algorithm for global illumination: direct + indirect lighting. The optional parts are: 1.Implement the ideal specular and glossy specular BRDF with multiple importance sampling. 2.Implement the translucent BRDF with refraction, e.g. glasses and diamonds.

2 IMPLEMENTATION DETAILS

In this section, I will introduce how I implement my program for each part respectively.

1.The realization of K-D Tree is relatively easy after being given the pseudocode shown in Tutorial 7. Here,in addition to changing the judgement of depth being even or not to the judgement of the value of depth mod 3(to evaluate which axis to be devided), I also make a change to choose the median value of the axis instead of the center of the median-triangles along the axis that is devided,which results a relative higher performance.

Algorithm BUILDKDTREE($P, depth$)

```
1. if  $P$  contains only one point
2.   then return a leaf storing this point
3. else if  $depth$  is even
4.   then Split  $P$  with a vertical line  $\ell$  through the
        median  $x$ -coordinate into  $P_1$  (left of or
        on  $\ell$ ) and  $P_2$  (right of  $\ell$ )
5.   else Split  $P$  with a horizontal line  $\ell$  through
        the median  $y$ -coordinate into  $P_1$  (below
        or on  $\ell$ ) and  $P_2$  (above  $\ell$ )
6.    $v_{\text{left}} \leftarrow \text{BUILDKDTREE}(P_1, depth + 1)$ 
7.    $v_{\text{right}} \leftarrow \text{BUILDKDTREE}(P_2, depth + 1)$ 
8.   Create a node  $v$  storing  $\ell$ , make  $v_{\text{left}}$  the left
        child of  $v$ , and make  $v_{\text{right}}$  the right child of  $v$ .
9. return  $v$ 
```

Fig. 1. KD-Tree pseudocode

2.After implementing KD-tree, here we are going to implement

the ideal diffusion BRDF(isotropic diffusion (Lambertian) BRDF), which is different from the Phong model used in assignment 3. I use Mally's method here to realize the cosine-weighted hemisphere sampling and calculate the corresponding pdf.

3.After implementing the ideal diffusion BRDF,we need to sample positions on the light surface for area light.Every time, I sample a random position from the uniform distributionas the default shape of the area light is a rectangle.

4.Then,we can start to render realistic images using Monte-Carlo Path Tracing.For the calculation of each pixel value, I sample 1024 rays with depth 4, and use the iterative implementation of path tracing shown as below.

Beta = 1, L = 0

Generate a ray from the camera to a pixel on the image plane

For i = 1 to max depth

- (0) Suppose the marching ray hits at $P(i)$
- (1) $L += \text{Beta} * \text{Direct lighting at } P(i)$
- (2) Sample ONE next ray according to $P(i)$'s BRDF and find the corresponding pdf
- (3) $\text{Beta} *= \text{BRDF} * \cos\theta / \text{pdf}$
- (4) Spawn and march the new ray

Fig. 2. Monte-Carlo Path Tracing

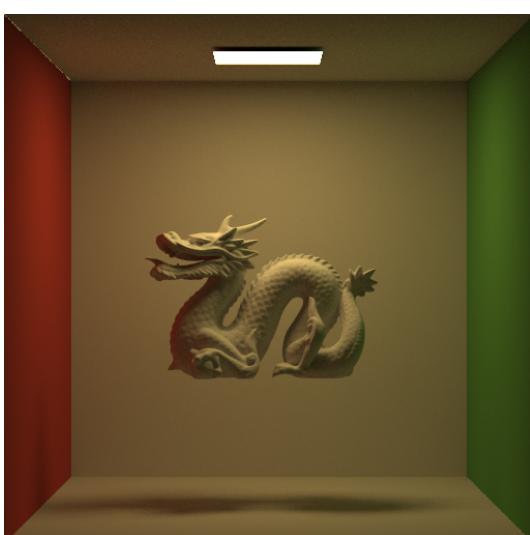
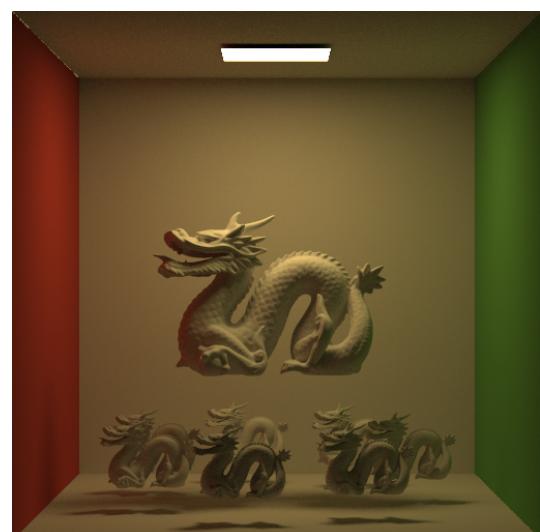
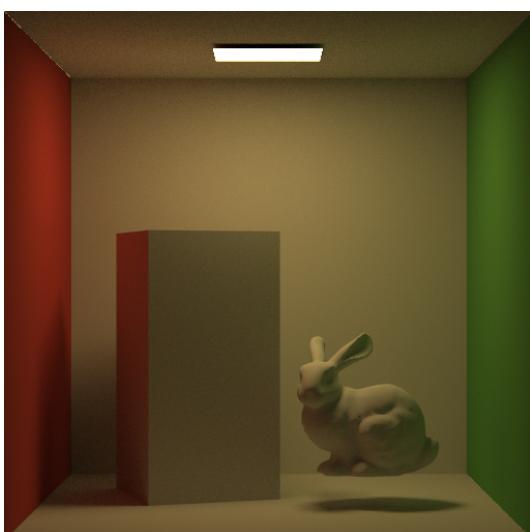
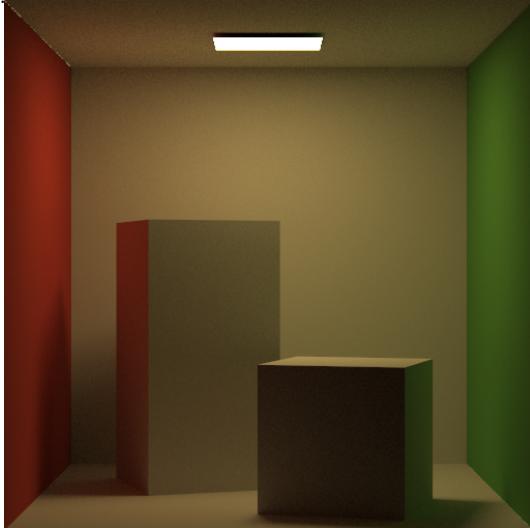
5.For the ideal specular part,it's relatively easy to calculate the out direction of the light given the surface normal and the in direction. For the glossy one, according to PBRT, as the definition of glossy is neither specular nor diffusion, so I implement it in the following way:1.Calculating the the ray direction of the ideal specular surface 2. Using cos-weighted sampling to sample around that direction.

6.To realize the ideal transmission, we need only to calculate the out direction of the ray using Snell's law. One thing that is worth being mentioned is that as the ray will not only refract but also reflect in the reality, so I also add the ideal specular into ideal transmission part to make it seem more real.As we can see in the result part, adding the relectivity will make the short box more translucent compared to the original one.(e.g.the bottom of two boxes.)

3 RESULTS

Below are the results of my programming work.

1:2 • Name: Xiaohan Wu
student number: 2019533093
email: wuxh@shanghaitech.edu.cn



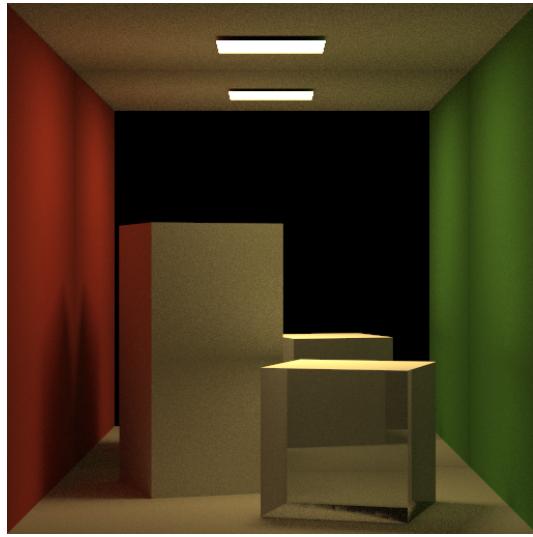


Fig. 3. SpecularWall

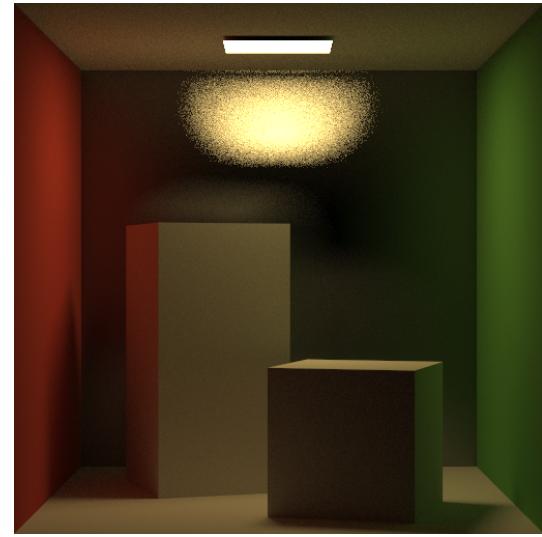


Fig. 5. glossywall

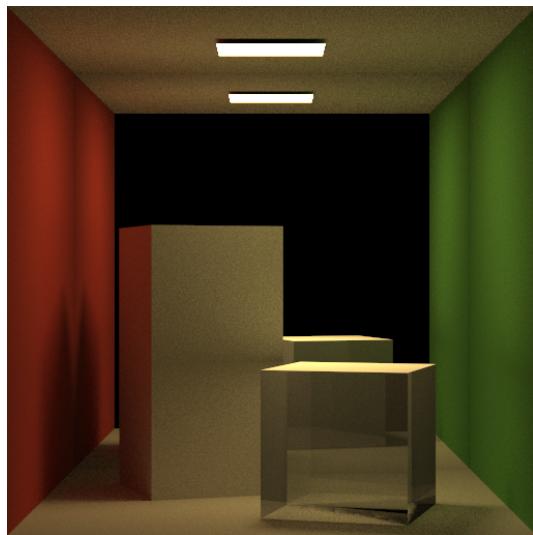


Fig. 4. SpecularWall with reflectivity

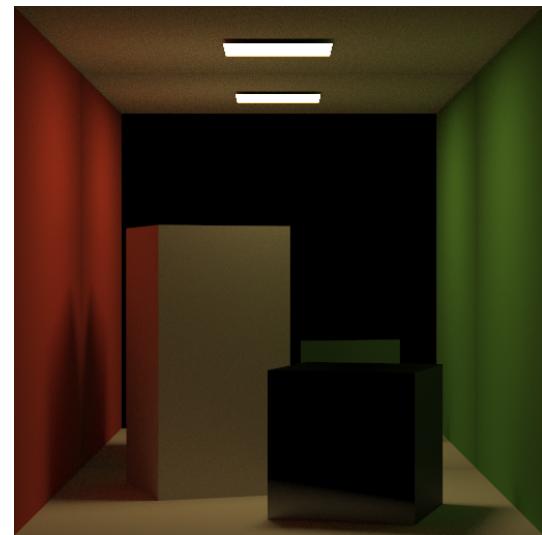


Fig. 6. glossybox

1:4 • Name: Xiaohan Wu
student number: 2019533093
email: wuxh@shanghaitech.edu.cn

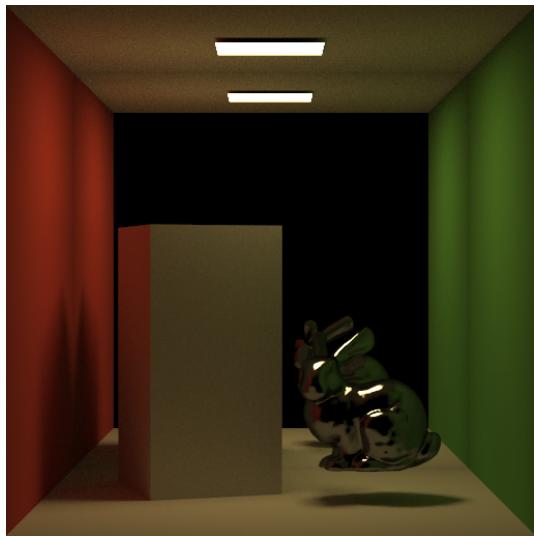


Fig. 7. glossybunny