Advanced Computer Graphics

HW4 Report

410685020 何冠勳

1. Method

When it comes to accelerating the ray tracing algorithms, the overall concept is to reduce the number of iterations or loops and all the unnecessary parts that might not be visible. The following is a pseudo code of the algorithm along with the parts that cause iterations or recursion highlighted in yellow.

|  |
| --- |
| **<Ray Tracing>**  *For every pixel*  *Construct a ray from the eye to current pixel*  *For every object in the scene*  *Find intersection with ray*  *Keep the closest object*  *Shade the object by*  *color = ambient*  *cast shadow ray to see whether shadowed*  *if mirror*  *color += color\_reflect + trace reflection*  *if transparent*  *color += color\_refract + trace refraction* |

As we can see, there are two loops in the algorithm. My intuition is to reduce the second for loop by a more efficient “checking intersection” mechanism. I utilize the bounding box for objects to check that “If the ray hit the bounding box then it has an opportunity to hit objects inside, otherwise it’ll never hit the object”. The algorithm now is described as the following pseudo code.

|  |
| --- |
| **<Bounding Box Acceleration>**  *For every pixel*  *Construct a ray from the eye*  *For every bounding box*  *If the ray hit the bounding box*  *For every object in the bounding box*  *Find intersection with ray*  *Keep if closest*  *Shade the object by*  *color = ambient*  *cast shadow ray to see whether shadowed*  *if mirror*  *color += color\_reflect + trace reflection*  *if transparent*  *color += color\_refract + trace refraction* |

This simple checking intersection mechanism can benefit all the blue-highlighted pieces as well. However, the bounding box can be divided into multiple boxes and execute the same tracing procedure recursively. As a consequent, how to segment the objects or space arises as another significant topics of further acceleration. In my code, I use equally segmentation and bounding boxed that are axis-aligned just for POV.

1. Speed Results

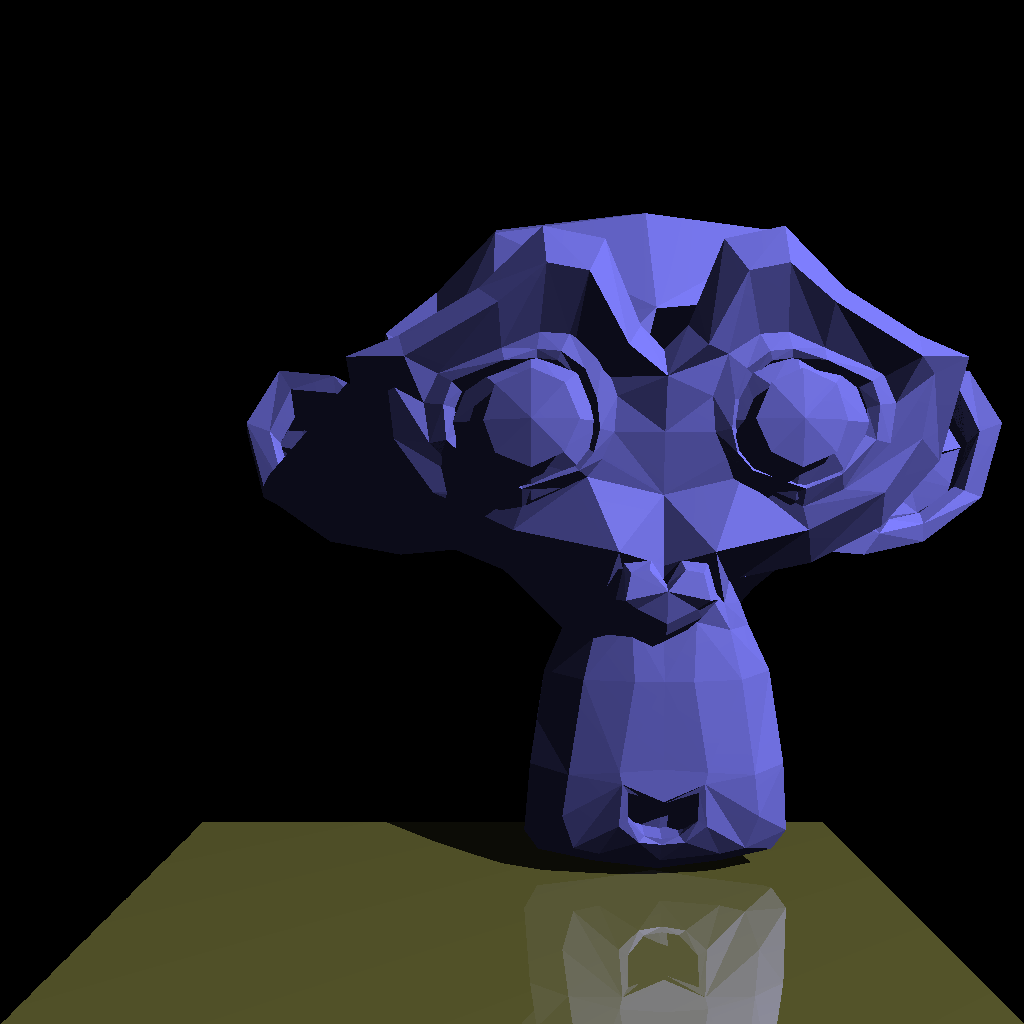
The following table is the speed test results with different input text for rendering. “HR” stands for high resolution, namely 1024 x 1024. We can see that with bounding box the rendering is indeed faster, and as the object gets more complicated, the acceleration becomes more apparent.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Input\_Suzanne | Input\_Suzanne-HR | Input\_Bunny | Input\_Bunny-HR |
| No bounding (s) | 15 | 246 | 2010 | 32837 |
| The best acceleration with bounding (s) | 13 | 209 | 1644 | 25024 |
| Segment size | 220 | 200 | 8000 | 12800 |
| Improvement (%) | 13.3% | 18.7% | 18.2% | 23.8% |

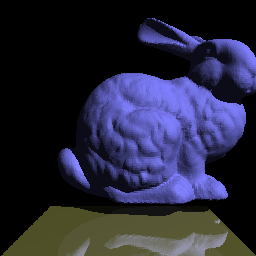
1. Rendering Results



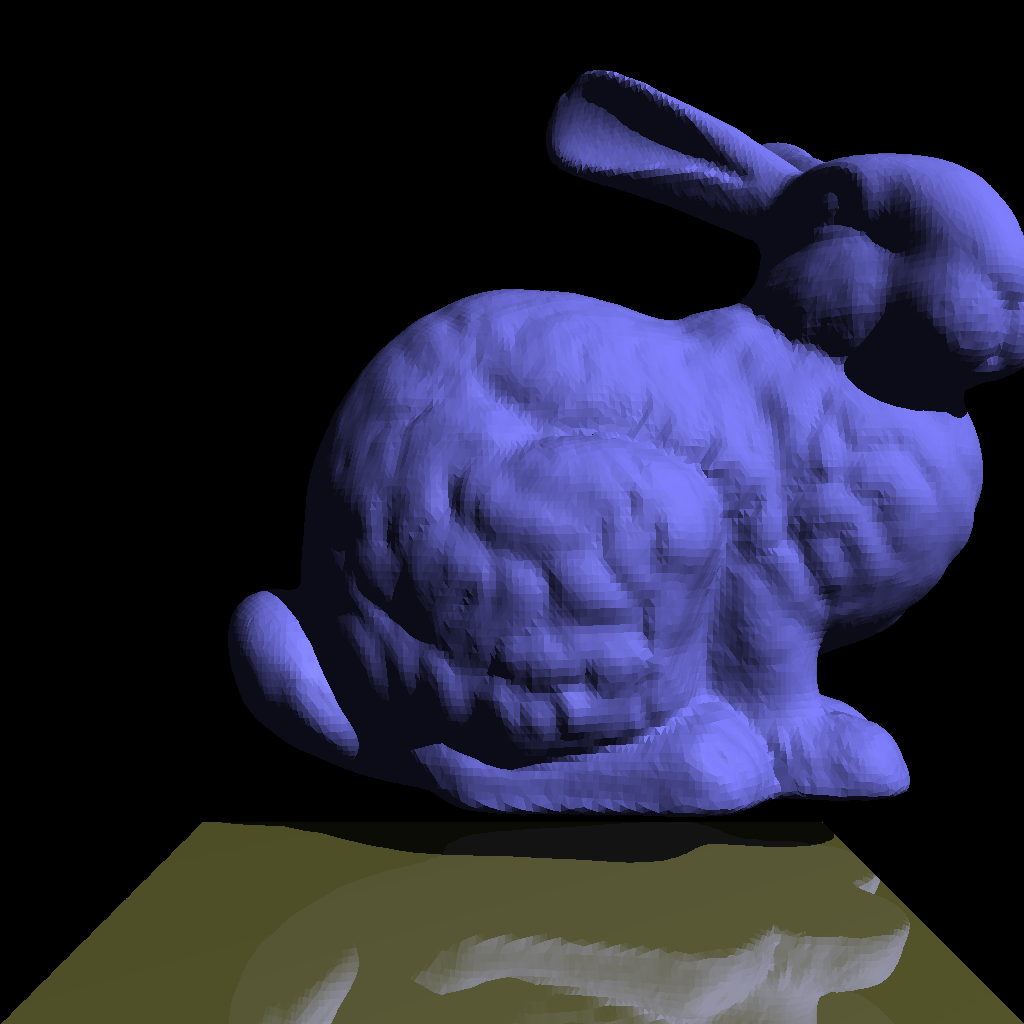
( Input\_Suzanne.ppm )



( Input\_Suzanne-HR.ppm )



( Input\_Bunny.ppm )



( Input\_Bunny-HR.ppm )