**Rendering HW1 Report**

**Heightfield-ray Intersection Algorithm**

Use 2D DDA to find the route of ray in heightfield, then walk through the heightfield according to the route.

First, find the start gird of heightfield, then walk through grids according to the direction of ray. In each grid the ray walk through, split the grid into top-left triangle and bottom-right triangle, then test the intersection of two triangles. If any triangle in grid is intersected with ray, the intersection function can return true. Otherwise, return false, there are no any intersections in the heightfield.

About the triangle intersection test, I have tried two ways. First, create the triangle mesh of grid(contains two triangles), then refined it and call the intersection method of each shape in the list return by refine function of triangle mesh. Second, borrow the intersection method of triangle class and modify it so that it don’t need to keep the information of the triangle mesh it belongs to. But this way should store all points in heightfield in the heightfield creation stage for convenience.

**Phong Interpolation Algorithm**

Calculate the barycentric coordinate of the intersection point in the triangle. Then use the coordinate to interpolate the normals of three points of triangle to calculate the normal of intersection point.

In the heightfield creation stage, calculate and store the normals of each point in heightfield. In the GetShadingGeometry method, calculate the intersection point and three vertices of triangle, then calculate the barycenteric coordinate. At the end, calculate the normal of intersection point and use it to calculate other corresponding information.

**Performance Comparison**

Figure 1

Render time(sec)

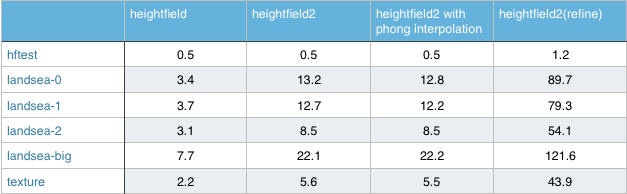


Figure 2

Execution time(sec)

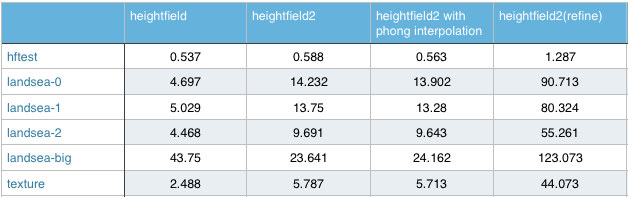
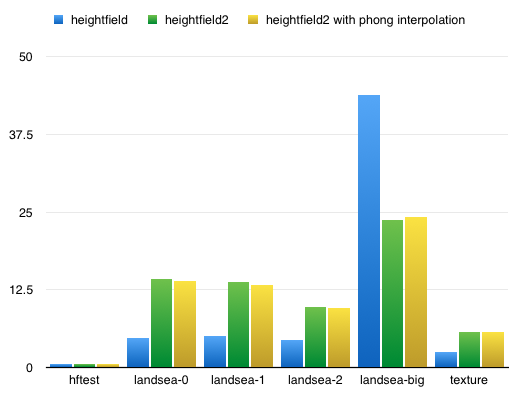


Figure 3

Execution time bar chart without heightfield2 refine



According to three figures above, I found that the performance of default heightfield and my heightfield are almost the same in hftest. I guess the scene is too simple so that there is no difference between them. But I found that my heightfield spends double to triple time than default heightfield in landsea 0-2 and texture. My heightfield only performs better than default in landsea big. At the beginning, I thought that calculating Phong Interpolation spends too much time. Thus, I recorded the time of my heightfield without Phong Interpolation and found that the performance is the same. Therefore, if the scene is not complicated to some degree, my heightfield won’t outperform the default one. Take landsea big as example, my heightfield spent around half time than default heightfield. But we can still find a thing from Figure1 & Figure2. The acceleration of default heightfield is very good. Compare the render time with my implementation, the render time of my heightfield is triple of the default one. This proves that the acceleration reduces many non-intersected object than mine implementation when rendering. If we can reduce the set up time of the acceleration, default heightfield could be more powerful.

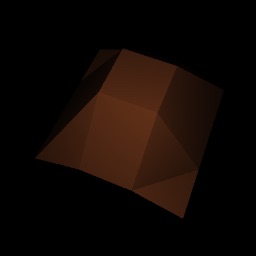
Another interesting thing is that the version using triangle mesh refine performs very bad. The time which refine version spends is the same as doing the triangle intersection test directly of all girds in the route without breaking the DDA loop. But my refine version indeed breaks the DDA loop when the ray hits something. Thus, my refine version will get correct results but the other version gets wrong results. I think the creation of triangle mesh and refine are time consuming so that the refine version is very slow.

**Environment**

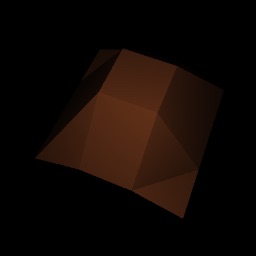
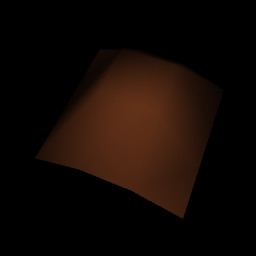
Mac OS X Yosemite 10.10.5, 2.7GHz Intel Core i5, 8G Ram, Intel Iris Graphics 6100, build pbrt project by Xcode

**Result**

hftest:

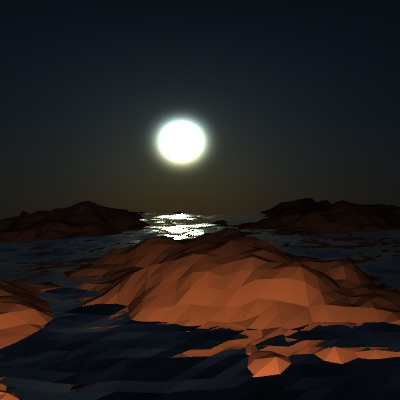
****

heightfield1

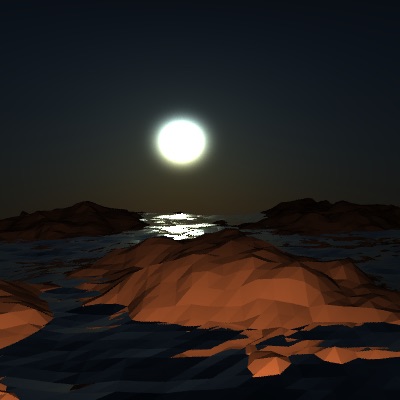
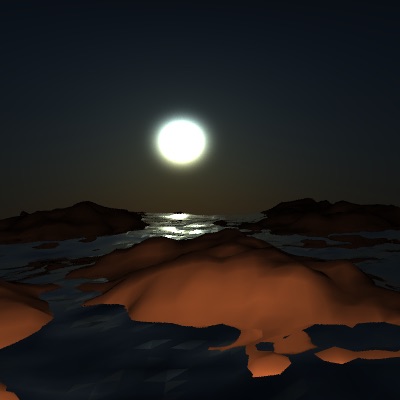
**** ****

heightfield2 heightfield2 with interpolation

landsea-0:

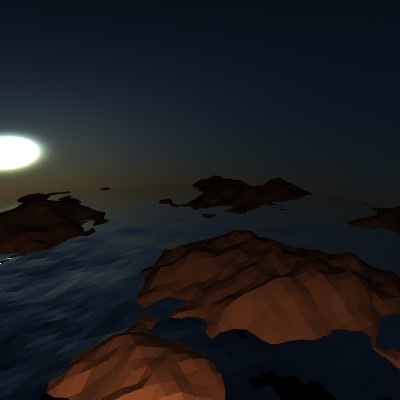
****

heightfield1

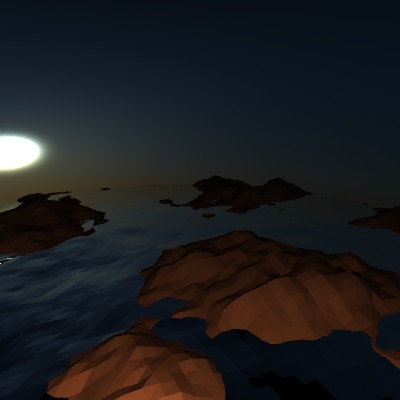
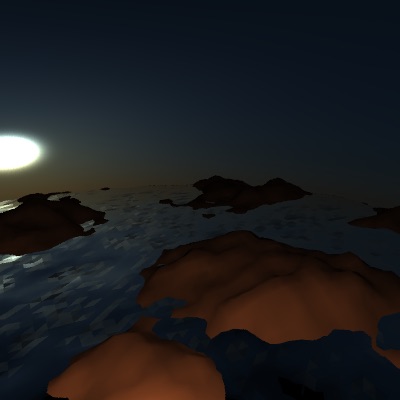
** **

heightfield2 heightfield2 with interpolation

landsea-1:

****

heightfield1

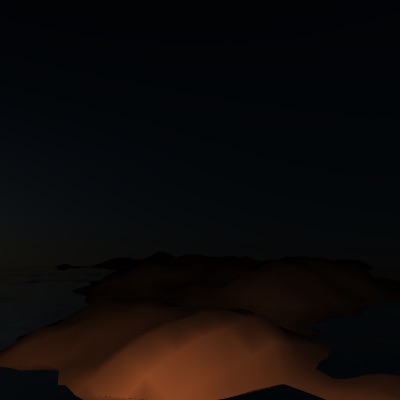
** **

heightfield2 heightfield2 with interpolation

landsea-2:

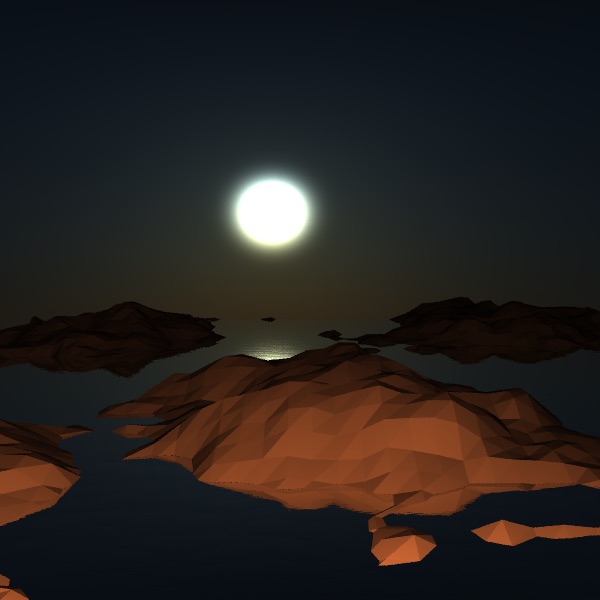
****

heightfield1

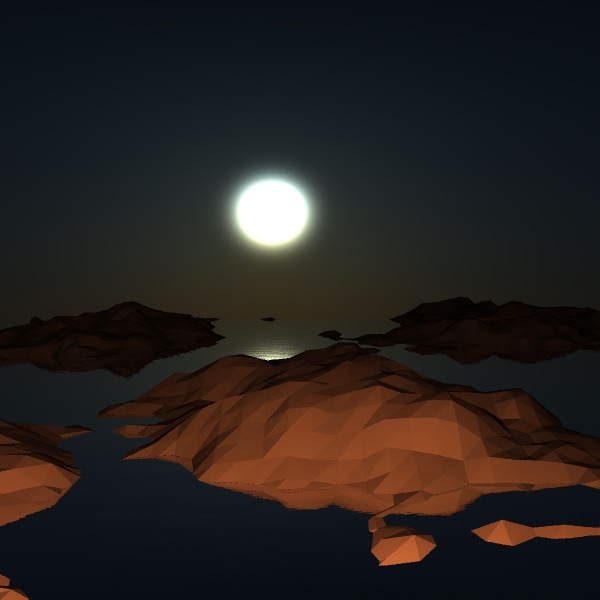
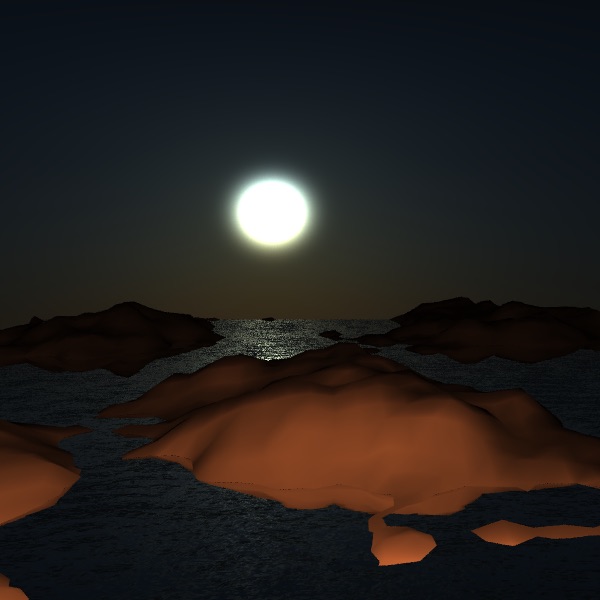
**** ****

heightfield2 heightfield2 with interpolation

landsea-big:

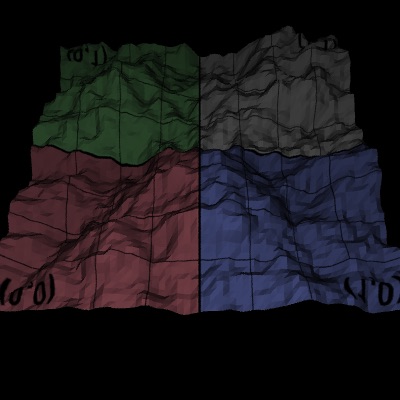


heightfield1

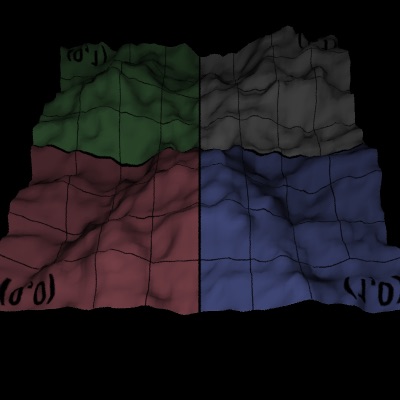
 

heightfield2 heightfield2 with interpolation

texture:

****

heightfield1

**** ****

heightfield2 heightfield2 with interpolation