

# Z 缓冲器算法程序说明

## 1 编程环境

VS2010+OpenGL

## 2 用户界面

程序启动时会自动扫描程序根目录下的 data 目录（**如果想测试其他的 obj 文件，请将其拷贝至该目录下**），找到所有 obj 文件后会提示用户输入要显示的 obj 文件编号。输入正确编号后，程序就会渲染出结果，并同时显示当前进度，面片数，顶点数和渲染时间。另外程序现在只支持窗口大小的改变，暂不支持用鼠标旋转模型。

## 3 数据结构

所有数据结构都在 PolygonAndEdgeTableStructure.h 中定义并添加了注释，基本与老师的 PPT 上的定义相同。

另外整个程序的算法也基本按照老师 PPT 上的算法流程，只有一点不一样。即（参考网上资料）把活化边表拆分开来，活化边表只存储一半的边，并且对应的活化边表是记录在相对应的活化多边形下。这样通过当前活化边表对于 x 的方向排序，每两个活化边表中边就形成一个边对。这样实现代码稍微简洁一些。

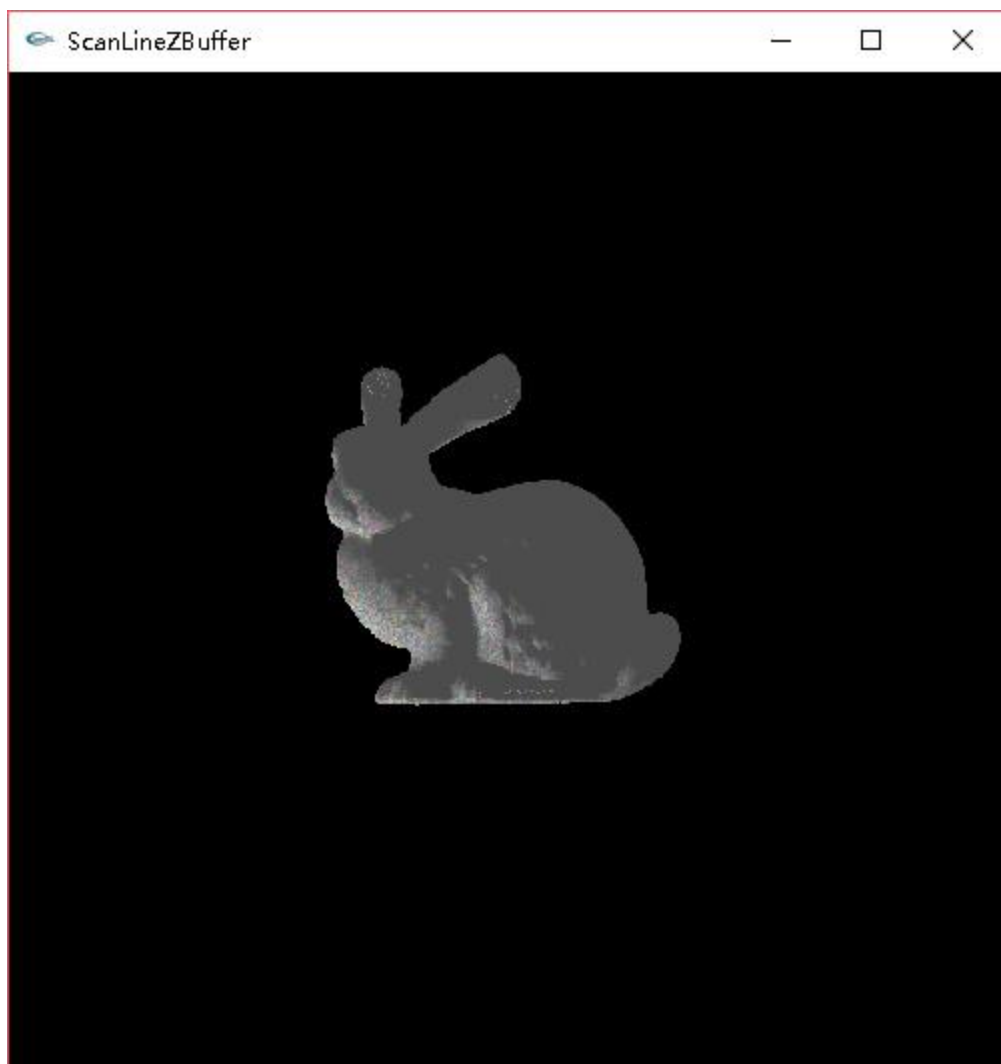
## 4 加速结构

利用 OpenMP 对相关 for 循环进行了加速。经过测试，结果表明，与不加速对比，速度要稍微快一些。

## 5 测试结果

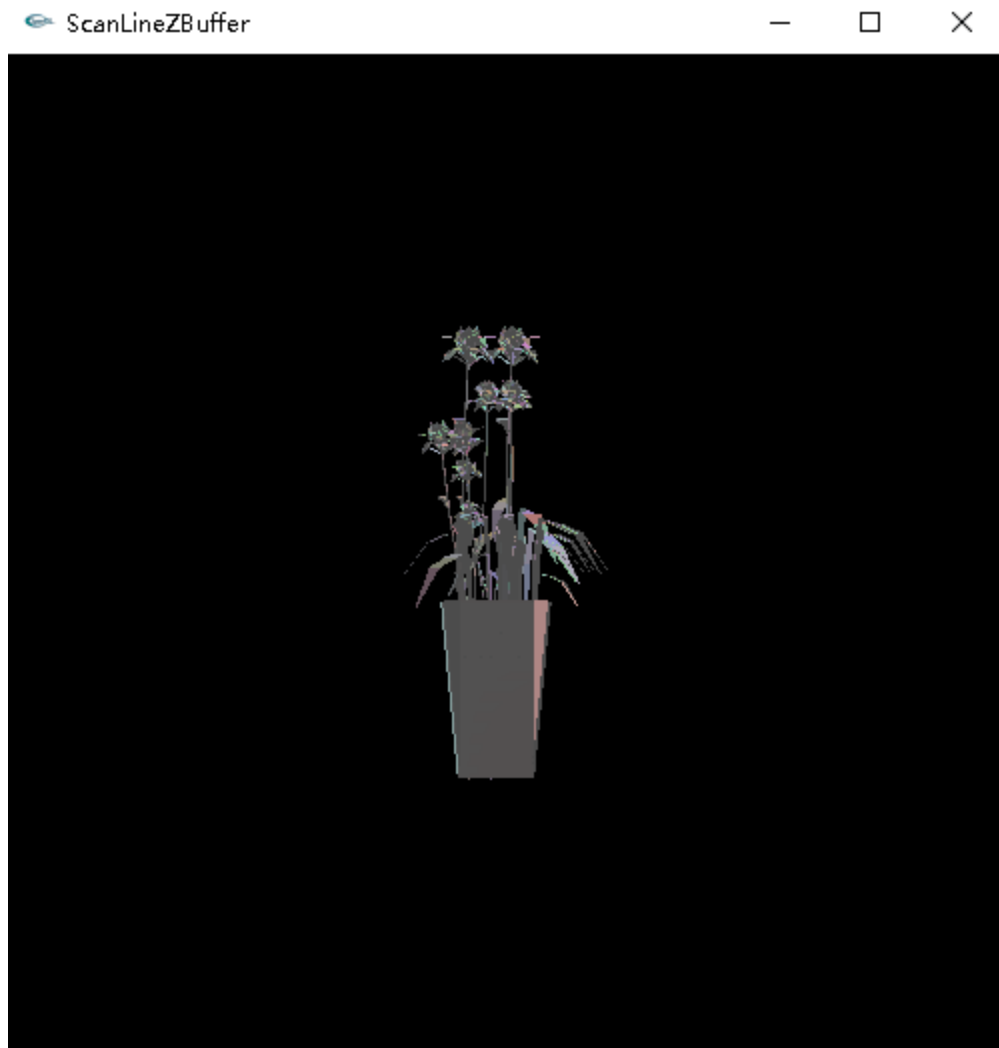
测试环境：win10 64bit + i5-3230M CPU(2.60GHZ) + 4G 内存

窗口分辨率：500\*500



```
loading obj file start...  
loading obj file end,which spent 0.873590 secs in total  
rendering start...  
progress:100%  
vertex num:34834,face num:69451  
rendering end,which spent 0.567375 secs in total
```

测试 1: bunny.obj



```
loading obj file start...
loading obj file end,which spent 0.383252 secs in total
rendering start...
progress:100%
vertex num:7608,face num:4061
rendering end,which spent 0.092060 secs in total
```

测试 2: flowers.obj

## 6 参考资料

1. <https://github.com/tiny656/scan-line-zbuffer-algorithm>
2. <https://github.com/knightliao/zbuffer>