

## ZBuffer 程序说明

编程环境:

OS: Win7 64 位系统

IDE: Microsoft visual studio 2010

Library: Qt4.8.0 boost\_1\_47\_0

用户界面说明:

- 1) 点击 File 菜单, 选择 obj 文件导入, 选择文件导入程序。
- 2) 或者拖动 obj 文件到程序主窗口中也可以显示 obj 文件。
- 3) 如果在外部编辑器中改动了读入的 obj 文件, 按<F5>可以刷新当前导入的 obj 文件。
- 4) 支持鼠标拖动进行旋转查看, 水平方向移动物体绕 y 轴旋转, 垂直方向移动物体绕 x 轴旋转。

数据结构说明:

有关于 ZBuffer 的数据结构都定义在 datastructure.h 文件中

主要包括:

- 1) PageTable: 页表, vector<pageList>类型, 其中 pageList 是 page 的一个 list, 将相同最高 y 值的 page 放入到相同的 pageList 中, 同时最高的 y 值作为 pageList 在 pageTable 中的位置坐标。
- 2) ActivePageList: 活化页表, list<Page>类型, 保存所有在扫描线中的页面。
- 3) EdgeTable: 边表, vector<PageEdgeMap>类型, PageEdgeMap 类型为 map<int, list<Edge>> 的一个映射表。索引的 key 为 Page 的 Id 数值, 将同一个 Page 的边都放在 map 的 second 元素中。同时, PageEdgeMap 的元素在 vector 的位置表示当前的所有边的 y 的最大值。建立这样子的结构, 在构造的时候有点慢, 但在后面扫描线运行时候, 可以根据快速地根据索引计算出当前运行页的所有边。
- 4) ActiveEdgeList: 活化边表, 保存了正在运行的边对。
- 5) GLModel: 保存了从 obj 文件中读取出的所有顶点, 法向数据等
- 6) ZBuffer: 读入 GLModel 数据, 同时建立对应的边表和页表。

加速说明:

利用 openMP 并行计算加速建立表的过程。

从 obj 文件读入顶点之后, 要将所有的顶点乘以旋转平移矩阵, 然后去建立对应的边表和页表。这个过程可以并行的计算。在对所有的三角形进行旋转矩阵计算的循环中, 利用 openMP 的编译器并行运行指令, 将循环分割到所有的 cpu 核心中去计算, 可以提高运行速度。

其它说明:

显示的图形是一个灰度图像, 颜色越白表示离视点越近, 其 z 值越大, 越黑则越远。