

# CG Project: Z buffer

---

杜宇 11121043 [answeror@gmail.com](mailto:answeror@gmail.com)

## Brief

渲染器仿 OpenGL 接口, 光栅化实现了 zbuffer, 可以读取只含几何信息的 OBJ 文件, 显示为深度图. Bunny(69451 个片面)在 800x600 的分辨率下可以达到 10FPS.

Github: <https://github.com/Answeror/cg>

## Environment

- Intel Core2 Quad Q6600 @ 2.40GHz
- 8GB RAM
- Nvidia GeForce 8800 GT
- Win7 x64
- VS2010 SP1
- Boost 1.4.7 (header only)
- Qt 4.7.2 (OpenGL disabled)
- CMake 2.8.7

## Source tree structure

- /bin: 可执行文件
- /docs: 所有文档
- /build.zbuffer: 构建目录
- /misc
  - models: 模型文件
  - 3rd: 第三方库和我自己的工具库(编译所需的 boost 和 qt 没有包括在内)
    - acmake: CMake 辅助函数
    - ans: 自己的常用工具库
    - cml(第三方): 数学库
    - cmlex: 我对 cml 的简单扩展
    - OvenToBoost(第三方): boost.range 的扩展
- /src/zbuffer: 源代码和单元测试
  - Include: 头文件(相应的原文件在同结构的 src 目录下)
  - widgets: Qt 控件

- aint: 鼠标交互类
- core: 核心算法
  - concepts: 泛型结构 concept
  - camera.hpp: 视图矩阵和投影矩阵的封装
  - gpu.hpp: CUDA 加速的矩阵变换, 因为效果不好, 不用了
  - log.hpp: 时间统计
  - matrix\_stack.hpp: 模拟 opengl 的矩阵堆栈
  - planf.hpp: 三角形平面方程
  - rasterize.hpp: zbuffer 算法接口
  - rasterize\_impl.hpp: zbuffer 算法实现
  - render\_operation.hpp: 对应于 OpenGL 的 begin/end 块的类型(GUARDS, TRIANGLES...)
  - renderer.hpp: 渲染器接口(仿照 OpenGL 的接口)
  - renderer\_impl.hpp: 渲染器实现, 包括变换, 消隐和光栅化, 没有剪裁
  - wavefront\_obj\_loader.hpp: OBJ 文件加载器(只能加载几何信息)

## Parallel Accelerations

以下加速环境均为:

- bunny(69451 个片面)
- 800x600 分辨率

加速:

- 视图, 投影, 视口变换用 PPL 加速: 20ms -> 8ms

## User Interface

程序显示的是深度图, 越白越近, 初始界面是 12 个面的方块. 左侧面板显示了程序启动以来的每帧渲染时间分布. 可以用鼠标旋转相机, 左右绕世界坐标系 y 轴旋转; 上下绕惯性坐标系 x 轴旋转. 效果如图 1.

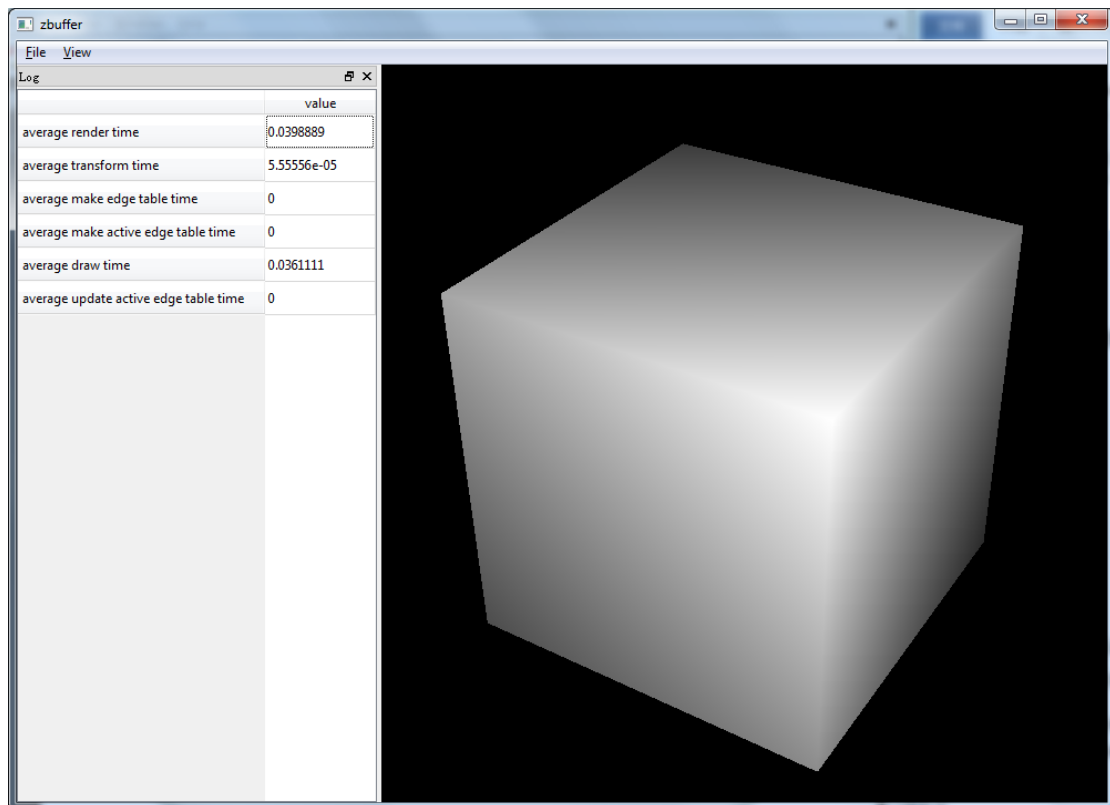


图 1: 初始界面

Log 各项参数意义:

- Average render time: 每帧渲染时间
- Average transform time: 视图, 投影, 视口变换总时间
- Average make edge table time: 构造边表的时间
- Average make active edge table time: 构造活化边表的时间
- Average draw time: 扫描时间
- Average update active edge table time: 更新活化边表的时间

点击 File->Open 加载 OBJ 文件(只能包含几何信息), 如图 2.

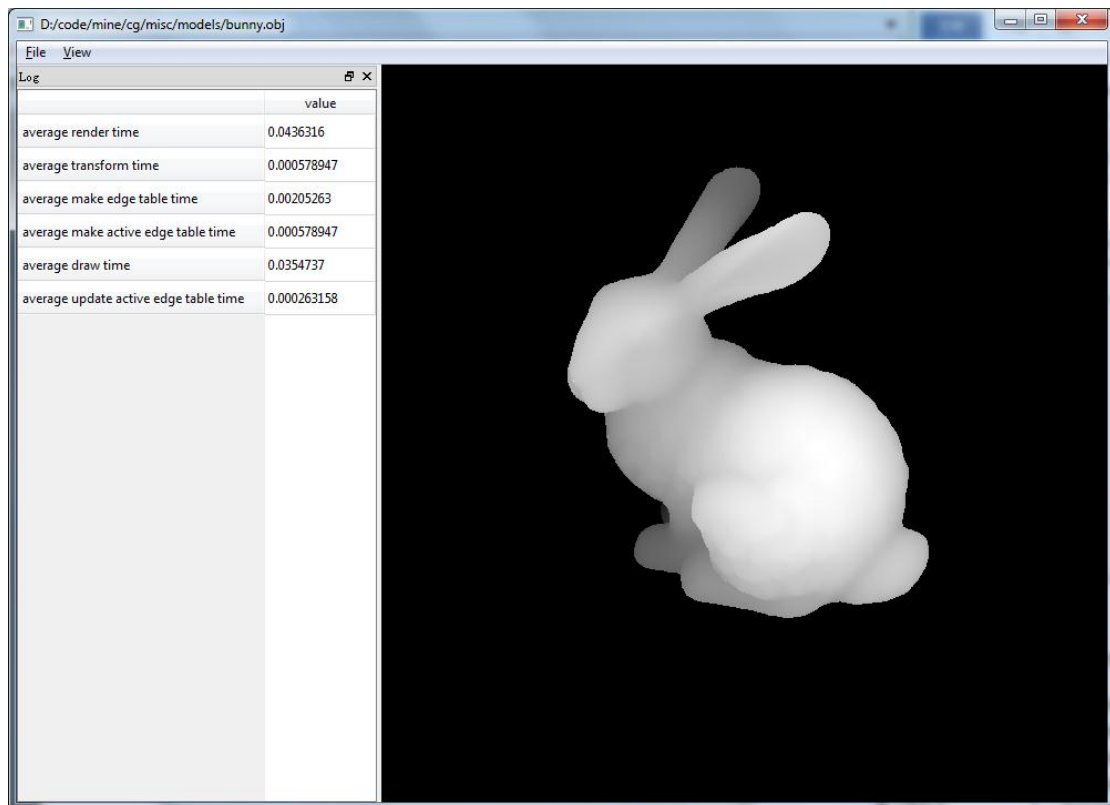


图 2: 加载 OBJ 文件

点击 View->Rotate 自动绕世界坐标系 y 轴旋转.