

# 基于 Bézier 曲线的三维造型与渲染

计 55 席照炜 学号：2015011354

## 一、项目名称：基于 Bézier 曲线的三维造型与渲染

## 二、文件组织

1. \*.h, \*.cpp: 源代码文件。
2. Makefile: makefile 文件，用于编译工程。
3. main: 可执行文件。
4. m1~m5.png: png 文件，渲染时用到的纹理文件（有一些是之前用过的，效果图中没有再用）。
5. object44.obj: obj 文件，贝塞尔曲面转网格的 obj 文件，可以打开查看贝塞尔曲面的形状。
6. scene.txt: txt 文件，场景输入文件，用于构建场景（贝塞尔曲面的坐标控制点写在了程序中）。
7. picture.png: png 文件，最后渲染生成的图片。
8. readme.txt: txt 文件，使用说明。
9. 设计文档: PDF 文件，即本文件，设计文档。

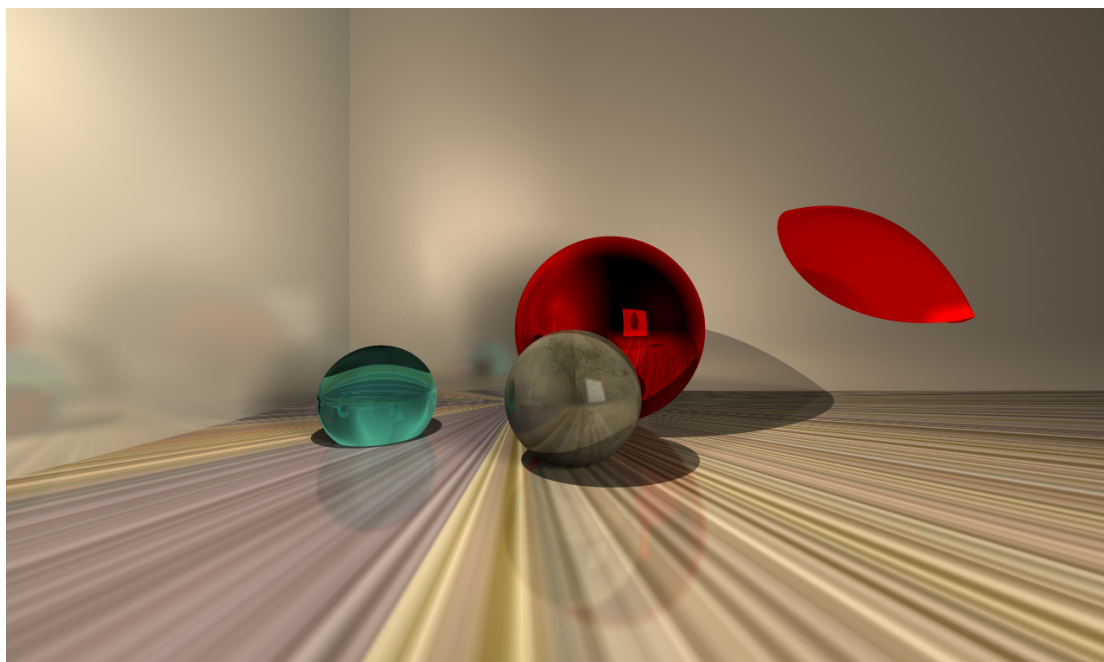
## 三、使用说明

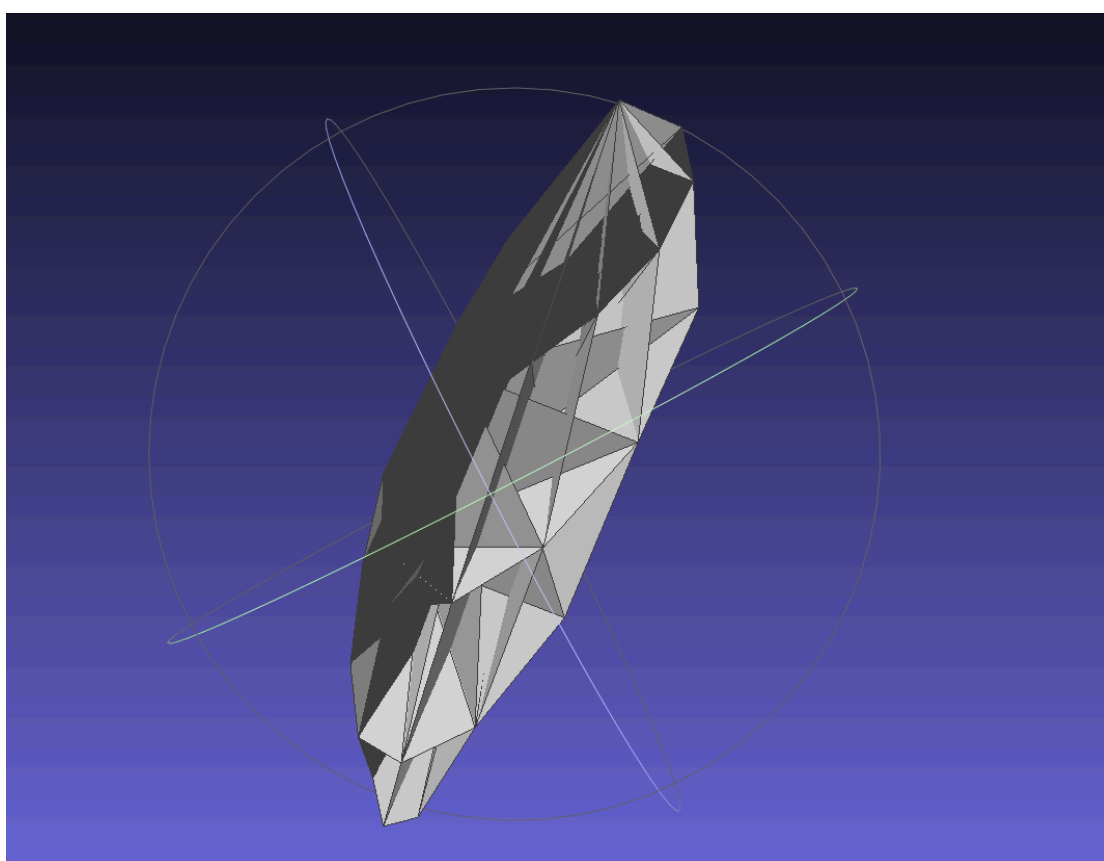
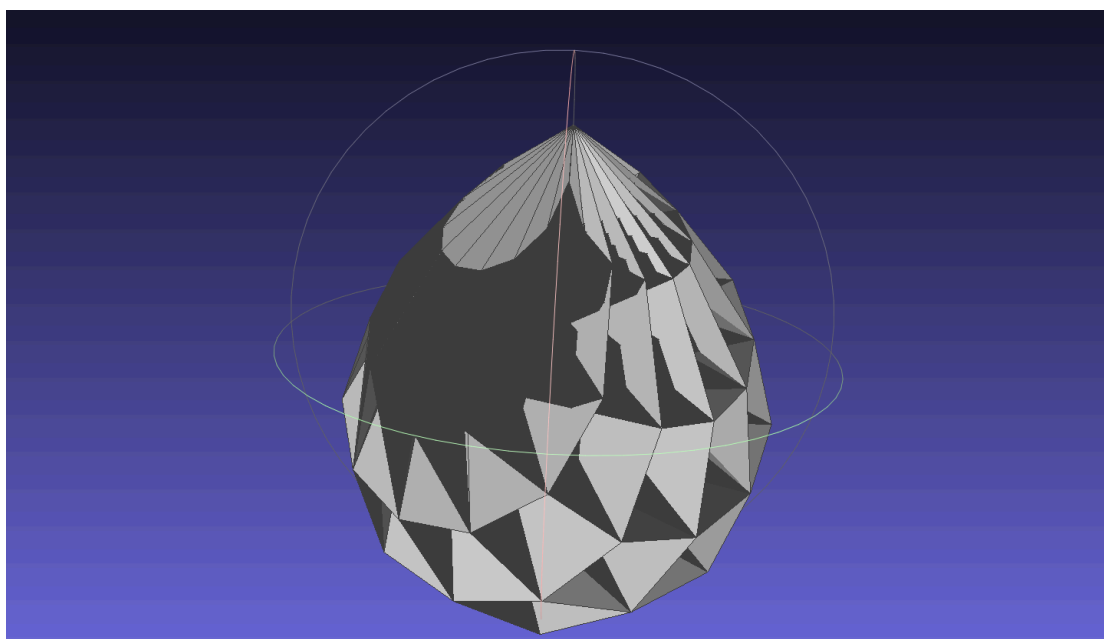
详见 readme.txt，程序使用了计算库 Eigen，请确保电脑正确安装了 Eigen。

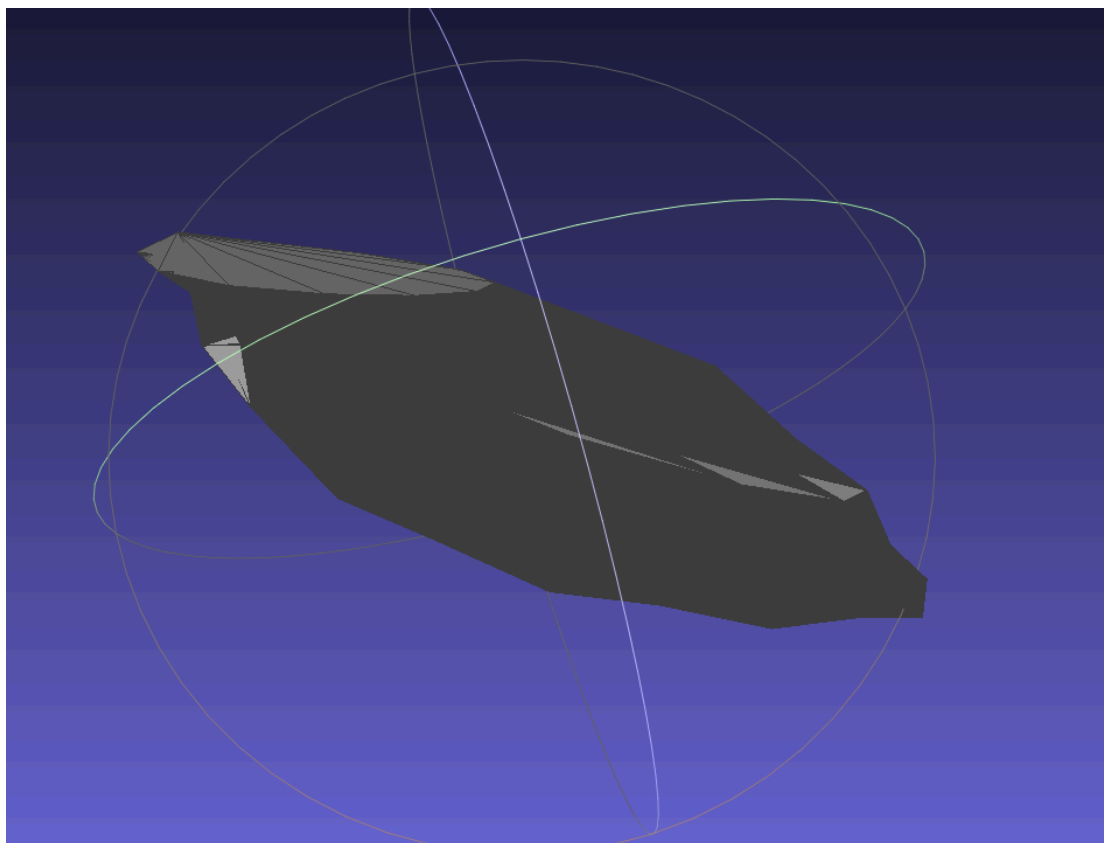
## 四、实现功能

- (1) 4\*4 控制点的双三贝塞尔曲面绘制和求交。(object.cpp, Bezier 部分)
- (2) 光线追踪方法渲染, 并通过设计表面属性混合(反射、折射、漫反射、镜面漫反射) 来实现漫反射等效果, 最后颜色等于各种属性的条件期望。  
(raytracer.cpp)
- (3) AABB 包围盒曲面求交加速。(求交见 AABB.cpp, 应用在 object.cpp 的 Bezier 部分)
- (4) 超采样抗锯齿。(raytracer.cpp, render() 中)
- (5) 点光源和面光源。(light.cpp)
- (6) 平面和球的纹理贴图。(参见 object.cpp 中的调用和 png.cpp 中的颜色计算)

## 五、最终成果图及贝塞尔曲面的 obj 网格图







## 六、算法说明

### 1. 光线跟踪

光线跟踪采用基本的光线跟踪方法，不过表面属性可以混合，通过条件期望来复合各种效果，漫反射不同于 path tracing 中的随机方向继续追踪，而是以一个漫反射系数的方式得到结果。

### 2. 曲面求交

对于球面和平面求交按照书上和老师课上教的方法即可正确求交，对于贝塞尔曲面的求交利用牛顿迭代法，但是在曲面较大的时候牛顿迭代法收敛非常慢，而且容易因为有多解而无法收敛到正确结果，非常取决于初始值的选择，即使引入随机采样，在严重增加时间成本的基础上效果依然不尽如人意。

所以采用的算法是对于贝塞尔曲面进行细分，将曲面四分（注意指的是参数

意义上的四分，即  $u, v$  的各自二分，而不是坐标意义上的四分），这样不断细分，在每一个很小的曲面上，牛顿迭代的速度和效果都有提高。

为了减少计算量，使用了包围盒加速，这里用的是 MAX-min 表示的 AABB 包围盒，然后利用 PPT 中提到的 Slabs 算法求交。这样可以有效减少时间成本，因为需要求交的曲面数量大大减少。

## 七、程序架构说明

程序采用面向对象的编程思想，对物品、光线、相机都建立了对象，保留了复用性和可拓展性。

### 1. 物品 (Object) 类

class Material: 材质类，定义物品的材质。

class Crash: 交点类，记录光线与物品的交点。

基类: class Object

子类: 平面 (Plane), 球 (Sphere), 贝塞尔曲面 (Bezier)

### 2. 灯光 (Light) 类

基类 class Light

子类点光源 (PointLight), 面光源 (AreaLight)

### 3. 相机 (Camera) 类

### 4. 场景 (Scene) 类

### 5. 光线追踪引擎 (Raytracer) 类

## 八、感想与收获

通过这次的基于 Bézier 曲线的三维造型与渲染的大作业，我接触到了图形学中非常基本也非常重要的光线追踪技术，在这个过程中了解到平时我们看到的动画和游戏中美轮美奂的特效是如何产生的，这让人非常兴奋。同时对于贝塞尔曲线的绘制和求交考验了我探索问题，应用课内所学解决问题的能力，让我受益颇多，最后要感谢老师和助教的传授和讲义，我学到了很多。