# 基于Bezier曲线的三维造型与渲染

2015011313 徐鉴劲　计５４

## 项目功能和思想简述

1. 光线追踪

对于每个像素点，从镜头到画面像素位置发射一条光线, 查找到光线与物体或光源的最近的碰撞点；如果物体具有漫反射性质，则使用 Phong 模型进行局部光照模型计算；如果物体能够反射，则求出反射和折射光线后继续追踪视线。

在项目中的类Raytracer中进行了实现。本项目在此处使用了OpenMP进行了多线程加速。

1. 光子映射

光子映射从光源和视点两方面同时开始追踪。　先从光源发射大量光子（本项目中１e7的个数较为合理）；光子与物体相交后，有概率被漫反射、反射、折射或吸收。在光子到达漫反射表面时，会被存下来，构成光子图，即每一个光子的位置。

最后还要进行光线跟踪，但不同之处在于，视点与漫反射物体表面相交时，使用光子图计算光亮度来代替 Phong 的局部光照模型。具体的实现方法还有直方图统计和核函数法，但前者容易不平滑，后者容易产生统计偏差，最优办法是Ｋ近邻法：以碰撞点为球心,找出最小的包围 k个光子的球，根据光子能量、物体BDRF、查找半径和总光子数计算得到该点的光亮度。

由于光子十分多而且是分布在空间中的，可以使用KD树进行加速。

在项目中的PhotonMapper中进行了实现。其中PhotonMap类是光子图，包含了加入光子、建立KD树和查找k近邻光子的算法。PhotonTracer实现了发射光子并追踪光子的算法。

本项目在此处也使用了OpenMP进行加速，通过多个线程同时发射光子就可以完成并行。

1. 渐进式光子映射

由于光子映射只有一次渲染，使得算法对光子数量要求非常高。如果光子数量不足，就会出现大量黑色的雾。

渐进式光子映射倒转了光子映射的流程。首先进行光线跟踪，但在视线到达漫反射表面时将交点存下来，构成碰撞图。碰撞图的组织方式和光子图相同。然后再发射光子，算法考虑更新碰撞点的光流量；此时，算法查找该光子附近的所有碰撞点，分别更新其光通量。每轮光子发射完成后，每个碰撞点会根据该轮到达的光子数，更新下一轮的查找半径。最后,算法遍历所有的碰撞点，用它们的光通量和查找半径计算像素点的光亮度。

渐进式光子映射在类PPM中得到了实现。其中HitPointMap完成了碰撞点的加入、建树、更新。PPM首先调用RayTracer的追踪函数进行初始碰撞点生成，然后调用PhotonTracer类的光子发射、追踪函数进行光子位置的获取和碰撞点的更新。

1. 三次Bezier曲线造型

四个点可以完成３次Bezier曲线的的建模。通过曲线的拼接可以完成复杂曲线的建模。限制就是相邻曲线的点必须相同，而且为了保证连续，一阶导数需要连续，即两条曲线相邻的三个点需要共线。

通过对曲线进行旋转，可以完成旋转体的建模。

Bezier曲线方程的形成在BezierCurve类中得到了实现；旋转体物体在RotationBody中得到了实现，并且在该类中实现了网格化。

1. 基于六次方程求解的三次Bezier求交

对于一条三次Bezier曲线，他的x分量和y分量都是参量t的三次函数；将其绕ｚ轴旋转，旋转面的x,y,z分量可以用转动角theta和t共同确定。而视线的x,y,z分别都是参量u的函数。联立这两个方程组，得到theta、t、ｕ的三个方程组。其中消去theta最为简单，这样问题就变成了u和t的方程组。此时可以应用二维牛顿迭代进行求解，但是由于牛顿法对初始值的敏感性，求解十分困难。

参考[1]·[2]中求解近似求解六次方程的方法，本项目将u、t的方程组进一步消元得到一元六次方程组，再利用Sturm's theorem进行求解。该方法的大致思路是构造一个多项式除法余数序列（即Sturm序列），通过判断端点值使序列正负号变化的个数，判断不等根的个数。然后将区间细分，继续计算区间内不等根的个数，如果等于一就使用牛顿法求出，如果大于一就继续迭代。

多项式在Polynomial类中进行了实现，并在其中findAllRoots方法中进行了实现。

1. 景深效果

景深效果指的是实际人们看物体时有一个焦平面，光线汇聚到焦平面上形成清晰像。在本项目中，通过光线的随机发射和最终结果的收敛来模拟景深效果。

实现在RayTracer中\_DOFTracing中。

1. 软阴影效果

对光源光线起点采样，求出每条阴影测试线与物体相交的次数,再除以总的采样次数就得到阴影的强度。本项目中只实现了矩形光源的软阴影效果。

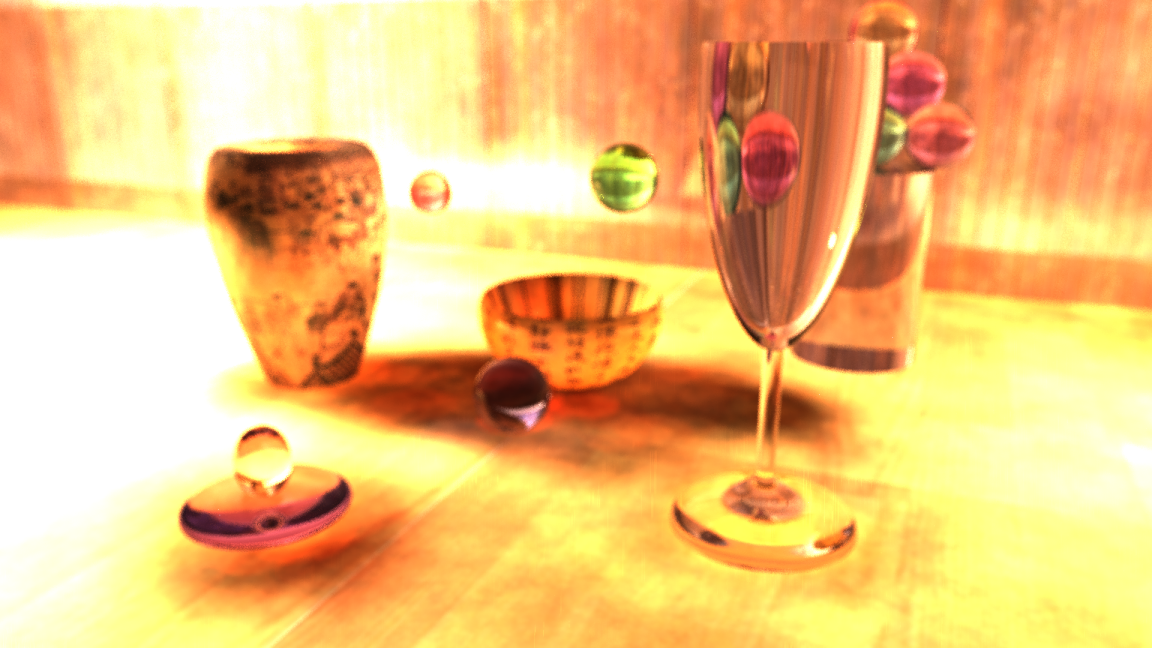
1. 纹理映射

对于平面,可指定纹理图片左上角，映射到平面上的什么位置，以及长宽在平面中的方向；对于球体,可指定球的轴方向以及纹理图片左边映射到球面的角度。对于圆柱体,可指定纹理的开始处映射到圆柱面的角度;对于Bezier曲线旋转体,可指定纹理左边映射到旋转体面的极角,以及每一旋转曲面在纹理图片中占的比例。

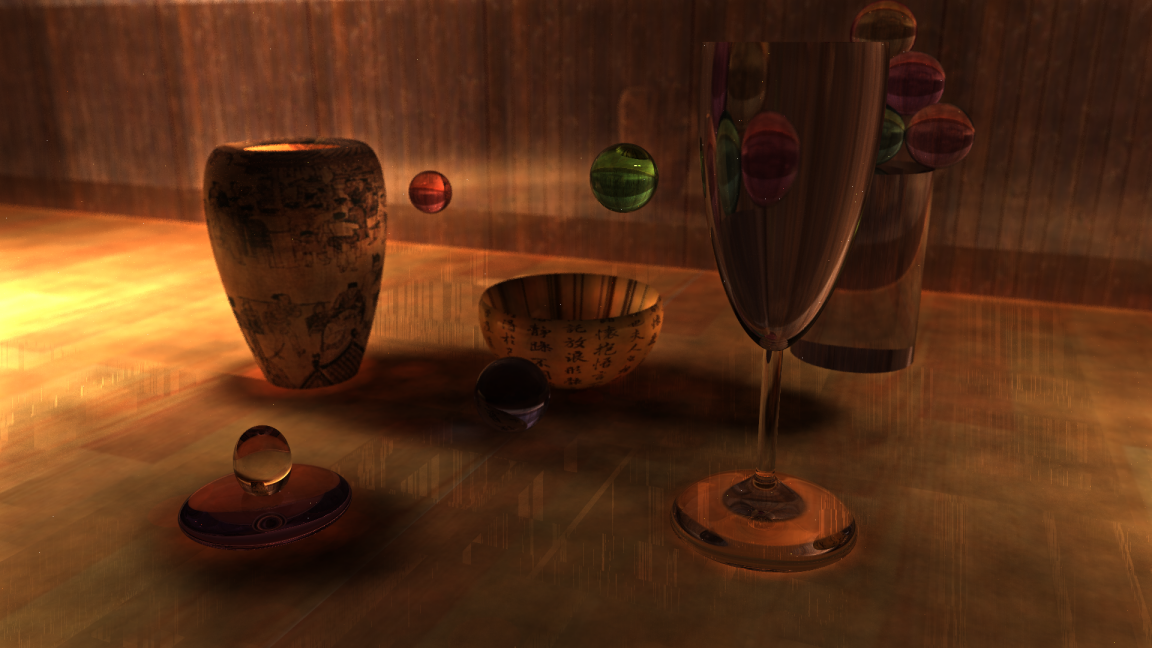
## 效果展示

最终光源功率的调整还有问题，还请谅解！

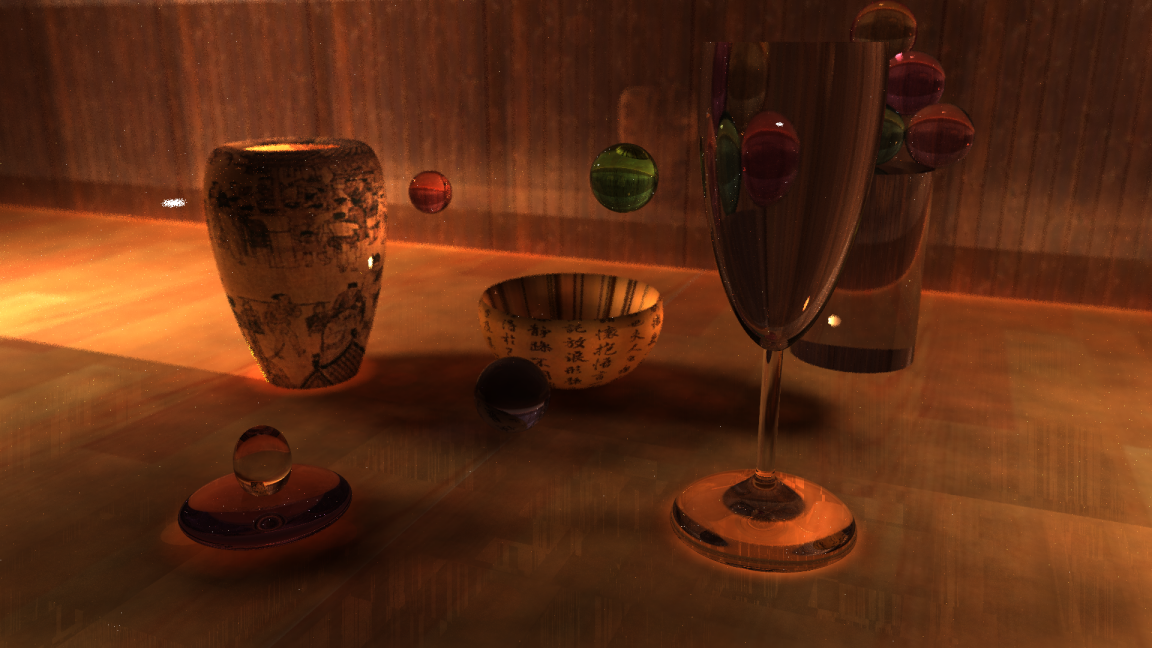
１６－DOF　（高光下DOF明显了）



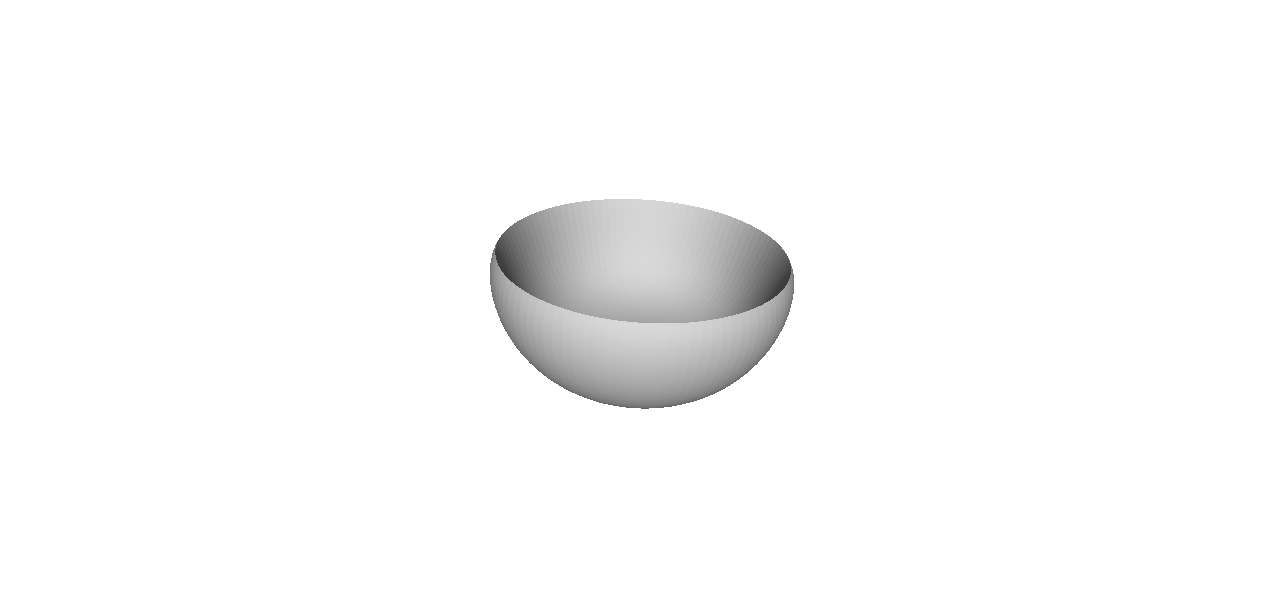
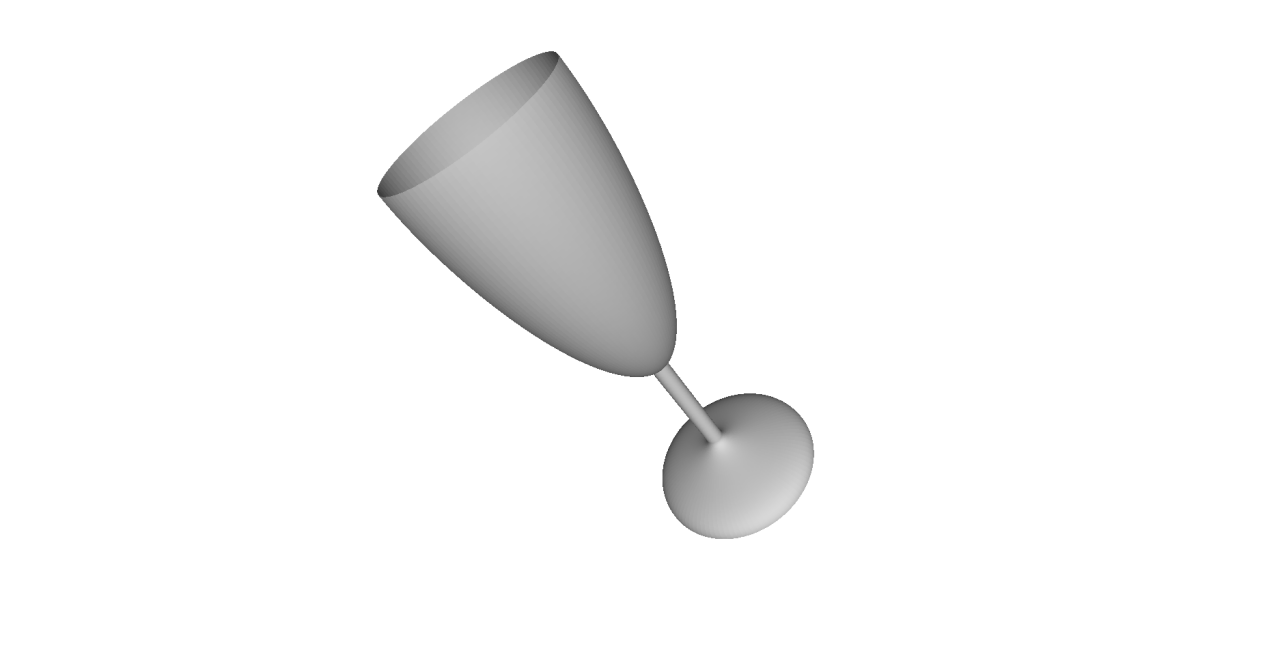
16-DOF　（光线不足时DOF不明显）

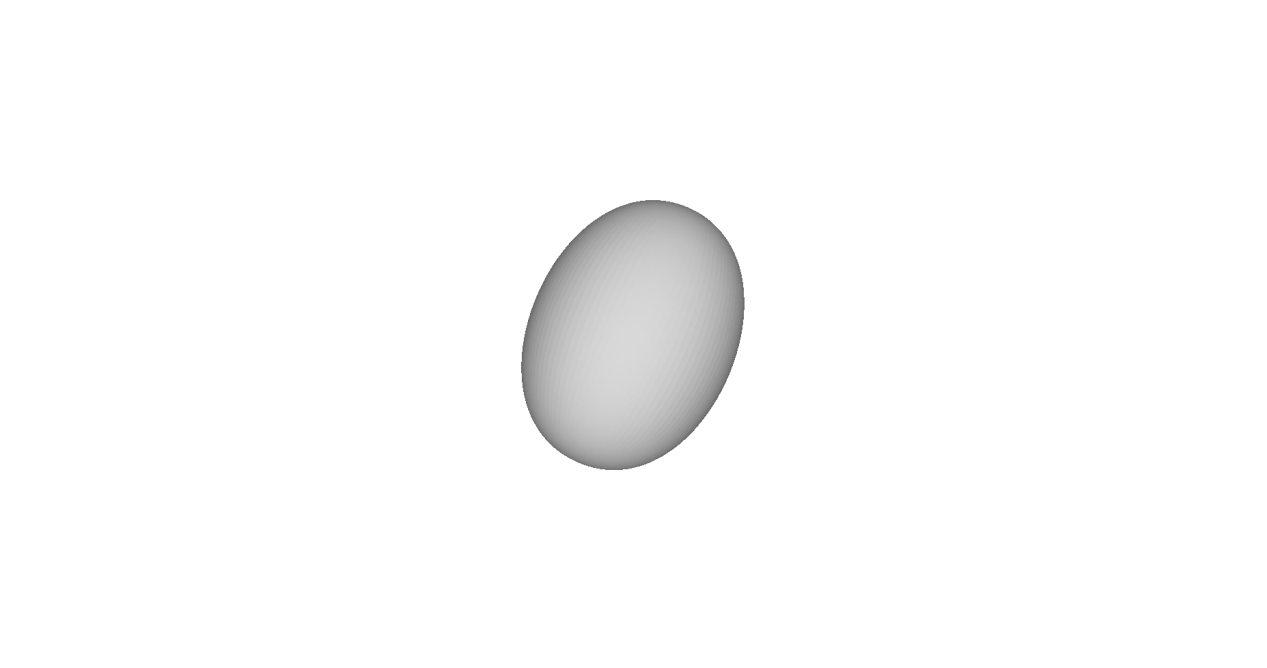
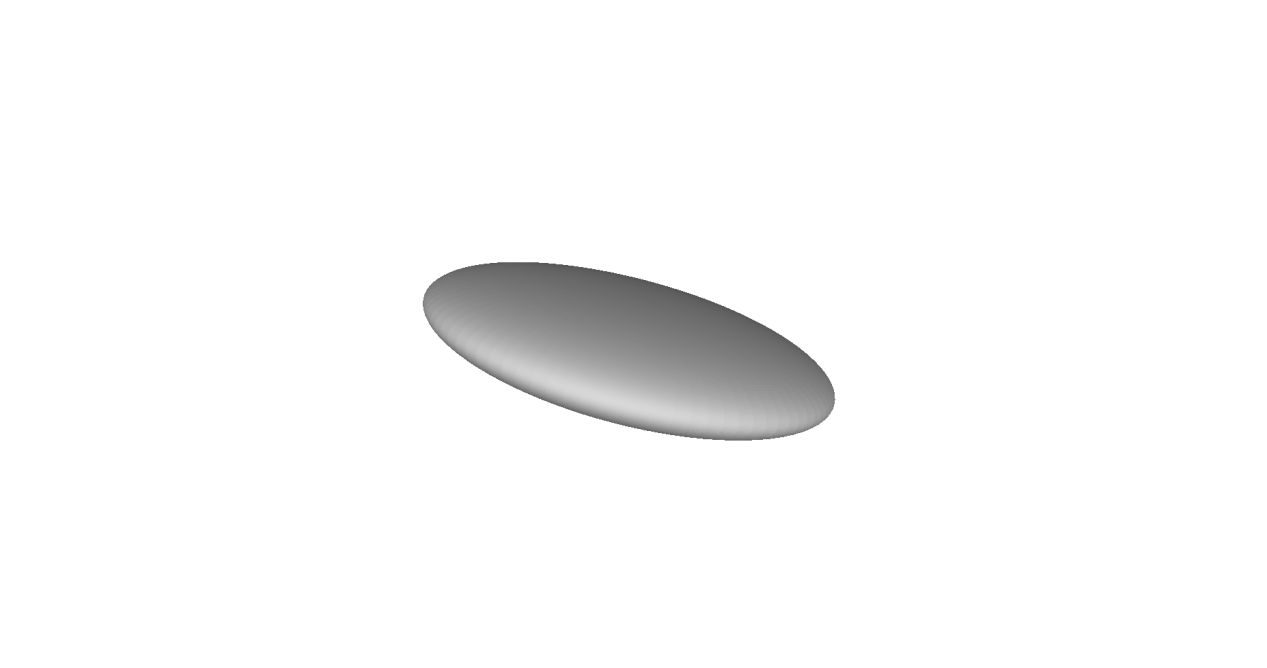


４－ＤＯＦ　（少景深对比图）



## Bezier造型模型





参考资料

[1] <http://m.blog.csdn.net/libing_zeng/article/details/54605140>

[2]http://m.blog.csdn.net/libing\_zeng/article/details/54606500