

光线追踪实验报告

柯云劼

计44 2014012086

2016年6月27日

1 基本功能

1.1 参数球与参数长方体的求交

1.1.1 球

计算直线与参数球方程的交点，选取最近的一个。
这里可利用胡老师上课介绍的方法进行快速计算。

1.1.2 长方体

计算直线与六个面的交点。选取三组相对的面中交点最近的面，再从中选取最远的面。

检验约束情况，计算出该面的交点是否在长方体实际面的约束内；若是则认为与该长方体有交点，且为该交点。

1.2 局部光照模型

使用Phong光照明模型

1.3 理想漫反射

计算表面点P的漫反射光

$$I_d = I_p K_d \cos(\theta)$$

其中 θ 为从点P指向光源的向量和点P的法向量间的夹角

1.4 镜面反射光

计算表面点P的镜面反射光

$$I_s = I_p K_s \cos^n(\alpha)$$

其中 α 为视线方向与反射方向的夹角

1.5 光线追踪的实现

根据物体表面的系数，先计算表明光照，然后根据反射折射定律计算发射光与折射光，递归地返回相应颜色值。

最后观测到的像素颜色即为局部光照与反射折射光的加权和。

2 工程实现

2.1 读入

读入config.txt文件来进行环境场景的配置和设置

2.2 输出

输出使用bmp类，将Color类矩阵的值保存为bmp文件。

2.3 类结构

Camera类：摄像头，管理最后返回的像素信息，发射光线

RayTrace类：光线追踪，获得摄像头发射的光线与场景类进行交互，返回计算的值给摄像头

Scene类：场景，存储光源与物体等信息

Object类：物体的虚基类，存储交点和材质信息，并有读取、求交等纯虚函数；派生出Sphere、Box、Plane、Mesh、Model等类，其中Model类内置了k-d tree，方便求交运算

Inter类：交点，存储交点位置、距离和轴向等信息

Light类：光源的虚基类，存储光源位置，并有读取、计算颜色等纯虚函数；派生出PointLight、PlaneLight类

3 拓展功能

3.1 抗锯齿效果（超采样）

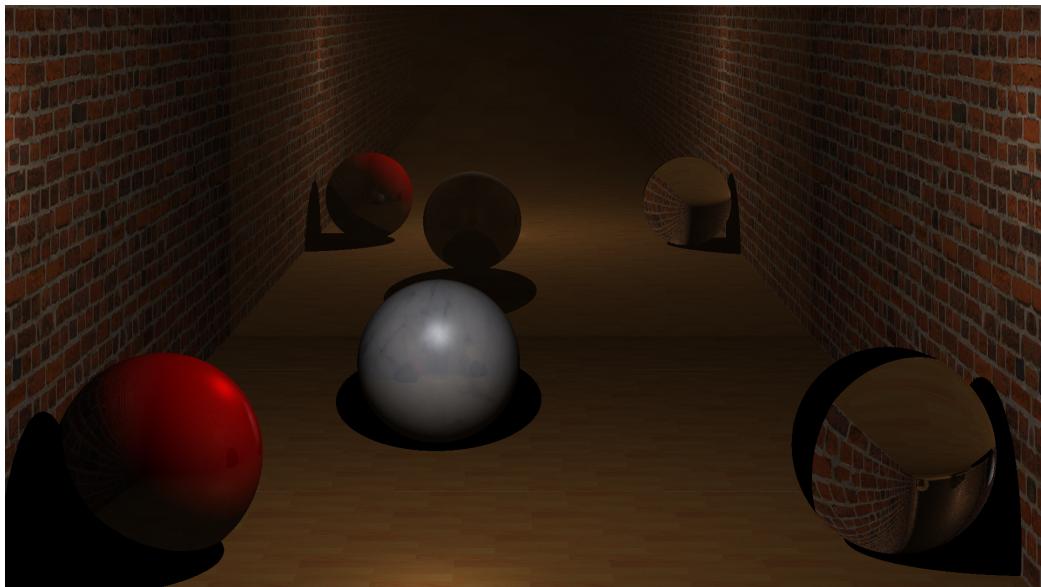


图 1: 无超采样

实现：对于每个像素在领域内密集采样，将采样结果平均

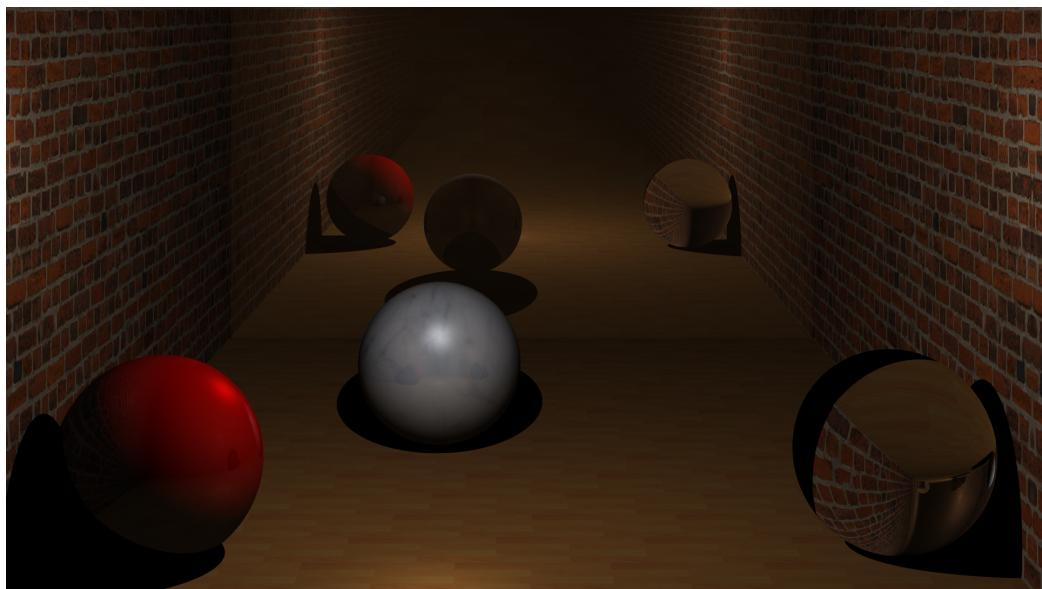


图 2: 16X超采样

3.2 物体求交加速 (*k-d tree*)

对于复杂模型而言使用k-d tree能够有效加速求交运算，将每次 $O(n^2)$ 的求交计算可降到 $O(n \log n)$ 。

3.3 复杂网格模型 (**obj**模型读取)

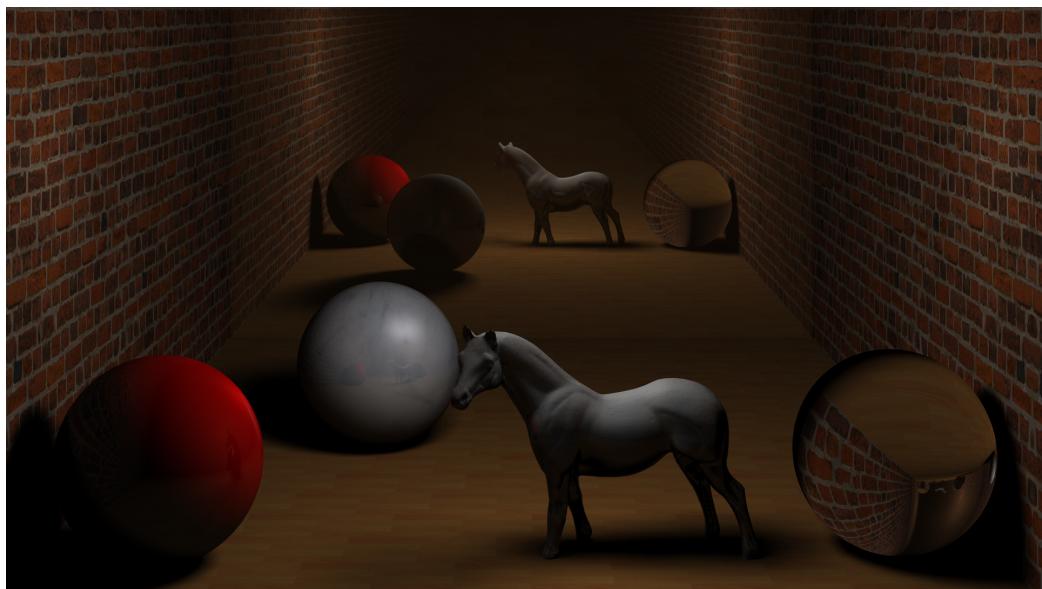


图 3: 复杂模型

读取obj格式文件，并生成数个面片物件，由模型物件统一管理，并构建k-d tree。

3.4 多样化的材质效果

可在配置文件中调整各参数比例，来实现多样化的材质效果，上图即实现了简单的玻璃球和大理石表面材质。

3.5 贴图（UV映射、凹凸贴图）

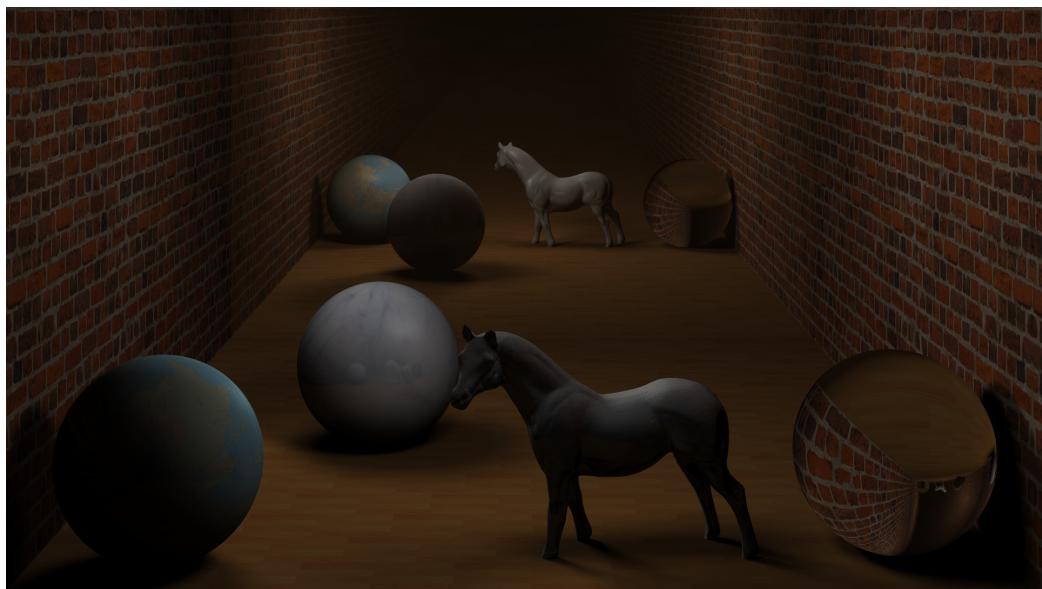


图 4: 贴图

如上所示，实现了UV映射，其中地球即是使用的UV映射。

3.6 软阴影（面光源采样）

实现：检测局部光照时，将面光源中一定数量的点视为点光源求阴影，最后将值平均即可

4 心得体会

工程中遇到了许多需要扩展的效果，良好的OOP设计能使得后续迭代更顺利。

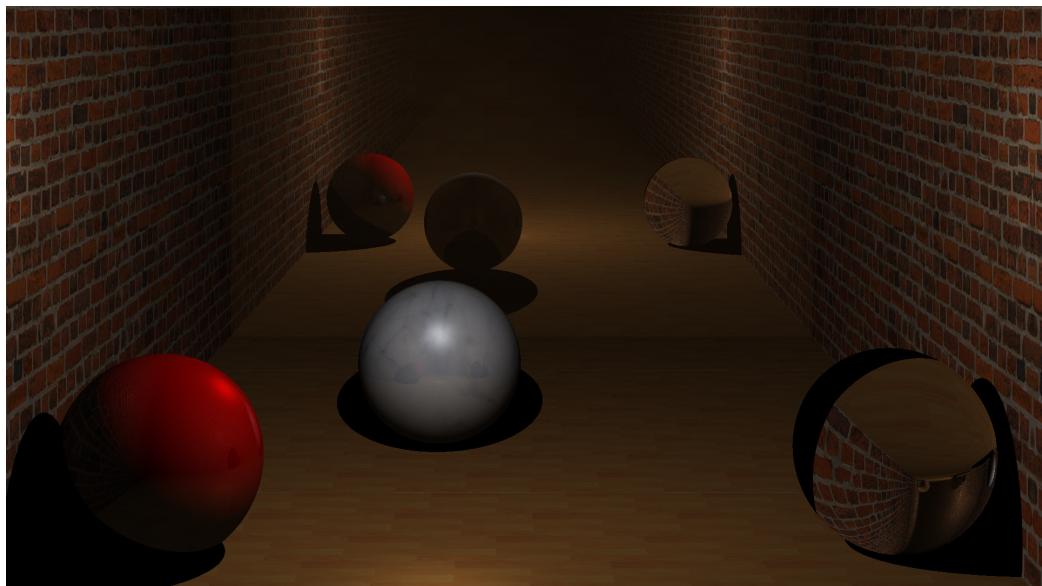


图 5: 硬阴影

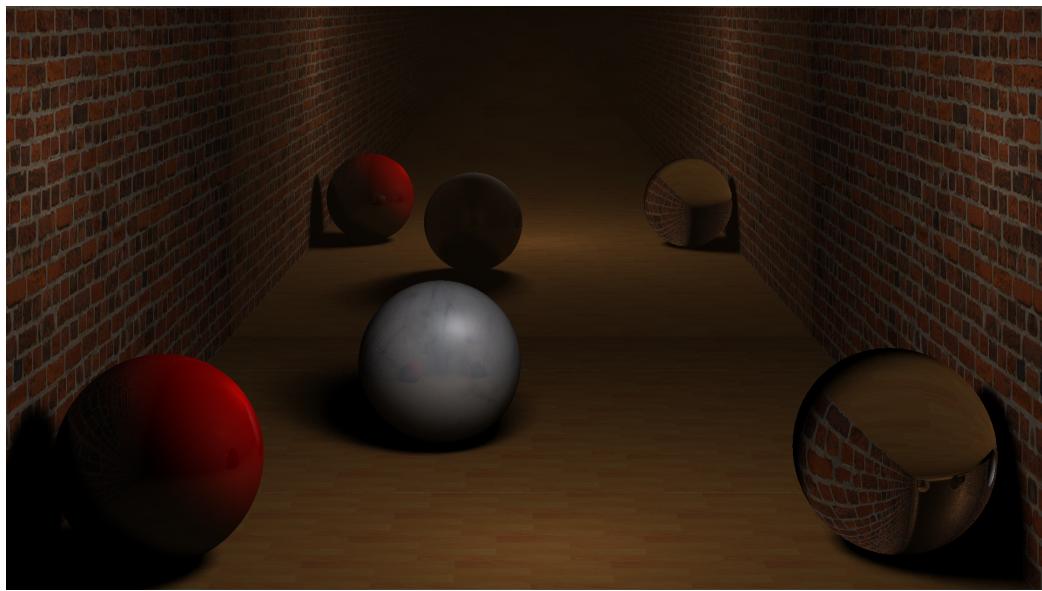


图 6: 软阴影