

# 《计算机图形学基础》课程大作业

## 光线追踪 实验报告

计 34 马也 2013011365

### 1 实现功能

本实验实现基于光线投射的递归式光线追踪算法，使用 Phong 模型作为光照模型，并实现了折射反射等基础功能。扩展功能有：

- 通过超采样和高斯近似实现抗锯齿
- Kd-Tree 求交加速
- 复杂场景的三角面片（obj 和 mtl 文件的解析和处理）
- 纹理贴图
- 软阴影
- 景深效果

### 2 实现方法

本实验由 C++ 语言实现，使用 Visual Studio 2012 作为集成开发环境，额外使用了 OpenCV 第三方库，但仅用其完成图片的写入（bmp 文件）和材质图片的读取（jpg 文件），没有使用任何其他功能。主要采用的结构体和类如下：

- 向量类 Vec：三维向量，用于坐标，颜色和法向量
- 相机类 Camera：包含相机的位置、方向、光圈信息，以及相机坐标系的坐标轴信息
- 光源类 Light：基类，包含两个派生类：点光源 PointLight 和面光源 PlaneLight，包含求交检测等成员函数
- 图片管理类 Film：负责保存颜色矩阵并写入图片
- 形状类 Primiti：基类，包含两个派生类：三角形 Triang 和球 Sphere，包含求交检测等成员函数
- BRDF：包含派生类 PhongBRDF，内含 Phong 模型的光照模型计算函数

- **File**: 基类, 包含派生类 **ObjFile**, 负责从 **Obj** 文件中读入场景信息
- **场景类 Scene**: 包含所需的场景信息, 如形状, 灯光等等
- **光线追踪器 RayTracer**: 负责光线追踪
- **KDTree**: 负责求交加速

其他一些其余用到的结构体有:

- **KDNode**: **KDTree** 节点
- **Sample**: 采样点
- **Ray**: 光线类
- **Texture**: 材质类

本实验采用老师课堂所讲的光线追踪作为基本框架, 实现了折射, 反射, 漫反射等基础功能, 在此不做赘述。在扩展功能方面, 主要实现的功能如下:

**抗锯齿**: 采用超采样抗锯齿的方法, 将一个像素点分割成 9 个小格, 每个小格发出一条光线, 在对这 9 个小格采用高斯近似的方法, 得到最终的颜色。

**求交加速**: 使用 **Kd-Tree** 进行求交加速, 效果显著, 渲染 20 万个三角形面片的图片可以十几秒就得到结果, 有效降低了算法复杂度。

**复杂场景三角面片**: 实现了 **Obj** 文件和 **mtl** 文件的读入和解析, 并寻找复杂模型的场景文件, 取得了不错的效果。

**纹理贴图**: 使用重心坐标系映射的方法, 将一张图片映射到一个三角形之中。

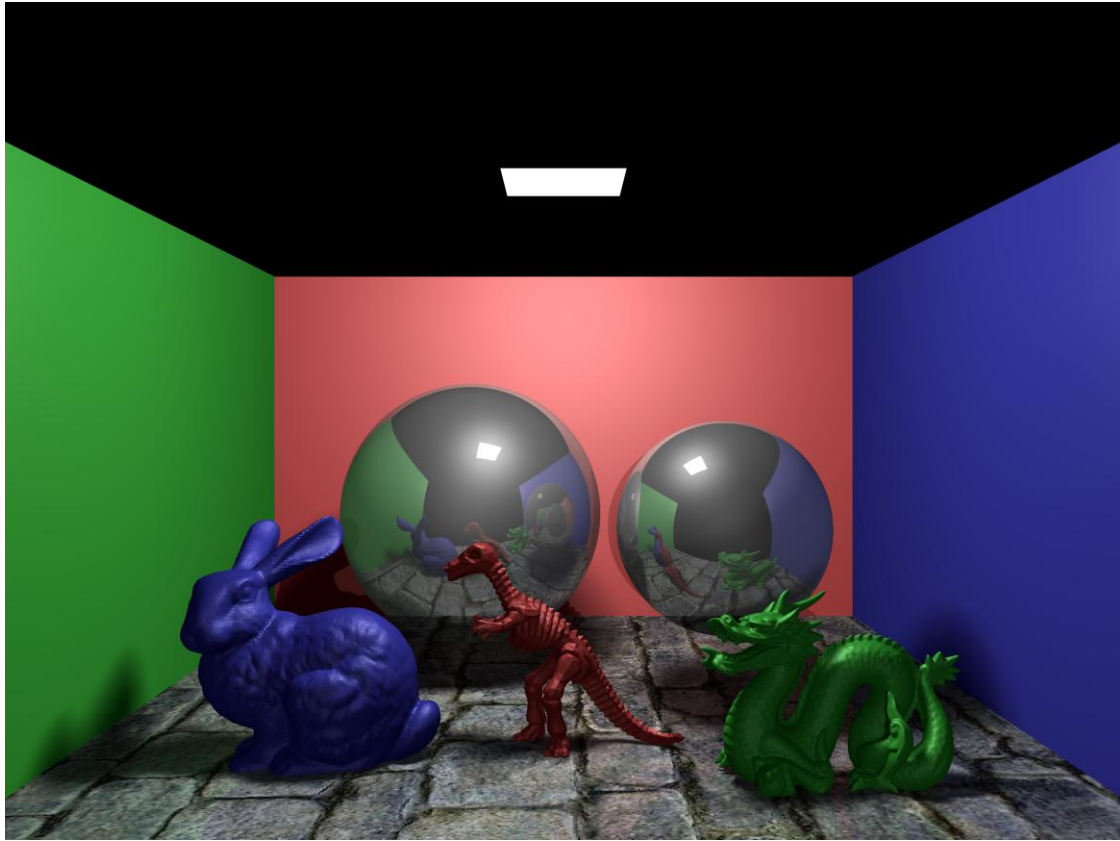
**软阴影**: 使用面光源超采样的方法, 将其分割成若干小格, 每一个进行求交检测, 得到交到的比例, 换算出光强。为了让锯齿效果减少, 采用了随机化的方法进行。

**景深效果**: 使用光圈、焦距等概念, 随机地发射出多条光线进行采样, 最终将他们取平均, 得到景深的效果。

### 3 效果图

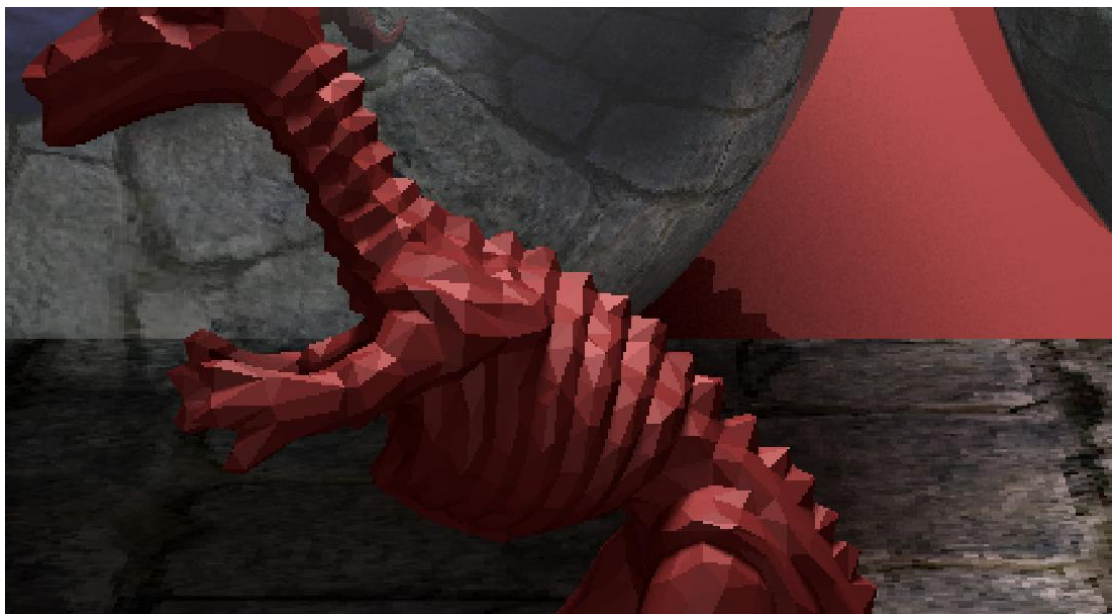
#### 3.1 整体效果 (不含景深)

包含复杂模型求交加速 (超过 30 万三角面片), 材质贴图 (地板), 软阴影 (面光源), 抗锯齿效果。



### 3.2 抗锯齿效果

以下四张图取自刚刚的成果图，前者为不使用抗锯齿，后者使用抗锯齿，可以发现后者的边缘更加柔和。

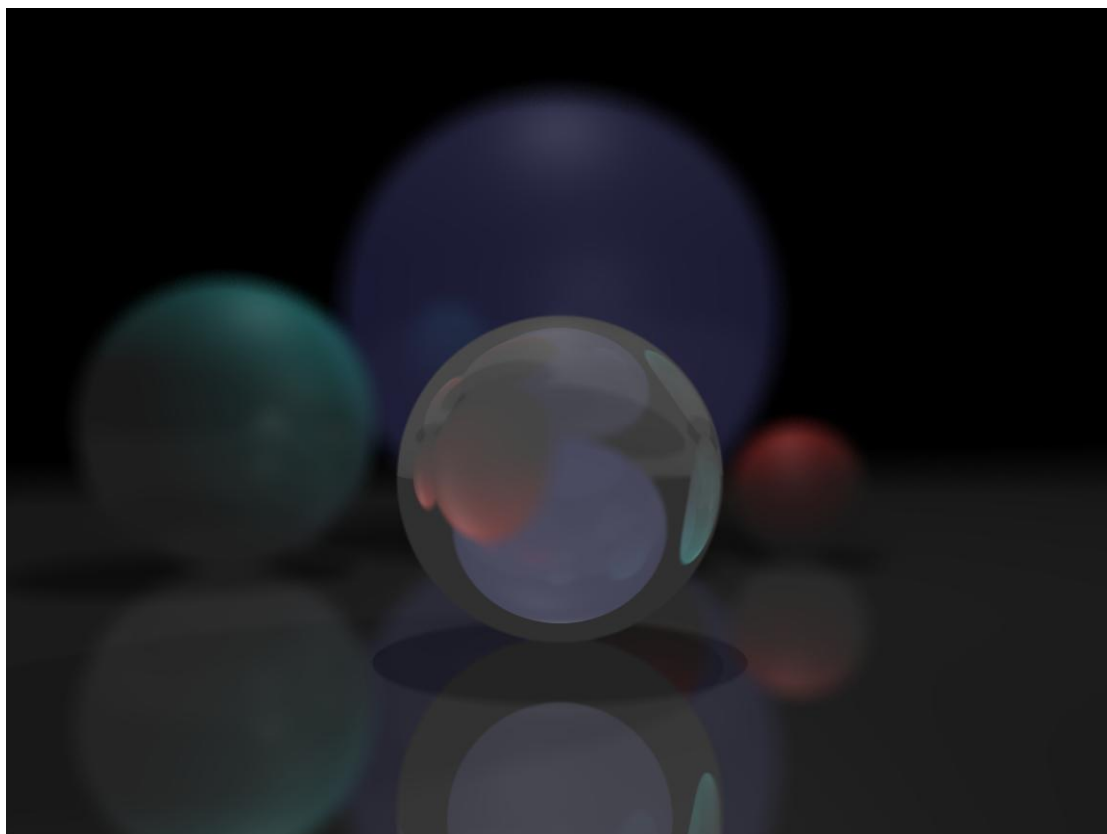


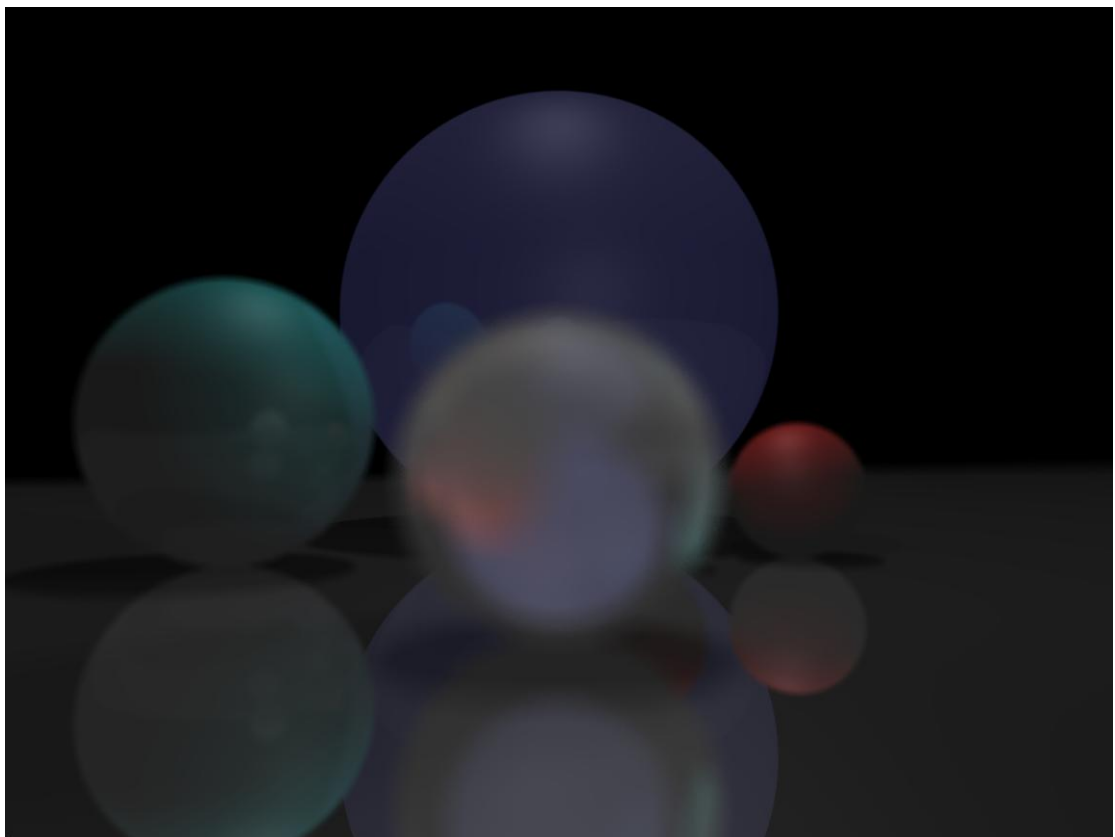
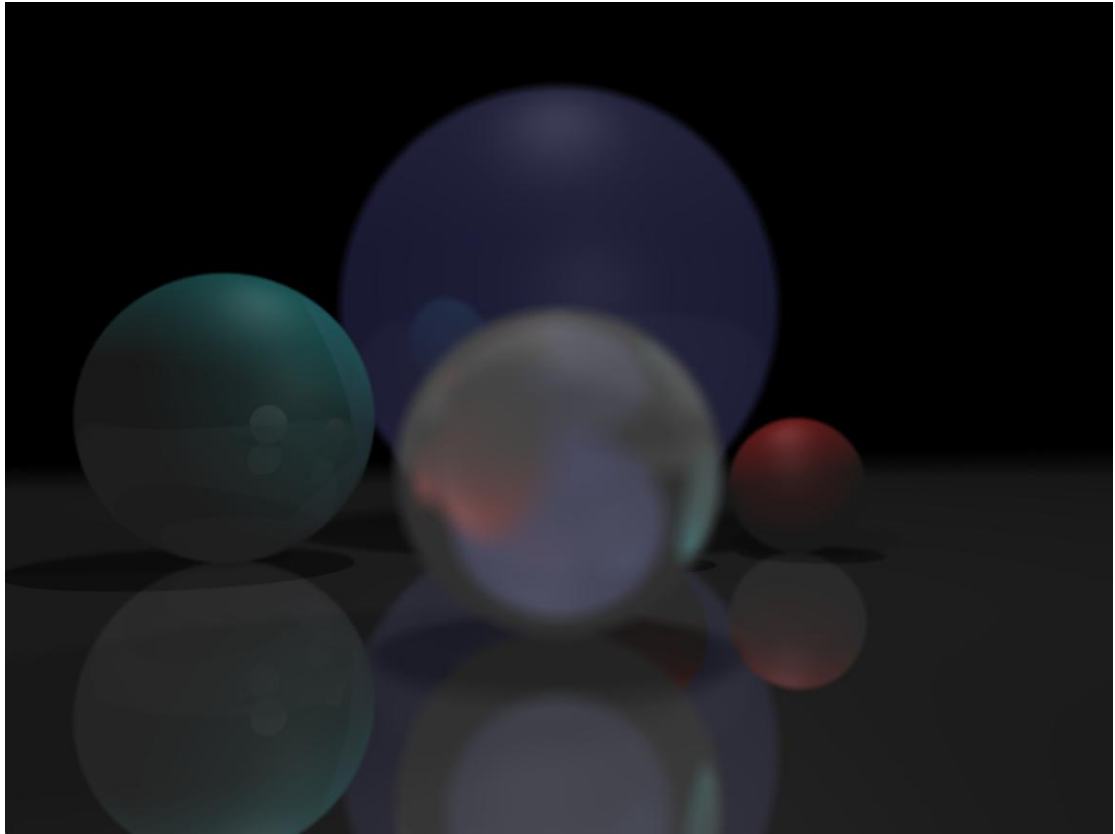


由于压缩的缘故，效果可能不是那么明显，可以再 Outputs 文件夹的 Cornell 文件夹下找到原图 `cornell_AA_0` 为不带抗锯齿，`cornell_AA_1` 为带抗锯齿。

### 3.3 景深效果

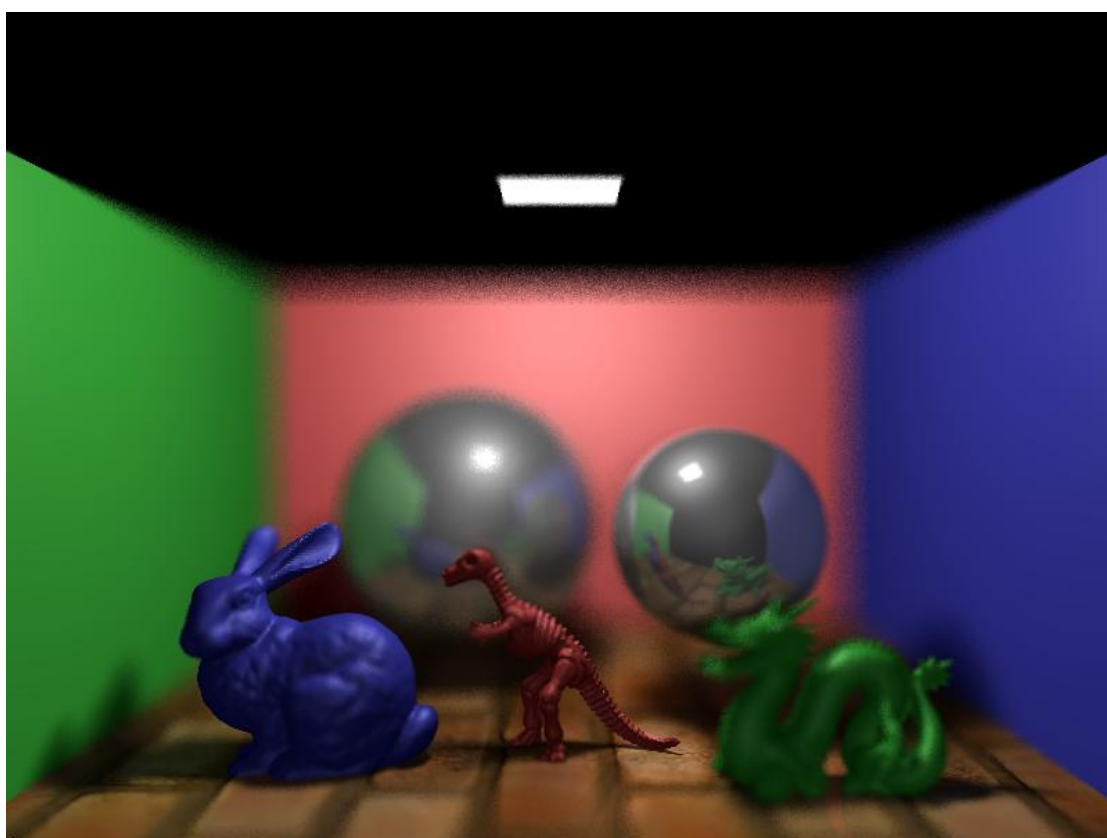
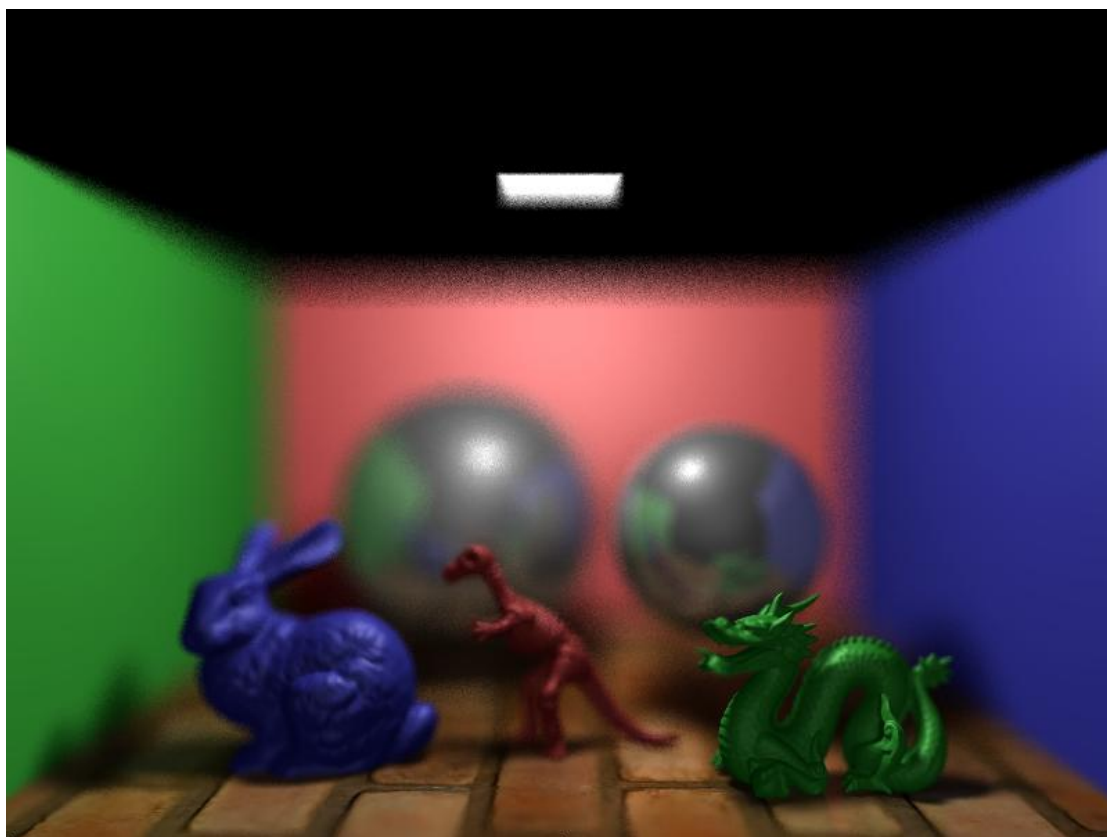
首先是简单场景下的景深效果（Outputs 文件夹下的 DOF 文件夹）：

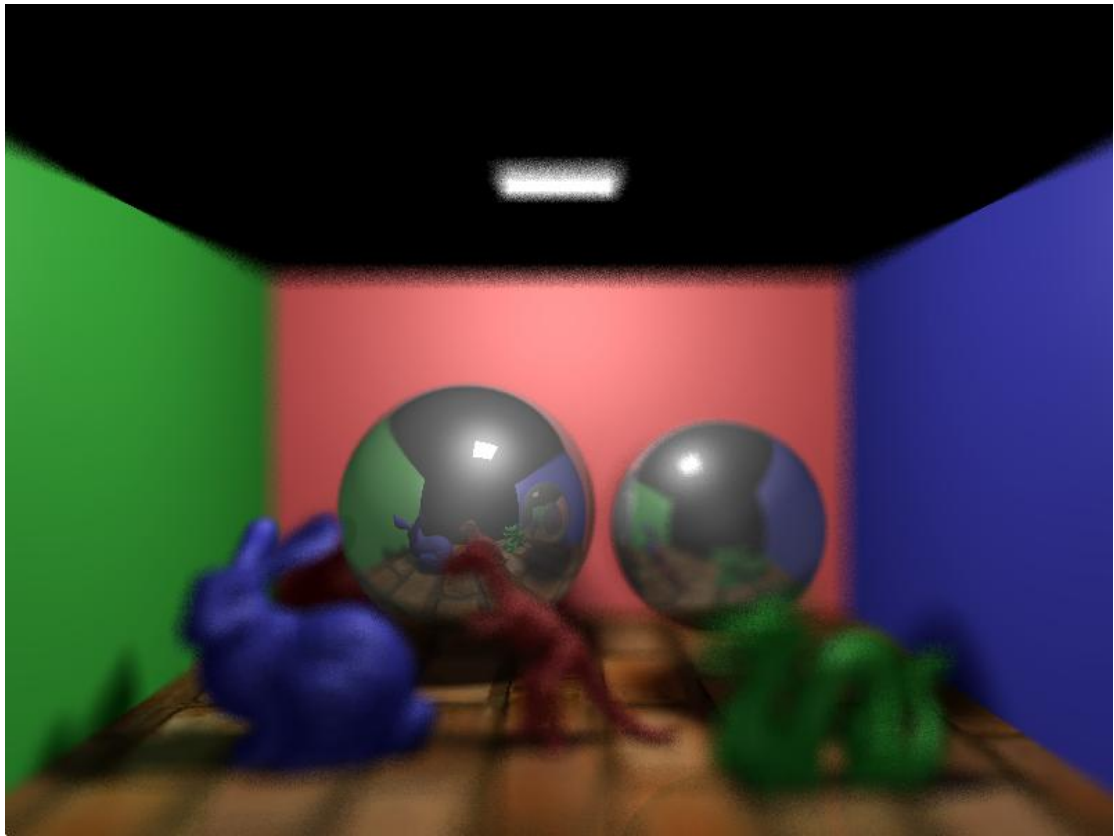




接着是复杂场景下的景深效果（Outputs 文件夹下的 Cornell 文件夹）：







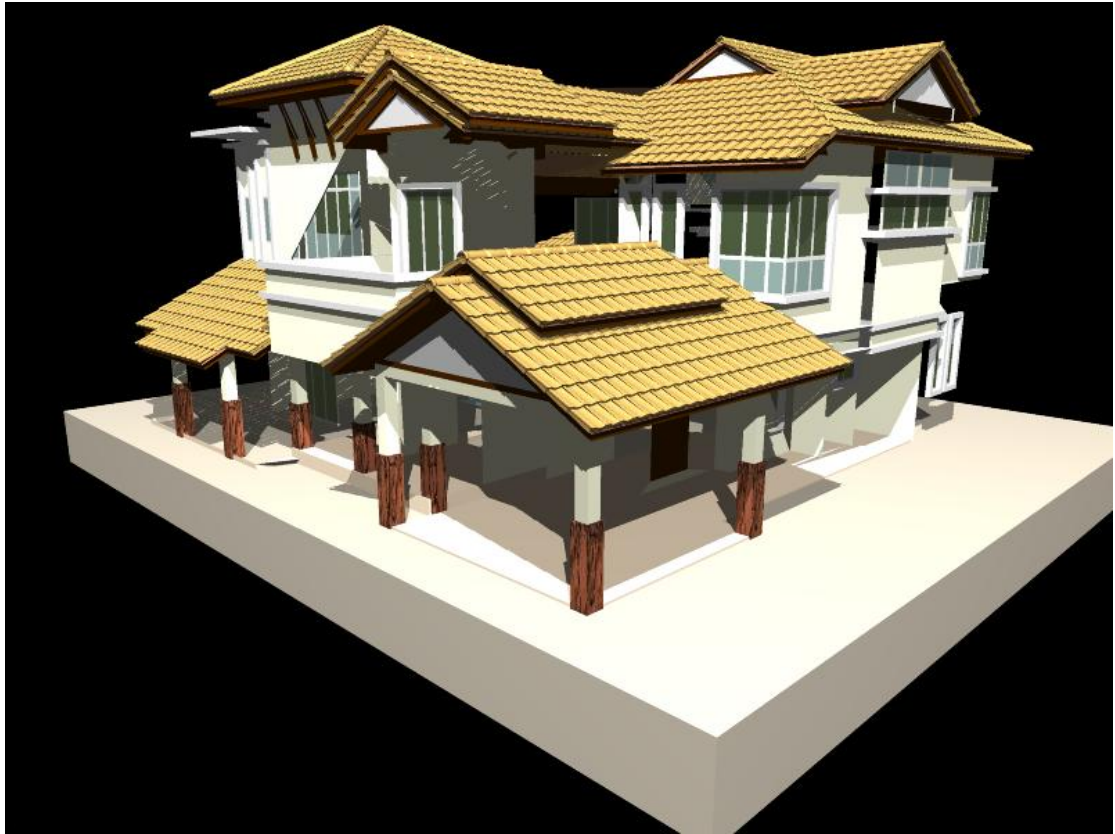
可以看出景深效果由近及远，十分显著。

### **3.4 复杂场景和模型、材质**

（以下模型均来自互联网）

首先是一个建筑模型：



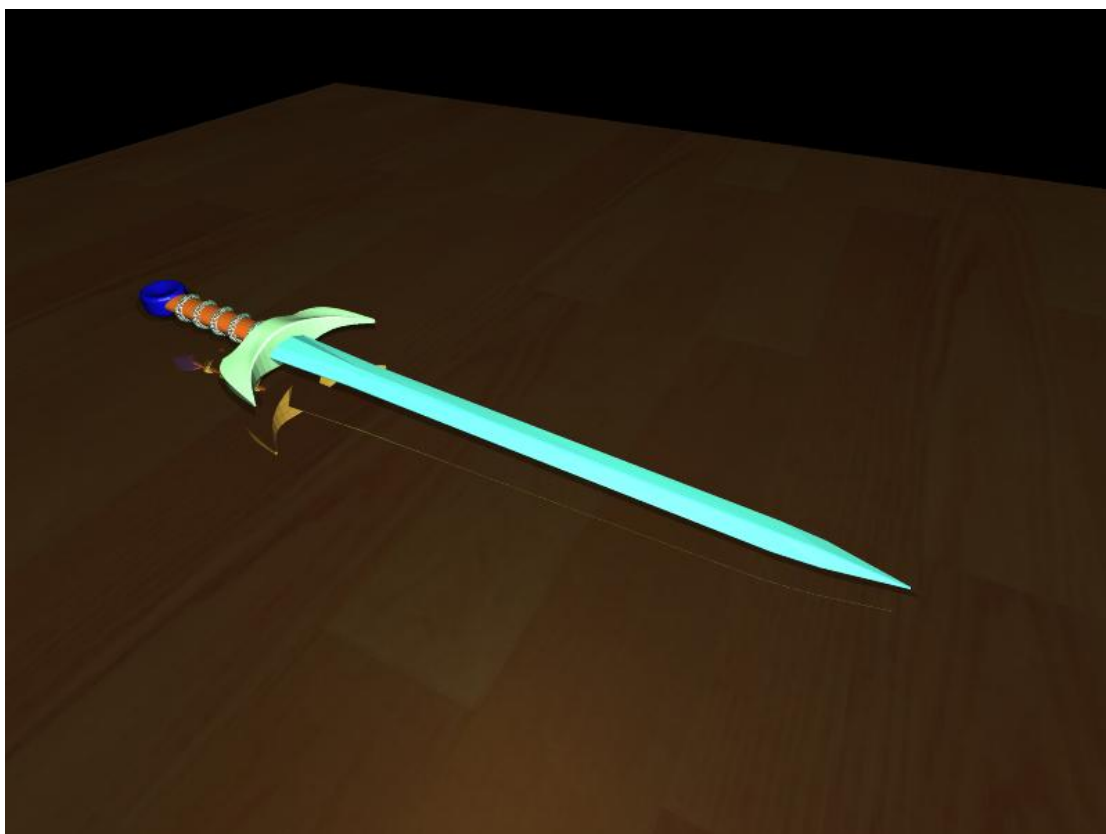


接着是带有纹理贴图的飞机模型：



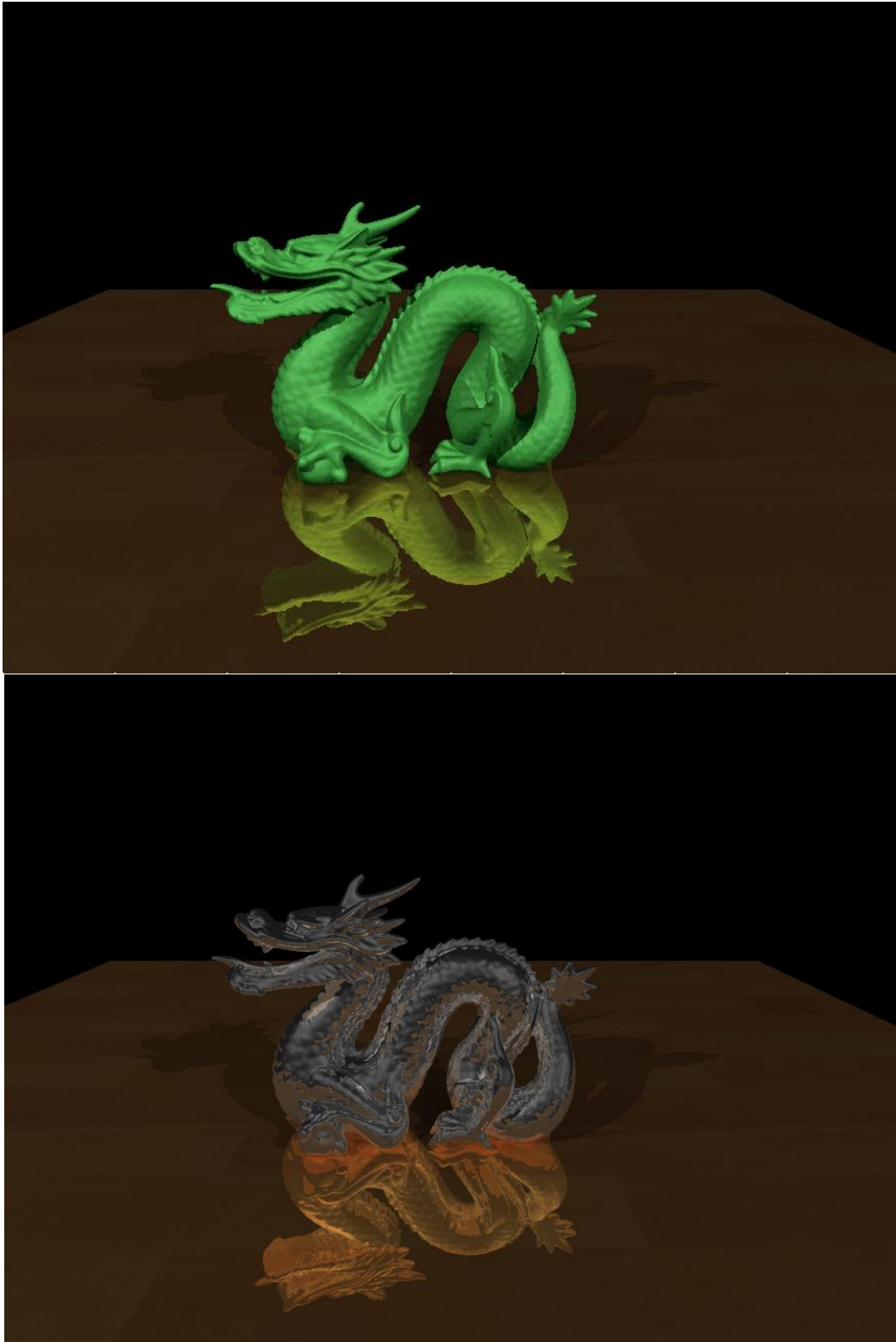


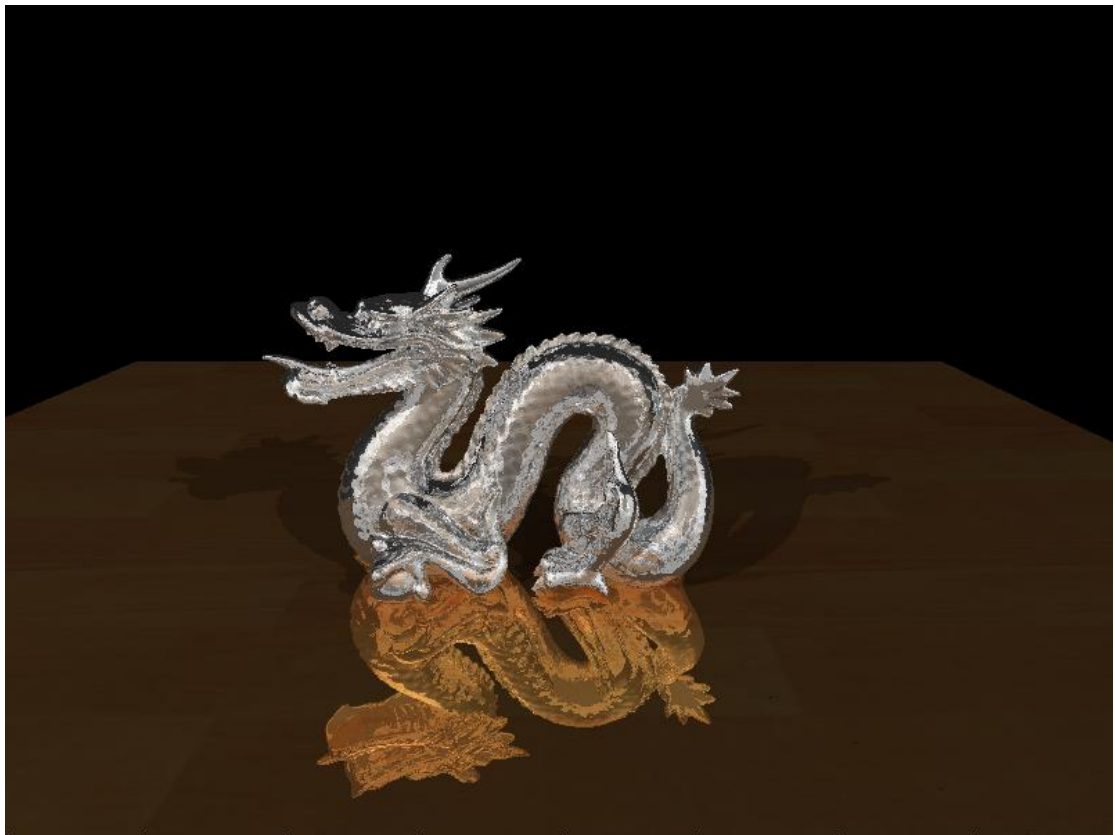
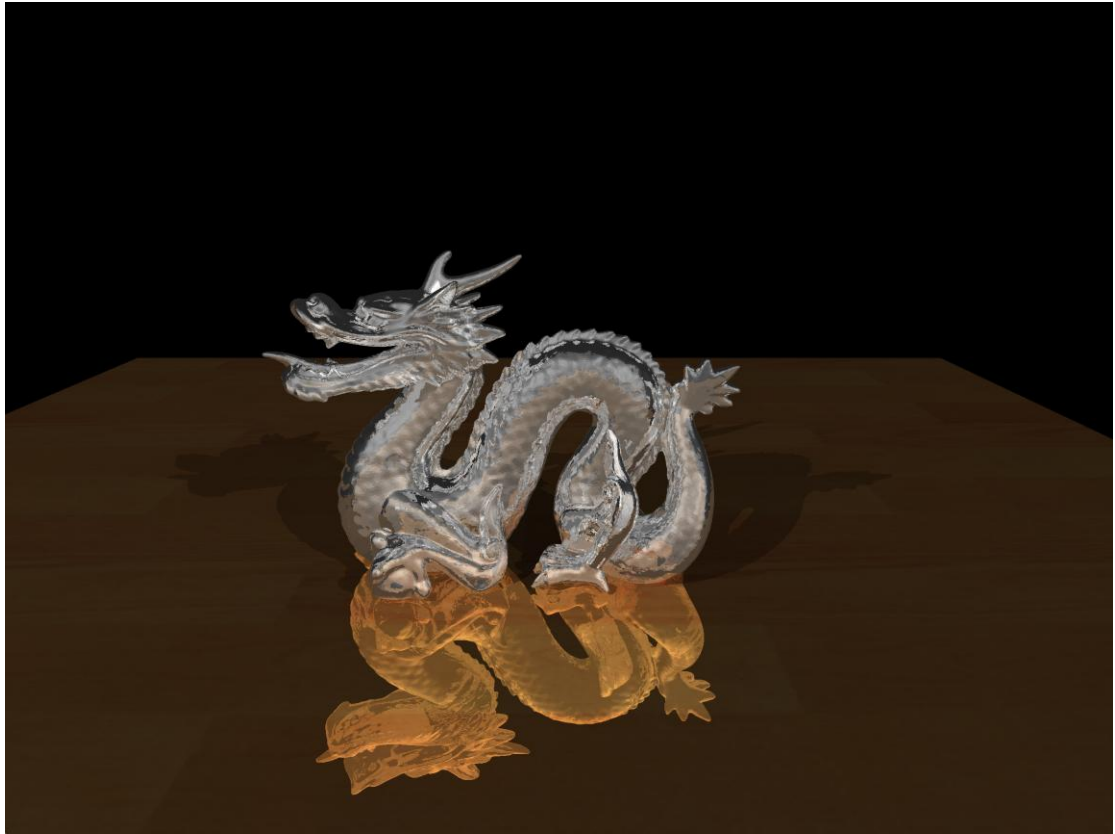
最后是一把剑：



### 3.5 光照模型

采用不同的光照模型（漫反射、反射、折射、高光），得到如下四种龙：





## 4 实验总结

通过这次实验，我掌握了光线追踪的基本实现方法，同时还添加了许多扩展功能，让程序跑得更快、更好、更逼真。这次编程大作业也是大学以来代码量最大的工程之一，它让我回顾了面向对象程序设计的一些概念，提高了我在代码架构方面的能力和水平。更重要的是，它让我对图形学有了更加深刻的认识。总之，这次实验基本圆满完成了。