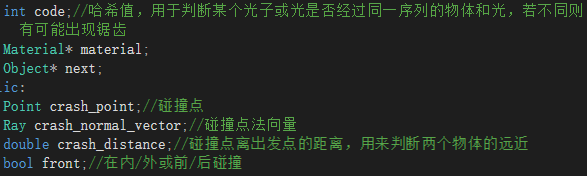
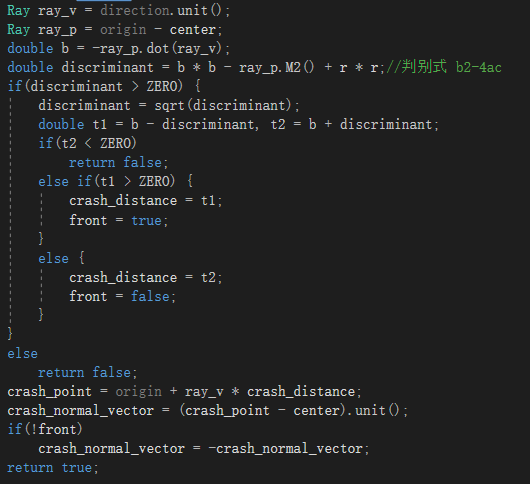
**光子映射实验报告**

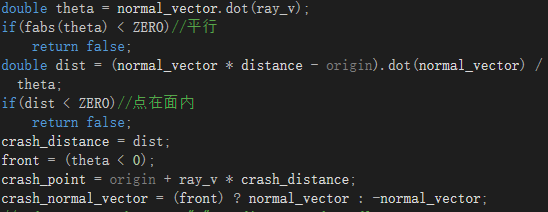
本次作业我参考了诸多资料，采用了光线跟踪加**光子映射**的形式实现。把场景的光照分为三个部分：直接光照(漫反射表面)，间接光照，镜面和折射光照，直接光照和镜面和折射光照由光线跟踪来计算，间接光照由光子图来计算，将光子图可视化。

1. 首先我定义了光线、颜色等基础类，包括基础的加减乘除，定义颜色的光强为rgb分量的平均值，定义了光线的模长、单位向量、垂直向量、随机向量、反射、折射、旋转、散射等函数。
2. 接着定义输出输入图片的辅助类，使用opencv读入输出jpg格式的图片。
3. 接着我定义了物品类，由object基类派生出Sphere、plain、bezier三个子类。Object中有一个Material类储存材质的信息，包括颜色（漫反射、反射、高光），吸收的颜色（光线通过折射的物体的内部），折射率、反射率等，以及**纹理贴图**的图片。在object类中还储存了哈希值，用以判断一束光线是否有相同的路径（即击中同一个序列的物品）。还有最近一次碰撞的信息，包括碰撞的位置、法向量、光从发射到碰撞所经过的距离，以及碰撞的方向（前后）。

在球体计算碰撞时，使用判别式解二次方程，得到碰撞点。



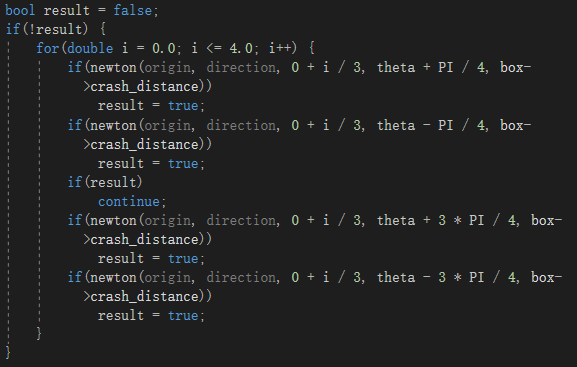
计算平面碰撞则计算光线方向和平面法向量的夹角，判断是否平行、光源是否在面内，以及碰撞方向。



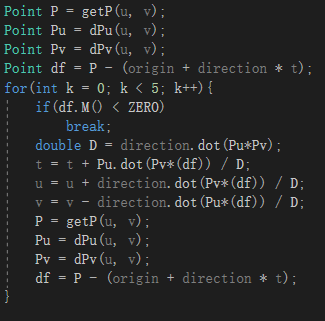
计算纹理时，则把碰撞点的球面坐标映射到平面上，得到其颜色并返回。

对于**Bezier**图形，通过输入起始点，起始方向得到图形的位置与构造方向。接着输入n组二维点作为控制点，其中第一维为起始方向的距离，第二维为与起始方向向量的垂直的垂直方向上的距离。

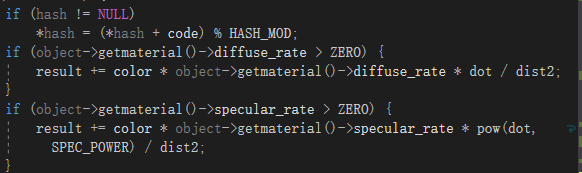
接着将这条曲线沿起始方向旋转，得到一个三维图形。使用一个**包围球**，球心为起始方向最大最小点的中点，半径为中心与离中心最远的点的距离。计算碰撞时，先判断是否与包围球碰撞，是的话取碰撞点离垂直方向的角度为初始角度，在（0-1）中取四个等距的点为u的初值，碰撞点的距离为t的初值，进行**牛顿迭代**。一开始我使用三维曲面来表示Bezier曲面，可是在求交上需要迭代的次数较多，效率较低。采用曲线旋转可以较好地确定包围球、以及迭代初值。



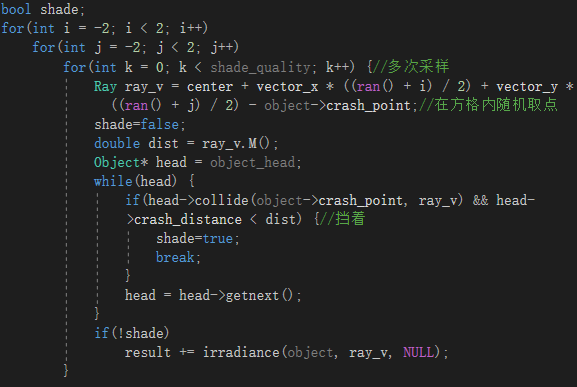
迭代方程采用书上的方程，最多进行五轮。对u求导即为对曲线求导再旋转，对v求导为对旋转过程中的三角函数求导，对t求导即为光的方向。



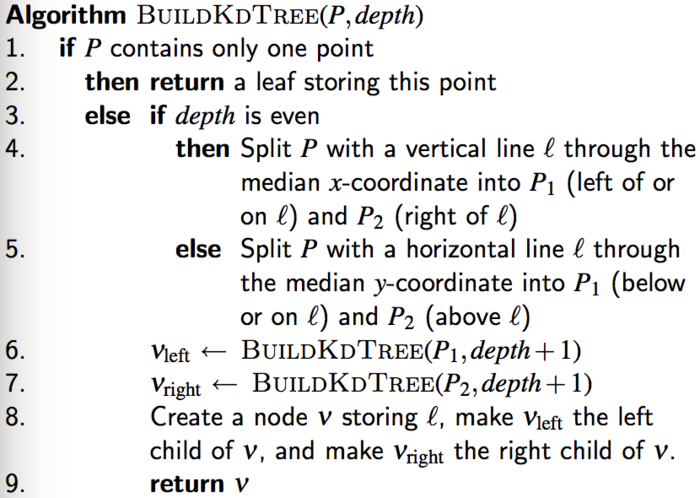
1. 光源类，由基类派生出平面光源和点光源。由光源计算从该光打在某个物体上的直接光照，包括漫反射和高光。



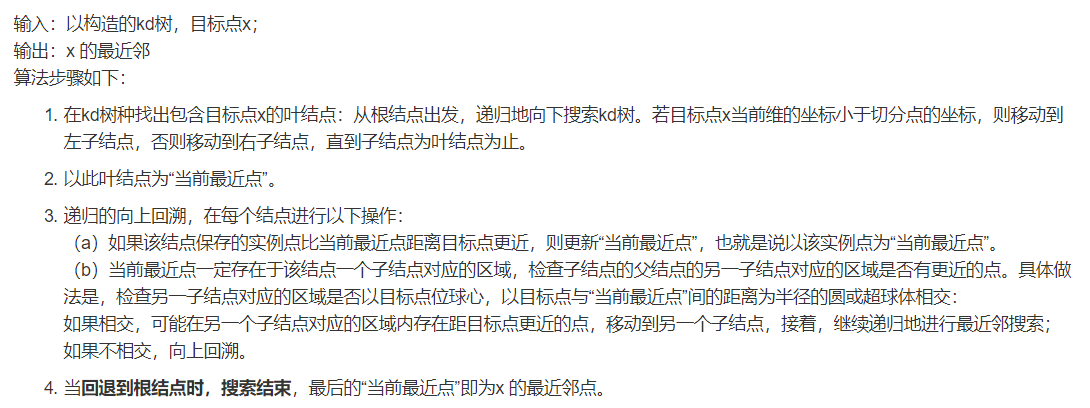
计算实际光照时，对点光源判断光源与物体之间是否有遮挡，有则返回空颜色（0,0,0），否则返回光照结果。对面光源使用随机多次采样，将面分为16个小方格，每个小方格内随机取几点发出光照，得到结果平均后即为该点直接光照。这个方法可以计算**软阴影**。



1. 构造场景类和相机类，其中场景类负责读入txt文件中的初始化数据，初始化一个场景，包括物体和光源。相机类负责储存图像的参数，包括大小、光子数、图片阴影质量。从相机中心发出光线，方向为图片的像素的相对于中心的角度。
2. 构造光子和光子图，光子包含颜色、位置、方向、以及在kd树中的分界面。在光子图中，使用数组储存kd树，递归的构建树。

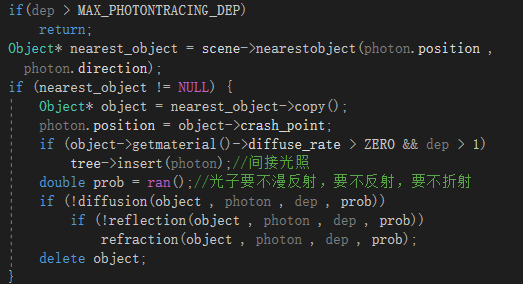


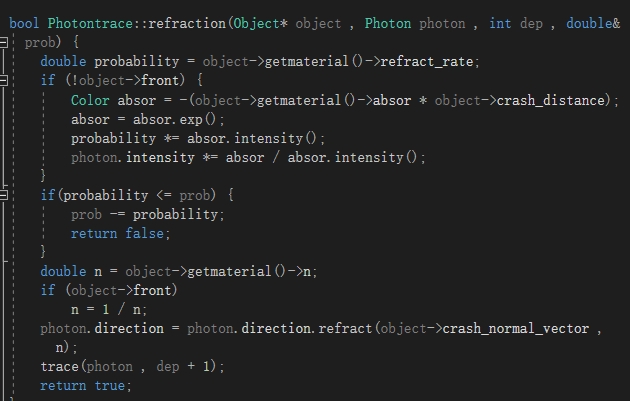
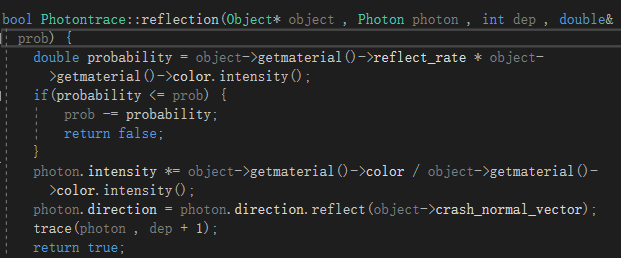
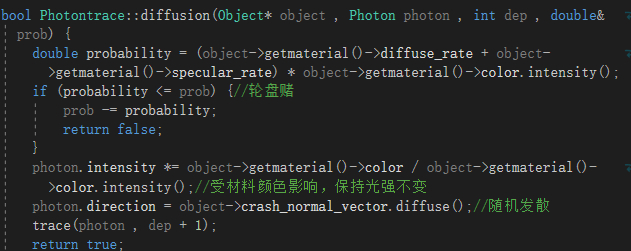
查询时则构造一个最近光子的结构体，存储位置以及得到的光子。采用算法，具体见文件。



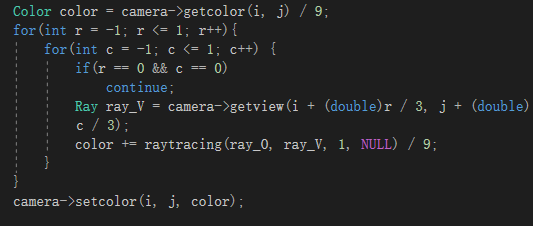
得到光子后，取其平均能量为该点的间接光照。

1. 光子映射算法类，由光源发出光子，对其进行追踪，如果在路径上有物体，使用轮盘赌判断它是漫反射、反射还是折射。漫反射使其沿法向量散射，折射则判断其在物体内部行走的距离，计算吸收光的能量。具体见pt.cpp文件，函数名即为其功能。在光子碰撞时，对反映间接光照的光子（即多次碰撞后、非镜面和折射面）进行储存。最多递归8层。

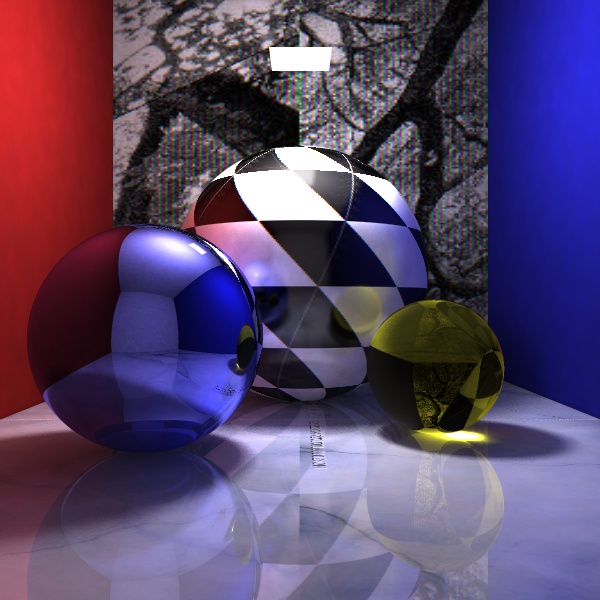




1. 光线跟踪算法，先由场景类构建场景，再从视点发出光线，对每一条光线进行跟踪。光线碰到物体后，根据物体的反射、折射、漫反射特性，分析其直接光照，间接光照和反射折射光照，追踪深度最大为8层。之后对结果进行**重采样**，如果一个像素点与周围像素点的hash值不同，则说明其有不同的碰撞路径，有可能出现锯齿。将该像素点分为9份，重新发出方向不同的光线进行追踪，得到结果平均后作为其最终结果。

具体见rt.cpp文件，函数名即为其功能。

**结果展示**

* 

焦散

折射

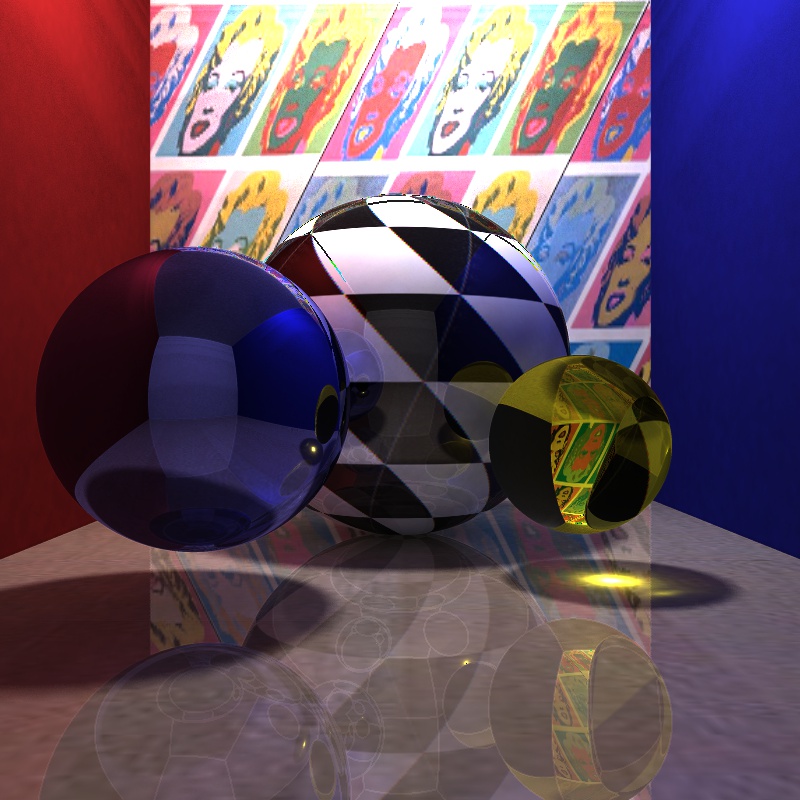
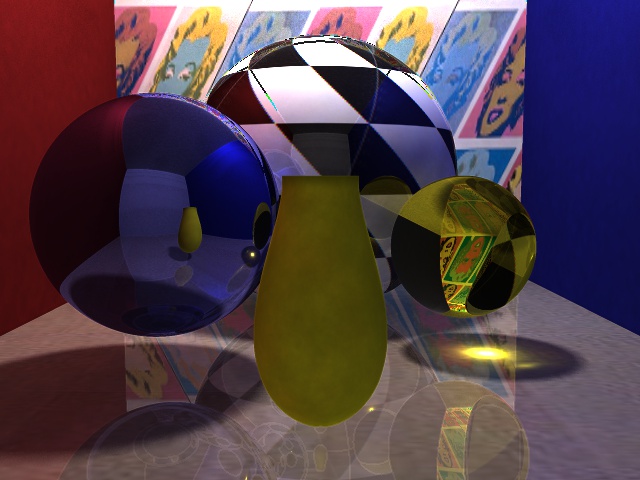
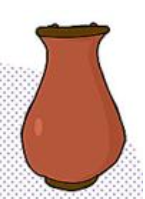
贴图

矩形光源

软阴影

反射

高光

* 第一次结果，由于光源较低，可以直接从图中看到，高光、焦散效果。但是后墙贴图选错了源图片，显示出错。光子量较低，可以在两侧看出明显的不均匀。
* 
* 第二次结果，上移了光源位置，更换了背景贴图，同时增加了光子数，使间接光照较为均匀。
* 
* 第三次结果，为了展示一个希腊式花瓶Bezier曲面的结构，上移了视点。可以从正面（背光）和蓝球反射（面光）中看出在光照下的结果。

**作业总结**

这次大作业我由于前期时间分配不充分，没有按时完成。在补做过程中，我先采用光线跟踪算法完成，再在此基础上完成光子映射。初次实验使用直接将光子图可视化的算法，发现结果较差，有不少噪点。于是我参考了网上的教程，采用直接光照(漫反射表面)，间接光照，镜面和折射光照这一模型构建光子映射算法模型。在Bezier曲线的计算过程中花了很多时间（3-4天），，可以发现只有上半部分完成了求交，而且光子似乎被拘禁在图形内多次反射，因此亮度很高。始终没有发现三维曲面算法的错误在哪，最后没办法采用了曲线旋转的算法，没法构造更复杂的图形，但算是实现了bezier曲面。