****

**计算机图形大作业报告**



**学 院 软件学院**

**专 业 软件工程**

**学 号 3017218135**

**姓 名 全康连**

目 录

目录

**摘要3**

**主体介绍3**

**结果讨论16**

**收获和建议17**

一．摘要

1. 项目名称：Project—Raytracing
2. 操作简介
3. 实现了光线检测碰撞图元cylinder和square
4. 实现了面光源类型sphere light
5. 实现了反射模糊的效果

二．主体介绍

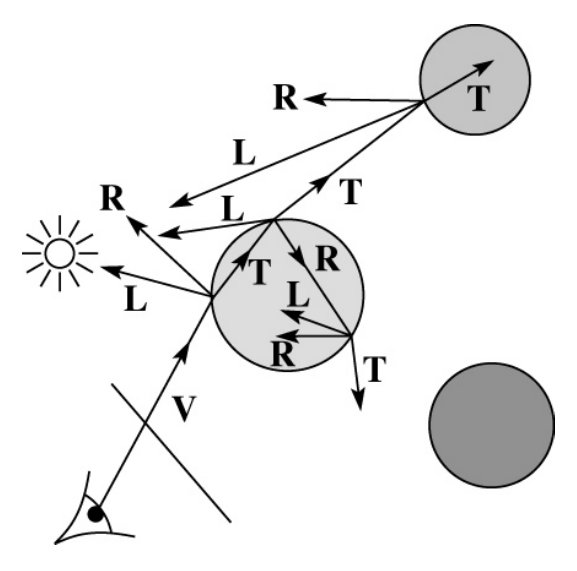
1. 程序简介

该程序完成了光线追踪的基本功能，用户可以使用程序提供的基本图元类型，光源类型书写创建场景文件；运行程序后进行相应的光线追踪并将场景渲染成指定图片保存；光线追踪中实现了光线与图元的碰撞检测，光线的反射追踪，折射追踪，漫反射追踪和反射的模糊效果

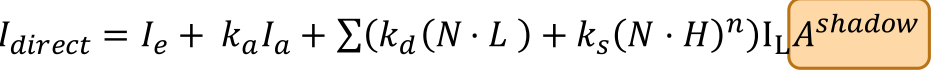
1. 算法
2. 光线追踪数学原理

光线追踪意为通过计算从照相机发出的光线在场景中的路径进而形成眼睛看到的图像；基基本算法思路如下：

1. 对于场景中的每一个像素，生成从照相机指向该像素的初始光线并进行路径跟踪计算
2. 对于计算出的碰撞交点，从该交点出发进行跟踪计算其shadow阴影系数，reflection反射光线，refraction折射光线



对于场景中的最终着色呈现则使用Phong光照模型进行计算，对于Phong光照的计算，要考虑shadow阴影系数的影响，当光线被遮挡时，阴影系数为0，眼睛看不到该处的场景；当光线未被遮挡时，阴影系数为1



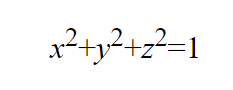
1. 基本图元与光线碰撞计算

光线追踪中很重要的一点是要计算光线与场景中物体的交点，本程序提供了球体，平面，正方体，圆柱体这几种基本图元，图元与光线碰撞的数学原理如下，其中

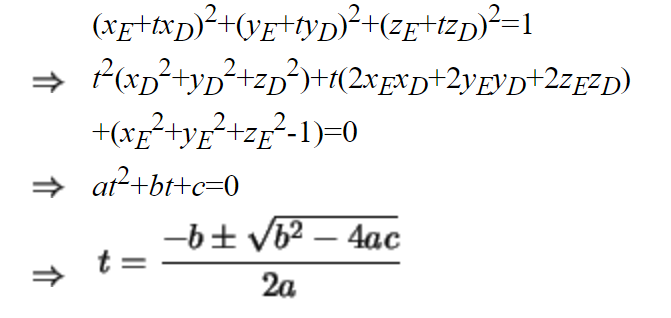
光线上任意一点的表示如下，其中E为光线源点，D为光线方向单位向量

1. **球体(以单位圆为例)**

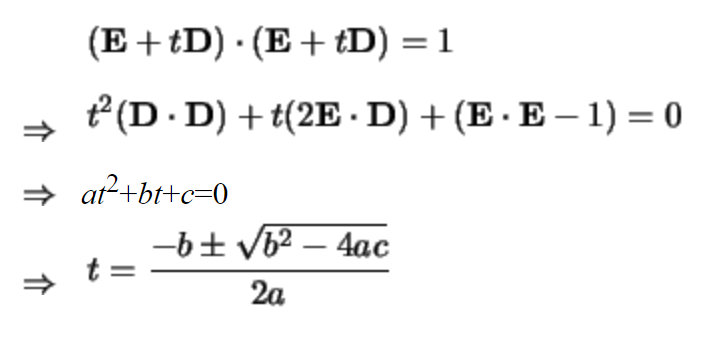
球体数学表示为



当光线与球体碰撞时有



向量表示为



根据此数学原理和公式，求光线与球体的碰撞交点步骤如下：

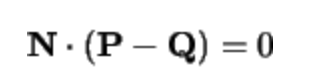
将光线方程带入球体方程计算**b\*b-4ac**与**0**的关系，是否存在合法t值；

取**t>0**中最小的t值为光源点到交点的距离；

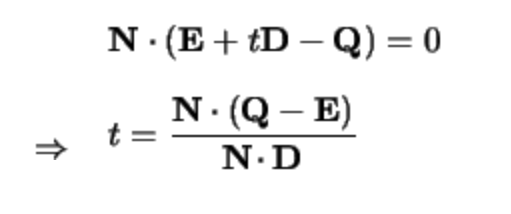
判断光线射出球体或射入球体，当t1,t2均大于0则光线射入球体；否则光线射出球体；

1. **平面**

一个平面可以使用其法向量和在平面上的点表示，其中P,Q为平面上的点，N为法向量



当光线与平面相交时有



根据此数学原理，求光线与平面的交点步骤如下：

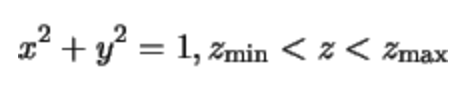
程序中平面具有中心原点O和法向量N，O点即为Q点，将光线方程带入平面方程；

在计算t值前要判断**N.D**是否为0，当其为0时光线与平面平行；计算t值，判断其与0的关系；t<0时平面位于光线的后面无法相交；t>0时存在相交点；

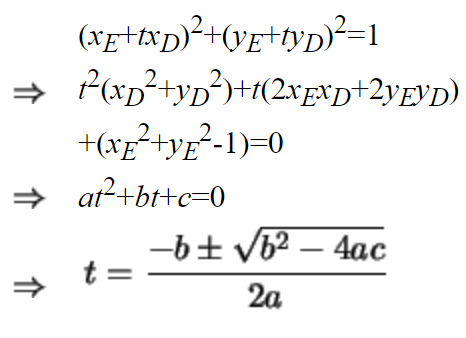
判断光线射入平面或射出平面，当**N.D>0**时光线射出平面；**N.D<0**时光线射入平面；

1. 圆柱体（以底面半径为1为例）

假设圆柱体的中轴为z轴，圆柱侧面表示为



当光线与圆柱侧面相交时有

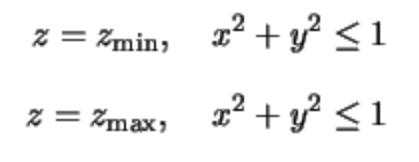


得到t值t1，t2，使用其计算出

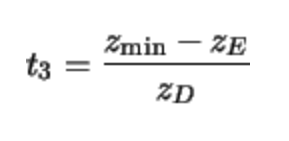
当交点在圆柱的侧面范围内时满足

取t>0中值最小的t值作为交点距离

圆柱底面表示为



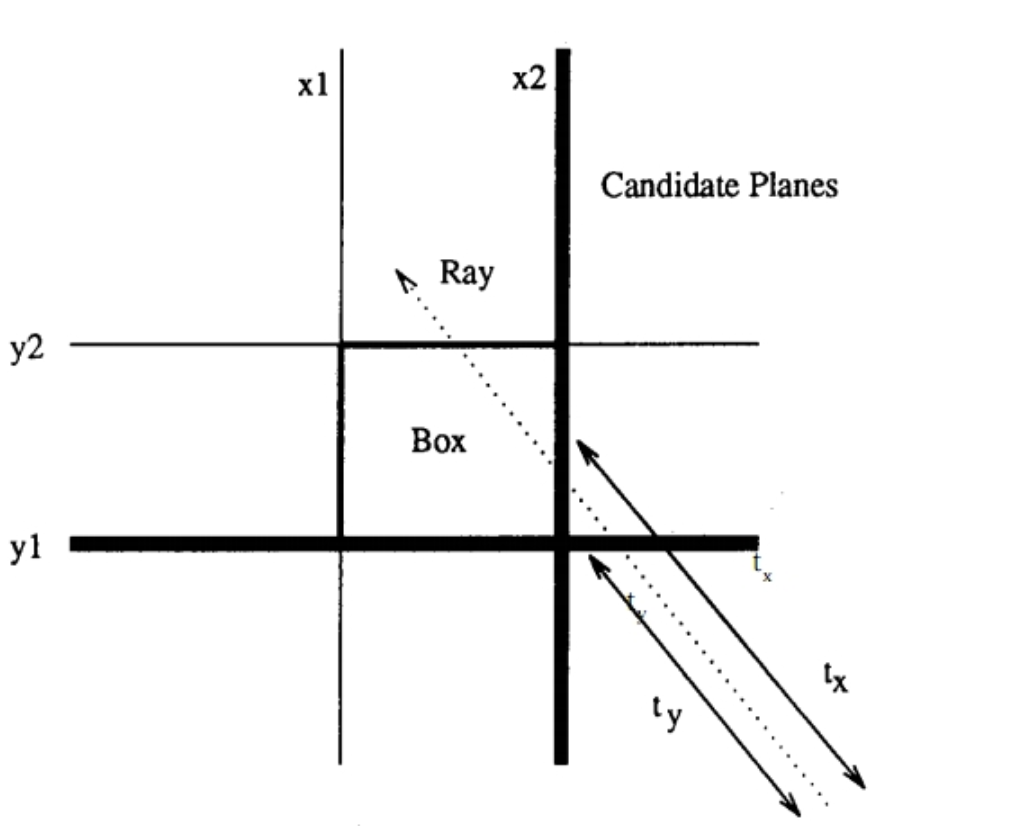
当光线与圆柱底面相交时有



使用计算出的t3求出交点P，设底面圆心为O，比较|P-O|与1的关系；

1. AABB盒算法

我们称一个正方体盒子为AABB盒，在3D空间中，先确定正对着射线的三个面，可以通过某种方式将盒子相对于射线ray的背面忽略掉，从而确定需要计算的三个候选面；（为了表示方便，我们使用2D平面来表示，同理可推导到3D空间）



在上图中，射线从右下角出发向左上角发射，所以候选面为y1，x2面；其中ty为射线到y1面的距离，tx为射线到x2面的距离，由图中可以看出，tx>ty，射线与x2平面相交，因此我们知道：**当射线与其中一个候选面相交时，射线ray的原点到这个面的距离要比到其他候选面的距离长**

光线在3D空间的数学表示为

为了求出光线与盒子的交点我们需要进行如下步骤：

* 确定候选面
* 确定候选面的方程表示
* 计算出t值，判断交点是否在盒子上

我们将盒子的范围使用盒子上最小点坐标Pmin（盒子的左下后方顶点），最大点坐标Pmax（盒子的右前上方顶点）表示，候选面的方程表示如下，其中P为面上的点，N为候选面的法向量，D为垂直于该面的轴向分量（例如正方体左面D为面上点的**P.x**）

因此当光线与盒子相交时我们有

候选面垂直于x轴的两个面时，

候选面垂直于y轴的两个面时

候选面垂直于z轴的两个面时

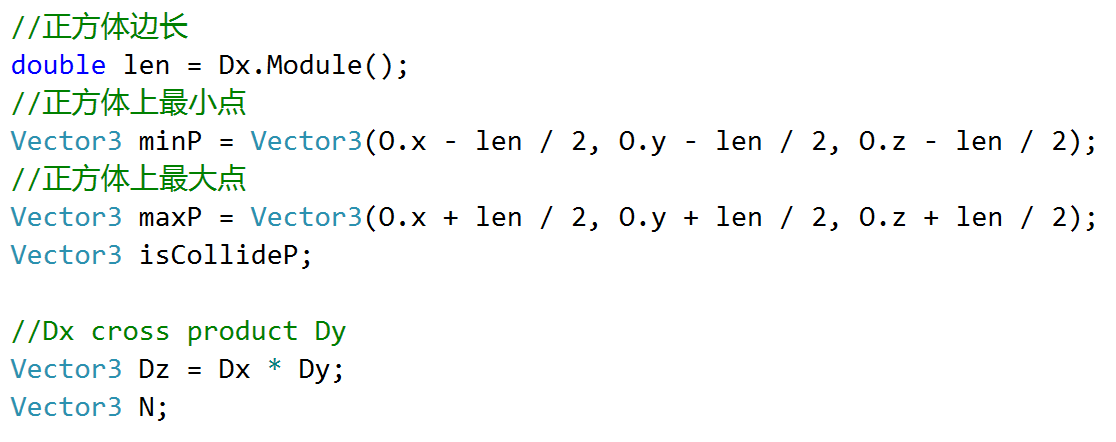
在求出t值后，还需要判断求出的交点是否在盒子范围内，设求出交点为I，有

1. 实现细节
2. 碰撞图元collidePrimitive的实现

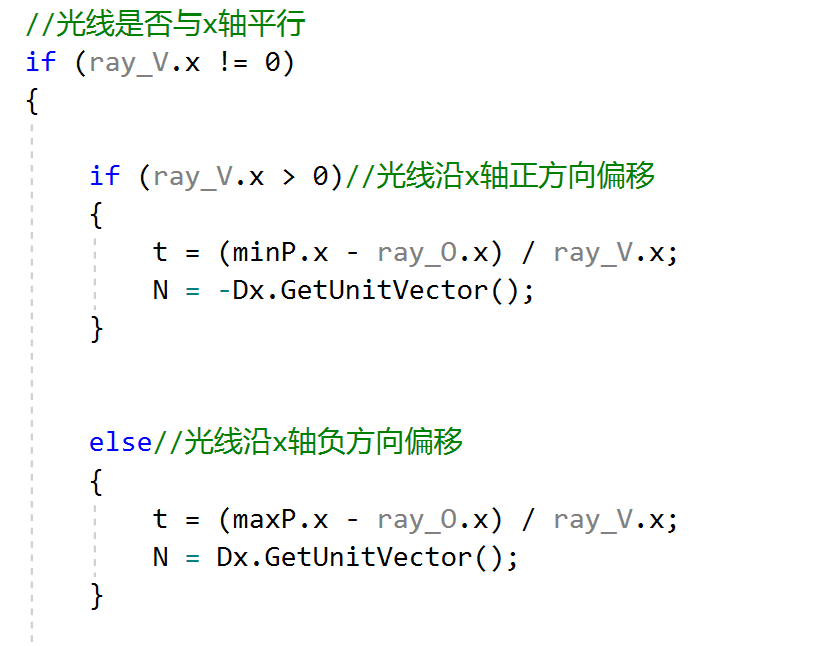
程序使用了老师提供的raytracing的代码基本框架，补充其功能。在原有功能上，已提供了基本碰撞检测图元球体和平面的计算实现，程序补充了正方体和圆柱体的碰撞图元实现。

* 正方体的图元实现类似于平面，六个平面组成一个正方体，在进行光线追踪时，需要对六个平面分别进行相应检测，寻找光线与正方体相交的最近距离交点；为了简化算法，进行光线碰撞检测时使用了AABB盒算法，通过光线方向和正方体坐标的条件组合判断，只需要计算最有可能与光线相交的面，减少了计算量。

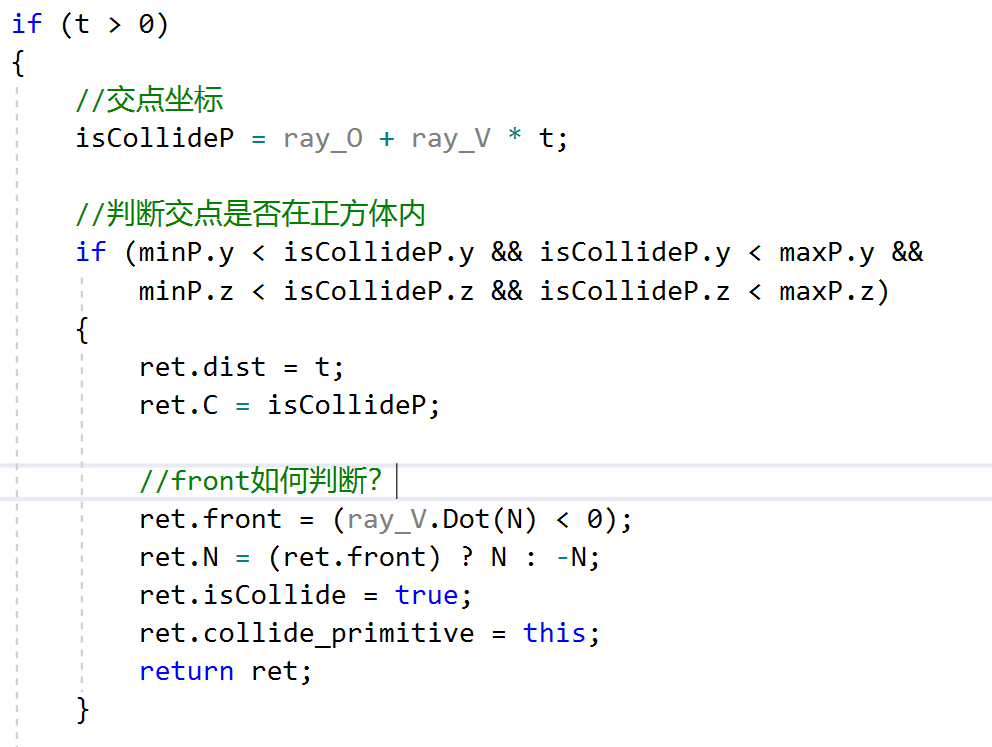
我们需要定义正方盒最小最大坐标点，使用叉乘求出z轴线向量



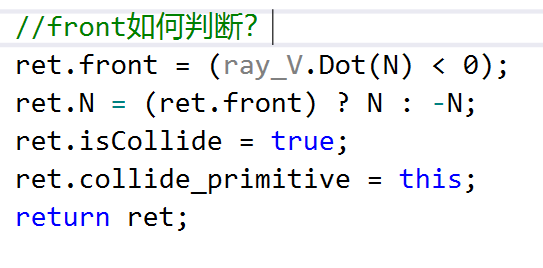
按照AABB盒算法，需要对光线方向的x，y，z分量分别进行检查，当某分量为0时，说明其与该轴平行，则不可能经过正方盒朝该轴向的两个面。计算平面交点时我们需要用到平面的法向量，其中朝向x轴分量的包围面法向量计算如图。首先检查x轴分量，当分量不为0比较光线x分量的朝向，朝向x>0则使用盒子最小坐标点计算；朝向x<0则使用盒子最大坐标点计算，同时记录平面的法向量



注意在AABB盒算法中，计算光线与平面的交点时使用沿轴向垂直的无限长平面公式，在计算出光源点到交点的距离后，判断距离与0的关系，当距离大于0时进一步计算出交点（距离小于0时表示平面在光源点的后面，光线无法与平面相交），并检查所求交点是否在计算平面的范围内。



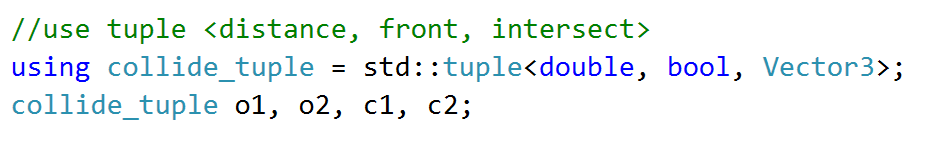
其中交点的前后面则通过判断光线方向与平面法向量的叉乘结果定义，当结果为负数时光线射入盒子，front为true，当结果为正数时光线射出盒子（光源点在盒子内），front为false；同时交点的法向量要结合交点的front值进行相应调整，当光线射入盒子时（front为true）交点法向量取平面法向量，当光线射出盒子时交点法向量取平面法向量负值



同理检测光线y轴，z轴的分量，按照x轴平面交点计算的数学原理实现其光线碰撞检测；只需要计算光线与正方盒相交的最近点即可。

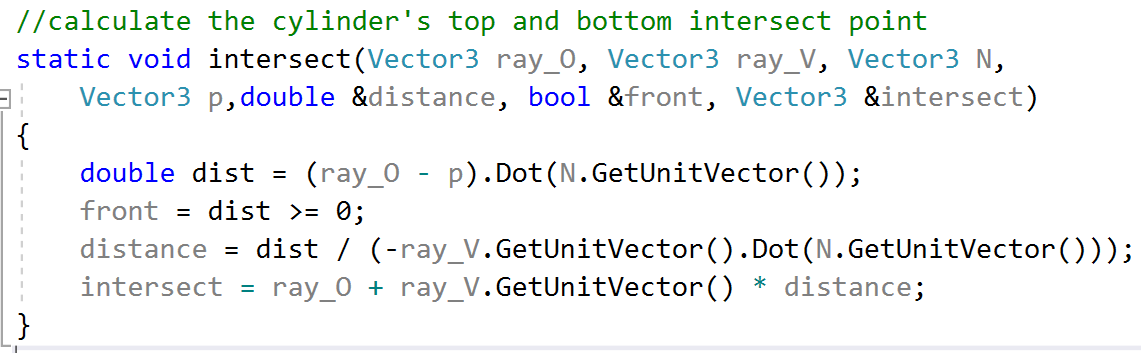
* 圆柱体的图元实现类似于球体，首先需要知道圆柱体的三维空间数学表示，结合光线的数学表示，结算出光线与圆柱体的相交点。要注意圆柱的侧面和上下底面的计算方式不同，因此需要分开计算；本程序的圆柱体图元实现借鉴了孟逊同学的代码，在此真诚感谢孟同学的耐心教导和无私的分享。

如图所示，代码中定义了数据结构元组，用来记录光线交点的参数，其中distance为交点到光源点的距离，front为交点的前后面，intersection为交点。设置四个元组o1, o2, c1, c2，分别表示圆柱的上下底面，侧面的两个交点的参数结构

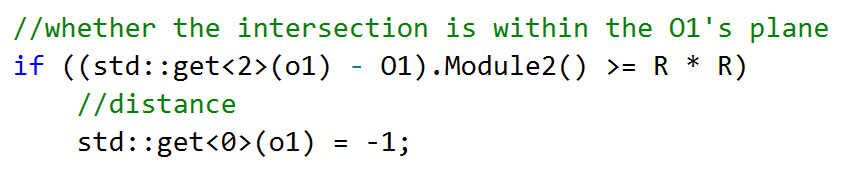


**圆柱的底面交点计算**：圆柱的侧面和上下底面计算方法不同，程序定义了函数intersect()用来计算圆柱上下底面的碰撞交点参数

对于底面的交点计算类似于平面的光线碰撞检测



Intersect()函数计算光线与平面的交点时使用的是圆柱的无限大底面，因此还应该对计算出的交点参数进行判断，判断出交点是否在圆柱底面的范围内。通过比较交点到圆柱底面圆心的距离是否在圆柱底面半径范围内来确定交点的合法性，若交点不合法，则将交点到光源点距离设为-1，用于之后取点的数据支持；使用同样的数学原理计算出上下底面的交点参数，并将数据保存至元组o1, o2 中

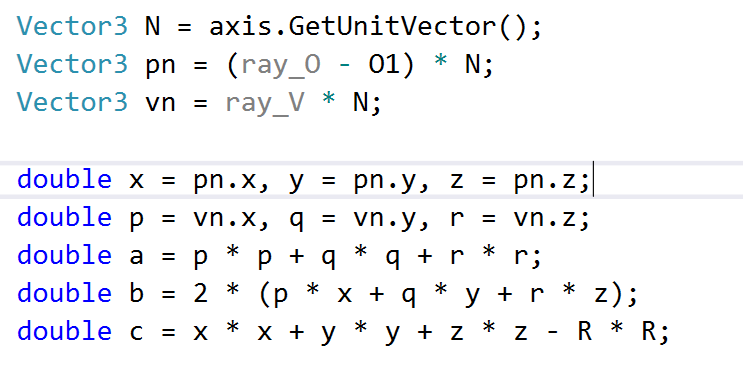


**圆柱的侧面交点计算**：圆柱的侧面交点计算类似于球体，有数学关系如下：

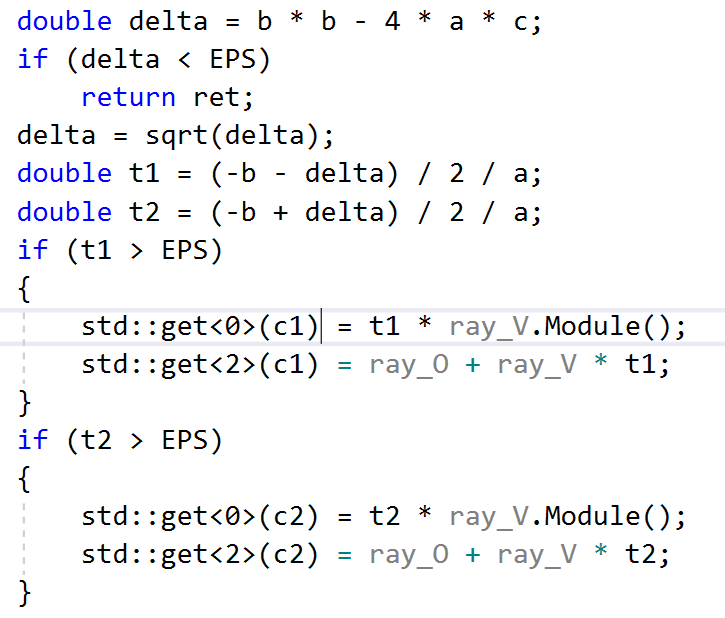
类似于球体的相交算法，求得

其中

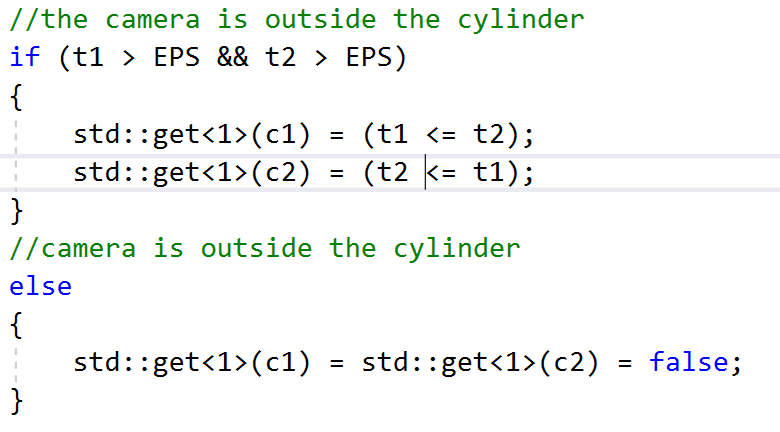
分别求出x, y, z, p, q, r



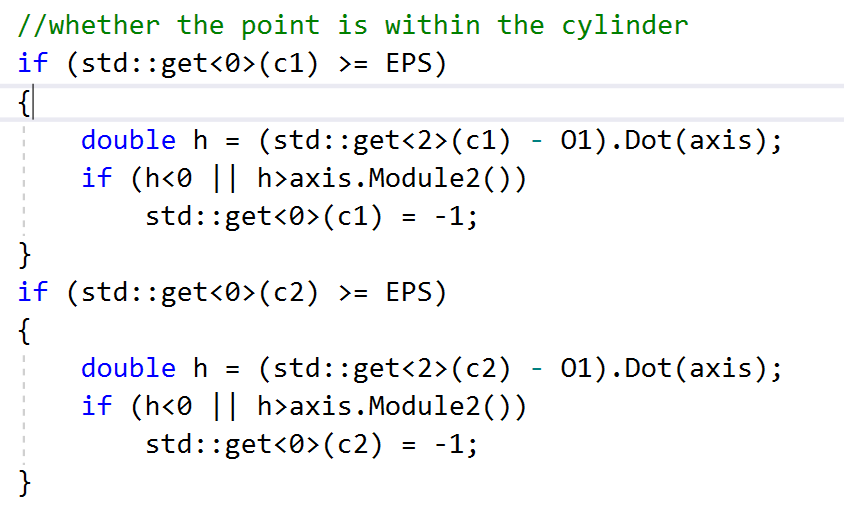
计算delta，当delta大于0时，计算侧面交点，与光源点的距离参数，，只取距离大于0的t值（t<0表示面在光线的后面无法相交）



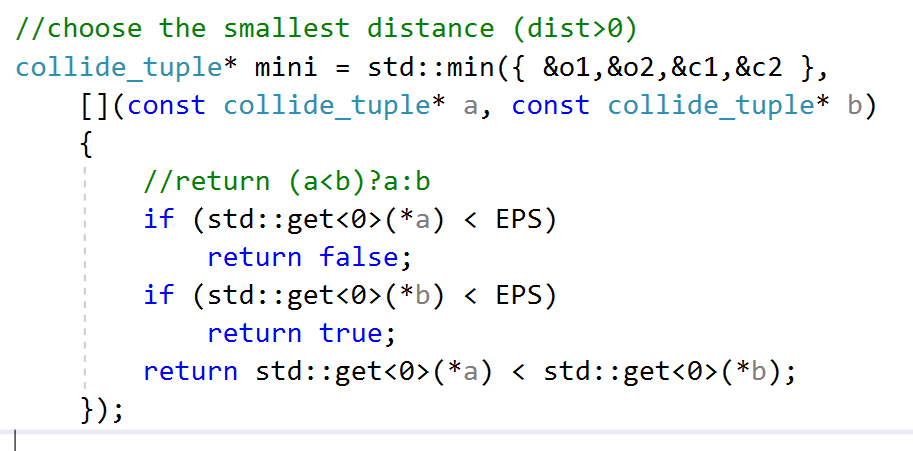
进行交点的前后面判断，当t1，t2值都大于0，表明光线从圆柱体外面射入，front为true，否则光线从圆柱体里射出，front为false



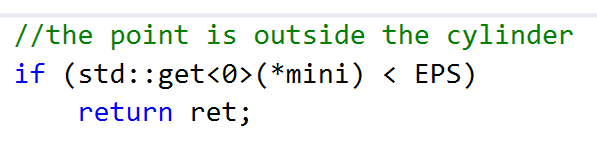
计算交点时使用了无限长的圆柱侧面进行计算，需要判断交点是否在圆柱范围内；计算交点在圆柱中轴的投影与圆柱中轴长度的关系，投影必须小于等于中轴长度交点才合法，当交点不合法时将距离参数设为-1



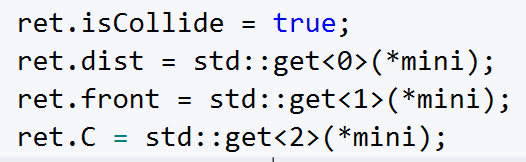
我们只需要取光线与圆柱体相交的最近点，使用标准库函数min取出四个元组中符合要求的交点的distance最小值，对应的交点即为最后所求的交点



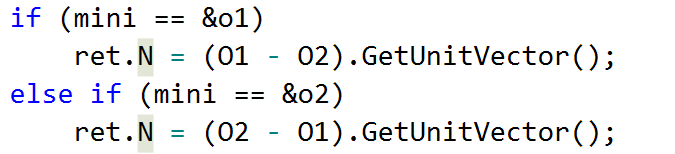
得到距离最小的交点后，还需要判断其交点的距离参数是否满足要求



最终光线交点的距离，front，交点参数使用元组即可得到



对于交点的法向量要分类讨论，上下底面的法向量如下



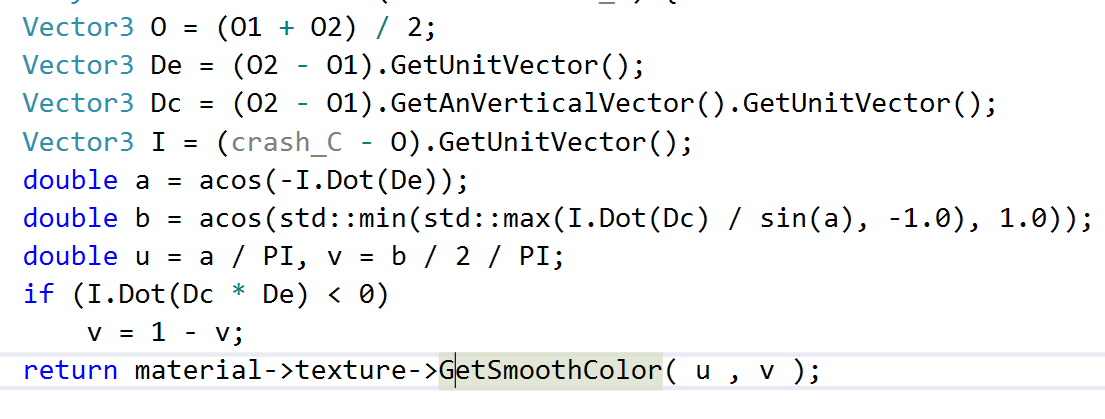
圆柱侧面的法向量如下：



1. 圆柱的纹理贴图

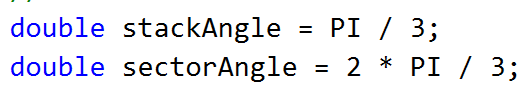
程序实现了圆柱的纹理贴图功能，使用圆柱的三维空间坐标计算出其二维纹理坐标映射，调用相应函数进行贴图。

圆柱的纹理坐标投影映射使用球体投影，纹理坐标计算和球体类似，其中De为球体的y轴向量基，Dc为x轴向量基；

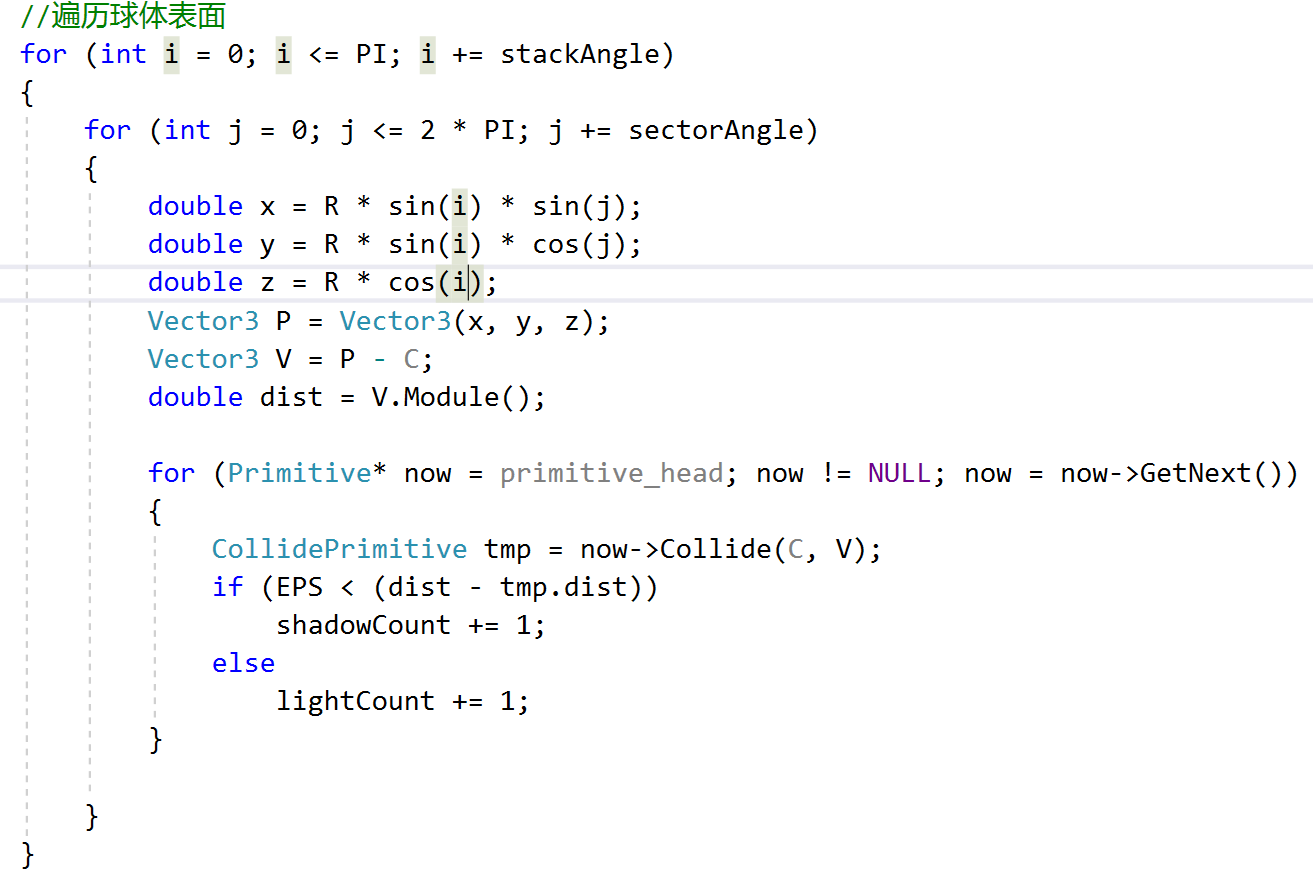


1. 面光源的实现

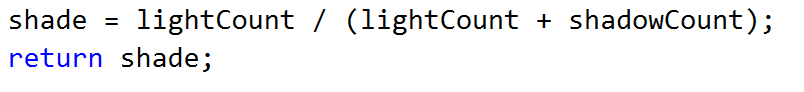
程序中的sphere light为会发光的球体表面，程序将球体按经纬度分块，将面光源分成点光源进行采样；考虑到电脑的计算性能，面光源的采样较少



遍历球体表面，对每一个采样使用点光源的计算阴影方法，记录每一个采样的阴影系数；被遮挡为0，不被遮挡为1

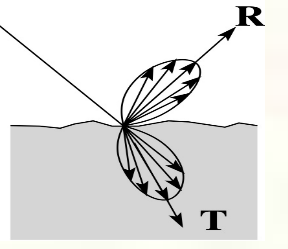


最终的阴影系数为不被遮挡的光线均匀分不到球体表面的值

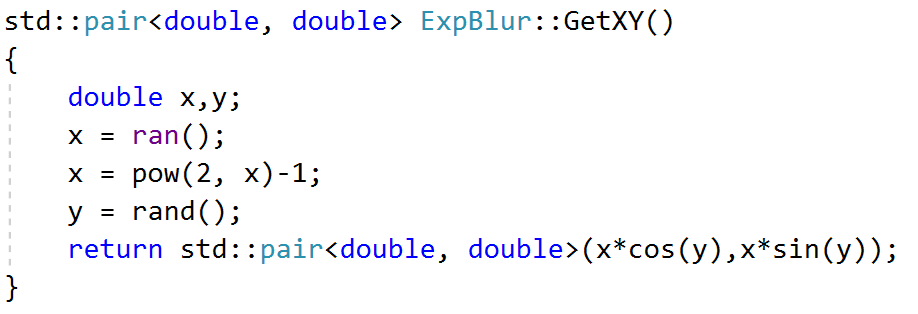


1. 反射的模糊效果

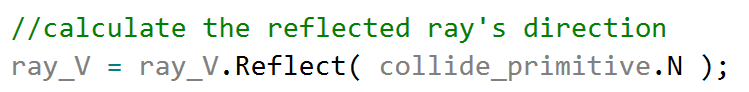
一般计算反射时都使用镜面反射效果来进行计算，但是并不是所有的物体都具有完全的镜面材质，所有进行光线反射追踪，需要实现反射的模糊效果。反射的模糊原理如下：



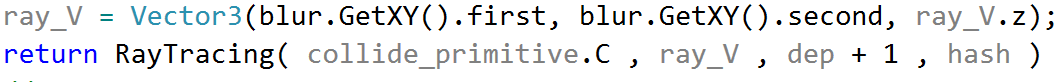
在镜面反射方向上的周围一定范围内任意选取反射向量作为反射的方向进行反射光线追踪；程序中的图元Blur提供了函数getXY()计算任意的反射方向



首先使用交点和reflect()函数计算出镜面反射方向



使用函数getXY()得到任意方向的反射向量并进行反射光线追踪



1. 结果讨论

在完成程序的过程中遇到了一些问题，通过上网查询资料，参考他人代码，对问题有了自己的理解和解决方案。

1. 圆柱体图元实现

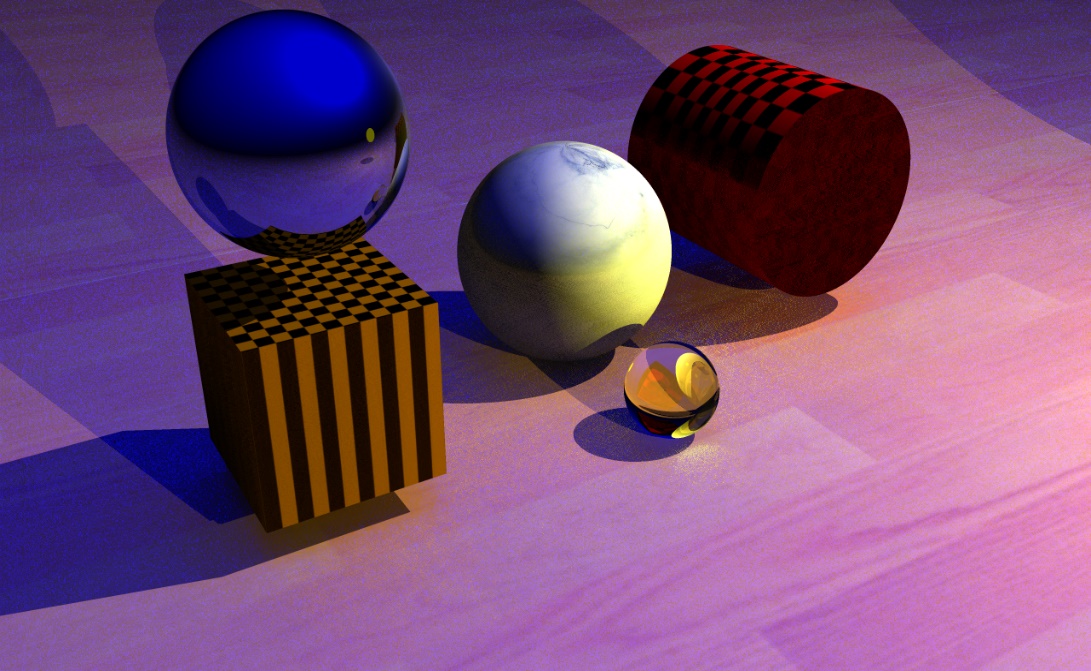
在初始阶段根据圆柱体与光线相交原理，理想化的认为圆柱体的中轴平行于x，y，z轴其中之一，使用此方法计算出的圆柱体畸形化，因为圆柱体的中轴可以是任意向量；在计算焦点，对于侧面和底面的分类讨论不正确，导致了呈现的图像十分奇怪；之后借鉴了孟逊同学的代码，认真阅读理解之后，自己尝试着实现了一遍

1. 面光源的实现

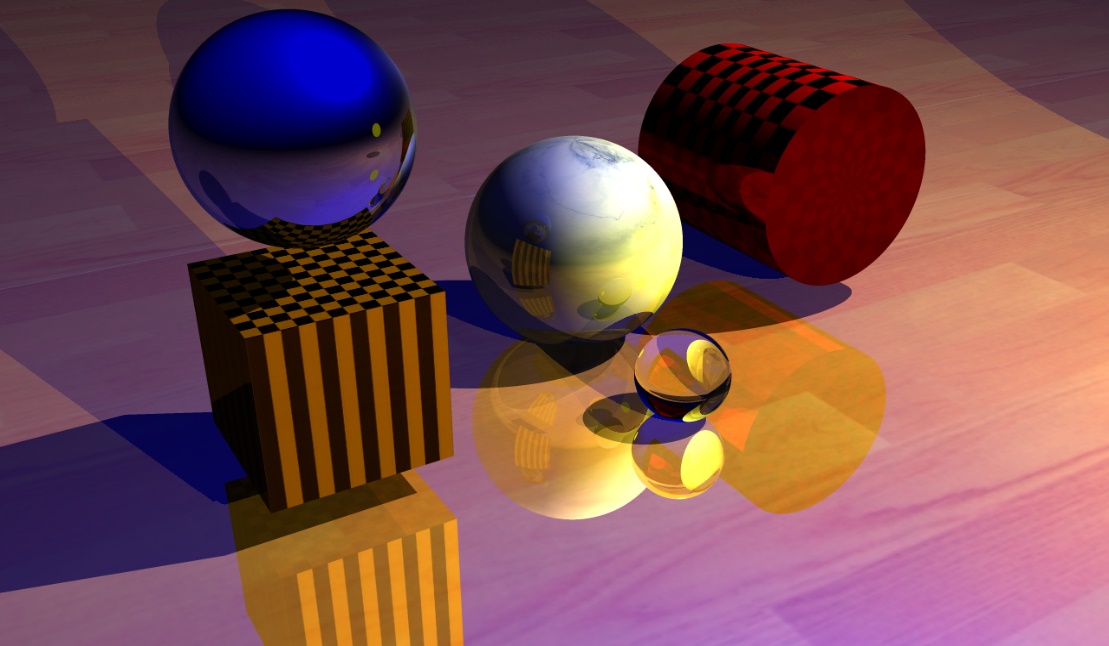
程序使用了均匀采样的方法对面光源进行遍历采样，将其当成点光源进行计算，最终得到阴影系数的平均值；测试时发现当面光源大小保持不变时，采样数越小效果最终成像效果越好，但计算越慢；当采样数不变时，面光源的面积越大，计算越慢；因此需要对采样数和面光源的面积进行这种考虑；程序中的采样算法尚不完善，还有改进和加速的空间，但鉴于时间有限，后期将查阅相关资料，继续完善面光源的阴影计算算法

1. 场景创建

如图为其中的一个场景创建图，该场景光源使用了点光源和球面光源，其中透明玻璃球可看到明显的光折射效果，而正方体和圆柱体的绘制形状正常，纹理贴图也正常，说明图元的实现算法较为成功；图中可以看到在某些图元表面有明显的噪点，这是反射模糊的效果呈现，



下图为未加反射模糊的效果，可以看到物体的光线反射效果十分刺眼，与真实场景中的效果相差极大



四．收获和建议

1. 课程收获

计算机图形学课程中，老师讲解清晰，实验课中代码原理实现细节讲解清楚。在课程中，老师会将实验中的基本实现步骤给出，对于本次作业有着很大的帮助。通过对老师的实验指导学习和本次实验的完成，我学习到了很多知识。

* 光线追踪的基本原理

为了完成该大作业的要求，查阅了许多网文资料和代码，了解了光线追踪的数学原理；要实现光线追踪首要是要计算出图元与光线的交点，将光线与图元的碰撞检测转化为三维空间中的数学公式，使用一元一次方程或一元二次方程计算出光源点和交点的距离，并判断交点合法性；每个基本图元都有其相应的数学计算原理；通过对图元交点的计算学习，复习了三维几何数学知识，交点合法性的判断增加了光线追踪的正确性；当计算出交点后，我们便可以使用Phong光照模型计算交点处的成像，其中需要计算漫反射，模糊反射，折射和环境光加成；模糊反射使得场景成像更加接近于真实世界的成像效果

1. 项目开发收获

通过本次实验，学习了基本图元光线检测，并将原理知识结合到实验中完成了程序要求的功能；在实验过程中收获颇丰

* 基本图元光线检测

通过查询外文资料，学习了三维空间中基本图元的数学表示，以及其与光线相交的计算方法；在求交点与光源点距离时可以转化为一元一次方程或一元二次方程求解t值；对于求解出的t值还需要判断其是否在图元范围内（计算焦点时使用的是无限长的图元面）；除了基本图元，其他物体都可以划分成基本图元用以计算光线碰撞；例如正方体的光线碰撞检测化为6个平面的光线碰撞检测计算；在计算交点时，三维空间中的图元位置是任意的，因此不可将图元的坐标轴基底理想化固定，应该将图元的数学表示一般化

1. 课程建议

* 希望每一次实验老师可以选取比较好的作品进行展示和讲解，这样可以学习他人好的代码思维，了解到更多的计算机图形学知识
* 希望老师对于下一节课的内容可以提前告知所讲范围，方便学生的预习