**光线追踪大作业报告**

完成了以下的功能：

1、基本图形的显示：球体，立方体，四面体。

2、光线反射，折射和投射。

3、软阴影。

4、平面纹理及球面高维纹理。

5、抗锯齿的尝试，但不理想。

**1、工程运行环境：**

运行环境：Ubuntu

编程环境：g++

所依赖的其他库：（编译时需要添加进VS的INCLUDE文件夹中）

OpenCV

编译指令：

g++ `pkg-config --cflags opencv` -o me my.cpp `pkg-config --libs opencv`

可执行文件me需在linux下运行。

**2、工程实现的简单说明：**

光线追踪：用了比较朴素的办法。通过一个相机类，由一个点和一个平面构成。对于屏幕上的每一个像素，从该相机点向平面上的对应位置打出一束光线。

如果遇到了镜面材质，则反射继续跟踪，但是反射光的大小要点乘表面的颜色值。

如果遇到了投射材质，则投射后继续往前跟踪。

如果遇到了漫反射材质，则跟踪停止，通过Phong模型来计算并返回该店的颜色值。

软阴影通过30个光源来实现。

平面纹理：先通过OpenCv读出每一个点的像素值，然后建立一个二维到二维的映射来完成。

球面纹理：先将原来的三维坐标转化为二维的球面坐标，再建立一个二维到二维的映射来完成。

抗锯齿的尝试：对于预设的采样值sams，对每一个像素点，将其分解成sams个子像素，然后对于子像素分别求像素值，最后取平均值来作为大像素的颜色值。

代码结构：

Vec类用于保存一个3元数组，用于表示点坐标和Rgb颜色等。

Ray类表示光线，由一个位置Vec和方向Vec组成。

Camera类模拟一个相机，有焦点和方向。

Basic\_obj类表示一个物体的基类，其中有虚函数求交intersect()和求法向量Normal\_V()

Sphere类，Plane类，Cube类，Tetrahedron类都继承Basic\_obj类，分别表示球，平面，立方体和四面体。

Vec Lights表示光源，

Basic\_obj things[]中包含了场景中的所有物体及其生成的生成参数。

radiance()函数为跟踪某一条光线的颜色值，该函数内部实现了求反射、折射、纹理和Phong模型等。

代码的结构和部分功能参考了Kevin Beason的smallpt资料，网址www.kevinbeason.com/smallpt/

**4、运行的效果：**

运行的效果已附在同级文件夹中。

**对结果图像的说明：**

图中的房间由六个面组成，其中背后的面设为黑色。前面的面的表面性质为镜面，地面为纹理处理。

最左边的球为投射材质，在恰当的位置作用时一个凸透镜，看到了倒立的像。

右边的球为高维的纹理，纹理的模糊是由原图的大小导致的。

四面体和立方体的表面材质均为漫反射。

上面的”灯”只是一个球露出下面的一小部分，并不是光源。