计算机图形学基础 光线追踪作业 算法说明

计算机系 计 15 刘晨

一、 提交文件目录

--bin 可执行文件

--src Visual Studio 2010 项目文件

--doc 说明

注：有三个纹理图片，在 VS2010 中必须与源代码同目录，单独运行可执行文件时，必须与可执行文件同目录。

二、 基本类的说明（详细说明见代码中的注释）

1. 颜色 colour 类：
   用三个整数表示 RGB 颜色，定义了颜色的加减乘除运算。
2. 向量 vec3f 类：
   该类定义了一个三维向量的模型，重载了基于三维向量的加减乘除法，定义了点积和叉积，定义了向量模的计算和向量标准化的过程。
3. 射线 line 类：
   射线由三维空间的一组点和一个单位化的标准向量组成，定义 getpoint 函数获得距离起点一定距离的点的坐标。
4. 光学属性 lproperty 类：
   该类用于表示由物体本身决定的光学属性，具体的参数意义如下表：

成员变量名	物理意义
rate	物体内部的折射率
throughrate	折射光强占总光强的比例
reflectrate	反射光强占总光强的比例
mirrorrate	反射光中镜面反射光强的比例
highlightrate	反射光中高光反射光强的比例
s	高光

5. 纹理 grain 类：
   该类主要提供物体表面的纹理信息，提供一个共同的接口 mapping，用于将物体坐标平面上的一个点映射到一个颜色上。
   grain 有如下几个派生类：
   - single_grain:单一色调纹理

   此纹理下的物体为单一色调，即 mapping 函数将物体上所有的点都映射到同一种颜色上。
   - line_grain:双色条状纹理

   paint 为此纹理提供了基本的颜色映射库，在 paint 中颜色的种类大于等于 2 时，按照所取点的三维坐标值的和决定取哪种颜色。所生成的纹理为双色条状，参数 interval 用于刻画条带的宽度。
   - cube_grain:格子纹理

   与 line_grain 的实现机理相似，cube_grain 将空间中的点映射到两种颜色中，形成格子的视觉效果，参数 interval 反映了格子边长的大小。

pic_grain:二维图片纹理

pic_grain 从外界读入图片 file.jpg，将空间中的一点沿着 xy、yz 或 xz 平面投影，所返回的颜色值就是 file.jpg 中投影点的颜色。设有参数 filename 表示文件的名字，direction 表示投影操作的方向。

- high_grain:高维纹理

该纹理方法与 pic_grain 方法相似，不过将三维的一个点映射到二维的一个点的方法不是投影，而是借助一个 2*3 的矩阵。pic_grain 是 high_grain 的一种特殊形式，high_grain 能够实现在不平坦的三维图形上刻画连续纹理的效果，也能实现 pic_grain 不能实现的缩放的效果。

6. 物体 obj 类：

这是表示物体的基类，包含有物体的纹理和光学属性信息。保留光线与物体表面相交求交点的接口，返回值为一个 pair<double,vec3f>类型，表示光线发出点到交点的距离，以及交点处法线的坐标。

7. 平面 plain 类：

物体 obj 类的派生类，用 a,b,c,d 四个双精度的浮点数表示其位置，代表平面的方程为 ax+by+cz+d=0。支持两种构造方式：以解析式的方式构造和给出平面中一个点和两个不共线的方向向量的方式构造。射线与平面求交点的计算采用解析式带入计算。

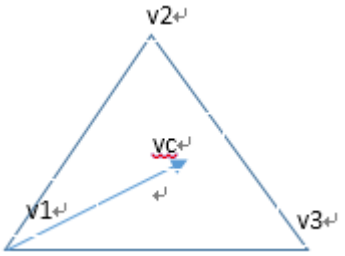
8. 球 ball 类：

物体 obj 的派生类，用球心和半径的值加以刻画，计算交点的时候将射线参数化，代入球的解析式中加以计算。

9. 多面体 poly 类：

物体 obj 的派生类，采用三角面片的表示方法，借助 tri 类和 tri_num 类实现，前者表示三维空间的一个三角形，后者表示点的索引。对多边形的每一个三角形进行求交计算，最后取距离最小的正数表示交点。

在空间三角形的求交计算中，先利用三角形所在平面的解析式求出其所在平面以射线的交点，再判断该交点是否在三角形内。判断方法是，将空间三角形投影到 xy，xz 或者 yz 平面上，确保其不退化为线段(如右图所示)。将 vc 投影到(v2-v1),(v3-v1)上，分解式为 vc=w1(v2-v1)+w2(v3-v1).有几何关系可知，当 w1,w2>0 且满足 w1+w2<1 时，vc 在三角形 v1v2v3 中，其余情况 vc 均落在三角形 v1v2v3 外。



由于多面体采用三角面片的方式存储和计算，所以 poly 类支持 obj 文件的读入。对于每一个多面体对象，我们通过确定该对象中所有点的 x,y,z 坐标中的最大值和最小值来给对象套一个外接的长方体，如果光线与该长方体不相交，则光线就不会与该物体相交；如果光线与长方体相交，再考虑光线与所有平面相交的情况

10. 场景 scene 类：

该类用于刻画一个三维的场景，支持多物体和多光源，其成员函数和成员变量的意义如下：

成员变量或函数名	意义
std::vector<vec3f> light	各光源的位置
std::vector<obj*> object	场景中的各物体
std::vector<double> weight	场景中各光源的对应强度
double overreflect	漫反射率

scene(double reflect)	构造函数，指定这个空间的漫反射率
void addlight(vec3f pos, double w)	加入一个光源，指定其强度(最终计算光强时候，会对光源强度归一化)
void addobj(obj)	加入一个物体

11. 照相机 camera 类：

该类最终完成光线跟踪的绘制和显示，包含了画布、视点，场景等信息。camera 类不决定三维坐标的，三维坐标不取决于任意一个类，它是独立存在的。因此 camera 类支持视角和画布的平移和旋转操作，支持景深效果的绘制，其成员函数和成员变量的意义如下表所示：

成员函数或成员变量	意义和说明
int height, width	画布的高度和宽度
int dis	视点距离画布中心的距离
int focus	景深的焦点，负数表示不考虑景深
double **depth	画布中所显示的各物体距离画布的距离，-1 表示该像素为空白
colour **buffer	第一轮光线跟踪存储的各点的颜色(留给后面处理景深时使用)
vec3f view	视点的位置
vec3f stpoint	画布的左上顶点坐标
vec3f xline, yline	画布中表示横纵坐标方向的单位向量
cv::Mat img	画布
camera(int w ,int h ,int d ,int f)	初始化画布的长宽，视点距离话画布的距离，景深效果焦点的值
void move(vec3f m)	平移画布和视点，平移向量为 m
void x_rotate(double alpha) void y_rotate(double alpha)	沿着 x 方向或 y 方向画布中点进行旋转画布，向左向上为正
void draw(scene& s)	绘制画布，场景为 s
void show()	显示画布

三、 一些算法：

1. 光线跟踪算法

在不考虑景深的情况下，物体某一点的光照强度表达式如下：

颜色=本身颜色*漫反射系数+本身颜色*(1-环境光系数)(吸收光强比例*(镜面反射光+高光效果^高光系数)+反射光强比例*(迭代反射光线)+折射光强比例*(迭代折射光线))

上式中，吸收光强比例+反射光强比例+折射光强比例=1。漫反射系数由场景类决定，其余的系数均由相交物体本身的光学属性决定，物体本身的颜色有相交点的坐标通过纹理映射决定。由于反射和折射的过程需要迭代，为了减小计算量，当反射和折射的强度小于一定值的时候就会停止迭代，在本算法中，光强的下限值定为 0.05.

2. 景深模糊算法

为了制造景深的效果，我们定义了一个距离与焦距差距衡量的指数：

f(x)=max{x/focus ,focus/x}

由此可见，与 focus 相差越远的点，其 f(x)的值越大，该值表示该点的模糊程度，我

们以该点为中心，在横纵坐标不相差 f(x)的前提下并且这些点距离视点的距离差小于一定范围时，任取 30 个点，这三十个点的颜色的平均值作为该点的最终颜色。

3. 一些抗噪音、抗锯齿的方法

为了避免 double 类型在精确计算中存在的一些误差，我们在需要比较的地方都设立了小量值，例如将 double 类型的变量与 0 比较，变为与 1e-6 比较。针对 double 转化为 int 类型的变量可能存在"去尾误差"，我们将正 double 量加上一个小量在转化，将负 double 量减去一个小量再转化。

采用采样，将整形离散的像素点分成 1/2 乃至 1/4 单位，可以起到更好的抗锯齿效果，不过这样做的话，会使程序的运行时间加倍。

四、 实现的效果：

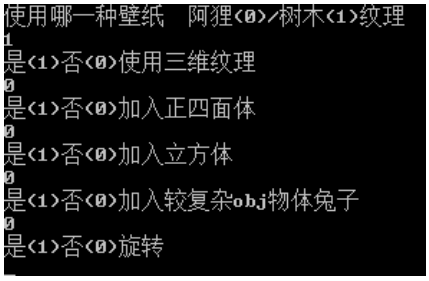
obj 类实现了球形、正方体和平面多面体的绘制。

camera 类实现了 phong 模型绘制、景深效果和多光源的软阴影，实现了镜头的旋转。

grain 类实现了光的反射、折射，达到了镜面和透明的效果，通过纹理映射实现了简单的纹理和高维纹理。

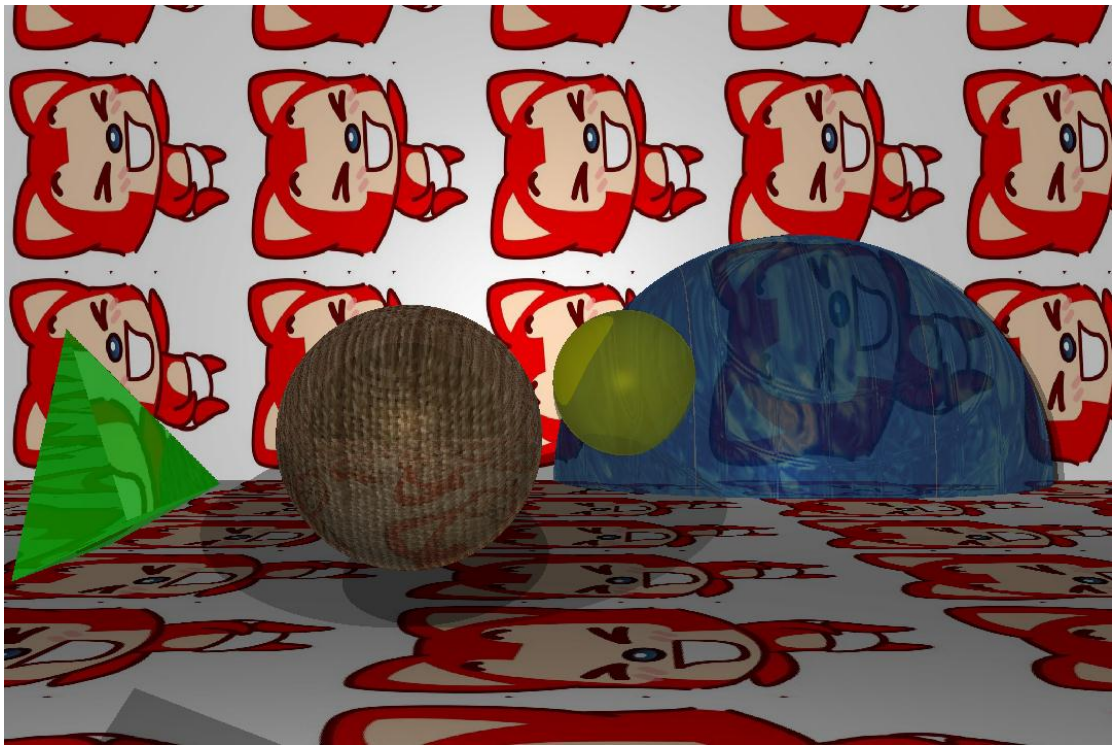
支持多面体 obj 文件的读入。

支持选择：

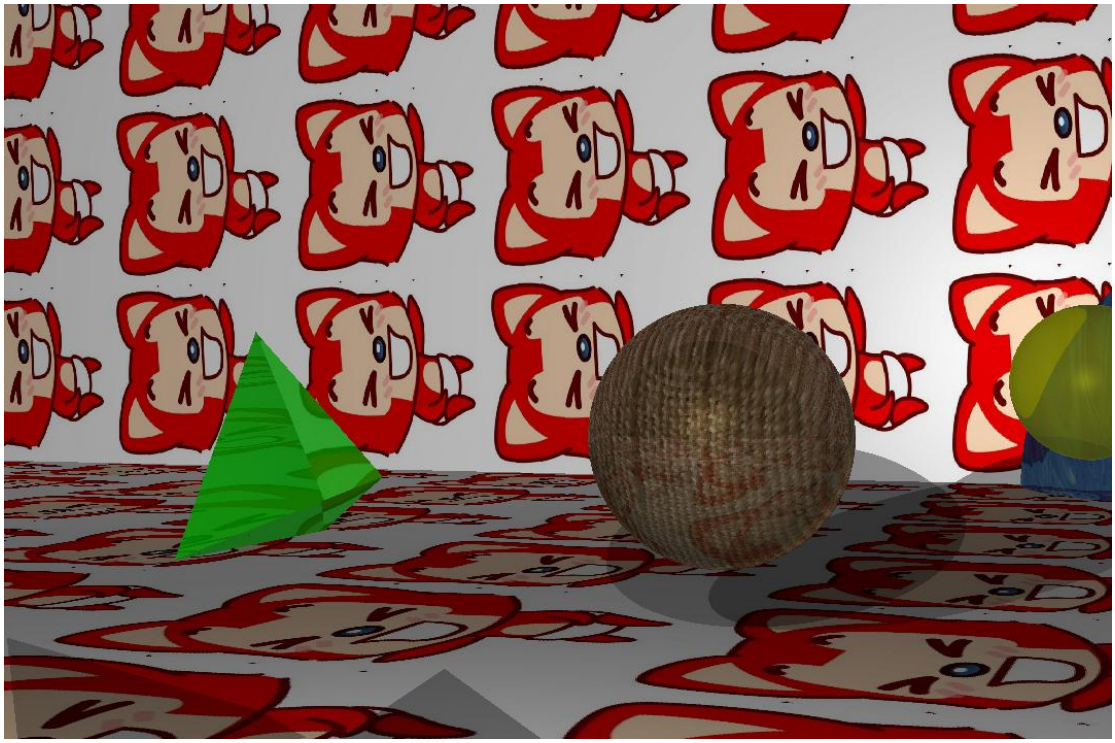


下面是绘制出的几幅图

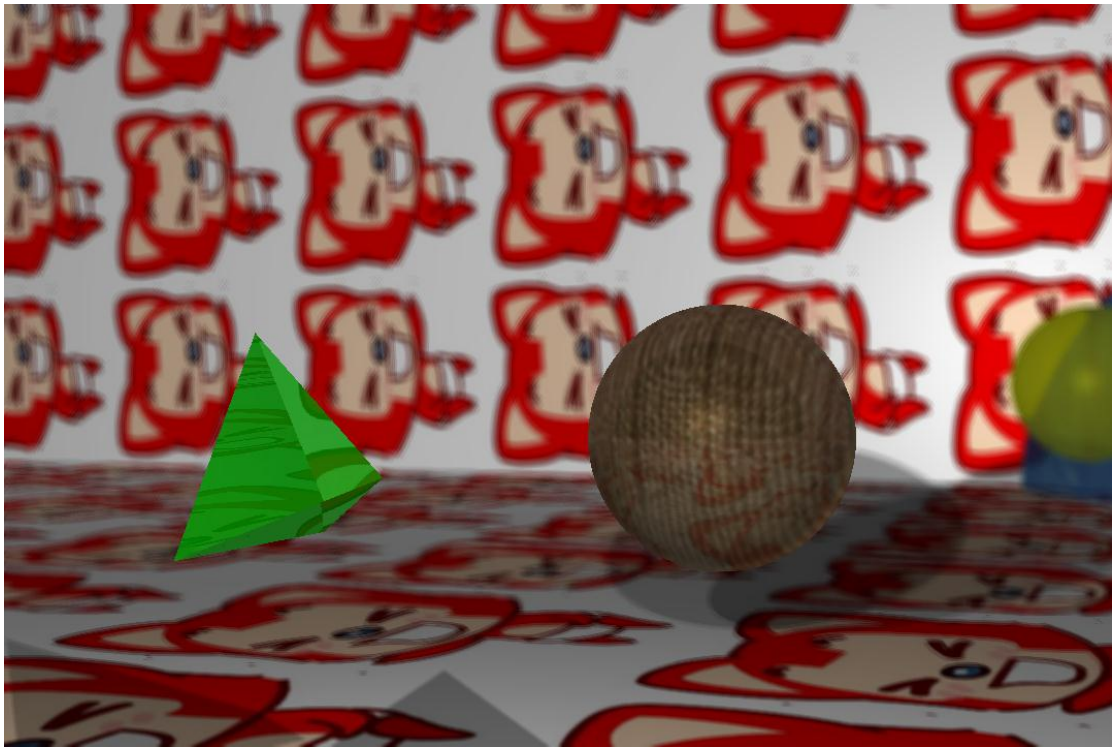
正面（本影半影，平面纹理，三维纹理，反射折射，phong 模型，球和多面体）



旋转 20 度之后：



加上景深效果之后(焦点对准绿色透明物)：



加上 obj 文件描述的复杂多面体之后（左上角的兔子）