

# 计算机图形学基础 PA0

## 实验报告

计 97 郭昊 2019013292

2021 年 4 月 16 日

## 1 光线投射

### 1.1 光线与球面求交

代码位于 `sphere.cpp` 中。在这里实现了用几何法进行球面的求交，基本思路与讲义类似，即先求截面中垂线与光线的交点，再视光线位置求出真正的交点。其中需要注意的是，由于 `transform` 类的存在，光线方向向量不一定为单位长度，因此需要单位化。而且，当视点位于球内时，对应的法向量方向不应为交点减去球心的方向，而应相反。

### 1.2 光线与平面求交

代码位于 `plane.cpp` 中，思路也与讲义类似，即求解方程组

$$\begin{aligned}P(t) &= R_o + tR_d \\ nP(t) + D &= 0\end{aligned}$$

需要注意的是这里的平面表示为  $ax + by + cz = d$ ，而非讲义中  $ax + by + cz + d = 0$ ，故  $d$  的符号相反。

### 1.3 光线与三角形求交

代码位于 `triangle.cpp` 中。在三角形的构造函数中，需要利用两边对应的方向向量（可以利用两点向量相减）叉乘计算出三角形的法向量。

求交思路与讲义类似，即求解线性方程组

$$R_o + tR_d = (1 - \beta - \gamma)P_0 + \beta P_1 + \gamma P_2$$

判断边界条件后，即可得  $t$  的值。

这里可能需要注意的是需要判断一下分母行列式是否为 0（或者接近 0），接近 0 则一定不会有交点（相当于线性相关，光线与三角形所在面平行），直接返回 `false` 即可，否则可能会导致浮点数错误（虽然在测例中没有体现出来）。

## 1.4 Group 类的实现

代码位于 `group.cpp`。这里采用了一个给定长度的 `vector` 作为容器，并且在 `intersect` 中对于每一个 `Object` 的相交返回值作“或”处理，以得到光线是否与这组物体相交。

## 1.5 Camera 类的实现

代码位于 `camera.hpp`。基本思路同介绍文档中类似，但需要推导一个公式，即  $f_x$  与  $f_y$  的计算。由相似三角形的方法，假设图片宽为  $w$ ，高为  $h$ ，视角为  $\theta$ ，可以得到

$$f_x = \frac{w}{2 \tan(\theta/2)}$$
$$f_y = \frac{h}{2 \tan(\theta/2)}$$

这里也应注意向量的单位化。

## 1.6 Phong 模型实现

代码位于 `material.hpp`。基本思路同介绍文档一致，但需要注意的是：在光线与反射方向相反时，我们才能看到最多的反射光（参考镜子），因此在计算 `specularColor` 的系数时，需要将正常点积得到的值加负号。

# 2 遇到问题

1. 相机那里理解较为困难，因为只有一个 `angle`，故在未上习题课时只能猜测画布为正方形，`angle` 同时适用于两个轴向，这里导致了编程时的

时间浪费。

2. 在求交时如果没有更新  $t$ , 必须返回 `false`, 否则会导致 `Runtime Error`。

剩余问题均在实现部分“需要注意”内提到。

### 3 讨论与借鉴

在完成本次作业时, 未与同学进行任何讨论, 并且只借鉴了讲义及介绍文档的思想, 代码均为原创。

### 4 使用说明

只需在项目根目录执行 `./run_all.sh`, 即可在 `output` 目录下输出测例的渲染结果。

### 5 未解决的问题

暂无, 如果仍有更多时间, 那么我可能会构造更多边界情况的测例(如光线与三角形平行)等进行测试。

### 6 建议

对于相机部分, 建议采取更加细化的描述, 如“该角度同时为水平和竖直方向的角度”。而且图像较易引起歧义, 可以改为平面图。