

README

王永浩 2016011420

Contents

1	光线投射算法逻辑及遇到的问题	2
1.1	光线投射算法	2
1.2	遇到的问题	2
1.2.1	测试文件中的平面参数描述	2
1.2.2	Phong 模型着色	2
1.3	generateRay 函数的实现	3
2	编译代码方式	3

1 光线投射算法逻辑及遇到的问题

1.1 光线投射算法

本次的光线投射整体的框架代码直接复用了说明文档中给出的实例代码，整体代码逻辑如图 1所示。

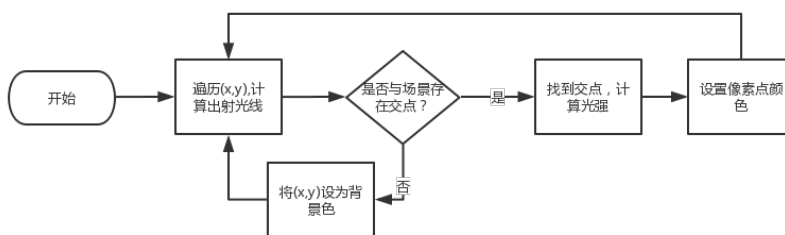


图 1. 代码逻辑

整体并没有对给出的逻辑做太大改动，需要添加的地方包括：**球、平面、三角面片的求交方法、实现 PerspectiveCamera 类和计算投射光线，实现 Shade (计算颜色) 方法**，在实现过程中并未和同学进行相关讨论也未参考其他代码。

1.2 遇到的问题

1.2.1 测试文件中的平面参数描述

给出的测试文件中平面的参数描述包括两个：**法向量: $normal$** 和 **$offset$** 。我们知道平面的方程一般写作 $\vec{n} \cdot \vec{p} + d = 0$ ，所以自然而然的一般人会把给出的 **$offset$** 参数当做公式中的 d ，然后根据求交点的公式 $t = -\frac{d + \vec{n} \cdot \vec{R}_0}{\vec{n} \cdot \vec{R}_d}$ 求取相应交点进行后续处理。而如果这样做的话，生成的图片会出现白色区域在上黑色区域在下这样一种与正确图片相反的情况，检查之后发现应该将 **$offset$** 当做公式中的 $-d$ 计算才能得到正确的结果，助教对这一点应在文档中加以说明。

1.2.2 Phong 模型着色

我在实现的时候对文档中给出的 **Phong 模型着色公式**

$$I_{pixel} = c_{ambient} * K_a + \sum_i (k_d clamp(\vec{L}_i \cdot \vec{N}) + k_s clamp(\vec{V} \cdot \vec{R}_i)^s)$$

中凭空出现的 **$clamp$** 实在无法理解，起初我并没有管这一项结果生成的图片始终出现一些诡异的高光，后来查阅课本知道里面的点积实际代表相应夹角的余弦值且其夹角都在 $(0, \pi/2)$ 之间，并且在网上查阅 **$clamp$** 之后才知道它在 C++ 中是一个限制取值范围的函数才明白 **$clamp$** 实际是在约束里面的点积结果在 $(0, 1)$ 之间，之后对代码进行了修改才达到正确结果。我深感助教应在文档中给

出相应的说明，对于没有相应经验的人来说既然在文档中给出了相应的公式就应该对每一项都有相应的说明，否则就直接让学生查阅课本及相应资料，给出一个不知所云的符号只是徒然增加难度。

1.3 generateRay 函数的实现

我在此处卡了两天，因为框架代码中提供的参数并没有文档中所描述的 (f_x, f_y, c_x, c_y) 这四个内参，而是给了一个 *angle* 参数，而这个参数又没有在文档中有详细的说明，只是在图上标了一下，但是标的又不清晰，给人的感觉这个 *angle* 参数是图片左上角与相机位置连线构成的向量和图片左下角与相机位置构成的向量的夹角，我试图通过列出上述两个向量形成夹角的表达式来找到通过 *angle* 得到四个内参的方法，最后发现它是一个欠定的问题。后来我看了 PA2 的代码才知道这个 *angle* 参数是相机与照片平面上下两边中点连线形成的夹角，而且默认相机位于照片平面中点， $f_x = f_y$ ，这样的默认对于做过实际摄像头标定的我来说是不可接受的，或许这在图形学领域中是众所周知的，但是作为面向初学者的文档来说就应该将这些约束情况明确的写出来，否则只是平白增加学习成本。

2 编译代码方式

本次作业已按照要求的文件树结构进行组织，直接在 code 目录下运行 `..\run_all.sh` 即可编译运行。