PA1 · 光线投射实验报告

计01　容逸朗　2020010869

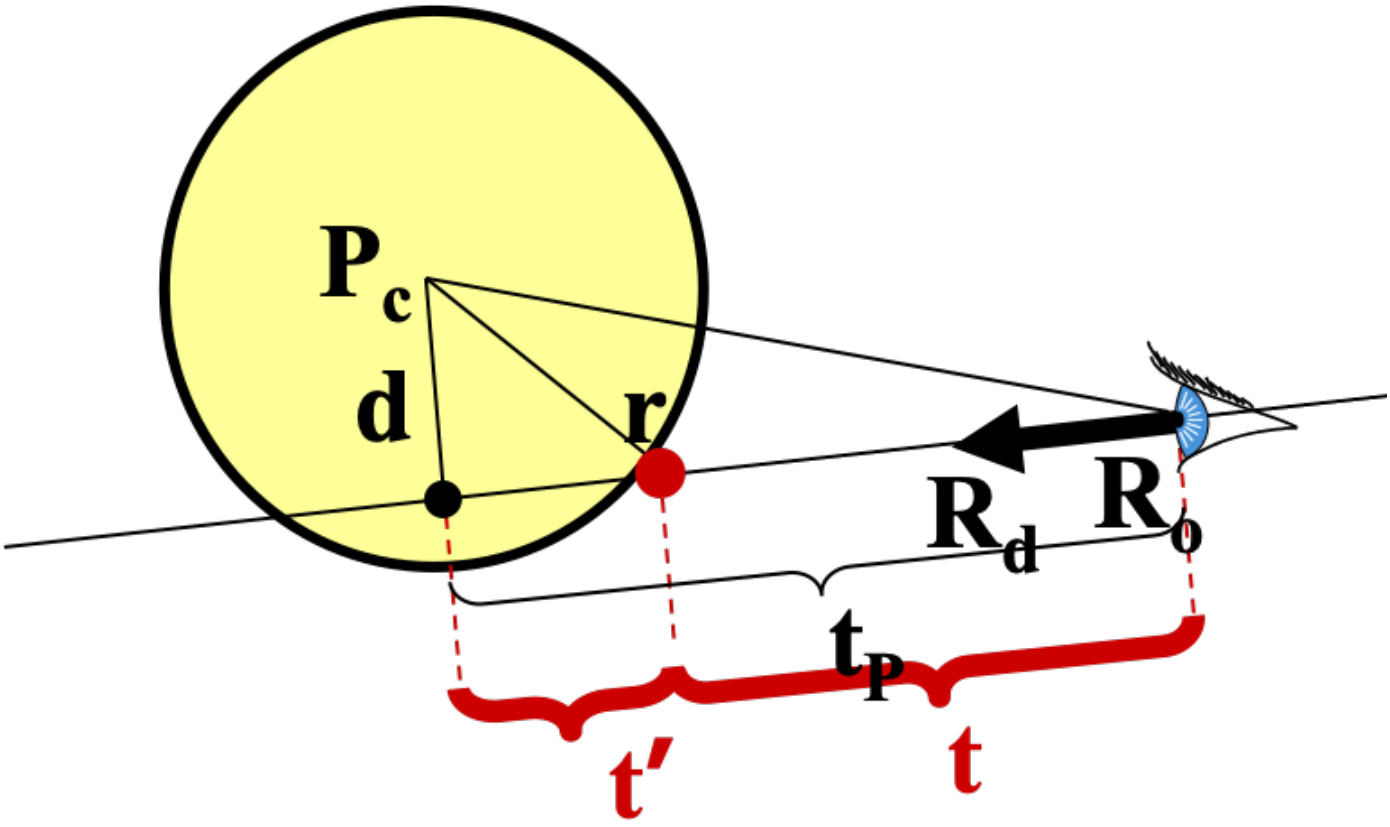
1　光线求交

光线求交的目标是计算光线 $P(t) = R_O + tR_d$ 与场景内几何体的交点。

1.1　球面求交

1.1.1　思路

利用几何方法，可以直接求得交点位置及该点的法向量。



1.1.2　算法

1. 计算光源指向球心的向量 $\vec{l} = P_c - R_0$
   (a)　$\vec{l}^2 < r^2$ ： 光源位于球体内部；
   (b)　$\vec{l}^2 > r^2$ ： 光源位于球体内部；
   (c)　$\vec{l}^2 = r^2$ ： 光源位于球面上；
2. 计算球心到光线所在直线的投影点 (垂足)： $t_p = \vec{l} \cdot R_d$

   如果光源在球体外部或球面上（情况 1(a), 1(b)）并且 $t_p < 0$，那么光线与球面不相交；
3. 计算球心到光线所在直线的距离 d， $d^2 = \vec{l}^2 - t_p^2$

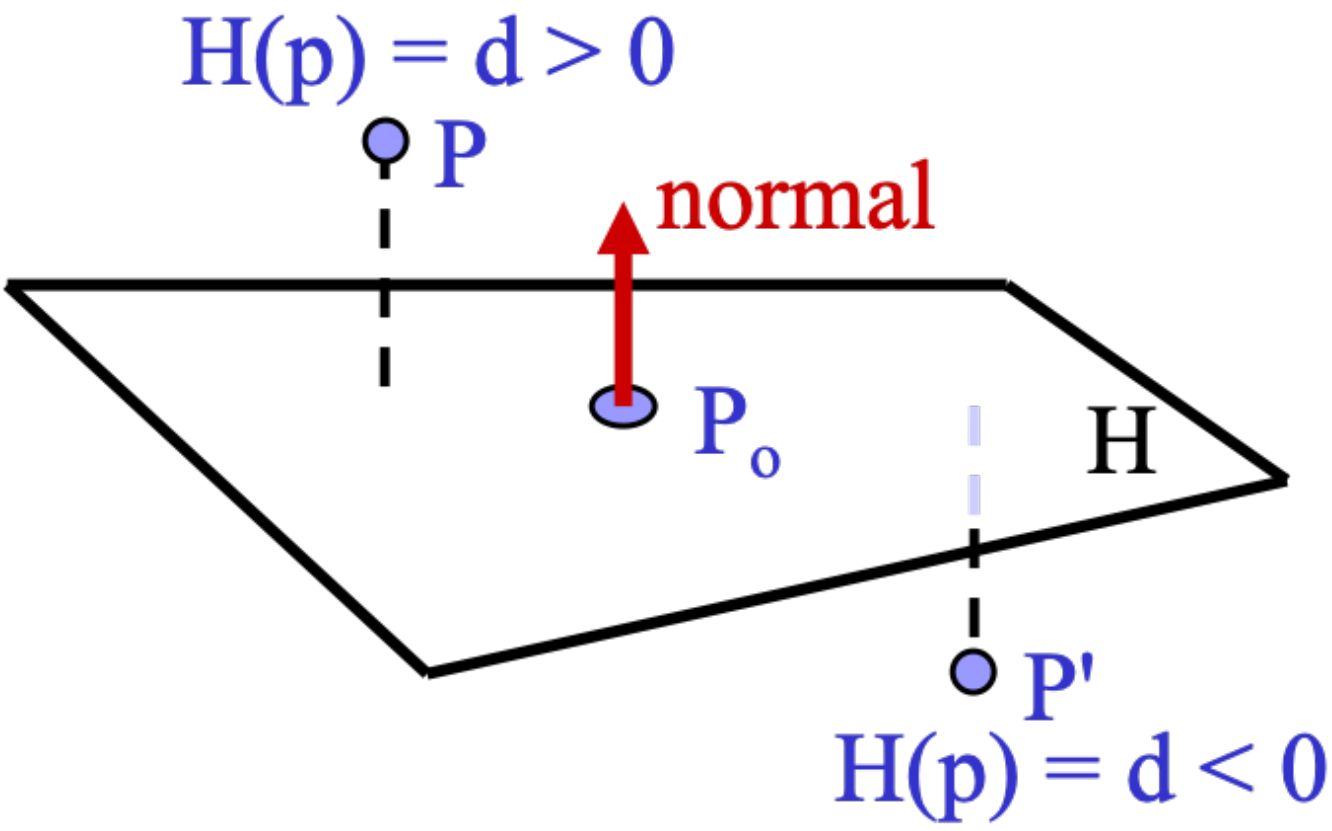
若 $d > r$ （也可以直接判断 $d^2 > r^2$），那么光线与球面不相交；

4. 计算投影点到光线与球面的交点的距离 t'，$t'^2 = r^2 - d^2$
   (a) 若光源在球体外部：$t = t_p - t'$
   (b) 若光源在球体内部：$t = t_p + t'$
5. 判断 t 是否为当前最靠近相机的点，若不是则捨弃此交点；
6. 计算垂足处的法向量：$\vec{n} = R_d * t - l$ 。

1.2 平面求交

三维空间内的平面可以隐式表示为：$H(P) = Ax + By + Cz + D = 0$，记向量 $\vec{n} = (A, B, C)$ ，那么有 $H(P) = n \cdot P + D = 0$ ，根据此定义可以得到点到平面的距离就是 $H(P)$ 。

现给定一条光线 $P(t) = R_O + tR_d$，要计算此线与平面的交点可以联立平面的方程作计算，最终得到：$t = -(D + n \cdot R_O)/(n \cdot R_d)$ ，若 $t > 0$ 且 t 为当前最靠近相机的点，那么此点为一个可视的交点，最后通过判断 $n \cdot R_d$ 的正负号可以判断出 t 点处法向的方向，算法结束。



1.3 三角形面片求交

三角形 ABC 内的一点 P 可以表示成一组向量之和：$P = \alpha A + \beta B + \gamma C$ ，其中 (α, β, γ) 是重心坐标，满足 $0 \le \alpha, \beta, \gamma \le 1, \alpha + \beta + \gamma = 1$ 。

将此线与光线方程联立，得到 $R_O + tR_d = (1 - \beta - \gamma)A + \beta B + \gamma C$ ，利用 Cramer 法则可以得到方程的解为：

$$\begin{pmatrix} t \\ \beta \\ \gamma \end{pmatrix} = \frac{1}{det(E_d, E_1, E_2)} \begin{pmatrix} det(S, E_1, E_2) \\ det(R_d, S, E_2) \\ det(R_d, E_1, S) \end{pmatrix}$$

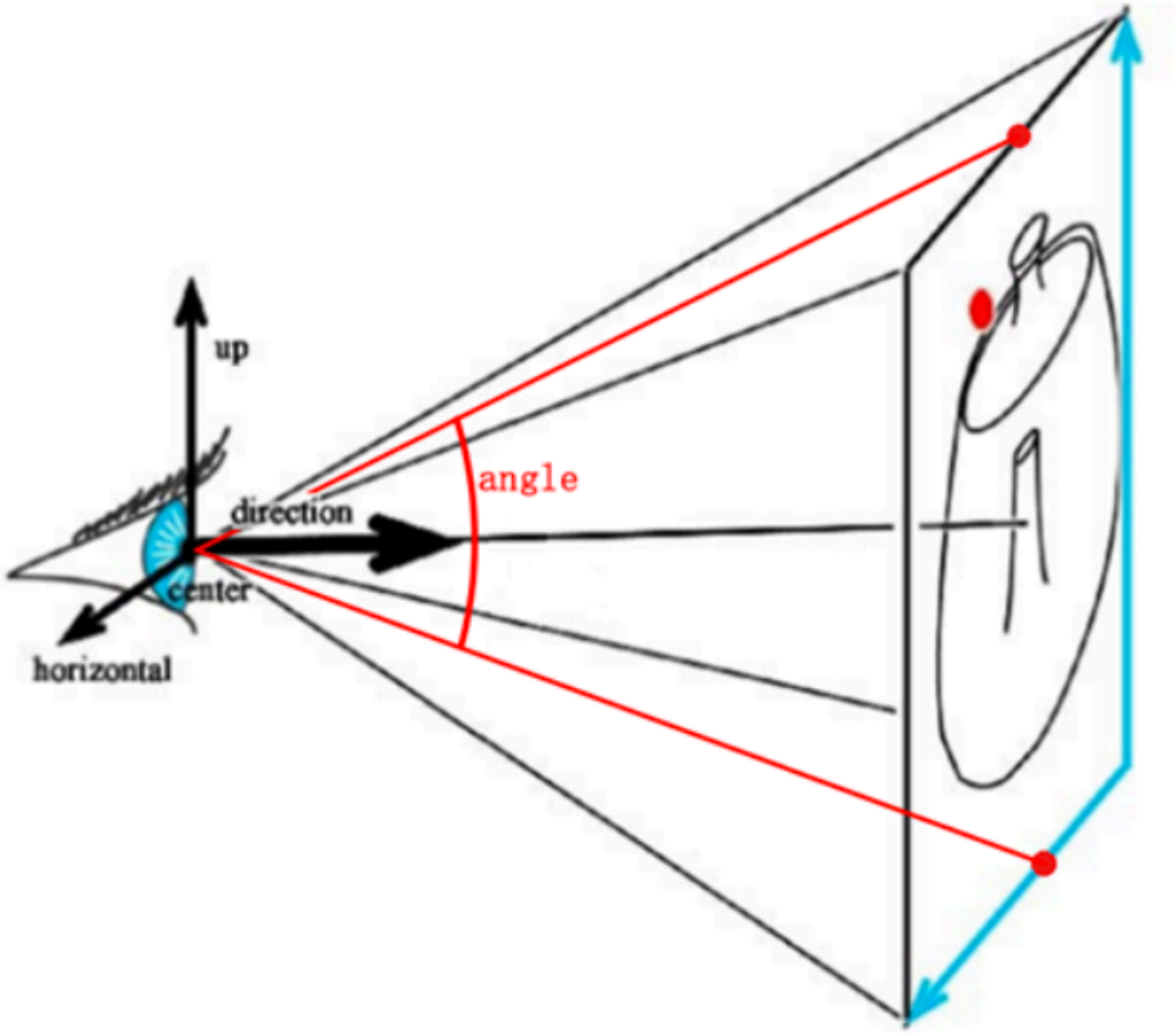
其中，$E_1 = A - B, E_2 = A - C, S = A - R_O$ 。

得到结果后需要通过 $t > 0$ 和 $0 \leq \beta, \gamma \leq 1, \beta + \gamma \leq 1$ 来检查交点是否在三角形内部。

2 透视相机构造

2.1 透视相机参数

常见的透视相机参数包括 $(w, h, f_x, f_y, c_x, c_y)$ ，其中 (w, h) 是所拍照片的宽度和高度，(c_x, c_y) 是光心位置，这里取图像中心点 $(w/2, h/2)$ 为光心位置，(f_x, f_y) 是图像空间到真实世界空间的尺度参数，即图像中一个像素的距离在真实场景中对应的长度。



在本次实验中，我们已经知道的参数有：$w, h, \overrightarrow{up}, \overrightarrow{direction}, angle, center$ 。

利用向量叉积可以算出另一坐标轴向量：$\overrightarrow{horizontal} = normalized(\overrightarrow{direction} \times \overrightarrow{up})$ ，通过视场角 $angle$ 可以得到 $f_x = \frac{w}{2tan(angle/2)}, f_y = \frac{h}{2tan(angle/2)}$ 。

2.2 光线生成

光线投射算法需要从 2.1 节提到的 center 处向屏幕射出光线，如果想要计算像素 (u, v) 处的射线，可以先计算相机空间下（即以相机为原点的坐标系）的射线 R_c：

$$O_{R_c} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \overrightarrow{d_{R_c}} = normalized \begin{pmatrix} \frac{u-c_x}{f_x} \\ \frac{c_y-v}{f_y} \\ 1 \end{pmatrix}$$

再通过坐标转换，将相机空间下的射线 R_c 转换到世界空间下的射线 R_w：

$$O_{R_w} = t, \overrightarrow{d_{R_w}} = [\overrightarrow{horizontal}, -\overrightarrow{up}, \overrightarrow{direction}] \cdot \overrightarrow{d_{R_c}}$$

3 模型着色

直接使用 Phong 模型计算局部光强，在不考虑环境光的情况下，每个像素最终的着色公式为：

$$I = \sum_i c_i \left(k_d \cdot \text{clamp}(\vec{L_i} \cdot \vec{N}) + k_s \cdot \text{clamp}(\vec{V} \cdot \vec{R_i})^s) \right)$$

其中第 i 条光线的反射光 $\vec{R_i} = 2(\vec{N} \cdot \vec{L_i})\vec{N} - \vec{L_i}$，将所有数值代入上式即可得到答案。

4 其他问题

1. **你在实现中遇到了哪些问题?**

   在实践的过程中，我曾经出现过几何体的大小与正确的图像不一，图像颜色错误，反射光线偏差等问题。经排查这些问题和向量正交化、公式输入错误等问题相关，修改后即可解决问题。

2. **你在完成作业的时候和哪些同学进行了怎样的讨论?是否借鉴了网上/别的同学的代码?**

   完成此次作业并没有和任何同学进行讨论，代码主要是参考了课件上的思路。

3. **如何编译你的代码并输出上述 7 个测试用例的渲染结果?如果你没有使用框架代码，请放上你渲染的图片。**

   直接运行代码中的脚本即可。

4. **你的代码有哪些未解决的 bug？如果给你更多时间来完成作业，你将会怎样进行调试?**

   代码完好，理论上不存在 bug。若有更多时间完成作业，我会利用更大的画布来测试所用算法的效能。

5. **你对本次作业有什么建议? 文档或代码中有哪些需要我们改进的地方?**

   我认为本次作业的框架和指引都十分清晰，而且任务量恰到好处，暂时没有需要改进的地方。