

Adding a Sphere

Xiaomx32

2022 年 6 月 21 日

总览

为光线追踪器加入一个物体。

加入球体是因为计算光线与之相交十分方便，在初期，为避免陷入细节，先用简单的例子来体会光线追踪的原理是十分有好处的。

直线与球体相交

球体方程：

$$(x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 = R^2,$$

其中 (c_x, c_y, c_z) 为球心。

直线方程：

$$P(t) = \mathbf{a} + t \cdot \mathbf{b},$$

联立求解：

$$(P(t) - \mathbf{c}) \cdot (P(t) - \mathbf{c}) = R^2$$

展开：

$$t^2 \mathbf{b} \cdot \mathbf{b} + 2t \mathbf{b} \cdot (\mathbf{a} - \mathbf{c}) + (\mathbf{a} - \mathbf{c}) \cdot (\mathbf{a} - \mathbf{c}) - R^2 = 0$$

这是一个关于 t 的一元二次方程，

1. $\Delta > 0$, 有两个不相等的实根，直线与球体有两个交点；
2. $\Delta = 0$, 有两个相等实根，直线与球体有一个交点；
3. $\Delta < 0$, 没有实根，直线与球体不相交。

代码解读

main.cpp

`hit_sphere` 函数：用于计算直线与球体是否相交，其中，球心默认位置为 $(0, 0, -1)$ 。

`ray_color` 函数：当直线击中球体时绘制颜色。

此部分最后的代码，是将球面法向量的值最为该点的颜色值输出。直线与球体相交，取最近的交点。

hitable.h

如果场景中不止一个球或其它物体，我们就需要对它们进行管理。这里使用一个 `sphere` 数组管理场景中的球体。用一个抽象类，任何可能与光线求交的物体实现时都需要继承这个类。

hittable_list.h

存放物体的列表。

参考文献

- [1] <https://zhuanlan.zhihu.com/p/128582904>
- [2] <https://www.cnblogs.com/lv-anchoret/p/10163205.html>
- [3] <https://blog.csdn.net/hackbuteer1/article/details/7558868>