

光线追踪算法理论

段元兴

2019 年 4 月 20 日

Content

1	摘要	3
2	基本原理	3
3	理论推导	3
3.1	基本模型的数学描述	3
3.1.1	颜色模型	3
3.1.2	光线的基本模型	3
3.1.3	平面模型	4
3.1.4	三角形模型	4
3.1.5	球形模型	4
3.1.6	圆盘模型	4
3.1.7	圆柱模型	4
3.1.8	圆锥模型	5
3.2	相交测试的推导	5
3.2.1	平面	5
3.2.2	三角形	5
3.2.3	球形	6
3.2.4	圆盘	6
3.3	光线传播的模拟	6
3.3.1	镜面反射	6

1 摘要

作为Computer Graphics的圣杯——光线追踪,是一个十分古老但是热门的算法. 接下来将简要的介绍光线追踪算法的理论推导及基本实现.

2 基本原理

真实世界的几何光学模型是光线从光源发出,经过折射,镜面反射,漫反射等方式进入人眼.而现有计算机图形学的基本渲染方法却是光栅化,即利用空间投影变换将各种基本图形元素(如三角形)投影到屏幕坐标然后上色等等.然而这种做法有先天性的巨大缺陷:在实现折射反射及阴影等效果的时候需要利用各种所谓的技巧才能不太精确的模拟真实情况.

而光线追踪算法突破了这些传统图形学的局限性.一般来说光线追踪算法分两种:

1.正向光线追踪:完全按照真实世界的几何光学模型,模拟的光线从光源发出,经过折射反射等作用进入人眼从而被看到.而由于一般只有少量从光源发出的光线被人眼看到,所以这种方法浪费了巨大的计算能力.

2.反向光线追踪:利用几何光线的光路可逆性,将原来的光线的方向改变从人眼出发,极大的降低了所需要的光线数量.但是在路径追踪里,通常也需要与正向追踪结合起来加快寻找光源的速度.

对于反向光线追踪我们获取了光线命中的图元的自发光颜色(如果是光源)并将其与现有光线颜色按照当前光线所占比例混合,直到最终没有与任何图元相交(取天空盒子的背景颜色).而对于反射系数不为0的界面,计算出新的法向代替原法向然后重复这一过程.而对于折射光线,则将分出一束折射光线然后重复.由于glsl语言不支持递归,所以这里需要对栈进行模拟.但是由渲染方程可知物体之间是存在漫反射和焦散效果的,但是由于目前尚未采用蒙特卡洛随机采样来进行路径追踪(性能是原因之一),故在此忽略...,但是对于光源的漫反射我们仍然会计算.

3 理论推导

3.1 基本模型的数学描述

由于计算机只认识数,所以我们要将当前的各种物理模型转换成数学语言.这里涉及到基本的矢量计算.

3.1.1 颜色模型

虽然自然生活中光的波长是连续分布的,但是在这里采用显示器的描述方式,即用 (R, G, B) 的混合来描述千千万万种个颜色. 而一个物体的光学属性则十分复杂,这里仅仅考虑其表面的自发光颜色 $g(\text{glow})$,表面的反射颜色 $r(\text{reflect})$,透射颜色 $t(\text{transmission})$,对光源的漫反射系数 $d(\text{diffuse})$ 和折射率 n .所以用类 $Color : \{r, t, g, d, n\}$ 来储存颜色信息. 而这些信息有可能随着在几何图元上的位置的不同而变化,因此有必要建立一个纹理坐标,使OpenGL能够从纹理数组读取这些信息.

3.1.2 光线的基本模型

一根光线由出发点 p_0 ,方向 \vec{n} (已经归一化),颜色构成.为了方便GPU优化计算,这里的 p_0 加上一个 $w = 1$ 分量构成 $p^\mu = (p_0, 1)$.

而颜色一般由 (R, G, B) 三个分量来表示.由于我们用的是反向光线追踪,所以还需要一个强度因子 (k_R, k_G, k_B) 来表示当前光线占总光线强度的比例.

光线经过的所有点可以由:

$$r^\mu = p_0^\mu + t\vec{n}, t \geq 0 \quad (1)$$

来表述.

3.1.3 平面模型

这里平面的定义为一个无穷大的三维中的二维平面,由三维笛卡尔坐标方程来描述:

$$Ax + By + Cz + W = 0. \quad (2)$$

为简化描述,我们用

$$n_p^\mu = (\vec{n}_p, W) = (A, B, C, W) \quad (3)$$

来描述这个平面,其中 \vec{n}_p 也是该平面的法向(保证已经归一化).

3.1.4 三角形模型

这里三角形用空间中三个点 p_0, p_1, p_2 来定义,并且法向 \vec{n}_p 由顺序 p_0, p_1, p_2 按照右手螺旋法则定义为正方向.由这些顶点可以计算出 \vec{n}_p (已归一化)和该三角形所在的平面 n_p^μ .而在三角形平面内的点可以用仿射坐标 $\vec{t} = (u', v')$ 来表示,并且该坐标的单位向量为 $\vec{e}_0 = p_0 p_1, \vec{e}_1 = p_0 p_2$.三角形的三个顶点又具有其在纹理图片上的纹理坐标

$$\begin{cases} \vec{t}_0 = (u_0, v_0) \\ \vec{t}_1 = (u_1, v_1) \\ \vec{t}_2 = (u_2, v_2) \end{cases} \quad (4)$$

从 (u', v') 计算实际纹理坐标 \vec{t} 可由以下公式得出:

$$\vec{t} = u' \vec{t}_1 + v' \vec{t}_2 + (1 - u' - v') \vec{t}_0. \quad (5)$$

3.1.5 球形模型

设球心的位矢为 p_1 ,半径为 R .为方便GPU储存与计算,用 $p_1^\mu = (\vec{p}_1, R^2)$ 来储该球的几何特征.纹理坐标这么表示:定义一个球的z方向(类似于地球的极轴,从地心指向北极点为正方向),定义一个x方向,根据右手系建立球坐标系 (r, θ, ϕ) .而纹理坐标 (u, v) 由 θ 和 ϕ 经过归一化得到:

$$\begin{cases} u = \frac{\phi}{2\pi} \\ v = 1 - \frac{\theta}{\pi} \end{cases}. \quad (6)$$

3.1.6 圆盘模型

类似于所有平面图形,圆盘由其所在平面 $p^\mu = (\vec{n}, W)$,圆心坐标和圆半径 $p_1^\mu = (\vec{p}_1, R^2)$ 组成.而为了计算纹理坐标,我们还要破坏其轴对称性,即在圆盘平面内架一个 uv 坐标系.因此首先在平面内指定一个方向为 u 坐标正方向 \vec{e}_0 ,然后由法向 \vec{n} 计算出另一个坐标的单位向量 \vec{e}_1 .设焦点相对圆心的位置为 \vec{d} ,则该点的纹理坐标为:

$$\begin{cases} u = \frac{\vec{d} \cdot \vec{e}_0}{2R} + \frac{1}{2} \\ v = \frac{\vec{d} \cdot \vec{e}_1}{2R} + \frac{1}{2} \end{cases}. \quad (7)$$

3.1.7 圆柱模型

这里的圆柱只描述其侧面(底面和顶面都是由圆来描述).圆柱可以由一个圆柱底面圆心坐标 \vec{c} ,圆柱轴线方向 \vec{n} (从底面圆心到顶面圆心为正方向),圆柱半径 R 和原址高度 l 这些参数表述.而侧面方程为:

$$|(\vec{r} - \vec{c}) \times \vec{n}| = R, \quad (8)$$

且满足:

$$0 \leq (\vec{r} - \vec{c}) \cdot \vec{n} \leq l. \quad (9)$$

至于纹理坐标,可以建立以下纹理坐标:以 \vec{c} 为坐标原点,以 \vec{n} 为z轴,再指定一个方向 \vec{e}_0 为 x轴正方向,建立右手柱坐标系 (r, θ, z) .然后纹理坐标为:

$$\begin{cases} u = \frac{\theta}{2\pi} \\ v = \frac{z}{l} \end{cases} . \quad (10)$$

3.1.8 圆锥模型

啊哈哈

3.2 相交测试的推导

如果我们想要求光线与几何图元的相交点,则必须求出方程1中的 t .

3.2.1 平面

方程2即可以写成:

$$r^\mu n^\mu = 0, \quad (11)$$

又因方程1,故

$$(p_0^\mu + t\vec{n})n^\mu = 0. \quad (12)$$

由此可以解出 t :

$$t = -\frac{p_0^\mu n_\mu}{\vec{n} \cdot \vec{n}_p}. \quad (13)$$

3.2.2 三角形

光线若与一个三角形相交,则必先与该三角形所在的平面相交.由方程13可求得 t ,由方程1可知位置矢量 r^μ .而为判断该交点是否在三角形内部,我们首先要计算该焦点在平面内的仿射坐标 (u, v) .设交点 \vec{r} 到三角形顶点 p_1 的位矢为:

$$\vec{d} = \vec{r} - \vec{p}_1 = u\vec{e}_1 + v\vec{e}_2. \quad (14)$$

则:

$$\begin{cases} \vec{d} \cdot \vec{e}_1 = |\vec{e}_1|^2 u + (\vec{e}_1 \cdot \vec{e}_2) v \\ \vec{d} \cdot \vec{e}_2 = (\vec{e}_1 \cdot \vec{e}_2) u + |\vec{e}_2|^2 v \end{cases} , \quad (15)$$

由此可以解出:

$$\begin{cases} u = \vec{k}_1 \cdot \vec{d} \\ v = \vec{k}_2 \cdot \vec{d} \end{cases} , \quad (16)$$

其中:

$$\begin{cases} \vec{k}_1 = \frac{|\vec{e}_2|^2 \vec{e}_1 - (\vec{e}_1 \cdot \vec{e}_2) \vec{e}_2}{s} \\ \vec{k}_2 = \frac{|\vec{e}_1|^2 \vec{e}_2 - (\vec{e}_1 \cdot \vec{e}_2) \vec{e}_1}{s} \end{cases} , \quad (17)$$

$$s = |\vec{e}_1 \times \vec{e}_2|^2. \quad (18)$$

而判断交点是否在三角形内则可由以下条件判断:

$$\begin{cases} u \geq 0 \\ v \geq 0 \\ u + v \leq 1 \end{cases} . \quad (19)$$

3.2.3 球形

光线若与一个球相交,则该直线到球心的距离小于 R .而这个距离平方可以表示为 $|(\vec{p}_1 - \vec{p}_0) \times \vec{n}|^2$.令 $\vec{d} = \vec{p}_1 - \vec{p}_0$, 则有:

$$s^2 = R^2 - |\vec{d} \times \vec{n}|^2 \geq 0. \quad (20)$$

球的方程为:

$$|\vec{r} - \vec{p}_1|^2 = R^2, \quad (21)$$

代入方程1可得:

$$|\vec{n}t - \vec{d}|^2 = R^2, \quad (22)$$

由此可以解出:

$$\begin{cases} t_1 = \vec{n} \cdot \vec{d} + \sqrt{R^2 - \left(|\vec{d}|^2 - (\vec{n} \cdot \vec{d})^2\right)} \\ t_2 = \vec{n} \cdot \vec{d} - \sqrt{R^2 - \left(|\vec{d}|^2 - (\vec{n} \cdot \vec{d})^2\right)} \end{cases}. \quad (23)$$

由于 $|\vec{d} \times \vec{n}|^2 = |\vec{d}|^2 - (\vec{n} \cdot \vec{d})^2$, 故:

$$\begin{cases} t_1 = \vec{n} \cdot \vec{d} + s \\ t_2 = \vec{n} \cdot \vec{d} - s \end{cases}. \quad (24)$$

当然只有其中一个解是最近相交点于是最小的正解即为我们所求的 t .

3.2.4 圆盘

同三角形,首先计算出相交位置 r^μ .然后判断是否在圆盘内:

$$|\vec{d} = \vec{r} - \vec{p}_1, |\vec{d}|^2 < R^2. \quad (25)$$

纹理坐标则是:

$$\begin{cases} u = \vec{d} \cdot \vec{e}_1 \\ v = \vec{d} \cdot \vec{e}_2 \end{cases}. \quad (26)$$

3.3 光线传播的模拟

为了达到真实的光学效果,需要计算光线传播过程中的镜面反射,与光源的漫反射和折射组成. 而且对于玻璃材质,需要考虑光学的菲涅尔公式才能真实模拟光线的透射率和反射率.设光线与某个几何图元相交,得到以下参数:

- 1.几何图元的法向 \vec{n}
- 2.光线入射方向向 \vec{n}_0
- 3.光线传播的距离 t
- 4.几何图元的折射率 n
- 5.几何图元的颜色信息 c

3.3.1 镜面反射

考虑法向 \vec{n} 的变化:

$$\vec{n}' = \quad (27)$$