佛山大学学报　第 14 卷第 6 期　1996 年 12 月
Journal of Foshan University　Vol.14 No.6　Dec.1996

光线跟踪的直线与球体求交的快速算法†

范彦斌*　沈自林*　王宇华*　冯心海*

摘　要　光线跟踪技术的实用性取决于光线与场景物体求交算法的速度. 本文就光线与球体的映射问题, 提出了一种光线与球体求交的非解析法, 提高了光线跟踪求交的速度.

关键词　计算机图形学　光线跟踪　计算方法

光线跟踪是一种高度真实感图形显示和绘制技术. 由于它能生成传统图形学算法难以模拟的整体光照效果, 且算法简单, 实现方便, 近年来越来越受到人们的重视. 然而, 在一个具有 $M\times N$ 的显示分辨率的屏幕上生成一幅图形, 必须从视点出发通过屏幕向景物发射 $M\times N$ 条光线. 假设每条光线在景物中经过反射和折射的平均派生的光线条数为 d, 并设每一景物的交点向光源发射 m 条阴影光线, 则对于一个 $M\times N$ 的显示分辨率的屏幕总的光线数 $(m+1)dMN$.

当 $m=2$, $d=5$, $M=N=512$ 时, 其光线总数接近 400 万条[1]. 这就意味着需要进行近 400 万次的直线与景物的求交计算才能完成图形的显示绘制. 庞大的求交计算量使图形显示的耗费大大增加, 生成一幅中等复杂程度的真实感图形需要几个甚至几十个小时. 因此, 寻求一种快速的直线与景物求交的计算方法, 对提高光线跟踪技术的应用价值具有重大意义.

本文在已有直线与球体求交的解析解基础上, 在用光线跟踪方法显示多球体场景的实践中, 提出了光线与球体求交的非解析计算方法.

1　光线与球体相交的代数解

定义一条光线为

光源　　$\vec{R}_0=[X_0, Y_0, Z_0]$

方向　　$\vec{R}_d=[X_d, Y_d, Z_d]$

其中　　$X_d^2+Y_d^2+Z_d^2=1$　(矢量归一化)

光线上的点　$\vec{R}(t)=\vec{R}_0+\vec{R}_d t$　$(t>0)$ (1)

方程(1)是一个参数化的显函数, 所有光线上的点都可以通过改变时间参数 t 而得到.

设球的方程为

球心　　$\vec{S}_c=[X_c, Y_c, Z_c]$

球半径　　S_r

球面上的点　　$\vec{S}_p=[X_p, Y_p, Z_p]$

约束方程　$(X-X_p)^2+(Y-Y_p)^2+(Z-Z_p)^2=S_r^2$　(2)

收稿日期: 1996－08－27
† 广东省自然科学基金资助项目
* 佛山大学思源机电一体化研究所, 528000　佛山

为了求出光线与球面的交点,将方程(1)代入方程(2)中得

$$(X_0+X_dt-X_c)^2+(Y_0+Y_dt-Y_c)^2+(Z_0+Z_dt-Z_c)^2=S_r^2,$$

展开形成为

$$At^2+Bt+C=0 \qquad (3)$$

其中

$$A=X_d^2+Y_d^2+Z_d^2=1;$$
$$B=2(X_d(X_0-X_c)+Y_d(Y_0-Y_c)+Z_d(Z_0-Z_c));$$
$$C=(X_0-X_c)^2+(Y_0-Y_c)^2+(Z_0-Z_c)^2-S_r^2.$$

可见方程(3)为一个一元二次方程,其解为

$$t_0=\frac{-B-\sqrt{B^2-4C}}{2},$$
$$t_1=\frac{-B+\sqrt{B^2-4C}}{2}.$$

当判别式 $B^2-4C<0$ 时,表示光线与球无交点.

由于时间 $t>0$,故偏小的正根是靠近光源与球相交的点.当时间 t 求出后,实际的交点为

$$\vec{r}_i=[X_i,Y_i,Z_i]=[X_0+X_dt,Y_0+Y_dt,Z_0+Z_dt],$$

交点的法向量为

$$\vec{r}_n=\left[\frac{X_i-X_c}{S_r},\frac{Y_i-Y_c}{S_r},\frac{Z_i-Z_c}{S_r}\right].$$

2 光线与球体相交的非解析法

从光线与球体相交的代数法中可以看出,使用算法语言实现求交计算的编程,需要对代数解中的开方和除法运算进行多次循环.一般一次开方运算的时间相当于是 15 到 30 倍的乘法计算时间,而除法运算也较乘法运算用的时间多,这对于百万次数量级的光线与球体景物的求交计算来说,其计算速度是相当慢的.

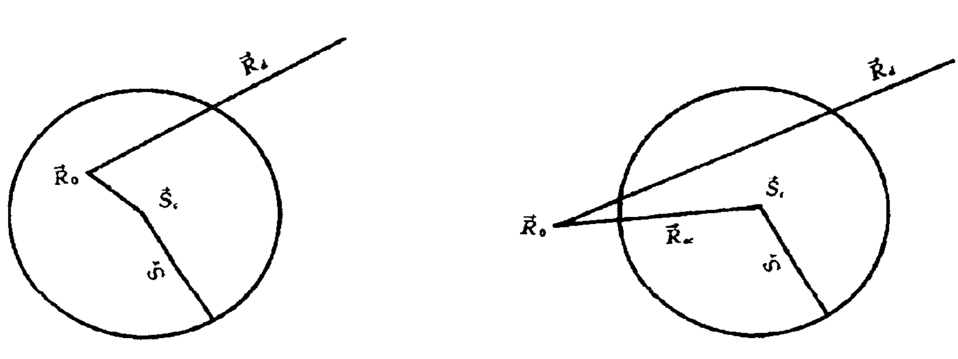
解决这类问题的最有效的方法是减少计算的次数.在实际的光线跟踪图形显示中,在很多情况下,光线与球体是无交的.因此,可以在求交计算之前,如对光线的光源与球体的相对位置、光源与球心的距离以及光线的照射方向等进行预判断,这样就可以做到有的放矢地进行求交计算,从而提高求交的计算速度.以下为光线与球体求交的非解析法在实际应用中的步骤:

(1)判断光源与球的相对位置

设光源到球心的矢量为

$$\vec{R}_{0c}=\vec{S}_c-\vec{R}_0 \qquad L_{0c}=\vec{R}_{0c}\circ\vec{R}_{0c}$$

如果 $L_{0c}<S_r^2$,光源在球体之内;如果 $L_{0c}\geq S_r^2$,光源在球体之上或在球体之外,如图 1 所示.



$(\vec{S}_c-\vec{R}_0)\circ(\vec{S}_c-\vec{R}_0)\geq S_r^2$ $(\vec{S}_c-\vec{R}_0)\circ(\vec{S}_c-\vec{R}_0)<S_r^2$

图 1 光源与球心的相对位置

(2) 光源到球心的最小距离

从光源沿着光线的照射方向到球心的最小距离为

$$d_{\min} = \vec{R}_{0c} \circ \vec{R}_d$$

如图 2 所示, 当 $d_{\min} < 0$ 时, 球心位置在光源位置之前, 光线是远离光源; 当 $d_{\min} > 0$ 时, 球心位置在光源位置之后, 但还不足以判断光线是否与球有交点. 由于当光源位于球体之内时, 光线一定与球有交, 所以, 对于 $d_{\min} < 0$ 时, 只有在光源位置位于球体外时, 才说明光线在照射方向上与球无交点.

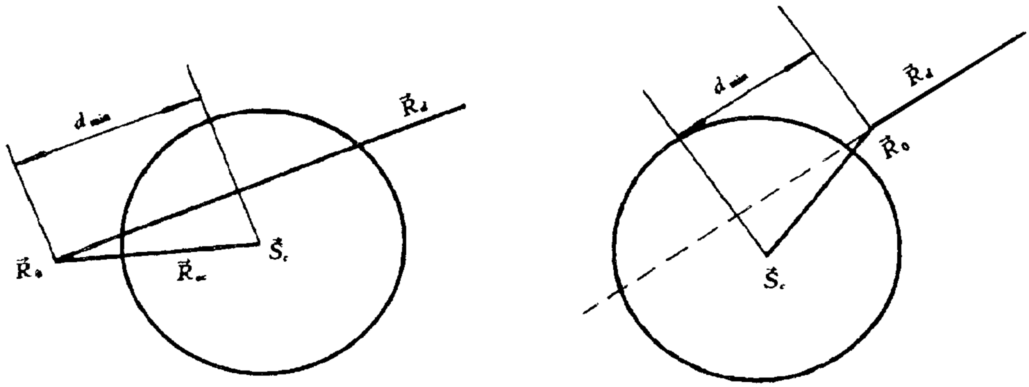


图 2　光线与球的相对方向

(3) 半弦长的平方值 d_{hc}^2

设 $|\vec{D}|$ 为球心到光线方向矢量的距离, 则有(如图 3 所示):

$$d_{hc}{}^2 = S_r^2 - \vec{D}^2$$

而　$$\vec{D}^2 = \vec{R}_{oc}^2 - d_{\min}^2$$

所以　$$d_{hc}^2 = S_r^2 - \vec{R}_{oc}^2 + d_{\min}^2.$$

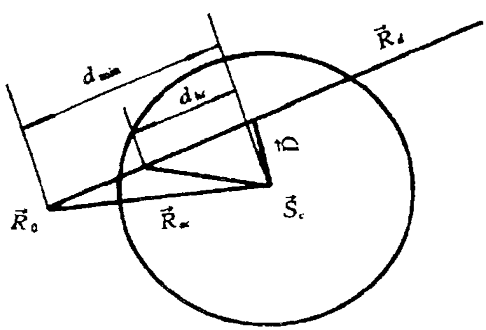


图 3　光线与球相交的几何特性

光源在球体外时, 如果 $d_{hc} < 0$ 时, 光线与球无交点. 因此, 在完成了第(2) 和(3) 步的判断之后, 就可以准确地得出光线与球是否有交.

(4) 计算光线与球体的交点

在判断完光线与球体有交之后, 可按下式直接计算出光线与球体的交点.

因为当光源在球之外时, 距离较小的值为第一交点

$$t = d_{\min} - \sqrt{d_{hc}}.$$

当光源在球之上或在球之内时, 距离较小的值为负值, 故取距离较大的作为交点:

$$t = d_{\min} + \sqrt{d_{hc}}.$$

3　代数解与非解析法的比较

从代数解和非解析法的算法来看, 在一些算式上有相似之处, 但是, 非解析方法的优点是通过第一个判断, 能够首先指出光线是否远离球体而与球体不相交, 从而减少解析法中不必要的开方计算. 这个判断在光线跟踪求交算法中十分重要, 因为在实际的计算中几乎一半以上的光线与球体无交点. 代数解法则要通过计算代数方程的判别式后才能

判断光线是否与球体有交点, 从而花费大量的时间.

作者对均匀分布的 30 个不同半径的球体在分辨率为 1024× 1024 的 SGI 工作站上, 使用光线跟踪的方法进行了真实感显示. 每条光线在景物中经过反射和折射的平均派生的光线条数为 5, 每一景物的交点和光源发射 2 条阴景光线, 则光线总数约为 1500 万条. 使用解析法完成该景物的真实感显示所需的计算时间约为 2 h, 如果使用非解析法则所需的计算时间约为 35 min.

针对特殊的景物特性, 本文所提出的非解析法在于加快光线跟踪的求交速度, 而代数解在计算一般的直线与球体求交问题中仍然是一种有效的方法.

参 考 文 献

1 唐荣锡, 汪嘉业, 彭群生. 计算机图形学教程. 北京: 科学出版社, 1990. 367~ 371

A Faster Algorithm for Finding the Intersection Points of a Line and Sphere in Ray Tracing

Fan Yanbin * Shen Zilin * Wang Yuhua * Feng Xinhai *

Abstract The speed of finding the intersection points of lines and objects is the key to the applications of ray tracing in computer graphics. A speed-up non-algebraic solution for finding the intersection points of a line and a sphere on the special characteristics of the ray is presented.

Keywords computer graphics, ray tracing, computing algorithm

*Siyuan Mechantronics Institute, Foshan University, 528000 Foshan