一个计算凸多面体间碰撞点的快速算法*

王兆其[1] 赵沁平[2] 汪成为[3]

[1](中国科学院计算技术研究所 北京 100080)

[2](北京航空航天大学计算机科学与工程系 北京 100083)

[3](中国人民解放军总装备部科学技术委员会 北京 100034)

E-mail: zqwang@ict.ac.cn

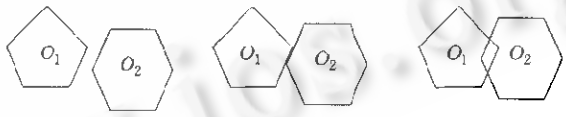
摘要 计算两个物体之间的碰撞点是碰撞响应的基础,也是一项系统开销很大的任务.因此,研究碰撞点快速求解算法对碰撞响应的实时性具有重要意义.该文提出了一个算法,当在虚拟环境中检测到碰撞时,应用此算法可以在碰撞响应之前快速计算出两个物体之间的准确碰撞时间,并能计算出此时两个物体之间的碰撞点.

关键词 碰撞点,碰撞响应,碰撞检测,虚拟现实.

中图法分类号 TP391

当在虚拟环境中检测到两个物体碰撞后,进行碰撞响应之前,需要获知以下信息[1]:① 两个物体之间的准确碰撞时间;② 碰撞时两个物体之间的所有碰撞点.

在已有的一些方法[2,3]中,常用检测到的碰撞时间作为发生碰撞的时间.实际上,这两者之间存在着差别.如图1所示,在t_i时刻,如图1(a)所示,O_1与O_2分离;在t'时刻,如图1(b)所示,O_1与O_2开始接触,其中$t_i<t'<t_{i+1}$;在t_{i+1}时刻,如图1(c)所示,O_1与O_2之间发生了穿透.即当在t_{i+1}时刻检测到碰撞时,早在t'时刻两个物体之间便发生了碰撞.当碰撞检测的时间步长$dt=(t_{i+1}-t_i)$较大时,得到的碰撞时间误差也会较大,从而引起碰撞点计算结果很不准确.为了提高计算精度,常需要增加碰撞检测次数,这不利于实时碰撞响应的实现.另外,两个物体之间的碰撞点仅在两个物体开始接触时有意义,而当两个物体已经发生穿透时,求解两个物体之间的碰撞点是一个不确定性问题.因此,快速而准确地计算出物体之间的碰撞时间t',并求解此时物体之间的碰撞点是进行实时碰撞响应的重要内容.



(a) t_i 时刻 (b) t' 时刻 (c) t'_{i+1} 时刻

图1

James K. Hahn的方法[2]是将时间步长$dt=(t_{i+1}-t_i)$取值非常小,并假设dt时间区域内物体的角速度为0.取dt太小则难以满足虚拟环境的实时交互要求.取dt较大时,则不能假设物体的角速度为0.Madhav K. Ponamgi等人通过计算两个物体的最短距离来确定两个物体是否发生了碰撞[3].当两个多面体的距离为0时,两个物体最早相交.为了保证每次检测到的碰撞物体之间的距离近似为0,我们仍然需要减小时间步长dt,因此会影响碰撞响应的实时性.为了找到准确的碰撞时间与碰撞点,Harmut Keller等人将环境中的物体限制为规则

---

* 本文研究得到国家863高科技项目基金和国家科技部"九五"攻关项目基金资助.作者王兆其,1966年生,博士,主要研究领域为虚拟现实.赵沁平,1948年生,博士,教授,博士生导师,主要研究领域为计算机软件,人工智能,虚拟现实.汪成为,1933年生,教授,博士生导师,中国工程院院士,主要研究领域为模拟计算机,数字计算机,系统仿真,人工智能,虚拟现实.

本文通讯联系人:王兆其,北京100080,北京2704信箱

本文1998-10-14收到原稿,1999-01-11收到修改稿

几何体，如立方体、球体等，但这种方法的通用性受到了限制.

本文提出了一种计算凸多面体间碰撞时间与碰撞点的快速算法 FCPF(fast collision points finding). 当检测到虚拟环境中某两个物体间在 t_{i+1} 时刻发生了碰撞时，FCPF 算法可以根据两个物体的速度 v_1、v_2 和角速度 ω_1、ω_2 快速计算出两物体间最早发生碰撞的时间 t'，并找出此时两个物体之间的所有碰撞点.

1　计算凸多面体间碰撞点的 FCPF 算法

1.1　基本假设

假设 O_1 与 O_2 是虚拟环境中的两个凸多面体，O_1 与 O_2 在 t_0 时刻不碰撞，而在 t_1 时刻碰撞，且在 $(t_0,t_1]$ 内两个物体只发生一次碰撞. 设 O_1 与 O_2 中心的线速度分别为 v_1 和 v_2，角速度分别为 ω_1 和 ω_2. 假设 v_1、v_2、ω_1 和 ω_2 在 $(t_0,t_1]$ 时间内均为常向量.

1.2　理论基础

定义 1. 当给一个平面 F 指定一个法向量 n 后，这个平面称为有向平面. 其法向量所指的一面称为该平面的正面，另一面称为负面. 对于多面体的面，我们作如下约定，其法向量均指向多面体的外部.

定理 1. O_1 与 O_2 是符合第 1.1 节中基本假设的两个多面体. 设 $l=P_1P_2$ 是 O_1 的一条边，F 是 O_2 的一个面，l 与 F 在 $(t_0,t_1]$ 内相交，则 l 与 F 最早相交时刻 $t_c\in(t_0,t_1]$，且最早相交必是下列情形之一，如图 2 所示.

(1) l 的一个端点 P_i 与 F 的内部相交(l 不与 F 的边相交)；

(2) l 与 F 的一条边相交于一点 P；

(3) l 与 F 相交于一线段 $P_1'P_2'$.

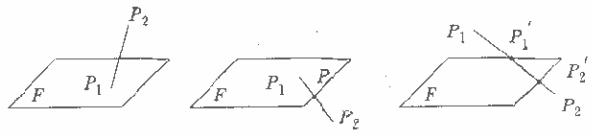


图2　O_1 的边与 O_2 的面相交的3种情形

定理 2. O_1 与 O_2 是符合第 1.1 节中基本假设的两个多面体，最早相交时刻为 $T_c\in(t_0,t_1]$. 设 $l=P_1P_2$ 是 O_1 的一条边，F 是 O_2 的一个面. 如果 l 与 F 在 $(t_0,t_1]$ 为的最早相交时刻为 t_c，则必有 $t_c\geqslant T_c$. 若 $t_c>T_c$，则：

(1) 如果 $l=P_1P_2$ 的一个端点 $P_i(i=1,2)$ 最早在 t_c 时刻与面 F 相交于 F 内部，则，

(a) 当 l 的另一顶点 P_j 位于 F 正面时，则 O_1 在顶点 P_i 处的其他邻边中必有一条边 l'，与 F 的最早相交时刻 $t_c'<t_c$；

(b) 当 l 的另一顶点 P_j 位于 F 负面时，则 O_2 中存在另一个面 $F'\neq F$，P_i 与 F' 的最早相交时刻 $t_c'<t_c$.

(2) 如果 l 最早在 t_c 时刻与 F 的一条边 l' 相交于点 P，则下列 3 种情形必有一种成立：

(a) O_2 在 P 处存在一相邻的面 $F'\neq F$，l 与 F' 的最早相交时刻 $t_c'<t_c$；

(b) O_1 在 P 处存在一相邻的面 F'，l' 与 F' 的最早相交时刻 $t_c'<t_c$；

(c) 如果(a)与(b)都不成立，则 l 与 l' 相交于 l 的一个端点 P_i，且 l 的另一端点 P_j 在 O_2 于 P 处所有邻面的负面，此时 O_2 存在一个面 $F'\neq F$，P_i 与 F' 的最早相交时刻 $t_c'<t_c$.

(3) 如果 l 最早于 t_c 时刻与 F 相交于一条线段 $P_1'P_2'$，则下列 3 种情形必有一成立：

(a) O_2 在 P_1' 或 P_i' 处存在一个相邻的面 F'，l 与 F' 的最早相交时刻 $t_c'<t_c$；

(b) O_2 在 P_1' 或 P_2' 存在 F 的一条边 l'，O_1 在 l 处存在一个相邻的面 F'，l' 与 F' 的最早相交时刻 $t_c'<t_c$；

(c) 如果(a)与(b)都不成立，则必有一交点 P_i' 是 l 的端点，如 $P_i'=P_1$. 存在 O_2 的一个面 $F'\neq F$，P_1 与 F' 的最早相交时刻 $t_c'<t_c$.

1.3　FCPF 算法

当检测到虚拟环境中有两个物体之间发生了碰撞时，如果这两个物体分别由凸多面体 O_1 与 O_2 表示，则可以取到一个物体的一条边 l 与另一个物体的一个面 F，设 $l\in O_1$，$F\in O_2$，l 与 F 相交. 求 l 与 F 的最早相交时刻 t_c

及 l 与 F 相交的情形.由定理 1 可知,l 与 F 相交只可能有 3 种情形.根据每种情形,检测是否有使 $t_c>T_c$ 成立的情形,即是否有 $t_c'<t_c$,O_1 与 O_2 在 t_c' 时刻相交.如果没有,则 t_c 即是 O_1 与 O_2 的最早相交时刻 T_c.否则,以 t_c' 为新的 t_c,并寻找更小的 t_c'.求出 T_c 后,便可以求出两个多面体间所有相交的点,这些点即为两个多面体间的碰撞点.算法 1 给出了 FCPF 算法.

算法 1. FCPF 算法.

(1) 初始化:

$t_c=t_1+1$;　　//取一个比 t_1 大的值

$R=\phi$;　　//R 表示碰撞结果.含碰撞类型、碰撞点集等

$S=\{\langle l,F,O_1\rangle\}$;//$S$ 中存放了尚需检测的情形,初始时为一个已知的边面相交实例

(2) 当 S 为空时,结束;

(3) 取出 S 的第 1 个元素 e;

(4) 如果$e=\langle l,F,O_1\rangle$ //需求边与面的最早相交时刻

求 O_1 中的 l 与 O_2 中的 F 相交的最早时刻 t_c';

如果 $t_c'>t_c$,转(2);

如果 $t_c'=t_c$,更新 R(加入新的碰撞点),转(2);//此时无需修改 S!

//　$t_c'<t_c$,

$t_c=t_c'$;更新 R;//先清空 R,再加入新的碰撞点

Case 1://l 的端点 P_1 与 F 相交于 F 中央(不与 F 的边相交)

如果 l 的另一端点 P_2 位于 F 正面

$L=\{O_1$ 在 P_1 处的边集$\}$;

$S=\{\langle l,F,O_1\rangle|l\in L\}$;

否则:$S=\{\langle l,P_2,O_2\rangle\}$

Case 2://l 与 F 的一条边 l' 交于点 P

$\Gamma_2=\{O_2$ 在 P 处的邻面$\}$;

$\Gamma_1=\{O_1$ 在 P 处的邻面$\}$;

如果交点 P 是 l 的一个顶点 P_1,且 l 的另一个顶点 P_2 在 P 处 O_2 所有邻面负面 $S'=\{\langle l,P_1,O_2\rangle\}$

否则:$S'-\phi;S=\{\langle l,F,O_1\rangle|F\in\Gamma_2\}\cup\{\langle l',F,O_2\rangle|F\in\Gamma_1\}\cup S'$;

Case 3://l 与 F 相交于一线段 $P_1'P_2'$

设 l 与 F 的边 l_1 相交于点 P_1'(若 P_1' 是 l 与 F 内部的交点,则 $l_1=\phi$.此时,设 $P_1'==P_1$);

l 与 F 的边 l_2 相交于点 P_2'(若 P_2' 是 l 与 F 内部的交点,则 $l_2=\phi$.此时,设 $P_2'=-P_2$);

$S_1''=(l_1==\phi)?\{\langle l,P_1,O_2\rangle\}:\phi$;

$S_2''=(l_2==-\phi)?\{\langle l,P_2,O_2\rangle\}:\phi$;

$\Gamma_{22}=\{O_2$ 在 P_2' 处的其他面$\}$;

$\Gamma_{21}=\{O_2$ 在 P_1' 处的其他面$\}$;

$\Gamma_{12}=\{O_1$ 在 P_2' 处的其他面$\}$;

$\Gamma_{11}=\{O_1$ 在 P_1' 处的其他面$\}$;

$S=\{\langle l,F,O_1\rangle|F\in\Gamma_{21}\cup\Gamma_{22}\}\cup\{\langle l_1,F,O_2\rangle|F\in\Gamma_{11}\}\cup\{\langle l_2,F,O_2\rangle|F\in\Gamma_{12}\}\cup S_1''\cup S_2'$;

(5) 否则//$e=\langle l,P,O\rangle$ //需求点与多面体的最早相交时刻

PO 过程:求 P 与 O 最早相交时刻 t_c' 及与之相交的多面体面 F

若 $t_c'<t_c$,则 $S=\{\langle l,F,O_1\rangle\}\cup S;t_c=t_c';R-\phi$;

(6) 转(2).

1.4 PO算法

在上述FCPF算法中,第(5)步需要求一个多面体(如O_1)上的顶点P与另一个多面体O(如O_2)的最早相交时刻t_c'及O发生相交的面.PO算法是这一过程的详细算法.

(1) $t_{max}=t_c$ //t_c为当前已知的最早相交时刻;

(2) 取O_2的一个面F,P在F的正面;

(3) 求P在(t_0,t_{max})时域内与F所在平面的最早相交时刻t',并求交点$P(t')$; //P一定与F相交,否则,P不与O_2相交;

(4) 取L为面F的边集;

(5) WHILE L不空

取$l\in L$

求l与$P(t')$的关系

如果$P(t')$在l外

取F因l而毗邻的面为F

$t_{max}=t'$

转(3);

(6) END WHILE;

(7) P最早与O_2的F相交,且相交时刻为t'.

2 算法分析

FCPF算法中求解l与F的最早相交时刻的过程及PO算法中求解P与F所在平面最早相交时刻的过程可由数值计算完成,且时间复杂性与多面体的复杂性无关.

2.1 算法的可终止性(收敛性)

在FCPF算法中会生成多个不同的t_c,我们将所有这些t_c按其产生的顺序为序组成一个序列$\{t_{c_n}\}$.由FCPF算法的终止条件可知,该算法的可终止性等价于该算法中$\{t_{c_n}\}$到T_c的收敛性.下面我们证明$\{t_{c_n}\}$可以在有限步内收敛到T_c.

命题 1. 对于同一个t_c,该算法只检测有限个状态.

证明:FCPF是一个循环过程,对于同一个t_c,算法所需循环的步骤数是与该t_c对应的集合S的元素个数.S的元素个数是一个有限数.当FCPF算法循环时,如果t_c不发生变化,则必减少S中的一个元素.因此,对于同一个t_c,FCPF算法只需循环有限步.也就是说,对于同一个t_c,该算法只检测有限个状态.

命题 2. $\{t_{c_n}\}$收敛.

证明:设当前$t_c=t_i$.由FCPF算法的第(4)步可知,如果$t_c'<t_c$,则用t_c'替代原来的t_c,即生成t_{i+1}.因此,对于所有的i,都有$t_i>t_{i+1}$,即$\{t_{c_n}\}$单调下降.另外,对于所有的t_i,均有$t_i>t_0$,即$\{t_{c_n}\}$有下界t_0.所以,$\{t_{c_n}\}$是一个单调下降且有下界的序列,必收敛.

命题 3. $\{t_{c_n}\}$必在有限步内收敛.

证明:在FCPF算法中,一个t_c是O_1(或O_2)的一条边l与O_2(或O_1)的面的最早相交时刻.不同的t_c对应于不同的边面对.因此t_c的个数不会多于O_1与O_2之间的边面对数.而O_1与O_2均为确定的多面体,其边数与面数都是确定的、有限的,所以,O_1与O_2之间的边面对数也是一个有限数.由此可以知道,t_c的个数是有限的,即$\{t_{c_n}\}$是一个有限序列,其收敛自然在有限步内完成.

由以上3个命题可知,$\{t_{c_n}\}$在有限步内收敛到T_c.

2.2 算法的时间复杂性

FCPF算法从第(2)~(6)步组成一个循环.整个循环过程是$\{t_{c_n}\}$到T_c的收敛过程,循环步数是$\{t_{c_n}\}$到T_c的收敛步数N_t.因为每个t_c对应于一个相交的边面对,因此,$\{t_{c_n}\}$到T_c的收敛步数N_t不会超过已相交的边面对

数.设检测到碰撞时,两个碰撞物体 O_1 和 O_2 发生相交的边数与面数分别为 n_1,m_1 和 n_2,m_2,则已相交的边面对数不超过 $(n_1\times m_2+n_1\times m_2)$.所以,$N_t\ll(n_1\times m_2+n_1\times m_2)$.设 $n=\max(n_1,n_2)$,$m=\max(m_1,m_2)$,则整个算法在最坏情况下的时间复杂性为 $O(n\times m)$.

一般来说,碰撞检测的时间步长 $dt=(t_1-t_0)$ 较小,所以两个物体之间发生相交的边和面自然也很少,因而 N_t 很小.即使在 dt 较大时,FCPF 算法仍然可以准确地计算出碰撞时间 t' 及所有碰撞点,而且可以有效地利用凸多面体的特性,使用 PO 算法加速收敛.

3　实现细节

FCPF 算法中以下两个求解过程可由数值方法完成:(1) 求解 O_1 中一条边 AB 与 O_2 中一个面 F 的最早相交时刻;(2) 求解 O_1 的顶点 P 与 O_2 一个面 F 的最早相交时刻.假设 O_1 与 O_2 是符合第 1.1 节中基本假设的两个凸多面体.AB 是 O_1 的一条边,F 是 O_2 的一个面.

3.1　坐标系

绝对坐标系:设绝对坐标系的原点为 O,3 个单位坐标向量分别为:i,j 和 k.

参考坐标系:参考坐标系的原点取 O_2 的重心(仍记为 O_2).当 $\omega_2=0$ 时,参考坐标系的 3 个单位向量不变:$i'=i,j'=j,k'=k$;$\omega_2\neq0$ 时,取 k' 为 ω_2 方向的单位向量,即 $K'=\omega_2/|\omega_2|$,i' 与 j' 则分别取与 k' 垂直的两个单位向量,且 i',j' 和 k' 构成右手坐标系.我们用上标 (R) 表示向量或点在参考坐标系中的表示.如:$r^{(R)}$,$P^{(R)}$ 即为向量和点在参考坐标系中的表示.

当 $\omega_2=0$ 时,参考坐标系为绝对坐标系的平移坐标系.当 $\omega_2\neq0$ 时,参考坐标系则为绝对坐标系绕平移轴的旋转坐标系.i',j' 和 k' 实际上是 t 的函数,我们记 i'_t,j'_t 和 k'_t 为 t 时刻参考坐标系的 3 个单位坐标向量.并用 i'_0,j'_0 和 k'_0 表示 t_0 时刻参考坐标系的 3 个单位坐标向量.

设:$[i'_0,j'_0,k'_0]^T=A_0[i,j,k]^T$,其中:

$$A_0=\begin{cases}\begin{bmatrix}a_x & a_y & a_z\\ b_x & b_y & b_z\\ \omega_{2x} & \omega_{2y} & \omega_{2z}\end{bmatrix}, & \text{如果 } \omega_2\neq0,\\ I_{3\times3}, & \text{如果 } \omega_2\neq0.\end{cases}\tag{1}$$

3.2　P,AB 及 F 在参考坐标系中的方程

多面体 O_1 的顶点 P 在参考坐标系中的运动方程为

$$P^{(R)}(t)=r_p^{(R)}-(\overrightarrow{O_2P}(t_c)+(v_1-v_2)(t-t_0)+\int_{t_0}^{t}\omega_1\times\overrightarrow{O_1P}dt)\cdot A_0^{-1}A_t^{-1},\tag{2}$$

A,B 两个顶点在参考坐标系中的运动方程分别为

$$A^{(R)}(t)=(\overrightarrow{O_2}(t_0)+(v_1-v_2)(t-t_0)+\int_{t_0}^{t}\omega_1\times\overrightarrow{O_1A}t)\cdot A_0^{-1}A_t^{-1},\tag{3}$$

$$B^{(R)}(t)=(\overrightarrow{O_2}(t_0)+(V_1-V_2)(t-t_0)+\int_{t_0}^{t}\omega_1\times\overrightarrow{O_1B}t)\cdot A_0^{-1}A_t^{-1},\tag{4}$$

线段 AB 于 t 时刻所在的直线在参考坐标系中的方程为

$$P^{(R)}(t)=A^{(R)}(t)\cdot[B^R(t)-A^R(t)]\cdot u,\tag{5}$$

其中 $t\in(t_0,t_1]$,$u\in[0,1]$,取 n 为 O_2 的面 F 的法向量,Q 为 F 的一个顶点,则 F 在参考坐标系中的方程为

$$n^{(R)}\cdot(P^{(R)}(t)-Q^{(R)})=0.\tag{6}$$

3.3　AB 与 F 的最早相交时刻

AB 与 F 的最早相交时刻即式(5)与式(6)联立式在区间 (t_0,t_1) 上使得 $u\in[0,1]$ 的最小解 t'.

3.4　求 P 与 F 所在平面的最早相交时刻

P 在 (t_0,t_{max}) 内与 F 所在平面的最早相交时刻 t' 是式(2)与式(6)联立式在 (t_0,t_{max}) 内的最小解 t'.

4 总 结

本文提出了一种计算凸多面体间碰撞点的快速算法.该算法可以快速求解出首次相碰的两个凸多面体间的精确碰撞时间和碰撞点,完成了基于物理特性的实时碰撞响应的一部分重要内容.本算法还有以下特点:(1) 检测到碰撞时,即使物体之间已发生很深的穿透,本算法仍能计算出物体之间的准确碰撞时间和碰撞点;(2) 当两个物体之间有多个碰撞点时,本算法可以找出所有的碰撞点;(3) 求解碰撞点时,不仅考虑了物体的平移,而且考虑了物体的旋转;(4) 因为一般的多面体都可以分解为若干凸多面体的组合,因此,本算法可以经过适当修改,应用于一般多面体的情形,具有很好的通用性.

参考文献

1 王兆其,赵沁平,汪成为.虚拟环境中物体物理特性的表示与处理.计算机研究与发展,1998,35(2):97～101
(Wang Zhao-qi, Zhao Qin-ping, Wang Cheng-wei. Study on representing and handling physical properties of objects in a virtual environment. Computer Research and Development, 1998,35(2):97～101)

2 Hahn J K. Realistic animation of rigid bodies. ACM SIGGRAPH, 1988,22(4):299～308

3 Lin Ming C. Efficient collision detection for animation and robotics [Ph. D. Thesis]. Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1993

A Fast Algorithm to Calculate Collision Point Between Convex Polygons

WANG Zhao-qi[1] ZHAO Qin-ping[2] WANG Cheng-wei[3]

[1](*Institute of Computing Technology The Chinese Academy of Sciences Beijing* 100080)

[2](*Department of Computer Science and Engineering Beijing University of Aeronautics and Astronautics Beijing* 100083)

[3](*Science and Technology Committee General Department PLA Beijing* 100034)

Abstract Collision point is basic information to physically-based collision response, but collision point finding is a burden task, which make real-time collision response very difficult. In this paper, the authors give an algorithm to find the very fast collision time and the collision points between two objects represented by convex polygon prior to collision response.

Key words Collision point, collision response, collision detection, virtual reality.