

# 基于矩形包围盒的多边形碰撞检测算法

周之平 张飒兵 吴介一 白伟冬

(东南大学 CIMS 中心, 南京 210096)

**摘要** 碰撞检测是计算机图形学领域中的一个普遍存在的问题。为了提高多边形碰撞检测的效率,针对简单形式刚性运动的多边形对象,提出了一种基于二维轴向矩形包围盒结构的平面简单多边形碰撞检测算法。该算法基于坐标轴的单调性对多边形进行分割,并通过矩形包围盒之间的预检来减少无关边对的相交测试,以加速算法的终止。由于采用轴向扫描线方法可以大大减少包围盒测试的数量和线段求交的数量,所以,经过少量的“边-边”相交判断就能求解到所有交点,同时能快速地获得两多边形干涉发生的第1位置。试验表明:(1)对于一般多边形,该算法的复杂度也远远低于 $O(N_P \times N_Q)$ ;(2)对于凸多边形对象,该算法的复杂度为 $O(N_P + N_Q)$ ,其中 $N_P, N_Q$ 为多边形 $P, Q$ 的顶点数。由此可见,算法能够获得较好的运算效率。

**关键词** 简单多边形 包围盒 预检测 碰撞检测

**中图分类号**: TP391.41 TP391.72 **文献标识码**: A **文章编号**: 1006-8961(2004)11-1294-10

## A Polygon Collision Detection Algorithm Based on Rectangular Bounding Volume

ZHOU Zhi-ping, ZHANG Sa-bing, WU Jie-yi, BAI Wei-dong

(The CIMS Center of Southeast University, Nanjing 210096)

**Abstract** Collision detection is a prevalent problem in computer graphics. A fast, accurate and feasible collision detection algorithm is important for an application. In this paper, a new planar simple polygon intersection algorithm, based on 2D axis-aligned bounding rectangle data structure, is presented for the polygons subjected to simple and unreformed movement. A new partition strategy for geometrical graphics according to the axial monotony, and the pre-checking process between 2D axis-aligned bounding rectangles reduce the number of unnecessary edge-pairs to be checked efficiently, so the algorithm can terminate promptly. After the partition along with coordinate axis, the interference checking between monotone chains proceeds. A novel search method based on sweep-line theory is adopted to eliminate the number of collision test for both segment-pairs and bounding volume-pairs drastically. So it can prompt the execution of algorithm. The accurate intersections, as well as the first occurrence of intersection between two objects subjected to a dynamic environment, are acquired by less arithmetical operation. The experimental results indicate that the complexity is far less than  $O(N_P \times N_Q)$  for generic polygons, even asymptotically  $O(N_P + N_Q)$  for two convex polygons, wherein  $N_P, N_Q$  denote the vertex number of two polygons  $P, Q$  respectively. So, it is a fast and efficient algorithm.

**Keywords** simple polygons, bounding volume, pre-check, collision detection

## 1 引言

计算机图形学、CAD/CAM、机器人学以及虚拟现实等领域的研究中,经常需要解决实体的移动问

题,如机器人的行走以及计算机辅助设计中图形的拼接、虚拟装配过程中物体的安装匹配等等。由于B-reps是表述三维实体的常用方法,因此针对用B-reps表示的三维物体,可以通过判定多边形的干涉来确定分析是否发生碰撞。设 $P$ 和 $Q$ 是平面内分别

具有  $N_P$  和  $N_Q$  条边的两个任意互不相交的简单多边形,则多边形干涉分析主要涉及以下几个问题:

(1)当  $P$  在平面内做某种运动时,是否与  $Q$  发生碰撞;(2)若  $P$  在平面内做某种运动时与  $Q$  碰撞,则可确定其最初的碰撞部位。

Edelsbrunner 提出了一种通过  $O(\log N_P + \log N_Q)$  同时次运算就可确定两个凸多边形的最近距离的算法<sup>[1]</sup>。Lin-Canny 通过建立一种层次结构树,运用几何的连贯性来逐步搜索最近距离,并通过  $O(1)$  次运算来获得最短距离<sup>[2]</sup>。文献<sup>[3]</sup>先采用“卷包裹”的方法来计算两个凸多边形并集的凸包,然后对新生成的两条“包络线”之间的边进行搜索,经过  $O(N_P + N_Q)$  次运算即可以找到交点的位置,但是上述这些方法只对凸多边形有效。

一种四叉树的非均匀层次分割碰撞检测算法<sup>[4]</sup>则先利用相对粒度尺寸来对物体进行非均匀分割,然后采用编码形式把发生干涉的孩子结点关联起来,再对产生干涉的结点逐次细分,直至到达某一精度为止,假如存在叶结点干涉,则判定多边形干涉。这种方法的优点在于能最大程度地减少对不会发生干涉的那些几何部分的细分,但是算法的准确性却受到分割精度的影响。通过对凹多边形进行三角化或凸化细分<sup>[5,6]</sup>,并通过求取三角形对或凸多边形对的交点来判定两个非凸多边形是否干涉也是一种比较常用的方法,但是由于以下 3 方面的原因:(1)细分一个凹多边形本来就是一个复杂的工作;(2)细分导致更多的边要做相交测试;(3)对产生的交点还必须进行点与凸多边形(三角形)的位置包含关系的测试,致使该算法的运算量大大增加,特别是在多边形含有较多的凹点,以及两多边形的交点数目比较多的情形时。Redon 提出一种连续时间的多边形碰撞检测方法<sup>[7]</sup>,但是运动方程的阶数不能大于 3,否则无法通过一般的计算方法求解。

本文通过对一般多边形进行结构分解,建立了一种基于链表与结点的混合数据结构,并采用二维包围盒预检测来尽可能地排除不发生相交的边,以减少边-边相交测试的次数和实现任意多边形之间的干涉判定。

## 2 理论基础

符号说明:

$P, Q$  分别为沿顶点序列走向的有向多边形;

$p_{top}, p_{bottom}$  分别为多边形  $P$  上最高点和最低点,  $q_{top}, q_{bottom}$  类似;

$P_{left}, P_{right}$  分别为多边形  $P$  的左、右分支,  $P_{left}$  为多边形  $P$  上逆时针方向从  $p_{top}$  到  $p_{bottom}$  之间边的集合,  $P_{right}$  为  $P$  上逆时针方向从  $p_{bottom}$  到  $p_{top}$  之间边的集合。  $Q_{left}, Q_{right}$  类似;

$p, q$  分别为  $P$  和  $Q$  上的一条关于  $Y$  坐标轴单调的折线,其包含了位于多边形  $P$  (或  $Q$ ) 上的一条有向边,或者彼此互相连接且具有相同单调性的多条有向边;

$V$  为多边形的顶点或边的端点;

$l = (V_{start} V_{end}) = \vec{V}_{start} \vec{V}_{end}$ ,  $l$  为有向边,其起点为  $V_{start}$ , 终点为  $V_{end}$ ;

$Box(\cdot)$  为几何对象的包围盒数据;

$Box(\cdot) = ([x_{min}, x_{max}], [y_{min}, y_{max}])$  代表几何对象的包围盒区域是由  $x = x_{min}, x = x_{max}, y = y_{min}$  和  $y = y_{max}$  4 条直线构成的封闭区域;

$E_P, E_Q$  为多边形  $P, Q$  所包含的有向边集合,即  $E_P = \{I_P\}, E_Q = \{I_Q\}$ ;

$x_p, y_p$  为点  $p$  的  $X, Y$  坐标分量。

### 2.1 简单多边形定义

简单多边形需满足下述条件:

(1)多边形的任意两条边,除共享顶点外,不能相交;(2)它是用若干直线段封闭起来的几何图形。本文将这类多边形称为简单直边多边形,为论述方便,文中简称为多边形。

### 2.2 多边形的旋向定义

设  $V_i$  为多边形的凸点,  $V_{i-1}, V_{i+1}$  是多边形顶点序列中与  $V_i$  相邻的两个顶点,设  $V_{mid}$  为  $V_{i-1}, V_{i+1}$  的中点,即  $V_{mid} = (V_{i-1} + V_{i+1})/2$ , 用于构造从  $V_i$  指向  $V_{mid}$  的向量  $\hat{V}$ , 若  $V_{i-1}$  位于向量的左侧,  $V_{i+1}$  位于向量  $\hat{V}$  的右侧,则称多边形为逆时针旋向,反之,称多边形为顺时针旋向(见图 1)。可用一包围盒(如图 1 虚线框所示)来界定多边形,并取  $V_i$  为落入矩形边界上的点。

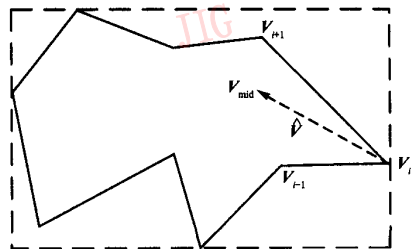


图 1 多边形的旋向判定

### 3 静态算法

#### 3.1 多边形的结构分解及其层次结构的建立

多边形的结构分解及其层次结构的建立步骤如下:

(1) 通过搜索多边形  $P$  中  $X, Y$  坐标方向具有最大最小坐标值的顶点来构造多边形的包围盒  $\text{Box}(P) = ([x_{\min}, x_{\max}], [y_{\min}, y_{\max}])$  (为研究方便起见, 假设构成多边形的两条链对  $Y$  轴的单调性比对  $X$  轴的单调性更好)。

(2) 设  $Y$  坐标最小最大的两点为  $p_{\text{bottom}}$  和  $p_{\text{top}}$ 。则构建多边形的步骤为: 首先根据多边形的走向, 以  $p_{\text{bottom}}$  和  $p_{\text{top}}$  为两个分离点, 将存放多边形顶点的循环链表  $L$  分割为  $S_{\text{left}}$  和  $S_{\text{right}}$  两部分, 其中,  $S_{\text{left}}$  中的这些有序点构成多边形  $P$  的左分支  $P_{\text{left}}$ ,  $S_{\text{right}}$  中的这些有序点构成多边形  $P$  的右分支  $P_{\text{right}}$ ; 然后建立两个新的链表  $L_1, L_2$ , 并将位于分支  $P_{\text{left}}$  (或  $P_{\text{right}}$ ) 上的各点存放在链表  $L_1$  (或  $L_2$ ) 中, 而顶点在链表  $L_1$  (或  $L_2$ ) 中的存放次序为: 沿着“从  $p_{\text{top}}$  出发, 终止于  $p_{\text{bottom}}$ ”的路线方向, 先将分支上的点有序地存放到链表中, 再根据  $L_1$  和  $L_2$  所包含的顶点数据计算, 并保留  $\text{Box}(P_{\text{left}})$  和  $\text{Box}(P_{\text{right}})$ 。此外, 还可根据  $L_1$  (或  $L_2$ ) 顶点存放的新次序来构造分支  $P_{\text{left}}$  (或  $P_{\text{right}}$ ) 所包含边的信息 (包含起止点的  $Y$  坐标变化趋势、包围盒数据、起止点索引值)。

(3) 对每一条分支  $P_{\text{left}}$  (或  $P_{\text{right}}$ ), 根据每条有向边的起止点坐标的变化趋势, 对分支所包含的边进行重组, 并将分支  $P_{\text{left}}$  (或  $P_{\text{right}}$ ) 重新表示为若干折线  $p$  的集合, 使每一条折线都满足对  $Y$  轴的单调性。对每条折线, 除了保存其包围盒数据外, 还记录所包含的边。

步骤(2)顶点的存放次序对于含有多条边的一对折线对测试很重要, 因为当这对折线的包围盒发生干涉时, 这种行进方向的限定可以大大减少包围盒测试的数量 (详细说明见 3.3.4 节中的分析)。

#### 3.2 基于包围盒层次结构的多边形碰撞检测算法

首先按 3.1 节对多边形  $P, Q$  进行结构分解, 并建立多边形层次结构, 然后进行以下操作:

(1) 判断是否满足  $\text{Box}(P) \cap \text{Box}(Q) \neq \emptyset$ , 若是, 转步骤(2); 否则, 返回结果为两个物体不相交, 退出。

(2) 进行两多边形的分支之间的测试。分支对的测

试顺序为  $(P_{\text{right}}, Q_{\text{left}}) \rightarrow (P_{\text{left}}, Q_{\text{right}}) \rightarrow (P_{\text{right}}, Q_{\text{right}}) \rightarrow (P_{\text{left}}, Q_{\text{left}})$ 。本文取一对分支  $(P_{\text{right}}, Q_{\text{left}})$  进行测试;

(3) 首先判断它们的包围盒是否重叠, 若不重叠, 转步骤(7), 然后再取其他的分支对测试; 否则转步骤(4);

(4) 选取分支对中的一对折线进行干涉测试, 设  $p, q$  分别为  $P_{\text{right}}, Q_{\text{left}}$  两折线集合中待处理的一对折线, 若  $\text{Box}(p) \cap \text{Box}(q) \neq \emptyset$ , 转步骤(5), 否则, 转步骤(6);

(5) 分以下两种情形, 分别处理, 以快速获得多边形的第 1 个交点;

① 假如  $p$  和  $q$  包含的线段数目都为 1, 则直接判断是否相交, 若相交, 则立即可以获得第 1 个交点, 返回结果为两多边形相交, 结束; 否则, 转步骤(6);

② 假如  $p$  (或  $q$ ) 包含的线段数目大于 1, 则按约定方向 (参见 3.3.4 节所述) 对折线的边步进, 以判断两折线是否相交, 若发生相交, 则返回两多边形相交, 结束; 否则, 转步骤(6);

(6) 判断当前分支对中所有的折线对是否都被测试过, 若是, 则转步骤(7); 否则, 转步骤(4);

(7) 若所有的分支对都被处理完, 则转步骤(8), 否则, 取下一对分支, 转向步骤(3);

(8) 判断多边形是否为包含关系, 若是, 则输出结果为包含关系; 否则输出两多边形不相交。

当步骤(4)中返回的结果为两个多边形相交时, 则采用步骤(4)中的类似方法来记录包围盒发生干涉的所有边对, 即可以求解到所有的交点。

### 3.3 算法所包括的主要实现环节

#### 3.3.1 点与有向边的位置关系判定

给定点  $V$  与有向边  $\overrightarrow{V_1V_2}$ , 若点  $V$  使得“有向链”  $V_1VV_2$  的转向为顺时针, 则点  $V$  在  $\overrightarrow{V_1V_2}$  的左侧 (图 2(a)); 若转向为逆时针, 则点  $V$  在  $\overrightarrow{V_1V_2}$  的右侧 (图 2(b)); 否则点  $V$  的坐标满足  $\overrightarrow{V_1V_2}$  的直线方程, 如果点  $V$  在  $V_1, V_2$  的坐标范围内时, 则点  $V$  在  $\overrightarrow{V_1V_2}$  有界区段上; 其他情况点  $V$  位于  $\overrightarrow{V_1V_2}$  的延长线上 (图 2(c))。

利用点与有向线段位置关系的判定方法可以很方便地确定点与多边形的包含关系。当多边形走向为逆时针时, 且点位于多边形某条边的右侧, 则可以判定点在多边形外部; 当多边形走向为顺时针时, 且点位于多边形某条边的左侧, 则可以判定点在多边形外

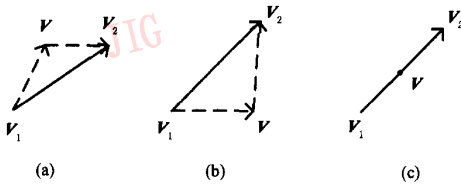


图 2 点与有向线段的位置关系

部,由此可见,其计算复杂性仅为边的线性函数。

**定理 1** 如果有向边  $l_1, l_2$  的包围盒不相交,则  $l_1, l_2$  不相交。

**证明:** 设线段  $l_1$  的两端点为  $A = (x_A, y_A)$ ,  $B = (x_B, y_B)$ , 线段  $l_2$  的两端点为  $C = (x_C, y_C)$ ,  $D = (x_D, y_D)$ 。为研究方便起见,不妨设  $x_A < x_B$ ,  $y_A < y_B$ ,  $x_C < x_D$ ,  $y_C < y_D$ , 且设包围盒在  $X$  方向不重叠,即  $x_{\max_1} < x_{\min_2}$  (不影响问题讨论的实质)。

$$x_{\min_1} = x_A, x_{\max_1} = x_B, y_{\min_1} = y_A, y_{\max_1} = y_B, \\ x_{\min_2} = x_C, x_{\max_2} = x_D, y_{\min_2} = y_C, y_{\max_2} = y_D.$$

假设  $l_1, l_2$  在有界范围内有交点, 设交点为  $p = (x, y)$ ,

则有  $x_{\max_2} \geq x \geq x_{\min_2} > x_{\max_1} \geq x$ , 即  $x > x$ , 与题设矛盾, 所以  $l_1, l_2$  必定不相交。

定理 1 可以扩充到一般几何图形对象, 例如, 折线, 多边形等等。

**定理 2** 若  $l_1$  (或者  $l_2$ ) 的两个端点都处于对方的左侧 (或者右侧), 则  $l_1, l_2$  必不相交。

**证明:** 设  $A$  和  $B$  分别为  $l_1$  的起点和终点,  $C$  和  $D$  为分别为  $l_2$  的起点和终点。记  $f_2(A), f_2(B)$  分别为  $A, B$  对应于  $l_2$  的位置符号,  $f_1(C), f_1(D)$  分别为  $C, D$  对应于  $l_1$  的位置符号。根据点与有向线段的位置关系判定 (详见 3.3.1 节), 做如下设定: 若  $A$  位于  $l_2$  的左侧, 则记  $f_2(A) > 0$ , 若  $A$  位于  $l_2$  的右侧, 则记  $f_2(A) < 0$ , 若  $A$  位于  $l_2$  的支撑线上, 则  $f_2(A) = 0$ 。

由题设知,  $f_2(B) \neq 0, f_2(A) \neq 0$ , 且  $f_2(B) \times f_2(A) > 0$ 。

假设  $l_1$  交  $l_2$  于点  $p$ , 则必有  $f_2(p) = 0$ 。

若  $p$  是  $l_1$  的顶点, 则说明  $p$  和  $A$  或者  $B$  重合, 即有  $f_2(A) \times f_2(B) = 0$ , 与题设矛盾。

若  $p$  不是  $l_1$  的顶点, 即  $p$  处于  $A, B$  之间, 则  $f_2(A) \times f_2(B) < 0$ , 即  $A, B$  处于  $l_2$  的异侧, 与题设矛盾, 所以  $l_1, l_2$  不相交。

**定理 3** 若  $l_1, l_2$  的两个端点都处于对方的异

侧, 则  $l_1, l_2$  必相交。

**证明:** 设  $A$  和  $B$  分别为  $l_1$  的起点和终点,  $C$  和  $D$  为分别为  $l_2$  的起点和终点。下面分两种情形进行分析。

(1) 如图 3 中的情形 (4)。记  $f_2(A), f_2(B)$  为  $A, B$  对应于  $l_2$  的位置符号,  $f_1(C), f_1(D)$  为  $C, D$  对应于  $l_1$  的位置符号。由题设知,  $f_2(A) \times f_2(B) < 0, f_1(C) \times f_1(D) < 0$ , 根据函数的连续性,  $l_1$  与  $l_2$  的支撑线交于点  $p$ 。同理,  $l_2$  与  $l_1$  的支撑线交于点  $q$ 。由于两条直线相交, 有且只有一个交点, 所以  $p, q$  必定重合, 也就是说,  $p, q$  都在  $l_1, l_2$  的限定区域上, 即交点为有效交点, 所以  $l_1, l_2$  相交。证明完毕。

(2) 图 3 中的情形 (6)。若  $l_1$  的两个端点都处于  $l_2$  的左右两侧, 而  $l_2$  有一个端点在  $l_1$  的支撑线上, 则  $l_1, l_2$  必相交。

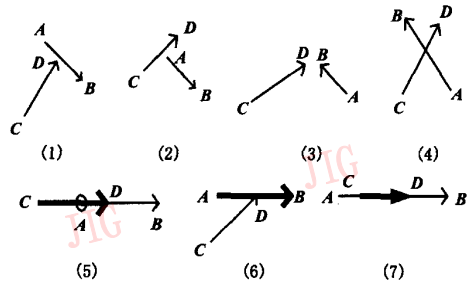


图 3 两线段相交关系判定的 7 种情形

记  $f_2(A), f_2(B)$  为  $A, B$  对应于  $l_2$  的位置符号,  $f_1(C), f_1(D)$  为  $C, D$  对应于  $l_1$  的位置符号。如, 因为  $f_2(A) \times f_2(B) < 0$ , 所以在  $l_1$  有限区段上上必然有一点位于  $l_2$  的支撑线上, 而由题设知,  $f_1(C) = 0$  (或者  $f_1(D) = 0$ )。由于两直线相交, 且只能有一个交点, 故  $C$  (或  $D$ ) 在  $l_1$  上。

**定理 4** 若  $l_1, l_2$  共线, 且包围盒重叠, 则  $l_1, l_2$  必相交。

由图 3 中的情形 (5) 和情形 (7) 可知, 显然  $l_1, l_2$  相交, 因为  $l_1$  (或  $l_2$ ) 至少有一个点位于  $l_2$  (或  $l_1$ ) 上。

### 3.3.2 矩形包围盒比较算法

设  $Box_1 = ([x_{\min_1}, x_{\max_1}], [y_{\min_1}, y_{\max_1}])$  和  $Box_2 = ([x_{\min_2}, x_{\max_2}], [y_{\min_2}, y_{\max_2}])$  为两个物体的包围盒数据。

如果两矩形框在  $X$  或者  $Y$  轴上的界定区间不发生重叠, 那么就可以判定两个包围盒不重叠, 即只要满足

$$x_{\min_1} > x_{\max_2}, x_{\max_1} < x_{\min_2}, y_{\min_1} > y_{\max_2}, y_{\max_1} < y_{\min_2}$$

这 4 个条件中的任意一个,就可以判定包围盒不重叠,也就是说,至多需要进行 4 次整型数的大小比较就能完成判断。

### 3.3.3 线段相交判断

设两条边分别为  $l_1, l_2$ , 其中  $A$  和  $B$  分别为  $l_1$  的起点和终点,  $C$  和  $D$  分别为  $l_2$  的起点和终点。若  $Box(l_1) \cap Box(l_2) = \emptyset$ , 则  $l_1$  与  $l_2$  不相交(如图 3 中情形(3)所示); 否则, 采用有向边形式描述, 通过边的顶点与有向边的位置关系来进行判定, 即可确定两条有向边是否相交。两条有向边的位置关系至多有如图 3 所示的 7 种情况。以下对之进行具体分析说明:

(1) 假如  $A, B$  位于  $l_2$  的同侧(或者  $C, D$  位于  $l_1$  的同侧), 且没有顶点位于对方的支撑线上时, 则  $l_1$  和  $l_2$  不相交, 如图 3 情形(1)和情形(2)所示。

(2) 假如  $A, B$  位于  $l_2$  的异侧, 而且  $C, D$  位于  $l_1$  的异侧,  $l_1, l_2$  相交, 如图 3 情形(4)和情形(6)所示(其中情形(6)是情形(4)的一种特殊情况, 交点位于线段的顶点上)。

(3) 如果  $l_1$  完全包含  $l_2$ (或者  $l_2$  完全包含  $l_1$ ), 则视为相交, 如图 3 情形(7)所示。

(4) 如果  $l_1$  部分包含  $l_2$ (或者  $l_2$  部分包含  $l_1$ ), 则视为相交, 如图 3 情形(5)所示。

(5)  $l_1, l_2$  没有重叠区域, 则  $l_1, l_2$  不相交, 如图 3 情形(3)所示。

由此可见, 图 3 中情形(1), 情形(2)和情形(3)不会产生交点, 情形(4)和情形(6)只有 1 个交点, 情形(5)和情形(7)则有无穷多个交点。

由于一次判断点与有向线段的位置关系需要 5 次加法和 2 次乘法操作, 所以判定两条直线的位置关系最多需要 20 次加法运算、8 次乘法运算以及 2 次比较操作。

特别地, 当两条线段包围盒干涉, 但是不相交时, 用点与边的位置关系来确定线段不相交比求解参数法更快, 因为当一条边的两个端点位于另一条边的同侧(同时没有顶点位于另一条边上时)而首先被检测到时, 就可以判定线段不相交, 而采用求解参数法来判断线段是否相交则必须通过 8 次乘法运算。

### 3.3.4 单调折线对的相交判定方法

设  $p$  和  $q$  分别是多边形  $P, Q$  某条分支的两条单调折线, 它们的包围盒之间发生干涉。为方便起见, 不妨设折线  $p$  属于  $P_{left}$ , 折线  $q$  属于  $Q_{right}$  (这不影响讨论问题的实质)。设当前测试的一对有向边为

$l_i$  和  $l_j$ , 其中  $l_i$  为折线  $p$  上的第  $i$  条有向边,  $l_j$  分别为折线  $q$  上的第  $j$  条有向边,  $i, j$  为索引变量,  $1 \leq i \leq m, 1 \leq j \leq n, m$  和  $n$  分别为折线  $p$  和  $q$  包含边的数目。  $V_i, V_{i+1}$  分别为有向边  $l_i$  的起点和终点,  $M_j, M_{j+1}$  分别为有向边  $l_j$  的起点和终点。如图 4 所示, 图中细实线表示折线  $p$ , 粗实线表示折线  $q$ , 虚线表示延长线。不妨假定  $y_{V_1} \geq y_{V_2} \geq \dots \geq y_{V_m} \geq y_{V_{m+1}}, y_{M_1} \geq y_{M_2} \geq \dots \geq y_{M_n} \geq y_{M_{n+1}}$ , 然后就可采用扫描线规则对两折线的边进行遍历, 步骤如下:

(1) 如果  $l_i$  和  $l_j$  的包围盒之间发生干涉, 则转步骤(3); 否则, 转步骤(2)。

(2) 用水平线扫描  $l_i, l_j$  的端点,

①若  $y_{V_{i+1}} > y_{M_{j+1}}$ , 则折线  $p$  的边前进一步, 即  $i = i + 1, j$  不变, 如图 4(a)所示。

②若  $y_{V_{i+1}} < y_{M_{j+1}}$ , 则折线  $q$  的边前进一步, 即  $i$  不变,  $j = j + 1$ , 如图 4(b)所示。

③若  $y_{V_{i+1}} = y_{M_{j+1}}$ , 则折线  $p$  和  $q$  的边都前进一步, 即  $i = i + 1, j = j + 1$ , 如图 4(c)所示。

(3) 用定理 3, 定理 4 判定有向边  $l_i, l_j$  是否相交, 若相交, 则返回结果, 结束; 否则用步骤(2)中的方法确定有向边的步进。

(4) 若  $i > m$  或  $j > n$ , 则两折线不相交, 结束; 否则转步骤(1)继续测试。

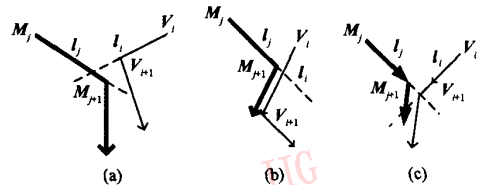


图 4 单调折线对的相交判定方法

下面对于步骤(2)中的①说明交点只能发生在索引  $i, j$  之后的有向边对上的这个规则给予详细证明。

对于图 4(a)而言, 设索引值分别为  $i = S, j = U$  的边  $l_S$  和  $l_U$  不相交, 假设交点  $p$  发生在  $i = S, j = U$  (其中  $U > W$ ) 的边对  $(l_S, l_U)$  上, 那么, 可能的相交情形如图 5 所示。显然, 由于  $y_{M_{U+1}} \geq y_{V_{S+1}} > y_{M_{W+1}}$ , 因此对于处于折线段  $p$  上的从  $M_{W+1}$  到  $M_{U+1}$  部分在  $Y$  方向不是单调的, 即整段折线  $p$  也不是单调的, 与题设矛盾, 所以交点只可能发生在  $i > S$  的边上, 即步进  $i$ 。

对于步骤(2)中的②和③两种情形, 同理可以证明。

假设  $i, j$  的初始值取为  $i = 1, j = 1$ , 因为每进行一次判定都导致  $i$  的值增加 1 或者  $j$  的值增加 1, 所

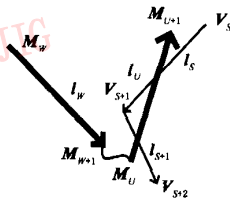


图 5 单调折线对的边的步进规则

以再至多进行  $m+n-1$  次判定就可以使得  $i>m$  或  $j>n$ , 这样由于通过至多  $4 \times (m+n)$  次点与有向边的位置测定, 就可以判定两折线段是否相交, 从而大大减少了测试次数。

当折线  $p$  (或者  $q$ ) 从起点到终点的  $Y$  坐标递增时, 反向遍历折线顶点同样可以实现快速相交测试。

### 4 动态算法

这里只考虑多边形  $P, Q$  做匀速直线运动和匀角速度旋转运动的情况。下面针对各种运动情况进行分析研究。

#### 4.1 多边形 $Q$ 静止, 多边形 $P$ 做直线运动

设运动速度为  $v$ ,  $v$  在  $X, Y$  方向的运动速度分量为  $v_x, v_y$ 。将仿真区间离散化为  $0, T, 2T, \dots, K \times T$ , 多边形  $P$  运动到某个时刻  $T_k = kT$  时, 多边形  $P$  上点  $p$  运动后为  $\hat{p}, \hat{p} = p + vT_k$ , 令  $\text{Box}(\hat{P}), \text{Box}(Q)$  分别为多边形  $\hat{P}, Q$  的矩形包围盒 ( $\hat{P}$  对应  $P$  运动到  $T_k$  时刻的几何数据,  $0 < k \leq K$ )。

假如  $\text{Box}(\hat{P})$  与  $\text{Box}(Q)$  不相交, 则终止, 否则构建其左右分支  $\hat{P}_{\text{right}}, \hat{P}_{\text{left}}, Q_{\text{right}}, Q_{\text{left}}$ , 并构造其包围盒  $\text{Box}(\hat{P}_{\text{right}}), \text{Box}(\hat{P}_{\text{left}}), \text{Box}(Q_{\text{right}}), \text{Box}(Q_{\text{left}})$ 。对包围盒重叠的分支对, 应首先更新顶点数据和各折线段的包围盒数据。包围盒  $\text{Box}_1$  的坐标数据  $([x_{\min}, x_{\max}], [y_{\min}, y_{\max}])$ , 经过运动后, 其坐标数据变为  $\hat{\text{Box}}_1 = ([x_{\min} + v_x T_k, x_{\max} + v_x T_k], [y_{\min} + v_y T_k, y_{\max} + v_y T_k])$ , 其所进行的操作只是在各个坐标分量上增加某一数值, 增量值为时刻  $T_k$  和坐标轴方向的速度分量的乘积。

#### 4.2 多边形 $P, Q$ 都做直线运动

设多边形  $P, Q$  运动的速度分别为  $v_1, v_2$ , 则假定  $T_0 = 0$ , 无碰撞,  $T_k = kT$  时, 多边形  $Q$  相对不动, 多边形  $P$  的相对运动速度为  $\hat{v}_1 = v_1 - v_2$ , 其余同 4.1 节。

#### 4.3 多边形 $P$ 绕原点旋转, 多边形 $Q$ 为静止不动时

此时, 在进行离散化和近似处理时, 要求采样时间

步长  $T$  短, 且旋转的角速度  $\omega$  很小 (即  $\omega, T$  趋近于 0)。

设  $T_0 (=0)$  时刻多边形  $P$  上任意一点  $p$  的坐标为  $(x_p, y_p)$ ,  $T_k$  时刻  $p$  点的坐标为  $(x_p^{(p)}, y_p^{(p)})$ 。

由坐标变换知:

$$x_{T_k}^{(p)} = x_p \cos(\omega T_k) - y_p \sin(\omega T_k) \quad (1)$$

$$y_{T_k}^{(p)} = x_p \sin(\omega T_k) + y_p \cos(\omega T_k) \quad (2)$$

根据式(1), 式(2), 可以很快地知道每运动到一个新位置, 物体顶点坐标在  $X, Y$  轴方向的增量  $\Delta x$  和  $\Delta y$ , 进而根据递推关系很容易获得下一时刻的坐标位置。

因为

$$\begin{aligned} \Delta x &= x_{T_{k+1}}^{(p)} - x_{T_k}^{(p)} \\ &= x_p \{ \cos(\omega T_{k+1}) - \cos(\omega T_k) \} - \\ &\quad y_p \{ \sin(\omega T_{k+1}) - \sin(\omega T_k) \} \end{aligned}$$

一阶泰勒近似, 所以有

$$\begin{aligned} \Delta x &= - \frac{(2k+1)\omega^2 T^2}{2} x_p - \omega T \times y_p \\ \Rightarrow x_{T_{k+1}}^{(p)} &= x_{T_k}^{(p)} + \Delta x \end{aligned} \quad (3)$$

因为

$$\begin{aligned} \Delta y &= y_{T_{k+1}}^{(p)} - y_{T_k}^{(p)} \\ &= x_p \{ \sin(\omega T_{k+1}) - \sin(\omega T_k) \} + \\ &\quad y_p \{ \cos(\omega T_{k+1}) - \cos(\omega T_k) \} \end{aligned}$$

一阶泰勒近似, 所以有

$$\begin{aligned} \Delta y &= - \frac{(2k+1)\omega^2 T^2}{2} y_p + \omega T \times x_p \\ \Rightarrow y_{T_{k+1}}^{(p)} &= y_{T_k}^{(p)} + \Delta y \end{aligned} \quad (4)$$

由此可见, 一旦多边形  $P$  确定, 采用增量算法<sup>[8]</sup>来对原有的包围盒旋转后的图形数据进行坐标比较, 就可以很方便地对折线的包围盒数据进行更新。由于考虑到物体的运动量很小, 所以能够获得比较近似的效果。可见, 只要保存上一个采样时刻的信息就可以获取当前时刻的数据, 否则, 假如每次运动一个时间步长都要比较顶点新近的坐标值和重新构建包围盒, 那么所花费的运算时间将很大 (包括坐标比较、折线重新分配等运算处理)。

#### 4.4 多边形 $P, Q$ 都绕原点做旋转运动

设多边形  $P, Q$  绕原点旋转的角速度分别为  $\omega_P, \omega_Q$  ( $\omega_P, \omega_Q$  按照旋转方向, 有正负值之分)。由于在同一坐标系中, 引用质点相对运动的理论, 可把多边形  $P$  和  $Q$  都当作一个质点, 因此, 可以知道多边形  $P$  相对于多边形  $Q$  的旋转角速度为  $\omega_P - \omega_Q$ , 其余步骤同 4.3 节。

以下证明之:

设  $l_1$  为原始多边形  $P$  上的一条有向边,  $A$  为起点,  $B$  为终点, 则  $l_1$  可用参数形式表示为

$$l_1 = A + (B - A)t, t \in [0, 1]$$

设  $l_2$  为原始多边形  $Q$  上的一条有向边,  $C$  为起点,  $D$  为终点, 则  $l_2$  可用参数形式表示为

$$l_2 = C + (D - C)s, s \in [0, 1]$$

经过绕原点的旋转运动后, 则

$$\begin{aligned} \hat{l}_1 &= \hat{A} + (\hat{B} - \hat{A})t \\ &= AG + (BG - AG)t \\ &= \{A + (B - A)t\}G \end{aligned}$$

其中  $t \in [0, 1], G = R(\omega_P \times T_k)$

$$\begin{aligned} \hat{l}_2 &= \hat{C} + (\hat{D} - \hat{C})s \\ &= CH + (DH - CH)s \\ &= \{C + (D - C)s\}H \end{aligned}$$

其中  $s \in [0, 1], H = R(\omega_Q \times T_k), G, H$  为坐标旋转变换矩阵。

假如  $\hat{l}_1, \hat{l}_2$  相交, 则交点  $p$  同时在  $\hat{l}_1, \hat{l}_2$  上, 即有

$$\{A + (B - A)t\}G = \{C + (D - C)s\}H$$

由于  $A, B$  为坐标旋转变换矩阵, 所以  $G, H$  可逆, 即有

$$\{A + (B - A)t\}GH^{-1} = \{C + (D - C)s\}$$

所以只要将多边形  $P$  以角速度  $\omega_P - \omega_Q$  旋转, 就可以实现两运动多边形的干涉判定, 而不必对多边形  $Q$  坐标进行旋转变换运算。

**4.5 多边形  $P$  做直线运动, 多边形  $Q$  绕原点做旋转运动**  
多边形  $P$  做直线运动(速度为  $v$ ), 多边形  $Q$  以角速度  $\omega_Q$  旋转。

可以等效处理为: 多边形  $P$  先以速度  $v$  直线运动, 然后做角速度为  $-\omega_Q$  的旋转运动。

证明过程如下:

设  $l_1$  为原始多边形  $P$  上的一条有向边,  $A$  为起点,  $B$  为终点, 则  $l_1$  可用参数形式表示为

$$l_1 = A + (B - A)t, t \in [0, 1]$$

设  $l_2$  为原始多边形  $Q$  上的一条有向边,  $C$  为起点,  $D$  为终点, 则  $l_2$  可用参数形式表示为

$$l_2 = C + (D - C)s, s \in [0, 1]$$

经过绕原点的旋转运动后的  $l_2$  为

$$\begin{aligned} \hat{l}_2 &= \hat{C} + (\hat{D} - \hat{C})s \\ &= CH + (DH - CH)s \\ &= \{C + (D - C)s\}H \end{aligned}$$

其中  $s \in [0, 1], H = R(\omega_Q \times T_k)$

经过直线运动后,  $l_1$  的参数方程为

$$\hat{l}_1 = A + (B - A)t + vT_k$$

若  $\hat{l}_1$  与  $\hat{l}_2$  相交, 则交点  $I$  同时在  $\hat{l}_1, \hat{l}_2$  上, 即有

$$A + (B - A)t + vT_k = \{C + (D - C)s\}H$$

即有

$$\{A + (B - A)t + vT_k\}H^{-1} = \{C + (D - C)s\}$$

其中  $H = R(-\omega_Q \times T_k)$

这样就可以将多边形  $P$  先以速度  $v$  做直线运动, 然后做角速度为  $-\omega_Q$  的旋转运动, 而不必对多边形  $Q$  坐标进行旋转变换运算。

对于运动形式复杂的情形, 可通过低阶多项式和离散化处理, 将其化解为简单的表达形式。采用上述方法, 同样可以获取碰撞发生的第 1 时刻和确切位置。

## 5 算法的复杂度分析与试验验证

### 5.1 算法的复杂度分析

假设多边形  $P, Q$  中边的数目分别为  $N_P$  和  $N_Q$ ,  $P_{\text{left}}$  和  $P_{\text{right}}$  中边的数目分别为  $M_1$  和  $M_2$ ,  $Q_{\text{left}}$  和  $Q_{\text{right}}$  中边的数目分别为  $N_1$  和  $N_2$ ,  $P_{\text{left}}$  和  $P_{\text{right}}$  所包含折线的数目分别为  $F_1$  和  $F_2$ ,  $Q_{\text{left}}$  和  $Q_{\text{right}}$  包含折线的数目分别为  $F_3$  和  $F_4$ 。设  $a_1, a_2, \dots, a_{F_1}$  为  $P_{\text{left}}$  中各单调折线所包含边的数目,  $b_1, b_2, \dots, b_{F_2}$  为  $P_{\text{right}}$  中各单调折线所包含边的数目, 同理  $c_1, c_2, \dots, c_{F_3}$  为  $Q_{\text{left}}$  中各单调折线所包含边的数目,  $d_1, d_2, \dots, d_{F_4}$  为  $Q_{\text{right}}$  中各单调折线所包含边的数目, 则包围盒测试的总的数量为  $(F_1 + F_2) \times (F_3 + F_4)$ 。一般说来, 因为当  $M, N$  值很大时,  $F_1 \ll M_1, F_2 \ll M_2, F_3 \ll N_1, F_4 \ll N_2$ , 所以包围盒测试的次数较少, 远小于  $O(N_P \times N_Q)$ 。最复杂情形, 当  $P, Q$  为星形多边形时, 由于多边形  $P$  与  $Q$  距离很近, 且位于其上的任意相邻两条边都不满足沿坐标轴方向的单调性时, 分割导致  $F_1 \approx M_1, F_2 \approx M_2, F_3 \approx N_1, F_4 \approx N_2$ , 因此, 此时包围盒测试的复杂度为  $O(N_P \times N_Q)$ 。

采用包围盒预检测, 可以排除大部分无关边对的对相交测试。对于包围盒发生干涉的折线对可采用 3.3.4 节的方法, 对于包含多于一条边的一对折线, 由于线性次数的线段相交测试就可以判断是否相交, 从而大大提高了算法的速度。

特别地, 当  $P, Q$  都为凸多边形时, 应考虑最坏的情形。若所有分支对的包围盒都产生干涉, 则应首先进行多边形的包围盒之间的测试和 4 次分支对测试。对每对包围盒干涉的分支对, 由于各自只有一条单调折线, 通过线性次比较就可以确定是否发生相

交,所以总的包围盒测试次数为  $O(N_P+N_Q)$ ,线段相交测试次数为  $O(N_P+N_Q)$ ,可见算法能够获得很好的效率。

### 5.2 静态分析

下面称本文提到的算法为方法 1。

另一种方法是先将多边形  $P$  中的每条有向边  $l_P$  与多边形  $Q$  进行包围盒预检测,同时将多边形  $Q$  中的每条有向边  $l_Q$  与多边形  $P$  进行包围盒测试,并保留  $E_P$  中那些满足  $Box(l_P) \cap Box(Q) \neq \emptyset$  的边集  $\hat{E}_P$  和  $E_Q$  中满足  $Box(l_Q) \cap Box(P) \neq \emptyset$  的边集  $\hat{E}_Q$ ,其中  $\hat{E}_P \subseteq E_P, \hat{E}_Q \subseteq E_Q$ ;然后对  $\hat{E}_P$  和  $\hat{E}_Q$  的边直接进行“边对”相交判断。本文称这种方法为方法 2。

第 3 种方法是先将多边形  $P$  中的每条有向边  $l_P$  与多边形  $Q$  进行包围盒预检测,同时将  $Q$  中的每条有向边  $l_Q$  与多边形  $P$  包围盒测试,保留  $E_P$  中满足  $Box(l_P) \cap Box(Q) \neq \emptyset$  的边集  $\hat{E}_P$  和  $E_Q$  中满足

$Box(l_Q) \cap Box(P) \neq \emptyset$  的边集  $\hat{E}_Q$ ,其中  $\hat{E}_P \subseteq E_P, \hat{E}_Q \subseteq E_Q$ ,然后仅对  $\hat{E}_P$  和  $\hat{E}_Q$  中的那些包围盒发生干涉的边对再进行相交判断,本文称这种方法为方法 3。

方法 2 和方法 3 在文献[9]中被提到,用来处理三维空间的多面体干涉检测。

对本文算法进行分析验证的结果如表 1、表 2、表 3 所示。该算法程序用 Visual C++ 6.0 编写,在 P IV 1.7GHz,256MB 内存下的 Windows 2000 系统环境中运行,对一对平面多边形  $P, Q$  进行了测试,共进行 1 000 次测试,取其平均值作为研究结果。表 1 针对两个随机产生的凹多边形模型进行比较,并就首次检测到碰撞的代价而论,将本文提到的算法与其他方法进行性能比较。表 2 对两形状相同的,而中心点位置存在偏移的星型多边形模型进行测试,就求解所有交点的代价,将本文提到的算法与其他方法进行性能比较。表 3 就拓扑结构相同而中心位置

表 1 检测第 1 个交点的平均代价(对两随机模型)

多边形 $P, Q$		方 法					
		方法 1		方法 2		方法 3	
顶点数 $N$	交点数	包围盒比较 (次)	线段求交 (次)	包围盒比较 (次)	线段求交 (次)	包围盒比较 (次)	线段求交 (次)
6	4	9.79	2.20	12	4.72	16.72	2.77
10	6	9.67	2.10	20	6.68	26.63	2.93
15	10	9.74	2.08	30	7.97	37.97	3.08
20	15	9.92	2.05	40	9.23	49.23	3.04
30	27	10.02	2.05	60	11.25	71.25	3.18
40	40	10.11	2.05	80	11.39	91.39	3.11
50	54	10.13	2.02	100	12.35	112.35	3.00

表 2 求解所有交点所需要的平均代价(对相同形状的星型多边形)

多边形 $P, Q$		方 法					
		方法 1		方法 2		方法 3	
顶点数 $N$	交点数	包围盒比较 (次)	线段求交 (次)	包围盒比较 (次)	线段求交 (次)	包围盒比较 (次)	线段求交 (次)
6	4	29	12	12	36	48	12
10	6	69.02	20.53	20	100	120	20.53
15	10	170.2	32.14	30	208.53	238.53	32.14
20	15	293.73	57.46	40	364.96	404.96	57.46
30	27	718.0	120.79	60	826	886	120.79
40	40	1338.67	207.27	80	1457.07	1537.07	207.27
50	54	2162.34	320.57	100	2279.5	2379.5	320.57

表 3 求解所有交点所需要的代价(对相同形状的椭圆型凸多边形)

多边形 $P, Q$		方 法					
		方法 1		方法 2		方法 3	
顶点数 $N$	交点数	包围盒比较 (次)	线段求交 (次)	包围盒比较 (次)	线段求交 (次)	包围盒比较 (次)	线段求交 (次)
6	4	29	8.81	12	24.47	36.47	8.81
10	6	44.29	10.03	20	55.74	77.74	10.03
15	10	63.23	10.84	30	108.49	138.49	10.84
20	15	82.03	12.09	40	182.81	222.81	12.09
30	27	119.67	13.27	60	379.35	439.35	13.27
40	40	157.35	14.25	80	650.38	730.38	14.25
50	54	194.84	15.14	100	988.14	1088.14	15.14

不相同的两椭圆形凸多边形进行测试比较,其中心位置的偏移是随机产生的。为分析方便起见,不妨设  $N_P = N_Q = N$ , 这不影响问题研究的实质。

对上述试验结果的分析:

从表1数据可知,采用新的基于包围盒和几何单调性的算法不仅可以大大减少线段求交的次数,并能快速找到首次相交点,其线段相交的次数基本上是一个常数,不仅远小于方法2的代价,而且包围盒计算的次数也很少。与方法3相比,线段对相交判定工作量降低50%。由于方法2检测过程过于粗略,因而导致进行很多不相交的线段对相交判定;而方法3其实很不合理,尽管其线段对相交判定次数很少,但是由于随着边数目的增加,进行两次不相干线段对排除所做的包围盒测试将远远大于线段相交判定所花费的代价,因此,本文方法所花费的计算量是可以接受的,而且受多边形边数目的影响不是很大,具有近似线性的复杂度。

表2中的测试的原型多边形的顶点是从一对同心圆上依次采样得到的。从表2数据可知,本算法求解所有交点所做的包围盒测试代价比较近似为  $O(N^2)$ 。这主要是由以下两个原因造成的:(1)交点数目多,可能发生干涉的边对随之增多;(2)多边形呈“星型”,导致分割后生成的单调分支数目很大。与方法2相比,本文方法线段对相交判定运算量减少80%以上,而包围盒测试运算量虽然近似为  $N^2$ ,但是,因为至多进行4次比较运算就能完成,所以算法效率远远胜于方法2。方法1求解所有交点所需的工作与方法3是相同的,虽都是针对包围盒干涉的边对进行相交判定,但是方法1所需的包围盒干涉测试次数比方法3减少9~28%,特别是多边形边的数目不多时( $N \leq 30$ ),性能更好。方法1是用点与有向边的位置来判断两干涉边对的交点是否为有效交点后,再用参数求解。这样,当两条不共线的边不相交时,经过两次位置比较就能够快速排除,由于这在一定程度上避免了不必要的参数计算,从而提高了判断的效率。

表3中的测试的原型多边形的顶点是从对椭圆均匀采样得到的。由于其具有较好的单调性,所以交点数目很少,一般为2~10。从表3数据可知,方法1在求解所有交点时,包围盒开销近似为  $4N$ ,且当  $N$  较大时,线段相交判定的次数远远小于  $N^2$ 。同时当交点较少时,随  $N$  值增加,变化不大;而方法2的线段对相交判定次数太大,显然不满足实际应用要求。

方法3虽然具有和本文算法相同的线段对相交判定复杂度,但是随着边数目的增多,其包围盒测试数量比方法1高出4~5倍(这是由于一次边与多边形的包围盒对进行预检测随机匹配造成的),可见本文方法性能上优于方法3。

### 5.3 动态分析

选取一连续运动时间  $[0, t]$ , 离散为  $[0, T, 2T, \dots, KT]$ , 对每个采样时刻进行分析判断,即预先对几何图形进行分割,建立层次包围树结构,并保存最初时刻的几何信息。整个算法中只要一次这样的计算即可。根据这些信息就可以非常容易地获得各个采样时刻的包围盒坐标数据,以及顶点的坐标信息。对于两个都发生运动的多边形来说,利用相对运动的观点,可以将运动多边形看作是一个物体相对另一个物体作旋转和平移运动,所以只要对其中一个图形进行坐标变换即可,同时只对发生相交的分支考虑其折线段的包围盒更新。对于平移运动,由于包围盒变换和顶点只需要做简单的数值“加”操作,其判断方法类似于静态,故运算量较小。对于旋转运动,各个采样时刻顶点的坐标可以经过公式(3),公式(4)求取,且根据运动的连续性,只要选取合理的采样步长和运动参数(速度和角速度),就能很容易地判断两个瞬间碰撞状态是否变化,也就可获取干涉发生的第1时间。

由于算法的准确性和高效性与存储需求是互相矛盾的,因此预处理过程必须尽可能的简单有效,否则反而会增加算法的复杂性。本文提出的包围盒方法,由于仅仅是通过坐标比较就能判定结果,其远远小于一次线段求交的计算量,因而对多边形进行基于轴向的单调性分割,在某种程度上可以大大减少包围盒测试的次数。

### 参考文献

- 1 Edelsbrunner H. Computing the extreme distances between two convex polygons[J]. Journal of Algorithms, 1985, 6(2): 213~224.
- 2 Lin M C, Canny J F. A fast algorithm for incremental distance calculation [A]. In: Proceedings of IEEE International Conference on Robotics and Automation[C], Sacramento, CA, USA, 1991: 1008~1014.
- 3 Toussaint G T. A simple linear algorithm for intersecting convex polygons[J]. The Visual Computer, 1985, 6(1): 118~123.
- 4 吴明华,余勇翔,周济. 采用空间分割技术的八叉树干涉检测算法[J]. 计算机学报, 1997, 20(6): 849~854.
- 5 Narkhede A, Manocha D. Fast polygon triangulation based on

Sleidel's algorithm(Graphics Gems V)[M]. Boston NJ, USA: Academic Press, 1995:394~397.

- 6 普雷帕拉塔(美),沙莫斯(美)著,庄心谷译. 计算几何导论[M]. 北京: 科学出版社, 1990.
- 7 Redon S, Kheddar A, Coquieeart S. An algebraic solution to the problem of collision detection for rigid polyhedral objects[A]. In: Proceedings of IEEE International Conference on Robotics and Automation[C], San Francisco, CA, USA, 2000: 3733~3738.
- 8 Klosowski J T, Held M. Efficient collision detection using bounding volume Hierarchies of k-DOPs[J]. IEEE Transactions on Visualization and Computer Graphics, 1998,4(1):21~36.
- 9 Zachmann G. Exact and Fast Collision Detection [D]. PhD thesis, University of Bonn, German, 1994.



**张斌兵** 1964年生,1986年毕业于南京工学院。高级工程师,主要研究方向为集成制造环境。



**吴介一** 1941年生,教授、博士生导师。1963年毕业于南京工学院自动控制专业。主要研究方向为计算机网络和先进制造技术。



**周之平** 1975年生,博士研究生。主要研究方向为虚拟装配和计算机图形学。1998年毕业于南昌大学,获自动控制专业学士学位。2002年于东南大学自动控制系提前攻读博士学位。

E-mail:zhousam@seu.edu.cn



**白伟冬** 1980年生,硕士研究生,2001年毕业于东南大学自动控制系。主要研究方向为虚拟装配交互性研究。