

文章编号: 1001-2486 (2000) 01-0015-05

一种基于 TIN 表达的碰撞检测方法研究*

余龙华, 沈林成, 常文森

(国防科技大学机电工程与自动化学院, 湖南 长沙 410073)

摘要: 在扼要介绍有关碰撞检测方法的基础上, 着重研究一种基于 TIN (Triangulated Irregular Network) 表达的物体之间的几何碰撞检测方法。有效地解决了其中的完整性控制问题, 并在路径规划等系统中得以成功运用。

关键词: TIN; 几何碰撞; 碰撞检测

中图分类号: TP242.6 **文献标识码:** A

A TIN-Based Collision Detection Method Research

SHE Long-hua, SHEN Lin-cheng, CHANG Wen-sen

(College of Mechatronics Engineering and Automation, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: After the description of some collision detection methods, we present a new method for collision detection for geometric models described by TIN. This method solves the problem of completeness control effectively. It has been used successfully in our route planning system.

Key words: TIN; geometry collision; collision detection

碰撞检测在地面车辆自主导航、机器人路径规划、虚拟现实场景中的多体运动协调控制及三维动画等领域有着广泛的应用价值。对于导航或规划系统, 碰撞检测的目的在于: 提供避碰策略。对于多体运动问题, 其目的在于: 提供必要的碰撞信息, 为计算碰撞后的物体行为提供依据。碰撞检测方法较大幅度地依赖物体的表达方式、相对运动的表现形式以及实时性要求等方面。文献[1]研究了时变参数曲面的碰撞检测方法, 文献[2]采用最小平移距离的概念研究了凸多面体的碰撞检测原理, 文献[6]介绍了形变、自交曲面的碰撞检测算法, 文献[7]利用区间数学和递归细分原理, 探讨了隐函数曲面的碰撞检测方法。本文着重研究基于 TIN 表达的物体之间的几何碰撞检测方法。

TIN 表达是指把物体表面剖分为三角形网格, 并以之表示该物体的几何形状。几何碰撞是指两个存在相对运动的物体所发生的零距离接触。关于 TIN 碰撞检测问题, 有少量文献^[8,9]作过研究, 但它们在求解的完整性控制方面存在严重不足。也就是说, 它们不能检测到某些形式下的碰撞。文献[8]是基于如下方法: 在每一个时间步长内检测 TIN 表达中的每一个顶点是否穿越 TIN 表达中的任何一个三角形, 如果穿越, 则碰撞发生。显然, 这种方法不能检测到边与边之间发生的碰撞。文献[9]提供的方法实际上只能检测到点与面、边与边之间发生的碰撞。本文将解决 TIN 碰撞检测的完整性控制问题, 同时也讨论了如何避免冗余解。

1 TIN 碰撞检测问题分析

当物体采用 TIN 表达时, 构成物体最基本的图元是三角形。因此, 当需要判断两个基于 TIN 表达的物体在运动过程中是否发生碰撞时, 只需判断 TIN 表达中存在相对运动的每两个三角形之间是否发生碰撞。

从不同的层次看, 三角形可理解为点 (三角形顶点)、边 (三角形边)、面 (三角形所界定的平面

* 收稿日期: 1999-08-30

作者简介: 余龙华 (1979), 男, 讲师, 博士。

区域)。因此,两个三角形之间发生碰撞的形式可能有:点-点碰撞、点-边碰撞、点-面碰撞、边-边碰撞、边-面碰撞和面-面碰撞。这六种形式可分别如图1所示。

因此,在三维空间中求解两个运动三角形之间的碰撞检测问题时,必须考虑以上六种可能出现的情形。但实际情况要更复杂一些,我们需要对以上六种形式作进一步归纳分析,以避免冗余解的出现。

首先,如果把三角形理解为包括顶点和边在内的有界平面闭区域,则(a)、(b)、(c)三种形式可归纳为点与三角形之间的碰撞,记之为(abc)。可以肯定(abc)形式碰撞的解是唯一的点。再次,我们可以发现:对于(f),即面-面碰撞,它同时可能包含(e)、(d)、(abc)三种形式的碰撞;对于

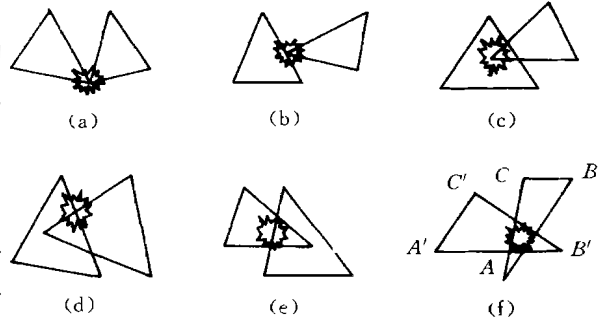


图1 运动三角形之间的碰撞形式
Fig.1 Collisions between moving triangulars

(e),即边-面碰撞,它同时可能包含(d)、(abc)两种形式的碰撞;对于(d),即边-边碰撞,它同时可能包含(abc)形式的碰撞。因此,为了避免冗余解的出现,必须按如下顺序求解:(f) → (e) → (d) → (abc),并且,对于顺序在前的碰撞求解时,必须把它所包含的其它形式的解作出标记,以便在对其它形式的碰撞求解时,不再考虑已作出标记的解。比如说,在求解(f)形式的碰撞检测时,发现三角形ABC与三角形A'B'C'之间存在面-面碰撞,但与此同时它包含边-面碰撞:AC-A'B'C'、AB-A'B'C'、B'C'-ABC、B'A'-ABC及边-边碰撞:AC-A'B'、AC-C'B'、AB-A'B'、AB-C'B'。因此,必须把这些边-面碰撞和边-边碰撞标记出来,以便在求解(e)形式和(d)形式的碰撞检测时,不再考虑它们,从而有效地避免了冗余解的出现。

总结以上分析,可以知道,对于TIN碰撞检测求解的关键在于:①如何独立求解(f)、(e)、(d)和(abc)形式下的碰撞检测;④如何避免冗余解的出现。对于④上面已经提供了较为基本的方法,下面着重考虑①。

2 四种形式的碰撞检测求解

2.1 (abc)形式下的碰撞检测

(abc)形式下的碰撞是指点与三角形之间的碰撞,记点为 Q_1 ,三角形为 $Q_2Q_3Q_4$,则问题转化为判断点 Q_1 与三角形 $Q_2Q_3Q_4$ 的运动轨迹是否相交。为此,需要求解如下方程:

$$q_1 = q_2 + (q_3 - q_2)u + (q_4 - q_2)v \tag{1}$$

其中:

$$q_i = Q_i + \int_{t_0}^t V_i(t) dt, i = 1, 2, 3, 4 \tag{2}$$

以上两式中,时间 t 的取值范围为 $[t_0, t_1]$, Q_i 表示点 Q_i 的起始空间坐标, $V_i(t)$ 表示 Q_i 的运动速度, u 和 v 是两个参变量。

可以看出,方程(1)的左边表示的是点 Q_1 的运动轨迹,其右边表示的是三角平面 $Q_2Q_3Q_4$ 的运动轨迹。如果方程(1)有解 (t', u', v') ,并且

$$\begin{cases} t_0 \leq t' \leq t_1 \\ u' \geq 0 \\ v' \geq 0 \\ u' + v' \leq 1 \end{cases} \tag{3}$$

则点 Q_1 与三角形 $Q_2Q_3Q_4$ 在运动过程中发生了碰撞,碰撞点 S 表示为:

$$S = Q_1 + \int_{t_0}^{t'} V_1(t) dt \tag{4}$$

假设物体 A 包含的顶点数为 m_A 、三角形数为 n_A , 物体 B 包含的顶点数为 m_B 、三角形数为 n_B , 且记以上求点与三角形之间的碰撞为一个基本算法单位, 则求物体 A 与物体 B 之间发生 (a b c) 形式碰撞的算法复杂度为 $O(m_{ANB} + m_{BNA})$ 。

2.2 (d) 形式下的碰撞检测

(d) 形式下的碰撞是指边与边之间的碰撞。实际上, 边-边之间发生的碰撞有可能是一个点, 也可能是一个线段, 在求解时必须注意此问题。记两条边分别为 E_1E_2 和 E_3E_4 , 则问题转化为判断边 E_1E_2 和 E_3E_4 , 在运动过程中是否相交。为此, 需要求解如下方程:

$$e_1 + u(e_2 - e_1) = e_3 + v(e_4 - e_3) \quad (5)$$

其中:

$$e_i = E_i + \int_0^t V_i(t) dt, \quad i = 1, 2, 3, 4 \quad (6)$$

以上两式中, 时间 t 的取值范围为 $[t_0, t_1]$, E_i 表示边的端点 E_i 的初始空间坐标, $V_i(t)$ 表示 E_i 的运动速度, u 和 v 是两个参变量。

可以看出, (5) 的左、右两边分别表示了边 E_1E_2 和 E_3E_4 的运动轨迹。如果方程(5)有唯一解 (t', u', v') , 并且:

$$\begin{cases} t_0 \leq t' \leq t_1 \\ 0 \leq u' \leq 1 \\ 0 \leq v' \leq 1 \end{cases} \quad (7)$$

则边 E_1E_2 和 E_3E_4 在运动过程中发生了碰撞。碰撞点 S 可表示为:

$$S = e_1(t') + u'e_2(t') - e_1(t') \quad (8)$$

如果方程(5)在某一时刻 $t' \in [t_0, t_1]$, 在区间 $0 \leq u \leq 1, 0 \leq v \leq 1$ 内有非唯一解, 则表明边 E_1E_2 和 E_3E_4 之间发生的是线段碰撞, 此时只需按相应方法求得线段两端点即可。

假设物体 A 包含的边数为 m_A , 物体 B 包含的边数为 m_B , 且记以上求边与边之间的碰撞为一个基本算法单位, 则求物体 A 与物体 B 之间发生(d)形式碰撞的算法复杂度为 $O(m_{AMB})$ 。

2.3 (e) 形式下的碰撞检测

(e) 形式下的碰撞是指边与面之间的碰撞, 显然, 如果发生碰撞, 则其结果必然是一条线段, 在求解以下方程时应注意此问题。记边为 F_1F_2 , 三角形为 $F_3F_4F_5$, 则问题转化为判断边 F_1F_2 和三角形 $F_3F_4F_5$ 在运动过程中是否以线段的形式相交。为此, 需要求解如下方程:

$$\begin{cases} f_1 + w(f_2 - f_1) = f_3 + (f_4 - f_3)u + (f_5 - f_3)v \\ (f_2 - f_1) \cdot ((f_4 - f_3) \times (f_5 - f_3)) = 0 \end{cases} \quad (9)$$

其中:

$$f_i = F_i + \int_0^t V_i(t) dt, \quad i = 1, \dots, 5 \quad (10)$$

以上两式中, 时间 t 的取值范围为 $[t_0, t_1]$, F_i 表示点 F_i 的初始空间坐标, u, v, w 是三个参变量。

可以看出, (9) 中第一个方程用于约束边与三角形的运动轨迹相交, 第二个方程用于约束边与三角形平行。二者共同约束可以保证相交形式为线段。如果方程组(9)有解 $(t', [u', u''], [v', v''], [w', w''])$, 并且

$$\begin{cases} 0 \leq w' \leq w'' \leq 1 \\ u'' \geq u' \geq 0 \\ v'' \geq v' \geq 0 \\ u'' + v'' \leq 1 \end{cases} \quad (11)$$

则边 F_1F_2 与三角形在运动过程中发生了碰撞, 碰撞结果为一条线段, 该线段的两个端点 S_1 和 S_2 分别为:

$$\begin{cases} S_1 = f_1(t') + w'(f_2(t') - f_1(t')) \\ S_2 = f_1(t') + w''(f_2(t') - f_1(t')) \end{cases} \quad (12)$$

设物体 A 包含的边数为 m_A , 三角形数 n_A , 物体 B 包含的边数为 m_B , 三角形数为 n_B , 且记以上求边与三角形之间的碰撞为一个基本算法单位, 则求物体 A 与物体 B 之间发生 (e) 形式碰撞的算法复杂度为 $O(m_A n_B + m_B n_A)$ 。

2.4 (f) 形式的碰撞检测

(f) 形式下的碰撞是指面与面之间的碰撞。显然, 如果发生碰撞, 则其结果必然是一个有界平面区域, 可能为三角形、四边形或五边形。记两个三角形分别为 $G_1 G_2 G_3$ 和 $G_4 G_5 G_6$, 则问题转化为判断三角形 $G_1 G_2 G_3$ 和 $G_4 G_5 G_6$ 是否在运动过程中以有界平面区域的形式相交。为此, 需要求解如下方程:

$$\begin{cases} g_1 + (g_2 - g_3)u + (g_3 - g_1)v = g_4 + (g_5 - g_4)\gamma + (g_6 - g_4)\alpha \\ \frac{A_1}{A_2} = \frac{B_1}{B_2} = \frac{C_1}{C_2} \end{cases} \quad (13)$$

其中:

$$g_i = G_i + \int_{t_0}^t V_i(t) dt, \quad i = 1, \dots, 6 \quad (14)$$

$$\{A_1, B_1, C_1\} = (g_2 - g_1) \times (g_3 - g_1) \quad (15)$$

$$\{A_2, B_2, C_2\} = (g_5 - g_4) \times (g_6 - g_4) \quad (16)$$

以上各式中, 时间 t 的取值范围为 $[t_0, t_1]$, G_i 表示点 G_i 的初始空间坐标, u, v, γ, α 为四个参变量。

可以看出, (13) 中的第一个方程用于约束两个三角形的运动轨迹相交, 第二个方程用于约束两个三角形平行。二者共同约束可以保证相交形式为有界平面区域。对于方程组 (13), 在求解时还需附加如下条件:

$$\begin{cases} u \geq 0 \\ v \geq 0 \\ u + v \leq 1 \\ \gamma \geq 0 \\ \alpha \geq 0 \\ \gamma + \alpha \leq 1 \end{cases} \quad (17)$$

联合求解 (13) 和 (17), 即可得到有关结论。如果有解, 则意味着发生了 (f) 形式的碰撞, 其解为有界平面区域, 此时, 只需确定其顶点即可。

设物体 A 包含的三角形数为 m_A , 物体 B 包含的三角形数为 m_B , 且记以上求三角形与三角形的碰撞为一个算法单位, 则求物体 A 与物体 B 发生 (f) 形式碰撞的算法复杂度为 $O(m_A m_B)$ 。

3 算法与实验

综合以上分析, 可得到如下 TIN 碰撞检测算法:

第一步: 检测 (f) 形式下的碰撞, 其解集为 Ω_f ; 并标记 Ω_f 中所包含的 (e)、(d) 和 (abc) 形式的解集, 分别记为 Ω_{f-e} 、 Ω_{f-d} 和 Ω_{f-abc} ;

第二步: 检测 (e) 形式下的碰撞, 其解集为 Ω_e ; 并标记 Ω_e 中所包含的 (d) 和 (abc) 形式的解集, 分别记为 Ω_{e-d} 和 Ω_{e-abc} ;

第三步: 检测 (d) 形式下的碰撞, 其解集为 Ω_d ; 并标记 Ω_d 中所包含的 (abc) 形式的解集, 分别记为 Ω_{d-abc} ;

第四步: 检测 (abc) 形式下的碰撞, 其解集为 Ω_{abc} ;

第五步: 求最终解集 Ψ :

$$\Psi = (\Omega_{abc} - \Omega_{d-abc} - \Omega_{e-abc} - \Omega_{f-abc}) \cup (\Omega_d - \Omega_{e-d} - \Omega_{f-d}) \cup (\Omega_e - \Omega_{f-e}) \cup \Omega_f$$

利用以上算法,我们在航线规划应用中得到了如图 2 所示一幅碰撞检测场景。其中飞行器和地形均以 TIN 表达,飞行器沿预先规划好的航线飞行时,在红线区域内发生碰地。这一结论将反馈给航线规划系统,以便对航线作出调整。

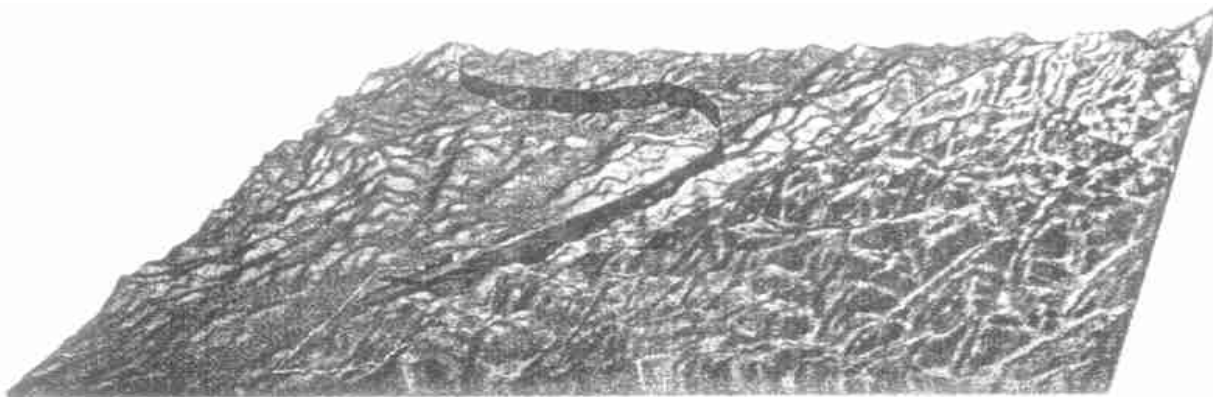


图 2 算法应用实例

Fig. 2 Application example of the algorithm

4 结论

本文讨论了如何解决 TIN 碰撞检测中的完整性控制问题,以及如何避免冗余解的问题,并给出了算法与实际应用结果。该方法的复杂度正比于被检测物体的点、边、面数目,故速度较慢,这也是该方法有待改进之处。需要指出,碰撞检测方法紧密依赖于物体的表达方式以及相对运动的复杂性,如何解决不基于同一类型表达的物体之间的碰撞检测是一个需要进一步研究的问题。

参考文献:

- [1] Von Herzen B, Barr A H. Geometric Collisions for Time-Dependent Parametric Surfaces [J]. Computer Graphics, 1990, 24 (4) .
- [2] Cameron S A, Culley R K. Determining the Minimum Translational Distance Between Two Convex Polyhedra [J]. IEEE International Conference on Robotics and Automation, 1986.
- [3] Culley R K, Kempf K G. A Collision Detection Algorithm Based on Velocity and Distance Bounds [J]. IEEE International Conference on Robotics and Automation, 1986.
- [4] Filip D, Magedson R. Surface Algorithms using Bounds on Derivatives [J]. Computer Aided Geometric Design, 1986, 3.
- [5] Uchiki T, Ohashi T. Collision Detection in Motion Simulation [J]. Computer & Graphics, 1983, 7 (3) .
- [6] Von Herzen B, Barr A H. Accurate Triangulations of Deformed, Intersecting Surfaces [J]. Computer Graphics 1987, 21 (4).
- [7] Duff T. Interval Arithmetic and Recursive Subdivision for Implicit Functions and Constructive Solid Geometry [J]. Computer Graphics 1992, 26 (2) .
- [8] Moore M, Wilhelms J. Collision Detection and Response for Computer Animation [J]. Computer Graphics 1988, 22 (4) .
- [9] Canny J. Collision Detection for Moving Polyhedra [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence 1986, 8 (2) .