# 空间三角形快速相交检测算法*

邹益胜，丁国富，何 邕，许明恒
(西南交通大学 机械工程学院，成都 610031)

**摘 要**：综述了典型的快速稳定的三角形相交检测算法的原理及实现方法，并根据算法原理将其分为标量判别型和矢量判别型算法。从计算量角度对各种算法的适用场合和性能进行了分析比较及验证，结果显示矢量判别型算法中的 Olivier Devillers & Philippe Guigue 算法整体性能最优，而标量判别型算法中的 Oren Tropp 算法最适合于三角形相交率较高的场合。

**关键词**：空间三角形；相交检测；标量判别；矢量判别
**中图分类号**：TP391 **文献标志码**：A **文章编号**：1001-3695(2008)10-2906-05

# Fast intersection algorithm between spatial triangles

ZOU Yi-sheng, DING Guo-fu, HE Yong, XU Ming-heng
*(College of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China)*

**Abstract:** This paper surveyed the principle and implementation of typically fast and robust triangle-triangle intersection algorithms, which was classified into scalar discrimination algorithm and vector discrimination algorithm according to their principle. Discussed the applicable occasions and performance of every algorithm by analyzing their calculation amount. The results of analysis and test show that the overall performance of the algorithm proposed by Olivier Devillers & Philippe Guigue, one of the vector discrimination algorithms, is optimal, and the algorithm proposed by Oren Tropp, one of the scalar discrimination algorithms, is most suitable for the occasions with high triangle-triangle intersection ration.

**Key words:** spatial triangle; intersection; scalar discrimination; vector discrimination

碰撞检测是机器人运动规划、计算机仿真、虚拟现实、游戏等领域不可回避的问题之一。随着三维几何模型越来越逼真、越来越复杂，虚拟环境的场景规模越来越大，这对碰撞检测的实时性提出了巨大挑战。提高碰撞检测的实时性一般从两方面处理：减少参与检测的图元数，如采用层次树等；减少图元间相交检测的时间，在基于图形的碰撞检测算法中，最终都要进行基本图元(如三角形等)的相交检测。三角形是最常用的基本图元，因此，对快速的三角形相交检测算法的研究有着重要意义。本文综述了国内外研究者提出的典型的空间三角形快速碰撞检测算法，以供相关领域的研究者交流和借鉴。

## 1 典型算法原理及实现方法

三角形和三角形相交测试方法很多，传统的有 brute-force(蛮力)方法。其基本原理为通过求解方程组检测一个三角形的每一条边与另一三角形的每一条边是否相交；如果出现相交，则两三角形相交。显然该方法检测效率较低[1]。为此研究者们采用不同的方法进行研究，提出了以下几种典型的快速三角形相交检测算法。

### 1.1 标量判别型算法

标量判别型算法是通过准确计算来获知两三角形相交关系的一类算法。这里介绍典型的 M ller、Held 和 Tropp 算法。其中 M ller 和 Held 算法从几何的角度对命题进行化简，最终分别化简成两线段的相交问题和线段与三角形的相交问题；Tropp 算法从代数的角度出发，充分利用求解过程中的中间值及矩阵操作的线性关系加速问题的解决。标量判别式可以直接求解出交点，但由于计算的累积误差，对其精度有一定的影响。

#### 1.1.1 M ller 算法[2~8]

设有两个三角形 $T_1$ 和 $T_2$，构造三角形所在的平面分别为 $\pi_1$、$\pi_2$，通过计算 $T_1$ 的顶点与平面 $\pi_2$ 的距离以及 $T_2$ 的顶点与平面 $\pi_1$ 的距离来判断三角形与平面的关系，从而快速排除部分不相交的三角形。如果两个三角形都分别与另外的三角形所在的平面相交，则其交线 $l_1$、$l_2$ 必定在两平面的交线 $L$ 上；若 $l_1$、$l_2$ 相交则两三角形相交。如果判断为共面，则通过适当地投影进行降维，转换为二维三角形相交的方法另行处理。二维三角形间的碰撞检测算法比较成熟，在此不再介绍。

具体描述如下：设有两个三角形 $T_1$ 和 $T_2$，顶点分别为 $V_{10}$、$V_{11}$、$V_{12}$和 $V_{20}$、$V_{21}$、$V_{22}$，三角形所在的平面分别为 $\pi_1$、$\pi_2$，其法向量分别为 $N_1$、$N_2$。

a) 计算平面 $\pi_2$ 的方程：

$$N_2 \times x + d_2 = 0 \tag{1}$$

其中：$x$ 为 $\pi_2$ 上任意一点；$N_2 = (V_{20} - V_{21}) \times (V_{22} - V_{21})$；$d_2 = -N_2 \times V_{21}$。

b) 将三角形 $T_1$ 的三个顶点分别代入平面方程 $\pi_2$，可得各

收稿日期：2007-10-15；修回日期：2007-12-27 基金项目：国家自然科学杰出青年基金资助项目(50525518)；国家“973”计划资助项目(2007CB714701)

作者简介：邹益胜(1980-)，男，浙江文成人，博士研究生，主要研究方向为 VP/VM、可视化(zysapple@sina.com)；丁国富(1972-)，男，四川乐至人，教授，博导，主要研究方向为 VP/VM/VR、先进制造技术、开放式数控技术、可视化；何邕(1983-)，男，四川达州人，博士研究生，主要研究方向为 VP/VM、可视化；许明恒(1947-)，男，陕西汉中人，教授，博导，主要研究方向为先进制造技术.

顶点到平面 $\pi_2$ 的距离为

$$d_{V_{1i}} = N_2 \times V_{1i} + d_2;\ i = 0, 1, 2 \tag{2}$$

( a) 如果 $d_{V_{1i}} = 0 (\ i = 0, 1, 2)$, 那么两三角形共面, 转而执行二维三角形和三角形相交测试。( b) 如果 $d_{V_{1i}} \neq 0 (\ i = 0, 1, 2)$ 且同号, 那么 $T_1$ 位于 $\pi_2$ 的同侧, 则不相交。这种早期的相交测试避免了大量三角形对的相交计算。

c) 计算平面 $\pi_1$ 的方程。

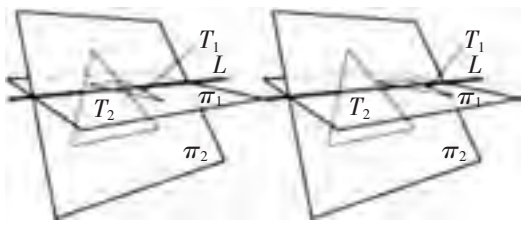
d) 将三角形 $T_2$ 的三个顶点分别代入平面方程 $\pi_1$, 排除顶点在 $\pi_1$ 同侧的三角形。

e) 经过前面的排除, 可以判定 $\pi_1$、$\pi_2$ 相交于一直线 $L$, 且 $L$ 必与两三角形相交。如果交线重叠, 则两三角形相交, 如图 1 ( a) 所示; 否则不相交, 如图 1( b) 所示。直线方程为

$$L = O + tD \tag{3}$$

其中: $D$ 为直线方向, $D = N_1 \times N_2$; $O$ 为 $L$ 上一点; $t$ 为 $L$ 上点的标量值。定义 $L$ 与两三角形交线的端点在 $L$ 上的标量值分别为 $t_1$、$t_2$、$t_3$、$t_4$, 通过三角形相交边与 $L$ 的投影关系( 图 2) 及相似三角形的性质, 求出 $t_1$、$t_2$、$t_3$、$t_4$。

f) 判断间隔 $t_1$、$t_2$ 和 $t_3$、$t_4$ 是否重叠, 如果是, 则两三角形相交。




(a) 两三角形相交　　(b) 两三角形不相交

图 1　三角形的位置关系

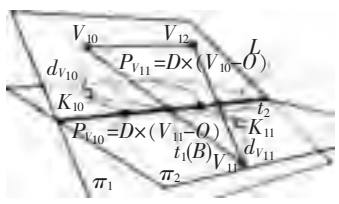



图 2　计算三角形与 $L$ 的交线

### 1.1.2　Held 算法[5, 6, 9]

Held 与 M ller 算法类似, 首先判断三角形 $T_2$ 的各个顶点是否位于平面 $\pi_1$ 的同侧进行早期排除。如果 $T_2$ 的三个顶点位于 $\pi_1$ 的两侧, 则可以构造线段 $s = T_2 \cap \pi_1$。测试线段 $s$ 是否与 $T_1$ 相交来判断三角形对的相交与否, 此相交测试可降维为二维的线段与三角形相交测试。

### 1.1.3　Tropp 算法[10]

Tropp 算法与前面提到的几种方法最大的不同之处在于: 前者致力于从几何角度解决问题, 后者的核心在于代数方法上。Tropp 采用蛮力算法的思路求解线性方程组, 指出方程组是强烈相关的, 并充分利用方程组中的共同单元以及矩阵操作中的线性关系来加速问题的解决。其算法的基本原理及实现过程如下:

设有 $A$、$B$ 两个空间三角形, 如果 $A$ 和 $B$ 相交, 则其中的一个三角形至少有一条边穿过另外一个三角形, 如图 3 所示。其中: $P$ 为 $p_1$、$p_2$ 的起始点; $Q_i$ 为 $q_i$ 的起始点, 则可得出

$$P + \alpha_1 \times p_1 + \alpha_2 \times p_2 = Q_i + \beta_i \times q_i;\ i = 1, 2, 3 \tag{4}$$

为保证交点在三角形内, 式( 4) 需满足约束条件: $0 \leq \beta_i \leq 1$, $\alpha_1$, $\alpha_2 \geq 0$ 且 $\alpha_1 + \alpha_2 \leq 1$。显然, 这需要求解六个方程。算法的主要贡献在于:

a) 重复利用方程组中的共同单元, 这样只需进行一次计算, 将其值保存供其他计算使用。

b) 利用矩阵计算的线性特性。算法的实施步骤如下( 返回结果为 "真" 表示两三角形相交, "假" 为不相交) :

( a) 部分求解方程组, 求得 $\beta$ 的值。

将式( 4) 写成矩阵形式, 并令 $r_i = Q_i - P$, 则

$$( p_1 \mid p_2 \mid q_i) ( \alpha_1 \alpha_2 - \beta_i)^{\mathrm{T}} = ( r_i) \tag{5}$$

定义 $A( v) = ( p_1 \mid p_2 \mid v)$, 则式( 5) 可写成

$$A( q_i)\ x_i = r_i;\ i = 1, 2, 3 \tag{6}$$

其中: $x_i = ( \alpha_1, \alpha_2, -\beta_i)^{\mathrm{T}}$。

采用克莱姆法则求解式( 6) 容易得到

$$\beta_i = - \mid A( r_i) \mid / \mid A( q_i) \mid \tag{7}$$

可以发现, $A( r_i)$、$A( q_i)$ 中含有相同的前两列 $p_1$、$p_2$, 因此 $A( r_i)$, $A( q_i)$ 可以按第三列展开, 则其子式只需计算一次。

由图 3 容易得出如下向量关系:

$$\begin{aligned} q_3 &= q_2 - q_1,\ Q_3 = Q_1 + q_1 \\ r_3 &= Q_3 - P = ( Q_1 + q_1) - P = r_1 + q_1 \end{aligned} \tag{8}$$

根据矩阵和行列式的计算性质, 有

$$\begin{aligned} \mid A( q_3) \mid &= \mid A( q_2 - q_1) \mid = \mid A( q_2) \mid - \mid A( q_1) \mid \\ \mid A( r_3) \mid &= \mid A( r_1 + q_1) \mid = \mid A( r_1) \mid + \mid A( q_1) \mid \end{aligned} \tag{9}$$

这样, 只需要有较少的计算量就能完成 $\beta_i$ 的计算。由此可见, 通过重利用方程组中的共同单元及矩阵的线性操作可以大大减少计算量。

( b) 如果 $\beta_i$ 的值不满足约束条件, 则返回测试结果为 "假", 测试结束。其目的是快速排除两三角形不相交的情况。如果所有 $\mid A( q_i) \mid$ 值为零, 则表示两三角形共面, 采用 M ller 算法中的共面计算方法。

( c) 构造三角形 $A$ 与 $B$ 所在的平面间的相交线段。如图 3 所示, 假设三角形 $A$ 与 $B$ 所在平面的交线为 $T + t$, 其中 $T$ 为三角形 $A$ 的一边与三角形所在平面的交点, 由前面的计算容易得到

$$\begin{aligned} T &= Q_1 + \beta_1 q_1 \\ t &= \beta_2 q_2 - \beta_1 q_1 \end{aligned} \tag{10}$$

( d) 如果线段与三角形 $B$ 相交, 则返回结果为 "真"; 否则返回为 "假"。线段与三角形 $B$ 的相交有两种情况: 与三角形的边相交; 线段在三角形内部。由此可以列出以下关系式:

$$\begin{aligned} P + \delta_1 p_1 &= T + \gamma_1 t \\ P + \delta_2 p_2 &= T + \gamma_2 t \\ ( P + p_1) + \delta_3 ( p_2 - p_1) &= T + \gamma_3 t \end{aligned} \tag{11}$$

其计算采用步骤中所述的方法。对于第一种相交情况, 当 $0 \leq \delta_i \leq 1$ 且 $0 \leq \gamma_i \leq 1$ 时, 表明线段与三角形相交; 对于第二种相交情况, 如果存在两个 $\delta \in [ 0, 1]$, 且其所对应的 $\gamma$ 值符号相反, 则表明线段包含在三角形中, 即两者相交。

## 1.2　矢量判别型算法

矢量判别型算法是通过一系列计算值的符号来判定两个三角形的位置关系, 继而判别其相交情况的一类算法。这里介绍典型的 Devillers & Guigue 算法和 Shen 算法。其中 Devillers & Guigue 算法通过三角形的各顶点构成的行列式正负的几何意义来判断三角形中点、线、面之间的相对位置关系; 而 Shen 算法则是采用一种基于分离平面的方法, 即将一个三角形的三条边扩展, 将其所在的平面分为七部分, 再通过计算带符号的线线距离来判别另一个三角形的边落在分离平面的哪个区域, 从而判别两个三角形的相交情况。总体上, 矢量判别型算法对计算的精度要求较低, 因此也具有比标量判别型算法更好的稳定性, 但这类算法对于交点的求解往往需要另外进行处理。

### 1.2.1　Devillers & Guigue[5, 6, 11, 12] 算法

Devillers & Guigue 算法( 简称 Devillers 算法) 通过三角形各顶点构成的行列式正负的几何意义来判断三角形中点、线、面之间的相对位置关系, 从而判断两三角形是否相交。其基本

原理如下:

给定四个空间点: $a=(a_x, a_y, a_z)$, $b=(b_x, b_y, b_z)$ $c=(c_x, c_y, c_z)$, $d=(d_x, d_y, d_z)$, 定义行列式:

$$[a,b,c,d]=\begin{vmatrix} a_x & a_y & a_z & 1\\ b_x & b_y & b_z & 1\\ c_x & c_y & c_z & 1\\ d_x & d_y & d_z & 1 \end{vmatrix}=\begin{vmatrix} a_x-d_x & a_y-d_y & a_z-d_z\\ b_x-d_x & b_y-d_y & b_z-d_z\\ c_x-d_x & c_y-d_y & c_z-d_z \end{vmatrix} \quad (12)$$

$[a,b,c,d]$ 采用右手螺旋法则定义了四个空间点的位置关系。$[a,b,c,d]>0$ 表示 $d$ 在 $a$、$b$、$c$ 按逆时针顺序所组成的三角形的正法线方向(即上方);$[a,b,c,d]<0$ 表示 $d$ 在 $\triangle abc$ 的下方;$[a,b,c,d]=0$ 表示四点共面。

设两个三角形 $T_1$ 和 $T_2$, 顶点分别为 $V_0^1$、$V_1^1$、$V_2^1$ 和 $V_0^2$、$V_1^2$、$V_2^2$, 三角形所在的平面分别为 $\pi_1$、$\pi_2$, 其法向量分别为 $N_1$、$N_2$。

算法先判别三角形和另一个三角形所在的平面的相互位置关系, 提前排除不相交的情况。通过计算 $[V_0^2, V_1^2, V_2^2, V_i^1]$ $(i=0,1,2)$ 来判断 $T_1$ 和 $\pi_2$ 的关系: 如果所有行列式的值都不为零且同号, 则 $T_1$ 和 $T_2$ 不相交; 否则 $T_1$ 和 $\pi_2$ 相交。相交又分为以下几种情况:

a) 如果所有行列式的值为零, 则 $T_1$ 和 $T_2$ 共面;

b) 如果其中一个行列式的值为零, 而其他两个行列式同号, 则只有一点在平面内, 测试顶点是否在 $T_2$ 内部, 是则相交, 否则不相交;

c) 否则, $T_1$ 的顶点位于平面 $\pi_2$ 的两侧(包含 $T_1$ 的一条边在平面 $\pi_2$ 中的情况)。

再按照类似的方法对 $T_2$ 和 $\pi_1$ 作进一步的测试。如果通过测试, 则每个三角形必有确定的一点位于另一个三角形所在平面的一侧, 而另外两点位于其另一侧。算法分别循环置换每个三角形的顶点, 以使 $V_0^1(V_0^2)$ 位于 $\pi_2(\pi_1)$ 的一侧, 另两点位于其另一侧; 同时对顶点 $V_1^2$、$V_2^2(V_1^1$、$V_2^1)$ 进行交换操作, 以确保 $V_0^1(V_0^2)$ 位于 $\pi_2(\pi_1)$ 的上方, 即正法线方向。经过以上的预排除和置换操作, $V_0^1$ 的邻边 $V_0^1V_1^1$、$V_0^1V_2^1$ 和 $V_0^2$ 的邻边 $V_0^2V_1^2$、$V_0^2V_2^2$ 与两平面的交线 $L$ 相交于固定形式的点上, 分别记为 $i, j, k, l$ $(i<j, k<l)$, 如图 4 所示。这些点在 $L$ 上形成的封闭区间为 $I_1=[i, j]$, $I_2=[k, l]$。至此, 两三角形的相交测试问题转换为封闭区间 $I_1$、$I_2$ 的重叠问题。若重叠则相交; 否则不相交。由于交点形式固定, 只需要满足条件 $k\leq j$ 且 $i\leq l$ 即表明区间重叠, 条件还可进一步缩减为判别式(13)是否成立, 示例如图 5 所示。

$$[V_0^1, V_1^1, V_0^2, V_1^2]\leq 0 \cap [V_0^1, V_2^1, V_2^2, V_0^2]\leq 0 \quad (13)$$

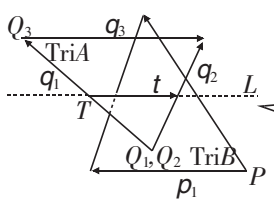



图 3 Tropp 算法原型描述

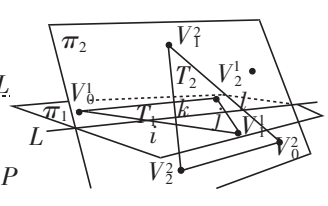



图 4 三角形和平面的位置示意图

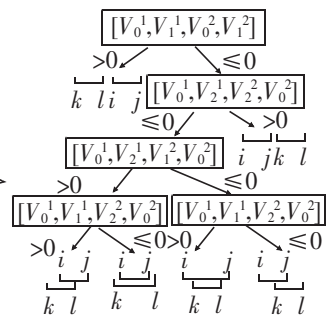



图 5 相交的符号表达

### 1.2.2 Shen[1] 算法

Shen 算法是对 M ller 算法的改进。算法改进体现在提出了一种基于分离平面来判别两三角形位置关系的新方法, 重点是针对两个交叉的三角形, 且三角形的顶点没有落在另一个三角形所在平面的情况。算法描述如下:

对于两条非平行直线 $L_1$ 和 $L_2$, 其方向向量分别为 $D_1$ 和 $D_2$, $N=D_1\times D_2$ 同时垂直于 $L_1$ 和 $L_2$。每条直线都包含在平面 $N\times U=c_i$。其中每条直线中的常量 $c_i=N\times P_i$, $P_i$ 为直线 $L_i$ 上的任一点。则有向的线线距离定义为 $d_{L_1L_2}=c_1-c_2$。算法采用左手法则判定符号, 如图 6 所示。$P_1P_1$ 平行于 $N$, 从 $L_1$ 到 $L_2$ 的有向距离就是从平面 $\pi$ 到 $P_1$ 的距离, 图中表示为正向。

设有两个三角形 $T_1$ 和 $T_2$, 顶点分别为 $V_0^1$、$V_1^1$、$V_2^1$ 和 $V_0^2$、$V_1^2$、$V_2^2$, 三角形所在的平面分别为 $\pi_1$、$\pi_2$, 其法向量分别为 $N_1$、$N_2$。如图 7 所示, 以 $T_1$ 为例, 直线 $L_i$ 分别为 $T_1$ 三条边的扩展, 将 $T_1$ 所在的平面 $\pi_1$ 划分成七个区域。三元组 $(S_{LL0}, S_{LL1}, S_{LL2})$ 用于判别直线 $L$ 与平面 $\pi_1$ 相交于哪个区域。其中 $S_{LLi}$ 为 $d_{LLi}$ $(i=1,2,3)$ 的符号。

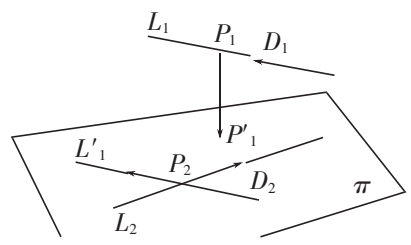



图 6 从 $L_1$ 到 $L_2$ 的有向距离

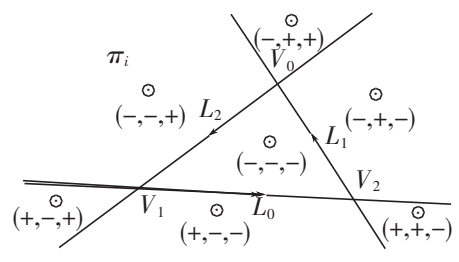



图 7 三角形的三边扩展成直线将其所在平面分成七个区域

与 Devillers & Guigue 算法类似, 该算法也要判别独点(即单独位于另一个三角形所在平面 $\pi$ 一侧的点)。如果三角形有一点位于另一个角形所在的平面 $\pi$ 内, 则位于平面两侧的任一点都可作为独点。由于独点的邻边都不与 $\pi$ 平等, 这两条邻边到 $\pi$ 内的任何一条线的线线距离都不为零。根据前面"一个三角形是否在另一个三角形一侧"的计算很容易获得独点。不失一般性, 记 $V_0^1$ 为三角形 $T_1$ 的独点, $V_0^2$ 为三角形 $T_2$ 的独点, $V_1^i$、$V_2^i(i=1,2)$ 的顺序影响 $N_i$ 的方向, 继而影响平面 $\pi_i$ 到点的距离的符号。不失一般性, 约定为: 调整 $V_1^i$、$V_2^i(i=1,2)$ 的顺序, 使得从 $V_0^1$ 到 $\pi_2$ 的距离以及 $V_0^2$ 到 $\pi_1$ 的距离为正。

两个边对 $\{L_{V_2^1V_0^1}, L_{V_2^2V_0^2}\}$ 和 $\{L_{V_0^1V_1^1}, L_{V_0^2V_1^2}\}$ 的线线距离决定了两个三角形是否相交。如果 $L_{V_0^2V_1^2}$ 到 $L_{V_0^1V_1^1}$ 的线线距离为负或者 $L_{V_2^2V_0^2}$ 到 $L_{V_2^1V_0^1}$ 的线线距离为正, 则两三角形不相交; 否则相交。

算法的实现可归纳为以下几步:

a) 通过计算排除一个三角形在另一个三角形同一侧的情况;

b) 获得独点, 并根据需要对其他两个点进行重排序;

c) 计算 $L_{V_0^2V_1^2}$ 到 $L_{V_0^1V_1^1}$ 的线线距离, 如果为负则不相交;

d) 计算 $L_{V_2^2V_0^2}$ 到 $L_{V_2^1V_0^1}$ 的线线距离, 如果为正则不相交;

e) 否则判断两三角形相交。

## 2 算法分析及测试

算法的时间消耗主要包括开辟内存和数学运算两部分, 涉及的运算包括四则运算、比较运算、绝对值运算和赋值运算。对于现行的主流计算机来说, 加减法、乘法与比较的计算时间可以认为是相同的, 除法计算时间大约是加法计算时间的 4 ~ 8 倍[11,12], 因此算法一般都避免除法以提高效率, 而赋值计算的时间相对而言可以忽略。Tropp 算法是在 M ller 和 Held 算法基础上的改进, 具有比它们更快的速度, 但它们都属于数值计算型, 其精度会受到机器误差和累积误差的影响, 这对算法的稳定性和准确性会有一定的影响。如果系统对算法的准确性要求较高, 可以采用 double 数值类型, 适当地牺牲一点速度

来保证精度。相对来说, Devillers 和 Shen 算法在这方面作了改进, 机器误差对其影响较小, 也降低了计算要求, 提高了算法的效率和稳定性。另外, Tropp 算法另辟蹊径从代数的角度对算法进行加速, 有一定的借鉴意义。

通过对各种算法及其实现代码进行研究, 分析如下( 由于没能获取 Held 算法源码, 此处主要针对文中提及的另四种算法进行分析和验证) :

a) 内存开辟分析。M ller 使用的变量数为 55; Devillers 使用的变量数为 18, Shen 使用的变量数为 38; Tropp 使用的变量数为 51。Devillers 算法最优。

b) 判定两三角形相交时各算法的计算量分析。如表 1 所示, Tropp 最优, 其总计算量为 103 /121, 其次为 Devillers。

表 1　判定两三角形相交时各算法的计算量比较

| 算法 | 加减 | 乘法 | 比较 | 绝对值 | 除法 | 赋值 | 计算量合计 |
|---|---|---|---|---|---|---|---|
| M ller( no div) | 57 | 63 /65 | 12 /24 | 3 | 0 | 69 | 135 /149 |
| Devillers | 76 | 52 | 10 /18 | 0 | 0 | 64 | 138 /146 |
| Shen | 70 | 56 /58 | 12 /82 | 0 | 0 | 62 | 138 /210 |
| Tropp( no div) | 42 /45 | 51 /57 | 10 /19 | 0 | 0 | 46 /52 | 103 /121 |

c) 排除不相交情况的计算量分析。四种算法都通过排除不相交的情况, 最后判定为相交, 这种方式可以有效提高检测的效率。各算法的排除次数和各次排除的计算量如表 2 所示。其中, Devillers 和 Shen 算法都采用四次排除, 对于相交率较低的三角形对集合的测试具有一定的优势; M ller 算法次之; 而 Tropp 算法只采用两次排除, 不利于计算过程中对于不相交情况的提前排除, 需要较大的计算代价。再对各次排除的计算量比较发现, 在第一次排除中, Tropp 算法的计算量相对较大; M ller、Devillers 和 Shen 算法的前两次排除的计算量相同; 第三、四次排除中的计算量, Devillers 优于 Shen 算法。

表 2　排除两三角形不相交情况的计算量比较

| 算法 | 第一次排除 | 第二次排除 | 第三次排除 | 第四次排除 | 确认相交 |
|---|---|---|---|---|---|
| M ller( no div) | 43 | 85/86 | 135/149 | / | 135/149 |
| Devillers | 43 | 85/86 | 114/122 | 138/146 | 138/146 |
| Shen | 43 | 85/86 | 114/186 | 138/210 | 138/210 |
| Tropp( no div) | 47/53 | 103/121 | / | / | 103/121 |

综上分析, 由于 Devillers 算法的内存开辟时间消耗少、较为适中的相交判别计算量, 再加上四次计算过程中不相交情况的排除, 整体上最优; Shen 算法次之, 其在相交率较低的场合可以发挥更大的作用; Tropp 算法则适用于相交率较高的三角形对集合。

为了测试各种算法的实际性能, 并验证分析结果, 按如下规则构造测试环境[1] :

a) 随机生成三维点 $V \in [0, 1]$ ;

b) 构成三角形的三个点不共线 ( 有的算法不能处理退化的三角形) ;

c) 一个三角形对由两个三角形 $T_1$ 和 $T_2$ 组成, 记为{ $T_1$, $T_2$} ;

d) 如果{ $T_1$, $T_2$} 相交( 采用 M ller 算法判别) , 写入相交集, 否则写入分离集;

e) 根据设定的相交率分别从相交集和分离集中取数据, 混合排序, 构成新的数据集。

测试条件为: Intel Core2 6300, 1. 86 GHz, 内存 1 GB, Windows XP 系统。取 11 种相交率, 对每种相交率取 5 个数据集样本, 每个样本由5 000个三角形对构成( 循环测试1 000次) , 5 个样本测试完成后取均值得到一种相交率的测试时间, 测试数据如表 3 所示。表 3 中的数据显示: 算法的时间开销随着相交率的降低而降低, 同时也验证了表 1、2 中对算法的理论分析。

表 3　测试不同相交率下各算法检测所用的平均时间 / ms

| 相交率 | M ller | Devillers | Shen | Tropp |
|---|---|---|---|---|
| 1.0 | 1 548.989 | 1 305.714 | 1 435.637 | 1 243.684 |
| 0.9 | 1 489.367 | 1 261.628 | 1 375.521 | 1 225.531 |
| 0.8 | 1 438.309 | 1 220.227 | 1 324.674 | 1 213.623 |
| 0.7 | 1 386.639 | 1 176.265 | 1 275.758 | 1 191.048 |
| 0.6 | 1 347.736 | 1 147.057 | 1 243.061 | 1 178.576 |
| 0.5 | 1 268.791 | 1 089.638 | 1 161.417 | 1 164.100 |
| 0.4 | 1 214.976 | 1 044.107 | 1 098.861 | 1 137.973 |
| 0.3 | 1 148.476 | 994.822 | 1 043.827 | 1 115.029 |
| 0.2 | 1 090.416 | 940.628 | 983.806 | 1 085.127 |
| 0.1 | 1 030.905 | 886.870 | 931.115 | 1 065.662 |
| 0.0 | 958.573 | 830.989 | 865.614 | 1 032.559 |

通过前面分析可知, 各算法对于三角形不相交的情况的排除能力不尽相同。以相交率为 0.5 的一组数据为例, 表 4 显示了各算法在各次不相交三角形排除计算中所排除的三角形数对算法时间消耗的影响。

表 4　相交率为 0.5 的各算法排除三角形数及检测所用的时间

| 算法 | 第一次排除 | 第二次排除 | 第三次排除 | 第四次排除 | 总排除数 | 时间消耗 / ms |
|---|---|---|---|---|---|---|
| M ller( no div) | 1 325 000 | 582 000 | 593 000 | / | 250 000 | 1 272.358 |
| Devillers | 1 324 000 | 583 000 | 284 000 | 309 000 | 250 000 | 1 087.816 |
| Shen | 1 324 000 | 583 000 | 298 000 | 295 000 | 250 000 | 1 165.851 |
| Tropp( no div) | 1 325 000 | 1 175 000 | / | / | 250 000 | 1 169.721 |

## 3　结束语

综上分析和验证可以得出结论: Devillers 算法总体上最优; Tropp 算法最适合于高相交率场合; Shen 算法比较适合中低相交率场合; M ller 算法虽然整体性能稍弱, 但它是其他算法的基础, 且应用广泛; 从相关材料分析结果看, Held 算法的整体性能还略逊于 M ller 算法, 但其与 M ller 算法类似, 也起着基础的作用, 其算法思想为其他算法所借鉴。以上算法的共同点都是在着力追求算法的快速性, 分别采用几何方法和代数方法实现, 同时从测试结果来看, 各算法也有较好的精度和稳定性。随着场景规模的大型化和复杂化, 检测速度是算法追求的主要指标之一。图形处理器( GPU) 的数据处理能力比 CPU 强很多, 因此许多研究者开始尝试着将 GPU 数据处理能力应用到通用计算领域, 并获得了成功, 特别是在流体计算领域, 展现了其强大的计算能力。这是一个新的领域, 有着巨大的潜力, 越来越多的研究者开始关注这一领域, 在三角形相交测试方面[13, 14] 以及相关的碰撞检测领域[15 ~20] 的研究工作也已开展。GPU 以比 CPU 更快的速度发展, 并进一步考虑了其在通用计算领域的应用。以 NVIDIA 公司为例, 其推出的新一代 GeForce 8 系列和 Quadro 系列 GPU 都采用了新的统一渲染构架, 取消了原来的顶点着色器和像素着色器, 取而代之的是统一的可编程流处理器, 增加了对 GPU 的利用率, GeForce 8 系列最多可拥有 128 个流处理器。为进一步挖掘 GPU 的通用计算性能, NVIDIA 公司推出了 CUDA 技术, 一种用于在 NVIDIA GPU 上进行计算的全新体系架构, 这是该领域内的首个 GPU 用的 C 编译器开发环境, 此举被誉为是揭开 GPU 计算革命的序幕。由此可见, 基于 GPU 的计算及其算法研究将逐渐成为研究的热点。由于 GPU 本身在处理上的并行性特点, 对更适合于在 GPU 上运行的三角形相交检测算法以及基于它的实时碰撞检测算法的研究将有着重要的科研和实践价值。

参考文献:

[ 1]　HEN Hao, HENG P A, TANG Z. A fast triangle-triangle overlap

test using signed distances[ J] . Journal of Graphics Tools, 2003, 8 ( 1) : 3-15.

[ 2] M LLER T. A fast triangle-triangle intersection test[ J] . Journal of Graphics Tools, 1997, 2( 2) : 25-30.

[ 3] 罗枫, 陈志杨, 张三元, 等. 基于 OBB树层次关系的相交体特征计算[ J] . 计算机应用研究, 2005, 22( 10) : 23-25, 29.

[ 4] 王志强, 洪嘉振, 杨辉. 碰撞检测问题研究综述[ J] . 软件学报, 1999, 10( 5) : 545-551.

[ 5] 门晓鹏, 吕晓峰, 马登武, 等. 虚拟场景中基本几何元素相交测试技术[ J] . 海军航空工程学院学报, 2006, 21( 3) : 379-382.

[ 6] 许强, 吕晓峰, 马登武. 三角形和三角形相交测试技术研究[ J] . 计算机仿真, 2006, 23( 8) : 76-78, 145.

[ 7] 高明向, 陈昆, 陈定方. 射线算法在碰撞检测中的应用[ J] . 湖北工学院学报, 2004, 19( 3) : 94-97.

[ 8] 陈学文, 刘静华, 丑武胜, 等. 快速计算虚拟物体之间精确接触位置的算法[ J] . 北京航空航天大学学报, 2005, 31( 7) : 799-804.

[ 9] HELD M. ERIT: a collection of efficient and reliable intersection tests [ J] . Journal of Graphics Tools, 1997, 2( 4) : 25-44.

[ 10] TROPP O, TAL A, SHIMSHONI I. A fast triangle to triangle intersection test for collision detection[ J] . Computer Animation and Virtual Worlds, 2006, 17( 5) : 527-535.

[ 11] DEVILLERS O, GUIGUE P. Faster triangle-triangle intersection tests, TR 4488[ R] . [ S. l. ] : INRIA, 2002.

[ 12] GUIGUE P, DEVILLERS O. Fast and robust triangle-triangle overlap test using orientation predicates[ J] . Journal of Graphics Tools, 2003, 8( 1) : 25-42.

[ 13] 范昭炜, 万华根, 高曙明. 基于流的实时碰撞检测算法[ J] . 软件学报, 2004, 15( 10) : 1505-1514.

[ 14] 于永彦, 夏宜蕃. 一种基于流计算的实时碰撞检测算法的研究与实现[ J] . 北京联合大学学报: 自然科学版, 2005, 19( 4) : 73-79.

[ 15] 柳有权, 刘学慧, 吴恩华. 基于 GPU 带有复杂边界的三维实时流体模拟[ J] . 软件学报, 2006, 17( 3) : 568-576.

[ 16] FERNANDO R. GPUGems: programming techniques, tips, and tricks for real-time graphics[ M] . Boston: Addison Wesley, 2004: 637-666.

[ 17] PHARR M. GPUGems: programming techniques for high-performance and general-purpose computation[ M] . Boston: Addison Wesley, 2005.

[ 18] GOVINDARAJU N K, LIN Ming C, MANOCHA D. Quick-CULLIDE: fast inter- and intra-object collision culling using graphics hardware[ C] // Proc of IEEE Virtual Reality Conference. Washington DC: IEEE Computer Society, 2005: 59-66, 319.

[ 19] HOFF KE ZAFERAKIS A, LIN Ming, *et al*. Fast 3D geometric proximity queries between rigid and deformable models using graphics hardware acceleration, TR02-004[ R] . Chapel Hill: University of North Carolina, 2002.

[ 20] PABST H F, SPRINGER J P, SCHOLLMEYER A. Ray casting of trimmed NURBS surfaces on the GPU[ C] // Proc of IEEE Symposium on Interactive Ray Tracing. Washington DC: IEEE Computer Society, 2006: 151-160.

[ 21] 谢凯, 杨杰. 一种基于虚拟手术的三维碰撞检测算法[ J] . 上海交通大学学报, 2007, 41( 6) : 866-869.

[ 22] 金汉均, 王晓荣, 王萌. 基于层次包围盒的碰撞检测算法的存储优化[ J] . 计算机工程与应用, 2007, 43( 16) : 61-63.

[ 23] KENSLER J, SHIRLEY P. Optimizing ray-triangle intersection via automated search[ C] // Proc of IEEE Symposium on Interactive Ray Tracing. Washington DC: IEEE Computer Society, 2006: 33-38.

[ 24] GOTTSCHALK S, LIN Ming C. Collision detection between geometric models: a survey[ C] // Proc of IMA Conference on Mathematics of Surfaces. 1998: 3-15.

[ 25] KLOSOWSKI J T, HELD M, MITCHELL J S B. Efficient collision detection using bounding volume hierarchies of k-DOPs[ J] . IEEE Trans on Visualization and Computer Graphics, 1998, 4( 1) : 21-36.

[ 26] SHEWCHUK J R. Adaptive precision floating-point arithmetic and fast robust geometric predicates[ J] . Discrete Computational Geometry, 1997, 18( 3) : 305-363.

( 上接第 2905 页)

## 5 结束语

人的视觉系统既是一个复杂的控制系统和高度完善的信息处理机, 又是一个高度非线性的动力学系统, 同时还是一个复杂的神经网络系统。视觉仿生研究涉及神经生理学、医学、控制科学、仿生学等多门学科, 是一个前沿交叉学科的研究对象, 需要用学科交叉的方法进行研究。本文从视觉的神经机制出发, 对视觉仿生研究的发展和前景进行了探索。目前, 机器视觉无论从哪方面看都远未达到人类视觉的水平, 人类对自身视觉在更高层次上的机理也还未完全探明, 因此, 视觉仿生研究的意义重大, 面临挑战, 任重而道远。

参考文献:

[ 1] 徐雨维, 高春圃. 工程生理学[ M] . 杭州: 浙江大学出版社, 1997: 282-303.

[ 2] 罗四维. 视觉感知系统信息处理理论[ M] . 北京: 电子工业出版社, 2006: 12-30.

[ 3] 刘伟, 袁修干. 人的视觉—眼动系统的研究[ J] . 人类工效学, 2000, 6( 4) : 41-44.

[ 4] ROBINSON D A, GORDON J L, GORDON S E. A model of the smooth pursuit eye movement system[ J] . Biological Cybernetics, 1986, 55( 1) : 43-57.

[ 5] LISBERGER S G, MORRIS E J, TYCHSEN L. Visual motion processing and sensory-motor integration for smooth pursuit eyemovements [ J] . Annual Review of Neuro Science, 1987, 10: 97-129.

[ 6] WAKAMATSU H, KUWANO M, SUDA H. Realization of physiological eyemovements by automatic selection of control laws using artificial neural network[ C] // Proc of the 3rd International Conference on Artificial Neural Networks. Brighton, UK: IEEE, 1993: 113-117.

[ 7] ZHANG X, WAKAMATSU H. An unified adaptive oculomotor control model[ J] . International Journal of Adaptive Control and Signal Processing, 2001, 15( 7) : 697-713.

[ 8] 張暁林, 若松秀俊. 眼眼球運動制御メカニズムの数学モデルと視軸制御システムの構築[ J] . 日本ロボット学会誌, 2002, 20( 1) : 89-97.

[ 9] HUBEL D H, WIESEL T N. Receptive fields of cells in striate cortex of very young, visually inexperienced kittens[ J] . Neurophysiol, 1963, 26: 994-1002.

[ 10] 張暁林, 川合拓郎. 眼固視微動を用いた立体エッジ画像生成法[ C] //第 11 回画像センシングシンポジウム講演論文集. 横滨: 日本情報処理学会, 2005: 303-306.

[ 11] ATTNEAVE F. Some informational aspects of visual perception[ J] . Psychological Review, 1954, 61( 3) : 183-193.

[ 12] KOSSLYN S M, FLYNN R A, AMSTEROAM J B, *et al*. Components of high-level vision: a cognitive neuroscience analysis and accounts of neurological syndromes[ J] . Cognition, 1990, 34( 2) : 203-277.