空间两三角形的相交问题

于海燕[1]， 何援军[2]

（1. 东华大学机械工程学院，上海 201620；2. 上海交通大学计算机系，上海 200240）

摘 要：重点考虑几何奇异问题，同时兼顾算法的效率。运用“分而治之”的方法从一维解得到二维解，进而得到三维解，将空间问题变为平面问题、线性问题。基于几何代数化依赖于坐标系，引入“计算坐标系”，简化了几何的表述与关系的类型，使“几何奇异”状态最后归结为平面上线段被三角形裁剪时的共点、共线问题，简单而明晰，从而可从理论上保证算法的鲁棒性，以平面处理的形式给出了两个空间三角形求交的完整解决方案。测试证明，几何关系、几何奇异类型与计算的简化足以弥补因“变换”而增加的额外开销。算法的速度也能达到实用要求——在笔记本电脑上也能达到每秒 100 万对三角形的相交计算。

关 键 词：几何计算；三角形相交；降维；几何奇异；计算坐标系
中图分类号：TP 391.72
文献标识码：A **文 章 编 号**：2095-302X (2013)04-0054-09

Testing the Intersection Status of Two Triangles

Yu Haiyan[1], He Yuanjun[2]

(1. College of Mechanical Engineering, Donghua University, Shanghai 201620, China;

2. Dept. of Computer Science & Engineering, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: This paper focuses on geometric singularity and algorithm speed. In our algorithm, a 3D problem is reduced to planes and is further divided to a linear problem. In order to simplify the representation of geometric problem, a computational coordinate system is constructed. Thus, the geometric singularity of two triangles is reduced to planar problems between a line and a triangle, which limits the singularity to co-points and co-lines. Consequently, the complex spatial geometric singularity can be represented by planar drawings simply and visually and the robustness of our algorithm can be certificated theoretically. This paper gives the whole algorithm and detailed schemes. Experiment tests show that the priority in simplifying geometric relations, geometric singularity and calculations is sufficient to make up for the cost of transformation. The speed is 1 million pairs of triangles per second on a notebook computer.

Key words: geometric computing; triangles intersection; dimension reduction; geometric singularity; computational coordinates

计算机构造的虚拟对象往往由成千上万个简单几何形状（如四边形面片，特别是三角形面片）组成。因此，碰撞检测算法归根结底是对这类简单几何形状的关系测试。高效的三角形与三

收稿日期：2013-04-29；**定稿日期**：2013-05-03
基金项目：国家自然基金资助项目（61073083）；中央高校基本科研业务费专项资金资助
作者简介：于海燕（1974-），女，黑龙江海林人，副教授，博士，主要研究方向为 CAD/CG。E-mail：yuhy@dhu.edu.cn

角形测试对于提高碰撞检测算法效率，增强虚拟场景的展现起至关重要的作用。

近年国内外对三角形与三角形求交测试的研究较为活跃，但通常将主要工作集中在如何加速算法的速度上。

Christer Ericson 对这种只偏重速度、忽视稳定性的研究方法表示担心：“They do reduce the number of floating-point operations compared to previous methods, but I don’t care.（这些算法与以前的算法相比是减少了浮点运算，但是我不感兴趣。）”他指出：“Frequently they test the primitives in some large number of random configurations and report the average time it took to do a test.（大都采用一些大规模随机产生的三角形间的相交计算的测试手段）”，“This kind of testing is very unlikely to hit the cases where the tests fail due to robustness issues.（这种测试很难检测到影响算法鲁棒性的状况）”[1]。

实际上，两个三角形边的平行、共面、相交于某一个点、某一条边的几何奇异现象在“随机”产生的测试例子中常常不会被碰到，因此，检测出现遗漏。而在应用中，对一个几何奇异的处理错误往往导致系统的崩溃。

本文重点考虑两个空间三角形位置的几何奇异情况，将几何奇异的现象归结为平面上少数几种简单情况，从理论上解决几何奇异对稳定性的影响。在此基础上追求算法的效率，达到应用要求。

1　研究现状

如图 1 所示，空间两三角形的关系通常为 4 种情况：

1） 相交，如图 1(a)所示，公共部分为一直线段；此时，两三角形所在平面相交。

2） 相叠，公共部分为一面区域，如图 1(b)所示，此时两三角形共面；

3） 相离，无公共部分，如图 1(c)所示，不论两个三角形所在平面是相交、平行或重叠，两三角形都可能相离。

4） 碰撞，两三角形处于接触的临界状态，包括共点和共边界两种状态，如图 1(d)所示；也可以看作是相交或相叠的奇异状态。

直接计算空间两三角形的公共部分（交点、交线段或共面区域）的过程一般比较复杂，计算复杂度较大。迅速排除两个三角形不相交是判断空间两三角形关系的常用策略，因为在实际应用中真正相关的三角形数目在整个模型中占的比例常较小。

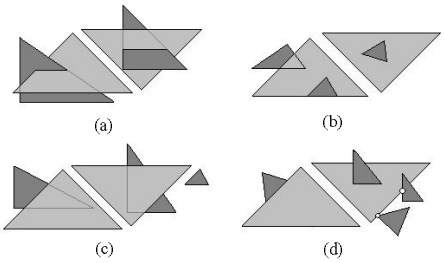
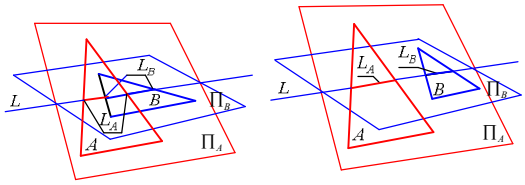


图 1　两个三角形在空间的分布情况

已经有多种关于三角形求交的算法，比较著名的有以下几种。

Möller 算法[2]原理，如图 2 所示，设两个三角形 A 和 B 所在的平面分别为 Π_A、Π_B。如果三角形 A、B 分别与平面 Π_B、Π_A 平面相交于 L_A、L_B，则 L_A、L_B 必定在两平面 Π_A、Π_B 的交线 L 上；若 L_A、L_B 有重叠部分，则两三角形相交，如图 2(a)所示，否则就相离，如图 2(b)所示。通过计算三角形 A 的顶点与平面 Π_B 的距离以及三角形 B 的顶点与平面 Π_A 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。这种方法需要建立两个三角形所在的平面。进行三角形分别与对方所在平面的 2 次“求交”及最后交线“重叠”部分的求取。



(a)　　(b)

图 2　两三角形空间求交的基本原理

Held[3]与 Möller 算法类似，首先判断三角形 B 的各个顶点是否位于平面 Π_A 的同侧进行早期排除。如果 B 的 3 个顶点位于 Π_A 的两侧，则构造线段 $s=B\cap\Pi_A$。通过测试线段 s 是否与 A 相交来判断三角形对的相交与否。这种方法将空间三角形的相交测试降维为二维的线段与三角形的相交测试。

Tropp 算法[4]与已有算法的重大不同是不用几何方法而采用代数方法，求解线性方程组，通过重利用方程组中的共同单元及矩阵的线性操作的策略减少计算量。

国内外学者在以上典型的三角形与三角形相交测试算法的基础上，提出了一些改进算法。张忠祥和王士同提出了基于 Ayellet 算法的改进算法[5]。首先快速排除掉两三角形不相交或共面的两种情况，然后分别计算一个三角形与另一个三角形所在平面相交线段，最后检测这两条线段是否有公共点。如果有公共点则三角形对相交，反之则不相交。邹益胜等对三角形求交的主要算法作了较为详细的描述并给出了测试结果[6]。

对于算法本身，很多改进算法都是在 Möller 和 Held 算法基础上进行的。在算法的稳定性方面，大多算法采用代数的方式描述，很难从理论上阐明对各种奇异状态的处理；在算法评估方面，重点在于算法速度的测试，包括不同相交率对算法性能影响的测试，以及计算量的统计等。而对于稳定性的测试方面缺少详细的分析与测试。

本文将 Möller 相交判断原理加以改进，并首次提出一种基于投影降维的方法，将各种空间状态的几何奇异降为平面问题；并建立基于视图的几何表述与求解机制，在理论上保证算法的稳定性。

2 算法原理与策略

本文将 Möller 求交判断的基本原理改造为以下两步（标识如图 2 所示），减少计算量。

1） 先求取 B 与 Π_A 的交线段得到 L_B；

2） 再求取 L_B 在 A 内的部分得到交线。

首先，建立一个合适的计算坐标系，使其中一个三角形在坐标平面上。再基于两面正投影原理，将空间两三角形的问题归结为平面上直线段与三角形（或直线段）的位置关系问题。最后，求交与重叠判定两项工作也在平面上解决。

实施中采用以下策略，使相交计算与测试更快速、稳定、精确：

1） 几何代数化依赖于坐标系，选取合适的标系，简化几何的表述与计算。

2） 尽可能利用简单的比较和计算，例如利用包围球、盒来排除无关的计算对象，尽量减少主计算。

3） 降低维数，如把三维降为二维、二维降为一维。

4） 能处理各种特殊情况，尽可能充分利用标准的算法，提高算法的鲁棒性。

5） 尽量避免那些开销量大的计算，如三角函数、平方根、除法等。

3 计算坐标系

一个合适的坐标系可以简化几何的表述与代数运算，本算法将建立一个计算坐标系。计算坐标系的构造方法如下：设有两三角形 $A(A_0, A_1, A_2)$与 $B(B_0, B_1, B_2)$，以 A 所在的平面作为 xy 坐标平面，该平面的法向作为 z 轴，并以 A 的一条边（例子中选取边向量 $\boldsymbol{A}_1\boldsymbol{A}_2$）作为 x 轴，如图 3 所示。（详见后面的“基本算法 1”）。

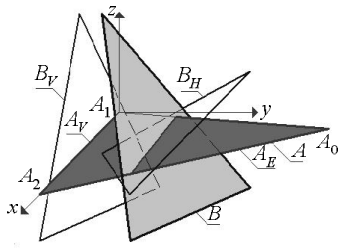


图 3 计算坐标系

记 xy 坐标平面为 H 面，zx 坐标平面为 V 面，A 与 B 在 V 面与 H 面上的投影分别为 A_V、A_H 与 B_V、B_H。

在这个计算坐标系下，三角形 A 与 B 的表述与关系具有下列 7 项性质：

1） A_H=A，即 A 所在平面就是 xy 坐标平面（V 面），且 A 只占 $y\geqslant 0$ 半平面。

2） A_V积聚成一条 x 轴上的线段，即 A_V=[X_L, X_R]。其中，X_L=min(0, x_{A2}, x_{A0})，X_R=max(x_{A1}, x_{A2}, x_{A0})。

3） B 在 V、H 面上的投影虽不定，但只可能是三角形或直线段 2 种。

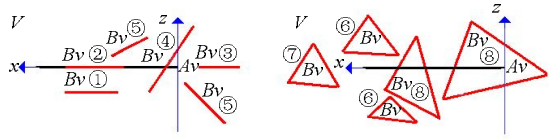
4） 如果 B_V的 3 个 z 坐标值同号，那么 B 与 A 相离（图 4 的①、⑤、⑥，有 0 值时是碰撞关系）。

5） 如果 B_V的 3 个 z 坐标值相同，B 与 A 所在平面平行；如果此时有 1 个 z 为 0，两平面就重叠，即在平行条件下只要有 1 个 B_V的$|z|>0$，B 与 A 就不重叠。

6） 如果 B_V 与 A_V 相交，那么交点坐标为(x_0,0,0)与(x_1,0,0)，交点可通过 B_V每条边向量端点的 z 坐标值求得（详见后面“基本算法 2”）。如果两个交点满足 max(x_0, x_1)$\leqslant X_L$ 或者 min(x_0, x_1)$\geqslant X_R$，那么 B 与 A 相离（图 4 的③、⑦，等于 0 时是碰撞关系）。

6） 在 V 面，A 与 B 的位置关系只有如下 2 种情况：

（1）水平直线段与直线段的关系，如图 4(a)所示；（2）水平直线段与三角形的关系，如图 4(b)所示。



(a) Av 与 Bv 均为直线段　　(b) Av 直线段而 Bv 三角形

图 4　计算坐标系下空间两三角形在 V 面投影的关系

图 4(a)中的 V 面上列出了 A 与 B 的线-线关系，其中①、③、⑤3 种情况显出 A_V 与 B_V 是相离的，因而空间的 A 与 B 也相离，只需对②、④两种情况作进一步的检测。同样，在图 4(b)中列出了 V 面上 A 与 B 的线−三角形关系，其中⑥、⑦两种情况也显出 A_V 与 B_V 是相离的，A 与 B 也就相离，只需对⑧作进一步的检测。因此，A 与 B 在①、⑤、⑥与③、⑦分布状态下的相离可以迅速予以排除，只需对②、④及⑧状态作进一步的检测，而这些检测均是在平面上进行的，后面会详细阐述对这些状态奇异性的解决策略。

引入计算坐标系不仅使得基于投影的两个空间三角形间的计算变得十分简单，更重要的是将空间几何奇异完全归结为平面问题，例如，能准确区分两个三角形“碰撞”与“相交”状态，使算法的稳定性和正确性大幅度提高。

4　总体算法设计与实现

4.1　两三角形计算的总体框架

在计算坐标系下，选取 $zx(V)$ 与 $xy(H)$ 投影面，求取两三角形顶点的两面正投影；根据两三角形的投影关系判断其空间位置关系，总体框架如图 5 所示。

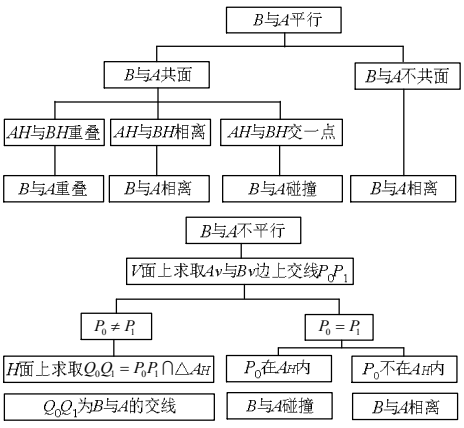


图 5　算法的总体框架

算法的主要步骤只需 3 步（标识如图 3 所示）。

1） 建立计算坐标系

（1）过 $A(A_0, A_1, A_2)$的 3 个顶点建立平 $xy(H)$ 平面，其法向作为 z 轴，并以 A 的一条边向量 A_1A_2 作为 x 轴，建立计算坐标系。（2）将 A 与 B 的顶点均变换到此坐标系下。

2） 在计算坐标系下求交计算：

（1）求取 B 与 A 所在平面 Π_A 的交线段 L_B。（2）求取 L_B 在 A 内的部分 Q_0Q_1。

Q_0Q_1 就是 B 与 A 在计算坐标系下的交线。

3） 变换回原坐标系

如果 A 与 B 有交线，将其端点变换回原坐标系，得到 A 与 B 在原坐标系下的交线段。

4.2　计算坐标系下两三角形交线的计算

根据计算坐标系下两三角形表述与关系性质，其交线必定在 xy 平面上，如图 6 所示。这可以简化求取 B 与 Π_A 的交线段 L_B 及求取 L_B 在 A 内的部分这两步工作，具体分析如下：

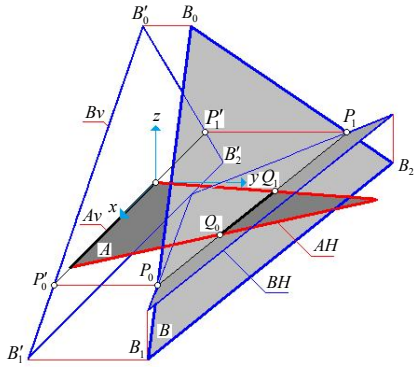


图 6　计算坐标系下两三角形交线的计算

Step 1　求取三角形 B 与 Π_A 的交线段 L_B。

通过 2 点的空间直线可用参数式表示，即：$x=x_0+t(x_1-x_0)$，$y=y_0+t(y_1-y_0)$，$z=z_0+t(z_1-z_0)$。从图 6 可知，B 与 Π_A 的交线 $L_B(P_0P_1)$在 H 面上(z=0)。由 B 与 B_V 所形成的梯形 $B'_0B'_1B_1B_0$ 可知，P'_0 在 $B'_0B'_1$ 上的参数 t'_0 与 P_0 在 B_0B_1 上的参数 t_0 相等；同理，P'_1 在 $B'_0B'_2$ 上的参数 t'_1 与 P_1 在 B_0B_2 上的参数 t_1 相等。而 P'_0 与 P'_1 均有 z=0 和 y=0，只需求出 x 就可。

因此，求取三角形 B 与 Π_A 的交线段 L_B 简化为线性问题：

先在 zx 坐标平面上求取 A_V 所在直线（在 x 轴上）与 B_V 的交线，即 L_B 在 zx 坐标平面上投影的两个端点 $P'_0(t_0$ 与 $x_0)$、$P'_1(t_1$ 与 $x_1)$，其中 t_i 为交

点在 B_V 边上参数，x_i 为交点的 x 坐标(i=0,1)，x_i 在 A_V 与 B_V 上是相同的，而 z_i 为 0；然后通过 t_i 在 B 的相应三维边上插出第三维坐标 y_i，最后得到 B 与 H 面(Π_4)的交线 P_0P_1。

Step 2 求取 L_B 在 A 内的部分。

L_B 本身就在 xy 平面上，因此，这项工作在 xy 平面上进行：

计算 $Q_0Q_1=P_0P_1\cap A_H$，如果 $Q_0Q_1=\Phi$，A 与 B 相离；如果 $Q_0=Q_1$，A 与 B 是碰撞关系（1 点接触）；如果 $Q_0Q_1\neq\Phi$，那么 Q_0Q_1 就是三角形 A 与 B 的交线。

下面是本文提出的计算坐标系下两个三角形求交算法。

计算坐标系下空间两三角形求交算法（标识见图 4）

输入：A，B 被检测的2个三角形

输出：*L Line3D 交线

返回值：0 三角形A与B 空间相离

1 三角形 A 与 B 空间相交

2 三角形 A 与 B 空间点碰撞点

3 三角形 A 与 B 空间线碰撞线

-1 三角形 A 与 B 共面 重叠

-2 三角形 A 与 B 共面 相离

-3 三角形 A 与 B 共面 点碰撞

-4 三角形 A 与 B 共面 线碰撞

-5 三角形 A 与 B 平行 但非共面相离

说明：本算法针对 A 与 B 均为空间三角形，需先排除 A 与 B 退化成空间直线段的情况

本算法在计算坐标系下进行，即已实施了以下操作：

(1)过三角形 A 的 3 个顶点建立平面作为 xy（H）面，其法向作为 z 轴，首边向量作为 x 轴，建立新坐标系;

(2)三角形 A 与 B 均已变换到此坐标系下； //both A_V and A_W are line segments

```
int IntTwoTriangles3D(TRiangle A, TRiangle B, Line3D
    *L)
{
//B∥H，both Bv and Bw are lines，①、②、③
//根据 B 的 3 点 z 坐标是否相同检测 B∥H
  if (B∥H ) { //B∥H
    if (B 与 A 共面) { //根据 B 的 z 坐标是否为 0 检测
  B 与 A 的共面性
      判断 AH 与 BH 两个三角形的关系；
      if (AH 与 BH 重叠) k=-1; //B 与 A 共面重叠
      if (AH 与 BH 相离) k=-2; //B 与 A 共面相离
      if (AH 与 BH 相交于一点) k=-3; //点碰撞
      if (AH 与 BH 相交于一线) k=-4; //线碰撞
    }
    else k=-5; //A 与 B 平行但非共面=相离
    return k;
  } //B∥H
//B 与 A 不平行，利用 V、H 平面计算
  在 V 面上求取 AV（直线）与 BV 边上交点 P'0(x0,0,0)
    与 P'1(x1,0,0)及对应的参数 t0,A;
  if (无交点) return 0; //③、⑤、⑥、⑦
  根据交点在 BV 边上参数 t0,A 在 B 上插出 y 三维坐标，
    得到 P0(x0,y0,0)与 P1(x1,y1,0); //⑧
  //x0、x1 的值在三角形 A 与 B 上是相同的，不必重算。
    z0、z1 在 H 面上，为 0。
  在 H 面上求取 Q0Q1=P0P1∩AH;
  if (Q0Q1=Φ) return 0; //三角形 A 与 B 不相交
  if (Q0=Q1) return 2; //三角形 A 与 B 共面点碰撞
  if (Q0Q1 与 AH 某边重叠) return 3; //线碰撞
  将 Q0、Q1 变换回原坐标系；//Q0Q1 即为交线。
  return 1;
  }
```

5 基本算法的实现

下面给出两空间三角形求交时用到的一些基本算法，除变换算法以外，其余算法均在平面上进行。

5.1 构筑基于三角形的计算坐标系

基本算法 1 设三角形由 P_1、P_2、P_3 三点给出，构筑基于这三点的计算坐标系 $x^*y^*z^*$，如图 7 所示：

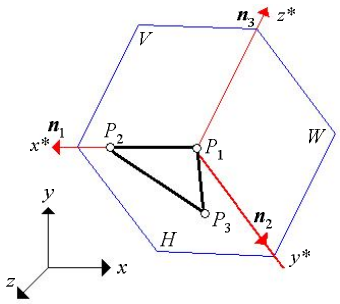


图 7 以三角形构筑计算坐标系

1） P_1P_2 单位向量作为 $\boldsymbol{n}_1(a_1\ b_1\ c_1)$

2） 通过 P_1、P_2、P_3 三点求取三角形的单位法向量为 $\boldsymbol{n}_3(a_3\ b_3\ c_3)$。

3） 作第三个单位向量 $\boldsymbol{n}_2(a_2\ b_2\ c_2)$，$\boldsymbol{n}_2=\boldsymbol{n}_3\times\boldsymbol{n}_1$

上述以 P_1 为原点而互相垂直的三个单位向

量 $\boldsymbol{n}_1$、$\boldsymbol{n}_2$ 与 $\boldsymbol{n}_3$ 构成计算坐标系 $x^*y^*z^*$，其中，$\boldsymbol{n}_1$、$\boldsymbol{n}_2$ 在三角形所在平面上，$\boldsymbol{n}_3$ 为平面的法向。新旧两坐标系间的坐标变换矩阵为：

$$\boldsymbol{T}_{xyz\_x^*y^*z^*}\begin{pmatrix} a_1 & a_2 & a_3 & 0 \\ b_1 & b_2 & b_3 & 0 \\ c_1 & c_2 & c_3 & 0 \\ d_1 & d_2 & d_3 & 1 \end{pmatrix}$$

和

$$\boldsymbol{T}_{x^*y^*z^*\_xyz}=\boldsymbol{T}^{-1}_{xyz\_x^*y^*z^*}=\begin{pmatrix} a_1 & b_1 & c_1 & 0 \\ a_2 & b_2 & c_2 & 0 \\ a_3 & b_3 & c_3 & 0 \\ D_1 & D_2 & D_3 & 1 \end{pmatrix}$$

其中，参数 d_1，d_2，d_3 与 D_1、D_2、D_3 根据新坐标系原点通过点 P_1 求得：

$$\begin{cases} d_1=-(a_1x_1+b_1y+c_1z_1) \\ d_2=-(a_2x_1+b_2y_1+c_2z_1) \\ d_3=-(a_3x_1+b_3y_1+c_3z_1) \end{cases}$$

$$\begin{cases} D_1=-(a_1d_1+a_2d_2+a_3d_3) \\ D_2=-(b_1d_1+b_2d_2+b_3d_3) \\ D_3=-(c_1d_1+c_2d_2+c_3d_3) \end{cases}$$

变换公式为：

$(X^*\ Y^*\ Z^*\ H^*)=(x\ y\ z\ 1)\boldsymbol{T}_{xyz_x^*y^*z^*}$及$(X\ Y\ Z\ H)=(x^*\ y^*\ z^*\ 1)\boldsymbol{T}_{x^*y^*z^*_xyz}$。

5.2　V 面上交点的计算

在计算坐标系下，A_V 是 x 轴上的一段线段$[X_L, X_R]$, B_V 可能是直线段或三角形。

1）　交点参数

基本算法 2　设 B_V 某个边的两端点为 $P_0(x_0, 0, z_0)$与 $P_1(x_1, 0, z_1)$，由于 A_V 在 x 轴上，若边 P_0P_1 与 A_V 有交点，则 P_0 与 P_1 必定一个在 x 轴上方，另一个在 x 轴下方，而交点的 z=0，如图 8 所示。有：

如果 z_0 与 z_1 的符号相同，边 P_0P_1 与 x 轴不相交；

（1）　如果 z_0 与 z_1 的符号相反，边与 x 轴相交，交点为 $Pt=P_0+t(P_1-P_0)$，式中 $t=|z_0|/(|z_0|+|z_1|)$。

（2）　如果 z_0 与 z_1 中有一个或两个为 0，则边与与 x 轴相交，交点为 z 坐标为零的端点。

（3）　A_H 上的三维交点坐标$(x, y, 0)$: $x=x_0+t(x_1-x_0)$, $y=y_0+t(y_1-y_0)$。

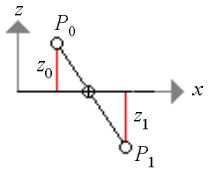


图 8　V 面上交点的计算

2）　A_V 与 B_V 的求交计算

基本算法 3　如果 B_V 是三角形，如图 9(a)所示，根据 z=0 的原则依次求取 B_V 的 3 条边与 x 轴的交点，一般情况下，这样的交点有 2 个。设交点的参数为 $t_i(i=0,1)$，根据 t_i 插出它的 x_i 坐标，得到交点的参数$(t_i,\ x_i,\ n_i)$，这里，n_i 是该交点在 B_V 上的边号，因为三角形有 3 条边，n_i 可能的值为 0、1、2。参数(t_i, n_i)将用于在 H 面上计算该交点的 y 值。

如果 B_V 是直线段，实际上是 B 的 3 边在 V 面的积聚投影，如图 9(b)所示，仍应作为 3 条边分别计算。虽然两交点的(t_i, x_i)参数是一样的，但是所在边号 n_i 不一样，用$(t_i,\ n_i)$去求交点在空间的 y 时就会得到 2 个不同的值。

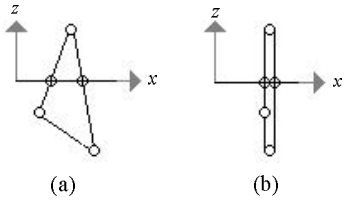


图 9　V 面上求交的统一

这样，不管 B_V 是三角形还是直线段，在本算法中在 V 面上求取交点的计算方式是统一的。

3）　排除 A_V 与 B_V 相离

基本算法 4　为加速算法，可用拒绝判定快速排除 A_V 与 B_V 的相离。在计算坐标系下，排除计算也十分简单。

在 z 方向，如果 B_V 的 3 个 z 坐标全为正或全为负，表明 B_V 完全在 A_V 的上方或下方，两者就无重叠关系（图 10①）；

在 x 轴上，如果 2 个交点均位于 A_V 右端点的右边（$\max(x_0,\ x_1)>X_R$）或均位于 A_V 左端点的左边（$\min(x_0, x_1)<X_L$），那么 A_V 与 B_V 就相离（图 10③）。

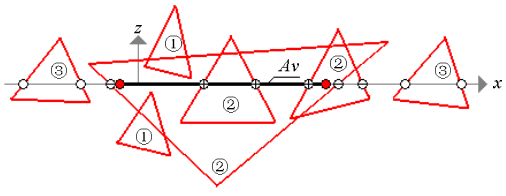


图 10　水平线段与三角形的位置分布图

5.3　H 面上直线段与三角形的交集的求取

基本算法 5　求取 L_B 在 A 内的部分就是在 H 面上求取 L_B 与 A_H 的交集。线段与三角形的交集可借用任何一种凸多边形裁剪算法，这里借用

Cyrus-Beck 参数化裁剪算法[7]，将 Cyrus-Beck 对凸多边形参数化裁剪算法改造成 Cyrus-Beck 三角形参数化裁剪算法，并增加了被裁剪线的端点或其本身在三角形边界上的奇异情况的处理（参见 7.2 H 面上线段与三角形的交集的几何奇异）。

6 几何奇异处理

6.1 *V* 面求交时的几何奇异

在 *V* 面上是直线段(A_V)与三角形(B_V)的三条边的求交，交点数可能是 1、2、3 个。

1） 交点数为 1

表示交点在 A_V 的左端点（x=0，图 11④）或右端点（$x=X_R$，图 11⑧），B_V 位于 A_V 的左侧或右侧。进一步对交点作 A_H 的包容性测试，如果交点在 A_H 的内部或边界上，则 *A* 与 *B* 为“碰撞”关系，否则为相离关系；

2） 交点数为 2

此时可能是正常交点（图 11⑤）、共边（图 11⑥）及共点（图 11⑦），前两者（⑤、⑥）进一步进行 A_H 的裁剪测试，所得结果即为 *A* 与 *B* 的交线；后者（⑦）的交点数为 1 的处理方法相同；

3） 交点数为 3

此时必有一个交点通过 B_V 的一个顶点（图 11①、②、③），解决方法为删除通过顶点的一个交点。因为参数 t_i 是基于 B_V 的边向量的，因此，通过 B_V 的一个顶点的交点的参数 t_i 必有一个为 0，另一个为 1，如果不计 t_i=0 的交点（也可不计 t_i=1 的交点），交点就减为正常的 2 个。设 3 个交点的排列为 0、1、2，处理方法是：

```
if (t0=0)  t0=t2;    n0=n2;   //图 11②
else if (t1=0)   t1=t2;    n1=n2;   //图 11①
n--;//n 为交点计数，原来为 3，现在为 2
```

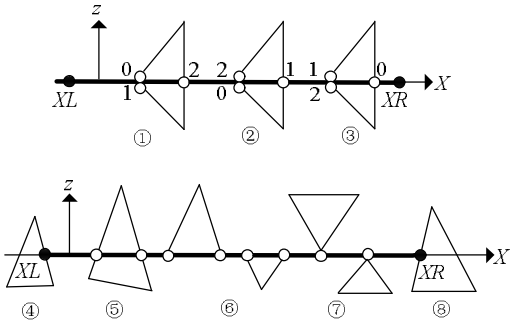


图 11 水平线段与三角形位置分布的奇异情况

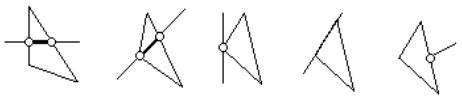
6.2 *H* 面上线段与三角形的交集的几何奇异

H 面上线段与三角形的交集的求取系借用 Cyrus-Beck 参数化裁剪算法，分为以下 5 种情况，奇异情况比较好处理，如图(12)所示。

1）正常的裁剪，如图 12(a)所示；

2）直线穿越三角形的一个顶点，使得线段与三角形有 3 个交点，破坏了直线进出三角形的奇偶性，如图 12(b)所示；

3）是直线通过顶点，但不“穿越”三角形，如图 12(c)所示。



(a) (b) (c) (d) (e)

图 12 三角形裁剪的奇异情况

4） 直线与三角形的某一边重合，在平面是共线，反映到空间是两三角形“共线”碰撞，如图 12(d)所示。

5） 直线段的一个端点在三角形的边界上，反映到空间是两三角形“共点”碰撞，如图(e)所示。

上述(1)、(2)空间两三角形有交线，(3)、(4)、(5)“碰撞”。

最后，当 *H* 面上的线段退化为一点时，需要检测该点是否在 A_H 的内部，如果是，空间两三角形“共点”碰撞，否则，两三角形空间相离。

6.3 两三角形共面时的几何奇异

当两个三角形共面时，可能出现相离、碰撞（共点、共线）、重叠（包括内含）等关系。

不能简单地用“判断 *B*(*A*)的顶点是否在 *A*(*B*)的内部”，因为，即使两个三角形的顶点均在对方之外，两者仍可能是相交关系。

本文采用“如果 *B* 各边与 *A* 有交集，那就有重叠（反之也然）”的策略判定平面上两三角形的重叠关系。将两个三角形的重叠关系的问题降为“一线段与一三角形的重叠关系”的判定。最后也归结到前节“线段被三角形裁剪”算法中的一些奇异情况。

1) B 中任一边被 A 三角形裁剪后如果有显示部分，两三角形为重叠关系；

2) B 中三边被 A 三角形裁剪后均无显示部分，两三角形为相离关系；

3) B 中任一边被 A 三角形裁剪后只显示 1 点，两三角形为点重叠关系；

4) B 中任一边与 A 三角形的某边界重合，两三角形为边重叠关系。

需要注意的是，4)比 3)的优先级高。

因为平面上 2 个三角形重叠时可能多达 6 个交点，最多可产生六边形，因此，“准确重叠部分的求取”并非一个简单的问题。考虑到大多数空间三角形关系处理的目标是判定他们的相离、碰撞和重叠关系等，只有在“曲面求交”等一类应用中才追究“准确重叠部分的求取”，因此，本算法不展开对它进一步讨论。而且这是一个平面问题，也可以参考文献[8]，用“两个平面多边形的布尔运算”方法解决。

7　测试与分析

7.1　奇异测试

两个空间三角形的奇异情况可在共面、相交情况下出现。共面情况下可能有共点、共线、内含等；相交情况下可能有共点、共线等。

本算法根据这些情况构造了相应的奇异测试例子。测试样本根据两三角形所位平面的关系分为平面平行、重合和相交 3 大类，如图 13 所示。

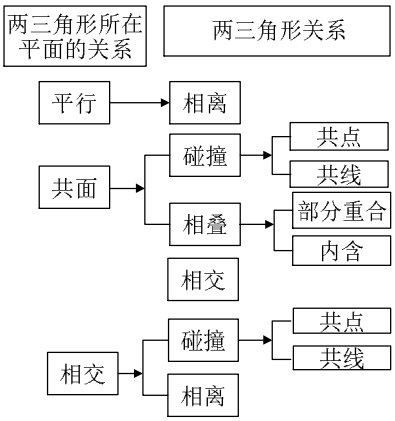


图 13　测试样本的分类

测试样本中，先将一个三角形 *A* 固定，变换另一三角形 *B* 的位置实现各种空间位置关系。对一个 *A*，设计了包括图 1、图 4、图 10、图 11 以及图 12 中描述位置关系在内的 23 种与 *A* 位置呈现含奇异状态的三角形 *B*。此外，还设计了其它 30 种空间位置关系的三角形，包含了各种直接分离、*H* 面检测等情况（图 14）。测试结果是本算法均能正确处理。

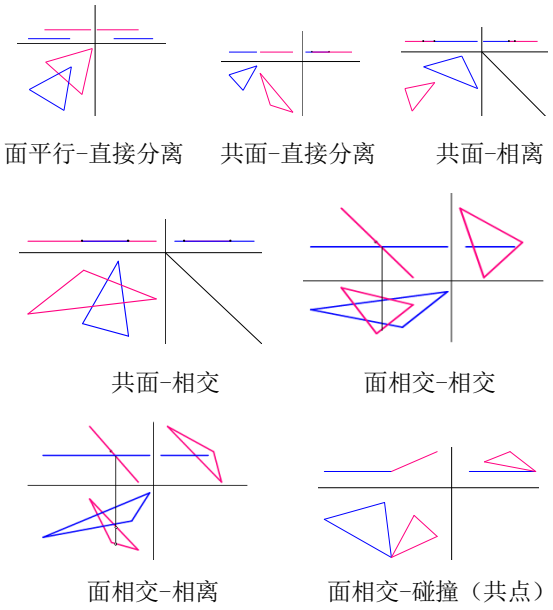


图 14　三视图表达的测试样本举例

7.2　速度测试

用 40 对三角形，重复 100 万次相交计算，分别在笔记本电脑与台式电脑两类机器上进行测试，测试结果如表 1 所示。

表 1　本算法对 40×1000000 对空间三角形进行相交测试的时间表(sec)

测试类型	测试时间	变换/总时间	测试机器
单变换	7		笔记本电脑
单求交	38	7/47=	IBM Thinkpad T60p,
变换+求交	47	0.148936	Windonws XP
单变换	5		台式电脑
单求交	28	5/31=	Pentium(R)
变换+求交	31	0.162903	Dual-Core E6500 2.93GHz,2.0GB 内存 Windows XP

计算坐标系下的测试，在笔记本电脑上，0.95 秒(38/40)可处理 100 万对三角形的相交计算，在台式电脑上，0.7 秒(28/40)可处理 100 万对三角形的相交计算。

在一般坐标系下测试，在笔记本电脑上，1.2 秒(47/40)可处理 100 万对三角形的相交计算，在台式电脑上，0.775 秒(31/40)可处理 100 万对三角形的相交计算。

7.3　算法分析

在粗略排除阶段，可以采用直接三维空间判断和投影判断相结合的方法。例如，图 4 中的①、⑤、⑥状态也可直接在三维空间进行判断：对三

角形 A 建立一个平面，$ax+by+cz+d=0$，将三角形 B 的 3 个顶点代入 A 的这个平面方程，得到 3 个数：$D_{bi}=ax_{bi}+by_{bi}+cz_{bi}+d(i=1, 2, 3)$，如果 D_{bi} 全部同号，则三角形 B 与 A 分离。而对于排除图 4 中的③、⑦状态用投影办法较好。

本算法需要进行“计算坐标系的建立”、“A 与 B 三个顶点的齐次变换”以及“交点变换回原坐标系”等变换计算。因此，我们专门考察了变换在本算法中所占时间的份额——最高为 16%左右。

实际上，在大规模的三角形求交计算中不需要对每对求交的三角形进行计算坐标系的建立与变换计算，例如，对每一个 A 建立计算坐标系，变换 1 次 A，而只对与 A 相关的（一系列）B 进行变换即可。

经计算坐标系下的投影降维，A 与 B 的关系可以直接用图形在平面上“画”出来，将空间几何关系降解为视图中直线段与三角形（或直线段）的关系，能严格区分两个三角形的“碰撞”、“相交”“相离”与“重叠”的关系，便于发现处理各种几何奇异情况；同时这种变换使算法变得更清晰，计算更简单，例如“两线段求交的参数求取只需 1 次加法，1 次除法”。因此，综合分析，本算法的变换开销利大于弊。

在计算策略上，充分利用直线的参数表述，采用分而治之的办法将空间问题变为平面问题、线性问题，使得几何表述更简单、几何间的关系更清晰，可以使用更多的“通用”的算法，例如，凸多边形裁剪算法，使算法更为规范。

8 总 结

本文给出了一种基于投影降维的空间两三角形求交检测方法的详细方案和算法。这是一种几何化的全新思路，能从理论上严格保各种奇异情况的发现与正确处理。也为其它基本几何元求交、碰撞检测等提供了一种新的解题思路。

参 考 文 献

[1] Christer Ericson. Triangle-triangle tests, plus the art of benchmarking [EB/OL]. http://realtimecollisiondetection.net/blog/?p=29.

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

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

[4] Tropp O Tala, Shimshoni I. A fast triangle to triangle intersection test for collision etection [J]. Computer Animation and Virtual Worlds, 2006, 17(5): 527-535.

[5] 张忠祥, 王士同. 三角形对的快速相交测试[J]. 计算机工程与设计, 2010, 3(4): 869-875.

[6] 邹益胜, 丁国富, 何 邕, 等. 空间三角形快速相检测算法[J]. 计算机应用研究, 2008, 25(10): 2907-2909.

[7] CYRUS M, BECK J. Generalized two-and three-dimensional clip-ping [J]. Computers and Graphics, 1978, 3(1): 23-28.

[8] 何援军. 计算机图形学(第 2 版)[M]. 北京: 机械工业出版社, 2009.