

# 利用点面提取的任意多面体快速靠接\*

胡华<sup>1,2</sup> 何志均<sup>1</sup> 高济<sup>1</sup> 张行功<sup>1</sup>

<sup>1</sup>浙江大学 CAD & CG 国家重点实验室 杭州 310027)

<sup>2</sup>(杭州商学院计算机系 杭州 310035)

**摘要** 文章主要讨论沿直线运动的多面体靠接问题,根据多面体的几何特性,通过一系列的筛选和变换将多面体的靠接问题转化成为计算少量点与平面多边形的距离问题,从而大大地提高了任意空间多面体靠接的计算速度.该算法可广泛地用于以多面体为数学模型的计算机图形学、机器人、多媒体和 CAD/CAM 等众多领域的系统中.

**关键词** 多面体,靠接,角度位移,点面集.

**中图法分类号** TP391

对物体的紧密靠接问题,以往的研究主要分为两类:第1类将物体简化成平面多边形进行<sup>[1~5]</sup>,平面问题的算法在二维动画和计算机辅助排料等方面得到了较好的应用;而以多面体为模型的第2类方法则应用在许多不能忽略物体三维特征的情况中(如不规则物体在复杂环境中降落或机器人行走在地形复杂的区域中等).J. W. Boyse 提出分别估计碰撞多面体的每一边与相对多面体的距离来解决凸多面体的内外靠接问题<sup>[6]</sup>;T. Uchikh 和 M. Moore 等人使用将物体球体化的方法来解决计算机中的模拟问题<sup>[7,8]</sup>;T. L. Perez 等人使用膨胀障碍物,同时收缩运动物体的方法来规划运动路径<sup>[9]</sup>;H. M. Yang 等人提出,采用层次二叉树模型和解四次方程的方法来模拟机械制造系统中的物体碰撞问题<sup>[10]</sup>;还有一些研究者采用分割时间序列进行重叠检测的方法,来检测模拟具有复杂曲面的空间物体碰撞问题.<sup>[11,12]</sup>例如,文献[6,10]的方法不仅计算量大,而且还要用其他算法将凹多面体分割为凸多面体,计算复杂度过高;文献[7~9]的方法不适合于任意形状多面体的精确内靠接和外靠接;文献[11,12]的方法依赖于时间区间的分割精度,且不适合于任意形状多面体的精确内靠接问题.本文根据多面体的空间几何特性,提出了一种基于提取多面体正向面和逆向面,并按角度计算空间点到平面多边形的位移方法,来实现任意连通多面体的快速内外靠接.该方法可以采用一系列的筛选变换来提高运算速度.

## 1 预备知识

多面体可用体、面、线和点的数据结构表示.不妨约定:多面体的每个面的外法线方向为由物体内部指向外部;各平面上包围多面体的多边形顶点排列次序(边的环绕方向)与面的外法线满足右手法则.由此可定义,如图1所示,其中{A, D, C, B}, {A, B, F, E}, {B, C, G, F}, {C, D, H, G}和{A, E, H, D}为正反面, {E, F, G, H}为逆向面.

**定义1.** 多面体中外法线方向与Z轴正方向夹角落在 $[0, \pi/2]$ 内的面为该多面体的正反面.

**定义2.** 多面体中外法线方向与Z轴正方向夹角落在 $(\pi/2, \pi]$ 内的面为该多面体的逆向面.

判断外法线方向与Z轴正方向的关系,可以通过计算多边形在OXY平面上的投影的有向面积来实现.设多边形的顶点序列为 $V_1, V_2, \dots, V_N, V_1$ 的坐标为 $(x_i, y_i, z_i)$ ,则有向面积<sup>[13]</sup>

$$SA = 1/2 \left( \sum_{i=2}^{N-1} (\vec{V}_i \times \vec{V}_{i+1}) \right)_{xy} = 1/2 \left( \sum_{i=2}^{N-1} ((x_i - x_1)(y_{i+1} - y_1) - (x_{i+1} - x_1)(y_i - y_1)) \right). \quad (1)$$

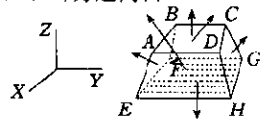


图1

\* 作者胡华,1964年生,博士生,副教授,主要研究领域为计算机图形,多媒体,CIMS.何志均,1923年生,教授,博士导师,主要研究领域为人工智能,软件工程,CIMS.高济,1946年生,教授,博士导师,主要研究领域为人工智能,CIMS.张行功,1973年生,硕士生,主要研究领域为计算机网络与分布处理,CAD/CG.

本文通讯联系人:胡华,杭州310012,杭州市下宁巷1号41524

本文1996-12-27收到原稿,1997-07-02收到修改稿

$SA > 0$  时, 该多边形所在面为逆向面;  $SA < 0$  时, 该多边形所在面为正向面; 至于  $SA = 0$ , 该平面与  $Z$  轴平行, 这种平面不会妨碍多面体平行于  $Z$  轴的运动, 我们不考虑.

**定理 1.** 点  $P_0$  与多面体  $V$  分离, 若沿  $Z$  轴反方向移动  $P_0$  时, 其与多面体  $V$  首先靠接于点  $P_1$ , 则  $P_1$  必在  $V$  的一个正向面上.

证明: 将视点定于  $P_0$  点, 视线沿  $Z$  轴反方向, 则显然视线首先被正向面挡住, 而挡住的点就是  $P_1$  点, 故  $P_1$  在正向面上. □

**定理 2.** 点  $P_0$  与多面体  $V$  分离, 若沿  $Z$  轴正方向移动  $P_0$  时, 其与多面体  $V$  首先靠接于点  $P_1$ , 则  $P_1$  必在  $V$  的一个逆向面上.

证明: 由于此时的视线方向为  $Z$  轴正方向, 故此时的遮挡面与定理 1 相反, 为逆向面, 所以,  $P_1$  点落在逆向面上. □

**引理 1.** 空间多面体  $V_1$  与  $V_2$  分离, 若沿  $Z$  轴正方向移动  $V_1$  时, 其与  $V_2$  外靠接, 则靠接的点必落在  $V_1$  的正向面和  $V_2$  的逆向面上.

**引理 2.** 空间多面体  $V_1$  在  $V_2$  中, 若沿  $Z$  轴正方向移动  $V_1$  时, 其与  $V_2$  内靠接, 则靠接的点必落在  $V_1$  的正向面和  $V_2$  的正向面上.

根据引理 1 和引理 2, 我们可通过坐标旋转, 把两个多面体的靠接问题转化为它们旋转后的正、逆向面(或正向面与正向面)之间的靠接问题.

**定理 3.** 线段  $P_1P_2$  与多边形  $R$  分离, 若沿  $Z$  轴正方向移动  $R$  时, 其将与  $P_1P_2$  碰撞, 则  $P_1P_2$  与  $R$  在  $OXY$  平面上的投影必有重叠.

证明: 线段  $P_1P_2$  与多边形  $R$  碰撞部位的投影就是重叠部分. □

由于我们处理的多边形都在多面体中不与  $Z$  轴平行的正向或逆向面上, 由定理 3, 多边形  $R$  所在平面不与  $Z$  轴平行, 则线段  $P_1P_2$  沿  $Z$  轴方向移动时, 其与  $R$  的可能碰撞点在  $P_1P_2$  的部分线段上, 这些线段在  $OXY$  平面上的投影就是  $P_1P_2$  与  $R$  在  $OXY$  平面上投影的重叠部分.

**定理 4.** 多边形  $R$  所在的平面不与  $Z$  轴平行, 线段  $P_1P_2$  在  $OXY$  平面上的投影全落在  $R$  在  $OXY$  平面上的投影  $R'$  上. 若沿  $Z$  轴正方向移动  $R$  时, 其与  $P_1P_2$  碰撞, 则最先碰撞的点为  $P_1P_2$  的一个端点.

证明: 由已知,  $R$  所在平面  $B$  不与  $Z$  轴平行, 且沿  $Z$  轴正方向移动  $R$  时, 其将与  $P_1P_2$  碰撞, 若  $P_1P_2$  与平面  $B$  平行, 则  $P_1P_2$  上的所有点同时与  $R$  相撞(这当然包括  $P_1P_2$  的端点); 否则, 设  $P$  为  $P_1P_2$  上的首碰点, 且  $P$  不是  $P_1$  或  $P_2$  之一,  $P'$  为  $R$  上对应于  $P$  的点, 则  $P'$  与  $P_1P_2$  所决定的平面  $C$  与  $Z$  轴平行, 并且过  $P'$  所作的平行于线段  $P_1P_2$  的直线  $P_1'P_2'$  也在该平面上. 分别过  $P_1$  和  $P_2$  作直线平行于  $Z$  轴, 它们分别与  $P_1'P_2'$  相交于  $P_1'$  和  $P_2'$ , 由于  $P'$  在线段  $P_1'P_2'$  上, 而且也在多边形  $R$  所在的平面  $B$  上, 而平面  $B$  不与  $Z$  轴平行, 故  $P_1'$  和  $P_2'$  分别在平面  $B$  的异侧, 这与  $P'$  对应的点  $P$  是首碰点的假设矛盾. 假设错误, 命题得证. □

**定理 5.** 多边形  $R$  所在的平面不与  $Z$  轴平行, 线段  $P_1P_2$  在  $OXY$  平面上的投影  $P_1'P_2'$  与  $R$  在  $OXY$  平面上的投影  $R'$  有重叠, 若沿  $Z$  轴反方向移动  $P_1P_2$  时, 其将与  $R$  碰撞, 则最先的碰撞点为投影对应于  $P_1'P_2'$  和  $R'$  重叠的各部分线段端点中的一个点.

证明: 设  $P_1P_2$  和  $R$  在  $OXY$  平面上的投影分别为  $P_1'P_2'$  和  $R'$ ,  $P_1'P_2'$  和  $R'$  重叠的部分分别为线段:  $B_1'B_2'$ ,  $B_2'B_3'$ , ...,  $B_n'B_{n+1}'$ ; 它们在  $P_1P_2$  上对应的线段分别为:  $B_1B_2, B_2B_3, \dots, B_nB_{n+1}$ ; 则由定理 3,  $P_1P_2$  与  $R$  的首碰点对应于线段族  $B_1B_2, B_2B_3, \dots, B_nB_{n+1}$  上的点, 而由定理 4,  $R$  与线段族  $B_1B_2, B_2B_3, \dots, B_nB_{n+1}$  的首碰点为  $B_i (i=1, \dots, n+1)$  之一, 命题得证. □

**引理 3.** 若线段  $P_1P_2$  和平面多边形  $R$  满足定理 5 的条件, 则首碰点为对应于线段  $P_1P_2$  上那些投影落在  $R'$  上(中)的各部分线段的端点中, 最靠近  $P_1$  或  $P_2$  的两个端点(可以是  $P_1$  或  $P_2$ )之一. □

证明: 与定理 4 的证明相似.

**定义 3.** 若沿方向  $\omega(X_\omega, Y_\omega, Z_\omega)$  移动线段  $S_1S_2$ , 其将与多边形  $A\{P_1(X_1, Y_1, Z_1), P_2(X_2, Y_2, Z_2), \dots, P_N(X_N, Y_N, Z_N)\}$  碰撞, 则碰撞前  $S_1S_2$  移动的距离为  $S_1S_2$  沿方向  $\omega$  到多边形  $A$  的  $\omega$  角角位移, 记为  $D_{\omega\omega}$ ; 否则, 若不碰撞, 则  $D_{\omega\omega} = \infty$ .

**定理 6.** 点  $P_p(X_p, Y_p, Z_p)$  沿  $Z$  轴方向移动时, 其能与多边形  $R\{P_1(X_1, Y_1, Z_1), P_2(X_2, Y_2, Z_2), \dots, P_N(X_N, Y_N, Z_N)\}$  靠接的必要条件为

- ①  $P$  在  $OXY$  平面上的正投影  $P_1$  落在  $R$  在  $OXY$  平面上的正投影  $R_1$  内;

$$\textcircled{2} Z_p \leq \max_{1 \leq i \leq N} Z_i.$$

证明: 假设(如图2所示)  $P$ 与 $R$ 靠接于点 $P'(X', Y', Z')$ , 由于点 $P'$ 在多边形 $R$ 上, 则有 $P'$ 在 $OXY$ 上的正投影 $P_0'$ 落在 $R$ 在 $OXY$ 平面上的正投影 $R_1$ 内, 且 $Z_p' \leq \max(Z_i) (i=1, 2, \dots, N)$ , 而点 $P'$ 又是点 $P$ 沿 $Z$ 轴方向移动的轨迹上的点, 则有 $Z_p' \geq Z_p$ , 且 $P$ 在线段 $P'P_0'$ 上, 即, 也是 $P$ 在 $OXY$ 上的正投影, 且 $Z_p \leq Z_p' \leq \max_{1 \leq i \leq N} Z_i$ .  $\square$

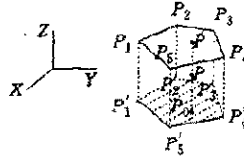


图2

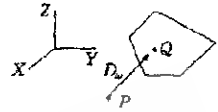


图3

定义4. 若经过点 $P(X_0, Y_0, Z_0)$ , 且方向为 $\omega(X_\omega, Y_\omega, Z_\omega)$ 的直线与多边形 $A(P_1(X_1, Y_1, Z_1), P_2(X_2, Y_2, Z_2), \dots, P_N(X_N, Y_N, Z_N))$ 相交于 $Q$ 点, 则矢量 $PQ$ 为点 $P(X_0, Y_0, Z_0)$ 沿方向 $\omega$ 到多边形 $A$ 的 $\omega$ 角位移, 记为 $D_\omega$ (如图3所示); 否则,  $D_\omega = \infty$ .

$D_\omega$ 的计算可通过坐标旋转和平移进行<sup>[15]</sup>, 其中旋转的公式为

$$[x'', y'', z'', 1] = [x, y, z, 1]K_T \quad (2)$$

$$K_T = R_1 * R_2 \quad (3)$$

$$R_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/v & b/v & 0 \\ 0 & -b/v & c/v & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \cos\alpha = c/v, \sin\alpha = b/v, v = (b^2 + c^2)^{1/2} \quad (4)$$

$$R_2 = \begin{bmatrix} v & 0 & a & 0 \\ 0 & 1 & 0 & 0 \\ -a & 0 & v & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \cos\beta = v, \sin(-\beta) = -a \quad (5)$$

这里,  $\omega$ 用方向余弦为 $(a, b, c)$ 的单位矢量表示(其中 $a^2 + b^2 + c^2 = 1$ ).

设旋转后的点为 $P''$ , 多边形为 $A$ , 再将原点平移至 $P''$ 点, 多边形变为 $B$ , 则 $D_\omega$ 就是多边形 $B$ 与 $Z$ 轴的交点的 $x$ 坐标. 若多边形 $B$ 的第1, 2和最后点的坐标分别为 $(x_1, y_1, z_1)$ ,  $(x_2, y_2, z_2)$ 和 $(x_n, y_n, z_n)$ , 则多边形所在的平面方程为

$$Ax + By + Cz + D = 0, A = \begin{vmatrix} y_1 & x_1 & 1 \\ y_2 & x_2 & 1 \\ y_n & x_n & 1 \end{vmatrix}, B = \begin{vmatrix} x_1 & x_1 & 1 \\ x_2 & x_2 & 1 \\ x_n & x_n & 1 \end{vmatrix}, C = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_n & y_n & 1 \end{vmatrix}, d = - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_n & y_n & z_n \end{vmatrix} \quad (6)$$

$$D_\omega = z = -D/C \quad (7)$$

而 $Q$ 点的坐标为

$$[X_Q, Y_Q, Z_Q, 1] = [X_0 + a * D_\omega, Y_0 + b * D_\omega, Z_0 + c * D_\omega, 1] \quad (8)$$

由于两个多面体的紧密靠接问题可通过计算坐标旋转后多面体之间的面与面间沿 $Z$ 轴的距离来解决, 而计算面之间的距离, 就是求这两组平面上多边形的边对相对多边形的最小正 $D_{\omega\omega}$ (这里 $\omega$ 与 $Z$ 轴同向或反向), 而最小正 $D_{\omega\omega}$ 又可转化为点到多边形的最小正 $D_\omega$ , 不妨设顶点 $P_0$ 与原点重合. 此时,  $D_\omega$ 就是多边形与 $Z$ 轴相交点的 $Z$ 坐标. 且 $D_\omega > 0$ 时, 相交于 $Z$ 轴的正向上, 此时的 $D_\omega$ 是候选 $D_\omega$ 之一, 即沿 $Z$ 轴正方向向前移动顶点 $P_0$ . 距离 $D_\omega$ 后, 将完成点与面的靠接;  $D_\omega = 0$ 时,  $P_0$ 已经完成与平面的靠接;  $D_\omega < 0$ 时, 相交于 $Z$ 轴的反方向上, 此时的靠接不符合原来的本意,  $D_\omega$ 可以忽略.

## 2 靠接算法

多面体靠接算法由正逆面向组提取, 面组预选和计算最小正 $D_\omega$ 并平移多面体3部分组成, 这里, 我们仅给出外靠接算法, 对于内靠接, 只需将算法中的提取逆面向改为提取正面向即可.

### 算法1. 提取正逆面向组

输入: 多面体 $A$ 和 $B$ 以及 $A$ 向 $B$ 靠接的方向 $\omega(a, b, c)$

输出: 多面体 $A$ 的正向面集合 $A_L'$ 与多面体 $B$ 的逆面向集合 $B_L'$

①根据公式(3)~(5)计算 $R_T$

②根据公式(2)旋转坐标系, 则多面体 $A, B$ 变换成多面体 $A', B'$ ;

③根据公式(1)提取多面体  $A'$  的正向面放入集合  $A_L'$ , 提取多面体  $B'$  逆向面放入集合  $B_L'$ , 返回。

#### 算法2. 面组预选与合并

输入: 面集合  $A_L'$  与  $B_L'$

输出: 点面集  $A_B''$  与  $B_A''$

①根据面组集合  $A_L'$ , 取其中的所有边, 删除其中的重复边, 然后放入边集合  $A_L''$ ;

②根据面组集合  $B_L'$ , 取其中的所有边, 删除其中的重复边, 然后放入边集合  $B_L''$ ;

③根据定理3、4和引理3, 将边集合  $A_L''$  中的所有边和面组集合  $B_L''$  中的面进行边分割和面预选, 得到点对和与点对对应的面组成的点面集合(每个元素为两个点和1个多边形)  $A_B''$ , 同样得到  $B_A''$ , 返回。

这里的第3步要用到对  $OXY$  平面进行正投影的运算, 并使用判断点是否在多边形内和多边形对线段裁剪的算法。这些算法已有成熟讨论, 有兴趣的读者可参考文献[3, 13, 14], 限于篇幅, 本文不详述。

#### 算法3. 计算最小正 $D_0$ 并平移多面体 $A$ , 使之与多面体 $B$ 靠接

输入: 点面集  $A_B''$  与  $B_A''$

输出: 最小正  $D_0$  并平移多面体  $A$ , 使之与多面体  $B$  靠接

①集合  $D$  置空;

②利用公式(6)和(7), 分别求点面集合  $A_B''$  中的每个元素的顶点至相应面的  $D_0$ , 其中的负值舍弃, 其他值并入集合  $\{D\}$ ;

③利用公式(6)和(7), 分别求点面集合  $B_A''$  中的每个元素的顶点至相应面的  $D_0$ , 其中的正值舍弃, 其他值取绝对值并入集合  $\{D\}$ ;

④从集合  $\{D\}$  中取最小值  $D_{\min}$ , 若  $D_{\min} = 0$ , 返回, 否则进入⑤;

⑤根据公式(8)和  $D_{\min}$  计算多面体  $A$  的新坐标, 至此, 完成靠接过程。

### 3 算法的复杂度分析

设两个多面体的顶点数分别为  $N_1$  和  $N_2$ , 平面数为  $M_1$  和  $M_2$ , 各多边形的最大顶点数为  $K_1$  和  $K_2$ , 则算法中第1步旋转所需循环次数为  $N_1 + N_2$ , 第2步提取面组所需循环次数最多为  $K_1 * M_1 + K_2 * M_2$ , 第3步计算  $D_0$  所需循环次数最多为  $N_1 * M_2 + N_2 * M_1$ , 整个算法的最高复杂度为  $O(N_1 + N_2 + K_1 * M_1 + K_2 * M_2 + N_1 * M_2 + N_2 * M_1) = O(\max(K_1 * M_1, K_2 * M_2, N_1 * M_2, N_2 * M_1))$ 。

尽管整个算法的最高计算复杂度并不低, 但经过算法中第1步和第2步的筛选, 本算法在实际中的计算量将远远小于最高计算复杂度中的计算量。

### 4 结论与分析

由于采用了面组提取和点面预选的方法, 本算法大大减少了空间多面体靠接的计算量, 从而具有速度快、精度高且结果可靠的优点; 此外, 在靠接的多面体是凸多面体的情况下, 计算有向面积的公式(1)可以进一步简化为

$$SP = ((x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)), \quad (9)$$

从而进一步降低算法的计算量。

致谢 感谢审者对本文的不足之处提出的指正意见。

#### 参考文献

- 覃中平, 张焕国. 确定凸多边形平移时最初碰撞部位的最优算法. 计算机学报, 1992, 15(3): 171~177  
(Qin Zhong-ping, Zhang Huan-guo. An optimal algorithm for determining the touch parts between two colliding convex polygons. Chinese Journal of Computers, 1992, 15(3): 171~177)
- 李庆华. 判定凸多边形可碰撞的最优算法. 计算机学报, 1992, 15(8): 589~596  
(Li Qing-hua. Optimal algorithm for deciding possible collision of convex polygons. Chinese Journal of Computers, 1992, 15(8): 589~596)
- 金廷赞. 计算机图形学. 杭州: 浙江大学出版社, 1988  
(Jin Ting-zan. Computer Graphics. Hangzhou: Zhejiang University Press, 1988)
- 汪嘉业. 平面上简单多边形平移时确定碰撞部位的最优算法. 计算机学报, 1992, 15(8): 583~588  
(Wang Jia-ye. An optimal algorithm for finding first contact between translating polygons. Chinese Journal of Computers, 1992, 15(8): 583~588)
- 胡华, 蔡昕, 姚俊. 任意连通多边形的靠接算法. 计算机学报, 1995, 18(11): 868~874  
(Hu Hua, Cai Xin, Yao Jun. An algorithm for polygon connection. Chinese Journal of Computers, 1995, 18(11): 868~874)
- Boyer J W. Interference detection among solids and surface. Communication of ACM, 1979, 22(1): 3~9

- 7 Uchihik T, Ohashi T, Tokoro M. Collision detection in motion simulation. *Computer & Graphics*, 1985,7(3~4)
- 8 Moore M, Wilbelms J. Collision detection and response for computer animation. *Computer Graphics*, 1983,22(4):289~298
- 9 Lozano-Perez T, Wesley Michael A. An algorithm for planing collision-free paths among polyhedral obstacles. *Communication of ACM*, 1979,22(10):560~570
- 10 Yang H M, Lu A, Zhou J. Collision detection among moving objects in matching process simulation. *Computer Aided Drafting, Design and Magnituring*, 1994,4(1):20~27
- 11 Egbert Parris K, Winkler Scott H. Collision-free object movement using vector fields. *IEEE Computer Graphics and Applications*, 1996,16(4):18~24
- 12 方伟,李叔梁.复杂运动曲面间的碰撞检测. *计算机学报*,1996,19(增刊):178~186  
(Fang Wei, Li Shu-liang. Collision detection between moving objects with complex surfaces. *Chinese Journal of Computers*, 1996,19(supplement):178~186)
- 13 唐荣锡,汪嘉业,彭群生. *计算机图形学教程*.北京:科学出版社,1990  
(Tang Rong-xi, Wang Jia-ye, Peng Qun-sheng. *Computer Graphics*. Beijing: Science Press, 1990)
- 14 孙家广,许隆文. *计算机图形学*.北京:清华大学出版社,1986  
(Sun Jia-guang, Xu Long-wen. *Computer Graphics*. Beijing: Tsinghua University Press, 1986)

### Contacting Polyhedron Fast by Abstracting Lines and Planes

HU Hua<sup>1,2</sup> HE Zhi-jun<sup>1</sup> GAO Ji<sup>1</sup> ZHANG Xing-gong<sup>1</sup>

<sup>1</sup>(State Key Laboratory of CAD & CG Zhejiang University Hangzhou 310027)

<sup>2</sup>(Department of Computer Engineering Hangzhou Institute of Commerce Hangzhou 310035)

**Abstract** The problem of polyhedron contact is concerned about in this paper. Based on geometric features of polyhedron, a method to speed up the contact of polyhedrons by translating the problem of polyhedron contact into the problem of calculating distance among a small number of points and polygons is presented. The algorithm presented here can be widely used in the fields of Graphics, Robot, Multimedia and CAD/CAM etc.

**Key words** Polyhedron, contact, angle movement, point-plane set.