INCLUSION TEST FOR GENERAL POLYHEDRA

F. R. FEITO and J. C. TORRES

1Departamento de Informática, Universidad de Jaén, Jaén, Spain
2Departamento Lenguajes y Sistemas Informáticos, Universidad de Granada, Granada, Spain

Abstract This paper presents a new algorithm which tests the inclusion of a point in a general polyhedron, manifold and non-manifold, without solving any equation system and without using trigonometric functions. The algorithm is simple and robust, and easy to apply in every case. © 1997 Elsevier Science Ltd

1. INTRODUCTION
Deciding whether a point is in a solid or not is a basic test in a great number of computer graphics applications. Usually inclusion test algorithms are based on the resolution of the equation system or use trigonometric functions [1], and most of them deal with a large number of singularities [2]. This implies, apart from inefficiency, robustness problems.

The 2-D inclusion algorithm has already been widely studied [1], but the solutions offered in 3-D are more complicated. Lane, Magedson, and Rarick proposed a solution based on the use of trigonometric functions [3]. More recently, Horn and Taylor gave a solution based on the computation of the distance between the point and every component of the solid [4].

Previously we have presented an algorithm to test the inclusion of a point in a 2-D polygon [5]. This paper may be considered as a generalization of that work [6].

The first part of this paper (Sections 2 and 3), presents a simple implementation of the algorithm presented in [5] to introduce the algorithm in three dimensions.

In Section 2 we present the fundamental definitions. In Section 3 we describe the implementation of the 2-D algorithm. In Section 4 we enunciate and demonstrate the 3-D algorithm. Finally we give a study of the computation times of the algorithm present here.

2. FUNDAMENTAL DEFINITIONS
The definitions that let us demonstrate and apply the algorithm are the following:

Definition 1. Let \( x \) be a real number, the sign function is defined as

\[
\text{sign}(x) = \begin{cases} 
1 & \text{if } x > 0 \\
0 & \text{if } x = 0 \\
-1 & \text{if } x < 0 
\end{cases}
\]

Definition 2. Let \( A, B \) and \( C \) be three points in \( R^2 \). The signed area of the triangle that has points \( A, B \) and \( C \) as vertices is denoted by \([ABC]\) and defined by

\[
[ABC] = \frac{1}{2} \begin{vmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_C & y_C & 1 \end{vmatrix}
\]

where \( A = (x_A,y_A), B = (x_B,y_B) \) and \( C = (x_C,y_C) \).

The triangle has positive orientation (counterclockwise) if the signed area is positive.

If we work with triangles in which one of the vertices coincides with the origin, we deal with original triangles. The edges that contain the origin are called original, and the remaining one non-original.

As we demonstrated in [5], to test if a point is in a polygon or not it is enough to check if it is in the original triangles, determined by each edge and the origin, and to sum the sign of each triangle properly.

We can check if a point is in the original triangle following the method expressed by the next lemma.

Lemma 1. Let \( OAB \) be an original triangle with positive orientation and \( Q \) an arbitrary point in 2-D. Then \( Q \) is in \( OAB \) if: \( \text{sign}([OQA]) \geq 0 \), \( \text{sign}([QAB]) \geq 0 \) and \( \text{sign}([QBO]) \geq 0 \).

Proof. \( Q \) is in the interior of the triangle if, and only if, it is on the left of the oriented segments \( AB \), \( BO \) and \( OA \), which implies that the signed areas \([OAB],[QBO],[QOA]\) are positive. Besides this, if one of the signed areas is zero, then the point is in the corresponding edge.

If the triangle \( OAB \) has negative orientation it would be enough to study the triangle \( OBA \).
Apparently it is necessary to calculate three determinants to decide the inclusion of a point in a triangle, but as we can see in Fig. 1 it is necessary to calculate only one determinant computing the other two by its associated minors.

We will now do a generalization of these concepts to three dimensions.

**Definition 3.** Let $A$, $B$, $C$ and $D$ be four points in $R^3$. The signed volume of the tetrahedron that has points $D$, $A$, $B$, and $C$ as vertices is denoted by $[DABC]$ and defined by (see [8])

$$[DABC] = \frac{1}{6} \begin{vmatrix} x_A - x_D & y_A - y_D & z_A - z_D \\ x_B - x_D & y_B - y_D & z_B - z_D \\ x_C - x_D & y_C - y_D & z_C - z_D \end{vmatrix}$$

where $D = (x_D, y_D, z_D)$, $A = (x_A, y_A, z_A)$, $B = (x_B, y_B, z_B)$ and $C = (x_C, y_C, z_C)$. The tetrahedron has positive orientation (from the side opposite to one point, the rest of vertices are counterclockwise) if the signed volume is positive [8].

If we work with tetrahedra in which one of the vertex coincides with the origin, we deal with original tetrahedra. The faces and edges that contain the origin are called original, and the remaining ones non-original.

As in two dimensions, the definition of signed volume lets us easily determine whether a point is in a tetrahedron or not. Besides, depending on the possible zero values, we can determine if the point is in the boundary of the tetrahedron. The next lemma gives this result.

**Lemma 2.** Let $OABC$ be an original tetrahedron with positive orientation and $Q$ an arbitrary point in $R^3$. $Q$ is in the tetrahedron if, and only if:

$$\text{sign}([QABC]) \geq 0 \text{ and } \text{sign}([QACO]) \geq 0 \text{ and } \text{sign}([-QAOB]) \geq 0 \text{ and } \text{sign}([QBOC]) \geq 0.$$  

**Proof.** $Q$ is in the tetrahedron if, and only if, it sees the three vertices of each face in the same orientation that the origin $O$ sees the vertices $A$, $B$, and $C$, which implies that the signed volumes $[QABC]$, $[QACO]$, $[QAOB]$ and $[QBOC]$ must be positive (see Fig. 2).

To obtain the algorithm we extend the previous definition to pyramids. We will consider that when the face opposite to the vertex is given in counterclockwise order the pyramid has positive orientation.

**Definition 4.** The signed volume of a pyramid $P$, with vertex $V$ and base $S$, is denoted by $[P]$ and coincides with the volume of $P$ if the pyramid has positive orientation and with minus the volume of $P$ if the orientation is negative.

If the vertex of the pyramid coincides with the origin we will say that the pyramid is an original pyramid.

We will now define the sign of a face of a polyhedron.

**Definition 5.** The sign of a face $F$ of a general polyhedron $P$, is the sign of the signed volume of the original pyramid determined by the origin and the face.

We can easily determine the sign of a face of a general polyhedron by means of the following lemma (see Fig. 3).

**Lemma 3.** Let $P = OF$ be an original pyramid and $Ax + By + Cz + D = 0$ the equation of the plane containing the base (the face), where $(A, B, C)$ is the plane normal pointing outward. Then, the sign of the pyramid (of the face), $\text{sign}(P)$, is $+1$ if $D$ is negative, $-1$ if $D$ is positive and $0$ if $D$ is zero.

**Proof.** Let $Q = (x, y, z)$ be a point of the plane containing the base. The vector $OQ$ has as coordinates $(x, y, z)$. The sign of the cosine of the angle that forms this vector with the normal to the plane $(A, B, C)$, can be calculated by means of the sign of the scalar product

$$\langle A, B, C \rangle \cdot (x, y, z) = Ax + By + Cz$$
Fig. 2. Demonstration lemma 2.
and this expression is equal to \(-D\) (because \(Q\) is on the plane). If the angle has more than 90°, the sign of the cosine is negative and so \(D\) is positive. If the angle has less than 90°, the sign of the cosine is positive and \(D\) is negative. If the plane contains the origin the value of \(D\) is zero and we have a degenerate pyramid.

With the definitions of this section we can introduce the algorithm in 2-D and 3-D.

3. TEST INCLUSION IN GENERAL POLYGONS

As we demonstrated in [5], to determine if a point \(Q\) is in a general planar polygon, it is enough to calculate the sum of the signs of the signed areas of all the original triangles that contain \(Q\), determined by the origin and each of the edges of the polygon.

To determine if it is in the boundary we previously check if it is in the edge or not.

The algorithm can be expressed as follows.

Let \(S = E_1 E_2 \ldots E_n\) be the 2-D-polygon, \(E_i\) its edges, \(T_1\) the original triangle determined by the origin and \(E_i\), and \(\text{int}(T_1)\) and \(\text{Bound}(T_1)\) the interior and boundary respectively of \(T_1\).

\[
\text{INCLUSION} = 0
\]

For \(i = 1\) To \(n\)
  
  If \(Q\) is in \(E_i\) Then \(\text{INSIDE} = \text{YES}\). Exit
  
  If \(Q\) is in \(\text{int}(T_1)\) then
    
    \[
    \text{INCLUSION} = \text{INCLUSION} + \text{sign}(\text{int}(T_1))
    \]
  
  If \(Q\) is in \(\text{Bound}(T_1)\) then
    
    \[
    \text{INCLUSION} = \text{INCLUSION} + 1/2 \times \text{sign}(\text{int}(T_1))
    \]

Endfor

\(\text{INSIDE} = (\text{INCLUSION} = = 1)\)

End

We can see that a point in a common original edge count \(1/2 \times \text{sign}([T_1])\) for each one. Anyway, in the first phase of the algorithm we can test if \(Q\) is on the edge or not.

In Fig. 4 we can see an example of the application of the algorithm in several cases.

For point \(Q_1\)

1. \(Q_1\) in \(T_1\) and \(\text{INCLUSION} = \text{sign}([OE_1]) = 1\)

2. \(Q_1\) in \(T_2\) and \(\text{INCLUSION} = \text{INCLUSION} + \text{sign}([-OE_2]) = 1 - 1 = 0\)

3. \(Q_1\) in \(T_3\) and \(\text{INCLUSION} = \text{INCLUSION} + \text{sign}([-OE_3]) = 0 + 1\)

4. \(Q_1\) is not in the rest of triangles and the final value of \(\text{INCLUSION}\) is 1

The successive values for \(\text{INCLUSION}\) are shown in Fig. 4.

We think that the importance of the algorithm is not only that time can be reduced, but that it is founded on concepts that can be easily generalized to superior dimensions, as we are going to show, studying the inclusion algorithm for general polyhedra in 3-D.

4. ALGORITHM IN 3-D

The inclusion test of points in polyhedra is an important problem in computer graphics, but it is more complex than the 2-D inclusion test.

Apart from the general algorithm based on the calculation of a number of intersections, Kalay presents another two algorithms based on reducing the problem into 2-D [2]. These two algorithms have a complex implementation, and in addition they present a great deal of singularities.

The foundation of our algorithm is the same as in the 2-D case, that is, it is based on the decomposition of any object as stated earlier. Now, the objects are considered in space and therefore expressed using original tetrahedra.

When the point of study is the origin, we can do any translation. For simplicity we make the new origin coincide with one of the vertices of the polyhedron.

We consider the original tetrahedra determined by the origin, a fixed vertex of each face and each of the rest of the edges of the same face to be non-incident in that vertex. To express the algorithm we use the following symbols:

- \(V^+(Q)\): Set of vertices determining original edges containing \(Q\) and either belonging to positive tetrahedra or are initial or final vertices of positive faces, according with the order in which appear in the face description.

- \(V^-(Q)\): Set of vertices determining original edges containing \(Q\) and either belonging to negative tetrahedra or are initial or final vertices of negative faces, according with the order in which appear in the face description.
Theorem 1. Let $P = F_1, F_2, \ldots, F_n$ a general polyhedron with faces $F_i$, $i = 1$ to $n$. Each face has vertices $F_i = V_{i1}, V_{i2}, \ldots, V_{i\ell_i}$. Then if $Q$ is a point outside the boundary of $P$

\begin{align*}
\text{value of } \text{sign}(\{T_i\}) & \\
Q_1 & : 1, -1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\
Q_2 & : 1, 0, 0, 0, 0, -1, 1, -1, 0, 0, 1, -1, 0, 1 \\
Q_3 & : 0, 0, 1/2, 1/2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
\end{align*}

\begin{align*}
\text{value of } \text{INCLUSION} & \\
1 & : 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 \\
1 & : 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, -1, -1, 0 \\
0 & : 0, 0, 1/2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
\end{align*}

\begin{align*}
\text{Fig. 4. Examples of 2-D inclusion.}
\end{align*}
The intersection point is a line segment that is coplanar with a face (notice that the point $Q$ is not on the face, but it is on the face plane). The sign is considered zero, and it does not suppose a transition.

The intersection point is interior to a face. Each intersection with a face $i$ implies an odd number of intersections with the tetrahedra associated with that face. Each intersection contributes with $\text{sign}(P_i)$ to the formula, so the total number is $+1$. This case is counted properly by the first value of $\lambda_i$.

The intersection point is interior to a face but it belongs to a non original edge common to two tetrahedra, the point $Q$ belongs to the common original face of two tetrahedra, and the contribution to the formula is $1/2 + 1/2 = 1$ if both tetraedra are positive, or $1/2 - 1/2 = 0$, if one is negative or $-1/2 - 1/2 = -1$ if both are negative. This case is counted properly by the second value of $\lambda_i$. This last case is similar to the bidimensional case of the point $Q_s$ in Fig. 4.

The intersection point belongs to an edge common to several faces. That number is 2 (for manifold polyhedra) or multiple of 2 (for non-manifold polyhedra). Two cases can be presented: to have a transition and so $+1$ intersection must be counted (see Fig. 5(b) in which $1/2 + 1/2$ is counted) or not to have a transition, case in which no intersection must be counted (see Fig. 5(a) in which $+1/2 - 1/2 = 0$ is counted). For non manifold polyhedra the situation is the same [Fig. 5(c and d)]. This case is counted properly by the second value of $\lambda_i$.

If the intersection point coincides with a vertex, we can have two cases: to have a transition or not. If we have a transition from inside to outside, or vice versa, that means that vertex is in $V^+(Q)$ and not in $V^-(Q)$, or vice versa [Fig. 6(a)]. In any case as for this vertex, $\text{card}(V^+(Q)) - \text{card}(V^-(Q))$ is $+1$ or $-1$, which is necessary for having a transition (and $\lambda_i$ is zero).

If there is not transition, as in the case of Fig. 6(b), then the vertex is in $V^+(Q)$ and in $V^-(Q)$ and $\text{card}(V^+(Q)) - \text{card}(V^-(Q))$ is 0 (and $\lambda_i$ is zero).

If $Q$ is not interior, it is exterior, as it is assumed that the point is not in the boundary. Then, every line beginning at $Q$ and continuing to infinity, intersects in an even number of points. Similarly as in the previous cases the formula is 0 which is different from 1.

This theorem allows us to decide whether a point belongs to a general polyhedron. The test can be easily implemented evaluating the formula of the theorem. The associated algorithm is shown in Fig. 7. The scheme of the algorithm is

\begin{verbatim}
INSIDE = FALSE; INCLUSION = 0; sets $V^+(Q) =$ empty, $V^-(Q) =$ empty

For each face $i$

If $Q$ in face$\langle i \rangle$ Then INSIDE = YES. EXIT
If $Q$ in first or last original edge associated to face$\langle i \rangle$
then study cases (sign of face and point in sets $V^+(Q)$ and $V^-(Q)$) (case 5)
else
For each tetrahedron$\langle j \rangle$ associate to face$\langle i \rangle$, study cases
point $Q$ in interior of an original triangle of tetrahedron$\langle j \rangle$ (case 3 or 4)
\end{verbatim}

Fig. 6. Demonstration of theorem 1. Black point denotes intersection point studied.
INCLUSION=0
V+ (Q)=empty
V- (Q)=empty
for i=1 to n {
    if (Q e Fi) { INSIDE=true; EXIT }
    if ((Q e Int(OVi) and ((sign(OFi)>0 and not Vj e V+ (Q) or
                      (sign(OFi)<0 and not Vj e V- (Q))} 
        INCLUSION=INCLUSION+sign(OFi)
        if (sign(OFi)<0)
            V+ (Q)= V+ (Q) + Vj
        else
            V- (Q)= V+ (Q)+ Vj
    } else if ((Q e Int(OVw) and ((sign(OFi)>0 and not Vj e V+ (Q) or
                      (sign(OFi)<0 and not Vj e V- (Q))))} 
        INCLUSIONS=INCLUSION+sign(OFi)
        if (sign(OFi)<0)
            V'(Q)= V'(Q) + Vj
        else
            V+(Q)= V-(Q)+ Vj
    } else for j=2 to ei-1 {
        if ((Q e Int (O,Vj,Vi) or (Q e Int (O,Vj,Vi+1))
            or (Q e Int (O,Vi+1,Vj,Vi))) 
            INCLUSION=INCLUSION + 1/2*sign(O,Vj,Vi,Vi+1)
        else if ((Q e Int (OV)) and ((sign(O,Vj,Vi,Vi+1))>0
                      and not Vj e V+ (Q)) or (sign(O,Vj,Vi,Vi+1))<0
                      and not Vj e V- (Q)) 
            INCLUSION=INCLUSION+sign(O,Vj,Vi,Vi+1)
            if (sign(O,Vj,Vi,Vi+1))<0
                V'(Q)= V'(Q) + Vj
            else
                V+(Q)= V+(Q)+ Vj
        } else if (Q e Int (O,Vj,Vi,Vi+1))
            INCLUSION=INCLUSION+sign(O,Vj,Vi,Vi+1)
    }
    INSIDE = (INCLUSION = 1)
}

Let P=F1,F2,...,Fn be a polyhedron, where Fi (i=1..n) are the faces, and the vertex of
face i are: Fi=V1i,V2i,...,Vni. Sign(OFi) is the sign of the pyramid OFi. The algorithm
decides if the point Q is inside the polyhedron; the result is obtained in the variable
INCLUSION, which will be 1 only if QeP.

Fig. 7. Algorithm.

point Q in an original_edge and point in sets
(point Q in interior of tetrahedron (case 2)
Endfor
Endif

We assume that the faces which determine the
polyhedron and the edges of each face are accessible
by means of an adequate data structure, and that the
polyhedron has a coherent orientation.

We need to implement the inclusion test for
original tetrahedra. This is solved by the lemma 3
of the Section 2 (see Fig. 2).

In Fig. 8 we can see an example of the application
of the algorithm in several cases. We suppose that Q1
belongs to tetrahedra OVs1Vs2Vs3 and OVsVsVs73

Note that case 1 occurs when Q is in plane face(i)
but Q is not in face(i)
(with $V_7 = V_32$) and does not belong to the rest of tetrahedra.

Similarly, we suppose that $Q_2$ belongs to tetrahedron $O V_31 V_33 V_34$ and does not belong to the rest of tetrahedra.

For point $Q_1$, the successive values for INCLUSION are:

$0,0,0,1,1,1,1,1,1,1,0,0,0,0,0$.

For point $Q_2$, the successive values for INCLUSION are:

$0,0,0,0,1,1,1,1,1,1,1,1,1,1,1$.

Note that the algorithm contemplates non-manifold polyhedra, as shown in Fig. 5(c and d).

The algorithm is robust because it is not necessary to solve equations and it is easy to implement because the special cases and the functions necessary are simpler. We have implemented the algorithms presented here, in a 486 Intel CPU, using Borland Pascal. Table 1 summarises the results of the test we have carried out. The first column gives the polyhedra we are dealing with (see Fig. 9), the second gives the execution time for the algorithm presented here. The time unit is millisecond.

<table>
<thead>
<tr>
<th>Solids</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure (a)</td>
<td>5.317</td>
</tr>
<tr>
<td>Figure (b)</td>
<td>7.404</td>
</tr>
<tr>
<td>Figure (c)</td>
<td>7.919</td>
</tr>
<tr>
<td>Figure (d)</td>
<td>4.073</td>
</tr>
<tr>
<td>Figure (e)</td>
<td>4.861</td>
</tr>
</tbody>
</table>

5. CONCLUSIONS

This paper has presented a new algorithm to the inclusion test of general polyhedra. The algorithm is a natural generalization of the 2-D algorithm present in [5]. The algorithm is robust and simple, and very easy to apply. The algorithm can be used for manifold and non-manifold polyhedra.

REFERENCES