COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING **26**, 118–125 (1984)

NOTE

An Efficient Point in Polyhedron Algorithm

JEFF LANE

Vulcan Software, Campbell, California 95008

AND

BOB MAGEDSON[1] AND MIKE RARICK[1]

Boeing Commercial Airplane Company, Seattle, Washington 84124

Received July 20, 1982; revised January 17, 1983

An algorithm is presented which efficiently determines whether a point is interior to or on a polyhedron boundary. Such algorithms are useful in 3-D CAD/CAM and solid modeling software as well as geoscience and mining software applications. The algorithm has advantages over others that have been published in terms of preprocessing time and ease of implementation. The algorithm presented decides whether or not a point is contained within a given polyhedron by examining how the polyhedron radially projects to the unit sphere centered at the point in question. If the point is inside the polyhedron, the net area covered by the projection is the total area of the sphere; if outside the net area covered is nil. Within the algorithm these determinations are made by using the Gauss–Bonnet formula to compute the areas of the regions on the sphere covered by radially projected faces.

1. INTRODUCTION

This paper describes an algorithm for determining whether or not an arbitrary point in $\mathbb{R}^3$ is contained in a compact region in three-space whose topological boundary is a polyhedron. The algorithm is valid even when the region has "holes" or is disconnected. Applications of the algorithm occur in geometric modeling for CAD/CAM and geoscience applications [6, 10].

The algorithm to be presented is a generalization of the line integral method for determining point containment by a planar polygonal region which need not be connected or simply connected (can contain holes and separate components) [6–9]. In the planar case, the polygonal region is represented as a set of oriented edges $\{e_1, e_2, \cdots e_n\}$. Each e_i is an edge of the bounding polygon, and in each case, its orientation is compatible with a fixed orientation on the region. If p is a point in the plane, then to determine whether or not p is in the given region, a triangle is constructed for each edge e_i by taking the endpoints of e_i and p as vertices. For each i, the angle opposite e_i is computed and signed positively, if the orientation on e_i moves counterclockwise about p, and negatively otherwise.

All of these signed angles are then summed. The result will be zero if p is outside of the region, π if it is on the interior of an edge, and 2π if it is inside. Since the result is essentially ternary in value, either 0, π, or 2π, the computation of the angles can be approximated if the maximum error per side is less than $\pi/2n$.

[1]Current address: Automation Technology Products, Campbell, California 95008.

0734-189X/84 $3.00

To motivate the extension of this technique to the three-dimensional analog—point containment by a polyhedron—we observe that the computation of the angle opposite the edge e_i is equivalent to measuring the arc length of the edge e_i radially projected onto the unit circle centered at p.

The obvious and, as we will soon show, the correct analogy in three dimensions is to compute and sum the signed (relative to orientation) surface areas of the polyhedron's faces radially projected to a unit sphere. This sphere should be centered at the point to be tested. If the polyhedron satisfies the relatively loose constraints laid down in the formal definitions that follow, the sum will be zero if the point is outside the polyhedron, 2π if on the interior of a face, and 4π if inside.

The algorithm as presented here will accommodate without preprocessing the so-called boundary representations of solids as described by Baumgart [1], Eastman and Weiler [2], and others. In fact, it should work well with any boundary representation that provides for each face a list of all oriented boundary loops. The algorithm complexity is linear with the number of edges, as is its planar analog and other 3-D algorithms occurring in the literature. The procedure is particularly simple, in essence requiring only the computation and summation of the interior angles of a polyhedron on the sphere, and is suitable for firmware or hardware implementation. Although algorithms with better than linear expected behavior are possible if the data (polyhedron) are presorted and optimized for the point in polyhedron tests [11], the sort is time and space consuming and not natural for other accesses to the data.

2. THE ALGORITHM

We now move to a description of the algorithm. For this purpose we rely on brief, intuitive definitions of the key concepts in an effort to avoid clouding the issue with excessive formalism. Sharp definitions and a formal proof of the algorithm's correctness will be provided in Section 3.

Briefly, a polyhedron is a closed, piecewise planar surface that bounds a "solid" in $\mathbb{R}^3$. A polyhedral region is the "solid" bounded by a polyhedron. The faces of a polyhedron are its planar pieces (assumed to be finite in number). The loops of a face are the closed, non-self-intersecting polygonal curves that bound it.

The orientation of a polyhedral region is provided by normals to its faces which point outward relative to the region. This along with the right-hand rule provides a preferred direction or flow about the polygon bounding each face.

Suppose we are given a polyhedral region R with bounding polyhedron $P(R)$, and a point p in three-space to be tested for containment by R; let $F_1, F_2, \ldots, F_n$ be the set of all the faces of $P(R)$.

If it happens that p is a vertex of any of the faces above, we are done—the point being tested is on the polyhedron. Otherwise, we wish to compute the area $A(F_i)$ of the spherical polygonal region resulting from the projection of F_i radially to the unit sphere centered at p. From an application of the Gauss–Bonnet formula [5] to $A(F_i)$ we have

$$A(F_i) = \sum_{j=1}^{l} \alpha j + (2(s - r) - l)\pi$$

where the α_j are the interior angles of the polygonal region F_i projected on the sphere and where s and r are the numbers of outer and inner loops, respectively, and l is the number of edges per face F_i. Now the α_j correspond to the oriented angles of intersection of the two planes passing through p and the sides of F_i, and are easily calculated, being careful to follow the orientation of the edges of F_i.

It is important to note that the case of coincidence with an edge must be detected during angle-between-plane calculation prior to the area calculations. When the list of faces is exhausted with no "on edge" condition detected, the sum

$$\sum_{i=1}^{n} \delta(i) A(F_i)$$

is computed where $\delta(i) = 1$ if the radial projection is orientation preserving and $\delta(i) = -1$ otherwise. The projection is said to be orientation-preserving on a face if it maps the flow about any outer loop to one that is compatible with the outward normal to the sphere. The result will be 4π if p is in R, 2π if p is interior to a face, and 0 if p is outside.

The case of the point on a polyhedron can be detected without computing the final sum. If for any F_i, $A(F_i) = 0$, then the point p lies on the plane of F_i and the planar point in the polygon algorithm (described in the Introduction) can be applied. Note that as in the planar case, the sum to be computed is ternary in value and therefore needs to be computed to no more than three bits of precision. If the maximum magnitude of the error per face area is less than π/n, then the correct answer can be computed.

We will now present a proof of the algorithm.

3. PROOF OF THE ALGORITHM

We begin by giving a few key definitions and representational ground rules on which the validity of the algorithm depends.

DEFINITION. A polygonal region is a nonempty open subset of a plane whose topological closure is compact and whose boundary, called a polygon, is a piecewise linear curve with a finite number of line segments. A closed polygonal region is the closure of a polygonal region.

DEFINITION. A polyhedral region, R, is a nonempty open subset of $\mathbf{R}^3$ whose closure is compact and whose boundary, called a polyhedron, is a union of a finite number of closed polygonal regions, called faces.

Remark. Both polygonal and polyhedral regions can be disconnected and have holes.

We will assume that $\mathbf{R}^3$ is endowed, once and for all, with the orientation determined by the right-hand rule. Then, if R is any polyhedral region in $\mathbf{R}^3$, an orientation is automatically induced on R which in turn induces an orientation on its boundary, $P(R)$, and all of its faces. This orientation on a face, F, uniquely determines an orientation or flow around the connected components of its bounding polygon and thus on its edges. From now on, whenever reference is made to an orientation of a face, it will always be this induced one. In the case of polygons and their edges, orientations will be used (and make sense) only when they are induced by the orientation on a specific containing face.

Remark. The orientation on a face F is equivalent to a preferred direction or flow around an outward-pointing (relative to the polyhedral region) normal to F. In our case, that flow is counterclockwise. By definition any polyhedron can be written as a finite union of faces, any two of which meet at most at their bounding edges. Further, the faces can be chosen so that the orientation on each (induced by R) is compatible with a fixed one on its containing plane. We will assume, then, that if R is any polyhedral region, $P(R)$ will be represented as such a union. It is not hard to see that the boundary of any face F can be written as a finite union of simple (non-self-intersecting) closed polygonal curves, which we call loops, with the following property.

If P is the plane containing F, then the bounded components in P determined by any two of these loops are either disjoint or one bounded component is contained entirely within the other.

Remark. It is well known that any loop, L, in a plane separates that plane into two disjoint connected components exactly one of which is metrically bounded. It is this bounded component that is referred to above, and it is called the bounded component determined by L.

Let F be a face of a given polyhedral region and let L be one of a system of loops, as above, of F. Let N be a normal to F that is outward-pointing relative to the polyhedral region.

DEFINITION. The loop L is said to be an outer loop if the orientation (flow) on L induced by F is counterclockwise about N (i.e., it is compatible with the orientation of F). Otherwise, L is said to be an inner loop.

Any face, F, of $P(R)$ will be assumed to be represented by two sets of loops, $O(F)$ (the outer loops of F) and $I(F)$ (the inner loops of F), which together satisfy the criteria defined above. Loops, in turn, are assumed to be represented by their (oriented) edges.

The verification of the mathematical correctness of the algorithm is based on the following well-known result. We state this without proof in the context of the definitions and notations of this paper.

THEOREM 1. *Let R be a polyhedral region in $\mathbb{R}^3$, $p \in \mathbb{R}^3$ with $p \notin P(R)$, and let L be a ray starting at p that has an empty intersection with any edge of $P(R)$. Then L intersects $P(R)$ in a finite number, n, of points; further, n is odd if p is inside R, and n is even if p is outside R.*

Now, we establish the key fact on which the algorithm is based.

First, by radial projection we mean the map, ρ, defined by $\rho(x) = x/\|x\|$ for $x \in \mathbb{R}^3$, $x \neq \theta$ and $\|x\|$ is the Euclidean norm of x.

THEOREM 2. *Without loss of generality assume p is the origin θ. Using the same hypothesis and notations of the previous section, and assuming that $\theta \notin P(R)$, we have*

$$\sum_i \delta(i) A(F_i) = 0, \text{ when } \theta \notin R$$

$$= 4\pi, \text{ when } \theta \in R$$

where

$$\delta(i) = 1 \quad \text{if radial projection is orientation preserving on } F_i$$

$$= -1 \quad \text{otherwise.}$$

Remark. The orientation on the unit sphere S^2 we assume to be given by the outward-pointing normal field and it is further assumed that this is compatible with the fixed orientation on $\mathbb{R}^3$.

Proof. Let X_i denote the characteristic function of the image of F_i on the sphere. i.e.,

$$X_i(x) = 1 \quad \text{if } x \in \text{image } F_i$$
$$= 0 \quad \text{otherwise.}$$

Then

$$\delta(i)A(F_i) = \int_{S^2} \delta(i) X_i \, dS$$

where dS is the standard element of surface area on the unit sphere, S^2. Thus, we can write

$$\sum_{i=1}^{n} \delta(i)A(F_i) = \int_{S^2} \sum_{i=1}^{n} \delta(i) X_i \, dS.$$

To prove the theorem, it is sufficient to prove that

$$\sum_{i=1}^{n} \delta(i) X_i = 1 \quad (\text{almost everywhere on } S^2) \text{ if } \theta \in R$$
$$= 0 \quad (\text{almost everywhere on } S^2) \text{ otherwise}$$

since $\int_{S^2} dS = 4\pi$. To see this, let

$$Z = \{x \in S^2 | x \text{ is in the image of some edge of } P(R)\}.$$

Clearly, Z has measure zero in S^2. If x_0 is in the complement of Z in S^2, then evidently the ray, L, starting at the origin and passing through x_0 meets $P(R)$ in finitely many points, $x_{i_1}, \ldots, x_{i_k}$, each of which are in the interiors of distinct faces, say $F_{i_1}, \ldots, F_{i_k}$, respectively. We assume that the ordering $i_1, i_2, \ldots, i_k$ is according to distance from θ with the intersection point x_{i_1} being closest.

First, we note that for $i \neq i_1, i_2, \ldots, i_k$, it is evident that

$$X_i(x_0) = 0,$$

which gives

$$\sum_{i=1}^{n} \delta(i) X_i(x_0) = \sum_{j=1}^{k} \delta(i_j) X_{i_j}(x_0).$$

Now, if θ is inside of R, then at the point x_{i_1}, the ray L, as we move away from θ, is exiting R. In view of our orientation conventions, this clearly implies that radial projection is orientation-preserving on F_{i_1}. Hence, $\delta(i_1) = 1$. But then L must be entering R at x_{i_2} and thus $\delta(i_2) = -1$. Arguing in this fashion, we see that the $\delta(i_j)$ will be alternating in sign as L alternately exits and enters R. Hence, we can conclude that

$$\delta(i_j) = 1 \qquad \text{if } j \text{ is odd}$$
$$= -1 \qquad \text{if } j \text{ is even.}$$

But by Theorem 1, k is odd so that

$$1 = \sum_{j=1}^{k} \delta(i_j) = \sum_{j=1}^{k} \delta(i_j) X_{i_j}(x_0)$$

since $X_{i_j}(x_0) = 1$ for $1 \leq j \leq k$, and this is precisely the desired conclusion.

Now, observe that if $\theta \notin R$, L must be entering R at x_{i_1} so that $\delta(i_1) = -1$. Now arguing as above and using Theorem 1 to deduce that k is even, we get

$$\sum_{j=1}^{k} \delta(i_j) X_{i_j}(x_0) = 0,$$

and the theorem is proved.

COROLLARY 3. *Let R be a polyhedral region in $\mathbf{R}^3$ and let $F_1, \ldots, F_n$ be the faces of $P(R)$. Suppose that θ is an interior point of F_j for some $1 \leq j \leq n$ (in the topology of the plane containing F_j). Then in the notation of Theorem* 2,

$$\sum_{i \neq j} \delta(i) A(F_i) = 2\pi.$$

Proof. Since θ lies interior to F_j, $\theta \notin F_i$ for any i, $i \neq j$. Thus, since each of the finitely many F_i is compact, there is an $r \notin \mathbf{R}$ with $r > 0$ such that the closed ball, $B(r)$, of radius r centered at θ is disjoint from F_i for $i \neq j$. Let T be a tetrahedron contained entirely in $B(r)$, one of whose faces, say H_0, contains θ in its interior, lies in the plane of F_j, and hence interior to F_j. Clearly, such a tetrahedron always exists.

Now, replace F_j by the new face $\tilde{F}_j$ constructed by removing the interior of H_0 from F_j.

Then the remaining faces of T, say H_1, H_2, H_3 along with $\tilde{F}_j$ and the F_i, $i \neq j$ form the face list for a new polyhedral region $\tilde{R}$ with the property that $\theta \notin P(\tilde{R})$. Assume that θ is inside $\tilde{R}$, which can always be arranged.

Then we have by Theorem 2

$$4\pi = \sum_{i \neq j} \delta(i) A(F_i) + \sum_{l=1}^{3} \delta(H_l) A(H_l) + \delta(\tilde{F}_j) A(\tilde{F}_j)$$

where the A's and δ's are defined as before.

Now since $\tilde{F}_j$ and θ are coplanar, the image of $\tilde{F}_j$ under radial projection is a curve on the sphere. More precisely, its image is exactly the great circle, S, on the sphere which is contained in the plane of $\tilde{F}_j$. As a result, $A(\tilde{F}_j) = 0$.

A moment's reflection on the construction of T, its convexity, and the "inside" status of θ will reveal that radial projection maps

$$H_1 \cup H_2 \cup H_3$$

in a one-to-one, orientation-preserving fashion onto one of the hemispheres determined by S. Hence

$$\sum_{l=1}^{3} \delta(H_l)A(H_l) = \sum_{l=1}^{3} A(H_l) = 2\pi$$

and the desired result is established.

Remark. Theorems 1 and 2 and Corollary 3 all have analogous formulations in $\mathbb{R}^n$ for any $n \geq 1$ and can be proved by virtually the same lines of argument as used above. Further, by somewhat more sophisticated techniques, one can arrive at similar results without the piecewise linear restriction. In particular, this is the case for the establishment of the validity of the three-valued planar containment algorithm outlined in the Introduction.

5. CONCLUSIONS

The algorithm has been implemented in PASCAL on a VAX 11/780. It appears reasonably fast and extremely robust. In the example below, the barbell consists of 232 faces with the "barbell axis" coincident with the z axis and one unit-radius "ball" centered at the origin and the other centered at $(0, 0, 4.2)$. The algorithm was tested with several points inside and outside the polyhedron as tabulated below. Related applications of the algorithm include mass property calculations and area filling [3, 4].

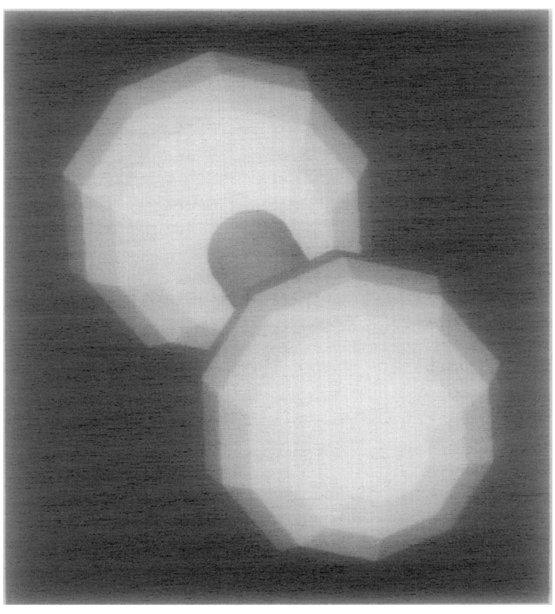

FIG. 5.1. Barbell consisting of 232 faces.

Point	In/out	CPU (sec)
(0, 0, 1)	In	3.41
(1, 1, 2)	Out	3.31
(0.3, 0.3, 2)	Out	3.19
(0.2, 0.2, 2)	In	3.37
(1, 0, 0)	On	3.29
(1.1, 0, 0)	Out	3.31

REFERENCES

1. B. G. Baumgart, Geometric Modeling for Computer Vision, Stanford Artificial Intelligence Laboratory, Memo AIM 249, Computer Science Department Report CS-463, October 1974.
2. C. M. Eastman and K. Weiler, Geometric Modeling Using the Euler Operators, Institute of Physical Planning, Carnegie–Mellon University, Pittsburgh, Pa., Research Report 78, February 1979.
3. A. M. Messner and G. Q. Taylor, Algorithm 550 solid polyhedron measures Z, *ACM Trans. Math. Software* **6**(1), 1980.
4. J. M. Lane, R. Magedson, and M. Rarick, An algorithm for filling regions on graphics display devices, to appear.
5. M. Spivak, *A Comprehensive Introduction to Differential Geometry*, Vol. III, Publish or Perish, Boston, Mass., 1975.
6. J. K. Hall, PTLOC—A FORTRAN subroutine for determining the position of a point relative to a closed boundary, *J. Math. Geol.* **7**(1) 1975, 75–79.
7. J. A. Griffith, Tape-oriented hidden line algorithm, *Comput. Aided Des.* **13**(1), 1982, 1–21.
8. M. A. Sandys, Letter to the editor, *Comput. J.* **21**(3), 1979.
9. R. F. Sproull and I. E. Sutherland, A Clipping Divider, Proc. AFIPS, FJCC, 1968, p. 765.
10. A. Requicha and H. Voelcher, eds., Solid modeling, *IEEE Computer Graphics and Applications*, March 1982.
11. K. B. Salmon, An efficient point in polygon algorithm, *Comput. Geosci.* **4**, 1978, 173–178.