PAK Publishing Group
Growing Knowledge for Future

Proceedings Book of ICETSR, 2014, Malaysia
Handbook on the Emerging Trends in Scientific Research
ISBN: 978-969-9347-16-0

A New Algorithm for 3D Containment in CAD and GIS Applications

Ali Said
Associate Professor in surveying and GIS, Civil Engineering Department, Faculty of Engineering, Tripoli University

Abstract

Computer Aided Design (CAD) and Geographical Information System (GIS) have been used extensively for a wide range of applications in recent decades. It is used to create more realistic images that represent different physical objects than were possible before. In the construction and processing of models, their information must be sufficient, complete and have no contradiction.
Some problems frequently arise during CAD and GIS applications and many of these problems can be naturally set in geometric representation. Solving these problems helps in many CAD and GIS applications. This paper focuses on one of these problems; (3D – Containment). A new algorithm has been designed to solve this problem. The results obtained showed a better performance than other algorithms that treat the same problem.
Keywords: 3D-object, Polyhedron, CAD, GIS.

1. Introduction

The need for an efficient algorithm that determines if an arbitrary point lies within a polyhedron boundary arises frequently in many applications concerned with information (GIS 3D applications). Such algorithms have therefore been studied extensively (e.g. Kalay, 1989; lane et al, 1984; Horn and Taylor, 1989; Linhart, 1990).

Several techniques may be used to determine the position of a point relative to 3D – object (polyhedron) in 3D – space. The simplest technique that is used to solve this problem might be the use of the parity number of intersections of a ray radiated from that point with the boundary of 3D - object (polyhedron). Some difficulties arise when the ray passes through a vertex or if the ray coincides with one of the edges or the faces of the polyhedron, (i.e. singularity cases). A new algorithm has been developed that treats the singularity cases in a good manner and solves the problem fast and in an efficient way.

2. Materials and Method

Several 3D – objects were created using AutoCAD software. The objects differ from each other in the number of entities (Vertices, Edges, and Faces) that construct each object, and in the shape of their faces. Because of these differences, the 3D – objects are considered to be as simple as, (tetrahedron, and hexahedron), or five points covering all cases (Inside, Outside, Vertex singularity, Edge singularity, and Face singularity) are tested against each 3D – object, and the processing time was computed.

3. Point in Polyhedron Test

The point – in – polyhedron test will determine the containment relationship in 3D – space. Given a point and a polyhedron, does the point lie inside or outside the polyhedron? Fig. 2, illustrates the general structure of the point – in – polyhedron test algorithm as developed. There is a three-stage approach in testing a point against a polyhedron. The stages are:

1- Reducing the dimensionality of the problem to the solvable two – dimensional case (point – in – polygon) using parallel projection method.
2- Applying the point – in – polygon test to determine whether the image of projected test point lies inside or outside the image of projected face using point – in – polygon algorithm developed by Saalfield (1987) and modified by Taylor (1989).
3- Comparing the test point against the plane of a face to determine whether it is in front or behind that face (In front – Behind rule).

For testing a point against each face of polyhedron, the face is defined by the plane equation as:

$$a_i x + b_i y + c_i z + d_i = 0$$

And

$$a_i = y_{21} \times z_{31} - z_{21} \times y_{31}$$
$$b_i = z_{21} \times x_{31} - x_{21} \times z_{31}$$
$$c_i = x_{21} \times y_{31} - y_{21} \times x_{31}$$

If the test point defined by its coordinates as (xtp, ytp, ztp), then the distance from test point to the plane face (SR) is calculated as:

$$SR = a_i x_{tp} + b_i y_{tp} + c_i z_{tp} + d_i$$

A positive value of SR indicates that the test point lies in front of the face, and it is behind the face if the sign of SR is negative. If the value of SR is equal to zero, then the test point lies on that face. All polyhedron faces should be tested against the test point and the number of in front or behind will be counted. If the number of in front or behind is odd, then the test point lies inside the polyhedron, otherwise it lies outside the polyhedron.

4. Results

The algorithm was written in C^{++} code and implemented on two computer operating systems (Windows and UNIX systems). The processing time was computed for each 3D – object in both systems. Tables (1 and 2) show the processing time for the test in both systems.

Table-1. Processing time of five cases using Widows operating system

The Case	Processing Time (sec.)		
	Tetrahedron	Hexahedron	Complex 3D-object
On a vertex	0.0051	0.0059	0.017
On an edge	0.0057	0.0043	0.019
Inside	0.0072	0.0061	0.019
Outside	0.0047	0.0039	0.015
On a face	0.008	0.0053	0.019

5. Conclusion

From figures (1, 2, and 3), it is clear that the processing time recorded by UNIX operating system appears to be much faster than the time recorded by Windows operating system. The processing time of this algorithm is controlled by two factors:

1- The total number of faces in the 3D – object to be tested.
2- The orientation of these faces.

Table-2. Processing time of five cases using UNIX operating system

The Case	Processing Time (sec.)		
	Tetrahedron	Hexahedron	Complex 3D-object
On a vertex	0.0007	0.0008	0.0023
On an edge	0.0008	0.0005	0.0026
Inside	0.0009	0.0008	0.0025
Outside	0.0005	0.0004	0.0017
On a face	0.0011	0.0007	0.0025

In some cases like tetrahedron and hexahedron, the orientation of the faces is more important than the number of faces. The algorithm spends more time testing the points against the tetrahedron than the time spent in testing the point against the hexahedron that has more faces, and this happen because of the faces orientation of both 3D – objects.

Fig-1. Timing analysis of five cases using Windows operating system

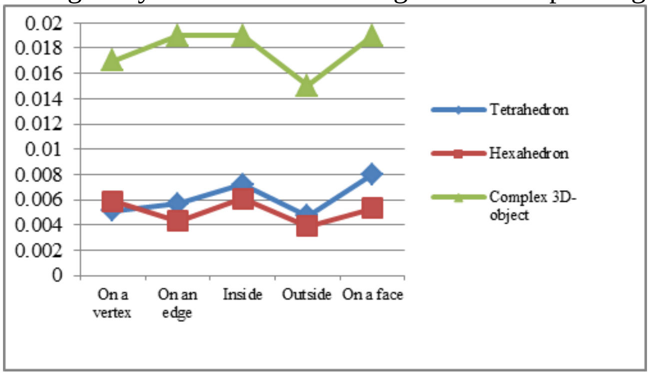


Fig-2. Timing analysis of five cases using UNIX operating system

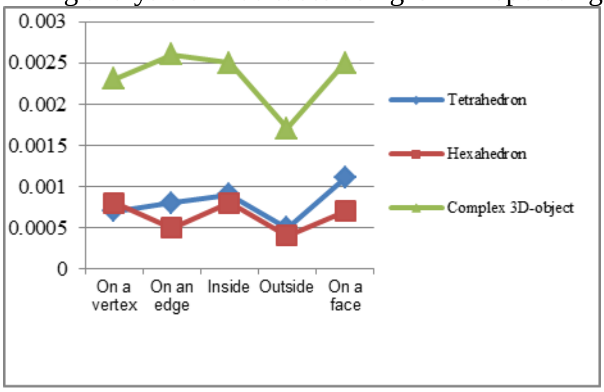
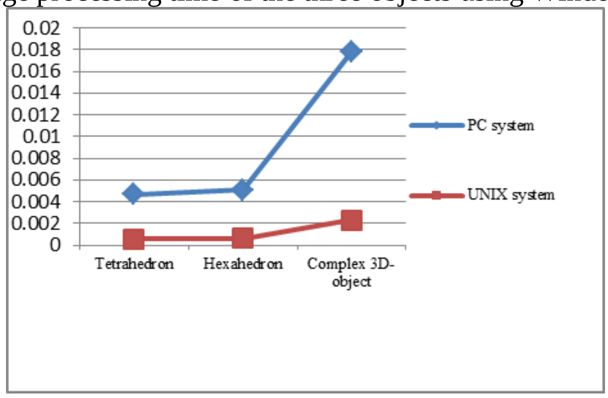


Fig-3. Average processing time of the three objects using Windows & UNIX



References

Buchanan, H. J., (1996). Computational techniques for determining three dimensional topology, in: Innovations in GIS 3, ed. by D. Parker. London, UK: Taylor & Francis.

Horn, W. P. and Taylor, D. L., (1989). A theorem to determine the spatial containment of a point – in – planner polyhedron. Computer Vision, Graphics and Image Processing, 45: 106 – 116.

Linhart, J., (1990). A quick point in polyhedron test. Computer Graphics, 14(3/4): 445 – 447.

Taylor, G. E., (1989). Addendum to saalfield (1987). International Journal of Geographical Information Systems, 3(2): 192 – 193.

Mr. Ali Said; Associated Professor at civil engineering department, faculty of engineering, Tripoli university. He got his B.Sc. in civil engineering 1986 from Tripoli university. his M.Phil. in surveying and GIS 1996 from Newcastle university, UK.

His interest in remote sensing applications for monitoring environment changes, mapping from satellite images, GIS applications in mining, map projection systems. He has a lot publications in these areas.