The Unidirectional Edge Method: A New Approach for Solving Point Enclosure Problem for Arbitrary Polygon

Faisal Mahbub Chowdhury and Kamrul Hasan Talukder
Computer Science and Engineering Discipline, Khulna University, Khulna 9208, Bangladesh

Abstract: For point enclosure problem relative to an arbitrary polygon, there exists two well known solutions – Ray-Shooting method and Signed Angle method. This paper explores a new approach for point enclosure problem which is relatively easier to implement with respect to the existing procedures. This method takes on account the direction of the edges from left to right. Along with the working process of the proposed method, a brief study on the existing methods is also included.

Key words: Arbitrary polygon, Ray-Shooting method, Signed Angle method

Introduction
A calculation that must be performed billions of times in every movie that uses computer graphics is to check if one object is hidden by another object in front of it. This calculation in turn is based on another calculation that checks whether or not a test point lies inside or outside the boundary of an object. Objects are often considered as polygons. A polygon is a closed curve in the plane composed of straight line segments. The segments are called the edges or sides of the polygon. It may be either convex or concave (Fig. 1).

Principally, two methods have been adopted for solving point enclosure problem in 2-D polygons. One of them is Ray-Shooting method in which infinite ray is drawn from the test point and the no. of intersections by that ray is used as the decision parameter. In the other method, known as Signed Angle method, the sum of the angles made between the test point and each pair of points making up the polygon is considered to take decision about the position of the test point.

In this paper, we presented a different strategy other than the existing methods which considers the edges of the polygon as unidirectional (either left to right or right to left) and counts how many times the test point is on the left (or right) to state its location with respect to the polygon.

A Quick Look to Existing Methods: In this section the methods that are used frequently for point enclosure problem is discussed briefly.

Ray-shooting Method: This method adopts the following strategy (Michael, 2002 and Eric Hiob, 2004):
Draw a ray (line) from the test point out to infinity in any direction.
Count the number of times the ray crosses the boundary of the object.
If it crosses an odd number of times then the test point is inside the object. If it crosses an even number of times then the test point lies outside. Fig. 2 shows the ray for some sample points and should make the technique clear.

Signed Angle Method: A solution forwarded by Philippe Reverdy is (Michael, 2002 and Paul Bourke, 2004):
Compute the sum of the angles made between the test point and each pair of points making up the polygon. If this sum is -2p then the point is an interior point, if 0 then the point is an exterior point.

Fig. 3 represents two examples of Signed angle method. Let Ai denote the signed angle at a relative to the boundary of Pi, where Pi has clockwise sense of rotation. In Fig. 3(i), the sum of the angles, A = A1 + A2 + A3 = 0 + 0 + 0 = 0, i.e. a lies outside of the polygon; while in Fig. 3(ii), the sum of the angles, A = A0 + A1 + A2 + A3 + A4 = -360 + 0 + 0 + 0 + 0 = -360.

Some Other Possible Solutions: A possible solution according to Dave Rusin (www.math.niu.edu/~rusin/known-math/95/polygon.interior) is: a point P lies inside the polygon P0 P1 P2 ... Pn P0 iff the area of the polygon is the sum of the areas of the two polygons Pn P0 P Pn and P P0 P1 P2 ... Pn P.
Another proposed solution by Liam Roche (www.math.niu.edu/~rusin/known-math/95/polygon.interior) is based on the concept of winding number. According to him:
Think of the polygon and the point as being in the complex plane.
Assume the point is at the origin (subtract it from all of the vertices of the polygon n).
Chowdhury and Talukder: The unidirectional edge method

List the arguments of all of the points of the polygon: a(0),...a(n).
Add up the angles subtended by each of the sides of the polygon: a(0) -a(1), .. a(n)-a(0) (mod 2*pi) with these angles being taken in the range (-pi, pi). If the answer is non-zero, the point is inside the polygon.

Proposed Method
The Unidirectional Edge Method: A well-known strategy for the convex polygons is --- one can consider such type of a polygon as a "path" from the first vertex. A point is on the interior of this polygon if it is always on the same side of all the line segments making up the path (in Fig. 4 'a' is not an interior point).
Our strategy is actually a modification of the approach mentioned above which works correctly for any arbitrary polygon.

Working Process of the Method: Instead of considering a path among the vertices (i.e. clockwise or anti-clockwise traversal through edges) we considered all the edges as unidirectional (i.e. either all of their directions are from left to right or from right to left). Hence, a point ‘a’ will lie outside of a polygon if it lies in total an even number of times ('0' will be considered as an even number) on any side of those edges (up or down, here 'up' means 'left' and 'down' means right) for which the x-coordinate value of ‘a’ is within the range of the x-coordinate values of the starting and ending points of those edges. For example, consider the polygon and the point ‘a’ in Fig. 5. The x-coordinate value of ‘a’ is within the range of the x-coordinate values of the starting and ending points of edge 1 and edge 5 and ‘a’ lies on the same side for both the edges (’2’ times on the right and ’0’ times on the left). Hence, ‘a’ lies outside of the polygon.
More examples are given in Fig. 6. Here, ‘a’ lie ’4’ times on the right and ’0’ times on the left, ‘b’ lie ’3’ times on the right and ’1’ time on the left, ‘c’ lies ’2’ times on the right and ’2’ times on the left and d lie ’1’ time on the right and ’1’ time on the left.
Hence, only the points ‘b’ and ‘d’ are interior of their respective polygons.

Special Cases: One special case is what to do when a point is located on an edge of a polygon. In that case, it depends on the required solution whether to consider the point as it is inside the polygon or not. Again, if the point’s x-coordinate is equal to that of a vertex of the polygon then there will be two edges for it --- one incoming and another outgoing. For this situation only one of those edges will be considered.

Pseudo Code of the Proposed Method: Given a polygon POLY and a point p. Consider the direction of the edges of POLY from left to right. The required solution is whether p is inside POLY or not.
Initialize a variable int_Left:=0
For each edge E of the polygon,
If the x-coordinate of p equals to that of one of the vertices of E (let the vertex is v) then
int_Left = int_Left+1
Mark E as Examined
Mark the other edge from v as Examined
If the x-coordinate of p lies within the range of the x-coordinate values of the vertices of the edge and p is on the left of E then
int_Left = int_Left+1
Mark E as Examined
If int_Left is odd then p is inside the polygon
Otherwise p is in outside.

Comparing the Proposed Method with the Existing Methods: For a sample point (3, 2) and a sample polygon with 6 vertices as (1, 1), (3, 3), (4, 6), (5, 8), (8, 6) and (7, 1) we tested the solution time required for our proposed algorithm, the Ray-shooting and the Signed-Angle algorithm. For the different number of times with the same data the responses of the algorithms are shown in Table 1.

The experiment clearly shows that the proposed method runs faster than the existing methods. The reasons for this better performance are -
In Ray-shooting method, for each of the edge it is checked that whether a "bounding box" for the edge encloses the test point If this condition satisfied then the y-coordinate of the point where the polygon's edge and the infinite vertical ray from the test crosses is calculated; which in turn is used to take the required decision.
Table 1: Execution time taken by different methods for different number of same test data. Times are rounded in ‘Seconds’. [For this experiment we used a PC having Intel Celeron 1.7 GHz processor and 128 D.D.R. RAM]

<table>
<thead>
<tr>
<th>Number of Data</th>
<th>Time Taken By The Proposed Unidirectional Edge Method</th>
<th>Time Taken By Ray-shooting Method</th>
<th>Time Taken By Signed-Angle Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>5,000</td>
<td>11 sec.</td>
<td>12 sec.</td>
<td>11 sec.</td>
</tr>
<tr>
<td>10,000</td>
<td>24 sec.</td>
<td>25 sec.</td>
<td>24 sec.</td>
</tr>
<tr>
<td>20,000</td>
<td>48 sec.</td>
<td>49 sec.</td>
<td>48 sec.</td>
</tr>
<tr>
<td>40,000</td>
<td>95 sec.</td>
<td>98 sec.</td>
<td>96 sec.</td>
</tr>
</tbody>
</table>

Whereas in our method, we just calculate whether the test point lies within the range of the x-coordinate values of the vertices of the edge; if it does so then whether it is on the on the left or right. So, our method is much simpler to implement and also will require less running time than those of the Ray-shooting method.
Chowdhury and Talukder: The unidirectional edge method

Fig. 6: More examples of the point enclosure problem solution based on unidirectional edge method

We do not need any extra operations to calculate quantities such as angles which is required in Signed-Angle method. So, our method performs better than Signed-Angle method, too.

Conclusions
Determining whether or not a point \((x,y)\) lies inside or outside a 2D polygonal bounded plane is necessary for example in applications such as polygon filling on raster devices, hatching in drafting software, determining the intersection of multiple polygons and in many other situations else. So, a faster, simple and efficient method for point enclosure problem can really help to speed up such applications. Our proposed method is unique in this stand.

References