

## 1 Winding Number Algorithm

The winding number algorithm was the algorithm that we first built to test insideness. We define the winding number as a function  $f$ . A point  $P = (x, y)$  where  $x, y \in R$  is inside the closed curve  $C$ , if for all  $V_i$  in  $C$  where  $0 \leq i \leq n - 1$ :

$$f(P, C) = \frac{1}{2\pi} \sum_{i=1}^{n-1} \arccos \frac{(V_i - P)(V_{i+1} - P)}{|V_i - P| \cdot |V_{i+1} - P|}$$

If a point is inside the curve  $C$  then the winding number has to be greater than or equal to 1.

## 2 Ray Casting Algorithm

The ray casting algorithm works by extending a given point  $Q$  to the right infinitely. Lets call that ray  $\vec{Q}$ . The approach here is to think of one point as an infinite ray. If the point lies outside the polygon, that implies that the sum of intersections that occurred should add up to zero. If the intersection index is anything but zero, then the point is outside the polygon. A intersection index is defined as a counter which counts the intersections. The figure 1 illustrates the

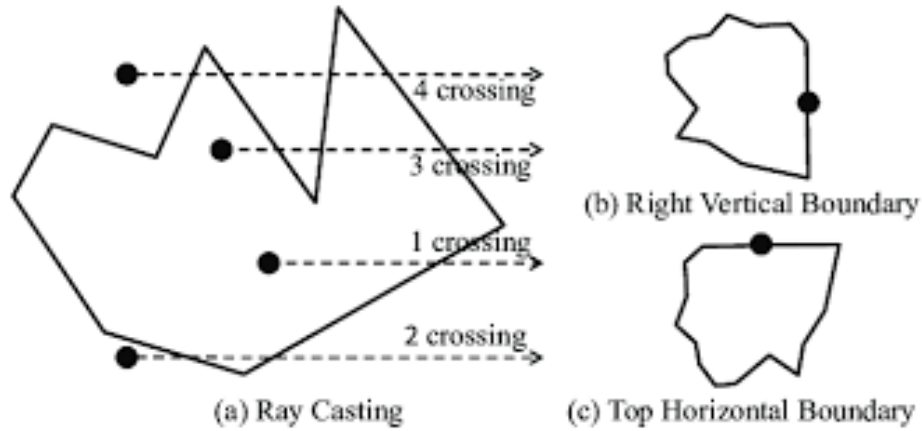


Figure 1: Ray Casting Visualization

algorithm. The pluses and minuses indicate intersections. A pluse indicates a intersection from the right of the polygon. A minus indicates a intersection from the left of the polygon. An intersection from the right adds one to the intersection index. An intersection from the left subtracts one from the intersection index. Therefore, the sum of the intersections of a point inside the polygon is 0. Whereas, the intersections are not zero of a point lying outside the circle. Thus, lets take two points from the curve  $C$ , lets call the points  $P_1$  and  $P_2$ .  $P_1$  has to be greater in some context. For example, if we have a parameterized circle with

one variable  $t$ , and  $P_1$  is defined at  $t_1$  and  $P_2$  is defined at  $t_2$  then the algorithm requires that  $t_1 > t_2$ . Thus, the line segment created from the points  $P_1$  and  $P_2$  is  $\overline{P_1P_2}$ . So, lets say that the infinite ray  $\overrightarrow{Q}$  intersects the line segment  $\overline{P_1P_2}$ , then if that is the only intersection, the point  $Q$  is inside the curve  $C$ . However, if there exists a line segment,  $\overline{P_iP_j}$  where  $\overrightarrow{Q}$  intersects the curve  $C$  again, then the point lies outside of the  $C$ .

## 2.1 Ray Casting Implementation: Bounding-Box Testing

Ray casting is powerful for a polygon with a small amount of edges. A circle is a infinitely edged shape, when trying to parameterize a circle in a finite environment the ray casting algorithm has problems. Depending on the size of the finite domain, the ray casting algorithm's accuracy varies. A solution lies in a implementation [1] which takes the ray casting algorithm and applies a faster intersection index. This implementation differs by restricting the domain to a smaller size based on a user defined box. Now, the ray  $\overrightarrow{Q}$  will only check intersections within that bounding boxes. This implementation allows for us to be able to focus on a smaller area while increasing the points.

## References

- [1] A. Mathias, U. Kanther, and R. Heidger. Insideness and collision detection algorithms. In *2008 Tyrrhenian International Workshop on Digital Communications - Enhanced Surveillance of Aircraft and Vehicles*, pages 1–7, Sep. 2008.