Visibility Polygon from a Given Point

--Christine

--Implementation of the paper:
A Linear Algorithm for Computing the Visibility Polygon from a Point
H. EL GINDY AND D. AVIS

Content:

- **Description of the program**

  The program deals with the visibility problem of a SIMPLE polygon from a given point inside it. User can either enter the vertices of the polygon using the GUI in a clockwise (CW) order, or can open a ". vp" file whose first line is the number of points n, and whose next n lines are the x y coordinates for each point, vertices also in a CW order.

  If a given point is outside the polygon, the program can detect the error and report to the console window.

  Users can open and save the ".vp" polygon information file whenever they want, and once the program is started, it can be run on different polygon models repeatedly, and don't need to be shut down every time you run on a model.

- **Major difficulties encountered**

  1. Using homogeneous coordinates to compute the intersection point

  In the program, the coordinate system of the canvas is up side down to the normal coordinate system (See Figure. 1). So the homogeneous coordinate system will be <**-W**, X, Y> instead of <**W**, X, Y>, as a result, the sign of Jack's favorite determinant will be opposite to the origin.

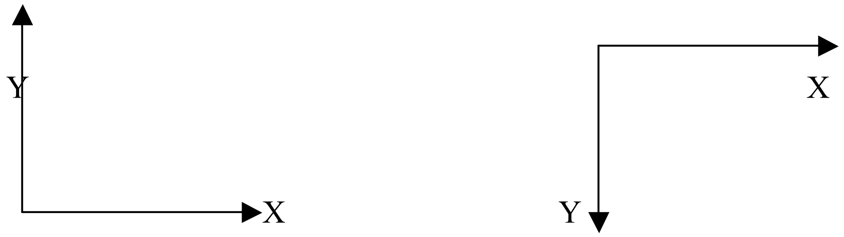


Figure 1.

$$\begin{vmatrix} 1 & px & py \\ 1 & qx & qy \\ 1 & rx & ry \end{vmatrix} \begin{matrix} & + & \text{right} & /CW \\ = & 0 & \text{r on pq} & \\ & - & \text{left} & /CCW \end{matrix}$$

Line equation through pq would be:

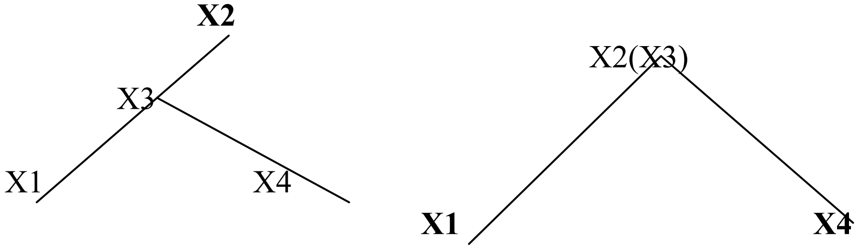
$$\begin{vmatrix} 1 & px & py \\ 1 & qx & qy \\ \mathbf{-W} & X & Y \end{vmatrix} = \begin{vmatrix} px & py \\ qx & qy \end{vmatrix} (\mathbf{-W}) - \begin{vmatrix} 1 & py \\ 1 & qy \end{vmatrix} X + \begin{vmatrix} 1 & px \\ 1 & py \end{vmatrix} Y$$

Intersect point on line l * m would be:

$$\begin{vmatrix} lw & lx & ly \\ mw & mx & my \\ \mathbf{-W} & X & Y \end{vmatrix} = \begin{vmatrix} lx & ly \\ mx & my \end{vmatrix} (\mathbf{-W}) - \begin{vmatrix} lw & ly \\ mw & my \end{vmatrix} X + \begin{vmatrix} lw & lx \\ mw & my \end{vmatrix} Y$$

2. Different degenerate cases

Decide whether two segments intersect.



If the following inequation returns true, then two segments intersect, otherwise, they don't.

(CCW(X1, Y1, X2, Y2, X3, Y3) * CCW(X1, Y1, X2, Y2, X4, Y4) <= 0)
&& (CCW(X3, Y3, X4, Y4, X1, Y1) * CCW(X3, Y3, X4, Y4, X2, Y2) <= 0))

➢ Known bugs

If users don't input the vertices in a CW order, or the polygon is not simple, the program will give wrong results or even might throw out some exceptions.

Due to some numerical rounding problem, (what exactly the rounding problem is and is there a way to fix it, I haven't figured it out yet.) the program might sometimes not be able to output the right results in some extreme situations. It will output and error message "Stack Empty" to the console window.

In the program, I assume no three vertices are collinear, if this condition is violated, the program sometimes can't give the right result.