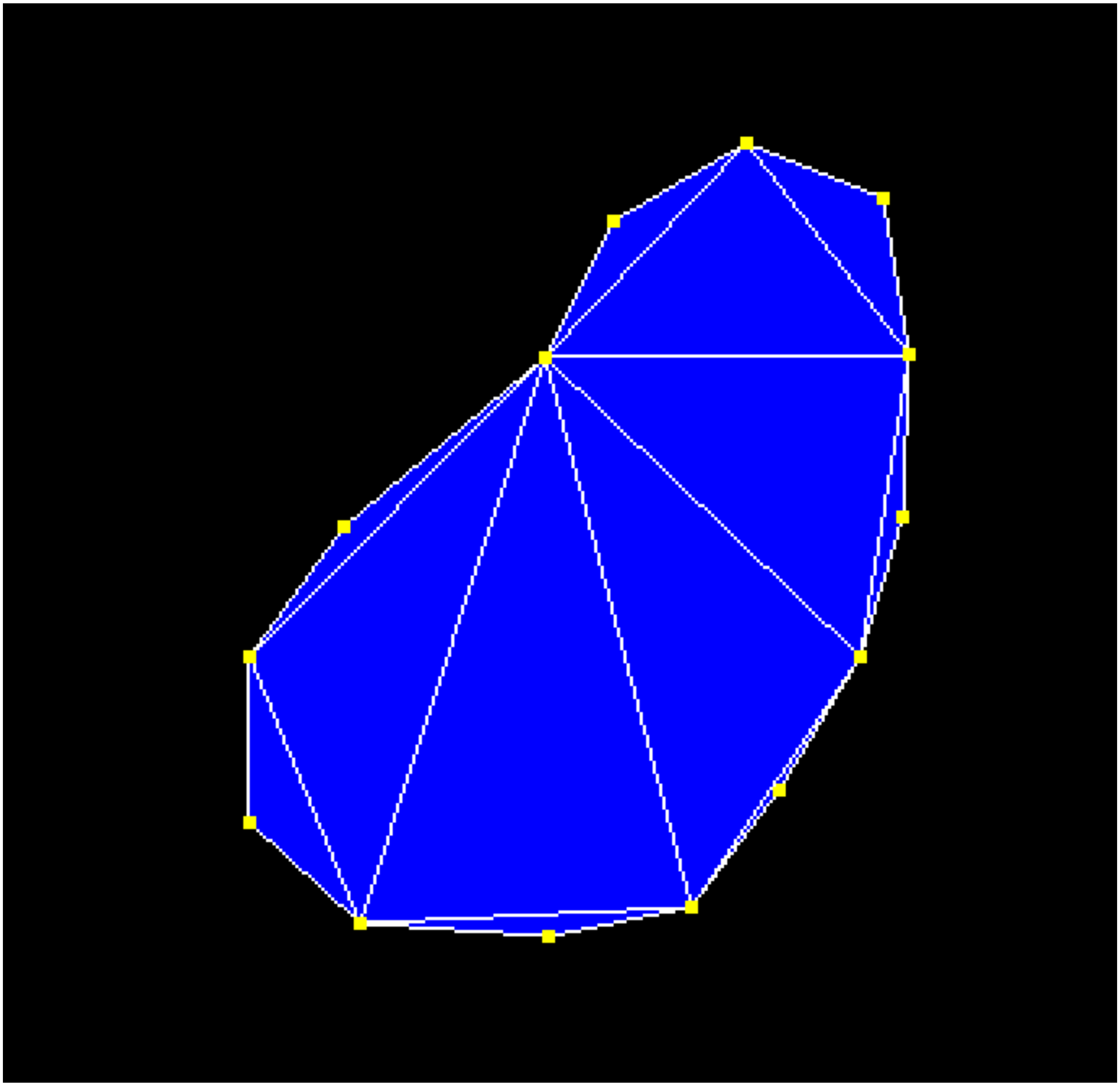
Triangulating Polygons

A common problem in computer graphics is to triangulate arbitrary polygon. In the case of polygons that do not self intersect, this can be achieved by the ear clipping algorithm. The algorithm has a runtime of O(n^2) but is relatively easy to implement and gives satisfactory triangle quality. The input is assumed to be an ordered set of points (CCW or CW) on the polygon.



An ear of a polygon is a triangle formed by three consecutive vertices for which no other vertex of the polygon is inside the triangle. The ear tip is the second of the three consecutive vertices. The Ear Clipping algorithm proceeds as follows:

1. For each point in the polygon, mark the point either as convex or concave based on the angle it forms with its immediate neighbors. Then, construct non-overlapping ears from the convex vertices of the polygon, i.e., identify the ear tips. Store the concave vertices and ear tips in separate linked lists (for efficient deletion later on).

2. While there are still ear tips in the linked list
    a. Remove the first ear tip from the linked list and construct a triangle from it and its neighboring vertices.
    b. Update the neighboring vertices. That is, if an adjacent vertex is an ear tip, check if it remains an ear tip by determining whether there is any concave vertex inside the triangle the ear tip forms with its adjacent vertices (Remember, the ear tip removed in step 2 a. is no longer a neighbor to this vertex). If an adjacent vertex is convex, then check if needs to be turned into an ear tip. Finally, if the adjacent vertex is concave, check if it is now convex and whether it needs to be turned into an ear tip.
    c. Delete the ear tip in 2 a. from the list of points that form the polygon.

Implementation: https://github.com/rohan-sawhney/triangulate-polygons