Hidden Lines Removal

Fake Hidden Line Removal (Using Painter’s algorithm)

We fill in the viewport with the background color. Then we start drawing the polygons in the order given by the Painter’s algorithm. In order to draw a polygon, we first do a fill of the entire polygon with the background color, and then we draw its outline. The filling will erase all earlier lines that lie under the polygon. The result of this filling looks like if only the edges of the polygon were drawn.

Remark: The sequence of filling first and drawing the outline second is very important to prevent the polygon filling from erasing the outline of the polygon.

Note: This algorithm works by filling a number of polygons one after another. The idea is “paint over”. This approach could be difficult on some devices, so we explain another approach to hidden line removal.

Hidden Line Removal

For this approach, no convexity is needed, and each element is drawn after a complete test of visibility has been performed. No “painting over” is attempted.

We want to draw the visible edges of a mesh object. A direct approach is to clip each edge of the object against all faces that occlude it.

Now we take into consideration the following facts:
1) If an edge is occluded by a back face, it must be occluded by a front page. So we don’t need to test each edge against all faces, but just test the remaining edges against front faces only.
2) An edge belongs to two faces. So if an edge belongs to two back faces, we don’t need to consider it.

Therefore after we remove the back faces and the corresponding back edges, we must clip every edge against the front faces. If an edge is divided into several segments, we must keep them, and test them against all remaining surfaces (faces)

Practical remarks:
 a) Put prominent faces (large and close to view plane) in the top of the faces list.
 b) Use a stack of edges
 c) Pop an edge and clip it against every front page, except the two defining this edge, and if it survives, draw it. If it broken into pieces, test one piece against the remaining faces, and push the other pieces into the stack. Keep popping pieces from the stack until all pieces are taken care of.
 d) When an edge has been worked out, start the cycle again with the next edge on the stack.
How to clip an edge against front faces?

We follow the same ideas found in the Painter’s algorithm, knowing that if a test does not decide, we go to the next one.

a) No x-overlap. Edge is visible
b) No y-overlap. Edge is visible
c) Edge is on the “inside” of the face. Edge is visible. Use the criteria $Ap_x + Bp_y + Cp_z + D > 0$, for both endpoints of the edge.
d) Edge penetrates the plane. Split the edge into two parts. The inside part is visible for that face, and goes into the “survivors” list. Further tests are required for the “outside” part.
e) If an edge is here, it should overlap with the surface and it should lie on the “outside” of it. If we arrive here with broken pieces, check for overlap on x and y before attempting this check, since the overlap was made with the original whole segment.

Compute the intersection of the edge with each edge of the face on the xy-plane (we are assuming that we have an orthographic projection, after pre-warping). Find the points of intersection, in terms of the parameter $t$, using the formula in 2D $p = \overline{a}(1-t) + \overline{b}t$, where $\overline{a}$ and $\overline{b}$ are the endpoints of the edge (or piece of edge) under consideration. Order the values of the parameter for the intersections, and analyze the list to determine where the visible portions for that face are situated. These are the survivors that are visible for that face. Note that the value of $t$ for survivors should lie between 0 and 1.

Example: Given the next figure,

Find the pieces that are visible for the face whose projection is the non-convex polygon.

We assume that the list of $t$-values is given by 0.6, 0.4, 0.1, -0.3, -0.8, 1.4, as we find the intersection of the line going through $\overline{a}$ a $\overline{b}$ . We order the list, -0.8, -0.3, 0.1, 0.4, 0.6,
1.4. We know that the first encounter is always from the outside, the second from the inside, etc. So we can write that

Inside pieces: -0.8 to -0.3, 0.1 to 0.4, 06 to 1.4
Outside pieces: -∞ to -0.8, -0.3 to 0.1, 0.4 to 0.6, 1.4 to ∞.

Looking at the outside pieces, we must restrict ourselves to values of t between 0 and 1. So we determine that we have two pieces:
Piece # 1: Given by the equation \( \bar{p} = \bar{a}(1-t) + \bar{b}t \), \( 0 \leq t \leq 0.1 \).
Piece # 2: Given by the equation \( \bar{p} = \bar{a}(1-t) + \bar{b}t \), \( 0.4 \leq t \leq 0.6 \).