Computer Graphics

(Graphics Primitives – Clipping)

Lecture 004

Clipping in 2D

- Usually only a specified region of a picture needs to be displayed. This region is called the clip window.
- An algorithm which displays only those primitives (or part of the primitives) which lie inside the clip window is referred to as a clipping algorithm.
- This lecture will describe a few clipping algorithms for particular primitives (line, polygon)

Clipping Algorithm (Cohen-Sutherland)

- Clipping a straight line against a rectangular clip window results in either.
  - A line segment whose endpoints may be different from the originals, or
  - Not displaying any part of the line. This occurs if the line is completely outside the clip window.
- The Cohen-Sutherland line clipping algorithm considers each point at a time and truncates the line with the clip window's edges until the line can be trivially accepted or trivially rejected.
- A line is trivially rejected if both endpoints lie on the same outside half-plane of some clipping edge.

Clipping Edges

- The *xy* plane is partitioned into nine segments by extending the clip windows' edges.
- Each segment is assigned a 4-bit code according to where it lies with respect to the clip window.
- For example bit 2 is set if and only if the region lies to the south of (below) the clip window.

Bit	Side	Inequality
1	N	Y > Ymax
2	S	Y < Ymin
3	E	X > Xmax
4	W	X < Xmin

The Algorithm (Single Line Clipping)

- The Cohen-Sutherland algorithm starts by assigning an outcode to the two line endpoints (call them c1 and c2)
- If both outcodes are 0 (c1 or c2 = 0) the line lies entirely in the clip windows and is thus trivially accepted.
- If the two outcodes have at least one bit in common (c1 and c2 /= 0), then they lie on the same side of the windows and the line is trivially rejected.
- If a line cannot be accepted or rejected, it is then truncated by an intersecting clip edge and the previous steps are repeated.

The Algorithm (Polygon Clipping)

- Clearly, a polygon can be clipped against a rectangular clip window by considering each clip edge at a time.
- For filled polygons, a new polygon boundary must be computed.
- At each clipping stage, a new polygon is created by removing the outside vertices and inserting the intersections with the clip boundary.

Clipping against an edge

- Given the polygon [v_1; v_2; … ; v_n], we can create a new polygon [w_1; w_2; … w_m] by considering the vertices in sequence and deciding which ones (and any intersections with the edge) have to be inserted into the clipped polygon vertex list.

Clipping against an edge (algorithm)

- Suppose that v_i has been processed in the previous step. There are four possible cases which need to be considered while processing the next vertex v_{i+1}
  - If v_i is outside the window and v_{i+1} is inside, then the intersection of the line $v_i \rightarrow v_{i+1}$ with the clipping edge, and the vertex v_{i+1} have to be inserted into the new polygon list.
  - If v_i and v_{i+1} are both outside the clip window then no point need to be inserted.
  - If both v_i and v_{i+1} are inside the windows, v_{i+1} is inserted into the list, and finally
  - If v_i is inside the window and v_{i+1} is outside, only the intersection of the line $v_i \rightarrow v_{i+1}$ with the clip boundary is inserted.

Problems with the algorithm

- The algorithm does not work well on certain concave polygons. In such cases the polygon must be split into two convex ones.