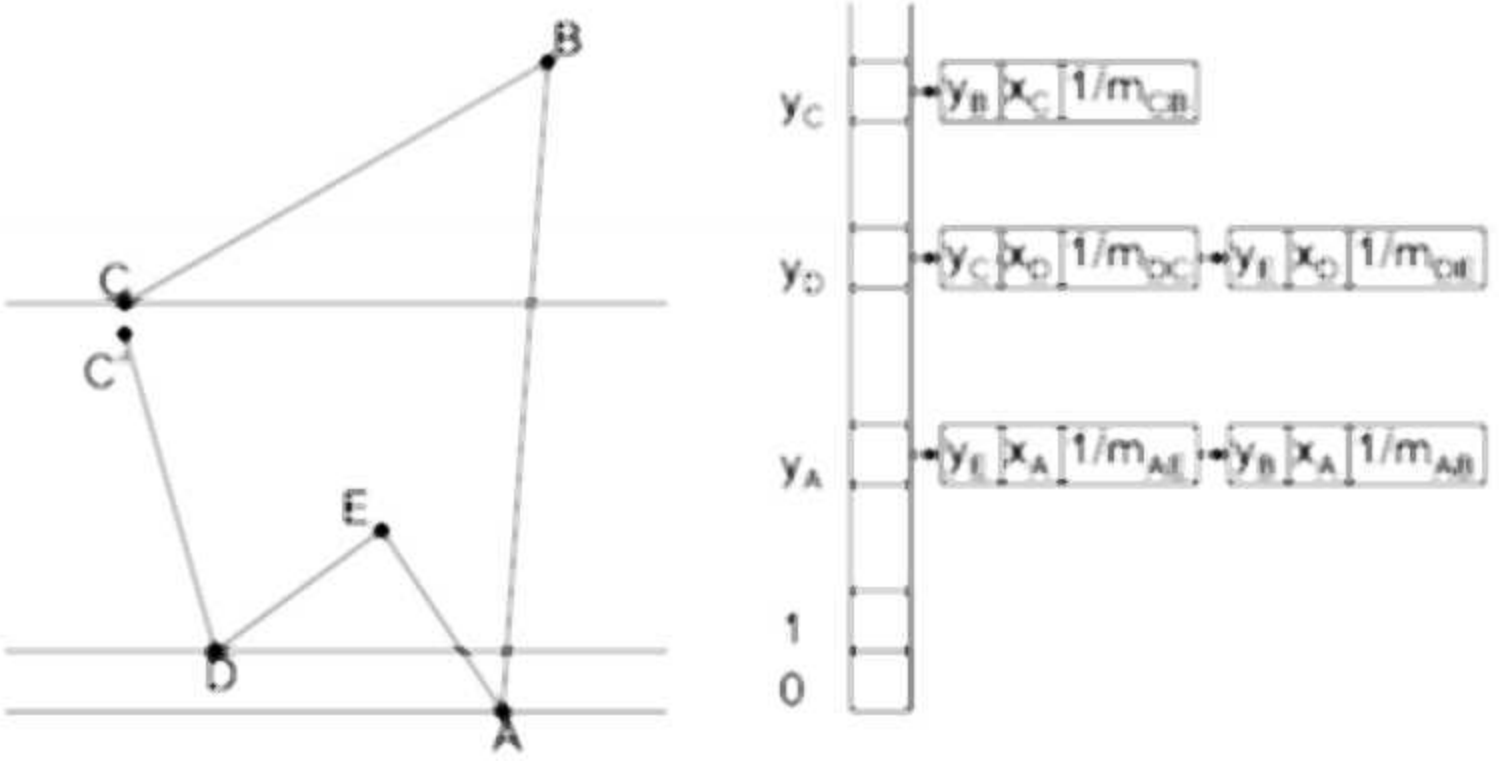
Scan-line algorithm

- Sample case: CD shortened to C'D
- Sorted edge table -> Active edge list



Scan-line algorithm

- Usual trick to avoid floating point arithmetic:

  Instead of decision parameter

  $k/m \geq 0.5$

  we use

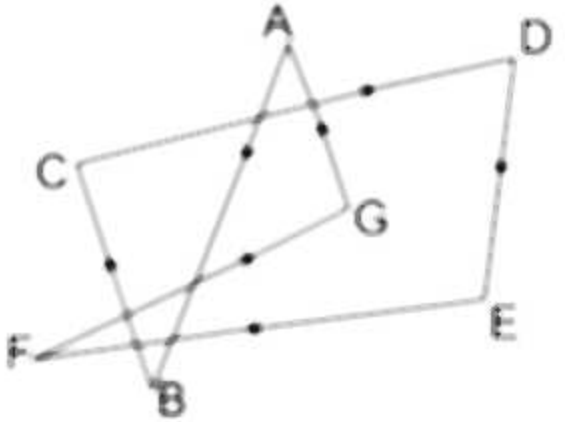
  $2\Delta X*(k/m - 0.5)$ integer!

Boundary Fill

- Basic idea:
  - Given a closed boundary of an area. The boundary is marked in some well defined way.
  - Find an interior point of the area.
  - Spread out in an orderly fashion and fill the area inside the boundary.
- Tricks:
  - Draw the boundary using a unique colour
  - Fill the interior
  - Repaint the boundary!

Inside-Outside Tests

- Typically a polygon is defined by non-intersecting edges.
- For these polygons the interior is unambiguous.
- For free hand drawings the situation can be much more complicated:



Inside/Outside test: Odd-Even Rule

- The odd-even rule classifies each point not on one of the edges as either interior or exterior.
- From any point shoot a straight line to infinity, not passing any vertices of the polygon
- Calculate the number of edges crossed, N .
- If N is odd then the point is interior.
- If N is even then the point is exterior.
- The scan-line algorithm works according to this rule.

Inside/Outside test:
Topological winding number

- Used in topology: which objects can be continuously transformed into each other?
- Counts the number of times the object winds around a given point in the counter-clockwise direction
- From the given point again draw a straight line to infinity. Initialize winding number w = 0.
- For each edge crossing the line from right to left add one: w -> w + 1
- For each edge crossing the line from left to right subtract 1: w -> w - 1
- If w ≠ 0 then the point is an interior point, otherwise it is exterior.

Inside/Outside test: Topological winding number

- May give results different from the odd-even rule!
- A practical way of calculating the winding number w is to examine the sign of the z-component of the vector product between a vector ***u*** along the line and the edge $E_{AB} = V_B - V_A$:

$$\text{sign}(\boldsymbol{u} \times E_{AB})$$

I/O test: Topological winding number

- The same result can be obtained by forming the vector u' which is perpendicular to the vector $u = (u_x, u_y)$ along the line, crossing the line in the right to left direction: $u' = (-u_y, u_x)$
- Now calculate the dot product of E_{AB} and u' and note the sign of the result:

$$\text{sign}\left(E_{AB} \cdot u' \right) = -E_{ABx}u_y + E_{ABy}u_x$$

Boundary fill

- Suppose the boundary is a uniquely defined pixel quality.
- A boundary fill algorithm starts from an interior point and expands until we reach the boundary.
- The problem is to make sure that we really cover all of the interior!

Boundary fill

- Use stencils to test for additional interior points.
- 4-connected stencil

- 8-connected stencil