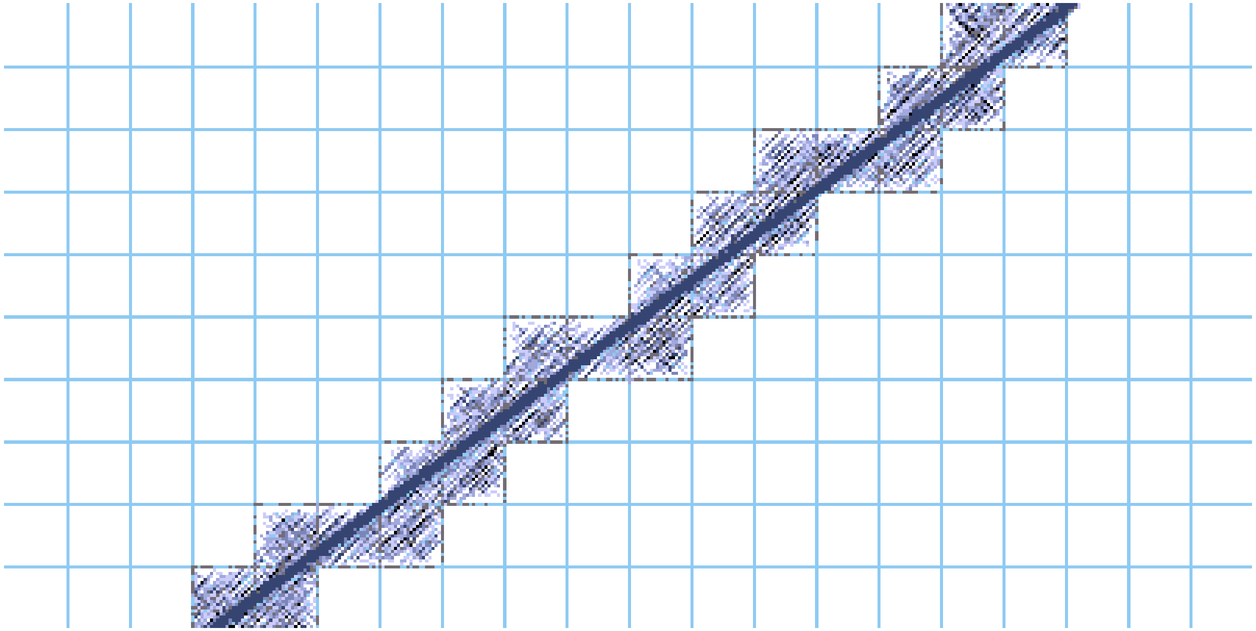
Rasterization

Turning ideal geometry into pixels

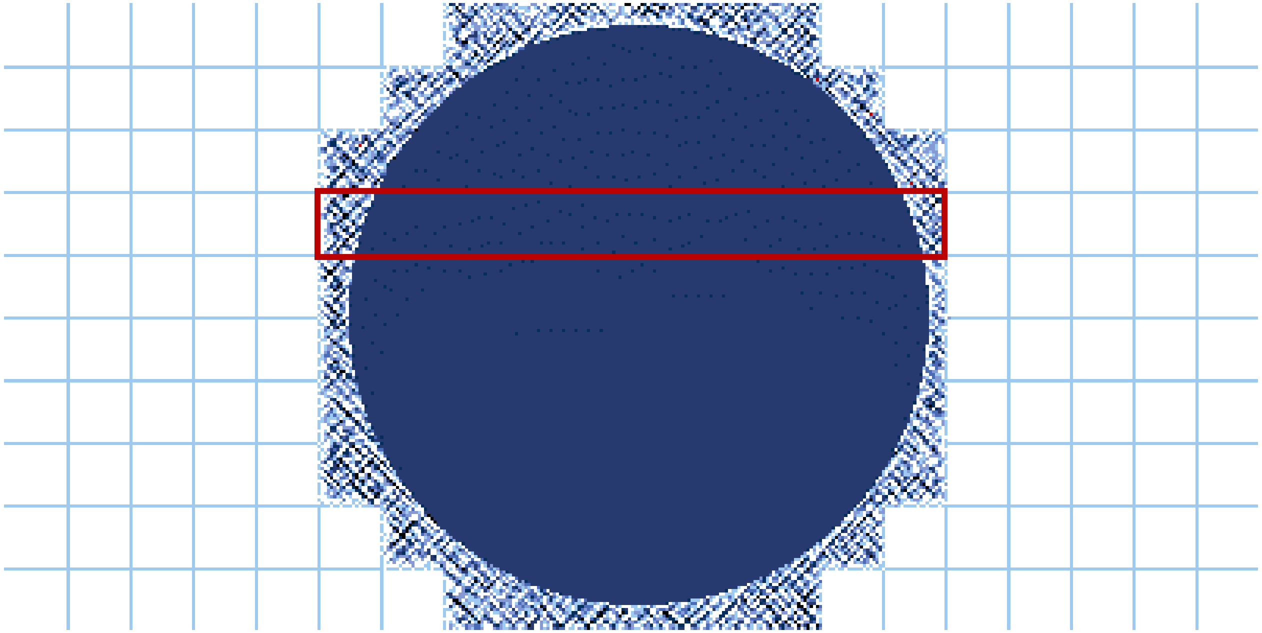
Background



- Mathemathically speaking, lines, curves, triangles etc. have infinite precision.
- In practise, screen consists of pixels.
- Conversion of ideal primitives into pixels is called rasterization, or scan conversion.

Scan Converting a Filld Circle

- Find width of the circle, in pixels, for each scanline.
- Draw the span.

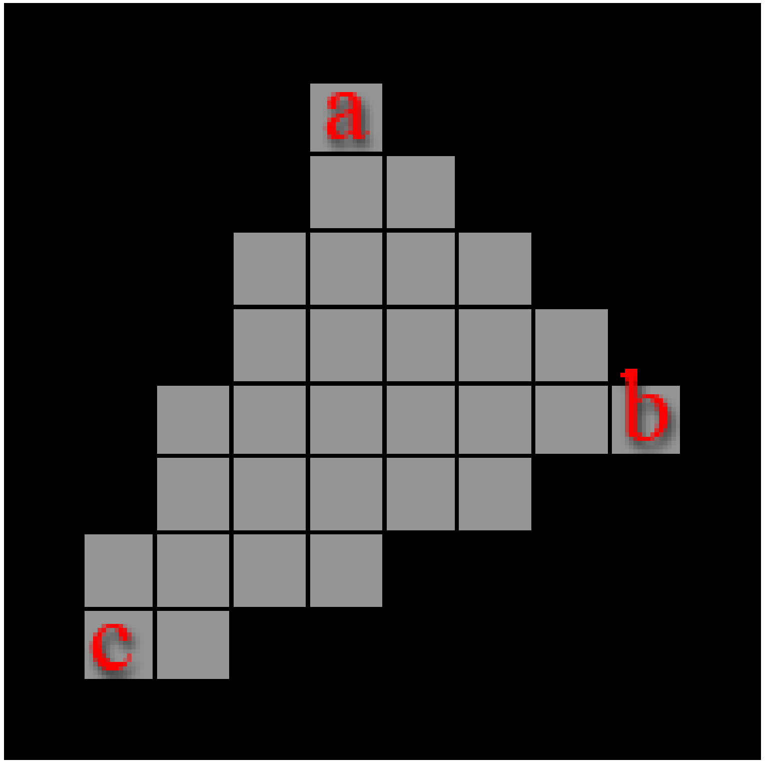


SCAN CONVERTING A LINE

- Find major axis.
  - Ystep = y_Δ / x_Δ
  - Xstep = x_Δ / y_Δ
  - If Xstep > Ystep, Y is the major, X otherwise.
- Loop through the major axis, plotting pixels.
  - For (x = x1, y=y0; x < x2; x++, y+=Ystep) …
- Special cases needed where x_Δ and/or y_Δ are zero.
- For more advanced line-drawing, look up Bressenham.

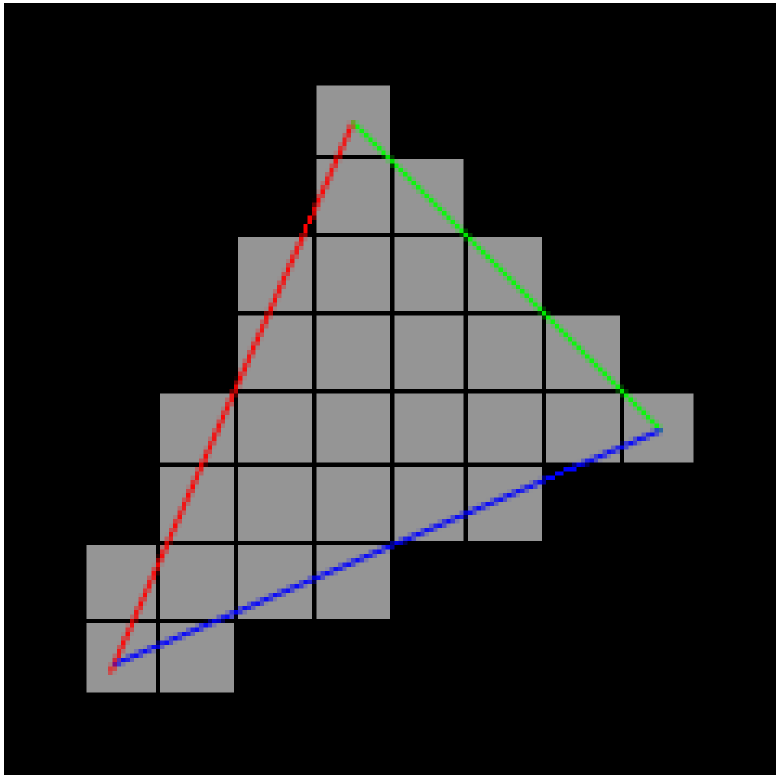
Scan Converting a Triangle

- Step 1: find Y order of vertices



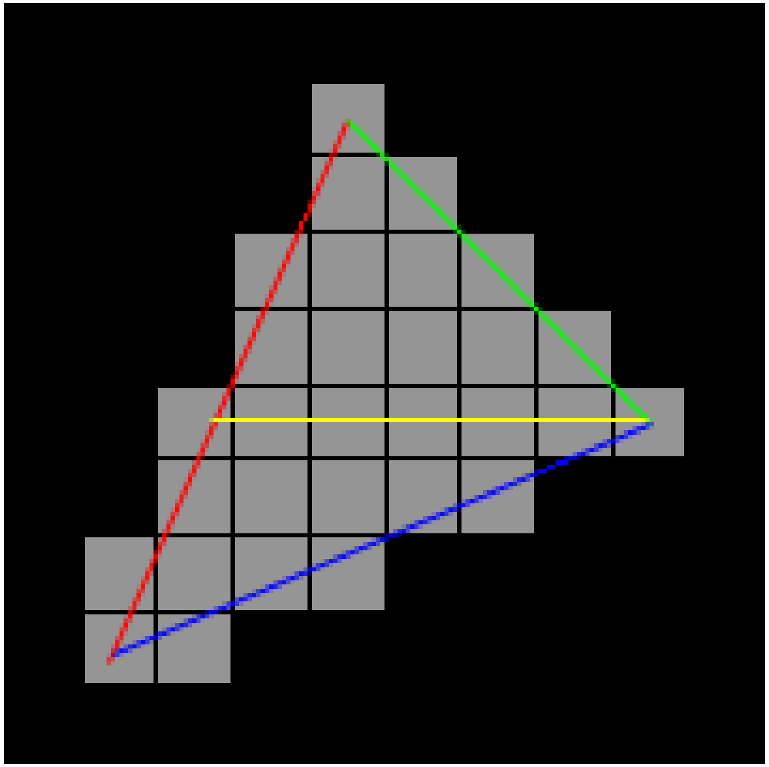
Scan converting a triangle

- Step 2: Trace left and right edges (a->c and a->b)



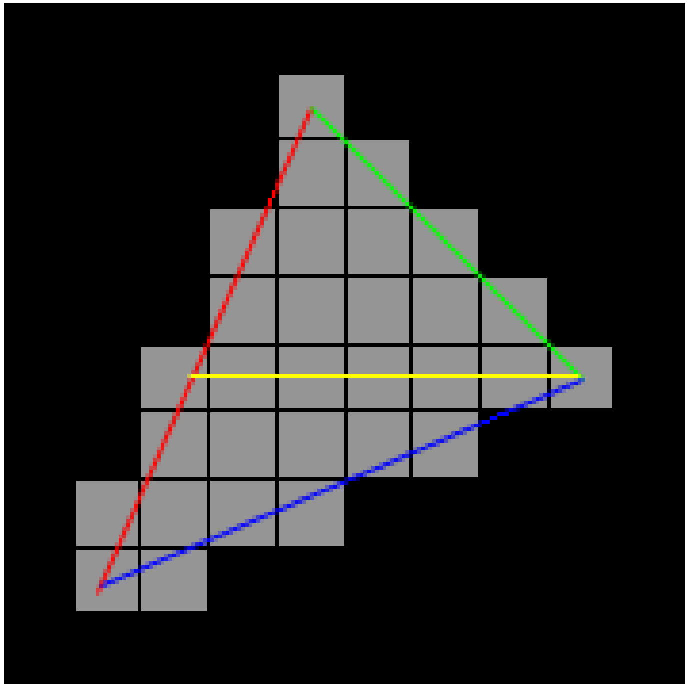
Scan Converting a Triangle

- Step 3: When vertex B is reached, replace a->b slope with b->c slope
- Continue tracing.



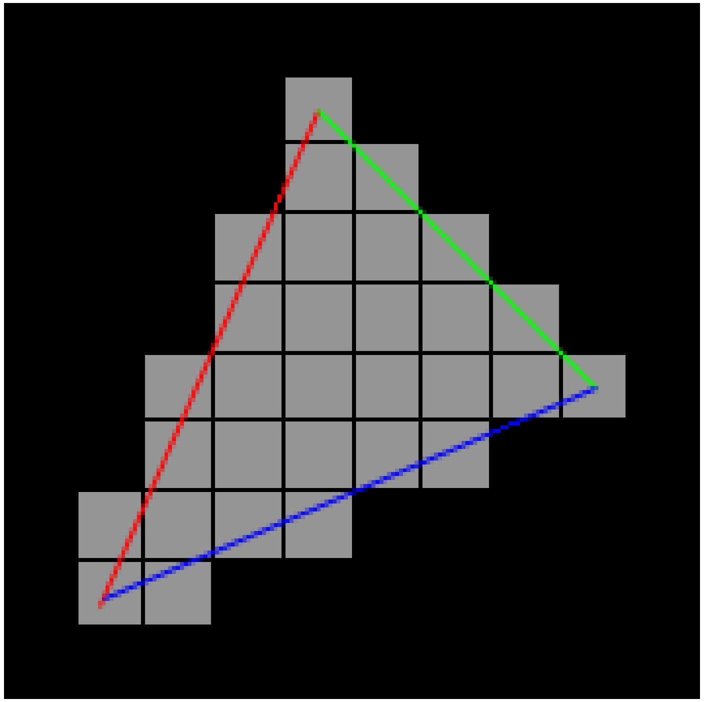
Scan Converting a Triangle

- Special cases:
  - A.y == C.y
    - Zero-area triangle. Skip rendering.
  - B.y == C.y
    - Bottom-flat triangle; we're done when we reach B.y
  - A.y == B.y
    - Top-flat triangle; skip the top part



Scan Converting a Triangle

- Alternate method:
  - Have two arrays for left/right edges.
  - Draw all triangle edges to these arrays.
  - In second pass, draw spans from left to right data.



Scan Converting a Triangle

- Other possible interpolation values along with vertex coordinates:
  - Color (Gouraud shading)
  - Normal (Per-pixel lighting)
  - Texture (UV) coordinates