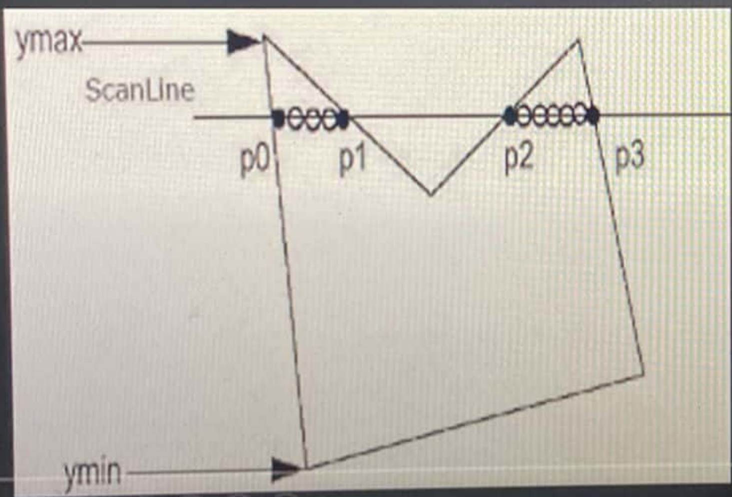
Scanline Polygon filling Algorithm

Scanline filling is basically the filling up of polygons using horizontal lines or scanlines.

This algorithm works by intersecting scanlines with polygon edges and fills the polygon between pairs of intersections.



NAMITHA RAMACHANDRAN

Step 1 – Find out the Ymin and Ymax from the given polygon.

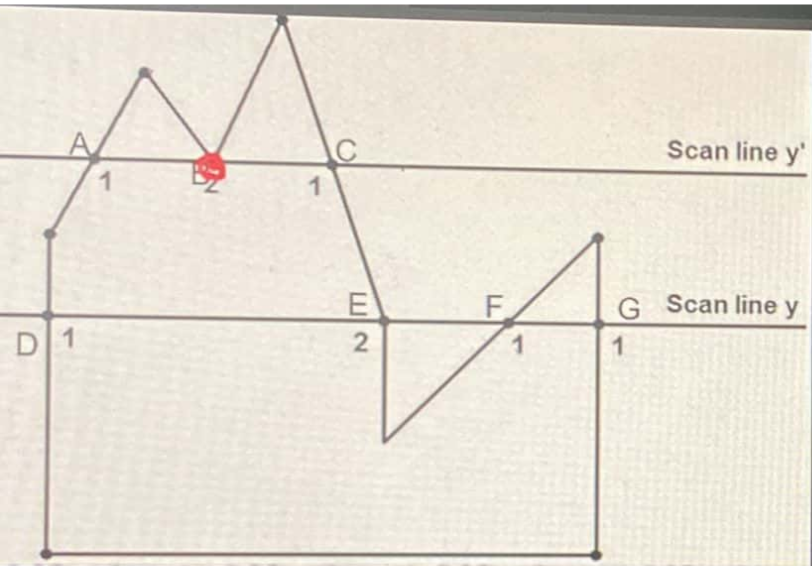
Step 2 – ScanLine intersects with each edge of the polygon from Ymin to Ymax. Name each intersection point of the polygon.

Example : consider previous figure ,intersections are named as p0,p1,p2,p3

Step 3 – Sort the intersection point in the increasing order of X coordinate i.e. (p0,p1) ,(p1,p2) ,(p2,p3).

Step 4 – Fill all those pair of coordinates that are inside polygons and ignore the alternate pairs.

{ ignore (p1,p2) }



Scan line Y'

Intersection points :

A,B,C ,Where B is a vertex and other ends of the 2 lines causes vertex B are on the same side of scan line, so take B two times .{A,B,B,C}

Pairs (A,B) ,(B,C)

Scan line Y

Intersection points :

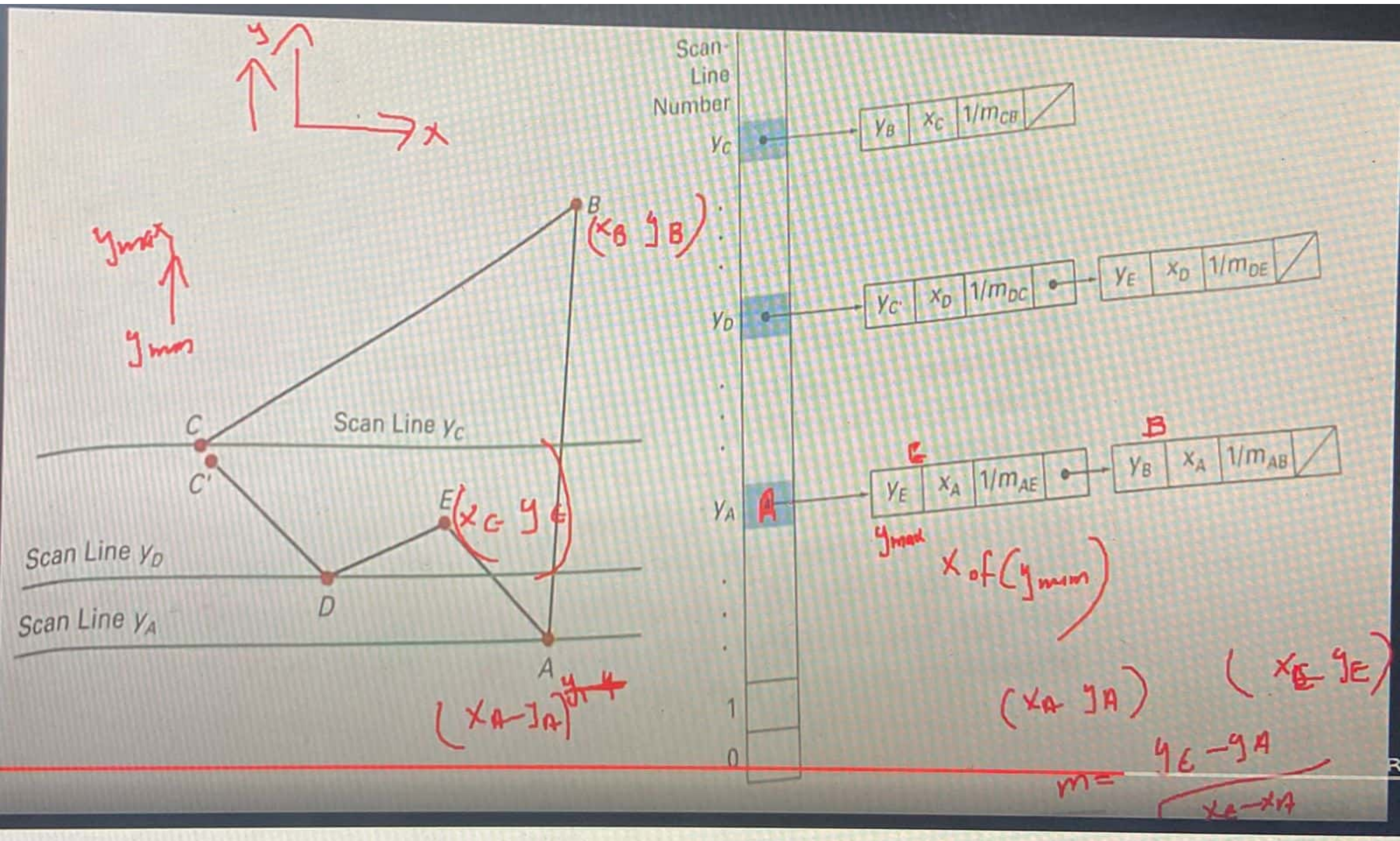
D,E,F,G , Where E is a vertex and other ends of the 2 lines causes vertex E are on opposite side of scan line, so take E once . {D,E,F,G}

Pairs (D,E) ,(F,G)

pecial cases of polygon vertices:

both lines intersecting at the vertex are on the same side of the scanline, consider it as two points.

lines intersecting at the vertex are at opposite sides of the scanline, consider it as only one point.



Scan-Line Number
y_C
y_D
y_A
1
0
y_B | x_C | $1/m_{CB}$
$y_{C'}$ | x_D | $1/m_{DC}$
y_E | x_D | $1/m_{DE}$
y_E | x_A | $1/m_{AE}$
y_B | x_A | $1/m_{AB}$
A
B
C
C'
D
E
Scan Line y_C
Scan Line y_D
Scan Line y_A
y
x
y_{max}
y_{min}
$(x_B\ y_B)$
$(x_A - y_A)$
$(x_A\ y_A)$
$(x_E\ y_E)$
y_{max}
x of (y_{min})
$m = \frac{y_E - y_A}{x_E - x_A}$

To effectively perform a polygon fill, we can store the polygon boundary in a **sorted edge table** that contains all the information necessary to process the scan lines efficiently, Each entry in the table for a particular scan line contains the maximum y value for that edge, the x-intercept value (at the lower vertex) for the edge, and the inverse slope of the edge.

For each scan line , edges are in sorted order from left to right.

y_{upper}

y_{lower}

Next process the scan lines from the bottom of the polygon to its top, and produce an active edge list for each scan line crossing the polygon boundaries.

The active edge list for a scan line contains all edges crossed by that scan line, with iterative coherence calculations used to obtain the edge intersections.

Coherence Properties

Coherence is simply that the properties of one part of a scene are related in some way to other parts of the scene so that the relationship can be used to reduce processing.

Slop of this polygon boundary can be expressed in terms of the scan line intersection coordinates :

$$m = \frac{(yk+1-yk)}{(xk+1-xk)}$$

Where , $yk+1-yk=1$

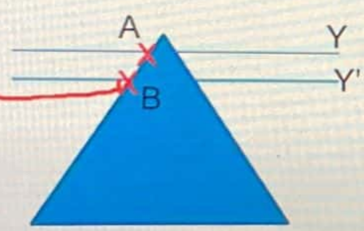
$m = 1/(xk+1-xk)$

$(xk+1-xk)=1/m$

$Xk+1= 1/m + xk$

Adjacent scanlines are separated by a distance of 1, which means y difference is unity.
y-y'=1

Using the properties of Coherence, after completing one scan line processing, we don't want to start from the beginning of the next scan line, instead, we can directly calculate the edge intersection point on the next scan line and proceed from that point .



$A(x_k, y_k)$

$B(x_{k-1}, y_{k-1})$

$B(x_{k-1}, y_k - 1)$

$m = \frac{y_k - (y_k - 1)}{x_k - (x_{k-1})}$ $\quad m = \frac{1}{x_k - (x_{k-1})}$

In the given diagram, after processing scan line "y" start with "y' ", but here we don't want to start processing from the beginning of "y' ", instead we can directly find coordinate of intersection "B" using coherence property and start from that point of scan line "y' " ,this will reduce complexity of algorithm .

$x_k - (x_{k-1}) = 1/m$

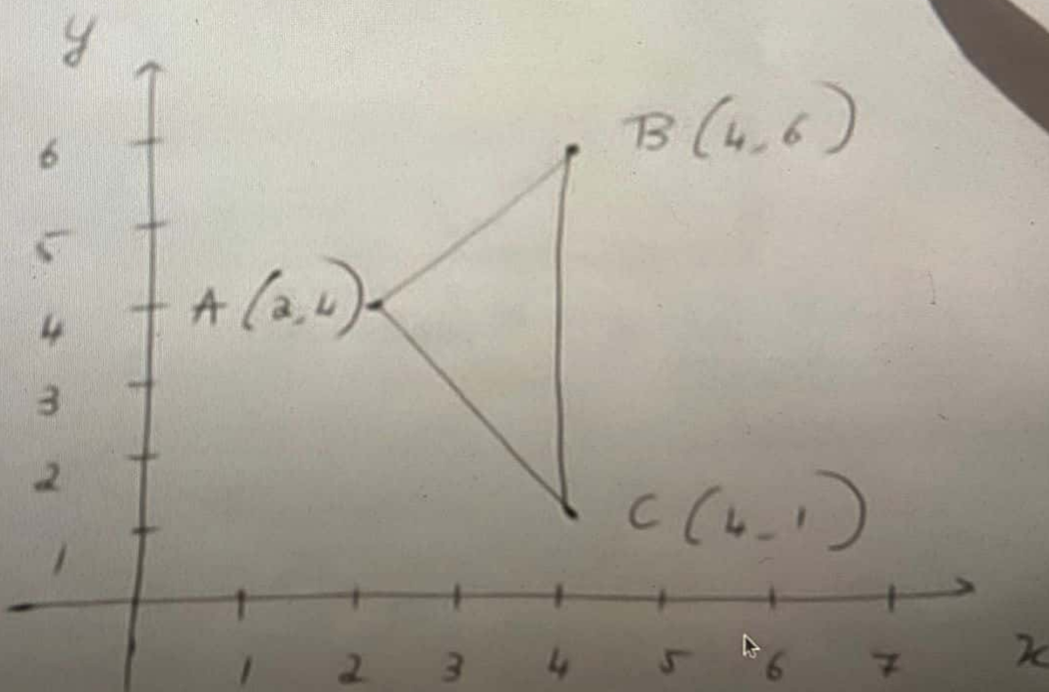
NAMITHA RAMACHANDRAN

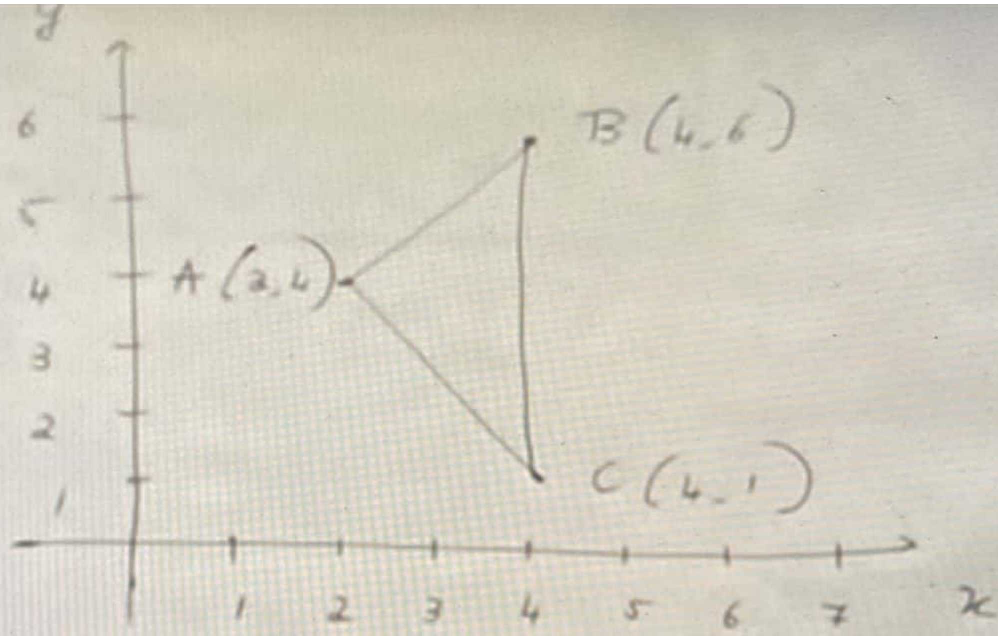
$x_{k-1} = x_k - 1/$

algorithm

(1) Explain scan line Polygon fill algorithm. Determine the content of the active Edge table to fill the polygon with vertices A (2, 4) B (4, 6) and C (4, 1) for $y = 1$ to $y = 6$

Sol^n





Sort the edges from $y_{min} \rightarrow y_{max}$

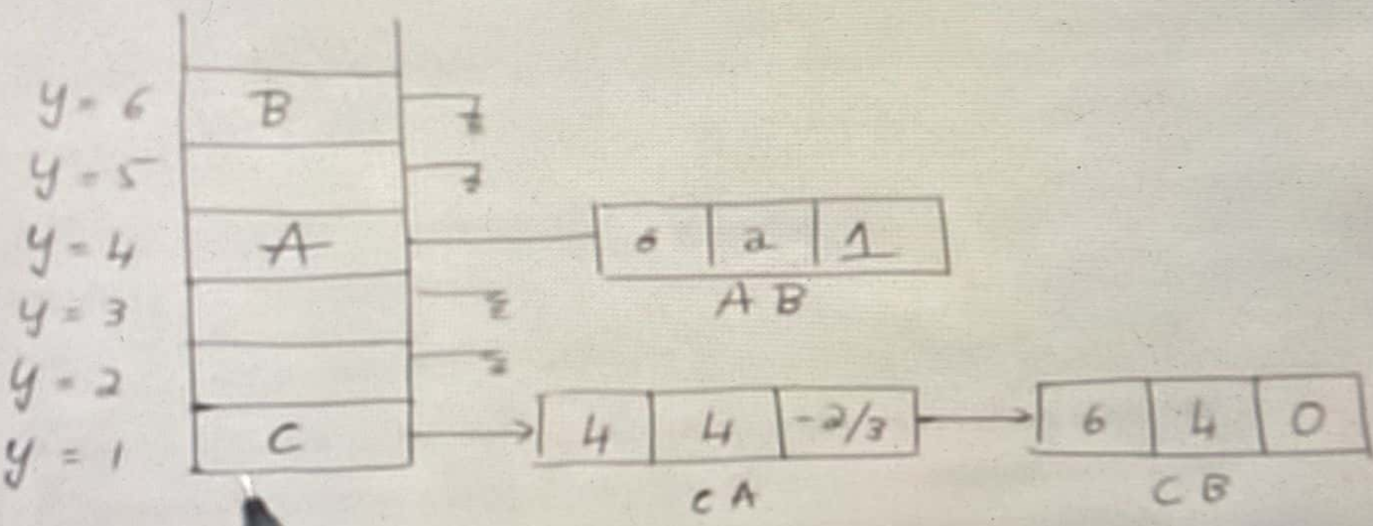
A (2,4) B (4,6) C (4,1)

Sort $y_{min} \rightarrow y_{max}$

C (4,1) A (2,4) B (4,6)

Create a Global Edge table (GET) - 7

It is an array of linked list.



(1) CA = $\overset{x_1\ y_1}{(4, 1)}$ and A $\overset{x_2\ y_2}{(2, 4)}$

ine Polygon Fill Algorithm

CA CB

(I) CA = C $\overset{x_1\ y_1}{(4,1)}$ and A $\overset{x_2\ y_2}{(2,4)}$

y_{max}	$x\,y_{min}$	$1/m$
4	4	$-2/3$

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{4-1}{2-4} = \frac{3}{-2}\ //$$

(II) CB = C $\overset{x_1\ y_1}{(4,1)}$ and B $\overset{x_2\ y_2}{(4,6)}$

y_{max}	$x\,y_{min}$	$1/m$
6	4	0

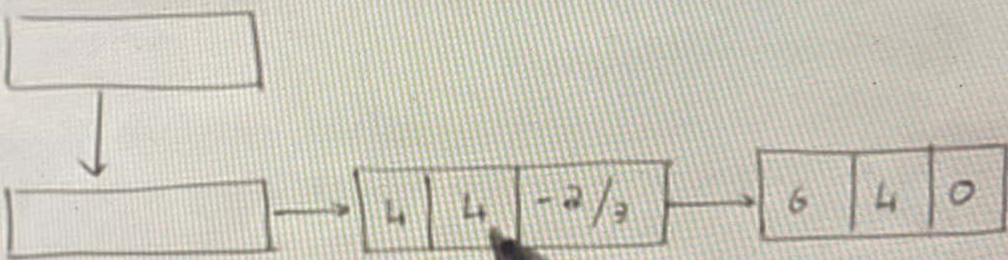
$$m = \frac{y_2 - y_1}{\ } = \frac{6-1}{4-4} = 5/0$$

Scanned with CamScanner

oblem 1 on Scan line Polygon Fill Algorithm

Create an Active Edge table (ACT) - (8)

Begins with null node & ends with null node.



$\underline{i = 1}$

$\{(4, 1)\ (4, 1)\}$ For pair

Draw a line and colo

{ (4 , 1) (4 , 1) } First pair
Draw a line and color

$\underline{\underline{i = 2}}$ [ ] → [4 | 4 | -2/3] → [6 | 4 | 0]

Increment with the slope

$$4 - \frac{2}{3} = \frac{12 - 2}{3} = \frac{10}{3}$$

$\underline{\underline{i = 2}}$ [ ] → [4 | 10/3 | -2/3] → [6 | 4 | 0]

{ (3.3 , 2) (4 , 2) } Second pair
Draw a line and color

$i = 2$ $\quad \boxed{} \rightarrow \boxed{4 \mid 10/3 \mid -2/3} \rightarrow \boxed{6 \mid 4 \mid 0}$

$\{(3.3, 2)\ (4, 2)\}$ Second pair

Draw a line and color

$j = 3$ $\quad \boxed{} \rightarrow \boxed{4 \mid 10/3 \mid -2/3} \rightarrow \boxed{6 \mid 4 \mid 0}$

Increment with the slope

$$\frac{10}{3} - \frac{2}{3} = \frac{8}{3}$$

$i = 3$ $\quad \boxed{} \rightarrow \boxed{4 \mid 8/3 \mid -2/3} \rightarrow \boxed{6 \mid 4 \mid 0}$

$\{(2.6, 3)\ (4, 3)\}$ Third

$i = 4$ [ ] → [4 | 8/3 | -2/3] → [6 | 4 | 0]

Add the node

$i = 4$ [ ] → [6 | 2 | 1] → [4 | 8/3 | -2/3] → [6 | 4 | 0]

With the 3 nodes we cannot make a pair so delete the node whose 1st value of i value is same

$i = 4$ [ ] → [6 | 2 | 1] → [6 | 4 | 0]

{ (2, 4), (4, 4) } Fourth pair

nodes we cannot make a pair so delete the node whose first value of i value is same

$J = 4$

[ ] → [6 | 2 | 1] → [6 | 4 | 0]

{ (2, 4), (4, 4) } Fourth pair

$J = 5$

[ ] → [6 | 2 | 1] → [6 | 4 | 0]

Increment with slope

$J = 5$

[ ] → [6 | 3 | 1] → [6 | 4 | 0]

{ (3, 5), (4, 5) } Fifth pair

...on Scan line Polygon Fill Algorithm

y = 5

[ ] → | 6 | 3 | 1 | → | 6 | 4 | 0 |

{ (3, 5), (4, 5) } Fifth pair

y = 6

[ ] → | 6 | 3 | 1 | → | 6 | 4 | 0 |

Increment with slope

y = 6

[ ] → | 6 | 4 | 1 | → | 6 | 4 | 0 |

{ (4, 6), (4, 6) } Sixth pair

Scan line Polygon Fill Algorithm

First pair (4, 1) (4, 1)
Second pair (3.3, 2) (4, 2)
Third pair (2.6, 3) (4, 3)
Fourth pair (2, 4) (4, 4)
Fifth pair (3, 5) (4, 5)
Sixth pair (4, 6) (4, 6)