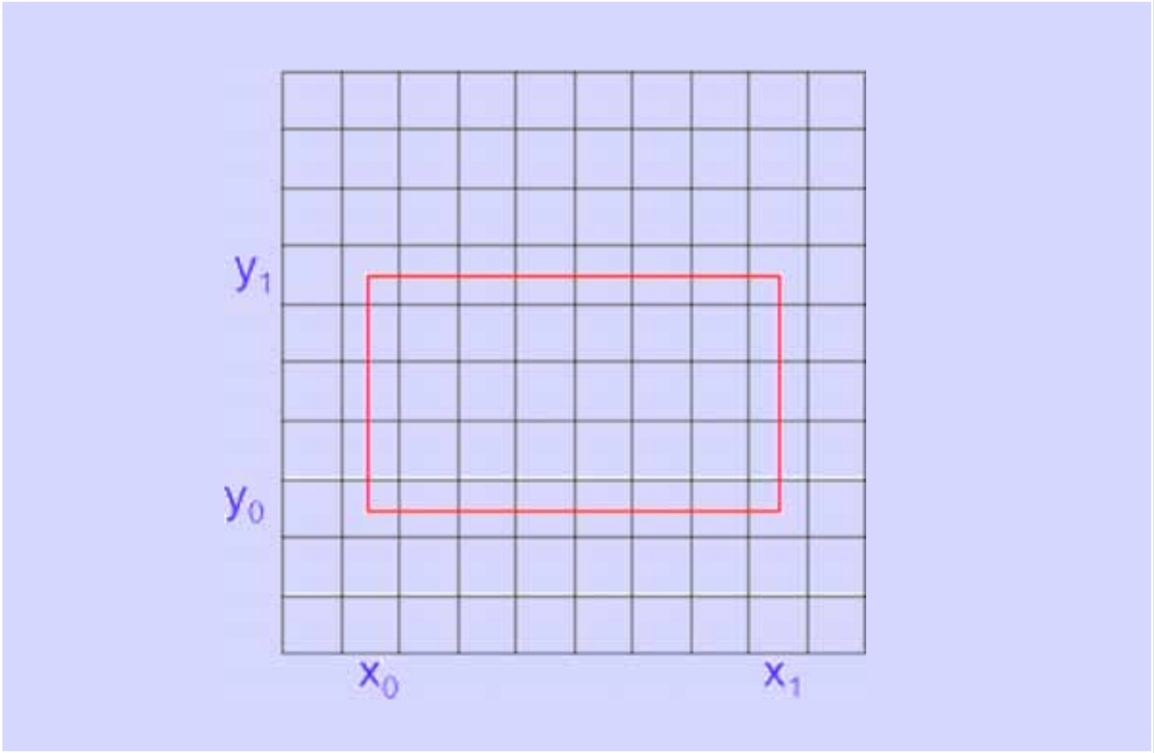
CMSC 435
Introductory Computer Graphics
Rasterization

Penny Rheingans
UMBC

Scan conversion

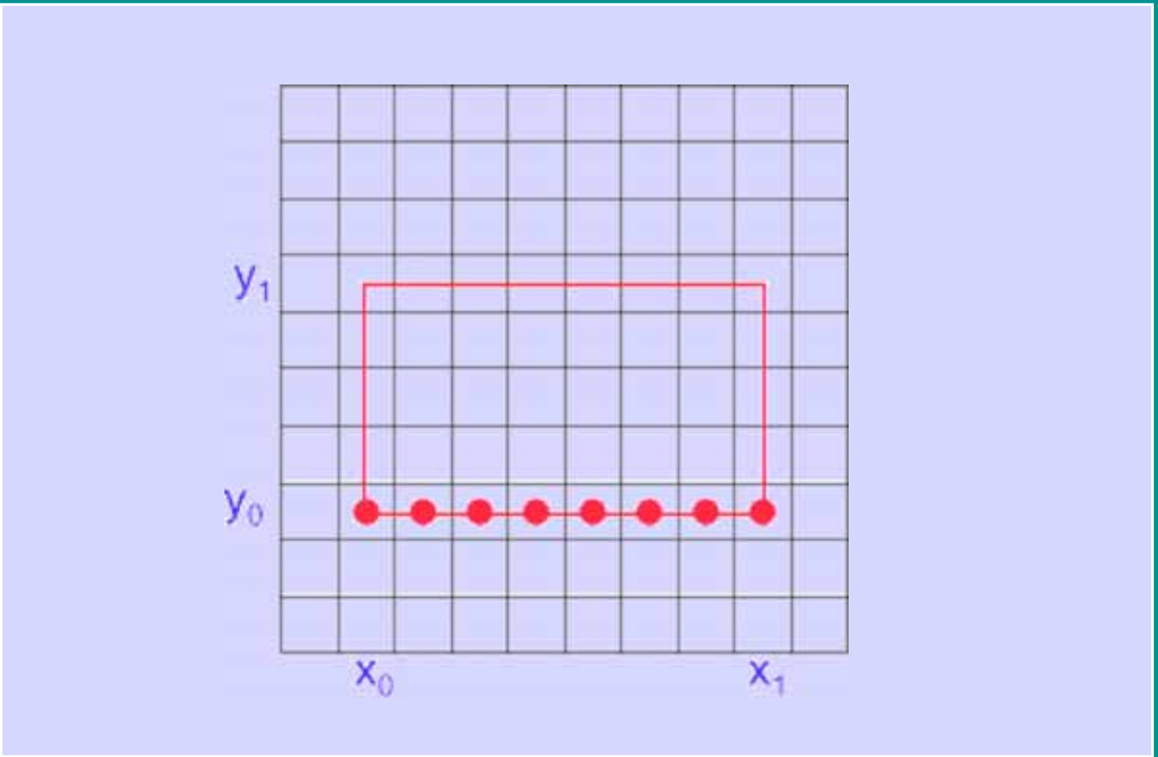
- Problem
  - How to generate filled polygons (by determining which pixel positions are inside the polygon)
  - Conversion from continuous to discrete domain
- Concepts
  - Spatial coherence
  - Span coherence
  - Edge coherence

Scanning Rectangles



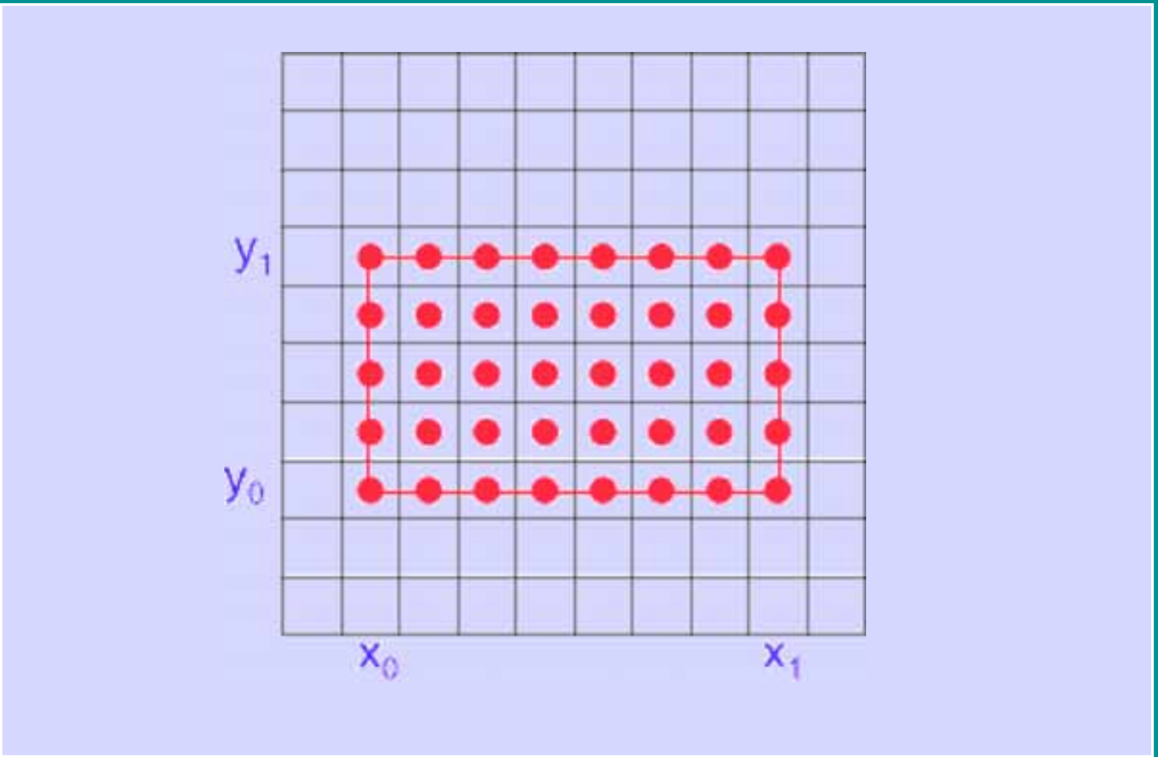

```
for ( y from y0 to yn )
    for ( x from x0 to xn )
        Write Pixel (x, y, val)
```

Scanning Rectangles (2)




```
for ( y from y0 to yn )
    for ( x from x0 to xn )
        Write Pixel (x, y, val)
```

Scanning Rectangles (3)

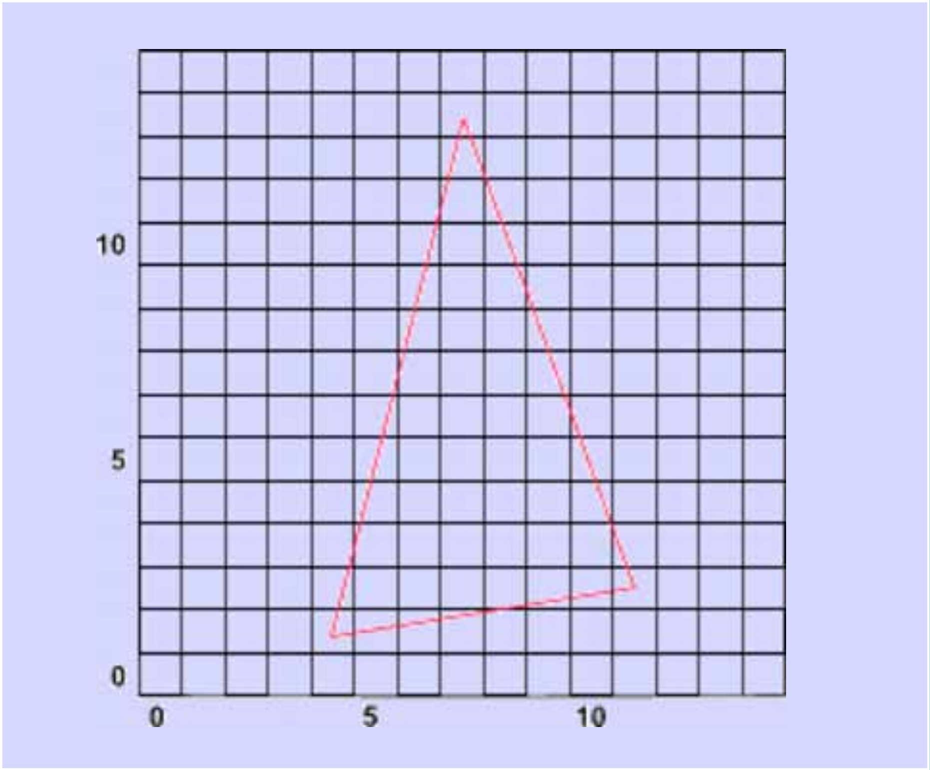



```
for ( y from y0 to yn )
    for ( x from x0 to xn )
        Write Pixel (x, y, val)
```

Scanning Arbitrary Polygons

- vertices:

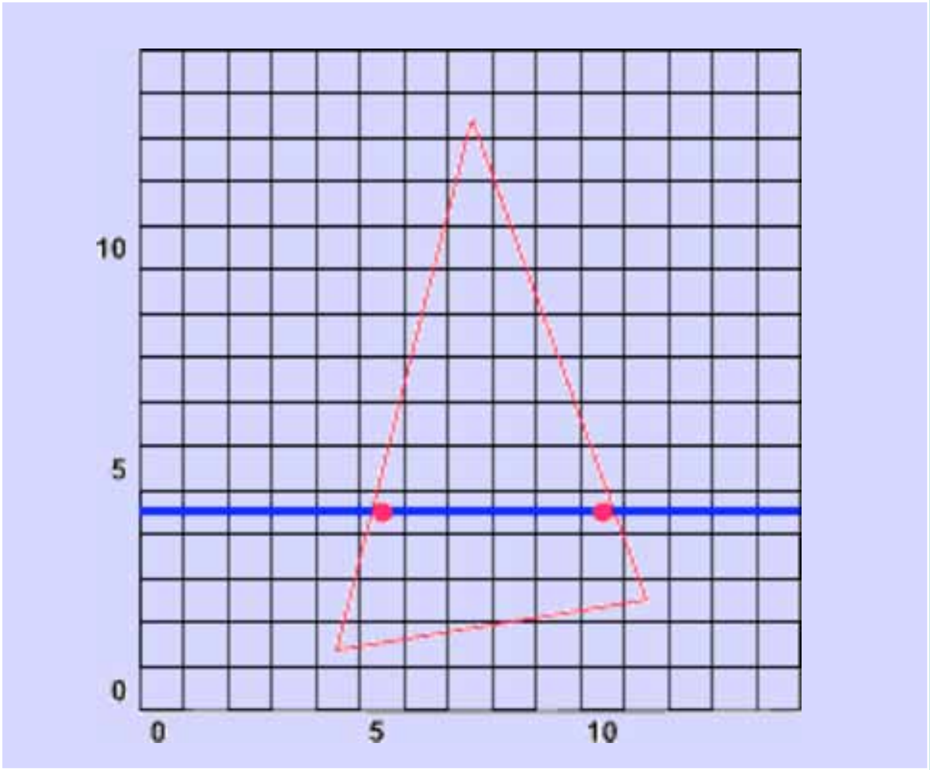
  (4, 1) , (7, 13) , (11 , 2)



Scanning Arbitrary Polygons (2)

- vertices:

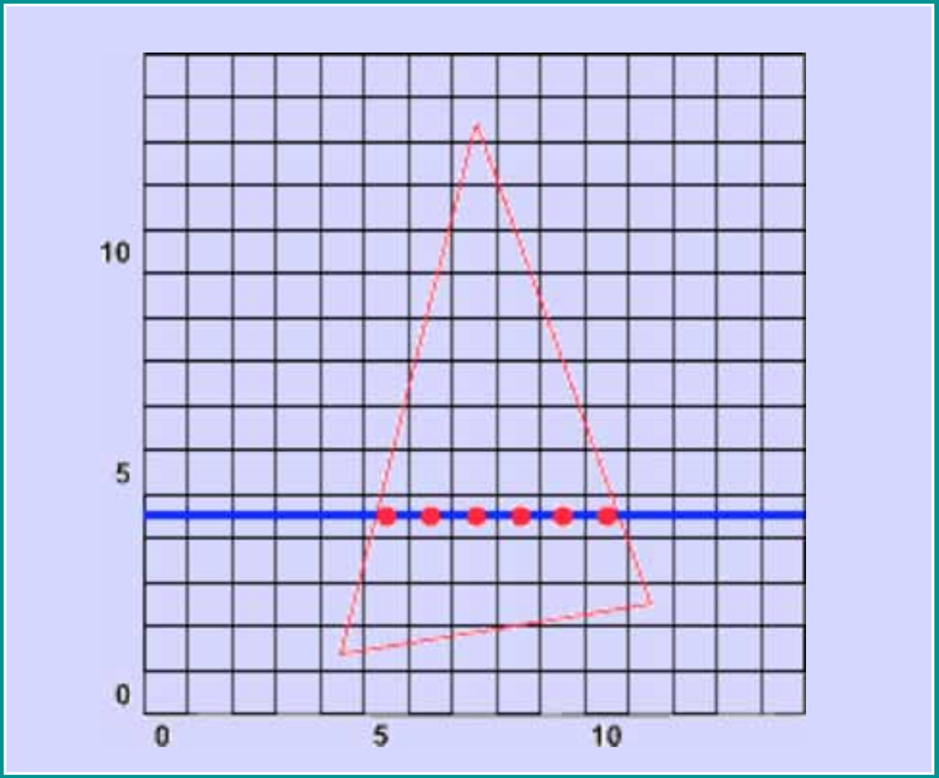
  (4, 1) , (7, 13) , (11 , 2)



- Intersect scanline w/pgon edges => span extrema

Scanning Arbitrary Polygons (3)

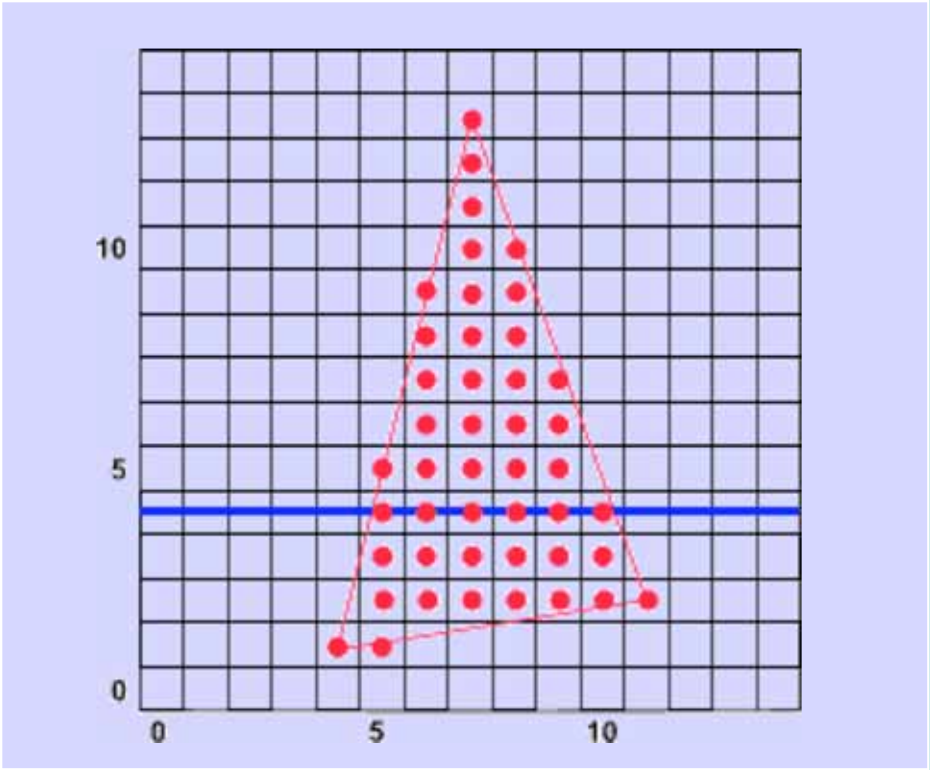
- vertices:
  (4, 1) , (7, 13) , (11 , 2)



- Intersect scanline w/pgon edges => span extrema
- Fill between pairs of span extrema

Scanning Arbitrary Polygons (4)

- vertices:
  (4, 1) , (7, 13) , (11 , 2)

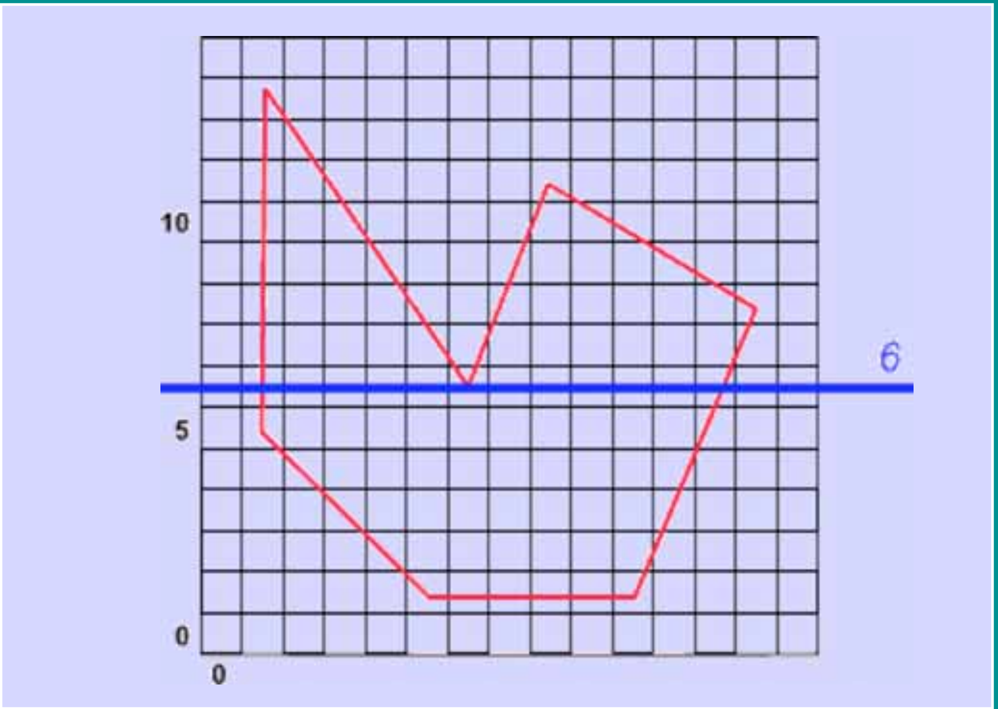


For each nonempty scanline

Intersect scanline w/pgon edges => span extrema

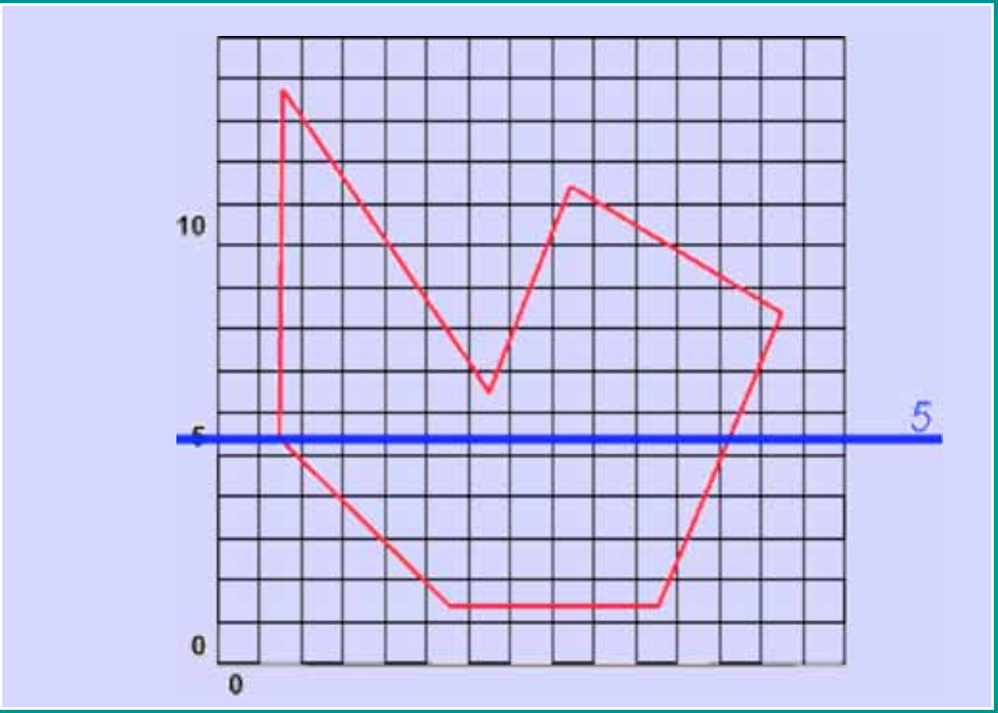
Fill between pairs of span extrema

Example Cases (2)



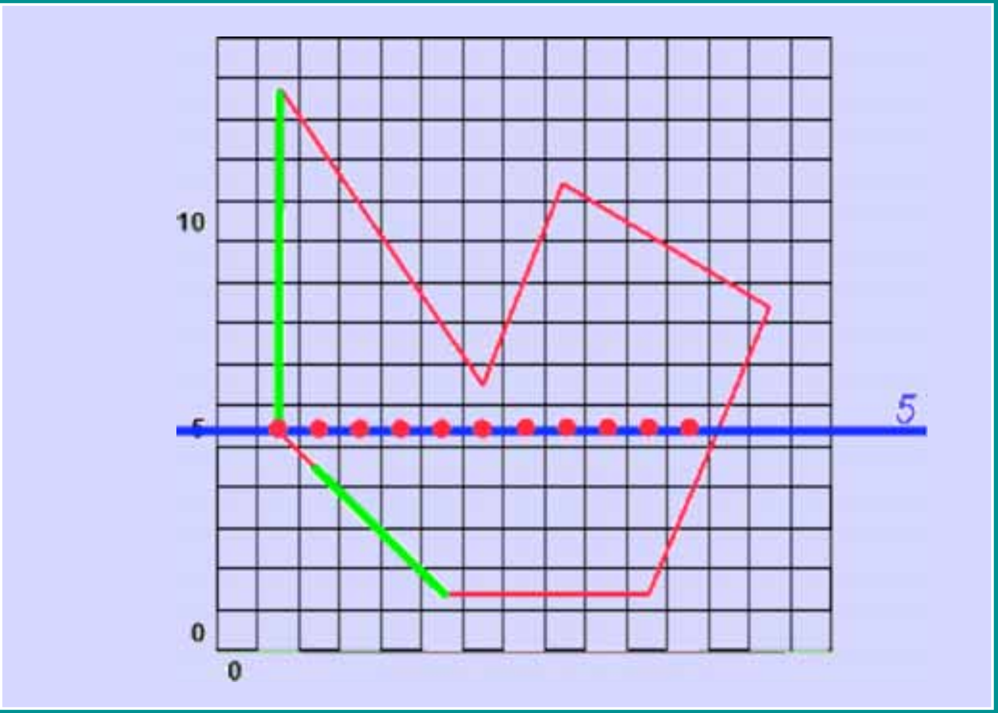
4 intersections w/ scanline 6 at x = 1, 6, 6, 12 1/7

Example Cases (3)



- 3 intersections w/scanline 5 at x = 1, 1, 11 5/7

Example Cases (4)



3 intersections w/scanline 5 at x = 1, 1, 11 5/7

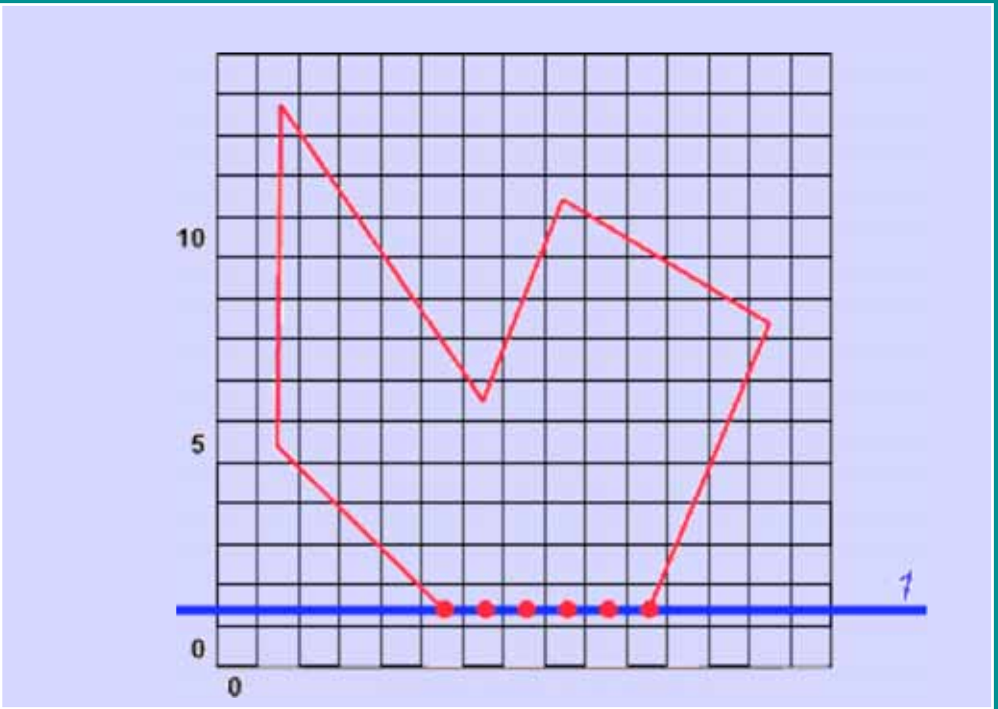
==>

Count continuing edges once (shorten lower edge) now x=1, 11 5/7

Example Cases (5)

4 intersections w/ scanline 1at x = 5, 5, 10, 10

Example Cases (6)



4 intersections w/ scanline 1 at x = 5, 5, 10, 10

=>

Don't count vertices of horizontal edges.

Now x = 5, 10

Scanline Data Structures

Sorted edge table:

all edges

sorted by min y

holds:

max y

init x

inverse slope

Active edge table:

edges intersecting current scanline

holds:

max y

current x

inverse slope

Scanline Algorithm

1. Bucket sort edges into sorted edge table
2. Initialize y & active edge table

    y = first non- empty scanline

    AET = SET [y]

3. Repeat until AET and SET are empty

    Fill pixels between pairs of x intercepts in AET

    Remove exhausted edges

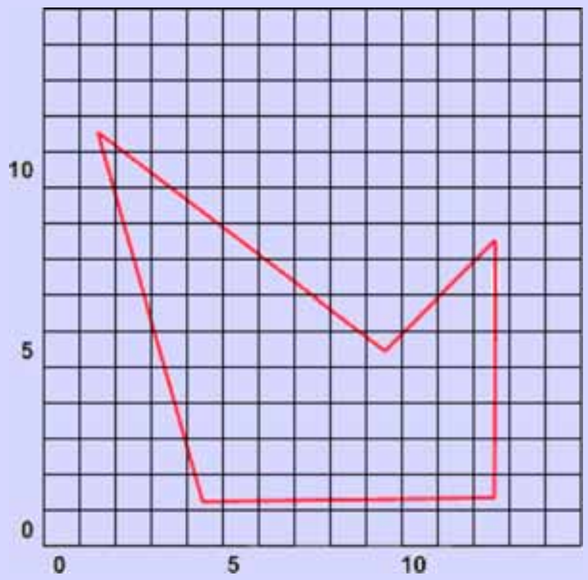
    Y++

    Update x intercepts

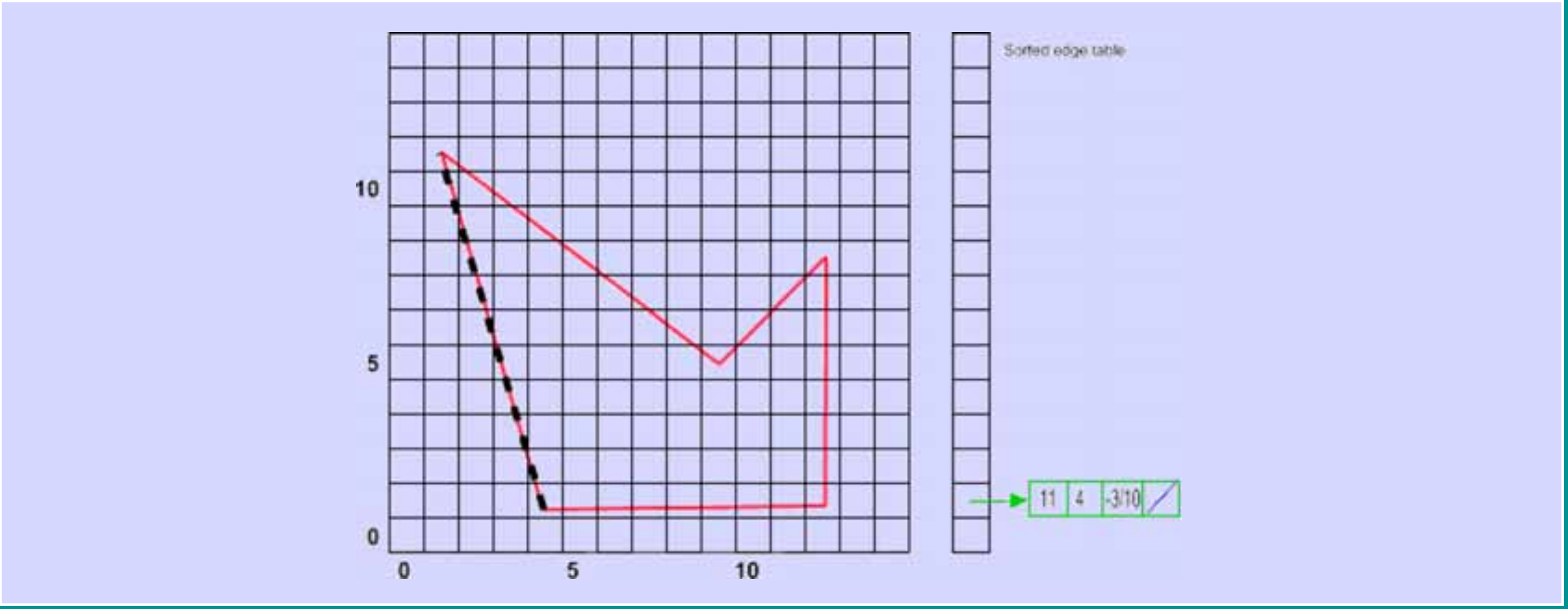
    Resort table (AET)

    Add entering edges

Example: vertices (4,1), (1,11), (9,5), (12,8), (12,1)



Example: vertices (4,1), (1,11), (9,5), (12,8), (12,1)



bucket sort edges into sorted edge table

sort on minY: 1

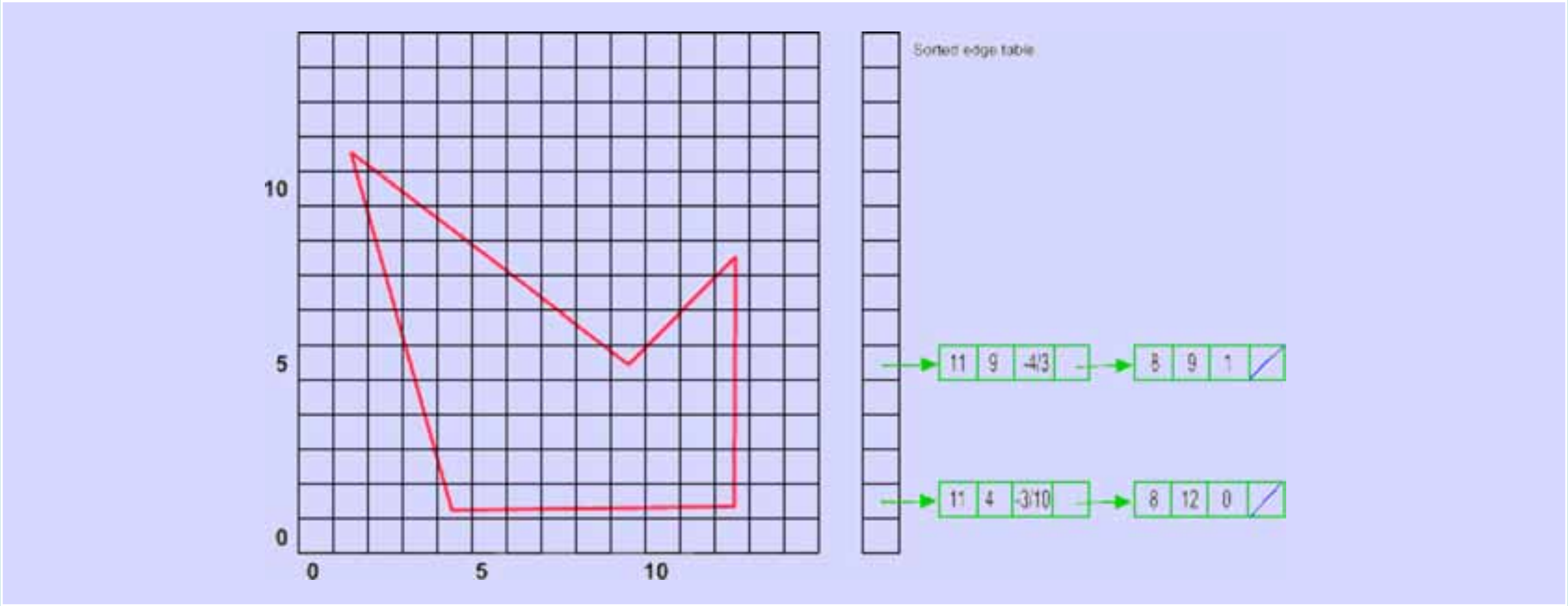
store:

  max Y: 11

  min X: 4

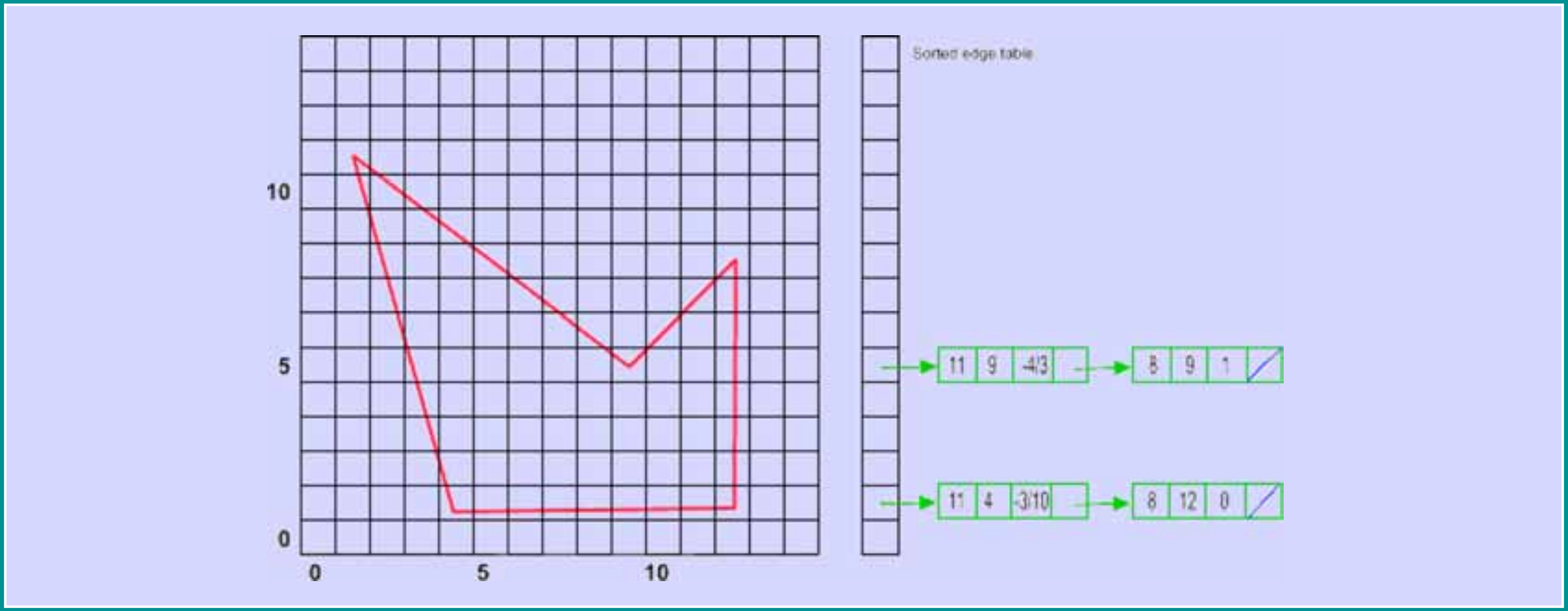
  1/m : (Xmax - Xmin) / (Ymax - Ymin) = (1 - 4) / (11 - 1) = -3 / 10

Example: vertices (4,1), (1,11), (9,5), (12,8), (12,1)



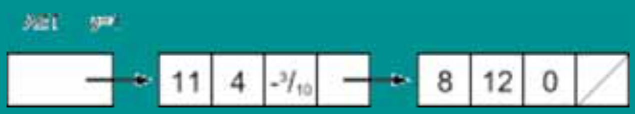
bucket sort edges into sorted edge table

Example: vertices (4,1), (1,11), (9,5), (12,8), (12,1)

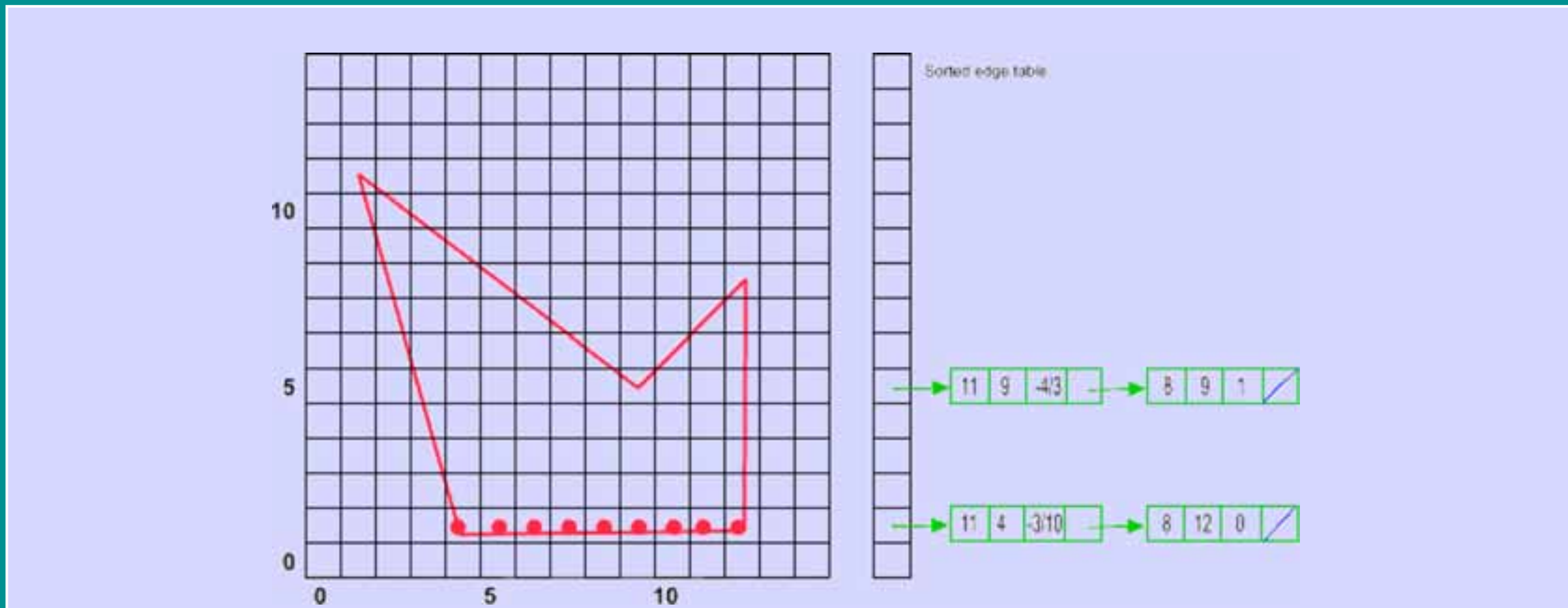


bucket sort edges into sorted edge table

initialize active edge list to first non empty scanline



Example: vertices (4,1), (1,11), (9,5), (12,8), (12,1)



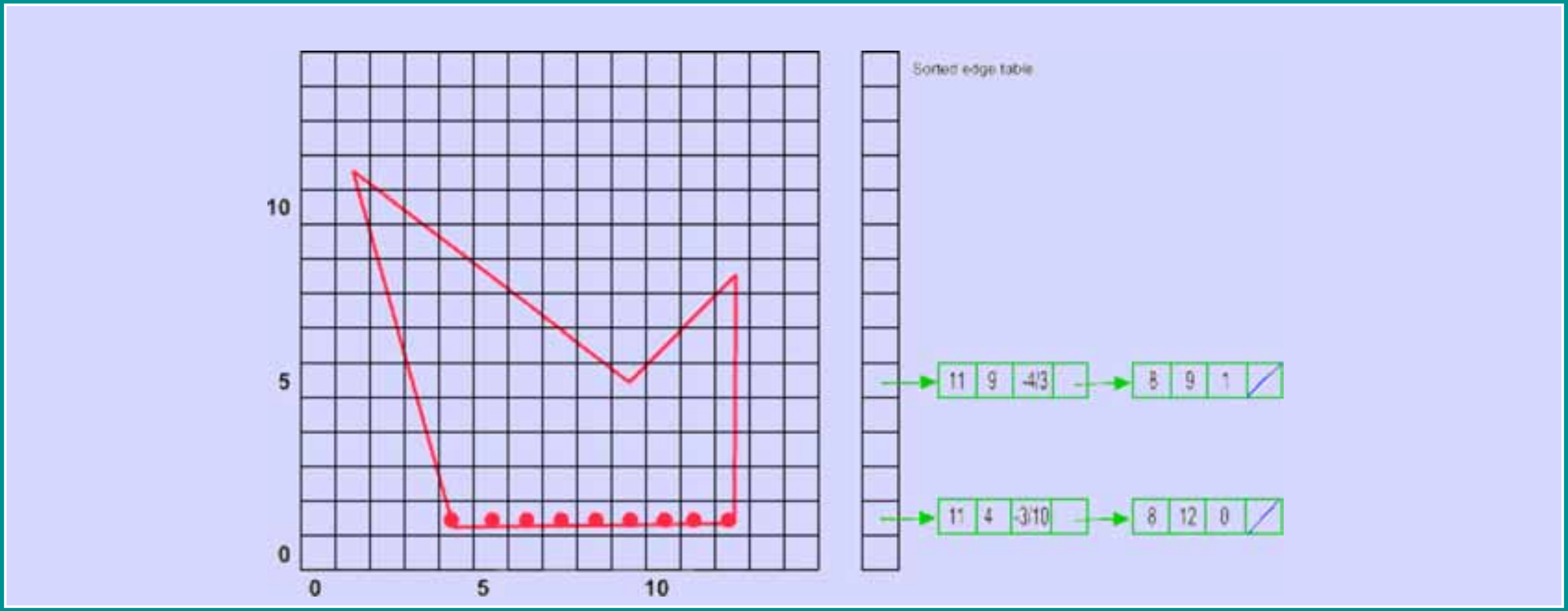
bucket sort edges into sorted edge table

initialize active edge list to first non empty scanline

for each non empty scanline

  fill between pairs (x=4,12)

Example: vertices (4,1), (1,11), (9,5), (12,8), (12,1)

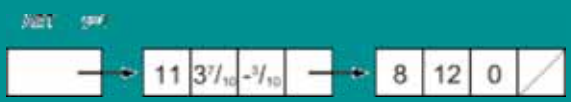


bucket sort edges into sorted edge table

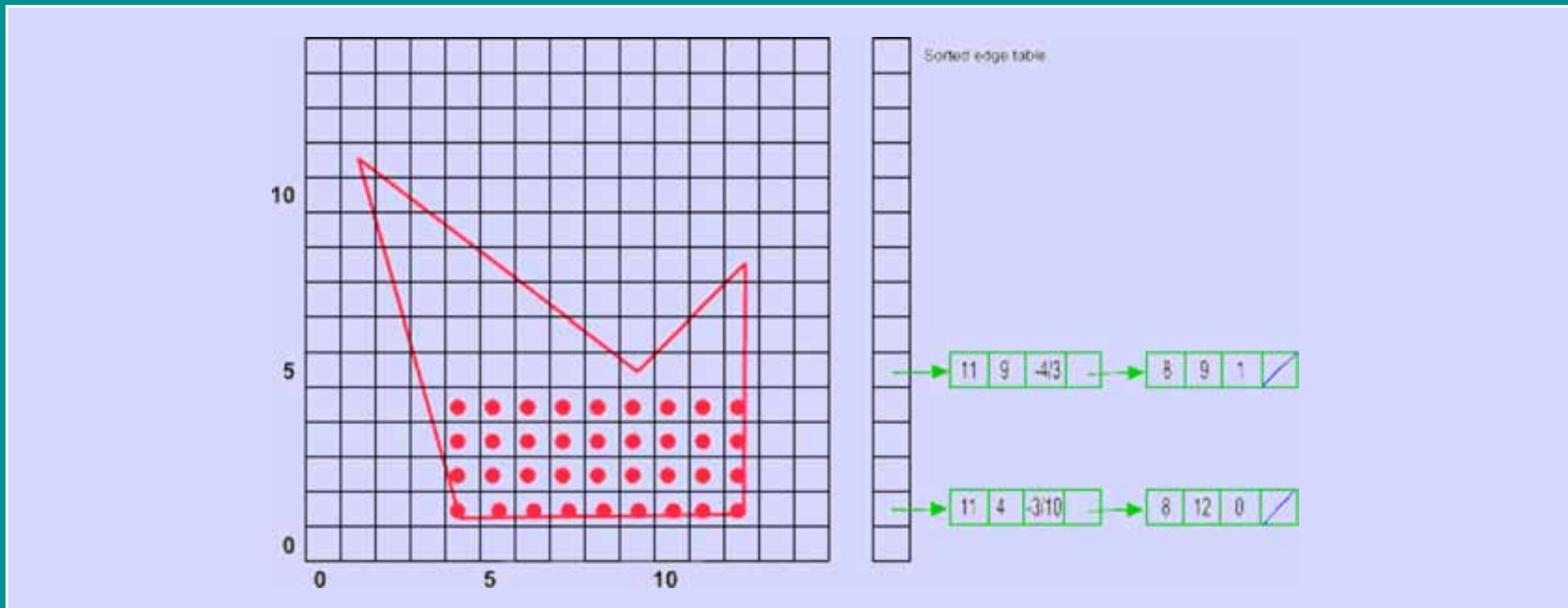
initialize active edge list to first non empty scanline

for each non empty scanline

- fill between pairs (x=4,12)
- remove exhausted edges
- update intersection points
- resort table
- add entering edges



Example: vertices (4,1), (1,11), (9,5), (12,8), (12,1)



bucket sort edges into sorted edge table

initialize active edge list to first non empty scanline

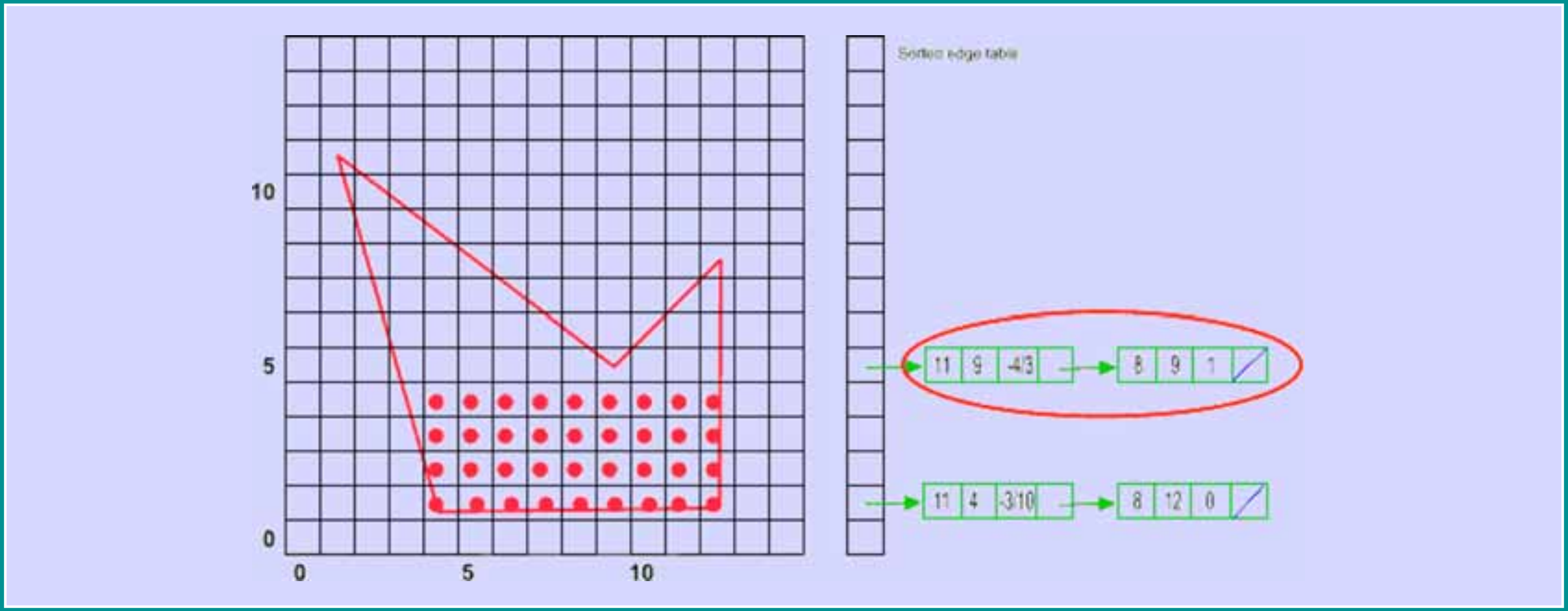
for each non empty scanline

  fill between pairs (x=3 1/10,12)

  remove exhausted edges

  update intersection points

Example: vertices (4,1), (1,11), (9,5), (12,8), (12,1)



bucket sort edges into sorted edge table

initialize active edge list to first non empty scanline

for each non empty scanline

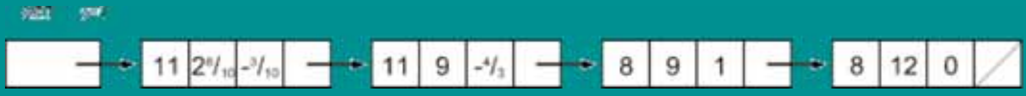
   fill between pairs (x=3 1/10,12)

   remove exhausted edges

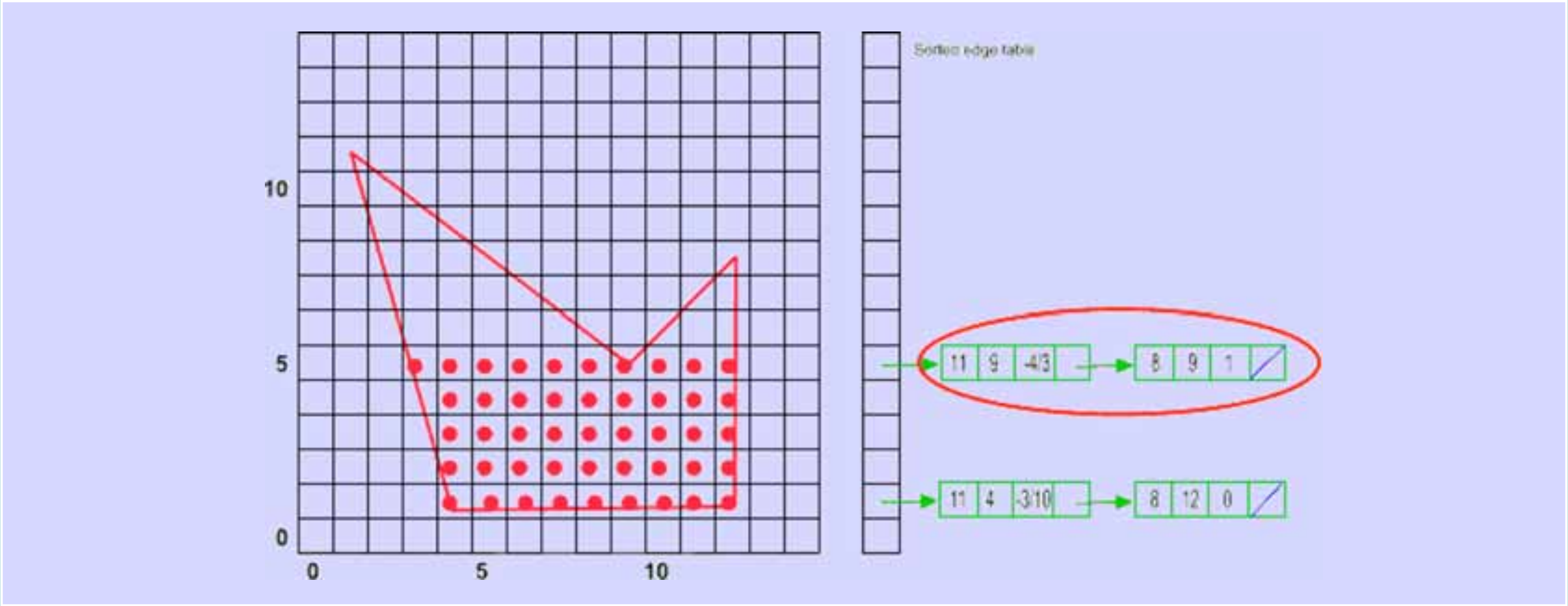
   update intersection points

   resort table

   add entering edges



Example: vertices (4,1), (1,11), (9,5), (12,8), (12,1)



bucket sort edges into sorted edge table

initialize active edge list to first non empty scanline

for each non empty scanline

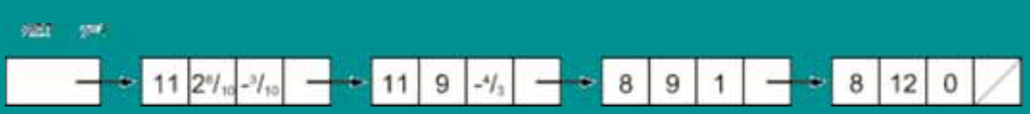
  fill between pairs (x = 2 8/10, 9; 9,12)

  remove exhausted edges

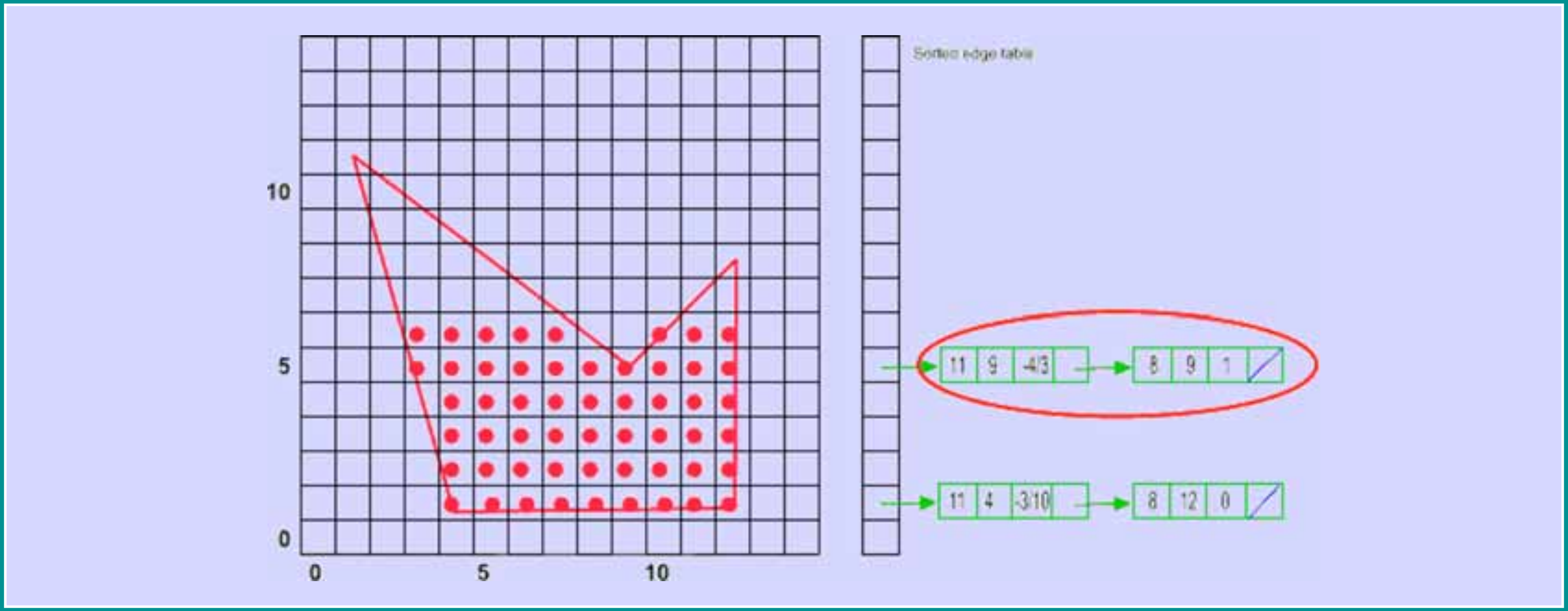
  update intersection points

  resort table

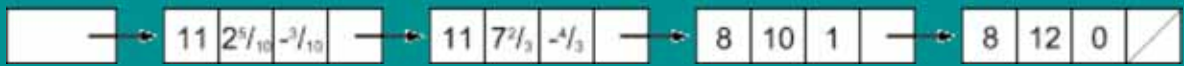
  add entering edges



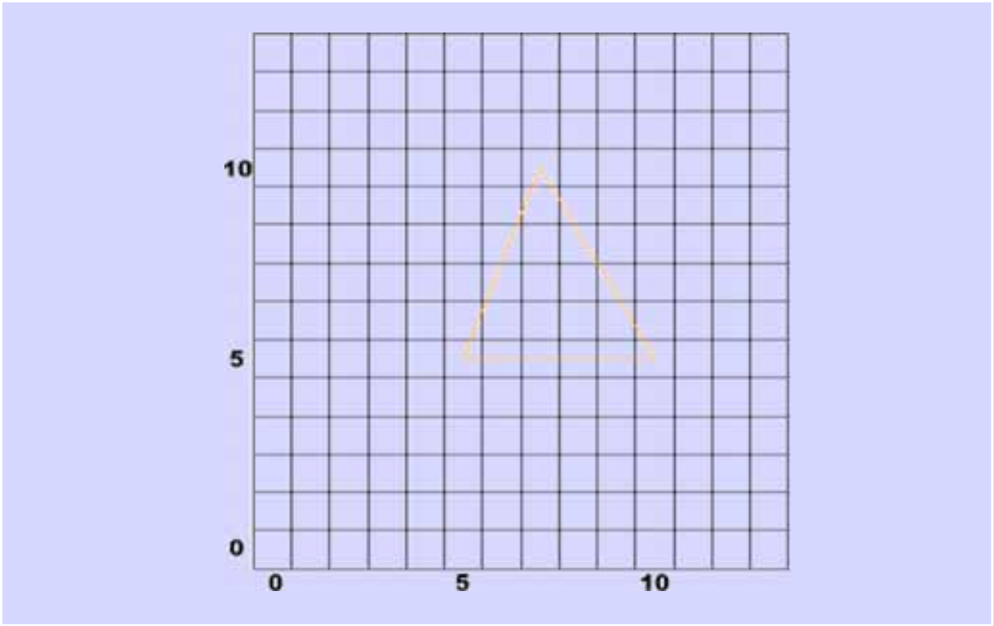
Example: vertices (4,1), (1,11), (9,5), (12,8), (12,1)



bucket sort edges into sorted edge table
initialize active edge list to first non empty scanline
for each non empty scanline
  fill between pairs (x=2 5/10, 7 2/3; 10,12)
  remove exhausted edges
  update intersection points
  resort table
  add entering edges



Fill Variants

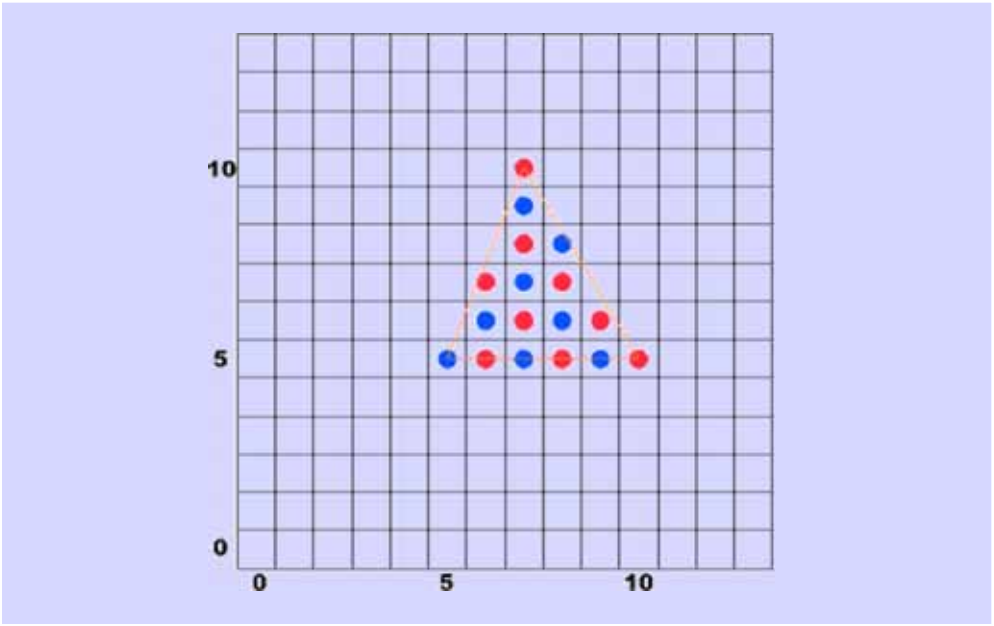


Fill between pairs:

```
for ( x = x1; x < x2; x++ )
    framebuffer [ x, y ] = c
```

Fill Variants (2)

- Pattern Fill

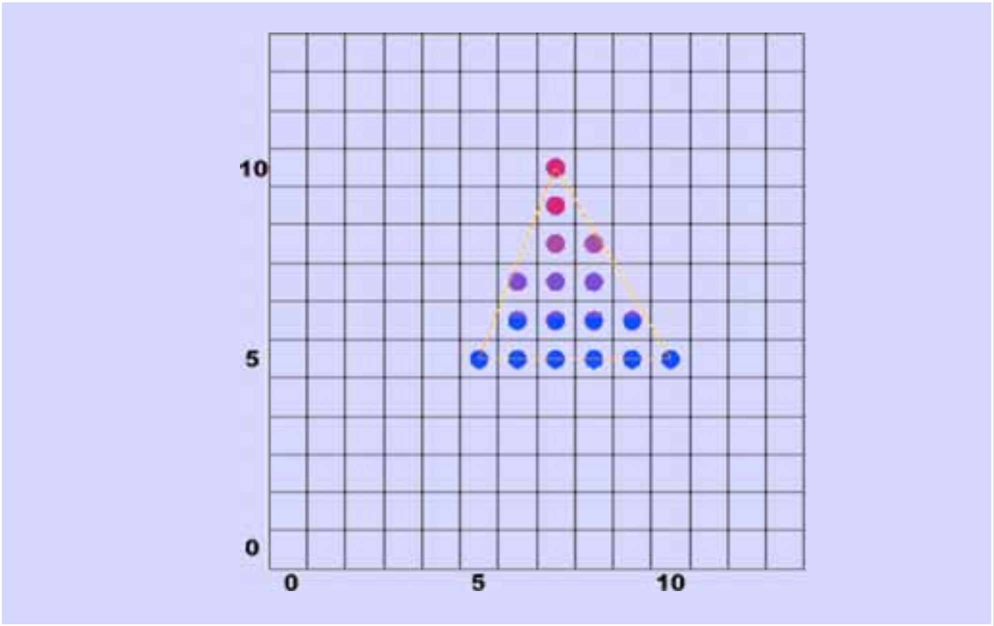


Fill between pairs:

```
for ( x = x1; x < x2; x++ )
      if ( ( x + y ) % 2 )
         framebuffer [ x, y ] = c1
      else
         framebuffer [ x, y ] = c1
```

Fill Variants (3)

- Colorwash
  Red to blue



Fill between pairs:

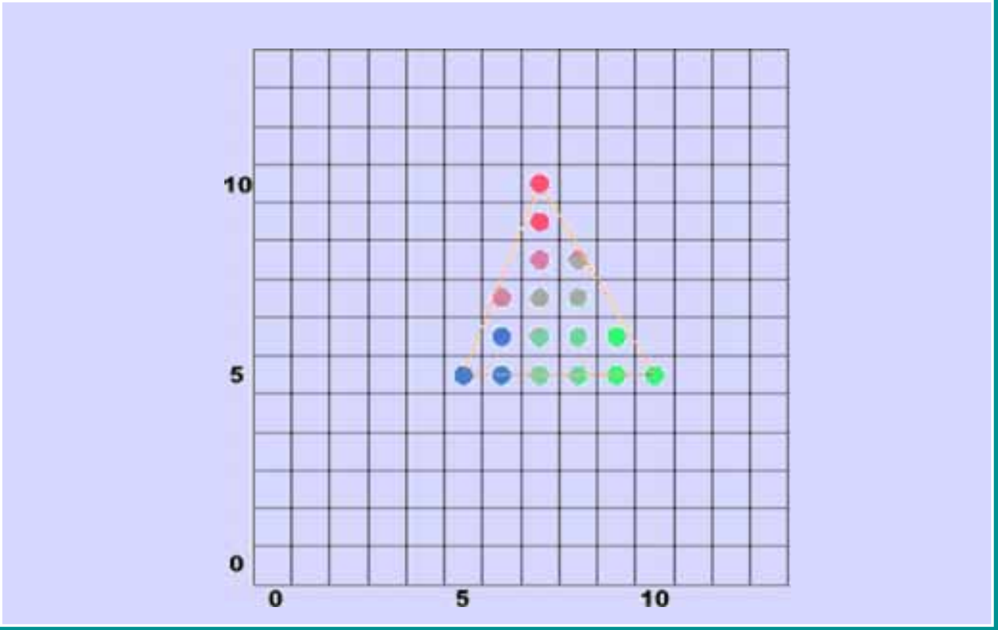
```
for ( x = x1; x < x2; x++ )
     framebuffer [ x, y ] = C0 + dC * ( x1 - x )
```

For efficiency carry C and dC in AETand calculate color incrementally

Fill Variants (4)

- Vertex colors

  Red, green, blue



Fill between pairs:

```
for ( x = x1; x < x2; x++ )
   framebuffer [ x, y ] =
      Cy1x1 + [(x - x1)/(x2 - x1)*(Cy1x2 - Cy1x1)]/dCx
```

For efficiency carry Cy and dCy in AET calculate dCx at beginning of scanline

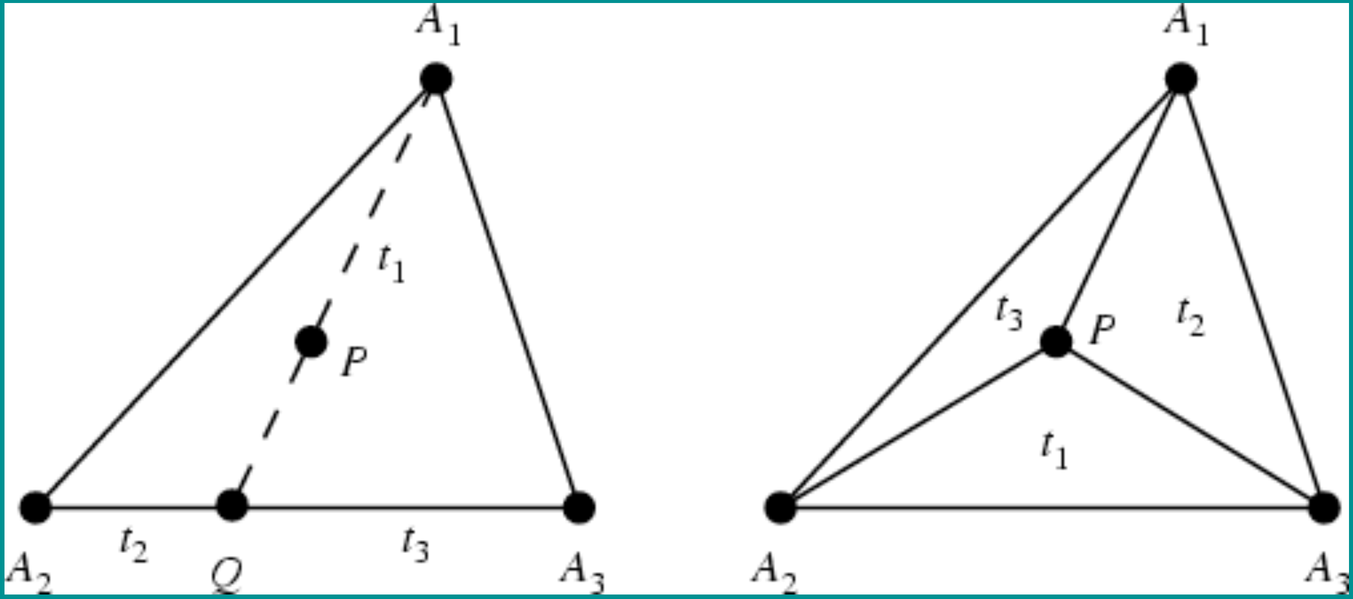
Barycentric Coordinates

- Use non-orthogonal coordinates to describe position relative to vertices

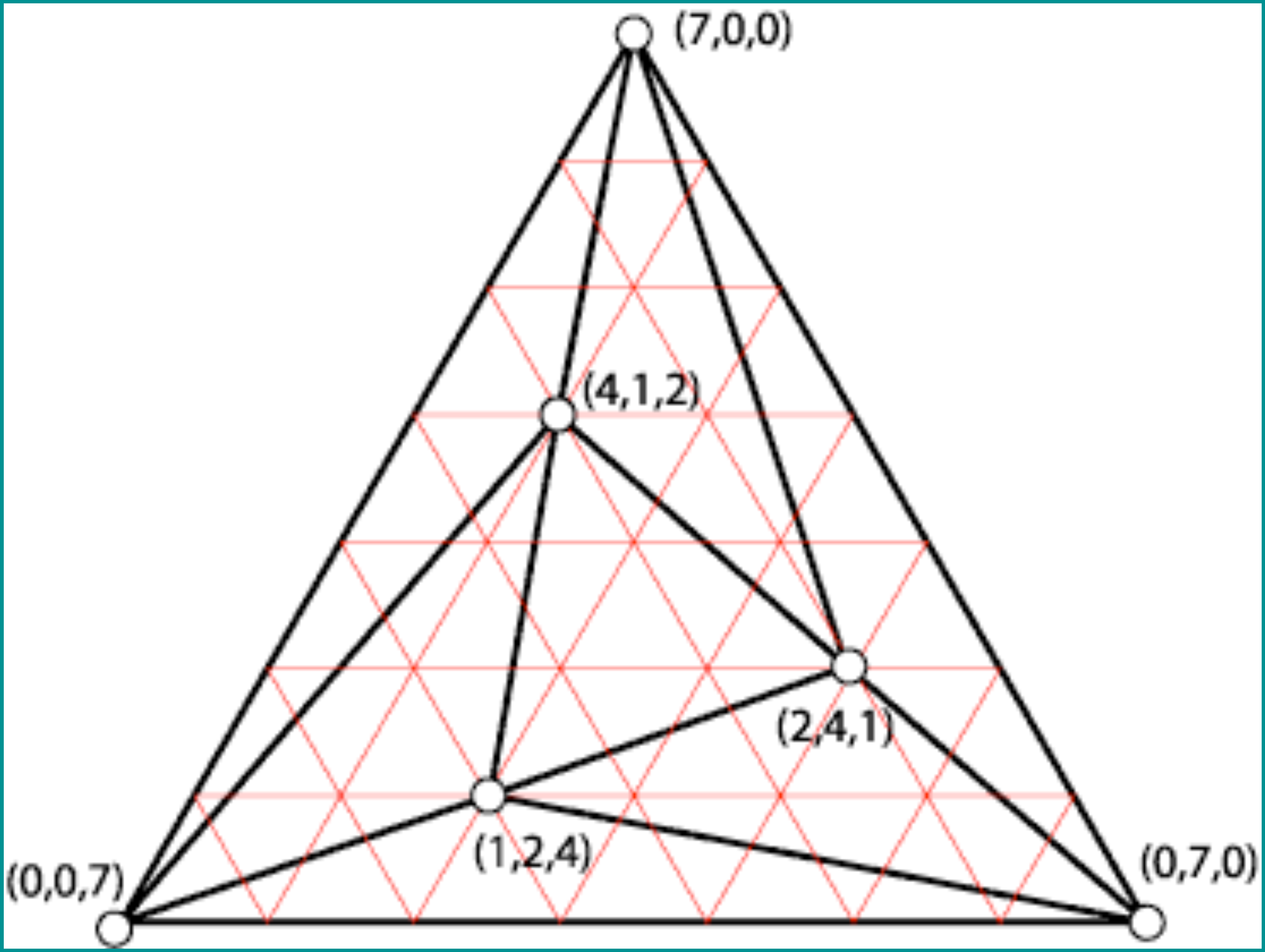
$$\mathrm{p} = \mathrm{a} + \beta(\mathrm{b} - \mathrm{a}) + \gamma(\mathrm{c} - \mathrm{a})$$

$$\mathrm{p}(\alpha, \beta, \gamma) = \alpha\mathrm{a} + \beta\mathrm{b} + \gamma\mathrm{c}$$

  - Coordinates correspond to scaled signed distance from lines through pairs of vertices



Barycentric Example



Barycentric Coordinates

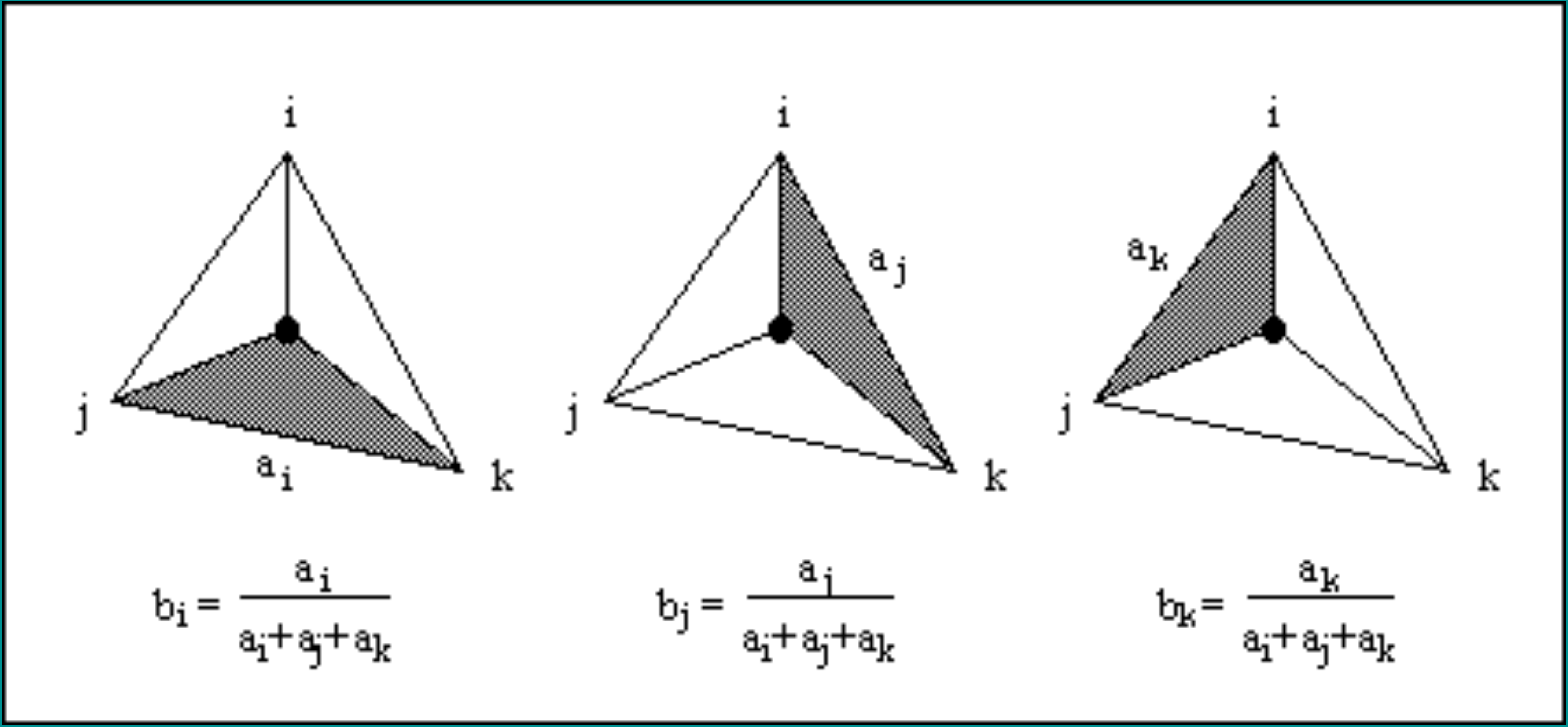
- Computing coordinates

$$\gamma = \frac{(y_a - y_b)x + (x_b - x_a)y + x_a y_b - x_b y_a}{(y_a - y_b)x_c + (x_b - x_a)y_c + x_a y_b - x_b y_a}$$

$$\beta = \frac{(y_a - y_c)x + (x_c - x_a)y + x_a y_c - x_b y_a}{(y_a - y_c)x_b + (x_c - x_a)y_b + x_a y_c - x_c y_a}$$

$$\alpha = 1 - \beta - \gamma$$

Alternative Computation



Barycentric Rasterization

```
For all x do
    For all y do
        Compute (α, β, γ) for (x,y)
        If (α ∈ [0,1] and β ∈ [0,1] and γ ∈ [0,1] then
            c = αc0 + βc1 + γc2
            Draw pixel (x,y) with color c
```

Barycentric Rasterization

x_{min} = floor(x_i)

x_{max} = ceiling(x_i)

y_{min} = floor(y_i)

y_{max} = ceiling(x_i)

for y = y_{min} to y_{max} do

for x = x_{min} to x_{max} do

$\alpha = f_{12}(x,y)/f_{12}(x_0,y_0)$

$\beta = f_{20}(x,y)/f_{20}(x_1,y_1)$

$\gamma = f_{01}(x,y)/f_{01}(x_2,y_2)$

If ($\alpha \in [0,1]$ and $\beta \in [0,1]$ and $\gamma \in [0,1]$ then

$c = \alpha c_0 + \beta c_1 + \gamma c_2$

Draw pixel (x,y) with color c

Barycentric Rasterization

- Computing coordinates

$$\gamma = \frac{f_{01}(x,y)}{f_{01}(x_2,y_2)} = \frac{(y_0 - y_1)x + (x_1 - x_0)y + x_0 y_1 - x_1 y_0}{(y_0 - y_1)x_2 + (x_1 - x_0)y_2 + x_0 y_1 - x_1 y_0}$$

$$\beta = \frac{f_{20}(x,y)}{f_{20}(x_1,y_1)} = \frac{(y_2 - y_0)x + (x_0 - x_2)y + x_2 y_0 - x_0 y_2}{(y_2 - y_0)x_1 + (x_0 - x_2)y_1 + x_2 y_0 - x_0 y_2}$$

$$\alpha = \frac{f_{12}(x,y)}{f_{12}(x_0,y_0)} = \frac{(y_1 - y_2)x + (x_2 - x_1)y + x_1 y_2 - x_2 y_1}{(y_1 - y_2)x_0 + (x_2 - x_1)y_0 + x_1 y_2 - x_2 y_1}$$