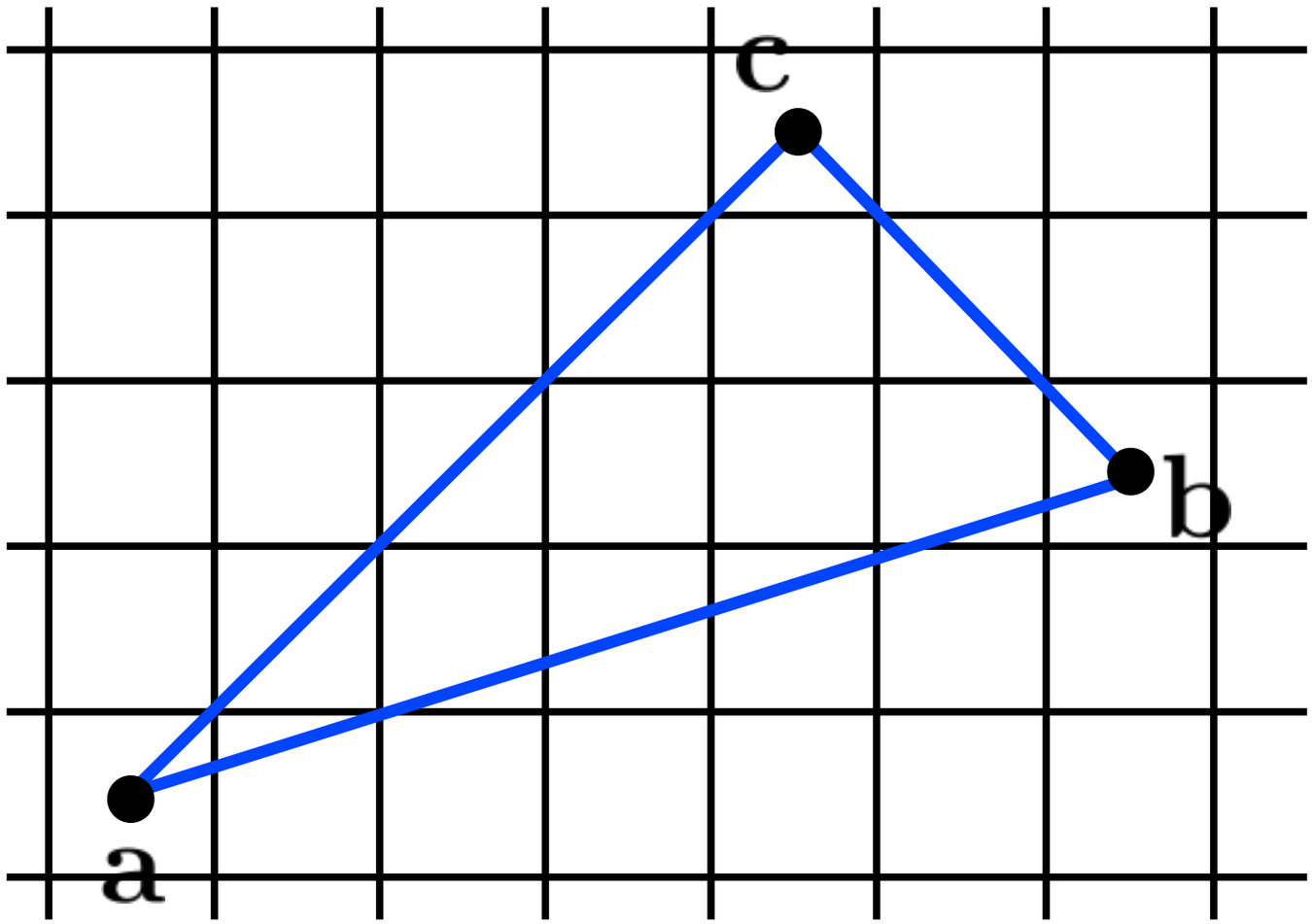
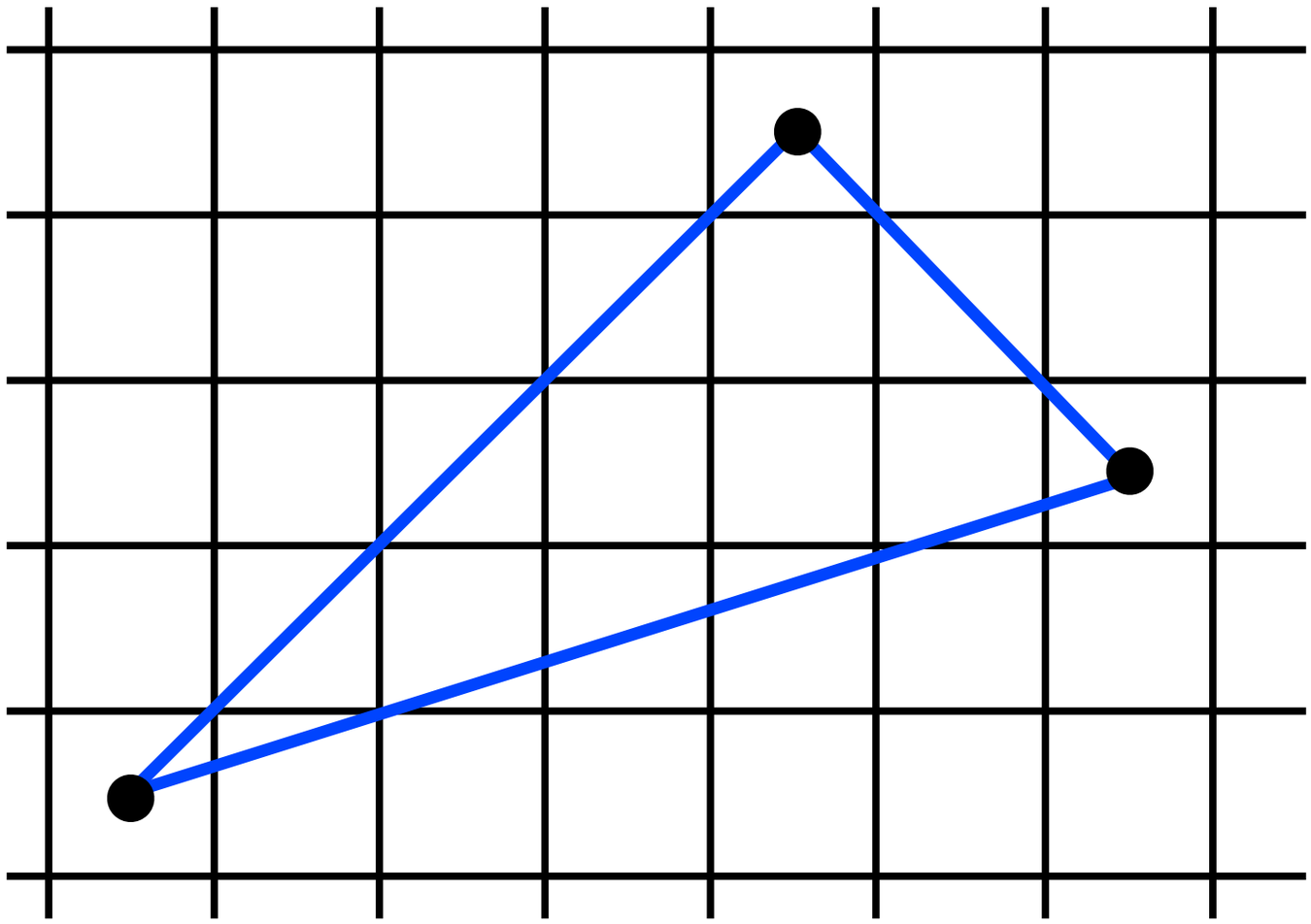
Triangle rasterization

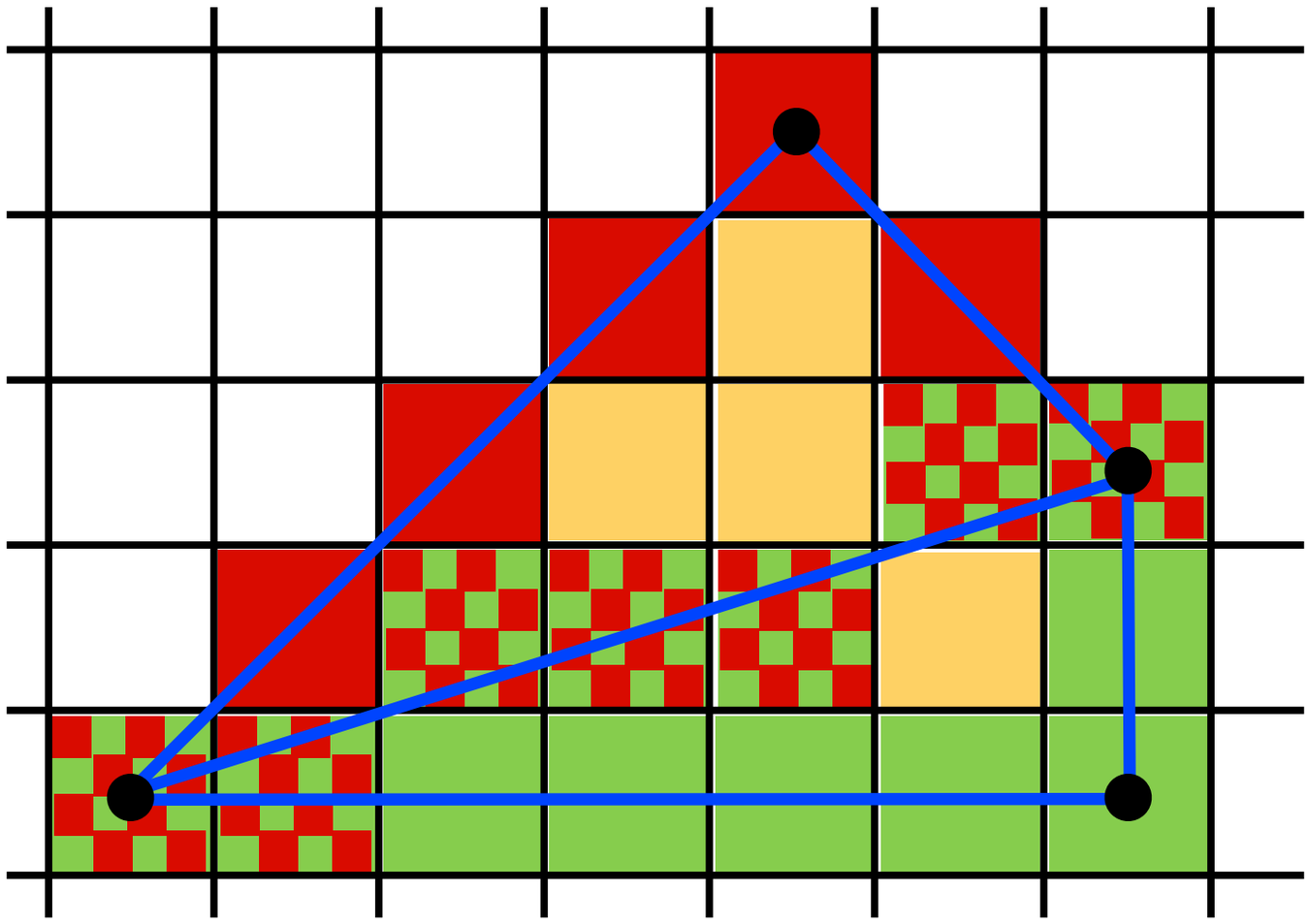
Which pixels should be used to approximate a triangle?



Triangle rasterization issues

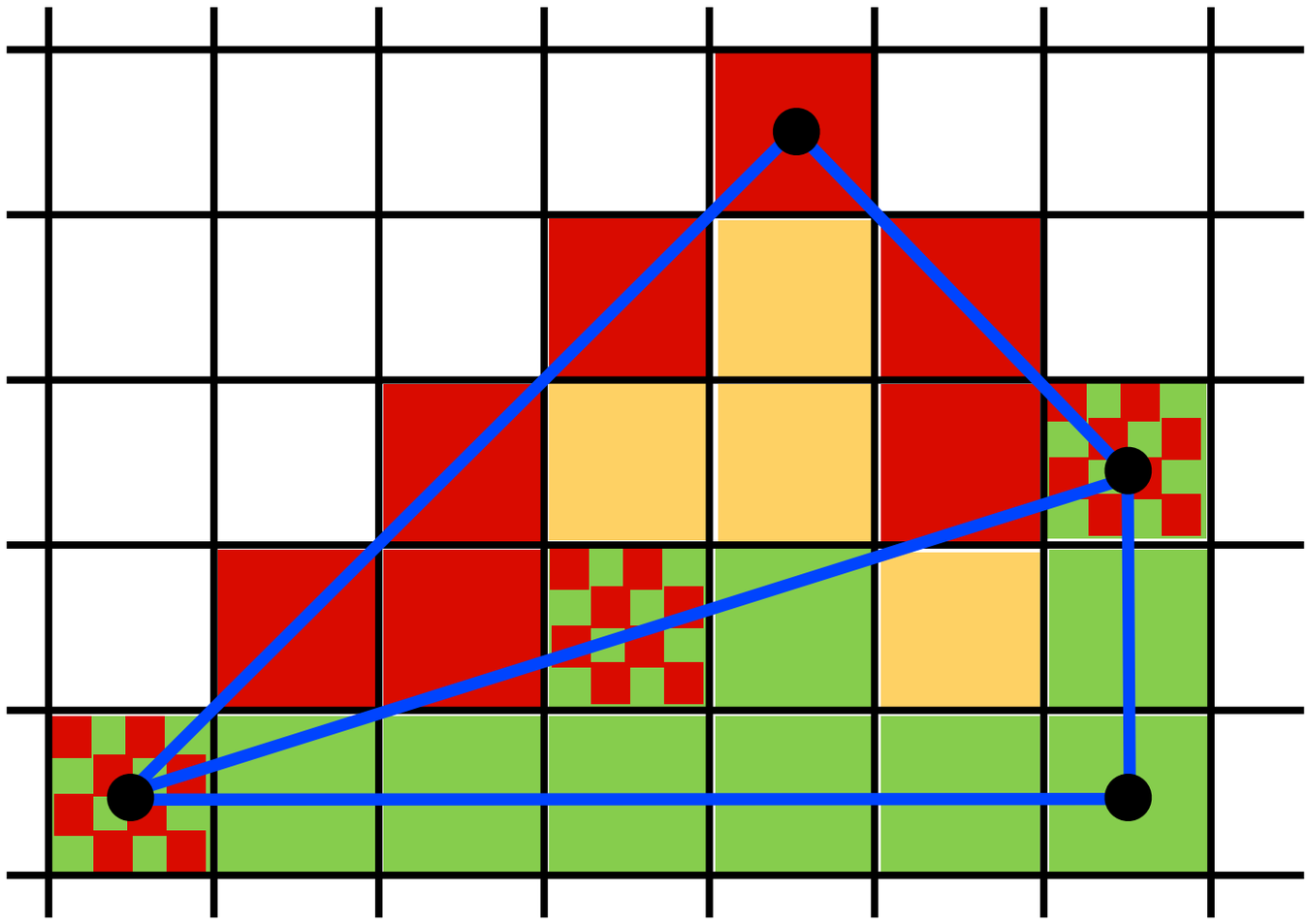


Which pixels should be used to approximate a triangle?



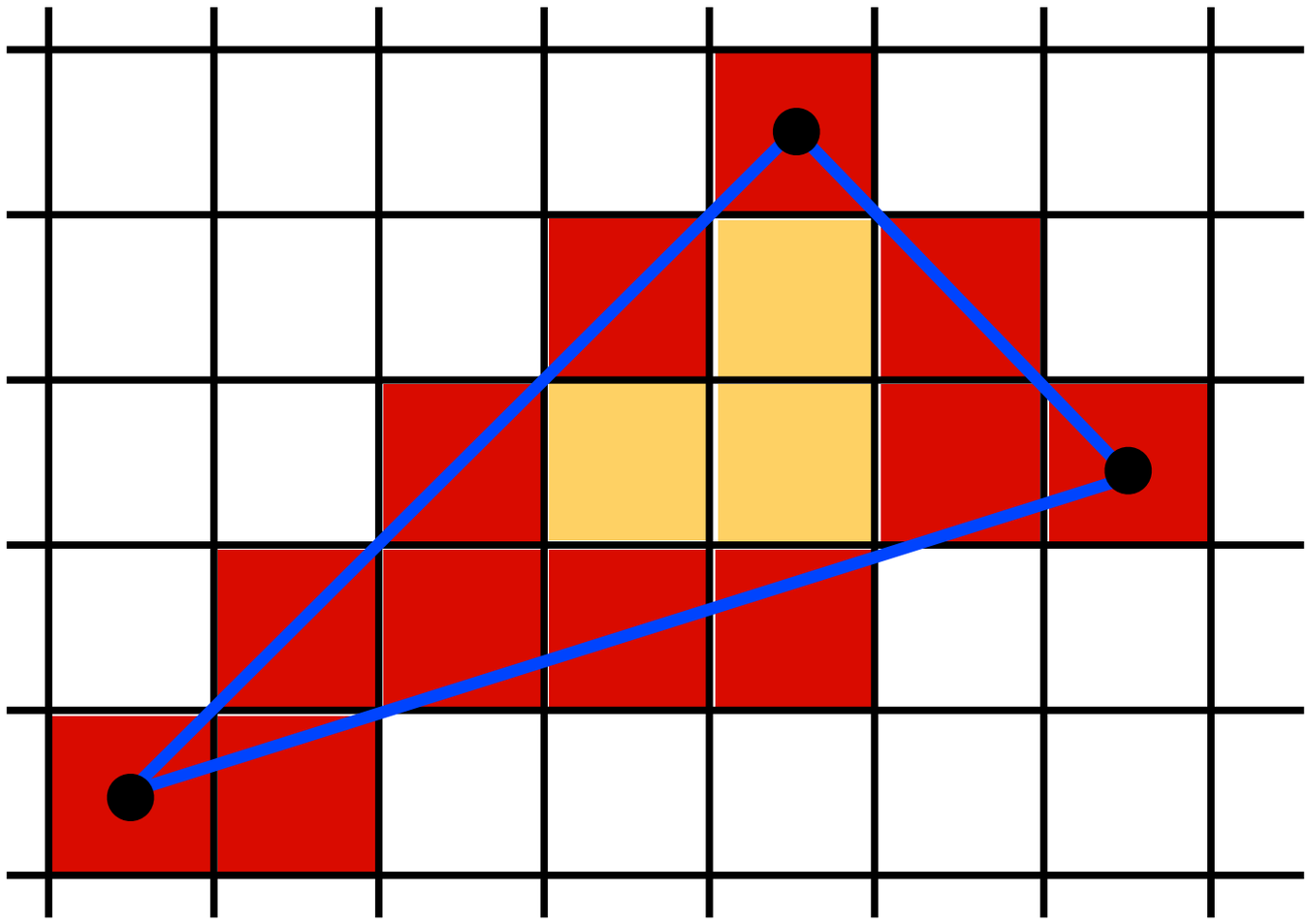
Who should fill in shared edge?

Which pixels should be used to approximate a triangle?



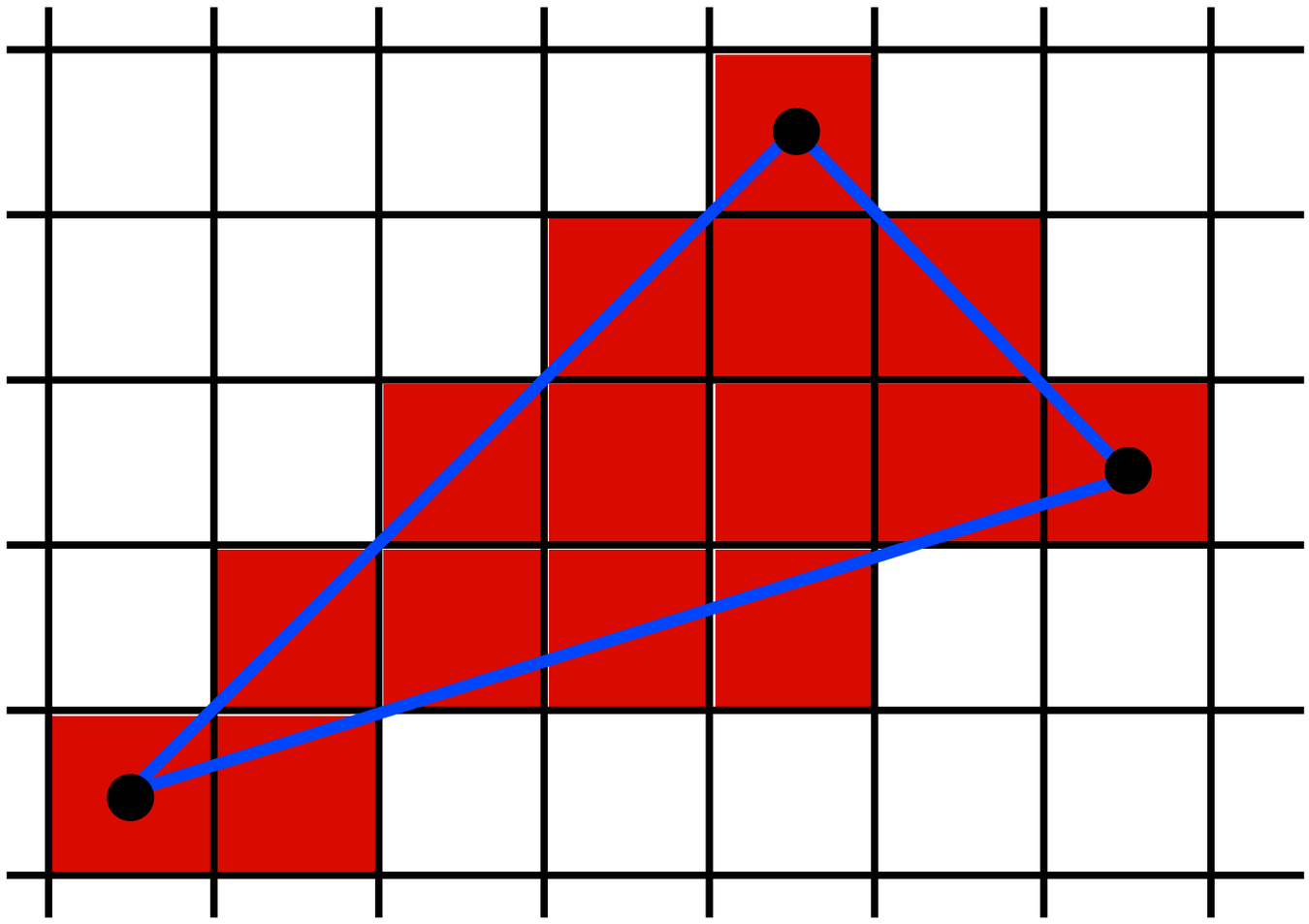
Who should fill in shared edge?

Which pixels should be used to approximate a triangle?



Use Midpoint Algorithm for edges and fill in?

Which pixels should be used to approximate a triangle?



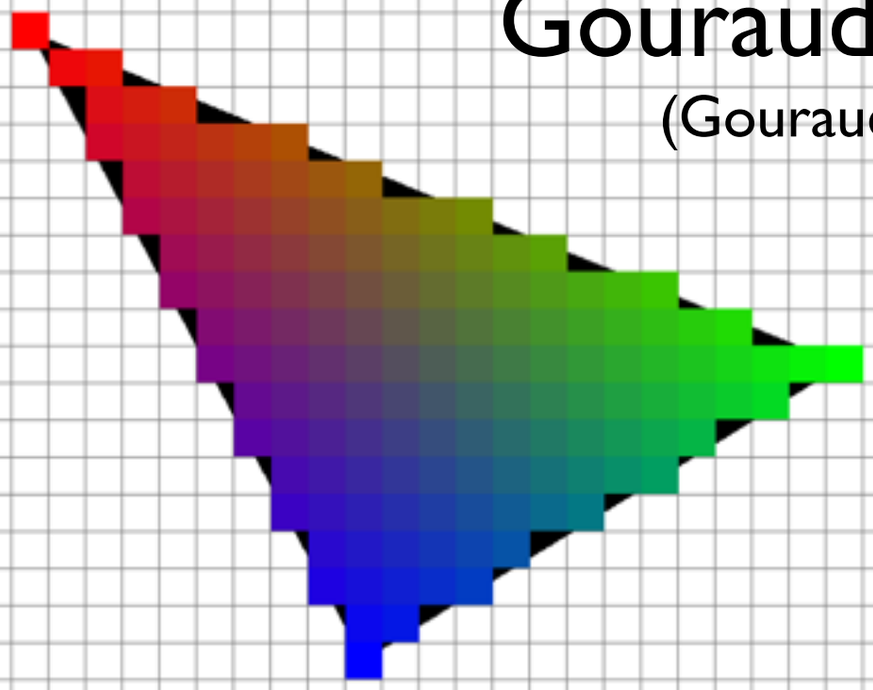
Use an approach based on barycentric coordinates

We can interpolate attributes using barycentric coordinates

$$\mathbf{c} = \alpha\mathbf{c}_0 + \beta\mathbf{c}_1 + \gamma\mathbf{c}_2$$

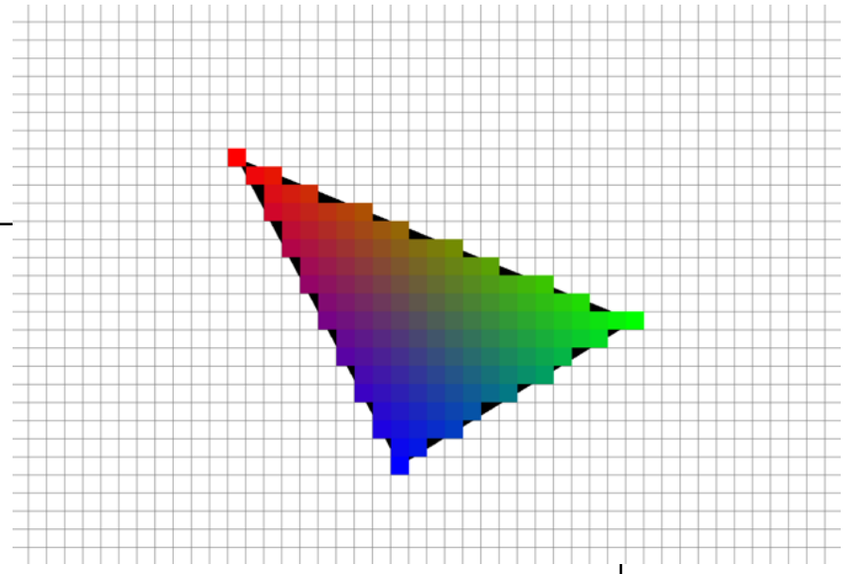
Gouraud shading

(Gouraud, 1971)



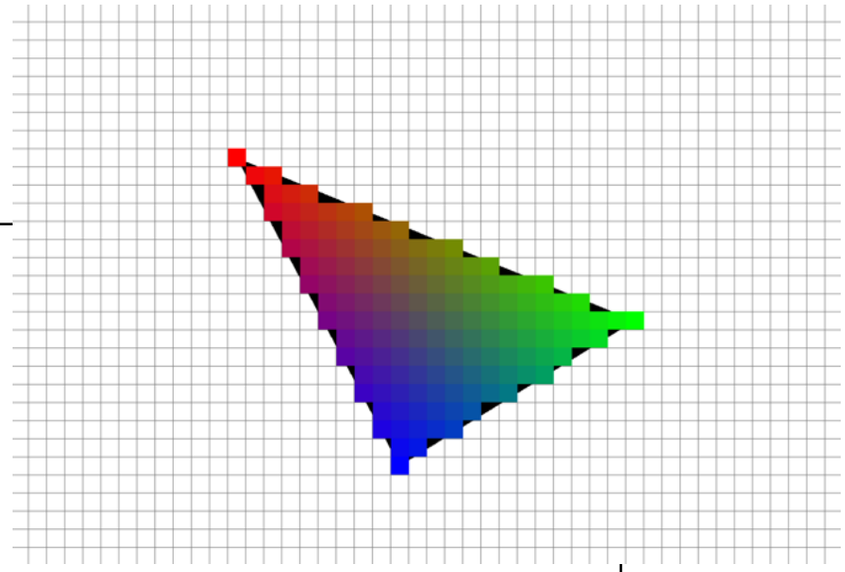
http://jtibble.dyndns.org/graphics/eecs487/eecs487.html

Triangle rasterization algorithm



```
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
            drawpixel(x,y) with color c
```

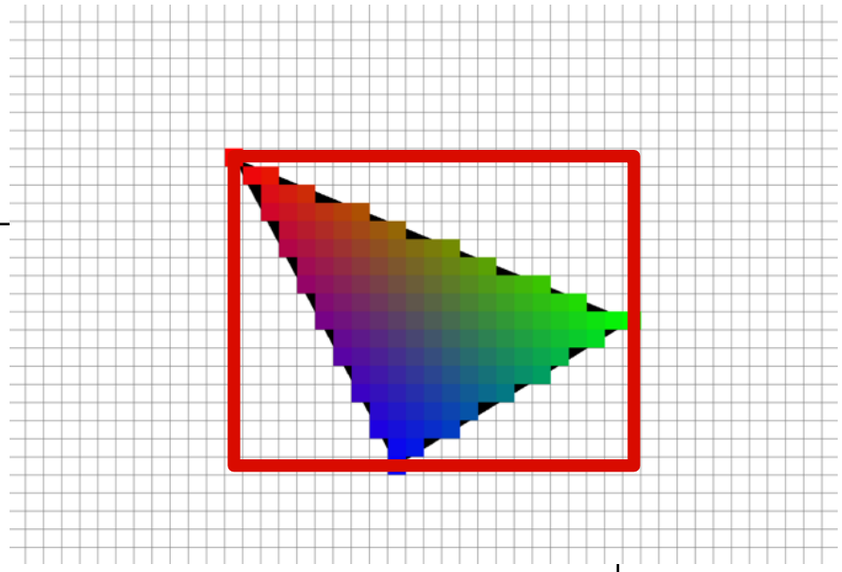
Triangle rasterization algorithm



for all x do
  for all y do
    compute (α, β, γ) for (x,y)
    if $(\alpha \in [0,1]$ and $\beta \in [0,1]$ and $\gamma \in [0,1])$ then
      $\mathbf{c} = \alpha\mathbf{c}_0 + \beta\mathbf{c}_1 + \gamma\mathbf{c}_2$
      drawpixel(x,y) with color c

Triangle rasterization algorithm

use a bounding rectangle



```
for x in [x_min, x_max]
    for y in [y_min, y_max]
        compute (α, β, γ) for (x,y)
        if (α ∈ [0, 1] and β ∈ [0, 1] and γ ∈ [0, 1]) then
            c = αc0 + βc1 + γc2
            drawpixel(x,y) with color c
```

Triangle rasterization algorithm

for x in [x_min, x_max]
  for y in [y_min, y_max]

$$\alpha = f_{bc}(x,y)/f_{bc}(x_a,y_a)$$

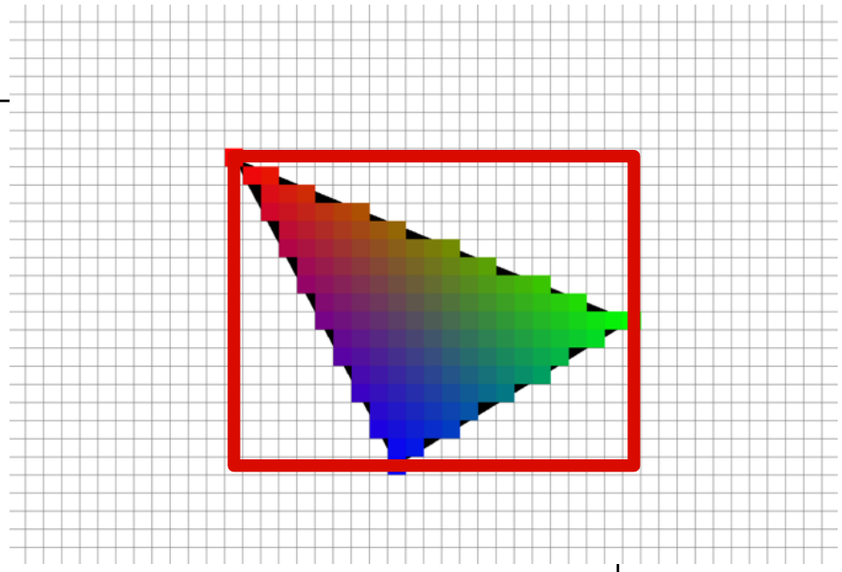
$$\beta = f_{ca}(x,y)/f_{ca}(x_b,y_b)$$

$$\gamma = f_{ab}(x,y)/f_{ab}(x_c,y_c)$$

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

$$\mathbf{c} = \alpha\mathbf{c}_0 + \beta\mathbf{c}_1 + \gamma\mathbf{c}_2$$

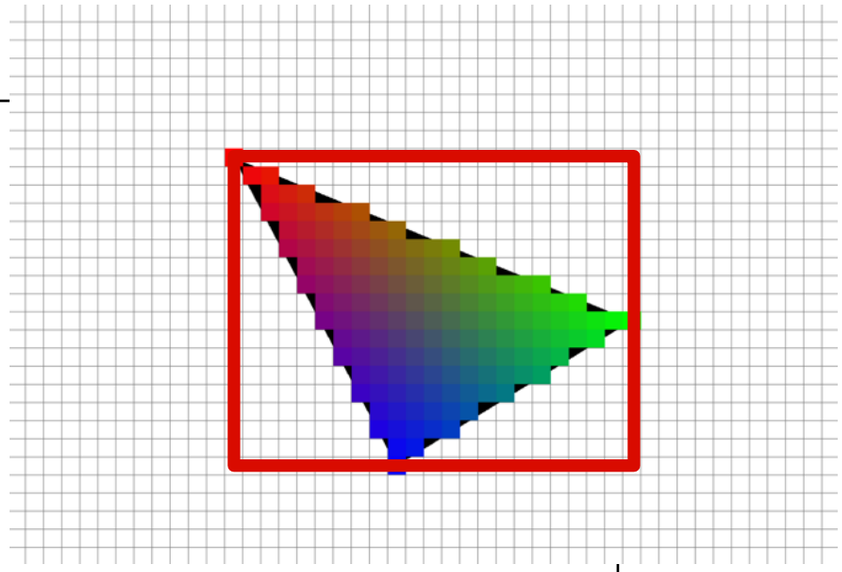
drawpixel(x,y) with color c



<whiteboard>

Triangle rasterization algorithm

Optimizations?



for x in [x_min, x_max]
  for y in [y_min, y_max]

$$\alpha = f_{bc}(x, y) / f_{bc}(x_a, y_a)$$

$$\beta = f_{ca}(x, y) / f_{ca}(x_b, y_b)$$

$$\gamma = f_{ab}(x, y) / f_{ab}(x_c, y_c)$$

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

$$\mathbf{c} = \alpha \mathbf{c}_0 + \beta \mathbf{c}_1 + \gamma \mathbf{c}_2$$

drawpixel(x,y) with color c

Triangle rasterization algorithm

Optimizations?

for x in [x_min, x_max]
  for y in [y_min, y_max]

$$\alpha = f_{bc}(x,y)/f_{bc}(x_a,y_a)$$

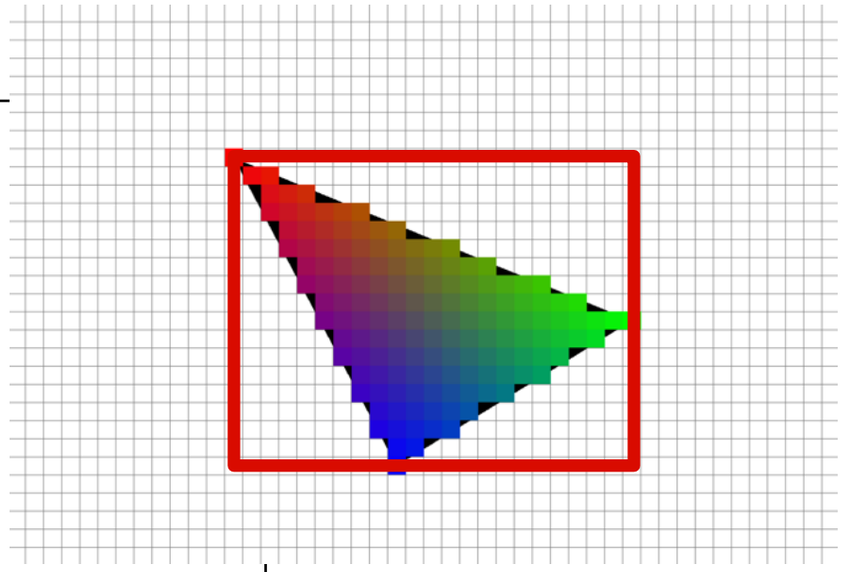
$$\beta = f_{ca}(x,y)/f_{ca}(x_b,y_b)$$

$$\gamma = f_{ab}(x,y)/f_{ab}(x_c,y_c)$$

if $(\alpha \geq 0 \text{ and } \beta \geq 0 \text{ and } \gamma \geq 0)$ then

$$\mathbf{c} = \alpha\mathbf{c}_0 + \beta\mathbf{c}_1 + \gamma\mathbf{c}_2$$

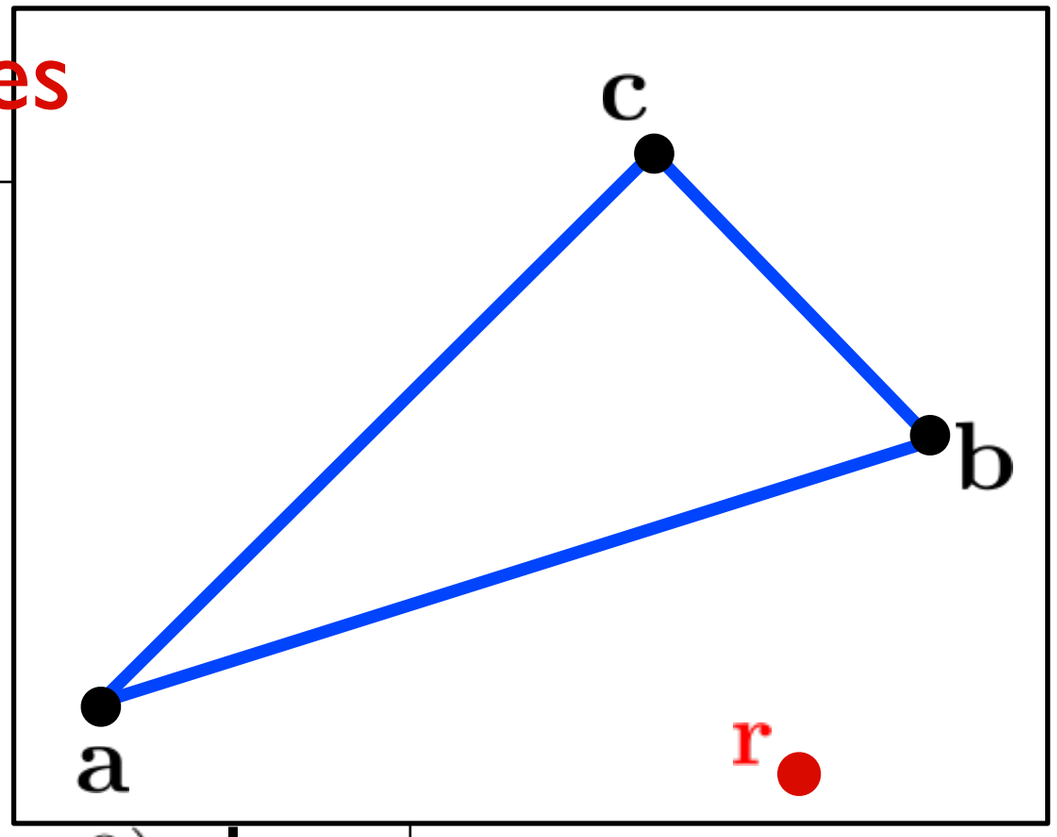
drawpixel(x,y) with color c



make computation of bary. coords. incremental
color can also be computed incrementally
don't need to check upper bound

Triangle rasterization algorithm

dealing with shared triangle edges



for x in [x_min, x_max]

for y in [y_min, y_max]

$\alpha = f_{bc}(x, y) / f_{bc}(x_a, y_a)$

$\beta = f_{ac}(x, y) / f_{ac}(x_b, y_b)$

$\gamma = f_{ab}(x, y) / f_{ab}(x_c, y_c)$

if $(\alpha \geq 0$ and $\beta \geq 0$ and $\gamma \geq 0)$ then

if $(\alpha > 0$ or $f_{bc}(\mathbf{a}) f_{bc}(\mathbf{r}) > 0)$ and
$(\beta > 0$ or $f_{ca}(\mathbf{b}) f_{ca}(\mathbf{r}) > 0)$ and
$(\gamma > 0$ or $f_{ab}(\mathbf{c}) f_{ab}(\mathbf{r}) > 0)$ then

$\mathbf{c} = \alpha \mathbf{c}_0 + \beta \mathbf{c}_1 + \gamma \mathbf{c}_2$

drawpixel(x,y) with color c