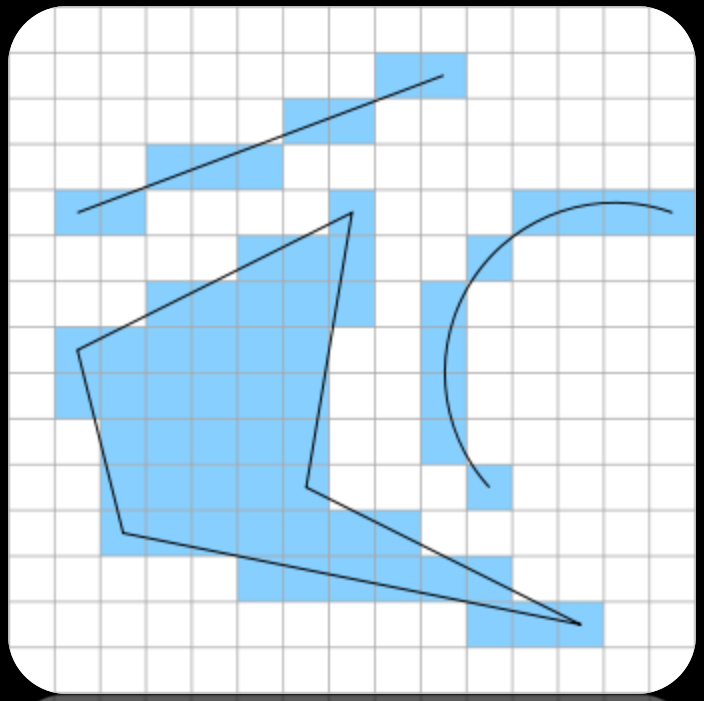
http://iloveshaders.blogspot.com/2011/05/how-rasterization-process-works.html



Rasterization

CS 148, Summer 2012
Introduction to Computer Graphics and Imaging

Justin Solomon

Render [**ren**-der]:

To generate an image or animation.

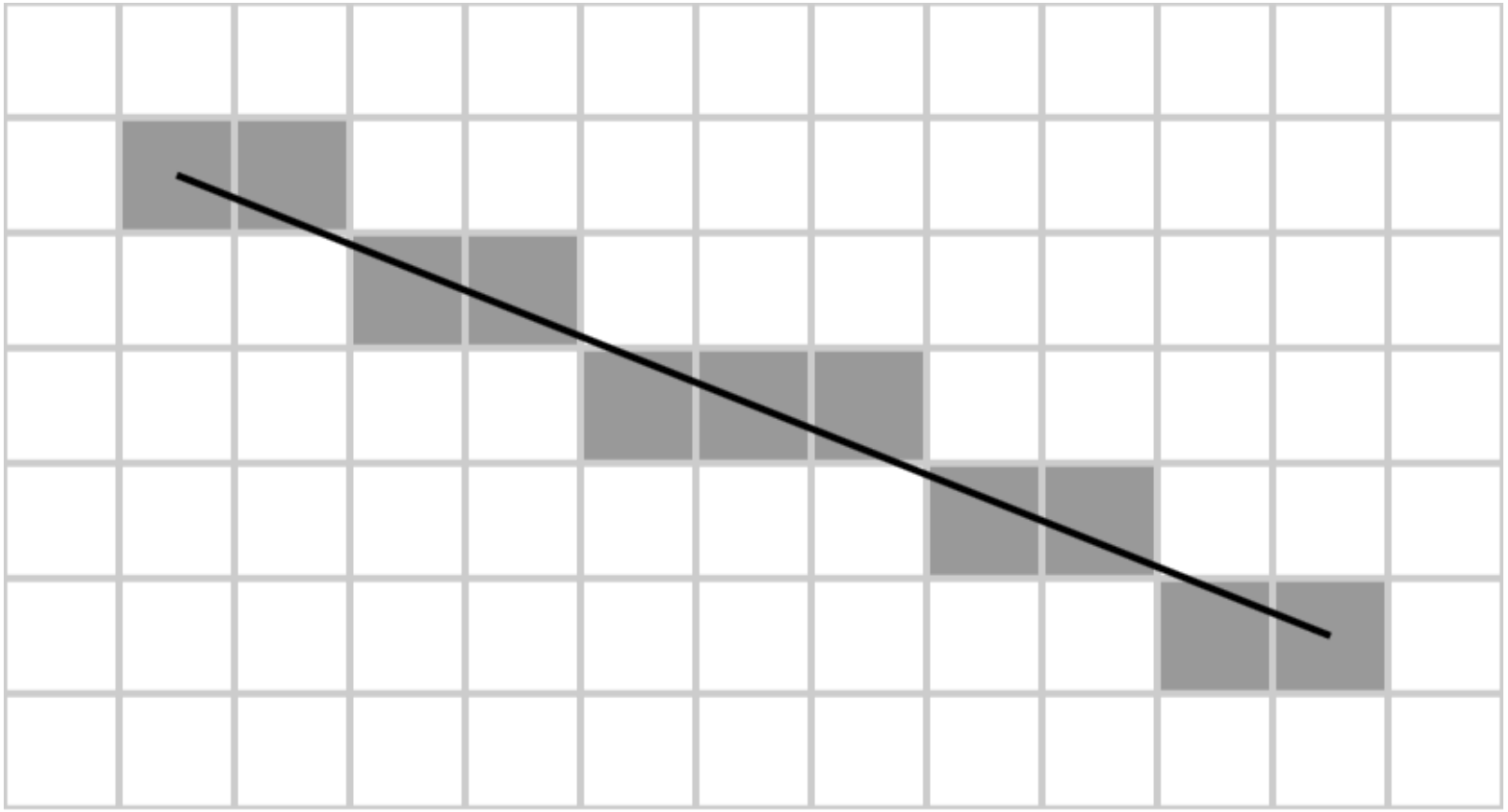
Two Ways to Make an Image

1. Drawing

2. Photography

http://www.publicdomainpictures.net/view-image.php?picture=female-photographer&image=4360&large=1
http://bloodredpencil.blogspot.com/2010/09/need-help-get-help.html

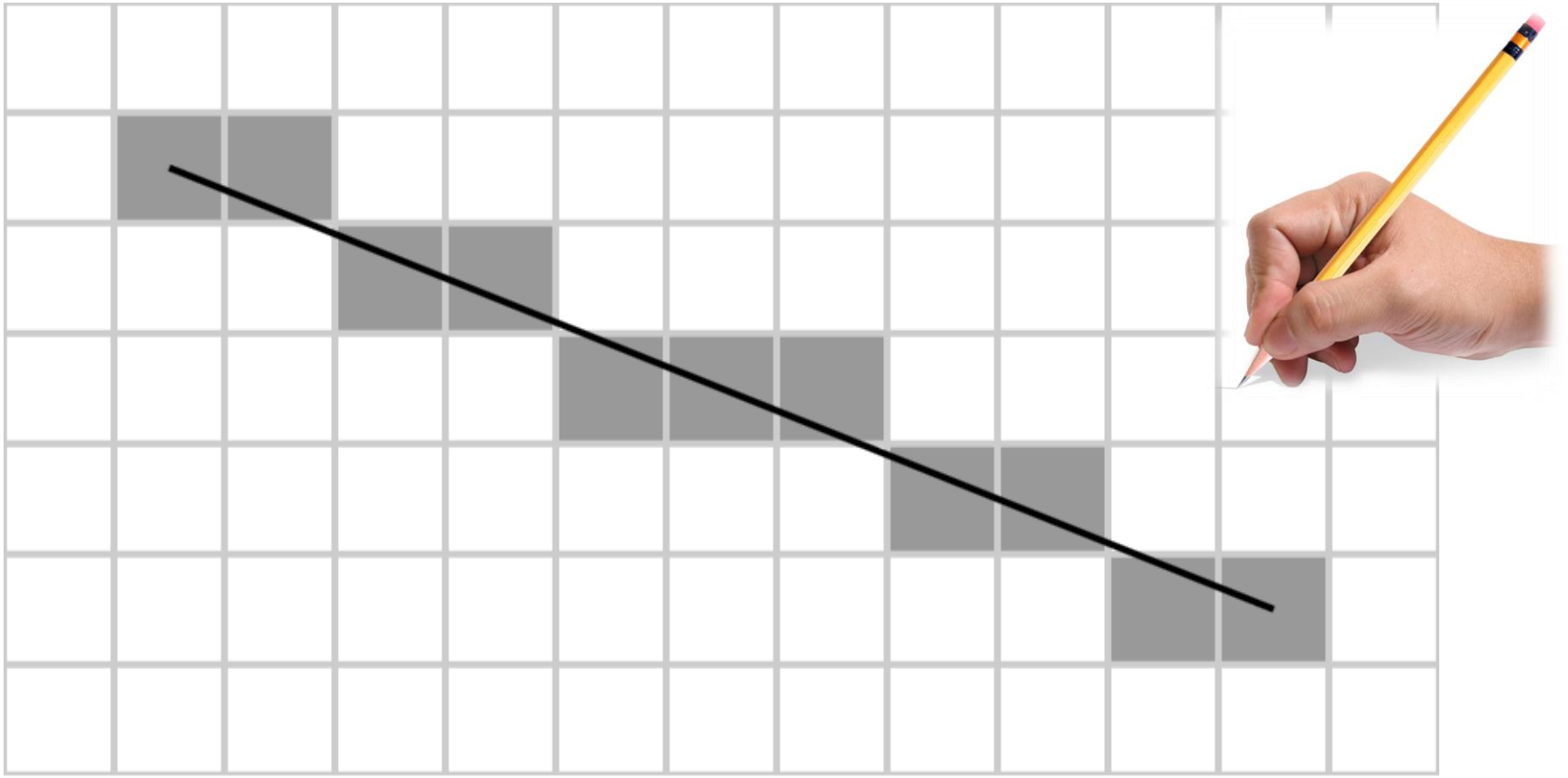
Two Ways to Render an Image



http://upload.wikimedia.org/wikipedia/commons/a/ab/Bresenham.svg

Rasterization

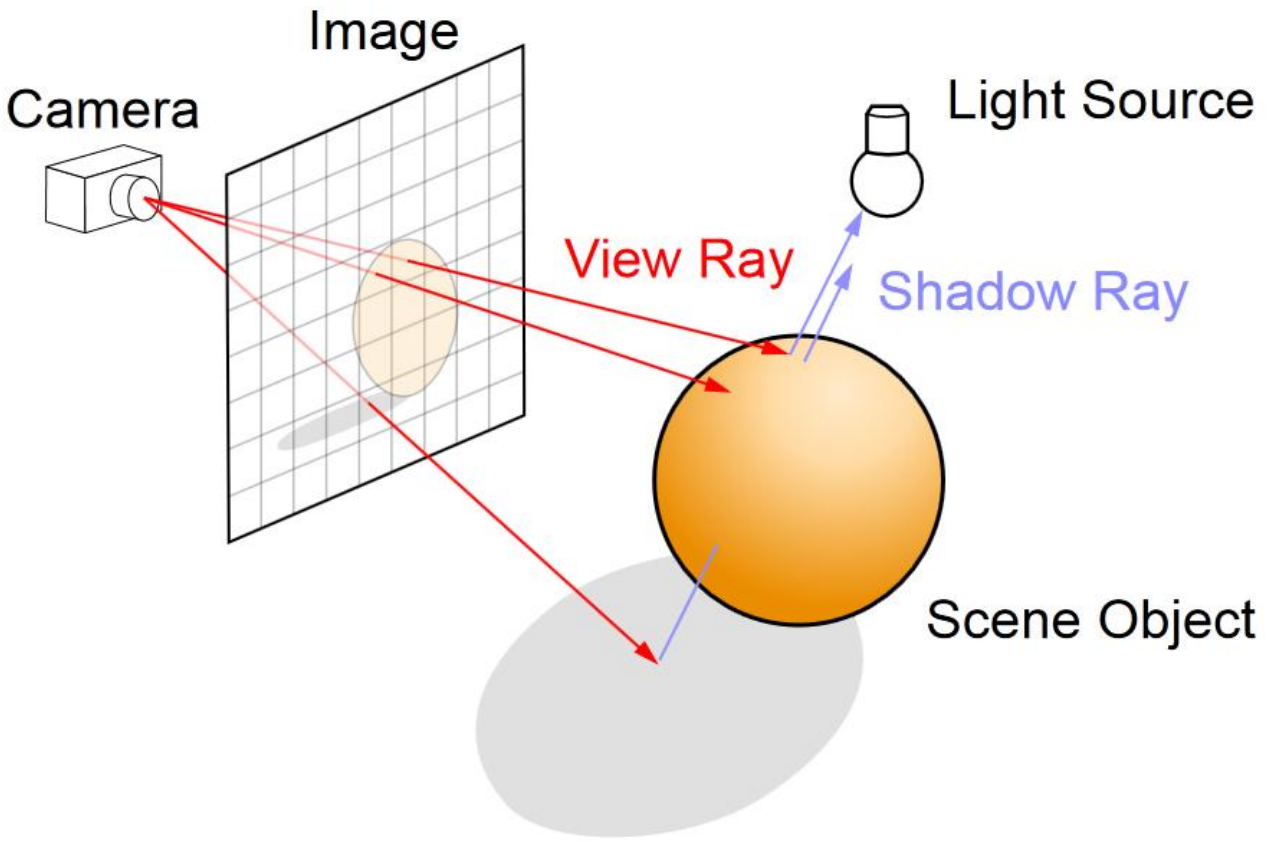
Two Ways to Render an Image



http://upload.wikimedia.org/wikipedia/commons/a/ab/Bresenham.svg

Rasterization

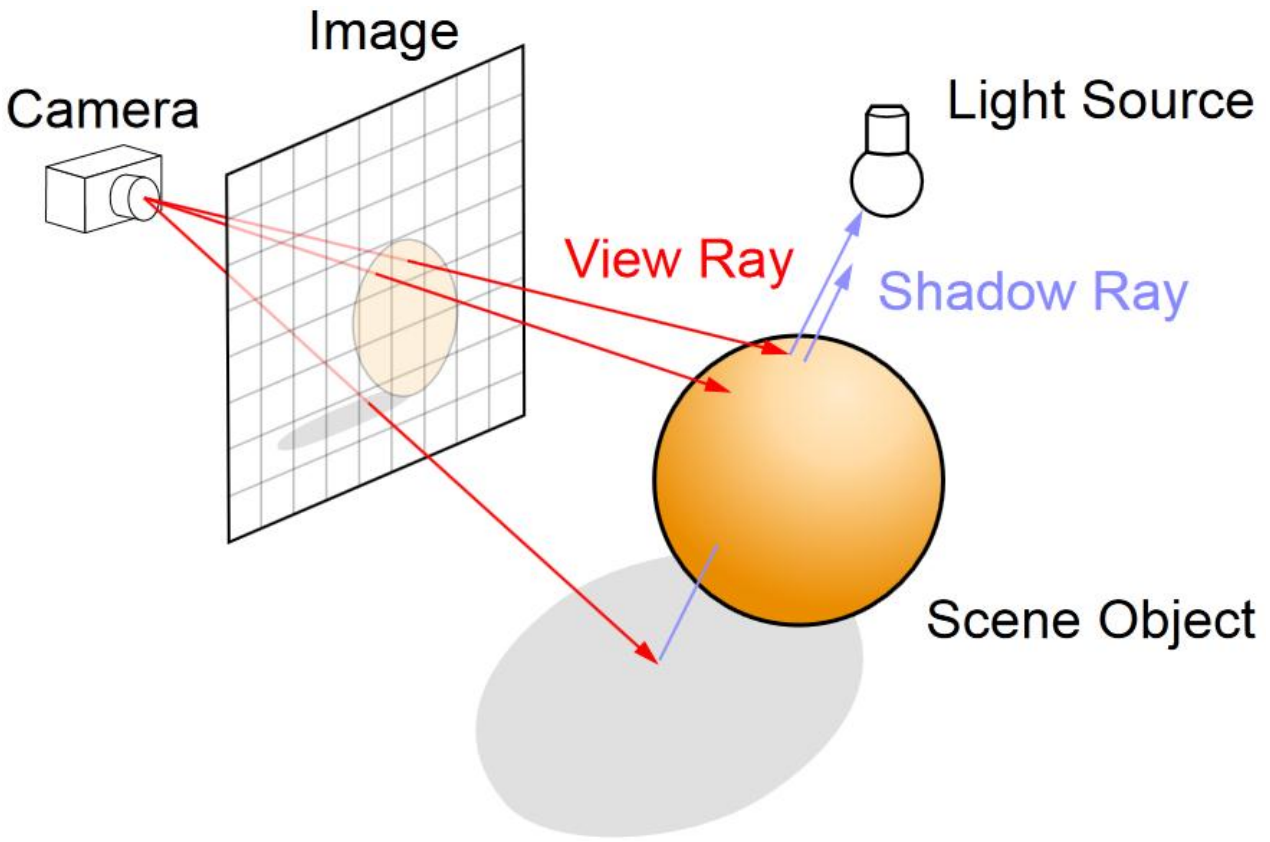
Two Ways to Render an Image



http://upload.wikimedia.org/wikipedia/commons/8/83/Ray_trace_diagram.svg

Ray Tracing

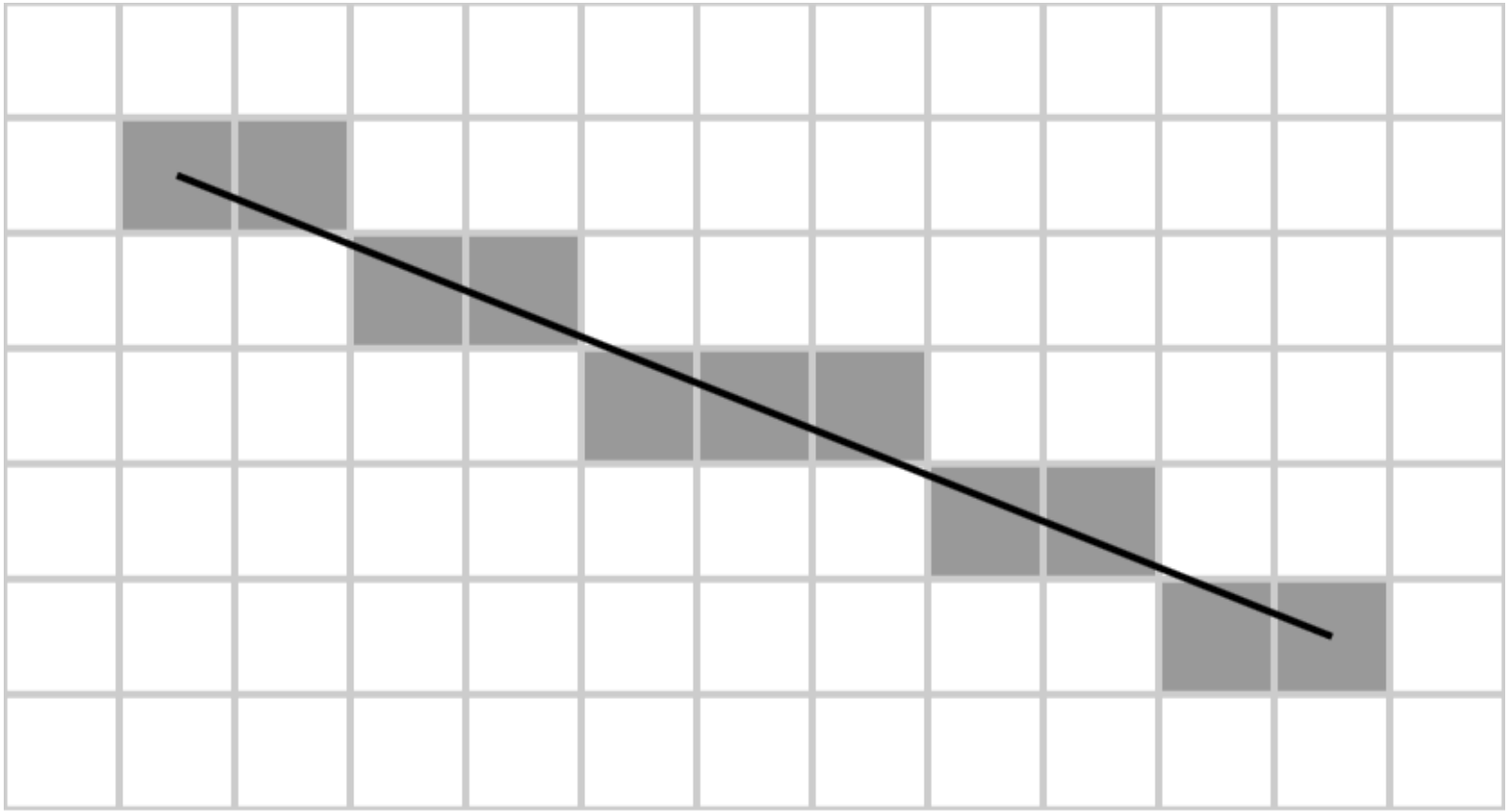
Two Ways to Render an Image



http://upload.wikimedia.org/wikipedia/commons/8/83/Ray_trace_diagram.svg

Ray Tracing

For Today



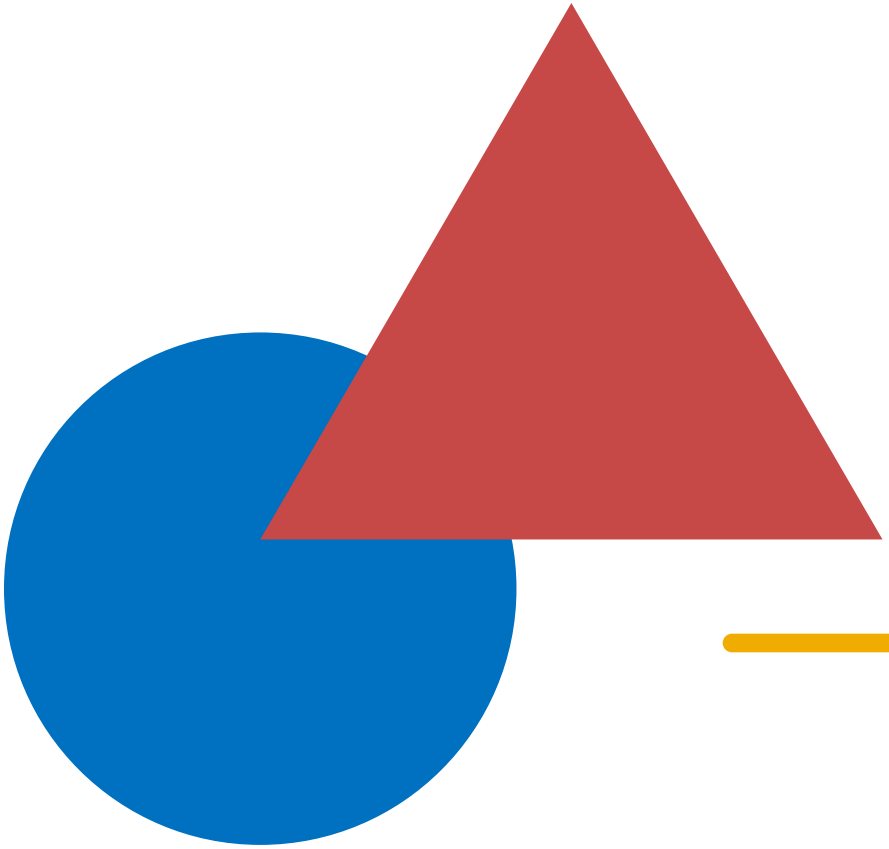
http://upload.wikimedia.org/wikipedia/commons/a/ab/Bresenham.svg

Rasterization

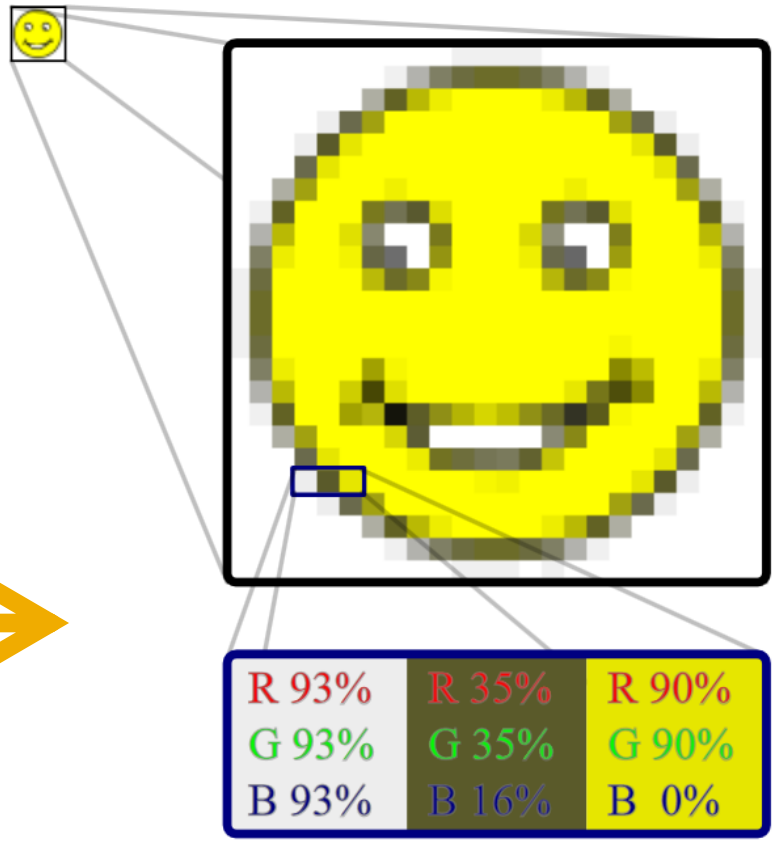
Rasterize [rastərʌɪz]:

To convert vector data to raster format.

Rasterization Process

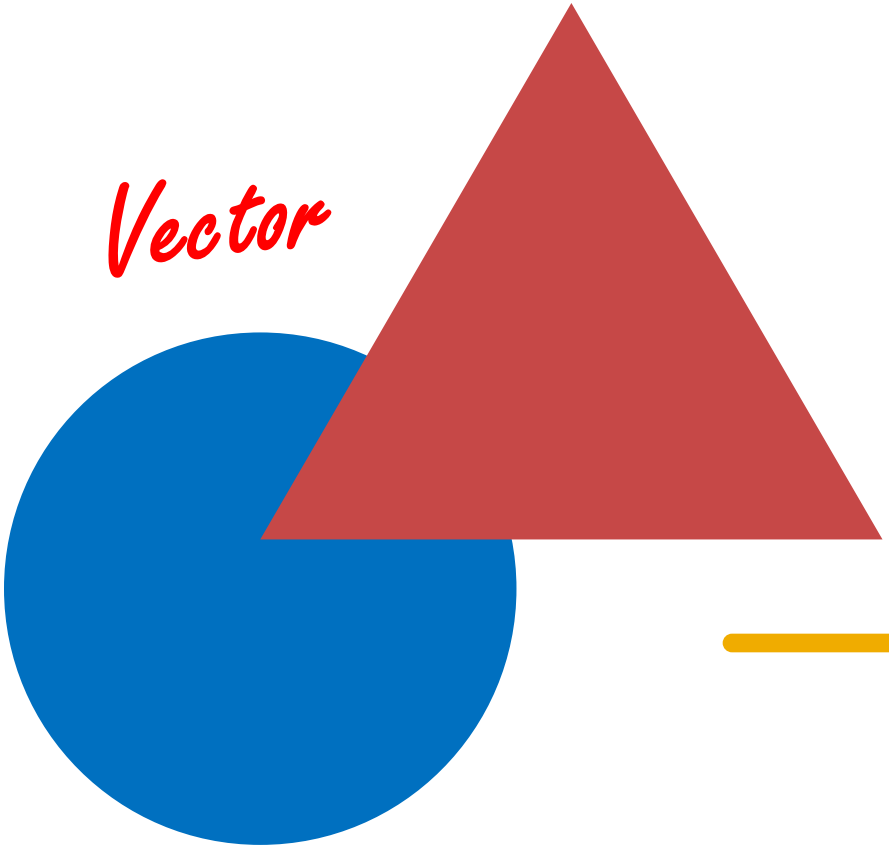


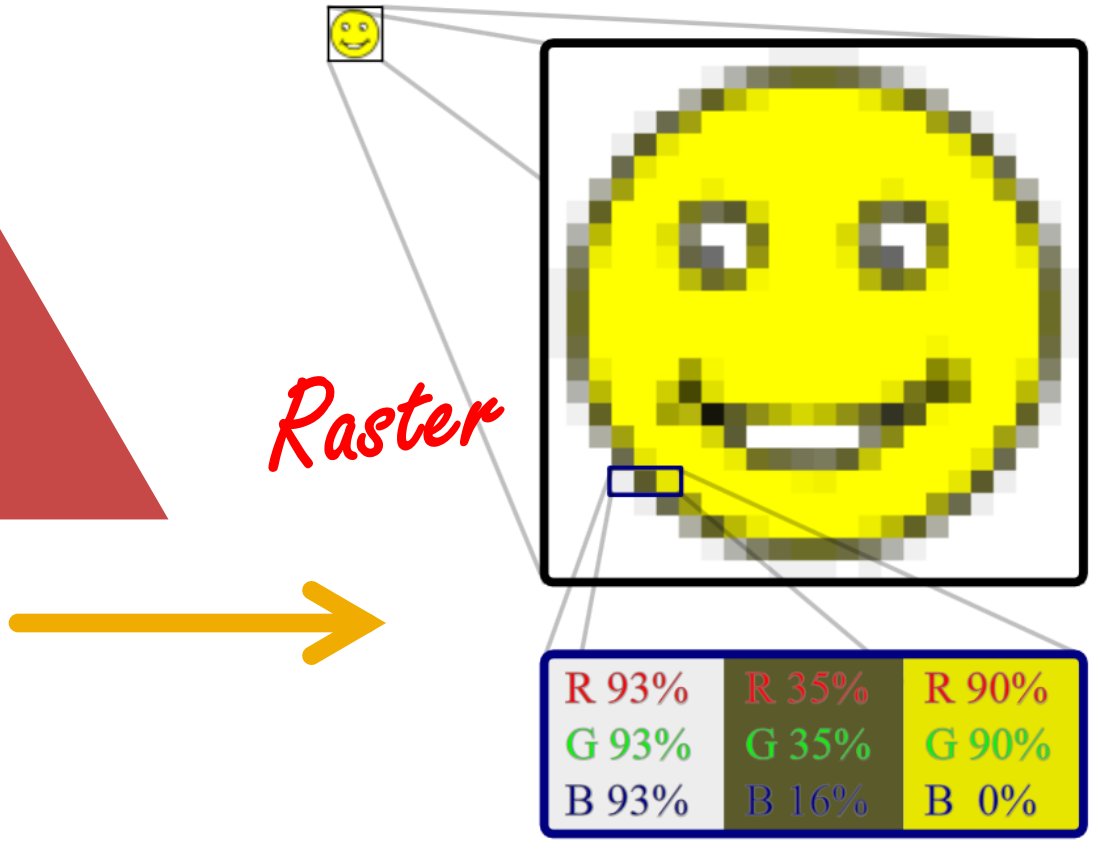
"A triangle is here, a circle is there, …"



"This pixel is yellow…"

Rasterization Process





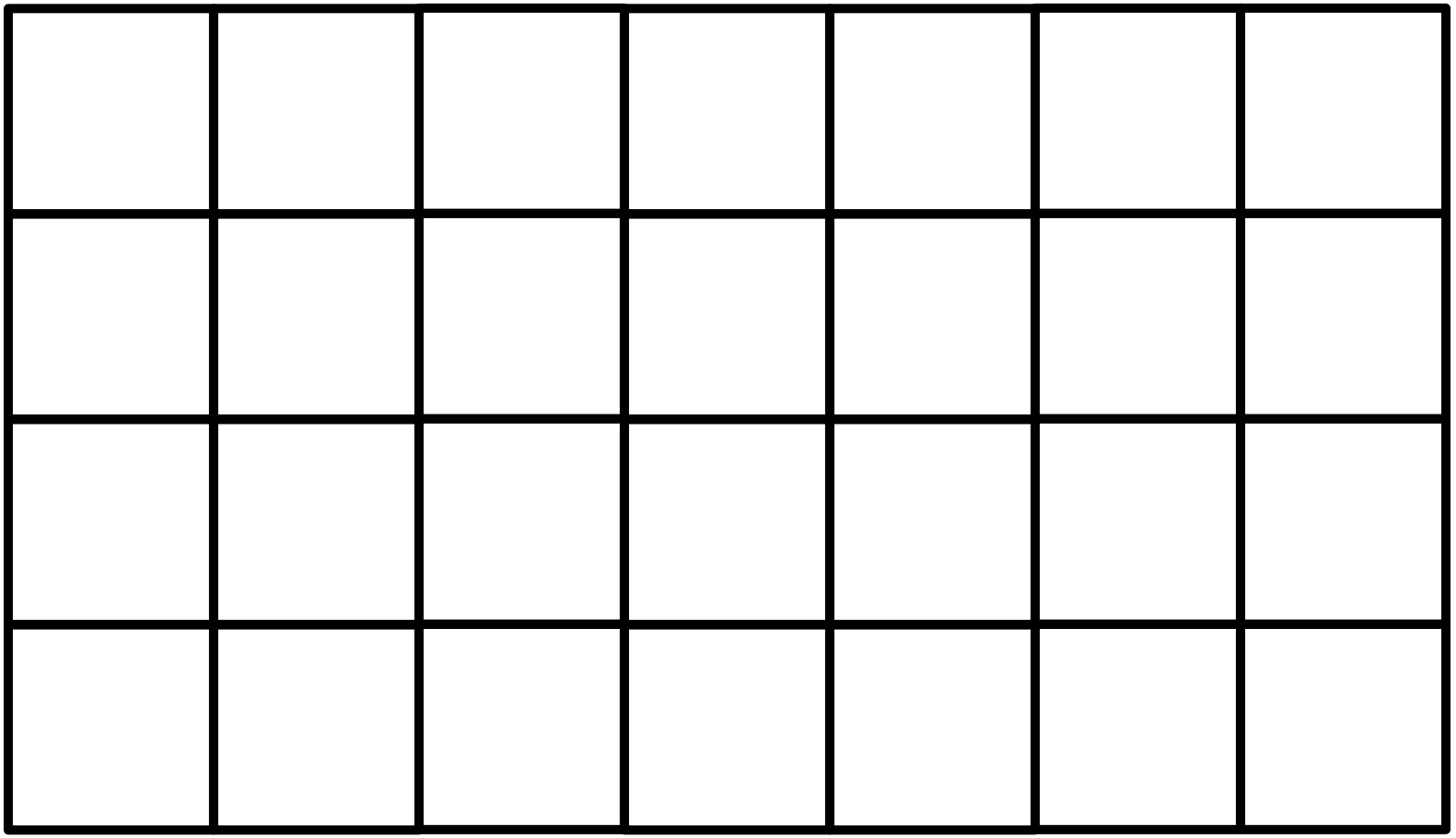
"A triangle is here, a circle is there, …"

"This pixel is yellow…"

Scan Conversion

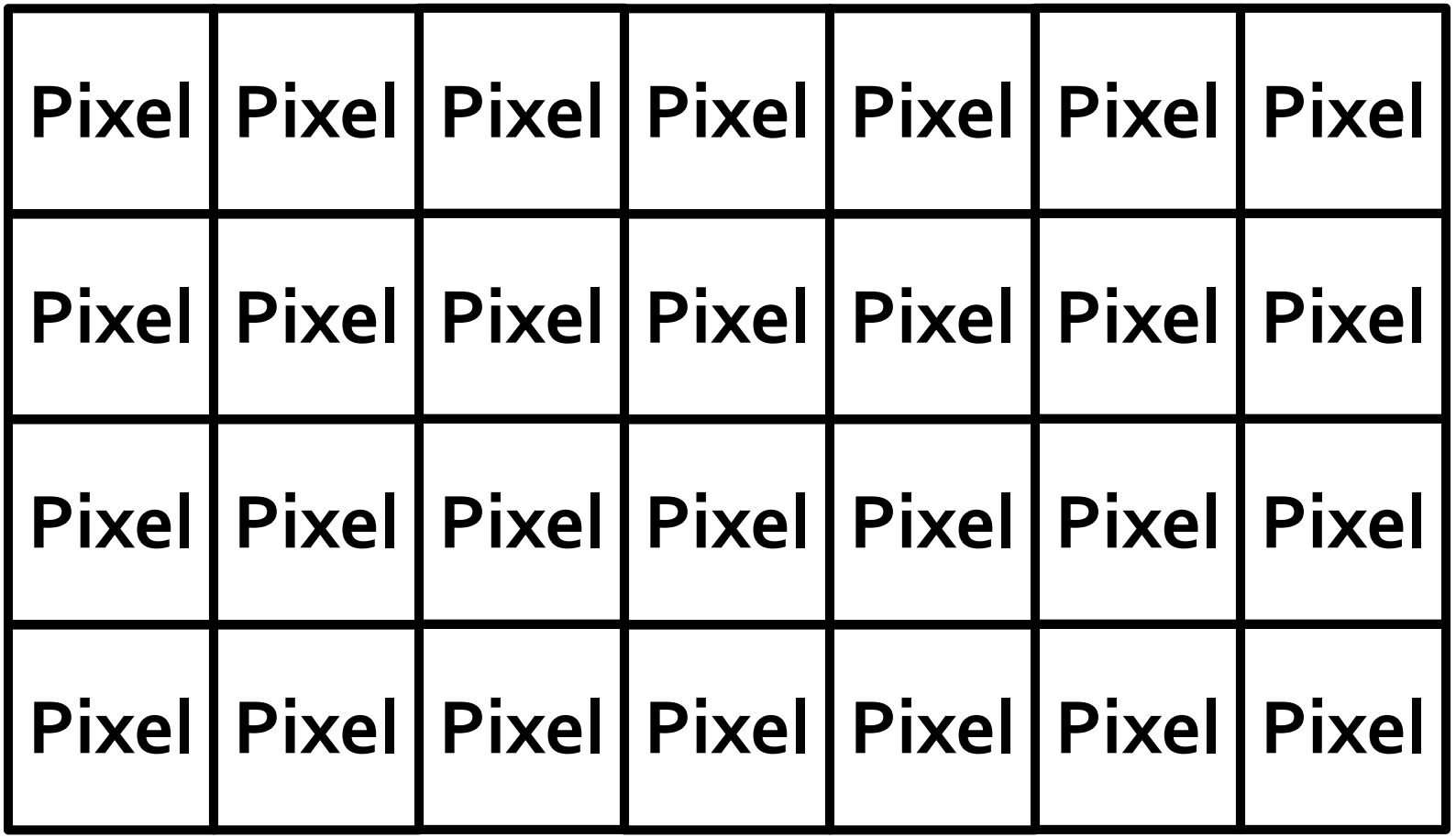
Figuring out which pixels to shade.

The Basics



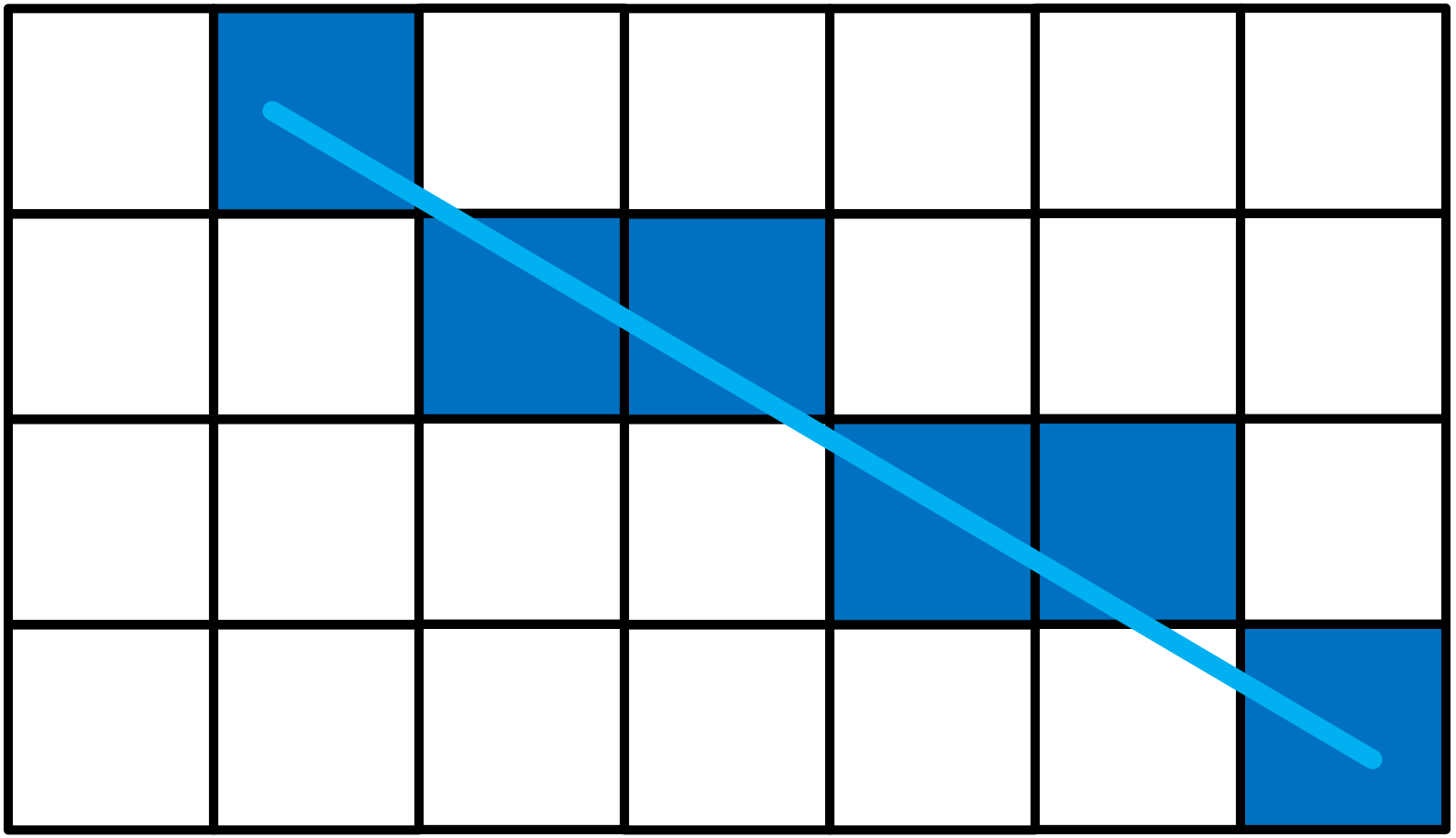
Framebuffer

The Basics



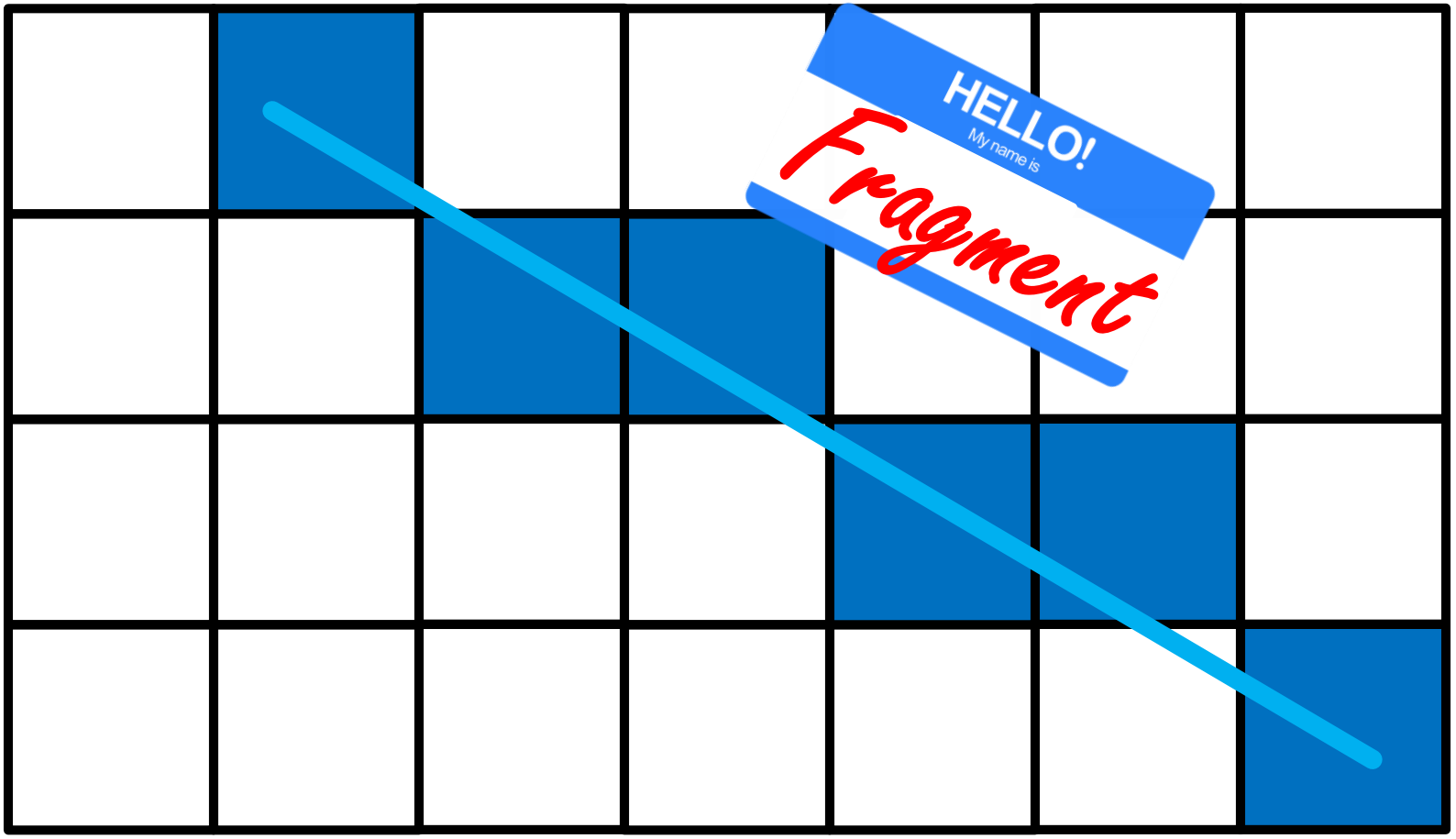
Framebuffer

The Basics



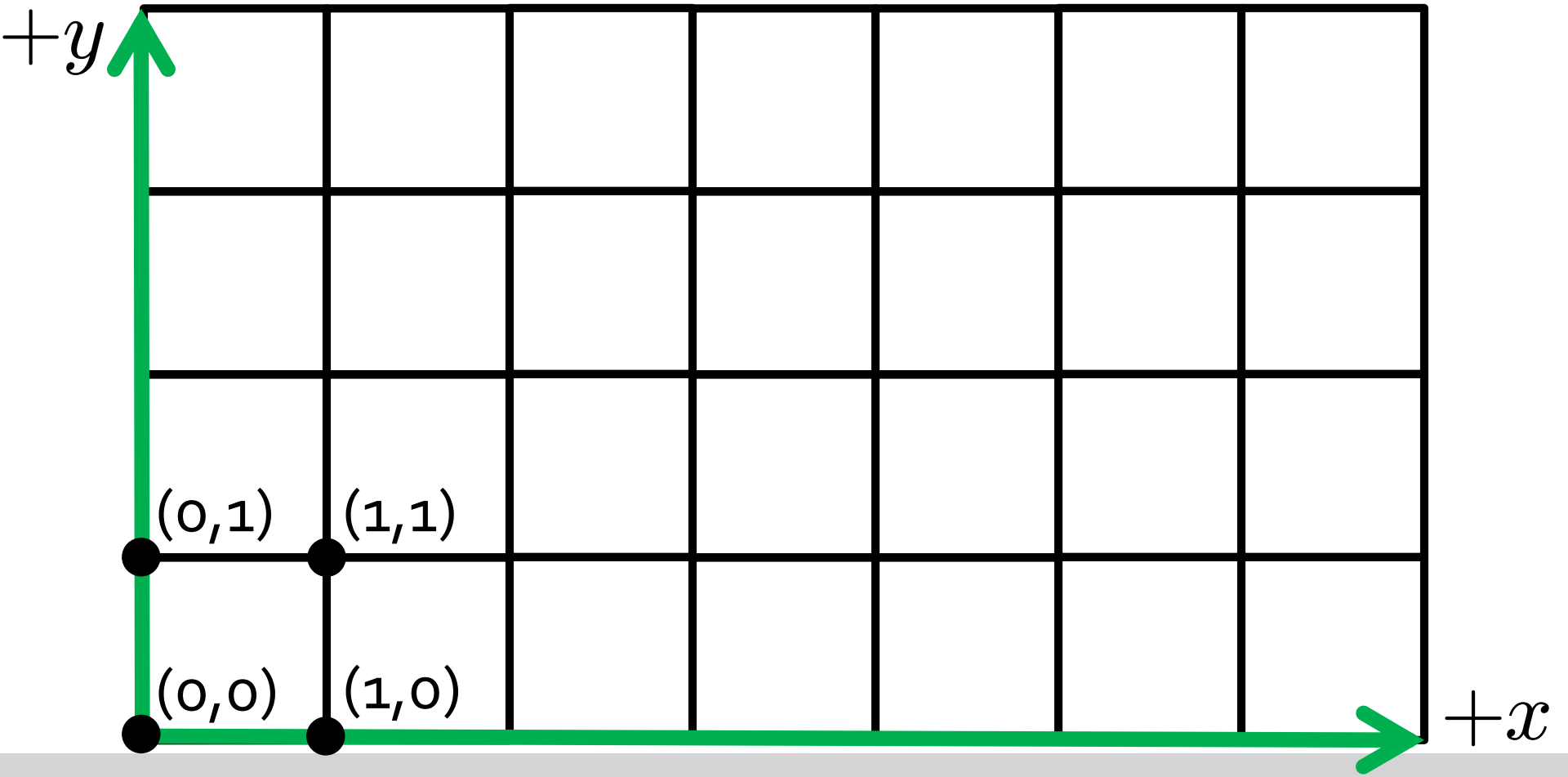
Scan Conversion

The Basics



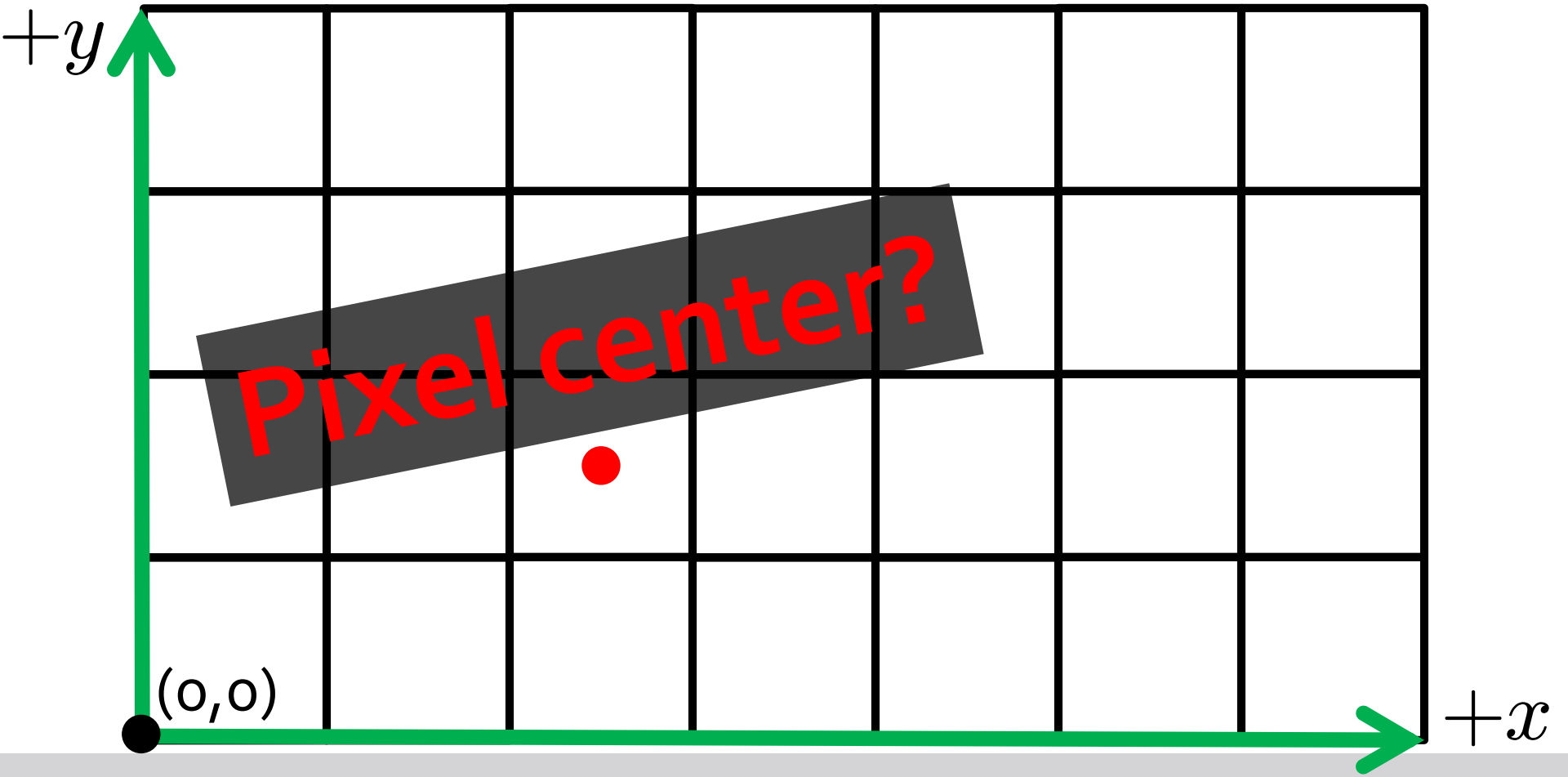
Scan Conversion

The Basics



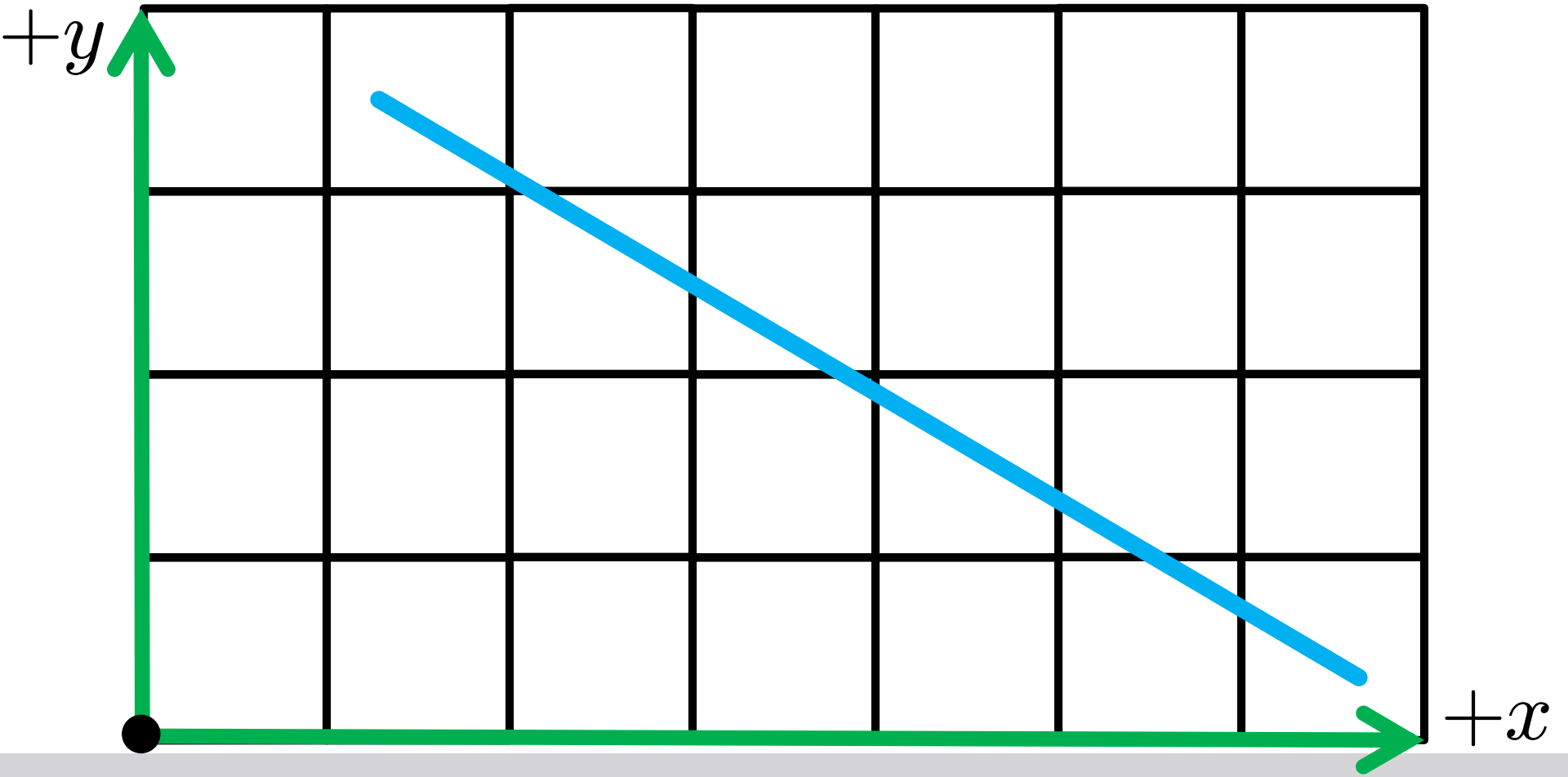
Framebuffer coordinates

The Basics



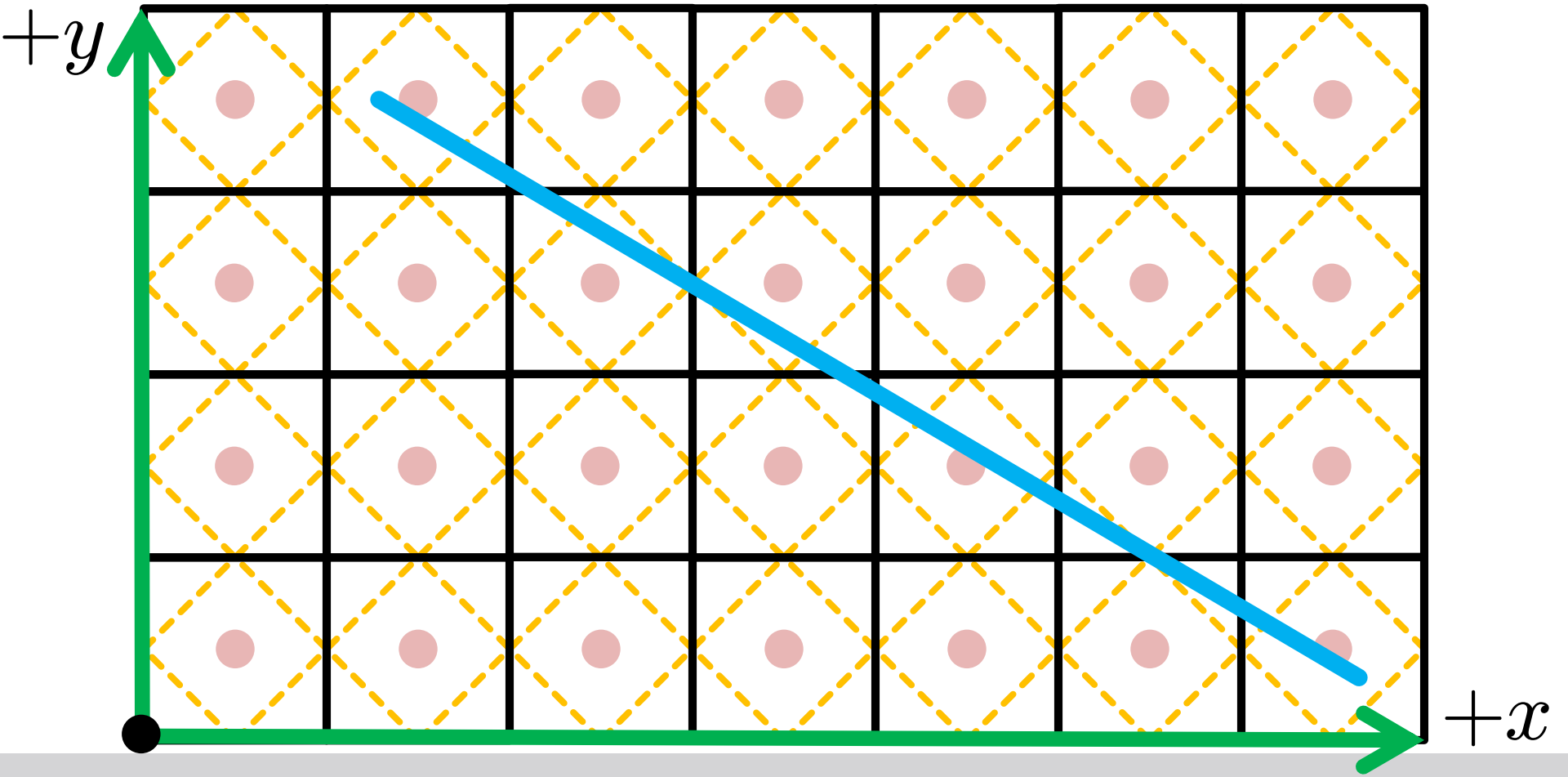
Framebuffer coordinates

Problem



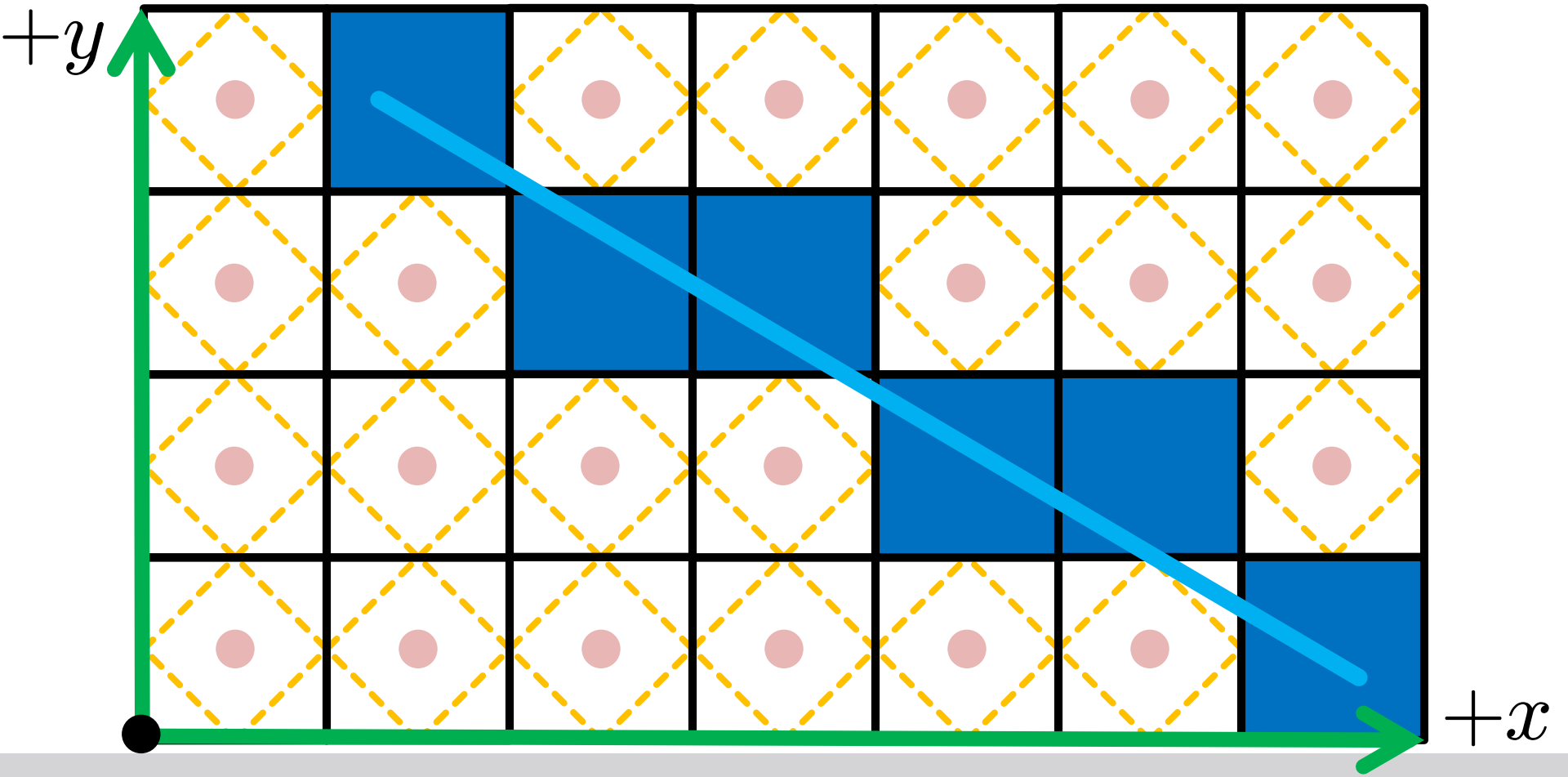
Where do lines go?

Convention



Must intersect diamond

Convention



Must intersect diamond

Desirable Properties

- Fast
- Simple
- Integer arithmetic

Bresenham's Algorithm

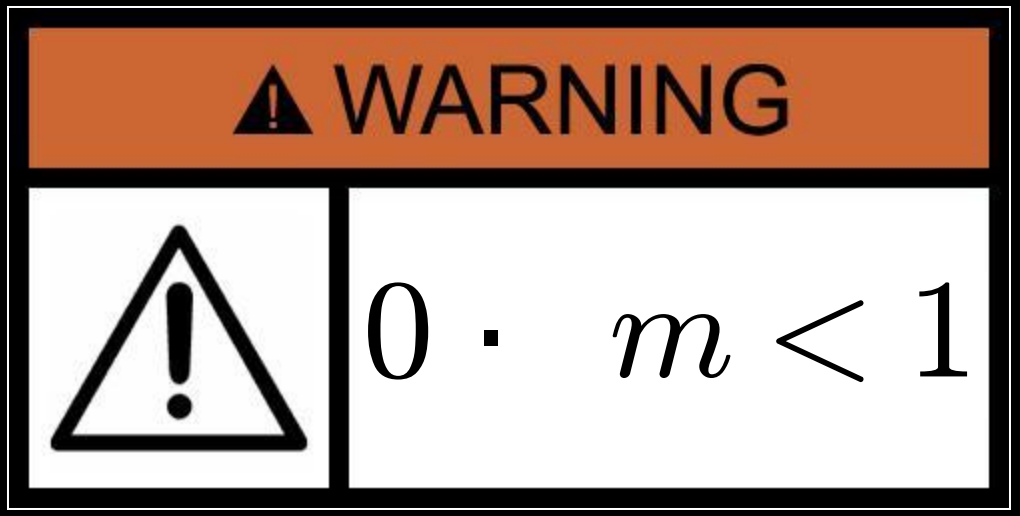
- Introduced in 1967
- Best-fit approximation under some conditions

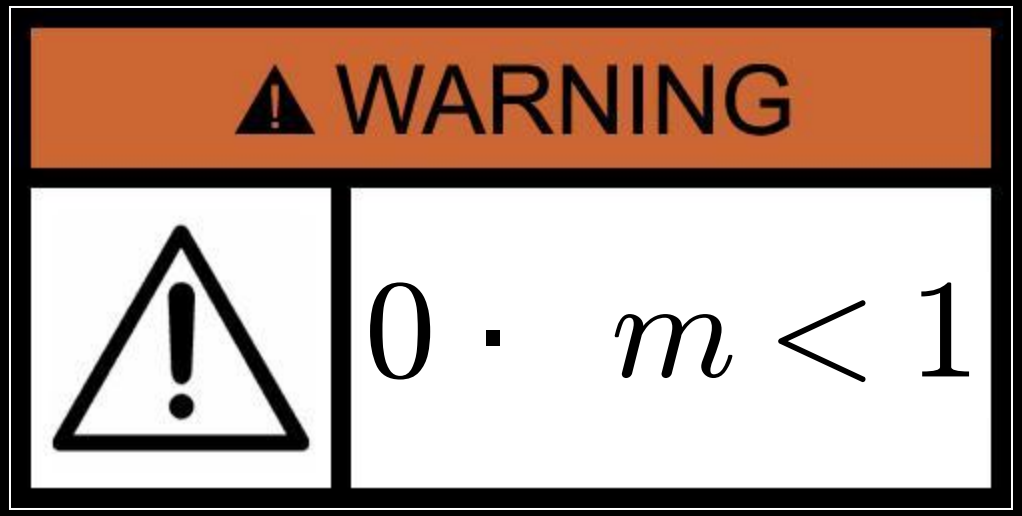
Bresenham's Algorithm

- Introduced in 1967

Variation by Pitteway (1967): "Midpoint Algorithm"

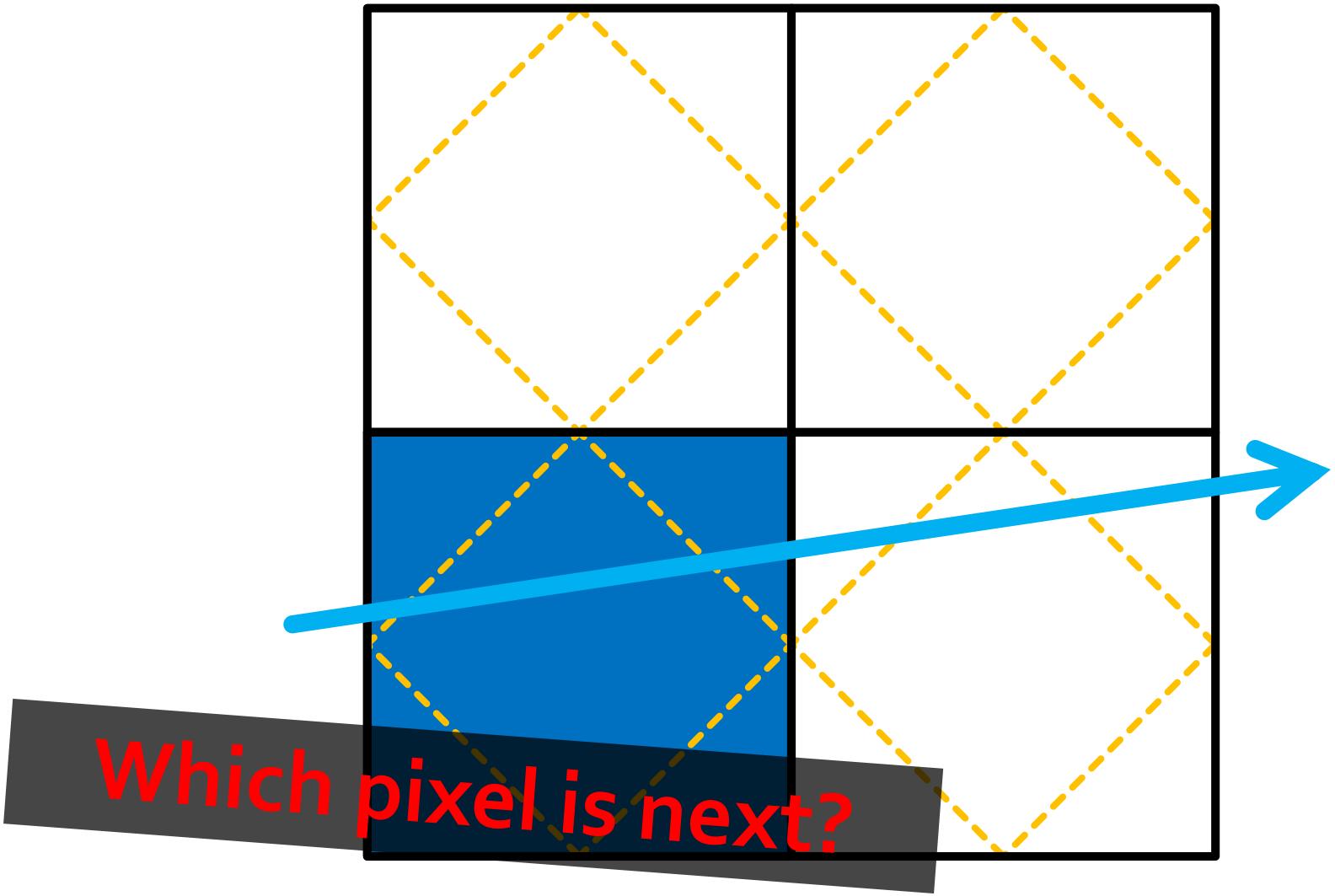
- Best-fit approximation under some conditions


⚠ WARNING
$0 \cdot \quad m < 1$

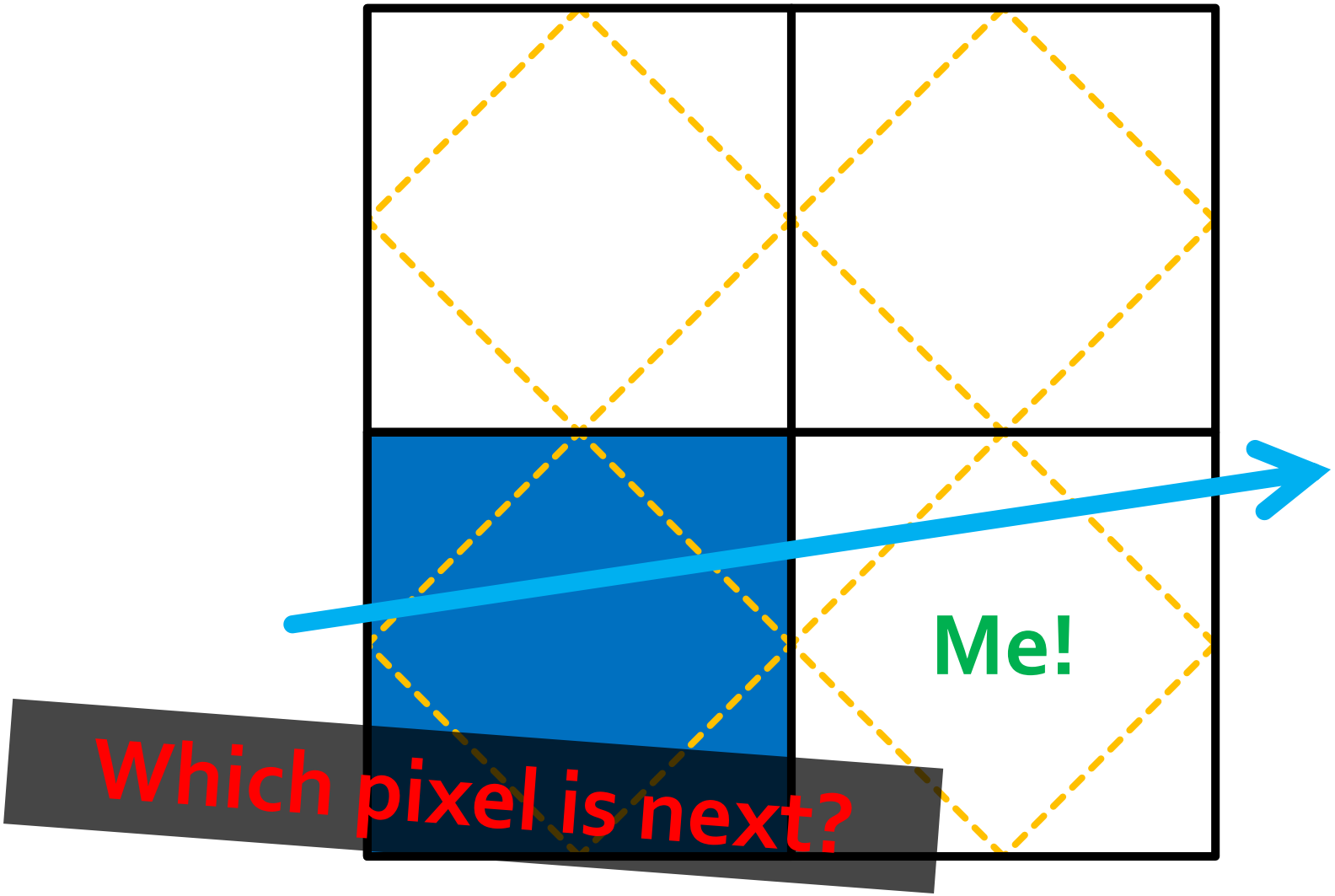


Easy fix via rotation/reflection

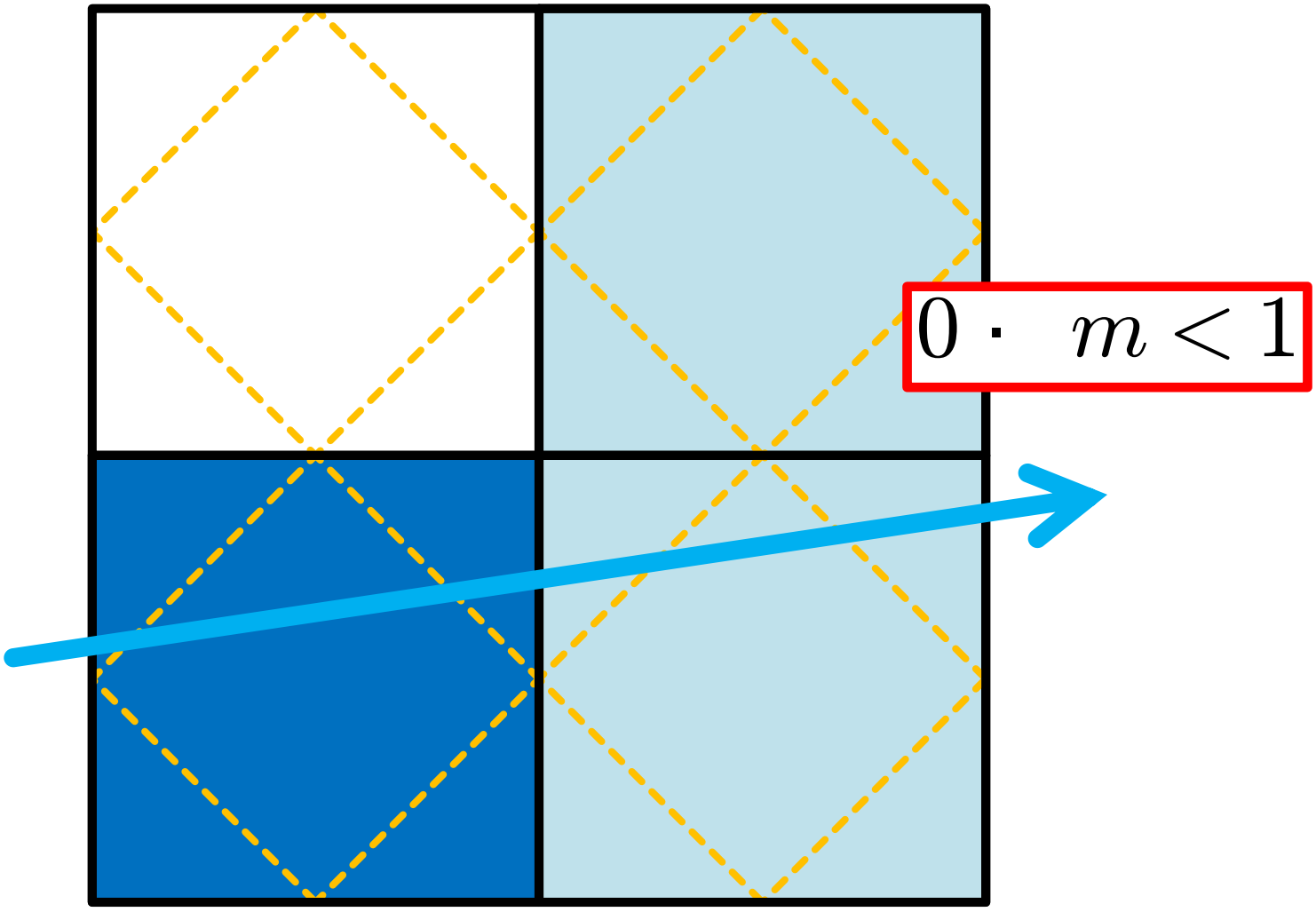
Bresenham's Algorithm



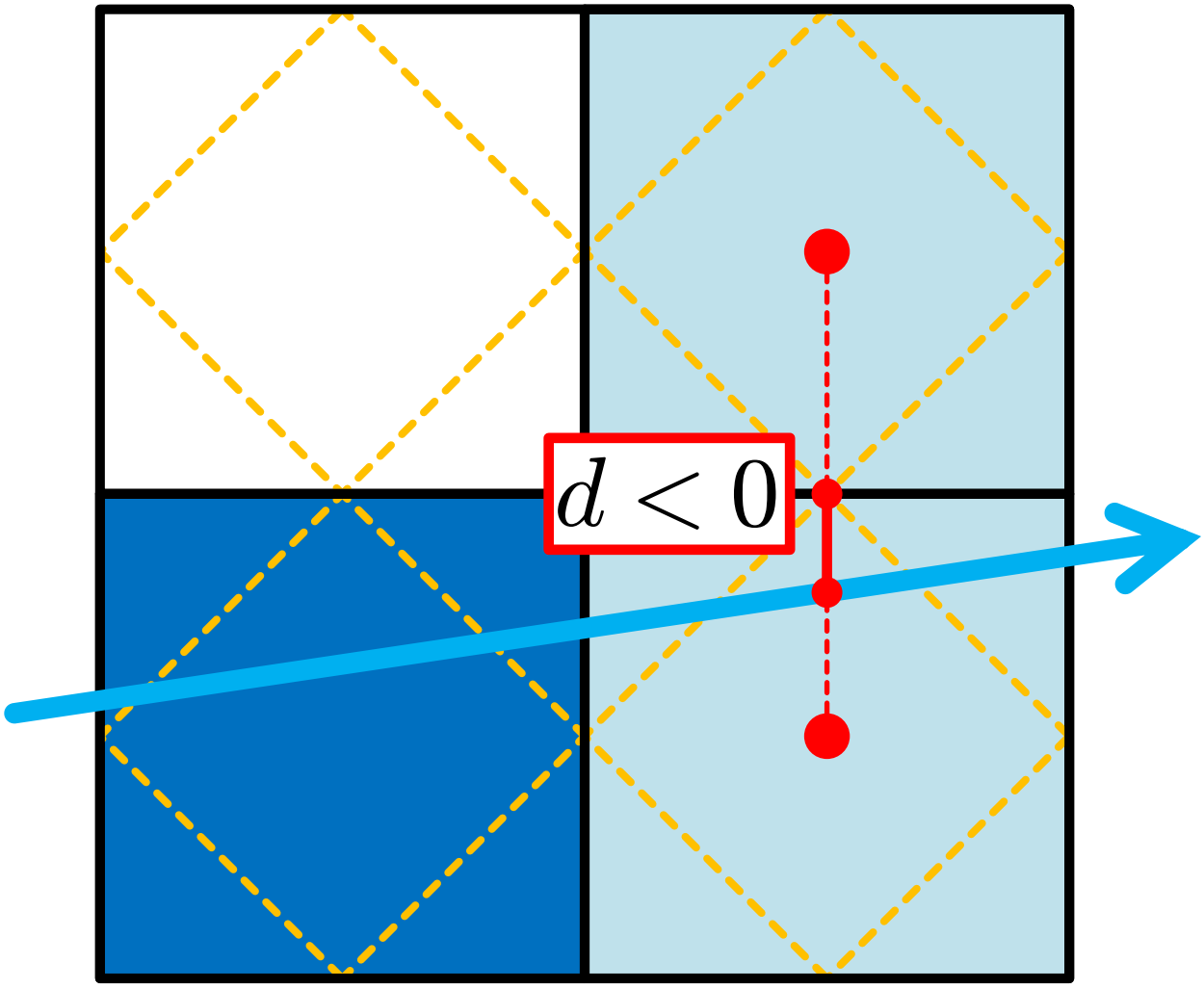
Bresenham's Algorithm



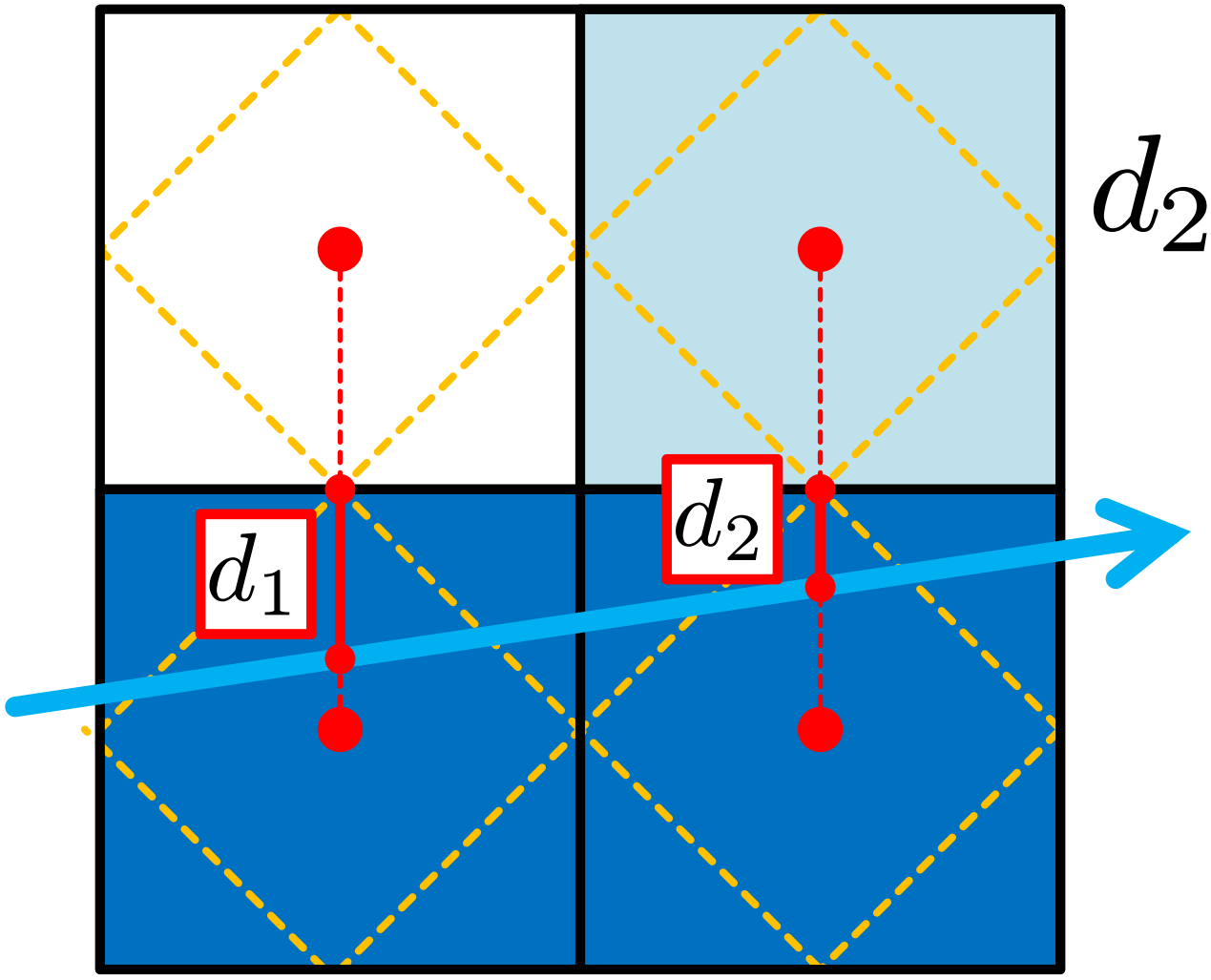
Bresenham's Algorithm



Bresenham's Algorithm

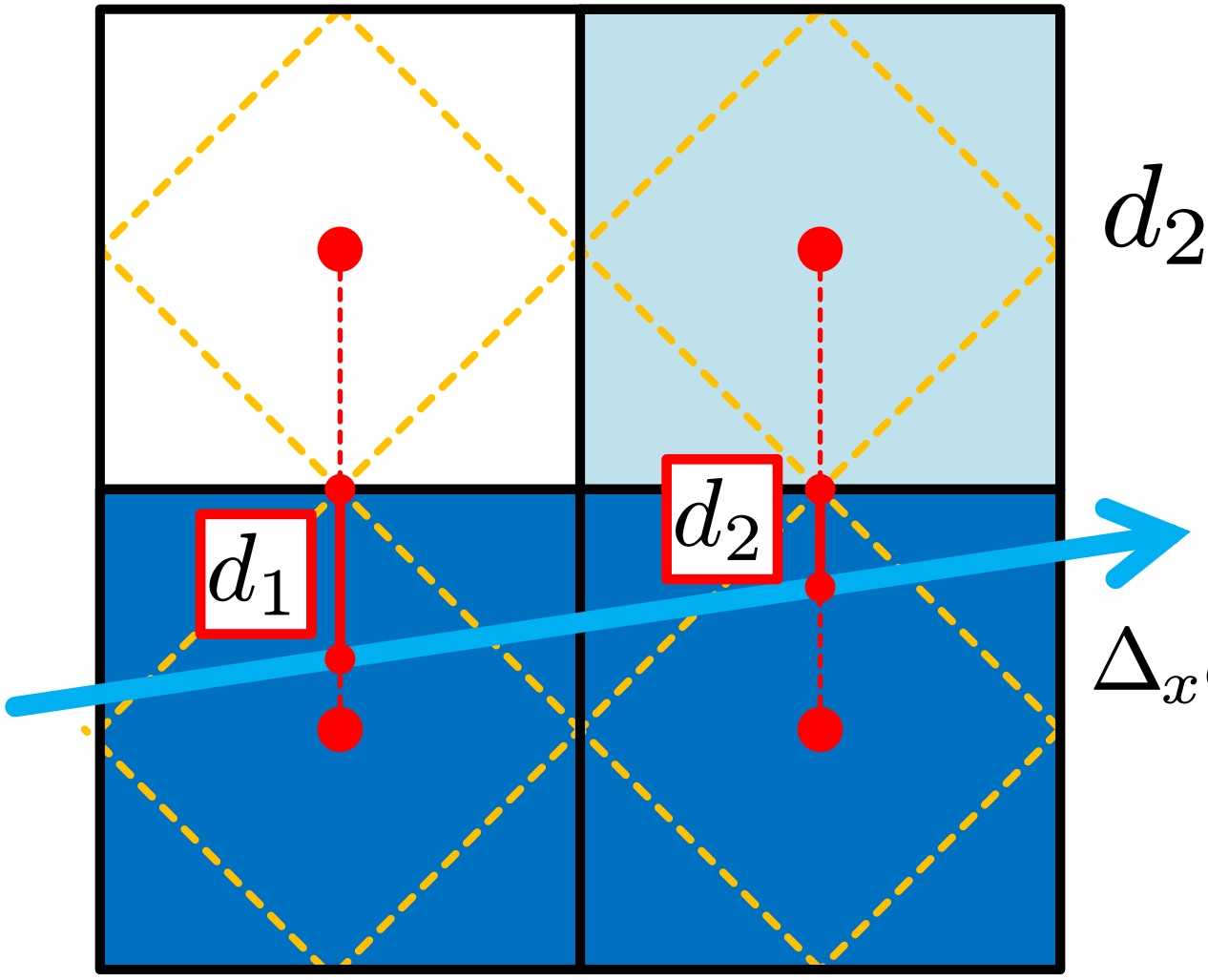


Bresenham's Algorithm



$$d_2 = d_1 + m$$

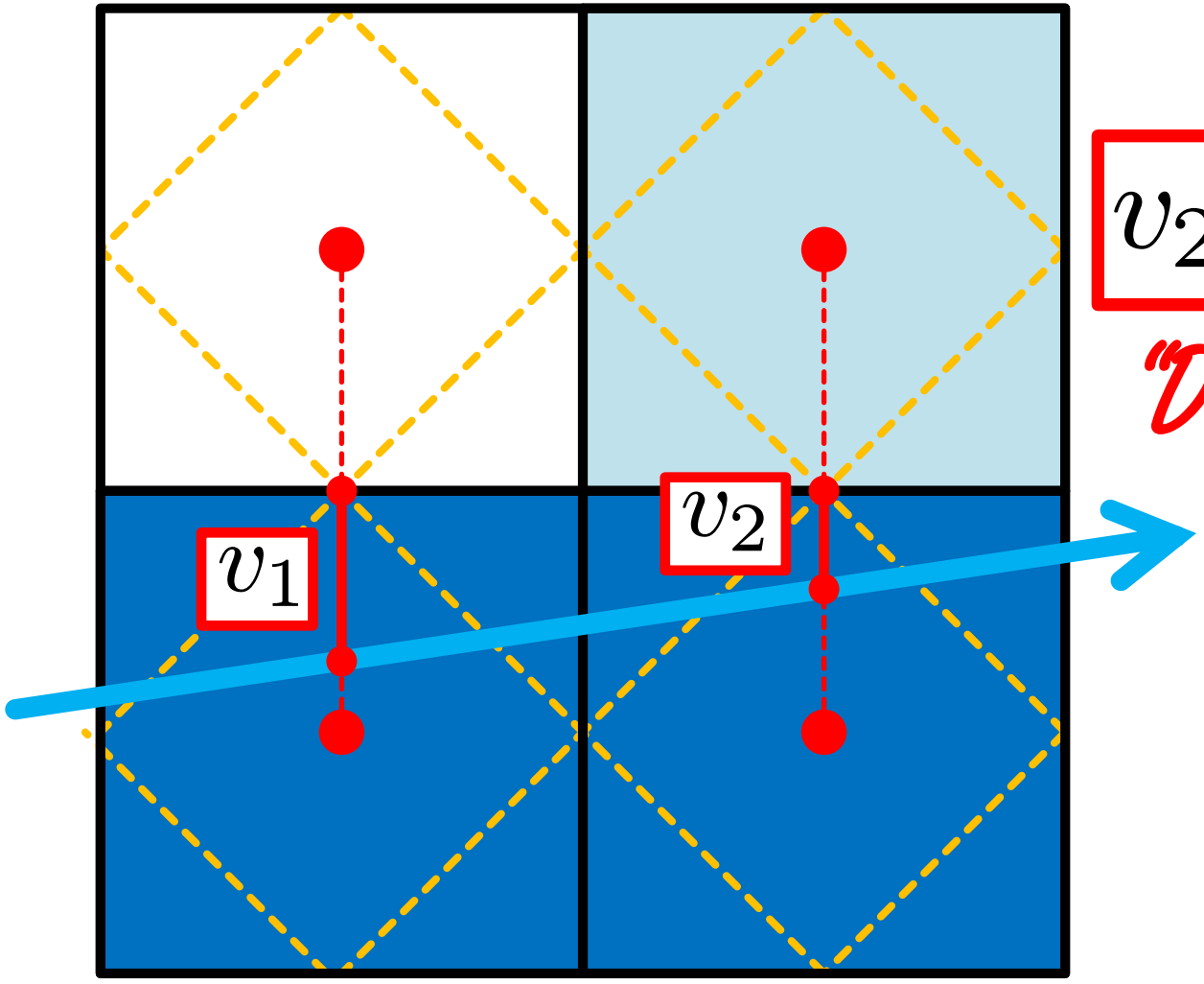
Bresenham's Algorithm



$$d_2 = d_1 + \frac{\Delta_y}{\Delta_x}$$

$$\Delta_x d_2 = \Delta_x d_1 + \Delta_y$$

Bresenham's Algorithm

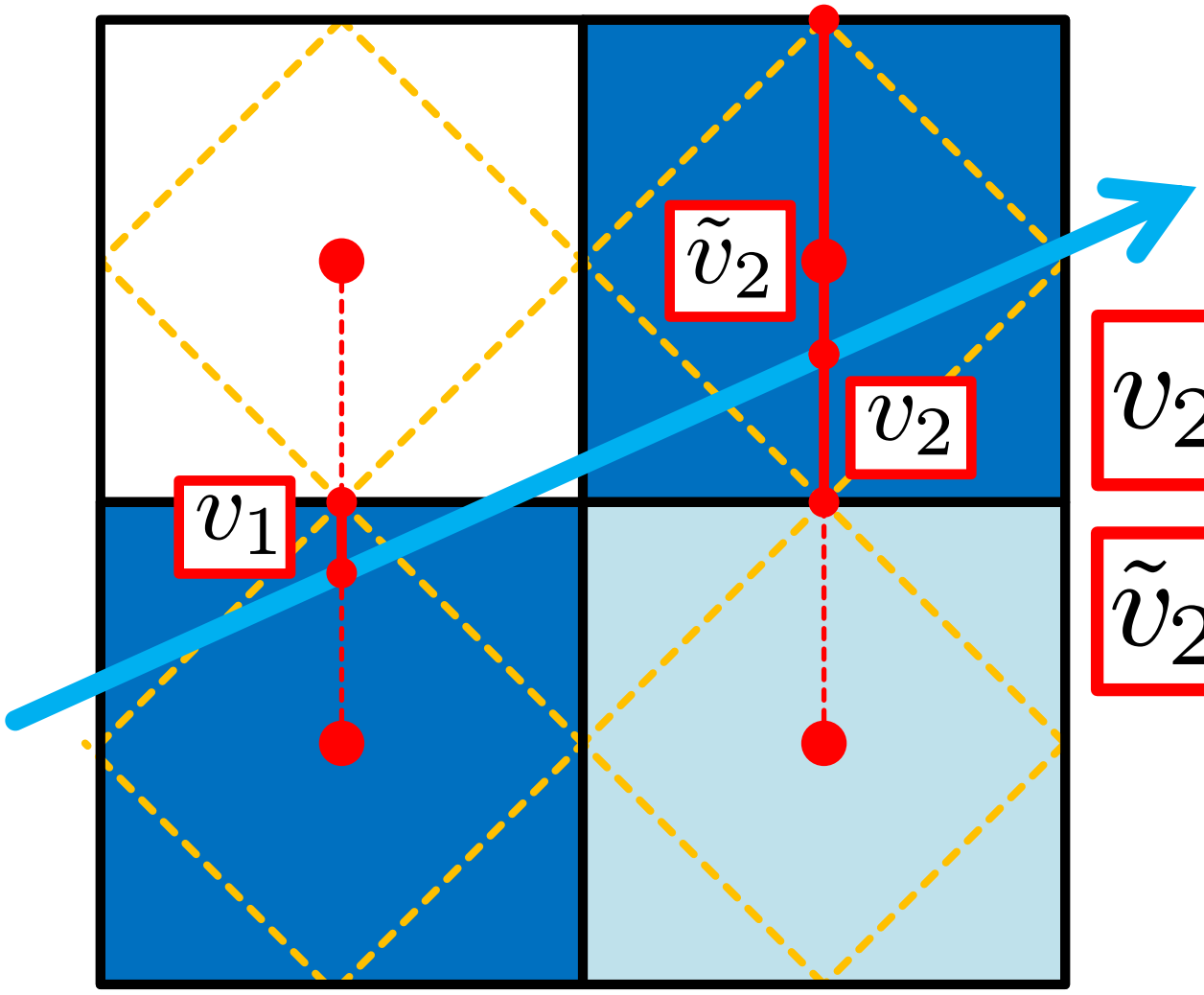


$$v_2 = v_1 + \Delta_y$$

"Decision variable"

$$v = \Delta_x d$$

Bresenham's Algorithm



$$v_2 = v_1 + \Delta_y$$

$$\tilde{v}_2 = v_2 - \Delta_x$$

Bresenham Iteration

Given v_1:

1. $v_2 = v_1 + \Delta_y$
2. If $v_2 \cdot \ 0$, move east.
3. If $v_2 > 0$, replace with $\tilde{v}_2 = v_2 - \Delta_x$ and move northeast.

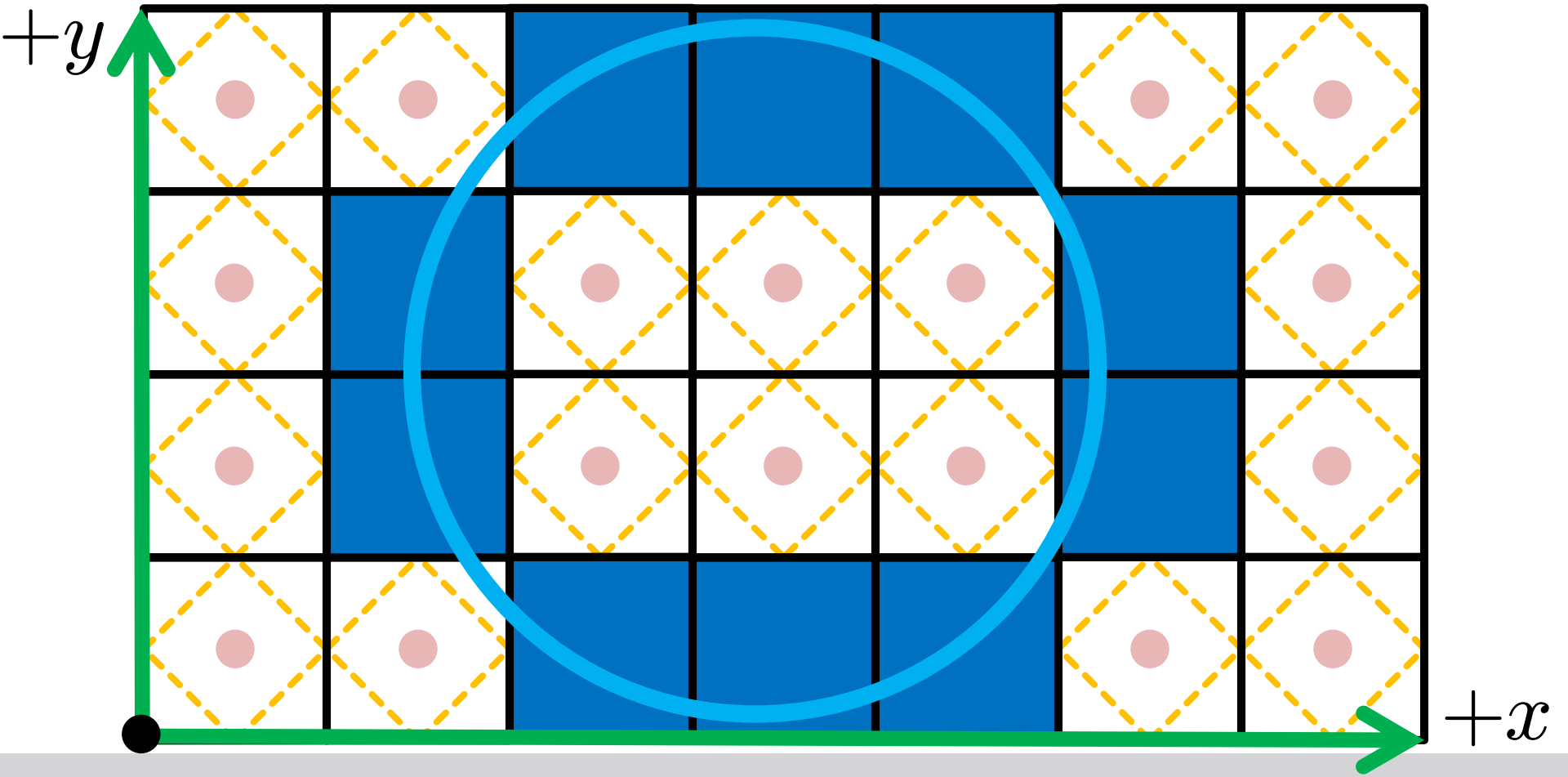
Bresenham Iteration

Given v_1:

1. $v_2 = v_1 + \Delta_y$
2. If $v_2 \cdot \ 0$, move east.
3. If $v_2 > 0$, replace with $\tilde{v}_2 = v_2 - \Delta_x$ and move northeast.

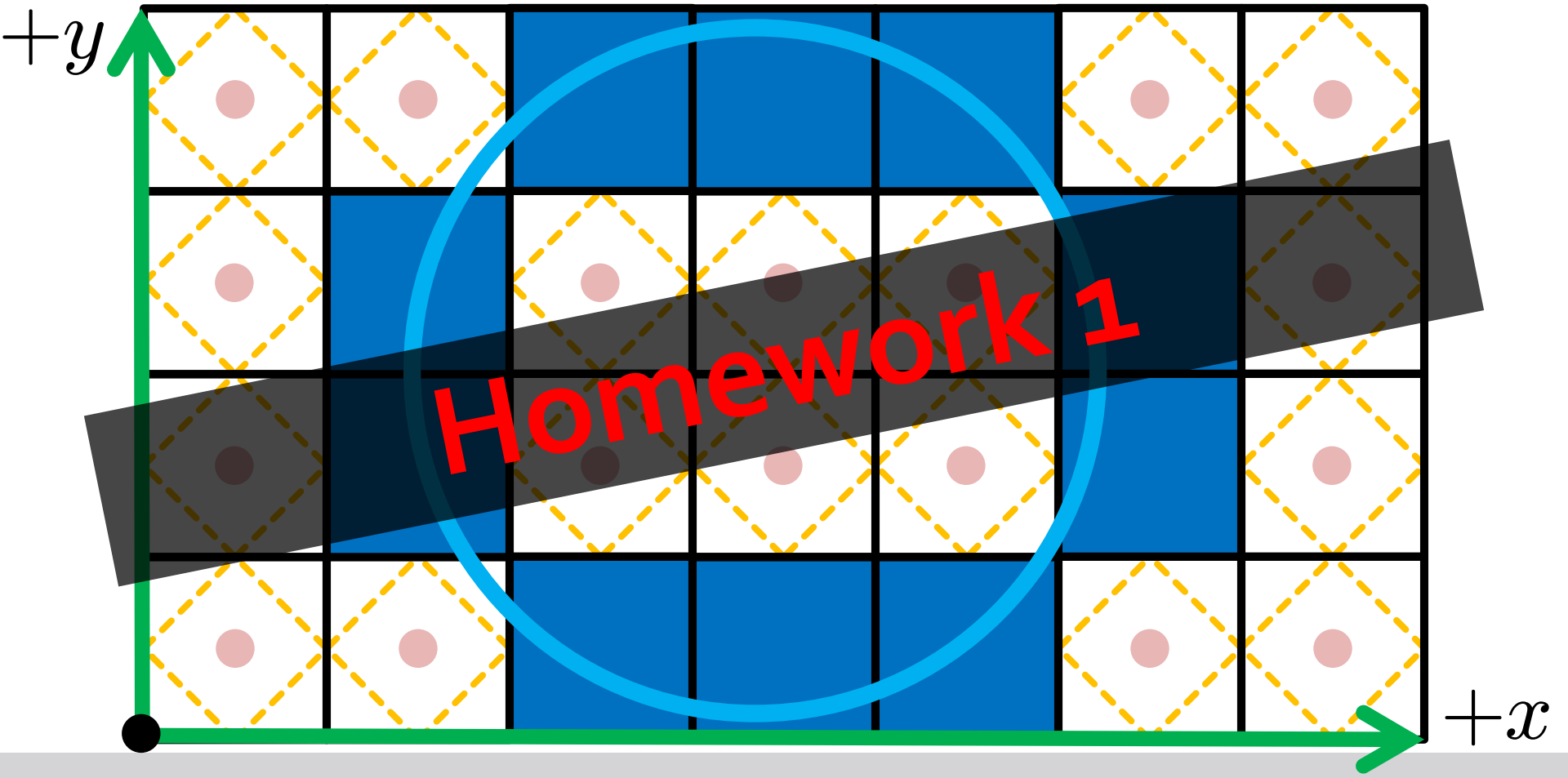
- Fast
- Simple
- Integer arithmetic

Interesting Fact



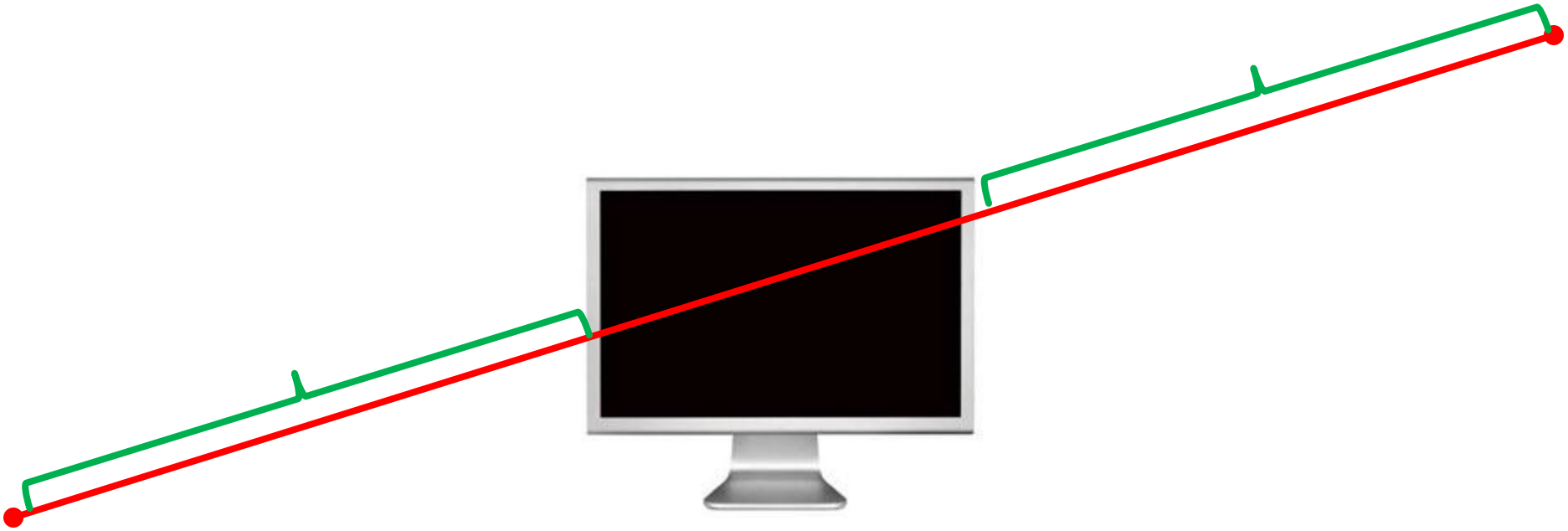
Works (with modification)

Interesting Fact



Works (with modification)

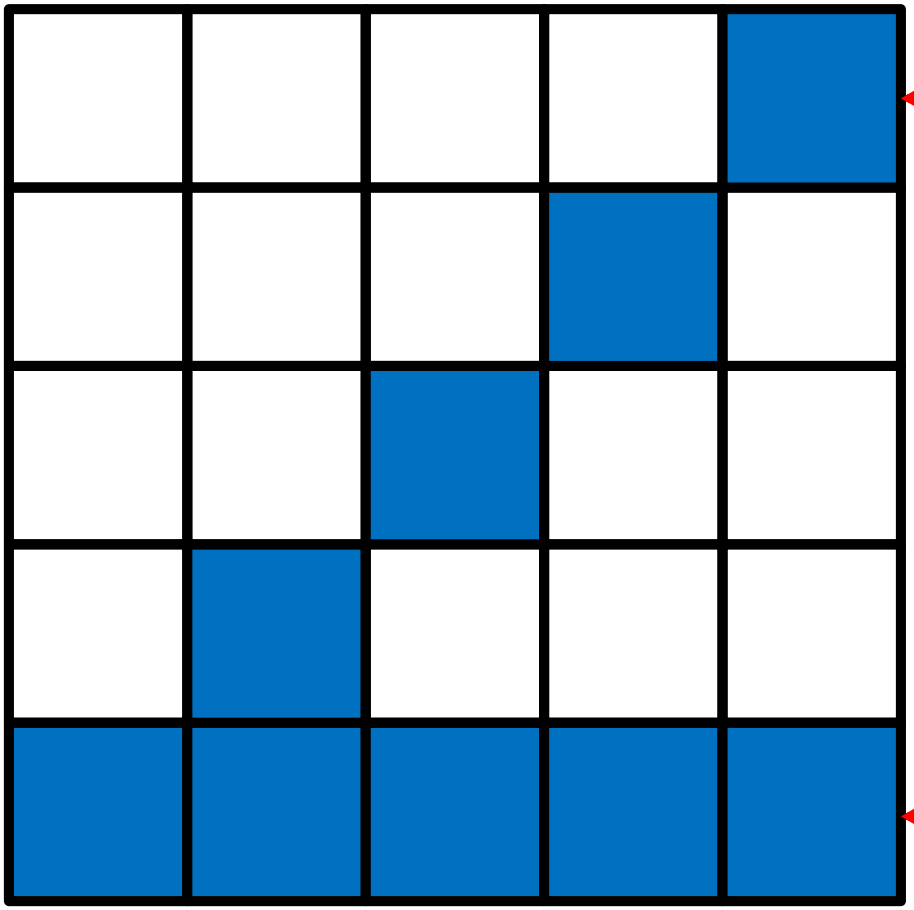
Important Issues We've Ignored



http://www.cheap-computermonitors.com/images/1-cheap-flat-screen-computer-monitors.jpg

Clipping

Important Issues We've Ignored



$l \approx 7.07$

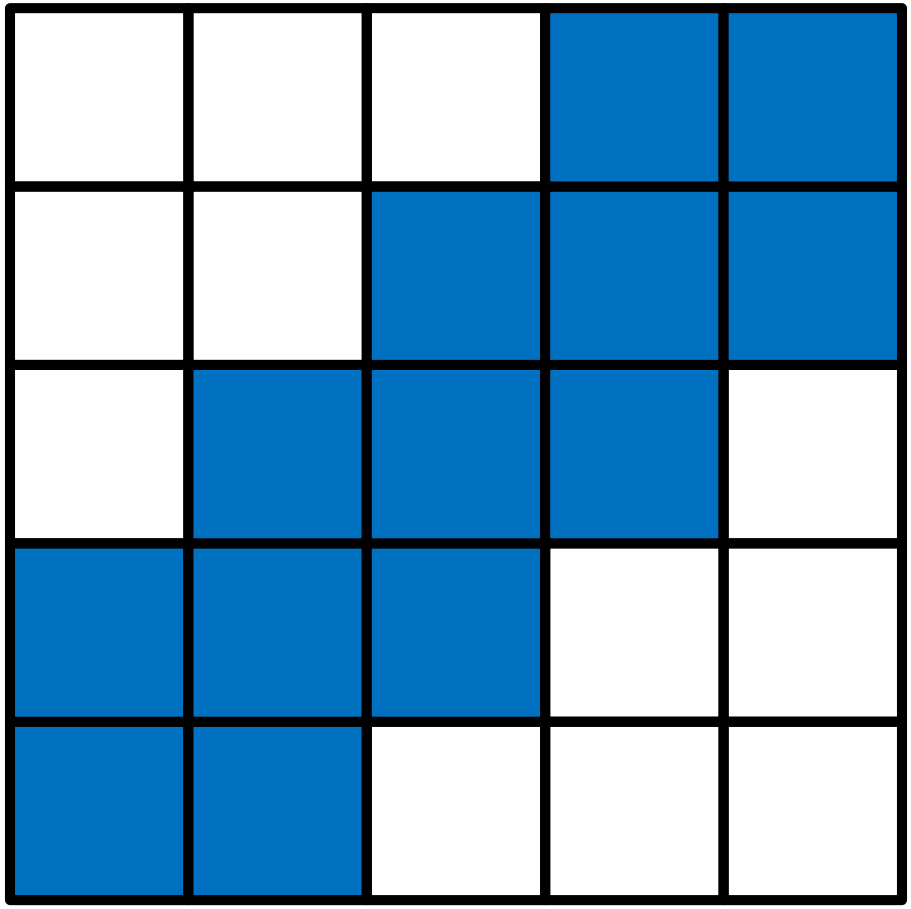
$p = 5$

$l = 5$

$p = 5$

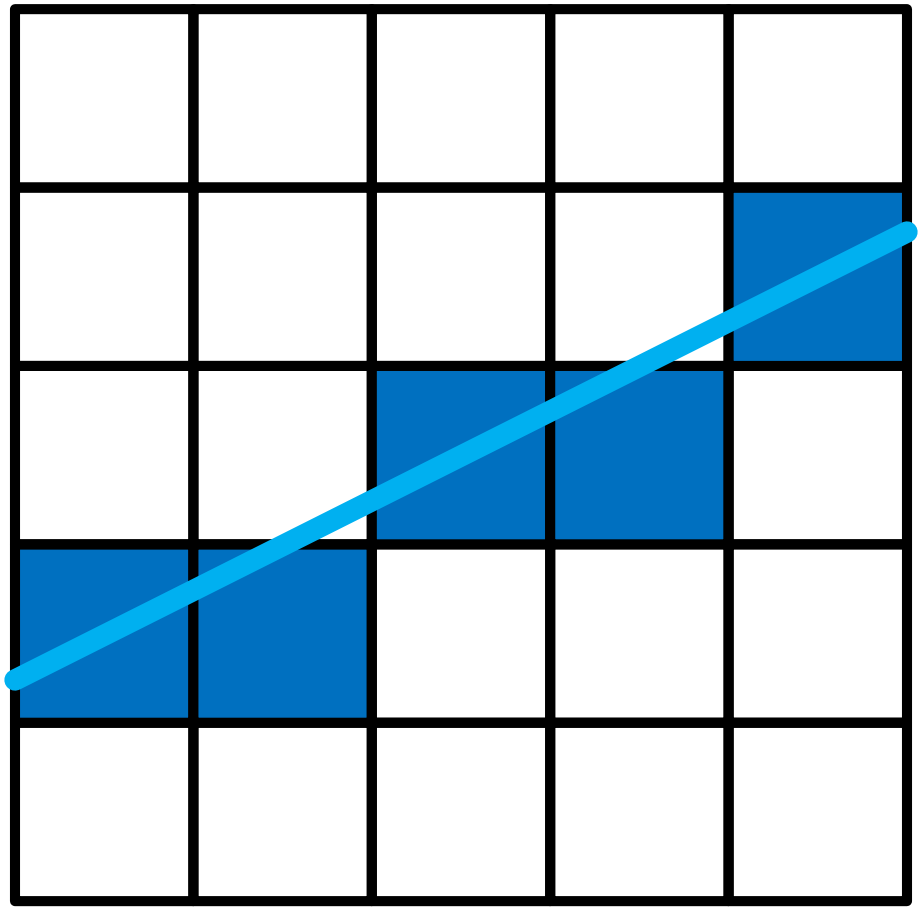
Line intensity

Important Issues We've Ignored



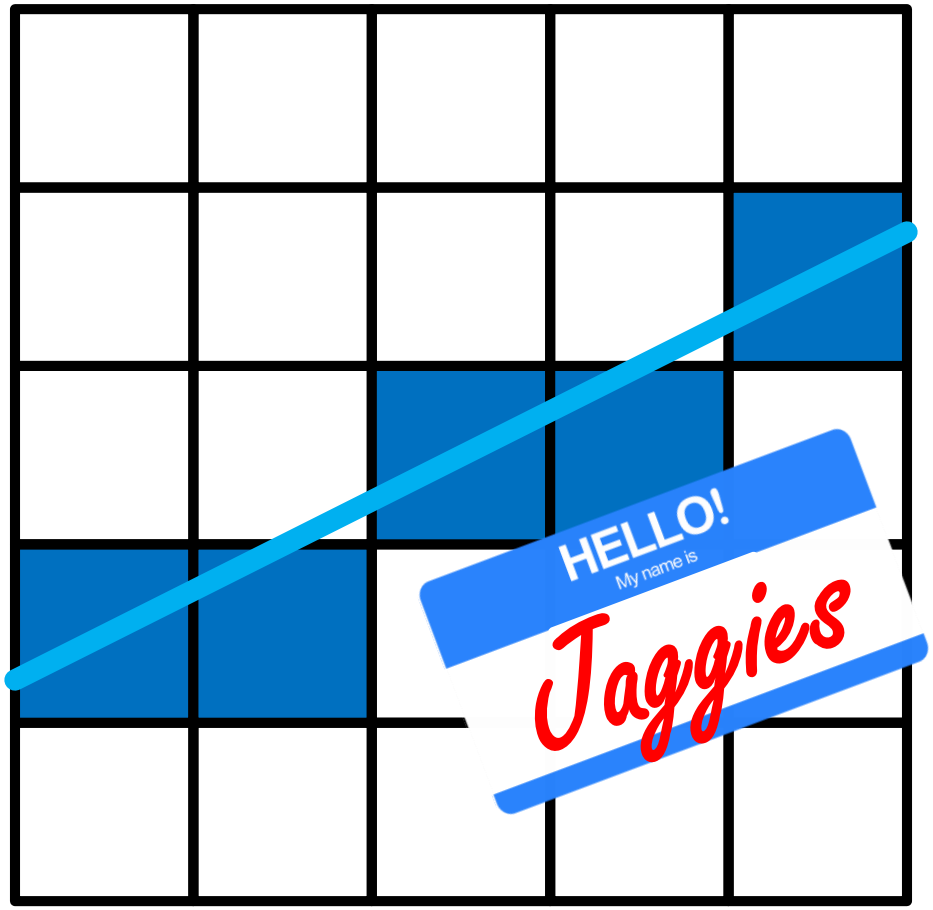
Line thickness

Important Issues We've Ignored



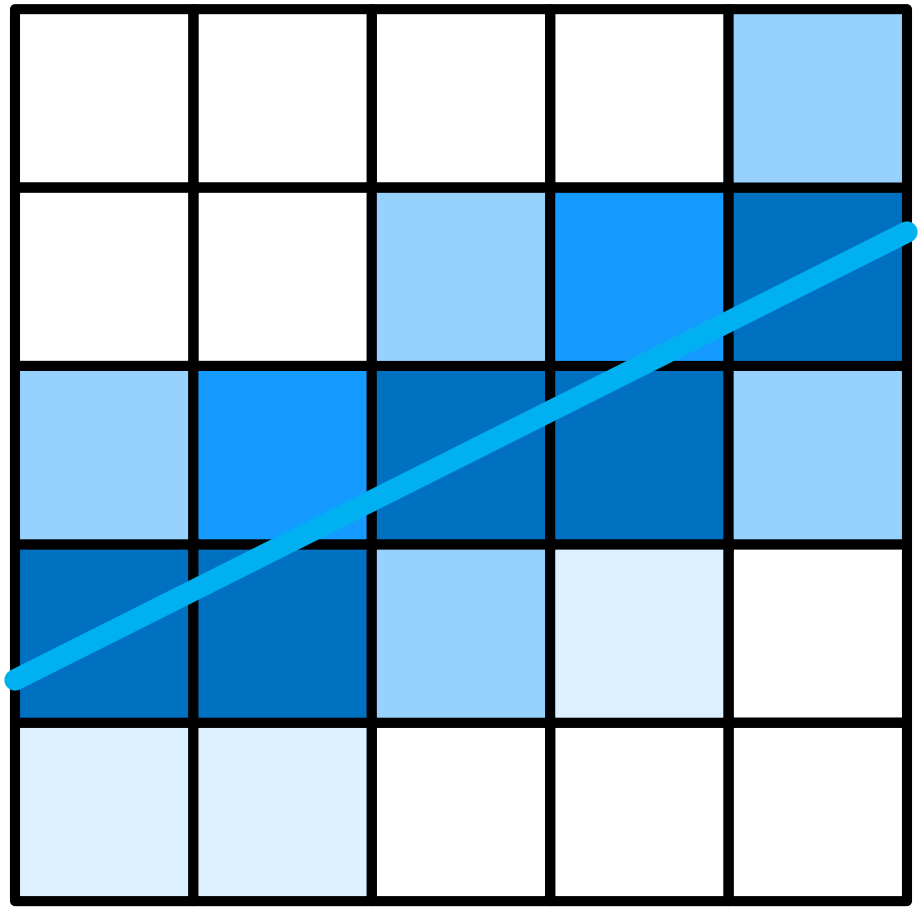
Antialiasing

Important Issues We've Ignored



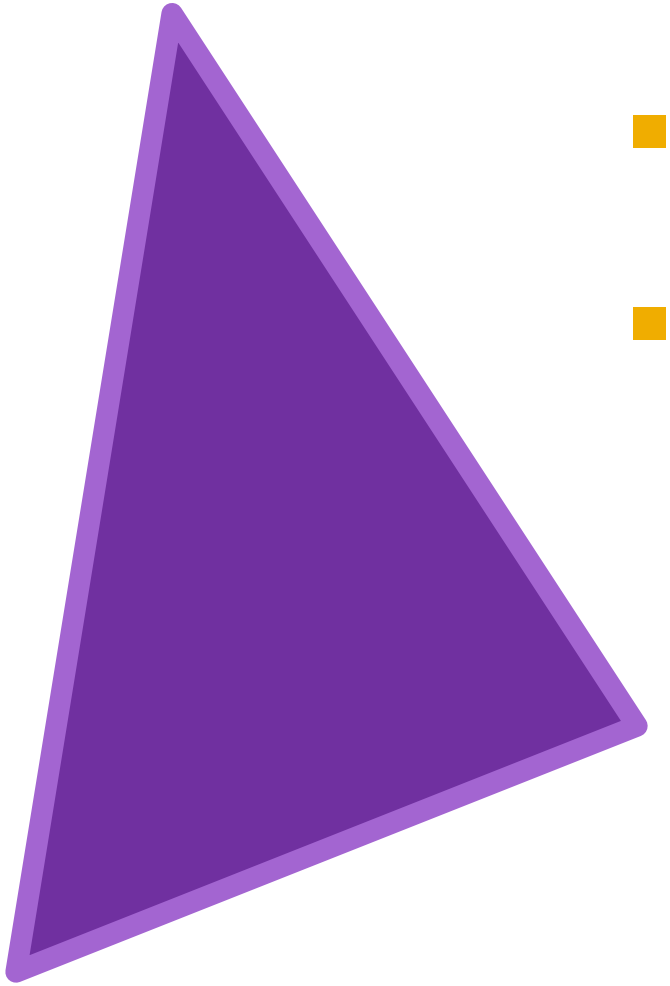
Antialiasing

Important Issues We've Ignored



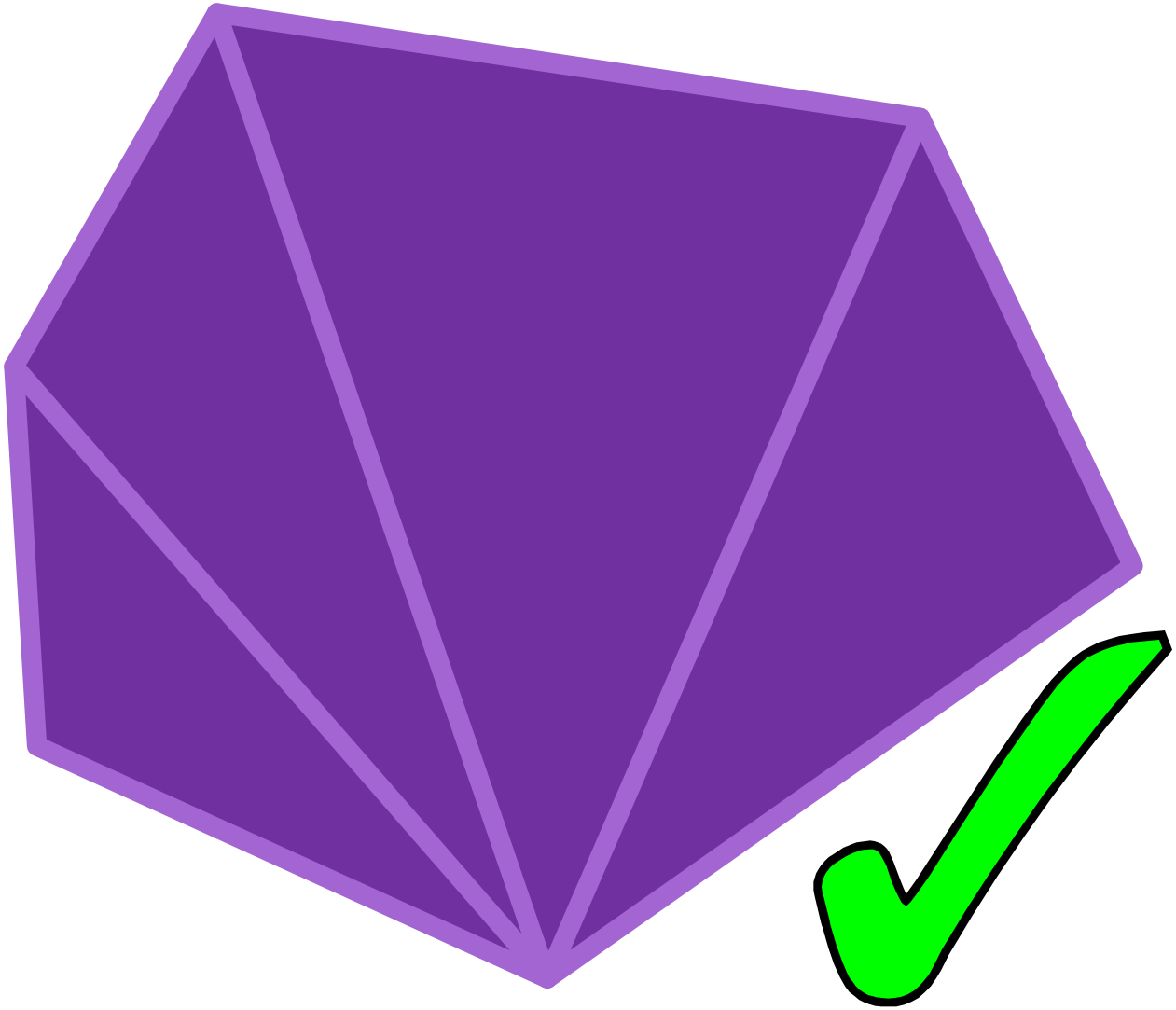
Antialiasing

Filling Triangles

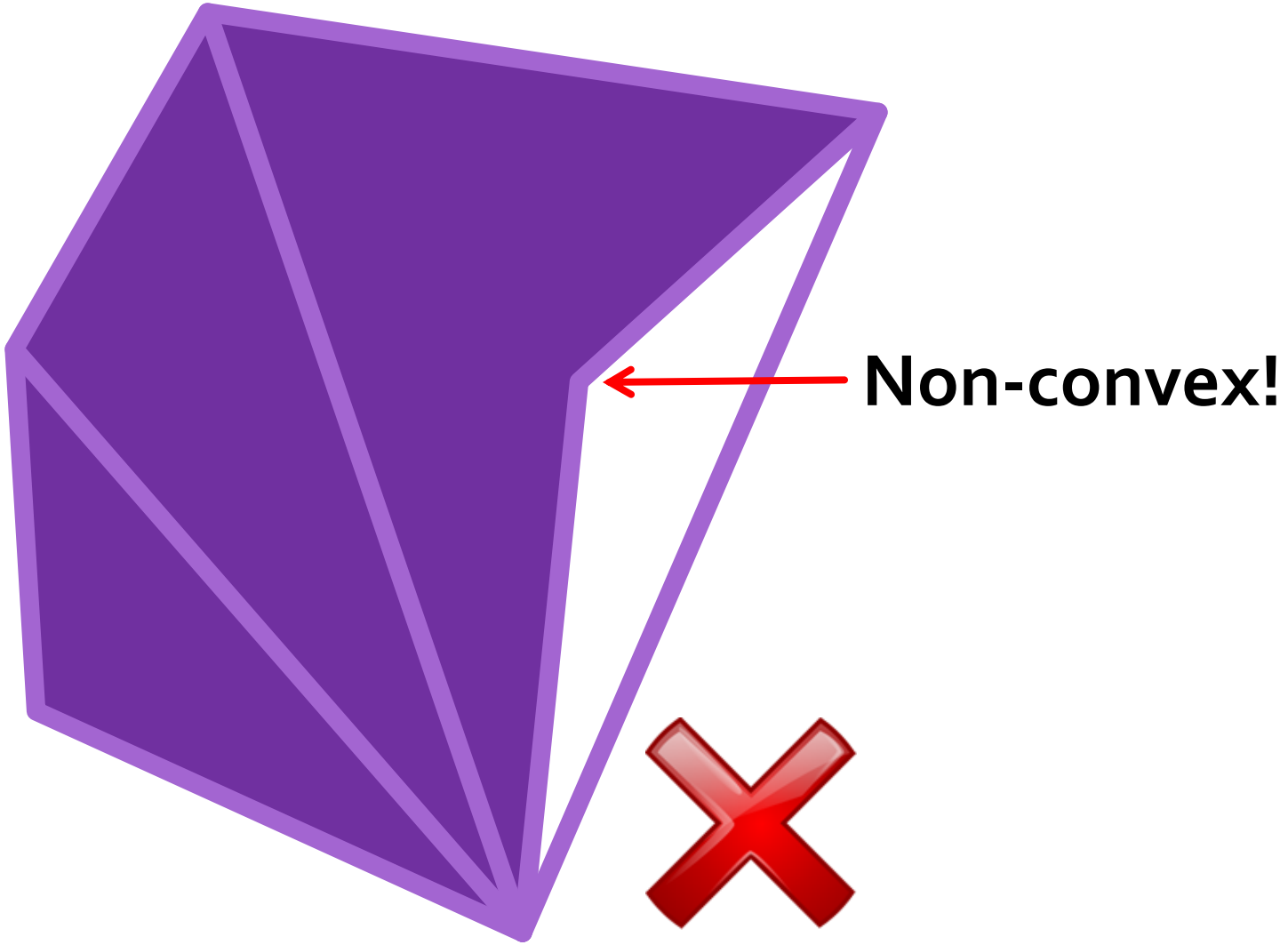


- Optimize a single primitive
- Nice properties:
  - Planar
  - “Inside” well-defined
  - Straightforward shading

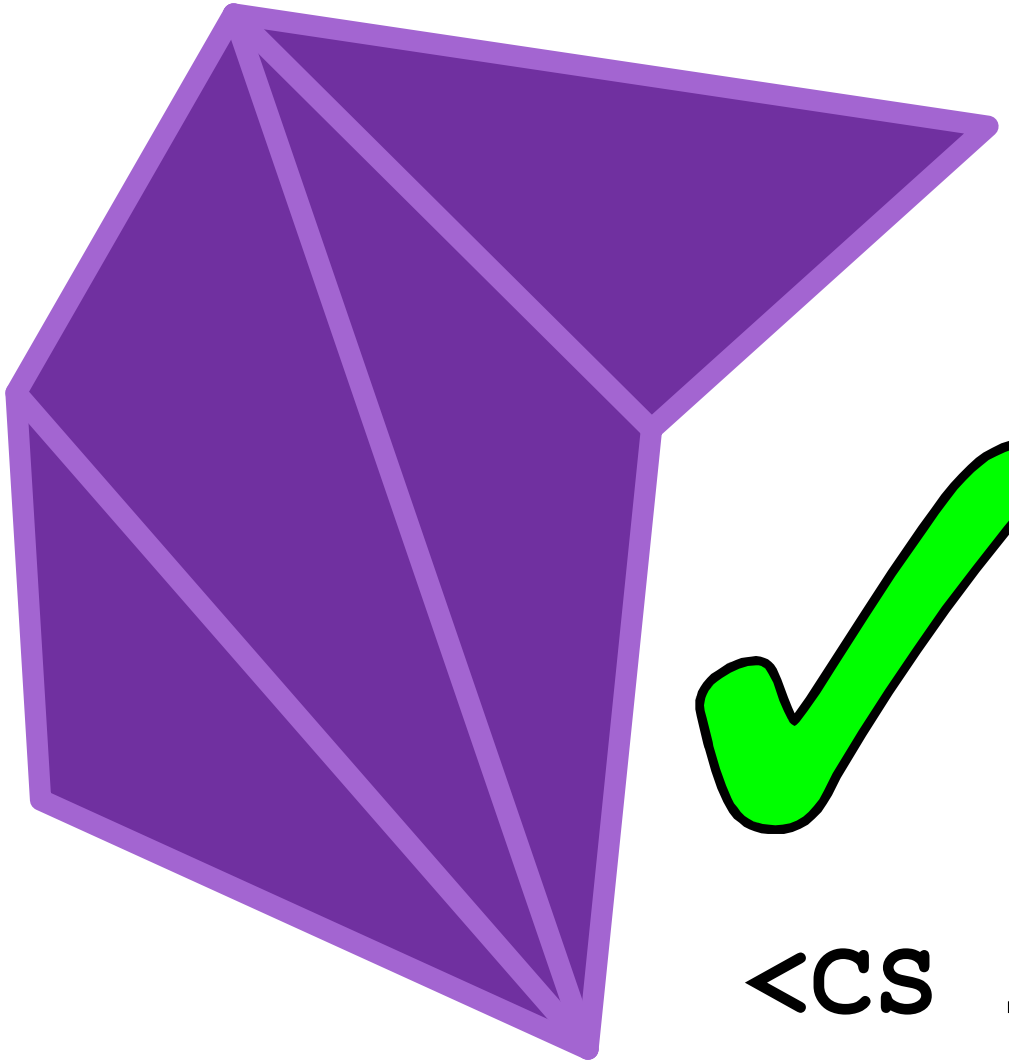
Have We Lost Anything?



Have We Lost Anything?

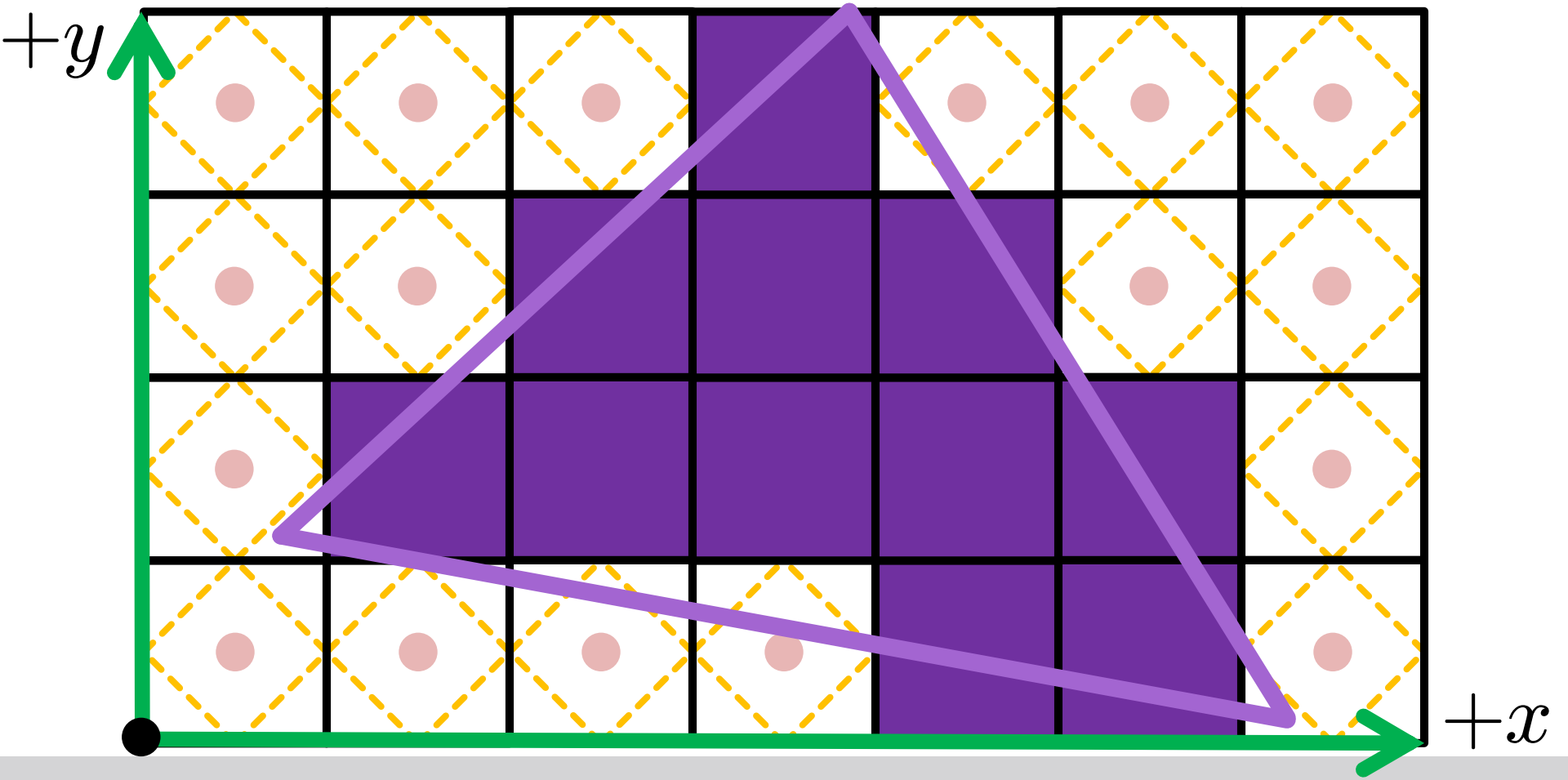


Have We Lost Anything?



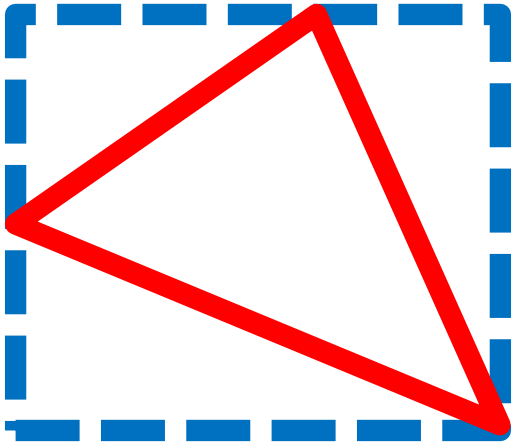
<CS 268 />

Convention

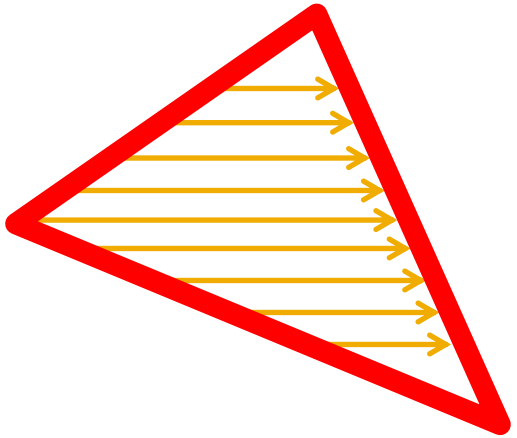


Must contain center

Two Methods for Filling

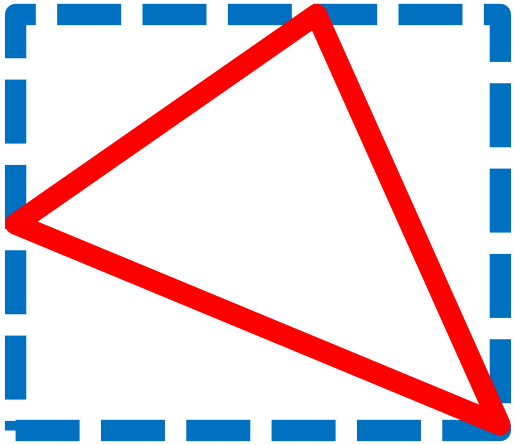


Check if each pixel in bounding box is inside the triangle.



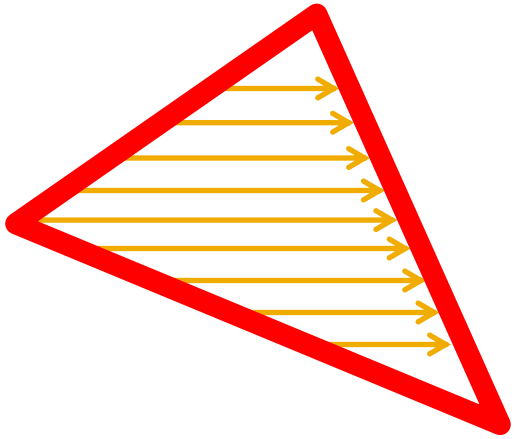
Rasterize border; sweep from left to right.

Two Methods for Filling



Check if each pixel in bounding box is inside the triangle.

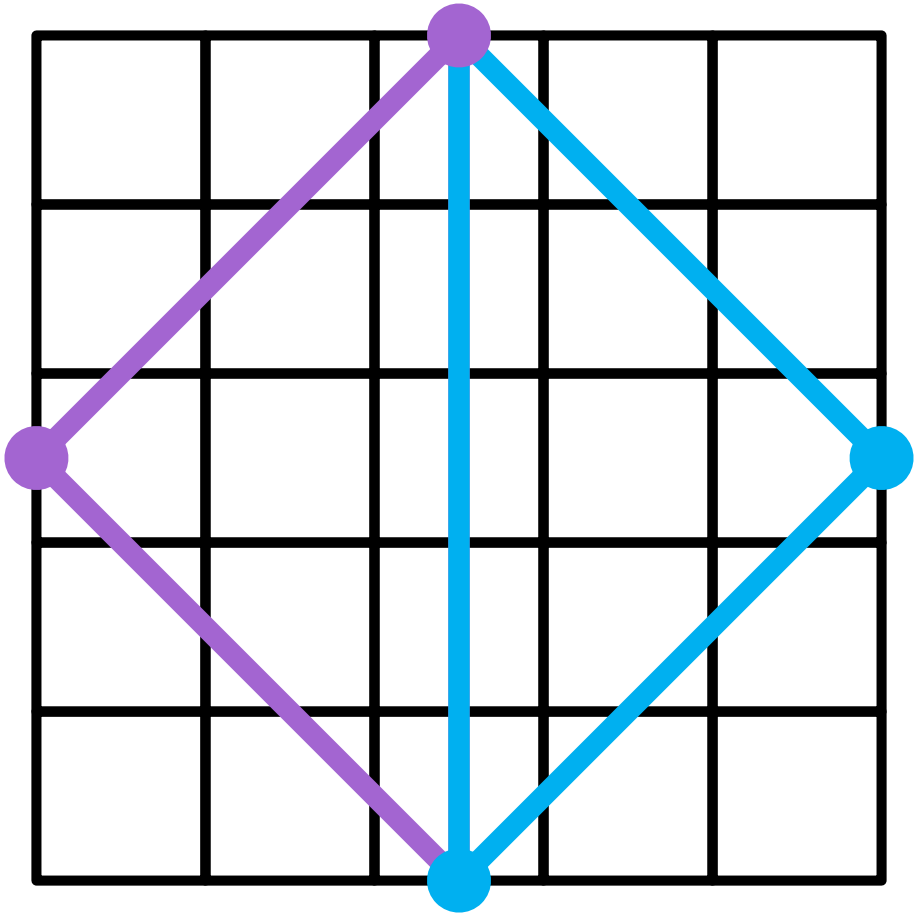
Parallelizable



Rasterize border; sweep from left to right.

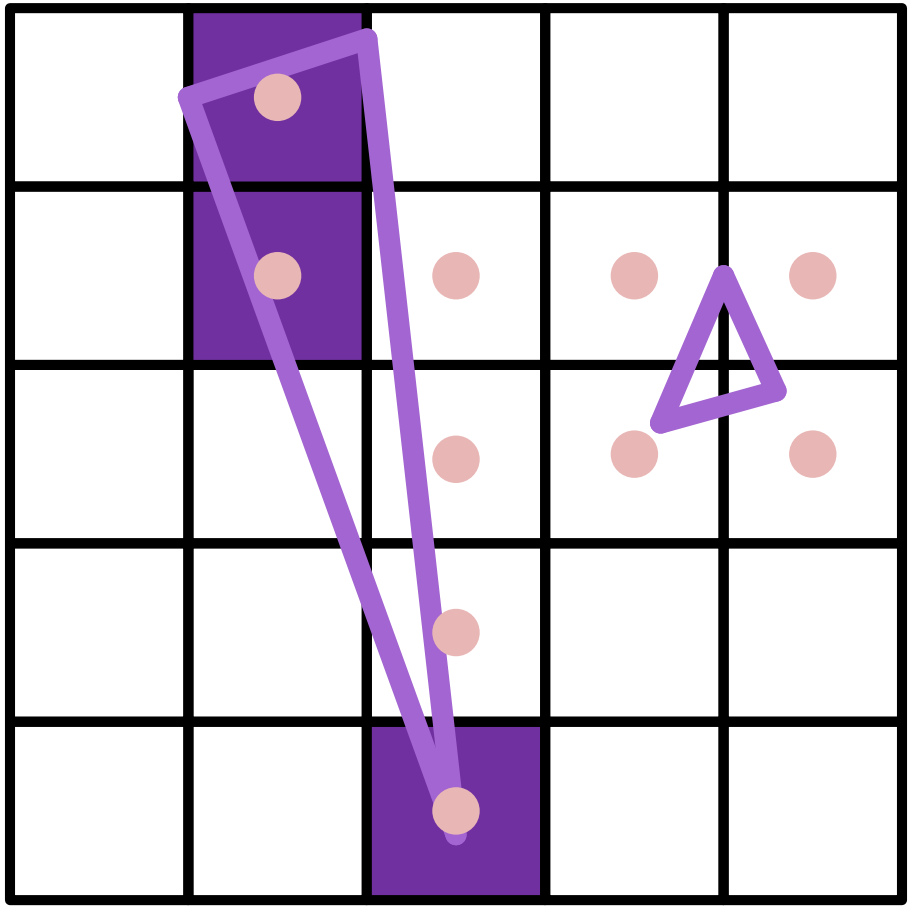
Less math

More Issues



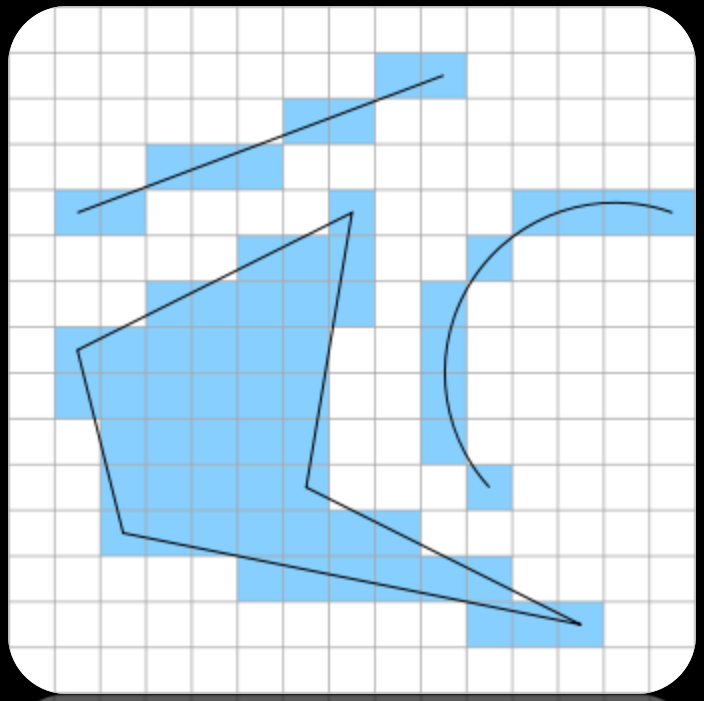
Adjacency & singularities

More Issues



Small triangles & slivers

http://iloveshaders.blogspot.com/2011/05/how-rasterization-process-works.html



Rasterization

CS 148, Summer 2012
Introduction to Computer Graphics and Imaging

Justin Solomon