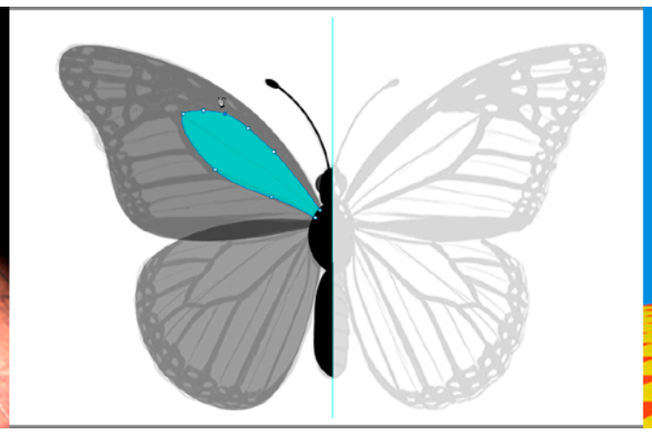
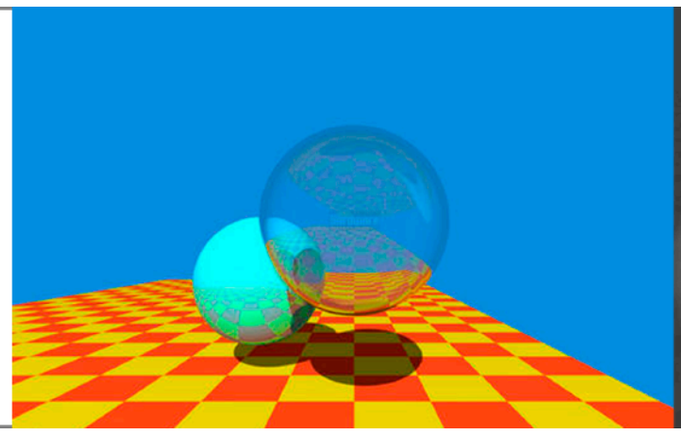
Introduction to Computer Graphics

GAMES101, Lingqi Yan, UC Santa Barbara

Lecture 5:
Rasterization 1 (Triangles)


http://www.cs.ucsb.edu/~lingqi/teaching/games101.html

Announcements

- Homework 0 — 188 submissions
  - No worries if you did not submit
- Homework 1 will be released today
  - Containing basic and advanced requirements (graded separately)
  - Pass or not pass depends on basic requirements only
- Asking on BBS
  - Please try to describe your question more clearly
- Today's lecture is pretty easy

Last Lecture

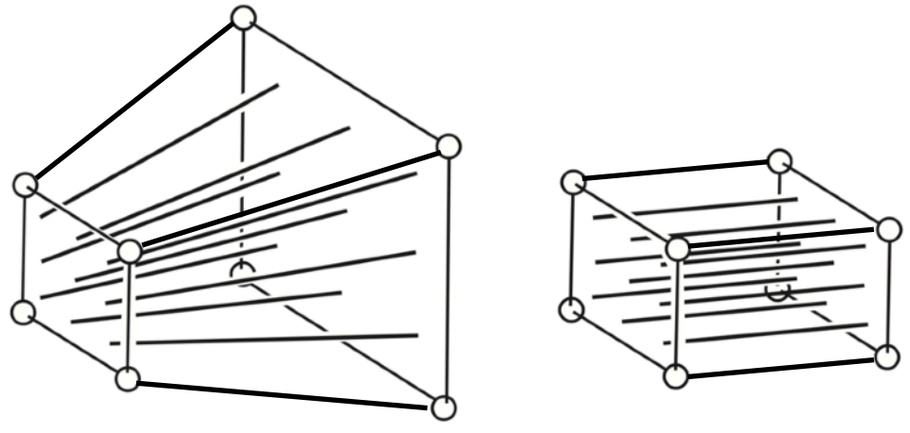
- Viewing (观测) transformation
  - View (视图) / Camera transformation
  - Projection (投影) transformation
    - Orthographic (正交) projection
    - Perspective (透视) projection

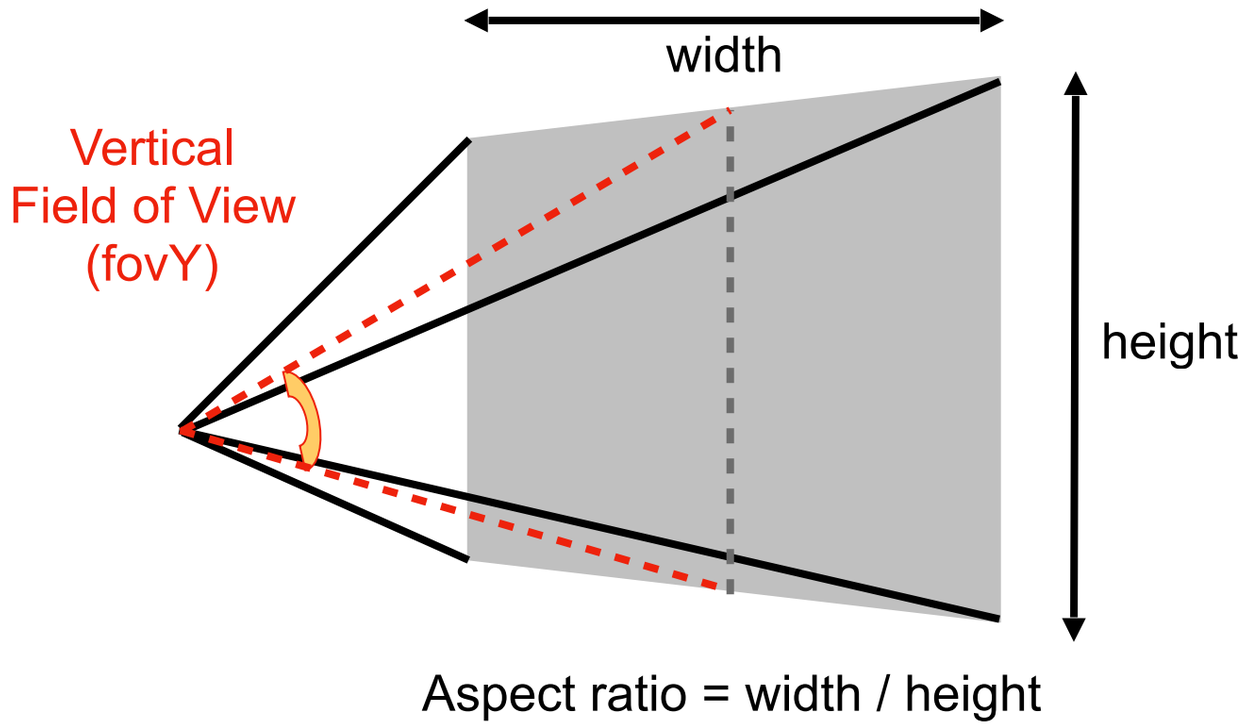
Today

- Finishing up Viewing
  - Viewport transformation
- Rasterization
  - Different raster displays
  - Rasterizing a triangle
- Occlusions and Visibility

Perspective Projection

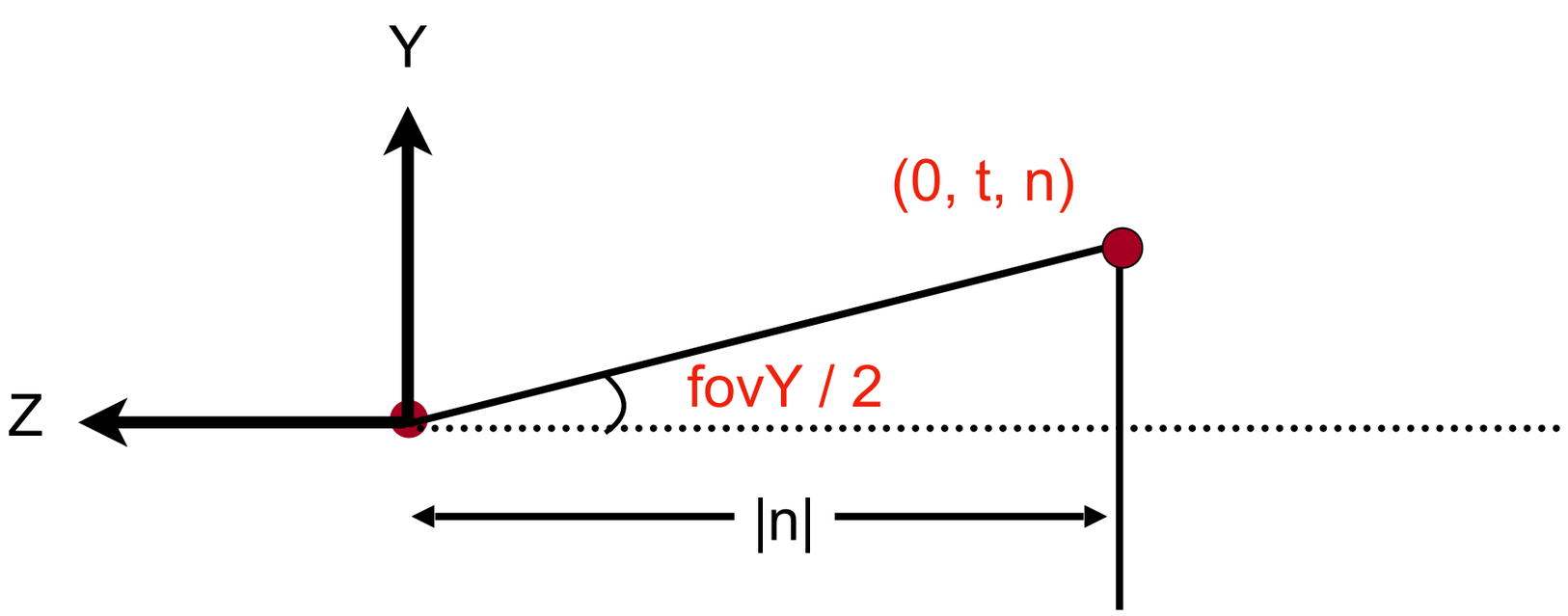
- What's near plane's l, r, b, t then?
  - If explicitly specified, good
  - Sometimes people prefer: vertical **field-of-view** (fovY) and **aspect ratio** (assume symmetry i.e. l = -r, b = -t)





Perspective Projection

- How to convert from fovY and aspect to l, r, b, t?
  - Trivial



$$\tan \frac{fovY}{2} = \frac{t}{|n|}$$

$$aspect = \frac{r}{t}$$

What’s after MVP?

- **M**odel transformation (placing objects)
- **V**iew transformation (placing camera)
- **P**rojection transformation
  - Orthographic projection (cuboid to “canonical” cube $[-1, 1]^3$)
  - Perspective projection (frustum to “canonical” cube)
- Canonical cube to ?

Canonical Cube to Screen

- What is a screen?
  - An array of pixels
  - Size of the array: resolution
  - A typical kind of raster display

光栅

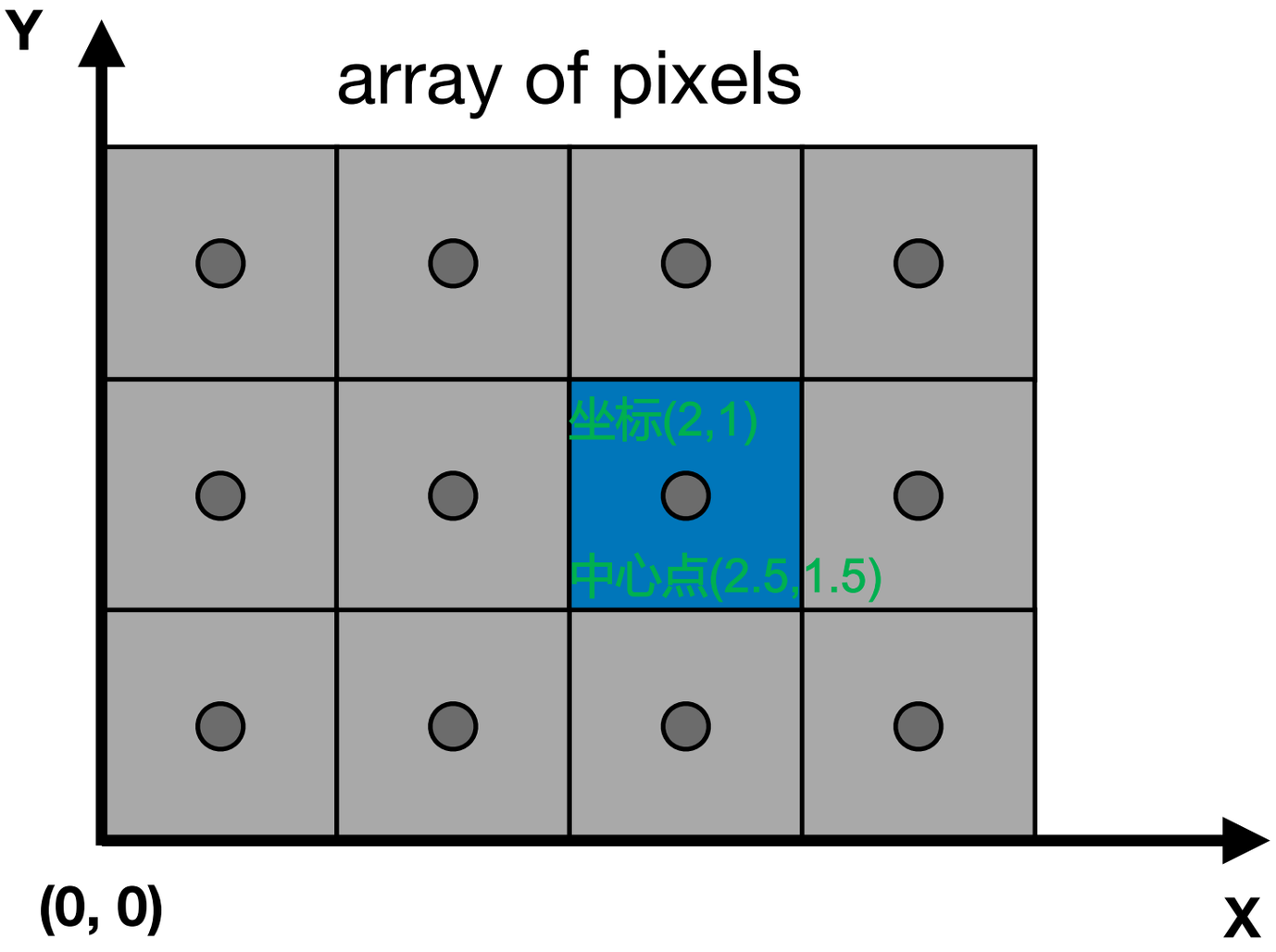
- Raster == screen in German

在德语中表示屏幕

  - Rasterize == drawing onto the screen
    把东西画在屏幕上...
- Pixel (FYI, short for “picture element”)
  - For now: A pixel is a little square with uniform color
  - Color is a mixture of (red, green, blue)

Canonical Cube to Screen

- Defining the screen space
  - Slightly different from the “tiger book”



Pixels’ indices are
in the form of (x, y), where
both x and y are integers

Pixels’ indices are from
(0, 0) to (width - 1, height - 1)

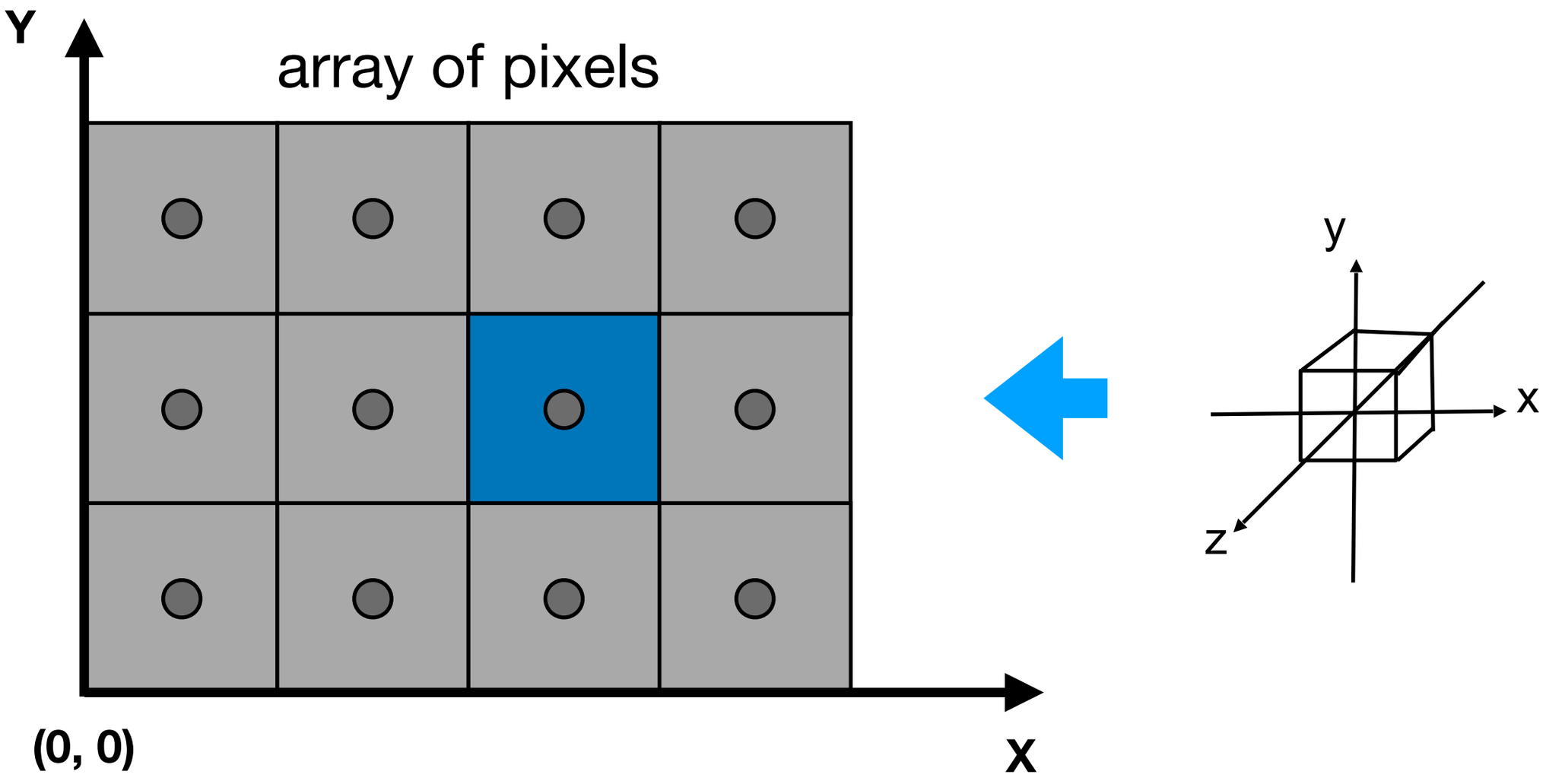
Pixel (x, y) is centered at
(x + 0.5, y + 0.5)

The screen covers range
(0, 0) to (width, height)

Canonical Cube to Screen

无关紧要

- Irrelevant to z
- Transform in xy plane: $[-1, 1]^2$ to [0, width] x [0, height]



Canonical Cube to Screen

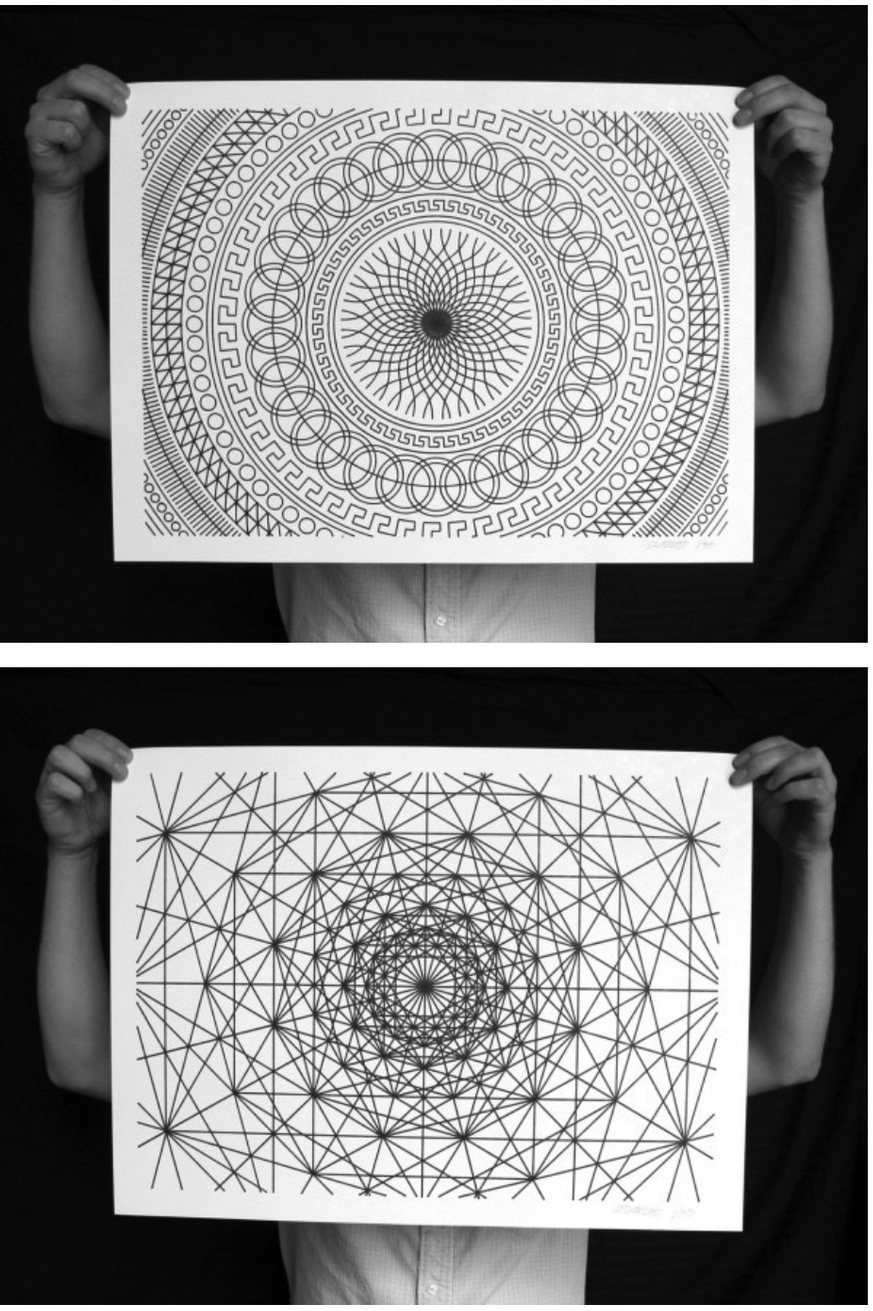
- Irrelevant to z
- Transform in xy plane: $[-1, 1]^2$ to [0, width] x [0, height]
- Viewport transform matrix:

$$M_{viewport} = \begin{pmatrix} \frac{width}{2} & 0 & 0 & \frac{width}{2} \\ 0 & \frac{height}{2} & 0 & \frac{height}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Next: Rasterizing Triangles into Pixels

Drawing Machines

CNC Sharpie Drawing Machine



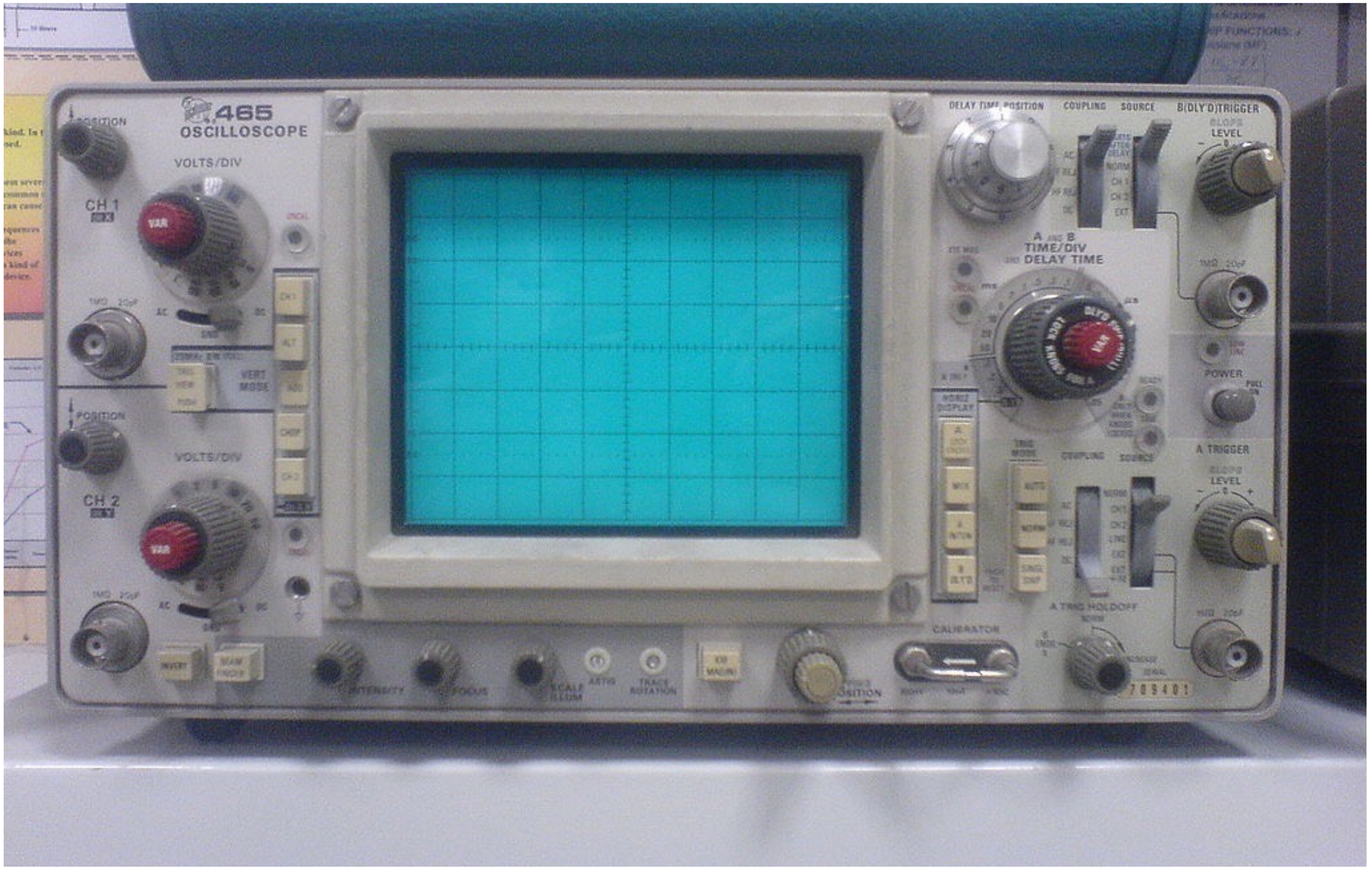
Aaron Panone with Matt W. Moore

http://44rn.com/projects/numerically-controlled-poster-series-with-matt-w-moore/

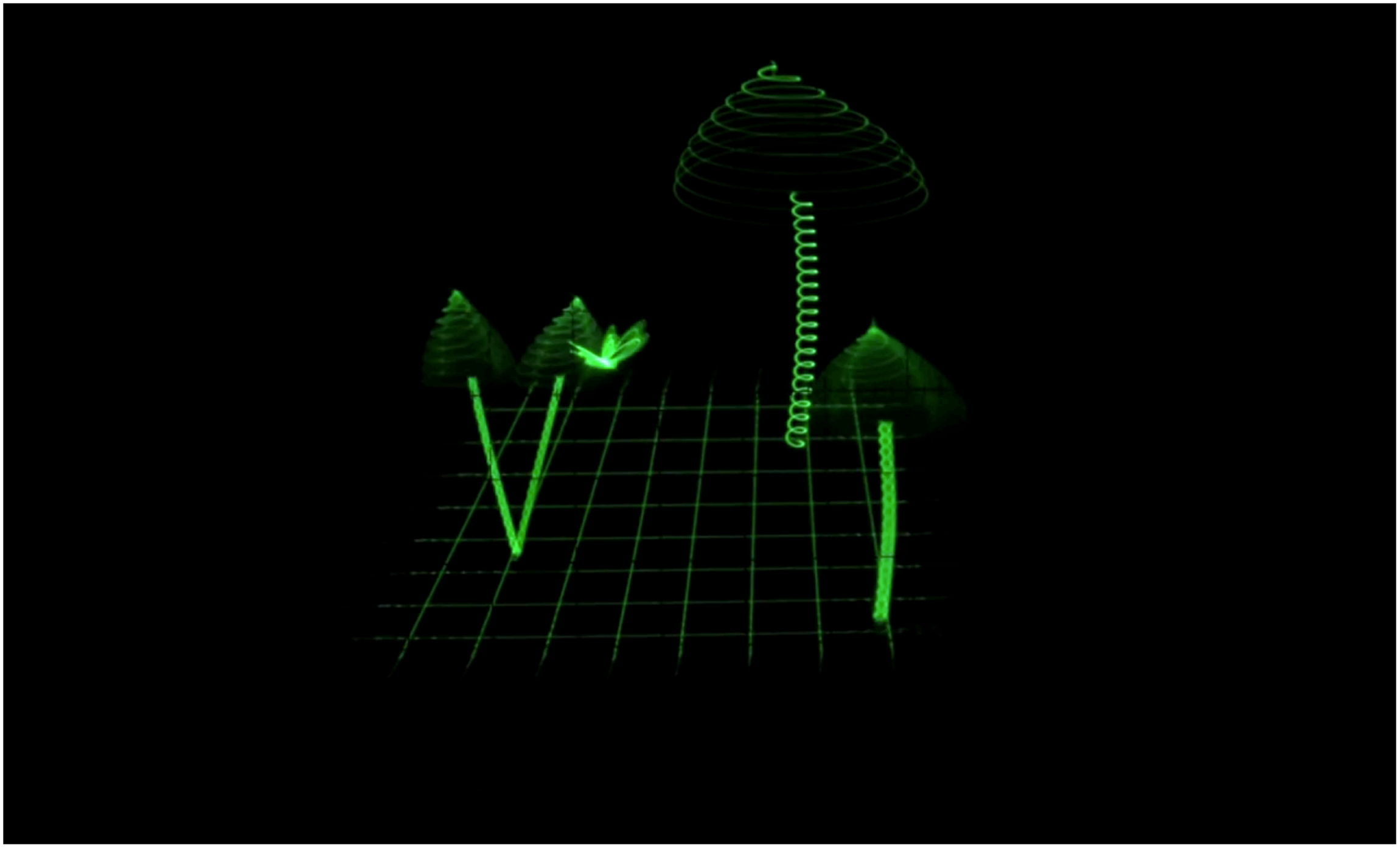
Laser Cutters

Different Raster Displays

Oscilloscope



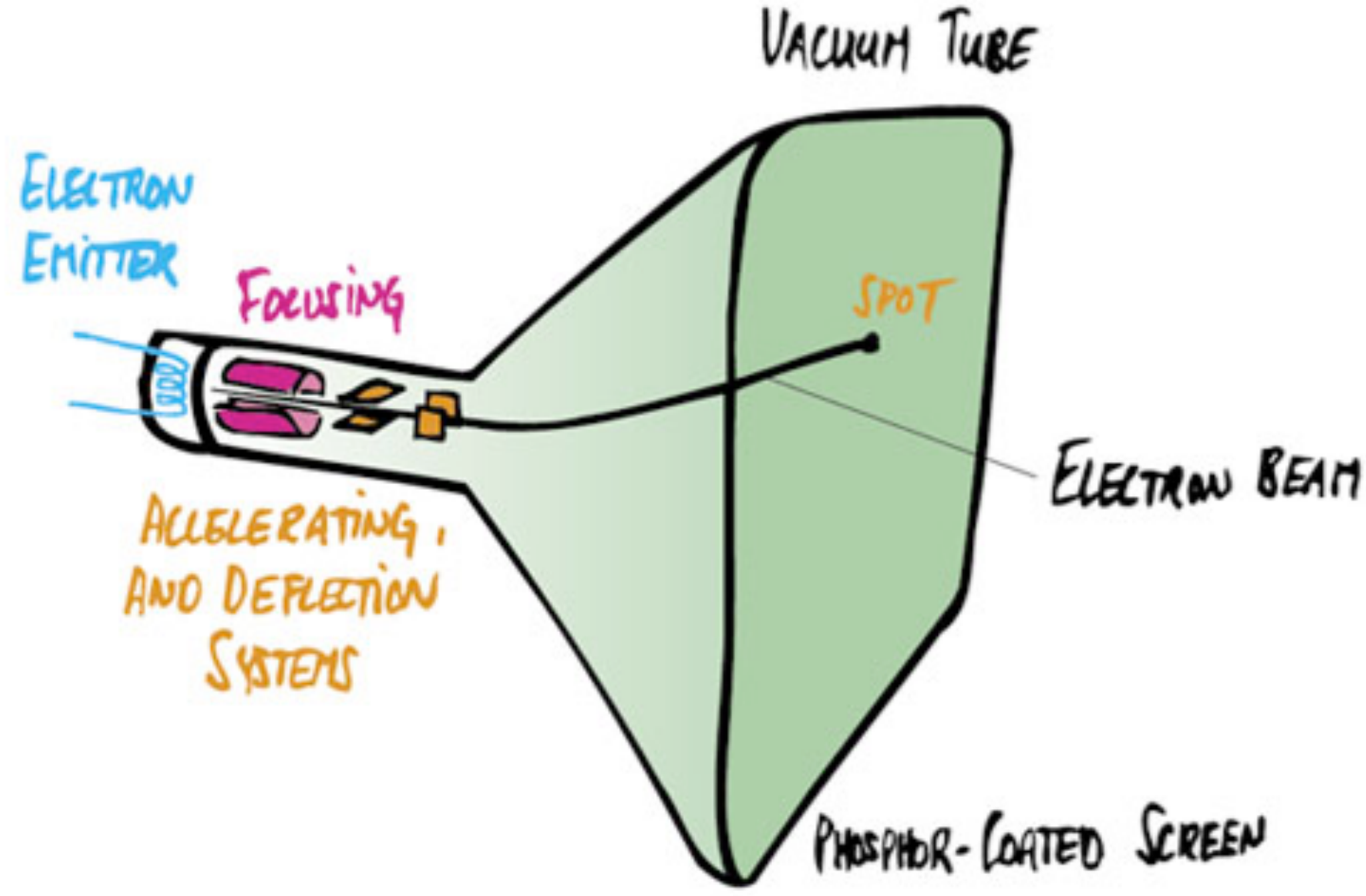
Oscilloscope Art



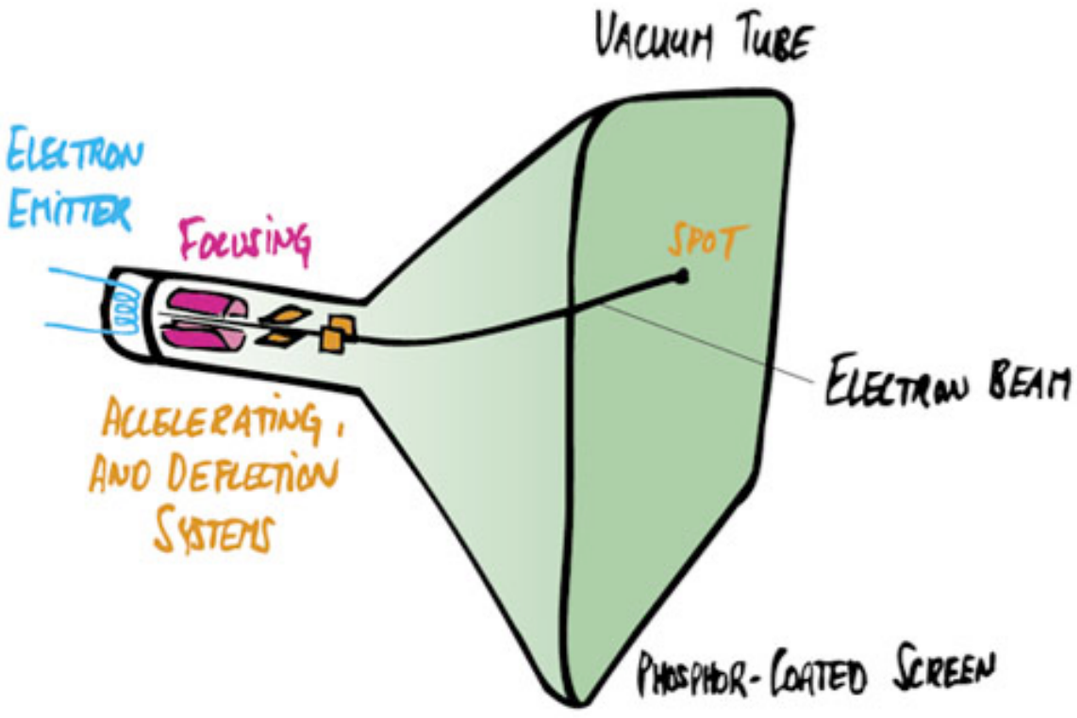
Jerobeam Fenderson

https://www.youtube.com/watch?v=rtR63-ecUNo

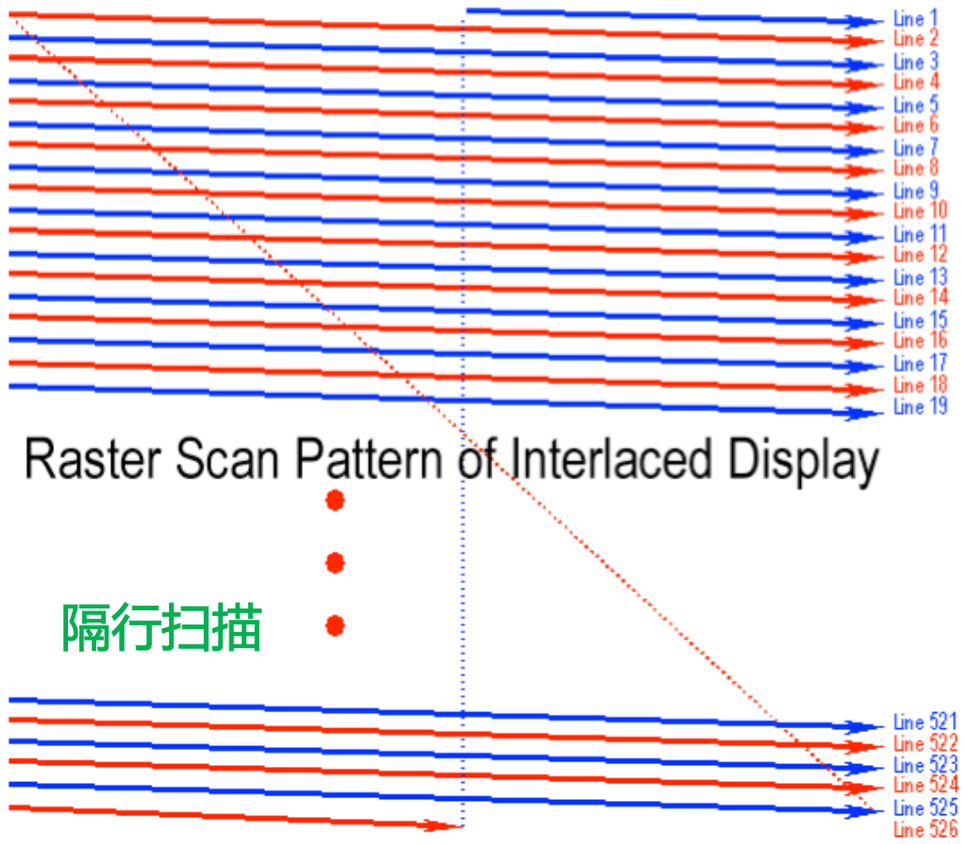
Cathode Ray Tube



Television - Raster Display CRT



Cathode Ray Tube



Raster Scan
(modulate intensity)

Frame Buffer: Memory for a Raster Display

DAC =
Digital to Analog Convertors

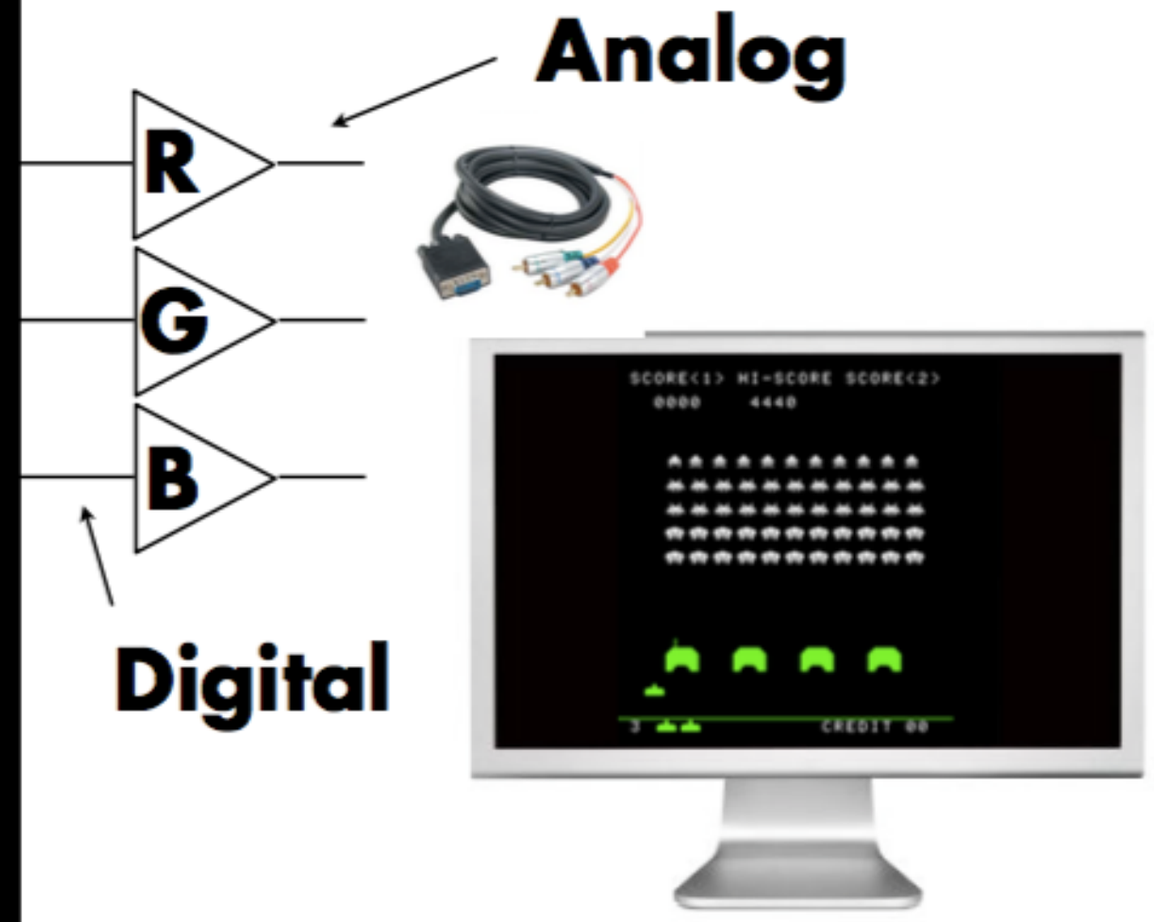


Image = 2D array of colors

Flat Panel Displays

Low-Res LCD Display

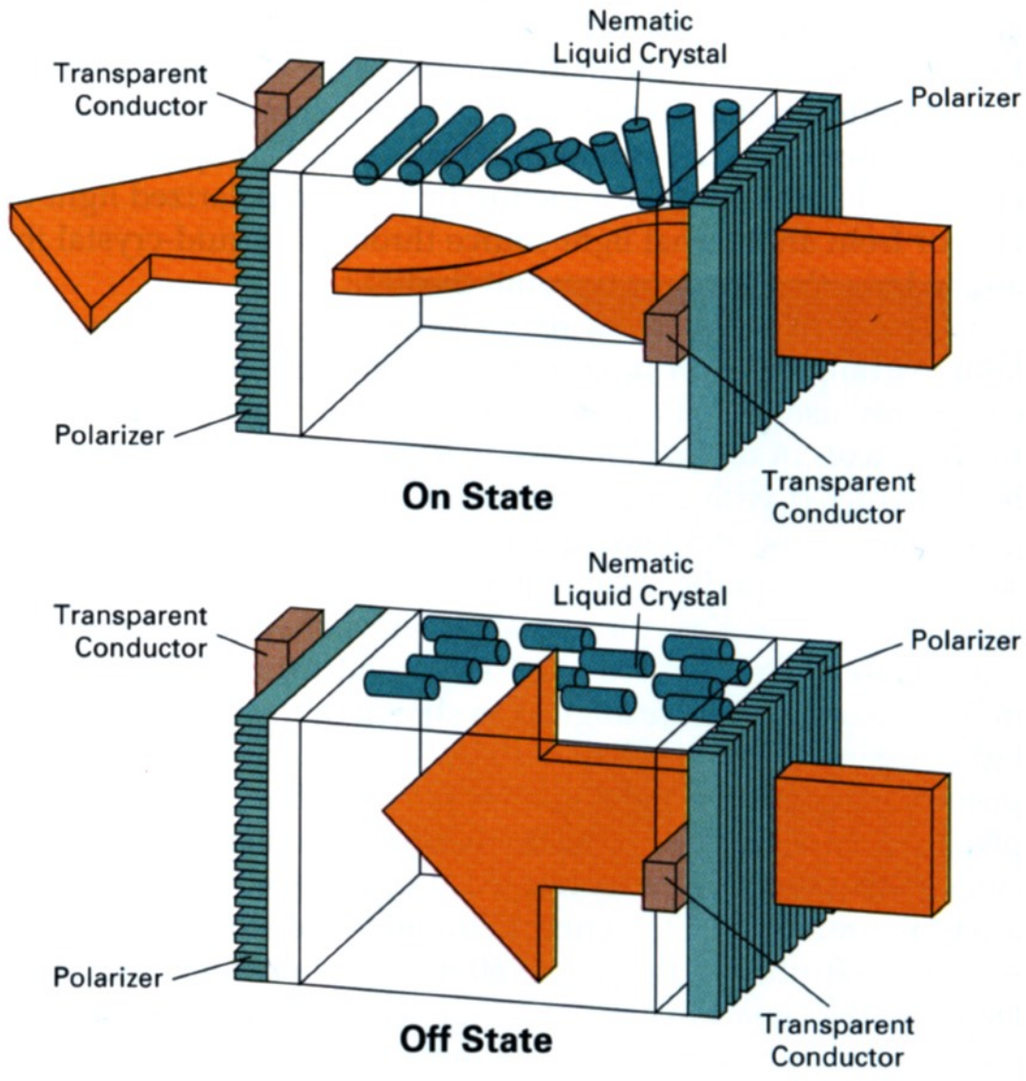
Color LCD, OLED, …

LCD (Liquid Crystal Display) Pixel

Principle: block or transmit light by twisting polarization

Illumination from backlight (e.g. fluorescent or LED)

Intermediate intensity levels by partial twist

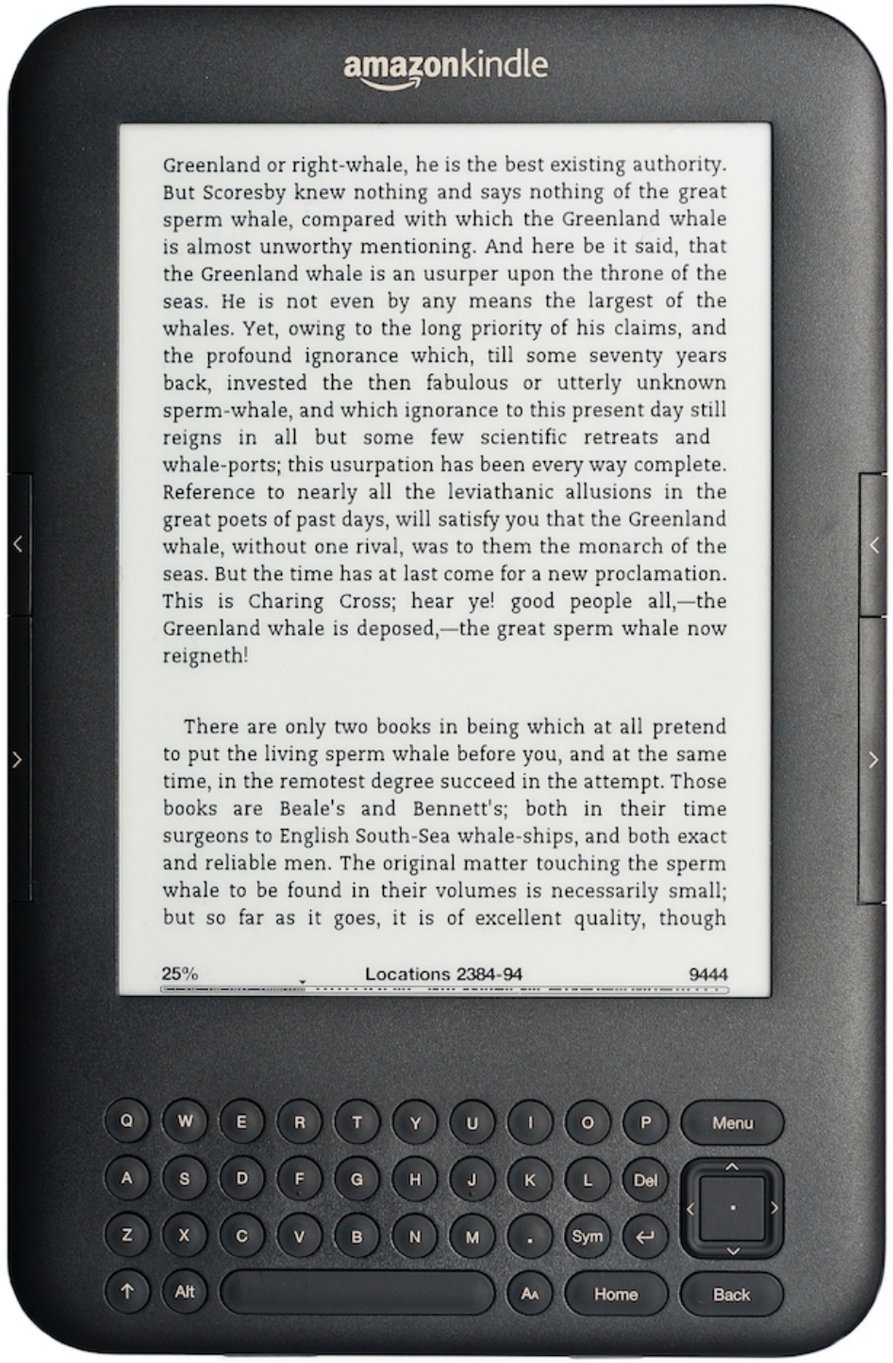


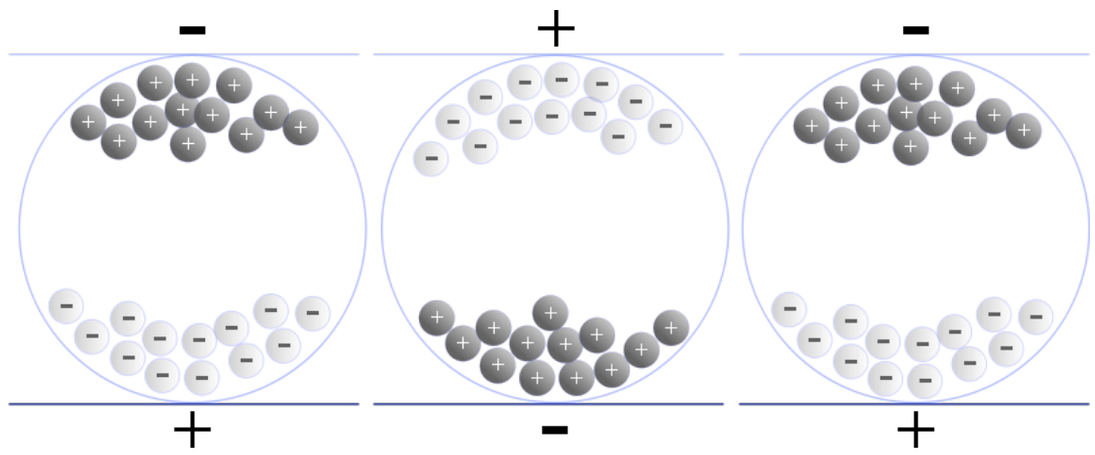
[H&B fig. 2-16]

LED Array Display

Light emitting diode array

Electrophoretic (Electronic Ink) Display





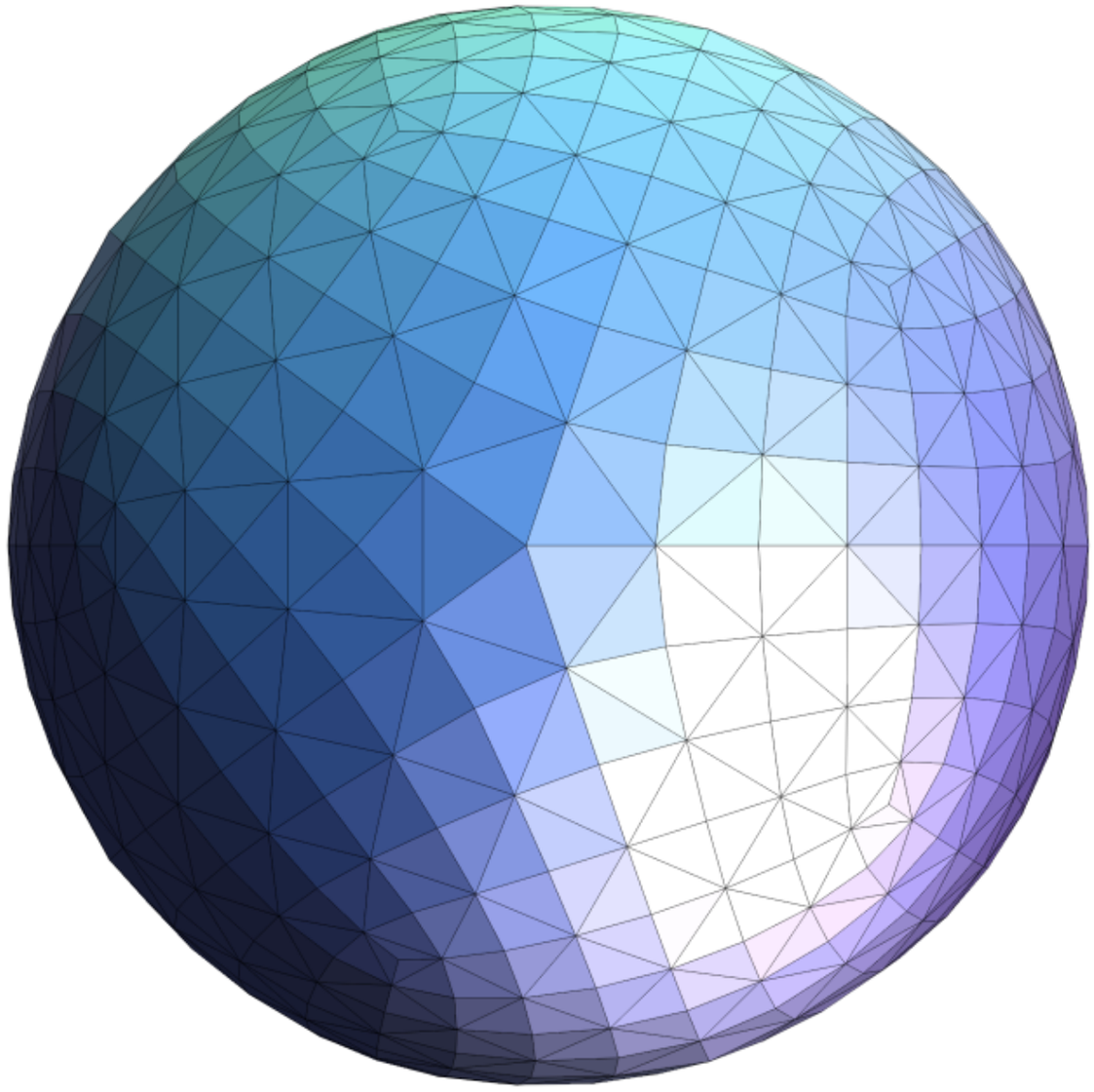
[Wikimedia Commons —Senarclens]

Rasterization:

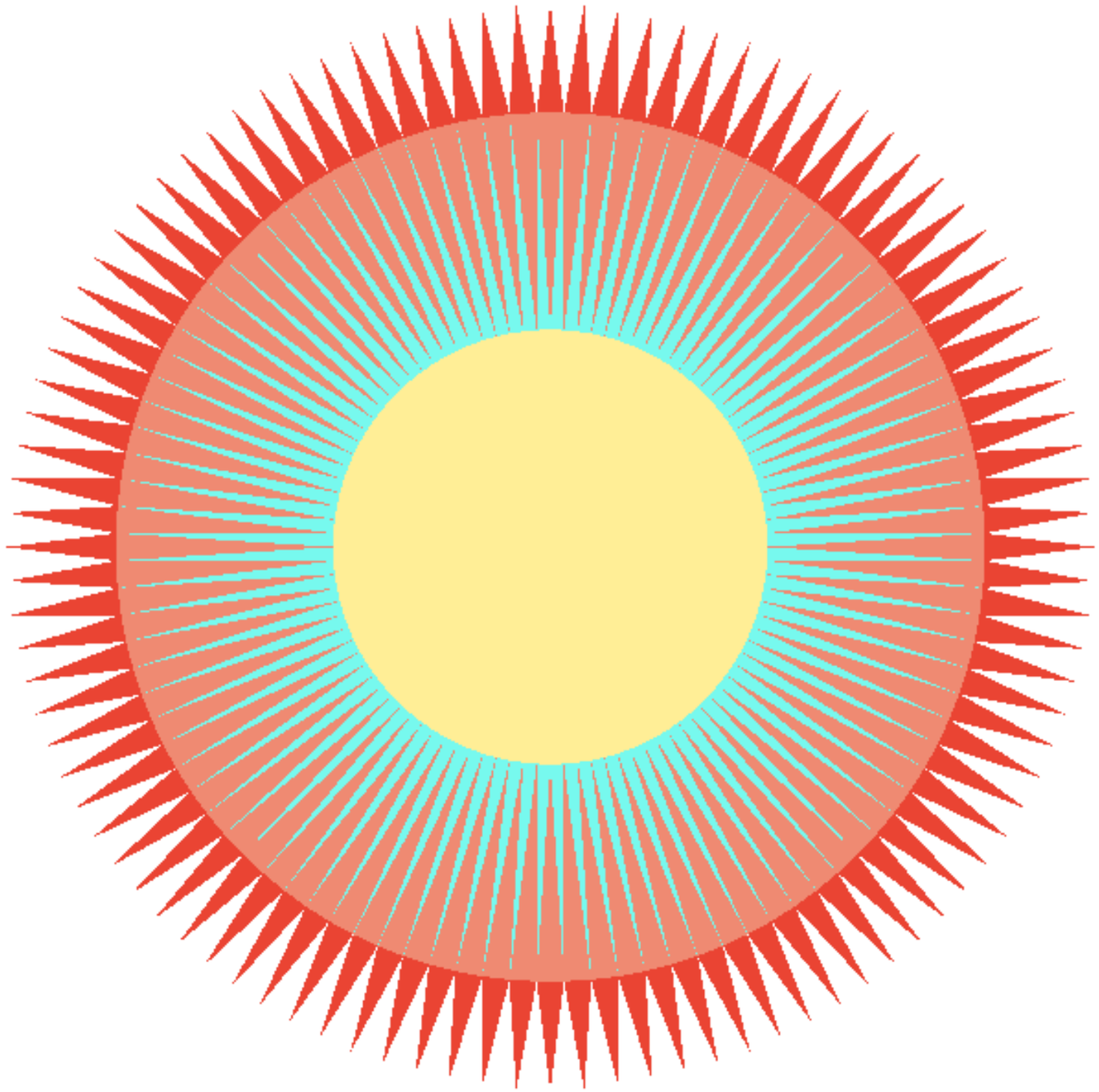
Drawing to Raster Displays

Polygon Meshes

Triangle Meshes



Triangle Meshes



Triangles - Fundamental Shape Primitives

Why triangles?

- Most basic polygon
  - Break up other polygons

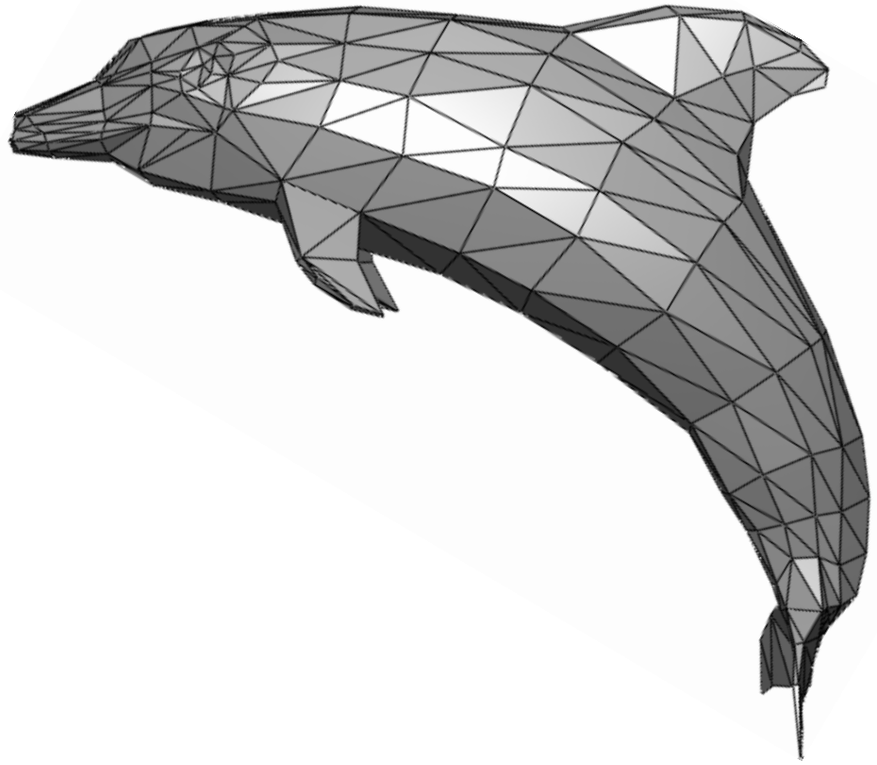
    分解其他图形

- Unique properties
  - Guaranteed to be planar
  - Well-defined interior
  - Well-defined method for interpolating values at vertices over triangle (barycentric interpolation)

    插值

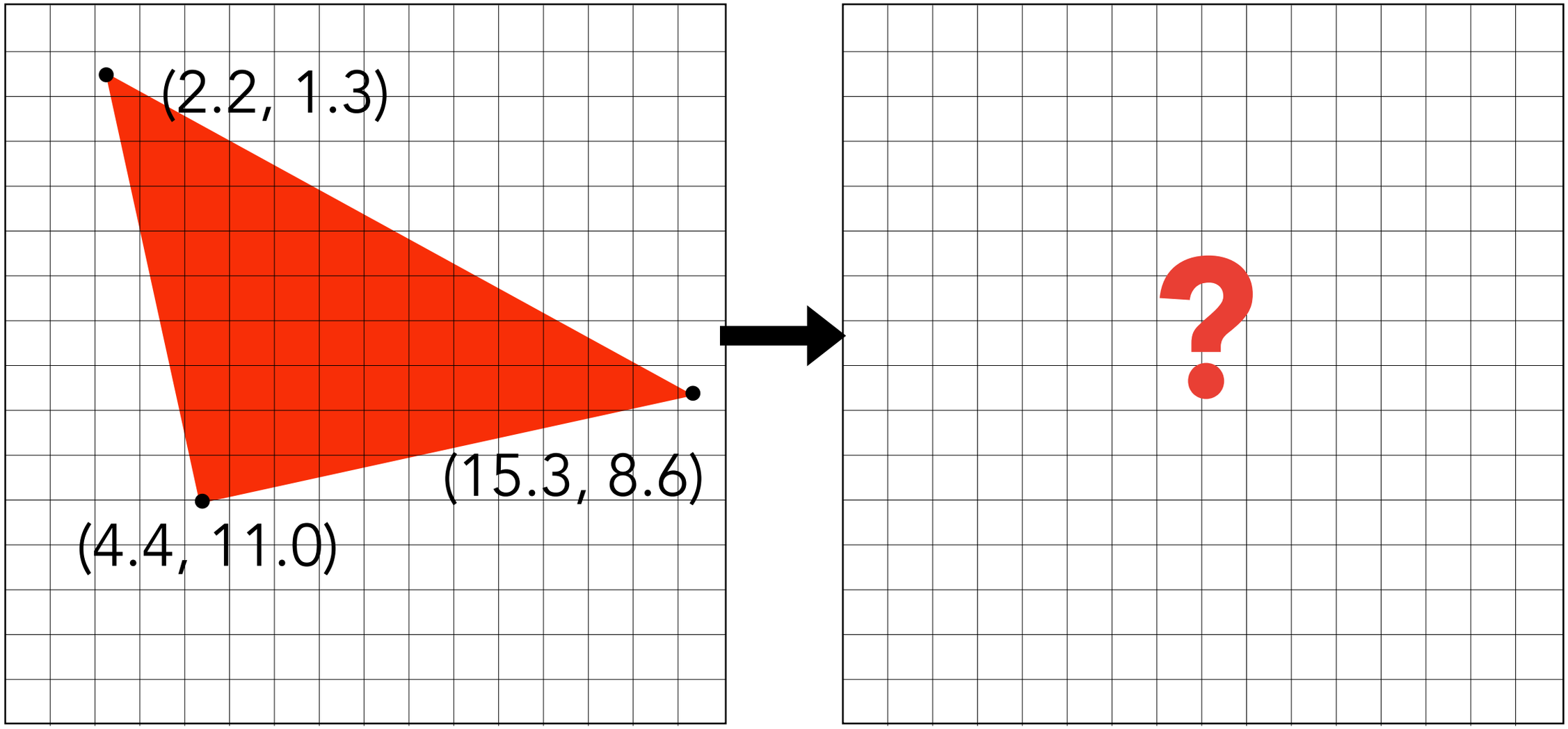
    质心的

    通过三角形内任意点与三个顶点的关系得到一个逐渐的变化



What Pixel Values Approximate a Triangle?

判断像素中心点与三角形的位置关系　像素如何显示颜色



Input: position of triangle vertices projected on screen

Output: set of pixel values approximating triangle

A Simple Approach: Sampling

Sampling a Function

给一个连续函数，在不同的地方问值是多少

采样：把一个函数离散化的过程，例如f(x)=sin(x),在x=1,2,3,...,100...的位置值是多少

Evaluating a function at a point is sampling.

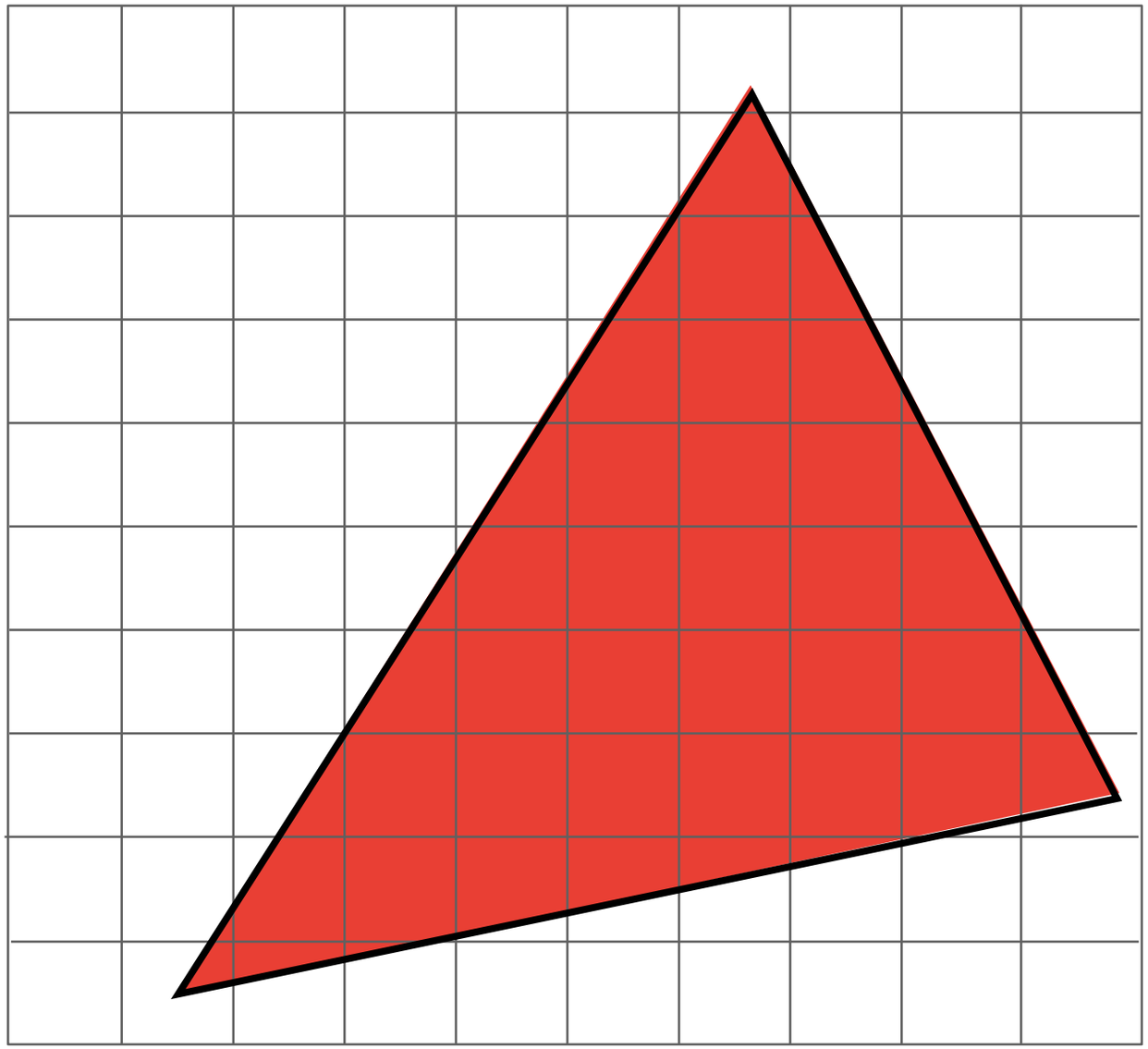
We can discretize a function by sampling.

```
for (int x = 0; x < xmax; ++x)
    output[x] = f(x);
```

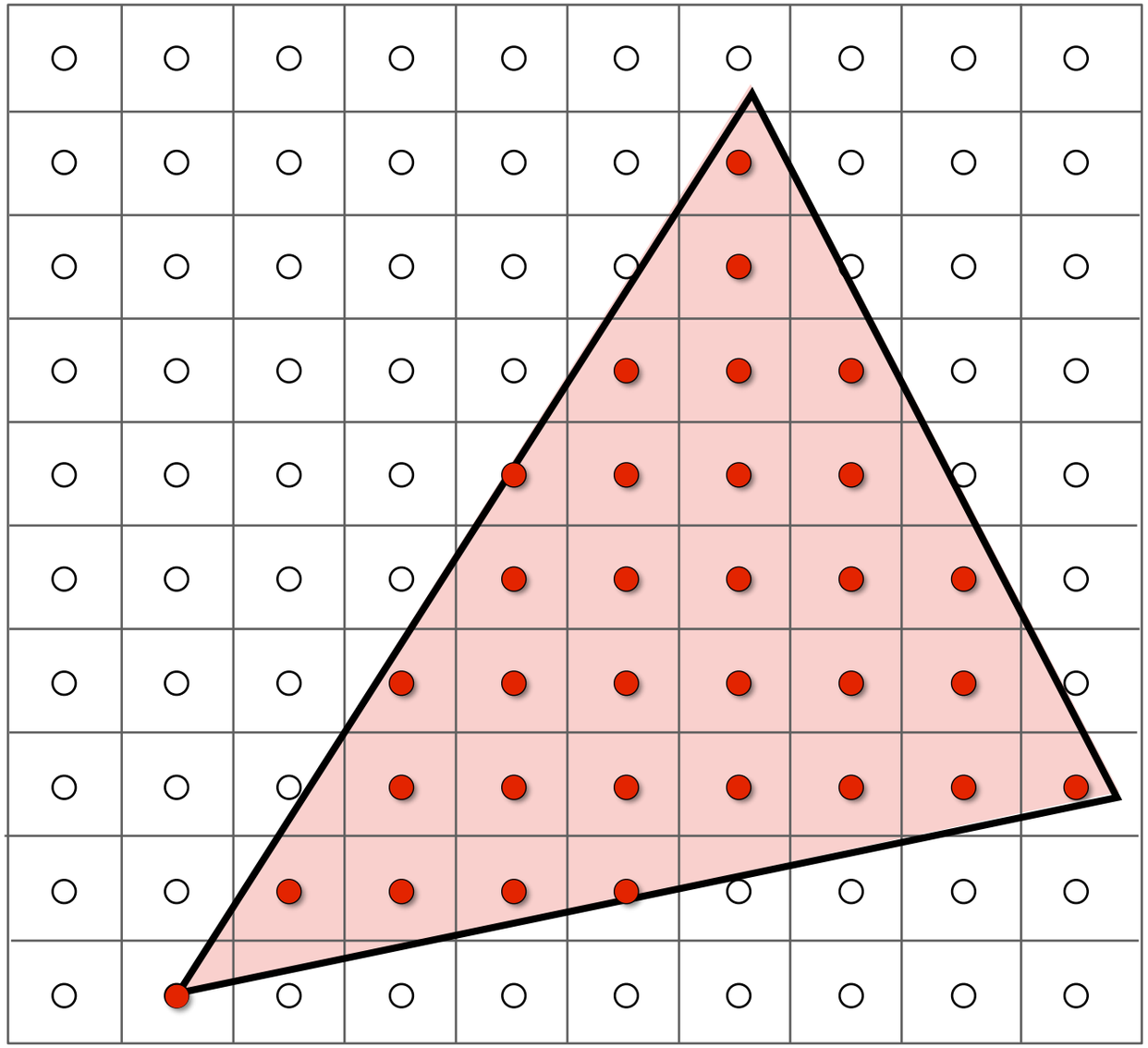
Sampling is a core idea in graphics.
We sample time (1D), area (2D), direction (2D), volume (3D) …

用像素中心对屏幕空间进行采样

Rasterization As 2D Sampling

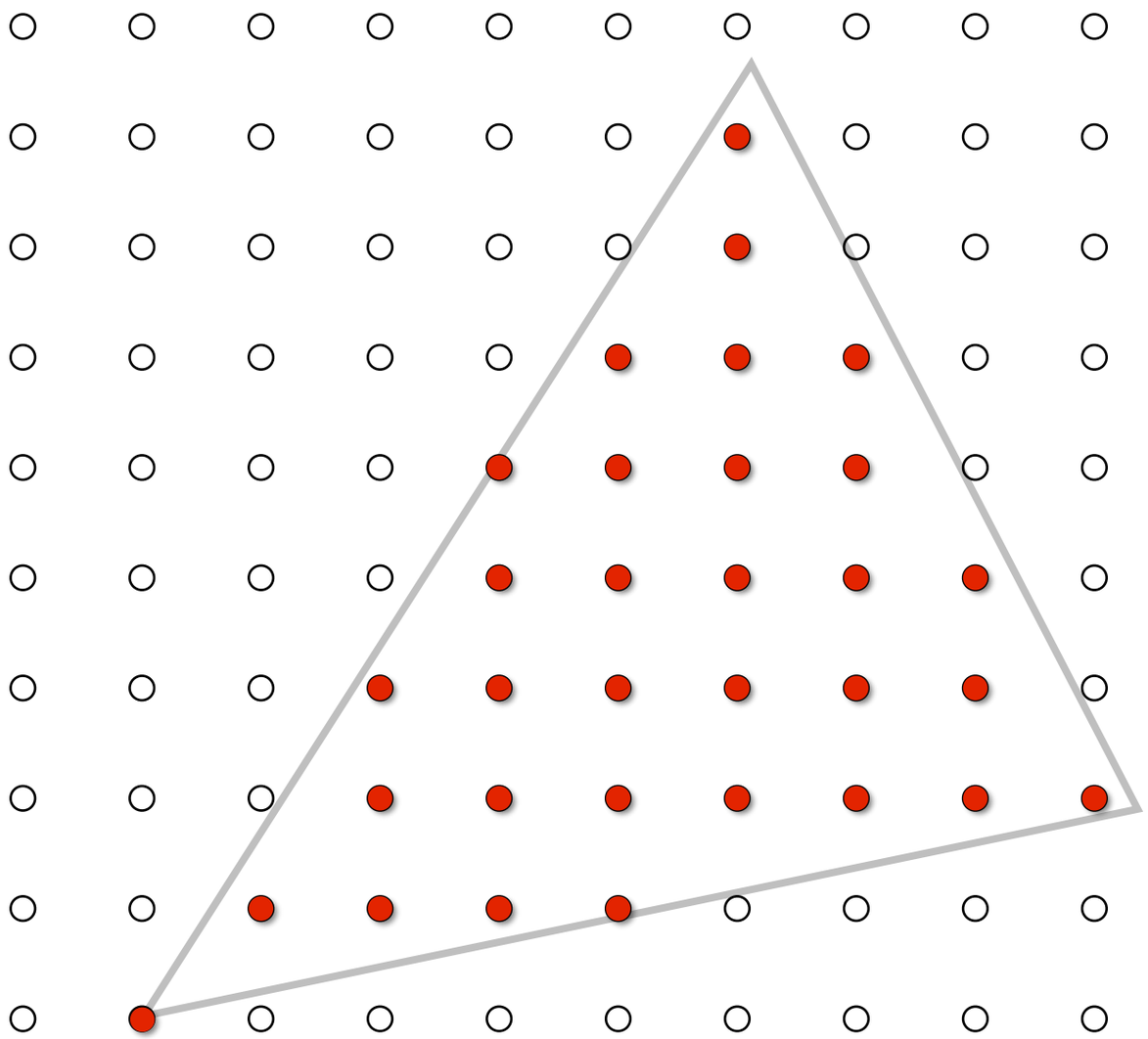


Sample If Each Pixel Center Is Inside Triangle



判断像素中心是否在三角形内

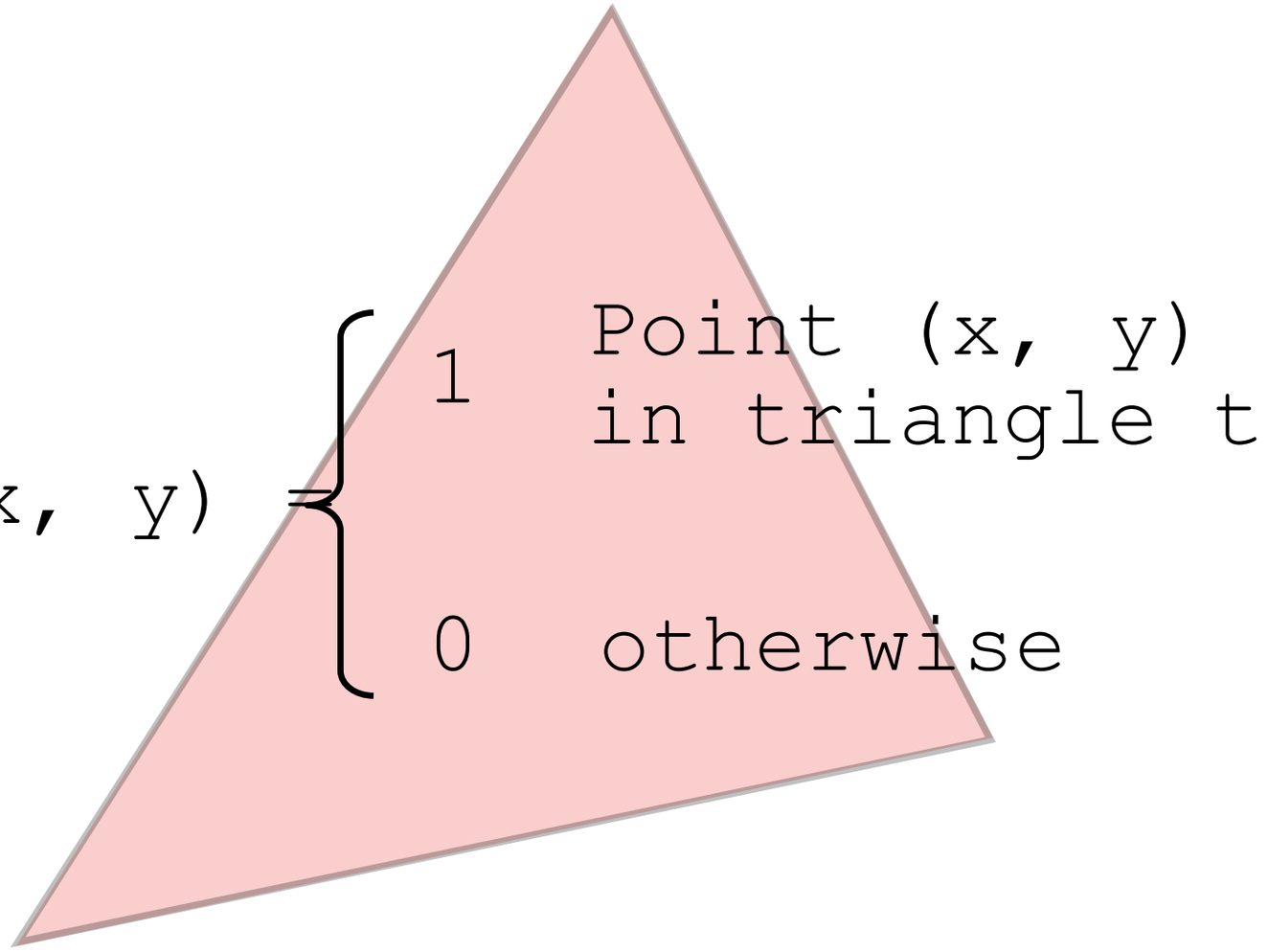
Sample If Each Pixel Center Is Inside Triangle



Define Binary Function: `inside(tri, x, y)`

x, y: not necessarily integers

$$\text{inside}(t, x, y) = \begin{cases} 1 & \text{Point } (x, y) \text{ in triangle } t \\ 0 & \text{otherwise} \end{cases}$$

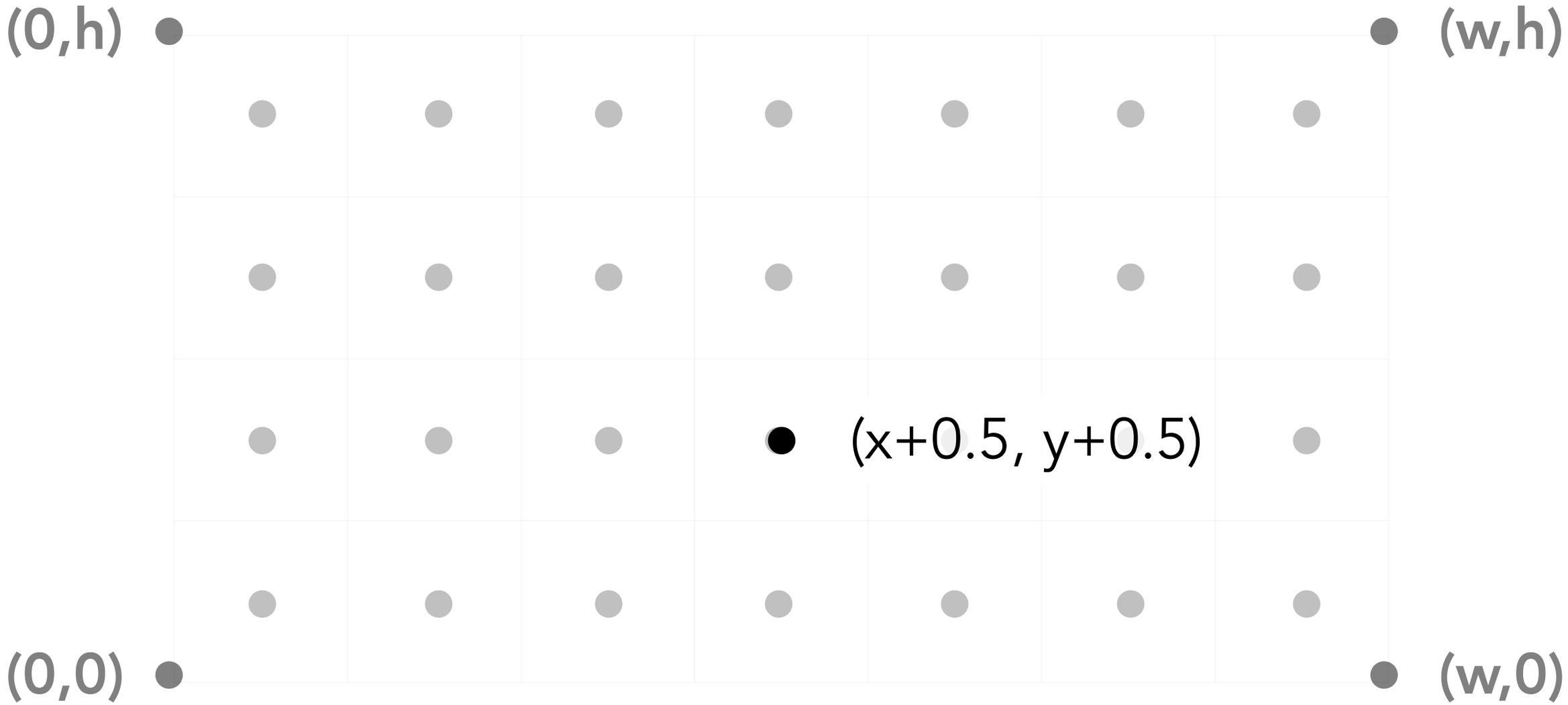


给一个三角形，和屏幕中的任意一点，判断是否在三角形内

Rasterization = Sampling A 2D Indicator Function

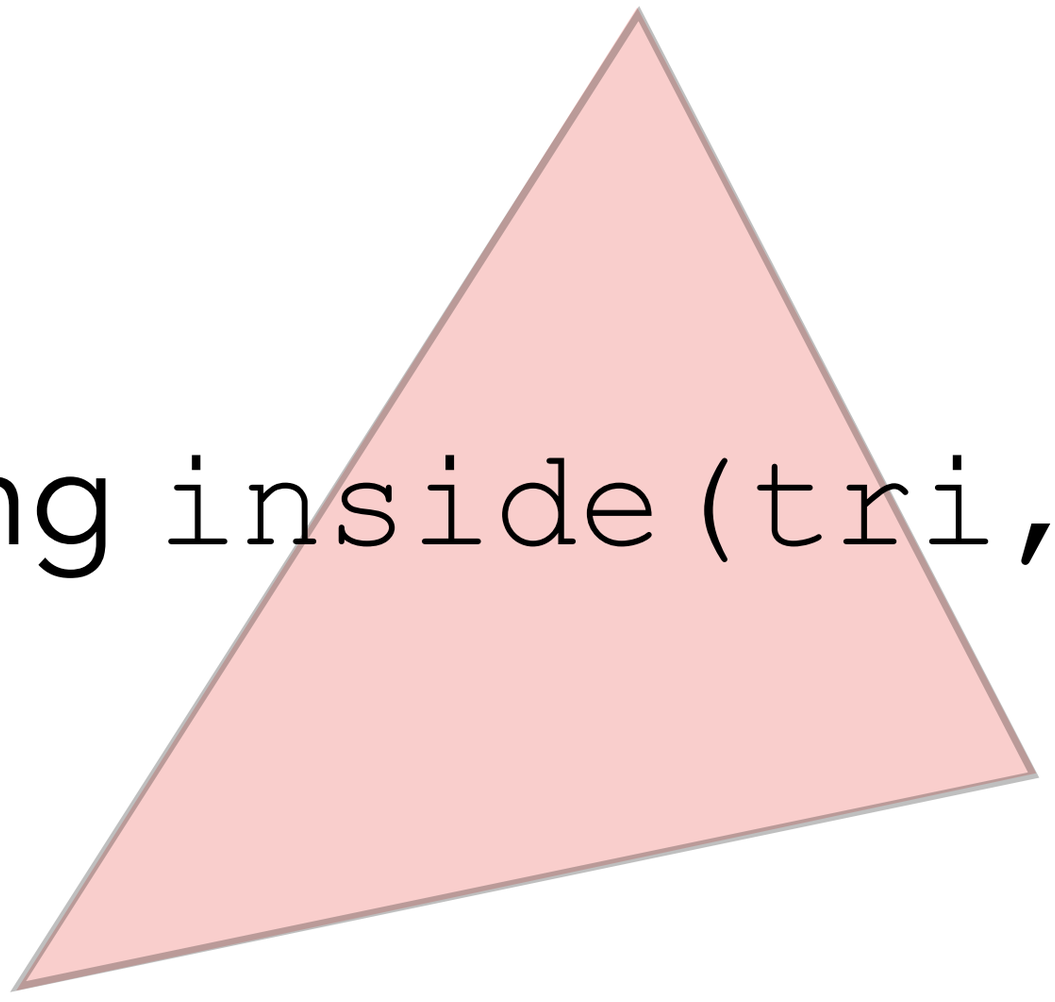
```
for (int x = 0; x < xmax; ++x)
  for (int y = 0; y < ymax; ++y)
    image[x][y] = inside(tri,
                         x + 0.5,
                         y + 0.5);
```

Recall: Sample Locations

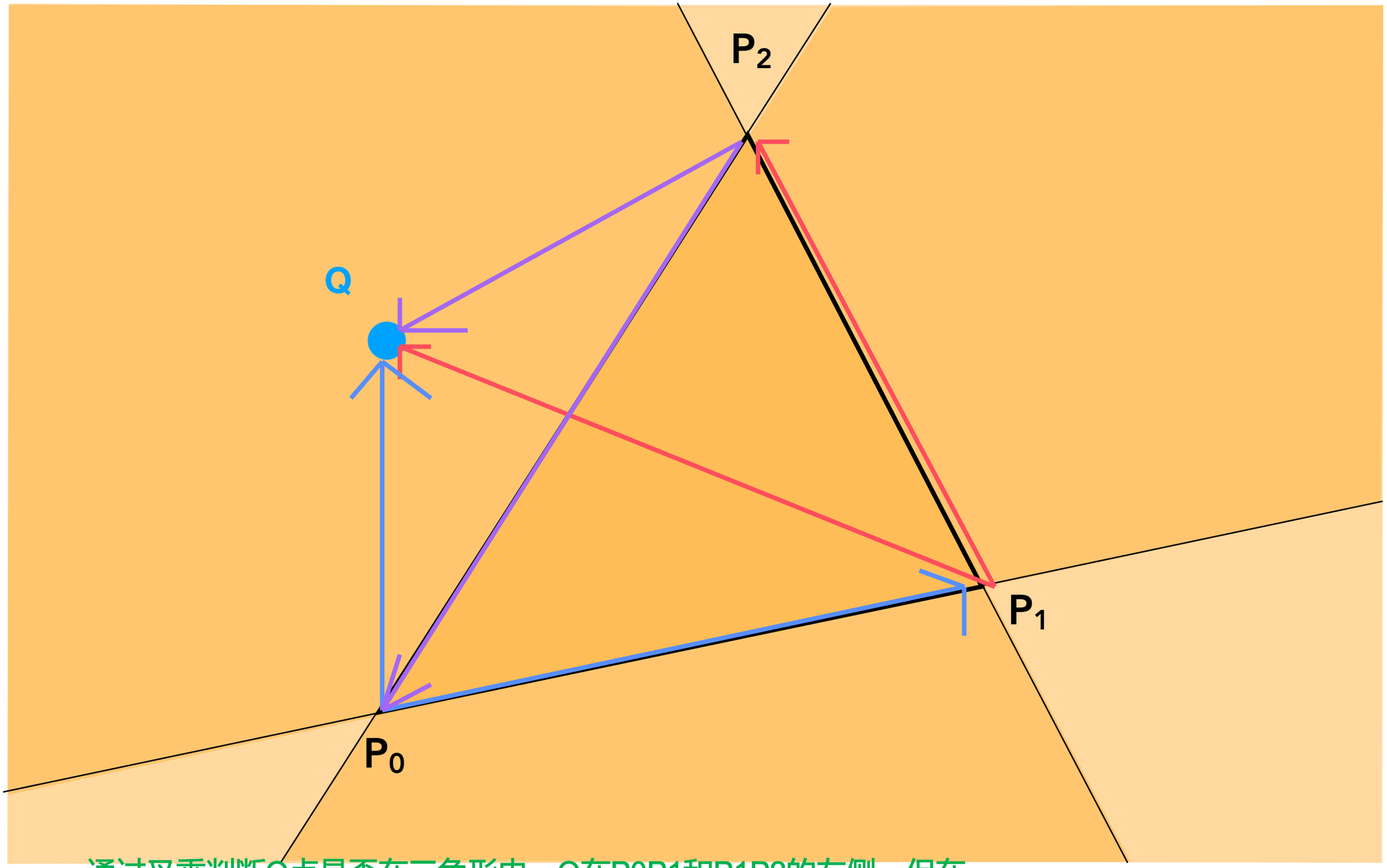


Sample location for pixel (x, y)

Evaluating `inside(tri, x, y)`



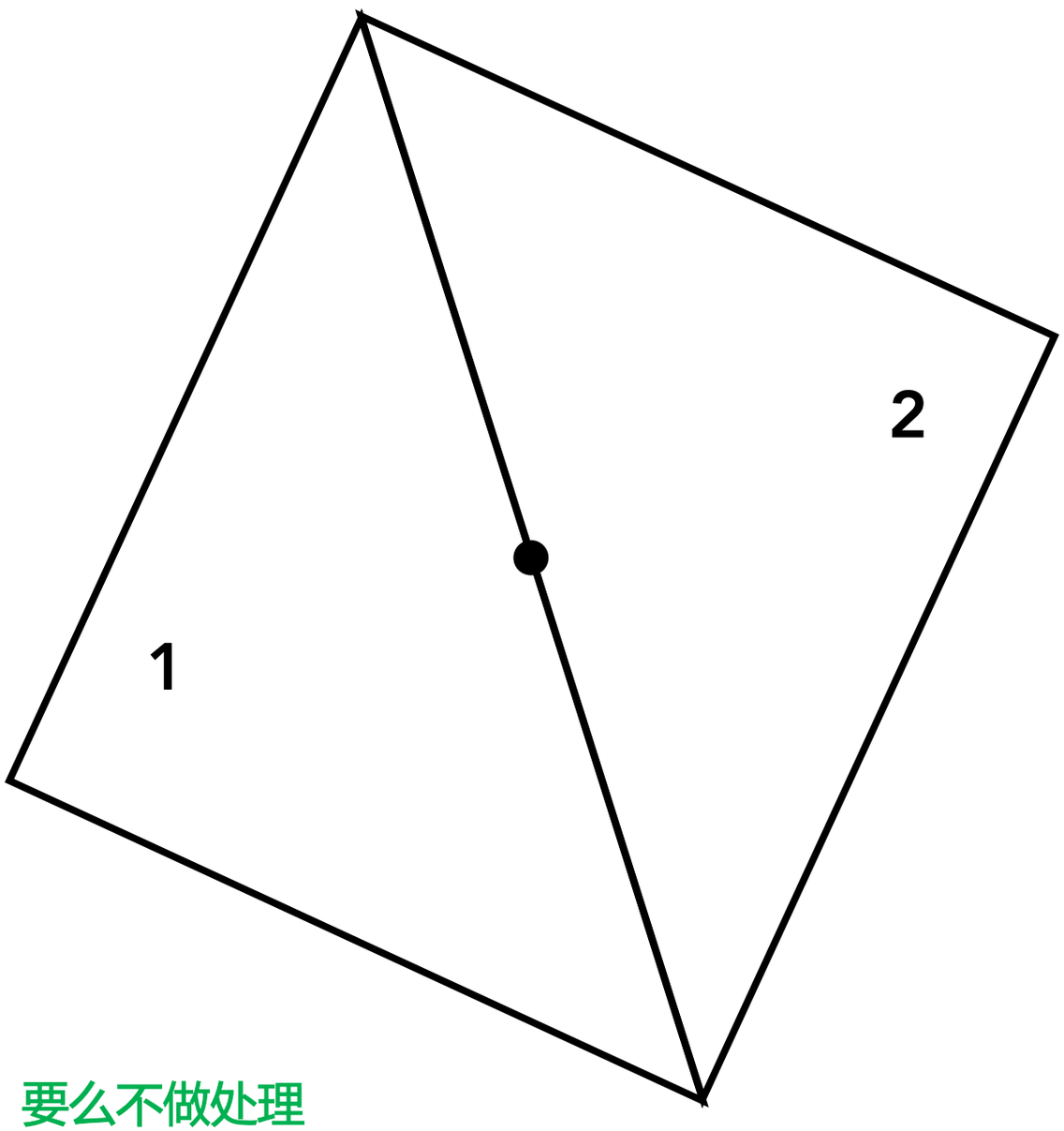
Inside? Recall: Three Cross Products!



通过叉乘判断Q点是否在三角形内，Q在P0P1和P1P2的左侧，但在P2P0的右侧，所以不在三角形内

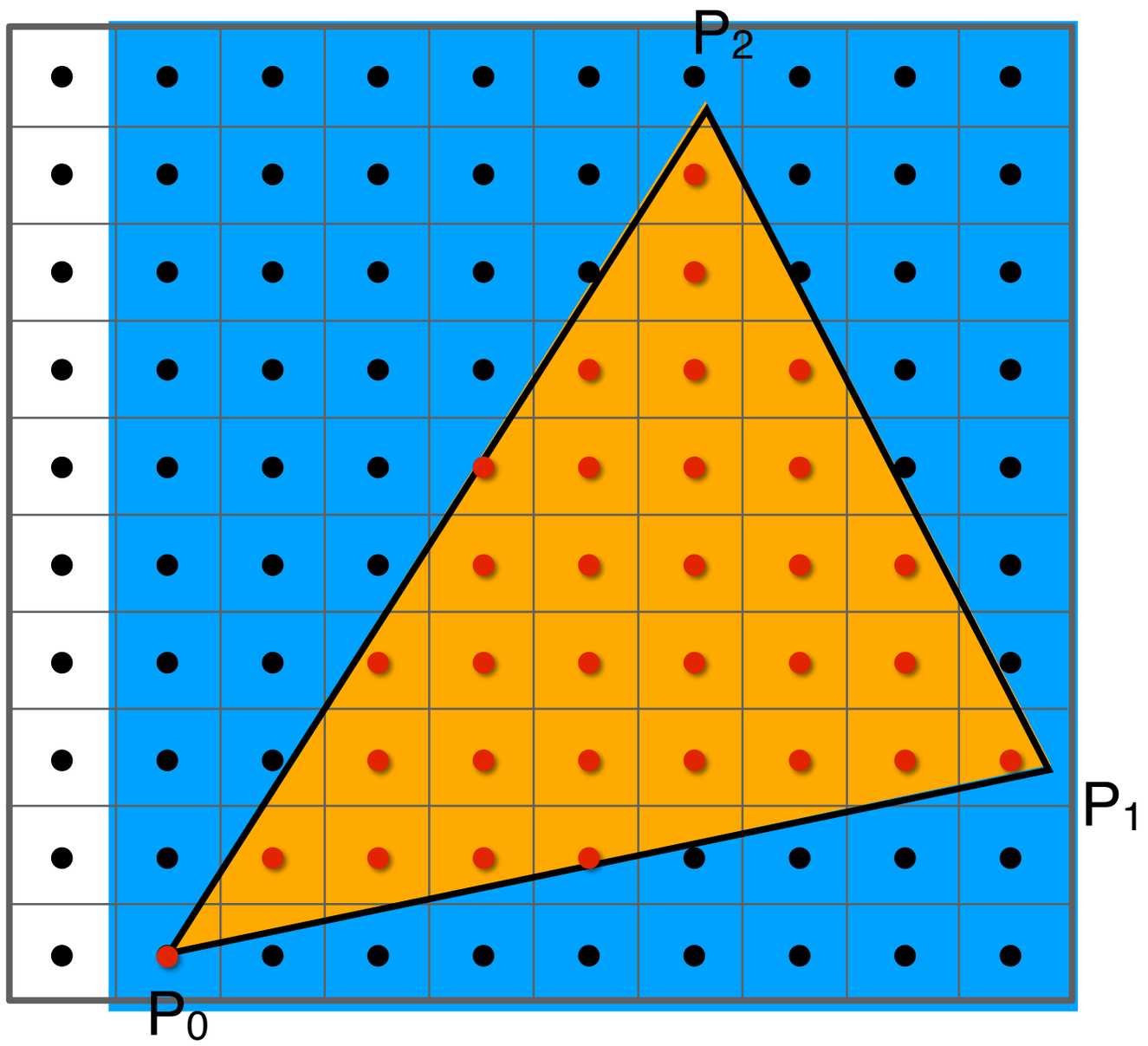
Edge Cases (Literally)

Is this sample point covered by triangle 1, triangle 2, or both?



要么自己规定，要么不做处理

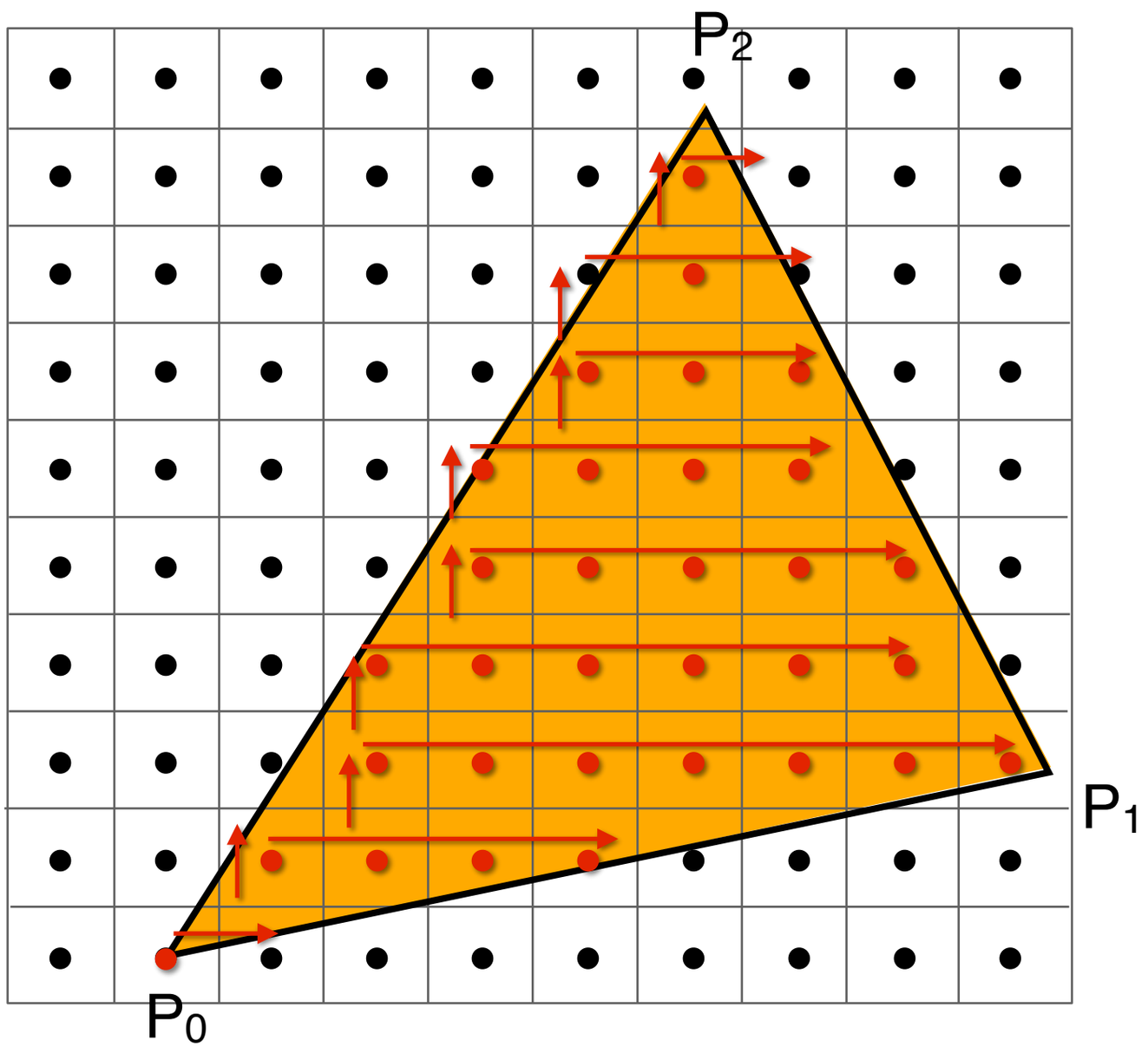
Checking All Pixels on the Screen?



Use a **Bounding Box!**

并非遍历所有像素，使用包围盒，判断三个顶点x和y最大值

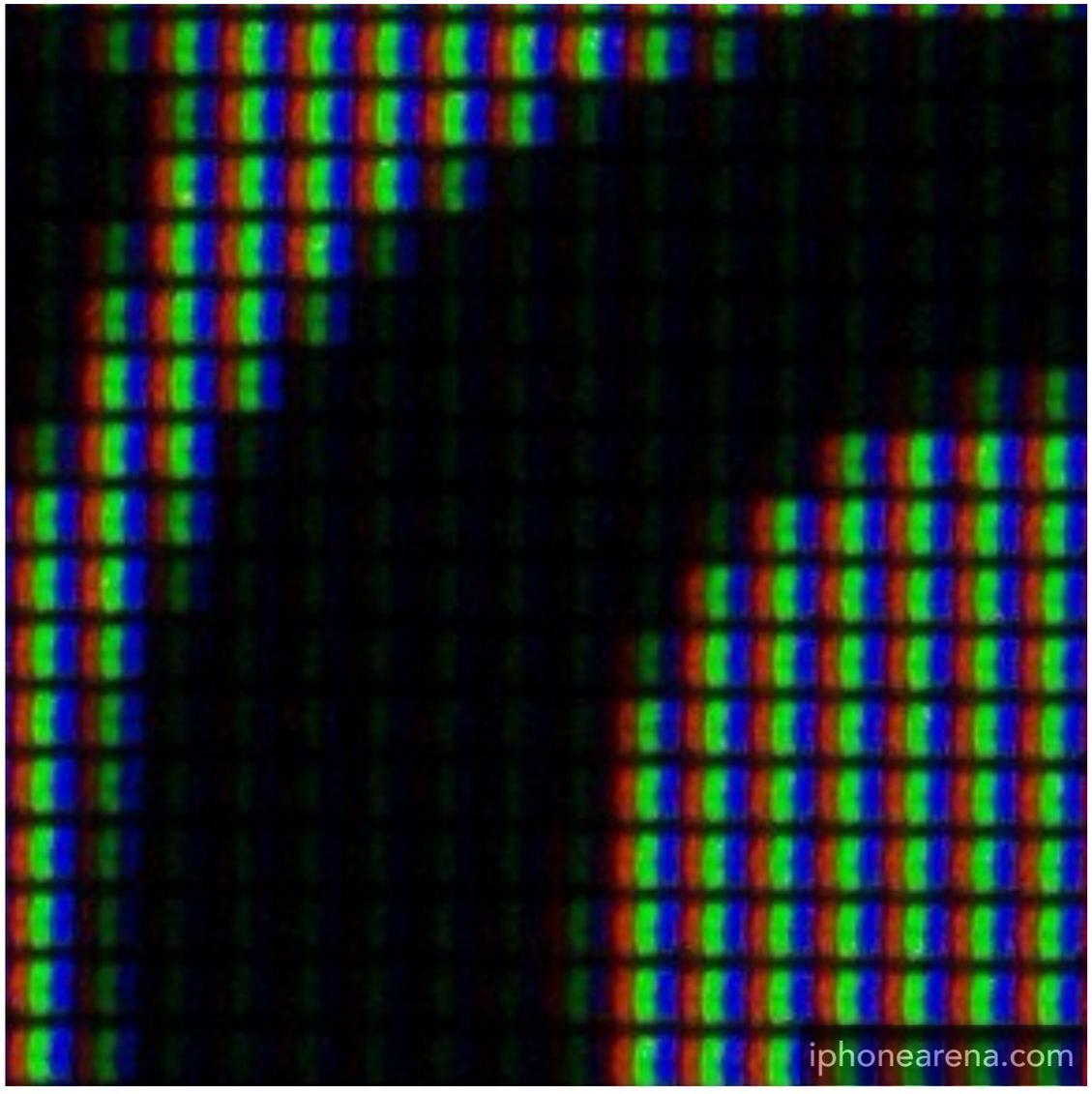
Incremental Triangle Traversal (Faster?)



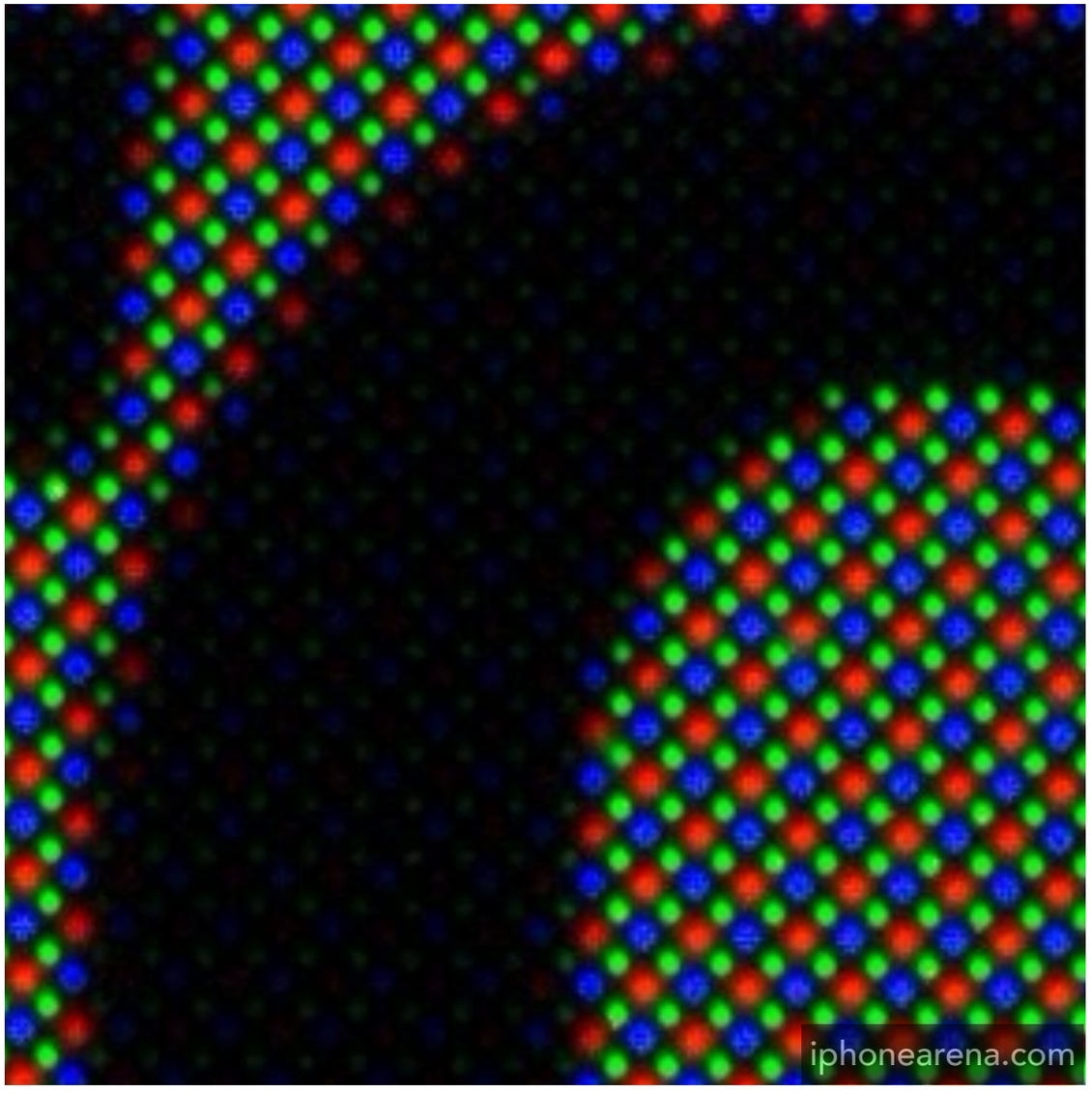
suitable for thin and rotated triangles

Rasterization on Real Displays

Real LCD Screen Pixels (Closeup)



iPhone 6S



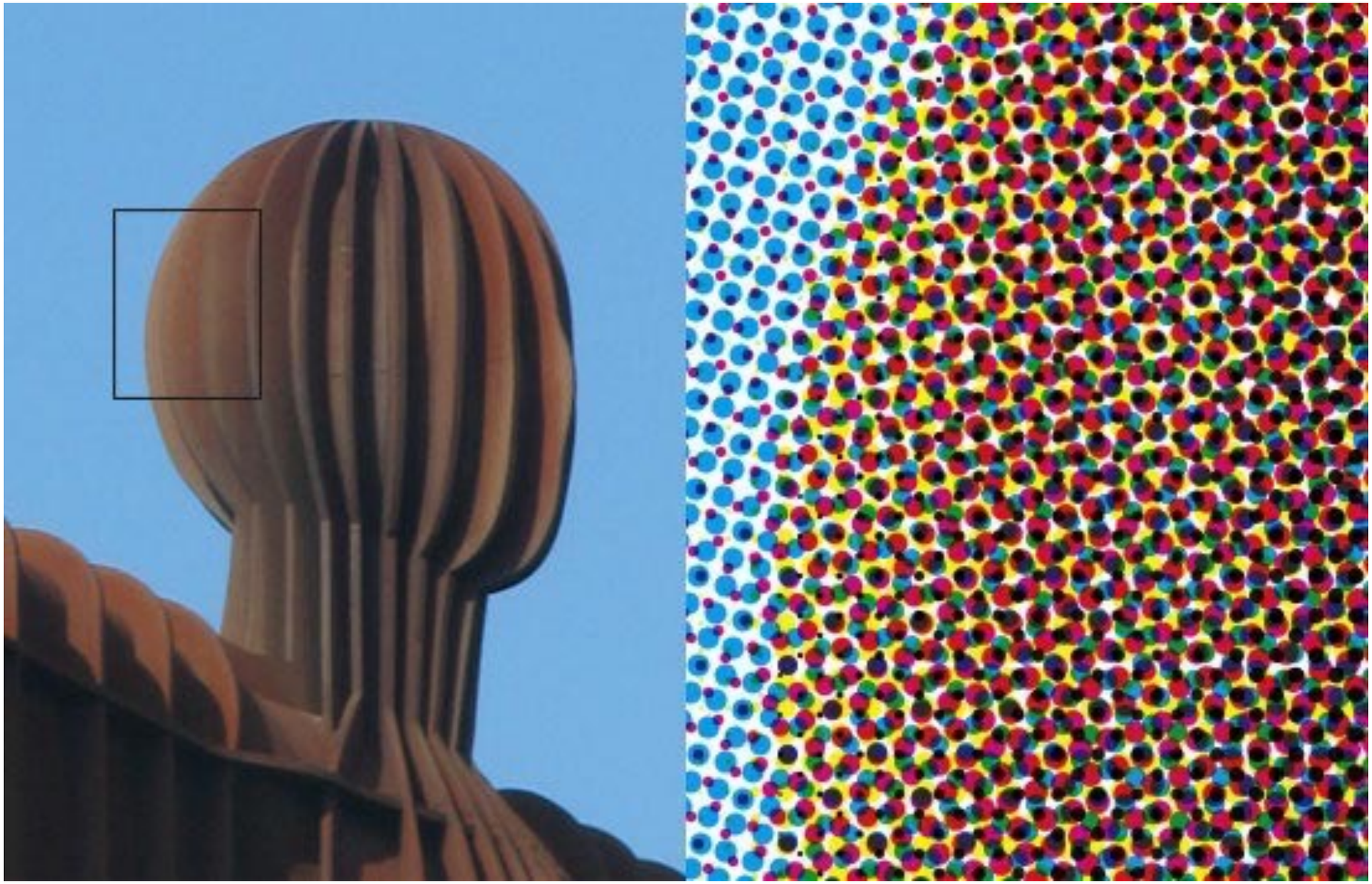
Galaxy S5

人眼对绿色更为敏感

Notice R,G,B pixel geometry! But in this class, we will assume a colored square full-color pixel.

Bayer pattern

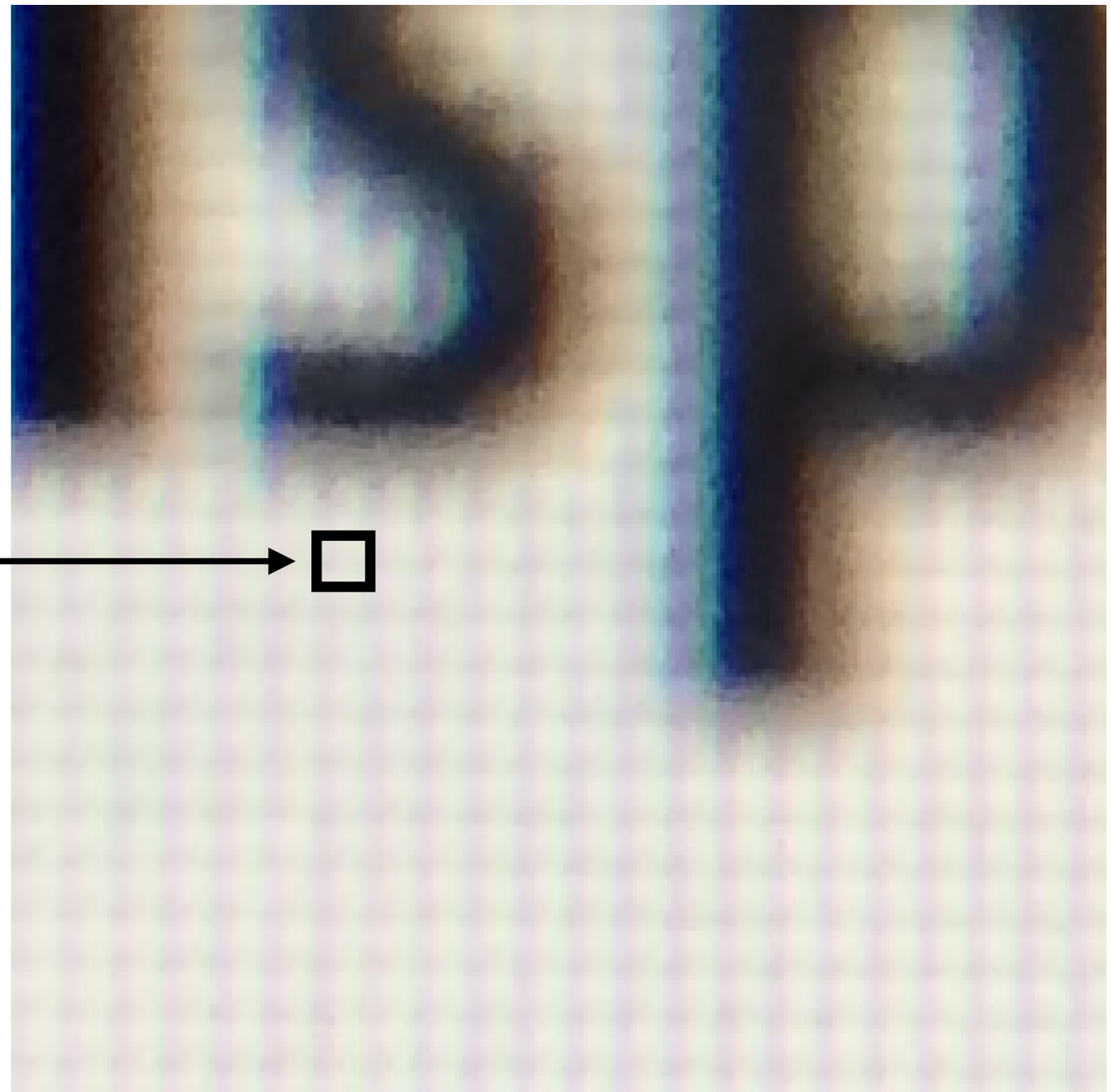
Aside: What About Other Display Methods?



Color print: observe half-tone pattern

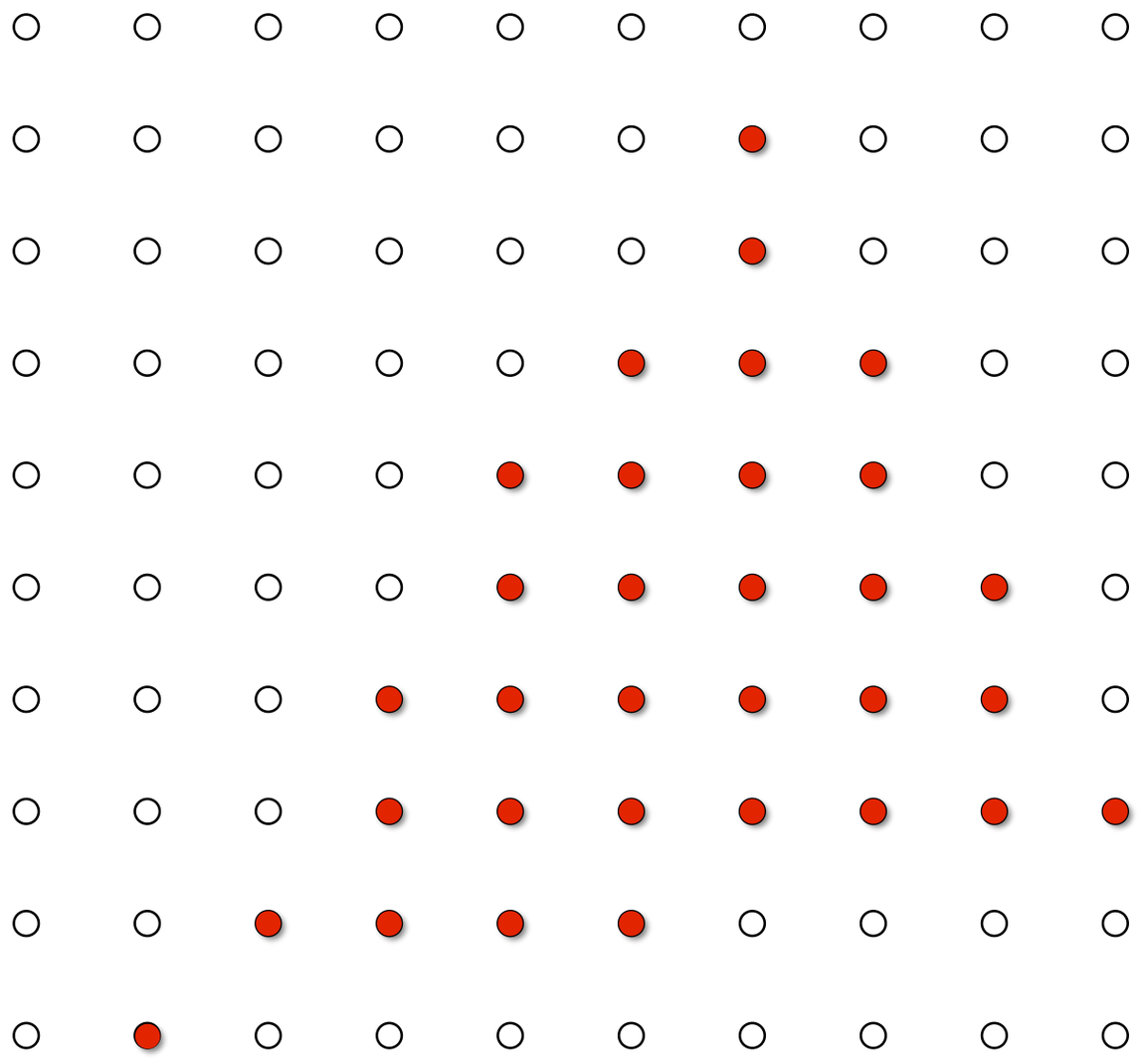
Assume Display Pixels Emit Square of Light

LCD pixel
on laptop

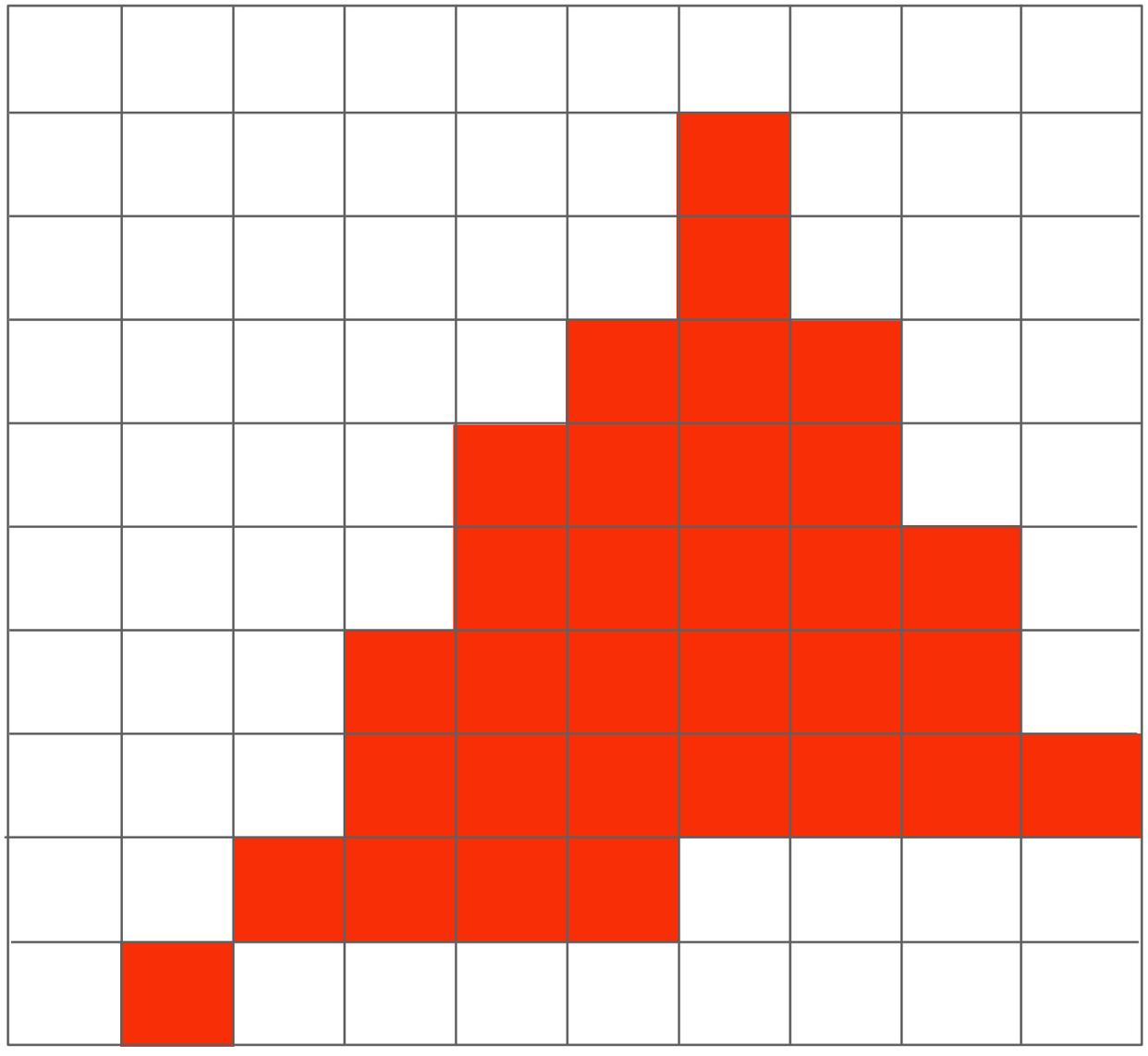


* LCD pixels do not actually emit light in a square of uniform color, but this approximation suffices for our current discussion

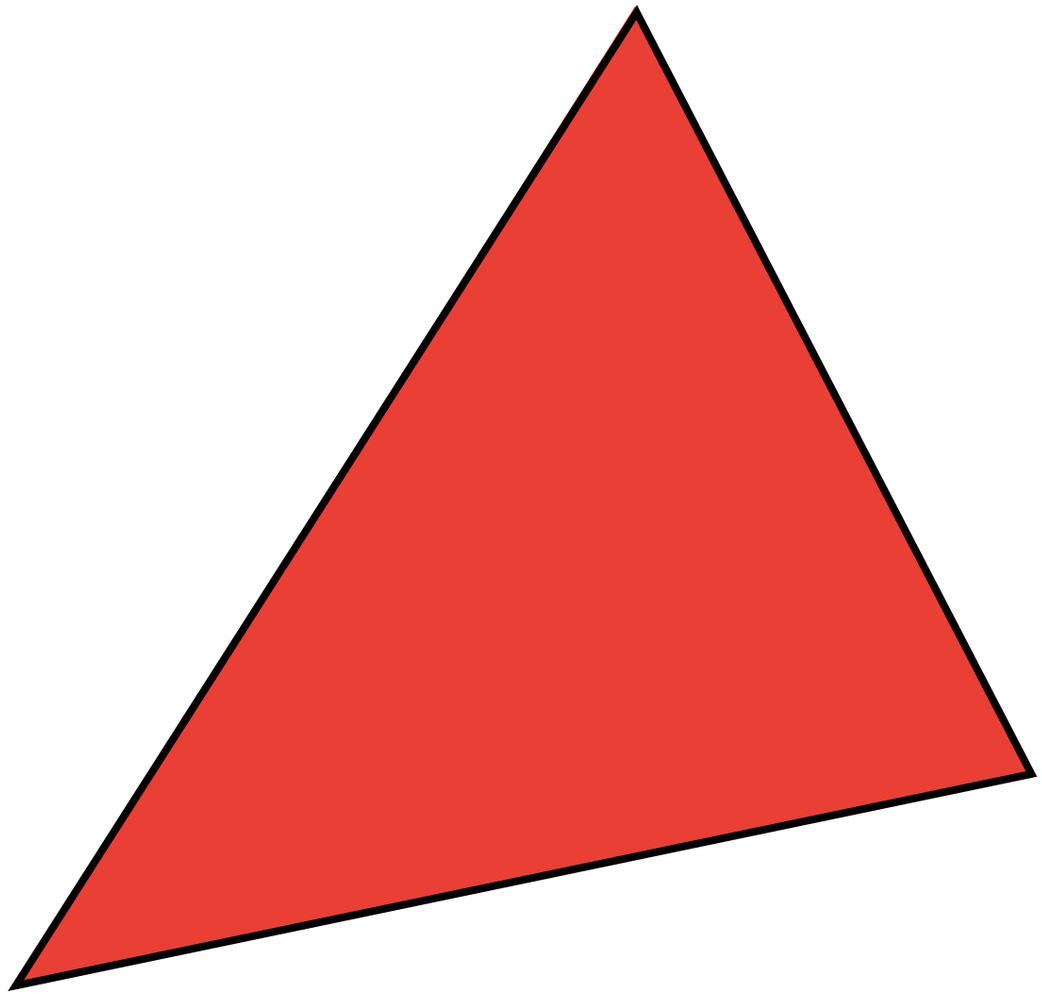
So, If We Send the Display the Sampled Signal



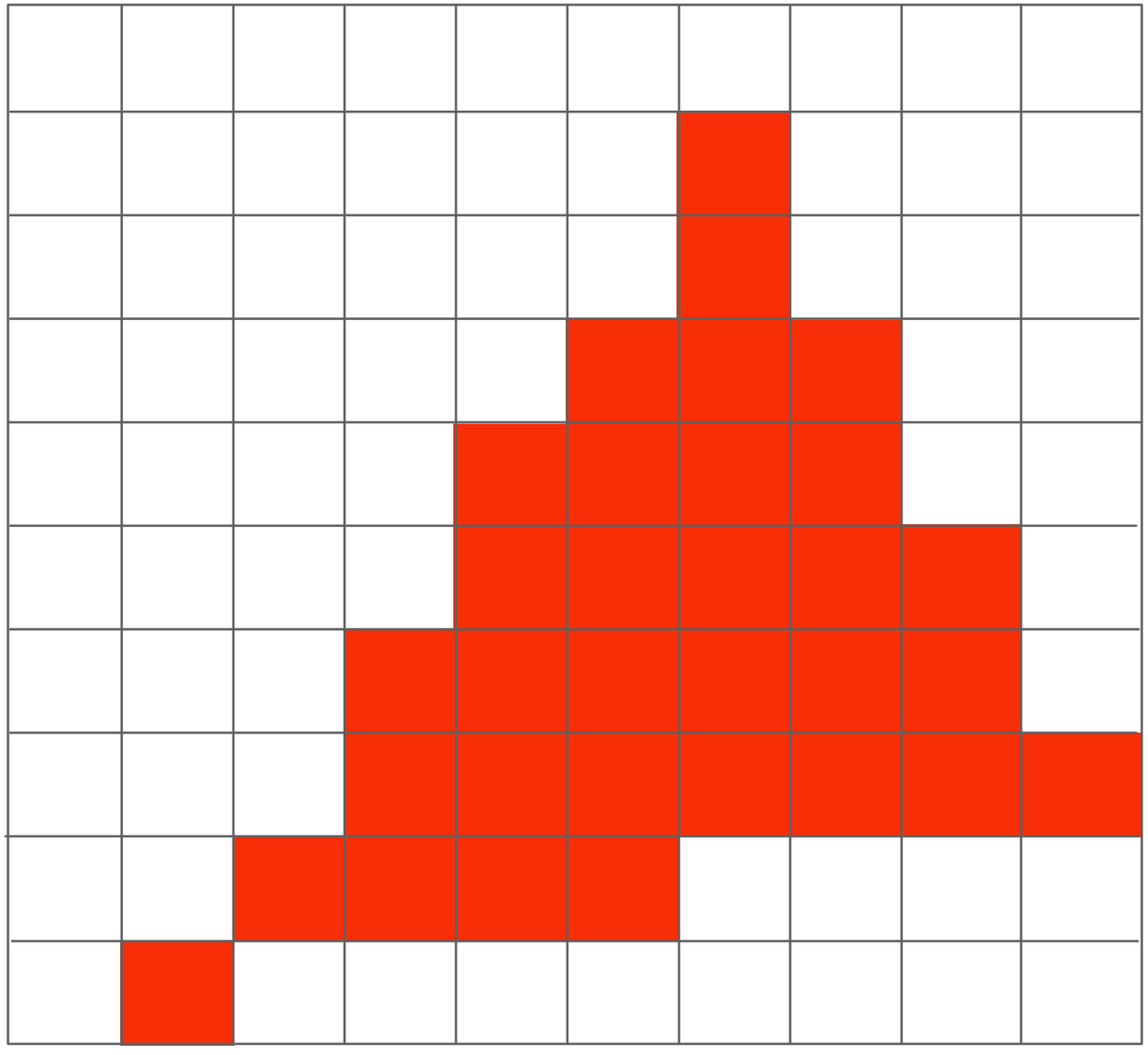
The Display Physically Emits This Signal



Compare: The Continuous Triangle Function

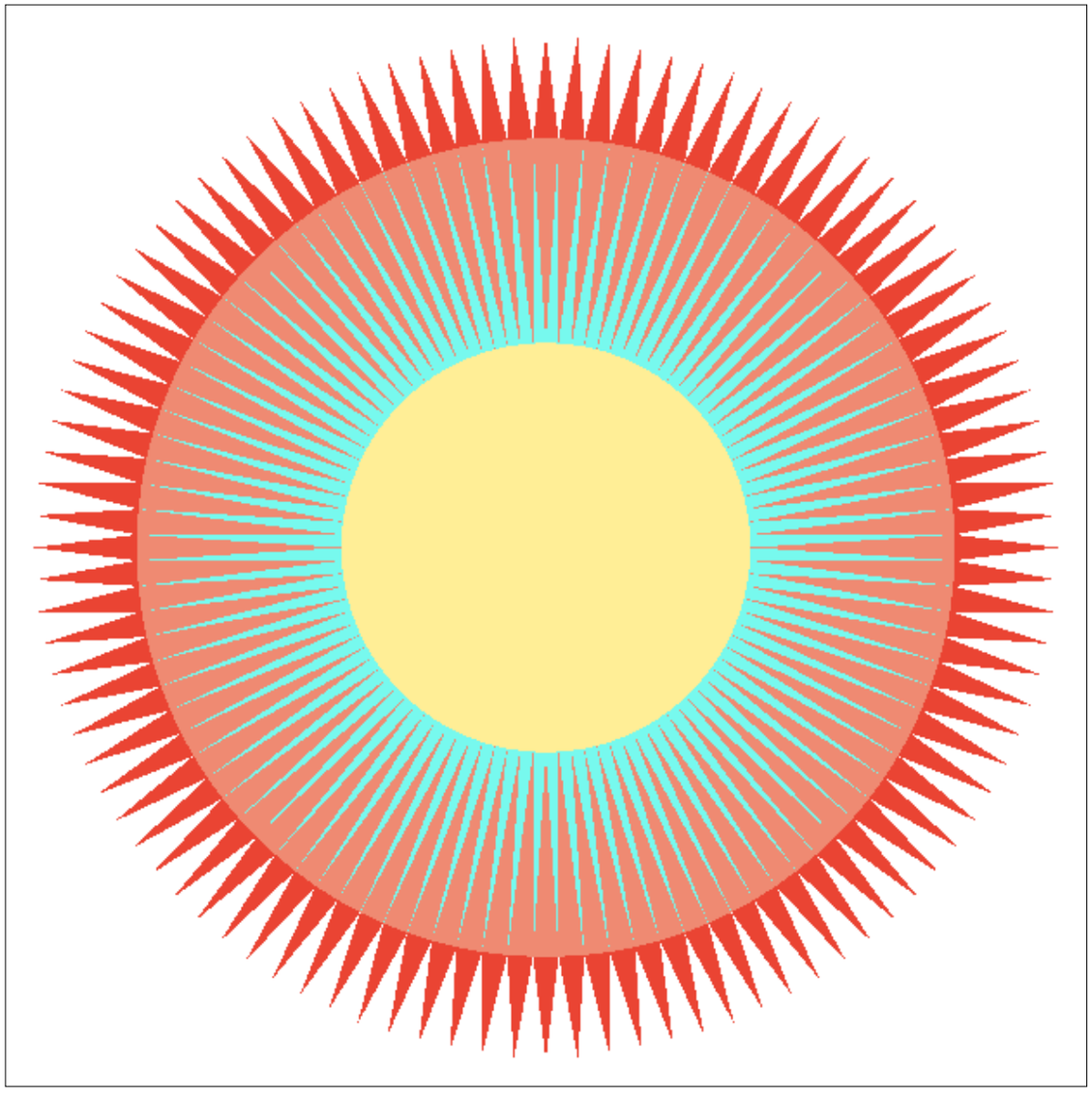


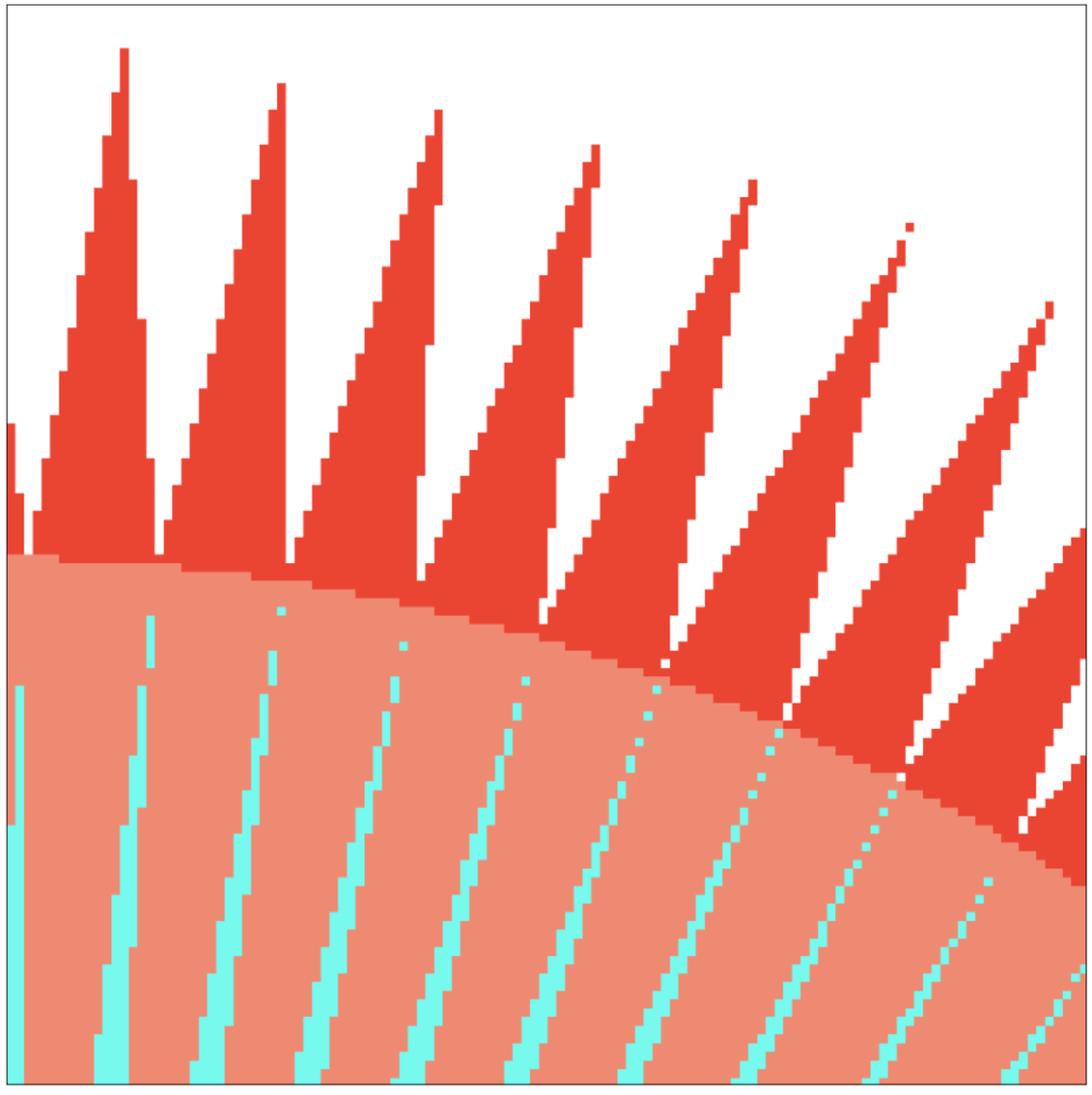
What's Wrong With This Picture?



Jaggies!

Aliasing (Jaggies)





Is this the best we can do?

Thank you!

(And thank Prof. Ravi Ramamoorthi and Prof. Ren Ng for many of the slides!)