Computer Graphics

Lecture 1

Images and Vectors

or: I ordered an image and all I got was this grid of colored boxes

Announcements

Announcements

- Please join the Discord server before Friday's class. Invite link is on the Syllabus page of Canvas.

Announcements

- Please join the Discord server before Friday's class. Invite link is on the Syllabus page of Canvas.
    - Set your nickname to First/preferredname Lastname

Announcements

- Please join the Discord server before Friday's class. Invite link is on the Syllabus page of Canvas.
    - Set your nickname to First/preferredname Lastname
    - Set your avatar to a photo of your face.

Announcements

- Please join the Discord server before Friday's class. Invite link is on the Syllabus page of Canvas.
  - Set your nickname to First/preferredname Lastname
  - Set your avatar to a photo of your face.
    - you can create a separate discord account for this class if you don't want to use your personal Discord account

Announcements

- Please join the Discord server before Friday's class. Invite link is on the Syllabus page of Canvas.
  - Set your nickname to First/preferredname Lastname
  - Set your avatar to a photo of your face.
    - you can create a separate discord account for this class if you don't want to use your personal Discord account
- HW0 and A0 out later today

Announcements

- Please join the Discord server before Friday's class. Invite link is on the Syllabus page of Canvas.
  - Set your nickname to First/preferredname Lastname
  - Set your avatar to a photo of your face.
    - you can create a separate discord account for this class if you don't want to use your personal Discord account
- HW0 and A0 out later today
  - Both due next Wednesday night

How do we graphics?

Let's design a simple graphics system.

The goal: draw a triangle on the screen.

How do we graphics?

Let's design a simple graphics system.

The goal: draw a triangle on the screen.

(why a triangle? more on this next time...)

How do we graphics?

Let's design a simple graphics system.

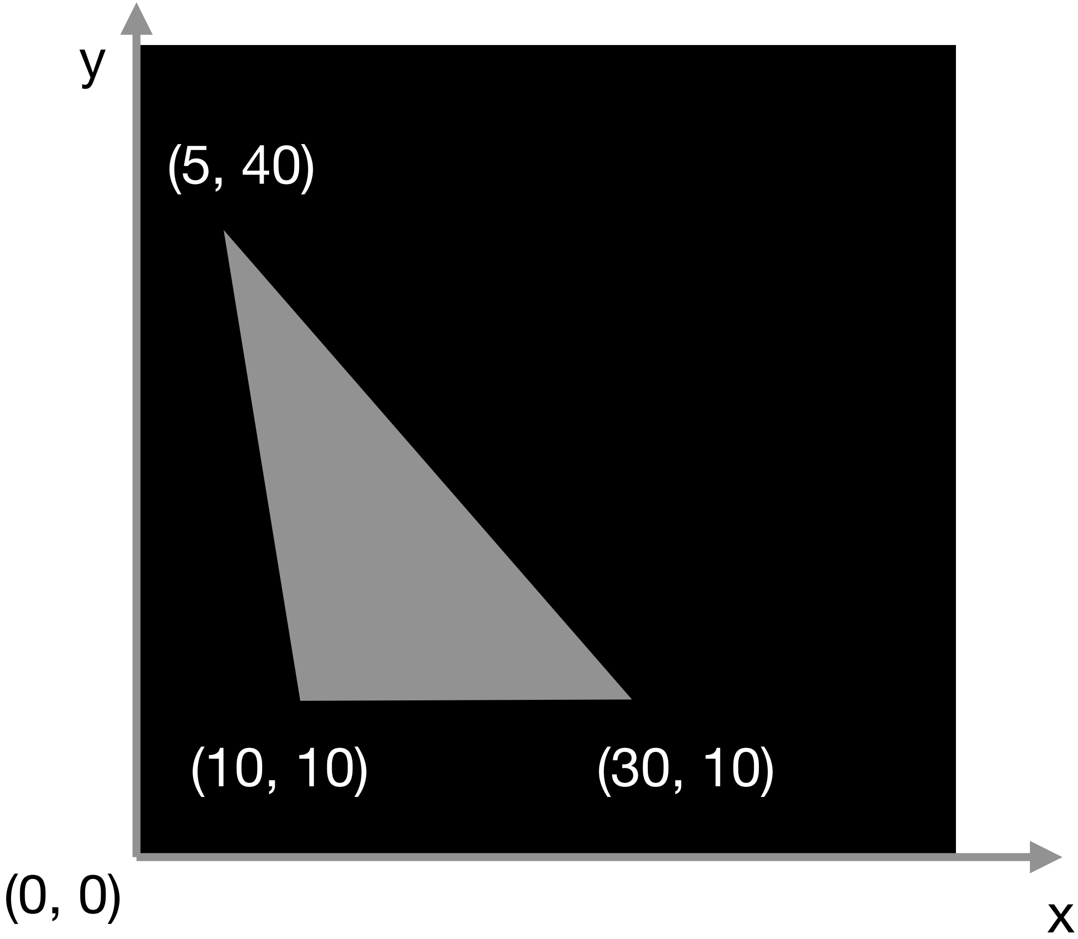
The goal: draw a triangle on the screen.

(why a triangle? more on this next time...)

Pseudocode for graphics:

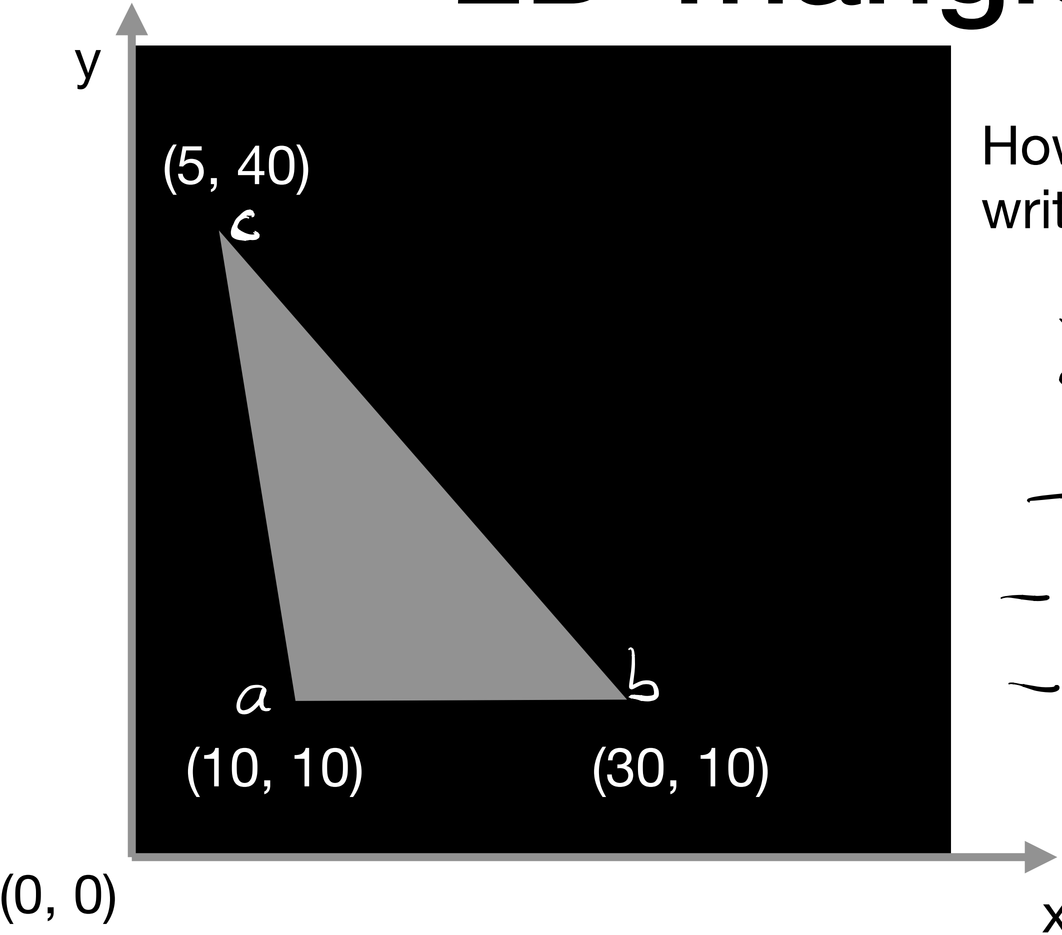
- Create a model of a scene represent triangle
- Render an image of the scene turn on pixels inside triangle

Create a model of the scene



Convention: list vertices in **counterclockwise** order.

2D Triangles



How many ways can I write down this triangle?

~~c b a~~

- a b c
- b c a
- c a b

Convention: list vertices in **counterclockwise** order.

Render an image of the model

what **is** that?

Render an image of the model

What **is** an image anyway?

- A photographic print?
- A photographic negative?
- The screen you're watching this on?
- Some numbers in RAM?

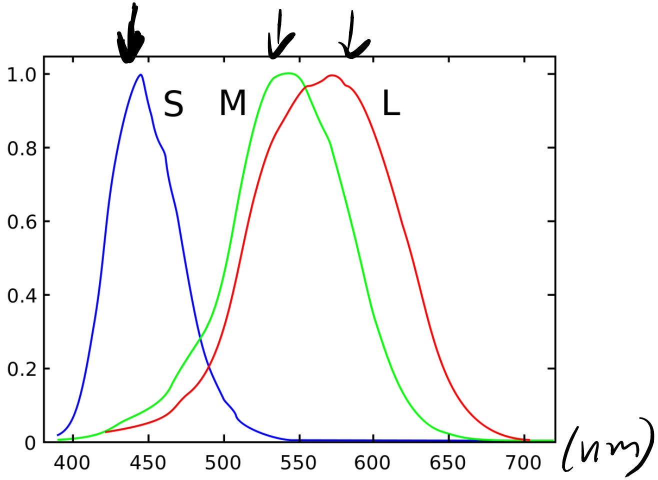
What is an image?

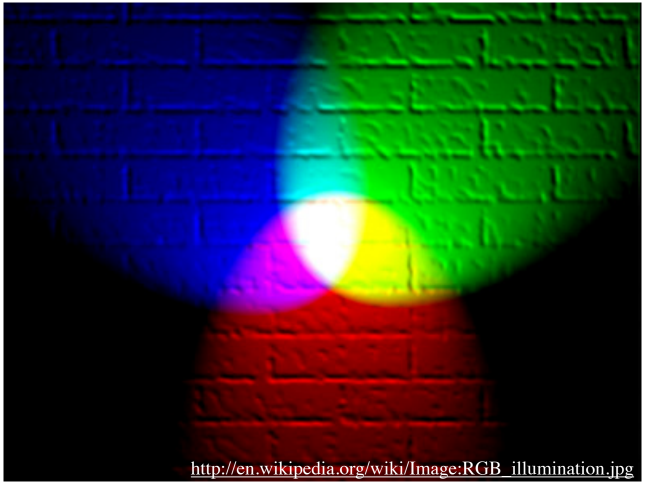
At its most formal and general: a **function** that maps *positions* in 2D to *distributions of radiant energy*

$$I : \mathbb{R}^2 \rightarrow ??$$

What about color?

- Humans are trichromatic, so we usually represent color as combinations or red, green, and blue





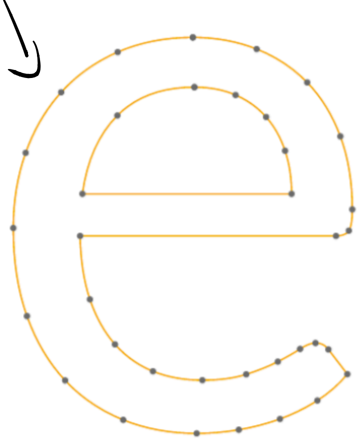
http://en.wikipedia.org/wiki/Image:RGB_illumination.jpg

How do we represent images?

- Raster formats - a 2D array of numbers

- Vector formats - mathematical description

Raster Image

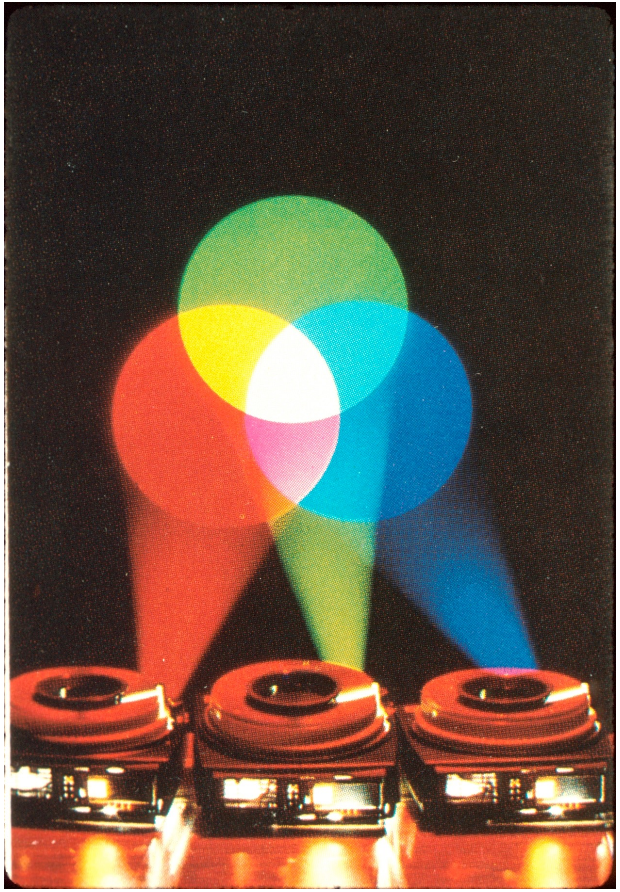


Pavithra Solai, kint.io

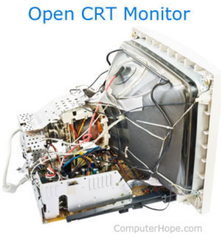
Vector Image

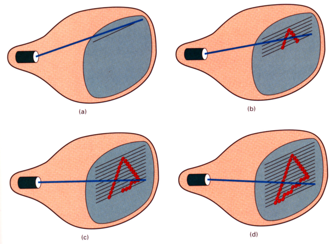
How do we display images?
Old School Edition

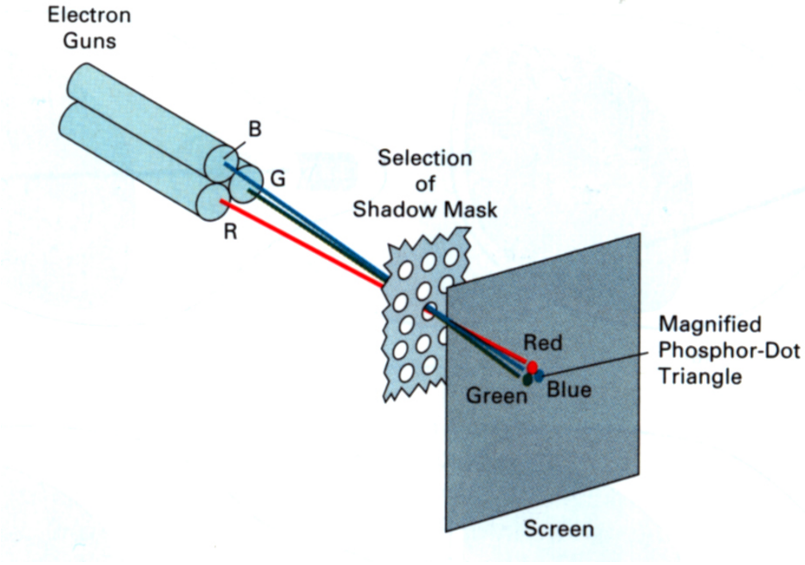
Color Projector



Cathode Ray Tube

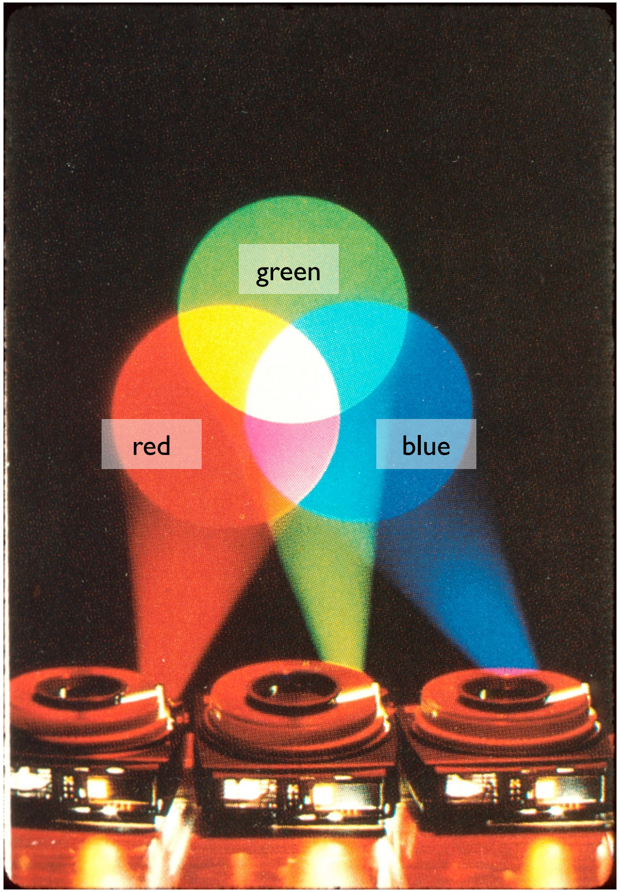




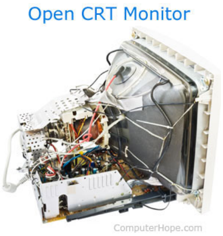


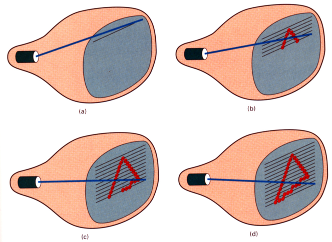
How do we display images?
Old School Edition

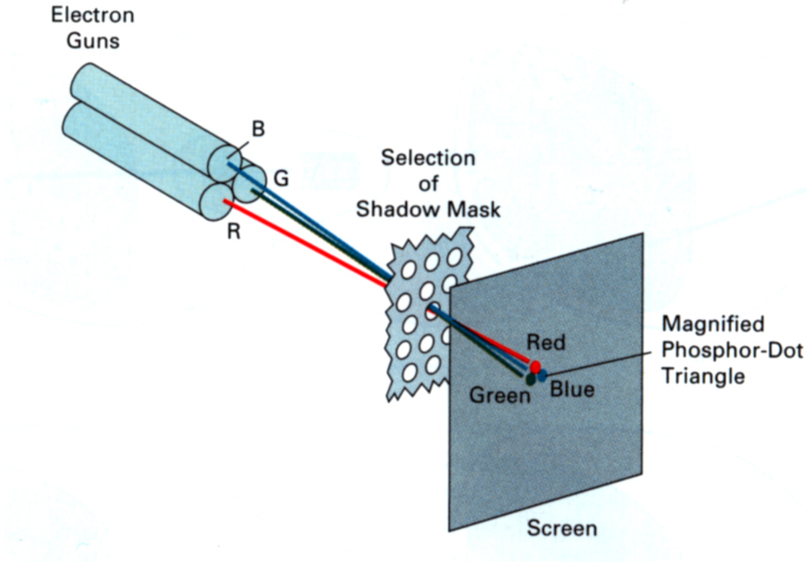
Color Projector



Cathode Ray Tube

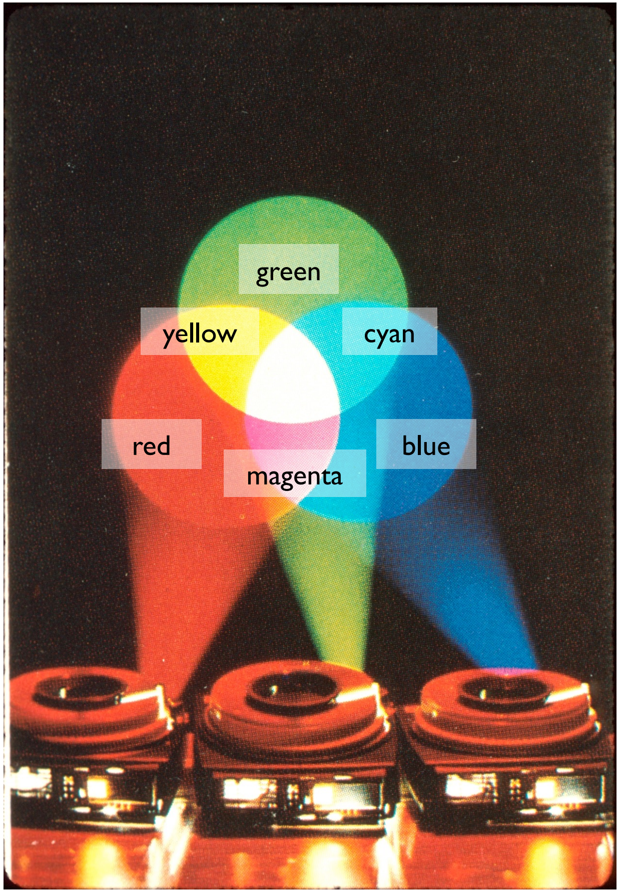




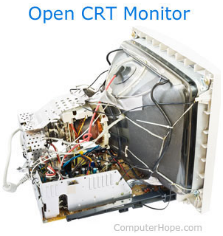


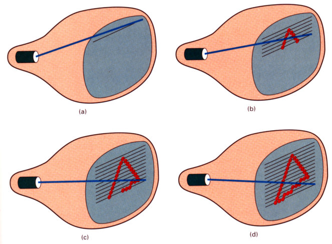
How do we display images?
Old School Edition

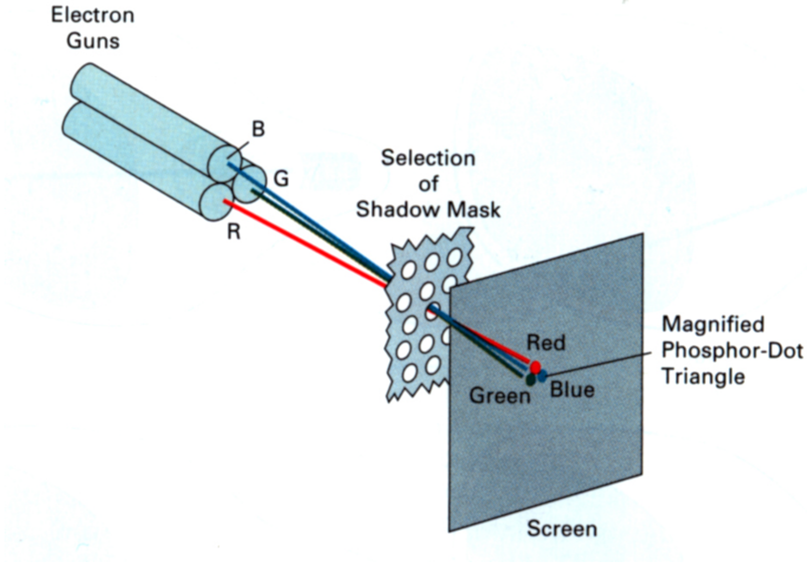
Color Projector



Cathode Ray Tube

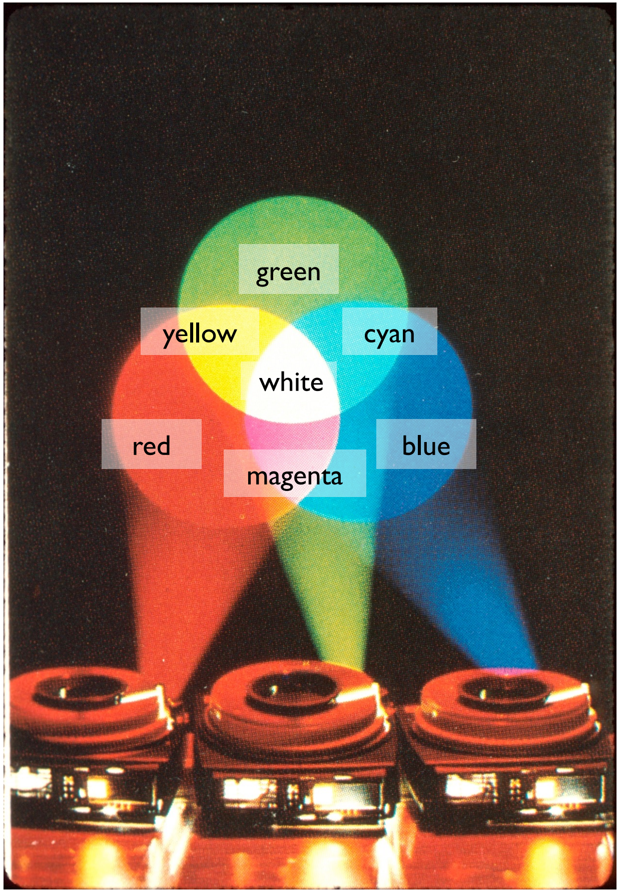




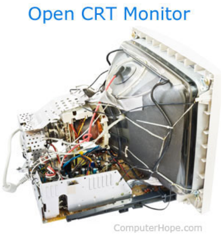


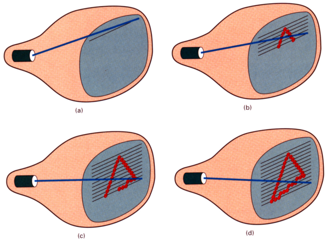
How do we display images?
Old School Edition

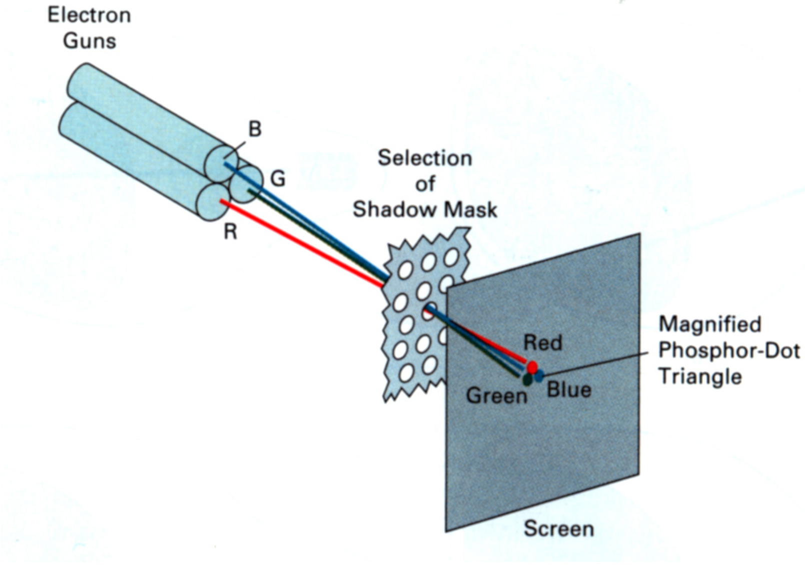
Color Projector



Cathode Ray Tube

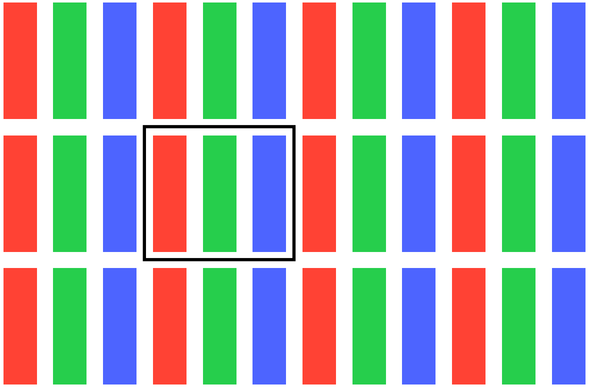




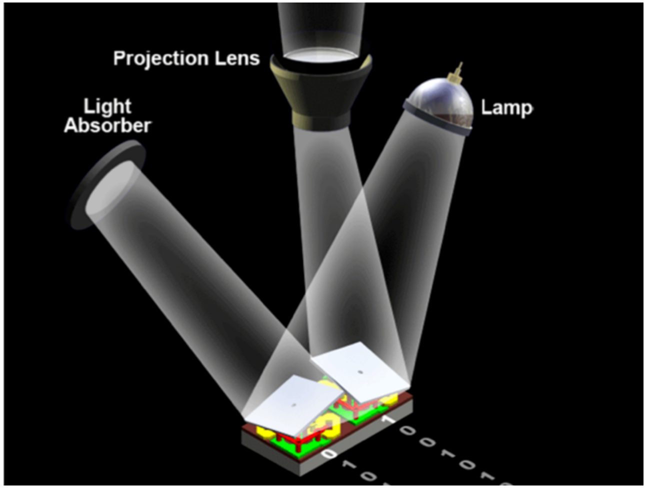


How do we display images?
Nowadays Edition

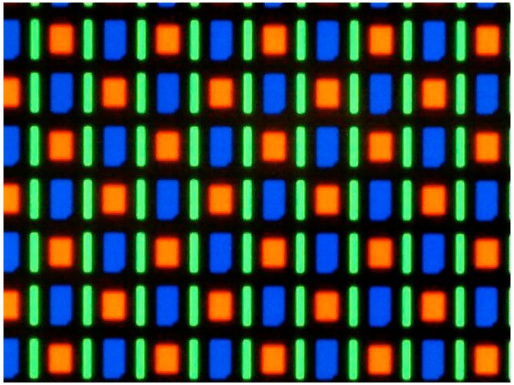
Liquid Crystal Display



Digital Light Processing



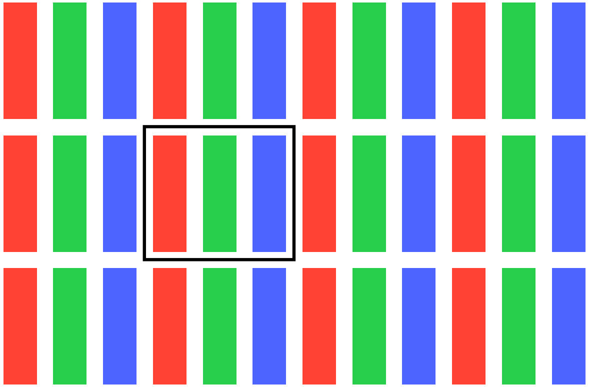
Light Emitting Diode Display



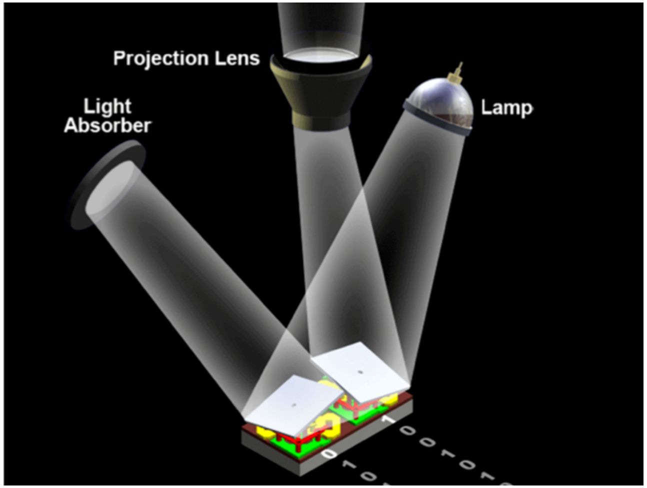
[Wikimedia Commons]

How do we display images?
Nowadays Edition

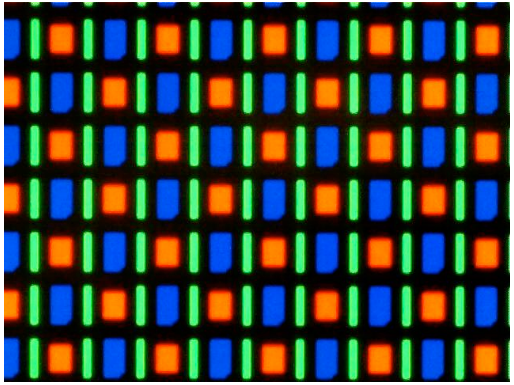
Liquid Crystal Display



Digital Light Processing



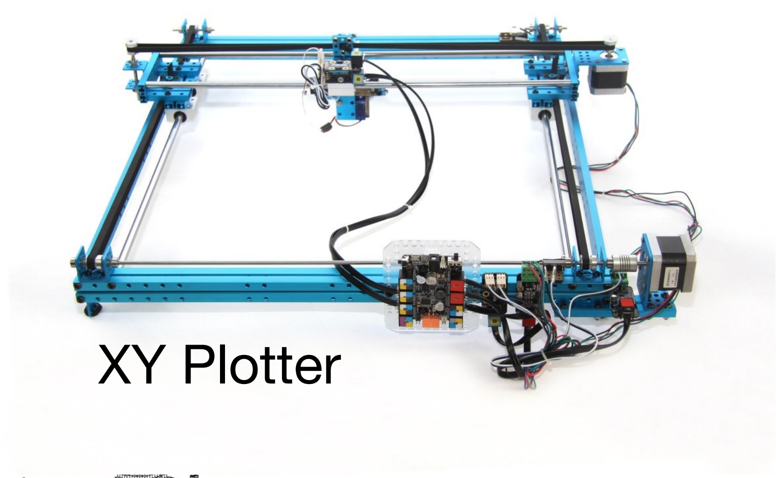
Light Emitting Diode Display



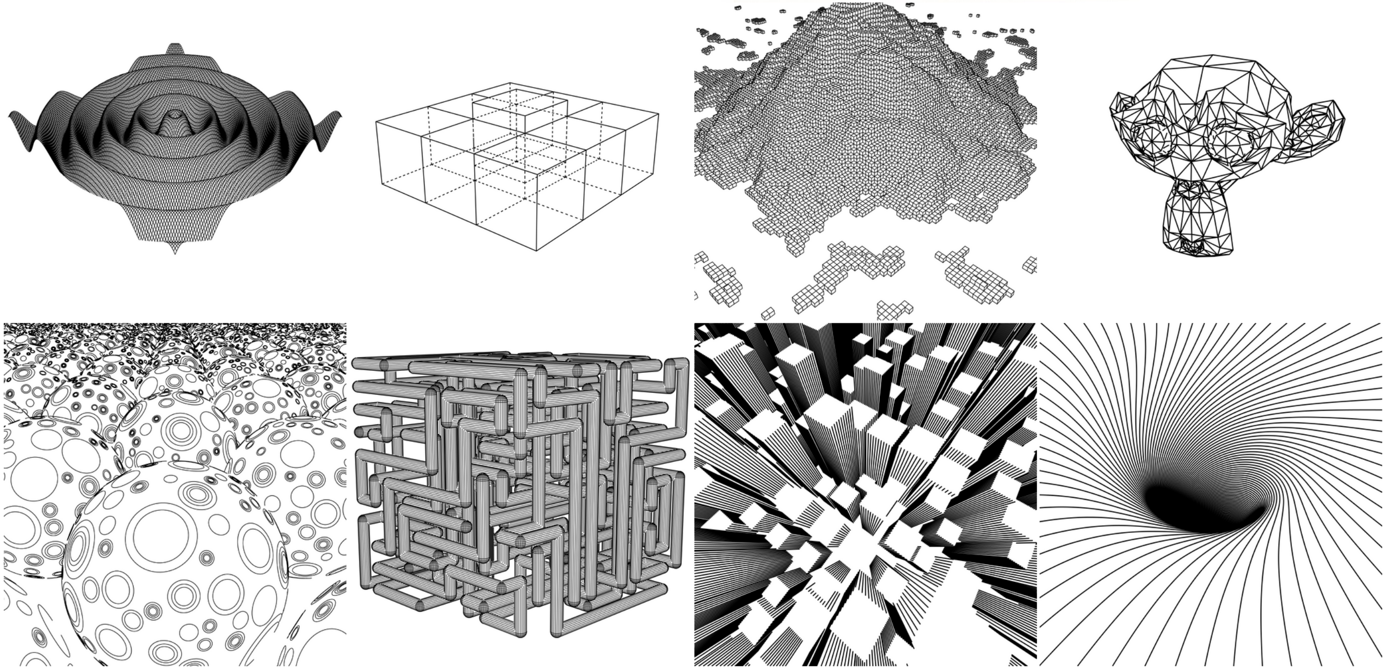
[Wikimedia Commons]

these are all examples of **raster displays**

Aside: It doesn't have to be this way...



XY Plotter

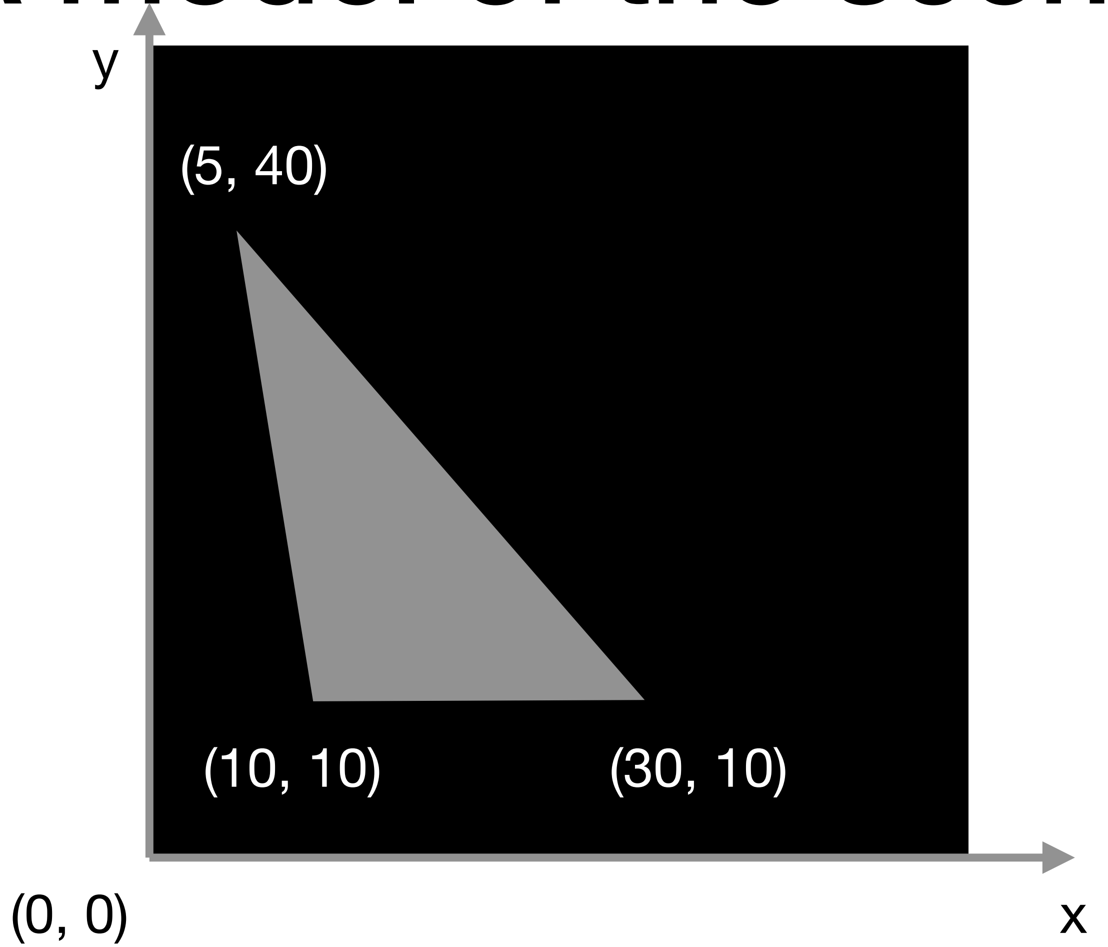


images: https://github.com/fogleman/ln

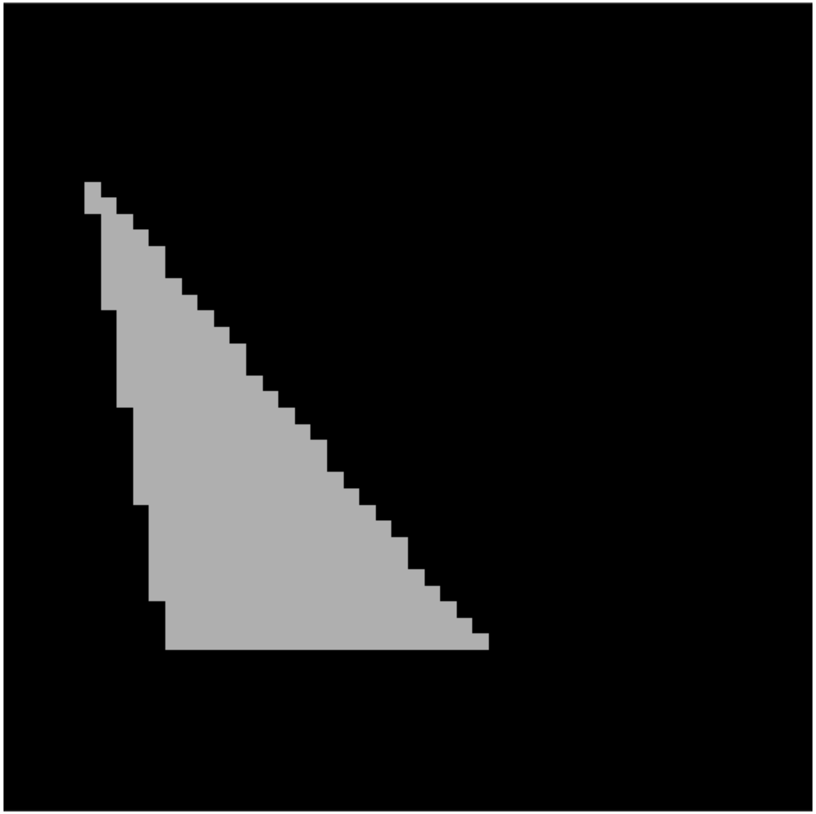
Raster Images

- Flexible
- Display-native
- Expensive
- Not ideal
- But darn useful

A model of the scene



A Raster Image of the Scene



Representing Raster Images: 2D Arrays of Numbers

- Bitmap (1 bit per pixel) $I : \mathbb{R}^2 \to \{0, 1\}$
- Grayscale (usually 8 bpp) $I : \mathbb{R}^2 \to [0 .. \mathrm{MAX}]$ (e.g. 255)
- Color (usually 24 bpp) $I : \mathbb{R}^2 \to [0 .. \mathrm{MAX}]^3$ (R, G, B)
- Floating-point (gray or color) $\mathbb{R}^2 \to \mathbb{R}^3$
  - Bad for display, but good for processing
  - Allows **high dynamic range**
  - For LDR, values range from 0-1 by convention

Raster Images: Storage

1 **megapixel** image - 1024x1024:

- Bitmap (1 bit per pixel) - **128 KB**
- Grayscale (8 bpp) - **1 MB**
- Color (24 bpp) - **3 MB**
- Floating-point (color) - **12MB**

Aside: Performance

Fact: A 1 **megapixel** image has 1024x1024 = 1048576 = 2^{20} pixels.

Aside: Performance

Fact: A 1 **megapixel** image has 1024x1024 = 1048576 = 2^{20} pixels.

Consequence: creating a 1 megapixel image requires making 2^{20} decisions.

Aside: Performance

Fact: A 1 **megapixel** image has 1024x1024 = 1048576 = 2^{20} pixels.

Consequence: creating a 1 megapixel image requires making 2^{20} decisions.

Implication: performance matters.

2D Arrays in Julia

- Image: A height-by-width array of pixels.

- For a color image, each pixel is 3 single-precision floats:

```
                      type                  h        w
canvas = zeros(RGB{Float32}, height, width)
```

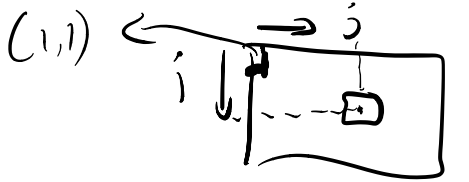
- Matrix-style 1-based indexing (row, column):

2D Arrays in Julia

- Image: A height-by-width array of pixels.

- For a color image, each pixel is 3 single-precision floats:

```
canvas = zeros(RGB{Float32}, height, width)
```



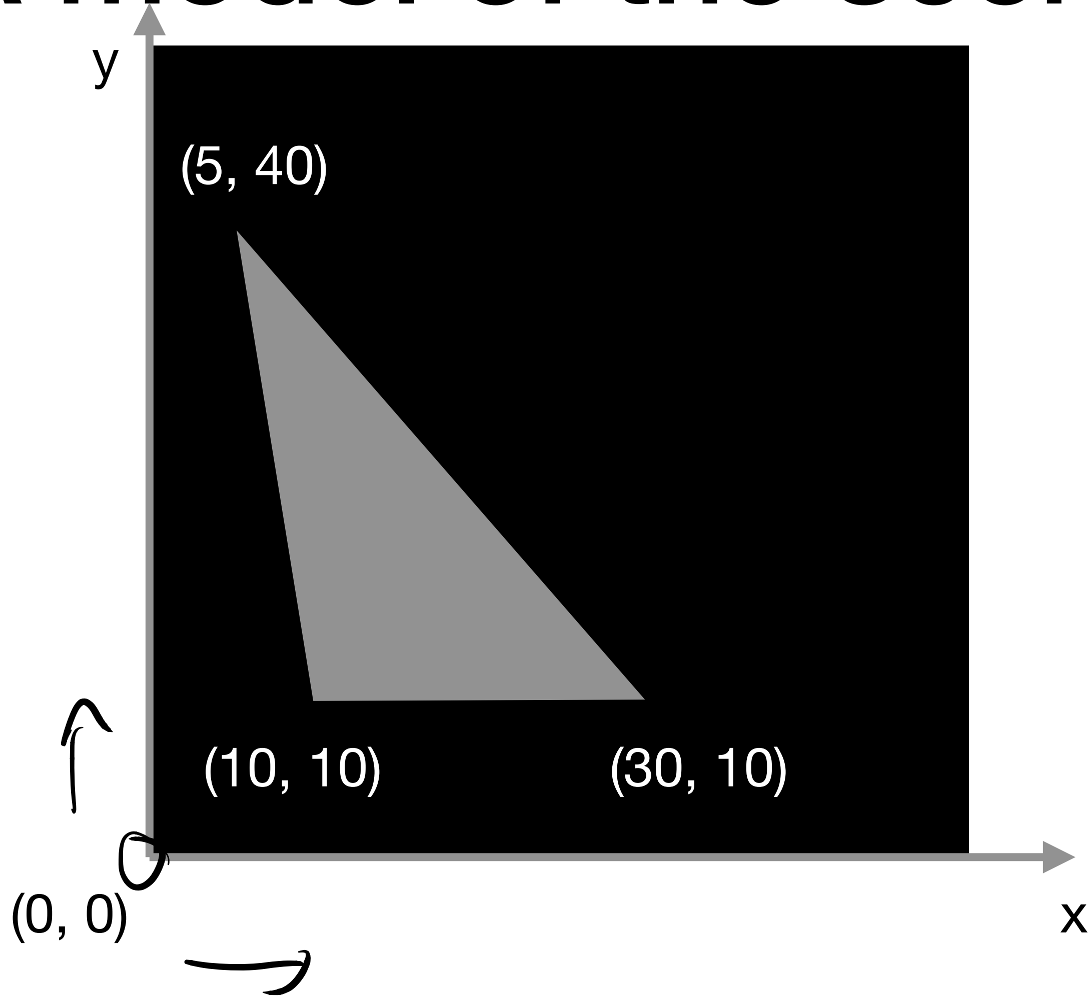
- Matrix-style 1-based indexing (row, column):

```
canvas[i, j] # is the i'th row, j'th column
```

Images in Julia: Demo

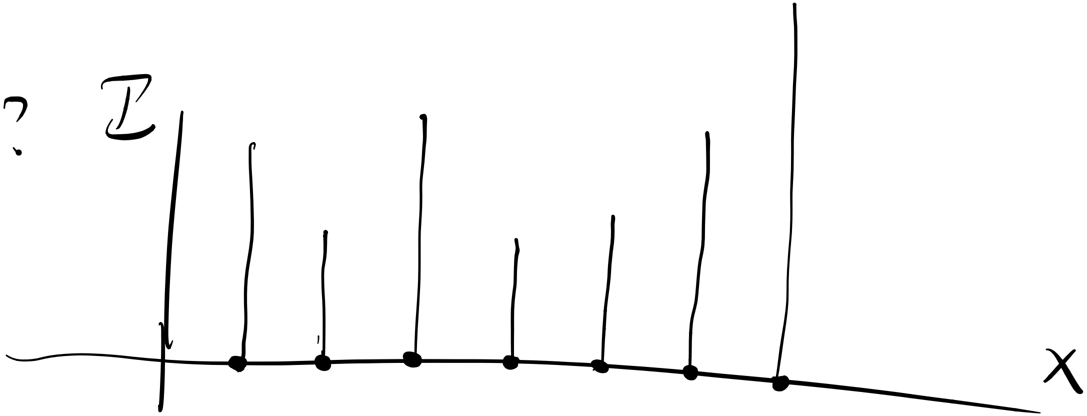
- Draw a rectangle on a canvas
- Demo colors

A model of the scene



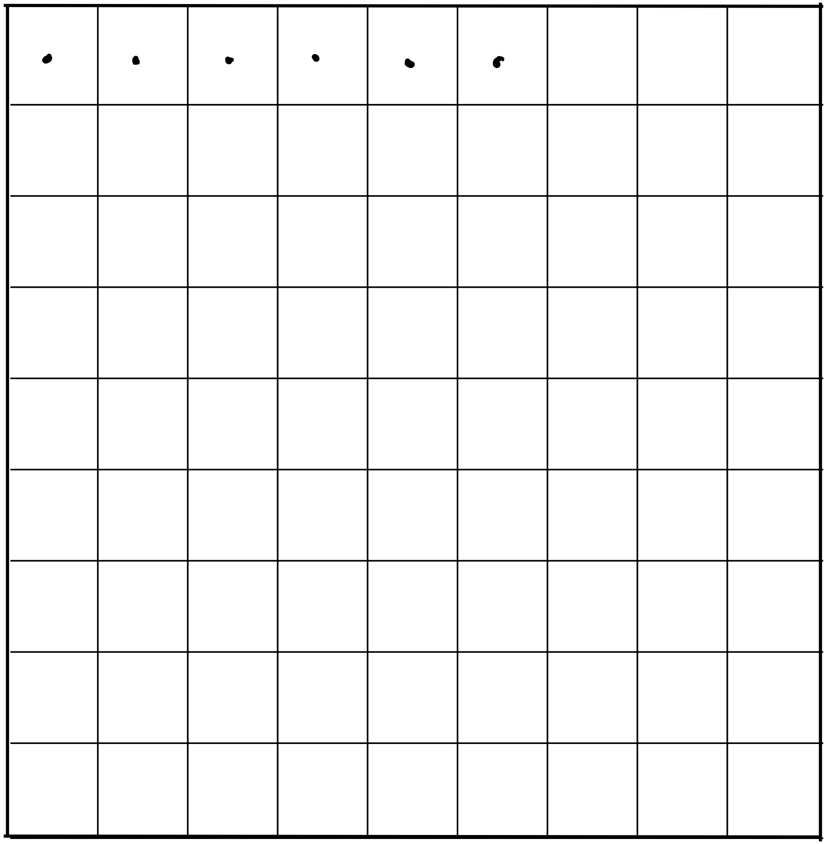
Raster images are *sampled*

$\mathcal{I}: \underset{\uparrow}{\mathbb{R}^2} \rightarrow ?$

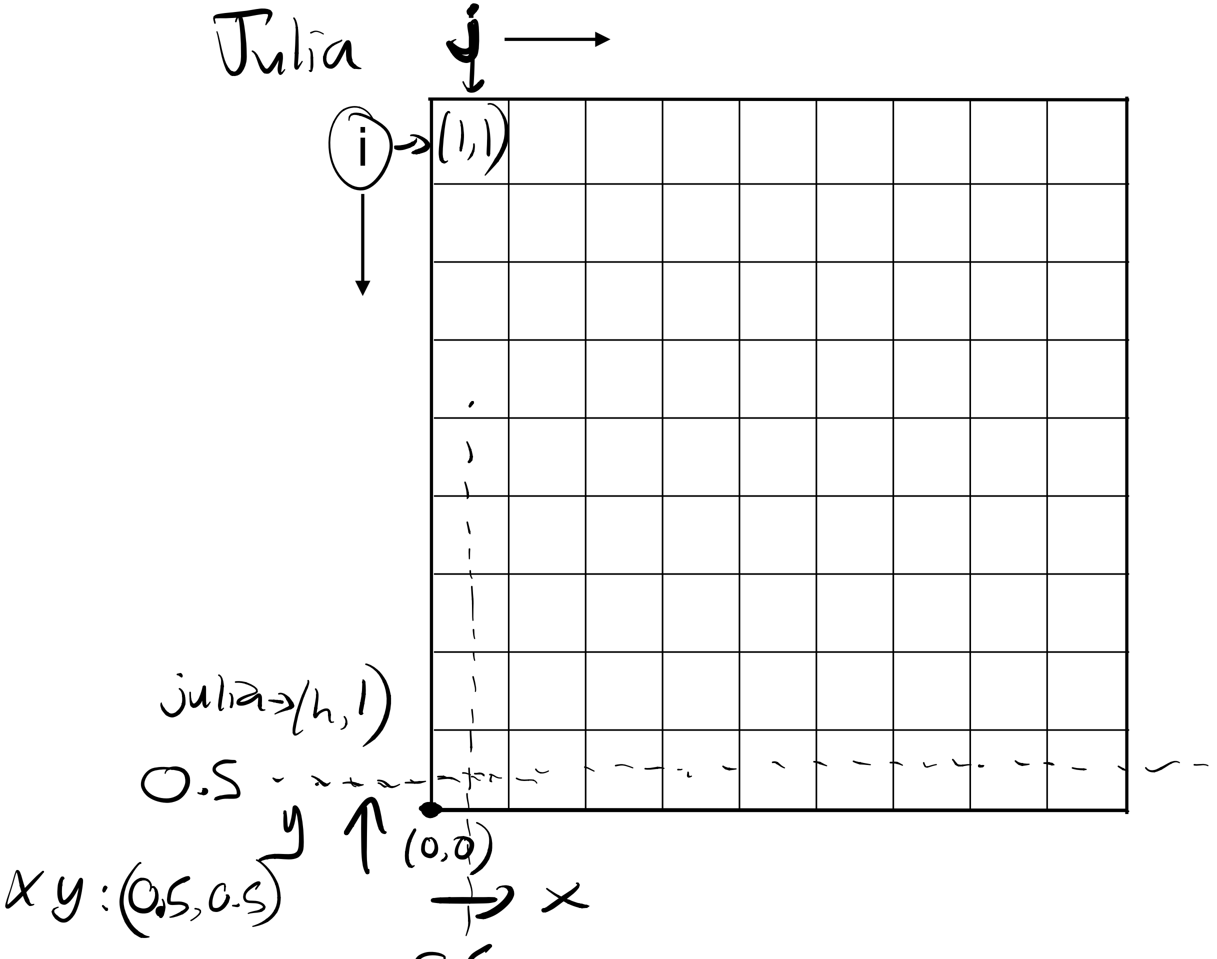


Representing Raster Images

What do pixels *mean*?

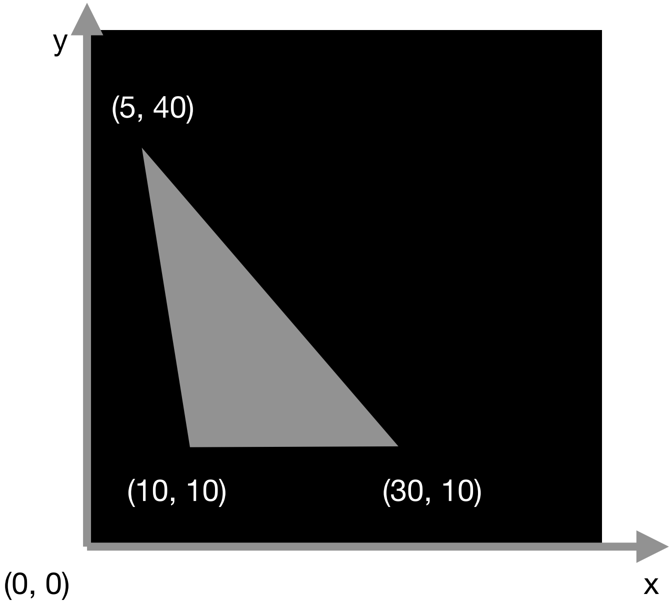


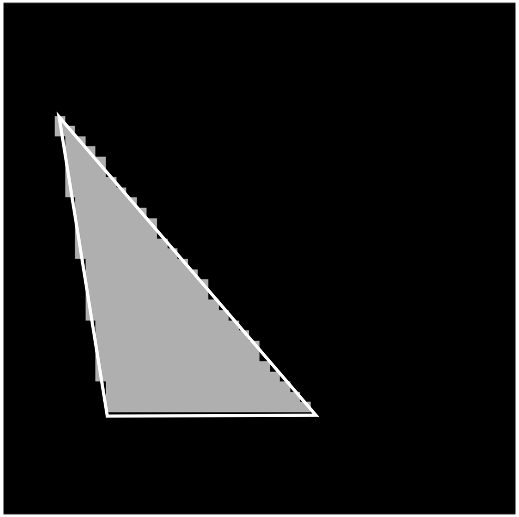
Raster Images: Coordinate Systems



0.5

A0: Rendering (*Rasterizing*) a Triangle

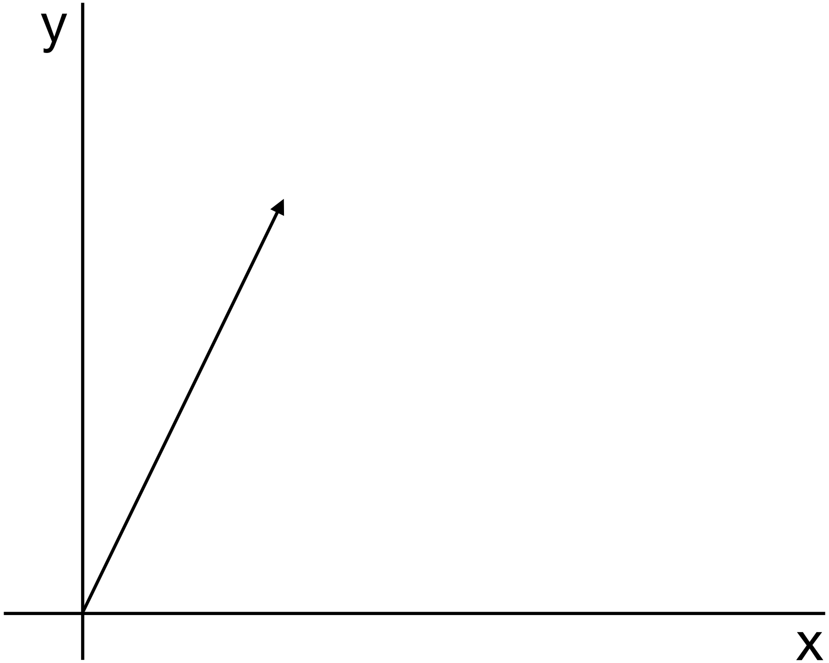




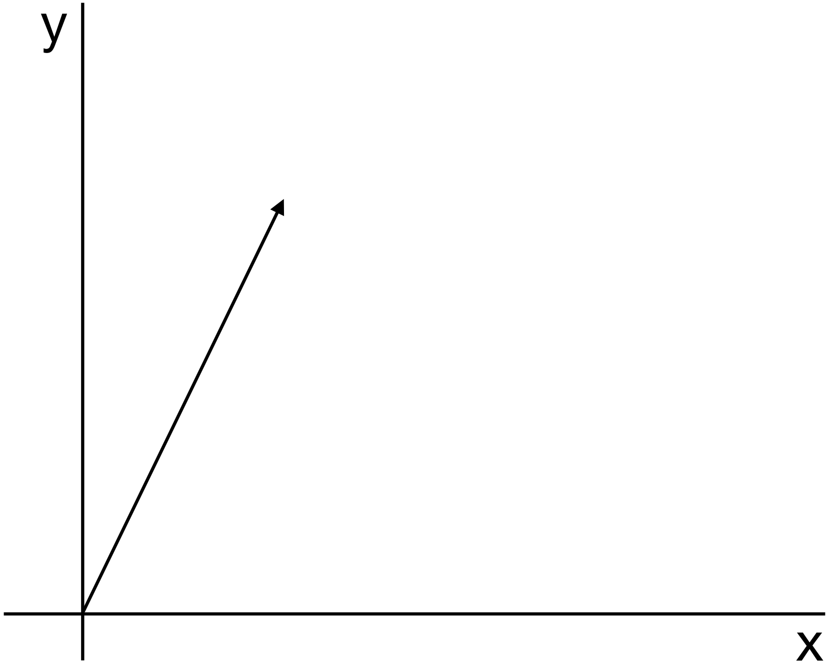
Pseudocode:

```
for each pixel
    if pixel is inside tri.
        color pixel
```

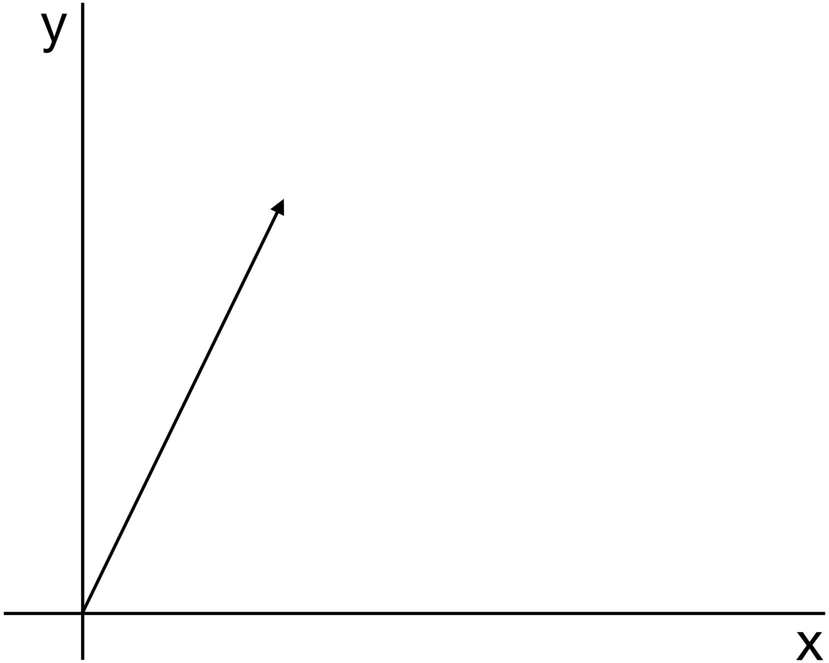
Vectors



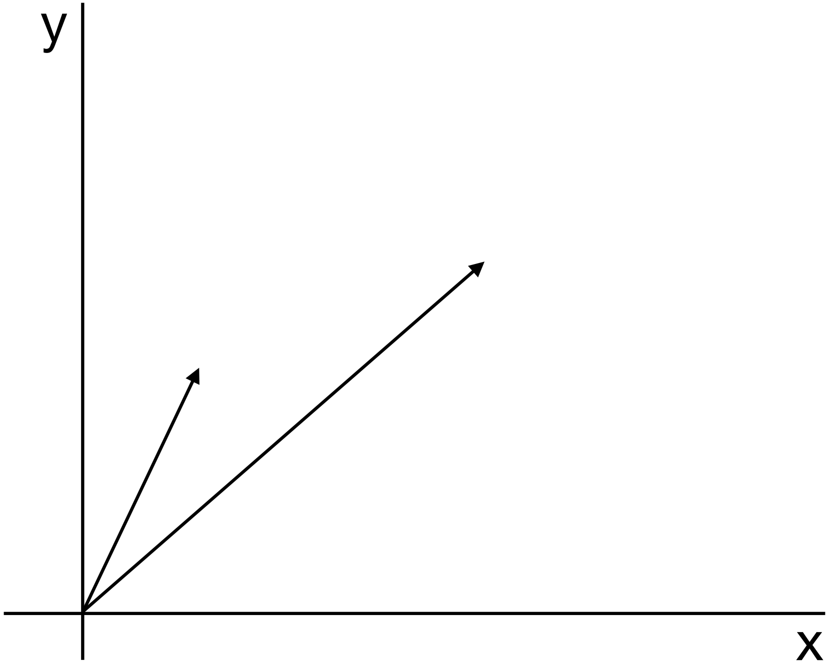
The Canonical Basis



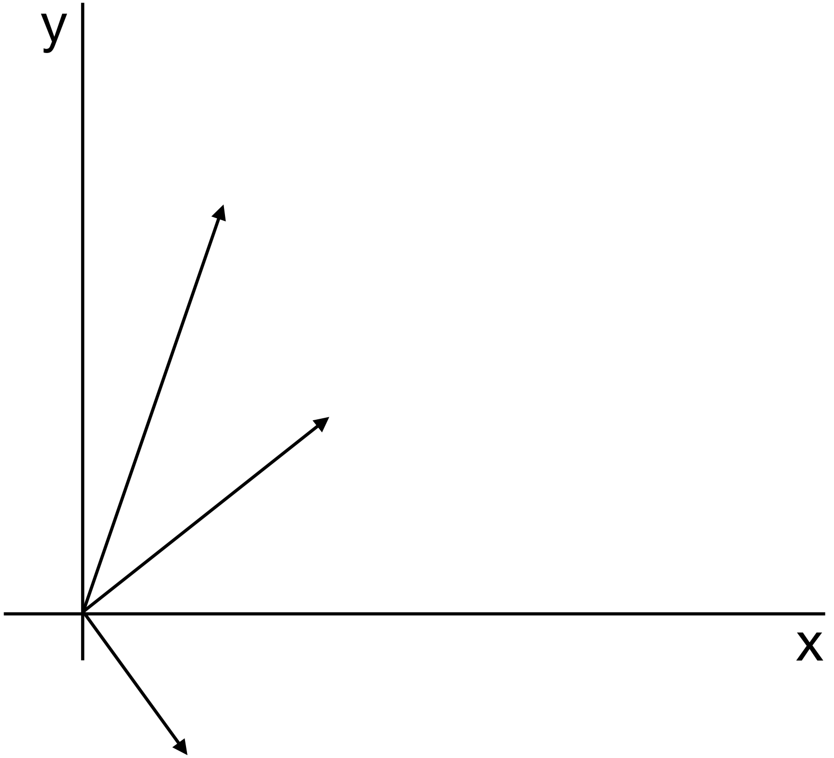
Magnitude (length)



The vector between two points



The dot product



Point-in-Triangle