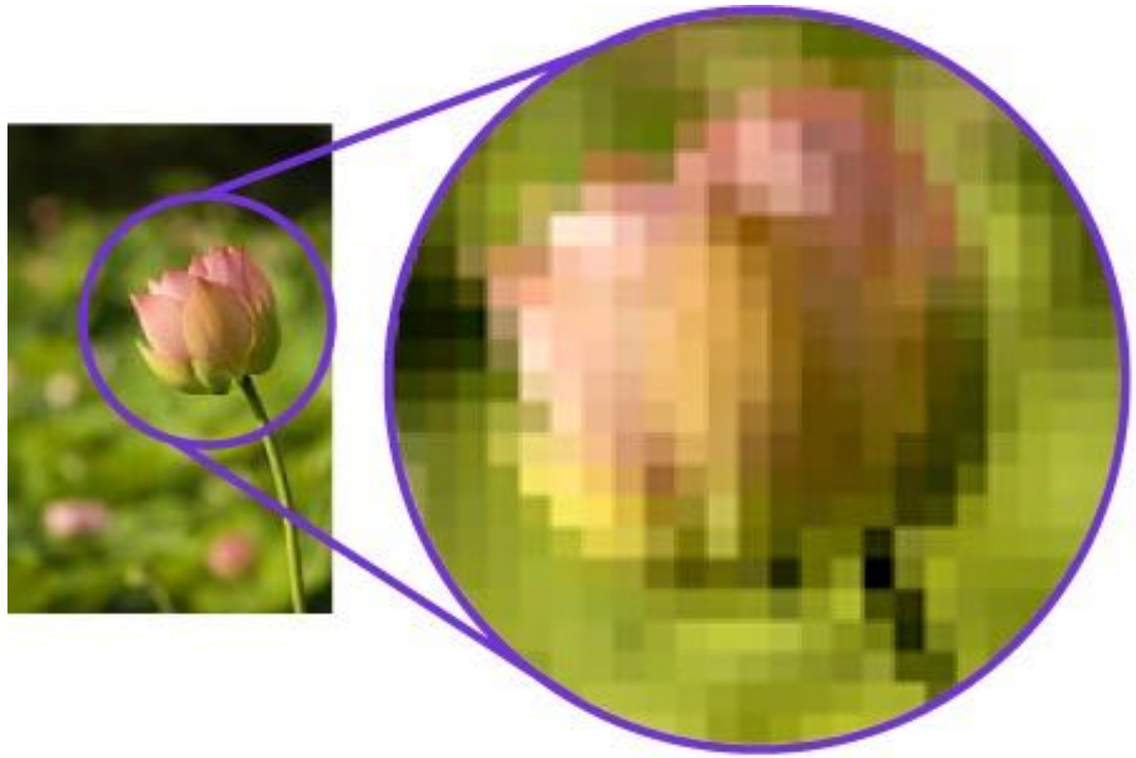
Raster Graphics and Displays

CS 355: Introduction to Graphics and Image Processing

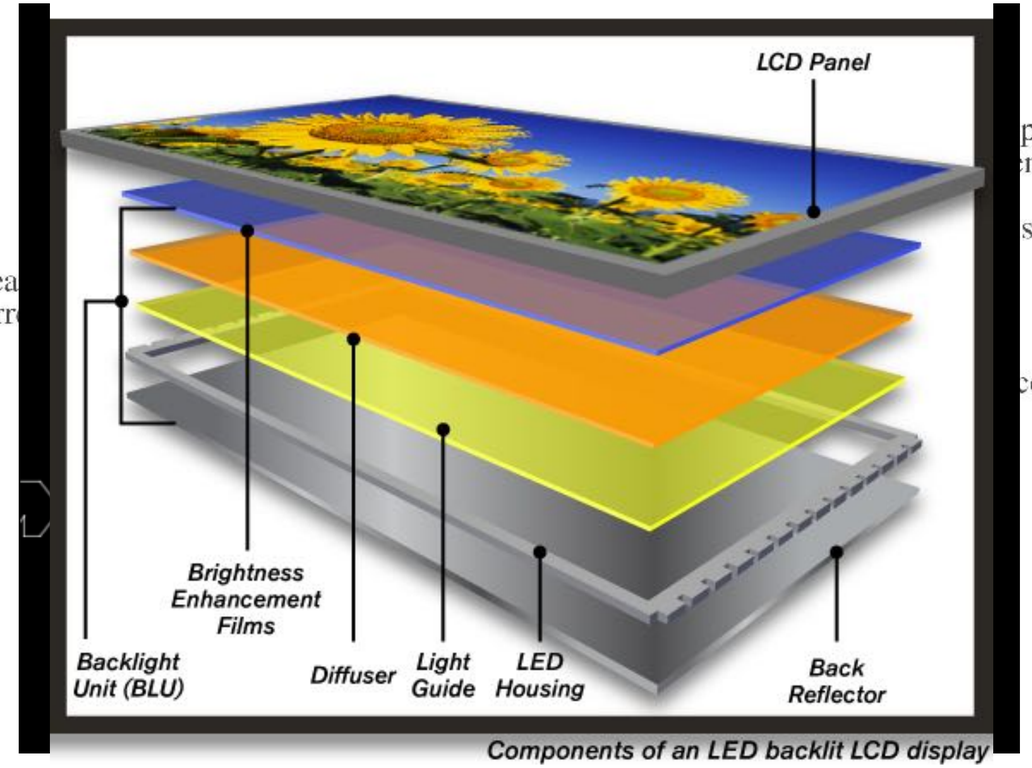
Raster Images



Most digital displays are made up of discrete dots called “pixels” (short for picture elements)

Quick History

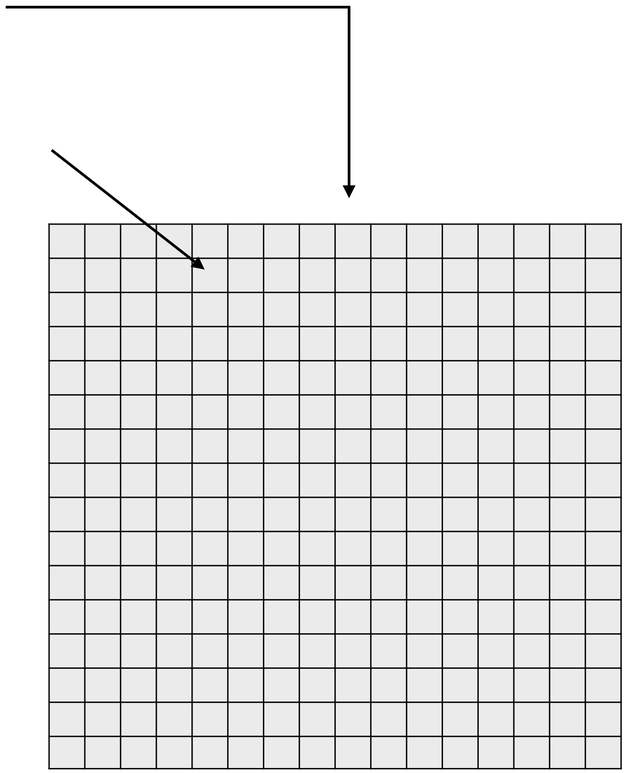
- Vector graphics
- Raster (CRT)
- Raster (Digital)



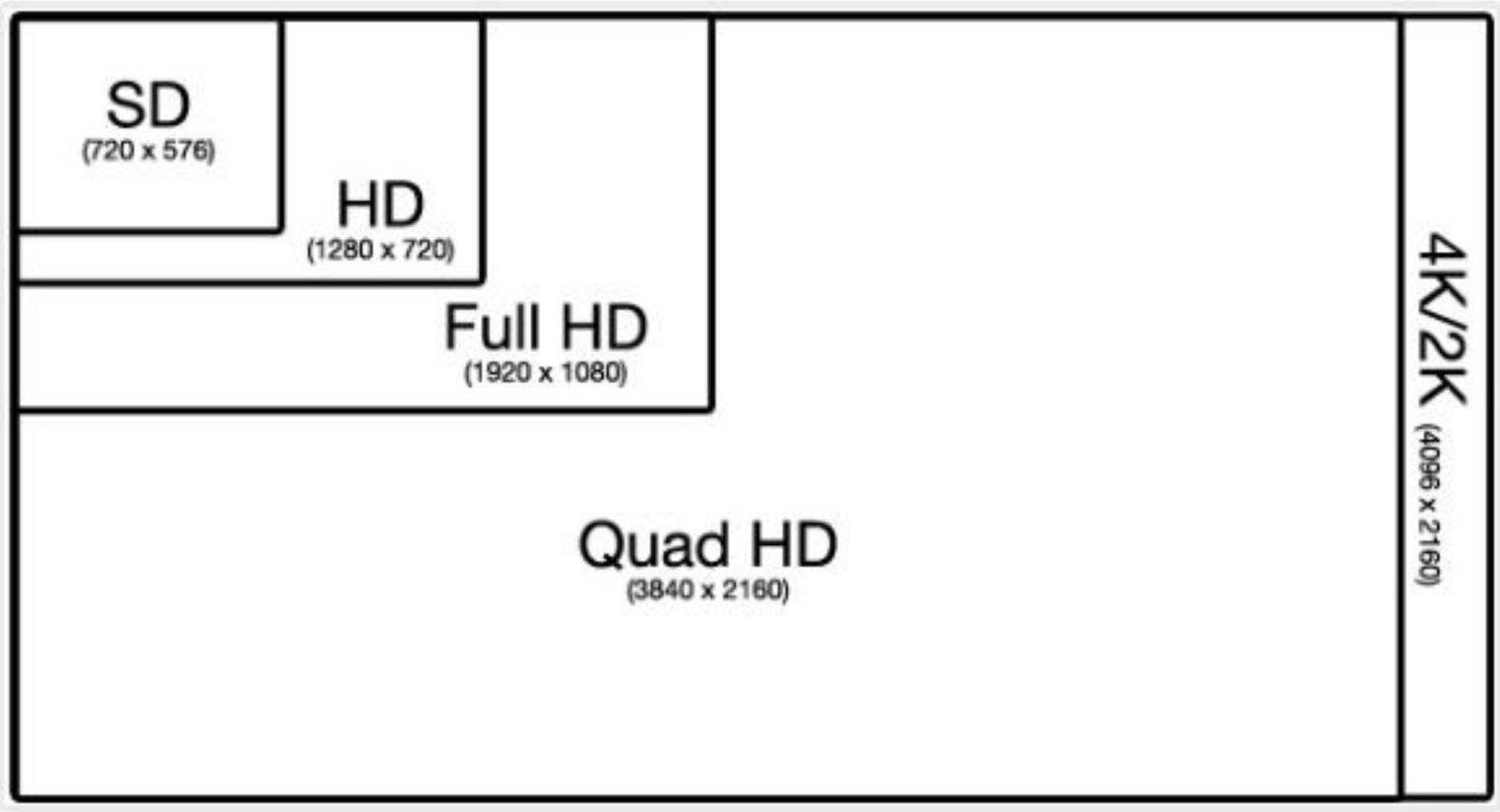
Components of an LED backlit LCD display

Raster Images

- Size: number of pixels (height x width)
- Bit depth: bits of precision for each pixel
  - Binary (true black and white)
  - 2-bit gray (4 shades)
  - 8-bit gray (256 shades)
  - 12-bit gray (X-rays / CT)
  - 24-bit color (8 bits each of R, G, B)



Common Display Sizes

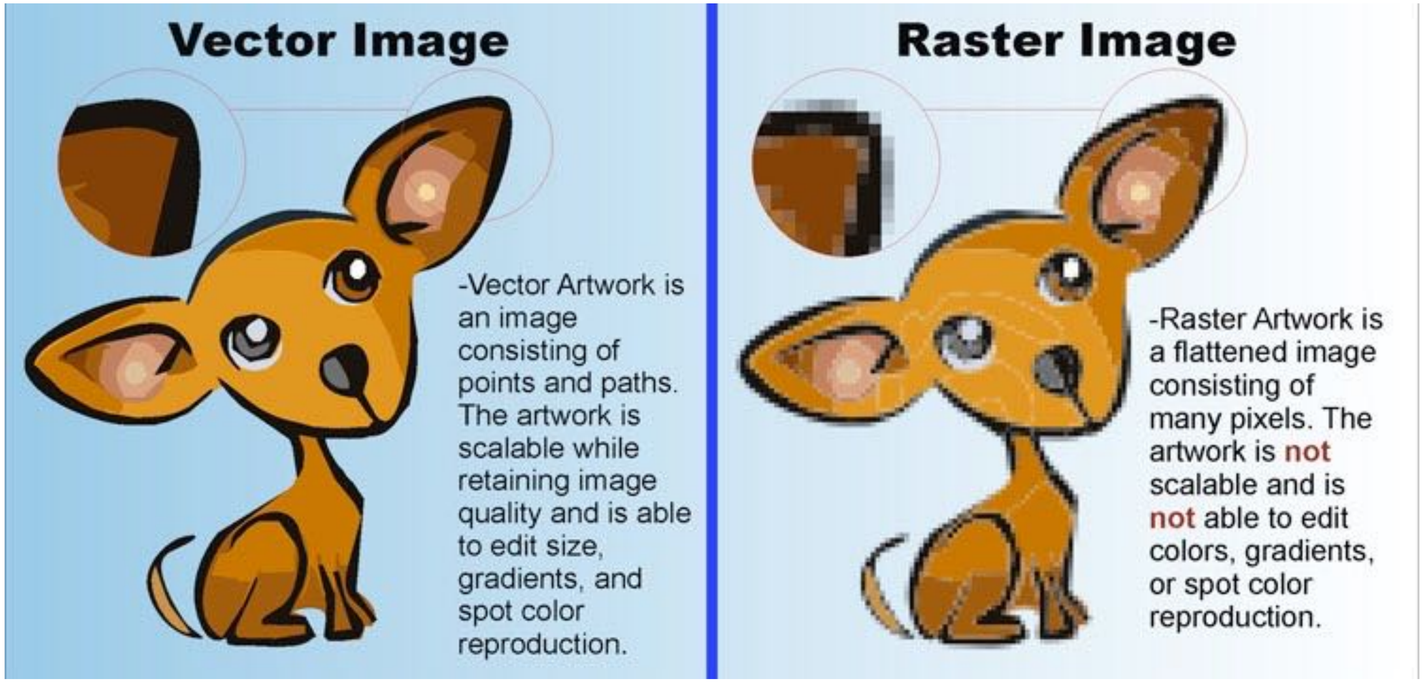


Interlacing

- Progressive
  - Full screen 60 frames / second
- Interlaced
  - Half screen *fields* 60 times / second
    - ➡ Odd lines, then even lines…
  - Equivalent of 30 frames/second

How you display pictures isn’t how you have to store them!

Raster vs. Vector Graphics



Vector:
Continuous Curves

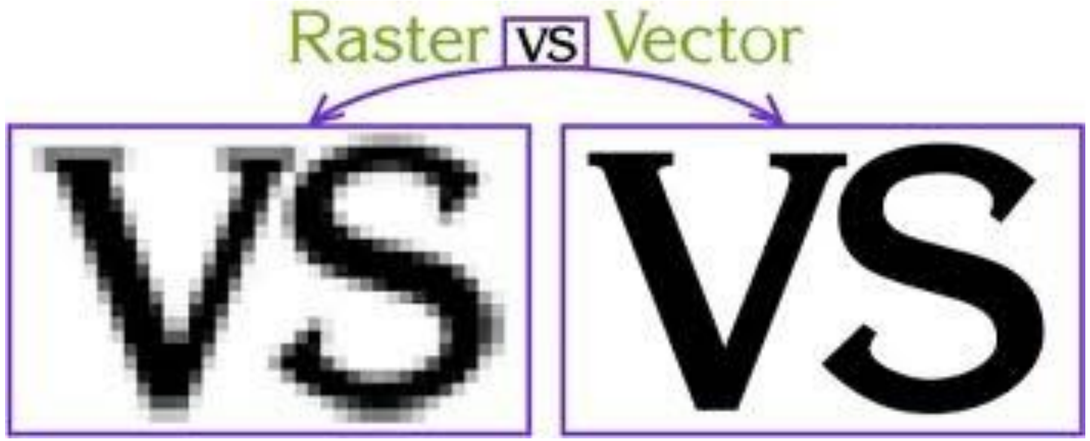
Raster:
Lots of dots

Rasterization

- Vector graphics are higher quality
- But everybody uses raster displays
- Have to convert to a raster image first
  (But get to target resolution to device!)
- This process is called rasterization
  (sometimes scan conversion)

Example: Text

- Most fonts aren’t stored as bitmaps (images)
- Stored instead as curves representing the outline of the characters
- Converted to whatever resolution device supports



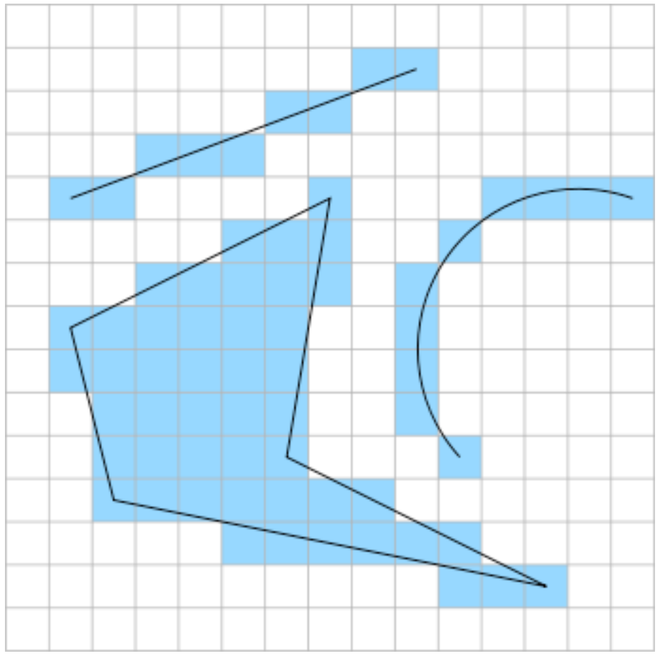
Example: Postscript

- When you send a Postscript file to a printer,
  - You aren't sending a picture of what to print (usually)
  - You are sending a program for how to print it

```
%!PS-Adobe-3.0
%%Title: QR Barcode 2-H, mask=6
%%Creator: JpGraph Barcode http://www.aditus.nu/jpgraph/
%%CreationDate: Mon 6 Jul 16:08:12 2009
%%DocumentPaperSizes: A4
%%EndComments
%%BeginProlog
%%EndProlog
%%Page: 1 1

%Module width: 3 pt

%Data: ABCDEFGH01234567
%Each line represents one row and the x-position for black modules: [xpos]

3.05 setlinewidth
[[12][15][18][21][24][27][30][45][51][66][69][72][75][78][81][84]] {{} forall 87 moveto 0 -3.05 rlineto stroke} forall
[[12][30][45][66][84]] {{} forall 84 moveto 0 -3.05 rlineto stroke} forall
[[12][18][21][24][30][36][39][42][45][51][66][72][75][78][84]] {{} forall 81 moveto 0 -3.05 rlineto stroke} forall
[[12][18][21][24][30][36][39][45][48][51][54][66][72][75][78][84]] {{} forall 78 moveto 0 -3.05 rlineto stroke} forall
[[12][18][21][24][30][45][66][72][75][78][84]] {{} forall 75 moveto 0 -3.05 rlineto stroke} forall
[[12][30][39][45][48][51][54][60][66][84]] {{} forall 72 moveto 0 -3.05 rlineto stroke} forall
[[12][15][18][21][24][27][30][36][42][48][54][60][66][69][72][75][78][81][84]] {{} forall 69 moveto 0 -3.05 rlineto stroke} forall
[[39][42][51][60]] {{} forall 66 moveto 0 -3.05 rlineto stroke} forall
[[21][24][30][33][39][48][51][54][60][75][78]] {{} forall 63 moveto 0 -3.05 rlineto stroke} forall
[[15][18][27][39][42][45][48][51][54][57][69][78][81][84]] {{} forall 60 moveto 0 -3.05 rlineto stroke} forall
[[12][21][27][30][33][36][42][51][54][57][60][66][75][78][84]] {{} forall 57 moveto 0 -3.05 rlineto stroke} forall
[[18][21][27][39][45][51][57][60][63][66][81][84]] {{} forall 54 moveto 0 -3.05 rlineto stroke} forall
[[12][15][18][24][27][30][33][39][51][57][63][69][75][84]] {{} forall 51 moveto 0 -3.05 rlineto stroke} forall
[[12][15][21][48][66][69][81][84]] {{} forall 48 moveto 0 -3.05 rlineto stroke} forall
[[12][15][18][21][24][30][36][51][54][57][72][78][81]] {{} forall 45 moveto 0 -3.05 rlineto stroke} forall
[[12][18][24][27][39][45][51][60][75][78][84]] {{} forall 42 moveto 0 -3.05 rlineto stroke} forall
[[12][24][27][30][33][42][48][57][60][63][66][69][72]] {{} forall 39 moveto 0 -3.05 rlineto stroke} forall
[[36][42][45][60][72][75][78][84]] {{} forall 36 moveto 0 -3.05 rlineto stroke} forall
[[12][15][18][21][24][27][30][36][39][48][51][54][57][60][66][72][81][84]] {{} forall 33 moveto 0 -3.05 rlineto stroke} forall
[[12][30][39][42][45][51][54][57][60][72][84]] {{} forall 30 moveto 0 -3.05 rlineto stroke} forall
[[12][18][21][24][30][36][42][45][51][54][60][63][66][69][72][75][78]] {{} forall 27 moveto 0 -3.05 rlineto stroke} forall
[[12][18][21][24][30][36][39][42][54][72][84]] {{} forall 24 moveto 0 -3.05 rlineto stroke} forall
[[12][18][21][24][30][39][45][51][60][63][66][72][75][81][84]] {{} forall 21 moveto 0 -3.05 rlineto stroke} forall
[[12][30][39][42][45][48][51][60][63][66][69][78][81]] {{} forall 18 moveto 0 -3.05 rlineto stroke} forall
[[12][15][18][21][24][27][30][42][48][54][57][60][63][66][69][72][75][78][81][84]] {{} forall 15 moveto 0 -3.05 rlineto stroke} forall

%End of QR Barcode

showpage

%%Trailer
```

What Could Possibly Go Wrong?

Which are in?



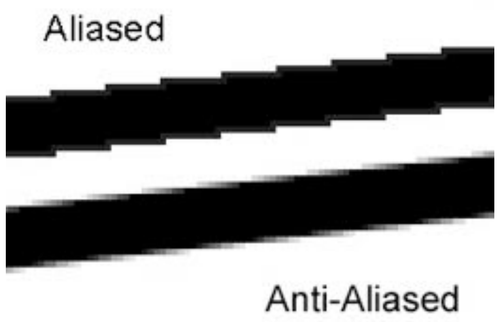
Which are out?

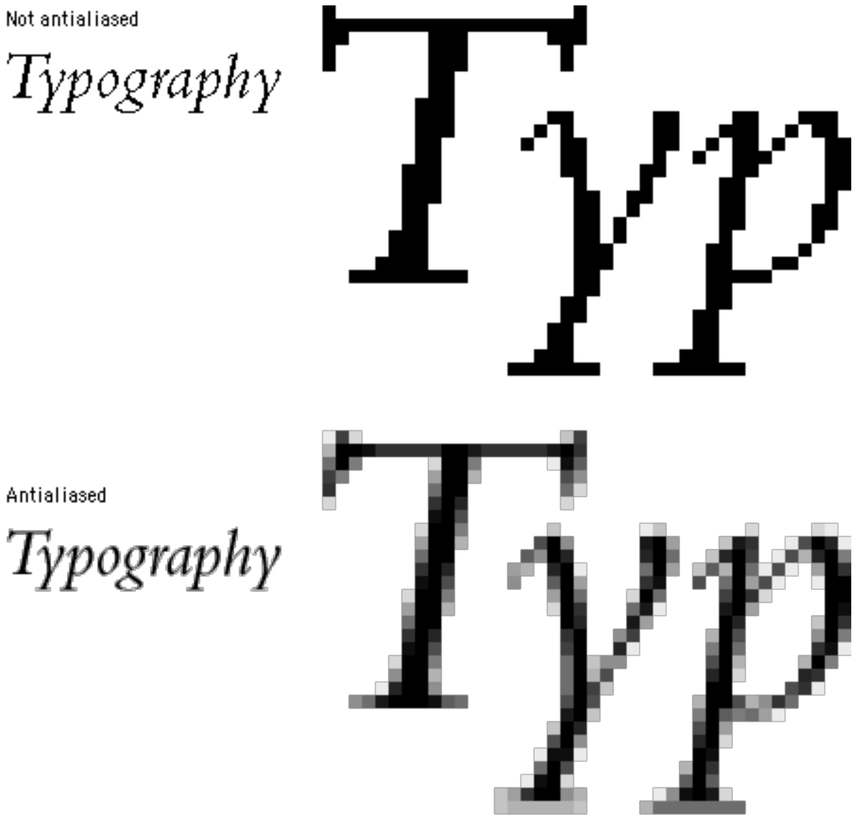
Aliasing

“Jaggies”

Antialiasing

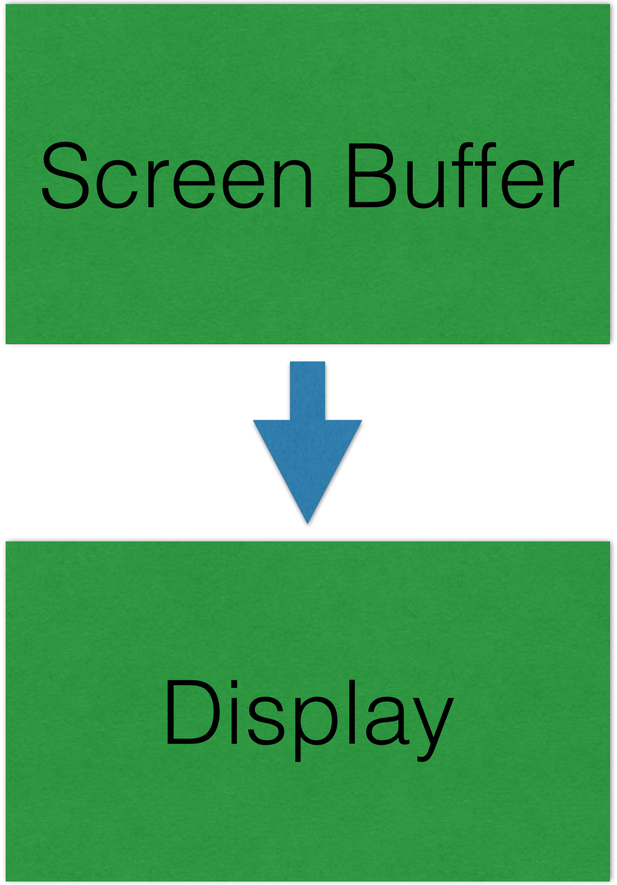
- Can make it look much sharper by blurring a bit (no, really!)
- Key: partially fill in pixels during rasterization





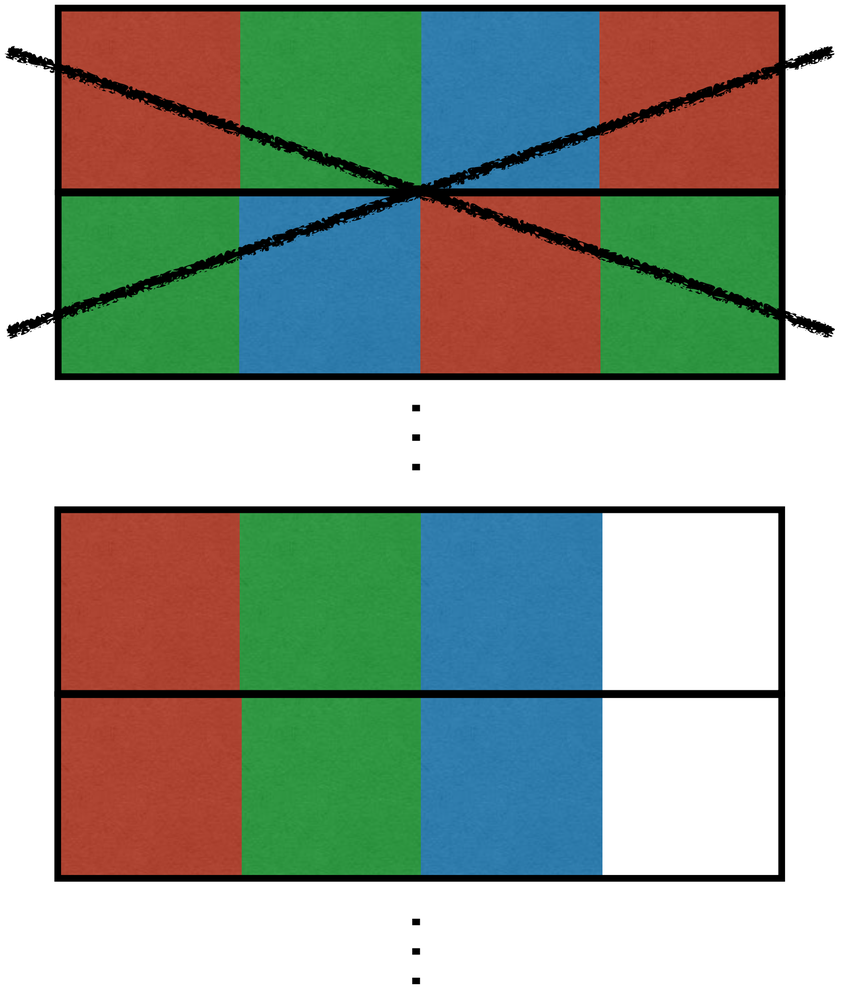
Screen Buffers

- To drive a raster display the computer must store a raster image of what goes on it (usually in graphics card)
- Called a ***screen buffer*** *or* ***video memory***
- Display hardware regularly sends the contents of the screen buffer to the screen
- You draw by writing into the buffer



Note about Color Buffers

- Color images are usually 24 bits (one byte each red, green, blue)
- Memory words are usually 32 bits (or some multiple of this)
- For speed, word-align your pixels (works for other data as well!)
- But, but...



Alpha Channels

- But what about the wasted byte per pixel?
- Usually used for the alpha channel
  - Controls transparency
  - Useful for compositing
  - 0 = transparent, 255 = opaque
- More on this later...

Handling Buffers

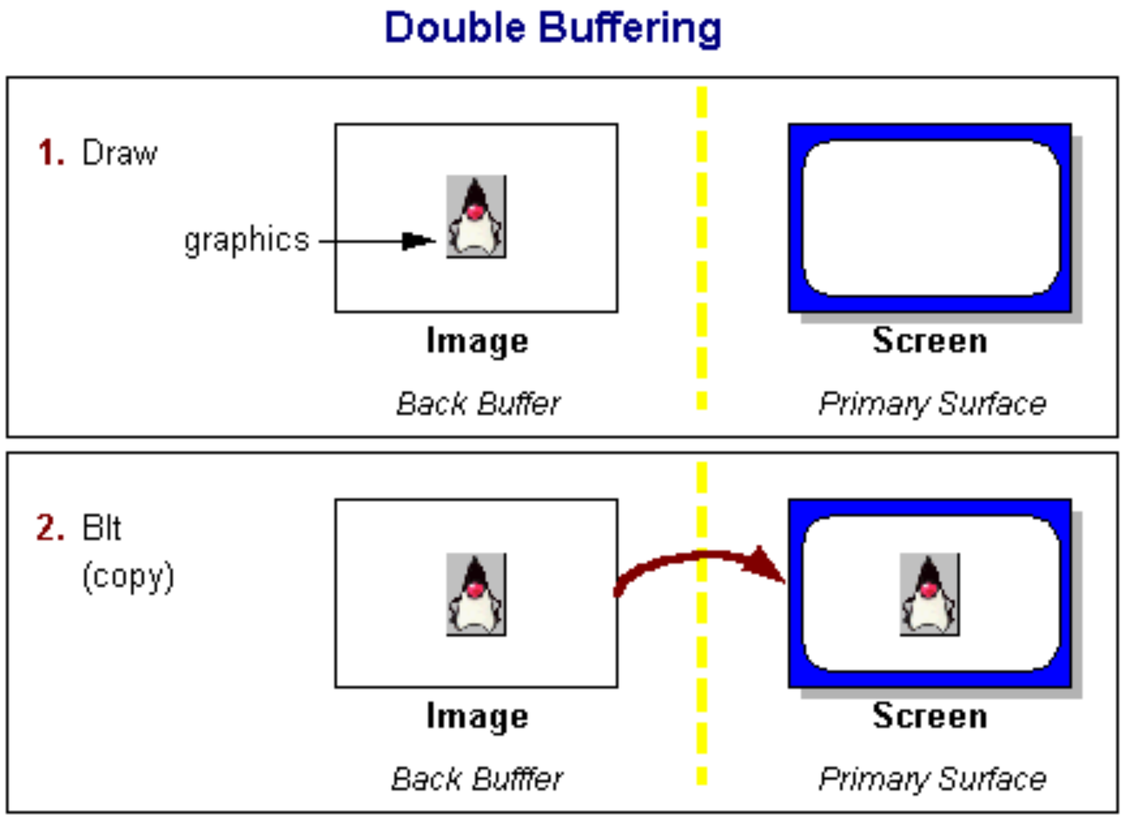
- What if it refreshes from the buffer while you're making changes?
- Two strategies:
  - Single buffer:
    erase / redraw only what's changed
  - Double buffer

Single Buffer

- Erase / redraw only what's changed
  - Use clipping
  - Determining what's in that area is usually not worth it
- Might lead to flicker in these areas, but only these areas
- Used mainly in things that don't update a lot

Double Buffering

- Don’t really draw to the real screen buffer
- Draw to ***offscreen buffer***
- Copy buffers (fast)
  - Some systems support switching with just pointers
- Most common for games, animation, etc.



Layers upon layers...

- In most systems there are lots of layers:
  - Drawing API
  - GUI
  - OS
  - Graphics drivers
  - Graphics cards
  - Display

Coming up...

- Level operations on images
  - Brightness
  - Contrast
  - …