Daniel Rozin, Wooden Mirror (1999)

Introduction to Computer Graphics

Review

- CRTs
  - Vector based
  - Raster based
  - Interlacing

Review

- Vector vs. Raster
  - Another place we see this… web-based graphics
    - Macromedia flash is vector based
    - JPG images are raster based

} So what…

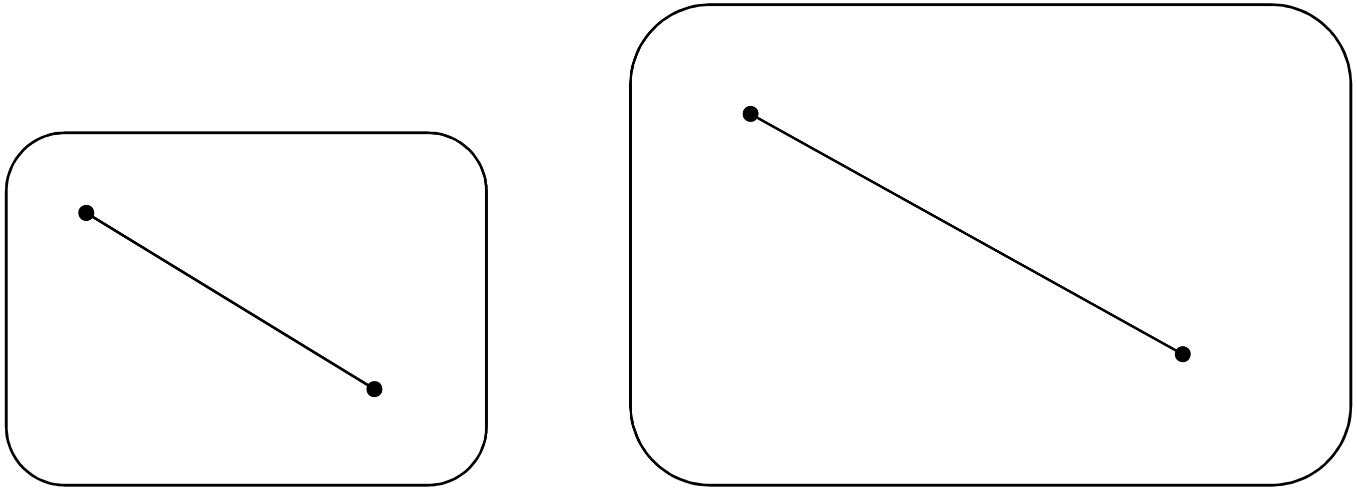
- Time to transmit vs. time to generate
  - *Bandwidth vs. CPU*
- Reuse of image description

Vector Graphics

- How to generate an image using vectors
  - A line is represented by endpoints (10,10) to (90,90)
  - The points along the line are computed using a line equation
    - y = mx + b
  - If you want the image larger, no problem…

Cheap transmission

Computation required

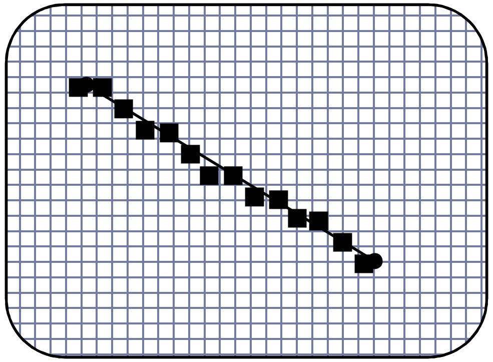


Raster Graphics

- How to generate a line using rasters
  - A line is represented by assigning some pixels a value of 1
  - The entire line is specified by the pixel values
    - What do we do to make image larger?

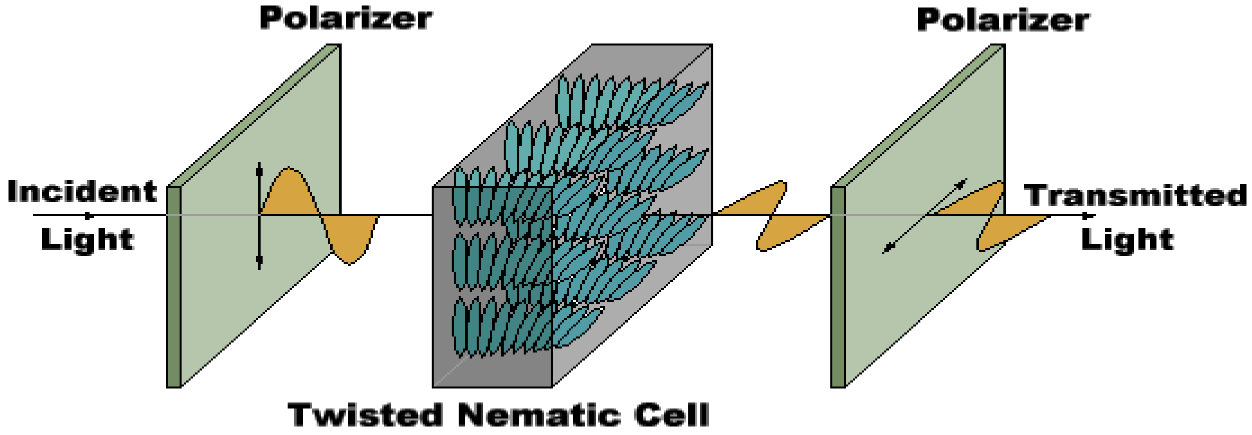
Lot's of extra info to communicate

No computation



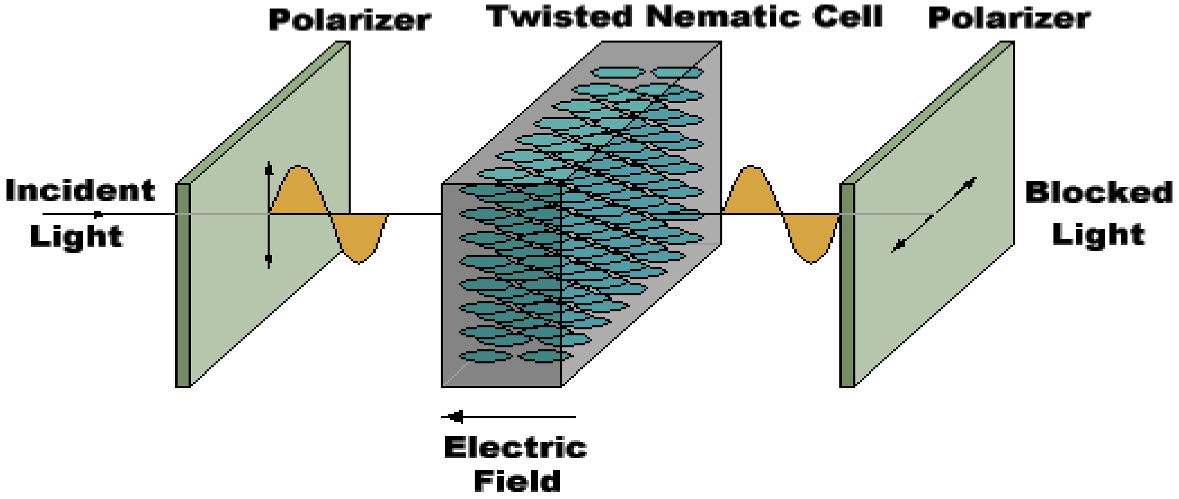
Display Technology: LCDs

- Liquid Crystal Displays (LCDs)
  - LCDs: organic molecules, naturally in crystalline state, that liquefy when excited by heat or E field
  - Crystalline state twists polarized light 90°.



Display Technology: LCDs

- Liquid Crystal Displays (LCDs)
  - LCDs: organic molecules, naturally in crystalline state, that liquefy when excited by heat or E field
  - Crystalline state twists polarized light 90°



Liquid Crystal Display (LCD)

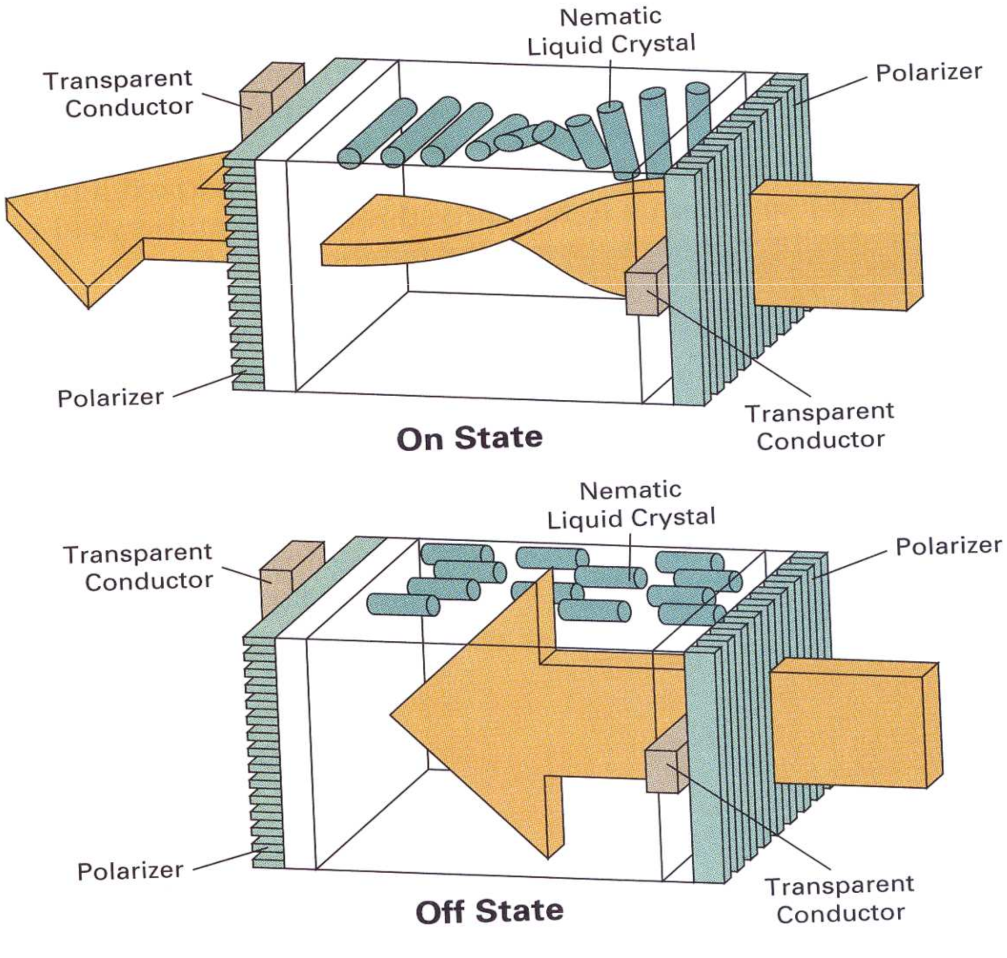
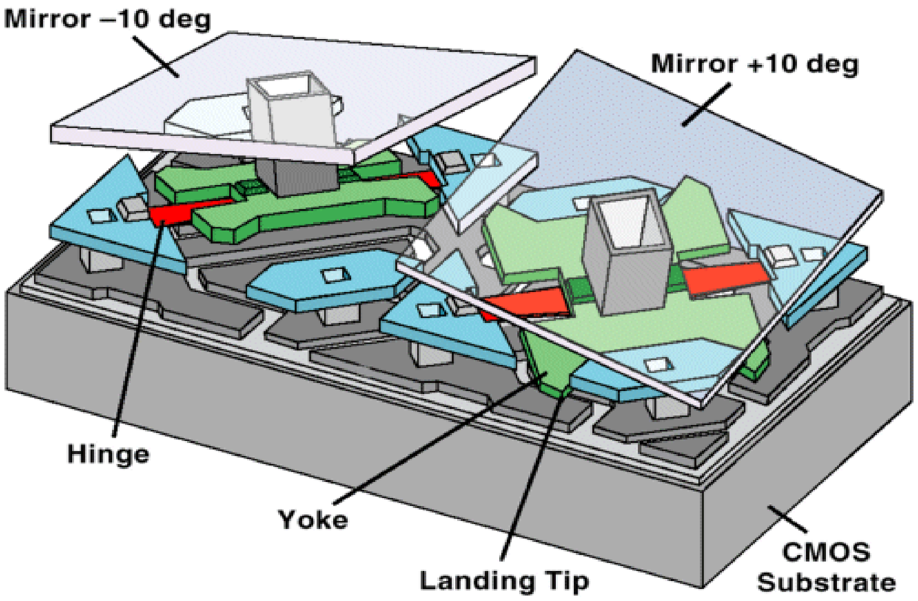
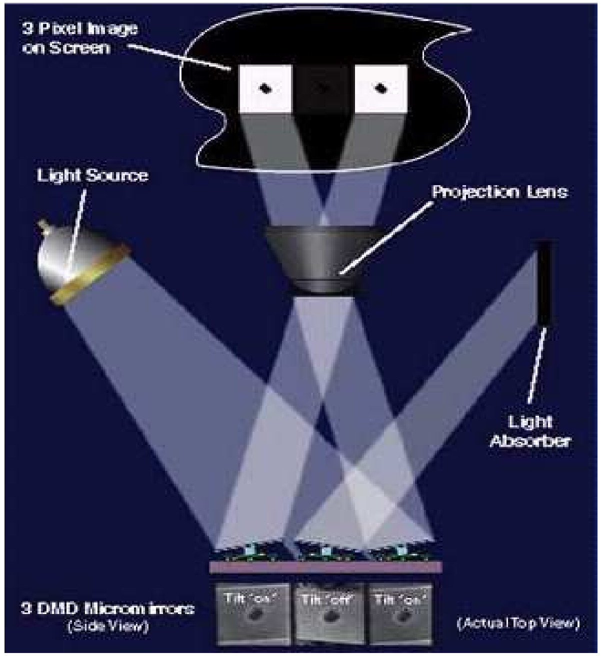


Figure 2.16 from Hearn and Baker

Display Technology: DMD / DLP

- Digital Micromirror Devices (projectors) or Digital Light Processing
    - Microelectromechanical (MEM) devices, fabricated with VLSI techniques



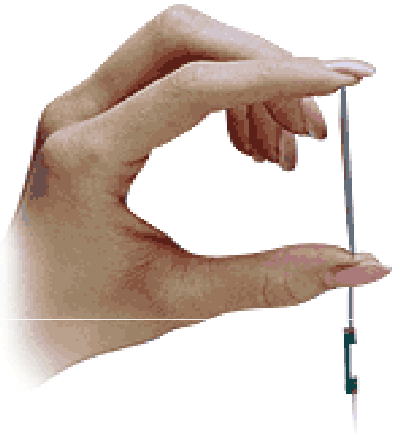


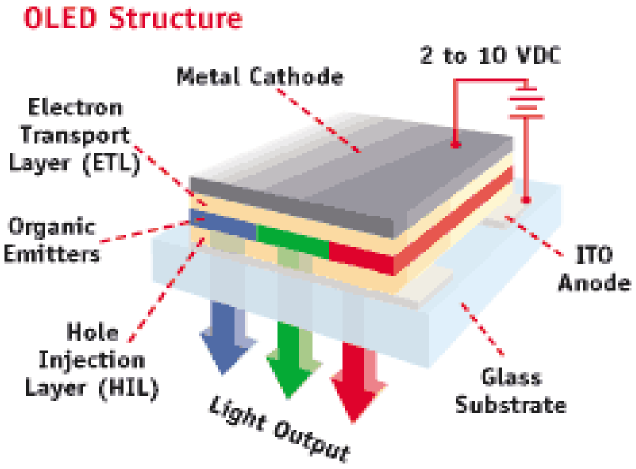
Display Technology: DMD / DLP

- DMDs are truly digital pixels
- Vary grey levels by modulating pulse length
- Color: multiple chips, or color-wheel
- Great resolution
- Very bright
- Flicker problems

Display Technologies: Organic LED Arrays

- Organic Light-Emitting Diode (OLED) Arrays
  - The display of the future? Many think so.
  - OLEDs function like regular semiconductor LEDs
  - But they emit light
    - Thin-film deposition of organic, light-emitting molecules through vapor sublimation in a vacuum.
    - Dope emissive layers with fluorescent molecules to create color.





http://www.kodak.com/global/en/professional/products/specialProducts/OEL/creating.jhtml

Display Technologies: Organic LED Arrays

- OLED pros:
  - Transparent
  - Flexible
  - Light-emitting, and quite bright (daylight visible)
  - Large viewing angle
  - Fast (< 1 microsecond off-on-off)
  - Can be made large or small
  - Available for cell phones and car stereos

Display Technologies: Organic LED Arrays

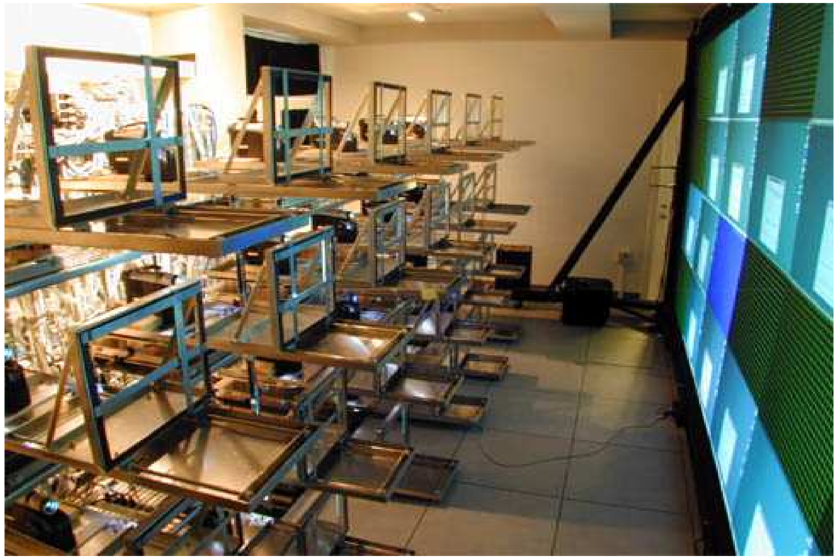
OLED cons:

- Not very robust, display lifetime a key issue
- Currently only passive matrix displays
  - *Passive matrix: Pixels are illuminated in scanline order (like a raster display), but the lack of phospherescence causes flicker*
  - *Active matrix: A polysilicate layer provides thin film transistors at each pixel, allowing direct pixel access and constant illumination*

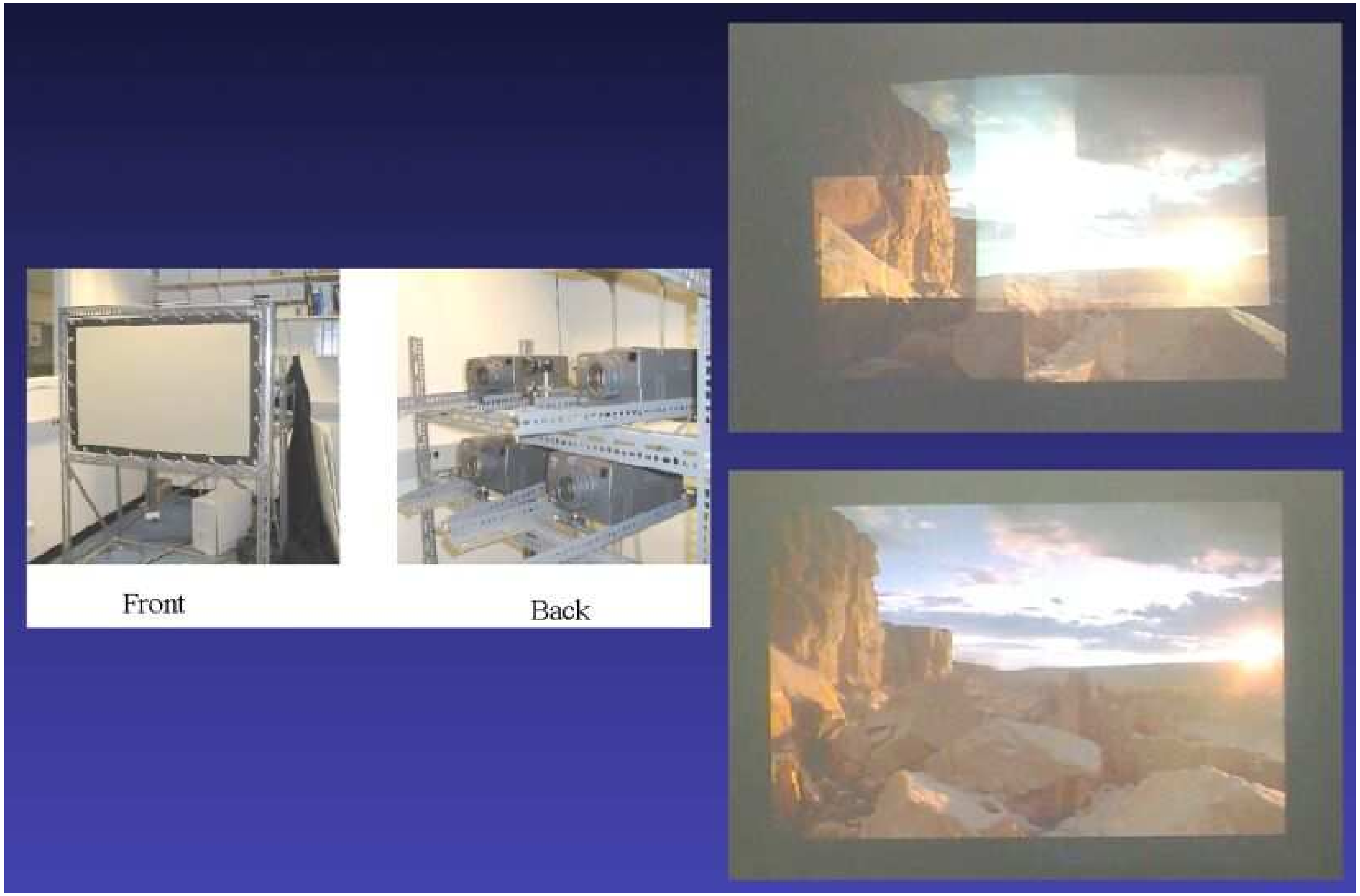
See http://www.howstuffworks.com/lcd4.htm for more info

Additional Displays

- Display Walls
  - Princeton
  - Stanford
  - UVa – Greg Humphreys



Display Wall Alignment



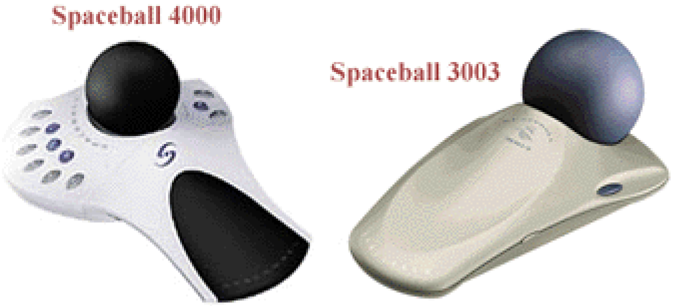
Additional Displays

- Stereo

Interfaces

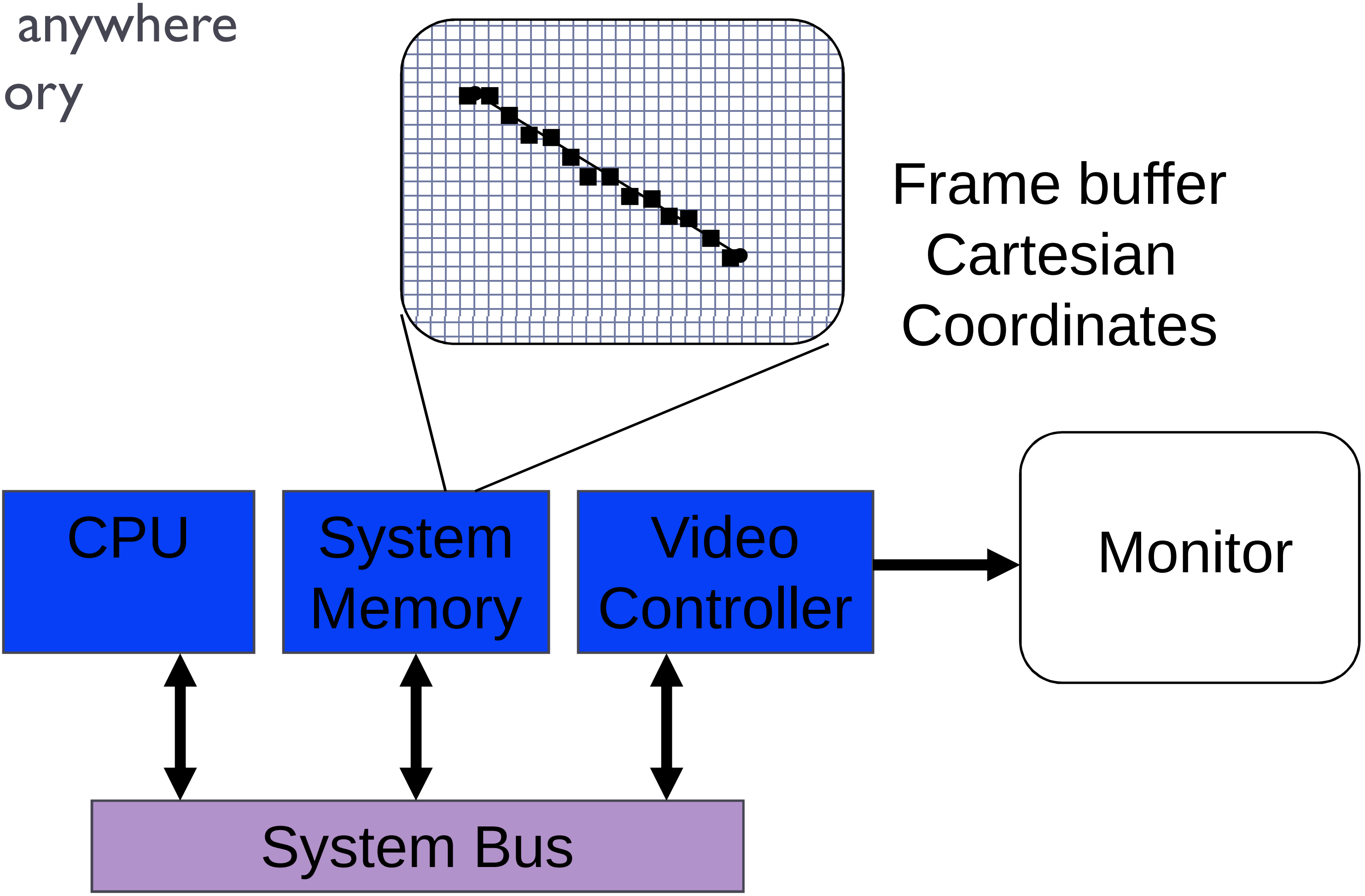
- What is spatial dimensionality of computer screen?
- What is dimensionality of mouse input?

- How many degrees of freedom (DOFs) define the position of your hand in space?
  - Space ball



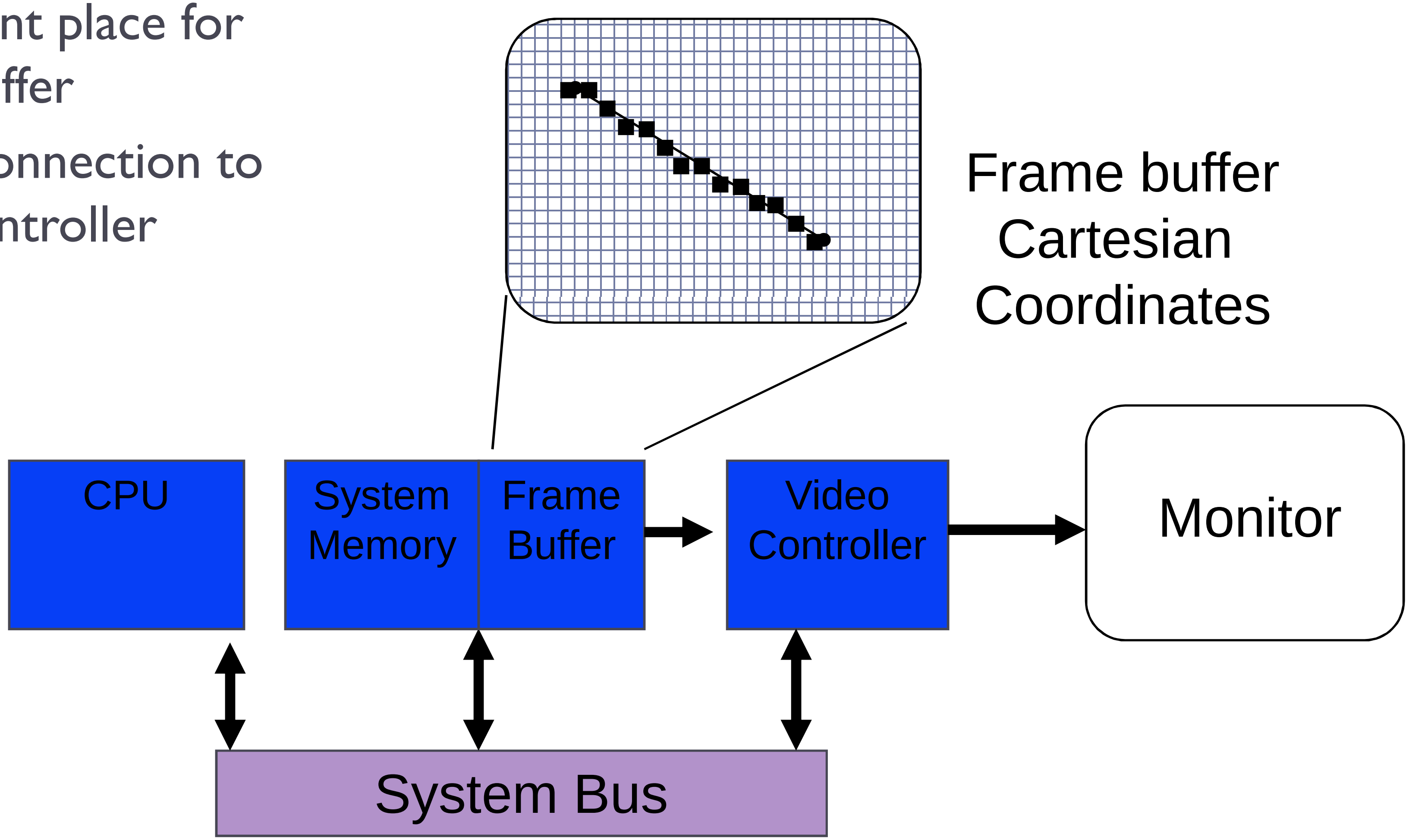
Video Controllers

- Graphics Hardware
  - Frame buffer is anywhere in system memory



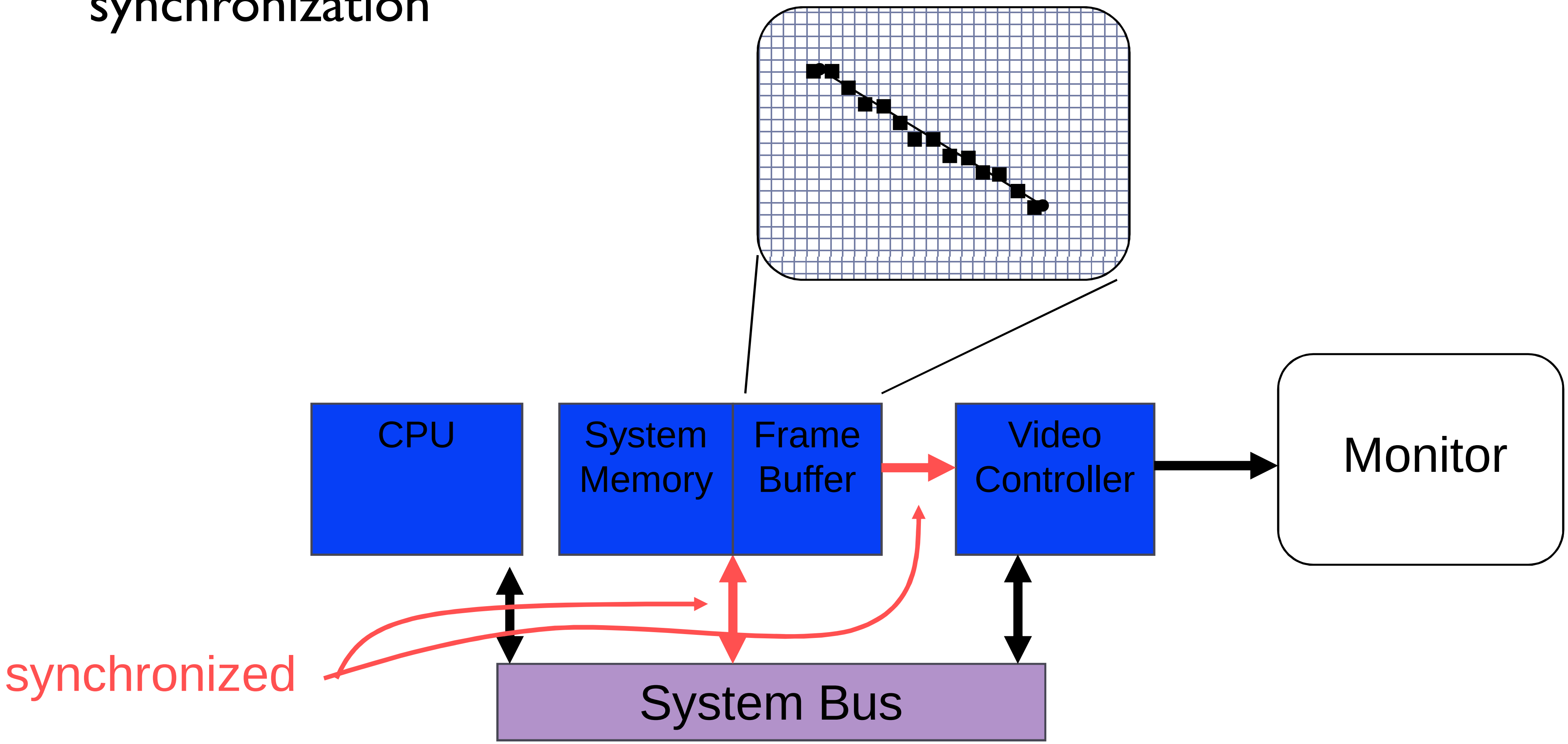
Video Controllers

- Graphics Hardware
  - Permanent place for frame buffer
  - Direct connection to video controller



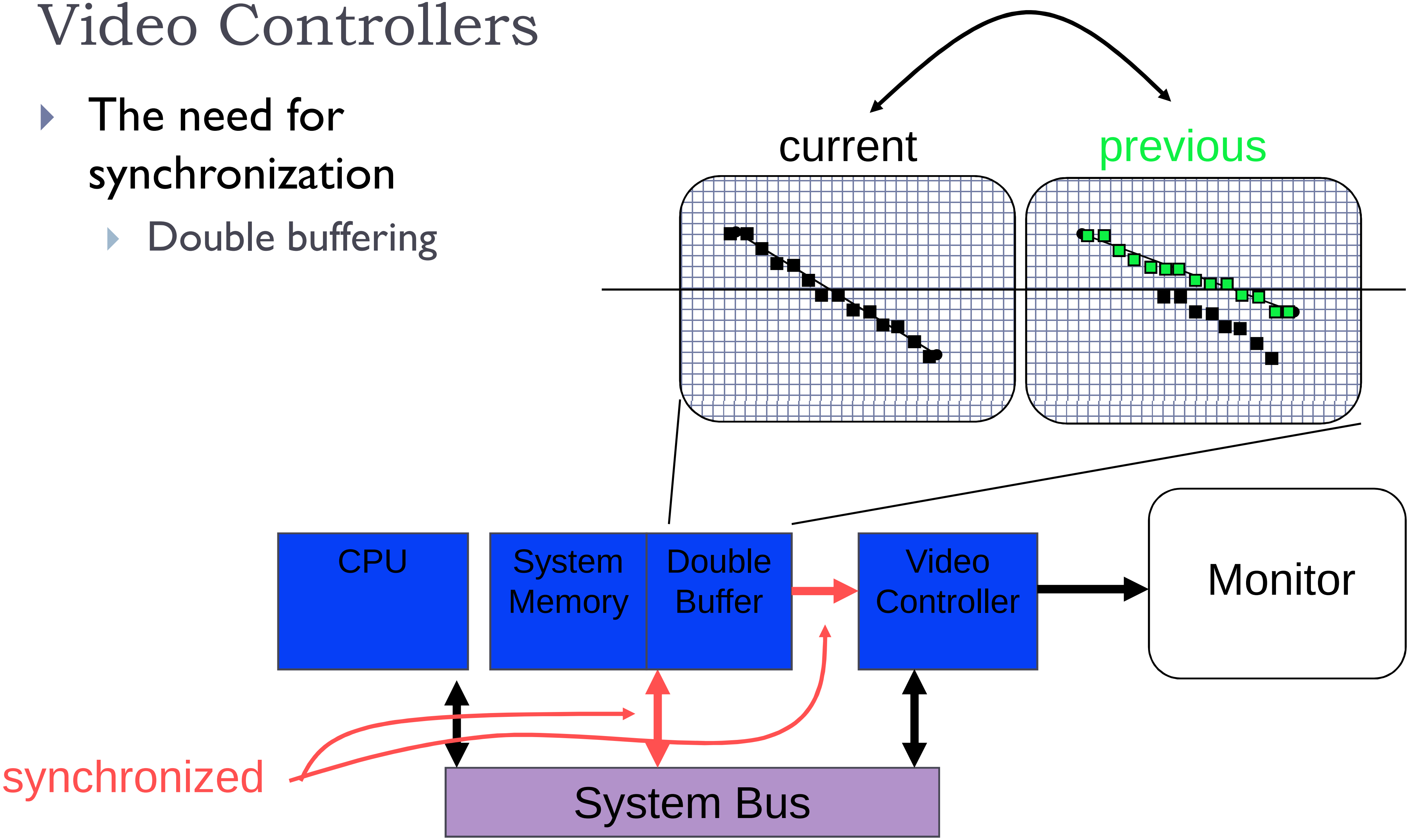
Video Controllers

- The need for synchronization



Video Controllers

- The need for synchronization
  - Double buffering



Raster Graphics Systems

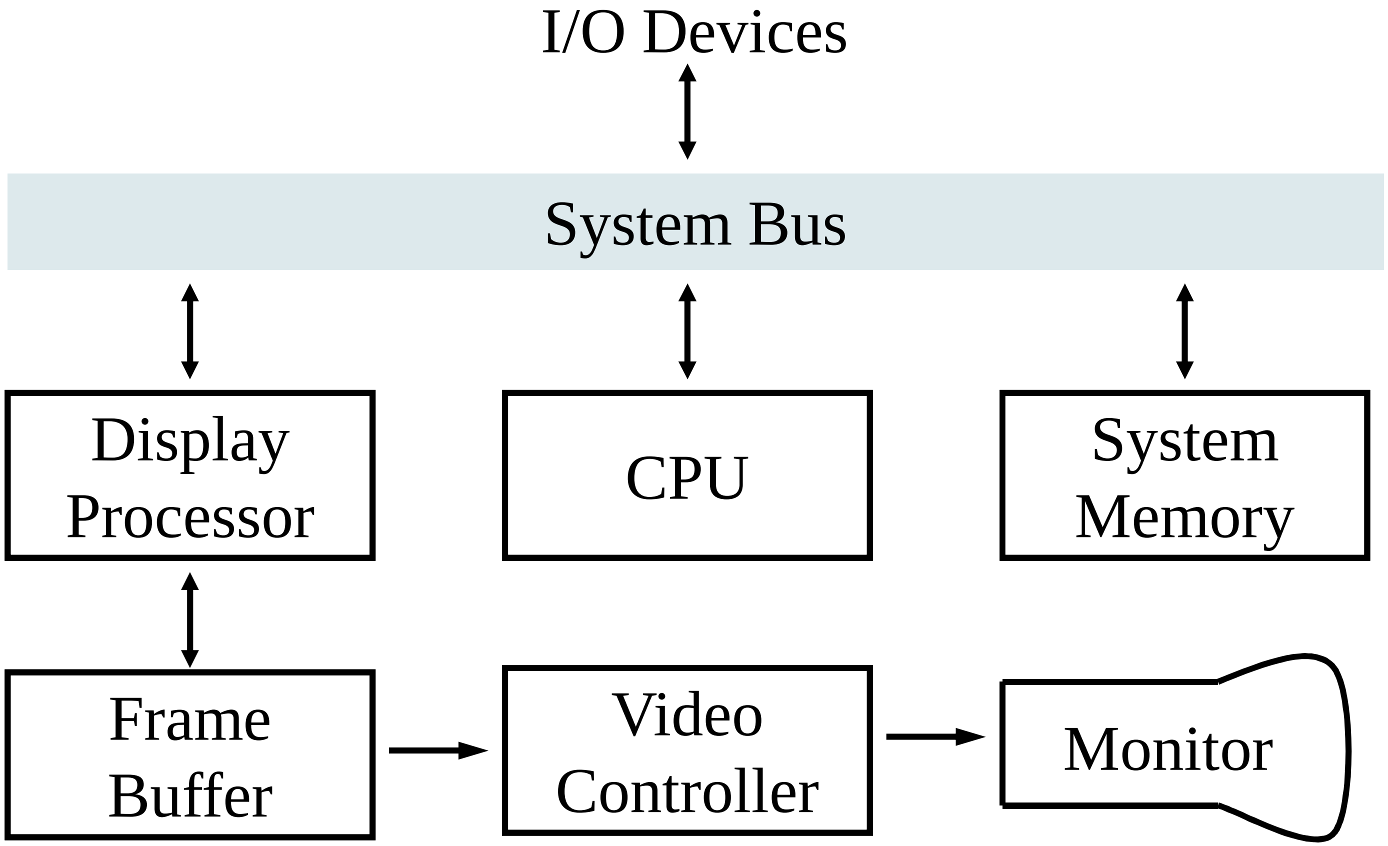
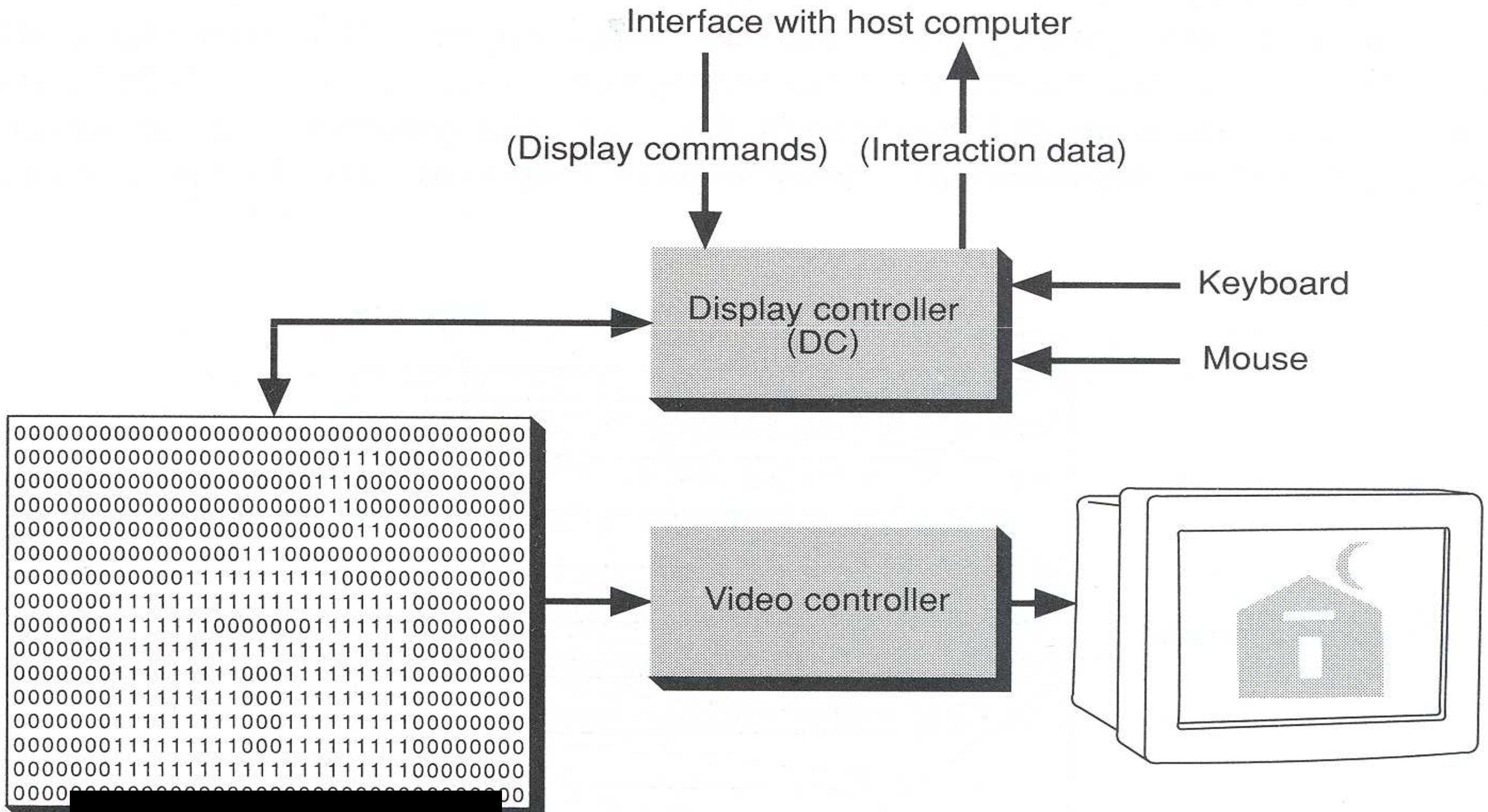


Figure 2.29 from Hearn and Baker

Frame Buffer

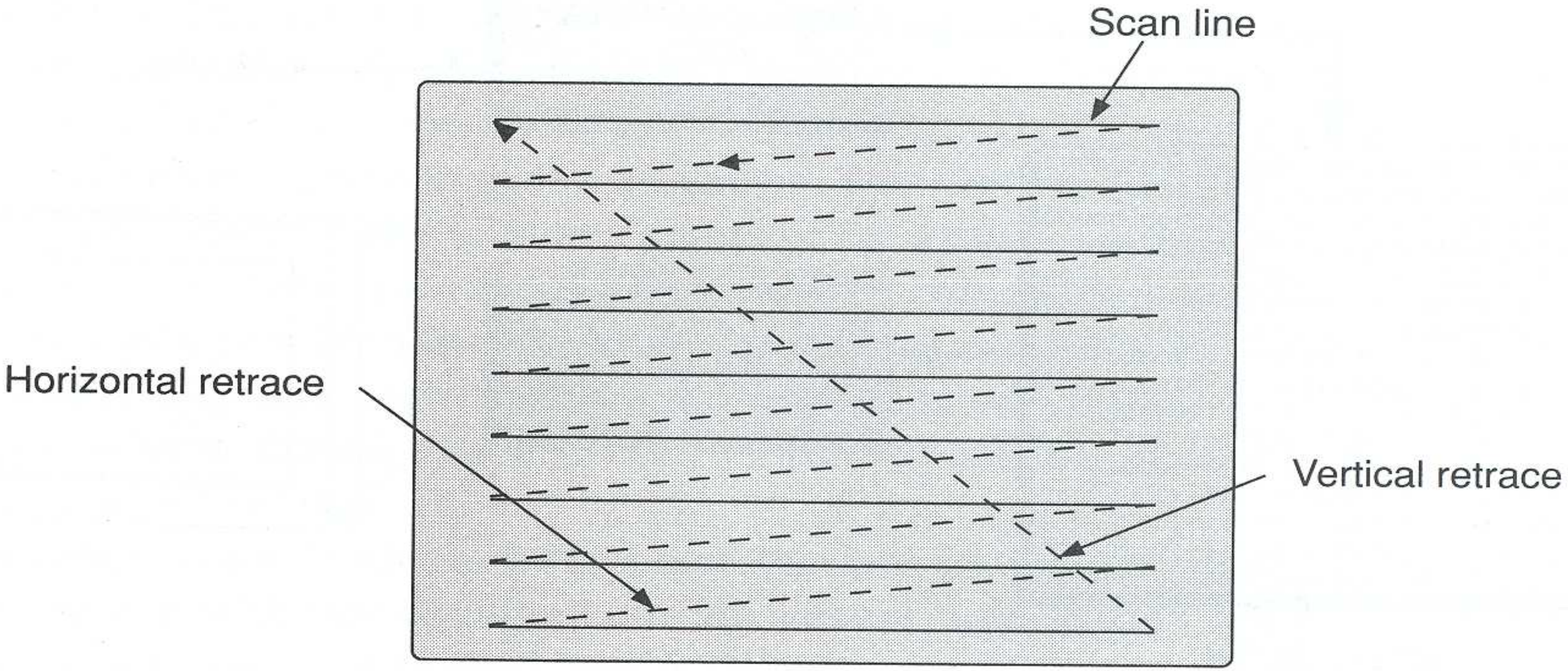


```
000000000000000000000000000000000000
000000000000000000000001110000000000
000000000000000000000111000000000000
000000000000000000000011000000000000
000000000000000000000000110000000000
000000000000000011100000000000000000
000000000000111111111110000000000000
000000011111111111111111111100000000
000000011111110000000111111100000000
000000011111111111111111111100000000
000000011111111100011111111100000000
000000011111111100011111111100000000
000000011111111100011111111100000000
000000011111111100011111111100000000
000000011111111111111111111100000000
000000000000000000000000000000000000
```

Frame Buffer

Figure 1.2 from Foley et al.

Frame Buffer Refresh

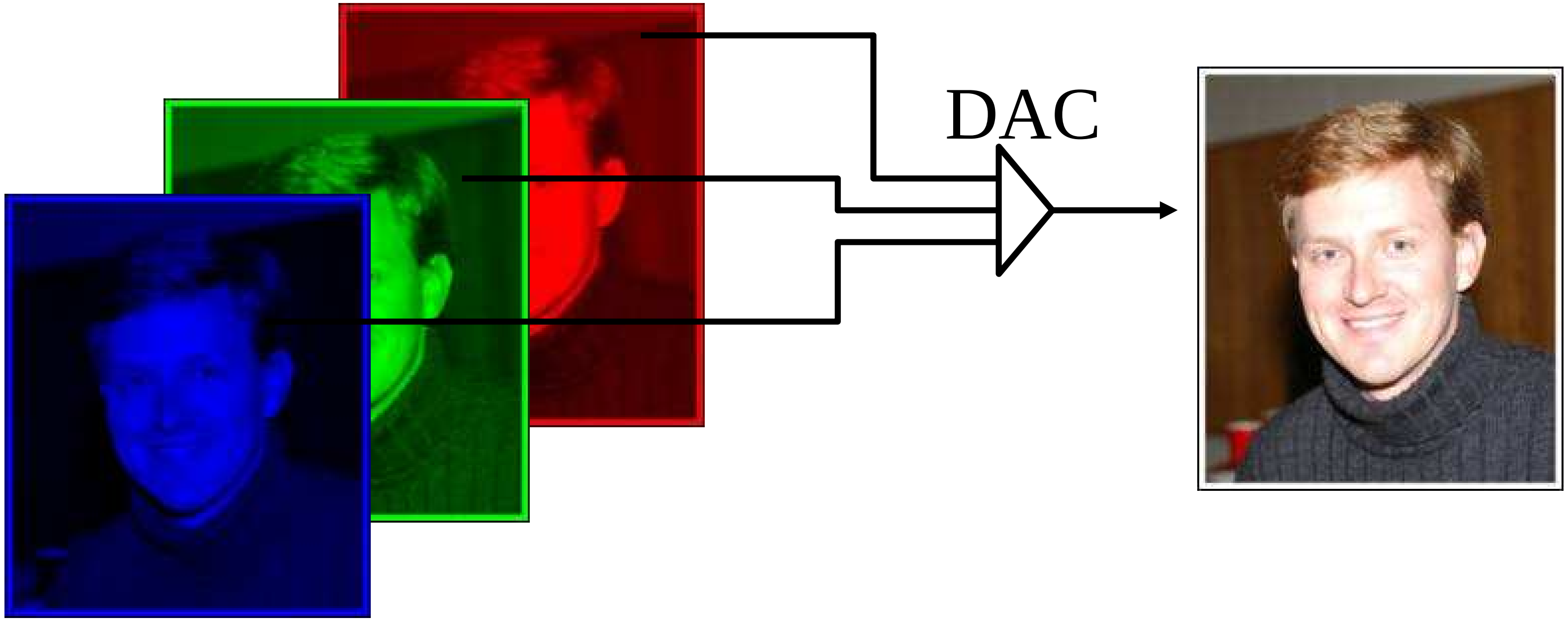


Refresh rate is usually 30-75Hz

Figure 1.3 from FvDFH

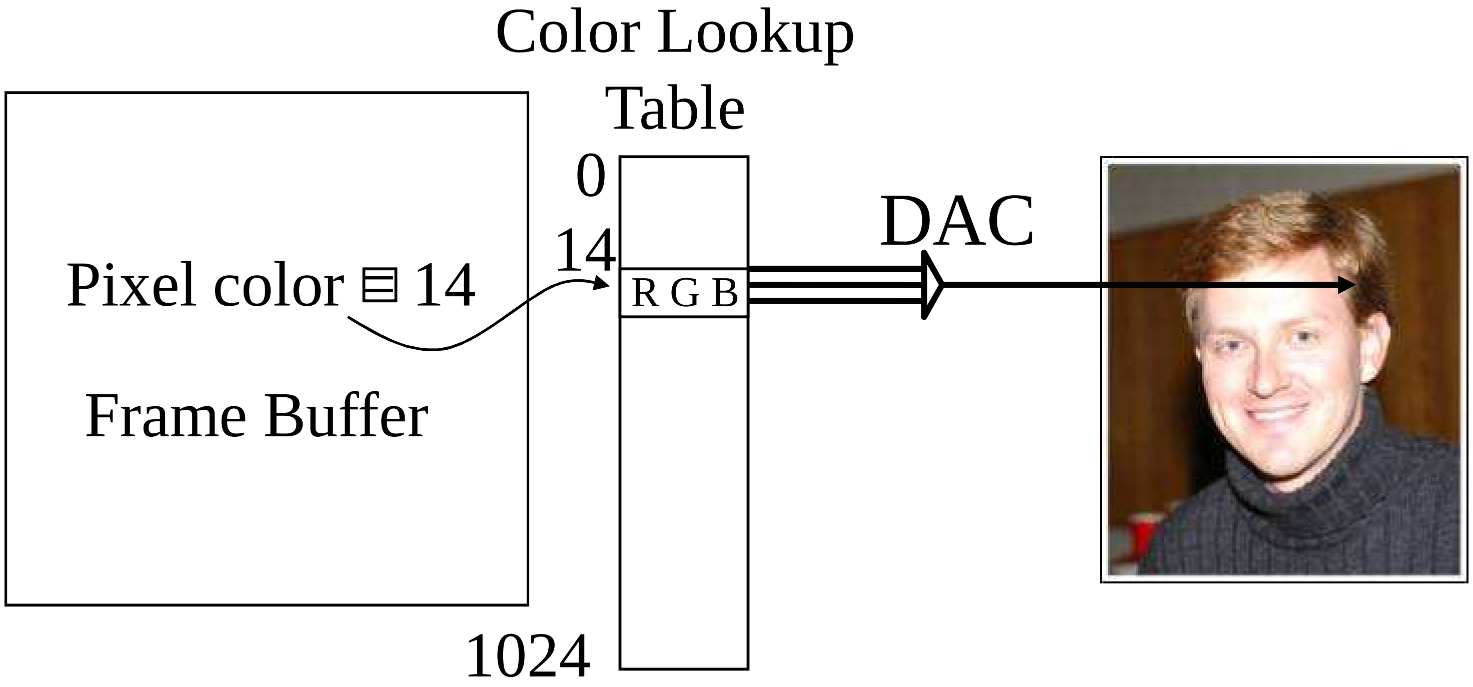
Direct Color Framebuffer

- Store the actual intensities of R, G, and B individually in the framebuffer
- 24 bits per pixel = 8 bits red, 8 bits green, 8 bits blue
  - 16 bits per pixel = ? bits red, ? bits green, ? bits blue



Color Lookup Framebuffer

- Store indices (usually 8 bits) in framebuffer
- Display controller looks up the R,G,B values before triggering the electron guns



Color Lookup Table

0

14

1024

R G B

DAC

Pixel color ⊟ 14

Frame Buffer

Software

- Hide the details
  - User should not need to worry about how graphics are displayed on monitor
  - User doesn't need to know about how a line is converted into pixels and drawn on screen (hardware dependent)
  - User doesn't need to rebuild the basic tools of a 3D scene
    - Virtual camera, light sources, polygon drawing
- OpenGL does this for you…

Software

- Hide the details
  - User doesn’t need to know how to read the data coming from the mouse
  - User doesn’t need to know how to read the keystrokes
- OpenGL Utility Toolkit (GLUT) does this for you…

Software

- Hide the details
  - User doesn’t have to build a graphical user interface (GUI)
    - Pull-down menus, scrollbars, file loaders
- Fast Light Toolkit (FLTK) does this for you…

Software

- Hide the details
  - User shouldn’t have to write code to create a GUI
    - Positioning text boxes, buttons, scrollbars
    - Use a graphical tool to arrange visually
    - Assign callback functions to hook into source code
- Fast Light User Interface Designer (FLUID) does this for you…

OpenGL Design Goals

- SGI's design goals for OpenGL:
  - High-performance (hardware-accelerated) graphics API
  - Some hardware independence
  - Natural, terse API with some built-in extensibility
- OpenGL has become a standard (competing with DirectX) because:
  - It doesn't try to do too much
    - Only renders the image, doesn't manage windows, etc.
    - No high-level animation, modeling, sound (!), etc.
  - It does enough
    - Useful rendering effects + high performance
  - Open source and promoted by SGI (& Microsoft, half-heartedly)

The Big Picture

- Who gets control of the main control loop?
  - **FLTK** – the code that waits for user input and processes it
    - Must be responsive to user… do as I say
  - **GLUT** – the code that controls the window and refresh
    - Must be responsive to windowing system and OS
  - **OpenGL** – the code that controls what is drawn
    - Must be responsive to the program that specifies where objects are located. If something moves, I want to see it.

The Big Picture

- Who gets control of the main control loop?
    - Answer: FLTK
        - We'll try to hide the details from you for now
        - But be aware of the conflict that exists
    - FLTK must be aware of GLUT and OpenGL state at all times
        - Must give code compute cycles when needed
    - We'll discuss OpenGL as if it were standalone