# CSC 470
# COMPUTER GRAPHICS

Administrativia
Why OpenGL
Course outline/Syllabus
Grading
Other issues

# Administrativia

- Instructor: Dr. Natacha Gueorguieva, 1N-205
- Office hours: M, W 1:15 pm-2:15 pm
- Books:
- Hearn D., Baker M., Carither W., Computer Graphics with OpenGL, 4/E , Prentice Hall, ISBN 10: 0 -13-605358-0, ISBN- 13 : 978 -0 -13-605358-3, 2011
- Edward Angel, Interactive Computer Graphics: A Top-Down Approach Using OpenGL, 6/E , Pearson, ISBN-10: 013254523-3, ISBN-13 : 978013254523-3
- Edward Angel, Interactive Computer Graphics: A Top-Down Approach Using OpenGL, 5/E , Pearson, ISBN-10: 0321535863, ISBN- 13 : 978 03215 35863
- Graham Sellers, Richard S Wright, Nicholas Haemel, OpenGL SuperBible: Comprehensive Tutorial and Reference, 6/E ed, Addison-Wesley, ISBN-13: 97803219 02948 ISBN-13: 9780321902948 2013

- Workload – as usual a lot. Two projects, six-seven homeworks, 1 midterm, 1 Final, 1 PPT

# What is OpenGL?

OpenGL is a computer graphics *rendering* API

- With it, you can generate high-quality color images by rendering with geometric and image primitives
- It forms the basis of many interactive applications that include 3D graphics
- By using OpenGL, the graphics part of your application can be
  - operating system independent
  - window system independent

# What is OpenGL?

- A software API with many functions that allow communication with graphics hardware.
- Cross-platform
- Commonly used in many graphics applications
  - Games
  - CAD
  - Visualization

# Why OpenGL?

- Cross platform
- Many companies use it (Raytheon, NUWC, Nvidia, etc.)
- Since OpenGL 3.1 a bit of a mess, but will work on it
- Latest version 4.5
- Managed by the Khronos Group
- Used in gaming, visual effects, movies, medicine, space and other simulators and research and many other

# Design and limitations

- OpenGL was designed to produce reasonably looking 3D images quickly and simply.
  - A lot of its design is a rough approximation of how visual phenomena behave in the real world.
  - Meant to only handle rendering, nothing else.
    - So no window management, event-handling, etc.
  - Meant to abstract away graphics hardware into a standard, clean interface.

# OpenGL is a state-based API

- Most functions manipulate global state.
- 3 types of functions:
  - Those that modify global state.
  - Those that query global state.
  - Those that cause something to be rendered.
    - e.g., `glEnd` or `glDrawElements`

# The new OpenGL

- They enforce a new way to program with OpenGL
  - Allows more efficient use of GPU resources
- If you’re familiar with “classic” graphics pipelines, modern OpenGL doesn’t support
  - Fixed-function graphics operations
    - lighting
    - transformations
- All applications must use shaders for their graphics processing

# OpenGL and Graphics

- Computer Graphics is about faking reality in a convincing manner.
- Or creating alternative reality.

# Fake reality

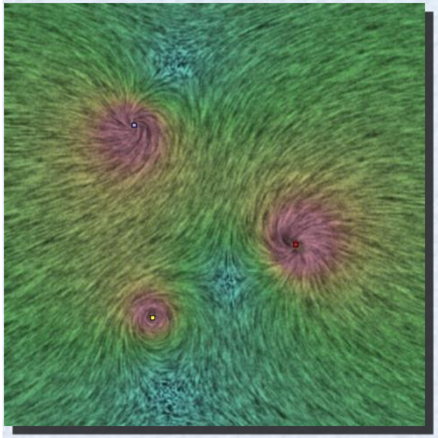
# Fake reality

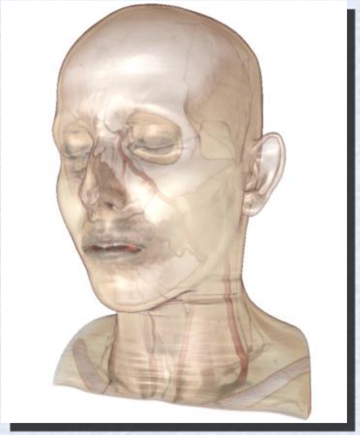
# Alternative reality

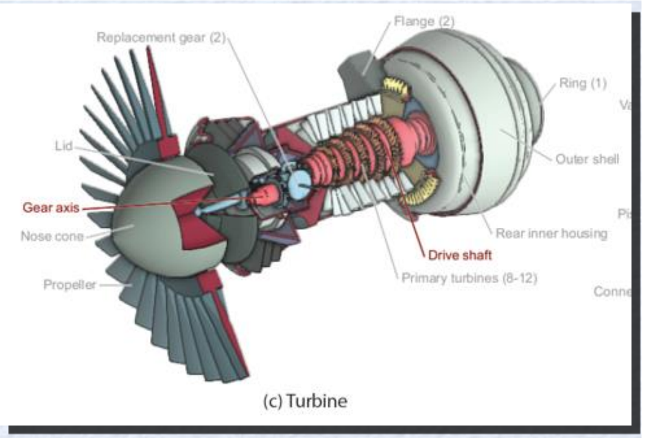
# Computer Graphics is

- 3D modeling / geometry
- Simulation / animation / character animation
- Lighting / light transfer
- Textures and color
- Post – processing / image processing
- Camera tricks / optics

# Computer Graphics is ….

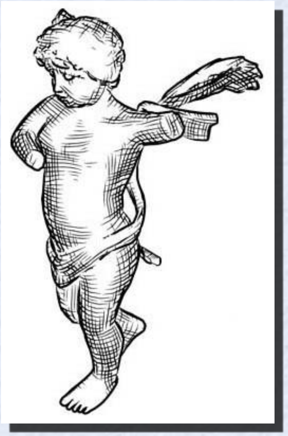


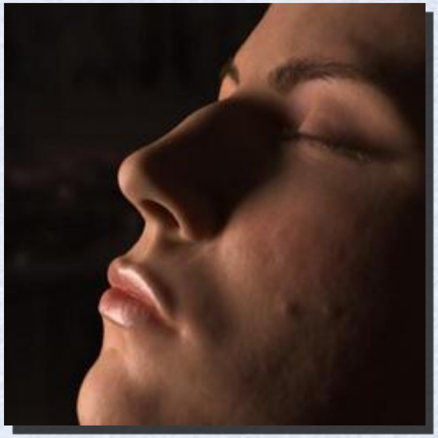





Scientific Visualization!

Illustration!





NPR / Art!

Computational Photography!

Virtual Life!

# Course outline/Syllabus

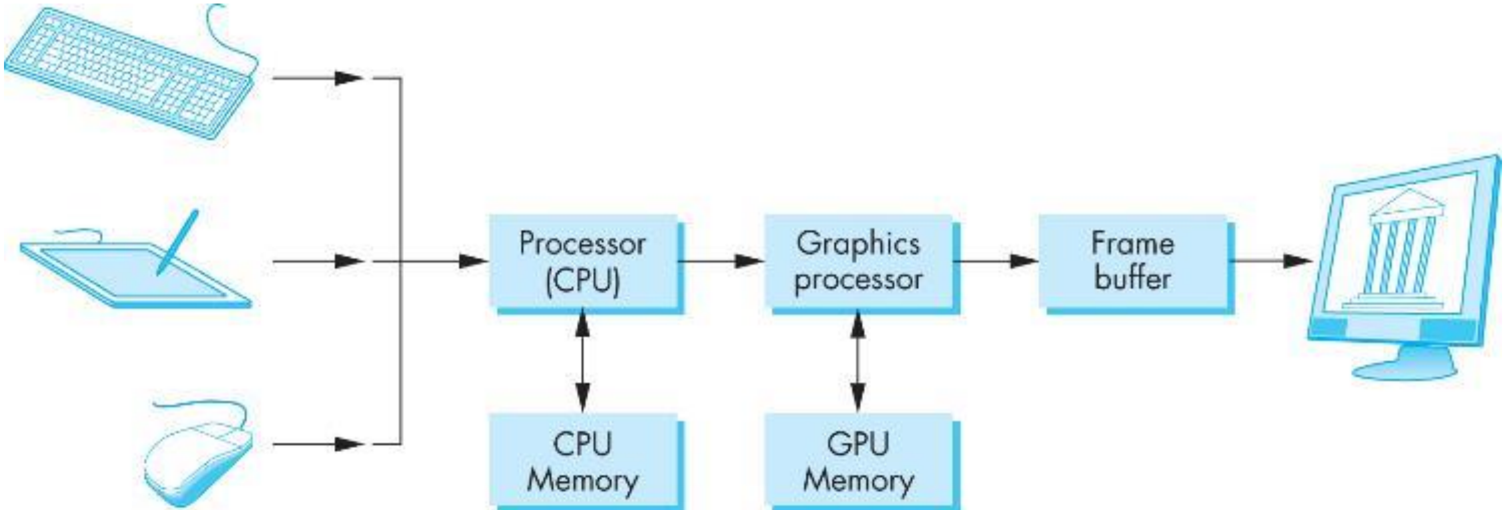
- **Introduction and Motivation**
  - Graphics history
  - 2D and 3D graphics
  - Graphics programming
  - Graphics software and hardware systems
- **Graphics System Hardware**
  - Hardware, display devices, I/O peripherals
  - Vector and raster graphics system
  - Interaction techniques
- **Fundamental Mathematics and Geometry**
  - Basic mathematics relevant to graphics
  - Coordinate systems
  - Points, lines, planes, and normals
  - Triangles and polygons
  - Geometric primitives
  - Curves, and surfaces
  - Solid and volumetric models
  - 2D and 3D geometric transformation
  - Parallel and perspective projection
- **Scene composition**
  - Coordinate system
  - 2D and 3D geometric transformation
  - Object hierarchies
  - Viewing and clipping
  - Parallel and perspective projection Object and image order rendering
- **Rendering**
  - Rendering pipeline
  - Scan-conversion: lines and polygons
  - Shading/lighting (illumination models) Visibility
- **Image-based techniques**
  - Sampling
  - Filtering
  - Anti-aliasing
- **Others**
  - Animation
  - Transparency and shadows
  - Texture mapping
  - Ray tracing, radiosity
  - Image-based rendering and modeling

# Grading

- Projects – 30%
- Homeworks + Exercises – 20%
- Midterm exam – 20%
- Final exam – 20%
- PPT – 10%

- Homeworks, Exercises, Projects and PPTs are assumed team assignments unless otherwise stated
- Exams are individual
- Cheating is bad – you will not learn much, both of us will have to deal with paperwork, administration and expulsion – it is not worth it.

# Basic Graphics System




Output device

Input devices

Image formed in frame buffer

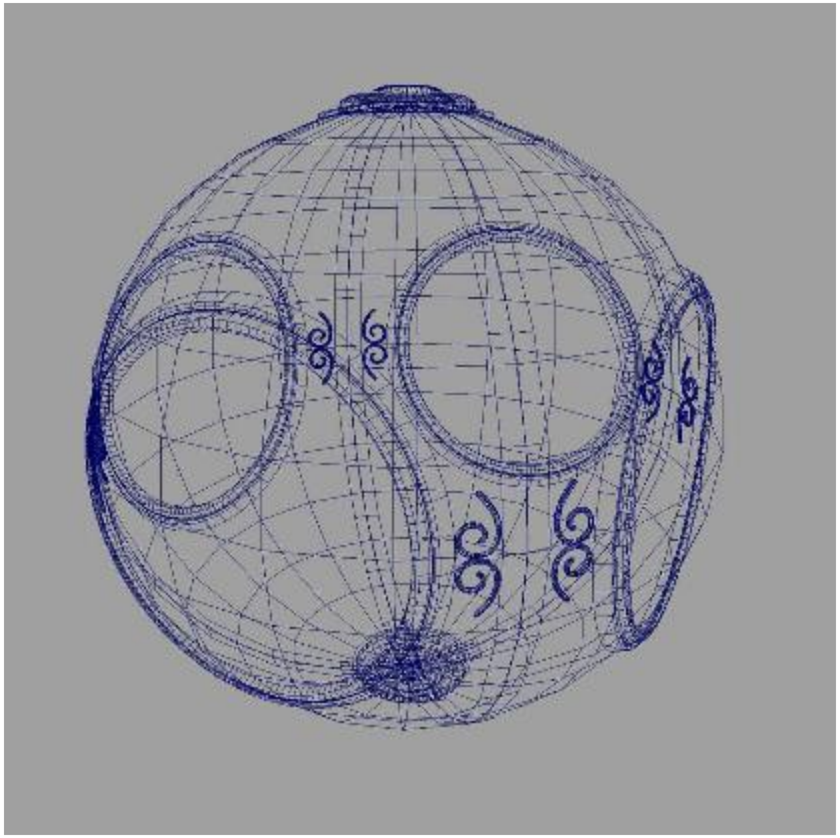
# Computer Graphics: 1950-1960

- Computer graphics goes back to the earliest days of computing
  - Strip charts
  - Pen plotters
  - Simple displays using A/D converters to go from computer to calligraphic CRT
- Cost of refresh for CRT too high
  - Computers slow, expensive, unreliable

# Computer Graphics: 1960-1970

- *Wireframe* graphics
  - Draw only lines
- Sketchpad
- Display Processors
- Storage tube

wireframe representation
of sun object

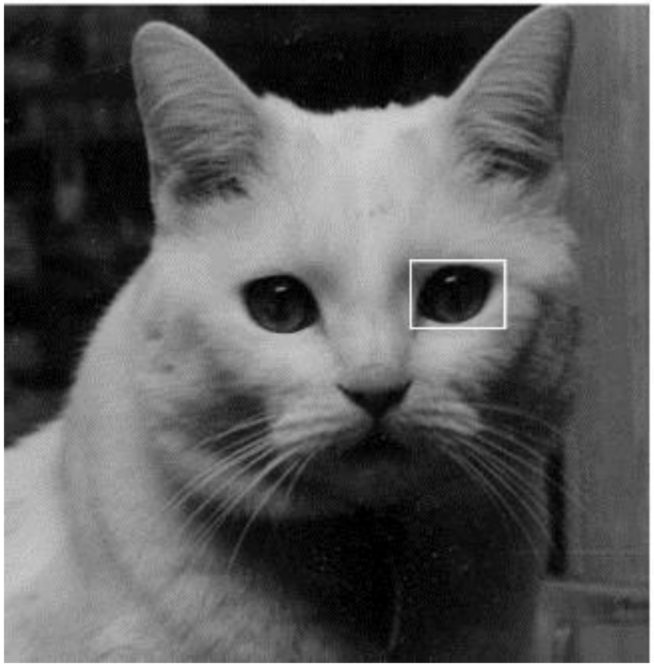


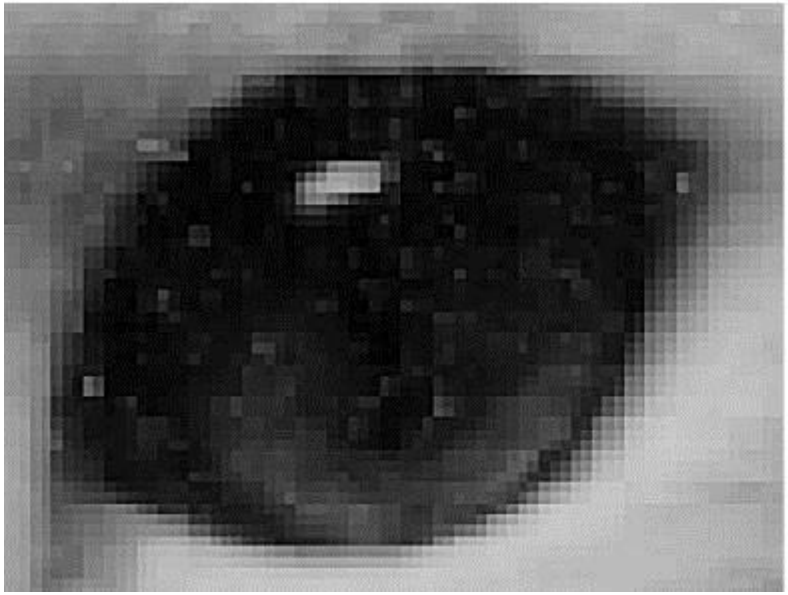
# Computer Graphics: 1970-1980

- Raster Graphics
- Beginning of graphics standards
  - IFIPS
    - GKS: European effort
      - Becomes ISO 2D standard
    - Core: North American effort
      - 3D but fails to become ISO standard
- Workstations and PCs

# Raster Graphics

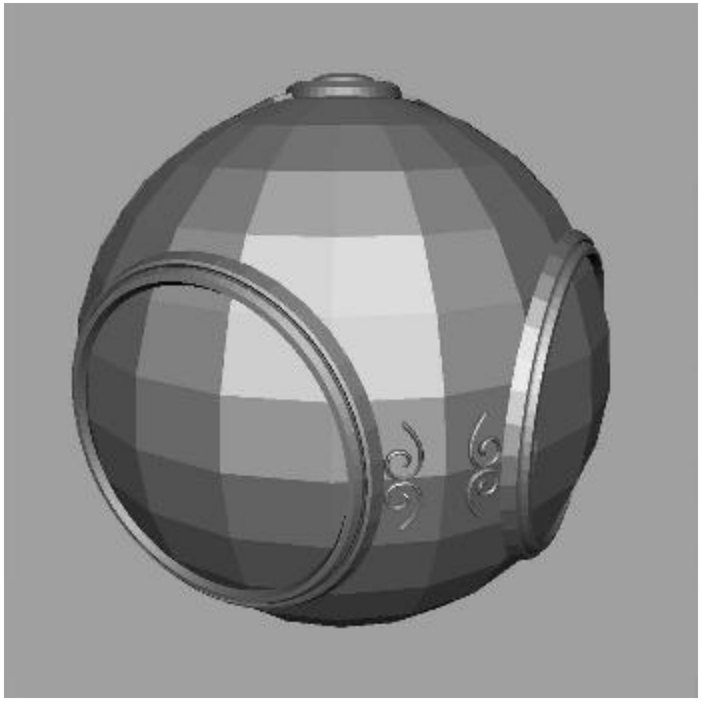
- Image produced as an array (the *raster*) of picture elements (*pixels*) in the *frame buffer*





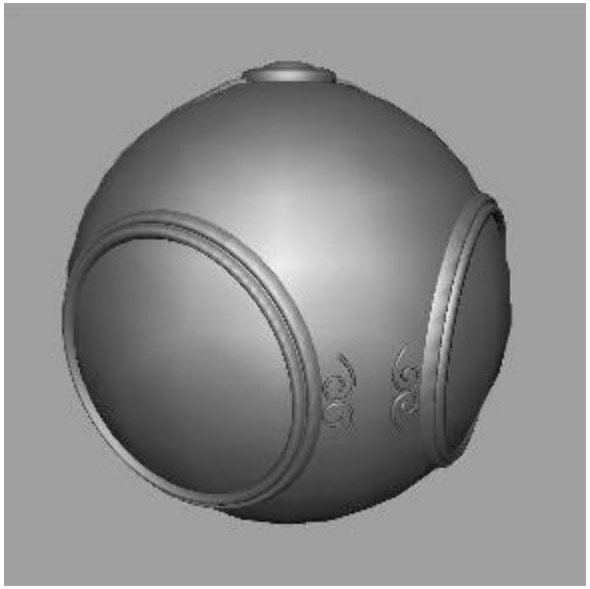
# Raster Graphics

- Allows us to go from lines and wire frame images to filled polygons



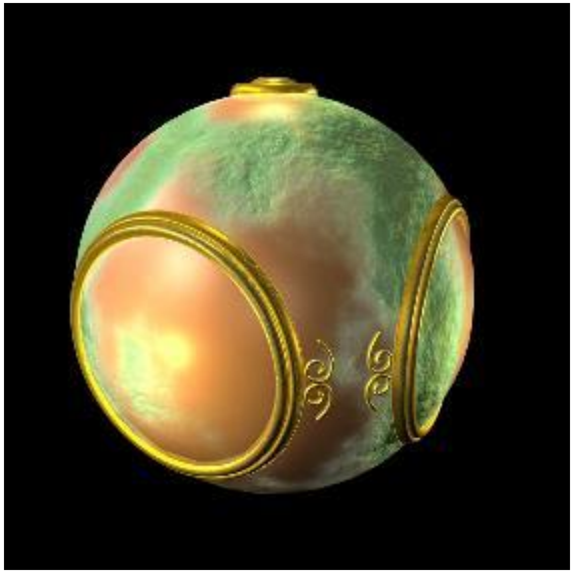
# Computer Graphics: 1980-1990

Realism comes to computer graphics



smooth shading

environment mapping



bump mapping

# Computer Graphics: 1980-1990

- Special purpose hardware
  - Silicon Graphics geometry engine
    - VLSI implementation of graphics pipeline
- Industry-based standards
  - PHIGS
  - RenderMan
- Networked graphics: X Window System
- Human-Computer Interface (HCI)

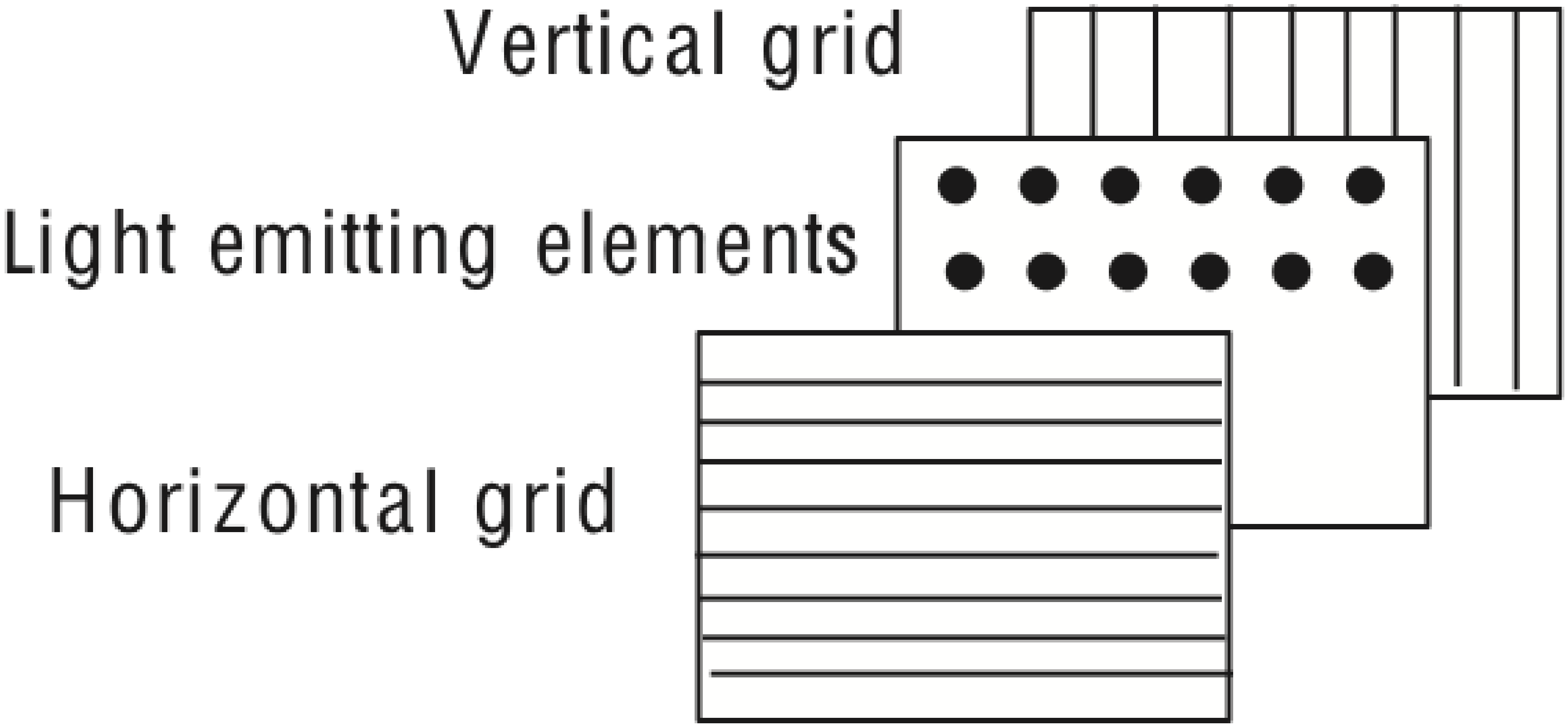
# Computer Graphics: 1990-2000

- OpenGL API
- Completely computer-generated feature-length movies (Toy Story) are successful
- New hardware capabilities
  - Texture mapping
  - Blending
  - Accumulation, stencil buffers

# Computer Graphics: 2000-2010

- Photorealism
- Graphics cards for PCs dominate market
  - Nvidia, ATI
- Game boxes and game players determine direction of market
- Computer graphics routine in movie industry: Maya, Lightwave
- Programmable pipelines

# Generic Flat Panel Display

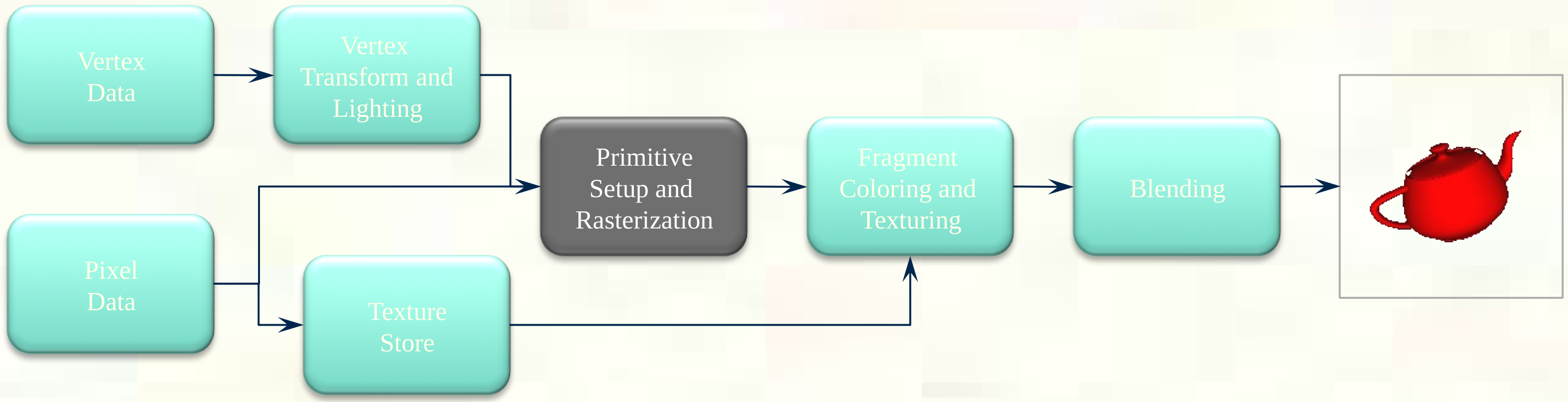


# Computer Graphics 2011-

- Graphics is now ubiquitous
  - Cell phones
  - Embedded
- OpenGL ES and WebGL
- Alternate and Enhanced Reality
- 3D Movies and TV

# OpenGL pipeline history

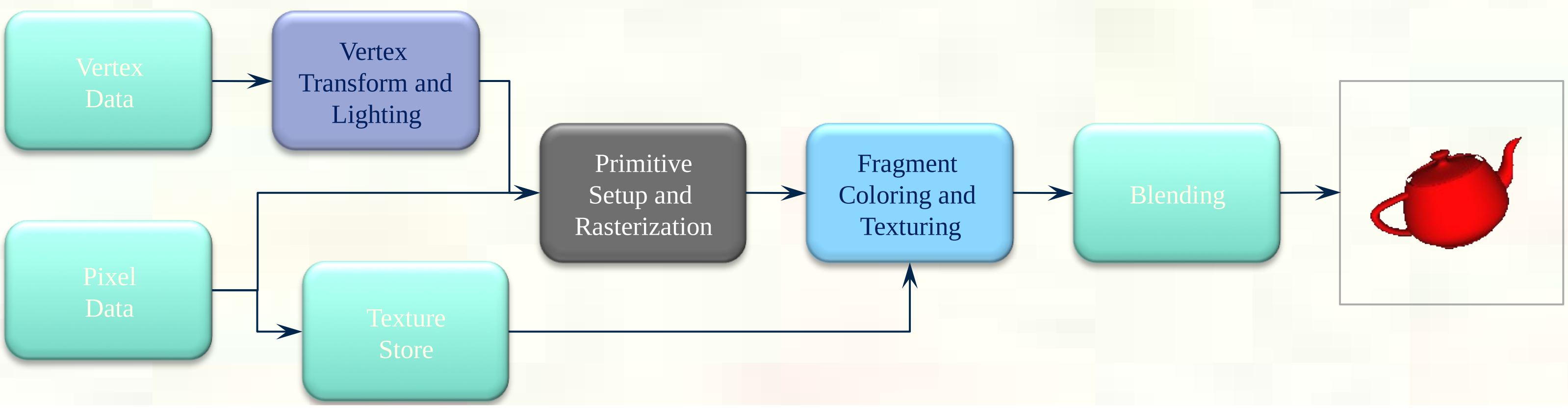
- OpenGL 1.0 was released on July 1st, 1994
- Its pipeline was entirely *fixed-function*
  - the only operations available were fixed by the implementation




- The pipeline evolved, but remained fixed-function through OpenGL versions 1.1 through 2.0 (Sept. 2004)

# OpenGL pipeline history

- OpenGL 2.0 (officially) added programmable shaders
  - *vertex shading* augmented the fixed-function transform and lighting stage
  - *fragment shading* augmented the fragment coloring stage
- However, the fixed-function pipeline was still available



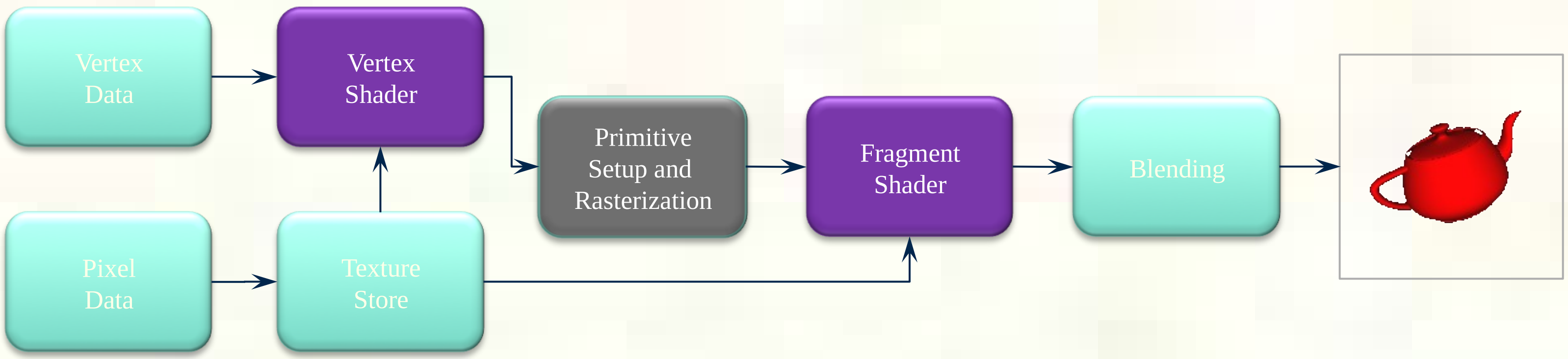
# OpenGL pipeline history

- OpenGL 3.0 introduced the *deprecation model*
  - the method used to remove features from OpenGL
- The pipeline remained the same until OpenGL 3.1 (released March 24th, 2009)
- Introduced a change in how OpenGL contexts are used

| Context Type | Description |
|---|---|
| Full | Includes all features (including those marked deprecated) available in the current version of OpenGL |
| Forward Compatible | Includes all non-deprecated features (i.e., creates a context that would be similar to the next version of OpenGL) |

# Exclusively programmable pipeline

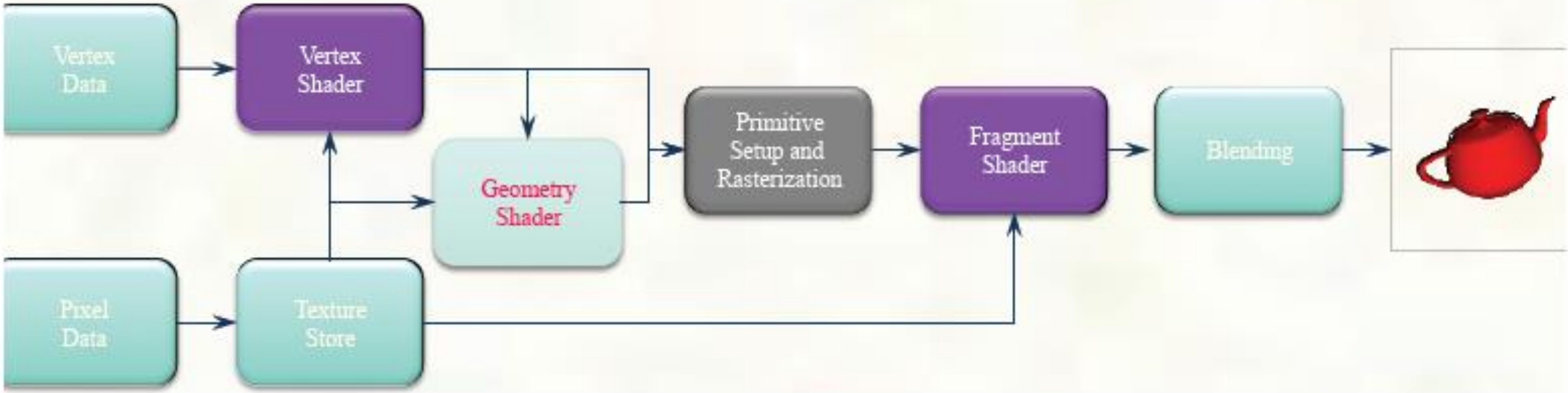
- OpenGL 3.1 removed the fixed-function pipeline
  - programs were required to use only shaders




- Additionally, almost all data is *GPU-resident*
  - all vertex data sent using buffer objects

# More programmability

- OpenGL 3.2 (released August 3rd, 2009) added an additional shading stage – *geometry shaders*



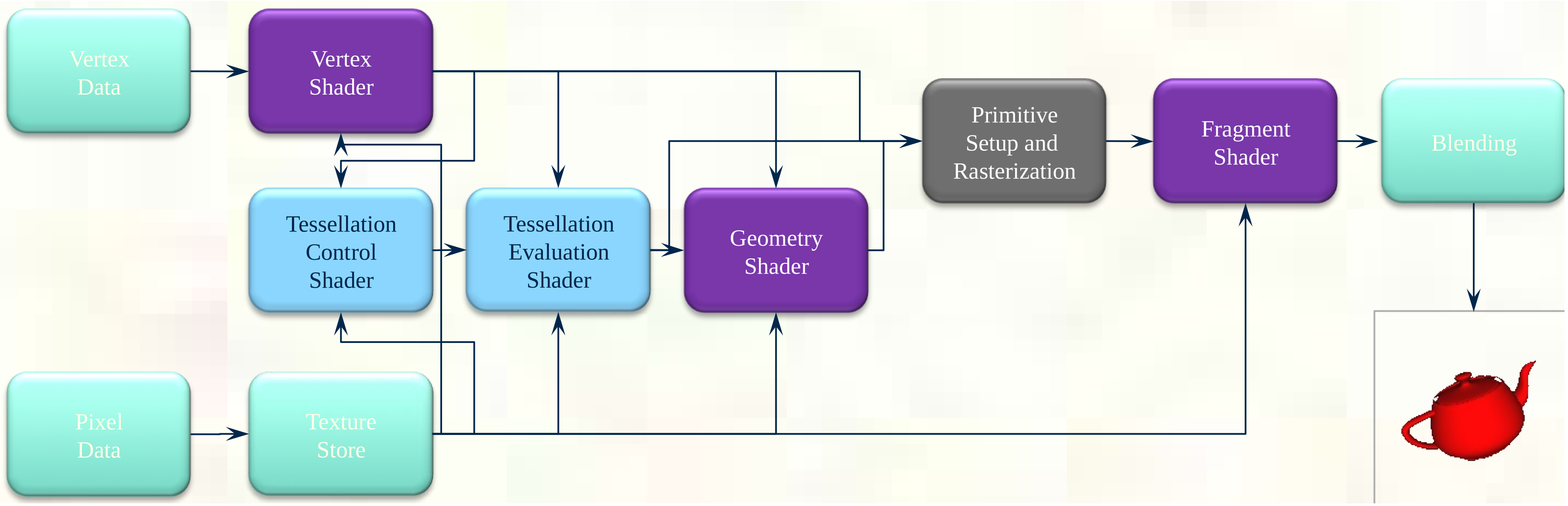
# Context profiles

- OpenGL 3.2 also introduced *context profiles*
  - profiles control which features are exposed
  - currently two types of profiles: *core* and *compatible*

| Context Type | Profile | Description |
|---|---|---|
| Full | core | All features of the current release |
| | compatible | All features ever in OpenGL |
| Forward Compatible | core | All non-deprecated features |
| | compatible | Not supported |

# Latest

- OpenGL 4.1 (released July 25th, 2010) included additional shading stages – *tessellation-control and tessellation-evaluation shaders*

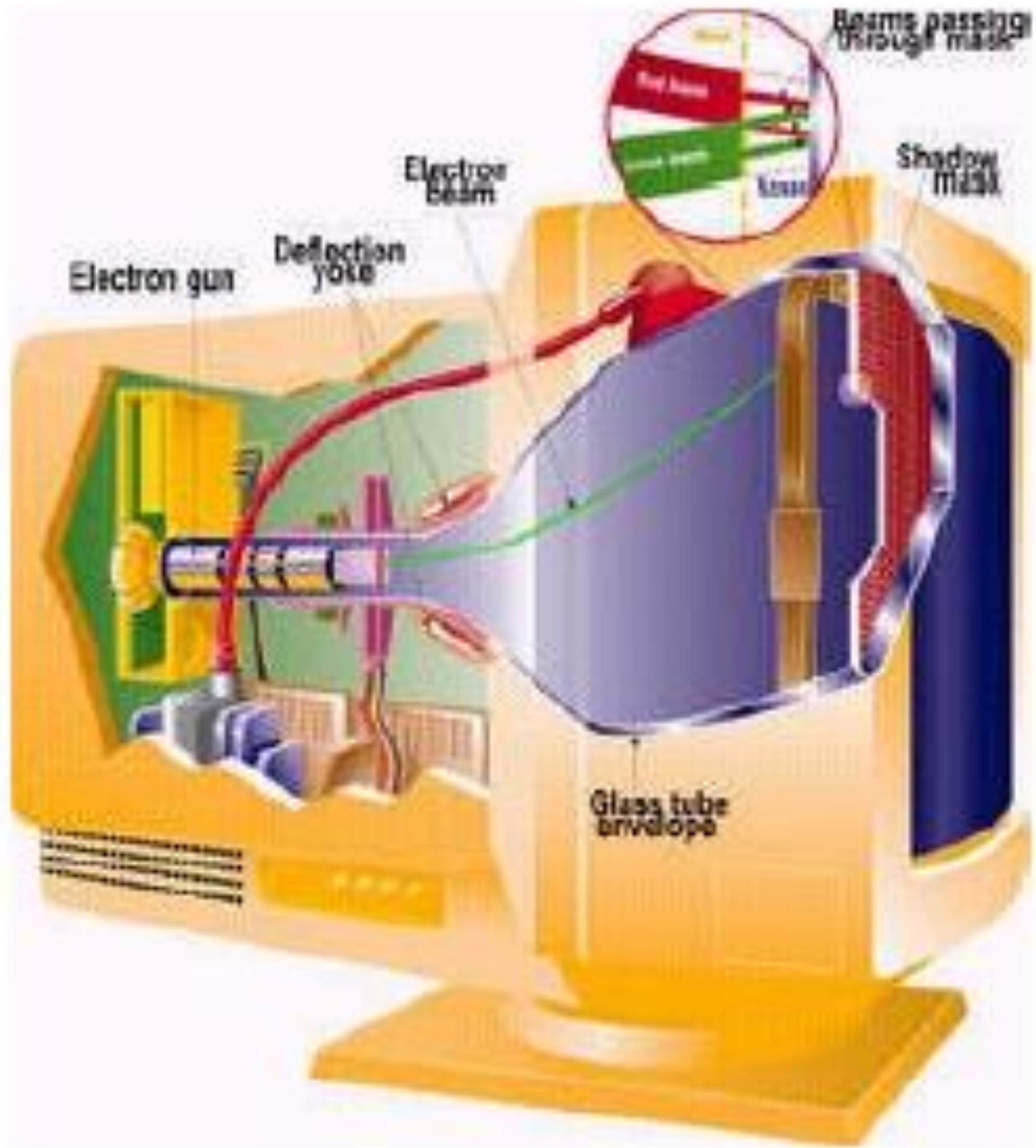


# OpenGL ES and WebGL

- OpenGL ES 2.0
  - Designed for embedded and hand-held devices such as cell phones
  - Based on OpenGL 3.1
  - Shader based
- WebGL
  - JavaScript implementation of ES 2.0
  - Runs on most recent browsers

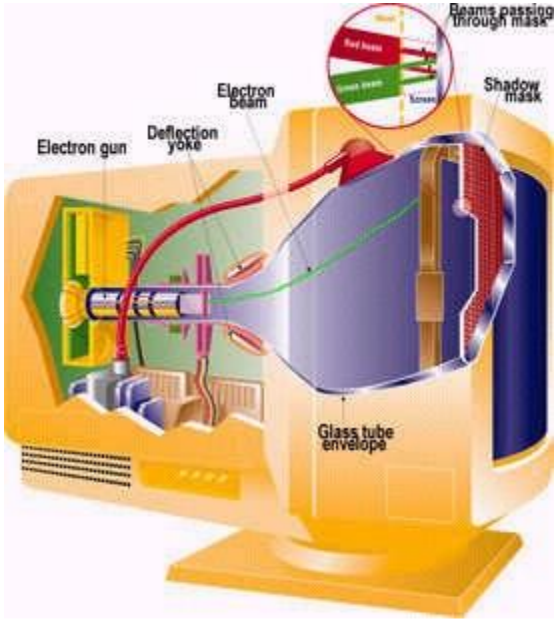
# CRT details

- Heating element (filament)
- Electrons pulled towards anode focusing cylinder
- Vertical and horizontal deflection plates
- Beam strikes phosphor coating on front of tube



# Display technologies

- Cathode Ray Tubes (CRTs)
  - Most common display device today
  - Evacuated glass bottle
  - Extremely high voltage



# Electron gun

- Contains a filament that, when heated, emits a stream of electrons
- Electrons are focused with an electromagnet into a sharp beam and directed to a specific point of the face of the picture tube
- The front surface of the picture tube is coated with small phospher dots
- When the beam hits a phospher dot it glows with a brightness proportional to the strength of the beam and how long it is hit

# CRT characteristics

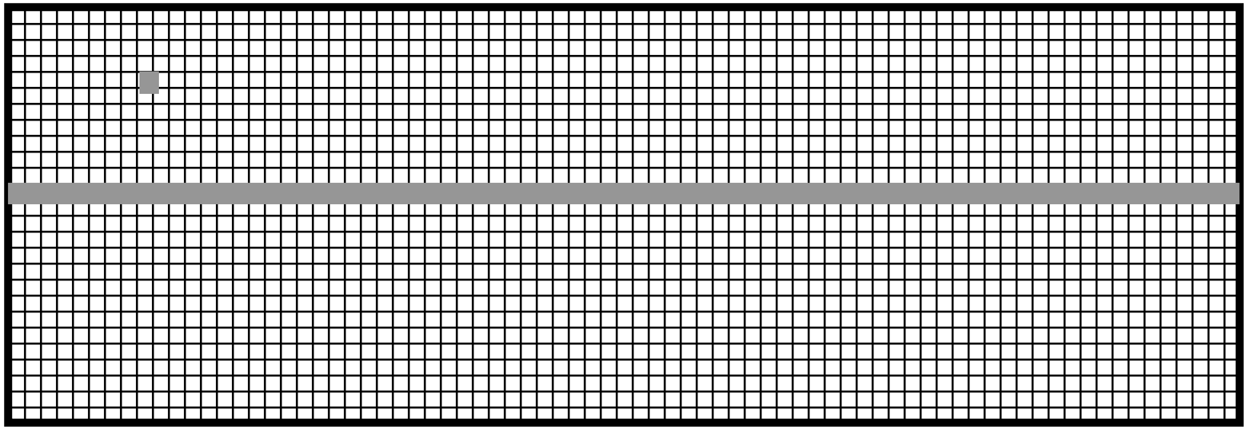
- What is the largest (diagonal) CRT you have seen?
  - Why is that the largest?
    - Evacuated tube == massive glass
    - Symmetrical electron paths (corners vs. center)
- How might one measure CRT capabilities?
  - Size of tube
  - Brightness of phosphers vs. darkness of tube
  - Speed of electron gun
  - Width of electron beam
  - Pixels?

# Display technologies: CRTs

- Vector Displays
  - Early computer displays: basically an oscilloscope
  - Control X,Y with vertical/horizontal plate voltage
  - Often used intensity as Z (close things were brighter)
- Name two disadvantages
- Just does wireframe
- Complex scenes cause visible flicker

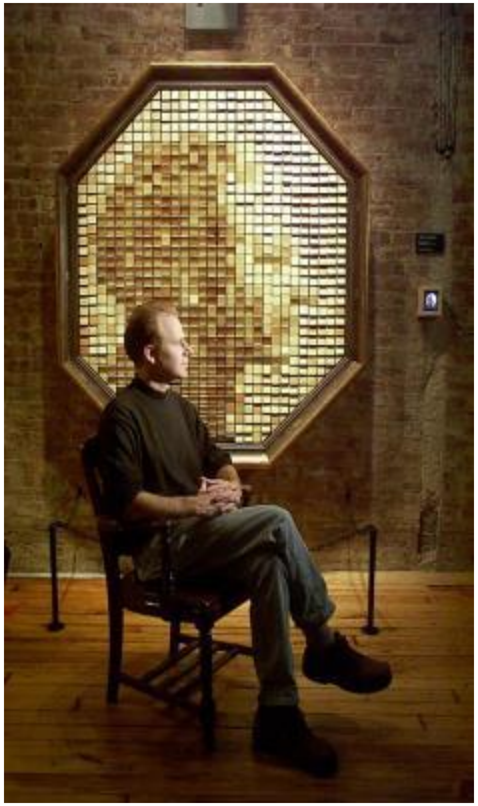
# Display technologies: CRTs

- Raster Displays
  - Raster: A rectangular array of points or dots
  - Pixel: One dot or picture element of the raster
  - Scan line: A row of pixels



# What is a pixel?

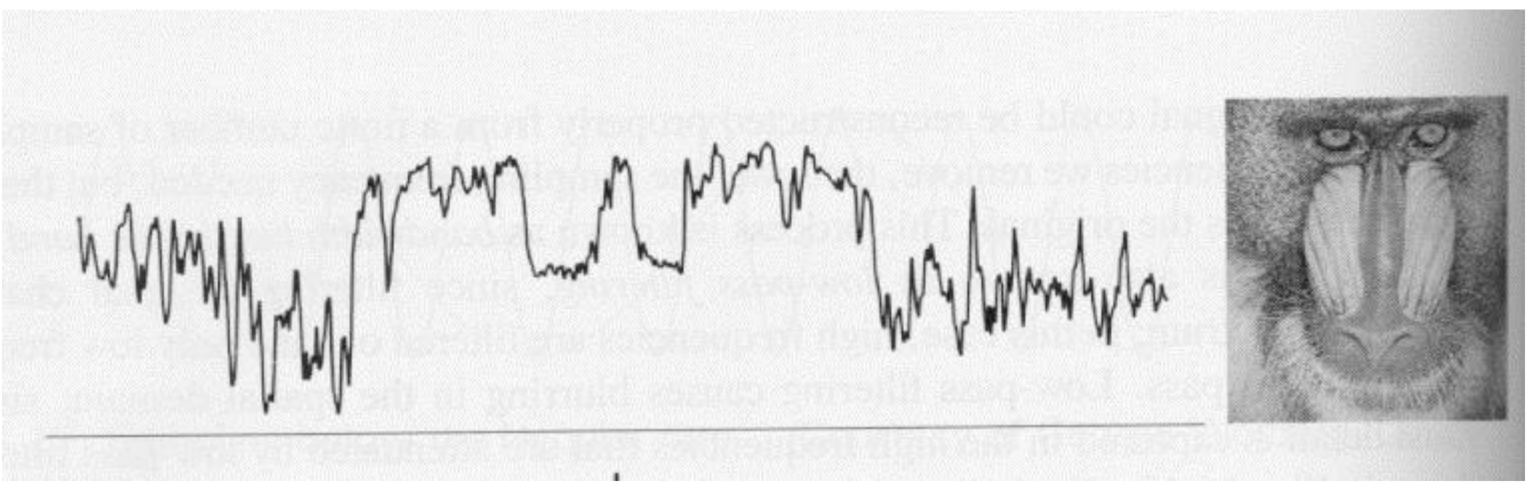
- Wood chips Chrome spheres Trash



Daniel Rozin – NYU: (movies) http://fargo.itp.tsoa.nyu.edu/~danny/art.html

# Display technologies: CRTs

- Raster Displays
  - Black and white television: an oscilloscope with a fixed scan pattern: left to right, top to bottom
    - As beam sweeps across entire face of CRT, beam intensity changes to reflect brightness
  - Analog signal vs. digital display



# Display technologies: CRT

- Can a computer display work like a black and white TV?
  - Must synchronize
    - Your program makes decisions about the intensity signal at the pace of the CPU…
    - The screen is “painted” at the pace of the electron gun scanning the raster
  - *Solution: special memory to buffer image with scan-out synchronous to the raster. We call this the* framebuffer*.*
  - *Digital description to analog signal to digital display*

# Display Technologies: CRTs

- Phosphers
  - Fluorescence: Light emitted while the phospher is being struck by electrons
  - Phosphorescence: Light emitted once the electron beam is removed
  - Persistence: The time from the removal of the excitation to the moment when phosphorescence has decayed to 10% of the initial light output

# Display Technologies: CRTs

- Refresh
  - Frame must be “refreshed” to draw new images
  - As new pixels are struck by electron beam, others are decaying
  - Electron beam must hit all pixels frequently to eliminate flicker
  - Critical fusion frequency
    - Typically 60 times/sec
    - Varies with intensity, individuals, phospher persistence, lighting...

# Display Technologies: CRTs

- Interlaced Scanning
  - Assume we can only scan all pixels of entire screen 30 times / second
  - To reduce flicker, divide frame into two "fields" of odd and even lines

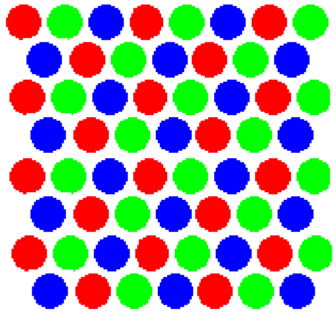
| 1/30 Sec | | 1/30 Sec | |
|---|---|---|---|
| 1/60 Sec | 1/60 Sec | 1/60 Sec | 1/60 Sec |
| Field 1 | Field 2 | Field 1 | Field 2 |
| Frame | | Frame | |

# Display Technologies: CRTs

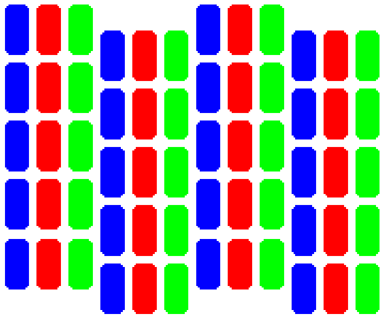
- CRT timing
  - Scanning (left to right, top to bottom)
    - Vertical Sync Pulse: Signals the start of the next field
    - Vertical Retrace: Time needed to get from the bottom of the current field to the top of the next field
    - Horizontal Sync Pulse: Signals the start of the new scan line
    - Horizontal Retrace: The time needed to get from the end of the current scan line to the start of the next scan line

# Display Technology: Color CRTs

- Color CRTs are much more complicated
  - Requires manufacturing very precise geometry
  - Uses a pattern of color phosphors on the screen:



Delta electron gun arrangement

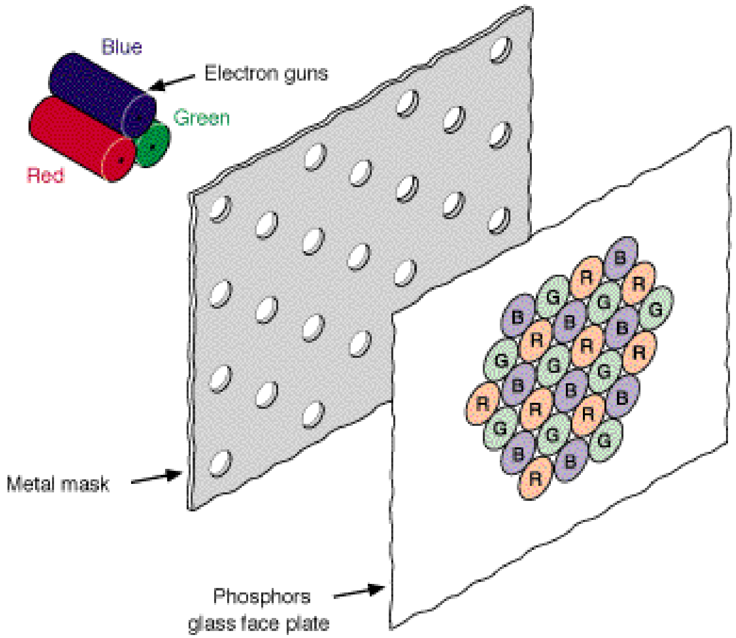


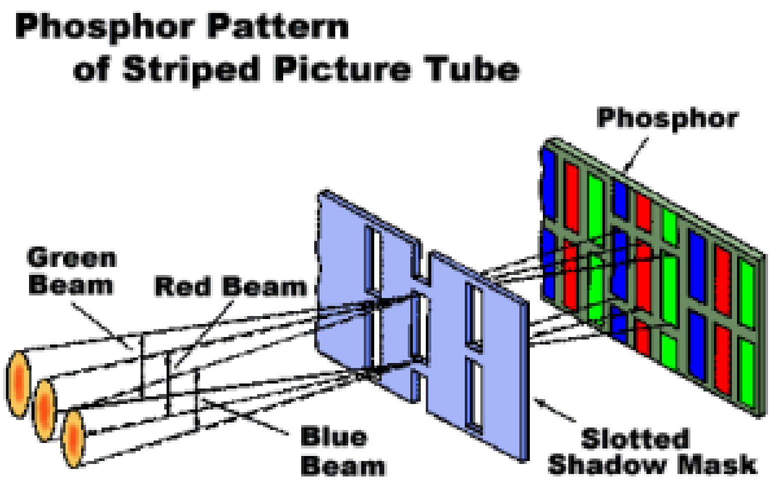
In-line electron gun arrangement

  - *Why red, green, and blue phosphors?*

# Display Technology: Color CRTs

- Color CRTs have
  - Three electron guns
  - A metal *shadow mask* to differentiate the beams



# Display Technology: Raster

- Raster CRT pros:
  - Allows solids, not just wireframes
  - Leverages low-cost CRT technology (i.e., TVs)
  - Bright!  Display *emits* light
- Cons:
  - Requires screen-size memory array
  - Discreet sampling (pixels)
  - Practical limit on size (call it 40 inches)
  - Bulky
  - Finicky (convergence, warp, etc)

# CRTs – A Review

- Raster Displays (early 70s)
  - like television, scan all pixels in regular pattern
  - use frame buffer (video RAM) to eliminate sync problems
- RAM
  - ¼ MB (256 KB) cost $2 million in 1971
  - Do some math…
    - 1280 x 1024 screen resolution = 1,310,720 pixels
    - Monochrome color (binary) requires 160 KB
    - High resolution color requires 5.2 MB

# Movie Theaters

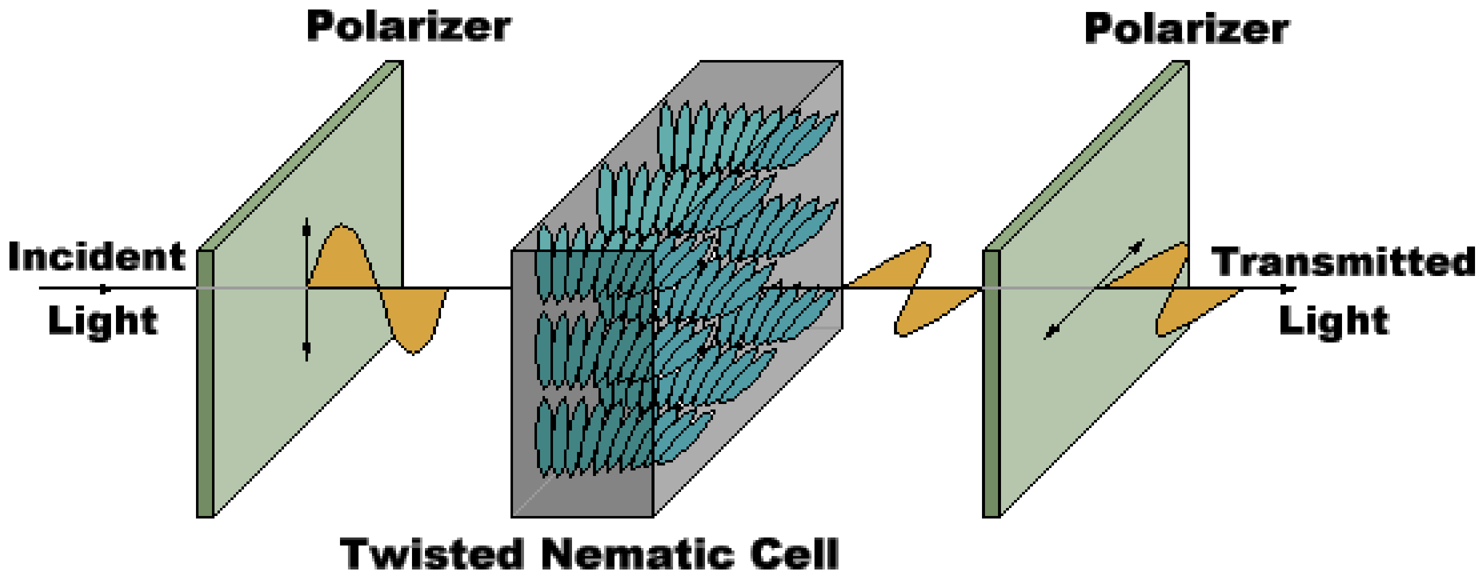
- U.S. film projectors play film at 24 fps
  - Projectors have a shutter to block light during frame advance
  - To reduce flicker, shutter opens twice for each frame – resulting in 48 fps flashing
  - 48 fps is perceptually acceptable
- European film projectors play film at 25 fps
  - American films are played ‘as is’ in Europe, resulting in everything moving 4% faster
  - Faster movements and increased audio pitch are considered perceptually acceptable

# Viewing Movies at Home

- Film to DVD transfer
  - Problem: 24 film fps must be converted to
    - NTSC U.S. television interlaced 29.97 fps 768x494
    - PAL Europe television 25 fps 752x582
- Use 3:2 Pulldown
  - First frame of movie is broken into first three fields (odd, even, odd)
  - Next frame of movie is broken into next two fields (even, odd)
  - Next frame of movie is broken into next three fields (even, odd, even)…

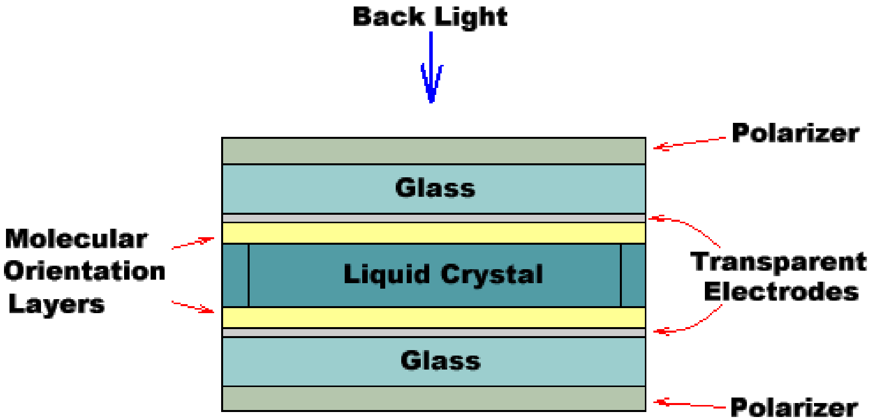
# Display Technology: LCDs

- 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 90º.



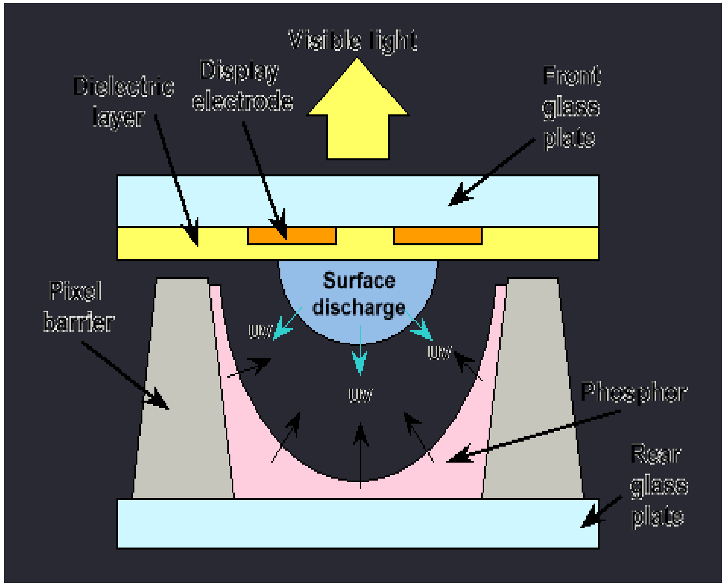
# Display Technology: LCDs

- Transmissive & reflective LCDs:
  - LCDs act as light valves, not light emitters, and thus rely on an external light source.
  - Laptop screen
    - backlit
    - transmissive display
  - Palm Pilot/Game Boy
    - reflective display



# Display Technology: Plasma

- Plasma display panels
  - Similar in principle to fluorescent light tubes
  - Small gas-filled capsules are excited by electric field, emits UV light
  - UV excites phosphor
  - Phosphor relaxes, emits some other color

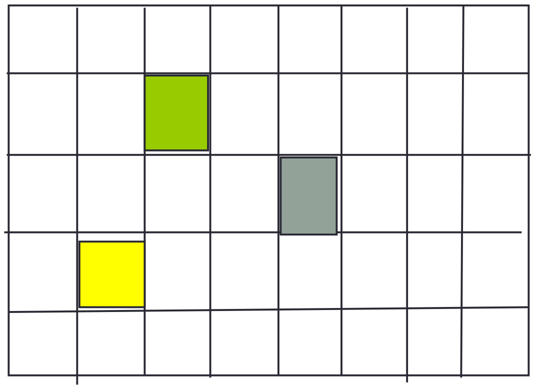


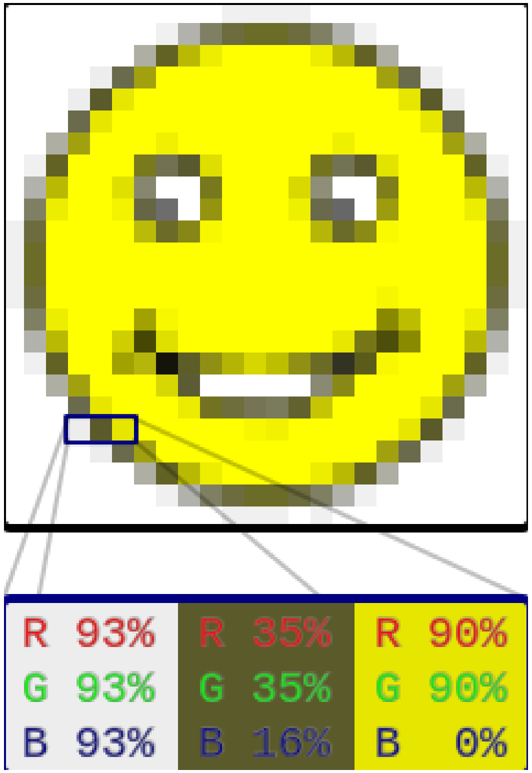
# Display Technology

- Plasma Display Panel Pros
  - Large viewing angle
  - Good for large-format displays
  - Fairly bright
- Cons
  - Expensive
  - Large pixels (~1 mm versus ~0.2 mm)
  - Phosphors gradually deplete
  - Less bright than CRTs, using more power

# Images and Raster Graphics

- Real world is continuous (almost)
- How to represent images on a display?
- Raster graphics: use a bitmap with discrete ***pixels***
- Raster scan CRT
- (paints image
  line by line)





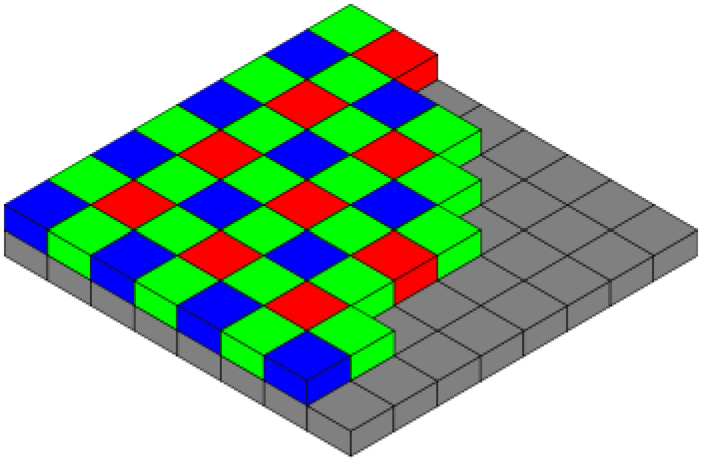

- Cannot be resized without loss
- Compare to vector graphics
  - Resized arbitrarily.  For drawings
  - But how to represent photos, CG?

# Displays and Raster Devices

- CRT, flat panel, television (rect array of pixels)
- Printers (scanning: no physical grid but print ink)
- Digital cameras (grid light-sensitive pixels)
- Scanner (linear array of pixels swept across)
- Store image as 2D array (of RGB [sub-pixel] values)
  - In practice, there may be resolution mismatch, resize
  - Resize across platforms (phone, screen, large TV)
- Vector image: description of shapes (line, circle, …)
  - E.g., line art such as in Adobe Illustrator
  - Resolution-Independent but must *rasterize* to display
  - Doesn’t work well for photographs, complex images

# Resolutions

- Size of grid (1920x1200 = 2,304,000 pixels)
  - 32 bit of memory for RGBA framebuffer 8+ MB
- For printers, pixel density (300 dpi or ppi)
  - Printers often binary or CMYK, require finer grid
  - iPhone “retina display” > 300 dpi. At 12 inches, pixels closer than retina’s ability to distinguish angles
- Digital cameras in Mega-Pixels (often > 10 MP)
  - Color filter array (Bayer Mosaic)
  - Pixels really small (micron)



# Monitor Intensities

- Intensity usually stored with 8 bits [0…255]
- HDR can be 16 bits or more [0…65535]
- Resolution-independent use [0…1] intermediate
- Monitor takes input value [0…1] outputs intensity
  - Non-zero intensity for 0, black level even when off
  - 1.0 is maximum intensity (output 1.0/0.0 is contrast)
  - Non-linear response (as is human perception)
  - 0.5 may map to 0.25 times the response of 1.0
  - Gamma characterization and gamma correction
  - Some history from CRT physics and exponential forms