Introduction to Computer Graphics

Graphics Systems and Models

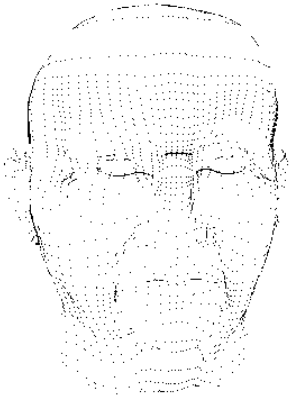
葉奕成 I-Cheng (Garrett) Yeh

Objectives

- In this lecture, we explore what computer graphics is about and survey some application areas
- We start with a historical introduction

Graphics Systems and Models

- What is Computer Graphics?
- ***Computer graphics*** deals with all aspects of creating images with a computer
  - Hardware
  - Software
  - Applications

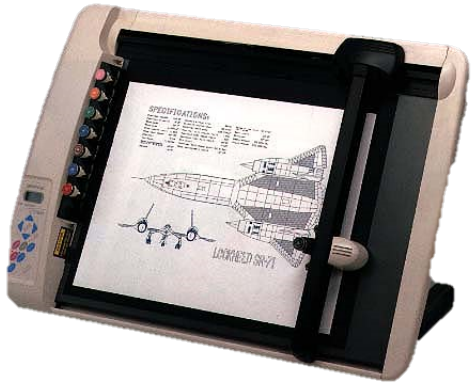


Example

- Where did this image come from?
- What hardware/software did we need to produce it?
  - Application: The object is an artist's rendition of the sun for an animation to be shown in a domed environment (planetarium)
  - Software: Maya for modeling and rendering but Maya is built on top of OpenGL
  - Hardware: PC with graphics cards for modeling and rendering

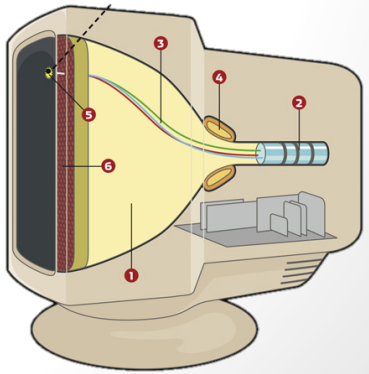
Computer Graphics: 1950-1960

- Computer graphics goes back to the earliest days of computing
  - Strip charts
  - Pen plotters

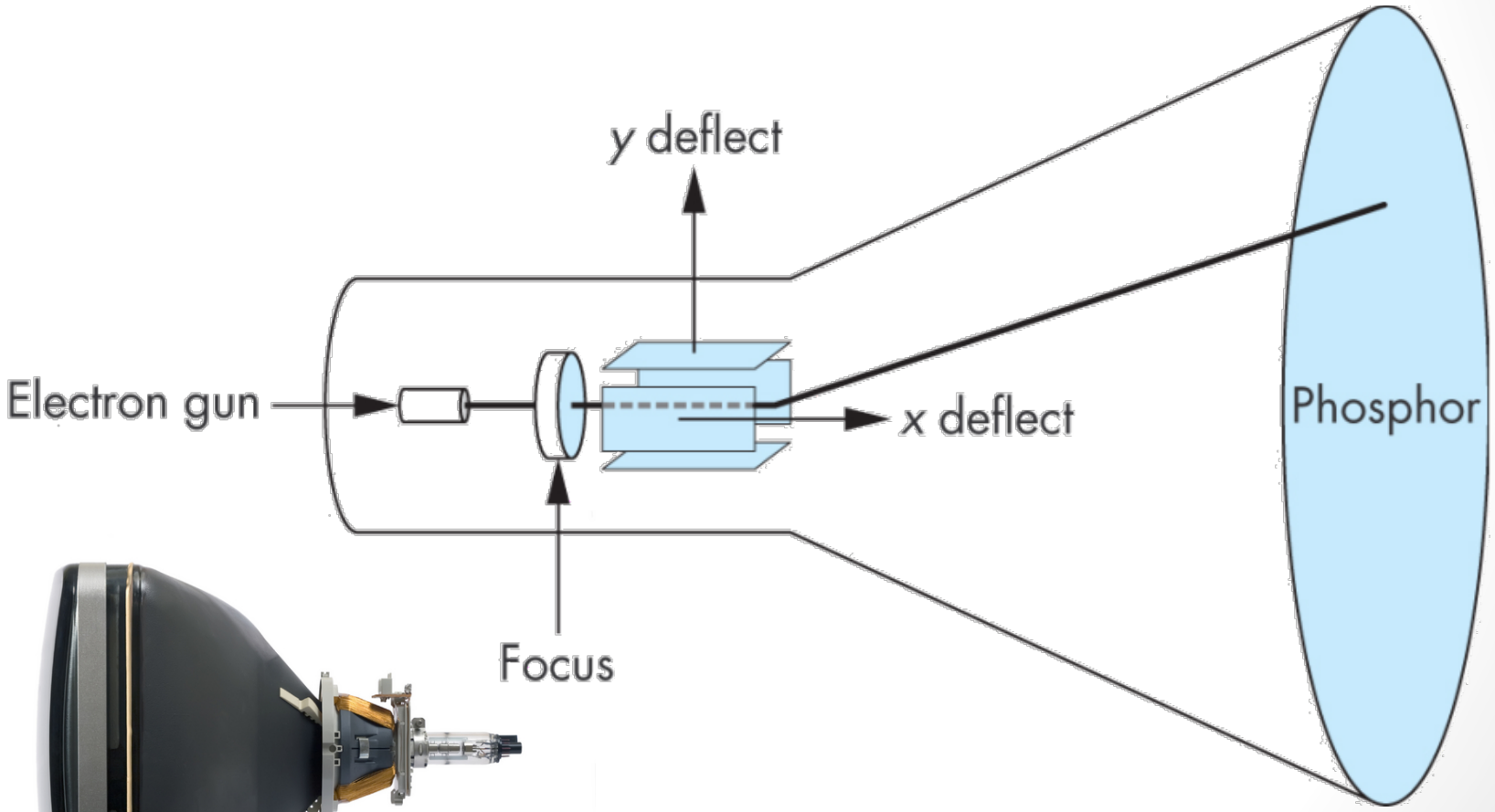


  - Simple displays using A/D converters to go from computer to calligraphic CRT (Line-drawing)

- Cost of refresh for CRT too high
  - Computers slow, expensive, unreliable

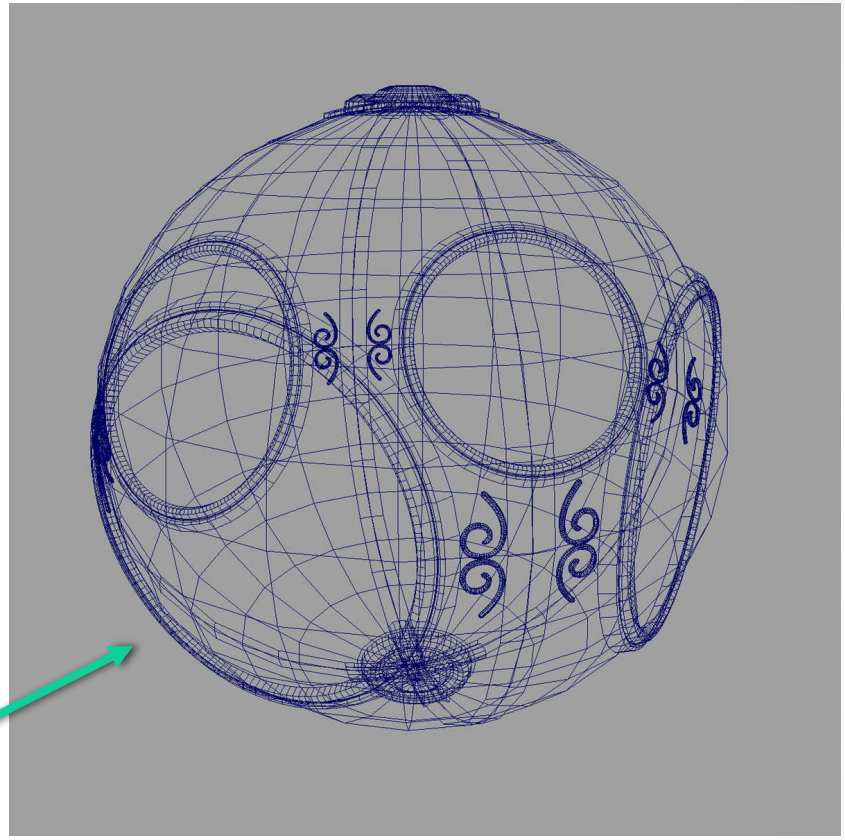


Cathode ray tubes (CRTs)



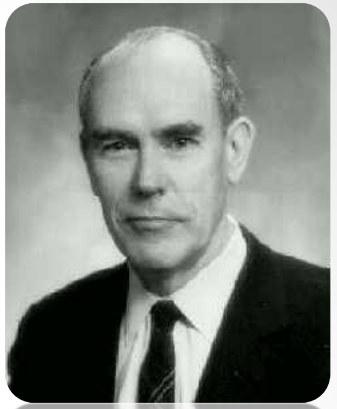
Computer Graphics: 1960-1970

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

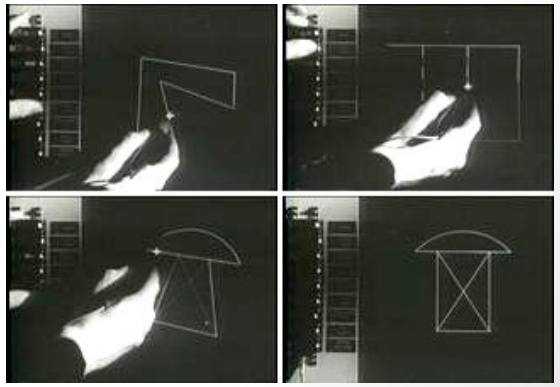


wireframe representation of sun object

Sketchpad



- Ivan Sutherland's PhD thesis at MIT
  - Recognized the potential of man-machine interaction
  - Loop
    - Display something
    - User moves light pen
    - Computer generates new display
  - Sutherland also created many of the now common algorithms for computer graphics

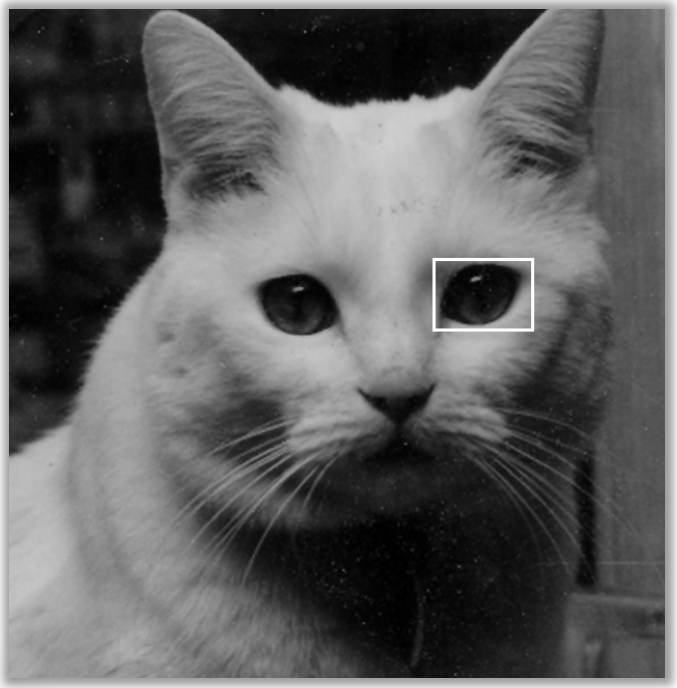


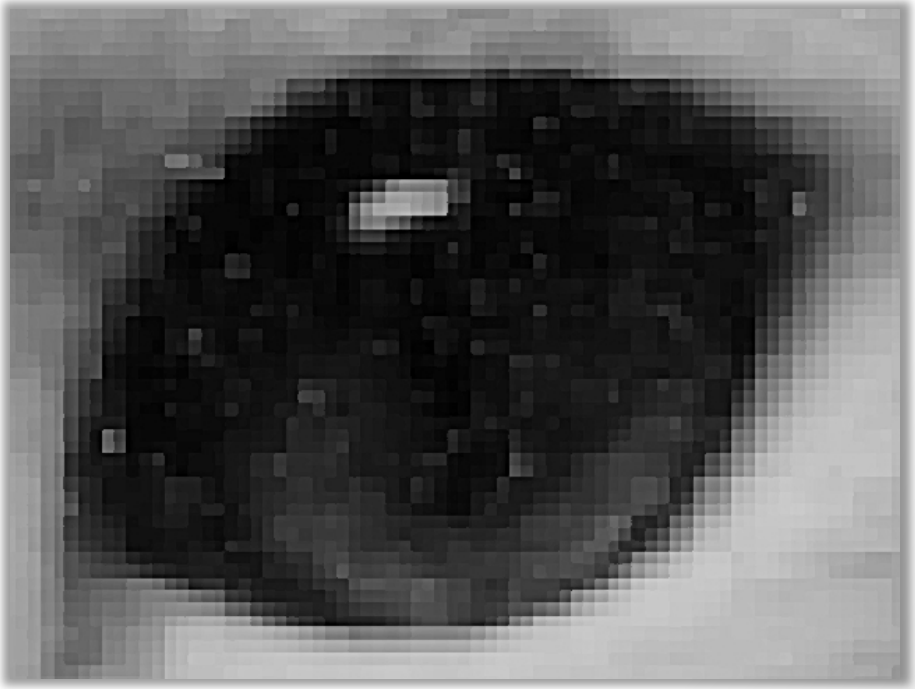
Computer Graphics: 1970-1980

- Raster Graphics
  - A dot matrix data structure representing a generally rectangular grid of pixels, or points of color, viewable via a monitor, paper, or other display medium
- 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

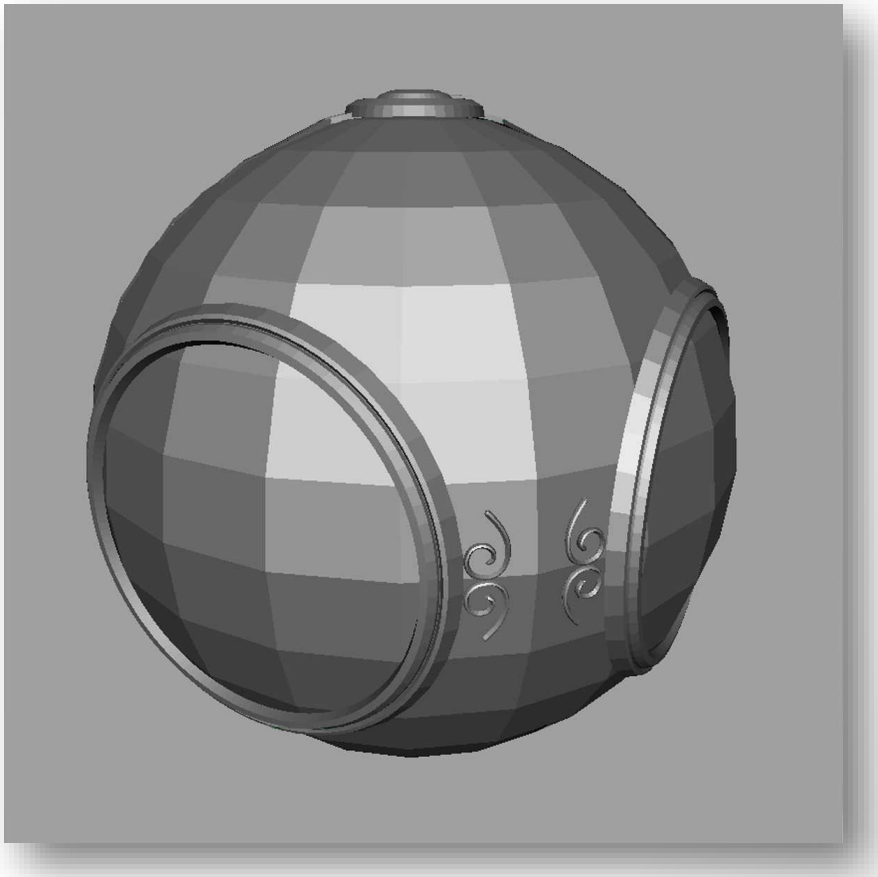
- Image produced as an array (the raster) of picture elements (pixels) in the frame buffer





Raster Graphics

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

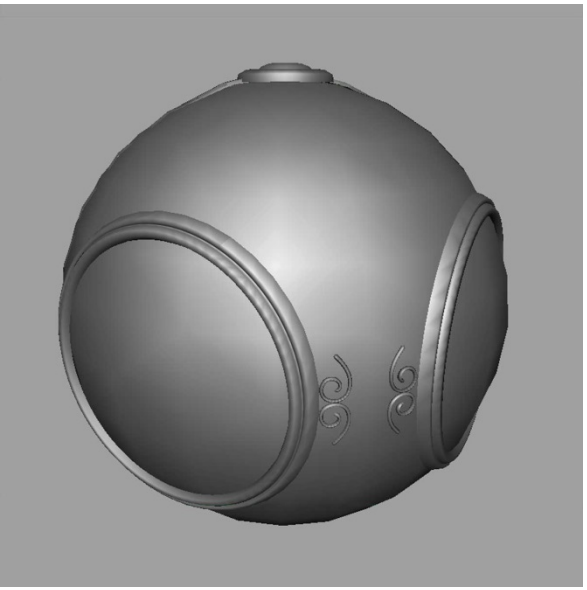


PCs and Workstations

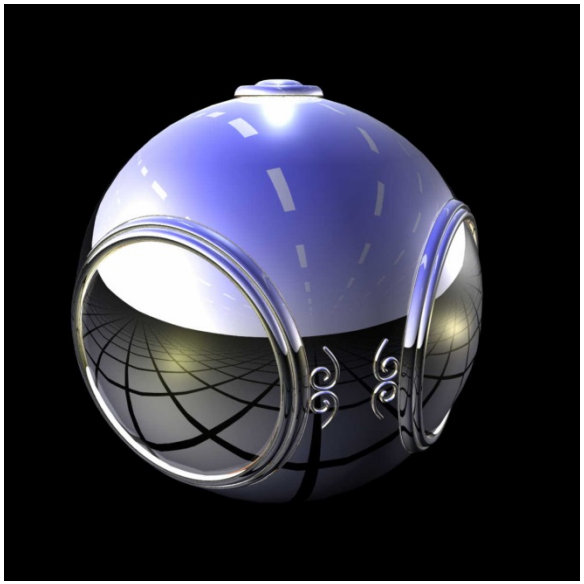
- Although we no longer make the distinction between workstations and PCs, historically they evolved from different roots
  - Early workstations characterized by
    - Networked connection: client-server model
    - High-level of interactivity, High performance rendering.
  - Early PCs included frame buffer as part of user memory
    - Easy to change contents and create images
    - Display existed content

Computer Graphics: 1980-1990

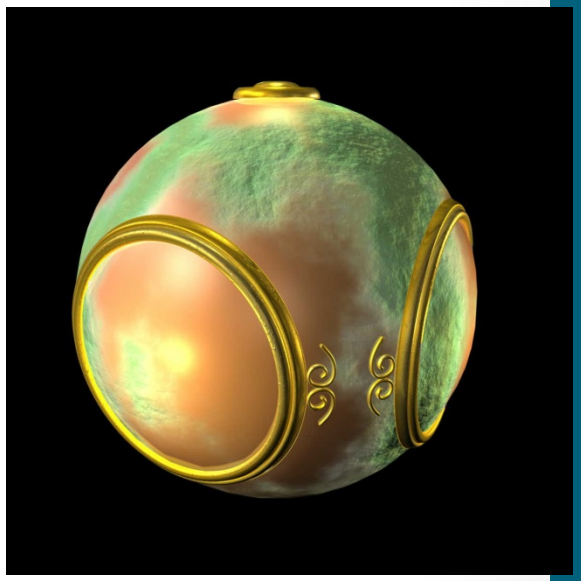
- Realism comes to computer graphics



smooth shading



environment mapping



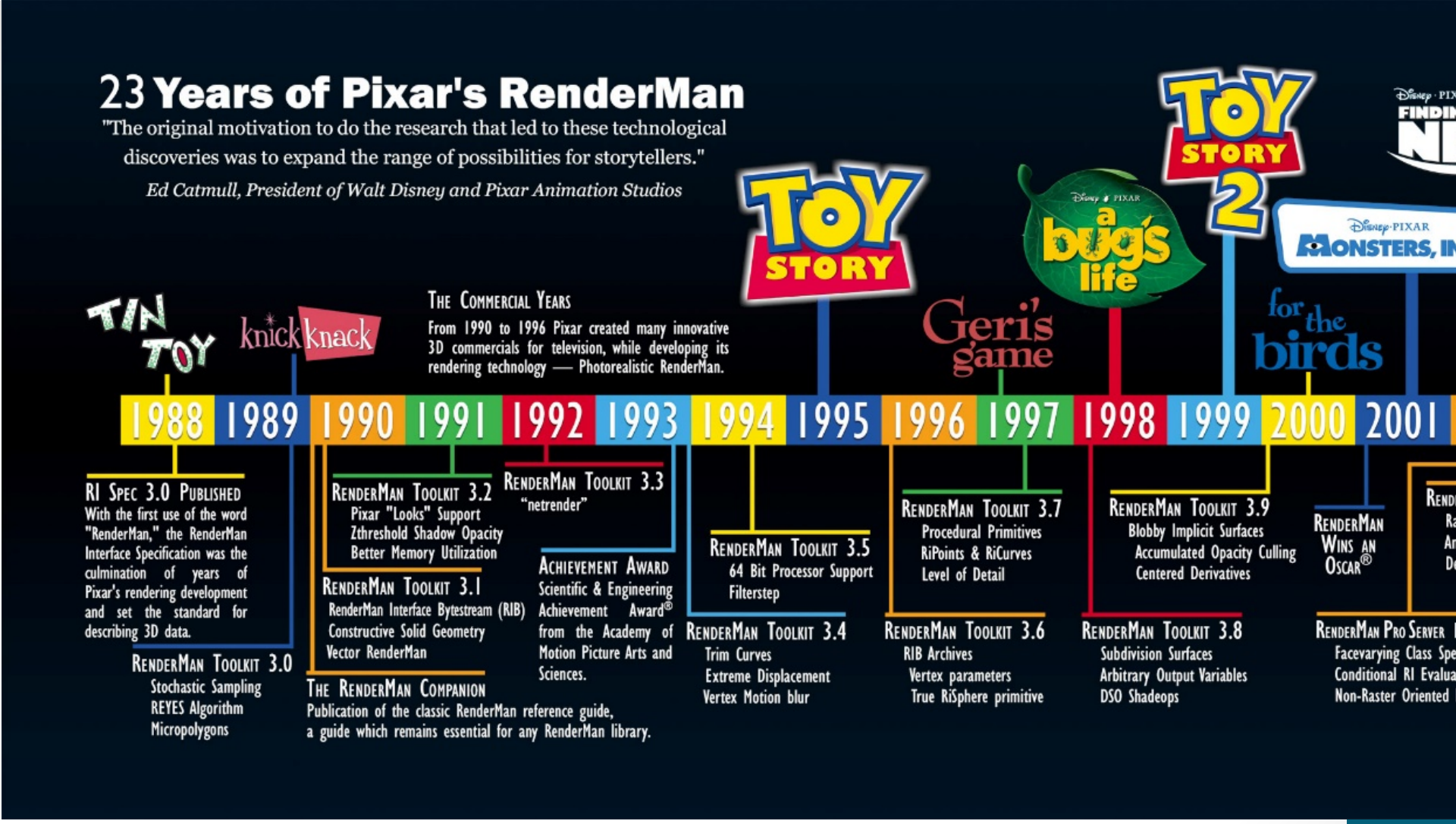
bump mapping

Computer Graphics: 1980-1990

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

SGI O2+
Silicon Graphics Int. Corp

PIXAR's RenderMan



RenderMan is both a software and an application programming interface (API) for network distributed rendering of complex and potentially ray-traced three dimensional views, employing a render farm of many client computers. The clients do not require 3D graphics cards, but may benefit from them if they are available.

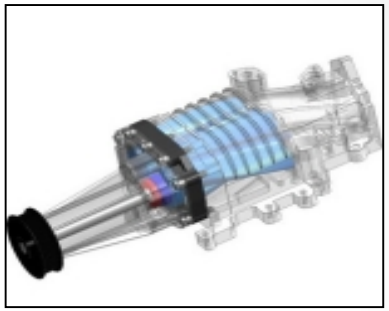
Computer Graphics: 1990-2000

- OpenGL API
  - proposed by SGI (current maintainer : Khronos Group)
- Completely computer-generated feature-length movies (Toy Story) are successful
- New hardware capabilities
  - Texture mapping
  - Blending
  - Accumulation, stencil buffers

Computer Graphics: 2000-

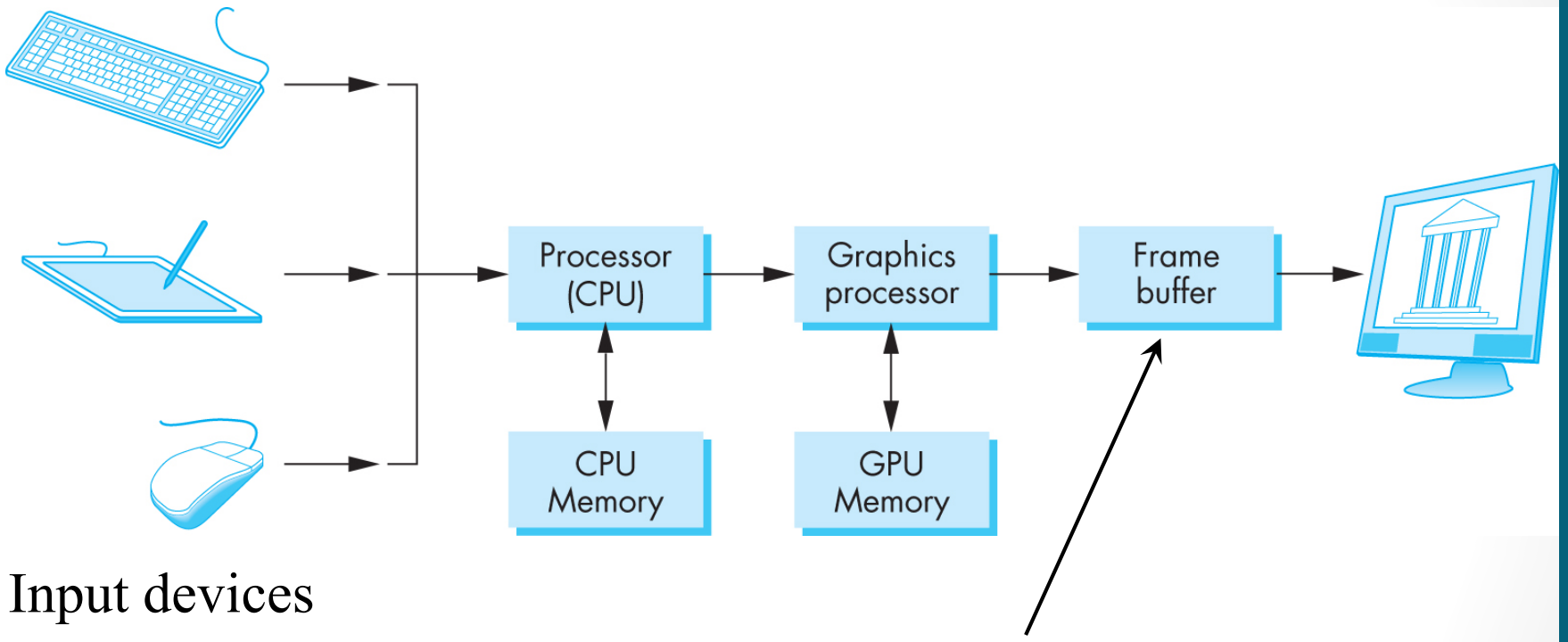
- Photorealism
- Graphics cards for PCs dominate market
  - NVidia, ATI -> AMD
- Game boxes and game players determine direction of market
- Computer graphics routine in movie industry: Maya, Lightwave
- Programmable pipelines

Applications of Computer Graphics



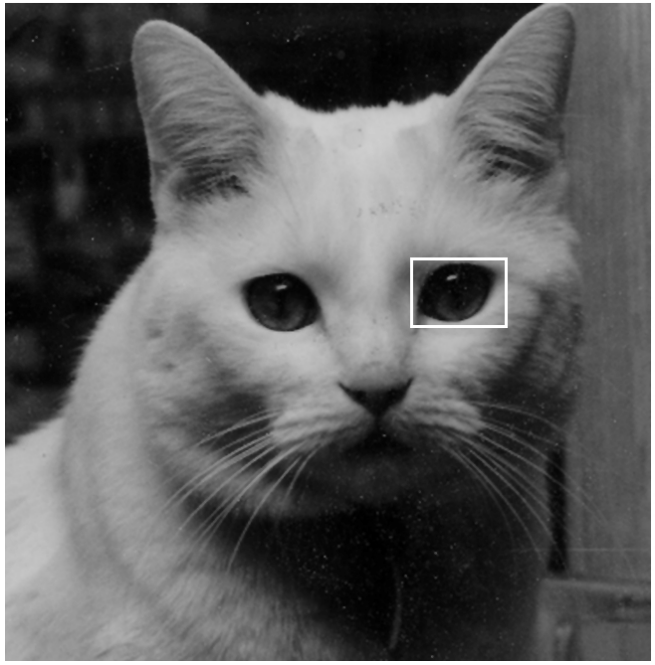
- Display of information
  - Google Map
- Design
  - CAD Software
- Simulation and animation
  - Microsoft Flight Simulator of Flight
  - Virtual reality - Wikipedia, the free encyclopedia
- User interface
  - History of graphical user interface

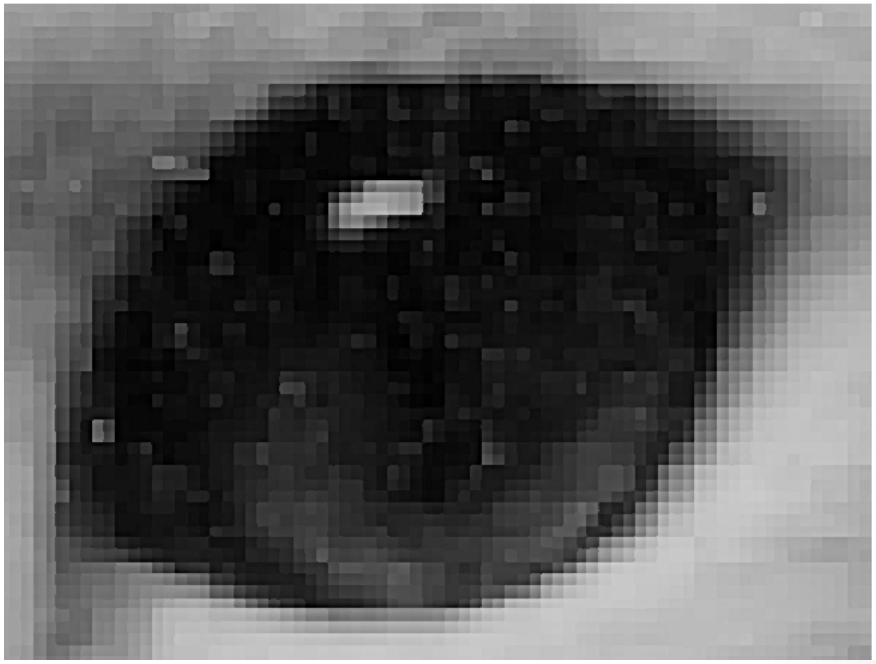
Basic Graphics System



Pixels and Frame Buffer

- **Pixels (raster)** – picture elements





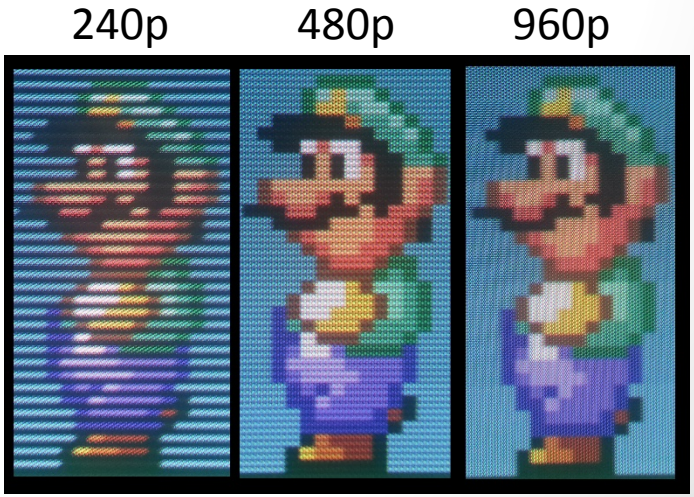
Pixels and Frame Buffer

- Frame buffer
  - Pixels are stored in a part of memory called the frame buffer
- (Color) Depth
  - Number of bits that are used for each pixel
- Resolution
  - Number of pixels in the frame buffer
- Rasterization
  - Converting of geometric entities to pixel assignments in the frame buffer

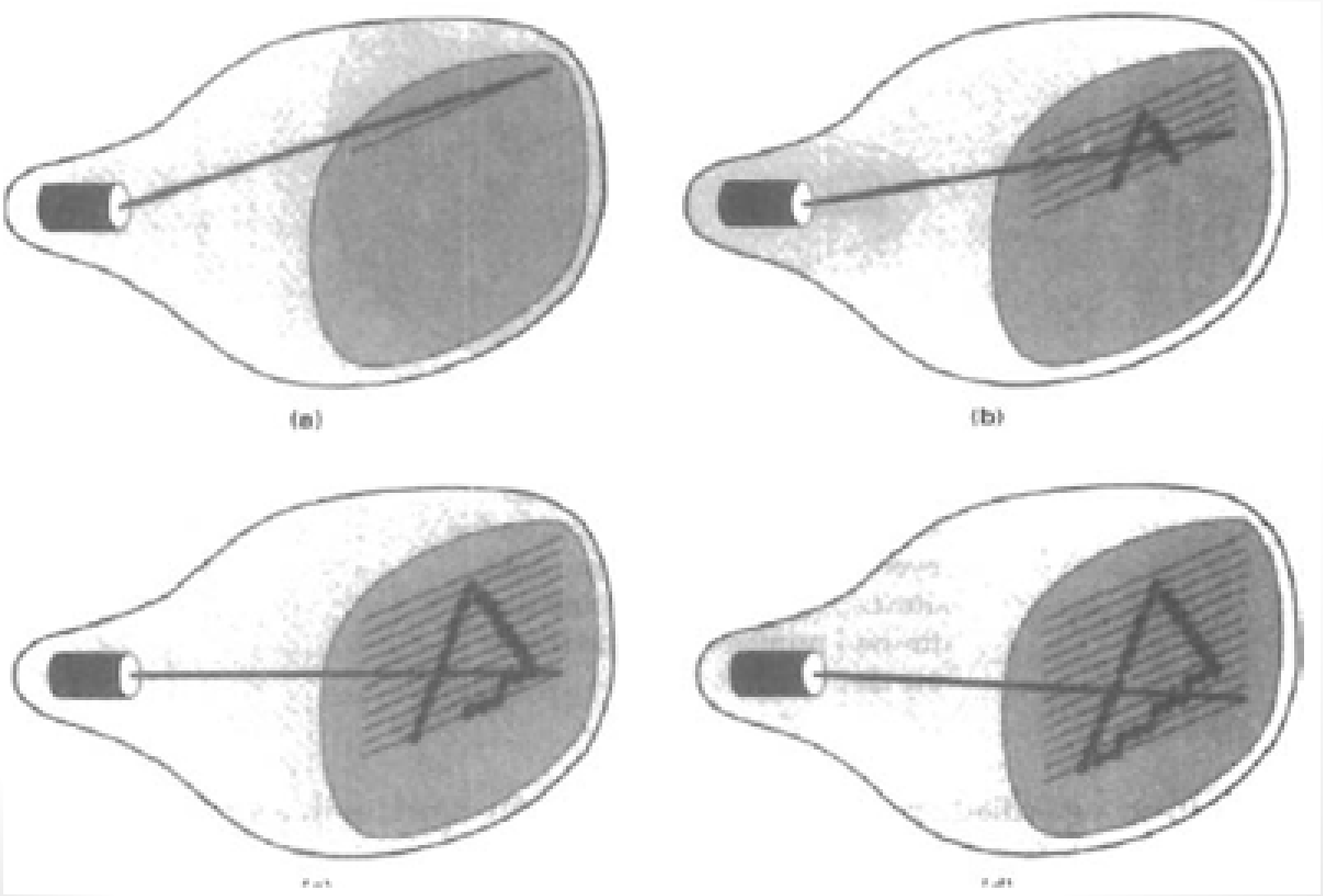
Output Devices

- In a Raster system
  - The graphics system takes pixels from the frame buffer and displays them as points on the surface of the display
  - Display rate should high enough to avoid flicker (refresh rate)
  - Two fundamental ways that pixels are displayed
    - Interlaced
    - Non-interlaced (= Progressive)

480i ? 1080p ??



Interlace Scanning



Output Devices

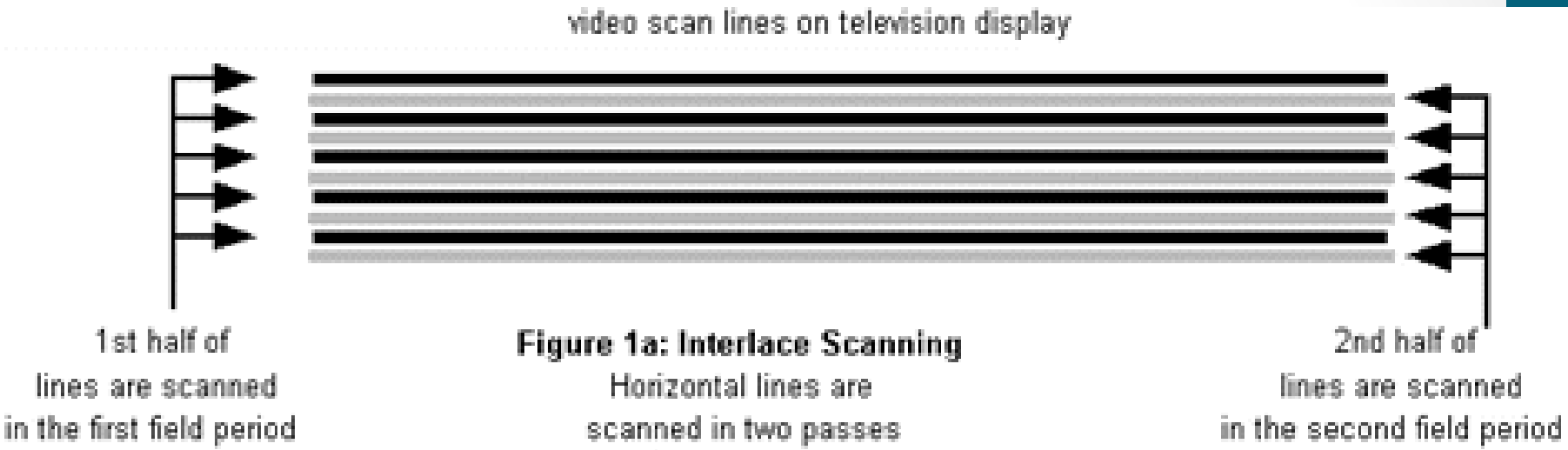


Figure 1a: Interlace Scanning
Horizontal lines are
scanned in two passes

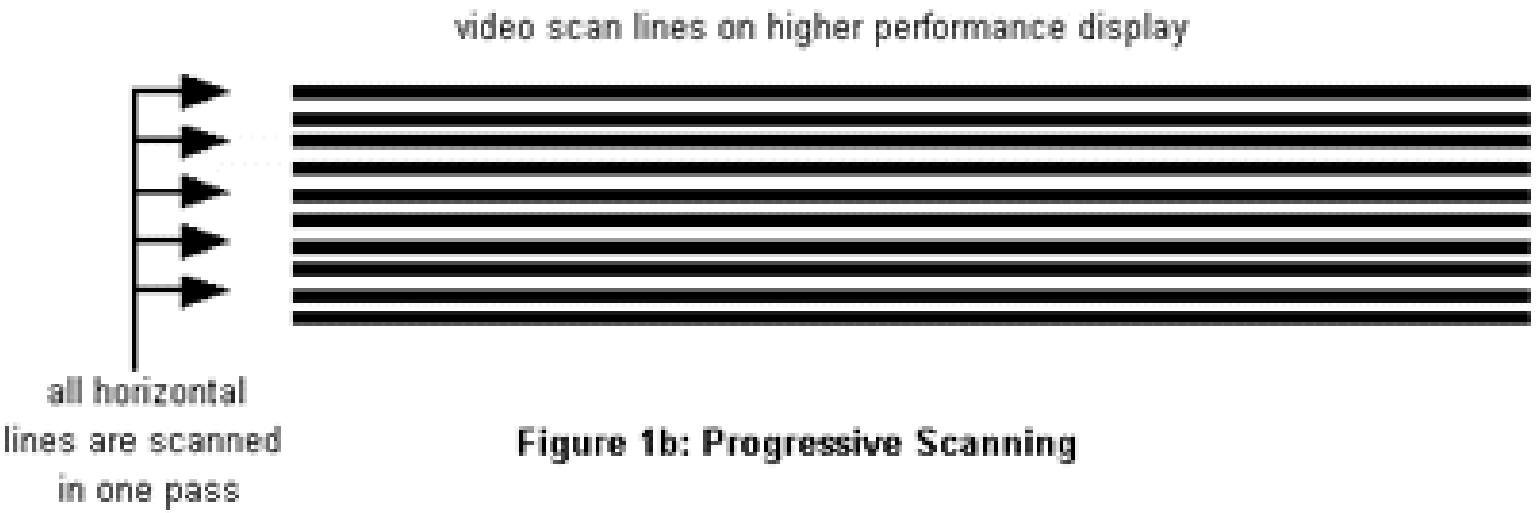
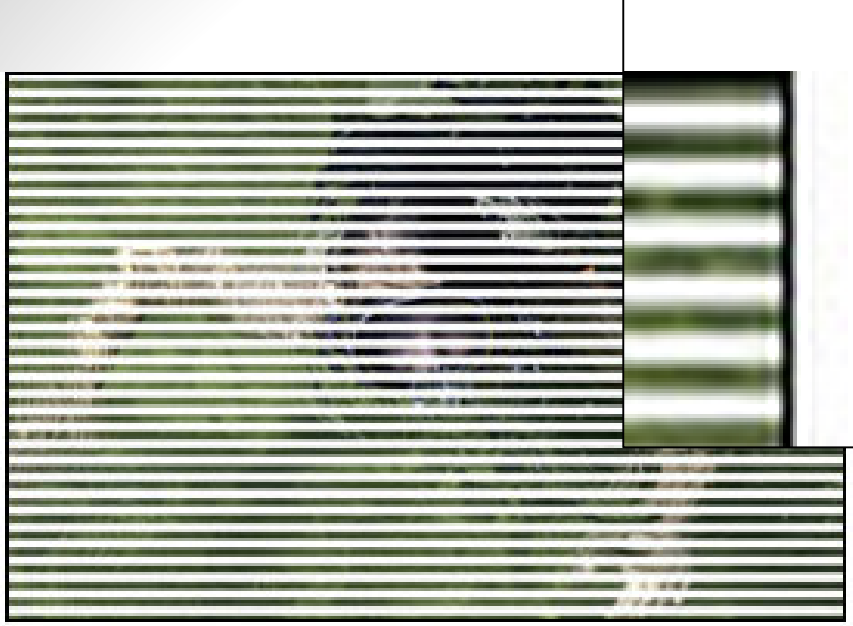
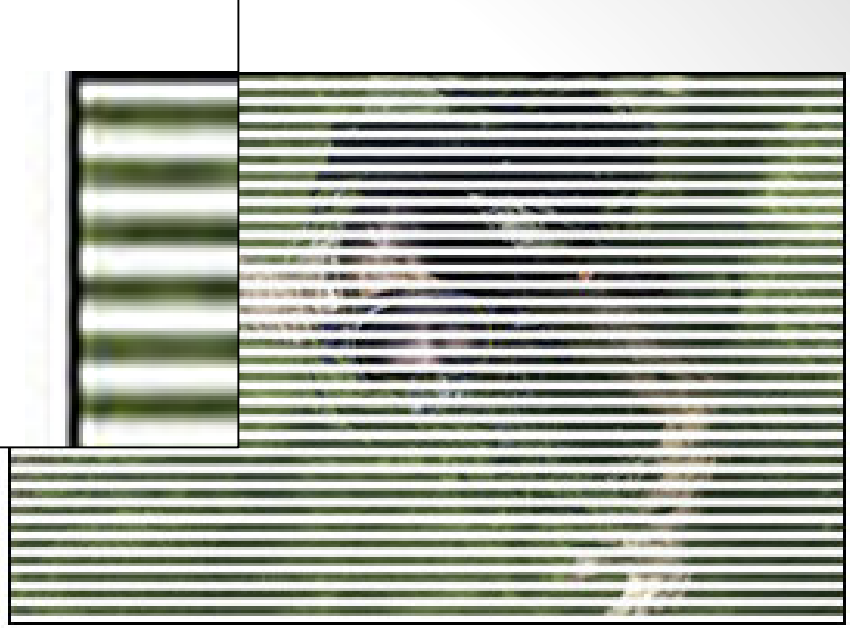


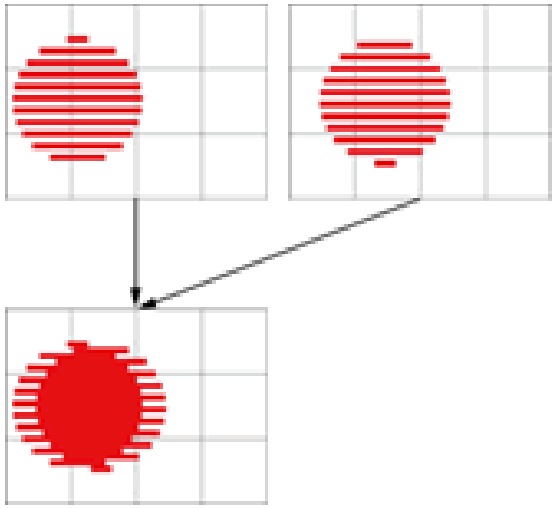
Figure 1b: Progressive Scanning



Lower field (even lines)
1/60 sec

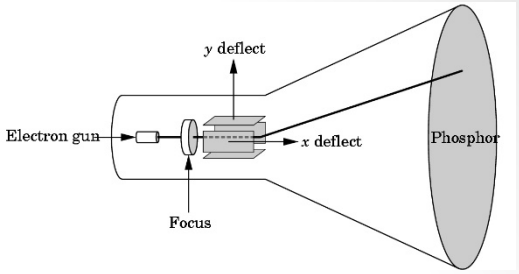


Upper field (odd lines)
1/60 sec

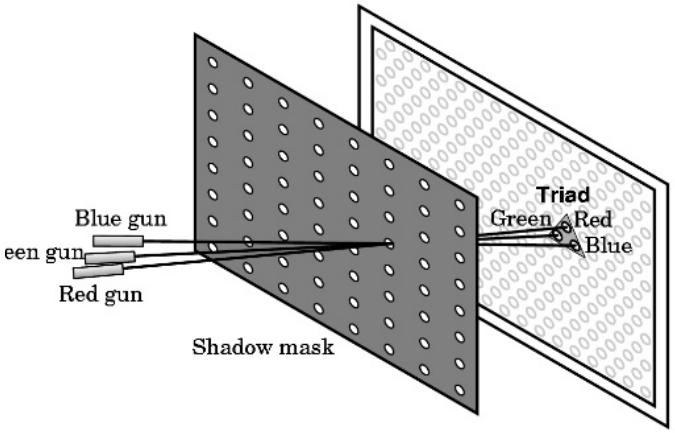


Upper + lower fields = full frame
1/30 sec

Output Devices



- CRT - can be used either as a line-drawing device (calligraphic) or to display contents of frame buffer (raster mode)
- LCD & Plasma



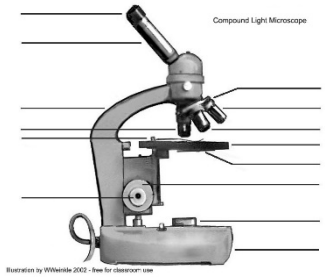
LCD

Input Devices

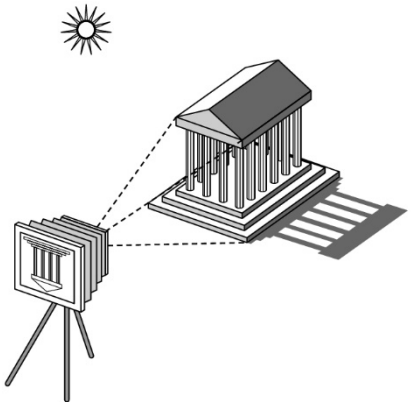
- Keyboard and Pointing devices
  - Indicate a particular location on the display
  - Mouse, 3D Mouse, joystick, data tablet

Images: Physical and Synthetic

- In computer graphics, we form images which are generally two dimensional using a process analogous to how images are formed by physical imaging systems
  - Cameras
  - Microscopes
  - Telescopes
  - Human visual system

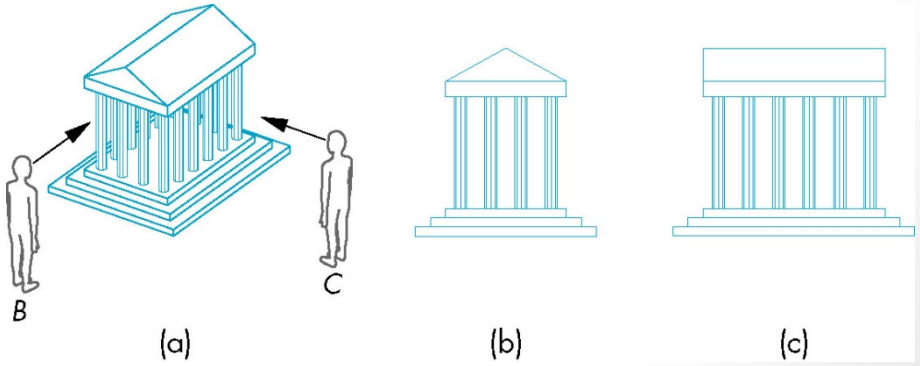


Elements of Image Formation



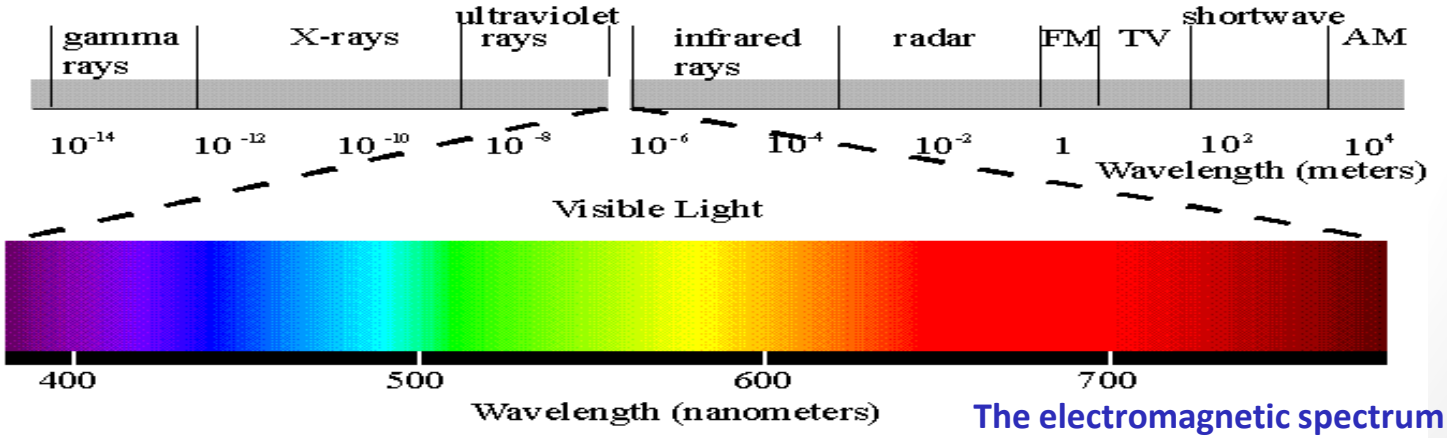
- Objects
- Viewer
- Light source(s)
- Attributes that govern how light interacts with the materials in the scene
- Note the independence of the objects, viewer, and light source(s)

Image seen by two different viewers



Light and Images

- Light is the part of the electromagnetic spectrum that causes a reaction in our visual systems
- Generally these are wavelengths in the range of about 350-780 nm (nanometers)
- Long wavelengths appear as reds and short wavelengths as blues

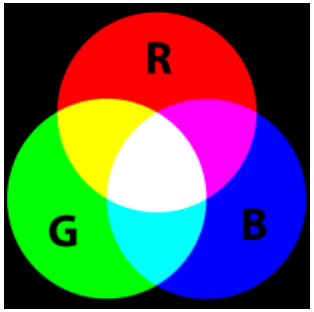


The electromagnetic spectrum

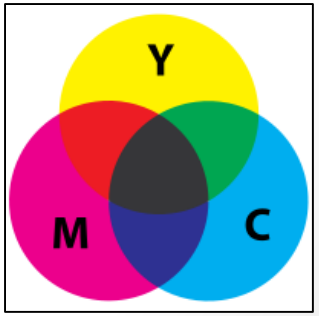
Table of Contents Visual Stimulus

Additive & Subtractive Colors

- Additive color
  - Form a color by adding amounts of three primaries
    - CRTs, projection systems, positive film
  - Primaries are Red(R),Green(G),Blue(B)



- Subtractive color
  - Form a color by filtering white light with cyan (C), Magenta (M), and Yellow (Y) filters
    - Light-material interactions
    - Printing
    - Negative film



Luminance and Color Images

- Luminance
  - Monochromatic
  - Values are gray levels
  - Analogous to working with black and white film or television
- Color
  - Has perceptional attributes of hue, saturation, and lightness
  - Do we have to match every frequency in visible spectrum? No!

Global vs Local Lighting

- Cannot compute color or shade of each object independently
  - Some objects are blocked from light
  - Light can reflect from object to object
  - Some objects might be translucent

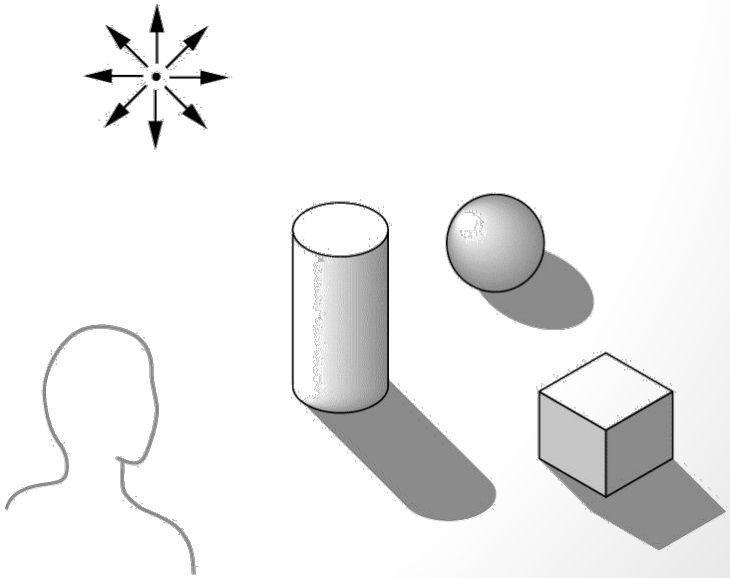
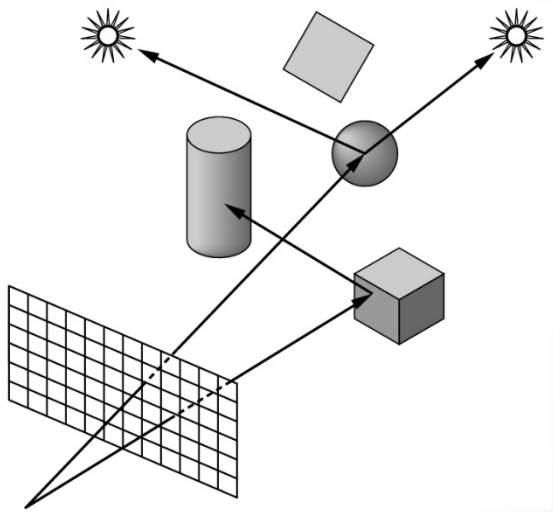


Image Formation Revisited

- Can we mimic the synthetic camera model to design graphics hardware software?
- Application Programmer Interface (API)
  - Need only specify
    - Objects
    - Materials
    - Viewer
    - Lights
- But how is the API implemented?

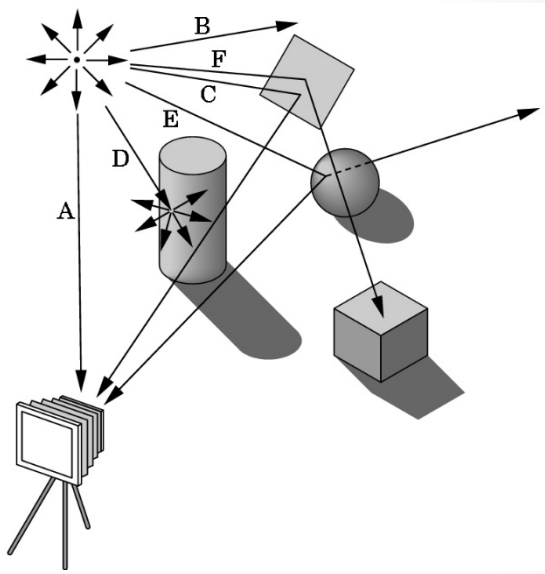
Physical Approaches

- Ray tracing: follow rays of light from center of projection until they either are absorbed by objects or go off to infinity
  - Can handle global effects
    - Multiple reflections
    - Translucent objects
  - Slow
  - Need whole data base
- Radiosity: Energy based approach
  - Very slow



Ray Tracing and Geometric Optics

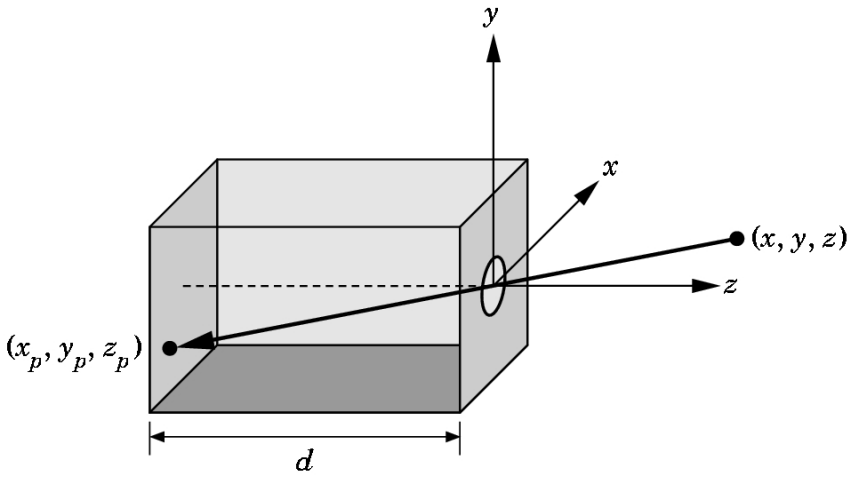
- A ray is a semi-infinite line that emanates from a point and travels to infinity in a particular direction
- One way to form an image is to follow rays of light from a point source determine which rays enter the lens of the camera.
- However, each ray of light may have multiple interactions with objects before being absorbed or going to infinity.

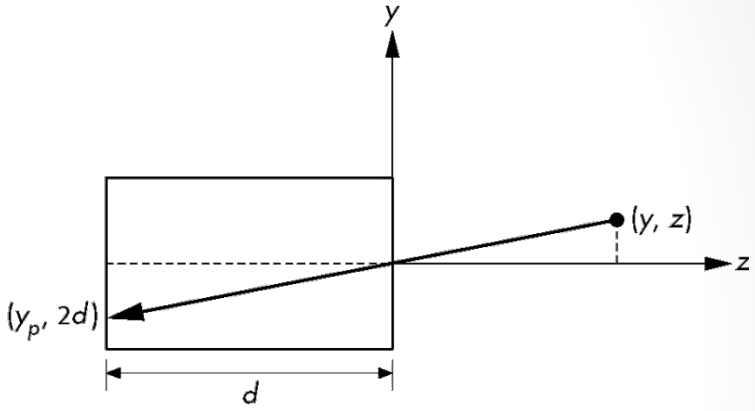


Why not Ray Tracing?

- Ray tracing seems more physically based so why don't we use it to design a graphics system?
- Possible and is actually simple for simple objects such as polygons and quadrics with simple point sources
- In principle, can produce global lighting effects such as shadows and multiple reflections but is slow and not well-suited for interactive applications

The Imaging Systems
(The Pinhole Camera)





Use trigonometry to find projection of a point

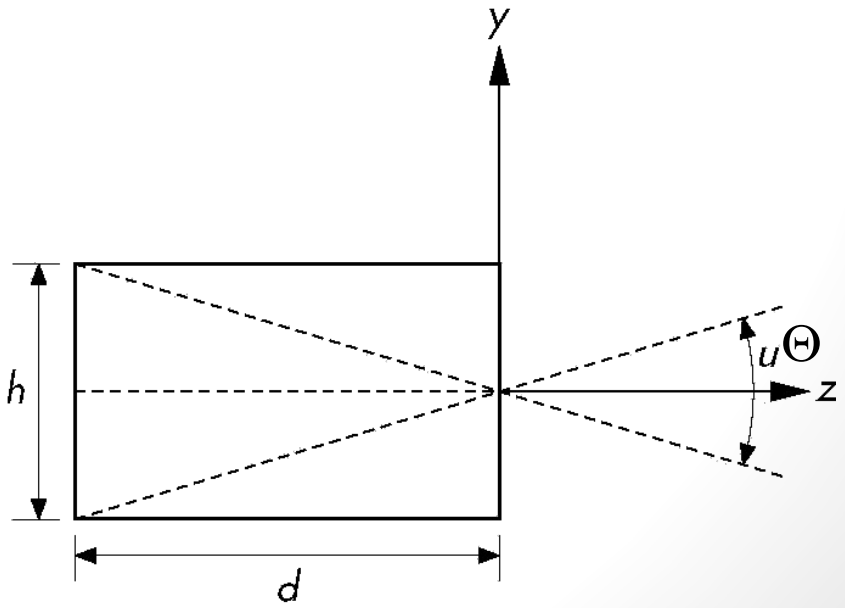
$$x_p = -x*(d/z) \qquad y_p = -y*(d/z) \qquad z_p = -d$$

These are equations of simple perspective

The Field (Angle of View)

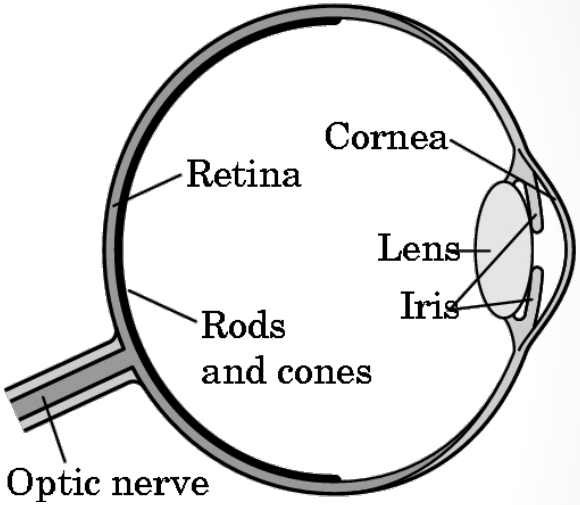
- The field of our camera is the angle made by the largest object that our camera can image on its film plane
- The ideal pinhole camera has an infinite depth of view: every point within its field is in focus

$$\Theta = 2\tan^{-1}\frac{h}{2d}$$



The Human Visual System Trichromatic Theory

- Human visual system has two types of sensors
  - Rods: monochromatic, night vision
  - Cones
    - Color sensitive
    - Three types of cone
    - Only three values are sent to the brain (the tristimulus values)

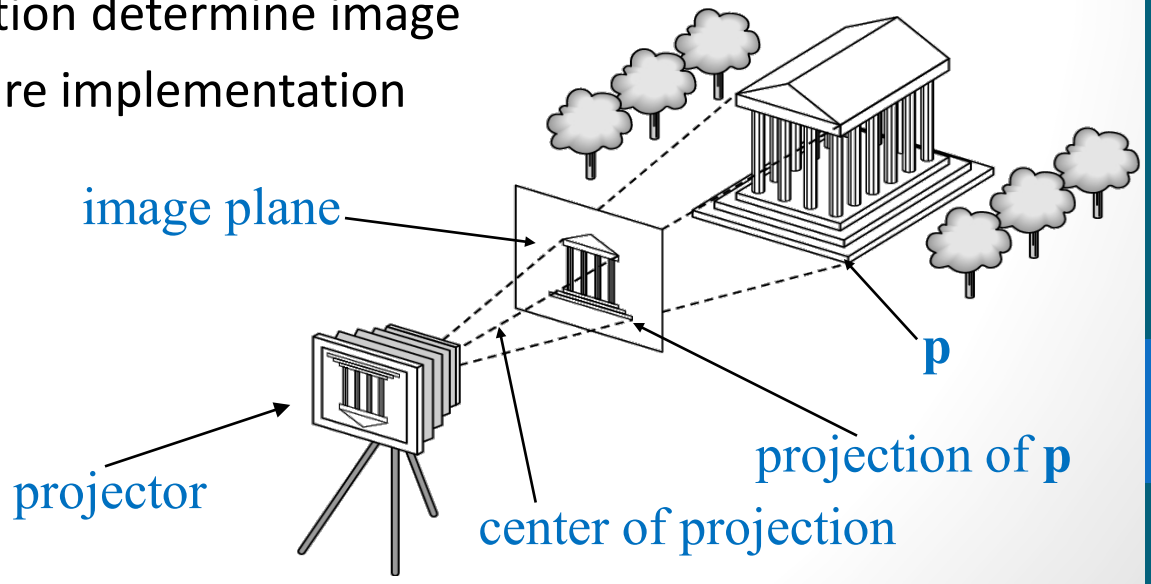


- Need only match these three values
  - Need only three primary colors

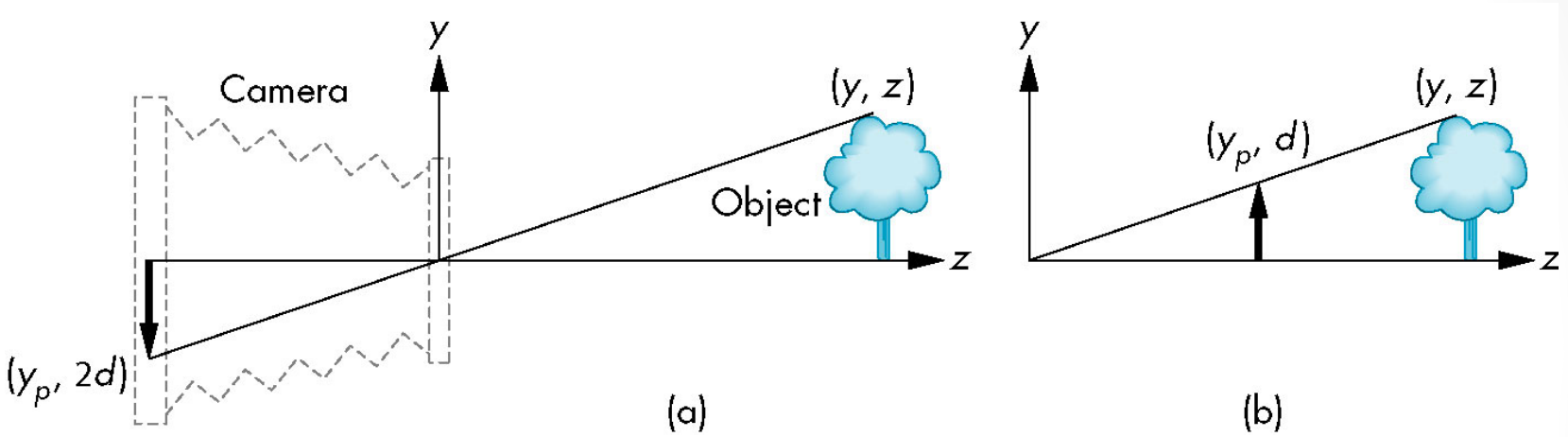
The Synthetic Camera Model

Advantages

- Separation of objects, viewer, light sources
- Two-dimensional graphics is a special case of three-dimensional graphics
- Leads to simple software API
  - Specify objects, lights, camera, attributes
  - Let implementation determine image
- Leads to fast hardware implementation
  - pipeline

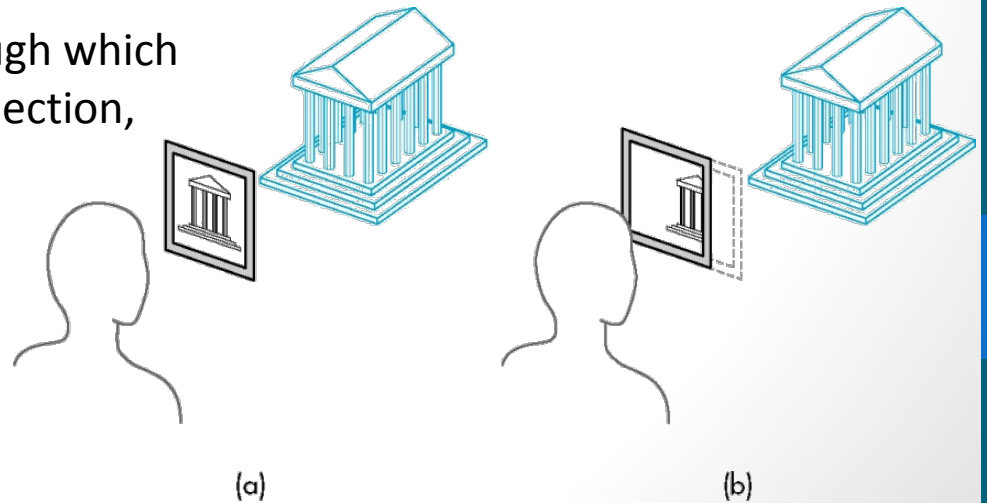


Equivalent Views of Image Formation



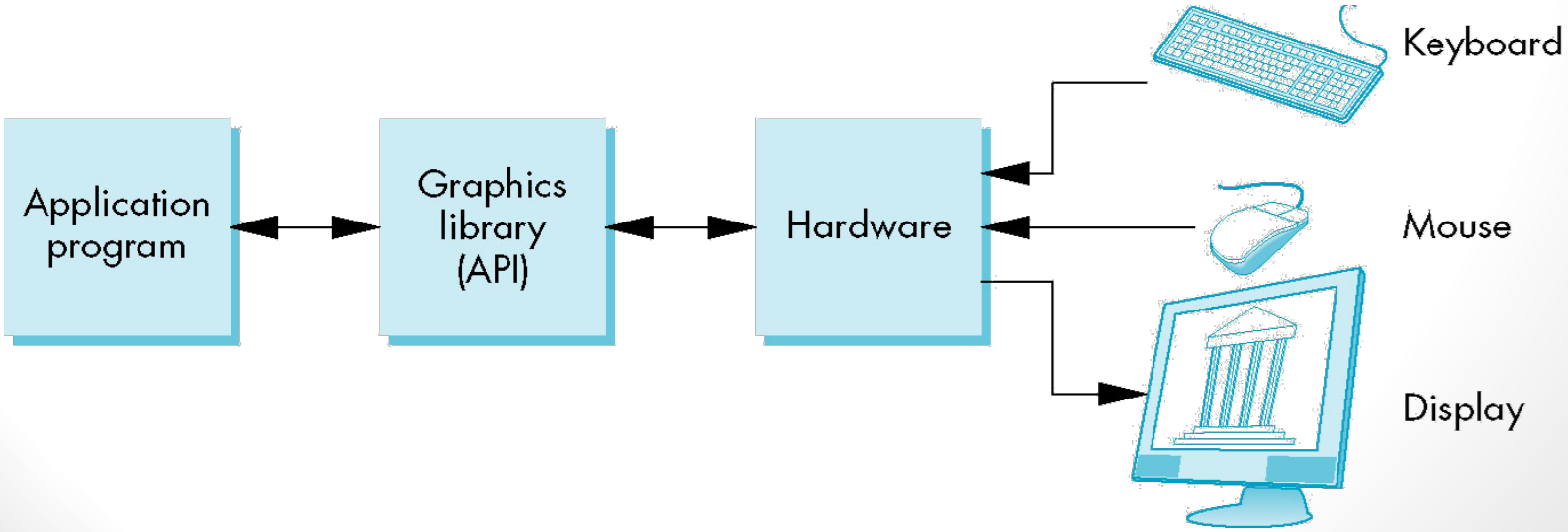
Clipping rectangle

This rectangle acts as a window through which a viewer, located at the center of projection, sees the world.



The Programmer's Interface

- Application programmer's model of software
- Programmer sees the graphics system through an interface - API
  - The interface between an application program and a graphics system can be specified through a set of functions that resides in a graphics library



API Contents

- Functions that specify what we need to form an image
  - Objects
  - Viewer
  - Light Source(s)
  - Materials
- Other information
  - Input from devices such as mouse and keyboard
  - Capabilities of system

Object Specification

- Most APIs support a limited set of primitives including
  - Points (1D object)
  - Line segments (2D objects)
  - Polygons (3D objects)
  - Some curves and surfaces
    - Quadrics
    - Parametric polynomial
- All are defined through locations in space or vertices

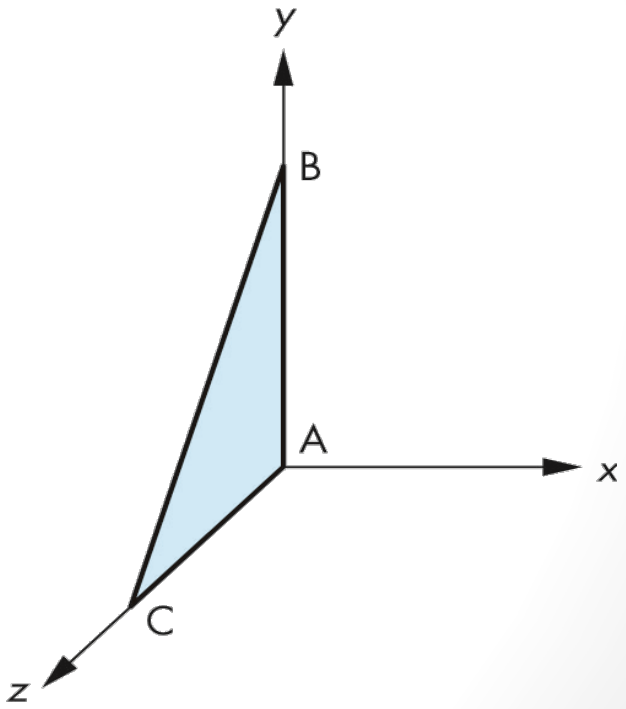
Example

type of object

location of vertex

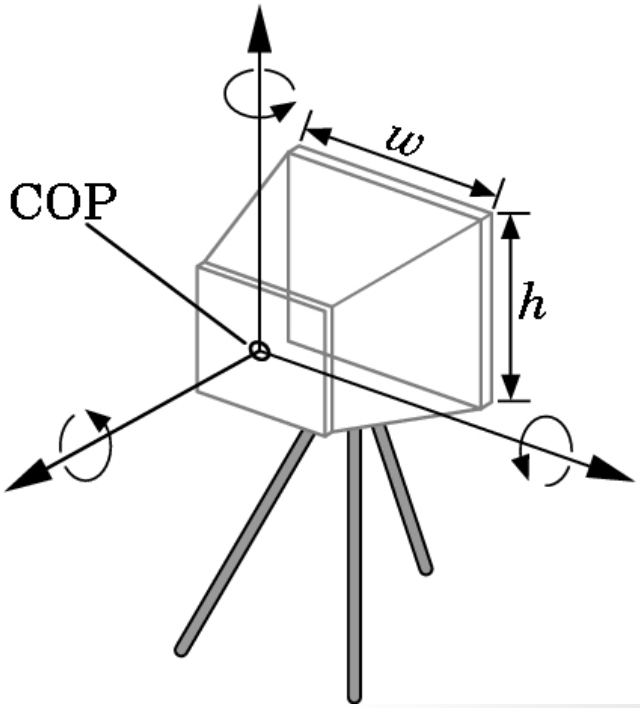
```
glBegin(GL_POLYGON);
        glVertex3f(0.0, 0.0, 0.0);
        glVertex3f(0.0, 1.0, 0.0);
        glVertex3f(0.0, 0.0, 1.0);
glEnd( );
```

end of object definition



Camera Specification

- Seven degrees of freedom (7 DOF)
  - Position of center of lens (3 DOF)
  - Orientation (3 DOF)
  - Filed-of-View (1 DOF)
    - Lens propriety
- Film size
- Orientation of film plane

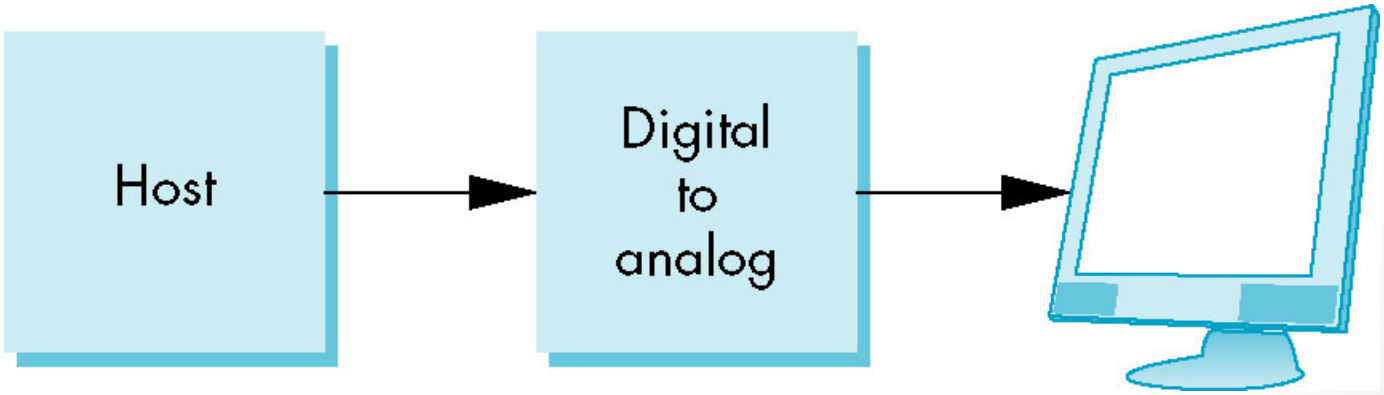


Lights and Materials

- Types of lights
  - Point sources vs distributed sources
  - Spot lights
  - Near and far sources
  - Color properties
- Material properties
  - Absorption: color properties
  - Scattering
    - Diffuse
    - Specular

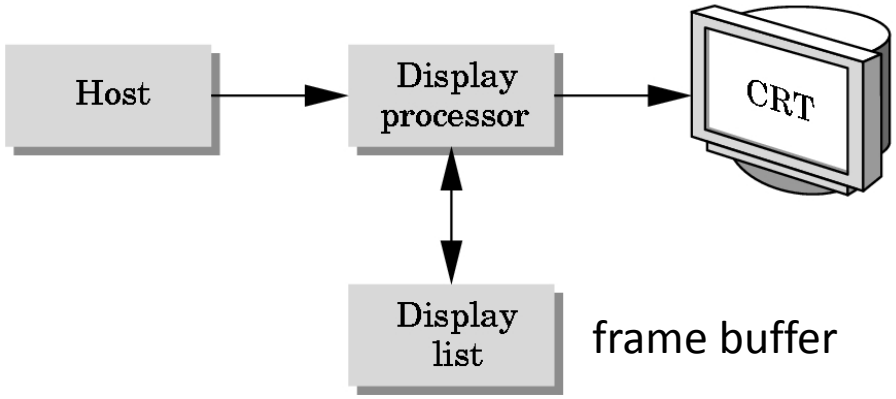
Graphics Architectures

- Early graphics system
  - General-purpose computer with von Neumann architecture – a single processor processes a single instruction at a time.



Display Processor

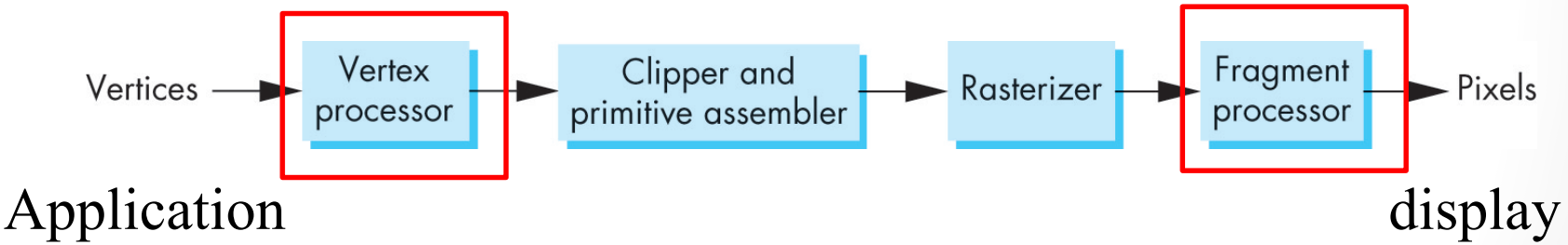
- The host computer try to refresh display use a special purpose computer called a display processor (DPU)



- Graphics stored in display list (display file) on display processor
- Host compiles display list and sends to DPU

Geometric Pipeline

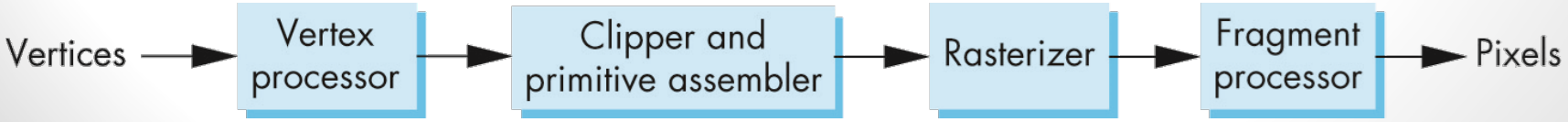
- Process objects one at a time in the order they are generated by the application
  - Can consider only **local lighting**
- Pipeline architecture



- All steps can be implemented in hardware on the graphics card

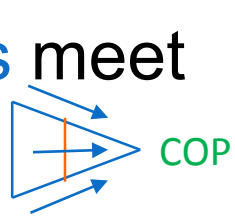
Vertex Processing

- Much of the work in the pipeline is in converting object representations **from one coordinate system to another**
  - Object coordinates
  - Camera (eye) coordinates
  - Screen coordinates
- Every change of coordinates is equivalent to a **matrix transformation**
- *Vertex processor also computes **vertex colors**

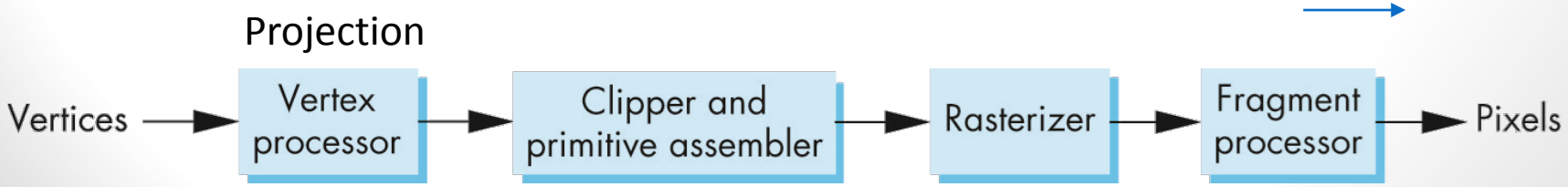


Projection

- *Projection* is the process that combines the 3D viewer with the 3D objects to produce the 2D image
  - **Perspective projections**: all projectors meet at the center of projection

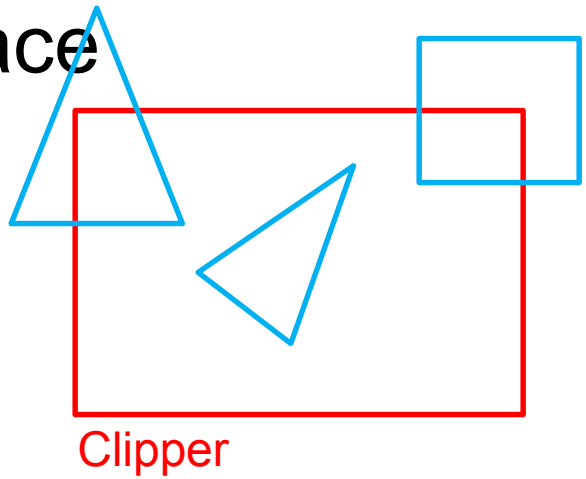


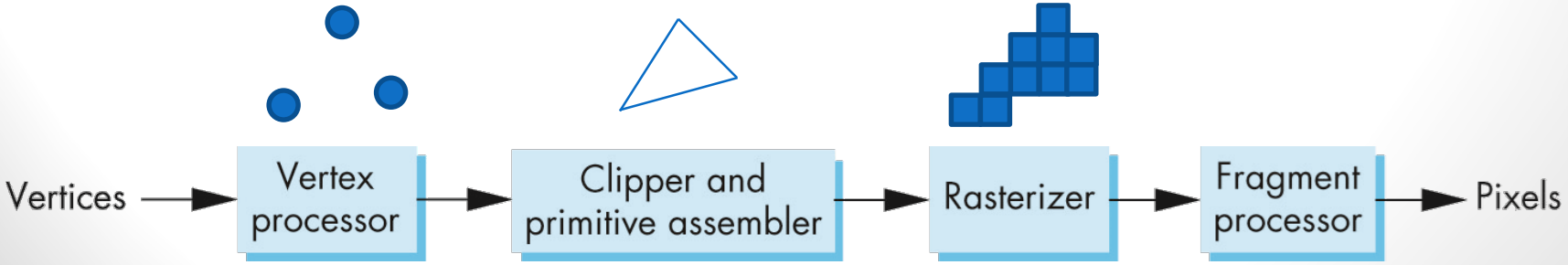
  - **Parallel projection**: all projectors are parallel, center of projection is replaced by a direction of projection



Primitive Assembly

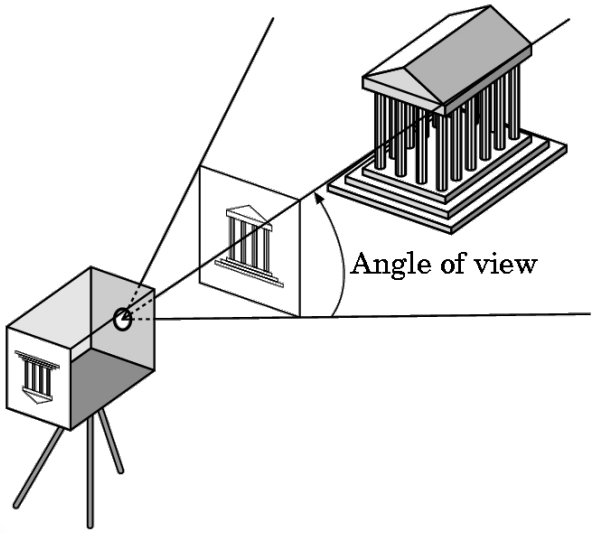
- Vertices must be collected into geometric objects/primitives before clipping and rasterization can take place
  - Line segments
  - Polygons
  - Curves and surfaces

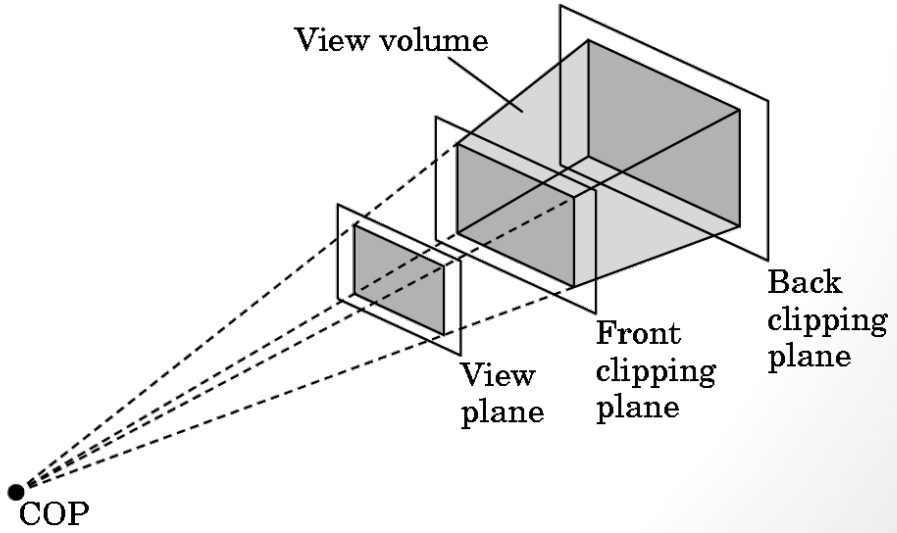




Clipping

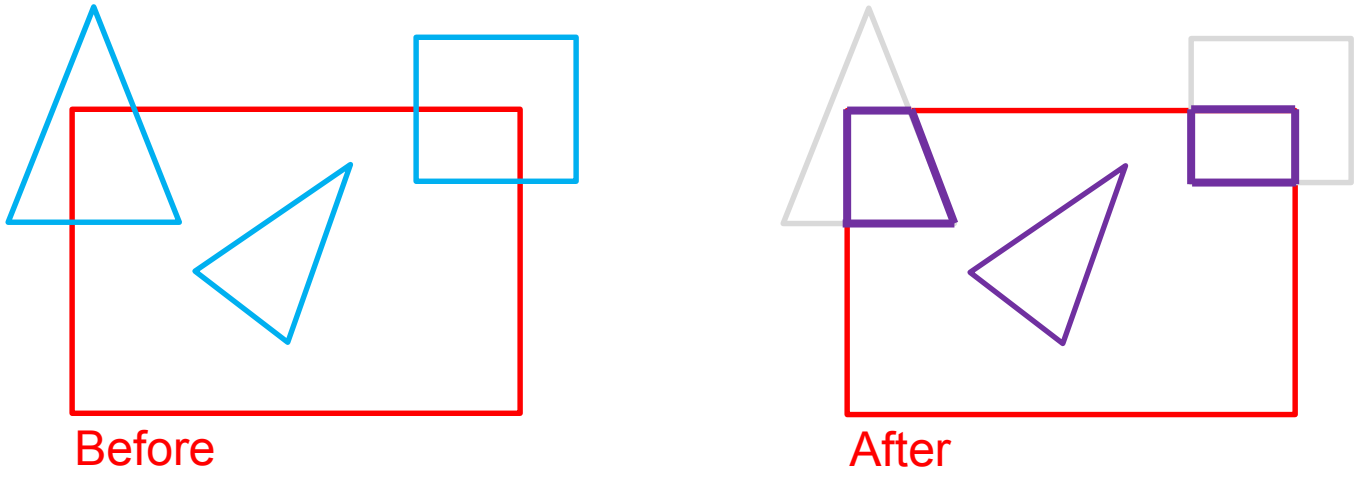
- Just as a real camera cannot “see” the whole world, the virtual camera can only see part of the world space
  - Objects that are not within this volume are said to be clipped out of the scene

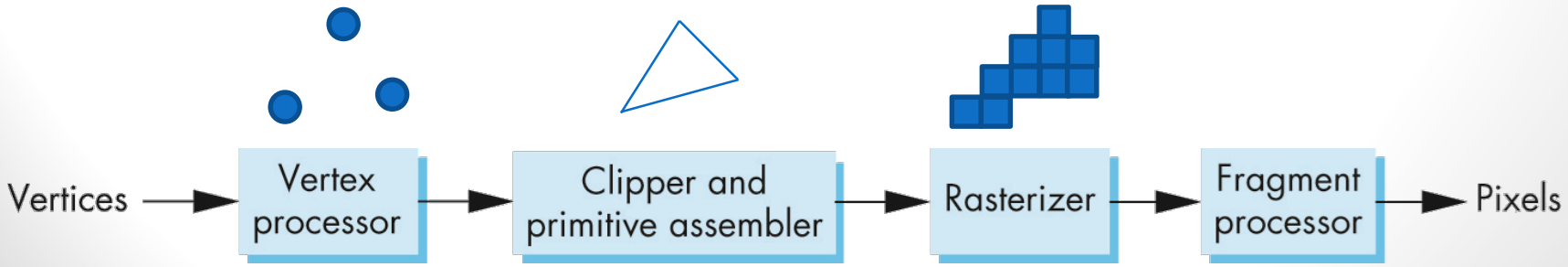




Clipping

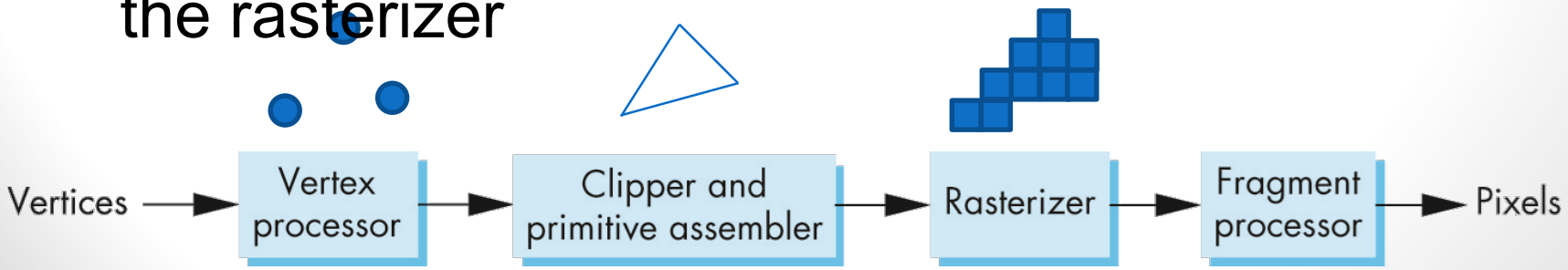
- The clipping and rasterization take place





Rasterization

- If an object is not clipped out, the appropriate pixels in the frame buffer must be assigned colors
- Rasterizer produces a set of fragments for each object
- Fragments are “potential pixels”
  - Have a location in frame buffer
  - Color and depth attributes
- Vertex attributes are interpolated over objects by the rasterizer

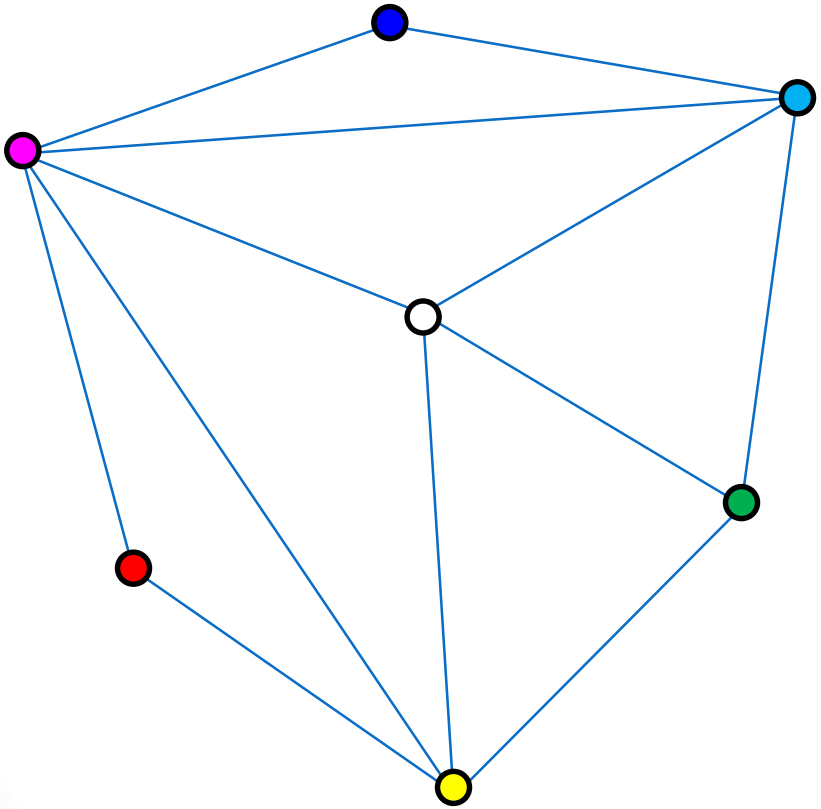


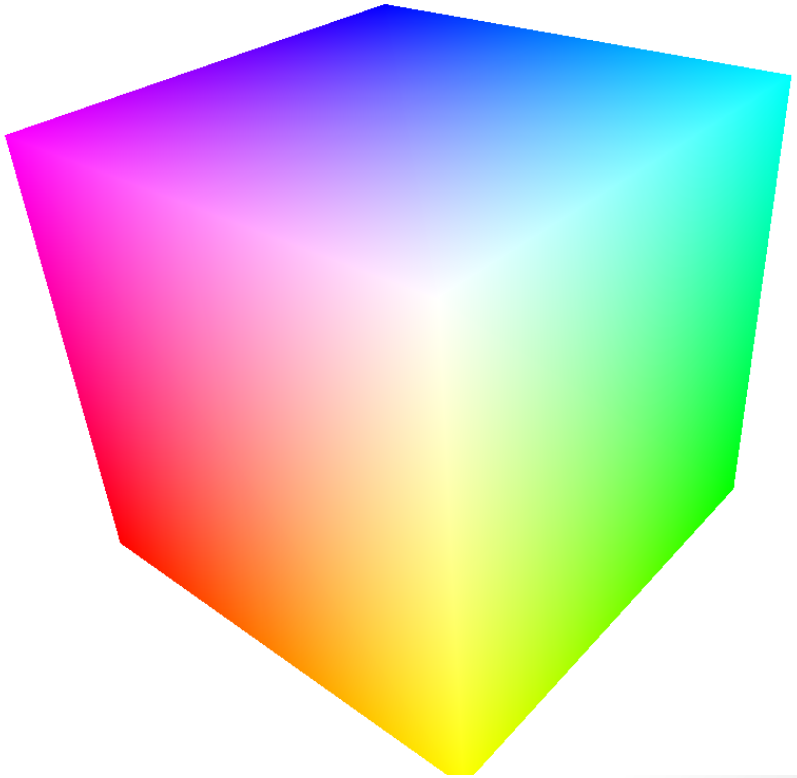
Fragment Processing

- Fragments are processed to determine the color of the corresponding pixel in the frame buffer
- Colors can be determined by texture mapping or interpolation of vertex colors
- Fragments may be blocked by other fragments closer to the camera
  - Use hidden-surface removal to determine it.
  - Visible fragments >> pixel on the viewport

become

The result





Extensive Readings

- Cornell University Program of Computer Graphics - What is Computer Graphics
- Wikipedia - 3D computer graphics
- A Brief History of Computer Graphics
- CGI Historical Timeline

Video Links

- SIGGRAPH 2015 Technical Papers Video Preview
  - Real Time Live shows
  - 2014, 2013
- Ivan Sutherland's Sketchpad
- History of Video Games (1972-2007) *NEW
- John Whitney "Catalog" 1961
- Frostbite 2 Engine - Real-Time Radiosity Tech Demo Video (SIGGRAPH 2010)

SUGGESTION!

OR

OBJECTION?

Let's stop here,

TAKE A BREAK