Computer Graphics

Bing-Yu Chen
National Taiwan University

Introduction

- What is Computer Graphics ?
- The Graphics Process
- Color Models
- Triangle Meshes
- The Rendering Pipeline

What is Computer Graphics ?

- Definition
  - the pictorial *synthesis* of real or imaginary objects from their computer-based models

INPUT \ OUTPUT	descriptions	images
descriptions		Computer Graphics
images	Computer Vision Pattern Recognition	Image Processing

What is Computer Graphics ?

- *Computer Graphics* deals with all aspects of creating images with a computer
  - hardware
  - software
  - applications

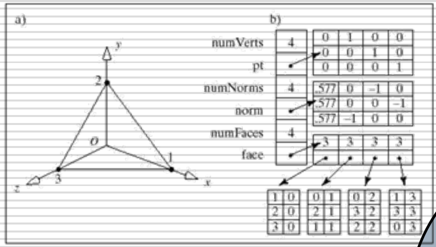
Example

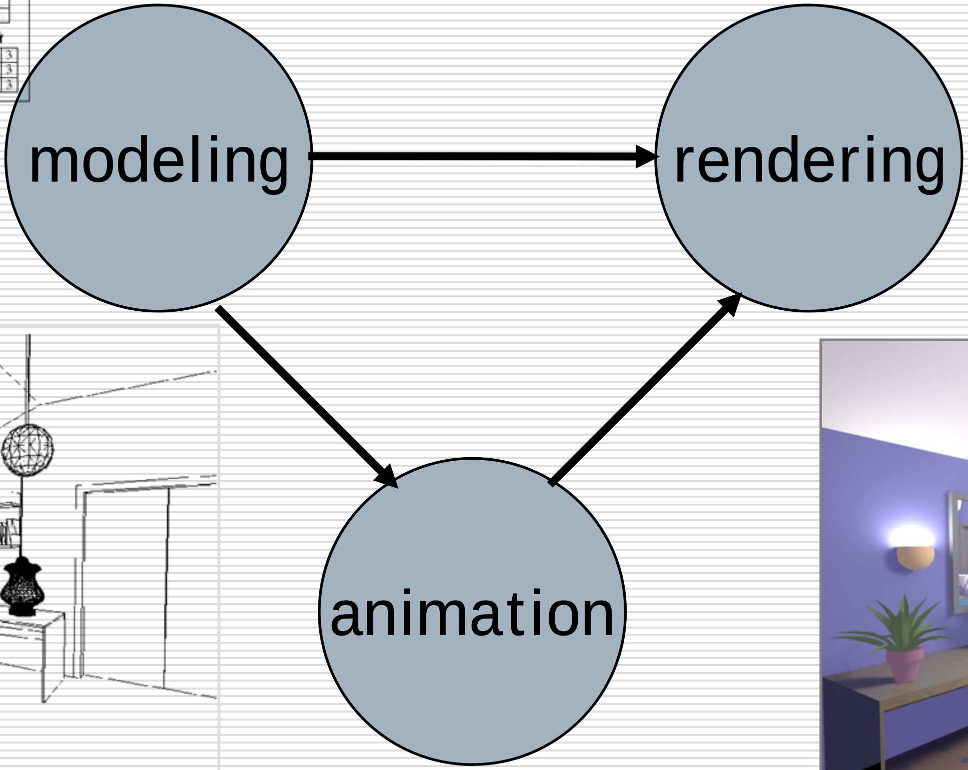

POV
is
great!
DELICIOUS
PRIMITIVES
POV
FLAKES
The Kitchen - Jaime Vives Piqueres - POVCOMP 2004

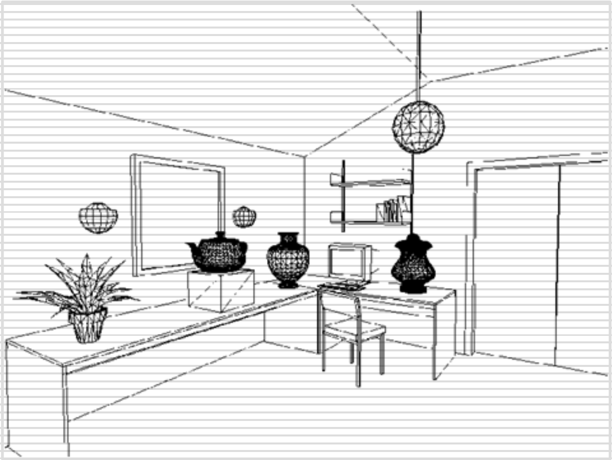
What is Computer Graphics ?

- 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

What is Computer Graphics ?







Applications

- Movies
- Interactive entertainment
- Industrial design
- Architecture
- Culture heritage

Animation Production Pipeline


story

text treatment

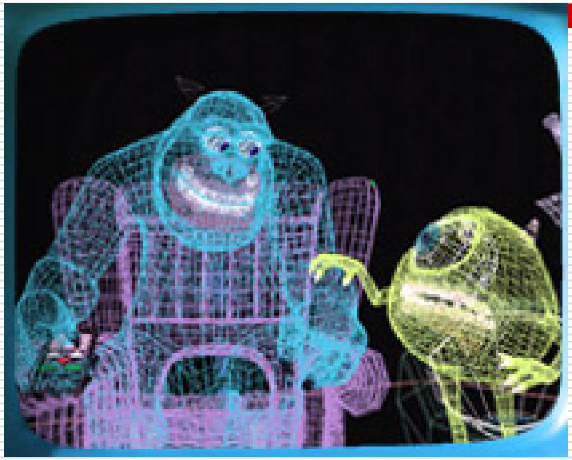

storyboard


voice


storyreal


look and feel

Animation Production Pipeline



modeling/articulation

layout

animation

shading/lighting

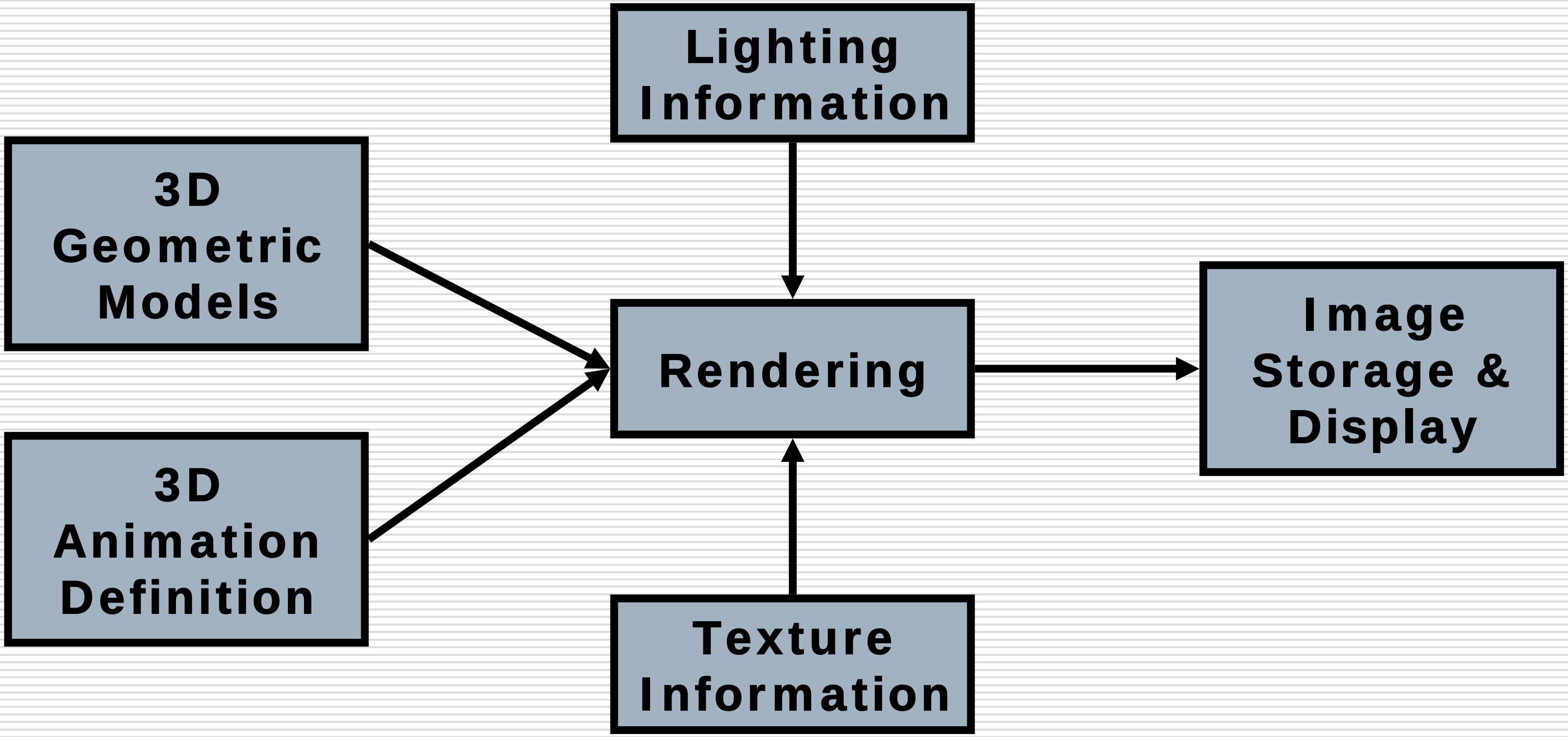
rendering

final touch

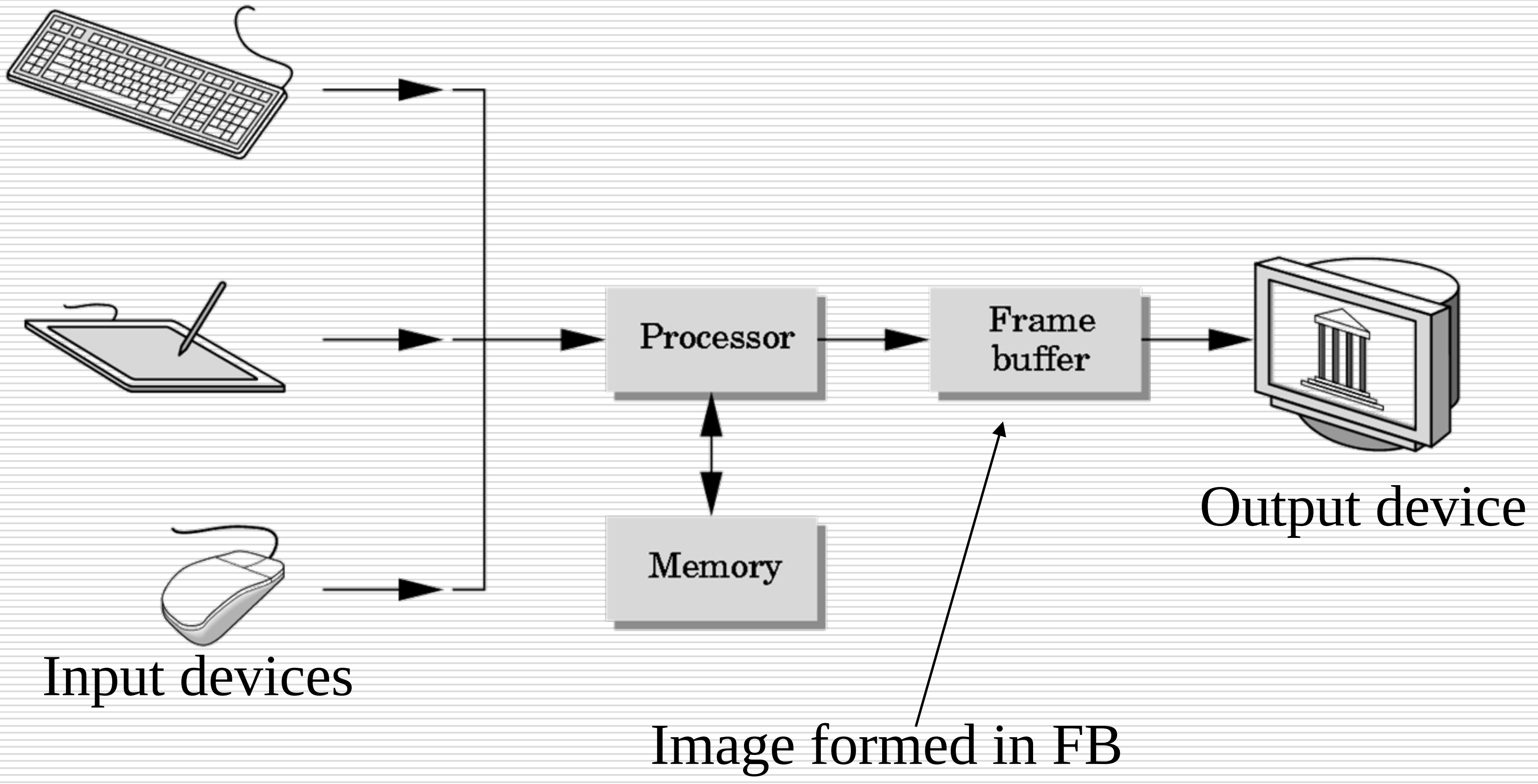
The Advantages of Interactive Graphics

- One of the most natural means of communicating with a computer
- A picture is worth then thousand words
- A *moving* picture is worth than thousand *static* ones
  - movie, motion dynamics
- Graphical User Interface

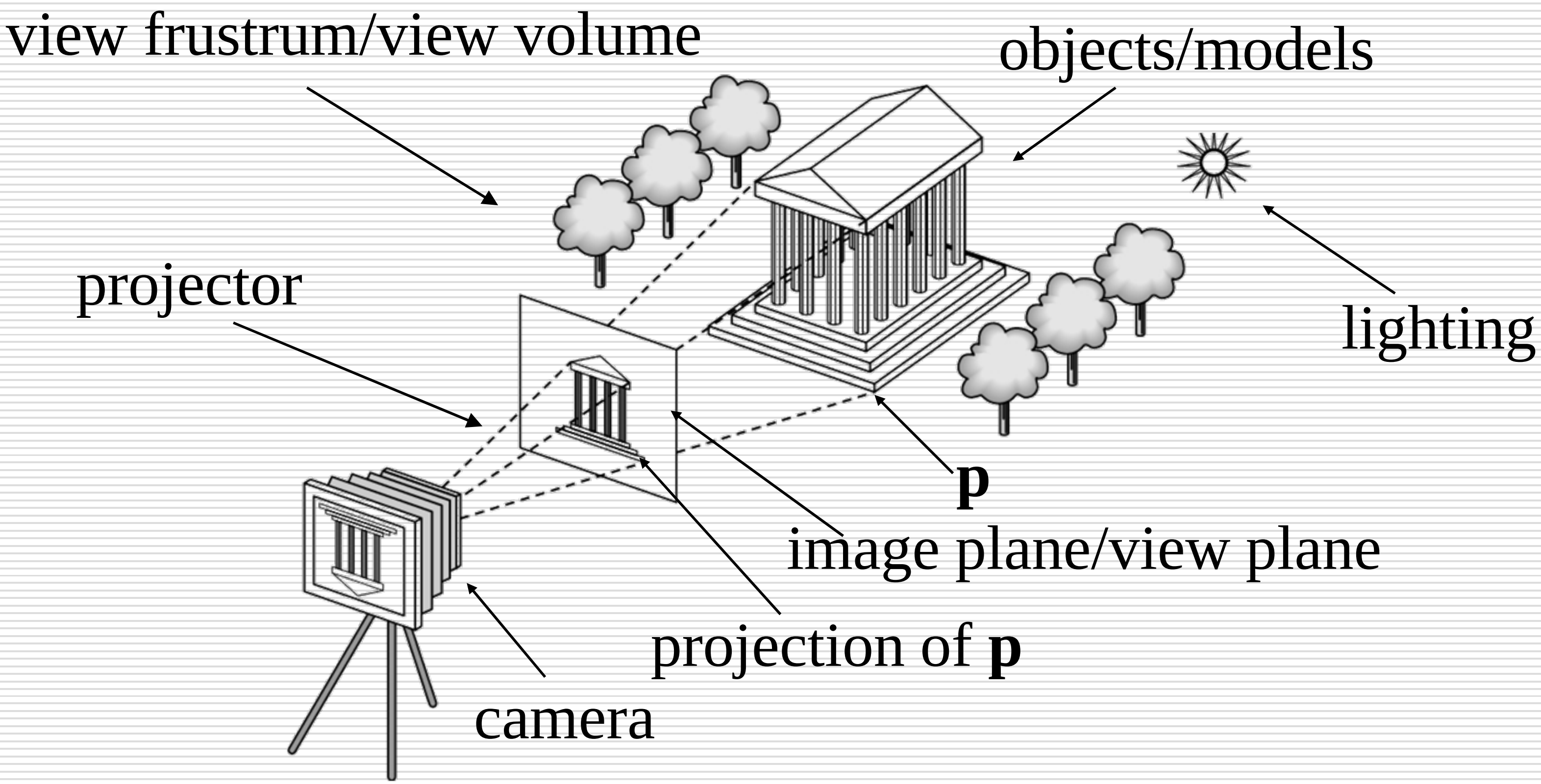
The Graphics Process



Basic Graphics System

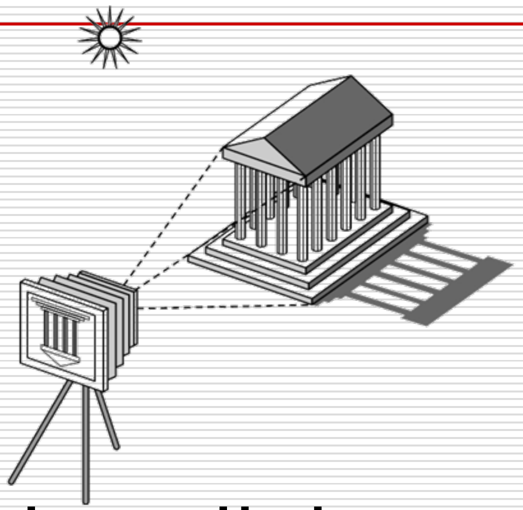


Synthetic Camera Model



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)

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

Light

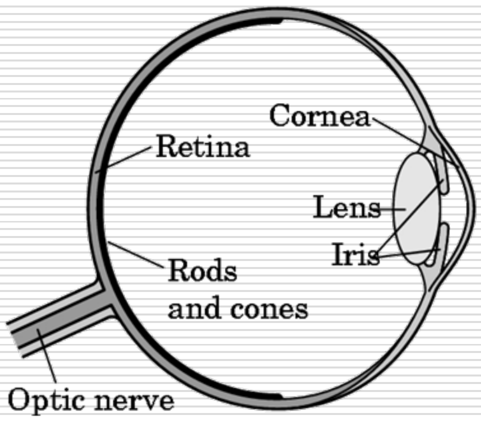
- *Light* is the part of the electromagnetic spectrum that causes a reaction in our visual systems
- Generally these are wavelengths in the range of about 350-750 nm (nanometers)
- Long wavelengths appear as reds and short wavelengths as blues

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!

Three-Color Theory

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

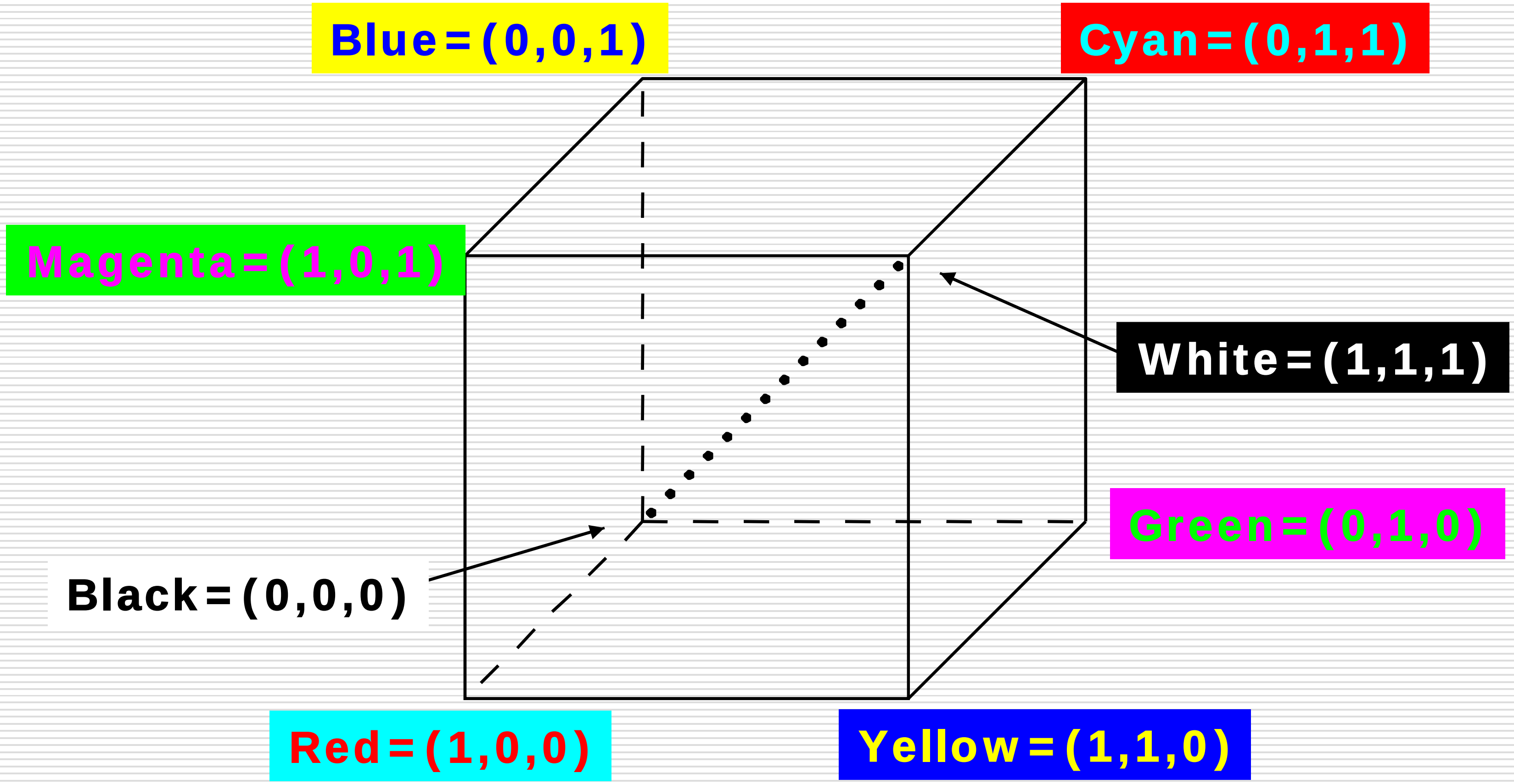


- Need only match these three values
  - Need only three *primary* colors

Additive and Subtractive Color

- 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

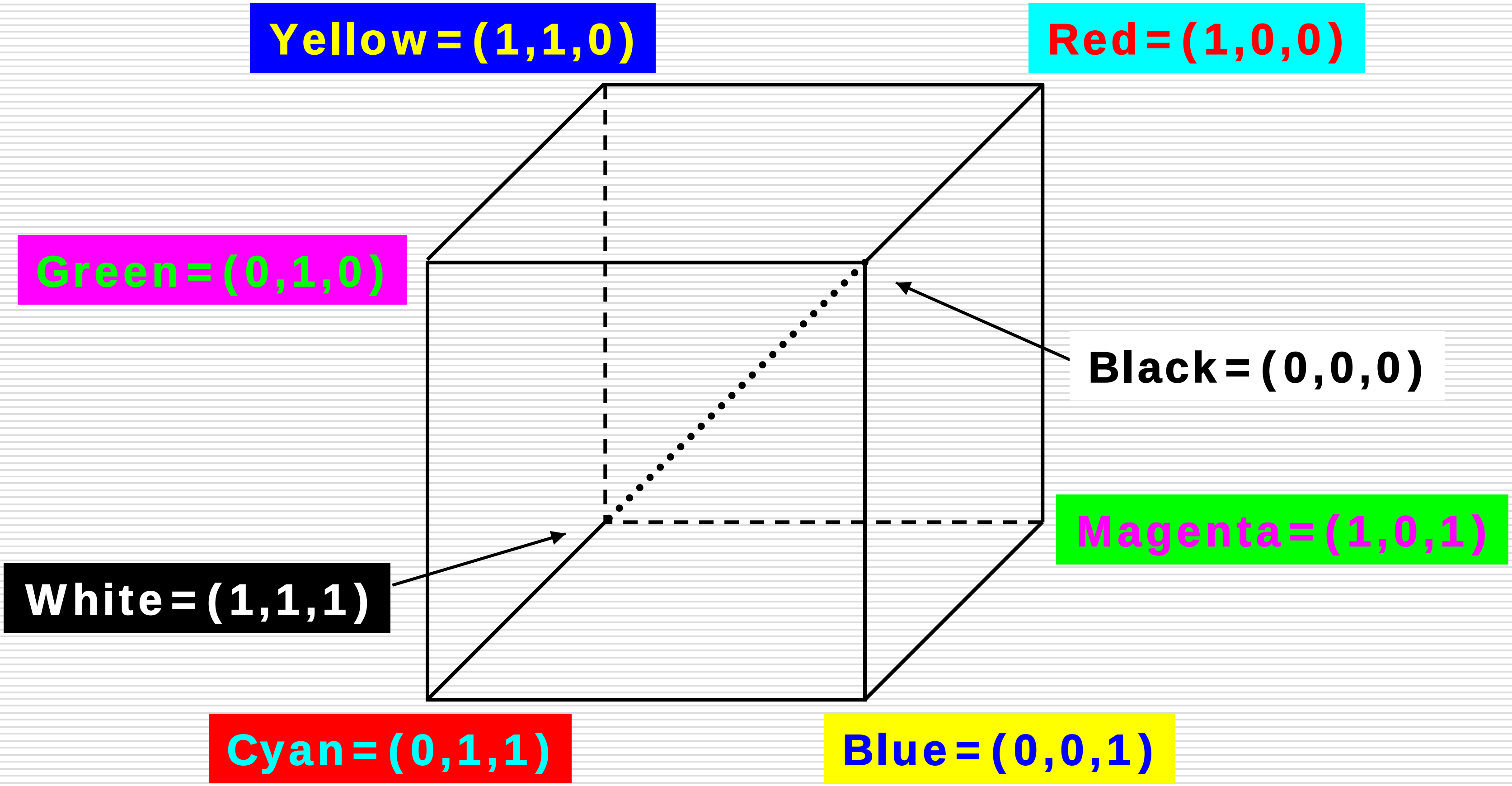
The RGB Color Model – for CRT



Color Depth

- Can choose number of bits for each of *r*, *g* and *b*
  - More bits per component means more colors can be distinguished, but image files will be larger
  - 8 bits (1 byte) per component: *24-bit color*, millions of colors
- If $r = g = b$, color is a shade of gray, so grayscale can be represented by a single value
  - 8 bits permits 256 grays

The CMY Color Model – for hardcopy



Undercolor Removal: CMYK System

- Real inks do not correspond to ideal subtractive primaries
- Combining three inks for black is undesirable
- Printers use *four process colors*, cyan, magenta, yellow and black
- CMYK gamut is not the same as RGB
  - Implications for using images prepared for print (CMYK) on the Web (RGB)

The CMYK Color Model – for hardcopy

- $C = G + B = W - R$
- $M = R + B = W - G$
- $Y = R + G = W - B$

$$\Downarrow$$

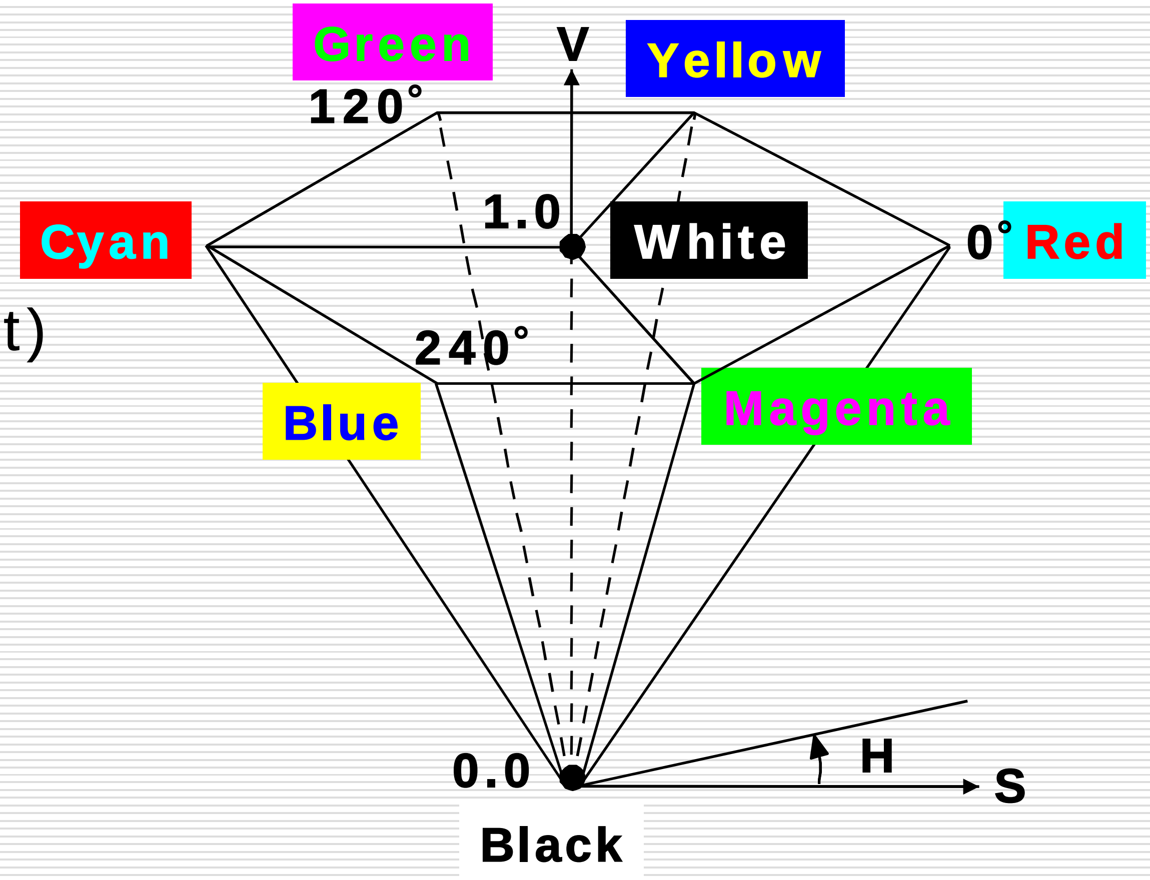
- $K = \min(C, M, Y)$
- $C \leftarrow C - K$
- $M \leftarrow M - K$
- $Y \leftarrow Y - K$

The HSV Color Model – for user-oriented

- Alternative way of specifying color
- *Hue* (roughly, dominant wavelength)
- *Saturation* (purity)
- *Value* (brightness)
- Model HSV as a cylinder: *H* angle, *S* distance from axis, *V* distance along axis
- Basis of popular style of *color picker*

The HSV Color Model – for user-oriented

- H : hue
- S : saturation
- V : value
  - (or B for blight)



Color Models in Video

- Largely derive from older analog methods of coding color for TV. Luminance is separated from color information.
- YIQ is used to transmit TV signals in North America and Japan. This coding also makes its way into VHS video tape coding in these countries since video tape technologies also use YIQ.
- In Europe, video tape uses the PAL or SECAM codings, which are based on TV that uses a matrix transform called YUV.
- Digital video mostly uses a matrix transform called YC_bC_r that is closely related to YUV.

The YUV Color Model – for PAL video

- Can be useful to separate brightness and color information, especially for video.
- *Y* is for **luminance** and *U* and *V* are for **chrominance** which are stored as two *color difference* values *B-Y* and *R-Y*.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

The YUV Color Model – for PAL video

- For dealing with composite video, it turns out to be convenient to contain *U* and *V* within the range −1/3 to +4/3. So *U* and *V* are rescaled:

$$U = 0.492111(B - Y)$$

$$V = 0.877283(R - Y)$$

- The chrominance signal = the composite signal *C*:

$$C = U \cdot \cos(\omega t) + V \cdot \sin(\omega t)$$

The YIQ Color Model – for NTSC color-TV

- Y : luminance
- I and Q : chromaticity
  (rotated version of U and V)

$$I = 0.492111(R - Y)\cdot\cos 33^\circ - 0.877283(B - Y)\cdot\sin 33^\circ$$

$$Q = 0.492111(R - Y)\cdot\sin 33^\circ + 0.877283(B - Y)\cdot\cos 33^\circ$$

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

The YC_bC_r Color Model – for digital video

$$C_b = ((B - Y)/1.772) + 0.5$$

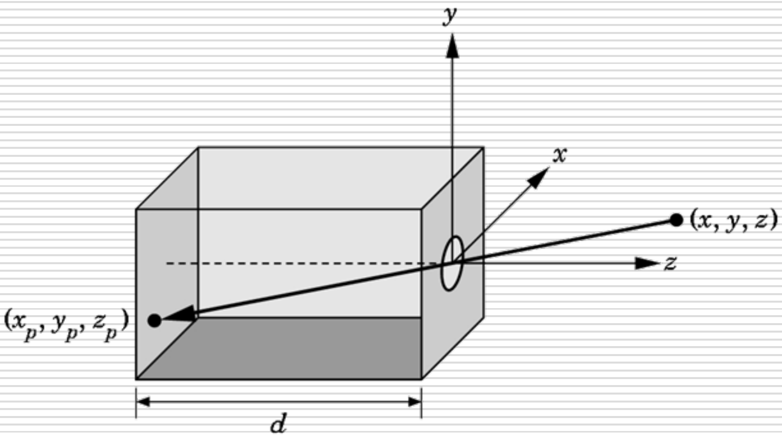
$$C_r = ((R - Y)/1.402) + 0.5$$

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168 & -0.332 & 0.5 \\ 0.5 & -0.418 & -0.082 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \end{bmatrix}$$

scaled to [0-255]

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

Pinhole Camera



Use trigonometry to find projection of a point

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

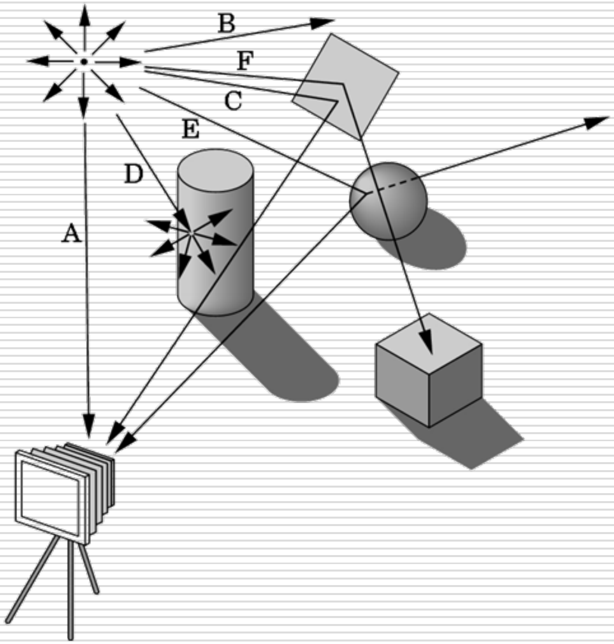
These are equations of simple perspective

Basics of Rendering

- Pipeline Based Rendering
  - Objects in the scene are rendered in a sequence of steps that form the Rendering Pipeline.
- Ray-Tracing
  - A series of rays are projected thru the view plane and the view plane is colored based on the object that the ray strikes

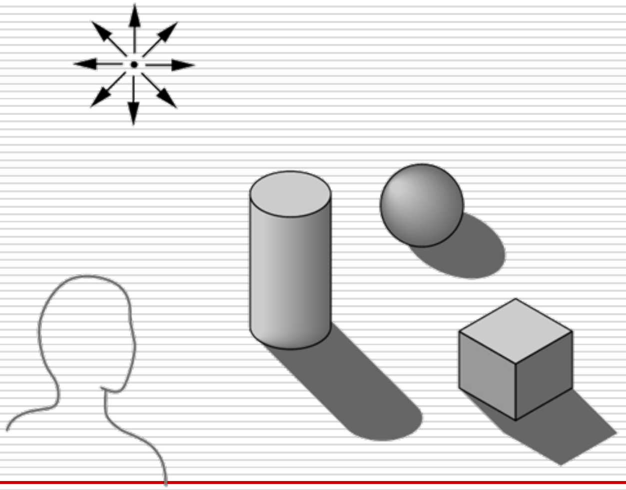
Ray Tracing and Geometric Optics

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.



Global vs. Local Lighting

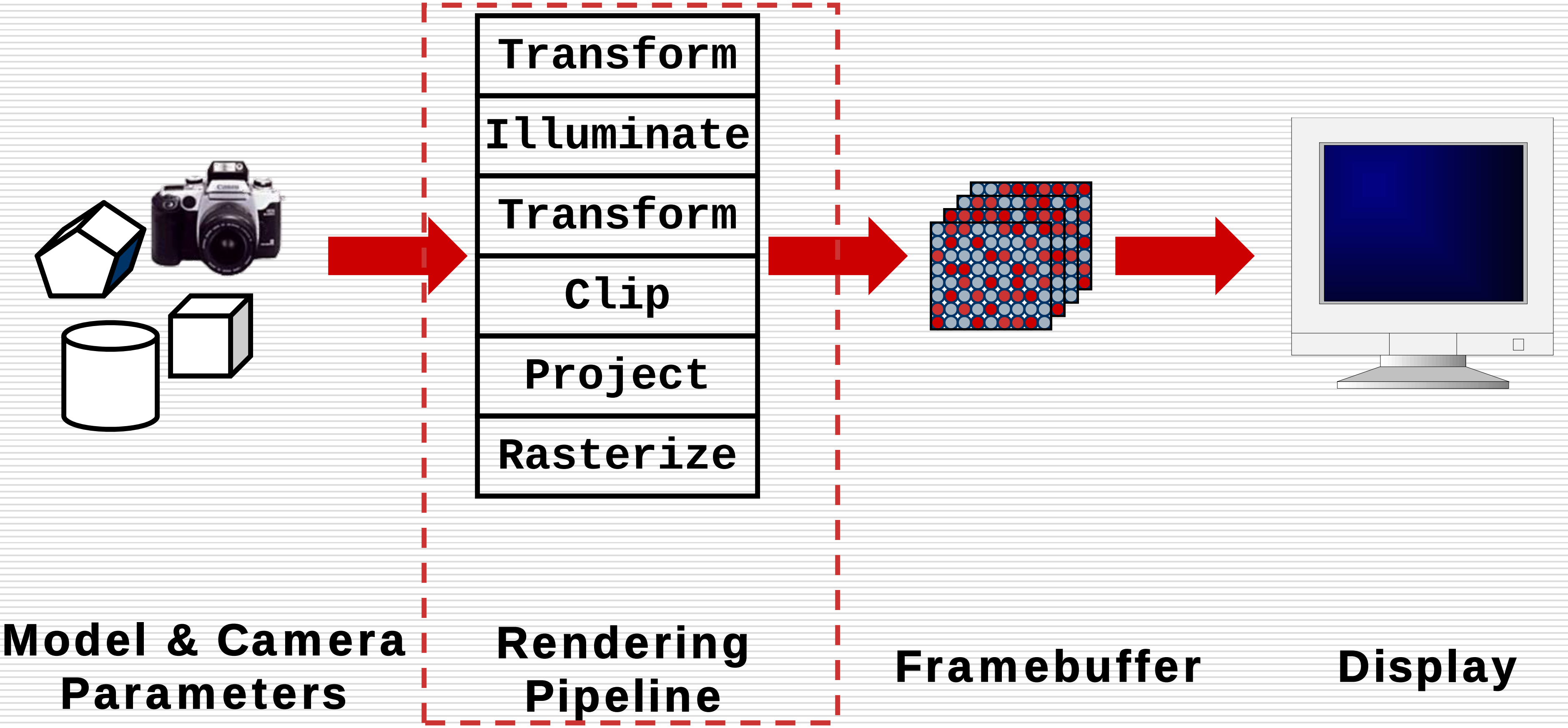
- Cannot compute color or shade of each object independently
  - Some objects are blocked from light
  - Light can reflect from object to object
  - Some objects might be translucent



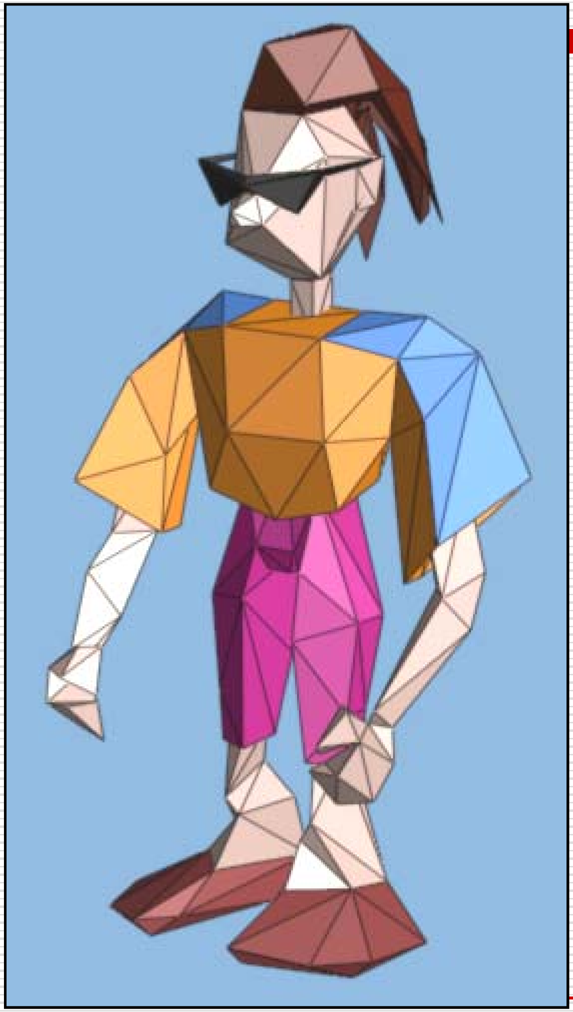
Why not ray tracing?

- 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

Pipeline Rendering



Definitions of Triangle Meshes



[Hoppe 99']

$\{f_1\} : \{ v_1 , v_2 , v_3 \}$ connectivity

$\{f_2\} : \{ v_3 , v_2 , v_4 \}$

…

$\{v_1\} : (x,y,z)$ geometry

$\{v_2\} : (x,y,z)$

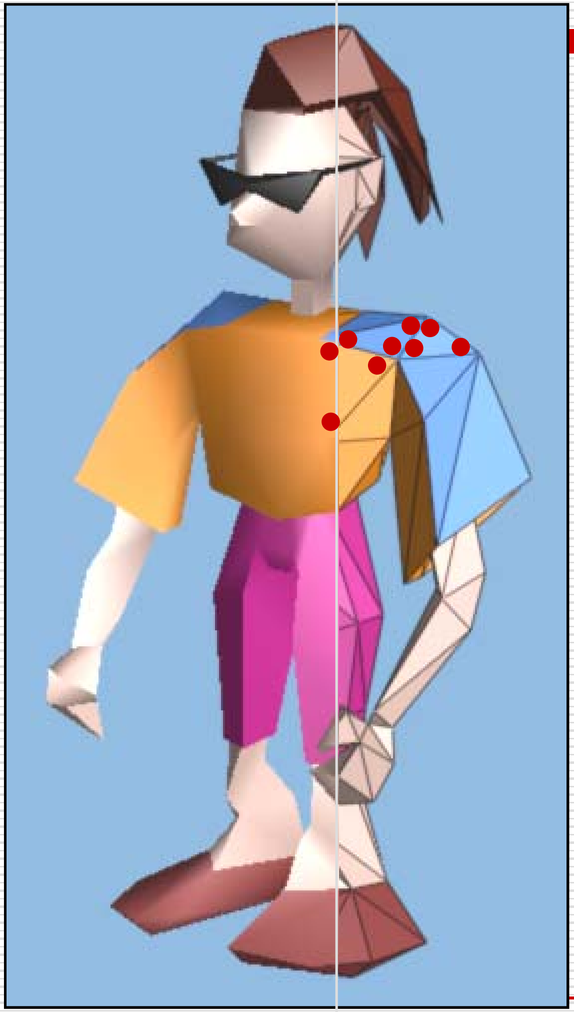
…

$\{f_1\}$: *"skin material"* face attributes

$\{f_2\}$: *"brown hair"*

…

Definitions of Triangle Meshes


[Hoppe 99']

$\{f_1\} : \{ v_1 , v_2 , v_3 \}$ connectivity

$\{f_2\} : \{ v_3 , v_2 , v_4 \}$

…

$\{v_1\} : (x,y,z)$ geometry

$\{v_2\} : (x,y,z)$

…

$\{f_1\}$: *"skin material"* face attributes

$\{f_2\}$: *"brown hair"*

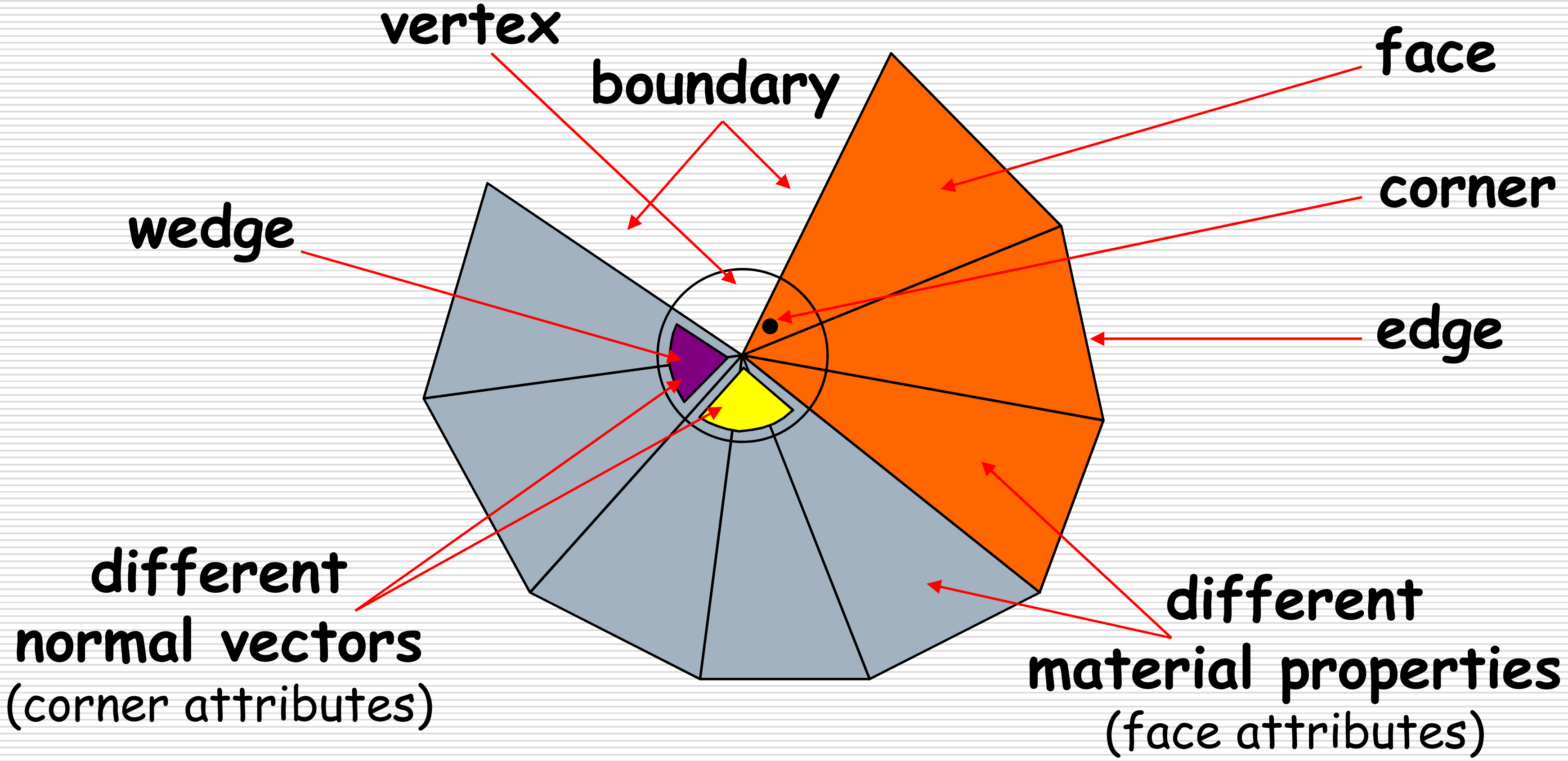
…

$\{v_2,f_1\} : (n_x,n_y,n_z)\ (u,v)$ corner attributes

$\{v_2,f_2\} : (n_x,n_y,n_z)\ (u,v)$

…

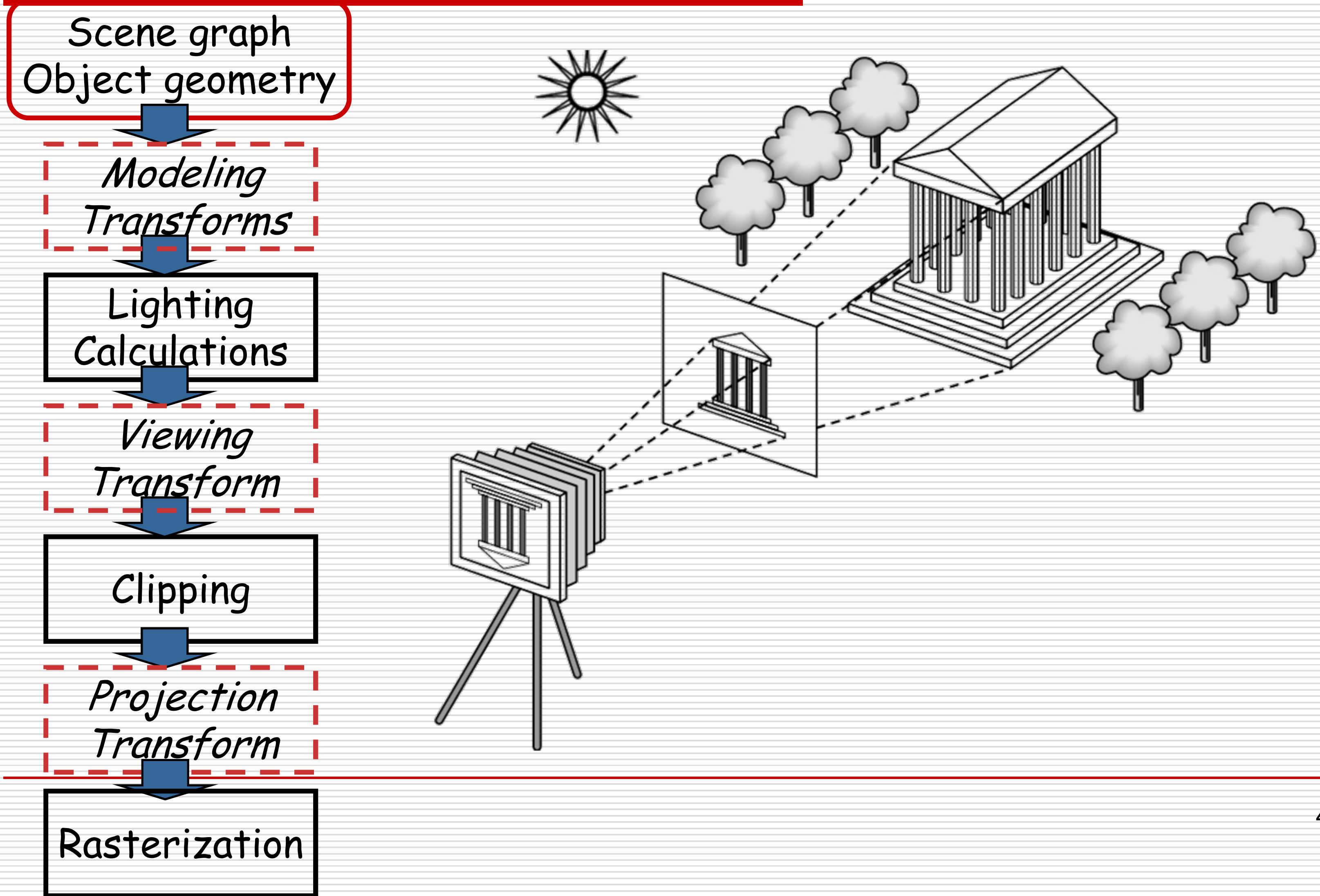
Definitions of Triangle Meshes



Rendering: Transformations

- So far, discussion has been in *screen space*
- But model is stored in *model space* (a.k.a. object space or world space)
- Three sets of geometric transformations:
  - Modeling transforms
  - Viewing transforms
  - Projection transforms

The Rendering Pipeline



APPENDIX:

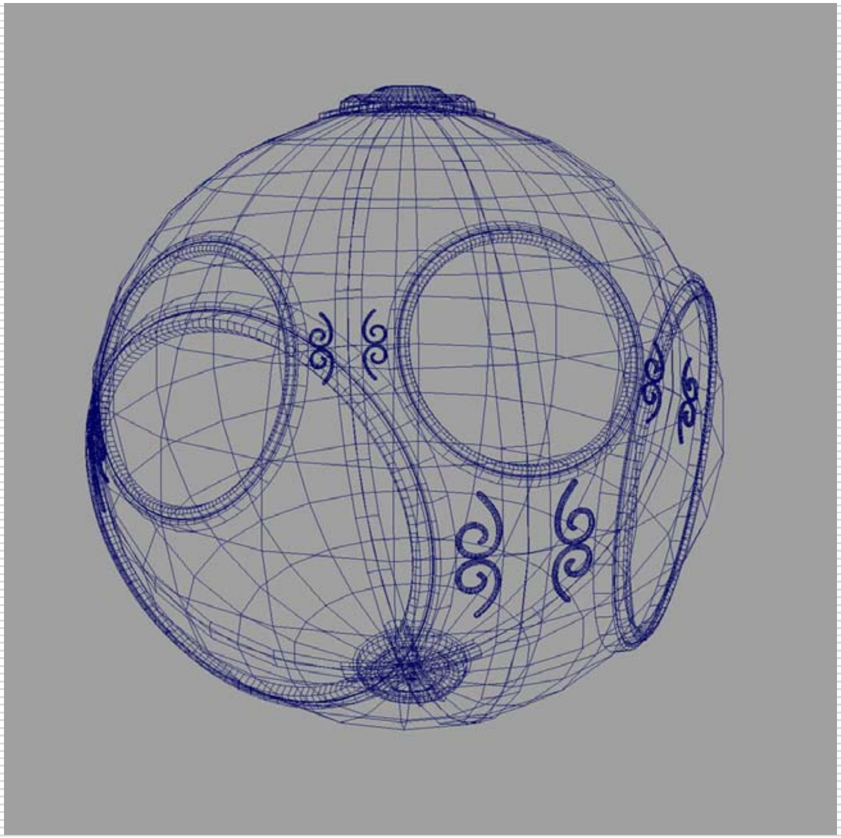
History of Computer Graphics

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
- Project Sketchpad
- Display Processors
- Storage tube

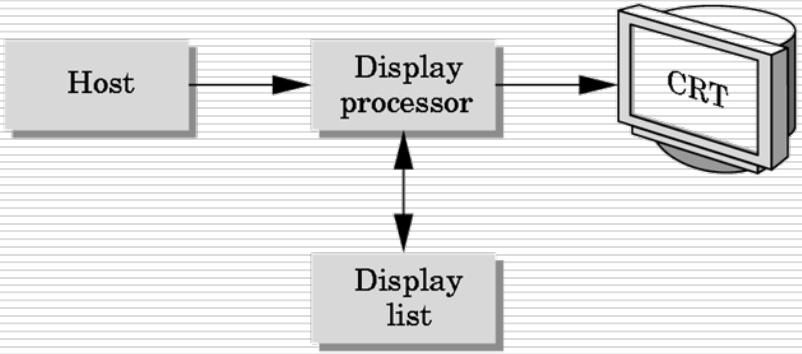


Project Sketchpad

- 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

Display Processor

- Rather than have 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

Direct View Storage Tube

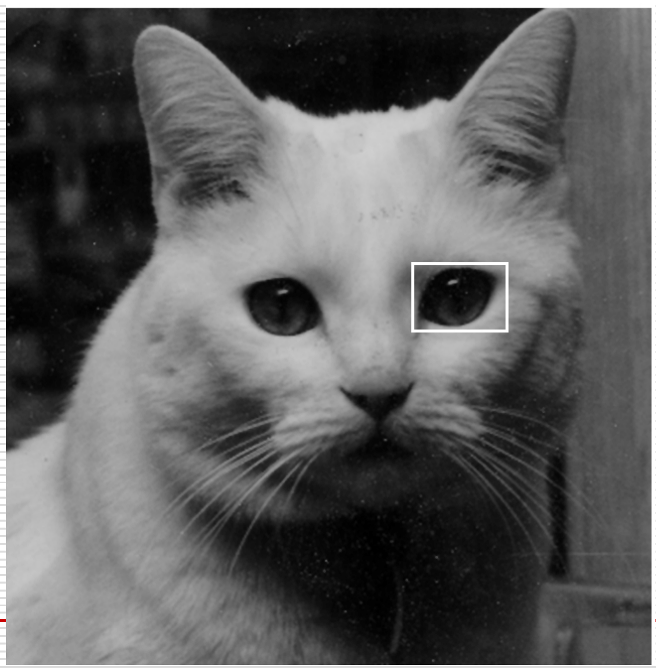
- Created by Tektronix
  - Did not require constant refresh
  - Standard interface to computers
    - Allowed for standard software
    - Plot3D in Fortran
  - Relatively inexpensive
    - Opened door to use of computer graphics for CAD community

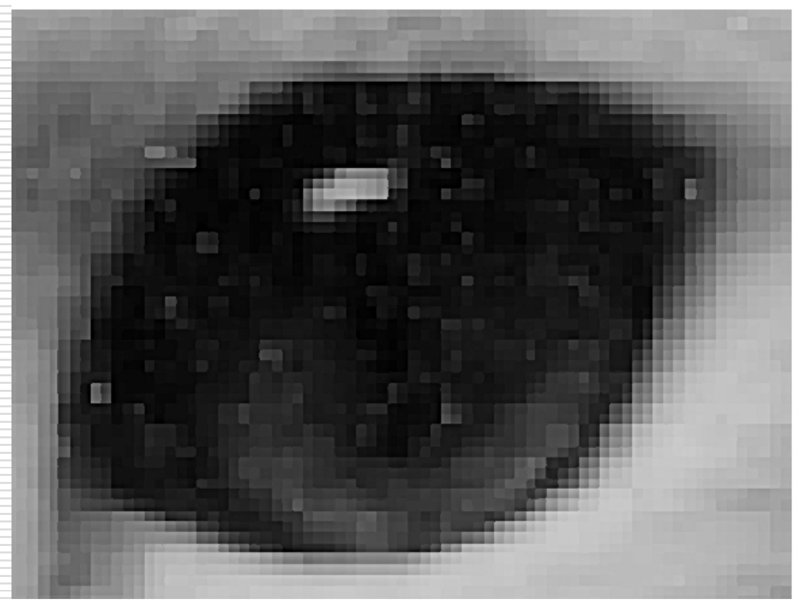
Computer Graphics: 1970-1980

- 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

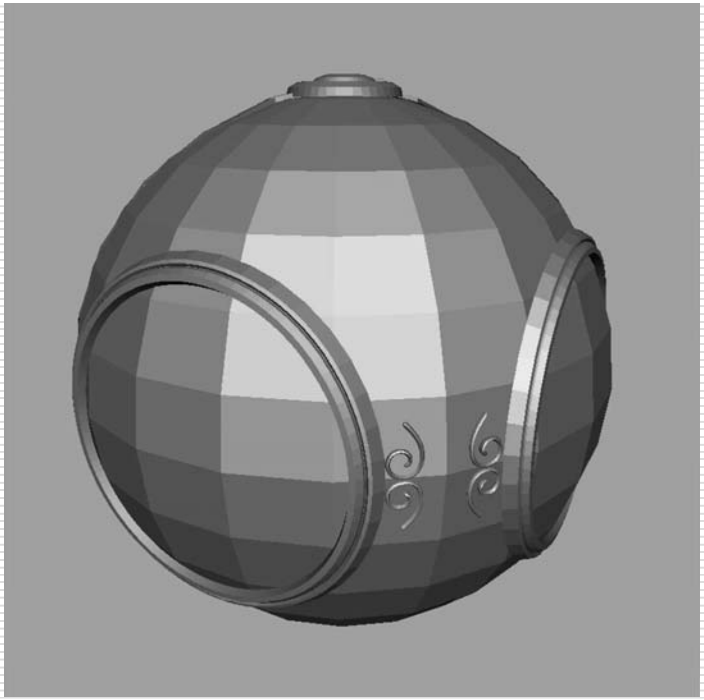
- Image produced as an array (the *raster*) of picture elements (*pixels*) in the *frame buffer*





Raster Graphics

- Allow us to go from lines and wireframes to filled polygons

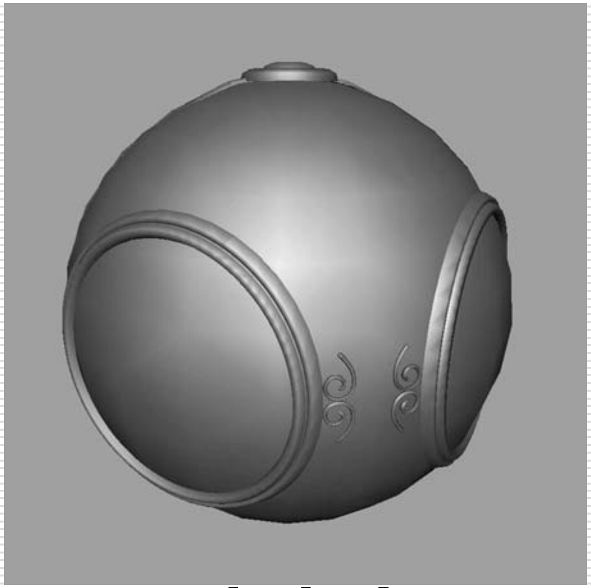


PCs and Workstations

- 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
    - High-level of interactivity
  - Early PCs included frame buffer as part of user memory

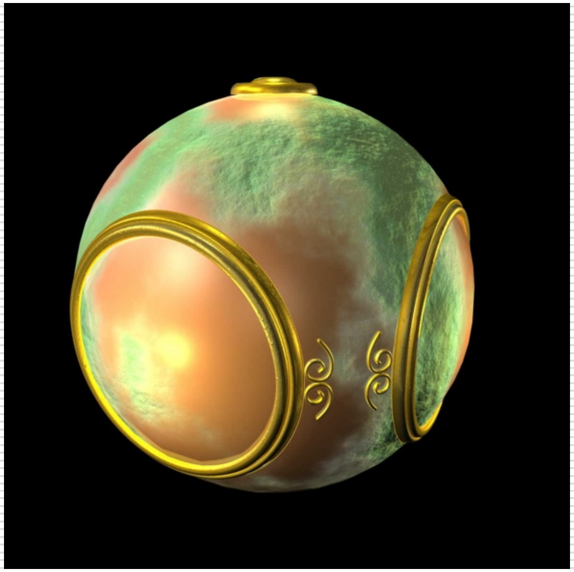
Computer Graphics: 1980-1990

- Realism comes to computer graphics



smooth shading

environmental mapping



bump mapping

Computer Graphics: 1980-1990

- Special purpose hardware
  - Silicon Graphics geometry engine
    - VLSI implementation of graphics pipline
- 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 buffer

Computer Graphics: 2000-

- Photorealism
- Graphics cards for PCs dominate market
  - Nvidia, ATI, 3DLabs
- Game boxes and game players determine direction of market
- Computer graphics routine in movie industry: Maya, Lightwave