# Introduction to Computer Graphics

# Introduction

Still from Pixar's Inside Out, 2015

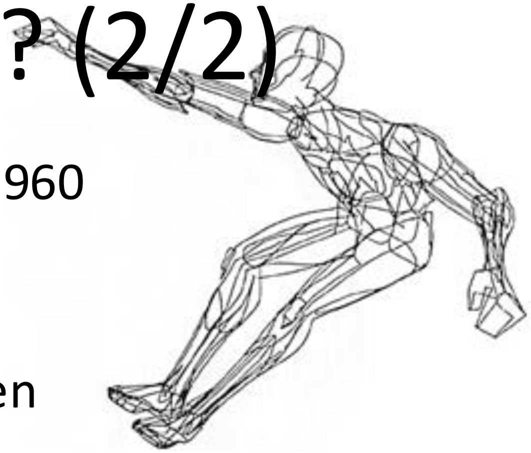
## CS423: Computer Graphics

# What is Computer Graphics? (1/2)

- Computer graphics generally means creation, storage and manipulation of models and images
- Such models come from diverse and expanding set of fields including physical, biological, mathematical, artistic, and conceptual/abstract structures

Frame from animation by William Latham, shown at **SIGGRAPH 1992**. Latham creates his artwork using rules that govern patterns of natural forms.

# What is Computer Graphics? (2/2)



- William Fetter coined term “computer graphics” in 1960 to describe new design methods he was pursuing at Boeing for cockpit ergonomics
- Created a series of widely reproduced images on “pen plotter” exploring cockpit design, using 3D model of human body.

“Perhaps the best way to define computer graphics is to find out what it is not. It is not a machine. It is not a computer, nor a group of computer programs. It is not the know-how of a graphic designer, a programmer, a writer, a motion picture specialist, or a reproduction specialist.

Computer graphics is all these – a consciously managed and documented technology directed toward communicating information accurately and descriptively.”

***Computer Graphics***, by William A. Fetter, 1966

# What is **Interactive** Computer Graphics?

- User controls content, structure, and appearance of objects and their displayed images via rapid visual feedback
- Basic components of an interactive graphics system
  - input (e.g., mouse, stylus, multi-touch, in-air fingers…)
  - processing (and storage of the underlying representation)
  - display/output (e.g., screen, paper-based printer, video recorder…)
- First truly interactive graphics system, **Sketchpad**, pioneered by Ivan Sutherland 1963 Ph.D. thesis *Sketchpad, A Man-Machine Graphical Communication System*
- Used TX-2 transistorized "mainframe" at MIT Lincoln Lab

Note CRT monitor, light pen, and function-key panels – the "organ console" showing bi-manual operation

# What is **Batch** Computer Graphics?

- Today, still use non-interactive *batch mode* for final production-quality video and film (special effects – FX). Rendering a single frame of Monsters University (a 24 fps movie) averaged 29 hours on a 24,000-core render farm!

Still from Monsters University


Render farm

# Enabling Modern Computer Graphics (1/5)

- Hardware revolution
  - **Moore's Law:** every 12-18 months, computer power improves by factor of 2 in price / performance as size shrinks
  - Newest CPUs are 64-bit with 2, 4, 6, 8, even up to 18 cores
  - Intel Skylake – consumer processor with 4 cores, 8 threads, and a fully featured graphics chip built in to the processor
  - Significant advances in commodity graphics chips every 6 months vs. several years for general purpose CPUs
    - NVIDIA GeForce GTX Titan X… 3072 cores, 12GB memory, and 7 teraflops of processing power in a single chip

# Enabling Modern Computer Graphics (2/5)

- Graphic subsystems
  - Offloads graphics processing from CPU to chip designed for doing graphics operations quickly
  - nVidia GeForce™, AMD Radeon™, and Intel HD and Iris Pro Graphics
  - GPUs originally designed to handle special-purpose graphics computations
  - Increasingly, GPUs used to parallelize other types of computation (known as **GPGPU**, or General-Purpose Computing on the Graphics Processing Unit)
- Hardware show and tell: Dept's NVIDIA GeForce GTX 460s
  - 1.35 GHz clock, 1GB memory, 37.8 billion pixels/second fill rate
  - Old cards: GeForce 7300 GT: 350 MHz clock, 256 MB memory, 2.8 billion fill rate

# Enabling Modern Computer Graphics (3/5)

- Input Devices
  - Mouse, tablet & stylus, multi-touch, force feedback, and other game controllers (e.g., Wii), scanner, digital camera (images, computer vision), etc.
  - Body as interaction device

Xbox Kinect


Leap Motion


Nimble UX

# Enabling Modern Computer Graphics (4/5)

- Many form factors
  - Smartphones/laptops/desktops/tablets
  - Smart watches
  - Head-mounted displays (HMDs)
    - [Oculus bird simulator video]
  - 3D immersive virtual reality spaces


Apple iPhone

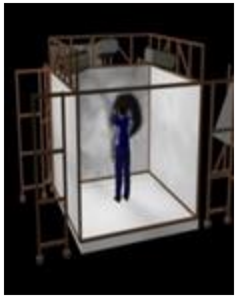

Android Phones


Tablets


Microsoft's first Surface


Apple Watch


Android Wear


Brown's old Cave


Microsoft Hololens


Oculus Rift


Google Cardboard

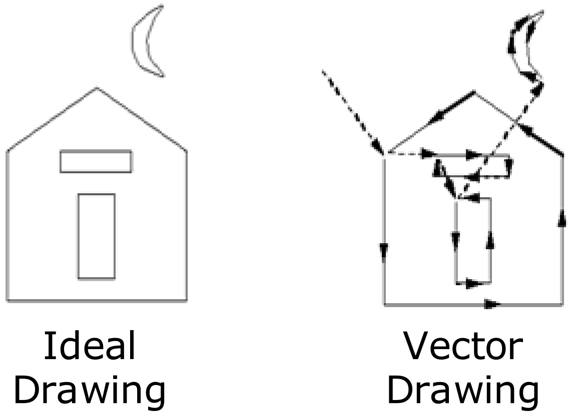
# Enabling Modern Computer Graphics (5/5)

- Software Improvements
  - Algorithms and data structures
    - Modeling of materials
    - Rendering of natural phenomena
    - “Acceleration data structures” for ray tracing and other renderers
  - Parallelization
    - Most operations are embarrassingly parallel: changing value of one pixel is often independent of other pixels
  - Distributed and Cloud computing
    - Send operations into ‘cloud’, get back results, don’t care how
    - Rendering even available as internet service!

# Graphics Display Hardware

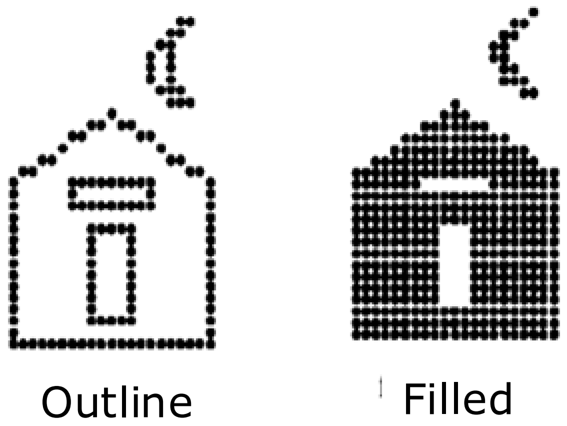
**Vector** (calligraphic, stroke, random-scan)

- Driven by display commands
  - (move (x, y), char("A") , line(x, y)…)
- Survives as "scalable vector graphics"




**Raster** (TV, bitmap, pixmap) used in displays and laser printers

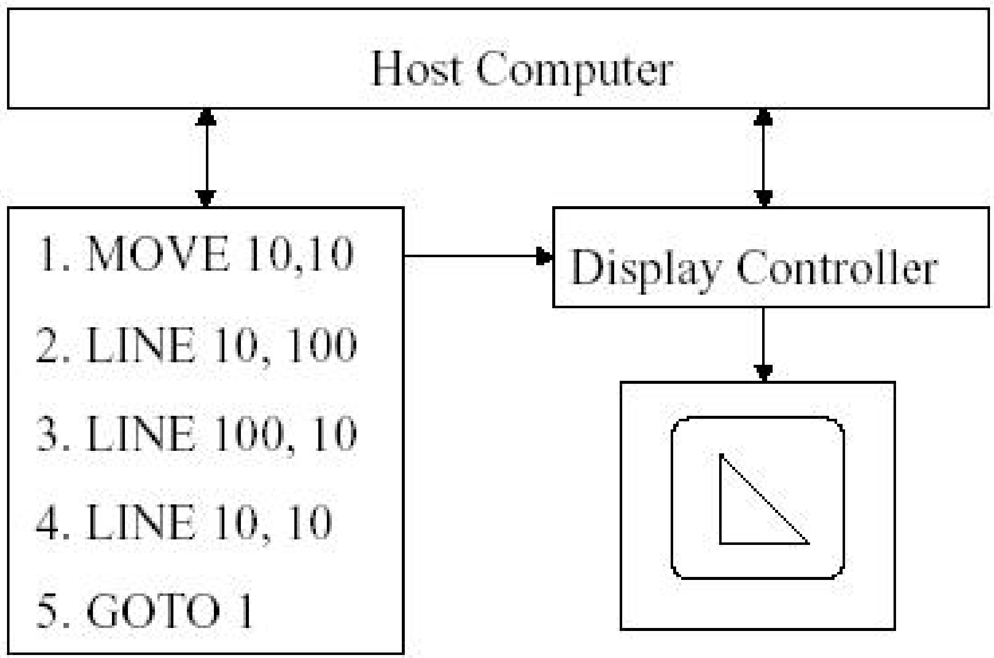
- Driven by array of pixels (no semantics, lowest form of representation)
- Note "jaggies" (aliasing errors) due to discrete sampling of continuous primitives



# Output Technology (1/3)

- Calligraphic Displays
  - also called vector, stroke or line drawing graphics
  - lines drawn directly on phosphor
    - display processor directs electron beam according to list of lines defined in a "**display list**“
    - phosphors glow for only a few micro-seconds so lines must be redrawn or **refreshed** constantly
    - deflection speed limits # of lines that can be drawn without flicker.

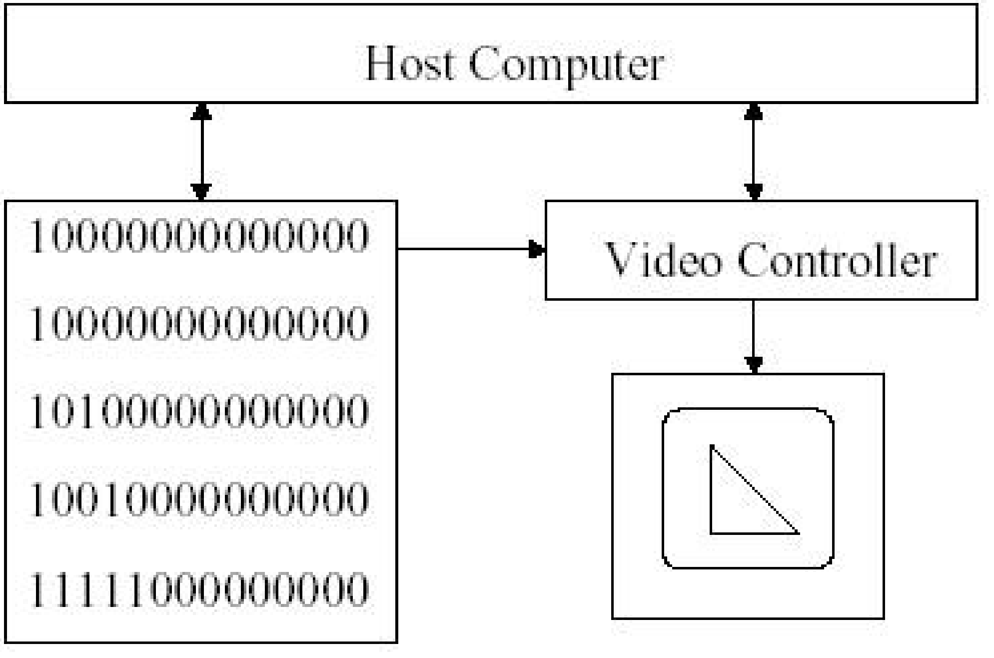
# Vector Display Architecture



# Output Technology (2/3)

- Raster Display
  - Display **primitives** (lines, shaded regions, characters) stored as pixels in **refresh buffer** (or **frame buffer**)
  - Electron beam scans a regular pattern of horizontal raster lines connected by horizontal retraces and vertical retrace
  - Video controller coordinates the repeated scanning
  - Pixels are individual dots on a raster line

# Raster displays Architecture



# Output Technology (cont)

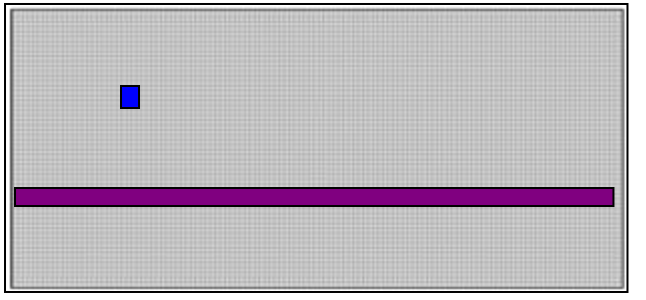
- Bitmap is the collection of pixels
- Frame buffer stores the bitmap
- Raster display store the display primitives (line, characters, and solid shaded or patterned area)
- Frame buffers
  - are composed of **VRAM** (video RAM).
- VRAM is dual-ported memory capable of
  - Random access
  - Simultaneous high-speed serial output: built-in **serial shift register** can output entire *scanline* at high rate synchronized to **pixel clock.**

# Pros and Cons

- Advantages to Raster Displays
  - lower cost
  - filled regions/shaded images
- Disadvantages to Raster Displays
  - a discrete representation, continuous primitives must be **scan-converted** (i.e. fill in the appropriate scan lines)
  - **Aliasing** or "**jaggies**" Arises due to sampling error when converting from a continuous to a discrete representation

# Basic Definitions

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



**Video raster devices display an image by sequentially drawing out the pixels of the scan lines that form the raster.**

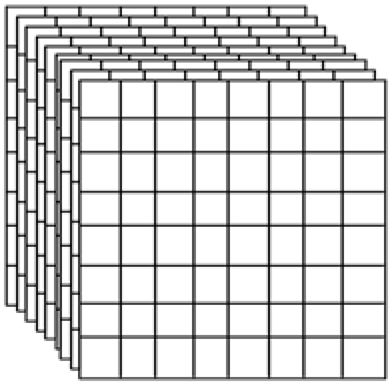
# Example

- **Television**
  - **NTSC** **640x480x8b** **1/4 MB**
  - **GA-HDTV** **1920x1080x8b** **~2 MB**
- **Workstations**
  - **Bitmapped display** **960x1152x1b** **~1 Mb**
  - **Color workstation** **1280x1024x24b** **5 MB**
- **Laserprinters**
  - **300 dpi** **(8.5”x300)(11”x300)** **1.05 MB**
  - **2400 dpi** **(8.5”x2400)(11”x2400) ~64 MB**
- **Film (line pairs/mm)**
  - **35mm (diagonal) slide (ASA25~125 lp/mm) = 3000**

    **3000 x 2000 x 3 x 12b ~27 MB**

# Frame Buffer

- A frame buffer is characterized by its size, x, y, and pixel depth.
- the **resolution** of a frame buffer is the number of pixels in the display. e.g. 1024x1024 pixels.
- Bit Planes or Bit Depth is the number of bits corresponding to each pixel. This determines the **color resolution** of the buffer.
- Dual ported (simultaneously writing values and displaying in the monitor)



**Bilevel or monochrome displays have 1 bit/pixel (128Kbytes of RAM)**
**8bits/pixel -> 256 simultaneous colors**
**24bits/pixel -> 16 million simultaneous colors**

# Comparing Raster and Vector (1/2)

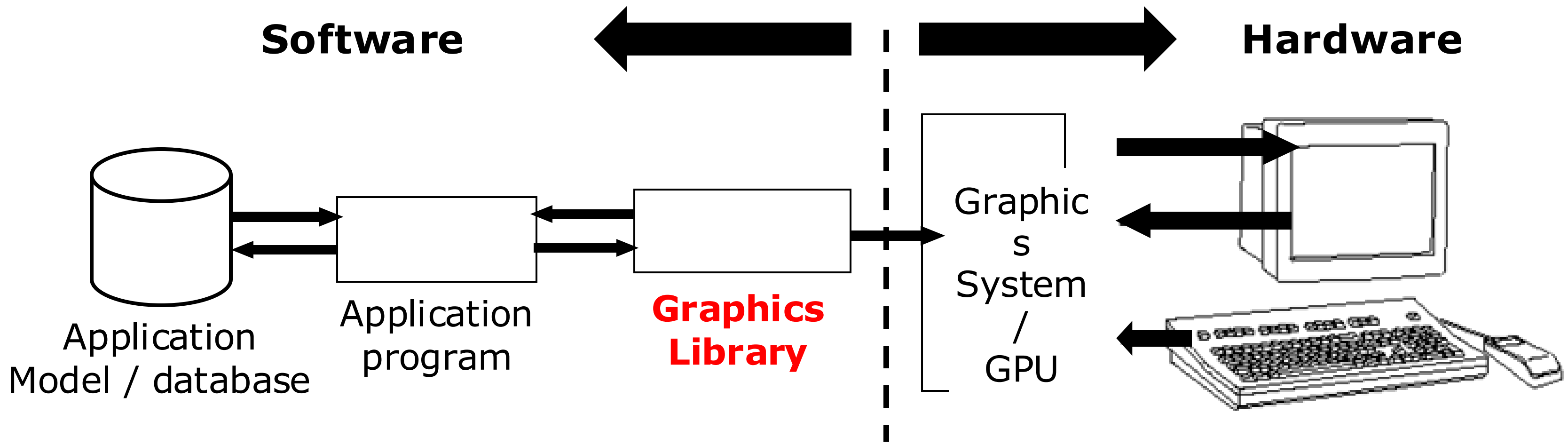
- advantages of vector:
  - very fine detail of line drawings (sometimes curves), whereas raster suffers from jagged edge problem due to pixels (aliasing, quantization errors)
  - geometry objects (lines) whereas raster only handles pixels
  - eg. 1000 line plot: vector disply computes 2000 endpoints
  - raster display computes all pixels on each line

# Comparing Raster and Vector (2/2)

- advantages of raster:
  - cheaper
  - colours, textures, realism
  - unlimited complexity of picture: whatever you put in refresh buffer, whereas vector complexity limited by refresh rate
  - HYBRIDS?

# Conceptual Framework for Interactive Graphics

- Graphics library/package is **intermediary** between application and display hardware (Graphics System)
- Application program maps application objects to views (images) of those objects by calling on graphics library. Application model may contain lots of non-graphical data (e.g., non-geometric object properties)
- User interaction results in modification of image and/or model
- This hardware and software framework is 5 decades old but is still useful

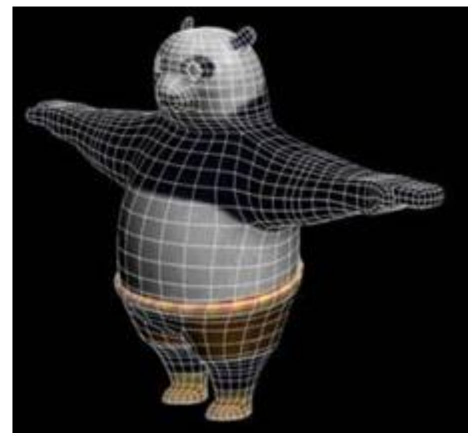


# Graphics Library

- Examples: OpenGL™, DirectX™, Windows Presentation Foundation™ (WPF), RenderMan™, HTML5 + WebGL™
- Primitives (characters, lines, polygons, meshes,…)
- Attributes
  - Color, line style, material properties for 3D
- Lights
- Transformations

# Application Distinctions: Two Basic Paradigms

Sample-based graphics *v/s* Geometry-based graphics



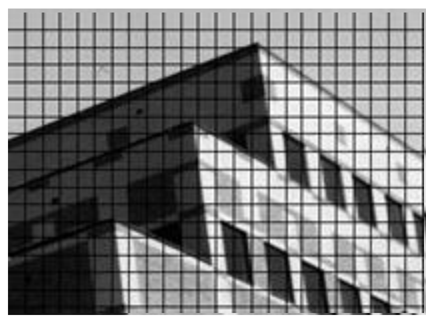
# Sampling an Image

- Lets do some sampling of a building

3D scene

- A color value is measured at every grid point and use[d] … g grid square

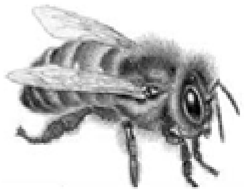
0 = white, 5 = gray, 10 = black



- … l image … d creat …

# What's the Advantage?

- Once image is defined in terms of colors at (x, y) locations on grid, can change image easily by altering location or color values
- E.g., if we reverse our mapping above and make 10 = white and 0 = black, the image would look like this:
- Pixel information from one image can be copied and pasted into another, replacing or combining with previously stored pixels



# What’s the Disadvantag

- WYSIAYG (What You See Is All You Get): No additional information
  - no depth information
  - can’t examine scene from different point of view
  - at most can play with the individual pixels or groups of pixels to change colors, enhance contrast, find edges, etc.
  - But increasingly great success in image-based rendering to fake 3D scenes and arbitrary camera positions. New images constructed by interpolation, composition, warping and other operations.
  - For a computational and cognitive science perspective, Take Thomas Serre’s *Computational Vision* (CLPS1520)

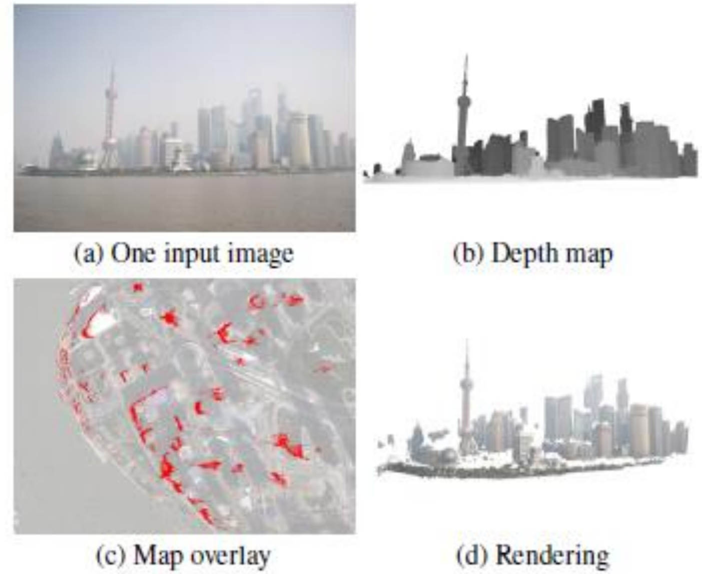
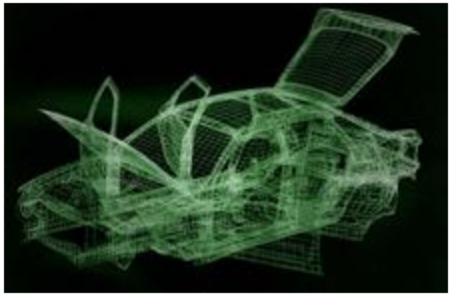


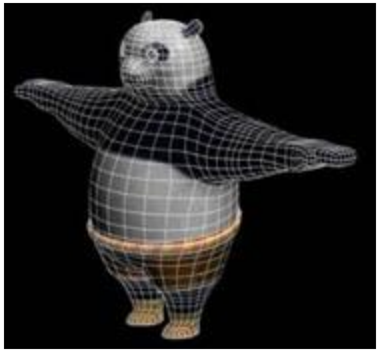

**Figure 15:** *Results on a challenging unstructured light field, obtained by hand-held capture (a) from a floating boat. (b) A resulting depth map. (c) Overlay of our reconstruction on a satellite image ©2013 DigitalGlobe, Google. (d) Rendering from a novel viewpoint.*

*“Scene Reconstruction from High Spatio-Angular Resolution Light Fields”* by Kim, Zimmer et al., 2013

# Geometry-Based Graphics

- **Geometry-based graphics** (also called scalable vector graphics or object-oriented graphics): geometrical model is created, along with various appearance attributes, and is then sampled for visualization (rendering, a.k.a image synthesis)
  - often some aspect of physical world is visually simulated, or "synthesized"
  - examples of 2D apps: Adobe Illustrator™, Adobe Freehand™, Corel CorelDRAW™
  - examples of 3D apps: Autodesk's AutoCAD™, Autodesk's (formerly Alias|Wavefront's) Maya™, Autodesk's 3D Studio Max™





# What is Geometric Modeling?

- What is a model?
- Captures salient features (data, behavior) of object/phenomenon being modeled
  - data includes geometry, appearance, attributes…
  - note similarity to OOP ideas
- Modeling allows us to cope with complexity
- Our focus: modeling and viewing simple everyday objects
- Consider this:
  - Through 3D computer graphics, we have abstract, easily changeable 3D forms, for the first time in human history
    - Has revolutionized working process of many fields – science, engineering, industrial design, architecture, commerce, entertainment, etc. Profound implications for visual thinking and visual literacy
    - "Visual truth" is gone in the Photoshop and FX-saturated world (but consider painting and photography…) – seeing no longer is believing…(or shouldn't be!)

# Modeling vs. Rendering

- **Modeling**
  - Create models
  - Apply materials to models
  - Place models around scene
  - Place lights in scene
  - Place the camera

- **Rendering**

  Take “picture” with camera

- Both can be done with commercial software:

  Autodesk Maya™ ,3D Studio Max™, Blender™, etc.

# A Brief Overview

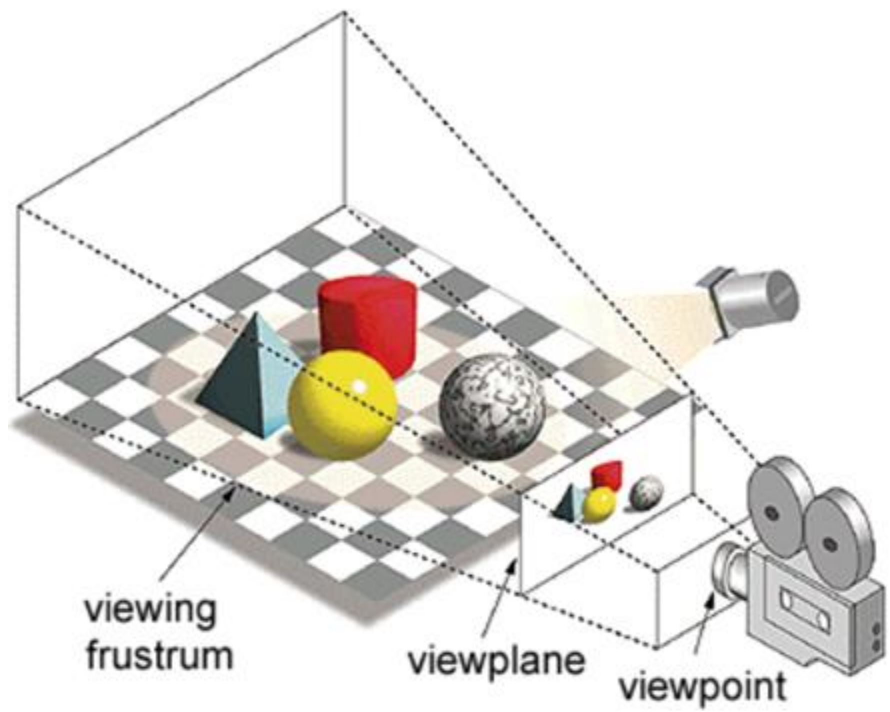
- Object space vs Image Space
- Computer graphics vs Image processing
- Computer graphics areas
  - *Modeling: A* coffee mug might be described as a set of ordered 3D points along with some interpolation rule to connect the points and a reflection model that describes how light interacts with the mug.
  - *Rendering:* creation of shaded images from 3D computer models.
  - *Animation:* create an illusion of motion through sequences of images.

# Major Application

- Video games
- Cartoons/Animated flims
- Visual effects
- CAD/CAM
- Simulator: MS flight simulator
- Medical imaging

# What is Computer Graphics

- Model real world
- Generate high quality 2d images efficiently, given the 3d model and camera parameters



# Background knowledge

- Solving equations
- Linear algebra
- Vector
- Trigonometry
- ODE, PDE (maybe ☹)

# Two Approaches

- Rasterization
  - Faster
  - Pipelined
  - Ex. Real-time applications
- Ray Tracing
  - High quality and More realistic
  - Ex. Animation movies

# Raster Images

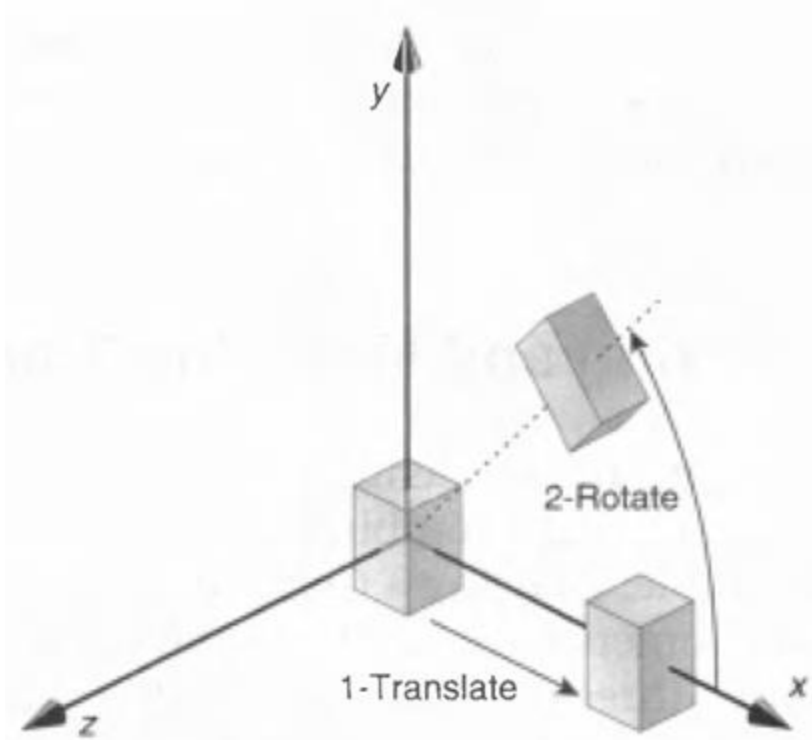
- Show images as rectangular arrays of *pixels*
- Output devices:
  - LCD (Transmissive), LED (Emissive)
  - Ink jet printers, Dye submission printers
- Input devices:
  - 2D array sensor: digital camera
  - 1D array sensor: flatbed scanner

# Rasterization Pipeline

- Modeling Transformation
- View Transformation
- Illumination
- Projection Transformation
- Clipping
- Rasterization or Scan conversion
- Texturing

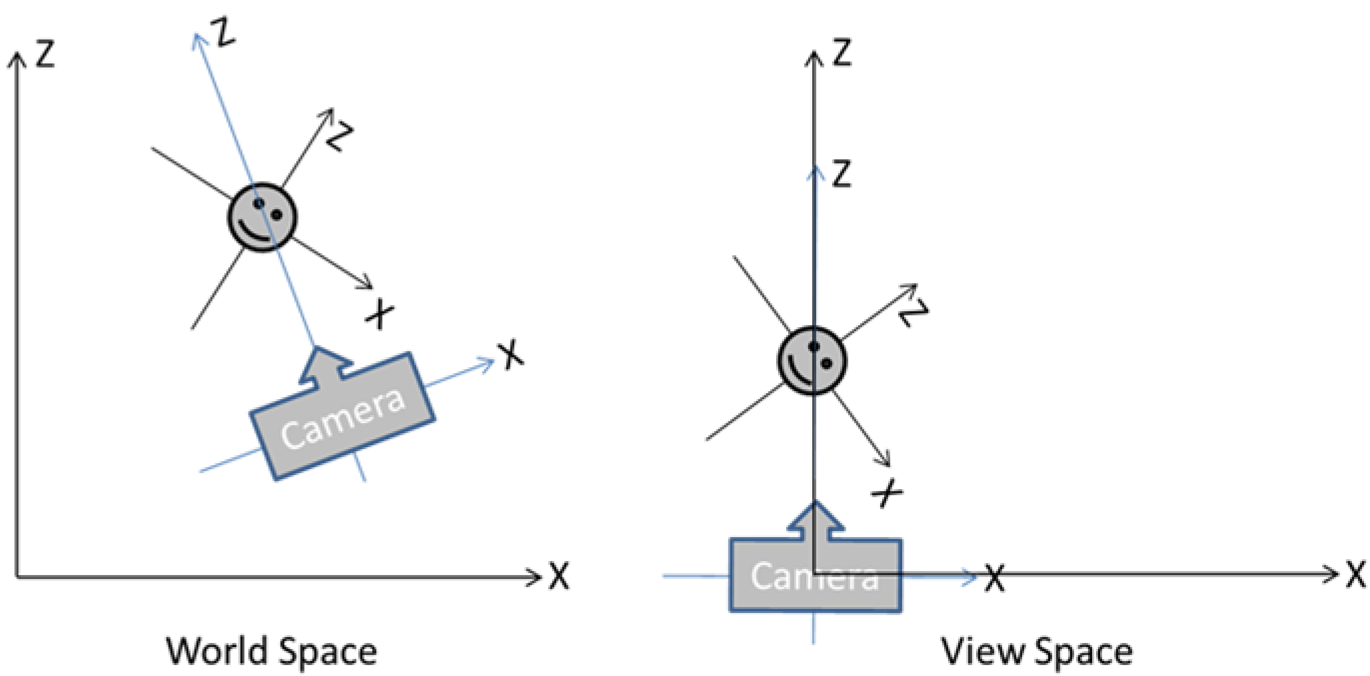
# Modeling Transformation

- Use transformation to position objects
- Reuse objects



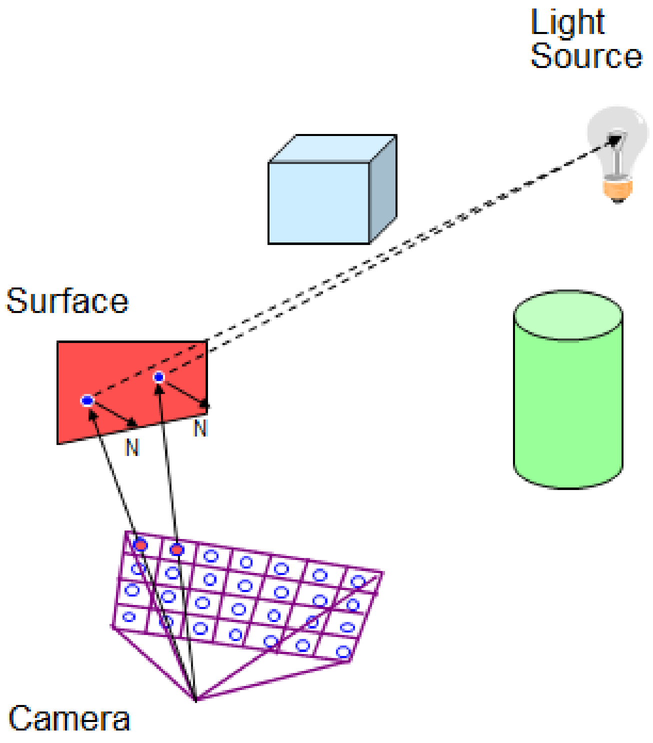
# View Transformation

- Translate camera to origin
- Set view direction along a principal axis



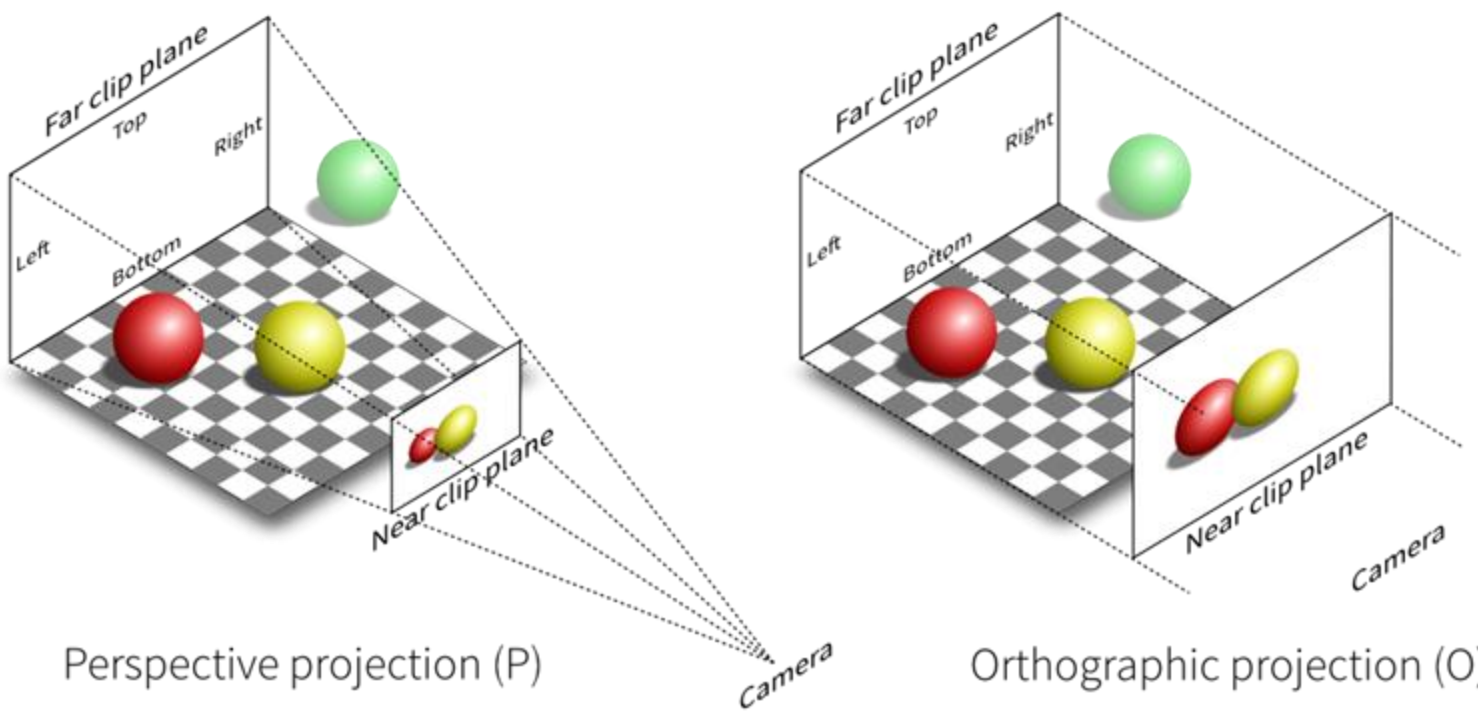
# Illumination

- Light the scene according to material properties of objects and light sources
- Highly important to generate realistic image



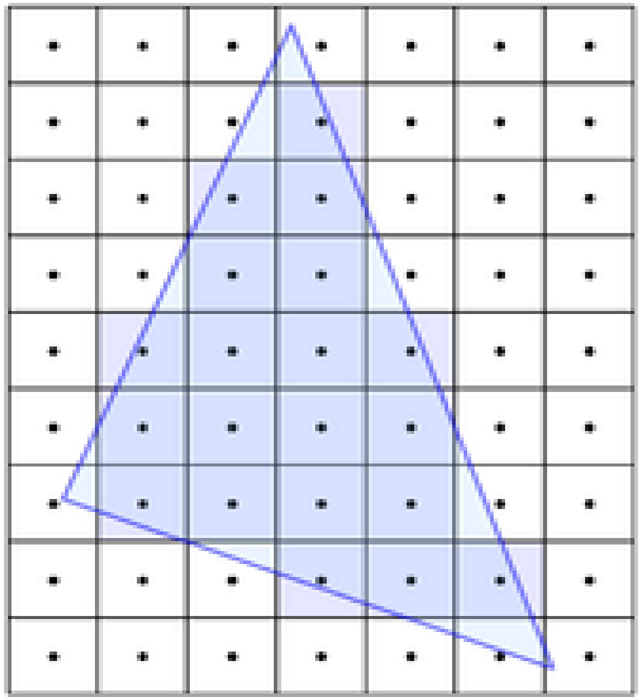
# Projection and Clipping

- Project the 3d model on a 2d screen
- Clip objects outside viewing frustum



# Rasterization

- Determine which pixels to light on the screen

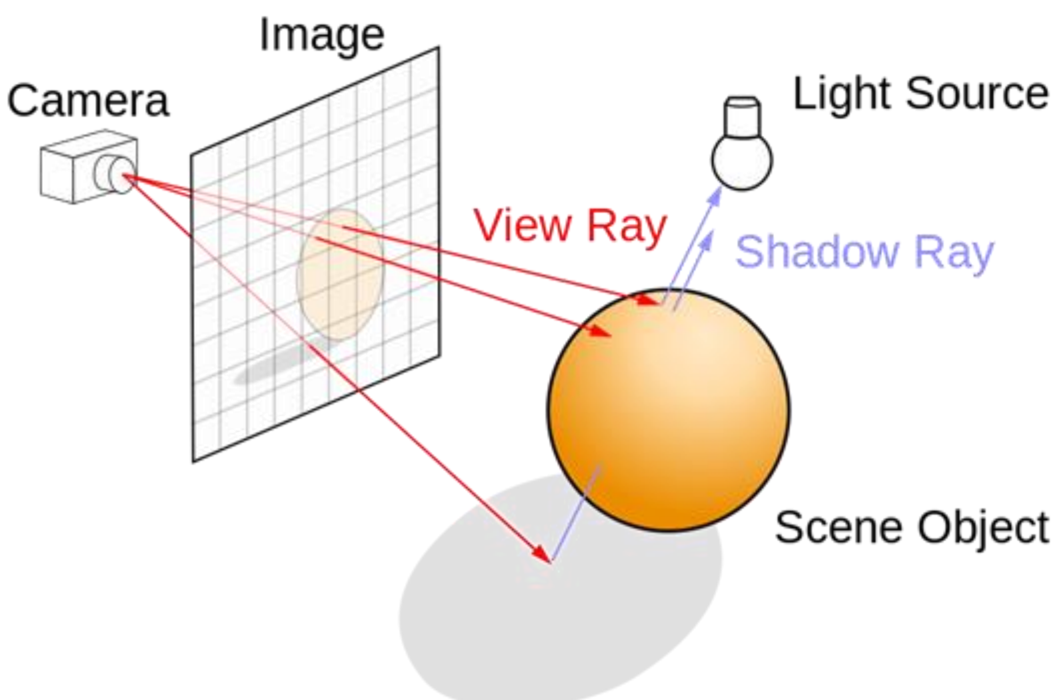


# Texturing

- Paste images on object surfaces

# Ray Tracing

- Cast rays from eye to each pixel to determine the color of pixel



# Ray Tracing

- Produces highly realistic image