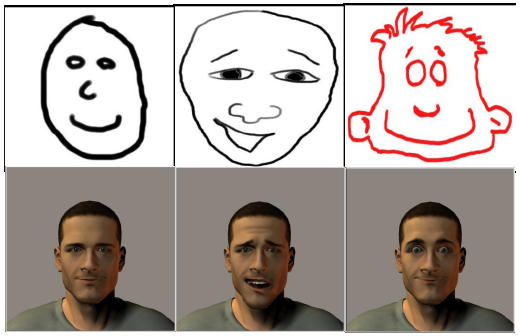
EECS 3431--- Fall 2022
Introduction to 3D Computer Graphics

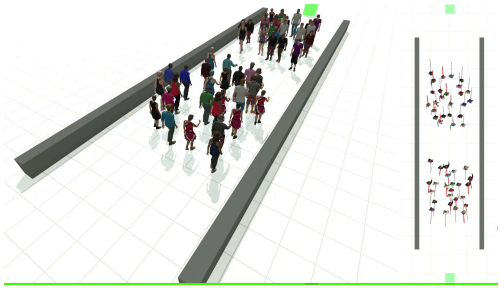
Instructor: Petros Faloutsos

Teaching Assistant: Tasneem Naheyan
tnaheyan@eecs.yorku.ca

Petros Faloutsos, Professor EECS, York, UHN-TRI

- Computer Animation
- Virtual Humans
  - Motor control, Facial Animation
  - Autonomous Behaviours
  - Crowd Simulation
- Serious Games for Rehabilitation
  - Speech Pathology Interventions





Applications of Computer Graphics

Entertainment

- Computer games
- Films
- Virtual reality

Scientific visualization

- Medical visualization
- Flight simulation
- Architecture
- Information visualization

Education

Movies

To reality and beyond !

Movies

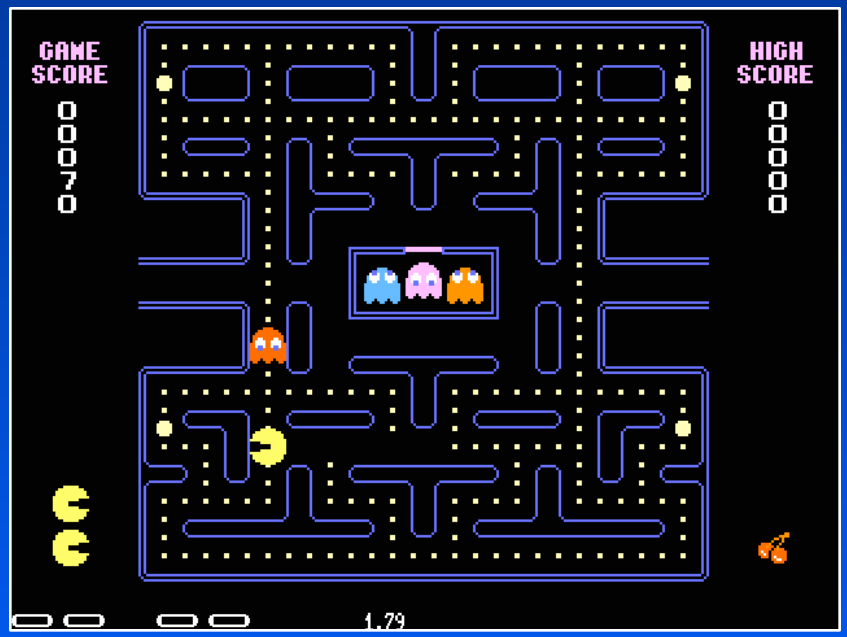
Special Effects

Movies

Compositing

Games

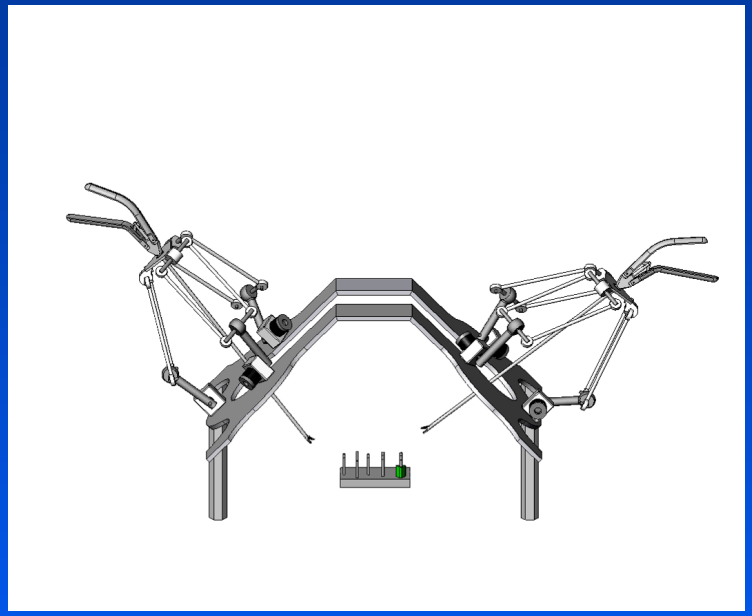
Focus on interactivity

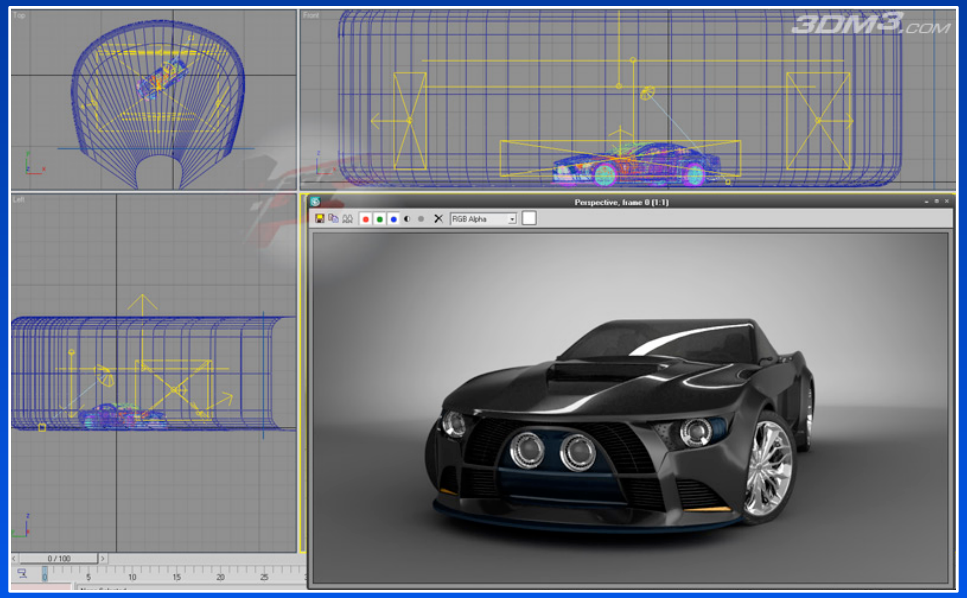


Computer-Aided Design

Precision modeling

Engineering visualization

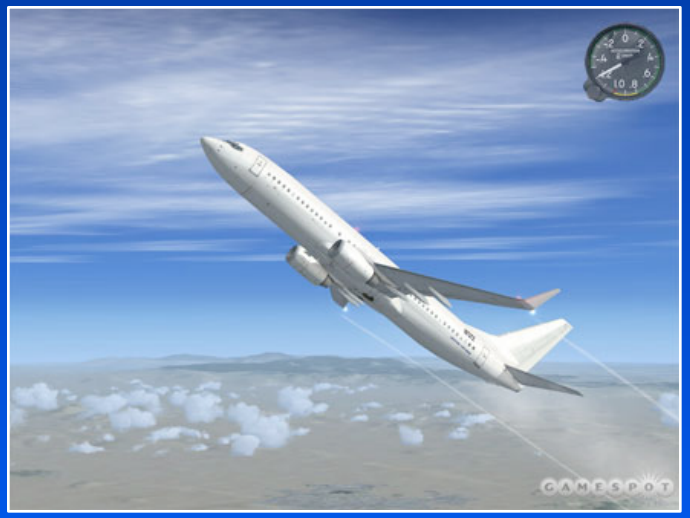


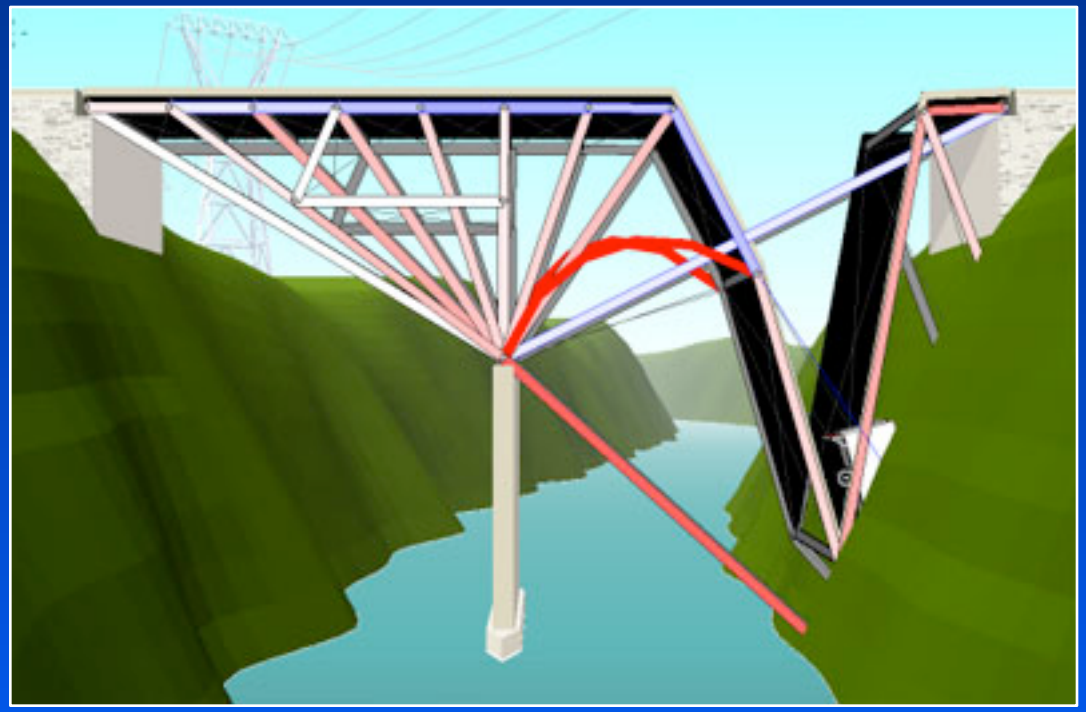


Computer Aided Design

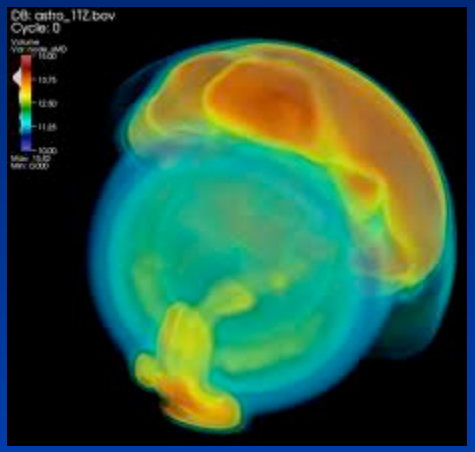
It is not just about visualization

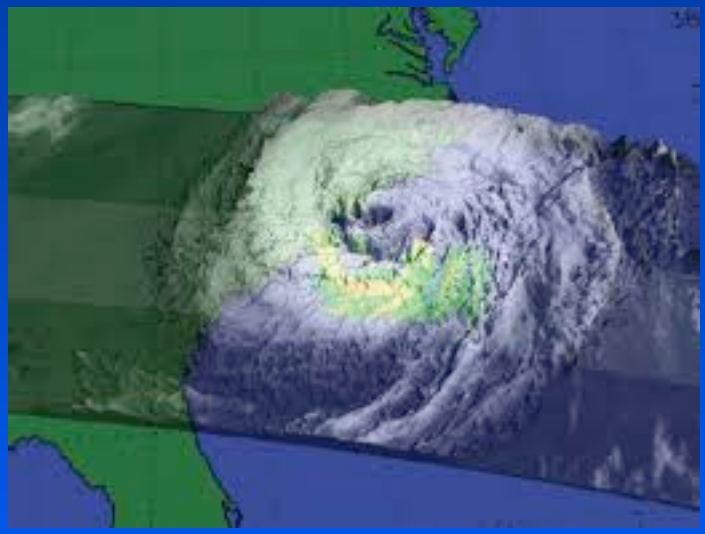
- Simulation is useful





Visualization: Scientific





Visualization: Architectural

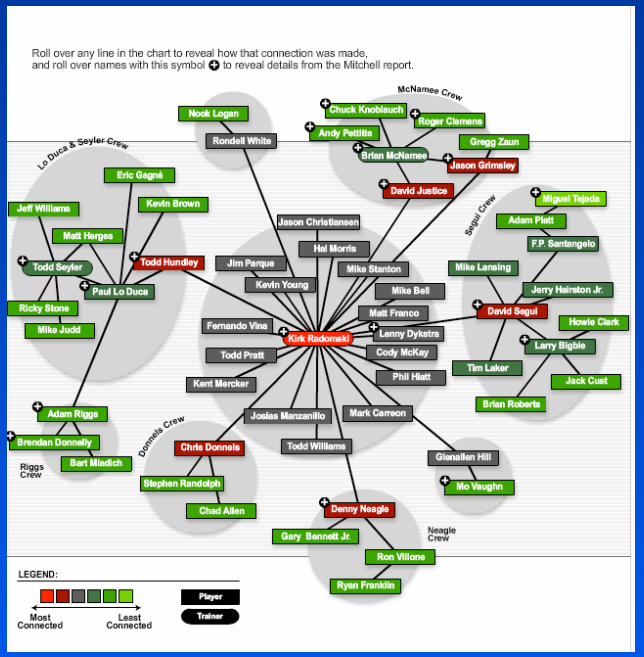
Energy consumption

Noise, lighting,

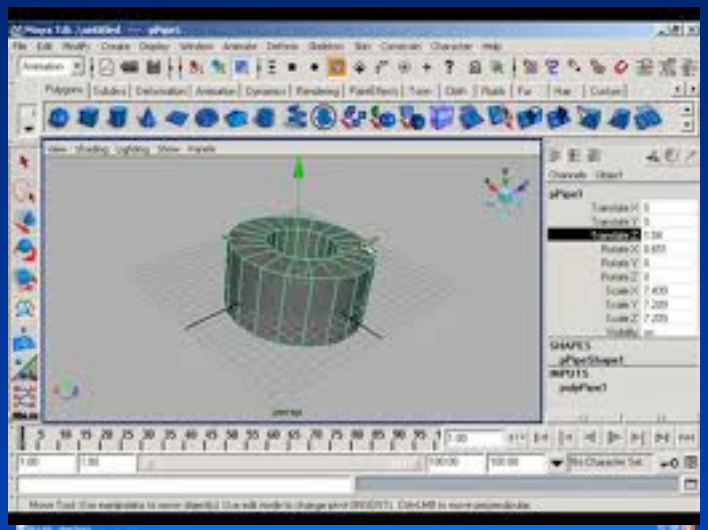
etc

Visualization: info

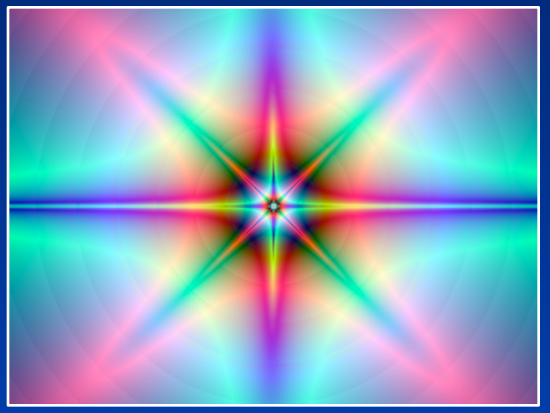
- Geographical Information systems
  - *Maps*
- Personal Information
- Massive dataset visualization

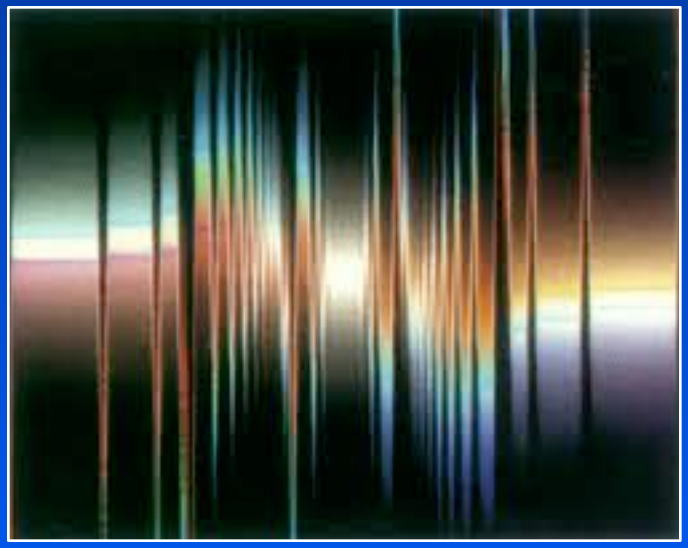


Graphical User Interfaces



Digital Art

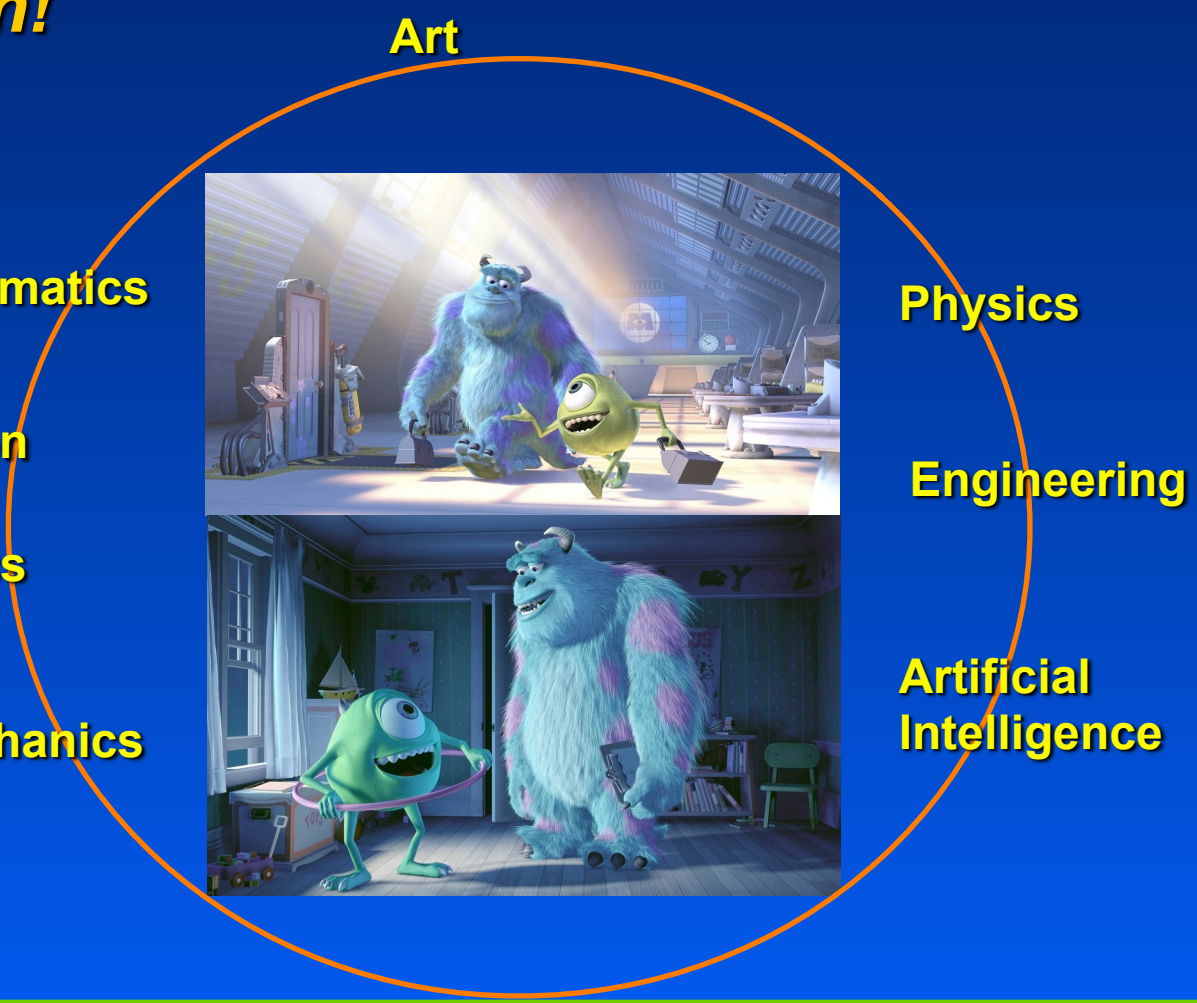




Computer Graphics

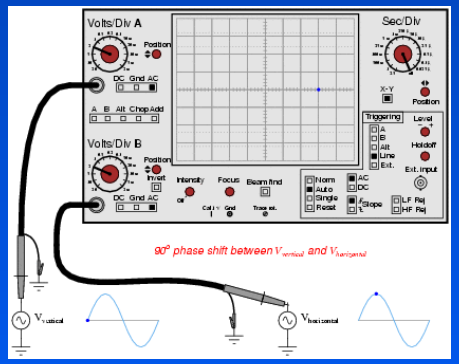
Pictures! Motion!

Interaction!

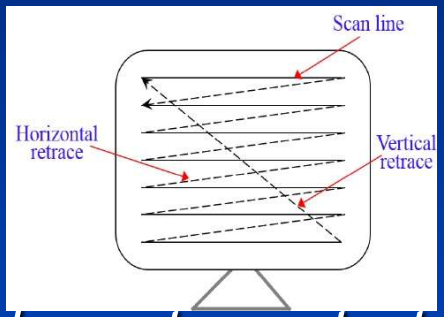


History

- 2000 B.C.
  - *orthographic projection*
- 1400s
  - *Perspective: Italian Renaissance*
- 1600s
  - *coordinate systems: Descartes,*
  - *optics: Huygens*
  - *calculus, physics, optics: Newton*
- 1897 oscilloscope: Braun



History

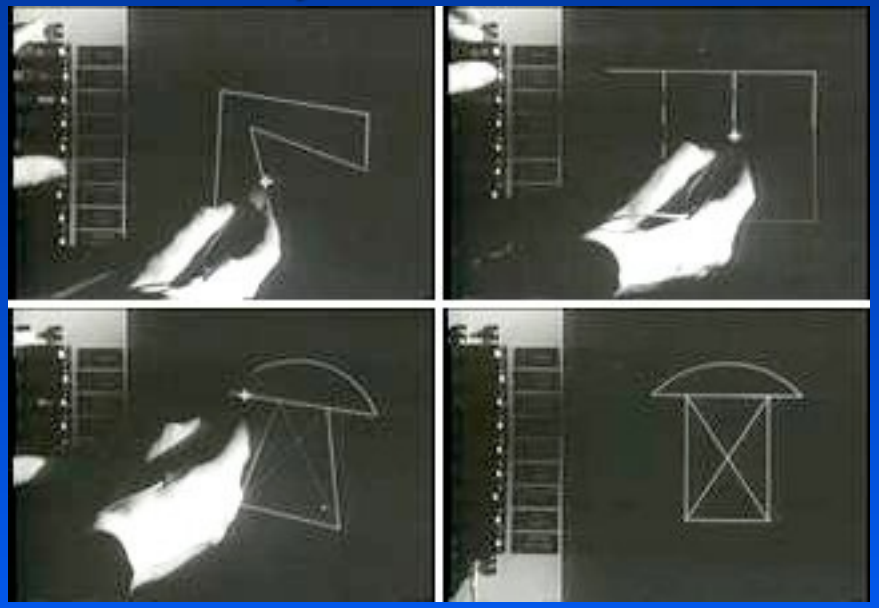


- 1950-1970
  - *computers with vector displays*
- 1966
  - *first true raster display*
- 1993
  - *1200x1200, 500k triangles/sec, 36-bit colour, stereo, texture mapping all at 60Hz*
- 1995
  - *video holography, feature-length CG films*
- Today 4K,8K … video, OLEDS

Genesis of Computer Graphics and Interactive Techniques

A PhD project at MIT in the early 1960s

- Ivan E. Sutherland, 1963
  - *"Sketchpad, a man-machine graphical communication system"*



Quiz

http://www.accad.ohio-state.edu/~waynec/history/timeline.html

- When was the term Computer Graphics first stated?
  - *William Fetter of Boeing coins the term "computer graphics" for his human factors cockpit drawings 1960.*
- Who and when developed the GUI?
  - *GUI developed by Xerox (Alan Kay) 1969*
- When was Tron released?
  - *Disney contracts Abel, III, MAGI and DE for computer graphics for the movie Tron released in 1981.*

Quiz (contd)

- Which is the first animated movie to employ CG?
  - *The Great Mouse Detective was the first animated film to be aided by CG in 1986.*
- When was DOOM released ?
  - *1993.*
- Which was the first totally computer generated movie?
  - *Toy Story 1995*

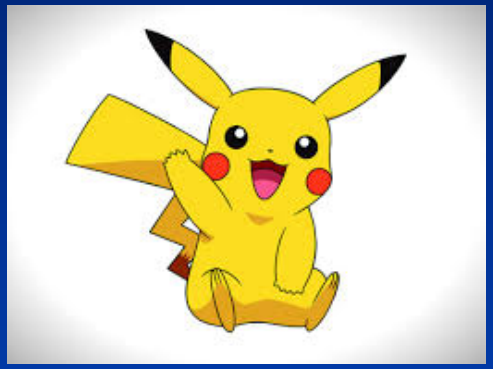
Quiz (contd)

- Which is bigger in gross revenue, the Gaming Industry or Hollywood?
  - *The Gaming Industry.*
- Which is the best selling game of all time?

Top games (for fun)

Grossing

- Super Mario
- Pokemon

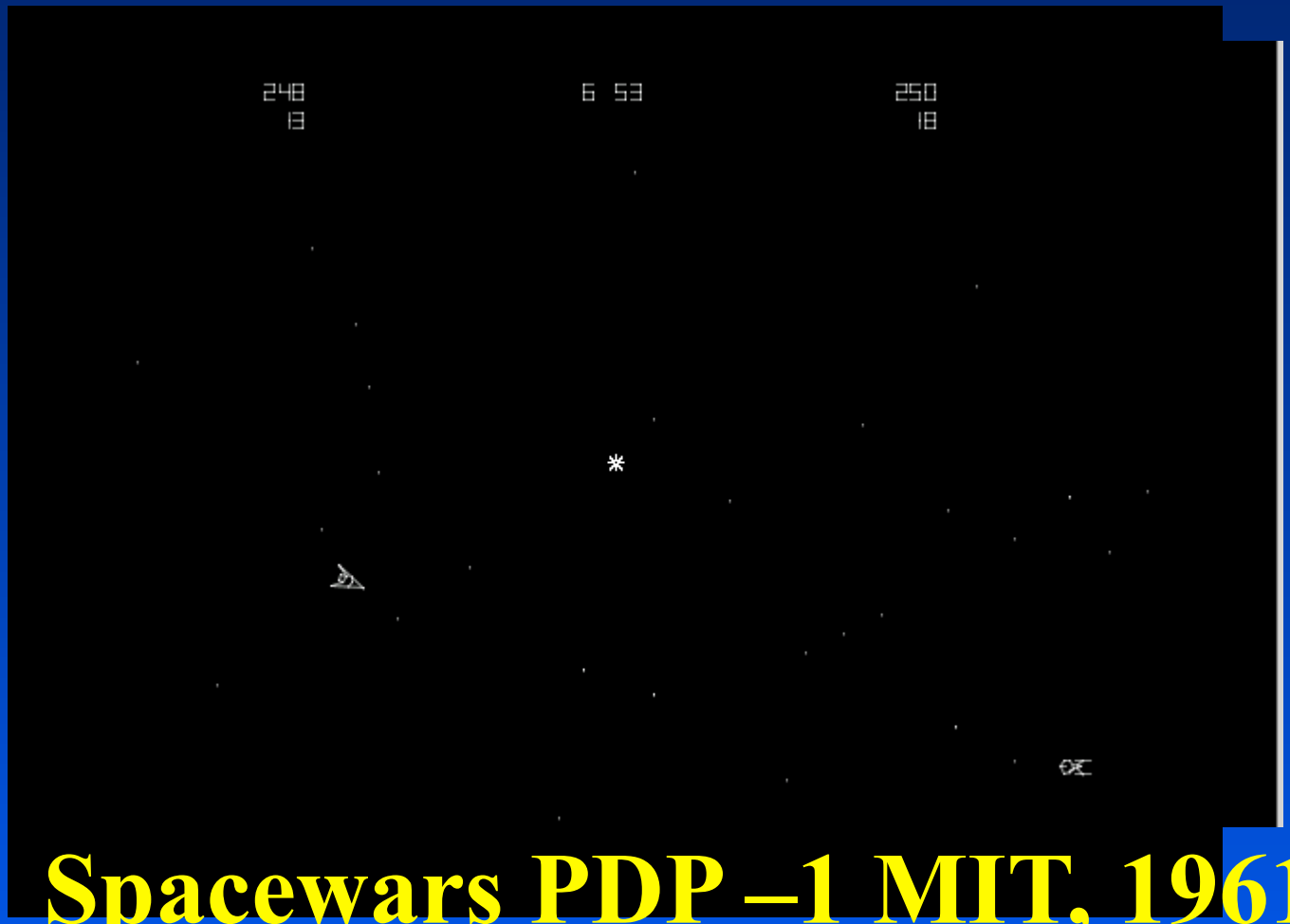


Most copies sold

- Tetris 495 Million

Source: Wikipedia

First computer game?



Spacewars PDP –1 MIT, 1961

Bottom line: Synthesize Images/Videos

What is an image or video

- An array of pixels (one or more numbers)
- A video is a sequence of images

How do we synthesize an image?

- From mathematical models to coloured pixels

Models of

- Objects in the world, light sources, materials
- Processes (e.g. Illumination, sampling)
- Imaging device (eye, camera)

Basic Elements

Modeling

Rendering

Animation

Interaction

Basic Technical Elements

- Modeling
  - *How do we model (mathematically represent) objects?*
  - *How do we construct models of specific objects?*
- Animation
  - *How do we represent the motion of objects?*
  - *How do give animators control of the motion?*
- Rendering
  - *How do we simulate the real-world behavior of light?*
- Interaction
  - *How do we enable humans and computers to interact?*
  - *How do we design human-computer interfaces?*

Modeling

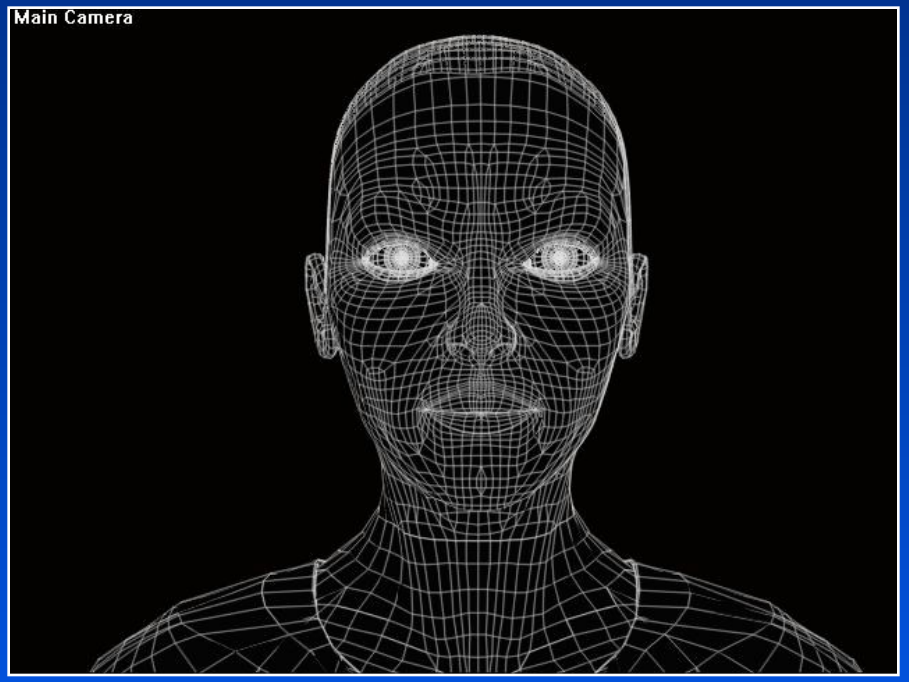
Primitives

- 3D points
- 3D lines and curves
- surfaces (BREPs): polygons, patches
- volumetric representations
- image-based representations

Attributes

- Color, texture maps
- Lighting properties

Geometric transformations



Rendering

Viewing

Visibility

Simulating light propagation

- Reflection
- Absorption
- Scattering
- Emission
- Interference

Animation

- *Keyframe, motion capture*
- *Physics-based animation*
- *Autonomous motion planning*

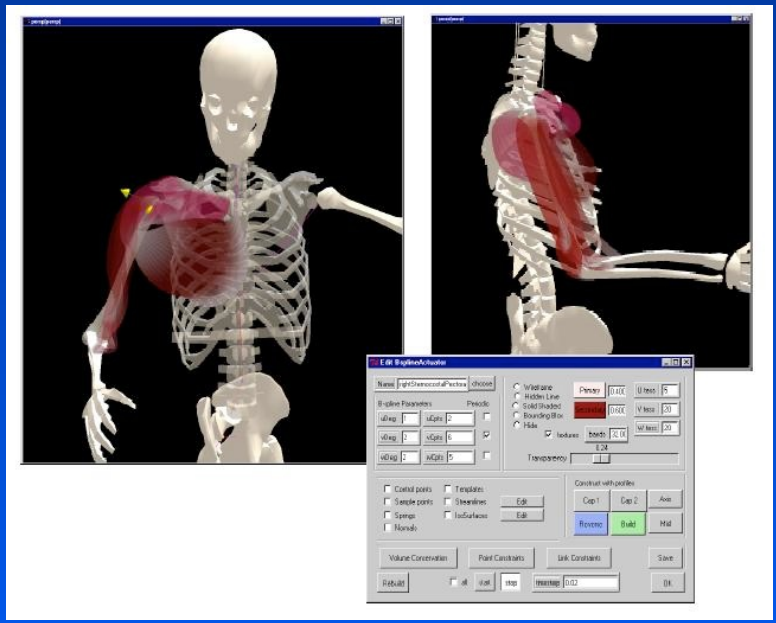
Interaction

Tools

- Modeling, rendering and animation

Input/Output Devices

Courtesy of Victor Ng-Thow-Hing



Examples of Research Results and Problems in Computer Graphics

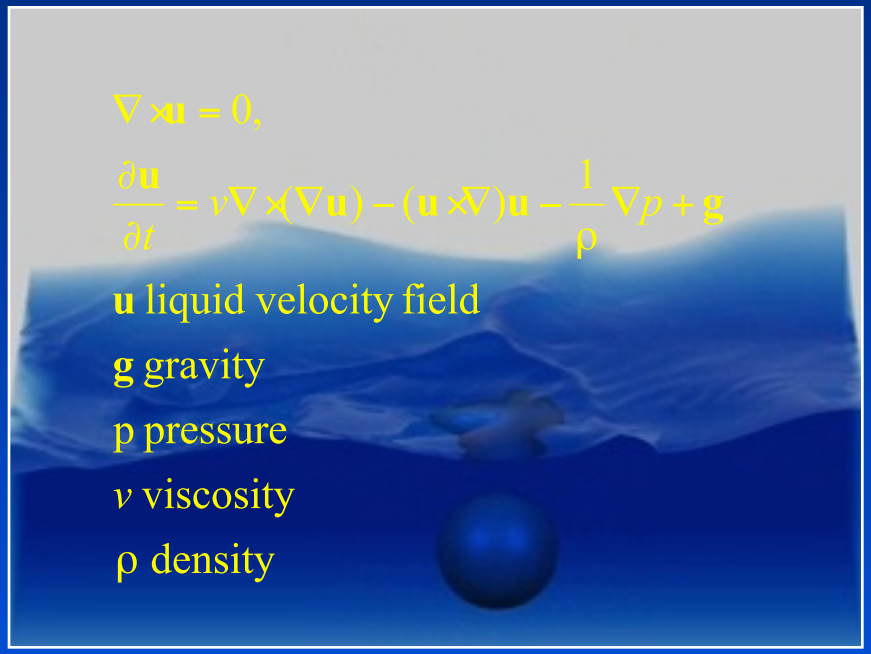
Fluid Simulation

Modeling

- Incompressibility
- Viscocity

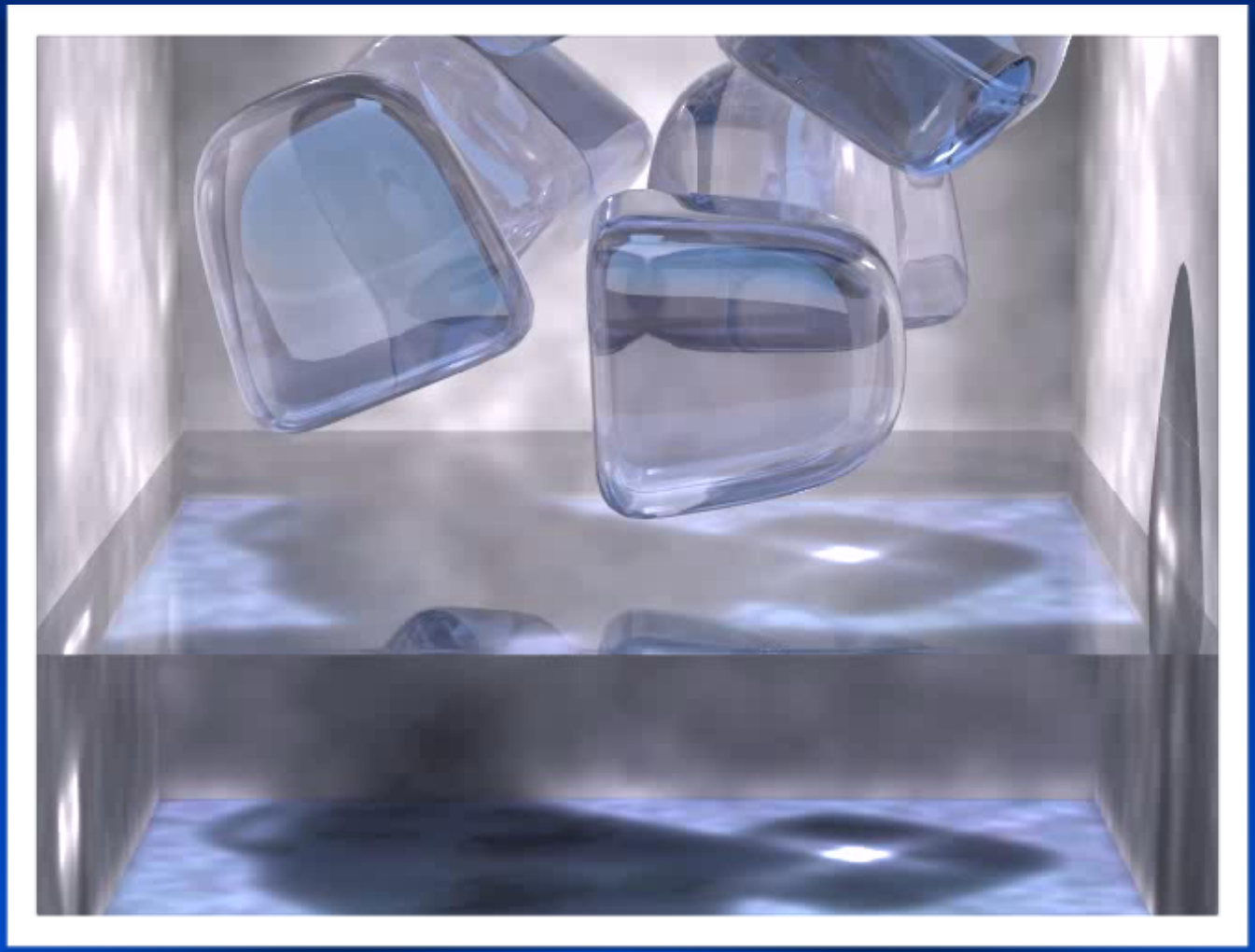
Navier Stokes

Level Sets



Phase transition

Courtesy of the Stanford Graphics group



Character Animation

Motion control

- Motion capture
- Physics-based
- AI

Facial Animation

Visual Speech

Expression control

Attention control

Hair simulation

Crowd simulation

[Shawn Sing, Mubbasir Kapadia, Glenn Reinman, Petros Faloutsos]

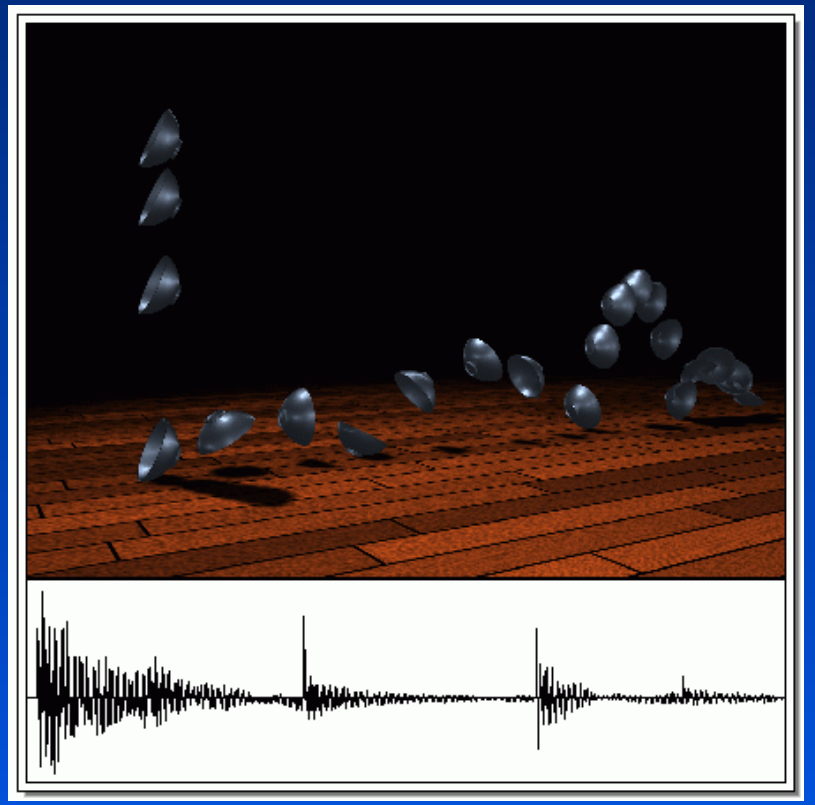
Sound simulation

Physics-based generation

$$\frac{\partial^2 p}{\partial t^2} = c^2 \nabla^2 p$$

p acoustic pressure

c speed of sound in the fluid



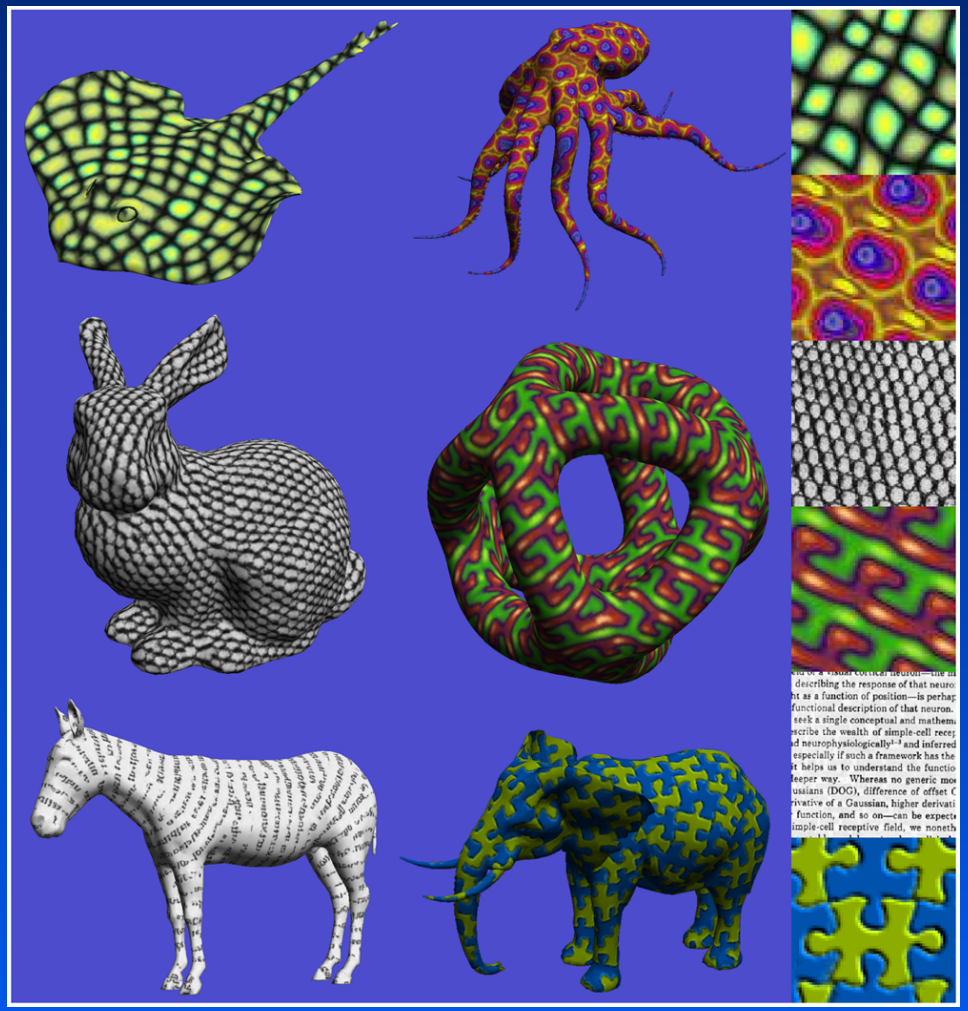
Sound simulation and rendering

Physics-based generation

Texture synthesis

Vector fields

Multilevel synthesis



Great! What are we going to do?

The basics and

… some animation.

EECS 3431

Introduction to Computer Graphics

- Introduction to (mostly interactive) graphics
- Software
- Hardware
- Applications

Shader based WebGL compatible with

- OpenGL ES 2.0
- OpenGL 3.3

https://eClass.yorku.ca

A computer graphics model of the instructor

(copyright PIXAR/DISNEY)

Prerequisites

- Good programming skills:
  - Javascript, C, C++
- Basic data structures
  - *Link lists*
  - *Arrays*
  - *Structures or Classes*
- Basic Geometry
- Basic linear algebra

Course Load

Tentative Marking scheme - Official on eClass

- Assignment 1: 10 %
  - *Predefined scene modelling and animation*
- Assignment 2: 15 %
  - *TBA*
- Assignment 3: 20 %
  - *Raytracing*
- Midterm: 20 %
- Final: 40 %

Selected Policies - For complete list see web page

Late submission policy

- Maximum 3 days with 10% penalty per 24 hours.
- Penalty computed at increments of 15 minutes.
- NO LATE assignments if original deadline extended.

Academic dishonesty policy

- York's policy is very clear on cheating (e.g. plagiarism) of any kind. All cases must and will be referred to the Dean of Students.

Zero mark policy

- Assignments that do not compile get ZERO marks.
- Assignments that produce nothing meaningful get ZERO marks.

Facts about the class

Challenges

- Too much material
- Fast pace
- Tough third project
- Tough exams
- A lot of programming
- A lot of math
- Hard to debug 3D scenes in 2.5 D

Benefits

- Skill building
- CV building

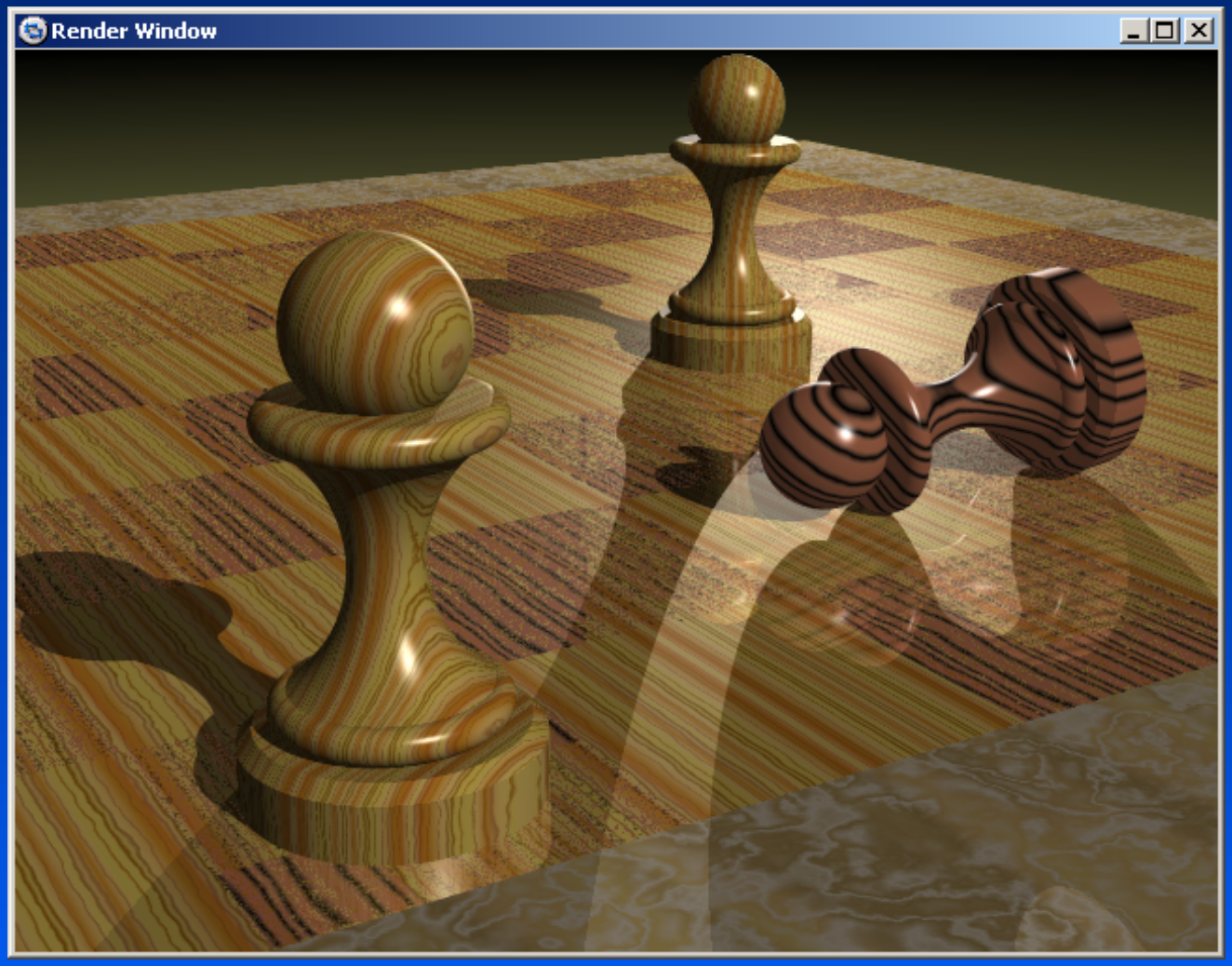
Advice

- Start the assignments EARLY!!
- Get HELP from us with the assignments EARLY!
- Do NOT do more than you are asked to unless you are done with the required part of the assignment.
- You will NOT get more marks for additional work.
- Refresh vector and matrix algebra and keep up with the math.

Resources

- Main OpenGL web site: www.opengl.org
  - *Documents*
  - *Coding resources and examples.*
- The OpenGL Reference guide - Blue Book
- The OpenGL Programmer's guide - Red Book
  - *The definitive reference*
  - *OpenGL 4.5*
- OpenGL ES 2.0 Programming Guide
- Javascript resources: http://www.w3schools.com/js/

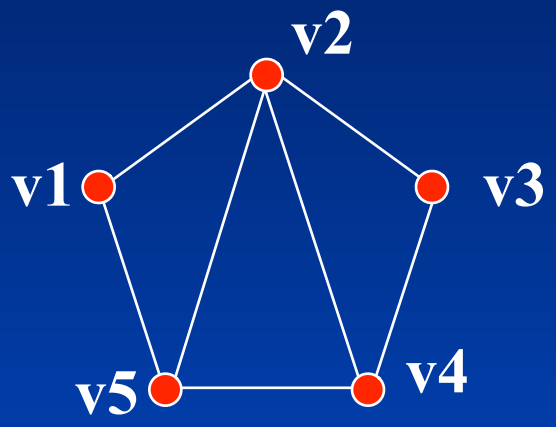
Topics in more detail



Interactive 3D Graphics

Basic primitives

- Lines
- Triangles
- Triangle meshes
- Attributes: color, texture, normal vectors et al.



These primitives go through a standardized pipeline

Why only triangles?

The graphics pipeline

Why a pipeline

- Well defined stages
- Parallelism
- Software and Hardware

Radeon RX by AMD

- Compute Units: 36
- Stream Processors: 2304
- Peak Pixel Fill Rate: 42GP/sec
- Texture units: 144
- Max Performance: 6.2 TFLOPS

Radeon RX 580 by AMD

Graphics pipeline (fixed or not)

Modelling

Viewing (Projection)

Illumination

Clipping

Visibility

Rasterization

Per Vertex and Per Pixel operations

Vector and matrix calculus

Vector spaces

Matrix algebra

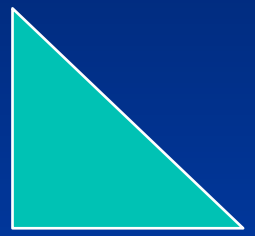
Coordinate systems

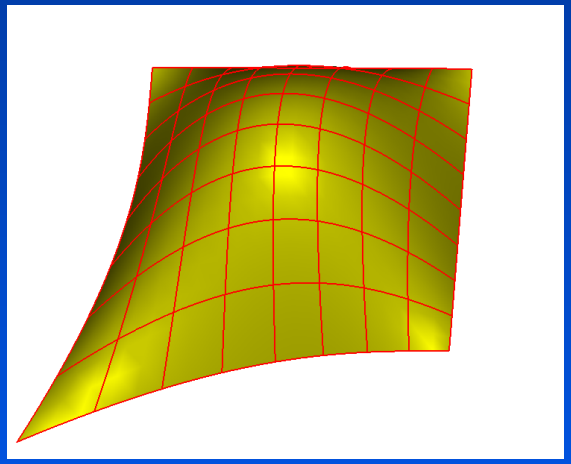
Affine transformations

Modeling

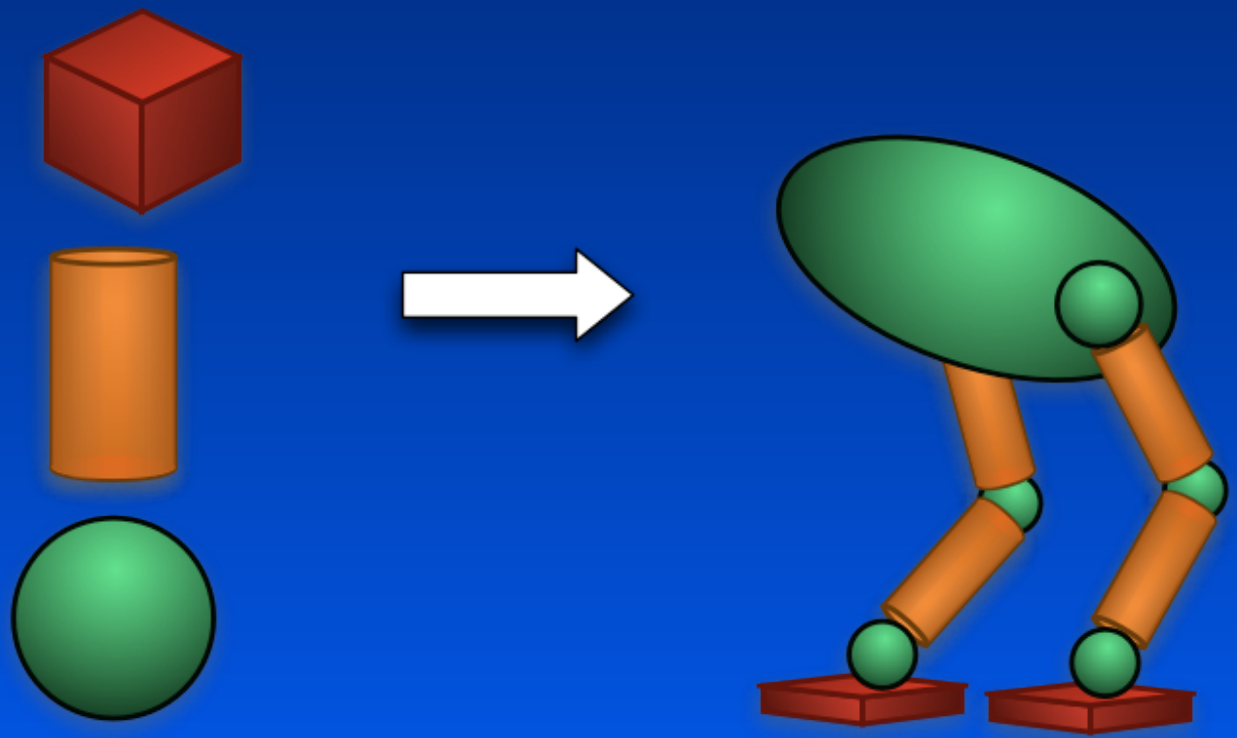
Geometric Primitives

- Triangles (polygons)
- Parametric surfaces
- Implicit surfaces

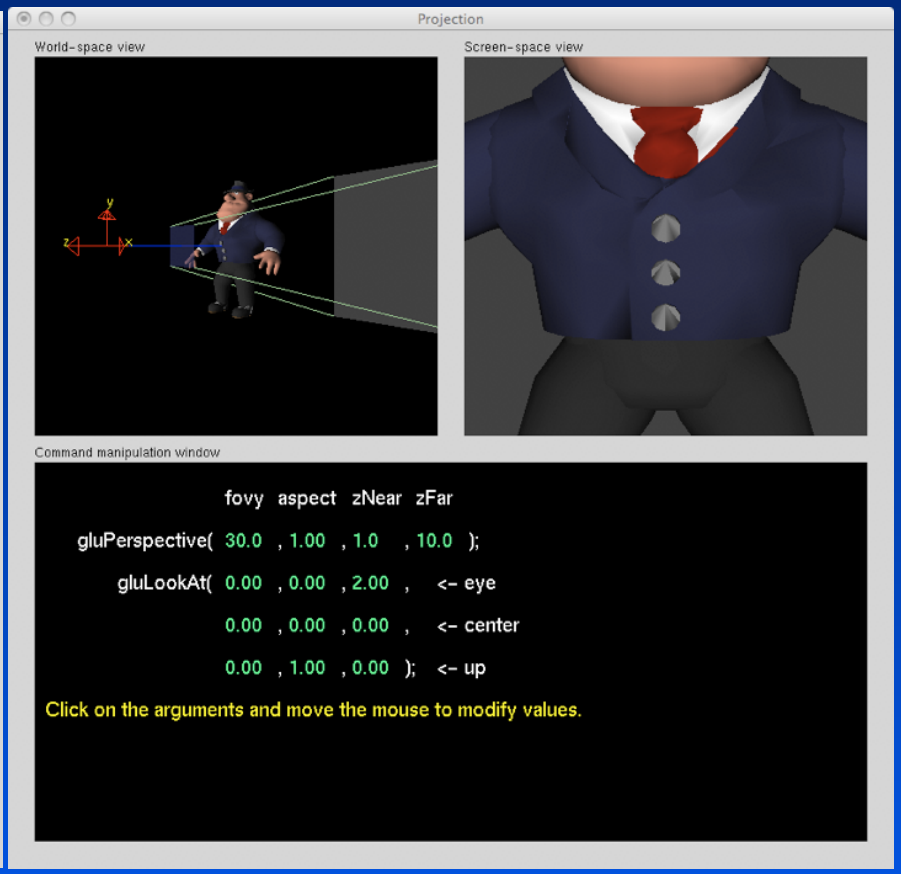




Modeling Transformations



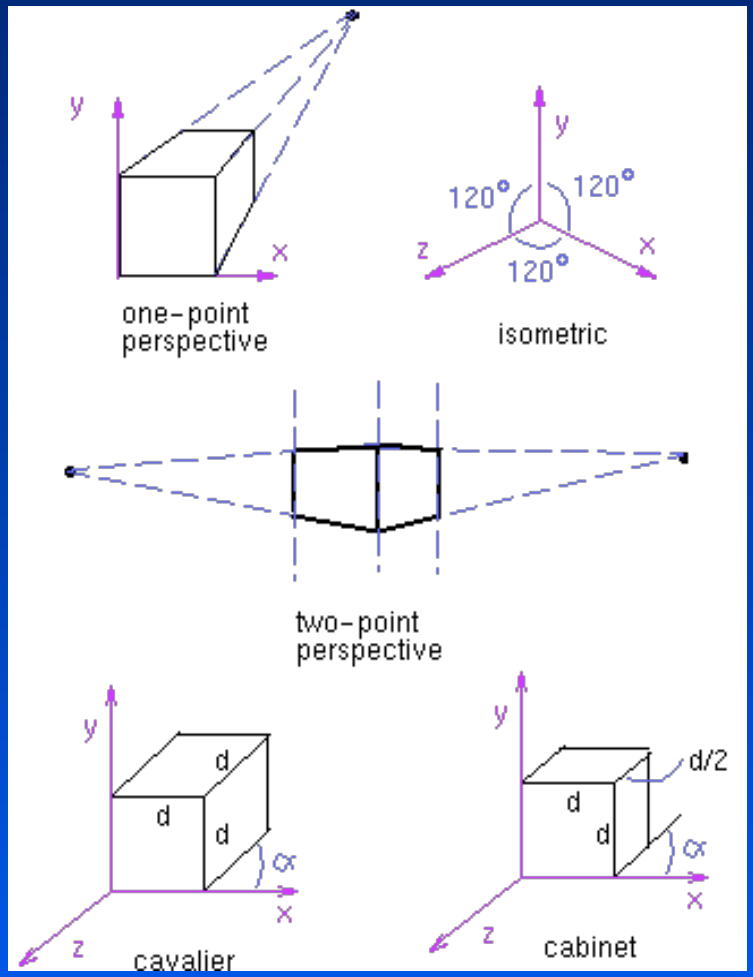
Viewing and Clipping



Viewing - Projection

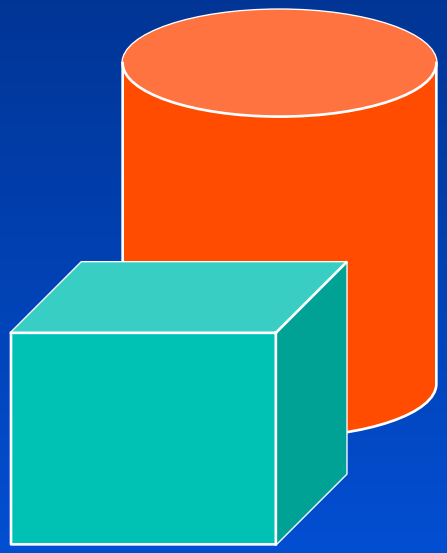
Perspective

Orthographic



Visibility

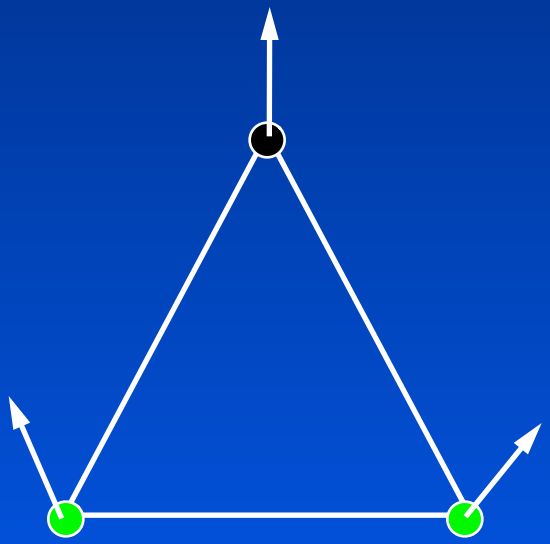
Resolve occlusions (efficiently)

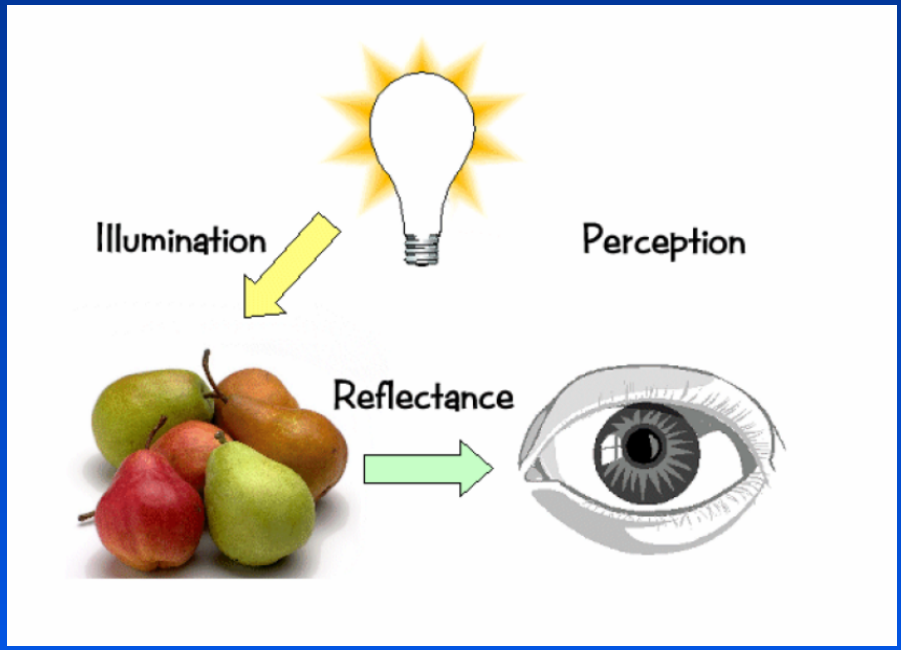


Illumination

Compute normals and colour

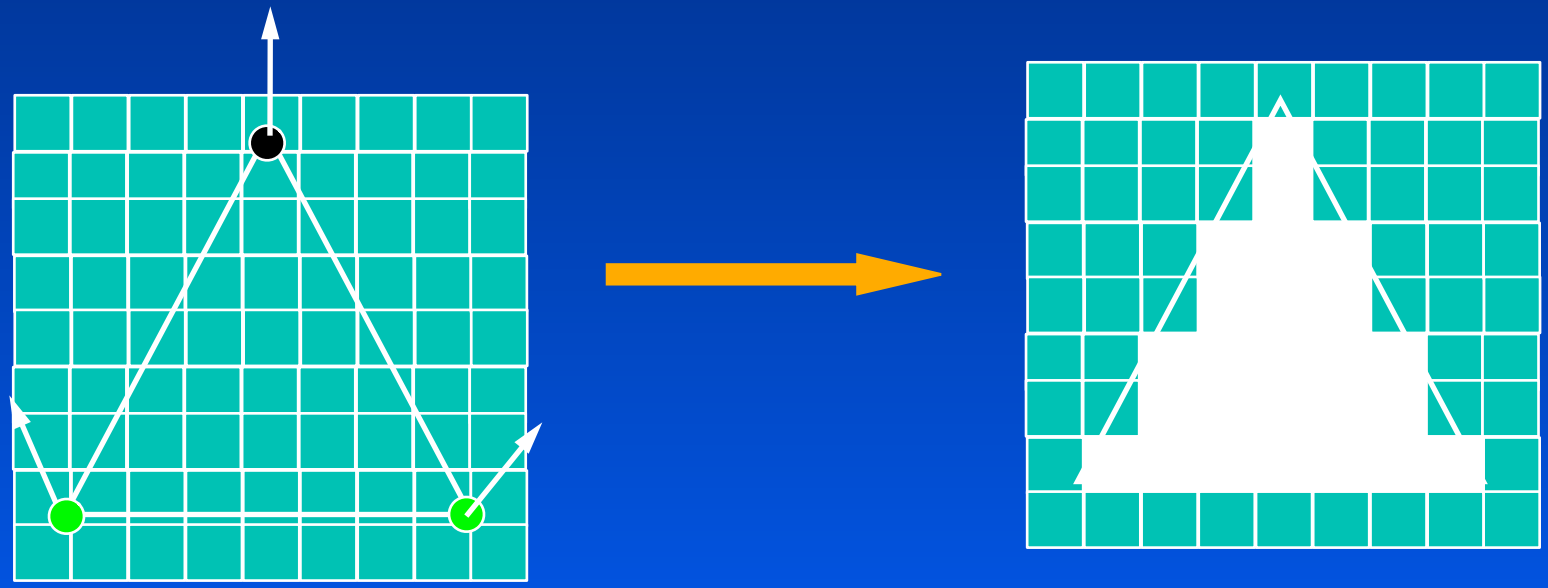
- Per vertex
- Per pixel





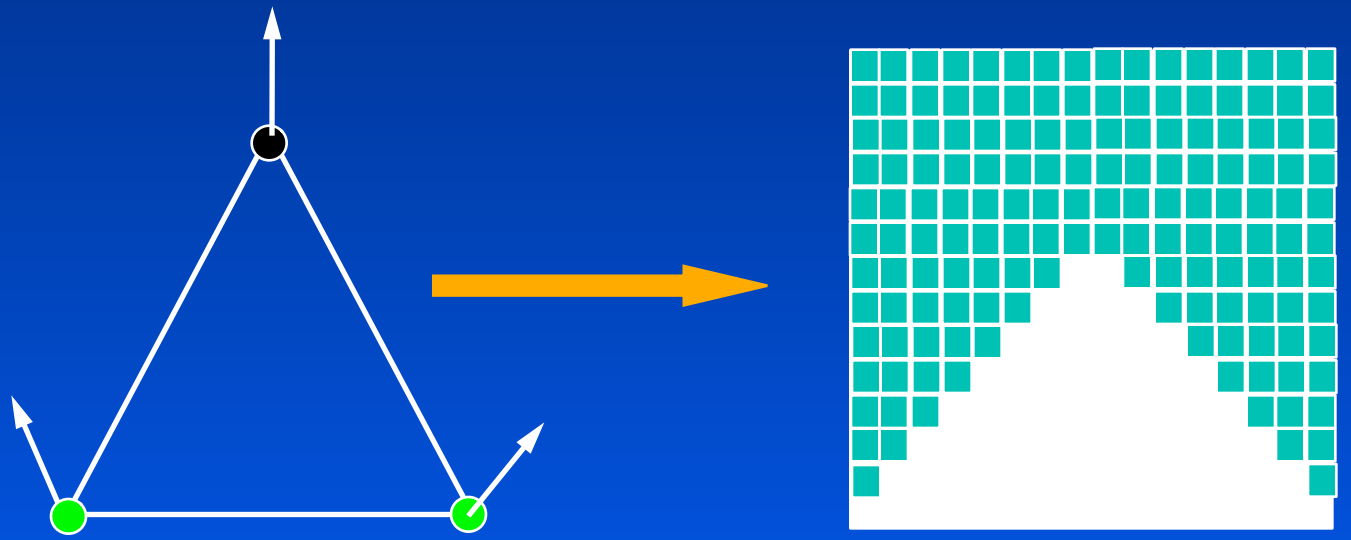
Rasterization

Projected model to actual pixels



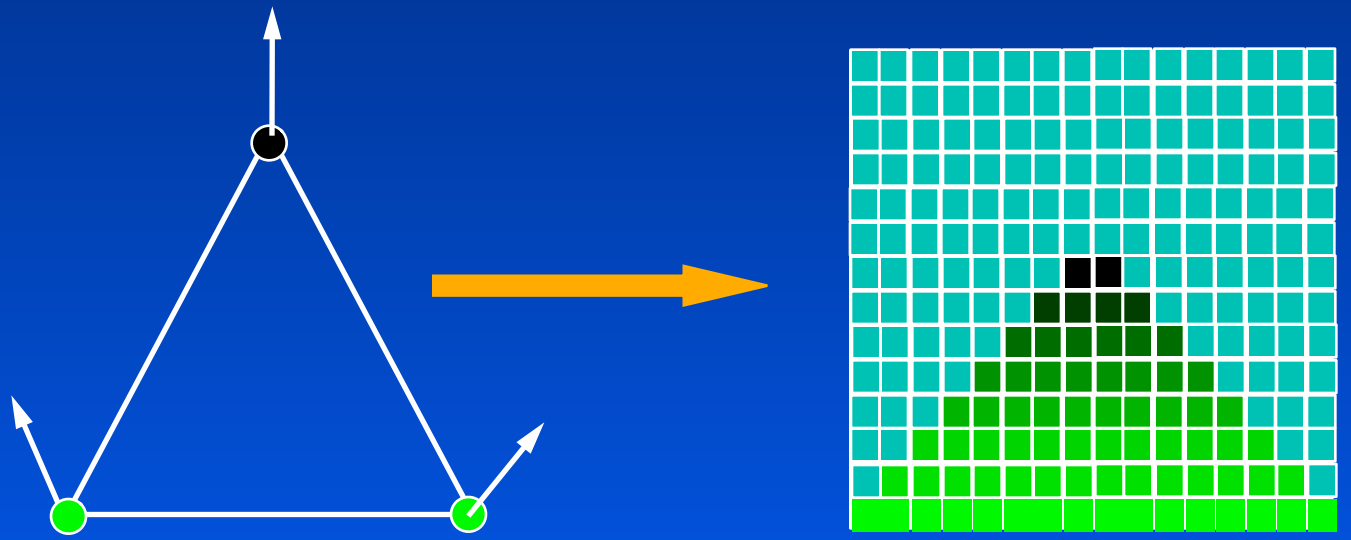
Rasterization (size of pixels matters!!)

Projected model to actual pixels

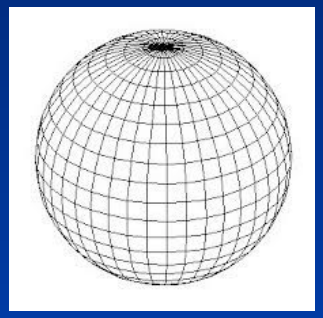


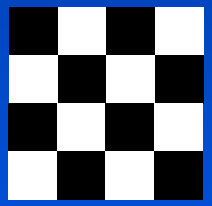
Shading

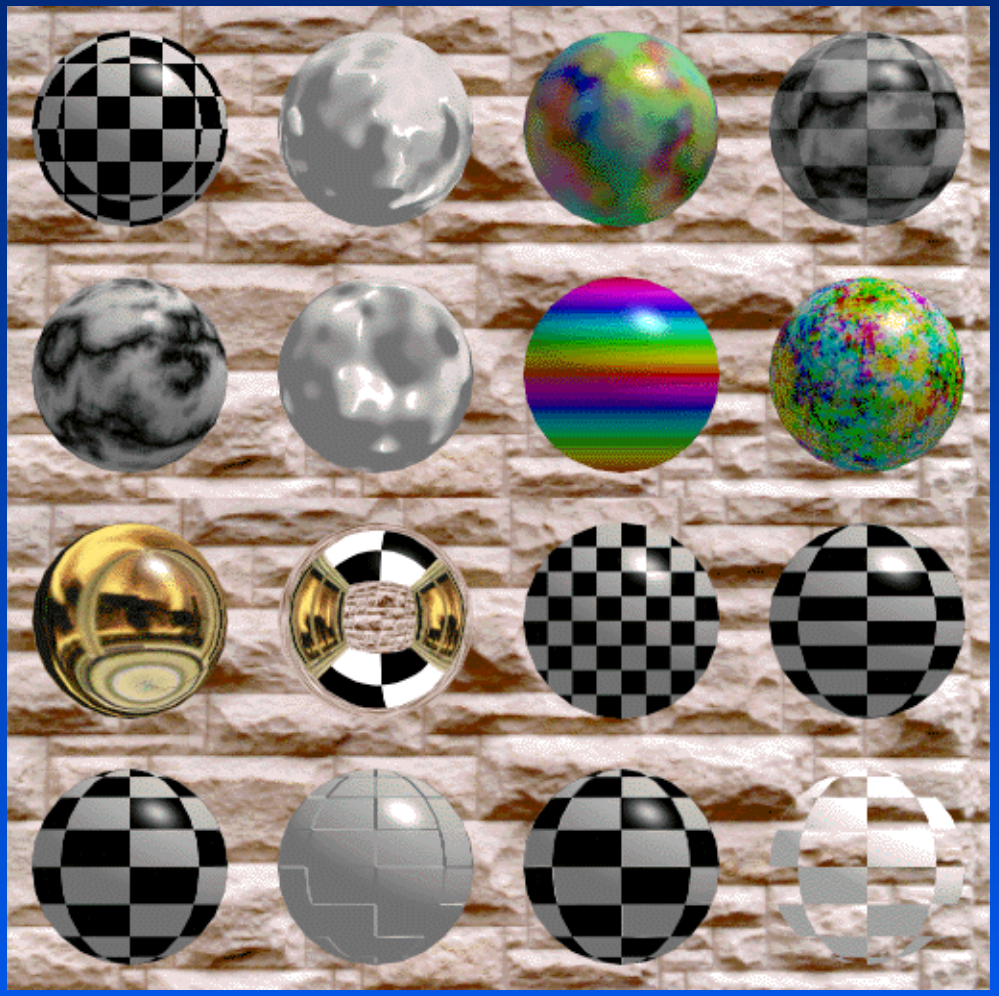
Projected model to colored pixels



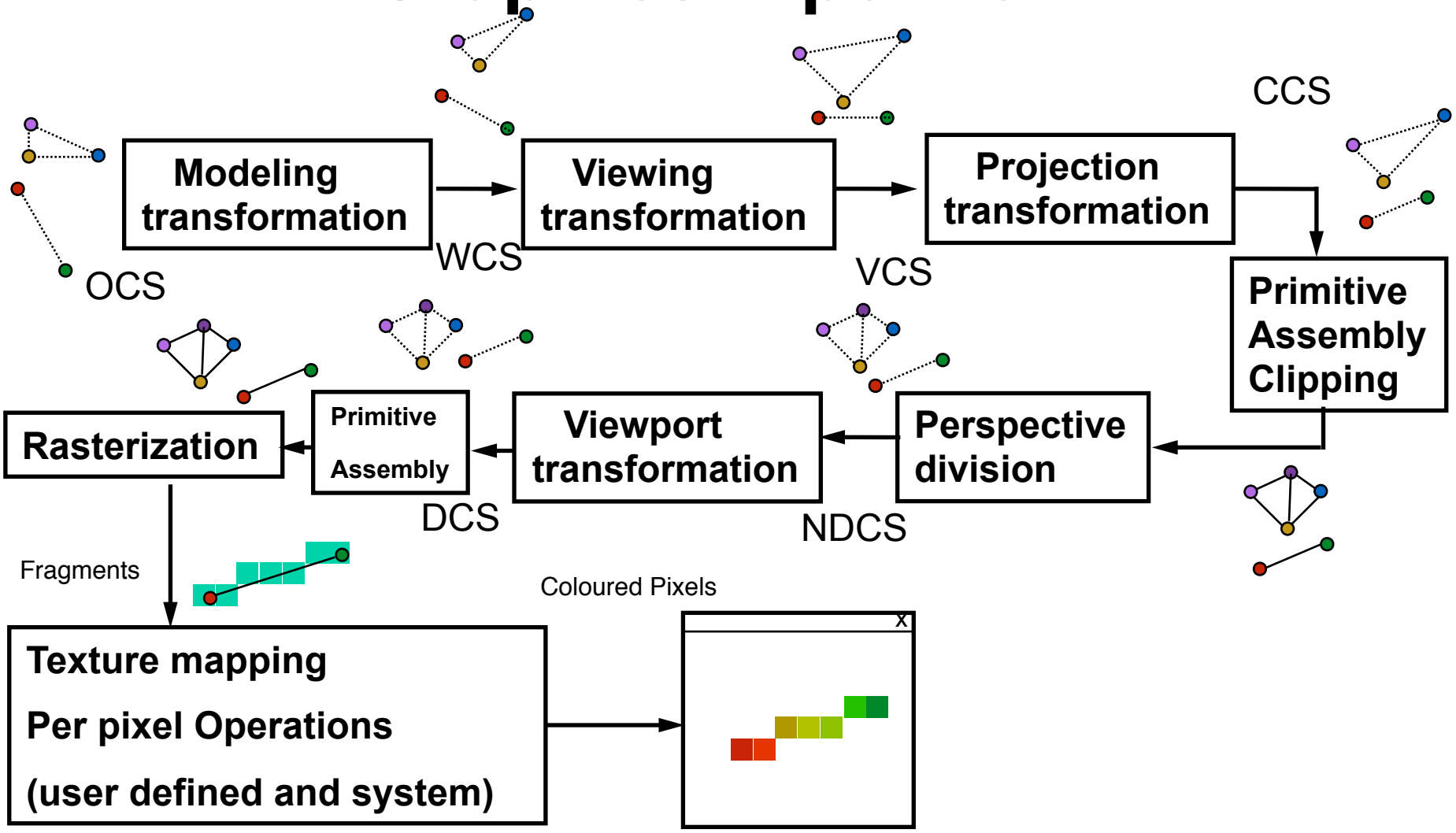
Texture Mapping







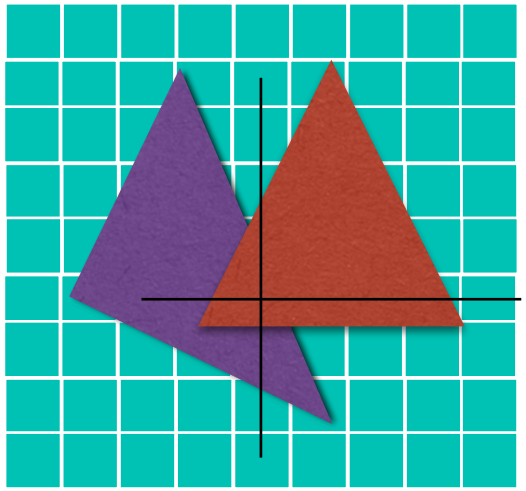
Graphics Pipeline



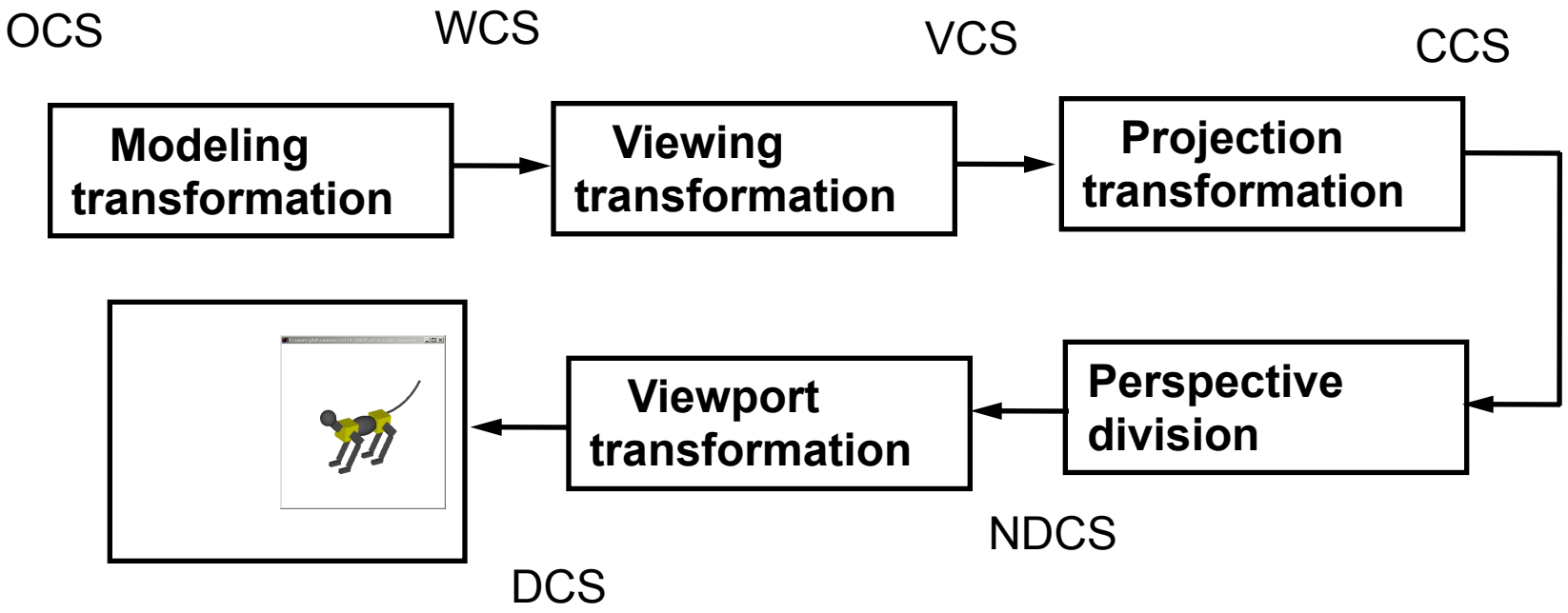
Z-buffer algorithm - Depth Test

After a triangle has been projected on the screen and during rasterization, the larger a point's z the further from the viewer

```
for all i,j {
  Depth[i,j] = MAX_DEPTH
  Image[i,j] = BACKGROUND_COLOUR
}
for all polygons P {
        for all pixels in P {
                if (Z_pixel < Depth[i,j]) {
                        Image[i,j] = C_pixel
                        Depth[i,j] = Z_pixel
                }
        }
}
```

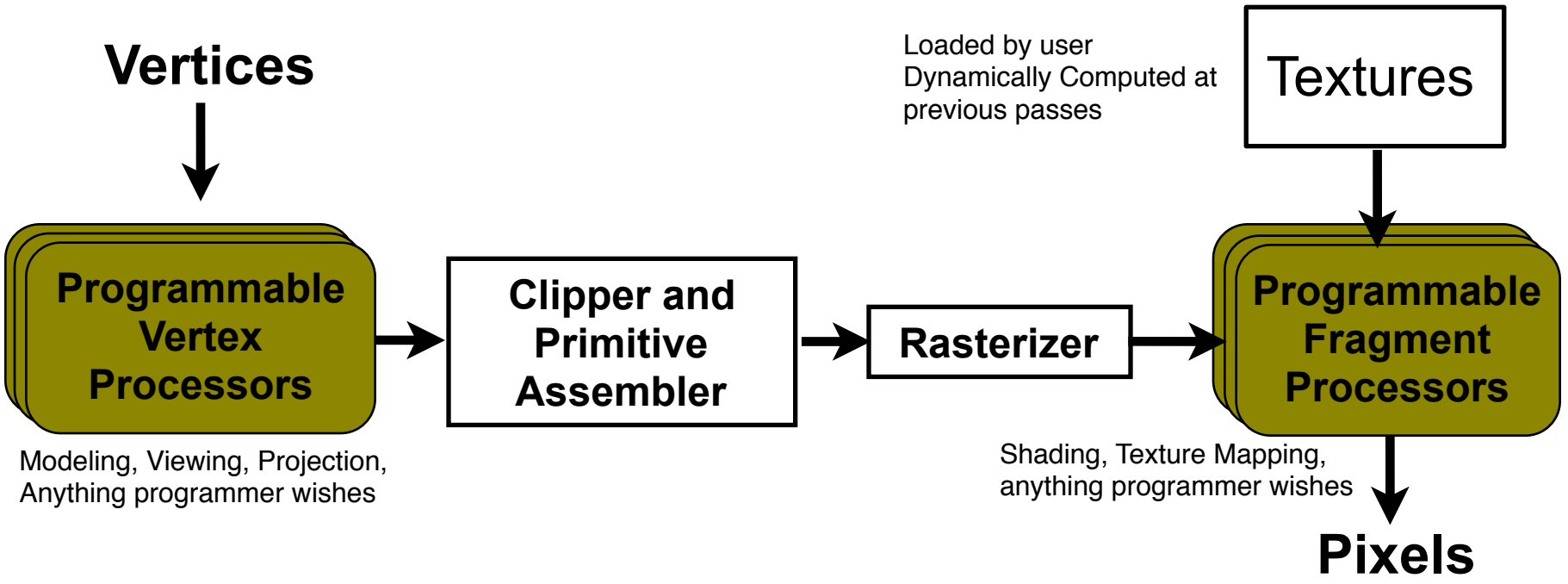


Shader Based Graphics Pipeline



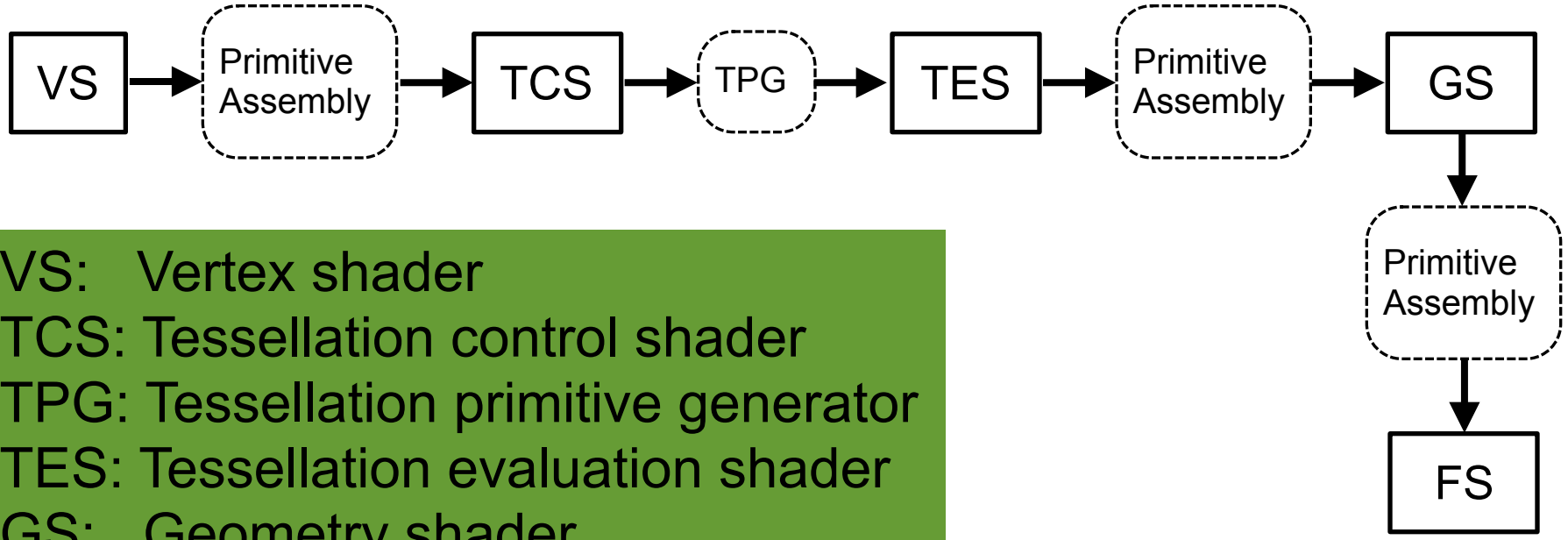
Rasterization into pixels (fragments)
Texture mapping

Shader-based Graphics Pipeline



- Streaming SIMD architecture
- However, it is still useful to follow conceptually and programmatically the old stages

Pipeline with all shaders (aside)

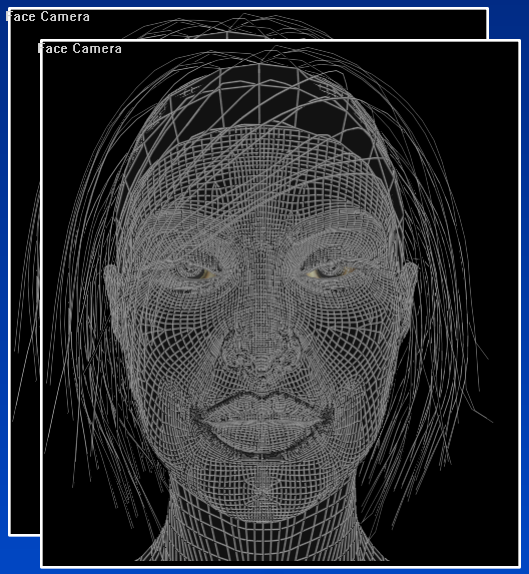


VS: Vertex shader
TCS: Tessellation control shader
TPG: Tessellation primitive generator
TES: Tessellation evaluation shader
GS: Geometry shader
FS: Fragment shader

Tessellation shaders and geometry shaders are optional

Animation

Modeling



$$Mesh = \mathbf{R}\sum_{i=1}^{n} w_i B_i$$

Real-time Rendering

$$Picture = f(Mesh, Materials, Lights)$$

Motion Synthesis

$$Mesh(t) = \mathbf{R}(t)\sum_{i=1}^{n} w_i(t) B_i$$

Modeling: Define an object and a set of parameters, e.g [R,w_i]

Rendering: Apply lights and compute an image

Motions synthesis: Compute [R(t), w_i(t)] to produce motion

Graphics system

Input devices

Rendering system

Output devices

Input devices

Keyboard

Mouse

Light Pen

Game controller

Pressure devices and sensors (Touch screens)

Data glove

Other sensors

Rendering system

Software

- Interface
- Primitives
- Techniques

Hardware

- CPUs
- Discreet Graphics Card or Integrated with CPU

Output devices

Output devices

Monitor

- CRT, Plasma, LCD, LED

Printer

- 3D Printers

Projectors

Plotter

Head mounted display

Cell Phones

Tablets

Other

Courtesy of Steve Mann

Courtesy of the Vesa Lab

Summary: Basic Elements of Computer graphics

Math

Modeling

Rendering

Animation

Interaction

Hardware

Example: Oktapodi

Winner of the SIGGRAPH 2008 Computer Animation Festival

- best in show, and
- audience's choice

awards !

Oktapodi

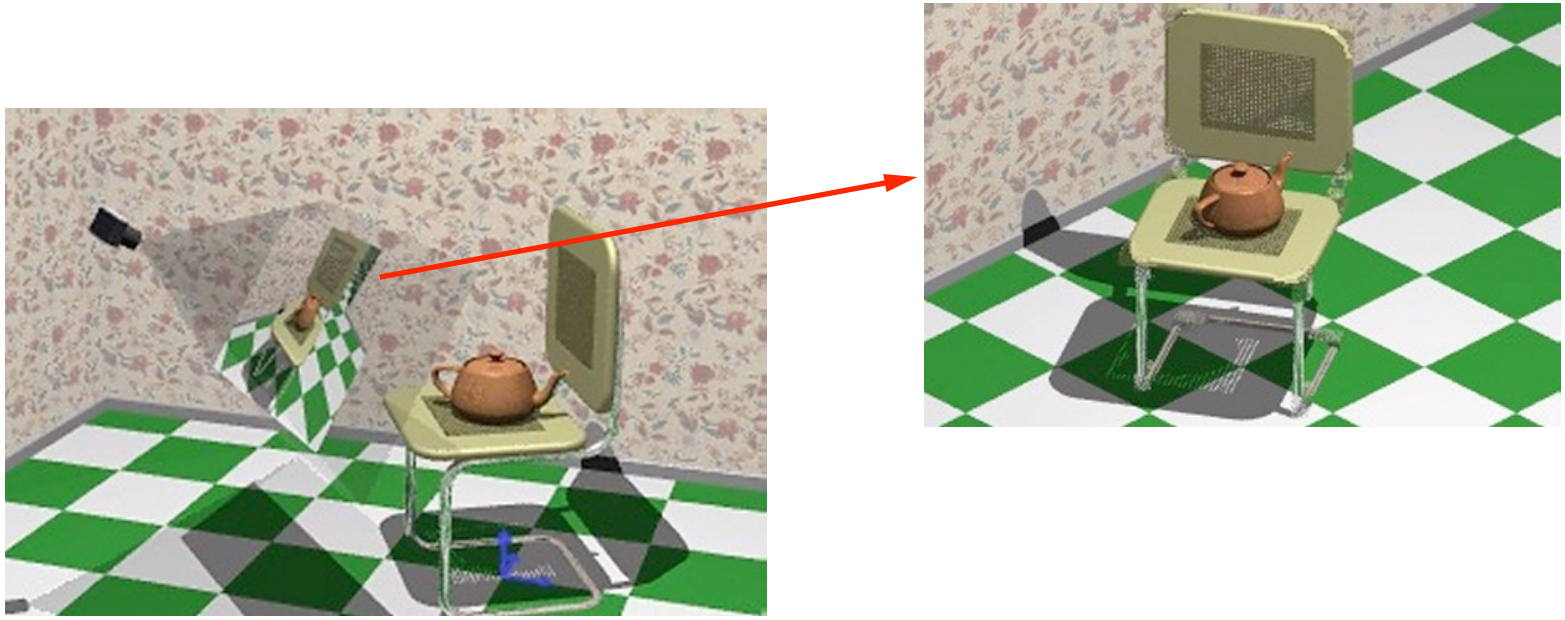
Image Formation

Ed Angel

Professor of Computer Science, Electrical and Computer Engineering, and Media Arts

University of New Mexico

Taking a snapshot of a 3D Scene



Imaging

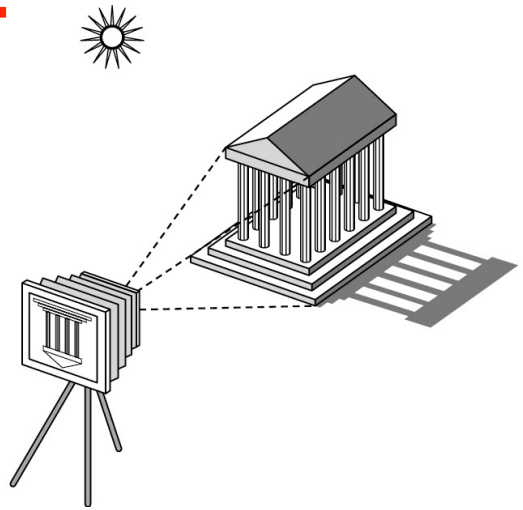
- Fundamental imaging notions
- Physical basis for image formation
  - Light
  - Color
  - Perception
- Synthetic camera model
- Other models

Image Formation

- 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)

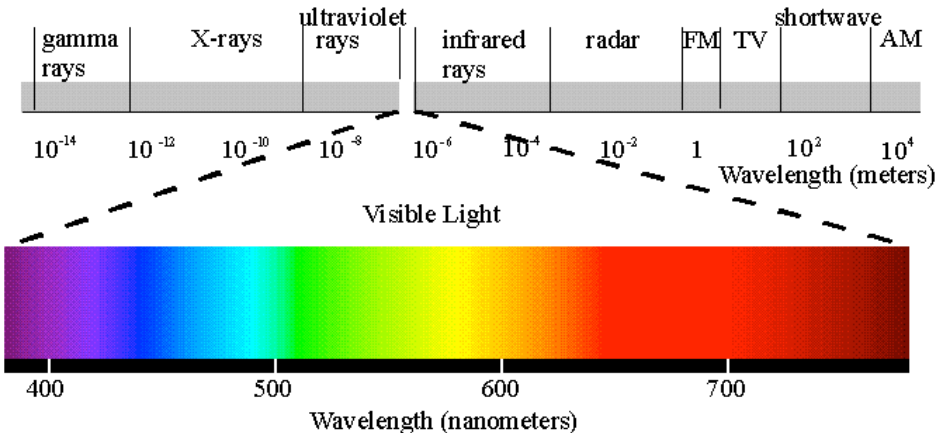


- Attributes that govern how light interacts with the materials in the scene
- Note the independence of the objects, the viewer, and the light source(s)

The University of New Mexico

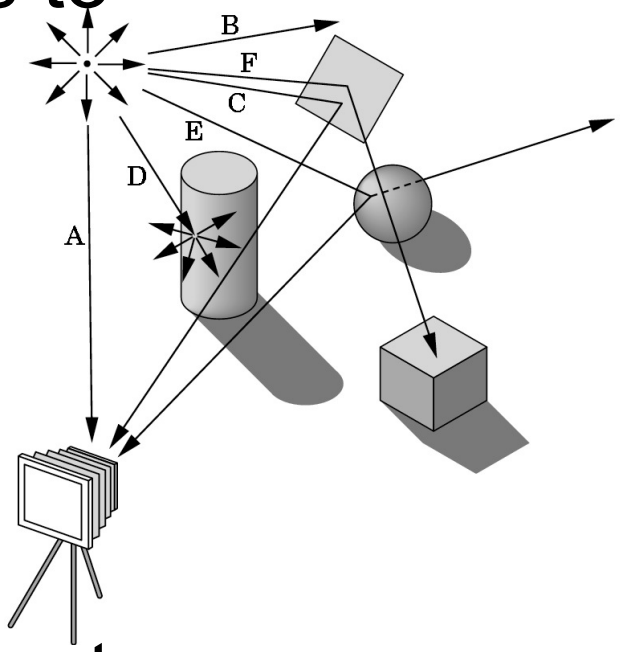
Light

- *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



Ray Tracing and Geometric Optics

One way to form an image is to follow rays of light from a point source finding 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.

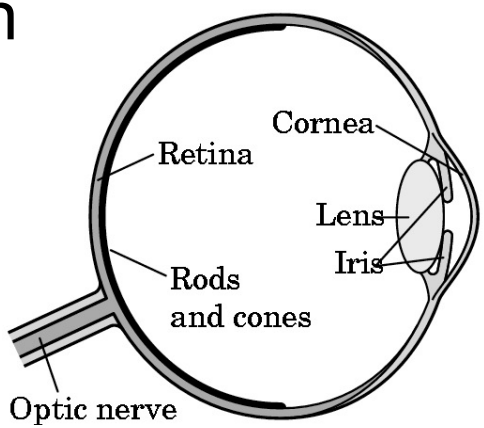


Luminance and Color Images

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

Three-Color Theory

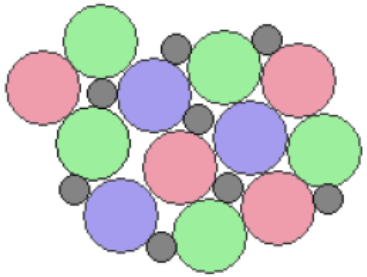
- Human visual system has two types of sensors
  - Rods: monochromatic, night vision
  - Cones
    - Color sensitive
    - Three types of cones
    - Only three values (the *tristimulus* values) are sent to the brain



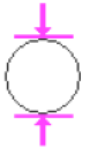
- Need only match these three values
  - Need only three *primary* colors

The Fovea

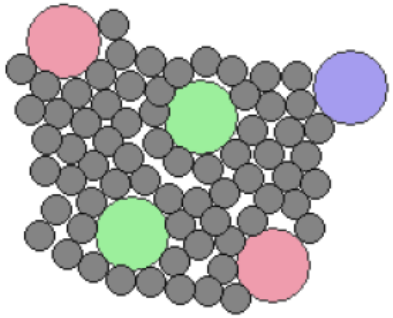
Cones are most densely packed within a region of the eye called the *fovea*.



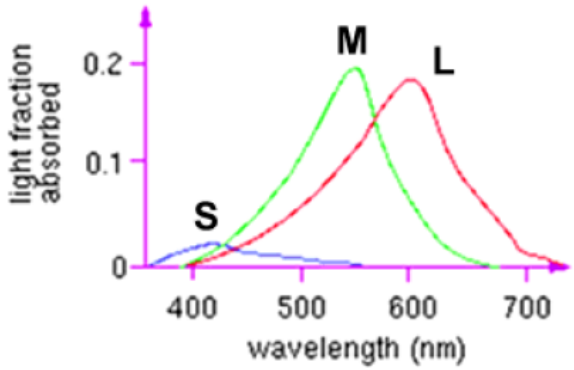
1.35 mm from rentina center



4 μm



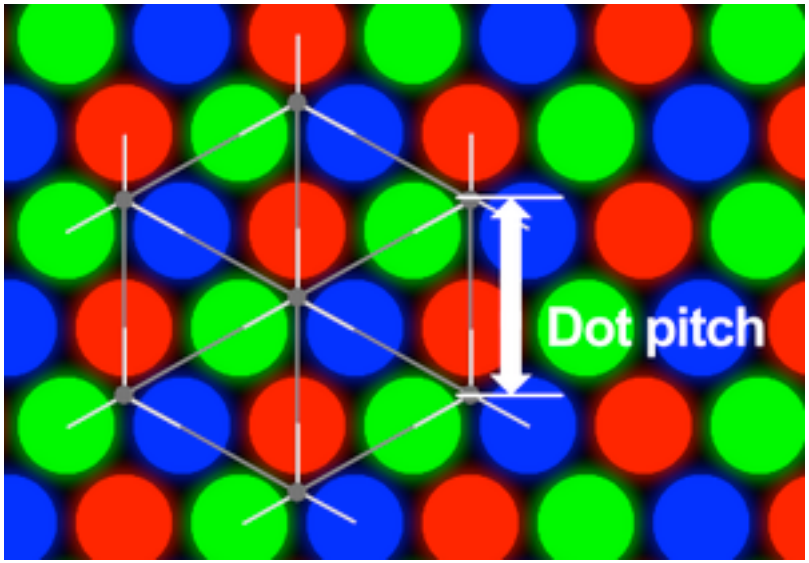
8 mm from rentina center



There are three types of cones, referred to as S, M, and L. They are roughly equivalent to blue, green, and red sensors, respectively. Their peak sensitivities are located at approximately 430nm, 560nm, and 610nm for the "average" observer.

The University of New Mexico

Monitor



Additive and Subtractive Color

- Additive color

  Form a color by adding amounts of three primaries
  - CRTs, projection systems, positive film

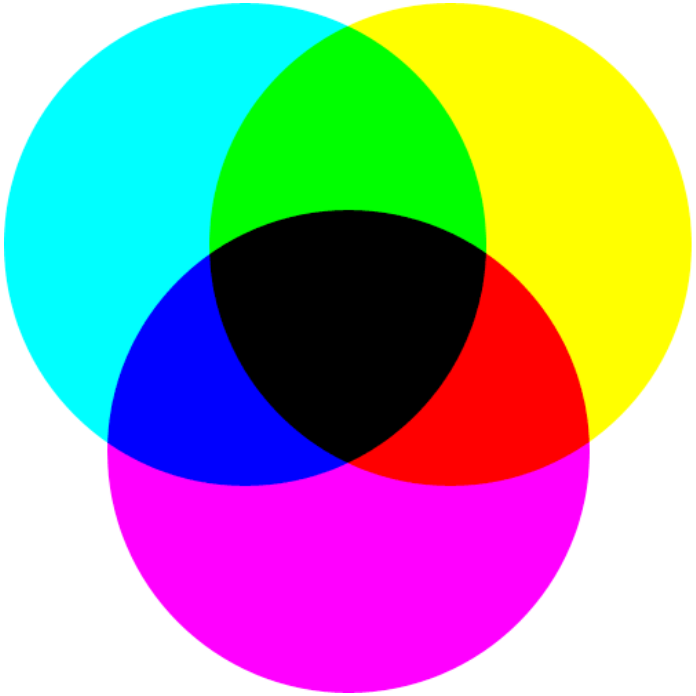
  Primaries are Red (R), Green (G), Blue (B)
- Subtractive color

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

RGB vs CMY

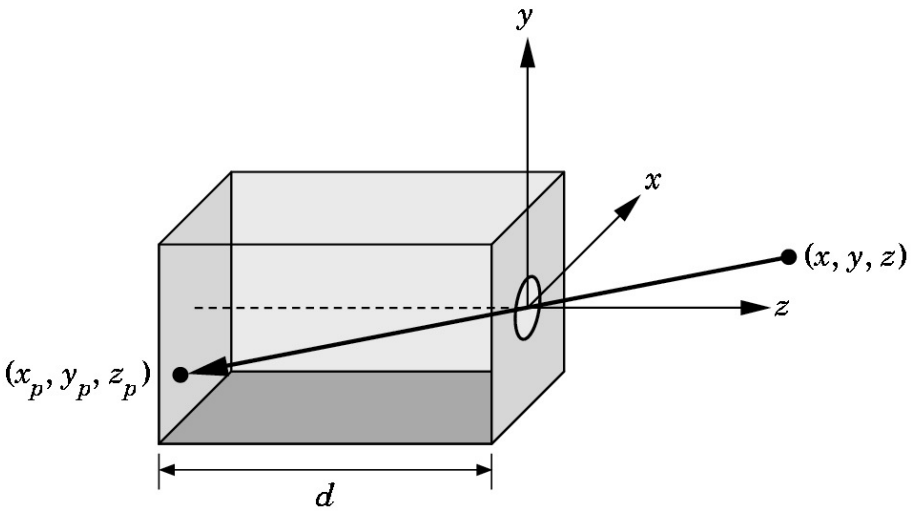
RGB

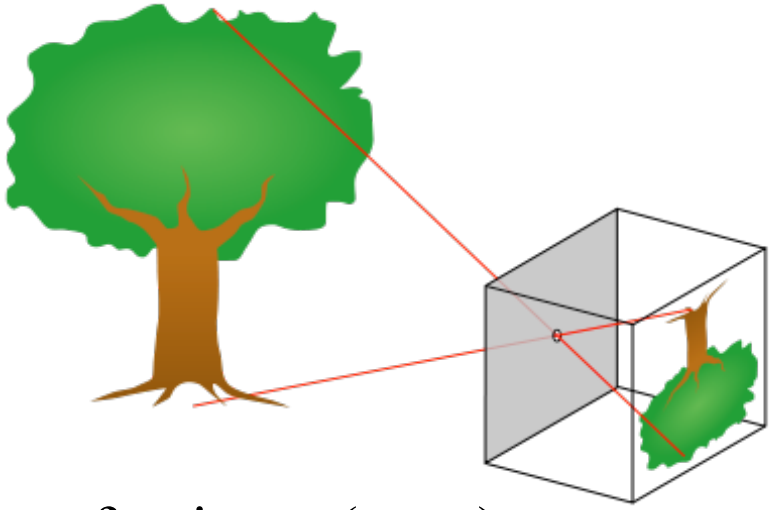
CMY



The University of New Mexico

Pinhole Camera



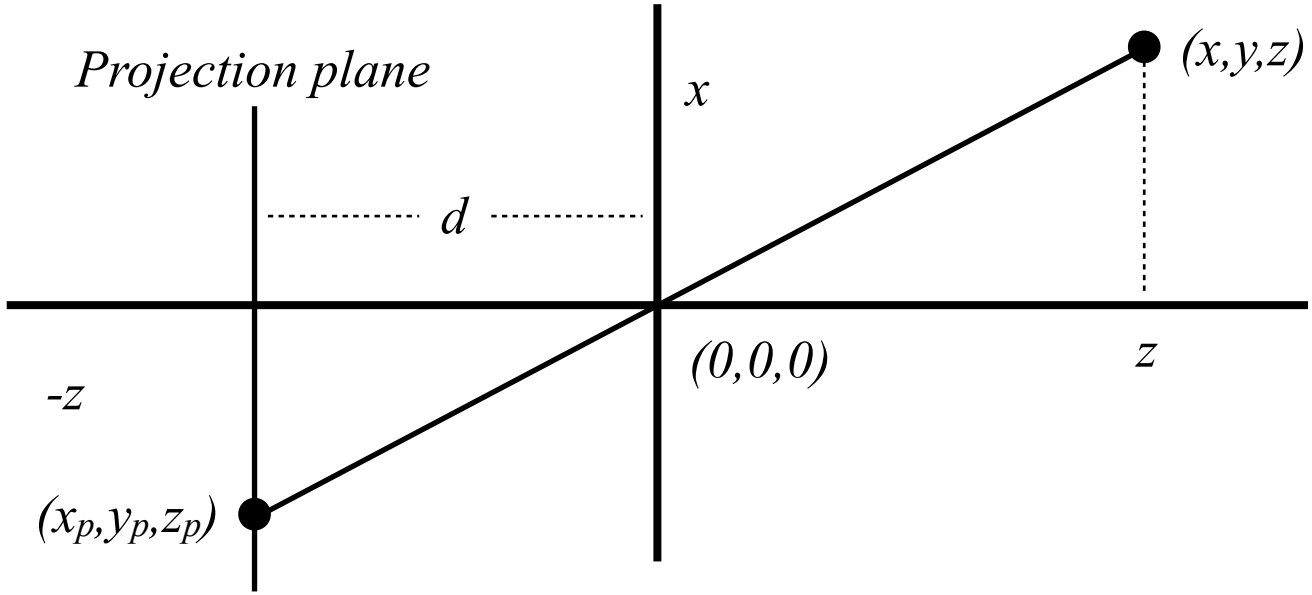


Use trigonometry to find projection of point at (x,y,z)

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

These are equations of simple perspective

Reminder: Similar triangles

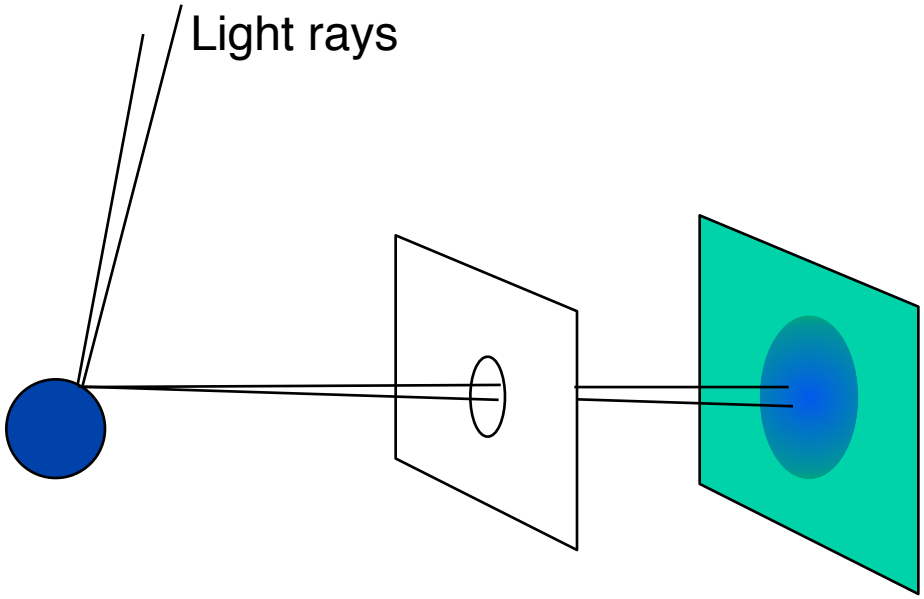


$$\frac{x}{z} = -\frac{x_p}{z_p} \rightarrow x_p = -z_p\frac{x}{z} = -d\frac{x}{z}$$

The University of New Mexico

Real camera ?

- Finite size



One object point projects on multiple locations

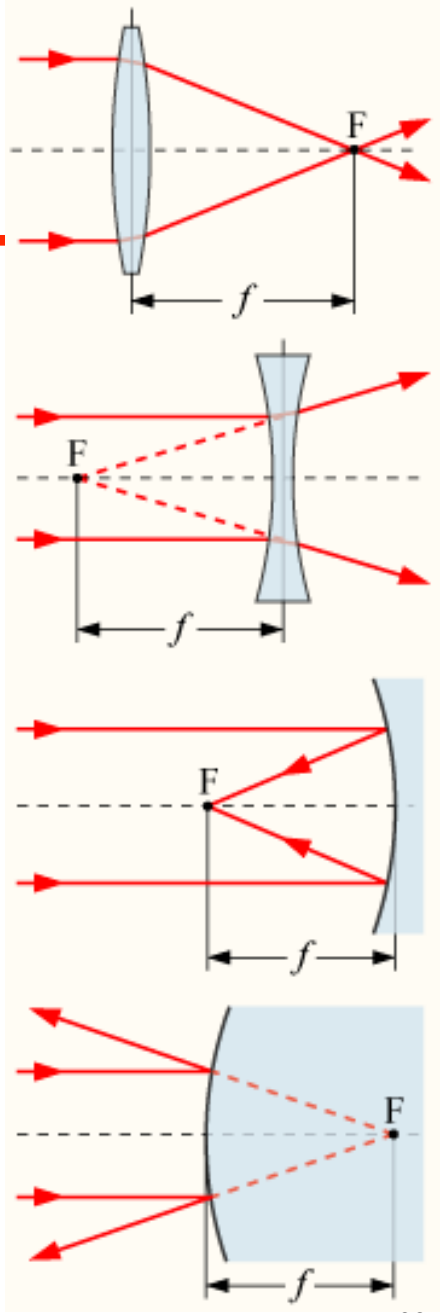
Different object points project on the same location

Result: blurred image

The University of New Mexico

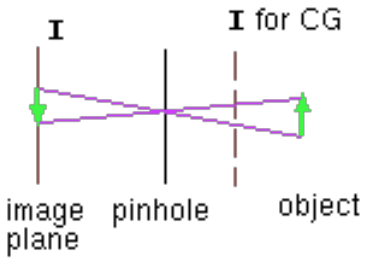
Lenses

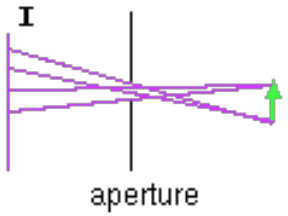
- Image
  Courtesy of Wikipedia

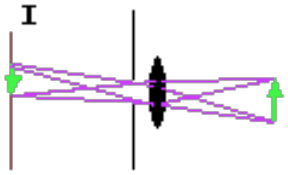


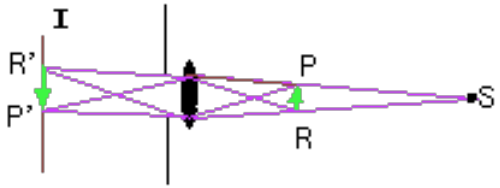
The University of New Mexico

Cameras and lenses



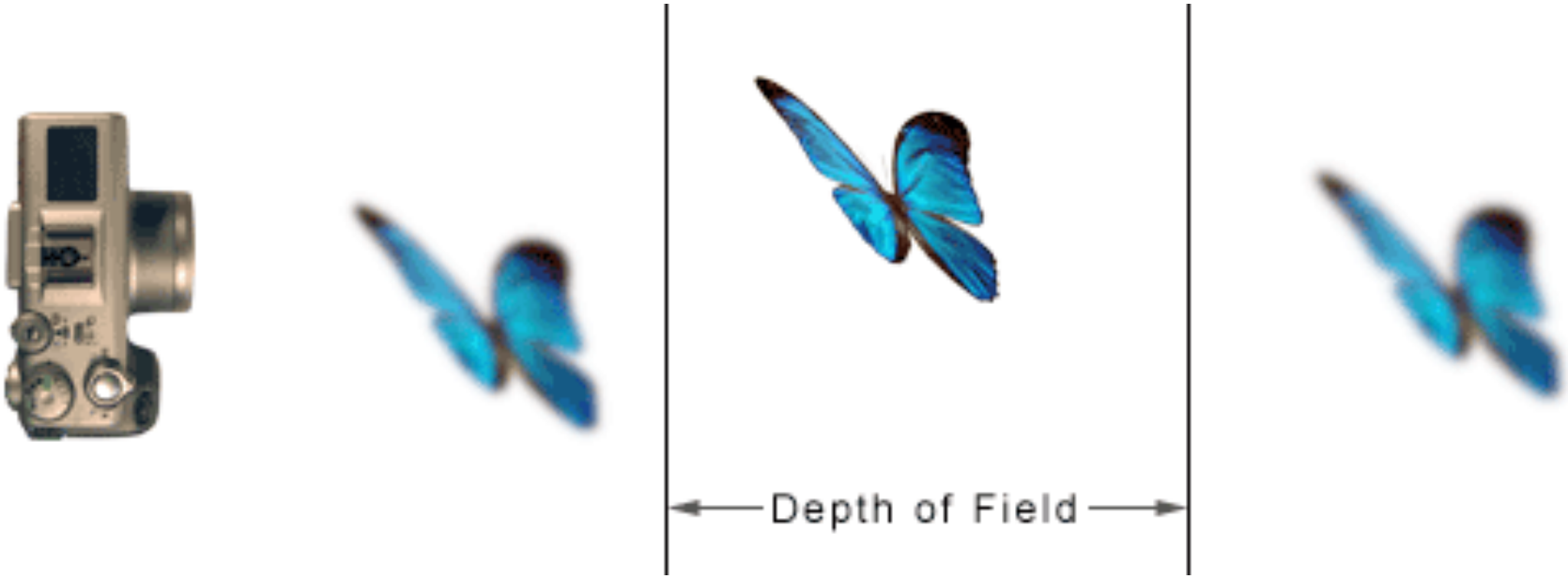






The University of New Mexico

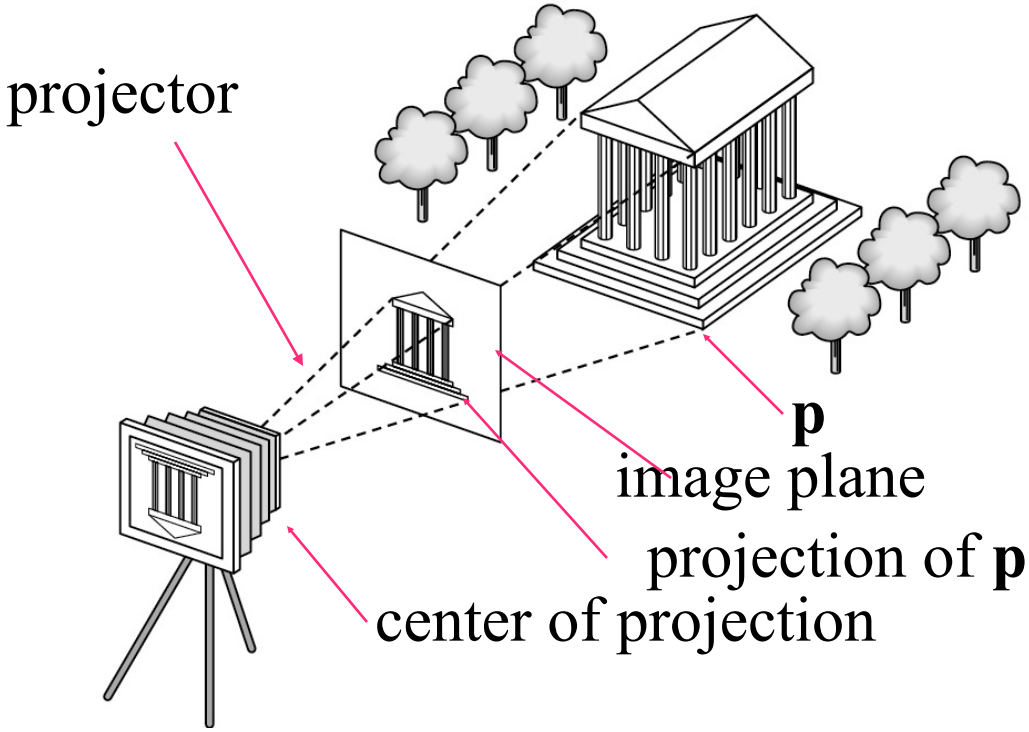
Depth of field



The University of New Mexico

Depth of field (contd)

Synthetic Camera Model



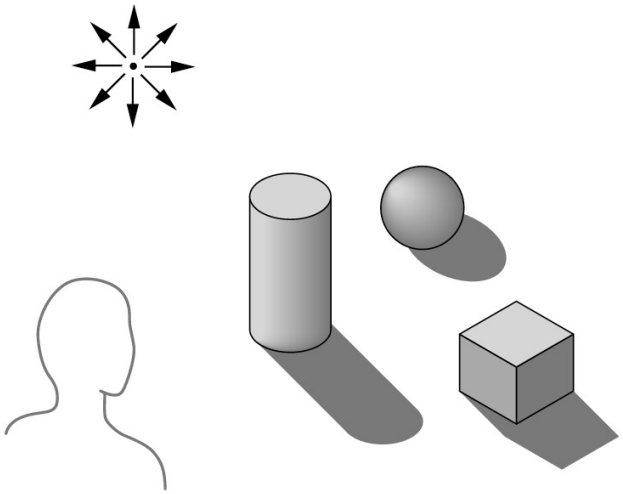
Ideal pinhole camera

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

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



Images

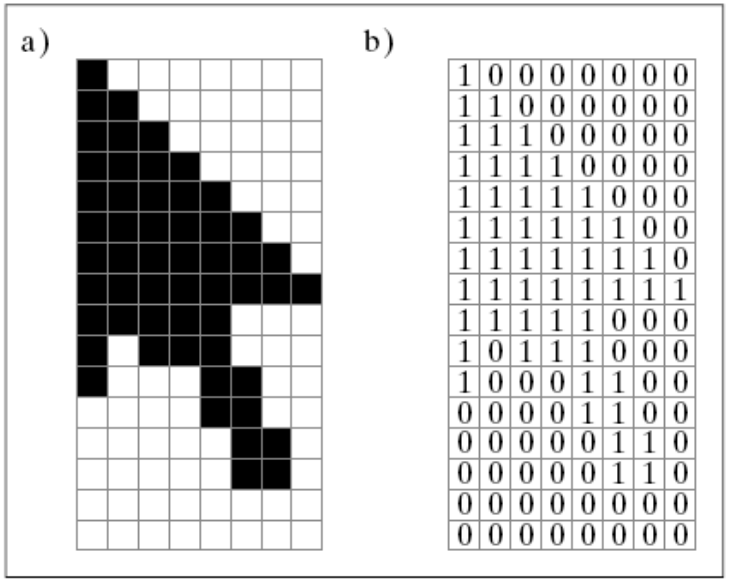
2D Arrays of color values (numbers)

Monochrome

Color

Monochrome

Black and White (Bitmaps)



Grayscale

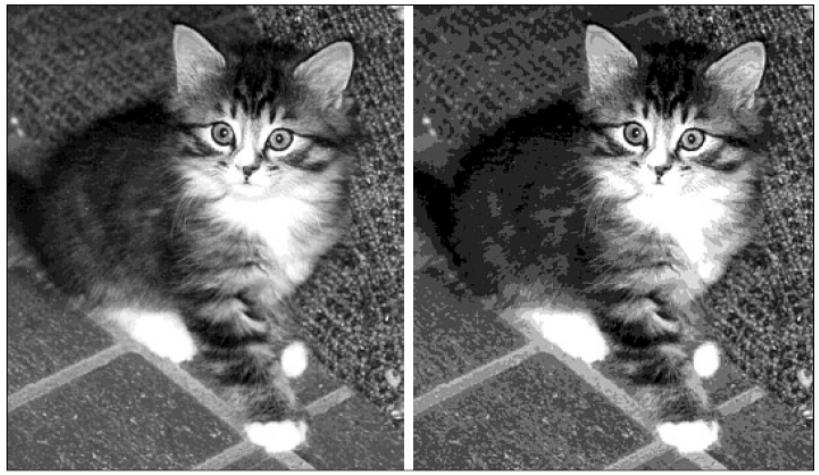


FIGURE 1.29 The image of Figure 1.22 reduced to (left) four bits per pixel and (right) three bits per pixel.

Color

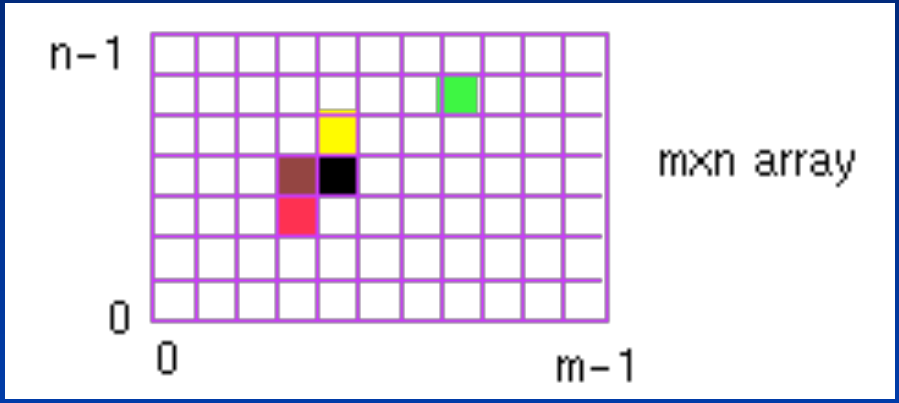
Common format RGB (3x8 = 24 bits per pixel)

Common format RGBA (4x8 = 32 bits per pixel)

Raster graphics

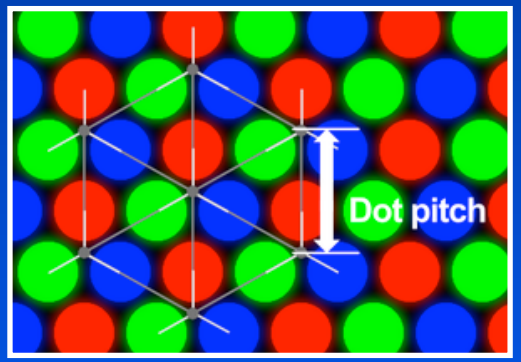
Virtual raster device

- Grid of m x n pixels

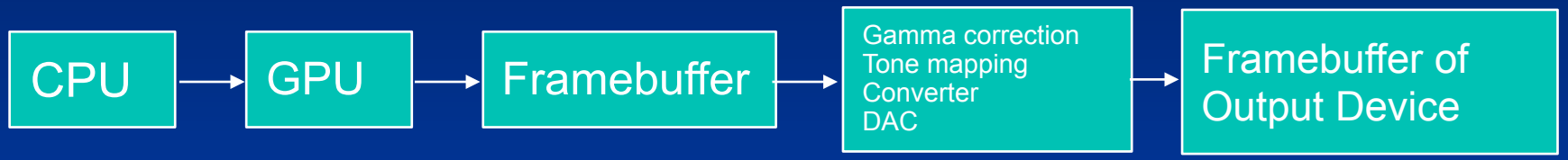


Real raster device

- Arrangement of 3 phosphors/LEDs per pixel
- Pixel may not be square



Basic display architecture



Pixels:

- Bitmap: 1bit/pixel
- Grey scale: 8 bits/pixel
- Colour map: 8 bits/pixel, indirect
- True color: 24 bits/pixel
- True color + Alpha Channel: 32 bits/pixel

Pixel format might be different for each device

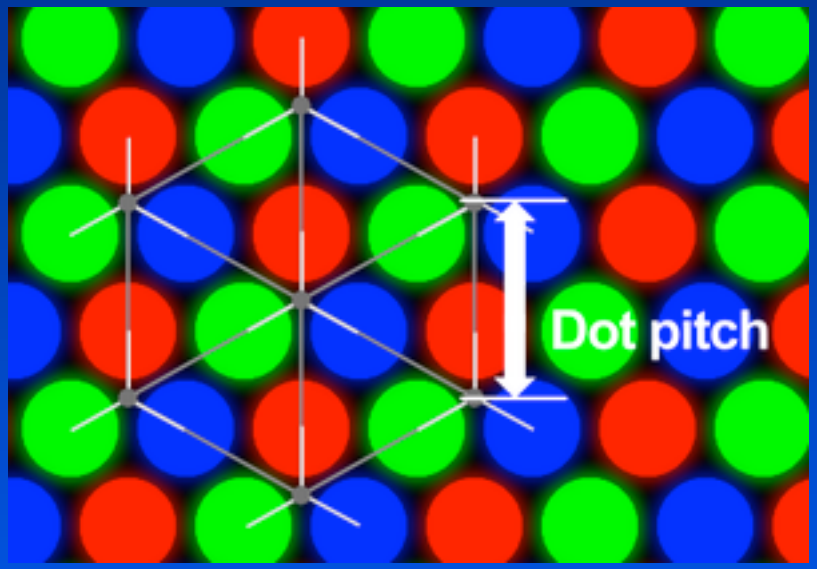
- Often we need to convert

Real video devices: Definitions

- Pixel aspect ratio: width/height usually 1
- Image aspect ration: m/n: 4/3 for NTSC, 16/9 for HDTV
- Refresh rate: nowadays can be between 60 to 480 Hz non-interlaced
- Phosphorescence: light emitted after element has turned off
- Monitor bandwidth
  - *Digital: bits or bytes per second*
  - *Analog: cycles per second (Hz)*
- Spot size: diameter of a single dot on the output device
- Resolution: number of pixels
- Dots per inch: In some sense the absolute size of a pixel
- Dot pitch: Distance between phosphors
- Contrast

Dot pitch

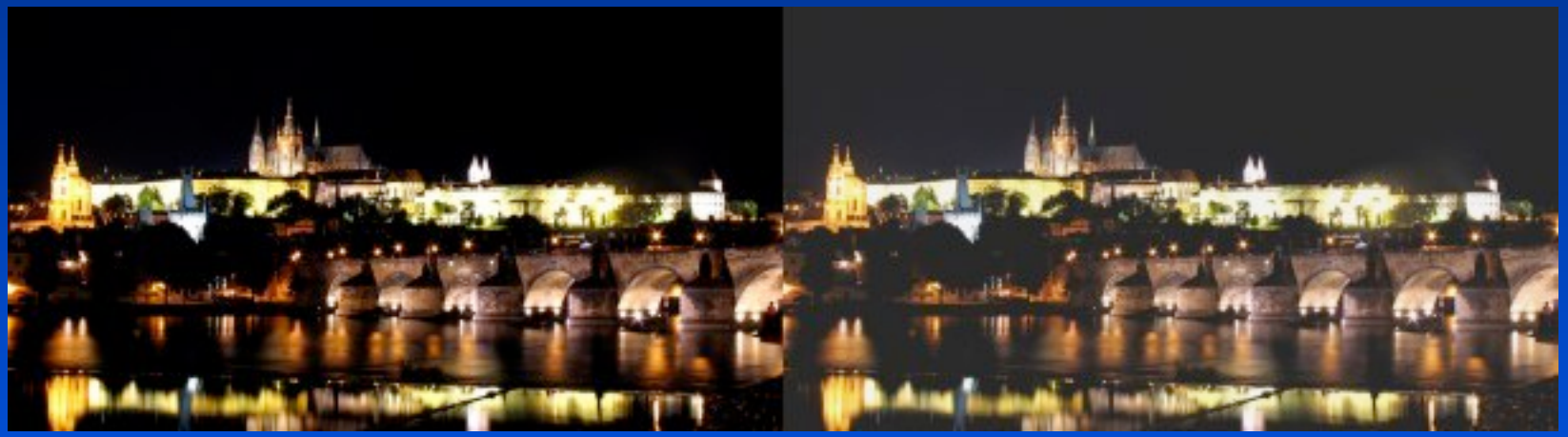
- Measure based on particular technologies
- For LCDs it is the distance between phosphors of the same color (usually the diagonal)



Credit: **Xilliah from Wikipedia**

Contrast and Contrast Ratio

- Same image displayed on two different monitors with different contrast ratio (white/black)



Credit: Geoffrey Morrison/CNET

Common video Standards

- **NTSC**
  - *North America & Japan*
  - *30 Hz, interlaced, 525 lines*
- **PAL**
  - *Commonwealth and Western Europe*
  - *25 Hz, interlaced, 625 lines*
- **SECAM**
  - *France, Eastern Europe, Middle East*
  - *26 Hz, interlaced, 625 lines*
- **HDTV**
  - *16:9 aspect ratio, digital, interlaced or progressive*
- **UHDTV**
  - *4K video, 4xHDTV*

Common display technologies

- CRT
- Passive Liquid Crystal
- Active Matrix (TFT) (transistors at grid points)
- Plasma Panel (neon bulbs)
- Active LEDs
- OLEDS

Summary

Polygon based Graphics Pipeline

Interactivity (real-time constraint)

RGB color space

Ideal camera

Z-buffer algorithm

Raytracing