Introduction to Computer Graphics

2016 Spring

National Cheng Kung University

Instructors: Min-Chun Hu 胡敏君

Shih-Chin Weng 翁士欽 (西基電腦動畫)

About This Course

- Lectures:
  - p.m. 3:10~6:00, Tuesday
  - R4263, CSIE
- Prerequisites
  - Programming skills in C/C++/Java
  - Data structures
- Lecturer
  - Min-Chun Hu, Assistant Professor
  - Shih-Chin Weng, Senior Enginneer @ CG Digital Contents
  - Email: anita_hu@mail.ncku.edu.tw
  - Office: R65B08, 11F, CSIE New Building

About This Course (Cont.)

- TAs:
  - 黃均暉 F74006030@ncku.edu.tw
  - 許友綸 F74012138@ncku.edu.tw
  - 朱承昱
  - 林季伯
  - Office: R65601, 6F, CSIE New Building

About This Course (Cont.)

- Textbooks:
  - E. Angel, Interactive Computer Graphics 6th Ed., Addison-Wesley, 2012.
  - D. Hearn, M.P.Baker, Computer Graphics with OpenGL 3rd Ed., Prentice Hall, 2004.
  - Tomas Akenine-Moller, Eric Haines, Naty Hoffman, Real-Time Rendering, 3rd Ed., 2008.
  - Matt Pharr, Greg Humphreys, Physically Based Rendering, From Theory To Implementation, 2nd Ed., 2010.
  - Christer Ericson, Real-Time Collision Detection (The Morgan Kaufmann Series in Interactive 3-D Technology), 2004.
  - Watt. 3D Computer Graphics, 3rd ed., Addison-Wesley, 1999.

What Can I Learn from This Course?

- Fundamentals of computer graphics techniques.
- Programming ability of OpenGL Shading Language (GLSL).
- Some of 2D image special effects and usage of editing tools.

Syllabus

- 2/23 Overview of Computer Graphics + HW1
- 3/1 Computer Animation Pipeline
- 3/8 Data Representation
- 3/15 Rendering Pipeline
- 3/22 Coordinates and Transformations
- 3/29 Introduction of GLSL+ HW2
- 4/5 Spring Vacation
- 4/12 Basic Shading Algorithms (Part I)
- 4/19 Basic Shading Algorithms (Part II) + HW3
- 4/26 Surface Reconstruction
- 5/3 Image Processing + HW4
- 5/10 Computer Animation
- 5/17 參訪西基電腦動畫股份有限公司
- 5/24 Physically Based Rendering
- 5/31 Global Illumination (Part I)
- 6/7 Global Illumination (Part II)
- 6/14 3D Stereo and 3D Compression
- 6/21 Final Project Demo

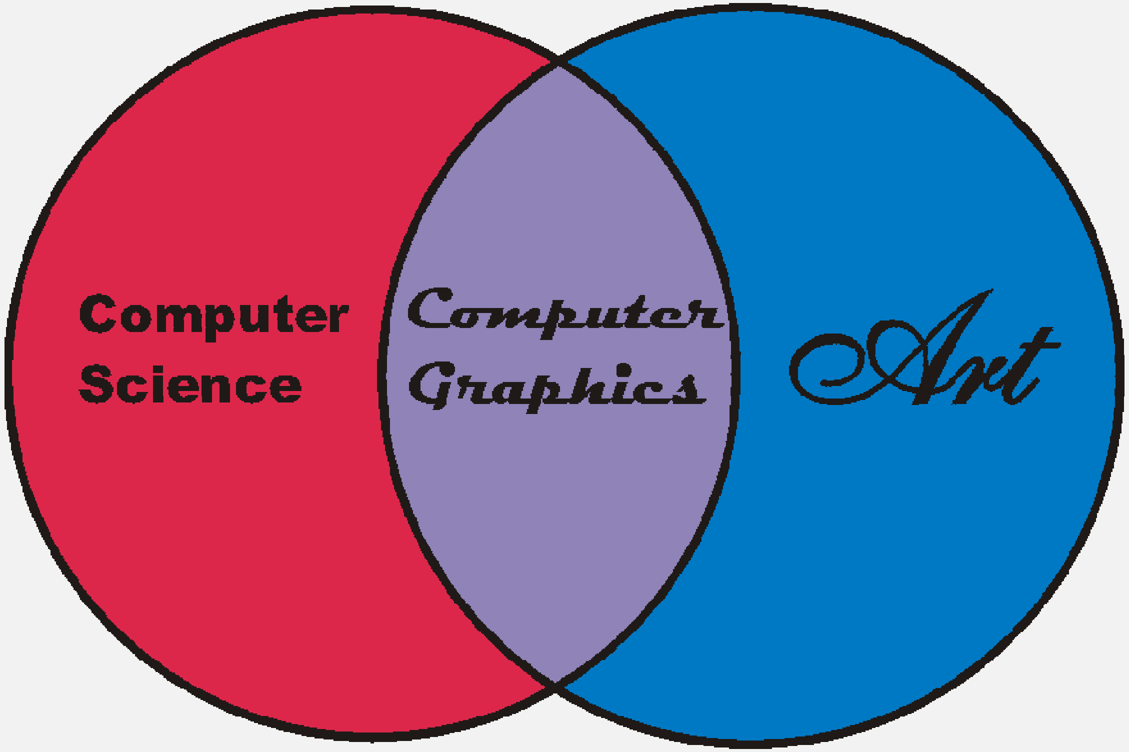
Grading

- HW1: Create your own 3D model by Blender 3D (10%)
  - Deadline: 02/29 pm 10:00
- HW2: GLSL (15%)
  - Deadline: 04/11 pm 10:00
- HW3: GLSL (15%)
  - Deadline: 05/02 pm 10:00
- HW4: GLSL (15%)
  - Deadline: 05/23 pm 10:00
- Midterm: Paper presentation each week (15%)
- Final: Project (30%)
  - Deadline: 06/20 pm 10:00

Course Notice

- Office hour:
  - By an appointment
- No late submission of HW !
- Discussion is encouraged, but plagiarize (even the codes from websites) is not allowed !
- Food is ok~
- Zzz…not that ok~

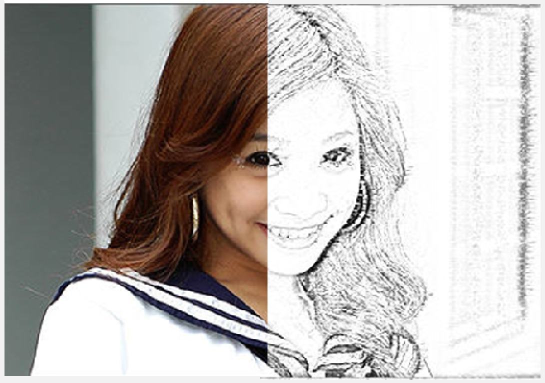
What’s Computer Graphics (CG)



What’s Computer Graphics (CG)

What’s Computer Graphics (CG)

- Computer Graphics
  - Producing pictures or images using computer.
  - Displaying a realistic virtual environment or synthesizing virtual objects in real time.
    - Mainly focusing on 3D graphics
  - Displaying a real scene/object with specific styles.




Behind The Scenes: Angry Birds 2 Launch Film

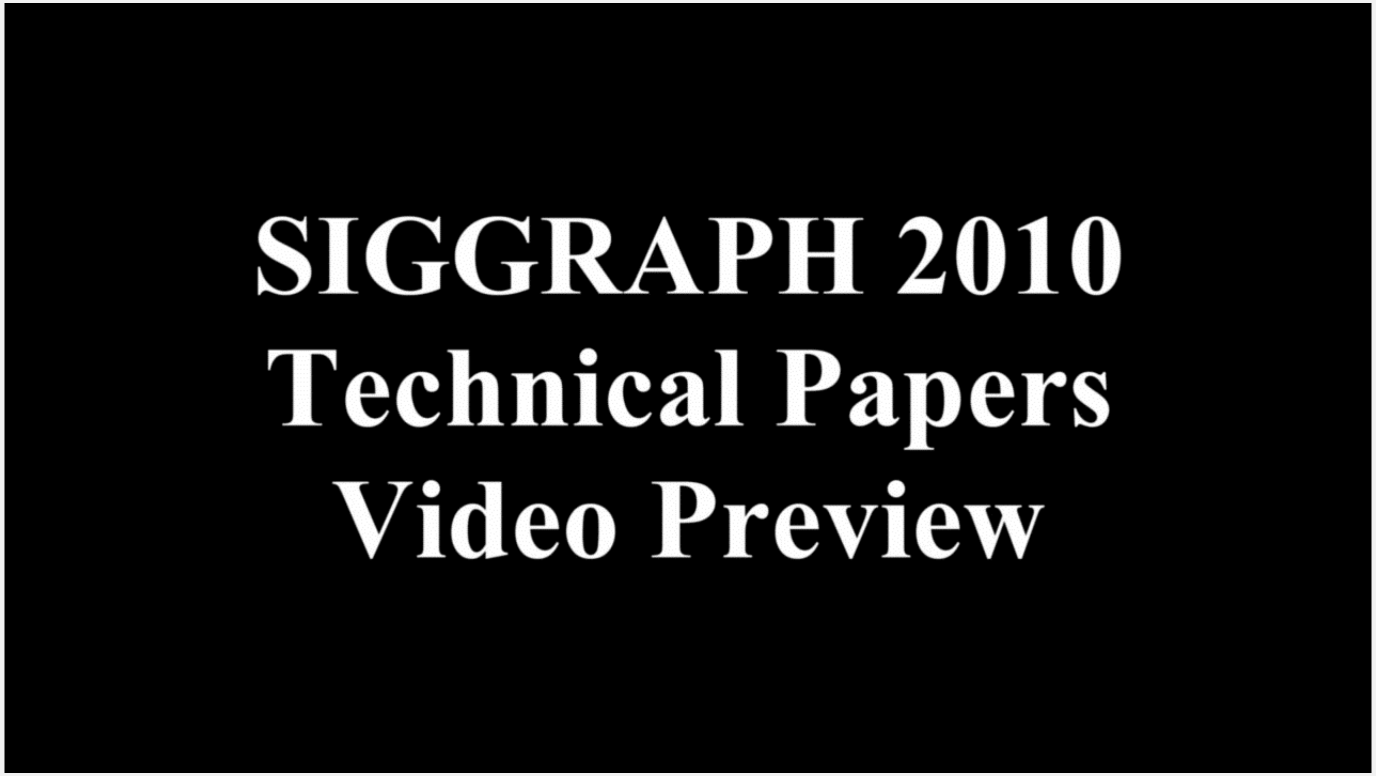
Face Transformation

Real Digital Human Face

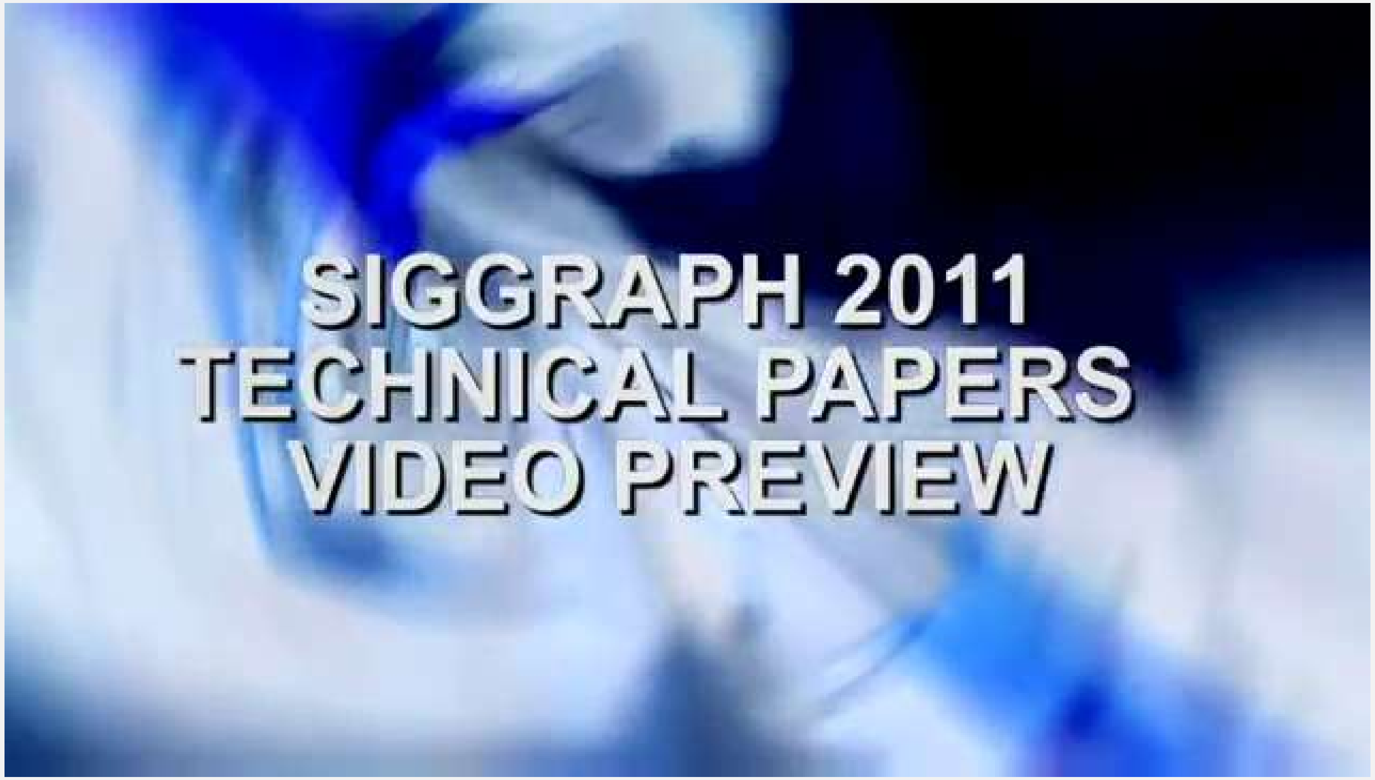
Facial Reenactment

3D Avatar Creation

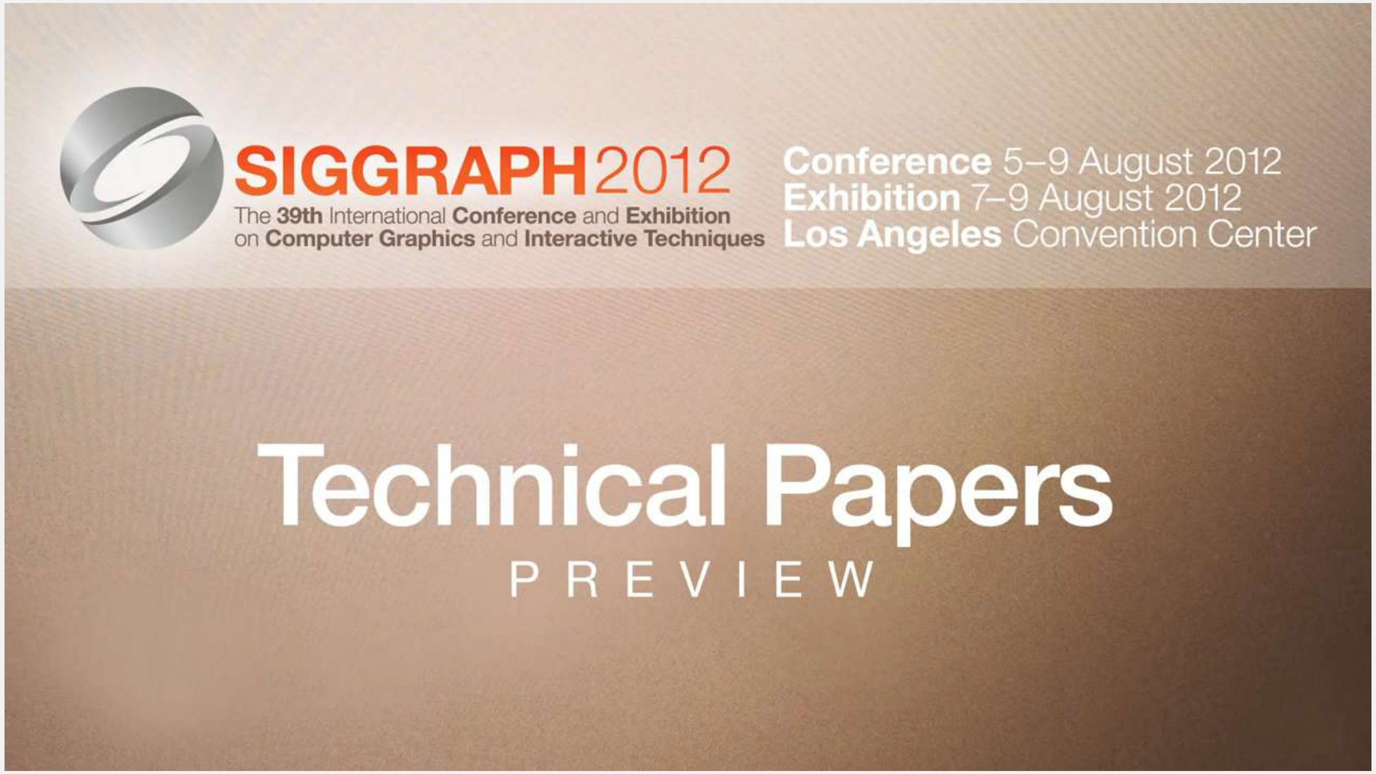
SIGGRAPH 2010



SIGGRAPH 2011



SIGGRAPH 2012



SIGGRAPH 2013

SIGGRAPH 2014

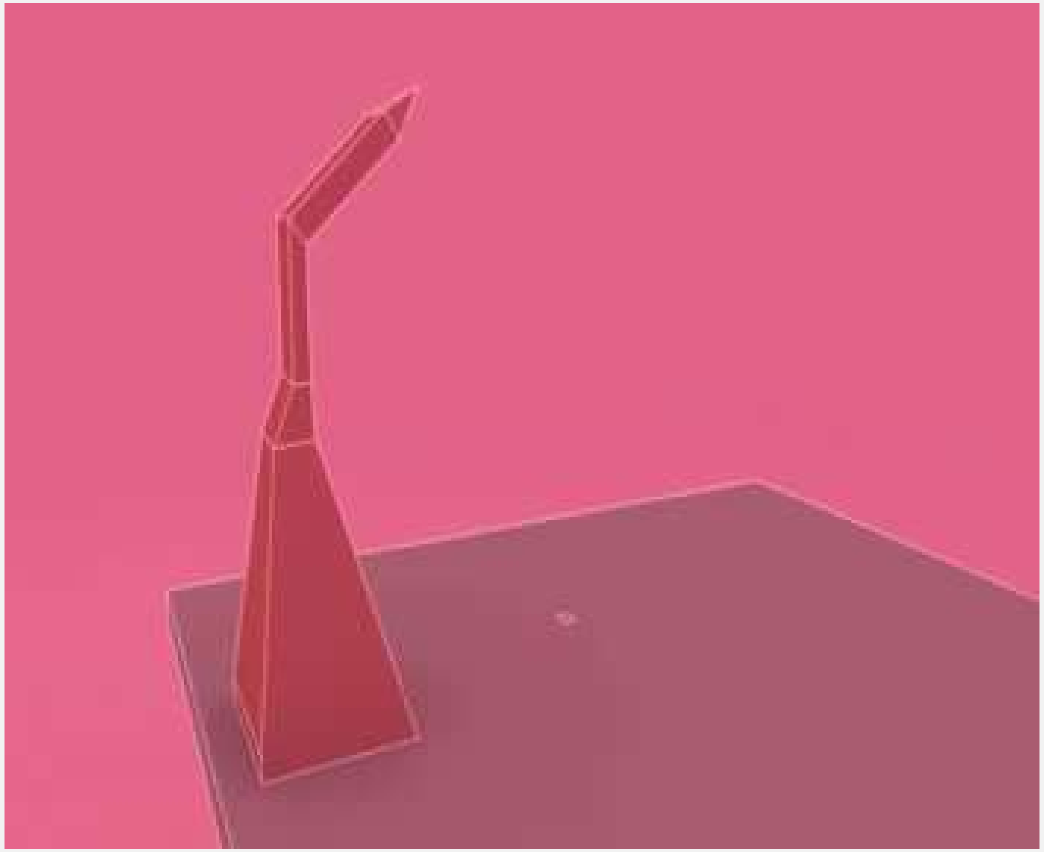
SIGGRAPH 2015

SIGGRAPH 2015 (Real Time Live)

SIGGRAPH 2015 (Computer Animation Festival)

CG or Magic ??

Proud of Taiwan !!!



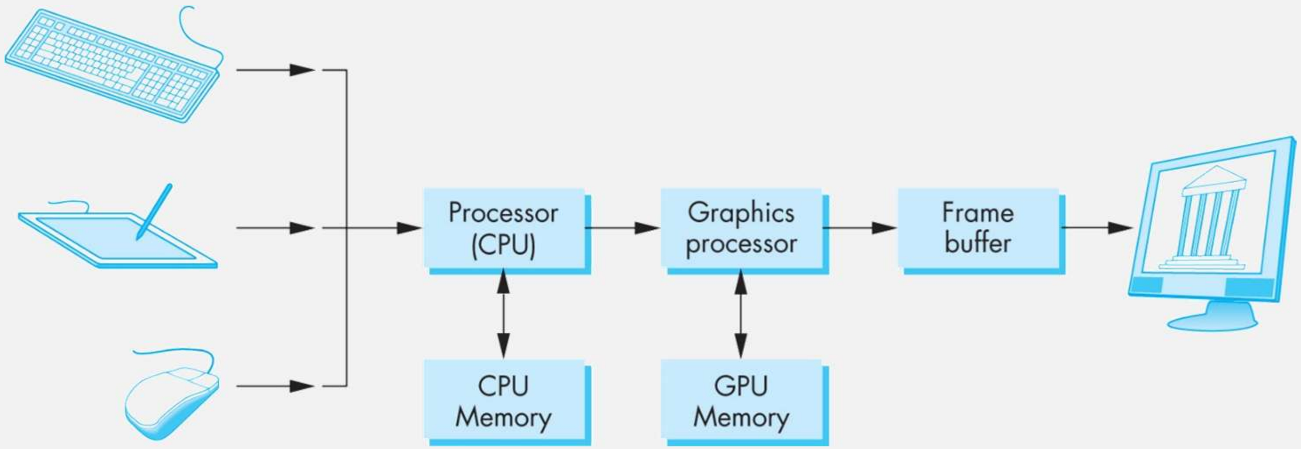
獲2005年ACM SIGGRAPH 國際動畫展「電子劇院」觀眾票選第一名。導演全明遠，編劇孫春望。

Applications of Computer Graphics

- Display/visualization of information
  - Architectural/mechanical drafting systems
  - Cartography
  - Plotting packages to visualize multiple large data sets
  - Medical imaging (CT/MRI)
- Design
  - Computer-aided design (CAD)
  - Very-large-scale integrated (VLSI) circuits design
- Simulation and animation
  - Training of pilots
  - 2D/3D motion-pictures in TV/advertising industries
  - Virtual Reality (VR) & Augmented Reality (AR)
- User interface
  - Windowing systems
  - Browser interface

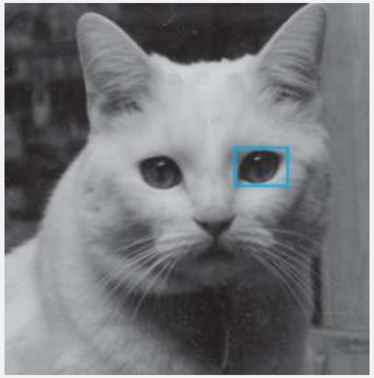
A Graphics System

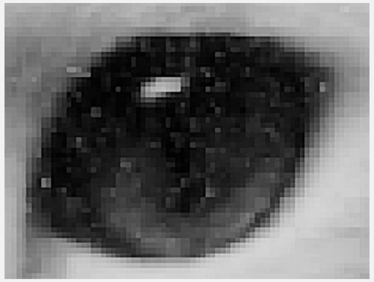
- Components of a general-purpose computer system:
  - Input devices
  - Central Processing Unit
  - Graphics Processing Unit
  - Memory
  - Frame buffer
  - Output devices



Pixels & Frame Buffer

- The image we see on the output device is an array (the raster) of picture elements (pixels) produced by the graphics system.
- Pixels are stored in a part of memory called the frame buffer.
- Resolution: the number of pixels in the frame buffer.
- Depth/Precision: the number of bits used for each pixel.
  - 1-bit-deep frame buffer: only two colors
  - 8-bit-deep frame buffer: 256 colors
  - Full-color/True-color/RGB-color system: 24 (or more) bits per pixel
  - HDR systems: 12 (or more) bits per color component

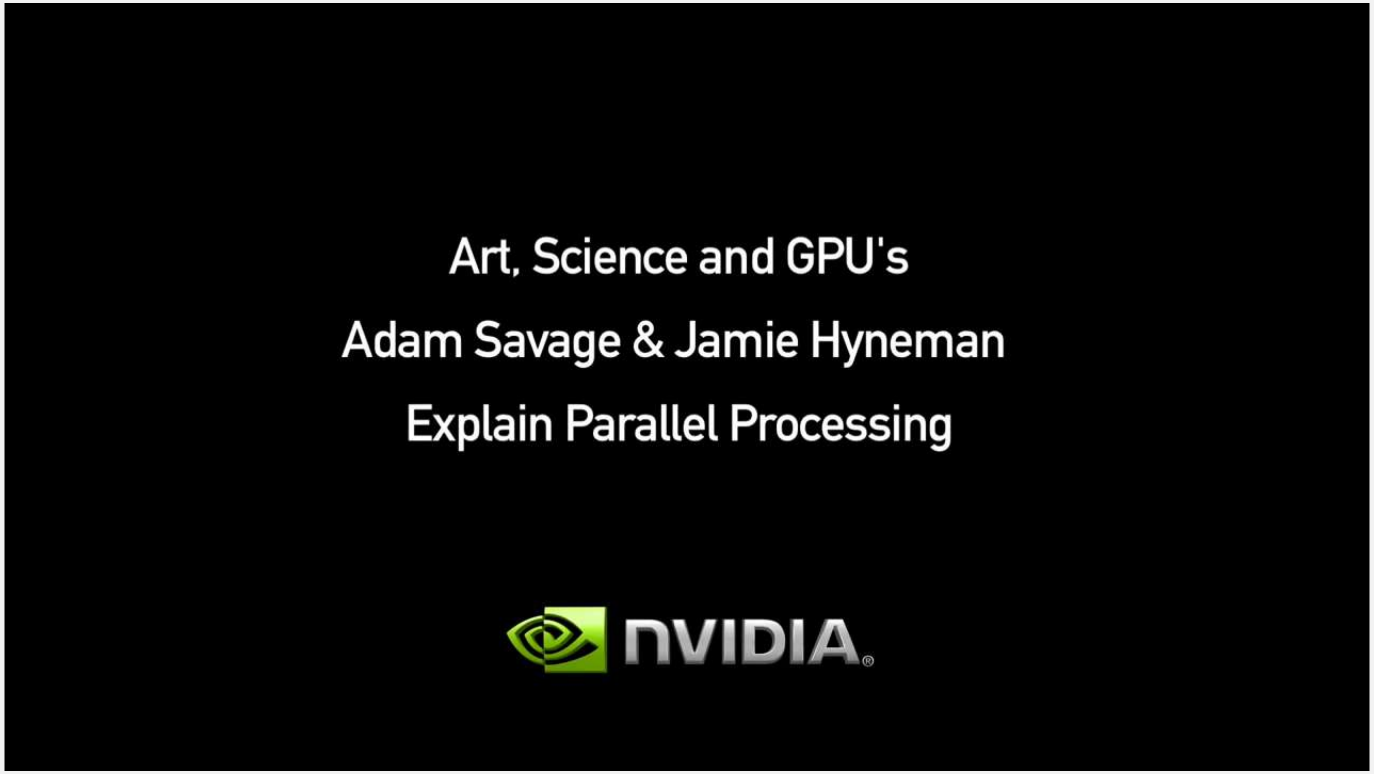




CPU & GPU

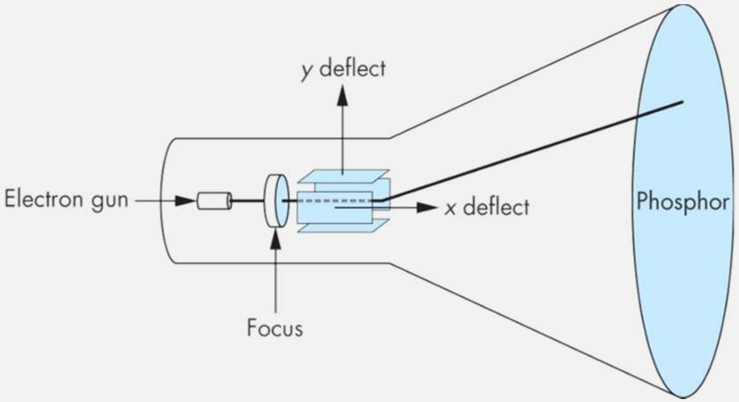
- Main graphical function of the processor:
  - Rasterization/Scan conversion: Conversion of geometric entities (such as lines, circles, polygons) to pixel colors and locations in the frame buffer.
- Frame buffer was part of the standard memory that could be directly addressed by the CPU in early graphics system.
- Today, graphical systems are characterized by special-purpose graphical processing units (GPUs) that can perform graphical operations with high degree of parallelism.
- GPU can be either on the mother board of the system or on a graphics card.

Art Science & GPU's

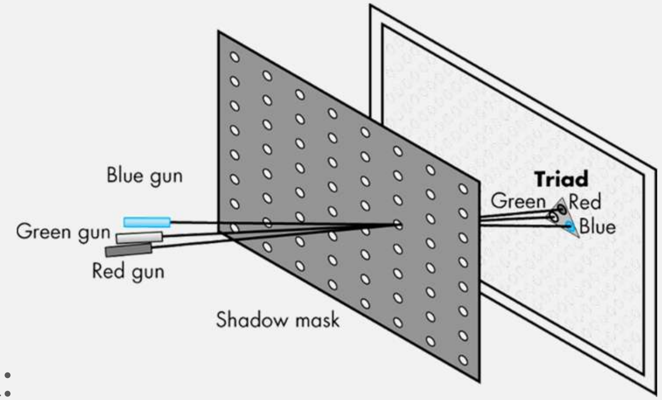


Output Devices

- Cathode-ray tube (CRT) display/monitor



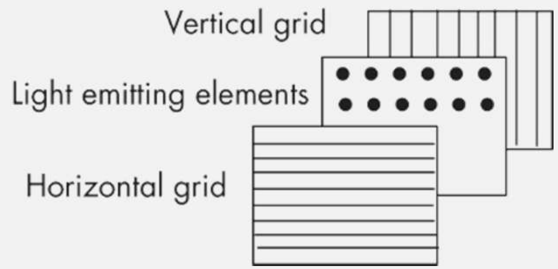
- Color CRT



- Raster system:
  - Noninterlaced system:
    - Pixels are displayed row by row or scan line by scan line
  - Interlaced system:
    - Odd rows and even rows are refreshed alternately

Output Devices (Cont.)

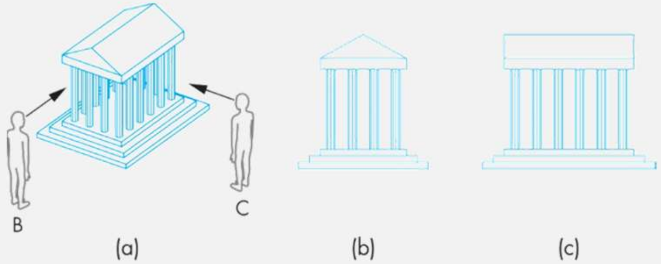
- Flat-panel monitors
  - Lighting-emitting diodes (LEDs)
  - Liquid-crystal displays (LCDs)
  - Plasma panels

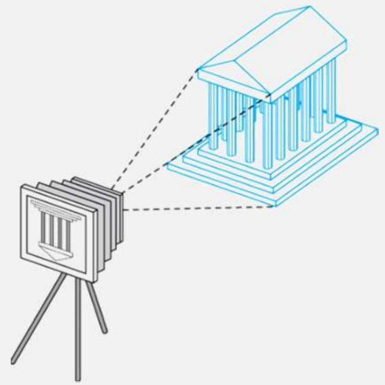


- Projection systems: usually raster-based devices
- Hard-copy devices: usually raster-based devices but cannot be refreshed

Objects & Viewers

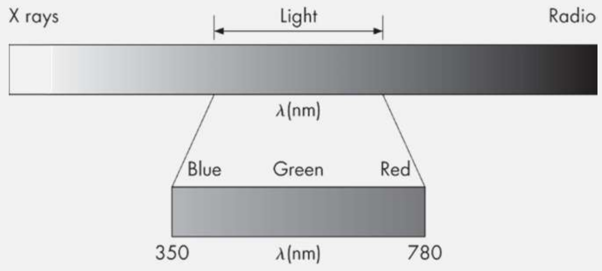
- Objects:
  - Can be defined/approximated by a set of locations in space, i.e. vertices.
- Viewers:
  - Who form the image of the objects.
- Both objects and viewers exist in a 3D world. However, the formed image is 2D.

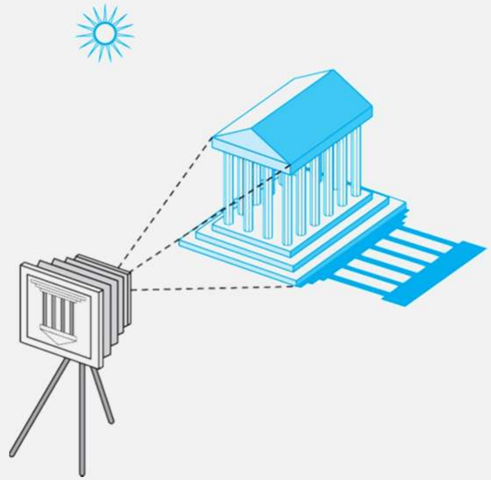




Light & Images

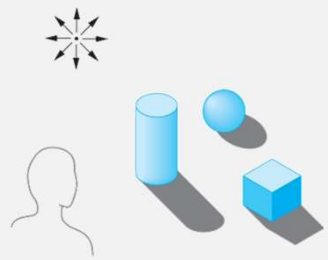
- Visible spectrum: 350~780 nm
- Point source:
  - Emits energy from a single location at one or more frequencies equally in all directions.
- Light bulb:
  - Emits light over an area and emitting more light in one direction than another.
- Complex light sources can be approximated by a number if carefully placed point sources.

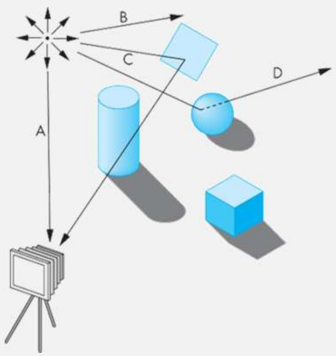




Imaging Models

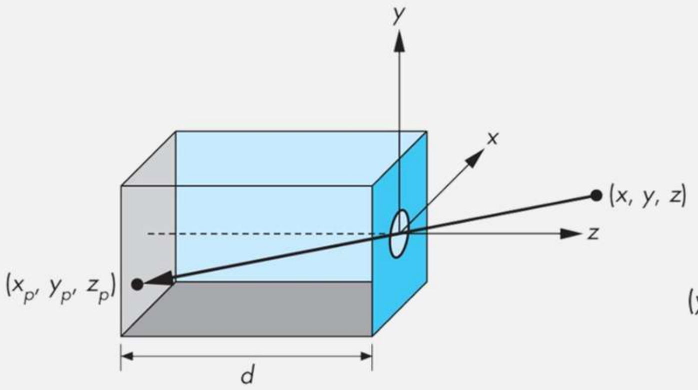
- How we can form images from a set of objects with different light-reflecting properties and different light sources?
  - Building the imaging model by following light from a source.
    - E.g. Raytracing/raycasting and photon mapping
    - Can provide a close approximation to the physical world, but is not well suited for real-time computation.
  - Conservation of energy.
    - E.g. Radiosity
    - Works best for surfaces that scatter the incoming light equally in all directions.

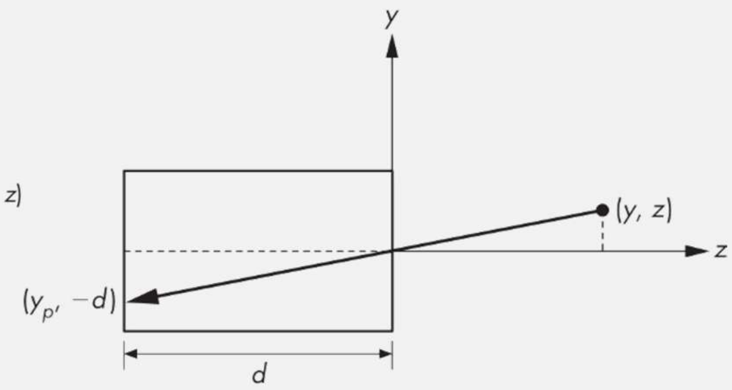




Imaging Systems: Pinhole Camera

- The pinhole camera:
  - Suppose that the camera is oriented along the z-axis, with the pinhole at the origin of our coordinate system.
  - Assume that the hole is so small that only a single ray of light from a point can enter it.



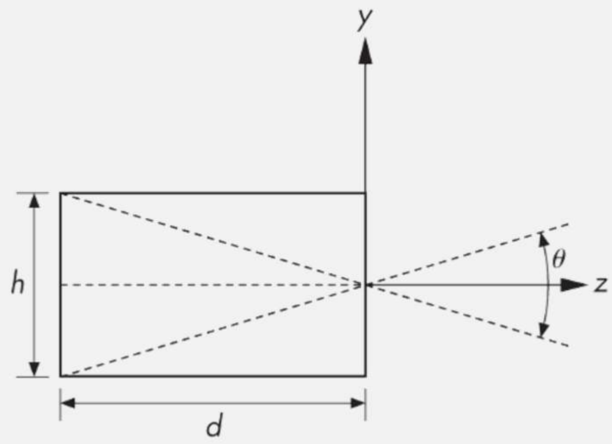


$$y_p = -\frac{y}{z/d}$$

$$x_p = -\frac{x}{z/d}$$

Imaging Systems: Pinhole Camera (Cont.)

- The field/angle of view:
  - 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 filed. (Every point in its filed of view is in focus)
- Disadvantages:
  - Admits only a single ray from a point source, and therefore almost no light enters the camera.
  - The camera cannot be adjusted to have a different angle of view.

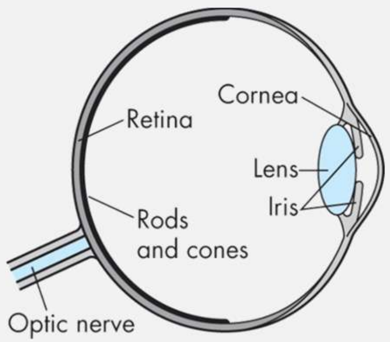


$$\tan\frac{\theta}{2} = \frac{h/2}{d}$$

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

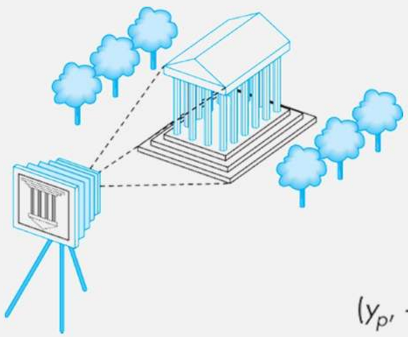
Imaging Systems: Human Visual System

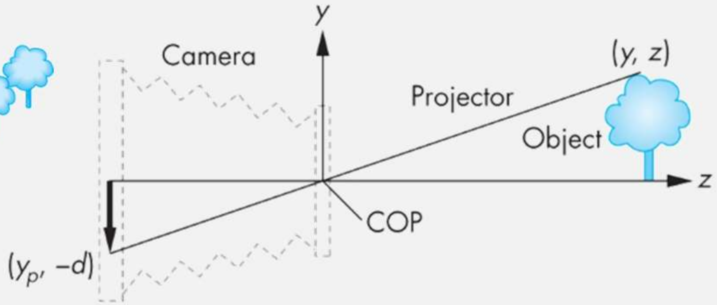
- Sensors in the human eye do not react uniformly to light energy at different wavelength.
  - Most sensitive to green light and least sensitive to red and blue.
- There are three types of cones and therefore we can use three standard primaries to approximate any color that we can perceive.
  - Intensity is a physical measure of light energy.
  - Brightness is a measure of how intense we perceive the light from an object.

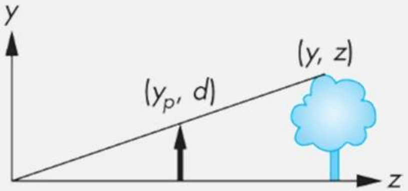


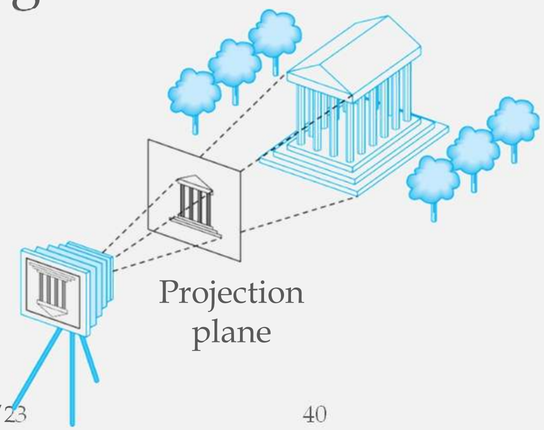
The Synthetic-Camera Model

- The specification of the objects is independent of the specification of the viewer.
  - Within a graphics library, there will be separate functions for specifying the objects and the viewer.
- We can compute the image using simple geometric calculation
- Draw another plane in front of the lens to avoid flipping



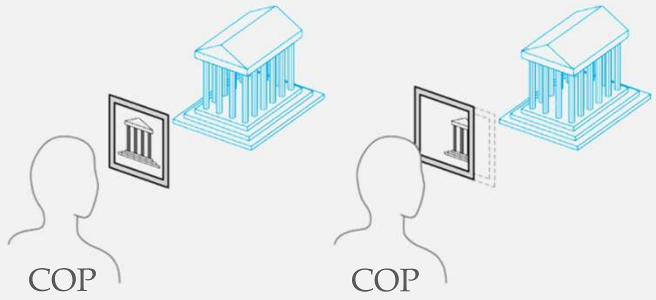






The Synthetic-Camera Model (Cont.)

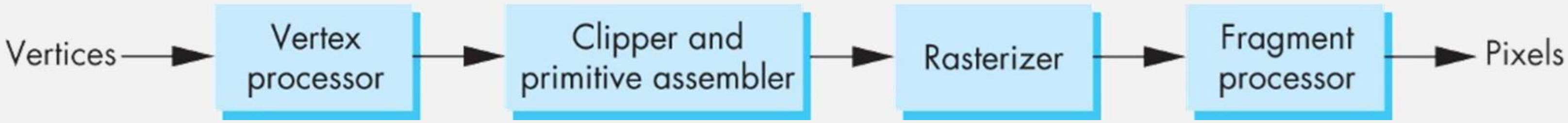
- Not all objects can be imaged onto the pinhole camera's film plane, and the synthetic camera move the limitation to the front by placing a clipping rectangle/window in the projection plane.
- What determines which object will appear in the image?
  - The location of the center of projection (COP)
  - The location and orientation of the projection plane
  - The size of the clipping rectangle



LookAt(COP, at, up);

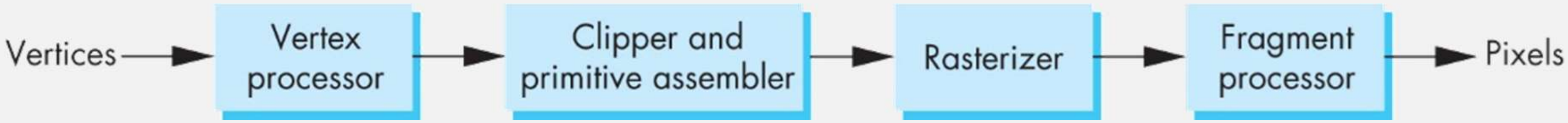
Perspective(field_of_view, aspect_ratio, near, far);

The Graphics Pipeline



- The graphics pipeline or rendering pipeline refers to the sequence of steps used to create a 2D raster representation of a 3D scene/model.
- Vertex processing
  - Each vertex is processed independently.
  - To carry out coordinate transformations.
    - Each change of the camera coordinate can be represented by a matrix.
  - To compute a color for each vertex.
- Clipper and Primitive Assembly
  - Efficient clipping must be done on a primitive-by-primitive basis rather than on a vertex-by-vertex basis.

The Graphics Pipeline (Cont.)



- Rasterization (Scan conversion)
  - Primitives emerging from the clipper are still represented in terms of their vertices and must be converted to pixels in the frame buffer.
  - Determine which pixels in the frame buffer are inside the polygon.
  - Output of rasterization is a set of fragments (potential pixels with color, location, and depth information) for each primitive.
- Fragment Processing
  - Update the pixels in the frame buffer according to the processed fragments. (Some surfaces may not be visible because of occlusion)
  - The color of pixels in each fragment can be altered by texture mapping or bump mapping.

Q & A ?