COSE436

Lecture 2: Graphics Systems

Sep 5, 2024

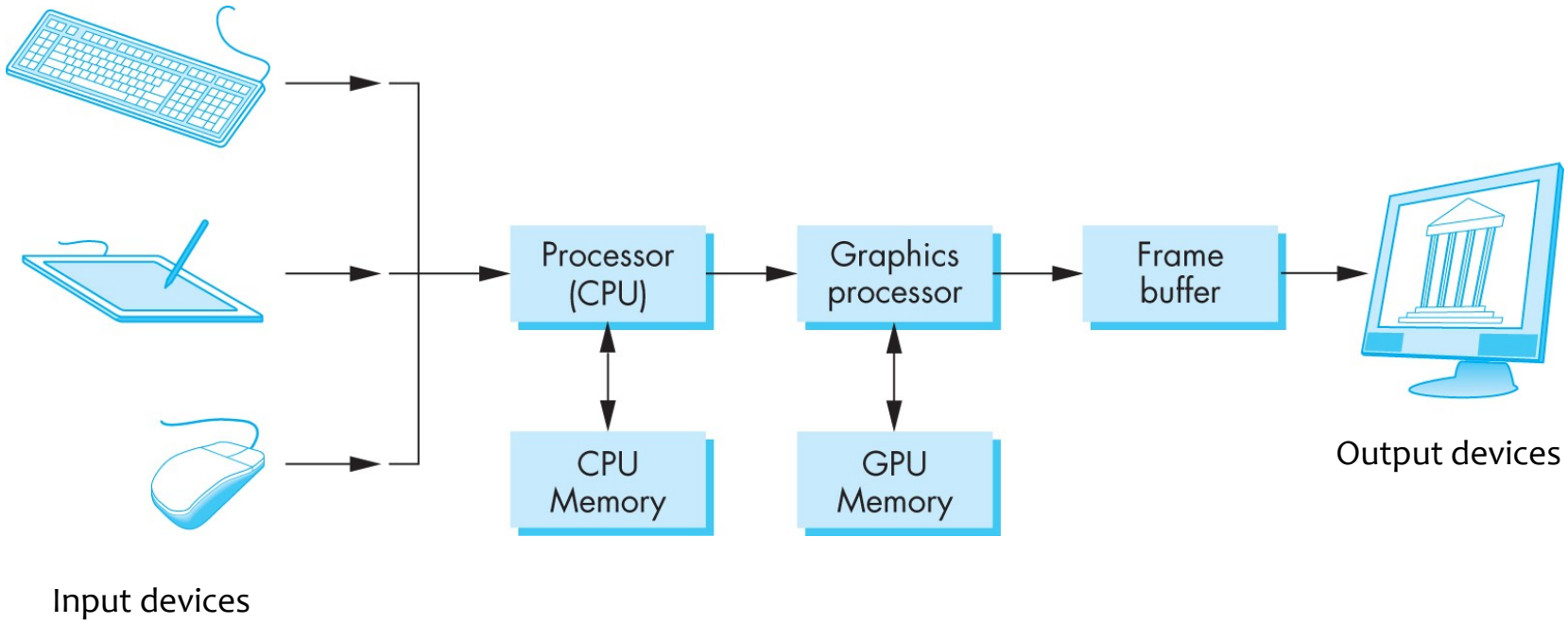
Won-Ki Jeong
(wkjeong@korea.ac.kr)

KOREA UNIVERSITY

Outline

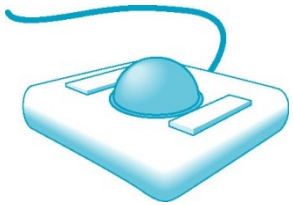
- Graphics system
- Image formation
- Raster graphics pipeline
- Rasterization

Graphics System

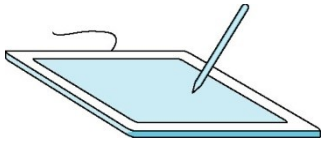


Input Devices

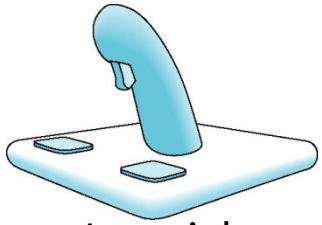
Mouse



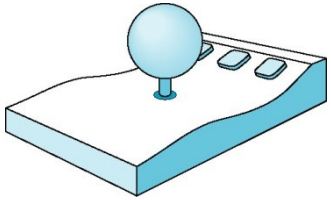
Trackball



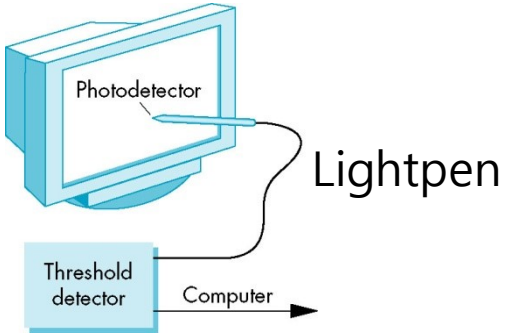
Data Tablet



Joystick



Spaceball



Input Devices

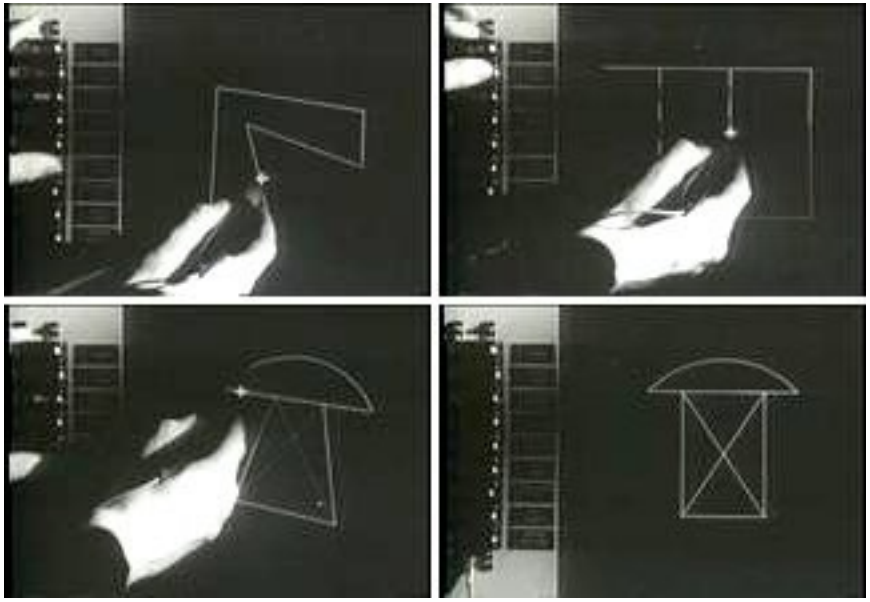
Haptic Device

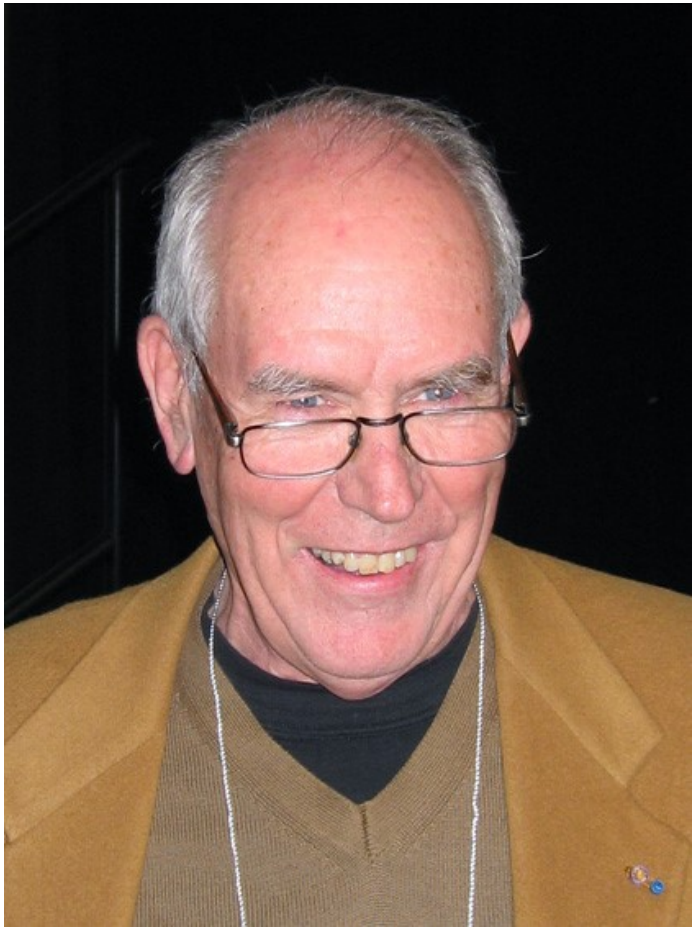
Camera sensors – Xbox, iPad

VR Remote – Motion Censor

Sketchpad (1962, Ivan Sutherland)

- Turing award (1988)





Wikipedia

Sketchpad (1962, Ivan Sutherland)

Alan Kay

True2Form (2014)

True2Form: 3D Curve Networks from 2D Sketches via Selective Regularization

Baoxuan Xu
William Chang
Alla Sheffer

University of British Columbia

Adrien Bousseau

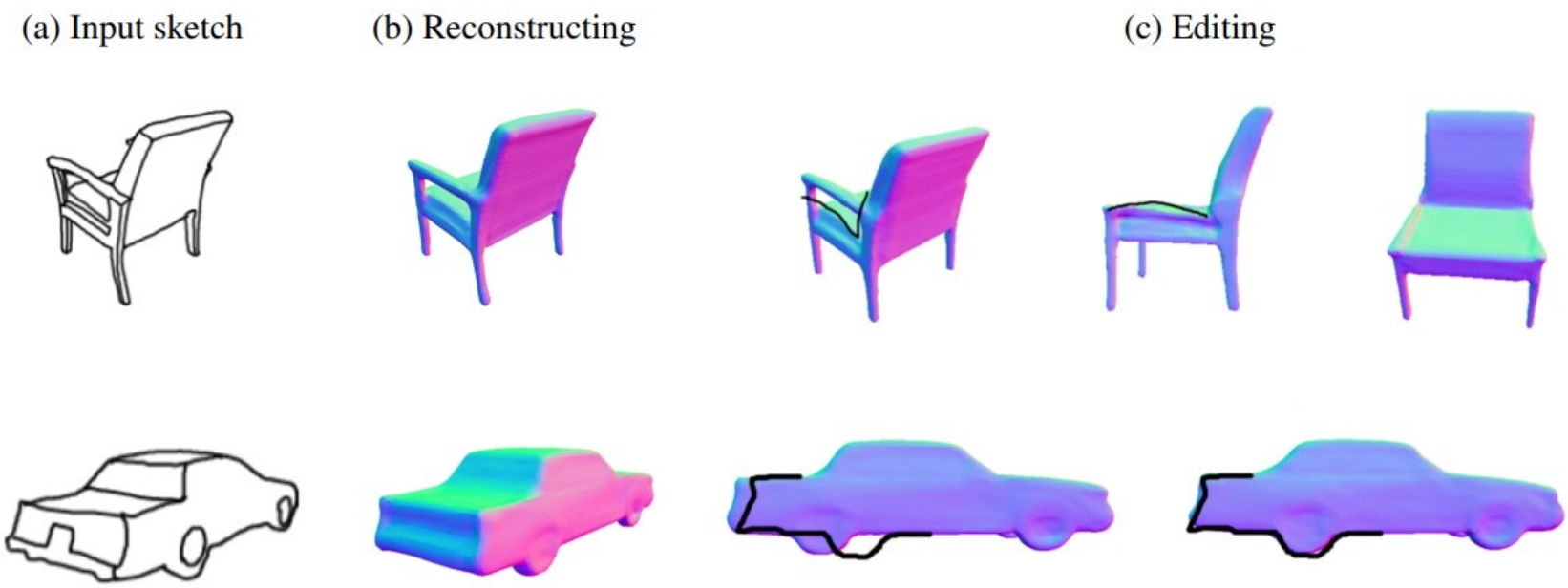
INRIA Sophia Antipolis

James McCrae
Karan Singh

University of Toronto

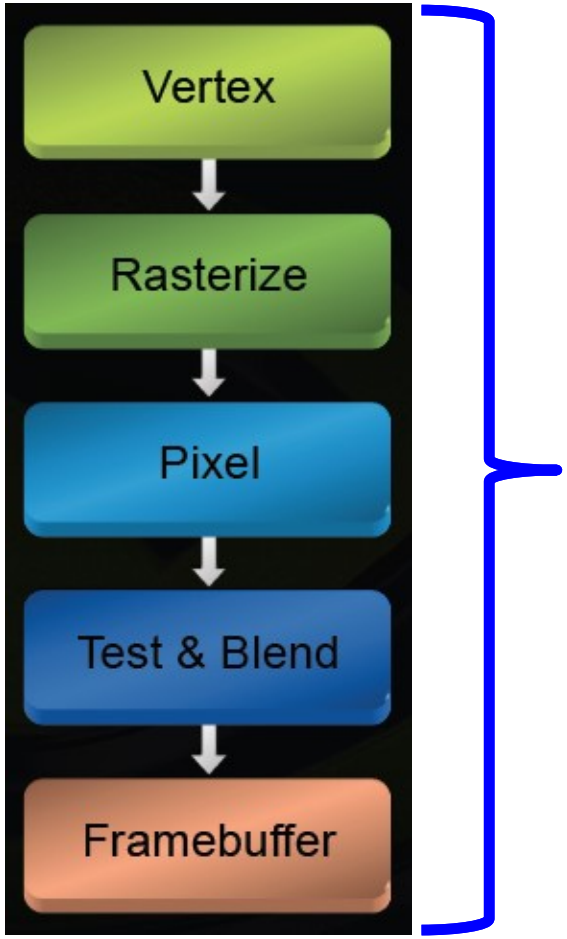
Sketch2Mesh (2021)

- Deep-learning-based



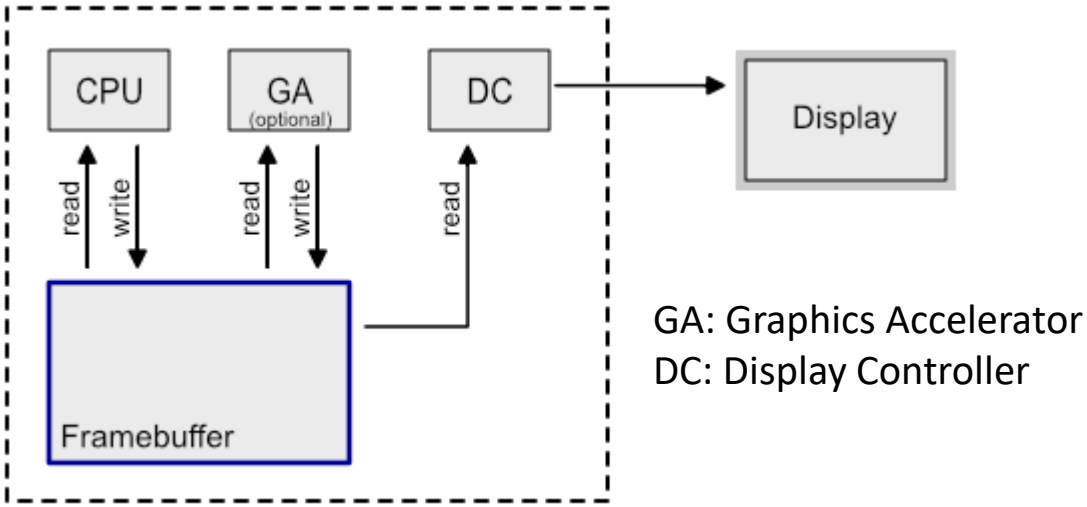
GPU (Graphics Processing Unit)

- Hardware dedicated to 3D graphics tasks



Framebuffer

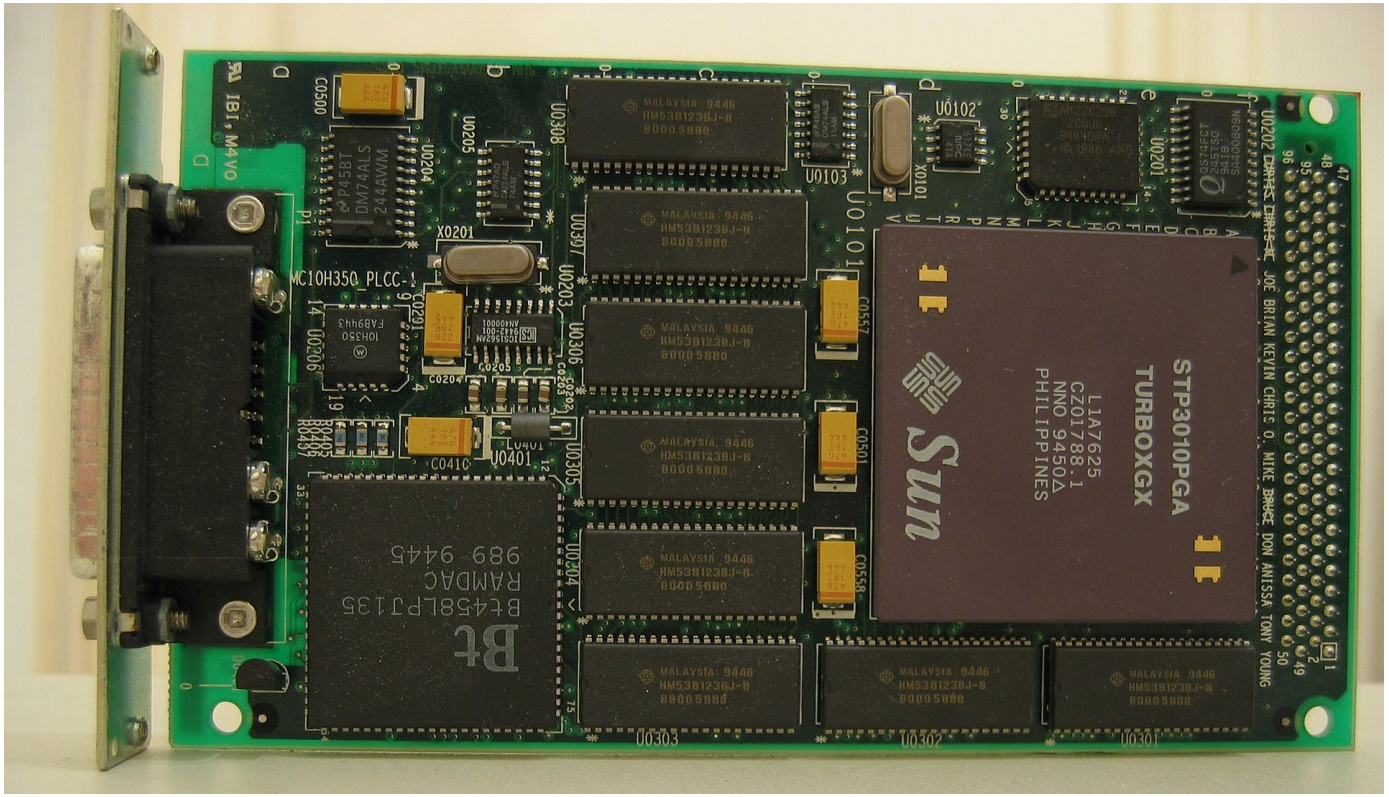
- A portion of memory containing a bitmap that drives a video display
  - Store entire screen-sized image
  - # of bits = color representation
    - 1 bit : black/white, 8 bit : 256 color
  - Required for raster graphics



GA: Graphics Accelerator
DC: Display Controller

https://doc.embedded-wizard.de/framebuffer-concepts?v=9.00

Framebuffer



Sun TGX Framebuffer

Raster Graphics

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

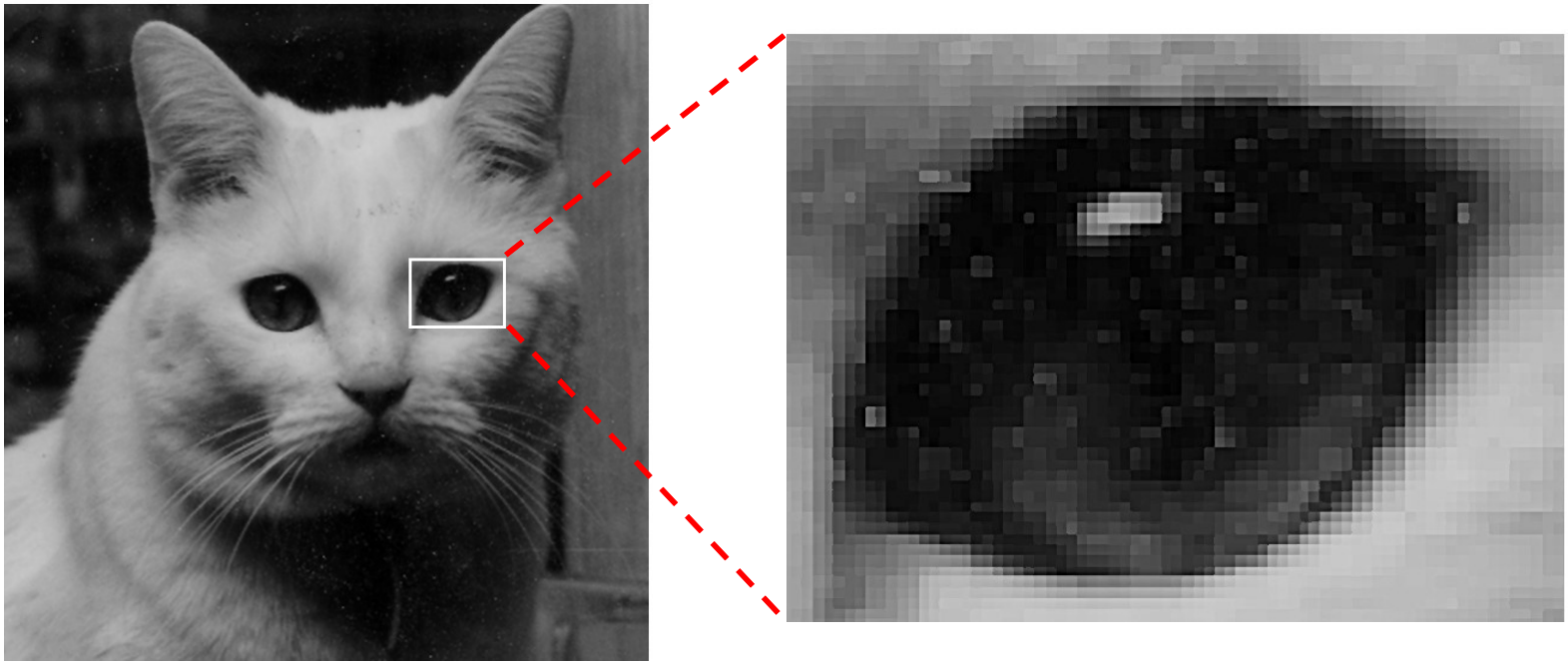
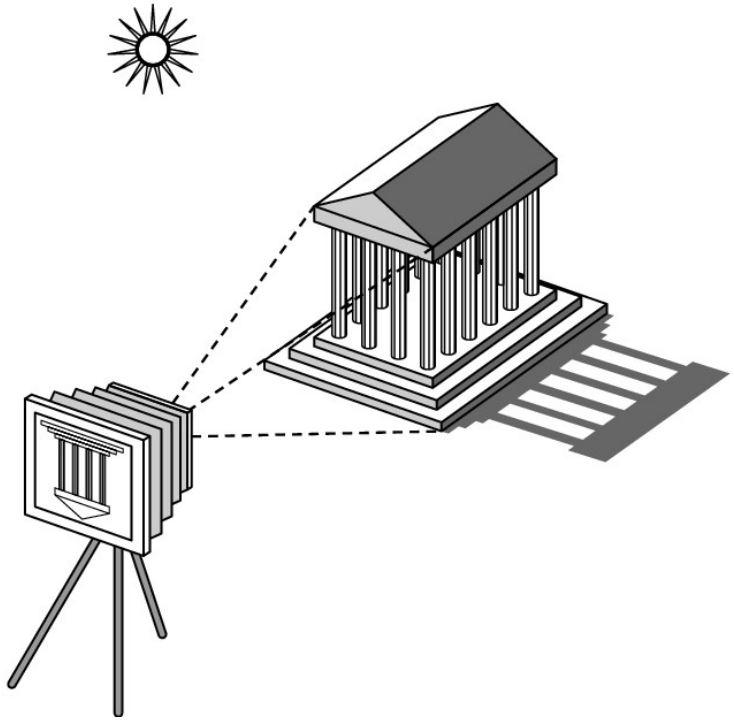


Image Formation

- In computer graphics, 2D images are formed using a process analogous to how images are formed by physical imaging systems
  - Cameras
  - Microscopes
  - Telescopes
  - Human visual system

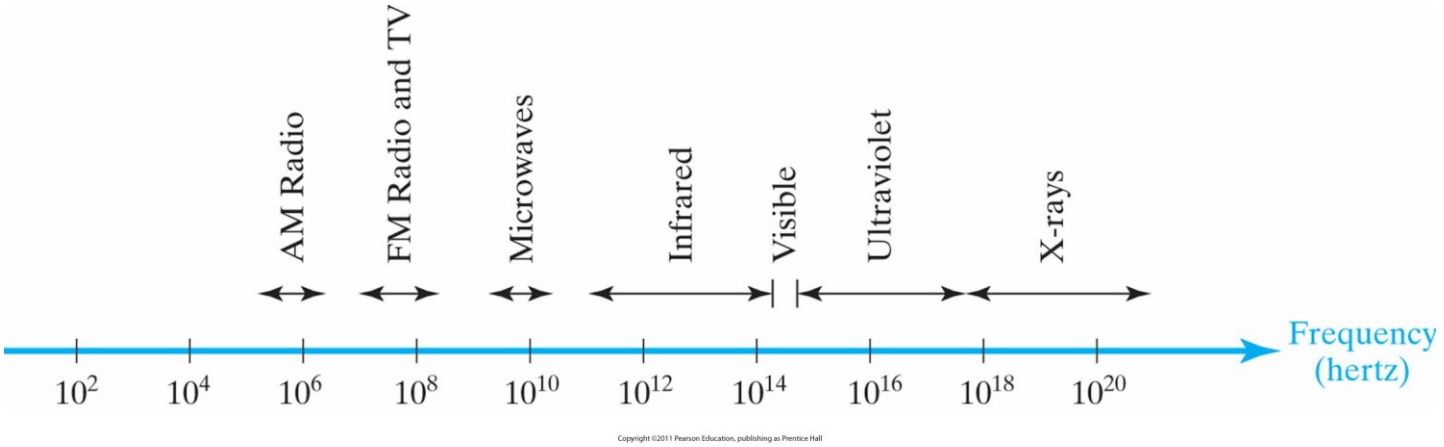
Element of Image Formation

- Objects
- Viewer
- Light sources



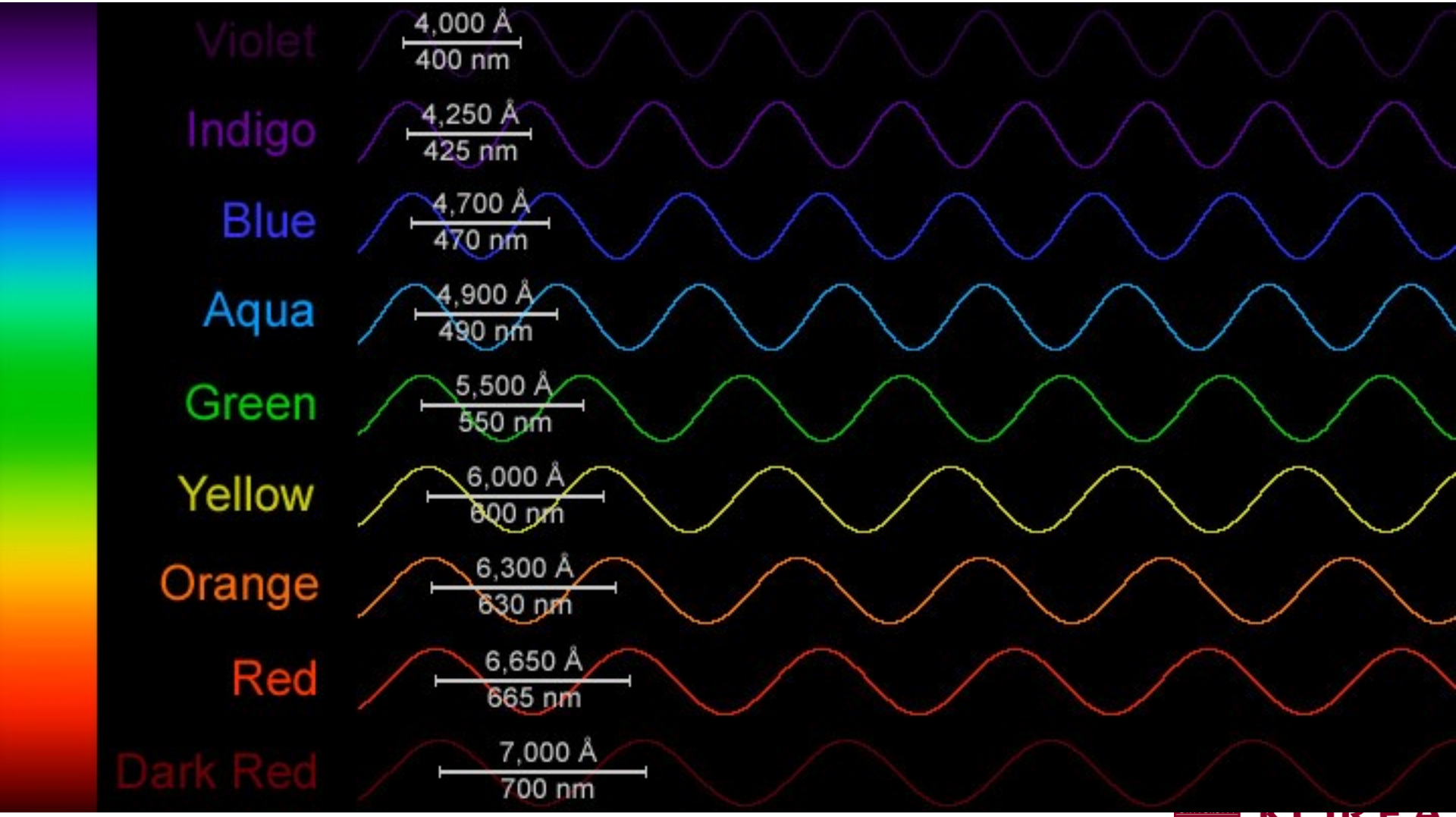
Light

- *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)



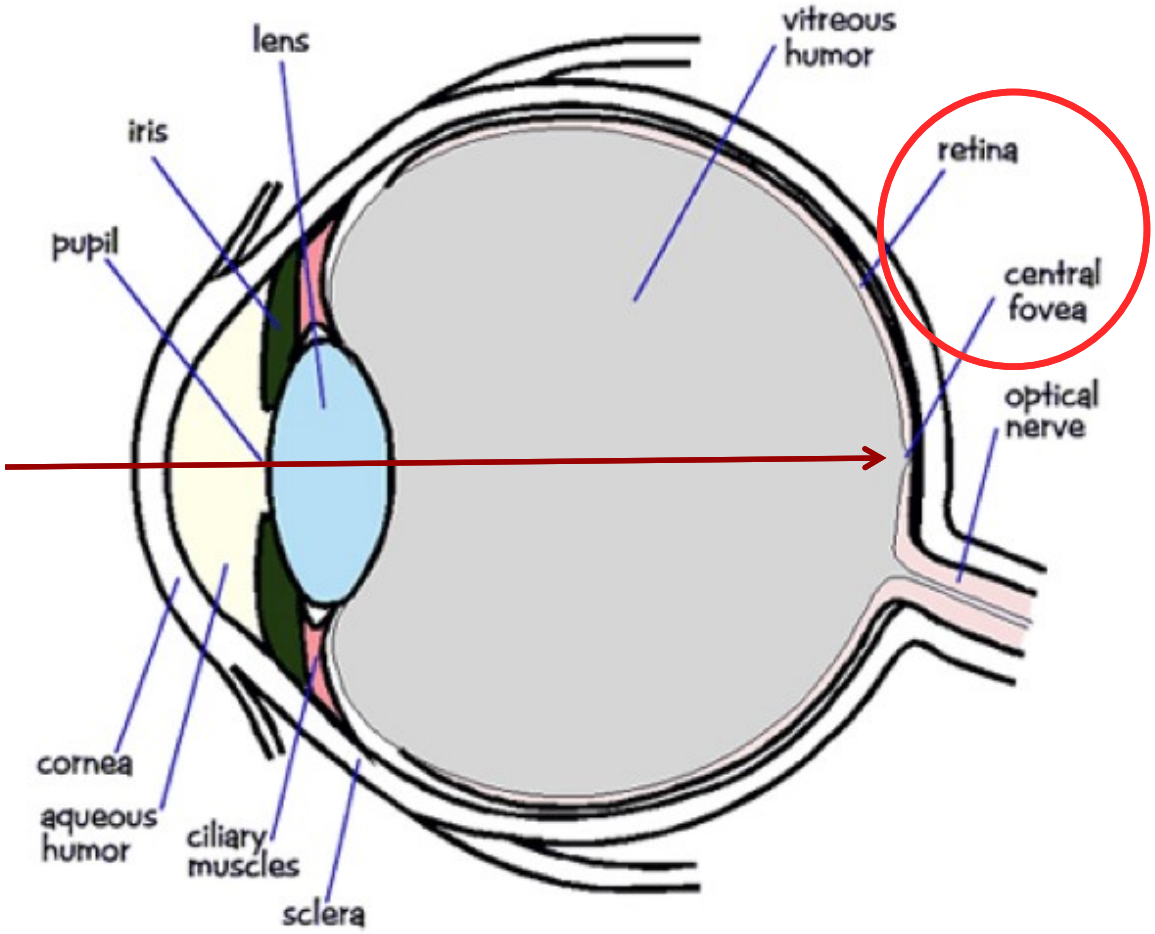
Adapted from Hearn

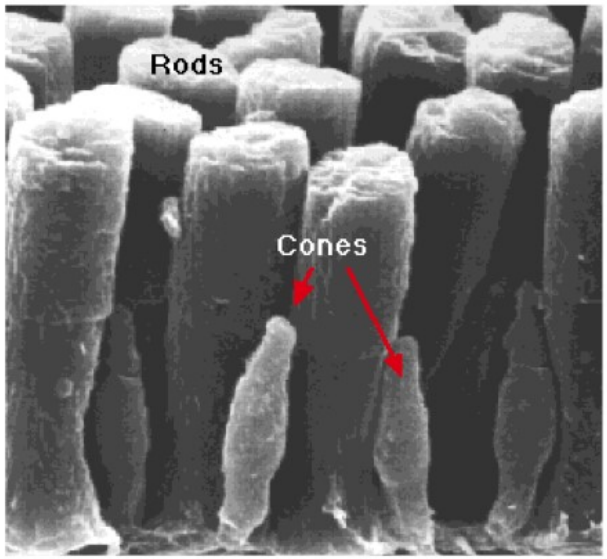
Wavelengths



Adapted from Russel

The Eye

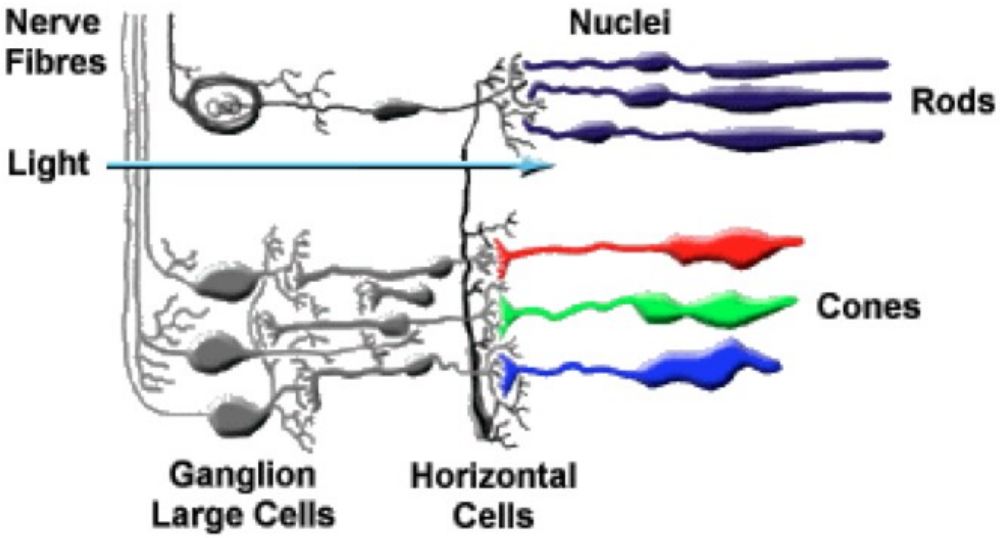




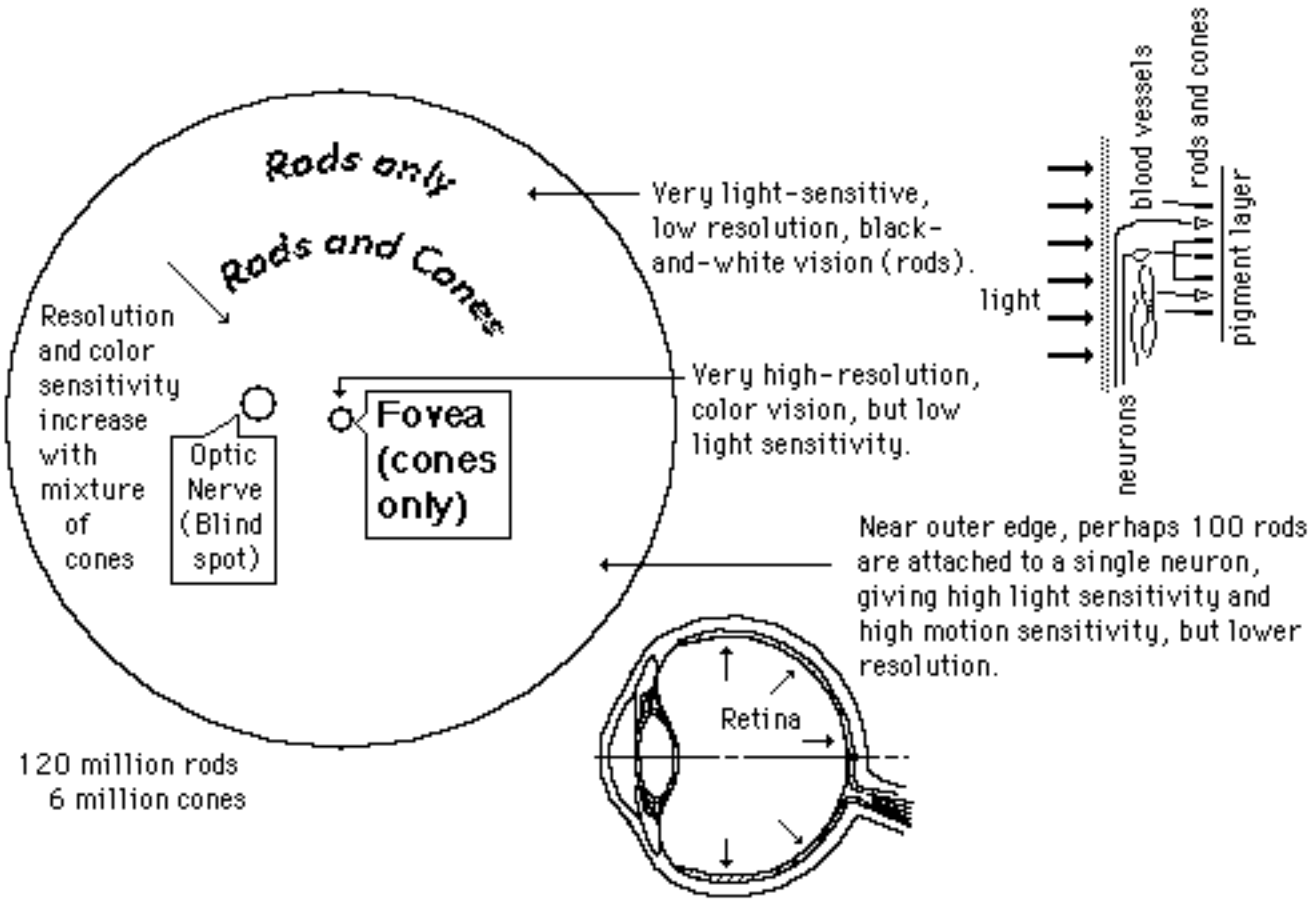
Adapted from Pfister

Two Types of Photoreceptors

- Rods
  - Monochromatic, night vision
- Cones
  - Color sensitive
  - Three types of cones



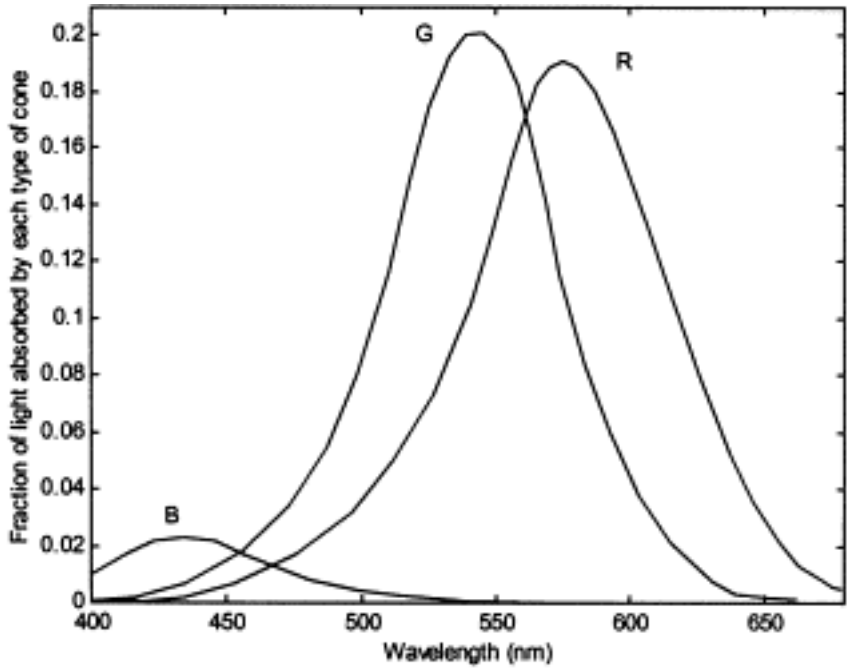
Retina



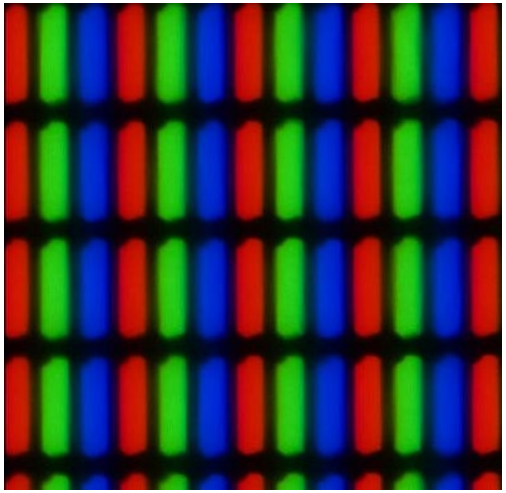
http://hyperphysics.phy-astr.gsu.edu/hbase/vision/retina.html#c1

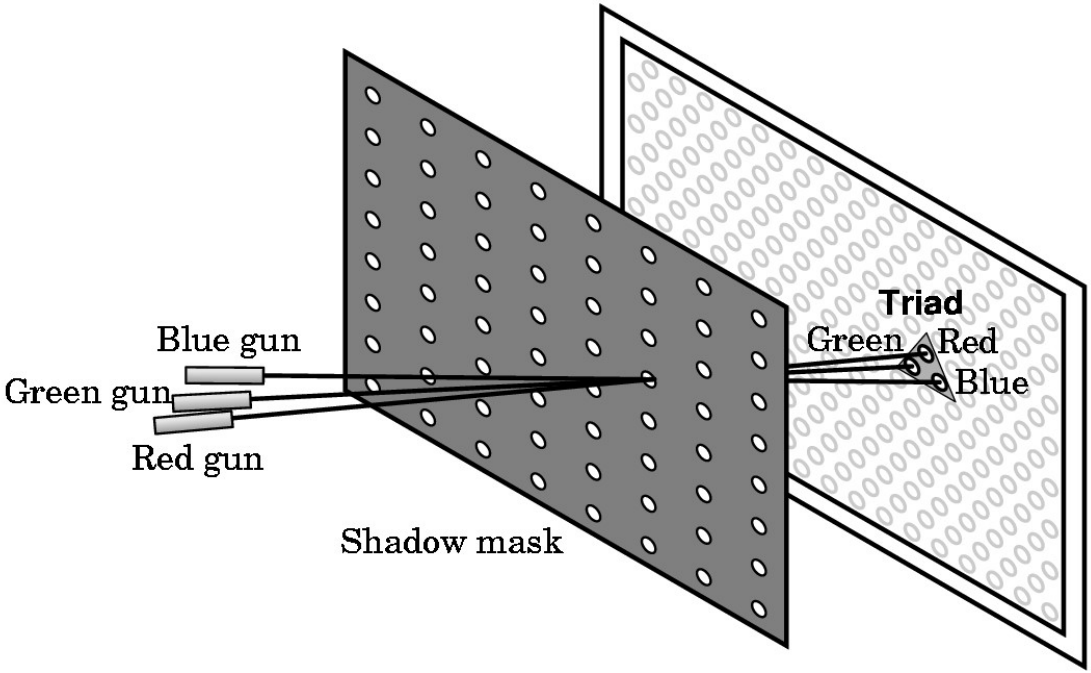
Three-Color Theory

- Only three values (the *tristimulus* values) from cones are sent to the brain
- Need only match these three values
  - Need only three *primary* colors



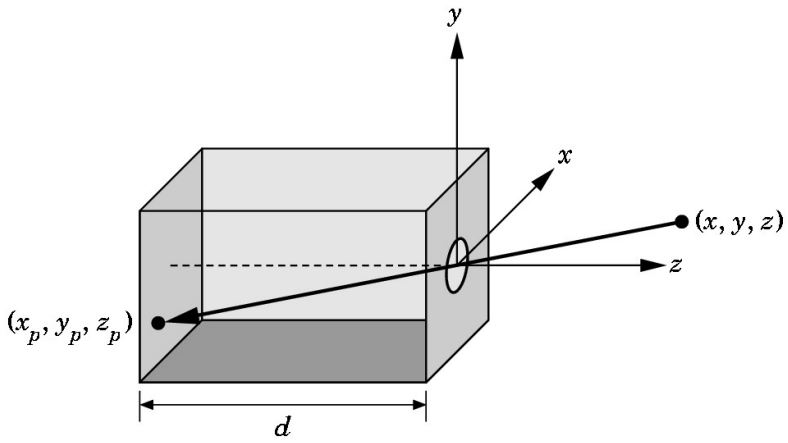
Shadow-mask CRT, LCD





Adapted from Angel

Pinhole Camera



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

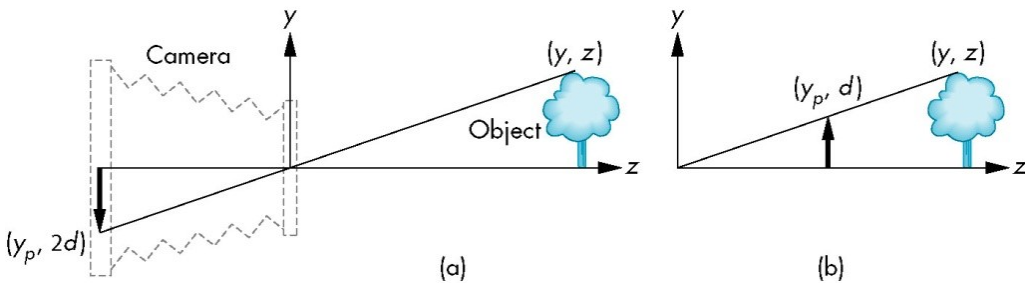
$x_p = -x/z/d$ $\quad y_p = -y/z/d$ $\quad z_p = d$

$z : x = -d : x_p$

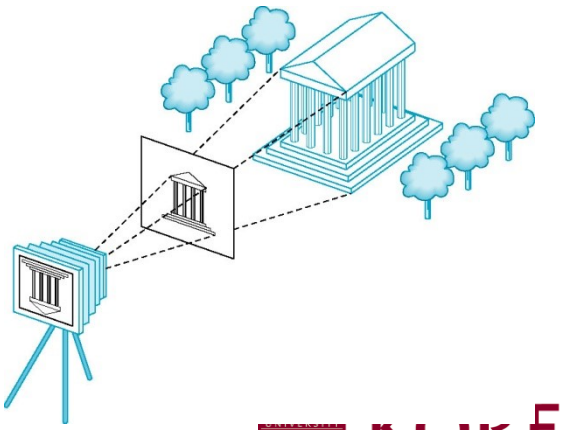
These are equations of simple perspective

Synthetic-Camera Model

- Projector
  - Line from the center of lens to a point on the object
- Center of Projection (COP)
  - The center of the lens
- Projection plane
  - Virtual image plane that are moved in front of the lens

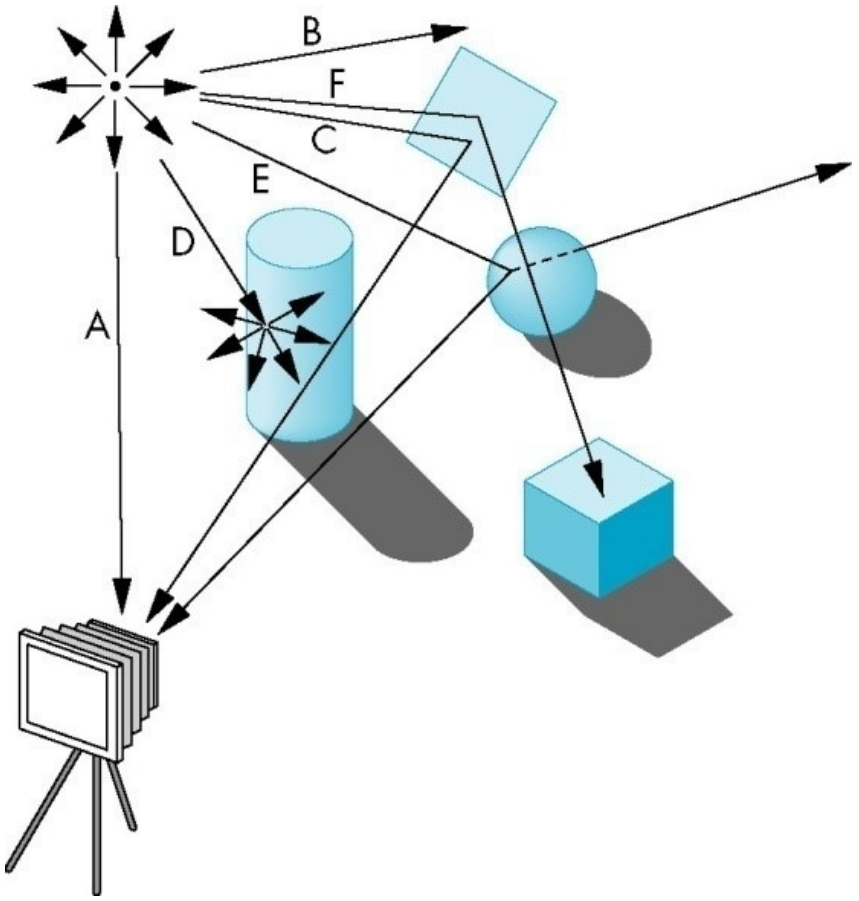


Equivalent views of image formation



Ray Tracing

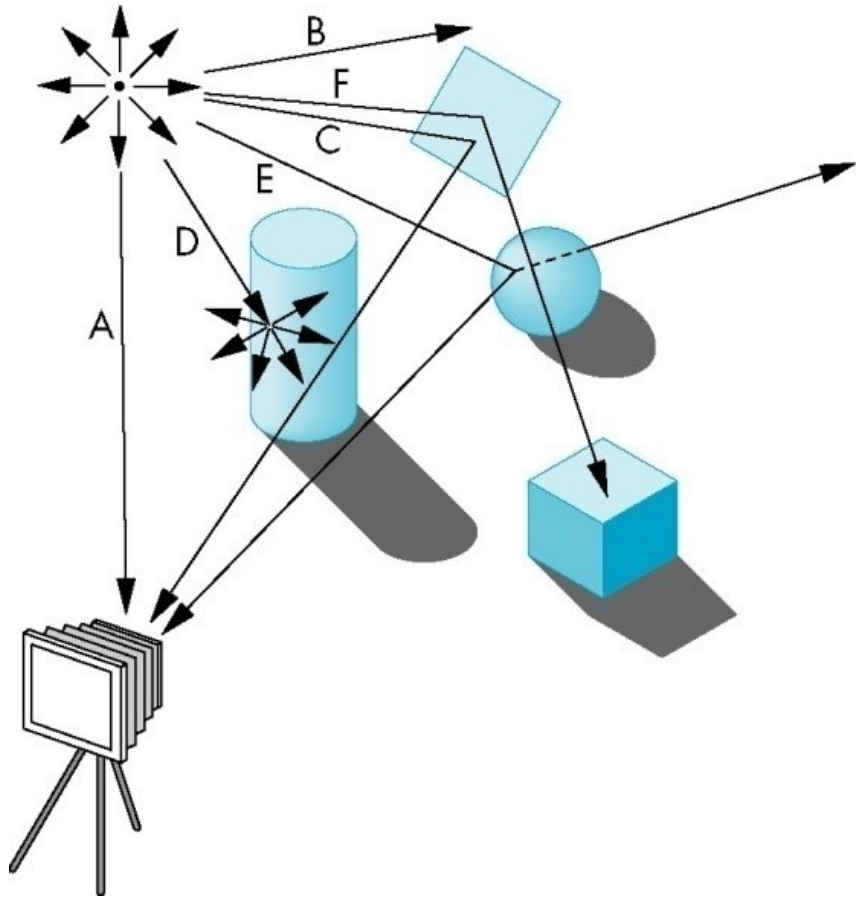
- Pros/Cons?



Angel

Ray Tracing

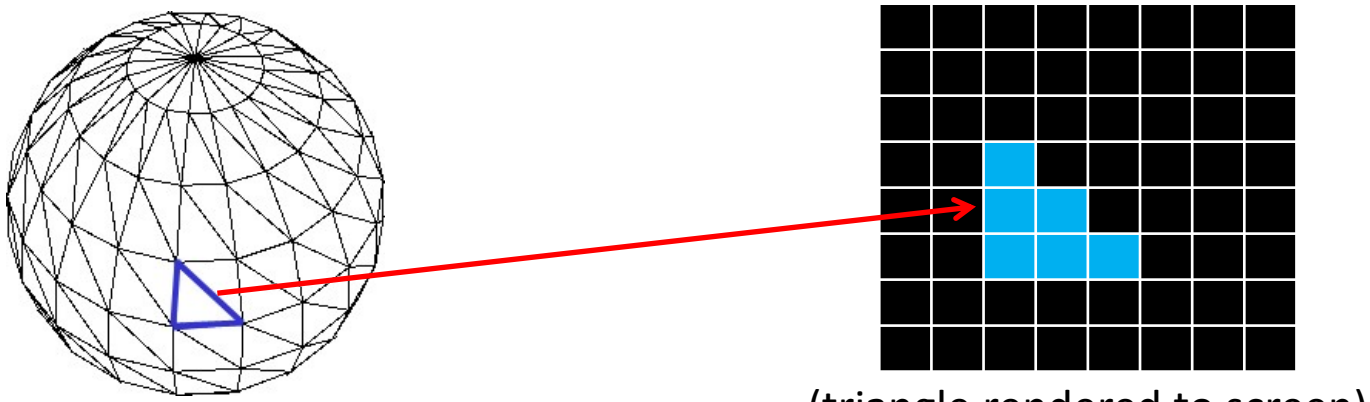
- Pros
  - Physically accurate
  - Global effect
    - Multiple reflection
    - Translucent
- Cons
  - Slow
  - Memory
    - Keep all data in memory



Angel

Rasterization

- Triangle to pixels
  - Process one triangle at a time



(triangle rendered to screen)

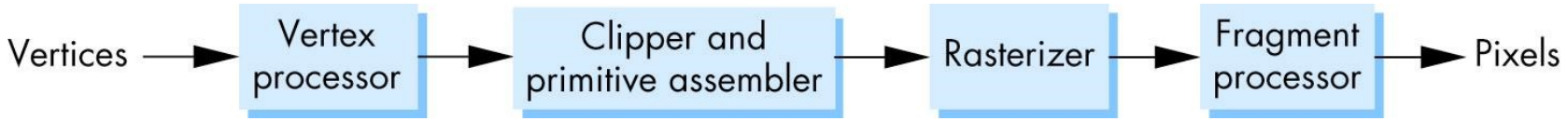
pros / cons ?

Rasterization

- Triangle to pixels
  - Process one triangle at a time
- Pros
  - Lightweight, no need to store entire data
  - Easy to implement in hardware
  - Parallel processing by pipelining
- Cons
  - No global lighting effect

Graphics Pipeline

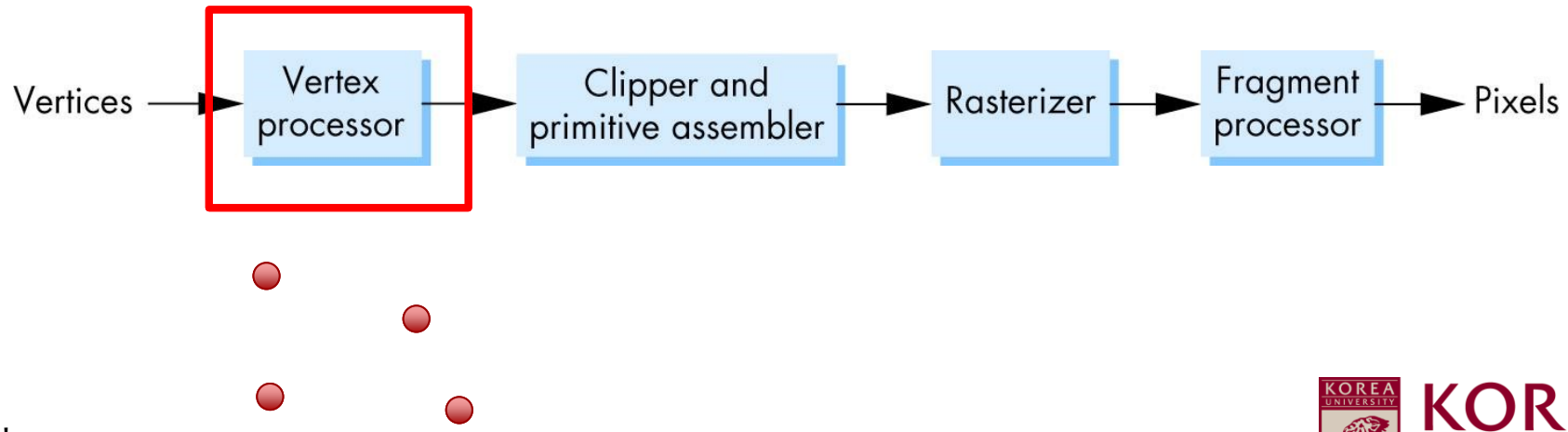
- Vertices to pixels



- Stages
  - Vertex processing
  - Clipping and primitive assembly
  - Rasterization
  - Fragment processing

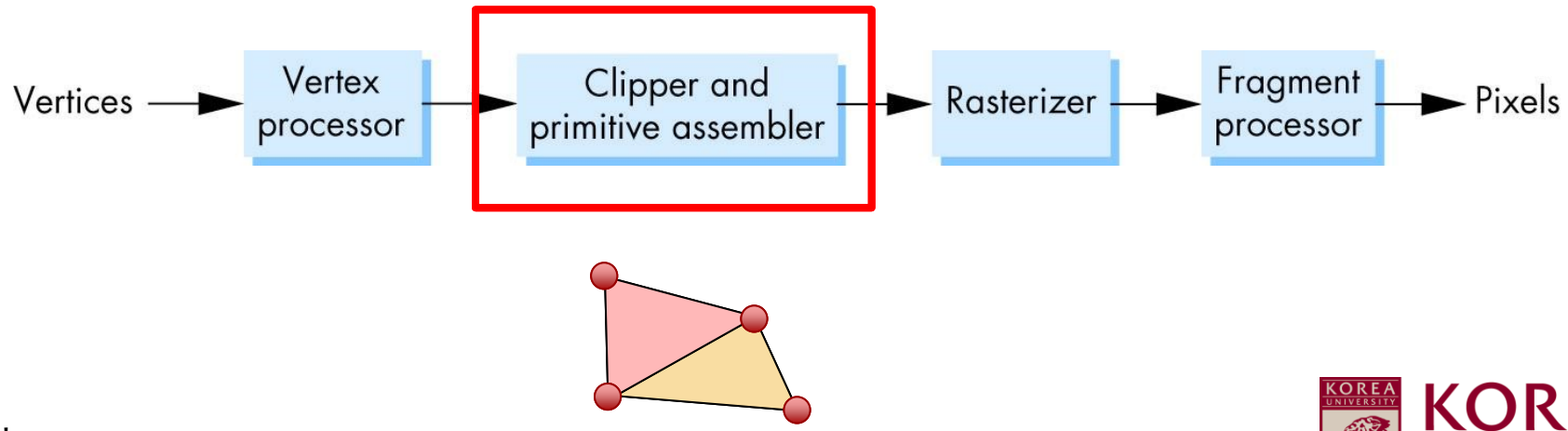
Vertex Processing

- Transform coordinates
  - Object coordinates
  - Camera coordinates
  - Screen coordinates
- Vertex color computation



Primitive Assembly

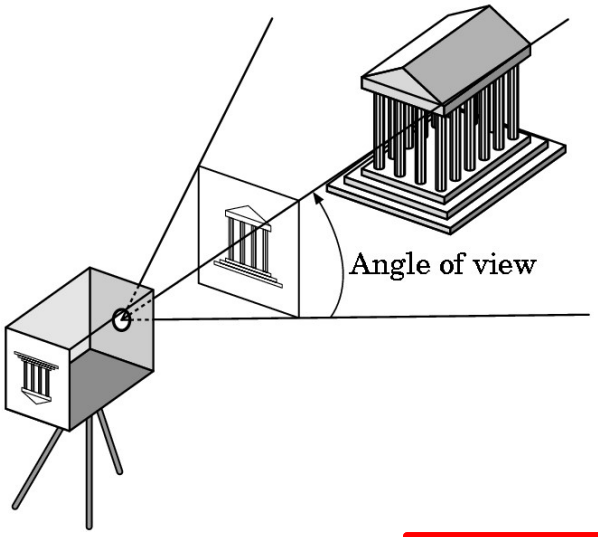
- Vertices to geometric objects
  - Line segments
  - Polygons
  - Curves and surfaces

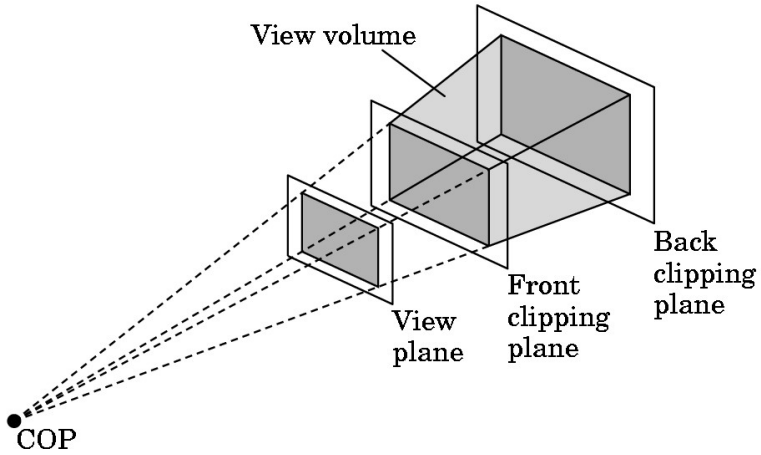


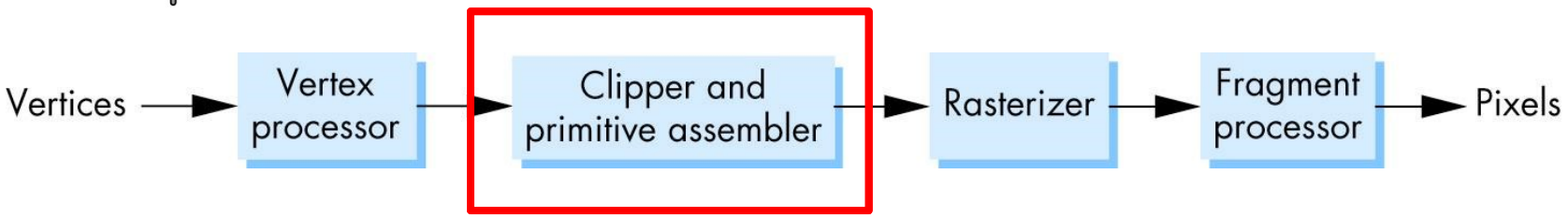
Angel

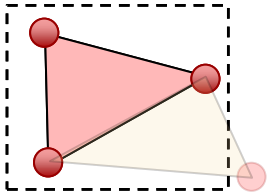
Clipping

- Discard invisible geometry





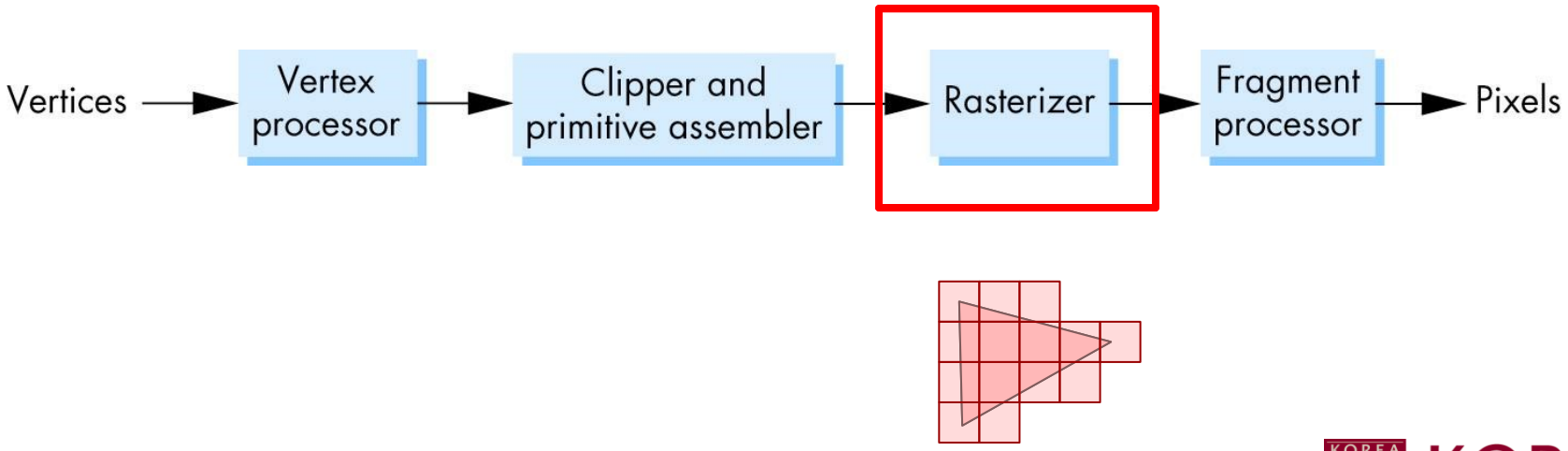




Angel

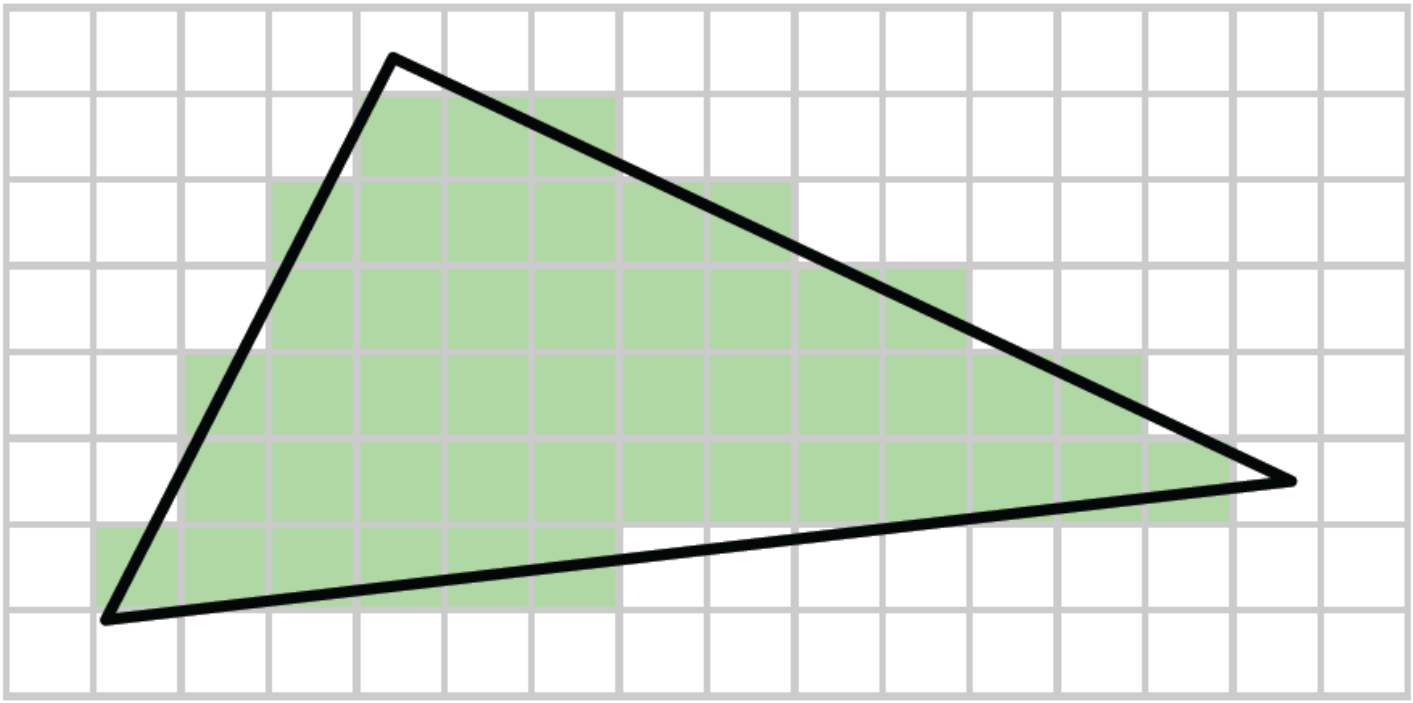
Rasterization

- If an object is not clipped out, the appropriate pixels in the frame buffer must be assigned colors
- Rasterizer produces a set of fragments for each object



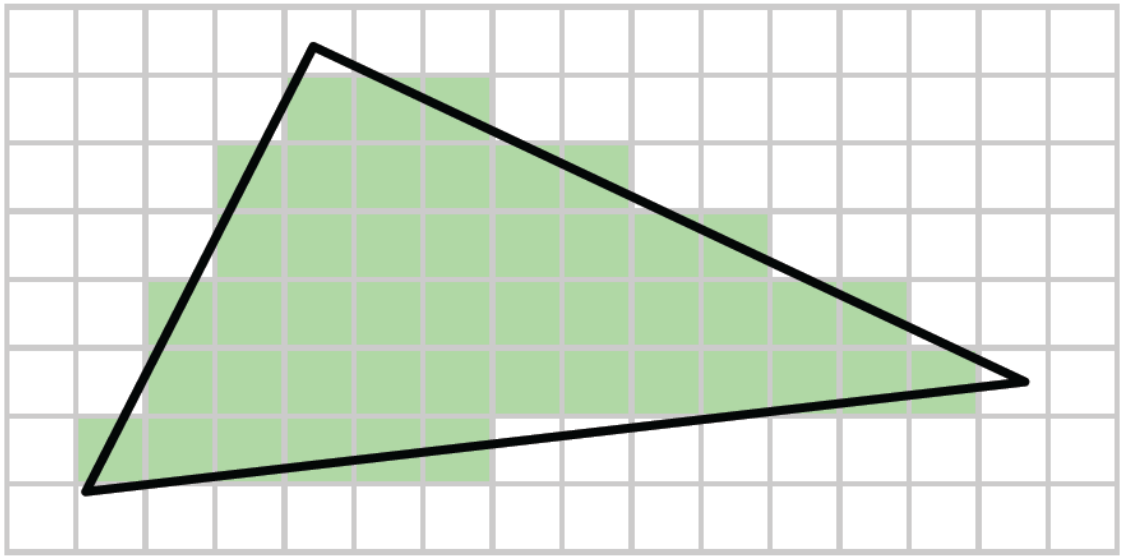
Rasterization (scan conversion)

- Creating fragments



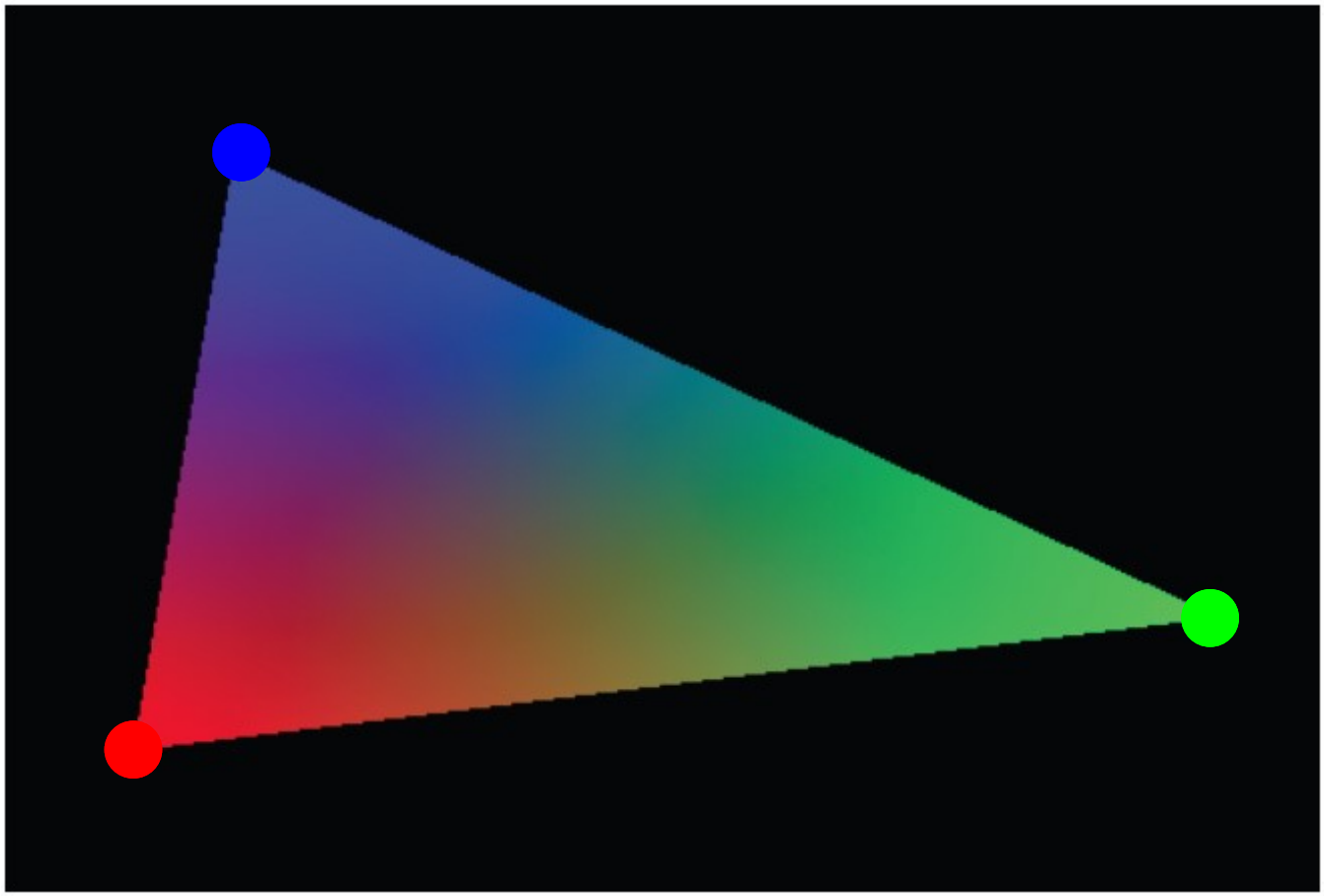
Fragments

- are potential pixel
- have a location in framebuffer
- have interpolated vertex attributes



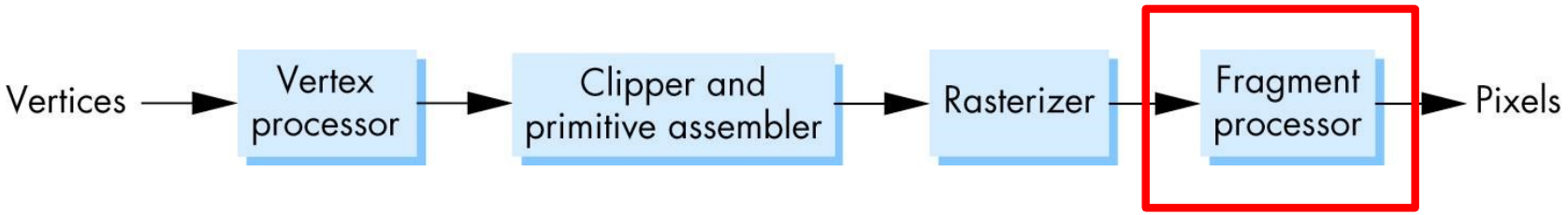
Attribute Interpolation

- Example: color



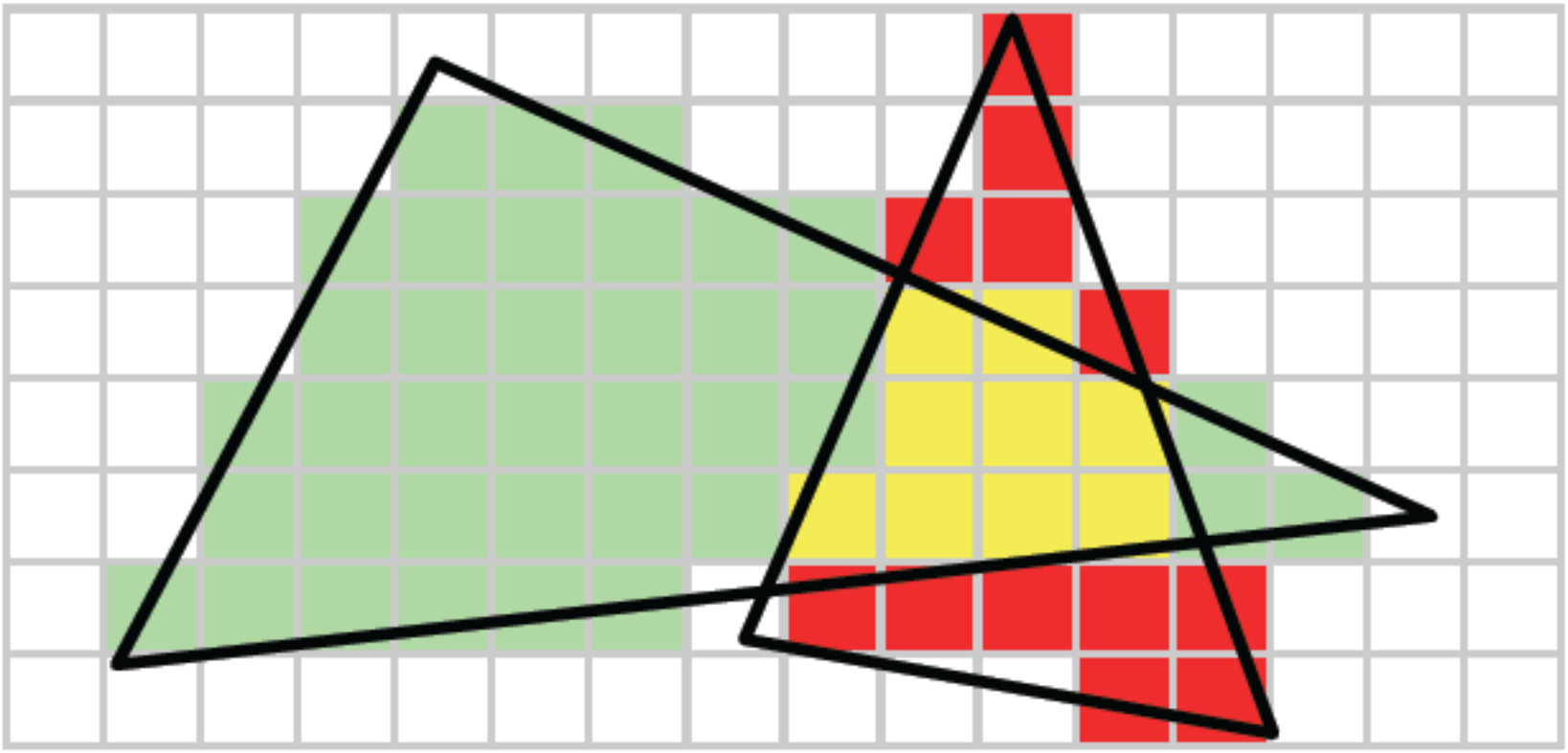
Fragment Processing

- Fragments are processed to determine the color of the corresponding pixel in the frame buffer
  - Texture mapping
  - Interpolation of vertex colors
- Depth test using Z-buffer



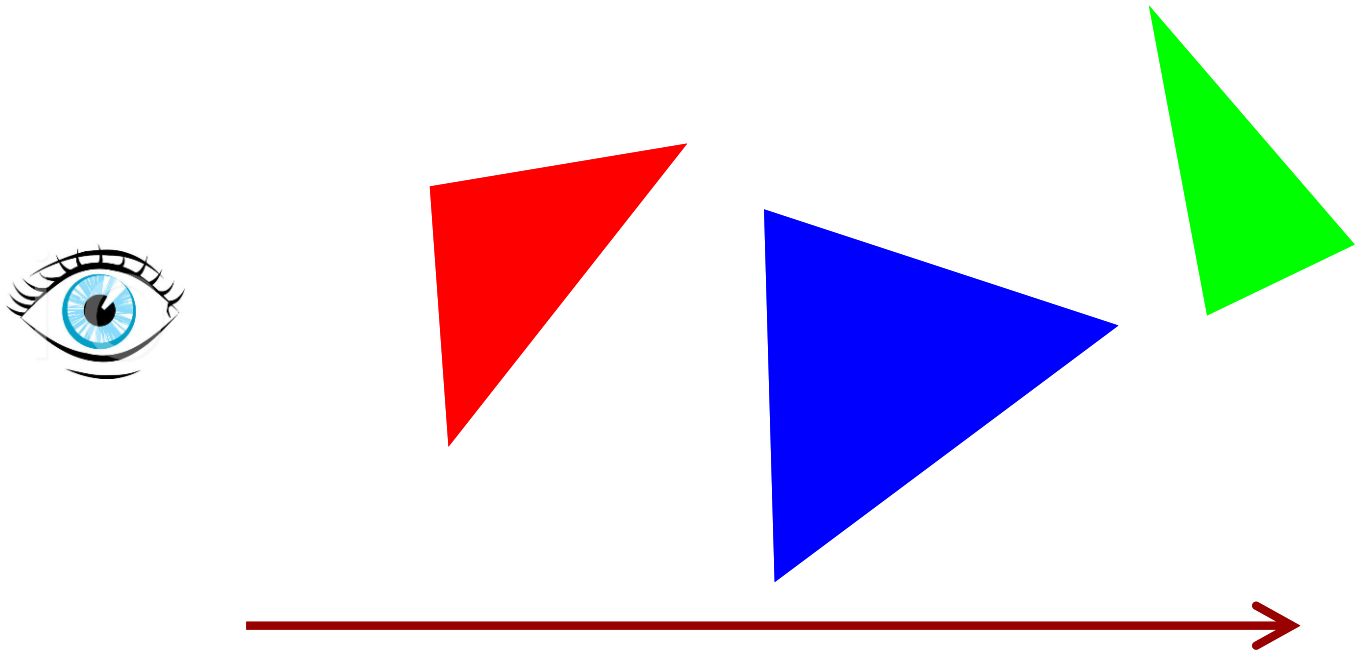
Fragments vs. Pixels

- Multiple fragments may assigned to a pixel

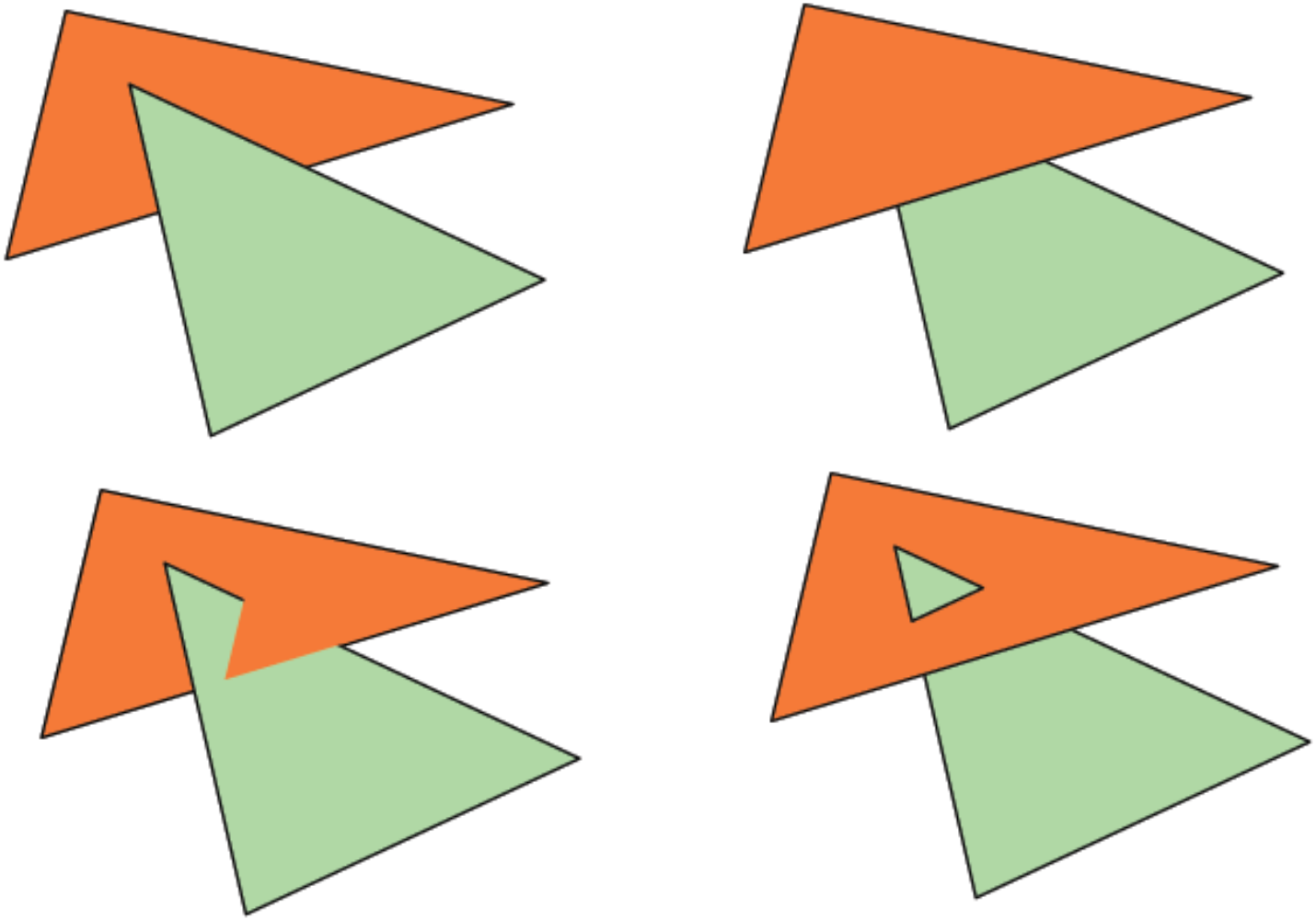


Painter's Algorithm

- Sort triangles along the view direction
- Draw triangles from back to front

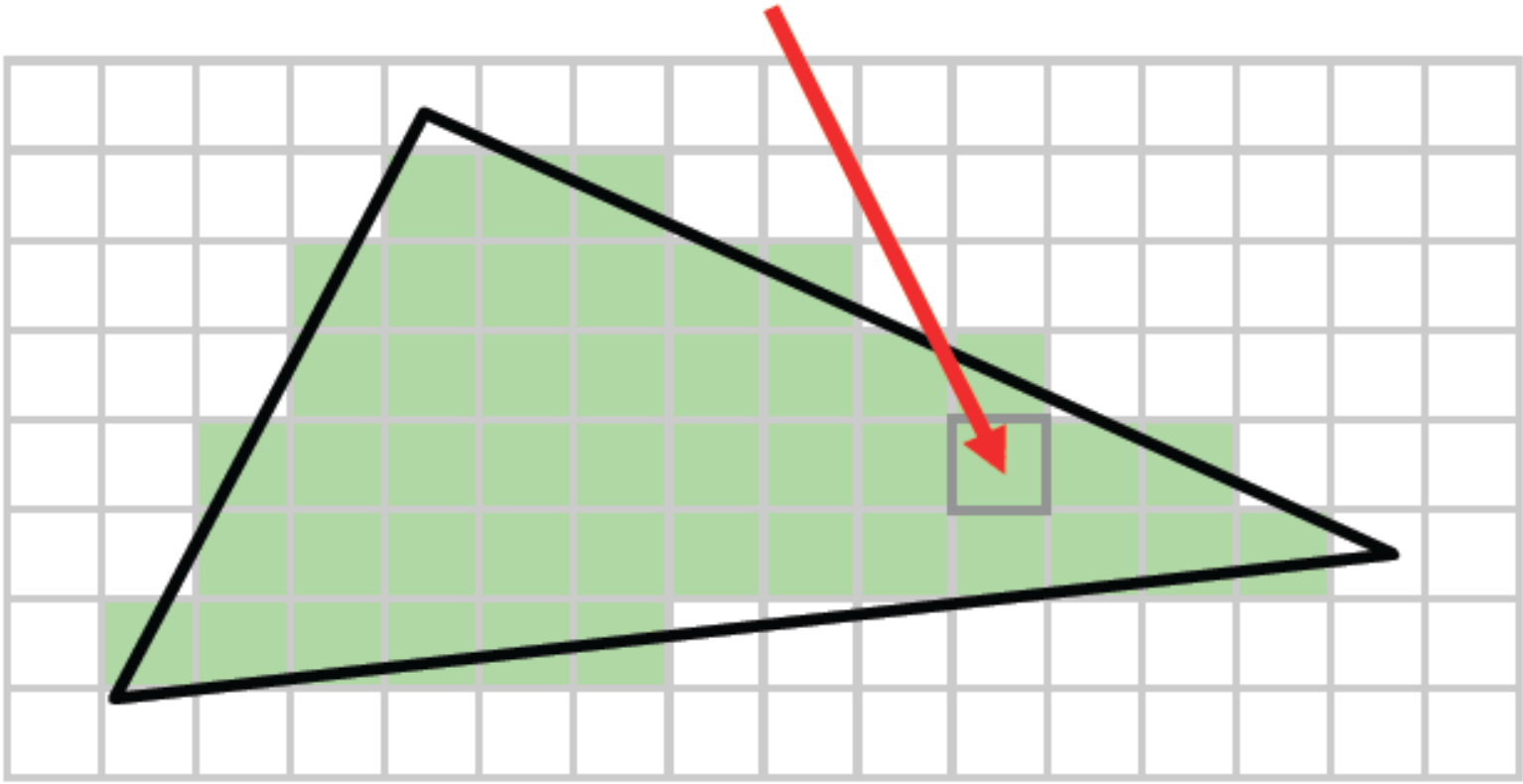


Which Triangle Wins?



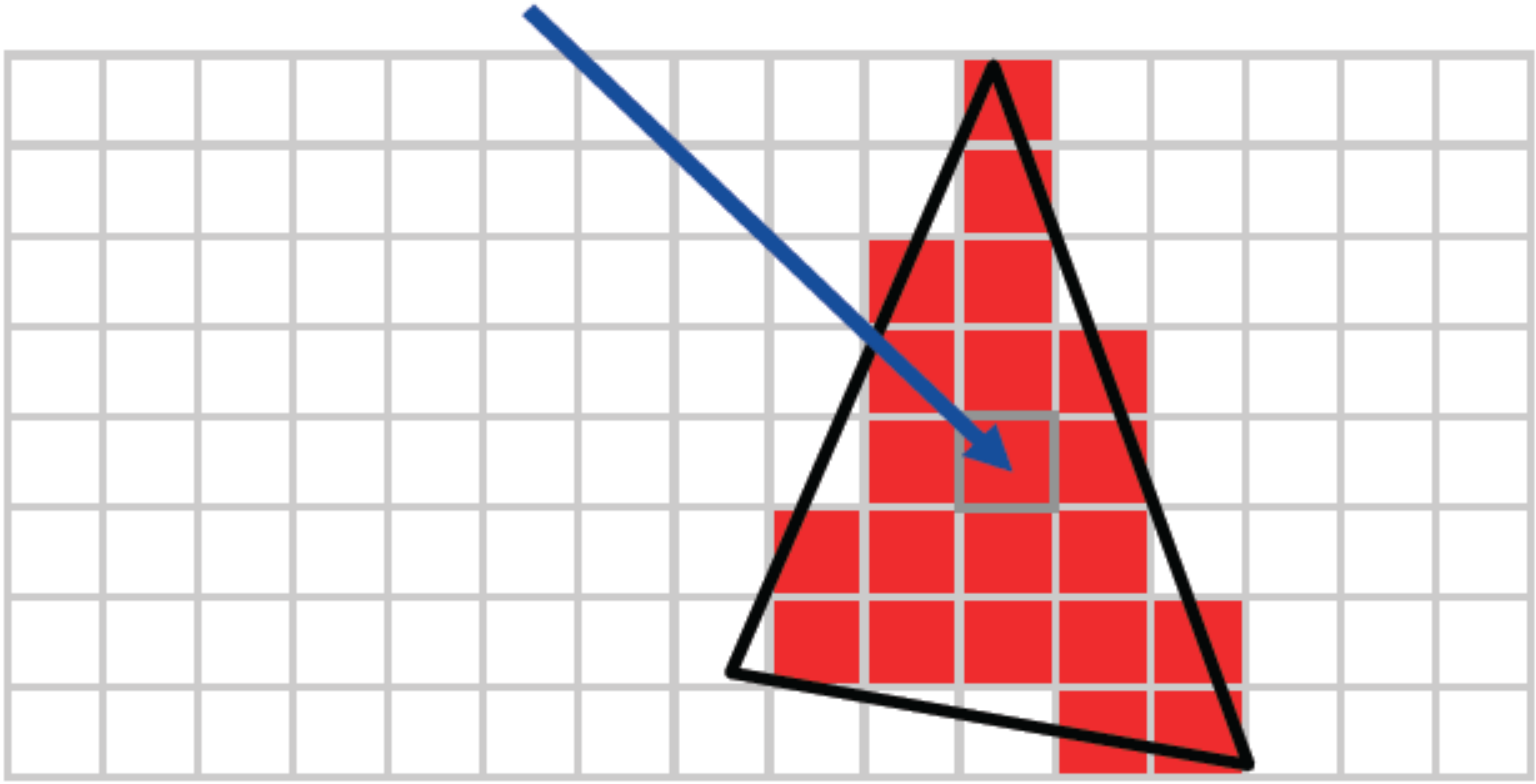
The Depth Buffer (Z-buffer)

- Example:
  - Fragment has z-value of 0.7



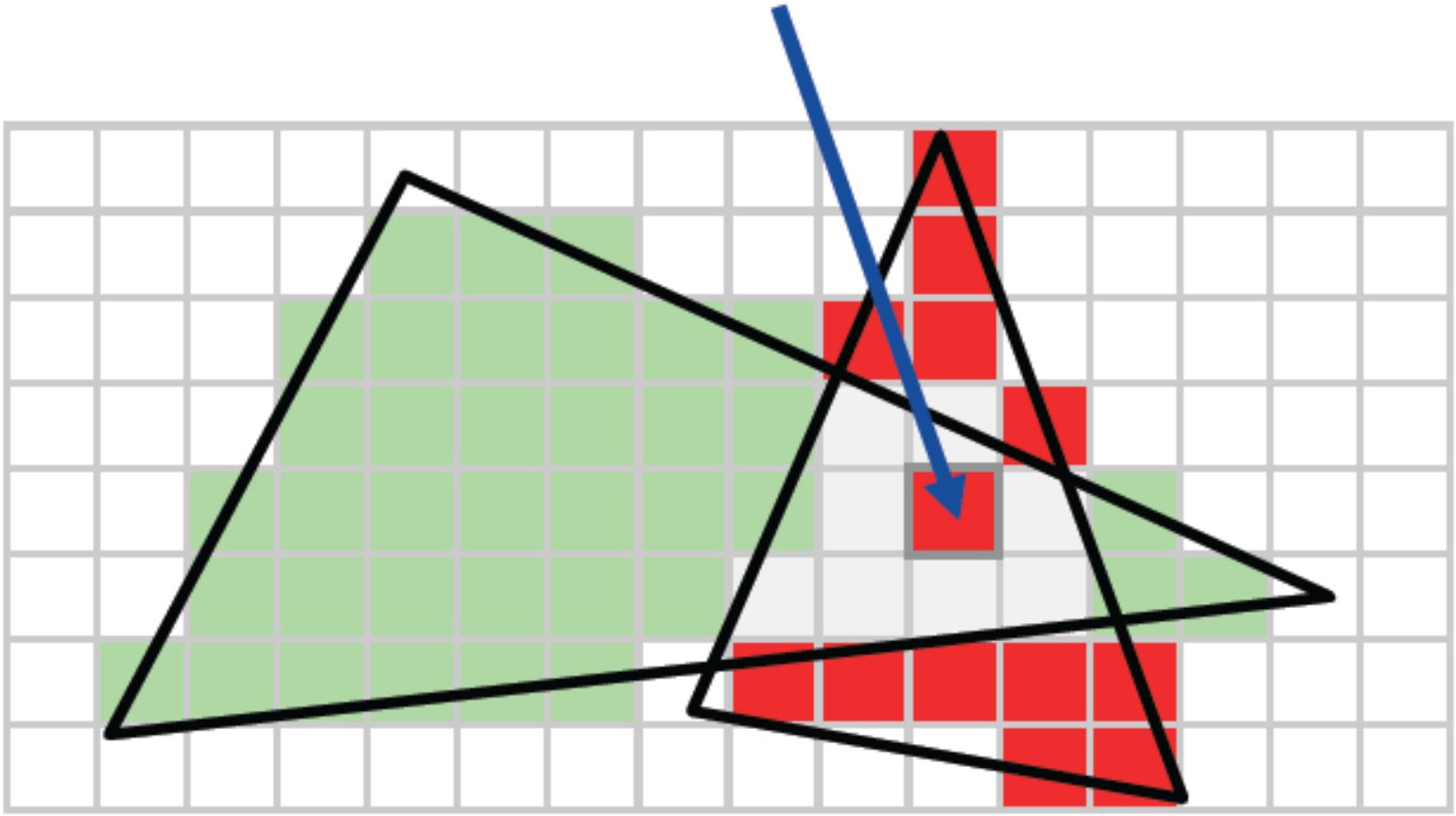
The Depth Buffer (Z-buffer)

- Example:
  - Fragment has z-value of 0.3



The Depth Buffer (Z-buffer)

- Since 0.3 < 0.7, the red fragment wins here



The Z-buffer

- Many fragments map to the same pixel location... how to efficiently track them?
  - Distance table

1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.1	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.1	0.1	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.2	0.3	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.3	0.3	0.4	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.3	0.4	0.4	0.5	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.4	0.4	0.5	0.5	0.5	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.4	0.5	1.0	1.0	1.0

Questions?

Gary's Game, Pixar