Computer Graphics Hardware Pipeline

Visual Imaging in the Electronic Age

Prof. Donald P. Greenberg

October 22, 2020

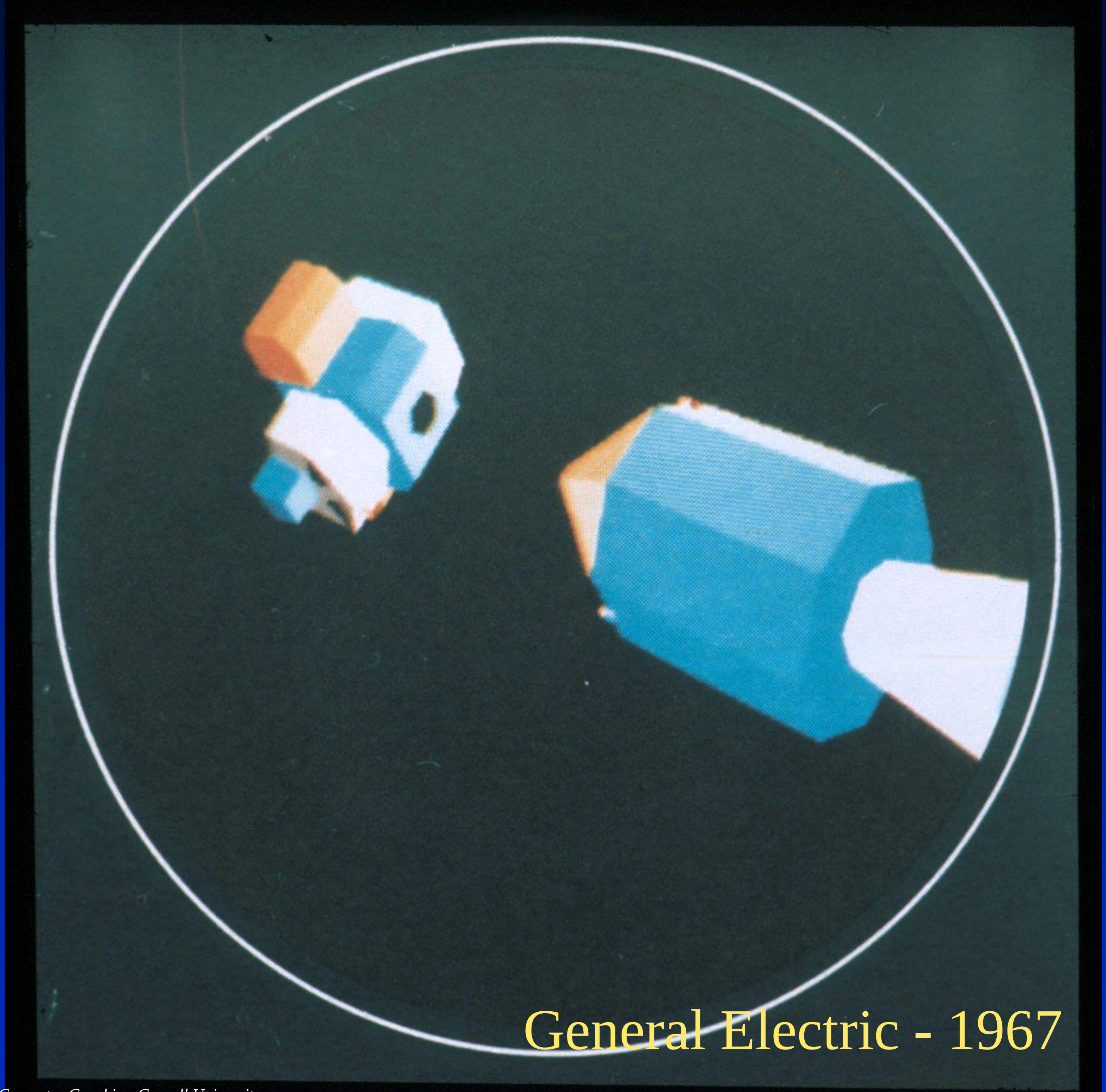
Lecture 15

Moore's Law

"Chip density doubles every 18 months."

Processing Power (P) in 15 years:

$$P = P_{today}(2)^{\frac{15 years}{18\, months}} = P_t(2)^{\frac{15}{1.5}}$$
$$= P_t(2)^{10} = 1000P_t$$



General Electric - 1967

Program of Computer Graphics, Cornell University

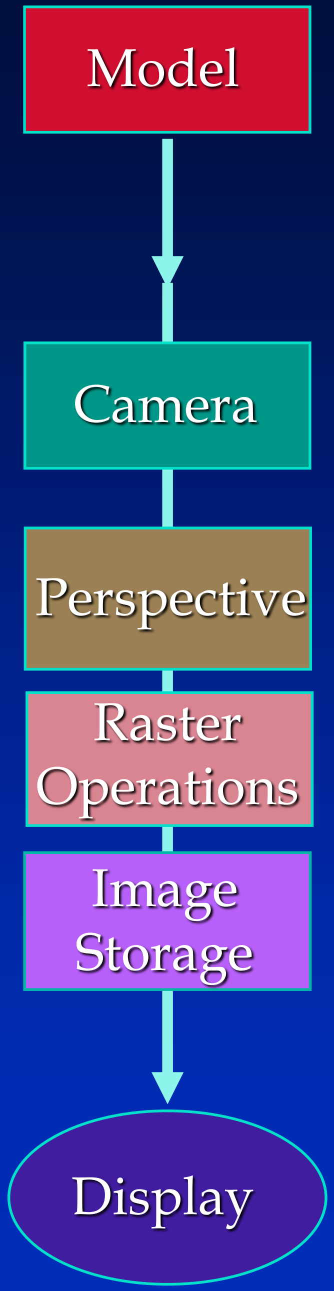
Xerox Sigma 5 1965-7

- Cost: $300,000
- 16K Magnetic memory

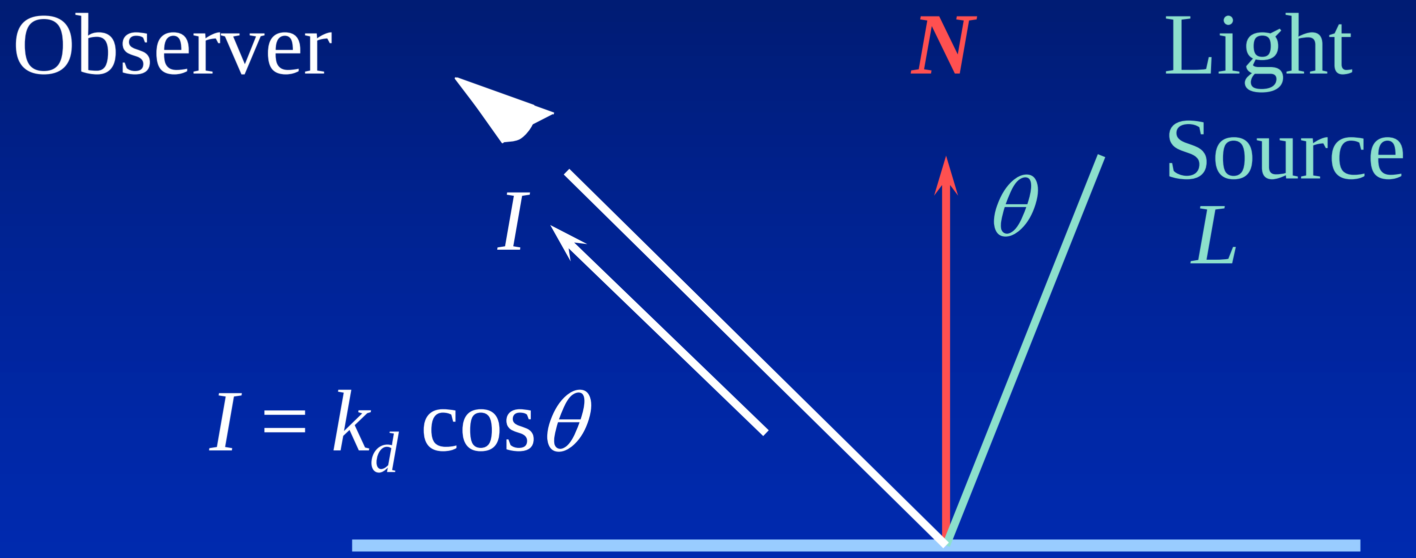
Cornell in Perspective Film

Program of Computer Graphics, Cornell University

Direct Illumination

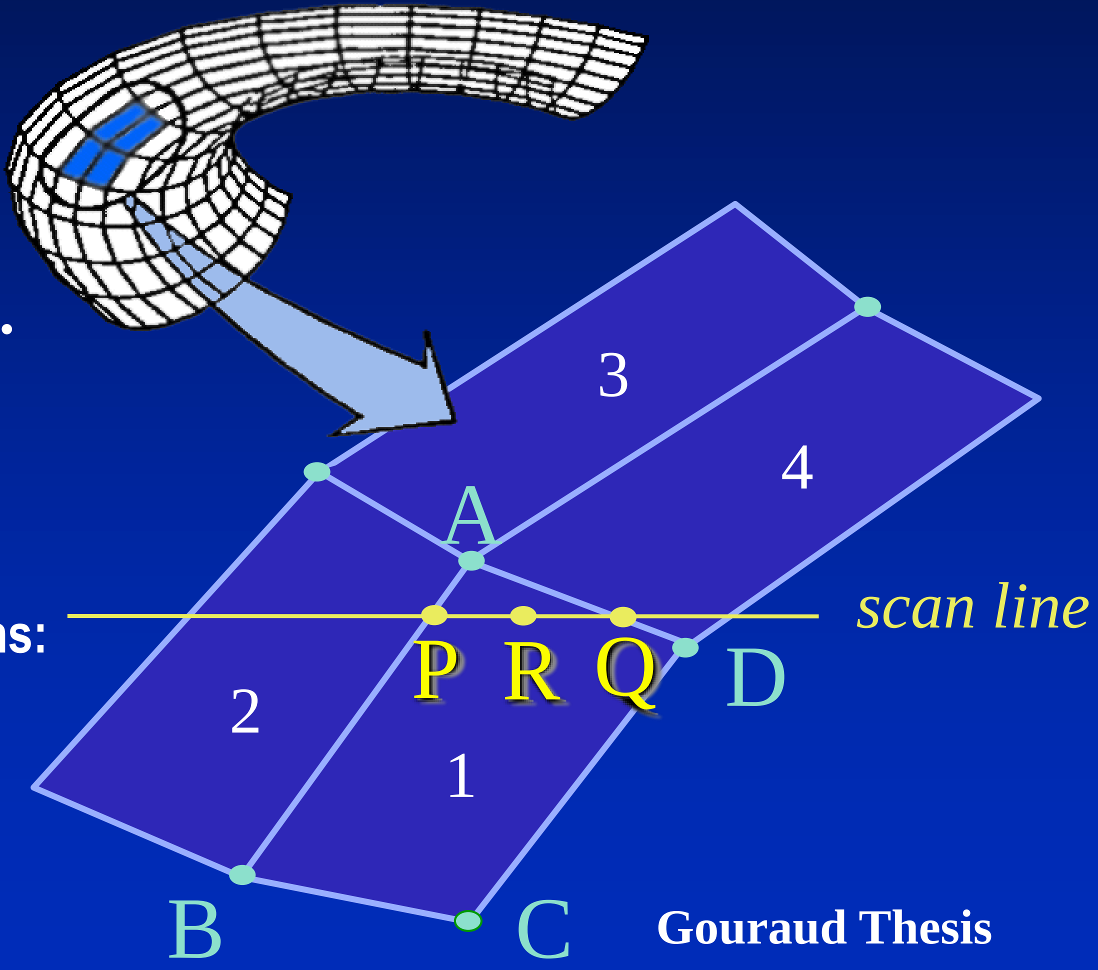


Diffuse Reflections



Gouraud Smooth Shading

Four polygons
approximating a surface
in the vicinity of point A.



Gouraud Thesis

The shading at point R is computed as

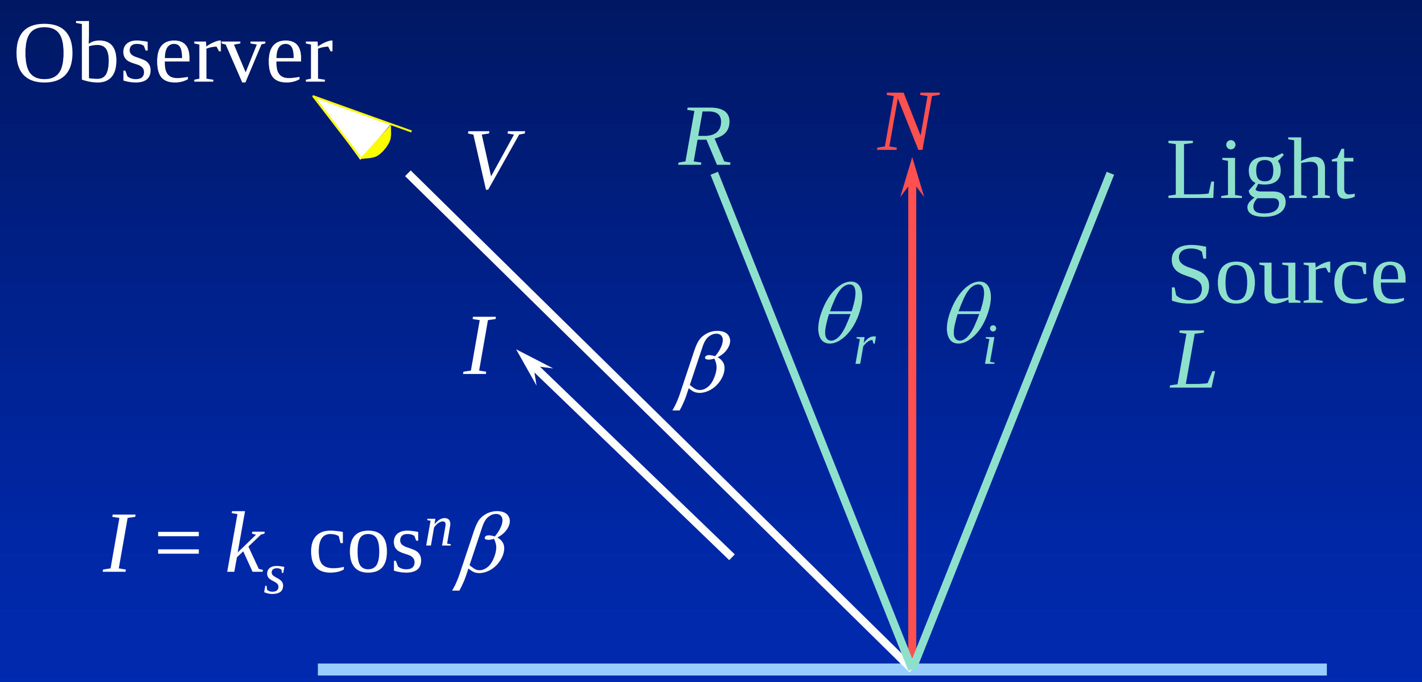
two types of successive linear interpolations:

across polygon edges: P between A and B,

Q between A and D;

across the scan line: R between P and Q.

Phong Model Specular Reflection



$$I = k_s \cos^n \beta$$

Phong Smooth Shading

Four polygons
approximating a surface
in the vicinity of point A.

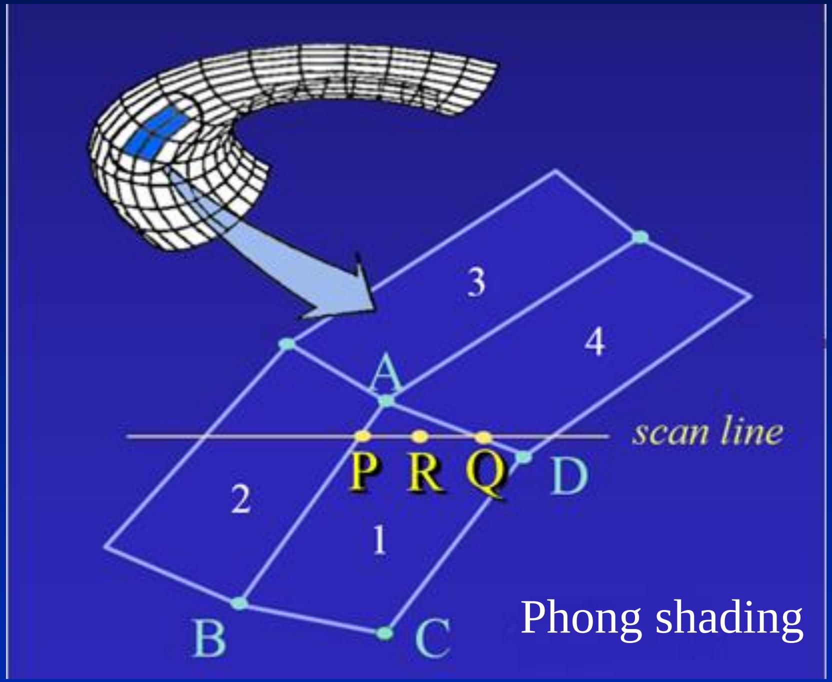
The shading at point R is computed as

two types of successive linear interpolations:

across polygon edges: P between A and B,

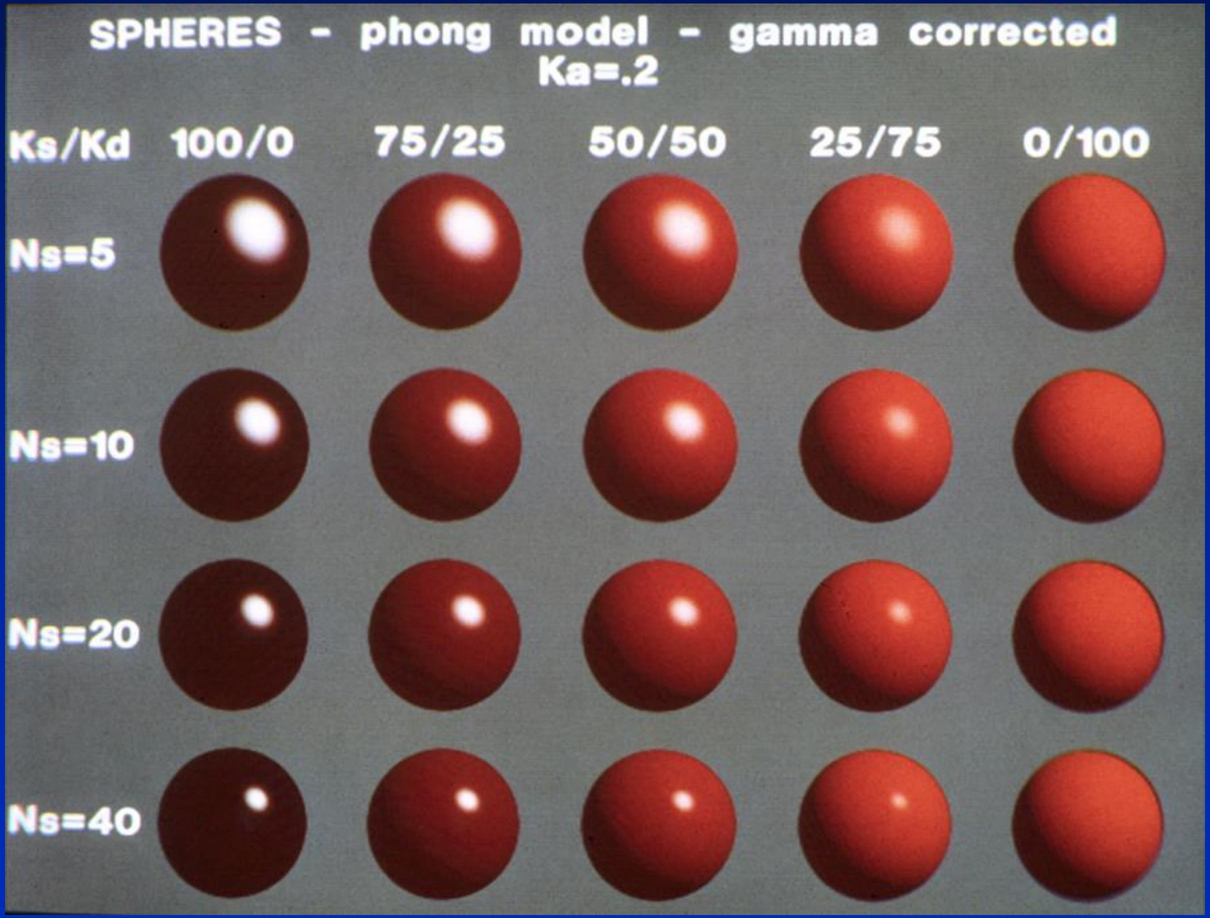
Q between A and D;

across the scan line: R between P and Q.



The only difference between Gouraud and Phong shading is that the colors at A, B, C, and D need to be computed to include the specular reflections.

Direct Lighting Phong Reflection Model



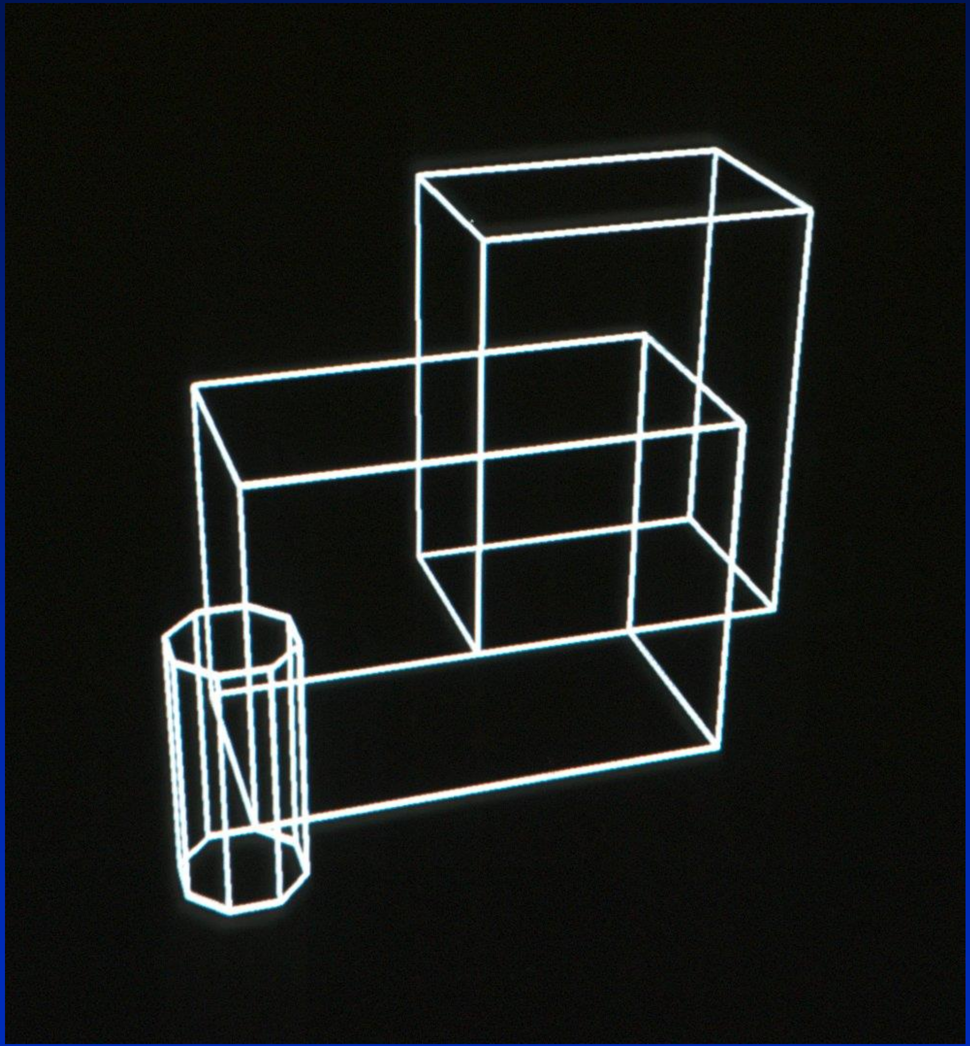
End of Review

Digital Equipment PDP 11/45 1973

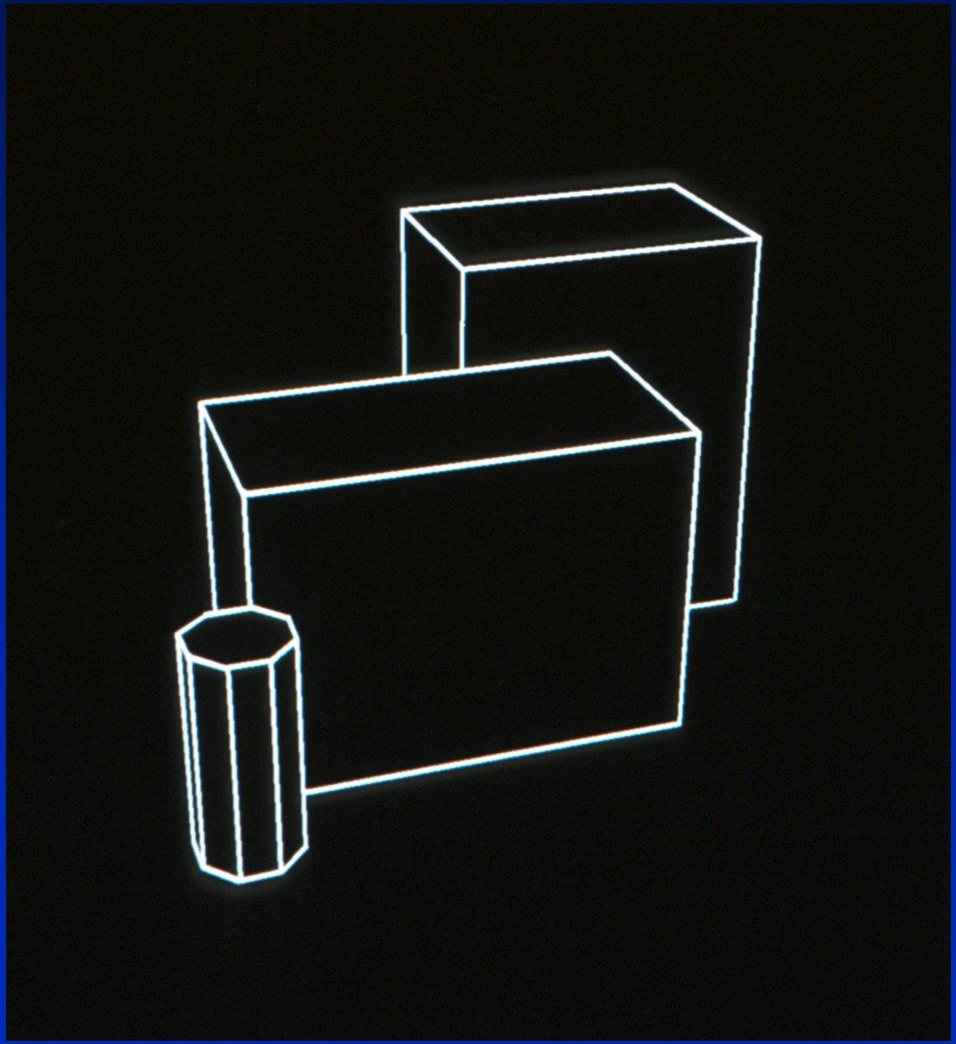
- Cost:


EVANS & SUTHERLAND COMPUTER CORPORATION
THE PICTURE SYSTEM

Hidden Line Algorithm



Hidden Line Algorithm

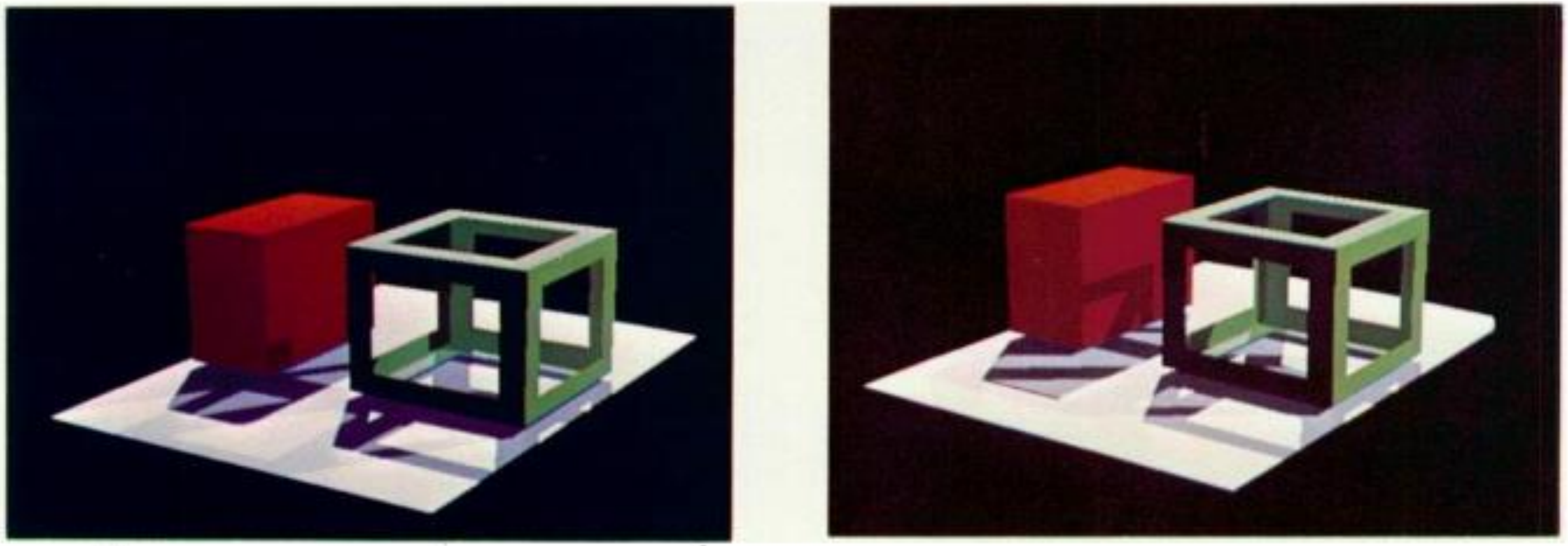



EVANS & SUTHERLAND COMPUTER CORPORATION
THE PICTURE SYSTEM

E & S Frame Buffer 1975

$80,000

Weiler Atherton Shadow Algorithm



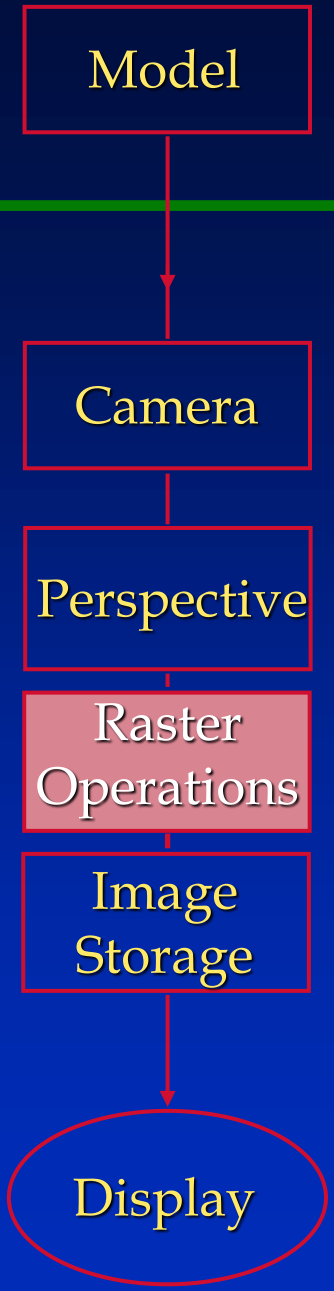
Figures 6. Shadowed Image Displays with Two Light Sources at Different Locations.

Assumptions and Transformations

- Lines into lines
- Planes into planes
- True perspective image
- Transformations preserves depth order

Raster Operations

- Conversion from polygons to pixels
- Hidden surface removal (z-buffer)
- Incremental shading



Visible Surface Algorithm
Z-Buffer Algorithm

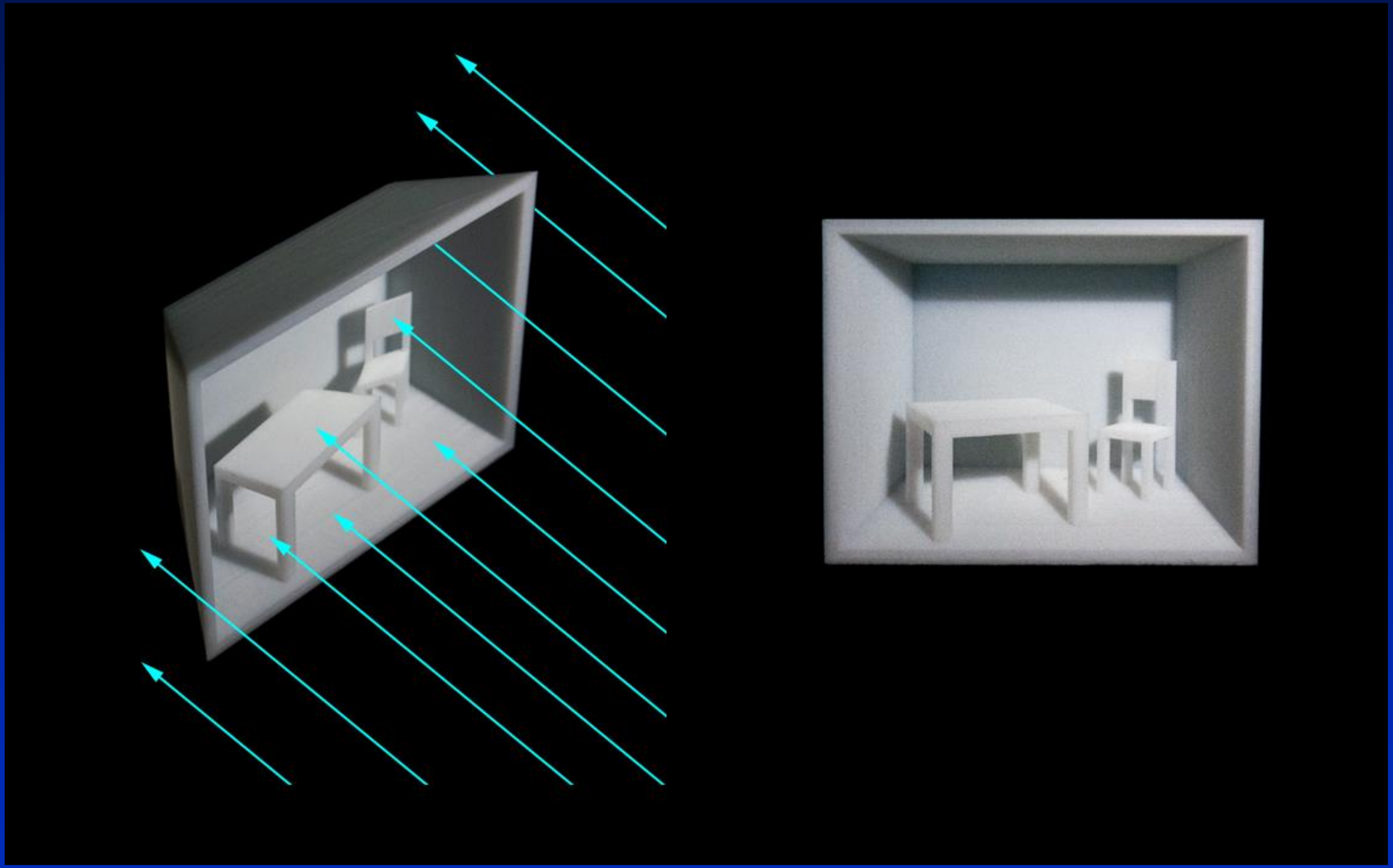
1. Set depth(x, y) = 1.0
   intensity(x,y) = background color

2. For each polygon, find all pixels covered

3. Calculate z(x,y) of each pixel covered by the polygon

Z-buffer

The distorted model in virtual image space is viewed from negative infinity and all rays are parallel to the view direction. (along the positive *Z*-axis)



The perspective image is identical.

Visible Surface Algorithm
Z-Buffer Algorithm

1. Set depth(x, y) = 1.0

    intensity(x,y) = background color

2. For each polygon, find all pixels covered

3. Calculate z(x,y) of each pixel covered by the polygon

4. If z(x,y) < depth(x,y), polygon is closer

    set depth(x,y) = z(x,y)

    change color

Depth Buffer Algorithm

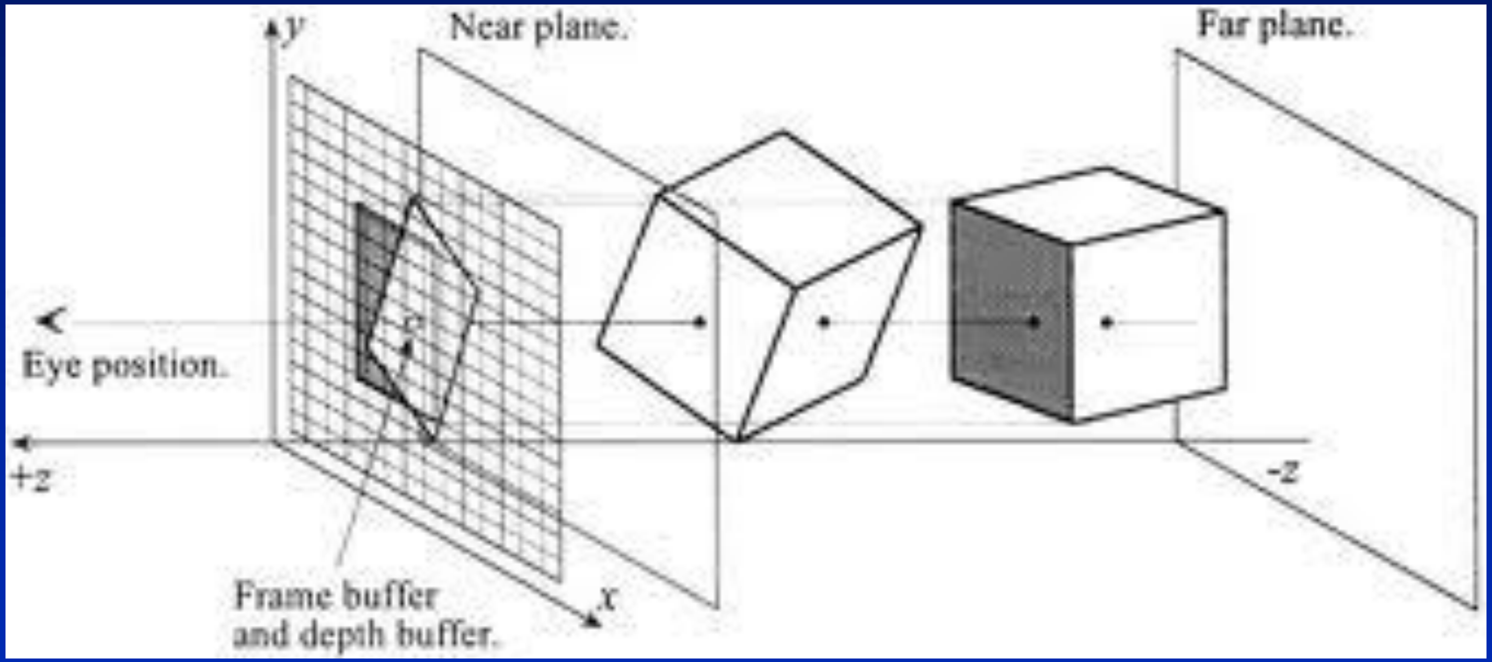


Image Storage

- Typical frame buffer

  1280 x 1024 pixels

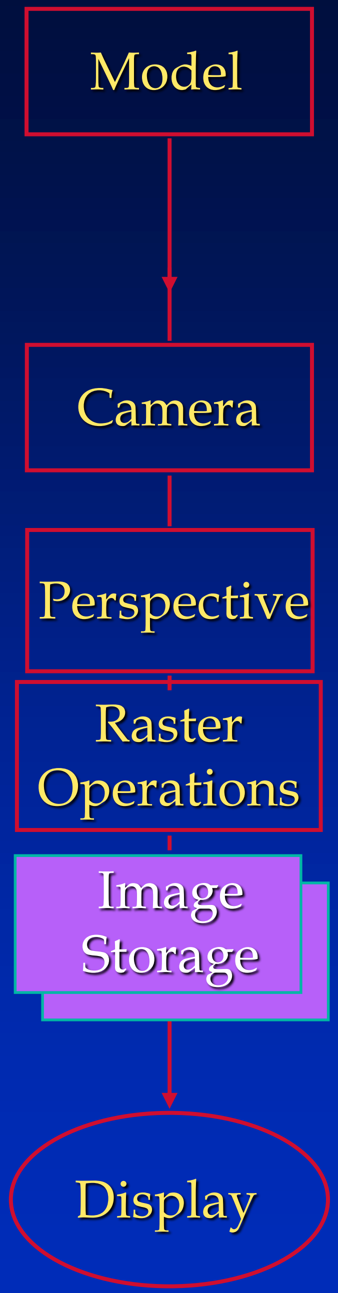
  3 channels (red, green, blue)

  1 byte/channel

- Total memory

  3 3/4 megabytes - single buffer

  7 1/2 megabytes - double buffer



Refresh vs. Update Rate

- The “refresh rate” is the number of times per second the entire image is drawn

- The “update rate” is the number of times per second the image is changed

Display

- Digital to analog conversion

  1920 x 1080 resolution

  60 frames per second

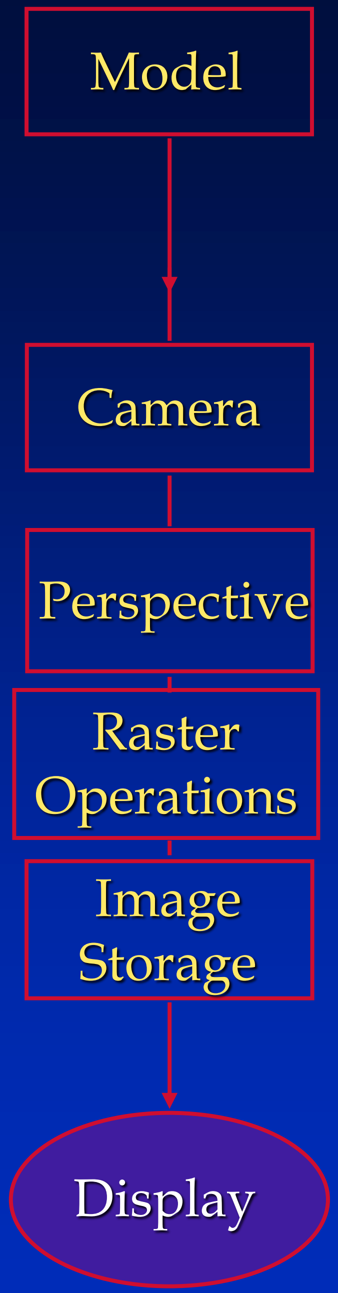
- Total data rate

  2 million pixels

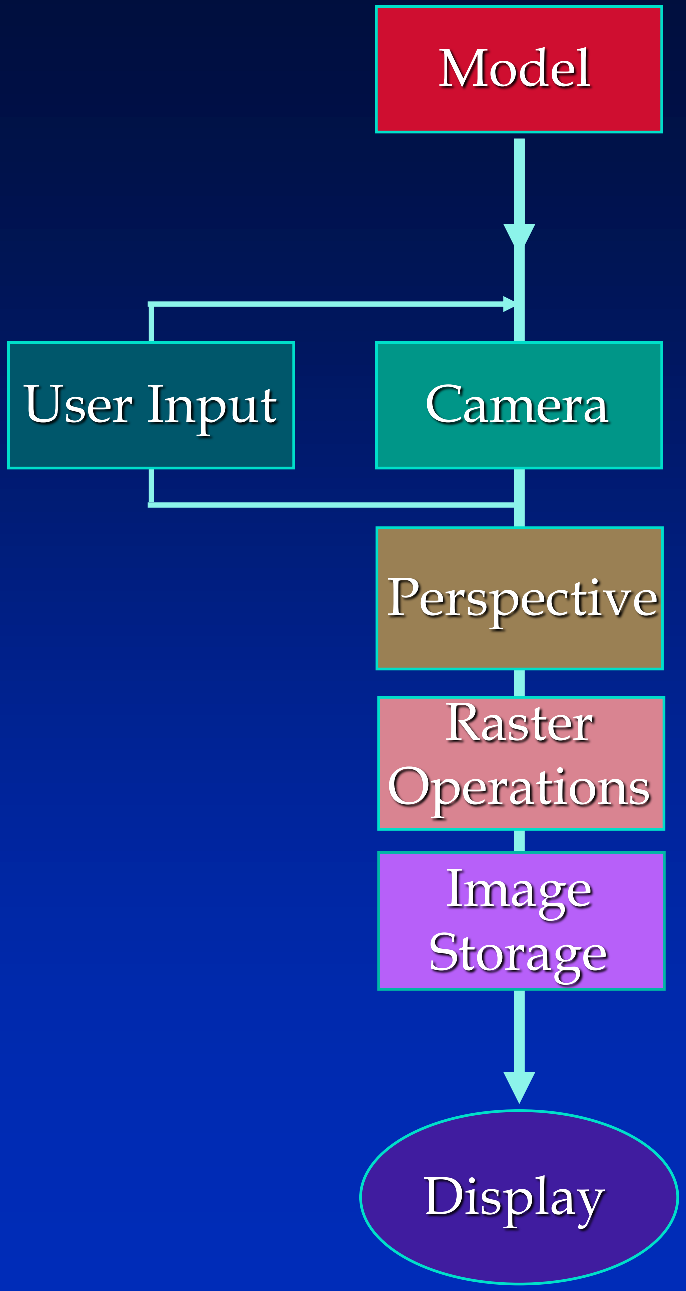
  x 3 bytes/pixel

  x 60 frames/second

  = 360 megabytes/second

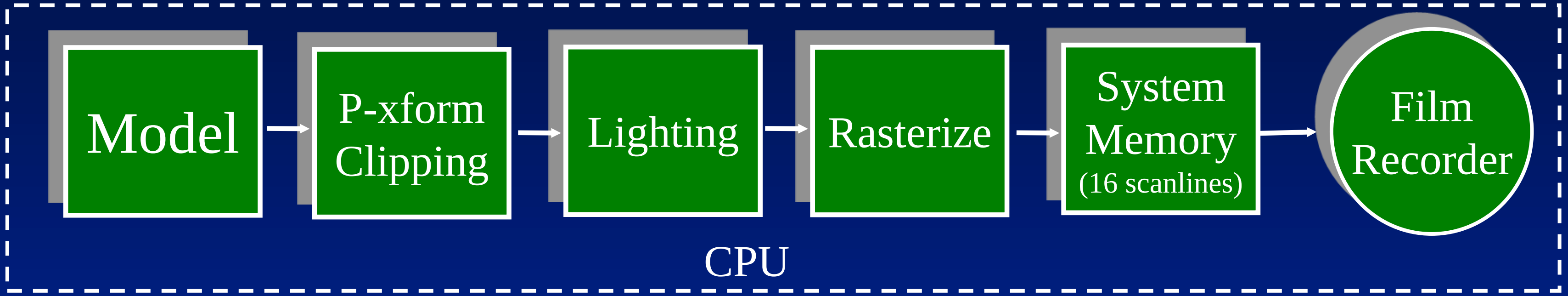


Direct Illumination



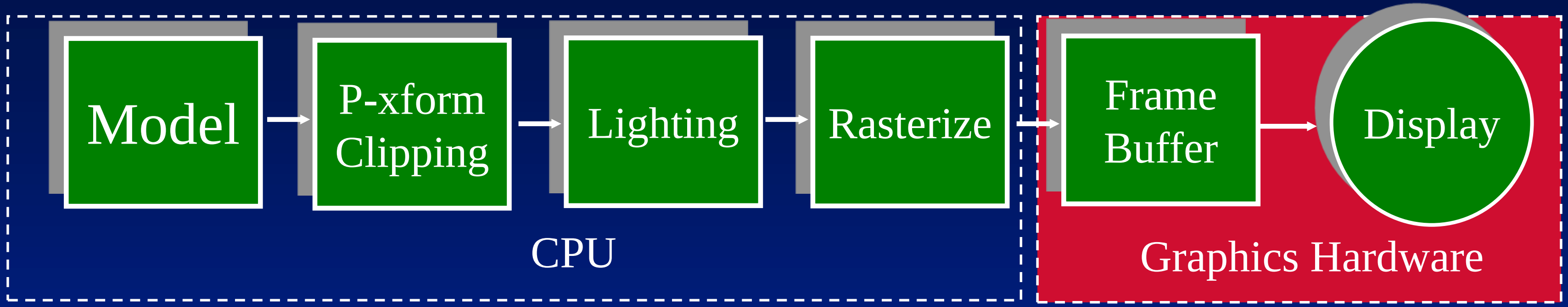
Graphics Hardware Pipeline

Graphics Hardware circa 1970



- System used to generate Phong goblet

Graphics Hardware circa 1975



Cost of Memory Prohibitive

- 512x480x8 bit frame buffer cost $80,000!
- No z-buffer (at 24 or 32 bits/pixel, it requires even more memory than FB)
- Only single frame buffer
- all work done in CPU until frame buffer(slow!)

Ford's Pipeline 1913

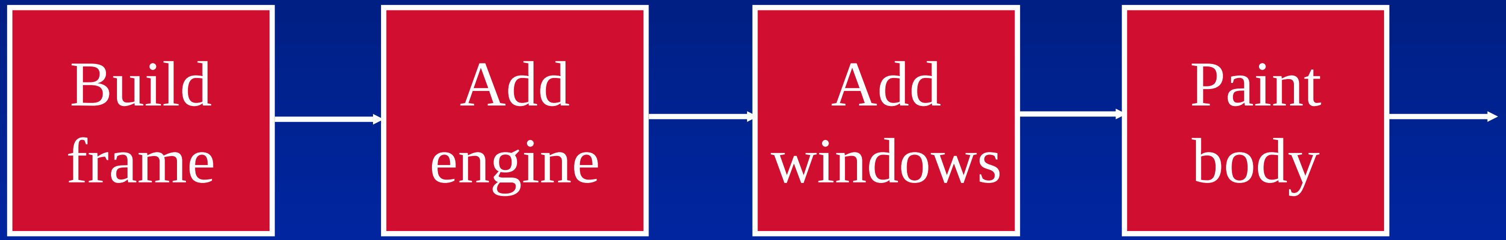
Why a Pipeline?

A pipeline allows multiple processes to occur in parallel. Example:

- An automobile assembly line. Assume 4 stations, each taking 2 minutes to do its task. It takes 8 minutes to make a car, but the *rate* at which cars are made is one every 2 minutes.

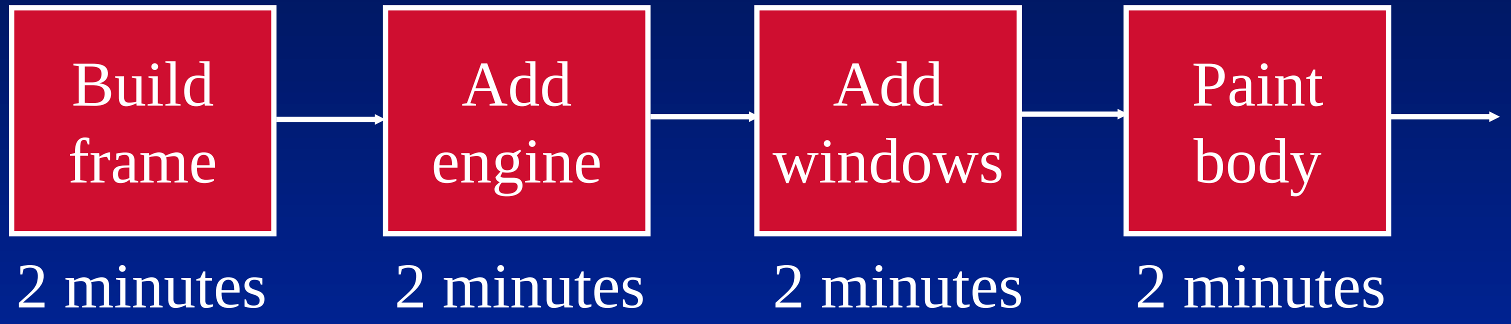
Example: Automobile Pipeline

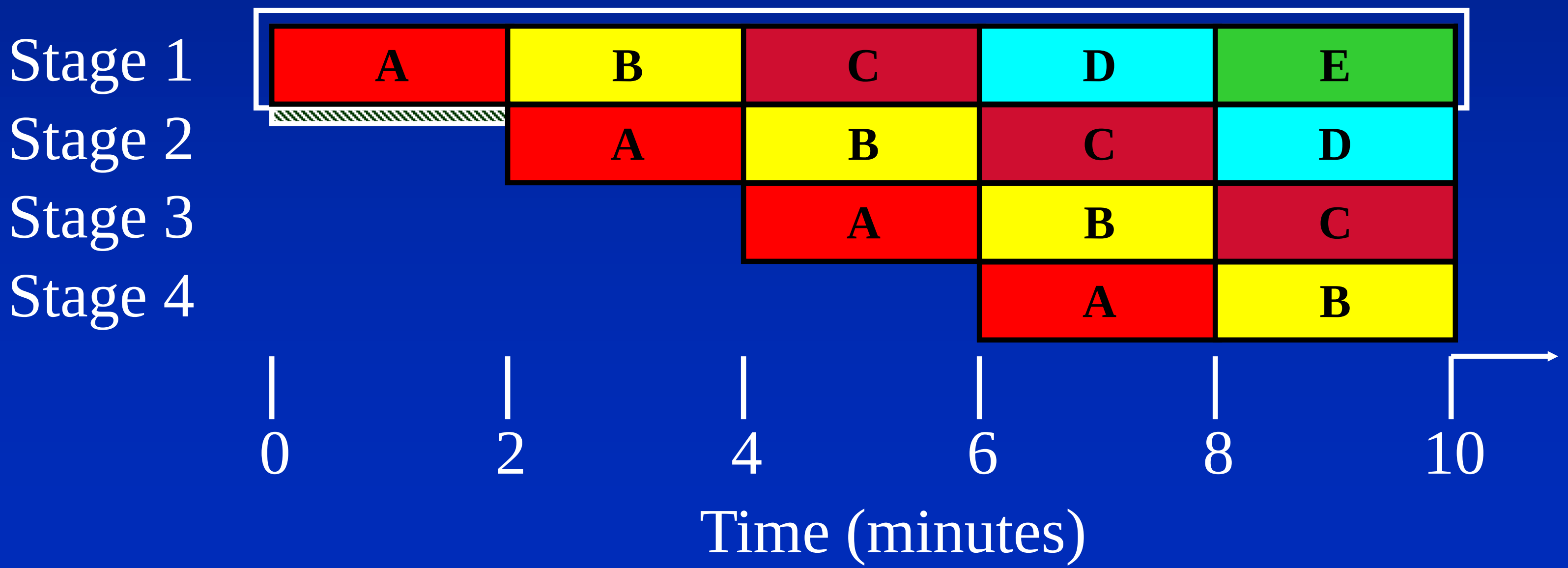
Automobile takes 8 minutes to make, but the assembly line makes a car every two minutes.



2 minutes 2 minutes 2 minutes 2 minutes

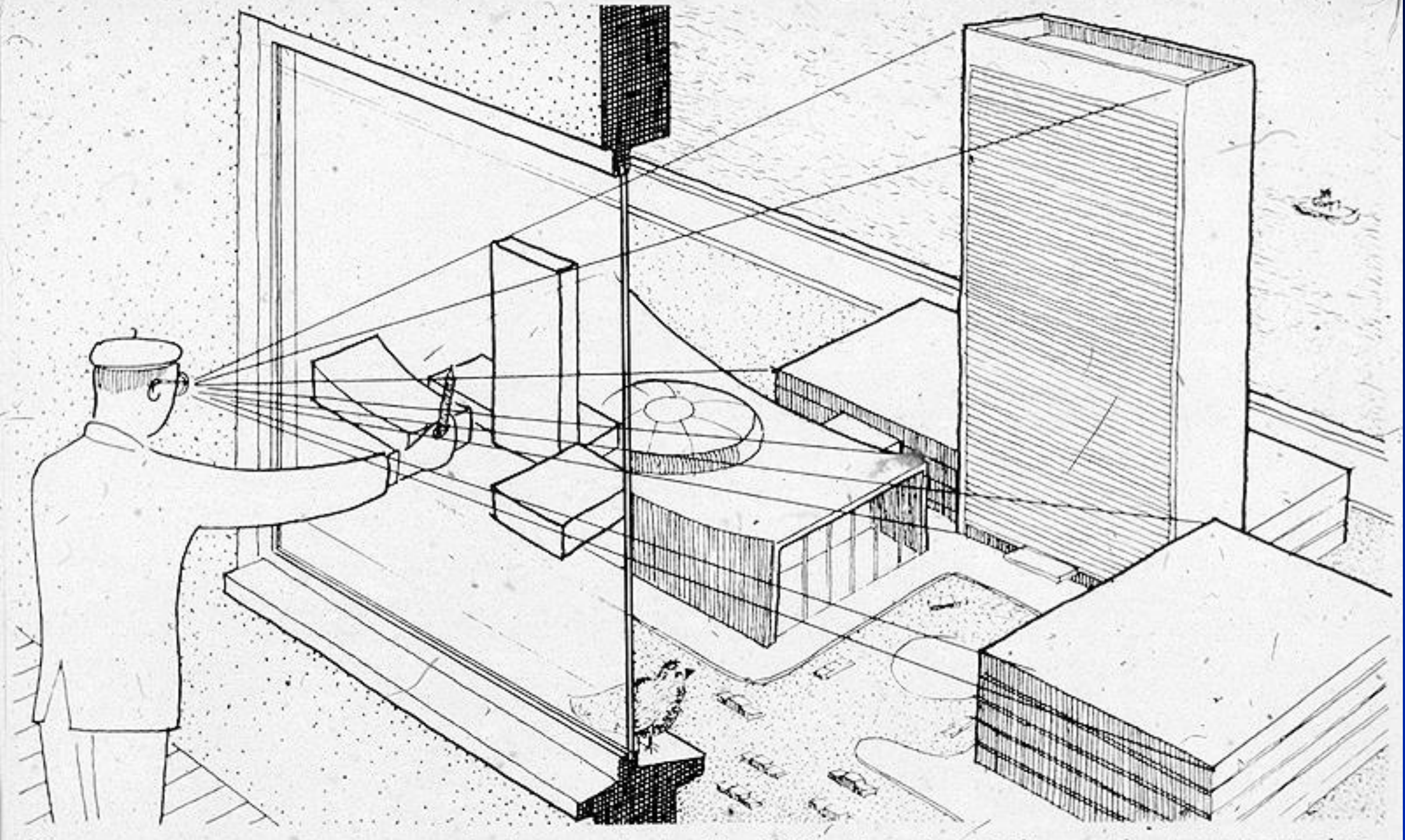
Automobile takes 8 minutes to make, but the assembly line makes a car every two minutes.





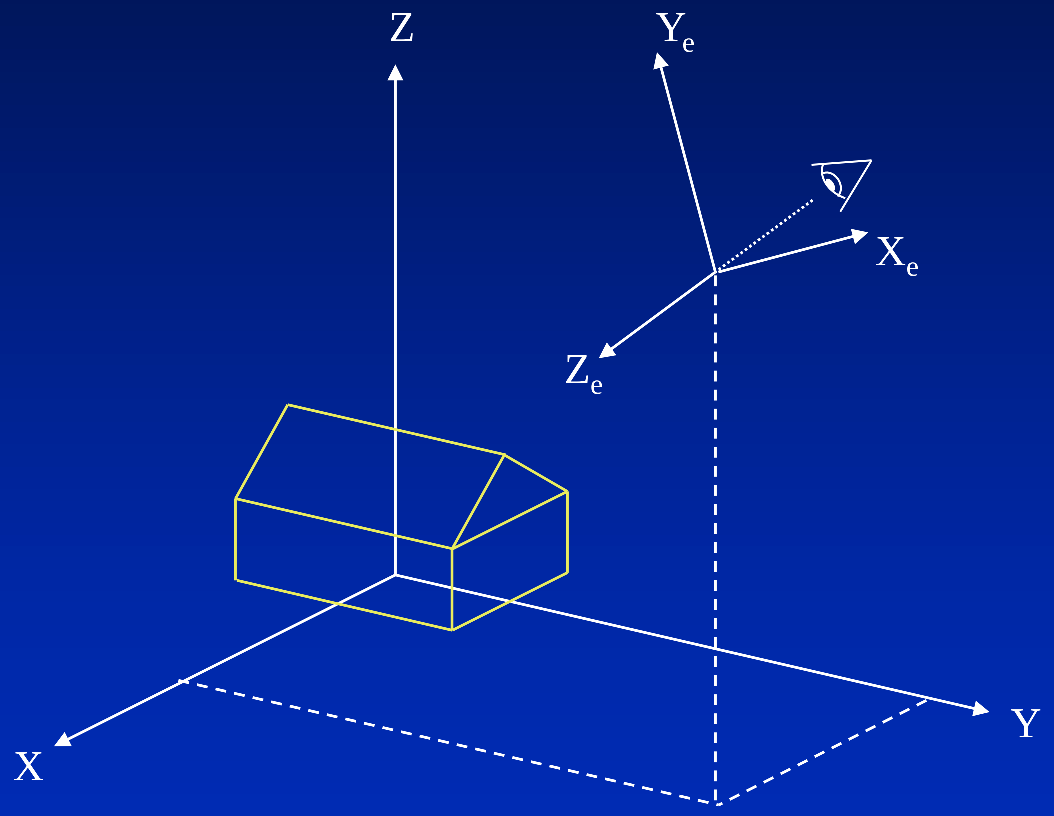
1984 - Clark's Geometry Engine

- Special Purpose four component vector floating point processor
- Same chip is software re-configurable to perform the three basic operations in computer graphics
  - –matrix transformations
  - –clipping
  - –mapping to output device coordinates



The concept of the picture plane may be better understood by looking through a window or other transparent plane from a fixed viewpoint. Your lines of sight, the multitude of straight lines leading from your eye to the subject, will all intersect this plane. Therefore, if you were to reach out with a grease pencil and draw the image of the subject on this plane you would be "tracing out" the infinite number of points of intersection of sight rays and plane. The result would be that you would have "transferred" a real three-dimensional object to a two-dimensional plane.

Eye Coordinate System

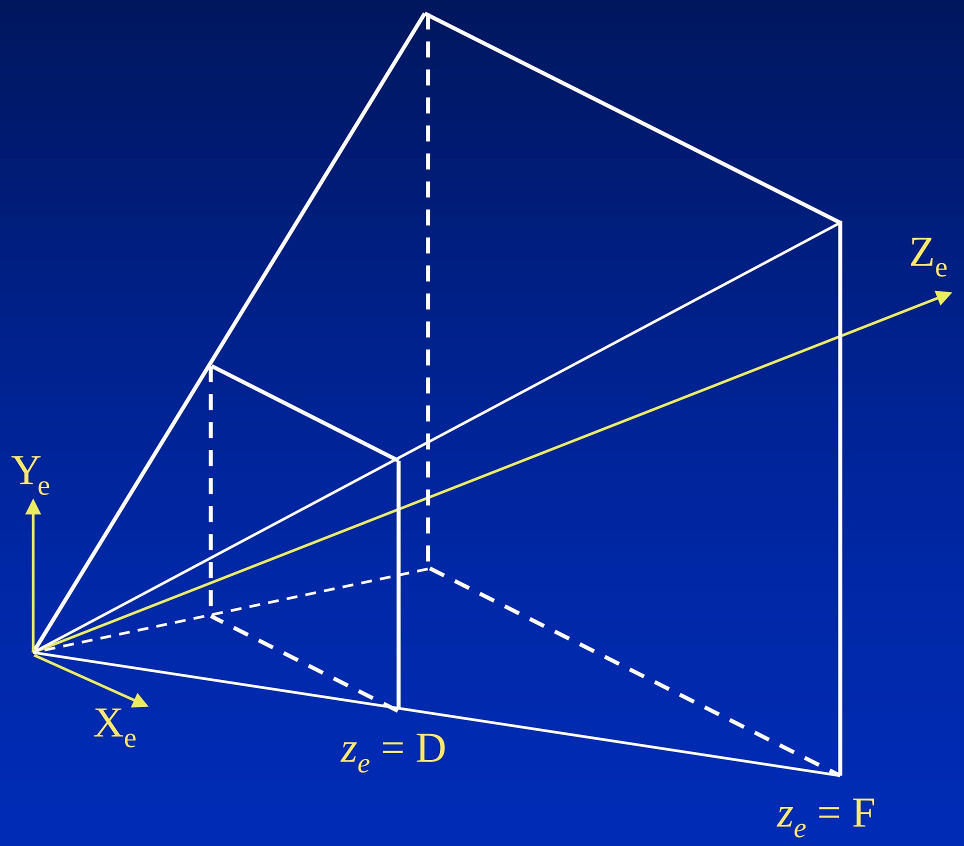


Perspective Transformation

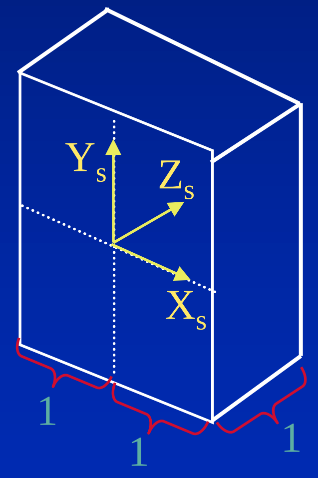
- Perspective transformation

  4 (4x4) Matrix multiplications

Clipping to a Viewing Frustum



Frustum of vision



Screen coordinate system

Clipping to a Viewing Frustum

- Perspective transformation

  4 (4x4) Matrix multiplications

- Clipping

  6 (4x4) Matrix multiplications

Viewport Mapping

- Perspective transformation

  4 (4x4) Matrix multiplications

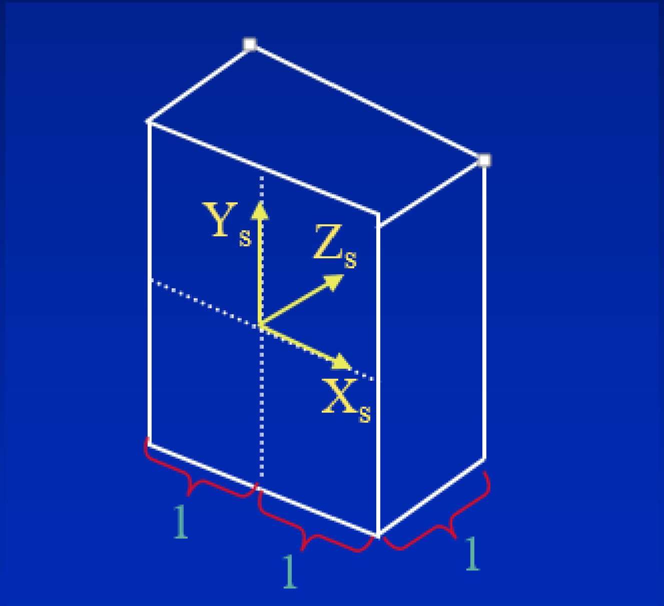
- Clipping

  6 (4x4) Matrix multiplications

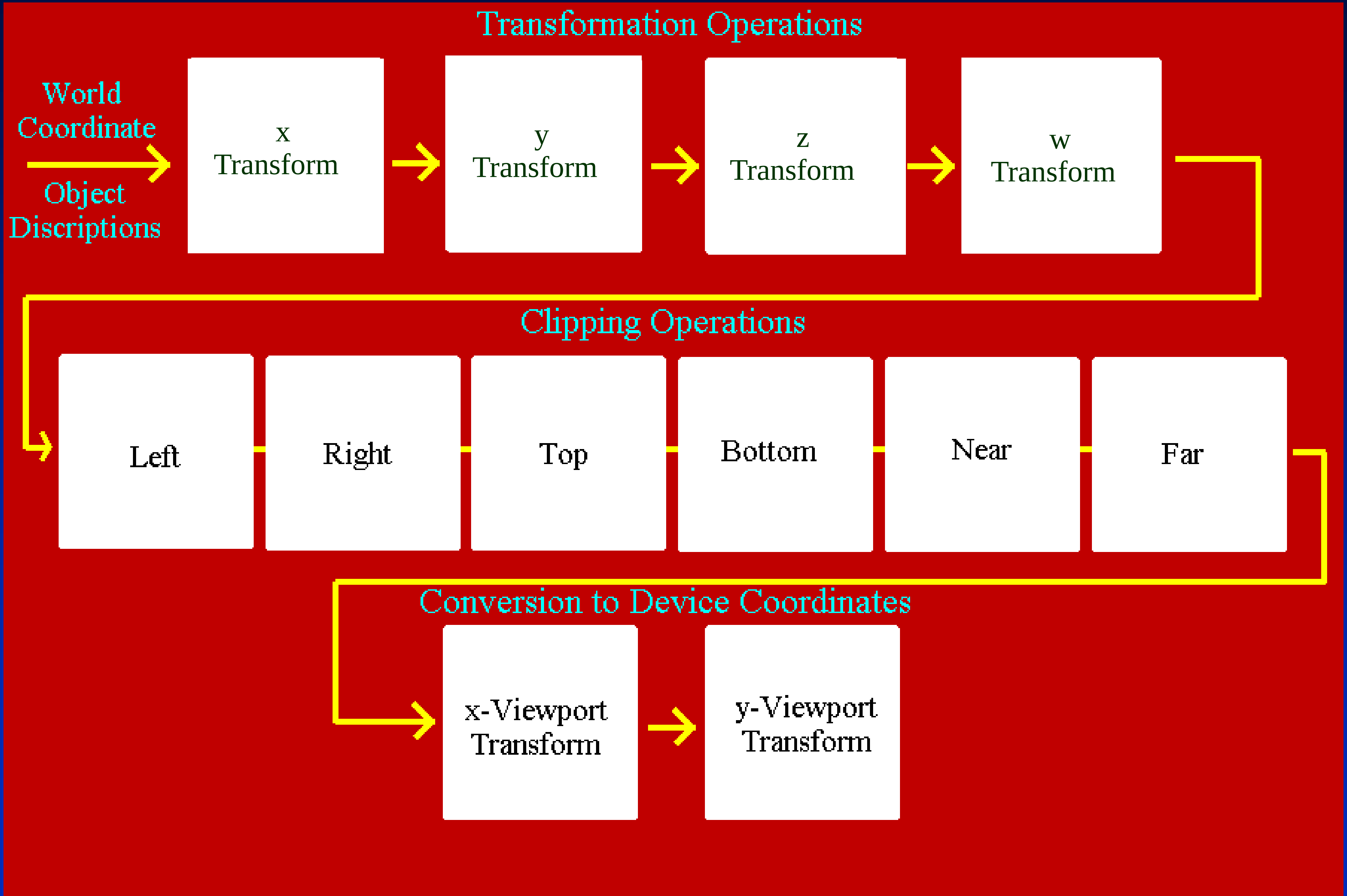
- Viewport mapping

  2 (4x4) Matrix multiplications

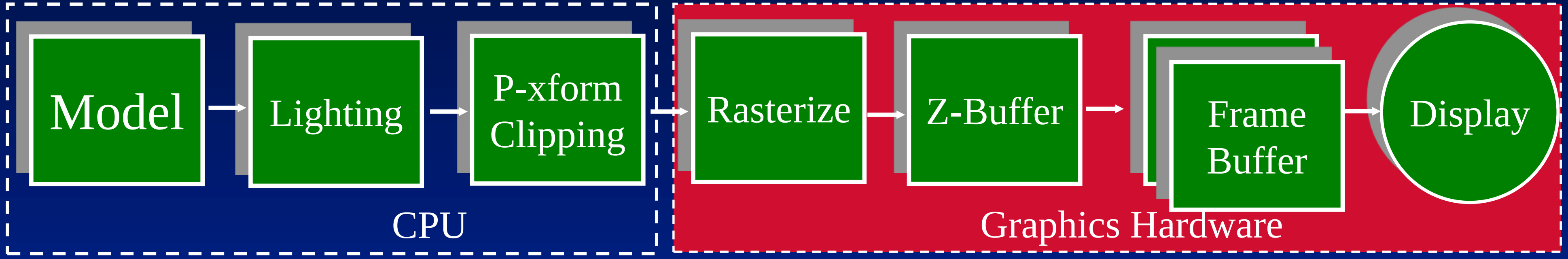
Viewport Mapping



1984 - Clark's Geometry Engine

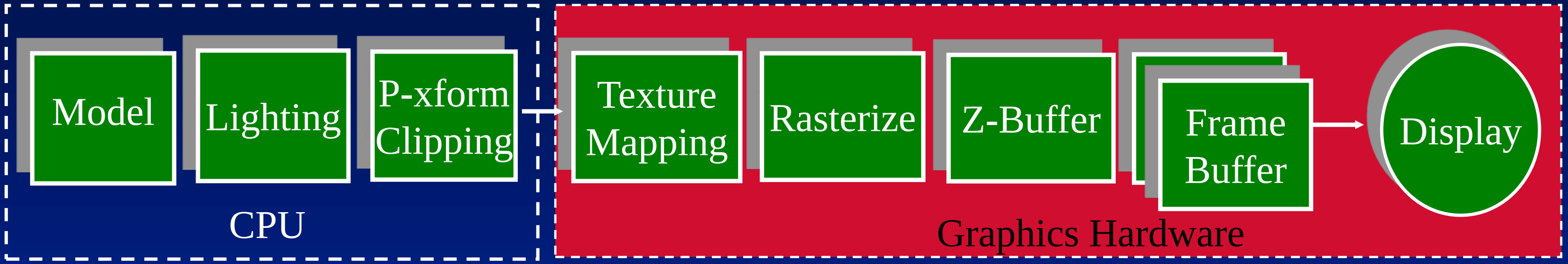


Graphics Hardware circa 1986



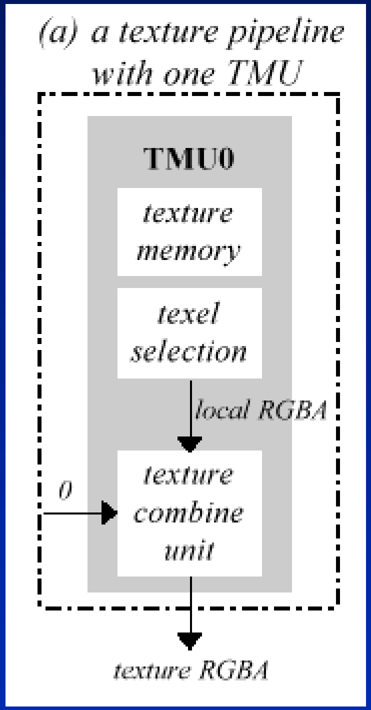
- Added Z-Buffer
- Added Double Frame Buffer
- Rasterization performed in hardware

Graphics Hardware 1999



- Addition of texture mapping units
- With texturing, high resolution detail is possible with relatively simple geometry

Texture Mapping Units - TMU's

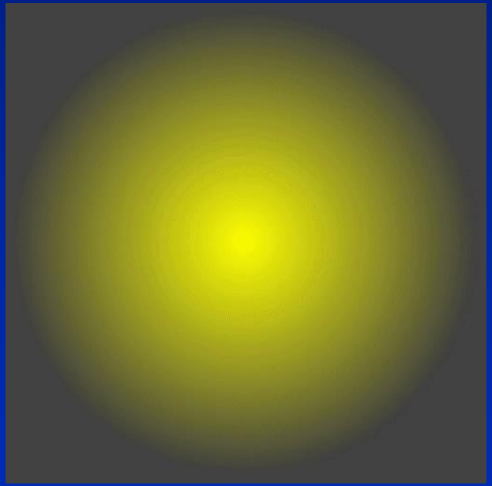


- A system with one TMU extracts the appropriate texel or texels from texture memory, minifies or magnifies it and filters it
- The texturecombine unit can scale the result

Multipass Example: Light Maps

- Two separate textures, one for the material's composition, one for the lighting

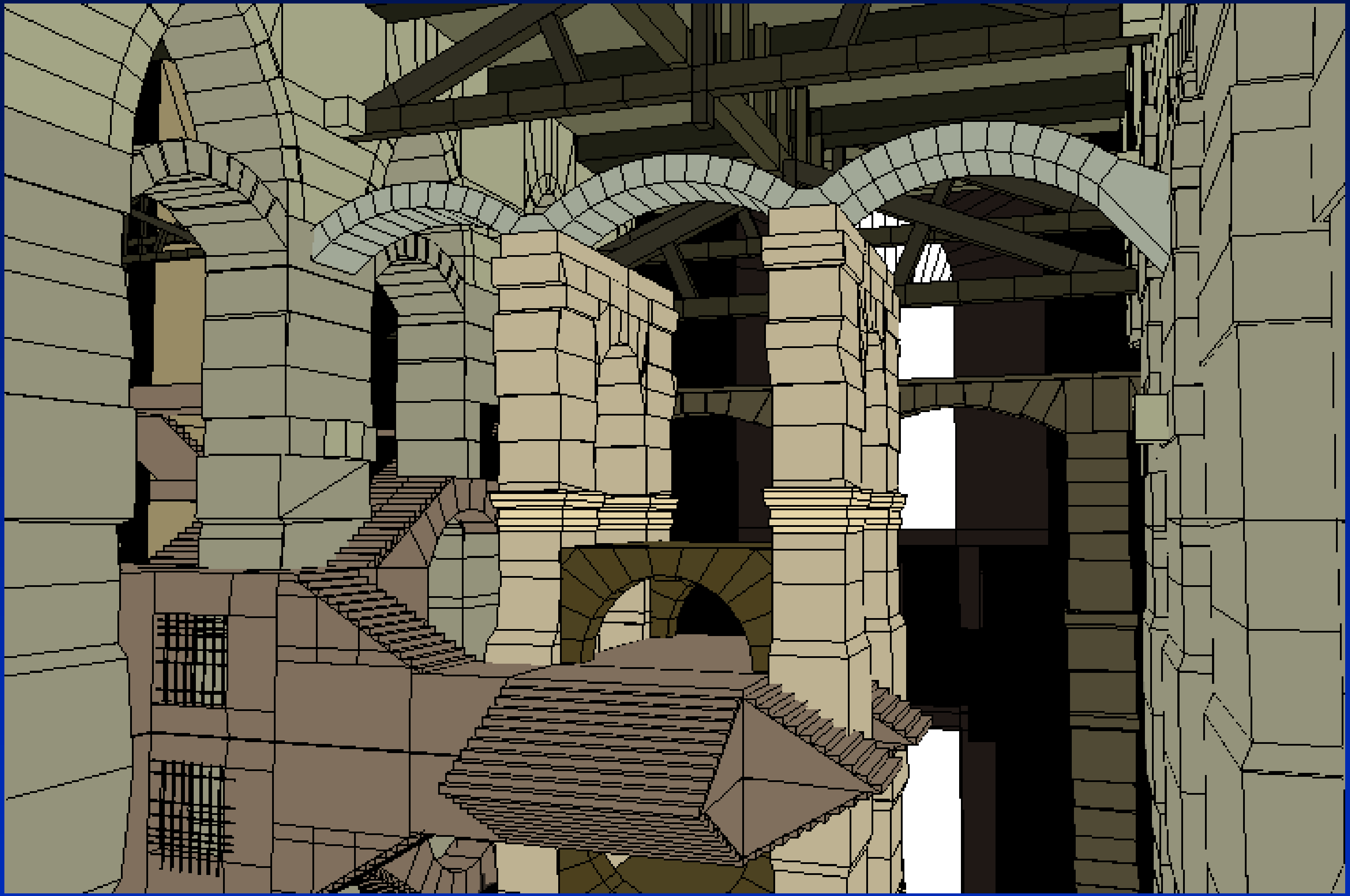
X



=

J.L.Mitchell, M. Tatro, and I. Bullard

Castle's Geometry

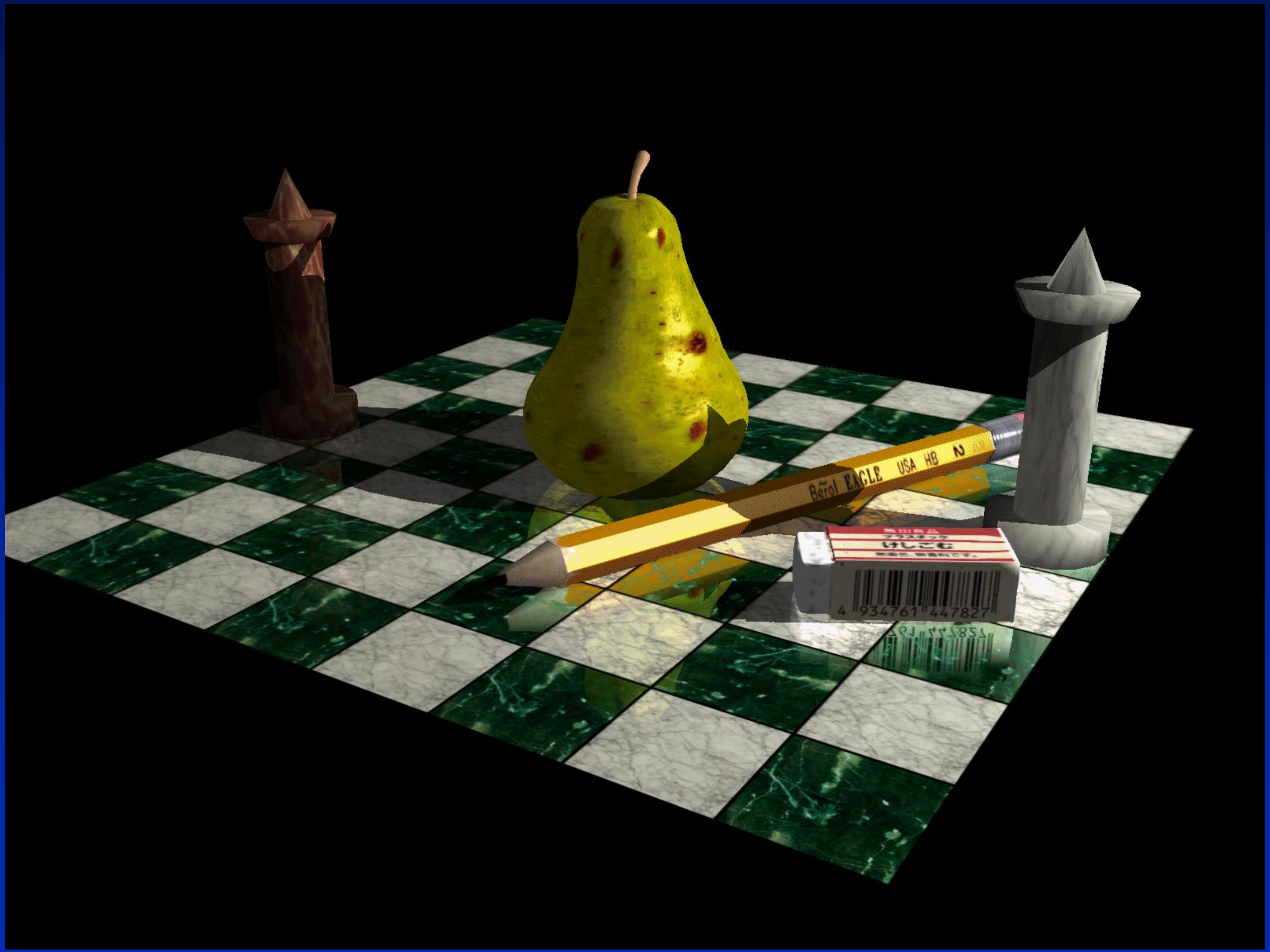


Agata & Andrzej Wojaczek, Advanced Graphics Applications Inc.

Reflection Example - Castle

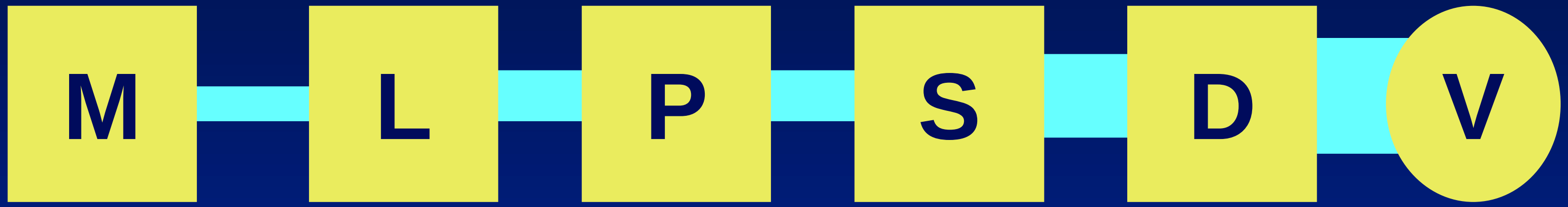
Agata & Andrzej Wojaczek, Advanced Graphics Applications Inc.

Putting it all together



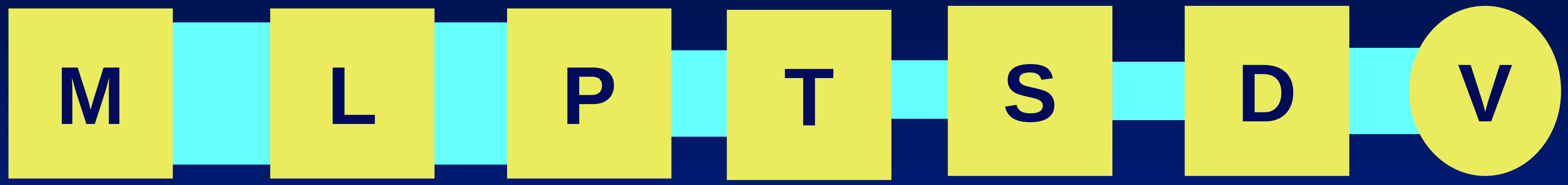
Gloss textures on pear, shadows on curved surfaces, reflections dropping off with depth from table.

Graphics Pipeline - 1980's



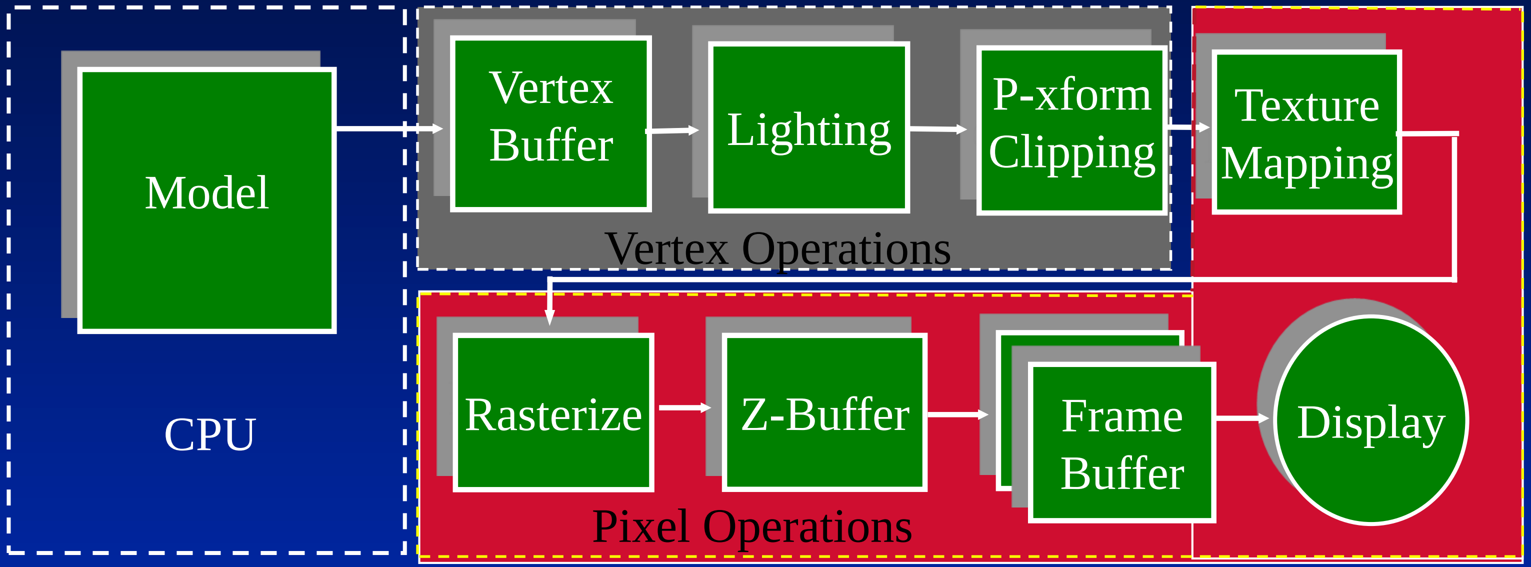
M — Model
L — Lighting
P — Perspective/Clipping
S — Scan Conversion/Z-buffer
D — Display Storage
V — Video

Graphics Pipeline - 2000 +



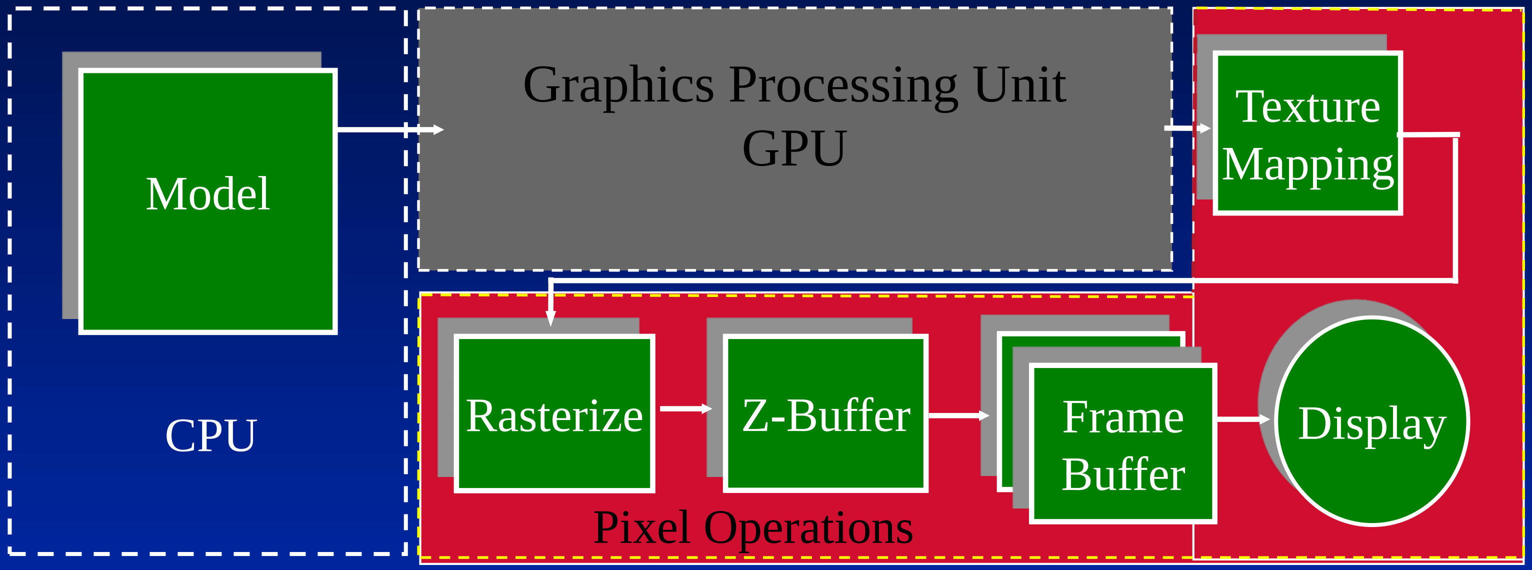
- M — Model
- L — Lighting
- P — Perspective/Clipping
- T — Texturing
- S — Scan Conversion/Z-buffer
- D — Display Storage
- V — Video

Graphics Hardware 2003

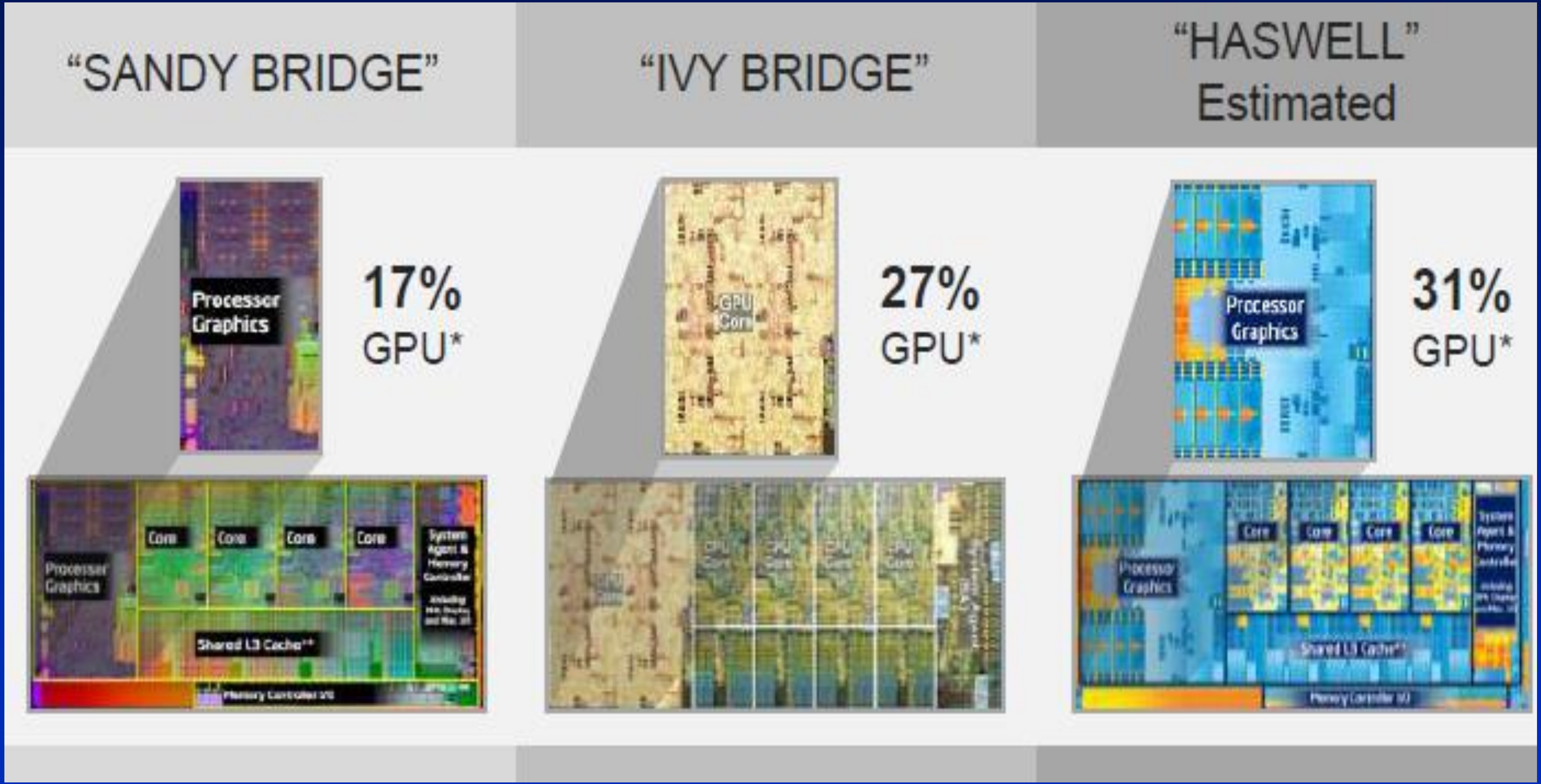


- Early GPU’s performed lighting and clipping operations on locally stored model

Graphics Hardware 2003+

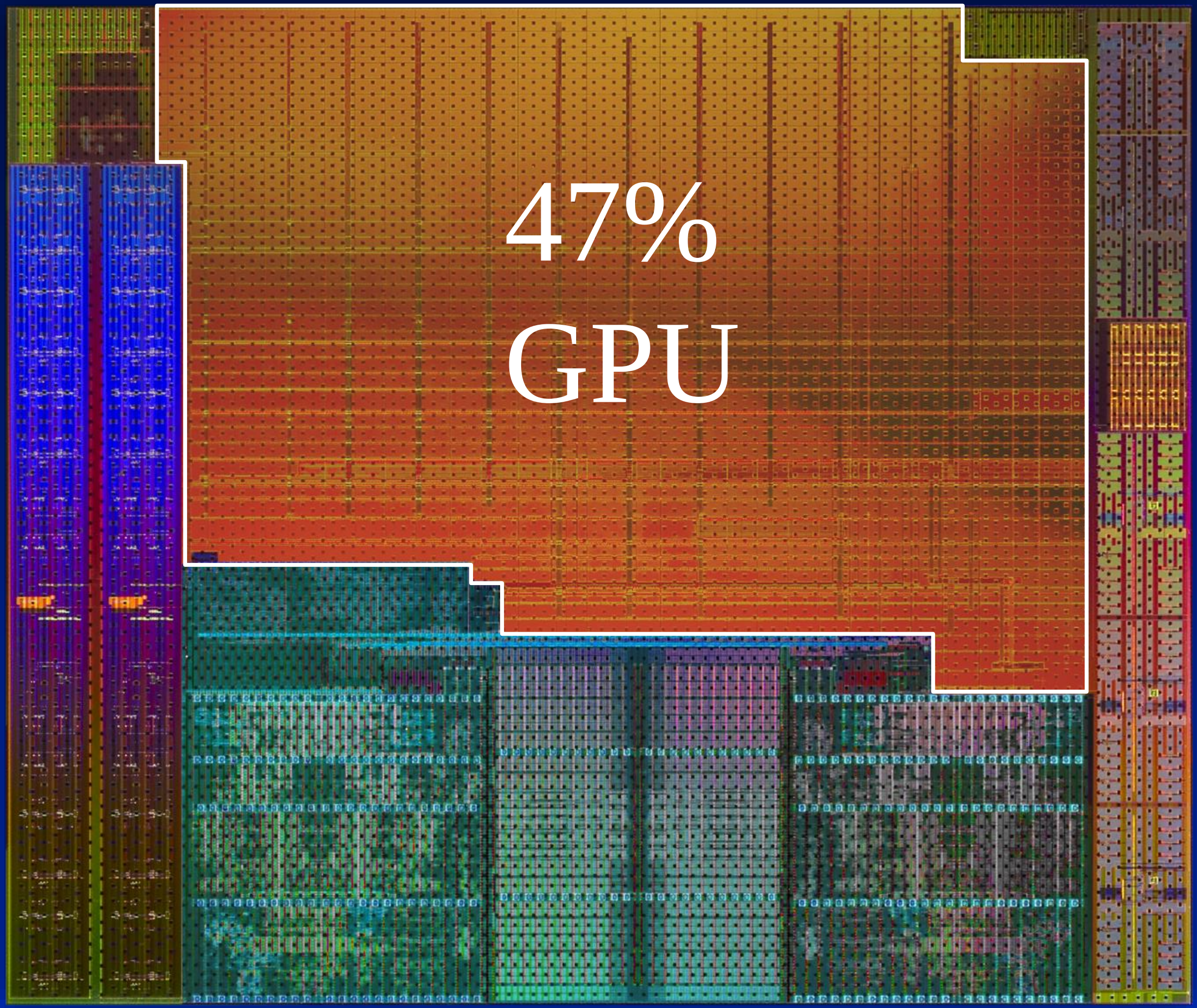


Intel – Integrated Graphics 2013

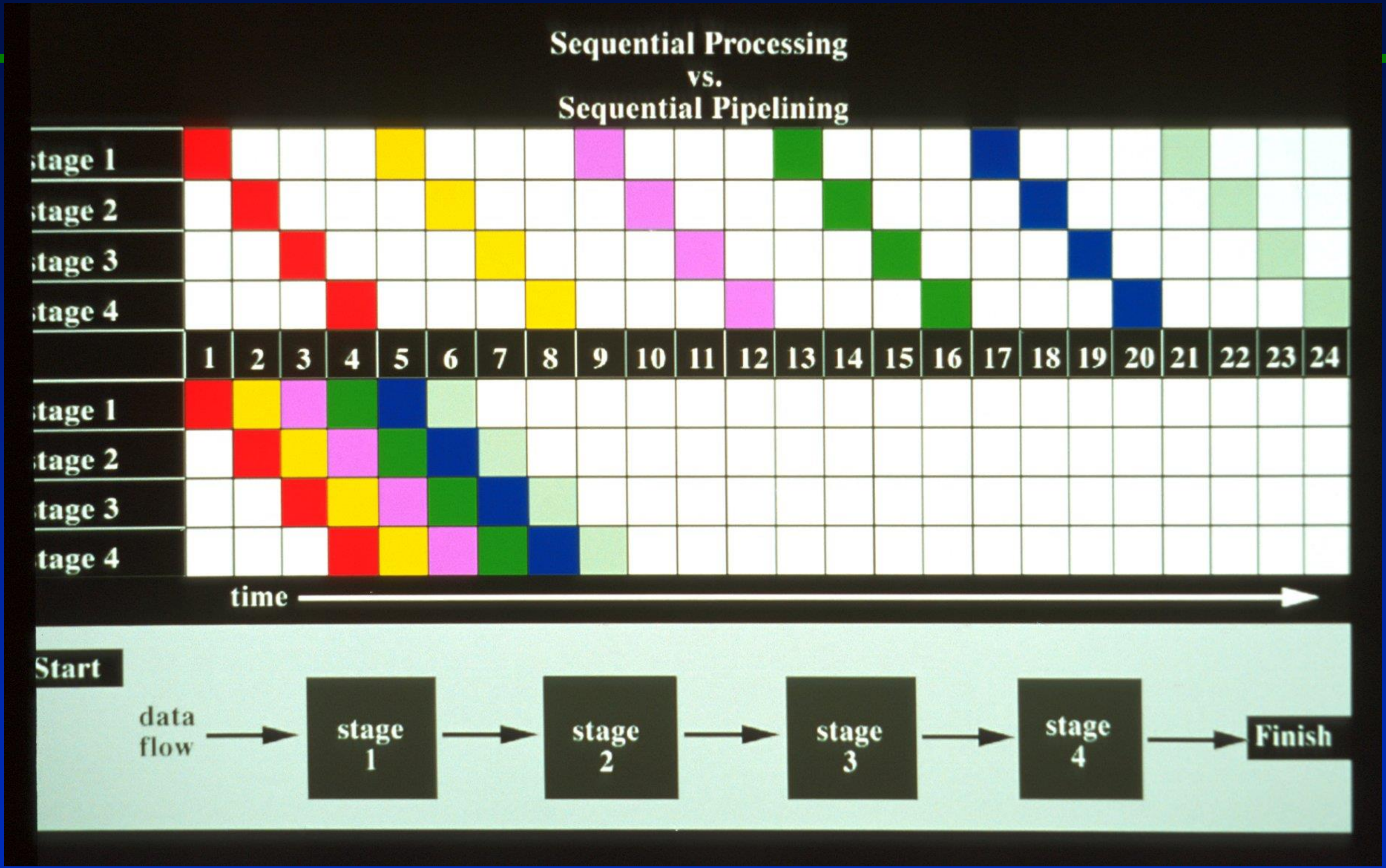


AMD – Integrated Graphics 2014

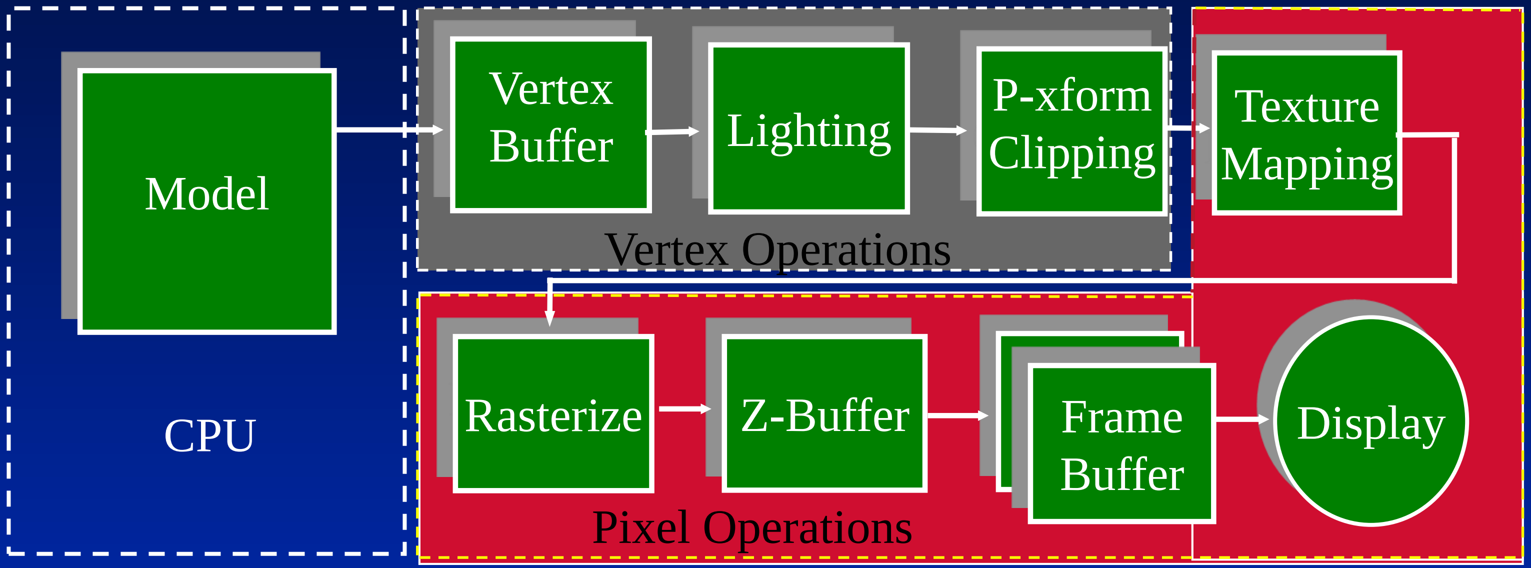
- “Kaveri”
- 28 nm
- 47% GPU



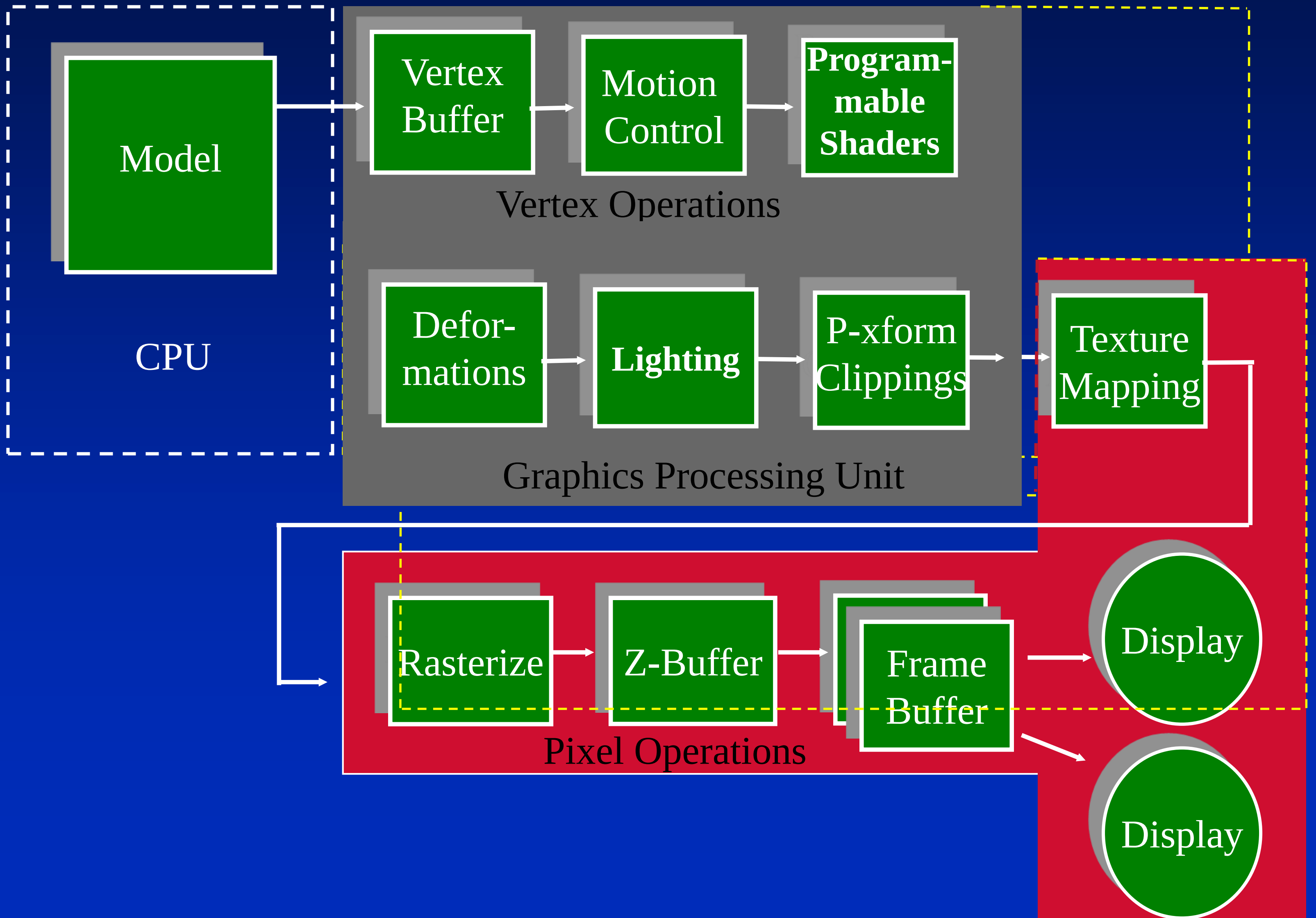
End


Sequential Processing
vs.
Sequential Pipelining
stage 1
stage 2
stage 3
stage 4
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
stage 1
stage 2
stage 3
stage 4
time
Start
data flow
stage 1
stage 2
stage 3
stage 4
Finish

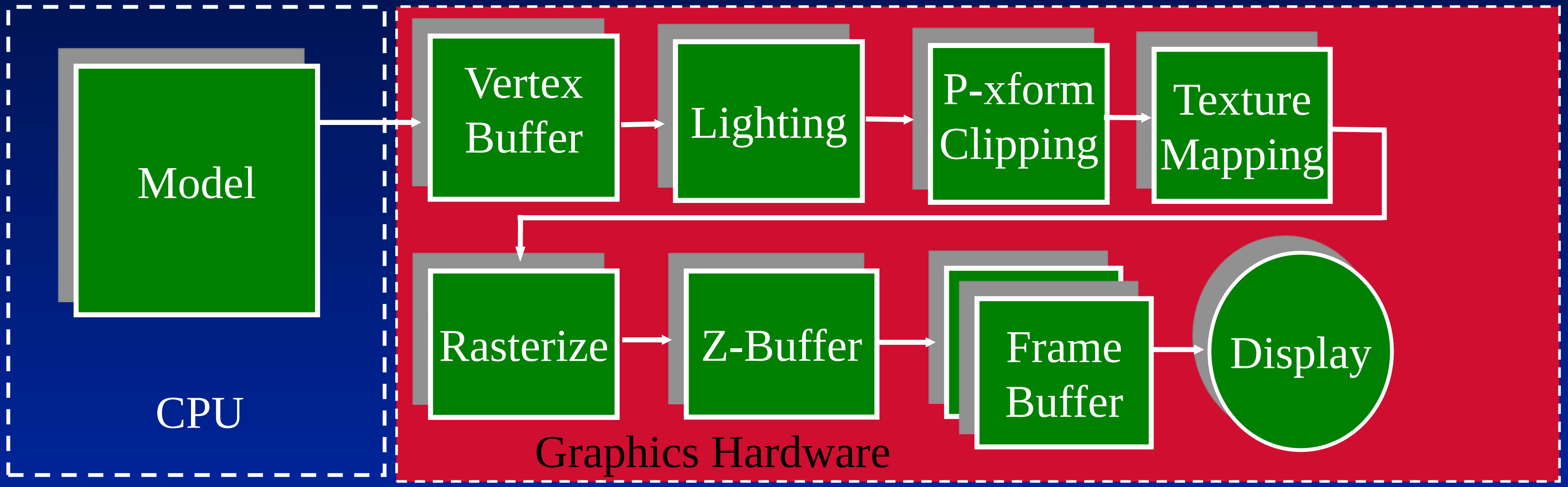
Graphics Hardware 2001



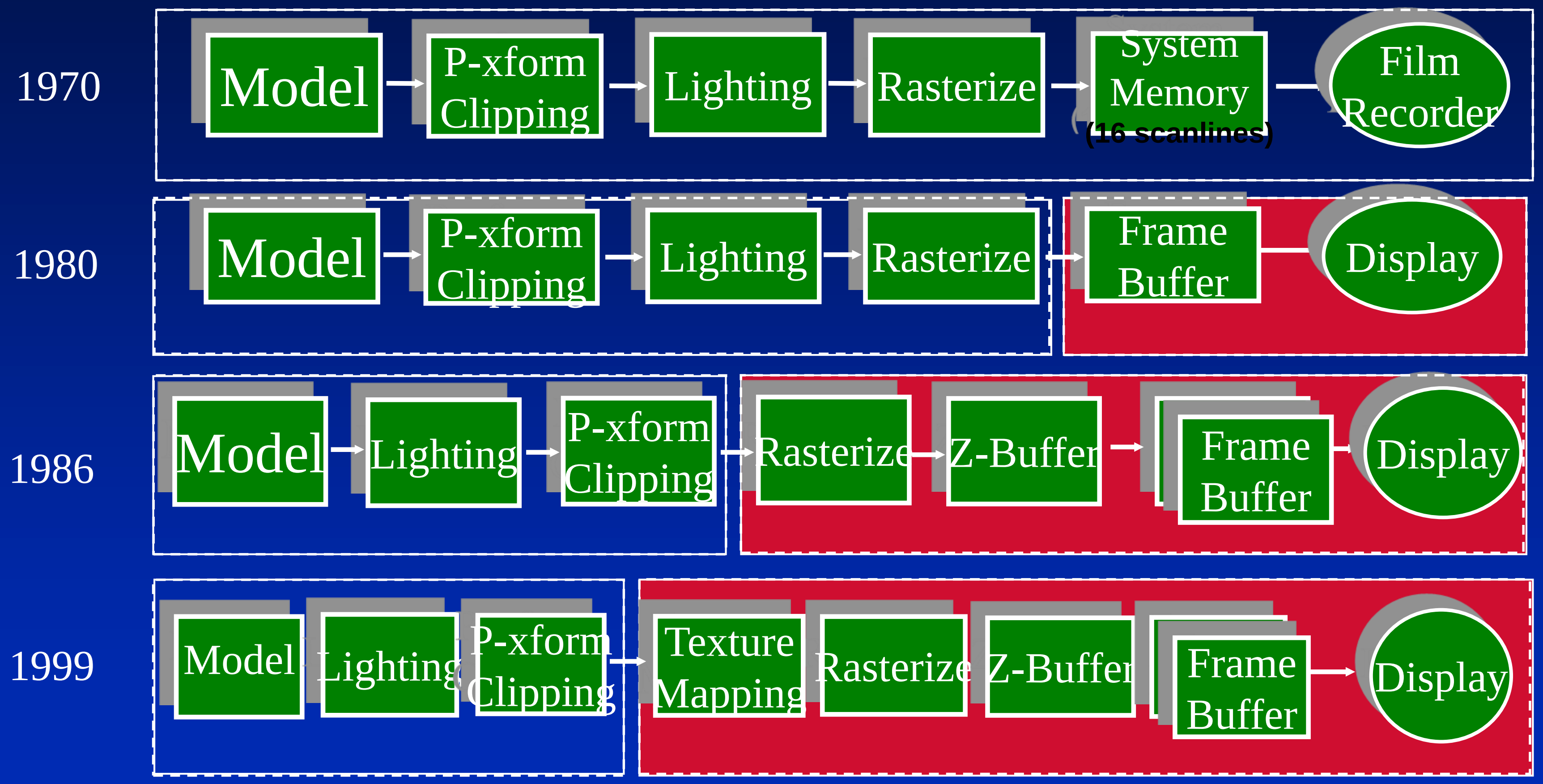
Graphics Hardware 2005



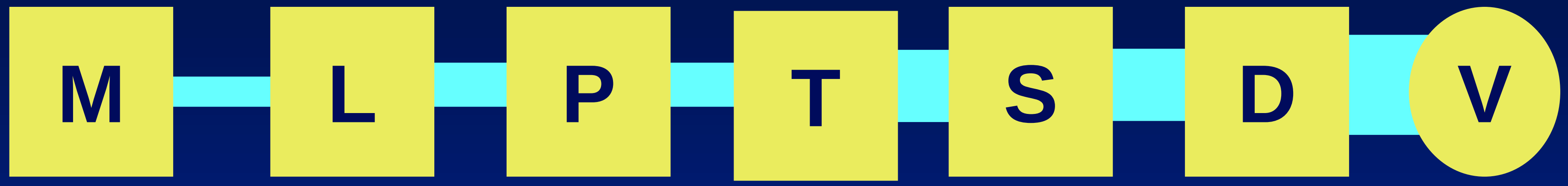
Graphics Hardware 2000



Graphics Hardware Recap

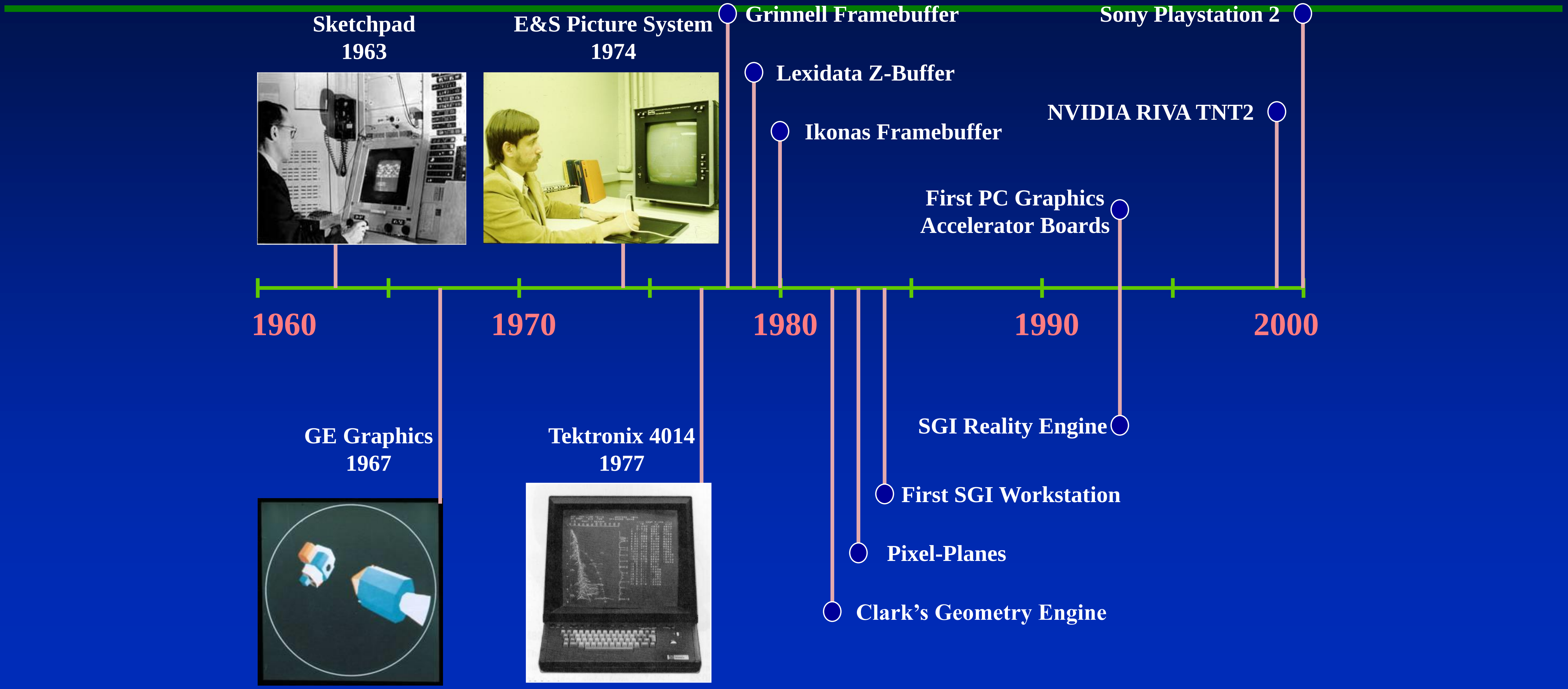


Graphics Pipeline - 1990's



M — Model
L — Lighting
P — Perspective/Clipping
T — Texturing
S — Scan Conversion/Z-buffer
D — Display Storage
V — Video

Computer Graphics Hardware Milestones



Computer Graphics Hardware Milestones

2000 2004 2006 2008 2010

Apple's iPod
October 2001

3G Network
October 2003
USA

2.5 inch disk
2004

USB Flash Drive
Approximately 2005

Apple's iPhone
January 9, 2007

4G mobile WiMAX
network in Baltic States
March 3, 2009

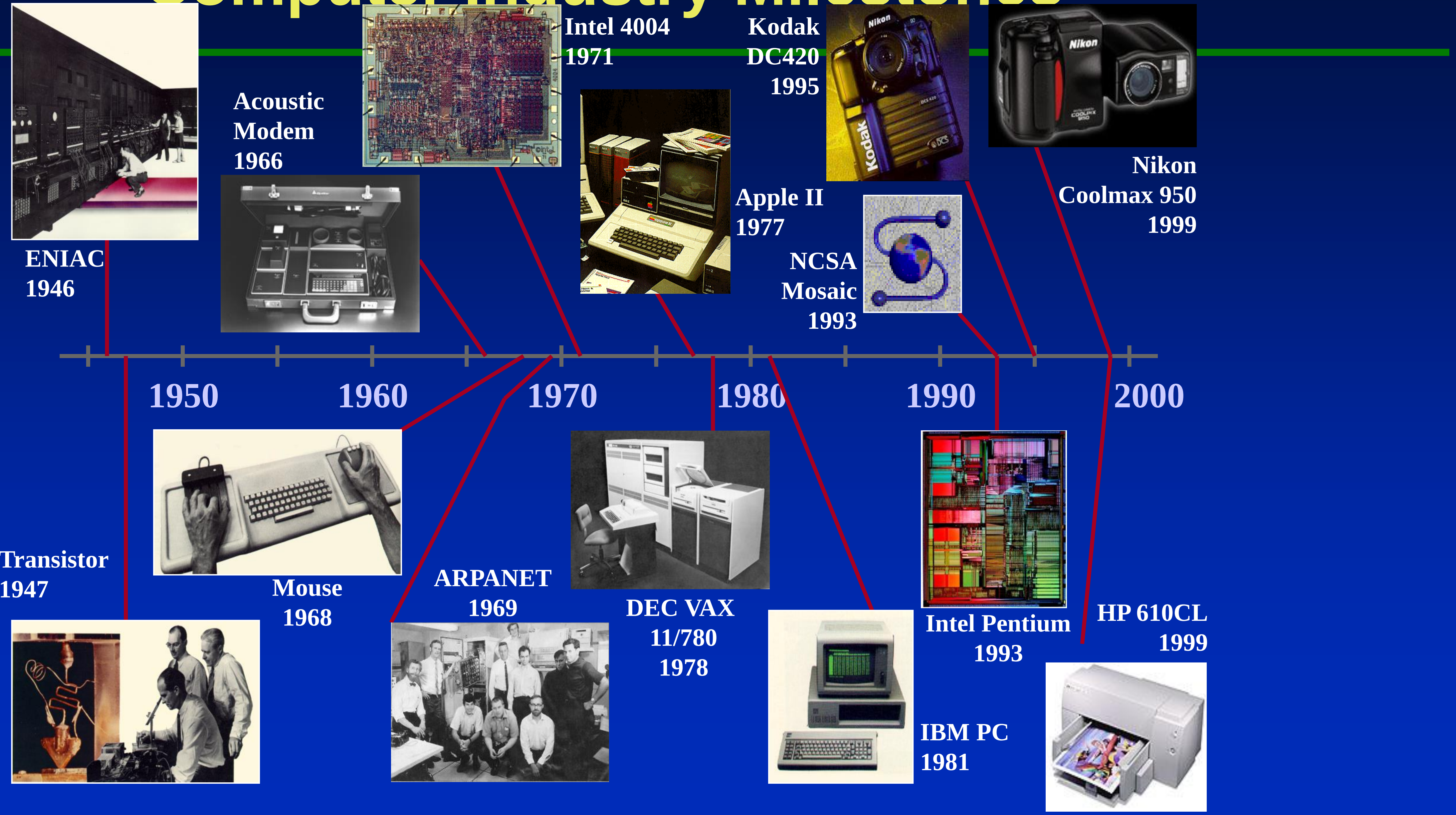
Intel started selling
32 nm Processor
January 7, 2010

> 40 megapixels cameras
Pentax 645D
March 10, 2010

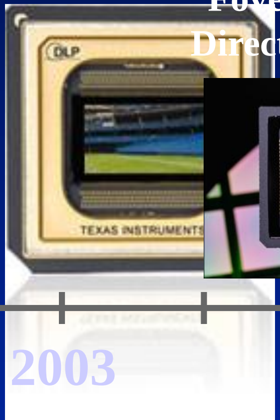
Apple's iPad
April 3, 2010

First
4G Smartphone
HTC Evo
June 5, 2010

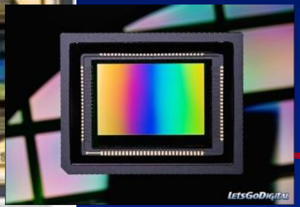
Computer Industry Milestones



Computer Industry Milestones

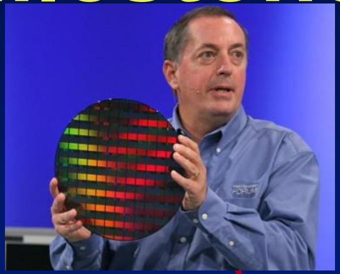


TI
DLP Chip



Foveon x3 Sensor
Direct image sensor

L.G. Philips
100 inch LCD Display

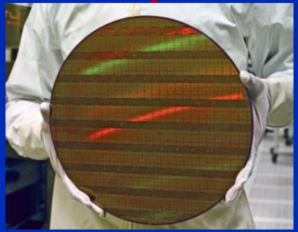


Intel 80
Core chip unveiled

iPhone (Apple)

2002 2003 2004 2005 2006 2007 2008

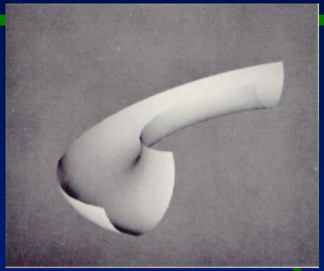
8th Generation
GeForce 8



SoC
45 nm process Technology

Microsoft
Surface

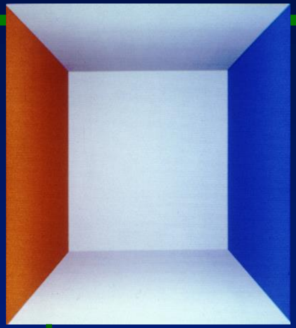
Computer Graphics Software Milestones



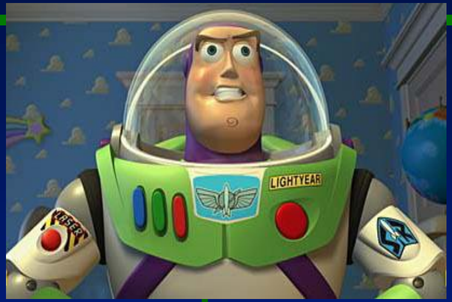
Gouraud Shading
1971

Phong Shading
1975

Cook-Torrance Model
1981



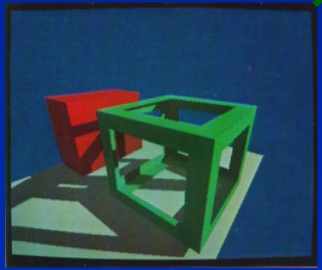
Radiosity
1984



Toy Story
1995

1960 1970 1980 1990 2000

Sketchpad
1963



First Color Images in SIGGRAPH
1977

Ray Tracing
1979

He Model
1991

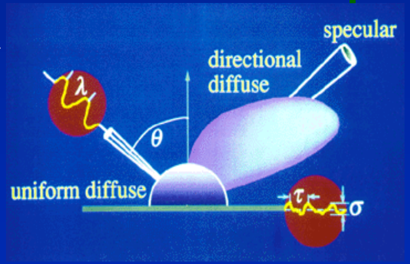
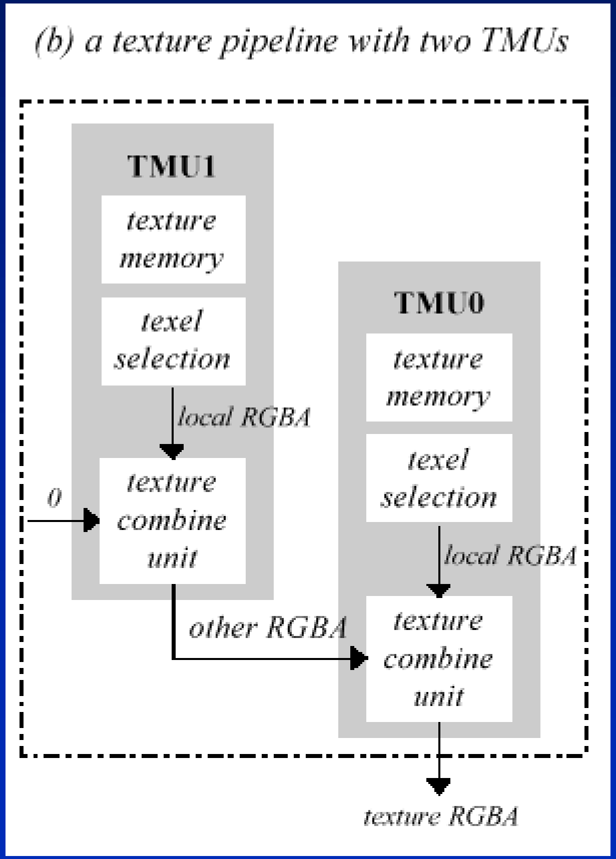


Image-Based Rendering
1993

Light Field
1996

Multitexturing



- Hardware architectures now support accessing more than one texture in a single pass

Camera

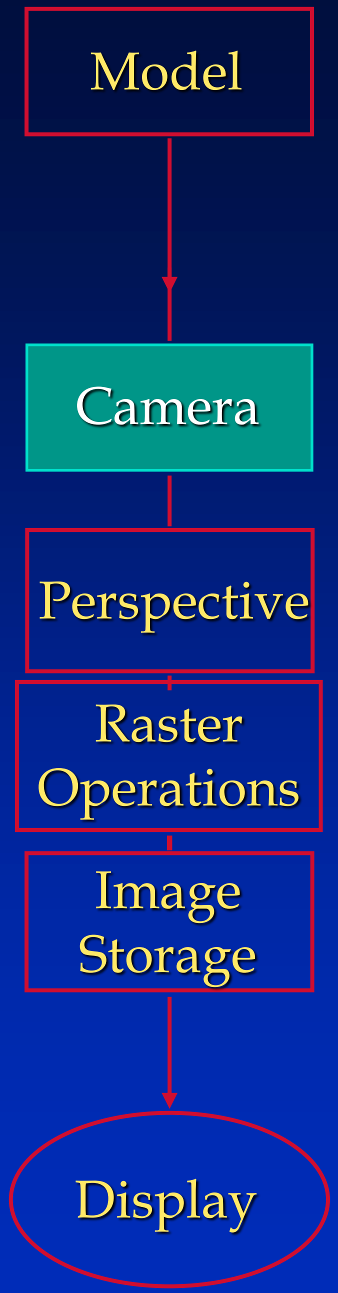
- Viewer Position
- Viewer direction
- Field of view

    Wide angle

    Telephoto
- Depth of focus

    Near

    Far



Model

- Environment

  Geometry & topology

  Material properties

  >Color, reflectance, textures

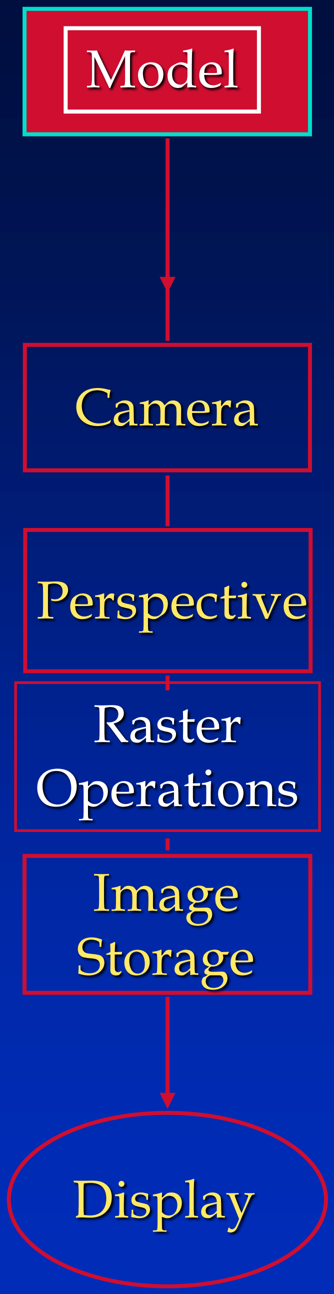
  >(Cost, strength, thermal properties)

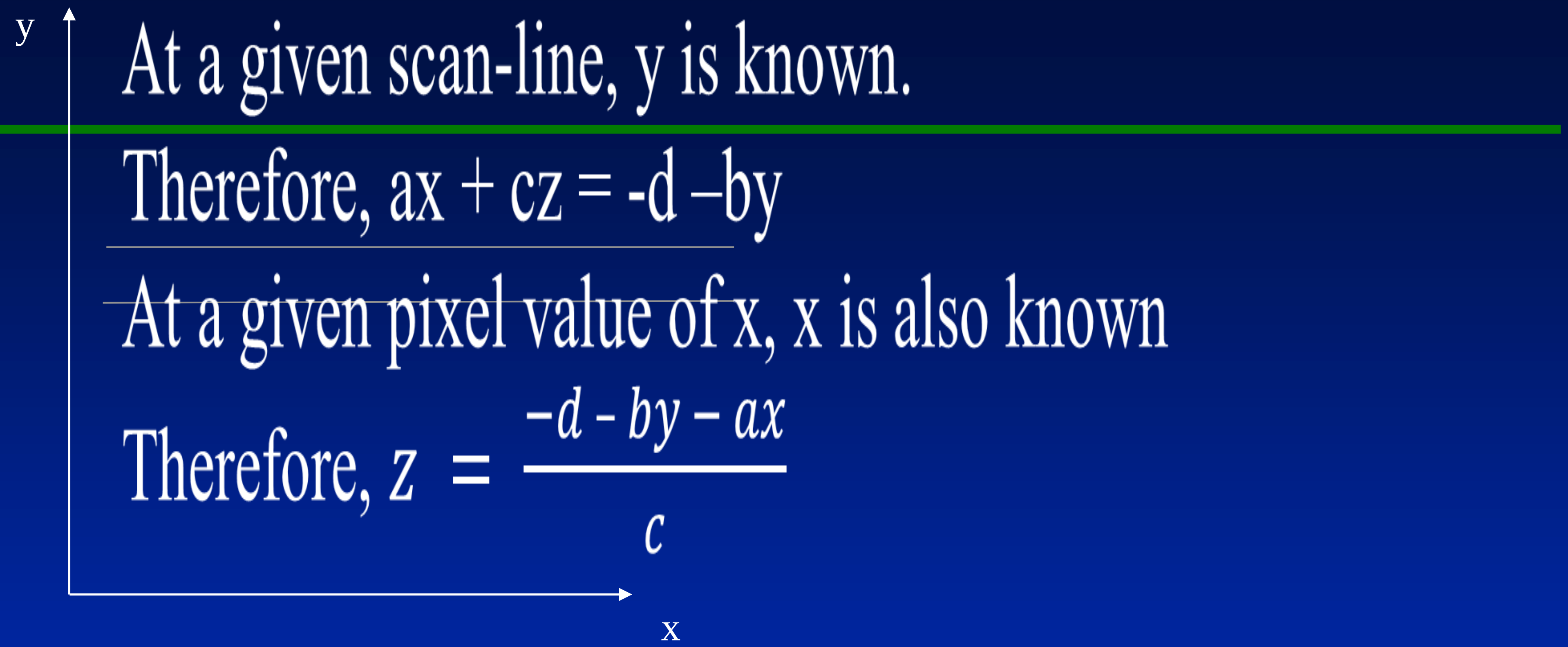
- Lighting

  Geometry & position

  Intensity, spectral distribution

  Direction, spatial distribution



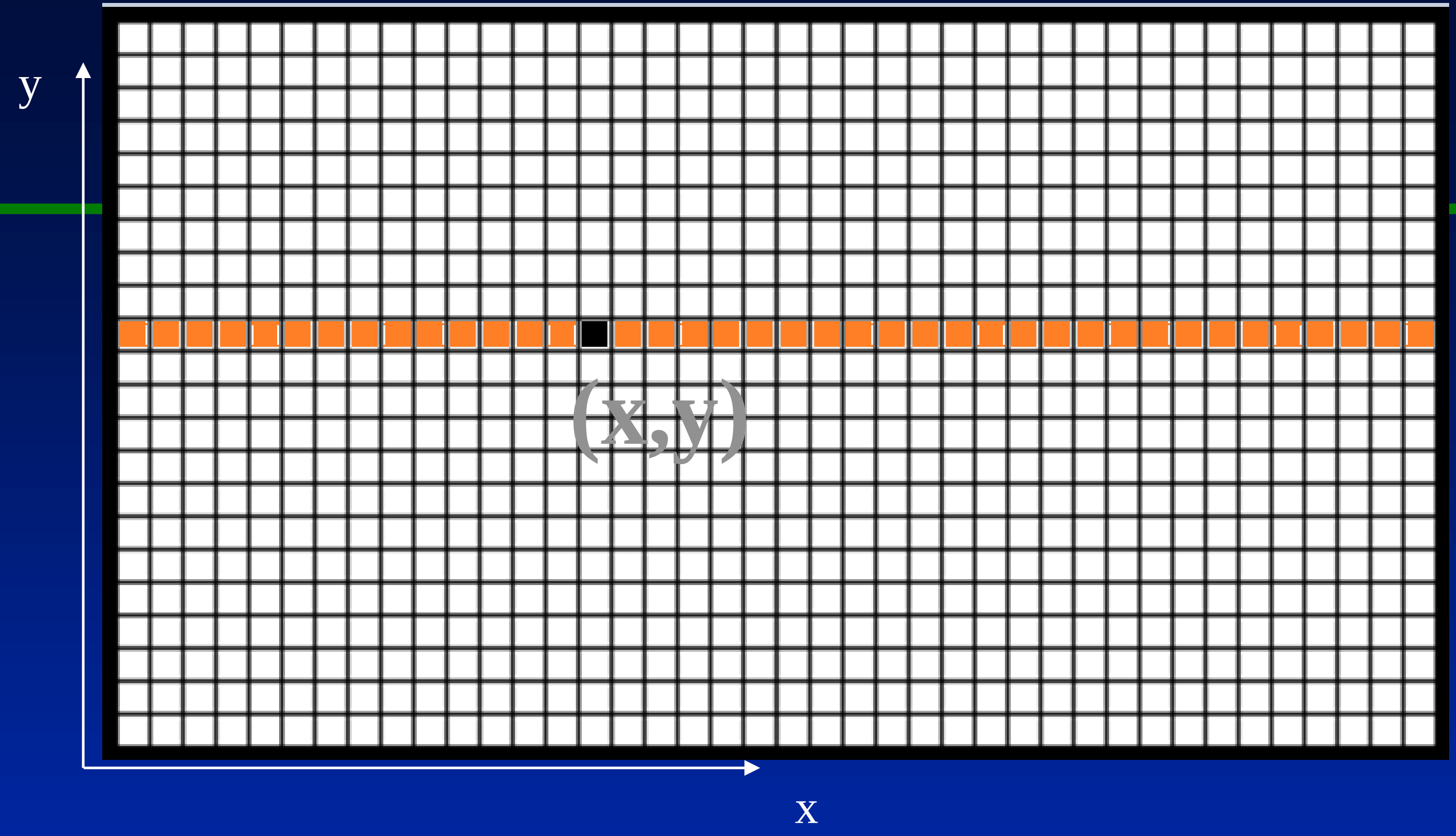


At a given scan-line, y is known.

Therefore, ax + cz = -d –by

At a given pixel value of x, x is also known

Therefore, $z = \frac{-d - by - ax}{c}$



Planar equation of a polygon:

$$ax + by + cz + d = 0$$

where a, b, c, & d are constants