Computer Graphics
Software & Hardware

NBAY 6120

Lecture 5

Donald P. Greenberg

March 15, 2018

Recommended Readings

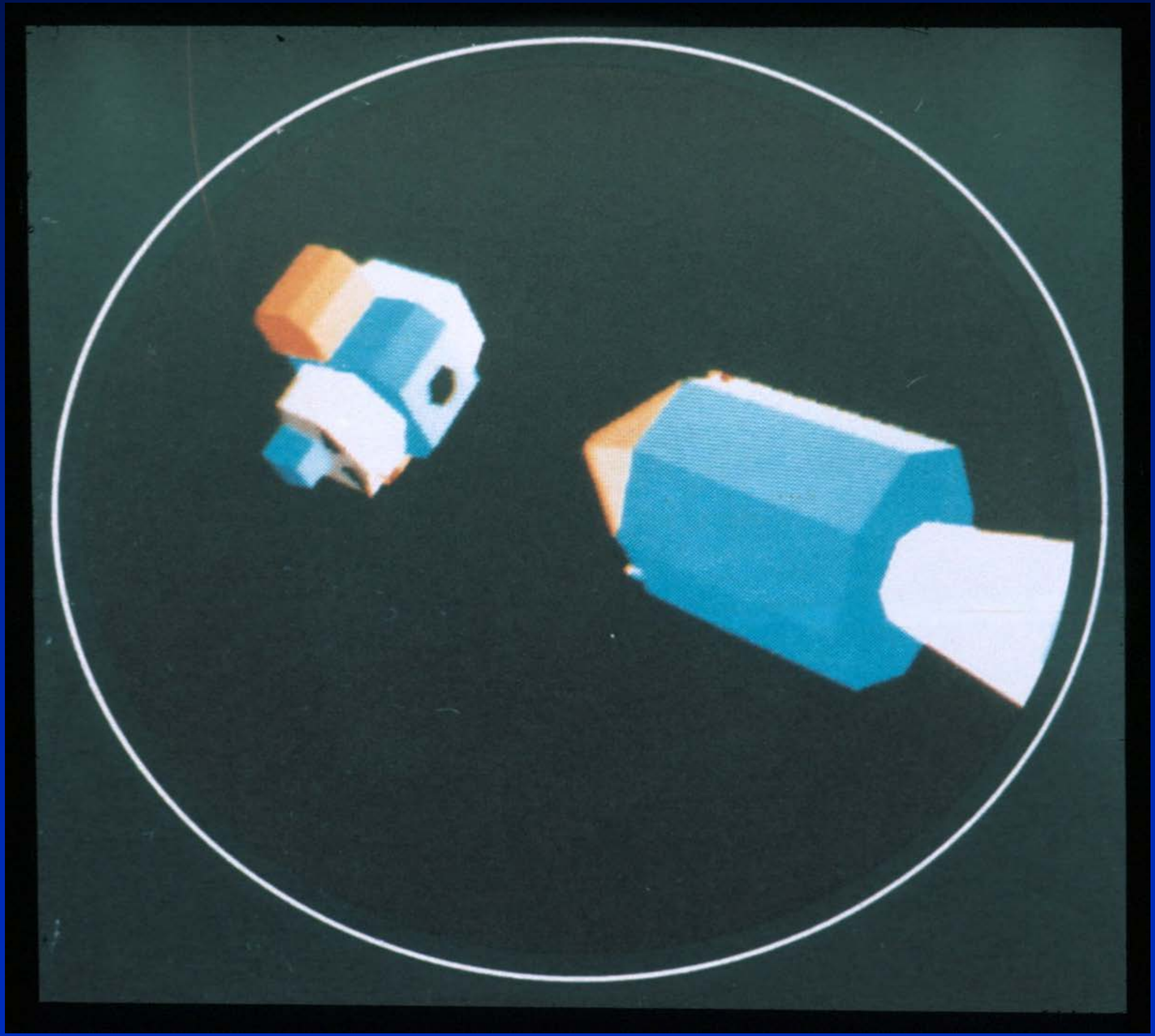
- Mike Seymour. "The State of Rendering, Part 1," fxguide.com, July 15, 2013. FXGuide.
- Mike Seymour. "The State of Rendering, Part 2," fxguide.com, July 17, 2013. FXGuide.

Why Is Graphics Important?

- 99% of our information intake is pictorial through our eyes
- Educational Modules
- Entertainment
- Games
- Advertising
- Medical
- Computer Aided Design
- Data Visualization

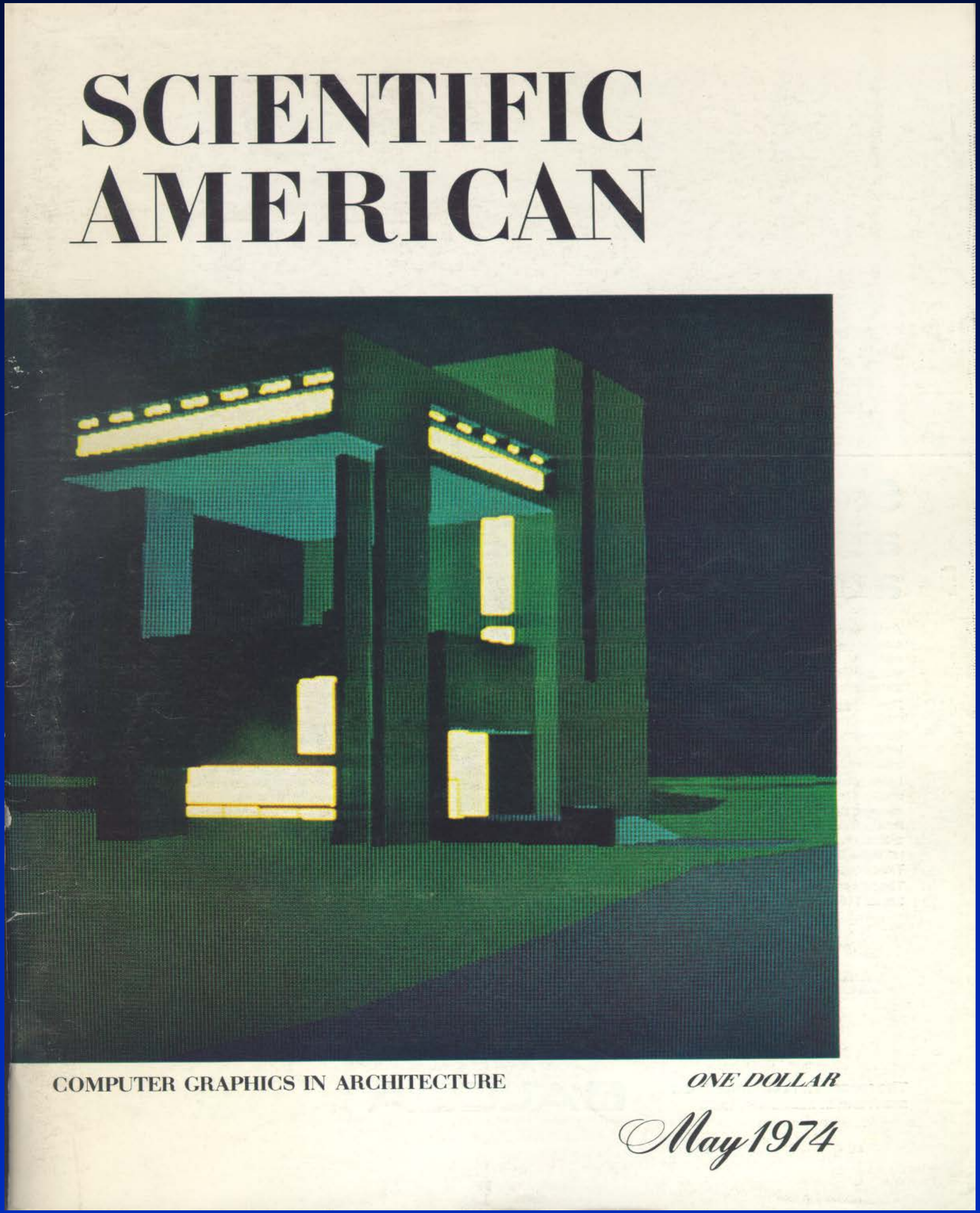
Ivan Sutherland 1963

General Electric 1967

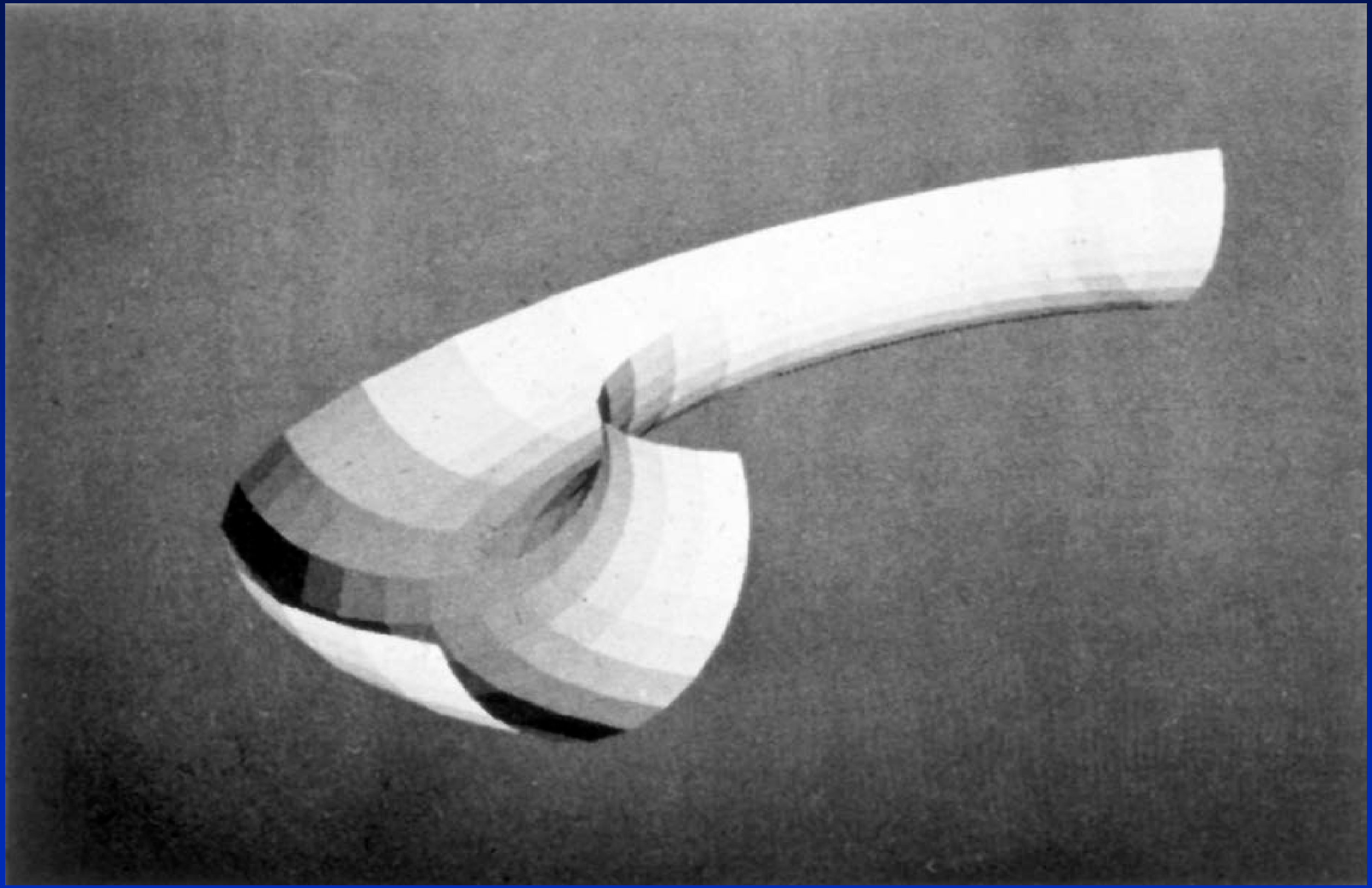


Program of Computer Graphics, Cornell University

Cornell in Perspective Film 1972


SCIENTIFIC
AMERICAN
COMPUTER GRAPHICS IN ARCHITECTURE
ONE DOLLAR
May 1974

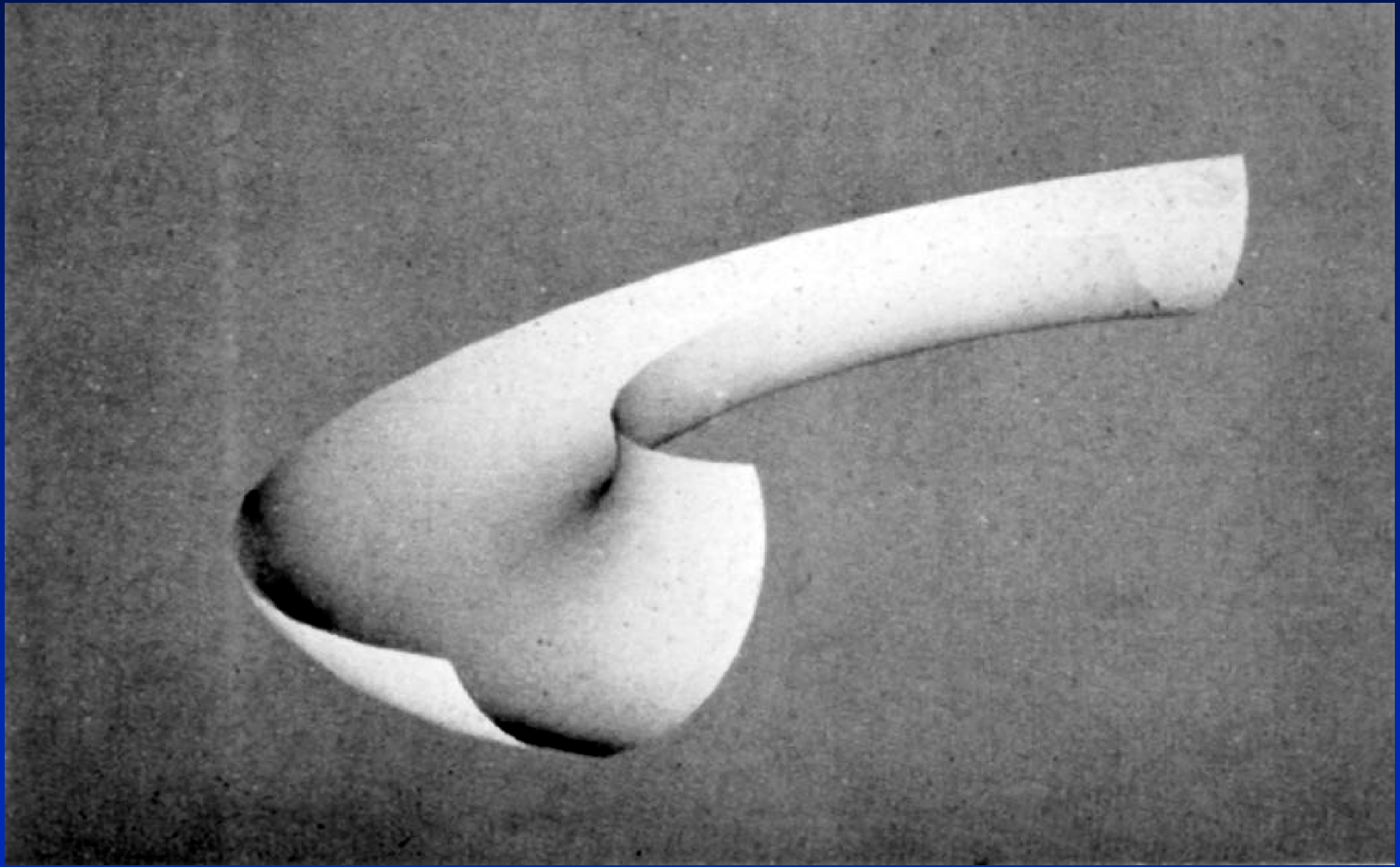
Gouraud Flat Polygon Shading 1972



Each polygon is shaded based on a single normal.

Gouraud Thesis

Gouraud Smooth Shading 1972

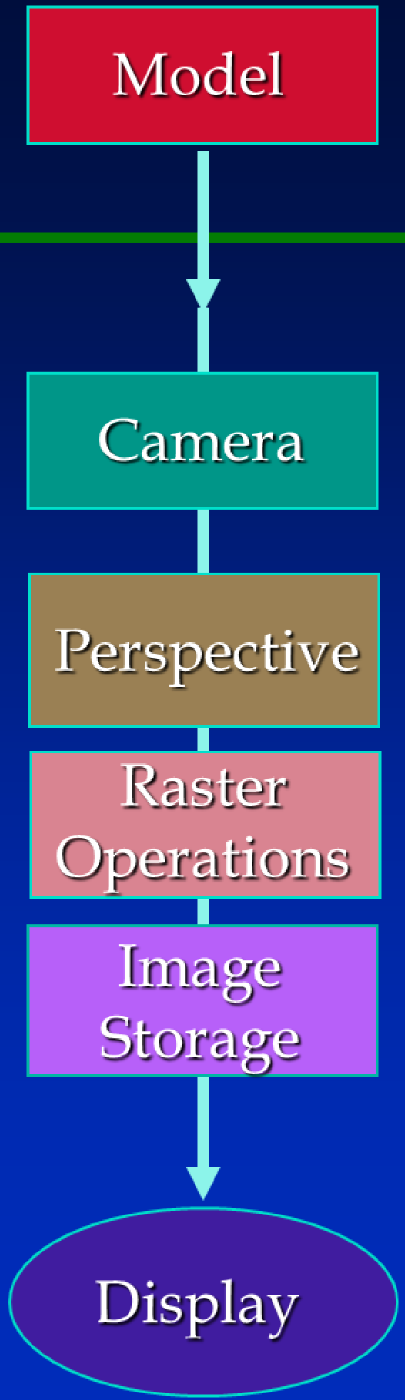


Each pixel is shaded by interpolating intensities computed in each of the polygon's vertices.

Gouraud Thesis

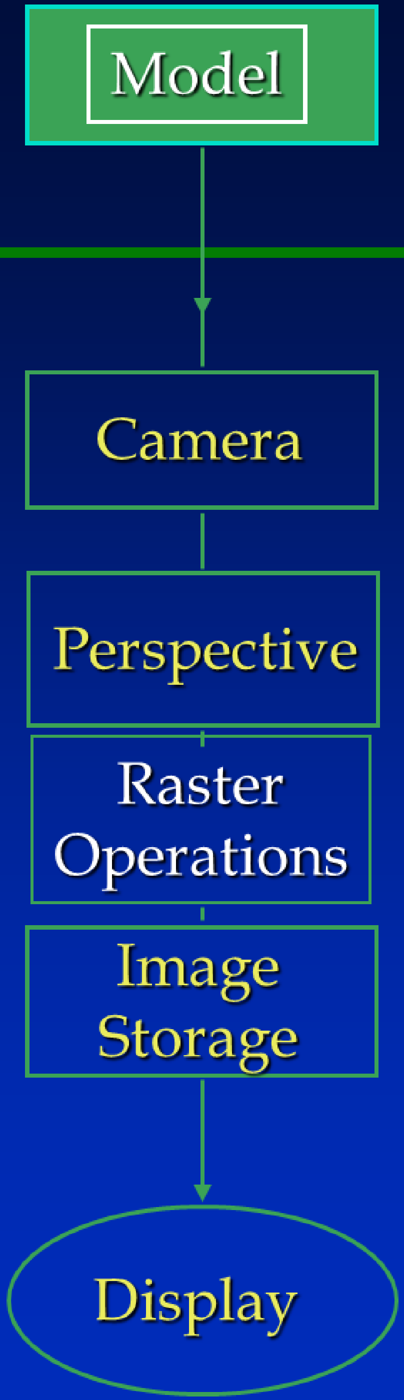
Phong Shading 1974

Direct Illumination



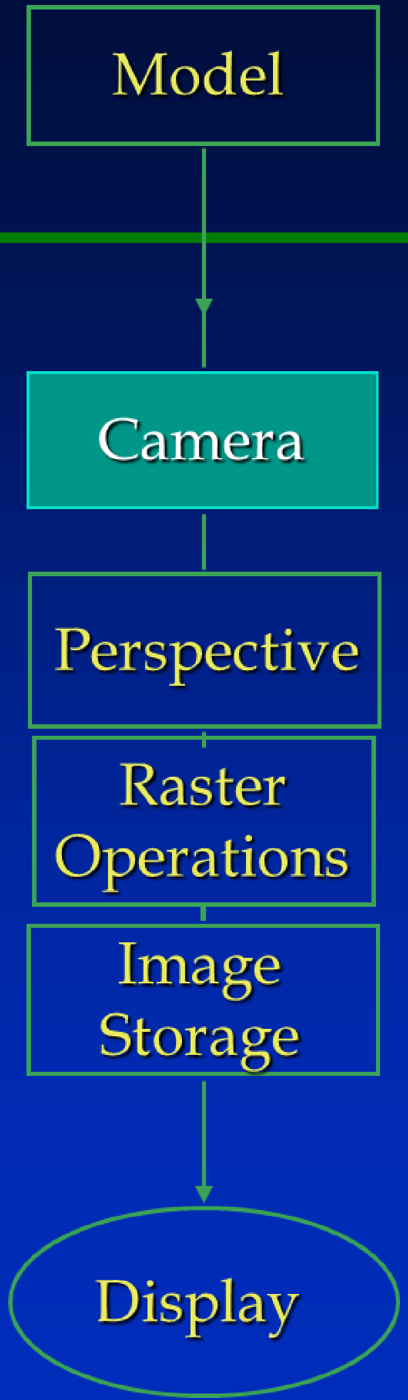
Model

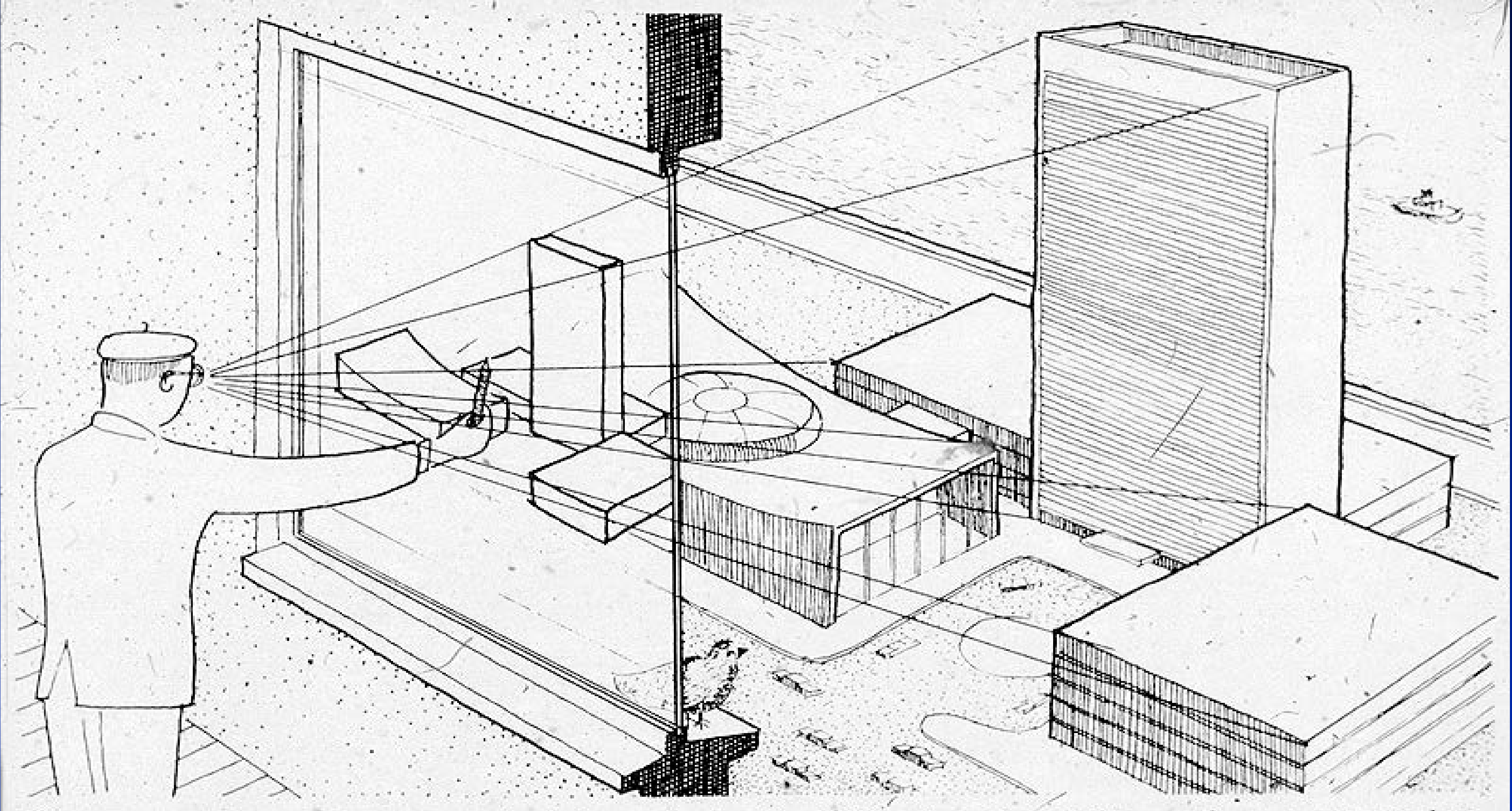
- Environment
  - Geometry & topology
  - Material Properties
    - > Color, reflectance, textures
    - > (Cost, strength, thermal properties)
- Lighting
  - Geometry & position
  - Intensity, spectral distribution
  - Direction, special distribution



Camera

- Viewer position
- Viewer direction
- Field of View
  - Wide angle
  - Telephoto
- Depth of focus
  - Near
  - Far

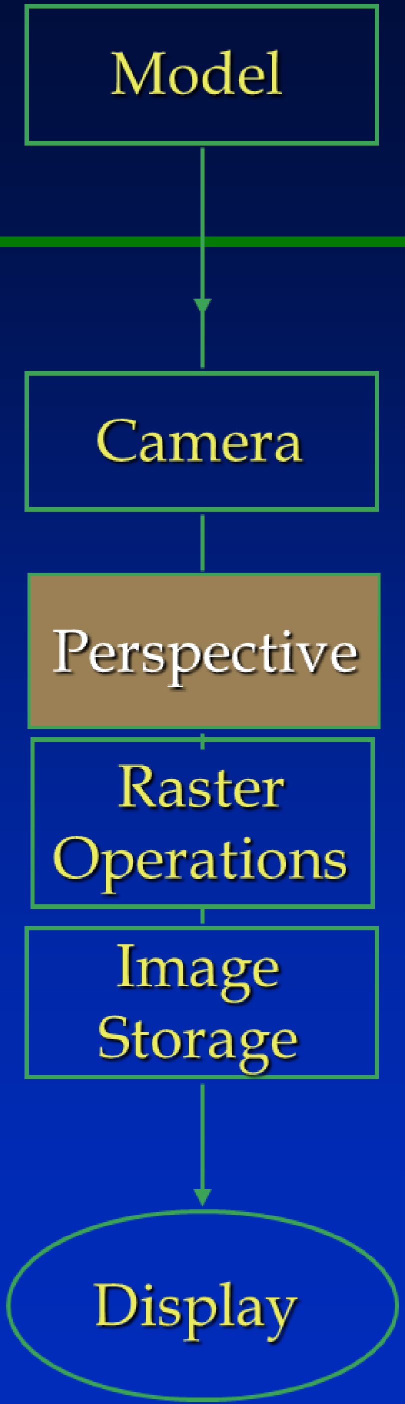




The concept of the picture plane may be better understood by looking through a window or other transparent plane from a fixed viewpoint. Your lines of sight, the multitude of straight lines leading from your eye to the subject, will all intersect this plane. Therefore, if you were to reach out with a 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.

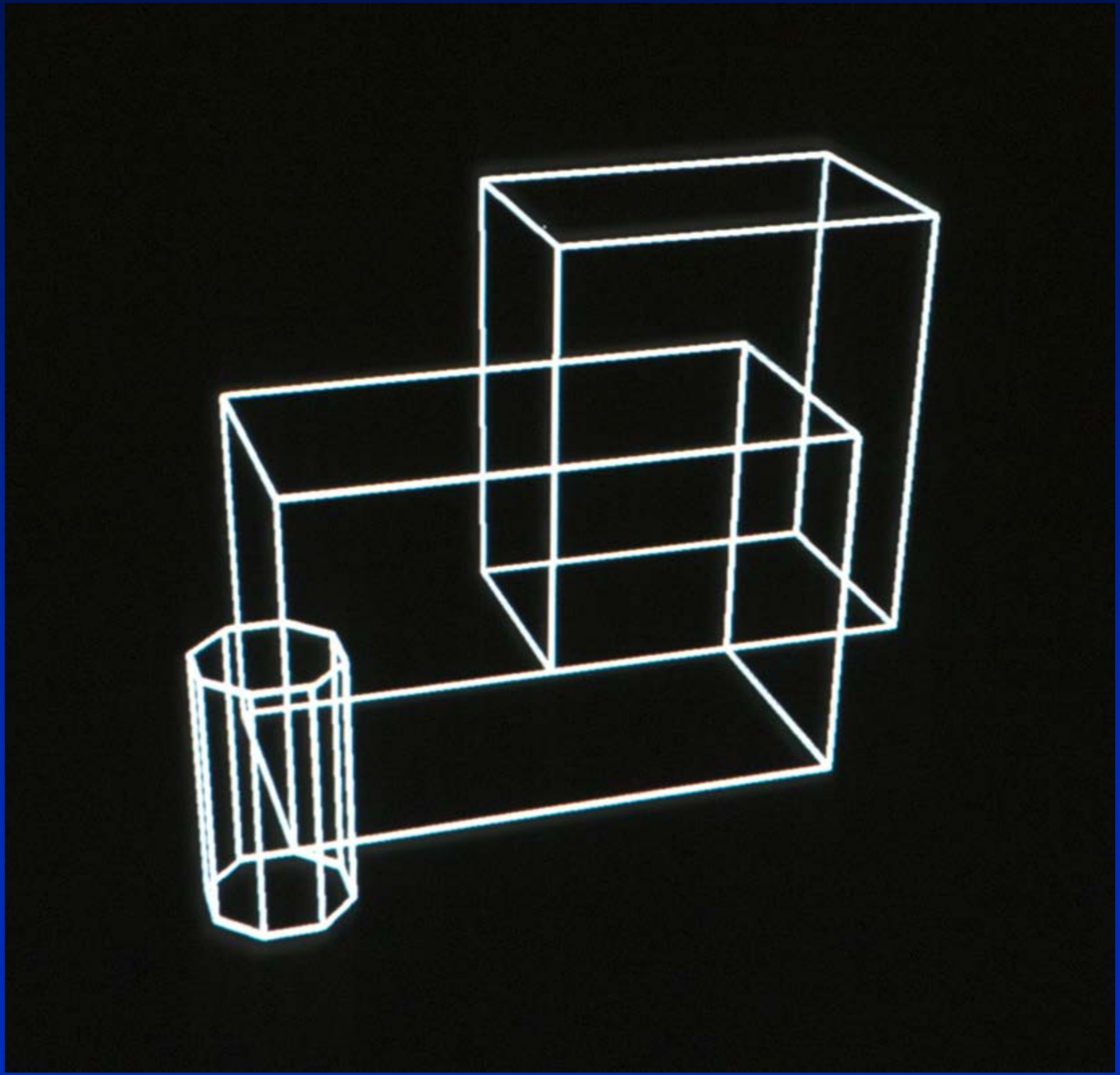
Perspective Transformation

- Perspective Transformation
  - Matrix multiplication (4 x 4)
- Clipping objects outside the field of view
- Culling back-facing surfaces

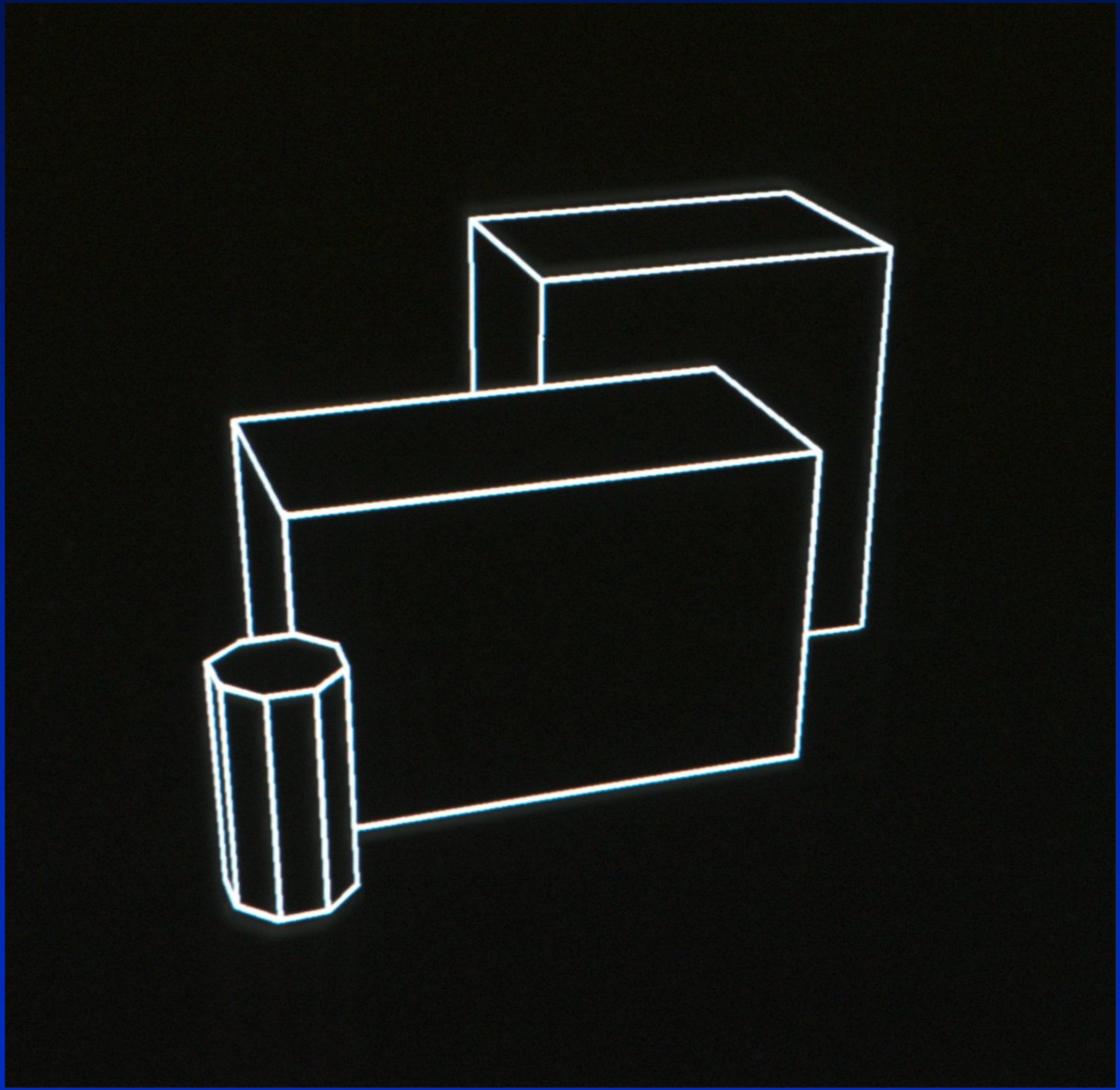


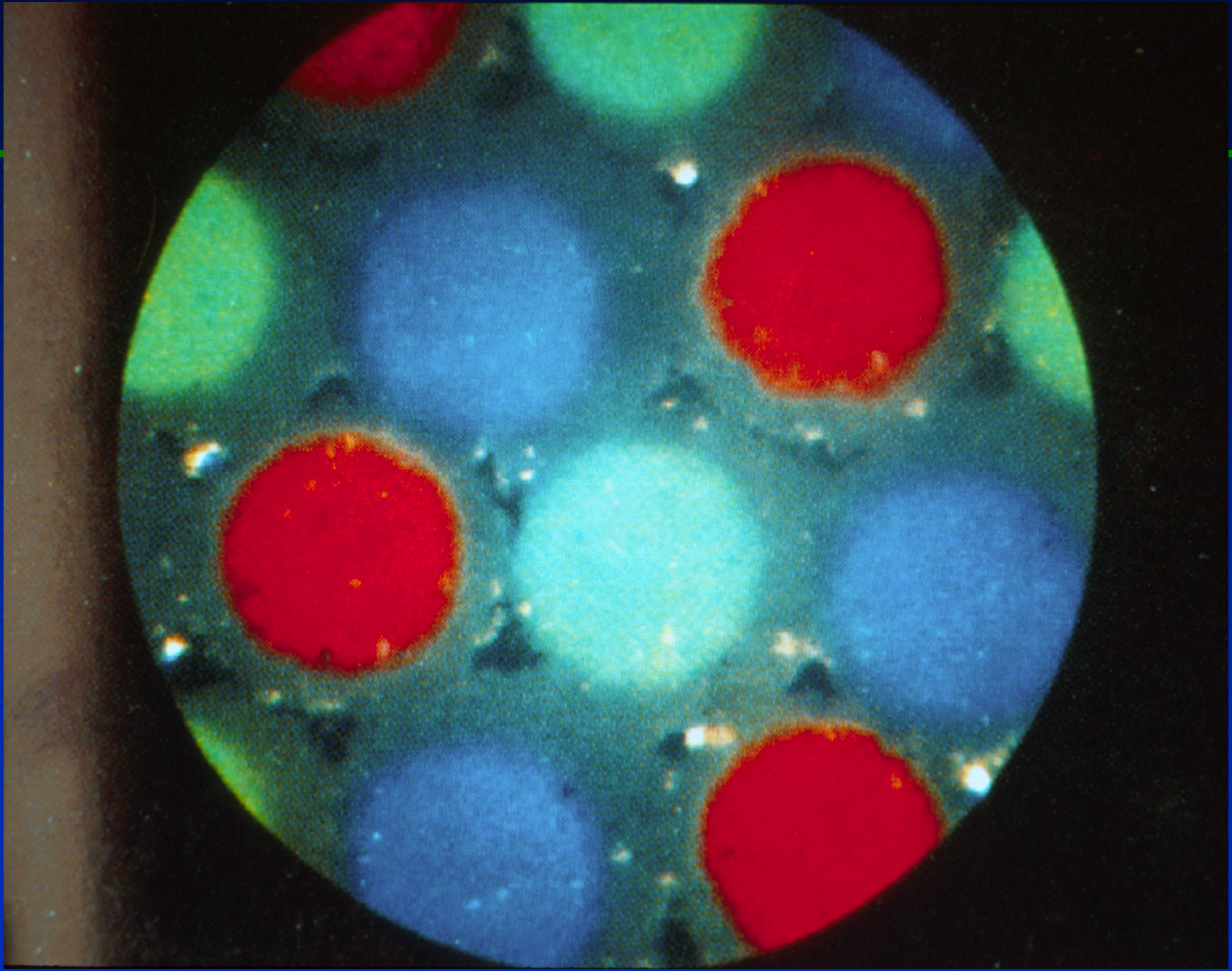
Brunelleschi’s Experiment

Hidden Line Algorithm



Hidden Line Algorithm





Raster Operations

- Conversion from polygons to pixels
  - Color computation
- Hidden surface removal (z-buffer)

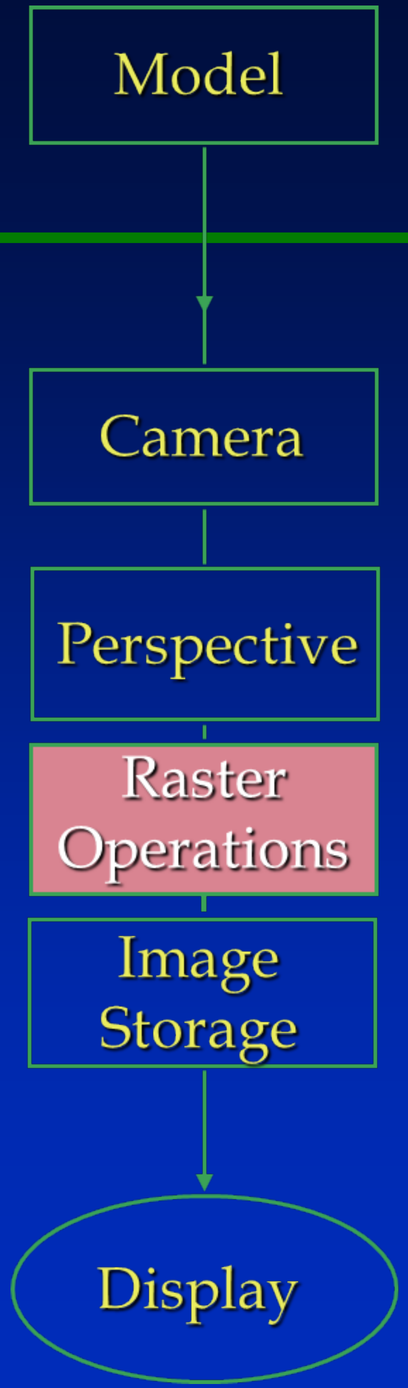
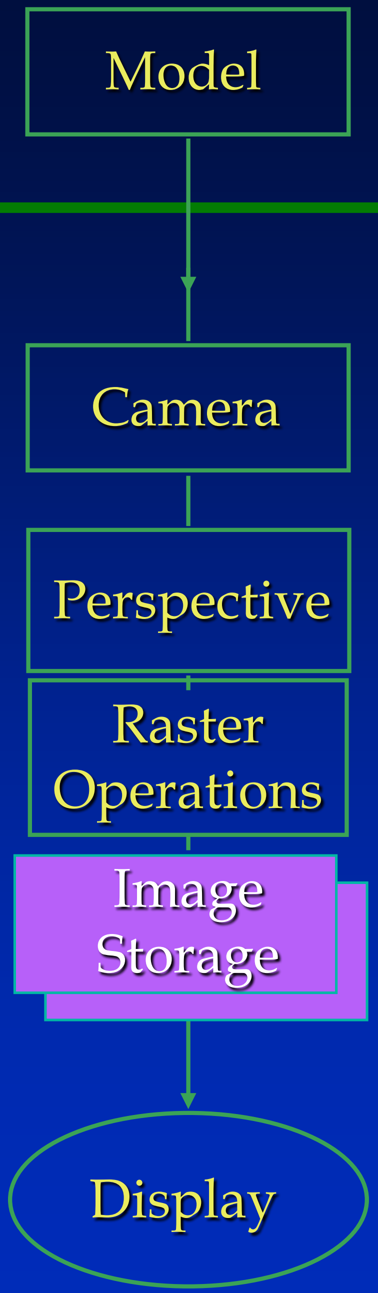


Image Storage

- 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



Display

- Digital to analog conversion
  - 1280 x 1024 resolution
  - 60 frames/second
- Total data rate

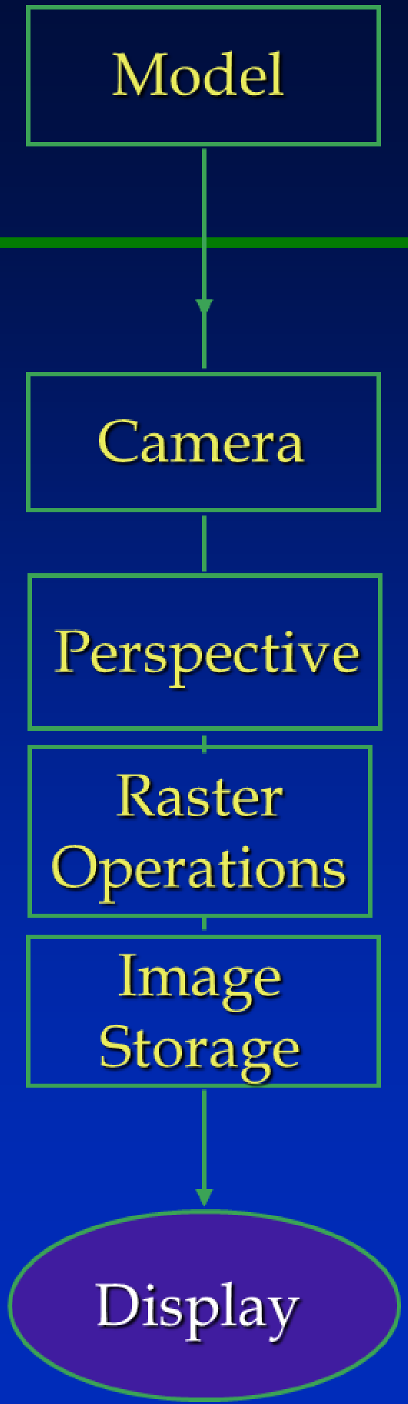
  1 ¼ million pixels

  x 3 bytes/pixel

  x 60 frames/second

  = 225 megabytes/second

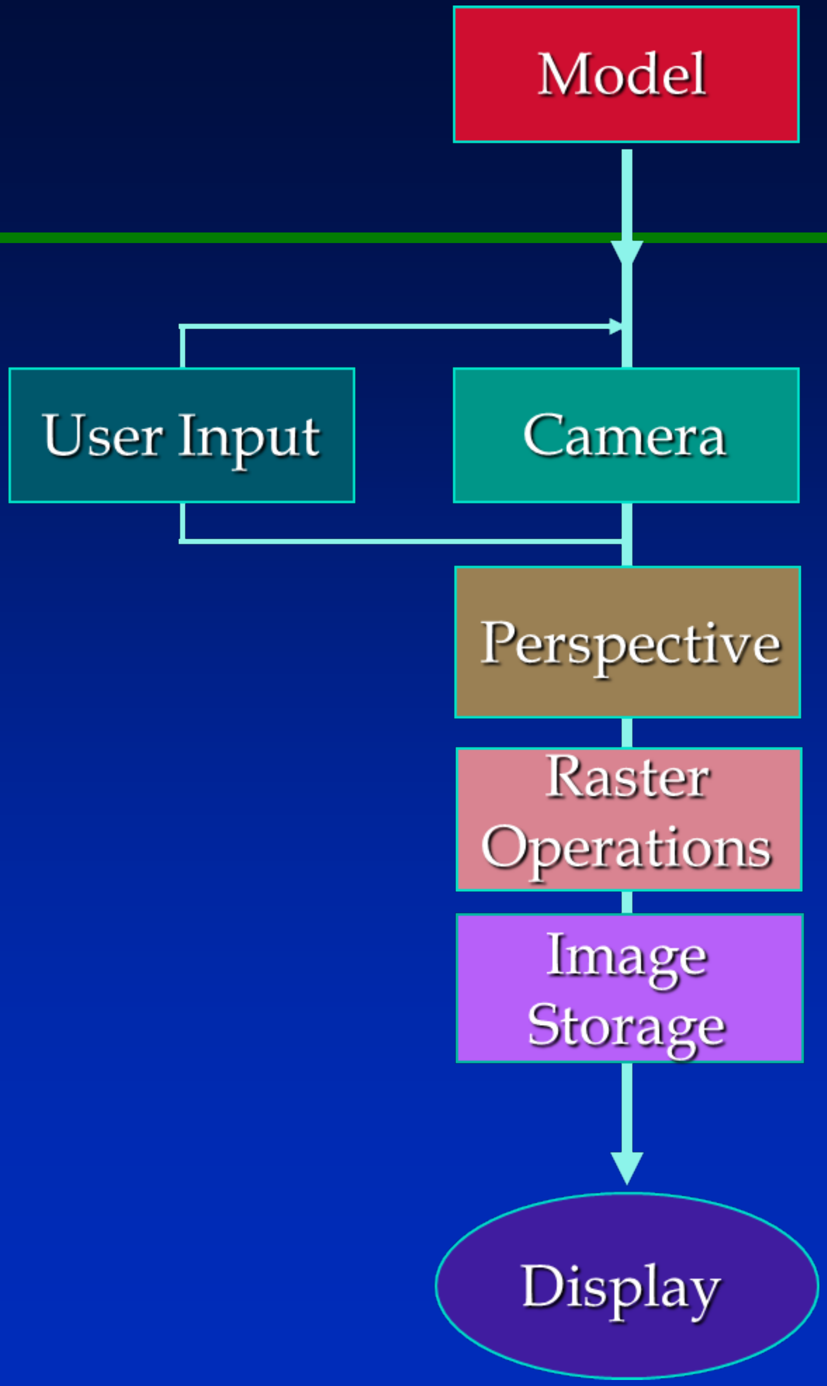
  = 1.8 gigabits/second



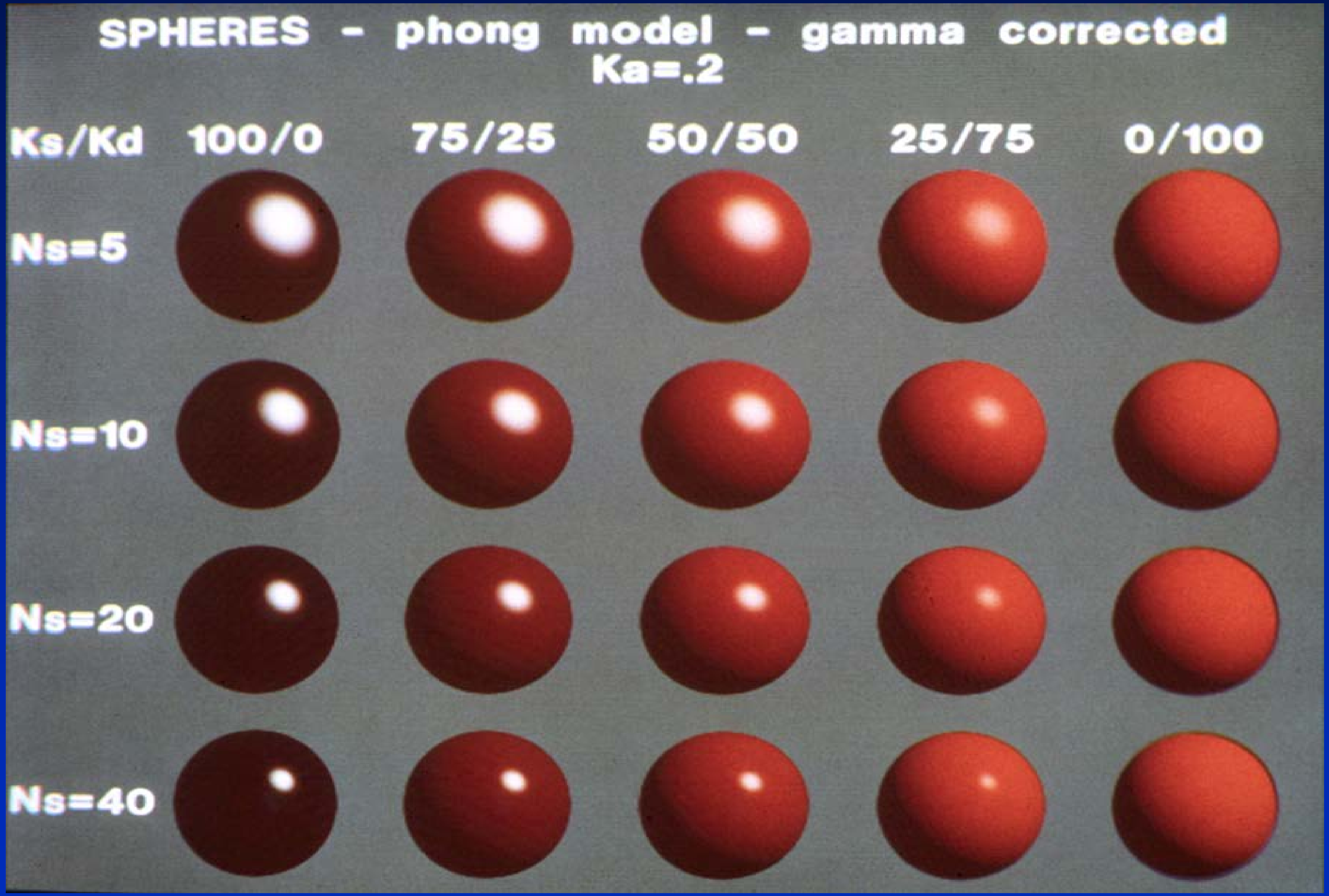
Refrsh vs. Update Rate

- 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

Direct Illumination



Phong Model: Variations of Specular Exponent



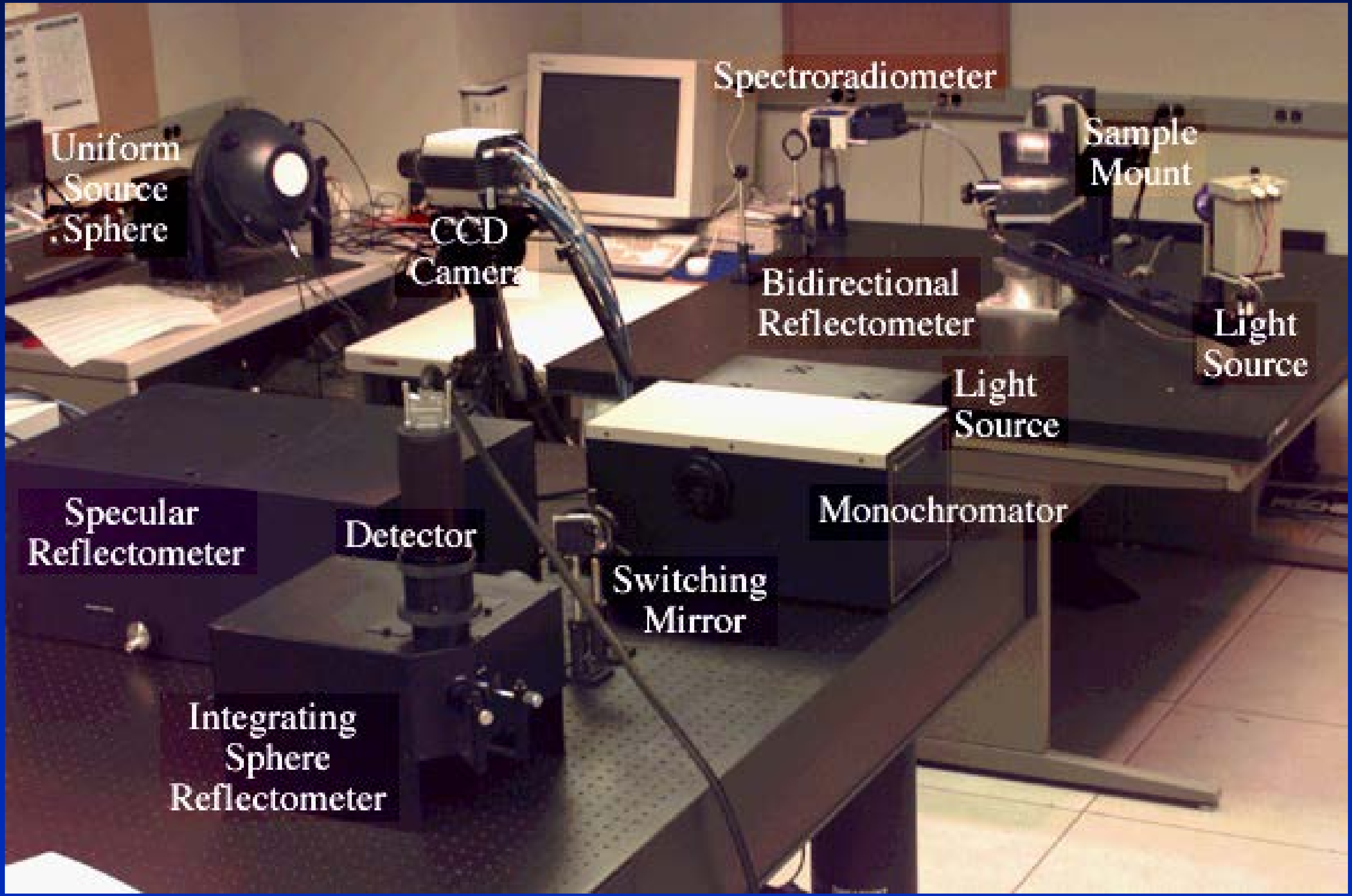
Roy Hall

Reflection Descriptions

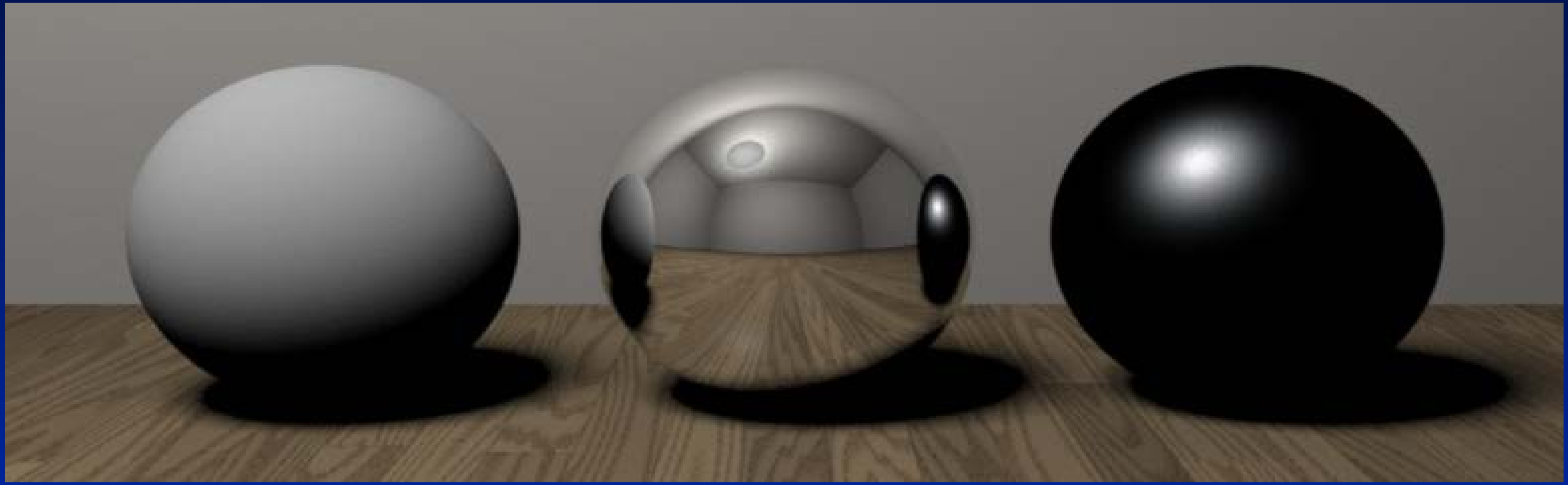
- Images weren’t realistic
  - Poor material representations
  - Lack of global illumination
- Need to measure how light reflects

  Need to derive algorithms to compute global illumination

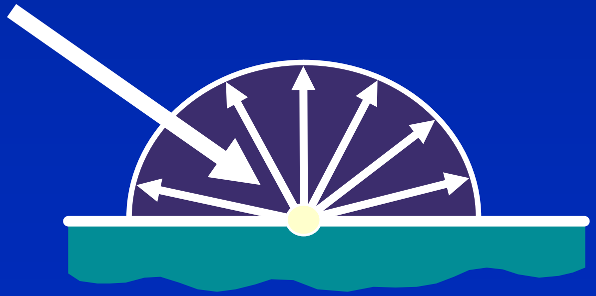
Light Measurement Laboratory

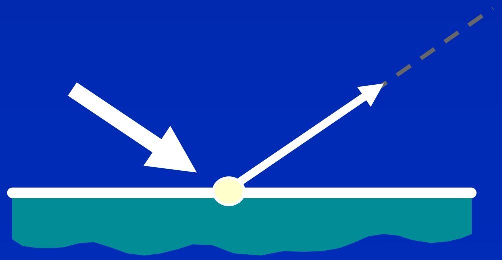


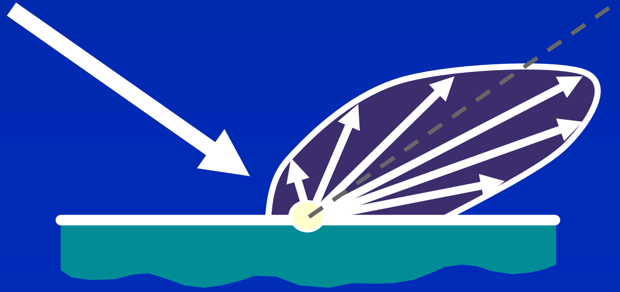
Reflectance   Three Approximate Components



Ideal diffuse (Lambertian) | Ideal specular | Directional diffuse

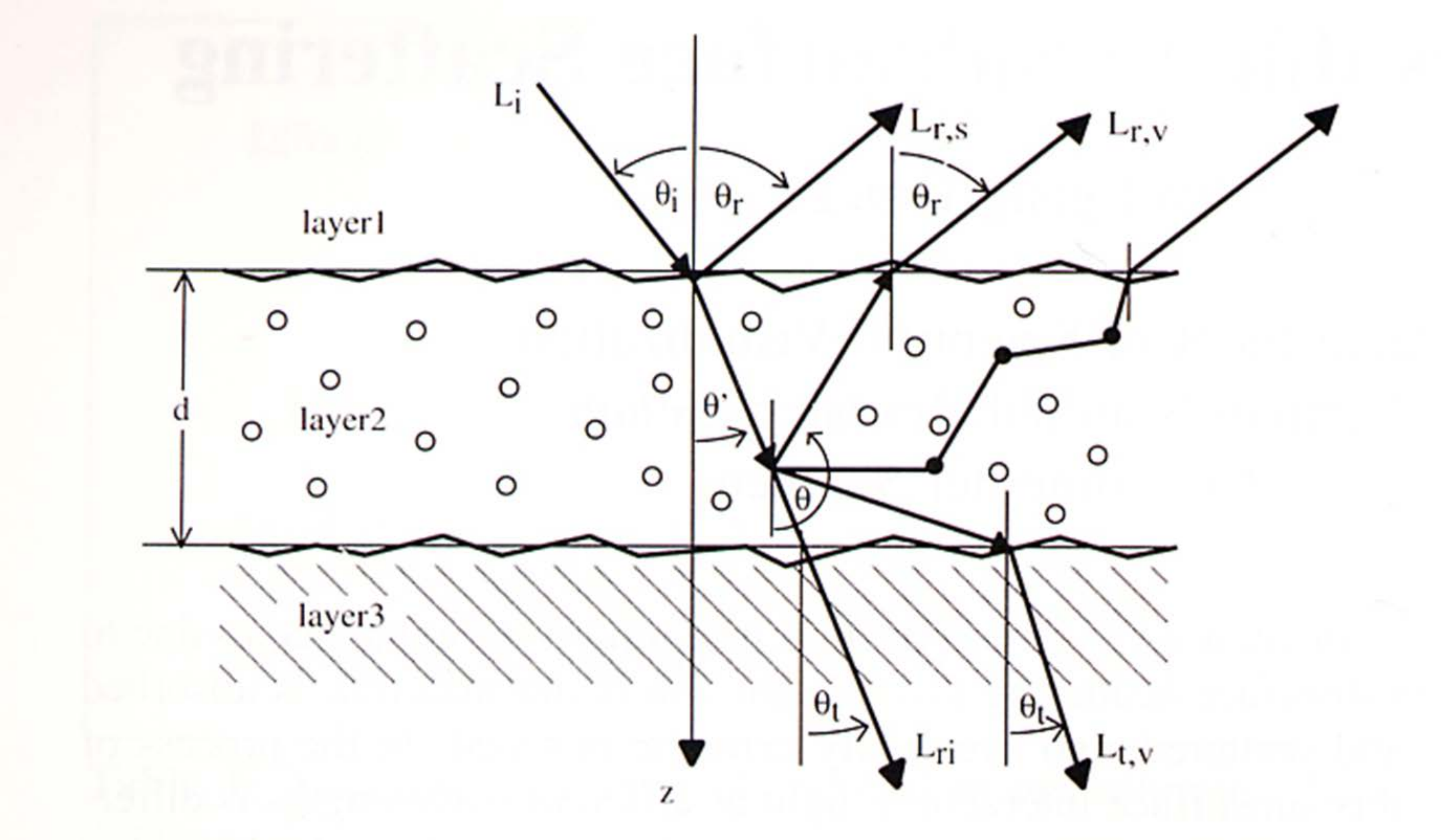






Cook-Torrance Renderings 1979

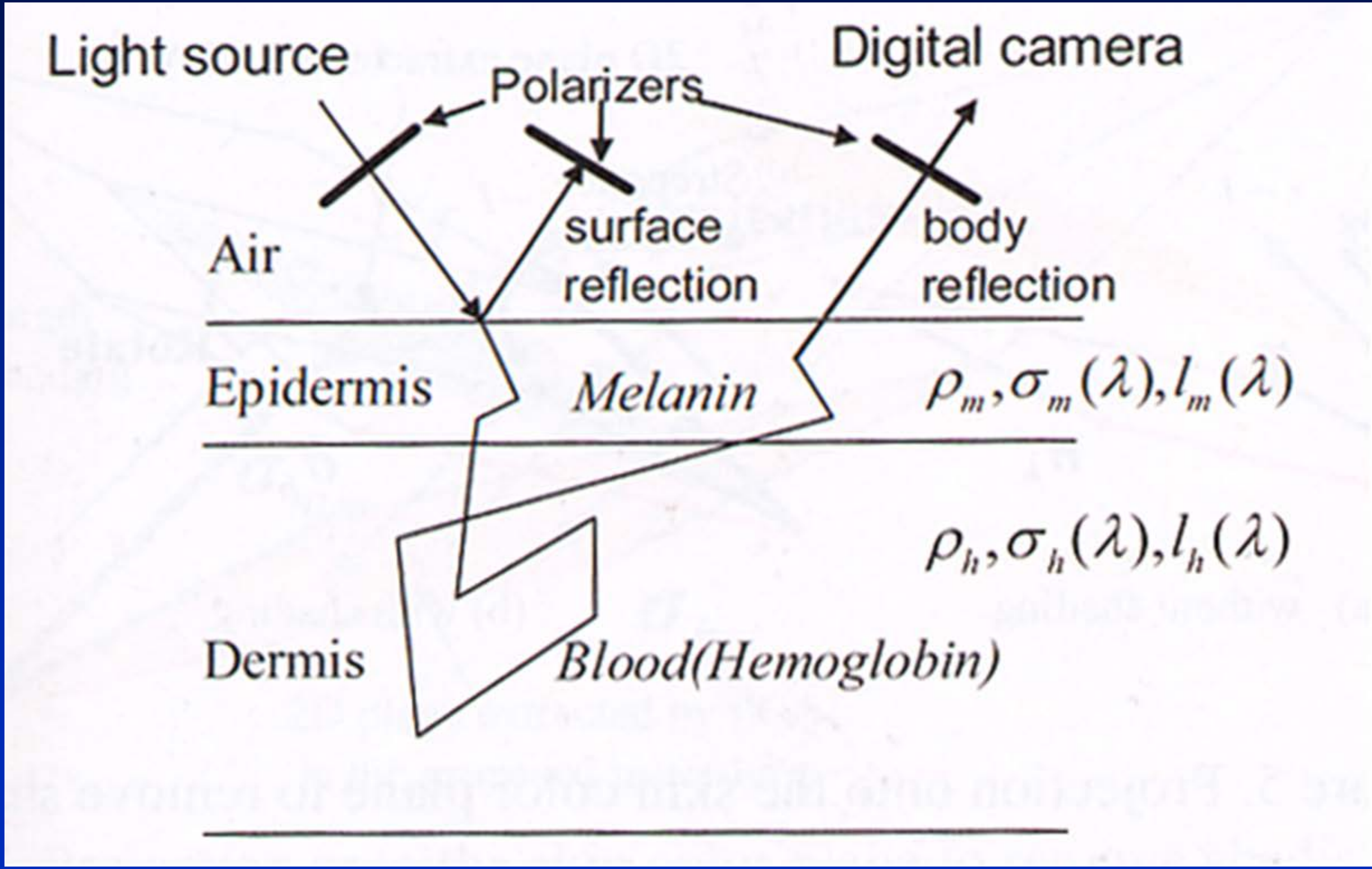
The Geometry of Scattering from a Layered Surface



acm Computer Graphics, Siggraph 1993 p. 166

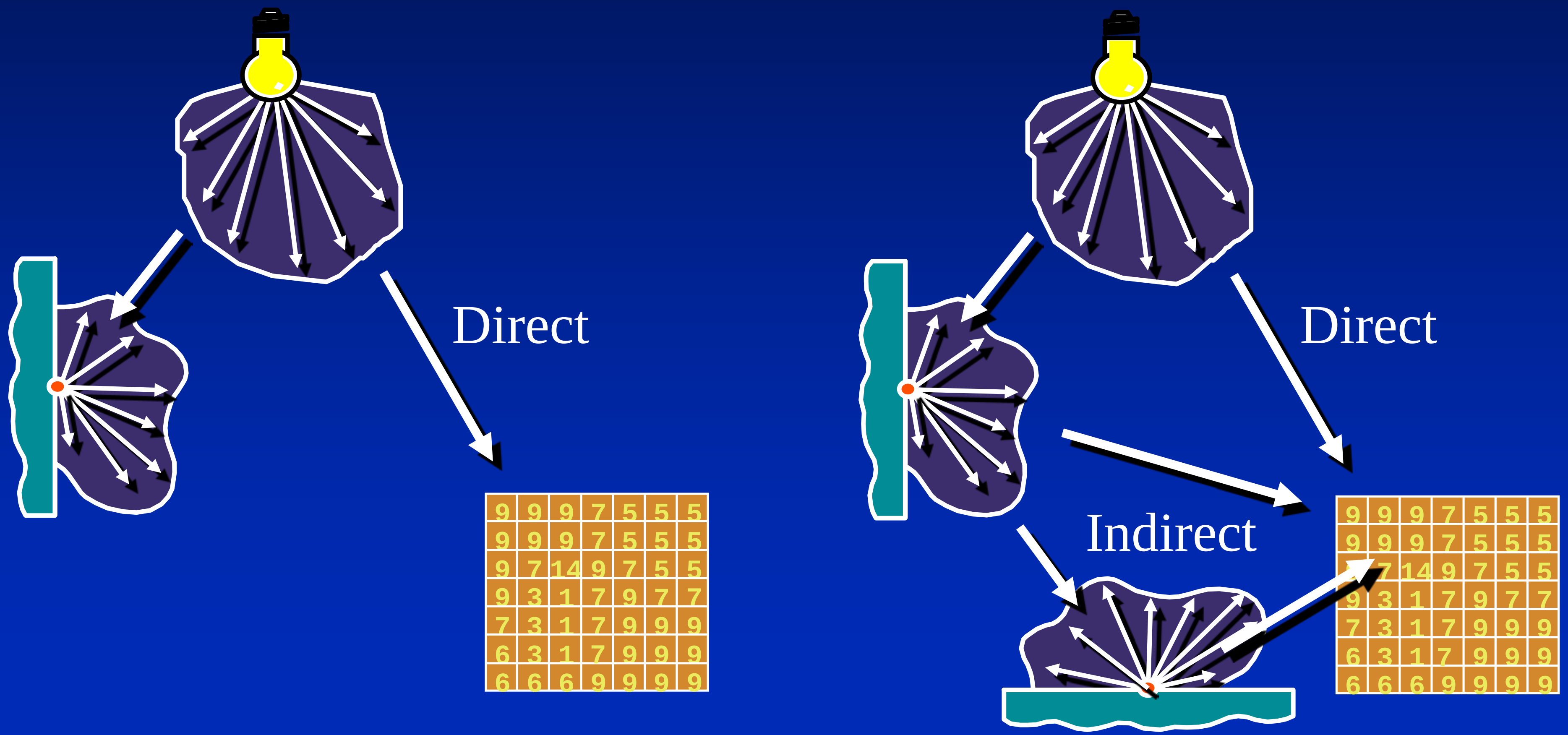
Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, Pat Hanrahan. "A Practical Model for Subsurface Light Transport," ACM Siggraph 2001, August 2001, Los Angeles, CA, pp. 511-518.

Schematic Model of the Image Process

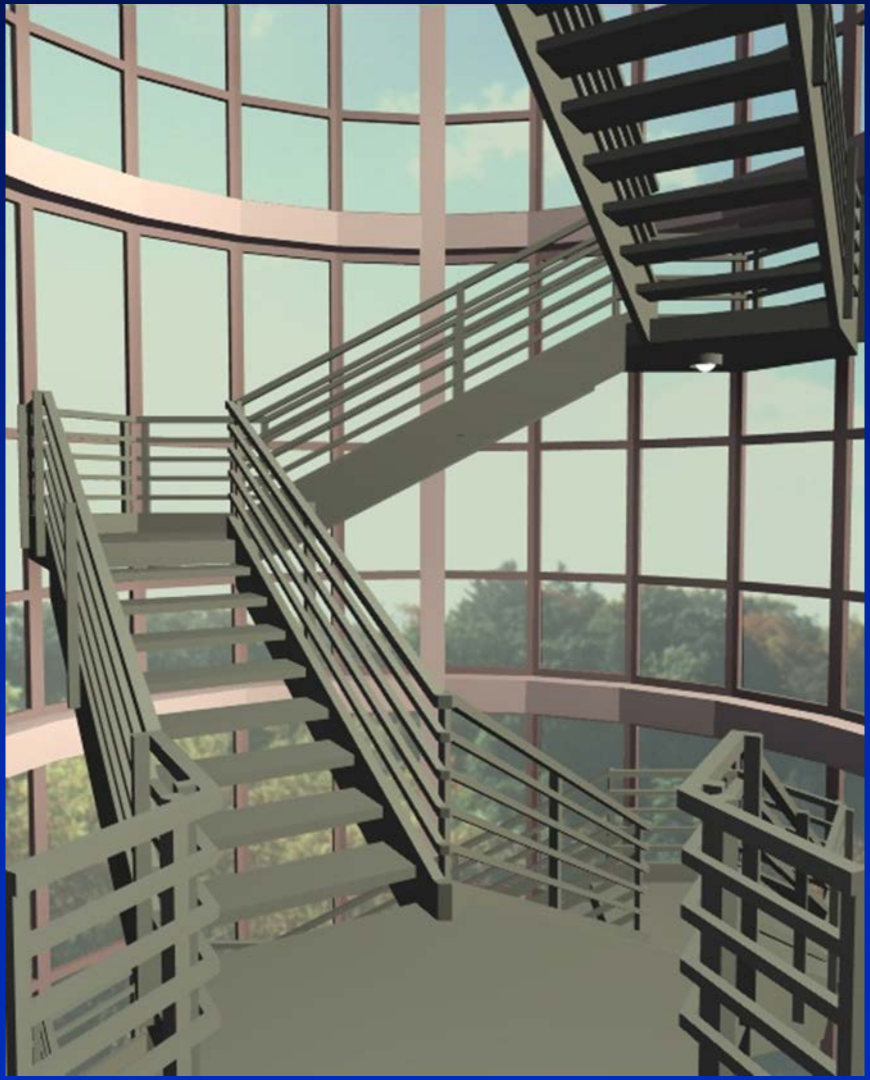


acm Transactions on Graphics, Siggraph 2003 p. 773

Direct Lighting and Indirect Lighting

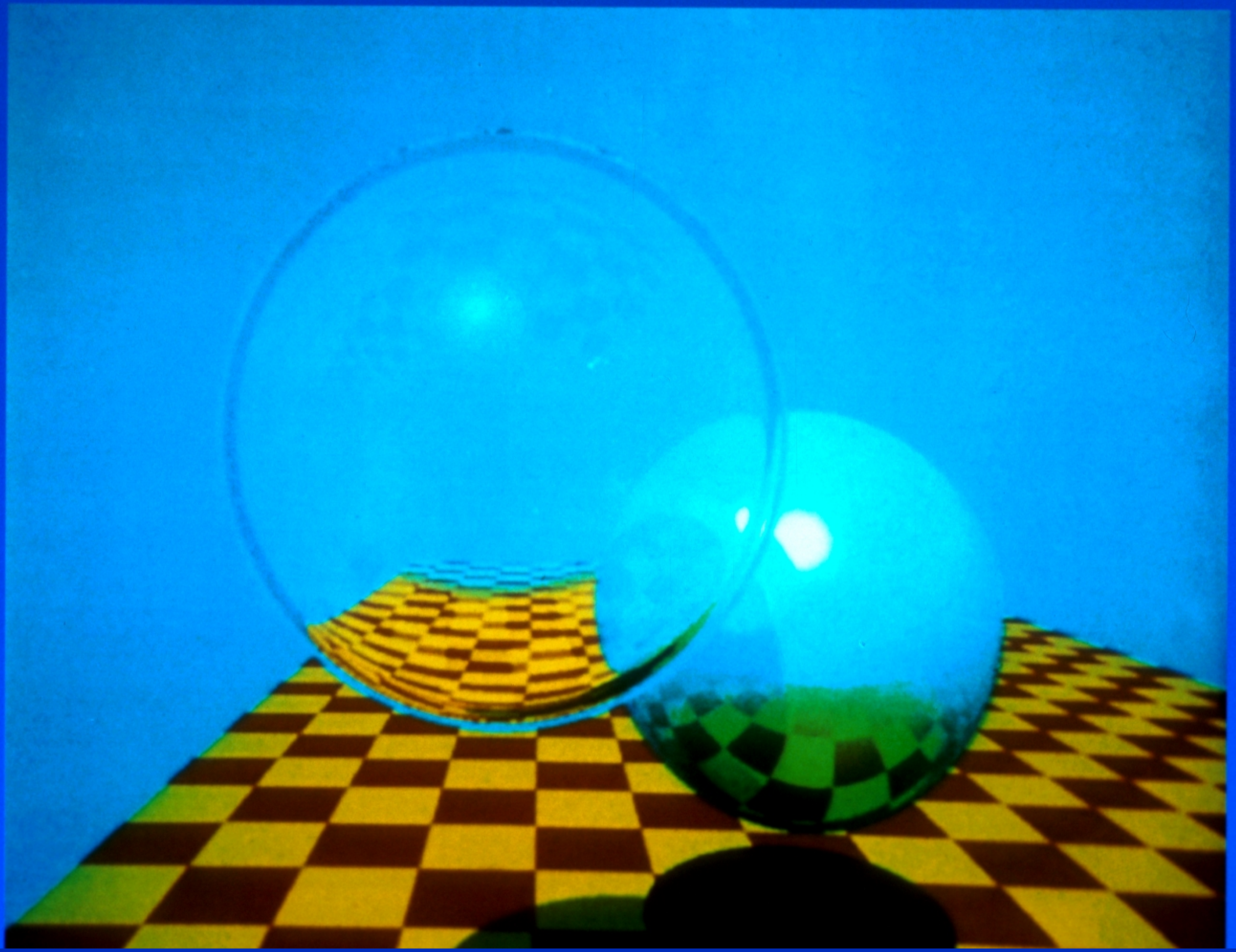


Direct Lighting Only



Global Illumination

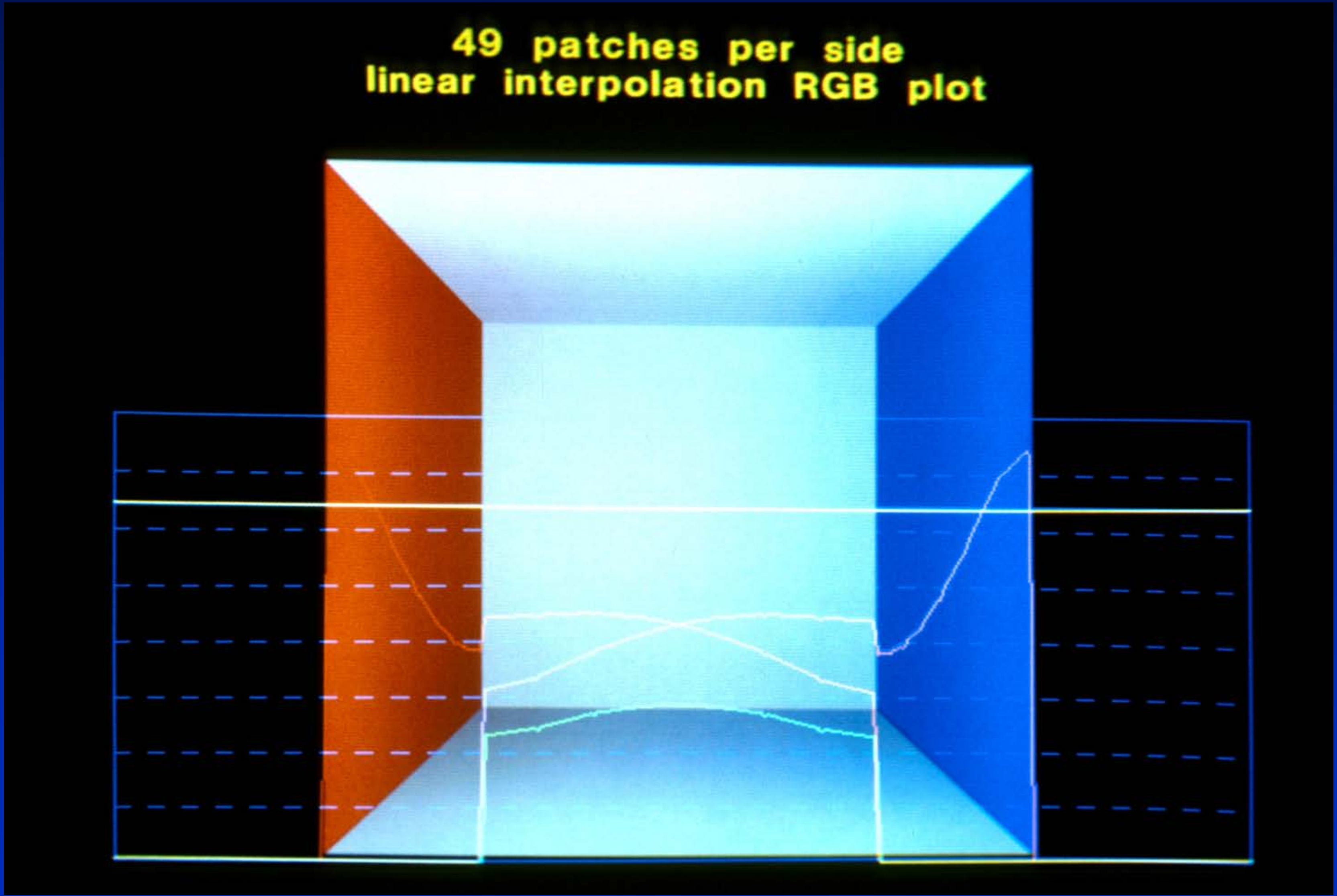
Ray Tracing
Turner Whitted, 1979



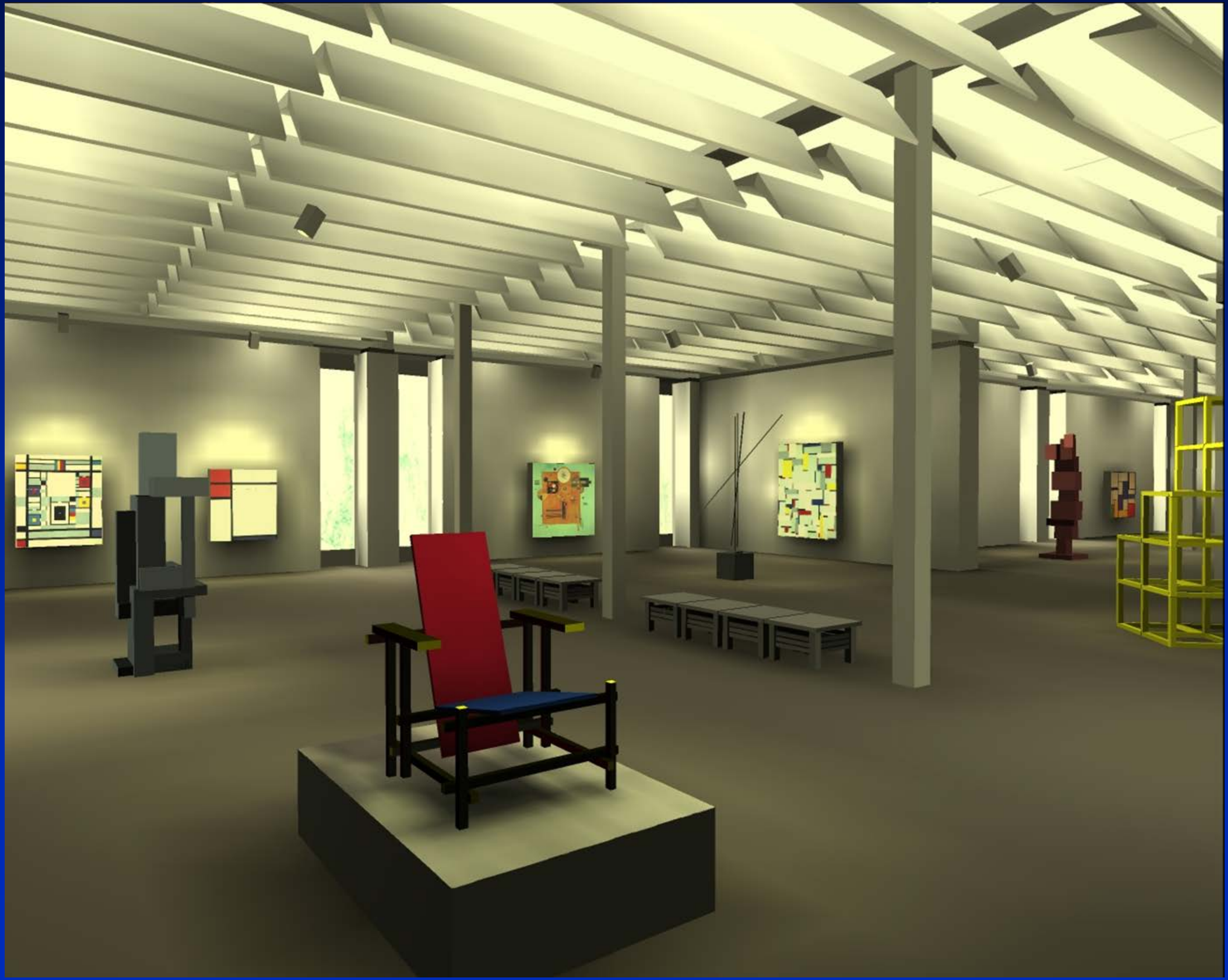
Ray Tracing Eric Haines 1985

Ray Tracing Jason Ardizzone 1990s

Radiosity 1984

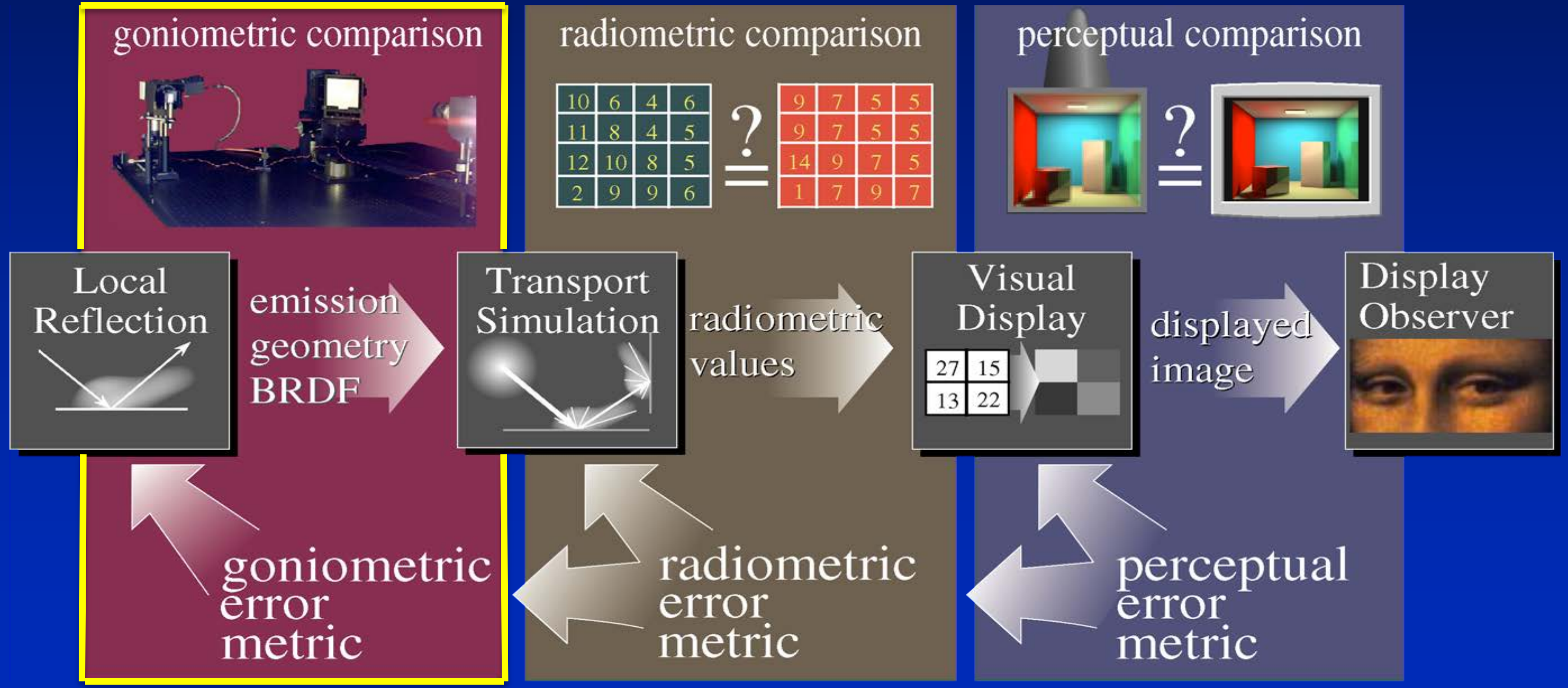


Radiosity Eric Chen 1986

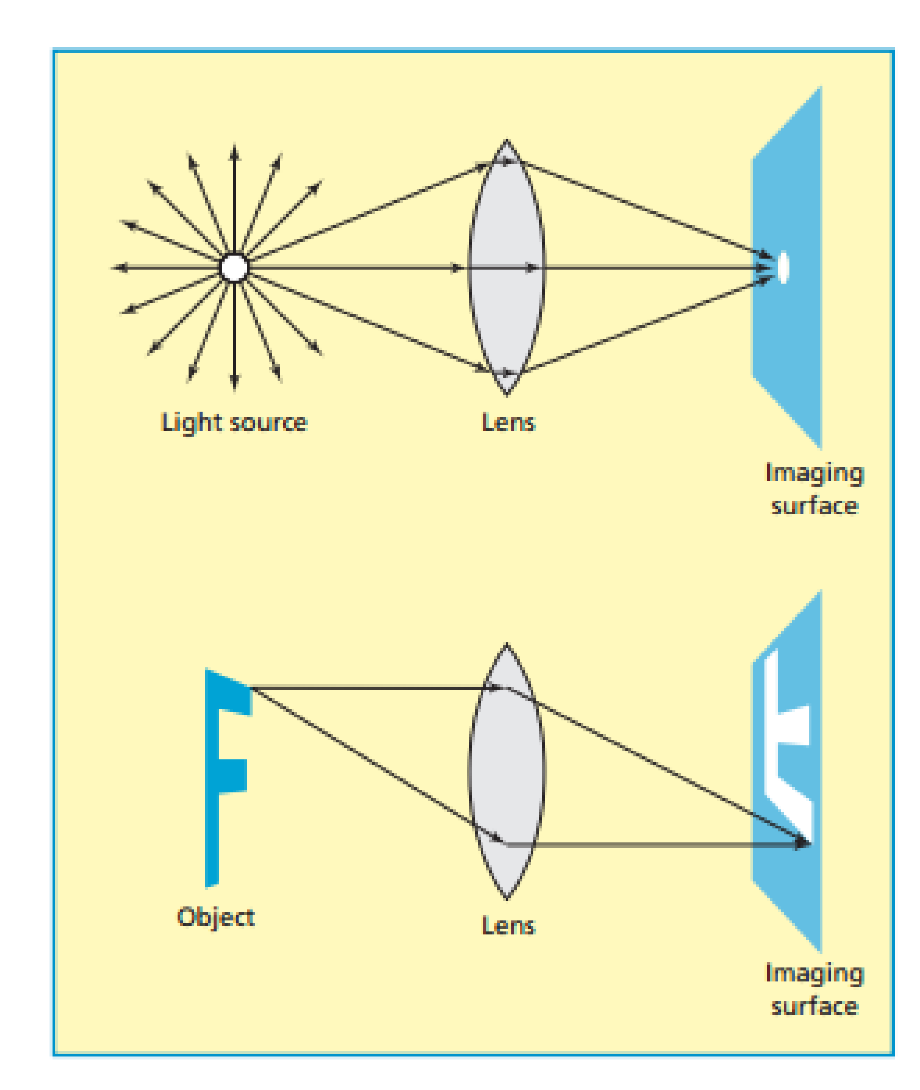


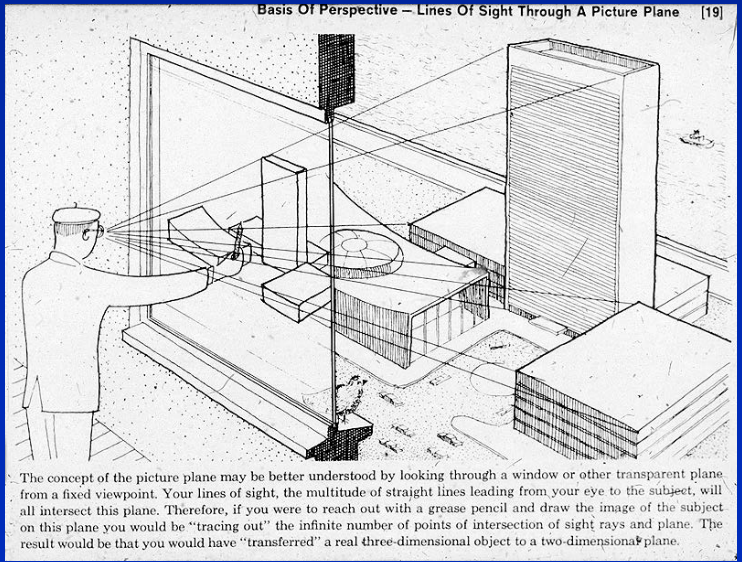
Radiosity 1990s

Rendering Framework 1997

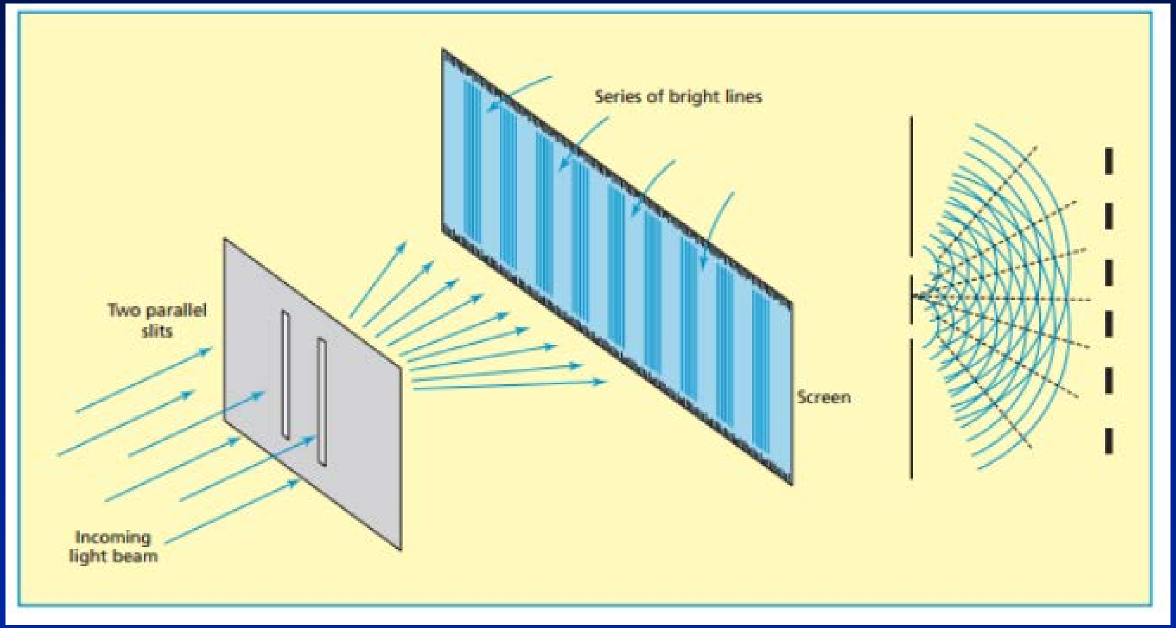


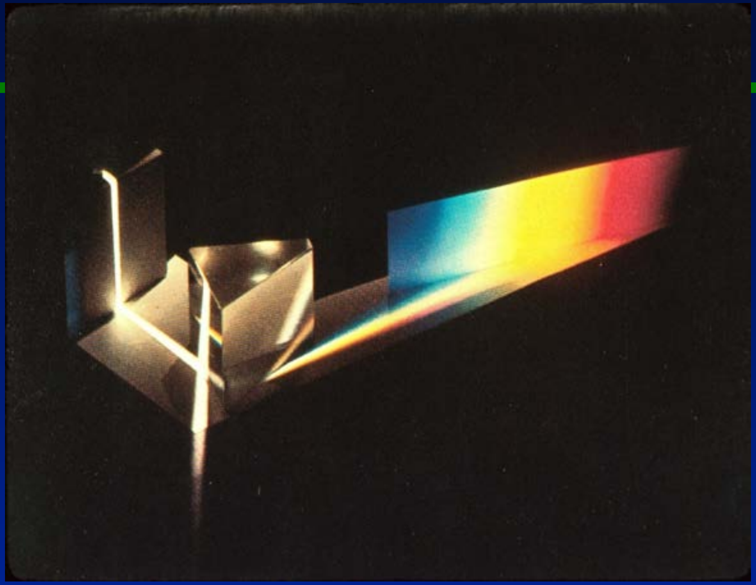
Light as Rays

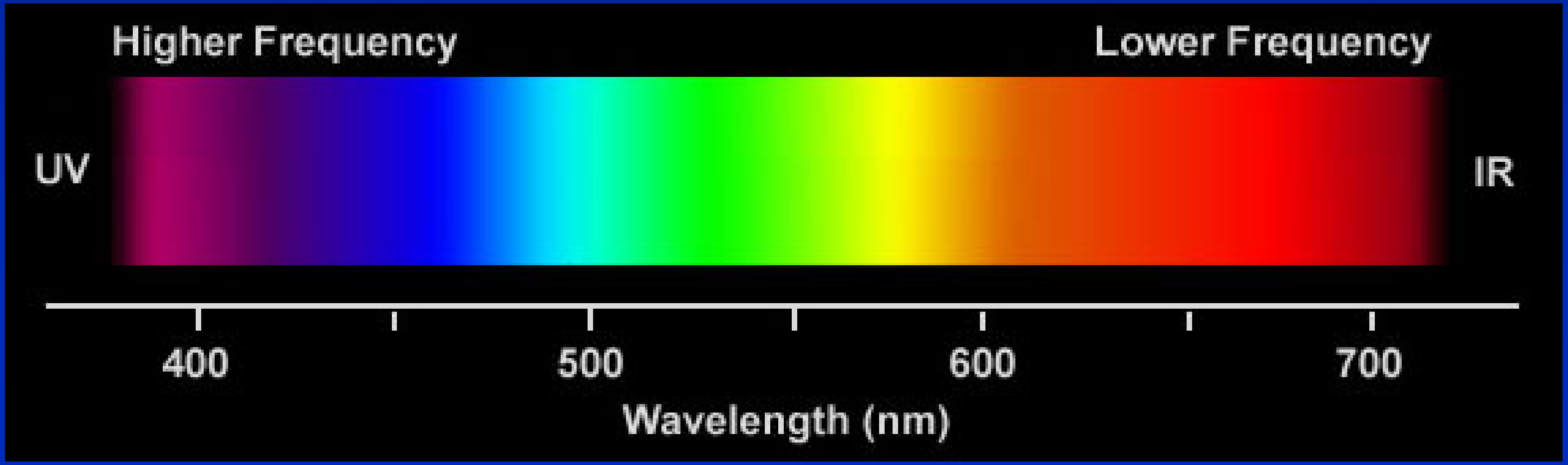




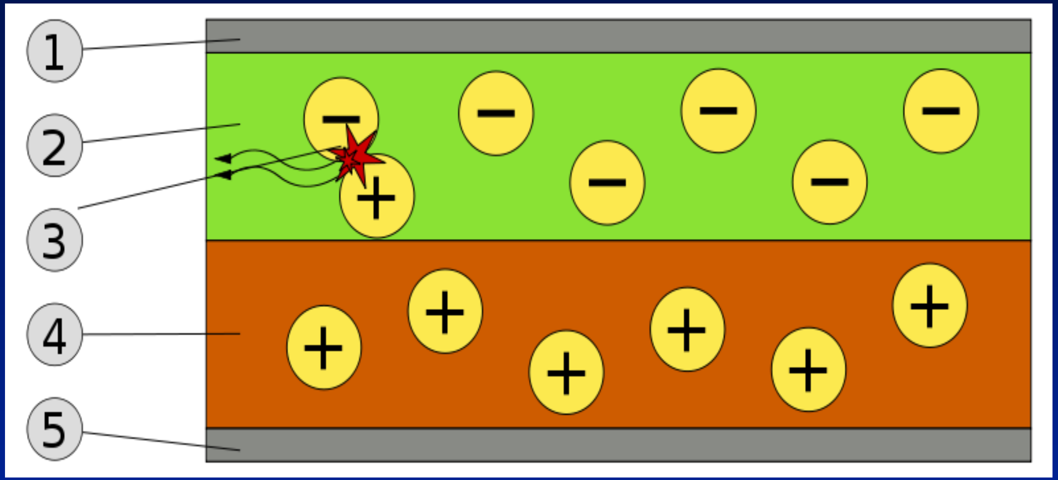
Light as Waves

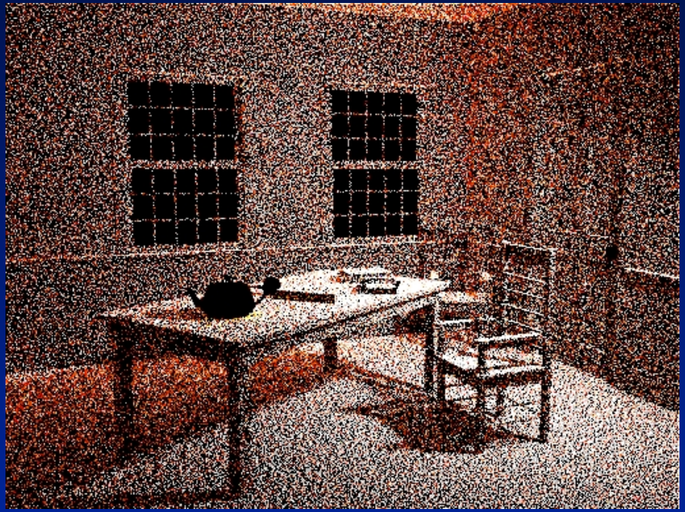


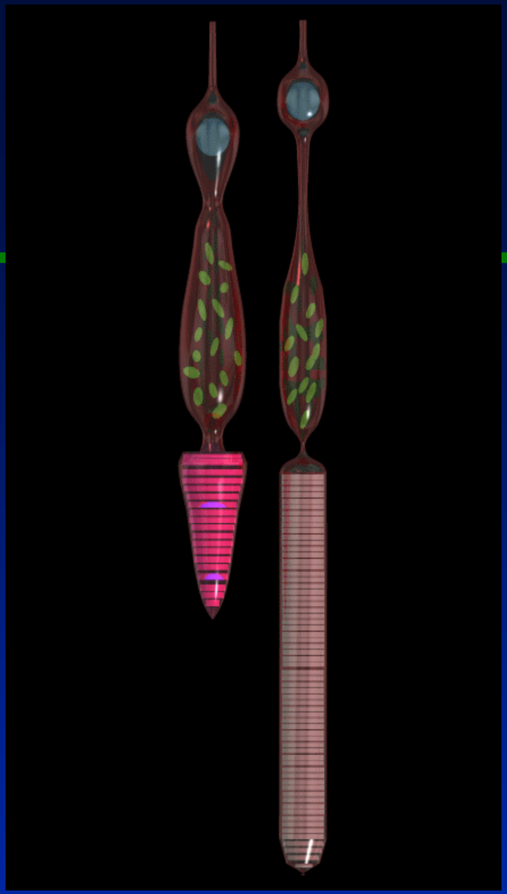


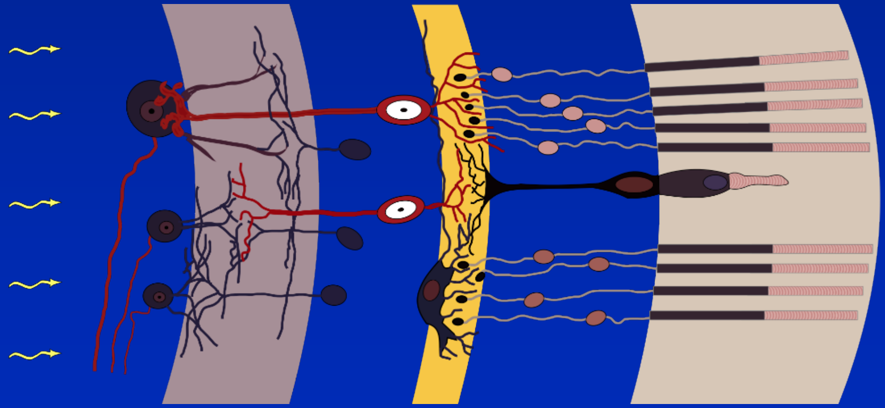


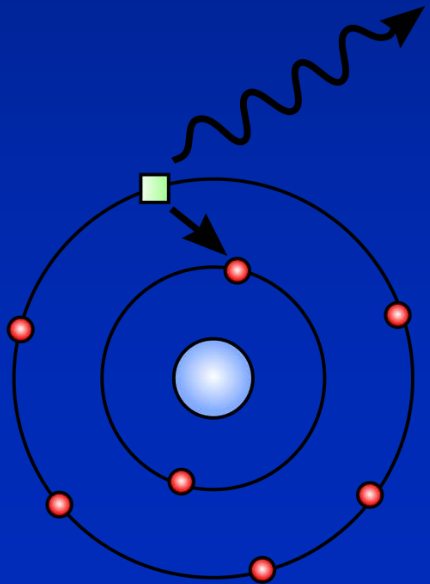
Light as Photons

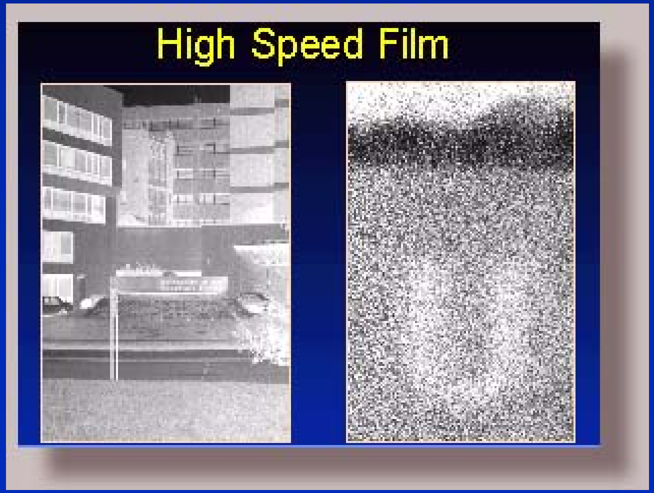




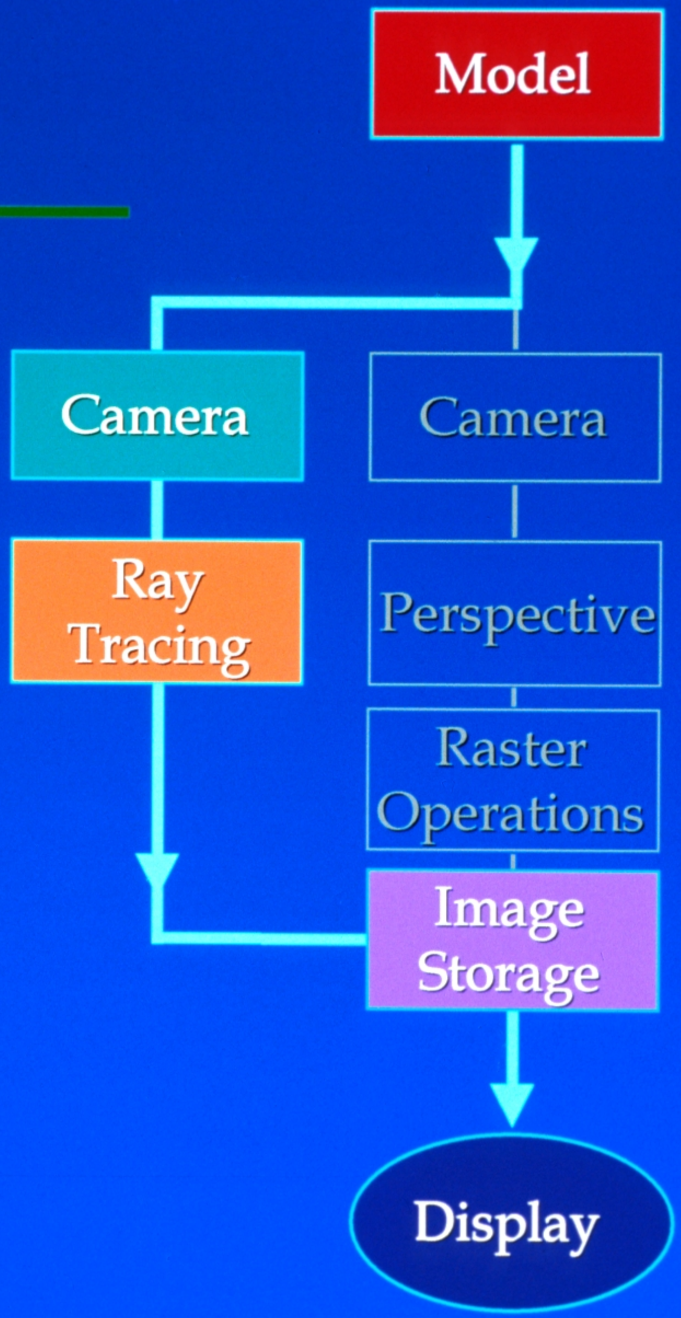






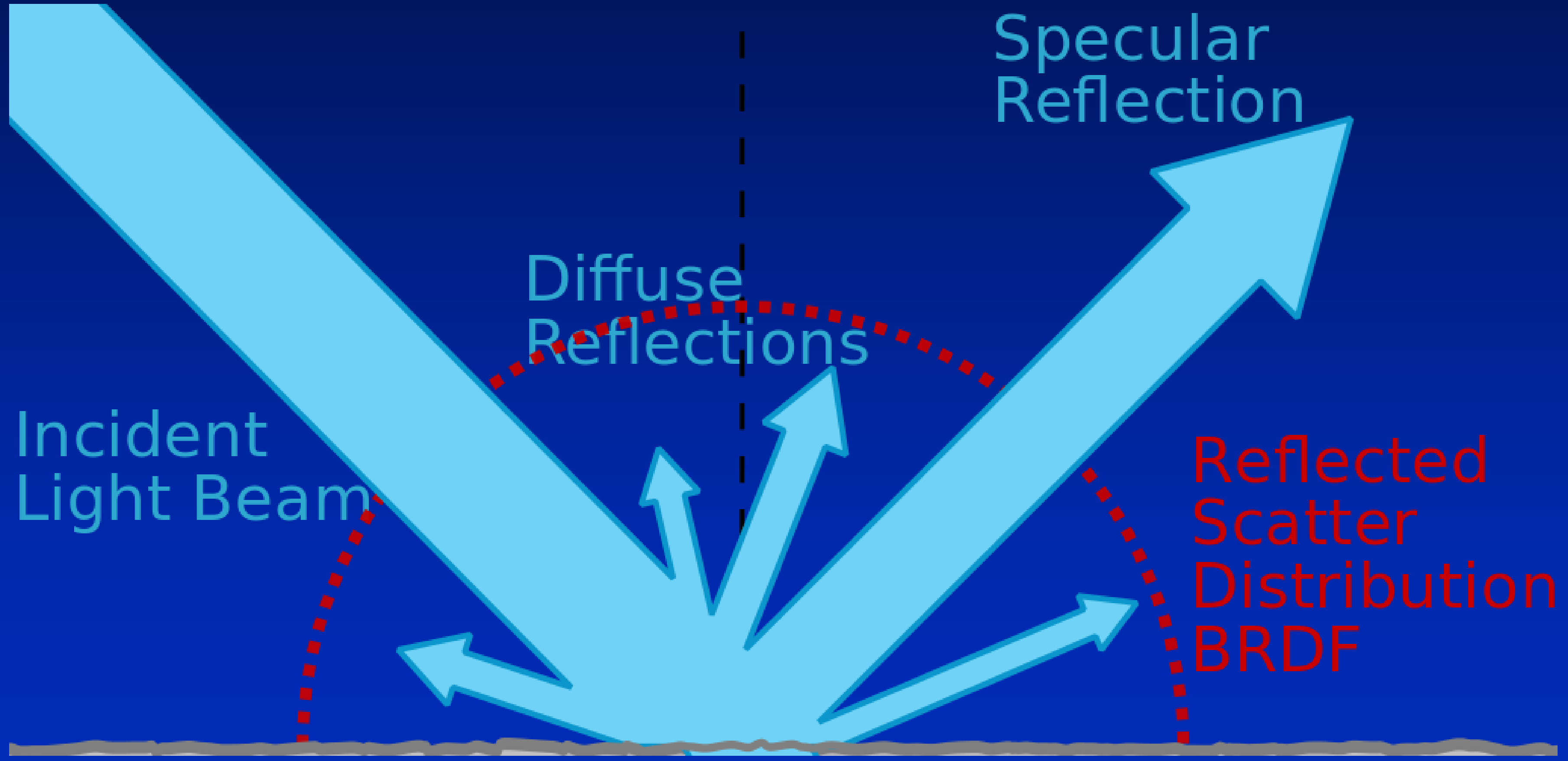


Ray Tracing

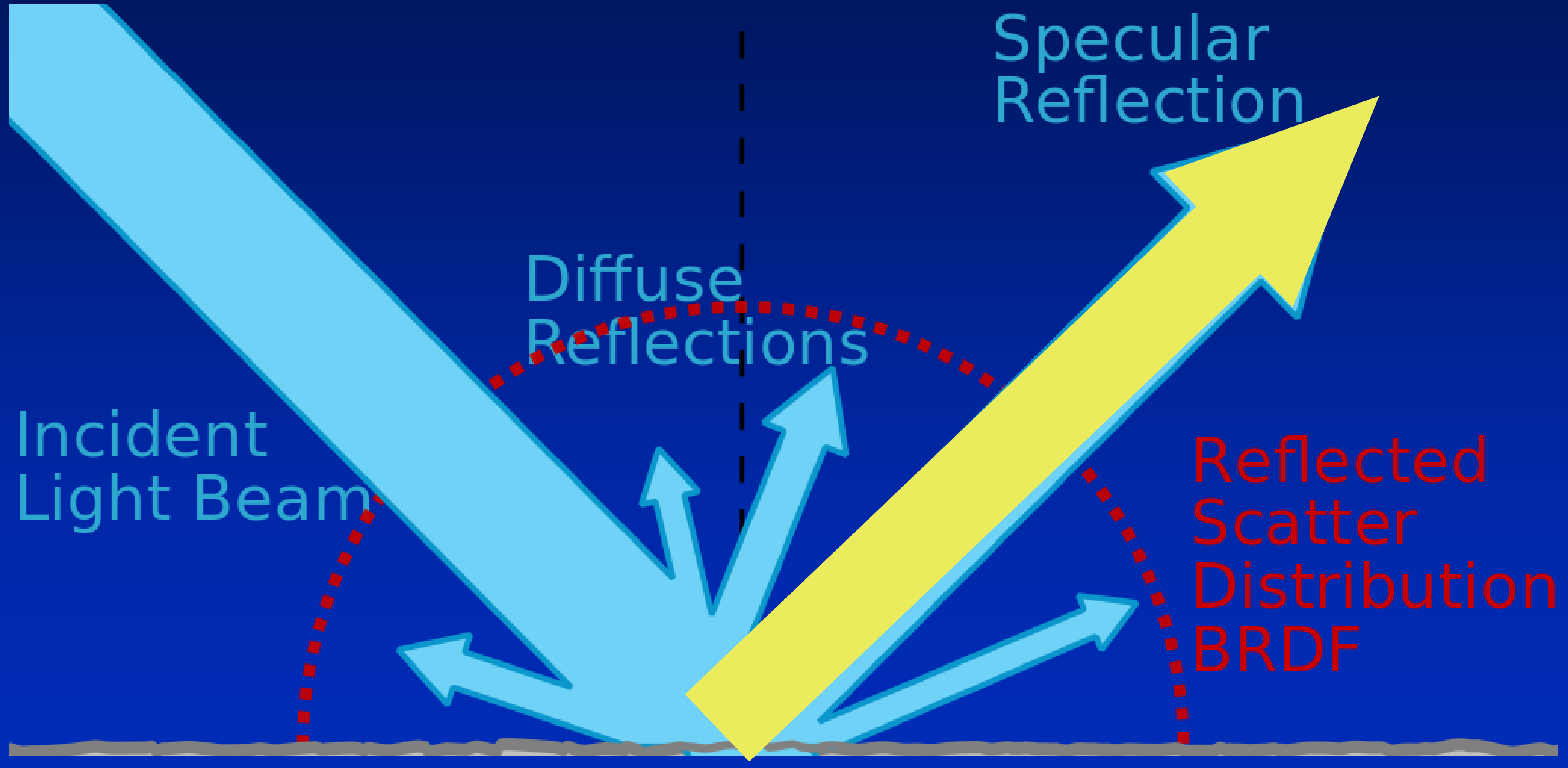


Model
Camera
Ray Tracing
Camera
Perspective
Raster Operations
Image Storage
Display

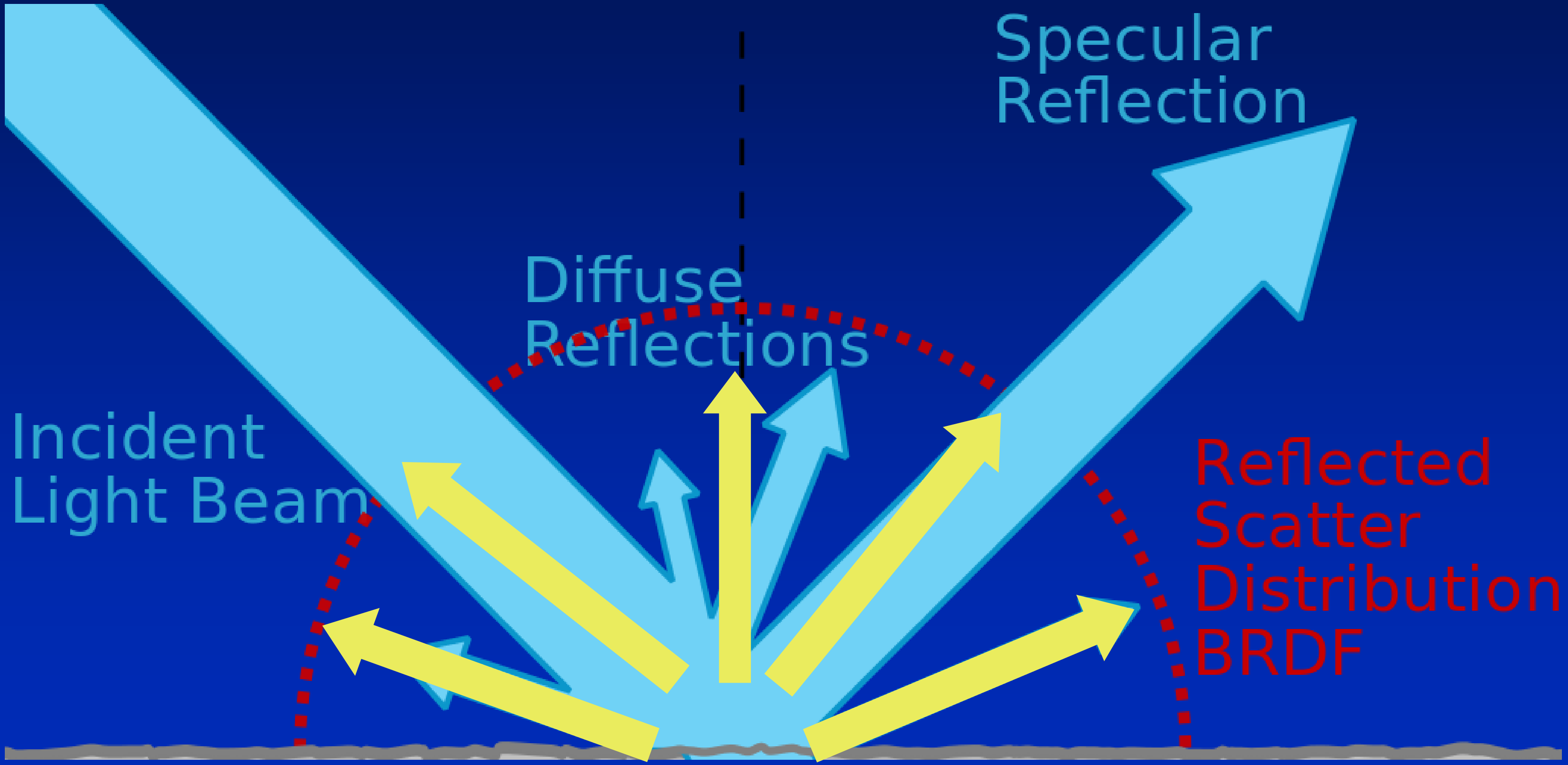
Surface Reflectance



Ray Tracing


Specular
Reflection
Diffuse
Reflections
Incident
Light Beam
Reflected
Scatter
Distribution
BRDF

Radiosity

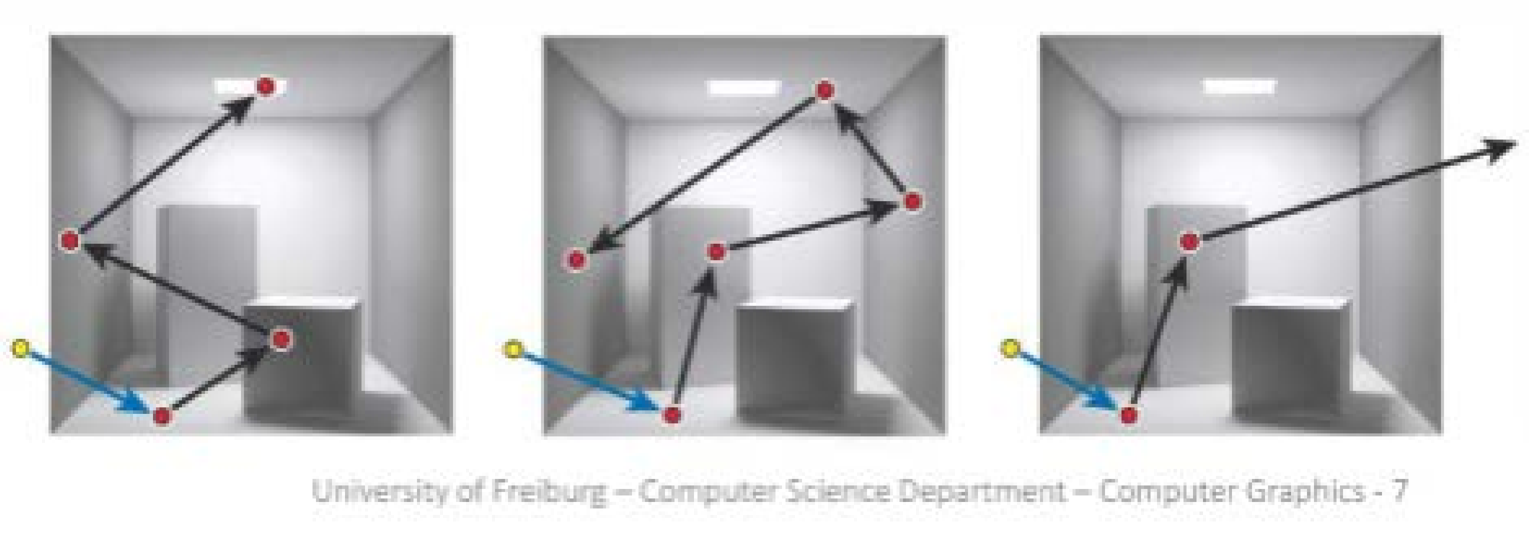

Specular
Reflection
Diffuse
Reflections
Incident
Light Beam
Reflected
Scatter
Distribution
BRDF

Path Tracing

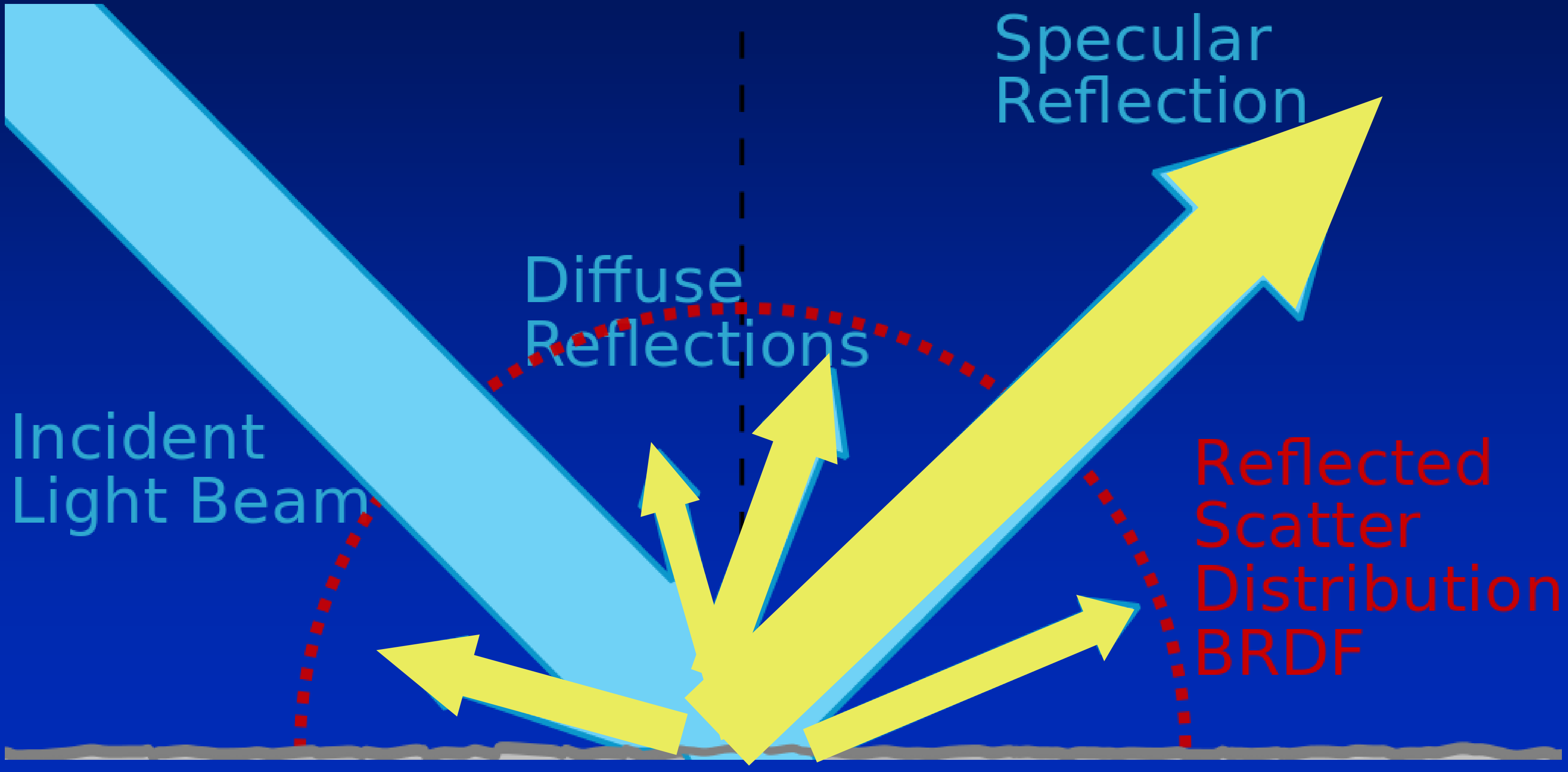
- Path Tracing is similar to ray tracing except that many rays are sent for each pixel.
- Rays are sent out on a probabilistic basis depending on the reflectance (transmittance) distributions of each surface that is struck.
- Computations can be accelerated by using “importance sampling”, where the ray directions are dependent on the magnitude of the potential effects.

Path Tracing

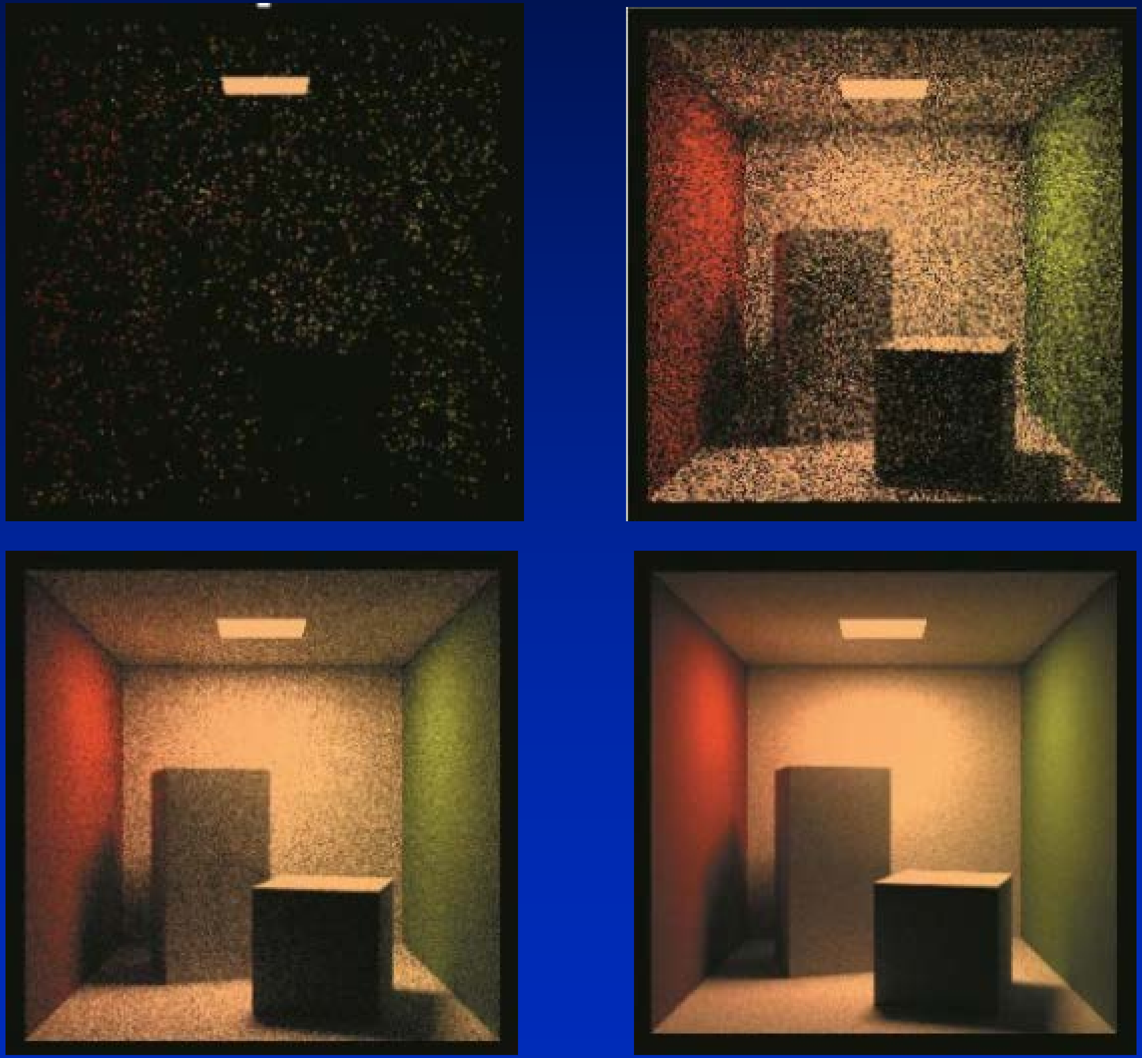
- Rays are cast to estimate the transported radiance.
- Recursion stops if
  - A light source is hit
  - A maximum depth/minimum radiance is reached
  - The ray leaves the scene/hits the background



Path Tracing



Path Tracing



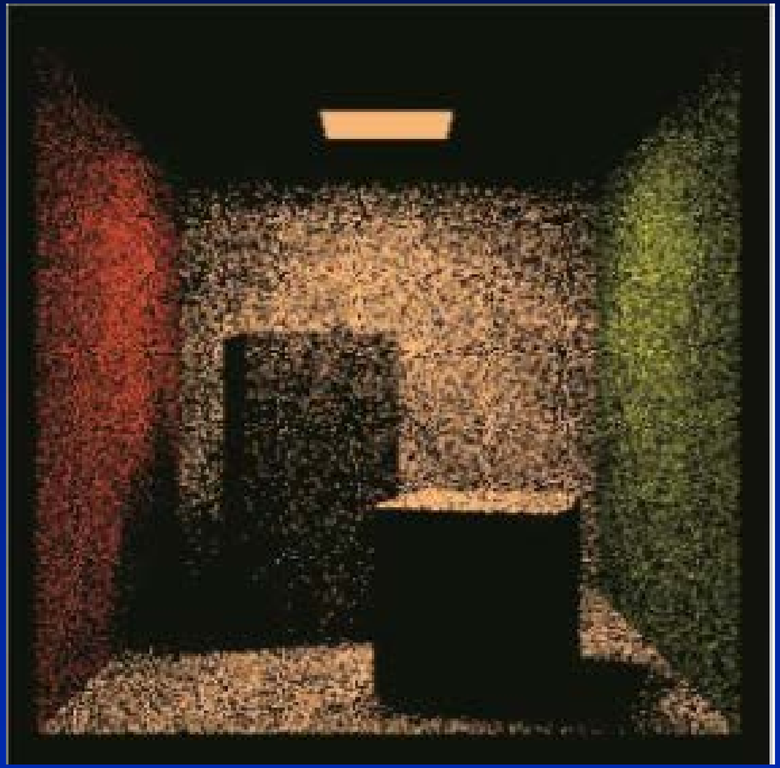
1 sample/pixel

100 samples/pixel

1,000 samples/pixel

10,000 samples/pixel

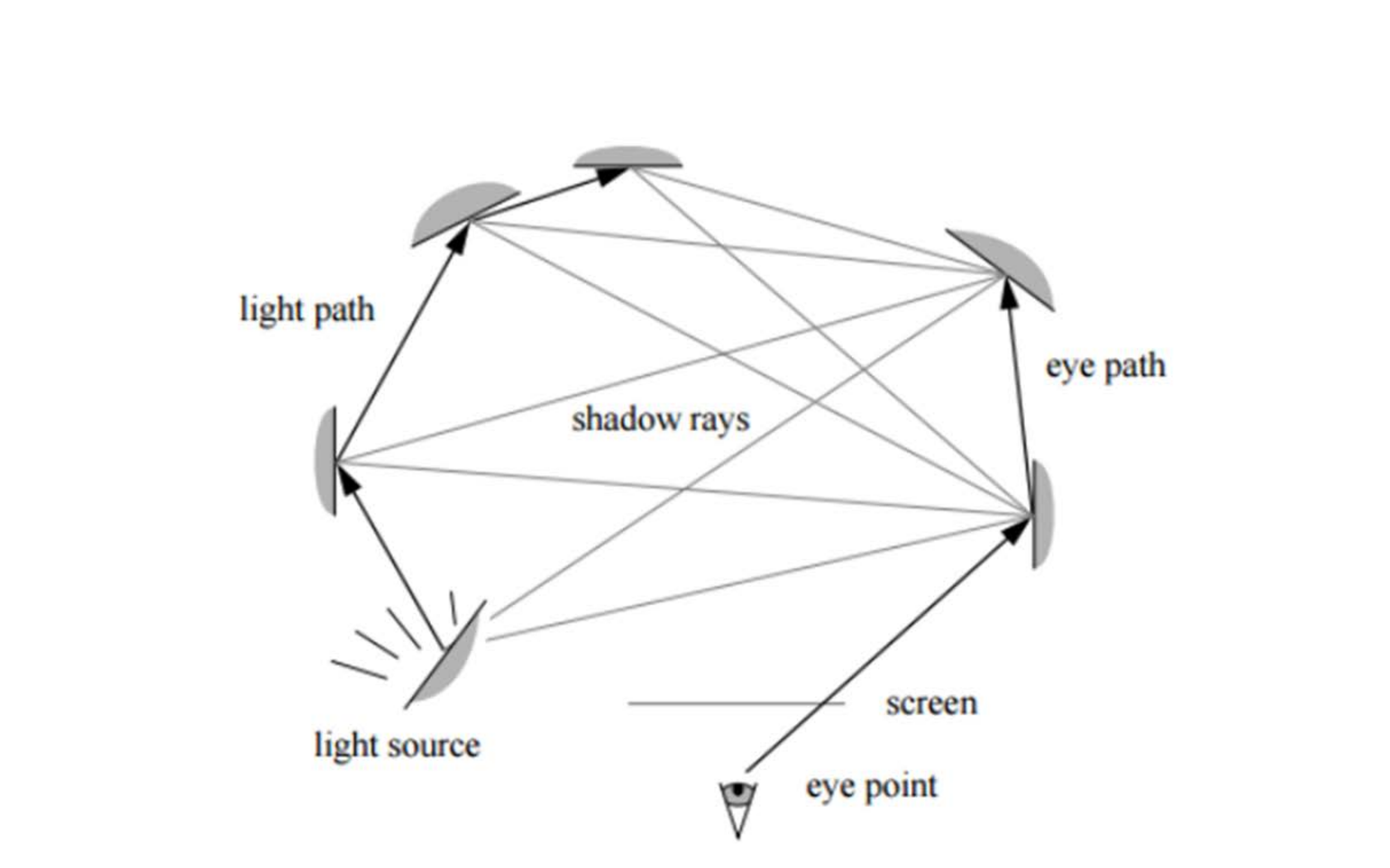
Path Tracing



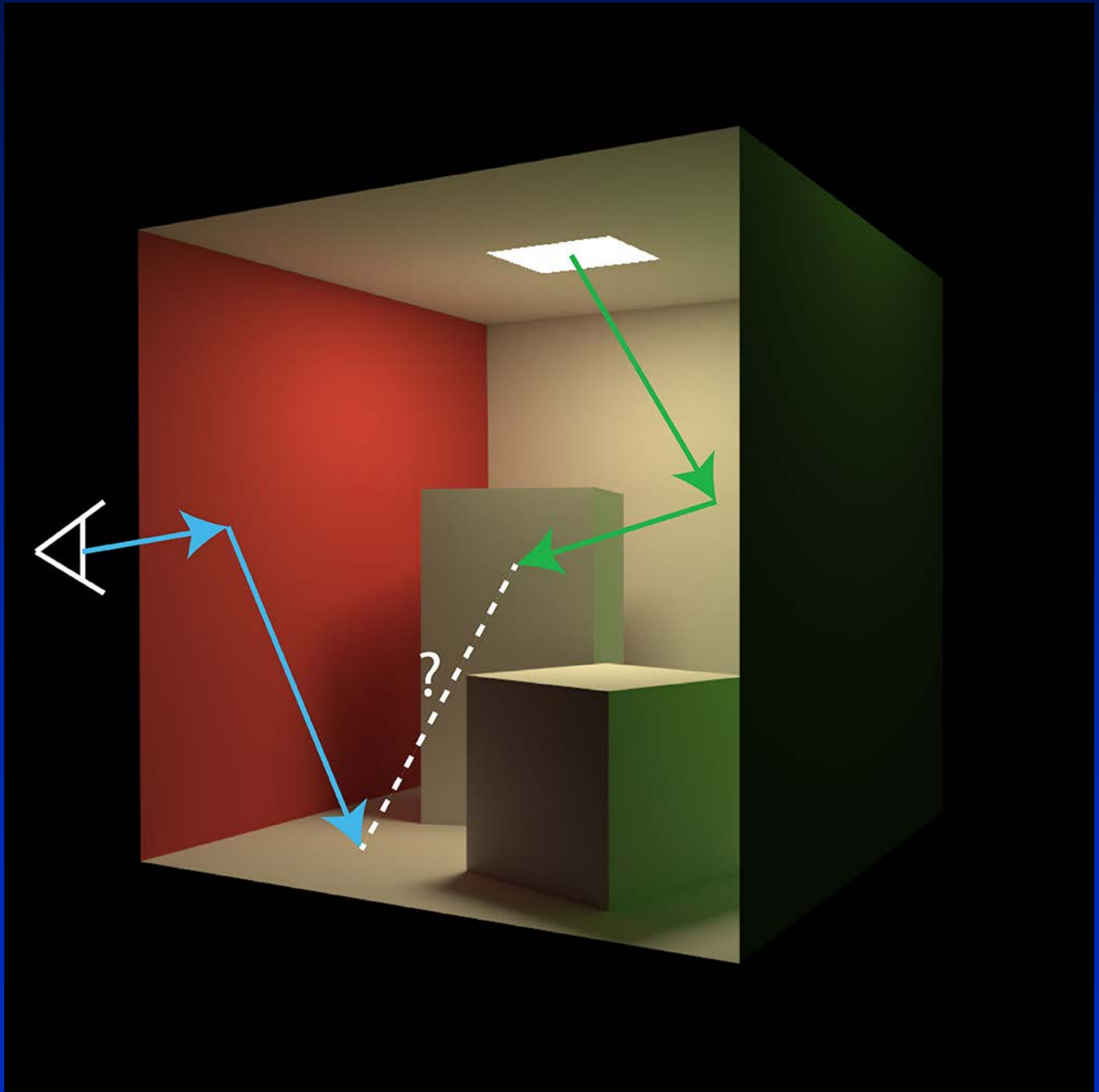
1 sample/light source
100 samples/pixel

100 samples/light source
100 samples/pixel

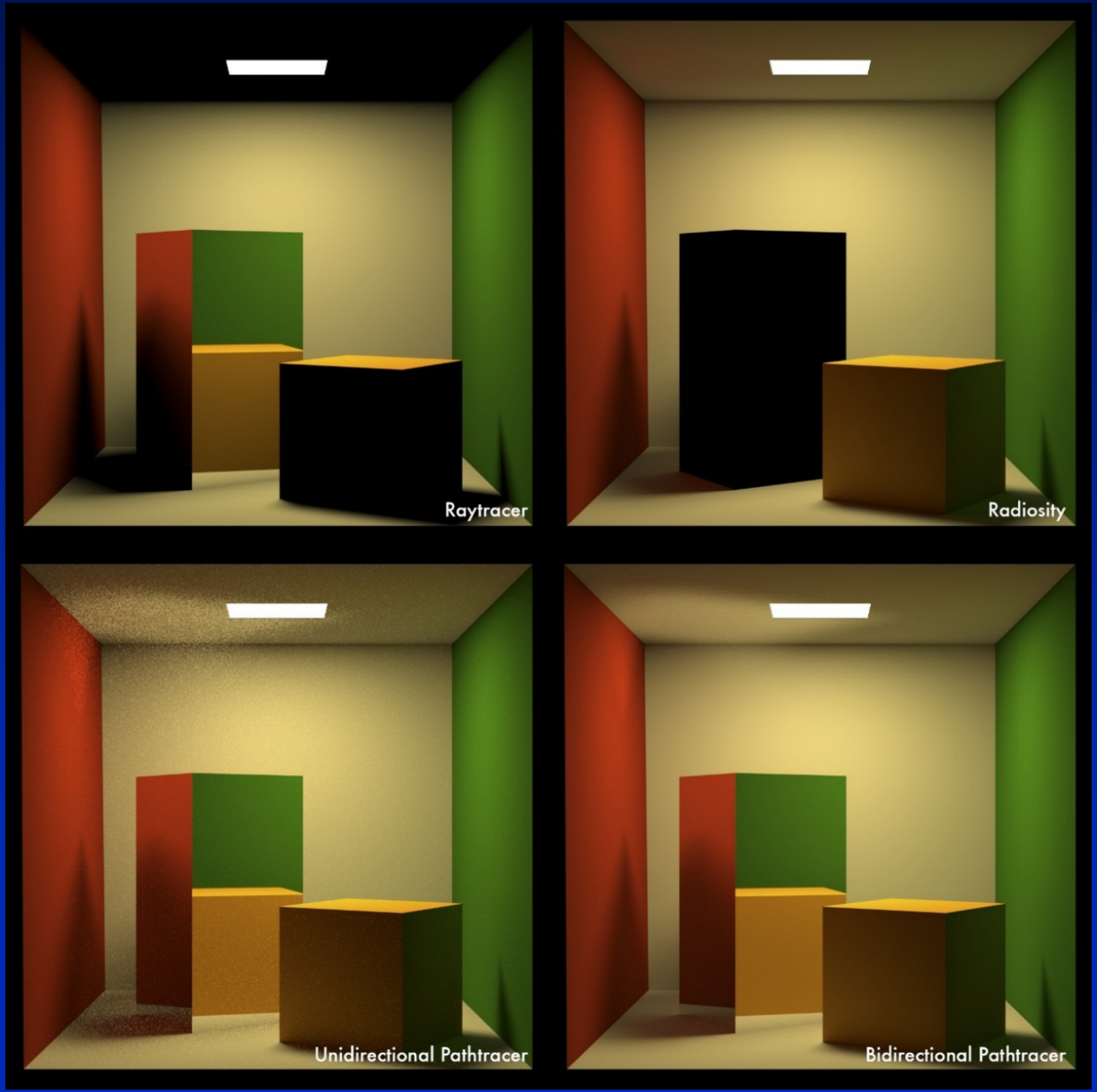
Bi-directional Path Tracing



Bi-Directional Path Tracing



Bi-directional Path Tracing



Graphics Pipeline Hardware

“Moore's Law is for wimps.”

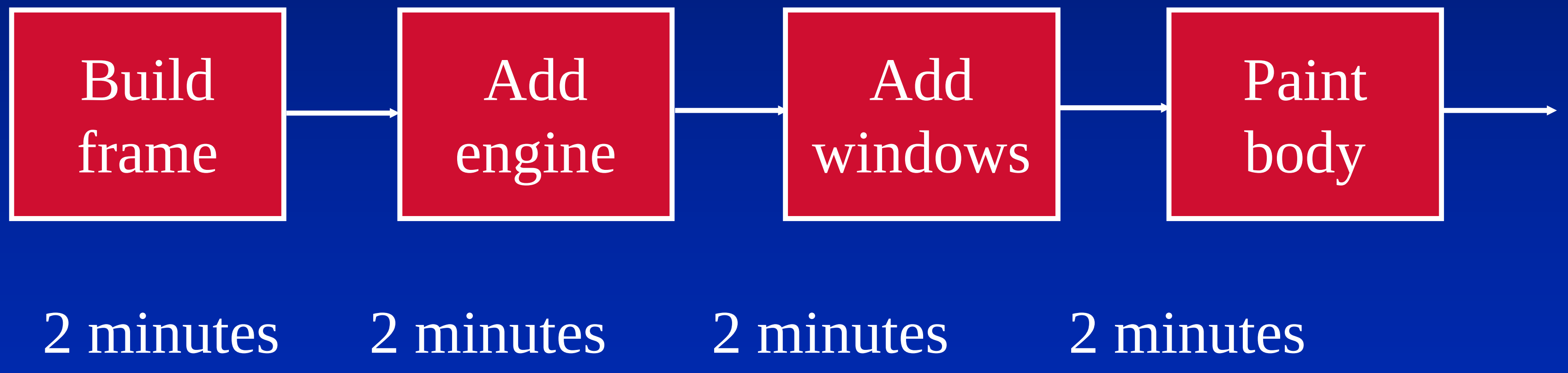
Why a Pipeline?

A pipeline allows multiple processes to occur in parallel.

- Example: 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.

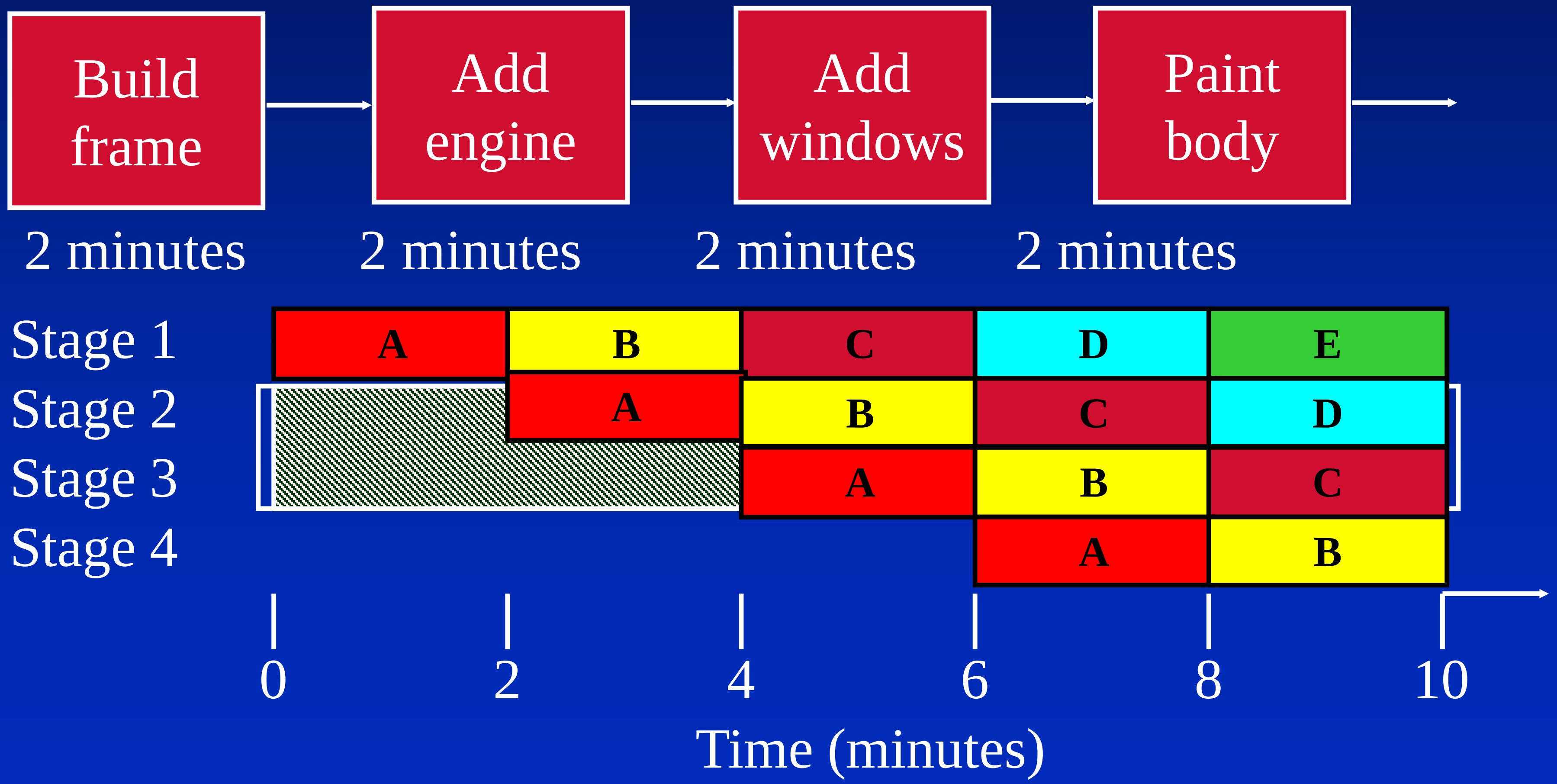
Example: Automobile Pipeline

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

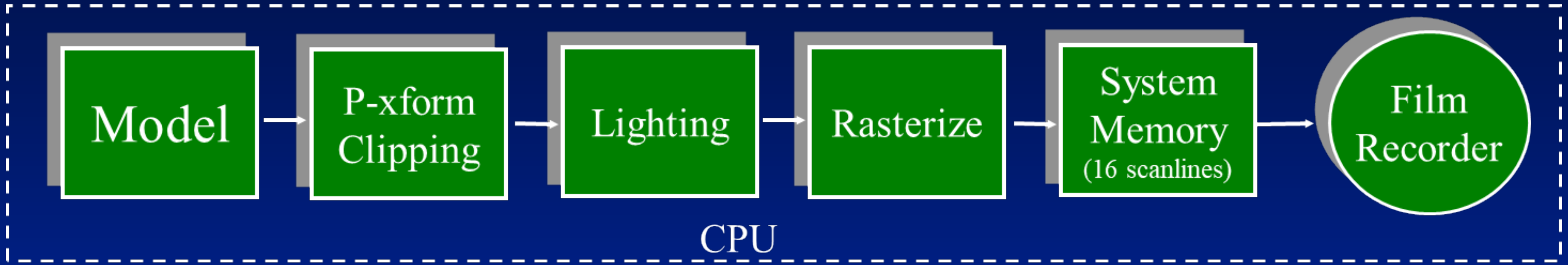


Example: Automobile Pipeline

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

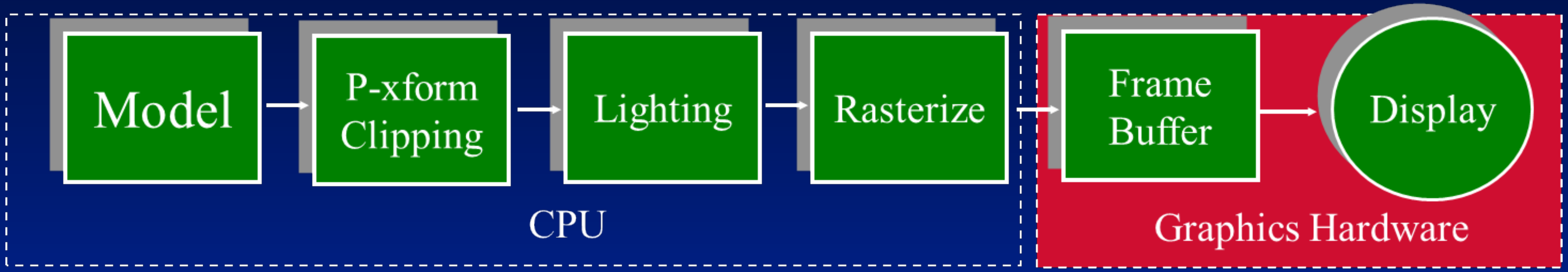


Graphics Hardware circa 1970



- System used to generate Phong goblet

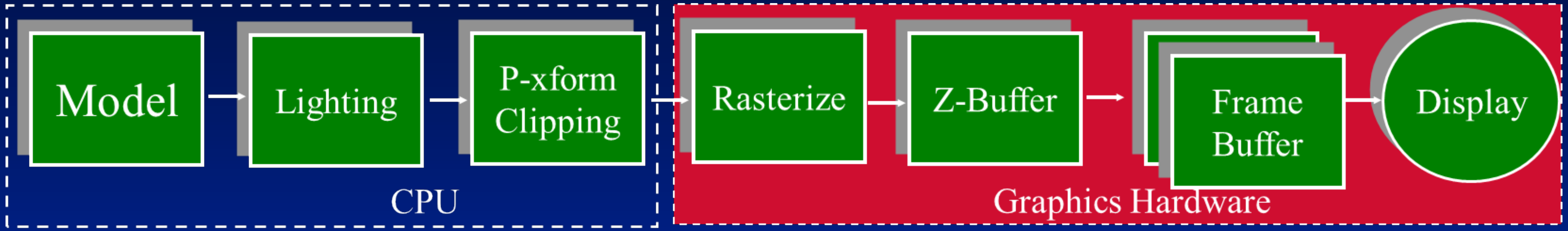
Graphics Hardware circa 1980



Cost of Memory was Prohibitive

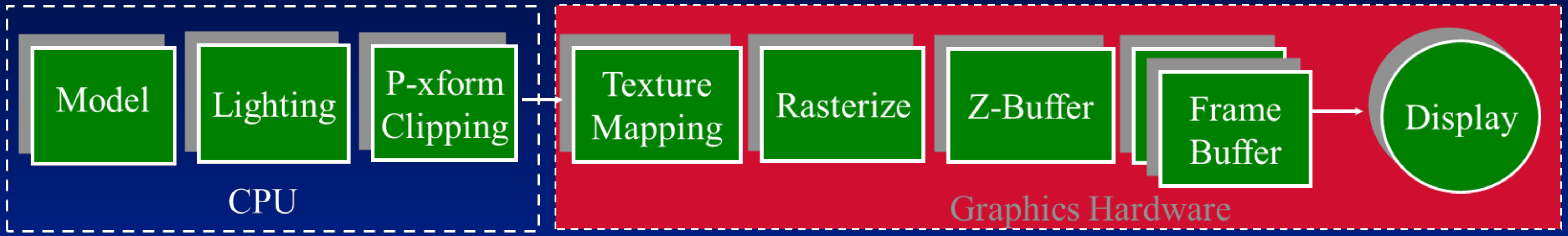
- 512x480x8 bit frame buffer cost $80,000!
- No z-buffer (at 24 or 32 bits/pixel, it requires even more memory than FB)
- Only single frame buffer
- All work done in CPU until frame buffer (slow!)

Graphics Hardware circa 1986



- Added Z-Buffer
- Added Double Frame Buffer
- Rasterization and visible surface computations performed in hardware

Graphics Hardware circa 1999

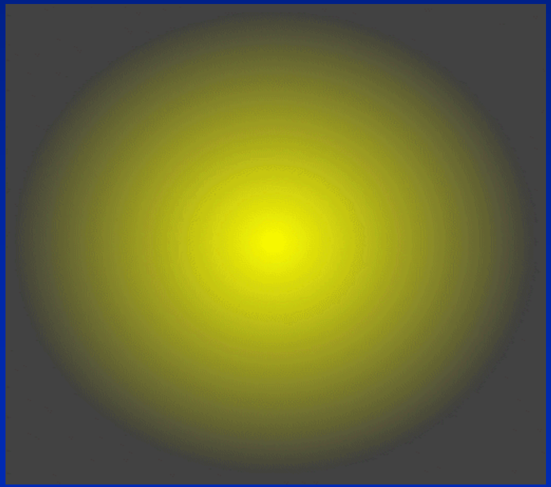


- Addition of texture mapping units
- With texturing, high resolution detail is possible with relatively simple geometry

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.

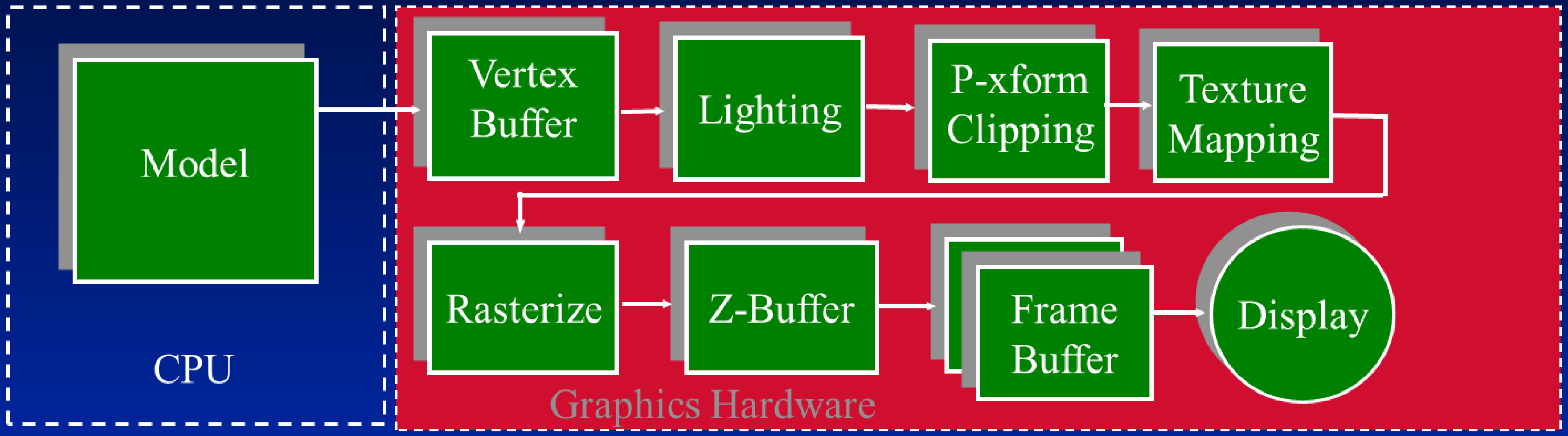
Reflection Example - Castle

Agata & Andrzej Wojaczek, Advanced Graphics Applications Inc.

Putting it all together

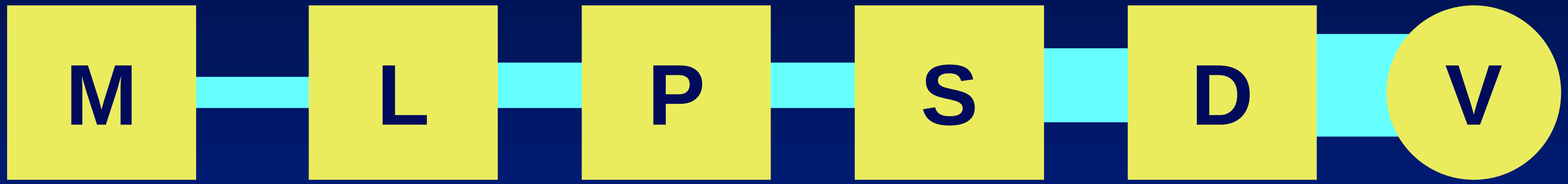
Gloss textures on pear, shadows on curved surfaces, reflections dropping off with depth from table.

Graphics Hardware 2000



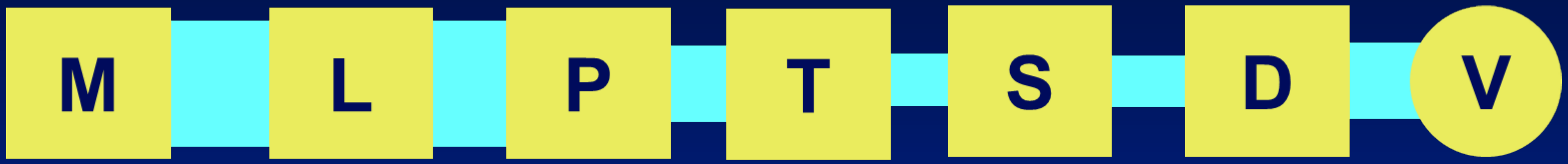
- Vertex buffer (model data) added to reduce bandwidth requirements between CPU and graphics board

Graphics Pipeline 1980s



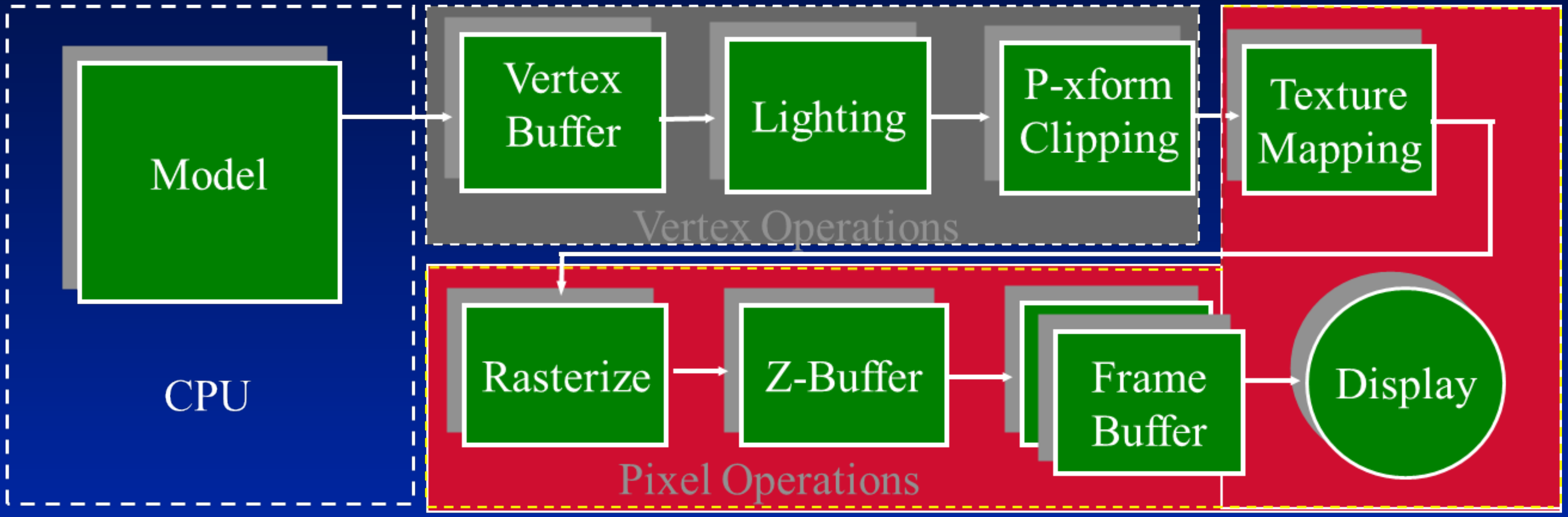
M — Model
L — Lighting
P — Perspective/Clipping
S — Scan Conversion/Z-buffer
D — Display Storage
V — Video

Graphics Pipeline 2000 +



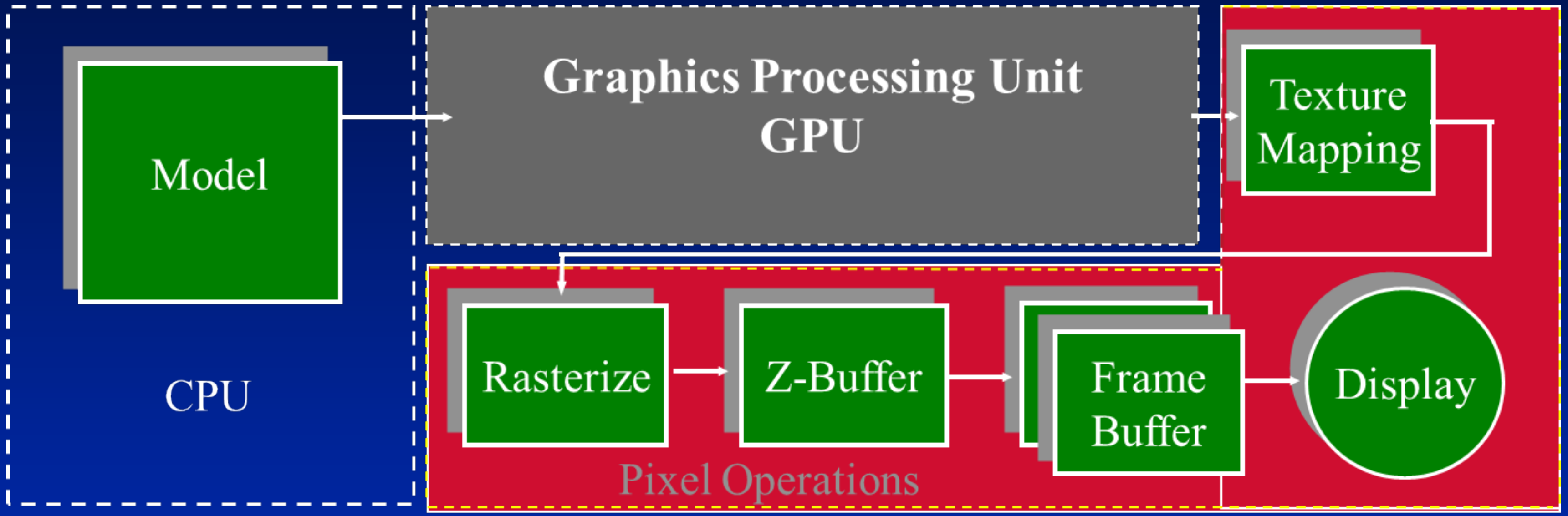
M — Model
L — Lighting
P — Perspective/Clipping
T — Texturing
S — Scan Conversion/Z-buffer
D — Display Storage
V — Video

Graphics Hardware 2003

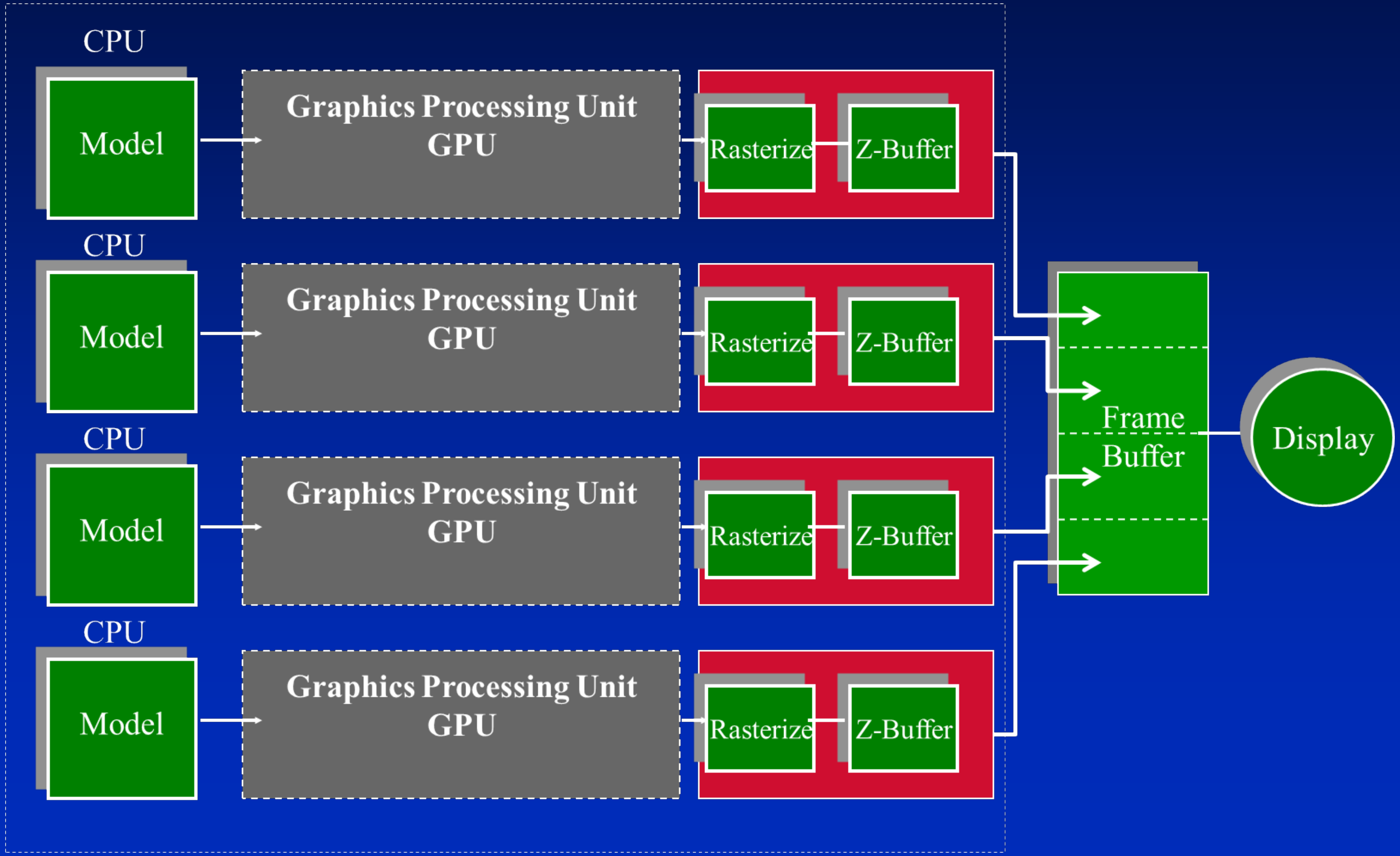


- Early GPU’s performed lighting and clipping operations on locally stored model

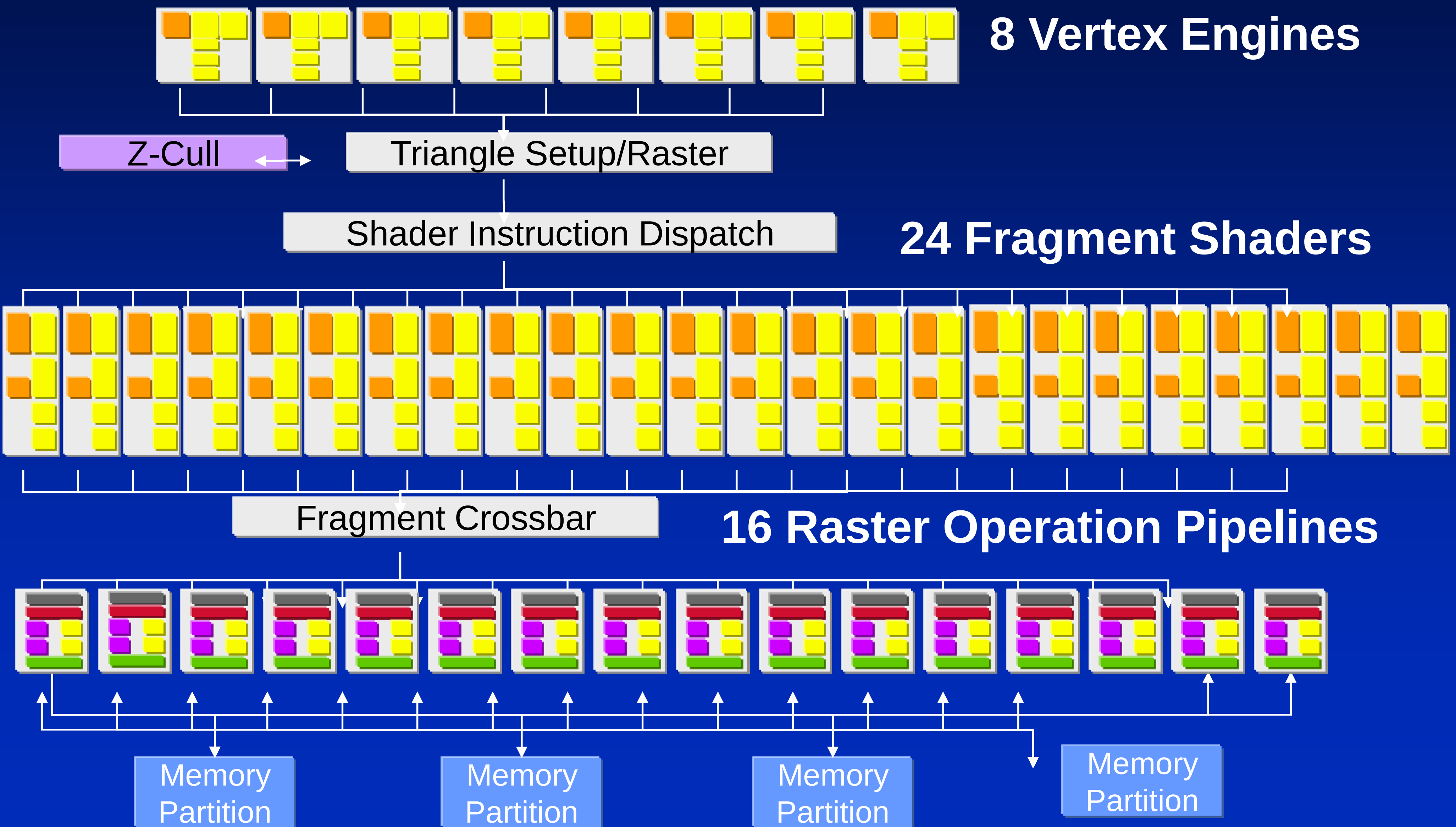
Graphics Hardware 2003 +



Graphics Hardware 2009



GeForce 7800 GTX Parallelism 2009



nVidia's new Kepler Chip 2012

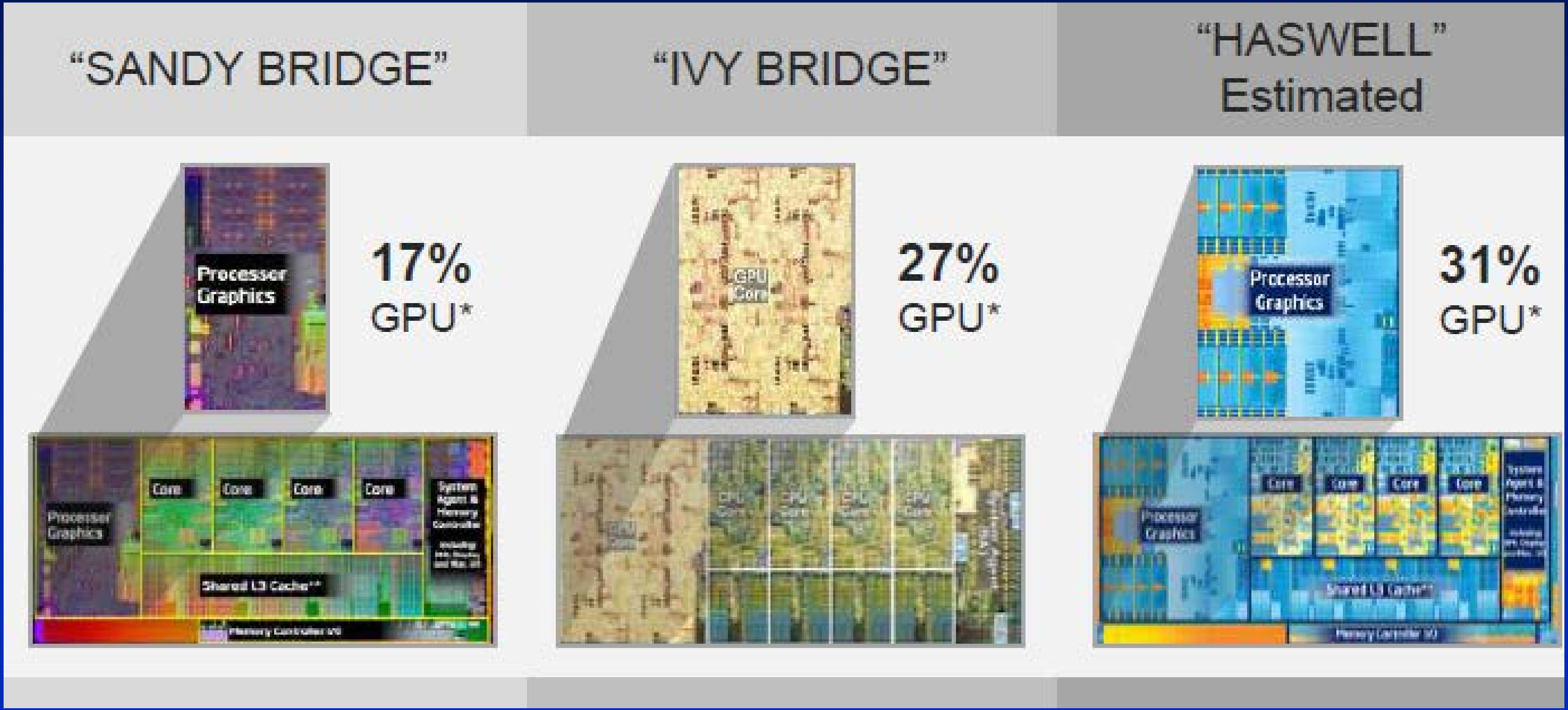
Moore’s Law – GPU Transistor Counts

Processor	Transistor count	Date of introduction	Manufacturer	Process	Area
R520	321,000,000	2005	AMD	90 nm	288 mm^2
R580	384,000,000	2006	AMD	90 nm	352 mm^2
G80	681,000,000	2006	NVIDIA	90 nm	480 mm^2
R600 Pele	700,000,000	2007	AMD	80 nm	420 mm^2
G92	754,000,000	2007	NVIDIA	65 nm	324 mm^2
RV790XT Spartan	959,000,000	2008	AMD	55 nm	282 mm^2
GT200 Tesla	1,400,000,000	2008	NVIDIA	65 nm	576 mm^2
Cypress RV870	2,154,000,000	2009	AMD	40 nm	334 mm^2
Cayman RV970	2,640,000,000	2010	AMD	40 nm	389 mm^2
GF100 Fermi	3,200,000,000	Mar 2010	NVIDIA	40 nm	526 mm^2
GF110 Fermi	3,000,000,000	Nov 2010	NVIDIA	40 nm	520 mm^2
GK104 Kepler	3,540,000,000	2012	NVIDIA	28 nm	294 mm^2
Tahiti RV1070	4,312,711,873	2011	AMD	28 nm	365 mm^2
GK110 Kepler	7,080,000,000	2012	NVIDIA	28 nm	561 mm^2
RV1090 Hawaii	6,300,000,000	2013	AMD	28 nm	438 mm^2
GM204 Maxwell	5,200,000,000	2014	NVIDIA	28 nm	398 mm^2
GM200 Maxwell	8,100,000,000	2015	NVIDIA	28 nm	601 mm^2
Fiji	8,900,000,000	2015	AMD	28 nm	596 mm^2
GP102 Pascal	12,000,000,000	2016	Nvidia	12 nm	471 mm^2
Vega 10	12,500,000,000	2017	AMD	14 nm	484 mm^2
GP100 Pascal	15,300,000,000	2016	Nvidia	16 nm	610 mm^2
GV100 Volta	21,100,000,000	2017	Nvidia	12 nm	815 mm^2

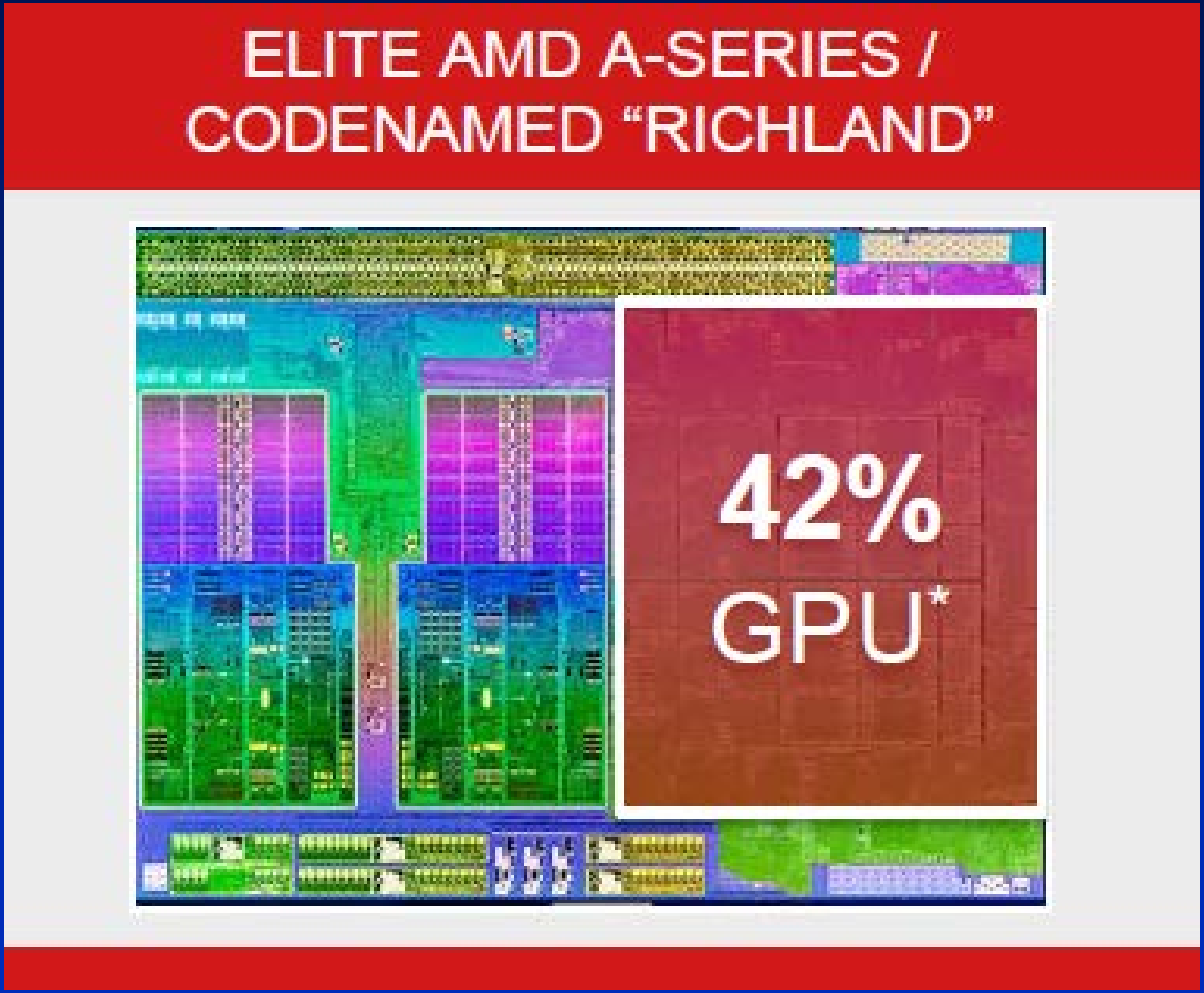
http://en.wikipedia.org/wiki/Transistor_count

Artificial Intelligence Systems Nvidia DGX

Intel – Integrated Graphics 2013

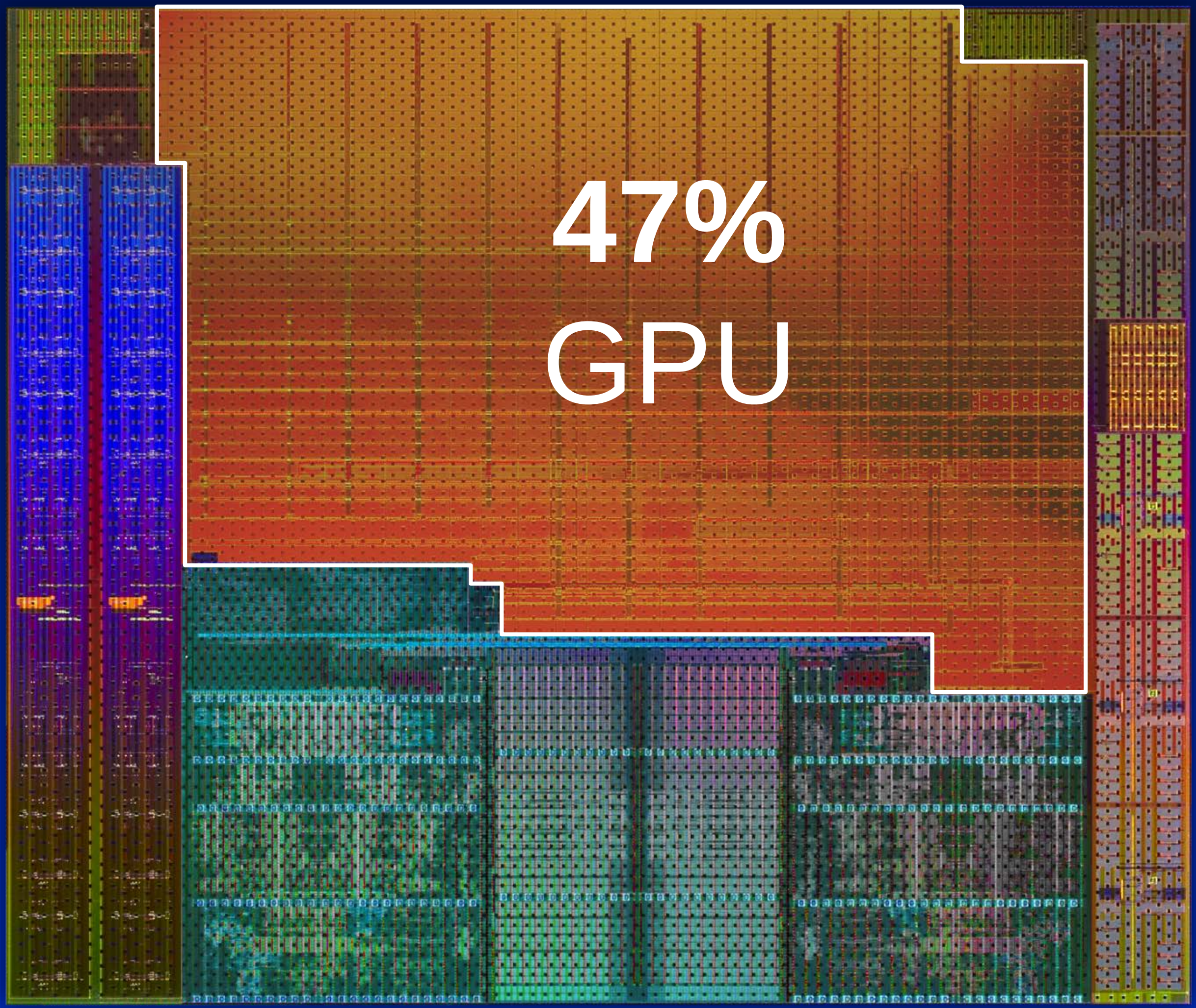


AMD – Integrated Graphics 2013



AMD – Integrated Graphics 2014

- “Kaveri”
- 28 nm
- 47% GPU



End. . .