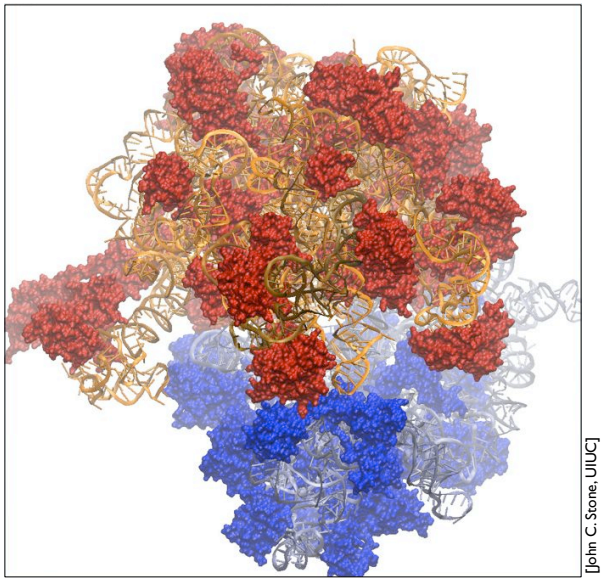
Cornell University CS 569: Interactive Computer Graphics

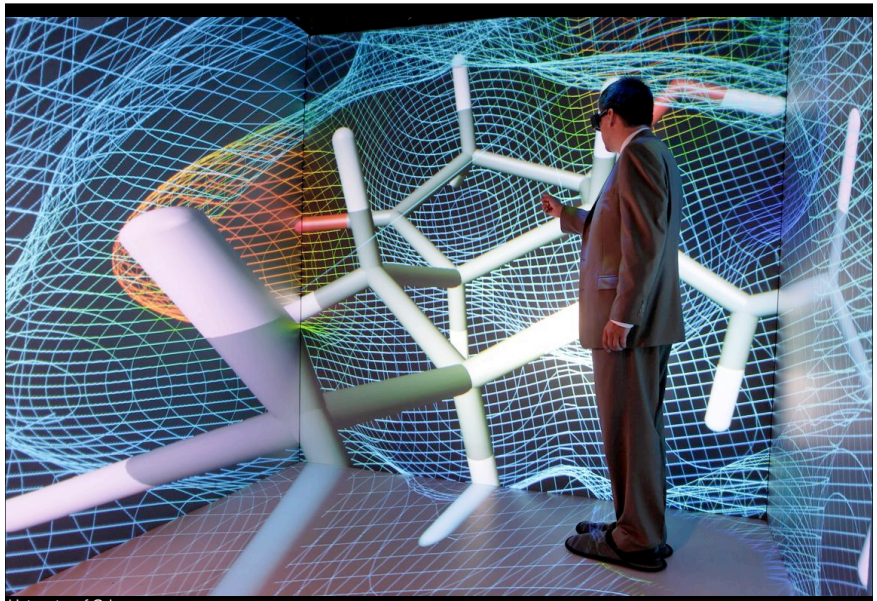
Introduction

Lecture 1



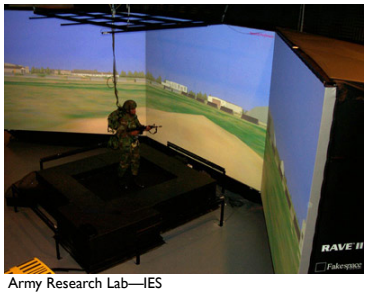
[John C. Stone, UIUC]

NASA

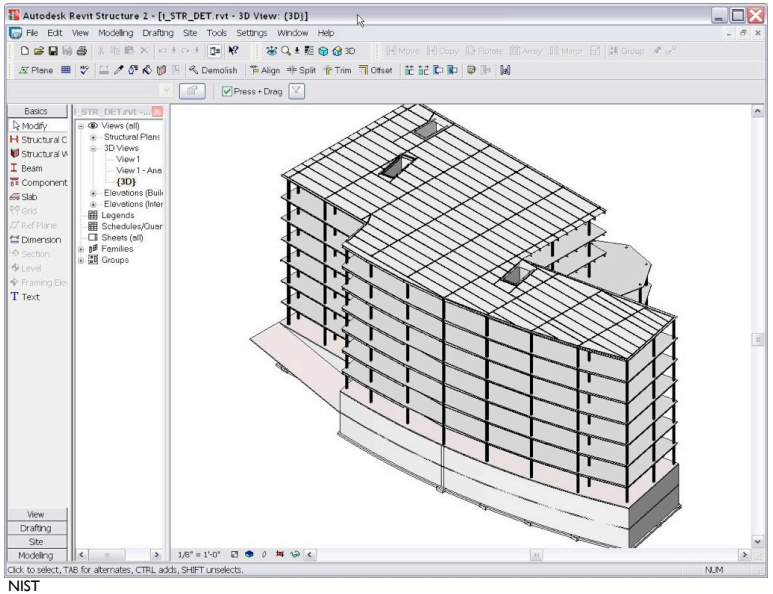


University of Calgary

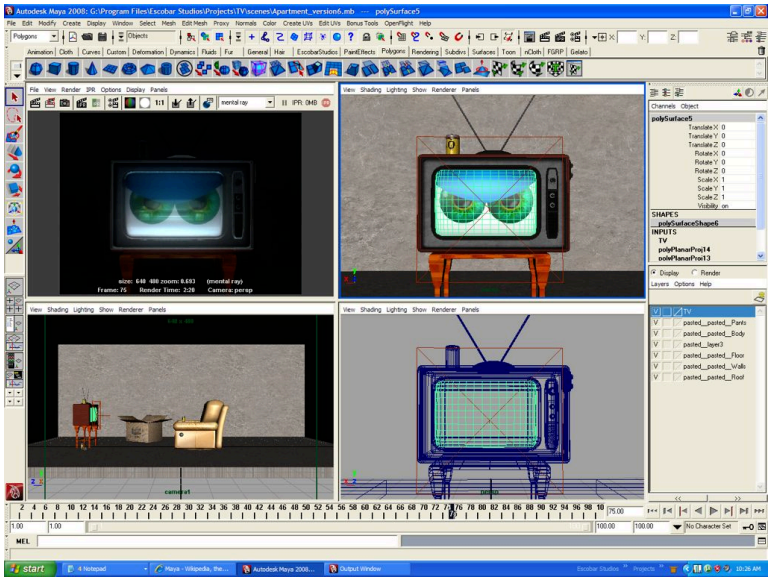
NASA Ames



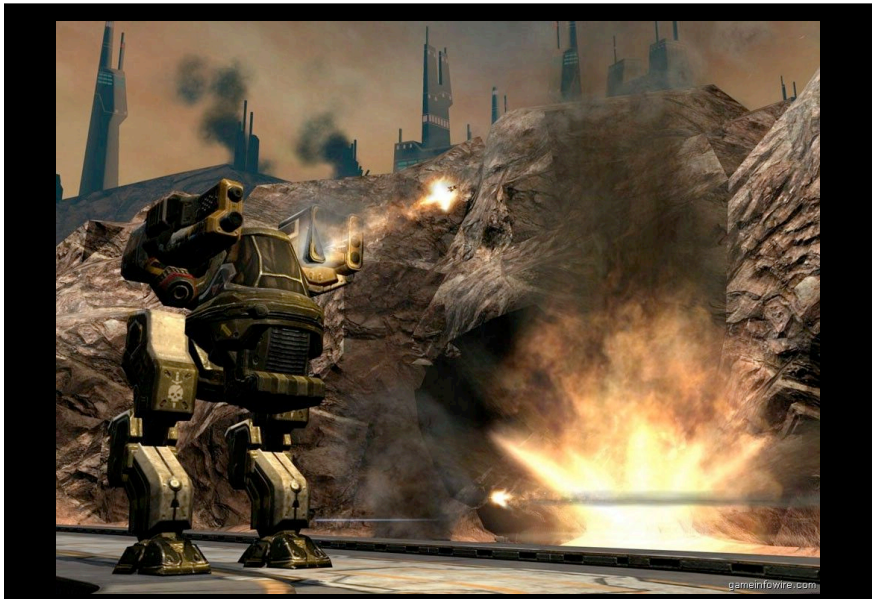
Army Research Lab—IES



NIST



Autodesk Maya (Wikimedia—Aaron1a12)



ID Software—*Quake 4* (2005)

TimeGate Studios—*F.E.A.R. Extraction Point* (2006)

Valve—*Half Life 2* (2006)

Ubisoft—Assassin's Creed (2007)

Valve—*Portal* (2007)

How To Draw a Triangle, c. 1985

- Transform vertices to screen coordinates
- Find all the pixels covered by the triangle
- Fill all the pixels with the triangle's color

How To Draw a Triangle, c. 1988

- Perform lighting calculations to find vertex colors
- Transform vertices to screen coordinates
- Find all the pixels covered by the triangle
- Fill all unoccluded pixels with the interpolated vertex colors and depth

How To Draw a Triangle, c. 1992

- Perform lighting calculations to find vertex colors
- Transform vertices to screen coordinates
- Find all the pixels covered by the triangle
- Look up a texture map value
- Fill all unoccluded pixels with a function of the texture and the interpolated vertex colors, as well as the depth

How To Draw a Triangle, c. 1999

- Perform elaborate lighting calculations to find vertex colors
- Transform vertices to screen coordinates
- Find all the pixels covered by the triangle
- Look up a value from one or more 1D, 2D, or 3D texture maps
- Fill all unoccluded pixels with a complicated, adjustable function of the textures and the interpolated vertex colors, as well as the depth

Pixar's RenderMan

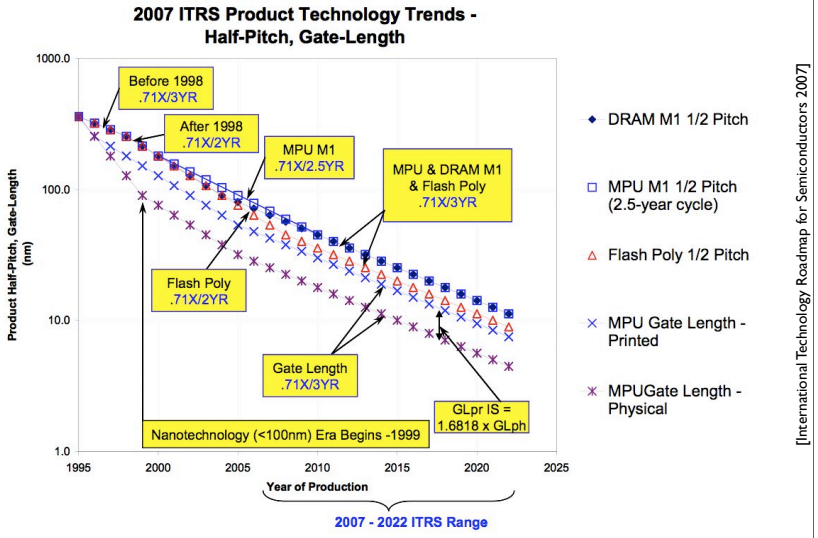
Pixar—Ratatouille (2007)

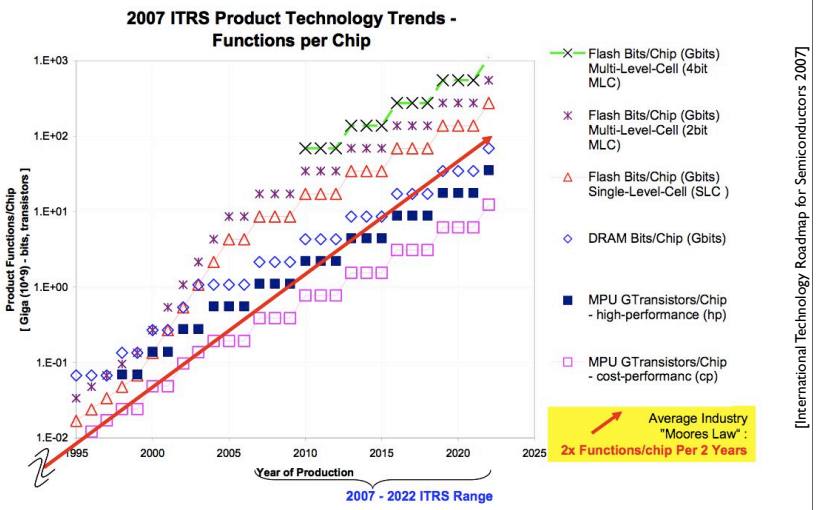
How To Draw a Triangle in 2008

- Execute a vertex program over all the vertices
- Find all the pixels covered by the triangle
- Execute a fragment program over all those pixels
- Fill all unoccluded pixels with the resulting color and depth

Development of Hardware Capabilities

- Workstation era
  - '85–'87: transform and render flat-shaded points, lines, polygons (no *z* buffer)
  - '88–'91: transform, light, and render smooth shaded polygons
  - '92–: transform, light, and render texture-mapped, antialiased polygons
- PC era
  - '95–'98: render texture-mapped polygons
  - '99–'00: transform, light, and render texture-mapped, antialiased polygons
  - '01–'06: execute vertex and fragment shaders over antialiased polygons
  - '07–: execute geometry, vertex, and fragment shaders over antialiased polygons





SGI RealityEngine Architecture (1992)

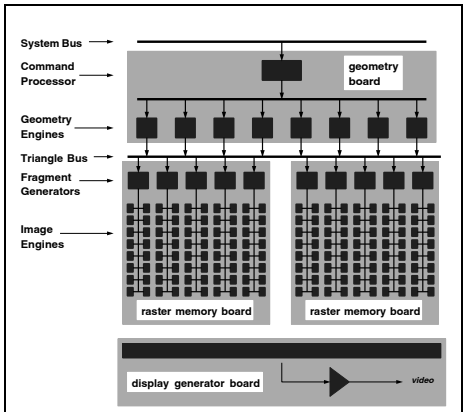
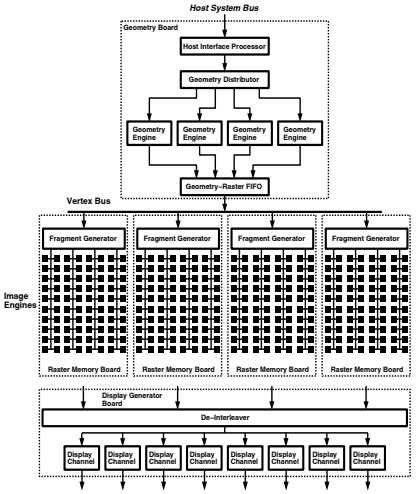
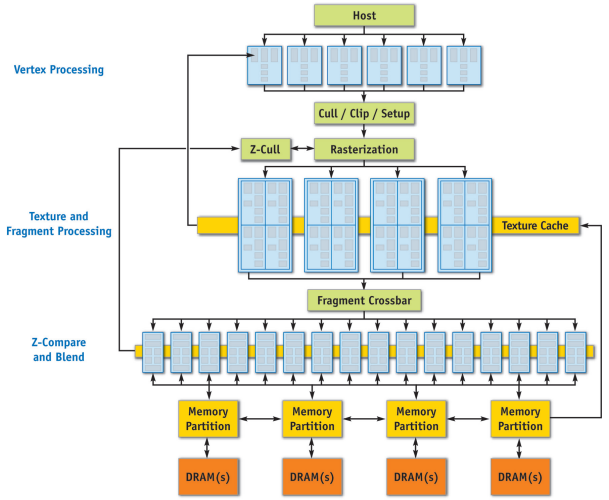


Figure 1. Board-level block diagram of an intermediate configuration with 8 Geometry Engines on the geometry board, 2 raster memory boards, and a display generator board.

SGI InfiniteReality Architecture (1996)

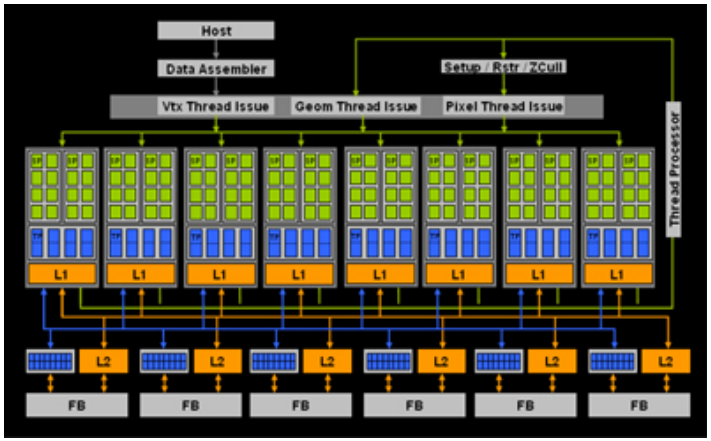


NVIDIA GeForce 6 Architecture (2005)



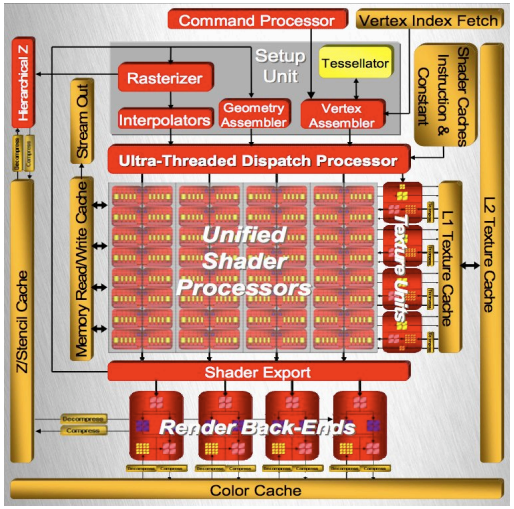
[GPU Gems 2]

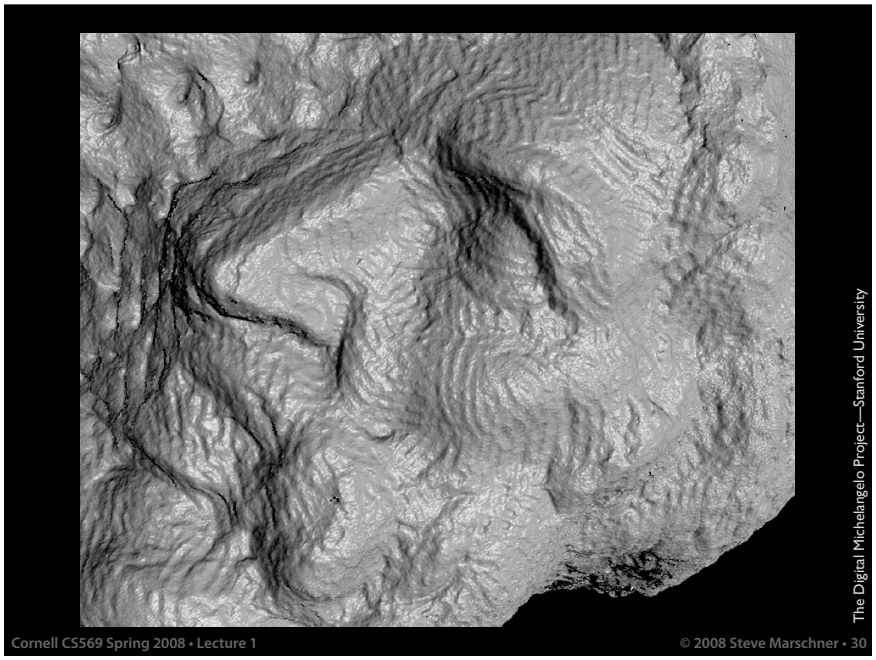
NVIDIA GeForce 8800 Architecture (2007)



[NVIDIA Corporation]

ATI Radeon HD 2900 Architecture (2007)




The Digital Michelangelo Project—Stanford University


The Digital Michelangelo Project—Stanford University


The Digital Michelangelo Project—Stanford University