Computer Graphics (CS 543) Lecture 1 (part 1): Introduction to Computer Graphics

Prof Emmanuel Agu

Computer Science Dept.

Worcester Polytechnic Institute (WPI)

What is Computer Graphics (CG)?

- Computer graphics: algorithms, mathematics, data structures ….. that **computer uses to generate PRETTY PICTURES**
- Techniques (e.g. draw a line, polygon) evolved over years
- Built into programmable libraries

Computer-Generated!
Not a picture!

Photorealistic Vs Real-Time Graphics

Not this Class

Maker's
Mark
WHISKY

This Class

M. BLV
exp sunday
ALL VEHICLES
EXP
WAY
NO TRUCKS
NVIDIA

- **Photo-realistic:** E.g ray tracing slow: may take **days** to render

- **Real Time graphics:** **Milliseconds** to render (30 FPS) But lower image quality

Uses of Computer Graphics

- **Entertainment:** games

Courtesy: Final Fantasy XIV

Courtesy: Super Mario Galaxy 2

Uses of Computer Graphics

- movies, TV, books, magazines

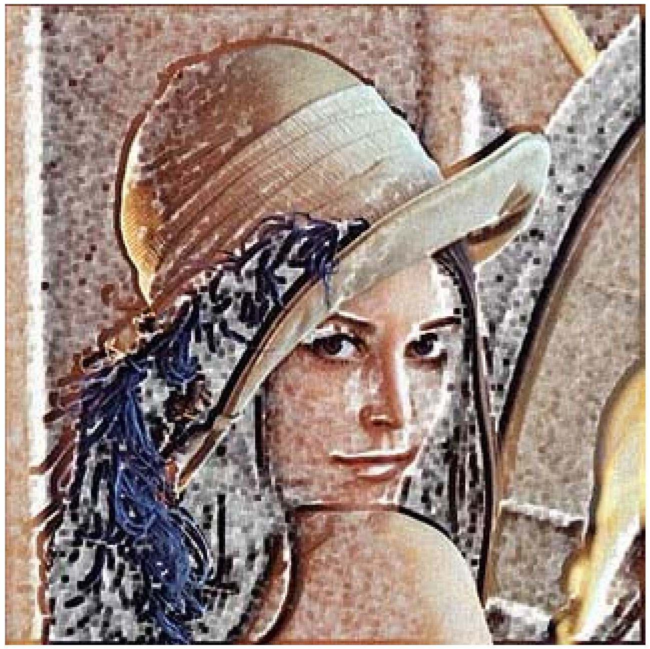
Courtesy: Shrek

Courtesy: Spiderman

Uses of Computer Graphics

- **Image processing:**
  - alter images, remove noise, super-impose images

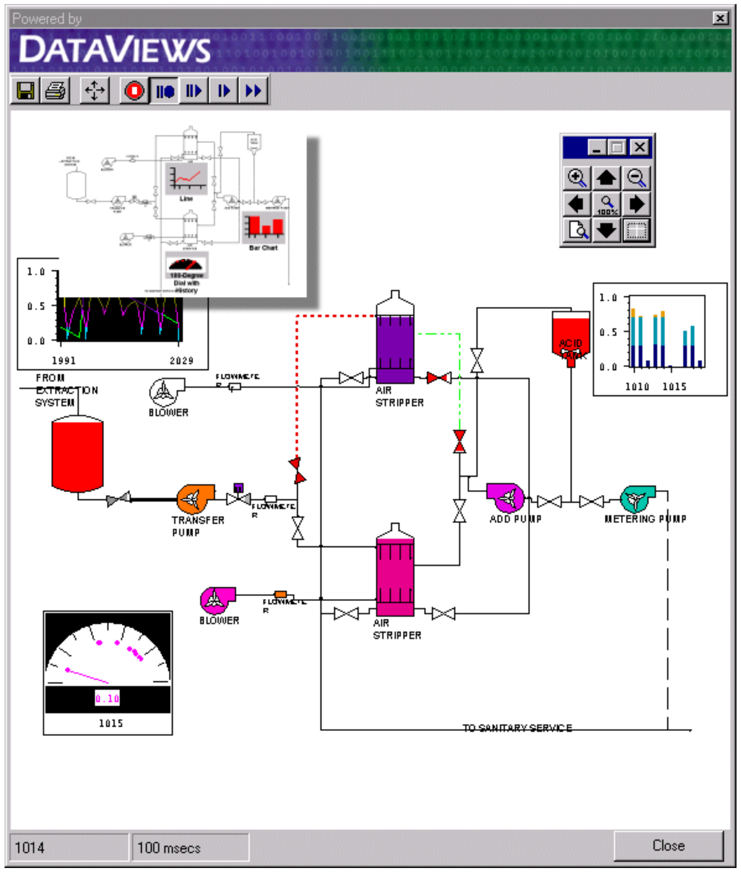
Original Image



Sobel Filter

Uses of Computer Graphics

- **Process monitoring:**
  - Layout of large systems or plants



Courtesy:

Dataviews.de

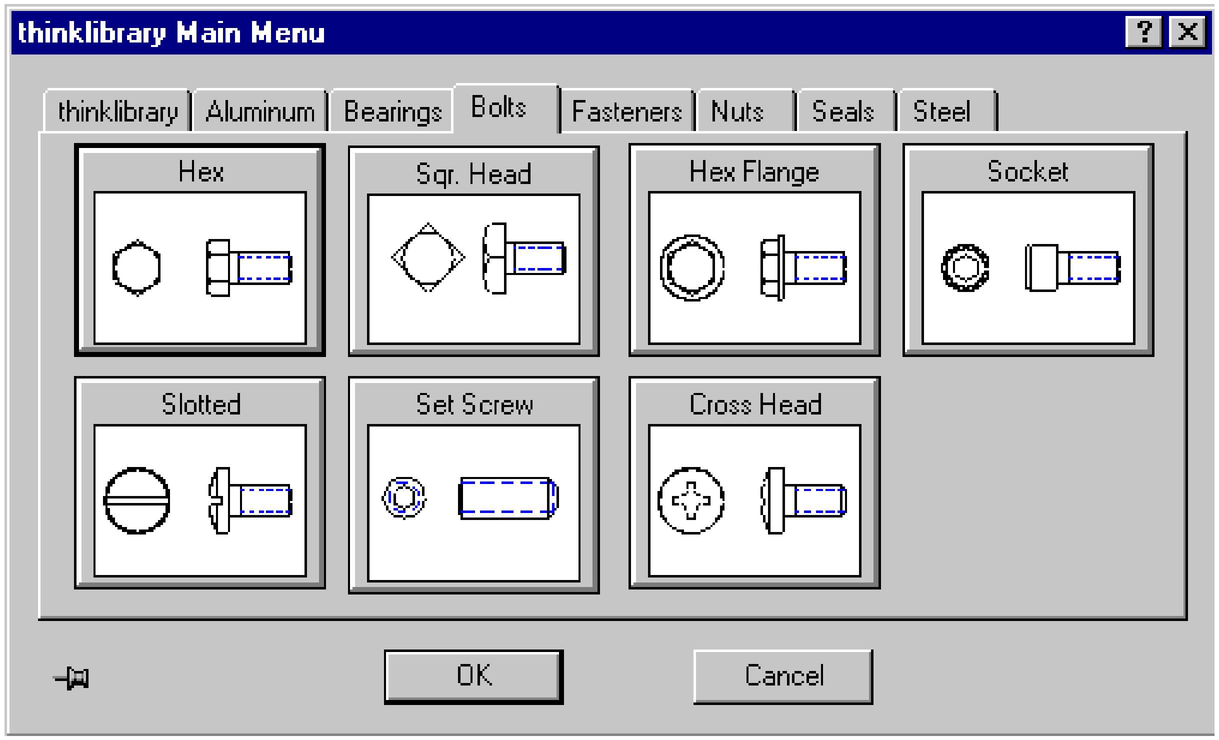
Uses of Computer Graphics

- **Display simulations:**
  - flight simulators, virtual worlds

Courtesy: Evans and Sutherland

Uses of Computer Graphics

- **Computer-aided design:**
  - architecture, electric circuit design

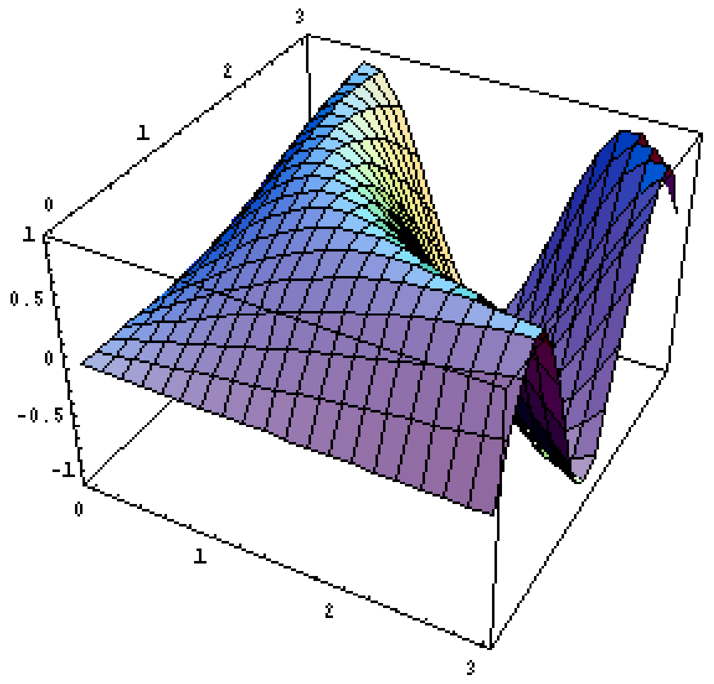


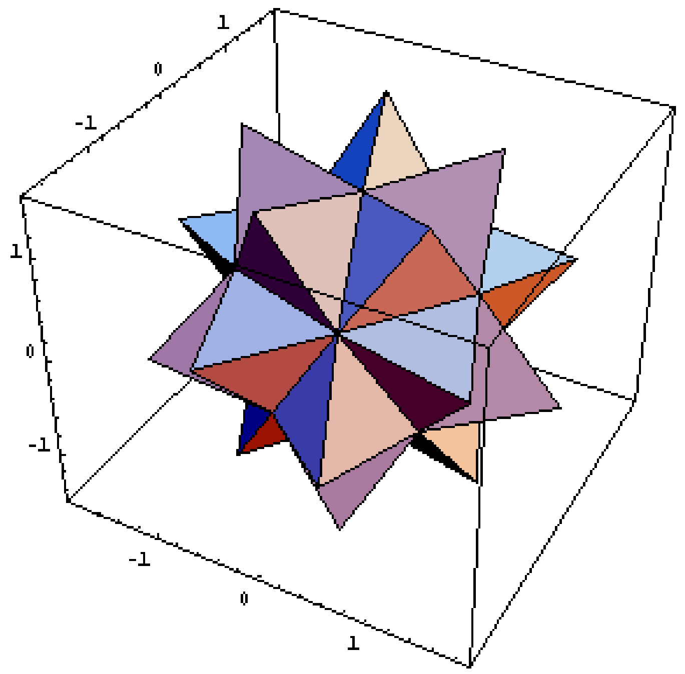
Courtesy:

cadalog.com

Displaying Mathematical Functions

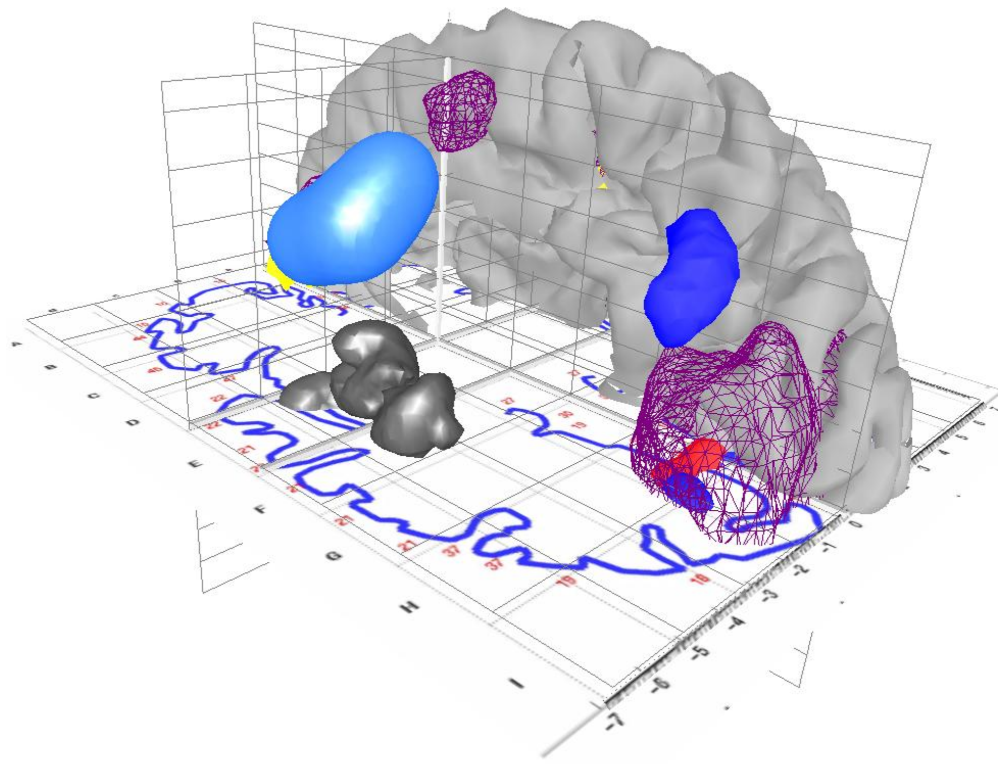
- E.g., Mathematica®





Uses of Computer Graphics

- **Scientific analysis and visualization:**
  - molecular biology, weather, matlab, Mandelbrot set

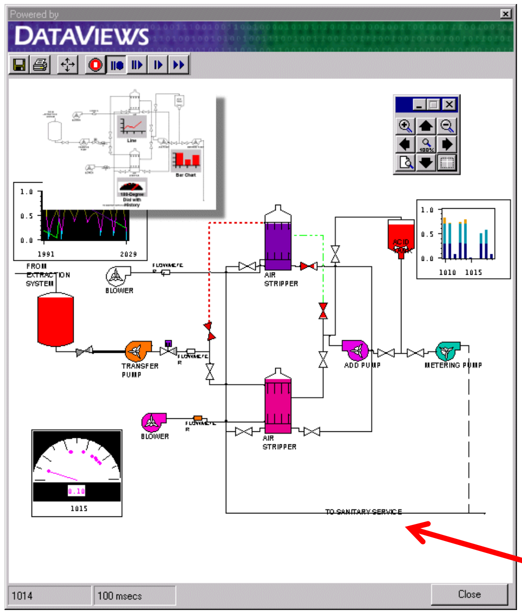


Courtesy:

Human Brain Project, Denmark

2D Vs. 3D

- 2-Dimensional
  - Flat
  - Only (x,y) color values on screen
  - Objects no notion of distance from viewer
- 3-Dimensional
  - (x,y,z) values on screen
  - Objects have distances from viewer



M. BLV
exp sunday
ALL VEHICLES
EXP WAY
NO TRUCKS
NVIDIA

This Class?

- Both 2D & 3D covered!
- Also interaction: Clicking, dragging

About This Course

- Computer Graphics has many aspects
  - **Computer Scientists** create graphics tools (e.g. Maya, photoshop)
  - **Artists** use CG tools/packages to create pretty pictures
  - Most hobbyists follow artist path. Not much math!
- This Course: Computer Graphics for computer scientists!!!
- Teaches concepts, uses OpenGL as concrete example
- Course is **NOT**
  - just about programming OpenGL
  - a comprehensive course in OpenGL. (Only parts of OpenGL covered)
  - about using packages like Maya, Photoshop

About This Course

- Class is concerned with:
  - How to build graphics tools
  - Underlying mathematics
  - Underlying data structures
  - Underlying algorithms
- This course is a lot of work. Requires:
  - Lots of coding in C/C++
  - Much more emphasis on shader programming than in past offerings
  - Lots of math, linear algebra, matrices
- We shall combine:
  - **Programmer's view:** Program OpenGL
  - **Under the hood:** Learn OpenGL internals (graphics algorithms, math, implementation)

Syllabus Summary

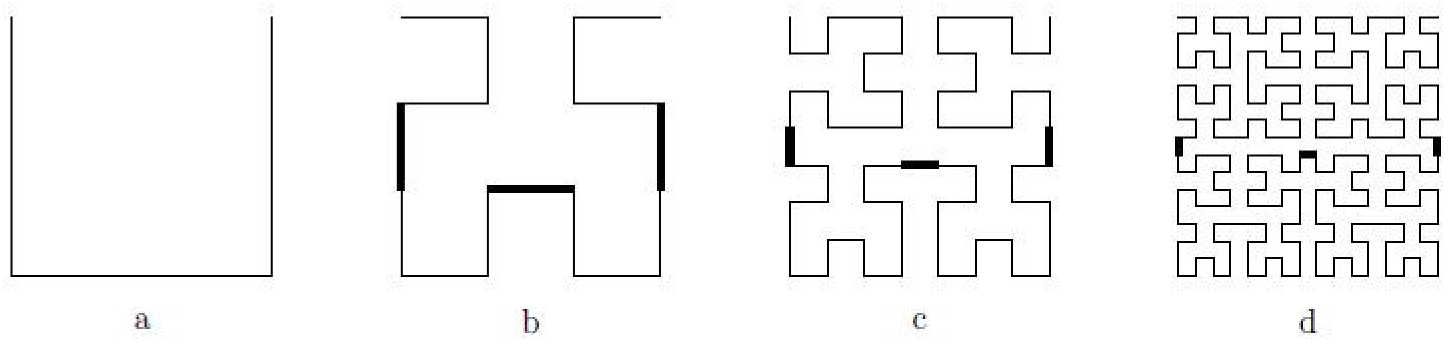
- 2 Exams (50%), 5 Projects (50%)
- Projects:
  - Develop OpenGL/GLSL code on any platform, must port to Zoolab machine
  - May discuss projects, turn in individual projects
- Class website: http://web.cs.wpi.edu/~emmanuel/courses/cs543/f12/
- Text:
  - Interactive Computer Graphics: A Top-Down Approach with Shader-based OpenGL by Angel and Shreiner (6th edition), 2012
- Cheating: Immediate 'F' in the course
- Advice:
  - Come to class
  - Read the text
  - Understand concepts before coding

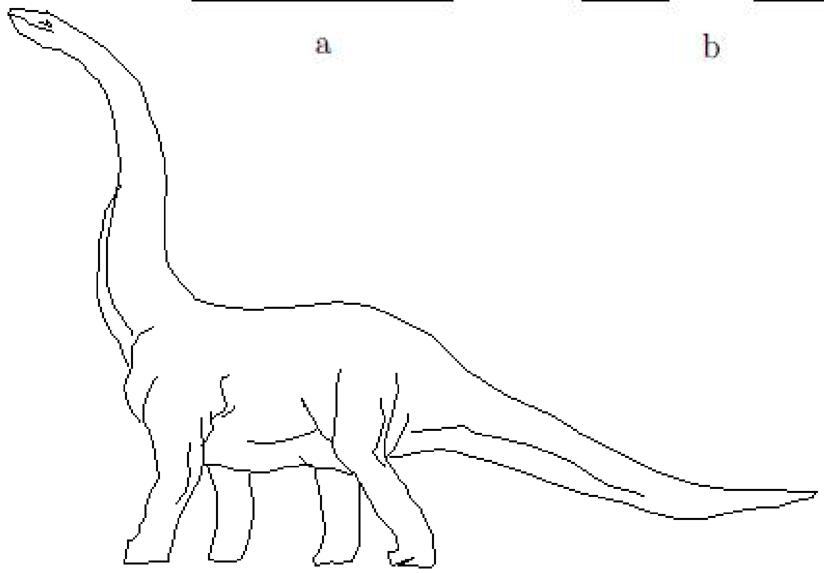
Elements of 2D Graphics

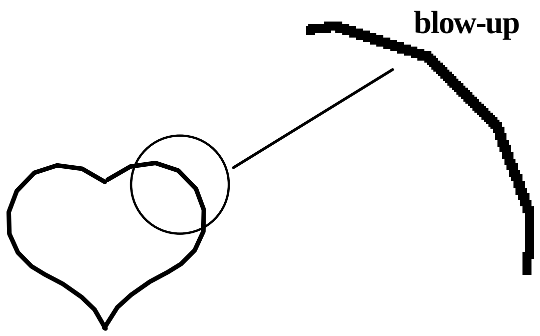
- Polylines
- Text
- Filled regions
- Raster images (pictures)

Elements of 2D Graphics

- **Polyline:** connected sequence of straight lines

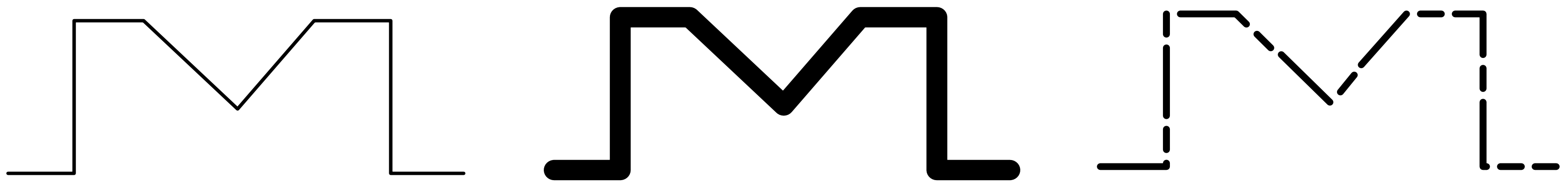






Polyline Attributes

- Color
- Thickness
- Stippling of edges (dash pattern)



Text

- Devices have:
  - **text mode**
  - **graphics mode.**
- **Graphics mode:** Text is drawn
- **Text mode:** Text not drawn uses built-in character generator
- **Text attributes:** Font, color, size, spacing, and orientation

Big Text

Little Text

Shadow Text

Distorted text

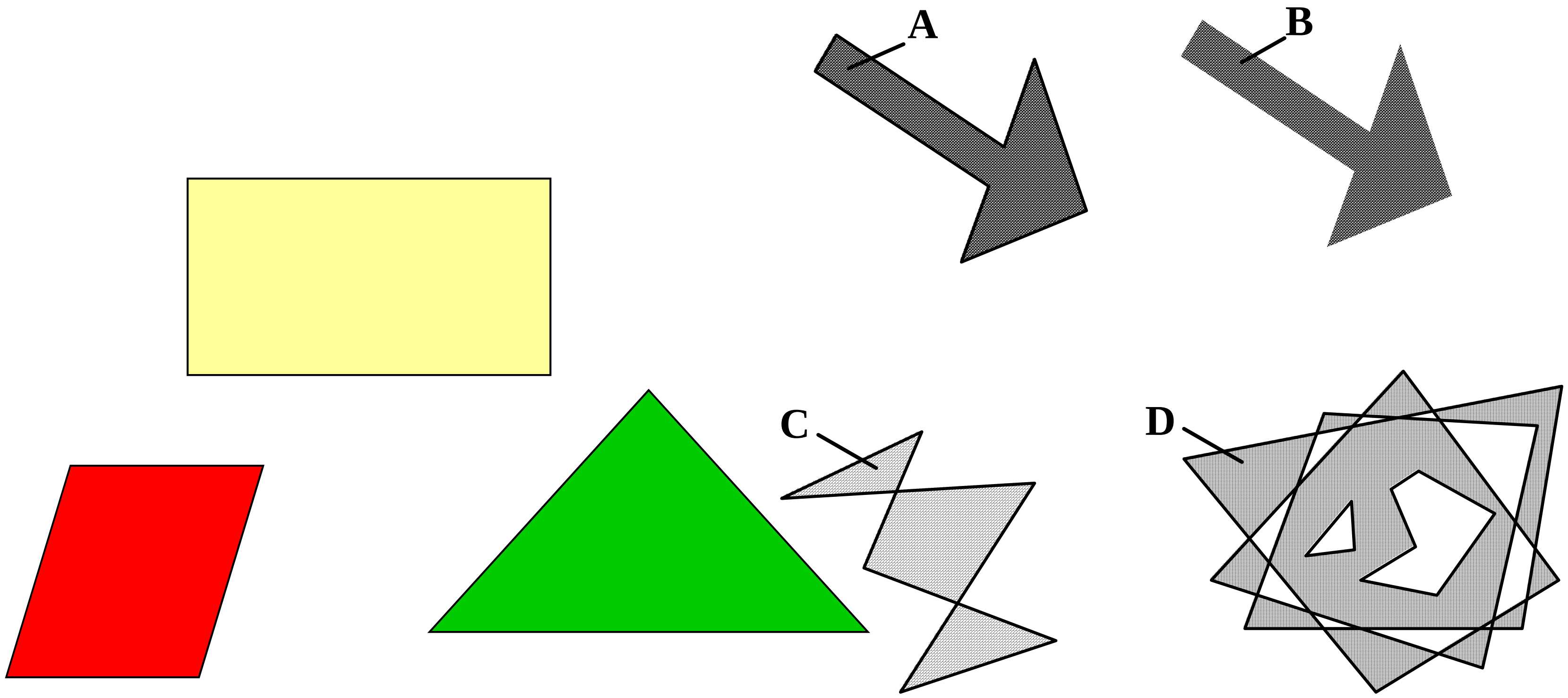
Rotated Text

Outlined text

SMALLCAPS

Filled Regions

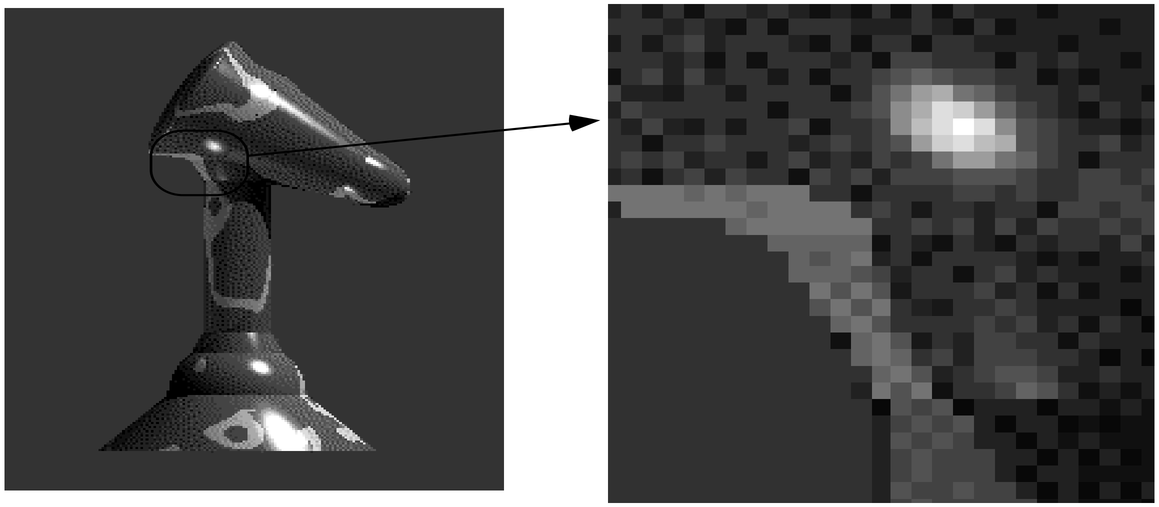
- **Filled region:** shape filled with some color or pattern
- Example: polygons



Raster Images

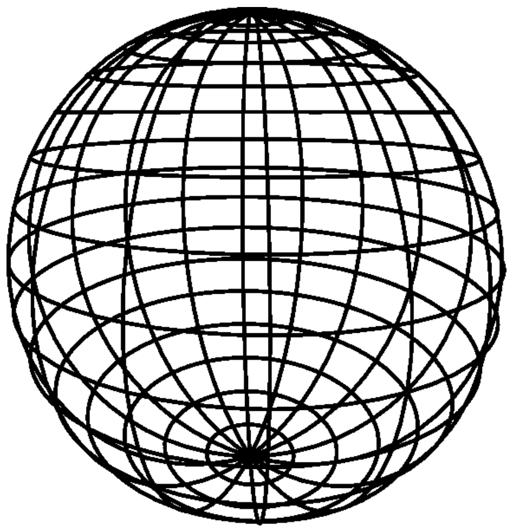
- Raster image (picture) is made up of many small cells (pixels, for “picture elements”), in different colors or grayscale.

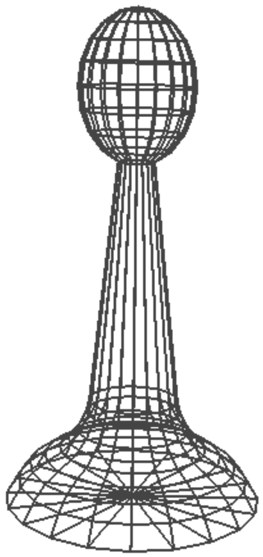
(Right: magnified image showing pixels.)

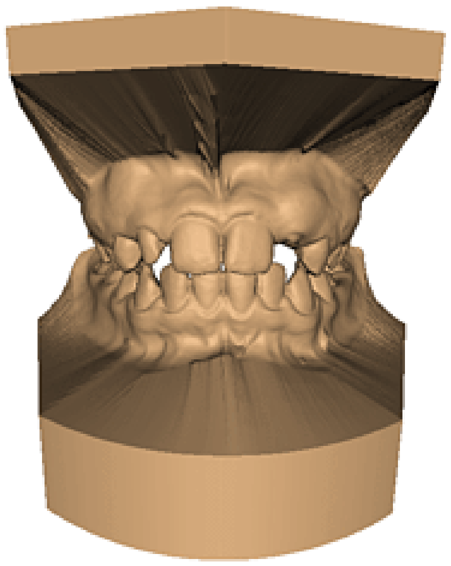


Creating 3D

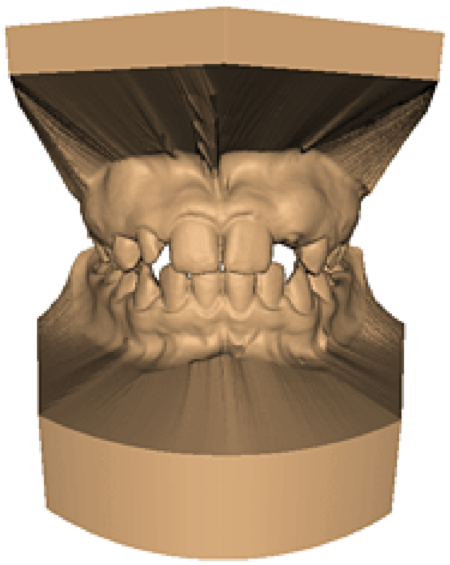
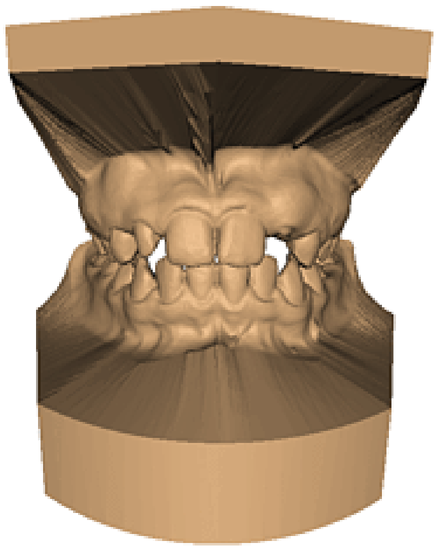
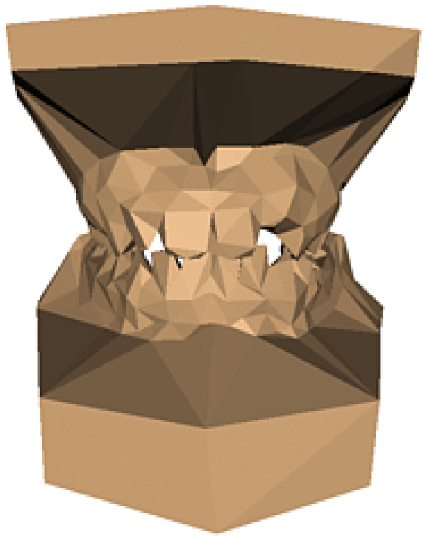
- Start with Basic 3D shapes (cube, sphere) or meshes
  - Scale shapes
  - Position or rotate shapes







3D Models at different Resolutions

Original: 424,000 triangles

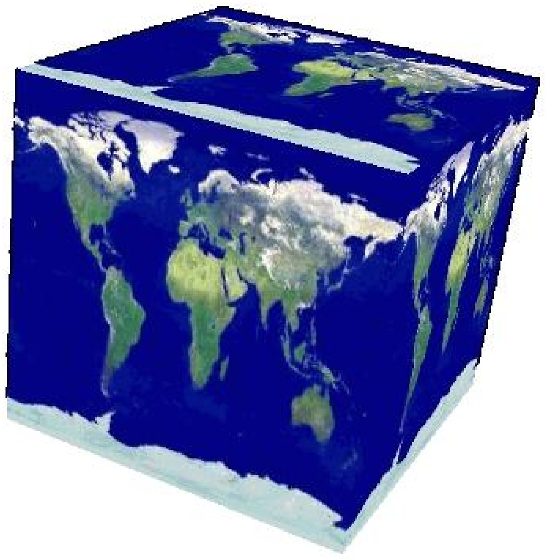
60,000 triangles (14%).

1000 triangles (0.2%)

(courtesy of Michael Garland)

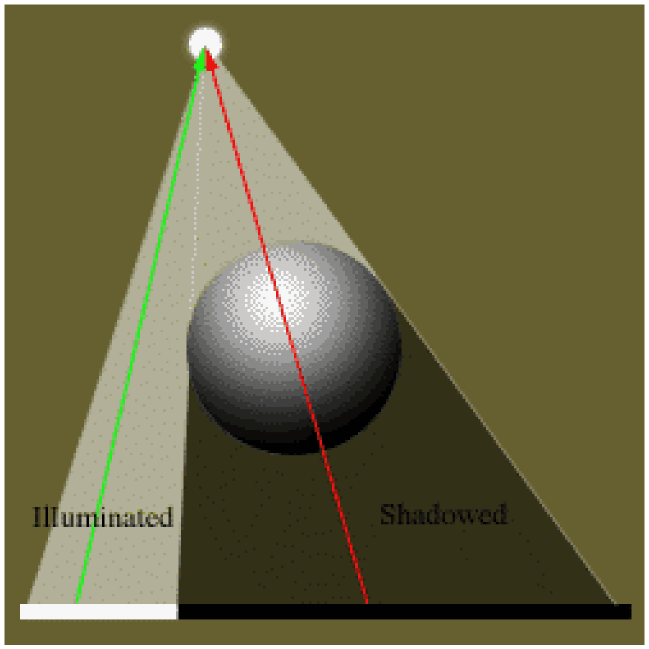
Creating 3D

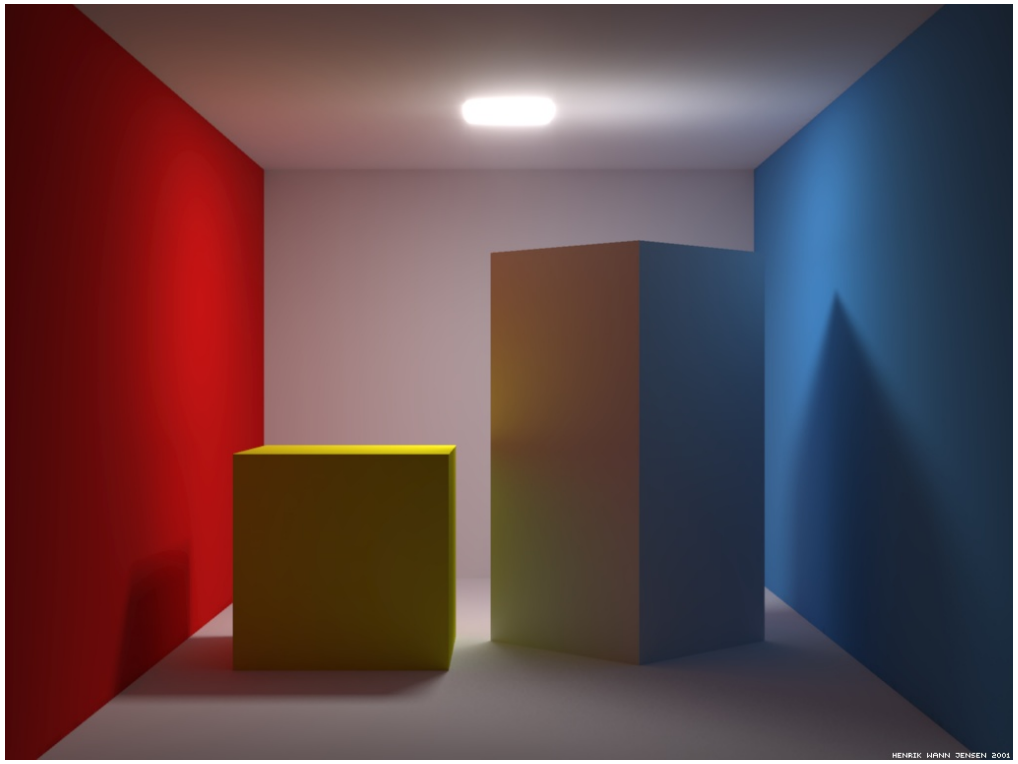
- Then, add 3D effects to make scene look real
  - Color and shading
  - Shadows
  - Texture mapping
  - Fog
  - Transparency and blending
  - Anti-aliasing



3D Effect: Texturing

3D Effects example: Shadows





Computer Graphics Tools

- **Require** hardware and software tools
- Hardware tools
  - **Output devices:** Video monitors, printers
  - **Input devices:** Mouse/trackball, pen/drawing tablet, keyboard
  - Graphics cards/accelerators (GPUs)
- Software tools (low level)
  - Operating system
  - Editor
  - Compiler
  - Debugger
  - Graphics Library (OpenGL)

Graphics Processing Unit (GPU)

- Entire OpenGL in hardware => FAST!!
- **Programmable:** in last 10 years (now as shaders)
- Located either on PC motherboard (Intel) or Separate graphics card (Nvidia or ATI)

On PC motherboard

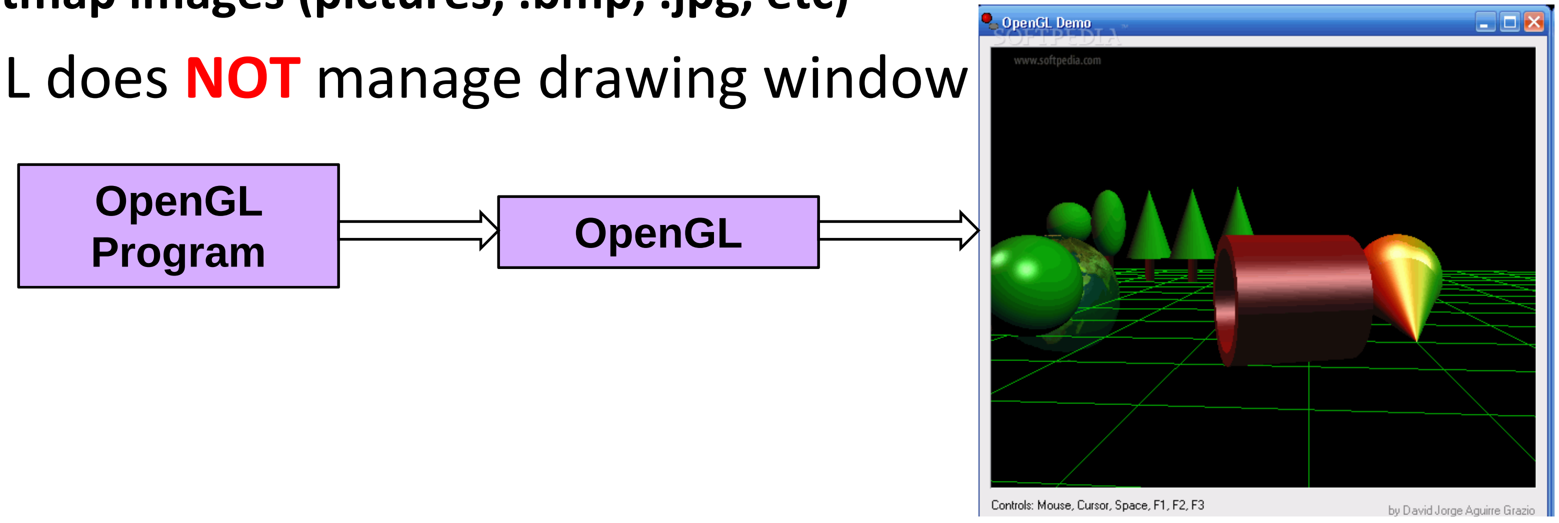
On separate PCI express card

Computer Graphics Libraries

- Functions to draw line, circle, image, etc
- Previously device-dependent
  - Different OS => different graphics library
  - Tedious! Difficult to port (e.g. move program Windows to Linux)
  - Error Prone
- Now device-independent libraries
  - **APIs:** OpenGL, DirectX, java3D
  - Working OpenGL program easily moved from Windows to Linux, etc

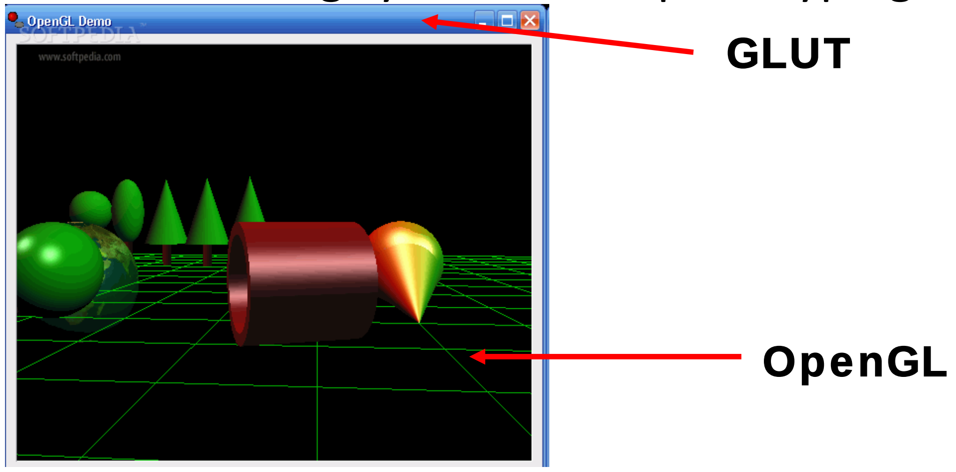
OpenGL Basics

- OpenGL's function – Rendering (or drawing)
- Rendering? – Convert geometric/mathematical object descriptions into images
- OpenGL can render:
  - **2D and 3D**
  - **Geometric primitives (lines, dots, etc)**
  - **Bitmap images (pictures, .bmp, .jpg, etc)**
- OpenGL does **NOT** manage drawing window


OpenGL Program
OpenGL
OpenGL Demo
www.softpedia.com
Controls: Mouse, Cursor, Space, F1, F2, F3
by David Jorge Aguirre Grazio

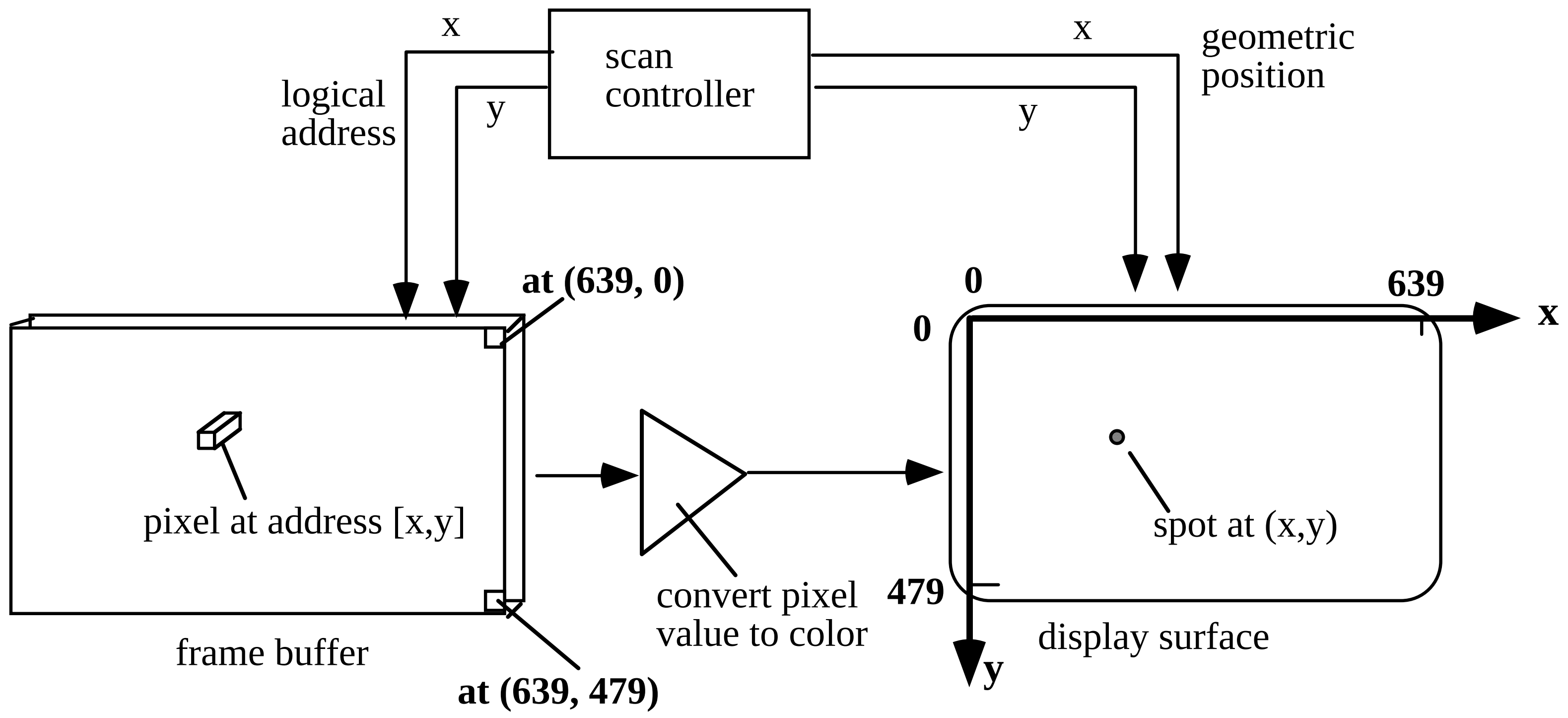
GL Utility Toolkit (GLUT)

- OpenGL
  - Window system independent
  - Concerned only with drawing
  - No window management (create, resize, etc), very portable
- GLUT:
  - Minimal window management
  - Interfaces with different windowing systems
  - Easy porting between windowing systems. Fast prototyping



Framebuffer

- Dedicated memory location:
  - Draw in framebuffer => shows up on screen
  - Located either on CPU (software) or GPU (hardware)



Types of Input Devices

- **String:** produces string of characters e.g. keyboard

- **Locator:** User points to position on display. E.g mouse

Types of Input Devices

- **Valuator:** generates number between 0 and 1.0
- **Pick:** User selects location on screen (e.g. touch screen in restaurant, ATM)

References

- Angel and Shreiner, Interactive Computer Graphics (6th edition), Chapter 1
- Hill and Kelley, Computer Graphics using OpenGL (3rd edition), Chapter 1