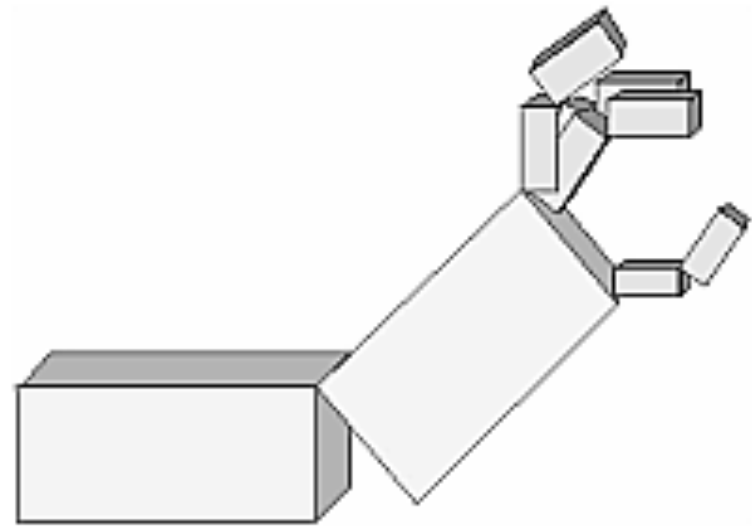
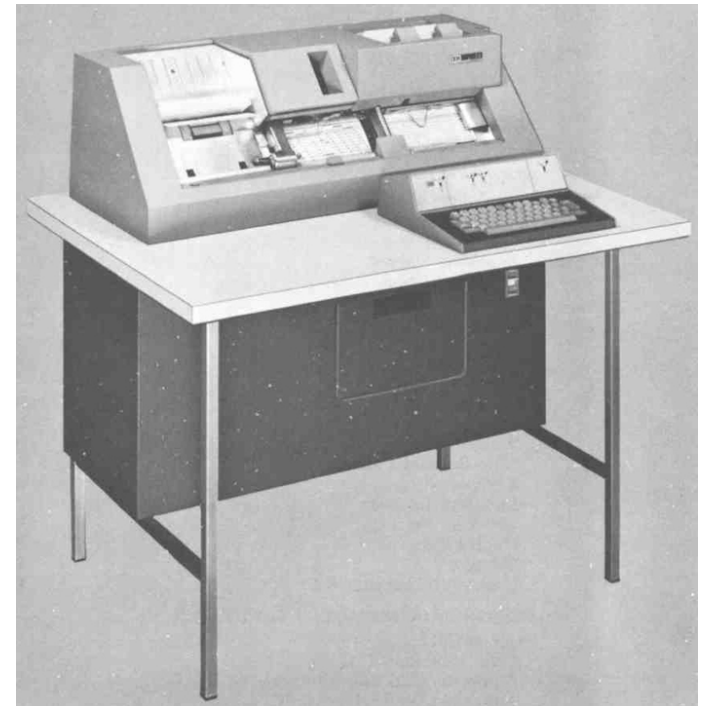
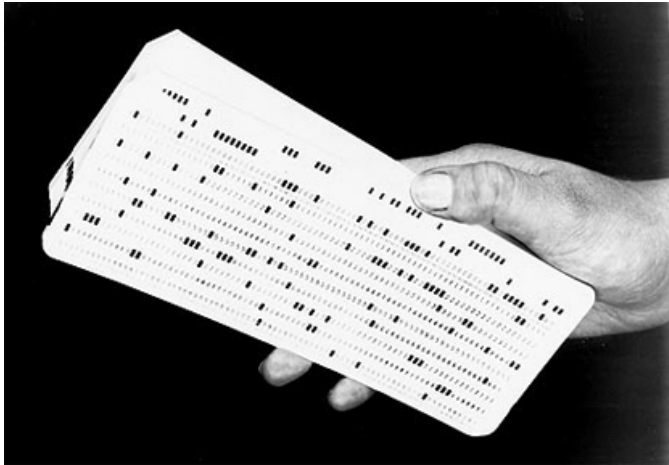


2. Overview of Graphics Systems



2-D Input/Output

- Historically, I/O lags behind computation
- Early computer, virtually no I/O devices
 - Teletype, tape punch
 - Punch cards, line printers
 - Keyboard, monitors (60s)



Output devices

- Visual (screens, projectors)
- Hard-copy devices
 - Laser
 - Ink-jet
 - 3D Printing (rapid prototyping machines)
 - NC Machining
- Tactile/haptic (often input/output)



Stereo and VR displays

- Active LCD Shutters synchronized to alternating display (120 Hz display, 60 Hz/eye)
- Head Mounted: LCD for each eye.



HMD



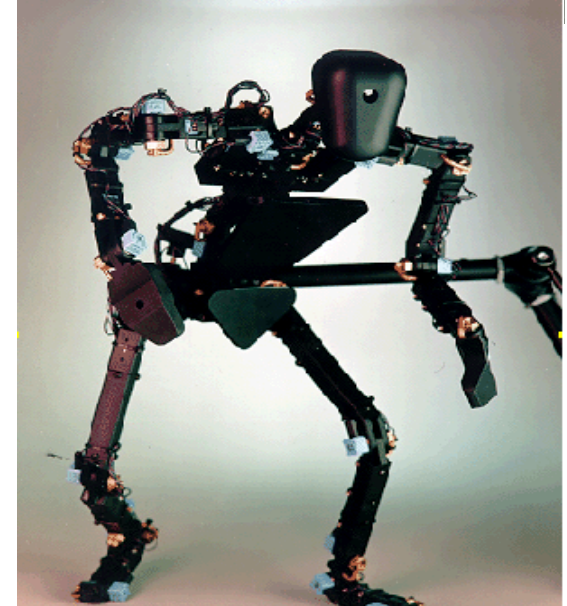
Fish tank

Input Devices

- Mouse introduced in early 60s, not used until introduction of GUI in 80s (Apple)
- Drawing tablet affordable and practical in 80s

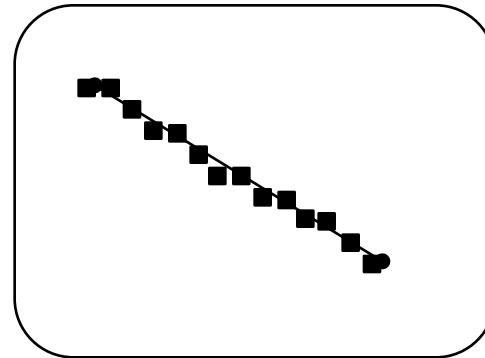


- Alternate Input devices
 - Sample motion track hand/body
e.g. mouse, glove
 - Sample other data
e.g. light, color, pressure, position
scanners, digital camera/camcorder

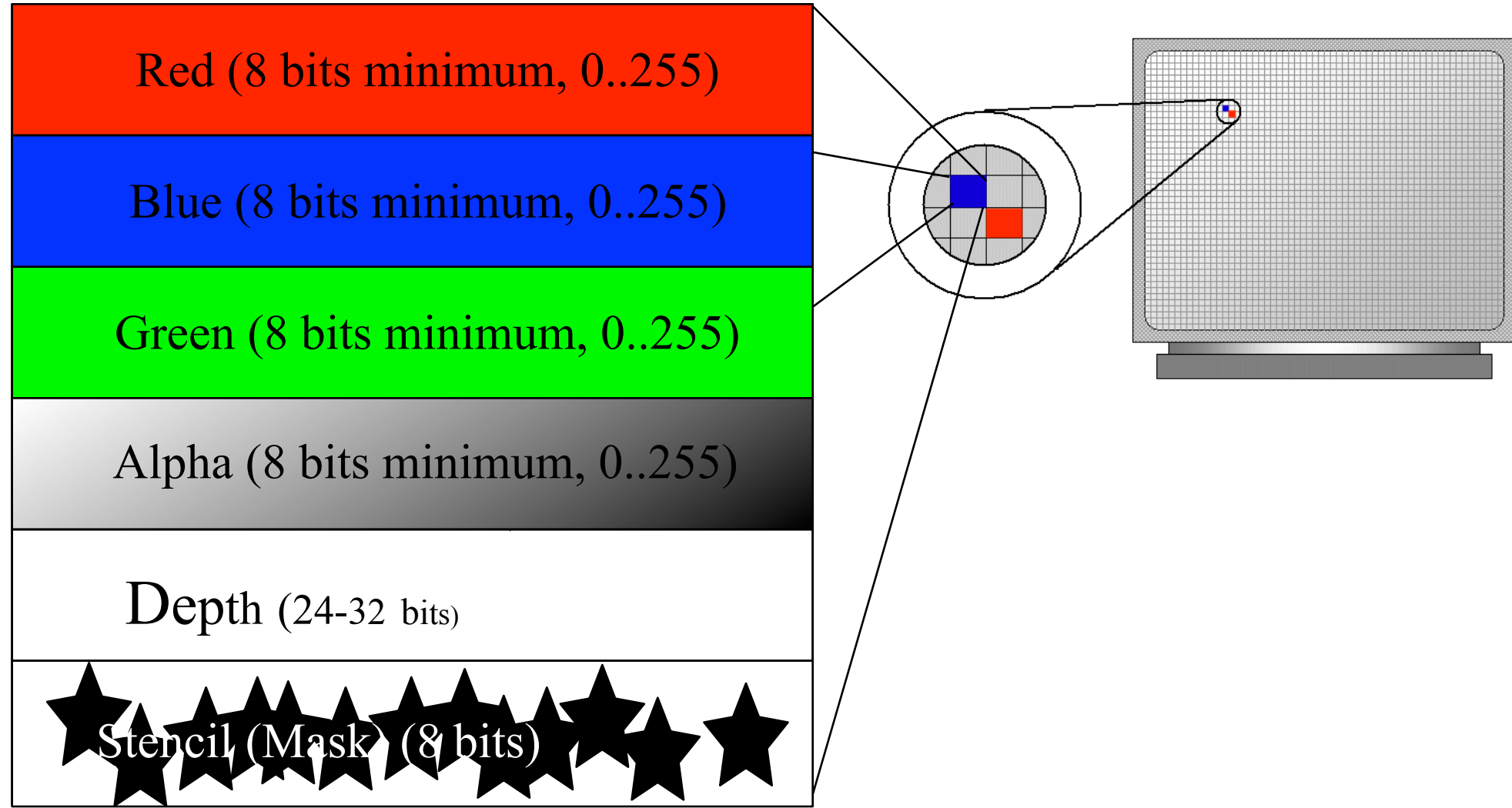


Raster Display

- Raster: rectangular array of pixels
- Pixel: picture element or one “dot” in the picture
- Scanline: one row of pixels
- Raster display: a display device that draws a memory composed of pixels (framebuffer)

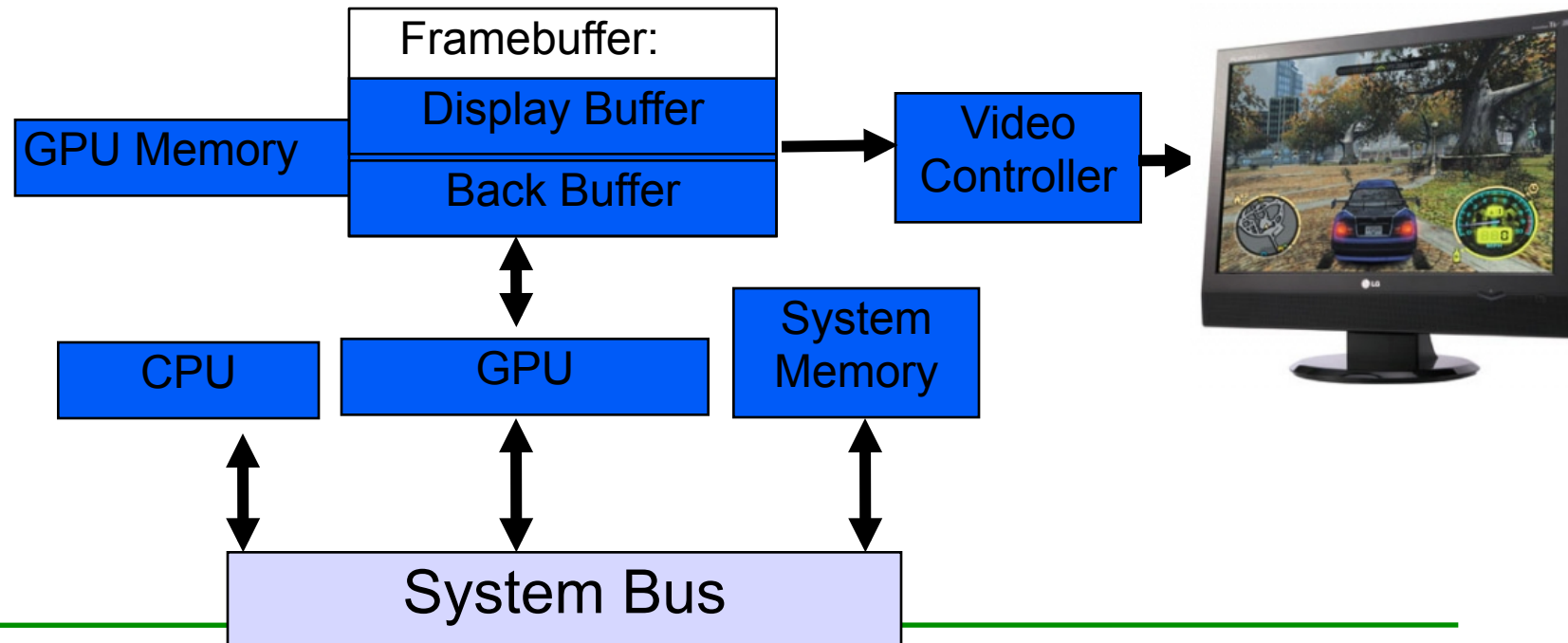


Pixel



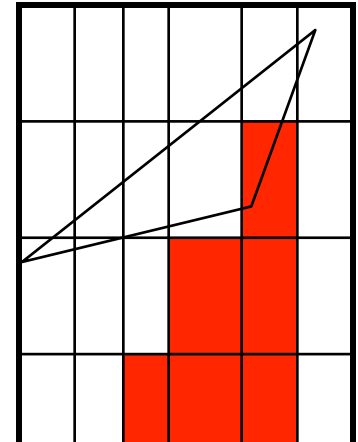
Display Processor

- Also graphics controller, GPU, or video card
- Framebuffer usually has 2 main buffers: display and refresh (also called “back buffer”)
 - While video controller draws display buffer, GPU writes next image to the refresh buffer, then swaps



Rendering to the Framebuffer

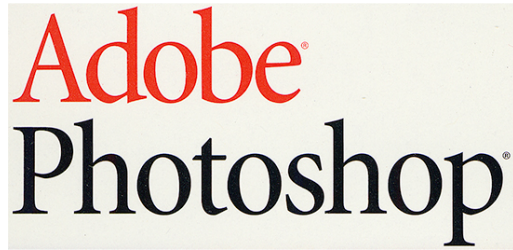
- Scan conversion
 - Digitize 3D models pixel by pixel
 - also called rasterization
- Can also use Ray-Tracing and other more realistic (but slower) methods.



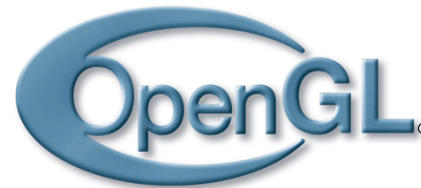
<http://www.oyonale.com/modeles.php?lang=en&page=40>

Graphics Software

- Special purpose packages
 - May have scripting language, but not general-purpose api
 - End-user products

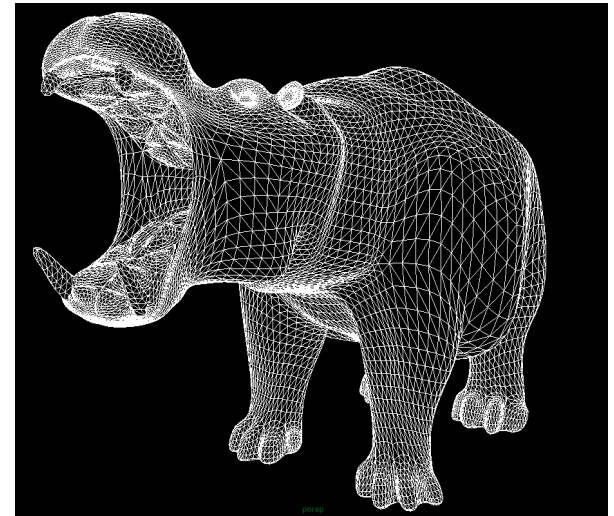


- General purpose APIs
 - Library of functions for 2D and 3D drawing
 - Draw geometry - lines, polygons
 - Set viewing transformations and parameters
 - Manage shaders



Graphics API Functions

- Graphics primitives
 - Basic building blocks for pictures
 - Character strings, points, polygons, curves, lines, spheres, cylinders, cones, etc.
- Graphics Attributes - display properties for primitives
 - Color, surface texture
 - Shader
- Transformations between reference frames
- Input functions

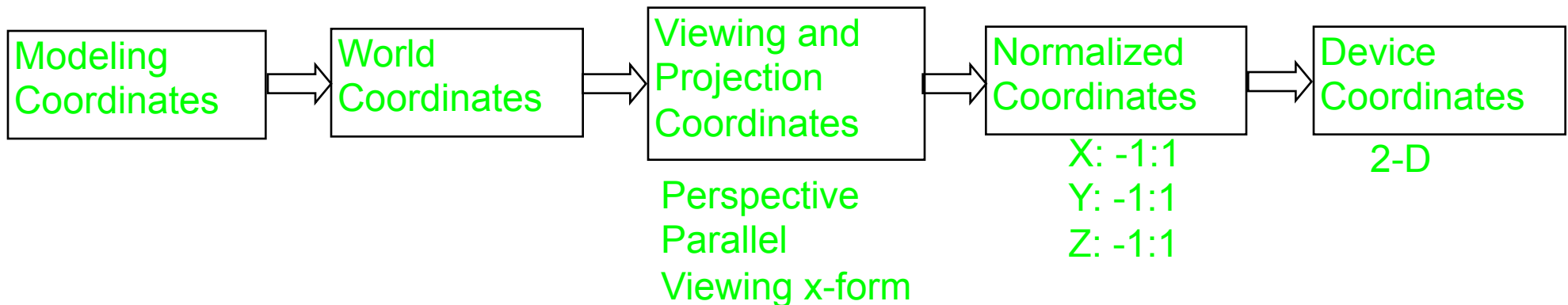


Reference frames

- Modeling coordinates
 - Local coordinate where an object is defined
 - E.g. the wheel of a car
- World coordinates
 - Coordinate where all the objects are placed
 - E.g. the 4 wheels, the body, etc. of a car placed in the “world”
- Viewing coordinates
 - Project scene onto a ‘virtual camera’
 - Coordinates relative to the camera
- Device coordinates
 - Position on-screen

Viewing pipeline

- The set of transformations from an object starting coordinates to screen coordinates



Most Popular APIs

- OpenGL
 - Hardware-independent open standard
 - Primary API for Mac, iPhone, Android, Unix
 - Works on Windows – but fewer tools than DirectX
 - In Javascript via WebGL
 - www.opengl.org
- DirectX
 - Not just graphics (sound, etc)
 - Frequent updates
 - Primary API on Windows and Xbox
 - DirectX



Introduction to OpenGL

- Graphics Rendering API
 - Only rendering, almost always implemented in hardware
- Create virtual scenes from geometric and image primitives
- Images linked to geometry via texture mapping
- Platform independent

```
glMatrixMode(GL_PROJECTION); // Matrix update
glLoadIdentity();
gluOrtho2D(0.0, 200.0, 0.0, 200.0);
glColor3f(1.0, 0.0, 0.0); // Update rendering state
glBegin( GL_TRIANGLES );
    glVertex3f( 50.0, 100.0, 0.0 );
    glVertex3f( 75.0, 150.0, 0.0 );
    glVertex3f( 100.0, 200.0, 0.0 );
glEnd();
```

Libraries

- OpenGL basic library (Core library)
 - Device independent
 - Platform-specific extensions for setup and window-management
 - WGL (windows), CGL (apple), GLX (x-windows)
 - OpenGL Utility (GLU)
 - Supplements OGL with some higher-level functions
 - 2D image scaling
 - Rendering 3D objects including spheres, cylinders, and disks
 - OpenGL Utility Toolkit (GLUT)
 - Platform-independent system to setup and manage OpenGL windows
 - Quadratic curves/surfaces
 - Input (keyboard and mouse, etc.) and event handling
 - <http://www.opengl.org/resources/libraries/glut/spec3/spec3.html>
-