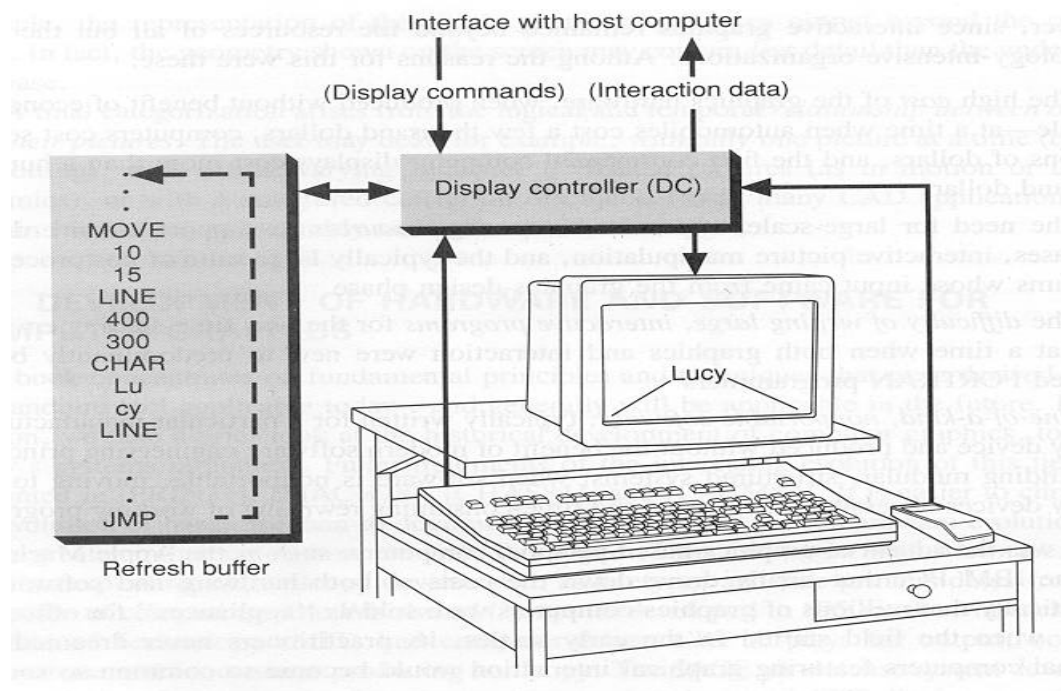


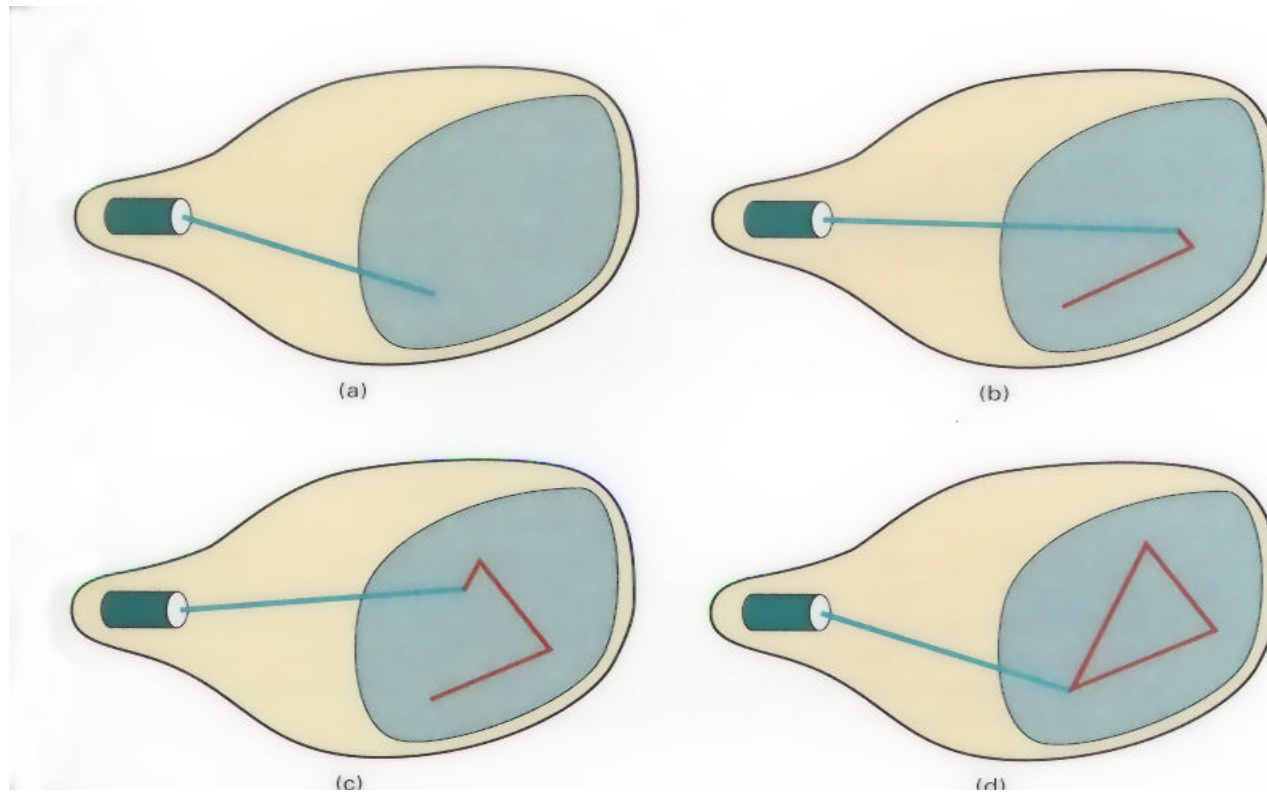
Development of Computer Graphics

- Early display devices:
- Vector, stroke, line drawing, calligraphic displays
- Architecture of Vector Display



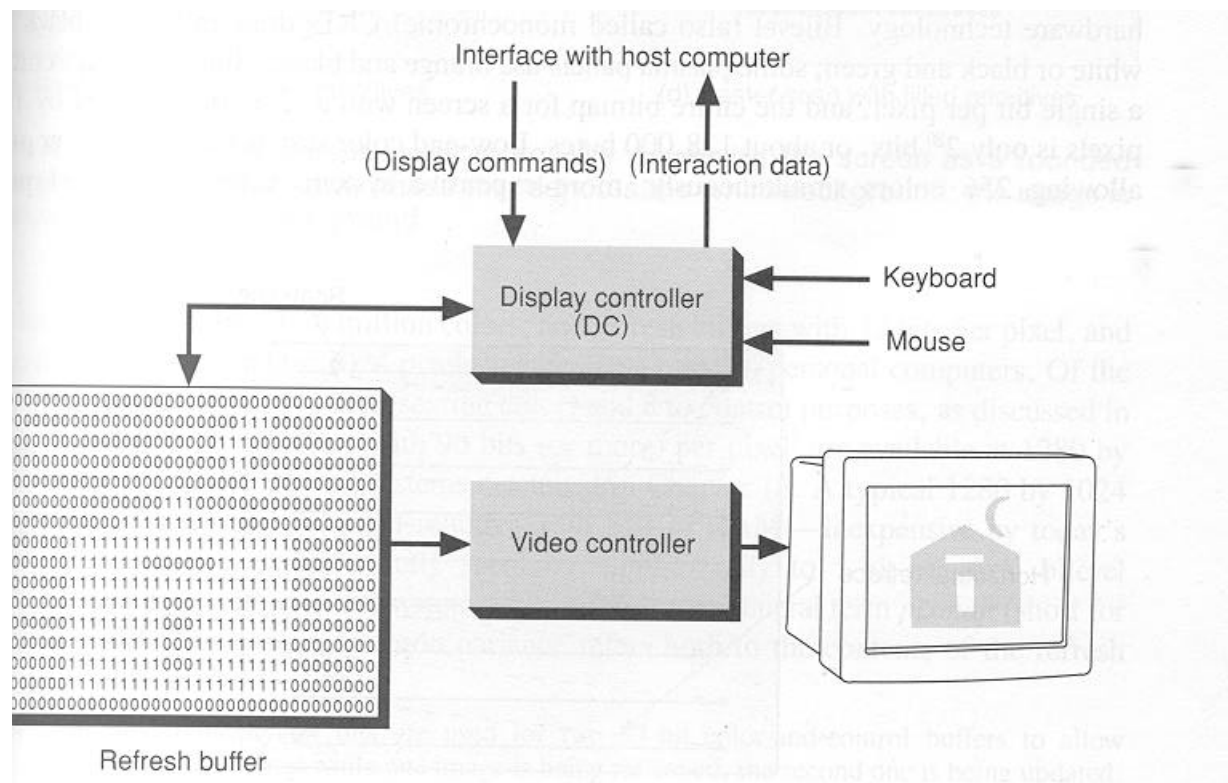
Development of Computer Graphics

- Architecture of a vector display - random scan
 - vector generator converts digital coordinates to beam deflections



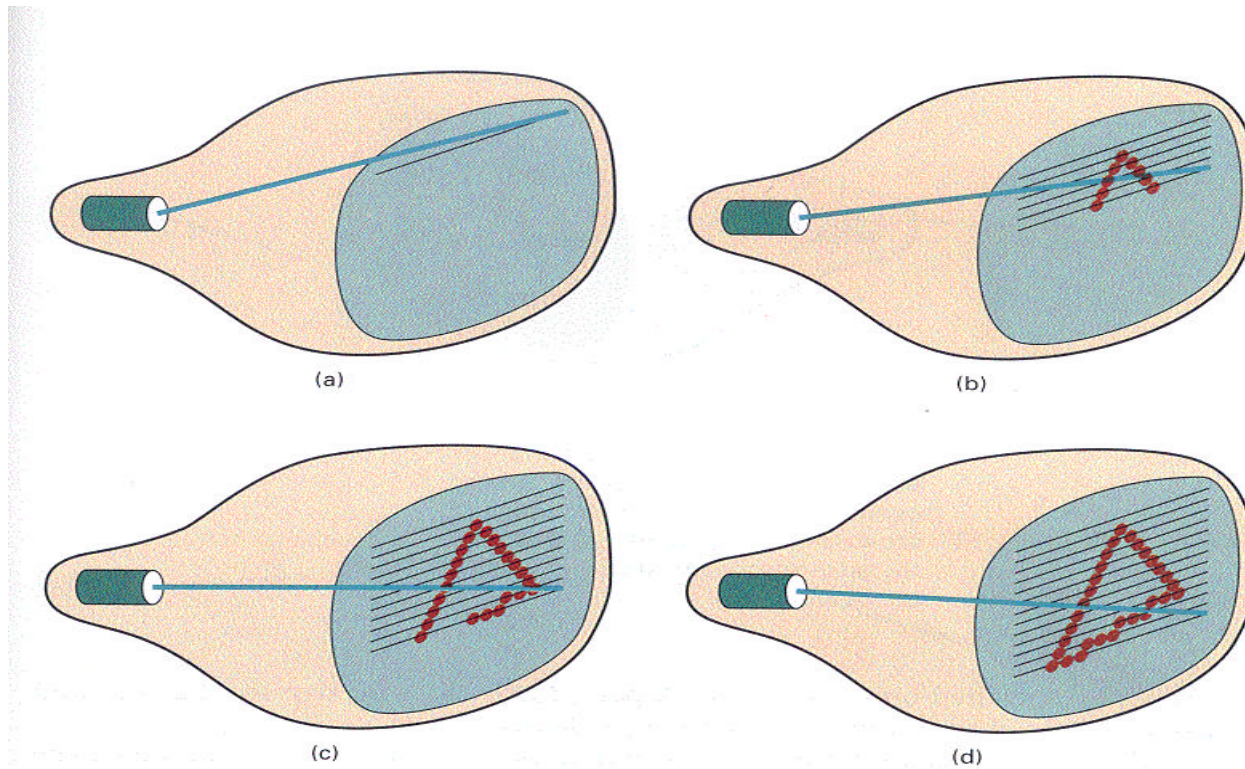
Development of Computer Graphics

■ Architecture of a raster display



Development of Computer Graphics

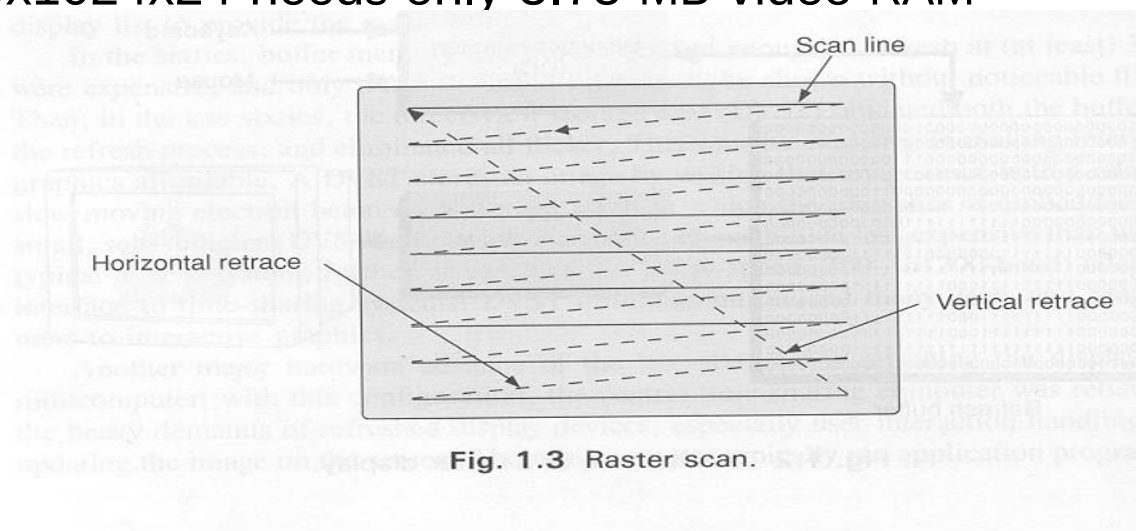
- Architecture of a raster display - raster scan
 - beams (3 beams) intensity set to reflect pixel intensity
 - scan speed: originally 30Hz now 60Hz



Development of Computer Graphics

■ Raster Scan

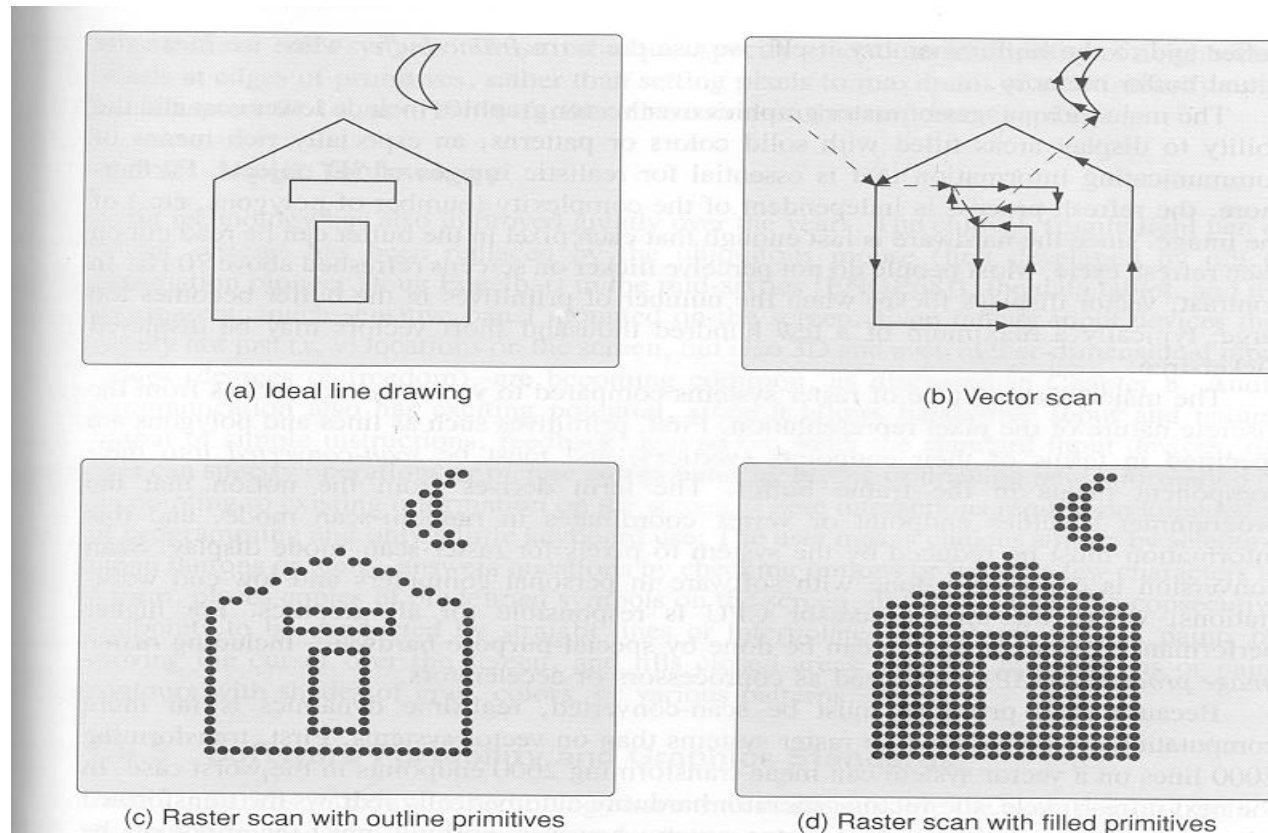
- need to store whole image
- $1024 \times 1024 \times n$ - n bits per pixel
- mono 1 bit, color 8 (256 color), 24 (16 million)
- 32 to 96 bits used (*double buffering, z-buffering*)
- $1280 \times 1024 \times 24$ needs only 3.75 MB video RAM



Development of Computer Graphics

■ Random Scan versus Raster Scan

- note ragged lines



Development of Computer Graphics



- Random Scan versus Raster Scan
- Raster advantages:
 - low cost, superior fill ability, refresh rate independent of complexity
- Raster disadvantages:
 - discrete nature of pixel representation, need for scan conversion in software or RIP chips
 - real-time dynamics more demanding
 - approximation of lines by pixels
 - | *aliasing* - jaggies or staircasing
 - | manifestation of sampling error in signal processing
 - | need for *anti-aliasing*

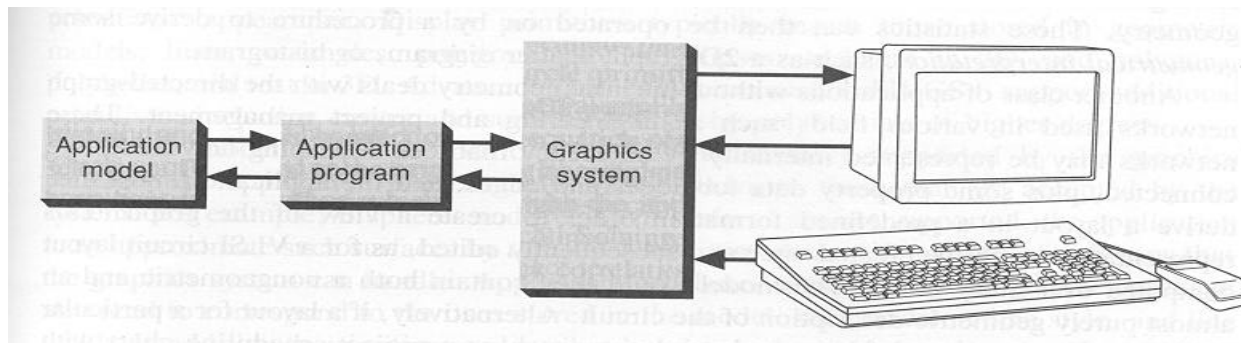
Development of Computer Graphics



- Input Devices
 - early light pens to modern mice
 - data tablet
 - touch sensitive screens
 - 3D input devices (spaceballs etc.)
 - button and dial boxes

Describing Scene to be viewed

- Application Program - creates application
- Application Model - independent of display system
 - program must extract geometry and convert to *primitives* of graphics system
 - primitives: points, lines, (rectangles, ellipses, text, polygons, polyhedra, spheres, curves, surfaces
 - application must convert geometry to primitives supported
 - *attributes* (line style, color, line width, fill style)



Graphics Systems



- Typically libraries: output subroutines
 - user program in *logical display device* terms
 - graphics library converts to device dependent instructions
 - abstraction of display device
 - *locator* - mouse, tablet, joystick etc
 - *sample* - return from locator
 - *event* - generated by user input

Graphics Systems



- Interaction Handling - *event driven loop*
 - **while** (!quit) {
 - | enable selection of commands/objects
 - | wait for user selection
 - | **switch** (selection) { process selection, updating model and screen as necessary}
 - }
- User interaction
 - change in screen appearance - does not involve update of model: application updates state and calls graphics package
 - change in model : must recalculate