Introduction to Computer Graphics

Farhana Bandukwala, PhD

Lecture 2: Raster Graphics

Outline

- More logistics
- Raster vs Vector graphics (Angel, Section 1.2)
- Advantages and Drawbacks
- Algorithms (Angel, Section 8.9)
- Bit and Pixel operations in Open GL (Angel, Chapter 7)

Logistics

- Web accounts, class web page
- Visual C++ tutorial
- Opengl: library to simplify graphics code
  - Windows libraries: opengl32.lib,glu32.lib,glut32.lib
  - headers:
    - GL/gl.h, GL/glu.h, GL/glut.h

What is an image?

- A 2-D array of intensity values which depict a scene
- Analog: picture from a regular camera
  - Pixel size is infinitesimally small
  - Maximum resolution & bandwidth
- Digital: scanned image, CCD or digital camera
  - Pixel size is finite
  - Quality of representation related to pixel size, number of pixels
- Graphics provides means to display an image on an output device (monitor, printed media, etc.)
- Image processing consists of pixel manipulation NOT display

Vector graphics

- Used in display devices of ’60s
- Components:
  - Display buffer: stores display list
  - Display processor: executes commands
  - CRT: beam deflected accordingly
- Advantages:
  - smooth lines
- Drawbacks:
  - Performance
  - no filled regions

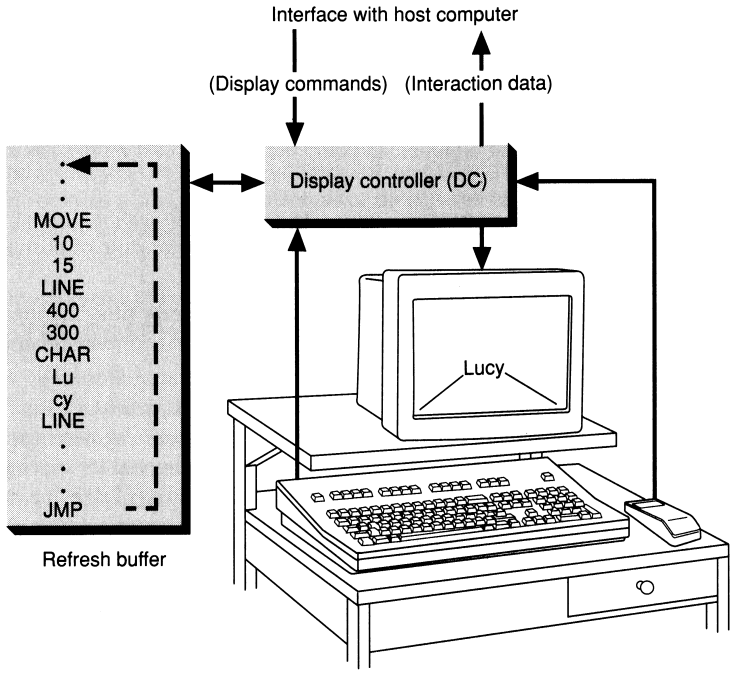


Fig. 1.1 Architecture of a vector display.

Raster graphics

- Raster: matrix of pixels representing screen space
- Primitives stored in buffer (memory) as pixels
- Video controller scans out buffer
- Beam's intensity determined by each pixel
- Bitmap vs Pixmap

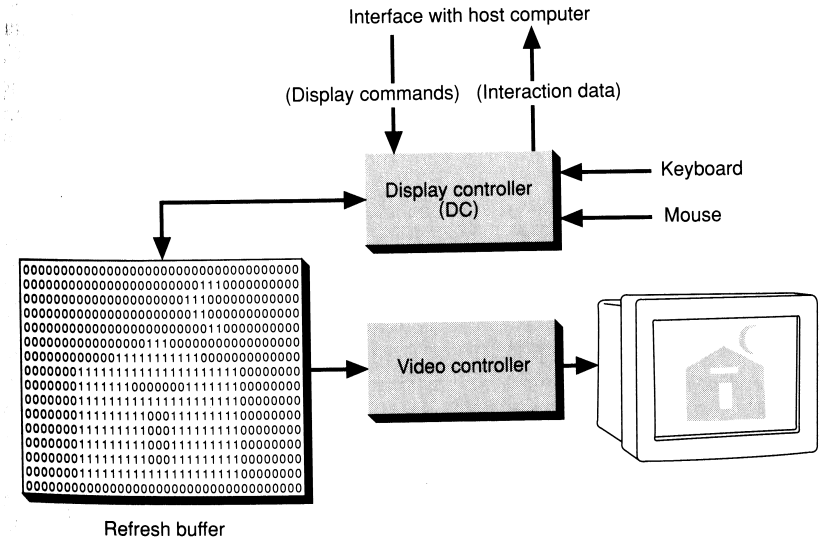
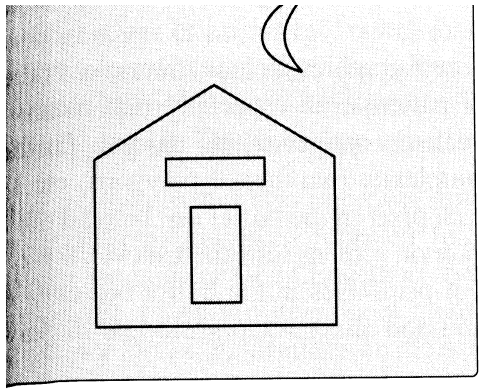
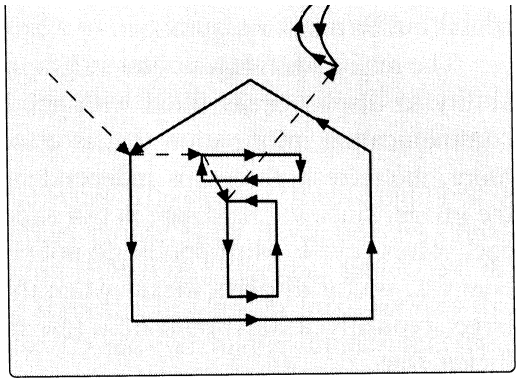


Fig. 1.2 Architecture of a raster display.

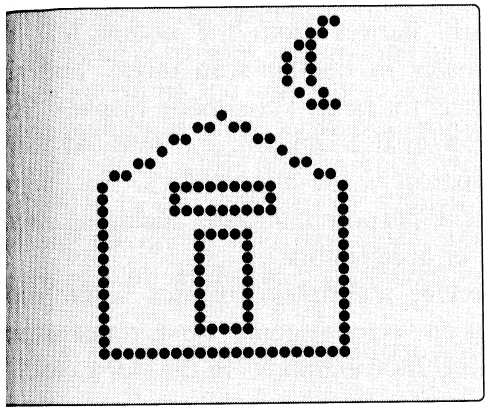
Vector vs Raster Graphics



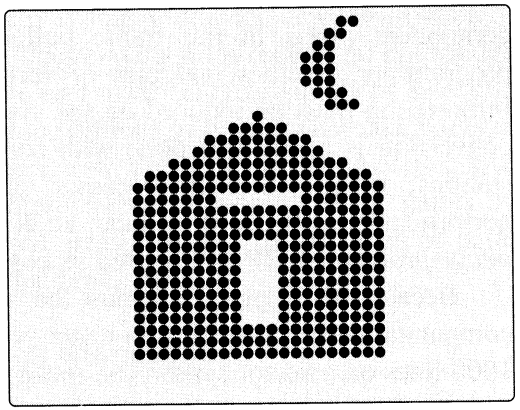
(a) Ideal line drawing



(b) Vector scan



(c) Raster scan with outline primitives



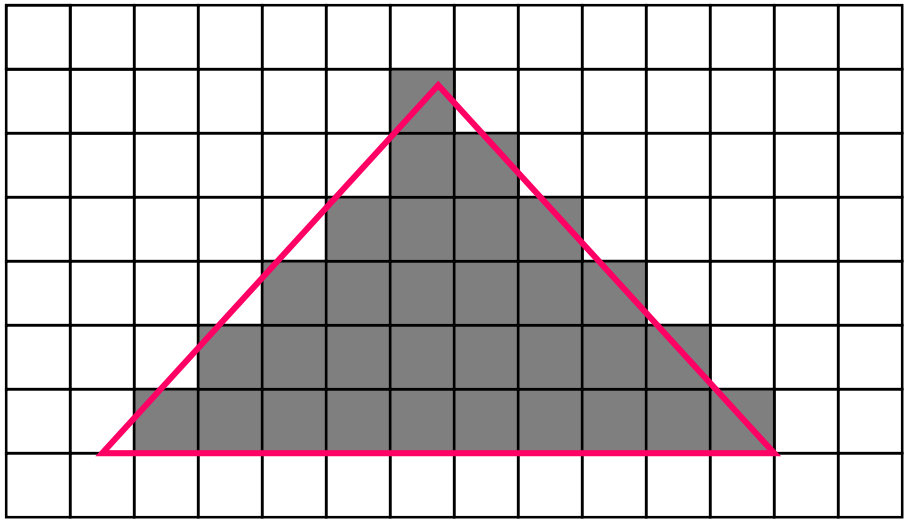
(d) Raster scan with filled primitives

Advantages of Raster Graphics

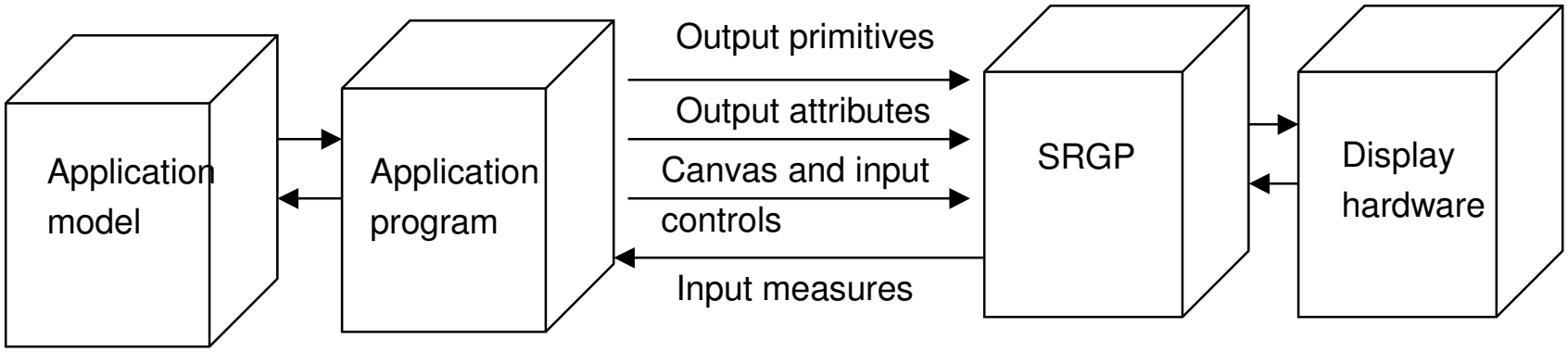
- Faster, lower cost refresh hardware
- True filled areas: realistic images
- Refresh rate independent of scene complexity

Drawbacks

- Primitives have to be scan converted
- Aliasing
- Solution: powerful algorithms

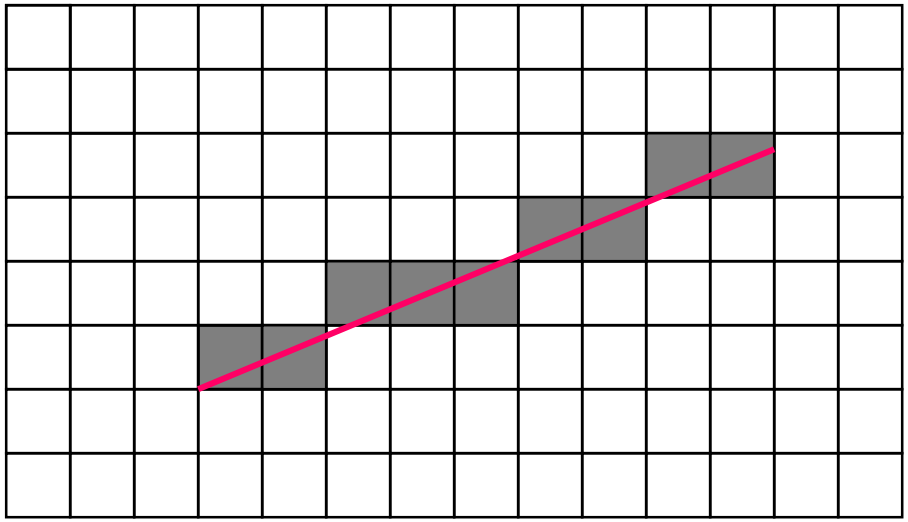


Simple Raster Graphics Package



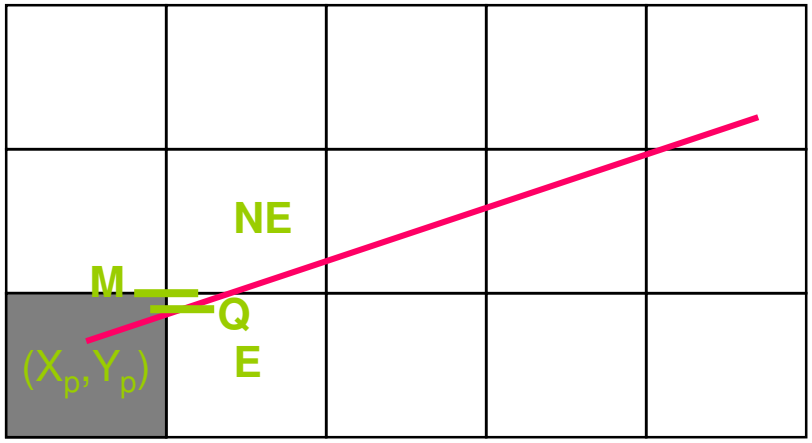
- Input: primitives, attributes, application controls
- Output: raster into frame buffer
- Package scan converts, clips and does anti-aliasing

Scan converting lines



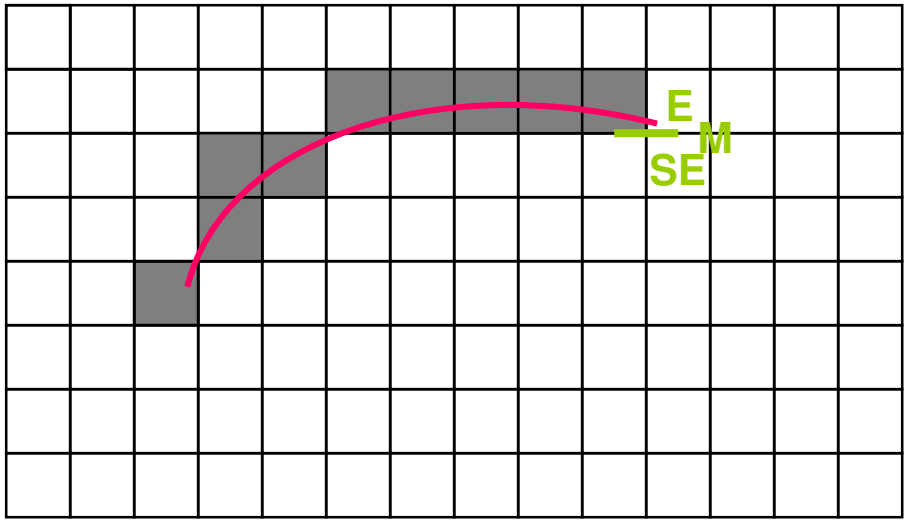
- Initialize: $x_0=x_{min}$, $y_0=y_{min}$
- Basic increment: $\Delta x=1$, $y_{i+1} = y_i+m$
- Simple but computationally expensive (rounding)

Scan conversion: contd



- Midpoint Line:
  - Compute difference of distances from E to Q, NE to Q
  - Use sign of difference to select next pixel (E or NE)
  - If midpoint above line choose E, otherwise choose NE
  - Use symmetry for other quarters

Scan converting circles

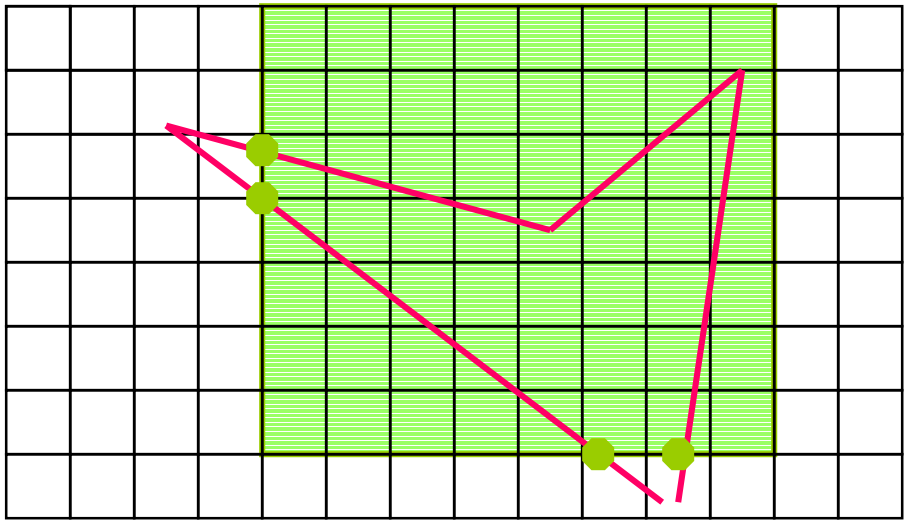


- Basic $y=+/-\sqrt{(r^2-x^2)}$
- Speedups:
  - Draw quarter circle & use symmetry
  - Midpoint circle algorithm

Scan converting filled polygons

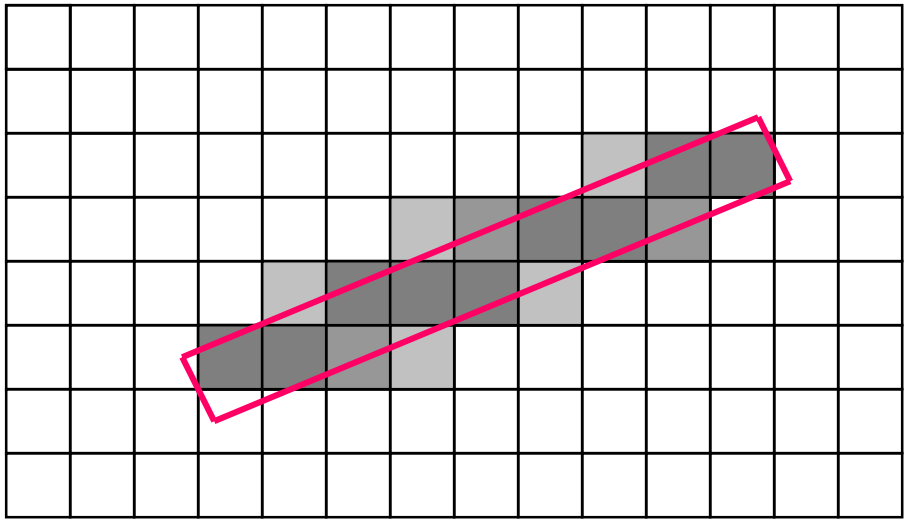
- Find all intersections (in green) at a scan line
- Sort in ascending x value
- Using parity checks, fill pixels along scan line (initially even parity, flip parity at intersection, color pixel at odd parity)

Clipping



- Simple BBox test
- Each polygon edge tested against each rectangle edge
- New vertices added if necessary

Anti Aliasing



- Assuming finite width primitives,
- Pixel's intensity proportional to amount of area covered (unweighted area sampling)
- Weighted area sampling: Radially decreasing weights from pixel center

Pixel operations in Open GL

- Various buffers: color, depth, stencil
- Logic operations on bitmaps & pixmaps
- Pack/Unpack to convert pixel to/from OpenGL format
- Bitmaps as masks
- Design raster fonts
- Look up tables
- For help with picking
- Texture mapping