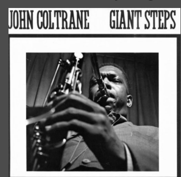
Now Playing:



Giant Steps
From *Giant Steps*
Recorded May 4-5, 1959
John Coltrane - Tenor Sax
Tommy Flanagan - Piano
Paul Chambers - Bass
Art Taylor - Drums

Rasterization and Real-time Graphics

COMP 575
August 21, 2007

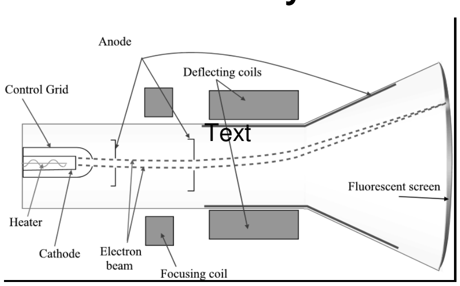
Announcements

- Change in office hours:
  - Wednesday office hours moved to 3:30-5:00pm
- Schedule is linked from the course home page:
  - http://www.cs.unc.edu/~skarbez/comp575
- Slides and web notes will be posted online over the weekend

What is "rasterization"?

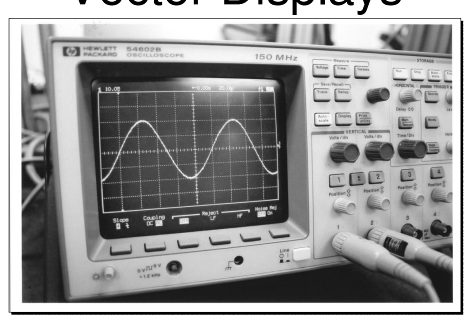
- Definition 1: The process of converting a vector image (shapes) to a raster image (dots)
- Why?
  - Dots are the only things modern displays can understand!

Cathode Ray Tube



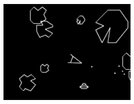
Karl Ferdinand Braun, 1897
Image courtesy of Wikipedia

Vector Displays

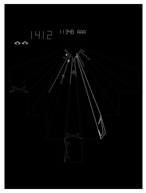


HP Oscilloscope

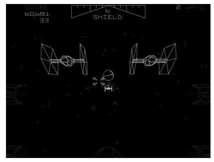
Vector Displays



Asteroids, 1979



Tempest, 1981



Star Wars, 1983

How Vector Displays Work

- They just draw line segments
  - User/computer:
    - Define start and end points
  - Display:
    - Move electron gun to start point
    - Turn electron gun on
    - Move electron gun to end point
    - Turn electron gun off

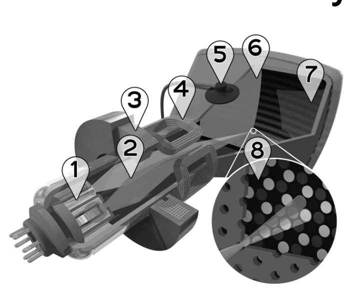
Advantages of Vector Displays

- Require very little memory
  - Important on a 64K system
  - Conceptually very simple
  - No <u>aliasing</u> of lines/curves
    - We'll come back to this later
  - No fixed timing
    - Refresh rate can be very high

Disadvantages of Vector Displays

- Really just one: Can only draw line segments
  - Time needed to draw a screen increases with number of lines drawn
    - If most of the image isn't black, you won't be able to finish drawing it in time!

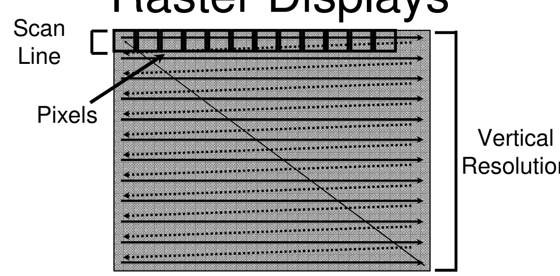
Cathode Ray Tube



1. Electron guns
2. Electron beams
3. Focusing coils
4. Deflection coils
5. Anode connection
6. Mask for separating beams for red, green, and blue part of displayed image
7. Phosphor layer with red, green, and blue zones
8. Close-up of the phosphor-coated inner side of the screen

Image courtesy of Wikipedia

Raster Displays



Disadvantages of Raster Displays

- Need a fairly large amount of memory
  - Draws the whole screen “at once”
  - Need a frame buffer that can hold the information for a whole image
- Aliasing!

Aliasing

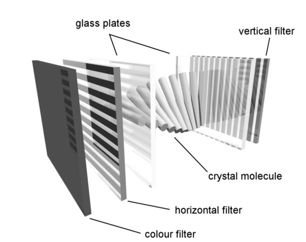
- This is what causes “jaggies”
- A signal processing problem:
  - The incoming signal (the desired image) can only be sampled at pixel centers on the display
  - Demonstrations

Advantages of Raster Displays

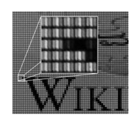
- Refresh rate is not dependent on the amount of pixels drawn
  - Very important for drawing any images that are not 90% black

Other raster displays

- Liquid Crystal Displays (LCD)



A single subpixel of an LCD display



Close-up of pixels on an LCD display

What is “rasterization”? (Part 2)

- Virtually all displays used today are raster displays
- So, technically, anything that produces an image on a screen “rasterizes”
- Definition 2: The rendering method used by current graphics cards

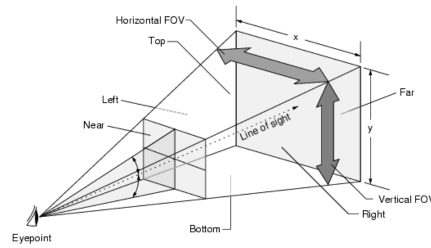
How do we draw an image?

- Start with geometry
  - We have some 3D model and/or environment in the system, and we want to draw it on the screen
- Problem: The display is, virtually always, only 2D
  - Need to transform the 3D model into 2D

3D to 2D (Projection)

- Problem: The display is, virtually always, only 2D
  - Need to transform the 3D model into 2D
- We do this with a virtual camera
  - Represented mathematically by a 3x4 projection (or P) matrix

3D to 2D (Projection)



$$\text{Aspect Ratio} = \frac{y}{x} = \frac{\tan(\text{vertical FOV}/2)}{\tan(\text{horizontal FOV}/2)}$$

Shading

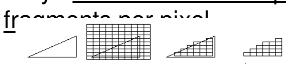
- Problem: How do we determine the color of a piece of geometry?
- In the real world, color depends on the object's surface color and the color of the light
  - It is the same way in computer graphics
- "Shading" is the process by which color is assigned to geometry

Clipping

- Problem: The camera doesn't see the whole scene
  - In particular, the camera might only see parts of objects
- Solution: Find objects that cross the edge of the viewing volume, and "clip" them
  - Clip: Cut a polygon into multiple parts, such that each is entirely inside or outside the display area

Rasterization

- Problem: How to convert these polygons on a plane to pixels on a screen?
- Have to figure out which pixels a polygon covers
  - A part of a polygon that covers one pixel is called a "fragment
    - Why? <u>There can be multiple fragments per pixel</u>



Rasterizing a triangle

Fragment Processing

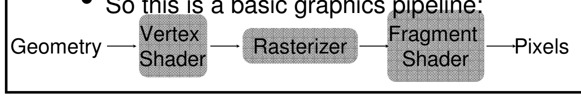
- Problem: How do we know which fragment to use to color a given pixel?
- Need to know which fragment is in front (not counting transparency)

Rendering process

- Start: Geometric model
- Compute color of geometry (Shading)
  - Based on lighting and surface color
- Project geometry (Projection)
- Clip geometry (Clipping)
- Generate fragments from geometry (Rasterization)
- Compute pixel colors from fragments (Fragment processng)
- End: Display pixels

Rendering pipeline

- Shading, projection, and clipping all operate on the original geometry
  - Combined into one unit, or “shader”
    - “Vertex shader” or “geometry shader”
- So this is a basic graphics pipeline:



Class Schedule

- Next Tuesday
  - Course Overview II: Real Cameras, Vision, and Ray Tracing
- Next Thursday
  - Math Basics: Matrices and Vectors
- Week 3
  - Transforms (2D & 3D)

Class Schedule

- Week 4
  - OpenGL
- Week 5
  - Geometry & Modeling
  - Lighting & Shading
- Week 6
  - Lighting & Shading
  - Clipping

Class Schedule

- Week 7
  - Clipping
  - Line Drawing
- Week 8
  - Polygon Drawing
  - Hidden Surface Removal
- Week 9
  - Texture Mapping