Raster Graphics

also today: processing tutorial, 2D project requirements

CS 4300/5310

Computer Graphics

ANNOUNCEMENTS

Course Survey: Due Today!

- Due tonight at 11:59pm

- Who are you?
- What do you want to learn?
- What level of experience?

- Aggregate results announced Tuesday

Assignment 1

- Due Tuesday, 11:59pm

- Getting help
  - Questions on Piazza
  - Form study groups
  - My office hours: 2 – 4pm Tuesday
    - …may be too late for getting started questions
  - I don't stay up late answering email!

Looking forward…

- 2D Project Proposal: January 22nd
- 2D Project Deadline: February 5th

Interested in Research?

- I am looking for research help!
- Undergrad **and** Masters
  - Potentially paid (!), we can discuss
  - Topics
    - Game programming (2D, Flash with Flixel)
    - Web programming (Python, Google App Engine)
    - Game analysis
- If interested: email me, attach your resume
  - List relevant course experience

RASTERIZATION

Graphics Problems

- Modeling
  - 2D and 3D representation
  - Curved surfaces
  - Procedural techniques
- **Rendering**
  - Realism
  - Speed
  - Non-realism
- Animation
  - Illusion of life
  - Motion capture
  - Keyframing
  - Physical simulation

Pixel Art

Yuriy Gusev

What is a Pixel?

- **Pic**ture **El**ement
- Not a little square.

Pixel Representation

- Pros
- Cons

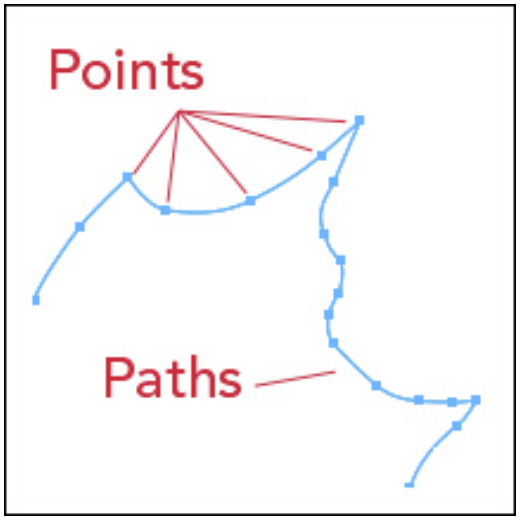
Pixel Representation

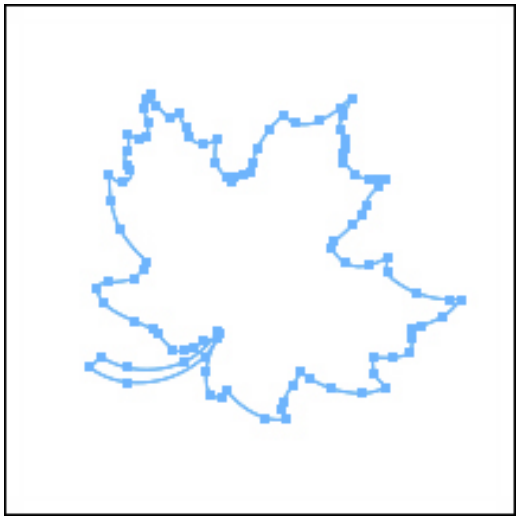
- Pros
  - Easy to read (and write) file type
  - Good for photo touch-ups
  - Lots of detail
- Cons
  - Resizing is horrible
  - Hard to significantly change image
  - Large file sizes

Vector Representation

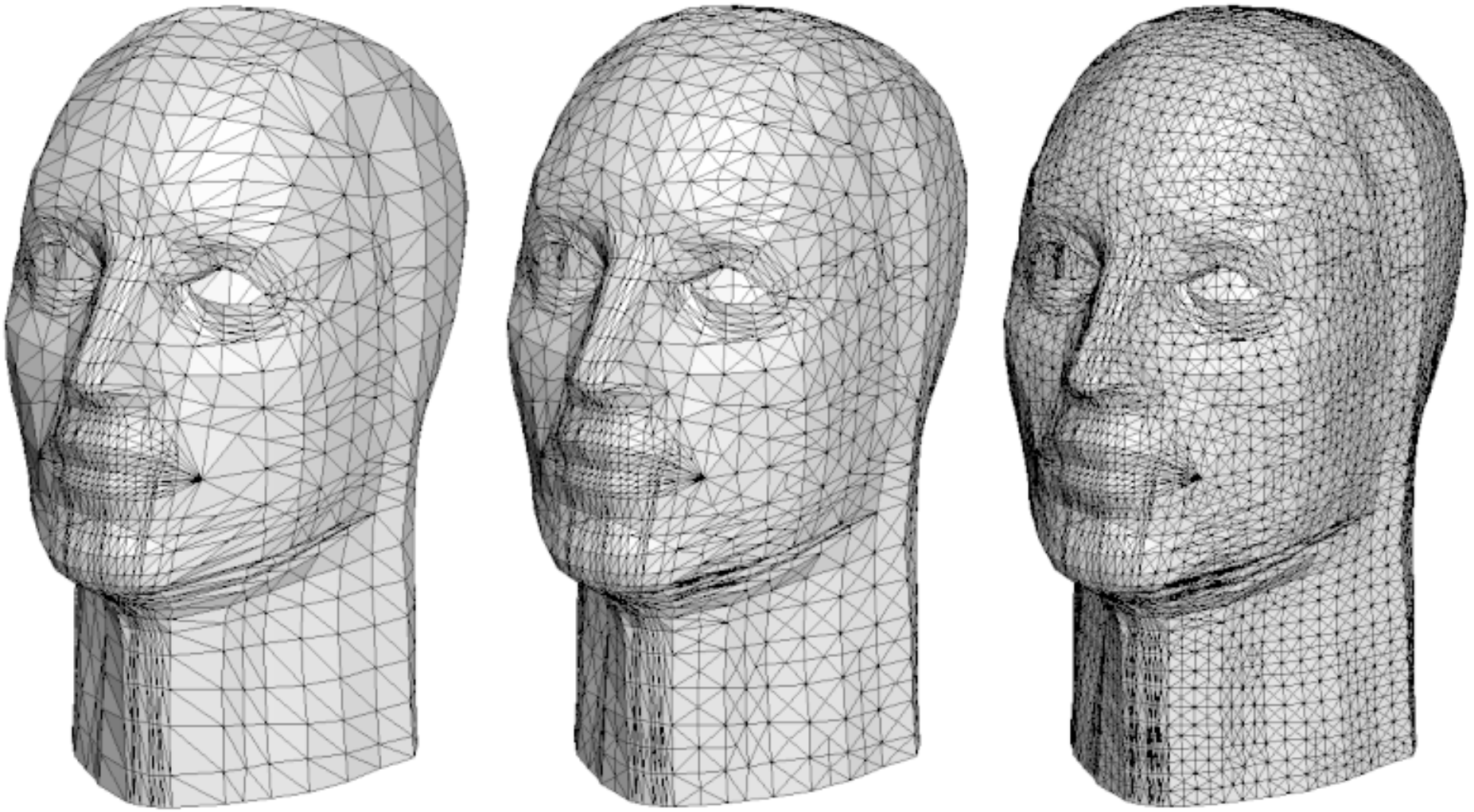
Vector Representation

- Pros
  - Easy to change objects
  - Convert to pixel art (**rasterize**)
  - Smaller file sizes (sometimes…)
- Cons
  - Cannot use for photos
  - Color blending

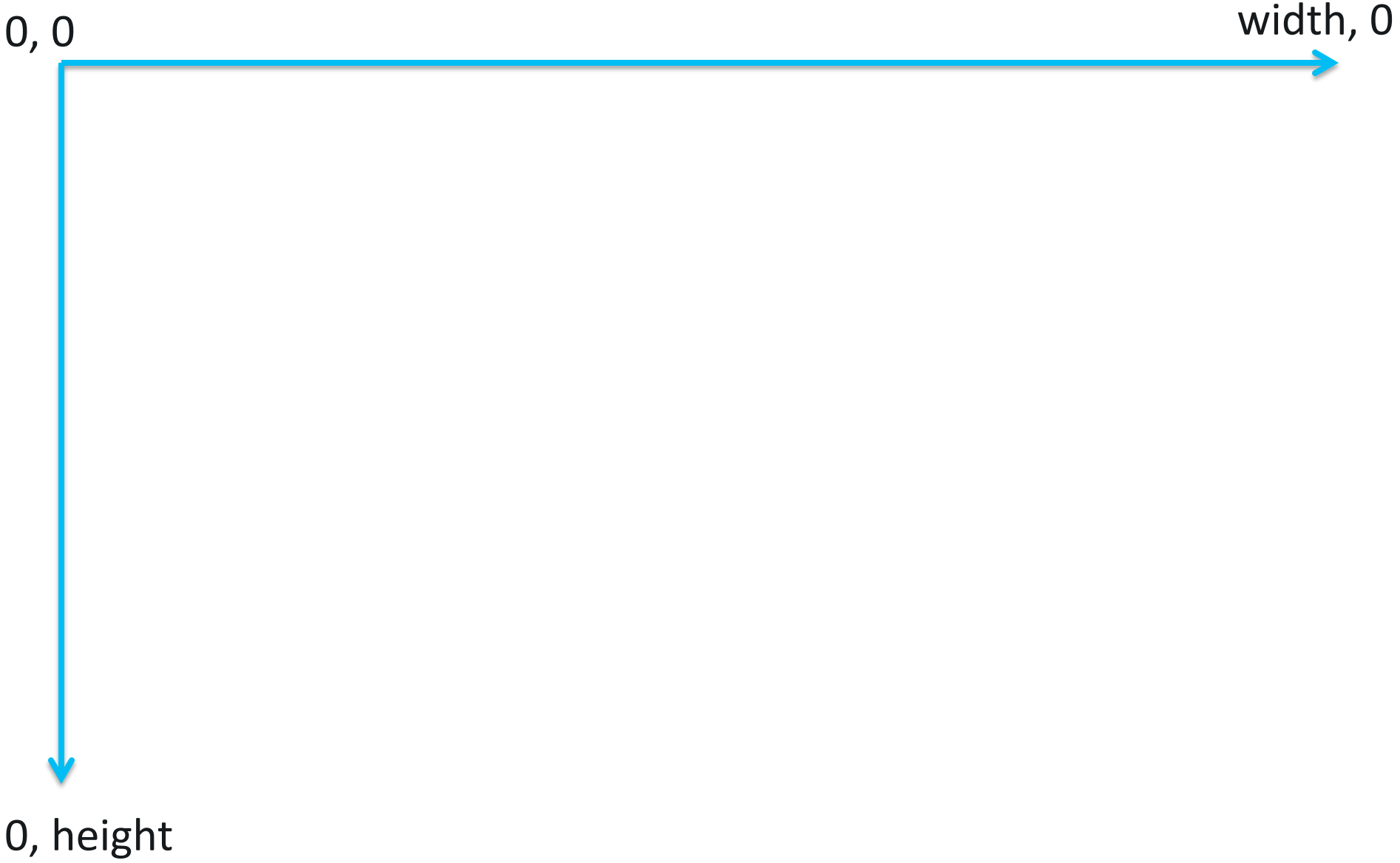




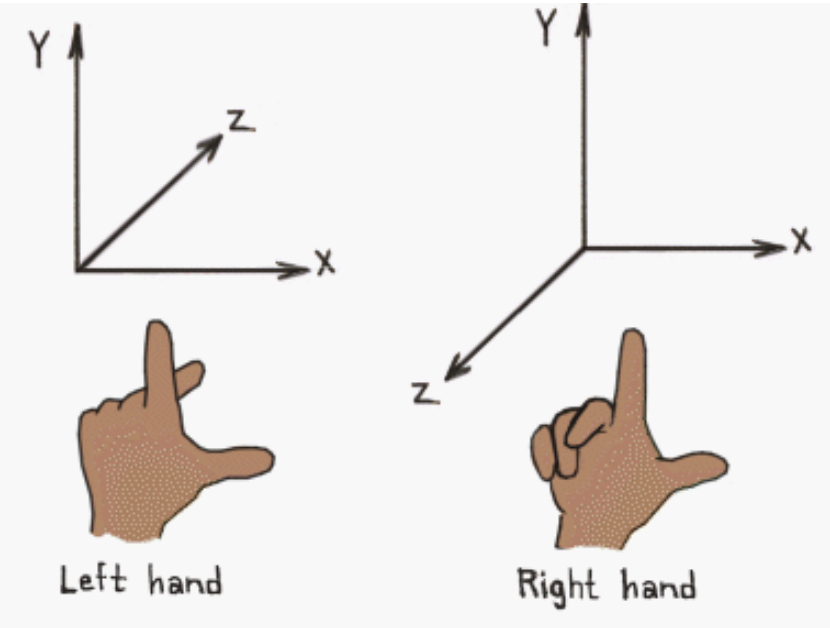
Aside: What about Art in 3D?



Aside: Aside: Coordinate Systems

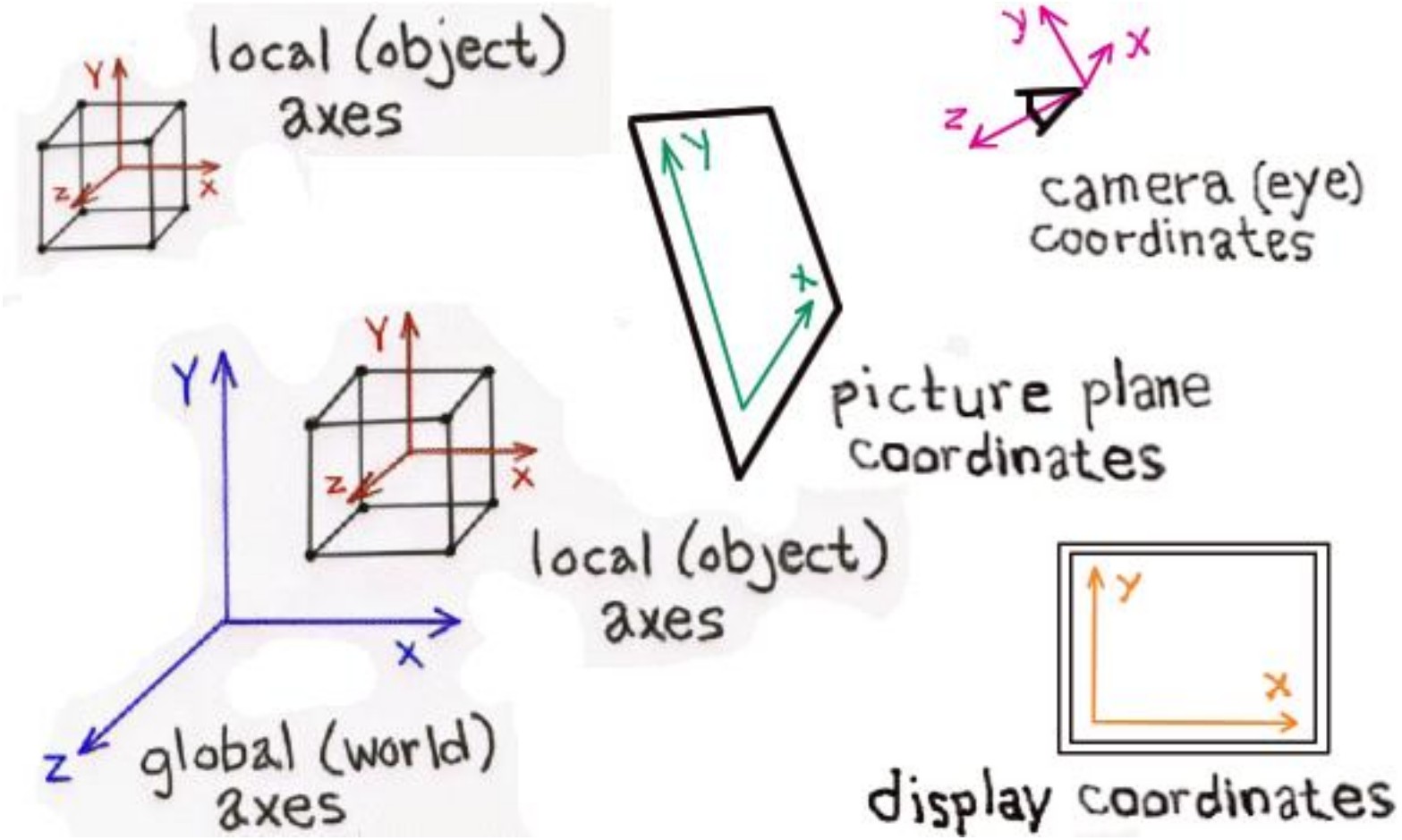


Aside: Aside: Coordinate Systems



http://viz.aset.psu.edu/gho/sem_notes/3d_fundamentals/html/3d_coordinates.html

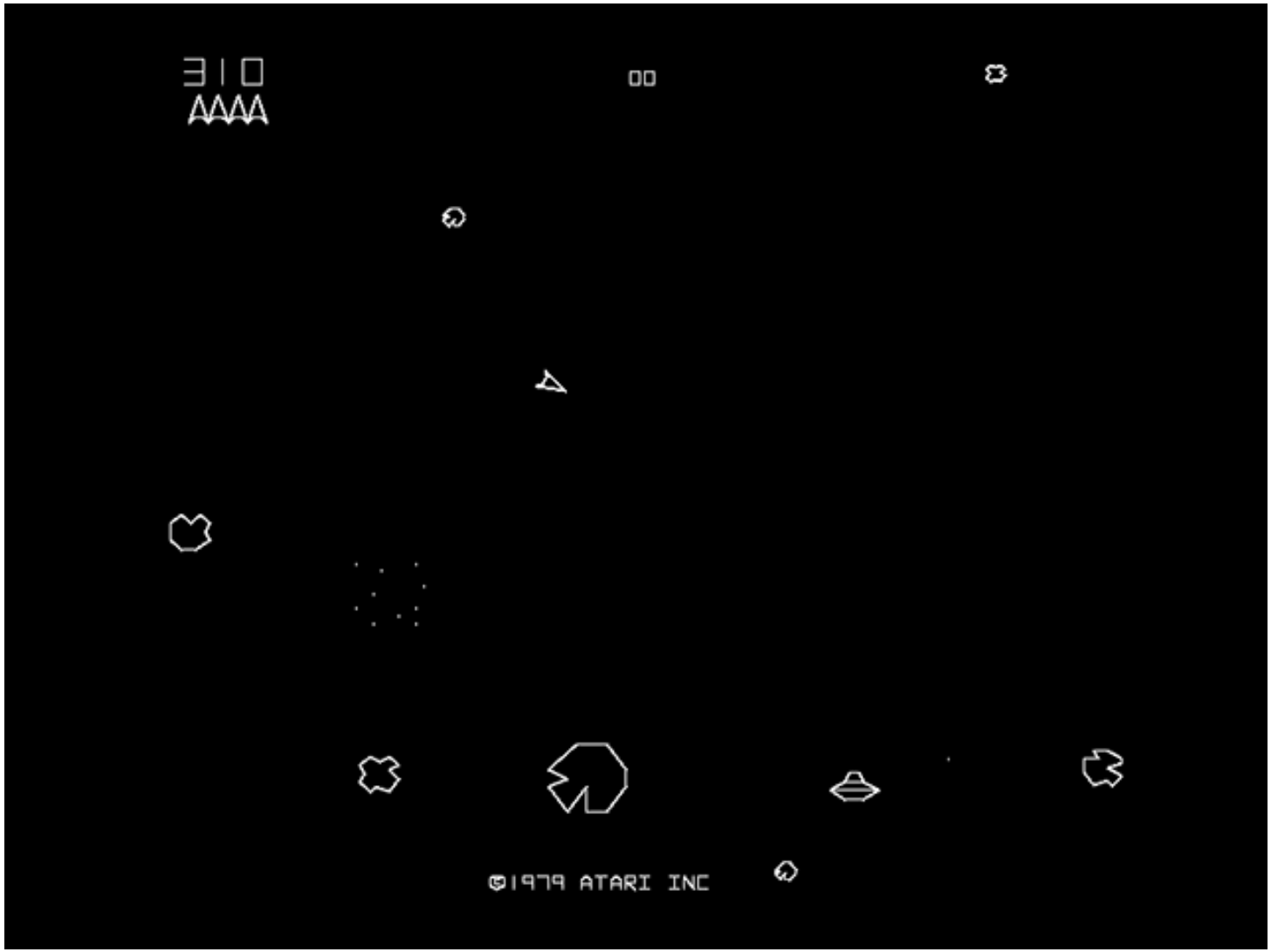
Aside: Aside: Coordinate Systems



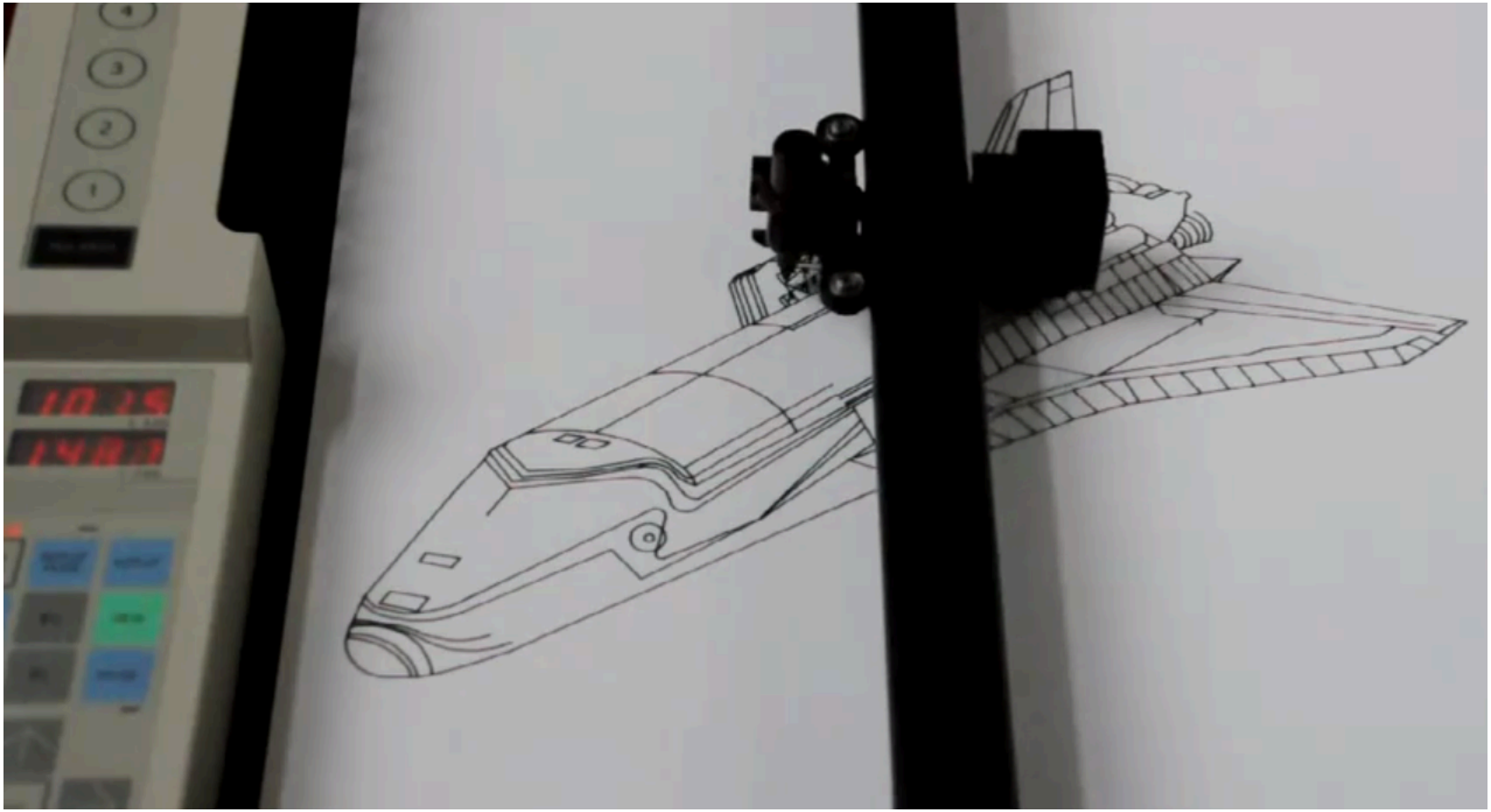
Raster Graphics

- Raster Images
  - Image stored as an array of pixels (usually RGB)
- Raster Devices
  - Displays that show images as pixels
  - Input devices that convert images into pixels

Vector Displays and Printers



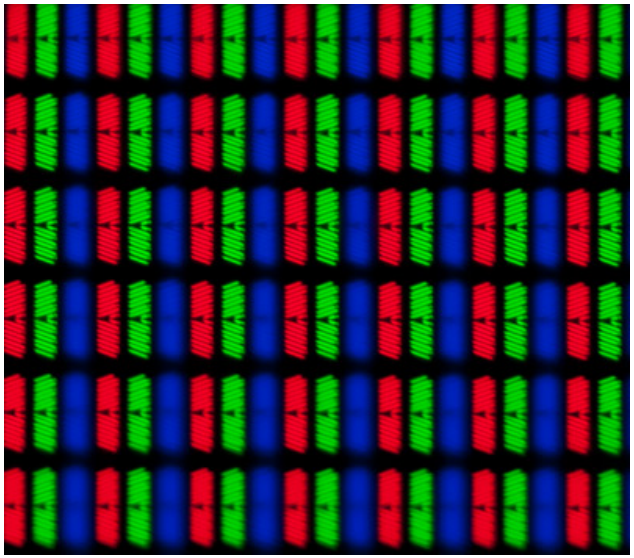
Vector Displays and Printers

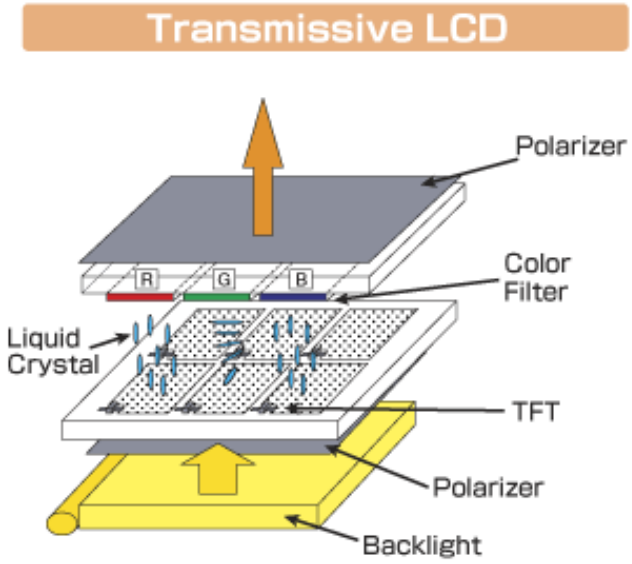


https://www.youtube.com/watch?v=iziP0cQhOFY

Raster Displays

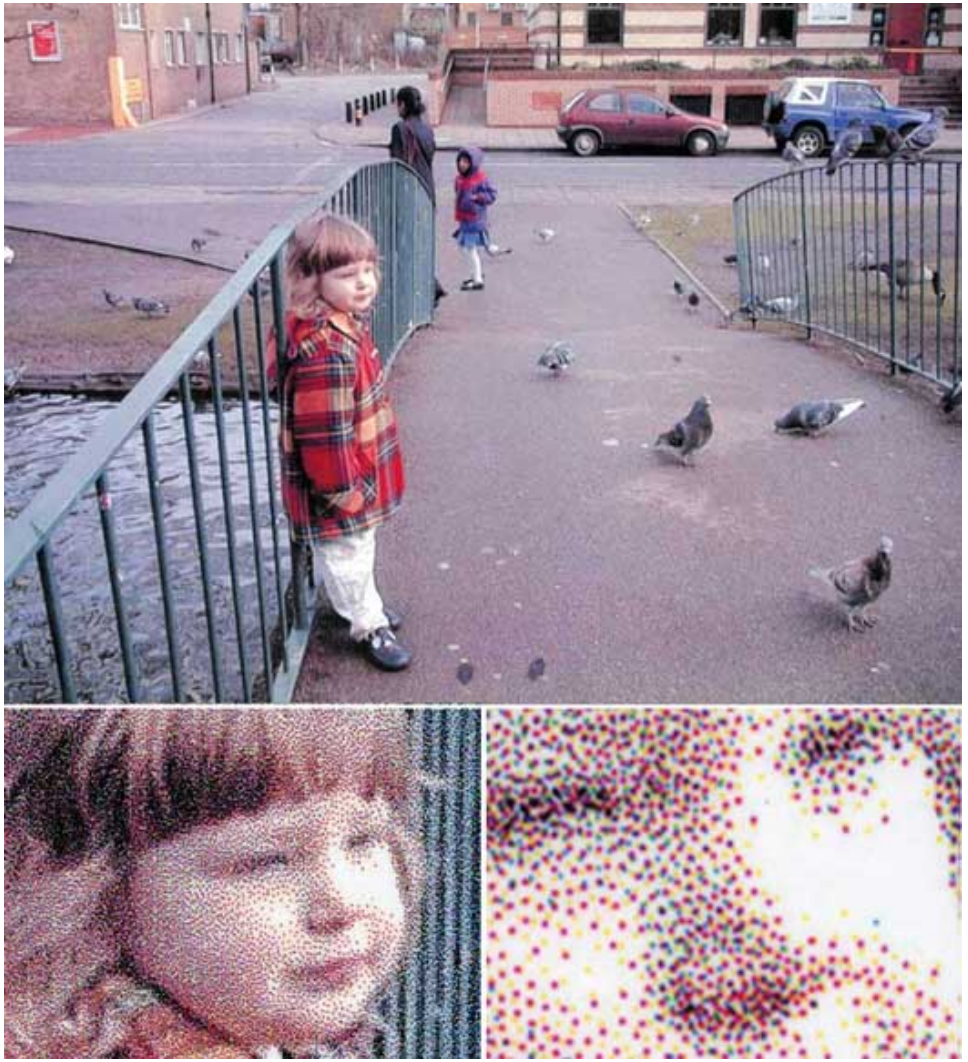
- Sub-pixels
  - Red
  - Green
  - Blue
- Emissive vs. Transmissive
- LCD
  - Voltage applied to liquid crystals twists light to pass through front polarizer





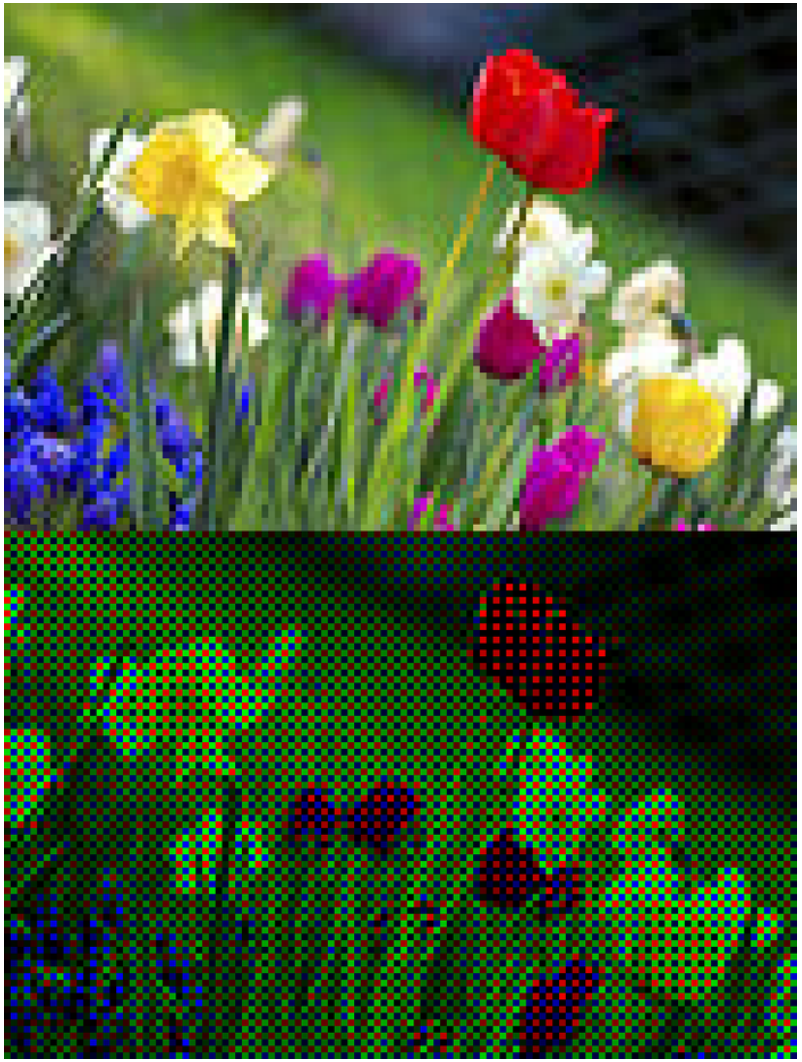
Raster Printers

- Binary images
- Dot patterns for color intensity (dithering)
- Sub-“pixels”
  - C
  - M
  - Y
  - K



Input Rasterization

- Capturing continuous signal into discrete pixels
- Charge-coupled devices (CCDs) record amount of light hitting each pixel as number
- Bayer Mosaics



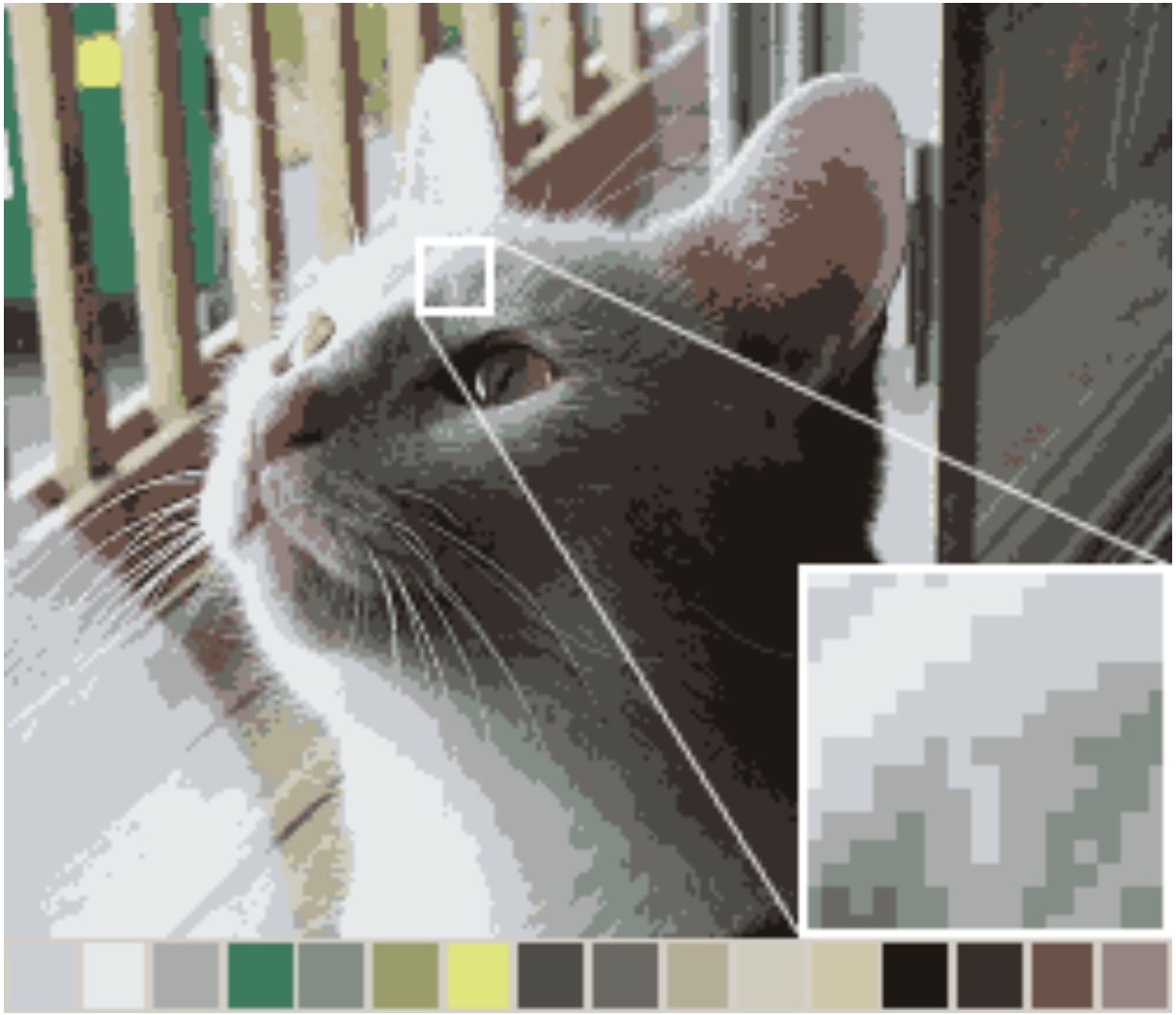
Pixel Values

- 1-bit grayscale
  - Black and white
- 8-bit RGB fixed-range
  - Web safe
- 12- to 14-bit fixed range
  - photography
- 16-bit float RGB
  - High dynamic range (HDR)
  - Real-time rendering intermediate
- 32-bit float RGB
  - High dynamic range

Clipping Artifact

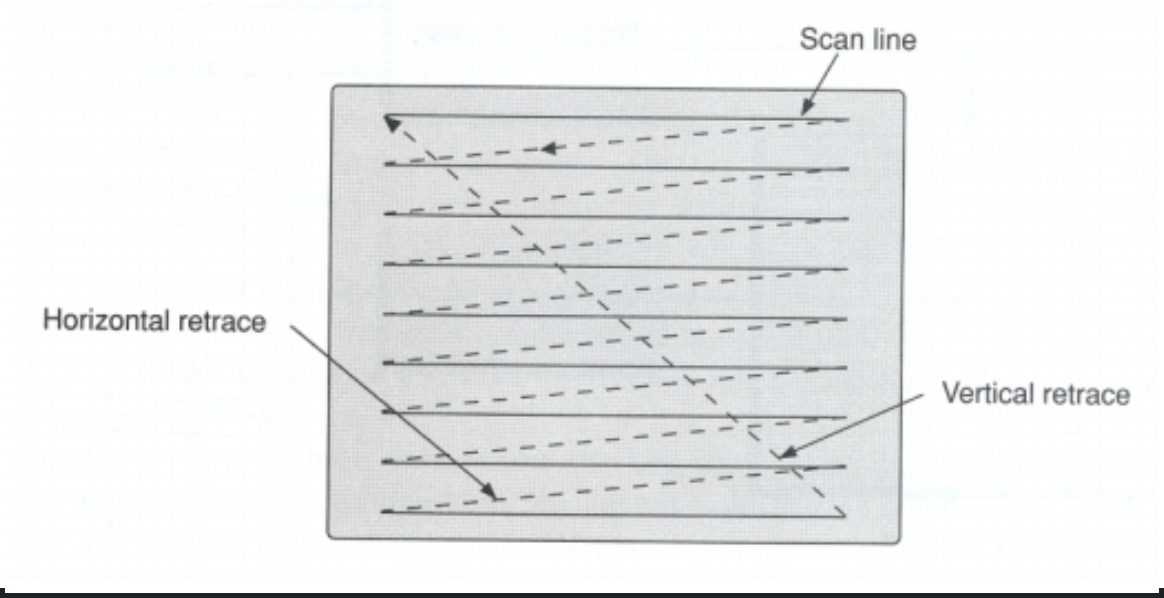
Clipping Artifact

Banding Artifacts



Frame Buffers

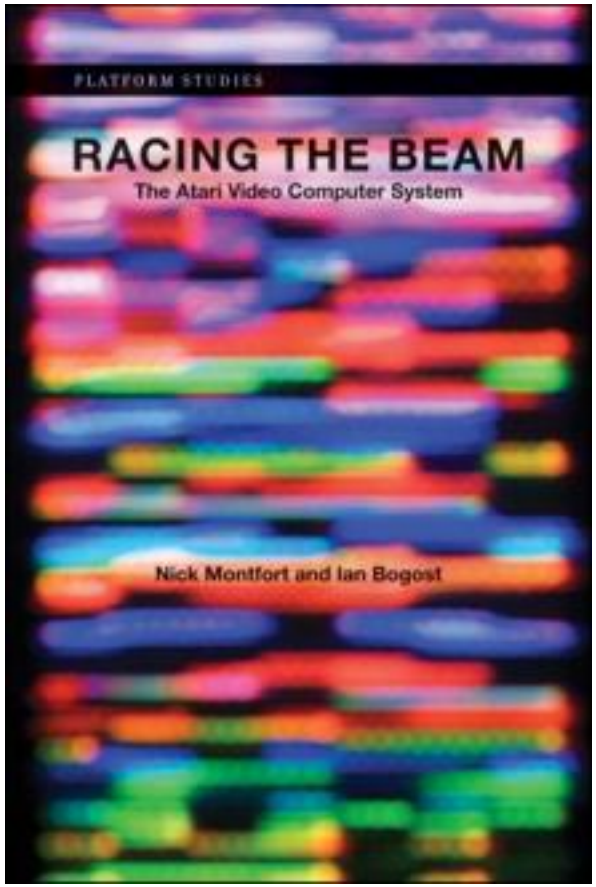
- Result of all rendering stored in hardware
- 2D grid of pixel values
  - Direct: all colors stored individually
  - Color lookup: store 8-bit index in framebuffer



Atari VCS: Racing the Beam

- Why do we have frame buffers?

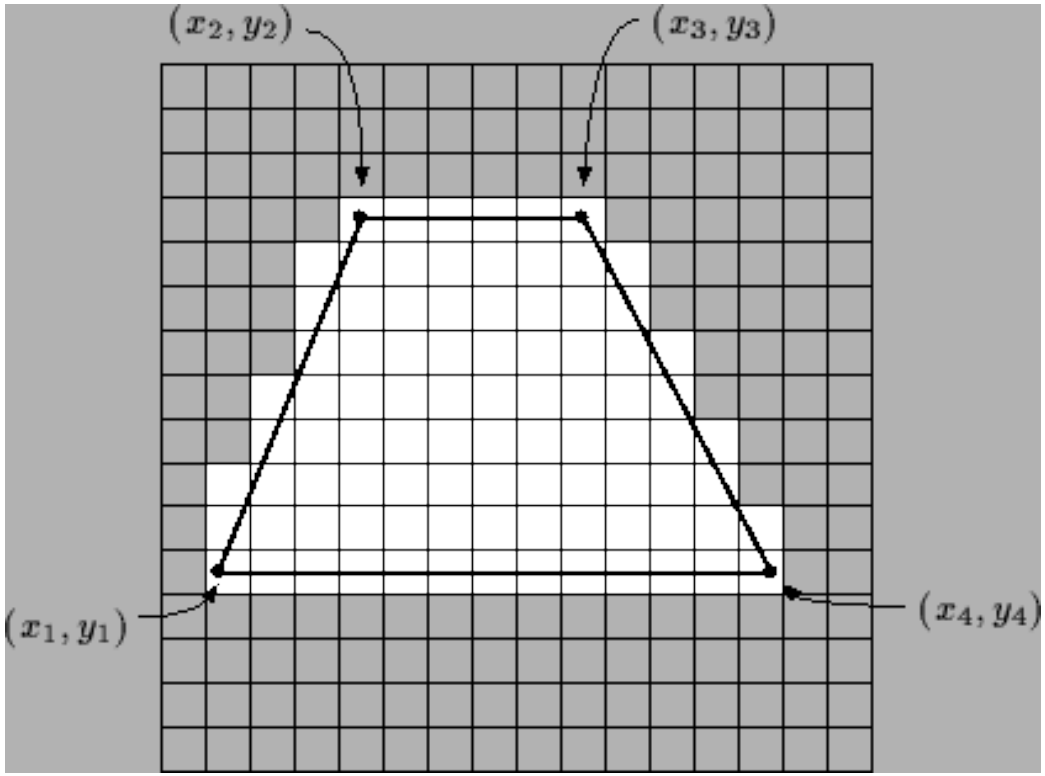
- What if we didn't have one?



Rasterization: Part of the Graphics Pipeline

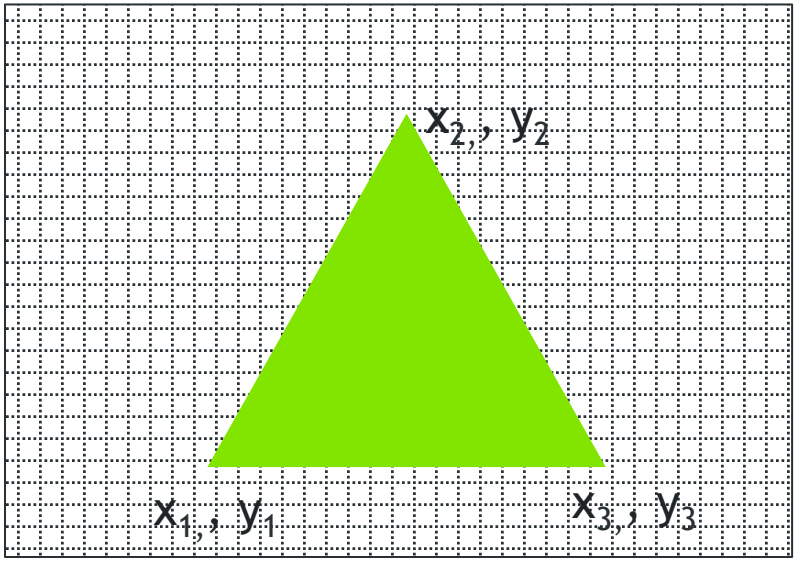
- 3D Primitives
- Modeling Transformation
- Lighting
- Viewing Transformation
- Clipping
- Projection to 2D space
- **Rasterization**
- Pixel Shading
- Frame Buffer

- Convert objects into pixels



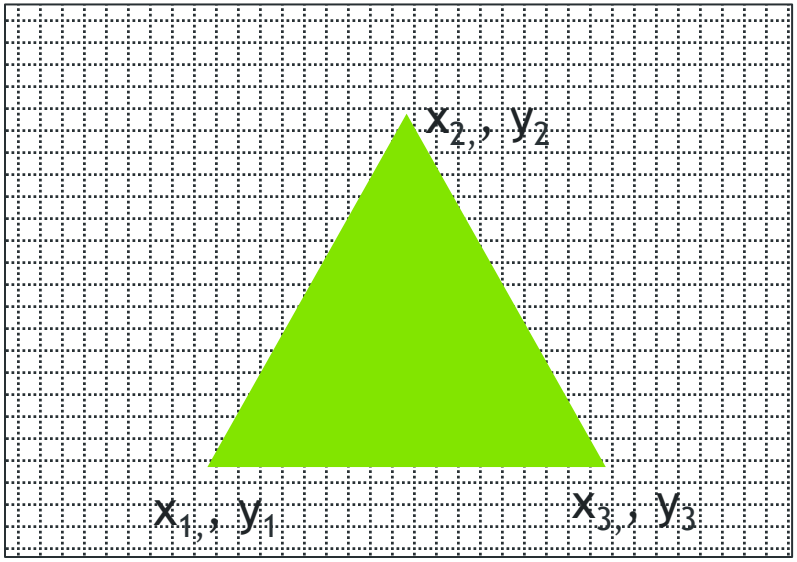
Rasterization

- How do we figure out what pixels in this triangle are “on”?



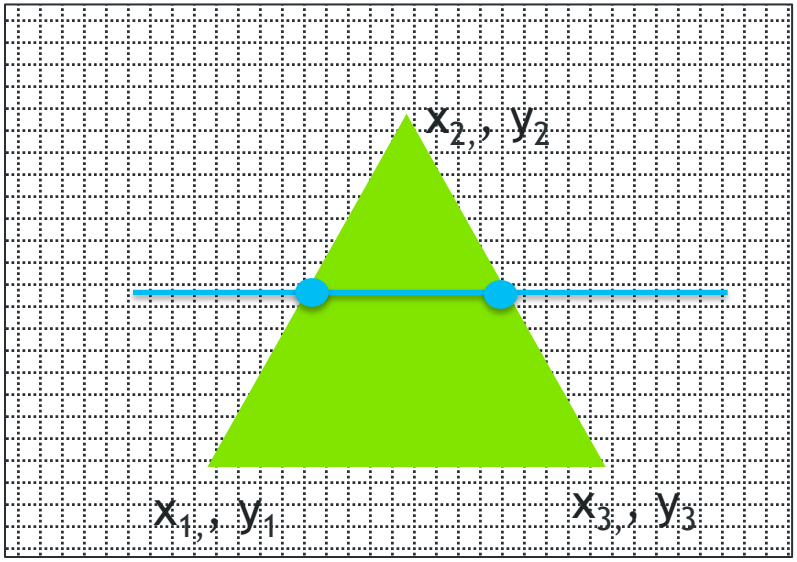
Rasterization: Brute Force

- Check if each pixel is inside or outside the triangle, record result
  - How can we optimize this?



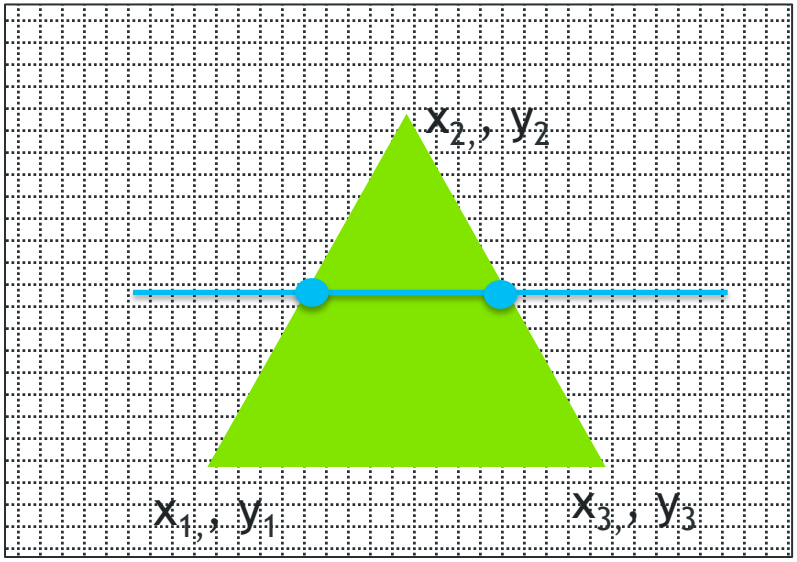
Rasterization: Scanline

- Pass line across screen
- Check for intersection with edges
- Span in between intersection points = filled



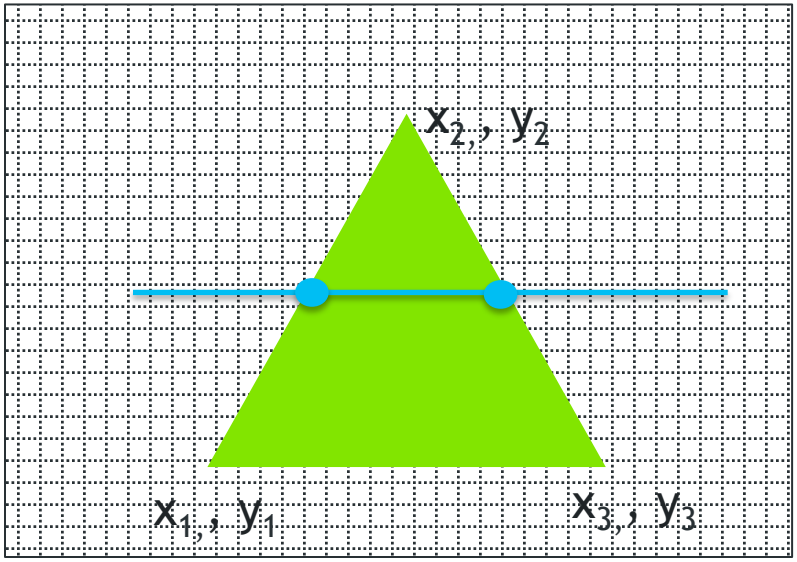
Rasterization: Scanline

- Maintain Active Edge Table
  - Sort edges by scanline direction
  - Run scanline down image, store intersected edges in active edge table
  - Compute spans
  - Sort active edge table by minX, fill in spans between edges



Rasterization: Quadtree

- Recursively subdivide the screen in to quarters
- Check intersection of triangles with each quad
  - If it intersects, divide section into more quarters
  - If it doesn't intersect, ignore!



coding live

PROCESSING TUTORIAL

2D PROJECT

Goals

- Build an interactive 2D graphics program
- Implement 2D transformations
  - Translation (i.e. movement)
  - Scaling
  - Rotation
- Implement 2D picking (i.e. selecting drawn objects on screen)
- Writing about software features
  - Planning your project
  - User manual

Proposal Requirements

- Aim for 4 – 6 pages
  - 1500 – 3000 words
  - Well-formed, full sentences (limit bullet points)
  - Pictures very, very welcome!
- Describing software **features** in detail
- Focus (for now) on what your program will do, not how to do it

Proposal Requirements

- Project Timeline
  - Who is responsible for what features
  - About how long are you estimating they will take?
- Soft copy due online: January 22nd
- Bring hard copy to class: January 24th

Group Formation Recommendations

- Goals
- Desired development environment
- Level of experience
  - Grad students and undergrad students?

GROUP PROJECT BRAINSTORMING