Lecture 2:

Digital Drawing

Computer Graphics and Imaging
UC Berkeley CS184/284A, Spring 2017
Drawing Machines
Oscilloscope
Oscilloscope Art

Jerobeam Fenderson
https://www.youtube.com/watch?v=rtR63-ecUNo
CNC Sharpie Drawing Machine

Aaron Panone with Matt W. Moore

Laser Cutters
Television

July 20, 1969, Moon Landing on TV
Raster Displays (Pixel Arrays)
Television - Raster Display CRT

Cathode Ray Tube

Raster Scan
(modulate intensity)
Frame Buffer

DAC = Digital to Analog Convertors

Analog

Digital

Image = 2D array of colors
LCD Flat Panel Display

Principle: block or transmit light by twisting polarization

Illumination from backlight (e.g. fluorescent or LED)

Intermediate intensity levels by partial twist

Fundamentally raster technology
Greenland or right-whale, he is the best existing authority. But Scoresby knew nothing and says nothing of the great sperm whale, compared with which the Greenland whale is almost unworthy mentioning. And here be it said, that the Greenland whale is an usurper upon the throne of the seas. He is not even by any means the largest of the whales. Yet, owing to the long priority of his claims, and the profound ignorance which, till some seventy years back, invested the then fabulous or utterly unknown sperm-whale, and which ignorance to this present day still reigns in all but some few scientific retreats and whale-ports; this usurpation has been every way complete. Reference to nearly all the levitatory allusions in the great poets of past days, will satisfy you that the Greenland whale, without one rival, was to them the monarch of the seas. But the time has at last come for a new proclamation. This is Charing Cross; hear ye! good people all,—the Greenland whale is deposed,—the great sperm whale now reigneth!

There are only two books in being which at all pretend to put the living sperm whale before you, and at the same time, in the remotest degree succeed in the attempt. Those books are Beazly's and Bennett's, both in their time surpenses to English South-Sea whale-ships, and both honest and reliable men. The original matter touching the sperm whale to be found in their volumes is necessarily small; but so far as it goes, it is of excellent quality, though
Projection Displays: DLP

[Diagram of DLP technology]
Graphics Pipeline = Abstract Drawing Machine

OpenGL commands → Per-vertex ops → Rasterizer → Texturing → Per-fragment ops → Frame buffer ops → Pixels

Triangles, lines, points, images → pixels in the framebuffer
Shape Primitives

Example shape primitives (OpenGL)
Triangle Meshes
Triangle Meshes
Triangle Meshes
Why triangles?

• Most basic polygon
• Break up other polygons
• Optimize one implementation
• Triangles have unique properties
  • Guaranteed to be planar
  • Well-defined interior
• Well-defined method for interpolating values at vertices over triangle (barycentric interpolation)
Drawing a Triangle To The Framebuffer ("Rasterization")
What Pixel Values Approximate a Triangle?

Input: position of triangle vertices projected on screen
Output: set of pixel values approximating triangle
What Pixel Value For These Triangles?

Which of these triangles affect this pixel? What value should this pixel have for each triangle?
Let’s Start With A Simple Approach: Sampling
Rasterization as a Sampling Process

Evaluating a function at a point is sampling.

\[
\text{for( int } x = 0; x < \text{ xmax}; x++ ) \\
\text{ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
Consider Binary Function $\text{inside}(\text{tri}, x, y)$

$$\text{inside}(\text{t}, x, y) = \begin{cases} 
1 & (x, y) \text{ in triangle t} \\
0 & \text{otherwise}
\end{cases}$$
Binary Pixel Value By Testing \texttt{inside}(\texttt{tri},x,y)

Example:
Here I chose the coverage sample point to be at a point corresponding to the pixel center.

\(\text{Pixel (x,y)}\)

\(\text{Triangle covers sample, fragment generated for pixel} \)

\(\text{Triangle does not cover sample, no fragment generated} \)
Rasterization = Sampling inside \((\text{tri}, x, y)\)

Input: position of triangle vertices projected on screen

Output: set of pixels where center is inside triangle
Rasterization = Sampling inside \((\text{tri}, x, y)\)
Rasterization = Sampling inside(tri, x, y)
Rasterization = Sampling inside \((\text{tri}, x, y)\)
Implementation Detail: Sample Locations

Sample location for pixel \((x,y)\)

- \((0,0)\)
- \((w,0)\)
- \((0,h)\)
- \((w,h)\)

\((x+1/2, y+1/2)\)
Evaluating inside \((tri, x, y)\)
Triangle = Intersection of Three Half Planes
Each Line Defines Two Half-Planes

Implicit line equation

- \( L(x,y) = Ax + By + C \)

- On line: \( L(x,y) = 0 \)
- Above line: \( L(x,y) > 0 \)
- Below line: \( L(x,y) < 0 \)
Line Equation Derivation

\[ T = P_1 - P_0 = (x_1 - x_0, y_1 - y_0) \]
Line Equation Derivation

General Perpendicular Vector in 2D

Perp\((x, y) = (-y, x)\)
Line Equation Derivation

Line Normal Vector

\[ N = \text{Perp}(T) = (- (y_1 - y_0), x_1 - x_0) \]
Line Equation Derivation

\[ P_0 \]

\[ P_1 \]

\[ P = (x, y) \]

\[ V = P - P_0 = (x - x_0, y - y_0) \]
Line Equation

\[ L(x, y) = V \cdot N = -(x - x_0)(y_1 - y_0) + (y - y_0)(x_1 - x_0) \]
Line Equation Tests

\[ L(x, y) = V \cdot N > 0 \]
Line Equation Tests

\[ L(x, y) = V \cdot N = 0 \]
Line Equation Tests

\[ L(x, y) = V \cdot N < 0 \]
Point-in-Triangle Test: Three Line Tests

\[ P_i = (X_i, Y_i) \]
\[ dX_i = X_{i+1} - X_i \]
\[ dY_i = Y_{i+1} - Y_i \]

\[ L_i(x, y) = -(x - X_i) \, dY_i + (y - Y_i) \, dX_i \]
\[ = A_i \, x + B_i \, y + C_i \]

\[ L_i(x, y) = 0 : \text{point on edge} \]
\[ < 0 : \text{outside edge} \]
\[ > 0 : \text{inside edge} \]

Compute line equations from pairs of vertices
Point-in-Triangle Test: Three Line Tests

\[ P_i = (X_i, Y_i) \]

\[ dX_i = X_{i+1} - X_i \]

\[ dY_i = Y_{i+1} - Y_i \]

\[ L_i(x, y) = -(x - X_i) \, dY_i + (y - Y_i) \, dX_i \]

\[ = A_i \, x + B_i \, y + C_i \]

\[ L_i(x, y) = 0 : \text{point on edge} \]

\[ < 0 : \text{outside edge} \]

\[ > 0 : \text{inside edge} \]

\[ L_0(x, y) > 0 \]
Point-in-Triangle Test: Three Line Tests

\[ P_i = (X_i, Y_i) \]

\[ dX_i = X_{i+1} - X_i \]

\[ dY_i = Y_{i+1} - Y_i \]

\[ L_i(x, y) = -(x - X_i) dY_i + (y - Y_i) dX_i \]
\[ = A_i x + B_i y + C_i \]

\[ L_i(x, y) = 0 \] : point on edge
\[ < 0 \] : outside edge
\[ > 0 \] : inside edge

\[ L_1(x, y) > 0 \]
Point-in-Triangle Test: Three Line Tests

\[ P_i = (X_i, Y_i) \]

\[ dX_i = X_{i+1} - X_i \]
\[ dY_i = Y_{i+1} - Y_i \]

\[ L_i(x, y) = -(x - X_i) \cdot dY_i + (y - Y_i) \cdot dX_i \]
\[ = A_i x + B_i y + C_i \]

\[ L_i(x, y) = 0 \text{ : point on edge} \]
\[ < 0 \text{ : outside edge} \]
\[ > 0 \text{ : inside edge} \]
Point-in-Triangle Test: Three Line Tests

Sample point \( s = (sx, sy) \) is inside the triangle if it is inside all three lines.

\[
inside(sx, sy) = \text{inside}(sx, sy) = \\
L_0(sx, sy) > 0 \land \\
L_1(sx, sy) > 0 \land \\
L_2(sx, sy) > 0; 
\]

Note: actual implementation of \( inside(sx, sy) \) involves \( \leq \) checks based on edge rules.
Edge Cases (Literally)

Is this sample point covered by triangle 1, triangle 2, or both?
OpenGL/Direct3D Edge Rules

When sample point falls on an edge, the sample is classified as within triangle if the edge is a “top edge” or “left edge”.

Top edge: horizontal edge that is above all other edges.

Left edge: an edge that is not exactly horizontal and is on the left side of the triangle. (triangle can have one or two left edges).

Source: Direct3D Programming Guide, Microsoft
Incremental Triangle Traversal (Faster?)
Modern Approach: Tiled Triangle Traversal

Traverse triangle in blocks
Test all samples in block in parallel

Advantages:
- Simplicity of wide parallel execution overcomes cost of extra point-in-triangle tests (most triangles cover many samples, especially when super-sampling)
- Can skip sample testing work: entire block not in triangle ("early out"), entire block entirely within triangle ("early in")

All modern GPUs have special-purpose hardware for efficient point-in-triangle tests
Signal Reconstruction on Real Displays
Real LCD Screen Pixels (Closeup)

Notice R,G,B pixel geometry! But in this class, we will assume a colored square full-color pixel.
Aside: What About Other Display Methods?

Color print: observe half-tone pattern
Assume Display Pixels Emit Square of Light

Each image sample sent to the display is converted into a little square of light of the appropriate color: (a pixel = picture element)

* LCD pixels do not actually emit light in a square of uniform color, but this approximation suffices for this discussion
So, If We Send The Display This Sampled Signal
The Display Physically Emits This Signal
Compare: The Real Coverage Function
What’s Wrong With This Picture?

Jaggies!
Jaggies (Staircase Pattern)

Is this the best we can do?
Discussion: What Value Should a Pixel Have?

Potential topics for your pair discussion:

• Ideas for “higher quality” pixel formula?
• What are all the relevant factors?
• What’s right/wrong about point sampling?
• Why do jaggies look “wrong”?
Things to Remember

Drawing machines

• Why framebuffers and raster displays?
• Why triangles?

We posed rasterization as a 2D sampling process

• Test a binary function inside (triangle, x, y)
• Evaluate triangle coverage by 3 point-in-edge tests
• Finite sampling rate causes “jaggies” artifact
  (next time we will analyze in more detail)
Acknowledgments

Thanks to Kayvon Fatahalian, Pat Hanrahan, Mark Pauly and Steve Marschner for slide resources.