Lecture 2: Digital Drawing

Computer Graphics and Imaging UC Berkeley CS184/284

Today: Drawing Triangles to the Screen by Sampling



CS184/284A

Drawing Machines

CNC Sharpie Drawing Machine



Aaron Panone with Matt W. Moore http://44rn.com/projects/numerically-controlled-poster-series-with-matt-w-moore/

Laser Cutters



Oscilloscope



Cathode Ray Tube



CS184/284A

Oscilloscope Art



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



Television - Raster Display CRT





Cathode Ray Tube

CS184/284A



Raster Scan (modulate intensity)

Frame Buffer: Memory for a Raster Display



Image = 2D array of colors

CS184/284A

A Sampling of Different Raster Displays

LED Array Display



Light emitting diode array

CS184/284A



LED Array Display



BAMPFA display in Berkeley

CS184/284A

Flat Panel Displays

HIJKU MNOPOR 456789:;<=>?@ABC



CS184/284A

Color LCD, OLED, ...







Flat Panel Displays



iPhone 6S

CS184/284A

Smartphone screen pixels under microscope

Ren Ng

Galaxy S5



LCD vs OLED Displays

LCD

Cover Glass Linear Polarizer Color Filter Glass Liquid crystal **Glass TFT Backplane Linear Polarizer** Backlight



OLED

Cover Glass

Circular Polarizer

Encapsulated Glass

Glass TFT Backplane with OLED

Heat Sink

Liquid Crystal Display

LCD pixels filter (block) light from uniform backlight; OLED pixels emit light

CS184/284A



Organic Light Emitting Diode Display

Digital Micromirror Device (DMD/DLP)









John Jackson, University of Rochester

CS184/284A

Larry Hornbeck

Digital Micromirror Device (DMD/DLP)



CS184/284A

Texas Instruments



Electrophoretic (Electronic Ink) Display

amazonkindle

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 leviathanic 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 Beale's and Bennett's; both in their time surgeons to English South-Sea whale-ships, and both exact 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





CS184/284A

Wikimedia Commons —<u>Senarclens]</u>

Drawing to Raster Displays

Polygon Meshes



CS184/284A



Triangle Meshes



CS184/284A

Triangle Meshes



CS184/284A



Shape Primitives



Example shape primitives (OpenGL)

Graphics Pipeline = Abstract Drawing Machine



CS184/284A

triangles, lines, points images





pixels in the framebuffer

Triangles - Fundamental Area Primitive

- 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?



CS184/284A

approximating triangle

Today, Let's Start With A Simple Approach: Sampling

Sampling a Function

Evaluating a function at a point is sampling.

We can discretize a function by periodic sampling.

for(int x = 0; x < xmax; x++) output[x] = f(x);

Sampling is a core idea in graphics. We'll sample time (1D), area (2D), angle (2D), volume (3D) ...

We'll sample N-dimensional functions, even infinite dimensional functions.

Let's Try Rasterization As 2D Sampling



CS184/284A

Sample If Each Pixel Center Is Inside Triangle



CS184/284A

Sample If Each Pixel Center Is Inside Triangle



CS184/284A



Define Binary Function: inside(tri,x,y)

1 inside(t,x,y) 0 otherwise

(x,y) in triangle t

Rasterization = Sampling A 2D Indicator Function

for(int x = 0; x < xmax; x++) for(int y = 0; y < ymax; y++) Image[x][y] = f(x + 0.5, y + 0.5);

Rasterize triangle tri by sampling the function f(x,y) = inside(tri,x,y)



Implementation Detail: Sample Locations



Sample location for pixel (x,y)

Evaluating inside(tri,x,y)

Triangle = Intersection of Three Half Planes



CS184/284A

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 Tangent Vector



CS184/284A





CS184/284A

CS184/284A

Line Equation

CS184/284A

Line Equation Tests

CS184/284A

Line Equation Tests

CS184/284A

Line Equation 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

Compute line equations from pairs of vertices

CS184/284A

$$P_i = (X_{i, Y_i})$$

$$dX_i = X_{i+1} - X_i$$
$$dY_i = Y_{i+1} - Y_i$$

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

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

 $L_0(x, y) > 0$

CS184/284A

$$P_i = (X_{i, Y_i})$$

$$dX_i = X_{i+1} - X_i$$
$$dY_i = Y_{i+1} - Y_i$$

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

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

 $L_l(x, y) > 0$

CS184/284A

$$P_i = (X_{i, Y_i})$$

$$dX_i = X_{i+1} - X_i$$
$$dY_i = Y_{i+1} - Y_i$$

$$L_i(x, y) = -(x - X_i) dY_i + (y - Y_i) dX_i$$

= $A_i x + B_i y + C_i$

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

 $L_2(x, y) > 0$

CS184/284A

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

inside(sx, sy) = $L_0(sx, sy) > 0 \&\&$ $L_1(sx, sy) > 0 \&\&$ $L_2(sx, sy) > 0;$

Note: actual implementation of *inside*(*sx*,*sy*) **involves** ≤ **checks based on edge rules**

Some Details

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"

Source: Direct3D Programming Guide, Microsoft

CS184/284A

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)

Incremental Triangle Traversal (Faster?)

CS184/284A

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

CS184/284A

Signal Reconstruction on **Real Displays**

Real LCD Screen Pixels (Closeup)

iPhone 6S

Notice R,G,B pixel geometry! But in this class, we will assume a colored square full-color pixel.

CS184/284A

Galaxy S5

Aside: What About Other Display Methods?

Color print: observe half-tone pattern

CS184/284A

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 pixel on laptop

* LCD pixels do not actually emit light in a square of uniform color, but this approximation suffices for our current discussion

So, If We Send The Display This Sampled Signal

CS184/284A

- 0 0
- 0 0
- 0 0
- 0 0
- 0 0
- 0
- 0
- • • •
- 0 0

The Display Physically Emits This Signal

CS184/284A

Compare: The Continuous Triangle Function

CS184/284A

What's Wrong With This Picture?

CS184/284A

Jaggies (Staircase Pattern)

Is this the best we can do?

CS184/284A

Discussion: What Value Should a Pixel Have?

CS184/284A

- 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

- Many possibilities
- 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.

CS184/284A