Rasterization Basics

In this model, can you reason about:
- Shadows?
- Area light sources?
- Diffraction?
- Energy transfer?
How is this image drawn on the computer screen?

An image is an array of raster elements called pixels. Every pixel has at least a colour value.

Let’s take a closer look.

How is the sphere drawn using the pixels?

To draw a geometrical figure...
...we assign the correct pixels with the correct colour. This process is called rasterization.

Continue the pixel colouring to get regions filled with colour.

The framebuffer is a memory buffer storing the colour value for each pixel displayed.

So the image model mimics the memory model from hardware. An alternate image model is a vector image model.

How to colour the correct pixels?

```plaintext
function line(int x0, int x1, int y0, int y1)
    int deltax = x1 - x0
    int deltay = y1 - y0
    float error = 0
    float deltaylor = deltay / deltax
    // Assume deltax != 0 (line is not vertical),
    // note that this division needs to be done in a way
    // that preserves the fractional part
    int y = y0
    for x = x0 to x1
        plot(x,y)
        error = error + deltaylor
        if error ≥ 0.5
            y = y + 1
            error = error - 1.0
    Bresenham's Line Drawing Algorithm
```
How to colour the correct pixels?

- Extension for all line directions.
- Optimize.
- Demo
- Curves - read

O(0,0)  
X
Y

Bresenham's Line Drawing Algorithm

function line(int x0, int y0, int x1, int y1)
    int delta = x1 - x0
    int deltay = y1 - y0
    float error = 0
    float deltaerr = deltay / deltax
    // Assume deltax != 0 (line is not vertical),
    // note that this division needs to be done in a way
    // that preserves the fractional part
    int y = y0
    for x = x0 to x1
        plot(x,y)
        error = error + deltaerr
        if error ≥ 0.5
            y = y + 1
            error = error - 1.0

How to fill pixels?

Scanfill Algorithm

The Edge List

- Edges in the edge list become active when the y-coordinate of the current scan line matches their Y_{min} value.
- First intersection point between an active edge and a scan line is always the endpoint corresponding to Y_{max}.
Scanfill Algorithm

* For monotonically increasing/decreasing edges across a shared vertex count one intersection.
* Else count two.
* Ignore horizontal edges.