CS 488 Lecture 7 - Polygons Pg 1

- Multiply by the perspective matrix
  → converts from pyramid to the cube
  → clip before or after normalization but before is better

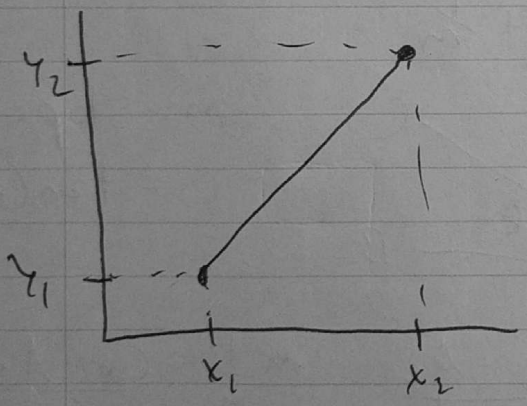
$$a = \frac{w_1 + x_1}{(w_1 + x_1) - (w_2 + x_2)} = \frac{BL_1}{BL_1 - BL_2}$$ ← for clipping

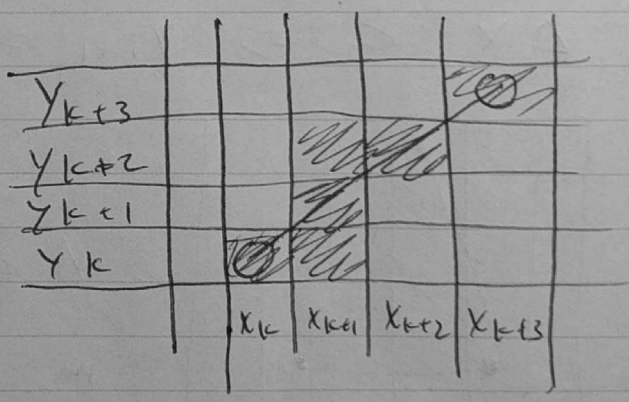
↳ same for the z, y planes

Polygons

Scan Conversion
Line Drawing algorithms

Line equation = $y = mx + b$





$\Delta y = m \Delta y$

- Digital Difference Analyzer (DDA)
  - assumption: line is processed left to right
  - line with positive slope

DDA

(A) $m \leq 1 \rightarrow \Delta x = 1$

$y_{k+1} = y_k + m$ Values rounded to nearest int

(B) $m > 1$ → we reverse roles of x & y

$\Delta y = 1 \quad x_{k+1} = x_k + 1/m$

If lines are processed right to left

(A) $y_{k+1} = y_k - m$ with $\Delta x = -1$

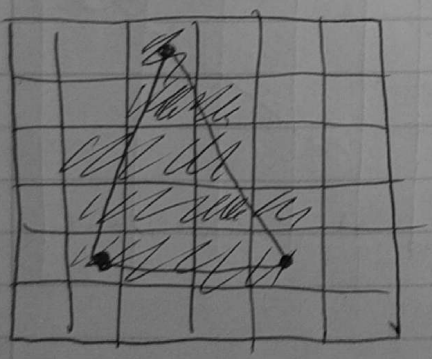
(B) $x_{k+1} = x_k - 1/m$ with $\Delta y = -1$

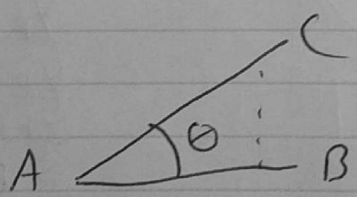
Scan Conversion

$y_i = m x_i + k \quad y_{i+1} = m x_{i+1} + k = y_i + m\Delta x$

if $y_{i+1} - y_i = 1$ (one scan line at a time)

$\Delta x = 1/m \rightarrow x_{i+1} = x_i + 1/m$





m = slope
$= \tan(\theta)$

$m = \frac{C_y - A_y}{C_x - A_x} \qquad m\frac{1}{m} = \frac{C_x - A_x}{C_y - A_y} = d_0$

$\frac{C_x - B_x}{C_y - B_y} = d_1$

Lecture 7 continued

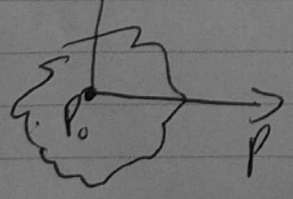
Hidden Surface / visibility Algorithms

- **Image Precision**
  - Determine which of n-objects are visible at each pixel in the image
  - brute force approach of cost $\simeq n \cdot p$ where p = # pixels

- **Object Precision**
  - compare objects directly with each other, eliminating entire objects or portions of them that are not visible
  - brute force approach is cost $\simeq n$~~p~~?
  - Superior for $n < p$ but individual steps are typically more complex and time consuming so it is often slower and more difficult to implement

Transforming Normals

Consider a plane implicitly defined by ~~$N^T(P-P_0)$~~ or $N^T \cdot P = 0$ if $P_0 = (0,0,0,1)$



where $P = [x, y, z, 1]$

Transforming Normals Continued

If we want to transform a point P using a matrix M, then to maintain $N^T P = 0$ we need to transform the normal $\vec{N}$ using a matrix Q such that $(Q \cdot N^T) \cdot MP = 0$

$$N^T Q^T M P = 0$$

$Q^T M$ must be the identity matrix

$$Q^T M = I$$
$$Q^T = M^{-1}$$
$$Q = (M^{-1})^T \qquad N' = (M^{-1})^T N$$

If M includes the perspective transformation then the inverse may not exist!
(M^{-1})

- usually do not transform normals
- use world coords to find normals later