CS6533/CS4533 Lecture 2 Slides/Notes

Scan Conversion for 2D Line Segments; HW1 Discussion; Introduction to OpenGL (Notes, Ch 1)

By Prof. Yi-Jen Chiang
CSE Dept., Tandon School of Engineering
New York University

1

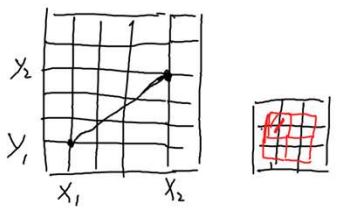
* Scan Conversion: Convert geometric entities to pixel assignment. (Rasterization)

Scan Conversion for Line Segments

DDA Algorithm (Digital Differential Analyzer)

A line: differential eq: $\frac{dy}{dx} = m \rightarrow$ slope.
Solve it numerically

Given: line segment endpoints (x_1, y_1) (x_2, y_2)
They are in frame buffer grid positions $\Rightarrow x_1, y_1, x_2, y_2$ are all integers. (pixel centers)



slope: $m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta y}{\Delta x}$ $\Delta y = y_2 - y_1$, $\Delta x = x_2 - x_1$

Assume: $m \in [0, 1]$

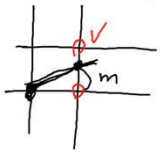
2

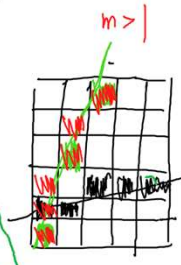
Alg: move from X_1 to X_2. [increase x by 1; y by $dy = m \cdot dx = m$] repeat.

Each x is an integer.

y is NOT ! ($\because$ m is a floating point number)

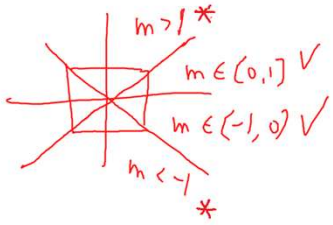
$\Rightarrow$ round y (choose the best y that's integer)





$m \in [0,1]$.

other slopes: by symmetry



Reason for $m \in [0,1]$:

For $m > 1$:
increase y by 1
choose the best x that's integer

3

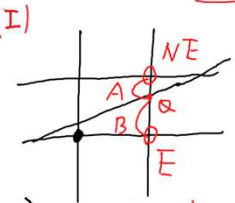
Bresenham's Alg.: Goal: Avoid all floating-pt computations

Assume: slope $m \in [0,1]$ → [increase x by 1; choose the best y] in each iteration.

Input: 2 endpts $(X_1, Y_1)(X_2, Y_2)$

Def: $\Delta y = Y_2 - Y_1 > 0$. $\Delta x = X_2 - X_1 \geq 0$. (Integers)

(I)



Defing: $D = \Delta x (A - B)$

$D < 0$ choose NE

$D \geq 0$: E.

(From picture) → $A < B$. choose NE

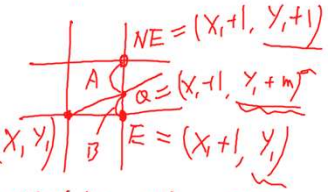
$A - B < 0$:

$\Delta x (A - B) < 0$:

: D.

(Note: $\Delta x > 0$)

(II) Compute D_{start} (first D value)



$A = 1 - m$

$B = m$.

$m = \frac{\Delta y}{\Delta x}$

$D_{start} = \Delta x (A - B) = \Delta x (1 - m - m) = \Delta x (1 - 2m)$

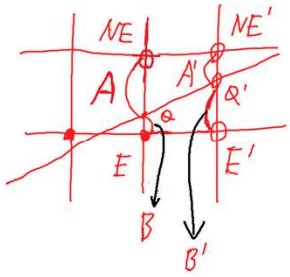
$= \Delta x \left(1 - 2 \cdot \frac{\Delta y}{\Delta x}\right) = \Delta x - 2\Delta y$.

($\Delta x, \Delta y$: integers $\Rightarrow D_{start}$: integer)

4

(III) Incremental Computation : Compute D_{new} from D_{old}.

(1) If E is chosen ($D_{old} \geq 0$)



$Y(Q') - Y(Q) = m$

$A' - A = -m$ —①

$B' - B = m$ —②

① - ② :

$(A' - A) - (B' - B) = -2m$

$(A' - B') = (A - B) - 2m$ ← $* \Delta X$ on both sides

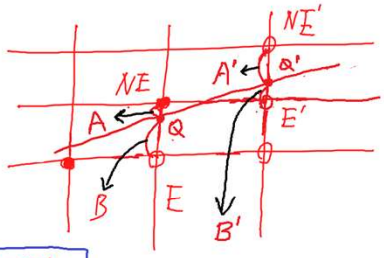
$\Delta X(A' - B') = \Delta X(A - B) - 2m \cdot \Delta X$

D_{new} D_{old} $-2 \cdot \frac{\Delta y}{\Delta x} \cdot \Delta x$

$D_{new} = D_{old} - 2\Delta y$ if $D_{old} \geq 0$

5

(2) If NE is chosen ($D_{old} < 0$)



$Y(Q') - Y(Q) = m$.

$A' - A = 1 - m$ —①

$B' - B = m - 1$ —②

① - ②

$(A' - A) - (B' - B) = 2(1 - m)$

$(A' - B') = (A - B) + 2(1 - m)$ ← $* \Delta X$ on both sides

$\Delta X(A' - B') = \Delta X(A - B) + 2 \cdot \Delta X\left(1 - \frac{\Delta y}{\Delta x}\right)$ $\left(m = \frac{\Delta y}{\Delta x}\right)$

D_{new} D_{old} $= 2\Delta x - 2\Delta y$

$D_{new} = D_{old} + 2\Delta X - 2\Delta y$ if $D_{old} < 0$.

Summary :

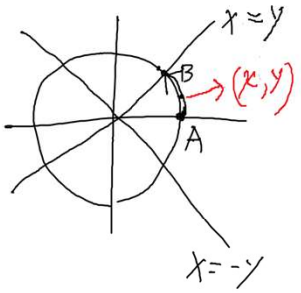
$D_{start} = \Delta X - 2\Delta y$

$D_{new} = D_{old} + \begin{cases} -2\Delta y & \text{if } D_{old} \geq 0 \\ 2(\Delta x - \Delta y) & \text{else.} \end{cases}$

6

HW 1 Part (a)

The points in the 8 symmetric regions.

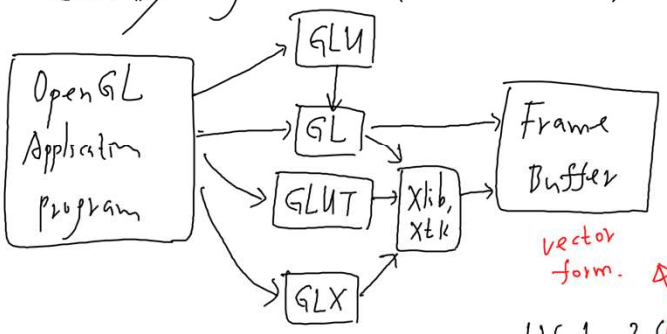


(x, y)	(y, x)
(-x, y)	(y, -x)
(x, -y)	(-y, x)
(-x, -y)	(-y, -x)

7

Introduction to OpenGL

* Library organization (for X window system)



GLU: graphics utility library

GL: OpenGL Library (indept. of window system)

GLUT: graphics utility toolkit (interface with windows system X window, MS windows)

GLEW

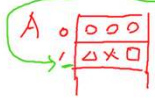
* OpenGL Command Syntax:

g l Vertex 3 f [v] ( )

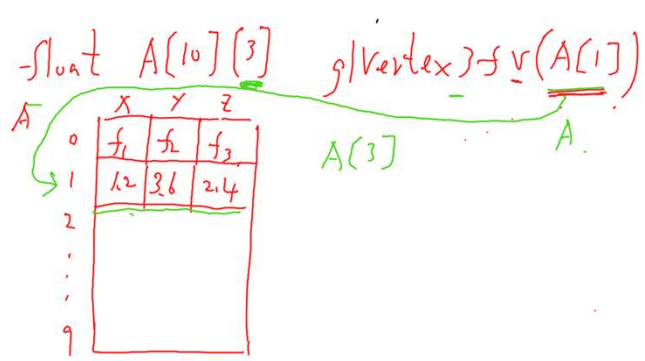
- # components: 2: (x, y) 4: (x, y, z, w) 3: (x, y, z)
- Data Type: i: integer, f: float ..
- v: vector form.

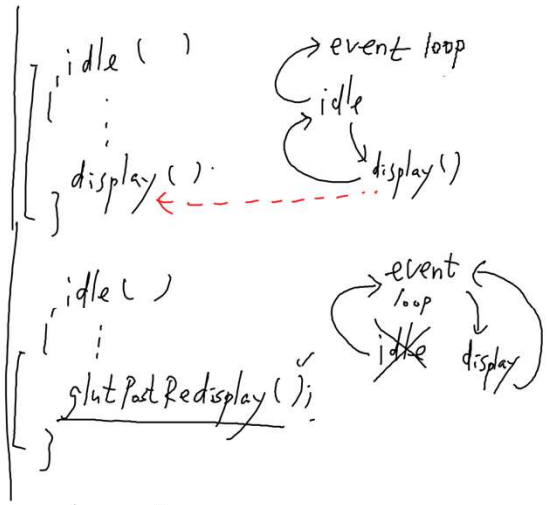
eg. glVertex3f(1.2, 3.6, 1.4)

glVertex3fv(ptr)



8





* Discussed the handout sample code "example.cpp".

9