Unit - II

24-12-19

Raster Scan Graphics & Transformation

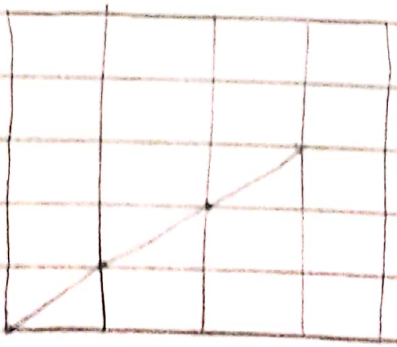
Pixel means discrete set of bright.

* Line drawing algorithm :

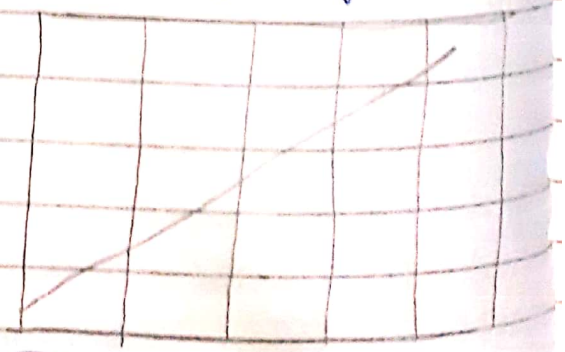
The cathode Rat tube is an raster display is consider a matrics of discrete finite areas (pixel) each of which can match it is not possible to directly draw a straight line from one point to another. The process of determining which pixel provide the base approximately to the desired line is known as rasterization.

When combine with this process of generating picture in scan like order it is known as scan conversion.

For horizontal & vertical 45° lines the choice of raster element is obirious for any other the choice is more difficult to shown in following fig:



Rasterization Scan Conversion



Rasterization of straight line.

It is useful to consider the general requirement for such i.e, what are the desirable characteristics for these line and that they start.

The primary design criteria are that straight line appear as a straight line & that they start at end accurately.

Display line should have constant brightness along their length, independent of the line length and ~~orient~~ orientation. finally the line should drawn rapidly.

As with most designed criteria not all can be completely satisfied.

The very nature of a raster display precludes the generation of completely stright line except for special case. It is possible for align to precisely being & end at satisfied locations.

Additional design consideration for line drawing algorithm include end points order i.e, rasterizing point P_2P_1 which ends the same result at rasterizing the point P_1P_2. End point shape i.e, Controlling the shape of end of the rasterizing of line.

Eg: Triangle, circle, polygon.

26-12-19

★ Digital Differential Analyzer. [DDA]

The line end points are (x_1, y_1) & (x_2, y_2) assume not equal.

Integer is the functions.

Note that many integer functions are float fun? i.e, integer $(-8.5 = -9)$ rather than (-8) the algorithm assume this is the case.

Sign returns $-1, 0, 1$ for arguments $<0, =0, >0$ resp. appro. the line length.

Algorithm :

```
if abs (x2 - x1) >= abs (y2 - y1) then
    length = abs (x2 - x1)
else
    length = abs (y2 - y1)
end if
  select the larger of Δx or Δy.
    Δx = (x2 - x1) / length
    Δy = (y2 - y1) / length
      x = x + 0.5
      y = y + 0.5
  begin main loop
    i = 1
  while (i <= length)
  set pixel (int x, int y)
    x = x + Δx
    y = y + Δy
    i = i + 1
  end while
  finish
```

One technique for obtaining a rasterize straight line is to solve the governing differential eqn for a straight line i.e,

$$\frac{dy}{dx} = \text{Constant}.$$

(OR) If $dy = y_2 - y_1$ $\quad dx = x_2 - x_1$.

The solution of the finite difference approximation

$$y_{i+1} = y_i + \Delta y$$

$$y_{i+1} = y_i + \frac{y_2 - y_1}{x_2 - x_1} \Delta x. \quad \text{---} \quad (2\text{-}1)$$

Where, $x_1 y_1$ & $x_2 y_2$ are the end points of the req. straight line & y_i is the inital value for any given step along the line.

In fact, in the eqn (2-1) represent the recurssion related for succesive values of y along the req. line.

It is use to rasterize line it is called digital differentiation analyzer.

for a simple DDA either Δx or Δy whichever is larger is chosen as 1 raster unit.

Eg: Consider a line from 0,0 to 5,5.

$x_1 = 0$

$y_1 = 0$

$x_2 = 5$

$y_2 = 5$

$\therefore$ length = 5.

$$\Delta x = \frac{5}{5} = 1.$$

$$\Delta y = \frac{5}{5} = 1.$$

i	Setpixel	x	y
1	(0,0)	0.5	0.5
2	(1,1)	1.5	1.5
3	(2,2)	2.5	2.5
4	(3,3)	3.5	3.5
5	(4,4)	4.5	4.5

The result are shown in above fig. Note that the selected pixels are equally spaced along the line. The appearance of the line is quite accepted.

The end point at (0,0) is apparently exact.

The pixel corresponding to the end point at (5,5) is not accurated.

The lines appears to be too short.

Eg: 0,0 to -8,-4.

$x_1 = 0$ $x_2 = -8$

$y_1 = 0$ $y_2 = -4.$

length = -8.

$$\Delta x = \frac{-8}{-8} = 1$$

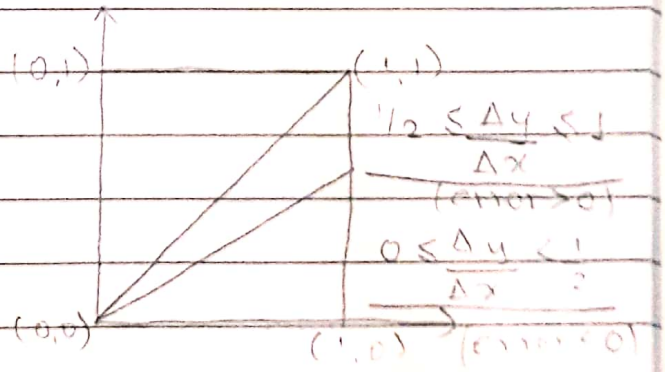
$$\Delta y = \frac{-4}{-8} = 0.5$$

i	set pixel	x	y
1	(0,0)	~~-0.5~~ ~~+0.5~~	~~+0.5~~ 0.
2	(-1,0)	-1.5	-0.5
3	(-2,-1)	-2.5	-1.0
4	(-3,-1)	-3.5	-1.5
5	(-4,-2)	-4.5	-2.0
6	(-5,-2)	-5.5	-2.5
7	(-6,-3)	-6.5	-3.0
8	(-6,-3)	-7.5	-3.5

30-12-19

Bresenham's line rasterization Algorithm for 1st octant.

The line end points are x_1y_1 & x_2y_2. assume not equal ~~or~~ $x, y, \Delta x, \Delta y$ are assume integer & e is real.



Initilize variable

$x = x_1$

$y = ~~y_2~~ y_1$

$\Delta x = x_2 - x_1$

$\Delta y = y_2 - y_1$

$m = \Delta y / \Delta x$

$e = m - 1/2$ (Initialize to compensate for a non zero intercept).

for i = 1 to Δx

set pixel (x, y)

while (e > 0)

$y = y + 1$

e = e - 1
end while
x = x + 1
e = e + m.
next i
finish

Eg: 1) $e = 1/4 - 1 \quad \frac{1}{4} \; \frac{-4}{4} \Rightarrow \frac{-3}{4}$.

2) $e = -3/4 + 3/8 = \frac{-6}{8} + \frac{3}{8} = \frac{-3}{8}$.

3) $e = -1/2 + 3/8 = \frac{-4}{8} + \frac{3}{8} = \frac{-1}{8}$.

4) $e = -1/8 + 3/8 = \frac{2}{8} = \frac{1}{4}$.

5) $e = \frac{1}{4} \; \frac{-3}{2} = \frac{1}{4} - \frac{6}{4} = \frac{-5}{4}$

6) $e = \frac{-1}{6} + \frac{2}{3} = \frac{-1}{6} + \frac{4}{6} = \frac{3}{6} = \frac{1}{2}$.

Eg: $x=0$; $y=0$, $\Delta x = 5$, $\Delta y = 5$., $m = 1$.

$\therefore e = m - \frac{1}{2} = 1 - \frac{1}{2} = \frac{1}{2}$.

Eg: (0,0) (-8,-4).

$x = 0$; $y = 0$

$\Delta x = -8$; $\Delta y = -4$.

$m = \frac{\Delta y}{\Delta x} = \frac{-4}{-8} = \frac{1}{2}$.

$e = \frac{1}{2} - \frac{1}{2} = 0$.

- It is originally developed by with digital plotters, this algorithm is equally suited for use with CRT raster device.
- The algorithm seek to select the optimum raster locations that represent a straight line.
- To accomplish this the algorithm always increment by 1 unit either x or y depending on the slope of the line.
- The increment in the other variable either 0 or 1 it is determined by examining the distance between actual lines & nearest grid locations.
- This distance is called error.

- The algorithm is cleverly constructed so that only the sign of error term need to be examine.

This is illustrated in above fig. for the line in first octant, i.e, for the line with slope between 0 & 1.

In the above fig. note that if the slope of the required line through (0,0) it is greater than ½ then it intercept with the line $x=1$ is closer to the line $y=1$ then the line $y=0$.

The raster pointer (1,1) better represent the path of the line that at (1,0).

If the slope is ($\leq$ ½) the opp. is true.

for a slope of ~~prisez~~ presizely half there is no clear choice.

Here the algorithm choses the raster point(1,0)

31-12-19

Not all lines pass presizely through a raster point. Where a line of a slope 3/8 initially passes through the raster point at (0,0) & subsequently crosses 3 pixel. Also illustrated is the calculation of the error in representing the line by discrete pixel. It is desirable to cheak only the error term, it is initialze to -½.

If the slope of the line is greater than or equal to half its value & at the next raster point by 1 unit i.e, 1,0.

$e = e + m$

↳ slope

$$e = -1/2 + 3/8 = \frac{-4}{8} + \frac{3}{8} = \frac{-1}{8}.$$

e is negative, therefore the line passes below the middle of the pixel.

Pixel at the same horizontal level better approx. the location of the line & y is not increment.

Again incrementing the error term by the slope is

$$e = -\frac{1}{8} + \frac{3}{8} = \frac{2}{8} = \frac{1}{4}.$$

At the next raster point is (2,0) here.

e is positive, which shows that the line passes above the mid point.

The raster element at the next higher vertical location (2,1) better approx. the position of the line, hence y is increment by 1 unit.

Before considering the next pixel it is necessary to reinitialise the error term.

This is accomplished by substrating 1 from it.

$$e = \frac{1}{4} - 1 = \frac{1}{4} - \frac{1}{1} = \frac{1}{4} - \frac{4}{4} = \frac{-3}{4}.$$

Note that the intercept of vertical line & $x=2$ & the desire line is $\frac{1}{4}$ with respect to the line $y=1$. Reinitialize to $-\frac{1}{2}$ related to 0 for the error term ~~ends as~~ yeilds as $-\frac{3}{4}$.

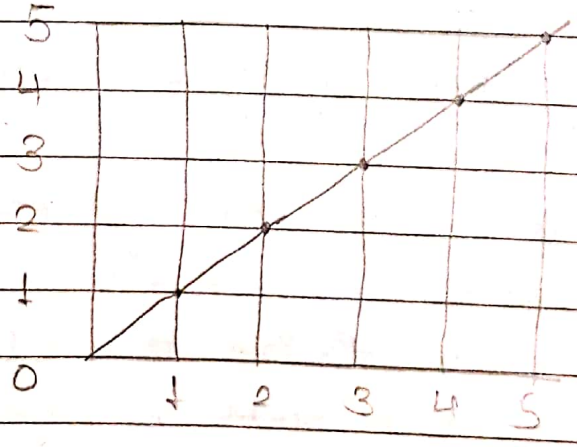
Continue the next raster unit

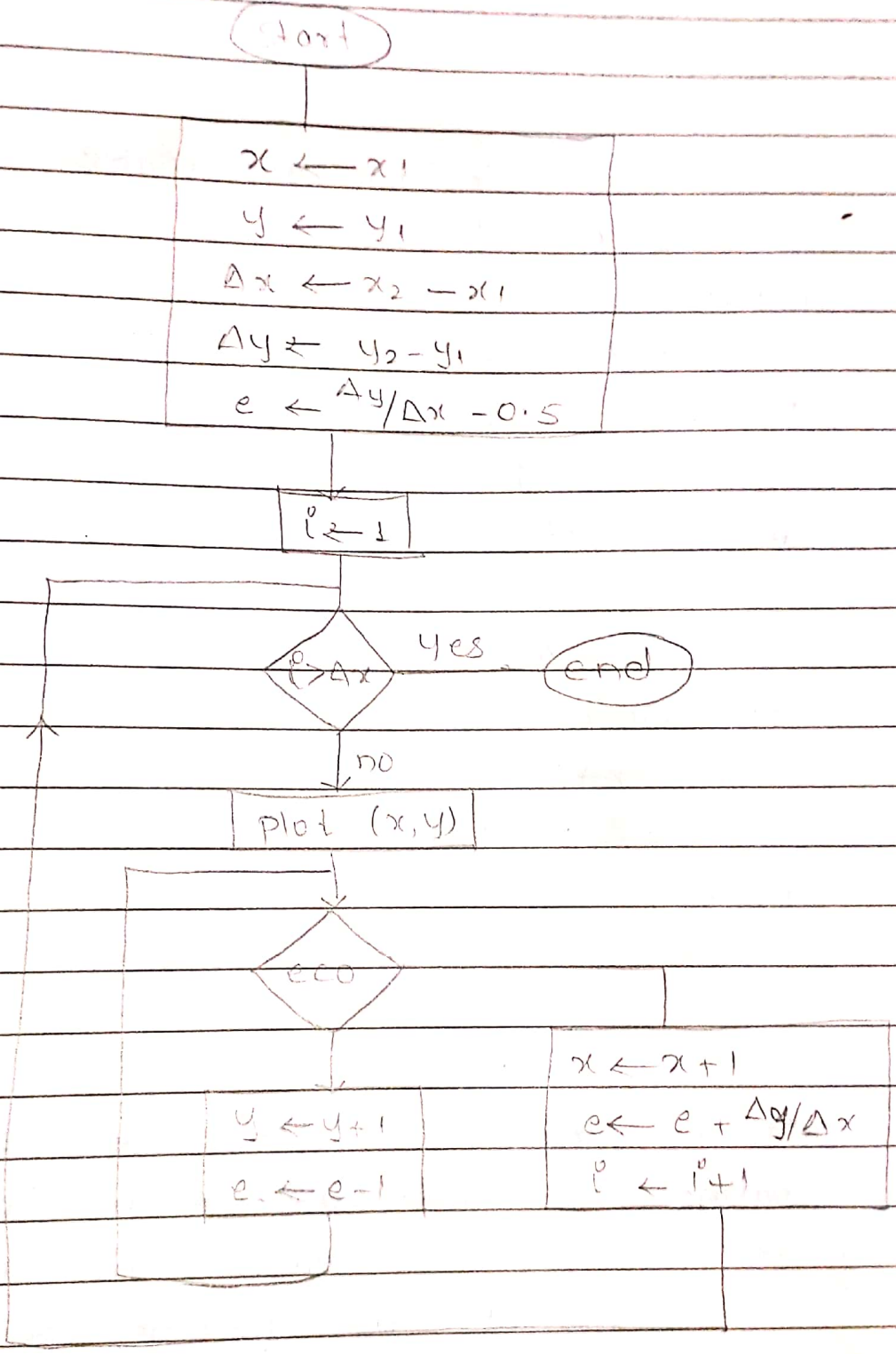
$$e = \frac{-3}{4} + \frac{3}{8} = \frac{-6}{8} + \frac{3}{8} = \frac{-3}{8}$$

e is negative, the y value is not incremented. It is easy to see that error term is major of the y intercept of the desired line & each raster element reference to $\frac{-1}{2}$.

Eg. $x=0 \quad y=0 \quad \Delta x=5 \quad \Delta y=5 \quad m=1. \quad e=1-\frac{1}{2}=\frac{1}{2}$

i	set pixel	e	x	y
1	(0,0)	1/2	0	0
		-1/2	0	1
2.	(1,1)	1/2	1	1
		-1/2	1	2
3	(2,2)	1/2	2	2.
		-1/2	2	3
4.	(3,3)	1/2	3	3
		-1/2	3	4
5.	(4,4)	1/2	4	4
		-1/2	4	5





01-01-2020

~~Integer~~ Bresenham Integer Algorithm for 1st octant

(x_1, y_1), (x_2, y_2) assume are not equal.

Initialize variable

```
x = x1
y = y1
Δx = x2 - x1        } Initilize variable
Δy = y2 - y1
ē = 2 * Δy - Δx   { Initilize ē to compensate
for i=1 to Δx     { begin loop
Setpixel (x, y)
while (ē > 0)
  y = y+1
  ē = ē - 2 * Δx
end while
  x = x+1
  ē = ē + 2 * Δy.
next i
finish
```

This algorithm is presented above requires use of floating point arithematic to calculate the slope of the line & to evaluate the error term.

The speed of the algorithm is increased by using integer arithematic. Because only the sign of error term is important, the simple transformation

$\bar{e} = 2e\Delta x$ of the error term, &

In the previous algorithm, yeilds & integer algorithm this allows algorithm to be efficently incremented

in hardware or firmware.
The integer algorithm for 1st octant i.e,
$0 \leq \Delta y \leq \Delta x$.

Increment x by -1			
Increment y by 1	$x = x-1$ $y = y+1$	$x = x+1$ $y = y+1$	Increment y by 1 Increment x by 1
Increment x by -1	$x = x-1$ $y = y-1$	$x = x+1$ $y = y-1$	Increment x by 1
Increment y by -1			Increment y by -1

Condition for general Bresenham Algorithm.

* Generalize Bresenham Integer Algorithm for All quadrants

All ~~various~~ variables assume integer.
The sign function (-1, 0, 1) as its argument
$<0, =0$ & >0.

Initialize variables
$x = x_1$
$y = y_1$
$\Delta x = abs(x_2 - x_1)$
$\Delta y = abs(y_2 - y_1)$
$s_1 = sign(x_2 - x_1)$
$s_2 = sign(y_2 - y_1)$

Interchange Δx & Δy depending on slope of line.

If $\Delta y > \Delta x$ then,

$10 = \Delta x$!

$\Delta x = \Delta y$

$\Delta y = 10$.

Interchange = 1 else interchange = 0

end if

$\bar{e} = 2 * \Delta y - \Delta x$ (initialize error term to compenscent non-zero intercept)

i = 1 to Δx

Set pixel (x, y)

while ($\bar{e} > 0$)

if interchange = 1 then $x = x + s_1$

else

$y = y + s_2$

end if

$\bar{e} = \bar{e} - 2 * \Delta x$

end while

If interchange = 1 then

$y = y + s_2$

else

$x = x + s_1$

end if

$\bar{e} = \bar{e} + 2 * \Delta y$

next i

finish.

eg. $(x_1, y_1) = (0, 0)$ $(x_2, y_2) = (-8, -4)$.

$x = 0$

$y = 0$

$\Delta x = abs(x_2 - x_1)$
$= abs(-8 - 0)$
$= -8$

$\Delta y = abs(y_2 - y_1)$
$= abs(-4 - 0)$
$= -4$.

$S_1 = sign(x_2 - x_1)$
$= sign(-8 - 0)$
$= -8$

$S_2 = sign(y_2 - y_1)$
$= sign(-4 - 0)$
$= -4$.

★ Assignment :

1. What is computer graphics & explain advantages of CG.
2. Explain applications of CG.
3. Explain CRT in details.
4. Explain colour CRT monitors.
5. Explain direct view storage tube in details.
6. Explain line drawing algorithm.
7. Write an algorithm for DDA
8. Write an algorithm for ~~line~~ Brekenham's line rasterization algorithm.
9. — " — for Bresenham's integer algorithm.
10. — " — Generalize Bresenham Algorithm.