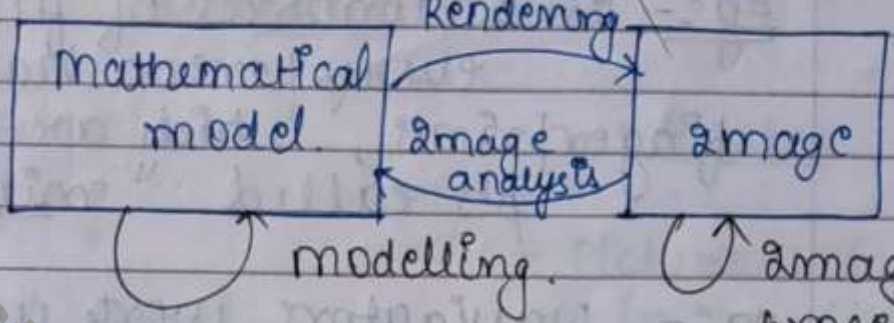
7 Jan, 19

CLASSTIME | Page No.
Date / /

# Computer Graphics.

→ First we need to create model for an image which can be represented either by eqn or by some pts.




One aspect of CG is modelling.

Creation of various models

To convert a model into image → Rendering

↓

covers aspects of covering color, texture, lighting effects, shadows etc.

not a part of model.

- Rendering is also called "Image Synthesis."
- Modification in image → Image processing.

↓

Aspect of Image processing is Image enhancement.

- If from an Image, we can get some useful info, then it is called Image analysis.

CLASSTIME | Page No.
Date / /

pattern recogonition.

Identifying certain common features from a given image.

eg:- In a biometrics of fingerprint every person has diff. fingerprints, which are made up of pts called "minutae".

eg:- Navigators work with help of satellite images which closely matches with images of confusted traffic & free places.

* Graphics → deals with models and converting it into images.
also deals with interaction.
eg:- Analysis used in stock markets.

Image analysis / Visualization.

- Computer Vision-
  ↓
  Includes rendering, image analysis, pattern recogonition.

  how are graphical devices (keyboard, mouse) are used to output images (plotter, printer).


Interaction

- Random/ Vector/ Calligraphy

- all images displayed on screen are stored in form of line drawing commands.

- The memory area is known as frame buffer.)

The controller takes image & displays it here, on screen.

- eg:- a landscape will be drawn in form of lines in frame buffer.

- similar to MS logo.

- realistic images/ curves are

- Raster

- Picture is stored in smallest unit i.e pixels.

- entire screen is considered as a rectangular grid of pixels

- Frame buffer stores value of each pixel.

- The controller picks up each pixel value & displays on screen

- Each value is converted into pixels.

Scan conversion

not very smooth.
Thus, generally used in architectural drawings.

- less commonly used as they are expensive

In terms of hardware & not efficiency.

- size of line drawing commands depends on complexity of image.
Also, takes a lot of time.

- colors & patterns to be drawn are much difficult as compared to raster graphics.

- for a resolution of 100 x 100 frame buffer size is 100x100. and thus it stores each pixel value.

Scan conversion

To draw a line, we need to find out what pts. lie on line & what are pixel pts in the background. (same technique as used to draw lines on graph)

Given 2 pts, first we find slope and they either by incrementing / decrementing we find pixels & then this would be send as foreground colour.

additional thing in raster graphics.
Done by the software

CLASSTIME | Page No.
Date / /

→ Primitives to be drawn ↓

① Line ② Circle ③ Ellipse.

NOTE:- Size of frame buffer is same as no. of pixels, no role of complex image
But time can vary.

We will deal with Raster Graphics & we have algos. for scan conversion.

• Drawbacks of Raster Graphics ↓

* aliasing / Staircase effect / Jagging :-

When we represent an image in form of pixels, If pixels are close to each other then we get a better image.
But if pixels are seen far apart, that means the resolution of screen is poor. & is called jagging.

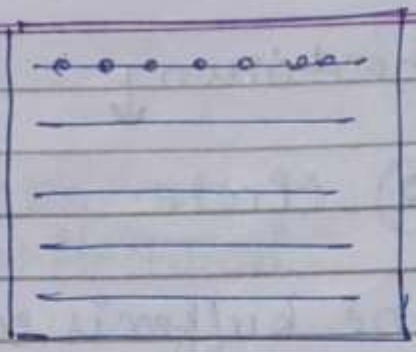
↓

→ To reduce this effect, we have anti-aliasing techniques.

Resolution - The screen is divided into no. of rows of pixels called scan lines.

∴ Resolution = no. of scan lines X no. of pixels on each scan line.

eg:-



200 scan lines & 100 pixels on each line

$= 200 \times 100$

- Aspect Ratio (related to display)

these pixels are w/o overlapping.

~~If no. of~~ How many pixels of vertical scan lines & horizontal scan lines are req. to be plotted to make lines of equal length.

If no. of vertical lines = no. of horizontal lines
↓
aspect ratio = 1

{Otherwise aspect ratio changes}

NOTE:- no. of pixels reqd. vertically to make a line of 1 inch is diff. from no. of pixels req horizontally to make a line of 1 inch.

→ aspect ratio → $\frac{3}{4}$ → Vertical (or 3:4)

{representation may vary}

3 pixels vertically and 4 pixels horizontally are req. to make line of equal length.

CLASSTIME | Page No.
Date / /

NOTE:- In an ideal situation, aspect ratio is 1.

- Persistence- (related to screen/display). The basic display device to be used ↓ CRT (cathode Ray Tubes)

Coated with phosphor material. To display a pixel on screen, $e^-$ are used which strike on phosphor. This raises energy of phosphor & light is emitted which appears as a pixel. which disappears after a given time.

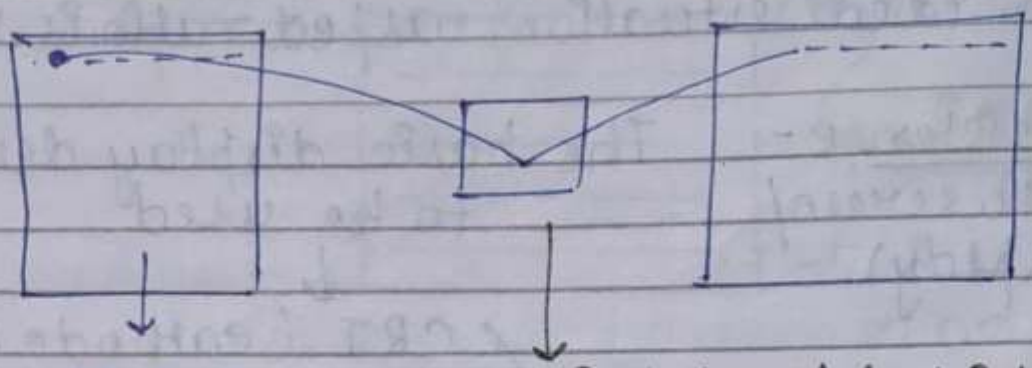
∴ The property of phosphor to persist to $\frac{1}{10}$ th of its intensity is persistence.

If persistence ↑

Refresh rate → how many times pixels take to refresh again per sec./ how many times frame buffer appears on sec.

If persistence ↑, refresh rate ↓es.
If persistence ↓, " " ↑es.

Interlacing- Image is in form of frame buffer which has set of values.



Digital.

First pixel is picked up and displayed

When beam moves from the last position of a line to the first position of 2nd line, nothing is printed on screen & this is called "Horizontal Retrace".

Vertical Retrace → no plotting of screen but only pen/ cursor moves from end to first {refresh}.

→ The time taken by vertical & horizontal retrace is very less.

→ Flicker - If monitor is not good, refresh rate is less, we can see a diff. of light & dark spots, called flicker.

↓

To reduce this :-

① Increase refresh rate

CLASSTIME | Page No.
Date / /

② Interlacing (interconnecting)

Plot the alternate scan lines in a given refresh rate, doubling the frame rate and flickering can be reduced & monitors using this technique are "interlaced monitors".

→ These lines are so close together that image does not flicker.

- Types of Display :-

① Monochrome - pixels have only single colour. eg - black white monitor.

② Grayscale - If system allows a different shade of grey.

If we want to make a pixel glow, it depends on shade of pixel.

Full power display — white
no " " — black
moderate " " — grey.

If a pixel is stored in only 1 bit, we can have 0 & 1 only.

black & white image.

In a row, m pixels } size of image
" " col, n pixels } m x n.

If for every pixel, we have 3 bits.

↓

∴ 8 values can be stored (0-7)

Thus, we can have 8 shades of grey

0 → 000 → black
1 → 001
2 → 010
⋮
7 → 111 → white

} grey shades

∴ Image size → m x n x 8

∴ 256 shades of grey can be stored.

As we use size of any pixel, size of image uses and thus more shades of grey can be encorporated.

(3) For a coloured image, shades will be divided into

↓

red blue green (~~rgb~~)

as they are primary colours

eg:- 24 bit system


→ for pure red pixel

111---1 000--- 000---

blue green

→ For white pixel

all 1's

1111---- 24 times.

all blue, red, green get value 1111---

NOTE:- For a colored image on a grayscale monitor, the monitor displays the image but in grayscale.

NOTE:- If System can store $2^{24}$ pixels but the monitor is not capable of, then we use concept of "lookup tables".

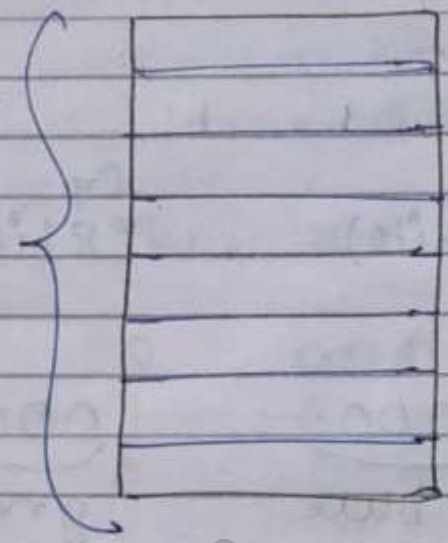
(at hardware level)

If monitor displays less colours but System can store more colour values then we need to match this mismatch.

display device displays — $2^{12}$ colour

acquired by pixels — $2^{24}$.

lookup table (like a palette).



$2^{12}$ colours at one time.

→ The colour of pixel is mapped in the lookup table & we pick up the 12 value & display it.

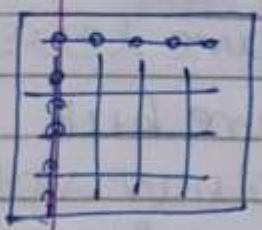
For some other colour, then we map another set of $2^{12}$ values into table.

* Numericals -

Q1 Find out the aspect Ratio of a Raster System using 8X10 inches screen and 100 pixels per inch.

8 → horizontal
10 → vertical.

$$\therefore \frac{8 \times 100}{10 \times 100} = \frac{8}{10} = \frac{4}{5}$$



{ horizontal scan lines / vertical scan lines }

CLASSTIME | Page No.
Date / /

Q2 How much time is spent scanning across each row of pixels during screen refresh on a raster system with resolution of 1280 X 1024 and refresh rate of 60 frames per sec.

{ 1280 ⟶ columns }
{ 1024 ⟶ rows }

Assuming horizontal & vertical retrace times are negligible.

1 frame ⟶ $\frac{1}{60}$ sec.

(1280 X 1024 scan lines)

For 1280 pixels ⟶ $\frac{1}{60 \times 1024}$ secs

{ as one scan row/line contain 1280 pixels }.

- Line Drawing Algorithms :-

⟶ A line is represented by 2 end pts. & a slope which predicts its type.

steep slope

gradual slope.

∴ line can have 4 cases.

CLASSTIME | Page No.
Date / /

$\therefore$ $0 < m < 1$
$-1 < m < 0$
$m > 1$
$m < -1$

2 algorithms for drawing a line.

① DDA ( Digital differential analysing)

or Incremental algo

- given after name of a mechanical device DDA that solves diff. eqn by numerical methods.

- This algo is also known as "Incremental line drawing algorithm". ↓

eg :- $y = mx + B$ ← eqn of line.

finding new value from old value

Inputs of DDA are $x_1, y_1$ and $x_2, y_2$ and then find slope.

$$y = \frac{dy}{dx}(x) + B$$

$$y \cdot dx = x \cdot dy + B$$

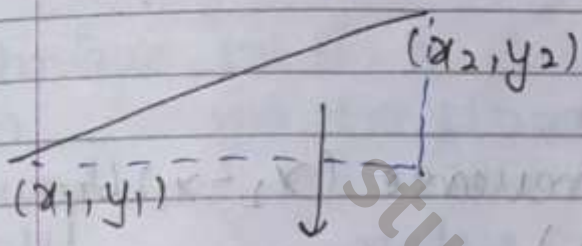
$x_{old}$ and $y_{old}$ ← old values ↓

$\therefore x_{new} = x_{old} + \underbrace{\Delta x}$ some incremental value of x

$y_{new} = y_{old} \pm \Delta y$

- **Gradual slope**

variation of x is more than that of y.




$x_2 - x_1 > y_2 - y_1$

$\therefore dx > dy$

$\therefore \frac{dy}{dx}(m) < 1$

$|m| < 1$

(Gradual slope)

→ Thus, Increment value of x & find value of y for this case.

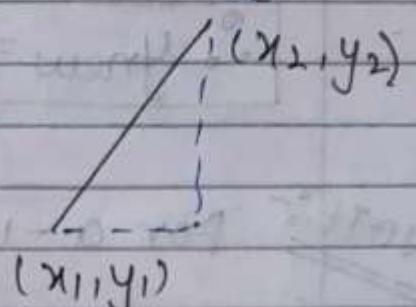
$\therefore \Delta x = 1$

(as pixels can be incremented/decremented in parts min value = 1)

$$x_{new} = x_{old} \pm 1$$

**Steep slope**

variation of y is more than x




$y_2 - y_1 > x_2 - x_1$

$\therefore \frac{dy}{dx}(m) > 1$

→ For this case, increment y & find x.

$\therefore \Delta y = 1$

$$y_{new} = y_{old} \pm 1$$

Now $\frac{dy}{dx} = m$

$\frac{dy}{m} = dx$

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

since $\Delta y = 1$

Now $\frac{dy}{dx} = m$

$\therefore dy = m \Delta x$

and $\Delta x = 1$

$dy = m$

$\therefore y_{new} = y_{old} + m$

$\therefore x = \frac{1}{m}$

$\therefore x_{new} = x_{old} + \frac{1}{m}$

NOTE:- For a -ve slope, reverse $(x_2 - x_1)$ by $(x_1 - x_2)$

① Given $x_1, y_1$ and $x_2, y_2$.

② If slope is -ve, reverse pts. checking (check abs. value) $x_1 < x_2$

Otherwise swap $x_1, x_2$ with $y_1, y_2$ and keep repeating these steps until every pixel is found.

NOTE:- { DT of slope ⟶ (m) }

↓

floating pt. value.

as $|m| < 1$

can be 0.05, ----

x, y ⟶ 2nteger values.

$y_{new} = y_{old} + (m)$ ⟶ float value

∴ We need to round off this value as converting it into float will truncate our value as y new is an integer.

Technique to round off.

eg - value = 1.2
+0.5
1.7

floor (1.7) = 1

value = 1.6
+0.5
2.1

floor (2.1) = 2

- Drawback → we use a floating pt. value and we are rounding it off i.e we are only able to solve using integral values.
Using floating pt. values increases time of solving.

Algorithm -

(1) Input line coordinates $x_1, y_1$ and $x_2, y_2$
(2) find slope of line as $(y_2 - y_1)/(x_2 - x_1)$
(3) If $|m| < 1$
increment x and find y as
$y = y + m$.

else increment x & find new y as $y = y + \frac{1}{m}$

To plot a pixel on screen, use

putpixel ( x1, y1, color);

If color not specified, it takes colour of setcolor

takes integral pts. always ∴ round its value to nearest integer

Since in C++ we don't have a roundfunc.

∴ y + 0.5 do its floor.

Round (yi) = FLOOR (0.5 + yi)

10 Jan. 19 Practical

Turbo C++ alt + F9 → run
alt + Enter → full screen.

help → index → line example.

Linking driver

initgraph ( &gdriver, &gmode, " ");
- closegraph() → to close graph
- reference variables
- specifies resolution.
- a path specifies for driver

CLASSTIME | Page No.
Date / /

→ initgraph (&gdriver, &gmode, " ") // initialises graphics to use screen in form of pixels.

int gdriver = DETECT, gmode, errorcode.

{ Ctrl + F1 → Help }.

starting pt. of line (0,0) → top left corner of the screen.

line (0,0), xmax, ymax) // draws a line.

rightmost corner

Options → Linker → Libraries → set Graphics library

[X] → set

- { F2 → Save file as . cpp }
- { fabs → absolute value }