

Computer Graphics

↓
System
↓
Data

↓
Geometrical
Shape
[Point → Line
• —
Curves]

[Pics → Picture
↓
Image]

- Computer graphics is an art of drawing pictures on computer screen with the help of programming.
- It deals with the creation, manipulation, and representation of visual images & animations using a computer.
- In Computer graphics objects are presented as a collection of discrete picture elements pixel.

{ Graphics devices = input devices + display devices }

- It is a rendering tool for the generation and manipulation of images.
- It also deals with the hardware as well as software support for generating images.

Computer graphics package

- Corel Draw
- Adobe Photoshop
- 3D Studio

Copyrighted by Codeleap.io
All rights reserved.

Why Computer graphics used

- Suppose a shoe manufacturing company want to show the sale of shoes for five years.
- For this vast amount of information is to store. So a lot of time and memory will be needed.
- This method will be tough to understand by a common man.
- In this situation graphics is better alternative.

[Graphics tools are charts & graphs]

- Using graphs, data can be represented in pictorial form. i.e. a picture can be easily understood just like a single look.
- Interactive computer graphics work using the concept of two-way communication b/w computer users.
- The computer will receive signals from the input device, and the picture is modified accordingly.

Computer Graphics

→ Generative Graphics

[used to create line, circle]

→ Image Analysis

[used to generation of pictures, image]

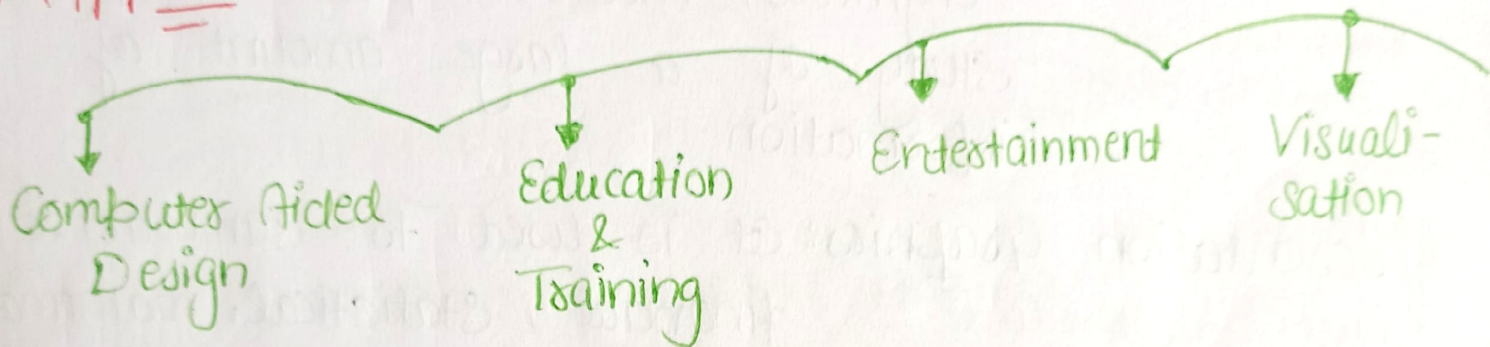
→ Cognitive

[used for photographic images]

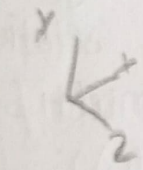
Advantages of Computer Graphics

- High quality graphics display on personal computer.
- Provide tools to produce the pictures.
- Produce Animations
- Using motion dynamic tools.

Application



- Computer Aided Design: It is a type of computer based tool used for drafting and designing.



[2D, 3D view]

- CAD is useful in various designing fields such as architecture, aircraft, spacecraft, automobiles
- Software packages for CAD application typically provide the designer with the multi-window environment.
- eg:- CAD, NX

- Education & Training: also used in classroom to enhance the educational experience and simplify teacher's work.

eg: flight

Simulator

• Ship simulator

- Computer-generated Model of the physical, financial is often used as educational aids.
- for some training applications, particular system are designed.

• Entertainment: CG methods are now commonly used in making motion, pictures, music video etc.

→ regularly used in many movies, TV series for generating some graphics scene.

• Visualisation: used for visualization of scientist, engineers, medical personnel, for the study of a large amount of information.
↓
techn: color coding, graphs, charts

• Presentation Graphics: It is used to summarize financial, statistical, mathematical, scientific, data for research reports and other type of report.
→ used to produce illustrations for reports or generate slides for use with projectors.
→ ex:- bar charts, line graphs, pie charts } showing relation b/w multiple parameters.

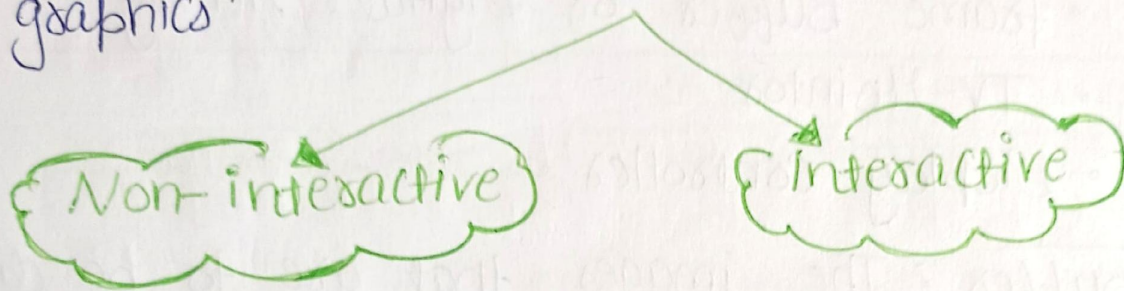
• Computer Art: It is widely used in both fine art and commercial art.
Ex: Cartoon drawing painting.
→ Artist use a variety of computer methods, including special-purpose hardware, artist paintbrush program, specially developed software.
→ electronic painting picture painted electronically on a graphic tablet using a stylus.

• Graphical User Interface: The use of pictures, image, icons, pop-up menus, graphical objects help in creating a user friendly environment where working is easy.
↓
Major Component
↓
window manager

• Medical applications: 2D colourful images of cross sections of human body or other organs are processed.

Types of Computer Graphics

There are two major types of Computer graphics.



• Non-interactive Computer graphics: } also known as Passive graphics.

→ It is one-way communication b/w the computer and user.

→ User does not have any control over the image i.e. user can see the produced image, and he cannot make any change in the image.

→ Image will work according to the instructions given in the stored program.

eg:- Screen savers, Titles show on TV

• Interactive Computer Graphics: } It is a two-way communication b/w computer and the user.

Copyrighted by Codeleap.io
All rights reserved.

→ User has some control over the image i.e. user can make any change in produced image.

→ User can make changes by giving different commands with input device.

- Higher quality
- Greater productivity

eg:- Ping-pong game

Drawing on touch screens

⇒ Components of interactive Graphics
(working)

- Frame buffer or Digital Memory
- TV Monitor
- Display Controller or Video Controller

• Frame buffer: The images that are to be displayed are stored in a frame buffer in the form of matrix of intensity value.

↓
[Vram]
↓
video

- helps to store the image in bit form.

• [Digital]

→ The frame buffer contains the image stored in binary form as a matrix of 0's and 1's which represent the pixel.

Memory in bits

$= 8 \times 1024 \times 1024$

$8 \times 1024 \times 1024$

[where picture def is stored]

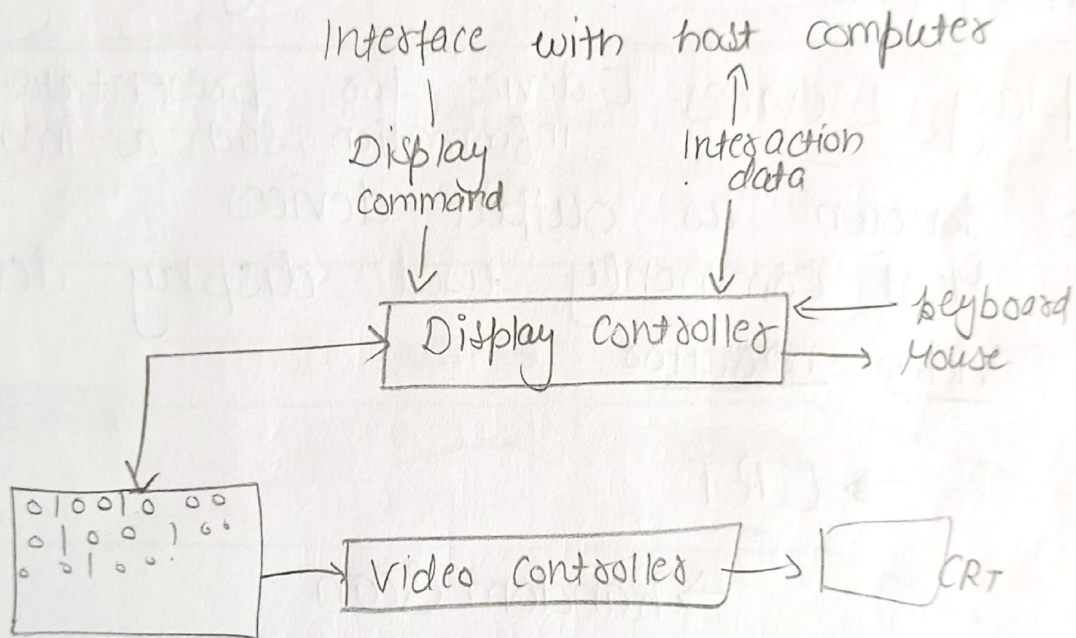
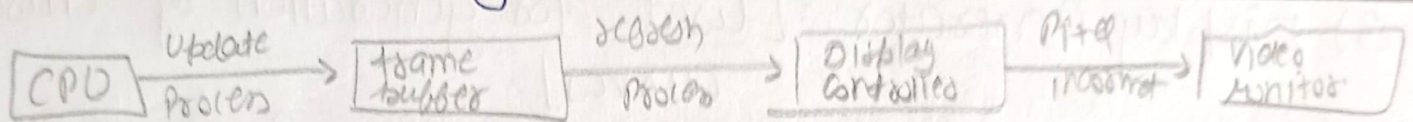
0 → darkness

1 → image

→ frame buffer retrieved and painted a intensity value on screen, one row at a time called scan line.

• TV Monitor: It helps to view the display, and they make use of CRT.

- Display Controller: It is a Memory Buffer & TV Monitor Interface. Its task is to pass frame buffer's contents to the monitor.
 - also capable of displaying image in colors
- The display controller read each byte of memory frame buffer data & convert 0's to 1's into the corresponding video signal.



Properties of Video Controller

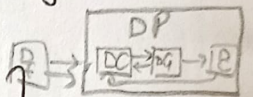
- Persistence
- Resolution
- Aspect ratio

→ as the time it takes the emitted light from screen to decay to one-tenth of its original intensity

- The phosphor crystals on screen continue to glow even after the e-beam is removed.
- Such a glow is known as phosphorescence
- The duration for which the phosphorescence exist is known as persistence.

Display Processor [used in convert the digital info. i.e digital to analog converter.]

- It is interpreter or piece of hardware that converts display processor code into pictures.
- Four main parts of display processor.
 - 1) Display File Memory: [Used for generation of pictures]
 - 2) Display Controller: [handles interrupt, maintain timing]
 - 3) Display Generator: [used for generation of characters, curves]
 - 4) Display Console: [contains CRT, light pen]



Display Devices [device for presentation of information, such as image, text].

- also known as output devices
- The most commonly used display device is a video monitor

→ CRT

→ Random Scan

→ Raster Scan

→ Colour CRT

→ DVST

→ Flat Panel Display

→ Plasma Panel Display

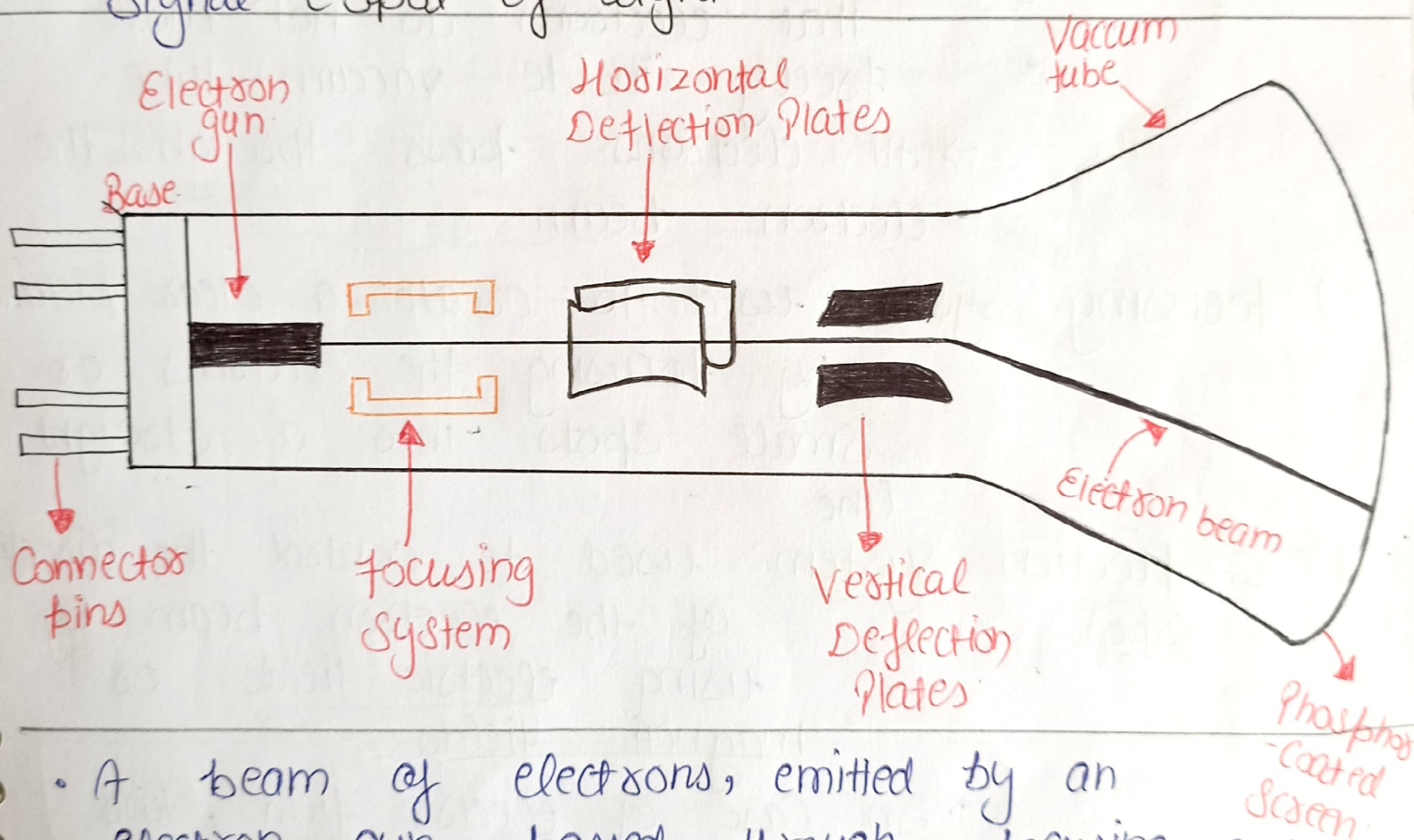
→ LCD

Main purpose of Display processor is scan conversion.
Scan conversion is a process of displaying different graphics object as a coll. of pixels.

CRT

- Resolution
- Persistence
- Aspect ratio

- Stands for "Cathode-Ray Tubes"
- CRT is a technology used in Traditional Computer monitor and televisions.
- It is a vacuum tube in which images are produced when an electron beam strikes a phosphor-coated surface.
- The function of Cathode ray tube is to convert electrical signals (electron beam) into visual signal (spot of light).



- A beam of electrons, emitted by an electron gun, passed through focusing and deflection system that direct the beam toward specified positions on the phosphor-coated screen.
- The phosphor emit a small spot of light at each position contacted by the electron beam. because the light emitted by phosphor fades very rapidly. some method need to maintain.

⇒ Elements of CRT

a) Electron Gun: Electron gun generates negatively charged electrons.

→ electron gun consist heating filament and cathode.

→ When heat is supplied to cathode by a heating filament its e^- become loose and get emitted from cathode surface.

b) Control Grid: It control the flow of e^- so that electrons do not move freely inside vacuum tube.

→ All electrons pass through the electron beam.

c) Focusing System: used to create a clear picture by focusing the electrons or small spots into a straight line.

d) Deflection System (Coke) - used to control the direction of the electron beam by using electric fields or magnetic fields.

→ In case of electric field, two pair of parallel plates are use.

- one for horizontal
- other for vertical deflection.

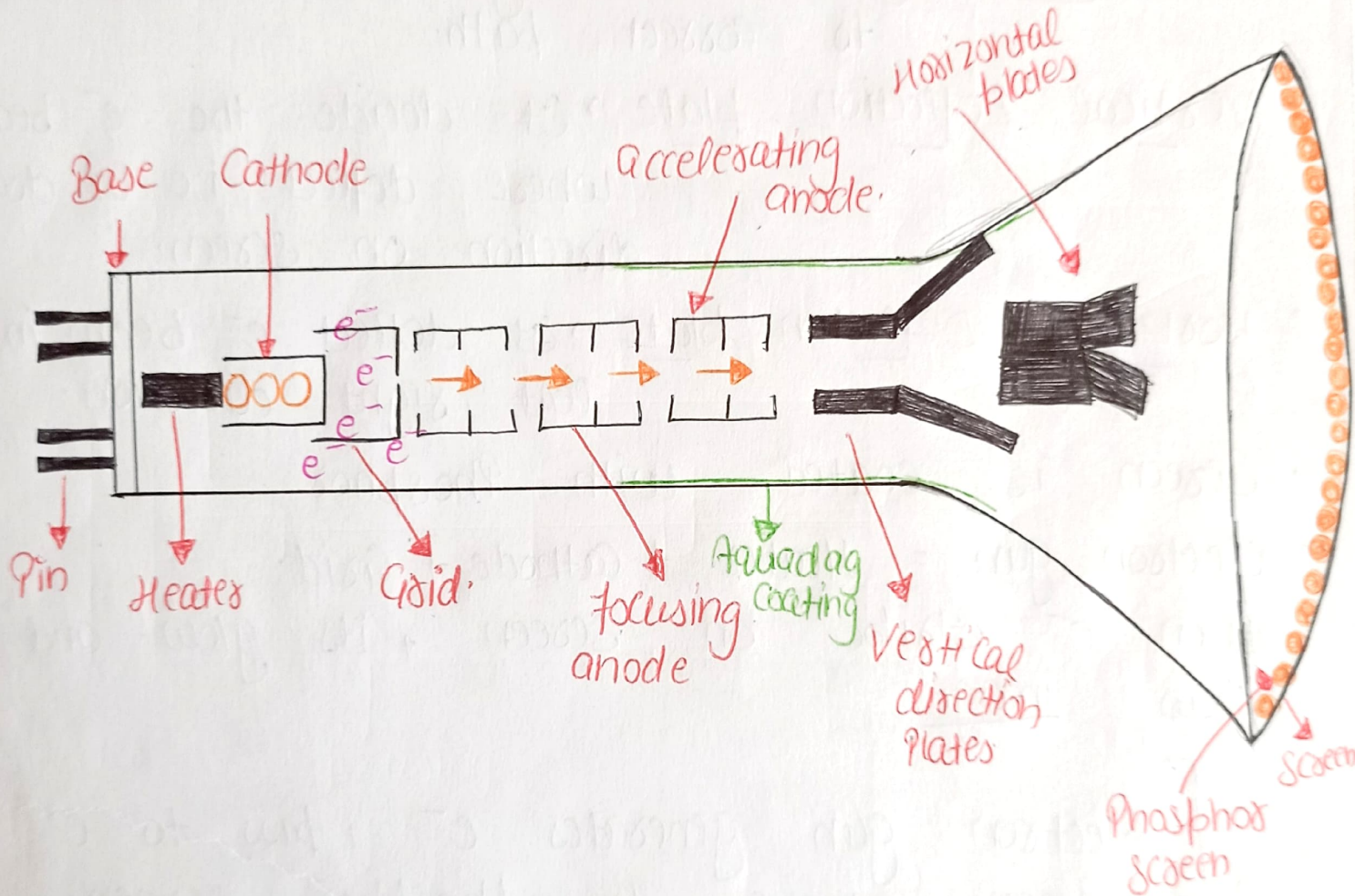
→ In case of magnetic field, two pair of coil are used.

- one for horizontal
- other for vertical deflection.

e) Phosphor-coated screen: The screen is coated with phosphorus crystals called phosphorus.

→ These phosphorus crystals emit small spots of light when the e^- beam strikes on them

→ and glow when a high-energy electron beam hits them.



⇒ Working of CRT

- Pin is a part where we apply voltage and it connects to base
- Heater → Give voltage to heater then heater produces heat and also red heat that is covered with a Cathode → It emits e^- when it heats up by heater and negatively charged e^- move in various

direction due to same charge they repel each other.

- Grid → to focus & pass e^- beam at high speed. Our main target is all e^- move in one direction i.e. apply some assembly to focus on e^-
→ In case any e^- move to other direction or outside for this protection use aquadag coating → they repel e^- and send to correct path.

- Vertical Deflection plate → It decide the e^- beam where deflect i.e. up, down direction on screen.

- Horizontal Deflection plate → It deflect e^- beam in left, right direction.

- Screen is coated with Phosphor.

- Electron gun = Heater + Cathode + Grid.

- When e^- strike on screen, it glow and show image.

[Electron gun generates e^- , pass to e^- beam move to phosphor screen, where they strike, glow the points]

Advantage

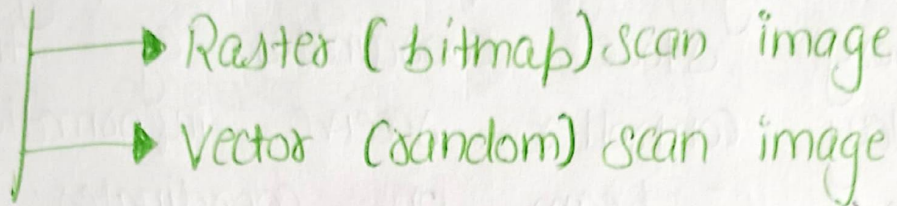
- reliable
- Cheap
- Capable of displaying rapidly changing pictures.

Disadvantage

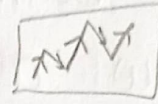
- High voltage required
- bulky & weighty.

Major Types of Graphical Image Display

- These are two major type

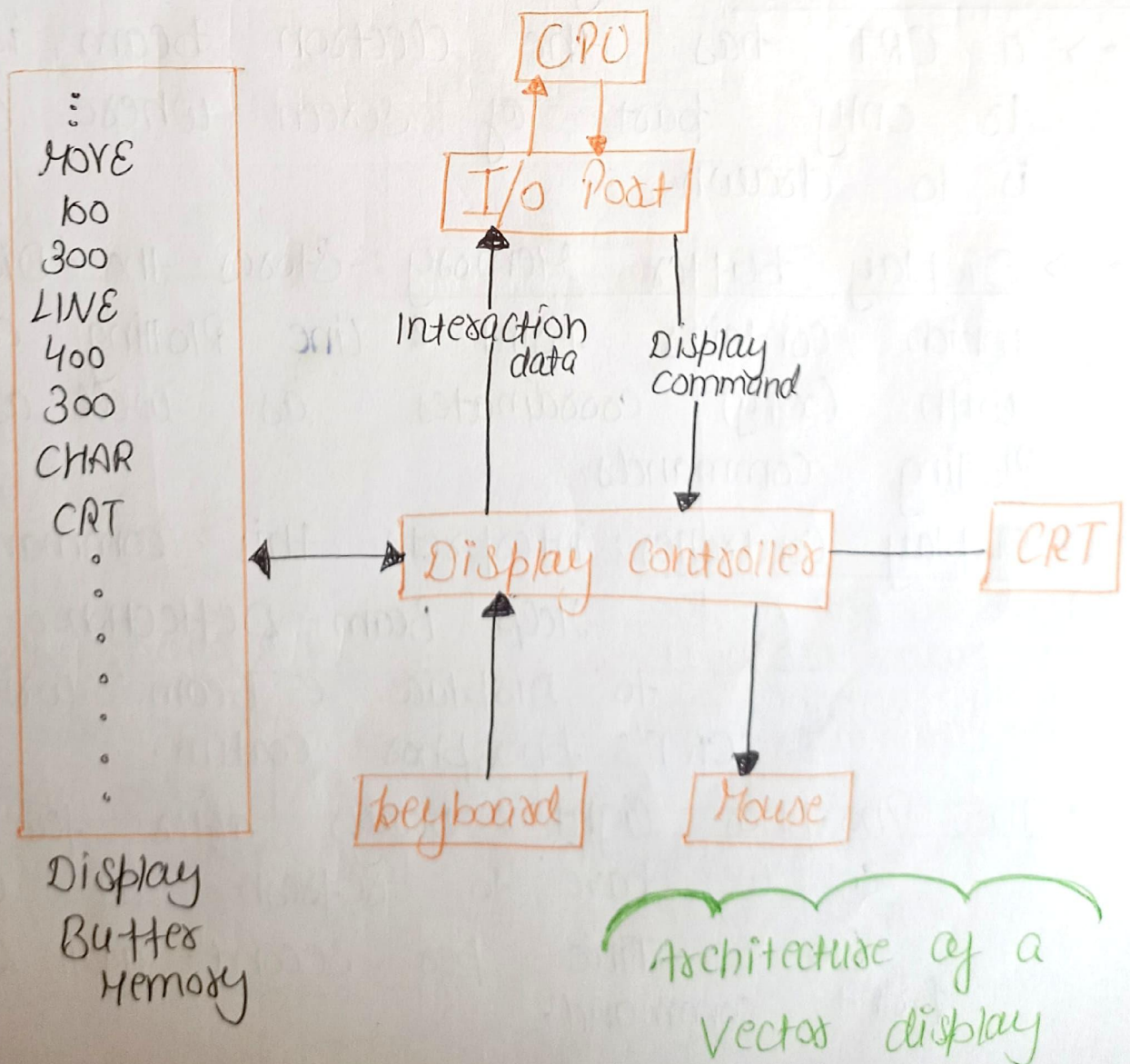


• Vector Scan display

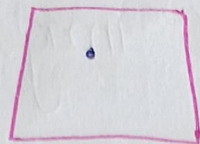


The image is constructed out of sequence of straight-line segments.

- Vector scan is a Technique used for producing image on screen.
- also known as Calligraphic, Stroke-writing display.



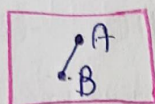
eg:



CRT Screen

Copyrighted by Codecap.io
All rights reserved.

- Show only one point, and display buffer memory store coordinates of that point
- Display Controller receive information and strike e^- beam on coordinates.
- Suppose draw a line with point A and B



CRT Screen

→ electron beam strike or on with A coordinate to B coordinate, phosphor coating is glow and display line.

→ a CRT has the electron beam is directed to only part of screen where a picture is to draw.

→ Display buffer Memory:- Stores the 'Display list' which contains point & line plotting commands with (x, y) coordinates as well as character plotting commands.

→ Display Controller: interprets this commands & read the content of the frame buffer & display it on monitor at the required refresh rate. to Displace e^- beam waiting on CRT's phosphor coating.

→ The Phosphor light decays after few Millisec, so we have to refresh at least 30-60 Time per second. again & again point command.

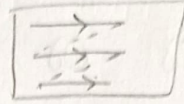
Advantage

- Higher resolution
[due to line command]
- produce smooth line
- Need less memory

Disadvantage

- Can't draw realistic image.
[only box point image]
- Limited Colour Capability
max four colour.

• Raster Scan Display

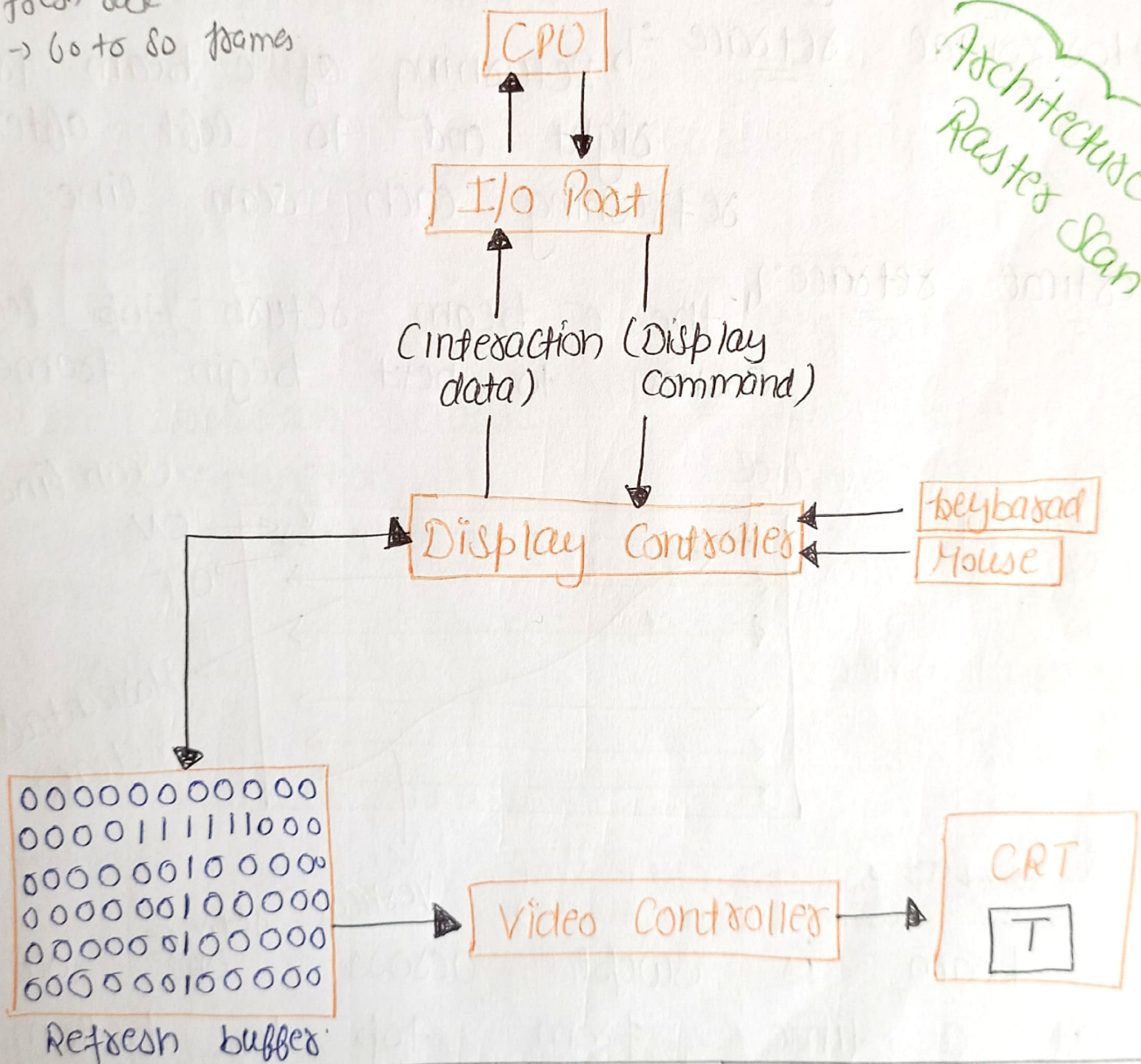


- also known as Bitmap Scan Display.

refresh rate

→ 60 to 80 frames

Architecture of
Raster Scan



- The Displayed img is stored in the form of 1's & 0's in the refresh buffer.
0's → Off
1's → ON [Create picture]

- The Video Controller reads the refresh buffer, & produces the actual images on the screen.

- It does this by scanning one scan line at a time from Top to bottom, and then back to top.

- Refresh buffer or frame buffer $\rightarrow$ stored picture

Properties

Size: resolution

Depth: color level

Speed: refresh speed

definition in memory i.e. set of

intensity value for all screen point.

$\Rightarrow$ Horizontal refresh: Beam refreshing is of 2 type.

Refreshing of e⁻ beam from right end to left after refreshing each scan line.

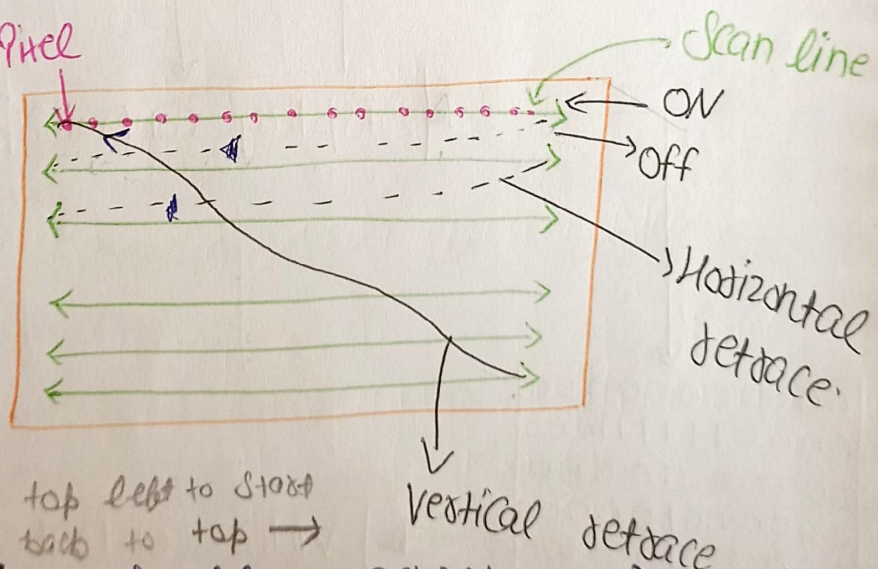
$\Rightarrow$ Vertical refresh: the e⁻ beam return top left corner to next begin frame.

- The process of moving the beam from right to left after completion of row known as horizontal refresh.

- When the beam reaches the bottom of the screen, it takes off and

rapidly retraced back to top left to start again & process of moving back to top $\rightarrow$

- e⁻ beam is swept across one row at a time from top to bottom as e⁻ beam move across each row the beam intensity is turned on/off to create a pattern.



Advantage

- Realistic image
- Million different color to be generated.
- Shadow scenes are possible.

Disadvantage

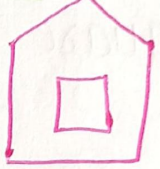
- low resolution
- Require screen size.

Difference b/w Raster & Random Scan

Raster scan

-  jagged line or zig-zag
- Composed of pixels
- low resolution because picture definition is stored as a intensity value.
- refresh rate 60 to 80.
- Show realistic image
- occupy more space
- eg:- • JPG
- Uses Shadow Mask technology

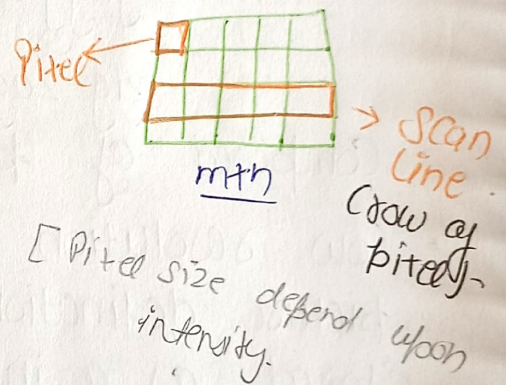
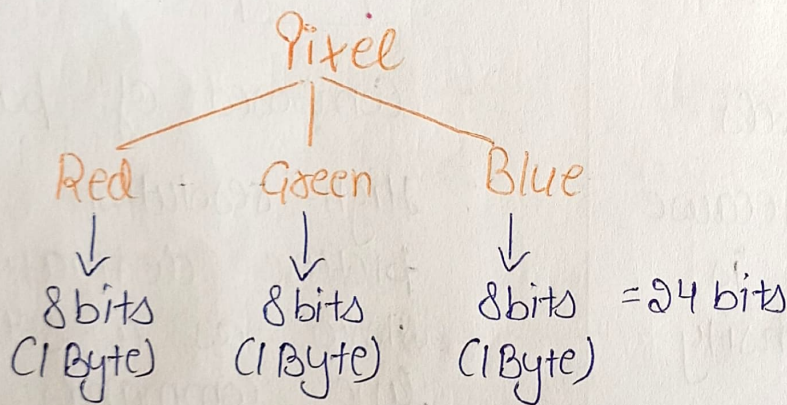
Random scan

-  smooth line
- Composed of path
- High resolution because picture definition is stored as a set of line commands.
- refresh rate per/sec. [30 to 60]
- not show.
- occupy less space.
- eg:- PDF
- Uses Beam Penetration technology.

Important terms related to display Devices

1) Pixel :- (or Picture element)

- The smallest addressable element of the screen.
- arranged in 2-Dim Grid (Row & Column)
- a screen is made up of a matrix of thousands or million of pixels.
- pixel is represented with a dot or square on a computer screen.



→ $2^8 = 256$ Shades of Red, Green, Blue

2) Resolution of image :- The no. of pixels per unit length (inch) in the horizontal as well as vertical direction.

max no. of points that can be displayed depend upon
 • Type of phosphor
 • focusing & deflection system

Resolution = Width $\times$ height

• Image resolution
 • Screen resolution
 [1280 x 1024 = high]

3) Dot pitch: It is the distance b/w any two dots of the same colour.

↓
 internal surface of the coated monitor screen is arranged into millions of tiny cells (red, green, blue)

- Measure of screen resolution.
- Smaller the dot pitch

↑ → resolution, sharpness.

4) Aspect ratio:- It is the ratio of the no. of x pixel to the y pixels.

→ width: height [Measured in unit length or no. of pixels]

SR = 800x600 AR = $\frac{800}{600} = \frac{4}{3} = 1.33$

5) Refresh rate:- The number of times per second the image is redrawn.
• 60 frames/sec = 60 Hz.
• SI unit is Hertz (Hz)

eg:- 60Hz refresh rate means the display refreshes the image 60 times per second.

⇒ Purpose

- 1) Smooth Motion
 - 2) Reduced Eye Strain
 - 3) Compatibility with Content.
- one way of keep phosphor glowing is to redraw the picture rapidly by quickly redirecting the e⁻ beam over same point. ← need higher refresh rate.
- To avoid flicker, we need to redraw or refresh the picture repeatedly i.e light emitted by phosphor fades very rapidly due to low refresh rate. So, we need to high refresh rate. Also avoid causing eye strain & headaches. [Some method to need to maintain the scan rate.]

6) Interlacing:- (refresh procedure) Technique used to half the bandwidth from memory to each frame is displayed in two passes. [Some method to need to maintain the scan rate.]

↓
Technique used to avoid flicker & improve image quality

1) Odd scan line

2) even scan line

• This technique used in some display system where "odd and even line are updated separately."

• The odd no. of line are refreshed in the first pass: 1-3-5-7 etc.

- In the second pass, even numbered lines are drawn: 2-4-6-8 etc.

Non-interlaced or progressive scan

- display line consecutively: 1-2-3-4-5-etc.
- scan line in a frame from top to bottom in one pass. [30 frames]

7) Bit Planes:

also known as bitmap.

- The block of memory which store bi-level intensity value for each pixel of a full screen pure black & white image.
- Bit planes refer to the individual binary planes that make up an image in computer graphics.
- In a digital image, each pixel is represented by a binary value, and these values are distributed across several planes based on their bit positions.

→ Image Compression

→ Visualization

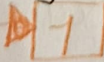
→ Color representation.

} Purposes

frame buffer

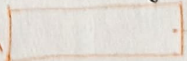


Register



DAC

Electron gun.

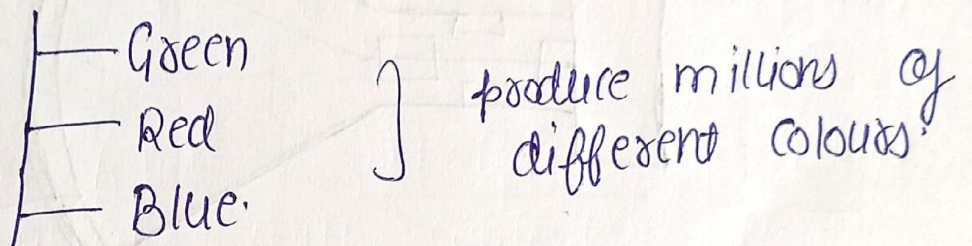


- Raster Graphics Device with a single bit plane having a Black & White frame buffer

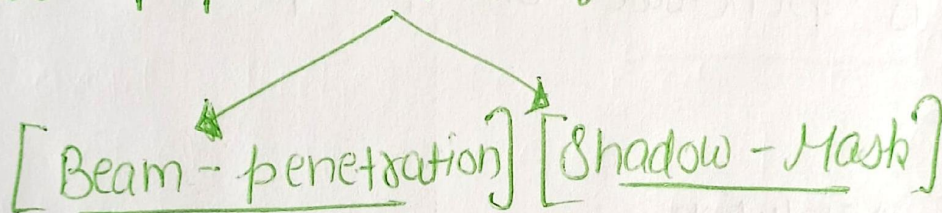
[Each pixel in the raster represented by at least one bit, this amount of memory of one bit per pixel called bit plane.

Color CRT → Normal CRT generate image of single colour due to limitation of phosphor coating.

- The CRT display color picture by using the combination of phosphors that emits different color light.
- By combining the emitted light from the different phosphors range of colors can be generate.
- The basic idea behind the color CRT Monitor is to combine three basic color



→ Two popular techniques



• Beam - Penetration :

used with random-scan monitors.

- CRT screen is coated with two phosphor colors.

→ Red

→ Green.

Controlled by beam-acceleration voltage

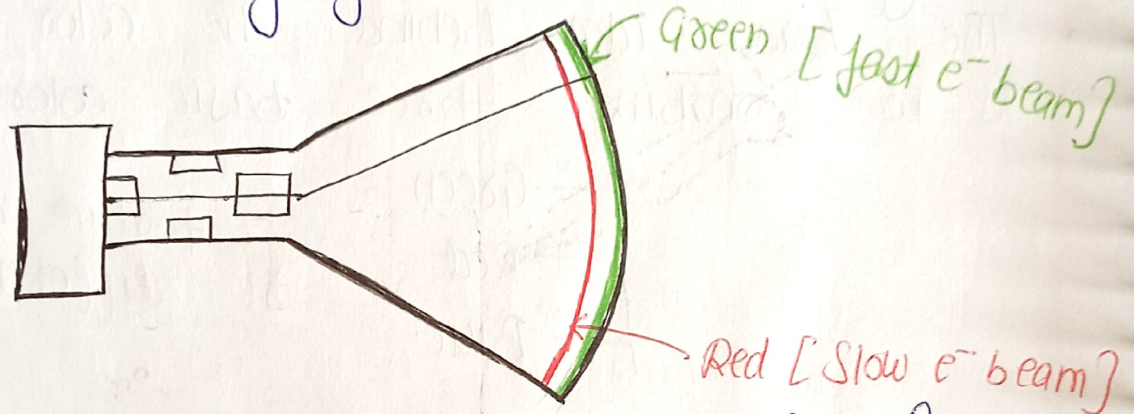
- Color depends on the speed of e^- beam (i.e. how far the e^- beam penetrates into phosphor layer)

Copyrighted by Codecap.io The inner layer is coated with red coloured phosphor.
All rights reserved.

- The outer layer is coated with

green coloured phosphor

- Slow moving e^- excites at red layer and emit red color
- High moving e^- hit at green layer and emit green color
- Intermediate speed will produce the combination of red & green and additional color orange & yellow



[Beam is penetrate on phosphor screen]

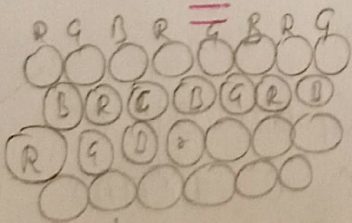
Advantage

- Better resolution
- Inexpensive

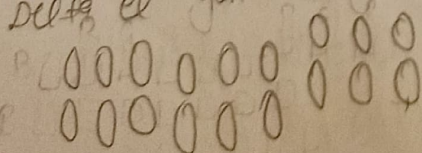
Disadvantages

- Only four colors are possible
- Quality of picture is not good

• Shadow Mask :



Delta e^- gun arrangement

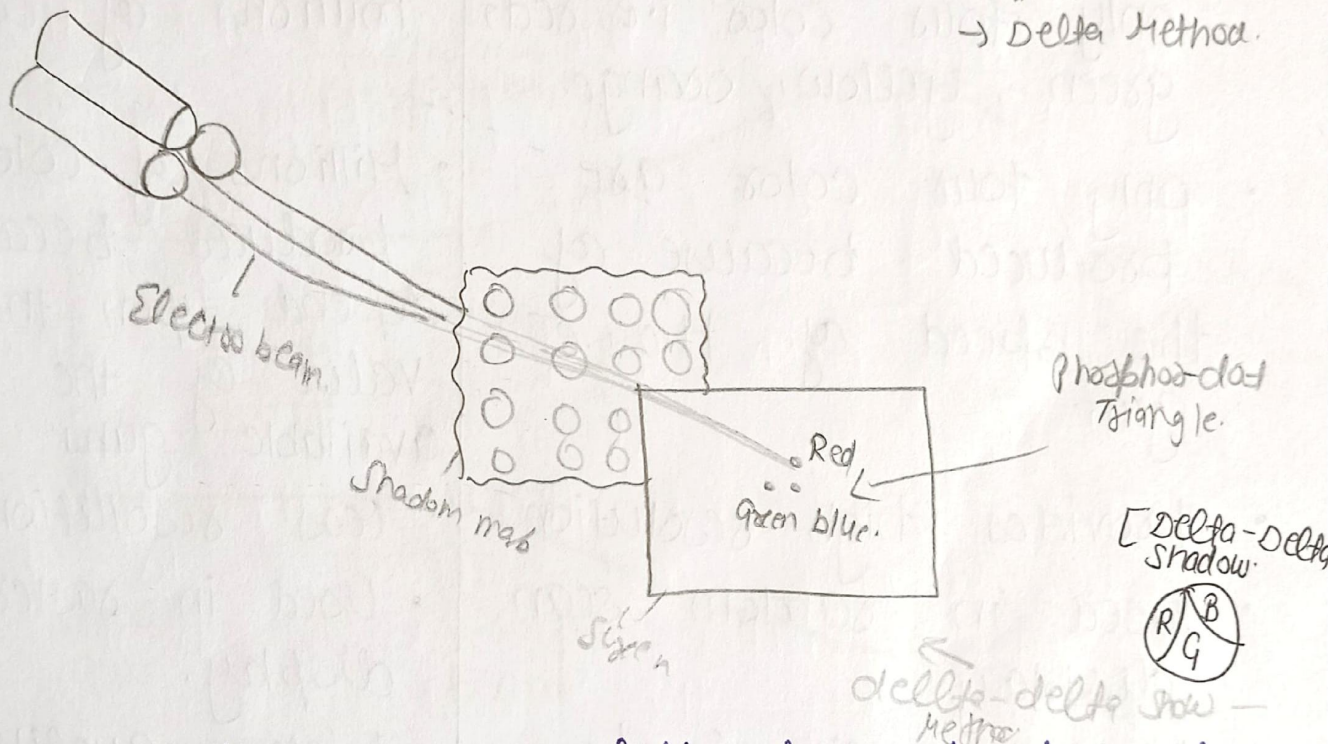


- Used in raster-scan, Display & produces wide range of color.
- Used in TV set & monitors.
- In this method, the CRT has three phosphor color dots.
 - red
 - green
 - blue

- This type of CRT has three e^- guns one for each color dot and a shadow mask grid just behind the phosphor coated screen.
- Shadow mask grid consists holes align with DOT patterns.

[Work according two tech.

- In-line Method
- Delta Method.



- Three e^- beams are deflected and focused as a group onto the shadow mask which contains a series of holes.
- When three beams pass through a hole in shadow mask they activate color dot triangle.
- one beam activate only one corresponding colour & produce a picture on screen.
- Colour generated by combination of 3 colors, by varying intensity More color is generated.

Advantage

- Display a wider range picture
- Display realistic images.

Copyrighted by Codecap.io
All rights reserved.

Disadvantage

→ Poor Resolution.

Difference b/w Beam Penetration & Shadow Mask.

Beam Penetration

- In this method, there is the production of only four colors i.e. red, green, yellow, orange.
- Only four colors are produced because of the speed of the e⁻ gun.
- provides high resolution
- used in random scan display
- picture quality is poor

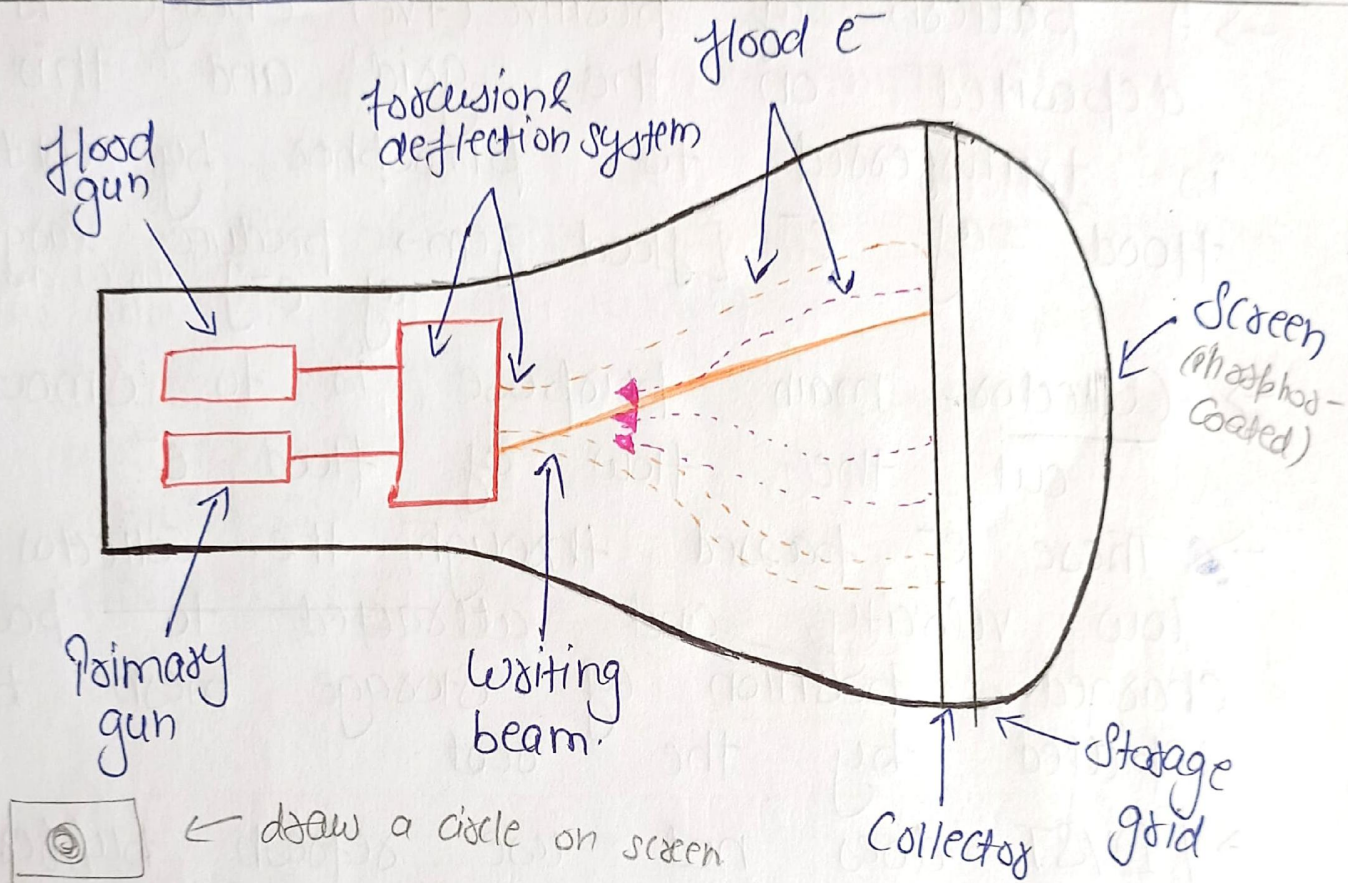
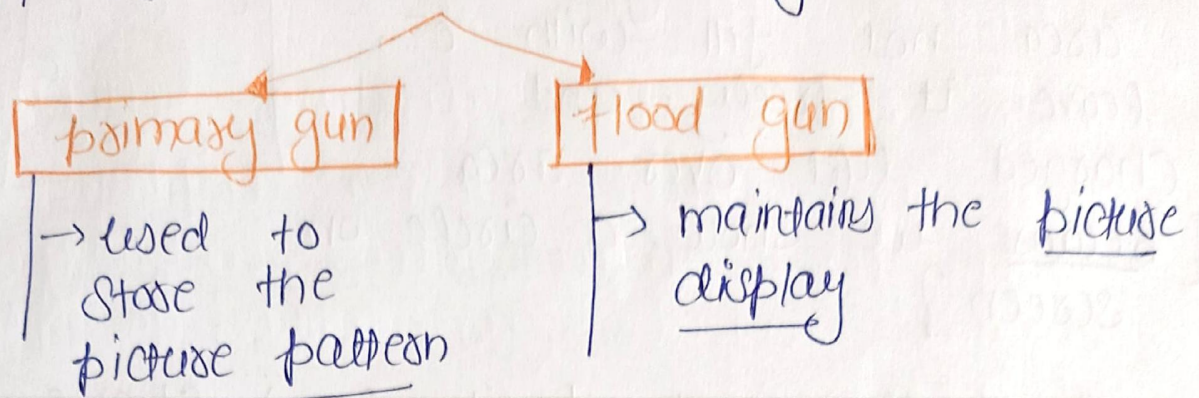
Shadow Mask

- In this method, there is the production of millions of colors.
- Millions of colors are produced because it depends upon the intensity value of the three available guns.
- less resolution.
- Used in raster scan display.
- picture quality is good

DVST

- Stand for direct-view Storage Tube
- In CRT, we need to refresh screen again and again but in case of DVST, no need to refresh the screen.
- DVST uses the storage grid which stores the picture information as a charge distribution just behind the phosphor coated screen.

- DVST consist two electron gun

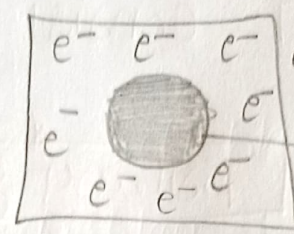


→ Storage grid:- Store image which show on screen.

→ Collector:-

- Primary gun which have information all about it, where draw the circle.
- Primary gun pass collector, go to storage grid i.e. collector is a transparent sheet, which we can easily cross. And draw pattern on grid.
- Primary gun strike e⁻ on storage grid fill every area with e⁻ except circle area.

• Where draw image, that area not fill with e^- , leave it freely and charged left over area [because we show a circle on screen]



know make the circle.
this area not charged by primary gun

→ A pattern of positive (+ve) charge is deposited on the grid and this pattern is transferred to phosphor by continuous flood of e^- [flood gun → produce large no. of e^-] this (-) charged grid reduce speed of these e^-

→ Collector main purpose is to smooth out the flow of flood e^- .

→ These e^- passed through the collector at low velocity and attracted to positively charged position of storage mesh but repelled by the rest.

→ [DVST does not use refresh buffer to store picture definition. Picture definition is stored in inside CRT in form of positive charged distribution]

Advantages

- Refreshing of CRT is not required.
- Very complex picture can be displayed at very high resolution without flicker.

Disadvantage

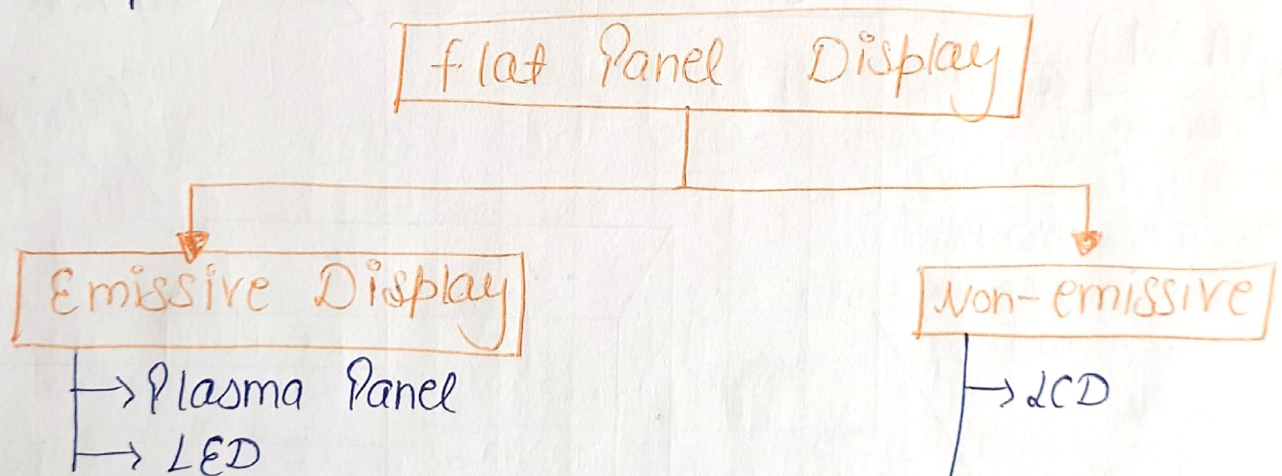
- Selected part of picture cannot be erased.
- not suitable for dynamic graphic applications.

Flat-Panel Display

[Flat screen]

[1964]

- It refers to a class of video devices that have a
 - [No. Consumers low power]
 - [Glass plates]
 - reduced volume, smaller, thinner
 - reduced weight
 - reduced power requirements compared to a standard CRT.
- A significant feature of flat-panel display is that they are **thinner than CRT** and we can hang on walls.
- eg: Small TV Monitors
pocket video games
laptop computers
portable monitors } current usage



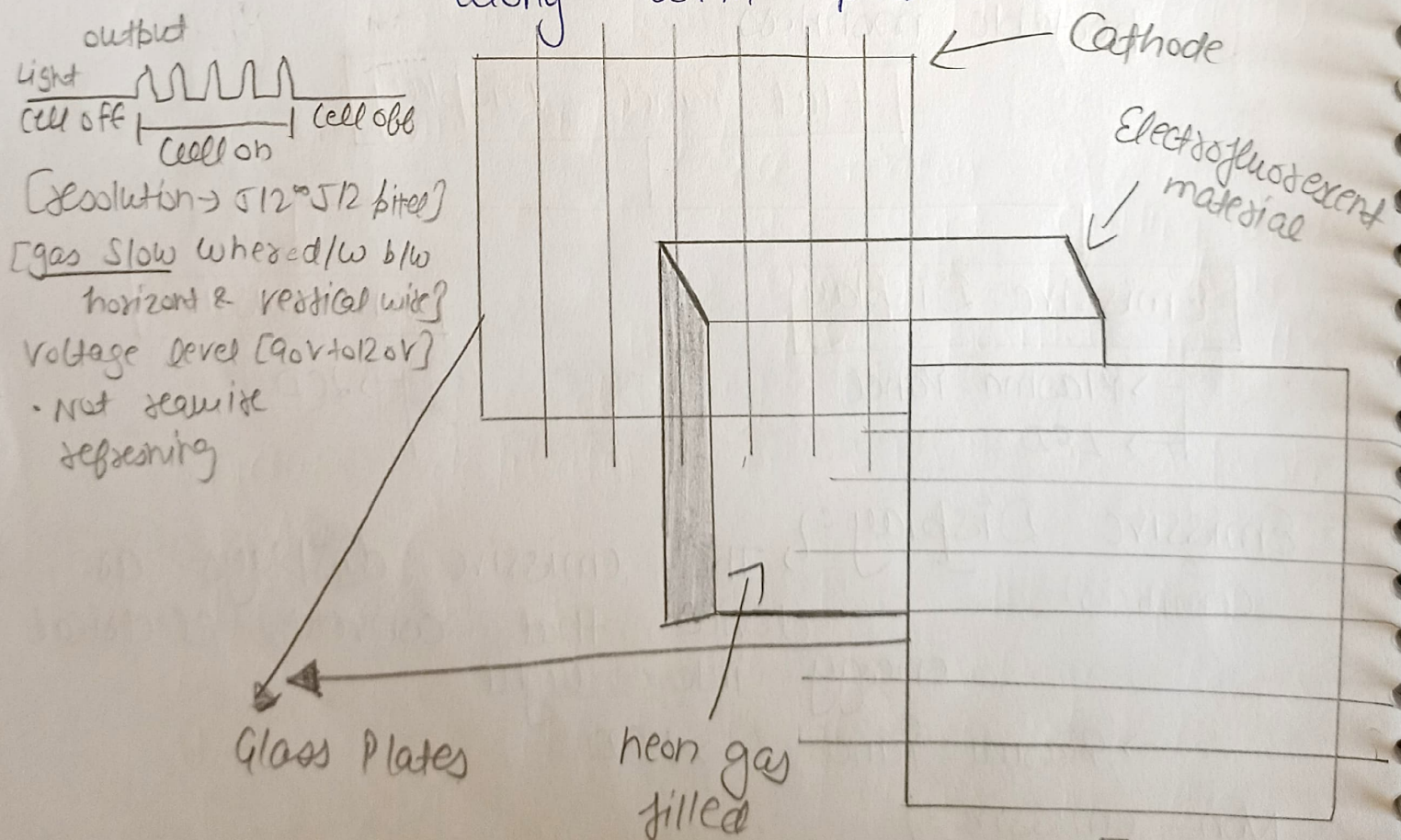
- Emissive Display : } The emissive displays are (emitters) devices that convert electrical energy into light.
 - Plasma Panel
 - LED
- Non-emissive : } use optical effect to convert (non-emitters) sunlight or light from other source into graphic patterns.
 - LCD

→ Plasma Panels

- also called gas-discharge display.
- They are called 'plasma' display because they use small cell containing electrically charged ionized gas, which are referred to as plasmas.

→ The essential components of plasma-display

- Cathode: It consists of fine wires.
 - delivers negative voltage to gas cells.
 - along with the negative axis
- Anode: It consists of line wires.
 - delivers positive voltage
 - along with positive axis



- Glass Plates: acts as capacitors
- Fluorescent cell: consist of small pockets of gas liquids when voltage apply to (neon gas) it emit light

Advantages

- High resolution
- large screen size is possible
- less volume
- less weight

Disadvantage

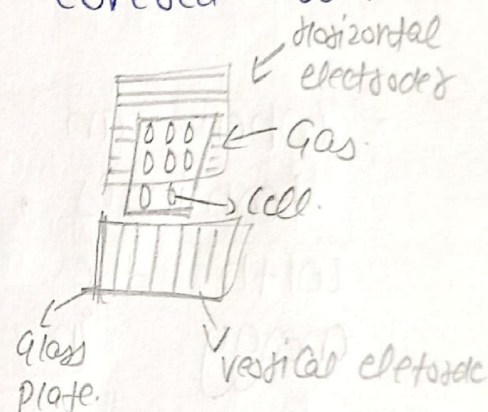
- Poor resolution
- Wiring req: anode & cathode is complex
- uses a lot of power

⇒ Working and Construction

- It consists of two sheets of glass with thin and close spaced gold electrodes attached to inner faces and covered with dielectric material.

- These are attached as series of vertical conductors on one glass plate

- horizontal on other glass plate



- The space b/w two glass plates is filled with neon based gas and is sealed.

- By applying voltage b/w electrodes the gas within panel is divided into tiny cells. and "each cell is independent of its neighbors."

- fixing voltage applied to pair of horizontal and vertical conductors cause the gas at intersection cell to break down into glowing plasma.

- This glow can be sustained by maintaining high frequency alternating voltage across cell.

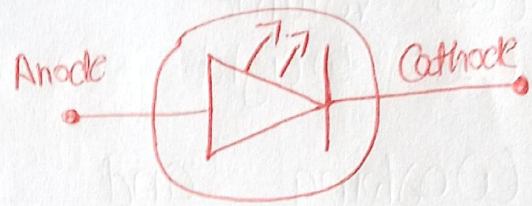
→ picture definition is stored in refresh buffer (refresh per sec)

→ LED

Copyrighted by Codecap.io
All rights reserved.

- Stand for Light - emitting diode

- two small [77] arrow denoting the emission of light



- LED is a diode or semi-conductor device with two terminals

LED Symbol.

→ anode
→ Cathode

- When an voltage is applied across the terminals, electrons and holes recombine with the semi-conductor material, releasing energy in the form of photons (light)

In this display: a matrix of multi-color light emitting diode is arranged to form the pixel position in the display.

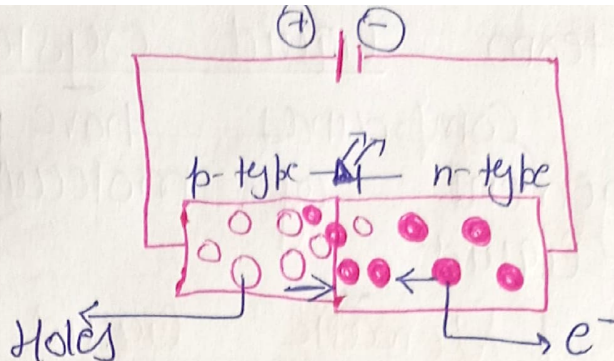
- picture definition is stored in refresh buffer.

- Information is read from the refresh buffer and converted to voltage levels that are applied to the diodes to produce the light patterns in the display

[Intensity of light can be controlled
→ Adv.

Working

- A light emitting diode is a two-lead semiconductor light source



- It is a p-n junction diode that emits light when activated.
- When a suitable voltage is applied to the leads, e^- are able to recombine with e^- holes within the device, releasing energy in the form of photons.
- This effect is called electroluminescence, and the color of the light is determined by the energy band gap of the semiconductor.

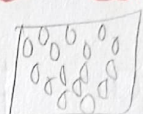
→ LCD

→ they compounds have a crystalline arrangement of molecules. [Smaller size, lower power consumption, lighter weight]

- Stands for Liquid Crystal Display
- It is a thin, flat panel display device mainly use in small system such as calculator and portable, laptop computer.
- Polarization of light is used here to display object
- These produce a picture by passing polarized light from the surroundings or from an internal light source through a liquid-crystal material that can be aligned to either block or transmit the light.

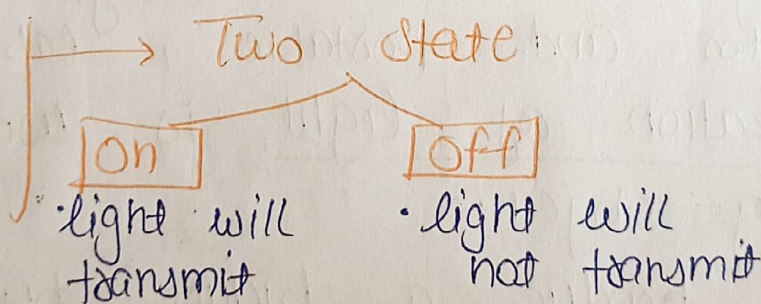
The pixels are switched on & off electronically using "liquid crystal" to rotate polarized light

- The term liquid crystal refer to fact that these compounds have a crystalline arrangement of molecules, they flow like a liquid.

flat panel display commonly use π Nematic liquid crystal compounds use  $\rightarrow$ Nematic phase.

Construction & Working

- Two glass plates, each containing a light polarizer at right-angles to other plate; sandwich the liquid-crystal material
- Row of horizontal transparent conductor are built into one glass plate, and columns of vertical conductors are put into the other plate.
- The intersection of two conductors defines a pixel position.



- Molecules are aligned in "on State".
- Polarized light passing through the material is twisted so that it will pass thro. the opposite polarizer.
- The light is then reflected back to the viewer.
- To turn off the pixel, we apply a voltage to the two intersecting conductors.

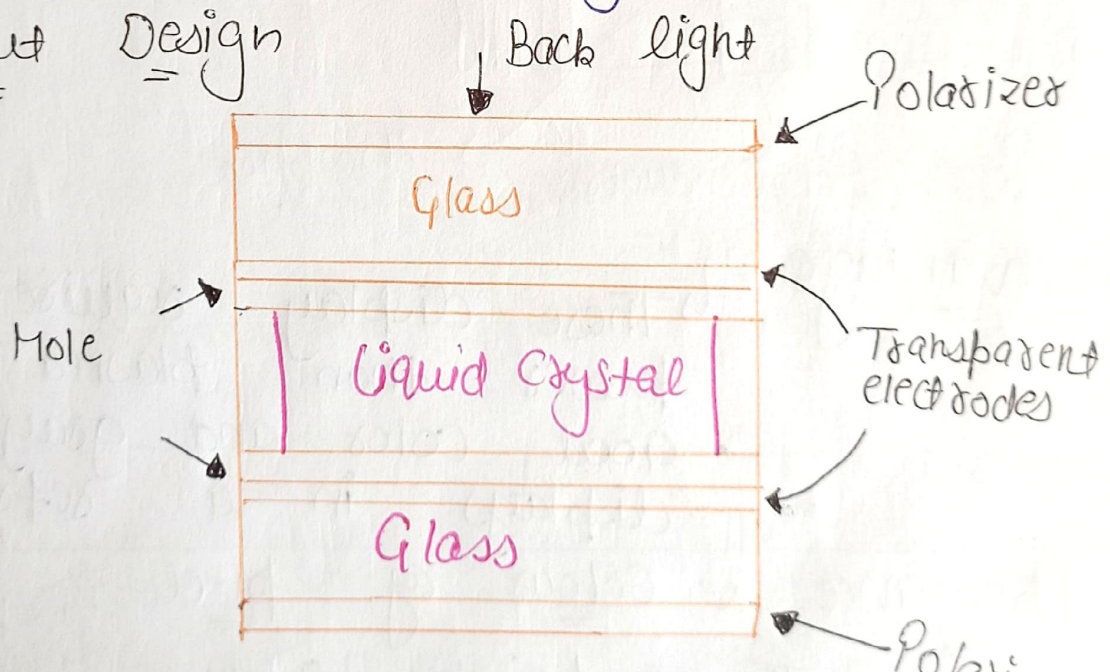
to align the molecules so that the light is not twisted.

→ it is referred as "passive Matrix" LCD.

→ Picture definition is stored in a refresh buffer and screen is refreshed at the rate of 60 frames per second.

→ "Active Matrix" display: This is the another method for constructing LCD is to place a transistor at each pixel position, using thin-film transistor technology. → transistors are used to control the voltage at pixel location and to prevent charge from gradually leaking out of the liquid-crystal cells.

→ Layout = Design



- Each pixel of an LCD consists of a layer of molecules aligned b/w two transparent electrodes and two polarizing filters, the axes of transmission of which are perpendicular to each other.

Advantages

- 1) Low power consumption
- 2) Small size
- 3) Low Cost

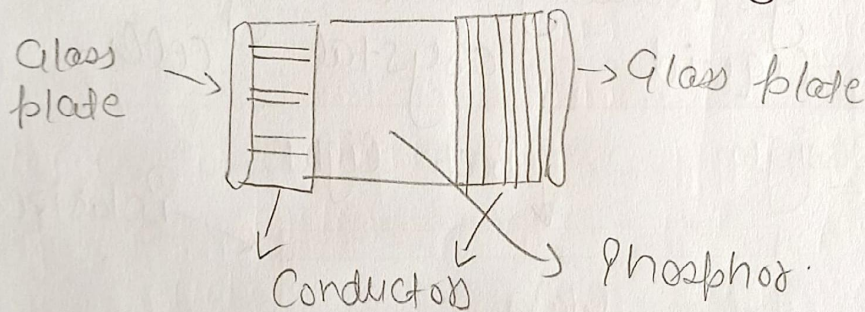
Copyrighted by Codecap.io
All rights reserved.

Disadvantage

- 1) Temperature dependent.

→ Thin-film Electroluminescent

- They are similar in construction to a plasma panel.
- The difference is that the region b/w the glass plates is filled with a phosphor instead of a gas.



Disadvantage:

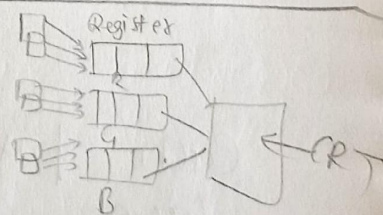
- These displays require more power than plasma panels.
- Good color and gray scale displays, in a refresh buffer.

Appearance & Colour of pixel

→ Three primary colours.

• Intensity of all e-beam is high result in a White pixel

• Intensity of 3 e-beam is low result in Black pixel



Color Depth: also known as bit depth.

- The number of memory bits required to store colour inform. about a pixel in a digital image.
- Common color depth

→ 1-bit: (2 colors). This depth allow for only two colors, usually black & white.

→ 8-bit: (256 colors). This depth provide 256 colors, which are typically selected from a palette of millions.

→ 16-bit (65,536 colors)

- Greater the bit depth, the more colors can be represented.
- It provide more realistic and accurate representations of image because they can display a wider range of colors and shades.

Color Palette

- It defined a set of colors that are available for use in an image or graphic.

• These colors are typically represented in a digital format using a combination of red, green and blue.

• The palette serves as a reference for choosing colors within an application.

and it helps maintain consistency.

Look-Up Table

- Mainly is used to map ^{transform} input ^{bits} value into desired output ^{pixel} value. And stored in a table.
 - Purpose:
 - Color Correction / ^{according to a predefining function} grading.
 - Gamma Correction [adjust the brightness].
 - Color Space Conversion [RGB to CMYK].
 - Shading and lighting Calculation.
- [Texture mapping]

→ Why we use

- In 256-Colour Mode the PC uses only 8 bits.
- It may use
 - 2 bits for blue
 - 3 bits for green
 - 3 bits for red.
- There is chance the most of the colours of a picture are not present.
- In such cases we use a palette or Look-up table.

→ Look-up table approach reduce the storage requirement and containing separate memory block of 256-Color.

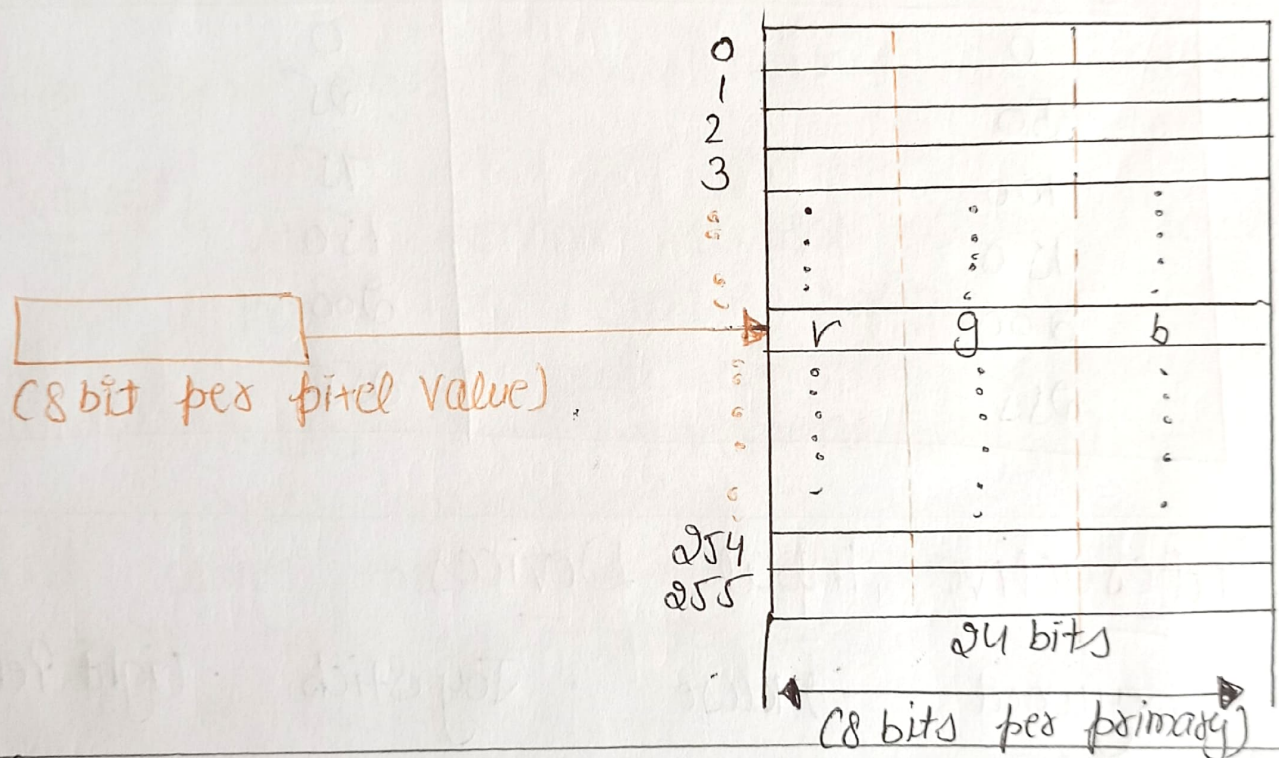
→ In this approach pixel value do not code colors directly, they are addresses or indices into a table of color value.

→ The color of a particular pixel is determined by color value in the table entry. rather than like 0-3 for blue 0-7 for red/green.

- The entries have address $0 - 255$. Each entry contains a 24-bit RGB color value.
- The color of a pixel whose value is i where $0 \leq i \leq 255$

↳ determined color value in table entry whose address is i

- It reduce the storage requirement of a 1000×1000 image to one billion byte plus 768 bytes for the color value.



Advantages

- Efficiency** - allow for efficient computation of complex transformation.
- Consistency** - It ensure consistent result across different platforms & software.
- Flexibility** - provide flexibility in applying various transformation to image or video without the need to modify the original data.

Ex: implementing a simple color correction.

- Suppose we have an image and apply a color filter.
- directly manipulate each pixel in RGB value and use a lookup table to map the original color to new colors based on the predefined value.

eg: grayscale image with pixel 0 to 255.

<u>Input</u> (Original Pixel value)	<u>Output</u> (Transformed)
0	0
50	25
100	75
150	150
200	200
255	255

Interactive Input Devices

- keyboard
- Mouse
- Joy Stick
- Light Pen
- Scanner

These are the important input Devices.

1) Keyboard: It helps to input data to the computer. The layout of keyboard is like that of typewriters.

- These are some additional keys provided for performing additional function.

- keyboards are of two sizes

84 keys → 101/102 keys

- but now keyboards with 104 keys or 108 keys are also available for windows.
- The key on the keyboard

→ Typing keys [include the letters (A-Z) & digits (0-9)]
 → Numeric keypad [used to enter numeric data (17 keys)]
 → Function key [12 keys are present which are arranged in row at top]
 → Control keys [provide cursor & screen control]

2) Mouse:

very famous cursor-control device having a small palm size box with a round ball at its base, which senses the movement of mouse & sends corresponding signals to CPU when the mouse buttons are pressed

→ It has two buttons called the left and right button

→ used to control the position of cursor on the screen, but it cannot be used to enter text into the computer.

- easy to use
- Not very expensive
- Move the cursor faster than the arrow keys of keyboard.

3) Joystick :-

Joystick is also a pointing device, which is used to move the cursor position on a monitor screen.

- It is stick having a spherical ball at its both lower and upper ends.

→ Lower Spherical balls:- moves in a socket.

- The joystick can be moved in all four directions.
- Mainly used in Computer Aided Designing and playing Computer games.

4) Lightpen :-

Similar to pen, used to select a displayed menu item or draw picture on monitor screen.

- Contain a photocell and optical system placed in small tube.

5) Digitizer :-

It converts analog information into digital form.

- also known as tablet or graphics table.
- allow to input drawing, sketches and notes into a computer.
- consist of a flat surface and pen-like instrument that use to draw or write on screen.

- 6) Scanner :-
- work like a photocopy machine
 - used when some information is available on paper and it is to be transferred to the hard disk of the computer
 - Scanner capture image from the source which are then converted into a digital form that can be stored on the disk.

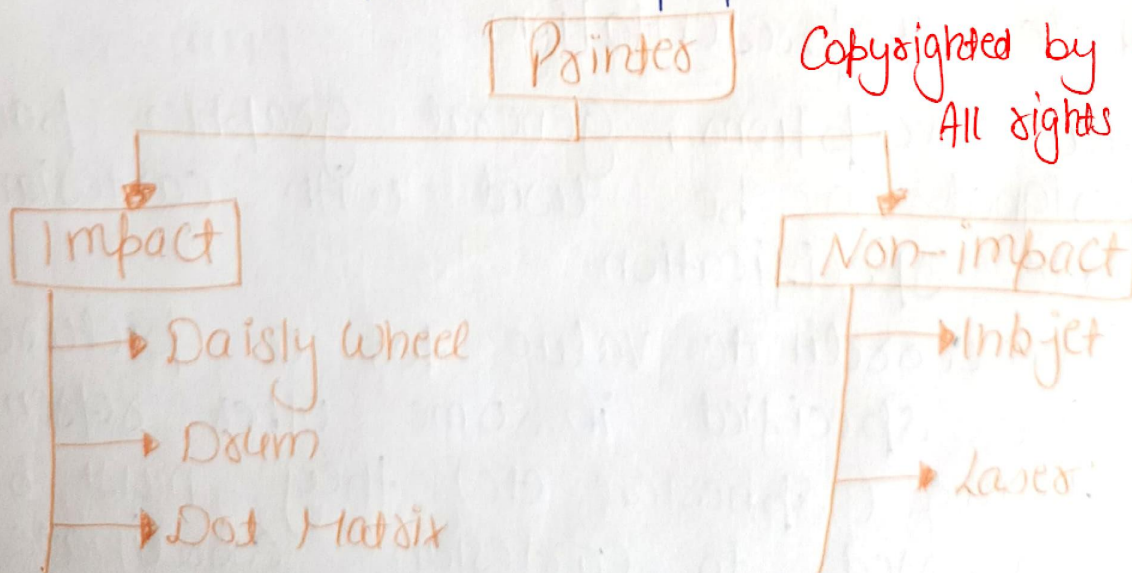
- 7) Bar Code Reader :-
- It is a device used for reading bar coded data.
- generally used in labelling goods, numbering the book etc.

Output devices

The computer give instructions and data from input devices and processes it and return the result.

ex: Printer, Plotter, Monitor.

- Printer :- use to represent the graphics or text on paper.



Copyrighted by Codecap.io
All rights reserved.

Graphic Software

- It refers to a wide range of application and tools used to create, manipulate and render visual content on a computer.
- There are two general classification for graphic software

General purpose package

- It provide an extensive set of graphics function that can be used in high level prog- language such as C or FORTRAN.
- Basic function include for generating picture components (circle, polygon), setting color & intensity value.
- Ex:- Graphics library

Special-purpose

- Those application package are customize for particular application which implement required facility and provide interface so user can use the graphic package without knowing the inner details.
- Ex:- CAD

Coordinate Representation

- With few exceptions, general graphics package are designed to be used with cartesian coordinate specification.

→ If coordinate value for a picture are specified in some other reference frame (spherical etc) they must be converted to cartesian coordinates before they can be input to the

graphics package.

- Special-purpose package may allow use of other coordinate which suits application.

→ Various Co-ordinate representations

1) Modeling Co-ordinate: → also known as local or master Co-ordinate.

- we can construct shape of object with separate coordinate system

2) World Co-ordinate: → Once individual object shapes has been specified, we can place the object into appropriate positions within the scene using a reference frame.

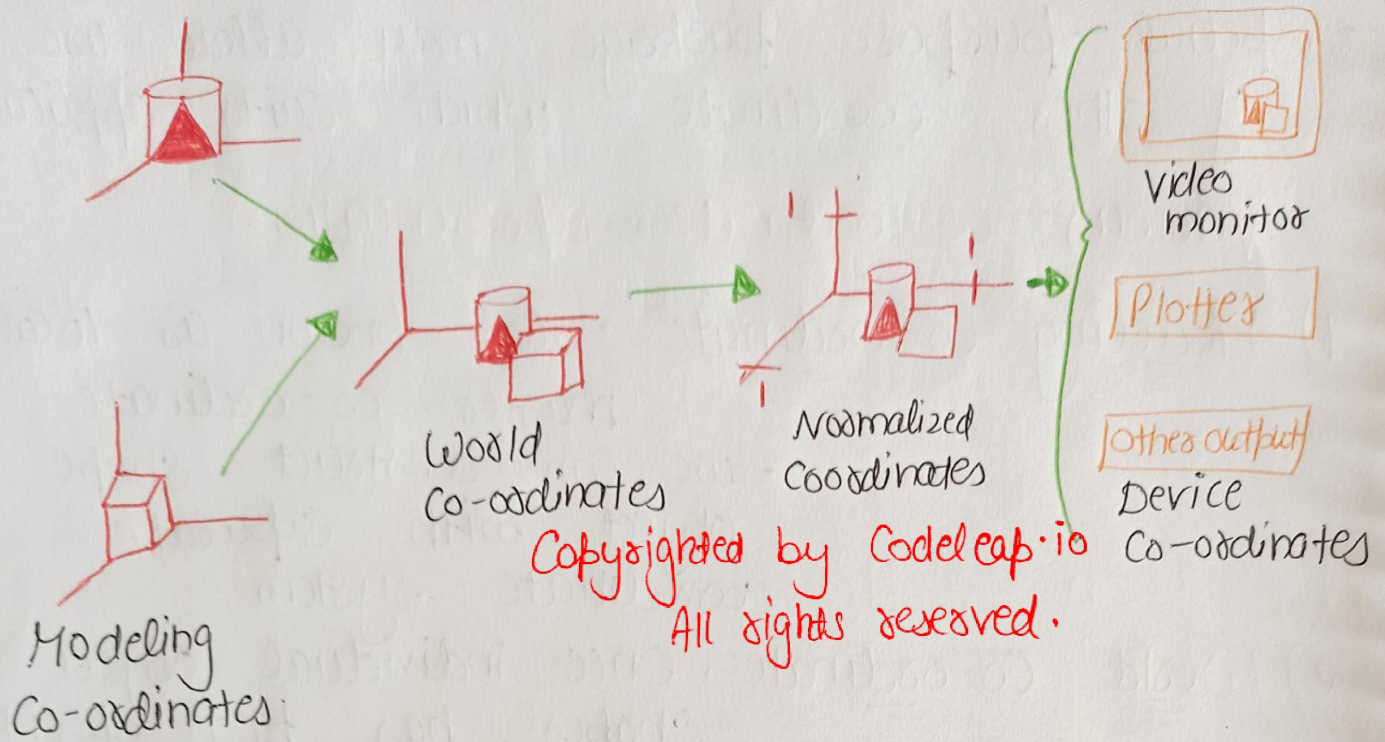
3) Device Co-ordinate: finally, the world Co-ordin. description of the scene is transferred to one or more output-device reference frames for display.

↓
on case video monitor
[Screen co-ordinates]

- a graphics system first convert world-coordinates to normalized device coordinate. in the range from 0 to 1.

$$\left[(x_{mc}, y_{mc}) \rightarrow (x_{wc}, y_{wc}) \rightarrow (x_{nc}, y_{nc}) \rightarrow (x_{dc}, y_{dc}) \right]$$

↓
Seq. of Co-ordinate transformation
from modeling co-ordinates position
(x_{mc}, y_{mc}) to device co-ordinates
(x_{dc}, y_{dc}).



Graphics function

- A general - purpose graphics package provide users with a variety functions for creating and manipulating devices pictures.
- These routine can be categorized according to wheather they deal with output, input, attributes, transformation, viewing.

→ Output primitives :

This is the basic building block for pictures.

- include character string and geometric entities such as points, straight lines.
- These basic tools provide the constructing picture.

→ Attributes :

properties of output primitives, that is, describe how a particular primitive is to be

displayed.

- This include intensity and color specifications, line style, text style.

→ Geometric Transformation / Modeling Transformation

↓
used to change the size, position or orientation of an object within an scene

↓
used to construct a scene using object description given in modeling coordinates.

→ Viewing Transformation

- used to view the complete or some portion of the picture. and also used to specify the position of output display area that is to be used.

→ Structure operations

- A picture is divided into parts.
- Each part is known as object or segment.
- Routines for processing structure carry out operation such as creation, modification and transformation of structures.

→ Other are

	Input function
	Control operation.

Software Standards

- The primary goal of standardized graphics software is portability.

• When packages are designed with standard graphics functions, software can be moved easily.

Point - Plotting Techniques

- The term pixel is a short form of the picture element. It is also called a point or dot. It is the smallest picture unit accepted by display devices.
- Each particular resolution uses 200 scan line, each containing 320 pixels across. This is referred as 320^o200 resolution.

⇒ Scan Conversion / Rasterization

or process of converting geometric shape or object into pixels on computer screen.

- The process of representing continuous graphics object as a collection of discrete pixels.

[Convert lines into pixel]

eg:- a line is defined by

two endⁿ pts line equation.

a) Circle is defined by

radius center position

⇒ Side effects of Scan

1) Aliasing :-

occurs when the appear stairs - steps

↓
[anti-aliasing] techn. reduce

this effect by blending colors

Gaps or overlaps

due to rounding errors or inaccuracies in the scan conversion process, small gaps or overlaps can appear b/w adjacent pixels.

[half-pixel] edges.

Side effects of Scan Conversion

1) Staircase or jagged

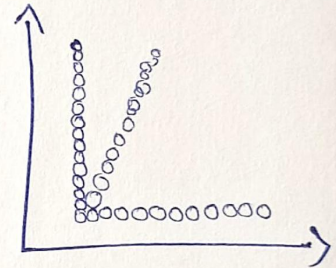
- Staircase like appearance is seen while scan converting line or circle.

2) Unequal Intensity

- an unequal intensity or brightness appears along different lines.

eg: an inclined line (or diagonal line) appear less bright as compared to horizontal and vertical line

- Side effects of scan conversion are also called aliasing.



↳ distortion of information due to low frequency.

- The method used to cancel aliasing is called as antialiasing.

↳ Postfiltering, prefiltering alg. are used for antialiasing.

2) Unequal intensity: This effect can occur due to inaccuracies in scan conversion process such as rounding errors or shading value b/w pixels.

↓
interpolation

3) Overstrike: occur when multiple object or layer overlap in the scene, resulting in unintended blending of pixel information.

↓
depth -
buffering

⇒ Advantage of developing alg. for scan conversion

- 1) generate graphics object at faster rate.
- 2) memory can be used efficiently.
- 3) Alg. can develop a higher level of graphical objects

⇒ Ex. of developing object which can be scan converted

1) Point

2) Line

3) Ellips

4) Poly

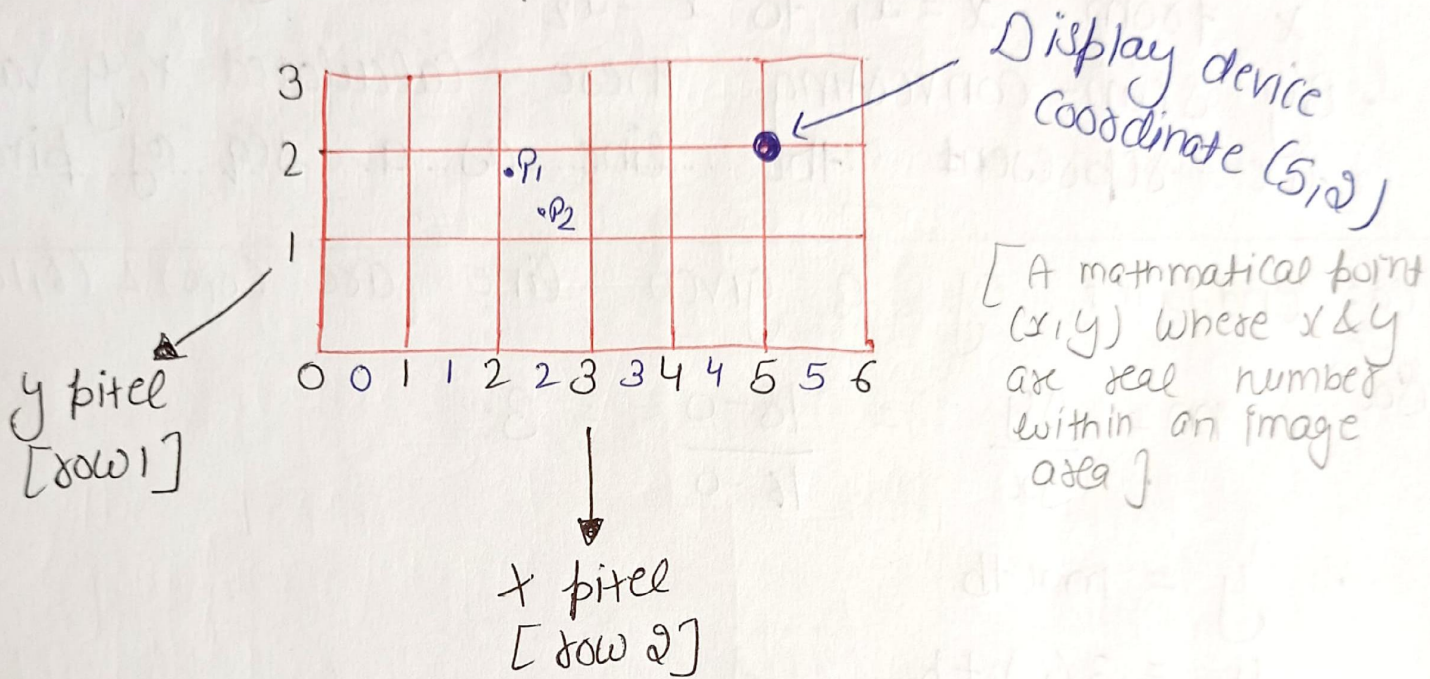
5) filled

In case of line, it can be possible that any pixel may have any floating value like (2.715) which will not be considered by the system in scan conversion. According to these assumption (0.5) to plot

Scan - Converting a Point

- Each pixel on the graphic display does not represent a mathematical point, it represents a region which theoretically can contain an indefinite number of points.

eg: $P_1 (x_1, y_1) = (2\frac{1}{4}, 1\frac{3}{4})$ & $P_2 (x_2, y_2) = (2\frac{2}{3}, 1\frac{1}{4})$

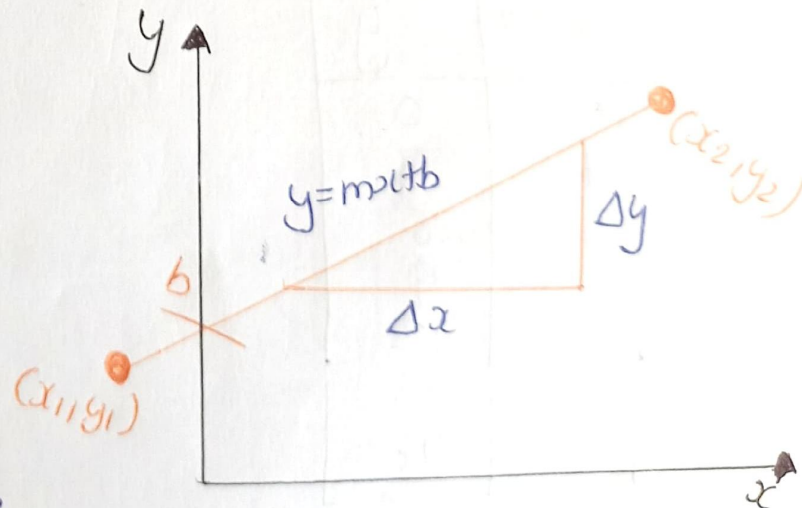


Scan - Converting a Straight Line

- A Straight Line may be defined by two endpoints & an equation.
- (x_1, y_1) & (x_2, y_2)

two — described by
end points

- eqⁿ of line is used to describe the x, y coordinates of all the points that lie b/w these two endpoints.



- Using the eqⁿ of straight line,

$$y = mx + b$$

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

b = the y intercept

- We can find the value of y by increm. x from $x = x_1$ to $x = x_2$.

- By scan-convecting these calculated x, y values we represent the line as a seq. of pixels.

[Direct use of line eqⁿ]

eg: endpoint of a given line are $(0,0)$ & $(6,18)$

$$\text{Sol}^n: m = \frac{\Delta y}{\Delta x} = \frac{18-0}{6-0} = 3$$

$$y = mx + b$$

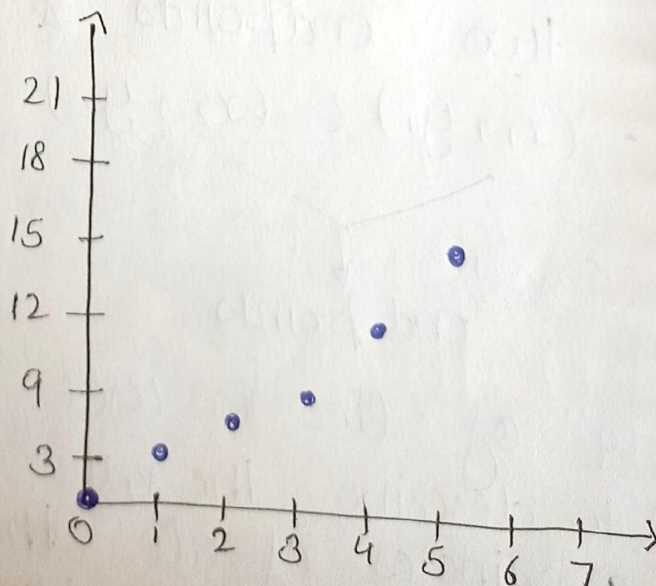
$$y = 3(0) + b$$

$$0 = b$$

Copyrighted by CodeCap.io
All rights reserved.

$$\text{eq}^n \text{ of line} = y = 3x$$

x	y
0	0
1	3
2	6
3	9
4	12
5	15
6	18



Algorithm $\Rightarrow$ Write steps are required to plot a line using the slope method?

Sol: 1) Compute $\underline{dx = x_2 - x_1}$

- $\cdot dy = y_2 - y_1$

- $\cdot m = \frac{dy}{dx}$ [Slop]

- $\cdot b = y_1 - m x_1$ [eqⁿ of line]

2) If $dx < 0$ then

- $\cdot x = x_2, y = y_2$

- $\cdot x_{end} = x_1$

If $dx > 0$ then

- $\cdot x = x_1, y = y_1$

- $\cdot x_{end} = x_2$

3) Test to determine wheather the entire line has been drawn.

if $x > x_{end}$ then Stop.

4) Plot (x, y)

5) increment $x = x + 1$

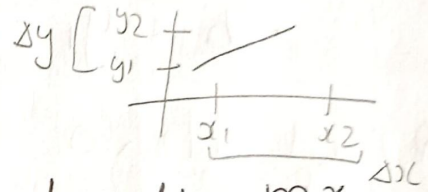
6) Compute the next value of y from eqⁿ $y = mx + b$

7) Go to Step (iii).

- The Simple DDA
- Symmetrical DDA
- Bresenham's Method of Line Drawing

Direct Line Drawing Algorithm

- 1) Two input points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$
- 2) Slope $m = \frac{y_2 - y_1}{x_2 - x_1}$
- 3) Compute y intercept i.e. $b = y_1 - mx_1$
- 4) Initialize point $x = x_1$ and $y = y_1$
- 5) if $|m| \leq 1$ then
 - 1) While $(x \leq x_2)$
 - 2) Increment $x = x + 1$
 - 3) Calculate $y = mx + b$
 - 4) Round off y value
 - 5) Plot pixel (x, y)
 - 6) End While



Else

- 1) While $(y \leq y_2)$
- 2) Increment $y = y + 1$
- 3) Calculate $x = \frac{y - b}{m}$
- 4) Round x value
- 5) Plot pixel (x, y)
- 6) End While.

[$|m| = 1$ both are incremented]

Ex:

$P_1(3, 4)$

$P_2(9, 5)$

$$\rightarrow m = \frac{1}{6}$$

$\rightarrow$ if $m \leq 1$

$$\rightarrow x = x + 1$$

$$y = mx + b$$

Simple DDA Algorithm

used for drawing line on display device.

- DDA stand for "Digital Differential Analyzer"
- also known as incremental Method of scan conversion, In this algorithm, we can perform the calculation in a step by step manner. We use the previous step result in next step.

→ General eqⁿ of line

$$y = mx + c$$

→ here m is slope

$$m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

$(x_b, y_b) \Delta (x_{b+1}, y_{b+1})$
↓
points.

• Main purpose :- efficiently calculate and draw straight line b/w two end point in 2-D space.

- Now, we find the slope b/w starting point and ending point, three case to discuss.

Case 1: if $m < 1$

$$x = x + 1$$

$$y = y + m$$

next point ← → current point

$$[x_{b+1} = x_b + 1]$$

$$[y_{b+1} = y_b + m]$$

Case 2:

if $m = 1$

$$x = x + 1 \quad [x \text{ moves in unit interval}]$$

$$y = y + 1 \quad [y \text{ "}]$$

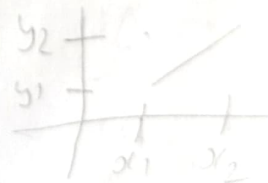
x_b	y_b	x_{b+1}	y_{b+1}	round off

Case 3:

if $m > 1$

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

$$y = y + 1$$



→ Algorithm

- 1) Two input points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$
- 2) Set $dx = (x_2 - x_1)$ and $dy = (y_2 - y_1)$
- 3) Slope $m = \frac{dy}{dx}$
- 4) Initial point $x = x_1$ and $y = y_1$

If $|m| \leq 1$ then

- i. While $x \leq x_2$
- ii. Increment $x = x + 1$
- iii. Calculate $y = y + m$
- iv. Round off y value
- v. Plot pixel (x, y)
- vi. End while

Else

- i. While $y \leq y_2$
- ii. Increment $y = y + 1$
- iii. Calculate $x = x + \frac{1}{m}$
- iv. Round off x value
- v. Plot pixel (x, y)
- vi. End while

→ Example Use DDA Alg. to draw a line from $(2, 3)$ to $(9, 8)$.

Solⁿ We have two coordinates

Starting point = $(2, 3)$

Ending point = $(9, 8)$

Step 1: first we calculate Δx , Δy and m

$$\begin{aligned} \bullet \Delta x &= x_2 - x_1 = 9 - 2 = 7 \\ \bullet \Delta y &= y_2 - y_1 = 8 - 3 = 5 \\ \bullet m &= \frac{y_2 - y_1}{x_2 - x_1} = \frac{5}{7} = 0.71 \end{aligned}$$

$$\boxed{m = 0.71}$$

Step 2: initial point $x = x_1$ & $y = y_1$
 $\boxed{x = 2}$ $\boxed{y = 9}$

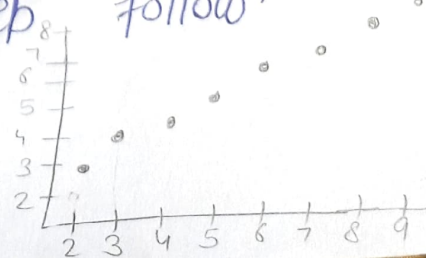
Step 3: if $|m| < 1$, Case 1 is satisfies
 i.e. $x = x + 1$
 $y = y + m$

Step 4: increase x and Calculate y .
 $x = x + 1$
 $y = y + \left(\frac{5}{7}\right)$

x	y	Pixel Plotted
2	3	2, 3
3	$3.71 = 4$	3, 4
4	$4.42 = 4$	4, 4
5	$5.13 = 5$	5, 5
6	$5.84 =$	6, 6
7	$6.55 =$	7, 7
8	$7.26 =$	8, 7
9	$7.97 =$	9, 8

• alg. Stop, we have reach end point
 of line (9, 8)
 • No need to next Step follow.

Plot
 $\rightarrow$



Advantage

- easy to implement
- avoid using the multiplication operation.
- faster Method

Disadvantage

- There is an extra overhead of using round off() function [increase time complexity]
- Resulted line are not smooth because round off()
- The points generated by this alg. are not accurate

Symmetrical DDA

- It is a variation of that takes into account of the line to optimize.
- It is useful for lines $-1 \leq m \leq 1$

• DDA generates lines from their differential eqⁿ $\frac{dy}{dx} = \frac{\Delta y}{\Delta x}$

• The DDA works by stepping through each pixel along the line path and setting pixel intensity value.

[Time consuming]

→ Algorithm

- 1) Two input points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$
- 2) Set $dy = y_2 - y_1$ $dx = x_2 - x_1$
- 3) Number of steps $N = \max(|dx|, |dy|)$
- 4) Calculate increments:

$$1) dx_sym = \frac{dx}{N}$$

$$2) dy_sym = \frac{dy}{N}$$

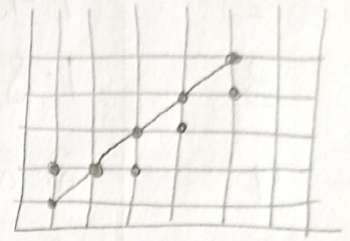
5) initialize $x = x_1$ and $y = y_1$ (1,1)

6) Plot pixel at $[round(x)]$
 $[round(y)]$

7) Update x and y :

$$x = x + dx - sym$$

$$y = y + dy - sym$$



→ Example: Use Symmetrical DDA for drawing a line from (1,1) to (8,4)

Solⁿ: Starting coordinates = 1,1
Ending coordinates = 8,4

Step 2: $dy = 4 - 1 = 3$
 $dx = 8 - 1 = 7$

Step 3: No. of Steps: $\max(dx, dy)$
 $= \max(7, 3)$
 $= \boxed{N = 7}$

Step 4: Increment
 $dx - sym = \frac{7}{7} = 1$
 $dy - sym = \frac{3}{7} = \underline{0.43}$

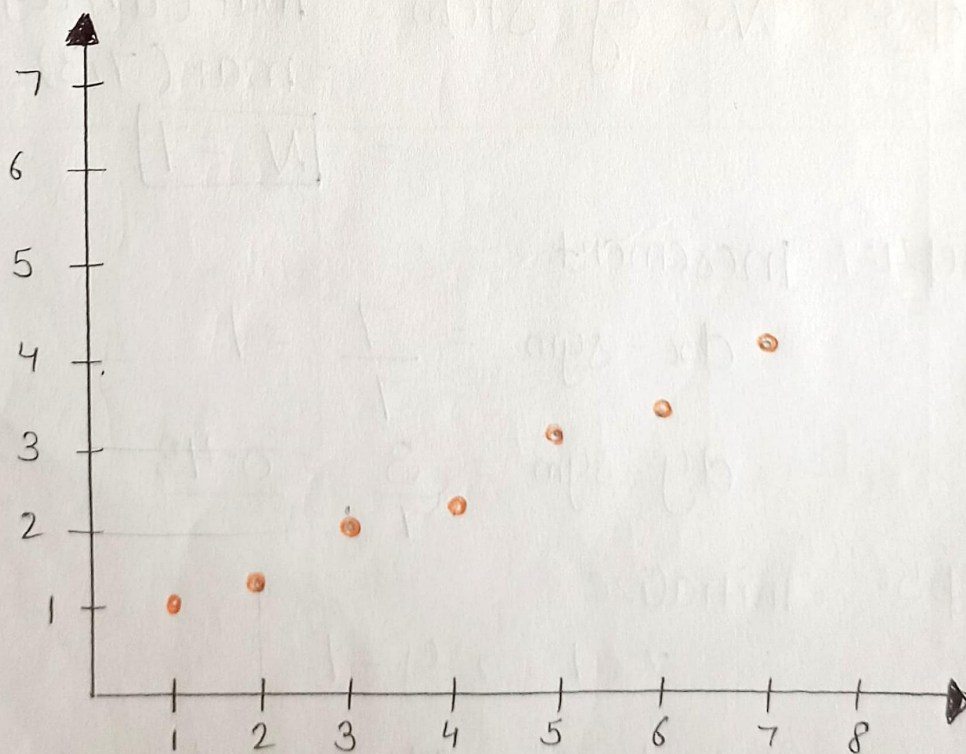
Step 5: Initialize
 $x = 1$, $y = 1$

Step 6: Plot pixel at round x, y
update x & y .

x	y	x (rounded)	y (rounded)	Pixel
1	1	1	1	(1,1)
2	1.43	2	1	(2,1)
3	1.86	3	2	(3,2)
4	2.29	4	2	(4,2)
5	2.72	5	3	(5,3)
6	3.15	6	3	(6,3)
7	3.58	7	4	(7,4)

$$\left[\begin{array}{l} x = x + dx - sym \\ x = 1 + 1 \\ x = 2 \end{array} \quad \begin{array}{l} y = y + dy - sym \\ y = 1 + 0.43 \\ y = 1.43 \end{array} \right]$$

Same procedure apply.



Bresenham Line Drawing Alg

[Jack E. Bresenham]

- efficient Method because it involves only integer addition, subtraction and multiplication operations. [used for scan converting a line]
- These operations can be performed very rapidly so lines can be generated quickly
- In this method, next pixel selected in order to form a close approximation to a straight line b/w two points.

→ Algorithm

- 1) Two input points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$
- 2) Set $dx = (x_2 - x_1)$
 $dy = (y_2 - y_1)$
- 3) Initial decision parameter $p = 2dy - dx$
- 4) Slope $m = \frac{dy}{dx}$
- 5) Initial point $x = x_1$ and $y = y_1$

If $|m| < 1$ then

I. While $x \leq x_2$

II. If $p < 0$ then

III. Increment $x = x + 1$

IV. $p = p + 2dy$

V. Else

VI. Increment $x = x + 1$

VII. Increment $y = y + 1$

VIII. $p = p + 2dy - 2dx$

IX. End if

X. Plot pixel (x, y)

XI. End while

Else ($m > 1$)

[$P > 0$]

I. While $y \leq y_2$

II. If $P < 0$ then

III. Increment $y = y + 1$

IV. $P = P + 2dx$

V. Else

VI. Increment $y = y + 1$

VII. Increment $x = x + 1$

VIII. $P = P + 2dx - 2dy$

IX. End if

X. Plot pixel (x, y)

XI. End While

→ Example Digitize the line with end points $(20, 10)$ and $(30, 18)$ using BTA

Sol:

- Starting Coordinates = $(20, 10)$
- Ending Coordinates = $(30, 18)$

Step 1: find Δx & Δy

$$dx = x_2 - x_1 = 30 - 20 = 10$$

$$dy = y_2 - y_1 = 18 - 10 = 8$$

Step 2: Calculate decision parameter

$$P_k = 2dy - dx$$

$$= 2(8) - 10$$

$$= 16 - 10$$

$$\boxed{P_k = 6}$$

Step 3: $m = \frac{8}{10} = 0.8$

Step 4: as $m \leq 1$ condⁿ satisfied

$$P_k < 0$$

$$x = x + 1$$

$$P = P + 2dy$$

Else

$P_k > 0$ follow

as $P_k > 0$ condⁿ satisfied.

$$x = x + 1$$

$$y = y + 1$$

$$P = P + 2dy - 2dx$$

Copyrighted by Codecap.io
All rights reserved.

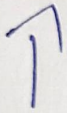
$$x = 20 + 1 = 21$$

$$y = 10 + 1 = 11$$

$$P = 6 + 16 - 20$$

$$22 - 20$$

$$P = 2$$



Same procedure
apply to
find Pixel Point

P_k	P_{k+1}	x_{k+1}	y_{k+1}	Pixel.
		20	10	(20, 10)
6	2	21	11	(21, 11)
2	-2	22	12	(22, 12)
-2	14	23	12	(23, 12)
14	10	24	13	(24, 13)
10	6	25	14	(25, 14)
6	2	26	15	(26, 15)
2	-2	27	16	(27, 16)
-2	14	28	16	(28, 16)
14	10	29	17	(29, 17)
10	5	30	18	(30, 18)

Adv:

- involve only integer arithmetic
- point generated by this alg. More accurate.
- avoid generation of duplicate points.

Dis

- This alg. is for basic line Drawing
- not handle diminishing jaggies.
- The resulted draw line is not smooth.

- This technique provides an efficient method for scan-converting line whose slopes are less than 45°

→ Algo whose slope is b/w 0° & 45°

1) Calculate the initial value

$$dx = x_2 - x_1$$

$$dy = y_2 - y_1$$

$$\text{Inc}_1 \quad p_1 = 2^\circ dy$$

$$\text{Inc}_2 \quad p_2 = 2^\circ (dy - dx)$$

$$p_2 = \underset{\text{Inc}_1}{p_1} - dx$$

2) If $dx < 0$ then

$$x = x_2, y = y_2$$

→ Algo: Whose slope is b/w 0° & 45°

- Obtain the coordinates of two endpoints of the line (x_1, y_1) & (x_2, y_2) .
- Calculate slope of line

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

- Initial Value.

$$\rightarrow dx = x_2 - x_1$$

$$\rightarrow dy = y_2 - y_1$$

$$\rightarrow p = 2^\circ dy - dx$$

$$\rightarrow \text{Inc}_1 = 2^\circ dy$$

$$\rightarrow \text{Inc}_2 = 2^\circ (dy - dx)$$

- If $dx < 0$ then

$$\rightarrow x = x_2$$

$$\rightarrow x_{\text{end}} = x_1$$

$$\rightarrow y = y_2$$

Scan - Converting a Circle

• If $dx > 0$ then

$$\rightarrow x = x_1$$

$$\rightarrow x_{end} = x_2$$

$$\rightarrow y = y_1$$

• Plot (x, y)

• Test to see whether the entire line has been drawn.

if $x > x_{end}$ then stop.

• Compute the location of next pixel.

if $d < 0$ then

$$\rightarrow p = p + inc_1$$

$$[p_{k+1} = p_k + inc_1]$$

else

$$\rightarrow p = p + inc_2 \text{ \& } y = y + 1$$

• Increment $x = x + 1$

• Plot (x, y)

• Go to Step (iv).

→ This alg. efficiently plot lines with a slope b/w 0° and 45° by making decision based on incremental calculation rather than using floating point. making the alg. efficient for real-time.

→ Incremental updating minimizes the number of calculations needed to plot the line.

→ Alg. b/w 45° & 90° using Bresenham's Method

• Input Coordinates: (x_1, y_1) & (x_2, y_2)

• Calculate Slope: $m = \frac{y_2 - y_1}{x_2 - x_1}$

• Compute the initial value:

$$\rightarrow dx = x_2 - x_1$$

$$\rightarrow dy = y_2 - y_1$$

$$\rightarrow inc_1 = 2^{\circ} dy$$

$$\rightarrow inc_2 = 2^{\circ} (dy - dx)$$

$$\rightarrow p = inc_1 - dx$$

• If $dy < 0$ then

$$\rightarrow y = x_2, x = y_2, x_{end} = y_1$$

If $dy > 0$ then

$$\rightarrow y = x_1, x = y_1, x_{end} = y_2$$

(x, y)
Starting

→ more possible of x value.

• Plot (y, x)

• Test to determine whether the entire line has been drawn.

If $x > x_{end}$ then Stop.

• Compute the location of next pixel.

If $d < 0$ then

$$p = p + inc_1$$

else

$$p = p + inc_2, y = y + 1$$

$$x = x + 1$$

• Plot (y, x) .

Copyrighted by Codecap.io
All rights reserved.

→ How decision parameters for Bresenham line Drawing alg. is determined?

→ The decision parameter is determined to decide the next pixel to plot while drawing a line b/w two given points.

- 1) Starting point
- 2) Slope of line
- 3) decision parameter (p). [based on slope of line]
- 4) Incrementing the Decision parameter
- 5) Drawing the line

→ How Bresenham better than DDA

- both are used for drawing line.
- Bresenham is prefer due to its efficiency and accuracy.

1) Efficiency :- It use only integer arithmetic, while DDA requires floating-point calculation to determine the increment value of x & y .

2) Accuracy :- It provide better accuracy compared to DDA, especially when drawing line with steep slopes.

- It chooses the nearest pixel to true line, resulting better accuracy.
- but DDA may suffer from rounding error. leads to inaccuracy.

3) Memory requirement: less than DDA

Difference blw DDA & Bresenham

DDA

- uses floating point, i.e. real arithmetic
- It uses multiplication & division operation
- Slower than Bresenham's line Algo because it uses real arithmetic.
- Less efficient
- It can draw circle & curves but are not accurate i.e. less accuracy

Bresenham

- used fixed point i.e. integer arithmetic.
- uses only addition operation
- faster than.
- More efficient
- Can draw circle & curves with more accuracy than DDA algorithm

How point is represented in a Cartesian Coordinate System.

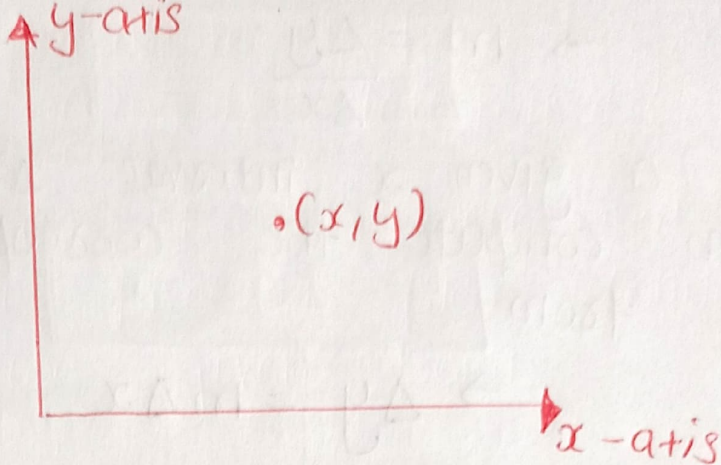
- It is represented by using two co-ordinates
 - x - coordinate
 - y - coordinate
- x - coordinate represents the horizontal position i.e. abscissa
- y - coordinate represents the vertical position i.e. ordinate.
- These coordinates determine the exact location of point within the Cartesian plane.

eg: (3, 4)

points coordinates.

→ i.e means 3 units
located to right

→ 4 units above
the x-axis.



• Horizontal axis

• Vertical axis

• origin
(where x & y intersect)

• Quadrants

• representation.

Line Drawing Algorithms

• A line can be defined as a straight one-dimensional. It has no thickness and can be extended endlessly in both directions.

• The line drawing algorithm is a graphical algorithm which is used to represent the line segment on discrete graphical media.

→ Eqⁿ of straight line

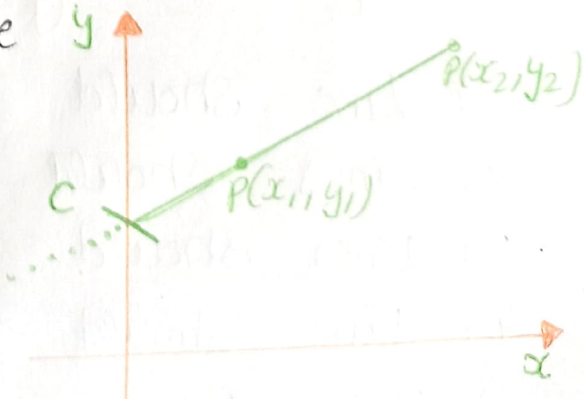
$$y = mx + c$$

Where

- (x, y) = axis of the line
- m = Slope of line
- c = y intercept

• We can determine the value for slope m .

$$\rightarrow m = \frac{y_2 - y_1}{x_2 - x_1}$$



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

- for a given x interval Δx along a line, we can compute the corresponding y interval Δy from

$$\rightarrow \Delta y = m \Delta x \quad [m < 1]$$

- lly, we can obtain x interval Δx by Δy :

$$\rightarrow \Delta x = \Delta y / m \quad [m > 1]$$

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

- if $|\Delta x| > |\Delta y|$
then $\Delta x = 1$

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

$$\rightarrow x_{i+1} = x_i + 1$$

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

$$= y_i + m \Delta x$$

$$y_{i+1} = y_i + m$$

- if $|\Delta x| < |\Delta y|$
then $\Delta y = 1$

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

$$\rightarrow x_{i+1} = x_i + \frac{\Delta y}{m}$$

$$x_{i+1} = x_i + \frac{1}{m}$$

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

$$y_{i+1} = y_i + 1$$

Properties of good Line Drawing Algo.

- 1) Line should appear straight.
- 2) Lines should terminate accurately.
- 3) Lines should have constant density.
- 4) Line should be drawn rapidly.
- 5) Line density should be independent of line length & angle.

Requisiments to draw a line

- It involves converting mathematical representation of line into pixel-based representation suitable for display on a screen.

- Two endpoints
- Line eqⁿ or Slope-intercept form.
- Choose increment Direction.
- Algorithm.
- Display device.

Scan - Converting a Circle

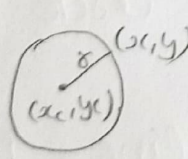
→ Circle:- It is defined as a set of points that all are at same distance (r) from a common point (x_c, y_c).

- Drawing a Circle

⇒ To draw a circle we need two things.

- Coordinates of centre (x_c, y_c)
- radius of circle (r)

⇒ eqⁿ of circle by Cartesian Co-ordinates

 for any point on the circle (x, y) and the centre at point (x_c, y_c) the eqⁿ of circle is

$$(x - x_c)^2 + (y - y_c)^2 - r^2 = 0$$

- $r =$
- If Circle has origin (0,0) as its centre then the above eqⁿ is

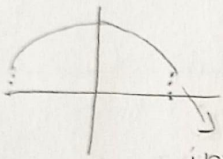
$$x^2 + y^2 = r^2$$

← Standard eqⁿ

- above eqⁿ of Circle, also write as

$$y = \pm \sqrt{r^2 - x^2}$$

- To calculate the position of points on a Circle circumference by stepping along the $(x_c - r) \leq x \leq (x_c + r)$ and calculating the corresponding 'y' value.



$$y = y_c \pm \sqrt{r^2 - (x - x_c)^2}$$

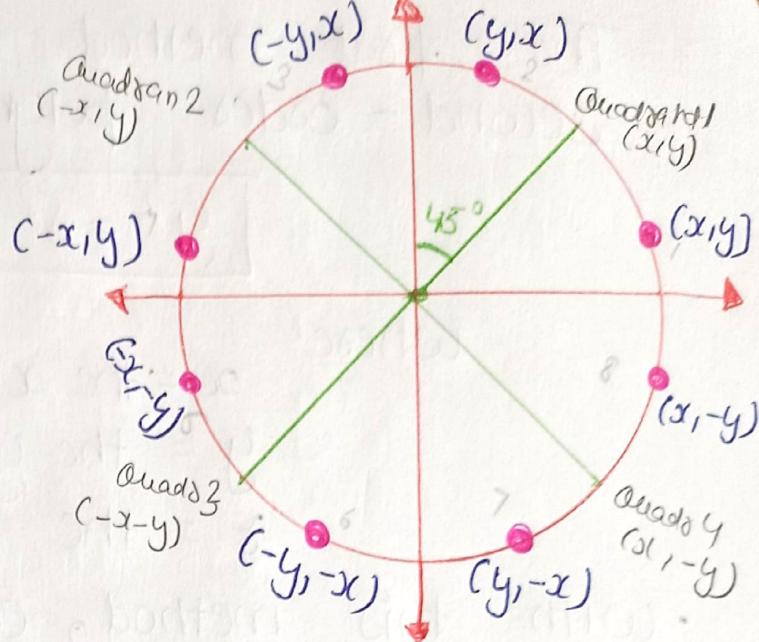
inefficient due to multiplication & square root operation

- ⇒ The circle is divided into 4 quadrants, each quadrant has two octant.
 ↳ So circle is known as eight-way symmetric figure.

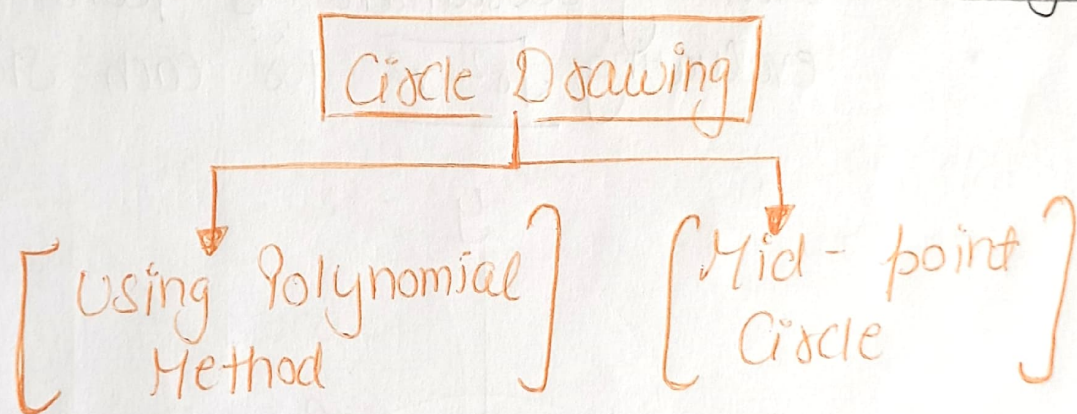
- ⇒ If the calculation of the point of one octant is done, then the other seven points can be calculated easily by using the concept of eight-way symmetry.

- ⇒ for drawing,
 ↳ We consider that circle centred at origin (0,0).
 • If a point (x,y) line in one octant then other easily calculate.
 • We will calculate only 45° arc.
 from each caluted point then whole circle can be determined easily.

- $P_1 = (x, y)$
- $P_2 = (y, x)$
- $P_3 = (-y, x)$
- $P_4 = (-x, y)$
- $P_5 = (-x, -y)$
- $P_6 = (-y, -x)$
- $P_7 = (y, -x)$
- $P_8 = (x, -y)$



→ also These seven points are calculated by using the property of reflection
 ↳ reversing x, y coordinates



⇒ Defining a circle using Polar Co-ordinates
 ⇒ Bresenham's Circle Algorithm

← Important

Defining a Circle Using Polynomial

- The first method defines a circle with the second-order polynomial eqⁿ.

$$y^2 = r^2 - x^2$$

Where

x = the x coordinate

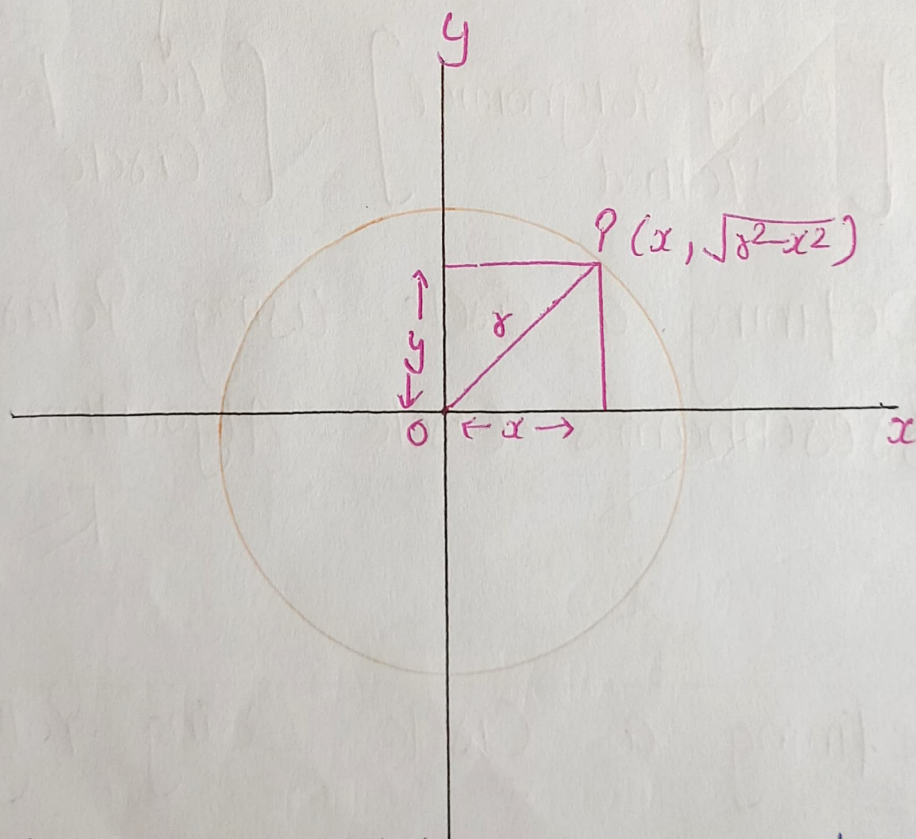
y = the y coordinate

r = the circle radius.

- With this method, each x coordinate in the sector, from 90° to 45° , is found by stepping

→ x from 0 to $\frac{r}{\sqrt{2}}$

→ each y coordinate is found by evaluating $\sqrt{r^2 - x^2}$ for each step of x .



- This is very inefficient method, because for each point both x and r must be squared and subtracted from each other,

⇒ Steps to generate a circle, using Polynomial

Step 1: Set the initial variable

$r = \text{Circle radius}$

$(h, k) = \text{Coordinate of Circle Center}$

$x = 0$

$i = \text{Step Size}$

$x_{\text{end}} = r/\sqrt{2}$

Step 2: Test the determine whether the entire circle has been Scan - Converted.

if $x > x_{\text{end}}$ then Stop

Step 3: Compute $y = \sqrt{r^2 - x^2}$

Step 4: Plot the eight pts found by symmetry with respect to Centre (h, k) , at the current (x, y) coordinates.

Plot $(x+h, y+k)$

Plot $(-x+h, -y+k)$

Plot $(y+h, x+k)$

Plot $(-y+h, -x+k)$

Plot $(-y+h, x+k)$

Plot $(y+h, -x+k)$

Plot $(-x+h, y+k)$

Plot $(x+h, -y+k)$

Step 5: increment $x = x + i$

Step 6: Go to Step 2.

Copyrighted by CodeLeap.io
All rights reserved.

Defining a Circle using Trigonometric

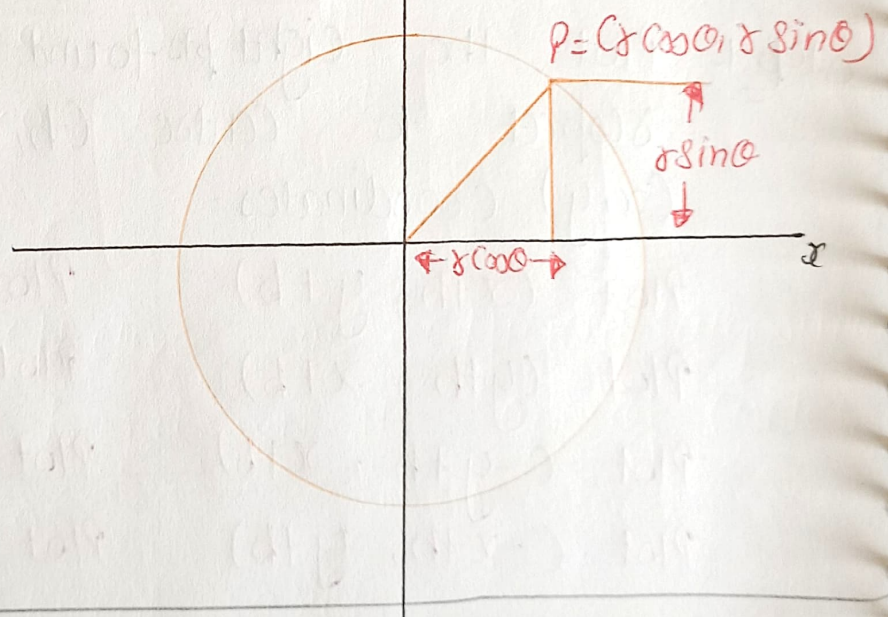
- The Second Method of defining a circle make the use of Trigonometric function.

$$x = r \cos \theta$$

$$y = r \sin \theta$$

Where θ = current angle
 r = Circle radius
 x = x coordinate
 y = y coordinate.

- By this method θ , is stepped from 0 to $\pi/4$ & each value of x and y is calculated.



→ Steps to required scan - converting a circle using Trigonometric Method

Step 1: Set the initial variable

r = Circle radius

(x_1, y_1) = Coordinates of Circle center

i = Step Size

$\theta_{end} = (2\pi/4)$ $\theta = 0$

Step₂ If $\theta > \theta_{end}$ then Stop

Step₃ Compute
 $x = r^p \cos \theta$ $y = r^p \sin \theta$

Step₄ Plot the eight points.

Plot $(x+th, y+tb)$

Plot $(-x+th, -y+tb)$

Plot $(y+th, x+tb)$

Plot $(-y+th, -x+tb)$

Plot $(-y+th, x+tb)$

Plot $(y+th, -x+tb)$

Plot $(-x+th, y+tb)$

Plot $(x+th, -y+tb)$

Step₅ Increment $\theta = \theta + i$

Step₆ Go to Step (iii)

Mid-point Circle Drawing Algorithm

- Mid point Circle Drawing Algorithm attempts to generate the point of one octant and then paint the along with their mirror points in other seven octants.
- Given the centre point and radius of circle
- Circle function with (0,0) as its center

$$f_c(x,y) = x^2 + y^2 - r^2$$

- It has 3 properties, at any point (x,y)
 $f_c(x,y) = \begin{cases} < 0, & \text{if (x,y) is inside the circle.} \\ = 0, & \text{is on the circle} \\ > 0, & \text{is outside the circle.} \end{cases}$

Advantage

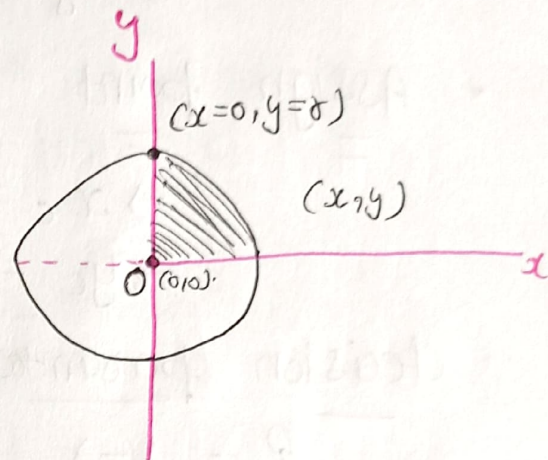
- 1) It is very effective and efficient algorithm.
- 2) This alg. is based on the eqⁿ of circle
$$\underline{x^2 + y^2 = r^2}$$
- 3) It can be implemented easily.

Disadvantage

- 1) This algo. is time consuming.
- 2) There is the problem of accuracy while generated points.

→ Algorithm

Given: Centre point of Circle = (x_c, y_c)
radius of Circle = r



Step 1: Input radius r and
center of Circle (x_c, y_c)

Step 2: Set initial value $x=0, y=r$

Step 3: Draw first point, Plot Circle point $(x+x_c, y+y_c)$

Step 4: Initial decision parameter $p = 1 - r$

I. While $(x \leq y)$

II. If $p < 0$ then

III. Increment $x = x + 1$

IV. Plot Circle point
 $(x+x_c, y+y_c)$

V. Compute $p = p + 2(x) + 3$

Else

I. Increment $x = x + 1$

II. Increment $y = y + 1$

III. Plot Circle pt $(x+x_c, y+y_c)$

IV. Compute $p = p + 2(x-y) + 5$

V. End if

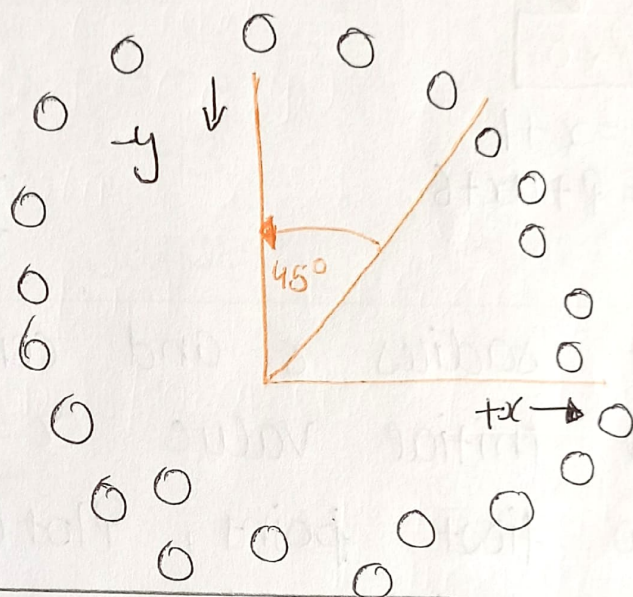
End While

Step 5: Stop

Bresenham's Circle Drawing Algorithm

- This algorithm select nearest pixel position to complete arc.
- It is very significant algorithm and faster than other circle drawing algorithm because it use only integer arithmetic.
- Points are generated from 90° to 45° moves will be made only in $+x$ & $-y$ direction.

If the eight-way symmetry of a circle is used to generate a circle, point will only have to be generated through a 45° angle. and if --



⇒ Advantage

- easy to implement.

• This algo based on the simple eqⁿ of circle.

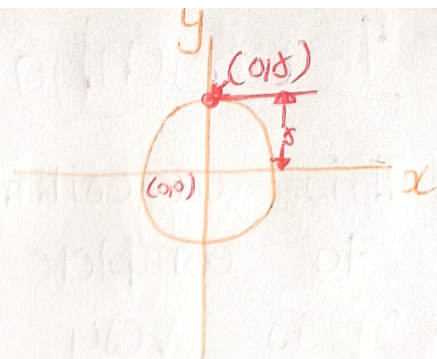
- efficiency
- simplicity
- accuracy
- Memory efficiency

⇒ Disadvantage

- There is the problem of accuracy while generating points
- This alg is not suitable for the complex and high graphic images

• limited precision

→ Algorithm



- If it is assumed that the circle is centred at the origin, then the first step

$$x = 0 \text{ \& } y = 8$$

- initial decision parameter

$$P = 3 - 2x$$

Two cases

$$P_k$$

$$x_{k+1} = x_k + 1$$

$$P < 0$$

$$\rightarrow x = x + 1$$

$$\rightarrow P = P + 4x + 6$$

incr

$$P > 0$$

$$\rightarrow x = x + 1$$

$$\rightarrow y = y - 1$$

$$\rightarrow P = P + 4(x - y) + 10$$

incr

Step 1: Input radius r and centre of circle (x_c, y_c)

Step 2: Set initial value $x = 0, y = 8$ (here $\rightarrow$ coordinates)

Step 3: Draw first point, Plot circle point $(x + x_c, y + y_c)$

Step 4: Initial decision parameters $P = 3 - 2x$

I. while $(x \leq y)$

II. If $P < 0$ then

III. Increment $x = x + 1$

IV. Compute $P = P + 4x + 6$

V. Plot eight points

Plot $(x + x_c, y + y_c)$

Plot $(y + x_c, x + y_c)$

Plot $(-y + x_c, x + y_c)$

Plot $(-x + x_c, y + y_c)$

Plot $(-x + x_c, y + y_c)$

⋮

Else

I. Increment $x = x + 1$

II. $y = y - 1$

III. Compute $P = P + 4(x - y) + 10$

IV. End if

End while

Ex: Given centre point coordinates $(0,0)$ and radius as 8, generates all the points to form a circle.

Sol: • Centre point = $(0,0)$

• $r = 8$

Step1: assign initial point ~~to~~ $[x=0, y=8]$

$$x = 0$$

$$y = 8$$

Step2: initial parameter $p = 3 - 2r$

• $p = 3 - 2(8)$

• $p = -13$

Step4: $p < 0$ condⁿ satisfied.

Thus,

$$x = x + 1$$

$$p = p + 4x + 6$$

Circle drawing Using Polar Coordinates

- The second method of defining a circle makes use of polar Coordinates.

$$x = r \cos \theta$$

$$y = r \sin \theta$$

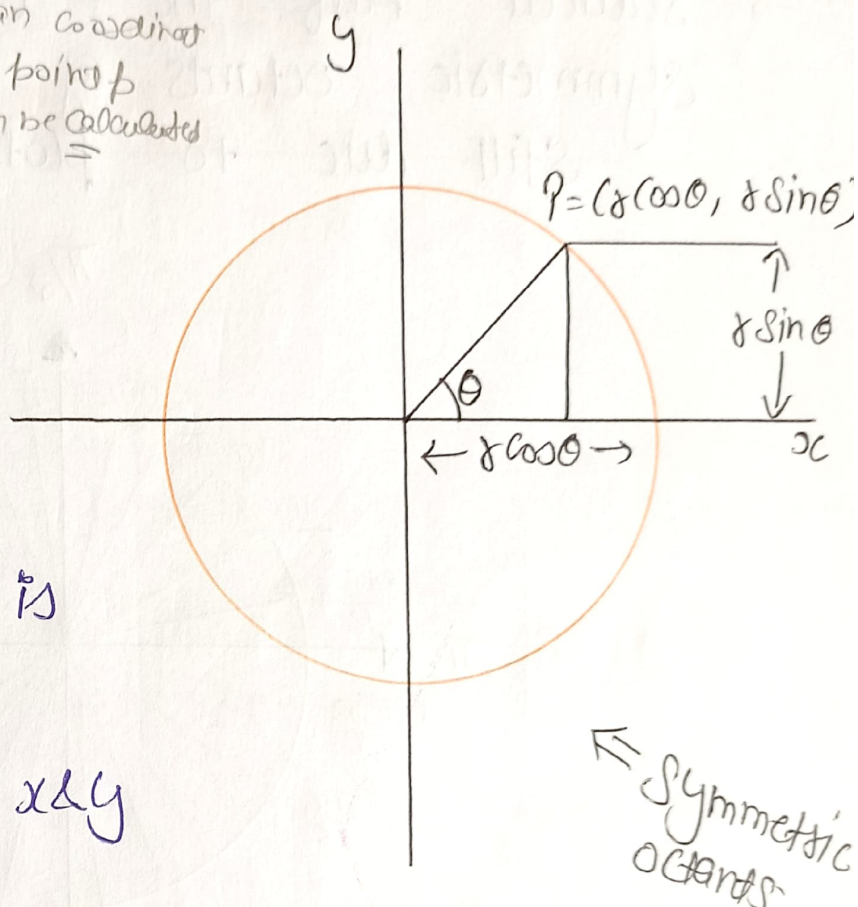
Cartesian coordinates of point P can be calculated =

→ θ = current angle

r = circle radius

x = x coordinate

y = y coordinate



→ By this method, θ is

stepped from 0 to $\frac{\pi}{4}$

and each value of x & y is calculated.

Polar Coordinates :

It is two-dimensional coordinate system that employs distance and angle to represent points on a plane.

→ radius (r)

↳ distance from the origin to point of circle

→ Angular coordinate (θ)

for 0 to 2π

$$x = x_c + r \cos \theta$$

$$y = y_c + r \sin \theta$$

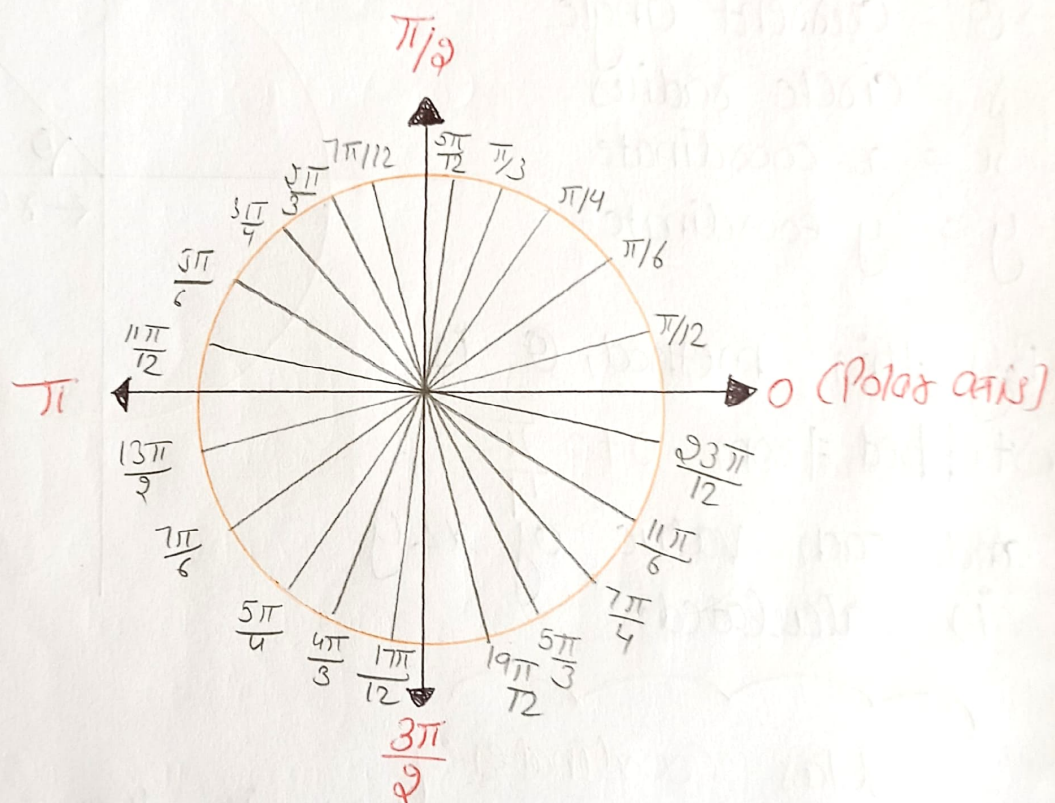
↳ angle b/w 0 to 2π

- It also solves problem of unequal space but in term of calculation involves floating point calculation.

→ but Calculation can be reduced by consider the Symmetry of Circle.

— Take one Quadrant with 45° ,

→ reduced half the Calculation by considering Symmetric octants of circle but inefficiency Still due to plotting - point Calculation.



[The polar Coordinate System]

→ Algorithm

Step 1: Input: Center Coordinates (x_c, y_c) & radius (r)

Step 2: Initialization: Set the angle θ to 0

Step 3: Incremental Step:

- Increment θ by certain angle
- Calculate new point on circle using polar coordinate.

$$\rightarrow x = r \times \cos \theta$$

$$\rightarrow y = r \times \sin \theta$$

• Plot the calculated point.

Step 4: Repeat Step 3 until θ completes a full circle (i.e. 360° degrees).

Ex: Center = (0,0)
radius = 5

Step 1: Set the angle θ to 0.

Step 2: Increment θ by certain angle
 $\theta = 45^\circ$ or $\frac{\pi}{4}$ radians.

• Calculate the new point on circle

$$\rightarrow x = r \times \cos(\theta)$$

$$\rightarrow y = r \times \sin(\theta)$$

1) for $\theta = 0$

$$\begin{aligned} \cdot x &= 5 \times \cos(0) = 5 \\ \cdot y &= 5 \times \sin(0) = 0 \end{aligned}$$

2) for $\theta = \frac{\pi}{4}$

$$\begin{aligned} \cdot x &= 5 \times \cos\left(\frac{\pi}{4}\right) = 3.5355 \\ \cdot y &= 5 \times \sin\left(\frac{\pi}{4}\right) = 3.5355 \end{aligned}$$

3) for $\theta = \frac{\pi}{2}$

$$\begin{aligned} \cdot x &= 5 \times \cos\left(\frac{\pi}{2}\right) = 0 \\ \cdot y &= 5 \times \sin\left(\frac{\pi}{2}\right) = 5 \end{aligned}$$

$$\begin{aligned} \cos 0 &= 1 \\ \sin 0 &= 0 \\ \cos \pi &= -1 \\ \sin \pi &= 0 \\ \cos \frac{\pi}{2} &= 0 \\ \sin \frac{\pi}{2} &= 1 \\ \cos \pi &= -1 \\ \sin \pi &= 0 \\ \cos \frac{3\pi}{2} &= 0 \\ \sin \frac{3\pi}{2} &= -1 \end{aligned}$$

4) for $\theta = \pi$ radians

- $x = 5 \times \cos(\pi) = -5$

- $y = 5 \times \sin(\pi) = 0$

• Points $(-5, 0)$

5) for $\theta = 2\pi$ radians

- $x = 5 \times \cos(2\pi) = 5$

- $y = 5 \times \sin(2\pi) = 0$

• Points $(5, 0)$

→ Advantage:

- versatility $\Rightarrow$ not only used for drawing circle but also for other shape.

- Precision: - It achieve higher precision compared to Bresenham. especially for large circle.

- Smoothness: In many case, interpolating b/w points in polar coordinate can produce smoother curves.

→ Disadvantage

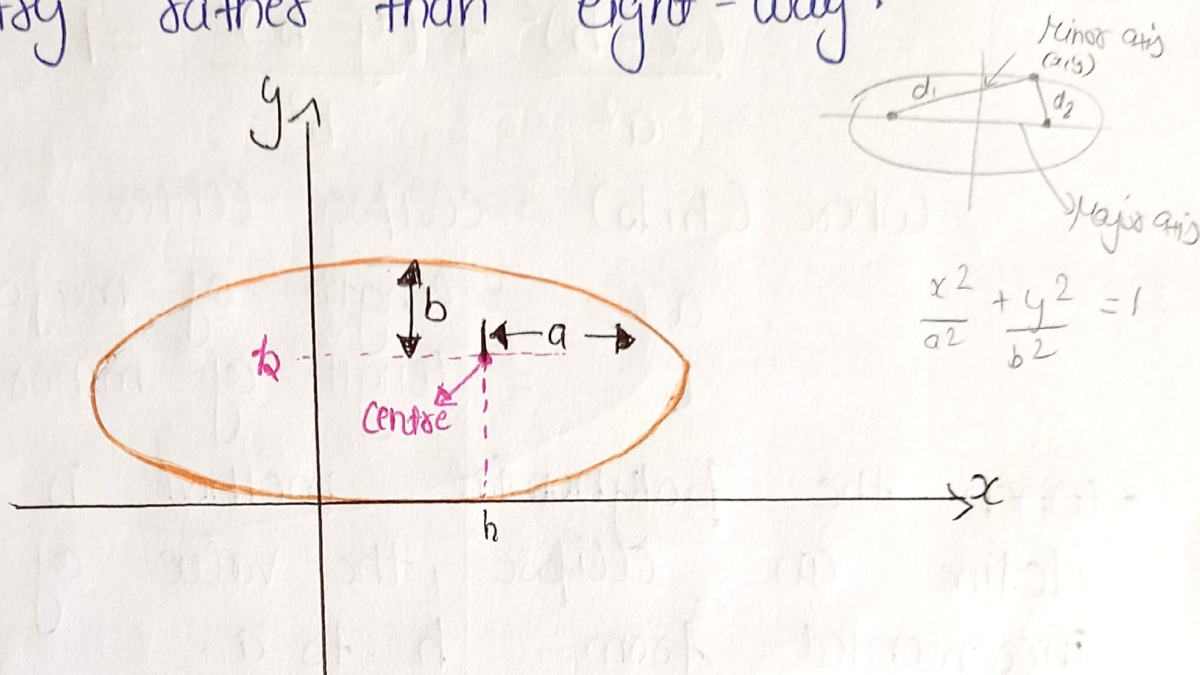
- Complexity

- Computational overhead [calculating trigonometric fun]

- Memory overhead

Scan - Converting a Ellipse

- The ellipse is also a symmetric figure like a circle but it is a four-way symmetry rather than eight-way.



- These are two methods of defining an Ellipse

→ Polynomial Method
 → Trigonometric Method

- The ellipse has a major and minor axis

- Major axis = $2a$

- Minor axis = $2b$

- In case of x-axis, ellipse eqⁿ

$$\boxed{\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1}$$

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1$$

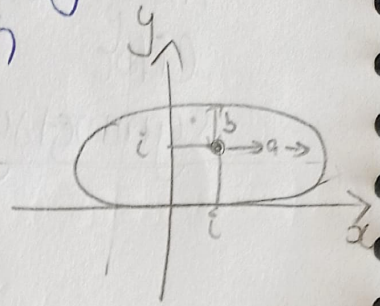
- In case of y-axis :

$$\boxed{\frac{x^2}{b^2} + \frac{y^2}{a^2} = 1}$$

Polynomial Method

- The polynomial method of defining an ellipse using below expression

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1$$



Where (h, k) = ellipse center

a = length of major axis

b = length of minor axis

- When the polynomial method is used to define an ellipse, the value of x is incremented from h to a .
- For each step of x , each value of y is found by evaluating the expression.

$$y = b \sqrt{1 - \frac{(x-h)^2}{a^2}} + k$$

⇒ Drawbacks

- It requires squaring of values. So floating point calculation is required.
- Routine developed for such calculation are very complex and slow.

⇒ Steps

Step 1: Set the initial variable

a = length of Major axis

b = length of Minor axis

$(h, k) = \text{Coordinate of center}$

$$x = 0$$

$$i = \text{StepSize}$$

$$x_{\text{end}} = a$$

Step 2: If $x > x_{\text{end}}$ then Stop

Step 3: Compute $y = b \sqrt{1 - \frac{x^2}{a^2}}$

Step 4: Plot the four points, found by symmetry, at the current (x, y) coordinates

Plot $(x+h, y+k)$

Plot $(-x+h, -y+k)$

Plot $(-x+h, y+k)$

Plot $(x+h, -y+k)$

Step 5: Increment $x = x + i$

Step 6: Go to Step 2.

Copyrighted by CodeCarp.io
All rights reserved.

Trigonometric Method

• A Second Method of defining an ellipse make use of trigonometric.

• The following eqⁿ define an ellipse trigonometric

$$x = a \cos \theta + h$$

$$y = b \sin \theta + k$$

Where (x, y) = the current coordinates

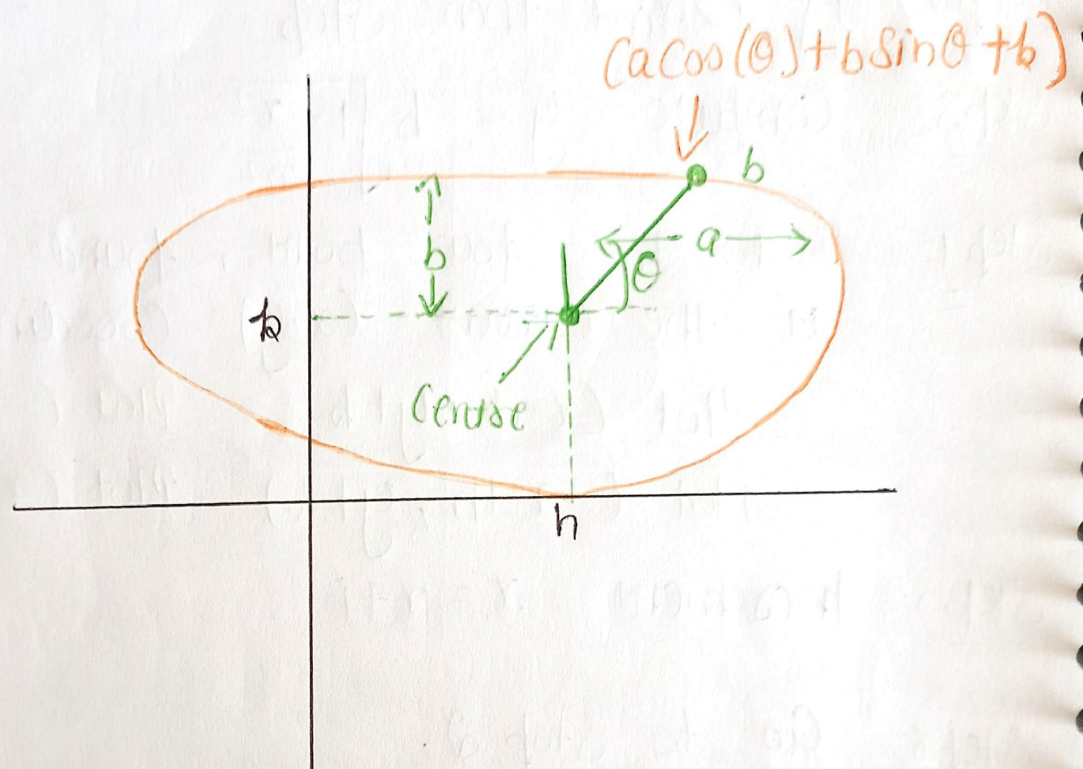
a = length of Major axis

b = length of minor axis

θ = current angle

(h, k) = center of ellipse

- In this method, the value of θ is varied from 0 to $\frac{\pi}{2}$ radians. The remaining points are found by symmetry.



Steps are required to scan convert an ellipse using the trigonometric method

Step 1: Set the initial variables

a = length of major axis

b = length of minor axis

(h, k) = Coordinate of ellipse centres

i = Step size

$\theta_{end} = \frac{\pi}{2}$

$\theta = 0$

Step 2: If $\theta > \theta_{end}$ then Stop.

Step 3: Compute

$$x = a \cos \theta \quad y = b \sin \theta$$

Step 4: Plot the four points, found by symmetry, at the current (x, y) coordinates

$$\text{Plot } (x+h, y+b)$$

$$\text{Plot } (-x+h, -y+b)$$

$$\text{Plot } (-x+h, y+b)$$

$$\text{Plot } (x+h, -y+b)$$

Step 5: Increment $\theta = \theta + i$

Step 6: Go to Step (ii)

Disadvantage

- 1) This is an inefficient method
- 2) It is not an interactive method for generating ellipse
- 3) Memory is required to store the value of θ

Polynomial

- ellipse are represented using polynomial eqn.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

- Simplified representation
- Computational efficiency
- Limited Range
- Parameter sensitivity

Trigonometric

- ellipse are represented using trigonometric function i.e. Sine and Cosine

$$x = a \cdot \cos \theta$$

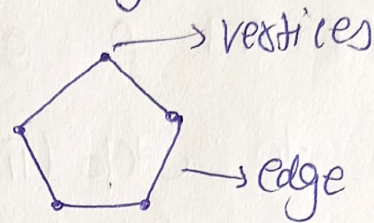
$$y = b \cdot \sin \theta$$

- Accuracy
- Rotation
- Computational overhead
- Parameterization Difficulty

Polygon Area filling

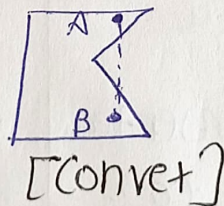
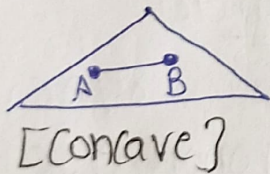
→ Polygon

- A closed figure represented by a collection of more than 2 line segments connected end to end.



→ Type of polygons

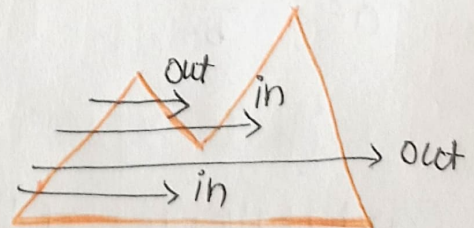
- Concave [Line segment joining any two points within the polygon may not lie completely inside polygon.]
- Convex [Line segment joining any two points within the polygon lies completely inside the polygon.]



→ Polygon Inside Test

→ Even - odd Test

- If the no. of intersection is even then the point is outside.
- If the no. of intersection is odd then the point lies inside.



→ Winding Number Method

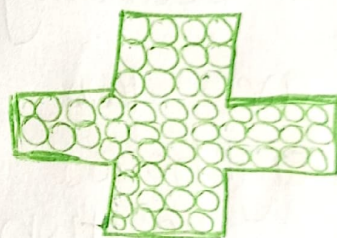
- This is an alternative method for finding the interior points of any polygon
- If the winding number is zero then the point is outside otherwise it is inside.

Region Filling

- Region filling is a process of filling image or region.



[Boundary filled region]



[Interior or flood fill]

- Filling can be boundary or interior region.
- Boundary fill algorithm used to fill the boundary
- Flood - fill algorithm are used to fill interior.

⇒ Boundary fill algorithm

- Boundary fill algorithm starts by accepting the coordinates of an interior point (x, y) , a fill color and a boundary color.

- Starting with some seed, any point inside the polygon, we examine the neighbouring position to check boundary pixel are coloured or not
- If the pixel are already filled or colored then leave
- If they are not coloured or filled the boundary pixel, pixels are highlighted with fill colour, and their neighbours are tested. This process is continued until boundary pixels are filled.
- There are two methods for proceeding to neighbours pixels from the current test position

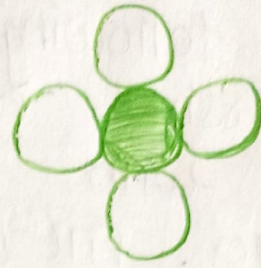
- 4 - Connected Method

- 8 - Connected Method

- Boundary fill algorithm, is particularly useful in interactive painting packages, where interior points are easily selected
- Using a graphics tablet or other interactive device, an designer can sketch a figure outline, select a fill color or pattern from the color menu, and pick an interior point. The system then paints the figure interior

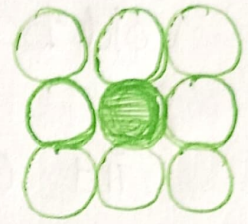
• 4 - Connection Method

→ In this approach, four neighbouring points, left, right, above, below are tested.



• 8 - Connection Method

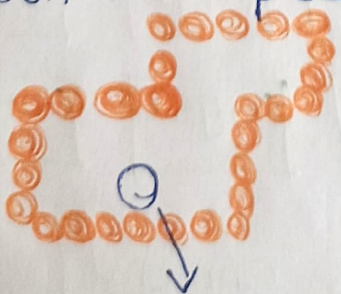
→ In this approach, eight neighbouring points, left, right, above, below, four diagonals are tested.



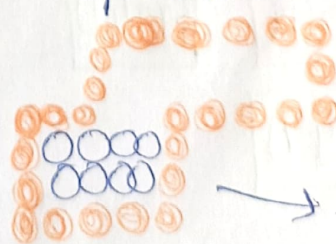
→ 4-Connected method is more suitable than 8-connected method. Boundary can be checked by seeing pixels from left and right first. Then pixels are checked by seeing pixels from top to bottom.

→ The algorithm take time and memory because some recursive calls are needed.

→ but 8-connected boundary fill algorithm would correctly fill the interior of area, not 4-connected boundary fill algorithm produce the partial fill shown.



Start position.



→ partially filled only & 8-connected filled this figure.

→ Algorithm

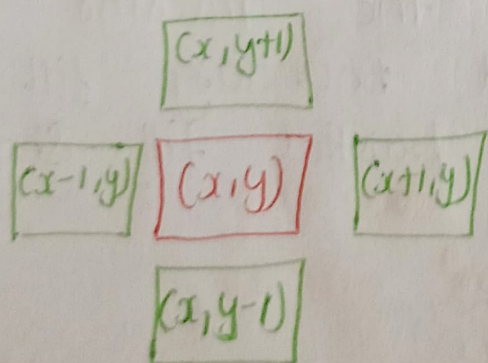
[get pixel function] → give the color of specified pixel
 [put pixel function] → draw the pixel with specified color

- The following procedure illustrate the recursive method for filling a 4-connected area with an intensity specified in parameter. fill [f-color] up to a boundary colour specified with parameter boundary

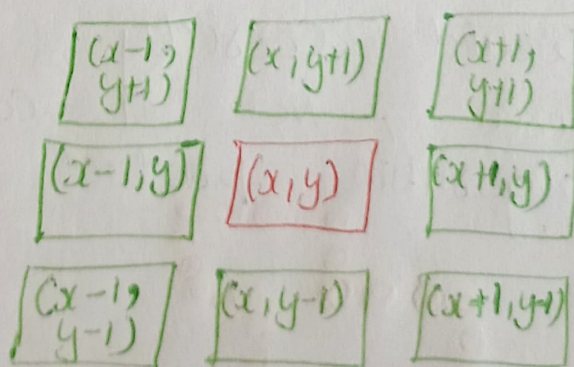
```
void boundaryfill4 (int x, int y, int fill, int boundary)
{
```

```
    int current;
    current = getpixel (x, y);
    if ((current != boundary) && (current != fill)) {
        setcolor (fill); setpixel (x, y);
        boundaryfill4 (x+1, y, fill, boundary);
        boundaryfill4 (x-1, y, fill, boundary);
        boundaryfill4 (x, y+1, fill, boundary);
        boundaryfill4 (x, y-1, fill, boundary);
    }
}
```

- Same procedure can be modified according to 8 connected region algorithm by including four additional statements to test diagonal positions such as $(x+1, y+1)$.



[4 - connected]



[8 - connected]

Problem with recursive boundary fill alg.

- Recursive boundary - fill algorithm may not fill regions correctly if some interior pixels are already displayed in the fill colour. This occurs because the algo. checks next pixels both for boundary color and for fill color.
- Encountering a pixel with the fill color can cause a recursive process to terminate
- Sol: we first change the color of any interior pixels that are initially set to the fill color before applying the boundary - fill algorithm.

⇒ Flood - fill Algorithm = (Seed-fill)

- Flood - fill algorithm is more suitable to fill an area that is defined by several different color boundary.

- Sometimes it is required to fill an area that is not defined within a single colour boundary.



- In such cases we can fill areas by replacing a specified interior colour instead of searching for a boundary colour. This approach is called a flood - fill algorithm.

- Like boundary fill algorithm, here we start with some seed and examine the neighbouring pixels. However, here pixels are checked for a specified interior colour instead of boundary colour and they are fitted replaced by new colour.
- Using either a 4-connected or 8-connected approach, we can step through pixel positions until all interior points have been filled.

→ Algorithm

Procedure: flood-fill (x, y , Old-Colour, new-Colour)

{

if (get pixel (x, y) = Old-Colour)

{

set pixel (x, y , new-Colour);

floodfill ($x+1, y$, Old-Colour, new-Colour);

floodfill ($x-1, y$, Old-Colour, new-Colour);

floodfill ($x, y+1$, Old-Colour, new-Colour);

floodfill ($x, y-1$, Old-Colour, new-Colour);

floodfill ($x+1, y+1$, Old-Colour, new-Colour);

floodfill ($x-1, y-1$, Old-Colour, new-Colour);

floodfill ($x+1, y-1$, Old-Colour, new-Colour);

floodfill ($x-1, y+1$, Old-Colour, new-Colour);

}

}

'get pixel' function give the colour of specified pixel.

• "putpixel" function draws the pixel with specified colour

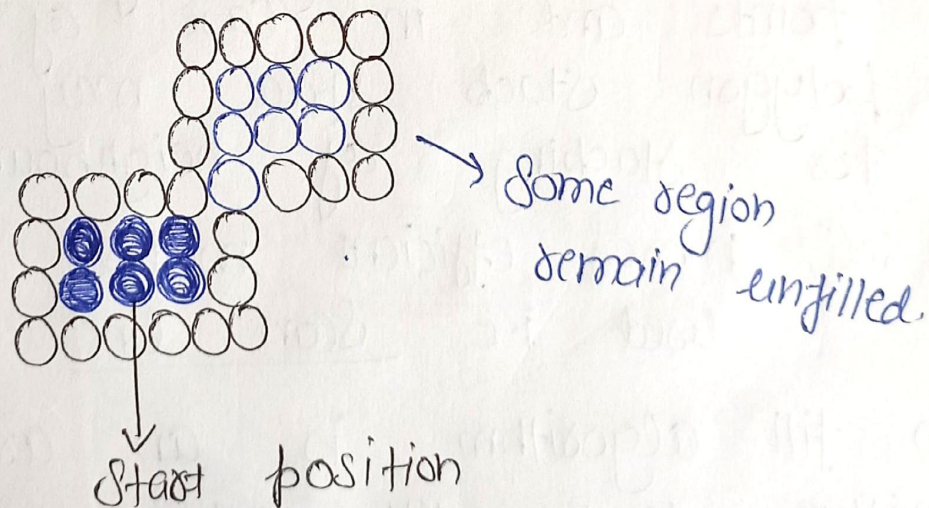
⇒ Situation the flood fill alg do not work properly.

- 1) Regions with holes
- 2) Anti-aliasing or Transparency
- 3) Gradient or smooth color transitions
- 4) Boundary intersections

⇒ Disadvantage

- Very slow algorithm
- May be fail for large polygons
- Initial pixel require more knowledge about surrounding pixels.

Eg: 4 Connected



- 4 - connected Method to fails the fill figure completely. This figure will be efficiently filled by using 8 - connected technique.

⇒ Scan Line Algorithm

- Recursive algorithm for seed fill methods have two difficulties inside

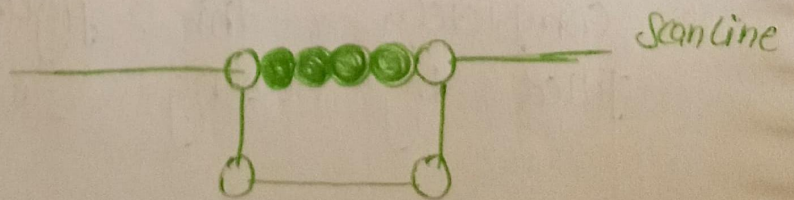
→ first difficulties: if some ↓ pixels are already displayed in fill colour then recursive branch terminates, leaving further internal pixels unfilled.

Solⁿ: We have to first change the colour of any internal pixels that are initially set to the fill colour before applying the seed fill procedures.

→ 2nd difficulties: It cannot be used for large polygons because recursive seed fill procedures require stacking of neighbouring points and in case of large polygon stack space may be insufficient for stacking of neighbouring point.

Solⁿ: more efficient method can be used i.e. Scan line algo.

⇒ Scan fill algorithm is an area-filling algorithm that fill colours by scanning horizontal lines.



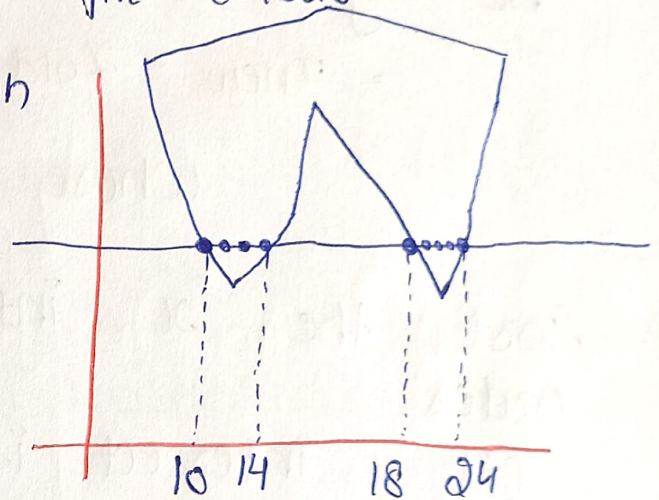
- These horizontal line intersect the boundaries of the polygon and fill colors b/w the intersection points.
- Main purpose: fill colors in the interior pixels of the polygon
- These intersection points are then sorted from left to right, and the corresponding positions b/w each intersection pairs are set to the specified fill colour.

2 It show four intersection position such as

$$x=10 \text{ to } x=14$$

$$\text{and from } x=18 \text{ to } x=24$$

figure show the scan-line procedure for solid filling of polygon area.



Scan line conversion algorithm for polygon filling

- 1) Read n , the number of vertices of polygon.
- 2) Read x and y coordinates for all vertices in array $x[n]$ and $y[n]$.
- 3) Find y_{min} and y_{max} .
- 4) Store info. about edges in structure edge list $[x_1, y_1, x_2, y_2, flag]$.
- 5) For each edge check $y_1 > y_2$ if not interchange (x_1, y_1) and (x_2, y_2) .
- 6) Set $y = y_{max}$.
- 7) Find the active edges and update active edge list.

if (x

if $((y > y_2) \&\& (y \leq y_1))$

edge is active edge

else

edge is not active

8. Compute the x intersects for all active edges for current y value and x intersects for successive y values can be given as

$$x_{\text{new}} = x_{\text{old}} - \frac{1}{m}$$

$$\text{where } m = \frac{y_2 - y_1}{x_2 - x_1}$$

9. Sort the x intersection in ascending order.

If x intersect is vertex i.e., x intersect = x_1 and $y = y_1$ then apply vertex test to check wheather to consider one intersect or two intersects.)

10) Now Extract pair of intersects from the sorted x intersection.

11) Pass pair of x value to line drawing routine to draw corresponding line segments.

12) Set $y = y - 1$

13) Repeat steps 7 to 12 until $y \geq y_{\text{min}}$.

14. Stop.

⇒ used for filling polygons

⇒ it is non-recursive algorithm / efficient

2-D Transformation

- When the transformation takes place on a 2D plane, it is called 2D Transformation.
- 2D means two Dimensional (x-axis & y-axis)

⇒ Transformation

- 'Transform' means 'to change'.
- This change can either be of shape, size or position of the object.
- To perform transformation on any object, object Matrix ' X ' is multiplied by the transformation Matrix T .

$$\begin{bmatrix} \text{Transformed object} \\ \text{matrix} \end{bmatrix} = \begin{bmatrix} \text{Original Matrix} \end{bmatrix} + \begin{bmatrix} \text{Transformation} \\ \text{Matrix} \end{bmatrix}$$

or

$$[X'] = [X][T]$$

- It includes changing the size of the object, its position on the screen, or its orientation.

• Why we used Transformation

- Modeling
- Viewing
- There are two points of view are used for describing the object movement.

1) Geometric Transformation

2) Coordinate Transformation

Representation of Points or objects in Matrix form.

- In 2-D co-ordinate system, any point is represented in terms of x and y co-ordinates.
- The point (x, y) can be converted into matrix in the following two ways:

a) Row-major Matrix

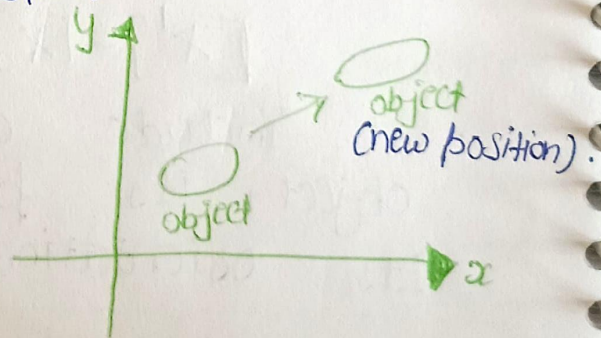
$$\begin{bmatrix} x & y \end{bmatrix}_{1 \times 2}$$

b) Column-major Matrix

$$\begin{bmatrix} x \\ y \end{bmatrix}_{2 \times 1}$$

Geometric Transformation

- The object itself is moved relative to a stationary co-ordinate system.
- The basic geometric transformations are translation, rotation and scaling.
- Other transformations that are often applied to object include reflection and shear.



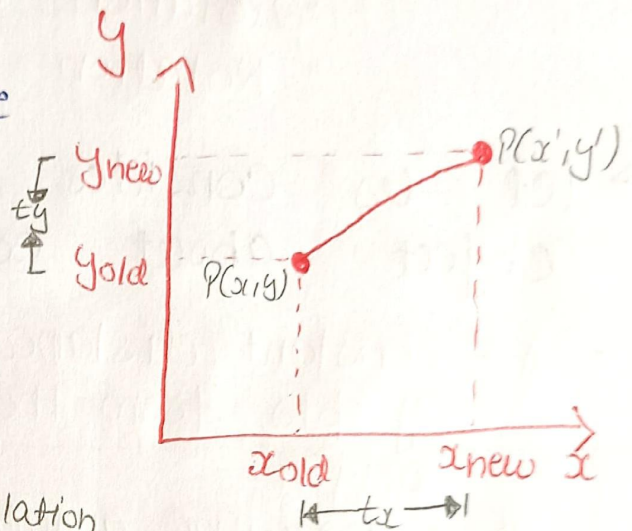
Translation:



→ It defines the changing the position of an object in a straight-line path from one coordinate location to another.

→ We can translate a two dimensional point by adding translation distance t_x and t_y .

→ The original coordinate position (x, y) to Move the point to a new position (x', y')



$$x' = x + t_x$$

$$y' = y + t_y$$

← Translation eqⁿ.

→ The translation distance pair (t_x, t_y) is called Translation vector or Shift vector.

Matrix form

$$P' = P + T$$
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

→ Translation is a rigid-body transformation that move object without deformation.
i.e every point on the object is translated by the same amount.

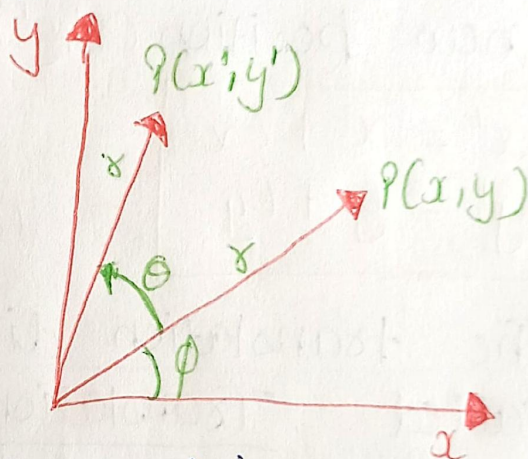
• Rotation = $\curvearrowright$ A two dimensional rotation is applied to an object by repositioning along a circular path in $x-y$ plane.

→ To generate a rotation, we specify two things

- Rotation angle (θ)
- Rotation point [in which the object is to be rotated]

→ let us consider the rotation of the object about origin.

- r = constant distance of pts from the origin
- ϕ = original angular position
- θ = rotation angle



→ Using standard Trigonometric eqn, we can express the transformed co-ordinate in term of θ and ϕ as

$$x' = r \cos(\phi + \theta)$$

$$y' = r \sin(\phi + \theta)$$

→ (1)

→ The original co-ordinates of the point

$$x = r \cos \phi$$

$$y = r \sin \phi$$

→ (2)

→ after substituting eqn (2) & (1), we get

$$\begin{aligned} x' &= x \cos \theta - y \sin \theta \\ y' &= x \sin \theta + y \cos \theta \end{aligned}$$

- Positive value for the rotation angle define Counter Clockwise and Anticlockwise rotation.

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

- Negative value define Clockwise direction.

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

→ Matrix form

$$\begin{aligned} \bullet P' &= P \cdot R \quad \left\{ \begin{array}{l} \text{Rotation Matrix} \end{array} \right. \\ \bullet \begin{pmatrix} x' \\ y' \end{pmatrix} &= \begin{pmatrix} x \\ y \end{pmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \end{aligned}$$

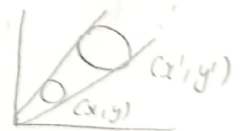
• Scaling :

It is a transformation that change the size or shape of an object.

→ Scaling with respect to origin can be carried out by multiplying the co-ordinate value (x, y) of each vertex of a polygon, by scaling factors S_x and S_y and produce the coordinates (x', y') .

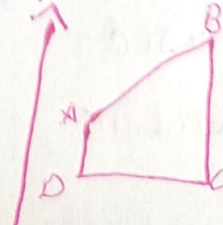
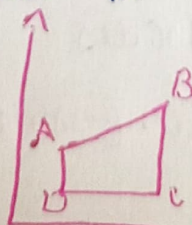
$$\begin{aligned} x' &= S_x \cdot x \\ y' &= S_y \cdot y \end{aligned}$$

← Scaling eqⁿ.



→ S_x scale object in x direction

→ S_y scale object in y direction.

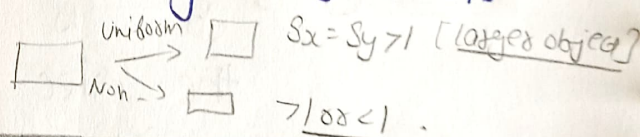


→ Matrix form

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

↓
Translation Matrix

- after applying scaling factor or matrix on any object, coordinate is either increased or decreased depending on the value of scaling factor S_x and S_y
- If scaling factor > 1 , then the object size is increased
- If scaling factor < 1 , then the object size is reduced



Homogeneous Coordinates

- To produce a sequence of transformation, such as translation followed by rotation and scaling.
 - Translate the coordinate
 - Rotate the translated coordinate
 - Scale the rotated coordinate
- But this sequential transformation process is not efficient, To solve this problem, we used homogeneous coordinates.
[combine three transformation into single]

- We use 3×3 transformation Matrix instead of 2×2 transformation Matrix, by introducing an additional dummy coordinate w .
- It allows to express all transformation eqⁿ as Matrix Multiplication.
- 2-D coordinate position (x, y) are represented by triple-coordinate (x, y, w) or $(x, y, 1)$.

→ Homogeneous Coordinate for Translation

$$T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\cdot \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

→ Homogeneous Coordinate for Scaling

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\cdot \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

→ Homogeneous Coordinate for Rotation

$$R = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$\downarrow$ Transformation Matrix $\downarrow$ Object Matrix

- $\downarrow$ Need
- Homogeneous coordinates are generally used in design and construction application. Here, we perform translation, rotation, scaling to fit the picture into proper position.
 - Mainly used in projective geometry, points at infinity are meaningful in this geometry, Homogeneous coordinate provide the representation of point at infinity.
 - Convenience in operation, it make easier to perform geometric operation, because 3x3 Matrix used rather than 2x2 Matrix.
 - Efficient representation of Transformation, simply the implementation of geometric operation and make it easier to compose multiple transformation.

Other Transformation

• Reflection: A reflection is a transformation that produces a mirror image of an object.

• In 2D reflection, we consider any line in 2D plane as the mirror, it is rotated by 180° .

• The size of reflected object is same as the size of original object.

a) Reflection about y-axis

→ A reflection about y-axis ($x=0$) flips x-coordinates while y-coordinate remains same.

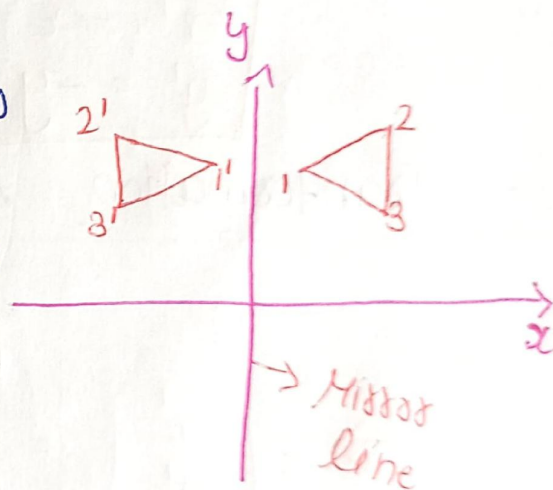
→ $P(x, y)$ to $P'(x', y')$

$$\begin{cases} x' = -x \\ y' = y \end{cases}$$

→ Transformation Matrix

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



$$[P'] = M_y(P)$$

- for homogenous coordinate

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

b) Reflection about x-axis

- for reflection about x-axis

($y=0$), x coordinate is not changed and sign of y -coordinate is changed.

- reflect point

$$\begin{aligned} x' &= x \\ y' &= -y \end{aligned}$$

- Transformation Matrix

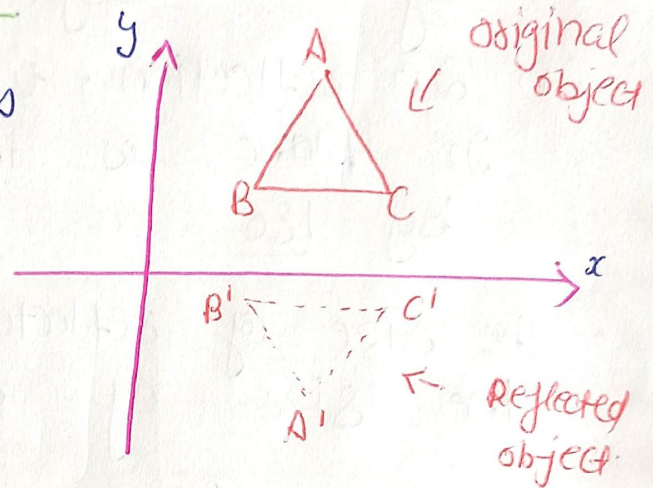
$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$P' = M_x(P)$$

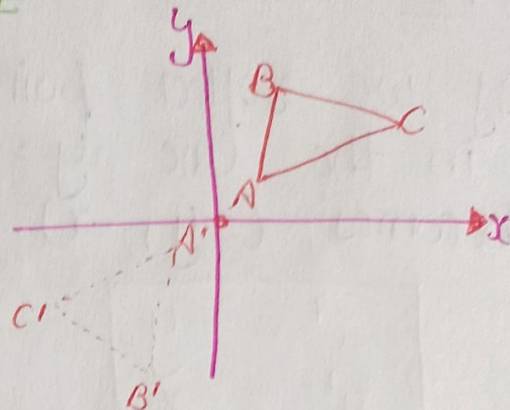
- for homogenous coordinate

$$\rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



c) Reflection about the origin

- In this case, choose the mirror line as the axis $\perp$ to xy plane and passing through the origin.
- after reflection point both x and y coordinate of object change



$$\text{i.e. } \begin{cases} x' = -x \\ y' = -y \end{cases}$$

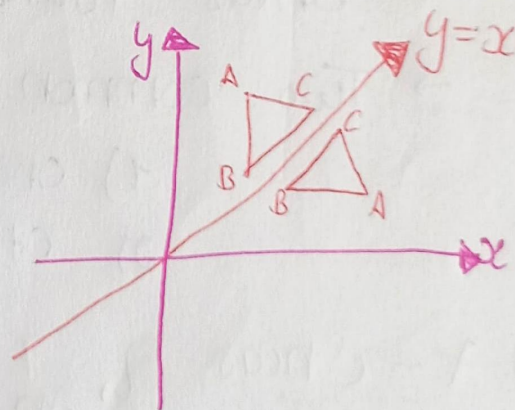
→ Transformation Matrix

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

d) Reflection about $y=x$

- If we reflect (x, y) point about line $y=x$, we get (y, x)

$$\text{i.e. } \begin{cases} x' = y \\ y' = x \end{cases}$$



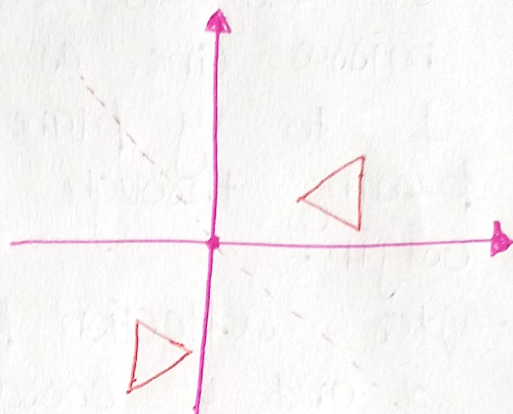
Transformation Matrix

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

c) Reflection about the Straight Line $y = -x$

• If we reflect point $P(x, y)$ in the line $y = -x$, it becomes $(-y, -x)$.

$$\begin{cases} x' = -y \\ y' = -x \end{cases}$$



• Transformation Matrix

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Shear

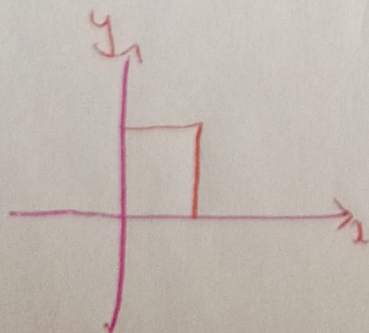
→ A transformation that slants the shape of object is called the Shear transformation.

→ Two common Shearing are used

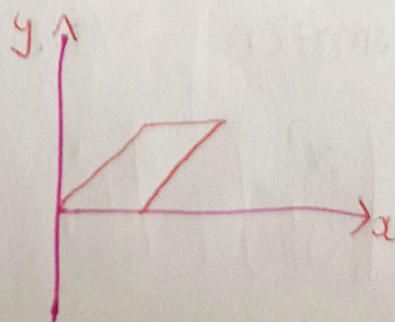
- 1) one shift x -coordinate value
- 2) other shift y -coordinate value.

a) x-Shear

In x -Shear, y -coordinate remain unchanged, but x is changed.



[original object]



[object after x Shear]

→ Transformation Matrix

$$\rightarrow x' = x + \text{Sh}_x y$$

$$y' = y$$

$$\rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \text{Sh}_x & 1 \end{bmatrix}$$

$\hookrightarrow x_{\text{Sh}}$

→ Homogenous Matrix

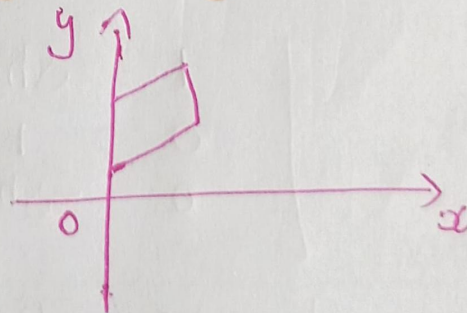
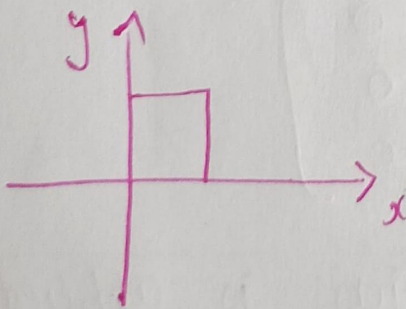
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \\ \quad \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

b) y-Shear : In y-shear, y co-ordinate is changed and x remain unchanged

$$\rightarrow x' = x$$

$$y' = y + (\text{Sh}_y x)$$

$$\rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} \quad \\ \quad \end{bmatrix}$$



→ Homogenous

Copyrighted by Codecap.io
All rights reserved.

C) X-Y directions

- both co-ordinates changes

- $x' = x + (Sh_x y)$

- $y' = y + (Sh_y x)$

→ Matrix = xy_{sh} = $\begin{bmatrix} 1 & Sh_y \\ Sh_x & 1 \end{bmatrix}$

⇒ Shearing Relative to other reference line

- Now apply x shear and y shear transform relative to other reference line.
- In x shear transformation we can use y reference line.

$$\begin{bmatrix} 1 & 0 & 0 \\ Sh_x & 1 & 0 \\ -Sh_x \cdot y_{ref} & 0 & 1 \end{bmatrix}$$

- In y shear transformation, use x reference line.

$$\begin{bmatrix} 1 & Sh_y & 0 \\ 0 & 1 & 0 \\ 0 & -Sh_y \cdot x_{ref} & 1 \end{bmatrix}$$

- When a Shearing Transformation is applied to an object, it distorts the shape of the object by shifting its points along one of the coordinate axes.
- Shearing can occur along x-axis, y-axis.

- Shearing operation can be expressed as a sequence of basic transformation. The sequence of basic transformation involve series of rotation and scaling transformation.

→ Shear Transformation may be expressed in term of rotation & scaling.

Sol: Shear Transformation

$$= \begin{bmatrix} 1 & Sh_y & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Scaling Matrix and rotation Matrix

$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S \cdot R = \begin{bmatrix} S_x \cos \theta & S_x \sin \theta & 0 \\ -S_y \sin \theta & S_y \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Compare Shear Matrix and rotation Matrix/scaling

$$Sh_x = -S_y \sin \theta$$

$$Sh_y = S_x \sin \theta$$

$$\bullet S_x \cos \theta = 1 \quad \text{and} \quad S_y \cos \theta = 1$$

$$\bullet S_x = \frac{1}{\cos \theta} \quad \bullet S_y = \frac{1}{\cos \theta}$$

- Substituting value of S_x and S_y .

$$\bullet Sh_x = \frac{1}{\cos \theta} \cdot \sin \theta = -\tan \theta$$

$$\therefore \text{Shy} = \frac{1}{\cos \theta} \cdot \sin \theta = \tan \theta$$

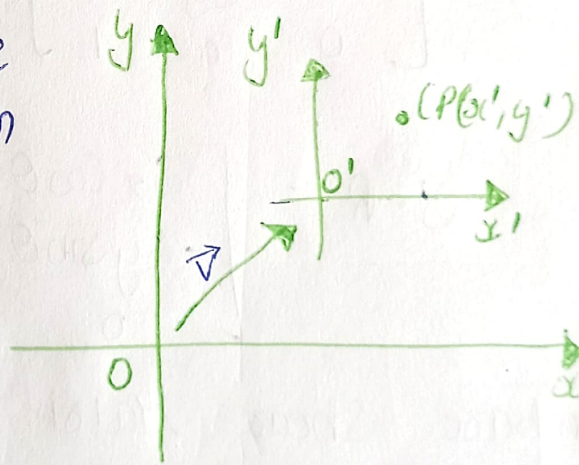
$\therefore$, the Shear transformation Matrix expressed in term of rotation and Scales is

$$\begin{bmatrix} 1 & \tan \theta & 0 \\ -\tan \theta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Co-ordinate Transformation

The object is held stationary while the coordinate system is moved relative to the object. This effect is attained through the application of Coordinate Transformation.

Assume that we have two co-ordinate system in the plane.



→ The first system is located at origin O & has co-ordinate axis xy .

→ The second coordinate system is located at origin O' & has coordinate axis $x'y'$.

Now each point in the plane has two coordinate description.

- i) (x, y) with respect to xy coordinate system.
- ii) (x', y') with respect to $x'y'$ coordinate system.

• If we think the second system $x'y'$ arises from a transformation applied to first system xy which is called coordinate transformation.

→ Translation

• If the xy coordinate system is displaced to new position, where the direction & displacement or distance is given by the vector.

$$\vec{r} = t_x \vec{i} + t_y \vec{j}$$

• The coordinate of a point in both system are related by the translation transformation

$$(x', y') = T_r(x, y)$$

$$\text{i.e. } \begin{aligned} x' &= x - t_x \\ y' &= y - t_y \end{aligned}$$

• Matrix form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

• 3x3 Notation

$$T_r = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Copyrighted by
All rights reserved.
CodeCap.io

→ Rotation

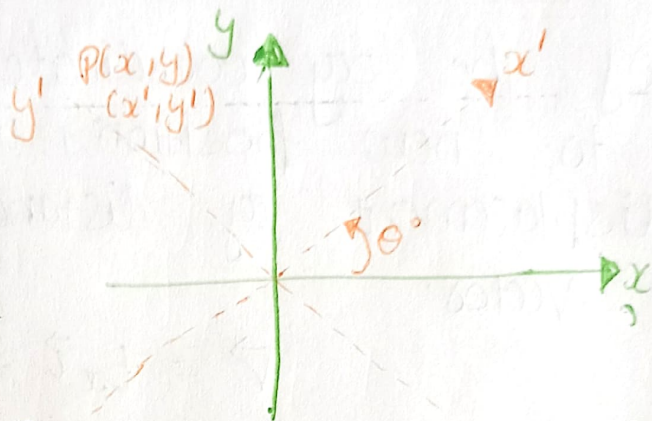
The xy system is rotated θ about the origin. Then the coordinates of a point on both system are related by rotation transformation $\vec{R}_\theta$

$$\rightarrow (x', y') = \vec{R}_\theta (x, y)$$

i.e

$$x' = x \cos \theta + y \sin \theta$$

$$y' = -x \sin \theta + y \cos \theta$$



Matrix

$$= \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$$

→ Scaling with origin

Scaling is performed by making new coordinate system while the origin and coordinate axis remain unchanged

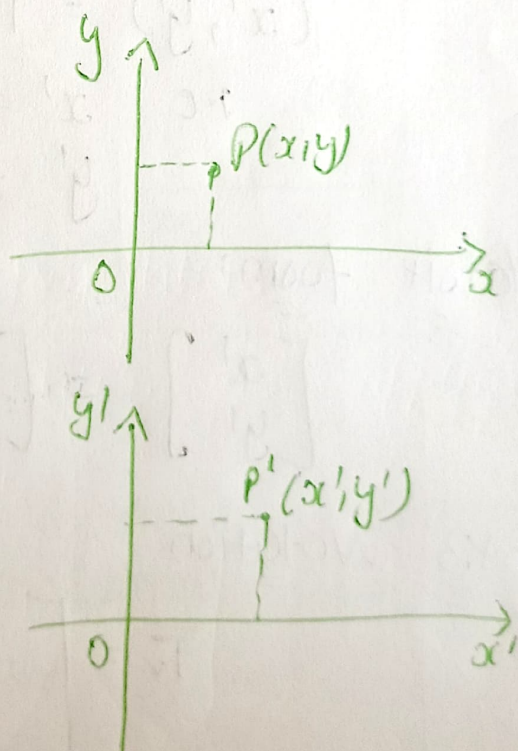
• but introducing different unit of measurement along the x axis and y - axis

• The new coordinate are obtained from the old units by scaling

S_x and S_y

↓
along x axis

↓
along y axis.



$$x' = \frac{1}{s_x} \cdot x$$

$$, y' = \frac{1}{s_y} \cdot y$$

• Matrix form

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{1}{s_x} & 0 \\ 0 & \frac{1}{s_y} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

→ Reflection

• If the new coordinate system is obtained by reflecting the old system either x or y axis, the relationship b/w coordinate is given by coordinate transformation along x-axis, y-axis.

• along x-axis

$$\rightarrow \begin{aligned} x' &= x \\ y' &= -y \end{aligned}$$

$$M_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

• along y-axis

$$\rightarrow \begin{aligned} x' &= -x \\ y' &= y \end{aligned}$$

$$\rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

↓
 M_y

Composite Transformation

- A Sequence of transformation can be combined into single one is known as Composite or concatenation Transformation.
- More complex geometric and coordinate transformation can be built from the basic transformation by using the process of Composite of function.

Ex: Suppose we want to perform rotation about an arbitrary point, then we can perform it by the sequence of three transformation

- 1) Translation
- 2) Rotation
- 3) Reverse Translation

→ Composition of two translations

- If two successive translation are applied then

$$\begin{matrix} T_1 & T_2 \\ \begin{bmatrix} 1 & 0 & tx_1 \\ 0 & 1 & ty_1 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 0 & tx_2 \\ 0 & 1 & ty_2 \\ 0 & 0 & 1 \end{bmatrix} \end{matrix} = \begin{bmatrix} 1 & 0 & tx_1 + tx_2 \\ 0 & 1 & ty_1 + ty_2 \\ 0 & 0 & 1 \end{bmatrix}$$

→ Composition of two Rotations

• If two successive rotation θ_1 & θ_2 then

$$\begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R\theta_1, R\theta_2$$

→ Composition of two Scaling

$$\begin{bmatrix} S_{x1} \cdot S_{x2} & 0 & 0 \\ 0 & S_{y1} \cdot S_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad [S_1 \cdot S_2]$$

Inverse Transformation

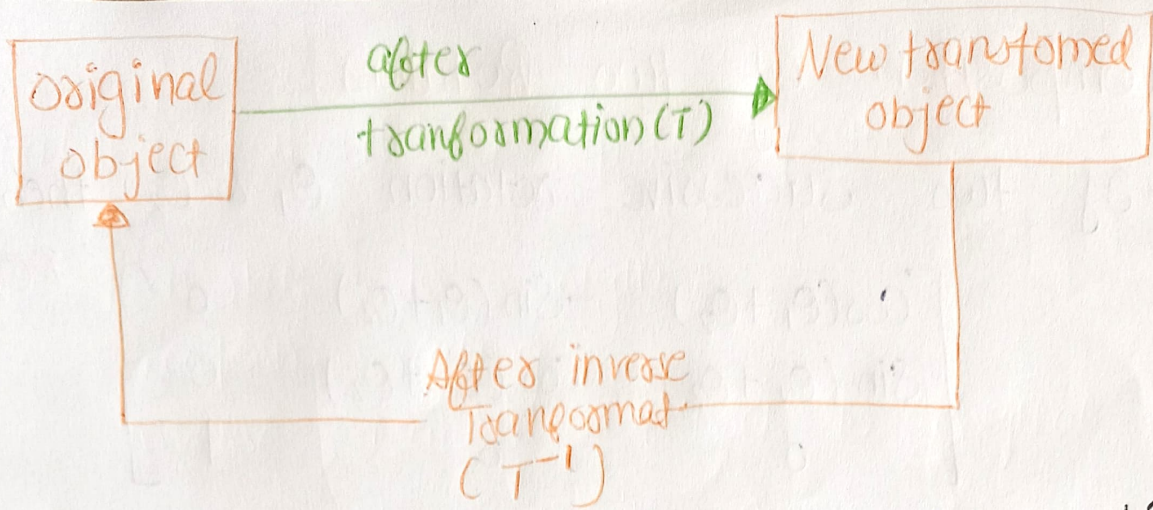
- Inverse Transformation is used to reverse the effects of a Transformation applied to an object.
- also known as Opposite Transformation.
- If T is a translation Matrix then inverse translation is representing using T^{-1} .

$$\boxed{T = T^{-1}} \quad \text{or} \quad \boxed{TT^{-1} = I}$$

- It may be a geometric or a coordinate transformation but both have an inverse

→ each geometric transformation has an inverse which is described by opposite operation performed by transformation.

$$\boxed{T_v^{-1} = T_v} \quad , \quad \boxed{R_\theta^{-1} = R_\theta} \quad , \quad \boxed{S_{x,y}^{-1} = S_{\frac{1}{x}, \frac{1}{y}}}$$



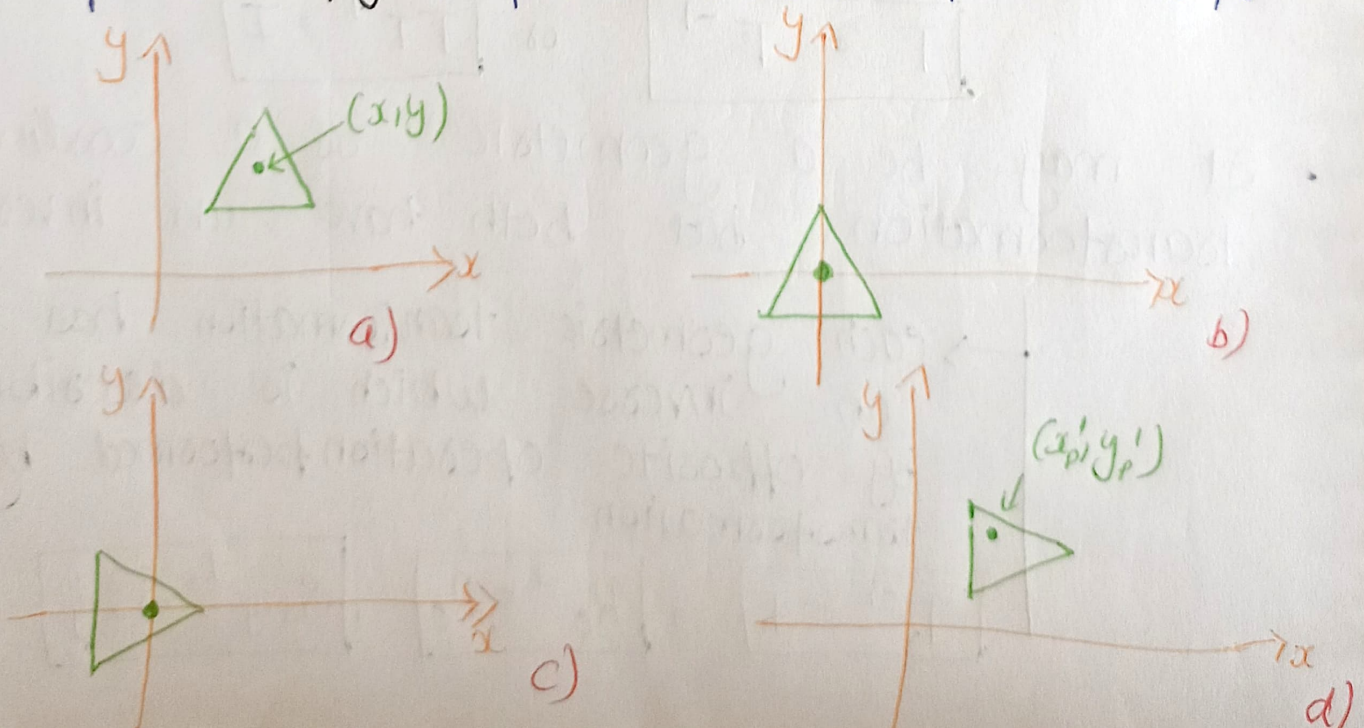
- The inverse transformation matrix $[T^{-1}]$ is equal to the transpose $[T]^T$ of the transformation matrix $[T]$.

$$[T]^T = [T]^{-1}$$

=> Rotation about an Arbitrary point

- 1) Translate the object at origin
- 2) Rotate by any angle about the origin.
- 3) Translate back to its original position (inverse translation).

→ To rotate an object about an arbitrary point (x_1, y_1) , follow above three steps.



In Matrix form:

$$[T] = [T_R][R_\theta][T_R]^{-1}$$

where

T_R = Translation Matrix

R_θ = Rotation matrix by an angle θ

$[T_R]^{-1}$ = Inverse translation matrix

• Translation: Move (x_p, y_p) to origin

$$\bullet t_x = -x_p$$

$$\bullet t_y = -y_p$$

$$T_R = \begin{bmatrix} 1 & 0 & -x_p \\ 0 & 1 & -y_p \\ 0 & 0 & 1 \end{bmatrix}$$

• Rotation:

$$R_\theta = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

• Translation: Move the image back to original position

$$\bullet t_x = x_p$$

$$\bullet t_y = y_p$$

$$[T_R]^{-1} = \begin{bmatrix} 1 & 0 & x_p \\ 0 & 1 & y_p \\ 0 & 0 & 1 \end{bmatrix}$$

→ Combined

$$T_R \cdot R_\theta \cdot T_R^{-1}$$

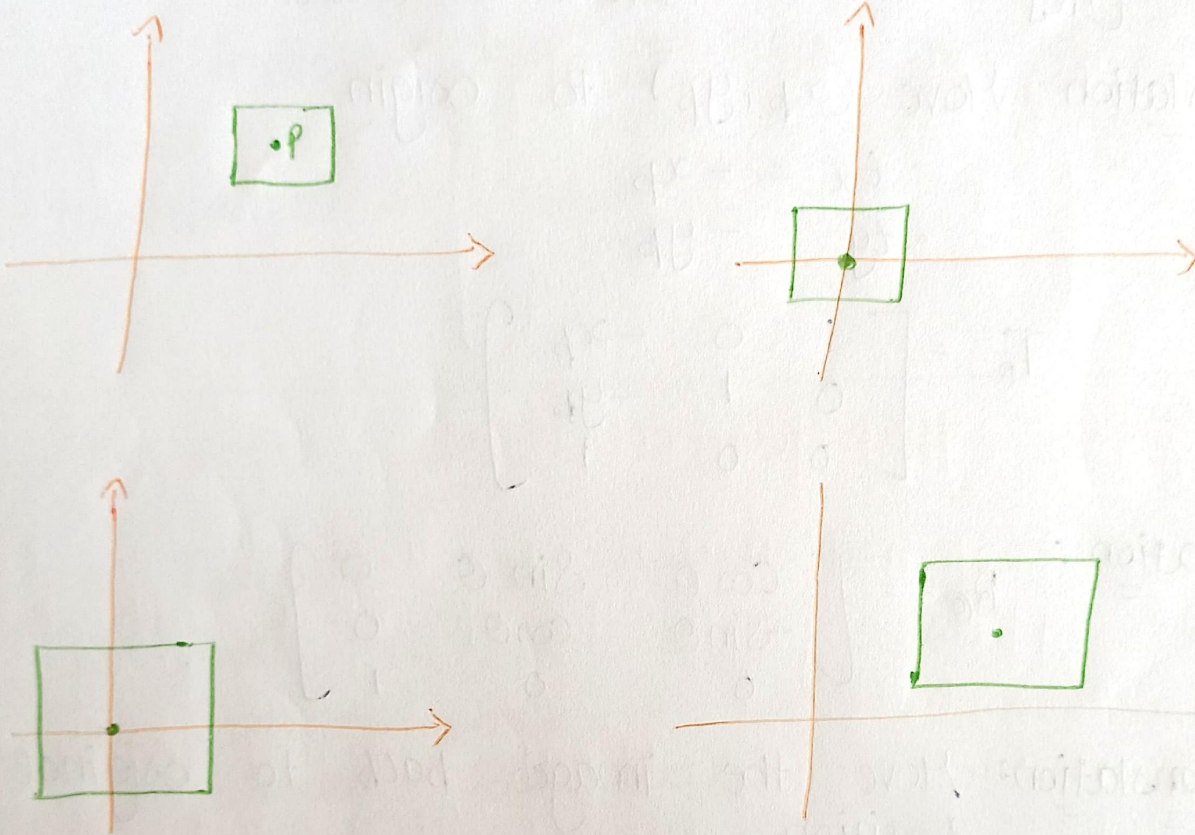
$$= \begin{bmatrix} \cos \theta & \sin \theta & -x_p \cos \theta + y_p \sin \theta + x_p \\ -\sin \theta & \cos \theta & -x_p \sin \theta - y_p \cos \theta + y_p \\ 0 & 0 & 1 \end{bmatrix}$$

→ Rotation

→ Scaling about a fixed point

Copyrighted by CodeCap.io
All rights reserved.

- Translate to origin
- Perform Scaling
- Inverse Translation (back to original position)



$$\rightarrow [T] [S_{xy}] [T]^{-1}$$

$$= \begin{bmatrix} S_x & 0 & x_p(1-S_x) \\ 0 & S_y & y_p(1-S_y) \\ 0 & 0 & 1 \end{bmatrix}$$

→ Reflection through an Arbitrary Line

$$[T] = [T_R] [R_O] [R_{ref}] [R_O]^{-1} [T_R]^{-1}$$

about
axis