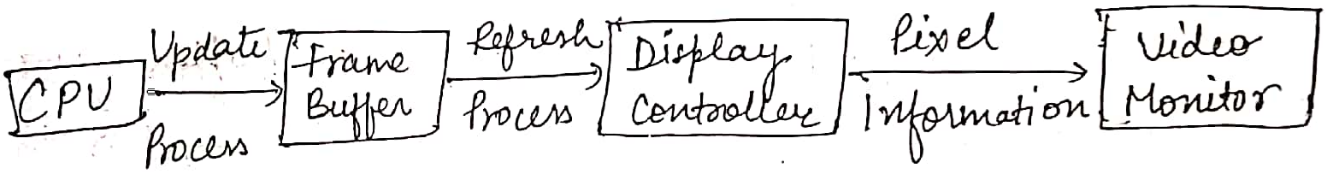
COMPUTER GRAPHICS

Computer Graphics is responsible for displaying art and image data effectively and meaningfully. Computer Graphics development has had a significant impact on many types of media and has revolutionized animation, movies, advertising, video games and Graphic Design.

In Computer Graphics, pictures or Graphics objects are presented as a collection of discrete picture elements called pixels (picture element). The pixel is the smallest addressable screen element. It is the smallest piece of the display screen which we can control. The control is achieved by setting the intensity and colour of the pixel which compose the screen.

✯ COMPONENTS OF COMPUTER GRAPHICS



Interactive computer Graphics consists of 3 components such as

→ Digital Memory Buffer (Frame Buffer)

→ TV Monitor

→ Display Controller

→ Digital Memory Buffer. (Frame Buffer) Sem-3

This is place where image / pictures are stored as an array (matrix of 0 and 1).

Frame Buffer is the Video RAM (V-RAM) that is used to hold / map the image displayed on the screen.

The amount of memory required to hold the image depends primarily on the resolution of the screen image and also the colour depth used per pixel.

Frame Buffer is implemented using rotating random access semiconductor memory. However frame buffer also can be implemented using Shift Register.

→ TV Monitor

Monitor helps us to view the display and they make use of CRT (Cathode Ray Tube).

→ Display Controller.

It is an interface b/w digital memory buffer and TV Monitor. The main function of this is to pass the contents of frame Buffer to the Monitor.

☆ APPLICATION OF COMPUTER GRAPHICS

Computer Graphics is used in a variety of fields including Business, industry, Engineering + Medicine govt. work, education & training, advertising, research art and entertainment.

Major application areas are :-

→ Computer Aided Design / Drafting (CAD and CADD)
→ Presentation Graphics
→ Entertainment
→ Computer Aided Learning (CAL)
→ Computer Art
→ Graphical User Interface (GUI)
→ Medical applications
→ Geographical information systems (GIS)

(Elaborate Yourself)

DISPLAY DEVICES

The most important part in a PC is the display system. The display systems where the graphics are rendered in the console screen of the computer. It is responsible for graphic display.

Some of the common types of display systems are:-

a) Raster scan display
b) Random scan display
c) Direct view storage tube
d) Flat panel displays

The display systems are often referred to as Video Monitor or Video Display unit. The most common video monitor that normally comes with a PC is the Raster Scan type. Every display system has three parts :-

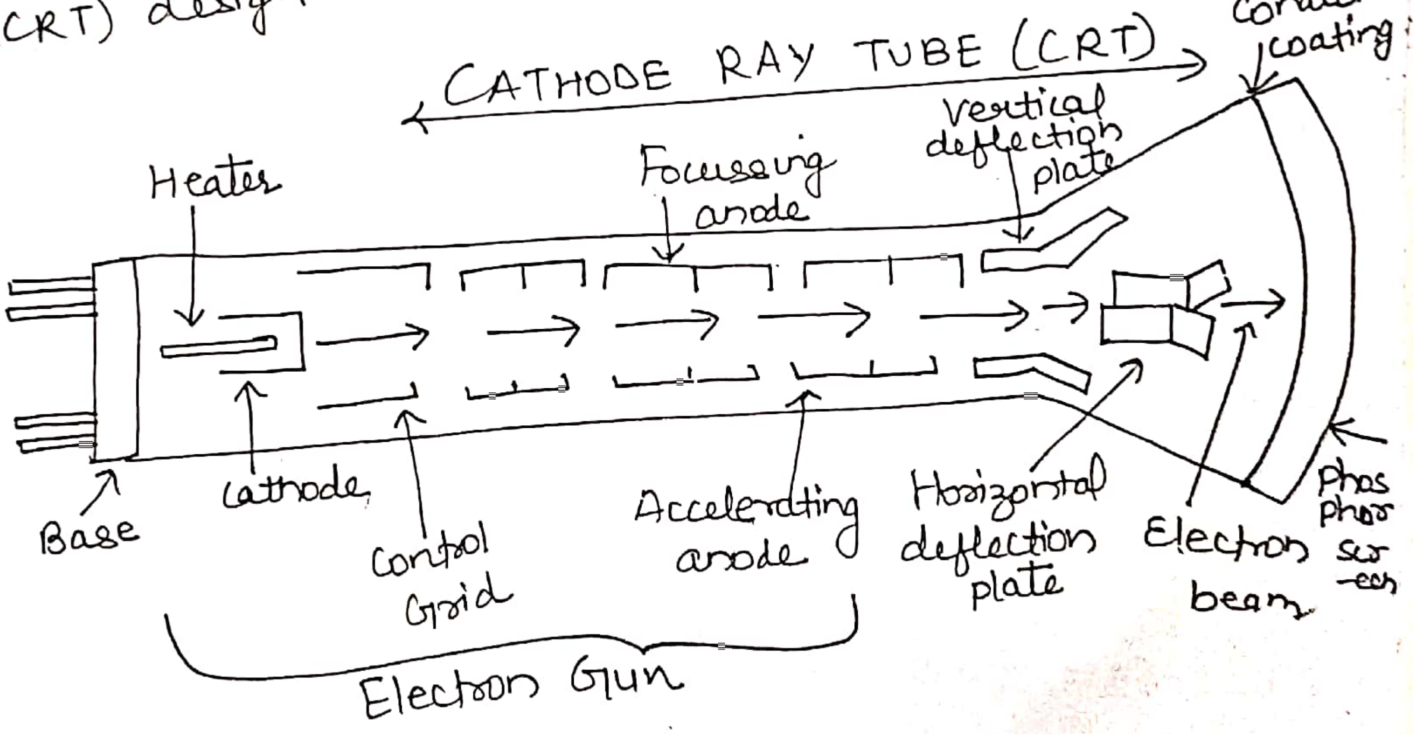
4) i) Display adapter :- that creates & hold the image information.

ii) Monitor :- which display that information.

iii) Cable :- that carries the image data b/w display adapter & the monitor.

The display devices are also known as output devices.

The most commonly used output device in a graphic system is video monitor and the operation of most video monitors is based on the Cathode-Ray Tube (CRT) design.



Cathode ray tube consists of a gas-filled glasstube in which two metal plates one negatively charged and the other positively charged, have been place.

Cathode ray tubes are special, electronic vaccum tubes that use focused electron beams to display images.

Scanned by CamScanner

when a very large voltage is placed across the electrodes, the neutral gas inside the tube will ionize into conducting plasma, and a current will flow as electrons travel from the cathode to the other side.

The flow of the electrons is natural, not forced.

When used inside a television set, a CRT's electrons are concentrated in a light beam by positively charged terminal, called an anode. An accelerating anode is then use to speed up the movement of the electrons. These fast-moving electrons fly through the tube's vacuum hitting the phosphor-coated screen and making it glow.

Cathode-ray tubes are found in oscilloscopes, and similar devices are used in TV picture tubes and computer displays.

Working :- The cathode in CRT is raised to a high temperature by the heater, and electrons evaporate from the surface of the cathode. The accelerating anode, with a small hole at its centre, is maintained at a high potential V_1, of the order of 1 to 20 kV, relative to the cathode. Electrons passing through the hole in the anode form an arrow beam and travel with constant horizontal velocity from the anode to the fluorescent screen. The area where the electrons strike the screen glows brightly.

⑥ <u>Control Grid</u> :- It regulates the number of electrons that reach the anode and hence the brightness of the ~~grid regulates the number of electrons~~ spot on the screen.

<u>Focussing anode</u> :- It ensures that electrons leaving the cathode in slightly different directions are focused down to a narrow beam and all arrive at the same spot on the screen.

<u>Electron Gun</u> :- The assembly of cathode, control grid, focusing anode and accelerating electrode is called electron gun.

The beam of electrons passes b/w two pairs of deflec-tion plates. An electric field b/w the second pair deflects them horizontally. If no deflecting ~~plates~~ fields are present, the electrons travel in a straight line from the hole in the accelerating anode to the center of the screen, where they produce a bright spot.

THE BEAM-PENETRATION CRT

The normal CRT can generate images of only single color due to limitations of its phosphor. A color CRT devices uses a multilayer phosphor and achieves color control by modulating a normally constant parameter, namely the beam accelerating potential. The screen is coated with a layer of green phosph-or over which a layer of red phosphor is deposited. When a low potential electron beam strikes the screen only red phosphor is excited thus producing a red trace. A higher velocity beam will penetrate into the green phosphor increasing the green component of the light output.

By varying the beam potential, different combinations of red & green light can produce a limited range of color such as orange, yellow etc. Beam penetration has been an inexpensive way to produce color in random-scan monitors, but only four colors are possible and quality of pictures is not as good as with other methods.

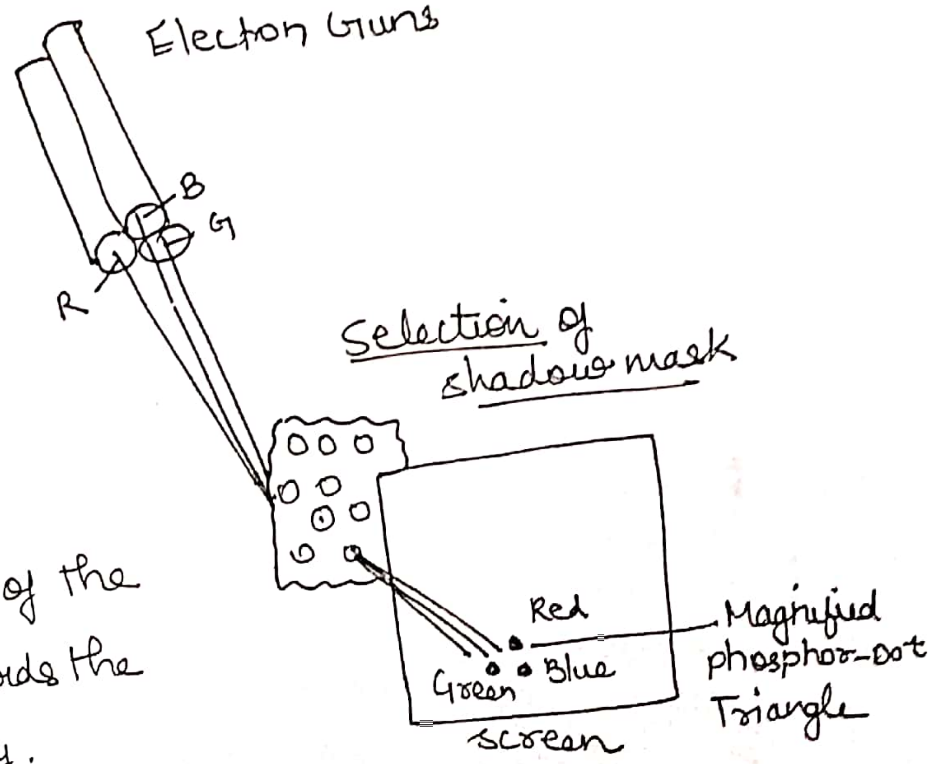
Advantages :- The biggest advantage is that it is at half cost of shadow mask & its resolution is better.

Disadvantage :- Change of color takes time which doesn't suit interactive graphics at all.

SHADOW MASK CRT

Shadow-mask methods are commonly used in raster-scan systems because they produce a much wider range of color than the beam penetration method.

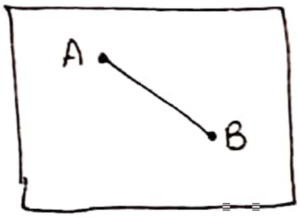
A shadow mask CRT has three phosphor color dots at each pixel position. One phosphor dot emits a red light, another emits a green light, and third emits a blue light. This type of CRT has three electron guns, one for each color dot, and a shadow-mask grid just behind the phosphor coated screen.



Doming :- center of the mask bulges towards the faceplate slightly.

RANDOM SCAN DISPLAY (MONITORS) (Vector scan)

In this technique, the electron beam is directed only to the part of the screen where the picture is to be drawn rather than scanning from left to right and top to bottom. It is also called vector display, stroke-writing display & calligraphic display.



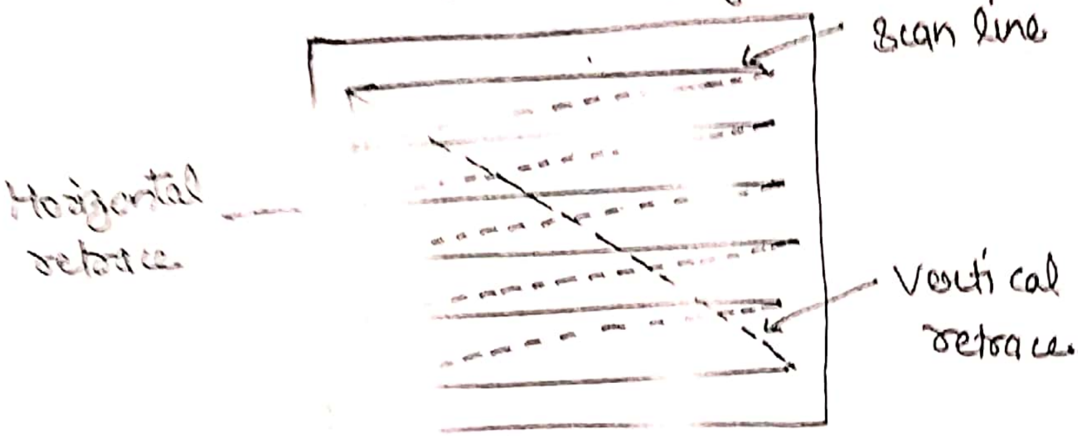
e.g. we want a line connecting point A to B on the vector graphics display, we simply drive the beam deflection circuitry, which will cause beam to go directly from point A to B.

RASTOR SCAN DISPLAY

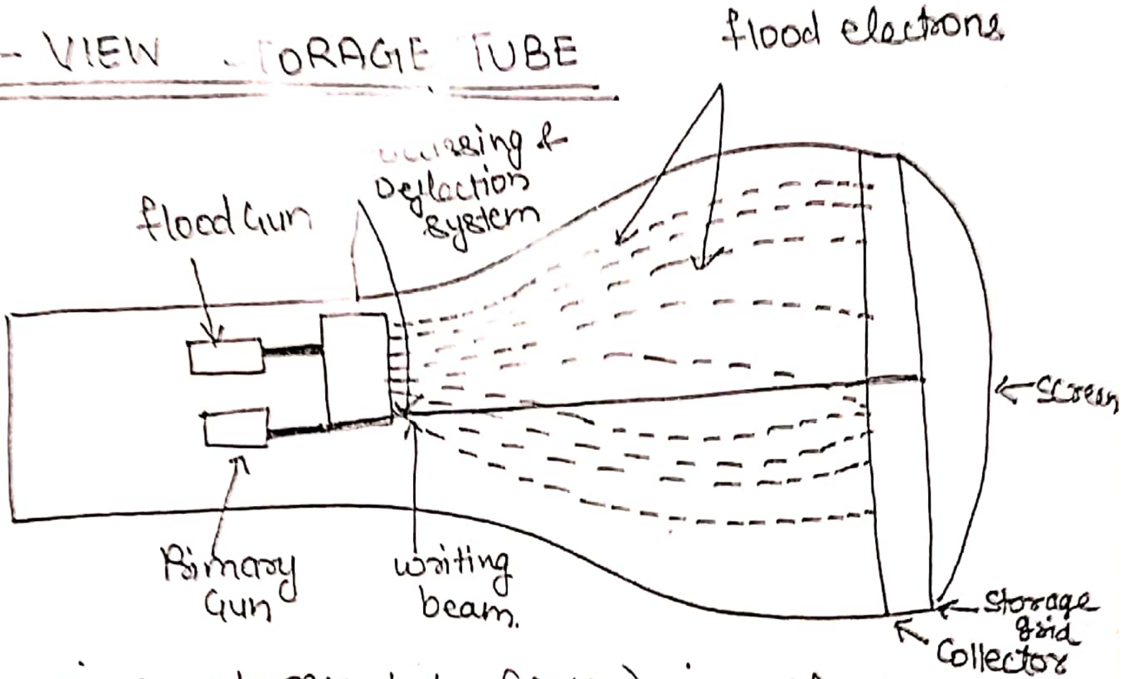
In a raster scan system, the electron beam is swept across the screen, one row at a time from top to bottom. As the electron beam moves across each row, the beam intensity is turned on and off to create a pattern of illuminated spots.

Picture definition is stored in memory area called the Refresh Buffer or Frame Buffer. This memory area holds the set of intensity values for all the screen points. Stored intensity values are then retrieved from the refresh buffer and "painted" on the screen one row at a time.

⑩ Each screen point is referred to as a pixel. At the end of each scan-line, the electron beam returns to the left side of the screen to begin displaying the next scan line.



DIRECT-VIEW STORAGE TUBE



A direct view storage tube (DVST) gives the alternative method of maintaining the screen image. A DVST uses the storage grid which stores the picture information as a charge distribution just behind the phosphor coated screen.

It consists of two electron guns:-

a) Primary Gun → stores the picture pattern

b) Flood gun → maintains the picture display.

In DVST, there is no refresh buffer, the images are created by drawing vectors or line segments with a relatively slow- moving electron beam.

The DVST contains also a second grid just behind the storage mesh, that is called collector. The main purpose of collector is to smooth out the flow of flood electrons.

Advantage

1) Refreshing of CRT is not required.
2) Because no refreshing is required, very complex pictures can be displayed at very high resolution without flicker.
3) It has flat screen.

Disadvantage

1) They do not display colors and are available with single level of line intensity.
2) Erasing requires removal of charge on the storage grid. Thus erasing and redrawing process takes several seconds.
3) Selective or part erasing of screen is not possible.
4) The performance of DVST is some what inferior to the refresh CRT.

FLAT-PANEL DISPLAY

A number display methods are in use that is designed to reduce the depth of the CRT display caused by the length of the tube. These devices are collectively known as <u>flat panel displays</u>.

These types of flat panel displays commonly in use with computer systems are Liquid Crystal Displays (LCD's), Gas Plasma Displays (GPD's) and Electroluminescent displays (ELD's)

The screen of these flat panel displays are made up of pair of electrodes. Each pair of electrodes is used to generate one picture element.

Two types of flat-panel displays
- Emissive displays
- Non-emissive displays

<u>Emissive</u> The emissive displays (emitter) are devices that convert electrical energy into light. e.g. Plasma panels, light emitting diodes

<u>Non-emissive</u> :- Non-emissive displays (non-emitters) use optical effects to convert sunlight or light from some other source into graphics pattern. e.g. LCD (Liquid crystal Device).

INTERACTIVE DEVICES (Input devices)

1) <u>Keyboard</u> → primary input device for entering text & numbers. It is a relatively simple device which consists of about 100 keys, each of which sends different codes to the CPU.
The number of keyboard varies from the original standard of 101 keys to the 104 keys windows keyboards.

2) <u>MOUSE</u> → A mouse functions as a pointing devices by detecting two-dimensional motion relative to its supporting surface. It can be used only with GUI based OS e.g. Windows.
A mouse is small hand-held box used to position the screen cursor. Two types of mouse → mechanical mouse, Optical mouse

<u>Mouse offers two main benefits</u> :-
↳ Mouse lets us position the cursor anywhere on the screen quickly without using the cursor movement keys
↳ Instead of forcing us to type or issue commands from the keyboard, The mouse-based operating systems let us choose commands for easy to use menus and dialog box.

3) <u>TRACKBALLS AND SPACEBALLS</u>

A trackball is a pointing device that works like an upside-down mouse. A trackball is a ball that can be rotated with the fingers or palm of the hand to produce scree cursor movement. Trackball requires less space than a mouse.

- A trackball is a two-dimensional positioning device
- Spaceball provides six degree of freedom (3-dimensional Positioning).

(4) TRACKPADS OR TOUCHPADS

Touchpad is a stationary pointing device.

(5) Joystick :- Joystick consists of a vertical lever (called the stick) which canbe swing around moving the cursor on the screen. Joystick are often used to control games.

(6) Touch screen

A touch-screen is a computer display screen that is sensitive to human touch, allowing a user to interact with the computer by touching pictures or words on the screen.

(7) Light Pen

A light pen is pointing device shaped like a pen and is connected to the computer. The tip of the light pen contains a light-sensitive element. Light pen can work with any CRT based monitor, but not with LCD screen, projectors or other display devices.

(8) DATA GLOVE :- It is an interface device that uses position tracking sensors and fibre optic strands running down each finger & connected to compatible computer.

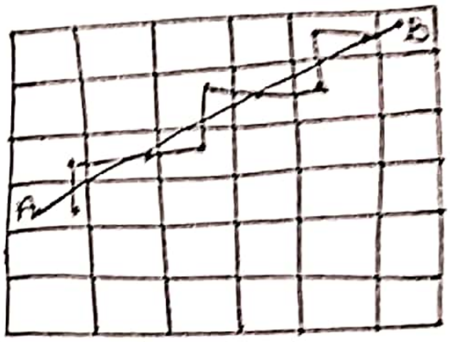
(9) SCANNER :- In computing, a scanner is a device that optically scans images, printed text, handwriting & converts it to a digital image.

(10) Voice system

The voice system or speech recognition system is a sophisticated input device that accepts voice or speech input from the user & transform it to digital data.

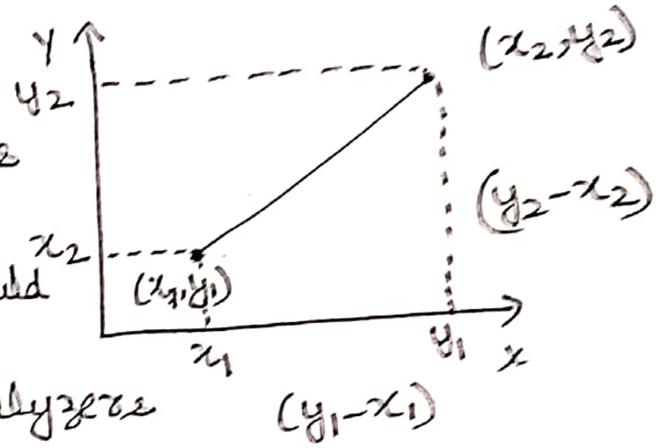
DDA (DIGITAL DIFFERENTIAL ANALYZER) ALGORITHM

★ **Stair-case effect**



The effect of having two or more pixels ON to approximating a line b/w two points say A & B is known as the Staircase effect.

For DDA :-



The vector generation algorithms which step along the line to determine the pixels which should be turned ON are sometimes called digital differential analyzers

We know that the slope of a straight line is given as:-

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

(x_1, y_1) ; (x_2, y_2) are the end points of a straight line.

We assume equation of line is $\boxed{y = mx + c}$

At $x = x_i$ we have $y_i = mx_i + c$

Similarly at $x = x_{i+1}$ we have $y_{i+1} = mx_{i+1} + c$

Now, case 1 when value of $\boxed{m \leq 1}$ (16)

Increment x by 1 unit i.e $\boxed{x_{i+1} = x_i + 1}$ — (1)

and by using eq^n of line $y = mx + c$, we have

$$y_{i+1} = m(x_i + 1) + c \qquad \boxed{y_i = mx_i + c}$$
$$= mx_i + m + c$$
$$\boxed{y_{i+1} = y_i + m} \quad \text{— (2)}$$

To plot the pixel :-

~~plot~~ put pixel (x, round(y))

Eq^n (1) & (2) imply that to approximate line for case 1 we have to move along x direction by 1 unit to have next value of x and we have to add slope m to initial y value to get next value of y.

Similarly, Case 2 :- $m > 1$

Increment y by 1 unit i.e. $\boxed{y_{i+1} = y_i + 1}$ — (3)

for value of x_{i+1}, we have

$$y_i = mx_i + c \Rightarrow c = y_i - mx_i$$

$$y_{i+1} = mx_{i+1} + c \Rightarrow y_{i+1} = mx_{i+1} + y_i - mx_i$$

$$y_{i+1} - y_i = m(x_{i+1} - x_i)$$

from (3)

$$1 = m(x_{i+1} - x_i) \Rightarrow \boxed{\frac{1}{m} + x_i = x_{i+1}} \quad \text{— (4)}$$

Eq^n (3) & (4) imply that to approximate line for case 2 we have to move along y direction by 1 unit to have next value of y and we have to add slope $1/m$ to initial x value to get next value of x.

E.g. Draw line segment from point (2,4) to (9,9) using DDA. (17)

Sol Points (2,4) & (9,9)
(x_1, y_1) (x_2, y_2)

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{9-4}{9-2} = \frac{5}{7} = \boxed{0.7 < 1}$$

$m < 1$

next point to be plotted

$x_{i+1} = x_i + 1 = 3$

$y_{i+1} = y_i + m = 4 + 0.7 = 4.7 = 5.$

point is [x, round(y)] i.e [3,5]

→ (4, round(5.4)) = (4,5)
→ (5, round(6.1)) = (5,6)
→ (6, round(6.8)) = (6,7)
→ (7, round(7.5)) = (7,8)
→ (8, round(8.2)) = (8,8)
→ (9, round(8.9)) = (9,9)

Points to be plotted b/w (2,4) → (9,9) are
(2,4) (3,5) (4,5) (5,6) (6,7) (7,8) (8,9) (9,9)

Q2 Plot (2,3) & (12,8) using DDA.

Advantages of DDA

1) DDA algorithm is faster than the direct use of the line eq^n since it calculates points on the line without any floating point multiplication.

2) It is a simplest algorithm and does not require special skills for implementation.

Disadvantage

1) A floating point addition is still needed in determining each successive point which is time consuming.

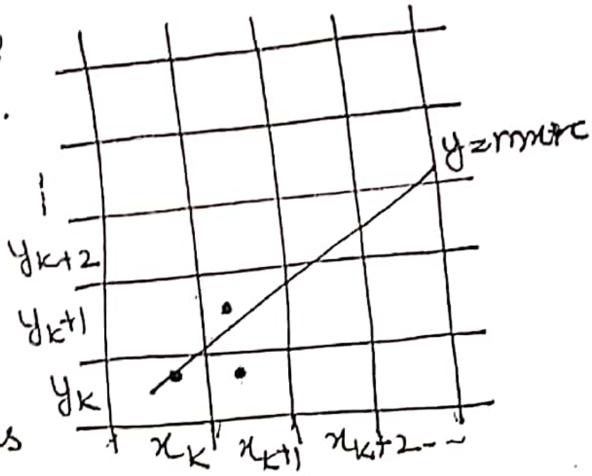
2) It is orientation dependent, due to this end point accuracy is poor.

BRESENHAM LINE GENERATION ALGORITHM

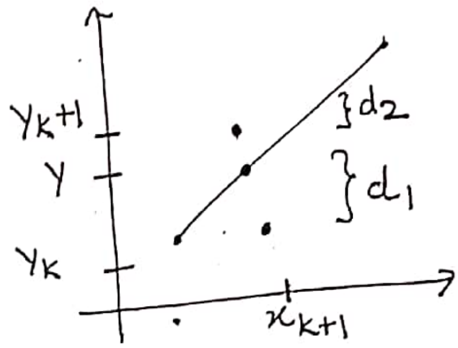
Bresenham algorithm is accurate and efficient raster line generation algorithm. This algorithm scans converts lines using only incremental integer calculation.

This algorithm uses only integer addition & subtraction and multiplication by 2.

We consider the scan-conversion process for lines with positive slope less than 1.



Assuming we have determined that the pixel at (x_k, y_k) is to be displayed, we now need to decide which pixel to plot in column x_{k+1}. Our choices are the pixels at positions (x_k+1, y_k) & (x_k+1, y_k+1).



At sampling position x_k+1, we label the vertical pixel separation from the mathematical line path as d_1 & d_2. The Y coordinate on the mathematical line at pixel column position x_k+1 is calculated as:-

$$y = m(x_k+1)+c$$

Now
$$d_1 = y - y_k \Rightarrow m(x_k+1)+c-y_k$$
$$d_2 = (y_k+1)-y \Rightarrow (y_k+1) - m(x_k+1)-c$$

Difference b/w these two separations is :-

$$d_1 - d_2 = [m(x_k+1)+c-y_k] - [(y_k+1) - m(x_k+1)-c]$$
$$= mx_k + m + c - y_k - y_k - 1 + mx_k + m + c$$
$$d_1 - d_2 = 2mx_k + 2m - 2y_k + 2c - 1$$

$$d_1 - d_2 = 2m(x_k+1) - 2y_k + 2c - 1$$

we know that $m = \frac{\Delta y}{\Delta x}$

$$d_1 - d_2 = 2\frac{\Delta y}{\Delta x}(x_k+1) - 2y_k + 2c - 1$$

$$\underbrace{(d_1 - d_2)\Delta x}_{P_K} = 2\Delta y(x_k+1) - 2y_k\Delta x + 2c\Delta x - \Delta x$$

P_K (decision parameter) $= \cancel{2\Delta y \Delta} = 2\Delta y x_K - 2\Delta x y_K + \underbrace{2\Delta y + \Delta x(2c-1)}_{\text{constant value}}$

$P_K = 2\Delta y x_k - 2\Delta x y_k + d$ ← independent of pixel position ①

Sign of P_K is same as the sign of $(d_1 - d_2)$

⇒ If pixel at y_K is closer to the line path than the pixel at y_k+1 that means $d_1 < d_2$ i.e decision parameters P_K is −ve∴ In that case we plot the ~~upper pixel~~ lower pixel, otherwise, we plot the upper pixel.

At step K+1, P_K is evaluated as:-

$$P_{K+1} = 2\Delta y\, x_{k+1} - 2\Delta x\, y_{k+1} + d \quad \text{—②}$$

By subtracting ① & ②

$$P_{k+1} - P_K = 2\Delta y\, x_{k+1} - 2\Delta x\, y_{k+1} + \cancel{d} - 2\Delta y\, x_k + 2\Delta x\, y_k - \cancel{d}$$

$$= 2\Delta y\, \cancel{x_k} + 2\Delta y - 2\Delta x\, y_{k+1} - 2\Delta y\, \cancel{x_k} + 2\Delta x\, y_k$$

$$P_{k+1} - P_K = 2\Delta y - 2\Delta x(y_{k+1} - y_k)$$

$$\boxed{P_{K+1} = P_K + 2\Delta y - 2\Delta x(y_{k+1} - y_k)}$$

$y_{k+1} - y_K$ is either 0 or 1, depending upon the sign of parameter P_K.

Starting pixel position (x_0, y_0) $(c=0)$ $m = \Delta y/\Delta x$, first parameter $\boxed{P_0 = 2\Delta y - \Delta x}$ by putting in eq ①.

Now, if $P_k < 0$, next point to plot is (x_{k+1}, y_k)

and
$$P_{k+1} = P_k + 2\Delta y$$

Else next point to plot is (x_k+1, y_k+1)
$$P_{k+1} = P_k + 2(\Delta y - \Delta x)$$

Q. Draw line segment joining (20,10) & (25,14) using bresenham line generation algorithm.

Sol (20,10) (25,14)

$\Delta y = 14 - 10 = 4$

$\Delta x = 25 - 20 = 5$

$P_0 = 2\Delta y - \Delta x = 2 \times 4 - 5 = 8 - 5 = 3$

$P_0 > 0$

point to be plot is (21, 11) ✓

$P_1 = 3 + 2(4-5) = 3 - 2 = 1 > 0$

point to be plot is (22,12) ✓

$P_2 = 1 + 2(4-5) = 1 - 2 < 0$

point (23,12) ✓

$P_3 = -1 + 2 \times 4 = -1 + 8 = 7 > 0$

(24,13) ✓

$P_4 = 7 + 2(4-5) = 7 - 2 = 5 > 0$

$\boxed{\text{point } (25,14)}$ ✓

Q. Consider the line from (5,5) to (13,9). Use the Bresenham algorithm to rasterize the line.

MID-POINT CIRCLE ALGORITHM

circle is defined as a set of points that are all at a given distance r from a centre positioned at (x_c, y_c)

This is represented mathematically by the eqn:-

$$(x-x_c)^2+(y-y_c)^2 = r^2 \quad \text{---} (1)$$

using eqn(1) we can calculate the value of y for each given value of x as $y = y_c \pm \sqrt{r^2-(x-x_c)^2}$ ---- (2)

Thus one could calculate different pairs by giving step increments to x and calculating the corresponding value of y. But this approach involves considerable computation at each step and also the resulting circle has its pixels sparsely plotted for areas with higher values of the slope of the curve.

Midpoint circle Algorithm uses an alternative approach wherein the pixel positions along the circle are determined on the basis of incremental calculations of a decision parameter.

Let

$$f(x,y) = (x-x_c)^2+(y-y_c)^2-r^2 \quad \text{---} (3)$$

Thus $f(x,y)=0$ represents the eqn of circle.

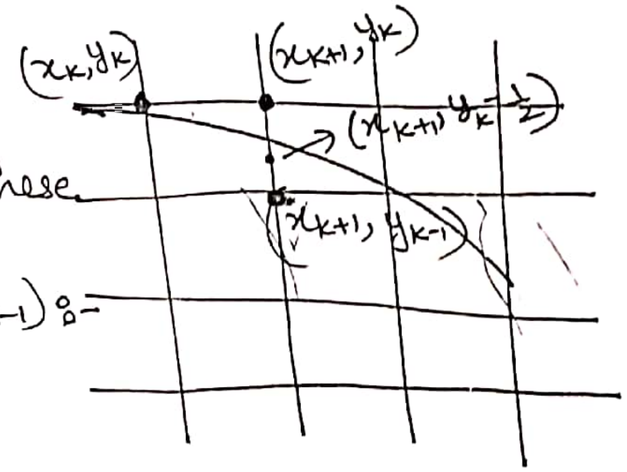
Further, we know from coordinate geometry, that for any point, the following holds:

1. $f(x,y)=0 \Rightarrow$ The point lies on the circle.
2. $f(x,y)<0 \Rightarrow$ The point lies within the circle.
3. $f(x,y)>0 \Rightarrow$ The point lies outside the circle.

In Midpoint circle Algorithm, the decision parameter at the k^{th} step is the circle function evaluated using the coordinates of the mid point of the two pixel centres which are the next possible pixel positions to be plotted.

Let us assume that we are giving unit increments to x in the plotting process and determining the y position using this algorithm. Assuming we have just plotted the k^{th} pixel at (x_k, y_k), we next need to determine whether the pixel at the position (x_{k+1}, y_k) or the one at (x_{k+1}, y_{k-1}) is closer to the circle.

Decision parameter P_k at the k^{th} step is the circle function evaluated at the midpoint of these two pixels.



Midpoint of (x_{k+1}, y_k) & (x_{k+1}, y_{k-1}) :-

$$= \left(\frac{x_k+1+x_k+1}{2}, \frac{y_k+y_k-1}{2}\right)$$

$$= \left(x_k+1, y_k-\frac{1}{2}\right)$$

Thus P_k,

$$P_k = f\left(x_k+1, y_k-\frac{1}{2}\right) = (x_k+1)^2 + \left(y_k-\frac{1}{2}\right)^2 - r^2 \quad \text{—(4)}$$

Successive decision parameters are obtained by using incremental calculations, thus avoiding a lot of computation at each step. We obtain a recursive expression for the next decision parameter i.e at the $(k+1)^{th}$ step, we have

$$P_{k+1} = f\left(x_{k+1}+1, y_{k+1}-\frac{1}{2}\right) = (\underbrace{x_{k+1}}_{x_k+1}+1)^2 + \left(y_{k+1}-\frac{1}{2}\right)^2 - r^2$$

$$P_{k+1} = (x_k+1+1)^2 + (y_{k+1}-\tfrac{1}{2})^2 - r^2 = (x_k+2)^2 + (y_{k+1}-\tfrac{1}{2})^2 - r^2 \quad \text{— (5)}$$

(5) − (4) gives:-

$$P_{k+1} - P_k = (x_k+2)^2 + (y_{k+1}-\tfrac{1}{2})^2 - \cancel{r^2} - (x_k+1)^2 - (y_k-\tfrac{1}{2})^2 + \cancel{r^2}$$

$$= \cancel{x_k^2 + 4 + 4x_k + 4}$$

$$= (x_k+2)^2 - (x_k+1)^2 + (y_{k+1}-\tfrac{1}{2})^2 - (y_k-\tfrac{1}{2})^2$$

$$= (\cancel{x_k}+2-\cancel{x_k}-1)(x_k+2+x_k+1) + (y_{k+1}-\cancel{\tfrac{1}{2}}-y_k+\cancel{\tfrac{1}{2}})(y_{k+1}-\tfrac{1}{2}+y_k-\tfrac{1}{2})$$

$$\boxed{P_{k+1} = P_k + (2x_k+3) + (y_{k+1}+y_k-1)(y_{k+1}-y_k)} \quad \text{— (6)}$$

If $P_k < 0$, then the midpoint of the two possible pixels lies within the circle, and selected pixel is (x_{k+1}, y_k).

By substituting this value in eqn (6) $(y_{k+1} = y_k)$

$$P_{k+1} = P_k + (2x_k+3) + (y_k+y_k-1)\underbrace{(y_k-y_k)}_{0}$$

$$\boxed{P_{k+1} = P_k + (2x_k+3)}$$

If $P_k \geq 0$, then the midpoint of the two possible pixels lies outside the circle, and selected pixel is (x_{k+1}, y_{k-1})

Now substitute $y_{k+1} = y_k - 1$

$$P_{k+1} = P_k + (2x_k+3) + (y_{k+1}+y_k-1)(y_{k+1}-y_k)$$

$$= P_k + (2x_k+3) + (y_k+y_k-1-1)(\cancel{y_k}-1-\cancel{y_k})$$

$$= P_k + (2x_k+3) + (2y_k-2)(-1) = P_k + (2x_k+3) - 2y_k + 2$$

$$\boxed{P_{k+1} = P_k + 2(x_k - y_k) + 5}$$

For the boundary condition, we have $x = 0$, $y = r$.

$$P_k = (x_k+1)^2 + (y_k-\tfrac{1}{2})^2 - r^2 = (0+1)^2 + (r-\tfrac{1}{2})^2 - r^2$$

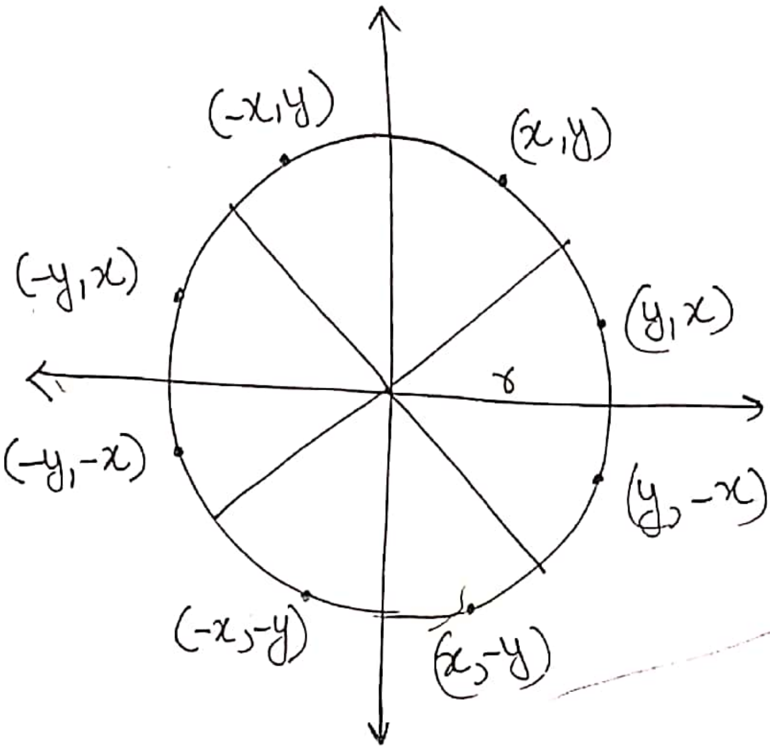
$$P_0 = 1 + \cancel{r^2} + \tfrac{1}{4} - r - \cancel{r^2} = \tfrac{5}{4} - r \text{ or } 1.25 - r$$

$$\boxed{P_0 = 1 - r}$$

If $P_K < 0$: $y_{K+1} = y_K$ and $P_{K+1} = P_K + (2x_K + 3)$

$P_K \geq 0$: $y_{K+1} = y_K - 1$ and $P_{K+1} = P_K + 2(x_K - y_K) + 5$

$P_0 = 1 - r$



Because of 8-way symmetry, By calculating a pixel position (x,y) to be plotted, we get 7 other points on the circle corresponding to it. These are:-

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

Q. Using midpoint circle algorithm plot a circle whose radius = 10 units.

Sol

$r = 10$

$(x, y) = (0, 10)$. First point

$P_0 = 1 - r = 1 - 10 = -9 \quad P_0 < 0$

$\therefore$ Point to be plotted is (x_{k+1}, y_k) $(1, 10)$ 2nd point

$P_1 = P_0 + (2x_k + 3) = -9 + 2 + 3 = -4 < 0$

3rd point $(2, 10)$

$P_2 = P_1 + (2x_k + 3) = -4 + 4 + 3 = 3 > 0$

4th point $(3, 9)$

$P_3 = P_2 + 2(x_k - y_k) + 5 = 3 + 2(3 - 9) + 5$

$= 3 + 2(-6) + 5 = 8 - 12 = -4 < 0$

5th point $(4, 9)$

$P_4 = P_3 + 2x_k + 3 = -4 + 8 + 3 = 7 > 0$

6th point $(5, 8)$

$P_5 = P_4 + 2(x_k - y_k) + 5 = 7 + 2(5 - 8) + 5 = 12 - 6 = 6 > 0$

7th point $(6, 7)$

$P_6 = P_5 + 2(x_k - y_k) + 5 = 6 + 2(6 - 7) + 5 = 11 - 2 = 9 > 0$

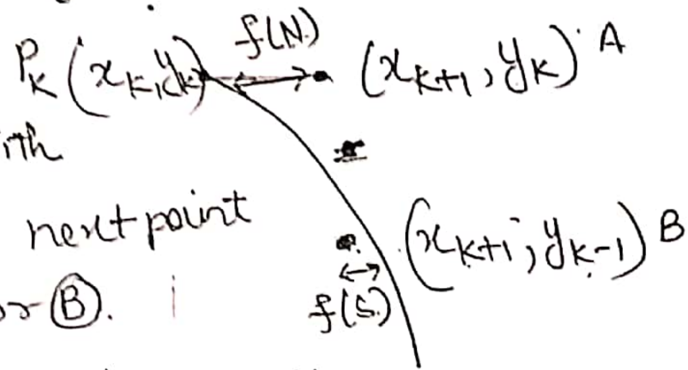
8th point $(7, 6)$ $\boxed{\text{stop as } x > y \text{ or } x = y}$

Scanned by CamScanner

BRESENHAM CIRCLE ALGORITHM

Bresenham's method of drawing the circle is an efficient method because it avoids trigonometric and square root calculation by adopting only integer operation involving squares of the pixel separation distances

This algorithm is called incremental because the position of the next pixel is calculated on the basis of last plotted one, instead of calculating the pixels from a formula.



If P_K is a current point with coordinates (x_K, y_K), then the next point ~~called~~ could be either (A) or (B).

To choose the pixel, we have to measure which point is close to the circle.

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

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

$$f(N) = (x_K + 1)^2 + (y_K)^2 - r^2$$

$$f(S) = (x_K + 1)^2 + (y_K - 1)^2 - r^2$$

$$d_i = \underset{(+ve)}{f(N)} + \underset{(-ve)}{f(S)}$$

decision parameter

Now if, $d_i > 0$
selected pixel is (x_{K+1}, y_{K-1})

else

(x_{K+1}, y_K)

As $d_i = f(N) + f(S)$

$= (x_k+1)^2 + y_k^2 - r^2 + (x_k+1)^2 + (y_k-1)^2 - r^2$

$d_i = 2(x_k+1)^2 + y_k^2 + (y_k-1)^2 - 2r^2$ —— ①

$d_{i+1} = 2(x_k+2)^2 + y_{k+1}^2 + (y_{k+1}-1)^2 - 2r^2$ —— ②

② - ①

$d_{i+1} - d_i = 2(x_k+2)^2 + y_{k+1}^2 + (y_{k+1}-1)^2 - \cancel{2r^2} - 2(x_k+1)^2 - y_k^2 - (y_k-1)^2 + \cancel{2r^2}$

$d_{i+1} - d_i = 2[(x_k+2+x_k+1)(\cancel{x_k}+2-\cancel{x_k}-1)] + ((y_{k+1}+y_k)(y_{k+1}-y_k)) + ((y_{k+1}-1+y_k-1)(y_{k+1}-1-y_k-1))$

$$\boxed{d_{i+1} = d_i + 2(2x_k+3) + ((y_{k+1}+y_k)(y_{k+1}-y_k)) + ((y_{k+1}-1+y_{k}-1)(y_{k+1}-1-y_{k-1}))}$$

$\boxed{\text{If } d_i < 0}$

$x_{k+1} = x_k + 1$; $y_{k+1} = y_k$

$d_{i+1} = d_i + 2(2x_k+3) + (y_k+y_k)\underbrace{(\cancel{y_k}-\cancel{y_k})}_{0} + (y_k-1+y_k-1)\underbrace{(\cancel{y_k}\cancel{-1}-\cancel{y_k}\cancel{+1})}_{0}$

$d_{i+1} = d_i + 2(2x_k+3) + 0 = d_i + 4x_k + 6$

$$\boxed{d_{i+1} = d_i + 4x_k + 6}$$

<u>Now if $d_i \geq 0$</u> $\quad x_{k+1} = x_k + 1$

$\quad y_{k+1} = y_k - 1$

$d_{i+1} = d_i + 2(2x_k+3) + (y_k-1+y_k)(\cancel{y_k}-1-\cancel{y_k}) + (y_k-1\cancel{-1}+y_k\cancel{-1})(\cancel{y_k}-1-1-\cancel{y_k}\cancel{+1}) = 0.$

$= d_i + 2(2x_k+3) + (2y_k-1)(-1) + (2y_k-3)(-1)$

$= d_i + 2(2x_k+3) + (1-2y_k) + (3-2y_k)$

$= d_i + 4x_k + 6 + 1 - 2y_k + 3 - 2y_k$

$= d_i + 4x_k + 10 - 2y_k - 2y_k = d_i + 4x_k - 4y_k + 10$

$$\boxed{d_{i+1} = d_i + 4(x_k - y_k) + 10}$$

Initial value of d_i can be obtained by replacing $x=0$ and $y=r$, in 1

$$d_0 = 2(x_k+1)^2 + y_k^2 + (y_k-1)^2 - 2r^2$$

$$= 2(0+1)^2 + r^2 + (r-1)^2 - 2r^2$$

$$= 2 + r^2 + (r^2+1-2r) - 2r^2 = \cancel{2+r^2+r^2+1-2r-2r}$$

$$= 2 + \cancel{r^2} + \cancel{r^2} + 1 - 2r - \cancel{2r^2} = 3 - 2r$$

$$\boxed{d_0 = 3 - 2r}$$ $3 - 2r$ $= -3$

initial parameter

If $d_i < 0$ (x_k+1, y_k)

$d_{i+1} = d_i + 4x_k + 6$

If $d_i \geq 0$ (x_k+1, y_k-1)

$d_{i+1} = d_i + 4(x_k - y_k) + 10$

Q. Plot a circle by bresenham's algorithm whose radius is 3 & centre is (0,0)

(0,3) (1,3)

2-D Transformation

Transformation means changing the size of the object, its position on the screen or its orientation. The implementation of such a change is called transformation.

To perform transformation on any object, object matrix 'x' is multiplied by the transformation matrix T.

$$\begin{bmatrix}\text{Transformed object} \\ \text{matrix}\end{bmatrix} = \begin{bmatrix}\text{Transformation} \\ \text{matrix}\end{bmatrix} \times \begin{bmatrix}\text{Original object} \\ \text{matrix}\end{bmatrix}.$$

The fundamental objective of 2-D transformation is to simulate the movement and manipulation of objects in the plane. The ability to transform these points and lines is achieved by :- translation, rotation, scaling and reflection

Transformation is described in two categories :-

Geometric transformation

object itself is moved relative to a stationary co-ordinate system or background.

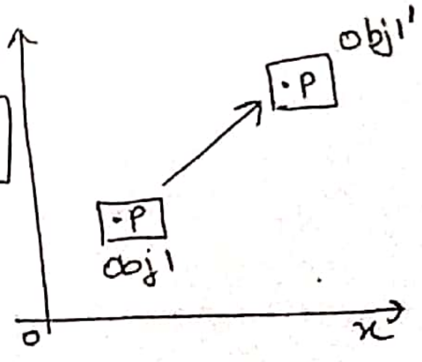
Coordinate transformation

object is held stationary, while the co-ordinate system is moved relative to the object.

TRANSLATION

Translation is the process of changing the position of an object. Let an object point $P(x, y) = xi + yj$ be moved to $p'(x', y')$ by the given translation vector $V = t_x i + t_y j$, where t_x and t_y is the translation factor in x & y directions, such that

$$\boxed{\begin{aligned} x' &= x + t_x \\ y' &= y + t_y \end{aligned}} \Rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}$$

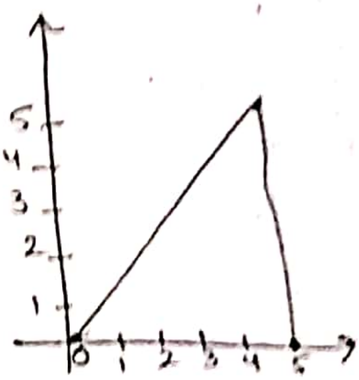


Q.1 Translate a triangle ABC with coordinates A(0,0), B(5,0) C(5,5) by 2 units in x-direction & 3 units in y-direction.

Sol

$x' = x + t_x$

$y' = y + t_y$



Given :- t_x , $t_y = 3$

A(0,0) ; A' = , $x' = x + t_x$

$= 0 + 2$

$x' = 2$

$y' = 0 + 3 = 3$

A'(2,3)

B(5,0) ; B' = ?

$x' = 5+2 = 7$; $y' = 0+3 = 3$

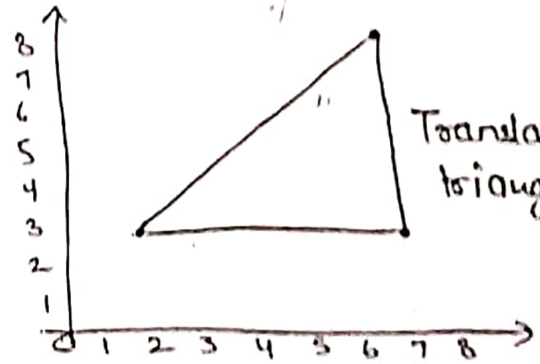
B'(7,3)

C(5,5) ; C' = ?

$x' = 5+2 = 7$; $y' = 5+3 = 8$

C'(7,8)

New triangle coordinates :- A'(2,3) B'(7,3) C'(7,8)



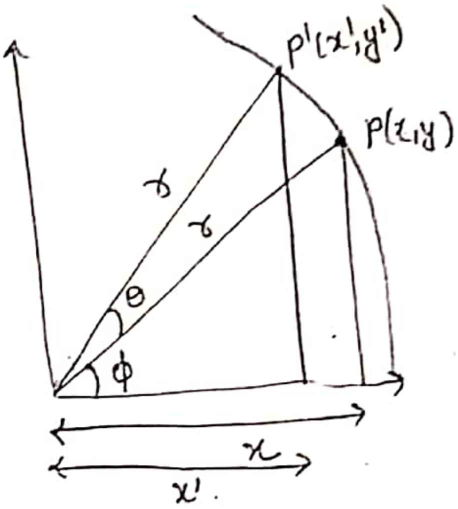
Translated triangle.

Q2 Translate a square ABCD with coordinates (0,0) (2,0) (0,2) (2,2) by 2 units in x-direction & 3 units in y-direction.

ROTATION

In 2-D rotation, an object is rotated by an angle θ with respect to the origin. This angle is assumed to be positive for anticlockwise rotation.

Suppose rotation by θ transform the point $P(x,y)$ into $P'(x',y')$. Because the rotation is about the origin, the distances from the origin to P and to P' is r are equal.



$$x = r\cos\phi \quad ; \quad y = r\sin\phi$$

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

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

Put the values of x and y, we get.

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

In matrix form :-

$$[x'] = [x][T] \quad \text{or} \quad [x' \; y'] = [x \; y][T]$$

$$[x\cos\theta - y\sin\theta \quad x\sin\theta + y\cos\theta] = [x \; y][T]$$

Hence, $[T] = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$; $\cos^2\theta + \sin^2\theta = 1$

$$\underbrace{\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}}_{\text{for anti-clockwise}}$$

+ve direction

$$\underbrace{\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}}_{\text{for clockwise}}$$

−ve direction

Q. find the matrix that represents rotation of an ~~object by~~ point (2,−4)P by 45° about the origin.

Sol

$$R_{45^\circ} = \begin{bmatrix} \cos 45^\circ & -\sin 45^\circ \\ \sin 45^\circ & \cos 45^\circ \end{bmatrix}$$

$$= \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 2 \\ -4 \end{bmatrix} = \begin{bmatrix} \frac{\cancel{2}\sqrt{2}}{\cancel{\sqrt{2}}} + \frac{\cancel{4}\,2\sqrt{2}}{\cancel{\sqrt{2}}} \\ \frac{\cancel{2}\sqrt{2}}{\cancel{\sqrt{2}}} - \frac{\cancel{4}\,2\sqrt{2}}{\cancel{\sqrt{2}}} \end{bmatrix}$$

$$\boxed{\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \sqrt{2}+2\sqrt{2} \\ \sqrt{2}-2\sqrt{2} \end{bmatrix}}$$

SCALING

Scaling is a transformation that changes the size or shape of an object. Scaling with respect to origin can be carried out by multiplying the co-ordinate values (x, y) of each vertex of a polygon or each endpoint of a line by scaling factors S_x and S_y respectively to produce the coordinates (x', y').

$$x' = S_x \cdot x$$
$$y' = S_y \cdot y \quad \Rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

If the scale factor is $0 < S < 1$, then it reduces the size of an object and if it is more than 1, it magnifies the size of the object along an axis.

Q. Scale the polygon A(2,5) B(7,10) & C(10,2) by 2 units in x-direction & 2 units in y-direction.

Sol $S_x = 2$, $S_y = 2$

$$\begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = \cancel{\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}} \begin{bmatrix} 2 & 5 \\ 7 & 10 \\ 10 & 2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

$$\begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = \begin{bmatrix} 4 & 10 \\ 14 & 20 \\ 20 & 4 \end{bmatrix}$$

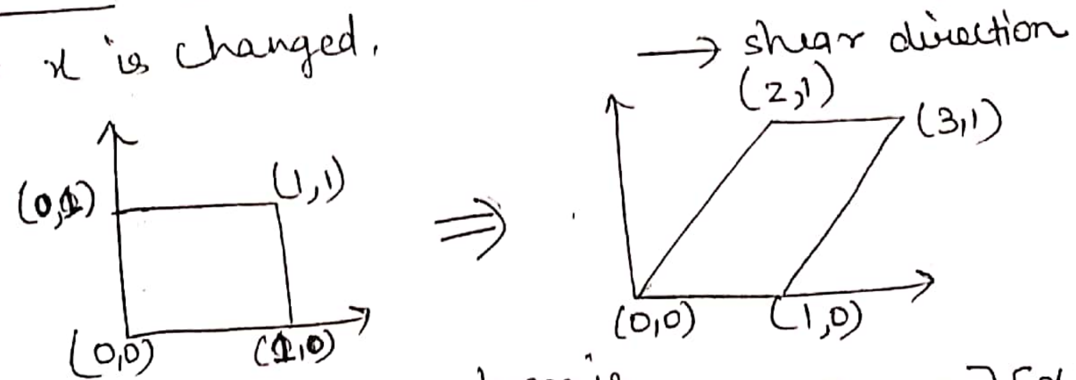
A' (4,10)
B' (14,20)
C' (20,4)

SHEARING

The transformation "shearing" when applied to any object results only in distortion of shape. In shearing, opposite and parallel layers of any object are simply slided with respect to each other.

Shearing can be done either in x-direction or y-direction.

(a) X-shear = In x-shear, y-coordinate remains unchanged, but x is changed.



Transformation for x-shear is

$$\left.\begin{aligned} x' &= x + ay \\ y' &= y \end{aligned}\right\} = Sh_x(a) \Rightarrow \underbrace{\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}}_{\text{shearing in x direction.}}$$

y-shear is :-

$$\left.\begin{aligned} x' &= x \\ y' &= bx + y \end{aligned}\right\} = Sh_y(b) = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ b & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

Scanned by CamScanner

Q. A square ABCD is given with vertices $A(0,0)$ $B(1,0)$ $C(1,1)$ $D(0,1)$. Illustrate the effect of x-shear. when $a=2$

Sol x-shear

$y' = y$

$x' = x +$

$A(0,0)$; $y' =$

$x' = 0 + 2\times 0 = 0$

$A'(0,0)$

$B(1,0)$ $y' = 0$

$x' = 1 + 2\times 0 = 1$

$B'(1,0)$

$C(1,1)$; $y' = 1$

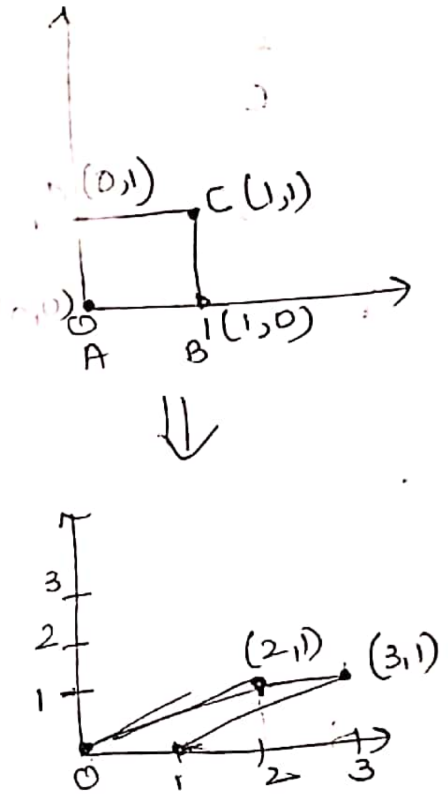
$x' = 1 + 2\times 1 = 3$

$C'(3,1)$

$D(0,1)$; $y' = 1$

$x' = 0 + 2\times 1 = 2$

$D'(2,1)$



★ Calculate y-shear when $b=3$

Ans $A'(0,0)$

$B'(1,3)$

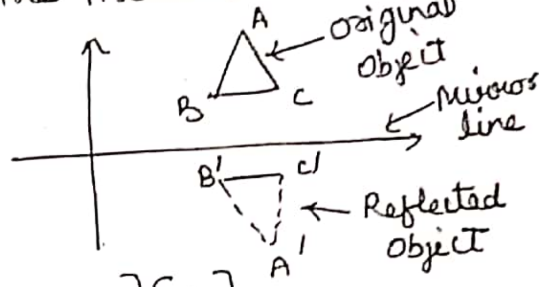
$C'(1,4)$

$D'(0,1)$

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.

a) Reflection about x-axis



x-coordinate is not changed and sign of y-coordinate is changed.

$(x, y) \Rightarrow (x, -y)$

$x' = x$

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

b) Reflection about y-axis :- flips x-coordinate, y-coordinate same.

$(x, y) \Rightarrow (-x, y)$

$x' = -x$

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

c) Reflection about Origin :- both x & y flips

$(x, y) \Rightarrow (-x, -y)$

$x' = -x$

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

d) Reflection about $y = x$.

If we reflect point about the line $y = x$, we get (y, x) i.e

$x' = y$

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

e) Reflection about $y = -x$

$x' = -y$

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

Q. Reflect the point (2,3) about x-axis

Sol
$$x' = x$$
$$y' = -y$$
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ -3 \end{bmatrix}$$
$$\boxed{\begin{matrix} x' = 2 \\ y' = -3 \end{matrix}}$$

HOMOGENEOUS COORDINATES

The general form of transformation matrix is in the form:

$[X'] = [T_1][X] + [T_2]$ where $T_1 \Rightarrow$ multiplicative factor
$T_2 \rightarrow$ Translational terms.

Such transformations cannot be combined to form a single resultant representative matrix.

This problem can be eliminated if we can combine $[T_1]$ and $[T_2]$ into a single transformation matrix. This can be done by expanding the 2x2 transformation matrix format into 3x3 form.

$$\begin{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} & \begin{matrix} m \\ n \end{matrix} \\ \begin{matrix} 0 & 0 \end{matrix} & 1 \end{bmatrix} \Rightarrow$$
a, b, c, d are the multiplicative factors of $[T_1]$
m, n are the translational factors of $[T_2]$.

There is a need to create a dummy ~~variable~~ coordinate to make 2x1 position vector matrix $\begin{bmatrix} x \\ y \end{bmatrix}$ to a 3x1 matrix

$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ where the third coordinate is dummy.

If we multiply $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ with a non-zero scalar 'h' then the matrix forms is :- $\begin{bmatrix} xh \\ yh \\ h \end{bmatrix} \Rightarrow \begin{bmatrix} x_h \\ y_h \\ h \end{bmatrix}$ which is known as homogeneous coordinates

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x_h \\ y_h \\ h \end{bmatrix} = h \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

'h' is known as weight.

∴ for geometric transformation, $h=1 \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$.

Translation

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

Rotation

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

Scaling :-

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

Q Translate the square ABCD whose coordinate are A(0,0) B(3,0) C(3,3) and D(0,3) by 2 units in both directions and then scale it by 1.5 units in x-direction & 0.5 units in y-direction.

Sol

Translation Matrix $= \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$

$\cancel{\text{Square ABCD} = \begin{bmatrix} 0 & 0 & 1 \\ 3 & 0 & 1 \\ 3 & 3 & 1 \\ 0 & 3 & 1 \end{bmatrix}}$ Square ABCD $= \begin{bmatrix} 0 & 3 & 3 & 0 \\ 0 & 0 & 3 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

Translation :-

$$\cancel{\begin{bmatrix} A' \\ B' \\ C' \\ D' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 3 & 0 & 1 \\ 3 & 3 & 1 \\ 0 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 7 \\ 3 & 0 & 7 \\ 3 & 3 & \end{bmatrix}}$$

Translation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 3 & 3 & 0 \\ 0 & 0 & 3 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{array}{c} \begin{matrix} A' & B' & C' & D' \end{matrix} \\ \begin{bmatrix} 2 & 5 & 5 & 2 \\ 2 & 2 & 5 & 5 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

A' (2,2)
B' (5,2)
C' (5,5)
D' (2,5)

Scaling matrix

$$\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 5 & 5 & 2 \\ 2 & 2 & 5 & 5 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 3 & 7.5 & 7.5 & 3 \\ 1 & 1 & 2.5 & 2.5 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

After scaling, new coordinates are :-

$A'(3,1)$ $B'(7.5,1)$ $C'(7.5,2.5)$ $D'(3,2.5)$

Q. A triangle is defined by $\begin{bmatrix} 2 & 4 & 4 \\ 2 & 2 & 4 \end{bmatrix}$ Find the transformed coordinates after the following transformation :-

(i) 90° rotation about origin

(ii) reflection about line $y=-x$

Ans (i) $\begin{bmatrix} -2 & -2 & -4 \\ 2 & 4 & 4 \\ 1 & 1 & 1 \end{bmatrix}$

Ans (ii) $\begin{bmatrix} -2 & -4 & -4 \\ 2 & 2 & -4 \\ 1 & 1 & 1 \end{bmatrix}$

Combined Transformations

i) Rotation about an Arbitrary Point

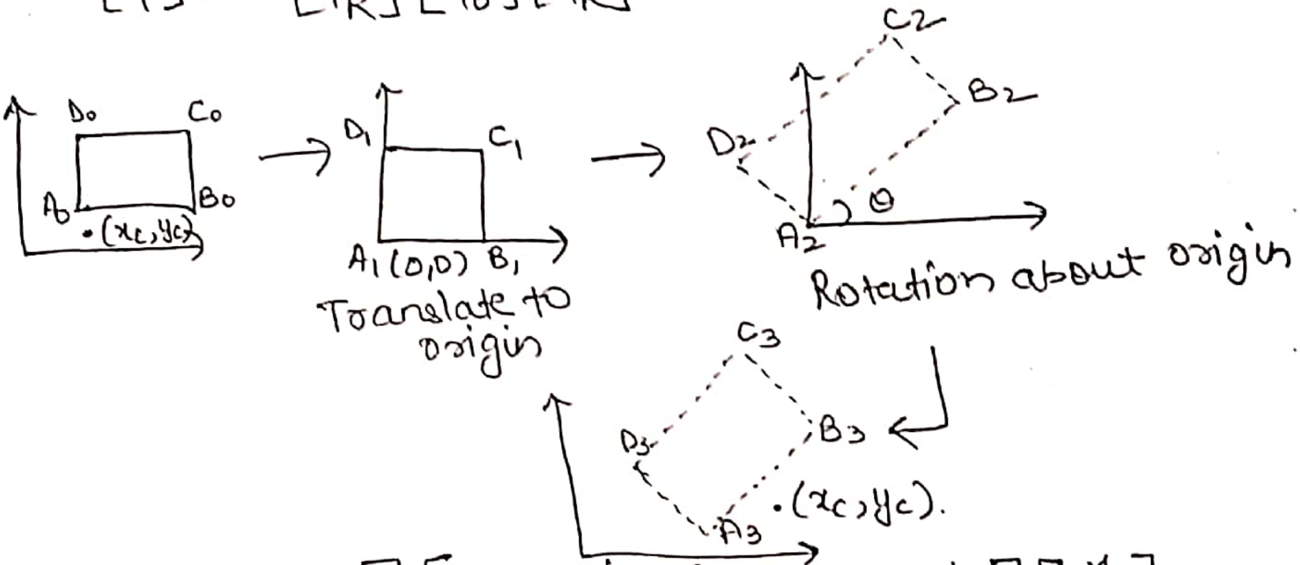
Step 1 :- Translate the object or body at the origin

Step 2 :- Rotate by any angle as given.

Step 3 :- Translate back to its original location.

In Matrix form :-

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



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

$$= \underbrace{\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & -t_x\cos\theta + t_y\sin\theta \\ \sin\theta & \cos\theta & -t_x\sin\theta - t_y\cos\theta \\ 0 & 0 & 1 \end{bmatrix}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

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

$$\boxed{\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & -t_x(\cos\theta - 1) + t_y\sin\theta \\ \sin\theta & \cos\theta & -t_y(\cos\theta - 1) - t_x\sin\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}$$

Rotation about arbitrary point

Q. Perform a 45° rotation of triangle A(0,0) B(1,1) C(5,2) about point P(-1,-1)

Sol

$$R_{45^\circ} = \begin{bmatrix} \cos 45^\circ & -\sin 45^\circ & 0 \\ \sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Translation point $(-1,-1)$ $\quad t_x = -1$ & $t_y = -1$

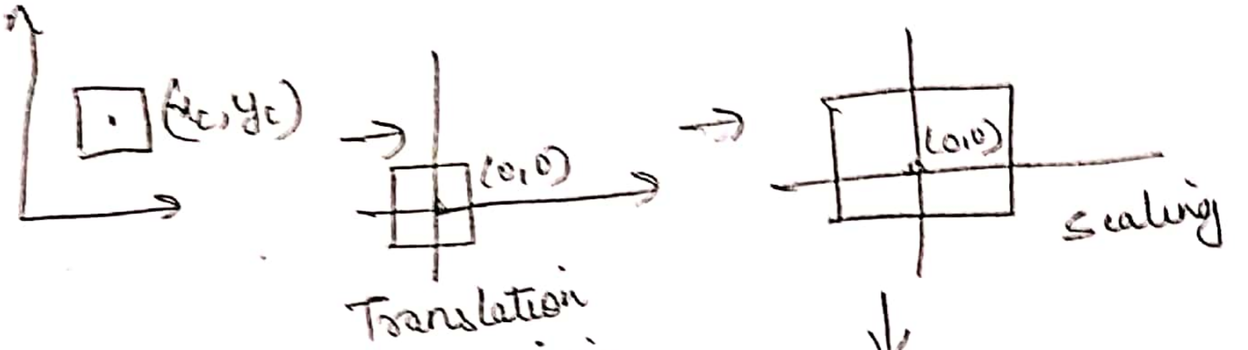
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}.$$

Ans.

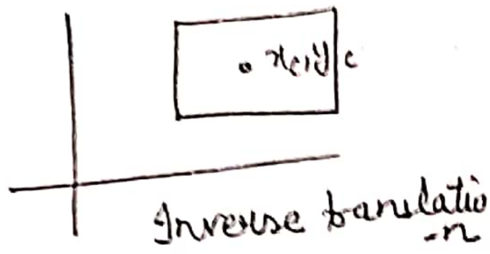
$$\boxed{\begin{aligned} A' &= 1, \sqrt{2}-1 \\ B' &= 1, 2\sqrt{2}-1 \\ C' &= \tfrac{3}{2}\sqrt{2}-1, \tfrac{9}{2}\sqrt{2}-1 \end{aligned}}$$

Scanned by CamScanner

2) Fixed-Point Scaling



(1) Translate point to origin
(2) Perform scaling
(3) Inverse translation



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

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

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

$$\boxed{\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & -t_x(S_x - 1) \\ 0 & S_y & -t_y(S_y - 1) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}$$

Scaling about fixed point

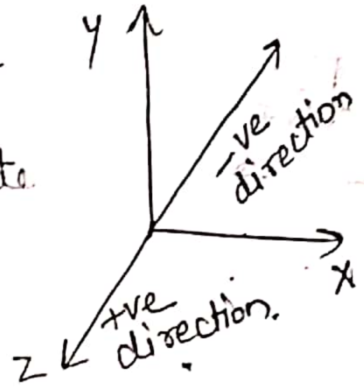
Q. Magnify the triangle with vertices A(0,0) B(1,1) & C(5,2) to twice its size ~~with~~ while keeping C(5,2) fixed

~~C(5,2) fixed.~~

Ans A'(−5, −2)
B'(−3, 0)
C'(5, 2)

3-D Transformation

To represent 3D object we need 3 parameters, x coordinate which represent width, y coordinate represent height & z coordinate represent depth.



In 3D - there are 3 axis :- x, y & z.

Translation

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

Scaling.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}.$$

Rotation about x-axis (change y & z)

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

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

about y-axis (change x & z)

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