A

Complete Note on
Computer Graphics

By : Bishwas Pokharel
=> Pulchowk Campus.

_What is Computer Graphics?

A graphic is an image or visual representation of an object. Therefore, computer graphics are simply images displayed on a computer screen. Graphics are often contrasted with text, which is comprised of characters, such as numbers and letters, rather than images.

In addition, Computer graphics is an art of drawing pictures, lines, charts, etc using computer with the help of programming. Basically, there are two types of computer graphics namely.

Interactive Computer Graphics: Interactive Computer Graphics involves a two-way communication between computer and user. Here the observer is given some control over the image by providing him with an input device.

For example, it helps to train the pilots of our airplanes. We can create a flight simulator which may help the pilots to get trained not in a real aircraft but on the grounds at the control of the flight simulator. The flight simulator is a mock-up of an aircraft flight deck, containing all the usual controls and surrounded by screens on which we have the projected computer-generated views of the terrain visible on take-off and landing. Flight simulators have many advantages over the real aircrafts for training purposes, including fuel savings, safety, and the ability to familiarize the trainee with a large number of the world's airports.

Other examples: video games, word, power point etc.

Non -Interactive Computer Graphics: In non-interactive computer graphics otherwise known as passive computer graphics. it is the computer graphics in which user does not have any kind of control over the image. Image is merely the product of static stored program and will work according to the instructions given in the program linearly. The image is totally under the control of program instructions not under the user.

Example: screen savers, movies, T.V etc.

What are the application field of computer graphics??

Computer graphics is used today in many different areas of science, engineering, industry business, education, entertainment, medicine art and training.

All of these are included in the following categories

1. User interfaces

most applications have user interfaces that rely on the desktop window systems to manage multiple simultaneous activities and on point and click facilities to allow users to select menu items, icons, and objects on the screen These activities fall under computer graphics. Typing is necessary only to input text to be stored and manipulated. For example, word processing spreadsheet and desktop publishing programs are the typical examples where user interface techniques are implemented.

2. Plotting

plotting 2D and 3D graphs of mathematical physical and economic functions use computer graphics extensively. The histograms, bar and pie charts, the task scheduling charts, are the most commonly used plotting. These are all used to present meaningfully and concisely the trends and patterns of complex data.

3. Office automation and electronic publishing

computer graphics has facilitated the office automation and electronic publishing which is also popularly known as desktop publishing, giving more power to the organizations to print the meaningful materials in house. Office automation and electronic publishing can produce both traditional printed (hardcopy) documents and electronic (softcopy) documents that contain text tables, graphs and other forms of drawn or scanned in graphics.

4. Computer aided drafting and design

one of the major uses of computer graphics is to design components and systems of mechanical, electrical, electrochemical and electronic devices, including structures such as buildings automobile bodies, airplane and ship hulls, very large scale integrated (VLSI) chips optical systems and telephone and computer networks. These designs are more frequently used to test structural, electrical and thermal properties of the systems.

5. Scientific and business visualization

Generating computer graphics for scientific, engineering and medical data sets is termed as scientific visualization whereas business visualization is related with the non-scientific data sets such as those obtained in economics. Visualization makes easier to understand the trends and patterns inherent in huge amount of data sets. It would otherwise be almost impossible to analyze those data numerically

6. Simulation

Simulation is the imitation of the conditions like those, which is encountered in real life.

Simulation thus helps to learn or to feel the conditions one might have to face in near future without being in danger at the beginning of the course. For example, astronauts can exercise the feeling of weightlessness in a simulator, similarly a pilot training can be conducted in a flight simulator. The military tank simulator the naval simulator, driving simulator, air traffic control simulator, heavy duty vehicle simulator and so on are some of the mostly used simulator in practice. Simulators are also used to optimize the system

For example: the vehicle, observing the reactions of the driver during the operation of the Simulator.

7. Entertainment

Disney movies such as Lion King and the beauty and the beast, and other scientific movies like star trek are the best examples of the application of computer graphics in the field of entertainment. Instead of drawing all the necessary frames with slightly changing scenes for the production of cartoon film only the key frames are sufficient for such cartoon film where the in between frames are interpolated by the graphics system dramatically decreasing the cost production while maintaining the quality. Computer and video games such as Fifa, Formula-1, Doom and Pools are few to name where computer graphics is used extensively

8. Art and commerce

Here computer graphics is used to produce pictures that expresses a message and attract attention such as a new model of a car moving along the ring of the Saturn. These pictures are frequently seen at transportation terminals, supermarkets, hotels etc. the slide production for commercial, scientific, or educational presentations is another cost-effective use of computer graphics. One of such graphics packages is “PowerPoint”.

9. Cartography

Cartography is a subject which deals with the making of maps and charts. Computer graphics is used to produce both accurate and schematic representations of geographical and other natural phenomena from measurement data. Examples include geographic maps, oceanographic charts, weather maps, contour maps and population density maps

Surfer is one of such graphics packages which is extensively used for cartography.

Explain In brief the History of computer graphics.

Years of research and development were made to achieve the goals in the field of computer graphics. In 1950 the first computer driven display was used to generate only simple pictures. This display made use of a cathode ray tube similar to the one used in television sets. During 1950's interactive computer graphics made little progress because the computers of that period were so unsuited to interactive use. These computers were used to perform only lengthy calculations.

The single vent that did the most to promote interactive computer graphics as an important new field was the publication in 1962 of a brilliant thesis by Ivan E. Sutherland. His thesis, entitled 'Sketchpad: A Man- Machine Graphical Communication System proved to many readers that interactive computer graphics was a viable, useful, and exciting field of research. By the mid - 1960's large computer graphics research projects were under taken at MIT, Bell Telephone Labs and General Motors. Thus, the golden age of computer graphics began. In 1970's thee researches began to bear fruit.

The instant appeal of computer graphics to users of all ages has helped it to spread into many applications throughout the world.

What is Monitor?

Monitor is another word for the computer *screen*. But "monitor" encompasses the whole piece of equipment, rather than just the screen part that you look at. You also might hear a monitor called a *display,* as in "Oooh, I got a new two-page display," or *VDT (video* display terminal), as in newspaper journalism, or *CRT* (cathode ray tube), which is the technical term for a picture tube. However, flat panel screens like *LCDS* are not referred to as monitors, even if they're housed externally from a computer.

Some monitors are built right into the computers, like in the small Macintoshes. When you purchase a larger Macintosh or most other kinds of computers, you must buy the moor separate from the computer itself (that's why they're called "modular"). Monitor size is measured like a television, from one corner to the diagonally opposite corner.

Some monitors are *monochrome,* meaning they can show only one color on a background, like black on white (Macs), green on black, or amber on black (pcs). *Grayscale* monitors can display different shades of gray, rather than imitating the different shades with combinations of black and white dots. And there are many different color monitors. A color monitor can display any of several levels of resolution and can display varying numbers of colors, determined by several factors, such as amount of memory in the computer or the type of *card* that is controlling the monitor.

By : Bishwa pokharel

CATHODE RAY TUBE:

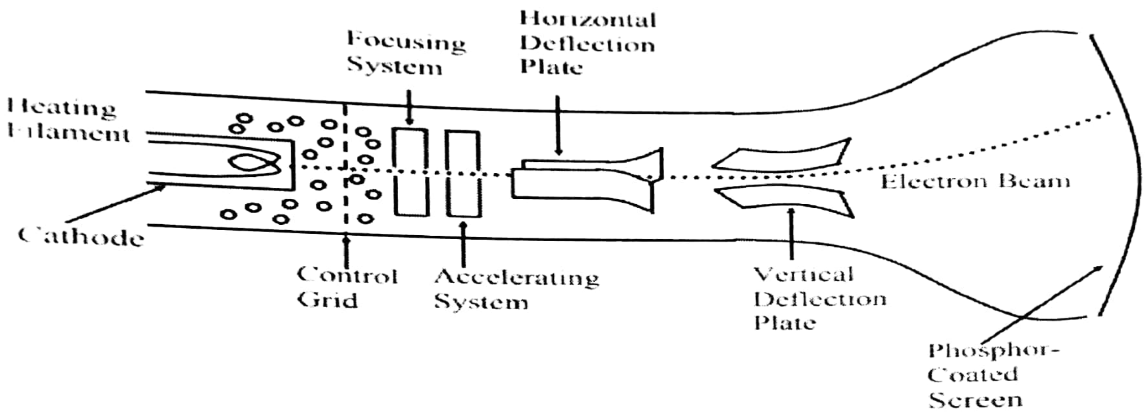


Figure :Cathode Ray Tube

The primary output device in a graphics system is a video monitor. The operation of most video monitors is based on the standard cathode ray tube (CRT) design.

A CRT is an evacuated glass tube. An electron gun at the rear of the tube produces a beam of electrons which is directed towards the front of the tube (screen). The inner side of the screen is coated with phosphor substance which gives off light when it is stroked by electrons. It is possible to control the point at which the electron beam strikes the screen, and therefore the position of the dot upon the screen, by deflecting the electron beam.

The beam is positioned on the screen by a deflection system of the cathode-ray-tube consists of two pairs of parallel plates, referred to as the vertical and horizontal deflection plates. The intensity of the beam is controlled by the intensity signal on the control grid.

The voltage applied to vertical plates controls the vertical deflection of the electron beam and voltage applied to the horizontal deflection plates controls the horizontal deflection of the electron beam. There are two techniques used for producing images on the CRT screen: Vector scan / random scan and Raster scan.

When the phosphor is hit by the electron beam it absorbs energy and jumps to a higher quantum-energy level. light emitted as these very unstable electrons lose their excess energy whole the phosphor is being struck by electrons is called **phosphors fluorescence.**

As it returns to its normal level it emits visible light i.e. it **phosphoresces.** In the phosphors used in graphics devices the persistence of the phosphorescence is typically 10-60 microseconds.

Before the human visual system can see a transient image it must be continually redrawn (refreshed) at a rate higher than the critical fusion frequency of the human visual system. To allow the human visual system to see a continuously refreshed image without flicker the refresh rate has to be at least 60 c/s.

What is a refresh rate and why is a monitor's refresh rate important?

An image appears on screen when electron beams strike the surface of the screen in a zig-zag pattern. A refresh rate is the number of times a screen is redrawn in one second and is measured in Hertz (Hz). Therefore, a monitor with a refresh rate of 85 Hz is redrawn 85 times per second. A monitor should be "flicker-free meaning that the image is redrawn quickly enough so that the user cannot detect flicker, a source of eye strain. Today, a refresh rate of 75 Hz or above is considered to be flicker-free

What is persistence?

The time it takes the emitted light from the screen to decay one tenth of its

original intensity is called as persistence.

How are refresh rates calculated?

Factors in determining refresh rates:

A refresh rate is dependent upon a monitor's horizontal scanning frequency and the- number of horizontal lines displayed. The horizontal scanning frequency is the number of times the electron beam sweeps one line and returns to the beginning of the next in one second. Horizontal scanning frequency is measured in kilohertz (kHz). A monitor with a horizontal scanning frequency of 110 kHz means 110,000 lines are scanned per second.

The number of horizontal lines on the screen depends upon the monitor's resolution. If a monitor is set to a resolution of 1024 x 768 then there are 768 horizontal lines (1024 is the number of pixels on one line). For a monitor set to a 1280 x 1024 resolution, there are 1024 horizontal lines.

Additionally, the time it takes for the electron beam to return to the top of the screen and begin scanning again must be taken into account. This is roughly 5% of the time it takes to scan the entire screen. Therefore, the total is multiplied by 0.95 to calculate the maximum refresh rate.

How to calculate maximum refresh rates?

The following formula is used to calculate maximum refresh rates fV = (fH / # of horizontal lines) x 0.95

fV = vertical scanning frequency (refresh rate) fH = horizontal scanning frequency

Example: A monitor with a horizontal scanning frequency of 96 kHz at a resolution of 1280 x 1024. Determine refresh rate based on the calculation.

$fV = (96{,}000 / 1024) \times 0.95$

$fV = 89.06$

This figure is rounded down to produce a maximum refresh rate of 89Hz.

If the same monitor is set to a resolution of 1600 x 1200, then the equation will be as follows:

$fV = (96{,}000 / 1200) \times 0.95$

$fV = 76$

The maximum refresh rate at this resolution is 76 Hz.

What Is pixel?

A pixel (short for picture element, using the common abbreviation "pix" for "picture") is one of the many tiny dots that make up the representation of a picture in a computer's memory. Each such information element is not really a dot, nor a square, but an abstract sample.

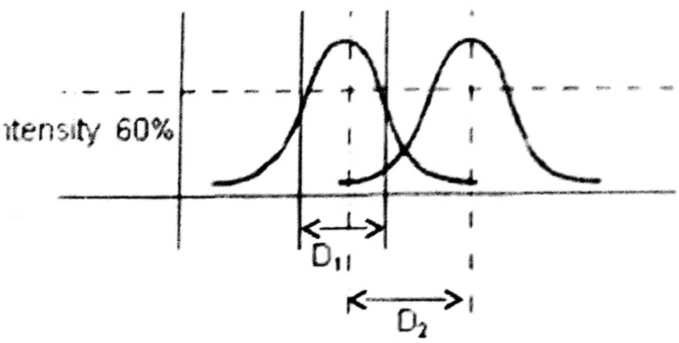
With care, pixels in an image can be reproduced at any size without the appearance of visible dots or squares; but in many contexts, they are reproduced as dots or squares and can be visibly distinct when not fine enough. The intensity of each pixel is variable; in color systems, each pixel has typically three or four dimensions of variability such as red, green and blue, or cyan, magenta, yellow and black.

Resolution

Resolution is defined as the maximum number of points that can be displayed horizontally and vertically without overlap on a display device.

Factors affecting the resolution are as follows

i. Spot profile: The spot intensity has a Gaussian distribution as depicted in figure. So two adjacent spots on the display device appear distinct as long as their separation D2 is greater than the diameter of the spot D1 at which each spot has an intensity of about 60 percent of that at the center of the spot.



ii. Intensity: as the intensity of the electron beam increases the spot size on the display tends to increase because of spreading of energy beyond the point of bombardment

Resolution of a CRT depends

o Type of phosphor

o Intensity to be displayed

o Focusing and deflection system

What is Image Resolution

Image resolution describes the detail an image holds. The term applies equally to digital images, film images, and other types of images. Higher resolution means more image detail. Image resolution can be measured in various ways. Basically, resolution quantifies how close lines can be to each other and still be visibly resolved.

Resolution units can be tied to physical sizes (e.g. lines per mm, lines per inch) or to the overall size of a picture (lines per-picture height, also - known simply as lines, or TV lines). Furthermore, line pairs are often used instead of lines. A line pair is a pair of adjacent dark and light lines, while a line counts both dark lines and light lines. A resolution of 10 lines per mm means 5 dark lines alternating with 5 light lines, or 5line pairs per mm. Photographic lens and film resolution are most often quoted in line pairs per mm.

Different Kinds of Resolutions in the Monitor

Resolution refers to the sharpness, or detail of the usual image. It's a primary function of the monitor & it's determined by the beam size & dot pitch. The screen is made up of a number of pixels.

A completes screen image consists of thousand of pixels & the screen resolution in the maximum no. of displayable pixels. Higher the resolution, the more pixels can be displayed. Resolutions are of different for different video standards as listed below :

(a) **VGA :** 1640 x 480

(b) **SVGA :** 800 x 600

(c) **XGA :** 1024 x 768

(d) **SXGA :** 1400 x 1050

What is Aspect ratio?

Aspect ratio is a fancy term for "proportion," or the ratio of width to height. for ex 4:3 for a computer screen. For instance, if a direction in a software manual tells you to "hold down the

Shift key while you resize a graphic in order to maintain the aspect ratio," it simply means that if you *don't* hold down the Shift key you will stretch the image out of proportion.

Some combinations of computers and printers have trouble maintaining the correct **aspect ratio** when the image goes from the screen to the printer, or when the image is transferred from one system to another, so the aspect ratio can be an important specification to consider when choosing hardware.

The aspect ratio of the screen determines the most efficient screen RESOLUTIONS and the most desirable shape for individual PIXELS, all of which may have to change upon the introduction of HIGH DEFINITION TELEVISION.

Difference between Raster Scan System and Random Scan System

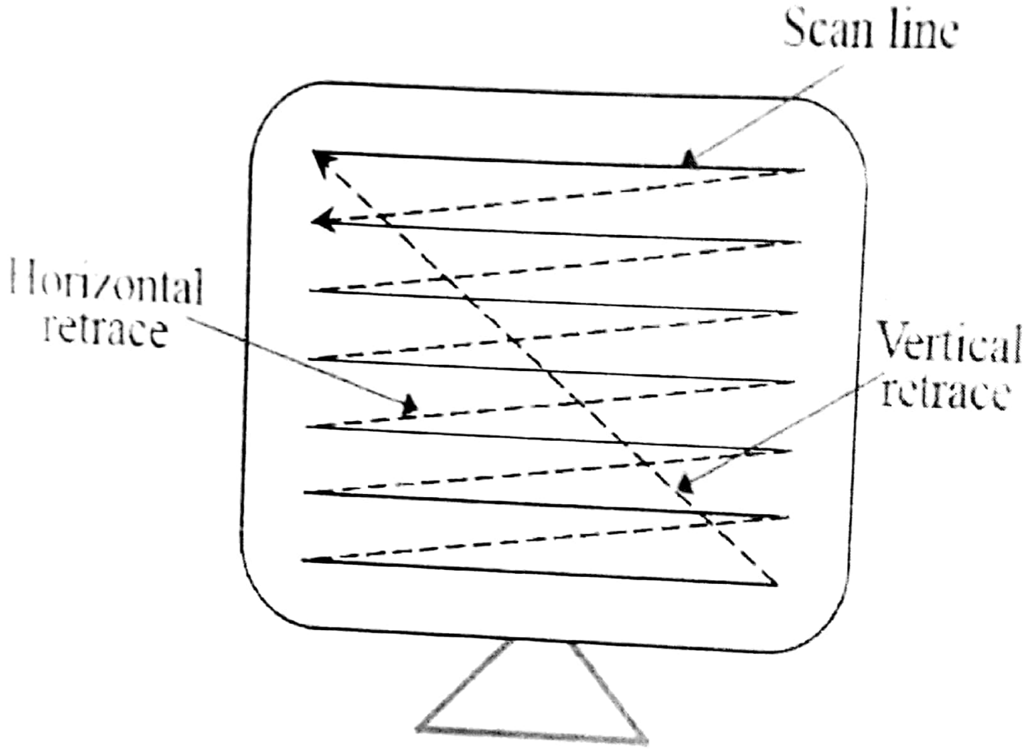
Base of difference	Raster	Random (vector)
Electric Beam	The electron beam is swept accross The screen one row at a time Top tp bottam	The electron beam is directed only to the parts of screen where a picture is to be drawn
Picture Defination	Its resolution system is poor because raster system in contrast producse zig zag lines that are plotted as discrete point sets	Its resolution system is good because this system producse smooth lines drawings because CRT beams directly follows the line path
Resolution	Picture defination is stored as a ste of intensity value for all screen points called pixel in a fresh buffer area	Picture defination is stored as a set of line drawing insructions in a display file
Realistic display	The capability of this system to store intensity values for pixel makes it well suited for the realstic display of scenes contains shadow and color pattern	This system are designed for line drawing and can't display realstic shaded scenes
Draw an Image	Screen pont pixel are used to draw an image	Mathematical function are used to draw an image

Raster Scan

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 (scan line) at a time as shown in the following illustration.

Each screen point is referred to as a **pixel (picture element)** or **pel**. 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.

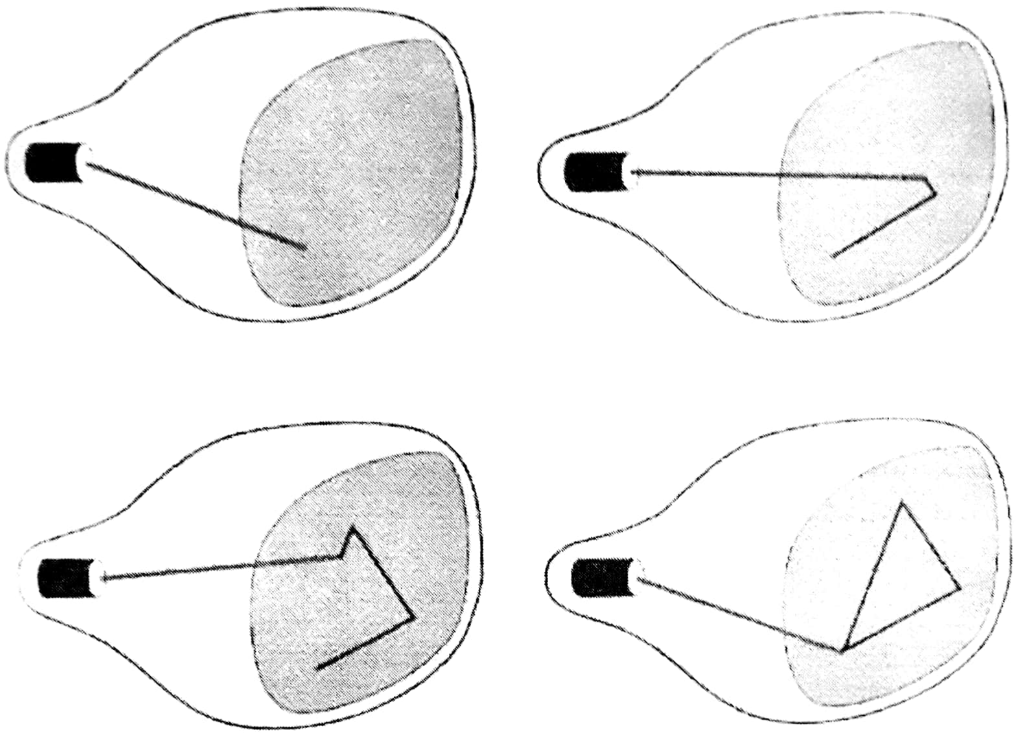


Random Scan (Vector Scan)

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 as in raster scan. It is also called **vector display, stroke-writing display,** or **calligraphic display**.

Picture definition is stored as a set of line-drawing commands in an area of memory referred to as the **refresh display file**. To display a specified picture, the system cycles through the set of commands in the display file, drawing each component line in turn. After all the line-drawing commands are processed, the system cycles back to the first line command in the list.

Random-scan displays are designed to draw all the component lines of a picture 30 to 60 times each second.



Explain about Color CRT?

In color CRT, the phosphor on the face of CRT screen are laid in to different fashion. Depending on the technology of CRT there are two methods for displaying the color pictures into the screen.

1. Beam penetration method 2. Shadow mask method

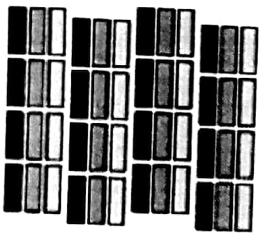
Beam Penetration method:

This method is commonly used for random scan display or vector display. In random scan display CRT, the two layers of phosphor usually red and green are coated on CRT screen. Display color depends upon how far electrons beam penetrate the phosphor layers. Slow electron excite only red layer so that we can see red color displayed on the screen pixel where the beam strikes. Fast electron beam excite green layer penetrating the red layer and we can see the green color displayed at the corresponding position. Intermediate is combination of red and green so two additional colors are possible – orange and yellow. So only four colors are possible so no good quality picture in this type of display method.

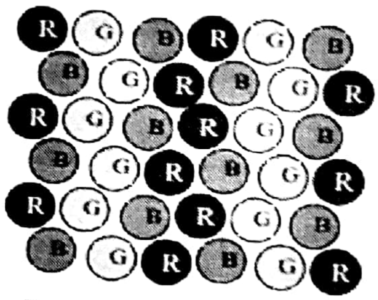
Shadow Mask Method:

Shadow mask method is used for raster scan system so they can produce wide range of colors. In shadow mask color CRT, the phosphor on the face of the screen are laid out in a precise geometric pattern. There are two primary variations.

1. The stripe pattern of inline tube
2. The delta pattern of delta tube



Stripe pattern



Delta Pattern

- In color CRT, the neck of tube, there are three electron guns, one for each red, green and blue colors. In phosphor coating there may be either strips one for each primary color, for a single pixel or there may be three dots one for each pixel in delta fashion.
- Special metal plate called a shadow mask is placed just behind the phosphor coating to cover front face.
- The mask is aligned so that it simultaneously allow each electron beam to see only the phosphor of its assigned color and block the phosphor of other two color.

Depending on the pattern of coating of phosphor, two types of raster scan color CRT are commonly used using shadow mask method.

1. Delta-Delta CRT:

- In delta-delta CRT, three electron beams one for each R,G,B colors are deflected and focused as a group onto shadow mask, which contains a series of holes aligned with the phosphor dots.

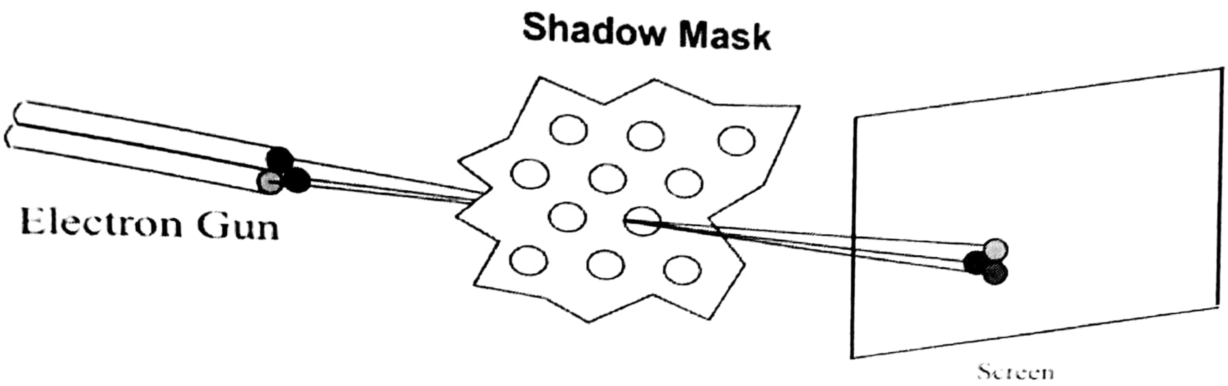


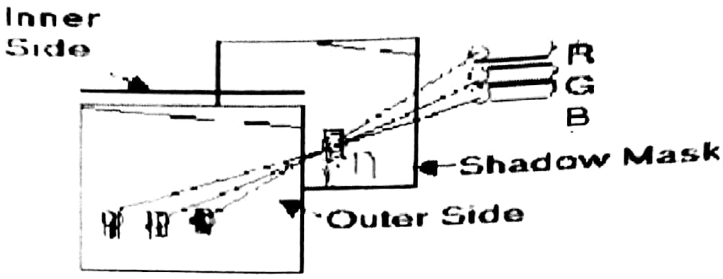
Figure: Shadow mask in Delta-Delta CRT

- Inner side of viewing has several groups of closely spaced red ,green and blue phosphor dot called triad in delta fashion.
- Thin metal plate adjusted with many holes near to inner surface called shadow mask which is mounted in such a way that each hole aligned with respective triad.
- Triad are so small that is perceived as a mixture of colors. When three beams pass through a hole in shadow mask, they activate the dot triangle to illuminate an small spot colored on the screen.
- The color variation in shadow mask CRT can be obtained by varying the intensity level of the three electron guns.

The main drawback of this CRT is due to difficulty for the alignment of shadow mask hole and respective triads.

2. A precision inline CRT:

This CRT uses strips pattern instead of delta pattern. Three strips one for each R, G, B color are used for a single pixel along a scan line so called inline. This eliminates the drawbacks of delta-delta CRT at the cost of slight reduction of image sharpness at the edge of the tube.



Normally 1000 scan lines are displayed in this method. Three beams simultaneously expose three inline phosphor dots along scan line.

Hardware Concepts

Input Devices

1. Tablet:

A tablet is digitizer. In general a digitizer is a device which is used to scan over an object, and to input a set of discrete coordinate positions. These positions can then be joined with straight-line segments to approximate the shape of the original object. A tablet digitizes an object detecting the position of a movable stylus (pencil-shaped device) or puck (link mouse with cross hairs for sighting positions) held in the user's hand. A tablet is flat surface, and its size of the tablet varies from about 6 by 6 inches up to 48 by 72 inches or more. The accuracy of the tablets usually falls below 0.2 mm. There are mainly three types of tablets.

a. Electrical tablet:

A grid of wires on 1/4 to 1/2 inch centers is embedded in the tablet surface and electromagnetic signals generated by electrical pulses applied in sequence to the wires in the grid induce an electrical signal in a wire coil in the stylus (or puck). The strength of the signal induced by each pulse is used to determine the position of the stylus. The signal strength is also used to determine roughly how far the stylus is from the tablet. When the stylus is within 1/2 inch from the tablet, it is taken as "near" otherwise it is either "far" or "touching". When the stylus is "near" or "touching", a cursor is usually shown on the display to provide visual feedback to the user. A signal is sent to the computer when the tip of the stylus is pressed against the tablet, or when any button on the puck is pressed. The information provided by the tablet repeats 30 to 60 time per second.

b. Sonic tablet:

The sonic tablet uses sound waves to couple the stylus to microphones positioned on the periphery of the digitizing area. An electrical spark at the tip of the stylus creates sound bursts. The position of the stylus or the coordinate values is calculated using the delay between when the spark occurs and when its sound arrives at each microphone. the main advantage of sonic tablet is that it does not require a dedicated working area for the microphones can be placed on any surface to form the "tablet" work area. This facilitates digitizing drawing on thick books. Because in an electrical tablet this is not convenient for the stylus can not get closer to the tablet surface.

c. Resistive tablet:

The tablet is just a piece of glass coated with a thin layer of conducting material. When a buttery-powered stylus is activated at certain position, it emits high-frequency radio signals, which induces the radio signals on the conducting layer. The strength of he signal received at the edges of the tablet is used to calculate the position of the Stylus. Several types of tablets are transparent, and thus can be backlit for digitizing x-rays films and photographic negatives. The resistive tablet can be used to digitize the objects on CRT because it can be curved to the shape of the CRT. The mechanism used in the electrical or sonic tablets can also be used to digitize the 3D objects.

2. Touch panel

The touch panel allows the users to point at the screen directly with a finger to move the cursor around the screen, or to select the icons. Following are the mostly used touch panels.

a. Optical touch panel

It uses a series of infra-red light emitting diodes (LED) along one vertical edge and along one horizontal edge of the panel. The opposite vertical and horizontal edges contain photo-detectors to form a grid of invisible infrared light beams over the display area. Touching the screen breaks one or two vertical and horizontal light beams, thereby indicating the finger's position. The cursor is then moved to this position, or the icon at this position is selected. It two parallel beams are broken, the finger is presumed to be centered between them; if one is broken, the finger is presumed to be on the beam. There is a low-resolution panel, which offers 10 to 50 positions in each direction.

b. Sonic panel:

Bursts of high-frequency sound waves traveling alternately horizontally and vertically are generated at the edge of the panel. Touching the screen causes part of each wave to be reflected back to its source. The screen position at the point of contact is then calculated using the time elapsed between when the wave is emitted and when it arrives back at the source. This is a high-resolution touch panel having about 500 positions in each direction.

c. Electrical touch panel:

It consists of slightly separated two transparent plates one coated with a thin layer of conducting material and the other with resistive material. When the panel is touched with a finger, the two plates are forced to touch at the point of contact thereby creating the touched position. The resolution of this touch panel is similar to that of sonic touch panel.

3. Light pen

It is a pencil-shaped device to determine the coordinates of a point on the screen where it is activated such as pressing the button. In raster display, Y is set at Ymax and X changes from 0 to Xmax for the first scanning line. For second line, Y decreases by one and X again changes from 0 to Xmax, and so on. When the activated light pen "sees" a burst of light at certain position as the electron beam hits the phosphor coating at that position, it generates a electric pulse, which is used to save the video controller's X and Y registers and interrupt the computer. By reading the saved values, the graphics package can determine the coordinates of the position seen by the light pen. Because of the following drawbacks the light pens are not popular now a days.

- Light pen obscures the screen image as it is pointed to the required spot
- Prolong use of it can cause arm fatigue
- It can not report the coordinates of a point that is completely black. As a remedy one can display a dark blue field in place of the regular image for a single frame Time
- It gives sometimes false reading due to background lighting in a room.

4. Keyboard

A keyboard creates a code such as ASCII uniquely corresponding to a pressed key. It usually consists of alphanumeric keys, function keys, cursor-control keys, and separate numeric pad. It is used to move the cursor, to select he menu item, pre- defined functions. In computer graphics keyboard is mainly used for entering screen coordinates and text, to invoke certain functions. Now-a-days ergonomically designed keyboard (Ergonomic keyboard) with removable palm rests is available. The slope of each half of the keyboard can be adjusted separately.

5. Mouse

A mouse is a small hand-held device used to position the cursor on the screen. Mice are relative devices, that is, they can be picked up, moved in space, and then put down gain without any change in the reported position. For this, the computer maintains the current mouse position, which is incremented or decremented by the mouse movements. Following are the mice, which are mostly used in computer graphics.

a. Mechanical mouse

When a roller in the base of this mechanical mouse is moved, a pair of orthogonally arranged toothed wheels, each placed in between a LED and a photo detector, interrupts the light path. the number of interrupts so generated are used to report the mouse movements to the computer.

b. Optical mouse

The optical mouse is used on a special pad having a grid of alternating light and dark lines. A LED on the bottom of the mouse directs a beam of light down onto the pad, from which it is reflected and sensed by the detectors on the bottom of the mouse. As the mouse is moved, the reflected light beam is broken each time a dark line is crossed. The number of pulses so generated, which is equal to the number of lines crossed, are used to report mouse movements to the computer.

Numericals

Q1. A system with 24bit pixel and screen resolution 1024 * 1024 require how much storage (in megabyte) in frame buffer?

Ans: 1024*1024 pixel = 1024*1024*24 bits = 3 MB

Q2. Question: Consider a RGB raster system is to be designed using 8 inch by 10 inch screen with a resolution of 100 pixels per inch in each direction. If we want to store 8 bits per pixel in the frame buffer, How much storage (in bytes) do we need for the frame buffer?

Solution: Size of screen = 8 inch *10 inch.

Pixel per inch(Resolution) = 100.

Then, Total no of pixels = 8*100 *10 * 100 pixels

Bit per pixel storage = 8

Therefore Total storage required in frame buffer = (800 *1000*8) bits

= (800*1000*8)/8 Bytes

= 800000 Bytes.

Q3. How long does it take to load a 640*480 frame buffer with 12 bit per pixel if 10^5 bit can be loaded in 1 sec?

Solution: total = 640*480 pixel

total storage needed= 640*480*12 bit

since,

10^5 bit loaded in 1 sec

1 bit loaded in 1/(10^5) sec

640*480*12 loaded in (640*480*12)/(10^5) sec.

Q4. Let the average time to execute an instruction in the display list be 33.33 microsec. If the refresh rate is 30 frame per microsec. Then obtain the maximum number of instructions that may be present in the display list.

Solution: let total instructions number in one frame buffer =N.

1 instruction required 33.33 microsec.

N instructions required 33.33*N microsec.

Now,

30 buffer frames in 1 sec.

1 buffer frame in 1/(30) sec =0.033 sec =33333.33 microsec

From above,

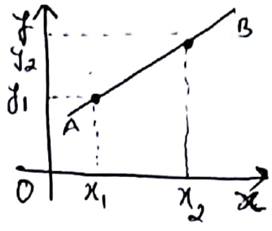
33.33*N= 33333.33

N=1000

By: Bishwas pokharel

Line Drawing Algorithms.

⇒ Eqn of st. line AB is,



$$y = mx + c \quad \text{—— (a)}$$

where,

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

or, $$m = \frac{\Delta y}{\Delta x} \quad \text{—— (b)}$$

Case (a), If,

$|m| < 1$ then,

$\Rightarrow \left|\frac{\Delta y}{\Delta x}\right| < 1$

$\Rightarrow |\Delta y| < |\Delta x| \quad \text{—— (c)}$

Results: Δx can be set proportional to small horizontal deflection voltage and then corresponding vertical deflection is then set proportional to Δy.

Meaning: Sampling is done at unit x-intervals and find corresponding y-values as,

$$\Delta y = m\Delta x \quad \text{—— (d)}$$

Case (b), If,

$|m| > 1$ then,

$\Rightarrow \frac{|\Delta y|}{|\Delta x|} > 1,$

$\Rightarrow |\Delta y| > |\Delta x| \quad \text{—— (e)}$

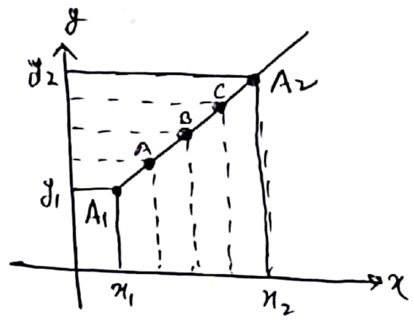
m

Results: Δy can be set proportional to small vertical deflection voltage with the corresponding horizontal deflection voltage set proportional to Δx.

Meaning: Sampling is done at unit y-intervals and find x-values as, $(\Delta x = \Delta y / m)$ —— (f)

DDA Algorithm:

The digital differential analyzer (DDA) is a scan conversion line algorithm based on calculating either Δx or Δy.



In such algorithm, two end points (x_1, y_1) and (x_2, y_2) is known. The task is to find all intermediate (discrete point) A, B, C between these given points.

from given fig slope of line A_1A_2,

$$m_{A_1A_2} = \left(\frac{y_2 - y_1}{x_2 - x_1}\right) \quad \text{—— (a)}$$

In general, if $y_2 = y_{k+1}$ then,
$y_1 = y_k$

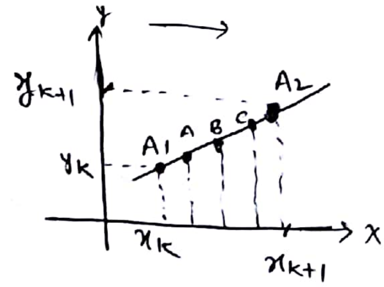
||ly. if, $x_2 = x_{k+1}$ then,
$x_1 = x_k$ where $k = 1, 2, 3 \dots$

from (a), in general,

$$m = \left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right) \quad \text{—— (b)}$$

Case (a) +ve slope

(1) Left to Right



Given two points A_1A_2 with A_1 as starting point & A_2 as ending point.

(i) $m > 1$,

$\frac{\Delta y}{\Delta x} > 1$

or, $\Delta y > \Delta x$, sampling must be done at y unit interval & calculate next x value.

So, $\boxed{y_{k+1} = y_k + 1}$ —— eq^n A

$\Rightarrow$ Here, y_k is initial y-value of A_1. If unit interval is increased the cordinate of A in Y- is added by 1 in y_k.

ie, ~~if $y_k = 2$,~~
if $A_1(2,3)$ then,
$A(x_{k+1}, 3+1)$

and from (b),

$$m = \left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right)$$

or, $$m = \left(\frac{\cancel{y_k} + 1 - \cancel{y_k}}{x_{k+1} - x_k}\right)$$

or, $$m = \frac{1}{x_{k+1} - x_k}$$

or, $$x_{(k+1)} - x_{(k)} = 1/m$$

or, $$\boxed{x_{k+1} = x_k + 1/m} \text{ —— eq}^n \text{ B}$$

Note: If $A_1 = (x_k, y_k)$ then, the cordinate of $A = (x_k + 1/m, y_k + 1)$.
ex: If $A_1 = (2,3)$ then,
$A = (2 + 1/m, 3+1) = (2 + 1/m, 4)$

(ii) $m \leq 1$,

$\frac{\Delta y}{\Delta x} \leq 1$

or, $\frac{\Delta y}{\Delta x} \leq 1 \Rightarrow \Delta y \leq \Delta x$

Sampling at $\Delta x = 1$ & find y successive value.

from (b), $\boxed{x_{k+1} = x_k + 1}$ —— eq^n C

$$m = \left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right) = \left(\frac{y_{k+1} - y_k}{\cancel{x_k} + 1 - \cancel{x_k}}\right)$$

or, $m = y_{k+1} - y_k$

$$\boxed{\text{or, } (y_{k+1} = y_k + m)} \text{ —— eq}^n \text{ D}$$

Note If $A_1 = (x_k, y_k)$ then, Cordinate of $A = (x_k + 1, y_k + m)$.
ex: If $A_1 = (2,3)$ then,
$A = (2+1, 3+m) = (3, 3+m)$.

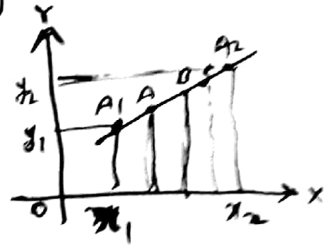
(2) Right to left

A_2 as starting point & A_1 as ending point.

(i) $m > 1$,

$\frac{\Delta y}{\Delta x} > 1$

$\Delta y > \Delta x$, But value of y decreases as move from R to L so, sampling at unit interval with -ve one i.e,



$$\boxed{y_{k+1} = y_k - 1} \quad \text{—— (E)}$$

from (b),

$$m = \left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right) = \left(\frac{\cancel{y_k} - 1 - \cancel{y_k}}{x_{k+1} - x_k}\right) = \left(\frac{-1}{x_{k+1} - x_k}\right)$$

$$\boxed{\text{or, } x_{k+1} = x_k - 1/m} \quad \text{—— (F)}$$

(ii) $m \leq 1$,

$\frac{\Delta y}{\Delta x} \leq 1$

$\Delta y \leq \Delta x$, Sampling at x interval, with unit value & find successive y-value.

$$\boxed{x_{k+1} = x_k - 1} \quad \text{—— (G)}$$ when move from Right to Left.

from (b),

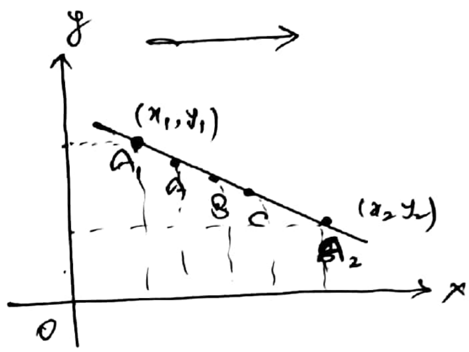
$$m = \left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right) = \left(\frac{y_{k+1} - y_k}{\cancel{x_k} - 1 - \cancel{x_k}}\right)$$

or, $y_{k+1} - y_k = -m$

$$\text{or, } \boxed{y_{k+1} = y_k - m} \quad \text{—— (H)}$$

Case (b) -ve slope

(1) Left to Right point.



Given, two points A_1, A_2 with A_1 as starting point & A_2 as the ending point.

(i) $m > 1$,

$$\frac{\Delta y}{\Delta x} > 1$$

or, $\Delta y > \Delta x$

Sampling must be done at y-intervals & Calculate next x-value.

$$\boxed{y_{k+1} = y_k - 1} \quad \text{—— (I)}$$

and from (b).

$$m = \left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right)$$

or, $m = \cancel{\dfrac{y_k + y_k - 1}{}} \left(\dfrac{y_k - 1 - y_k}{x_{k+1} - x_k}\right)$

or, $m = \dfrac{-1}{x_{k+1} - x_k}$

or, $x_{k+1} - x_k = -1/m$

$$\boxed{\text{or, } \left(x_{k+1} = x_k - 1/m\right)} \quad \text{—— (J)}$$

(ii) $m \le 1$

$$\frac{\Delta y}{\Delta x} \le 1$$

$\Delta y \le \Delta x$

Sampling must be done at X-unit interval & find successive Y-value.

$$\boxed{x_{k+1} = x_k + 1} \quad \text{(K)}$$

from (b),

$$m = \left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right)$$

$$= \frac{y_{k+1} - y_k}{\cancel{x_k} + 1 - \cancel{x_k}}$$

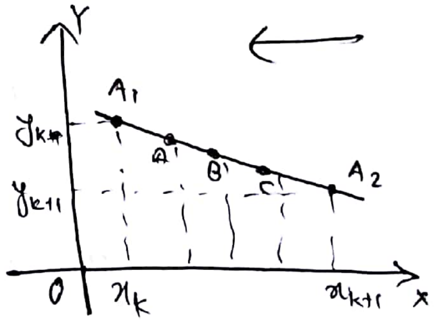
$$= y_{k+1} - y_k$$

or, $m = y_{k+1} - y_k$

$$\boxed{\text{or } y_{k+1} = y_k + m} \quad \text{(L)}$$

(2) Right end to left end point.

A_2 as starting point and
A_1 as ending point.



(i) $m > 1$.

$$\frac{\Delta y}{\Delta x} > 1$$

$$\frac{\Delta y}{\Delta x} > 1 \Rightarrow \Delta y > \Delta x$$

Sampling must be done at y-intervals with unit value and find successive X-values.

$$\boxed{y_{k+1} = y_k + 1} \quad \text{—— (M)}$$

from (b),

$$m = \left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right)$$

$$= \left(\frac{\cancel{y_k} + 1 - \cancel{y_k}}{x_{k+1} - x_k}\right)$$

$$= \frac{1}{x_{k+1} - x_k}$$

$$\text{or, } \boxed{x_{k+1} = x_k + 1/m} \quad \text{—— (N)}$$

(ii) $m \leq 1$.

$\frac{\Delta y}{\Delta x} \leq 1$

$\Delta y \leq \Delta x$ Sampling must be done on x-interval with unit value and find successive next y-values.

from (b),

$$m = \left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right) \qquad \boxed{\because x_{k+1} = x_k + 1} \text{ —— (O)}$$

$$\text{or, } m = \frac{y_{k+1} - y_k}{\cancel{x_k} + 1 - \cancel{x_k}}$$

$$\text{or, } m = y_{k+1} - y_k$$

$$\boxed{\therefore y_{k+1} = y_k + m} \quad \text{—— (P)}$$

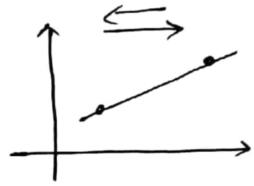
Example: Plot a st. line between (0,0) & (4,5).

Solution: ⇒

$$m = \left(\frac{5-0}{4-0}\right)$$

$$= 5/4 \quad \left(\because m = 5/4 \Rightarrow 1/m = 4/5 = 0.8\right)$$

$\therefore$ m is +ve and greater than zero.



⇒ Take any eqⁿ either from left to Right or, Right to Left.

⇒ Take left to Right then,

y → unit intervals

x → $x_k + 1/m$

$$\therefore \boxed{(x_{k+1}, y_{k+1}) = (x_k + 1/m,\ y_k + 1)}$$

x	y	x-plot	y-plot	(x,y)
0	0	0	0	(0,0) ⇒ starting
0 + (0.8) = 0.8	(0+1)	0.8 ≃ 1	1	(1,1)
0.8 + 0.8 = 1.6	1+1	1.6 ≃ 2	2	(2,2)
1.6 + 0.8 = 2.4	2+1	2.4 ≃ 2	3	(2,3)
2.4 + 0.8 = 3.2	3+1	3.2 ≃ 3	4	(3,4)
3.2 + 0.8 = 4	4+1	4	5	(4,5) → Ending

Stop.

By: Bishwas Pokharel

Plot the st. line between $(0,0)$ & $(5,4)$

Here,

$$\text{Slope}(m) = \frac{(4-0)}{(5-0)}$$
$$= 4/5$$
$$= 0.8 < 1. \quad \left(\because m = \frac{\Delta y}{\Delta x} < 1 \text{ or, } \Delta y < \Delta x\right)$$

So,

$$\text{eq}^n \text{ is, } (x_{k+1}, y_{k+1}) = (x_k + 1, y_k + m)$$

x	y	x-plot	y-plot	(x-plot, y plot)
0	0	0	0	(0, 0) ⇒ Starting
1	0+0.8=0.8	1	≃ 1	(1, 1)
2	0.8+0.8=1.6	2	≃ 2	(2, 2)
3	1.6+0.8=2.4	3	≃ 2	(3, 2)
4	2.4+0.8=3.2	4	≃ 3	(4, 3)
5	3.2+0.8=4	5	4	(5.4) ⇒ Ending

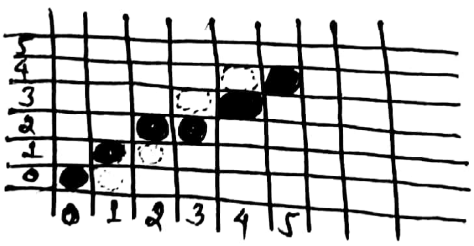


fig (A)

Scanned by CamScanner

DDA algorithm:

- Define the nodes, i.e. end points in form of (x1, y1) and (x2, y2).
- Calculate the distance between the two end points vertically and horizontally, i.e. dx=|x1-x2| and dy =|y1-y2|.
- Define new variable name 'pixel', and compare dx and dy values,
  if dx > dy then
  pixel=dx
  else
  pixel =dy.
- dx=dx/pixel
  and dy = dy/pixel
- x=x1;
  y=y1;
- while (i<=pixel) compute the pixel and plot the pixel with x=x+dx and y=y+dy.

DDA→PROBLEMS & IMPROVEMENT

- PROBLEMS
  - Accumulations of round off error in successive additions of the floating point increment– this causes the calculated pixel drift away from the true line path for long line segments→ **ALIASING EFFECT**
  - Rounding operations and floating-point arithmetic operations are still time consuming

- IMPROVEMENT
  - Can be removed by making increments m and 1/m into integer and fractional parts so that all calculations are reduced to integer operations

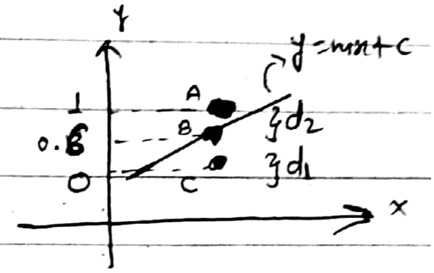
By: Bishwas Pokharel.

Basic Concept:

If you get 0.6 in Calculation, you may confuse whether to take the integer value 0 or 1. In Bresenham's algorithm floating point is convert into nearest integer by such concept,

d_1 = distance between calculated value and 0.

d_2 = distance between calculated value and 1.



⇒ if d_1 is greater than d_2 then it implies point B is far from C and nearest to the A. so it takes 1. ie,

$(d_1 - d_2)$ = +ve then it takes 1. $(y_K + 1)$

else,

$(d_1 - d_2)$ = -ve then it takes 0. (y_K)

⇒ Bresenham's Algorithm

Case (a) +ve slope

(i) $|m| < 1$

$\Rightarrow \left|\frac{\Delta y}{\Delta x}\right| < 1$

$\Rightarrow \Delta y < \Delta x$

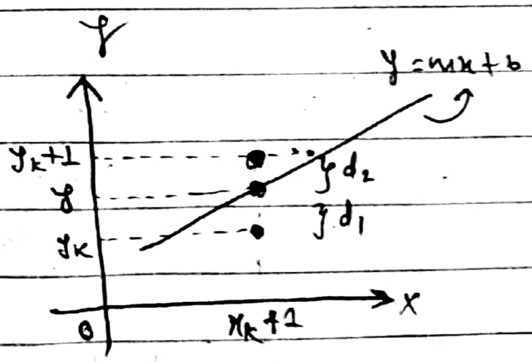


fig (a)

we know, sampling is to be done at x-axis with unit intervals.

so,

$$x_{K+1} = (x_K + 1) \quad \text{— (a)}$$

from fig (a),

$$d_1 = y - y_K \quad \text{——— (b)}$$

$$d_2 = y_K + 1 - y \quad \text{——— (c)}$$

Now,

$$(d_1 - d_2) = (y - y_K) - (y_K + 1 - y)$$

$$= y - y_K - y_K - 1 + y$$

$$= 2y - 2y_K - 1$$

$$= 2y - 2y_K - 1 \quad \text{——— (d)}$$

$\because$ $y = mx + c$, at $x = (x_K + 1)$

$$y = m(x_K + 1) + c \quad \text{——— (e)}$$

from (d) and (e),

$$(d_1 - d_2) = 2[m(x_K + 1) + c] - 2y_K - 1$$

$$= 2m(x_K + 1) + 2c - 2y_K - 1$$

$$= 2m(x_K + 1) - 2y_K + (2c - 1)$$

$$= 2mx_K + 2m - 2y_K + (2c - 1)$$

$$= 2mx_K - 2y_K + (2m + 2c - 1) \quad \text{——— (f)}$$

$\because$ m is a slope which value may be in floating point so to avoid floating point calculation m is replace by $\frac{\Delta y}{\Delta x}$ which is the difference of an integers.

from (f)

$$(d_1 - d_2) = 2\left(\frac{\Delta y}{\Delta x}\right)x_K - 2y_K + \left(2\frac{\Delta y}{\Delta x} + 2c - 1\right)$$

or, $(d_1-d_2) = (2x_k \cdot \Delta y - 2\Delta x \cdot y_k + 2\Delta y + 2\Delta x c - \Delta x) / \Delta x$

or, $(d_1-d_2)\Delta x = 2x_k\Delta y - 2\Delta x y_k + (2\Delta y + 2\Delta x c - \Delta x)$

or $(d_1-d_2)\Delta x = 2x_k\Delta y - 2\Delta x y_k + (2\Delta y + 2\Delta x \cdot c - \Delta x)$

$\because y = mx + c$

or, $c = y - mx$

or, $c = y - \left(\frac{\Delta y}{\Delta x}\right)x$

or, $(d_1-d_2)\Delta x = 2x_k\Delta y - 2\Delta x y_k + \left(2\Delta y + 2\Delta x\left[y - \frac{\Delta y}{\Delta x}x\right] - \Delta x\right)$

or, $(d_1-d_2)\Delta x = (2x_k\Delta y - 2\Delta x y_k) + (2\Delta y + 2\Delta x y - 2\Delta y x - \Delta x)$

$\because P_k = (d_1-d_2)\Delta x$, let then,

$$P_k = (2x_k\Delta y - 2\Delta x y_k) + (2\Delta y + 2\Delta x y - 2\Delta y x - \Delta x) \quad \text{--- (g)}$$

<u>at initial point (x_k, y_k),</u>

from (g)

$P_k = (\cancel{2x_k\Delta y} - \cancel{2\Delta x y_k}) + (2\Delta y + \cancel{2\Delta x y_k} - \cancel{2\Delta y x_k} - \Delta x)$

$$\boxed{P_0 = (2\Delta y - \Delta x) \quad \text{--- (h)}}$$ $\because k=0$

from (g)

$$\boxed{P_k = (2x_k\Delta y - 2\Delta x y_k) + C_0} \quad \text{--- (i)}$$

let, $C_0 = 2\Delta y + 2\Delta x y - 2\Delta y x - \Delta x$

$\because P_k = (d_1-d_2)\Delta x$ As we know $\Delta x > 0$ the sign of (d_1-d_2) determine the sign of P_k.

⇒ P_K is +ve,
⇒ $(d_1 - d_2)$ is +ve
⇒ $d_1 > d_2$
⇒ point is nearest to $(y_K + 1)$

ie,
$$y_{K+1} = (y_K + 1)$$

⇒ P_K is -ve
⇒ $(d_1 - d_2)$ is -ve
⇒ $d_2 > d_1$
⇒ point is nearest to (y_K)

ie,
$$y_{K+1} = y_K$$

Since, Cordinates changes along the line Occur in unit steps either on x or y dirn. so at (K+1) steps we also have to determine the sign of P_{K+1} so, to calculate P_{K+1} we have,

from (i)

$$P_K = 2x_K \Delta y - 2\Delta x\, y_K + C_0$$
$$P_{K+1} = 2x_{K+1}\Delta y - 2\Delta x\, y_{K+1} + C_0$$

$$\therefore P_{K+1} - P_K = 2\Delta y(x_{K+1} - x_K) - 2\Delta x(y_{K+1} - y_K) \quad \because x_{K+1} = x_K + 1$$
$$= 2\Delta y(\cancel{x_K} + 1 - \cancel{x_K}) - 2\Delta x(y_{K+1} - y_K)$$
$$= 2\Delta y - 2\Delta x(y_{K+1} - y_K)$$

$$\therefore P_{K+1} = P_K + 2\Delta y - 2\Delta x(y_{K+1} - y_K) \quad \text{— (j)}$$

If P_K is +ve,
$$y_{K+1} = y_K + 1$$
or, $P_{K+1} = P_K + 2\Delta y - 2\Delta x(\cancel{y_K} + 1 - \cancel{y_K})$

$$\boxed{\text{or, } P_{K+1} = P_K + 2\Delta y - 2\Delta x}$$

If P_K is -ve
or, $y_{K+1} = y_K$
or, $P_{K+1} = P_K + 2\Delta y - 2\Delta x(y_K - y_K)$

$$\boxed{\text{or, } P_{K+1} = P_K + 2\Delta y}$$

Scanned by CamScanner

(Q) Digitize the line with endpoints (20,10) & (30,18)

Soln:

Here,

$$\Delta x = (30-20) = 10$$

$$\Delta y = (18-10) = 8$$

$$m = \frac{8}{10} = 0.8$$

Now,

$$P_0 = 2\Delta y - \Delta x$$
$$= 2(8) - 10$$
$$= 16 - 10$$
$$= 6$$

again, for two cases,

$$= 2\Delta y - 2\Delta x \qquad = 2\Delta y$$
$$= 2\times 8 - 2\times 10 \qquad = 2\times 8$$
$$= -4 \qquad = 16$$

plot $(x_0, y_0) = (20, 10)$

K	P_k	(x_{k+1}, y_{k+1})
0	$P_0 = 6$ (+ve)	(21, 11)
1	2 (+ve)	(22, 12)
2	−2 (−ve)	(23, 12)
3	14 (+ve)	(24, 13)
4	10 (+ve)	(25, 14)
5	6 (+ve)	(26, 15)
6	2 (+ve)	(27, 16)
7	−2 (−ve)	(28, 16)
8	14 (+ve)	(29, 17)
9	10 (+ve)	(30, 18)

[∵ $y_{k+1} = y_k + 1$ when P_k = +ve]

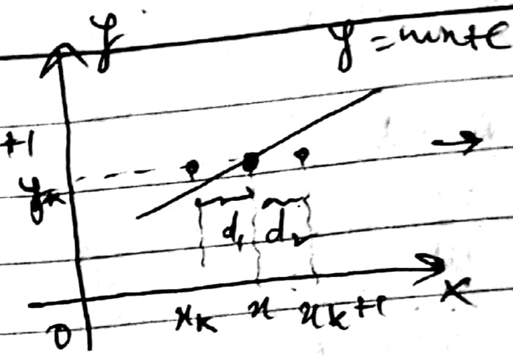
$P_{k+1} = 6 + (-4) = 2$

[∵ $y_{k+1} = y_k$ when P_k = −ve]

$P_{k+1} = -2 + 16 = 14$

(1) (b) $m > 1$

$\frac{\Delta y}{\Delta x} > 1$



$\Delta y > \Delta x$

Sampling at y-axis,
$y_{k+1} = y_k + 1$ and have to find
Succesive x-value.

$d_1 = x - x_k$ ——— (i)

$d_2 = (x_k + 1) - x$ ——— (ii)

$\therefore (d_1 - d_2) = [x - x_k] - [x_k + 1 - x]$

$= x - x_k - x_k - 1 + x$

$= 2x - 2x_k - 1$ ——— (iii)

we have,

$y = mx + c$

$x = \left(\frac{y-c}{m}\right)$ at $y_{k+1} = y_k + 1$

$= \left(\frac{y_k + 1 - c}{m}\right)$

from (iii)

$(d_1 - d_2) = 2\left(\left(\frac{y_k+1-c}{m}\right)\right) - 2x_k - 1$

$= \frac{2y_k}{m} + \frac{2}{m} - \frac{2c}{m} - 2x_k - 1$

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

Replace, $m = \frac{dy}{dx}$ & multiply by dy.

or, $\boxed{dy(d_1 - d_2) = 2 y_k \Delta x - 2 x_k \Delta y + k\,dy}$ —— (IV)

Now;

$$P_k = \underbrace{dy}_{+ve}(d_1 - d_2)$$

+ve : ($dy > 0$) so, P_k's sign depends upon the sign of $(d_1 - d_2)$.

from (IV),

$$P_k = 2 y_k \Delta x - 2 x_k \Delta y + k\,dy \quad \text{—— (V)}$$

$$P_{k+1} = 2 y_{k+1} \Delta x - 2 x_{k+1} \Delta y + k\,dy \quad \text{—— (VI)}$$

from (VI) − (V),

$$\boxed{P_{k+1} - P_k = 2(y_{k+1} - y_k)\Delta x - 2(x_{k+1} - x_k)\Delta y} \quad \text{—— (VII)}$$

⇒ If $P_k > 0$ then,

$d_1 > d_2$

⇒ $x_{k+1} = x_k + 1$

from (VII),

$$\boxed{P_{k+1} = P_k + 2\Delta x - 2\Delta y}$$

⇒ If $P_k < 0$ then,

$d_1 < d_2$

⇒ $x_{k+1} = x_k + 1$

from (VII)

$$\boxed{P_{k+1} = P_k + 2\Delta x}$$

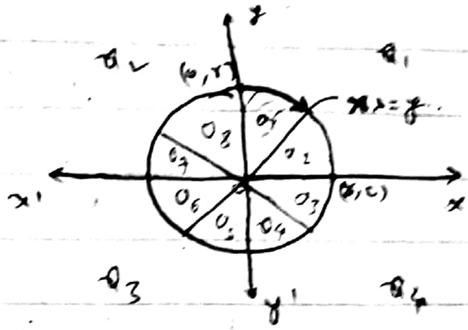
IIIly from (IV) you can get, at (x_0, y_0)

$$\boxed{P_0 = (2\Delta x - \Delta y)}$$

Scanned by CamScanner

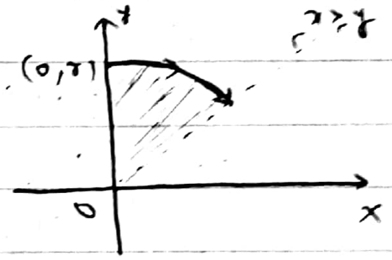
Mid point circle

Consider a circle with a radius (r), in xy plane. Since, radius is r the circle is symmetry along the xy-plane.

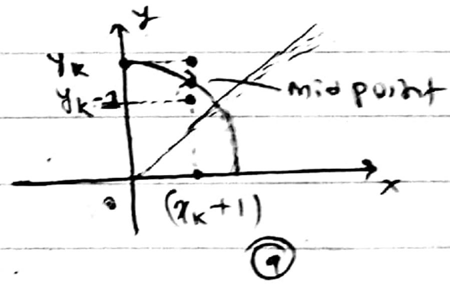


||ly, if in quadrant Q_1, the arc in divided into two equal halves then it's known as Octave. (O_1 and O_2). where each Cordinates in O_1 is inverse of O_2. so, our task is to find the Cordinates of first octave O_1, where end point of O_1 is $x \geq y$.

Now,



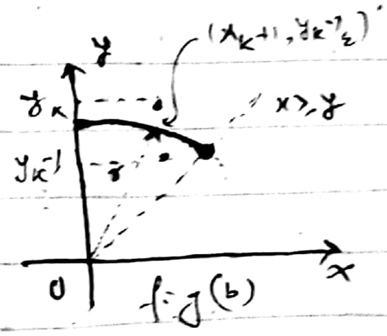
we are moving along x-axis so sampling is done at unit x-axis interval, with y as downwards so, next y Cordinates in either (y_k) or $(y_k - 1)$.



ie, (x_k+1, y_k) or (x_k+1, y_k-1)

but,

In fig (a) we can say circle passes almost from in between y_k and $(y_k - 1)$ so, midpoint is,



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

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

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

Scanned by CamScanner

we have the formula for circle is,

$P_K = x^2+y^2-r^2$, P_K is decision parameter

so,

or $P_K = (x_k+1)^2 + (y_k - 1/2)^2 - r^2$ —— (a)

for $(k+1)^{th}$ term,

$$P_{K+1} = (x_{k+1}+1)^2 + (y_{k+1} - 1/2)^2 - r^2 \text{ —— (b)}$$

$\therefore$ $P_{K+1} - P_K = (x_{k+1}+1)^2 + (y_{k+1} - 1/2)^2 \cancel{-r^2} - (x_k+1)^2 - (y_k - 1/2)^2 \cancel{+r^2}$

$= [(x_k+1)+1]^2 + [y_{k+1} - 1/2]^2 - (x_k+1)^2 - (y_k - 1/2)^2$

$= \cancel{(x_k+1)^2} + 2(x_k+1) + 1 + (y_{k+1})^2 - (y_{k+1}) + \cancel{1/4}$

$- \cancel{(x_k+1)^2} - (y_k)^2 + y_k - \cancel{1/4}$

$$\boxed{\therefore P_{k+1} - P_k = 2(x_k+1) + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k) + 1} \text{ —— (c)}$$

For initial decision parameter, we have to consider the starting point which is $(0, r) = (x_k, y_k)$

so from (a),

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

or $P_K = 1 + \cancel{r^2} - \frac{2}{2}r + 1/4 \cancel{-r^2}$

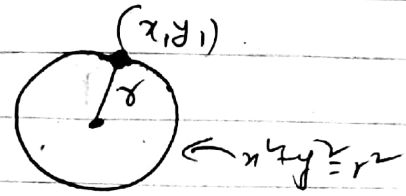
$$\boxed{\therefore P_K = 5/4 - r} \text{ —— (d)}$$

Scanned by CamScanner

we know,

if,

(a) $x_1^2 + y_1^2 = r^2$ (point x_1, y_1 lies on surface of circle)



(b) $x_1^2 + y_1^2 > r^2$

i.e, $(x_1^2 + y_1^2) - r^2 > 0$ (outside circle)

(c) $x_1^2 + y_1^2 < r^2$

i.e, $(x_1^2 + y_1^2) - r^2 < 0$ (inside circle)

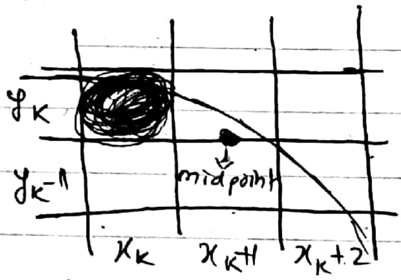
⇒ $||^{ly}$, if,

$P_K > 0$, the midpoint lies outside the circle and the pixel on the scanline $(y_k - 1)$ is closer to the circle boundary.

$\therefore$ $\underline{y_{k+1} = y_k - 1}$ and,

from (c),

$P_{k+1} = P_K + 2(x_k+1) + [(y_k - 1)^2 - y_k^2) - (y_k - 1 - y_k) + 1$



(x_k, y_k) is already plotted.

$$\boxed{\therefore P_{k+1} = P_K + 2(x_{k+1}) + 1 - 2y_{k+1}} \quad \text{— (e)}$$

⇒ If $P_K \leq 0$, the midpoint lies inside the circle so boundary lies near y_k i.e, $\underline{y_{k+1} = y_k}$

from (c),

$$\boxed{P_{k+1} = P_K + 2(x_{k+1}) + 1} \quad \text{— (f)}$$

Based on the first octants, we have to fill the remaining seven octants to complete the circle.

① Draw a circle with radius 8.

Sol^n:

$P_0 = 1 - r = 1 - 8 = -7$

K	(x_k, y_k)	P_k	(x_{k+1}, y_{k+1})	
0	(0,8)	-7	(1,8)	$\because P_{k+1} = -7 + 2(0+1) + 0 - 0 + 1$ $= -4$
1	(1,8)	-4	(2,8)	$\because P_{k+1} = -4 + 2(1+1) + 1$ $= 1$
2	(2,8)	1	(3,7)	$\because P_{k+1} = P_k + 2(x_k+1) + 1 - 2(y_k - 1)$ $= 1 + 2(2+1) + 1 - 2(8-1)$ $= 1 + 2(3) + 1 - 2(7)$ $= 1 + 6 + 1 - 14$ $= -6$
3	(3,7)	-6	(4,7)	$\because P_k = -6 + 2(x_k+1) + 1$ $= -6 + 2(3+1) + 1$ $= -6 + 8 + 1$ $= +3$
4	(4,7)	3	(5,6)	$\because P_k = 2$
5	(5,6)	2	(6,5)	$[\because 6 > 5]$ **so stop**

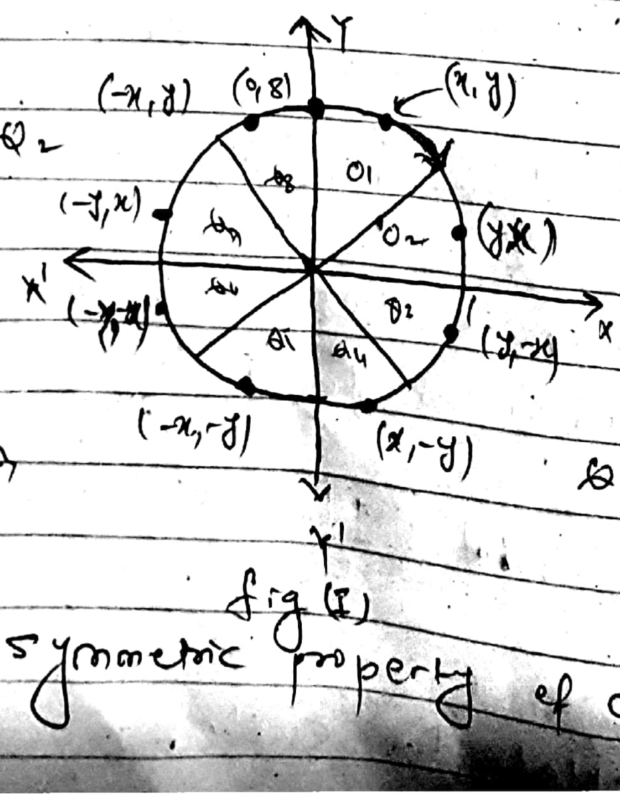


fig (I)
symmetric property of circle

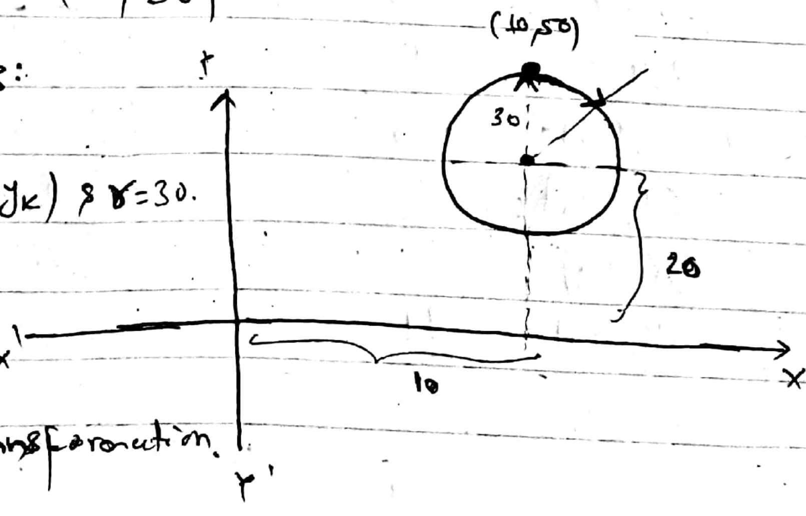
Q_1		Q_2		Q_3	
(0,8)	(x,y) in octant 1 i.e. O_1	(0,8)	$(-x,y)$ in octave 8	(0,8)	$(-x,-y)$ in octant
(1,8)		(-1,8)		(-1,-8)	
(2,8)		(-2,8)		(-2,-8)	
(3,7)		(-3,7)		(-3,-7)	
(4,7)		(-4,7)		(-4,-7)	
(5,6)		(-5,6)		(-5,-6)	
(6,5)	(y,x) in octant 2 i.e. O_2	(-6,5)	$(-y,x)$ in octave 7, O_7.	(-6,-5)	
(7,4)		(-7,4)		(-7,-4)	
(7,3)		(-7,3)		(-7,-3)	
(8,2)		(-8,2)		(-8,-2)	
(8,1)		(-8,1)		(-8,-1)	
(8,0)		(-8,0)		(-8,0)	

In this way all points are plotted to make a circle but on examination we ~~also~~ only need to plot for one octants.

Q2. Draw a circle with a radius 10.

Q3. Draw a circle with radius 30 and center (10,20)

Hints:



Option 1

Use (10,50) as (x_k, y_k) & $r=30$.

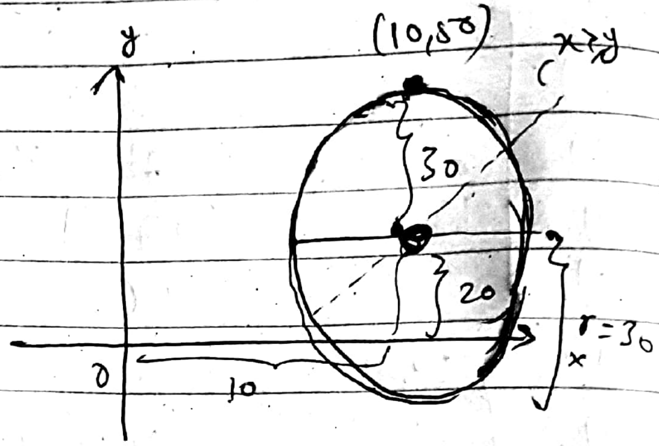
Option 2

Use center as (0,0) and add it later using concept of transformation.

① Draw a circle with a radius of 30 and center (10, 20).

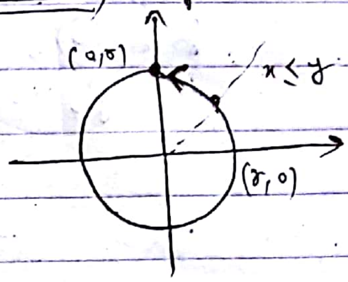
Here,

$P_0 = 1 - r$
$= 1 - 30$
$= -29$



K	(x_k, y_k)	P_k	(x_{k+1}, y_{k+1})
0	(10,50)	-29	(11,50)
1	(11,50)	-6	(12,50)
2	(12,50)	19	(13,49)
3	(13,49)	-52	(14,49)
4.	(14,49)	-23	(15,49)
5.	(15,49)	8	(16,48)
6.	(16,48)	-55	(17,48)
7	(17,48)	-20	(18,48)
8.	(18,48)	17	(19,47)
9.	(19,47)	-38	(20,47)
10.	(20,47)	3	(21,46)
11.	(21,46)	-46	(22,46)
12.	(22,46)	-1	(23,46)
13.	(23,46)	46	(24,45)
14.	(24,45)	5	(25,44)
15.	(25,44)	-32	(26,44)
16.	(26,44)	21	(27,43)
- - -	- - -	- - -	- - -
25	(35,37)	24	(36,36)

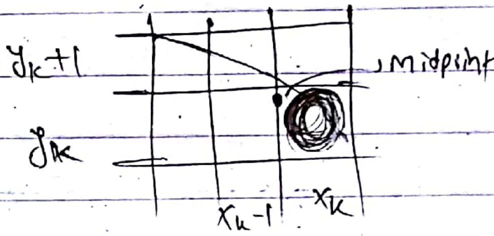
Midpoint circle with starting point (r, 0).



As we know,

moving from (r, 0) towards origin, y-value increases while x-value decreases, so,

$$m = \frac{dy \uparrow}{dx \downarrow} > 1$$



So, sampling done at y-axis with unit intervals & find successive x-value, whether next coordinate is (x_k, y_k+1) or (x_k-1, y_k+1).

Using midpoint formula,

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

Now, decision parameter p_k,

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

$$= \left(x_k - \tfrac{1}{2}\right)^2 + \left(y_k + 1\right)^2 - r^2 \quad \text{— (a)}$$

For $(k+1)^{th}$,

$$p_{k+1} = \left(x_{k+1} - \tfrac{1}{2}\right)^2 + \left(y_{k+1} + 1\right)^2 - r^2 \quad \text{— (b)}$$

(b) − (a),

$$p_{k+1} = \left[\left(x_{k+1}^2 - x_{k+1} + \tfrac{1}{4}\right) + \left\{\left(y_k + 1\right) + 1\right\}^2\right] - \left[\left(x_k^2 - x_k + \tfrac{1}{4}\right) + \left(y_k + 1\right)^2\right]$$

$$\Rightarrow \left(x_{k+1}^2 - x_k^2\right) - \left(x_{k+1} - x_k\right) + 2\left(y_k + 1\right) + 1$$

$$\boxed{\therefore\ p_{k+1} = 2\left(y_k + 1\right) + \left(x_{k+1}^2 - x_k^2\right) - \left(x_{k+1} - x_k\right) + 1 + p_k} \quad \text{— (c)}$$

Scanned by CamScanner

For initial decision parameter $(r, 0)$.

from ⓐ,

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

$$= \left(r - \frac{1}{2}\right)^2 + (0+1)^2 - r^2$$

$$= r^2 - r + \frac{1}{4} + 1 - r^2$$

$$= \frac{5}{4} - r$$

$$\boxed{P_k \simeq 1 - r} \text{ — ⓓ}$$

ⓐ If $P_k \geq 0$ then,

$$(x_{k+1} = x_k - 1)$$

so,

$$P_{k+1} = P_k + 2(y_k+1) + 1 + [\cancel{x_k^2} - 2x_k + 1 - \cancel{x_k}] - (\cancel{x_k} - 1 \cancel{x_k})$$

$$= P_k + 2(y_k+1) + 1 + [-2x_k + 2]$$

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

$$\boxed{P_{k+1} = P_k + 2\underbrace{(y_k+1)}_{\text{or, } 2y_{k+1}} + 1 - 2x_{k+1}} \quad \boxed{\because x_{k+1} = x_k - 1}$$

$[\because y_k + 1 = y_{k+1}]$

ⓑ If $P < 0$, then,

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

so,

$$\boxed{P_{k+1} = P_k + 2\underbrace{(y_k + 1)} + 1}$$

$\because 2(y_k + 1) \Rightarrow 2y_k + 2$

$= 2y_{k+1}$

or,

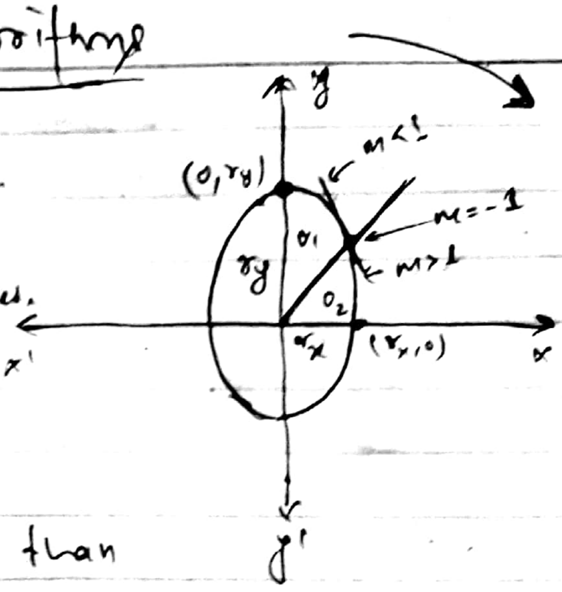
or, If $P \geq 0$ then,

$$P_{k+1} = P_k + 2y_{k+1} + 1 - 2x_{k+1}$$

If $P < 0$ then,

$$P_{k+1} = P_k + 2y_{k+1} + 1$$

Ellipse Drawing Algorithms



Ellipse has two radius r_x and r_y.
In octant O_1, slope is less than 1 because x-increases & y-decreases when moving from $(0, r_y)$ to $(r_x, 0)$

||'ly,
In octants O_2, slope is greater than 1 because x-decreases & y-increases when moving from $(r_x, 0)$ to $(0, r_y)$.

We know,

In octant 1,	In Octant 2.
$m<1$	$m>1$
x-unit interval	y-unit interval
$y = ?$	$x = ?$
i.e, (x_k+1, y_k) (x_{k+1}, y_k-1)	i.e, (x_k, y_k-1) (x_{k+1}, y_k-1)

So, where to stop octant 1 and start octant 2, !?
It's the point where $m=-1$. So, we have to find the stopping criteria,

the eq^n of ellipse,

$$\frac{x^2}{r_x^2} + \frac{y^2}{r_y^2} = 1$$

$$\text{or, } x^2 r_y^2 + y^2 r_x^2 = r_x^2 r_y^2$$

$$\boxed{\text{or, } x^2 r_y^2 + y^2 r_x^2 - r_x^2 r_y^2 = 0} \text{ —— (a)}$$

$$\left(\because m = \frac{dy}{dx} \right)$$

$$\text{or, } 2x\, r_y^2 + 2y\, r_x^2 \frac{dy}{dx} = 0$$

$$\text{or, } \boxed{\frac{dy}{dx} = \frac{-2x r_y^2}{2y r_x^2}}$$

Scanned by CamScanner

$$\frac{dy}{dx} = m = -1 \text{ so,}$$

or, $-1 = -\frac{2 r_y^2 x}{2 r_x^2 y}$

or, $2 r_y^2 x = 2 r_x^2 y$

So,

we move out of region 1 whenever,

$$\boxed{2 r_y^2 x \geq 2 r_x^2 y} \quad \text{— (b)}$$

In octant 1,

$m < 1$ so,

(x_k+1, y_k) (x_k+1, y_k-1)

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

applying value in eq^n (a) then,

$$\boxed{\therefore \; P_{1k} = (x_k+1)^2 r_y^2 + \left(y_k - \tfrac{1}{2}\right)^2 r_x^2 - r_x^2 r_y^2} \quad \text{— (c)}$$

$$\therefore \; P_{1k+1} = (\underbrace{x_{k+1}}_{(x_k+1)} + 1)^2 r_y^2 + \left(y_{k+1} - \tfrac{1}{2}\right)^2 r_x^2 - r_x^2 r_y^2$$

$$\boxed{P_{1k+1} = [(x_k+1)+1]^2 r_y^2 + \left(y_{k+1} - \tfrac{1}{2}\right)^2 r_x^2 - r_x^2 r_y^2} \quad \text{— (d)}$$

(d) $-$ (c),

$$P_{1k+1} - P_{1k} = r_y^2 + 2(x_k+1) r_y^2 + r_x^2 (y_{k+1} - y_k)^2 - r_x^2 (y_{k+1} - y_k)$$

$$\boxed{P_{1k+1} = P_{1k} + r_y^2 + 2(x_k+1) r_y^2 + r_x^2 (y_{k+1} - y_k)^2 - r_x^2 (y_{k+1} - y_k)} \quad \text{— (e)}$$

For, initial decision parameter, substitute $(0, r_y)$ in P_{1K}.

or, $P_{1K} = r_y^2 + (r_y - 1/2)^2 r_x^2 - r_x^2 r_y^2$

or $P_{1K} = \cancel{r_y^2} + \cancel{r_y^2 r_x^2} - r_y r_x^2 + 1/4 r_x^2 - \cancel{r_x^2 r_y^2}$

$$\boxed{\therefore \quad P_{1K} = \frac{r_x^2}{4} + r_y^2 - r_y r_x^2} \quad \text{— (f)}$$

If $P_{1K} \geq 0$

$\Rightarrow (x_k + 1, y_k - 1)$

~~else~~, If $P_{1K} < 0$

$\Rightarrow (x_k + 1, y_k)$

In Octant 2, $m > 1$ so,

$\cancel{(x_k, y_k)}$ $(x_k, y_k - 1)$ $(x_k + 1, y_k - 1)$

midpoint $= \left((x_k + 1/2), (y_k - 1)\right)$

applying value on eq^n (a) then,

$$\boxed{P_{2K} = (x_k + 1/2)^2 r_y^2 + (y_k - 1)^2 r_x^2 - r_x^2 r_y^2} \quad \text{— (g)}$$

$$\boxed{P_{2K+1} = (x_{k+1} + 1/2)^2 r_y^2 + ((y_k - 1) - 1)^2 r_x^2 - r_x^2 r_y^2} \quad \text{— (h)}$$

(h) − (g)

$$\boxed{P_{2K+1} = P_{2K} + r_x^2 + r_y^2 (x_{k+1} - x_k) + r_y^2 (x_{k+1}^2 - x_k^2) - 2 r_k^2 (y - 1)} \quad \text{— (i)}$$

when cond^n $2 r_y^2 x \geq 2 r_x^2 y$ becomes true i.e. the point for initial decision parameter in Octant 2.

so,

If $P_{2K} \geq 0$,

$(x_k, y_k - 1)$

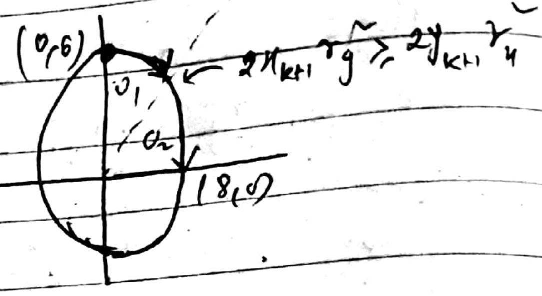
If $P_{2K} < 0$,

$(x_{k+1}, y_k - 1)$

① Draw an ellipse with $r_x = 8$ and $r_y = 6$.

Here,

For octant 1,



$$P_{10} = r_y^2 + \frac{r_x^2}{4} - r_y r_x^2$$

$$= -332 < 0$$

k	(x_k, y_k)	P_{1k}	(x_{k+1}, y_{k+1})	$2x_{k+1} r_y^2$	$2y_{k+1} r_x^2$
0	(0,6)	−332	(1,6) (−ve)	72 <	768
		$\boxed{P_{1k+1} = -224 < 0}$			
1	(1,6)	−224 (−ve)	(2,6)	144 <	768
		$\boxed{P_{1k+1} = -44 < 0}$			
2	(2,6)	−44	(3,6)	216 <	768
		$\boxed{P_{1k+1} = 208 > 0}$			
3	(3,6)	208	(4,5)	288 <	640
		$\boxed{P_{1k+1} = -108 < 0}$			
4	(4,5)	−108	(5,5)	360 <	640
		$\boxed{P_{1k+1} = 288 > 0}$			
5	(5,5)	288	(6,4)	432 <	512
		$\boxed{P_{1k+1} = 244 > 0}$			
6.	(6,4)	244	(7,3)	504 <	384

True

Enter on region 2 (O_2).

substitue (7,3) in $P_{2k} = (x_k + \frac{1}{2})^2 r_y^2 + (y_k - 1)^2 r_x^2 - r_x^2 r_y^2$

$\boxed{P_{20} = -23 < 0}$

k	(x_k, y_k)	P_{2k}	(x_{k+1}, y_{k+1})
7.	(7,3)	−23	(8,2)
		$\boxed{P_{2k+1} = 361 > 0}$	
8.	(8,2)	361	(8,1)
		$\boxed{P_{2k+1} = 297 > 0}$	
9.	(8,1)	297	(8,0) → Stop //

2-D Transformations:

Changing the position of an object

Some are:

- ⇒ translation
- ⇒ Shearing
- ⇒ Scaling
- ⇒ Reflection
- ⇒ Rotation

(a) Translation:

p(x, y) be any point before translation

If t_x and t_y are translation parameter then

$\Rightarrow x + t_x = x'$ (say)

$\Rightarrow y + t_x = y'$ (say)

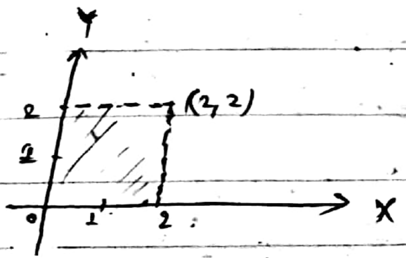
p'(x', y') be any point after translation.

In Matrix form,

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

Example: translate square (0,0) (2,0) (0,2) (2,2) with $t_x = 2$ and $t_y = 3$.

Soln:



Before translation

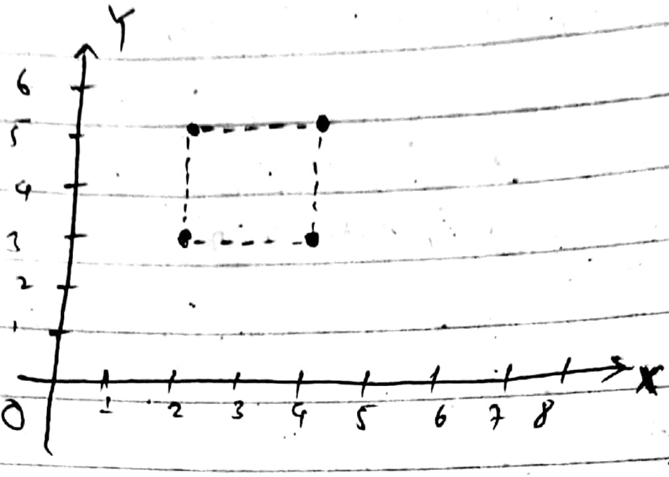
Now, $t_x = 2$, $t_y = 3$

$\Rightarrow (0,0) = (0+2, 0+3) = (2,3)$ ⇒ $(2,2) = (2+2, 2+3) = (4,5)$

$\Rightarrow (0,2) = (0+2, 2+3) = (2,5)$

$\Rightarrow (2,0) = (2+2, 0+3) = (4,3)$

After translation



(b) Rotation

Anti clockwise

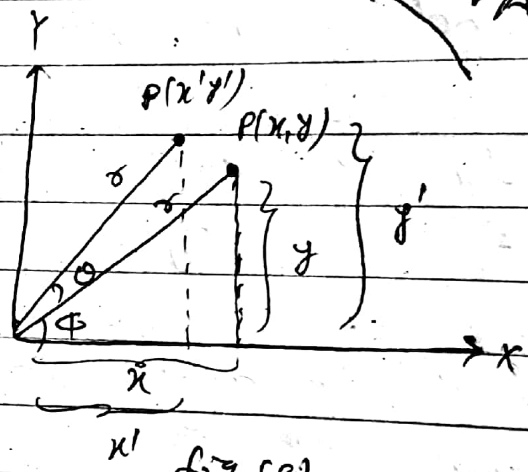


fig (a)

P(x,y) point before rotation and after rotation p(x,y) be p(x'y').

Now from fig (a).

$\Rightarrow \cos\phi = x/r$

$\therefore x = r\cos\phi$

$\Rightarrow \sin\phi = y/r$

$\therefore y = r\sin\phi$

New angle after rotation by θ. be $(\theta+\phi)$

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

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

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

$\therefore y' = r\sin(\theta+\phi)$. —— (b) we know formula,

$$\cos(A+B) = \cos A . \cos B - \sin A . \sin B$$
$$\sin(A+B) = \sin A . \cos B + \cos A . \sin B .$$

from (a).

$$x' = r\cos(\theta+\phi)$$
$$= r[\cos\theta . \cos\phi - \sin\theta . \sin\phi]$$
$$= r\cos\theta . \cos\phi - r\sin\theta . r\sin\phi$$
$$= (r\cos\phi)\cos\theta - (r\sin\phi)\sin\theta$$
$$= x\cos\theta - y\sin\theta$$

$$\boxed{\therefore \; x' = x\cos\theta - y\sin\theta}$$

from (b).

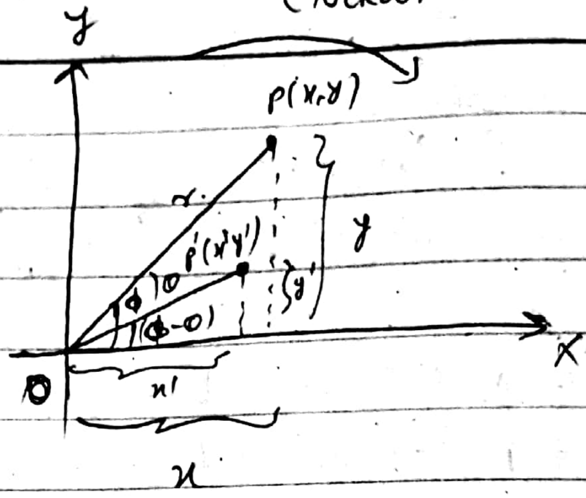
$$y' = r\sin(\theta+\phi)$$
$$= r[\sin\theta . \cos\phi + \cos\theta . \sin\phi]$$
$$= (r\cos\phi)\sin\theta + (r\sin\phi)\cos\theta$$
$$= x . \sin\theta + y\cos\theta$$

$$\boxed{\therefore \; y' = x\sin\theta + y\cos\theta}$$

In matrix form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta \\ +\sin\theta & \cos\theta \end{bmatrix}}_{\boxed{R(\theta)}} \begin{bmatrix} x \\ y \end{bmatrix}$$

Anticlockwise
θ is +ve .



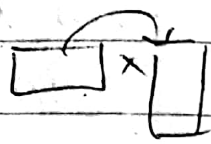
from fig:

$x = r\cos\phi$

$y = r\sin\phi$

Now,

After rotation by θ the new angle be $(\phi-\theta)$. with new point $P'(x'y')$.



Now,

$$x' = r\cos(\phi-\theta) \quad \text{—— (a)}$$

$$y' = r\sin(\phi-\theta) \quad \text{—— (b)}$$

we have formula,

$$\cos(A-B) = \cos(A)\cdot\cos B + \sin A\cdot\sin B.$$

$$\sin(A-B) = \sin A\cdot\cos B - \cos A\cdot\sin B.$$

from (a),

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

$$= r\cos\phi\cdot\cos\theta + r\sin\phi\sin\theta$$

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

$$\boxed{\therefore x' = x\cos\theta + y\sin\theta}$$

from (b)

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

$$= r\sin\phi\cos\theta - r\cos\phi\sin\theta$$

$$\boxed{\therefore y' = y\cos\theta - x\sin\theta}$$

<u>In matrix form:</u>

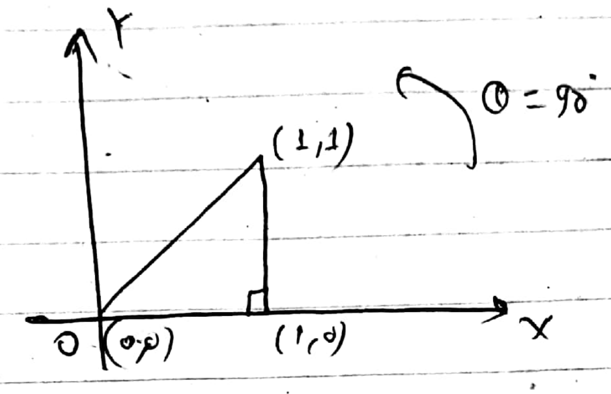
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & +\sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$\Downarrow$

$R(-\theta) \Rightarrow$ clockw

① Rotate Δ (0,0) (1,0) (1,1) with $\theta = 90°$ (Anticlockwise

sol^n :-



we know,

$$\cos 90° = 0$$
$$\sin 90° = 1$$

from formula of Rotation in Anticlockwise

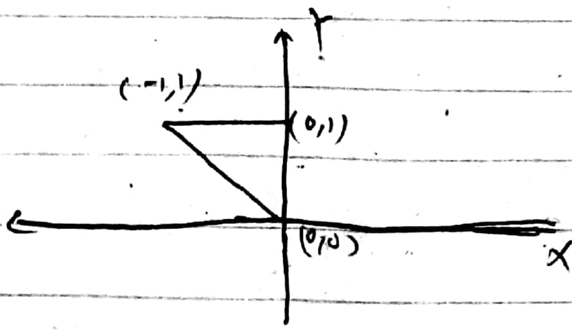
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$\therefore$ (0,0) $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

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

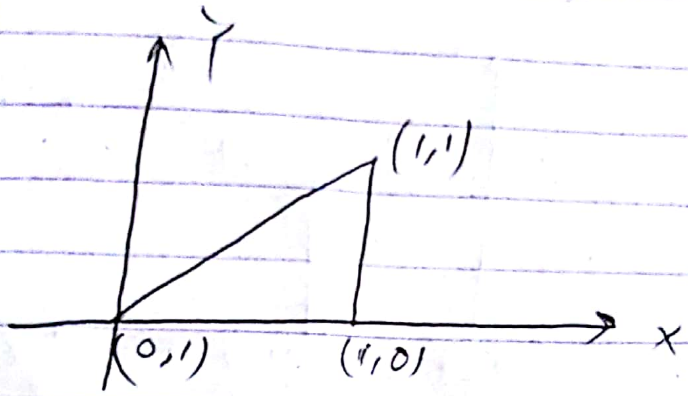
(1,1) $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$

P/A



① Rotate Δ (0,0) (1,0) (1,1) with (θ = 90) clockwise.

Sol:-



we have,

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

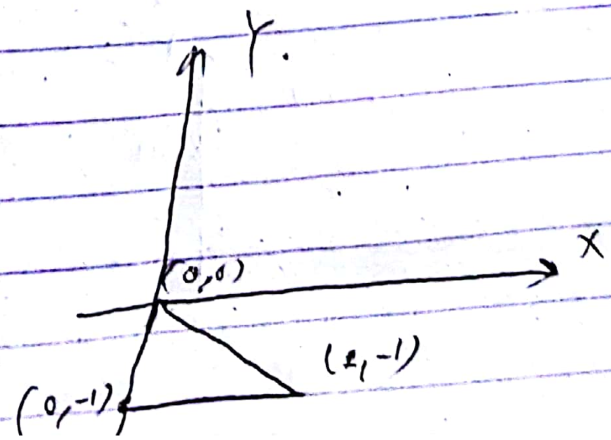
⇒ (0,0) $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x=0 \\ y=0 \end{bmatrix}$

$= \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

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

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

plot



Scanned by CamScanner

(c) Scaling (Re-size the Object)

let, S_x and S_y are scaling factor.

(a) If S_x and S_y are in between 0 and 1

- point is closer to origin
- size decreases.

(b) if S_x and S_y are > 1

- point away from origin
- size increases.

(c) If S_x and S_y are equal.

- scaling done uniformly.

Suppose, $P = (x, y)$ are point before scaling

If S_x and S_y are scaling factors, then

$P' = (x' y')$ are point after scaling.

$\therefore$ $x' = x \cdot S_x$

$y' = y \cdot S_y$

In matrix form:

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

Example: Scale square (0,0), (2,0), (0,2), (2,2) with $s_x = 2$, $s_y = 3$.

Sol^n: ↓

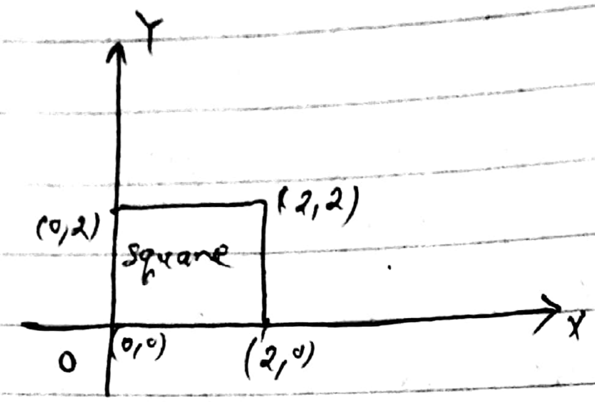


fig: Before scaling.

Now, using formula,

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

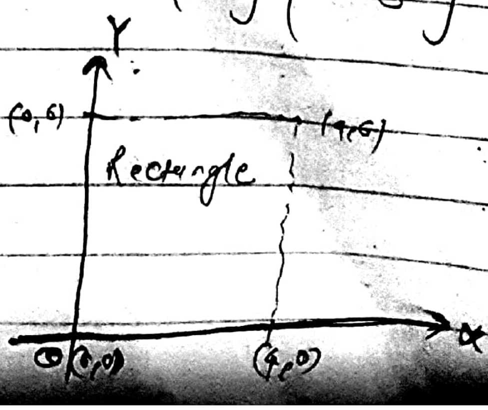
$\Rightarrow$ (0,0)
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$\Rightarrow$ (2,0)
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$$

$\Rightarrow$ (0,2)
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 0 \\ 6 \end{bmatrix}$$

$\Rightarrow$ (2,2)
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

Plot



Example: Square (0,0), (2,0), (0,2), (2,2)

(a) $S_x = 0.5$, $S_y = 0.5$

Soln:

$x' = x \cdot S_x$

$y' = y \cdot S_y$.

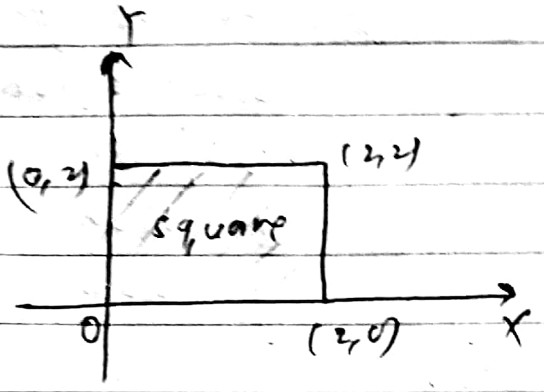


fig: Before scaling.

(0,0)

$x' = 0 \cdot 0.5 = 0$

$y' = 0 \cdot 0.5 = 0$.

$(x', y') = (0, 0)$

(2,0)

$x' = 2 \cdot 0.5 = 1$

$y' = 0 \cdot 0.5 = 0$

(0,2)

$x' = 0 \cdot (0.5) = 0$

$y' = 2 \cdot (0.5) = 1$

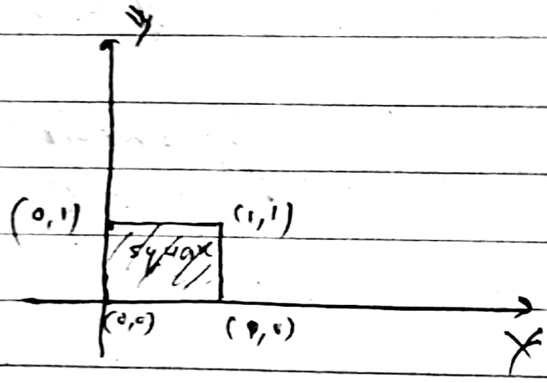


fig: After Scaling:

(2,2)

$x' = 2 \times 0.5 = 1$

$y' = 2 \times 0.5 = 1$

So, Translation $\rightarrow$ $P' = T + P$

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

Rotation $\rightarrow$ $P' = R(\theta) \cdot P$

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

Scaling $\rightarrow$ $P' = S \cdot P$

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

General expression:

$$P' = M_1 P + M_2$$

where

P' = new cordinates

M_1 = multiplicative matrix

M_2 = additive matrix

P = Old cordinates

To express any two dimensional transformation as matrix multiplication, we represent each Cartesian Cordinate position (x, y) with homogeneous cordinates triple (x_h, y_h, h) where,

$x = x_h / h$ and, $x_h = hx$

$y = y_h / h$ $y_h = hy$

for simplication $h = 1$.

For example:

(1) $(2,3) \underset{h=2}{\Rightarrow} (2\times 2, 3\times 2, 2)$
$= (4, 6, 2)$

(2) $(4,6,2) \Rightarrow (4/2, 6/2)$
$(2,3)$

Using Concept of homogenous matrix:

(a) Translation,

$$x' = t_x + x$$
$$y' = t_y + y.$$

In matrix form

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

In homogenous form

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

(b) Rotation:

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

In homogenous form,

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

Scanned by CamScanner

(c) Scaling.

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

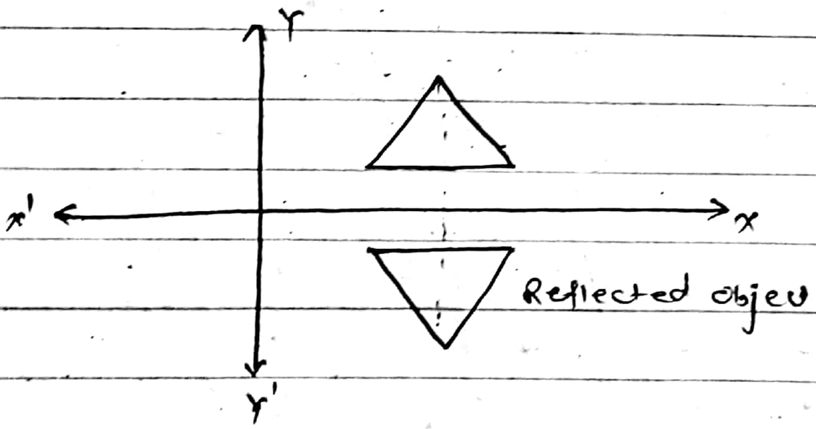
$\Downarrow$

$$\begin{bmatrix} x' \\ y' \\ h \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & h \end{bmatrix} \begin{bmatrix} x \\ y \\ h \end{bmatrix} \quad [\because h = 1]$$

x——x——x——x——x

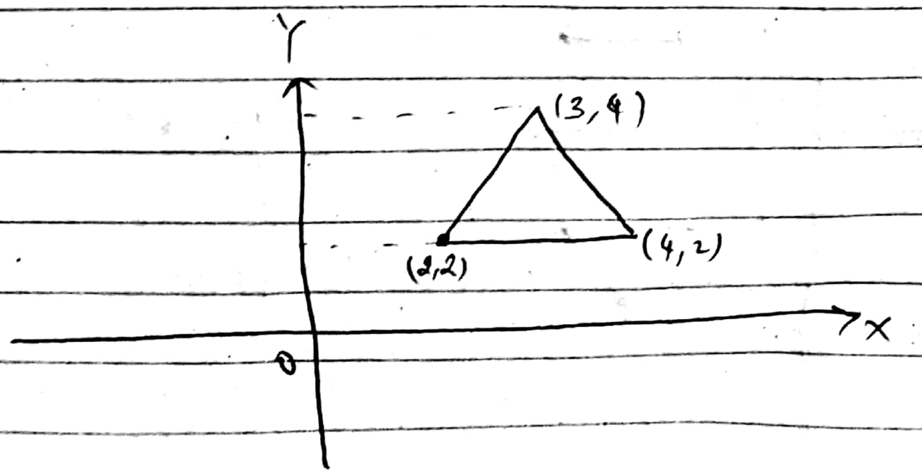
(d) Reflection (180°)

(a) X-axis.



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

Examples



Now, we have formula for reflection along X-axis.

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

$\Rightarrow$ (2, 2)

$$\begin{bmatrix} x' \\ y' \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \\ 1 \end{bmatrix} \Rightarrow (2, -2)$$

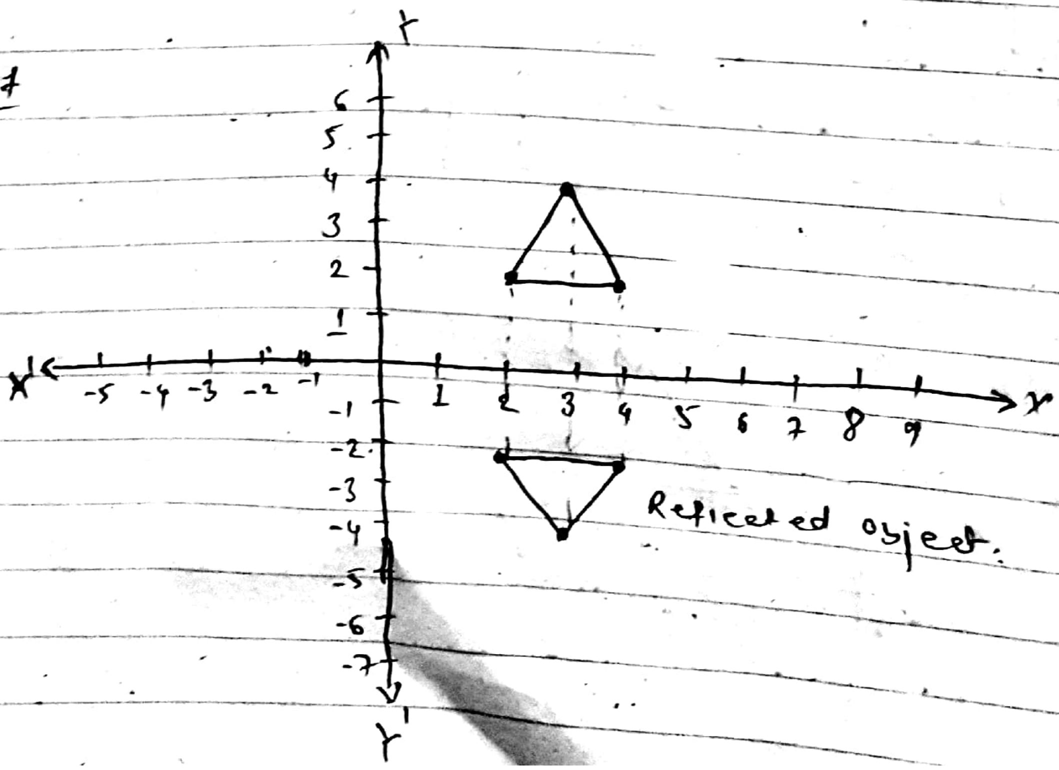
$\Rightarrow$ (4, 2)

$$\begin{bmatrix} x' \\ y' \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ -2 \\ 1 \end{bmatrix} \Rightarrow (4, -2)$$

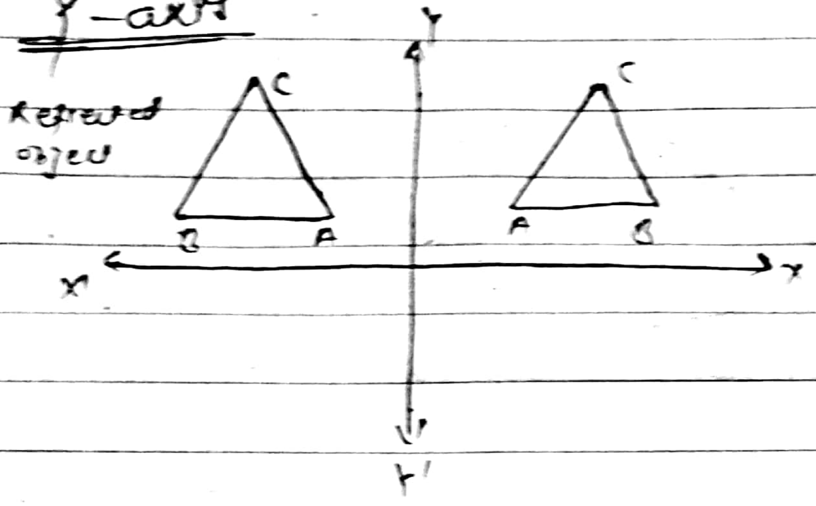
$\Rightarrow$ (3, 4)

$$\begin{bmatrix} x' \\ y' \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ -4 \\ 1 \end{bmatrix} \Rightarrow (3, -4)$$

Plot



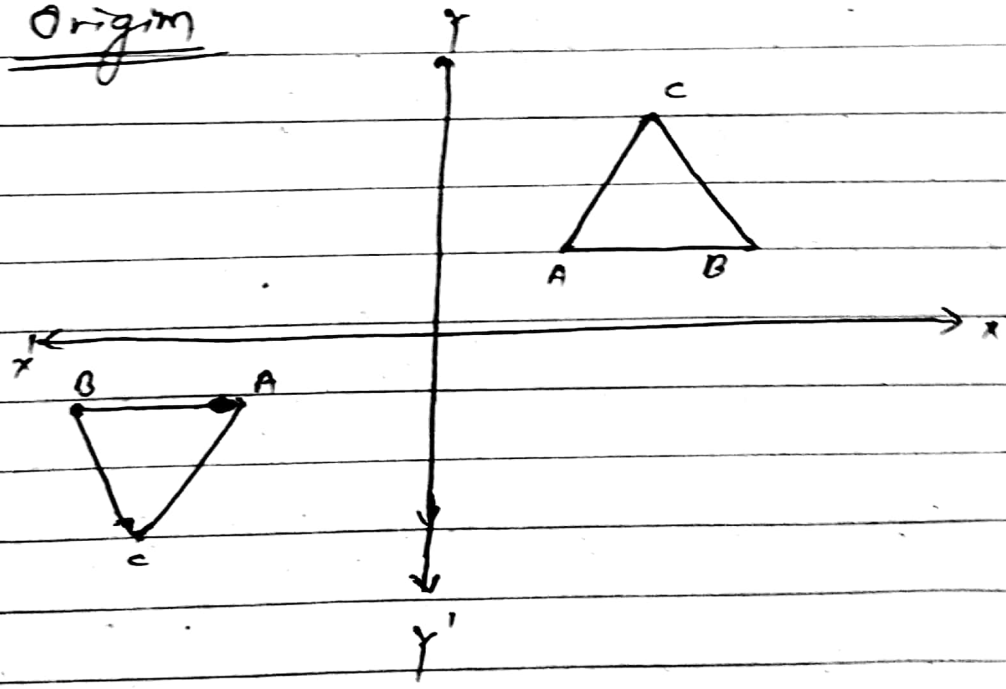
② Y-axis



formula,

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

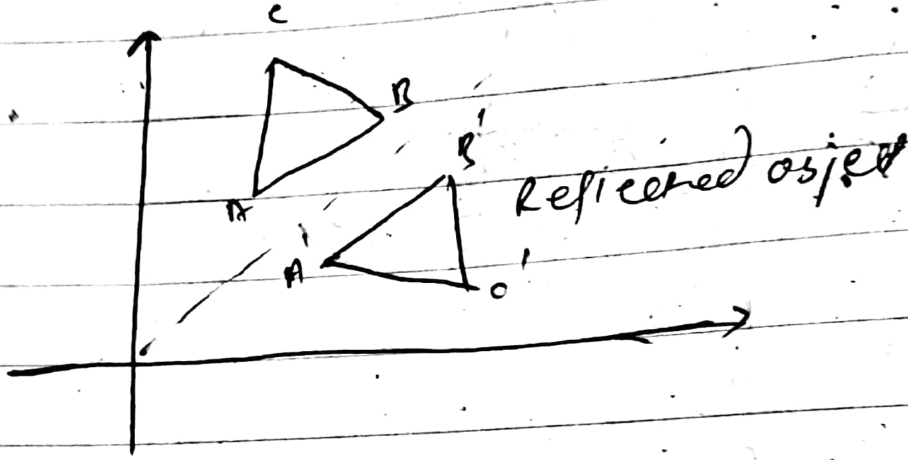
③ Origin



formula,

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

④ $y = x$

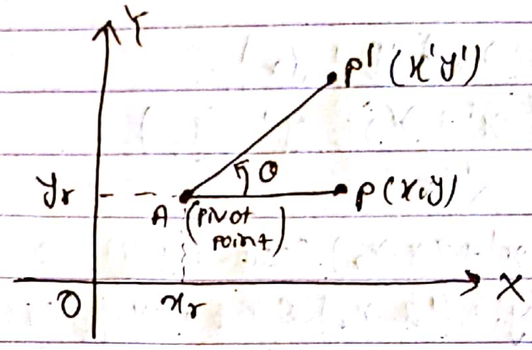


formula,

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

vr9

(1) Rotation of a point (x, y) about any point (x_r, y_r).



Steps:

(a) Translate an object so as to coincide pivot to origin.

(b) Rotate object about an origin.

(c) Translate object back so as to return pivot to original position.

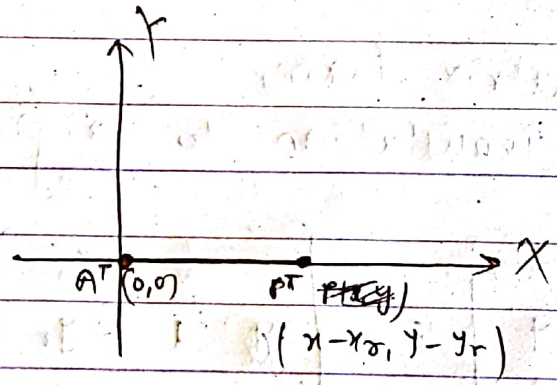
Steps:

(a) Translate to origin

For translation,

$A^T(x_r - x_r, y_r - y_r)$

$P^T(x - x_r, y - y_r)$

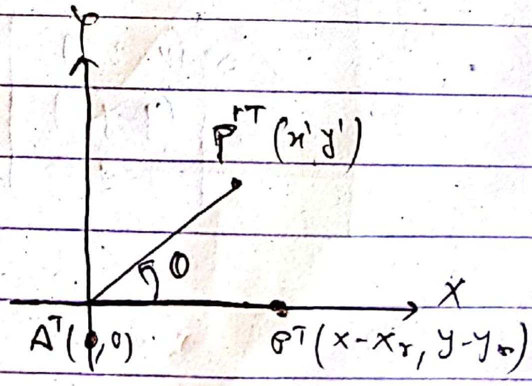


(b) Rotate object with angle θ.

We have for Anti clockwise dir^n,

$x' = (x - x_r)\cos\theta - (y - y_r)\sin\theta$

$y' = (x - x_r)\sin\theta + (y - y_r)\cos\theta.$



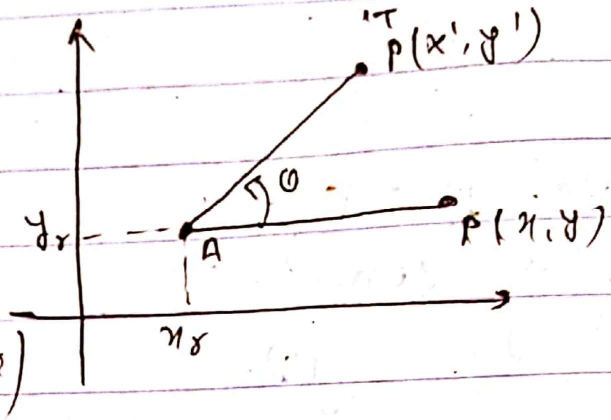
(C) Translate to origin cordinate.

A $(0+x_r, 0+y_r)$

$P'^T(x'+x_r,\ y'+y_r)$

$\Downarrow$

$P'^T(x'+(x-x_r)\cos\theta-(y-y_r)\sin\theta,$
$y'+(x-x_r)\sin\theta+(y-y_r)\cos\theta)$



So,
New cordinates of P'^T after Rotation of P at fixed point (x_r, y_r) with cordinate of $P=(x,y)$ is,

$x'+(x-x_r)\cos\theta-(y-y_r)\sin\theta$
$y'+(x-x_r)\sin\theta+(y-y_r)\cos\theta.$

In matrix form,
(a) Translation to origin

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \text{—— (a)}$$

$$\Rightarrow \begin{bmatrix} (x-x_r) \\ (y-y_r) \\ 1 \end{bmatrix}$$

(b) Rotation about origin.

$$\begin{bmatrix} x' \\ y' \\ h \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (x-x_r) \\ (y-y_r) \\ (1) \end{bmatrix} \quad \text{—— (b)}$$

$$= \begin{bmatrix} (x-x_r)\cos\theta - (y-y_r)\sin\theta \\ (x-x_r)\sin\theta + (y-y_r)\cos\theta \\ 1 \end{bmatrix}$$

(c) Translate again to original point:

$$\begin{bmatrix} x' \\ y' \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (x-x_r)\cos\theta - (y-y_r)\sin\theta \\ (x-x_r)\sin\theta + (y-y_r)\cos\theta \\ 1 \end{bmatrix} \quad \text{— (c)}$$

$$= \begin{bmatrix} x_r + (x-x_r)\cos\theta - (y-y_r)\sin\theta \\ y_r + (x-x_r)\sin\theta + (y-y_r)\cos\theta \\ 1 \end{bmatrix}$$

Composite Transformation is,

$$\underset{\substack{T(x,y) \\ \text{step (c)}}}{\begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix}} \underset{\substack{R(\theta) \\ \text{step (b)}}}{\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}} \underset{\substack{T(-x,-y) \\ \text{step (a)}}}{\begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix}} = T(x,y).R(\theta).T(-x,-y)$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & x_r(1-\cos\theta) + y_r\sin\theta \\ \sin\theta & \cos\theta & y_r(1-\cos\theta) - x_r\sin\theta \\ 0 & 0 & 1 \end{bmatrix} \quad \text{—— (d)}$$

Scanned by CamScanner

① Direct using ⓓ point P is Rotate to P'' T.

as,

$$\begin{bmatrix} x' \\ y' \\ h \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & x_r(1-\cos\theta)+y_r\sin\theta \\ \sin\theta & \cos\theta & y_r(1-\cos\theta)-x_r\sin\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} x\cos\theta - y\sin\theta + x_r(1-\cos\theta) + y_r\sin\theta \\ x\sin\theta + y\cos\theta + y_r(1-\cos\theta) - x_r\sin\theta \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} x_r + (x-x_r)\cos\theta - (y-y_r)\sin\theta \\ y_r + (x-x_r)\sin\theta + (y-y_r)\cos\theta \\ 1 \end{bmatrix}$$

General fixed point Scaling at pivot point (x_f, y_f)

①

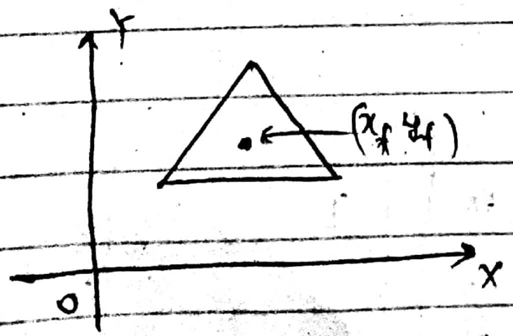


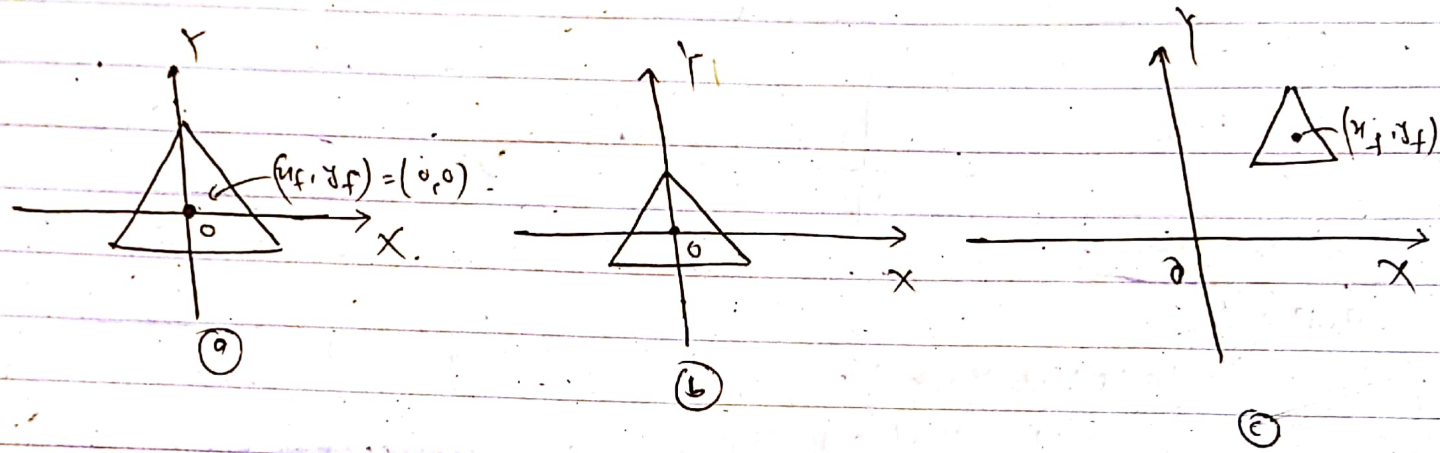
Fig: Original

① steps

(1) Translate (x_f, y_f) to origin.

(2) Scale object with S_x and S_y scaling factor.

(3) Translate (x_f, y_f) to of original position.



Net Transformation =

$$\begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix} \longleftarrow$$

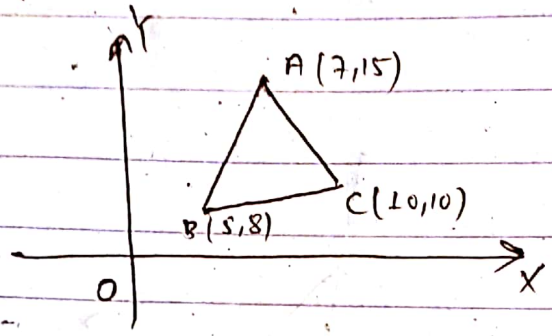
$$= \begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & -x_f S_x \\ 0 & S_y & -y_f S_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_x & 0 & -x_f S_x + x_f \\ 0 & S_y & -y_f S_y + y_f \\ 0 & 0 & 1 \end{bmatrix}$$

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

Examples :-

① Rotate the △ABC by 45° clockwise about the origin and scale it, by (2, 3) about the origin.



Sol^n :-

Steps :- (1) Rotation by 45° (clockwise)

(2) Scaling by (2,3).

∴ Net transformation = $S(2,3)\,R(-45)$

$$= \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 45 & \sin 45 & 0 \\ -\sin 45 & \cos 45 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2/\sqrt{2} & 2/\sqrt{2} & 0 \\ -3/\sqrt{2} & 3/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} = T$$

The transformation points are:

$$A' = TA$$

$$= \begin{bmatrix} 2/\sqrt{2} & 2/\sqrt{2} & 0 \\ -3/\sqrt{2} & 3/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 15 \\ 1 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

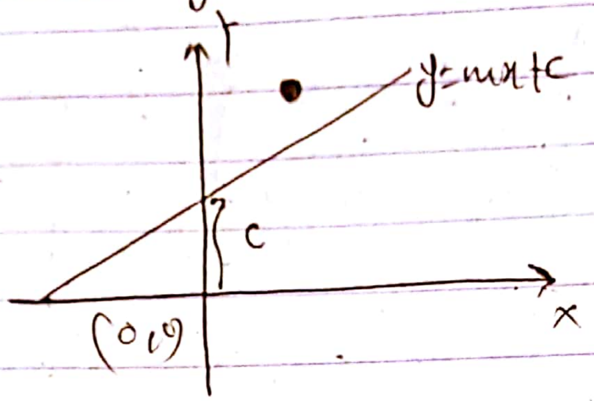
$$B' = TB$$

$$= \begin{bmatrix} 2/\sqrt{2} & 2/\sqrt{2} & 0 \\ -3/\sqrt{2} & 3/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 8 \\ 1 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

$$C' = TC$$

$$= \begin{bmatrix} 2/\sqrt{2} & 2/\sqrt{2} & 0 \\ -3/\sqrt{2} & 3/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ 10 \\ 1 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

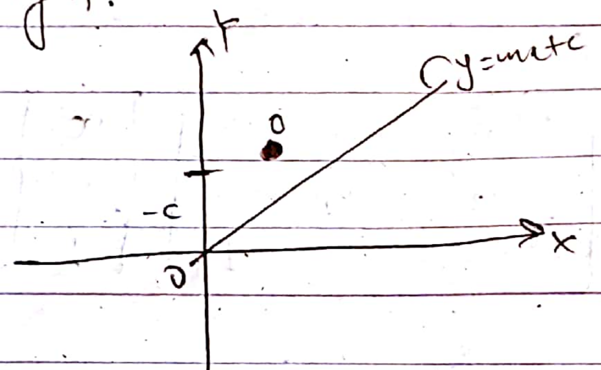
Reflection about y=mx+c



(1) Steps:

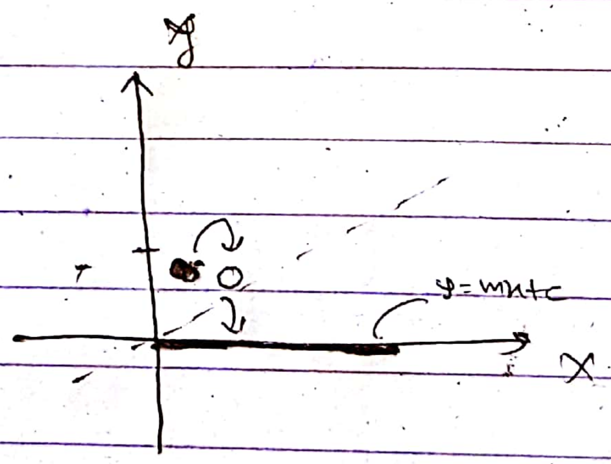
(a) First translate the line so that it passes thru' origin.

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



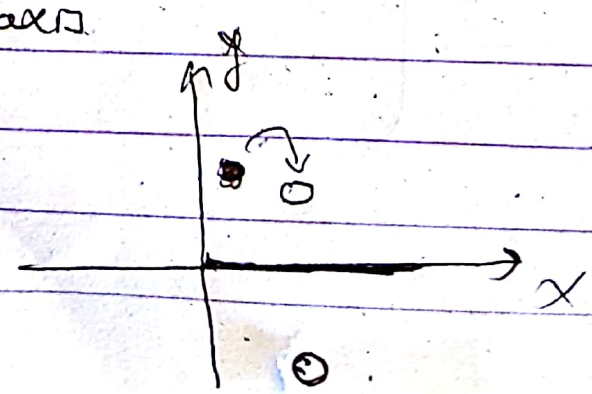
(b) Rotate line onto one of the coordinate axes (say x-axis) and reflect about that x-axis.

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



Reflect about x-axis

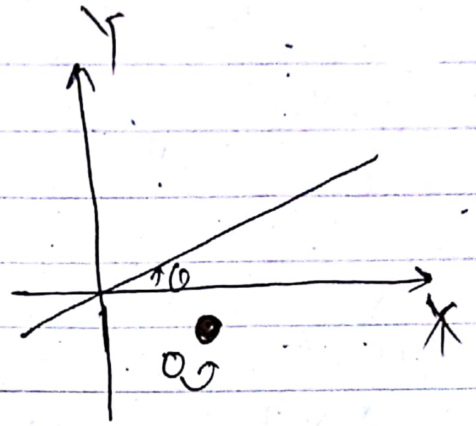
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Scanned by CamScanner

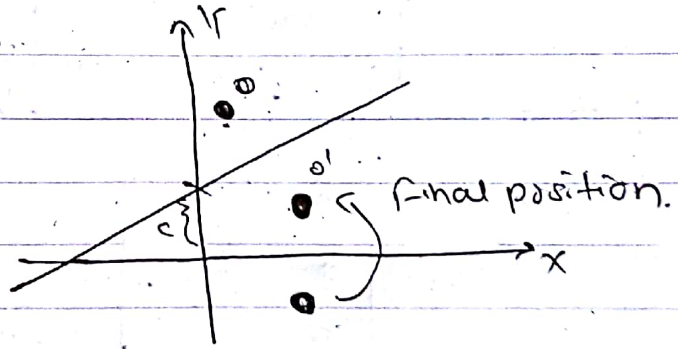
(d) Inverse rotate now,

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



(e) ~~Rotate~~ Translation finally to the original position.

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



Thus, Composite matrix for reflection about $y = mx + c$ is

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & +\sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -c \\ 0 & 0 & 1 \end{bmatrix}}_{\text{solved it by yourself}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Determine transformation matrix responsible for reflection of object about the line $y = x$.

Sol^n:→

(1) $R(-45)$

(2) R_{fx}

(3) $R(45°)$

Net transformation,

$R(45) \; R_{fx} \; R(-45)$

$$= \begin{bmatrix} \cos 45° & -\sin 45° & 0 \\ \sin 45° & \cos 45° & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 45° & \sin 45° & 0 \\ -\sin 45° & \cos 45° & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

The reflection along the line $y = x$ is equivalent to the reflection along the X-axis followed by counterclockwise rotation by α (degree). Find angle α.

Sol^n:→ (a) Reflection along $y = x$

we have,

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow R_{yx}$$

(b) Reflection along X-axis.

R_x

(c) Counterclockwise Rotation, $R(\theta)$

$\therefore$ $\boxed{R_{yx} = R(\theta) \, R_x}$ Find θ which is α

2. Shearing :-

⇒ position fixed on One Cordinates and distortion on other part of an Object.

⇒ x - shear } - skewing.
y - shear }

Now,

x - shear :-

$$y' = y$$
$$x' = x + sh_x \cdot y$$

y - shear :-

$$x' = x$$
$$y' = y + sh_y \cdot x$$

Examples:

(a) X - shear

$sh_x = 2$ units

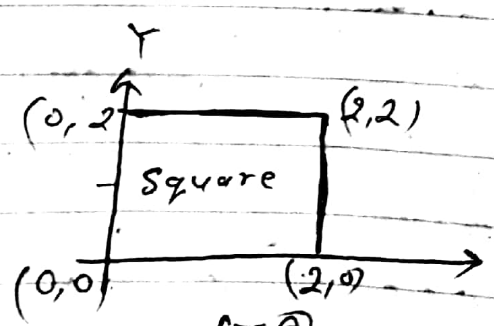


fig (a)

(0,0) $y' = 0$

$x' = x + sh_x y = 0 + 2 \times 0 = (0)$

$(x', y') = (0, 0)$

$\underline{(2,0)}$ $\quad y' = 0$

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

$(x'y') = (2,0)$

$\underline{(0,2)}$ $\quad y' = 2$

$x' = x + sh_x y = 0 + 2 \times 2 = 4$

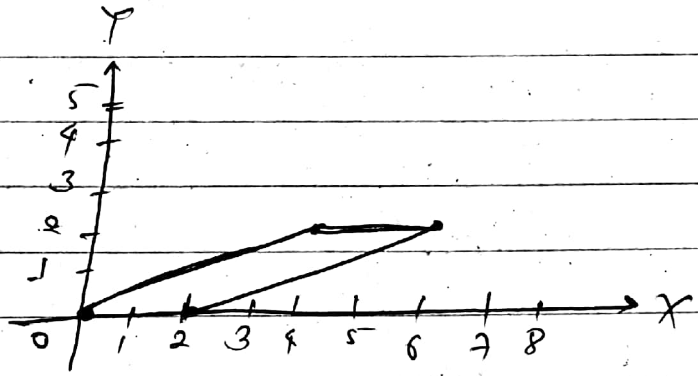
$(x'y') = (4,2)$

$\underline{(2,2)}$ $\quad y' = 2$

$x' = x + sh_x y = 2 + 2 \cdot 2 = 6$

$(x'y') = (6,2)$

Plot



Assignment: In fig (a), use shear on Y-axis with $sh_y = 2$ units.

In matrix form:

X-shear

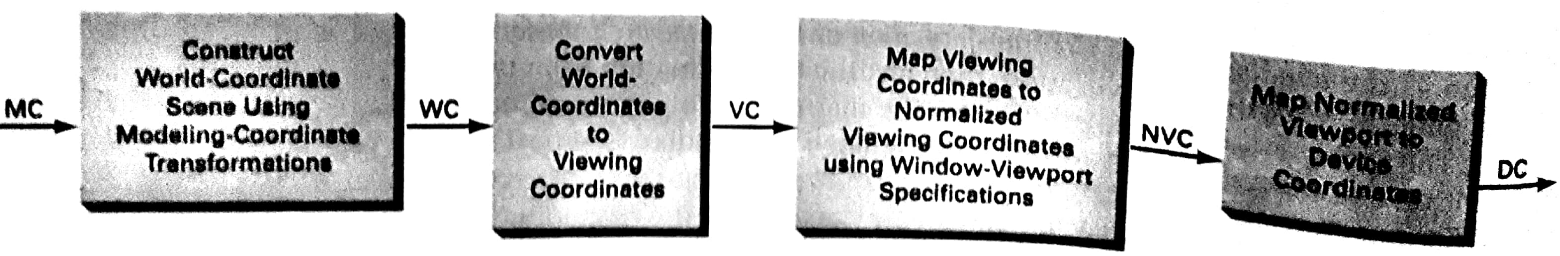
$$\begin{bmatrix} x' \\ y' \\ h \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Y-shear

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

Some graphics packages that provide window and viewport operations allow only standard rectangles, but a more general approach is to allow the rectangular window to have any orientation. In this case, we carry out the viewing transformation in several steps, as indicated in Fig. 6-2. First, we construct the scene in world coordinates using the output primitives and attributes discussed in Chapters 3 and 4. Next, to obtain a particular orientation for the window, we can set up a two-dimensional **viewing-coordinate system** in the world-coordinate plane, and define a window in the viewing-coordinate system. The viewing-coordinate reference frame is used to provide a method for setting up arbitrary orientations for rectangular windows. Once the viewing reference frame is established, we can transform descriptions in world coordinates to viewing coordinates. We then define a viewport in normalized coordinates (in the range from 0 to 1) and map the viewing-coordinate description of the scene to normalized coordinates. At the final step, all parts of the picture that lie outside the viewport are clipped, and the contents of the viewport are transferred to device coordinates. Figure 6-3 illustrates a rotated viewing-coordinate reference frame and the mapping to normalized coordinates.

By changing the position of the viewport, we can view objects at different positions on the display area of an output device. Also, by varying the size of viewports, we can change the size and proportions of displayed objects. We achieve zooming effects by successively mapping different-sized windows on a



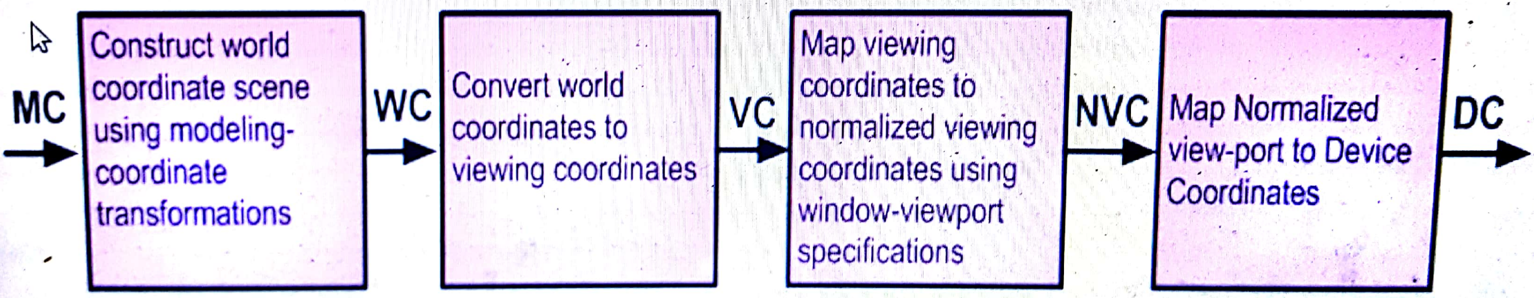
fixed-size viewport. As the windows are made smaller, we zoom in on some part of a scene to view details that are not shown with larger windows. Similarly, more overview is obtained by zooming out from a section of a scene with successively larger windows. Panning effects are produced by moving a fixed-size window across the various objects in a scene.

Viewports are typically defined within the unit square (normalized coordinates). This provides a means for separating the viewing and other transformations from specific output-device requirements, so that the graphics package is largely device-independent. Once the scene has been transferred to normalized coordinates, the unit square is simply mapped to the display area for the particular output device in use at that time. Different output devices can be used by providing the appropriate device drivers.

When all coordinate transformations are completed, viewport clipping can be performed in normalized coordinates or in device coordinates. This allows us to reduce computations by concatenating the various transformation matrices. Clipping procedures are of fundamental importance in computer graphics. They are used not only in viewing transformations, but also in window-manager systems, in painting and drawing packages to eliminate parts of a picture inside or outside of a designated screen area, and in many other applications.

2-D VIEWING TRANSFORMATION PIPELINE

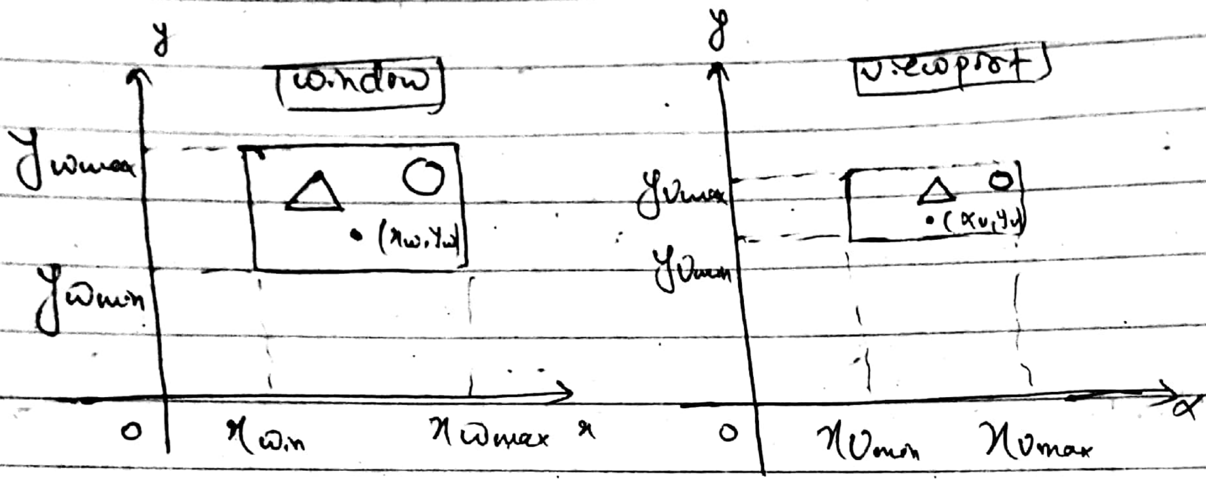
- Procedures for displaying views of a two-dimensional picture on an output device:
  - Specify which parts of the object to display (clipping window, or world window, or viewing window)
  - Where on the screen to display these parts (viewport).
- Clipping window is the selected section of a scene that is displayed on a display window.
- Viewport is the window where the object is viewed on the output device.



MC-Modeling Coordinates, WC-world Coordinates
VC-Viewing Coordinates NVC- Normalized Viewing Coordinates
DC- Device Coordinates

window to viewport transformation

↳ the process of finding the view cordinates from the world cordinates.



- Here, the size may vary but the relative position is always same.

$$\frac{x_w - x_{wmin}}{x_{wmax} - x_{wmin}} = \frac{x_v - x_{vmin}}{x_{vmax} - x_{vmin}} \quad \text{—— (i)}$$

and,

$$\frac{y_w - y_{wmin}}{y_{wmax} - y_{wmin}} = \frac{y_v - y_{vmin}}{y_{vmax} - y_{vmin}} \quad \text{—— (ii)}$$

from (i),

$$x_v - x_{vmin} = (x_{vmax} - x_{vmin})\left(\frac{x_w - x_{wmin}}{x_{wmax} - x_{wmin}}\right)$$

$$= (x_w - x_{wmin})\underbrace{\left(\frac{x_{vmax} - x_{vmin}}{x_{wmax} - x_{wmin}}\right)}_{S_x}$$

S_x → scaling factor

$\therefore X_v = X_{vmin} + (X_w - X_{wmin}) \cdot S_x$

from ①,

$\parallel^{ly}$,

$$\boxed{y_v = y_{vmin} + (y_w - y_{wmin}) \cdot S_y}$$

S_x and S_y are scaling factors.

Example:

$X_{wmin} = 20$ & $X_{vmin} = 30$

$X_{wmax} = 80$. & $X_{vmax} = 60$.

$y_{wmin} = 40$ & $y_{vmin} = 40$

$y_{wmax} = 80$ & $y_{vmax} = 60$

$(X_w, y_w) = (30, 80)$ & $(X_v, y_v) = ?$

solution:

X_v & y_v use formula,

you get,

$X_v = 35$

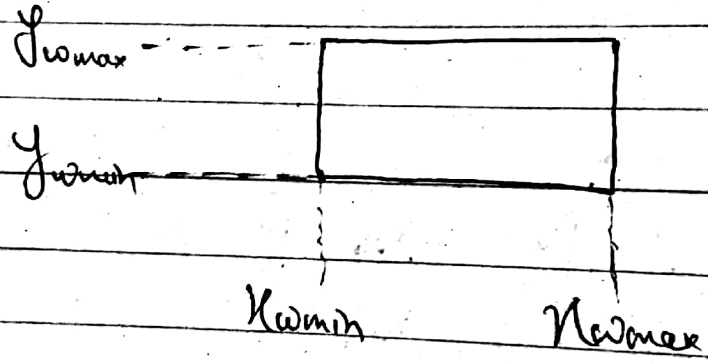
$y_v = 60$

① clipping :-

the process of decarding the portion outside the window.

(1) point clipping
(2) line clipping
(3) polygon clipping.

① Point clipping :-



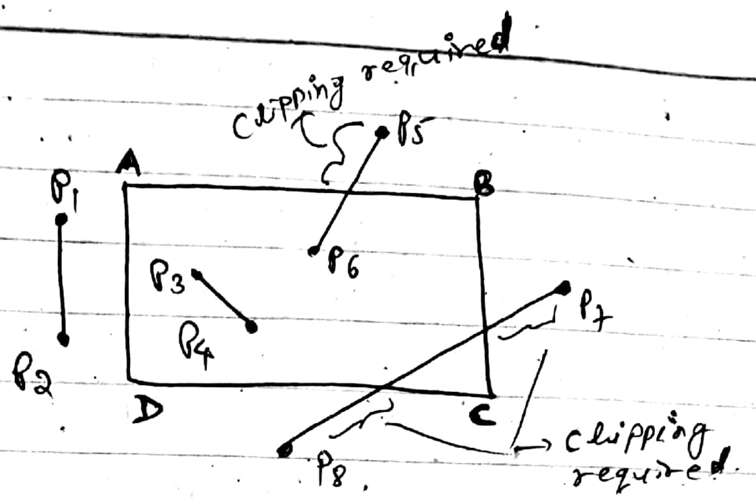
If (x,y) be any given point then it must satisfy the condn

$$\boxed{x_{wmin} \leq \boxed{x} \leq x_{wmax}}$$
$$\boxed{y_{wmin} \leq \boxed{y} \leq y_{wmax}}$$

to lies on the window. (inside the window)
If one condn in above eqn is not valid then the point lies outside the window.

(a) Line clipping i.g.

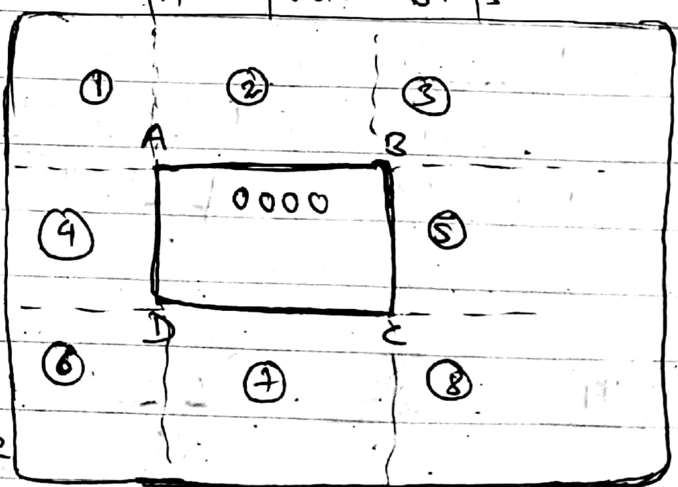
P_1P_2 → Reject
P_3P_4 → Consider (accept)
P_5-P_6 → clipping required
P_7P_8 → clipping required



In P_5P_6 and P_7P_8 clipping required because some or whole points (endpoints) lies outside the window, of the straight lines.

There are many algorithms, among them
(i) Cohen - Sutherland

In this algorithm, each region is assigned with a Region code, where initially window region is assigned with four bits (0000) code.
and, Other region around the window is assigned with specific code based on such rules



ABRL
where A = above
B = below
R = right
L = left

In each region, the code is assigned 0 or 1. If region belong on ABRL then 1 else 0.

Region ① ⇒ top-left ⇒ 1001

" ② ⇒ top ⇒ 1000

" ③ ⇒ top(above) - right ⇒ 1010

④ ⇒ left ⇒ 0001

" ⑤ ⇒ right ⇒ 0010

" ⑥ ⇒ left-below ⇒ 0101

" ⑦ ⇒ Below ⇒ 0100

" ⑧ ⇒ Right-below ⇒ 0110

Now,

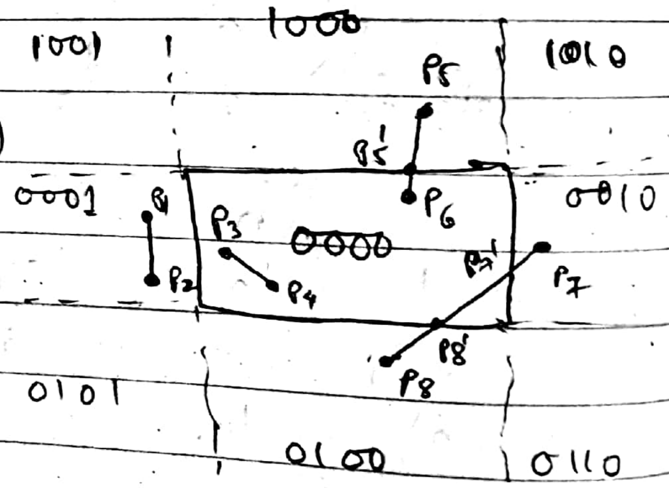
Consider, line P_1P_2 in which region ??

(I)

P_1 = 0001 (non-zero)

P_2 = 0001 (non-zero)

do → And (operation) = 0001 (non-zero)



⇒ If Both points are in region non-zero and "result" is non-zero, reject line

(II) P_3 & P_4

P_3 = 0000 - (zero)

P_4 = 0000 - (zero)

⇒ and = 0000 - (zero)

⇒ If both points are on region 0000 and result is 0000 the accept the line.

(III) P_5 & P_6

P_5 = 1000 (non-zero)
P_6 = 0000 (zero)

0000 (zero) ⇒ clipping required.

⇒ If ~~both~~ one point is in region 1000 and other in 0000 and result is (0000) then clipping must be done, at some portion, Because one point line in window and other not.

⇒ Find a point of the line that is intersect with the boundary of the window.
say it's, P_5'

P_5' = 0000
P_6 = 0000

0000 accept.

(IV) P_7 & P_8

P_7 = 0010 non-zero
P_8 = 0100 non-zero

AND ⇒ 0000 zero

Now,

P_7' and P_8.

P_7' ⇒ 0000 zero
P_8 ⇒ 0100 non-zero

0000 ⇒ clipping required

again,

P_8' & P_7',

P_7' ⇒ 0000
P_8' ⇒ 0000

0000 $\boxed{\text{accept}}$

① Given a clipping window A(10,10), B(40,10), C(40,40), D(10,40) using Cohen-Sutherland line clipping. Find region code for each end points of lines P_1P_2, P_3P_4, P_5P_6, where $P_1(5,15)$, $P_2(25,30)$, $P_3(15,15)$, $P_4(35,30)$, $P_5(5,8)$ and $P_6(40,15)$. Also find the clipped line using above parameters.

Solⁿ :

(I) P_1P_2.

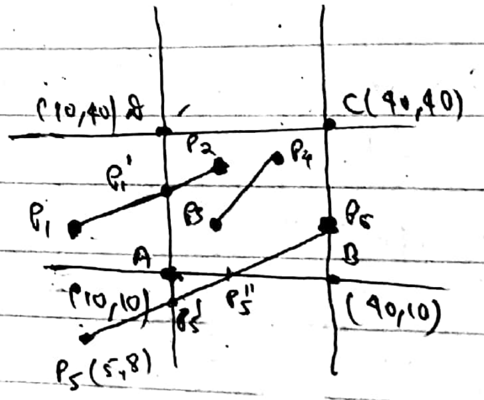
$P_1 \rightarrow 0001$ (non-zero)

$P_2 \rightarrow 0000$ (zero)

AND → 0000 (zero)

⇓

clipping required.



Now,
we have to find the cordinate of P_1'.

$x = 10$

$y = ?$

~~we have~~

We have,

1001	1000	1010
0001	0000 window	0010
0101	0100	0110

we have two point formula for slope,

$P_1(5,15)$ $P_2(25,30)$

$$m = \frac{30-15}{25-5} = \frac{15}{20} = 0.75$$

Again,

$\therefore y = y_1 + m(x - x_1)$ passes thru $P_1(5,15)$

$= 15 + 0.75(10-5)$

$= 18.75$

$\boxed{P_1'(10, 18.75)}$

$P_1' = 0000$ zero

$P_2 = 0000$ zero

AND $\Rightarrow$ 0000 zero $\Rightarrow$ accept.

(ii) P_3 & P_4

$P_3 = 0000$ $\Rightarrow$ zero

$P_4 \Rightarrow 0000$ $\Rightarrow$ zero

AND 0000 zero $\Rightarrow$ accept.

(iii) P_5 & P_6

$P_5 \Rightarrow 0101$ non-zero

$P_6 \Rightarrow 0000$ zero

0000 zero

$\Downarrow$

$\Rightarrow$ Now, clipping required.

P_5' at region (0100).

$P_5'(x, y) = (?, ?) = (10, ?)$ passes thru' $(5, 8)$

$$\therefore \quad m_{P_5P_6} = \frac{15-8}{40-5} = 0.2$$

$$\text{or} \quad y = 8 + (10-5) \times 0.2$$
$$= 8 + 5 \times 0.2$$
$$= 9$$

$$\boxed{\therefore \quad P_5' = (10, 9)}$$

$\Rightarrow$ Now, $P_5' \Rightarrow 0100$ (non-zero)

$P_6 \Rightarrow 0000$ (zero)

0000 (zero)

$\Downarrow$

clipping required.

$P_5'' \Rightarrow 0000$ (zero)

$P_6 \Rightarrow 0000$ (zero)

0000 (zero)

Now,

Find cordinate of P_5''.

$m = 0.2$

so, $y_1 + (x - x_1) m$

$y' = 9 + (x - 10)\, 0.2$

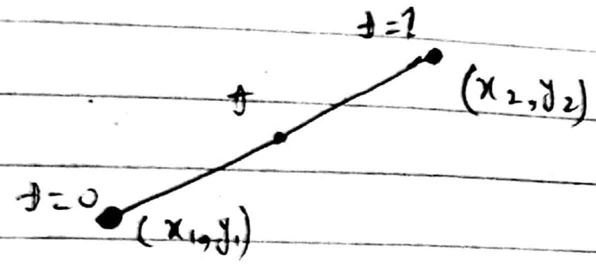
or, $10 = 9 + (x-10) \times 0.2$

or, $\frac{1}{0.2} + 10 = x$

$\therefore x = 15$.

$$\boxed{P_5'' = (15, 10)}$$

Liang - Barsky



⇒ time interval at initial point is $t = 0$.

⇒ time interval at final point is $t = 1$.

⇒ let say, interval $t\ (x, y)$ in between them, where $0 < t < 1$

⇒ so, from parametric eqn of line is

$$x = t \cdot x_2 + (1 - t)\, x_1$$
$$= t x_2 + x_1 - t x_1$$
$$= x_1 + t\,(x_2 - x_1)$$
$$= x_1 + t \Delta x \quad \text{where } \Delta x = (x_2 - x_1) \quad — (a)$$

||'ly,

$$y = y_1 + t \Delta y \quad \text{where } \Delta y = (y_2 - y_1) \quad — (b)$$

we have point clipping as,

$$x_{wmin} \le x \le x_{wmax} \quad — (c)$$
$$y_{wmin} \le y \le y_{wmax} \quad — (d)$$

from (a), (b) (c), and (d)

$$x_{wmin} \le (x_1 + t \Delta x) \le x_{wmax}$$
$$y_{wmin} \le (y_1 + t \Delta y) \le y_{wmax}$$

$x_1 + t \cdot \Delta x \geq x_{wmin}$ — (e)

$x_1 + t \cdot \Delta x \leq x_{wmax}$ — (f)

$y_1 + t \cdot \Delta y \geq y_{wmin}$ — (g)

$y_1 + t \cdot \Delta y \leq y_{wmax}$ — (h)

Now,

General eqⁿ for above eqⁿs are

$$t \cdot p_k \leq q_k \quad \text{— (I)} \quad k = 1, 2, 3, 4 \ldots$$

So,

Converting all above eqⁿs in General form,

$t \cdot \Delta x \geq (x_{wmin} - x_1)$ — (i)

$t \cdot \Delta x \leq (x_{wmax} - x_1)$ — (ii)

$t \cdot \Delta y \geq (y_{wmin} - y_1)$ — (iii)

$t \cdot \Delta y \leq (y_{wmax} - y_1)$ — (iv)

again,

$$\left.\begin{aligned} -t \cdot \Delta x &\leq (x_1 - x_{wmin}) \\ t \cdot \Delta x &\leq (x_{wmax} - x_1) \\ -t \cdot \Delta y &\leq (y_1 - y_{wmin}) \\ t \cdot \Delta y &\leq y_{wmax} - y_1 \end{aligned}\right\}$$

Comparing with (I),

$p_1 = (-\Delta x)$ $\qquad q_1 = x_1 - x_{wmin}$

$p_2 = \Delta x$ $\qquad q_2 = x_{wmax} - x_1$

$p_3 = (-\Delta y)$ $\qquad q_3 = y_1 - y_{wmin}$

$p_4 = \Delta y$ $\qquad q_4 = y_{wmax} - y_1$

Case (I), If $p_k < 0$ then

$$t_1 = \max\left(0, \frac{q_k}{p_k}\right)$$

$$\left.\begin{aligned} x &= x_1 + t_1 \cdot \Delta x \\ y &= y_1 + t_1 \cdot \Delta y \end{aligned}\right\}$$

where,
t_1 changes, $t_1 \neq 0$
implies outside window

Case (II) If $p_k > 0$ then,

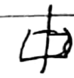
$$t_2 = \min\left(1, \frac{q_k}{p_k}\right)$$

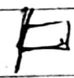
$$x = x_1 + t_2 \Delta x$$
$$y = y_1 + t_2 \Delta y$$

where,
t_2 changes, $t_2 \neq 1$
implies outside
window.

Case (III) If $p_k = 0$ $\Rightarrow$ line parallel to window

$q_k < 0$ $\rightarrow$ line outside 

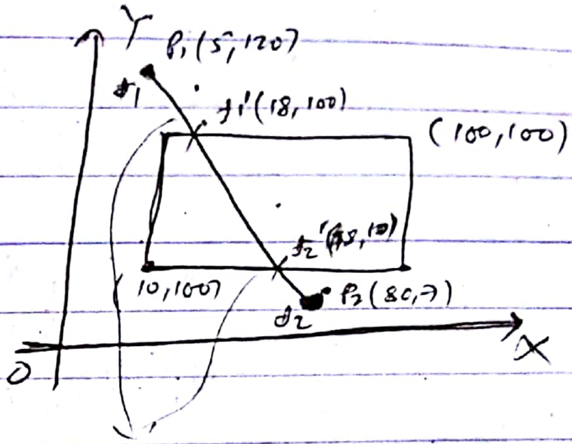
$q_k > 0$ $\rightarrow$ line inside / partial inside 

$q_k = 0$ $\rightarrow$ within boundary / partial 

Example:

Find the clipped region in window of diagonal vertex (10,10) and (100,100) for line P_1 (5,120) and P_2 (80,7) using Liang-Barsky.

Solution:



$P_1 (5,120)$ and $P_2 (80,7)$

$\therefore \Delta x = 80-5 = 75$

$\Delta y = 7-120 = -113$

$\Rightarrow P_1 = -\Delta x = -75$

$P_2 = \Delta x = 75$

$P_3 = -\Delta y = 113$

$P_4 = \Delta y = -113$

$\Rightarrow q_1 = 5-10 = -5$

$q_2 = 100-5 = 95$

$q_3 = 120-10 = 110$

$q_4 = 100-120 = -20$

$\therefore t_1' = (18,100)$

$t_2' = (78,10)$

$\Rightarrow$ Case I

$P_k < 0 \quad (P_1, P_4)$

$\Rightarrow t_1 = \max\left(0, \frac{q_1}{P_1}, \frac{q_4}{P_4}\right)$

$= \max\left(0, \frac{-5}{-75}, \frac{20}{113}\right)$

$= \frac{20}{113}$

t_1 changes

$x = 5 + \frac{20}{113} \times 75 = 18.27$

$y = 120 + \frac{20}{113} \times -113 = 100$

Case II

$P_k > 0 \quad (P_2, P_3)$

$\Rightarrow t_2 = \min\left(1, \frac{q_2}{P_2}, \frac{q_3}{P_3}\right)$

$= \min\left(1, \frac{95}{75}, \frac{110}{113}\right)$

$= \frac{110}{113}$

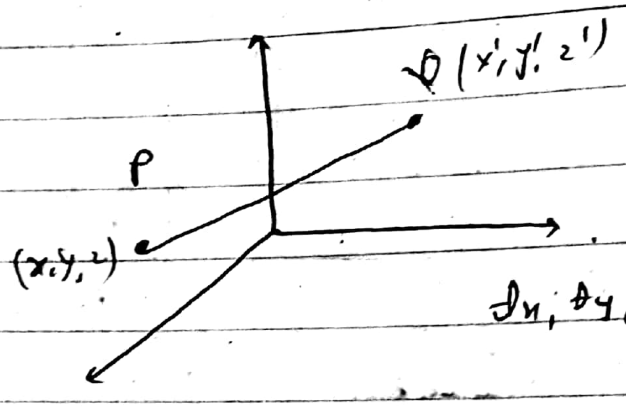
t_2 changes

$x = 5 + \frac{110}{113} \times 75 = 78$

$y = 120 + \frac{110}{113} \times -113 = 120 - 110 = 10$

3-D transformation:

① Translation:



$t_x, t_y, t_z \Rightarrow$ translation parameter

$$x' = x + t_x$$
$$y' = y + t_y$$
$$z' = z + t_z.$$

Matrix;

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

$$P' = P + T$$

In homogeneous matrix form.

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

or $P' = T \cdot P$

② Rotation:

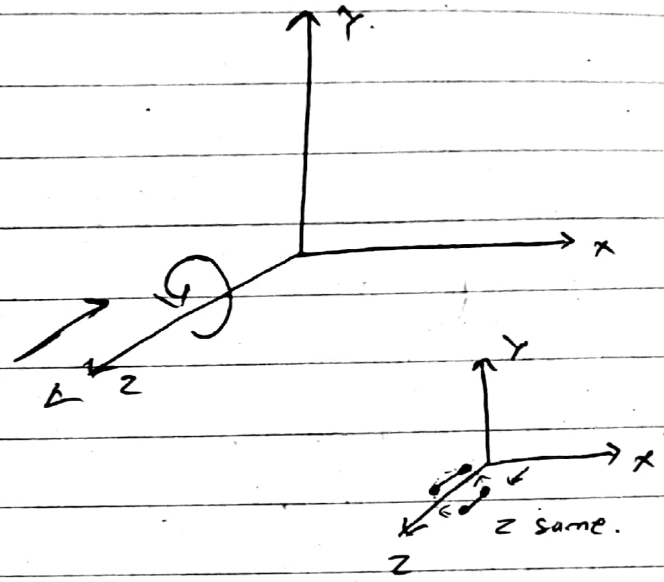
Case ①

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

Z-Axis Rotation

Now,

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



$$\Rightarrow P' = R_z(\theta) . P$$

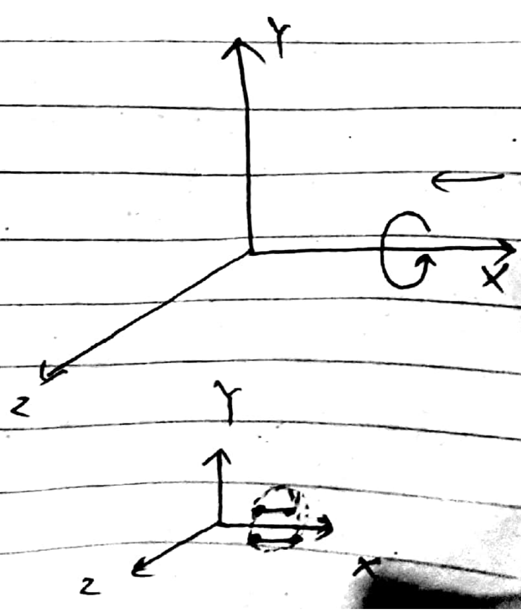
Case ②

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

X-axis Rotation

Now,

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

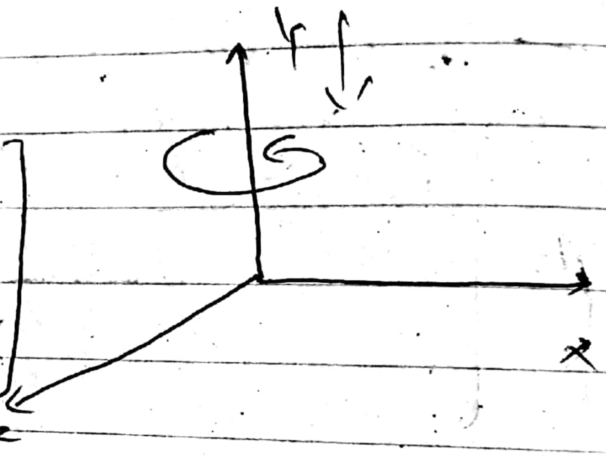


$$\Rightarrow P' = R_x(\theta) . P$$

Case ©

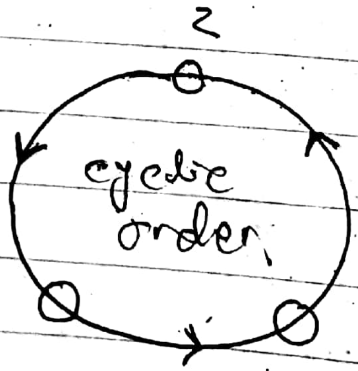
$$z' = z\cos\theta - x\sin\theta$$
$$x' = z\sin\theta + x\cos\theta$$
$$y' = y$$

Y-axis Rotation



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

or, $P' = R_y(\theta) \cdot P$



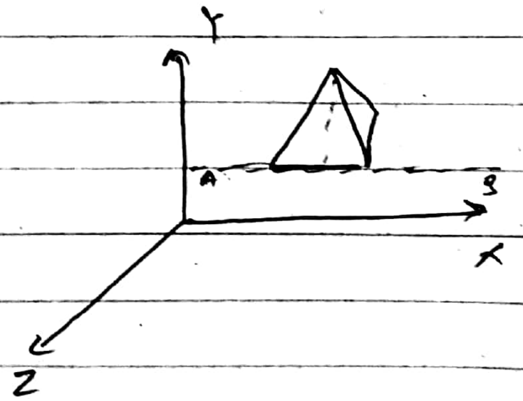
$x \to y \to z \to x$

Rotation about axis parallel to Co-ordinate axis:

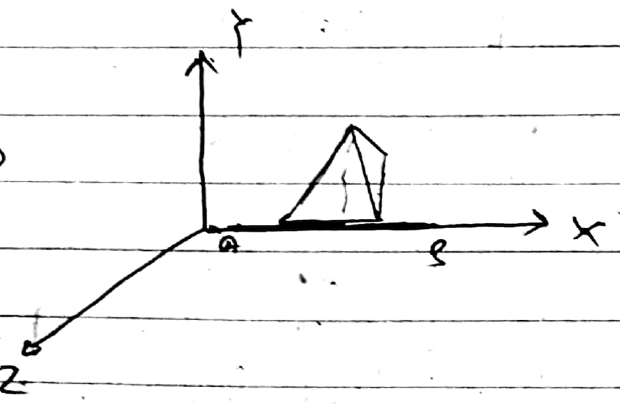
(a) When an Object is to be rotated about an axis that is parallel to one of the Co-ordinate axis, we need to perform some series of transformation.

(1.) Translate the Object so that the rotation axis Coincides with the parallel Co-ordinate axis.

$T(-a, -b, -c)$ where (a, b, c) is any point on the rotation axis.

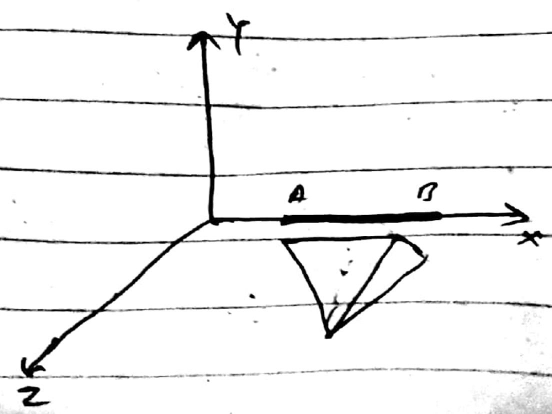


$\Rightarrow$

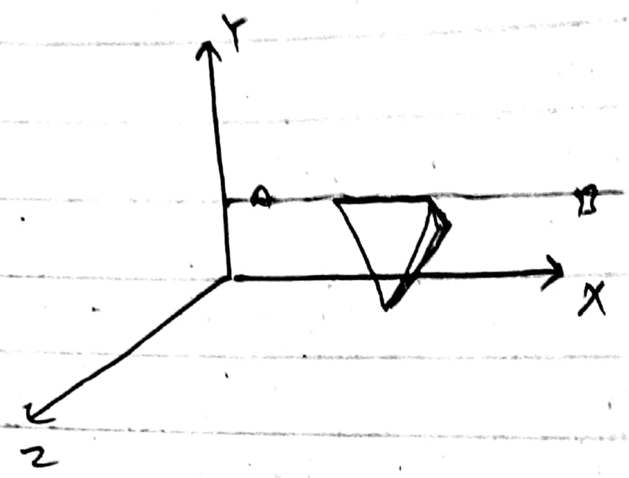


(2) Performed specific Rotation about the required axis. $R(\theta)$

for ex: Here required axis is x-axis so $R_x(\theta)$.



(3) Translate the Object so that the rotation axis is moved to its original position.
$T(a,b,c)$

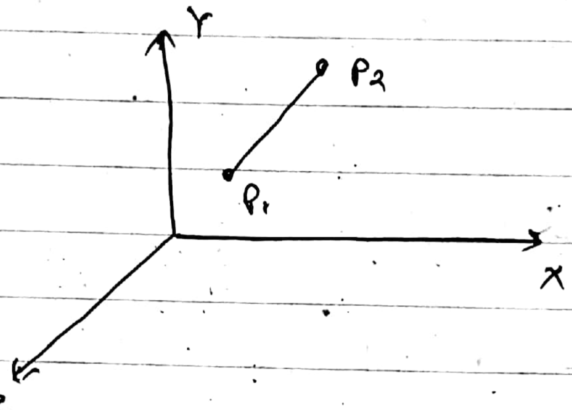


Net Transformation Matrix is
$M_{net} = T(a,b,c)\, R(\theta)\, T(-a,-b,-c)$
If $P(x,y,z)$ be any point then,

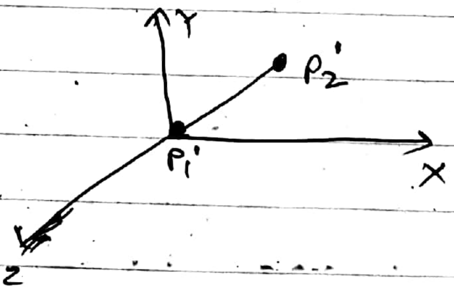
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = M_{net} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

(b) Not parallel to any of the Co-axis.

when an object is to be rotated about an axis that is not parallel to one of the Cordinate axes, we need to perform some series of transformation.



(i) Translate the object such that rotation axis passes thru cordinate origin.

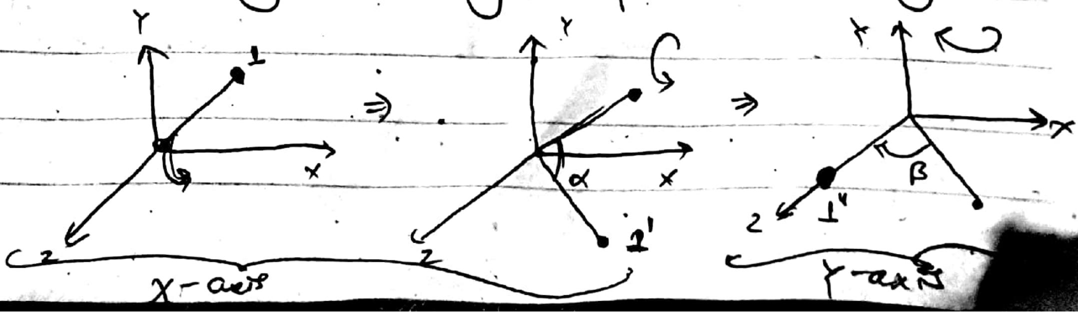


$$T = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(ii) Rotate the axis such that axis of rotation coincides with one of the Cordinate axis.

(a) It's first rotation by an angle α about X-axis

(b) Rotate by an angle β about y-axis.



(9)

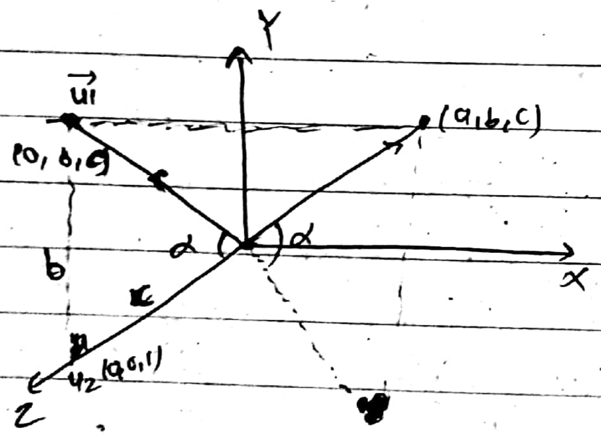
If $P_1(x_1, y_1, z_1)$ and $P_2(x_2, y_2, z_2)$ then,

$$v = (P_2 - P_1) = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$$

Unit vector,

$$u = \frac{\vec{v}}{|\vec{v}|} = \frac{(x_2 - x_1, y_2 - y_1, z_2 - z_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}}$$

$= (a, b, c)$ where a, b, c are direction Cosines.



$$\Rightarrow \cos\alpha = \frac{\vec{u'} \cdot \vec{u_2}}{|\vec{u'}| \cdot |\vec{u_2}|}$$

$$\Rightarrow \frac{(0, b, c) \cdot (0, 0, 1)}{\sqrt{b^2 + c^2} \cdot (1)}$$

$$= \frac{c}{\sqrt{b^2 + c^2}}$$

$$= \frac{c}{\sqrt{b^2 + c^2}} \quad \left[\because d = \sqrt{b^2 + c^2}\right]$$

$$\Rightarrow \sin\alpha = \frac{\vec{u'} \times \vec{u_2}}{|\vec{u'}| \, |\vec{u_2}|}$$

$$= \frac{(0, b, c) \times (0, 0, 1)}{\sqrt{b^2 + c^2}}$$

$$= \frac{b}{d}$$

⇒ Rotation along x-axis,

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & -b/d & 0 \\ 0 & b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(b) Rotation along y-axis,

$$R_Y(\beta) = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{——— (a)}$$

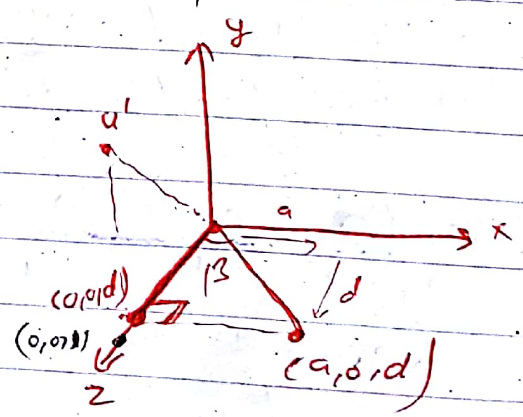
Now how to find $\cos\beta$ and $\sin\beta$??

$\sin\beta = a/l \qquad \cos\beta = d/l$

$\because l^2 = a^2 + b^2 + c^2$

$l^2 = a^2 + d^2$

$\therefore l = \sqrt{a^2 + d^2}$



(c) Rotation along z-axis

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scanned by CamScanner

(iii) Reverse Transformation

$$\Rightarrow T^{-1} R_x^{-1}(\alpha) R_y^{-1}(\beta)$$

$$= \begin{bmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

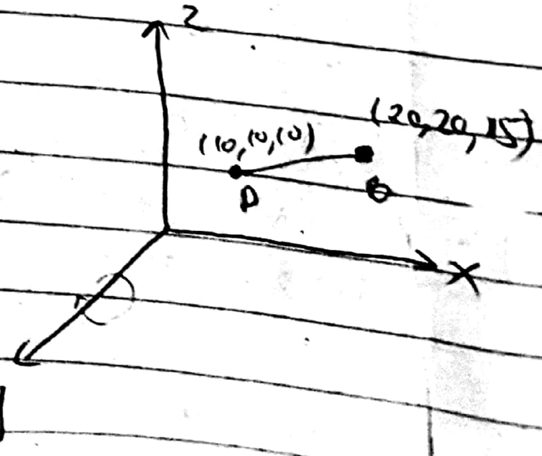
Now,

(iv) Net transformation matrix,

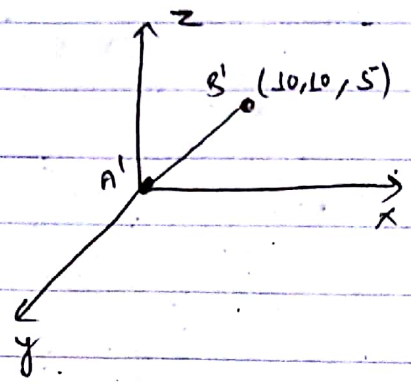
$$= T^{-1} R_x^{-1}(\alpha) R_y^{-1}(\beta) R_z(\theta) R_y(\beta) R_x(\alpha) T$$

Q: How can you perform three dimensional rotations of an object about some arbitrary axis? Explain.

Q. perform rotation of a line (10,10,10) (20,20,15) about Y-axis in clockwise dirn. by 90°.

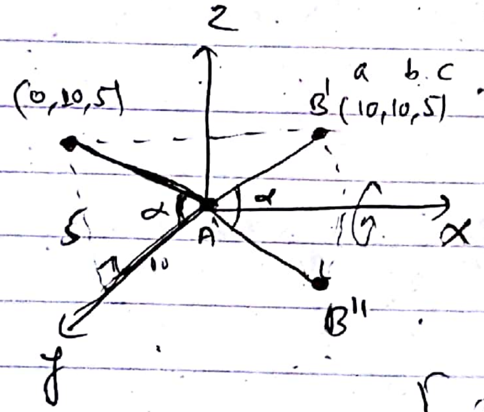


Step 1. Translate point A to origin.



$$T = \begin{bmatrix} 1 & 0 & 0 & -10 \\ 0 & 1 & 0 & -10 \\ 0 & 0 & 1 & -10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 2. Rotate about z-axis by angle β and Rotate about x-axis by angle α.

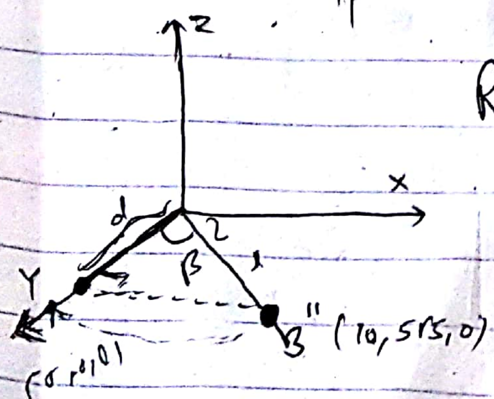


$$\sin\alpha = \frac{5}{\sqrt{5^2+10^2}} = \frac{5}{\sqrt{125}} = \frac{5}{5\sqrt{5}}$$

$$\cos\alpha = \frac{10}{\sqrt{5^2+10^2}} = \frac{10}{5\sqrt{5}} = \frac{2}{\sqrt{5}}$$

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2/\sqrt{5} & -1/\sqrt{5} & 0 \\ 0 & 1/\sqrt{5} & 2/\sqrt{5} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 3. Rotate axis A'B'' about z-axis by an angle β. untill it coincides with the z-axis.



$$R_z(\beta) = \begin{bmatrix} \cos\beta & -\sin\beta & 0 & 0 \\ \sin\beta & \cos\beta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$l^2 = 10^2 + 10^2 + 5^2$

$l = \sqrt{225}$

$\cancel{\sin\beta = \frac{10}{5\sqrt{5}}}$

$$\sin\beta = \frac{10}{\sqrt{225}}$$

$$\cos\beta = \frac{5\sqrt{5}}{\sqrt{225}}$$

step 4. Rotate line 90° about y-axis clockwise.

$$R_y(-90) = \begin{bmatrix} \cos 90° & 0 & \sin 90 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin 90° & 0 & \cos 90° & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

step 5

Finally, with inverse matrix as well.

$$\underbrace{R(\theta)}_{\text{Net matrix}} = \underbrace{T^{-1} R_x^{-1}(\alpha) R_z^{-1}(\beta) R_y(\theta) R_z(\beta) R_x(\alpha) T}$$

Solve by yourself.

step 6

multiply 4x4 matrix obtained as $R(\theta)$ with original plane's point.

$$P' = \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}_{4\times4} \cdot \overbrace{\begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}^{P}$$

&

$$P' = \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} 20 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 15 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Q. Find the new cordinates of unit cube 90° rotated about an axis by its endpoints $A(2,1,0)$ and $B(3,3,1)$.

③ Scaling :-

(i) About origin

$x' = x \cdot S_x$

$y' = y \cdot S_y$

$z' = z \cdot S_z$

Now matrix form,

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

S_x, S_y and S_z are Scaling parameters having the (+) ve Values.

(ii) About any fixed point (x_f, y_f, z_f)

Step ⓐ. Translate fixed point to the origin.
$T(-x_f, -y_f, -z_f)$

Step ⓑ Scale the Object relative to the Cordinate origin.
$S(S_x, S_y, S_z)$

Step ⓒ Translate the fixed point back to the origin position. $T(x_f, y_f, z_f)$

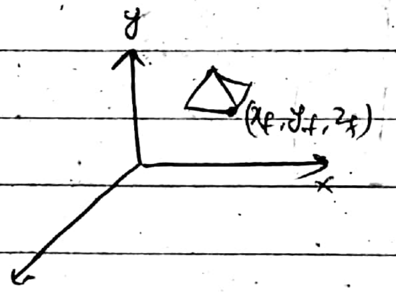
Net transformation: ↓

$$T = T(x_f, y_f, z_f) \cdot S(s_x, s_y, s_z) \cdot T(-x_f, -y_f, -z_f)$$

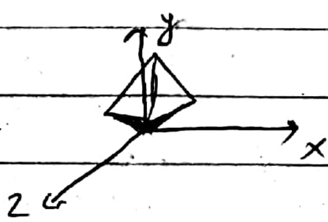
$$\Rightarrow \begin{bmatrix} 1 & 0 & 0 & x_f \\ 0 & 1 & 0 & y_f \\ 0 & 0 & 1 & z_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -x_f \\ 0 & 1 & 0 & -y_f \\ 0 & 0 & 1 & -z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} s_x & 0 & 0 & (1-s_x)x_f \\ 0 & s_y & 0 & (1-s_y)y_f \\ 0 & 0 & s_z & (1-s_z)z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

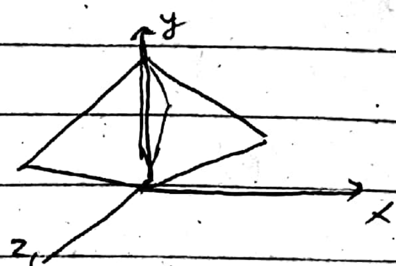
Given;



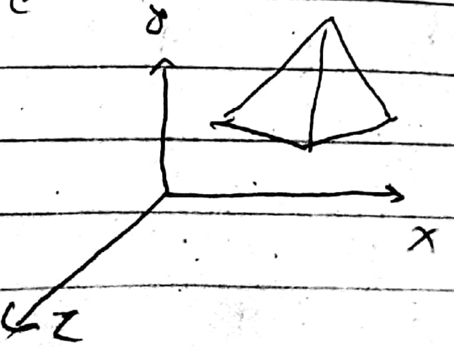
step ⓐ



step ⓑ



step ⓒ



(4) Reflection:

(i) About Axis:

Axis reflection is equivalent to 180° rotation about the axis in 3D space.

(a) X-axis,

$$R_{f(x)} = R_x(180^\circ)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 180^\circ & -\sin 180 & 0 \\ 0 & \sin 180 & \cos 180 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(b) Y-axis,

$$R_{f(y)} = R_y(180)$$

$$= \begin{bmatrix} \cos 180^\circ & 0 & \sin 180 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin 180 & 0 & \cos 180 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(c) Z-axis,

$$R_{fz} = R_z(180)$$

$$\begin{bmatrix} \cos 180 & -\sin 180 & 0 & 0 \\ \sin 180 & \cos 180 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(d) about xy-plane (z-axis)

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

(e) about xz plane (y-axis)

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

(f) about yz-plane (x-axis)

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

(3) Shearing :-

⇒ In 2D shearing about x-axis means x-value changes by amount proportional to y-values, remains same ~~the~~ y-value

⇒ In 3D, z-axis means x and y value changes by amount proportional to the z-value and z-value remains same.

Z-axis shear

$$x' = x + Sh_x \cdot z$$
$$y' = y + Sh_y \cdot z$$
$$z' = z$$

or,

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

X-axis shear

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

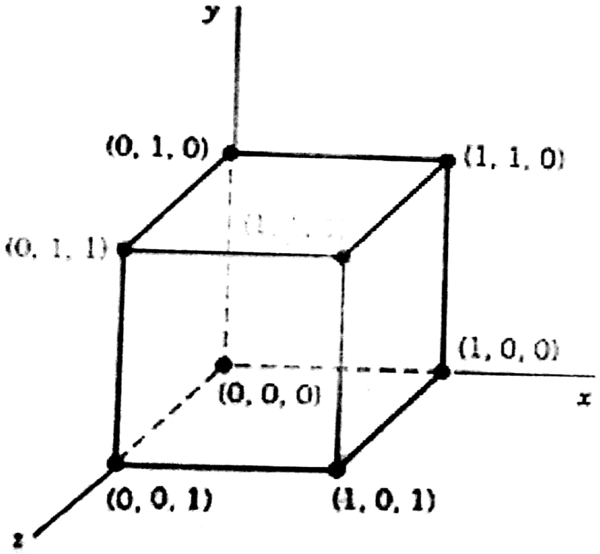
Y-axis shear

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

Scanned by CamScanner

Example

Find the new coordinates of a unit cube 90°-rotated about an axis defined by its endpoints A(2,1,0) and B(3,3,1).

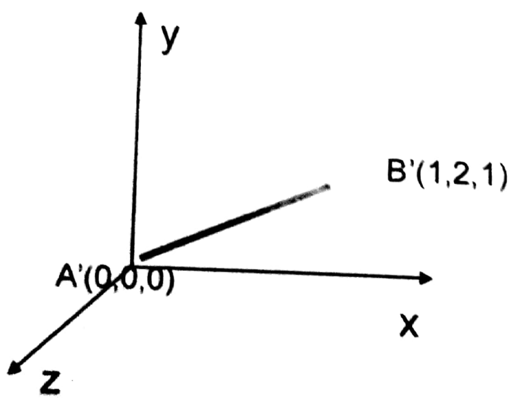


A Unit Cube

ASSIGNMENT

Example

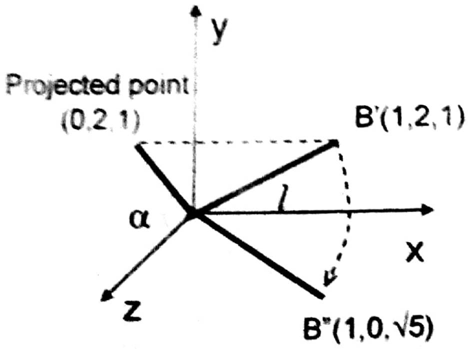
- Step1. Translate point A to the origin



$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example

- Step 2. Rotate axis $A'B'$ about the x axis by and angle α, until it lies on the xz plane.



$$\sin\alpha = \frac{2}{\sqrt{2^2+1^2}} = \frac{2}{\sqrt{5}} = \frac{2\sqrt{5}}{5}$$

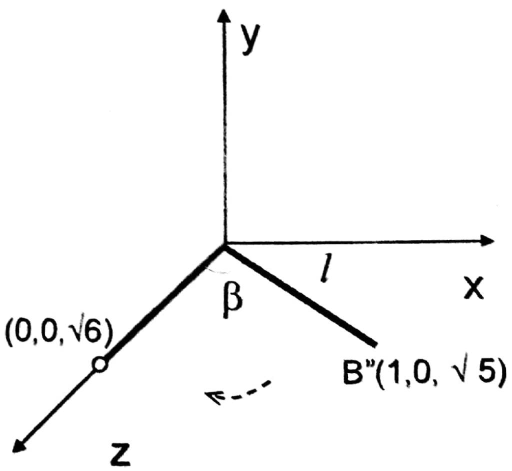
$$\cos\alpha = \frac{1}{\sqrt{5}} = \frac{\sqrt{5}}{5}$$

$$l = \sqrt{1^2+2^2+1^2} = \sqrt{6}$$

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{5}}{5} & -\frac{2\sqrt{5}}{5} & 0 \\ 0 & \frac{2\sqrt{5}}{5} & \frac{\sqrt{5}}{5} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example

- Step 3. Rotate axis $A'B''$ about the y axis by and angle β until it coincides with the z axis.



$$\sin\beta = \frac{1}{\sqrt{6}} = \frac{\sqrt{6}}{6}$$

$$\cos\beta = \frac{\sqrt{5}}{\sqrt{6}} = \frac{\sqrt{30}}{6}$$

$$\mathbf{R}_y(\beta) = \begin{bmatrix} \frac{\sqrt{30}}{6} & 0 & -\frac{\sqrt{6}}{6} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\sqrt{6}}{6} & 0 & \frac{\sqrt{30}}{6} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example

- Step 4. Rotate the cube 90° about the z axis

$$\mathbf{R}_z(90°) = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Finally, the concatenated rotation matrix about the arbitrary axis AB becomes,

$$\mathbf{R}(\theta) = \mathbf{T}^{-1}\mathbf{R}_x^{-1}(\alpha)\mathbf{R}_y^{-1}(\beta)\mathbf{R}_z(90°)\mathbf{R}_y(\beta)\mathbf{R}_x(\alpha)\mathbf{T}$$

Example

$$\mathbf{R}(\theta)=\begin{bmatrix}1&0&0&2\\0&1&0&1\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&\frac{\sqrt{5}}{5}&\frac{2\sqrt{5}}{5}&0\\0&-\frac{2\sqrt{5}}{5}&\frac{\sqrt{5}}{5}&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}\frac{\sqrt{30}}{6}&0&\frac{\sqrt{6}}{6}&0\\0&1&0&0\\-\frac{\sqrt{6}}{6}&0&\frac{\sqrt{30}}{6}&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}0&-1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$\begin{bmatrix}\frac{\sqrt{30}}{6}&0&-\frac{\sqrt{6}}{6}&0\\0&1&0&0\\\frac{\sqrt{6}}{6}&0&\frac{\sqrt{30}}{6}&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&\frac{\sqrt{5}}{5}&-\frac{2\sqrt{5}}{5}&0\\0&\frac{2\sqrt{5}}{5}&\frac{\sqrt{5}}{5}&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&-2\\0&1&0&-1\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$=\begin{bmatrix}0.166&-0.075&0.983&1.742\\0.742&0.667&0.075&-1.151\\-0.650&0.741&0.167&0.560\\0&0&0&1\end{bmatrix}$$

Example

- Multiplying $\mathbf{R}(\theta)$ by the point matrix of the original cube

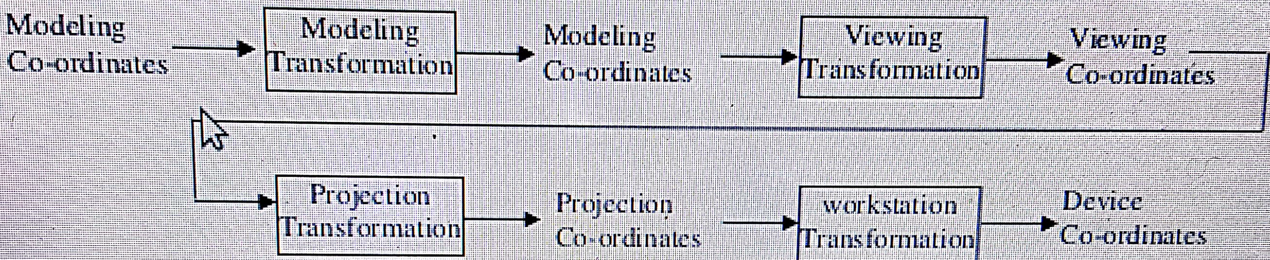
$$[\mathbf{P}'] = \mathbf{R}(\theta) \cdot [\mathbf{P}]$$

$$[\mathbf{P}'] = \begin{bmatrix} 0.166 & -0.075 & 0.983 & 1.742 \\ 0.742 & 0.667 & 0.075 & -1.151 \\ -0.650 & 0.741 & 0.167 & 0.560 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2.650 & 1.667 & 1.834 & 2.816 & 2.725 & 1.742 & 1.909 & 2.891 \\ -0.558 & -0.484 & 0.258 & 0.184 & -1.225 & -1.151 & -0.409 & -0.483 \\ 1.467 & 1.301 & 0.650 & 0.817 & 0.726 & 0.560 & -0.091 & 0.076 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

3D Viewing pipeline:

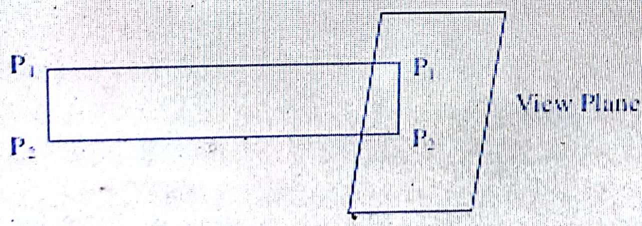
The steps for computer generation of a view of 3D scene are analogous to the process of taking photograph by a camera. For a snapshot, we need to position the camera at a particular point in space and then need to decide camera orientation. Finally when we snap the shutter, the seen is cropped to the size of window of the camera and the light from the visible surfaces is projected into the camera film.



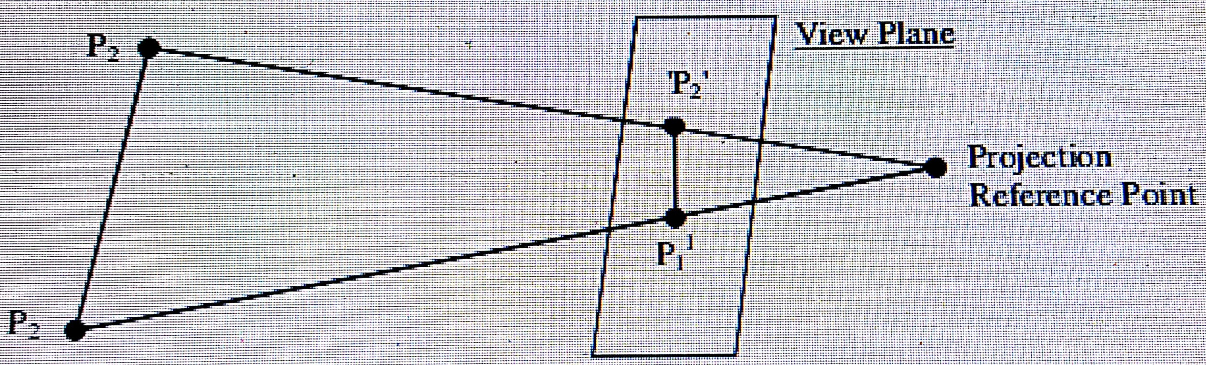
Projections:

Once world co-ordinate description of the objects in a scene are converted to viewing co-ordinates, we can project the three dimensional objects onto the two dimensional view plane. There are two basic projection methods:

Parallel projection: In parallel projection, co-ordinates positions are transformed to the view plane along parallel lines.



Prospective projection: In prospective projection, objects positions are transformed to the view plane along lines that converge to a point called projection reference point (centre of projection). The projected view of an object is determined by calculating the intersection of the projection lines with the view plane.



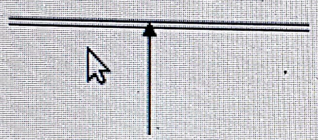
Prospective projection

A parallel projection preserve relative proportions of objects and this is the method used in drafting in drafting to produce scale drawing of three-dimensional objects. Accurate view of various sides of 3D object is obtained with parallel projection. But it does not given a realistic appearance of a 3D-object.

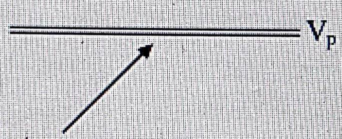
A prospective projection, on the other hand, produces realistic views but does not preserve relative proportions. Projections of distance objects from view plane are smaller than the projections of objects of the same size that are closer to the projection place.

Parallel Projection: We can specify parallel projection withy projection vector that specifies the direction of projection line. When the projection lines are perpendicular to view plane, the projection is orthographic parallel projections.

It projection line are not parallel to view plane then it is oblique parallel projection.

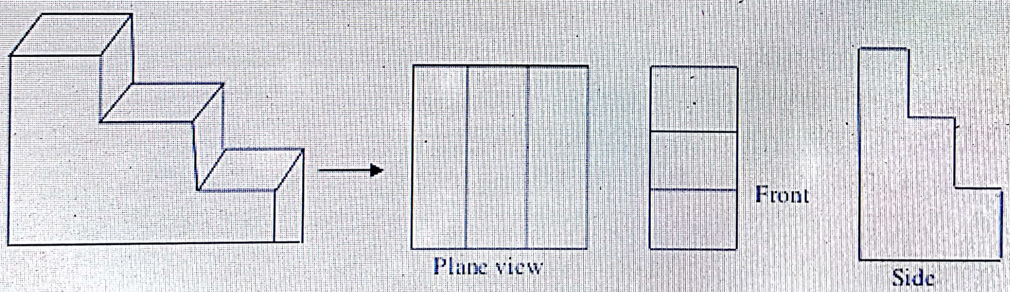


Orthographic projection



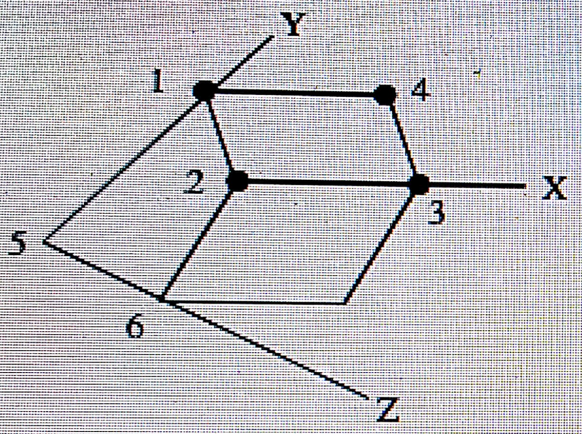
Oblique parallel projection

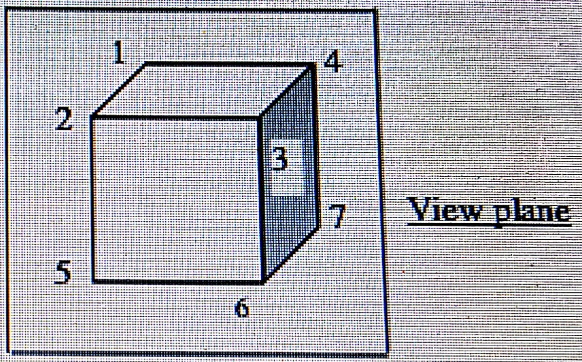
- Orthographic projections are most often used to produce the front, side, and top views of an object. Front, side and rear orthographic are called elevations and the top orthographic view of object is known as plan view. Engineering and Architectural drawings commonly employ these orthographic projections.



We also from orthographic projections that display more than one face of an object. Such views are called axonometric orthographic projections. the most commonly used axonometric projection is the isometric projection.

We also from orthographic projections that display more than one face of an object. Such views are called axonometric orthographic projections. the most commonly used axonometric projection is the isometric projection.





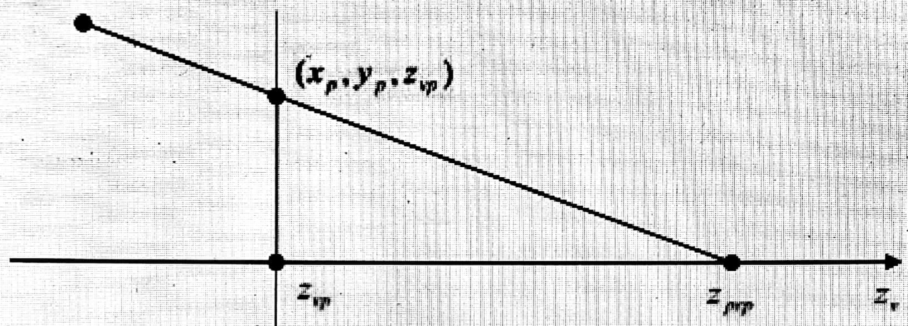
The transformation eq^n for orthographic projection is

$x_p = x,$ $\quad y_p = y,$ $\quad z$ – Coordinate value is preserved for the depth information:

Prospective projections:

To obtain a prospective projection of a three-dimensional object, we transform points along projection lines that meet at a point called projection reference point.

Suppose we set the projection reference point at position Z_{prp} along the Z_v axis, and we place the view plane at Z_{vp} as shown is fig



Prospective projection of a point $P(x, y, z)$ to position (x_p, y_p, z_{vp}) on the view plane.

We can write equations describing co-ordinates positions along this prospective projection line in parametric form as

$$x' = x - xu$$
$$y' = y - yu$$
$$z' = z - (z - z_{prp})u$$

Where u takes value from $0 to 1$. If $u = 0$, we are at position $P = (x, y, z)$. If $u = 1$, we have projection reference point $(0, 0, z_{prp})$. On the view plane, $z' = z_{vp}$ then

$$u = \frac{z_{vp} - z}{z_{prp} - z}$$

Substituting these value in eq^n for x', y'.

$$x_p = x - x\left(\frac{z_{vp} - z}{z_{prp} - z}\right) = x\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right) = x\left(\frac{dp}{z_{prp} - z}\right)$$

Similarly,

$$y_p = y\left(\frac{z_{prp} - z_{vp}}{z - z_{prp}}\right) = y\left(\frac{dp}{z_{prp} - z}\right)$$

Where $dp = z_{prp} - z_{vp}$ is the distance of the view plane from projection reference point.

Using 3-D homogeneous Co-ordinate representation, we can write prospective projection transformation matrix as

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & z_{vp}/dp & \dfrac{-z_{vp}}{z_{prp}/dp} \\ 0 & 0 & 1/dp & z_{prp}/dp \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

In this representation the homogeneous factor $h = \frac{z - z_{prp}}{dp}$

Projection co-ordinates,

$$xp = xh/h, \; yp = yh/h$$

There are special case for prospective transformation.

When $z_{prp} = 0$;

$$x_p = x\left(\frac{z_{prp}}{z_{prp} - z}\right) = x\left(\frac{1}{1 - z/z_{prp}}\right)$$

$$y_p = y\left(\frac{z_{prp}}{z_{prp} - z}\right) = y\left(\frac{1}{1 - z/z_{prp}}\right)$$

Some graphics package, the projection point is always taken to be viewing co-ordinate origin. In this ease,' $z_{prp} = 0$

$$\therefore x_p = x\left(\frac{zvp}{z}\right) = x\left(\frac{1}{z/zvp}\right)$$

$$yp = y\left(\frac{zvp}{z}\right) = y\left(\frac{1}{z/zvp}\right)$$

Scanned by CamScanner

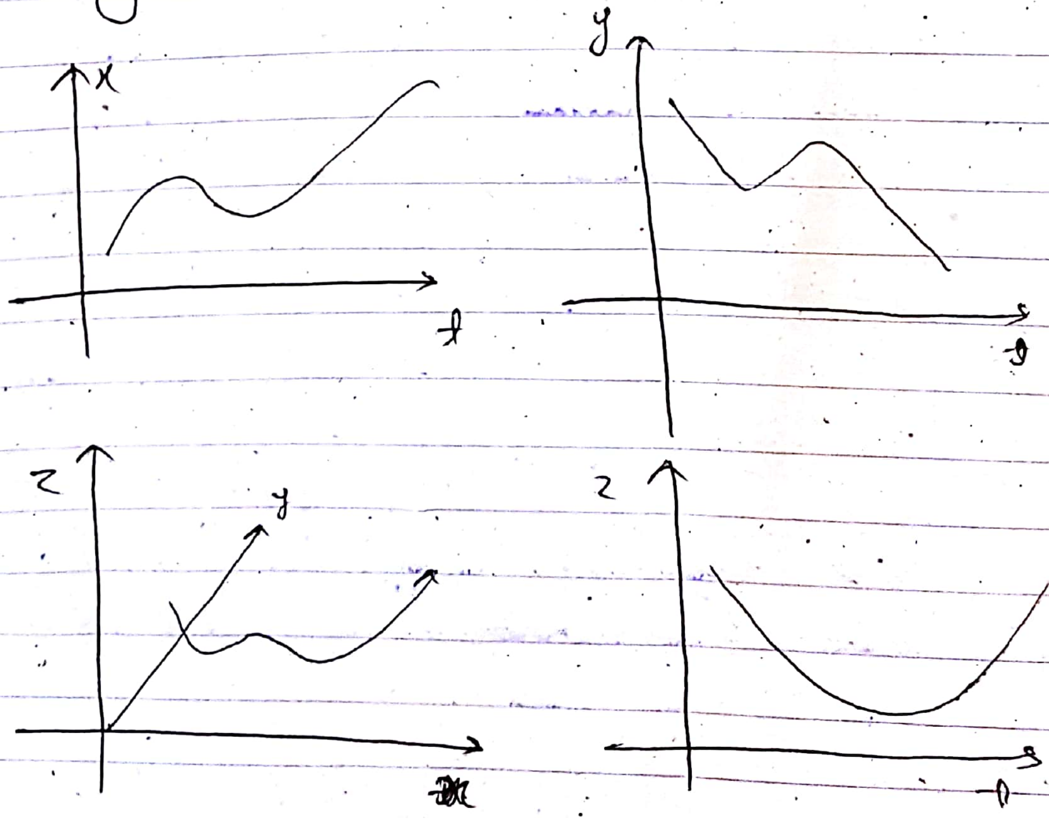
Parametric Cubic Curves:

Cubic Curves are Commonly used in graphics because Curves of lower order Commonly have too little flexibility, While Curves of higher order are usually Considered unnecessarily complex and make it easy to introduce undesired wiggles.

A Parametric Cubic Curve in 3D is defined by,

$$x(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$
$$y(t) = b_3 t^3 + b_2 t^2 + b_1 t + b_0$$
$$z(t) = c_3 t^3 + c_2 t^2 + c_1 t + c_0$$

Usually, we Consider $t = [0 .. 1]$



A Compact version of the parametric eqns can be written as,

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}$$

$\therefore$ $x(t) = T.A$

$||^{ly}$, we can write,

$$y(t) = TB$$
$$z(t) = TC$$

4. Spline Representation

A Spline is a flexible strips used to produce smooth curve through a designated set of points. A curve drawn with these set of points is spline curve. Spline curves are used to model 3D object surface shape smoothly.

Mathematically, spline are described as pice-wise cubic polynomial functions. In computer graphics, a spline surface can be described with two set of orthogonal spline curves. Spline is used in graphics application to design and digitalize drawings for storage in computer and to specify animation path. Typical CAD application for spline includes the design of automobile bodies, aircraft and spacecraft surface etc.

Interpolation and approximation spline

- Given the set of control points, the curve is said to interpolate the control point if it passes through each points.
- If the curve is fitted from the given control points such that it follows the path of control point without necessarily passing through the set of point, then it is said to approximate the set of control point.

Bezier curve and surface

This is spline approximation method, developed by the French Engineer Pierre Bezier for use in the design of automobile body. Beizer spline has a number of properties that make them highly useful and convenient for curve and surface design. They are easy to implement. For this reason, Bezier spline is widely available in various CAD systems.

In general Bezier curve can be fitted to any number of control points. The number of control points to be approximated and their relative position determine the degree of Bezier polynomial. The Bezier curve can be specified with boundary condition, with characterizing matrix or blending functions. But for general blending function specification is most convenient.

Suppose we have n+1 control points: $p_k(x_k,y_k,z_k)$, k = 0,1,2,3,4,.........n. These co-ordinate points can be blended to produce the following position vector p(u) which describes path of an approximating Bezier polynomial function p_0 and p_n.

$$p(u) = \sum_{k=0}^{n} p_k.BEZ_{k,n}(u), \qquad 0\leq u\leq 1 \text{ ------------------------- } 1$$

The Bezier belending function $BEZ_{k,n}(u)$ are the Bernstein polynomial as,

$$BEZ_{k,n}(u) = c(n,k)u_k(1-u)^{n-k}$$

The vector equation (1) represents a set of three parametric equation for individual curve conditions.

i.e.
$$x(u) = \sum_{k=0}^{n} x_k.BEZ_{k,n}(u)$$

$$y(u) = \sum_{k=0}^{n} y_k.BEZ_{k,n}(u)$$

$$z(u) = \sum_{k=0}^{n} z_k.BEZ_{k,n}(u)$$

Bezier curve is a polynomial of degree one less than control points i.e. 3 points generate parabola, 4 points a cubic curve and so n.

Properties of Bezier Curve:

1. It always passs through initial and final control points. i.e $p(0) = p_0$ and $p(1)=p_n$.

2. Values of the parametric first derivatives of a Bezier curve at the end points can be calculated from control points as-

$$p'(0) = -np_0 + np_1$$
$$p'(1) = -np_{n-1} + np_n$$

3. The slope at the beginning of the curve is along the line joining the first two points and slope at the end of curve is along the line joining last two points.

4. Parametric second derivative at a Bezier curve at end points are-

$$p''(0) = n(n-1)[(p_2-p_1) - (p_1-p_0)]$$
$$p''(1) = n(n-1)[(p_{n-2}-p_{n-1}) - (p_{n-1}-p_n)]$$

Derivation

1. Derive the equation for Quadratic Bezier Curve.

Soln:

we know,

$$P(u) = \sum_{k=0}^{n} P_k \cdot BEZ_{k,n}(u) \quad 0 \le u \le 1 \quad \text{— (a)}$$

where,

$$BEZ_{k,n}(u) = c(n,k)\, u^{k} (1-u)^{n-k} \quad \text{— (b)}$$

ie,

$$x(u) = \sum_{k=0}^{n} x_k\, BEZ_{k,n}(u) \quad \text{— (I)}$$

$$y(u) = \sum_{k=0}^{n} y_k \cdot BEZ_{k,n}(u) \quad \text{— (II)}$$

$$z(u) = \sum_{k=0}^{n} z_k \cdot BEZ_{k,n}(u) \quad \text{— (III)}$$

Now,

for quadratic eqn $n = 2$

from (a)

$$P(u) = \sum_{k=0}^{2} P_k\, BEZ_{\cancel{k,n}\,2}(u)$$

$= P_0\, BEZ_{0,2}(u) + P_1\, BEZ_{1,2}(u) + P_2\, BEZ_{2,2}(u)$ —— (c)

from (b),

$$BEZ_{0,2}(u) = \underbrace{C(2,0)}\, u^0 (1-u)^2 = 1 \times (1-u)^2 = (1-u)^2$$

$$BEZ_{1,2}(u) = \underbrace{C(2,1)}\, u^1 (1-u)^{2-1} = 2u(1-u)^{2-1} = 2u(1-u)$$

$$BEZ_{2,2}(u) = \underbrace{C(2,2)}\, u^2 (1-u)^0 = 1 \times u^2 = u^2$$

—— (d)

from (c) & (d),

$$\boxed{P(u) = (1-u)^2 P_0 + 2u(1-u) P_1 + u^2 P_2} \quad \text{—— (e)}$$

Illy,

Q. Derive the eqn for cubic Bezier Curve.

Ans: $P(u) = (1-u)^3 P_0 + 3(1-u)^2 u P_1 + 3(1-u) u^2 P_2 + u^3 P_3$

$0 \leq u \leq 1$

Q1 Construct Bezier Curve for Control points $\underbrace{(4,2)}_{P_0}$ $\underbrace{(8,8)}_{P_1}$ and $\underbrace{(16,4)}_{P_2}$

Sl^n:-

$$P(u) = \sum_{k=0}^{2} P_k \cdot B_{k,2}(u) \quad , \quad 0 \le u \le 1$$

we know, By solving this eqn we get,

$$P(u) = (1-u)^2 P_0 + 2u(1-u) P_1 + u^2 P_2 \quad \text{--- (i)}$$

Now,

$$P(u) = (1-u)^2 (4,2) + 2u(1-u)(8,8) + u^2 (16,4)$$

$$= \left((1-u)^2 \times 4 + 2u(1-u) \times 8 + u^2 \times 16 \; , \; (1-u)^2 \times 2 + 2u(1-u) \times 8 + u^2 \times 4 \right)$$

solve and get

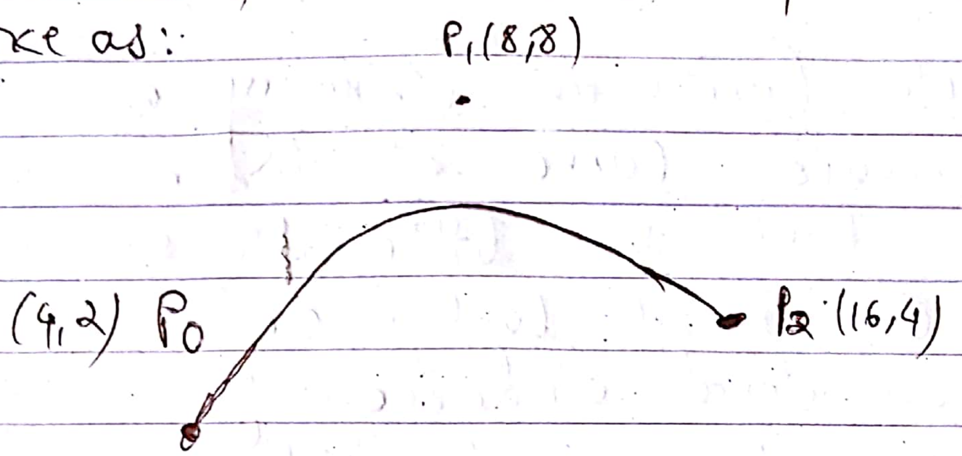
$$= \left(\underbrace{4u^2 + 8u + 4}_{x(u)} \; , \; \underbrace{-10u^2 + 12u + 2}_{y(u)} \right)$$

we know, $0 \le u \le 1$

If

u	x(u)	y(u)	
u = 0	4	2	⇒ P_0
u = 0.2	5.76	4	
u = 0.4	7.84	5.20	
u = 0.6	10.24	5.6	
u = 0.8	12.96	5.2	
u = 1	16	4	⇒ P_2

Plot all $(x(u), y(u))$ point in graph. expected outcomes may look like as:



Imp Q, The Cordinates of four controll points relative to Curve are given by $P_1(2,2,0)$, $P_2(2,3,0)$ $P_3(3,3,0)$ $P_4(3,2,0)$ write eqn of Bezier Curve. Also find cordinate pixels of curve for $u = 0, 1/4, 1/2, 3/4, 1$ and plot on the graph.

Geometric and Parametric Continuity

Geometric Continuity

- G0: curves are joined
- G1: first derivatives are proportional at the join point
  The curve tangents thus have the same direction, but not necessarily the same magnitude. i.e., C1'(1) = (a,b,c) and C2'(0) = (k*a, k*b, k*c).
- G2: first and second derivatives are proportional at join point

Parametric Continuity

- C0: curves are joined
- C1: first derivatives equal
- C2: first and second derivatives are equal
  If t is taken to be time, this implies that the acceleration is continuous.
- Cn: nth derivatives are equal

Cubic spline:

It is most often used to set up path for object motions o tot provide a representation for an existing object or drawing. To design surface of 3D object any spline curve can be represented by piece-wise cubic spline.

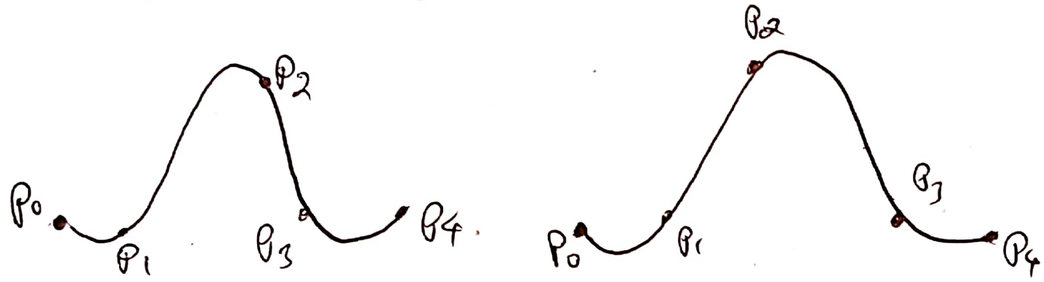
Cubic polynomial offers a reasonable compromise between flexibility and speed of computation. Cubic spline requires less calculations with comparison to higher order polynomials and require less memory. Compared to lower order polynomial cubic spline are more flexible for modeling arbitrary curve shape.

Given a set of control points, cubic interpolation spines are obtained by fitting the input points with a picewise cubic polynomial curve that passes through every control points.

Suppose we have n+1 control points specified with co-ordinates.

$p_k = (x_k, y_k, z_k)$, $k = 0,1,2,3\ldots\ldots\ldots n$

A cubic interpolation fit of those points is



We can describe the parametric cubic polynomial that is to be fitted between each pair of control points with the following set of parametric equations.

$$x(u) = a_x u^3 + b_x u^2 + c_x u + d_x$$
$$y(u) = a_y u^3 + b_y u^2 + c_y u + d_y \qquad (0 \le u \le 1)$$

$z(u) = a_z u^3 + b_z u^2 + c_z u + d_z$

There are three equivalent methods for specifying a particular spline representation.

1. Set of boundary conditions.

   For the parametric cubic polynomial for the x-coordinate along the path of spline section

   $x(u) = a_x U^3 + b_x U^2 + c_x U + d \qquad 0 \le U \le 1$

   Boundary condition for this curve be the set on the end point coordinate x(0) and x(1) and in the first derivatives at end points x'(0) and x'(1). These four boundary condition are sufficient to determine the four coefficient a_x, b_x, c_x, d_x.

2. From the boundary condition, we can obtain the characterizing matrix for spline. Then the parametric equation can be written as-

$$x(u) = [\ u3 \ \ u2 \ \ u \ \ 1] \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix}$$

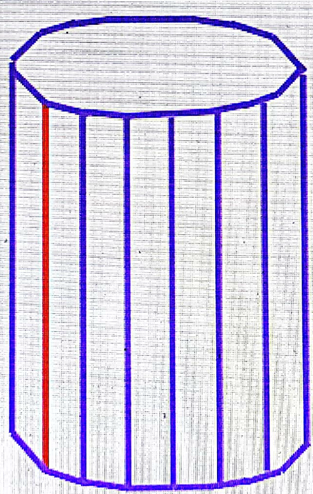
3. Blending function that determines how specified geometric constraints on the curve are combined to calculate position along the curve path.

$$x(u) = \sum_{k=0}^{3} g_k . BF_k(u)$$

   Where g_k are the geometric constraint parameters such as control points co-ordinate and slope of the curve at control point. $BF_k(u)$ are the polynomial Blending functions.

3-D OBJECT REPRESENTATION

- Most commonly used representation for 3-d object
- Speeds up the surface rendering
- wired frame applications can be displayed quickly to give a general indication of surface structure.
- Greater the number of smaller surfaces, better the approximations



Polygon Surface Approximation of Cylinder

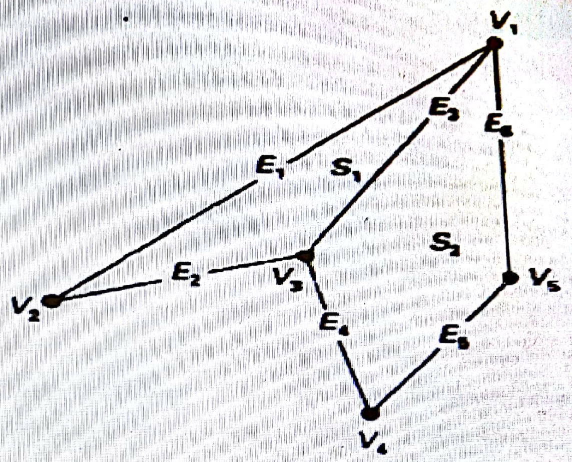
3-D OBJECT REPRESENTATION

Data Requirements for Polygon Surface Representation

- Polygon data tables– to store information
- Polygon Data Tables consists of
  - Geometric Tables- Consist of parameters that specify polygon vertices and spatial orientation of polygons
    - Vertex Table-stores the coordinates of vertices
    - Edge Table-contains the pointer back to the vertex table to identify the vertices for each polygon edge.
    - Polygon Table-contains pointer back into the edge table to identify the edges for each polygon
  - Attributes Tables- Consist of parameters that specify transparency, surface texture, color etc.

3-D Object Representation

- Convenient Organizations for storing geometric data is to create three lists
  - Vertex Table
  - Edge Table
  - Polygon Surface Table



VERTEX TABLE	
V_1:	x_1, y_1, z_1
V_2:	x_2, y_2, z_2
V_3:	x_3, y_3, z_3
V_4:	x_4, y_4, z_4
V_5:	x_5, y_5, z_5

EDGE TABLE	
E_1:	V_1, V_2
E_2:	V_2, V_3
E_3:	V_3, V_1
E_4:	V_3, V_4
E_5:	V_4, V_5
E_6:	V_5, V_1

POLYGON-SURFACE TABLE	
S_1:	E_1, E_2, E_3
S_2:	E_3, E_4, E_5, E_6

Guidelines to Generate Error Free Table

1. Every vertices listed as an end point for at least two edges
2. Every edge is part of at least one polygon
3. Every polygon is closed.
4. Every polygon has at least one shared edge

If edge table contains pointers to polygons, every edge referenced by a polygon pointer has a reciprocal pointer back to polygon

Spatial Orientation of Surface

- Orientation of surface normal
- How to find surface normal if surface vertices (x_1,y_1,z_1), (x_2,y_2,z_2) and (x_3,y_3,z_3) are given ?
  - Plane equation:

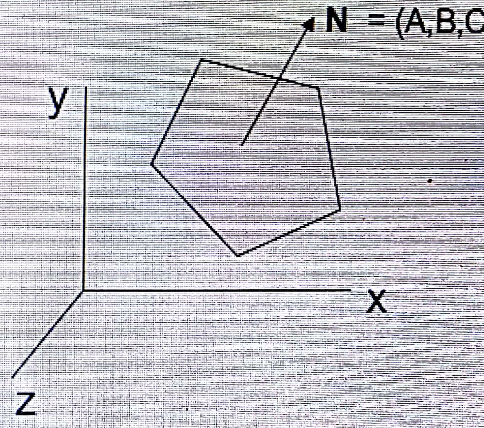
$$Ax + By + Cz + D = 0$$

  - Put the vertex coordinates (x_1,y_1,z_1), (x_2,y_2,z_2) and (x_3,y_3,z_3) in plane equation to get

Solution obtained from cramer' s rule

$$A = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix} \qquad B = \begin{vmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{vmatrix}$$

$$C = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \qquad D = -\begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$



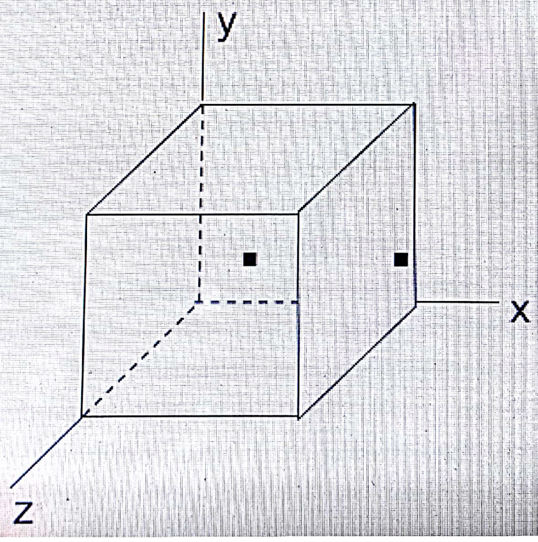
- Surface Normal **N** to the plane Ax+By+Cz+D=0 has the Cartesian components (A,B,C)

Two sides of a surface (Inside or Outside the Plane Surface)

- For right handed cartesian system (i.e. if vertices are taken in Counter-Clockwise order to evaluate A,B,C and D)

if $Ax + By + Cz + D < 0$, the point (x, y, z) is inside the surface

if $Ax + By + Cz + D > 0$, the point (x, y, z) is outside the surface



Visible Surface Detection Methods

(Hidden surface elimination)

Visible surface detection or Hidden surface removal is major concern for realistic graphics for identifying those parts of a scene that are visible from a choosen viewing position. Several algorithms have been developed. Some require more memory, some require more processing time and some apply only to special types of objects.

Visible surface detection methods are broadly classified according to whether they deal with objects or with their projected images.

These two approaches are

- **Object-Space methods:** Compares objects and parts of objects to each other within the scene definition to determine which surface as a whole we should label as visible.

- **Image-Space methods**: Visibility is decided point by point at each pixel position on the projection plane.

Most visible surface detection algorithm use image-space-method but in some cases object space methods are also used for it.

BACK-FACE DETECTION(Plane Equation method)

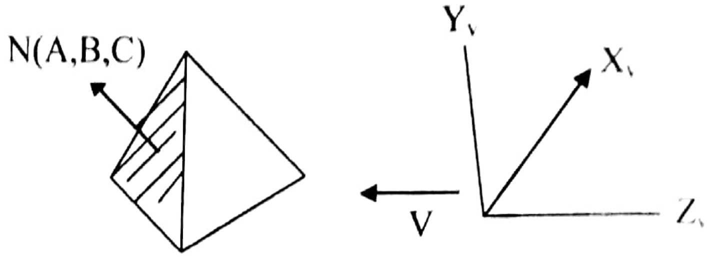
A fast and simple object space method used to remove hidden surface from a 3D object drawing is known as "Plane equation method" and applied to each side after any rotation of the object takes place. It is commonly known as back-face detection of a polyhedron is based on the "inside-outside" tests.

A point (x, y, z) is inside a polygon surface if

$Ax + By + Cz + D < O$

We can simplify this test by considering the normal vector N to a polygon surface which has Cartesian components (A,B,C)
If V is the vector in viewing direction from the eye position then this polygon is a back face if,

V.N>0



In the equation $Ax+By+Cz+D=0$, if A,B,C remains constant, then varying value of D results in a whole family of parallel planes. One of which (D = 0) contains the origin of the co-ordinates system and ,
If D > 0, plane is behind the origin(Away from observer)
If D < 0, plane is in front of origin(towards the observer)

If we clearly defined our object to have centered at origin, the all those surface that are viewable will have negative D and unviewable surface have positive D.

So , simply our hidden surface removal routine defines the plane corresponding to one of 3D surface from the co-ordinate of 3 points on it and computing D , visible surface are detected.

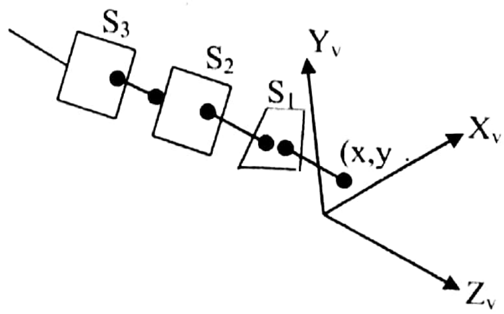
DEPTH-BUFFER-METHOD:
Depth Buffer Method is the commonly used image-space method for detecting visible surface. It is also know as z-buffer method. It compares surface depths at each pixel position on the projection plane. It is called z-buffer method since object depth is usually measured from the view plane along the z-axis of a viewing system.

Each surface of scene is processed separately, one point at a time across the surface. The method is usually applied to scenes containing only polygon surfaces, because depth values can be computed very quickly and method is easy to implement. This method can be apply to non planer surfaces.

With object description converted to projection co-ordinates, each (x,y,z) position on polygon surface corresponds to the orthographic projection point (x,y) on the view plane. Therefore for each pixel position (x,y) on the view plane, object depth is compared by z. values.

With objects description converted to projection co-ordinates, each (X,Y,Z) position on polygon surface correspond to the orthographic projection point (X,Y) on the view plane. The object depth is compared by Z-values.

In figure, three surface at varying distance from view plane X_vY_v, the projection along (x,y) surface S_1 is closest to the view-plane so surface intensity value of S_1 at (x,y) is saved.



In Z-buffer method, two buffers area are required. A depth buffer is used to store the depth value for each (x,y) position or surface are processed, and a refresh buffer stores the intensity value for each position. Initially all the position in depth buffer are set to 0, and refresh buffer is initialize to background color. Each surface listed in polygon table are processed one scan line at a time, calculating the depth (z-val) for each position (x,y). The calculated depth is compared to the value previously stored in depth buffer at that position. If calculated depth is greater than stored depth value in depth buffer, new depth value is stored and the surface intensity at that position is determined and placed in refresh buffer.

Algorithm: Z-buffer

1. Initialize depth buffer and refresh buffer so that for all buffer position (x,y)
   depth (x,y) =0, refresh (x,y) = $I_{background}$.
2. For each position on each polygon surface, compare depth values to previously stored value in depth buffer to determine visibility.
   - Calculate the depth Z for each (x,y) position on polygon
   - If Z> depth (x,y) then
     depth (x,y) =Z
     refresh (x,y) = $I_{surface}$ (x,y)

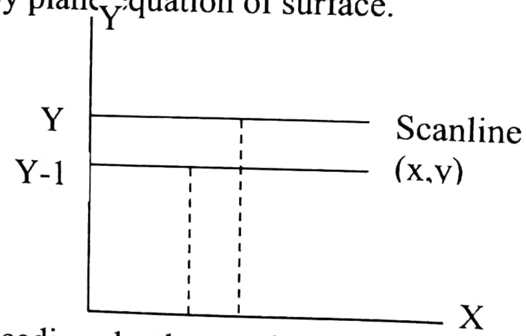
Where $I_{background}$ is the intensity value for background and $I_{surface}$ (x,y) is intensity value for surface at pixel position (x,y) on projected plane. After all surface are processed, the depth buffer contains the depth value of the visible surface and refresh buffer contains the corresponding intensity values for those surface. The depth value of the surface position (x,y) are calculated by plane equation of surface.

$$Z = \frac{-Ax - By - D}{C}$$

Let Depth Z' at position (x+1,y)

$$Z' = \frac{-A(x+1) - By - D}{C}$$

$$\Rightarrow Z' = Z - \frac{A}{C} \qquad (1)$$

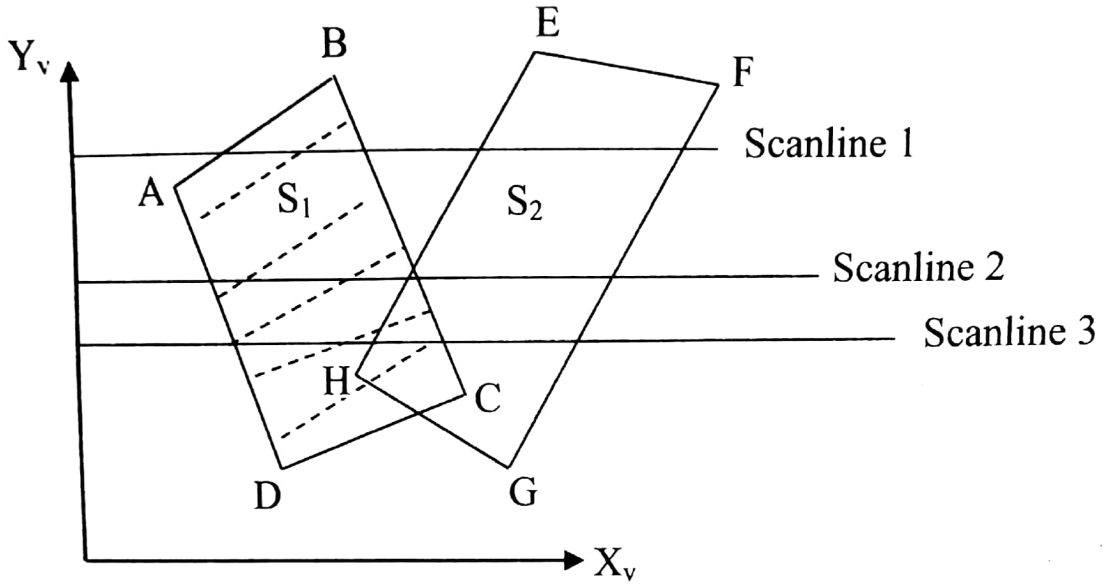


$-\frac{A}{C}$ is constant for each surface so succeeding depth x ax+1 across a scan line are obtained form preceding values by simple calculation.

SCAN LINE METHOD :

This is image-space method for removing hidden surface which is extension of the scan line polygon filling for polygon interiors. Instead of filling one surface we deal with multiple surface here.

As each scan line is processed, all polygon surface intersecting that line are examined to determine which are visible. We assume that polygon table contains the co-efficient of the plane equation for each surface as well as vertex edge, surface information, intensity information for the surface, and possibly pointers to the edge table.



- In figure above, the active edge list for scan line 1 contains information from edge table for edge AB, BC, EH, FG.
- For positions along this scan line between edge AB and BC, only the flag for surface S_1 is on
- Therefore no depth calculation must be made using the plane coefficients for two surface and intensity information for surface S_1 is entered from the polygon table into the refresh buffer.

→ Similarly between EH&FG. Only the flag for S_2 is on. No other positions along scan line 1 intersect surface. So intensity values in the other areas are set to background intensity

→ For Scanline 2 & 3, the active edge list contains edges AD, EH BC, FG. Along scaline 2 from edge AD, to edge EH only surface flag for S_1 is on, but between edges EH& BC, the flags for both surface is on. In this interval, depth calculation is made using the plane coefficients for the two surfaces.

For example, if Z of surface S_1 is less than surface S_2, So the intensity of S_1 is loaded into refresh buffer until boundary BC is encountered. Then the flag for surface S_1 goes off and intensities for surface S_2 are stored until edge FG is passed.

Any no of overlapping surface are processed with this scan line methods.

Illumination and Surface Rendering:

- Realistic displays of a scene are obtained by perspective projections and applying natural lighting effects to the visible surfaces of object.
- An illumination model is also called lighting model and some times called as a shading model which is used to calculate the intensity of light that we should see at a given point on the surface of a object.
- A surface-rendering algorithm uses the intensity calculations from an illumination model.

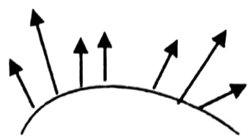
Light Sources:

- Sometimes light sources are referred as light emitting object and light reflectors. Generally light source is used to mean an object that is emitting radiant energy e.g. Sun.

Point Source: Point source is the simplest light emitter e.g. light bulb.

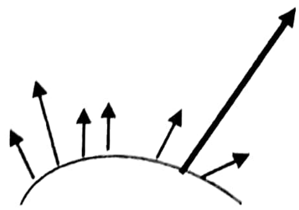
Distributed light source: Fluorescent light

- When light is incident on an opaque surface part of it is reflected and part of it is absorbed.
- Surface that are rough or grainy, tend to scatter the reflected light in all direction which is called diffuse reflection.



Diffuse reflection

- When light sources create highlights, or bright spots, called specular reflection



Specuular reflection

Illumination models:

Illumination models are used to calculate light intensities that we should see at a given point on the surface of an object. Lighting calculations are based on the optical properties of surfaces, the background lighting conditions and the light source specifications. All light sources are considered to be point sources, specified with a co-ordinate position and an intensity value (color). Some illumination models are:

1. **Ambient light**:

- This is a simplest illumination model. We can think of this model, which has no external light source-self-luminous objects. A surface that is not exposed directly to light source still will be visible if nearby objects are illuminated.
- The combination of light reflections form various surfaces to produce a uniform illumination is called ambient light or background light.
- Ambient light has no spatial or directional characteristics and amount on each object is a constant for all surfaces and all directions. In this model, illumination can be expressed by an illumination equation in variables associated with the point on the object being shaded. The equation expressing this simple model is

$$I = K_a$$

Where I is the resulting intensity and K_a is the object's intrinsic intensity.

If we assume that ambient light impinges equally on all surface from all direction, then

$$I = I_a K_a$$

Where I_a is intensity of ambient light. The amount of light reflected from an object's surface is determined by K_a, the ambient-reflection coefficient.

K_a ranges from 0 to 1.

2. Diffuse reflection:

Objects illuminated by ambient light are uniformally illuminated across their surfaces even though light are more or less bright in direct proportion of ambient intensity. Illuminating object by a point light source, whose rays enumerate uniformly in all directions from a single point. The object's brightness varies form one part to another, depending on the direction of and distance to the light source.

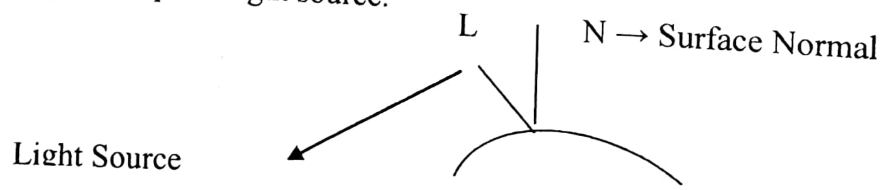
- The fractional amount of the incident light that is diffusely reflected can be set for each surface with parameter K_d, the coefficient of diffuse-reflection.
- Value of K_d is in interval 0 to 1. If surface is highly reflected, K_d is set to near 1. The surface that absorbs almost incident light, K_d is set to nearly 0.
- Diffuse reflection intensity at any point on the surface if exposed only to ambient light is

$$\mathrm{Im}\,abdiff = I_d K_d$$

- Assuming diffuse reflections from the surface are scattered with equal intensity in all directions, independent of the viewing direction (surface called. "Ideal diffuse reflectors") also called Lambertian reflectors and governed by Lambert's cosine law.

$$Idiff = K_d I_l \cos\theta$$

Where I_l is the intensity of the point light source.



It N is unit vector normal to the surface & L is unit vector in the direction to the point slight source then

$$I_l, diff = K_d I_l (N.L)$$

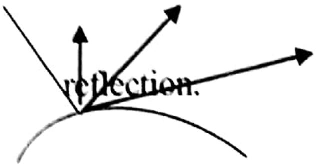
In addition, many graphics packages introduce an ambient reflection coefficient K_a to modify the ambient-light intensity I_a

$$Idiff = K_a I_a + K_d I_l (N.L)$$

3. Specular reflection and pong model.

When we look at an illuminated shiny surface, such as polished metal, a person's forehead, we see a highlight or bright spot, at certain viewing direction. Such phenomenon is called specular reflection. It is the result of total or near total reflection of the incident light in a concentrated region around the " specular reflection angle = angle of incidence".

Let SR angle = angle of incidence as in figure.



N - unit vector normal to surface at incidence point

R - unit vector in the direction of ideal specular reflection.

L - unit vector directed to words point light source.

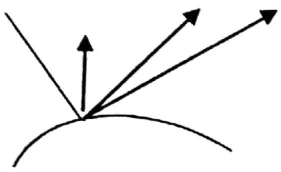
V - unit vector pointing to the viewer from surface.

$\phi -$ the viewing angle relative to the specular reflection direction.

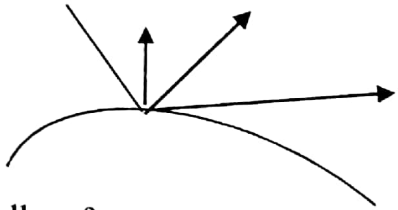
For ideal reflector (perfect mirror), incident light is reflected only in the specular reflection direction.

$\therefore V$ & R concide ($\phi = 0$)

- Shiny surface have narrow ϕ and dull surface wider ϕ.
- An empirical model for calculating specular-reflection range developed by Phong Bui Tuong-called "Phong specular reflection model (or simply Phong model), sets the intensity of specular reflection proportional to $\cos^{ns}\phi \rightarrow 0 to 90^0$.
- Specular reflection parameter n_s is determined by type of surface
- Very shiny surface with large value n_s (say 100 or more) and dull surface , smaller n_s (down to 1)
- Rough surface such as chalk , $n_s = 1$.



Shhiny surface
(large n_s)



Dull surface
(small n_s)

- Intensity of specular reflection depends upon material properties of the surface and θ. Other factors such as the polarization and color of the incident light.
- For monochromatic specular intersity variations can approximated by SR coefficient $w(\theta)$

- Fresnal's law of reflection describe specular reflection intensity with θ and using $w(\theta)$, Phong specular reflection model as

$$I_{spec} = w(\theta) I_l \cos^{n_s} \phi$$

Where I_l is intensity of light source. ϕ is viewing angle relative to SR direction R.
For a glass, we can replace $w(\theta)$ with constant K_s specular reflection coefficient.

So , $I_{spec} = K_s I_l \cos^{n_s} \phi$

$= K_s I_l (V.R)^{n_s}$ Since $\cos\phi = V.R$

Polygon (surface) Rendering Method

- Application of an illumination model to the rendering of standard graphics objects those formed with polygon surfaces are key technique for polygon rendering algorithm.
- Calculating the surface normal at each visible point and applying the desired illumination model at that point is expensive. We can describe more efficient shading models for surfaces defined by polygons and polygon meshes.
- Scan line algorithms typically apply a lighting model to obtain polygon surface rendering in one of two ways. Each polygon can be rendered with a single intensity, or the intensity can be obtained at each point of the surface using an interpolating scheme.

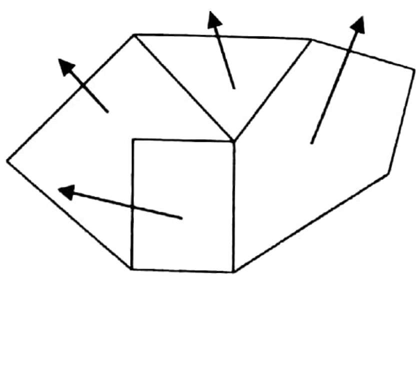
1. **Constant Intensity Shading**: (Flat Shading)

The simplest model for shading for a polygon is constant intensity shading also called as **Faceted Shading** Or flat shading. This approach implies an illumination model once to determine a single intensity value that is then used to render an entire polygon. Constant shading is useful for quickly displaying the general appearance of a curved surface.

This approach is valid if several assumptions are true:

1. The light source is at infinity, so **N.L** is constant across the polygon face.
2. The viewer is at infinity, so **N.V** is constant across the polygon face.
3. The polygon represents the actual surface being modeled and is not an approximation to a curved surface.

Even if all conditions are not true, we can still reasonably approximate surface -lighting effects using small polygon facets with fast shading and calculate the intensity for each facet, at the centre of the polygon of course constant shading does not produce the variations in shade a cross the polygon that should occur.



2. **Interpolated Shading**:

An alternative to evaluating the illumination equation at each point on the polygon, we can use the interpolated shading, in which shading information is linearly interpolated across a triangle from the values determined for its vertices. Gouraud generalized this technique for arbitrary polygons. This is particularly easy for a scan line algorithm that already interpolates the z- value across a span from interpolated z-values computed for the span's endpoints.

Gouraud Shading:

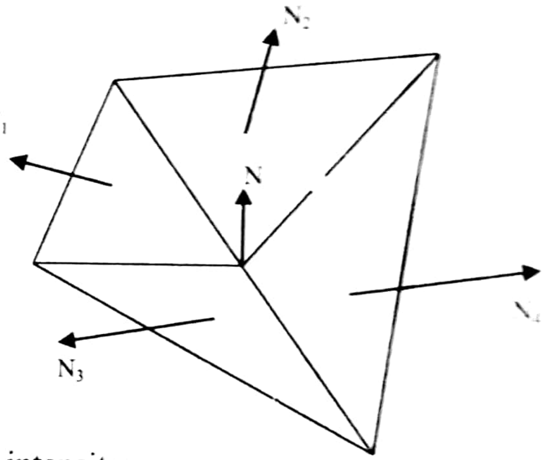
Gouraud shading , also called intensity interpolating shading or color interpolating shading, eliminates intensity discontinuities that occur in flat shading. Each polygon surface is rendered with Gouraud shading by performing following calculations.

1. Determine the average unit normal vector at each vertex. At each polygon vertex, we obtain a normal vertex by averaging the surface normals of all polygons sharing the vertex as:

$$N_v = \frac{\sum_{k=1}^{n} N_k}{|\sum_{k=1}^{n} N_k|}$$

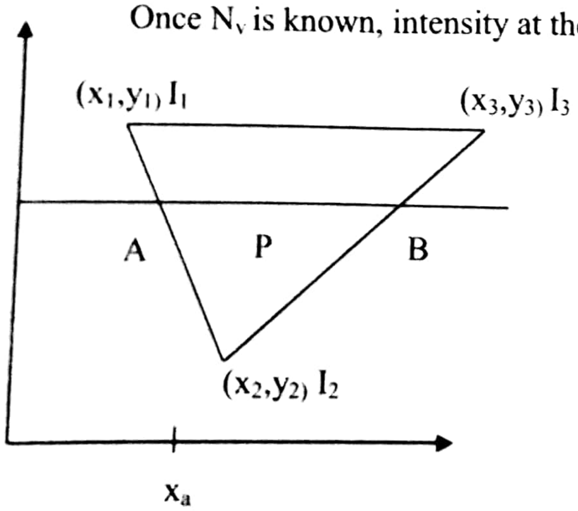
Here $N_v = \dfrac{N_1 + N_2 + N_3 + N_4}{|N_1 + N_2 + N_3 + N_4|}$

Where N_v is normal vector at a vertex sharing Four surfaces as in figure.



2. Apply illumination model to calculate each vertex intensity.
3. Linearly interpolate the vertex intensity over the surface of the polygon.

Once N_v is known, intensity at the vertices can obtain from lighting model.



-Here in figure, the intensity of vertices 1,2,3 are I_1 , I_2, I_3 . are obtained by averaging normals of each surface sharing the vertices and applying a illumination model.

- For each scan line , intensity at intersection of line with Polygon edge are linearly interpolated from the intensities at the edge end point.

So Intensity at intersection point A, I_a is obtained by linearly interpolating intensities of I_1 and I_2 as`

$$I_a = \frac{y_a - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_a}{y_1 - y_2} I_2$$

Similarly, the intensity at point B is obtained by linearly interpolating intensities at I_2 and I_3 as

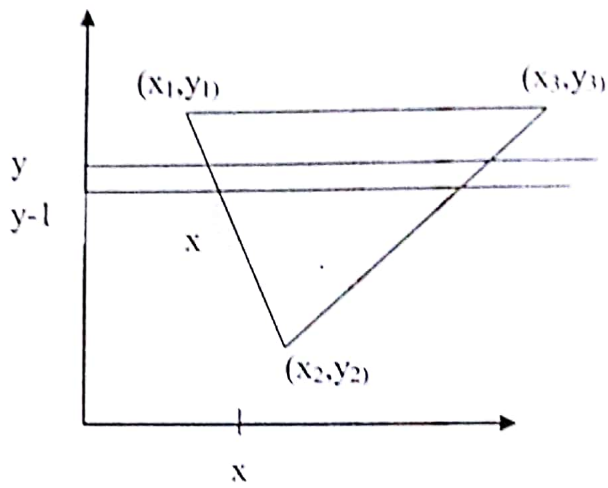
$$I_b = \frac{y_a - y_2}{y_3 - y_2} I_3 + \frac{y_3 - y_a}{y_3 - y_2} I_2$$

The intensity of a point P in the polygon surface along scan-line is obtained by linearly interpolating intensities at I_a and I_b as,

$$I_p = \frac{x_p - x_a}{x_b - x_a} I_b + \frac{x_b - x_p}{x_b - x_a} I_a$$

Then incremental calculations are used to obtain Successive edge intensity values between scan-lines as :

$$I = \frac{y - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y}{y_1 - y_2} I_2$$



Then we can obtain the intensity along this edge for next scan line at y -1 position as

$$I' = \frac{y - 1 - y_2}{y_1 - y_2} I_1 + \frac{y_1 - (y - 1)}{y_1 - y_2} I_2$$

$$= I + \frac{I_2 - I_1}{y_1 - y_2}$$

Similar calculations are made to obtain intensity successive horizontal pixel.

Advantages: Removes intensity discontinuities at the edge as compared to constant shading.

Disadvantages: Highlights on the surface are sometimes displayed with anomalous shape and linear intensity interpolation can cause bright or dark intensity streak called mach-bands.

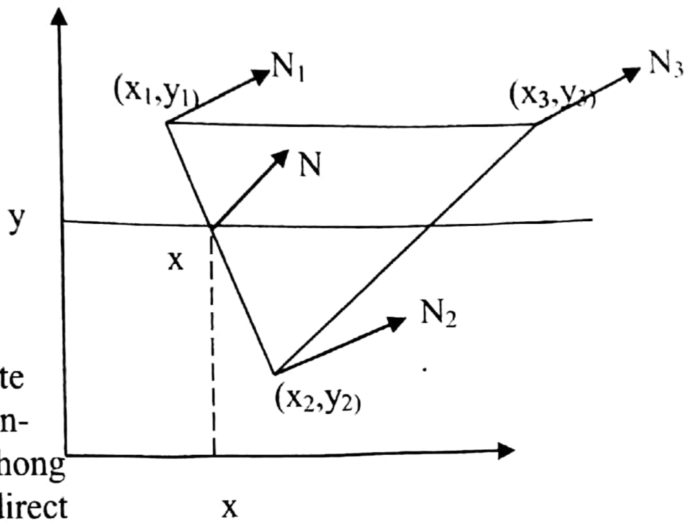
Phong Shading:

A more accurate method for rendering a polygon surface is to interpolate normal vector and then apply illumination model to each surface point. This method is called Phong shading or normal vector interpolation method for shading. It displays more realistic highlights and greatly reduce the mach band effect.

A polygon surface is rendered with Phong shading by carrying out following calculations.

- Determine the average normal unit vectors at each polygon vertex.
- Linearly interpolate vertex normals over the surface of polygon.
- Apply illumination model along each scan line to calculate the pixel intensities for the surface point.

In figure, N_1, N_2, N_3 are the normal unit vectors at each vertex of polygon surface. For scan-line that intersect an edge, the normal vector N can be obtained by vertically interpolating normal vectors of the vertex on that edge as.



$$N = \frac{y - y_2}{y_1 - y_2} N_1 + \frac{y_1 - y}{y_1 - y_2} N_2$$

Incremental calculations are used to evaluate normals between scanlines and along each individual scan line as in Gouraud shading. Phong shading produces accurate results than the direct interpolation but it requires considerably more calculations.

Fast Phong Shading:

Fast Phong shading approximates the intensity calculations using a Taylor series expansion and Triangular surface patches. Since Phong shading interpolates normal vectors from vertex normals , we can express the surface normal N at any point (x,y) over a triangle as

$N = Ax + By + C$

Where A,B,C are determined from the three vertex equations.

$N_k = Ax_k + By_k + C$, $k = 1,2,3$ for (x_k, y_k) vertex.

Omitting the reflectivity and attenuation parameters

$$I_{diff}(x,y) = \frac{L.N}{|L|.|N|} = \frac{L.(Ax + By + C)}{|L|.|Ax + By + C|} = \frac{(L.A)x + (L.B)y + (L.C)}{|L|.|Ax + By + C|}$$

Re writing this

$$I_{diff}(x,y) = \frac{ax + by + c}{(dx^2 + exy + fy^2 + gx + hy + i)^{\frac{1}{2}}} \text{ --------------------(1)}$$

Where a,b,c,d….. are used to represent the various dot products as

$a = \frac{L.N}{|L|}$ ....and so on

Finally, denominator of equation (1) can express as Taylor series expansions and relations terms up to second degree in x,y. This yields,

$$I_{diff}(x,y) = T_5x^2 + T_4xy + T_3y^2 + T_2x + T_1y + T_0$$

Where each T_k is a function of parameters a,b,c,d,……. And so forth.

This method still takes twice as long as in Gouraud shading. Normal Phong shading takes six to seven times that of Gouraud shading