TutorialsDuniya.com

Computer Graphics Notes (DU)

Contributor: Riya Goel

[KMV CS(H)]

Visit https://www.tutorialsduniya.com for Compiled books, Notes, books, programs, question papers with solutions etc.

Facebook: https://www.facebook.com/tutorialsduniya

Youtube: https://www.youtube.com/user/TutorialsDuniya

LinkedIn: https://www.linkedin.com/company/tutorialsduniya

https://www.tutorialsduniya.com

3 Jan, 19.

{uses mathematical representation}.

Computer Graphics

images, ads, visualization, multimedia, graphs.

→ Image processing (processing of images)

→ Visualization

→ Analysis

→ Computer Vision

→ Computer Graphics - models that can be displayed on the screen.

e.g plotting a line on graph using points.

Formats of images → jpeg, bitmap, png.

For images, we have a file of images that consists of a matrix of values that specifies position of image & its colour.

Image processing.

If we process images using models, we call it computer graphics.

eg:- representing boundary of mountain using curve which can be written in a mathematical way.

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

graphics.h (used to draw primitives)

Including graphics in c++.

→ Computer graphics uses implementing all the graphics functions using models.

→ Transformations can be applied on 2D & 3D primitives using CG.

(rotation & translation)

• Computer graphics req. 3 things- (Fields of CG)

(1) Modeling (starting & ending pt. of a line)

(2) Rendering (how a surface is stored. The data str. can be a mesh, pixel position)

(3) Interaction.

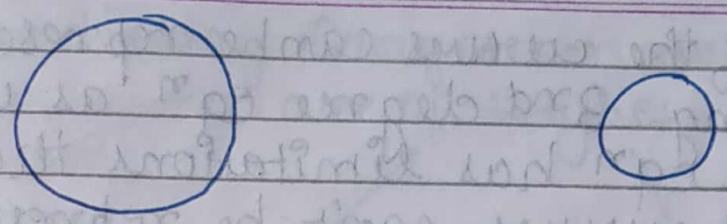
Classes of CG

For a 3D figure (cuboid)
models can be l, b, h, origin (0, 0, 0)

→ For 3D fig. (sphere) circle (2D)
orientation ✓ orientation ✗

Tutorialsduniya.com

https://www.tutorialsduniya.com



The 3D figures are first stored in a form of wiremesh.

polygon mesh (entire 3-D surface is divided in set of polygons).

Basically a Δ'r mesh

coordinates present on surface.

* Model of 3D surface :-

(1) set of small triangles.

(2) normal (as normal tells the orientation of a 3D figure).

Methods to represent 3D surfaces

(1) using polygon mesh.

(2) using polynomial curve/equation normally 3rd degree.

eqn of sphere → $x^2+y^2+z^2=a^2$

TutorialsDuniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

NOTE:- Most of the curves can be represented using 3rd degree eqn as using quad. eqn has limitations that all eqn curves can't be represented.

- Rendering - (1) converting a model into an image.

(2) After that, including colours, depth, shadow and all.

Rendering

Photo realistic rendering	non photo realistic rendering.
creating an image of real time.	every image has a boundary around it eg:- cartoons.
- rendering image in a way that it looks real.	- these effects are generated.

- Animations - can be in form of motion or applying forces on an obj. & changing shapes.

- Interaction - Interaction with user and images using graphical devices.

TutorialsDuniya.com

https://www.tutorialsduniya.com

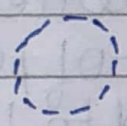
CLASSTIME Page No.
Date / /

→ We display images on monitor (screen), which are of 2 types

Vector scan Systems / Random graphics. (used to draw smooth lines).

- They are also known as calligraphic displays.
- Algorithms / techniques used to draw an image is with help of smooth lines.

eg:- To draw a circle.



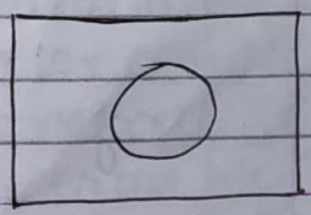
→ drawing small line segments.

- Colour done will also be in terms of line segments.
- Image is converted into algo. which consists of lines.
- It is difficult to display realistic features.

Raster Scan systems.

- The entire screen is divided into small units known as pixels.
- The frame buffer here stores entire screen which consists of color of each pixel position.

{pixel}
↓



frame buffer

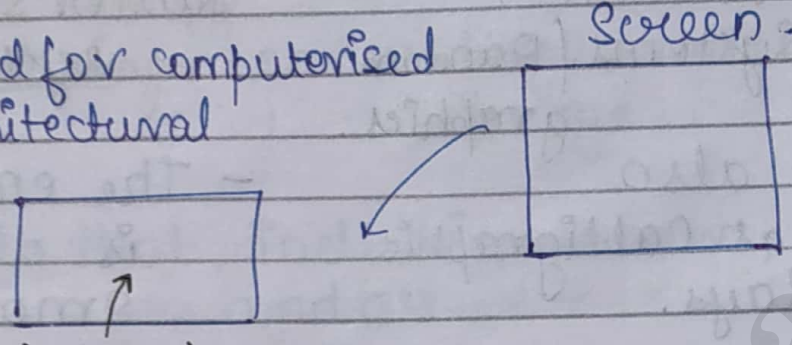
Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

- More CPU & memory efficient
- may req. more space.

- used for computerised architectural designs. (CAD's)



(memory area which stores the image to be displayed on screen)

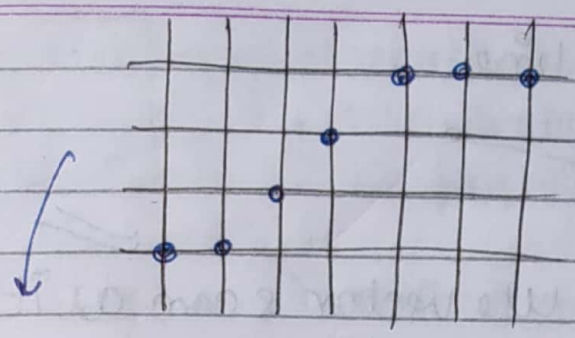
- frame buffer in ~~case~~ vector scan consists of algo. of lines for drawing image.
- req. more time as the primitive/image has to be converted into a set of points. (we need to find all the pts. lying on a line)

Algorithms used to find pts. lying on a line using start & final pts. is Raster scan algorithms.

On screen, these pts. will be drawn & not the complete figure.

TutorialsDuniya.com

https://www.tutorialsduniya.com



→ Screen display in Raster graphics.

Here, if resolution of screen i.e dist. b/w pts. is less then it appears as an image rather than pts.

→ If resolution is less, it appears as pts. (Jagged images)

- In vector scan, geometric images are drawn better.
- In raster scan, realistic pictures can be drawn better.
- efficiency depends on complexity of scene.
- inexpensive.
- used for coloured images & filling of patterns ~~them~~.
- no matter screen is simple or complex, time for refresh is same as same pts. are plotted.

NOTE:- The hardware displays every picture multiple times in 1 sec. so that image doesn't fade. (Refresh).

- Image in frame buffer must be refreshed again & again.

Contributor: Riya Goel

TutorialsDuniya.com

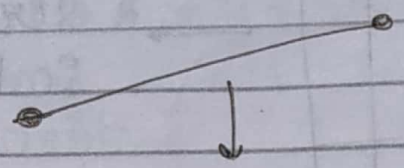
Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc of following subjects from https://www.tutorialsduniya.com

- C and C++
- Programming in Java
- Data Structures
- Computer Networks
- Android Programming
- PHP Programming
- JavaScript
- Java Server Pages
- Python
- Microprocessor
- Artificial Intelligence
- Machine Learning
- Computer System Architecture
- Discrete Structures
- Operating Systems
- Algorithms
- DataBase Management Systems
- Software Engineering
- Theory of Computation
- Operational Research
- System Programming
- Data Mining
- Computer Graphics
- Data Science

- Compiled Books: https://www.tutorialsduniya.com/compiled-books
- Programs: https://www.tutorialsduniya.com/programs
- Question Papers: https://www.tutorialsduniya.com/question-papers
- Python Notes: https://www.tutorialsduniya.com/python
- Java Notes: https://www.tutorialsduniya.com/java
- JavaScript Notes: https://www.tutorialsduniya.com/javascript
- JSP Notes: https://www.tutorialsduniya.com/jsp
- Microprocessor Notes: https://www.tutorialsduniya.com/microprocessor
- OR Notes: https://www.tutorialsduniya.com/operational-research

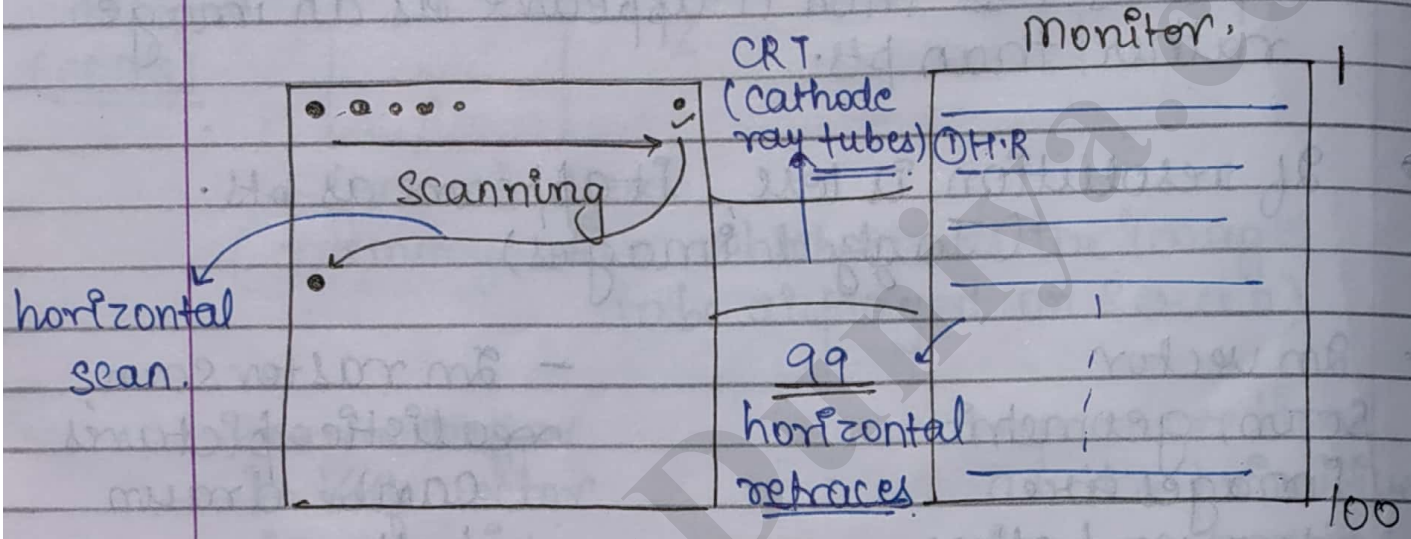
https://www.tutorialsduniya.com

eg:- To draw a line



use vector scan as it is fast.

Here, if we use raster scan we need to find all pts. on line & plot them.



The time when the last point of a line is done & we come to next line, it is called retracing.

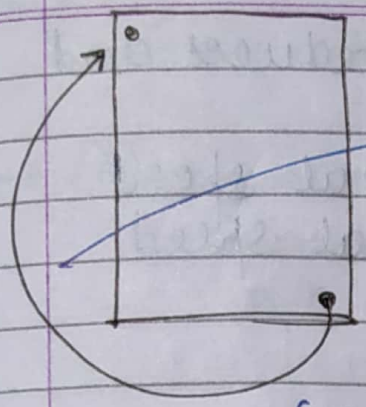
If retracing time is more,

- Frames per sec (Refresh rate) – no. of frames displayed per second.

→ 99 horizontal retraces req. for 1 frame.
Speed → 60 frames/s

∴ 99×60

https://www.tutorialsduniya.com



vertical retrace when we return from final position to initial position.

Vertical retrace occurs when we refresh our page to again start process

60 frames/s

59 vertical retraces occur, as once the frame is present.

For a screen of 100 X 100 & frame/s = 60.

horizontal retraces → 99

vertical retraces → 59

NOTE:- If frame rate is not good, we get an effect of flicker.

To improve this, increase frame speed.

- Interlacing - When we draw a figure, we draw in parts by first taking half of the lines and lines are so closely placed that no such effect occurs.

(Technique to reduce flickering)

1. plot odd no. lines
2. plot even " "

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

This is useful as fading reduces and also time reduces.
So, we have increased logical speed while keeping physical speed same.

TutorialsDuniya.com

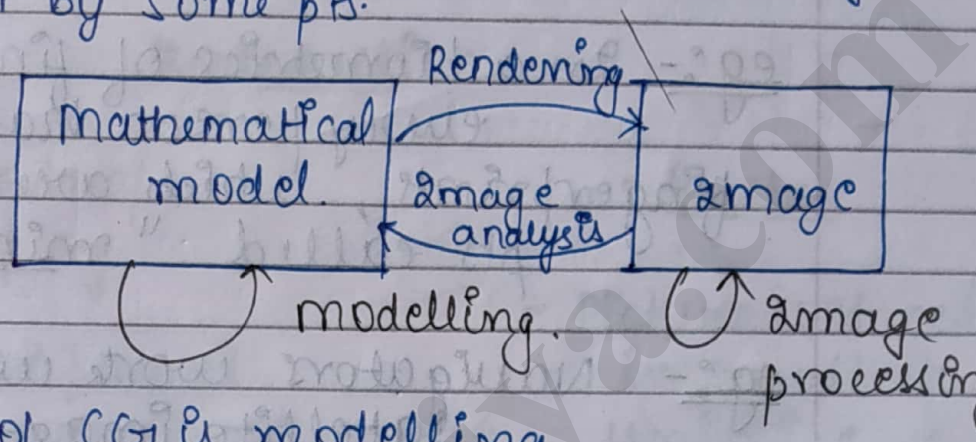
Contributor: Riya Goel

https://www.tutorialsduniya.com

7 Jan, 19

Computer Graphics.

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



One aspect of CG is modelling.
creation of various models

To convert a model into image → Rendering
↓
covers aspects of covering color, texture, lighting effects, shadows etc.
not a part of model.

- Rendering is also called "Image synthesis."
- Modification in image → Image processing.
  ↓
  Aspect of image processing is Image enhancement.
- If from an image, we can get some useful info, then it's called Image analysis.

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

pattern recogonition.

identifying certain common features from a given image.

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

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

* Graphics → deals with models and converting it into images.

also deals with interaction.

eg:- Analysis used in stock markets.

image analysis / Visualization.

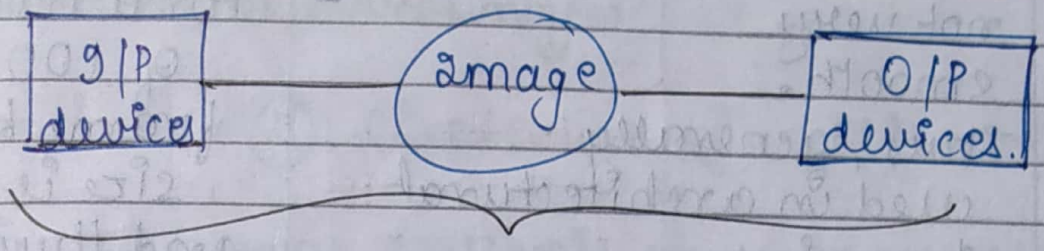
- Computer Vision-
  ↓
  includes rendering, image analysis, pattern recogonition.

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

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /



Interaction

- Random/ Vector/ Calligraphy

- all images displayed on screen are stored in form of line drawing commands.
- The memory area is known as frame buffer.
  The controller takes image & displays it here, on screen.
- eg:- a landscape will be drawn in form of lines in frame buffer.
- similar to ms logo.
- realistic images/ curves are

- Raster

- Picture is stored in smallest unit i.e pixels.
- entire screen is considered as a rectangular grid of pixels
- Frame buffer stores value of each pixel.
- The controller picks up each pixel value & displays on screen
- Each value is converted into pixels.

Scan conversion

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

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

- less commonly used as they are expensive

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

In terms of hardware & not efficiency.

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

- For a resolution of 100 X 100 frame buffer size is 100X100, and thus it stores each pixel value.

Scan conversion

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

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

additional thing in raster graphics. Done by the software

TutorialsDuniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

→ Primitives to be drawn ↓

① Line ② Circle ③ Ellipse.

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

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

• Drawbacks of Raster Graphics ↓

* aliasing / Staircase effect / Jagging :-

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

↓

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

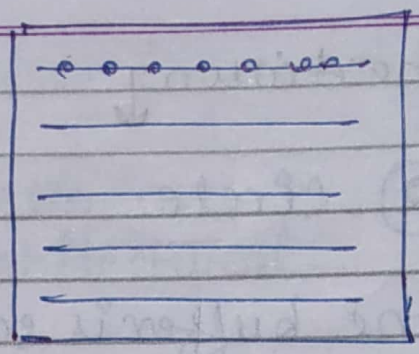
Resolution - The screen is divided into no. of rows of pixels called scan lines.

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

Tutorialsduniya.com

https://www.tutorialsduniya.com

eg :-



200 scan lines & 100 pixels on each line

$= 200 \times 100$

- Aspect Ratio (related to display) — these pixels are w/o overlapping.

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

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

{Otherwise aspect ratio changes}

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

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

{representation may vary}.

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

Contributor: Riya Goel

TutorialsDuniya.com

Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc of following subjects from https://www.tutorialsduniya.com

- C and C++
- Programming in Java
- Data Structures
- Computer Networks
- Android Programming
- PHP Programming
- JavaScript
- Java Server Pages
- Python
- Microprocessor
- Artificial Intelligence
- Machine Learning
- Computer System Architecture
- Discrete Structures
- Operating Systems
- Algorithms
- DataBase Management Systems
- Software Engineering
- Theory of Computation
- Operational Research
- System Programming
- Data Mining
- Computer Graphics
- Data Science

- Compiled Books: https://www.tutorialsduniya.com/compiled-books
- Programs: https://www.tutorialsduniya.com/programs
- Question Papers: https://www.tutorialsduniya.com/question-papers
- Python Notes: https://www.tutorialsduniya.com/python
- Java Notes: https://www.tutorialsduniya.com/java
- JavaScript Notes: https://www.tutorialsduniya.com/javascript
- JSP Notes: https://www.tutorialsduniya.com/jsp
- Microprocessor Notes: https://www.tutorialsduniya.com/microprocessor
- OR Notes: https://www.tutorialsduniya.com/operational-research

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

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

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

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

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

If persistence ↑

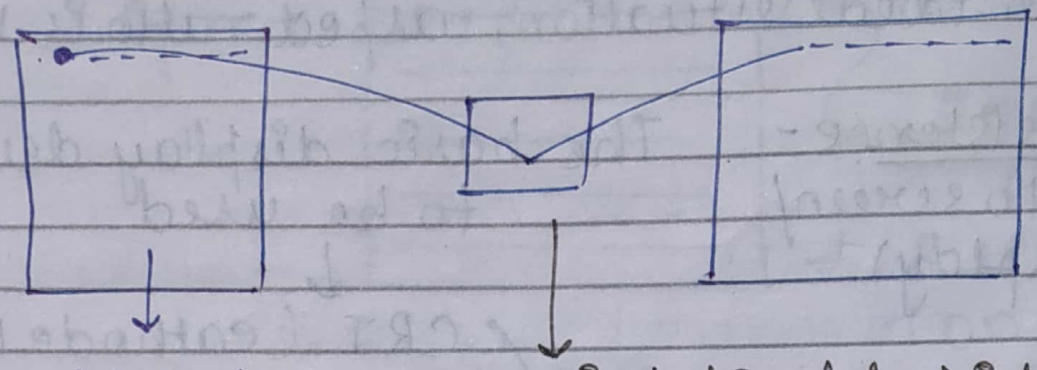
Refresh rate → how many times pixels take to refresh again per sec. / how many times frame buffer appears on sec.

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

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

Tutorialsduniya.com

https://www.tutorialsduniya.com



Digital.

First pixel is picked up and displayed

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

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

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

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

↓

To reduce this :-

(1) Increase refresh rate

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

② Interlacing (Interconnecting)
↓
Plot the alternate scan lines in a given refresh rate, doubling the frame rate and flickering can be reduced & monitors using this technique are "Interlaced monitors".

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

• Types of Display :-

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

② Grays cale - If system allows a different shade of grey.
If we want to make a pixel glow, it depends on shade of pixel.

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

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

In a row, m pixels
" " col, n pixels } size of image $m \times n$.

Tutorialsduniya.com

https://www.tutorialsduniya.com

If for every pixels, we have 3 bits.
↓
∴ 8 values can be stored (0-7)

Thus, we can have 8 shades of grey

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

∴ Image size → $\underbrace{m \times n \times 8}$

∴ 256 shades of grey can be stored.

As we ↑se size of any pixel, size of image ↑ses and thus more shades of grey can be encorporated.

(3) For a coloured image, shades will be divided into
↓
$\underbrace{\text{red blue green}}_{\text{as they are primary colours}}$ (rgb)

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

eg:- 24 bit system

R	B	G
(8 bits)	(8 bits)	(8 bits)

→ For pure red pixel

111---1 000--- 000---
blue green

→ For white pixel

all 1's

1111------ 24 times.

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

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

NOTE:- If system can store 2^{24} pixels but the monitor is not capable of, then we use concept of "lookup tables".
(at hardware level)

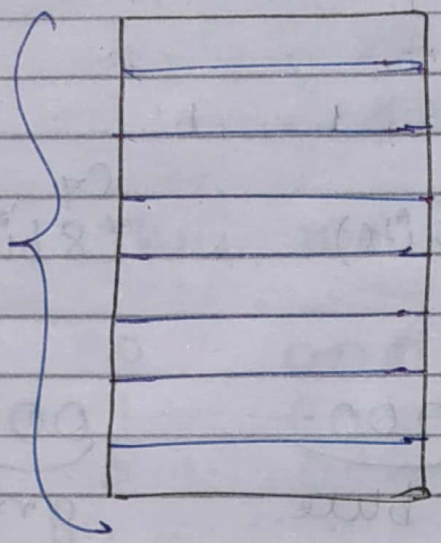
If monitor displays less colours but system can store more colour values then we need to match this mismatch.

display device displays colour — 2^{12}
acquired by pixels — 2^{24}.

Contributor: Riya Goel

https://www.tutorialsduniya.com

lookup table (like a palette).



2^{12} colours at one time.

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

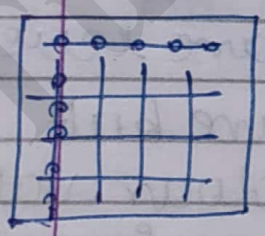
For some other colour, then we map another set of 2^{12} values into table.

* Numericals -

Q1 Find out the aspect Ratio of a Raster System using 8x10 inches screen and 100 pixels per inch.

8 → horizontal
10 → vertical.

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



$\left\{ \frac{\text{horizontal scan lines}}{\text{vertical scan lines}} \right\}$

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

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

{ 1280 → columns
1024 → rows }

Assuming horizontal & vertical retrace times are negligible.

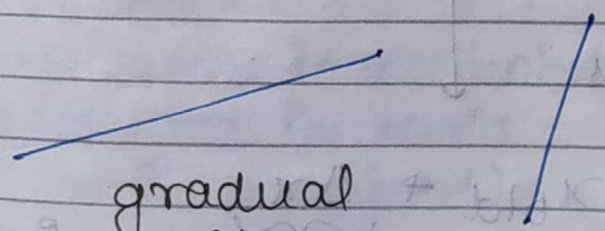
1 frame → $\frac{1}{60}$ sec.
(1280 X 1024 scan lines)

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

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

- Line Drawing Algorithms :-

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



gradual slope.

steep slope

∴ line can have 4 cases.

Contributor: Riya Goel

https://www.tutorialsduniya.com

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

2 algorithms for drawing a line.

① DDA (Digital differential analysing)

or Incremental algo – given after name of a mechanical device DDA that solves diff. eq^n by numerical methods.

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

eg :- $\underbrace{y = mx + B}_{\text{eq}^n \text{ of line.}}$

finding new value from old value

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

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

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

$\underbrace{x_{old} \text{ and } y_{old}}_{\text{old values}}$

$$\therefore x_{new} = x_{old} + \underbrace{\Delta x}_{}$$

some incremental value of x

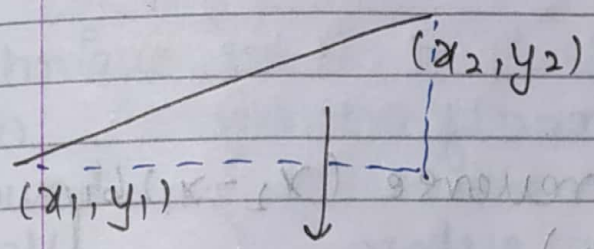
https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

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

- Gradual slope

variation of x is more than that of y.



$x_2 - x_1 > y_2 - y_1$

$\therefore dx > dy$

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

$|m| < 1$

(Gradual slope)

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

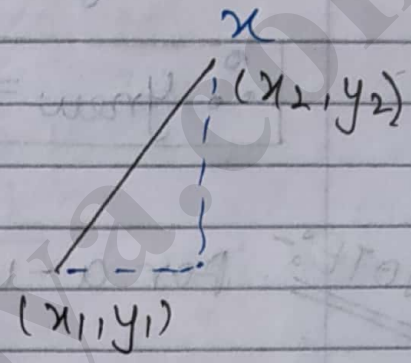
$\therefore \Delta x = 1$

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

$$\boxed{x_{new} = x_{old} + 1}$$

Steep slope

variation of y is more than x



$y_2 - y_1 > x_2 - x_1$

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

→ For this case, increment y & find x.

$\therefore \Delta y = 1$

$$\boxed{y_{new} = y_{old} + 1}$$

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

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

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

since $\Delta y = 1$

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

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

$\therefore dy = m\Delta x$
and $\Delta x = 1$

$dy = m$

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

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

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

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

① Given x_1, y_1 and x_2, y_2.

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

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

NOTE:- {DT of slope $\longrightarrow$ (m)}
↓
floating pt. value.

as $|m| < 1$
can be 0.05, - - - .

$x, y \longrightarrow$ Integer values.

$y_{new} = y_{old} + m \longrightarrow$ float value

Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

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

Technique to round off.

eg – value = 1.2
+ 0.5
1.7

floor (1.7) = 1

value = 1.6
+ 0.5
2.1

floor (2.1) = 2

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

Algorithm-

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

Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

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

To plot a pixel on screen, use

putpixel(x1, y1, color);

If color not specified, it takes colour of setcolor

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

Since in C++ we don't have a round func.

∴ $y + 0.5$ do its floor.

Round(yi) = FLOOR (0.5 + yi)

10 Jan, 19 Practical

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

help → index → line example.

Linking driver

initgraph (&gdriver, &gmode, " ");

closegraph()

gdriver, gmode → reference variables

gmode → specifies resolution.

" " → a path specified for driver

closegraph() → to close graph

Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

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

int gdriver = DETECT, gmode, errorcode.

{ Ctrl + F1 → Help }.

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

line (0, 0), $\underbrace{x\,max, y\,max}_{\text{rightmost corner}}$) // draws a line.

Options → linker → libraries → set Graphics library
[X] → set

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

TutorialsDuniya.com

TutorialsDuniya.com

Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc of following subjects from https://www.tutorialsduniya.com

- C and C++
- Programming in Java
- Data Structures
- Computer Networks
- Android Programming
- PHP Programming
- JavaScript
- Java Server Pages
- Python
- Microprocessor
- Artificial Intelligence
- Machine Learning
- Computer System Architecture
- Discrete Structures
- Operating Systems
- Algorithms
- DataBase Management Systems
- Software Engineering
- Theory of Computation
- Operational Research
- System Programming
- Data Mining
- Computer Graphics
- Data Science

- Compiled Books: https://www.tutorialsduniya.com/compiled-books
- Programs: https://www.tutorialsduniya.com/programs
- Question Papers: https://www.tutorialsduniya.com/question-papers
- Python Notes: https://www.tutorialsduniya.com/python
- Java Notes: https://www.tutorialsduniya.com/java
- JavaScript Notes: https://www.tutorialsduniya.com/javascript
- JSP Notes: https://www.tutorialsduniya.com/jsp
- Microprocessor Notes: https://www.tutorialsduniya.com/microprocessor
- OR Notes: https://www.tutorialsduniya.com/operational-research

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

14 Jan, 19.

Line Drawing Algorithms

→ We can't plot a floating pt. pixel value thus we round off value in DDA algo.

y_0

error { (e_0) $y_1 = y_0 + m$
$(2e_0)$ $y_2 = y_1 + m$
⋮ } error keeps on increasing and adding up

Thus actual line slightly gets drifted from original position.

Drawback of DDA

To get rid of these algorithms, we must have an algorithm that takes integer values. Thus, we have another line drawing algorithm

↓

Bresenham's Line Drawing algorith
(or mid pt. Line drawing algo.)

- use of mid pt. concept.
- In DDA, we take 2 cases $|m| > 1$ or $|m| < 1$
- For Bresenhams, we have 4 cases

↓

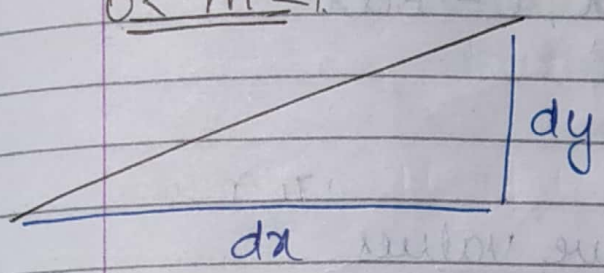
Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

+ve slope, gradual
-ve slope, gradual

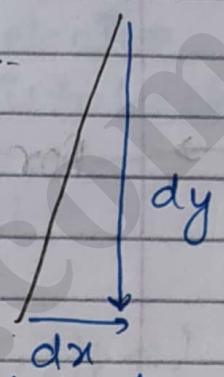
$0 < m \leq 1$



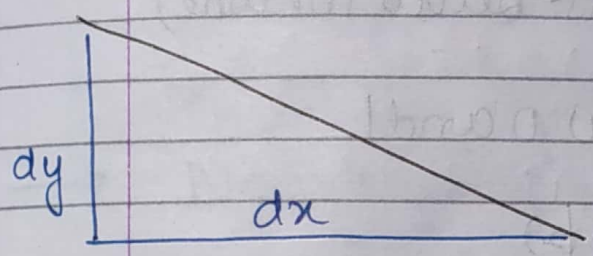
$dx \geq dy$ (+ve slope)

+ve slope, steep
-ve slope, steep.

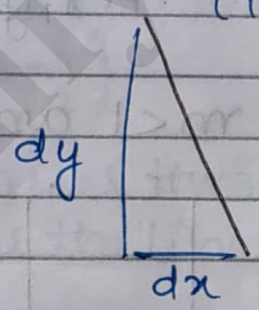
$m > 1$



$dx < dy$
(+ve slope)



$(dx > dy)$
$m \leq 1$ on -ve side



$dx < dy$
$m > 1$ on -ve side
(b/w 0 and -1)

* Derivation ↓

For a pt. (x, y) lying on the line we have

$$F(x, y) = ax + by + c = 0 \quad \text{--- (1)}$$

and $y = mx + B$

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

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

$$y\,dx = x\,dy + B\,dx.$$

Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

$$x\,dy - y\,dx + B\,dx = 0 \quad \text{--- (2)}$$

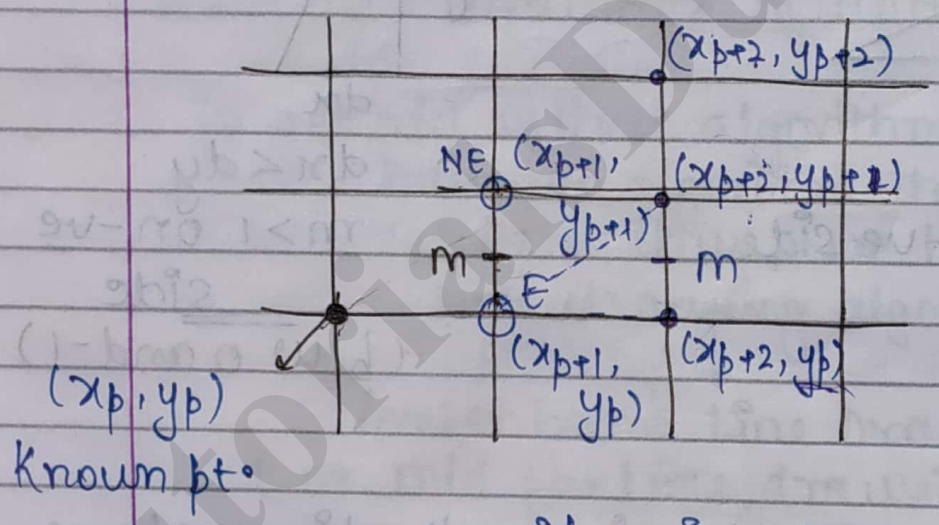
comparing (1) and (2)

$$a = dy, \quad b = -dx, \quad c = B\,dx$$

→ For any pt. (x, y)]

$F(x,y)$ can have values
$F(x,y) = 0$ (on the line)
$F(x,y) < 0$ (pts. above the line)
$F(x,y) > 0$ (pts. below the line)

Case I :- $m < 1$ and m is b/w 0 and 1



If m is $0 < m < 1$
i.e line is gradual
we increment x and keeps ~~increasing~~ finding values of y

This is because $dx > dy$ ∴ for every value of x we get a y as x contains more no. of pts.
If we increment y, then there are some

Contributor: Riya Goel

https://www.tutorialsduniya.com

pts. for which no x is there.

$(x_p, y_p) \rightarrow (x_{p+1}, y_p)$ or (x_{p+1}, y_{p+1})

Now how to make this choice whether pt. E or pt. NE is to be chosen

on basis of mid pt.

mid pt. of (x_{p+1}, y_p) and (x_{p+1}, y_{p+1})

$\downarrow$

$(x_{p+1}, y_p + \frac{1}{2})$

→ Now our line either passes through this mid pt., or below the line or above the line.

→ On basis of pt. M we decide whether the line passes above the value of M, then NE has to be plotted.
If line lies below M, then E has to be plotted
If line lies on M, we can choose either pt. E or NE.

(1) Find $F(x_{p+1}, y_p + \frac{1}{2})$] 3 cases
$< 0, > 0$ or $= 0$.

Case I - value of mid pt. is > 0.

deciding factor { $F(x_{p+1}, y_p + \frac{1}{2}) > 0$ mid pt. lies

Contributor: Riya Goel

https://www.tutorialsduniya.com

below the line
∴ choose pt. NE

Now we choose amongst
(x_{p+2}, y_{p+1}) and (x_{p+2}, y_{p+2})
↓
mid pt. → $(x_{p+2}, y_p + 3/2)$

deciding factor → $F(x_{p+2}, y_p + \frac{3}{2})$
(d_{new})
$= a(x_{p+2}) + b(y_p + \frac{3}{2}) + c$ — (3)

(2) Write d_{new} in terms of d_{old}
↓
(3) can be rewritten as.

$$d_n = a(x_p + 1 + 1) + b(y_p + 1 + \frac{1}{2}) + c$$
$$= a((x_p + 1) + 1) + b(y_p + \frac{1}{2} + 1) + c$$
$$= \underbrace{\boxed{a(x_p + 1) + b(y_p + \frac{1}{2}) + c}}_{d_{old}} + a + b$$

$$\left\{ d_{new} = d_{old} + \underbrace{a + b}_{\Delta NE} \right\}$$

If we choose pt. NE, then we add an increment of ΔNE to obtain next pt. iteration

Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Case II - Value is < 0.

Initial d value $= F(x_p+1, y_p+\frac{1}{2}) < 0$

Choose $E(x_p+1, y_p)$

Now we have choice of 2 pts.

$\underbrace{(x_p+2, y_p) \text{ and } (x_p+2, y_p+1)}$

mid pt $\rightarrow \underbrace{(x_p+2, y_p+1/2)}_{d_{new}}$

$d_{new} = F(x_p+2, y_p+1/2)$

$= a(x_p+2) + b(y_p+\frac{1}{2}) + c$

$= a(x_p+1) + b(y_p+\frac{1}{2}) + c + a$

$d_{new} = d_{old} + a \rightarrow \underline{\Delta E}$

NOTE:- First d value is calculated by 1st pt. on line given by user.

$d_0 = F(x_0+1, y_0+\frac{1}{2})$

$= a(x_0+1) + b(y_0+\frac{1}{2}) + c$

$= \underbrace{ax_0 + by_0 + c}_{F(x_0, y_0) = 0} + a + b/2$

pt. lies on line.

Since it is not known, we need to remove it

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

$\therefore d_0 = a + \frac{b}{2}$

On this value, decide whether pt. E or NE is chosen

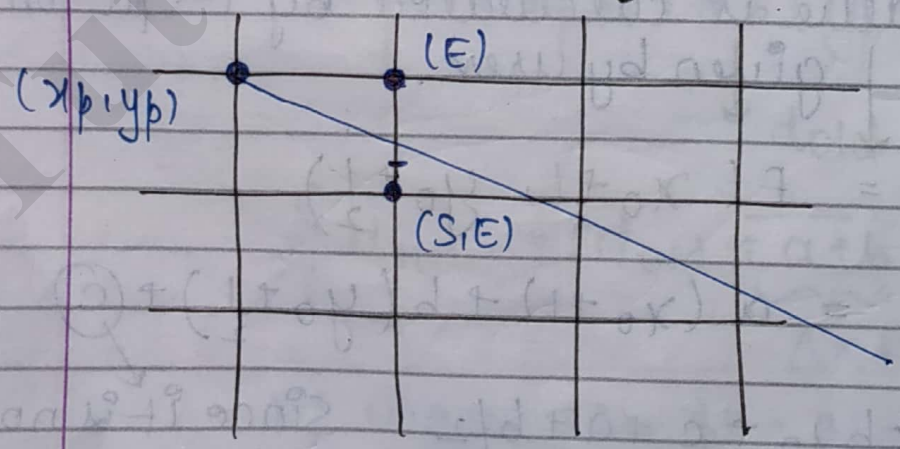
If NE is chosen
$d_0 = a + \frac{b}{2} + \Delta NE$

If E is chosen
$d_0 = a + \frac{b}{2} + \Delta E$

→ We follow these steps until (x_n, y_n) last pt. of line.

NOTE: On screen, the line ╱ will be appearing as ╲ as origin is from top left corner.

- For -ve slope side ($m < -1$)



Initial value
↓
$(x_0 + 1)$ and
$(y_0 - \frac{1}{2})$
instead of $y_0 + \frac{1}{2}$

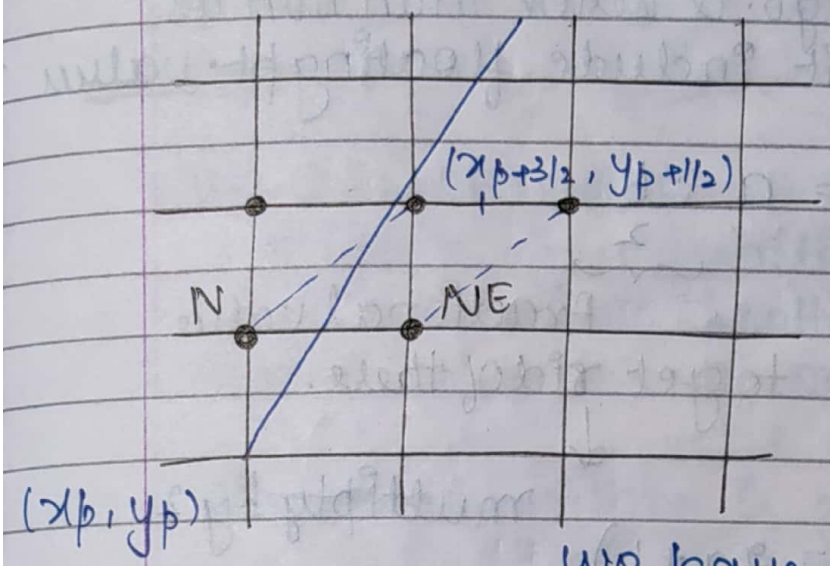
Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

→ Slope of line is $m > 1$ (steep line) {case 3}.



Here we increment y and find values of x

coordinates changed.

Known (x_p, y_p)

we have 2 choices N and NE

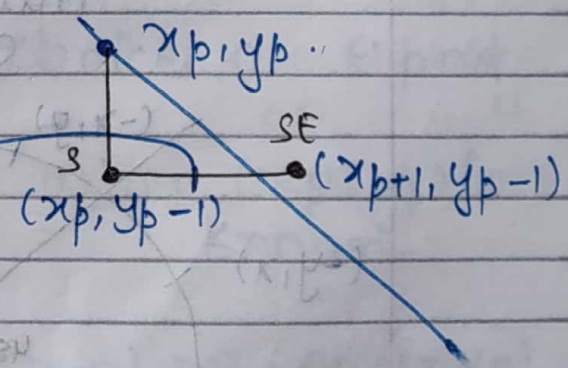
deciding factor → (x_p+1, y_p+1) {interchange x & y}

choices are either
North point (x_p, y_p+1)
or North East point (x_p+1, y_p+1)

mid pt. → $(x_p+1/2, y_p+1)$.

→ Case 4 :- $m > -1$ b/w $0 \geq m \geq -1$.

$(x_p+\frac{1}{2}, y_p-1)$



Tutorialsduniya.com

https://www.tutorialsduniya.com

NOTE: Line drawing algo. is better than DDA as it doesn't include floating pt. values

$$d_0 = a + \underbrace{\frac{b}{2}}_{\text{Fractional value}}$$

to get rid of there.
↓
multiply by 2

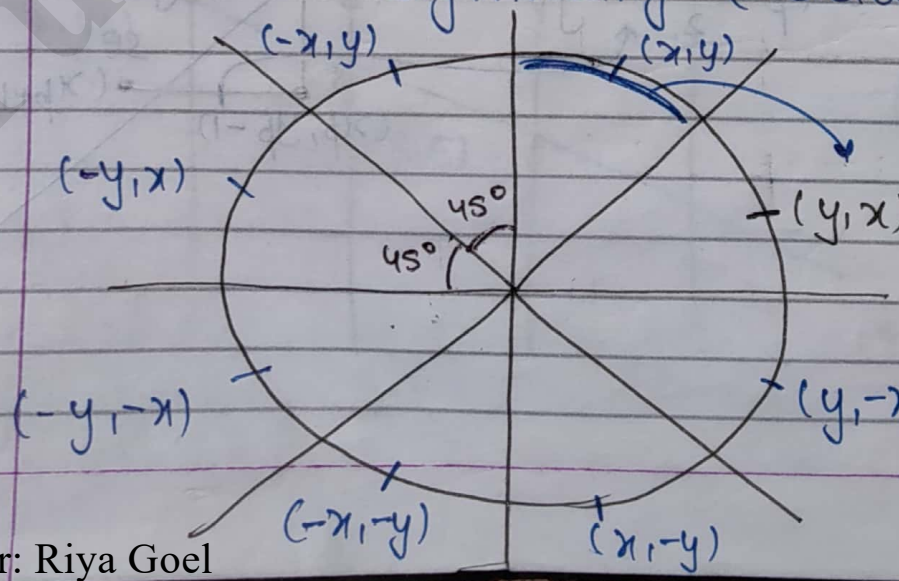
$$(d_0 = 2a + b)$$

$F(x,y) = 0$
$2F(x,y) = 0$
⋮
$\underbrace{nF(x,y) = 0}_{\text{no effect of multiplied constant.}}$

$\Delta E \rightarrow a$ NOW ΔE becomes $2a$
$\Delta NE \rightarrow a+b$ NOW $\Delta NE \rightarrow 2(a+b)$

- Circle Drawing Algorithm-
  → has curves & is symmetrical (advantage over line)

i.e If we know one pt., we can find multiple pts. using 8 way symmetry (octants).



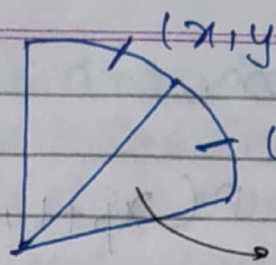
If we find this value of octant, then we can find other octants

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

{xy was known}



Since it is at 45° this becomes mirror image of (x,y)

We can feed all the 8 octants in a function instead of plotting 1 pt. of octant

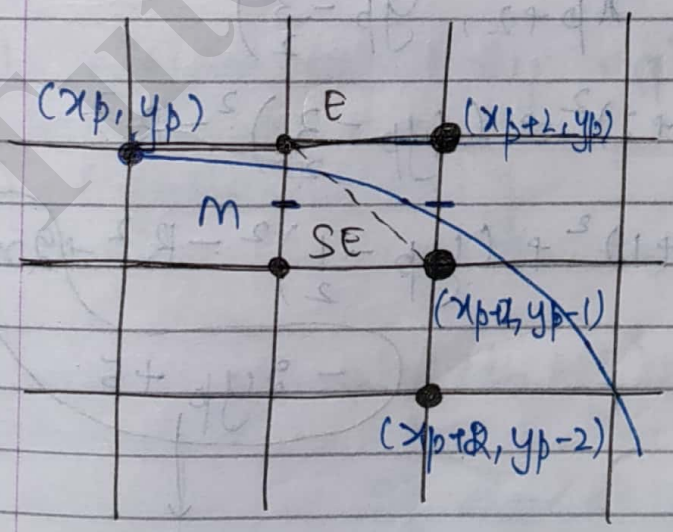
circle algorithm is given by Bresenham's
∴ It is known as. {uses same concept}

- Bresenham's Circle Drawing algo. / mid pt. circle drawing algo

→ eqn of circle → $x^2+y^2=R^2$
{Assuming centre of circle is origin (0,0)}

→ If centre (h,k) ↓

$$(x-h)^2+(y-k)^2=R^2.$$



choices are E and SE when (xp, yp) is known.

$F(x,y)=0$ (on circle)
$F(x,y)>0$ (above circle)
$F(x,y)<0$ (below circle)

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

→ Deciding factor.

$$d_{old} = F(x_p+1, y_p - \tfrac{1}{2})$$

if < 0 ↓ choose pixel E

$$E \rightarrow (x_p+1, y_p)$$

$$d_{new} = F(x_p+2, y_p - \tfrac{1}{2})$$

↓

$$= (x_p+2)^2 + (y_p - \tfrac{1}{2})^2 - R^2$$

$$= (x_p+1)^2 + (y_p - \tfrac{1}{2})^2 - R^2 + \underbrace{2x_p + 3}_{\Delta E}$$

→ If $d > 0$
(outside circle)
↓
choose SE. (x_p+1, y_p-1)

as line is below mid pt. thus closer to SE

$$d_{new} = F(x_p+2, y_p - \tfrac{3}{2})$$

$$= (x_p+2)^2 + (y_p - \tfrac{3}{2})^2 - R^2$$

$$= (x_p+1)^2 + (y_p - \tfrac{1}{2})^2 - R^2 + 2x_p - 2y_p + 5$$

↓

ΔSE

TutorialsDuniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

NOTE: In case of line drawing algos, increments are constant i.e independent of x_p and y_p but in circle, every circle is dependent on values of x_p and y_p.
We need to calculate increments of circle at every pt.

initial pt. $\longrightarrow (0, r)$ → as radius is known.

∴ First initial d value

$$d_0 = F(x_0+1, y_0-\tfrac{1}{2})$$

$$= F(1, r-\tfrac{1}{2})$$

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

$$d_0 = \frac{5}{4} - r$$

{can be $=0$, <0, >0}

again a floating pt. value.

Sol^n :-

(1) either multiply eq^n by 4.

(2) (Better sol^n)

$$d_0 = \frac{5}{4} - r$$

$$h \leftarrow d_0 - \frac{1}{4} = 1 - r$$

{subtracting $\frac{1}{4}$ from both sides}

If $d_0 < 0 \Longrightarrow h < 0$
$\quad\;\; > 0 \qquad\;\; h > 0$

Contributor: Riya Goel

https://www.tutorialsduniya.com

$$\underbrace{d_0 - \frac{1}{4} < 0}$$

We are comparing d_0 by 0
∴ This new value can be compared by $\frac{1}{4}$

effective value of $\frac{1}{4}$ becomes 0 only } But value b/w 0 and $\frac{1}{4}$ can't come as we are using integral values

∴ We can use $(1-r)$ and safely compare d_0 with 0.

∴ $\boxed{d_0 = 1-r}$ → now d_0 can't get any value of $\frac{1}{4}$ as 1 and r are both integers.

→ Now we will write a func. to plot 8 pixels.

```
func ()
{
    // Call putpixel 8 times
    with diff. coordinates.

}
```

→ On screen, we will be able to see only 1 quadrant as origin is top left corner

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

∴ We need to take radius and centre from user & shift origin to centre

↓

$(x-h)^2 + (y-k)^2 = r^2$

$\left\{ \therefore \begin{array}{l} x = X+h \\ y = Y+h \end{array} \right\}$.

15 Jan. 19

- Practical

→ How to do CG programs in DevC++ :-

1. Copy graphics.h and winbgim.h to c:\Program Files(X86)\Dev-CPP\MinGW64\Include

2. Copy libbgi.a to c:\ProgramFiles(X86)\Dev-Cpp\MinGW64\lib.

3. Copy 6-ConsoleAppGraphics.template and ConsoleApp_cpp_graph.txt to c:\ProgramFiles(X86)\Dev-Cpp\Templates.

4. From compiler selection dropdown choose TDM.GCC X.X.X 32bit Release.

↓

5. Done.

Select 32bit Release.

File → New → Project → Graphics Console → OK → make a folder → Project → Project Options → Parameters → Linker window

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

- lbgi - lgdi32 - luser32 (by default included)
- ~~lg~~ lbgi
- lgdi32
- lcomdlg32
- luuid
- loleaut32
- lole32

TutorialsDuniya.com

TutorialsDuniya.com

Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc of following subjects from https://www.tutorialsduniya.com

- C and C++
- Programming in Java
- Data Structures
- Computer Networks
- Android Programming
- PHP Programming
- JavaScript
- Java Server Pages
- Python
- Microprocessor
- Artificial Intelligence
- Machine Learning
- Computer System Architecture
- Discrete Structures
- Operating Systems
- Algorithms
- DataBase Management Systems
- Software Engineering
- Theory of Computation
- Operational Research
- System Programming
- Data Mining
- Computer Graphics
- Data Science

- Compiled Books: https://www.tutorialsduniya.com/compiled-books
- Programs: https://www.tutorialsduniya.com/programs
- Question Papers: https://www.tutorialsduniya.com/question-papers
- Python Notes: https://www.tutorialsduniya.com/python
- Java Notes: https://www.tutorialsduniya.com/java
- JavaScript Notes: https://www.tutorialsduniya.com/javascript
- JSP Notes: https://www.tutorialsduniya.com/jsp
- Microprocessor Notes: https://www.tutorialsduniya.com/microprocessor
- OR Notes: https://www.tutorialsduniya.com/operational-research

https://www.tutorialsduniya.com

23 Jan, 19

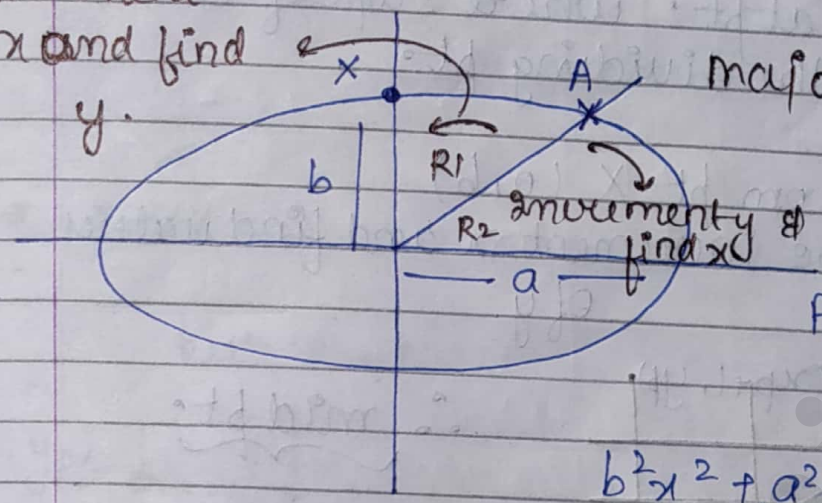
CLASSTIME Page No.
Date

Symmetric figure in 4 parts.

Ellipse Drawing Algorithm

through mid pt. drawing algorithm.

Increment x and find y.



major and minor axis decide the eq^n ↓

$$f(x,y) = \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

$$b^2x^2 + a^2y^2 - a^2b^2 = 0$$

Along 1 quadrant, the slope is varying in ellipse.

At pt. A ⟶ slope of tangent is $\underline{-1}$ and gradient is $\underline{1}$ ($\perp$)

⟶ The deciding factor for dividing into regions is by finding gradient.
At pt. where gradient = 1, then we move from R_1 to R_2 i.e slope of tangent is opp.

- Gradient ↓ (partial derivative wrt x + partial derivative wrt y).

$$f(x,y) = \frac{\partial f}{\partial x}\hat{i} + \frac{\partial f}{\partial y}\hat{j}$$

$$= 2b^2x\hat{i} + 2a^2y\hat{j}$$

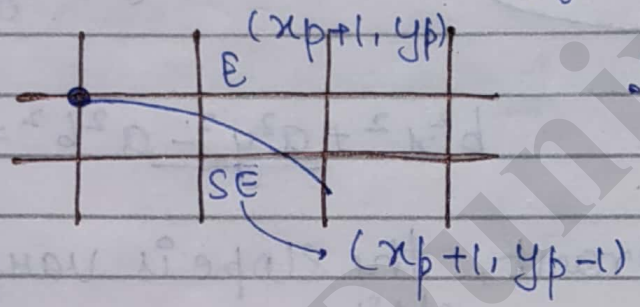
Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

→ In region 1, the j component of the gradient is greater / larger than the i component. Both the components are equal at A i.e at pt. where slope of curve is -1 or the dividing pt.

- We start from pt. X (0, b)
  Thus, we increment x and find values of y.



∴ mid pt.

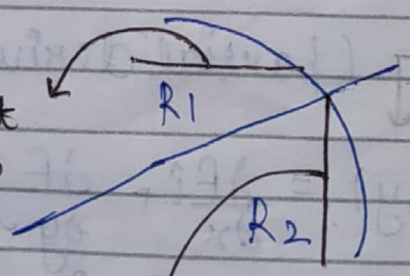
$x_{p+1}, y_p - 1/2$.

We keep on checking this value ensuring j component is greater than i component. The moment we get $j < i$ we move to Region 2.

$$a^2 \underbrace{(y_p-1)^2}_{\text{j component}} \leq b^2 \underbrace{(x_p+1)}_{\text{i component}}$$

deciding factor how to move from R_1 to R_2.

Now for region 2 we decrement y and find all values of x

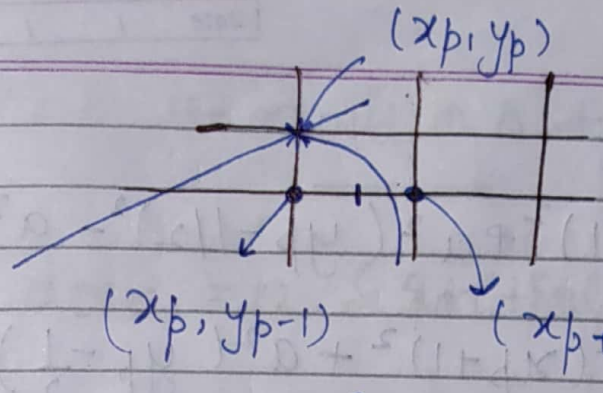


x component has more pts.

y component has more pts.

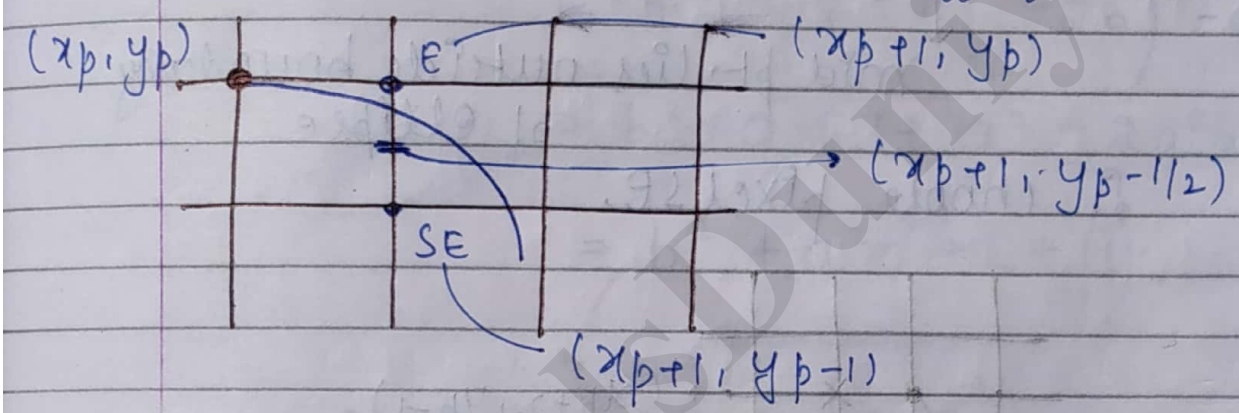
Contributor: Riya Goel

https://www.tutorialsduniya.com



mid pt → $(x_p + 1/2, y_p - 1)$

→ Now we need to find initial decision variable (d_0)
but derivations are diff. for R_1 and R_2.

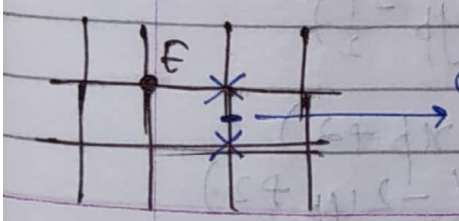


$\therefore d_{old} = F(x_p+1, y_p - 1/2)$

Now $d_{old} < 0$ then mid pt. lies inside the boundary of ellipse.

$\therefore$ Boundary is closer to pixel E and thus choose pt. E

If E is chosen ↓



$d_{new} = F(x_p+2, y_p - 1/2)$

$= b^2(x_p+2)^2 + a^2(y_p - 1/2)^2$

Contributor: Riya Goel

https://www.tutorialsduniya.com

$-a^2b^2 = 0$

$$b^2(x_p+1+1)^2 + a^2(y_p - 1/2)^2 - a^2b^2$$

$$= b^2(x_p+1)^2 + a^2\left(y_p - \frac{1}{2}\right)^2 - a^2b^2 + \underbrace{b^2(2x_p+3)}_{\text{extra part}}$$

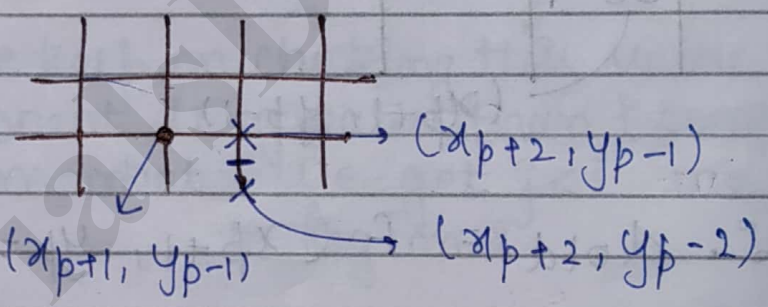
$$= d_{old} + \Delta E \quad (1)$$

If $d_{old} > 0$

↓

mid pt. lies outside boundary of ellipse

∴ choose pixel SE.



∴ mid pt → $(x_p+2, y_p - 3/2)$

$$d_{new} = F(x_p+2, y_p - 3/2)$$

$$= b^2(x_p+2)^2 + a^2(y_p - 3/2)^2$$

$$= b^2(x_p+1)^2 + a^2\left(y_p - \frac{1}{2}\right)^2 - a^2b^2 + b^2(2x_p+3) + a^2(-2y_p+2)$$

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

$= d_{old} + \Delta SE$ —— (2)

$d_{old} = 0$ (initial deciding factor).

Pt. known $(0, b)$

(for region 1) $\rightarrow d_0 = F(0+1, b-\frac{1}{2})$

$= b^2(1)^2 + a^2(b-\frac{1}{2})^2 - a^2b^2$

$= b^2 + a^2(b^2 + \frac{1}{4} - 2b) - a^2b^2$

$= b^2 + \cancel{a^2b^2} + \frac{a^2}{4} - 2a^2b - \cancel{a^2b^2}$

$= b^2 + a^2(-b + \frac{1}{4})$ —— (3)

Now we find whether this value is

< 0
add ΔE

> 0
add ΔSE
pixel plotted will be
(x_p+1, y_p-1)

$\therefore$ plotting of 4 pixels.
(4 quadrants)

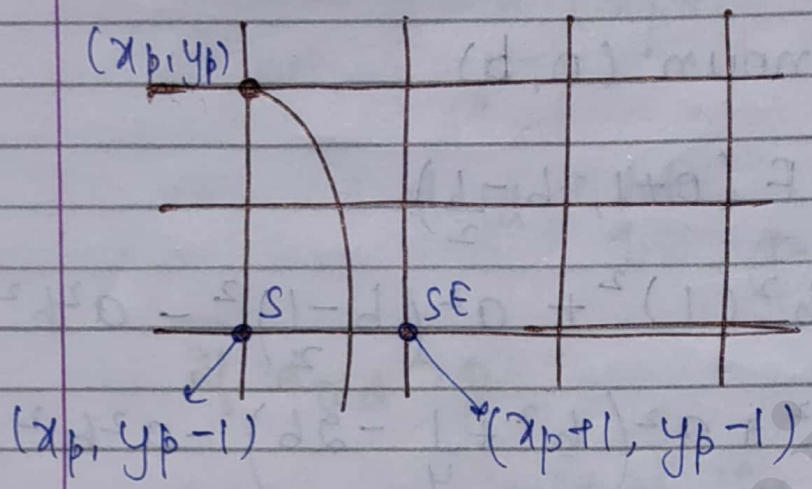
Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

Now for region 2 (Slope is in -ve dir^n)

We will decrement y and find value of x.



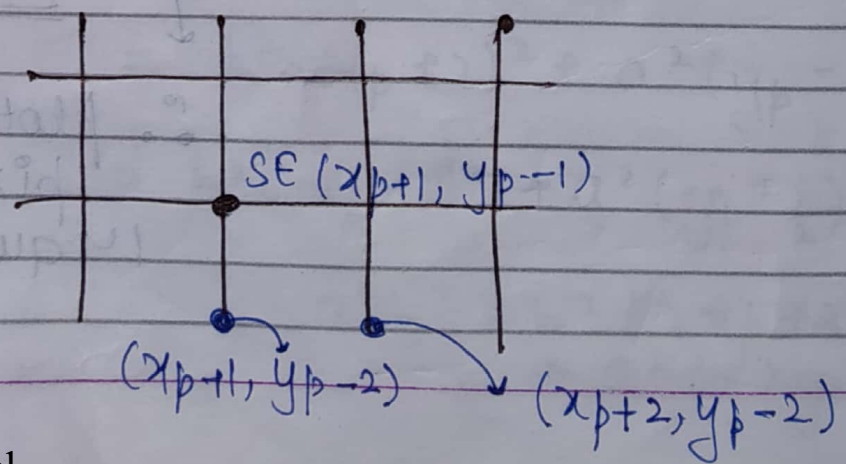
$\therefore$ mid pt $\rightarrow (x_p + 1/2, y_p - 1)$

$$d_{old} = f(x_p + \tfrac{1}{2}, y_p - 1)$$

$$= b^2\left(x_p + \tfrac{1}{2}\right)^2 + a^2(y_p - 1)^2 - a^2b^2$$

If $d_{old} < 0$, choose SE as mid pt. lies inside boundary & boundary is closer to SE

If we choose SE



https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

mid pt → $\underline{\underline{x_p + \frac{3}{2}, y_p - 2}}$

$$d_{new} = F\left(x_p + \frac{3}{2}, y_p - 2\right)$$

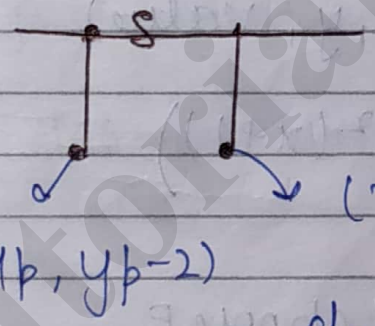
$$= b^2\left(x_p + \frac{3}{2}\right)^2 + a^2(y_p - 2)^2 - a^2b^2$$

$$= b^2\left(x_p + \frac{1}{2}\right)^2 + b^2(+2x_p + 2) + a^2(y_p - 1)^2 + a^2(-2y_p + 3)$$

$$= d_{old} + \underbrace{b^2(2x_p + 2) + a^2(-2y_p + 3)}_{\Delta SE}$$

(4)

If $d_{old} > 0$ choose $S(x_p, y_p - 1)$



midpt. I

$$\left(x_p + \frac{1}{2}, y_p - 2\right)$$

$$d_{new} = F\left(x_p + \frac{1}{2}, y_p - 2\right)$$

$$= b^2\left(x_p + \frac{1}{2}\right)^2 + a^2(y_p - 2)^2 - a^2b^2.$$

$$= b^2\left(x_p + \frac{1}{2}\right)^2 + a^2(y_p - 1)^2 + \underbrace{a^2(-2y_p + 3)}_{\Delta S}$$

(5)

Contributor: Riya Goel

https://www.tutorialsduniya.com

d_0 for region 2

↓

$(x_p + 1/2, y_p - 1)$

Where the region 1 finishes, the last pt. $(x_p + \frac{1}{2}, y_p - 1)$ will give us the d_0 for region 2.

$d_0 = F(x + \frac{1}{2}, y - 1)$ where (x, y) is last pt. of Region 1.

* ~~Algorithm~~ ↓

$x = 0, \; y = b$

$d_0 = d_{OR1} = b^2 - a^2 b + \frac{a^2}{4}$

Plot ellipse pts. (x, y, value)

while $(a^2(y - \frac{1}{2}) > b^2(x+1))$

{

if $d < 0$ // choose E

$d_0 \mathrel{+}= b^2(2x+3)$

else

{

$d_0 \mathrel{+}= b^2(2x+3) + a^2(-2y+3)$;

// select SE.

$y--$;

}

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

```
    x++;
    plot ellipse pts.
}
```

last x and y values from 1st while loop.

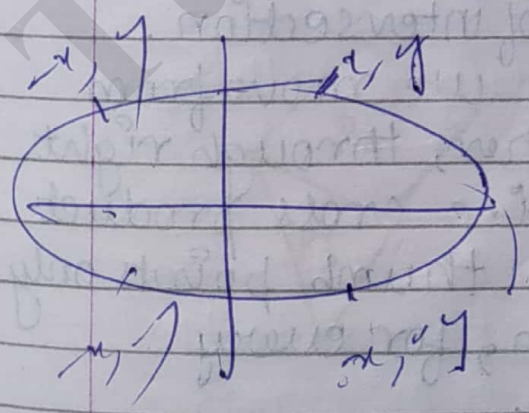
$$d_0 = d_{0R2} = b^2\left(x+\frac{1}{2}\right)^2 + a^2(y-1)^2 - a^2b^2$$

```
while (y>0)
{
    if (d0<0)
    {
        d0 += b^2(2x+2) + a^2(-2y+3);
        x++;
    }
    else
    {
        d0 += a^2(-2y+3);
        y--;
    }
    plot ellipse (      );
}
}
```

at y=0 we have touched the x axis and thus now we are in R2.

NOTE: Take all values of variables as double.

as there is problem of precision.



Tutorialsduniya.com

TutorialsDuniya.com

Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc of following subjects from https://www.tutorialsduniya.com

- C and C++
- Programming in Java
- Data Structures
- Computer Networks
- Android Programming
- PHP Programming
- JavaScript
- Java Server Pages
- Python
- Microprocessor
- Artificial Intelligence
- Machine Learning
- Computer System Architecture
- Discrete Structures
- Operating Systems
- Algorithms
- DataBase Management Systems
- Software Engineering
- Theory of Computation
- Operational Research
- System Programming
- Data Mining
- Computer Graphics
- Data Science

- Compiled Books: https://www.tutorialsduniya.com/compiled-books
- Programs: https://www.tutorialsduniya.com/programs
- Question Papers: https://www.tutorialsduniya.com/question-papers
- Python Notes: https://www.tutorialsduniya.com/python
- Java Notes: https://www.tutorialsduniya.com/java
- JavaScript Notes: https://www.tutorialsduniya.com/javascript
- JSP Notes: https://www.tutorialsduniya.com/jsp
- Microprocessor Notes: https://www.tutorialsduniya.com/microprocessor
- OR Notes: https://www.tutorialsduniya.com/operational-research

https://www.tutorialsduniya.com

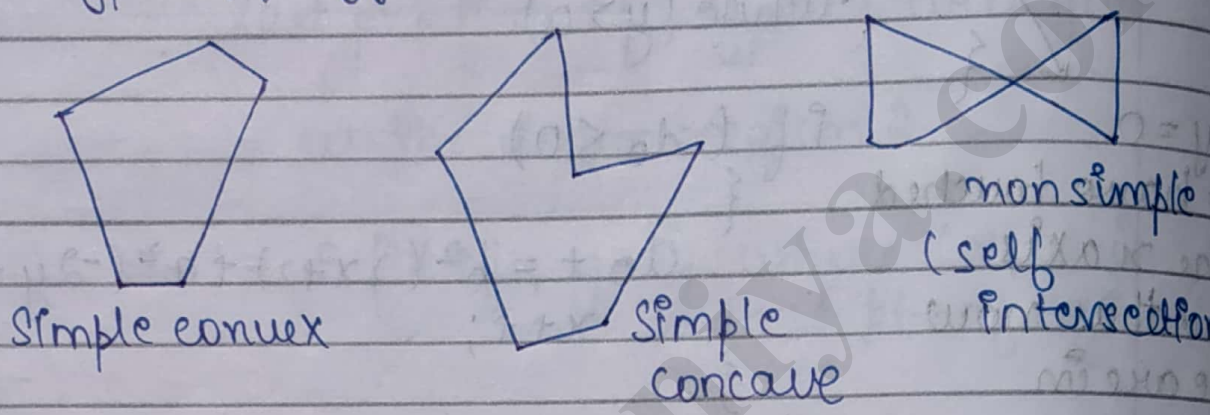
28 Jan, 19

Filling Polygons

↓

Filling colours in polygons.

3 types of polygons



On a system, we don't first make outline and then fill colours. But filling and drawing goes hand in hand
∴ colouring is done with scan conversions.

all algos of DDA, Bresenhams
(The name is so because we scan each line i.e scan line).

- Convex polygon - only 2 pts. of intersection. For a given edge vector, we move from one edge to another, through right hand thumb rule i.e cross product of 2 edge vectors, then thumb points only in one direction, for every edge vector.

Tutorialsduniya.com

https://www.tutorialsduniya.com

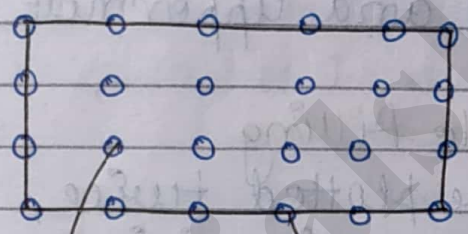
CLASSTIME Page No.
Date / /

- Concave polygon

If polygon is concave, the part where concavity starts, the z component comes out to be -ve.
And for a concave polygon, we can have multiple pts. of intersection.
Also, acc. to thumb rule, thumb doesn't always points in one direction.

NOTE- Filling polygon algo. are only based for convex polygons.
For concave polygons, we divide them into diff. convex polygons.

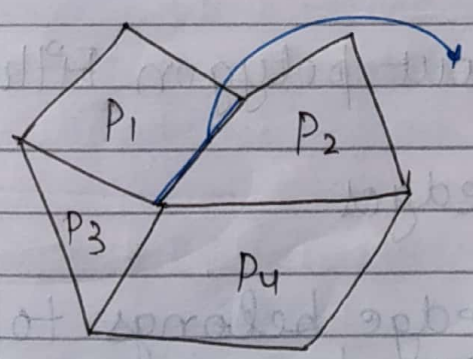
What pixels to be considered ?



These pixels lie inside polygon

They lie on the boundary.

Now we need to decide whether these pts. must be painted or not.



This is a shared edge/ Shared pixels of P_1 and P_2
If P_1 is drawn first these pixels get red color or else yellow if P_2 is plotted.

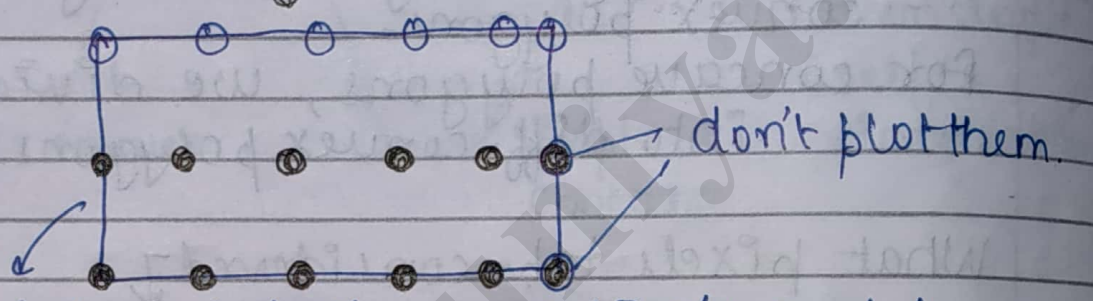
Tutorialsduniya.com

https://www.tutorialsduniya.com

To get rid of this problem, we need to have some common criteria.

So, for shared pixels, we must have some method.

For adjacent polygon figures



plot the left and bottommost pixels and leave the right and uppermost pixels.

→ Disadvantages of Double Filling

(1) If shared pixels are plotted twice, algo becomes inefficient

(2) If polygon have diff color, then final color depends on the order in which polygon is drawn.

→ General Ideas about polygon Filling

- Rules for Shared edges

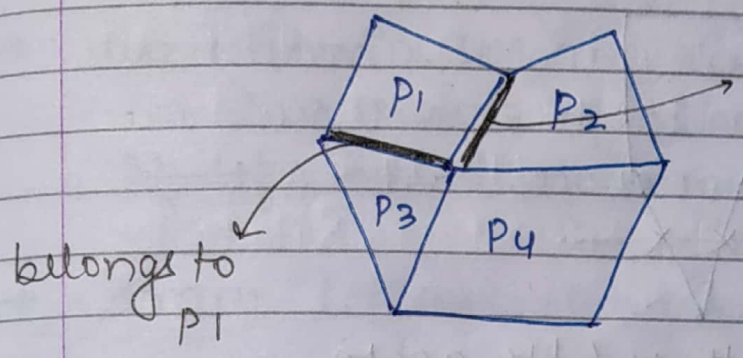
- A shared vertical edge belongs to the rightmost of the two sharing shapes

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

- A shared non vertical edge belongs to upper shape.

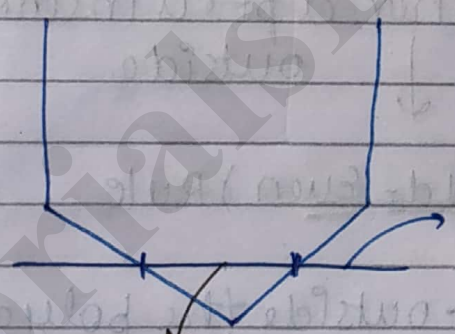


bottom and left pixels will be part of primitive.

- Fill in polygons by computing intersections of boundaries with scan lines.

→ Edges of polygon are to be stored in form of vertices.

- Span
- coherence
- Spatial coherence
- edge coherence



{collection of adjacent pixels on single scan line which lie inside primitive} (Span)

Spatially coherent (as they will be getting same colour)

→ The algo stops when y attains its max. value.

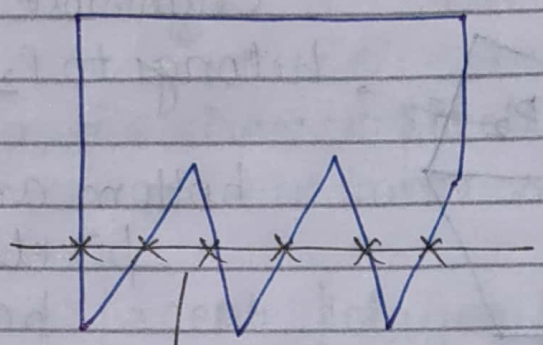
Algorithm -

(1) Find intersection of scan line with all edges of polygon. (min = 0 and max = ∞)

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

2. Sort intersections by increasing x coordinates.



Sort these 6 pts. acc to ascending order.

3. Fill in all pixels b/w pairs of intersections that lie in interior to polygon.

→ How do we find and sort intersections.

→ How to find whether a pt. is inside or outside

Imp.

Parity (odd-Even) Rule (6-7 marks)

Begin from a pt. outside the polygon, counting no. of edges crossed so far, a pixel is inside polygon if no. of edges crossed so far (parity) is odd and outside if no. of edges crossed is even. This is known as odd-even parity rule.

Q Dry-run the algo given initial & final pts. (circle, line, ellipse) Make table. (6-7 marks)

TutorialsDuniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Q How to deal with special case of intersection at integer pixel coordinates

↓

→ use criteria we are using for avoiding conflicting b/w shared edges

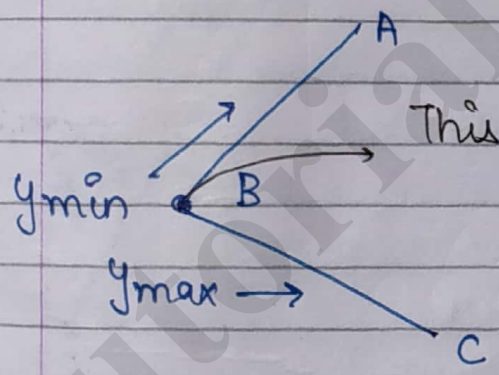
~~If uppmost pixel in a span has~~

→ draw leftmost and bottommost pixels.

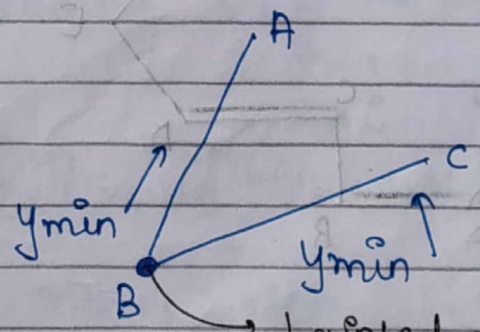
→ How to deal with special case of shared vertices

↓

count ymin vertex of an edge in parity calculations but not ymax vertex.



This will be plotted with A.



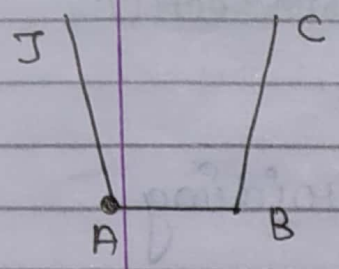
painted only once. (due to parity)

→ Bottom edges are drawn but not top edges.

Tutorialsduniya.com

https://www.tutorialsduniya.com

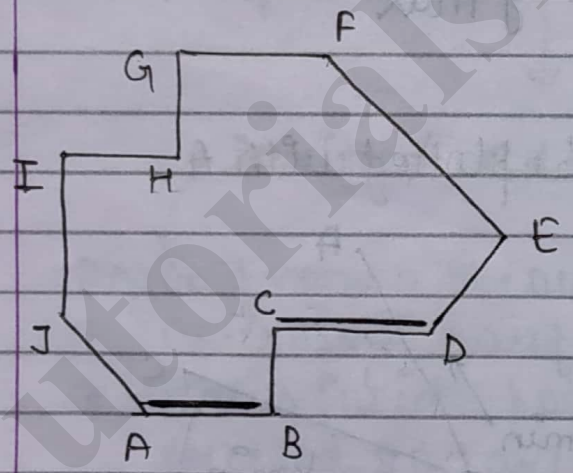
y_{min} - contribute to parity
y_{max} - no contribution of parity.



A is y_{min} for AJ ∴ parity becomes odd from initial 0 parity. Now we encounter line AB, which is a horizontal line and no horizontal line, we have no concept of y_{min} and y_{max} ∴ horizontal line doesn't contribute to parity rule.

At B → B is y_{min} for BC ∴ parity changes from odd to even.
∴ donot print B.

NOTE:- Parity will only change for a particular scan line with y_{min} as intersection or any other pixel except y_{max}.



Tutorialsduniya.com

TutorialsDuniya.com

Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc of following subjects from https://www.tutorialsduniya.com

- C and C++
- Programming in Java
- Data Structures
- Computer Networks
- Android Programming
- PHP Programming
- JavaScript
- Java Server Pages
- Python
- Microprocessor
- Artificial Intelligence
- Machine Learning
- Computer System Architecture
- Discrete Structures
- Operating Systems
- Algorithms
- DataBase Management Systems
- Software Engineering
- Theory of Computation
- Operational Research
- System Programming
- Data Mining
- Computer Graphics
- Data Science

- Compiled Books: https://www.tutorialsduniya.com/compiled-books
- Programs: https://www.tutorialsduniya.com/programs
- Question Papers: https://www.tutorialsduniya.com/question-papers
- Python Notes: https://www.tutorialsduniya.com/python
- Java Notes: https://www.tutorialsduniya.com/java
- JavaScript Notes: https://www.tutorialsduniya.com/javascript
- JSP Notes: https://www.tutorialsduniya.com/jsp
- Microprocessor Notes: https://www.tutorialsduniya.com/microprocessor
- OR Notes: https://www.tutorialsduniya.com/operational-research

https://www.tutorialsduniya.com

- Edge Coherence

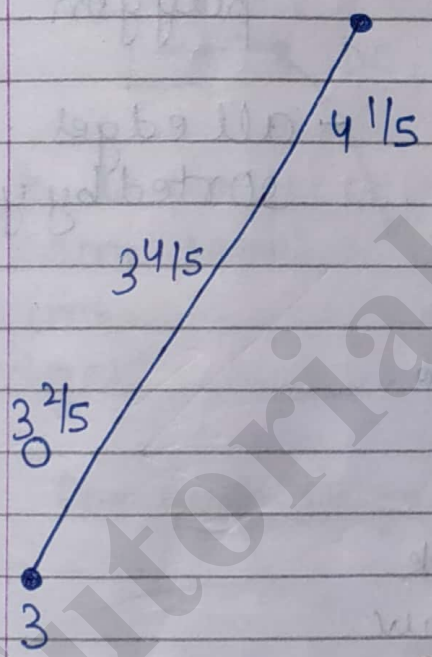
If an edge intersects with a scan line, and slope of edge is m, then successive scan line intersections can be found from

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

(scan line count)

$$\frac{1}{m} = (x_1 - x_0) / (y_1 - y_0)$$

The floating pt. arithmetic can be avoided by storing numerator comparing to the denominator, and incrementing x when it overflows.



$$m = \frac{6-1}{5-3} = \frac{5}{2}$$

$$\frac{1}{m} = \frac{2}{5}$$

$$x_{min} = 3$$

$\therefore$ seq $\rightarrow$ $3 + \frac{2}{5} \Rightarrow 3\ 2/5$ $+2/5$

x_i

x_i $3\ 4/5$, $3\ 6/5 = 4\ 1/5$

num > den.

$\therefore$ rather than adding $1/m$ we don't increment x value unless num becomes > than den.

$\therefore$ increment $x \rightarrow 4$ from 3.

$\therefore 4\frac{1}{5}$

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

{ numerator → 2 → 4 → 1
denominator → 5. }

- Algo:

(1) We need a DS to store the polygon.
2 data structures

AET
Active edge Table
- Stores info abt. all edges which are being currently intersected by scan line.
- Stores edges intersected by scan line y sorted by x coordinate
- Each entry in AET contains ymax coordinate of edge, the x coordinate of intersection pt. and 1/m.

→ to check whether the previous edge/polygon part is painted or not.

GET
Global edge Table.
- DS which is used to store edges of the polygon.
- all edges sorted by ymin

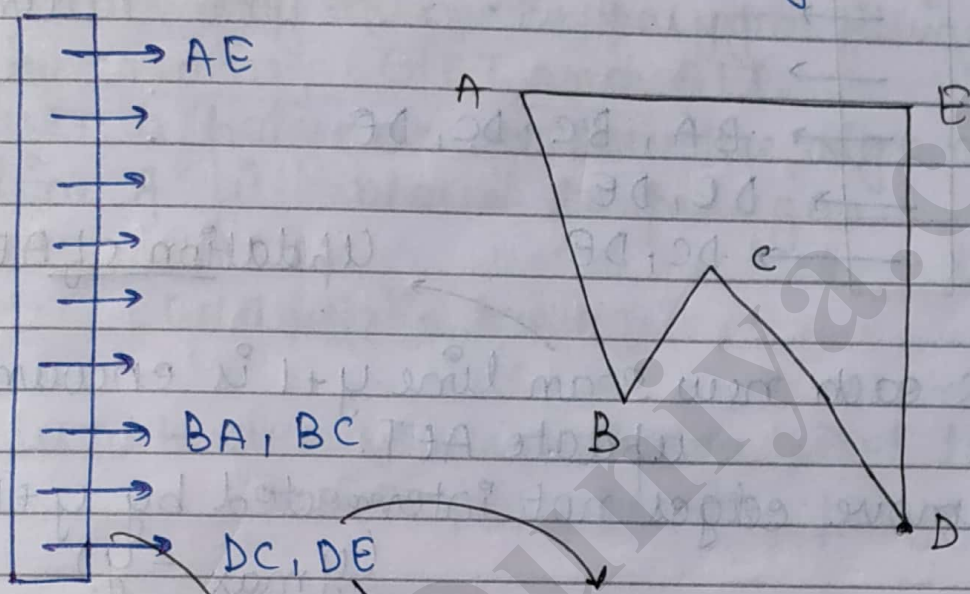
TutorialsDuniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

- GET with all edges sorted by y_{min}
- AET containing
  - edges that intersect current scanline.
  - Their pts. of intersection.
  - Sorted by x coordinate left to right.

GET



Have same y_{min}

y_{min} is on same level for same edges.

(no. of entries = no. of scanlines)

sorting on basis of x coordinate.

- For each edge, we need.

x_{min} and y_{max} and slope of edge.

y_{min} is not req. as we are sorting acc. to that only.

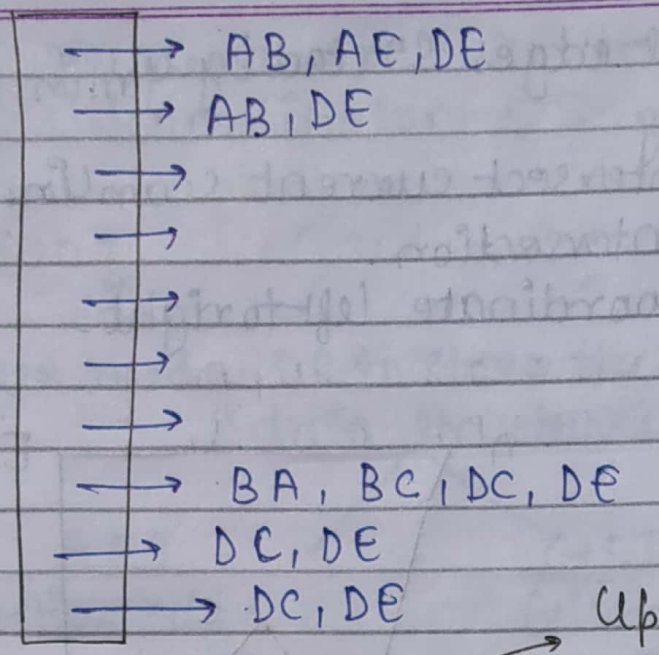
Stored in GET and not AET → stores x coordinate as pt. of intersection. and m and y_{max}.

for calculating x (pt. of intersection)

$x_n = x_0 + 1/m$

Contributor: Riya Goel

https://www.tutorialsduniya.com



{ info. of edges currently intersected by a scan line }.

Updation of AET

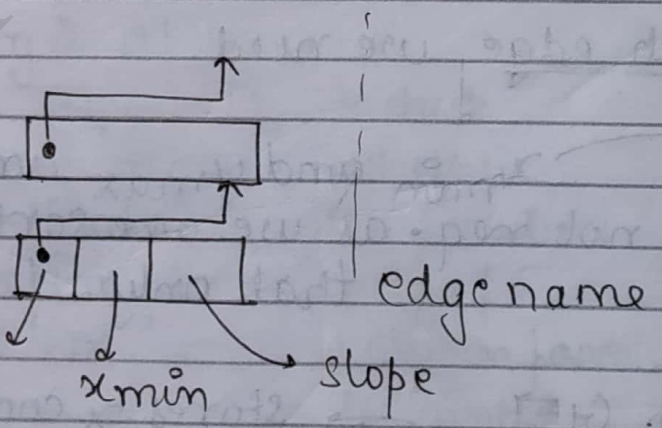
→ As each new scan line y+1 is encountered, update AET.

(1) Remove edges not intersected by y+1 ($y_{max} = y$)

(2) Add edges intersected by y+1 ($y_{min} = y+1$) from GET

(3) Calculate new x intersections. $\{x_{new} = x + \frac{1}{m}\}$

- GET



now fill the AET using values from GET. and we continue algo until we get empty AET.

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

- Scan line Algorithms

— Find & sort intersections
— Fill spans

} 2 subproblems for polygon filling.

Q (in exam) dry run the polygon filling algo. by creating GET and AET.

Q Given a polygon, follow the algo of scan line & fill colour to polygon.

4th Feb, 19 Odd parity rule

used to decide whether a pixel lies inside / outside the polygon.

* Scan line Algorithm

(1) Set y to the smallest y coordinate that has an entry in GET, that is 'y' for the first non empty bucket.

(2) Initialise AET to be empty.

(3) Repeat until AET and GET are empty.

→ Move from GET bucket y to AET those edges whose $y_{min} = y$ (entering edges)

→ Remove from AET those entries for which $y = y_{max}$ (edges not involved in next scan line) then sort AET on x (made easier because GET is pre-sorted).

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

→ Fill in desired pixel values on scan line y by using pairs of x coordinates from AET.

→ Increment y by 1 (to coordinate of next scan line).

(For vertical edge, x is same)

→ For each non vertical edge remaining in AET, update x for new y.

Also, for existing edges in AET, update their x coordinates for finding pt. of intersection with scan line.

(Update x for new value of y)

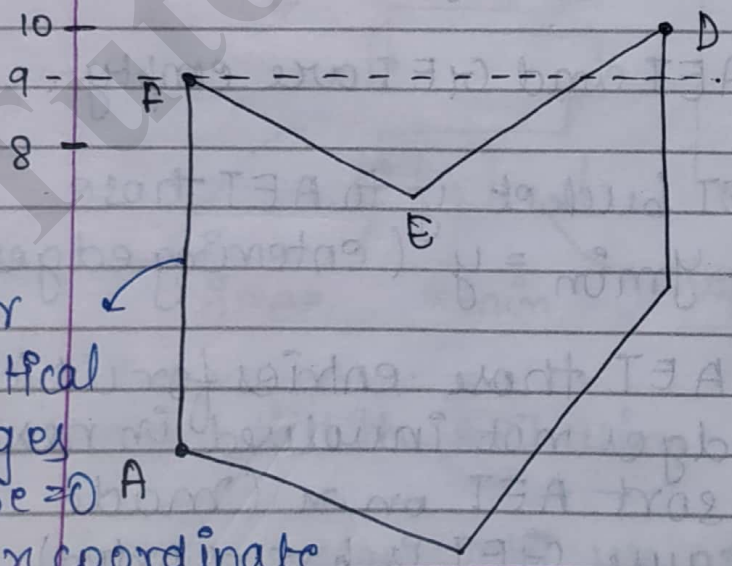
$$x_{new} = x + \frac{1}{m}.$$

→ For GET and AET

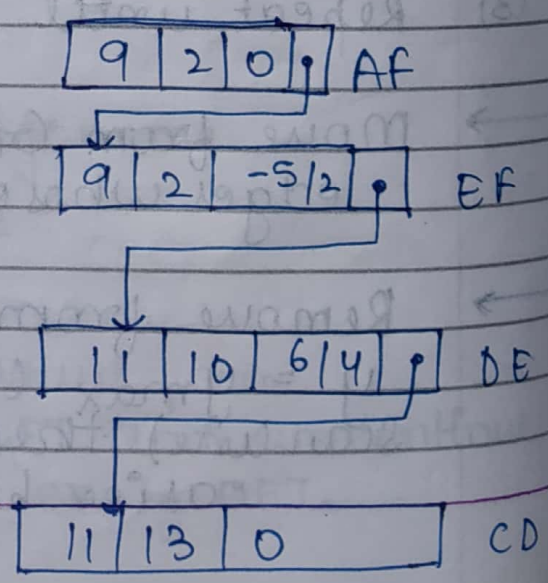
array linked list req.

→ Singly linked list as it is updated for every scan line.

→ We also need to keep a record of edges.



for vertical edges slope = 0 ∴ x coordinate remains same



Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

→ We are on scan line 9
Now we see that at scan line we have y_{max} of FE and FA as 9
∴ remove AF and FE.

And if there is an edge with $y_{min} = 9$ then add to AET.

Now increment $y \rightarrow y+1$ i.e 10

scan line no. 10.

Also increment x (pt. of intersection)

- Outline :-

(1) Drawing with Thick primitives
(2) Halftone Approximation.
(3) Anti-aliasing

→ One of the attributes to draw a figure / primitive is thickness

In all examples above, we had 1 pixel thickness.

* How to draw a thick primitive

(1) We can do pixel replication to ↑se thickness but will depend upon the slope. (either row-wise or column-wise).

by replicating pixels.

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

→ Column (Row) Replication :-

(1) → pixel replication is col. wise

(2) → pixel replication is row wise

But for steep and gradual slope

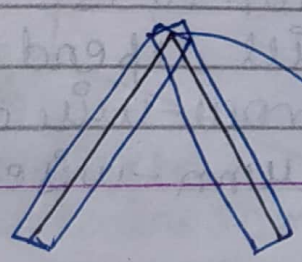
$|m|>1$ → $\frac{dy}{dx}>1$ → replication is column wise.

$|m|<1$ → replication is row wise

- Disadvantages :-

(1) One of the problem with replication is even width.
(primitives will not have uniform width)
Pixel replication is better with odd width.

(2) another problem is with ends i.e pixel ends are flattened



gap is encountered since ends gets flattened.

TutorialsDuniya.com

https://www.tutorialsduniya.com

→ Increasing width will enlarge/ shrink image..

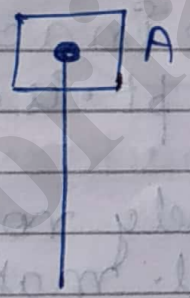
CLASSTIME Page No.
Date / /

When we join them we get gaps at corners & they need to be filled.

(3) When we draw lines with equal thickness, then perceived thickness is not proper, i.e seen thickness will vary with diff. slopes.
Max. thickness is seen for horizontal & vertical line & the perceived thickness of a line with a particular slope is generally becomes as $\frac{t}{\sqrt{2}}$ → reduces

→ 2nd method ↓
A moving pen

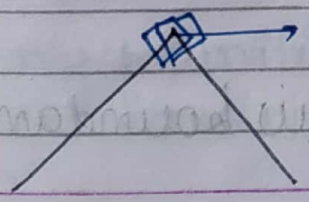
- Replace a pixel with a matrix of pixels.



Suppose a grid of pixels is of 3X3

Now in place of point A, it is replaced by a grid of pixels.

The problem of flattened ends gets solved. gap gets filled by pixels.



→ no gap here.

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

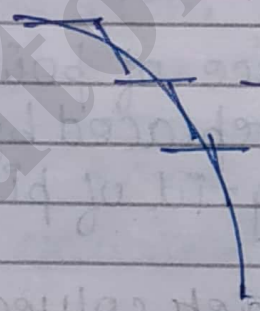
→ Disadvantages -

(1) Pixel repetition.
↓
There is overlapping of pixels.

(2) Here, in this case there is problem of perceived thickness.
↓
Slope with a particular line appears thickest and vertical & horizontal lines appear less thick.

opp. to that of 1st method

(3) For curved figures.
↓
eg - circle (Here in a circle, slope at each pt. is diff.)



→ alternatively row & col. matrix colouring.

→ 3rd method ↓

Filling Areas B/w boundaries

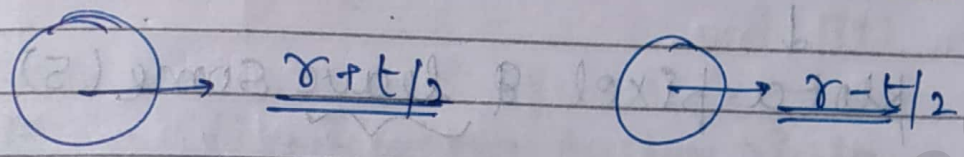
Instead of drawing pixels, we fill area above & below the line.

Contributor: Riya Goel

https://www.tutorialsduniya.com

We make a rectangle & fill pts. b/w that rectangle.

eg - To draw a circle of thickness 't' we draw 2 concentric circles

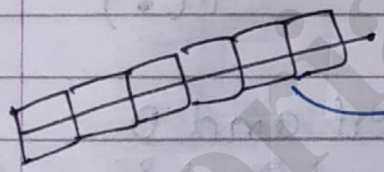


* 4th method ↓

any pixel with thick polylines.

multiple lines

approximate a line with multiple lines.



making non overlapping grids.

→ 2nd attribute of any primitive is "colour"
putpixel (x, y, color).

→ 3rd attribute → style/pattern.

any primitive drawn till now was of solid style

Tutorialsduniya.com

https://www.tutorialsduniya.com

→ For dashed lines (- - - -)
↓
plot pixels and leave some.

→ For dotted lines (. . . . .)
↓
plot a pixel & leave some (5)

leave means don't send them in putpixel, just evaluate them.

- Dashed dotted lines (— . — . —)
↓
Take an array with bit values of 0 & 1's.

1	1	1	1	1	0	0	1	0	1	1	- -

5 pixels make a (—) | for spaces | 1 pixel for 1 (.)

1 means pixel is plotted and 0 means no pixel is plotted i.e a blank.

To repeat this procedure, we do modular arithmetic.
↓
(circular array) → index % maxsize + 1

at last index, after evaluation it again returns to 1st index.

Tutorialsduniya.com

TutorialsDuniya.com

Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc of following subjects from https://www.tutorialsduniya.com

- C and C++
- Programming in Java
- Data Structures
- Computer Networks
- Android Programming
- PHP Programming
- JavaScript
- Java Server Pages
- Python
- Microprocessor
- Artificial Intelligence
- Machine Learning
- Computer System Architecture
- Discrete Structures
- Operating Systems
- Algorithms
- DataBase Management Systems
- Software Engineering
- Theory of Computation
- Operational Research
- System Programming
- Data Mining
- Computer Graphics
- Data Science

- Compiled Books: https://www.tutorialsduniya.com/compiled-books
- Programs: https://www.tutorialsduniya.com/programs
- Question Papers: https://www.tutorialsduniya.com/question-papers
- Python Notes: https://www.tutorialsduniya.com/python
- Java Notes: https://www.tutorialsduniya.com/java
- JavaScript Notes: https://www.tutorialsduniya.com/javascript
- JSP Notes: https://www.tutorialsduniya.com/jsp
- Microprocessor Notes: https://www.tutorialsduniya.com/microprocessor
- OR Notes: https://www.tutorialsduniya.com/operational-research

https://www.tutorialsduniya.com

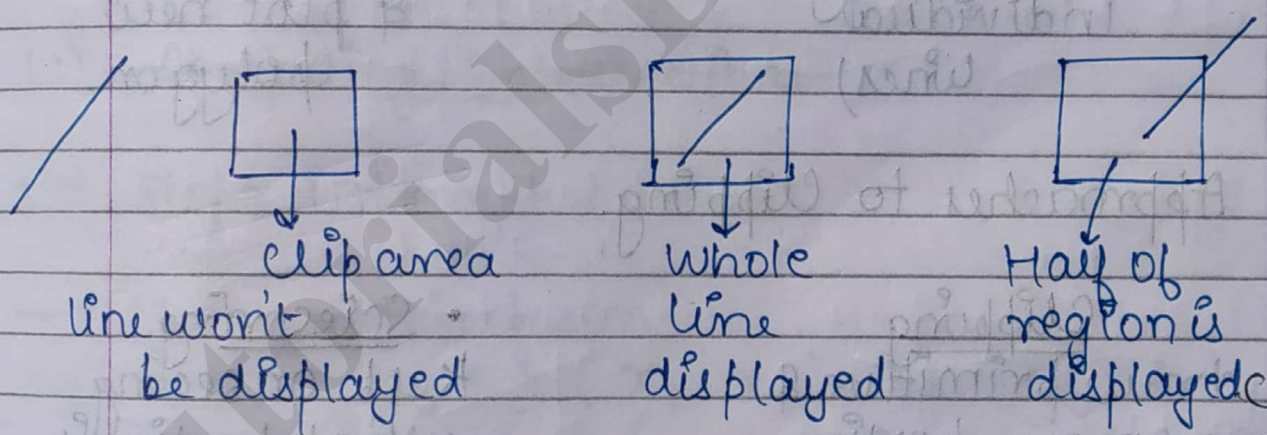
(5th algorithm)

→ For every line, we will now add color, thickness and style.

NOTE:- We can implement any pattern using 'Bitstring'.

6th Feb, 19 Clipping on a Raster Display (3rd algorithm)

- Remove points outside a region of interest
- Show an image in a particular area of screen. (clip area)
- Discard (parts of) primitives outside of window.



can be of any shape (circle--)

To be considered
↓
{ primitive - line.
clip area - window }

- Point clipping -
- Remove pts. outside window. (by comparing coordinates of pt. & clip

Contributor: Riya Goel

https://www.tutorialsduniya.com

$x_c > x_{min}$ and $x_c < x_{max}$ → pt. is inside clip window

window).

- A pt. can either be entirely inside / outside.

• Line Clipping.
- Remove portion of line segment outside window.

• Polygon Clipping
- Remove portion of polygon outside window

clip each edge for window (not clipping individual lines)

find vertices of region lying inside polygon in clip area & plot new polygon.

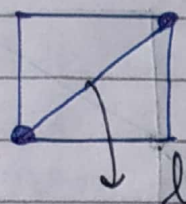
→ Approaches to Clipping.

• Clipping
draw primitive & remove portion outside window (first plot entire primitive & then check)

• Scissoring
Keep on seeing what portion is inside clip window & plot primitive. (find pts. of intersection of line with clip region)

Contributor: Riya Goel

https://www.tutorialsduniya.com



line completely inside clip window

- scan convert after finding out which portion is inside clip window. & draw only that portion.

Clipping (End pts)

→ Inside $x_{min} <= x <= x_{max}$

$y_{min} <= y <= y_{max}$.

* One of the basic methods to clip line

(1) Computing Intersections

→ Slope intercept form.

→ parametric form

$x = x_0 + t(x_1 - x_0)$, $y = y_0 + t(y_1 - y_0)$

end pt.

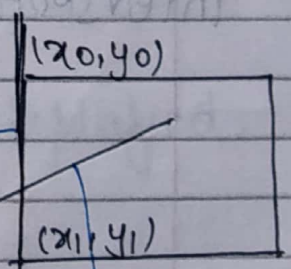
parameter t in [0, 1].

If t is b/w 0 & 1

pt. of intersection is inside on clip window

1 t for edge and 1 for line.

edge representation by parametric form



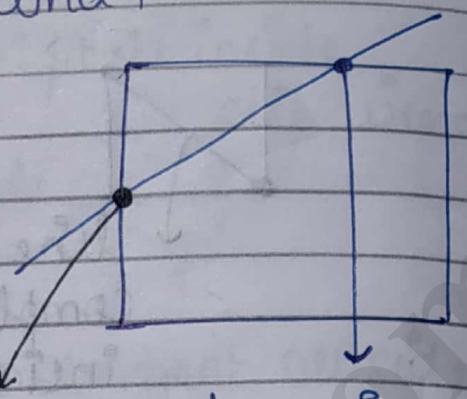
line representation by param. form.

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

If t is outside 0 and 1 then, pt. lies outside clip window



Brute force techniques.

t_{line} = t_{edge} = lies b/w 0 and 1

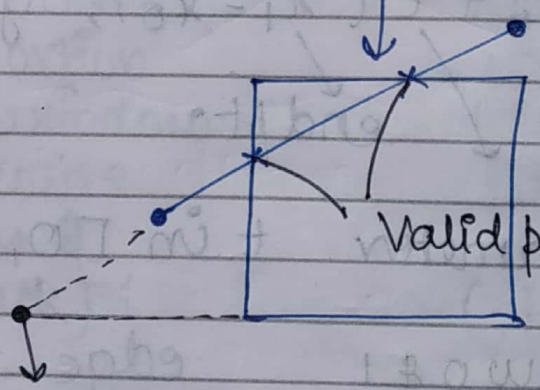
t_{line} is outside 0 & 1
t_{edge} is inside 0 & 1

Find parameter t b/w line & b/w each edge.

max. 2 pts of intersection
choose 2 values of t b/w 0 & 1 → Valid pt. of intersection of x and y.

If both end pts. inside window
2 valid pts. of intersection.

If both end pts. outside window



This is t value b/w bottom and right edge. But since it is outside 0 and 1 → discard.

If one end pt. is inside window
1 valid pt of intersection

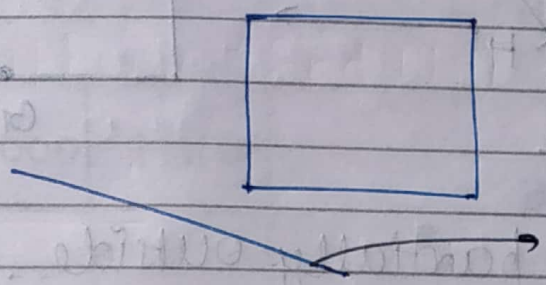
Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

- Solve for t edge and t line
  If both lie in [0, 1] then real intersection.
- Still need to test for lines ∥ to clip rectangular edges.

Still involves much calculations unnecessary.



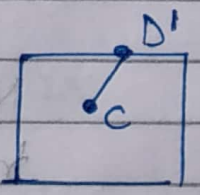
Both end pts. lie outside region but all values of t are calculated & then result is evaluated.

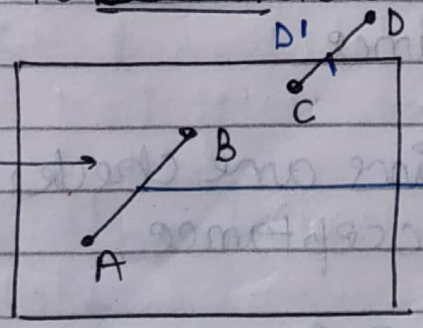
(2) Cohen Sutherland Algo (line)

(3) Sutherland Hodgeman Algo (polygon clipping).

→ Line clipping.

cases to be considered :-





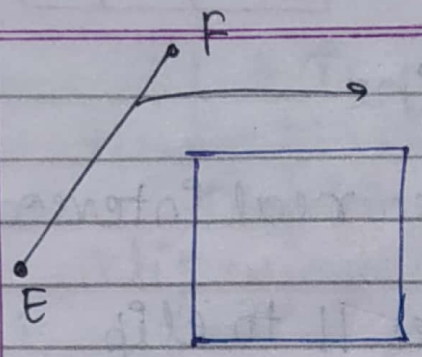
(i) → entire line displayed.

Contributor: Riya Goel

https://www.tutorialsduniya.com

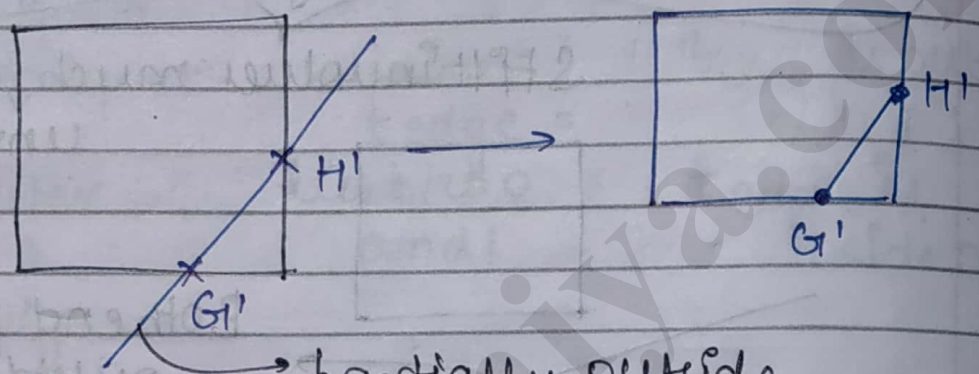
CLASSTIME Page No.
Date / /

(iii)



→ whole line rejected.

(iv)



→ partially outside.

→ Cohen Sutherland algo

(1) performs initial tests on a line to determine whether intersections calculations can be avoided.

If initially we can predict whether line is

accepted → trivial acceptance

rejected.

→ First end point pairs are checked for Trivial acceptance.

→ If line can't be trivially accepted, region checks are done for Trivial

Contributor: Riya Goel

https://www.tutorialsduniya.com

rejection.

→ If line can neither trivially accepted or rejected, it is divided into 2 segments at a clip edge. (9 diff. regions)

We assign each region a region code / bit code
min 4 bits are req. to identify a region.

1001	1000	1010
0001	0000	0010
0101	0100	0110

If line intersects at all bits 0000 then line is inside region.

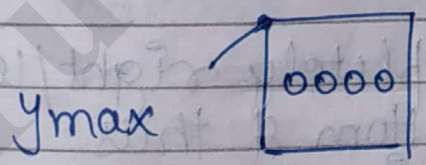
0 bit is set

bit 1 → $y < y_{min}$

bit 2 : $x > x_{max}$

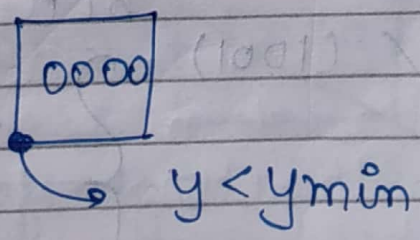
bit 3 : $x < x_{min}$

bit 4 : $y > y_{max}$



Now $y > y_{max}$

1001 1000 1010



0101 0100 0110

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

0001

for bit 3

$x < x_{min}$ ✓

$x \not> x_{max}$ → bit 2 becomes 0

$x \not< y_{min}$ → bit 1 will be set to 0

$x \not> y_{max}$ → bit 0 is 0.

decided on basis of this whether line is trivially accepted/ rejected

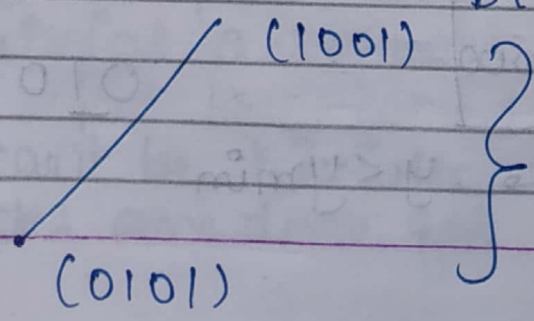
Trivially accepted ↓

code of end pts. is 0000.
draw line & no other calc.

- A line is trivially accepted if outcodes of both endpts. are zero

- A line is trivially rejected if bitwise AND of outcodes of endpts. is not zero.

either line is completely right/left/ top/ bottom & thus atleast there will be 1 common bit



totally towards left

Contributor: Riya Goel

https://www.tutorialsduniya.com

1 bit common.

```
 1001
 0101
 ----
 0001
```

→ 1 bit set i.e non zero value.

```
 1001
 0100
 ----
 0000
```

→ no common bit i.e zero value.

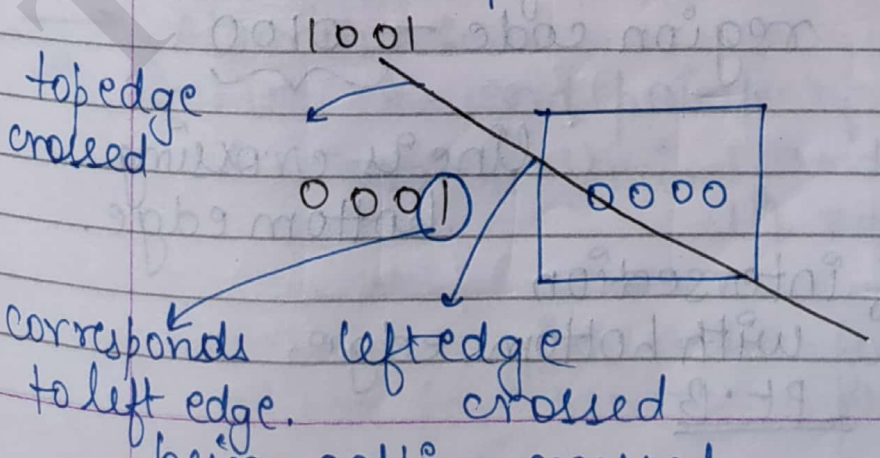
- All lines ∥ to clip window will be rejected.

acc. to above logic.

Although there are certain cases, where line segments are rejected but after calculations & not directly trivially rejected.

Disadvantage.

- A key property of outcode is that bits that are set in non zero outcode corresponds to the edges crossed.



non zero bit indicates the edge crossed.

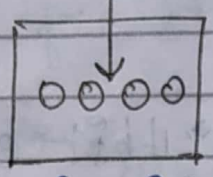
Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

1000

top edge
∴ intersects / crosses top edge.

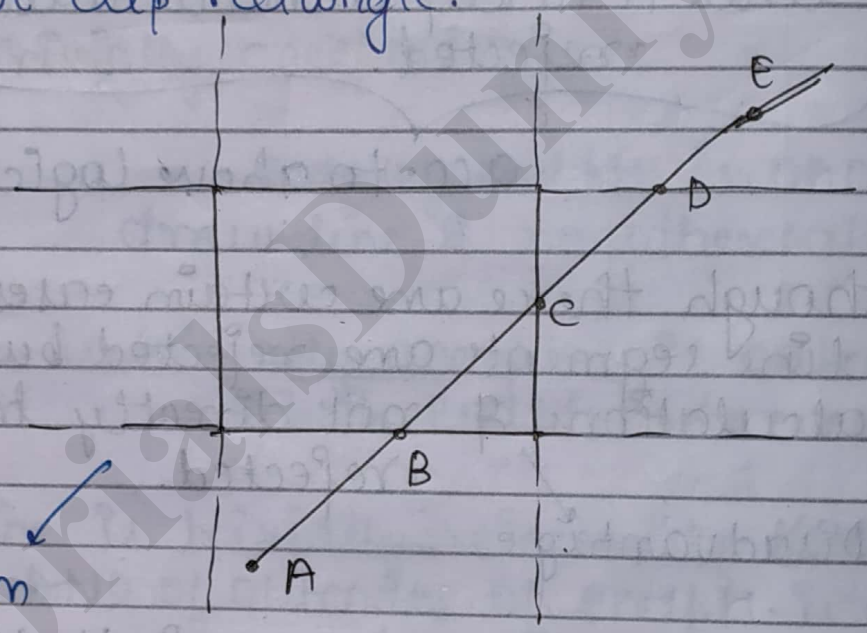


only to check when a line comes from outside to inside for pt. of intersection

We don't check for inside → outside as bits will be 0000.

It's clip edge & not clip rectangle.

eg



Start from a pt. outside clip region

Start from A
region code → 0100

line is crossing bottom edge.

Find pt. of intersection of line with bottom edge
Pt. B

Tutorialsduniya.com

Contributor: Riya Goel

TutorialsDuniya.com

Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc of following subjects from https://www.tutorialsduniya.com

- C and C++
- Programming in Java
- Data Structures
- Computer Networks
- Android Programming
- PHP Programming
- JavaScript
- Java Server Pages
- Python
- Microprocessor
- Artificial Intelligence
- Machine Learning
- Computer System Architecture
- Discrete Structures
- Operating Systems
- Algorithms
- DataBase Management Systems
- Software Engineering
- Theory of Computation
- Operational Research
- System Programming
- Data Mining
- Computer Graphics
- Data Science

- Compiled Books: https://www.tutorialsduniya.com/compiled-books
- Programs: https://www.tutorialsduniya.com/programs
- Question Papers: https://www.tutorialsduniya.com/question-papers
- Python Notes: https://www.tutorialsduniya.com/python
- Java Notes: https://www.tutorialsduniya.com/java
- JavaScript Notes: https://www.tutorialsduniya.com/javascript
- JSP Notes: https://www.tutorialsduniya.com/jsp
- Microprocessor Notes: https://www.tutorialsduniya.com/microprocessor
- OR Notes: https://www.tutorialsduniya.com/operational-research

https://www.tutorialsduniya.com

Now find region code of B. (0000)

A (0100)

trivially rejected.

on the clip edge ∴ all zero's

Now consider E → (1010)

crossing right or top edge.

We need to see for algo. whether checking is for top first or right

E
D
} trivially rejected.

D (0010)

non zero value ∴ rejected

right edge intersection = C

C → 0000

C
B
→ Trivially accepted line.

→ Algorithm :-

Input the end points of line & coordinates (xmin, ymin) and (xmax, ymax) of clip region.

Compute region codes for endpts.

check for trivial acceptance

→ (both codes = 0000)

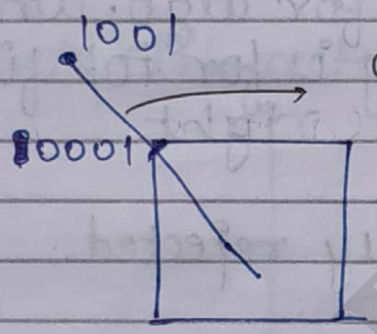
Contributor: Riya Goel

https://www.tutorialsduniya.com

If $\underbrace{\text{code}_1 \,\&\, \text{code}_2 \,!= 0}_{\text{reject.}}$

Take a pt. outside clip window.
↓
& code of pt. to top (1000)

1 will be there in AND only if the pt. crosses top.



crosses top edge.

Find pt. of intersection and again find region code.

Continue till both end pts. are inside / trivially rejected.

For finding pt. of intersection

Top → $x = x_0 + (x_1 - x_0) / (y_1 - y_0) \cdot (y_{max} - y_0)$
↳ we have y, find x.

Right → $y = y_0 + (y_1 - y_0) / (x_1 - x_0) * (x_{max} - x_0)$

Bottom → $x = x_0 + (x_1 - x_0) / (y_1 - y_0) \cdot (y_{min} - y_0)$

Contributor: Riya Goel

https://www.tutorialsduniya.com

left → $y = y_0 + (y_1 - y_0) / (x_1 - x_0) * (x_{min} - x_0)$.

↓

We have x to find y

- Polygon Clipping ↓

(1) apply Cohens Southerland on all line segments.

or

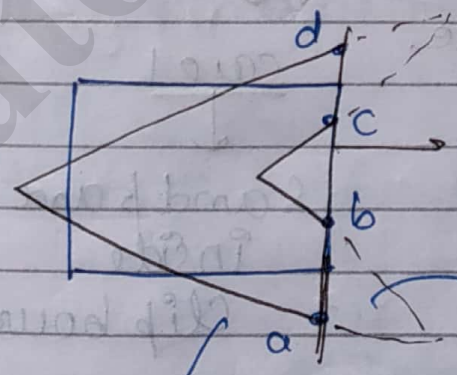
(2) Sutherland Hodgemen Algo.

↓

→ for a concave polygon, we will get more than 1 polygon after clipping.

→ for convex → only 1 polygon

- In this algo, we clip polygon against each edge seperately.



clipping by right boundary

polygon rejected on RHS.

addition of new vertices after right clipping

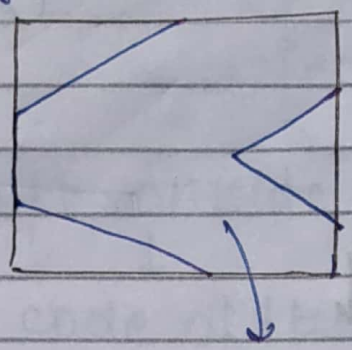
(a, b, c, d)

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Clipping from bottom → left → top.

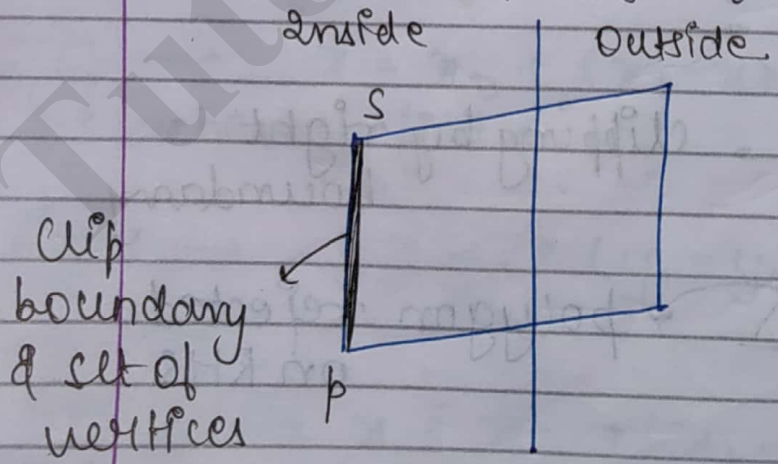


resultant fig.

(1) Initial set of vertices
(2) first passed through one clip boundary, and we remove all pts. outside boundary
(3) new vertices introduced at every stage
(4) again check against boundary,

Repeat it & final set of vertices to be plotted.

4 Cases of polygon filling ↓



Case 1
↓
S and P are inside clip boundary.

Tutorialsduniya.com

Contributor: Riya Goel

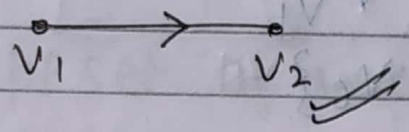
https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

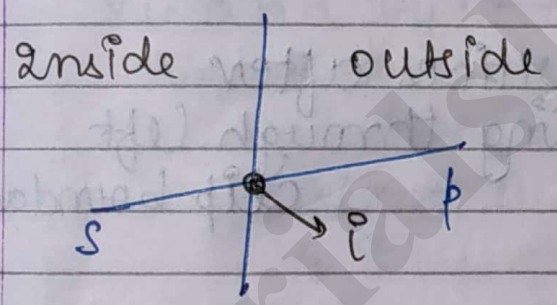
for top edge ↓

y of s and p must be < y_{max}.

Case 1 → If both vertices are inside clip boundary, then in O/P set we only include 2nd pt. / vertex & not first as polygon is closed so 1st pt. will be included afterwards.



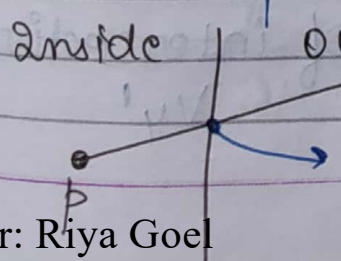
Case 2 → If out of 2, one is inside and other is outside, then O/P set of vertices will contain only pt. of intersection.



Case 3 - If both s & p are outside clip boundary, both are rejected in O/P set.

Case 4 → If 1st vertex is outside & 2nd is inside, then include 2 vertices in O/P set
(Reverse of case 2)

pt. of intersection & 2nd vertex.



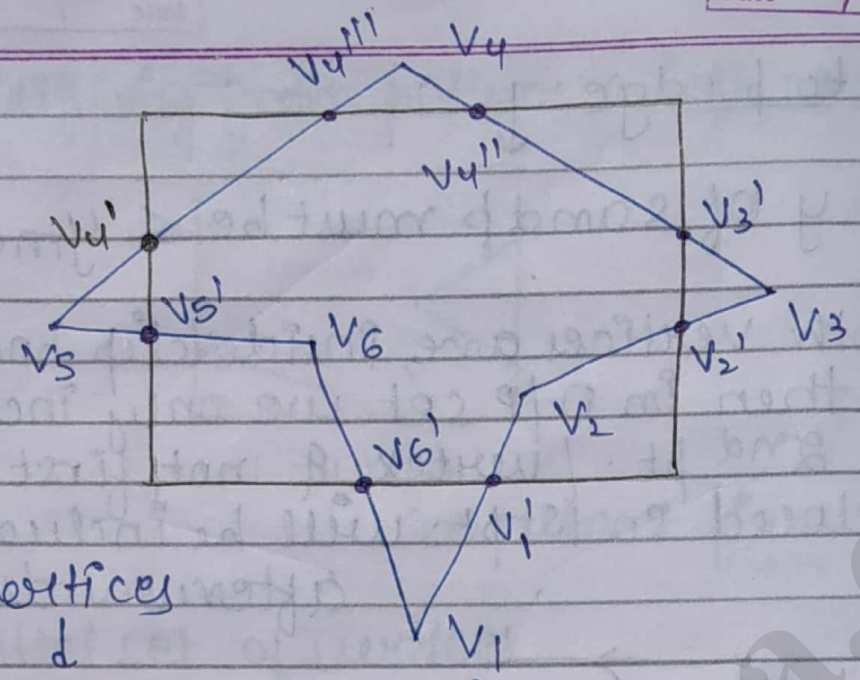
include i and p

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

eg:-

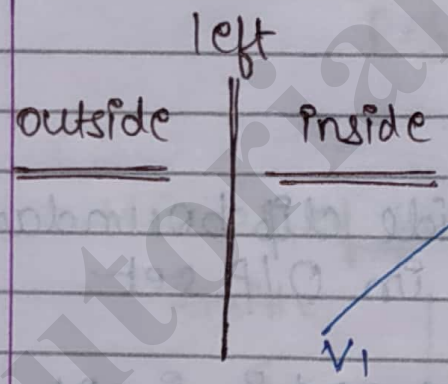


o/p set of vertices

$\{V_1, V_2, V_3, V_4, V_5, V_6\}$

Pass set of vertices to one of the clip boundary

→ left → o/p set of vertices after passing through left clip boundary



→ both inside V_2 ✓

V_2 & V_3 → V_3 ✓

V_3 & V_4 → V_4 ✓

V_4 & V_5 → pt. of intersection V_4'

V_4 → inside, V_5 → outside

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

V_5 and V_6 → pt. of intersection + 2nd pt.
↳ inside
outside

V_5' and V_6.

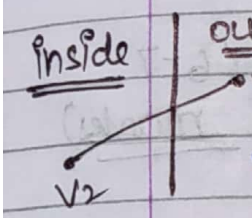
V_6 & V_1 → both inside V_1

not included in case 1

$V = \{V_2, V_3, V_4, V_4', V_5', V_6, V_1\}$.

↓

o/p set of vertices for 2nd boundary right



V_2 & V_3 → V_2' (pt. of intersection)
V_3 & V_4 → V_3' and V_4
V_4 and V_4' → V_4' (both inside)
V_4' & V_5' → V_5'
V_5' & V_6 → V_6
V_6 & V_1 → V_1
V_1 & V_2 → V_2

$\{V_2', V_3', V_4, V_4', V_5', V_6, V_1, V_2\}$

Now for top ↓

$\{V_3', V_4'', V_4''', V_4', V_5', V_6, V_1, V_2, V_2'\}$

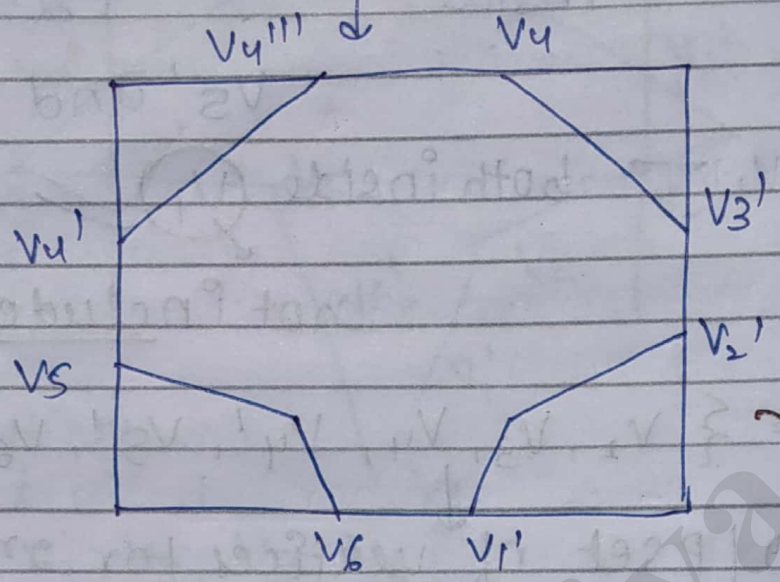
Now for bottom ↓

$\{V_4'', V_4''', V_4', V_5', V_6, V_6', V_1', V_2, V_2', V_3'\}$

Contributor: Riya Goel

https://www.tutorialsduniya.com

final set of vertices



Q (2m exam) either dry run algos + (6-7 marks)
algo writing

11 Feb, 19

Aliasing and Antialiasing.

- Aliasing - name given to jagged effect of lines.
- Jagged effect caused by finite set of pixels
- when change rows get step.
- Increase resolution steps (jaggies) get less step.
- } increase in resolution.
- double resolution → quadruple memory, bandwidth, and scan conversion time.

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Major drawback of Raster Graphics.
↓
Jagged lines

- The lines images is not smooth. This effect is Staircase problem or jagging.
This is called 'aliasing'.

Sol^n to these problems is 'anti aliasing' (reduces defect)

{This defect can't be removed but only reduced}

→ One simple sol^n → (1) Increase resolution so that many pixels will be plotted in a particular area.
But this actually doesn't reduce aliasing.

- 2 more techniques used are ↓ (based on area sampling)

Area sampling — Weighted / unweighted.

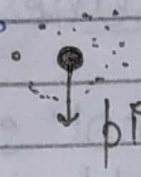
We see the pixel to be plotted covers how much area or how much it is close to the line.

- In this, we plot extra pixels but they are

Contributor: Riya Goel

https://www.tutorialsduniya.com

plotted with variable intensity / reduced intensity, depending on how much area of pixel is being covered by line.



pixel to be plotted.

Sharpness of staircase effect / jaggies will increase through area sampling

- Unweighted Area sampling -
  considering a pixel, every part of it contributes to intensity of pixel.

→ If line covers whole of pixel, we plot that pixel with full intensity.

every area of pixel contributes equally to intensity of pixel.

- Weighted Area Sampling -
  The area at centre covered will be plotted with more intensity
  eg - 2 pixels

more intensity. ✓ ○→ covers 1/4 at centre

○→ covers 1/4 at side.

* Unweighted Area Sampling :-
- Assume background white - lines black
- Recogonize that primitive has non zero width.

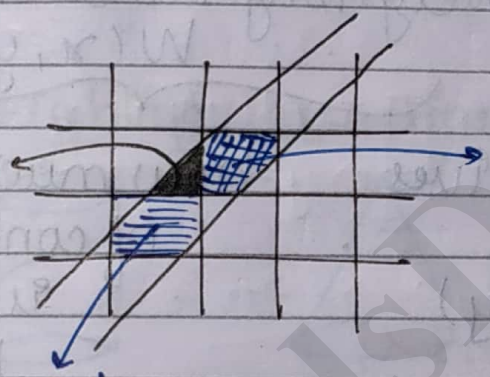
Contributor: Riya Goel

https://www.tutorialsduniya.com

(thinnest line - 1 pixel)

- Consider line as thin rectangle.
  covers diff. pixels to diff extent.
  squares.

- In most cases should not set a single pixel black.
  - Set intensity of pixel differently for each pixel covered.
  - Only horizontal & vertical lines effect only by 1 pixel per row.



area covered by a pixel.

plotted with less intensity, as area covered is less.

plotted with more intensity as area overlapped is more.

{If a pixel is away from line → less intensity.}

- works for non horizontal & vertical lines.

Geometry of pixels.
non overlapping squaretiles
lines contributes to intensity ∝ area of pixels.

Properties :-

Contributor: Riya Goel

TutorialsDuniya.com

Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc of following subjects from https://www.tutorialsduniya.com

- C and C++
- Programming in Java
- Data Structures
- Computer Networks
- Android Programming
- PHP Programming
- JavaScript
- Java Server Pages
- Python
- Microprocessor
- Artificial Intelligence
- Machine Learning
- Computer System Architecture
- Discrete Structures
- Operating Systems
- Algorithms
- DataBase Management Systems
- Software Engineering
- Theory of Computation
- Operational Research
- System Programming
- Data Mining
- Computer Graphics
- Data Science

- Compiled Books: https://www.tutorialsduniya.com/compiled-books
- Programs: https://www.tutorialsduniya.com/programs
- Question Papers: https://www.tutorialsduniya.com/question-papers
- Python Notes: https://www.tutorialsduniya.com/python
- Java Notes: https://www.tutorialsduniya.com/java
- JavaScript Notes: https://www.tutorialsduniya.com/javascript
- JSP Notes: https://www.tutorialsduniya.com/jsp
- Microprocessor Notes: https://www.tutorialsduniya.com/microprocessor
- OR Notes: https://www.tutorialsduniya.com/operational-research

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

(1) Intensity decreases with increasing dist. from pixel to edge.

(2) Primitives don't influence pixels that they don't intersect

(3) Equal areas of pixels contribute equal intensity.

Unweighted Area Sampling
↓
considering weighing function $W(x, y)$

- height at (x, y) gives weight for area dA at (x, y) — how much amt. of contribution is given by a pixel.
- unweighted - graph of W is box i.e a cube i.e weight is constant.
- Thus, whatever part we may consider, we only see area covered & not which part is covered.

* Weighted Area Sampling
- equal areas may not contribute equally & every part also is not contributing equally.
- Change 3rd property of Unweighted Area Sampling

Tutorialsduniya.com

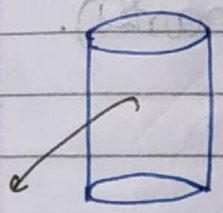
Contributor: Riya Goel

https://www.tutorialsduniya.com

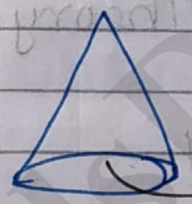
(areas closer to pixel centre contribute more)

- Need to change "geometry" of pixel to preserve 2nd property. □□ → ⬭⬭
  (pixel is circle larger than square and they are overlapping).
- If line intersects circle it contributes.
- Terms weighted & unweighted come from idea of weighing function.

• Weighing function → weighing func. is (in form of a cone).



each part contributes equally.



→ centre part contributes max. than sides

Weighing function $W(x,y)$

- choose simplest graph with height proportional to distance.
- graph of W is circular cone.
- choose radius as distance b/w pixel centres.

(Amount of area as well as which part is covered matters)

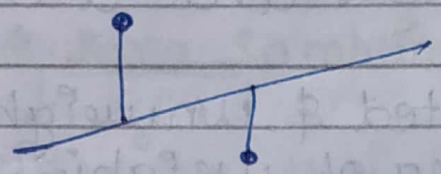
Consequences of weighting
↓
not necessary that equal area

Contributor: Riya Goel

https://www.tutorialsduniya.com

pixels are plotted with equal intensity.

→ Gupta - Sproull Antialiasing
↓
Whenever we want to plot a pixel, we plot its neighbouring pixels also but with variable intensity depending on how far the pixel is from line.



(calculating perpendicular distances)

Q (2m exam) Aliasing & antialiasing. (Theory - 5 marks).

Theory - CG

- Display (Video Display Device)

- CRT (most commonly used)
- Other popular display types.
  - Liquid Crystal Display
  - Plasma Display
  - Field Emission Displays
  - LED's
  - 3D display devices

Cathode Ray Tube
↓

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

(1) cathode rays emitted by electron gun.
(2) focussing & deflection.
(3) when electron beams contact screen phosphor emits light.
(4) light fades redraw req. in a small period (refresh)

all values of pixel before coming to CRT are in digital values.

→ $x_{coord.}$, $y_{coor.}$, intensity.

The intensity is fed as a voltage into CRT
Before feeding into CRT, digital values are converted to analogue.

→ Accelerating anode (+) and focussing anode. – focusses e^- in a single line.

→ $x_{co.}$, $y_{co.}$ → given as I/P to deflection plates. They deflect e^- beam or tilt e^- beam to particular pixel position.

(deflection plates – horizontal & vertical).

→ When e^- atom hits phosphor screen, the energy is raised & is released, emitting light.
(energy released ∝ voltage).

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

→ But this light will be available for shorter period of time.

This is persistence.

Thus pixel needs to be refreshed again & again.

→ To ↑se Speed, we have interlaced monitors. depends on frames/sec.

→ Control grid → controles power of e^- fired.

• CRT Types

(1) DVST - Direct View Storage Tubes.

no need for refresh, pictures stored as permanent change on phosphor screen.

(2) Calligraphic refresh CRT - line drawing or vector random scan, need refreshing, image is stored in form of line drawing commands.

vector scan.

(3) Raster-scan (point by point refreshing)

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

- Color displays-

(1) Beam penetration method:
special phosphors emitted diff. colors of different intensity of e^-

(2) Shadow mask displays.

- inside section of CRT is coated with red (outer layer) and green (inner layer) phosphors.

If e^- are slow they penetrate only outer layer thus giving red light.

If e^- moves fast, they give green light.

The e^- speeds can be adjusted in a way by combination of red & green. i.e yellow & orange.

quality of image is diminished.

only 4 / limited colors available.
use in vector scan.

The screen has 3 phosphor colored dots which emits colors red, green & blue.

- We use 3 e^- guns for emitting colors red, green, blue

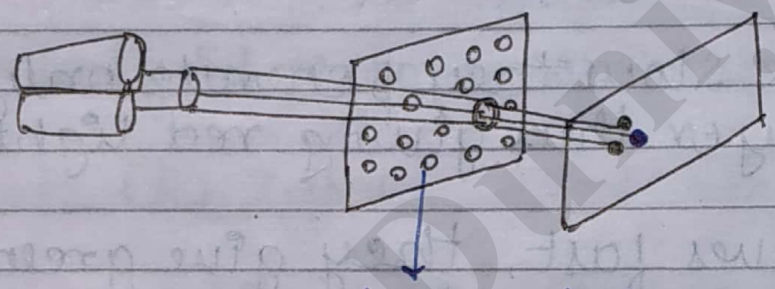
Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

- Thus voltage of e^- is such that colors appear in combination of rgb.

- For white color – all 3 beam hit with equal intensity.
- For black – all 3 beams with no intensity.
- For red – e^- gun for red is max, e^- gun for green & blue is 0.



perforated mask (shadow mask).

* LCD Displays :-

- No tube & e^- beams.

- A crystalline material is present which is known as "Nematic".

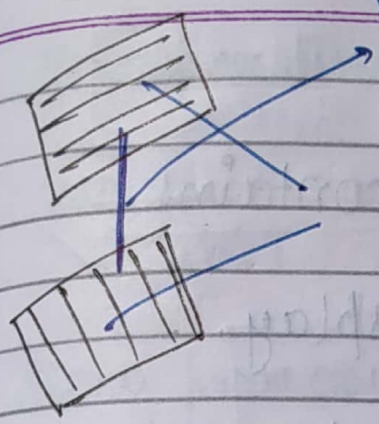
- Blocking / unblocking of light through polarized crystals.

- A matrix of LC cells one for each pixel.

- no refresh unless screen changes. Colors 3 cells per pixel.

Contributor: Riya Goel

https://www.tutorialsduniya.com



contains nematic crystalline material.

polarizers are in ⊥ direction. To pass the light, we need to twist light.

role of crystalline molecules (nematic molecules).

used to glow a pixel position.

The pixels / pts. that don't twist their direction don't get plotted.

* Plasma Panel Displays :-

- use of neon gas b/w 2 plates. (sandwiching neon gas b/w 2 plates).
- Each plate is coated with conductive print. Print on one plate contains vertical conductive lines and other has horizontal conductive lines.
- Together 2 plates form a grid.
- Gas at intersection glows, creating a point of light / pixel.
- can be seen as collection of very small neon bulbs.

Contributor: Riya Goel

https://www.tutorialsduniya.com

Q (in exam) I/P & O/P devices. / Working of CRT, LCD

CLASS TIME DATE NO.
Date / /

* Light Emitting Diodes

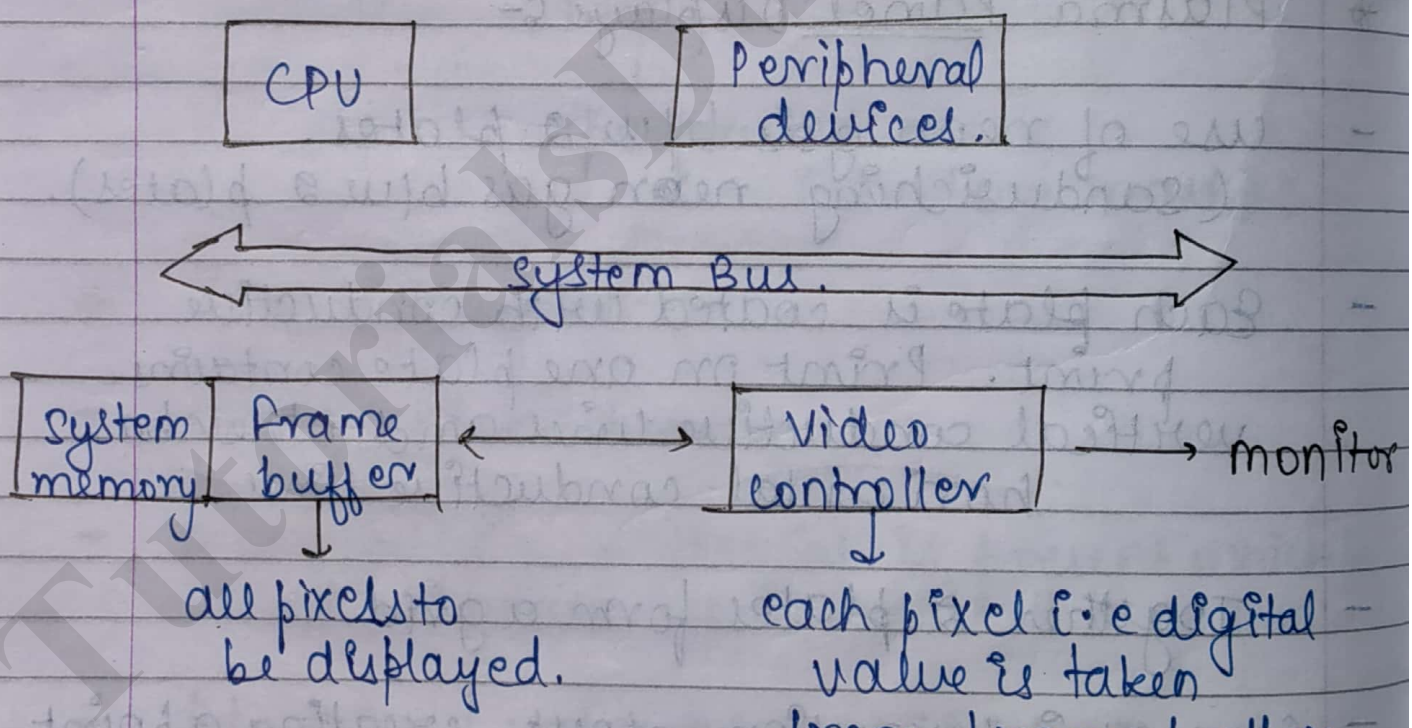
→ cross section of 2 plates contains LED's.

→ Similar str. as plasma display.

* Simple Raster Display Systems

Frame buffer : Stored pixel map of screen.

Video controller just refreshes frame buffer on monitor periodically.



Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

- Inexpensive
- We need scan conversions of O/P primitives (lines, rectangles). done by CPU.
- Slow process
- As refresh cycle ↑es, memory cycle used by video controller.

↓

* Graphics Display processor — digitizes a picture definition.

- A processor is added along with CPU which does all graphics work.
- System becomes faster.
- Purpose of display processor is to free CPU from graphics chores. A seperate display processor memory area can be provided.
- also called graphics coprocessor.

Contributor: Riya Goel

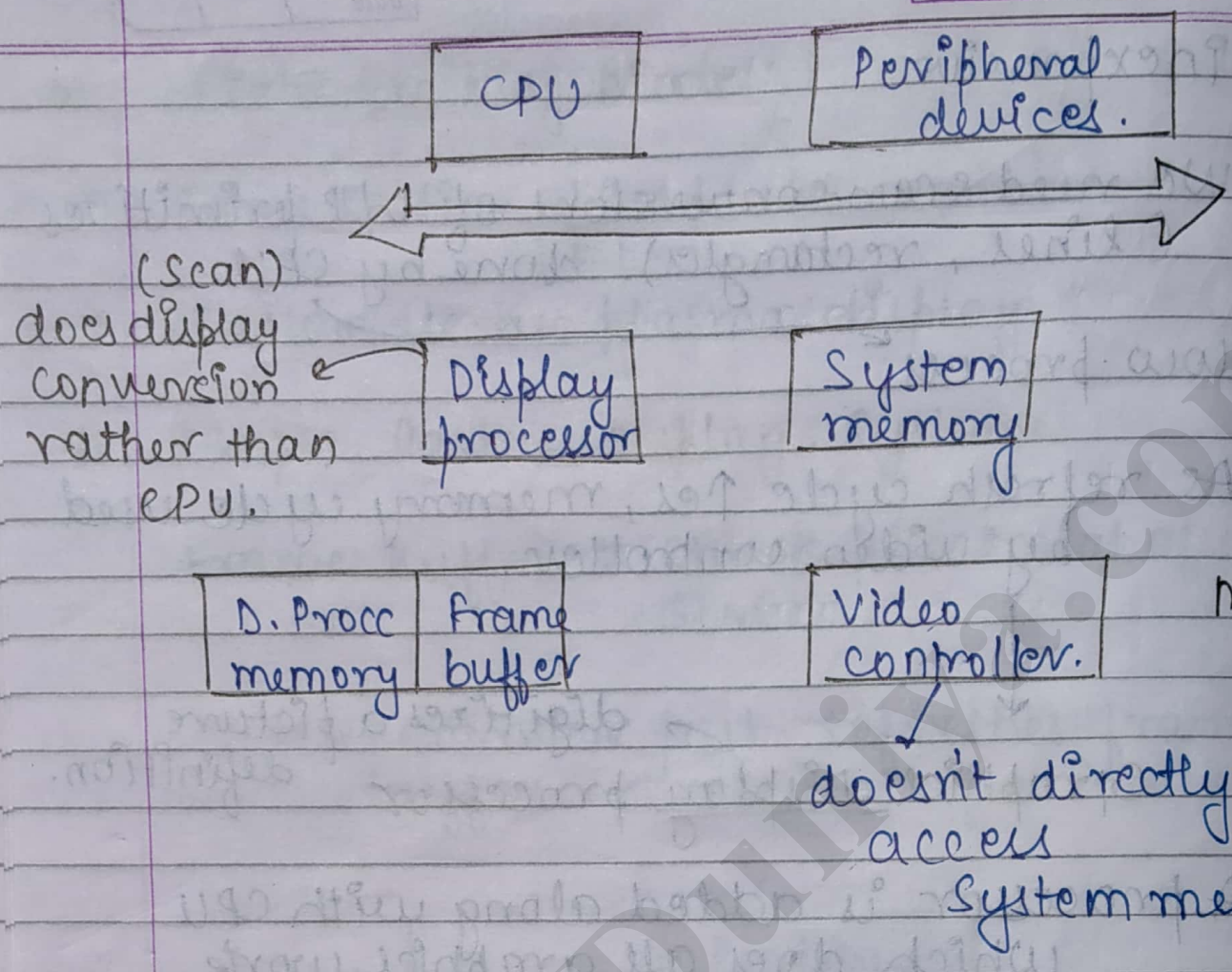
TutorialsDuniya.com

Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc of following subjects from https://www.tutorialsduniya.com

- C and C++
- Programming in Java
- Data Structures
- Computer Networks
- Android Programming
- PHP Programming
- JavaScript
- Java Server Pages
- Python
- Microprocessor
- Artificial Intelligence
- Machine Learning
- Computer System Architecture
- Discrete Structures
- Operating Systems
- Algorithms
- DataBase Management Systems
- Software Engineering
- Theory of Computation
- Operational Research
- System Programming
- Data Mining
- Computer Graphics
- Data Science

- Compiled Books: https://www.tutorialsduniya.com/compiled-books
- Programs: https://www.tutorialsduniya.com/programs
- Question Papers: https://www.tutorialsduniya.com/question-papers
- Python Notes: https://www.tutorialsduniya.com/python
- Java Notes: https://www.tutorialsduniya.com/java
- JavaScript Notes: https://www.tutorialsduniya.com/javascript
- JSP Notes: https://www.tutorialsduniya.com/jsp
- Microprocessor Notes: https://www.tutorialsduniya.com/microprocessor
- OR Notes: https://www.tutorialsduniya.com/operational-research

https://www.tutorialsduniya.com



* CG softwares.
- Line drawing.
- Piece-wise polynomial (spline) curves.
- Implicit surfaces

* Algos.

(1) Transformation → modifying a figure.
(2) Clipping
(3) Hidden surface removal → to display & identify all visible parts remove all hidden parts.
(4) Rasterization.

Included in Translation:
- Scaling – enlarging a figure
- Rotation – different orientation
- Reflection – mirror image of figures.

3-D viewing / projections.

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

- Picking - Select 3D obj. by clicking an I/P device over a pixel location.
- Shading & illumination - show image with a particular effect (lighting/shading/material)
- Animation - movement by rendering sequence of frames.

TutorialsDuniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

13 Feb, 19

Transformations

* 2-D Transformation :-
  ↓
  vertex is represented by 2 coordinates

→ Transformations are any geometric change applied to an obj. from current to next state.

→ Transformations help to modify obj. w/o redrawing once obj. is scan converted.

- 2 ways of Transformation.

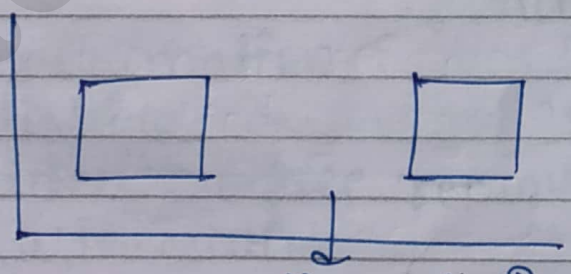
① Objective Transformation

Imp

- Alter coordinates description
- Translation, rotation, scaling
- coordinate system unchanged.

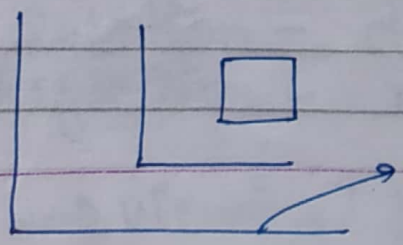
② coordinate Transformation

- Produce diff. coordinate system.



(i) Translation (coord. changed & sq. is drawn.

(ii)



coordinate system changed but sq remains same pos.

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

* Matrix Math → used for Transformations.

- more convinient organisation of data
  1 coord. → every row / col.

- more efficient processing
  no. of vertices = no. of rows.

- enables combination of various concatenations, rather than applying indivi-dual transformations.
  for series of transformation.

- There is a diff. b/w possible representations.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} e \\ f \end{bmatrix}$$ → vertex in form of col vector
then we post multiply object with Transformation matrix

$$\begin{bmatrix} e & f \end{bmatrix} \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

vertex is in form of row vector ∴ we pre multiply object with Transformation matrix.

each coordinate has 2 vertices.

For square
↓ 4 vertices.

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

$$\begin{bmatrix} a & b & c & d \end{bmatrix}$$

→ order will be 2X4 (obj. matrix)

Contributor: Riya Goel

https://www.tutorialsduniya.com

can be derived from Basic transformations { Reflection → rotation by 180°, Shearing }.

CLASSTIME Page No.
Date / /

Transformation matrix. (2X2)

$$[\ ]_{2\times2} \cdot [\ ]_{2\times4} = \underline{\underline{2\times4}}$$

$$[\ ]_{4\times2} \cdot [\ ]_{2\times2} = \underline{\underline{4\times2}}$$

NOTE:- We will use column vector representation of a point.
i.e pre multiplication of Transformation matrix / post multiplication of obj. matrix.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \underbrace{\begin{bmatrix} ax+by \\ cx+dy \end{bmatrix}}_{\text{resultant Transformed coordinate}}$$

① Translation.

- moving a pt. to a new position along same straight line.
- The path is represented by vector called translation/shift vector
- Translation factors

along x — along y

$p'x = px + t_x$ $\quad$ $p'y = py + t_y$

Tutorialsduniya.com

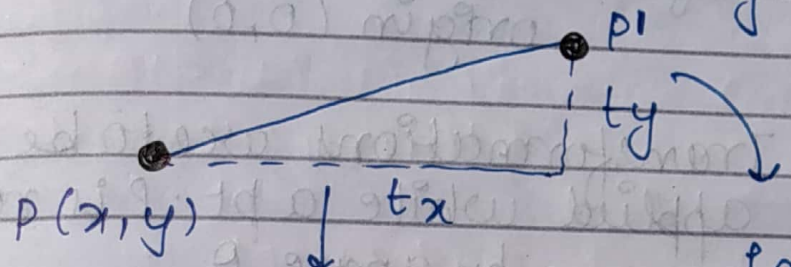
https://www.tutorialsduniya.com

Basic Transformations

Rotation Scaling Translation.

CLASSTIME Page No.
Date / /

For diagonal Translation, changes will be in both x or y.



factor

add Translation of x along Px and y along Py.

$$P'x = Px + tx$$
$$P'y = Py + ty$$

In matrix form.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} tx \\ ty \end{bmatrix}$$

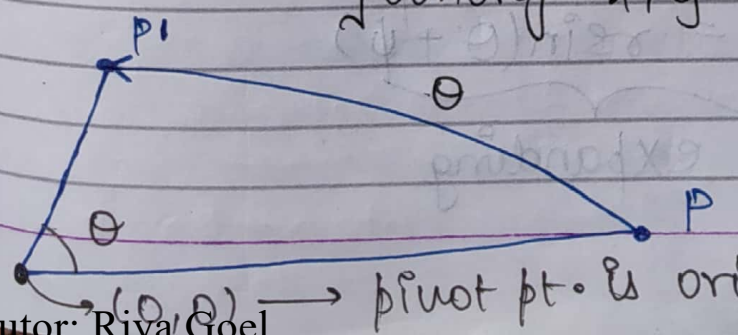
Translation factors are always added.

② Rotation

- repositions all pts. in an object along a circular path in plane centred at the pivot pt.

- rotation of a pt. means rotating its position vector.

joining x, y to origin.



→ pivot pt. is origin.

Contributor: Riya Goel

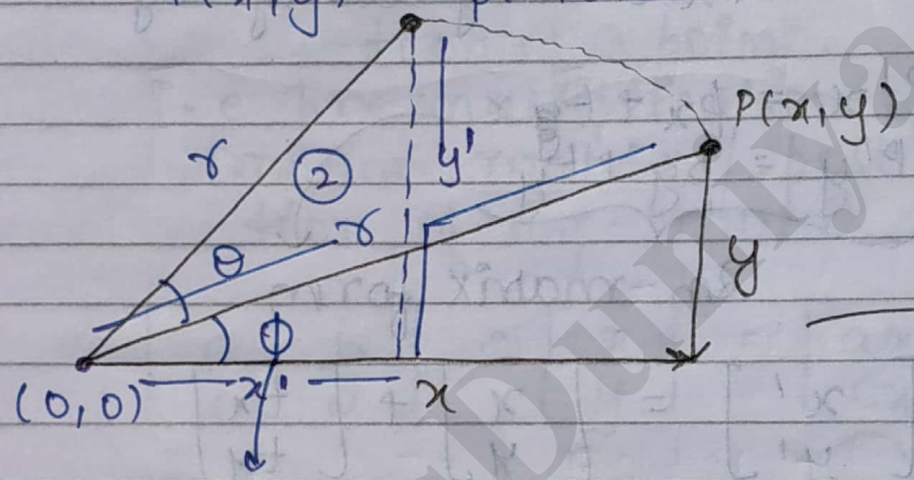
https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Rotation is always along a particular pt. If not mentioned, that pt. is origin (0,0)

* What Transformations are to be applied while a pt. P is rotated _____ by angle θ.

→ P(x,y) → pt. to be rotated. in anticlock-wise dir^n (+ve)



angle that position vector p makes with axis.

$$\left\{\begin{aligned} x &= r\cos\phi \\ y &= r\sin\phi \end{aligned}\right\}$$

We rotate P to a new pt P'(x', y') by θ.

Acc. to Δ ②

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

expanding

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

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

$$x' = x\cos\theta - y\sin\theta \quad \text{—— (1)}$$

$$\left\{ \begin{array}{l} r\sin\phi = y \\ r\cos\phi = x \end{array} \right\}$$

$\therefore$ θ is known & ϕ is unknown.

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

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

$$y' = x\sin\theta + y\cos\theta \quad \text{—— (2)}$$

Here x' and y' are represented in terms of previous (x, y) coordinates.

Transformed coordinate
$$\left\{ \begin{array}{l} p'x = px\cos\theta - py\sin\theta \\ p'y = px\sin\theta + py\cos\theta \end{array} \right.$$

$$P' = R \cdot P \quad (\text{matrix form})$$

NOTE :- θ can be clockwise (−ve) or counterclockwise (+ve).

$(\theta = -\theta)$ $(\theta = \theta)$

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

If px and py is taken as row vector.

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

Transformation matrix changes.

Transpose of matrix!

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

NOTE:- When obj. matrix pos. changes, then Change pos. of Transformation matrix

Q (in exam) numericals (10-15 marks) + theory (10 marks)

Q Find transformed pt. P' caused by rotating P = (5, 1) abt origin through 90°.

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

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

$$= \begin{bmatrix} 5\cos 90 - 1\sin 90 \\ 5\sin 90 + 1\cos 90 \end{bmatrix}$$

new Transformed pt. $= \begin{bmatrix} -1 \\ 5 \end{bmatrix}$

(3) Scaling - - Scaling changes the size of an object & involves 2 scale factors S_x and S_y for x & y coordinates.

- Scales are about origin.

- Scaling along x for a square changes into rectangle (same along y)

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

If scaling factor along x & y is same

Uniform Scaling
(Shape doesn't change)

If scaling factors > 1
size ↑es
else ↓es.

size may ↑se/
↓se depending
upon S_x & S_y

- scaling is done around a pivot pt.
(normally origin)

$$\begin{cases} p'_x = S_x \cdot p_x \\ p'_y = S_y \cdot p_y \end{cases}$$

in matrix → $P' = S \cdot P$

Scale matrix $S = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}_{2\times 2}$

diagonal values are 0

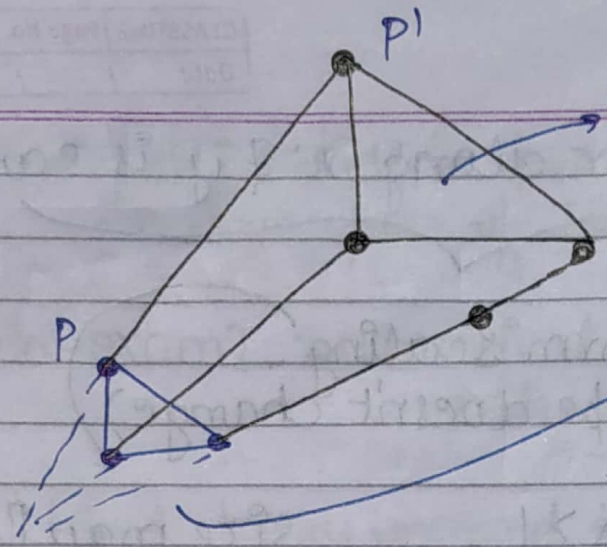
as it is 2-D Transformation

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

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

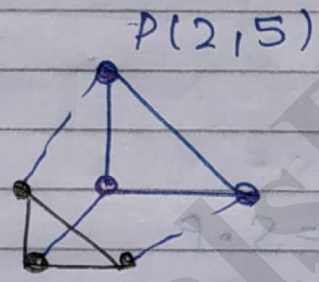


Scaling factors > 1

If scaling factors < 1

Q If scale factors are in b/w 0 and 1

pts. will be moved closer to origin i.e object becomes smaller



$S_x = 0.5$
$S_y = 0.5$

$$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \end{bmatrix} = \begin{bmatrix} \\ \end{bmatrix}$$

If scale factors > 1

pts. moved far from origin i.e object becomes larger

$S_x = S_y$ (uniform scaling)
only change in size.

$S_x \neq S_y \rightarrow$ (differential scaling)
change in size & shape.

TutorialsDuniya.com

TutorialsDuniya.com

Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc of following subjects from https://www.tutorialsduniya.com

- C and C++
- Programming in Java
- Data Structures
- Computer Networks
- Android Programming
- PHP Programming
- JavaScript
- Java Server Pages
- Python
- Microprocessor
- Artificial Intelligence
- Machine Learning
- Computer System Architecture
- Discrete Structures
- Operating Systems
- Algorithms
- DataBase Management Systems
- Software Engineering
- Theory of Computation
- Operational Research
- System Programming
- Data Mining
- Computer Graphics
- Data Science

- Compiled Books: https://www.tutorialsduniya.com/compiled-books
- Programs: https://www.tutorialsduniya.com/programs
- Question Papers: https://www.tutorialsduniya.com/question-papers
- Python Notes: https://www.tutorialsduniya.com/python
- Java Notes: https://www.tutorialsduniya.com/java
- JavaScript Notes: https://www.tutorialsduniya.com/javascript
- JSP Notes: https://www.tutorialsduniya.com/jsp
- Microprocessor Notes: https://www.tutorialsduniya.com/microprocessor
- OR Notes: https://www.tutorialsduniya.com/operational-research

https://www.tutorialsduniya.com

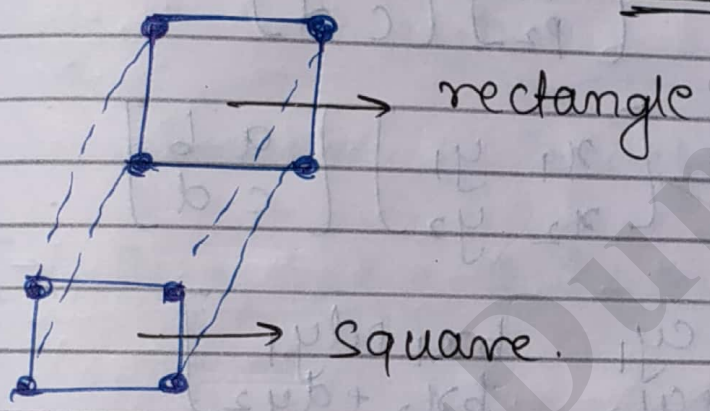
sq → rectangle

If scaling factors = 1

size remains same

If scaling factors are -ve

???



Transformation of Straight line (Proof)

→ Any generalised 2-D Transformation transforms a S.L into a S.L

$p_1 = [x_1, y_1]$, $p_2 = [x_2, y_2]$

Transforming line with matrix

$$T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$p_1^* = p_1 T = [x_1 \ y_1] \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

$$= [ax_1 + cy_1 \quad bx_1 + dy_1]$$

$$P_2^* = P_2 T = [x_2 \; y_2] \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$= [ax_2 + cy_2 \quad bx_2 + dy_2]$$

Transformation of whole object

∴ This can be written as

$$\begin{bmatrix} P_1^* \\ P_2^* \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

row wise

$$= \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$= \begin{bmatrix} ax_1 + cy_1 & bx_1 + dy_1 \\ ax_2 + cy_2 & bx_2 + dy_2 \end{bmatrix}$$

$$= \begin{bmatrix} x_1^* & y_1^* \\ x_2^* & y_2^* \end{bmatrix}$$

Transformed pts. are same as P_1^* & P_2^*

∴ SL always Transforms to a S.L

Proof of Transformation of SL

It can only be proved that every pt. on original line corresponds to a pt. in the new line

→ Take any pt. (mid pt.) on original line.

$$\frac{x_1 + x_2}{2}, \quad \frac{y_1 + y_2}{2}$$

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Transform mid pt. using TM

Find mid pt. of Transformed line & if they become equal. then it is prooved.

* mid pt. Transformation.

$$\underbrace{\frac{ax_1+cy_1+ax_2+cy_2}{2}, \quad \frac{bx_1+dy_1+bx_2+dy_2}{2}}_{\text{mid pt. of Transformed line.}}$$

$$[x_1^* \quad y^*] = \left[\frac{x_1^*+x_2^*}{2} \quad \frac{y_1^*+y_2^*}{2}\right]$$

$$= \left[\frac{(ax_1+cy_1)+(ax_2+cy_2)}{2}, \frac{(bx_1+dy_1)+(bx_2+dy_2)}{2}\right]$$

$$= \left[a\frac{(x_1+x_2)}{2}+c\frac{(y_1+y_2)}{2} \quad b\frac{(x_1+x_2)}{2}+d\frac{(y_1+y_2)}{2}\right] \quad (1)$$

Transformation of original line

$$[x_1 \quad y_1]T = \left[\frac{x_1+x_2}{2} \quad \frac{y_1+y_2}{2}\right]\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$= \left[a\frac{(x_1+x_2)}{2}+c\frac{(y_1+y_2)}{2} \quad b\frac{(x_1+x_2)}{2}+d\frac{(y_1+y_2)}{2}\right]$$

same as (1)

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

20 Feb, 19.

2-D Transformations

→ Slope of Transformed Line. (Proof-2)

Parallel lines remain ∥ upon transformation i.e slope remains same.

(x_1, y_1) and (x_2, y_2)
↳ Slope $= m = \frac{y_2 - y_1}{x_2 - x_1}$

And slope of transformed line is

$$m^* = \frac{y_2^* - y_1^*}{x_2^* - x_1^*} = \frac{(bx_2 + dy_2) - (bx_1 + dy_1)}{(ax_2 + cy_2) - (ax_1 + cy_1)}$$

Transformed coordinates

$$= \frac{b(x_2 - x_1) + d(y_2 - y_1)}{a(x_2 - x_1) + c(y_2 - y_1)}$$ {Rearranging}

$\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$

$$= \frac{b + d\frac{(y_2 - y_1)}{(x_2 - x_1)}}{a + c\frac{(y_2 - y_1)}{(x_2 - x_1)}}$$ {divide by $(x_2 - x_1)$ throughout}

Transformed Slope $= \boxed{\frac{b + dm}{a + cm}}$ —— (i)

m^* is thus independent of $\underbrace{x_1, y_1, x_2, y_2}_{\text{Transformed coordinate}}$

Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

∴ A set of parallel lines will remain parallel after transformation.

eg:- L_1 m_1 ⇓ L_2 m_2

For ∥ lines $m_1 = m_2$

$$m_1 = \frac{b + dm_1^*}{a + cm_1^*} \qquad m_2 = \frac{b + dm_2^*}{a + cm_2^*}$$

since a, b, c, d are same & $m_1 = m_2$

∴ As long as previous slope & transformation matrix is same, transformed lines will have same slope.

If we transform all coordinates of a ∥gm, parallel lines will remain parallel but shape may change.

Transformation of Intersecting lines.

Do they intersect after Transformation.

Do they intersect at same/diff. angles

Consider 2 intersecting lines

$y = m_1 x + b_1$

$y = m_2 x + b_2$

Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Writing in matrix ↓.

$$y - m_1 x = b_1$$
$$y - m_2 x = b_2$$

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 1 & -m_1 \\ 1 & -m_2 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix}$$

(col. vector form)

$$\underbrace{[x \quad y] \begin{bmatrix} -m_1 & -m_2 \\ 1 & 1 \end{bmatrix}}_{\text{gives row vector.}} = [b_1 \quad b_2]$$

Obj. matrix

Transformation matrix.

$$X m = B$$
$$X = B m^{-1}$$

$$m^{-1} = \frac{\text{Adj } m}{|m|}$$

$$|m| = -m_1 + m_2$$

Interchange main diagonal elements & reverse sign for other elements.

$$\text{Adj } m = \begin{bmatrix} +1 & +m_2 \\ -1 & -m_1 \end{bmatrix}$$

$$m^{-1} = \begin{bmatrix} \frac{1}{m_2 - m_1} & \frac{m_2}{m_2 - m_1} \\ \frac{-1}{m_2 - m_1} & \frac{-m_1}{m_2 - m_1} \end{bmatrix}$$

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

$$X = [b_1 \quad b_2]_{(1\times 2)} \begin{bmatrix} \frac{1}{m_2-m_1} & \frac{m_2}{m_2-m_1} \\ \frac{-1}{m_2-m_1} & \frac{-m_1}{m_2-m_1} \end{bmatrix}_{(2\times 2)}$$

$$[x_i \quad y_i] = \underbrace{\begin{bmatrix} \frac{b_1-b_2}{m_2-m_1} & \frac{b_1m_2-b_2m_1}{m_2-m_1} \end{bmatrix}}_{\text{pt. of intersection}}_{(1\times 2)}$$

→ Find pt. of intersection of Transformed line

⇓

After Transformation

$$\underbrace{\begin{aligned} y^* &= m_1^* x^* + b_1^* \\ y^* &= m_2^* x^* + b_2^* \end{aligned}}_{\text{Transformed set of lines}}$$

$\underbrace{m_1^* \text{ and } m_2^*}$ slopes also get Transformed

⇓

Pt. of intersection of these lines through previous method gives us.

$$\underbrace{[x_i^* \quad y_i^*]}_{\text{Transformed coordinates}} = \begin{bmatrix} \frac{b_1^*-b_2^*}{m_2^*-m_1^*} & \frac{b_1^*m_2^*-b_2^*m_1^*}{m_2^*-m_1^*} \end{bmatrix}$$

↳ If we transform them through generalised 2-D transformation matrix.

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

$$\therefore \begin{bmatrix} x^* & y^* \end{bmatrix} = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\left\{ \begin{aligned} x^* &= ax + cy \\ y^* &= bx + dy \end{aligned} \right\} \text{——— (a)}$$

→ Find out b^* in terms of b.

$$y^* = m^* x^* + b^*$$

$$b^* = y - m^* x^* \text{ ——— (1)}$$

From (a)

$$b_i^* = (bx + dy) - m^*(ax + cy)$$

$$b_i^* = bx + dy - m^* ax - m^* cy.$$

$$b_i^* = bx + dy - \left(\frac{b + dm_i}{a + cm_i}\right)(ax + cy) \quad \text{from (i)}$$

(Simplify)

$$= bx + dy - \frac{(bax + bcy + admx + dmcy)}{a + cm}$$

$$b_i^* = \underline{\underline{b_i}}(d - cm_i^*)$$

$$y - mx$$

$$\cancel{abx} + bxcm + ady + \cancel{dycm} - \cancel{bax} - bcy - admx - \cancel{dmcy}.$$

$$bxcm + ady - bcy - admx.$$

$$ad(y - mx) - bc(y - mx)$$

$$\frac{(y - mx)(ad - bc)}{(a + cm)}$$

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Replacing m_i^*

$$b_1^* = b_i \left[\frac{ad-bc}{a+cm_i}\right] \longrightarrow (3)$$

Now we get x_i^* and y_i^* as after substitution (1) & (3)

$$\begin{matrix} x_i^* \\ y_i^* \end{matrix} \left[\frac{a(b_1-b_2)+c(b_1m_2-b_2m_1)}{m_2-m_1}, \frac{b(b_1-b_2)+d(b_1m_2-b_2m_1)}{m_2-m_1}\right] \quad (4)$$

pts. of intersection of transformed line.

Put & transform original pt. of intersection with $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$

$\Downarrow$

We get same result as (4)

$\therefore$ Pts. of intersection are same after transformation.

$$\begin{bmatrix} \frac{b_1-b_2}{m_2-m_1} & \frac{b_1m_2-b_2m_1}{m_2-m_1} \end{bmatrix}_{(1\times 2)} \begin{bmatrix} a & b \\ c & d \end{bmatrix}_{(2\times 2)}$$

$$= \begin{bmatrix} \frac{a(b_1-b_2)+c(b_1m_2-b_2m_1)}{m_2-m_1} & \frac{b(b_1-b_2)+d(b_1m_2-b_2m_1)}{m_2-m_1} \end{bmatrix}_{(1\times 2)}$$

same as (4)

Contributor: Riya Goel

https://www.tutorialsduniya.com

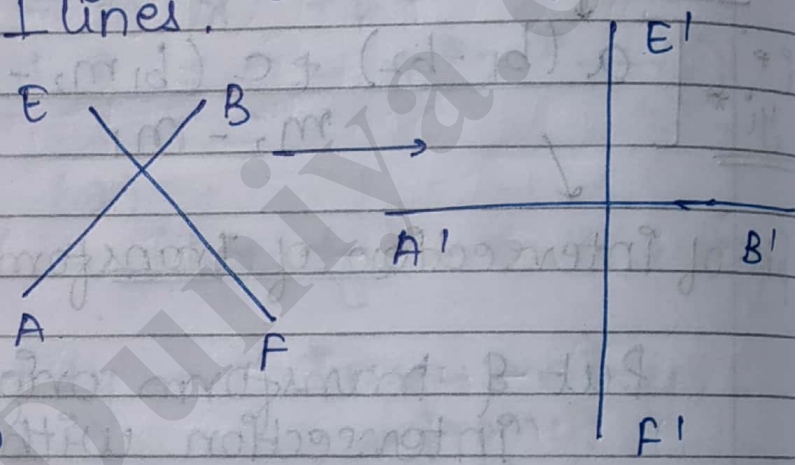
CLASSTIME Page No.
Date / /

→ Intersecting lines remain intersecting

But now we need to see whether angle intersected is same or not.

(1) A pair of non ⊥ lines get transformed to ⊥ lines.

(2) By inverse T^{-1} a pair of ⊥ lines can get transformed to non ⊥ lines.



→ Intersecting lines & rigid body Transformations

When do ⊥ lines transform as ⊥ lines?

Consider scalar & vector products of 2 edge vectors

① — $V_1 \cdot V_2 = V_{1x} V_{2x} + V_{1y} V_{2y}$. $(x_1 x_2 + y_1 y_2)$

$(x_2 - x_1)$ $(y_2 - y_1)$

2 lines l_1 & l_2

$= \underbrace{|V_1|}_{\text{magnitude of } V_1} |V_2| \cos\theta$

② — $V_1 \times V_2 = (V_{1x} V_{2y} - V_{1y} V_{2x})\,\bar{K}$

$(x_1 y_2 - y_1 x_2)$

Contributor: Riya Goel

https://www.tutorialsduniya.com

{unit vector ⊥ to plane containing V_1 & V_2}

$$= |V_1|\,|V_2|\,\bar{k}\sin\theta.$$

Let us now transform these vectors by 2x2 general transformation matrix & find dot & cross product.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} V_{1x} & V_{2x} \\ V_{1y} & V_{2y} \end{bmatrix} = \quad (1)$$

$$\begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix} = \begin{bmatrix} \underbrace{aV_{1x} + cV_{1y}}_{V_{1x}^*} & \underbrace{aV_{2x} + cV_{2y}}_{V_{2x}^*} \\ \underbrace{bV_{1x} + dV_{1y}}_{V_{1y}^*} & \underbrace{bV_{2x} + dV_{2y}}_{V_{2y}^*} \end{bmatrix}$$

$$(a^2+b^2)V_{1x}V_{2x} + (c^2+d^2)V_{1y}V_{2y} + (ac+bd)(V_{1x}V_{2y} + V_{1y}V_{2x})$$

$$= |V_1'|\,|V_2'|\cos\theta.$$

Find dot & cross prod. of Transformed Vectors

$$V_1^* V_2^* = V_{1x}^* V_{2x}^* + V_{1y}^* V_{2y}^*$$

$$V_1^* \times V_2^* = (V_{1x}^* V_{2y}^* - V_{1y}^* V_{2x}^*)\bar{k}$$

$$V_1^* V_2^* = (aV_{1x} + cV_{1y})(aV_{2x} + cV_{2y}) + (bV_{1x} + dV_{1y})(bV_{2x} + dV_{2y})$$

From (1)

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

$$a^2 V_{1x} V_{2x} + ac V_{1x} V_{2y} + ac V_{1y} V_{2x} + c^2 V_{1y} V_{2y} + b^2 V_{1x} V_{2x} + bd V_{1x} V_{2y} + db V_{1y} V_{2x} + d^2 V_{1y} V_{2y}$$

$$= (a^2+b^2) V_{1x} V_{2x} + (c^2+d^2) V_{1y} V_{2y} + (ac+bd)(V_{1x} V_{2y} + V_{2x} V_{1y})$$

(3)

$$= |V_1^*||V_2^*| \cos\theta$$

$$V_1^* \times V_2^* = (V_{1x}^* V_{2y}^* - V_{1y}^* V_{2x}^*)\hat{n}$$

$$= (aV_{1x} + cV_{1y})(bV_{2x} + dV_{2y}) - (bV_{1x} + dV_{1y})(aV_{2x} + cV_{2y})$$

$$= ab\cancel{V_{1x}V_{2x}} + ad V_{1x} V_{2y} + cb V_{1y} V_{2x} + cd\cancel{V_{1y}V_{2y}} - (ab\cancel{V_{1x}V_{2x}} + bc V_{1x} V_{2y} + da V_{1y} V_{2x} + dc\cancel{V_{1y}V_{2y}})$$

$$= V_{1x} V_{2y} (ad-bc) + (cb-da) V_{1y} V_{2x}$$

$$= (V_{1x} V_{2y} - V_{2x} V_{1y})(ad-bc)\hat{n}$$

(4)

$$= |V_1^*||V_2^*| \sin\theta \hat{n}$$

Tutorialsduniya.com

Contributor: Riya Goel

TutorialsDuniya.com

Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc of following subjects from https://www.tutorialsduniya.com

- C and C++
- Programming in Java
- Data Structures
- Computer Networks
- Android Programming
- PHP Programming
- JavaScript
- Java Server Pages
- Python
- Microprocessor
- Artificial Intelligence
- Machine Learning
- Computer System Architecture
- Discrete Structures
- Operating Systems
- Algorithms
- DataBase Management Systems
- Software Engineering
- Theory of Computation
- Operational Research
- System Programming
- Data Mining
- Computer Graphics
- Data Science

- Compiled Books: https://www.tutorialsduniya.com/compiled-books
- Programs: https://www.tutorialsduniya.com/programs
- Question Papers: https://www.tutorialsduniya.com/question-papers
- Python Notes: https://www.tutorialsduniya.com/python
- Java Notes: https://www.tutorialsduniya.com/java
- JavaScript Notes: https://www.tutorialsduniya.com/javascript
- JSP Notes: https://www.tutorialsduniya.com/jsp
- Microprocessor Notes: https://www.tutorialsduniya.com/microprocessor
- OR Notes: https://www.tutorialsduniya.com/operational-research

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

- If we want ⊥ lines to remain ⊥, then after & before transformation vector & scalar prod. would remain same

∴ (1) must be equal to (3) and (2) must be equal to (4)

⇓

$a^2+b^2=1$
$c^2+d^2=1$ {equating (1) & (3)}.
$ac+bd=0$

from (2) and (4)

$ad-bc=1$

For a Transformation matrix $\begin{bmatrix} a & c \\ b & d \end{bmatrix}$ then, acc. to this, this matrix must hold all of these above properties so as to make orientation same.

The matrix that holds all these 4 properties is an "orthogonal matrix"

Transformation with this matrix will have ⊥ lines as ⊥ only.

Putting a, b, c, d as rotational matrix.

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

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

A pure rotational matrix is an orthogonal matrix.

$$ac + bd = -\sin\theta\cos\theta + \sin\theta\cos\theta = 0$$

$$a^2 + b^2 = \sin^2\theta + \cos^2\theta = 1$$

$$c^2 + d^2 = \sin^2\theta + \cos^2\theta = 1$$

Thus, pure rotation makes angle same.

NOTE:- Transformations where angle remains preserved is

solid body / rigid body Transformation matrix.

as after pure rotation, shape doesn't change.

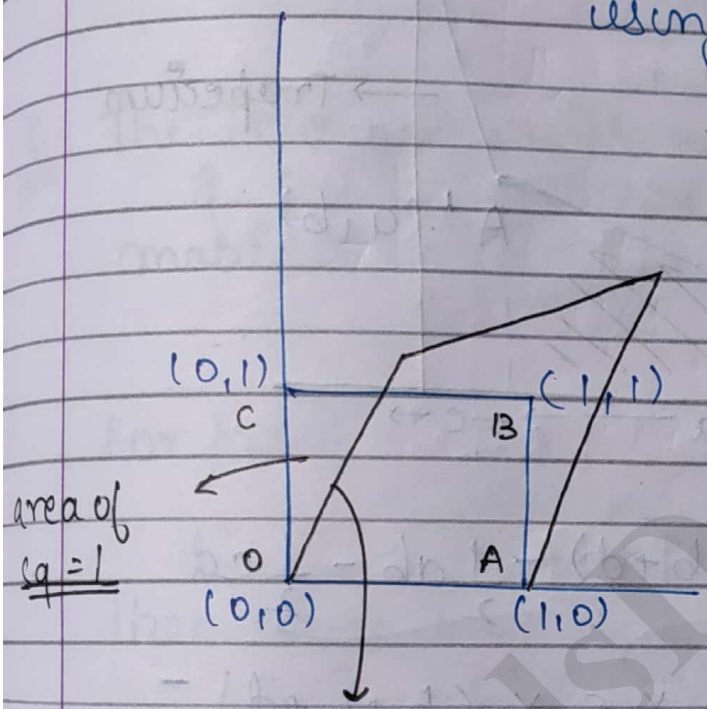
Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

What happens to area when fig. is transformed using 2-D Transformation matrix.

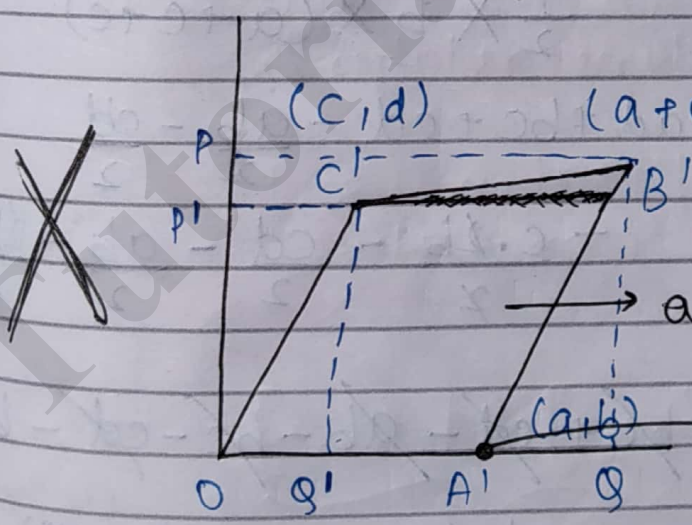
Consider a unit square. Transforming it using generalised 2-D matrix.



$$\begin{array}{c} O \\ A \\ B \\ C \end{array}\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}_{(4\times 2)} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix}_{(2\times 2)}$$

$$\begin{bmatrix} 0 & 0 \\ a & b \\ a+c & b+d \\ c & d \end{bmatrix}_{(4\times 2)} \begin{array}{l} \to O \\ \to A' \\ \to B' \\ \to C' \end{array}$$

after Transformation



area is ??

This is not (a,b).

Area of $OA'B'C'$ = area of $O'PB'Q$ − Area ($OP'C'$) − area ($PB'C'P'$) − area ($A'B'Q$)

Tutorialsduniya.com

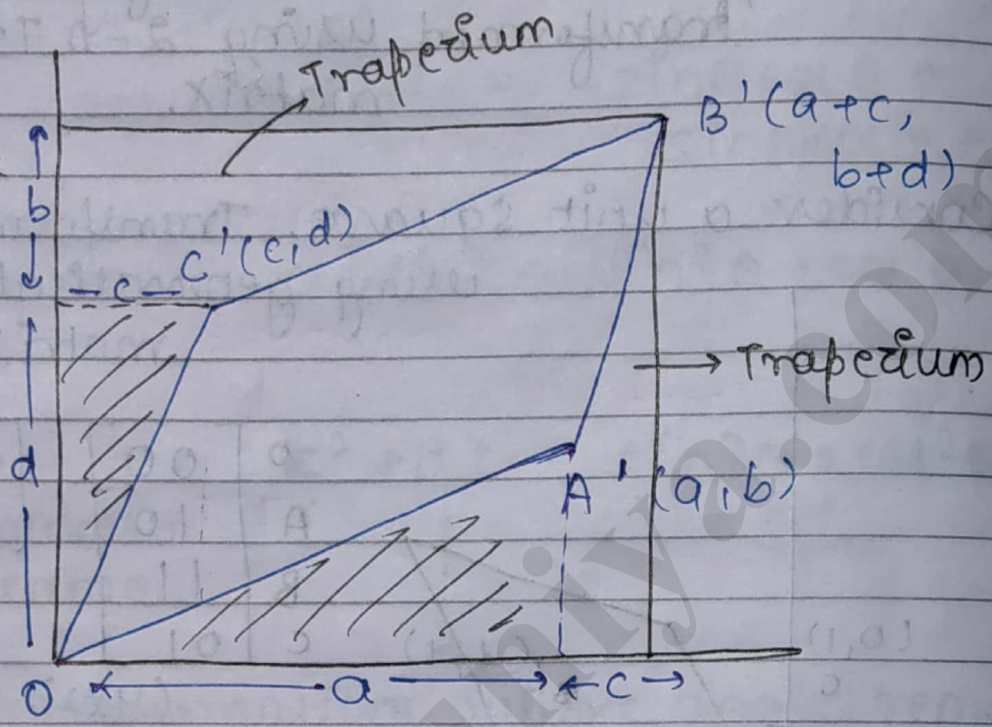
Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

$$ar(O'PB'Q) = (a+c)(b+d)$$

Values of a, b, c, d are diff. Thus figure can't be predicted



$$\text{Area} = (a+c)(b+d) - \frac{1}{2}ab - \frac{1}{2}cd - \frac{1}{2} \times c \times (b+b+d) - \frac{1}{2} \times b \times (a+c+c)$$

$$= ab + ad + bc + cd - \frac{ab}{2} - \frac{cd}{2} - \frac{c \cdot 2b}{2} - \frac{cd}{2} - \frac{ab}{2} - \frac{2bc}{2}$$

$$= \cancel{ab} + ad + \cancel{bc} + \cancel{cd} - \cancel{ab} - \cancel{bc} - \cancel{cd} - bc$$

$$= \underbrace{ad - b \cdot c}_{} = \det(T)$$

acc. to Transformation matrix, this is Δdet of matrix

Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

→ Area of Transformed fig → $|X \det(T)$

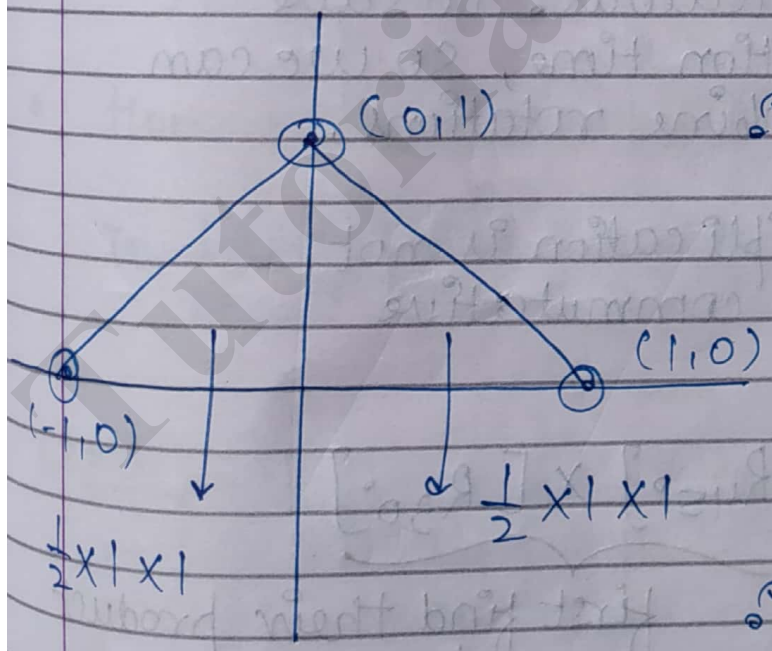
→ Area of any figure is ↓

$$\boxed{\text{area of original fig} \times \det(T)}$$

Q The Triangle with vertices (1,0), (0,1) & (-1,0) is transformed using a matrix $\begin{bmatrix} 3 & 2 \\ -1 & 2 \end{bmatrix}$. Find area of Transformed fig.

For proof
Find area of original matrix.

Then transform coordinates, and again find its area. Both will be Same.



∴ area of Δ
$= \frac{1}{2} + \frac{1}{2}$
$= 1$

$|\det| = 6 + 2$
$= 8$

∴ area of Transformed fig $= 8$.

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

NOTE:- In case of Translation, we perform addition b/w matrices

NOTE:- For rotation & scaling, we perform multiplication b/w matrices.

eg - For 2 rotations $R_{45°}$ $R_{30°}$

$$[O_1] \cdot [R_{45°}]$$

$$\Downarrow$$

$$[O_2] [R_{30°}]$$

$$= [O_3]$$

This is tedious. To save computation time, so we can combine rotations.

Matrix multiplication is not commutative

$$\Downarrow$$

$$\underbrace{[R_{45°}] \times [R_{30°}]}_{\text{first find their product}}$$

easy.

$$\downarrow$$

$$R_1$$

$$[O_1] \times [R_1] = [O_2]$$

Tutorialsduniya.com

https://www.tutorialsduniya.com

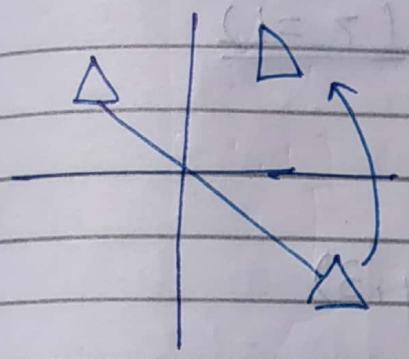
CLASSTIME Page No.
Date / /

- To perform all 3 rotation, scaling & translation, how can be form a combined Transformation matrix.

↓

we "homogenize" coordinates

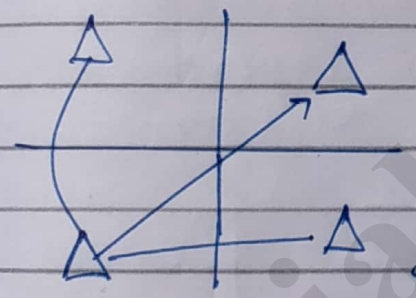
I



Translation (1)

(2) Anticlockwise rotation.

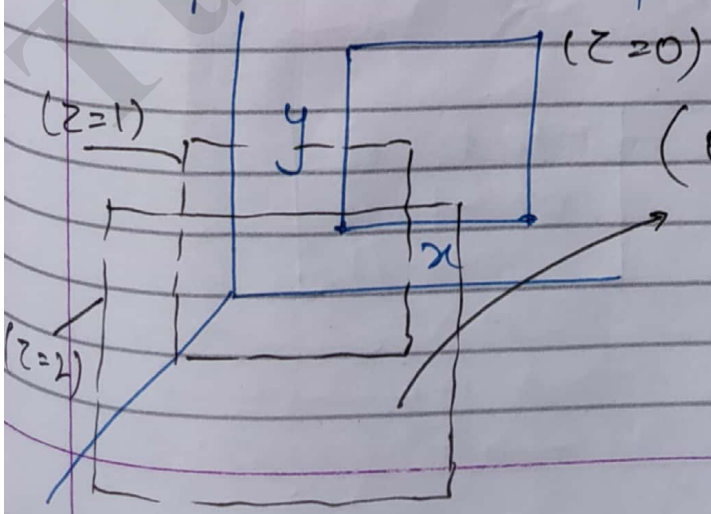
II



position can differ (first rotation then Translation).

- Homogeneous coordinates (to form composite Transformation matrix)

To homogenize 2-D coordinates, move 2-D plane in 3-D space.



(Only 1 coordinate is fixed
pt. shifts in space with a diff. value)

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

The pts. when combined lie in straight line as they are multiplied by scale of z.

$(X, Y) \longrightarrow (x, y, 1)$

3rd dimension
$(z = 1)$

At $z = 2$

$(2x, 2y, 2)$

At $z = 3$ $\longrightarrow$ $(3x, 3y, 3)$

TutorialsDuniya.com

Contributor: Riya Goel

TutorialsDuniya.com

Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc of following subjects from https://www.tutorialsduniya.com

- C and C++
- Programming in Java
- Data Structures
- Computer Networks
- Android Programming
- PHP Programming
- JavaScript
- Java Server Pages
- Python
- Microprocessor
- Artificial Intelligence
- Machine Learning
- Computer System Architecture
- Discrete Structures
- Operating Systems
- Algorithms
- DataBase Management Systems
- Software Engineering
- Theory of Computation
- Operational Research
- System Programming
- Data Mining
- Computer Graphics
- Data Science

- Compiled Books: https://www.tutorialsduniya.com/compiled-books
- Programs: https://www.tutorialsduniya.com/programs
- Question Papers: https://www.tutorialsduniya.com/question-papers
- Python Notes: https://www.tutorialsduniya.com/python
- Java Notes: https://www.tutorialsduniya.com/java
- JavaScript Notes: https://www.tutorialsduniya.com/javascript
- JSP Notes: https://www.tutorialsduniya.com/jsp
- Microprocessor Notes: https://www.tutorialsduniya.com/microprocessor
- OR Notes: https://www.tutorialsduniya.com/operational-research

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

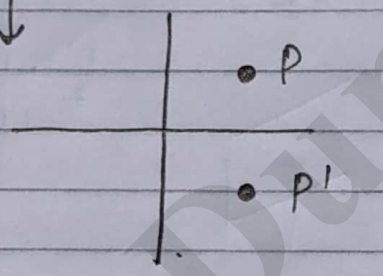
11 March, 19

Reflection

→ Order of operations matter, resultant position of objects changes.

Reflection can be achieved by rotation also (180°)
↓
Axis of Reflection.

- x axis ↓



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

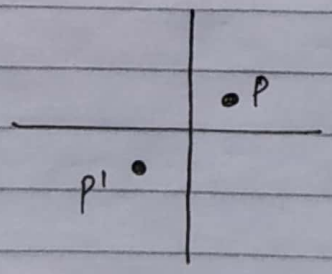
y coordinate becomes -ve.

- y axis. ↓

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

x coordinate becomes -ve.

* Reflection abt. origin



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

x & y → -ve

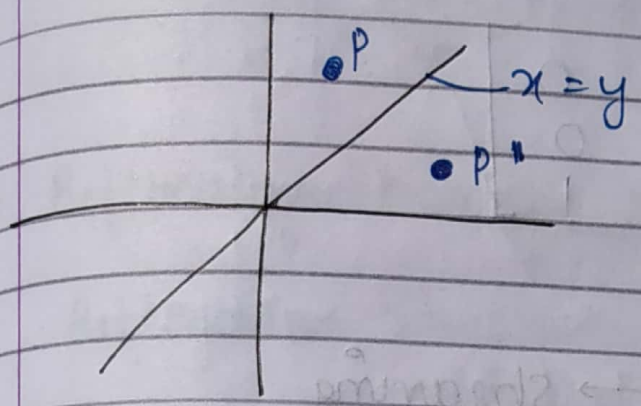
Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Reflection about a line



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

x changes to y & y changes to x

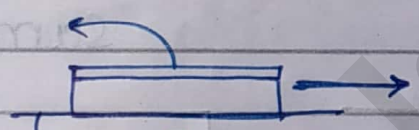
magnitude remains same.

* Shear Transformation

achieved using other Transformations.

- apply a force in x or y direction.

ydist = max. (most movement)



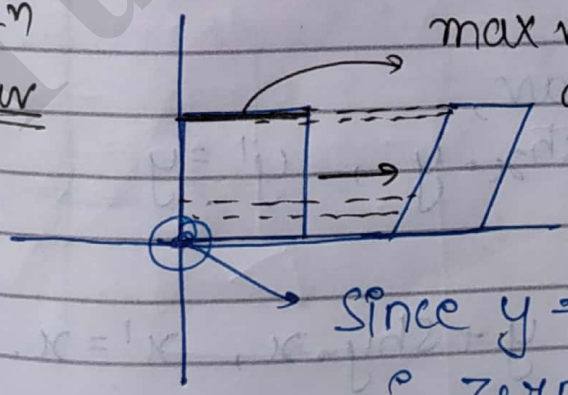
y dist = 0

Force applied ∝ dist of y.

Force ∝ x dist.

max movement (as max x)

x dir^n Shear



max movement at this layer

Since y = 0

∴ zero movement but all pts. will have equal movement

Contributor: Riya Goel

https://www.tutorialsduniya.com

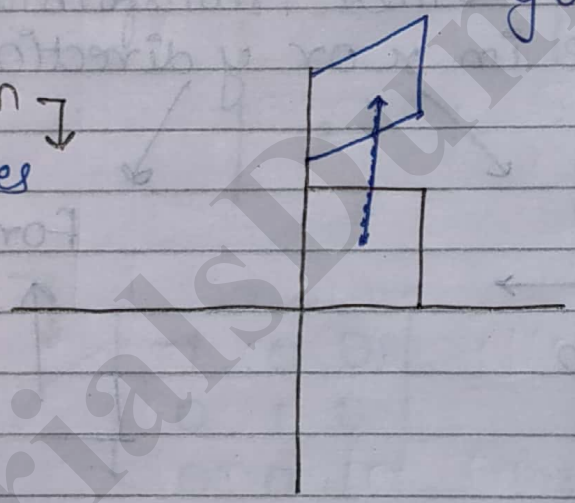
- x dir^n] (y remains same)

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

$x \rightarrow x + y\,sh_x$ → shearing factor (force applied in x dir^n)

used to apply force & pts. that are sheared depends on y dir^n.

- y dir^n] (y coordinates change depending upon x) (x coordinate remains same).



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

$y \rightarrow y + x\,sh_y$.

NOTE:- For x direction shear,
$x' = x + sh_x \cdot y$, $y' = y$

NOTE:- For y dir^n shear,
$y' = y + sh_y \cdot x$, $x' = x$.

For both dir^n shear →

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

$$\begin{bmatrix} 1 & Sh_x & 0 \\ Sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{matrix} \rightarrow x \text{ factor} \\ \rightarrow y \text{ factor.} \end{matrix}$$

- Reflection Through an Arbitrary Line :-

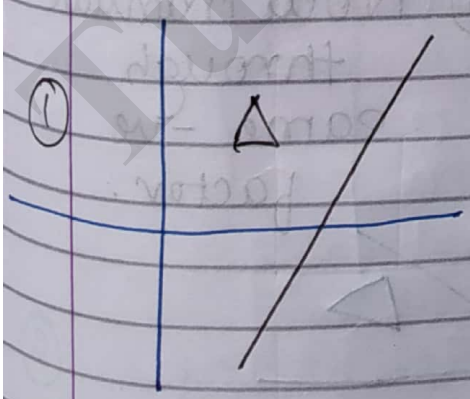
Reflection about y axis & x axis

$$T_y = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad T_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

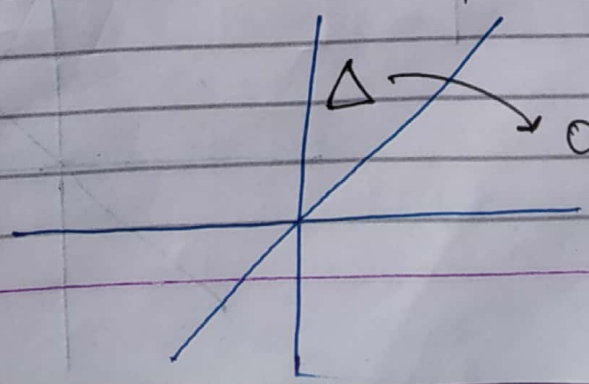
Bring line to an axis for which we have direct reflection matrixes.

To coincide line with axis, we apply Transformations ∴ object will also change.

After that apply reflections & inverse reflections to make line at original pos.

① 

→ ② Translate line s.t it passes through origin.



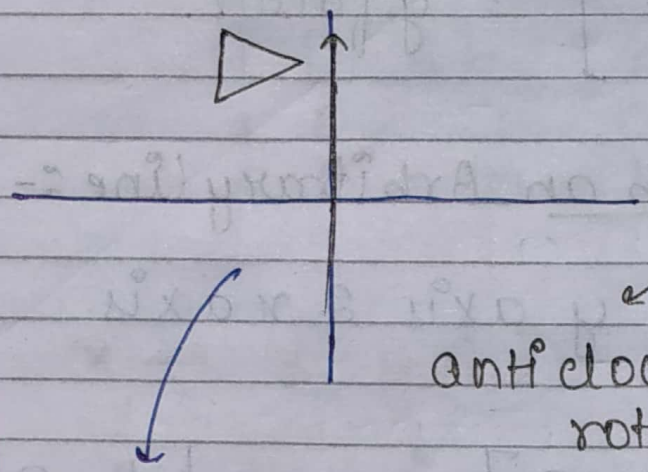
→ obj. will also Translate

Contributor: Riya Goel

https://www.tutorialsduniya.com

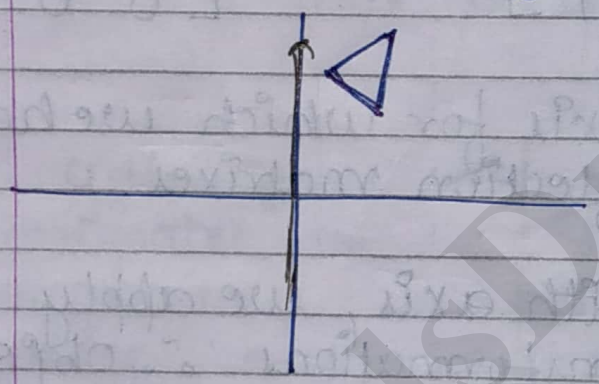
CLASSTIME Page No.
Date / /

③. Coincide line to one of the axis
rotate the line s.t it coincides to y axis.
(angle of rotation depends on eqⁿ of line).

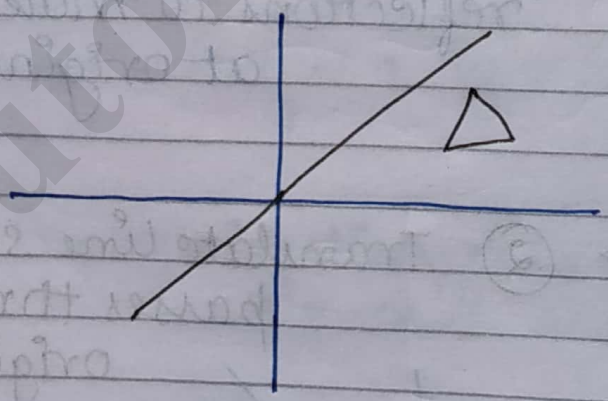


anticlockwise rotation.

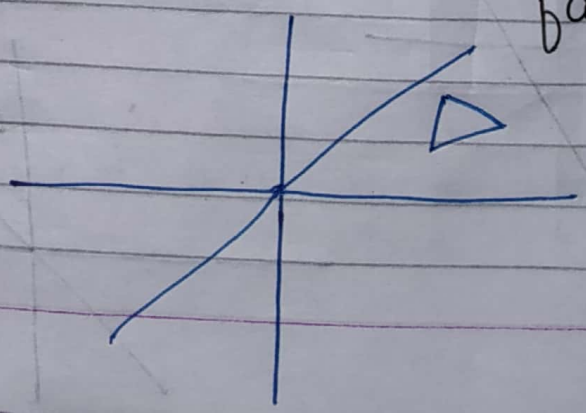
④ Reflection abt y axis.



⑤ Now reverse the reflections. (inverse transformations)
↓
First rotate (clockwise dirⁿ)



⑥ Now translate through same -ve factor.



Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Note:- If a line is ∥ to one of the axis, just do Translation, no need of rotation.

For $y = 2$
Translate to $y = -2$, so as to bring it to origin.

eg:- A Δ with vertices (2, 4, 1), (4, 6, 1) and (2, 6, 1)

line → $y = (x+4)/2$.

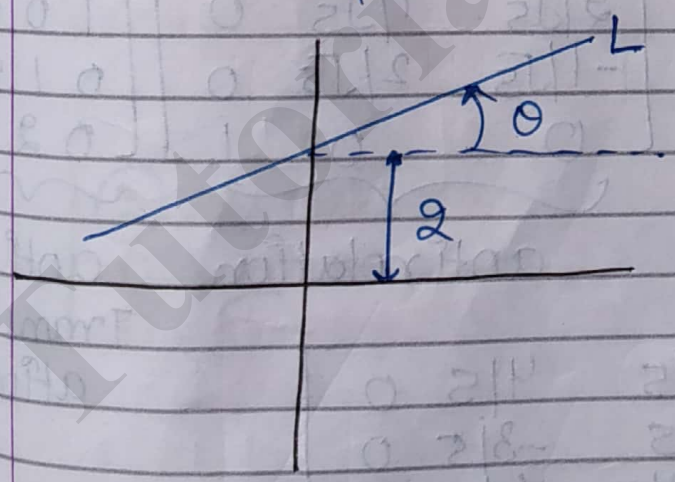
$y = \frac{x}{2} + 2$

$y = mx + c$

$m = 1/2$

$\frac{1}{2} = m = \tan\theta$ → angle with x axis.

Intercept = 2



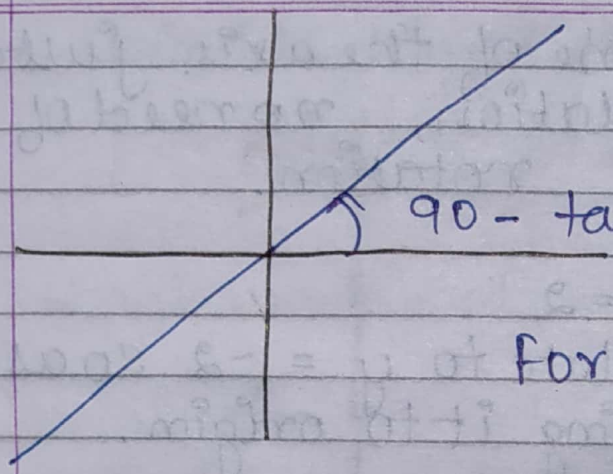
Slope $= \tan^{-1}(1/2)$
$= 26.57°$

① Translate line to pass through origin
$(y = -2,\ x = 0)$

② L passes through origin.

Contributor: Riya Goel

https://www.tutorialsduniya.com



$90 - \tan^{-1}(1/2) = \theta'$

For coinciding with x axis

$-(90 - 26.57°)$

For coinciding with y axis ↓

$(90 - 26.57°)$

$$\therefore \underline{\underline{T}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}}_{\text{Translation}} \underbrace{\begin{bmatrix} 2/\sqrt{5} & -1/\sqrt{5} & 0 \\ 1/\sqrt{5} & +2/\sqrt{5} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{rotation}} *$$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{reflection}} \underbrace{\begin{bmatrix} 2/\sqrt{5} & 1/\sqrt{5} & 0 \\ -1/\sqrt{5} & 2/\sqrt{5} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{antirotation}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}}_{\text{anti Transl-ation.}}$$

$$= \begin{bmatrix} 3/5 & 4/5 & 0 \\ 4/5 & -3/5 & 0 \\ -8/5 & 16/5 & 1 \end{bmatrix}$$

Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

∴ pre multiply obj.

$$\begin{bmatrix} 2 & 4 & 1 \\ 4 & 6 & 1 \\ 2 & 6 & 1 \end{bmatrix} T = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} *$$ composite matrix.

* Overall Scaling - (can be achieved through scaling matrix).

→ scaling an obj. uniformly in x & y dirⁿ.

→ If $S_x > S_y$ → no uniform scaling.

→ changing coordinates uniformly in both dirⁿ.

→ $[x^* \ y^* \ h] = [x \ y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & S \end{bmatrix}$

$= [x \ y \ S]$.

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

S_x, S_y

} size of obj. doubles. This can be achieved by changing z coordinate to a diff. value to make it 3-D.

Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

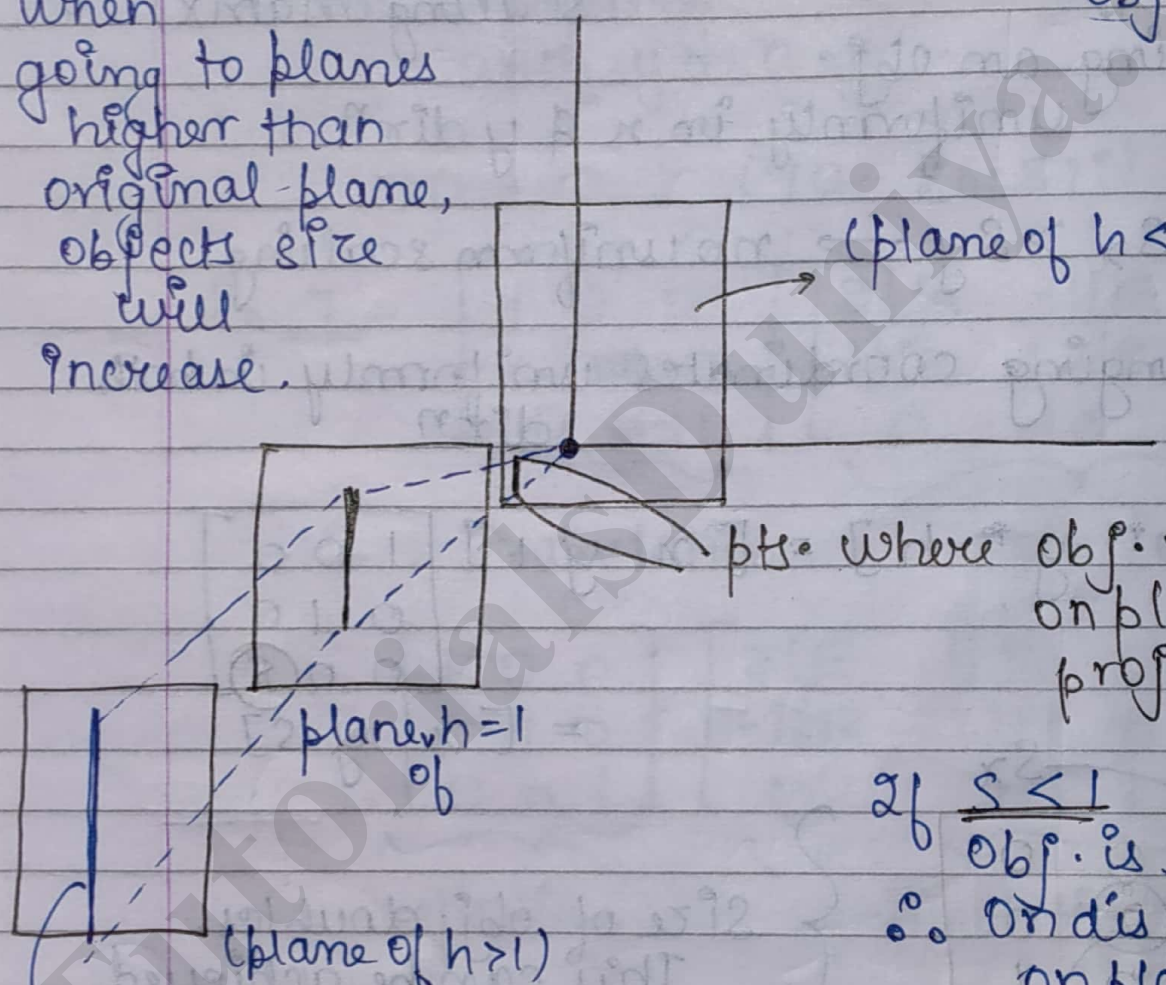
CLASSTIME Page No.
Date / /

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{bmatrix}$$

changing this value is used for overall scaling.

Changing plane from 2-D to 3-D will affect size of object.

When going to planes higher than original plane, objects size will increase.



If $s < 1$
obj. is small
∴ on displaying on plane h=1 size of obj. increases.

If $s > 1$
ultimate size of obj. reduces.
(as we need to represent on plane =1)
∴ we are moving backwards.

Contributor: Riya Goel

https://www.tutorialsduniya.com

NOTE:- S is a kind of reverse scaling factor

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

If $S = 2$ (>1)
$S_x = \frac{1}{2} = 0.5$ ($\therefore$ value reduces).

If $S = \frac{1}{2}$
$S_x = 2$ ($\therefore$ value increases).

* Classification of Transformations

rigid	Affine.
- (shape of obj. doesn't change).	- parallelism preserving.
- distance preserving	- shape may change but ∥ lines remains parallel
- eg - Translation & rotation or uniform scaling.	- eg - Translation, rotation, scaling, shearing.
- obj. is not distorted.	- preserves parallelism & not lengths & angles.
- intersecting lines remains intersecting.	

TutorialsDuniya.com

TutorialsDuniya.com

Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc of following subjects from https://www.tutorialsduniya.com

- C and C++
- Programming in Java
- Data Structures
- Computer Networks
- Android Programming
- PHP Programming
- JavaScript
- Java Server Pages
- Python
- Microprocessor
- Artificial Intelligence
- Machine Learning
- Computer System Architecture
- Discrete Structures
- Operating Systems
- Algorithms
- DataBase Management Systems
- Software Engineering
- Theory of Computation
- Operational Research
- System Programming
- Data Mining
- Computer Graphics
- Data Science

- Compiled Books: https://www.tutorialsduniya.com/compiled-books
- Programs: https://www.tutorialsduniya.com/programs
- Question Papers: https://www.tutorialsduniya.com/question-papers
- Python Notes: https://www.tutorialsduniya.com/python
- Java Notes: https://www.tutorialsduniya.com/java
- JavaScript Notes: https://www.tutorialsduniya.com/javascript
- JSP Notes: https://www.tutorialsduniya.com/jsp
- Microprocessor Notes: https://www.tutorialsduniya.com/microprocessor
- OR Notes: https://www.tutorialsduniya.com/operational-research

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

3-D Transformations

3rd dimension added.
(Right Handed coordinate system)

① Translation in 3D is a simple extension from 2D

$$T = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

If obj. is post multiplied.

for pre multiplication of obj. matrix

$$\begin{bmatrix} & & & \\ & & & \\ & & & \\ dx & dy & dz & 1 \end{bmatrix}$$

② Scaling in 3D

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

(3x3 changes to 4x4)

③ 2-D rotation is just a 3D rotation abt. z axis.

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

2-D rotation matrix

TutorialsDuniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Rotation is always along origin i.e plane ⊥ to x & y i.e z axis

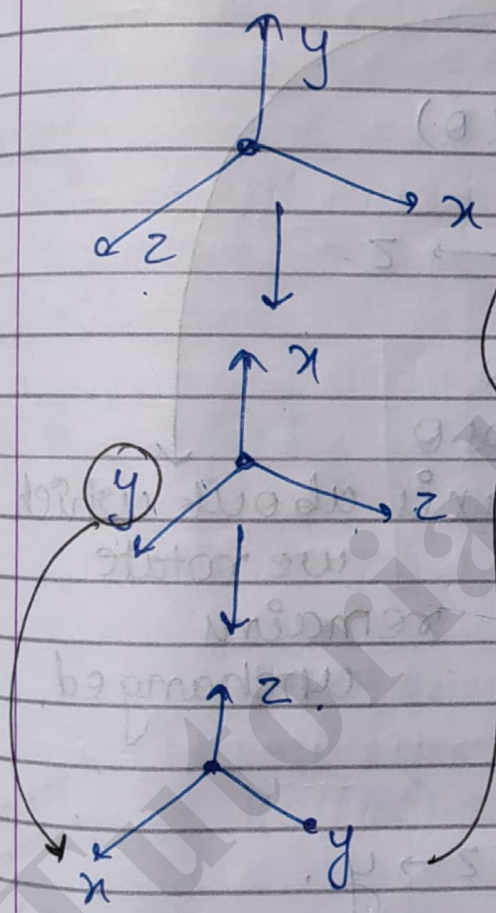
3D rotation
(rotation abt. x, y & z axis).

→ for rotation abt y axis. (rotation whole 3-D plane 2 times) =

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

↓

Put y = z
x = z
and x = y



Changing coordinates in anti-clockwise order.

→ Now rotation abt x axis.

⇓

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

Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Imp: Rotation about z

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

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

Rotation about y $R_y(\theta)$

$z \rightarrow y \rightarrow x \rightarrow z$

$z' = z\cos\theta - x\sin\theta$
$x' = z\sin\theta + x\cos\theta$
$y' = y.$ } the axis about which we rotate remains unchanged.

Rotation about x axis

$y \rightarrow x \rightarrow z \rightarrow y.$

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

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \text{Rotation abt. } x$$

Theorem- In 3D, order of sequence of rotations matters

$$R_x \rightarrow R_y \neq R_y \rightarrow R_x$$

as matrix multiplication is non commutative.

Proof - $R_z(\alpha) R_y(\beta) = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} = [A]$

$R_y(\beta) R_z(\alpha) = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} = [B]$

We see that $[A] \neq [B]$.

Theorem:- $R_z(\alpha)$ and then $R_z(\beta)$

$$\Downarrow$$

(Proof can come) $R_z(\alpha+\beta)$

Reflection in 3-D

Reflection through x & y → Reflection through z axis.

Contributor: Riya Goel

https://www.tutorialsduniya.com

$\underbrace{x \text{ \& } y \text{ coordinate}}_{\text{unchanged}}$

z coordinate → becomes −ve.

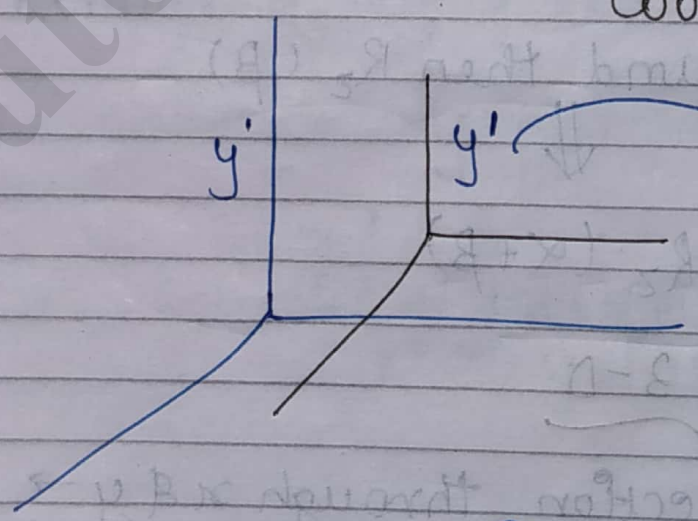
Reflection through xy plane =
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Reflection through xz plane. → y coordinate −ve
x & z unchanged.

Ref. through yz plane. → x coordinate −ve
y & z unchanged.

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

* Rotation abt an axis parallel to coordinate axis.



① Translate y' so that it's coincides with y axis.

② Rotate

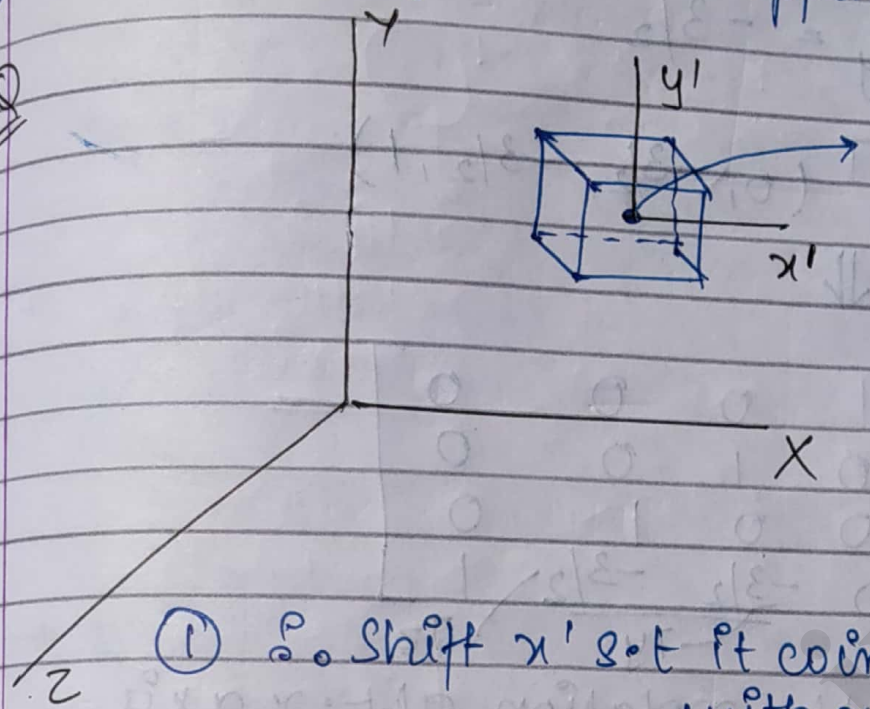
Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

③ anti Transformations applied.

Q



To find centroid of cube $\frac{x+y+z}{3}$

Rotate block by $\theta = 30°$ abt. local x' axis.

① i.e. Shift x' s.t it coincides with x axis.

②

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array} \begin{array}{c} \begin{array}{cccc} x & y & z & 1 \end{array} \\ \begin{bmatrix} 1 & 1 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 2 & 1 & 1 \end{bmatrix} \end{array}$$

vertices of cube.

centroid of cube

$$\frac{12}{8}, \frac{12}{8}, \frac{12}{8}, \frac{8}{8}$$

$$= \left[\frac{3}{2}, \frac{3}{2}, \frac{3}{2}, 1\right]$$

③ One method is to shift cube to origin

$$\left[-3/2, -3/2, -3/2, 1\right]$$

Or we can translate with y & z axis, s.t x' coincides with x.

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

$\therefore T_x = 0 \qquad T_z = -3/2$
$T_y = -3/2$

$\therefore T(0, -3/2, -3/2, 1)$

$\Downarrow$

$$\begin{array}{c} \\ T_x \end{array}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -3/2 & -3/2 & 1 \end{bmatrix}$$

T_y T_z

(4) Now, apply rotation abt. x axis by angle of 30°.

$$\therefore \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -3/2 & -3/2 & 1 & \end{bmatrix}}_{\text{Translation}} * \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin 30^\circ & \cos 30^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{Rotation}}$$

$$* \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 3/2 & 3/2 & 1 \end{bmatrix}}_{\text{Inverse Translation}}$$

in row vector format obj. is premultiplied.

$\therefore [O] = TRT^{-1}$

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

{ 2-D → arbitrary axis
3-D → arbitrary axis in space }

In 3D, first object must pass through origin. But that line is in air so, we first want that line to be in a plane ∴ one additional step is rotation so that it comes in one of the planes.

additional step in 3-D

→ If we coincide a line at x axis, we perform double rotation on axis other to coinciding axis. (i.e y & z)

→ But how much rotation is needed?

* Rotation about an Arbitrary axis

(1) Make axis P1P2 coincide with z axis
(2) Translation to move P1 to origin :-
$T(-x_1, -y_1, -z_1)$
(3) Coincides one pt. of axis with origin.
(4) Rotation to coincide the shifted axis with z axis.
(5) R1 : rotation around x s.t axis lies on xz plane.
(6) R2 : rotation around y s.t axis coincides with z axis.
(7) R3 : rotate scene around z axis by an angle θ.
(8) Inverse transformations of R2, R1 & T1 to bring back axis to

Contributor: Riya Goel

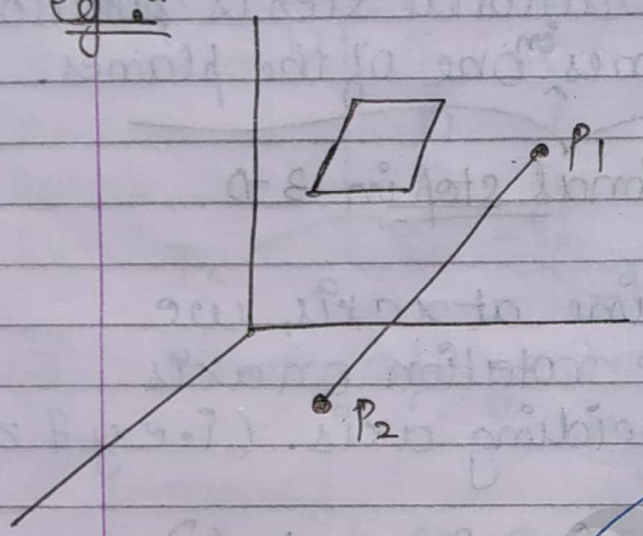
https://www.tutorialsduniya.com

original position.

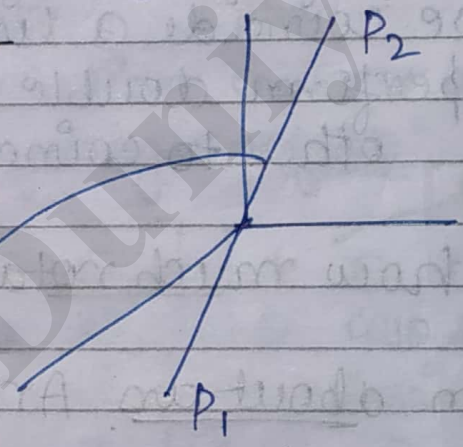
$$M = T^{-1} R_1^{-1} R_2^{-1} R_3 R_2 R_1 T.$$

actual rotation abt. z axis

eg:-



① Translate line



line vector coordinates

$P_2 - P_1$

$(x_2 - x_1,\ y_2 - y_1,\ z_2 - z_1)$

Now $\hat{u} = \frac{V}{|V|} = (a, b, c)$

unit vector along dir^n of P_1P_2.

$$\frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2}}$$

② Rotate u abt. X s.t it coincides with XZ plane.

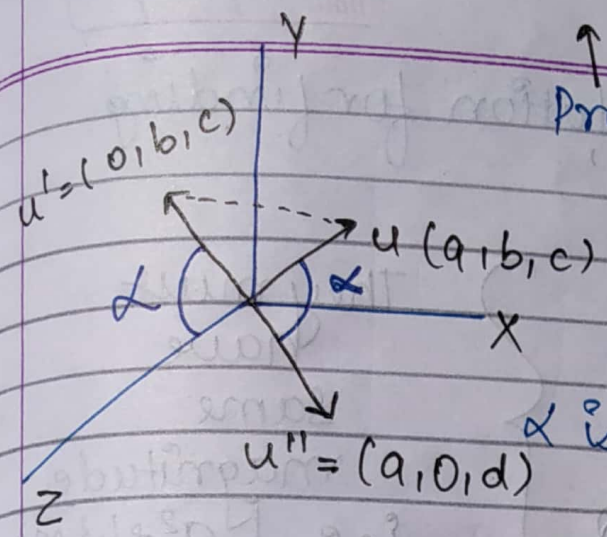
Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

coinciding end pts with plane

CLASSTIME Page No.
Date / /



Project un on YZ plane
$= u'(0,b,c)$

x coordinate becomes 0.

α is angle by u' with z axis.

unit vector along z → (0,0,1)
and $u' = (0,b,c)$

$$u' \cdot z = |u'|\,|z| \cos\theta$$

$$c = \sqrt{b^2+c^2}\,(1)\cos\theta$$

$$\cos\theta = \cos\alpha = \frac{c}{\sqrt{b^2+c^2}} = \frac{c}{d}$$

Also $u' \times z = |u'|\,|z| \sin\alpha$.

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

NOTE:- We did projection of 'u' so as to calculate 'α' and we know that angle b/w u' and z = angle b/w u & u''.

in rotation matrix, we have cosθ & sinθ

$$R_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & -b/d & 0 \\ 0 & b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

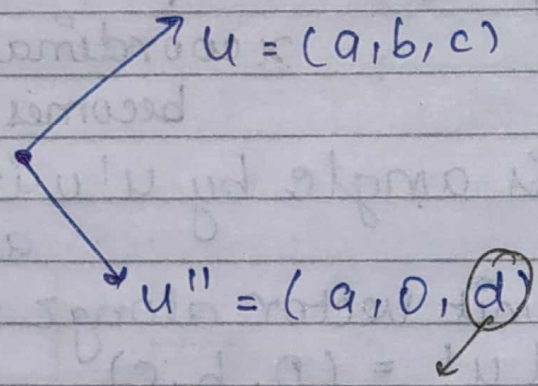
rotation abt x axis

Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

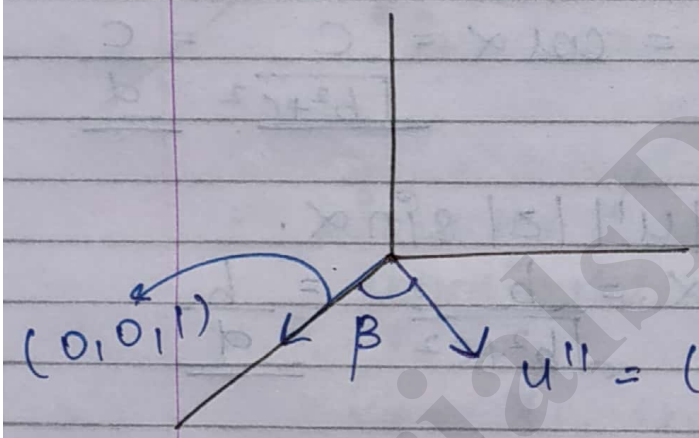
This was the 1st rotation for finding angle 'α'



They must have same magnitude i.e $\sqrt{a^2+b^2+c^2}$

∴ This becomes $\sqrt{b^2+c^2}$

- Rotate u'' abt y axis s.t it concides with z axis.



$$d = (1)(\sqrt{a^2+d^2})\cos\beta$$

$$\cos\beta = \frac{d}{\sqrt{a^2+d^2}}$$

$$= \frac{d}{\underbrace{\sqrt{a^2+b^2+c^2}}_{\text{mag. is 1}}} = d$$

$$\sin\beta = a$$

$$\therefore R = \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Q Rotation matrices of rotation abt. x and y axis

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Rotate by θ abt z axis

$$M = T^{-1} R_1^{-1} R_2^{-1} R_3(\theta) R_2(\beta) R_1(\alpha) T$$

post multiplication of obj.

NOTE:- If obj. is pre multiplied, order will reverse.

- Coincide a plane (arbitrary) with a planes given (XY, YZ, ZX)
- To do this, with a plane we have a normal ∴ we coincide normal with a ~~plane~~ ∥ axis. instead of coinciding plane.
- Coinciding normal with z axis coincides it with xy plane.

Reflection of plane through an arb. plane (neither XY, YZ or XZ)

* How to find coordinates of Normal?

If in a plane we have 3 pts. coordinates of normal are cross product of adjacent edge vectors. Those normal planes can be used as coordinates of the axis.

$$[M] = [T][R_x][R_y][R_f][R_y]^{-1}[R_x]^{-1}[T]^{-1}$$

reflection abt xy plane

Tutorialsduniya.com

TutorialsDuniya.com

Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc of following subjects from https://www.tutorialsduniya.com

- C and C++
- Programming in Java
- Data Structures
- Computer Networks
- Android Programming
- PHP Programming
- JavaScript
- Java Server Pages
- Python
- Microprocessor
- Artificial Intelligence
- Machine Learning
- Computer System Architecture
- Discrete Structures
- Operating Systems
- Algorithms
- DataBase Management Systems
- Software Engineering
- Theory of Computation
- Operational Research
- System Programming
- Data Mining
- Computer Graphics
- Data Science

- Compiled Books: https://www.tutorialsduniya.com/compiled-books
- Programs: https://www.tutorialsduniya.com/programs
- Question Papers: https://www.tutorialsduniya.com/question-papers
- Python Notes: https://www.tutorialsduniya.com/python
- Java Notes: https://www.tutorialsduniya.com/java
- JavaScript Notes: https://www.tutorialsduniya.com/javascript
- JSP Notes: https://www.tutorialsduniya.com/jsp
- Microprocessor Notes: https://www.tutorialsduniya.com/microprocessor
- OR Notes: https://www.tutorialsduniya.com/operational-research

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

3-D Shearing

2D

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

$$x' = x + Sh_x \cdot y.$$

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

y coordinates change wrt. distance from x.

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

In both dir^n → both x and y coordinate change.

$$Sh_{xy} = \begin{bmatrix} 1 & Sh_x & 0 \\ Sh_y & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

For 3D]

4X4 → matrix → (Shearing of x with be done wrt Dist. from y and z).

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a & b & 0 \\ c & 1 & d & 0 \\ e & f & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

3-D Shearing

2D

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

$$x' = x + Sh_x \cdot y.$$

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

y coordinates change wrt. distance from x.

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

In both dir^n → both x and y coordinate change.

$$Sh_{xy} = \begin{bmatrix} 1 & Sh_x & 0 \\ Sh_y & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

For 3D]

4X4 → matrix → (Shearing of x with be done wrt Dist. from y and z).

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a & b & 0 \\ c & 1 & d & 0 \\ e & f & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

$x' = x + ay + bz$ → shx wrt z.

shx wrt y

$y' = cx + y + dz$

$z' = ex + fy + z$

* Translation →
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ tx & ty & tz & 1 \end{bmatrix}$$

* Scaling →
$$\begin{bmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

* Rotation →
$$\begin{bmatrix} \begin{bmatrix} \text{changes} \\ \text{are} \end{bmatrix} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

$$\left[\begin{array}{ccc|c} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ \hline m & n & o & p \end{array}\right]$$

4x4 generalised matrix divided into 4 parts.

$[m \quad n \quad o]_{1\times3}$ → responsible for Translation.

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Scaling
Reflection
Rotation
Shearing.

$$\begin{bmatrix} a & b & c \\ e & f & g \\ i & j & k \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & s \end{bmatrix}$$ → scaling Transformation.

$\Updownarrow$

s → responsible for overall scaling.

$$\begin{bmatrix} 1/s & 0 & 0 & 0 \\ 0 & 1/s & 0 & 0 \\ 0 & 0 & 1/s & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scaling factor → s.

If $s > 1$ → size reduces
If $s < 1$ → expanding size of element.

∴ [p] → responsible for overall scaling.

Now, last column.

$$\begin{bmatrix} d \\ h \\ l \\ p \end{bmatrix} \rightarrow \text{generally} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$ → s for overall scaling

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

All 3D Transformations are classified as Affine & Perspective Transformation.

Affine → When last col values are

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

(parallel lines remain parallel after transformation)

Whenever these values change acc. then known as perspective

TutorialsDuniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

27 march 19

* Isometric projections :-

$$\{ f_x = f_y = f_z \}$$

$$f_x^2 = f_y^2$$

$$\sin^2\phi = \frac{\sin^2\theta}{1-\sin^2\theta} \quad \text{--- (1)}$$

$$f_y^2 = f_z^2$$

$$\sin^2\phi = \frac{1-2\sin^2\theta}{1-\sin^2\theta} \quad \text{--- (2)}$$

equating (1) & (2)

$$\sin^2\theta = 1/3$$

$$\sin\theta = \pm\sqrt{1/3}$$

$$\therefore \sin^2\phi = \frac{\frac{1}{3}}{1-\frac{1}{3}} = \frac{1}{3} \times \frac{3}{2} = \frac{1}{2}$$

$$\phi = 45^\circ$$

Substituting ϕ and θ in $R_y(\phi)\,R_x(\theta)$ gives us req. rotation matrices.

→ There are 4 possible values of isometric projections

$\pm\theta, \pm\phi$ } combinations

Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

To find foreshortening factors

$$f_x = \sqrt{\cos^2\theta} = \sqrt{2/3}$$
$$= \underline{0.8165}$$

i.e 0.81% is projected length out of the total length.

An isometric projection is a special case of a dimetric projection.

→ The angle that the projected x axis makes with the horizontal is imp. in manual construction of isometric projections.

Transforming unit vector along x axis using isometric projection matrix yields.

$$[V_x^*] = [1\ 0\ 0\ 1] \underbrace{\begin{bmatrix} \cos\phi & \sin\phi\sin\theta & 0 & 0 \\ 0 & \cos\theta & 0 & 0 \\ \sin\phi & -\cos\phi\sin\theta & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{R_y(\phi)\,R_x(\theta)\,P_z}$$

$$= [\cos\phi \quad \underline{\sin\phi\sin\theta} \quad 0 \quad 1]$$

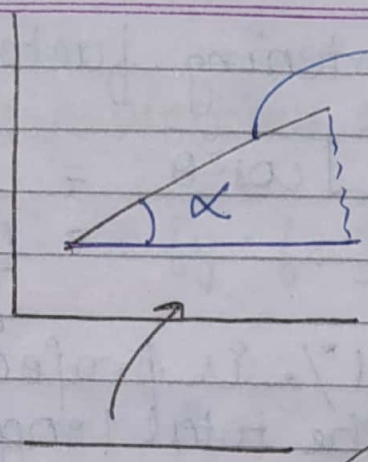
Angle b/w projected x axis & horizontal.

$$\tan\alpha = \frac{y_x^*}{x_x^*} = \frac{\sin\phi\sin\theta}{\cos\phi}$$
$$= \underline{\pm\sin\theta}$$

Tutorialsduniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com



line upon projection

$\therefore \alpha = \tan^{-1}(\pm 35.26°)$

$= \pm 30°$

{angle of isometric projection}

EX3-15 (Book) $\phi = -45°$
$\theta = +35.26°$

Q (Derivation of Isometric, Dimetric & Trimetric projection matrix)

- Oblique Projections -

↓

parallel projection (C.O.P = ∞)
projectors are parallel to each other but they are making an oblique angle (<90°) with POP.

On basis of angle made

↙ ↘

cavalier cabinet.

→ The face which is ∥ to POP, only that face will have its true shape & size.

Tutorialsduniya.com

https://www.tutorialsduniya.com

Cavalier Projection

CLASSTIME Page No.
Date / /

- A cavalier projection is obtained when angle b/w oblique projectors & plane of projection is 45°

- Foreshortening factors for all 3 principal directions are equal. Thus resulting fig. appears too thick.

- Cabinet Projection

→ Cabinet projection is one for which foreshortening factors for edges ⊥ to POP is one-half.

↓

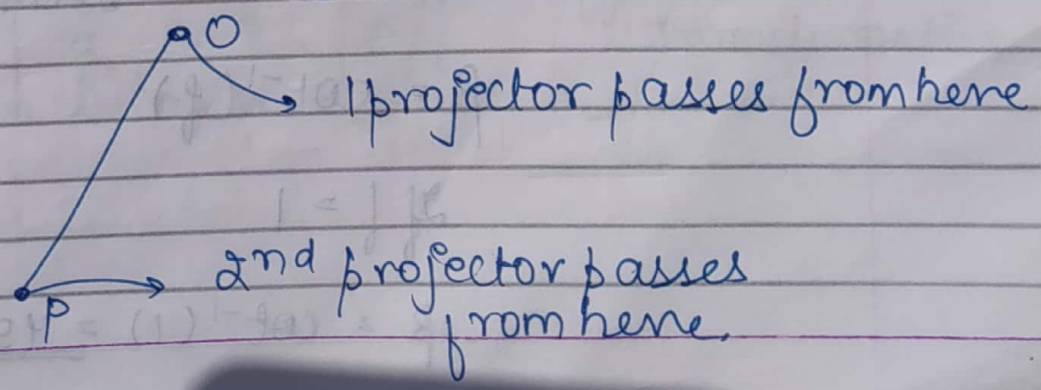
(provided plane is x-y)

→ lines ∥ to x & y foreshortens by an amt but lines ∥ to z axis gets foreshortened by 1/2.

→ angle b/w projectors and plane of projection is $\cot^{-1}(1/2) = 63.43°$.

* How to get Oblique projection matrix-

Consider unit vector in z dir^n
[0 0 1]



Contributor: Riya Goel

https://www.tutorialsduniya.com

→ If all projectors are made to meet at a pt., then they intersect by making an angle β which is same as angle b/w projected line & projector / unit vector.

② Translate pt. P by −a and −b.

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

in 3D →

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

This becomes shearing matrix in 3-D

$$\left\{ \begin{array}{l} a = f\cos\alpha \\ b = f\sin\alpha \end{array} \right\}$$

angle with horizontal axis.

length of projected line.

For cabinet projection → this f will become 1/2.

$$\beta = \cot^{-1}(f)$$

If $f = 1$

$$\beta = \cot^{-1}(1) = \underline{45^\circ}.$$

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Oblique projection matrix

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

$\left\{ \begin{array}{l} 2f_f = 1 \text{ (cavalier)} \\ 2f_f = 1/2 \text{ (cabinet)} \end{array} \right\}$

- Perspective Transformations :-

C.O.P = finite distance
projectors are not parallel to each other i.e They meet at a pt.

If in its col → values are other than 0
then perspective, otherwise
(lengths & angles are not affine. preserved)

$$\begin{bmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ l & m & n & 1 \end{bmatrix} \rightarrow \text{affine Transformation}$$

TutorialsDuniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

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

perspective Transformation.

$$\begin{bmatrix} & & 0 & p \\ & & 0 & 0 \\ & & 0 & 0 \\ & & 0 & 0 \end{bmatrix}$$

Projection on $z = 0$

perspective projection.

or perspective Transf. followed by orthographic Transformation.

$$\begin{bmatrix} & & & p \\ & & & q \\ & & & r \\ & & & 1 \end{bmatrix}$$

If any of $p, q, r \neq 0$ single pt. perspective Transformation.

Based on p, q, r values as 0, they tell where is centre of projection.

If $q \neq 0$ c.o.p lies along y axis.

TutorialsDuniya.com

Contributor: Riya Goel

https://www.tutorialsduniya.com

If $p \neq 0$ C.O.P lies along x axis
If $r \neq 0$ " " " z axis.

finite dist. along one of the axis

Single pt. perspective Transformation.

$$[x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \ y \ z \ rz+1]$$

To homogenize coordinates, last value must be 1

$$[x^* \ y^* \ z^* \ 1] = \left[\frac{x}{rz+1} \ \frac{y}{rz+1} \ \frac{z}{rz+1} \ 1 \right]$$

A perspective projection on $z=0$ is

$$[T] = [P_T][P_z]$$

orthographic projection

Single pt perspective Transformation matrix

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

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

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

$$[x\ y\ z\ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= [x\ \ y\ \ 0\ \ rz+1]$$

$$[x^*\ \ y^*\ \ z^*] = \left[\frac{x}{rz+1}\ \ \frac{y}{rz+1}\ \ 0\ \ 1\right]$$

If C.O.P — x_c

$$p = \frac{-1}{x_c}$$

If C.O.P — z_c

$$r = -\frac{1}{z_c}$$

If C.O.P — y_c

$$q = \frac{-1}{y_c}$$

* Vanishing point :-

TutorialsDuniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

Perspective Projections

1 point (gives less info. as seen from 1 pt.)

2 point

3 point. (most info. can be obtained)

4x4 Transformation 3-D

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

Affine Transformation (|| lines remain parallel)

Other than these values

Perspective Transformation

(last col. Other than 0001)

$$\begin{bmatrix} & & p \\ & & q \\ & & r \\ & & 1 \end{bmatrix}$$

If $p \neq 0$ or $q \neq 0$ or $r \neq 0$ (1 point) one value is non zero.

If 2 of these values are non-zero (2 point) $p \neq 0, r \neq 0, q = 0$; $p \neq 0, q \neq 0, r = 0$.

If all 3 are non zero (p, q, r) (3 point).

Contributor: Riya Goel

https://www.tutorialsduniya.com

Possible Directions

CLASSTIME Page No.
Date / /

$p = \frac{-1}{x_c}$ $\begin{bmatrix} p \\ 0 \\ 0 \\ 1 \end{bmatrix}$ → C.O.P lies along +ve x axis. (x_c)

$q = \frac{-1}{y_c}$ $\begin{bmatrix} 0 \\ q \\ 0 \\ 1 \end{bmatrix}$ → C.O.P lies along +ve y axis. (y_c)

$r = \frac{-1}{z_c}$ $\begin{bmatrix} 0 \\ 0 \\ r \\ 1 \end{bmatrix}$ → C.O.P lies along z axis. (z_c)

Perspective Transformation followed by orthographic Transformation/projection along an axis.

Perspective ~~Transformation~~ Projection

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{P_{Tatz}} \quad \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{P_Z}$$

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

→ After perspective Transformation, lines no longer remain parallel. Thus earlier pt. lying at ∞ but now the intersection pts. become visible

↓

∴ Vanishing pt.
(can be found using C.O.P)

If C.O.P → x_c
∴ Vanishing pt. → $-x_c$

If C.O.P → z_c
Vanishing pt. meets z axis at a particular pt. i.e $-z_c$

If C.O.P → y_c
Vanishing pt. → $-y_c$

* Principle Vanishing Pt.

* 3 types of single pt. Perspective Transformation

↓

2n Transformation → 3D pt. remains 3D
2n Projection → 3D pt. becomes 2D (one col. becomes 0)

(1)
$$\begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Type 1 Transformation
C.O.P — x axis.

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

② along y axis (y_c)

③ along z axis (z_c)

eg:- If C.O.P → $(0, 0, -z_c, 1)$
Vanishing pt.
↓
$(0, 0, -z_c, 1)$

If given p, q, r we can find values of x_c, y_c & z_c

$$p = \frac{-1}{x_c}, \quad q = \frac{-1}{y_c}, \quad r = \frac{-1}{z_c}$$

To get a better view, we have 2 pt. perspective Transformations
↓
2 non zero values
2 vanishing pts

$$\begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1 in opp. x dir^n & other in opp. y dir^n

$COP_x = (0, 0, -1/p, 1)$ ✗

$COP_y = (0, 0, -1/q, 1)$ ✗

→ $COP_x = (\underbrace{-1/p}_{x_c}, 0, 0, 1)$

→ $COP_y = (0, \underbrace{-1/q}_{y_c}, 0, 1)$

Tutorialsduniya.com

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

∴ Vanishing pts.

$COP_x = (1/p, 0, 0, 1)$

$COP_y = (0, 1/q, 0, 1)$

Most info. is obtained by 3 pt. perspective Transformation

↓

3 diff COP & 3 diff. vanishing pts.

$$\begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ques in exam

Q. Given position vectors of an object. The obj. is Transformed with 2 pt. with $x_c = -10$ and C.O.P along y is -10 (i.e $y_c = -10$). Perspective transformation is to applied & project on z axis.

$x_c = -10, \quad p = 1/10$

$y_c = -10, \quad q = 1/10$

$$\therefore \text{matrix} = \begin{bmatrix} 1 & 0 & 0 & 1/10 \\ 0 & 1 & 0 & 1/10 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

∴ Projection →

Contributor: Riya Goel

https://www.tutorialsduniya.com

CLASSTIME Page No.
Date / /

$$[X] \begin{bmatrix} 1 & 0 & 0 & 1/10 \\ 0 & 1 & 0 & 1/10 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{Orthographic Transformation along } z.}$$

object.

for a cube
↓
{8 vertices} Obj matrix → 8x4.

Q Apply 3 pt. Transformation at $x_c = -10$, $y_c = -10$ and $z_c = 10$.
Projection is along $z = 0$ plane.

$p = 1/10$, $q = 1/10$, $r = -1/10$.

$$[X] \underbrace{\begin{bmatrix} 1 & 0 & 0 & 1/10 \\ 0 & 1 & 0 & 1/10 \\ 0 & 0 & 1 & -1/10 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{P_r} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \underbrace{0}_{\text{along } z.} & 1 \end{bmatrix}$$

Q Translate an object and apply 1 pt. Transformation projection at z axis.

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & 0 & 1 \end{bmatrix}}_{\text{Translation in 3D}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{1 pt. Transformation matrix.}}$$

along x

along y.

Tutorialsduniya.com

Contributor: Riya Goel

TutorialsDuniya.Com

Visit https://www.tutorialsduniya.com/question-papers to get Delhi University Question Papers of all the following courses,

- General Electives
- AECC
- B.Sc. (H) Computer Science
- B.Sc. (H) Mathematics
- B.Sc. (H) Chemistry
- B.Sc. (H) Physics
- B.Sc. (H) Electronics
- B.Sc. (H) Statistics
- B.Sc. (H) Botany
- B.Sc. (H) Geology
- B.Sc. (H) Home Science
- B.Sc. (H) Zoology
- B.Com (H)
- B.Sc. (H) Microbiology
- B.Sc. Physical Science
- B.Sc. Mathematical Science
- B.A. Programme
- BMS
- B.A. (H) Psychology
- B.Sc. Life Science
- B.A. (H) Economics
- B.A. (H) English
- B.A. (H) Hindi
- B.A. (H) History
- B.A. (H) Journalism
- B.A. (H) Philosophy
- B.A. (H) Political Science
- B.A. (H) Geography
- B.A. (H) Sanskrit
- B.A. (H) Sociology

❖ **Get FREE Compiled Books, Notes, Programs, Books, Question Papers with Solution* etc at https://www.tutorialsduniya.com**

- **For any query, contact us at *help@tutorialsduniya.com***

✓ **Facebook: https://www.facebook.com/tutorialsduniya**

✓ **Youtube: https://www.youtube.com/user/TutorialsDuniya**

✓ **LinkedIn: https://www.linkedin.com/company/tutorialsduniya**