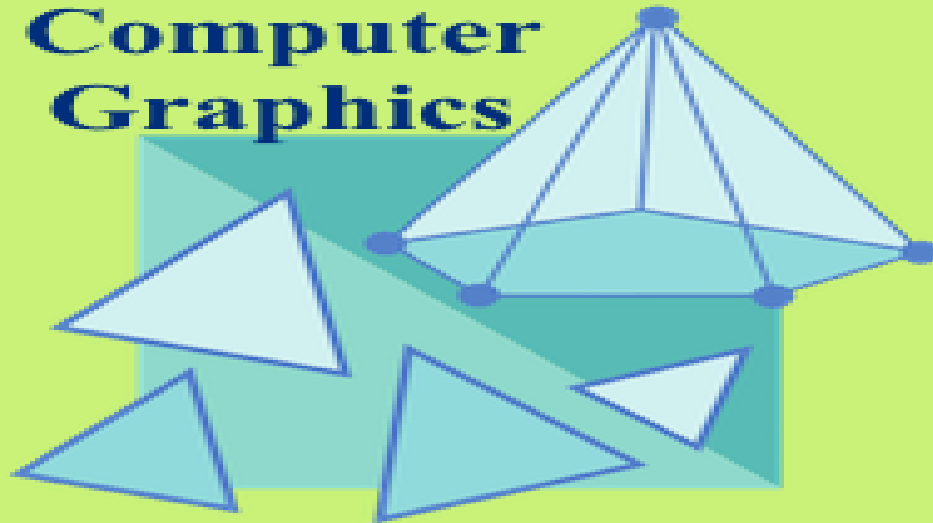


Computer Graphics with OpenGL



ASAN BAKER

**Computer
Graphics**



Introduction of Computer Graphics



- Computer graphics is an art of drawing pictures, lines, charts, etc using computers with the help of programming. Computer graphics is made up of number of pixels.
- Pixel is the smallest graphical picture or unit represented on the computer screen.
- Computer Graphics is the creation of pictures with the help of a computer. The end product of the computer graphics is a picture it may be a business graph, drawing, and engineering design.
- Today computer graphics is entirely different from the earlier one. It is not possible. It is an interactive user can control the structure of an object of various input devices.

Computer Graphics: Possible Definitions



- **Computer graphics is a sub-field of computer science which studies methods for digitally synthesizing and manipulating visual content.** Although the term often refers to the study of three-dimensional computer graphics, it also encompasses two-dimensional graphics and image processing.
- It is the use of computers to create and manipulate pictures on a display device. It comprises of software techniques to create, store, modify, represents pictures.

Application of Computer Graphics



- **1. Education and Training:** Computer-generated model of the physical, financial and economic system is often used as educational aids. Model of physical systems, physiological system, population trends or equipment can help trainees to understand the operation of the system.
- For some training applications, particular systems are designed. For example Flight Simulator.



- **Flight Simulator:** It helps in giving training to the pilots of airplanes. These pilots spend much of their training not in a real aircraft but on the ground at the controls of a Flight Simulator.
- Advantages:
- Fuel Saving
- Safety
- Ability to familiarize the training with a large number of the world's airports.



- **2. Use in Biology:** Molecular biologist can display a picture of molecules and gain insight into their structure with the help of computer graphics.
- **3. Computer-Generated Maps:** Town planners and transportation engineers can use computer-generated maps which display data useful to them in their planning work.
- **4. Architect:** Architect can explore an alternative solution to design problems at an interactive graphics terminal. In this way, they can test many more solutions that would not be possible without the computer.



5. Presentation Graphics: Example of presentation Graphics are bar charts, line graphs, pie charts and other displays showing relationships between multiple parameters. Presentation Graphics is commonly used to summarize

- Financial Reports
- Statistical Reports
- Mathematical Reports
- Scientific Reports
- Economic Data for research reports
- Managerial Reports
- Consumer Information Bulletins
- And other types of reports



- **6. Computer Art:** Computer Graphics are also used in the field of commercial arts. It is used to generate television and advertising commercial.
- **7. Entertainment:** Computer Graphics are now commonly used in making motion pictures, music videos and television shows.
- **8. Visualization:** It is used for visualization of scientists, engineers, medical personnel, business analysts for the study of a large amount of information.



- **9. Educational Software:** Computer Graphics is used in the development of educational software for making computer-aided instruction.
- **10. Printing Technology:** Computer Graphics is used for printing technology and textile design.

Tasks of Computer Graphics



- Imaging : representing 2D images
- Modeling : representing 3D images
- Rendering : constructing 2D images from 3D images
- Animation: simulating changes of objects over time

Interactive and Passive Graphics



(a) Non-Interactive or Passive Computer Graphics:

- In non-interactive computer graphics, the picture is produced on the monitor, and the user does not have any control over the image, i.e., the user cannot make any change in the rendered image. One example of this is titles shown on T.V.
- Non-interactive Graphics involves only one-way communication between the computer and the user, User can see the produced image, and he/she cannot make any change in the image.



(b) Interactive Computer Graphics:

- In interactive Computer Graphics user have some controls over the picture, i.e., the user can make any change in the produced image. One example of it is the ping-pong game.
- Interactive Computer Graphics require two-way communication between the computer and the user. A User can see the image and make any change by sending his command with an input device.



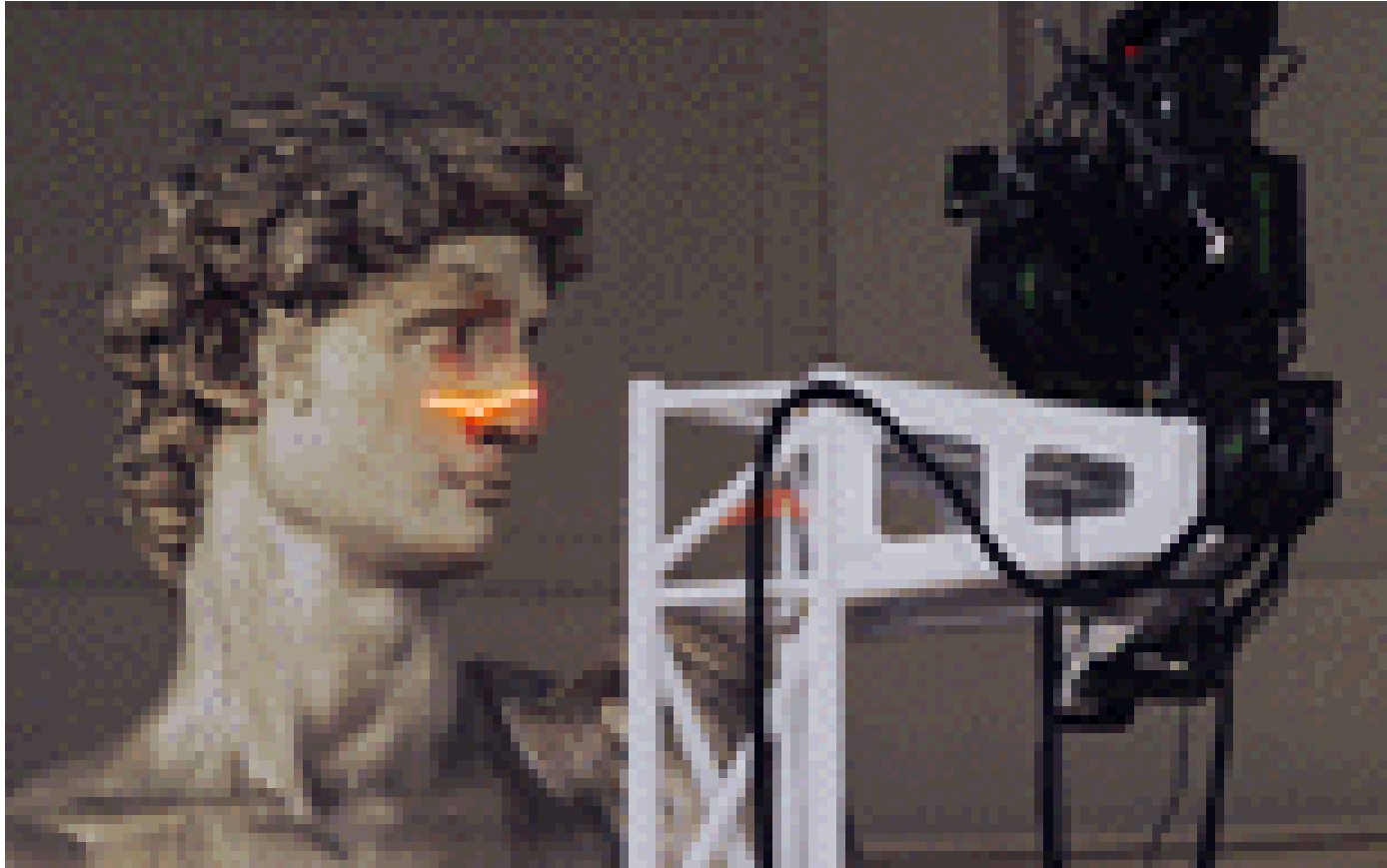
Advantages of Interactive Computer Graphics :

- Higher Quality
- More precise results or products
- Greater Productivity
- Lower analysis and design cost
- Significantly enhances our ability to understand data and to perceive trends.



Examples of computer graphics

3D Models - Scanning



(<http://graphics.stanford.edu/projects/mich/>).

Scanned Model



On the left is a photograph of Michelangelo's David.
On the right is a computer rendering made from a geometric model.
(<http://graphics.stanford.edu/projects/mich/>).

3D models - Generated



Realistic modeling and rendering of plant ecosystems

Geometry – Projection, Visibility



The Walkthru Project

<http://www.cs.unc.edu/~walk/>

Lighting



- Modeling Lighting
- Reflectance
- Texture
- Shadows (visibility)
- Interreflections

Lighting

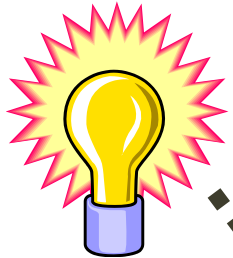


(from Debevec)

Light



Source emits photons



And then some reach the eye/camera.



Photons travel in a straight line

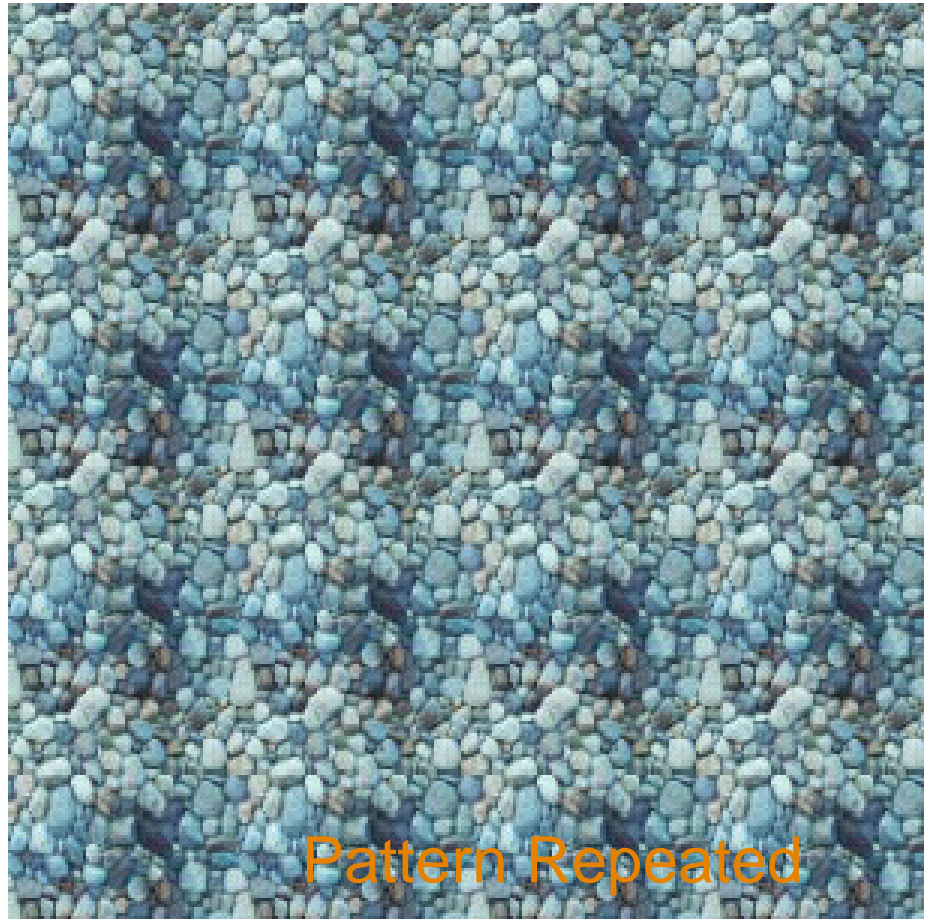


- When they hit an object they:
- bounce off in a new direction
 - or are absorbed
 - (exceptions later).

Texture

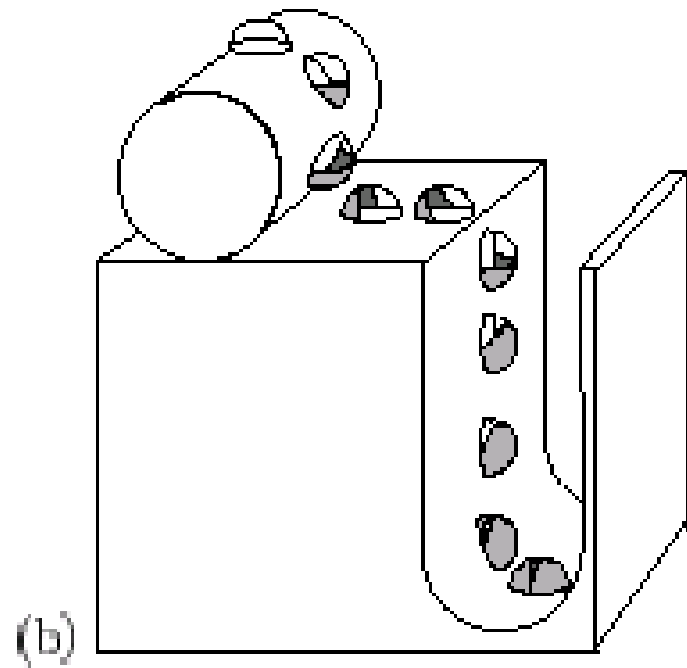
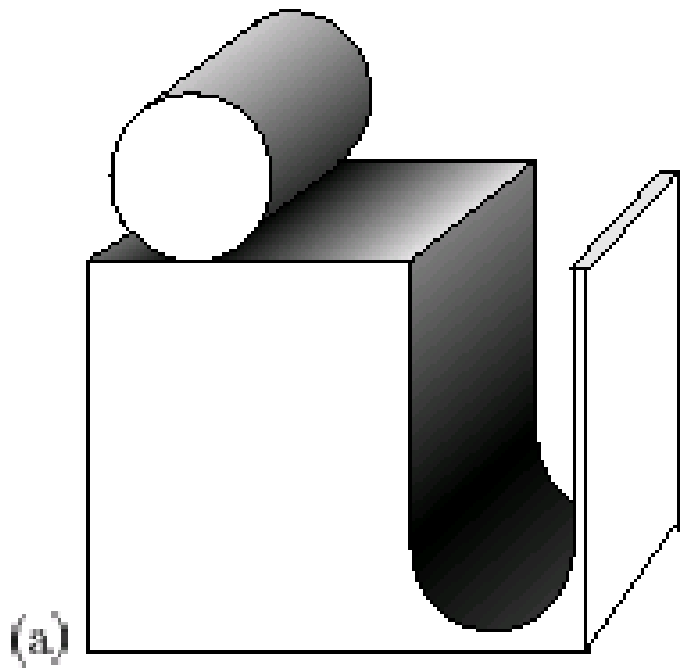


Photo



Pattern Repeated

Shadows



(from Langer and Zucker)

Interreflections



Falling Water model, Bruce Walter thesis.

<http://www.graphics.cornell.edu/%7Ebjw/bwthesis.pdf>

Physically real motion

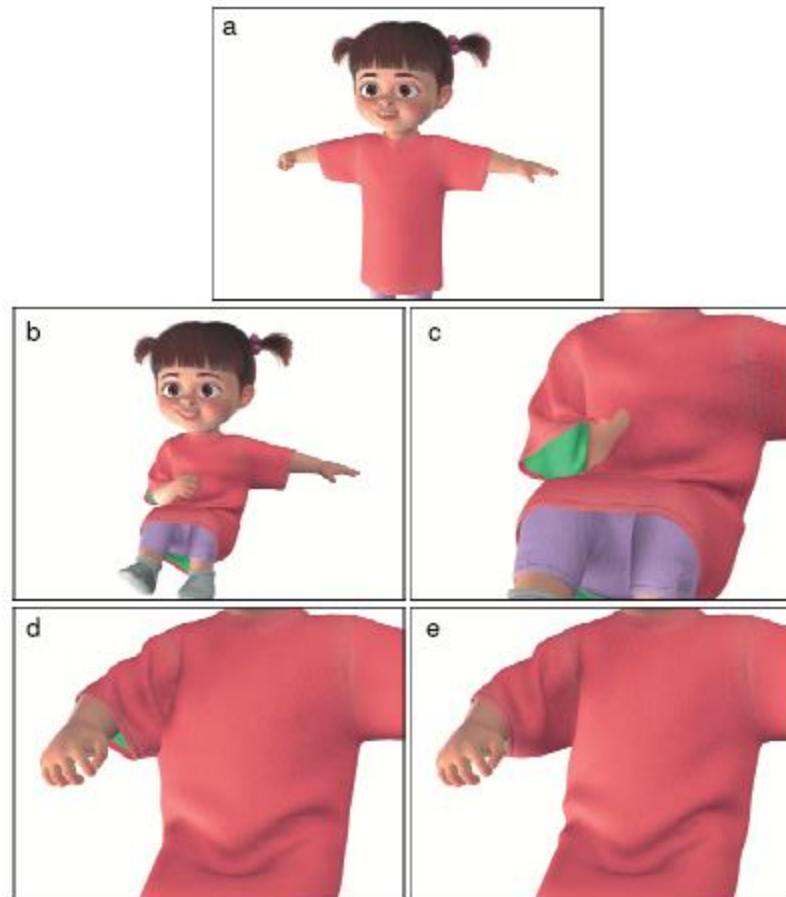


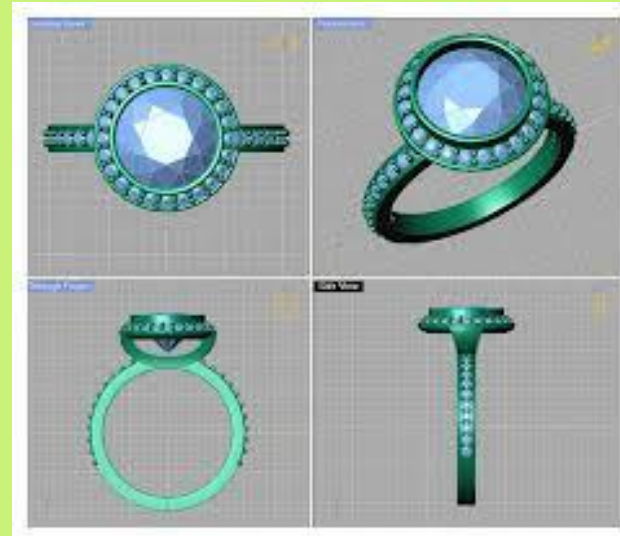
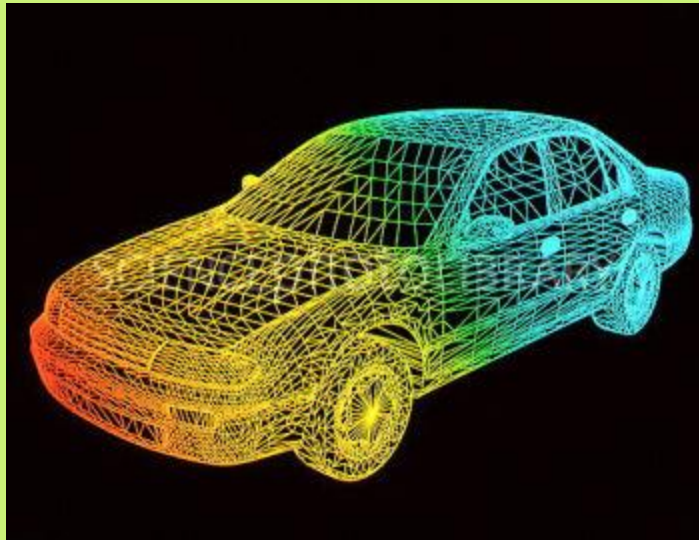
Figure 10: (a) Starting pose. (b) Arm moves in tightly. (c) Close-up view of (b) with right arm invisible. Note how the arm position forces cloth to intersect both itself and the body. (d) Without GIA, a cloth/cloth intersection persists as the arm pulls out, snagging the sleeve. (e) The same frame as (d), but using GIA, the cloth doesn't snag as the arm pulls out.

Images based on realism



(De Carlo and Santella)

Computer-Aided Design



Architectural Walkthroughs



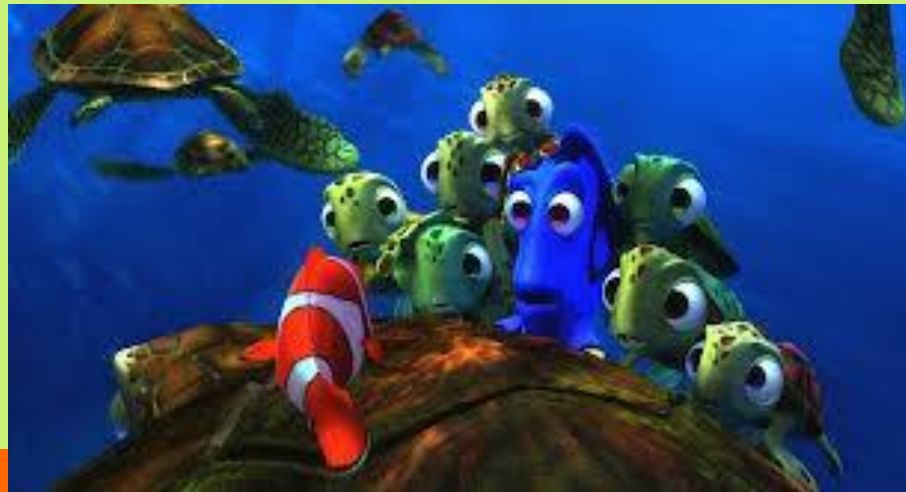
Medical Imaging



Computational Simulations



Film Production



Computer Games



Computer Graphics Hardware

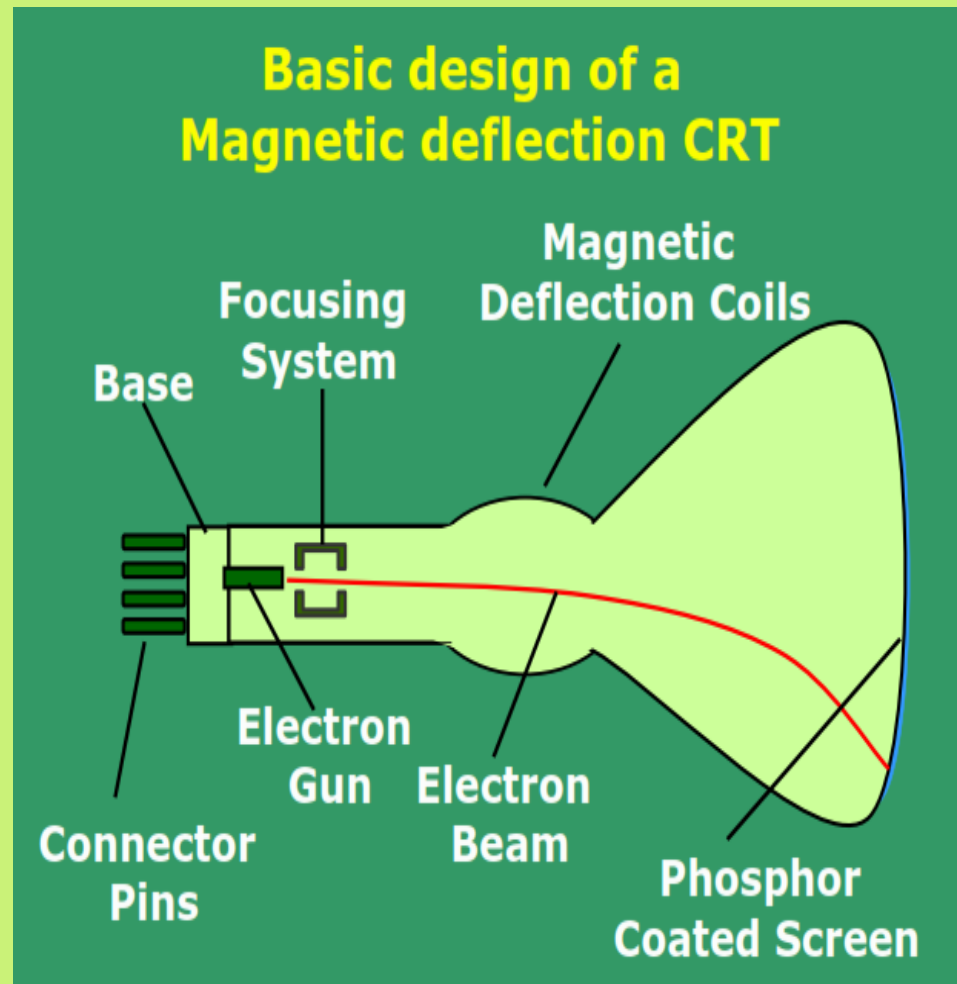


- **Video display devices:**
 - Cathode Ray Tube (CRT)
 - Liquid Crystal Display(LCD)
 - Plasma panels
 - Thin-film electroluminescent display
 - Light-emitting diodes (LED)
- **Hard copy devices**
 - Ink-jet printer
 - Laser printer
 - Film recorder
 - Pen plotter

CRT details



- A beam of electrons (cathode rays) emitted by an electronic gun, passes through focusing and deflection systems that direct the beam toward specified positions on the phosphor-coated screen.
- The frequency at which the picture is redrawn on the screen is referred to as the **refresh rate**.



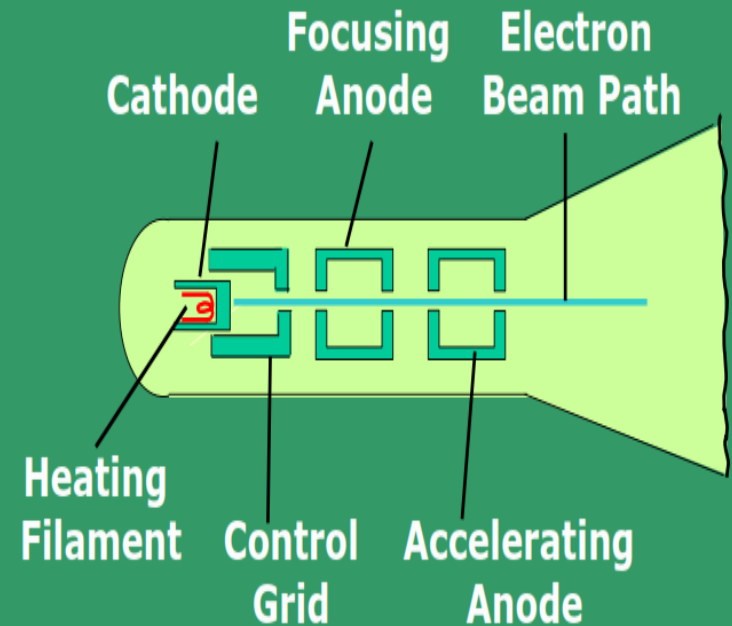


Electron gun

- Contains a filament that, when heated, emits a stream of electrons
- Electrons are focused with an electromagnet into a sharp beam and directed to a specific point of the face of the picture tube
- The front surface of the picture tube is coated with small phosphor dots
- When the beam hits a phosphor dot it glows with a brightness proportional to the strength of the beam and how long it is hit

- The accelerating voltage can be generated with a positively charged metal coating on the inside of the CRT envelope near the phosphor screen, or an accelerating anode, can be used to provide the positive voltage.
- Sometimes the electron gun is designed so that the accelerating anode and focusing system are within the same unit.

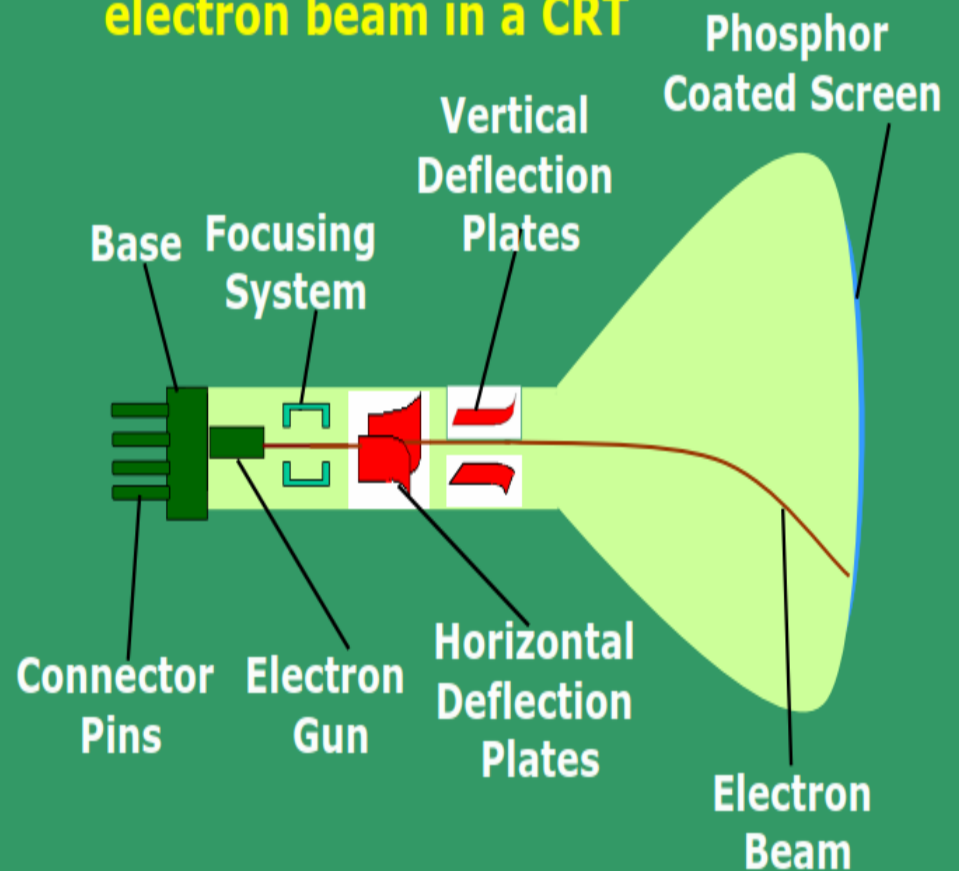
Operation of an electron gun with an accelerating anode



- When electrostatic deflection is used two pairs of parallel plates are mounted inside the CRT envelope. One pair of plates is mounted horizontally to control the vertical deflection, and the other pair is mounted vertically to control horizontal deflection.



Electrostatic deflection of the electron beam in a CRT



CRT characteristics



- **Intensity** is proportional to the number of electrons repelled in beam per second (**brightness**)
- **Resolution** is the maximum number of points that can be displayed without overlap; is expressed as number of horizontal points by number of vertical points; points are called pixels (picture elements); example: resolution 1024 x 768 pixels. Typical resolution is 1280 x 1024 pixels.
- High-definition systems: high resolution systems.



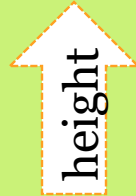
- **Focusing**
 - **Focusing** forces the electron beam to converge to a point on the monitor screen
 - Can be electrostatic (lens) or magnetic (field)
- **Deflection**
 - **Deflection** directs the electron beam horizontally and/or vertically to any point on the screen
 - Can be controlled by electric or magnetic fields
 - Magnetic coils: two pairs (top/bottom, left/right) of tube neck
 - Electric plates: two pairs (horizontal, vertical)



Graphics Elements



- Pixel (picture element): the smallest element of an image
- Pixel attributes:
 - Coordinates
 - Intensity
- Resolution: number of pixels in an image
eg: 700p X 800p





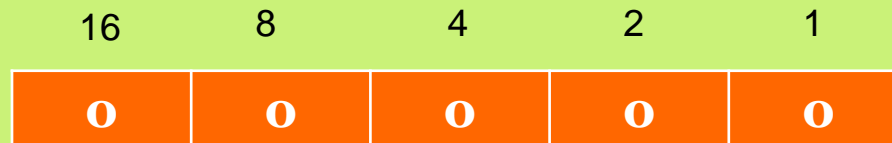
- Max number of colors(intensity) = $2^{\text{No. of bits}}$
 $= 2^8 = 256$

Each pixel have 0 or 1

Example : suppose a pixel with 5 bits find:

Maximum number of colors = $2^5 = 32$

Black = 0



White = 31



More bits in a pixel means more color(intensity)= better quality and more detailed image

Examples



- Q1: given an image with 8bit/pixel find the maximum number of intensities

$$=2^8 = 256$$

- Q1: given an image with 8bit/pixel find the maximum intensity value of this image.

8 bit pixel



Graphics Models



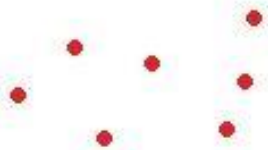
Raster display model

Vector display model

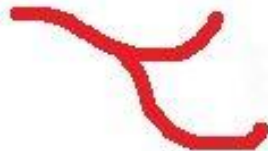
Raster Graphics	Vector Graphics
They are composed of pixels.	They are composed of paths.
In Raster Graphics, refresh process is independent of the complexity of the image.	Vector displays flicker when the number of primitives in the image become too large.
Graphic primitives are specified in terms of end points and must be scan converted into corresponding pixels.	Scan conversion is not required.
Raster graphics can draw mathematical curves, polygons and boundaries of curved primitives only by pixel approximation.	Vector graphics draw continuous and smooth lines.
Raster graphics cost less.	Vector graphics cost more as compared to raster graphics.
They occupy more space which depends on image quality.	They occupy less space.
File extensions: .BMP, .TIF, .GIF, .JPG	File Extensions: .SVG, .EPS, .PDF, .AI, .DXF



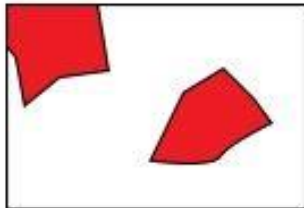
Vector



Points

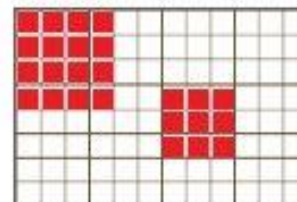
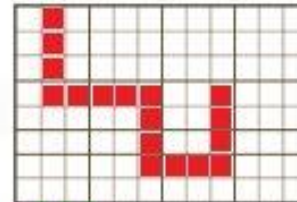
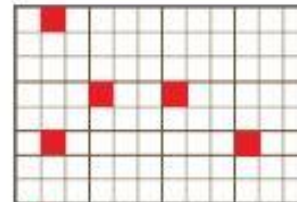


Lines



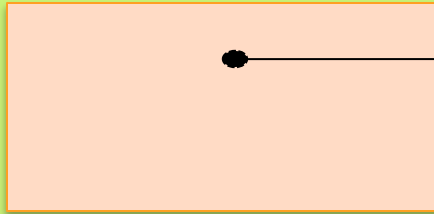
Areas

Raster



Raster types

1. Indexed images

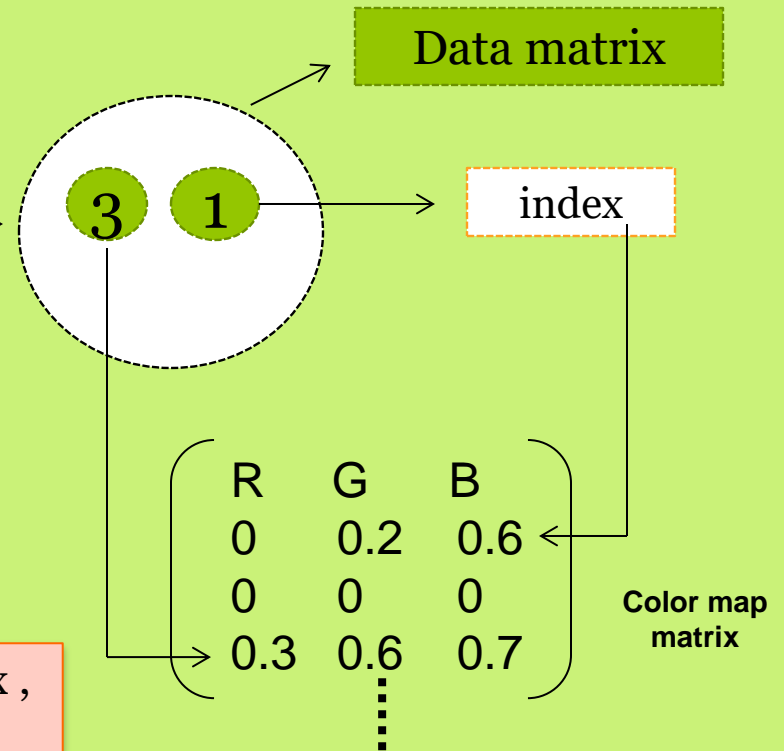


Suppose that this is an image
this image consist of

- Data matrix
- Color map matrix

Data matrix contain indices of the color map matrix ,
the resolution is the same as the original image

Color map matrix
No. of columns are 3 always (R,G,B)
No. of rows no. of colors
The values ranges from 0 to 1





- What is the resolution of the data matrix and the color map matrix for an images with 300 x 300 and black.

data matrix = 300 x 300

color map matrix = 3 x 1 ← No. of colors

↑
No. of columns
R,G,B



- Consider an index image of ten colors and 20 x 20 pixels, what is the resolution of color map matrix.

Color map matrix = 3 x 10

Data matrix = 20 x 20



2. Intensity images (gray scale images)

Range (0 , 255) for 8 bit pixel

Range (0 ,31) for 5 bit pixel

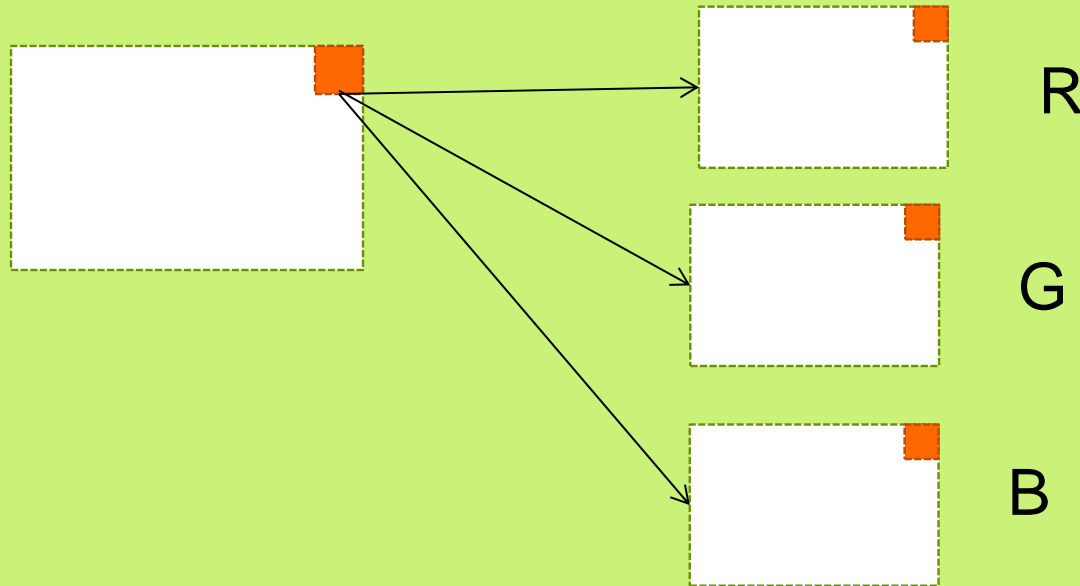
Depending on no. of
bits per pixels

color gradient between black and white [0 , 255]

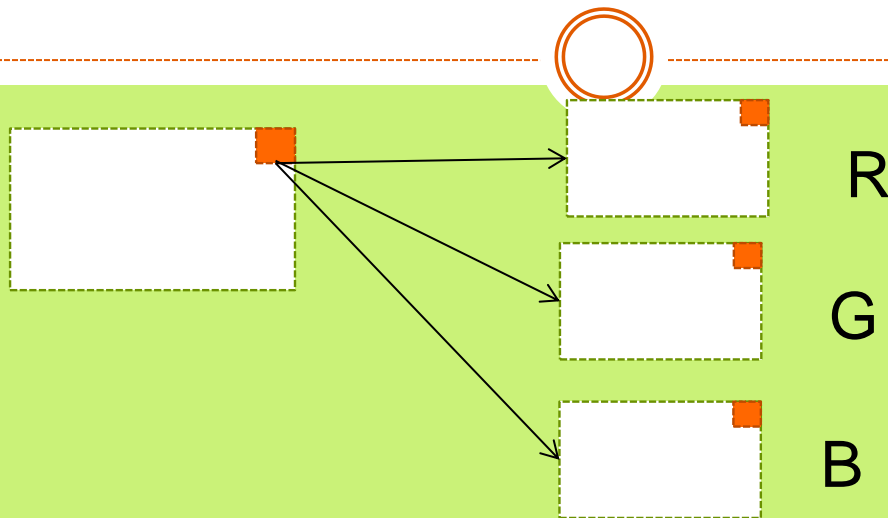
Is the same [0 , 1]when you divide each bit to 255



3. R,G, B image (true color image)



An M by N by 3 data array that define (red) , (green), and (blue) component for each pixel



- Suppose the pixel above is 8 bit What is the no. of color (intensities)

$$\text{no. of color (intensities)} = 2^{3*8} = 16,777,216$$

$$\text{no. of color (intensities) in R,G,B image} = 2^{3*\text{no.of bits}}$$



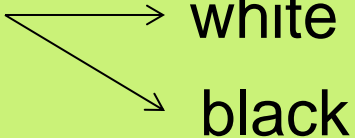
4. Binary Images

Each pixel take only one value of two values :on (white) and off (black)

1 = white

0= black

The pixel in binary image has only one bit

No. of colors = $2^1 = 2$ 



Graphics primitives

Line Drawing



- We should have two points , starting point P1 and ending point P2



$$P1=(x1,y1)$$

$$P2=(x2,y2)$$

$$m= \frac{(y2-y1)}{(x2-x1)}$$

Line equation : $y= mx +b$

m(slope)

b:intercept

Line drawing algorithms



- There are two algorithms:
- DDA(Digital Differential Analyzer)
- Bresenham's Algorithm

DDA(Digital Differential Analyzer)



- First step find the direction of the drawing , left to right or right to left
- Second step find the slop (m).



- According to the slope value:

Left to right	Right to left
If $ m \leq 1$ $\Delta x = 1$ $\Delta y = m$	If $ m \leq 1$ $\Delta x = -1$ $\Delta y = -m$
If $ m > 1$ (for positive m) $\Delta x = 1/m$ $\Delta y = 1$	If $ m > 1$ (for positive m) $\Delta x = -1/m$ $\Delta y = -1$
If $ m > 1$ (for negative m) $\Delta x = -1/m$ $\Delta y = -1$	If $ m > 1$ (for negative m) $\Delta x = 1/m$ $\Delta y = 1$

Example



- Given the points $p_1(0,0)$ and $p_2(8,6)$ find the line points that will be produced using DDA algorithm

Sol:

$P_1 \rightarrow P_2$ Left to right

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{6 - 0}{8 - 0} = 0.75$$

$$\Delta x = 1, \quad \Delta y = m = 0.75$$

Changes in $x=1$

Changes in $y = 0.75$

What is the value of y when $x = 5$?

$$y = 3.75$$

Starting point

x	y	R(y)
0	0	0
1	0.75	1
2	1.5	2
3	2.25	2
4	3	3
5	3.75	4
6	4.5	5
7	5.25	5
8	6	6

ending point

Example 2



- Suppose we want to draw a line starting at pixel (2,3) ending with (12,8) what are the values of variables x and y at each timestep, what are the pixels colored using DDA.

- Left to right
- Time step= 12-2=10

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{8 - 3}{12 - 2} = 0.5$$

$$\Delta x = 1, \Delta y = m = 0.5$$

t	x	y	R(x)	R(y)
0	2	3	2	3
1	3	3.5	3	4
2	4	4	4	4
3	5	4.5	5	5
4	6	5	6	5
5	7	5.5	7	6
6	8	6	8	6
7	9	6.5	9	7
8	10	7	10	7
9	11	7.5	11	8
10	12	8	12	8

Example 3



- Given the points $p_1(1,1)$ and $p_2(-1,7)$ find the line points that will be produced using DDA algorithm

Right to left

- $m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{7 - 1}{-1 - 1} = \frac{-6}{-2} = -3$
- $\Delta x = \frac{1}{m} = -0.3, \quad \Delta y = 1$

x	R(x)	y
1	1	1
0.7	1	2
0.4	0	3
0.1	0	4
-0.2	0	5
-0.5	-1	6
-0.8	-1	7

Example 4(quiz)



- Given the points $p_1(2,-5)$ and $p_2(-2,4)$ find the line points that will be produced using DDA algorithm.

Bresenham's Algorithm



- $|m| < 1$

For given points $(x_1, y_1)(x_2, y_2)$
 $x = x_1, y = y_1$
 $\Delta x = x_2 - x_1$
 $\Delta y = y_2 - y_1$

$\Delta y - \Delta x$ fixed

$P = 2 \Delta y - \Delta x$

While $x < x_2$

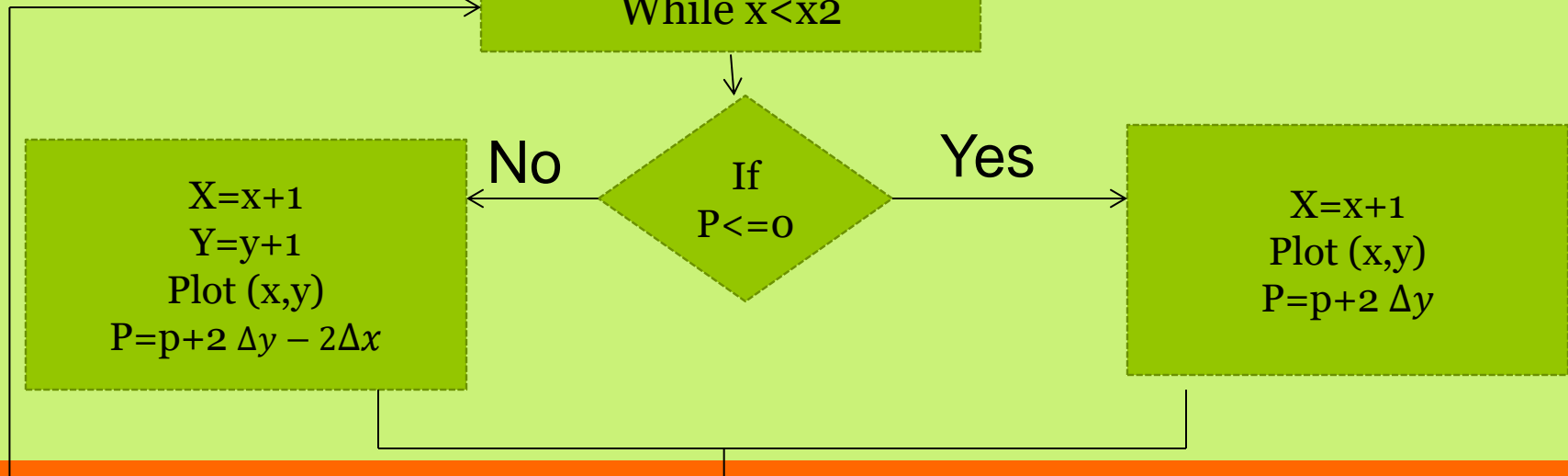
No

If
 $P \leq 0$

Yes

$X = x + 1$
 $Y = y + 1$
Plot (x, y)
 $P = p + 2 \Delta y - 2 \Delta x$

$X = x + 1$
Plot (x, y)
 $P = p + 2 \Delta y$



Example 1



- Generate (calculate) the points between the starting point $p_1(9,18)$ and ending point $p_2(14,22)$ using bresenham's algorithms

- $\Delta x = 5, \Delta y = 4$
- $P = 2 \Delta y - \Delta x = 3$

X	Y	
9	18	3
10	19	1
11	20	-1
12	20	7
13	21	5
14	22	

Example2



- Generate (calculate) the points between the starting point $p_1(0,0)$ and ending point $p_2(8,6)$ using bresenham's algorithms
- $\Delta x = 8, \Delta y = 6$
- $P = 2 \Delta y - \Delta x = 4$

x	y	P
0	0	4
1	1	0
2	2	-4
3	2	8
4	3	4
5	4	0
6	5	-4
7	5	8
8	6	

Note: P is repeated

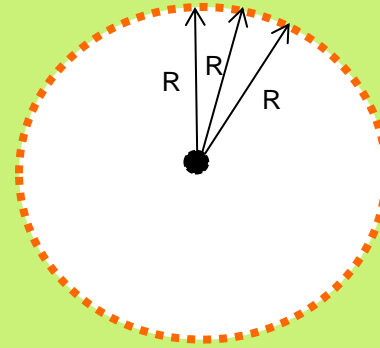


- Generate (calculate) the points between the starting point $p_1(0,0)$ and ending point $p_2(8,6)$ using bresenham's algorithms

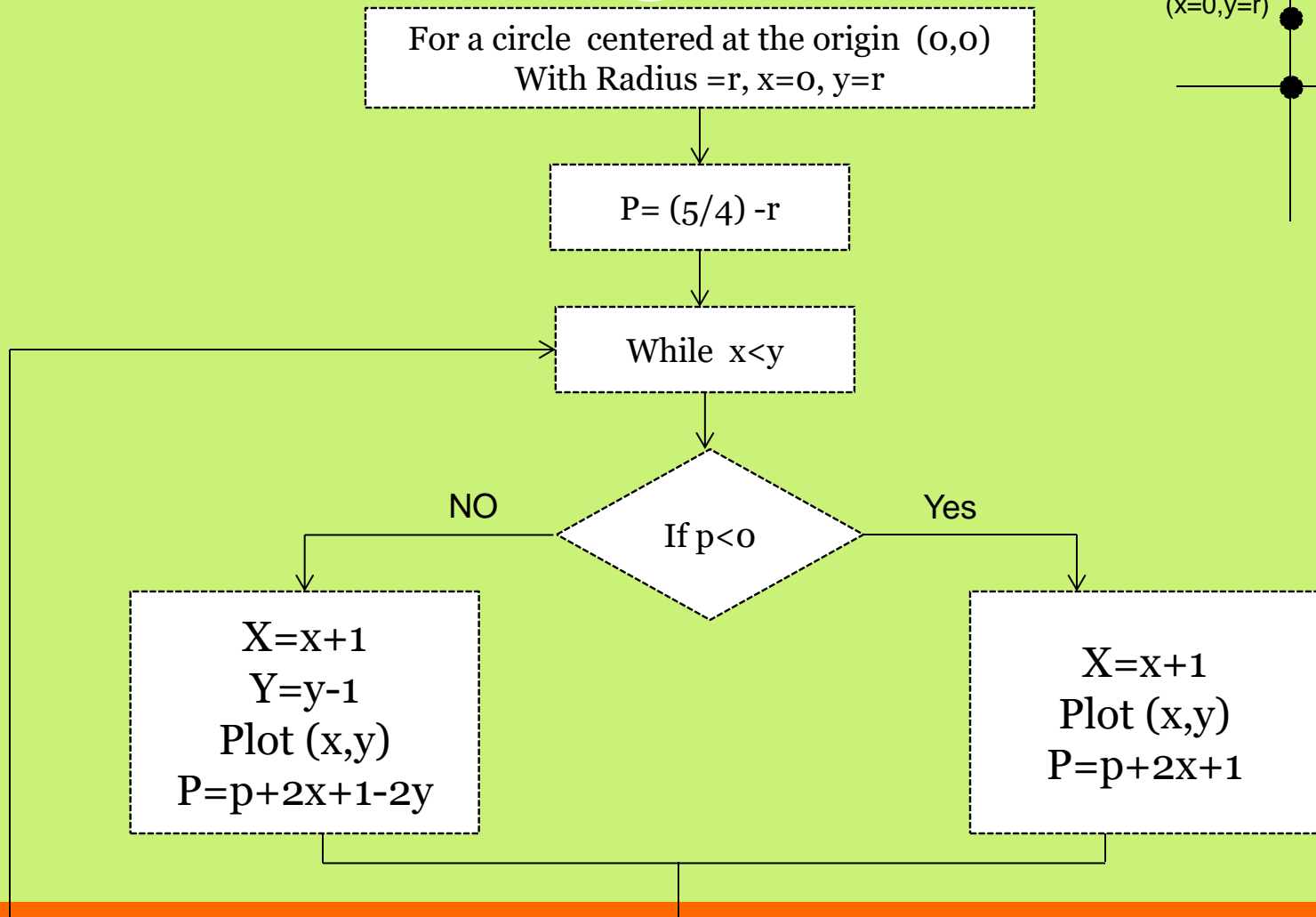
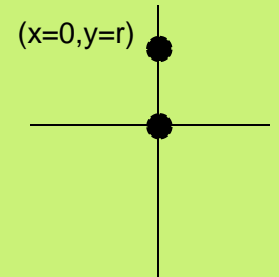
Circle drawing



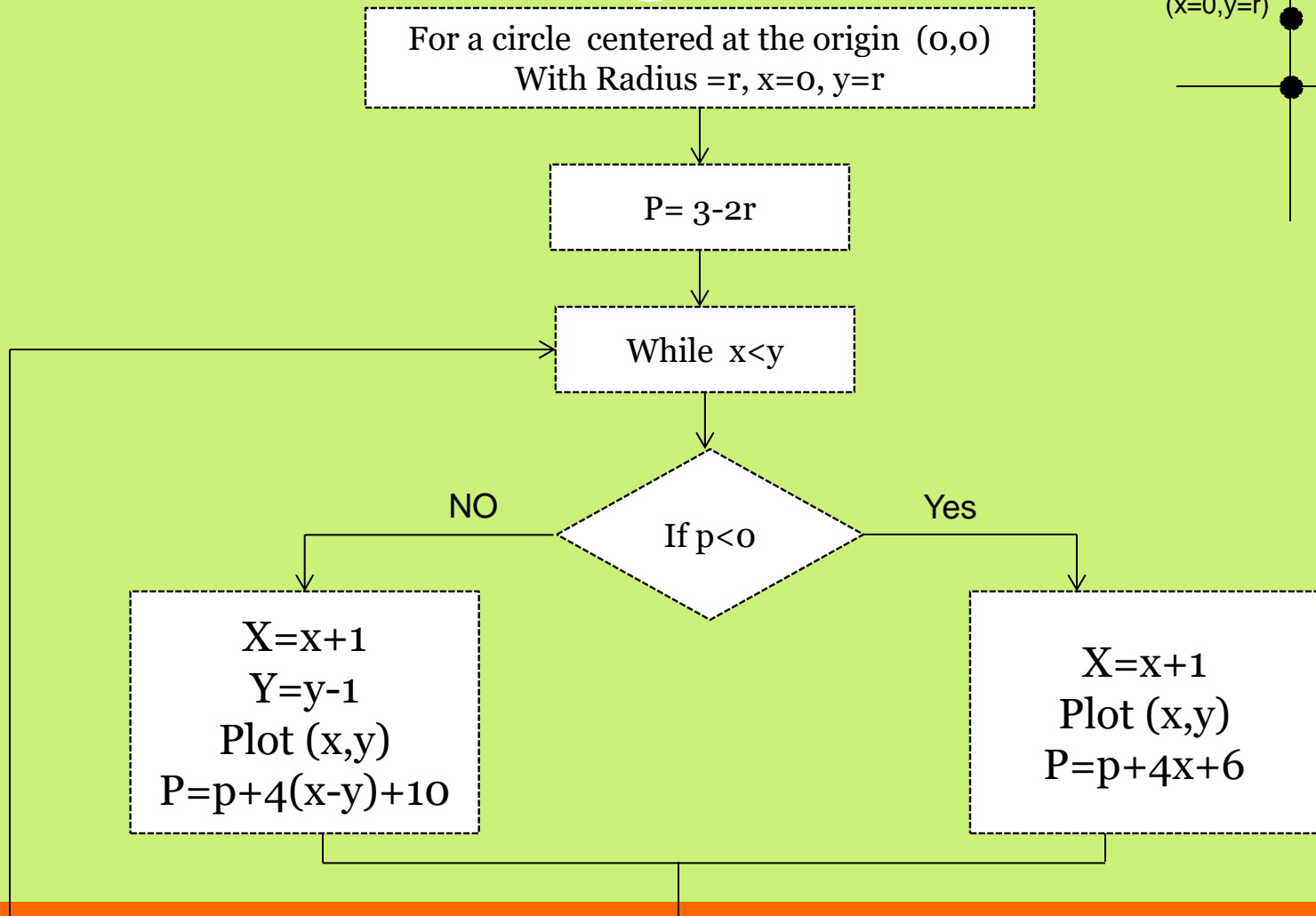
- We should know
 - Radius
 - Center coordinates
- Circle drawing Algorithms:
 1. Mid-point Algorithm
 2. Bresenham's algorithm



Mid-point Algorithm



Bresenham's algorithm



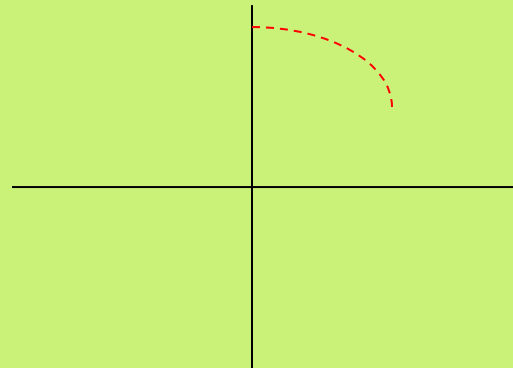
Example1



- Given a circle centered at the origin and radius $r=13$, what positions should be generated using Mid-point algorithm?

x	y	p
0	13	-11.75
1	13	-8.75
2	13	-3.75
3	13	3.25
4	12	-11.75
5	12	-0.75
6	12	12.25
7	11	5.25
8	10	2.25
9	9	X=y , not less than y

R=13
Y=r
X=0



Problems of using Mid-point and Bresenham's algorithms



- Both works when the circle centered at the origin $(0,0)$
- The points generated is octant of the circle not full circle.

But its enough to draw a circle

Example2

If the circle was not centered in at the origin

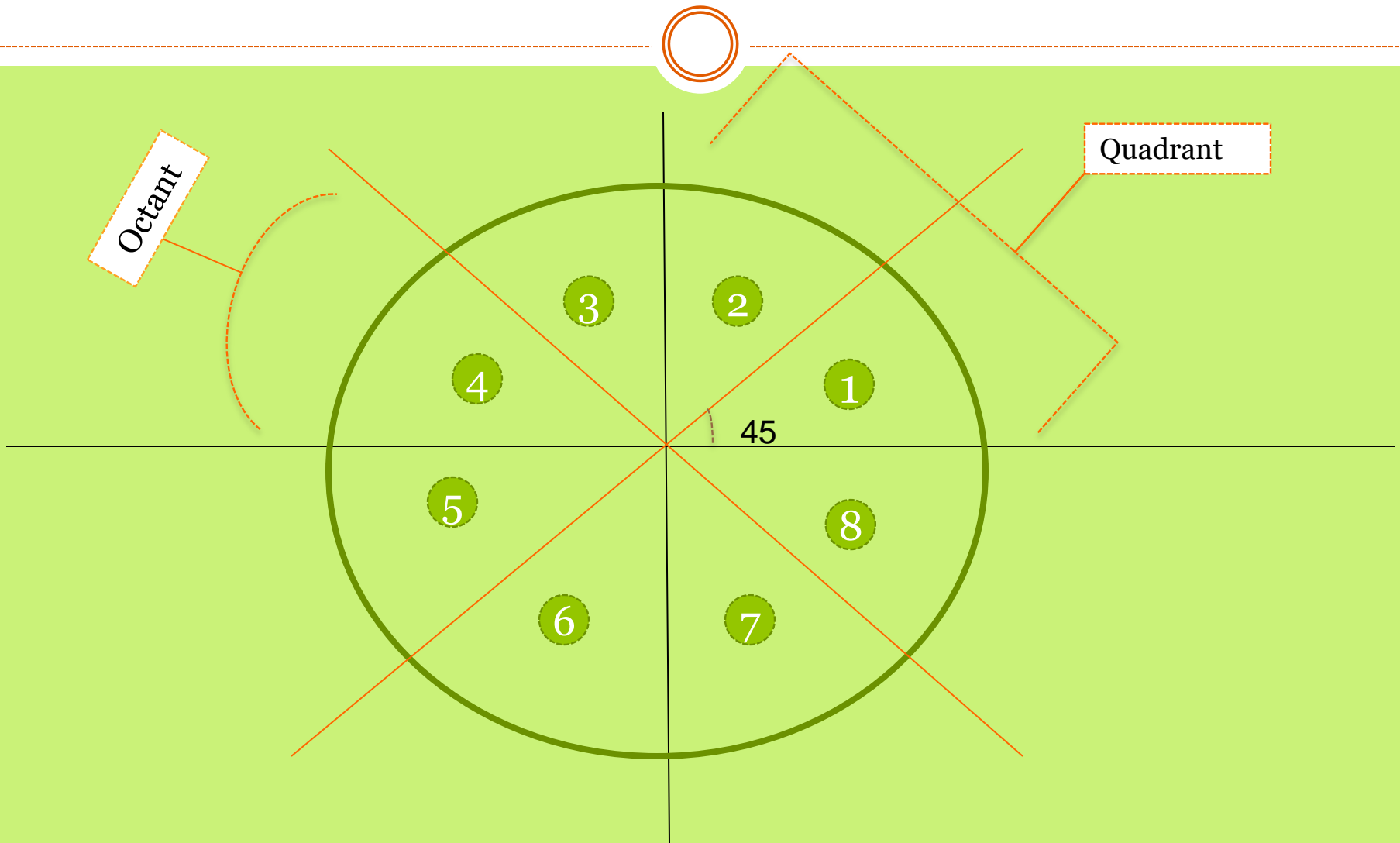
- Given a circle with center $(-1,2)$ and radius $r=10$, what positions should be generated using Bresenham's algorithm?

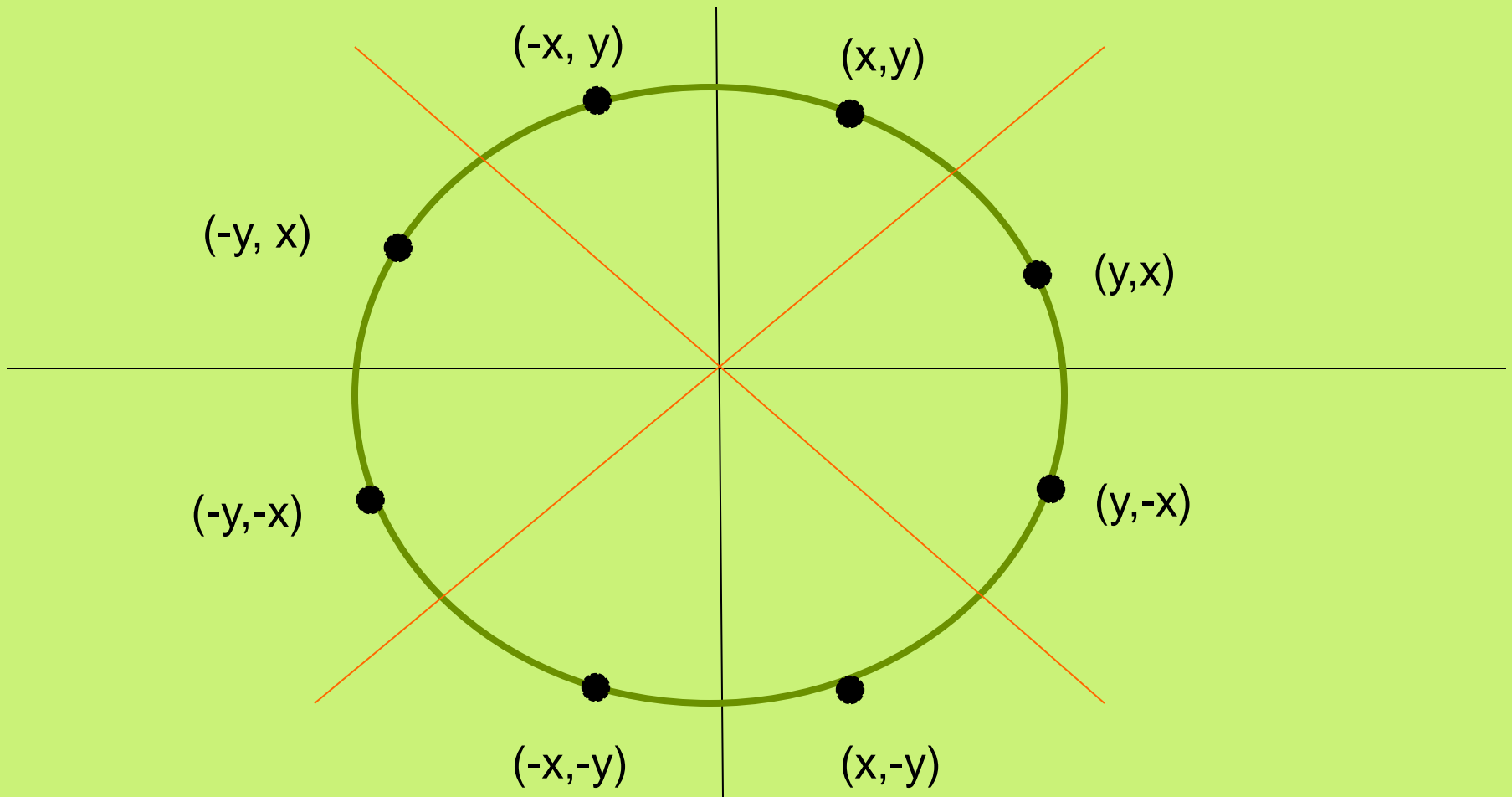
$$X_c, Y_c = (-1, 2)$$

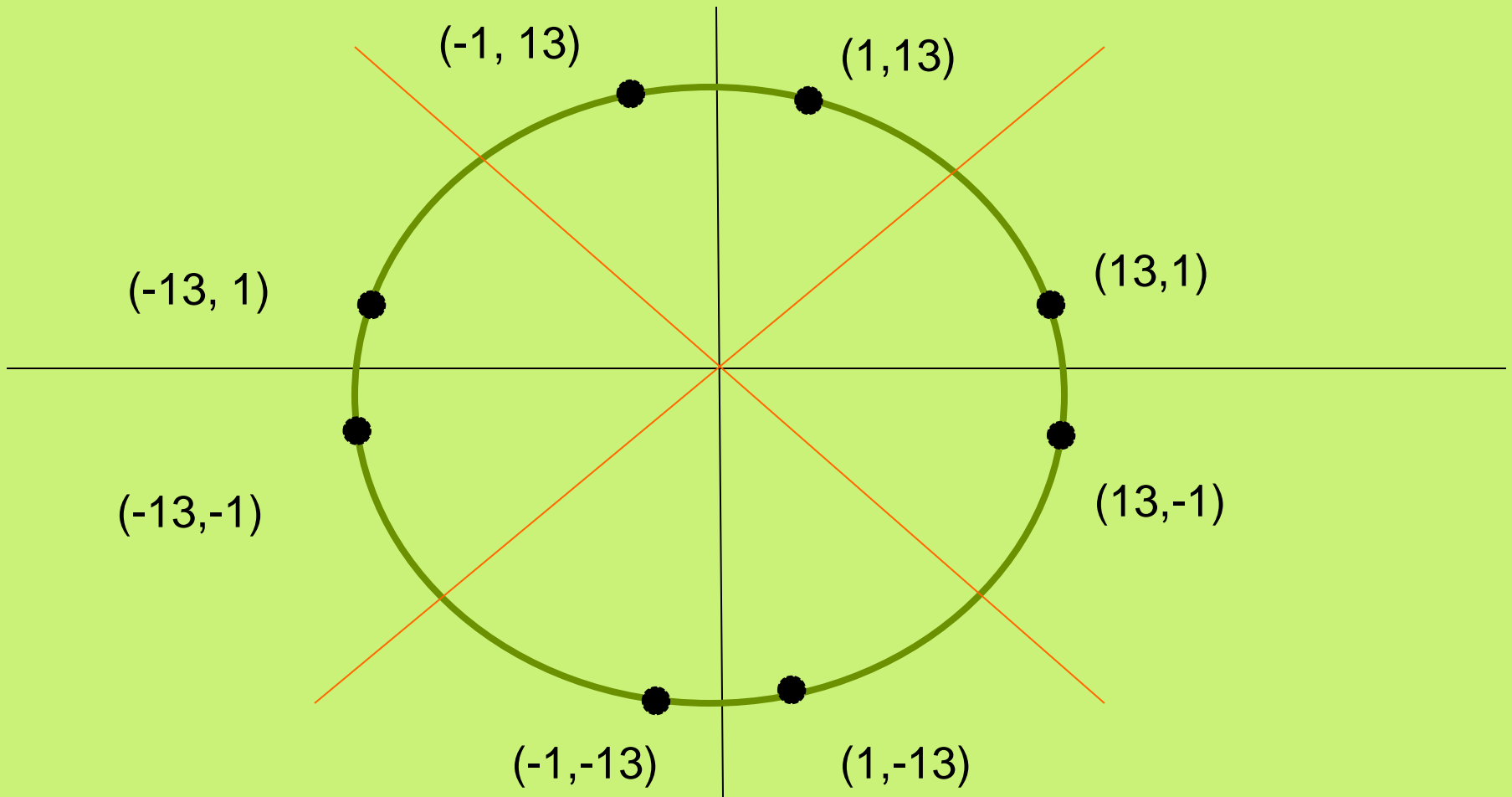
x	y	p	X+X _c	Y+Y _c
0	10	-17	-1	12
1	10	-7	0	12
2	10	7	1	12
3	9	-7	2	11
4	9	15	3	11
5	8	13	4	10
6	7	19	5	9
7	6	33	6	8

The problem will be solve as if the circle at the origin point then the exact center(X_c, Y_c) will be added to X and Y

Circle symmetry



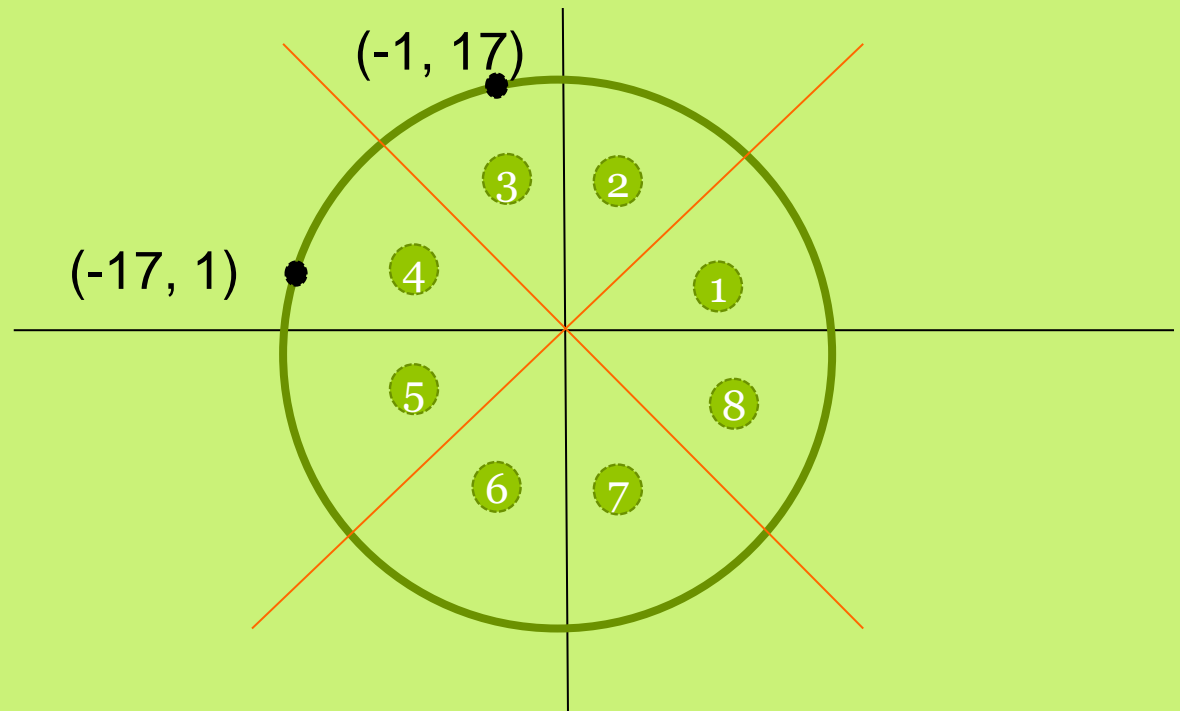






- In circle drawing the equivalent y-coordinate output point $(-17, 1)$ in the third octant is

- a. 1
- b. 17
- c. -17
- d. -1





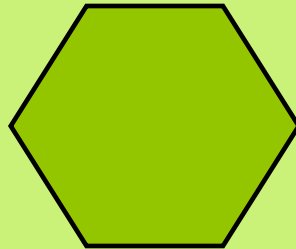
Polygon Filling

Polygon Filling

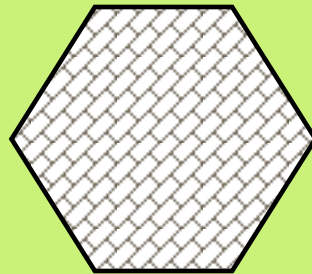


Types of filling

- **Solid-fill**



All the pixels inside the polygon's boundary are illuminated.



- **Pattern-fill**

the polygon is filled with an arbitrary predefined pattern.

Polygon Representation



- The polygon can be represented by listing its n vertices in an ordered list.
 - $P = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$.
- The polygon can be displayed by drawing a line between (x_1, y_1) , and (x_2, y_2) , then a line between (x_2, y_2) , and (x_3, y_3) , and so on until the end vertex. In order to close up the polygon, a line between (x_n, y_n) , and (x_1, y_1) must be drawn.
- One problem with this representation is that if we wish to translate the polygon, it is necessary to apply the translation transformation to each vertex in order to obtain the translated polygon.



- For objects described by many polygons with many vertices, this can be a time consuming process.
- One method for reducing the computational time is to represent the polygon by the (**absolute**) **location** of its first vertex, and represent subsequent vertices as **relative positions** from the previous vertex. This enables us to translate the polygon simply by changing the coordinates of the first vertex.

Inside-Outside Tests



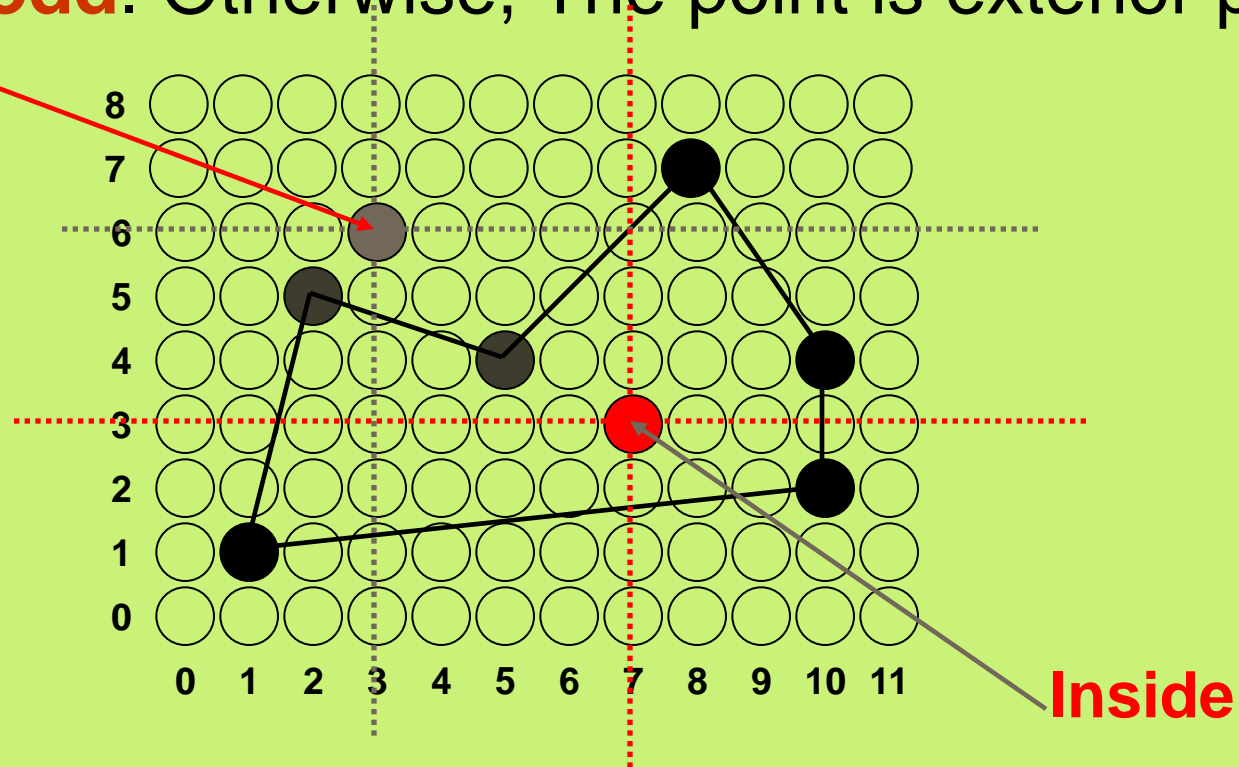
- when filling polygons we should decide whether a particular point is interior or exterior to a polygon.
- A rule called the **odd-parity** (or the **odd-even rule**) is applied to test whether a point is interior or not.
- To apply this rule, we conceptually draw a line starting from the particular point and extending to a distance point outside the coordinate extends of the object in any direction such that **no polygon vertex intersects with the line.**

Inside-Outside Tests



The point is considered to be **interior** if the number of intersections between the line and the polygon edges is **odd**. Otherwise, The point is exterior point.

Outside

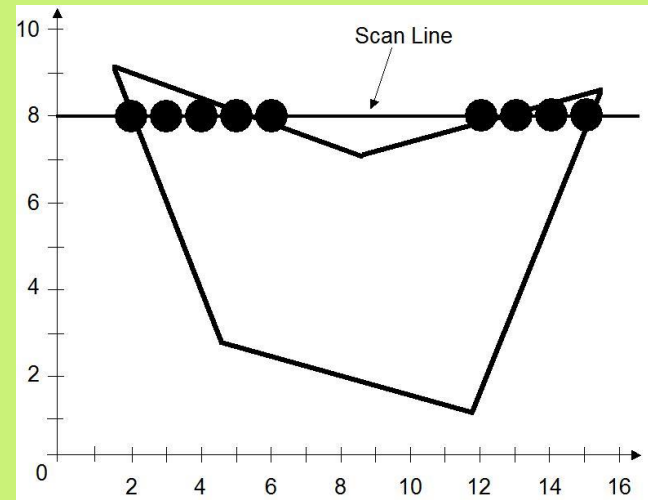


The Scan-Line Polygon Fill Algorithm



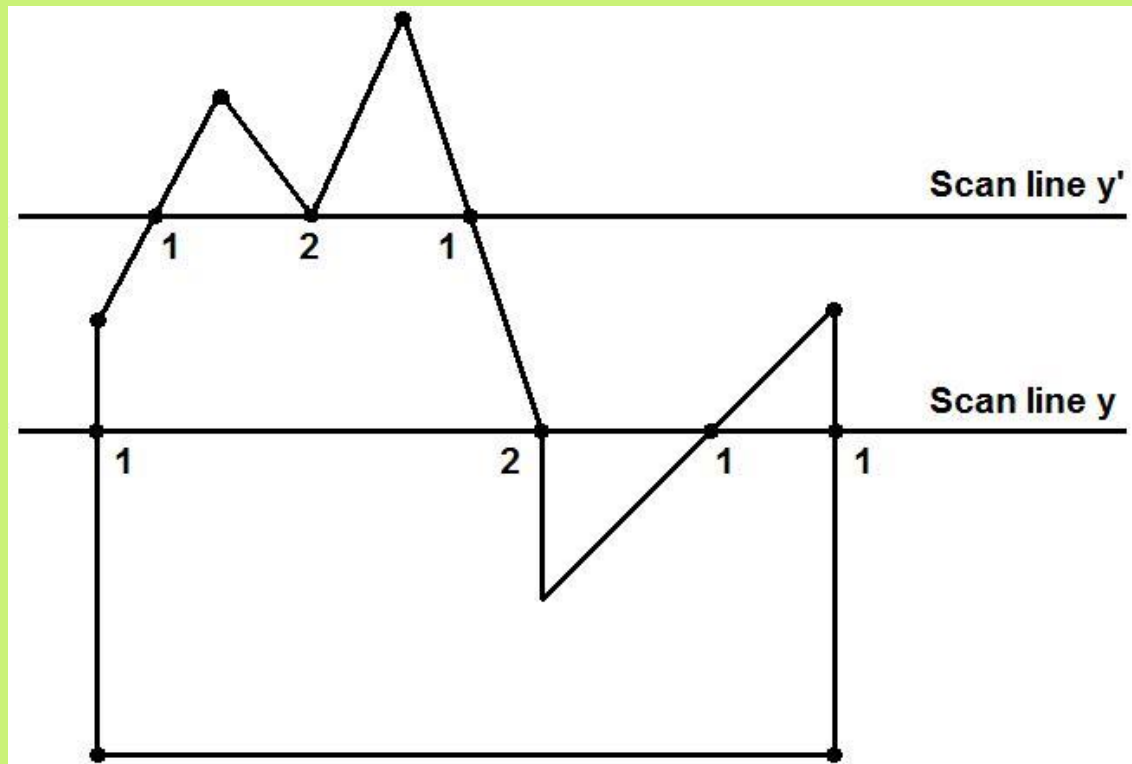
The scan-line polygon-filling algorithm involves

- the **horizontal scanning** of the polygon from its **lowermost** to its **topmost** vertex,
- identifying which edges intersect the scan-line,
- and finally drawing the interior horizontal lines with the specified fill color. process.



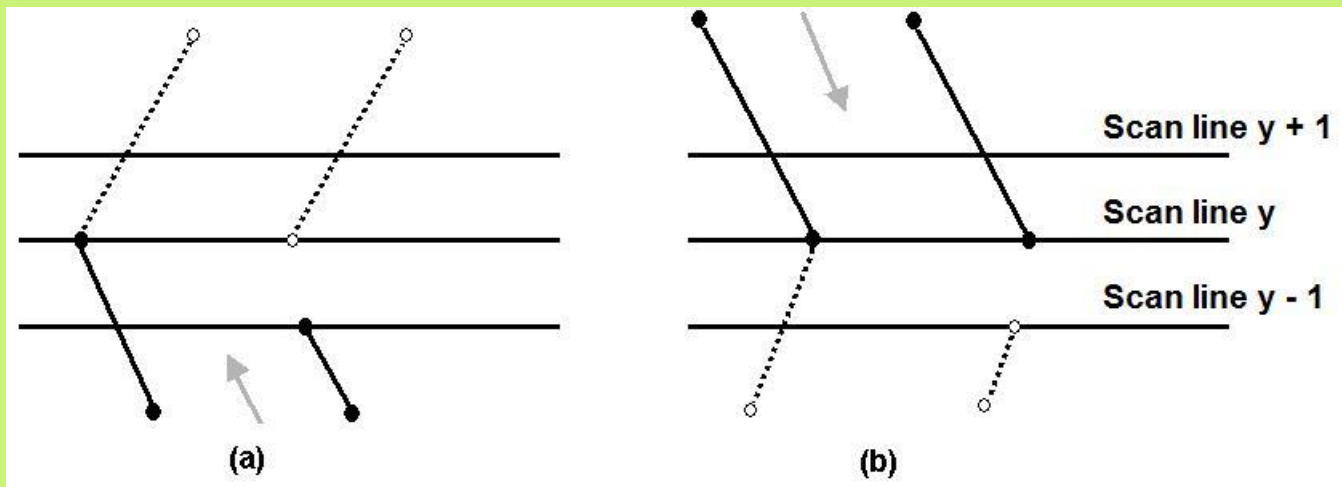
The Scan-Line Polygon Fill Algorithm

Dealing with vertices





- When the endpoint **y** coordinates of the two edges are **increasing**, the **y** value of the upper endpoint for the **current edge** is decreased by one (a)
- When the endpoint **y** values are **decreasing**, the **y** value of the **next edge** is decreased by one (b)



The Scan-Line Polygon Fill Algorithm

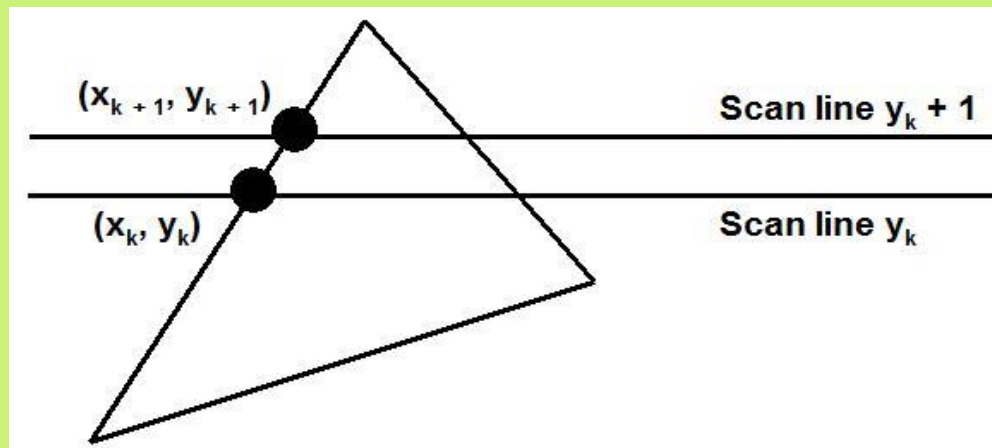


Determining Edge Intersections

$$m = (y_{k+1} - y_k) / (x_{k+1} - x_k)$$

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

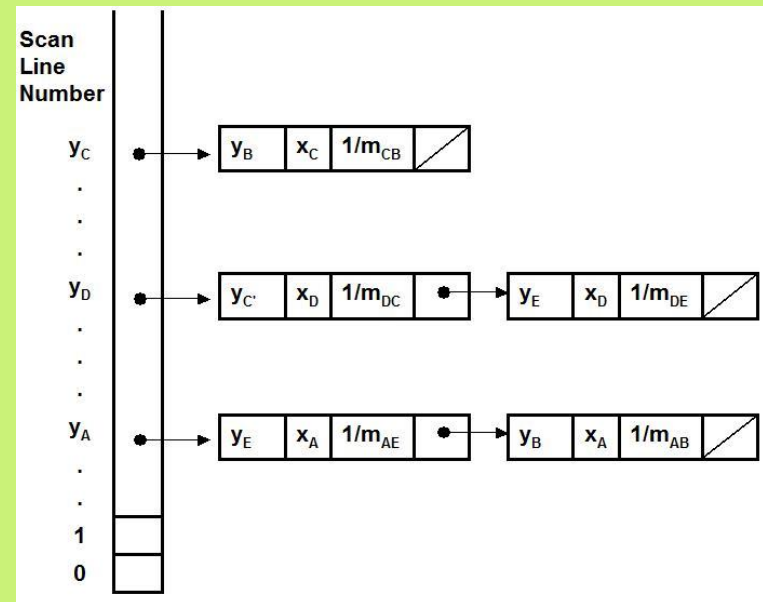
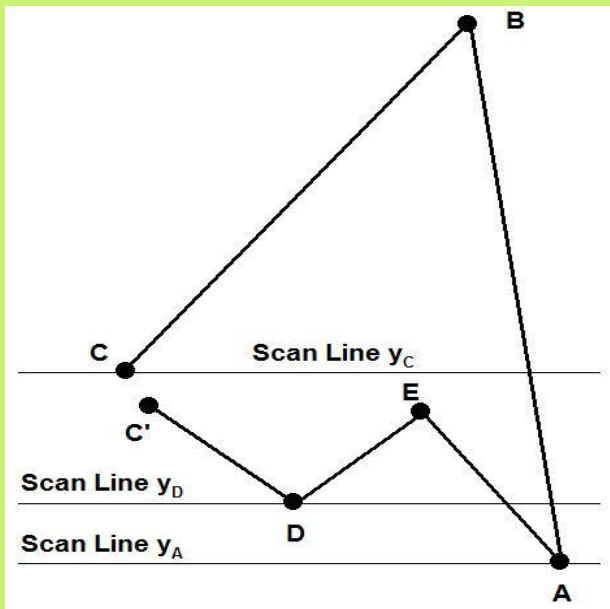
$$x_{k+1} = x_k + 1/m$$



The Scan-Line Polygon Fill Algorithm



- Each **entry** in the table for a particular scan line contains the **maximum y** value for that edge, the **x-intercept** value (at the lower vertex) for the edge, and the **inverse slope** of the edge.

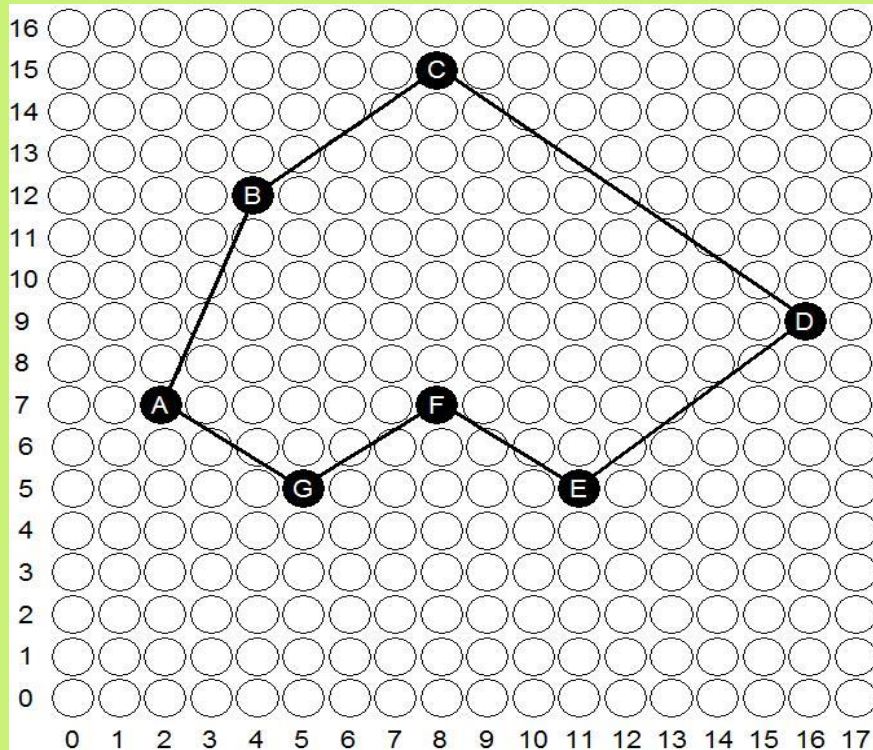


The Scan-Line Polygon Fill Algorithm



(Example) Polygon = {A, B, C, D, E, F, G}

Polygon = {(2, 7), (4, 12), (8, 15), (16, 9), (11, 5), (8, 7), (5, 5)}



Edge Table						
#	Edge		1/m	ymin	x	y _{max}
0	A (2, 7)	B (4, 12)	2/5	7	2	12
1	B (4, 12)	C (8, 15)	4/3	12	4	15
2	C (8, 15)	D (16, 9)	-8/6	9	16	15
3	D (16, 9)	E (11, 5)	5/4	5	11	9
4	E (11, 5)	F (8, 7)	-3/2	5	11	7
5	F (8, 7)	G (5, 5)	3/2	5	5	7
6	G (5, 5)	A (2, 7)	-3/2	5	5	7

The Scan-Line Polygon Fill Algorithm

(Example)



<p>If $y_P < y_C < y_N$ Then y_C is decreased by one. The new edges become $(x_P, y_P) \rightarrow (x'_C, y_C - 1)$ and $(x_C, y_C) \rightarrow (x_N, y_N)$</p>	<p>If $y_P > y_C > y_N$ Then y_C is decreased by one. The new edges become $(x_P, y_P) \rightarrow (x_C, y_C)$ and $(x'_C, y_C - 1) \rightarrow (x_N, y_N)$</p>
<p>$m = (y_P - y_C) / (x_P - x_C)$ $x'_C = x_P + (1/m)(y_C - 1 - y_P)$</p>	<p>$m = (y_N - y_C) / (x_N - x_C)$ $x'_C = x_N + (1/m)(y_C - 1 - y_N)$</p>

The Scan-Line Polygon Fill Algorithm

(Example)



Previous Vertex	Current Vertex	Next Vertex	$y_P ? y_C ? y_N$	Current Vertex Type	Action
G (5, 5)	A (2, 7)	B (4, 12)	$y_P < y_C < y_N$	Not local extremum	Split A
A (2, 7)	B (4, 12)	C (8, 15)	$y_P < y_C < y_N$	Not local extremum	Split B
B (4, 12)	C (8, 15)	D (16, 9)	$y_P < y_C > y_N$	Local Maximum	None
C (8, 15)	D (16, 9)	E (11, 5)	$y_P > y_C > y_N$	Not local extremum	Split D
D (16, 9)	E (11, 5)	F (8, 7)	$y_P > y_C < y_N$	Local Minimum	None
E (11, 5)	F (8, 7)	G (5, 5)	$y_P < y_C > y_N$	Local Maximum	None
F (8, 7)	G (5, 5)	A (2, 7)	$y_P > y_C < y_N$	Local Minimum	None

- **Vertex A** should be split into two vertices **A'** ($x_{A'}$, 6) and **A**(2, 7)

$$m = (5 - 7) / (5 - 2) = -2/3$$

$$x'_{A'} = 5 + (-3/2)(7 - 1 - 5) = 7/2 = 3.5 \cong 4$$

The vertex **A** is split to **A'** (4, 6) and **A**(2, 7)

The Scan-Line Polygon Fill Algorithm

(Example)



Previous Vertex	Current Vertex	Next Vertex	$y_P ? y_C ? y_N$	Current Vertex Type	Action
G (5, 5)	A (2, 7)	B (4, 12)	$y_P < y_C < y_N$	Not local extremum	Split A
A (2, 7)	B (4, 12)	C (8, 15)	$y_P < y_C < y_N$	Not local extremum	Split B
B (4, 12)	C (8, 15)	D (16, 9)	$y_P < y_C > y_N$	Local Maximum	None
C (8, 15)	D (16, 9)	E (11, 5)	$y_P > y_C > y_N$	Not local extremum	Split D
D (16, 9)	E (11, 5)	F (8, 7)	$y_P > y_C < y_N$	Local Minimum	None
E (11, 5)	F (8, 7)	G (5, 5)	$y_P < y_C > y_N$	Local Maximum	None
F (8, 7)	G (5, 5)	A (2, 7)	$y_P > y_C < y_N$	Local Minimum	None

- **Vertex B** should be split into two vertices **B'** ($x_{B'}$, 11) and **B**(4, 12)

$$m = (7 - 12) / (2 - 4) = 5/2$$

$$x'_{A'} = 2 + (2/5)(12 - 1 - 7) = 18/5 = 3.6 \cong 4$$

The vertex **B** is split to **B'** (4, 11) and **B**(4, 12)

The Scan-Line Polygon Fill Algorithm

(Example)



Previous Vertex	Current Vertex	Next Vertex	$y_P ? y_C ? y_N$	Current Vertex Type	Action
G (5, 5)	A (2, 7)	B (4, 12)	$y_P < y_C < y_N$	Not local extremum	Split A
A (2, 7)	B (4, 12)	C (8, 15)	$y_P < y_C < y_N$	Not local extremum	Split B
B (4, 12)	C (8, 15)	D (16, 9)	$y_P < y_C > y_N$	Local Maximum	None
C (8, 15)	D (16, 9)	E (11, 5)	$y_P > y_C > y_N$	Not local extremum	Split D
D (16, 9)	E (11, 5)	F (8, 7)	$y_P > y_C < y_N$	Local Minimum	None
E (11, 5)	F (8, 7)	G (5, 5)	$y_P < y_C > y_N$	Local Maximum	None
F (8, 7)	G (5, 5)	A (2, 7)	$y_P > y_C < y_N$	Local Minimum	None

- **Vertex D** should be split into two vertices **D(16, 9)** and **D' ($x_{D'}, 8$)**

$$m = (5 - 9) / (11 - 16) = 4/5$$

$$x'_{D'} = 11 + (5/4)(9 - 1 - 5) = 59/4 = 14.75 \cong 15$$

The vertex **D** is split to **D(16, 9)** and **D' (15, 8)**

The Scan-Line Polygon Fill Algorithm

(Example)



Modified Edge Table						
#	Edge		1/m	y _{min}	x	y _{max}
0	A (2, 7)	B' (4, 11)	2/5	7	2	11
1	B (4, 12)	C (8, 15)	4/3	12	4	15
2	C (8, 15)	D (16, 9)	- 8/6	9	16	15
3	D' (15, 8)	E (11, 5)	5/4	5	11	8
4	E (11, 5)	F (8, 7)	- 3/2	5	11	7
5	F (8, 7)	G (5, 5)	3/2	5	5	7
6	G (5, 5)	A' (4, 6)	- 3/2	5	5	6

Activation Table											
y	5	6	7	8	9	10	11	12	13	14	15
Activated Edge #s	3, 4, 5, 6		0		2			1			

The Scan-Line Polygon Fill Algorithm

(Example)



Edge number 0

#	Edge	1/m	Y _{min}	x	Y _{max}
0	A (2, 7) B' (4, 11)	$2/5 = 0.4$	7	2	11

Scan line	x-intersection
y = 7	2
y = 8	$2 + 0.4 = 2.4 \sim 2$
y = 9	$2.4 + 0.4 = 2.8 \sim 3$
y = 10	$2.8 + 0.4 = 3.2 \sim 3$
y = 11	4

Edge number 1

#	Edge	1/m	Y _{min}	x	Y _{max}
1	B (4, 12) C (8, 15)	$4/3 = 1.3$	12	4	15

Scan line	x-intersection
y = 12	4
y = 13	$4 + 1.3 = 4.3 \sim 4$
y = 14	$4.3 + 1.3 = 5.6 \sim 6$
y = 15	8

The Scan-Line Polygon Fill Algorithm

(Example)



Edge number 2

#	Edge	1/m	y _{min}	x	y _{max}
2	C (8,15) D (16, 9)	$-8/6 = -1.3$	9	16	15

Scan line	x-intersection
y = 9	16
y = 10	$16 - 1.3 = 14.7 \sim 15$
y = 11	$14.7 - 1.3 = 13.4 \sim 13$
y = 12	$13.4 - 1.3 = 12.1 \sim 12$
y = 13	$12.1 - 1.3 = 10.8 \sim 11$
y = 14	$10.8 - 1.3 = 9.5 \sim 10$
y = 15	8

Edge number 3

#	Edge	1/m	y _{min}	x	y _{max}
3	D' (15, 8) E (11, 5)	$5/4 = 1.25$	5	11	8

Scan line	x-intersection
y = 5	11
y = 6	$11 + 1.25 = 12.25 \sim 12$
y = 7	$12.25 + 1.25 = 13.5 \sim 14$
y = 8	15

The Scan-Line Polygon Fill Algorithm

(Example)



Edge number 4

#	Edge		1/m	y _{min}	x	y _{max}
4	E (11, 5)	F (8, 7)	$-3/2 = -1.5$	5	11	7

Scan line	x-intersection
y = 5	11
y = 6	$11 - 1.5 = 9.5 \sim 10$
y = 7	8

Edge number 5

#	Edge		1/m	y _{min}	x	y _{max}
5	F (8, 7)	G (5, 5)	$3/2 = 1.5$	5	5	7

Scan line	x-intersection
y = 5	5
y = 6	$5 + 1.5 = 6.5 \sim 7$
y = 7	8

Edge number 6

#	Edge		1/m	y _{min}	x	y _{max}
6	G (5, 5)	A' (4, 6)	$-3/2 = -1.5$	5	5	6

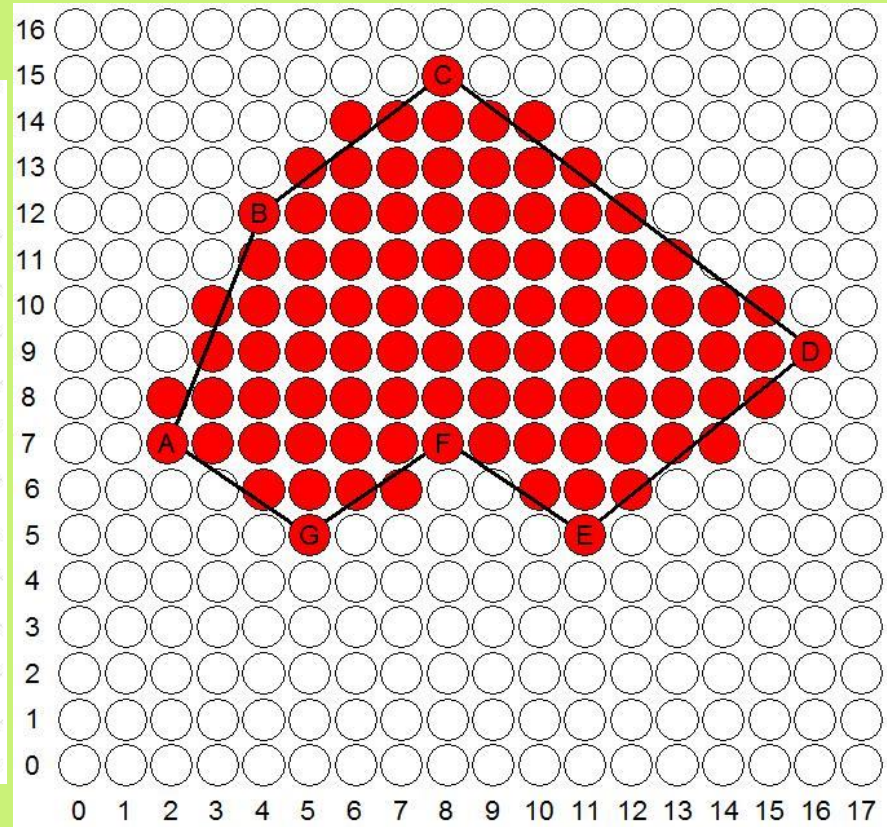
Scan line	x-intersection
y = 5	5
y = 6	4

The Scan-Line Polygon Fill Algorithm

(Example)



Scan line	x-intersections						x-intersections pair Ascending order	
	Edge#							
	0	1	2	3	4	5		6
5				11	11	5	5	(5, 5), (11, 11)
6				12	10	7	4	(4, 7), (10, 12)
7	2			14	8	8		(2, 8), (8, 14)
8	2			15				(2, 15)
9	3		16					(3, 16)
10	3		15					(3, 15)
11	4		13					(4, 13)
12		4	12					(4, 12)
13		4	11					(4, 11)
14		6	10					(6, 10)
15		8	8					(8, 8)



Boundary Fill Algorithm



- Another approach to area filling is to start at a point inside a region and paint the interior outward toward the boundary.
- If the boundary is specified in a single color, the fill algorithm processed outward pixel by pixel until the boundary color is encountered.
- A boundary-fill procedure accepts as input the coordinate of the interior **point (x, y)**, a **fill color**, and a **boundary color**.



```
void BoundaryFill4(int x, int y,  
                  color newcolor, color edgecolor)  
{  
    int current;  
    current = ReadPixel(x, y);  
    if(current != edgecolor && current != newcolor)  
    {  
        BoundaryFill4(x+1, y, newcolor, edgecolor);  
        BoundaryFill4(x-1, y, newcolor, edgecolor);  
        BoundaryFill4(x, y+1, newcolor, edgecolor);  
        BoundaryFill4(x, y-1, newcolor, edgecolor);  
    }  
}
```

Boundary Fill Algorithm

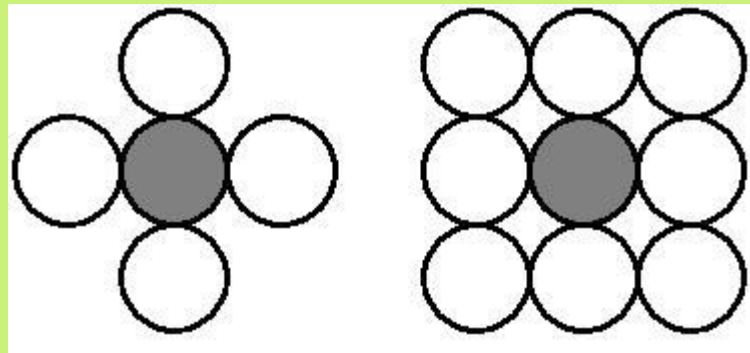


- The following steps illustrate the idea of the **recursive** boundary-fill algorithm:
- **1.** Start from an interior point.
- **2.** If the current pixel is **not already** filled and if it is not an edge point, then set the pixel with the fill color, and store its neighboring pixels (**4** or **8-connected**) in the stack for processing. Store only neighboring pixel that is **not already** filled and is not an edge point.
- **3.** Select the next pixel from the stack, and continue with step **2**.

Boundary Fill Algorithm

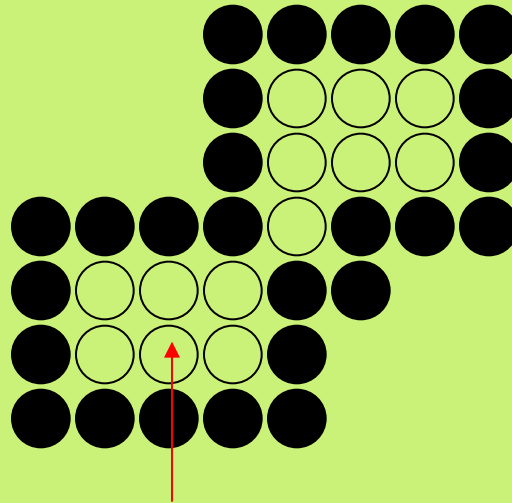


The order of pixels that should be added to stack using **4-connected** is above, below, left, and right. For **8-connected** is above, below, left, right, above-left, above-right, below-left, and below-right.



Boundary Fill Algorithm

4-connected (Example)

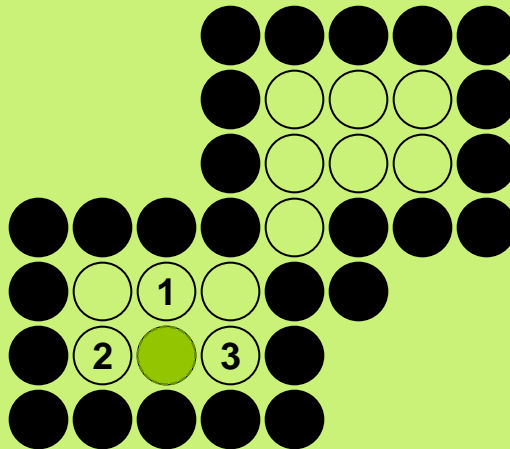


Start Position



Boundary Fill Algorithm

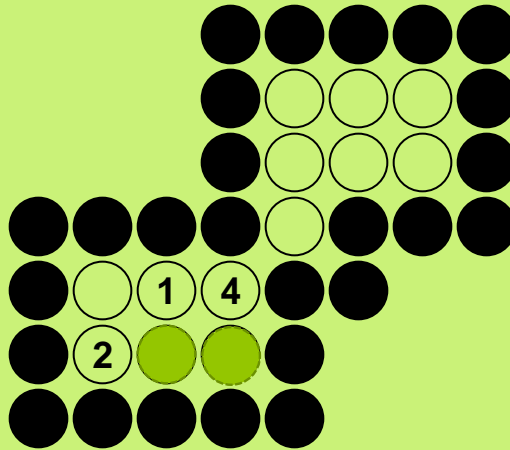
4-connected (Example)



3
2
1

Boundary Fill Algorithm

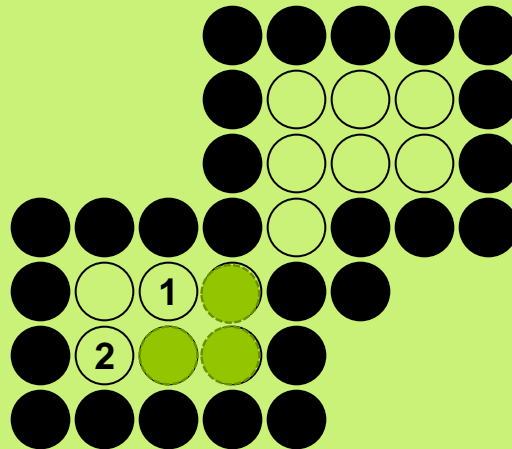
4-connected (Example)



4
2
1

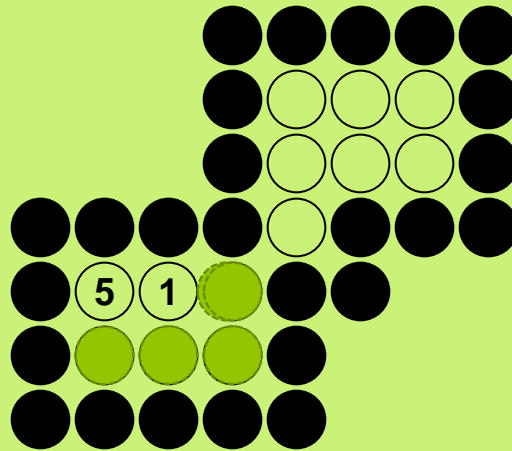
Boundary Fill Algorithm

4-connected (Example)



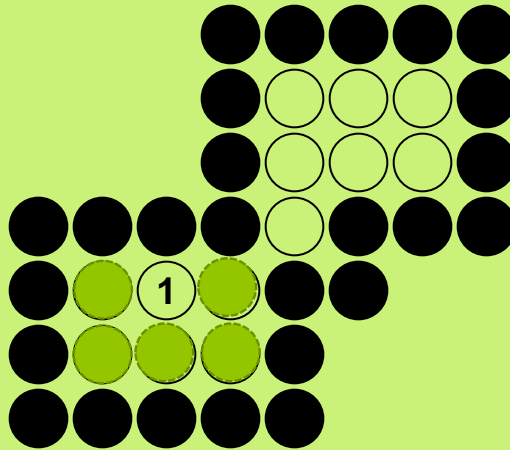
Boundary Fill Algorithm

4-connected (Example)



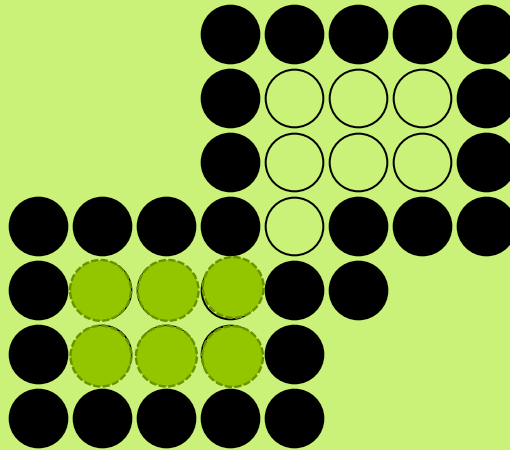
Boundary Fill Algorithm

4-connected (Example)



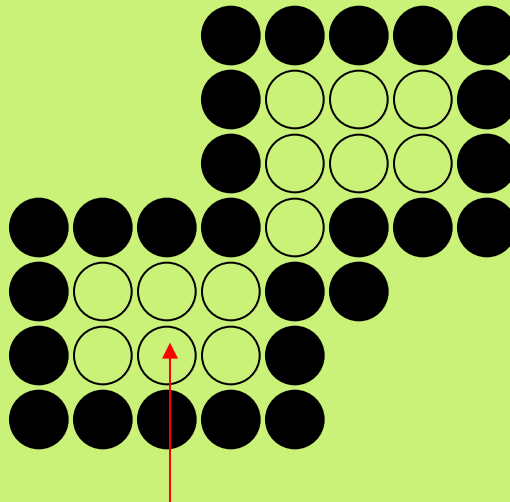
Boundary Fill Algorithm

4-connected (Example)



Boundary Fill Algorithm

8-connected (Example)

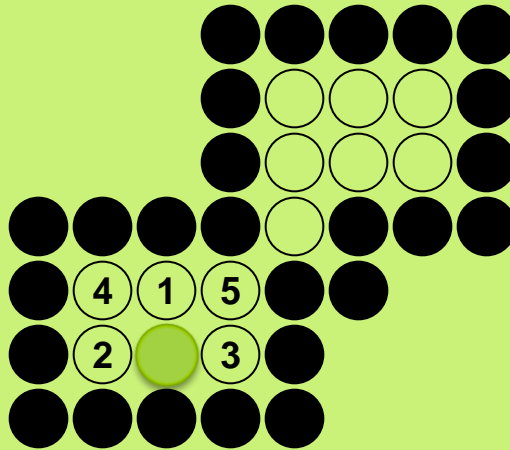


Start Position



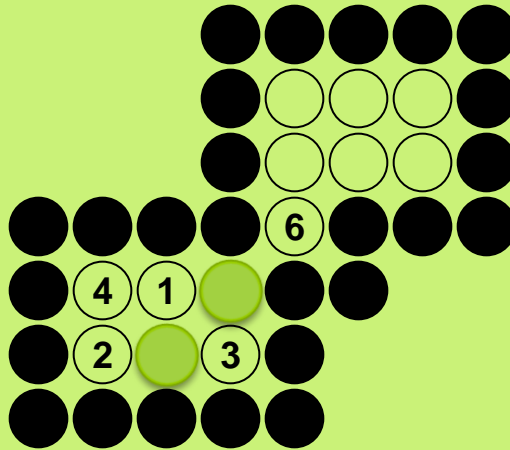
Boundary Fill Algorithm

8-connected (Example)



Boundary Fill Algorithm

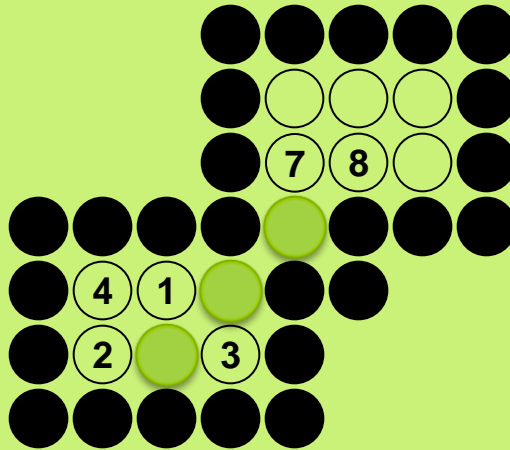
8-connected (Example)



6
4
3
2
1

Boundary Fill Algorithm

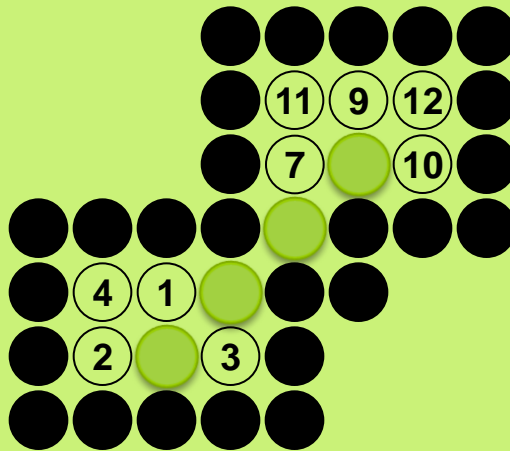
8-connected (Example)



8
7
4
3
2
1

Boundary Fill Algorithm

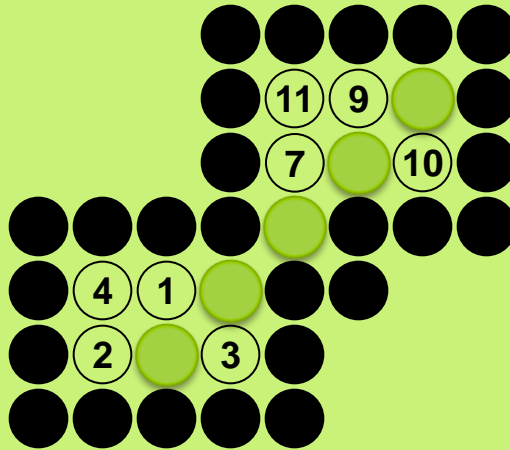
8-connected (Example)



12
11
10
9
7
4
3
2
1

Boundary Fill Algorithm

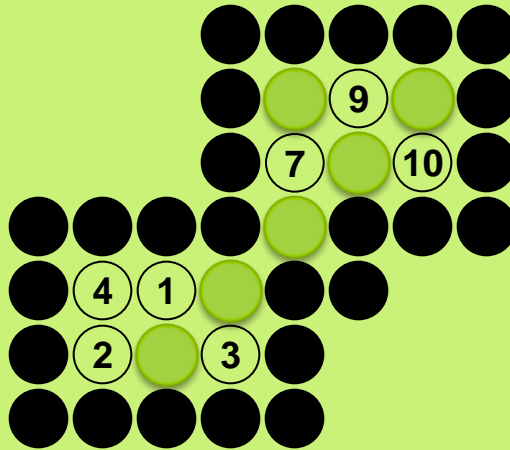
8-connected (Example)



11
10
9
7
4
3
2
1

Boundary Fill Algorithm

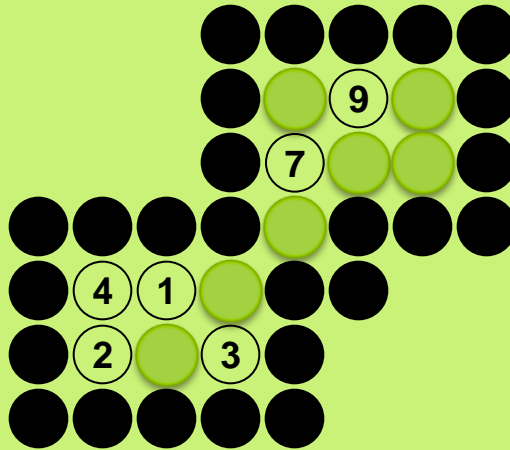
8-connected (Example)



10
9
7
4
3
2
1

Boundary Fill Algorithm

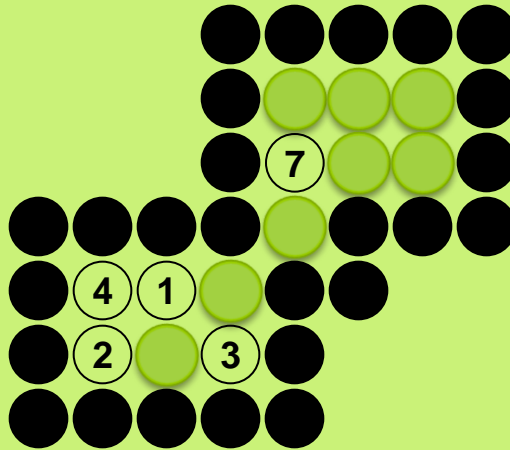
8-connected (Example)



9
7
4
3
2
1

Boundary Fill Algorithm

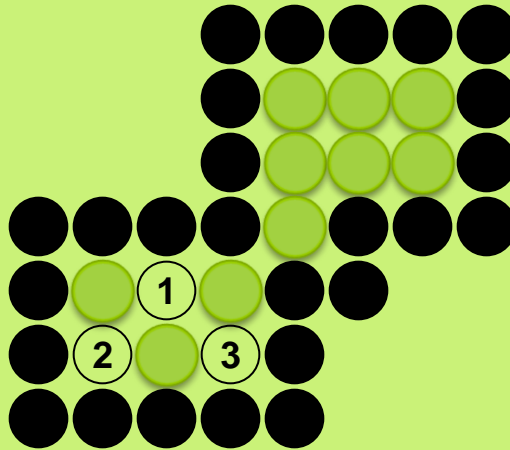
8-connected (Example)



7
4
3
2
1

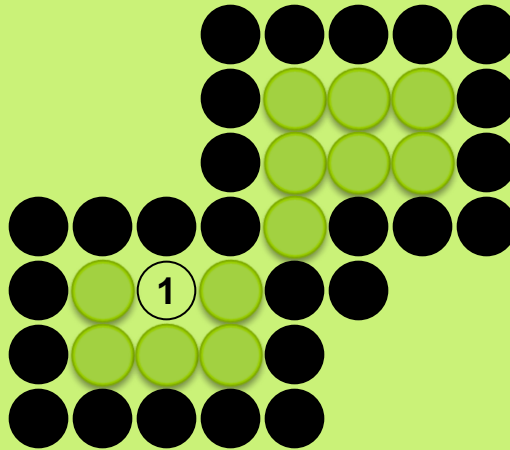
Boundary Fill Algorithm

8-connected (Example)



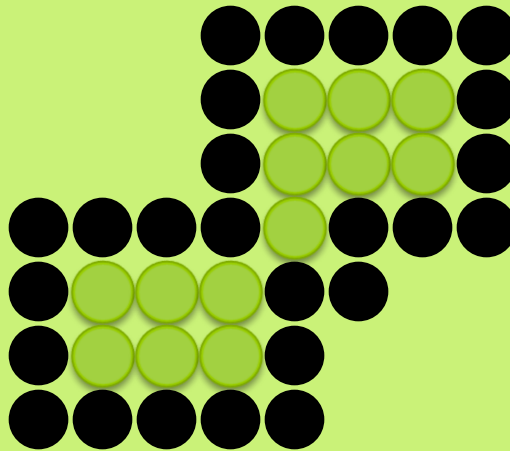
Boundary Fill Algorithm

8-connected (Example)



Boundary Fill Algorithm

8-connected (Example)



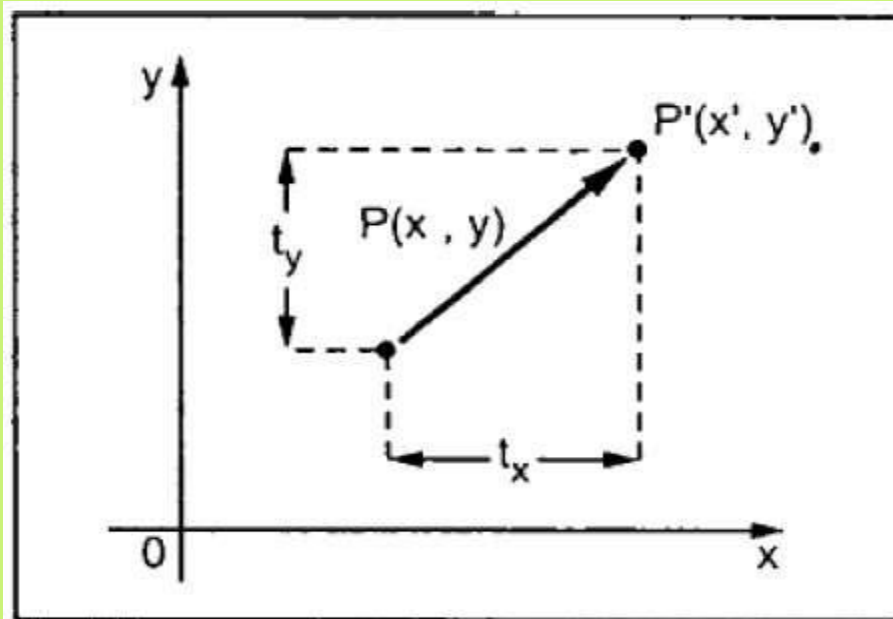


2D Geometric Transformation

2 D Translation



- A translation moves an object to a different position on the screen. You can translate a point in 2D by adding translation coordinate (t_x, t_y) to the original coordinate X, Y to get the new coordinate X', Y'





- From the above figure, you can write that –
- $X' = X + t_x$
- $Y' = Y + t_y$
- The pair (t_x, t_y) is called the translation vector or shift vector. The above equations can also be represented using the column vectors.

- $$P = \begin{bmatrix} X \\ Y \end{bmatrix} \quad p' = \begin{bmatrix} X' \\ Y' \end{bmatrix} \quad T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

- We can write it as –
- **$P' = P + T$**

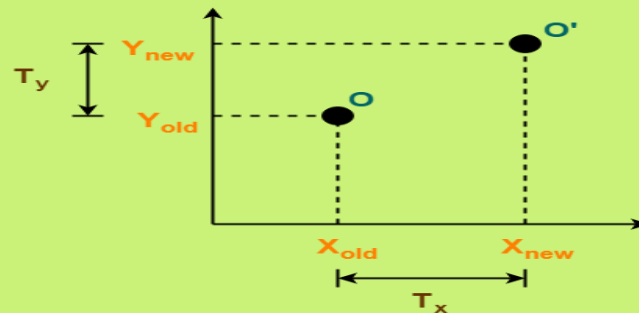
In an other word



- Let-
- Initial coordinates of the object $O = (X_{old}, Y_{old})$
- New coordinates of the object O after translation = (X_{new}, Y_{new})
- Translation vector or Shift vector = (t_x, t_y)

- Given a Translation vector (t_x, t_y) -

- t_x defines the distance the X_{old} coordinate has to be moved.
- t_y defines the distance the Y_{old} coordinate has to be moved.





- In Matrix form, the above translation equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \end{bmatrix} = \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}$$

Translation Matrix

- The homogeneous coordinates representation of (X, Y) is (X, Y, 1).
- Through this representation, all the transformations can be performed using matrix / vector multiplications.



- The above translation matrix may be represented as a 3 x 3 matrix as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ 1 \end{bmatrix}$$

Translation Matrix

(Homogeneous Coordinates Representation)

Example1



- Given a circle C with radius 10 and center coordinates $(1, 4)$. Apply the translation with distance 5 towards X axis and 1 towards Y axis. Obtain the new coordinates of C without changing its radius.



Solution-

Given-

Old center coordinates of C = $(X_{\text{old}}, Y_{\text{old}}) = (1, 4)$

Translation vector = $(T_x, T_y) = (5, 1)$

Let the new center coordinates of C = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the translation equations, we have-

$$X_{\text{new}} = X_{\text{old}} + T_x = 1 + 5 = 6$$

$$Y_{\text{new}} = Y_{\text{old}} + T_y = 4 + 1 = 5$$

Thus, New center coordinates of C = $(6, 5)$.



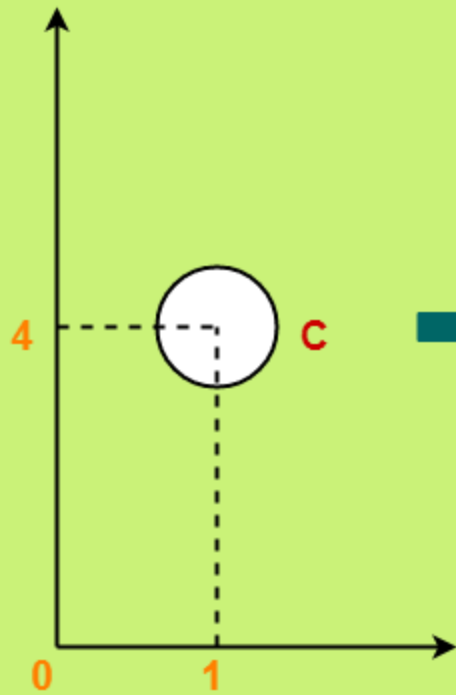
- **Alternatively,**
- In matrix form, the new center coordinates of C after translation may be obtained as-


$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \end{bmatrix} = \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}$$

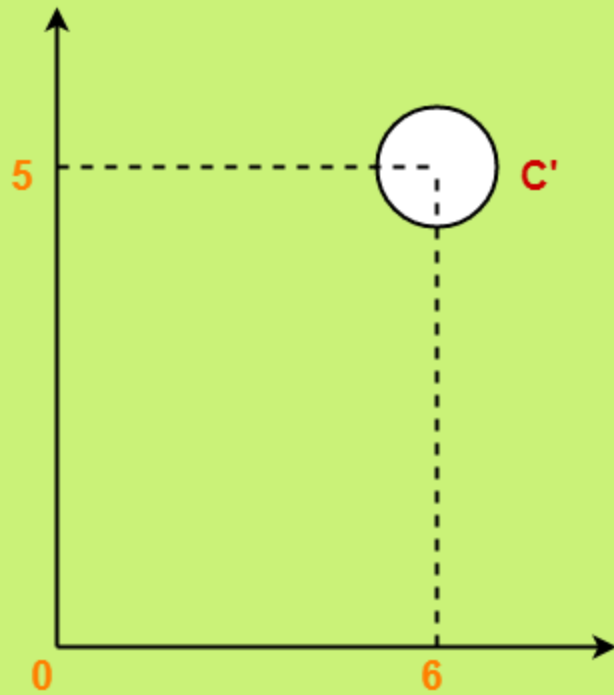
$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \end{bmatrix} + \begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \end{bmatrix}$$

- Thus, New center coordinates of C = (6, 5).



After Translation 



Example 2



- Given a square with coordinate points $A(0, 3)$, $B(3, 3)$, $C(3, 0)$, $D(0, 0)$. Apply the translation with distance 1 towards X axis and 1 towards Y axis. Obtain the new coordinates of the square.



- **Solution-**

Given-

Old coordinates of the square = A (0, 3), B(3, 3), C(3, 0), D(0, 0)

Translation vector = $(T_x, T_y) = (1, 1)$

For Coordinates A(0, 3)

Let the new coordinates of corner A = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the translation equations, we have-

$$X_{\text{new}} = X_{\text{old}} + T_x = 0 + 1 = 1$$

$$Y_{\text{new}} = Y_{\text{old}} + T_y = 3 + 1 = 4$$

Thus, New coordinates of corner A = (1, 4).



For Coordinates B(3, 3)

Let the new coordinates of corner B = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the translation equations, we have-

$$X_{\text{new}} = X_{\text{old}} + T_x = 3 + 1 = 4$$

$$Y_{\text{new}} = Y_{\text{old}} + T_y = 3 + 1 = 4$$

Thus, New coordinates of corner B = $(4, 4)$.





For Coordinates C(3, 0)

Let the new coordinates of corner C = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the translation equations, we have-

$$X_{\text{new}} = X_{\text{old}} + T_x = 3 + 1 = 4$$

$$Y_{\text{new}} = Y_{\text{old}} + T_y = 0 + 1 = 1$$

Thus, New coordinates of corner C = $(4, 1)$.



For Coordinates D(o, o)

Let the new coordinates of corner D = $(X_{\text{new}}, Y_{\text{new}})$.

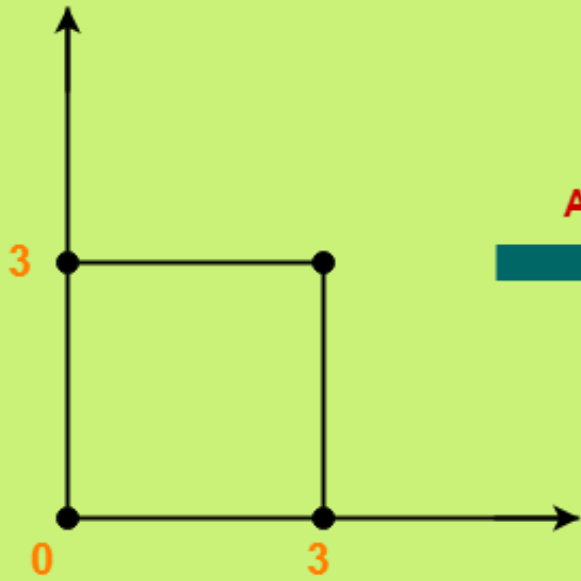
Applying the translation equations, we have-

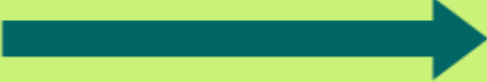
$$X_{\text{new}} = X_{\text{old}} + T_x = 0 + 1 = 1$$

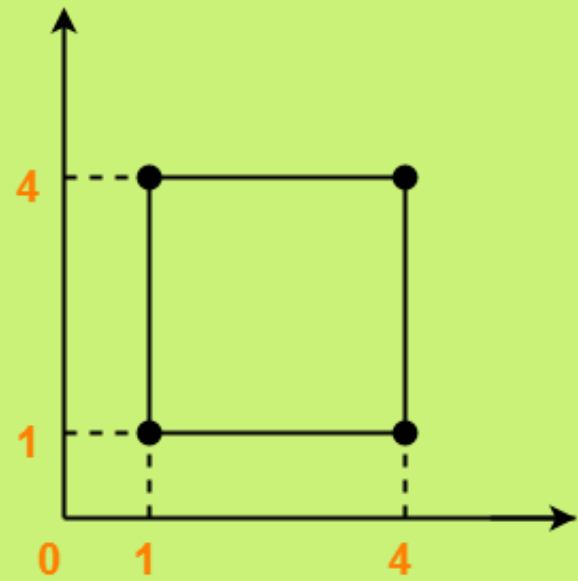
$$Y_{\text{new}} = Y_{\text{old}} + T_y = 0 + 1 = 1$$

Thus, New coordinates of corner D = (1, 1).

Thus, New coordinates of the square = A (1, 4), B(4, 4), C(4, 1), D(1, 1).



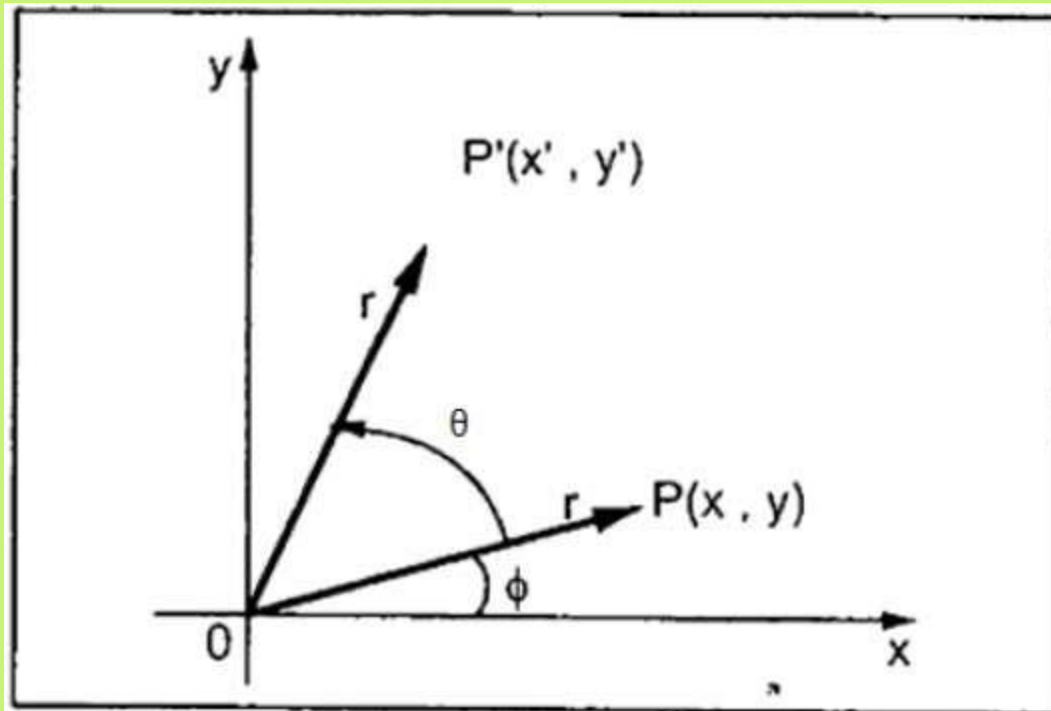
After Translation 



Rotation



- In rotation, we rotate the object at particular angle θ theta from its origin. From the following figure, we can see that the point P X,Y is located at angle φ phi from the horizontal X coordinate with distance r from the origin.
- Let us suppose you want to rotate it at the angle θ . After rotating it to a new location, you will get a new point P' X',Y'





- Using standard trigonometric the original coordinate of point P X,Y can be represented as –
- $X = r \cos \phi \dots \dots (1)$
- $Y = r \sin \phi \dots \dots (2)$
- Same way we can represent the point P' X',Y' as –
- $x' = r \cos(\phi + \theta) = r \cos \phi \cos \theta - r \sin \phi \sin \theta \dots \dots (3)$
- $y' = r \sin(\phi + \theta) = r \cos \phi \sin \theta + r \sin \phi \cos \theta \dots \dots (4)$



Substituting equation 1 & 2 in 3 & 4 respectively, we will get

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

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

Representing the above equation in matrix form,

$$[X'Y'] = [XY] \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \text{ OR}$$

$$P' = P \cdot R$$



- Where R is the rotation matrix

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

- The rotation angle can be positive and negative.
- For positive rotation angle, we can use the above rotation matrix. However, for negative angle rotation, the matrix will change as shown below –

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

$$= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} (\because \cos(-\theta) = \cos\theta \text{ and } \sin(-\theta) = -\sin\theta)$$

In an other word



Consider a point object O has to be rotated from one angle to another in a 2D plane.

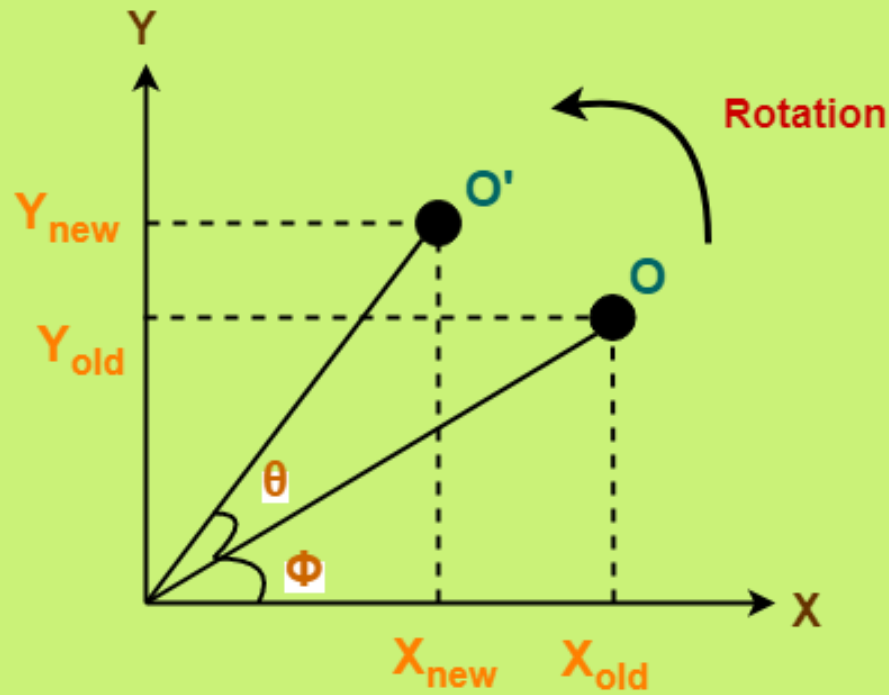
Let-

Initial coordinates of the object O = $(X_{\text{old}}, Y_{\text{old}})$

Initial angle of the object O with respect to origin = Φ

Rotation angle = θ

New coordinates of the object O after rotation = $(X_{\text{new}}, Y_{\text{new}})$



2D Rotation in Computer Graphics



- This rotation is achieved by using the following rotation equations-

$$X_{\text{new}} = X_{\text{old}} \times \cos\theta - Y_{\text{old}} \times \sin\theta$$

$$Y_{\text{new}} = X_{\text{old}} \times \sin\theta + Y_{\text{old}} \times \cos\theta$$



- In Matrix form, the above rotation equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \end{bmatrix}$$

Rotation Matrix



- For homogeneous coordinates, the above rotation matrix may be represented as a 3 x 3 matrix as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ 1 \end{bmatrix}$$

Rotation Matrix

(Homogeneous Coordinates Representation)

Example 1



- Given a line segment with starting point as $(0, 0)$ and ending point as $(4, 4)$. Apply 30 degree rotation anticlockwise direction on the line segment and find out the new coordinates of the line



- **Solution-**

-
- We rotate a straight line by its end points with the same angle. Then, we re-draw a line between the new end points.
-
- Given-
- Old ending coordinates of the line = $(X_{\text{old}}, Y_{\text{old}}) = (4, 4)$
- Rotation angle = $\theta = 30^\circ$
-
- Let new ending coordinates of the line after rotation = $(X_{\text{new}}, Y_{\text{new}})$.



Applying the rotation equations, we have-

$$\begin{aligned} X_{\text{new}} &= X_{\text{old}} \cos\theta - Y_{\text{old}} \sin\theta \\ &= 4 \times \cos 30^\circ - 4 \times \sin 30^\circ \\ &= 4 \times (\sqrt{3} / 2) - 4 \times (1 / 2) \\ &= 2\sqrt{3} - 2 \\ &= 2(\sqrt{3} - 1) \\ &= 2(1.73 - 1) \\ &= 1.46 \end{aligned}$$



$$\begin{aligned} Y_{\text{new}} &= X_{\text{old}} \times \sin\theta + Y_{\text{old}} \times \cos\theta \\ &= 4 \times \sin 30^\circ + 4 \times \cos 30^\circ \\ &= 4 \times (1/2) + 4 \times (\sqrt{3}/2) \\ &= 2 + 2\sqrt{3} \\ &= 2(1 + \sqrt{3}) \\ &= 2(1 + 1.73) \\ &= 5.46 \end{aligned}$$

Thus, New ending coordinates of the line after rotation = (1.46, 5.46).



- **Alternatively,**
- In matrix form, the new ending coordinates of the line after rotation may be obtained as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \end{bmatrix}$$

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \end{bmatrix} = \begin{bmatrix} \cos 30 & -\sin 30 \\ \sin 30 & \cos 30 \end{bmatrix} \times \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

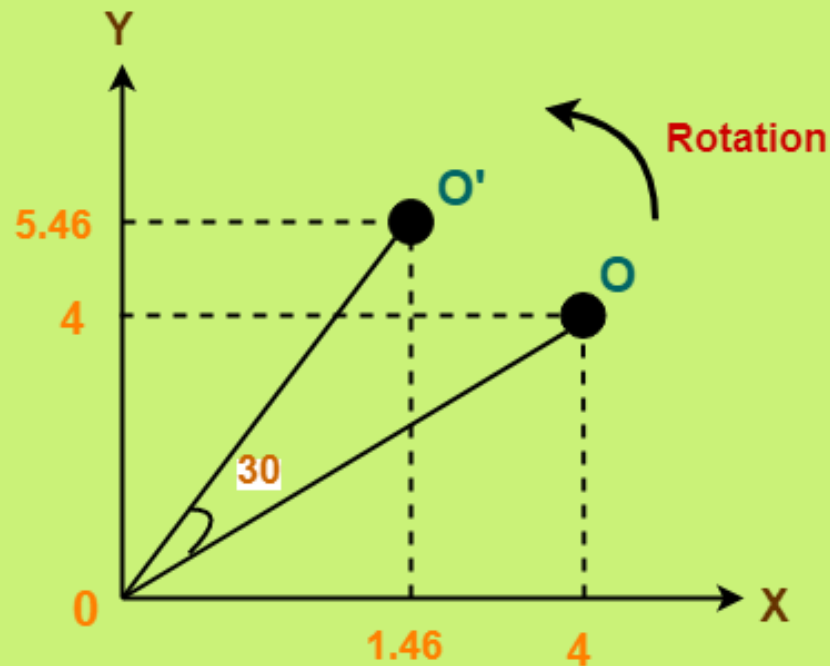
$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \end{bmatrix} = \begin{bmatrix} 4 \times \cos 30 - 4 \times \sin 30 \\ 4 \times \sin 30 + 4 \times \cos 30 \end{bmatrix}$$

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \end{bmatrix} = \begin{bmatrix} 4 \times \cos 30 - 4 \times \sin 30 \\ 4 \times \sin 30 + 4 \times \cos 30 \end{bmatrix}$$

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \end{bmatrix} = \begin{bmatrix} 1.46 \\ 5.46 \end{bmatrix}$$



- Thus, New ending coordinates of the line after rotation = $(1.46, 5.46)$



example2



- Given a triangle with corner coordinates $(0, 0)$, $(1, 0)$ and $(1, 1)$. Rotate the triangle by 90 degree anticlockwise direction and find out the new coordinates.



Solution-

We rotate a polygon by rotating each vertex of it with the same rotation angle.

Given-

Old corner coordinates of the triangle = A (0, 0), B(1, 0), C(1, 1)

Rotation angle = $\theta = 90^\circ$



For Coordinates A(0, 0)

Let the new coordinates of corner A after rotation = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the rotation equations, we have-

$$\begin{aligned} X_{\text{new}} &= X_{\text{old}} \times \cos\theta - Y_{\text{old}} \times \sin\theta \\ &= 0 \times \cos 90^\circ - 0 \times \sin 90^\circ \\ &= 0 \end{aligned}$$

$$\begin{aligned} Y_{\text{new}} &= X_{\text{old}} \times \sin\theta + Y_{\text{old}} \times \cos\theta \\ &= 0 \times \sin 90^\circ + 0 \times \cos 90^\circ \\ &= 0 \end{aligned}$$

Thus, New coordinates of corner A after rotation = $(0, 0)$.



For Coordinates B(1, 0)

Let the new coordinates of corner B after rotation = $(X_{\text{new}}, Y_{\text{new}})$.

$$\begin{aligned} X_{\text{new}} &= X_{\text{old}} \times \cos\theta - Y_{\text{old}} \times \sin\theta \\ &= 1 \times \cos 90^\circ - 0 \times \sin 90^\circ \\ &= 0 \end{aligned}$$

$$\begin{aligned} Y_{\text{new}} &= X_{\text{old}} \times \sin\theta + Y_{\text{old}} \times \cos\theta \\ &= 1 \times \sin 90^\circ + 0 \times \cos 90^\circ \\ &= 1 + 0 \\ &= 1 \end{aligned}$$

Thus, New coordinates of corner B after rotation = $(0, 1)$.



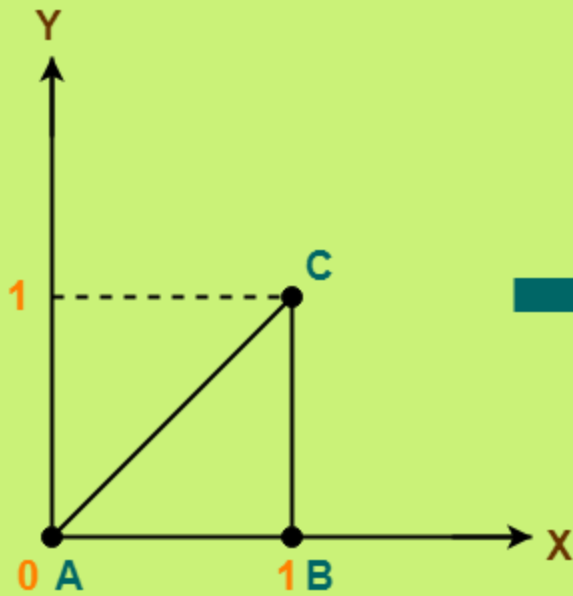
For Coordinates C(1, 1)

Let the new coordinates of corner C after rotation = $(X_{\text{new}}, Y_{\text{new}})$.

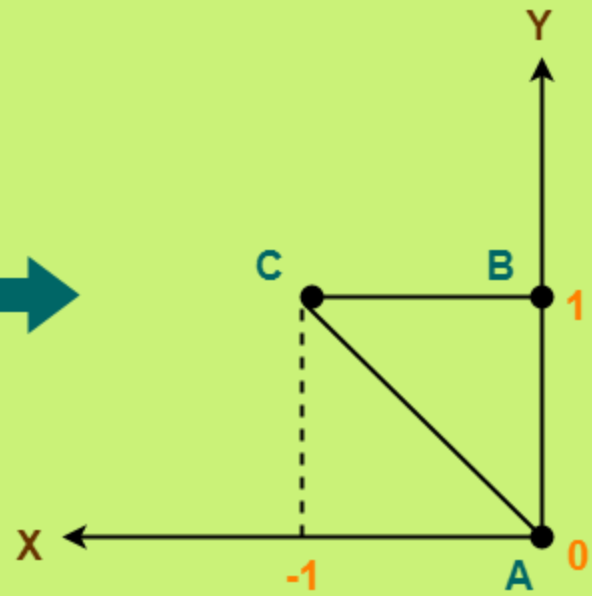
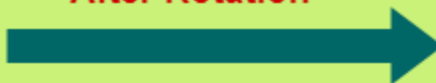
$$\begin{aligned} X_{\text{new}} &= X_{\text{old}} \times \cos\theta - Y_{\text{old}} \times \sin\theta \\ &= 1 \times \cos 90^\circ - 1 \times \sin 90^\circ \\ &= 0 - 1 \\ &= -1 \end{aligned}$$

$$\begin{aligned} Y_{\text{new}} &= X_{\text{old}} \times \sin\theta + Y_{\text{old}} \times \cos\theta \\ &= 1 \times \sin 90^\circ + 1 \times \cos 90^\circ \\ &= 1 + 0 \\ &= 1 \end{aligned}$$

Thus, New coordinates of corner C after rotation = $(-1, 1)$.



After Rotation



Scaling



- To change the size of an object, scaling transformation is used. In the scaling process, you either expand or compress the dimensions of the object. Scaling can be achieved by multiplying the original coordinates of the object with the scaling factor to get the desired result.
- Let us assume that the original coordinates are X, Y , the scaling factors are (S_x, S_y) , and the produced coordinates are X', Y' . This can be mathematically represented as shown below –

$$X' = X \cdot S_x \text{ and } Y' = Y \cdot S_y$$



- The scaling factor S_X , S_Y scales the object in X and Y direction respectively. The above equations can also be represented in matrix form as below –

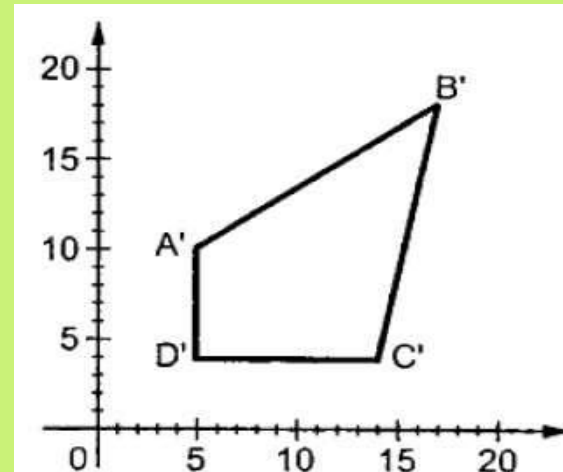
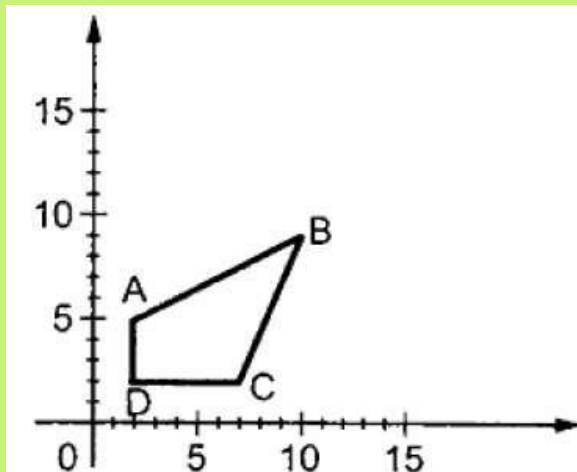
$$\begin{pmatrix} X' \\ Y' \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

- OR

$$\mathbf{P}' = \mathbf{P} \cdot \mathbf{S}$$



- Where S is the scaling matrix. The scaling process is shown in the following figure.

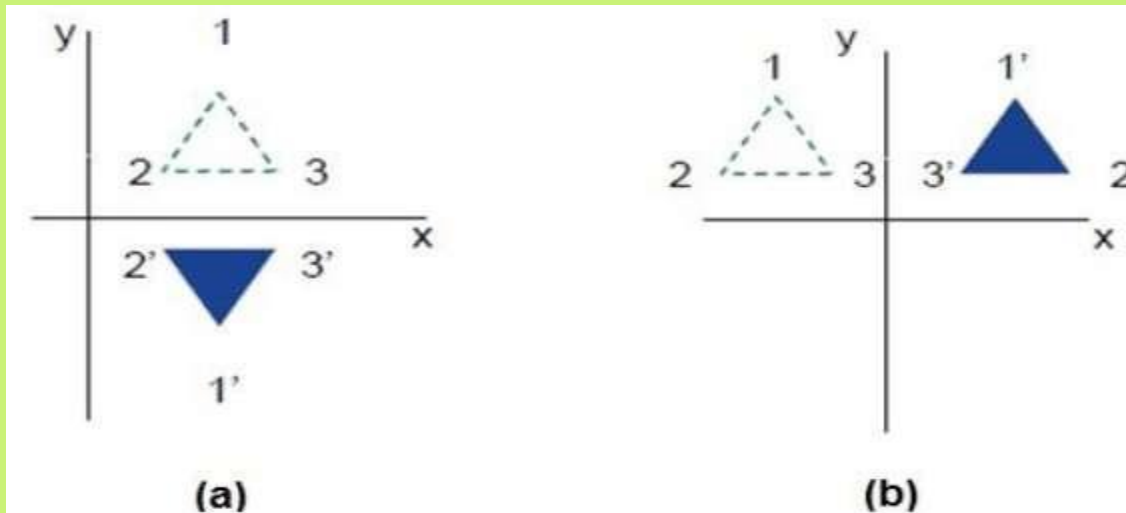


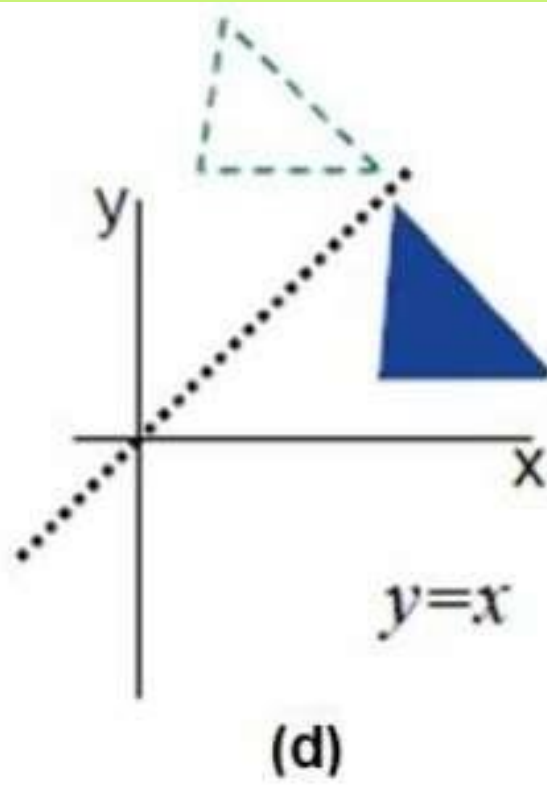
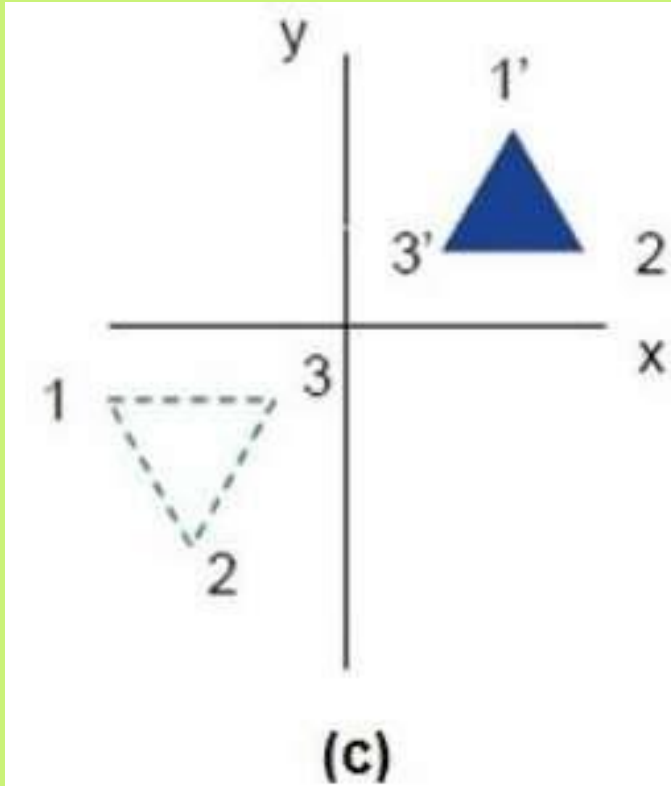
- If we provide values less than 1 to the scaling factor S , then we can reduce the size of the object. If we provide values greater than 1, then we can increase the size of the object.

Reflection



- Reflection is the mirror image of original object. In other words, we can say that it is a rotation operation with 180° . In reflection transformation, the size of the object does not change.
- The following figures show reflections with respect to X and Y axes, and about the origin respectively.





In other word



- Consider a point object O has to be scaled in a 2D plane.
- Let-
- Initial coordinates of the object O = (X_{old}, Y_{old})
- Scaling factor for X-axis = S_x
- Scaling factor for Y-axis = S_y
- New coordinates of the object O after scaling = (X_{new}, Y_{new})
- This scaling is achieved by using the following scaling equations:

$$X_{new} = X_{old} \times S_x$$

$$Y_{new} = Y_{old} \times S_y$$



- In Matrix form, the above scaling equations may be represented as:

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \end{bmatrix}$$

Scaling Matrix



- For homogeneous coordinates, the above scaling matrix may be represented as a 3 x 3 matrix as

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ 1 \end{bmatrix}$$

Scaling Matrix

(Homogeneous Coordinates Representation)

Example1



- Given a square object with coordinate points $A(0, 3)$, $B(3, 3)$, $C(3, 0)$, $D(0, 0)$. Apply the scaling parameter 2 towards X axis and 3 towards Y axis and obtain the new coordinates of the object.



Solution-

Given-

Old corner coordinates of the square = A (0, 3), B(3, 3), C(3, 0), D(0, 0)

Scaling factor along X axis = 2

Scaling factor along Y axis = 3



- **For Coordinates A(0, 3)**

Let the new coordinates of corner A after scaling = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the scaling equations, we have-

$$X_{\text{new}} = X_{\text{old}} \times S_x = 0 \times 2 = 0$$

$$Y_{\text{new}} = Y_{\text{old}} \times S_y = 3 \times 3 = 9$$

Thus, New coordinates of corner A after scaling = $(0, 9)$.



For Coordinates B(3, 3)

Let the new coordinates of corner B after scaling = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the scaling equations, we have-

$$X_{\text{new}} = X_{\text{old}} \times S_x = 3 \times 2 = 6$$

$$Y_{\text{new}} = Y_{\text{old}} \times S_y = 3 \times 3 = 9$$

Thus, New coordinates of corner B after scaling = $(6, 9)$.



For Coordinates C(3, 0)

Let the new coordinates of corner C after scaling = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the scaling equations, we have-

$$X_{\text{new}} = X_{\text{old}} \times S_x = 3 \times 2 = 6$$

$$Y_{\text{new}} = Y_{\text{old}} \times S_y = 0 \times 3 = 0$$

Thus, New coordinates of corner C after scaling = $(6, 0)$.



For Coordinates D(0, 0)

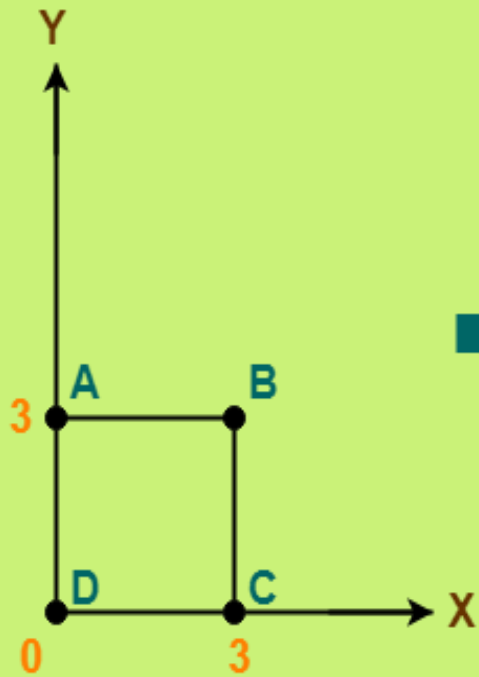
Let the new coordinates of corner D after scaling = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the scaling equations, we have-

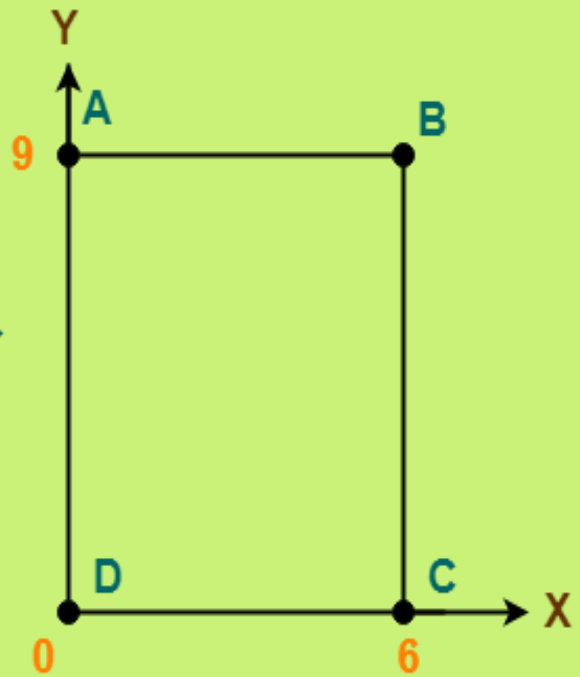
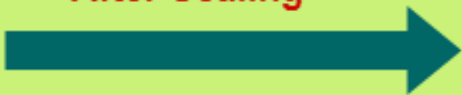
$$X_{\text{new}} = X_{\text{old}} \times S_x = 0 \times 2 = 0$$

$$Y_{\text{new}} = Y_{\text{old}} \times S_y = 0 \times 3 = 0$$

Thus, New coordinates of corner D after scaling = $(0, 0)$.



After Scaling



Reflection



- Reflection is a kind of rotation where the angle of rotation is 180 degree.
 - The reflected object is always formed on the other side of mirror.
 - The size of reflected object is same as the size of original object.
 - Consider a point object O has to be reflected in a 2D plane.
- Let-
- Initial coordinates of the object O = (X_{old}, Y_{old})
- New coordinates of the reflected object O after reflection = (X_{new}, Y_{new})



- **Reflection On X-Axis:**
-
- This reflection is achieved by using the following reflection equations-
- $X_{\text{new}} = X_{\text{old}}$
- $Y_{\text{new}} = -Y_{\text{old}}$



- In Matrix form, the above reflection equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \end{bmatrix}$$

Reflection Matrix
(Reflection Along X Axis)



- For homogeneous coordinates, the above reflection matrix may be represented as a 3 x 3 matrix as

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ 1 \end{bmatrix}$$

Reflection Matrix

(Reflection Along X Axis)

(Homogeneous Coordinates Representation)



- **Reflection On Y-Axis:**
-
- This reflection is achieved by using the following reflection equations-
- $X_{\text{new}} = -X_{\text{old}}$
- $Y_{\text{new}} = Y_{\text{old}}$



- In Matrix form, the above reflection equations may be represented as:

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \end{bmatrix}$$

Reflection Matrix
(Reflection Along Y Axis)



- For homogeneous coordinates, the above reflection matrix may be represented as a 3 x 3 matrix as:

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ 1 \end{bmatrix}$$

Reflection Matrix

(Reflection Along Y Axis)

(Homogeneous Coordinates Representation)

Example1



- Given a triangle with coordinate points $A(3, 4)$, $B(6, 4)$, $C(5, 6)$. Apply the reflection on the X axis and obtain the new coordinates of the object.



Solution-

Given-

Old corner coordinates of the triangle = A (3, 4), B(6, 4), C(5, 6)

Reflection has to be taken on the X axis

For Coordinates A(3, 4)

Let the new coordinates of corner A after reflection = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the reflection equations, we have-

$$X_{\text{new}} = X_{\text{old}} = 3$$

$$Y_{\text{new}} = -Y_{\text{old}} = -4$$

Thus, New coordinates of corner A after reflection = (3, -4).



For Coordinates B(6, 4)

Let the new coordinates of corner B after reflection = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the reflection equations, we have-

$$X_{\text{new}} = X_{\text{old}} = 6$$

$$Y_{\text{new}} = -Y_{\text{old}} = -4$$

Thus, New coordinates of corner B after reflection = $(6, -4)$.



For Coordinates C(5, 6)

Let the new coordinates of corner C after reflection = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the reflection equations, we have-

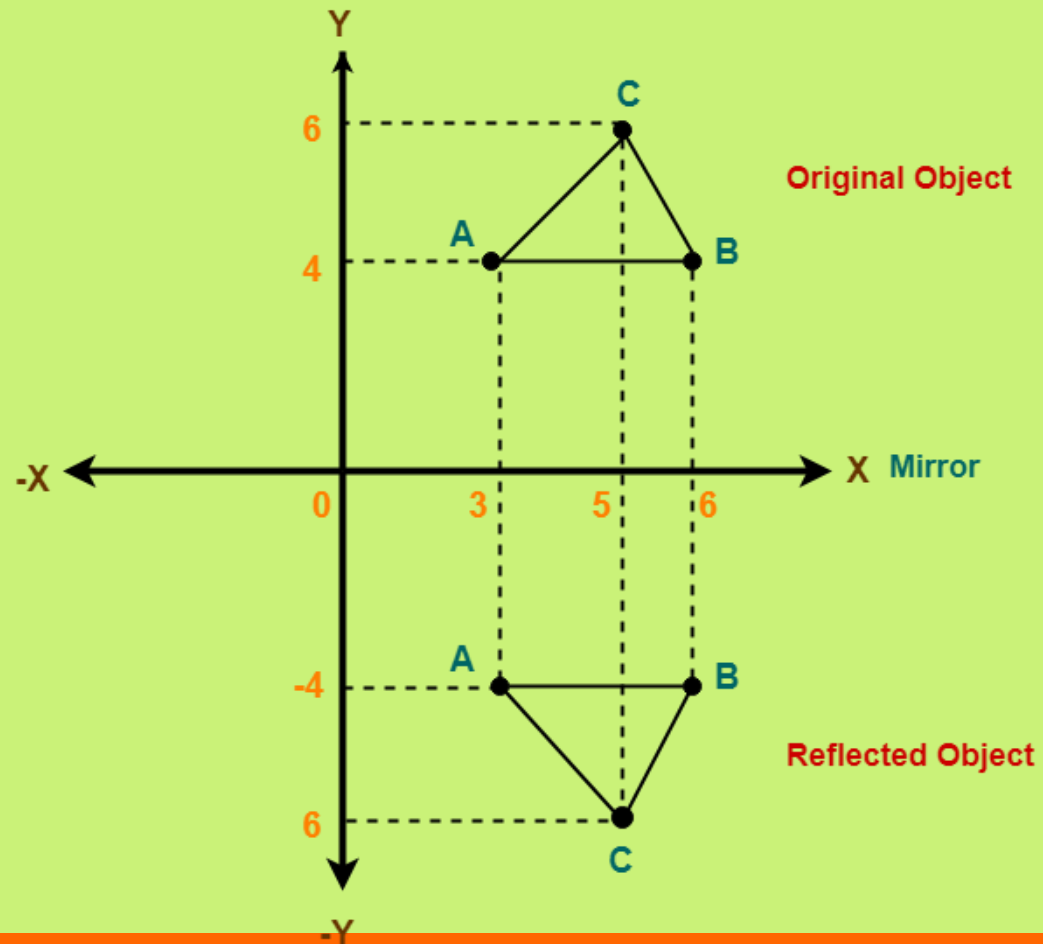
$$X_{\text{new}} = X_{\text{old}} = 5$$

$$Y_{\text{new}} = -Y_{\text{old}} = -6$$

Thus, New coordinates of corner C after reflection = (5, -6).



- Thus, New coordinates of the triangle after reflection = A (3, -4), B(6, -4), C(5, -6).



Example 2



- Given a triangle with coordinate points $A(3, 4)$, $B(6, 4)$, $C(5, 6)$. Apply the reflection on the Y axis and obtain the new coordinates of the object.



- **Solution-**
-
- Given-
- Old corner coordinates of the triangle = A (3, 4), B(6, 4), C(5, 6)
- Reflection has to be taken on the Y axis
-
- **For Coordinates A(3, 4)**
-
- Let the new coordinates of corner A after reflection = $(X_{\text{new}}, Y_{\text{new}})$.
-
- Applying the reflection equations, we have-
- $X_{\text{new}} = -X_{\text{old}} = -3$
- $Y_{\text{new}} = Y_{\text{old}} = 4$
-
- Thus, New coordinates of corner A after reflection = $(-3, 4)$.



For Coordinates B(6, 4)

Let the new coordinates of corner B after reflection = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the reflection equations, we have-

$$X_{\text{new}} = -X_{\text{old}} = -6$$

$$Y_{\text{new}} = Y_{\text{old}} = 4$$

Thus, New coordinates of corner B after reflection = $(-6, 4)$.



For Coordinates C(5, 6)

Let the new coordinates of corner C after reflection = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the reflection equations, we have-

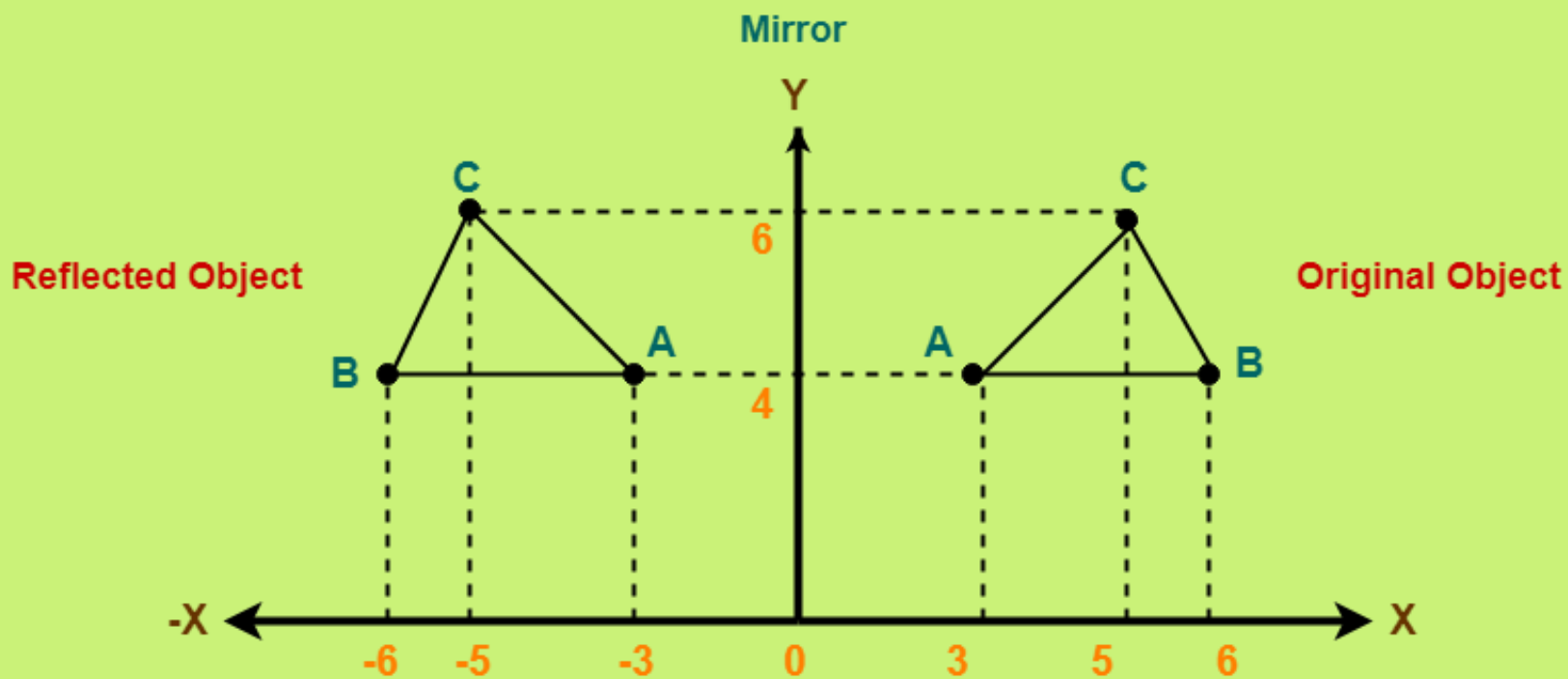
$$X_{\text{new}} = -X_{\text{old}} = -5$$

$$Y_{\text{new}} = Y_{\text{old}} = 6$$

Thus, New coordinates of corner C after reflection = $(-5, 6)$.



- Thus, New coordinates of the triangle after reflection = A (-3, 4), B(-6, 4), C(-5, 6).



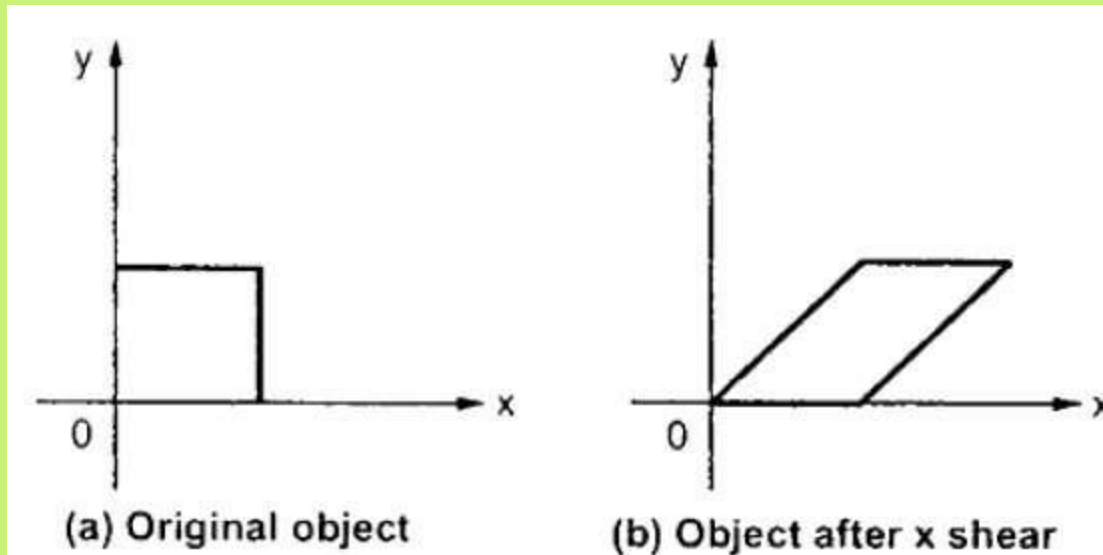
Shear



- A transformation that slants the shape of an object is called the shear transformation. There are two shear transformations **X-Shear** and **Y-Shear**. One shifts X coordinates values and other shifts Y coordinate values. However; in both the cases only one coordinate changes its coordinates and other preserves its values. Shearing is also termed as **Skewing**.



- X-Shear
- The X-Shear preserves the Y coordinate and changes the X coordinates, which causes the vertical lines to tilt right or left as shown in the below figure.





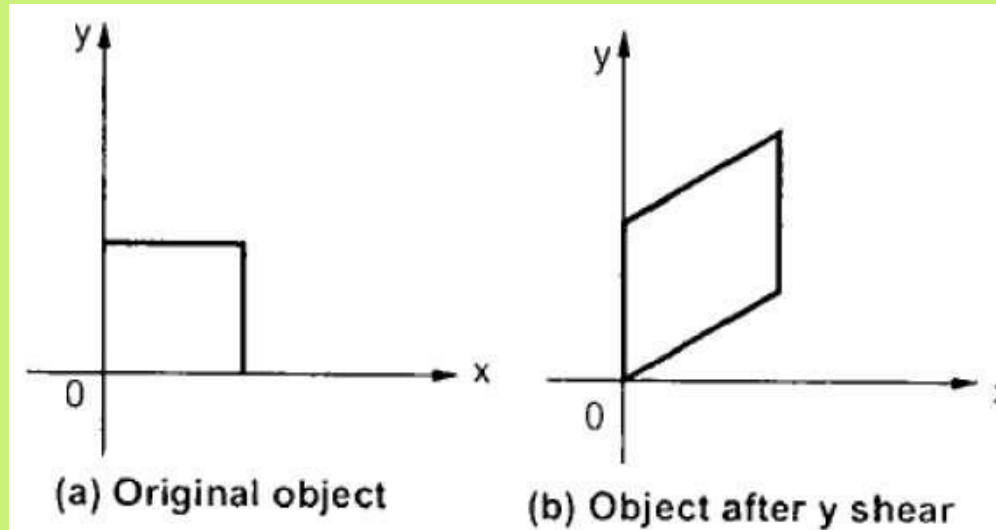
- The transformation matrix for X-Shear can be represented as :

$$X_{sh} = \begin{bmatrix} 1 & shx & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Y' = Y + Sh_y \cdot X$$



- Y-Shear
- The Y-Shear preserves the X coordinates and changes the Y coordinates which causes the horizontal lines to transform into lines which slopes up or down as shown in the following figure.





- The Y-Shear can be represented in matrix form as :

$$Y_{sh} \begin{bmatrix} 1 & 0 & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$X' = X + Sh_x \cdot Y$$

$$Y' = Y$$

In other word



Consider a point object O has to be sheared in a 2D plane.

Let-

Initial coordinates of the object O = (X_{old}, Y_{old})

Shearing parameter towards X direction = Sh_x

Shearing parameter towards Y direction = Sh_y

New coordinates of the object O after shearing = (X_{new}, Y_{new})





- **Shearing in X Axis-**

- Shearing in X axis is achieved by using the following shearing equations-

- $X_{\text{new}} = X_{\text{old}} + Sh_x \times Y_{\text{old}}$

- $Y_{\text{new}} = Y_{\text{old}}$



- In Matrix form, the above shearing equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \end{bmatrix} = \begin{bmatrix} 1 & Sh_x \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \end{bmatrix}$$

Shearing Matrix

(In X axis)



- For homogeneous coordinates, the above shearing matrix may be represented as a 3 x 3 matrix as:

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ 1 \end{bmatrix}$$

Shearing Matrix

(In X axis)

(Homogeneous Coordinates Representation)



- **Shearing in Y Axis-**

Shearing in Y axis is achieved by using the following shearing equations-

$$X_{\text{new}} = X_{\text{old}}$$

$$Y_{\text{new}} = Y_{\text{old}} + Sh_y \times X_{\text{old}}$$



- In Matrix form, the above shearing equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ Sh_y & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \end{bmatrix}$$

Shearing Matrix

(In Y axis)



- For homogeneous coordinates, the above shearing matrix may be represented as a 3 x 3 matrix as:

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & Sh_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ 1 \end{bmatrix}$$

Shearing Matrix

(In Y axis)

(Homogeneous Coordinates Representation)

Example1



- Given a triangle with points $(1, 1)$, $(0, 0)$ and $(1, 0)$. Apply shear parameter 2 on X axis and 2 on Y axis and find out the new coordinates of the object.



Solution-

Given-

Old corner coordinates of the triangle = A (1, 1), B(0, 0), C(1, 0)

Shearing parameter towards X direction (Sh_x) = 2

Shearing parameter towards Y direction (Sh_y) = 2

Shearing in X Axis-

For Coordinates A(1, 1)

Let the new coordinates of corner A after shearing = (X_{new} , Y_{new}).

Applying the shearing equations, we have-

$$X_{new} = X_{old} + Sh_x \times Y_{old} = 1 + 2 \times 1 = 3$$

$$Y_{new} = Y_{old} = 1$$

Thus, New coordinates of corner A after shearing = (3, 1).



- **For Coordinates B(0, 0)**
-
- Let the new coordinates of corner B after shearing = $(X_{\text{new}}, Y_{\text{new}})$.
-
- Applying the shearing equations, we have-
- $X_{\text{new}} = X_{\text{old}} + Sh_x \times Y_{\text{old}} = 0 + 2 \times 0 = 0$
- $Y_{\text{new}} = Y_{\text{old}} = 0$
-
- Thus, New coordinates of corner B after shearing = $(0, 0)$.
-



- **For Coordinates C(1, 0)**
-
- Let the new coordinates of corner C after shearing = $(X_{\text{new}}, Y_{\text{new}})$.
-
- Applying the shearing equations, we have-
- $X_{\text{new}} = X_{\text{old}} + Sh_x \times Y_{\text{old}} = 1 + 2 \times 0 = 1$
- $Y_{\text{new}} = Y_{\text{old}} = 0$
-
- Thus, New coordinates of corner C after shearing = (1, 0).
- Thus, New coordinates of the triangle after shearing in X axis = A (3, 1), B(0, 0), C(1, 0).



Shearing in Y Axis-

For Coordinates A(1, 1)

Let the new coordinates of corner A after shearing = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the shearing equations, we have-

$$X_{\text{new}} = X_{\text{old}} = 1$$

$$Y_{\text{new}} = Y_{\text{old}} + Sh_y \times X_{\text{old}} = 1 + 2 \times 1 = 3$$

Thus, New coordinates of corner A after shearing = $(1, 3)$.



For Coordinates B(0, 0)

Let the new coordinates of corner B after shearing = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the shearing equations, we have-

$$X_{\text{new}} = X_{\text{old}} = 0$$

$$Y_{\text{new}} = Y_{\text{old}} + S h_y \times X_{\text{old}} = 0 + 2 \times 0 = 0$$

Thus, New coordinates of corner B after shearing = $(0, 0)$.



For Coordinates C(1, 0)

Let the new coordinates of corner C after shearing = $(X_{\text{new}}, Y_{\text{new}})$.

Applying the shearing equations, we have-

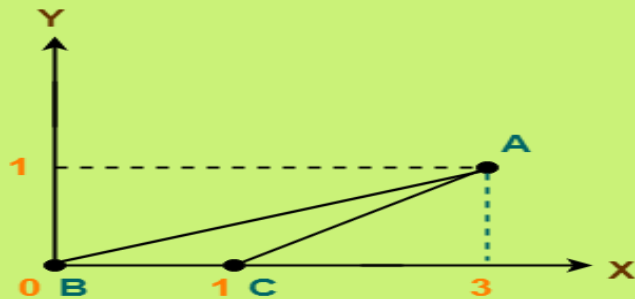
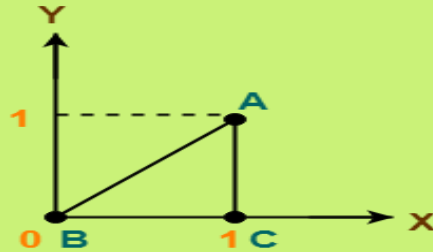
$$X_{\text{new}} = X_{\text{old}} = 1$$

$$Y_{\text{new}} = Y_{\text{old}} + Sh_y \times X_{\text{old}} = 0 + 2 \times 1 = 2$$

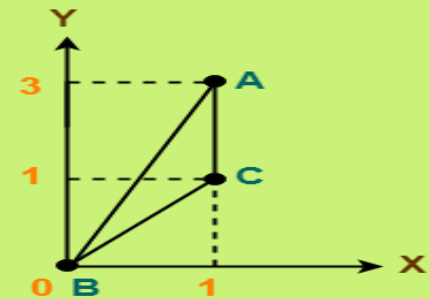
Thus, New coordinates of corner C after shearing = $(1, 2)$.



- Thus, New coordinates of the triangle after shearing in Y axis = A (1, 3), B(0, 0), C(1, 2).



Shearing in X Axis



Shearing in Y Axis

Composite transformation



- Sequence of transformation

Ex: What is the value (X' , Y') after applying sequence of transformation on the point (4, 4)

1. Scaling (0.5, 0.5)
2. Clockwise rotation 90 degrees
3. Translation with the vector (1, 1).

Note :The first transformation will be the last matrix



Translation

$t_x = 0.5$

$t_y = 0.5$

Rotation

$\cos -90 = 0$

$\sin -90 = -1$

Scaling

$S_x = 0.5$

$S_y = 0.5$

$$\begin{pmatrix} X' \\ Y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos -90 & -\sin -90 & 0 \\ \sin -90 & \cos -90 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 \\ 4 \\ 1 \end{pmatrix}$$
$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



$$\begin{pmatrix} X' \\ Y' \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 \\ 4 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} X' \\ Y' \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 0.5 & 1 \\ -0.5 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 \\ 4 \\ 1 \end{pmatrix}$$

$$X' = 3$$

$$Y' = -1 \longrightarrow \text{point} = (3, -1)$$



3D Transformation in Computer Graphics

3D Translation in Computer Graphics



- In Computer graphics,
- 3D Translation is a process of moving an object from one position to another in a three dimensional plane.

Translation



- Consider a point object O has to be moved from one position to another in a 3D plane.

Let-

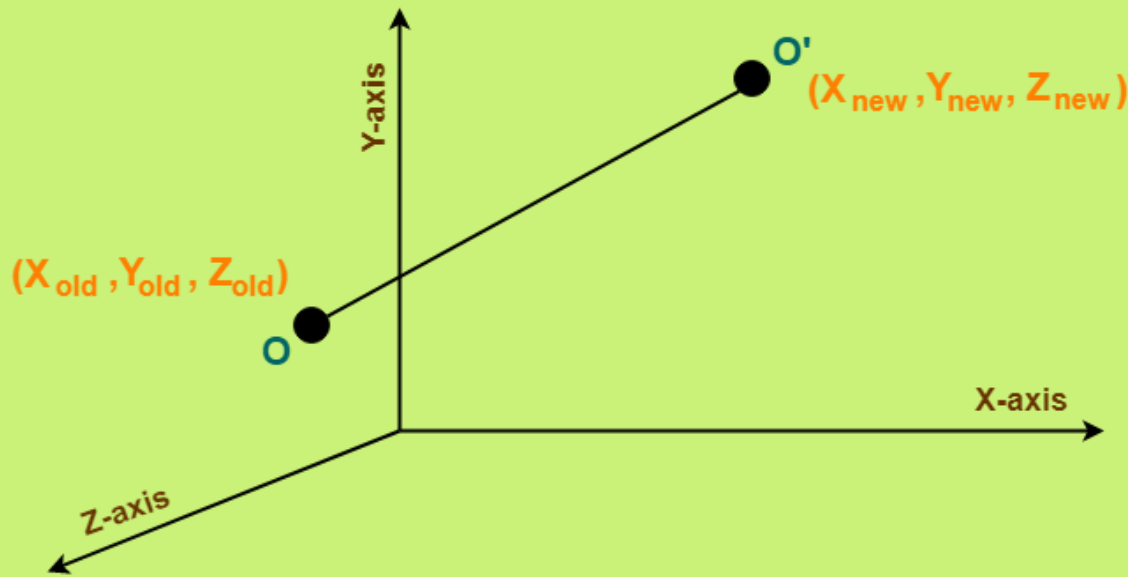
Initial coordinates of the object O = $(X_{old}, Y_{old}, Z_{old})$

New coordinates of the object O after translation = $(X_{new}, Y_{new}, Z_{old})$

Translation vector or Shift vector = (T_x, T_y, T_z)



- Given a Translation vector (T_x, T_y, T_z) -
 T_x defines the distance the X_{old} coordinate has to be moved.
 T_y defines the distance the Y_{old} coordinate has to be moved.
 T_z defines the distance the Z_{old} coordinate has to be moved.



3D Translation in Computer Graphics



- This translation is achieved by adding the translation coordinates to the old coordinates of the object as:
- $X_{\text{new}} = X_{\text{old}} + T_x$ (This denotes translation towards X axis)
- $Y_{\text{new}} = Y_{\text{old}} + T_y$ (This denotes translation towards Y axis)
- $Z_{\text{new}} = Z_{\text{old}} + T_z$ (This denotes translation towards Z axis)



- In Matrix form, the above translation equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ Z_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ Z_{\text{old}} \\ 1 \end{bmatrix}$$

3D Translation Matrix

Example 1



- Given a 3D object with coordinate points $A(0, 3, 1)$, $B(3, 3, 2)$, $C(3, 0, 0)$, $D(0, 0, 0)$. Apply the translation with the distance 1 towards X axis, 1 towards Y axis and 2 towards Z axis and obtain the new coordinates of the object



- **Solution-**
- Given-
- Old coordinates of the object = A (0, 3, 1), B(3, 3, 2), C(3, 0, 0), D(0, 0, 0)
- Translation vector = $(T_x, T_y, T_z) = (1, 1, 2)$

For Coordinates A(0, 3, 1)

Let the new coordinates of A = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the translation equations, we have-

$$X_{\text{new}} = X_{\text{old}} + T_x = 0 + 1 = 1$$

$$Y_{\text{new}} = Y_{\text{old}} + T_y = 3 + 1 = 4$$

$$Z_{\text{new}} = Z_{\text{old}} + T_z = 1 + 2 = 3$$

Thus, New coordinates of A = (1, 4, 3).



For Coordinates B(3, 3, 2)

Let the new coordinates of B = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the translation equations, we have-

$$X_{\text{new}} = X_{\text{old}} + T_x = 3 + 1 = 4$$

$$Y_{\text{new}} = Y_{\text{old}} + T_y = 3 + 1 = 4$$

$$Z_{\text{new}} = Z_{\text{old}} + T_z = 2 + 2 = 4$$

Thus, New coordinates of B = $(4, 4, 4)$.

For Coordinates C(3, 0, 0)

Let the new coordinates of C = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the translation equations, we have-

$$X_{\text{new}} = X_{\text{old}} + T_x = 3 + 1 = 4$$

$$Y_{\text{new}} = Y_{\text{old}} + T_y = 0 + 1 = 1$$

$$Z_{\text{new}} = Z_{\text{old}} + T_z = 0 + 2 = 2$$

Thus, New coordinates of C = $(4, 1, 2)$.



For Coordinates D(0, 0, 0)

Let the new coordinates of D = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the translation equations, we have-

$$X_{\text{new}} = X_{\text{old}} + T_x = 0 + 1 = 1$$

$$Y_{\text{new}} = Y_{\text{old}} + T_y = 0 + 1 = 1$$

$$Z_{\text{new}} = Z_{\text{old}} + T_z = 0 + 2 = 2$$

Thus, New coordinates of D = (1, 1, 2).

- Thus, New coordinates of the object = A (1, 4, 3), B(4, 4, 4), C(4, 1, 2), D(1, 1, 2).

3D Rotation in Computer Graphics-



- In Computer graphics,
- 3D Rotation is a process of rotating an object with respect to an angle in a three dimensional plane.
- Consider a point object O has to be rotated from one angle to another in a 3D plane.

Let-

Initial coordinates of the object O = $(X_{old}, Y_{old}, Z_{old})$

Initial angle of the object O with respect to origin = Φ

Rotation angle = θ

New coordinates of the object O after rotation = $(X_{new}, Y_{new}, Z_{new})$



- In 3 dimensions, there are 3 possible types of rotation-
- X-axis Rotation
- Y-axis Rotation
- Z-axis Rotation



• For X-Axis Rotation-

This rotation is achieved by using the following rotation equations-

$$X_{\text{new}} = X_{\text{old}}$$

$$Y_{\text{new}} = Y_{\text{old}} \times \cos\theta - Z_{\text{old}} \times \sin\theta$$

$$Z_{\text{new}} = Y_{\text{old}} \times \sin\theta + Z_{\text{old}} \times \cos\theta$$

In Matrix form, the above rotation equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ Z_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ Z_{\text{old}} \\ 1 \end{bmatrix}$$

3D Rotation Matrix
(For X-Axis Rotation)



• For Y-Axis Rotation-

This rotation is achieved by using the following rotation equations-

$$X_{\text{new}} = Z_{\text{old}} \times \sin\theta + X_{\text{old}} \times \cos\theta$$

$$Y_{\text{new}} = Y_{\text{old}}$$

$$Z_{\text{new}} = Y_{\text{old}} \times \cos\theta - X_{\text{old}} \times \sin\theta$$

In Matrix form, the above rotation equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ Z_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ Z_{\text{old}} \\ 1 \end{bmatrix}$$

3D Rotation Matrix
(For Y-Axis Rotation)



• For Z-Axis Rotation-

This rotation is achieved by using the following rotation equations-

$$X_{\text{new}} = X_{\text{old}} \times \cos\theta - Y_{\text{old}} \times \sin\theta$$

$$Y_{\text{new}} = X_{\text{old}} \times \sin\theta + Y_{\text{old}} \times \cos\theta$$

$$Z_{\text{new}} = Z_{\text{old}}$$

In Matrix form, the above rotation equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ Z_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ Z_{\text{old}} \\ 1 \end{bmatrix}$$

3D Rotation Matrix
(For Z-Axis Rotation)



- Given a homogeneous point $(1, 2, 3)$. Apply rotation 90 degree towards X, Y and Z axis and find out the new coordinate points.



- **Solution-**

Given-

Old coordinates = $(X_{old}, Y_{old}, Z_{old}) = (1, 2, 3)$

Rotation angle = $\theta = 90^\circ$

- **For X-Axis Rotation-**

Let the new coordinates after rotation = $(X_{new}, Y_{new}, Z_{new})$.

Applying the rotation equations, we have-

$$X_{new} = X_{old} = 1$$

$$Y_{new} = Y_{old} \times \cos\theta - Z_{old} \times \sin\theta = 2 \times \cos 90^\circ - 3 \times \sin 90^\circ = 2 \times 0 - 3 \times 1 = -3$$

$$Z_{new} = Y_{old} \times \sin\theta + Z_{old} \times \cos\theta = 2 \times \sin 90^\circ + 3 \times \cos 90^\circ = 2 \times 1 + 3 \times 0 = 2$$

Thus, New coordinates after rotation = $(1, -3, 2)$.



- **For Y-Axis Rotation-**

Let the new coordinates after rotation = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the rotation equations, we have-

$$X_{\text{new}} = Z_{\text{old}} \times \sin\theta + X_{\text{old}} \times \cos\theta = 3 \times \sin 90^\circ + 1 \times \cos 90^\circ = 3 \times 1 + 1 \times 0 = 3$$

$$Y_{\text{new}} = Y_{\text{old}} = 2$$

$$Z_{\text{new}} = Y_{\text{old}} \times \cos\theta - X_{\text{old}} \times \sin\theta = 2 \times \cos 90^\circ - 1 \times \sin 90^\circ = 2 \times 0 - 1 \times 1 = -1$$

- Thus, New coordinates after rotation = $(3, 2, -1)$.



- **For Z-Axis Rotation-**

Let the new coordinates after rotation = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the rotation equations, we have-

$$X_{\text{new}} = X_{\text{old}} \times \cos\theta - Y_{\text{old}} \times \sin\theta = 1 \times \cos 90^\circ - 2 \times \sin 90^\circ = 1 \times 0 - 2 \times 1 = -2$$

$$Y_{\text{new}} = X_{\text{old}} \times \sin\theta + Y_{\text{old}} \times \cos\theta = 1 \times \sin 90^\circ + 2 \times \cos 90^\circ = 1 \times 1 + 2 \times 0 = 1$$

$$Z_{\text{new}} = Z_{\text{old}} = 3$$

Thus, New coordinates after rotation = $(-2, 1, 3)$.

3D Scaling in Computer Graphics-



- In computer graphics, scaling is a process of modifying or altering the size of objects.
 - Scaling may be used to increase or reduce the size of object.
 - Scaling subjects the coordinate points of the original object to change.
 - Scaling factor determines whether the object size is to be increased or reduced.
 - If scaling factor > 1 , then the object size is increased.
 - If scaling factor < 1 , then the object size is reduced.



- Consider a point object O has to be scaled in a 3D plane.

Let-

Initial coordinates of the object O = $(X_{old}, Y_{old}, Z_{old})$

Scaling factor for X-axis = S_x

Scaling factor for Y-axis = S_y

Scaling factor for Z-axis = S_z

New coordinates of the object O after scaling = $(X_{new}, Y_{new}, Z_{new})$



- This scaling is achieved by using the following scaling equations-

- $X_{\text{new}} = X_{\text{old}} \times S_x$

- $Y_{\text{new}} = Y_{\text{old}} \times S_y$

- $Z_{\text{new}} = Z_{\text{old}} \times S_z$



- In Matrix form, the above scaling equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ Z_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ Z_{\text{old}} \\ 1 \end{bmatrix}$$

3D Scaling Matrix

Example



- Given a 3D object with coordinate points $A(0, 3, 3)$, $B(3, 3, 6)$, $C(3, 0, 1)$, $D(0, 0, 0)$. Apply the scaling parameter 2 towards X axis, 3 towards Y axis and 3 towards Z axis and obtain the new coordinates of the object.



- **Solution-**

Given-

Old coordinates of the object = A (0, 3, 3), B(3, 3, 6),
C(3, 0, 1), D(0, 0, 0)

Scaling factor along X axis = 2

Scaling factor along Y axis = 3

Scaling factor along Z axis = 3



- **For Coordinates A(0, 3, 3)**

Let the new coordinates of A after scaling = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the scaling equations, we have-

$$X_{\text{new}} = X_{\text{old}} \times S_x = 0 \times 2 = 0$$

$$Y_{\text{new}} = Y_{\text{old}} \times S_y = 3 \times 3 = 9$$

$$Z_{\text{new}} = Z_{\text{old}} \times S_z = 3 \times 3 = 9$$

Thus, New coordinates of corner A after scaling = $(0, 9, 9)$.



- **For Coordinates B(3, 3, 6)**

Let the new coordinates of B after scaling = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the scaling equations, we have-

$$X_{\text{new}} = X_{\text{old}} \times S_x = 3 \times 2 = 6$$

$$Y_{\text{new}} = Y_{\text{old}} \times S_y = 3 \times 3 = 9$$

$$Z_{\text{new}} = Z_{\text{old}} \times S_z = 6 \times 3 = 18$$

Thus, New coordinates of corner B after scaling = $(6, 9, 18)$.



- **For Coordinates C(3, 0, 1)**

Let the new coordinates of C after scaling = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the scaling equations, we have-

$$X_{\text{new}} = X_{\text{old}} \times S_x = 3 \times 2 = 6$$

$$Y_{\text{new}} = Y_{\text{old}} \times S_y = 0 \times 3 = 0$$

$$Z_{\text{new}} = Z_{\text{old}} \times S_z = 1 \times 3 = 3$$

Thus, New coordinates of corner C after scaling = $(6, 0, 3)$.



- **For Coordinates D(0, 0, 0)**

Let the new coordinates of D after scaling = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the scaling equations, we have-

$$X_{\text{new}} = X_{\text{old}} \times S_x = 0 \times 2 = 0$$

$$Y_{\text{new}} = Y_{\text{old}} \times S_y = 0 \times 3 = 0$$

$$Z_{\text{new}} = Z_{\text{old}} \times S_z = 0 \times 3 = 0$$

Thus, New coordinates of corner D after scaling = $(0, 0, 0)$.



3D Reflection in Computer Graphics-



- Reflection is a kind of rotation where the angle of rotation is 180 degree.
- The reflected object is always formed on the other side of mirror.
- The size of reflected object is same as the size of original object.

Consider a point object O has to be reflected in a 3D plane.

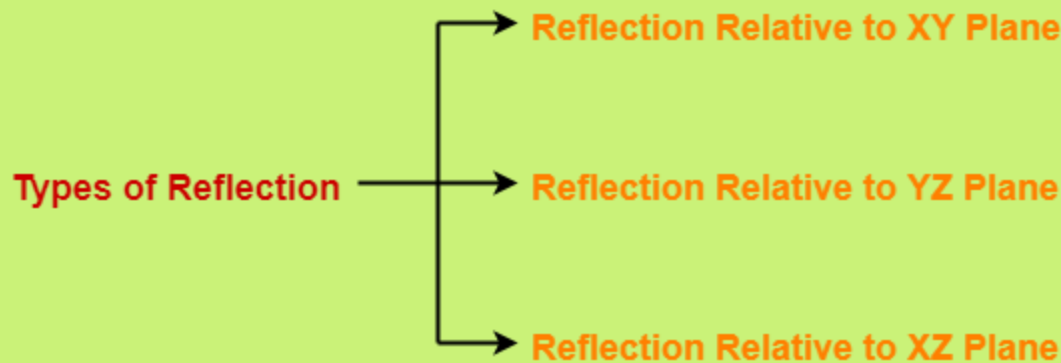
Let-

Initial coordinates of the object O = $(X_{old}, Y_{old}, Z_{old})$

New coordinates of the reflected object O after reflection = $(X_{new}, Y_{new}, Z_{new})$



- In 3 dimensions, there are 3 possible types of reflection-



- Reflection relative to XY plane
- Reflection relative to YZ plane
- Reflection relative to XZ plane



- **Reflection Relative to XY Plane:**

This reflection is achieved by using the following reflection equations-

$$X_{\text{new}} = X_{\text{old}}$$

$$Y_{\text{new}} = Y_{\text{old}}$$

$$Z_{\text{new}} = -Z_{\text{old}}$$

In Matrix form, the above reflection equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ Z_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ Z_{\text{old}} \\ 1 \end{bmatrix}$$

3D Reflection Matrix

(Reflection Relative to XY plane)



• Reflection Relative to YZ Plane:

This reflection is achieved by using the following reflection equations-

$$X_{\text{new}} = -X_{\text{old}}$$

$$Y_{\text{new}} = Y_{\text{old}}$$

$$Z_{\text{new}} = Z_{\text{old}}$$

In Matrix form, the above reflection equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ Z_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ Z_{\text{old}} \\ 1 \end{bmatrix}$$

3D Reflection Matrix

(Reflection Relative to YZ plane)



• Reflection Relative to XZ Plane:

This reflection is achieved by using the following reflection equations-

$$X_{\text{new}} = X_{\text{old}}$$

$$Y_{\text{new}} = -Y_{\text{old}}$$

$$Z_{\text{new}} = Z_{\text{old}}$$

In Matrix form, the above reflection equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ Z_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ Z_{\text{old}} \\ 1 \end{bmatrix}$$

3D Reflection Matrix

(Reflection Relative to XZ plane)

Example



- Given a 3D triangle with coordinate points $A(3, 4, 1)$, $B(6, 4, 2)$, $C(5, 6, 3)$. Apply the reflection on the XY plane and find out the new coordinates of the object.



- **Solution-**

Given-

Old corner coordinates of the triangle = A (3, 4, 1), B(6, 4, 2), C(5, 6, 3)

Reflection has to be taken on the XY plane

- **For Coordinates A(3, 4, 1)**

Let the new coordinates of corner A after reflection = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the reflection equations, we have-

$$X_{\text{new}} = X_{\text{old}} = 3$$

$$Y_{\text{new}} = Y_{\text{old}} = 4$$

$$Z_{\text{new}} = -Z_{\text{old}} = -1$$

Thus, New coordinates of corner A after reflection = (3, 4, -1).



- **For Coordinates B(6, 4, 2)**

Let the new coordinates of corner B after reflection = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the reflection equations, we have-

$$X_{\text{new}} = X_{\text{old}} = 6$$

$$Y_{\text{new}} = Y_{\text{old}} = 4$$

$$Z_{\text{new}} = -Z_{\text{old}} = -2$$

Thus, New coordinates of corner B after reflection = $(6, 4, -2)$.



- **For Coordinates C(5, 6, 3)**

Let the new coordinates of corner C after reflection = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the reflection equations, we have-

$$X_{\text{new}} = X_{\text{old}} = 5$$

$$Y_{\text{new}} = Y_{\text{old}} = 6$$

$$Z_{\text{new}} = -Z_{\text{old}} = -3$$

Thus, New coordinates of corner C after reflection = $(5, 6, -3)$.

Thus, New coordinates of the triangle after reflection = A $(3, 4, -1)$, B $(6, 4, -2)$, C $(5, 6, -3)$.

Example



- Given a 3D triangle with coordinate points $A(3, 4, 1)$, $B(6, 4, 2)$, $C(5, 6, 3)$. Apply the reflection on the XZ plane and find out the new coordinates of the object.



- **Solution-**

- Given-

- Old corner coordinates of the triangle = A (3, 4, 1), B(6, 4, 2), C(5, 6, 3)
- Reflection has to be taken on the XZ plane

- **For Coordinates A(3, 4, 1)**

Let the new coordinates of corner A after reflection = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the reflection equations, we have-

$$X_{\text{new}} = X_{\text{old}} = 3$$

$$Y_{\text{new}} = -Y_{\text{old}} = -4$$

$$Z_{\text{new}} = Z_{\text{old}} = 1$$

Thus, New coordinates of corner A after reflection = (3, -4, 1).



- **For Coordinates B(6, 4, 2)**

- Let the new coordinates of corner B after reflection = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the reflection equations, we have-

$$X_{\text{new}} = X_{\text{old}} = 6$$

$$Y_{\text{new}} = -Y_{\text{old}} = -4$$

$$Z_{\text{new}} = Z_{\text{old}} = 2$$

Thus, New coordinates of corner B after reflection = $(6, -4, 2)$.



- **For Coordinates C(5, 6, 3)**

Let the new coordinates of corner C after reflection = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the reflection equations, we have-

$$X_{\text{new}} = X_{\text{old}} = 5$$

$$Y_{\text{new}} = -Y_{\text{old}} = -6$$

$$Z_{\text{new}} = Z_{\text{old}} = 3$$

Thus, New coordinates of corner C after reflection = $(5, -6, 3)$.

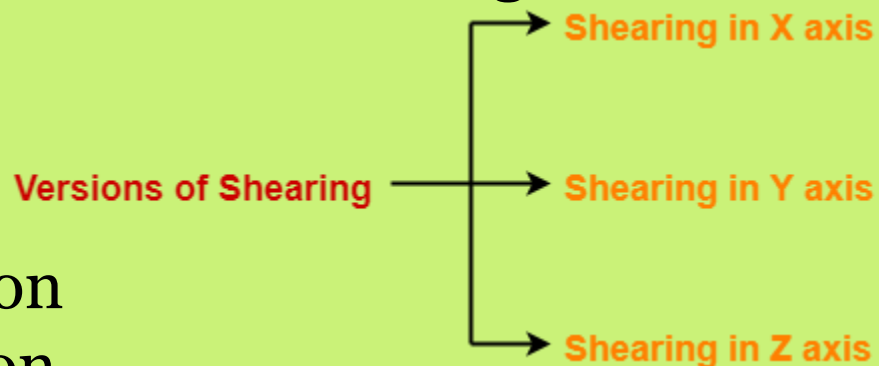
Thus, New coordinates of the triangle after reflection = A $(3, -4, 1)$, B $(6, -4, 2)$, C $(5, -6, 3)$.



3D Shearing in Computer Graphics-



- In Computer graphics, 3D Shearing is an ideal technique to change the shape of an existing object in a three dimensional plane.
- In a three dimensional plane, the object size can be changed along X direction, Y direction as well as Z direction.
- So, there are three versions of shearing-



- Shearing in X direction
- Shearing in Y direction
- Shearing in Z direction



- Consider a point object O has to be sheared in a 3D plane.

Let-

- Initial coordinates of the object O = $(X_{old}, Y_{old}, Z_{old})$
- Shearing parameter towards X direction = Sh_x
- Shearing parameter towards Y direction = Sh_y
- Shearing parameter towards Z direction = Sh_z
- New coordinates of the object O after shearing = $(X_{new}, Y_{new}, Z_{new})$



- **Shearing in X Axis-**

Shearing in X axis is achieved by using the following shearing equations-

$$X_{\text{new}} = X_{\text{old}}$$

$$Y_{\text{new}} = Y_{\text{old}} + Sh_y \times X_{\text{old}}$$

$$Z_{\text{new}} = Z_{\text{old}} + Sh_z \times X_{\text{old}}$$

In Matrix form, the above shearing equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ Z_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ Sh_y & 1 & 0 & 0 \\ Sh_z & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ Z_{\text{old}} \\ 1 \end{bmatrix}$$

3D Shearing Matrix

(In X axis)



• Shearing in Y Axis-

Shearing in Y axis is achieved by using the following shearing equations-

$$X_{\text{new}} = X_{\text{old}} + Sh_x \times Y_{\text{old}}$$

$$Y_{\text{new}} = Y_{\text{old}}$$

$$Z_{\text{new}} = Z_{\text{old}} + Sh_z \times Y_{\text{old}}$$

In Matrix form, the above shearing equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ Z_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & Sh_x & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & Sh_z & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ Z_{\text{old}} \\ 1 \end{bmatrix}$$

3D Shearing Matrix

(In Y axis)



• Shearing in Z Axis-

Shearing in Z axis is achieved by using the following shearing equations-

$$X_{\text{new}} = X_{\text{old}} + Sh_x \times Z_{\text{old}}$$

$$Y_{\text{new}} = Y_{\text{old}} + Sh_y \times Z_{\text{old}}$$

$$Z_{\text{new}} = Z_{\text{old}}$$

In Matrix form, the above shearing equations may be represented as-

$$\begin{bmatrix} X_{\text{new}} \\ Y_{\text{new}} \\ Z_{\text{new}} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & Sh_x & 0 \\ 0 & 1 & Sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{\text{old}} \\ Y_{\text{old}} \\ Z_{\text{old}} \\ 1 \end{bmatrix}$$

3D Shearing Matrix

(In Z axis)

Example



- Given a 3D triangle with points $(0, 0, 0)$, $(1, 1, 2)$ and $(1, 1, 3)$. Apply shear parameter 2 on X axis, 2 on Y axis and 3 on Z axis and find out the new coordinates of the object.





- **Solution-**

Given-

- Old corner coordinates of the triangle = A (0, 0, 0), B(1, 1, 2), C(1, 1, 3)
- Shearing parameter towards X direction (Sh_x) = 2
- Shearing parameter towards Y direction (Sh_y) = 2
- Shearing parameter towards Z direction (Sh_z) = 3



- **For Coordinates A(0, 0, 0)**

- Let the new coordinates of corner A after shearing = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the shearing equations, we have-

$$X_{\text{new}} = X_{\text{old}} = 0$$

$$Y_{\text{new}} = Y_{\text{old}} + Sh_y \times X_{\text{old}} = 0 + 2 \times 0 = 0$$

$$Z_{\text{new}} = Z_{\text{old}} + Sh_z \times X_{\text{old}} = 0 + 3 \times 0 = 0$$

Thus, New coordinates of corner A after shearing = $(0, 0, 0)$.



- **For Coordinates B(1, 1, 2)**

Let the new coordinates of corner B after shearing = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the shearing equations, we have-

$$X_{\text{new}} = X_{\text{old}} = 1$$

$$Y_{\text{new}} = Y_{\text{old}} + Sh_y \times X_{\text{old}} = 1 + 2 \times 1 = 3$$

$$Z_{\text{new}} = Z_{\text{old}} + Sh_z \times X_{\text{old}} = 2 + 3 \times 1 = 5$$

Thus, New coordinates of corner B after shearing = (1, 3, 5).



- **For Coordinates C(1, 1, 3)**

- Let the new coordinates of corner C after shearing = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the shearing equations, we have-

$$X_{\text{new}} = X_{\text{old}} = 1$$

$$Y_{\text{new}} = Y_{\text{old}} + Sh_y \times X_{\text{old}} = 1 + 2 \times 1 = 3$$

$$Z_{\text{new}} = Z_{\text{old}} + Sh_z \times X_{\text{old}} = 3 + 3 \times 1 = 6$$

Thus, New coordinates of corner C after shearing = (1, 3, 6).

Thus, New coordinates of the triangle after shearing in X axis = A (0, 0, 0), B(1, 3, 5), C(1, 3, 6).



- **Shearing in Y Axis-**
-
- **For Coordinates A(0, 0, 0)**

Let the new coordinates of corner A after shearing = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the shearing equations, we have-

$$X_{\text{new}} = X_{\text{old}} + Sh_x \times Y_{\text{old}} = 0 + 2 \times 0 = 0$$

$$Y_{\text{new}} = Y_{\text{old}} = 0$$

$$Z_{\text{new}} = Z_{\text{old}} + Sh_z \times Y_{\text{old}} = 0 + 3 \times 0 = 0$$

Thus, New coordinates of corner A after shearing = $(0, 0, 0)$.



- **For Coordinates B(1, 1, 2)**
-
- Let the new coordinates of corner B after shearing = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.
-
- Applying the shearing equations, we have-
- $X_{\text{new}} = X_{\text{old}} + Sh_x \times Y_{\text{old}} = 1 + 2 \times 1 = 3$
- $Y_{\text{new}} = Y_{\text{old}} = 1$
- $Z_{\text{new}} = Z_{\text{old}} + Sh_z \times Y_{\text{old}} = 2 + 3 \times 1 = 5$
-
- Thus, New coordinates of corner B after shearing = (3, 1, 5).



- **For Coordinates C(1, 1, 3)**

Let the new coordinates of corner C after shearing = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the shearing equations, we have-

$$X_{\text{new}} = X_{\text{old}} + Sh_x \times Y_{\text{old}} = 1 + 2 \times 1 = 3$$

$$Y_{\text{new}} = Y_{\text{old}} = 1$$

$$Z_{\text{new}} = Z_{\text{old}} + Sh_z \times Y_{\text{old}} = 3 + 3 \times 1 = 6$$

Thus, New coordinates of corner C after shearing = $(3, 1, 6)$.

Thus, New coordinates of the triangle after shearing in Y axis = A $(0, 0, 0)$, B $(3, 1, 5)$, C $(3, 1, 6)$.





- **Shearing in Z Axis-**
-
- **For Coordinates A(0, 0, 0)**

Let the new coordinates of corner A after shearing = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the shearing equations, we have-

$$X_{\text{new}} = X_{\text{old}} + Sh_x \times Z_{\text{old}} = 0 + 2 \times 0 = 0$$

$$Y_{\text{new}} = Y_{\text{old}} + Sh_y \times Z_{\text{old}} = 0 + 2 \times 0 = 0$$

$$Z_{\text{new}} = Z_{\text{old}} = 0$$

Thus, New coordinates of corner A after shearing = $(0, 0, 0)$.



- **For Coordinates B(1, 1, 2)**

Let the new coordinates of corner B after shearing = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the shearing equations, we have-

$$X_{\text{new}} = X_{\text{old}} + Sh_x \times Z_{\text{old}} = 1 + 2 \times 2 = 5$$

$$Y_{\text{new}} = Y_{\text{old}} + Sh_y \times Z_{\text{old}} = 1 + 2 \times 2 = 5$$

$$Z_{\text{new}} = Z_{\text{old}} = 2$$

Thus, New coordinates of corner B after shearing = $(5, 5, 2)$.



- **For Coordinates C(1, 1, 3)**

Let the new coordinates of corner C after shearing = $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$.

Applying the shearing equations, we have-

$$X_{\text{new}} = X_{\text{old}} + Sh_x \times Z_{\text{old}} = 1 + 2 \times 3 = 7$$

$$Y_{\text{new}} = Y_{\text{old}} + Sh_y \times Z_{\text{old}} = 1 + 2 \times 3 = 7$$

$$Z_{\text{new}} = Z_{\text{old}} = 3$$

Thus, New coordinates of corner C after shearing = $(7, 7, 3)$.

- Thus, New coordinates of the triangle after shearing in Z axis = A $(0, 0, 0)$, B $(5, 5, 2)$, C $(7, 7, 3)$.



Illumination and Shading



- **Lighting**
 - Phong illumination model
- **Shading**
 - Flat shading
 - Gouraud shading
 - Phong shading

Illumination Models:



- An illumination Model is a formula in variables associated with the surface properties and light conditions to calculate the intensity of light reflected from a point on a surface.

1. Ambient Light
2. Diffused Light
3. Specularly Reflected Light



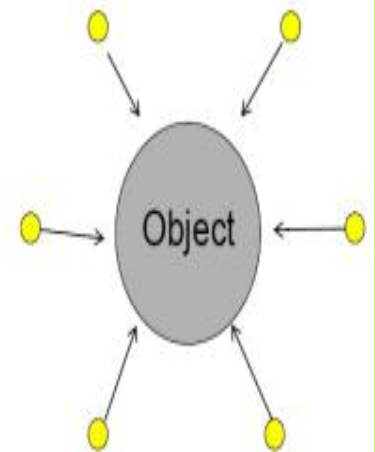
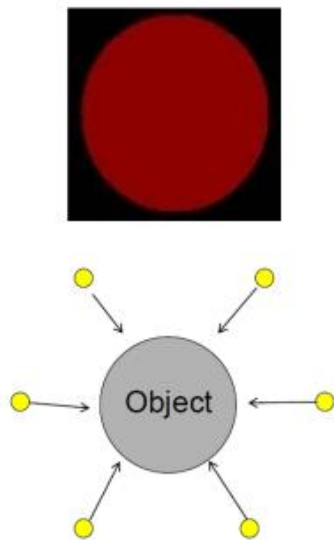
Ambient Light:



- An object may be visible even if it is not directly exposed to a light source. That is because some light is always scattered from the nearby illuminated objects and surroundings. This is called **Ambient light**. This light is diffused and non-directional and it is assumed to be incident with uniform intensity on all objects in a scene.



- Light reflected or scattered from other objects in the scene
- Environmental light
- Precise simulation of this is very hard!



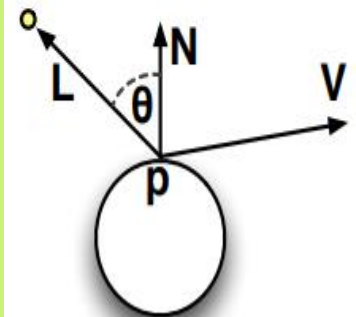
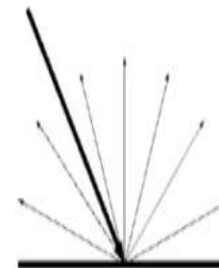
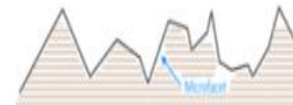
Diffused Light:



- Unlike in the case of ambient lighting where a distributed light source is assumed, intensity profile across surface changes when exposed to point light source. This is because light from the source is incident at a different angle at different points of the surface. The model which represents such diffused reflection is based on Lambert's cosine law and it is given by,
- $I_{\text{diff}} = I_1 k_d \cos\theta$



- When light hits an object with a
- rough surface, it is reflected in all directions
 - Amount of light hitting the surface depends on the angle between the normal vector and the incident vector of the incoming light.
 - The larger the angle (up to 90 degrees), the larger the area the incident light is spread over



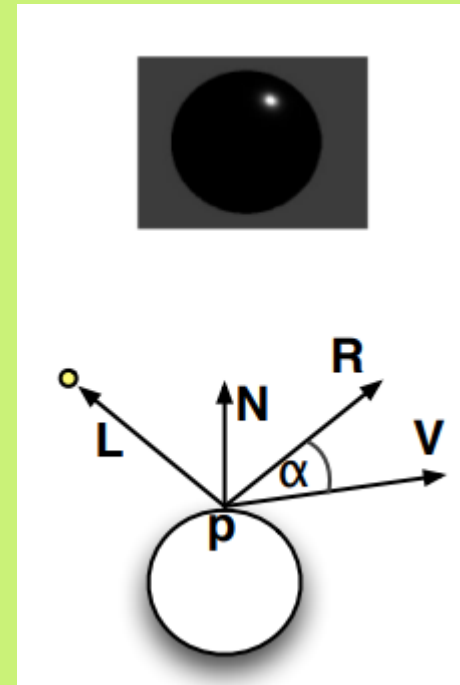
Specularly Reflected Light:



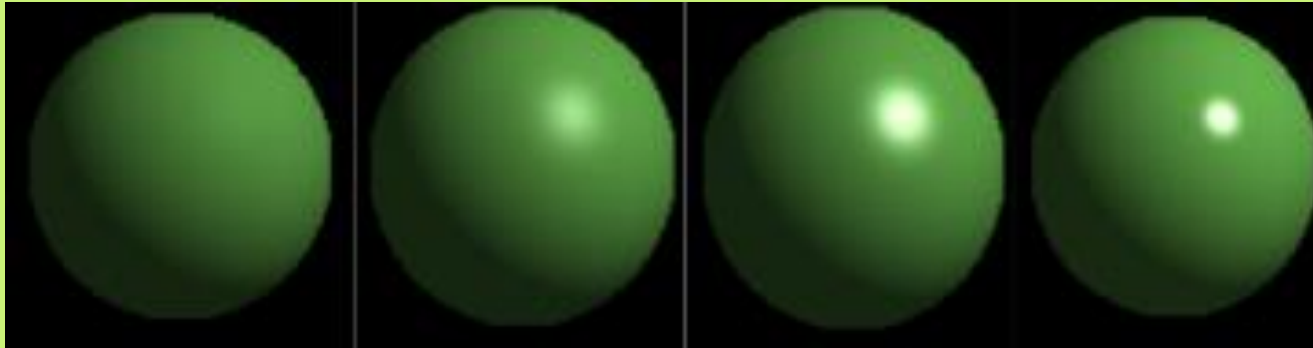
- Typical shiny surfaces like polished metal, oily skins, porcelain etc, that are neither diffuse reflector nor ideal reflector like reflect light specularly when exposed to point light source. Such reflection is identified by a highlight or bright spot on the surface when viewed from particular directions. Most interestingly such spots appear to move over the surface with the movement of the viewpoint. This is called **Specularly Reflected Light**.



- Direct reflections of the light source off of a shiny surface
- Smooth surfaces



Combining diffuse and specular reflection





- **Distance Factor:**
- We know that the intensity of light energy falls off as it travels farther from the light source. So, the intensity profile of two identical surfaces, under even lighting conditions, can't be identical if they are at different distances from the light source and/or viewpoint. A factor commonly is known as a distance factor.



- **Color Factor:**

- In the illumination models considering only monochromatic lights and surfaces. To model colour light and colored surfaces we can assume there basic components of incident light intensity, namely I_{RED} , I_{GREEN} and I_{BLUE} and each of the three coefficient of reflection (k_a , k_d , k_s) as three element (k_{RED} , k_{GREEN} and k_{BLUE}) vector. This is called as Color Factor. It means for a given surface we can have nine different characteristics reflection coefficients that one-to-one correspondence with similar color lights only.



- **Shading:**
- We have learnt how to calculate the intensity of light reflected from a point in a given lighting and surface condition. Now, we have to use a technique for finding the intensity of a surface as a whole to simulate its appearance under the given lighting condition. This technique is called as **Shading**.

Combined lighting models



- Combining ambient, diffuse and specular highlights gives the Phong Illumination model





Shading models

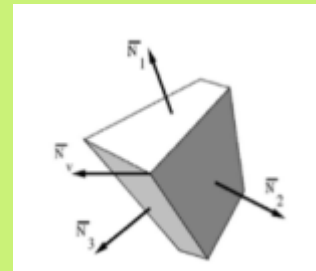
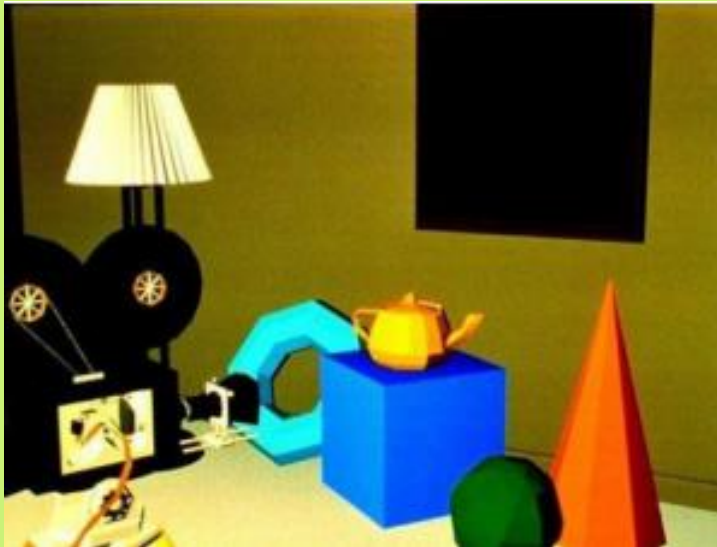


- Flat shading (one lighting calculation per polygon)
- Gouraud shading (one lighting calculation per vertex)
- Phong shading (one calculation per pixel)

Flat shading



- Colour is computed once for each polygon
- All pixels in a polygon are set to the same colour
- Works for objects made of flat faces



Flat shading

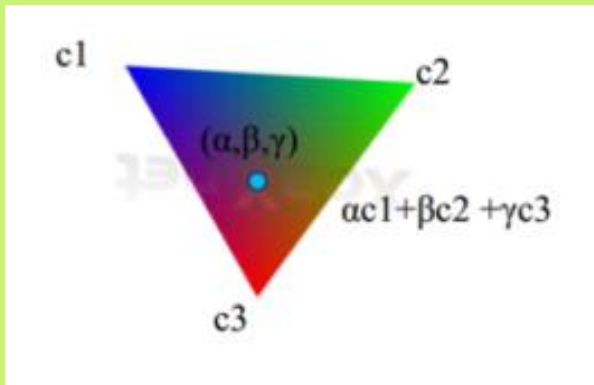


- Suffers from an effect called Mach banding
- Eyes are sensitive to sudden changes in brightness
- Artificial changes in brightness are introduced on either side of the boundary

Gouraud shading



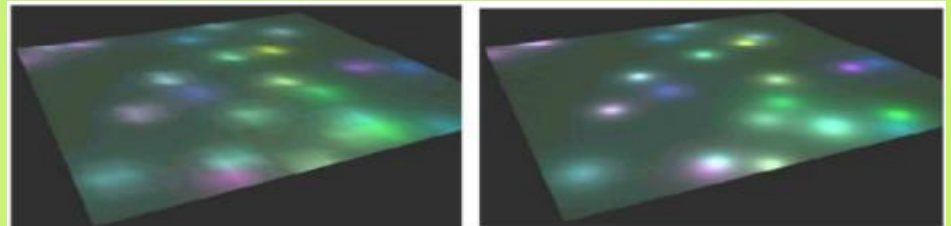
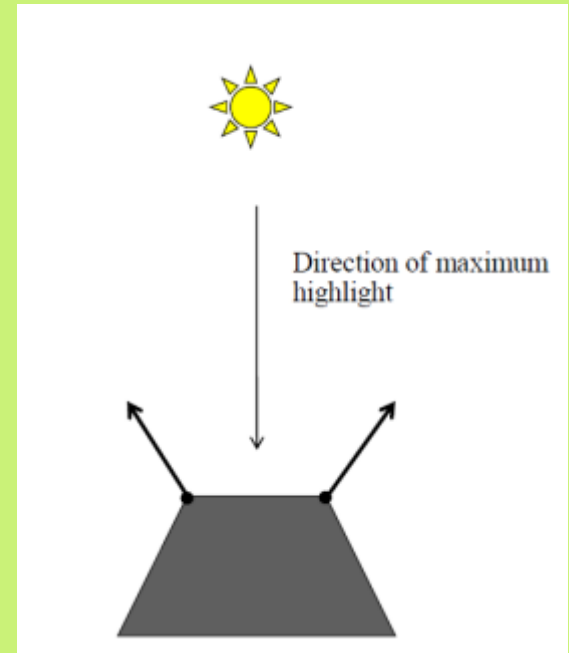
- Colour is computed once per vertex using the local illumination model
- Polygons interpolate colours over their surface



Problems with Gouraud shading



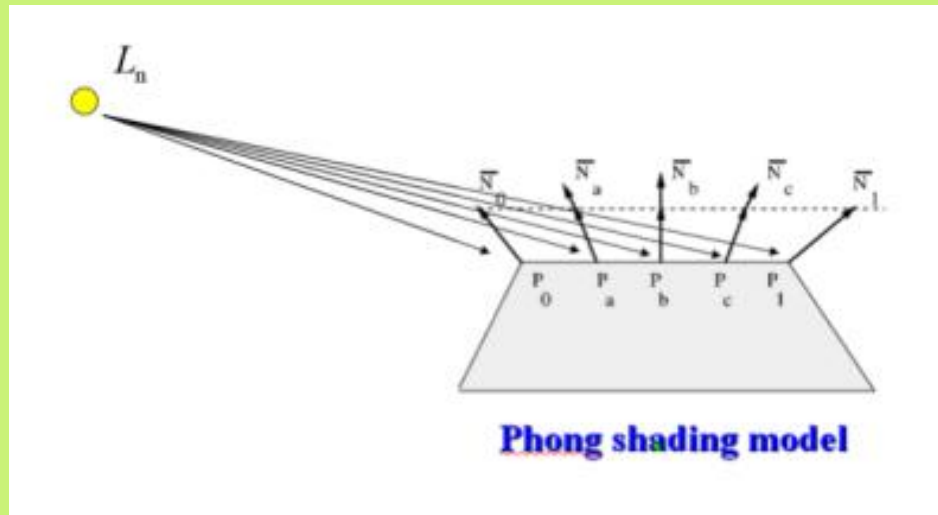
- In specular reflection the highlight can be sharp, depending on the shape of $\cos \alpha$
- Gouraud shading interpolates linearly and so can make the highlight much bigger
- Gouraud shading can miss highlights that occur in the middle of a polygon



Phong shading



- Lighting computation is performed at each pixel
- Normal vectors are interpolated over the polygon





- Phong shading Able to produce highlights that occur in the middle of a polygon

