Two Dimensional Geometric Transformations

Unit 3

Nipun Thapa (Computer Graphics)

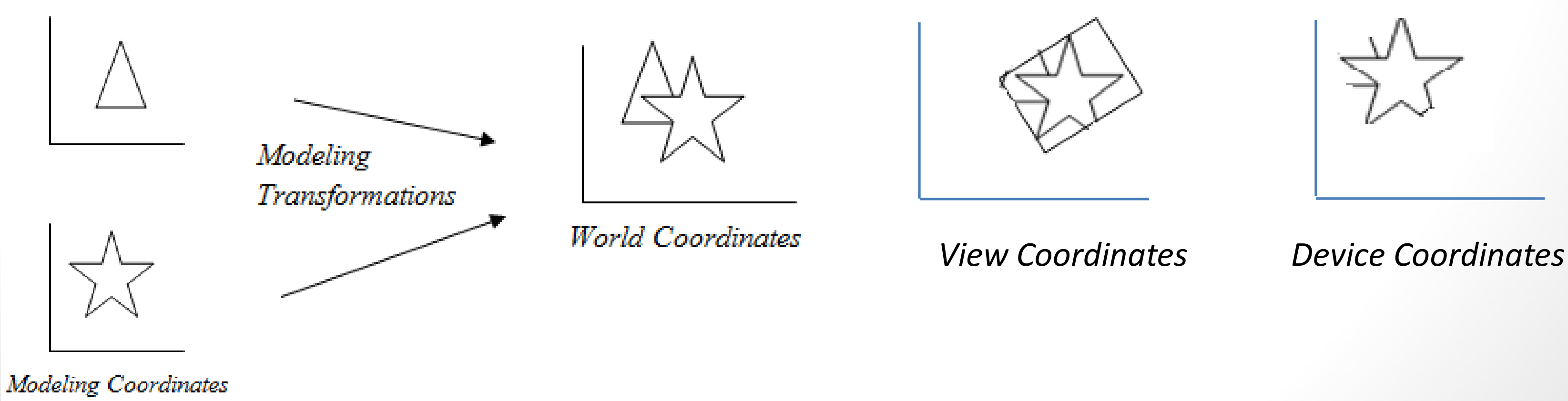
Two Dimensional Viewing

- Two-dimensional viewing is the mechanism for displaying views of a picture on an output device. Much like what we see in real life through a small window on the wall or the viewfinder of a camera, a computer-generated image often depicts a partial view of a large scene. For a 2-D picture, a view is selected by specifying a subarea of the total picture area.

2D Coordinate System

1. Modeling Coordinates

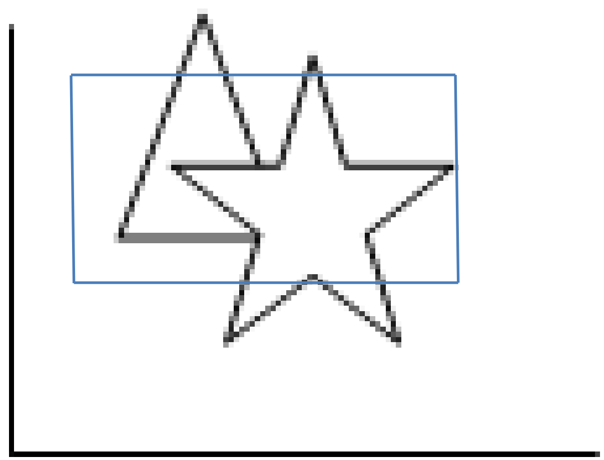
- Modeling coordinates are used to construct shape of individual parts (*objects or structures*) of a 2D scene. For example, generating a circle at the origin with a "radius" of 2 units.
- Here, origin (0, 0), and radius 2 units are modeling coordinates.
- **Modeling coordinates** define **object shape**.
- Can be floating-point, integers and can represent units like km, m, miles, feet etc.



2D Coordinate System..

2. World Coordinates

- ***World coordinates are used to organize the individual parts into a scene.***
- **World coordinates units define overall scene to be modeled**.
- World coordinates represent relative positions of objects.
- Can be floating-point, integers and can represent units like km, m, miles etc.

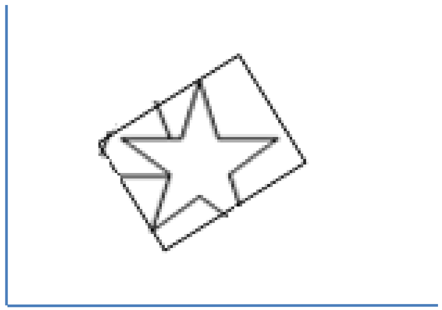


World Coordinates

2D Coordinate System..

3. Viewing Coordinates

- Viewing coordinates are used to define particular **view of the user.** Viewer‟s position and view angle i.e. rotated/translated.
- Viewing coordinates specify the portion of the output device that is to be used to present the view.
- ***Normalized viewing coordinates*** are viewing coordinates between 0 and 1 in each direction. They are used to make the viewing process independent of the output device (paper, mobile).

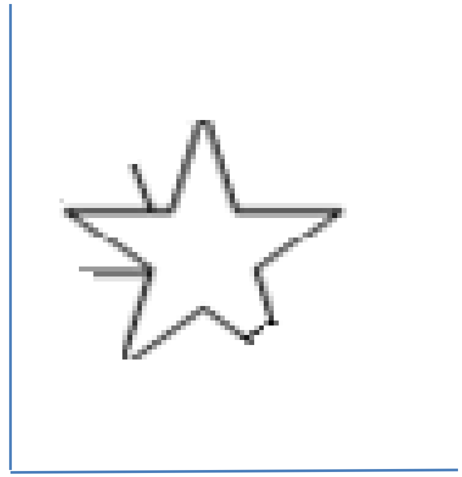


View Coordinates

2D Coordinate System..

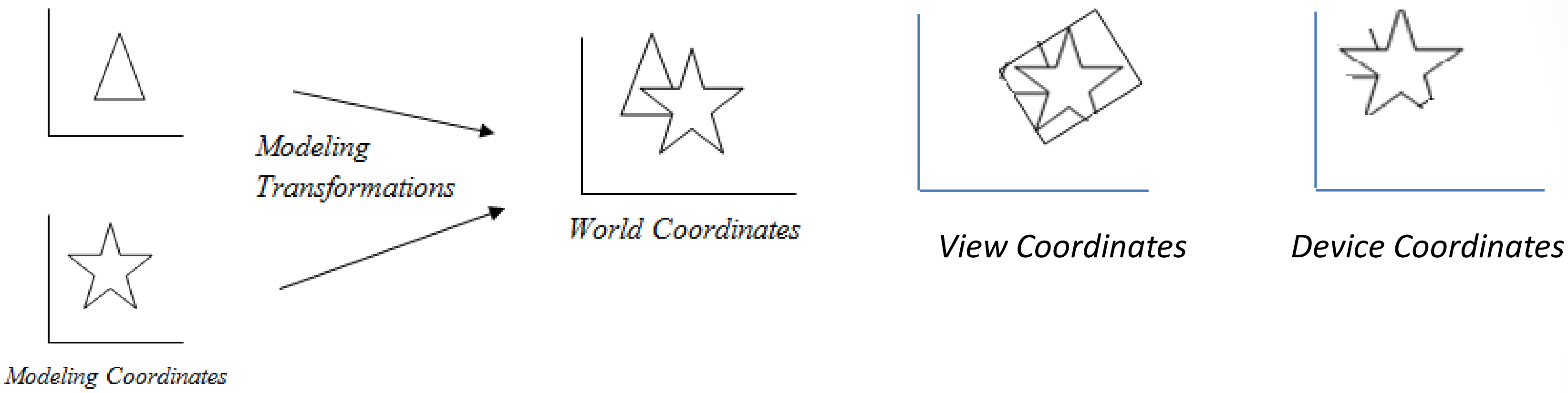
4. Device Coordinates or Screen Coordinates

- The display coordinate system is called device coordinate system. Device coordinates are specific to output device.
- Device coordinates are integers within the range (0, 0) to (xmax, ymax) for a particular output device.



Device Coordinates

2D Coordinate System..



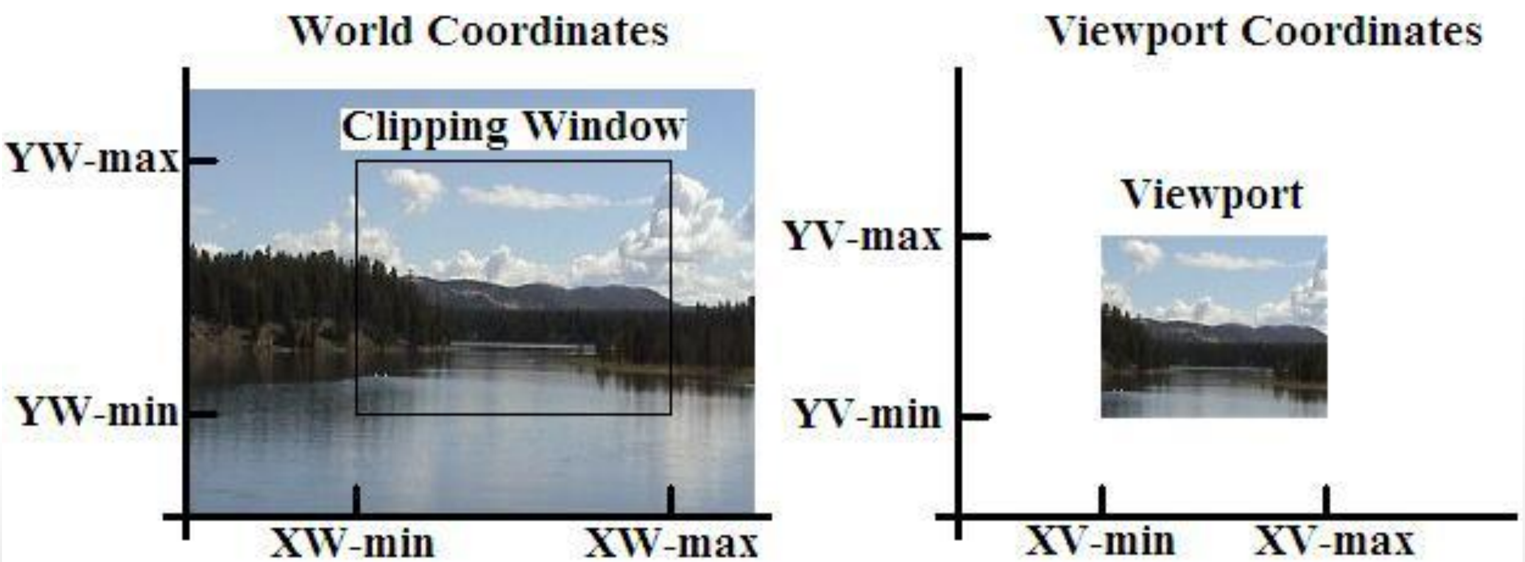
Window and Viewport

Window:

- A world-coordinate area selected for display is called a *window or clipping window*. That is, window is the section of the 2D scene that is selected for viewing.
- The window defines what is to be viewed.

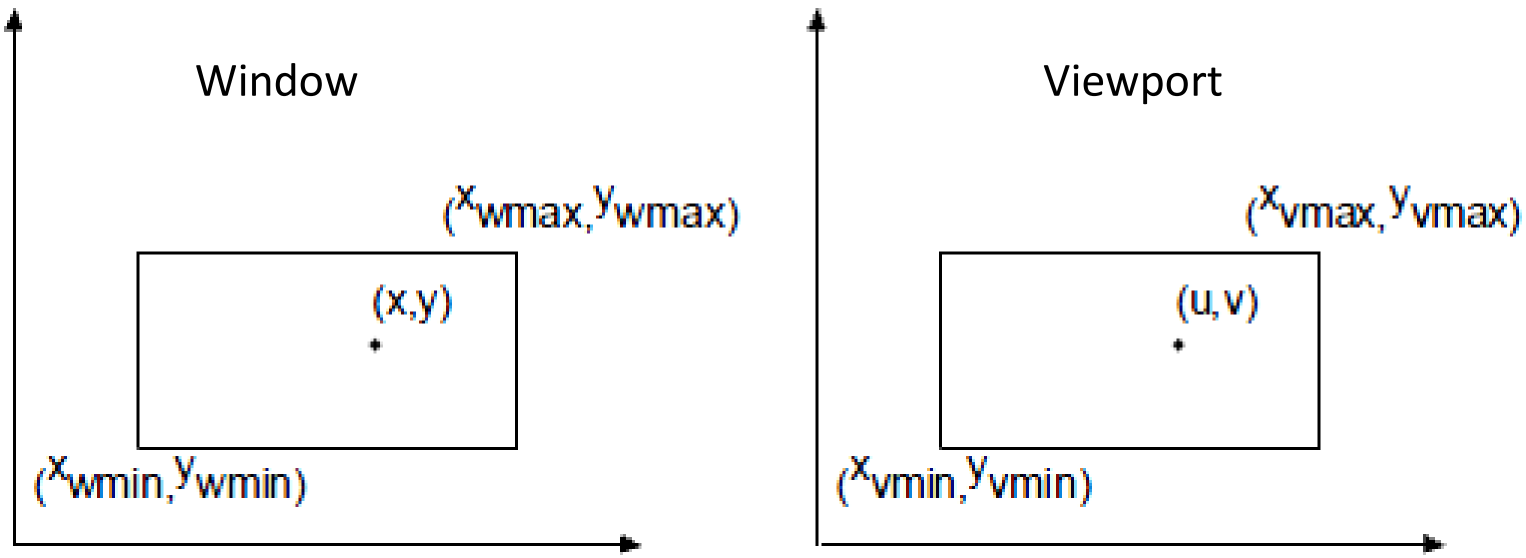
Viewport

- An area on a display device to which a window is mapped is called a viewport.
- The viewport indicates where on an output device selected part will be displayed.



Window to Viewport transformation

A window is specified by four world coordinates: X_{wmin}, X_{wmax}, Y_{wmin} and Y_{wmax} (see Fig.). Similarly, a viewport is described by four normalized device coordinates: X_{vmin}, X_{vmax}, Y_{vmin} and Y_{vmax}.



1. Translate the window to the origin. That is, apply T (-Xwmin, -Ywmin)
2. Scale it to the size of the viewport. That is, apply S (sx, sy)
3. Translate scaled window to the position of the viewport. That is, apply T(Xvmin, Yvmin).

Therefore, net transformation,

T_{WV} = T (Xvmin, Yvmin). S(sx, sy).T(-Xwmin, -Ywmin)

- Let (x, y) be the world coordinate point that is mapped onto the viewport point (u, v), then we must have

$$\frac{u - x_{v\min}}{x_{v\max} - x_{v\min}} = \frac{x - x_{w\min}}{x_{w\max} - x_{w\min}}$$

$$u = x_{vmin} + \frac{x_{v\max} - x_{v\min}}{x_{w\max} - x_{w\min}} \cdot (x - x_{wmin})$$

$$\frac{v - y_{v\min}}{y_{v\max} - y_{v\min}} = \frac{y - y_{w\min}}{y_{w\max} - y_{w\min}}$$

$$v = y_{vmin} + \frac{y_{v\max} - y_{v\min}}{y_{w\max} - y_{w\min}} \cdot (y - y_{wmin})$$

But, we know that

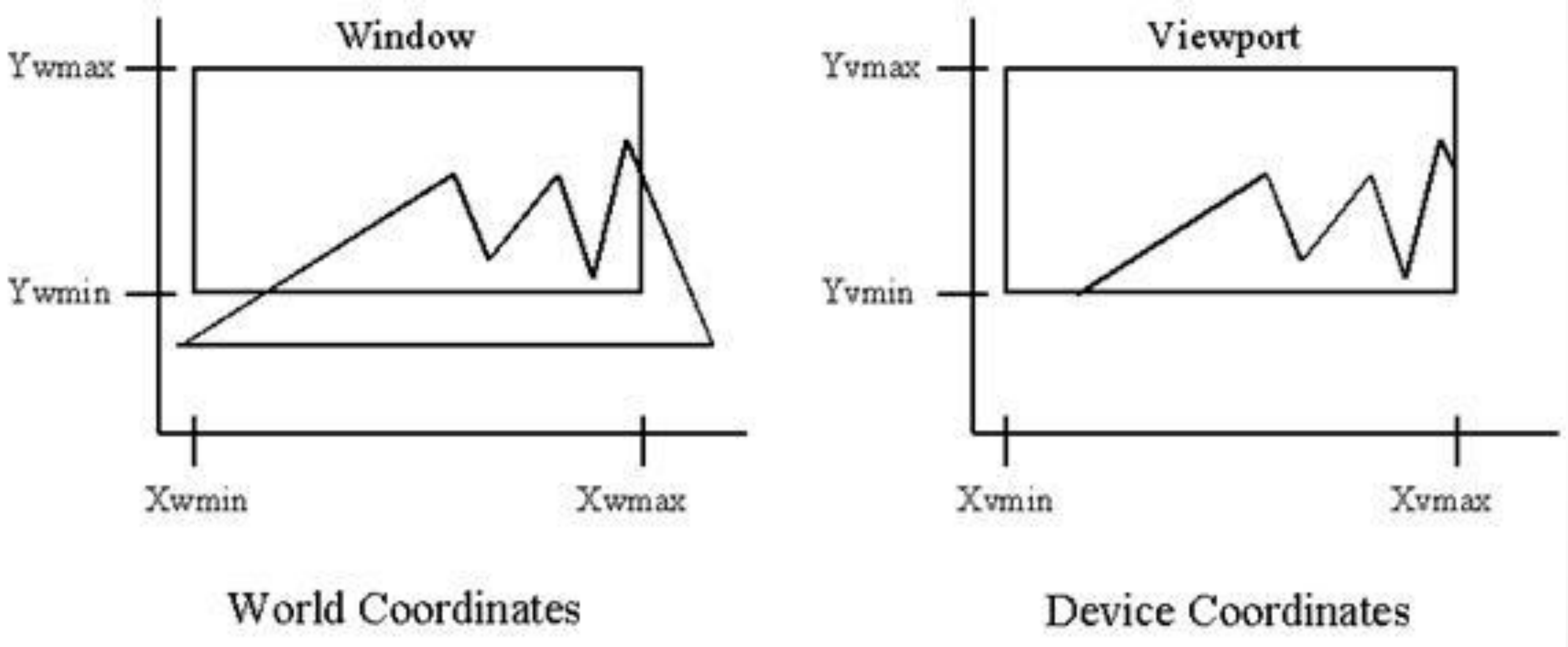
$$u = x_{vmin} + s_x(x - x_{wmin})$$
$$v = y_{vmin} + s_y(y - y_{wmin})$$

Where

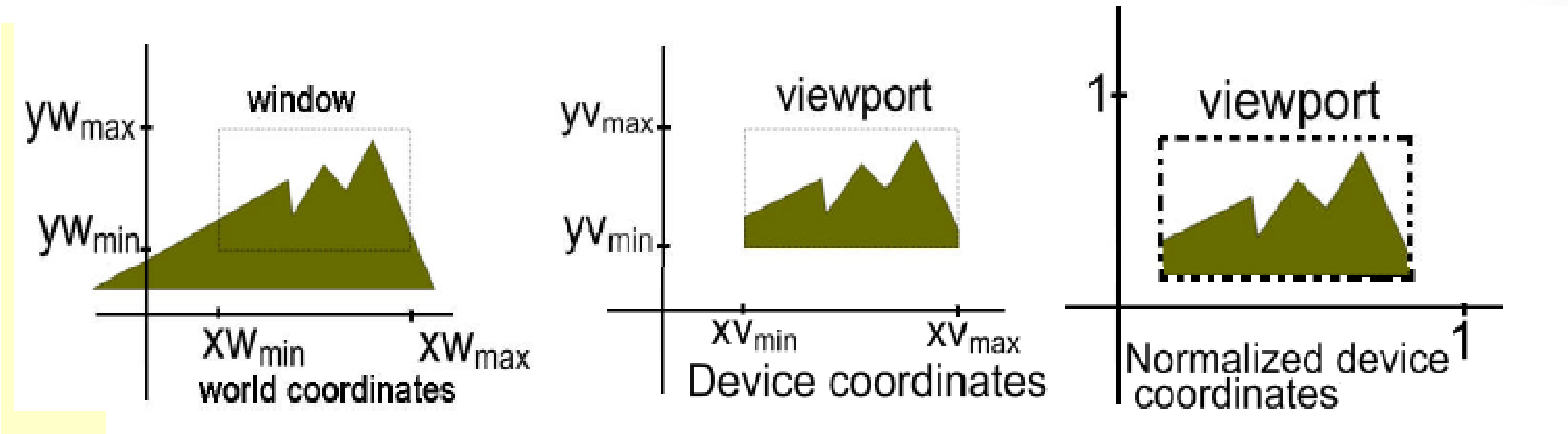
$$s_x = \frac{x_{v\max} - x_{w\min}}{x_{w\max} - x_{w\min}}$$

$$s_y = \frac{y_{v\max} - y_{v\min}}{y_{w\max} - y_{w\min}}$$

Window to Viewport transformation



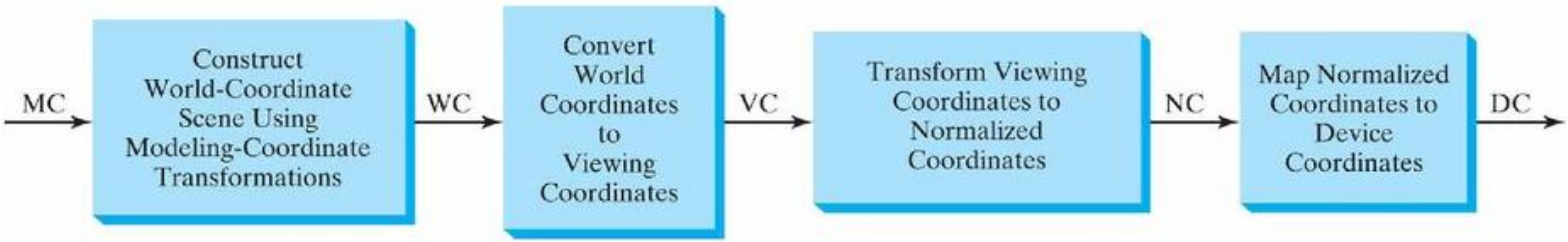
Window to Viewport transformation



Window and Viewport……

- *Window and viewport are often rectangular in standard positions, because it simplifies the transformation process and clipping process. Other shapes such as polygons, circles take longer time to process. By changing the position of the viewport, we can view objects at different positions on the display area of an output device. Also by varying the size of viewports, we can change size of displayed objects. Zooming effects can be obtained by successively mapping different-sized windows on a fixed-sized viewport.*

Two-Dimensional Viewing Pipeline



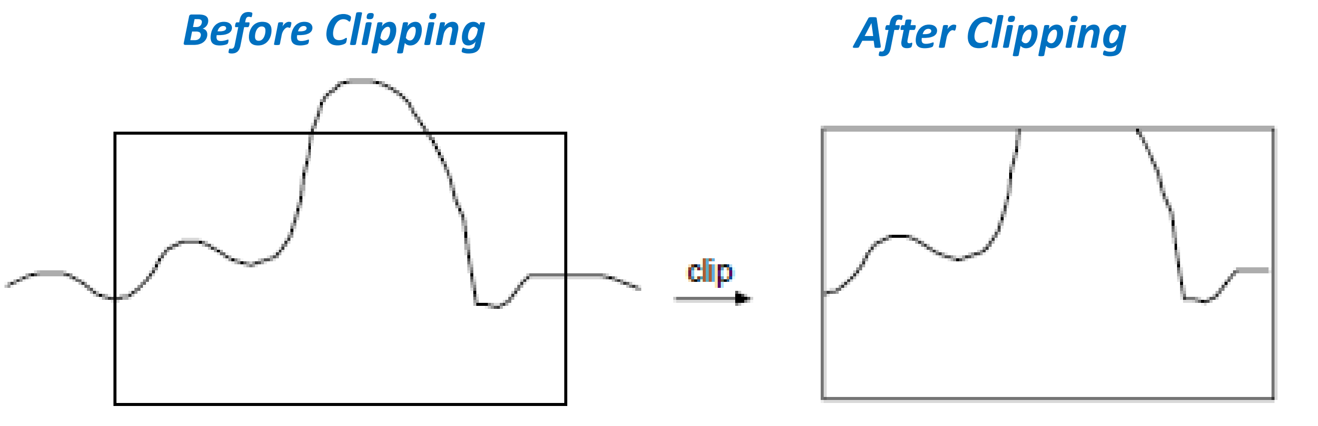
- The mapping of a part of a world co-ordinate scene to a device co-ordinate is referred to as viewing transformation.
- Sometimes, the 2D viewing transformation is simply referred to as the window–to–viewport transformation or the windowing transformation.
- The transformation that maps the window into the viewport is applied to all of the output primitives (lines, rectangles, circles) in world coordinates.
- This viewing transformation involves several steps.

Clipping

- The process of identifying those portions of a picture that are either inside or outside of the specified region of space is called *clipping*.

Two possible ways to apply clipping in the viewing transformation:

1. Apply clipping in the world coordinate system: ignore objects that lie outside of the window.
2. Apply clipping in the device coordinate system: ignore objects that lie outside of the viewport.



Clipping..

***Why Clipping*?**

1. Excludes unwanted graphics from the screen.
2. Improves efficiency, as the computation dedicated to objects that appear off screen can be significantly reduced.

***Applications*:**

1. In drawing and painting, it is used to select picture parts for copying, erasing, moving.
2. Displaying a multi-window environment.

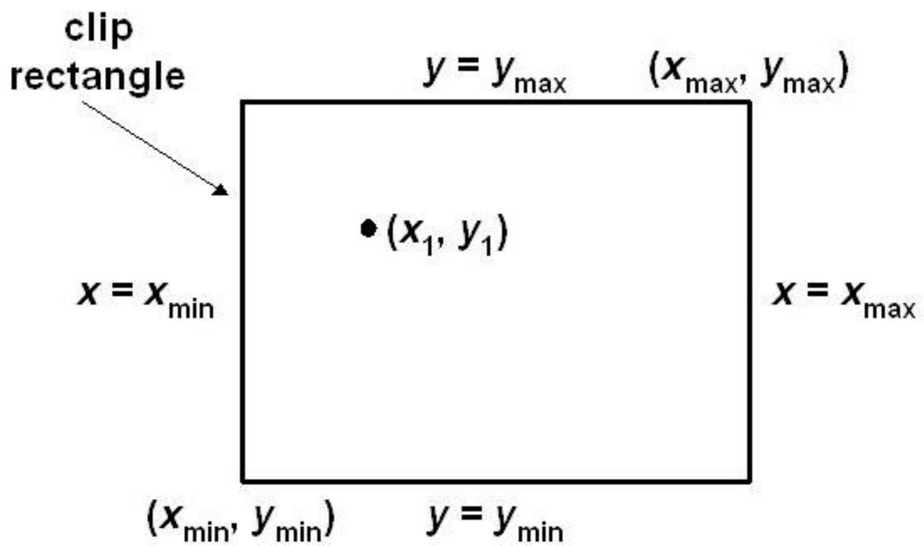
Point Clipping

- Let W denote a clip window with coordinates (xwmin, ywmin), (Xwmin, Ywmax), (Xwmax, Ywmin), (Xwmax, Ywmax), then a vertex (x, y) is displayed only if following "point clipping" inequalities are satisfied:

$$Xwmin \le x \le Xwmax,$$

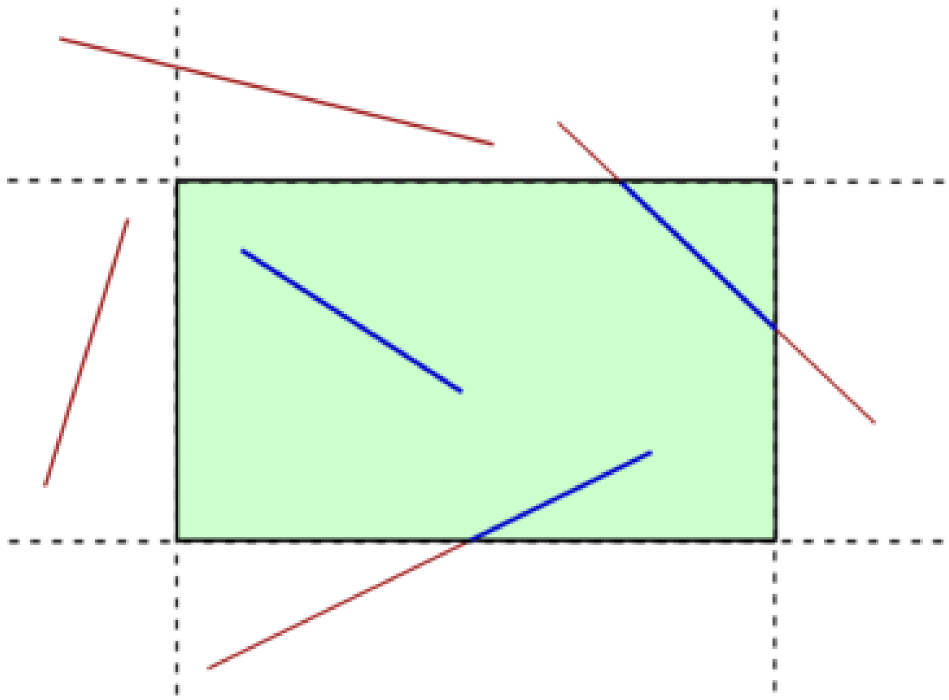
$$Ywmin \le y \le Ywmax.$$

- Very simple and efficient.
- Only works for vertices.



Line Clipping

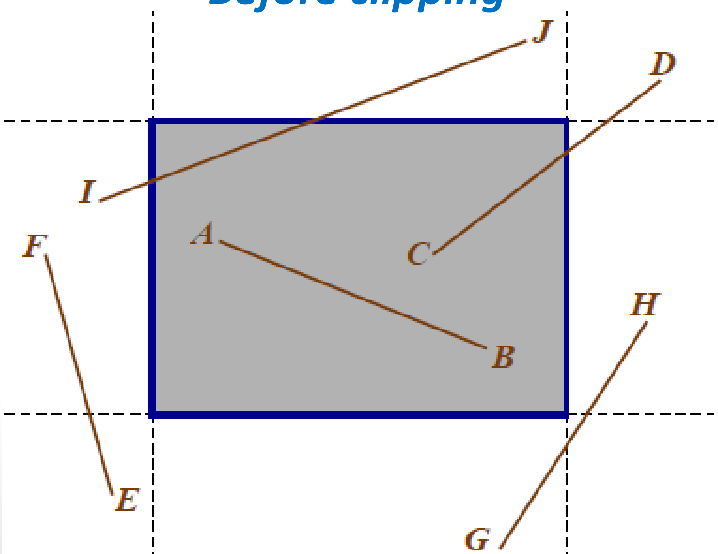
- Lines that do not intersect the clipping window are either completely inside the window or completely outside the window. On the other hand, a line that intersects the clipping window is divided by the intersection point(s) into segments that are either inside or outside the window.



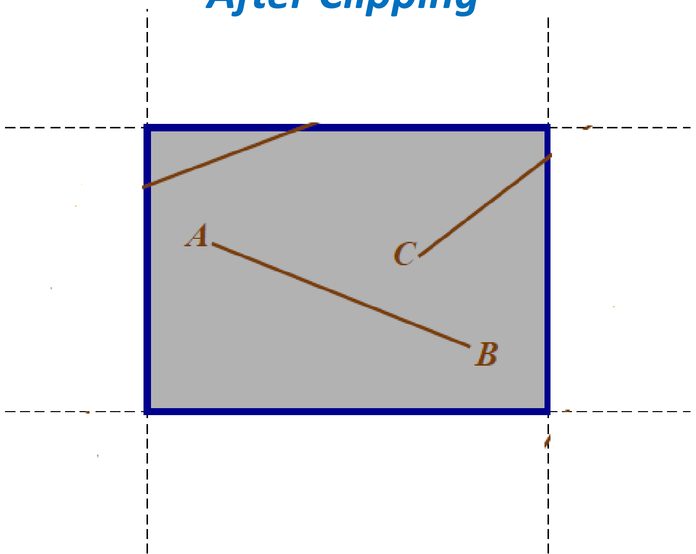
For any particular line segment:

a. Both endpoints are inside the region (line ***AB***).
   - No clipping necessary.
b. One endpoint is inside and one is outside of the clipping window (line ***CD***).
   - Clip at intersection point.
c. Both endpoints are outside the region:
   - No intersection (lines ***EF***, ***GH***)
   - Discard the line segment.
   - Line intersects the region (line ***IJ***)
   - Clip line at both intersection points.

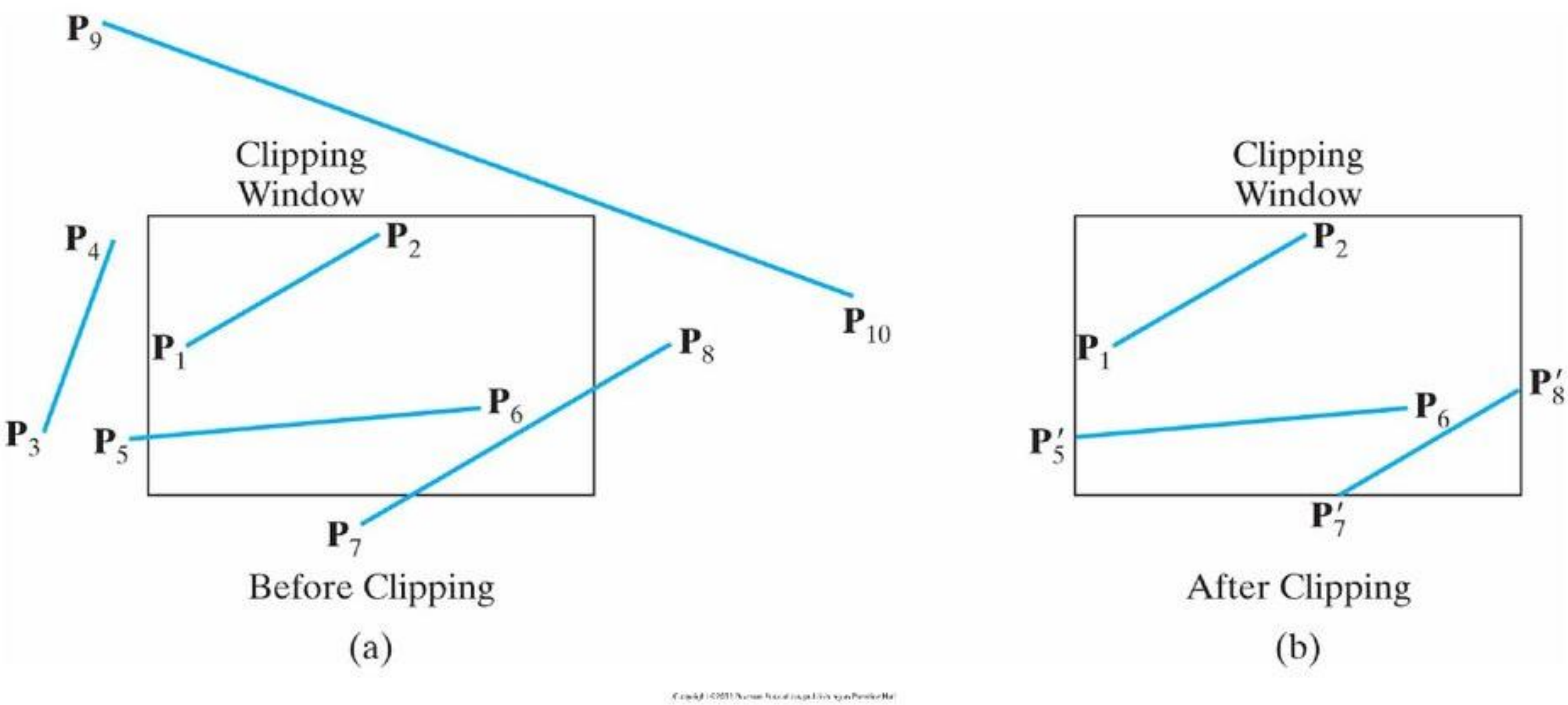
Before clipping



After Clipping

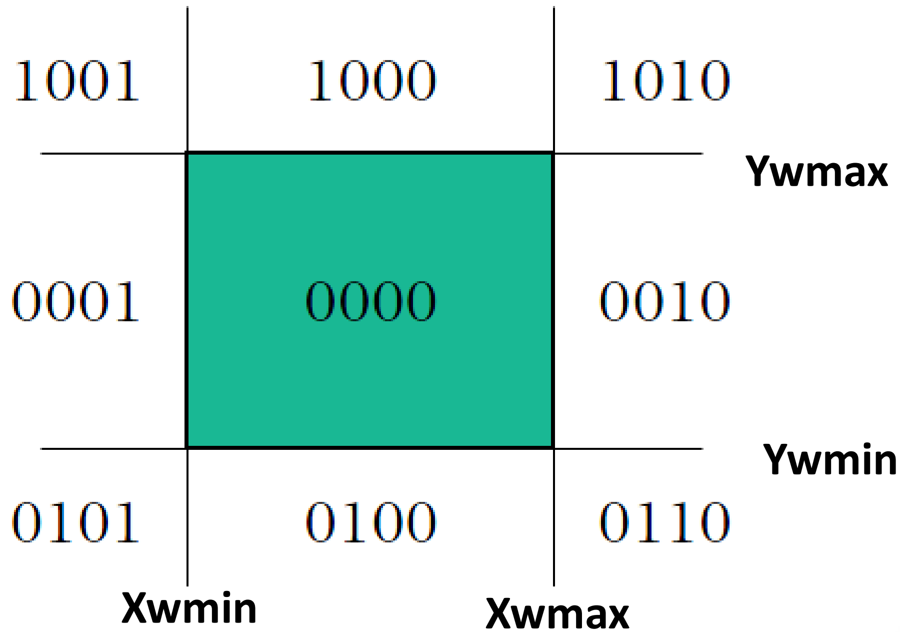


Line Clipping…



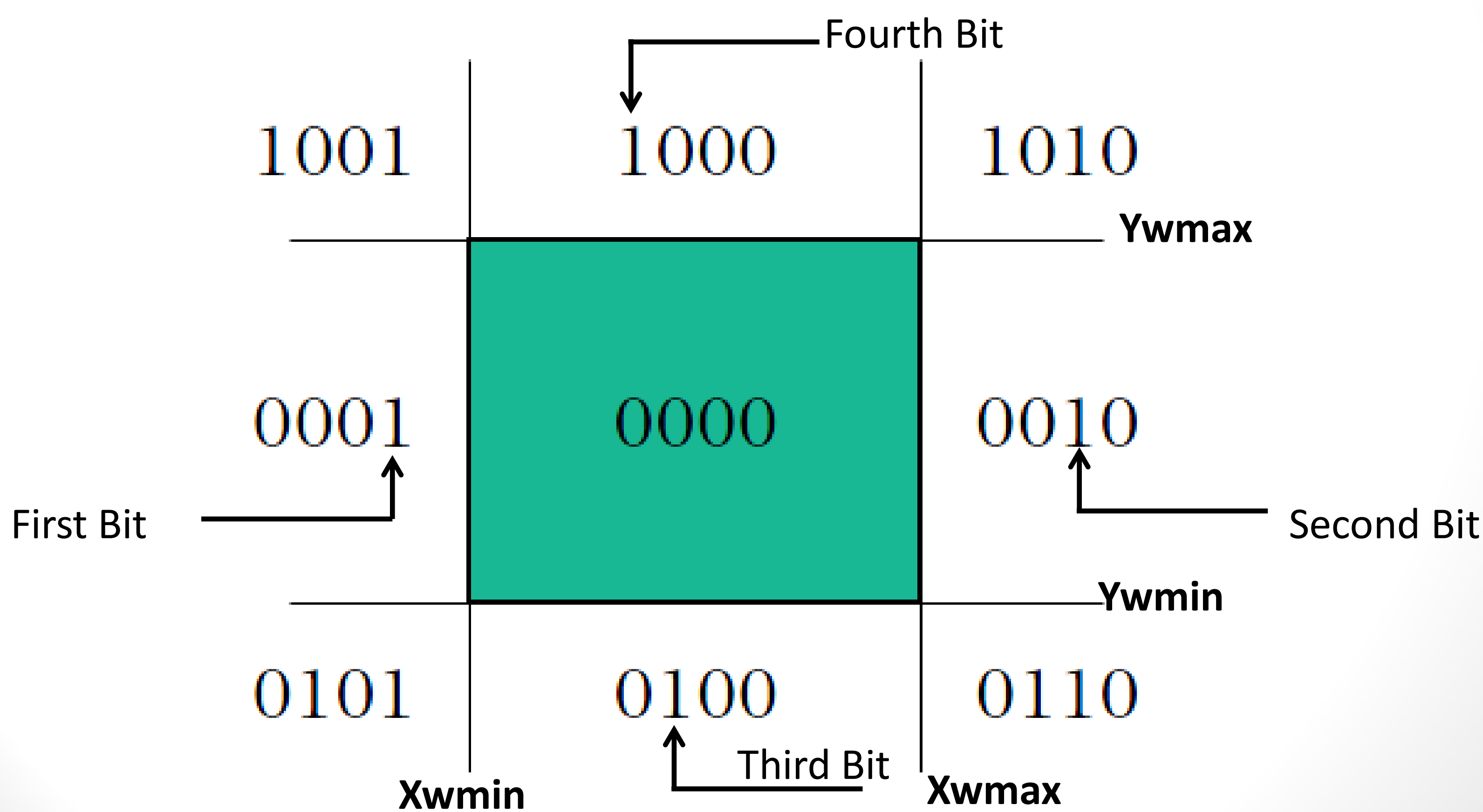
Cohen-Sutherland Line Clipping Algorithm

- One of the oldest and most popular line-clipping algorithms. In this method, coordinate system is divided into nine regions. All regions have their associated region codes. Every line endpoint is assigned a four digit binary code (*region code or out code*). Each bit in the code is set to either a 1(true) or a 0(false). Assign a bit pattern to each region as shown:



Step 1 : Establish Region code for all line end point

- First bit is 1, if x < Xwmin (Point lies to left of window) , else set it to 0
- Second bit is 1, if x > Xwmax (Point lies to right of window) , else set it to 0
- Third bit is 1, if y < Ywmin (Point lies to below window), else set it to 0
- Fourth bit is 1, if y > Ywmax (Point lies to above window) , else set it to 0



Step 2: Determine which lines are completely inside window and which are not

a) If both end points of line has region codes ‘0000’ line is completely inside window.

b) If logical AND operation of region codes of two end points is NOT ‘0000’. The line is completely outside (some bit position have 1’s)

Astart 0000
Aend 0000

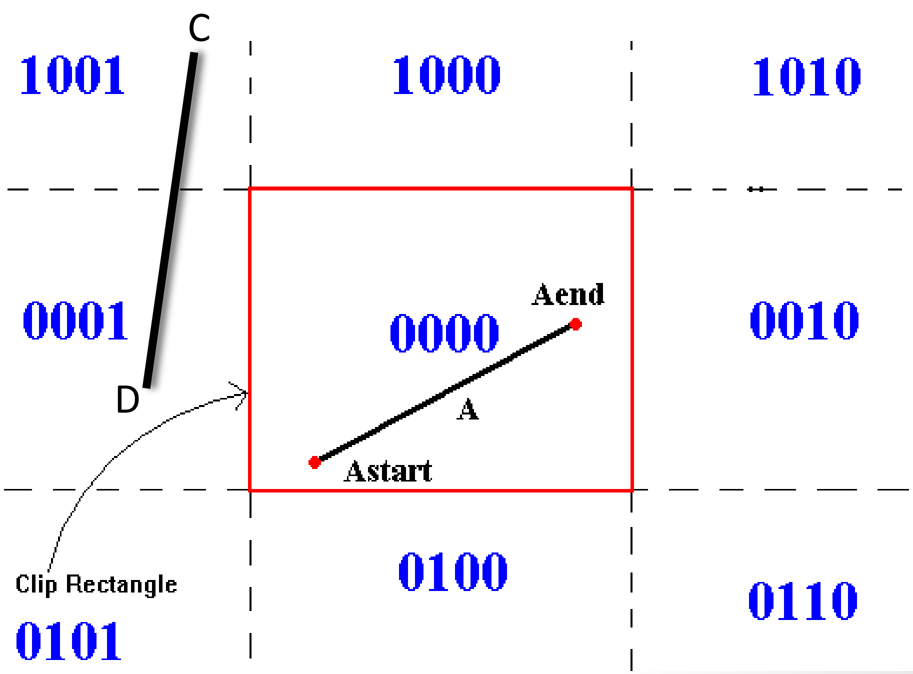
Completely inside

C 1001
D 0001

AND

0001

Completely Outside



Step 3 : If both tests fail then line is partially visible so we need to find the intersection with boundaries of window .

a) If 1st bit is 1 then line intersect with **Left** boundary and

- $Y_i = Y_1 + m (X - X_1)$ *where* X= Xwmin

b) If 2nd bit is 1 then line intersect with **Right** boundary and

- $Y_i = Y_1 + m (X - X_1)$ *where* X= Xwmax

c) If 3rd bit is 1 then line intersect with **Bottom** boundary and

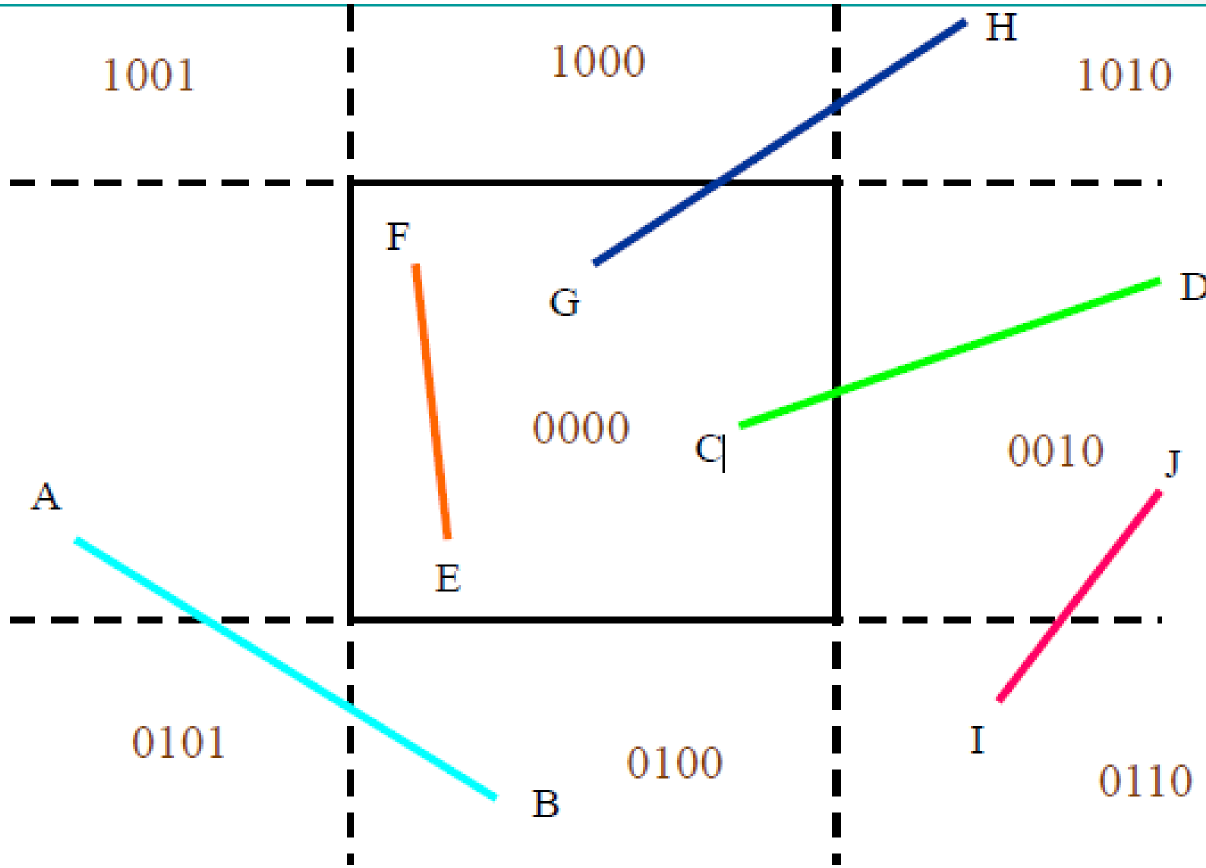
- $X_i = X_1 + (1/m) (Y - Y_1)$ *where* Y= Ywmin

d) If 4th bit is 1 then line intersect with **Top** boundary and

- $X_i = X_1 + (1/m) (Y - Y_1)$ *where* Y= Ywmax

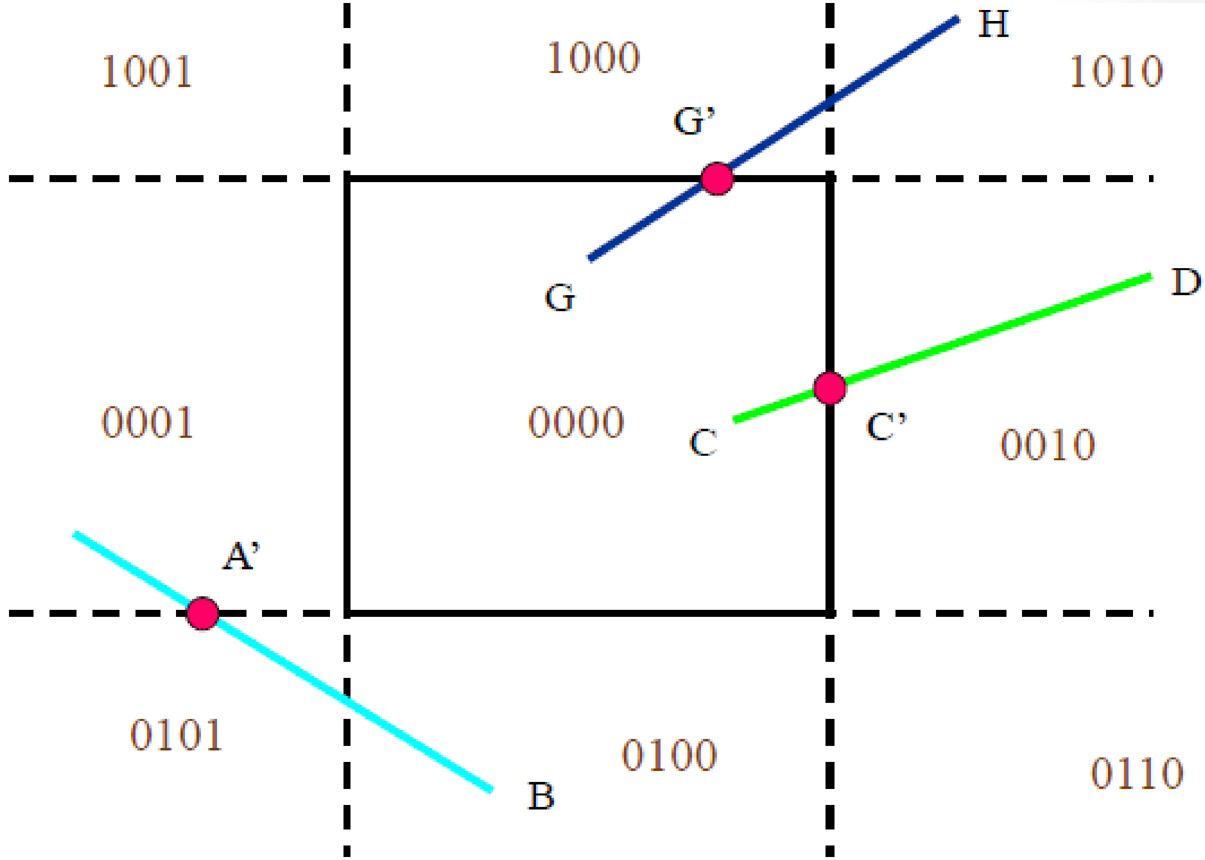
Here, (Xi , Yi) are (X ,Y) intercepts for that the step 4 repeat step 1 through step 3 until line is completely accepted or rejected

Cohen-Sutherland Line Clip Examples



A	0001	C	0000	E	0000	G	0000	I	0110
B	0100	D	0010	F	0000	H	1010	J	0010
OR	0101	OR	0010	OR	0000	OR	1010	OR	0110
AND	0000	AND	0000	AND	0000	AND	0000	AND	0010
clip		**clip**		**accept**		**clip**		**reject**	

Cohen-Sutherland Line Clip Examples



remove A'B	A 0001 A' 0001 OR 0001 AND 0001 **reject**	**remove** C'D	C 0000 C' 0000 OR 0000 AND 0000 **accept**	**remove** G'H	G 0000 G' 0000 OR 0000 AND 0000 **accept**

Numerical

Q. Clip a line with end point A(5,30) , B(20,60) against a clip window with lower most left corner at P1(10,10) and upper right corner at P2(100,100)

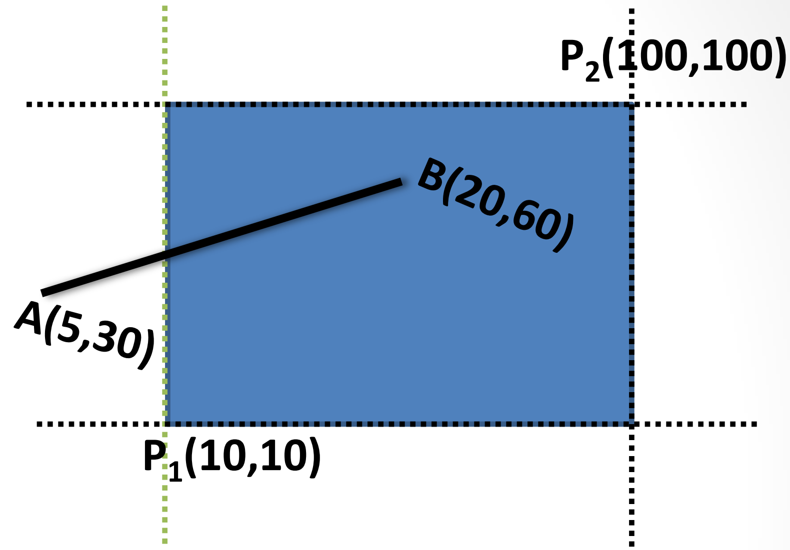
Ans:

Step 1: Establish the region

codes for end point A,B

A(5,30)
5<10 : (true)= 1
5>100 : (false) =0
30<10 : (false) =0
30>100 : (false) =0
A(5,30) = 0001

B(20,60)
20<10: (false) =0
20>100: (false)=0
60<10 : (false)=0
60>100: (false)=0
B(20,60)=0000



Step 2: Visibility check

A: 0001
B: 0000
AND
0000

Partial Visibility

Step 3: Intersection point with boundary

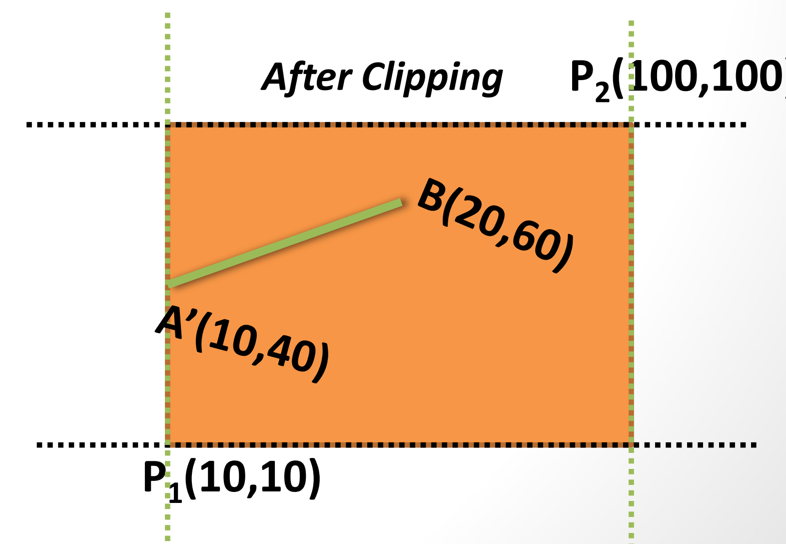
A: 0001 , Cond^n of 1$^{st\ bit}$ is 1

$Y_i = Y_1 + m\,(X - X_1)$ *where ,* X= 10

= 30 + 2 (10-5)

= 40

(X,Y)=(10,40)



Q. Clip a line with end point A(5,15), B(110,60) against a clip window with lower most left corner at P1(10,10) and upper right corner at P2(100,100)

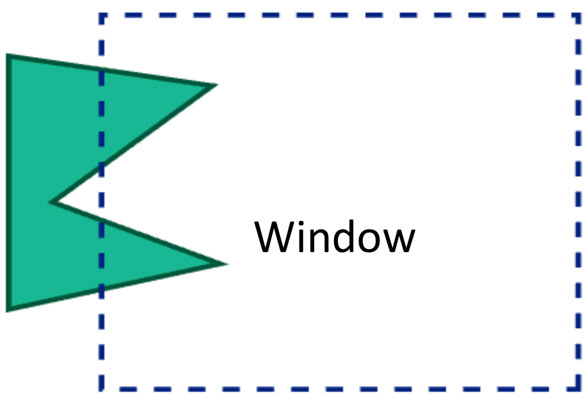
Q. Given a clipping window P (30,0), S(0,20) use the Cohen Sutherland algorithm to determine the visible portion of the line A(10,30) and B(40,0).

Q. . Given a clipping window P(0,0), Q(10,20), S(0,20) use the Cohen Sutherland algorithm to determine the visible portion of the line A(10,30) and B(40,0).

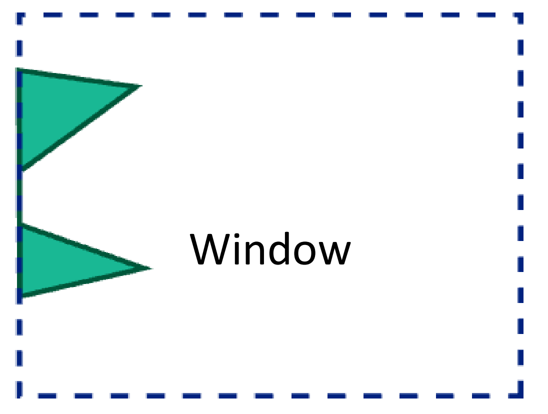
Q. Clip a line with end point A(1,60) , B(20,120) against a clip window with lower most left corner at P1(10,10) and upper right corner at P2(100,100)

Polygon Clipping

- A polygon can be defined as a geometric object "consisting of a number of points (called vertices) and an equal number of line segments (called sides or edges). Polygon clipping is defined as the process of removing those parts of a polygon that lie outside a clipping window. Consider a general polygon that is clipped with respect to a rectangular viewing region.



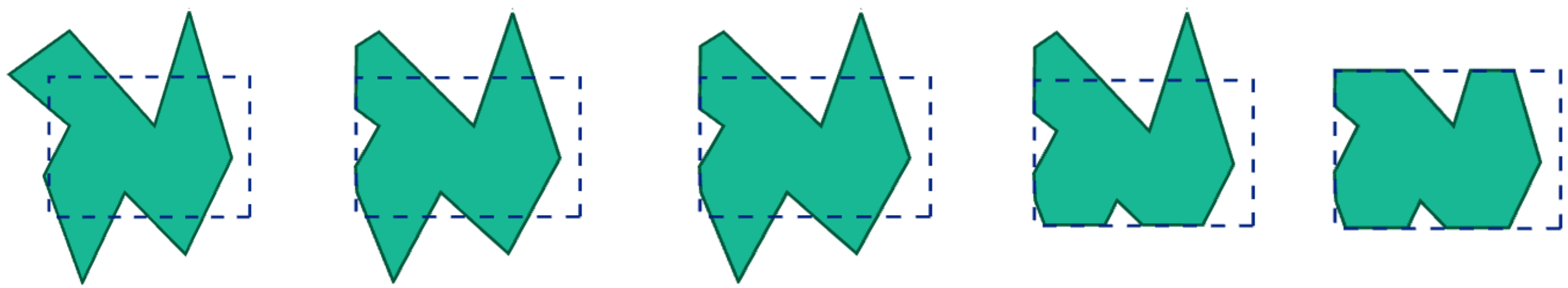
Before clipping



After clipping

Sutherland-Hodgeman Polygon Clipping Algorithm

- The Sutherland-Hodgeman Polygon-Clipping Algorithms clips a given polygon successively against the edges of the given clip-rectangle. It starts with the initial set of polygon vertices, first clips the polygon against the left rectangle boundary of the window. Then successively against the right boundary, bottom boundary, and finally against the top boundary, as shown in figure.



Original Polygon　　Clip Left　　Clip Right　　Clip Bottom　　Clip Top

2D Geometric Transformation

Unit 2

Introduction

Basic Transformations

The orientation, size, and shape of the output primitives are accomplished with geometric transformations that alter the coordinate descriptions of objects. The basic geometric transformations are **translation, rotation, and scaling**. Other transformations that are often applied to objects include **reflection and shear**. In these all cases we consider the reference point is origin so if we have to do these transformations about any point then we have to shift these point to the origin first and then perform required operation and then again shift to that position.

Introduction

Basic Transformations

1. Translation
2. Rotation
3. Scaling
4. Reflection
5. shearing

HOMOGENOUS FORM

Expressing position in homogeneous coordinates allows us to represent all geometric transformations equation as matrix multiplications.

$$(x, y) \text{ -------------------- } (x_h, y_h ,h)$$

$x= x_h /h \quad y= y_h /h$

where h is any non zero value

For convenient h=1

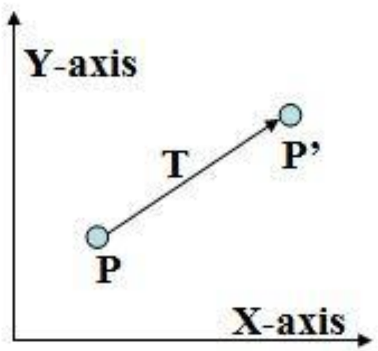
$$(x,y) \text{-------------------------} (x, y, 1).$$

1. Translation

A translation is applied to an object by repositioning it along a straight-line path from one coordinate location to another. We translate a two-dimensional point by adding translation distances, tx, and ty, to the original coordinate position (x, y) to move the point to a new position (x ' , y').

x’ = x + tx

y’ = y + ty where the pair (tx, ty) is called the ***translation vector*** or ***shift vector***.



*Translating a point from position **P** to position **P'** With translation vector **T**.*

1. Translation

We can write equation as a single matrix equation by using column vectors to represent coordinate points and translation vectors. Thus,

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

$$P' = P + T$$

So we can write

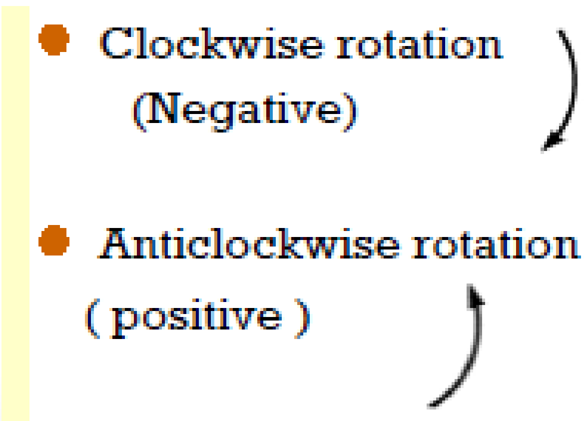
In homogeneous representation if position P = (x, y) is translated to new position P'= (x', y') then:

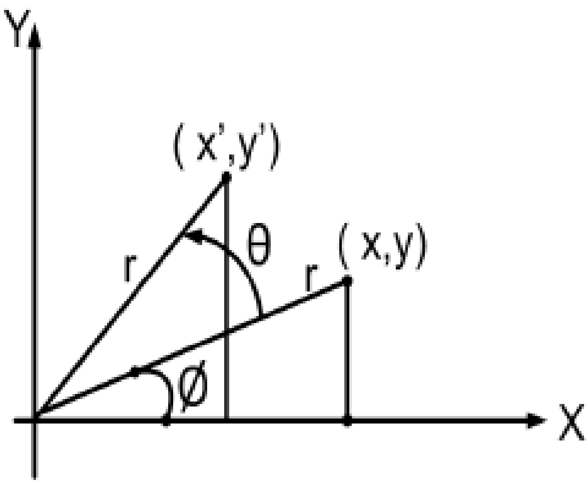
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow P' = T(t_x, t_y).P$$

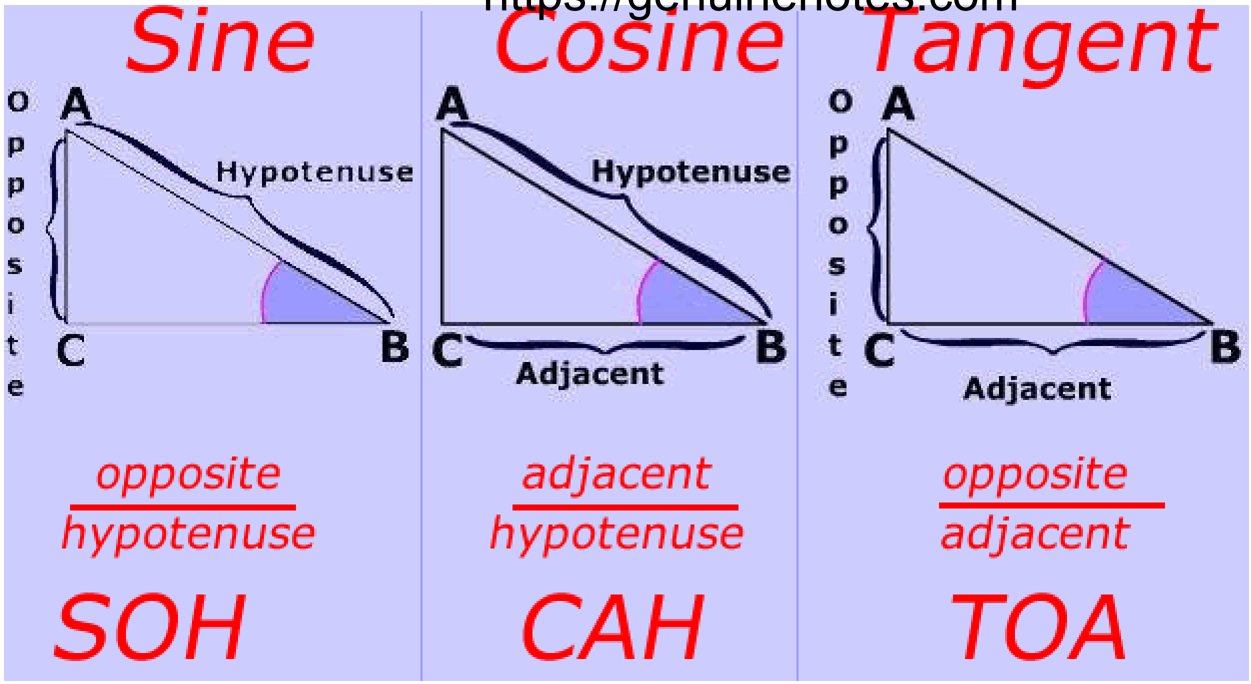
2. ROTATION

A two-dimensional rotation is applied to an object by repositioning it along a circular path in the **xy** plane. To generate a rotation, we specify a rotation angle θ and the position (x_r, y_r) of the rotation point (or pivot point) about which the object is to be rotated.

+ Value for 'θ' define *counter-clockwise* rotation about a point
- Value for 'θ' defines *clockwise* rotation about a point





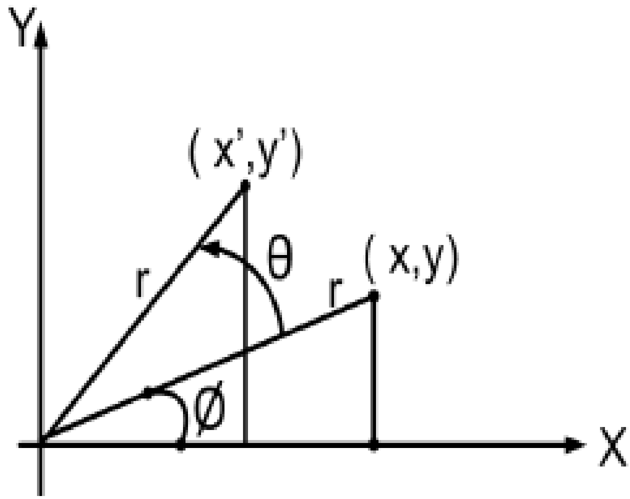


$$
\begin{aligned}
\sin(\alpha+\beta) &= \sin\alpha\ \cos\beta + \cos\alpha\ \sin\beta \\
\cos(\alpha+\beta) &= \cos\alpha\ \cos\beta - \sin\alpha\ \sin\beta \\
\sin(\alpha-\beta) &= \sin\alpha\ \cos\beta - \cos\alpha\ \sin\beta \\
\cos(\alpha-\beta) &= \cos\alpha\ \cos\beta + \sin\alpha\ \sin\beta
\end{aligned}
$$

Anticlockwise direction

2. ROTATION

At Origin



Coordinates of point (x,y) in polar form

$$x = r\cos\phi, \qquad y = r\sin\phi$$

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

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

$$P' = R\,.\,P$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \mathrm{Cos}\theta & -\mathrm{Sin}\theta \\ \mathrm{Sin}\theta & \mathrm{Cos}\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

ROTATION MATRX

2. ROTATION

In homogeneous co-ordinate

Anticlockwise direction

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow P' = R(\theta).P$$

Clockwise direction

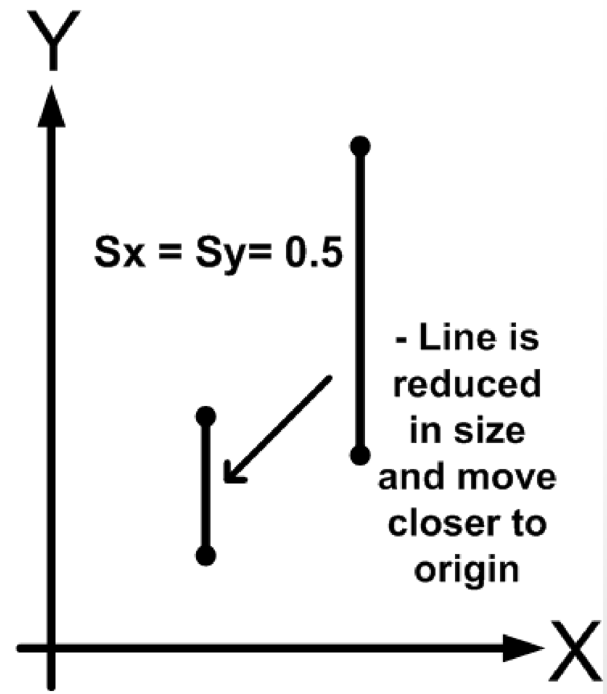
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow P' = R(\theta).P$$

3.Scaling

A scaling transformation alters the size of an object. This operation can be carried out for polygons by multiplying the coordinate values (x, y) of each vertex by scaling factors $\mathbf{S_x}$ and $\mathbf{S_y}$ to produce the transformed coordinates (x', y').

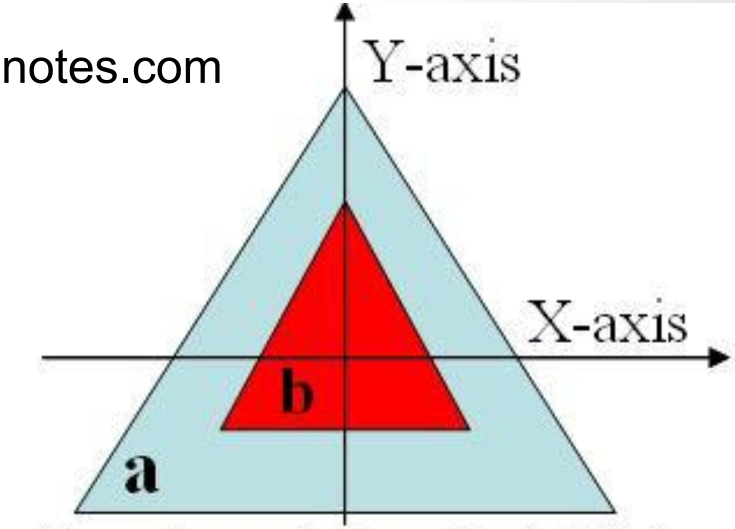
- s_x scales object in 'x' direction
- s_y scales object in 'y' direction

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.



Nipun Thapa-Computer Graphics

3.Scaling



Turning a triangle (a) Into a triangle (b) with scaling factors s_x = -2 and s_y = - 2

Thus, for equation form,

$x' = x.\ s_x$ and

$y' = y.\ s_y$

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

- Values greater than 1 for s_x , s_y produce ***enlargement***
- Values less than 1 for s_x , s_y *reduce* size of object
- s x = s y = 1 leaves the size of the object ***unchanged***
- When s_x , s_y are assigned the same value s x = s y = 3 or 4 etc then a ***Uniform Scaling*** is produced

3.Scaling

P'= S.P

In homogeneous co-ordinate

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow P' = S(s_x, s_y).P$$

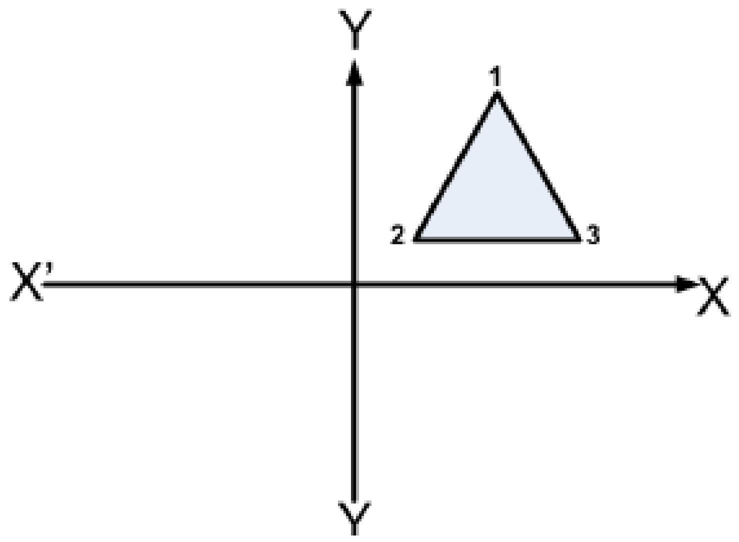
4. Reflection

A reflection is a transformation that produces a mirror image of an object. The mirror image for a 2D reflection is generated relative to an axis of reflection by rotating the object 180" about the reflection axis. We can choose an axis of reflection in the **xy**-plane or perpendicular to the xy plane. When the reflection axis is a line in the xy plane, the rotation path about this axis is in a plane perpendicular to the **xy**-plane. For reflection axes that are perpendicular to the **xy**-plane, the rotation path is in the xy plane.

4. Reflection

(i) Reflection about x axis or about line y = 0

Keeps **X** value same but flips **Y** value of coordinate points



x' = x
y' = -y

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

Homogeneous co-ordinate

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

4. Reflection

(ii) Reflection about y axis or about line x = 0

Keeps 'y' value same but flips x value of coordinate points

$x' = -x$

$y' = y$

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

Homogeneous co-ordinate

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

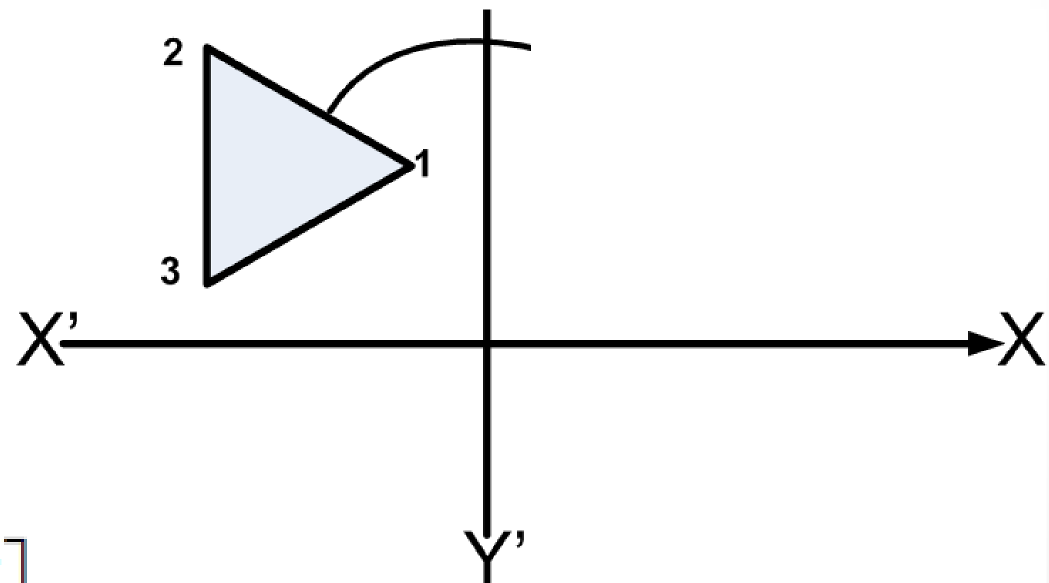


Fig: Reflection of object about y-axis (x=0)

4. Reflection

(iii) Reflection about origin

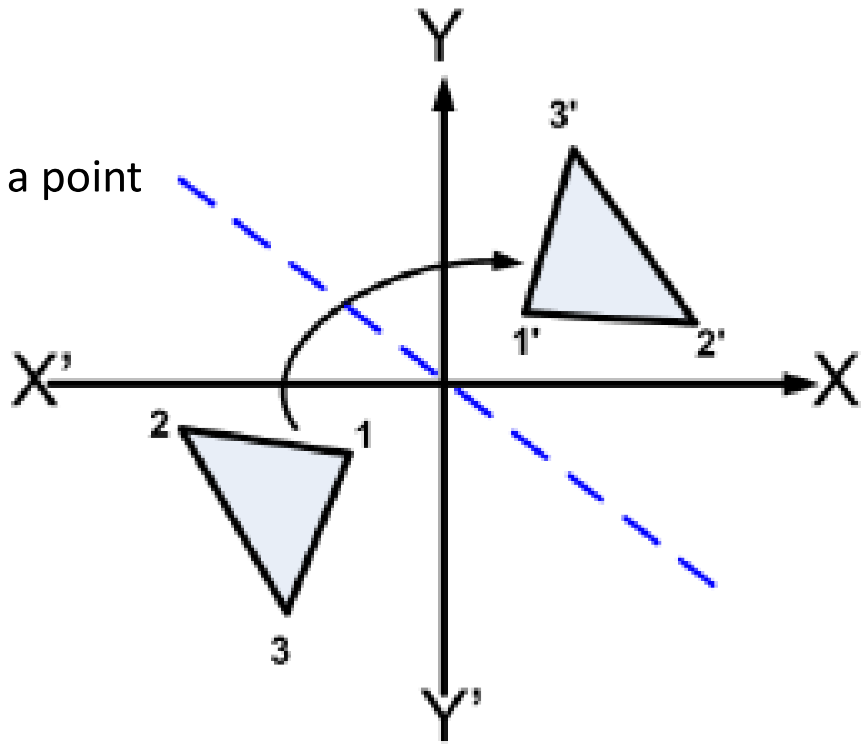
Flip both ‘x’ and ‘y’ coordinates of a point

x’ = -x
y’ = -y

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

Homogeneous co-ordinate

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



4. Reflection

(iv) Reflection about line y = x

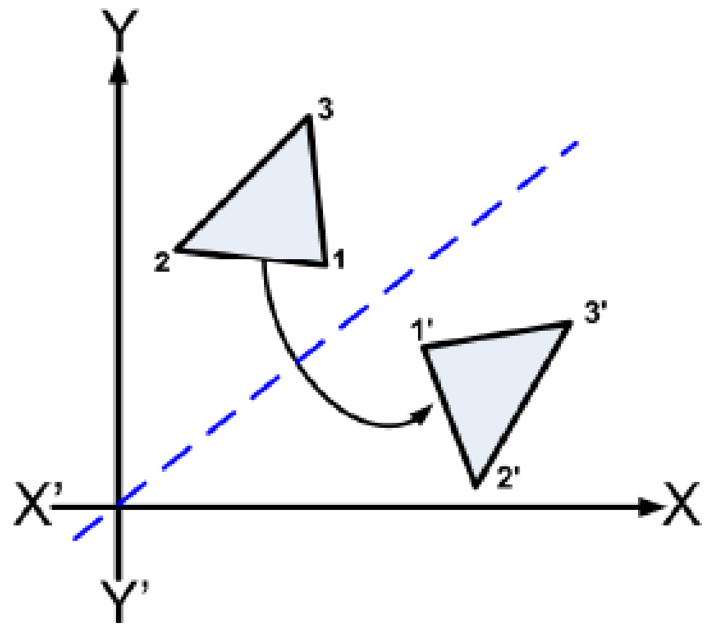
$x' = y$
$y' = x$

Thus, reflection against x=y-axis (i.e. θ =45)

$$R_{x=y} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Hence

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



Homogeneous co-ordinate

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

Equivalent to:

- Reflection about x-axis
- Rotate anticlockwise **90˚**

OR

- Clockwise rotation 45 ˚
- Reflection with x-axis
- anticlockwise rotation 45 ˚

4. Reflection

(v) Reflection about line y = -x

$x' = -y$

$y' = -x$

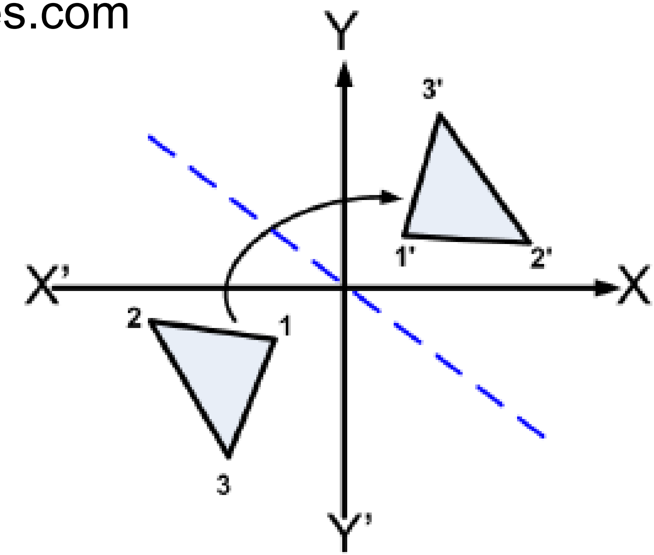
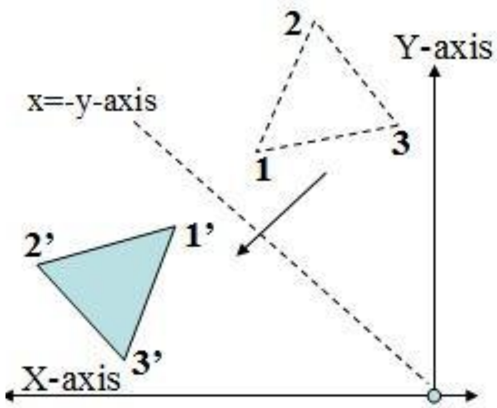
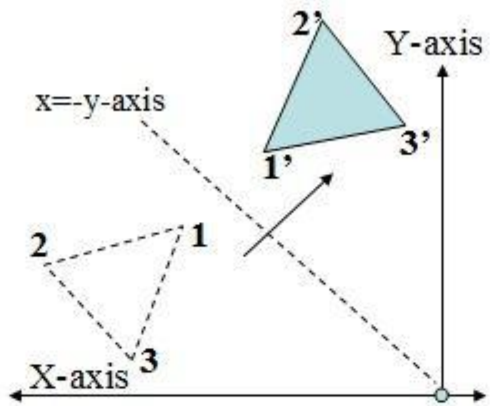


Fig: Reflection of object about y = - x



Reflection of an object about x= -y in anti-clockwise direction



Reflection of an object about x= -y in clockwise direction

Thus, reflection against x=y-axis
in anti-clockwise direction (i.e. θ = 45)

Hence

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad \textit{Where} \quad R_{x=-y} = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$

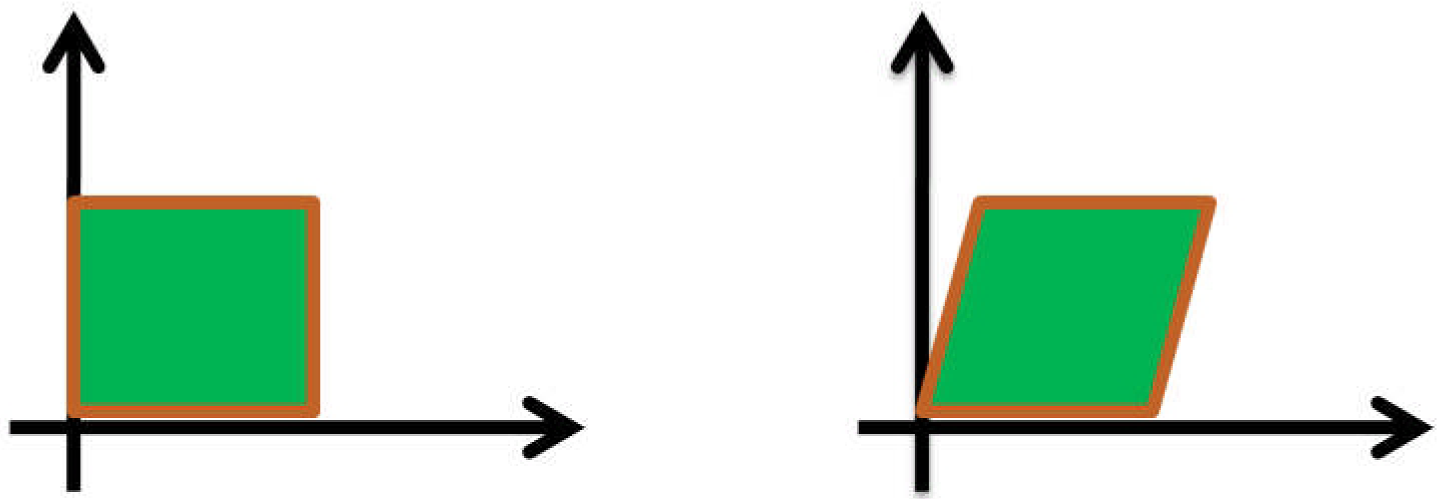
Thus, reflection against x=y-axis
in clockwise direction (i.e. θ = - 45)

Hence

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad \textit{Where} \quad R_{x=-y} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

5. Shearing

It distorts the shape of object in either ‘x’ or ‘y’ or both direction. In case of single directional shearing (e.g. in ‘x’ direction can be viewed as an object made up of very thin layer and slid over each other with the *base* remaining where it is). Shearing is a ***non-rigid-body transformation*** that moves objects with deformation.



5. Shearing

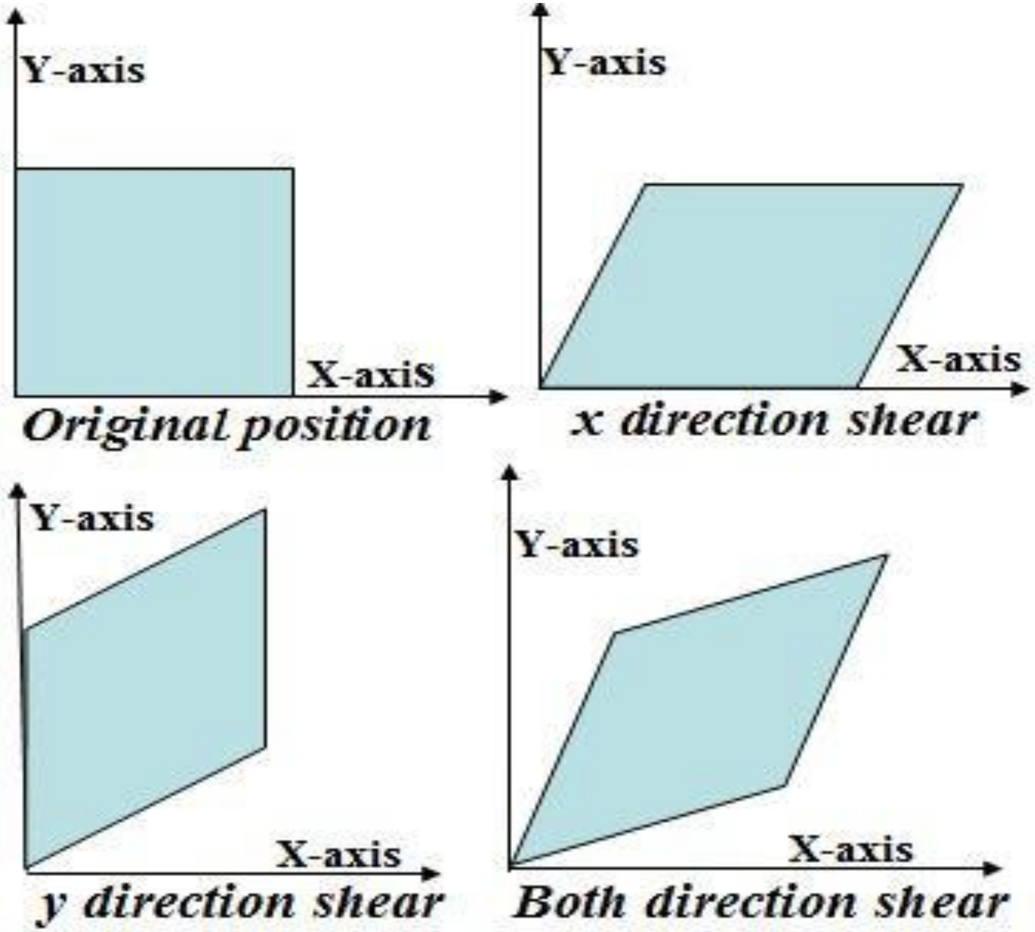


Fig. Two Dimensional shearing

5. Shearing

In 'x' direction,

$x' = x + S_{hx} \cdot y$

$y' = y$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & S_{hx} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

In 'y' direction,

$x' = x$

$y' = y + S_{hy} \cdot x$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ S_{hy} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

In both directions,

$x' = x + S_{hx} \cdot y$

$y' = y + S_{hy} \cdot x$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & S_{hx} \\ S_{hy} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Homogenous Coordinates

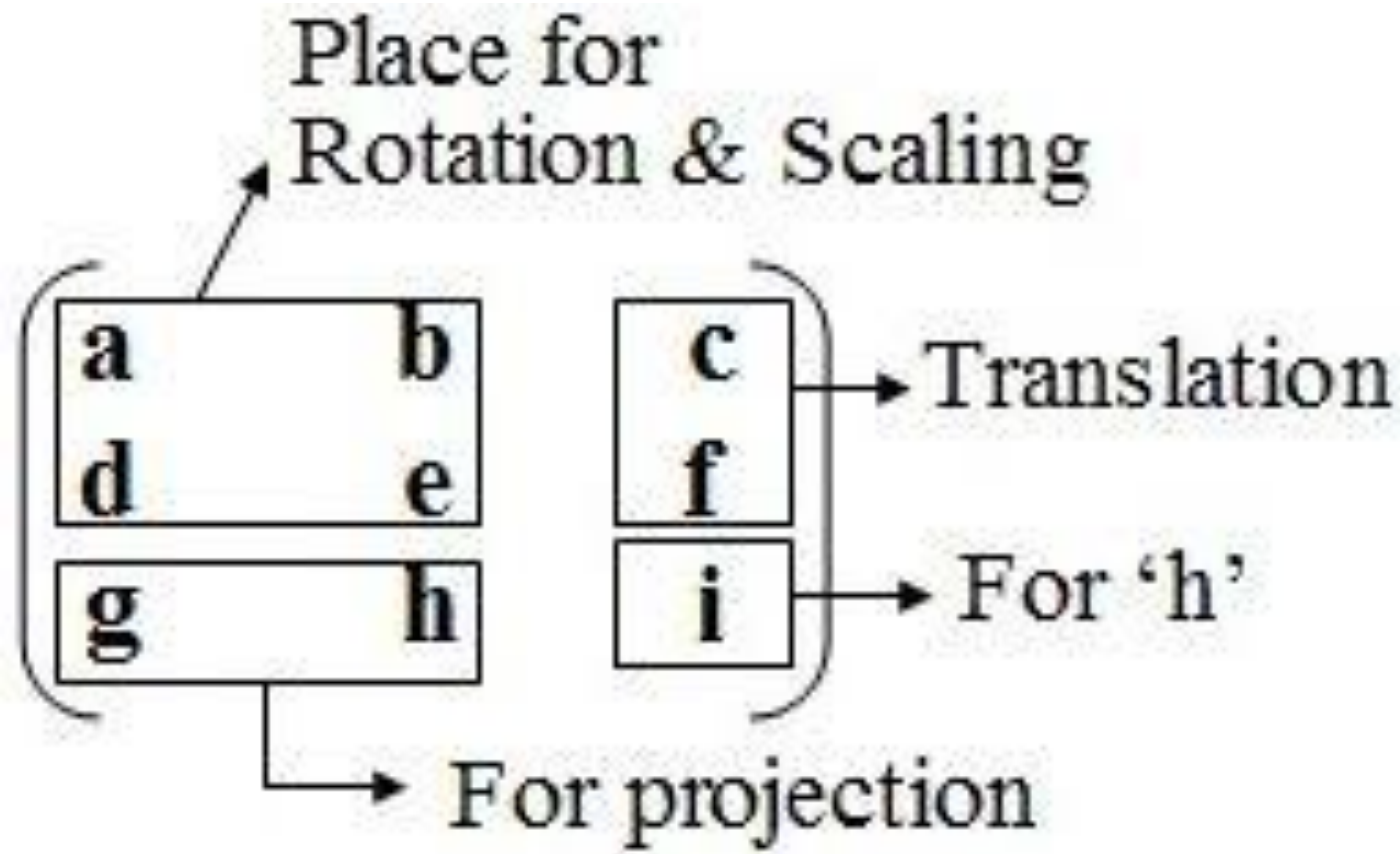
The matrix representations for translation, scaling and rotation are respectively:

- Translation: **P’ = T + P** (*Addition*)
- Scaling: **P’ = S . P** (*Multiplication*)
- Rotation: **P’ = R . P** (*Multiplication*)

Since, the composite transformation such as include many sequence of translation, rotation etc and hence the many naturally differ addition & multiplication sequence have to perform by the graphics allocation. Hence, the applications will take more time for rendering.

Thus, we need to treat all three transformations in a consistent way so they can be combined easily & compute with one mathematical operation. If points are expressed in homogenous coordinates, all geometrical transformation equations can be represented as matrix multiplications.

Homogenous Coordinates



Homogenous Coordinates

Here, in case of homogenous coordinates we add a third coordinate ‘*h*’ to a point (x, y) so that each point is represented by (*h*x, *h*y, *h*). The ‘*h*’ is normally set to 1. If the value of ‘h’ is more the one value then all the co-ordinate values are scaled by this value.

Homogenous Coordinates

- **For translation**

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

With T(tx , ty) as translation matrix, inverse of this translation matrix is obtained by representing tx, ty with -tx, -

- **For rotation**

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

(a)

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

(b)

Here, figure-a shows the Counter Clockwise (CCW) rotation & figure-b shows the Clockwise (CW) rotation.

- **For scaling**

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} S_{hx} & 0 & 0 \\ 0 & S_{hy} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- **For Reflection**
  - Reflection about x-axis

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

  - Reflection about y-axis

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

  - Reflection about y=x-axis

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

  - Reflection about y=-x-axis

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

  - Reflection about any line y=mx+c

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{-(m^2-1)}{(m^2+1)} & \frac{2m}{(m^2+1)} & \frac{2mc}{(m^2+1)} \\ \frac{2m}{(m^2+1)} & \frac{(m^2-1)}{(m^2+1)} & \frac{2c}{(m^2+1)} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

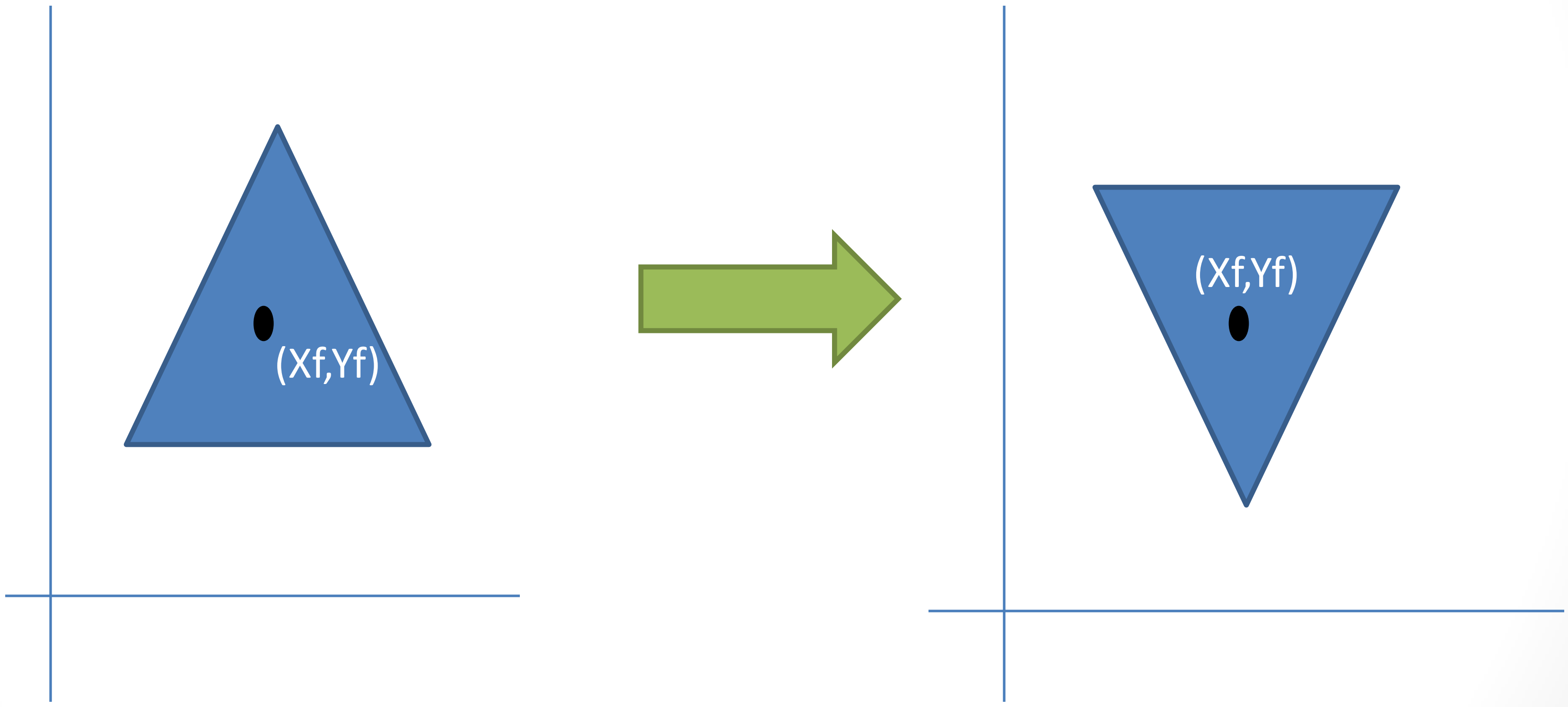
Composite Transformation

With the matrix representation of transformation equations it is possible to setup a matrix for any sequence of transformations as a composite transformation matrix by calculating the matrix product of individual transformation. Forming products of transformation matrices is often referred to as a **concatenation,** or **composition,** of matrices. For column matrix representation of coordinate positions we form composite transformation by multiplying matrices in order from right to left.

Right to Left

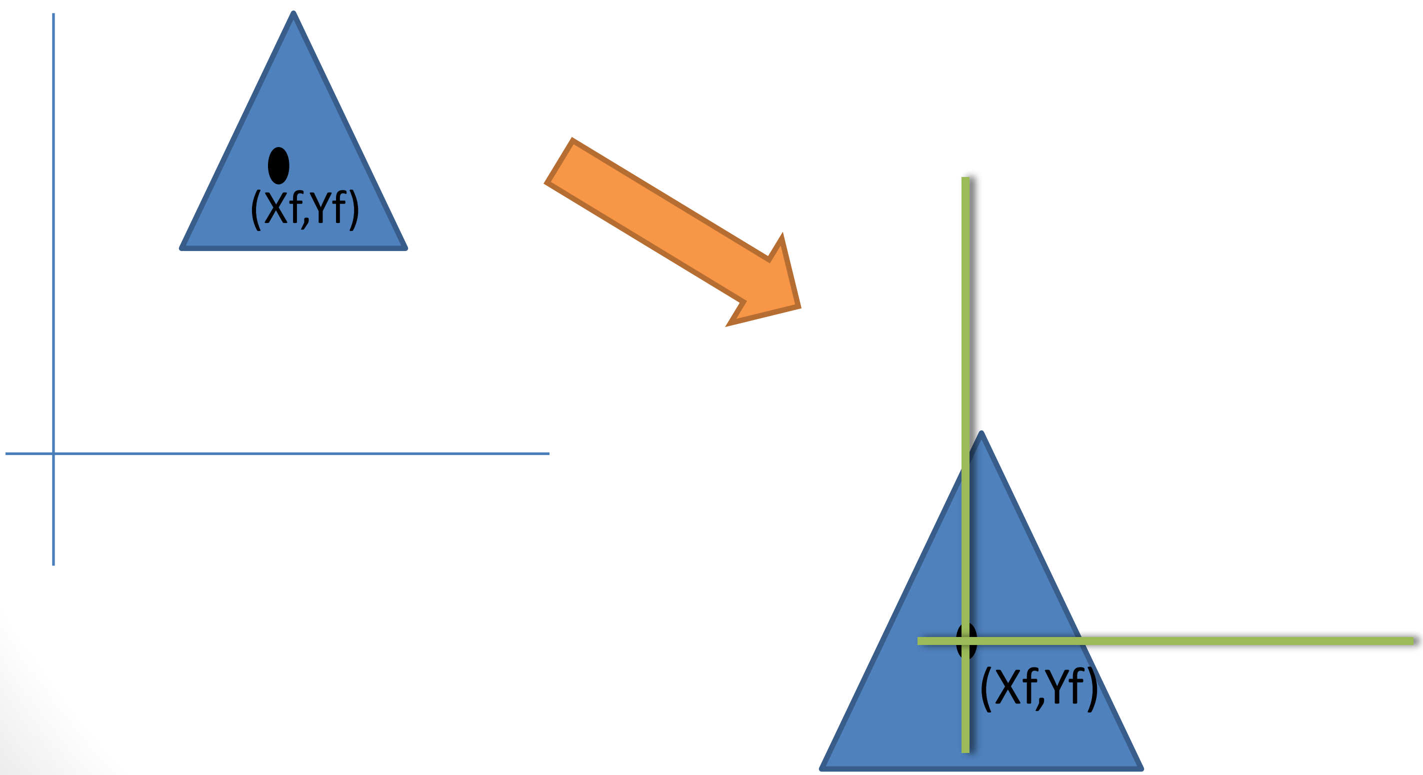
A.B != B.A

Fixed Point Rotation



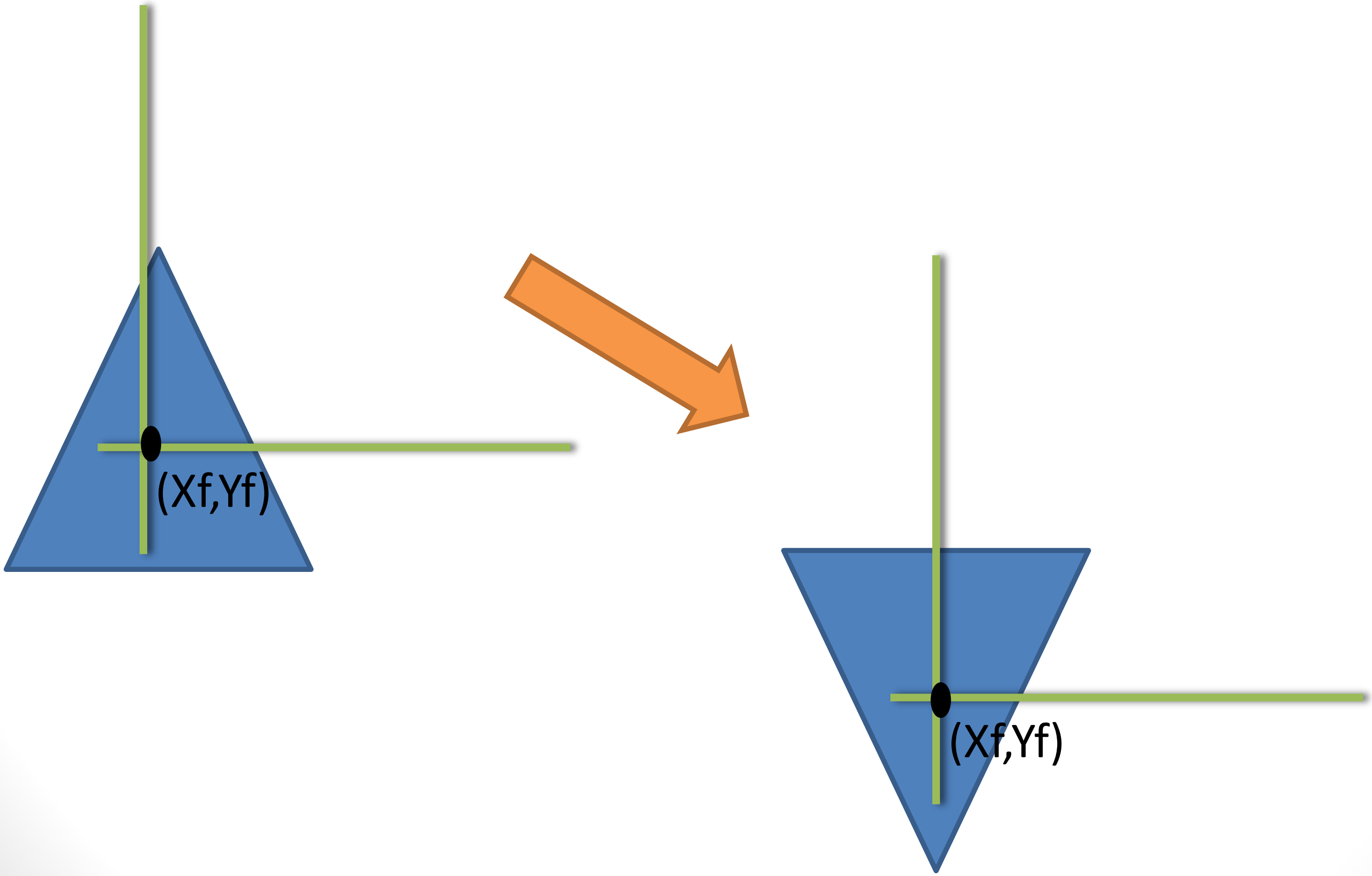
Fixed Point Rotation

Step 1: The fixed point along with the object is translated to coordinate origin.



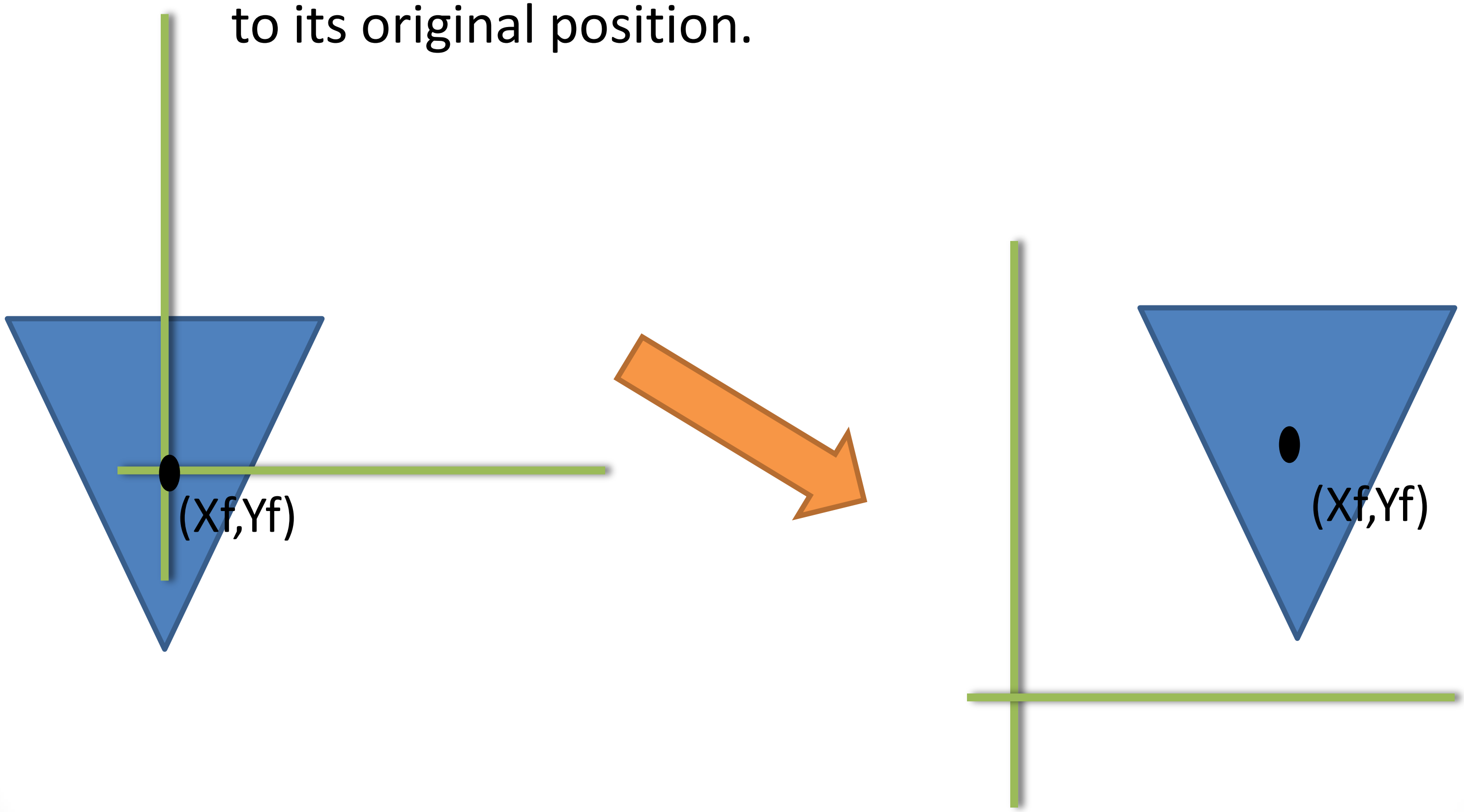
Fixed Point Rotation

Step 2: Rotate the objet about origin

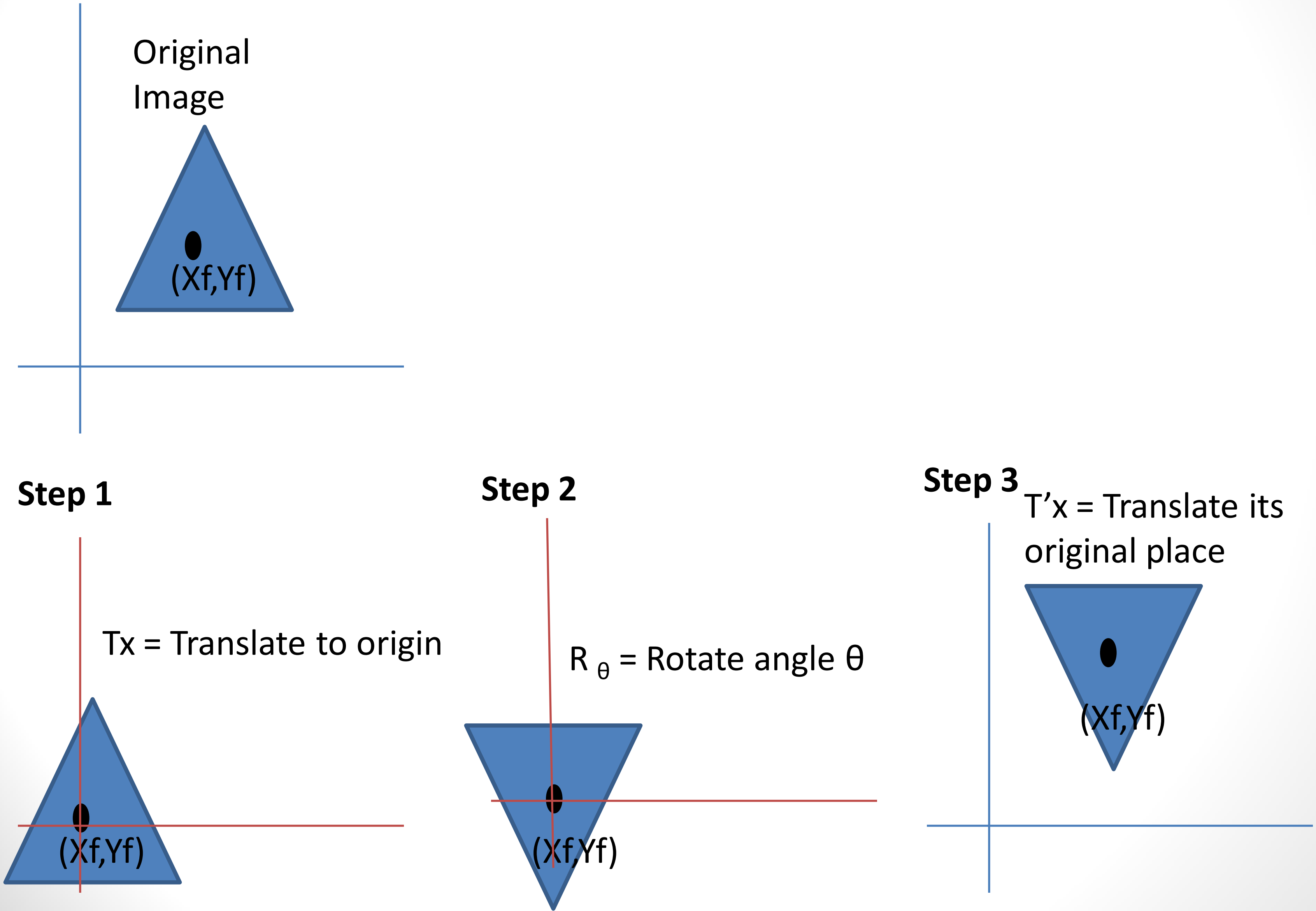


Fixed Point Rotation

Step 3: The fixed point along with the object is translated back to its original position.



Fixed Point Rotation



Fixed Point Rotation

Composite Transformation

$$= T'_{(Xf\ ,Yf)} \cdot R_{\theta} \cdot T_{(-Xf\ ,-Yf)}$$

$$= \begin{bmatrix} 1 & 0 & Xf \\ 0 & 1 & Yf \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \text{Cos}\,\theta & -\text{Sin}\,\theta & 0 \\ \text{Sin}\,\theta & \text{Cos}\,\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -Xf \\ 0 & 1 & -Yf \\ 0 & 0 & 1 \end{bmatrix}$$

Exercise 1

Q. N. > Rotate the triangle (5, 5), (7, 3), (3, 3) in counter clockwise (CCW) by 90 degree.

A :

P' = R. P

$$= \begin{pmatrix} Cos90 & -Sin90 & 0 \\ Sin90 & Cos90 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 5 & 7 & 3 \\ 5 & 3 & 3 \\ 1 & 1 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 5 & 7 & 3 \\ 5 & 3 & 3 \\ 1 & 1 & 1 \end{pmatrix}$$

$$= \begin{bmatrix} -5 & -3 & -3 \\ 5 & 7 & 3 \\ 1 & 1 & 1 \end{bmatrix}$$

Exercise 2

Q.> *Rotate the triangle (5, 5), (7, 3), (3, 3) about fixed point (5, 4) in counter clockwise (CCW) by 90 degree.*

A : *Solution*

Here, the required steps are:

1. Translate the fixed point to origin.
2. Rotate about the origin by specified angle θ .
3. Reverse the translation as performed earlier.

Thus, the composite matrix is given by

Com = T(xf, yf) . R_θ .T(-xf, -yf)

$$= \begin{pmatrix} 1 & 0 & 5 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Cos90 & -Sin90 & 0 \\ Sin90 & Cos90 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -5 \\ 0 & 1 & -4 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 5 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -5 \\ 0 & 1 & -4 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 5 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 4 \\ 1 & 0 & -5 \\ 0 & 0 & 1 \end{pmatrix}$$

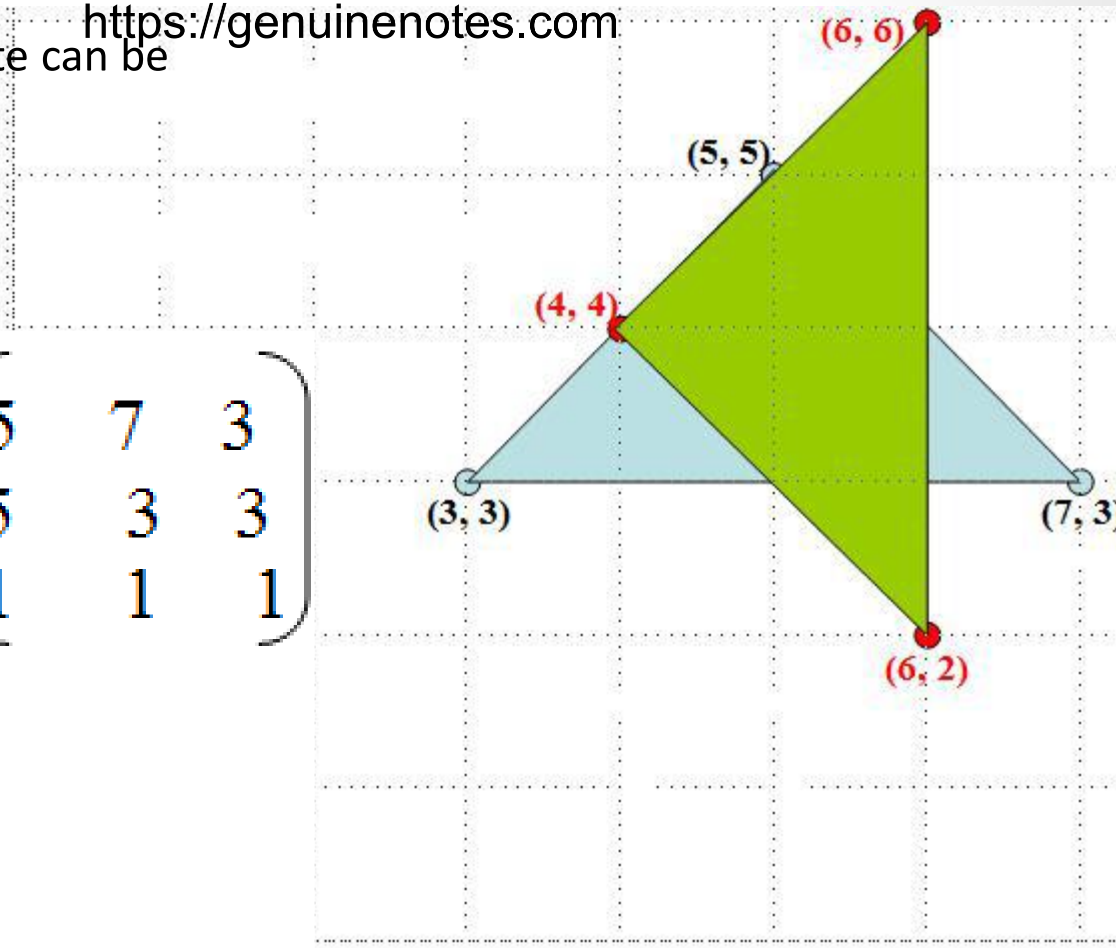
$$= \begin{pmatrix} 0 & -1 & 9 \\ 1 & 0 & -1 \\ 0 & 0 & 1 \end{pmatrix}$$

Now, the required co-ordinate can be calculated as:

P' = Com . P

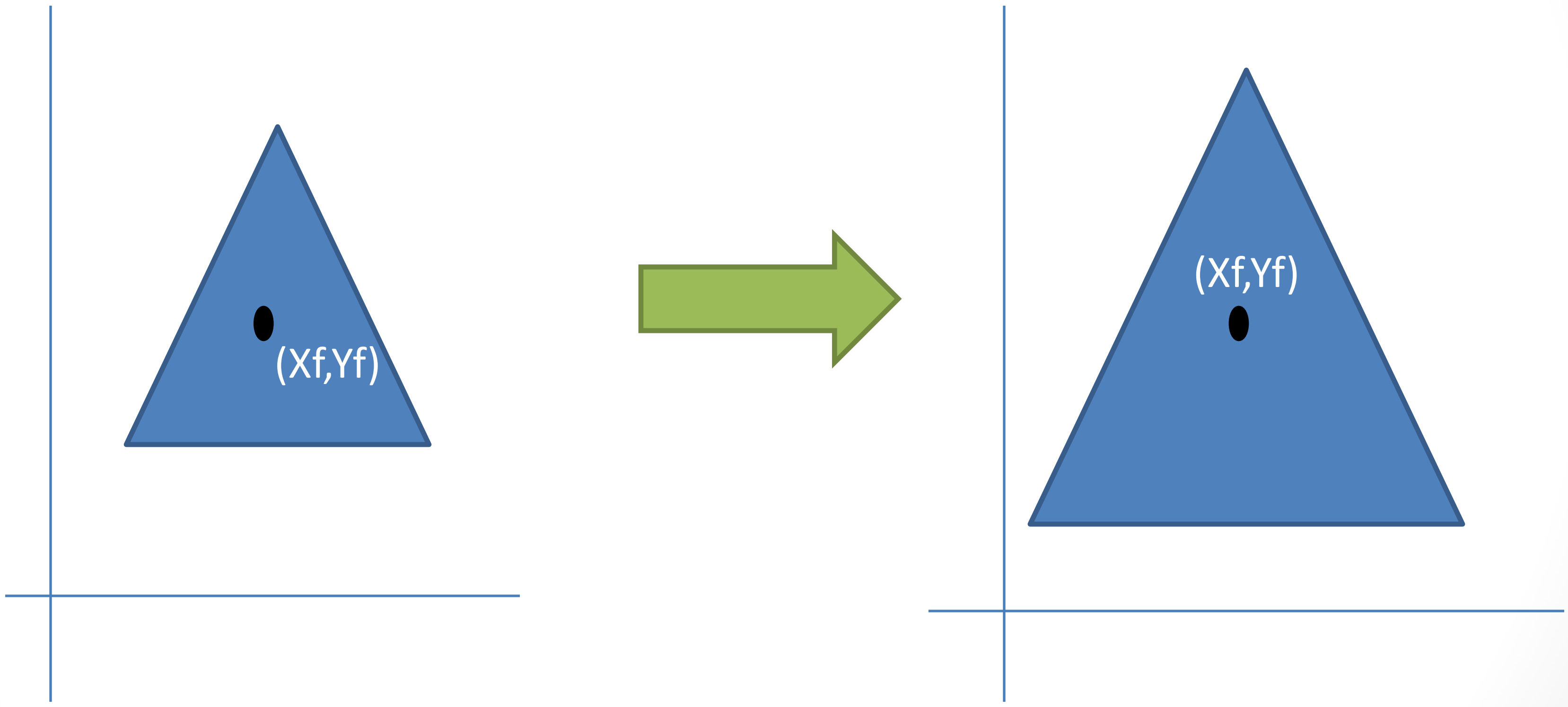
$$= \begin{pmatrix} 0 & -1 & 9 \\ 1 & 0 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 5 & 7 & 3 \\ 5 & 3 & 3 \\ 1 & 1 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 4 & 6 & 6 \\ 4 & 6 & 2 \\ 1 & 1 & 1 \end{pmatrix}$$



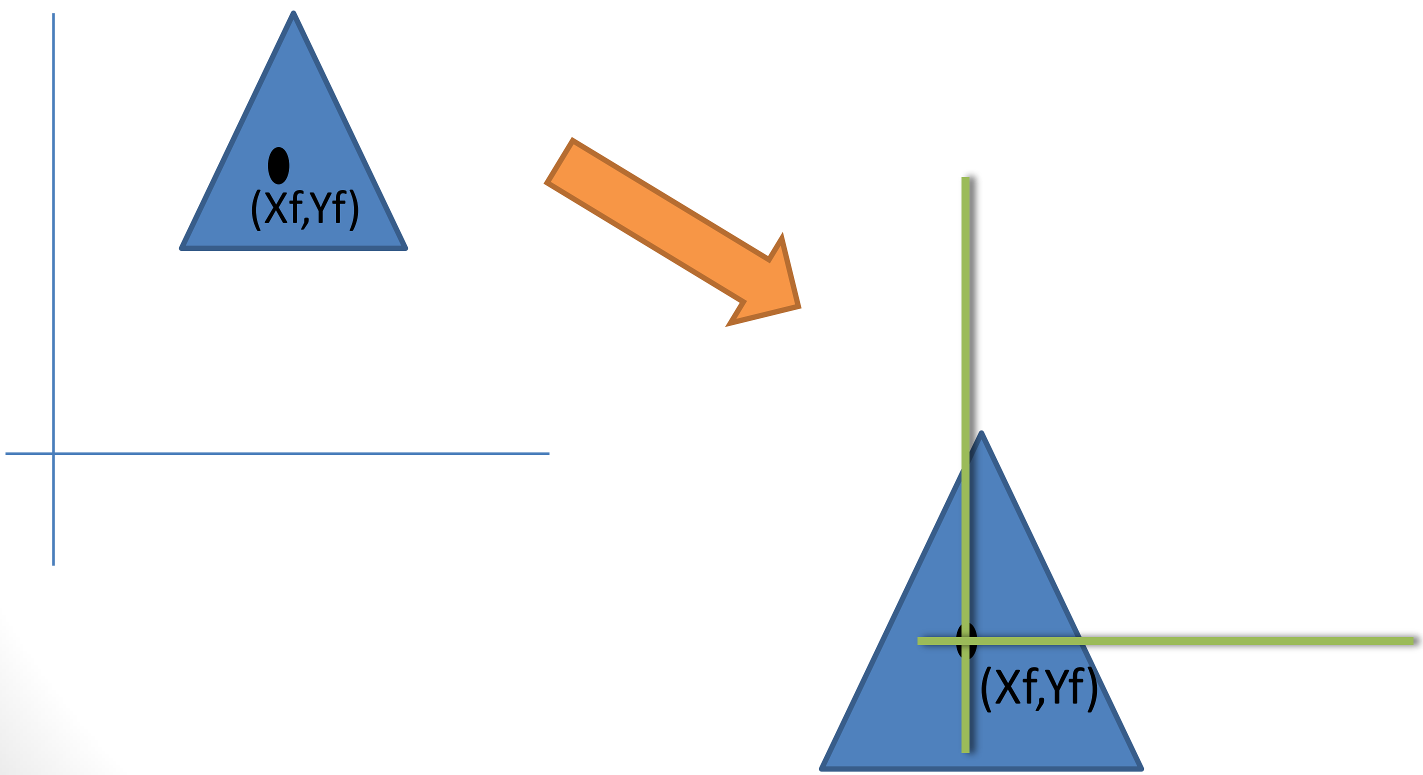
Hence, the new required coordinate points are **(4, 4), (6, 6) & (6, 2).**

Fixed Point Scaling



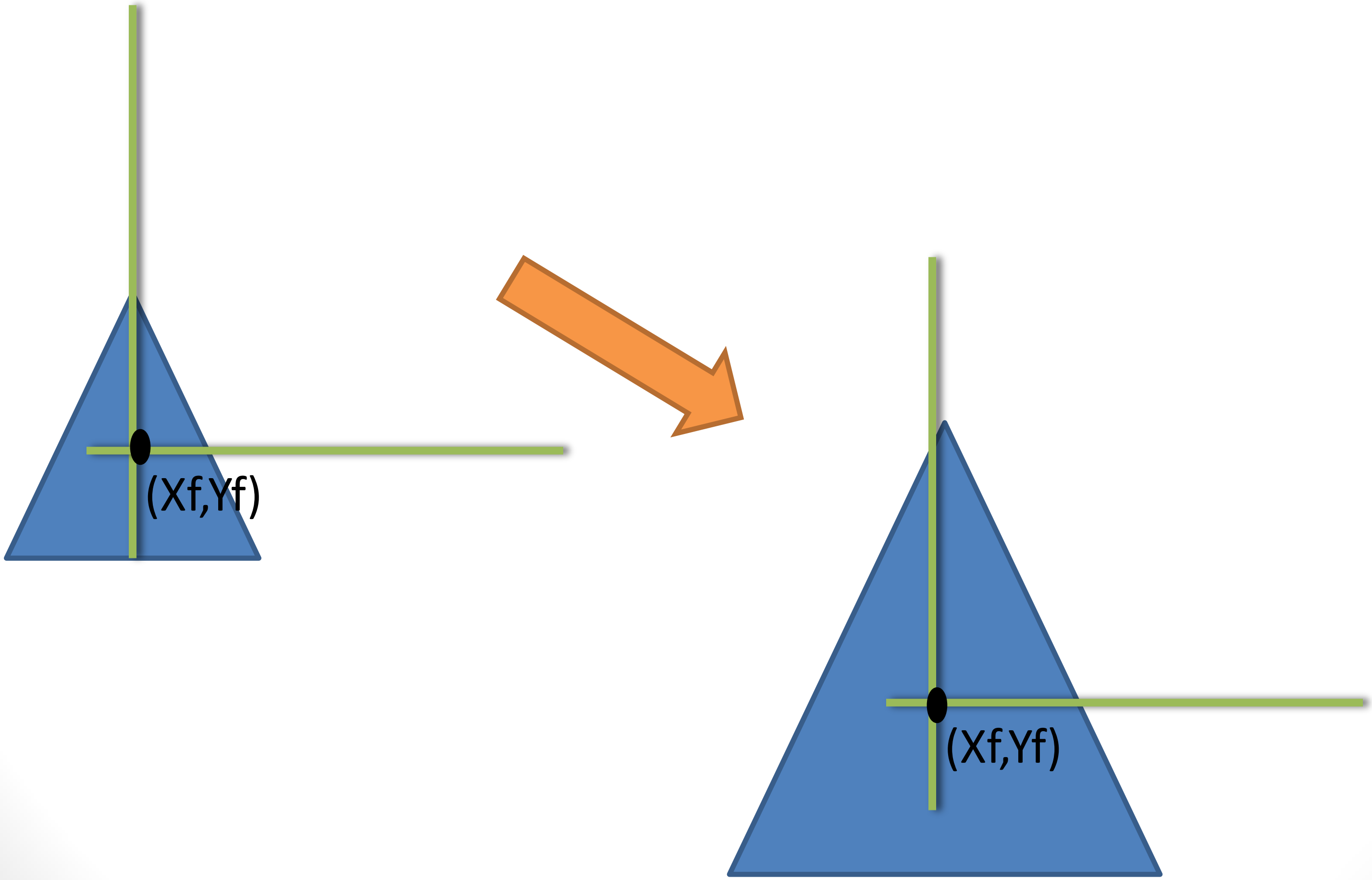
Fixed Point Scaling

Step 1: The fixed point along with the object is translated to coordinate origin.



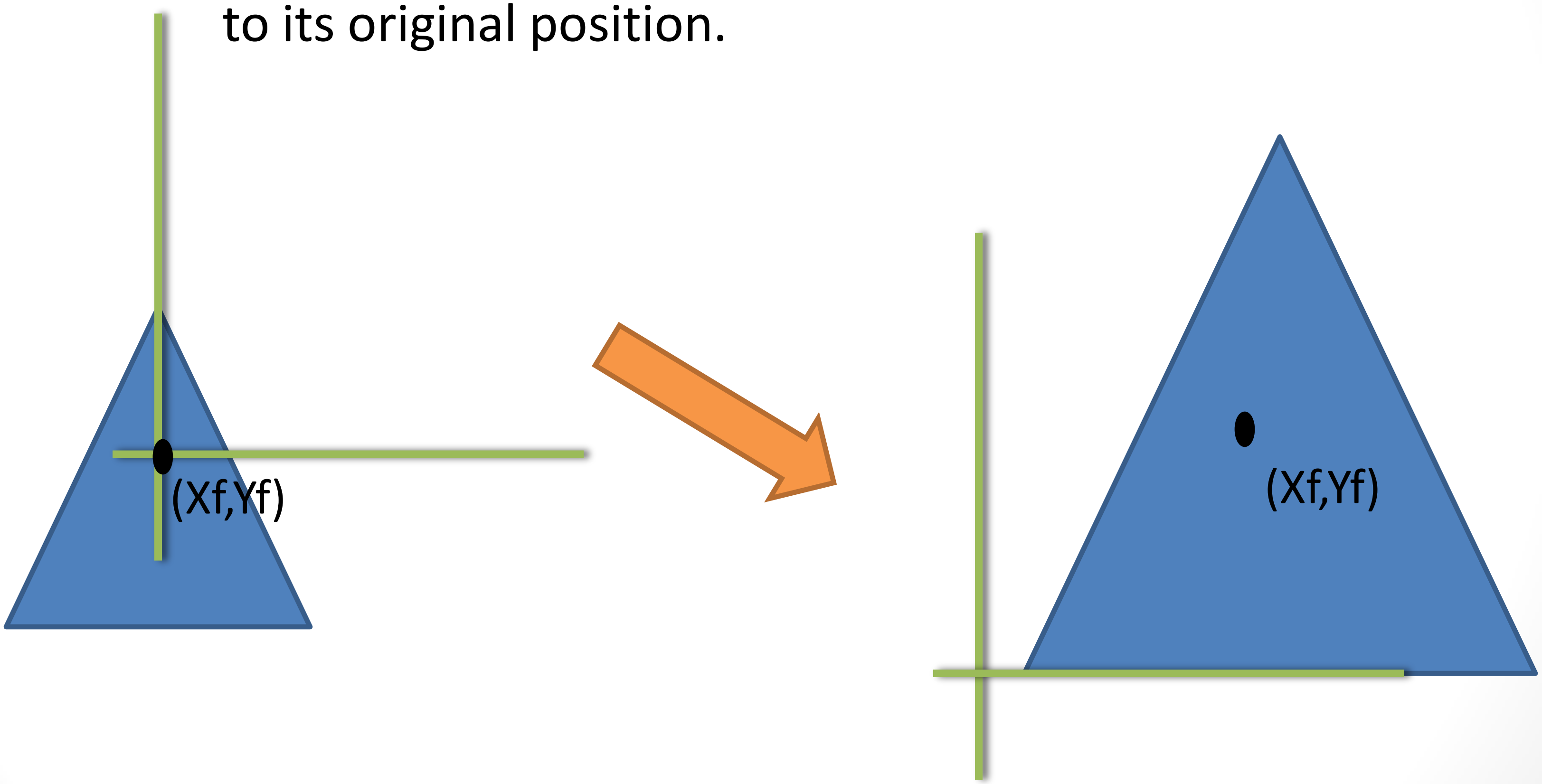
Fixed Point Scaling

Step 2 : Scaling the objet about origin

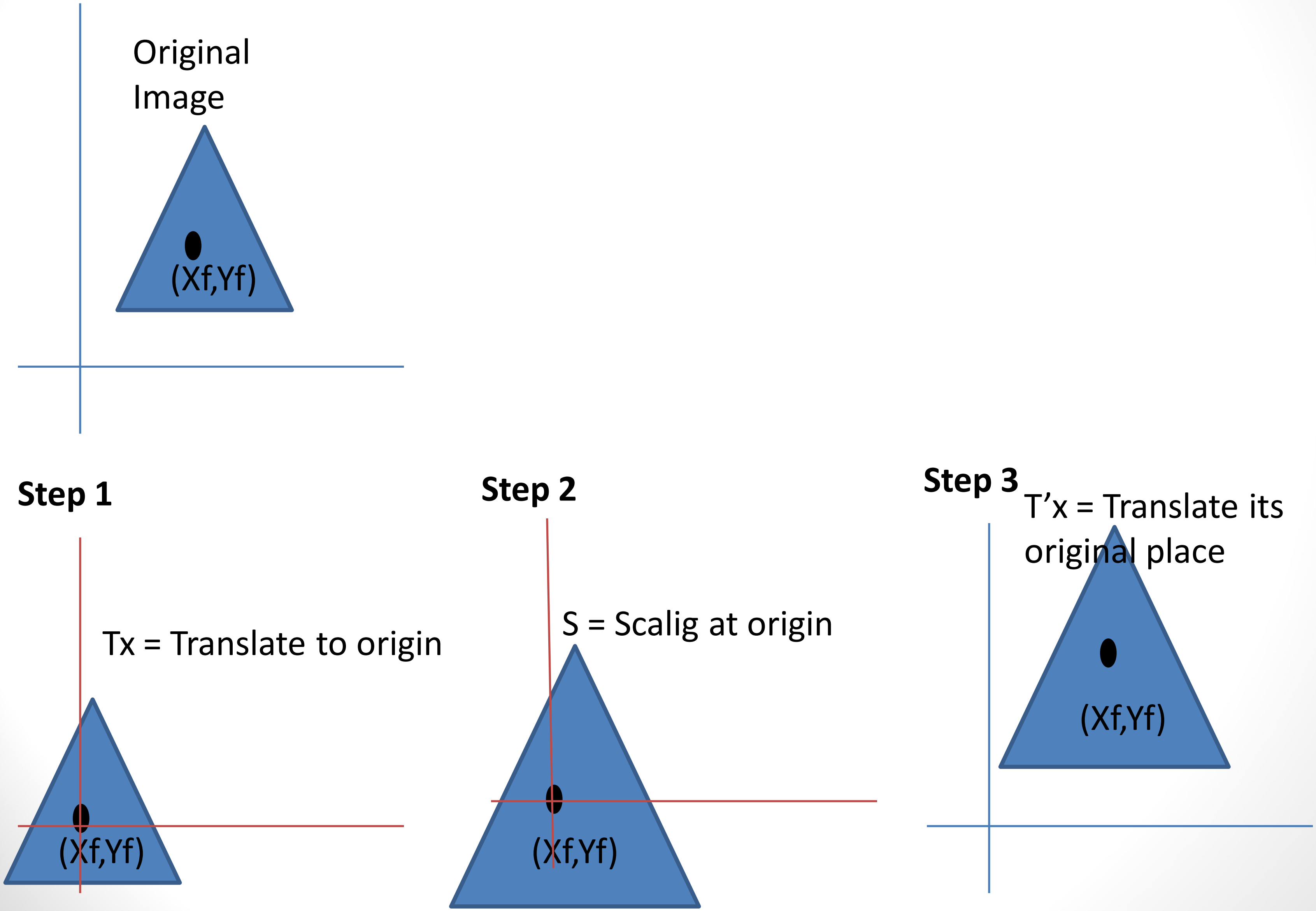


Fixed Point Scaling

Step 3: The fixed point along with the object is translated back to its original position.



Fixed Point Scaling



Fixed Point Rotation

Composite Transformation

$$= T'_{(Xf\ ,Yf)} \cdot S_{axix} \cdot T_{(-Xf\ ,-Yf)}$$

$$= \begin{bmatrix} 1 & 0 & Xf \\ 0 & 1 & Yf \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -Xf \\ 0 & 1 & -Yf \\ 0 & 0 & 1 \end{bmatrix}$$

Exercise 2

Q.N.1. > Find the coordinate of a triangle A(1,3) , B(2,5) and C(3,3) after twice its original size

a) about origin

b)about fixed point (2,3)

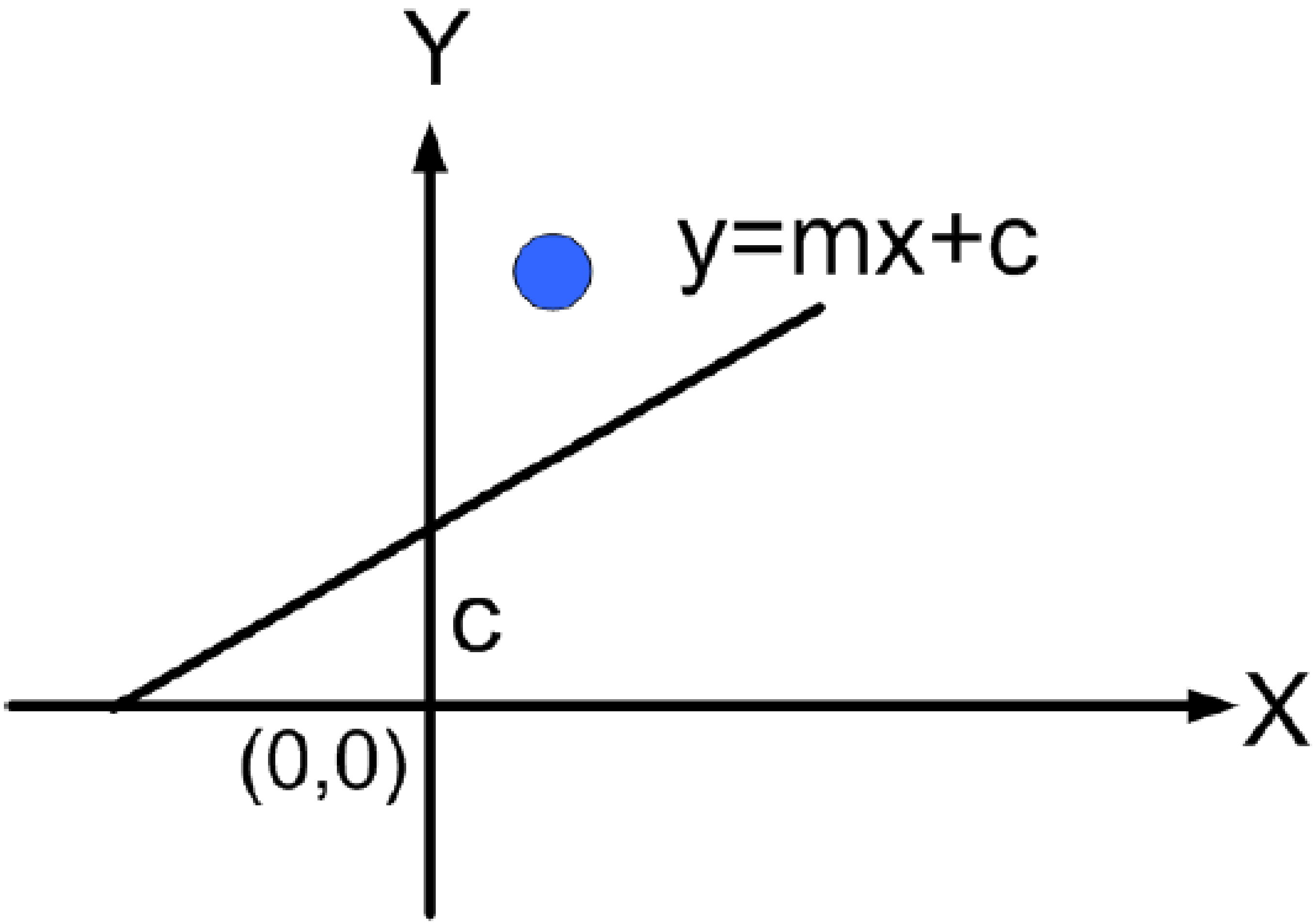
Q.N.2. > Find the coordinate of a triangle A(1,3) , B(2,5) and C(3,3) after being rotated about fixed point p(2,4) by 45 in clockwise direction and then translate (3,4) .

Q.N.3.>Find the coordinate of a triangle A(1,3) , B(2,5) and C(3,3) after being rotated about fixed point p(2,4) by 45 in clockwise direction and then translate by 3 unit along x-direction.

Q.N.4 >Find the coordinate of a *triangle (5, 5), (7, 3), (3, 3) after scaling (2,3) and then rotate 45 CCW then translate (1,1) and then reflect y=0 line.*

1. Find the coordinate of a *triangle (5, 5), (7, 3), (3, 3) after translate (1,0) then scaling (3,3) then rotate 45 CW then translate (1,1) and then reflect x=0 line.*
2. Find the coordinate of a *line (5, 5), (3, 3) after rotate 45^0 ccw at fixed point (3,3) then translate (1,1) then reflect at x-axis.*
3. Find the coordinate of a *point (5, 5) after reflection y-axis then scaling twice then rotate 30 CW then rotate 30^0 CW at fixed point (3,4)*
4. *Draw a circle at (3,4) with radius 10.*
5. *Draw an ellipse where x-radius is 6 and y-radius is 8 at center (0,0)*
6. *Digitize the line (1,10) and (10,9) using BLA.*
7. *Clipping the line A(1,5) and B(20,55) where windows has P1(5,5) and P2(50,50).*

Reflection about Line y=mx+c

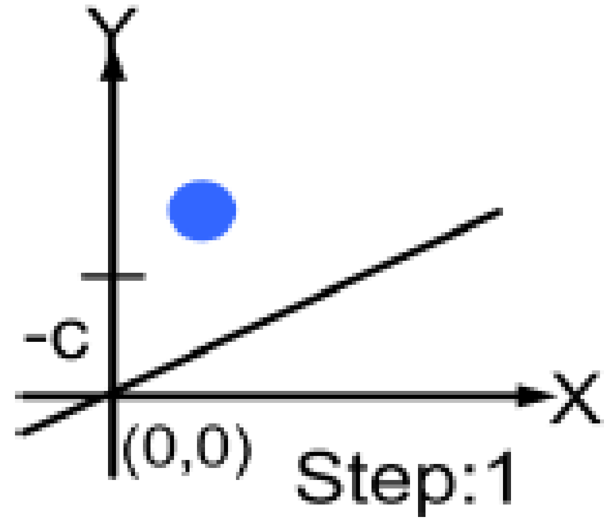


Reflection about Line y=mx+c

1. First translate the line so that it passes through the origin

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

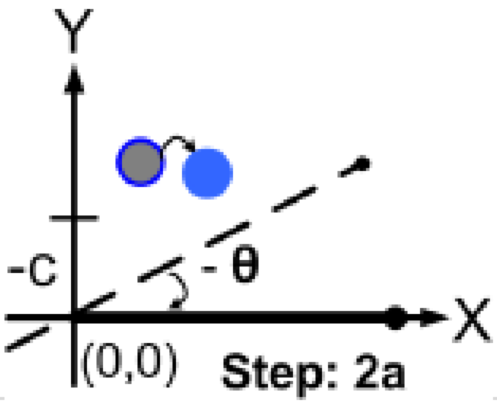
translation



Reflection about Line y=mx+c

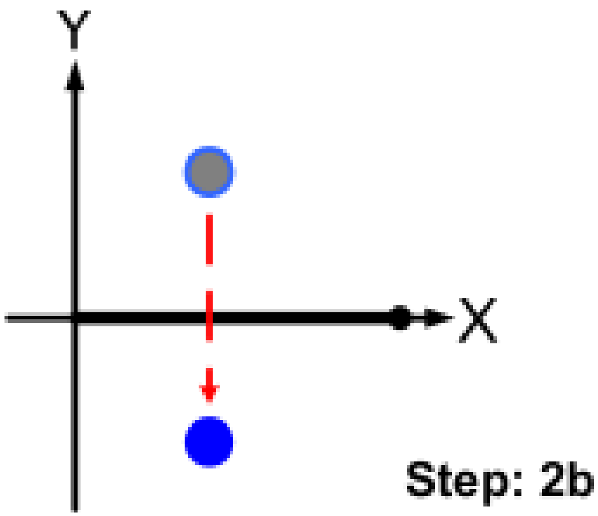
2. Rotate the line onto one of the coordinate axes(say x-axis) and reflect about that axis (x-axis)

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



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

reflection

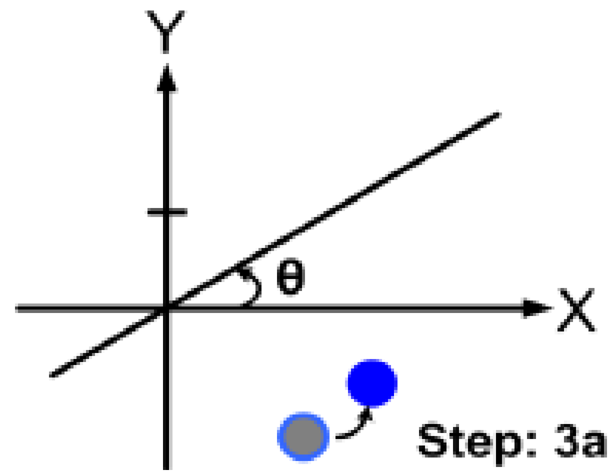


Reflection about Line y=mx+c

3. Finally, restore the line to its original position with the inverse rotation and translation transformation.

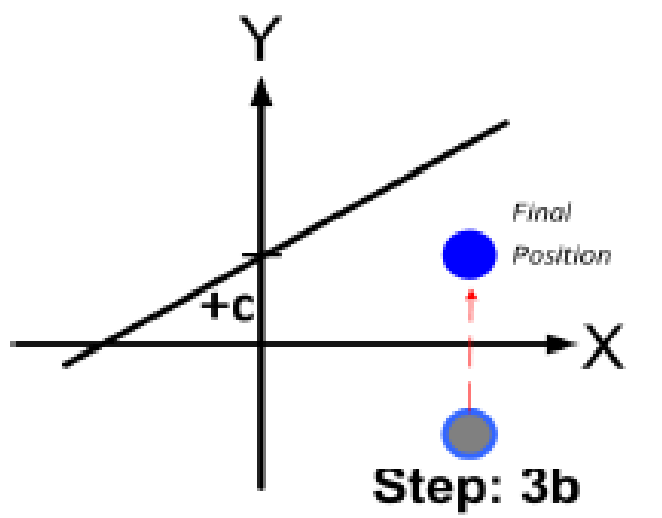
$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

rotation



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

translation



Reflection about Line y=mx+c

$$CM = T'_{(0,C)} \cdot R'_{\theta} \cdot R_{refl} \cdot R_{\theta} \cdot T_{(0,-C)}$$

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

Exercise

Q.N.1 > Reflect an object (2, 3), (4, 3), (4, 5) about line $y = x + 1$.

Solution

Here,

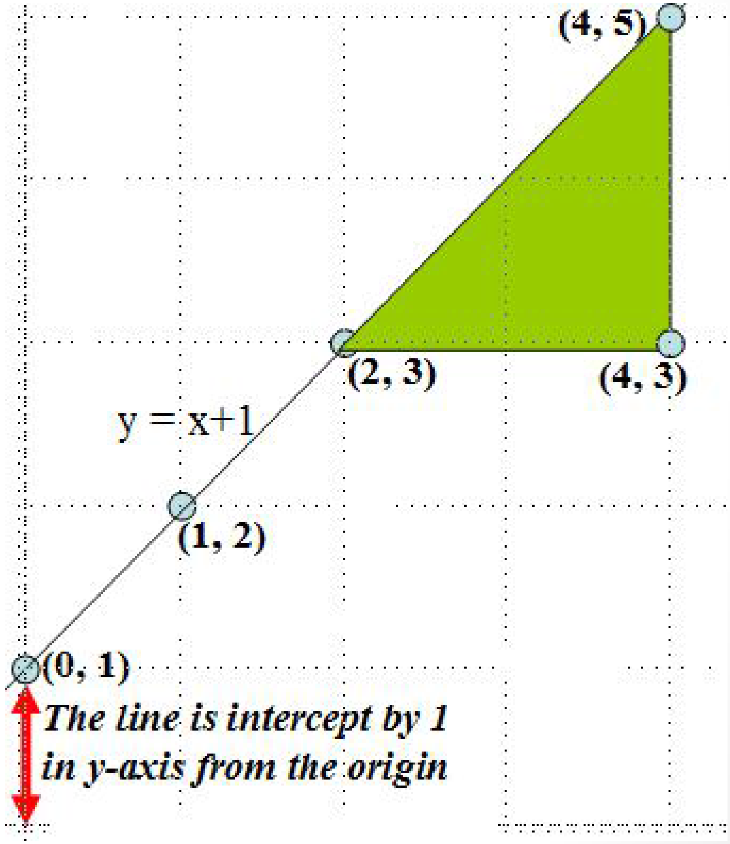
The given line is $y = x + 1$.

Thus,

When $x = 0, y=1$

When $x = 1, y=2$

When $x = 2, y=3$



Also,

The slope of the line (m) = 1

Thus, the rotation angle (θ) = $Tan^{-1}(m) = Tan^{-1}(1) = 45^0$

Here, the required steps are:

- Translate the line to origin by decreasing the y-intercept with one.
- Rotate the line by angle 45^0 in clockwise direction so that the given line must overlap x-axis.
- Reflect the object about the x-axis.
- Reverse rotate the line by angle -45^0 in counter-clockwise direction.
- Reverse translate the line to original position by adding the y-intercept with one.

Thus, the composite matrix is given by:

$$CM = T'_{(0,C)} \cdot R'_{\theta} \cdot R_{refl} \cdot R_{\theta} \cdot T_{(0,-C)}$$

Addition y-intercept | ***CCW Rotation*** | ***Reflection about x-axis*** | ***CW Rotation*** | ***Reduce y-intercept***

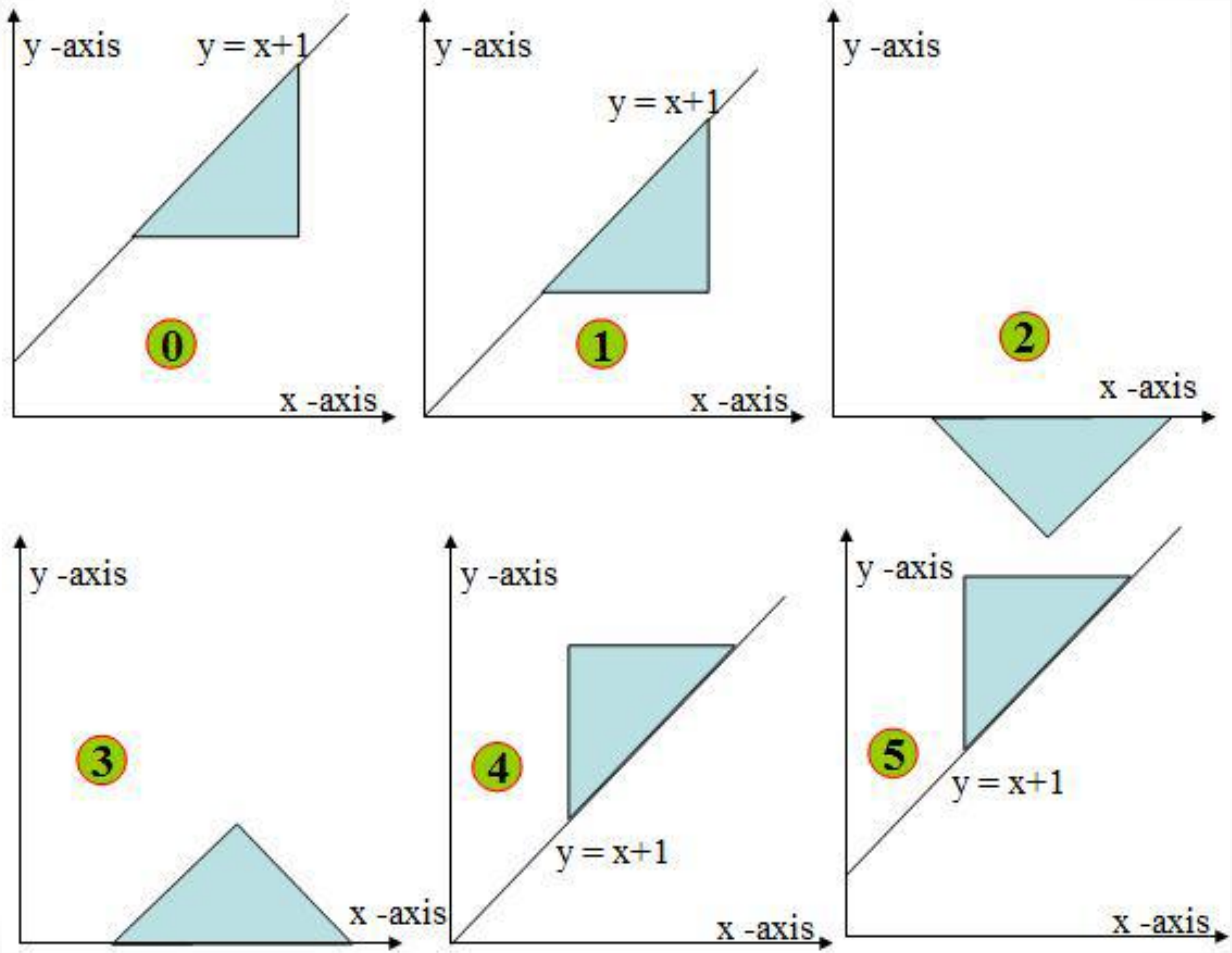
$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{Cos}45 & -\mathrm{Sin}45 & 0 \\ \mathrm{Sin}45 & \mathrm{Cos}45 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{Cos}45 & \mathrm{Sin}45 & 0 \\ -\mathrm{Sin}45 & \mathrm{Cos}45 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}$$

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

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

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

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & -1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & -1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$


y -axis
y = x+1
x -axis
0
y -axis
y = x+1
x -axis
1
y -axis
x -axis
2
y -axis
x -axis
3
y -axis
y = x+1
x -axis
4
y -axis
y = x+1
x -axis
5

Now, the required co-ordinate can be calculated as:

P' = Com x P

$$= \begin{pmatrix} 0 & 1 & -1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 & 4 \\ 3 & 3 & 5 \\ 1 & 1 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 2 & 2 & 4 \\ 3 & 5 & 5 \\ 1 & 1 & 1 \end{pmatrix}$$

Hence, the final coordinates are (2, 3), (2, 5) & (4, 5).

Exercise

Q.N.> Reflect a triangle A(1,8) B(3,8) and C(1,6) about line y= x+2

X	Y
0	2
1	3
2	4
3	5

Exercise (Imp)

Q.N> *A mirror is placed such that it passes through (0,10), (10, 0). Fin the mirror image of an object (6,7), (7, 6), (6, 9).*

Solution

Here,

The given mirror or line is passing through the points *(0, 10) & (10, 0).*

Now, the slope of the line

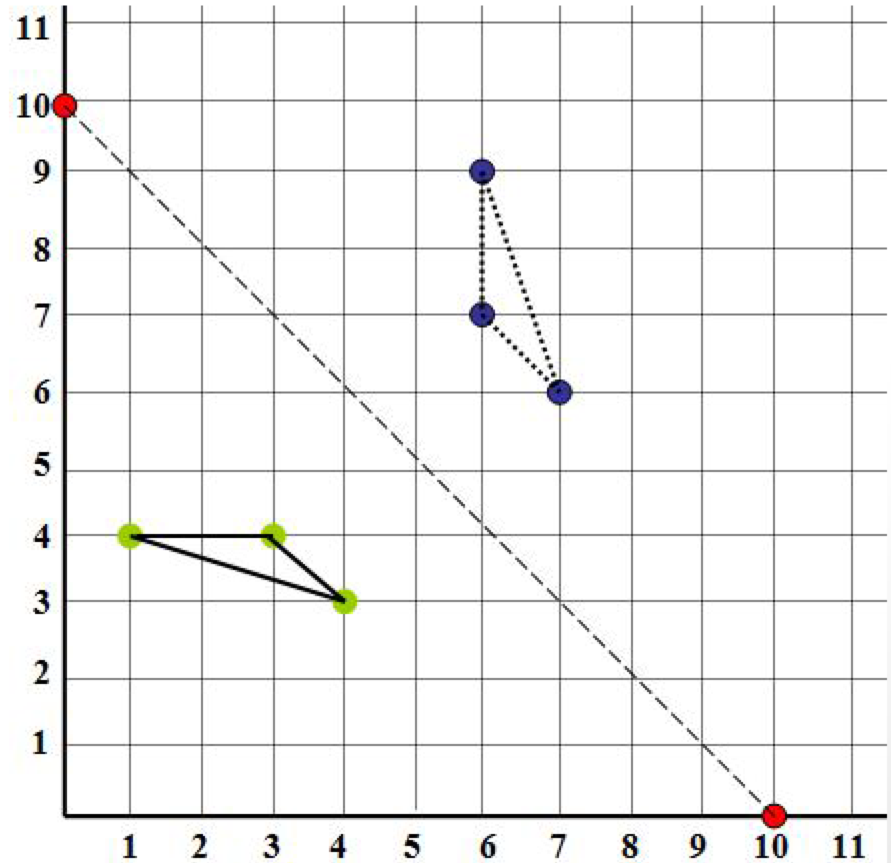
(m) = (y2-y1) / (x2-x1)

= (0 - 10) / (10 - 0) = -1

Thus, the rotation angle (**θ**)

= $\text{Tan}^{-1}(m) = \text{Tan}^{-1}(-1)$

= -45^0



The composite matrix is given by:

Com

$$= T_{(0,\ 10)\ or\ (10,\ 0).}\ R_{\theta\ in\ CW}\ .R_{fx}\ .R\ _{\theta\ in\ CCW}\ .T_{(0,\ -10)\ or\ (-10,\ 0)}$$

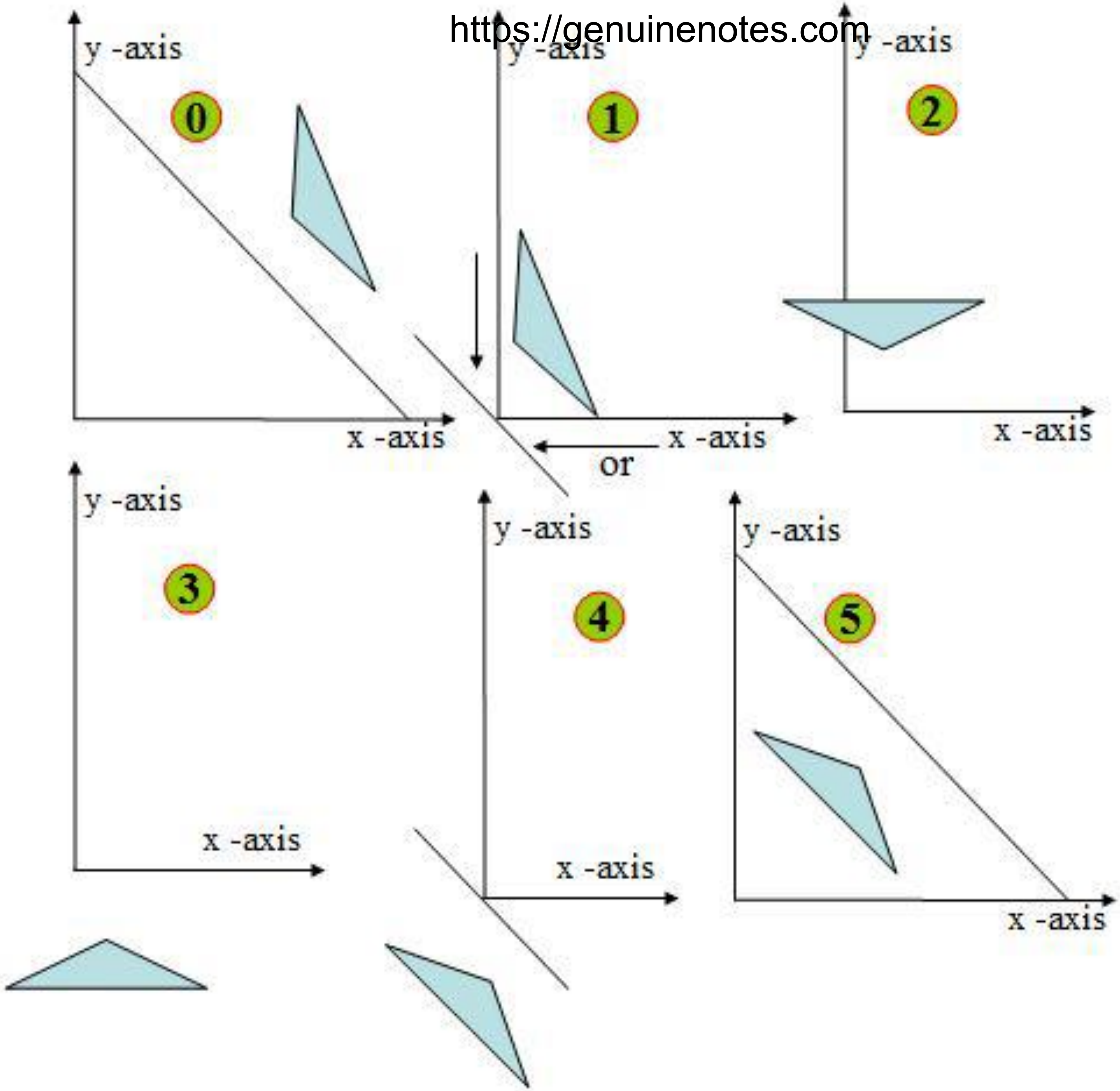
$$= \overset{\underline{\textit{Addition x-intercept}}}{\begin{pmatrix} 1 & 0 & 10 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}} \overset{\underline{\textit{CW Rotation}}}{\begin{pmatrix} \text{Cos45} & \text{Sin45} & 0 \\ -\text{Sin45} & \text{Cos45} & 0 \\ 0 & 0 & 1 \end{pmatrix}} \overset{\underline{\textit{Reflection about x-axis}}}{\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}} \overset{\underline{\textit{CCW Rotation}}}{\begin{pmatrix} \text{Cos45} & -\text{Sin45} & 0 \\ \text{Sin45} & \text{Cos45} & 0 \\ 0 & 0 & 1 \end{pmatrix}} \overset{\underline{\textit{Reduce x-intercept}}}{\begin{pmatrix} 1 & 0 & -10 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}$$

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

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

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

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


y -axis
x -axis
0
1
2
3
4
5
or

Now, the required co-ordinate can be calculated as:

P' = Com . P

$$= \begin{pmatrix} 0 & -1 & 10 \\ -1 & 0 & 10 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 6 & 7 & 6 \\ 7 & 6 & 9 \\ 1 & 1 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 3 & 4 & 1 \\ 4 & 3 & 4 \\ 1 & 1 & 1 \end{pmatrix}$$

Hence, the final coordinates are (3, 4), (4, 3) & (1, 4).

Home Work

- What do you mean by homogeneous coordinates? Rotate a triangle A(5,6), B(6,2) and C(4,1) by 45 degree about an arbitrary pivot point (3,3). (2072 TU).
- Use Bresenham's algorithm to draw a line having end points (25, 20) and (15, 10). (2072 TU).
- Given a clipping window P(0,0), Q(30,20), S(0,20) use the Cohen Sutherland algorithm to determine the visible portion of the line A(10,30) and B(40,0). (2072 TU)

Chapter 3 Finished