DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Course-Computer Graphics & Multimedia (CGM)

UNIT TWO

Transformation

Basic concepts

- Transformation means changing some graphics into something else by applying rules. We can have various types of transformations such as translation, scaling , rotation, shearing, etc.
- When a transformation takes place on a 2D plane, it is called 2D transformation.
- When a transformation takes place on a 3D plane, it is called 3D transformation.

Matrix Representation

Matrix can be represented in 2D as

$$\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \quad \text{OR} \quad \begin{bmatrix} 1 & 4 & 7 \\ 3 & 5 & 8 \\ 2 & 6 & 9 \end{bmatrix} \quad \text{OR} \quad \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

$$C = A.B = \begin{bmatrix} 48 & 60 \\ 57 & 72 \\ 66 & 84 \end{bmatrix}$$

Types of Transformations

There are various types of transformations in computer graphics through which an image can be processed, edited ad altered.

Some basic and most commonly used types of these transformations are:

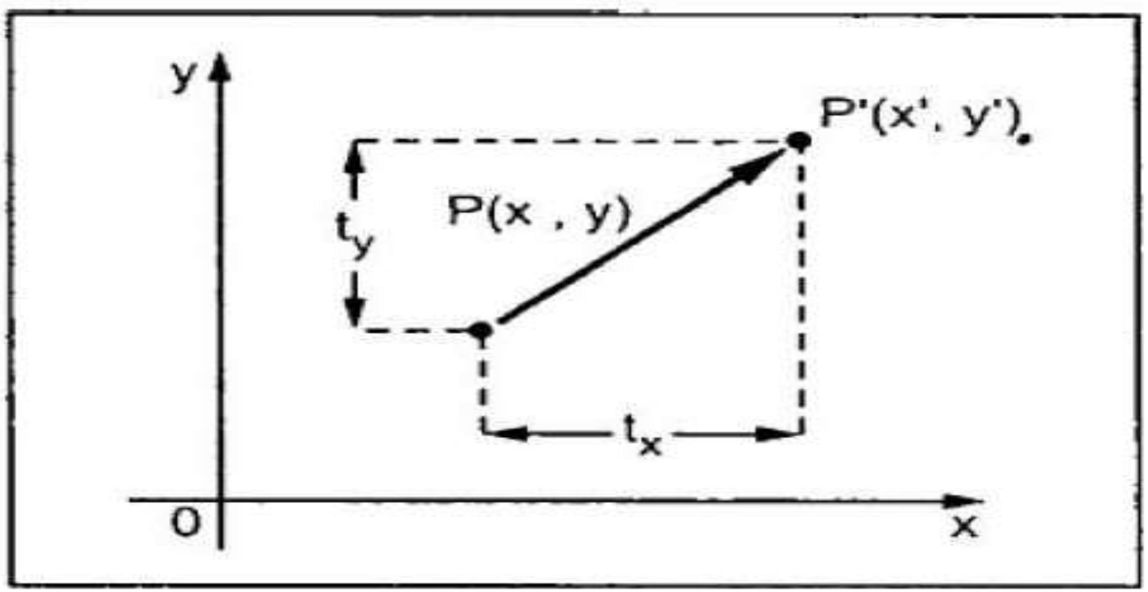
- **Translation:** Repositioning an object
- **Scaling:** Resizing an object
- **Rotation:** Rotating an object in 360^{o}

Other transformation:

- **Reflection:** Mirror image of an object
- **Shearing:** Slanting an object

Translation Transformation

- A translation moves an object to a different position on the screen.
- A translation is applied to an object by repositioning it along a straight line path from one coordinate location to another.
- You can translate a point in 2D by adding translation distance coordinate (tx, ty) 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 as

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

We can write it as −

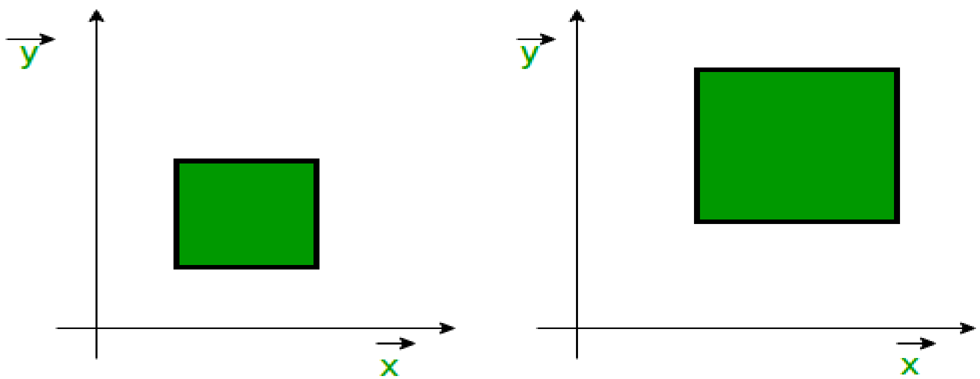
$$P' = P + T$$

Numerical:

Translate the polygon with coordinates A(2,5), B(7,10) and C(10,2) by 3 units in X direction & 4 units in Y direction

Scaling Transformation

- A scaling transformation alters size of an object. In the scaling process, we either compress or expand the dimension of the object.
- Scaling operation can be achieved by multiplying each vertex coordinate (x, y) of the polygon by scaling factor s_x and s_y to produce the transformed coordinates as (x’, y’).



So, $X' = X * S_x$ and

$Y' = Y * S_y$

The scaling factor S_x, S_y scales the object in X and Y direction respectively.

So, the above equation can be represented in matrix form:

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}$$

We can write it as −

P’ = S. P

Depending upon values of scaling factors, we will get different results.

Case 1: If Sx = Sy = 1

Then scaling do not change the size of the object

Case 2: If Sx & Sy > 1

Then scaling stretches the object, increases the size

Case 3: If Sx & Sy < 1

Then scaling reduces the object, decreases the size

Case 4: If Sx = Sy

Then scaling scales the object uniformly in both directions without disturbing the shape

Case 5: If Sx ≠ Sy

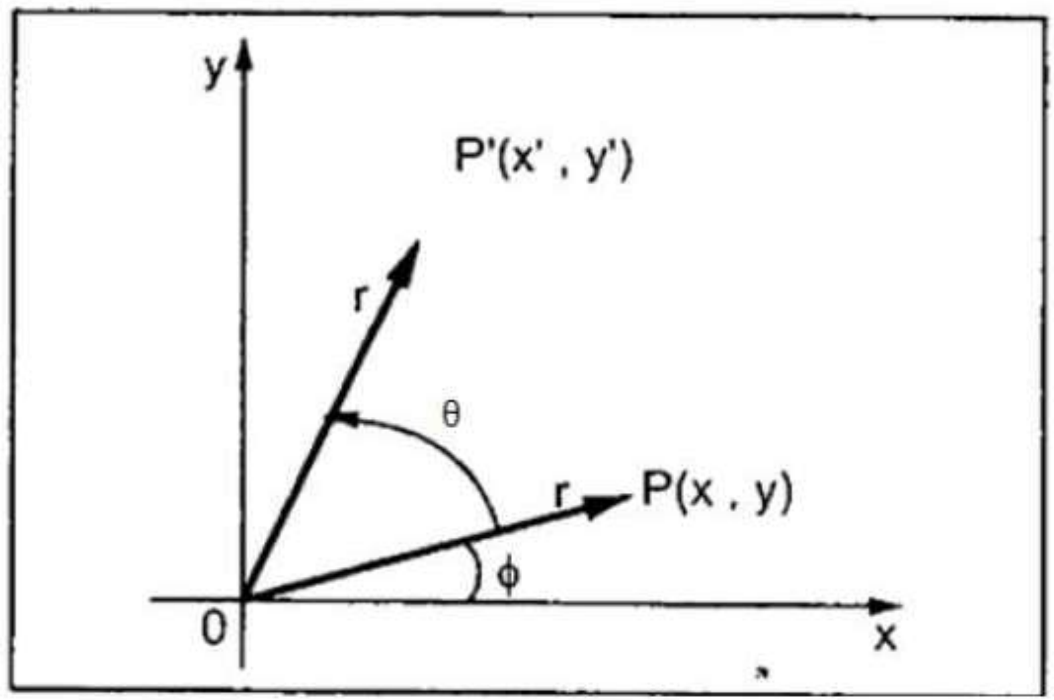
Then scaling scales the object to change its size as well as shape of the object

Numerical:

Scale the polygon with coordinates A (2,5), B (7,10), C (10,2) by 2 units in X direction and 2 units in Y direction

Rotation Transformation

- In Computer graphics, 2D Rotation is a process of rotating an object with respect to an angle in a two dimensional plane.
- 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 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 −

$$X = r\cos\phi \ldots\ldots(1)$$

$$Y = r\sin\phi \ldots\ldots(2)$$

Same way we can represent the point P' (X',Y') as −

$$X' = r\cos(\phi+\theta) = r\cos\phi\cos\theta - r\sin\phi\sin\theta \ldots\ldots(3)$$

$$Y' = r\sin(\phi+\theta) = r\cos\phi\sin\theta + r\sin\phi\cos\theta \ldots\ldots(4)$$

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,

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

$$OR$$

P' = P . 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 following rotation matrix.

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

However, for negative angle rotation, the matrix will change as shown below −

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

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

$$(\because cos(-\theta) = cos\theta \; and \; sin(-\theta) = -sin\theta)$$

Numerical:

1. A polygon with coordinates A (2,2), B (8,2), C (5,5). Rotate the polygon by an angle of 90^o

2. A point (4,3) is rotated counterclockwise by an angle of 45^o. Fine the rotation matrix & resultant point

Homogeneous coordinate system

- In design & picture formation process, many times we may require to perform translation , scaling and rotation to fit the picture components into their proper positions.
- In order to combine sequence of transformations, we have to represent matrix in 3 x 3 format instead of 2 x 2 by introducing an additional dummy coordinate **W.**
- Here points are specified by three numbers instead of two. This co-ordinate system is called as homogeneous co-ordinate system.
- It allows us to express all transformation equations as matrix multiplication.
- The homogeneous co-ordinate is represented by a triplet **(Xw, Yw, W).**

Homogeneous coordinate system

- For 2 D transformations, we can have a homogeneous parameter W to be nonzero value, but it is convenient to have **W=1.**
- Therefore each 2D position can be represented with homogeneous co-ordinate as **(X, Y, 1).**
- 3D graphics hardware can be specified to perform matrix multiplication on 4X4 matrix.

Following are matrix for two-dimensional transformation in homogeneous coordinate:

Translation Transformation:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} \text{ or } \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Scaling Transformation:

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

Rotation Transformation:

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

Clockwise Rotation

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

Anti-clockwise Rotation

Composite Transformation

- A number of transformations or sequence of transformations can be combined into single one called as composition.
- The resulting matrix is called as composite matrix. The process of combining is called as concatenation.

For example:

- If a transformation of the plane T1 is followed by a second plane transformation T2, then the result itself may be represented by a single transformation T which is the composition of T1 and T2 taken in that order.
- **This is written as T = T1·T2**

Rotation about pivot (arbitrary) point

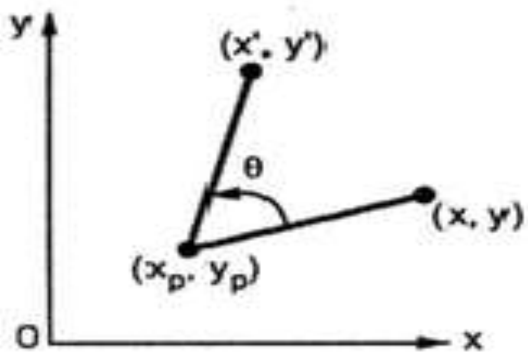
If we want to rotate an object or point about an arbitrary point, we have to carry out the following steps:

1. Translate the point about which we want to rotate to the origin.
2. Then rotate point or object about the origin, and
3. We again translate it to the original place.

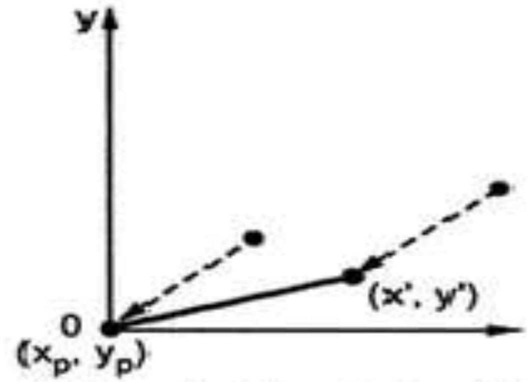
We get rotation about an arbitrary point.

Rotation about pivot (arbitrary) point

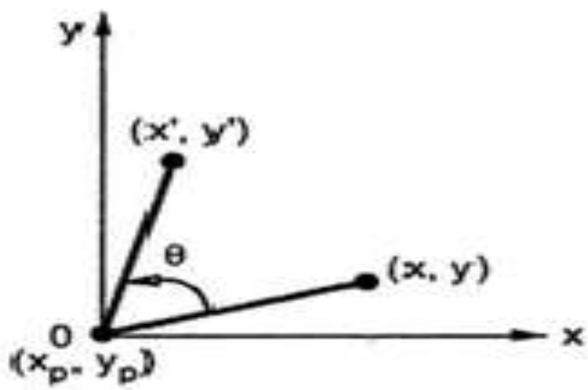
Let us find the transformations matrices to carry out individual steps, to rotate about arbitrary point (Xp, Yp).



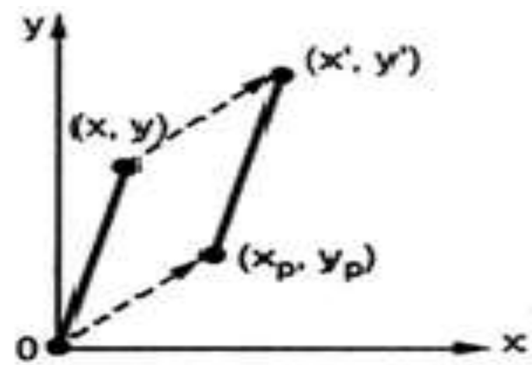
(a) Rotation about an arbitrary point



(b) Step 1 : Translate point (x_p, y_p) to the origin



(c) Step 2 : Rotate it about the origin



(d) Step 3 : Translate back to the original position

Rotation about pivot (arbitrary) point

The translation matrix to move point (x_p, y_p) to the origin is given as,

$$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_p & -y_p & 1 \end{bmatrix}$$

The rotation matrix for counterclockwise rotation of point about the origin is given as,

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

The translation matrix to move the center point back to its original position is given as,

$$T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_p & y_p & 1 \end{bmatrix}$$

Rotation about pivot (arbitrary) point

Therefore, the overall transformation matrix for a counterclockwise rotation by an angle θ about the point (x_p, y_p) is given as,

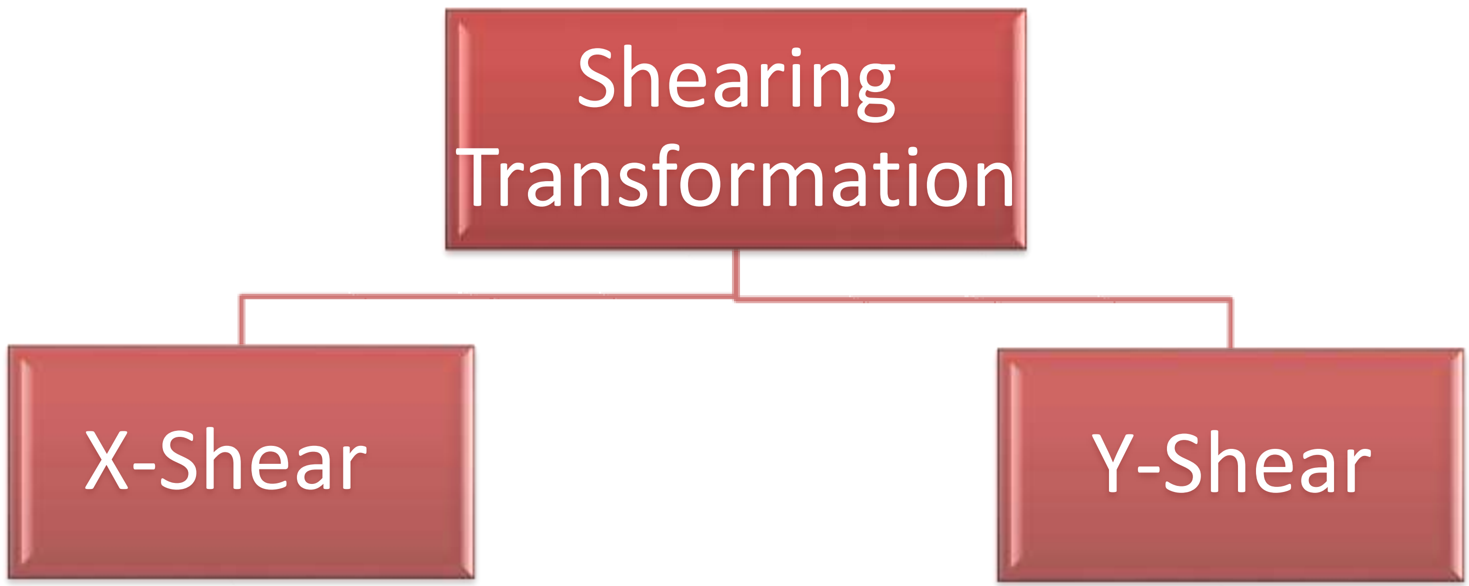
$$T_1 \cdot R \cdot T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_p & -y_p & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_p & y_p & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ -x_p\cos\theta + y_p\sin\theta & -x_p\sin\theta - y_p\cos\theta & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_p & y_p & 1 \end{bmatrix}$$

$$T_1 \cdot R \cdot T_2 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ -x_p\cos\theta + y_p\sin\theta + x_p & -x_p\sin\theta - y_p\cos\theta + y_p & 1 \end{bmatrix}$$

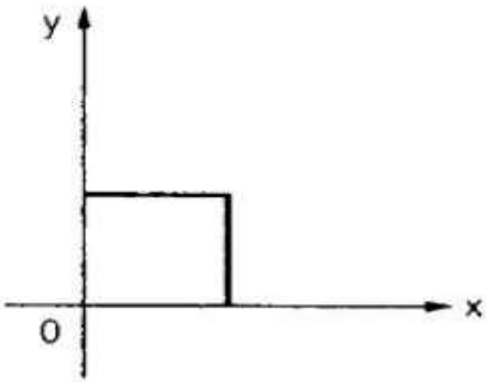
Shearing Transformation

- 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.

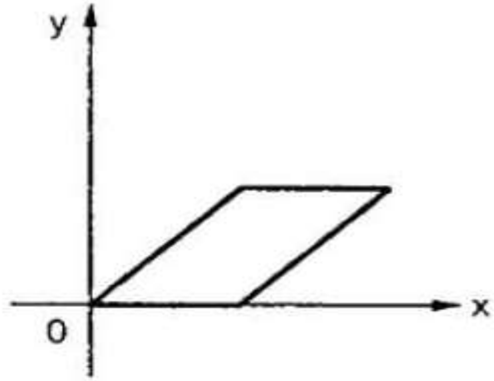


X-Shear

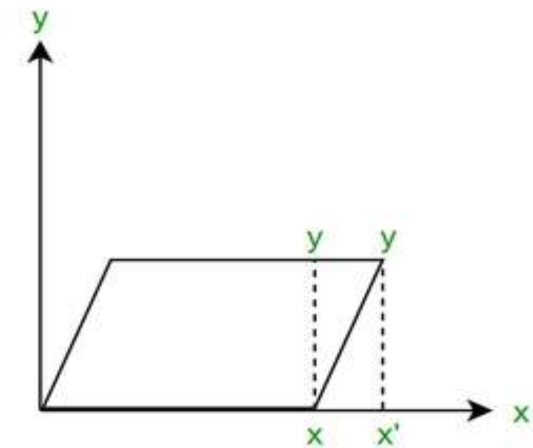
The X-Shear preserves the Y coordinate and changes are made to X coordinates, which causes the vertical lines to tilt right or left as shown in below figure.



(a) Original object



(b) Object after x shear



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

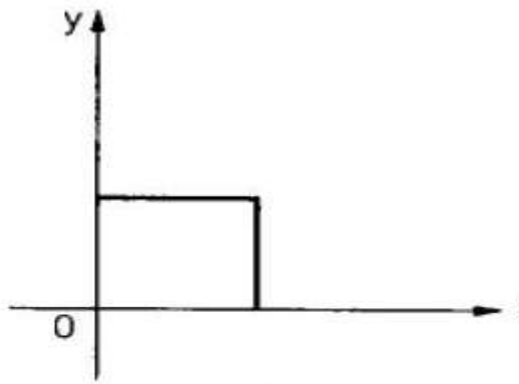
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$X' = X + Sh_x . Y$$

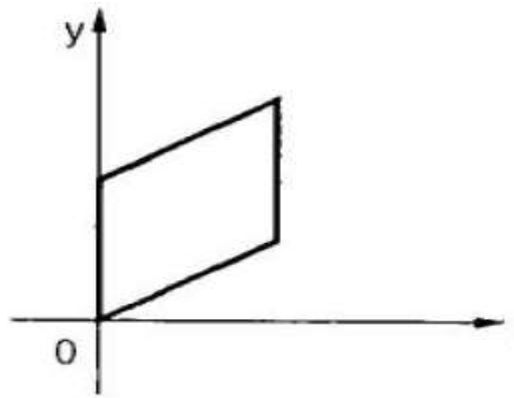
$$Y' = Y$$

Y-Shear:

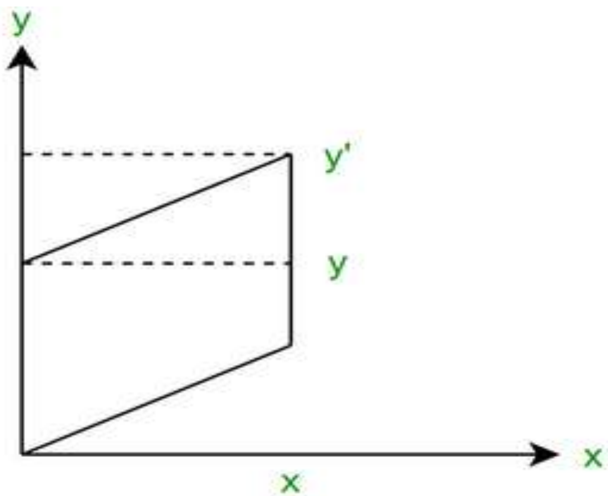
The Y-Shear preserves the X coordinates and changes the Y coordinates which causes the horizontal lines to transform into lines which slopes up or down as shown in the following figure.



(a) Original object



(b) Object after y shear



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

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

$$Y' = Y + Sh_y \,.\, X$$

$$X' = X$$

Shearing relative to other reference line:

We can apply x shear and y shear transformations relative to other reference lines. In x shear transformations we can use y reference line and in y shear we can use x reference line.

X shear with y reference line (Y=Yref):

We can generate x-direction shears relative to other reference lines with the transformation matrix
The transformation matrices is given below.

$$\begin{bmatrix} 1 & sh_x & -sh_x \cdot y_{ref} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$x' = x + sh_x \cdot (y - y_{ref})$$

$$y' = y$$

Y shear with y reference line (X=Xref):

We can generate Y-direction shears relative to other reference lines with the transformation matrix

The transformation matrices is given below.

$$\begin{bmatrix} 1 & 0 & 0 \\ sh_y & 1 & -sh_y \cdot x_{ref} \\ 0 & 0 & 1 \end{bmatrix}$$

$$x' = x$$

$$y' = sh_y \cdot (x - x_{ref}) + y$$

NUMERICAL

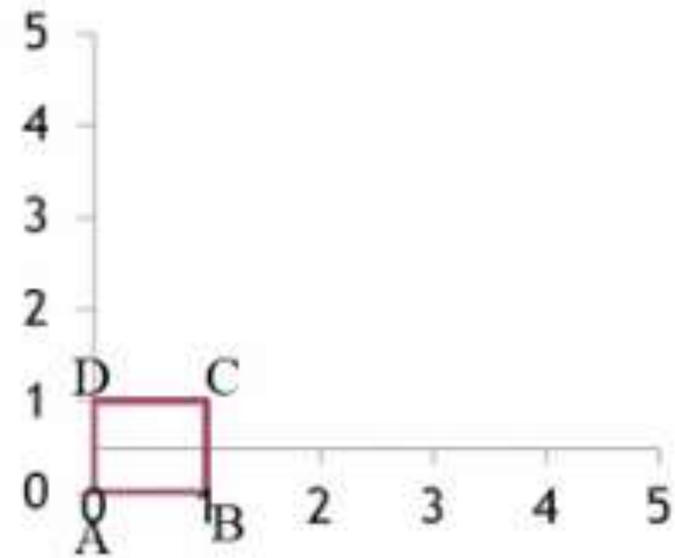
- Apply the shearing transformation to square with A(0,0),B(1,0),C(1,1),D(0,1) as given below
  a) Shear parameter value of 0.5 relative to the line yref=-1
  b) Shear parameter value of 0.5 relative to the line xref=-1

Solution:

Given

shx=0.5

shy=0.5



SOLUTION

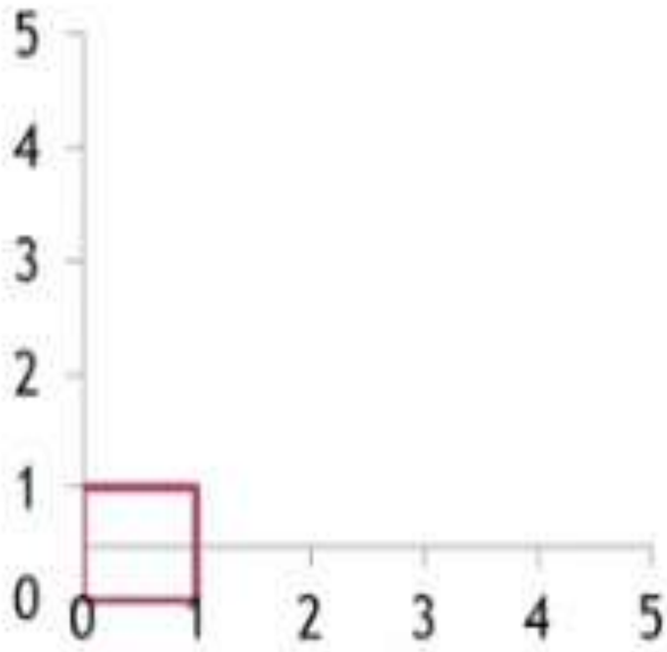
a) x direction shear relative to reference line (y = yref)

shx=0.5,yref=-1

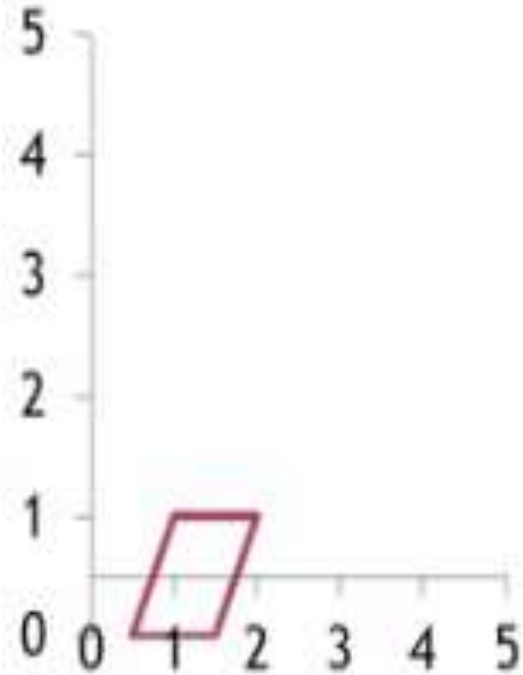
$$\begin{bmatrix} 1 & sh_x & -sh_x \cdot y_{ref} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.5 & 1.5 & 1 & 2 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

BEFORE SHEARING



AFTER SHEARING



SOLUTION

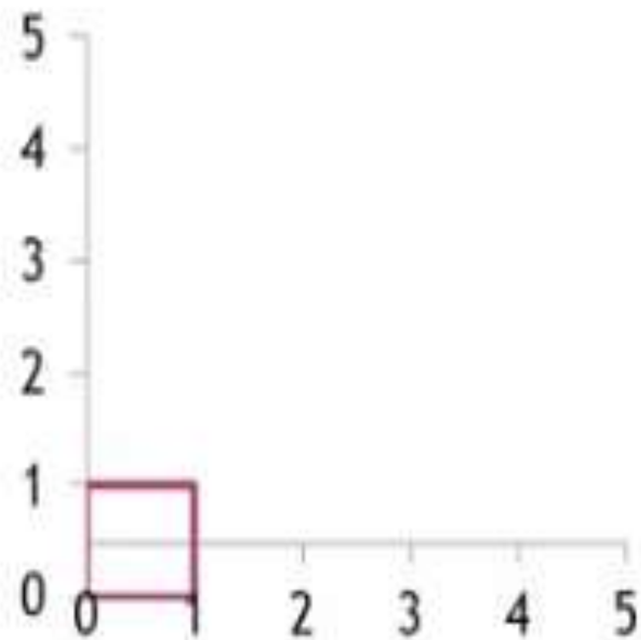
b) y direction shear relative to reference line (x = xref)

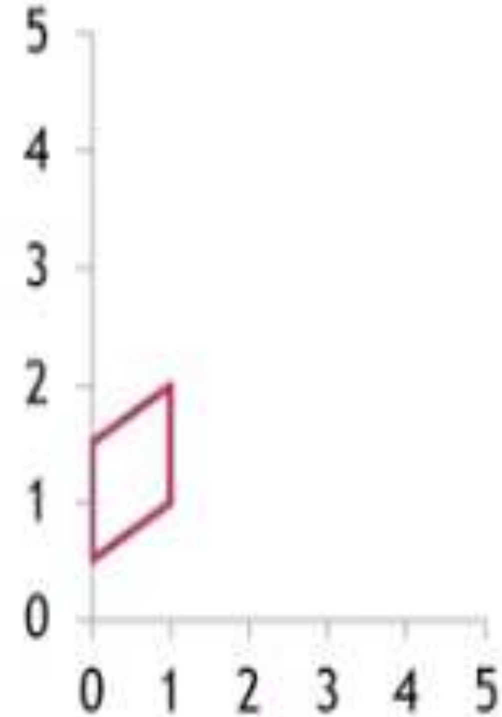
shx=0.5,xref=-1

$$\begin{bmatrix} 1 & 0 & 0 \\ sh_y & 1 & -sh_y \cdot x_{ref} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0.5 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0.5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0.5 & 1 & 1.5 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

BEFORE SHEARING AFTER SHEARING





Reflection Transformation:

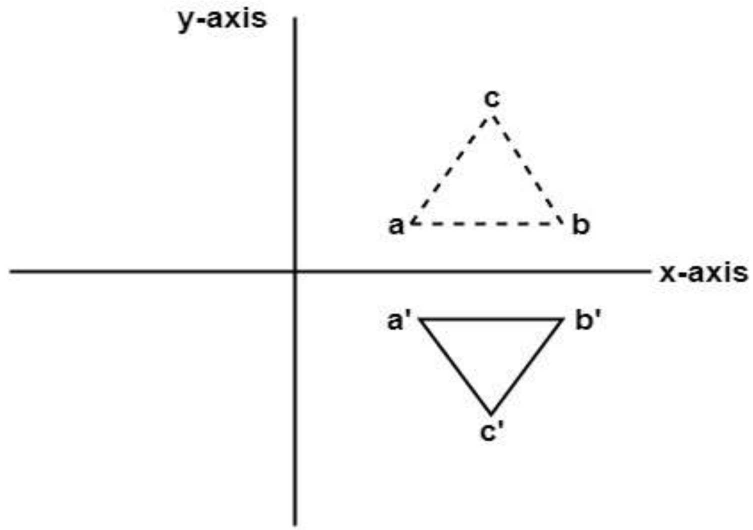
It is a transformation which produces a mirror image of an object. The mirror image can be either about x-axis or y-axis. The object is rotated by180°.

Types of Reflection:

- Reflection about the x-axis
- Reflection about the y-axis
- Reflection about an axis perpendicular to xy plane and passing through the origin
- Reflection about line y=x

Reflection about the x-axis:

In this transformation value of x will remain same whereas the value of y will become negative.
Following figures shows the reflection of the object axis. The object will lie another side of the x-axis.



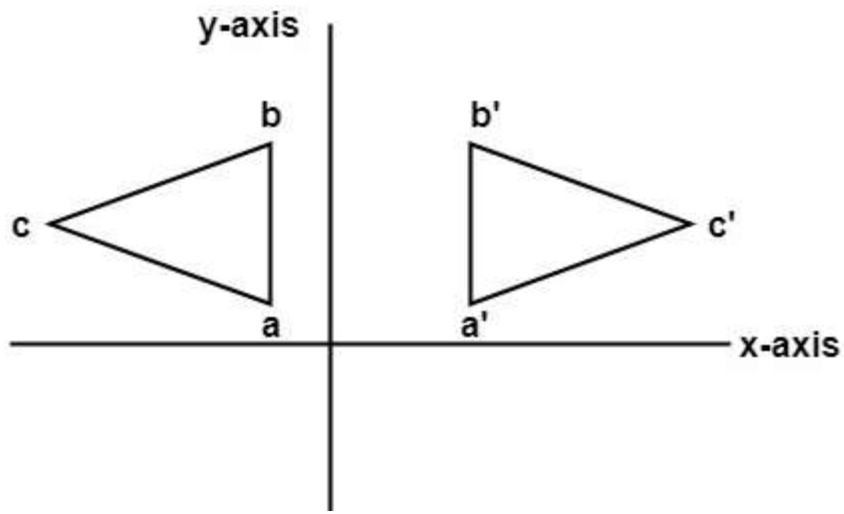
The object can be reflected about x-axis with the help of the following matrix

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

Reflection about the Y-axis

Here the values of x will be reversed, whereas the value of y will remain the same. The object will lie another side of the y-axis.

The following figure shows the reflection about the y-axis



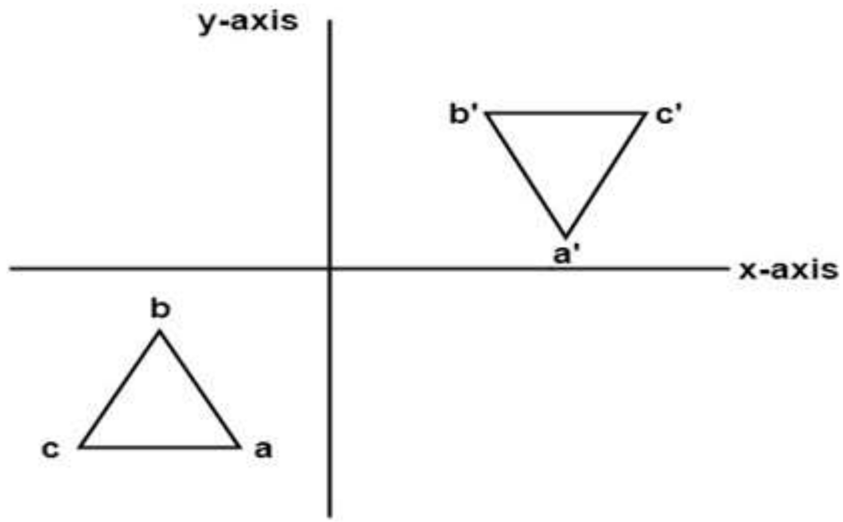
The object can be reflected about y-axis with the help of following transformation matrix

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

Reflection about origin

(An axis perpendicular to xy plane and passing through origin):

In this value of x and y both will be reversed. This is also called as half revolution about the origin.



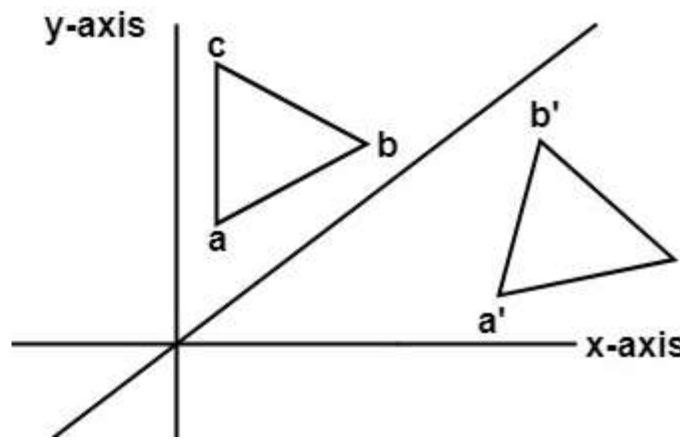
In the matrix of this transformation is given below

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

Reflection about line y=x:

First of all, the object is rotated at 45°. The direction of rotation is clockwise. After it reflection is done concerning x-axis.

The last step is the rotation of y=x back to its original position that is counterclockwise at 45°.

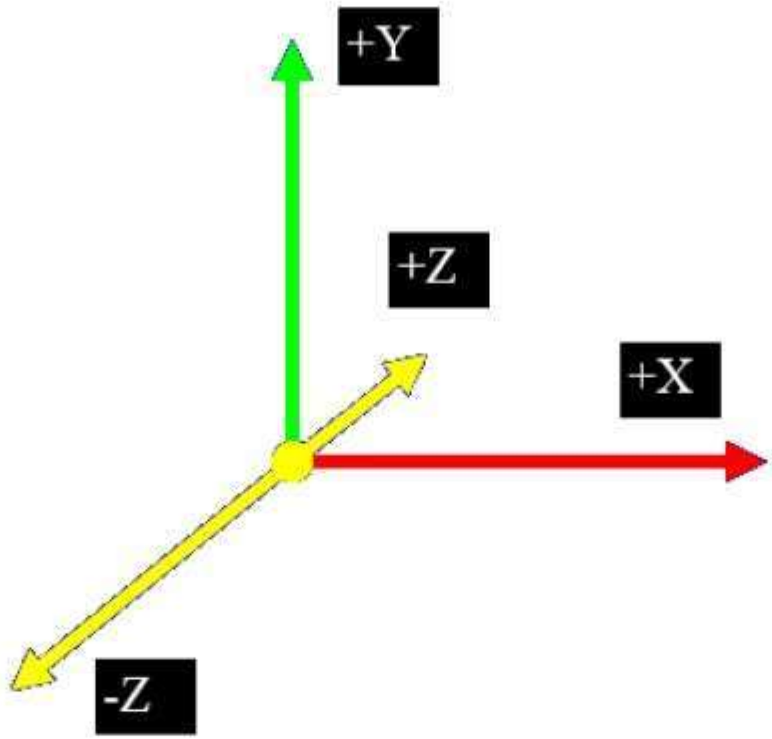


The object may be reflected about line y = x with the help of following transformation

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

3D Transformation:

In the 2D system, we use only two coordinates X and Y but in 3D, an extra coordinate Z is added.
3D graphics techniques and their application are fundamental to the entertainment, games, and computer-aided design industries. It is a continuing area of research in scientific visualization.



3D Transformation:

So when transformation takes place on a 3D plane, it is called as 3D transformation.

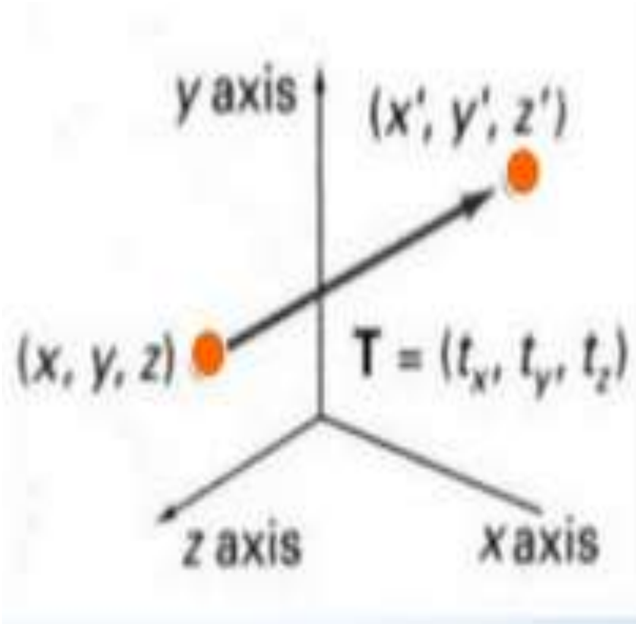
There are three basic types of **3D transformation**:

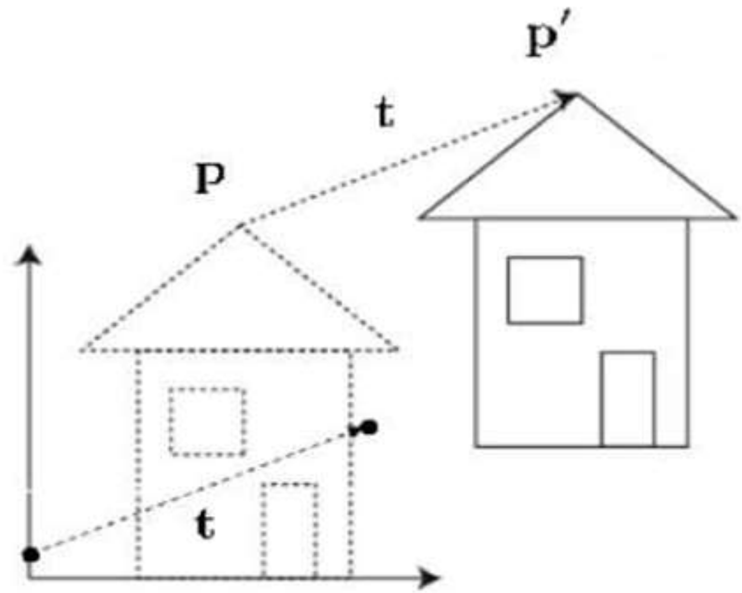
- **Translation:** Repositioning an object
- **Scaling:** Resizing an object
- **Rotation:** Rotating an object in 360^{o}

3D Translation:

Moving of object is called translation

In 3D translation, we transfer the Z coordinate along with the X and Y coordinates. The process for translation in 3D is similar to 2D translation. A translation moves an object into a different position on the screen.
The following figure shows the effect of translation –





A point can be translated in 3D by adding translation coordinate (tx, ty, tz) to the original coordinate X,Y,Z to get the new coordinate X′,Y′,Z′.

3D Translation:

This can be mathematically represented as shown below –

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

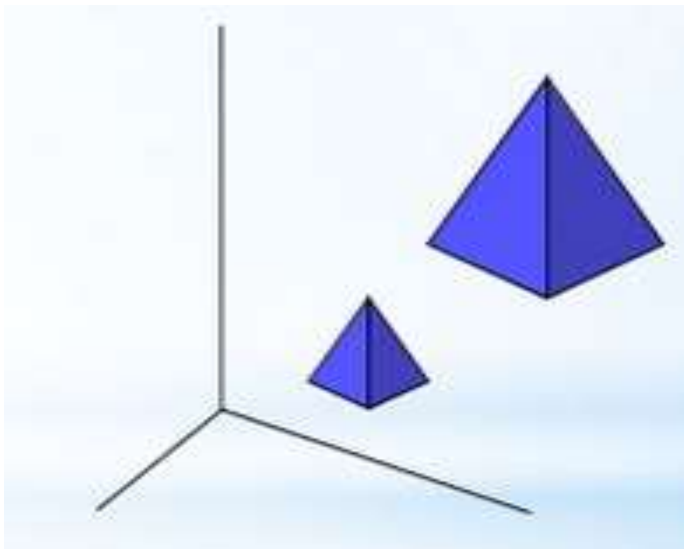
$$P' = P \cdot T$$

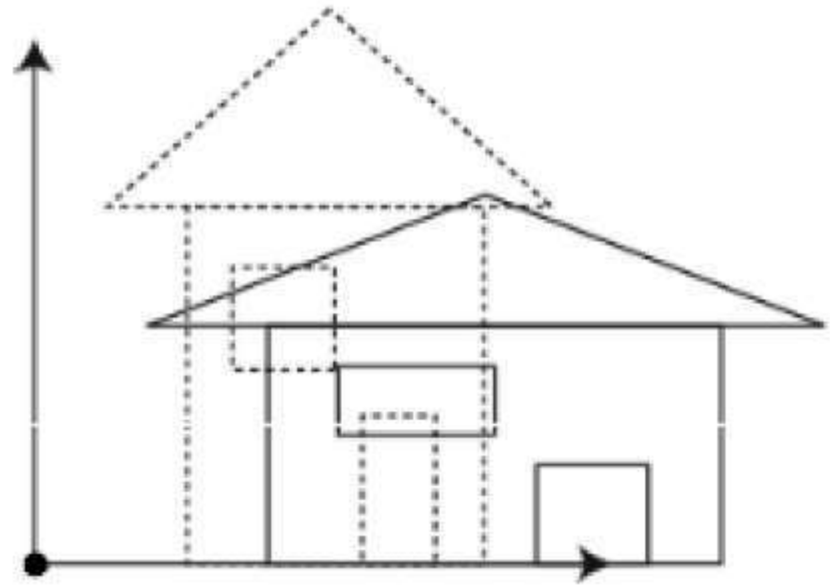
$$[X' \; Y' \; Z' \; 1] = [X \;\; Y \;\; Z \;\; 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

$$[X' \; Y' \; Z' \; 1] = [X + t_x \;\; Y + t_y \;\; Z + t_z \;\; 1]$$

3D Scaling:

You can change the size of an object using scaling transformation.
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.
The following figure shows the effect of 3D scaling –





In 3D scaling operation, three coordinates are used. Let us assume that the original coordinates are X,Y,Z, scaling factors are (SX, SY, Sz) respectively, and the produced coordinates are X′,Y′,Z′.

3D Scaling:

This can be mathematically represented as shown below –

$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

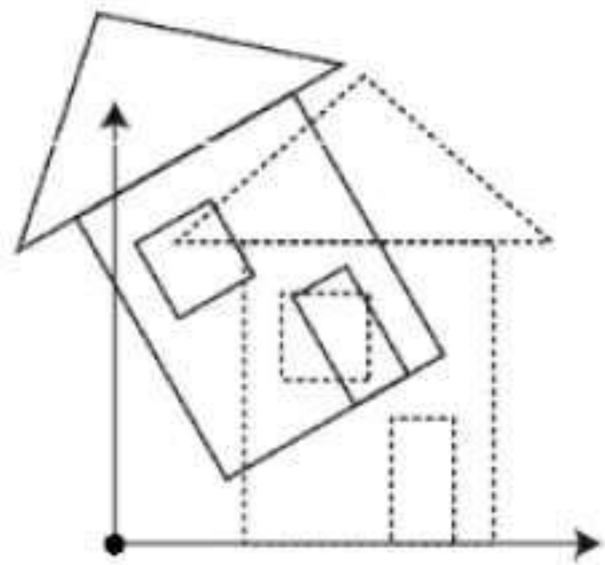
$$P' = P \cdot S$$

$$[X' \ \ Y' \ \ Z' \ \ 1] = [X \ \ Y \ \ Z \ \ 1] \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[X' \ \ Y' \ \ Z' \ \ 1] = [X.S_x \ \ Y.S_y \ \ Z.S_z \ \ 1]$$

3D Rotation:

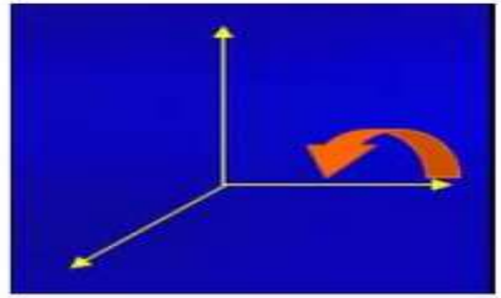
3D rotation is not same as 2D rotation. In 3D rotation, we have to specify the angle of rotation along with the axis of rotation. We can perform 3D rotation about X, Y, and Z axes.



3D Rotation:

They are represented in the matrix form as below –

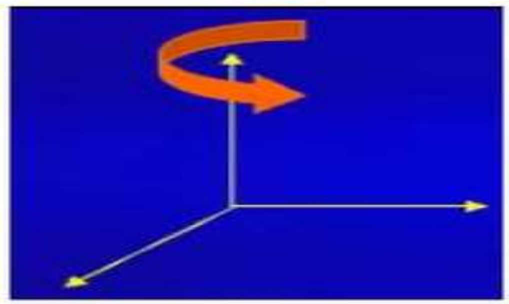
$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta & -sin\theta & 0 \\ 0 & sin\theta & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Rotation about x-axis

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

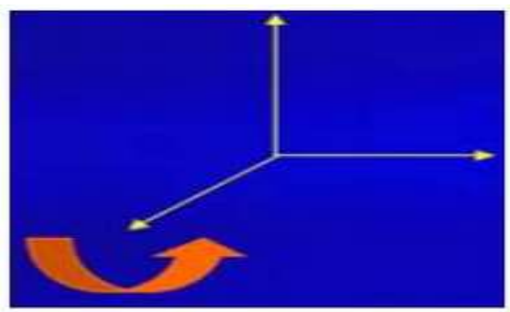
$$R_y(\theta) = \begin{bmatrix} cos\theta & 0 & sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -sin\theta & 0 & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Rotation about y-axis

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

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



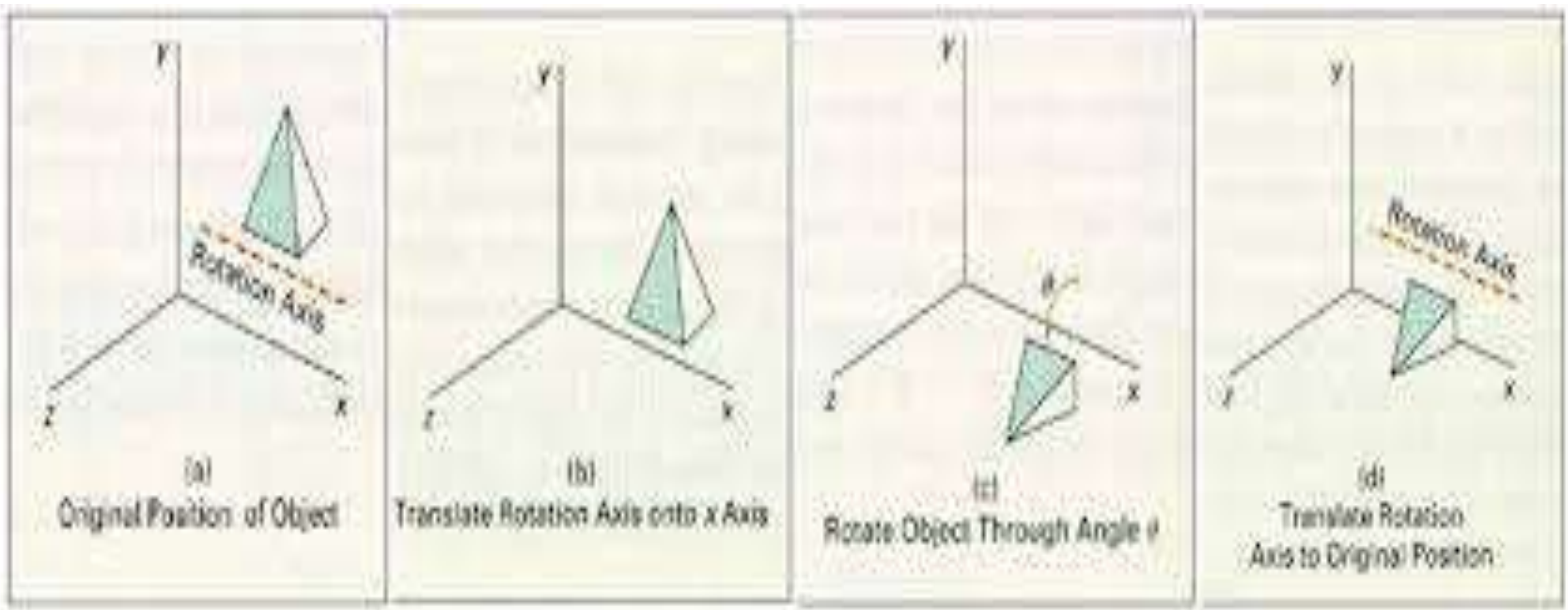
Rotation about z-axis

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

Rotation about Arbitrary Axis:

When the object is rotated about an axis that is not parallel to any one of co-ordinate axis, i.e., x, y, z. Then additional transformations are required. First of all, alignment is needed, and then the object is being back to the original position. Following steps are required

- Translate the object to the origin
- Rotate object so that axis of object coincide with any of coordinate axis.
- Perform rotation about co-ordinate axis with whom coinciding is done.
- Apply inverse rotation to bring rotation back to the original position.



(a) Original Position of Object

(b) Translate Rotation Axis onto x Axis

(c) Rotate Object Through Angle θ

(d) Translate Rotation Axis to Original Position

Rotation about Arbitrary Axis:

The actual transformation for a rotation θ about an arbitrary axis is given by the product of above transformation

$$R_{\theta}=TRxRyRzRy^{-1}Rx^{-1}T^{-1}$$