

## **Chapter 5 – Geometrical Transformations**

### **2.1 Introduction to 2D Transformation**

Transformation means changing some graphics into something else by applying rules. We can have various types of transformations such as translation, scaling up or down, rotation, shearing, etc. We require moving pictures for example games, animation, and different model where we show certain objects moving or rotating or changing their sizes. When a transformation takes place on a 2D plane, it is called 2D transformation. Transformations play an important role in computer graphics to reposition the graphics on the screen and change their size or orientation.

#### **Homogeneous Coordinates**

To perform a sequence of transformation such as **translation** followed by **rotation** and **scaling**, we need to follow a sequential process –

- Translate the coordinates,
- Rotate the translated coordinates, and then
- Scale the rotated coordinates to complete the composite transformation.

To shorten this process, we have to use  $3 \times 3$  transformation matrix instead of  $2 \times 2$  transformation matrix. To convert a  $2 \times 2$  matrix to  $3 \times 3$  matrix, we have to add an extra dummy coordinate W.

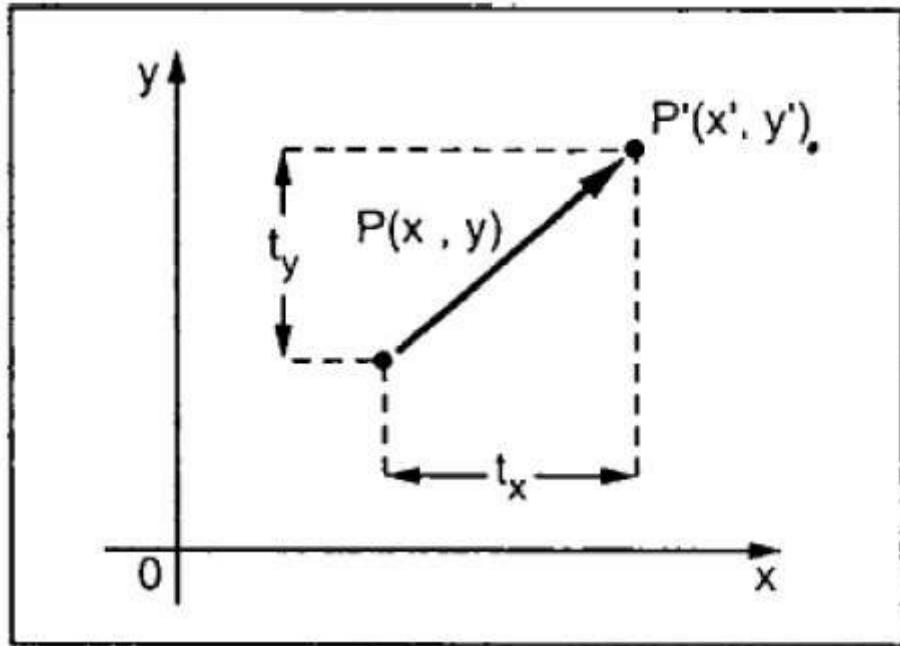
In this way, we can represent the point by 3 numbers instead of 2 numbers, which is called **Homogenous Coordinate** system. In this system, we can represent all the transformation equations in matrix multiplication. Any Cartesian point  $P(X, Y)$  can be converted to homogenous coordinates by:

$$P' (X_h, Y_h, h).$$

Basically there are 5 types of 2D Transformations such as Translation, Rotation, Scaling, Sheer and Reflection.

#### **2.1.1 Translation**

A translation moves an object to a different position on the screen (i.e change in the position). 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 –

$$\mathbf{X}' = \mathbf{X} + \mathbf{t}_x$$

$$\mathbf{Y}' = \mathbf{Y} + \mathbf{t}_y$$

The pair  $(\mathbf{t}_x, \mathbf{t}_y)$  is called the translation vector or shift vector. The above equations can also be represented using the column vectors.

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

We can write it as –

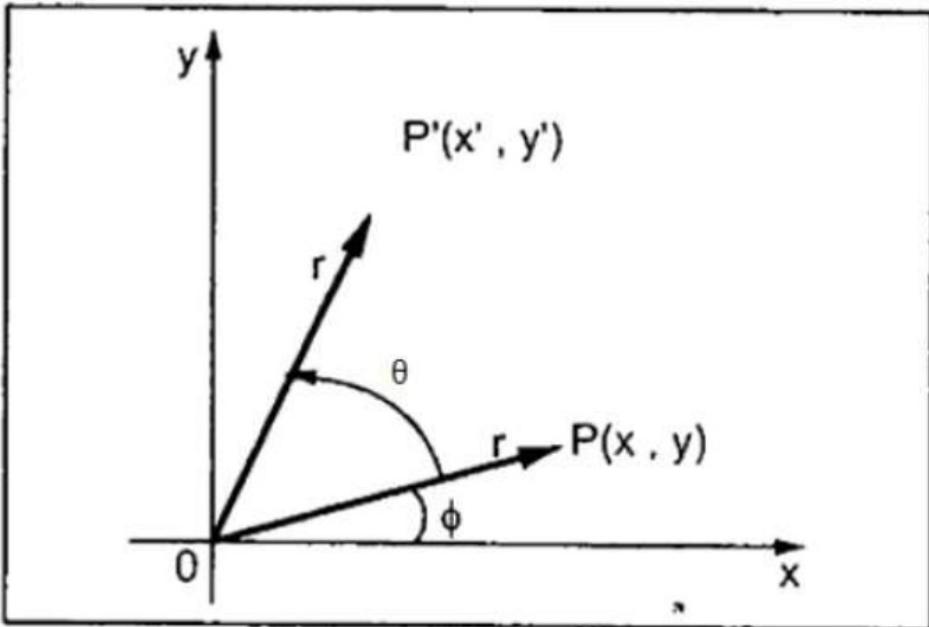
$$\mathbf{P}' = \mathbf{P} + \mathbf{T}$$

### 2.1.2 Rotation

In rotation, we rotate the object at particular angle  $\theta$  (theta) from its origin. From the following figure, we can see that the point  $P(X, Y)$  is located at angle  $\phi$  from the horizontal X coordinate with distance  $r$  from the origin.

Let us suppose you want to rotate it at the angle  $\theta$ . After rotating it to a new location, you will get a new point

$$\mathbf{P}' (\mathbf{X}', \mathbf{Y}').$$



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,

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

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

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 –

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

### 2.1.3 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 –

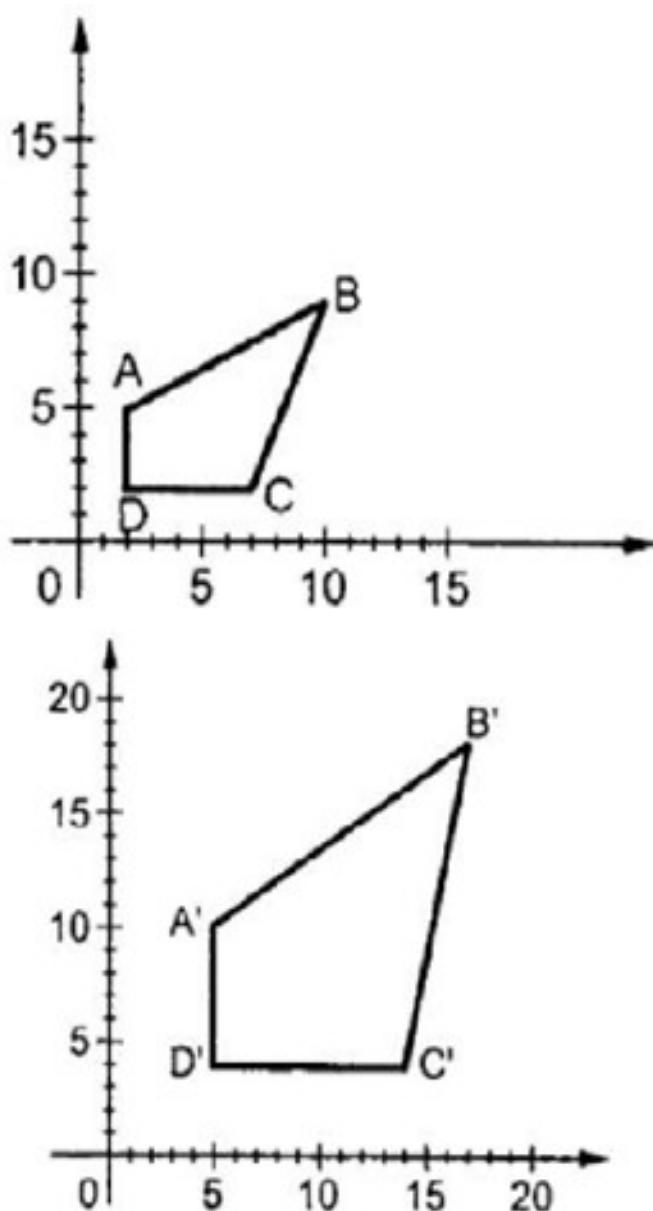
$$X' = X \cdot S_x \quad \text{and} \quad Y' = Y \cdot S_y$$

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 –

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

$$\text{Or} \quad \mathbf{P}' = \mathbf{P} \cdot \mathbf{S}$$

Where  $\mathbf{S}$  is the scaling matrix. The scaling process is shown in the following figure.



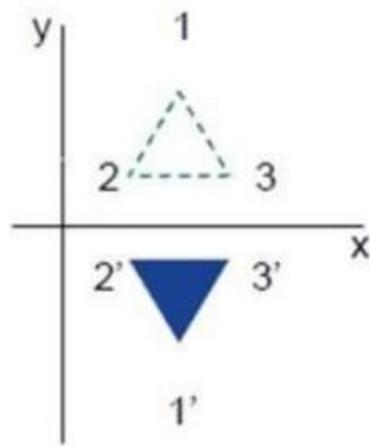
If we provide values less than 1 to the scaling factor S, then we can reduce the size of the object.

If we provide values greater than 1, then we can increase the size of the object.

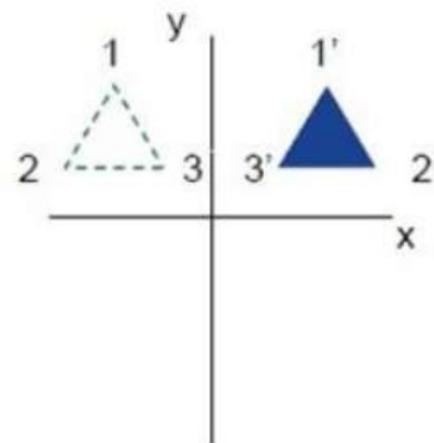
#### 2.1.4 Reflection

Reflection is the mirror image of original object. In other words, we can say that it is a rotation operation with  $180^\circ$ . In reflection transformation, the size of the object does not change.

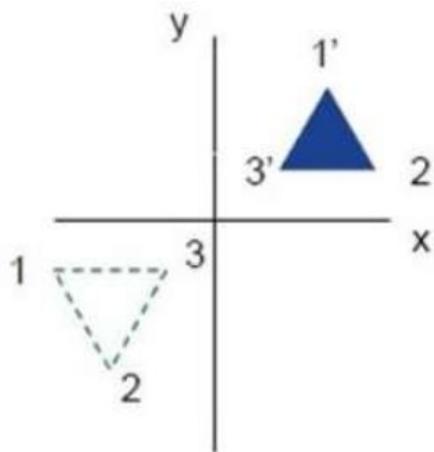
The following figures show reflections with respect to X and Y axes, and about the origin respectively.



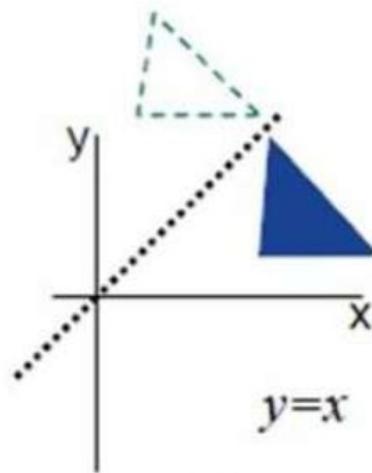
(a)



(b)



(c)



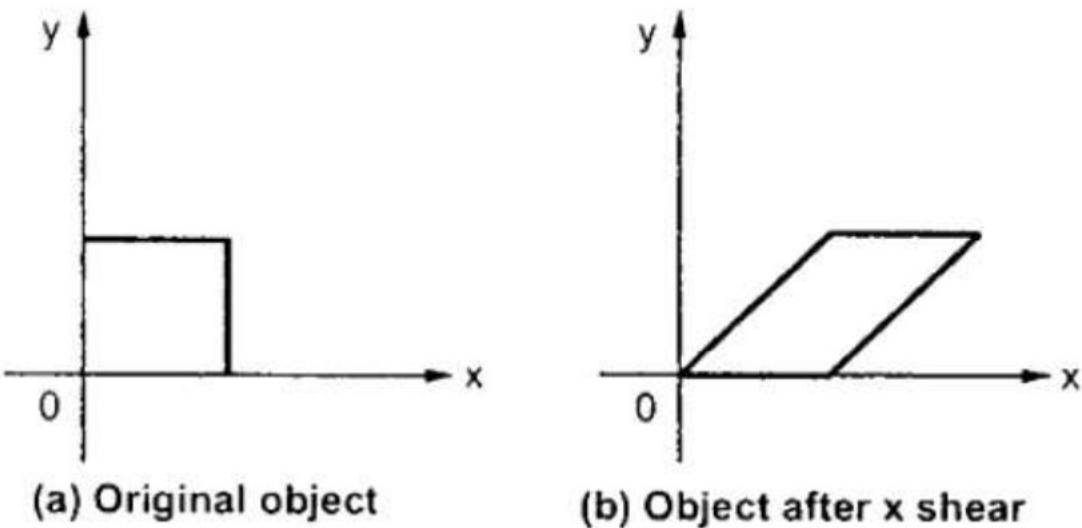
(d)

### 2.1.5 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 are made to X coordinates, which causes the vertical lines to tilt right or left as shown in below figure.



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

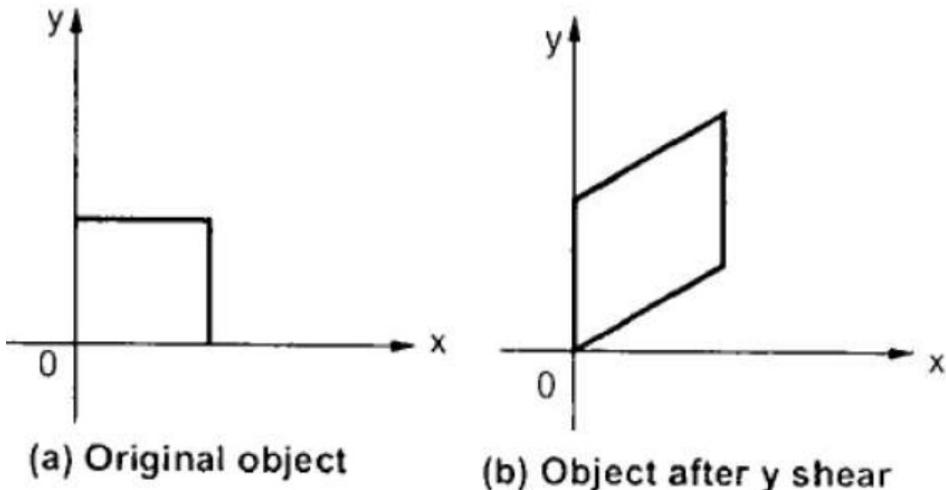
$$X_{sh} = \begin{bmatrix} 1 & shx & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

$$X' = X$$

### **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 –

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

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

$$Y' = Y$$

## 2.2 Composite Transformation

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  $T = T_1 \cdot T_2$ .

Composite transformation can be achieved by concatenation of transformation matrices to obtain a combined transformation matrix.

A combined matrix –

$$[T][X] = [X] [T_1] [T_2] [T_3] [T_4] \dots [T_n]$$

Where  $[T_i]$  is any combination of

- Translation
- Scaling
- Shearing
- Rotation
- Reflection

The change in the order of transformation would lead to different results, as in general matrix multiplication is not cumulative, that is  $[A] \cdot [B] \neq [B] \cdot [A]$  and the order of multiplication. The basic purpose of composing transformations is to gain efficiency by applying a single composed transformation to a point, rather than applying a series of transformation, one after another.

For example, to rotate an object about an arbitrary point  $(X_p, Y_p)$ , we have to carry out three steps –

- Translate point  $(X_p, Y_p)$  to the origin.
- Rotate it about the origin.
- Finally, translate the centre of rotation back where it belonged.

## 2.3 Introduction to 3D Transformation

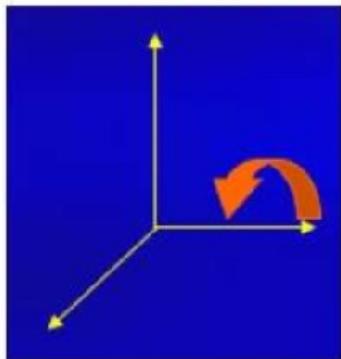
### 2.3.1 Rotation

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. They are represented in the matrix form as below –

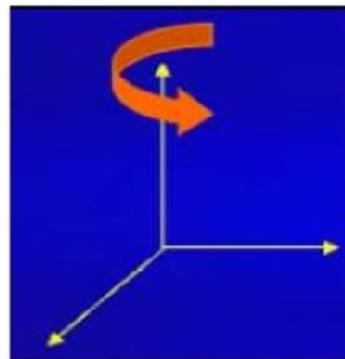
$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} R_z(\theta)$$

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

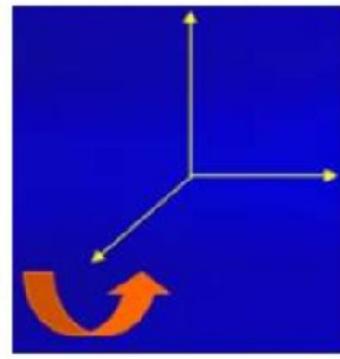
The following figure explains the rotation about various axes –



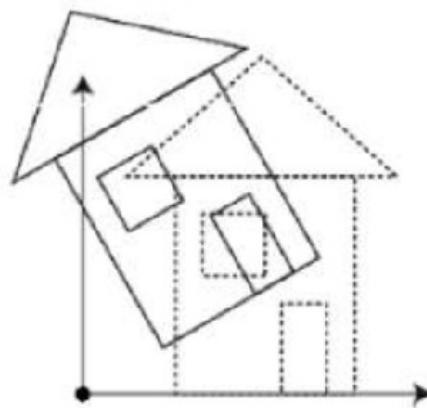
**Rotation about x-axis**



**Rotation about y-axis**

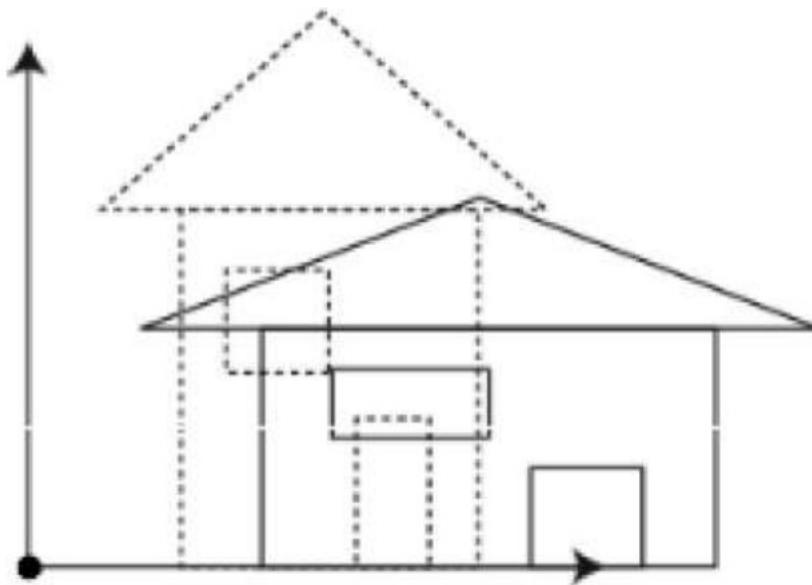


**Rotation about z-axis**



### 2.3.2 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  $(X', Y', Z')$ . This can be mathematically represented as shown below-

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

$$P' = P \cdot S$$

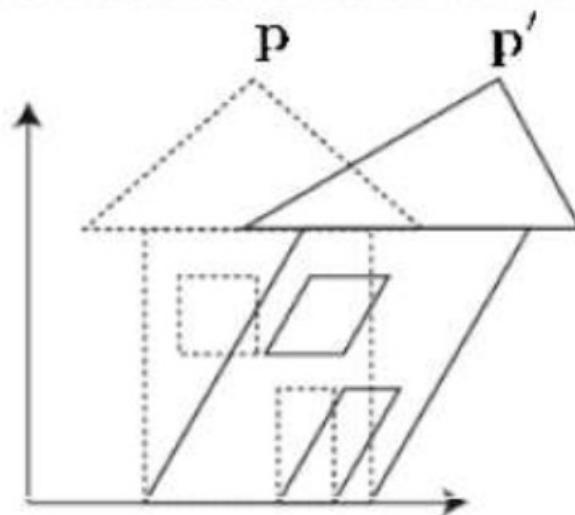
$$\begin{bmatrix} X' & Y' & Z' & 1 \end{bmatrix} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= [X \cdot S_x \quad Y \cdot S_y \quad Z \cdot S_z \quad 1]$$

### 2.3.3 Shear

A transformation that slants the shape of an object is called the **shear transformation**. Like in 2D shear, we can shear an object along the X-axis, Y-axis, or Z-axis in 3D.

# Shear



As shown in the above figure, there is a coordinate  $P$ . You can shear it to get a new coordinate  $P'$ , which can be represented in 3D matrix form as below –

$$Sh = \begin{bmatrix} 1 & sh_x^y & sh_x^z & 0 \\ sh_y^x & 1 & sh_y^z & 0 \\ sh_z^x & sh_z^y & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P' = P \cdot Sh$$

$$X' = X + Sh_x^y Y + Sh_x^z Z$$

$$Y' = Sh_y^x X + Y + sh_y^z Z$$

$$Z' = Sh_z^x X + Sh_z^y Y + Z$$

## 2.4 Transformation Matrices

Transformation matrix is a basic tool for transformation. A matrix with  $n \times m$  dimensions is multiplied with the coordinate of objects. Usually  $3 \times 3$  or  $4 \times 4$  matrices are used for transformation. For example, consider the following matrix for various operations.

$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$	$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$Sh = \begin{bmatrix} 1 & sh_x^y & sh_x^z & 0 \\ sh_y^x & 1 & sh_y^z & 0 \\ sh_z^x & sh_z^y & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
<b>Translation Matrix</b>	<b>Scaling Matrix</b>	<b>Shear Matrix</b>
<b>Rotation Matrix</b>		

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

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

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