2-D Transformation

Chapter Outline

8.1. INTRODUCTION

It is essential for a graphics system to allow the user to change the way objects appear. In the real world we frequently rearrange objects or look at them from different angles. The effects that we desire include changing the size of the object, its position on the screen, or its orientation. The implementation of such a change is called *transformation*. The term 'transform' means 'to change'. This change can either be of shape, size or position of the object. To perform transformation on any object, object matrix 'X' is multiplied by the transformation matrix T.

$$\begin{bmatrix}\text{Transformed object}\\ \text{matrix}\end{bmatrix} = \begin{bmatrix}\text{Original object}\\ \text{matrix}\end{bmatrix} \times \begin{bmatrix}\text{Transformation}\\ \text{matrix}\end{bmatrix}$$

or

$$[X^*] = [X]\ [T]$$

The various transformations possible on an object are as follows:

- Rotation
- Scaling
- Shearing etc.
- Reflection
- Translation

Each of the above transformation is carried out by using a different transformation matrix.

The fundamental objective of 2-D transformation is to simulate the movement and manipulation of objects in the plane. Points and lines which join them, along with appropriate drawing algorithm are used to represent objects. The ability to transform these points and lines is achieved by translation, rotating, scaling and reflection. Two points of view are used for describing the object movement. The first is that the object itself is moved relative to a stationary co-ordinate system or background. The mathematical statement of this view is described by *geometric transformations* applied to each point of the object. The second point of view holds that the object is held stationary, while the co-ordinate system is moved relative to the object. This effect is described by *co-ordinate transformation.*

Thus transformation is described in two categories:

(*a*) Geometric transformation

(*b*) Co-ordinate transformation

The transformations are used directly by application programs and within many graphic subroutines.

8.2. REPRESENTATION OF POINTS OR OBJECTS IN MATRIX FORM

In 2D co-ordinate system, any point is represented in terms of x and y co-ordinates. The point (x, y) can be converted into matrix in the following two ways:

(*a*) **Row-major matrix**

$$[x \quad y]_{1\times 2}$$

(*b*) **Column-major matrix**

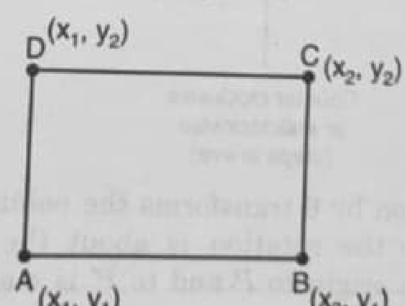
$$\begin{bmatrix} x \\ y \end{bmatrix}_{2\times 1}$$

The above two matrices are frequently called *position vectors.* A series of points, each of which is a position vector relative to some co-ordinate system, is stored in a computer as a matrix or array of numbers. The position of these points is controlled by manipulating the matrix which defines the points. Lines are drawn between the points to generate lines, curves or pictures.

Suppose, we represent a rectangle in matrix form. Let (x_1, y_1) and (x_2, y_2) be the opposite vertices of a rectangle. Then, the four vertices of rectangle will be: $(x_1\ y_1)$, $(x_2\ y_1)$, $(x_2\ y_2)$, (x_1, y_2).

In order to represent this rectangle in matrix form as

$$\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_1 \\ x_2 & y_2 \\ x_1 & y_2 \end{bmatrix}$$



8.3. GEOMETRIC TRANSFORMATIONS

Changes in orientation, size and shape are accomplished with geometric transformations that alter the co-ordinate descriptions of objects. The basic geometric transformations are translation, rotation and scaling. Other transformations that are often applied to objects include reflection and shear.

8.3.1. Translation *(UPTU 2009)*

Translation consists of a shift of the object parallel to itself in any direction in the (x, y) plane. Any such shift can be accomplished by a shift in x-direction plus a shift in y-direction.

If the amount of x-shift is called t_x and the amount of y-shift t_y, the translation of the point (x, y) into the point (x', y') is expressed by the formulas.

$$x' = x + t_x$$
$$y' = y + t_y$$
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

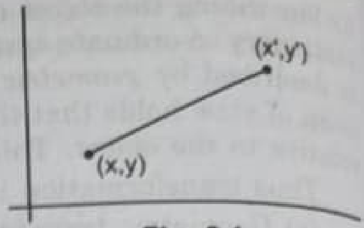


Fig. 8.1.

To translate an object with multiple points, we just translate each point individually and connect them together.

Translation is a rigid-body transformation that moves objects without deformation. That is, every point on the object is translated by the same amount. Translation is the only transformation that is not in relation to a reference point. Its effect is independent of the original position of the object.

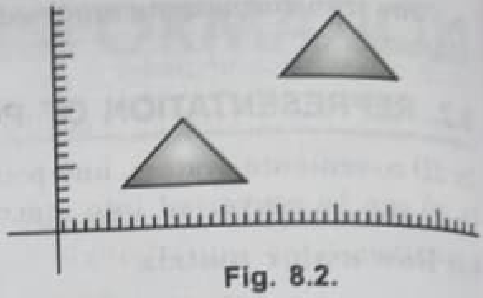
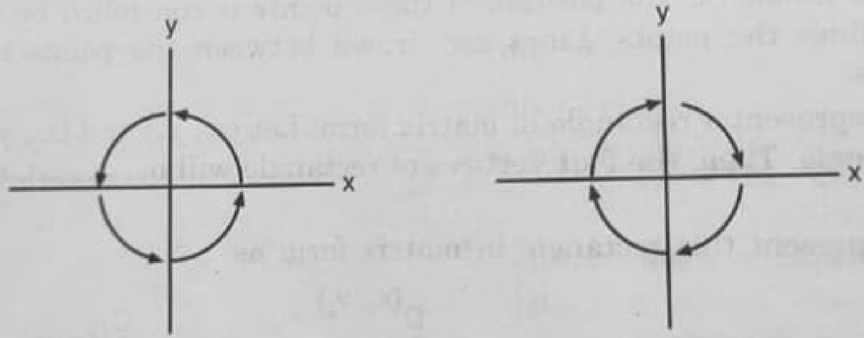


Fig. 8.2.

(UTPU 2009)

8.3.2. Rotation

A two-dimensional rotation is applied to an object by repositioning it along a circular path in the xy plane. Points can be rotated through an angle θ about the origin. The sign of angle determines the direction of rotation. Positive values for the rotation angle defines counterclockwise rotations and negative values rotate objects in clockwise directions.



Counter clockwise or anticlockwise (angle is +ve)

Clockwise (angle is – ve)

Suppose rotation by θ transforms the point $P(x, y)$ into $P'(x', y')$. Because the rotation is about the origin, the distances from the origin to P and to P' is r are equal.

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

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

Put the values of x and y, we get

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

In matrix form (Row-major order)

$$[X'] = [X]\,[T] \text{ or } [x' \quad y'] = [x \quad y]\,[T]$$

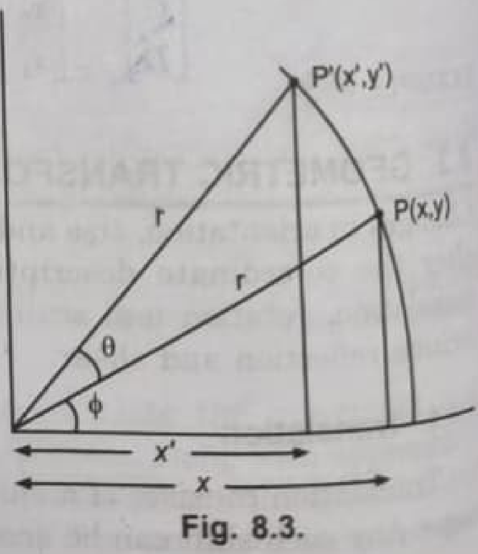


Fig. 8.3.

Put the value of x' and y', we get

$[x \cos\theta - y \sin\theta \;\; x \sin\theta + y \cos\theta] = [x \;\; y]\,[T]$

Hence
$$[T] = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

$$|T| = \cos^2\theta + \sin^2\theta = 1. \quad \text{i.e.} \quad [x' \; y'] = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} [x \;\; y]$$

Note: Determinant of a pure rotation matrix is always + 1.

In column-major order

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

Suppose we want to rotate P' back to point P *i.e.*, to perform the inverse transformation or rotation, the required angle is $-\theta$. then,

$$[T] = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

$$[T^{-1}] = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) \\ -\sin(-\theta) & \cos(-\theta) \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Now
$$[T]\,[T^{-1}] = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$= \begin{bmatrix} \cos^2\theta + \sin^2\theta & -\cos\theta\sin\theta + \sin\theta\cos\theta \\ \cos\theta\sin\theta + \sin\theta\cos\theta & \sin^2 + \cos^2\theta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [1]$$

Thus, the transformation matrix for rotation in

- Anticlockwise direction will be

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

- Clockwise direction will be

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

EXAMPLE 8.1.

Write 2 × 2 transformation matrix for each of the following rotation about the origin

(*i*) counter clockwise by π.

(*ii*) clock wise by $\pi/2$

Solution: (*i*) We know for counter clockwise the rotation matrix is

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

Here $\theta = \pi$

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

(ii) For clockwise rotation, the rotation matrix is

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

i.e.,
$$R_{\pi/2} = \begin{bmatrix} \cos\frac{\pi}{2} & -\sin\frac{\pi}{2} \\ \sin\frac{\pi}{2} & \cos\frac{\pi}{2} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

8.3.3. Scaling

Scaling is a transformation that changes the size or shape of an object. Scaling with respect to origin can be carried out by multiplying the co-ordinate values (x, y) of each vertex of a polygon, or each endpoint of a line by scaling factors S_x and S_y respectively to produce the coordinates (x', y').

The mathematical expression for pure scaling is

$$x' = S_x * x$$
$$y' = S_y * y$$

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

or symbolically $[X'] = [T]\ [X]$

After applying scaling factor or matrix on any object, coordinate is either increased or decreased depending on the value of scaling factors S_x and S_y. If it is greater than one, then enlargement of the object will occur and if it is less than one compression will occur.

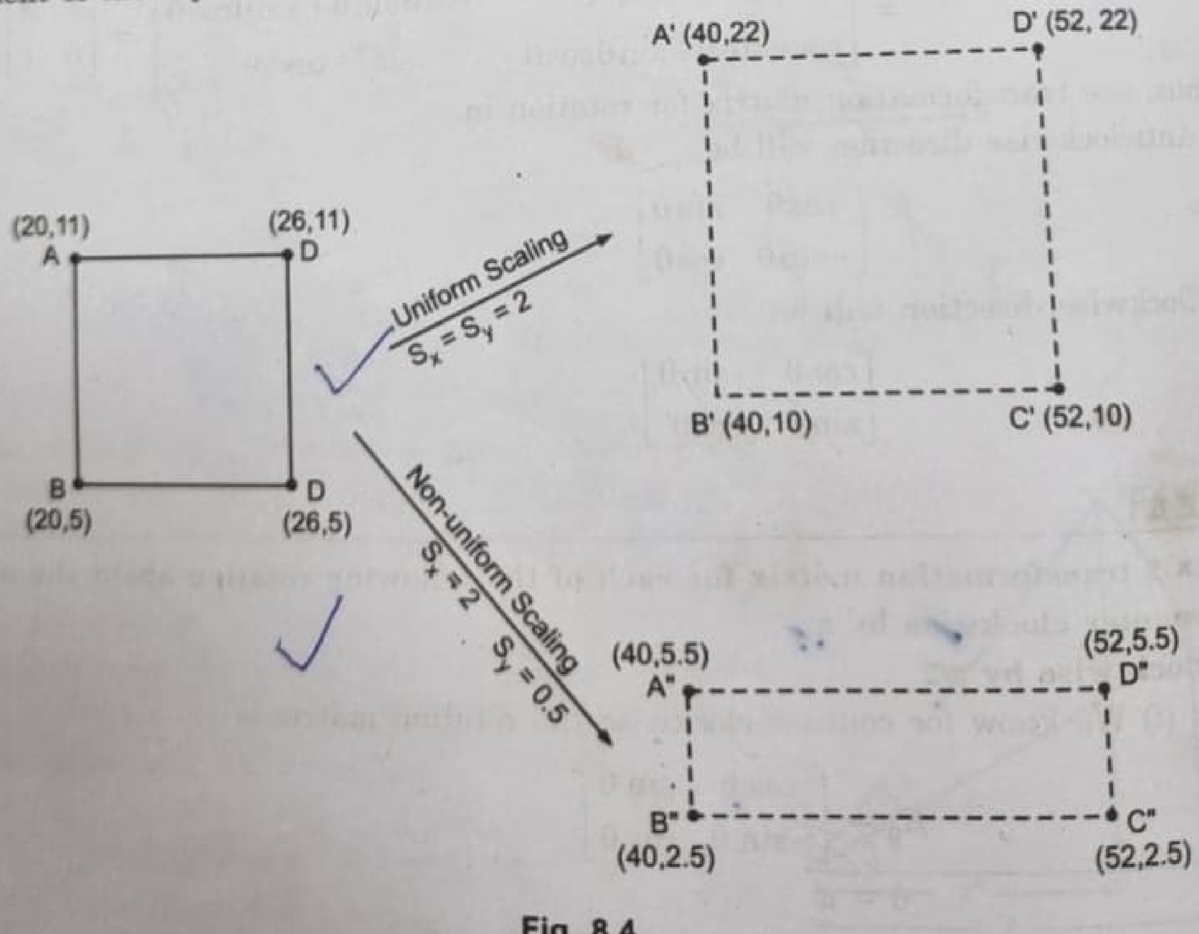


Fig. 8.4.

In general, for *uniform scaling* if $S_x = S_y > 1$, then a uniform compression occurs *i.e.*, the object becomes larger. If $S_x = S_y < 1$, then a uniform compression occurs *i.e.*, the object gets smaller. Non-uniform expansions or compressions occur, depending on whether S_x and S_y are individually > 1 or < 1 but unequal, such scaling is also known as *differential scaling*. While the basic object shape remains unaltered in uniform scaling, the shape and size both changes in differential scaling.

Thus, in pure uniform scaling with factors < 1 moves objects closer to the origin while factors > 1 moves object farther from origin, at the same time decreasing or increasing the object size.

EXAMPLE 8.2.

Write 2 × 2 transformation matrix for each of the following scaling transformation:

(*i*) **The entire picture is 3 times as large.**

(*ii*) **The entire picture is 1/3 as large.**

(*iii*) **The x direction is 4 times as large and the y direction unchanged.**

(*iv*) **The y length is reduced to 2/3 their original value and the x length unchanged.**

(*v*) **The x direction reduced to 3/4 the original and y direction increased by 7/5 times.**

Solution: (*i*) We know the scaling transformation matrix is

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

Put $S_x = 3$ and $S_y = 3$

$S_{3,3} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$ which will transform the picture three times of its original size.

(*ii*) Here $S_x = \dfrac{1}{3}$ and $S_y = \dfrac{1}{3}$

So
$$S_{\frac{1}{3},\frac{1}{3}} = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{3} \end{bmatrix}$$

(*iii*) Here, $S_x = 4$ and $S_y = 1$ (*i.e.*, unchanged)

So,
$$S_{4,1} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$

(*iv*) Here $S_x = 1$ and $S_y = \dfrac{2}{3}$

Therefore,
$$S_{1,\frac{2}{3}} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{2}{3} \end{bmatrix}$$

(*v*) Here $S_x = \dfrac{3}{4}$ and $S_y = \dfrac{7}{5}$

Therefore,
$$S_{\frac{3}{4},\frac{7}{5}} = \begin{bmatrix} \frac{3}{4} & 0 \\ 0 & \frac{7}{5} \end{bmatrix}$$

8.3.4. Shearing

The transformation "shearing", when applied to any object results only in distortion of shape. In shearing, the opposite and parallel layers of any object are simply slided with respect to each other.

Shearing can be done either in x-direction or y-direction.

(a) X-shear: In x-shear y co-ordinate remains unchanged, but x is changed as shown in Fig. 8.5.

Transformation matrix for x-shear is

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

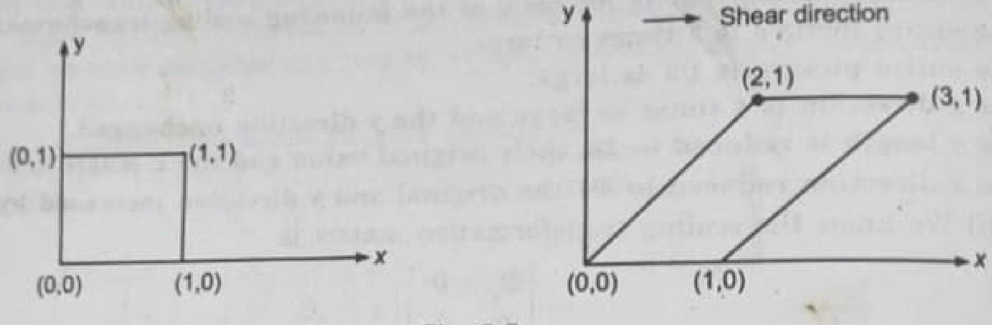


Fig. 8.5.

$$[x', y] = [x, \ y]^* \begin{bmatrix} 1 & 0 \\ b & 1 \end{bmatrix} = [x + yb, \ y]$$

y-shear: In y-shear, y co-ordinate is changed and x-remains unchanged.

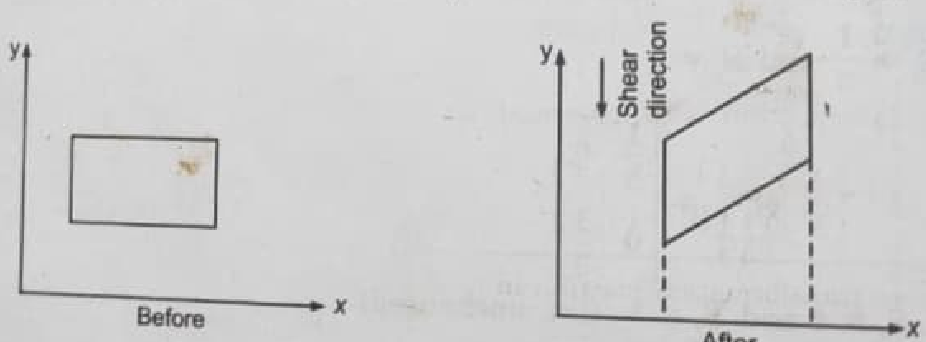


Fig. 8.6.

Transformation matrix for y-shear is

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

Thus

$$[x, \ y'] = [x, y] * \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} = [x, \ xa + y]$$

.3.5. Reflection

A reflection is a transformation that produces a mirror image of an object. In 2D reflection e consider any line in 2D plane as the mirror. The reflection is also described as the rotation 180°.

(a) **Reflection about x-axis:**

The basic principles behind reflection transformation are: *(UPTU 2009)*

(i) The image of an object is formed on the side opposite to where the object is lying w.r.t. mirror line.

(ii) The perpendicular distance of the object from the mirror line is same to the distance of the reflected image.

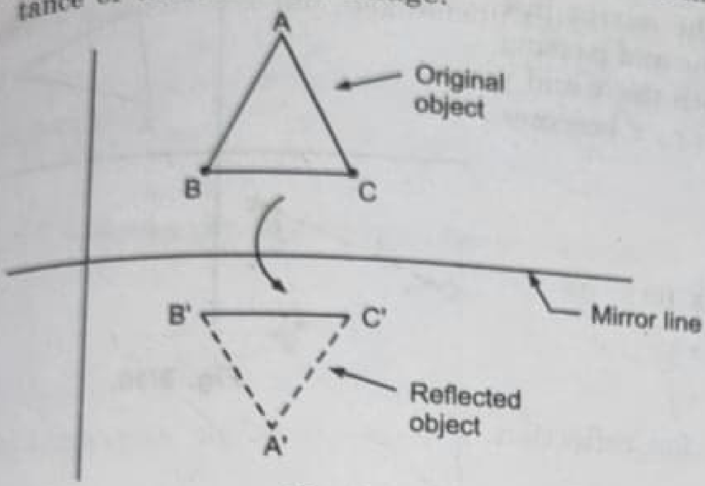


Fig. 8.7.

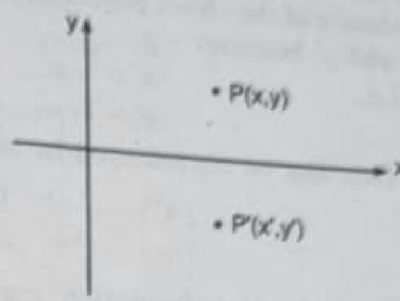


Fig. 8.8.

For reflection about x-axis, x co-ordinate is not changed and sign of y-coordinate is changed. Thus if we reflect point (x, y) in the x-axis, we get $(x, -y)$

i.e.,
$$x' = x$$
$$y' = -y$$

So, the transformation matrix for reflection about x-axis or $y = 0$ axis is,

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

and the transformation is represented as

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

Note: Suppose transformation matrix can be found as

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} * \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ -y \end{pmatrix}$$

as
$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

(b) **Reflection about Y-axis:**

A reflection about Y-axis $(x = 0)$ flips x-coordinates while y-coordinates remains the same. For reflection of $P(x, y)$ to $P'x', y)$

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

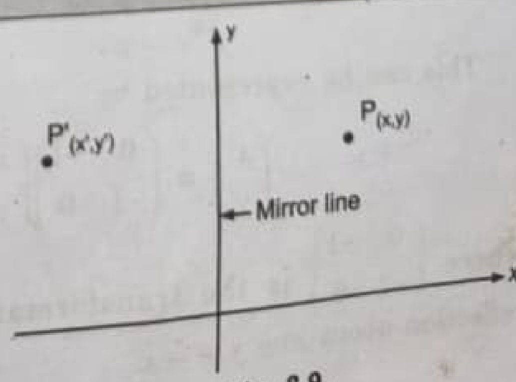


Fig. 8.9.

This transformation is identified by the reflection transformation matrix as

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

(c) Reflection about the Origin:

In this case, we actually choose the mirror line as the axis perpendicular to the xy plane and passing through the origin. After reflection both the x and y coordinate of the object point is flipped *i.e.*, x' becomes $-x$ and y' becomes $-y$.

i.e.,
$$x' = -x$$
$$y' = -y$$

This can be represented in matrix form as

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

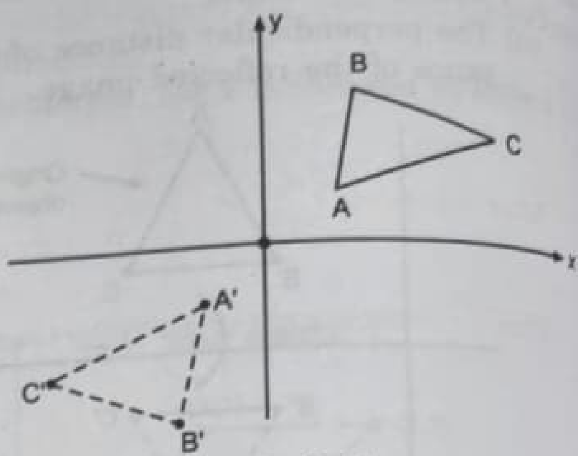


Fig. 8.10.

Thus the transformation matrix for reflection about origin is

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

(d) Reflection about the straight line $y = x$

If we reflect (x, y) point about the line $y = x$, we get (y, x) i.e.,

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

This can be represented by,

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

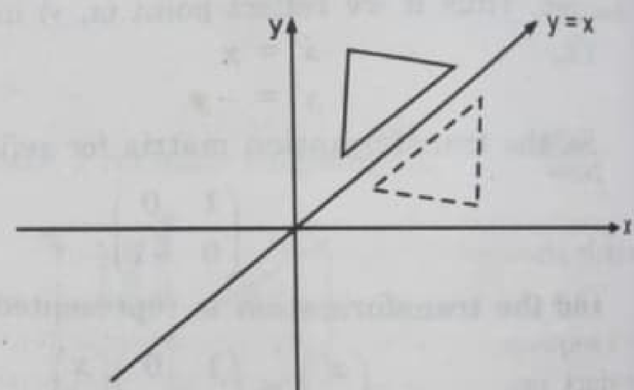


Fig. 8.11.

where $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ is the transformation matrix for reflection about line $y = -x$.

(e) Reflection about the straight line $y = -x$

If we reflect point $P(x, y)$ in the line $y = -x$ it becomes $(-y, -x)$

i.e.,
$$x' = -y$$
$$y' = -x$$

This can be represented by

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

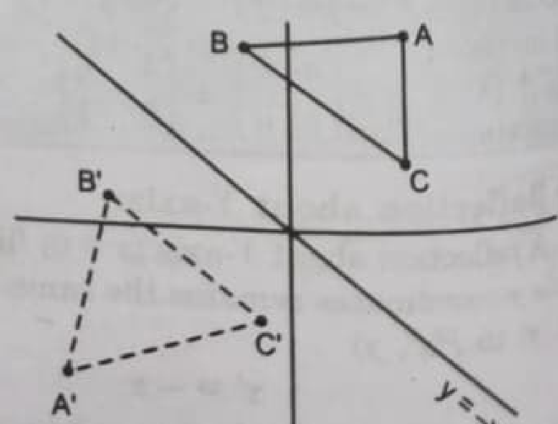


Fig. 8.12.

where $\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$ is the transformation matrix for reflection about line $y = -x$.

8.4. HOMOGENEOUS COORDINATES

(*UPTU 2004*)

We have seen how the shape, size, position and orientation of 2D objects. In some cases, however a desired orientation of an object may require more than one transformation to be applied successively. For example let us consider a case which requires 90° rotation of a point $\begin{pmatrix} x \\ y \end{pmatrix}$ about origin followed by reflection through the line $y = x$.

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

This $\begin{pmatrix} x' \\ y' \end{pmatrix}$ then undergoes reflection to produce $\begin{pmatrix} x'' \\ y'' \end{pmatrix}$ as

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

These successive matrix operations can be symbolically expressed as

$$[X''] = [T_M]_{y=x}\,[X']$$
$$[X''] = [T_M]_{y=x}\,\{[T_R]_{90°}[X]\}$$

i.e., $[T_M]_{y=x}[T_R]_{90°} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

Now $[X''] = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ -y \end{pmatrix}$

which shows the same result as before.

This process of calculating the product of matrix of a number of different transformations in a sequence is known as concatenation or combination of transformations and the resultant product matrix is referred to as *composite or concatenated transformation matrix*.

Application of concatenated transformation matrix on the object coordinates eliminates the calculation of intermediate coordinate values after each successive transformation.

But the real problem arises when there is a translation or rotation or scaling about an arbitrary point other than the origin involved among several successive transformations. The reason being the general form of expression of such transformations is not simply $[X'] = [T][X]$ involving the 2 × 2 array, $[T]$ containing multiplicative factors, rather it is in form $[X'] = [T_1]\,[X] + [T_2]$ where $[T_2]$ is the additional two element column matrix containing the translational terms. Such transformations cannot be combined to form a single resultant representative matrix. This problem can be eliminated of we can combine $[T_1]$ and $[T_2]$ into a single transformation matrix. This can be done by expanding the 2 × 2 transformation matrix format into 3 × 3 form. The general 3 × 3 form will be something like.

$\begin{bmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} & \begin{matrix} m \\ n \end{matrix} \\ \begin{matrix} 0 & 0 \end{matrix} & 1 \end{bmatrix}$ where the elements a, b, c, d of the upper left 2 × 2 submatrix are the multiplicative factors of $[T_1]$ and m, n are the respective x, y translational factors of $[T_2]$. But such 3 × 3 matrices are not comfortable for multiplication with 2 × 1 2D position vector matrices. Herein lies the need to include a dummy coordinate to make 2×1 position vector matrix $\begin{pmatrix} x \\ y \end{pmatrix}$ to a 3×1 matrix $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ where the third coordinate is dummy.

Now, if we multiply, $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ with a non-zero scalar 'h' then the matrix it forms is

$\begin{pmatrix} xh \\ yh \\ h \end{pmatrix}$ or symbolically say $\begin{pmatrix} x_h \\ y_h \\ h \end{pmatrix}$ which is known as the *homogeneous coordinates* or homogeneous position vector of the same point $\begin{pmatrix} x \\ y \end{pmatrix}$ is 2D

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} x_h \\ y_h \\ h \end{pmatrix} = h.\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

The extra coordinate h is known as a *weight*, which is homogeneously applied to the cartesian components. Thus a general homogeneous coordinate representation of any point $P(x, y)$ is (x_h, y_h, h) that implies $x = \frac{x_h}{h}$ $y = \frac{y_h}{h}$. As 'h' can have any non-zero value, there can be infinite number of equivalent homogeneous representation of any given point in space e.g., (4, 6, 2), (8, 12, 4), (2, 3, 1), (–2, –3, –1) all represent the physical point (2, 3).

But so far as geometric transformation is concerned our choice is simply $h = 1$ and the corresponding homogenous coordinate triple $(x, y, 1)$ for representation of point position (x, y) in xy-plane. Other values of parameter 'h' are needed frequently in matrix formulation of three dimensional viewing transformation.

In 3×3 matrix form for homogeneous co-ordinates, the translation equations are

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

Equations of rotation and scaling with respect to coordinate origin may be modified as

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

and $$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$ respectively.

Similarly the modified general expression for reflection may be

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

where values of a, b, c, d depend upon choice of co-ordinate axes or diagonal axis as mirror line.

8.5. COMBINED TRANSFORMATIONS

(UPTU 2009)

To control the size of an object, we need to perform matrix operations on the position vector which defines the vertices. It is not necessary that we get the required orientation by applying single transformation, it may require more than one transformation. Since matrix multiplication is non commutative, the order of application of the transformation is important.

We can set up a matrix for any sequence of transformations as a composite transformation matrix by calculating the matrix product of the individual transformations.

(i) If two successive translation are applied then

$$\begin{pmatrix} 1 & 0 & t_{x1} \\ 0 & 1 & t_{y1} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & t_{x2} \\ 0 & 1 & t_{y2} \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_{x1}+t_{x2} \\ 0 & 1 & t_{y1}+t_{y2} \\ 0 & 0 & 1 \end{pmatrix}$$

(ii) If two successive rotations θ_1 and θ_2 then

$$\begin{pmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

(iii) If two successive scaling are applied then

$$\begin{pmatrix} S_{x1}\cdot S_{x2} & 0 & 0 \\ 0 & S_{y1}\cdot S_{y2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

8.5.1. Inverse Transformation

For each geometric transformation there exists an inverse transformation which describes just the opposite operation of that performed by the original transformation. Any transformation followed by its inverse transformation keeps an object unchanged in position, orientation, size and shape. We will use inverse transformation to nullify the effects of already applied transformation.

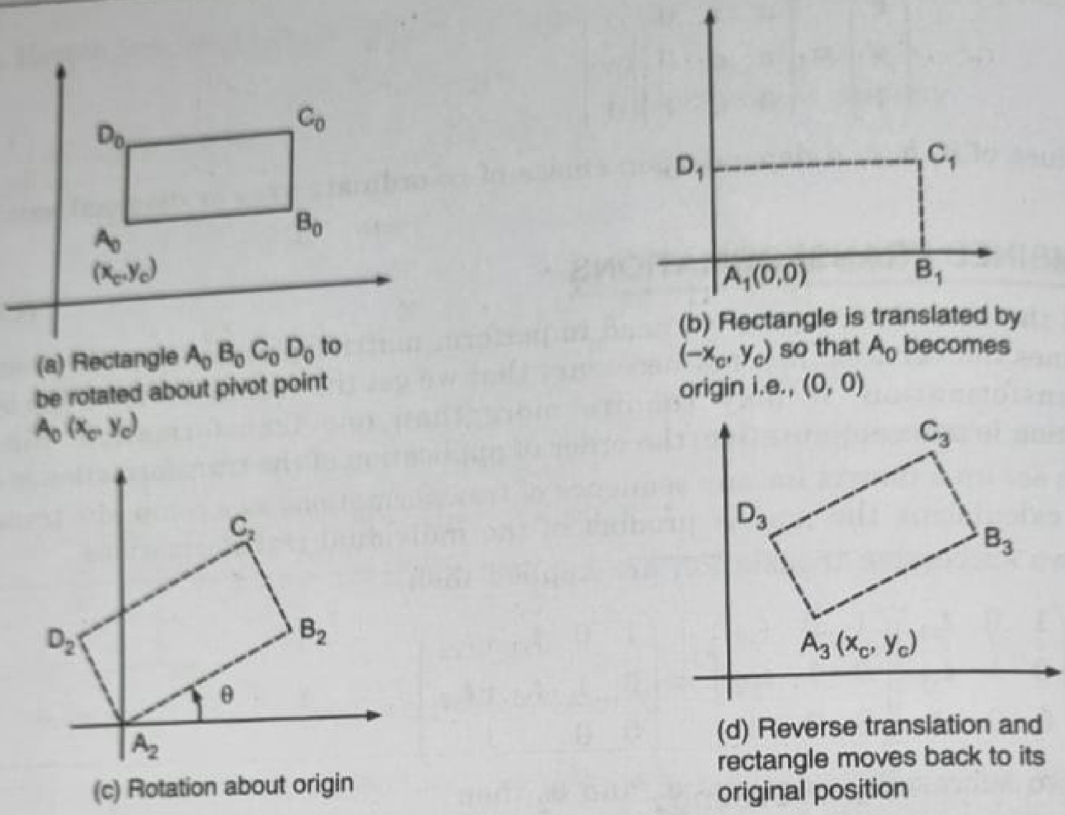
8.5.2. Rotation About an Arbitrary Point

We discussed earlier about rotation, which occurs only about the origin but now our aim is to rotate the object about a point other than origin. For this purpose homogeneous coordinate system will provide a method to accomplish rotation about any point. This can be done in the following order:

Step 1: Translate the object or body at the origin (translation)

Step 2: Rotate by any angle as given (Rotation)

Step 3: Translate back to its original location (inverse translation).



(a) Rectangle $A_0 B_0 C_0 D_0$ to be rotated about pivot point $A_0 (x_c, y_c)$

(b) Rectangle is translated by $(-x_c, y_c)$ so that A_0 becomes origin i.e., (0, 0)

(c) Rotation about origin

(d) Reverse translation and rectangle moves back to its original position

Fig. 8.13.

In matrix form, it can be shown as

$$\boxed{[T] = [T_R]\,[R_\theta]\,[T_R]^{-1}}$$

where $T_R \rightarrow$ Translation matrix
$R_\theta \rightarrow$ Rotation matrix by an angle $\theta°$
$[T_R]^{-1} \rightarrow$ Inverse translation matrix.

Let us obtain transformation matrix for anticlockwise rotation about point (x_c, y_c)

(*i*) **Translation:** Moves (x_c, y_c) to origin $t_x = -x_c\ t_y = -y_c$

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

(*ii*) **Rotation:** $R_\theta = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$

(*iii*) **Translation:** Moves the image back to its original position
$t_x = x_c$ and $t_y = y_c$

$$[T_r]^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_c & y_c & 1 \end{pmatrix}$$

Now combined transformation matri

$$[x' \;\; y' \;\; 1] = [x \;\; y \;\; 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_c & -y_c & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_c & y_c & 1 \end{bmatrix}}$$

$$= [x \;\; y \;\; 1] \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_c & -y_c & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ x_c & y_c & 1 \end{bmatrix}}$$

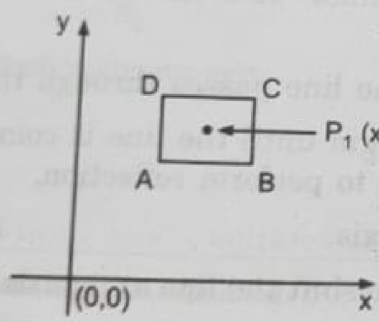
$$= [x \;\; y \;\; 1] \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ -x_c\cos\theta + y_c\sin\theta + x_c & -x_c\sin\theta - y_c\cos\theta + y_c & 1 \end{bmatrix}$$

So,

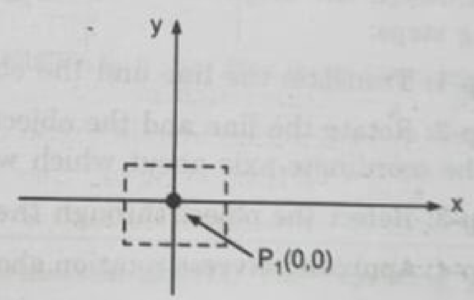
$$\boxed{[x' \;\; y' \;\; 1] = [x \;\; y \;\; 1] \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ -x_c(\cos\theta - 1) + y_c\sin\theta & -y_c(\cos\theta - 1) - x_c\sin\theta & 1 \end{bmatrix}}$$

8.5.3. General Fixed-Point Scaling

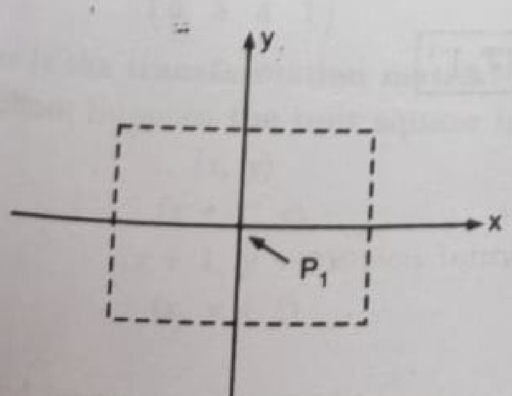
The scaling matrix discussed previously performs scaling about the origin. But sometimes it may be required to perform scaling without altering the objects position.



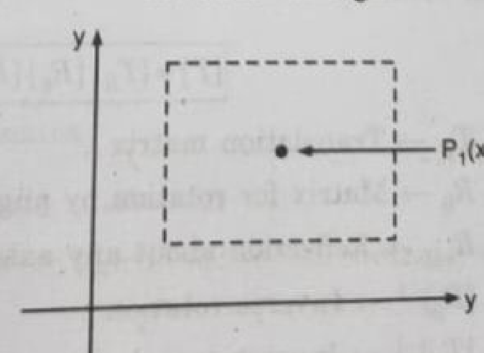
(a) Original position of the object which want to be scaled about its centroid P_1



(b) Translation of object so that P_1 coincides with origin.



(c) Scaling of object w.r.t. origin



(d) Reverse translation of scaled object such that P_1 returns to its initial position

Fig. 8.14.

Thus, following three steps will have to be followed

1. Translate point to origin
2. Perform scaling
3. Inverse translation

Hence,
$$[T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_c & -y_c & 1 \end{bmatrix} \underbrace{\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_c & y_c & 1 \end{bmatrix}}$$

$$= \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_c & -y_c & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ x_c & y_c & 1 \end{bmatrix}}$$

$$= \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ x_c(1-S_x) & y_c(1-S_y) & 1 \end{bmatrix}$$

8.5.4. Reflection Through an Arbitrary Line y = mx + c *(UPTU, 2012-13)*

We have already discussed reflection through the line $x = 0$, $y = 0$, $y = x$, $y = -x$. All these lines pass through origin. But when reflection is to be performed about a line that neither passes through the origin nor is parallel to the co-ordinate axis can be solved using the following steps:

Step-1: Translate the line and the object so that the line passes through the origin.

Step-2: Rotate the line and the object about the origin until the line is coincident with one of the coordinate axis about which we are familiar to perform reflection.

Step-3: Refect the object through the co-ordinate axis.

Step-4: Apply the inverse rotation about the origin to shift the line at translated position.

Step-5: Apply inverse translation to send back the object (i.e., line) to its original position.

In matrix rotation

$$\boxed{[T] = [T_R][R_\theta][R_{ref}][R_\theta]^{-1}[T_R]^{-1}}$$

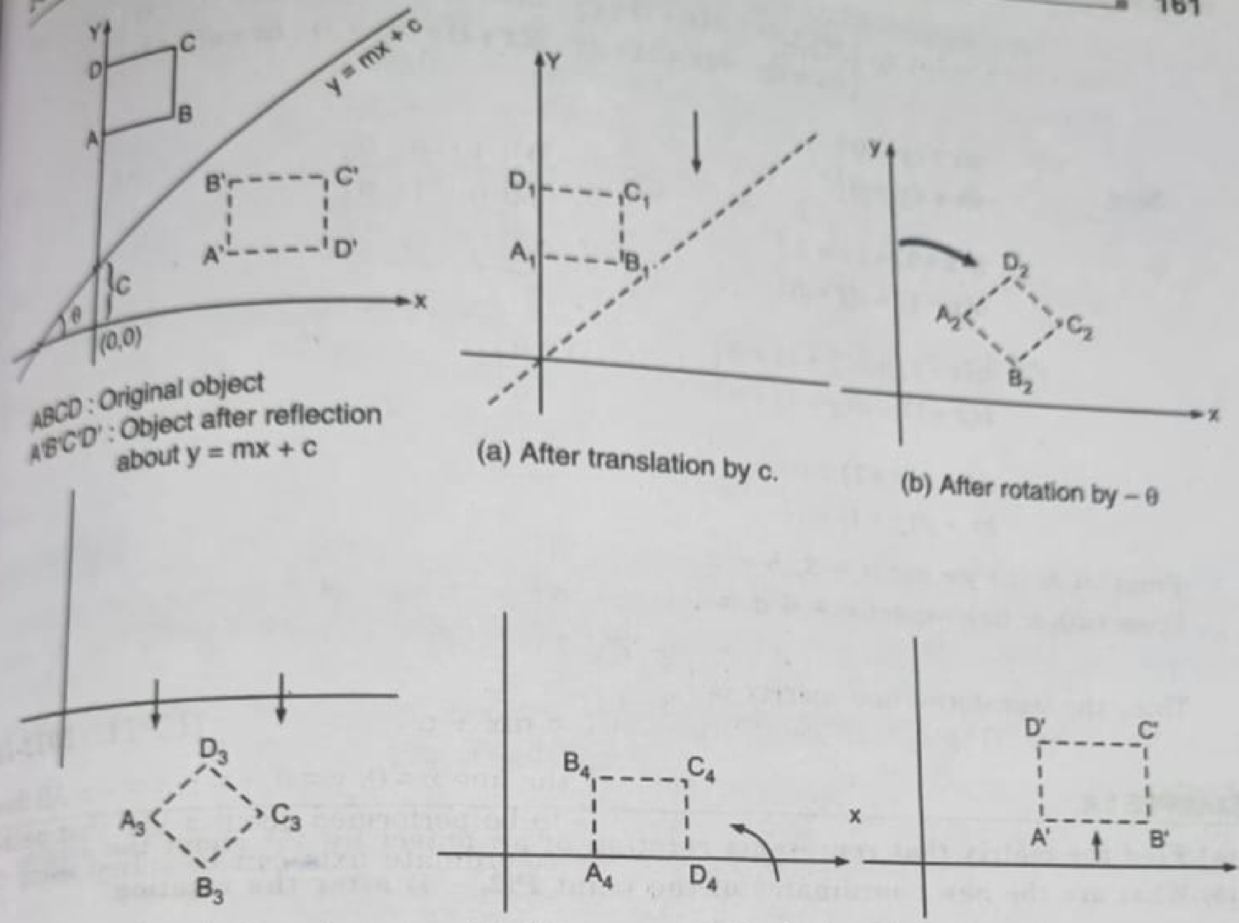
where $T_R \rightarrow$ Translation matrix

$R_\theta \rightarrow$ Matrix for rotation by angle θ

$R_{ref} \rightarrow$ Reflection about any axis

$[R_\theta]^{-1} \rightarrow$ Inverse rotation

$[T_R]^{-1} \rightarrow$ Inverse translation



(a) After translation by c.

(b) After rotation by – θ

(c) After reflection about x-axis

(d) After reverse rotation by θ

(e) After reverse translation by c

Fig. 8.15.

EXAMPLE 8.3.

A unit square is transformed by a 2 × 2 transformation matrix. The resulting position vectors are

$$\begin{pmatrix} 0 & 2 & 8 & 6 \\ 0 & 3 & 4 & 1 \end{pmatrix}$$

what is the transformation matrix?

Solution: Suppose the unit square have coordinates

$$(x, y)$$
$$(x + 1, y)$$
$$(x + 1, y + 1)$$
$$(x, y + 1)$$

and let the transformation matrix be $\begin{pmatrix} a & c \\ b & d \end{pmatrix}$

So, $$\begin{pmatrix} 0 & 2 & 8 & 6 \\ 0 & 3 & 4 & 1 \end{pmatrix} = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \begin{pmatrix} x & x+1 & x+1 & x \\ y & y & y+1 & y+1 \end{pmatrix}$$

$$= \begin{pmatrix} ax+cy & a(x+1)+cy & a(x+1)+c(y+1) & ax+c(y+1) \\ bx+dy & b(x+1)+dy & b(x+1)+d(y+1) & bx+d(y+1) \end{pmatrix}$$

Now $\left.\begin{aligned} ax+cy=0 \\ bx+dy=0 \end{aligned}\right\}$ ...(i)

$$\left.\begin{aligned} a(x+1)+cy=2 \\ b(x+1)+dy=3 \end{aligned}\right\} \quad ...(ii)$$

$$\left.\begin{aligned} a(x+1)+c(y+1)=8 \\ b(x+1)+d(y+1)=4 \end{aligned}\right\} \quad ...(iii)$$

$$\left.\begin{aligned} ax+c(y+1)=6 \\ bx+d(y+1)=1 \end{aligned}\right\} \quad ...(iv)$$

From (i) & (ii) we get $a = 2$, $b = 3$.
From (iii) & (iv) we get $c = 6$ $d = 1$

Thus, the transformation matrix is $\begin{pmatrix} 2 & 6 \\ 3 & 1 \end{pmatrix}$.

EXAMPLE 8.4.

(a) Find the matrix that represents rotation of an object by 45° about the origin.
(b) What are the new coordinates of the point $P(2, -4)$ after the rotation?

Solution: (a) $R_{45°} = \begin{pmatrix} \cos 45° & -\sin 45° \\ \sin 45° & \cos 45° \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$

(b) The new co-ordinates can be found by

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 2 \\ -4 \end{pmatrix} = \begin{pmatrix} \sqrt{2}+2\sqrt{2} \\ \sqrt{2}-2\sqrt{2} \end{pmatrix}$$

i.e., new coordinates are $\left(\sqrt{2}+2\sqrt{2},\ \sqrt{2}-2\sqrt{2}\right)$

EXAMPLE 8.5.

A triangle is defined by $\begin{pmatrix} 2 & 4 & 4 \\ 2 & 2 & 4 \end{pmatrix}$

Find the transformed coordinates after the following transformation
(i) 90° rotation about origin
(ii) reflection about line $y = -x$.

Solution: (*i*) After 90° rotation about origin, the transformed coordinate are

$$\begin{pmatrix} \cos 90^\circ & -\sin 90^\circ & 0 \\ \sin 90^\circ & \cos 90^\circ & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 2 & 4 & 4 \\ 2 & 2 & 4 \\ 1 & 1 & 1 \end{pmatrix}$$

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

Finally after reflection about line $y = -x$, the transformation matrix is

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} -2 & -2 & -4 \\ 2 & 4 & 4 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} -2 & -4 & -4 \\ 2 & 2 & -4 \\ 1 & 1 & 1 \end{bmatrix}$$

EXAMPLE 8.6.

Translate the square *ABCD* whose co-ordinate are *A*(0, 0), *B*(3, 0), *C*(3, 3), and *D*(0, 3) by 2 units in both directions and then scale it by 1.5 units in *x*-direction and 0.5 units in *y*-direction.

Solution: Here first operation is translation and second operation is scaling.

$$T_x = 2 \quad \text{and} \quad T_y = 2$$

$$\text{Translation matrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$

$$\text{Square } ABCD \text{ in matrix form} = \begin{bmatrix} 0 & 0 & 1 \\ 3 & 0 & 1 \\ 3 & 3 & 1 \\ 0 & 3 & 1 \end{bmatrix}$$

Performing translation operation, we get

$$\begin{bmatrix} 0 & 0 & 1 \\ 3 & 0 & 1 \\ 3 & 3 & 1 \\ 0 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 1 \\ 5 & 2 & 1 \\ 5 & 5 & 1 \\ 2 & 5 & 1 \end{bmatrix}$$

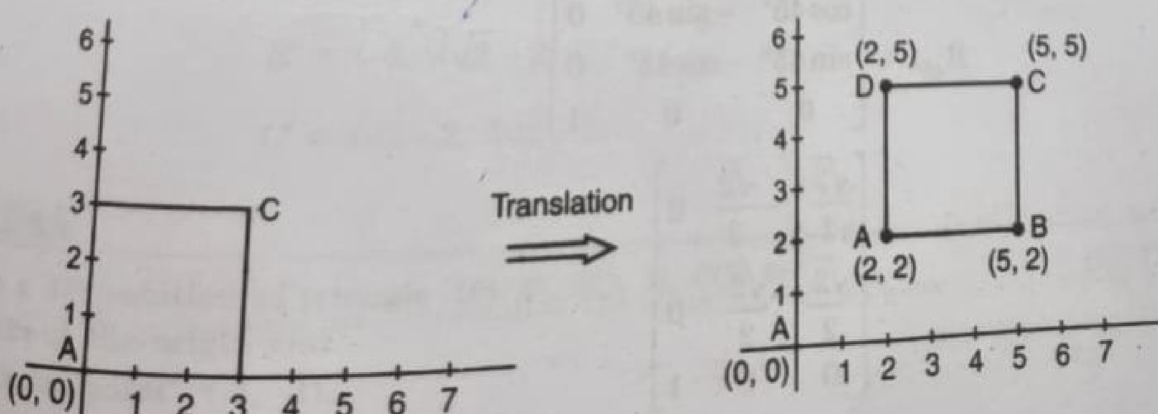


Fig. 8.16.

Now the 2nd operation is scaling so $S_x = 1.5$ $S_y = 0.5$

$$\text{and scaling matrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

After scaling we will get

$$\begin{bmatrix} 2 & 2 & 1 \\ 5 & 2 & 1 \\ 5 & 5 & 1 \\ 2 & 5 & 1 \end{bmatrix} \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 1 \\ 7.5 & 1 & 1 \\ 7.5 & 2.5 & 1 \\ 3 & 2.5 & 1 \end{bmatrix}$$

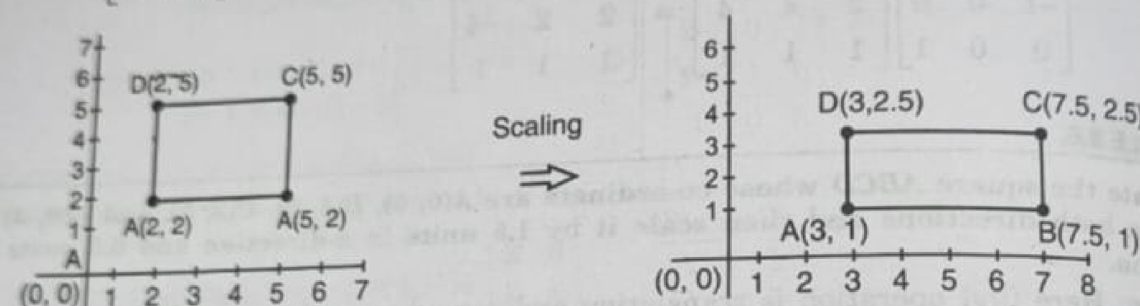


Fig. 8.17.

After transformation, new co-ordinates are

$A(3,1)$
$B(7.5,\ 1)$
$C(7.5,\ 2.5)$
$D(3,\ 2.5)$

EXAMPLE 8.7.

Rotate a triangle $A(0,\ 0)$, $B(2,\ 2)$, $C(4,\ 2)$ about the origin and about $P(-2,\ -2)$ by an angle of $45°$.

Solution: The given triangle ABC can be represented by a matrix formed from the homogenous coordinates of the vertices as

$$\begin{bmatrix} 0 & 2 & 4 \\ 0 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$R_{45°} = \begin{bmatrix} \cos 45° & -\sin 45° & 0 \\ \sin 45° & \cos 45° & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The coordinates of the rotated triangle ABC are,

$$\begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 2 & 4 \\ 0 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \sqrt{2} \\ 0 & 2\sqrt{2} & 3\sqrt{2} \\ 1 & 1 & 1 \end{bmatrix}$$

Now, rotate about $P(-2, -2)$

$$R_{45^\circ} \cdot P = [T_R] \times [R_{45^\circ}] \cdot [T_R]^{-1}$$

$$= \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} & -2 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 2\sqrt{2}-2 \\ 0 & 0 & 1 \end{bmatrix}$$

Now, $$[A'\ B'\ C'] = R_{45^\circ} \times P \times [\text{A B C}]$$

$$[A'\ B'\ C'] = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} & -2 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 2\sqrt{2}-2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 4 \\ 0 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -2 & -2 & \sqrt{2}-2 \\ 2\sqrt{2}-2 & 4\sqrt{2}-2 & 5\sqrt{2-2} \\ 1 & 1 & 1 \end{bmatrix}$$

Thus, $$A' = (-2,\ 2\sqrt{2}-2)$$

$$B' = (-2,\ 4\sqrt{2}-2)$$

$$C' = (\sqrt{2}-2,\ 5\sqrt{2}-2)$$

EXAMPLE 8.8. (UPTU 2003)

Perform a 45° rotation of triangle $A(0, 0)$, $B(1, 1)$, $C(5, 2)$

(a) about the origin and

(b) about point $P(-1, -1)$.

$$= \begin{bmatrix} -1 & -1 & \left(\frac{3}{2}\sqrt{2}-1\right) \\ (\sqrt{2}-1) & (2\sqrt{2}-1) & \left(\frac{9}{2}\sqrt{2}-1\right) \\ 1 & 1 & 1 \end{bmatrix}$$

Thus

$$A' = \left(1-, \sqrt{2}-1\right)$$

$$B' = \left(1-, 2\sqrt{2}-1\right)$$

$$C' = \left(\frac{3}{2}\sqrt{2}-1, \frac{9}{2}\sqrt{2}-1\right)$$

EXAMPLE 8.9.

Find the transformation matrix that transforms the square *ABCD* whose center is at (2, 2) is reduced to half of its size, with center still remaining at (2, 2). The coordinates of square *ABCD* are *A*(0, 0) *B*(0, 4) *C*(4, 4) and *D*(4, 0). Find the co-ordinates of new square.

Solution: Square ABCD in matrix form as

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 4 & 1 \\ 4 & 4 & 1 \\ 4 & 0 & 1 \end{pmatrix}$$

For reducing square *ABCD* to half of its size we scale it by $S_x = \frac{1}{2}$ and $S_y = \frac{1}{2}$. So, scaling matrix is

$$\begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

After scaling we get

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 4 & 1 \\ 4 & 4 & 1 \\ 4 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 2 & 1 \\ 2 & 2 & 1 \\ 2 & 0 & 1 \end{pmatrix}$$

After scaling the co-ordinates of square *ABCD* are

$A = (0, 0)$

$B = (0, 2)$

$C = (2, 2)$

$D = (2, 0)$

and now the centre is (1, 1) but we want the centre of *ABCD* (2, 2). Therefore, we translate the square *ABCD* by $T_x = 1$ and $T_y = 1$.

Translation matrix

Solution: Triangle ABC in matrix form is given as

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 5 & 2 & 1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

(a) Matrix of rotation is

$$R_{45^\circ} = \begin{pmatrix} \cos 45^\circ & -\sin 45^\circ & 0 \\ \sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Now find the co-ordinates $A'\ B'\ C'$ of the rotated triangle ABC can be found as

$$[A'\ \ B'\ \ C'] = R_{45^\circ} \cdot \begin{pmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & \frac{3\sqrt{2}}{2} \\ 0 & \sqrt{2} & \frac{7\sqrt{2}}{2} \\ 1 & 1 & 1 \end{pmatrix}$$

Thus

$$A' = (0, 0)$$

$$B' = \left(0, \sqrt{2}\right)$$

$$C' = \left(\frac{3\sqrt{2}}{2}, \frac{7\sqrt{2}}{2}\right)$$

(b) The rotation matrix is

$$\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & -1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \left(\sqrt{2}-1\right) \\ 0 & 0 & 1 \end{pmatrix}$$

Now, $$[A'\ \ B'\ \ C'] = \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & -1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \left(\sqrt{2}-1\right) \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

After translation we get

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

Thus, the new co-ordinates for square *ABCD* are

$A \rightarrow (1, 1)$
$B \rightarrow (1, 3)$
$C \rightarrow (3, 3)$
$D \rightarrow (3, 1)$

EXAMPLE 8.10.

Consider the square A(1, 0) $B(0, 0)$ $C(0, 1)$ $D(1, 1)$. Rotate the square ABCD by 45° clockwise about A(1, 0).

Solution: Square *ABCD* in matrix form as follows.

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

Matrix for rotation in clockwise about origin.

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

We have to rotate square *ABCD* about point *A*(1, 0). We first translate square *ABCD* by $T_x = -1$ and $T_y = 0$ *i.e.*,

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

Now we rotate in clockwise $\theta = 45°$

$$\begin{pmatrix} \cos 45° & -\sin 45° & 0 \\ \sin 45° & \cos 45° & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 1 \\ 0 & \sqrt{2} & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 1 \end{pmatrix}$$

Now, we translate square ABCD back to its position $T_x = 1$ & $T_y = 0$ i.e.,

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

EXAMPLE 8.11.

Magnify the triangle with vertices A(0, 0) B(1, 1) and C(5, 2) to twice its size while keeping C(5, 2) fixed.

Solution: First we write the required transformation matrix.

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

Triangle ABC in matrix form as follows

$$\begin{pmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix}$$

Now, the new transformation matrix

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

i.e., the new co-ordinates after transformation are

$A \to (-5, -2)$
$B \to (-3, 0)$
$C \to (5, 2)$

EXAMPLE 8.12.

Prove that two-dimensional rotation and scaling commutative if $S_x = S_y$ or $\theta = n\pi$.

Solution: The transformation matrix for rotation about origin in anticlockwise direction.

$$\begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and scaling matrix is } \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

To prove commutative property holds for $S_x = S_y$

$$R.S = S.R$$

i.e.,

$$R.S = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} S_x\cos\theta & S_x\sin\theta & 0 \\ -S_x\sin\theta & S_x\cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$S.R. = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} S_x\cos\theta & S_x\sin\theta & 0 \\ -S_x\sin\theta & S_x\cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thus R.S = S.R

Now for $\theta = n\pi$

$$R.S = \begin{pmatrix} \cos n\pi & \sin n\pi & 0 \\ -\sin n\pi & \cos n\pi & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\cdot\begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$S.R = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \cos n\pi & \sin n\pi & 0 \\ -\sin n\pi & \cos n\pi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thus RS = SR if $S_x = S_y$ or $\theta = n\pi$

EXAMPLE 8.13.

The reflection along the line $y = x$ is equivalent to the reflection along the x-axis followed by counter clockwise rotation by θ degrees. Find the value of θ.

Solution: The transformation matrix for reflection about the line $y = x$ is

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad ...(i)$$

Reflection about x-axis, the matrix is $\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ and for counter clockwise rotation by θ about origin is

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

For successive application, the resultant transformation is

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

It is given that

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

i.e., $\cos\theta = 0$ and $\sin\theta = 1$ and $-\cos\theta = 0 \Rightarrow \theta = 90°$

EXAMPLE 8.14.

Show that the 2 × 2 matrix

$$[T] = \begin{bmatrix} \frac{1-t^2}{1+t^2} & \frac{2t}{1+t^2} \\ \frac{-2t}{1+t^2} & \frac{1-t^2}{1+t^2} \end{bmatrix}$$ **represents pure rotation.**

Solution: We know that for pure rotational transformation determinant of the transformation matrix is always equal to 1.

So, the determinant of $[T] = \left\{\left(\frac{1-t^2}{1+t^2}\right)^2 - \left(-\frac{2t}{1+t^2}\right)^2\right\} = \frac{(1-t^2)^2}{(1+t^2)^2} + \frac{4t^2}{(1+t^2)^2} = \frac{(1-t^2)^2+4t^2}{(1+t^2)^2}$

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

EXAMPLE 8.15.

Show that a 2D reflection through X axis followed by a 2D reflection through the line $y = -x$ is equivalent to pure rotation about the origin.

Solution: The transformation matrix for reflection about x-axis is $\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ and reflection through $y = -x$ is $\begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

When applied successively, we get

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

The transformation matrix for rotation about origin by an angle $\theta = 270°$ is

$$\begin{pmatrix} \cos 270° & -\sin 270° & 0 \\ \sin 270° & \cos 270° & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Note: In general, if two pure reflection transformation about line passing through the origin are applied successively, the result is a pure rotation about origin.

EXAMPLE 8.16.

Show that transformation matrix for a reflection about $y = -x$ is equivalent to reflection relative to the y axis followed by a counter clockwise rotation by 90°.

Solution: Transformation matrix for reflection about line $y = -x$ is $\begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Transformation matrix for reflection relative to y axis is $\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

and the transformation matrix for counterclockwise rotation is

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

Here, $\theta = 90°$

So, $$\begin{pmatrix} \cos 90° & \sin 90° & 0 \\ -\sin 90° & \cos 90° & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

When applied successively, we get

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

which is equal to the transformation matrix for reflection about line $y = -x$.

EXAMPLE 8.17.

A mirror is placed vertically such that it passes through the points (10, 0) and (0, 10). Find the reflected view of triangle ABC with coordinates $A(5, 50)$, $B(20, 40)$ $C(10, 70)$.

Solution: We plot the mirror passing through (0, 10) and (10, 0)

From figure we easily get $\tan\theta = \dfrac{10}{10} = 1$, which implies that $\theta = 45°$

To make the line coincident with the x-axis we first translate it to make it pass through origin and then rotate it by $\theta = 45°$ about origin.

Co-ordinates of triangle ABC in matrix form is

$$\begin{pmatrix} 5 & 50 & 1 \\ 20 & 40 & 1 \\ 10 & 70 & 1 \end{pmatrix}$$

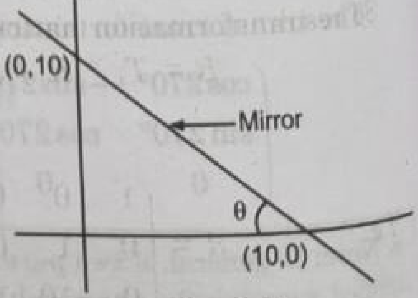


Fig. 8.18.

We translate mirror, so that it passes through origin.

$$t_x = 0 \qquad t_y = -10$$

$$T_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -10 & 1 \end{pmatrix}$$

Immediately we write inverse transformation matrix for translation by $t_x = 0$ and $t_y = 10$

$$T_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 10 & 1 \end{bmatrix}$$

Now we rotate the mirror by 45° anticlockwise so that it matches with origin

$$R_1 = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos 45° & \sin 45° & 0 \\ -\sin 45° & \cos 45° & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Immediately, we write inverse transformation matrix for rotation by 45°

$$R_1^{-1} = \begin{pmatrix} \cos(-45°) & \sin(-45°) & 0 \\ -\sin(-45°) & \cos(-45°) & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Transformation matrix for reflection about x-axis is

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

The steps are:

1. Translate the mirror and object so that it passes through origin *i.e.*, T_1
2. Rotate mirror and object by 45° in anticlockwise *i.e.*, R_1
3. Now mirror matches with x-axis then reflect triangle ABC about x-axis *i.e.*, R_{ref}
4. Rotate mirror and object by 45° clockwise *i.e.*, R_1^{-1}
5. Then translate mirror and object back to its position matrix *i.e.*, T_1^{-1}.

So resultant transformation matrix is

$$R = T_1 * R_1 * R_{ref} * R_1^{-1} * T_1^{-1}$$

i.e.,

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -10 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 10 & 1 \end{pmatrix}$$

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

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

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

Now, the new co-ordinates for Δ ABC

$$\begin{pmatrix} 5 & 50 & 1 \\ 20 & 40 & 1 \\ 10 & 70 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 10 & 10 & 1 \end{pmatrix} = \begin{pmatrix} -40 & 5 & 1 \\ -30 & -10 & 1 \\ -60 & 0 & 1 \end{pmatrix}$$

Thus after reflection, the new co-ordinates are

$A \rightarrow (-40, 5)$

$B \rightarrow (-30, -10)$

$C \rightarrow (-60, 0)$

EXAMPLE 8.18.

Prove that simultaneous shearing in both directions (x & y directions) is not equal to the composition of pure shear along x-axis followed by pure shear along y-axis.

Solution: We know the simultaneous shearing

$$S_h = \begin{pmatrix} 1 & a \\ b & 1 \end{pmatrix} \qquad \text{...(i)}$$

Shearing in x-direction is $\begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}$ and in y-direction is $\begin{pmatrix} 1 & 0 \\ b & 1 \end{pmatrix}$. Therefore, shearing in x-direction followed by y-direction is

$$\begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ b & 1 \end{pmatrix} = \begin{pmatrix} 1+ab & a \\ b & 1 \end{pmatrix}$$

which is not equal to Eq^n(i).

EXAMPLE 8.19.

Show that the composition of two rotations is additive by concatenating the matrix representations for $R(\theta_1)$ and $R(\theta_2)$ to obtain $R(\theta_1)\cdot R(\theta_2) = R(\theta_1 + \theta_2)$. *(UPTU 2003)*

Solution: We know the rotation matrix as

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

So $$R(\theta_1) = \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{pmatrix} \quad \text{and} \quad R(\theta_2) = \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 \\ \sin\theta_2 & \cos\theta_2 \end{pmatrix}$$

Then $$R(\theta_1)\cdot R(\theta_2) = \begin{pmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) \end{pmatrix} = R(\theta_1 + \theta_2)$$

$\therefore \quad R(\theta_1)\, R(\theta_2) = R(\theta_1 + \theta_2).$

EXAMPLE 8.20.

Reflect the triangle ABC about the line $3x - 4y + 8 = 0$. The position vector of the coordinate ABC is given as $A(4, 1)$, $B(5, 2)$, $C(4, 3)$.

Solution: Equation of line $3x - 4y + 8 = 0$

$$m = \frac{3}{4} = \tan\theta.$$

$$\tan\theta = \frac{3}{4} \text{ so, } \sin\theta = \frac{3}{5} \quad \cos\theta = \frac{4}{5}$$

The intersection of the line $3x - 4y + 8 = 0$ with

$x = 0$ is $y = 2 \Rightarrow (0, 2)$

$y = 0$ is $x = -8/3 \Rightarrow (-8/3, 0)$

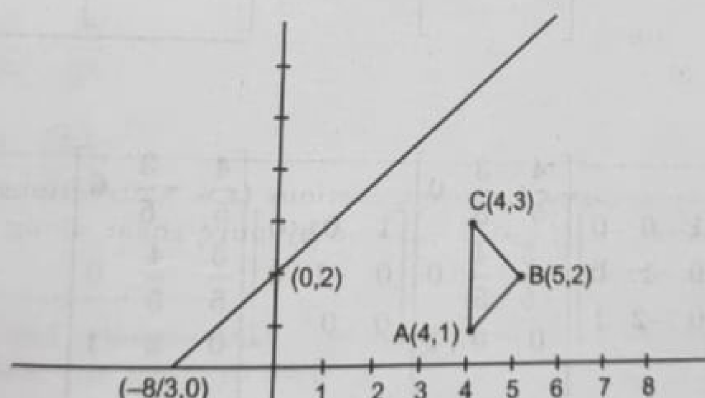


Fig. 8.19.

We know the composite transformation matrix $[T]$ for reflection about the line which does not pass through origin is

$$[T] = [T_R]\,[R_\theta]\,[R_{ref}]\,[R_\theta]^{-1}\,[T_R]^{-1}$$

$$[T_R] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix} \quad [T_R]^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}$$

$$R_\theta = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad [R_\theta]^{-1} = \begin{pmatrix} \cos(-\theta) & \sin(-\theta) & 0 \\ -\sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$(R_\theta)^{-1} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Reflection about x-axis *i.e.*,

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

$$\therefore \quad [T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

Put the values of $\cos\theta$ and $\sin\theta$, we get

$$[T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} \frac{4}{5} & \frac{3}{5} & 0 \\ -\frac{3}{5} & \frac{4}{5} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \frac{4}{5} & -\frac{3}{5} & 0 \\ \frac{3}{5} & \frac{4}{5} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} \frac{4}{5} & \frac{3}{5} & 0 \\ -\frac{3}{5} & \frac{4}{5} & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{4}{5} & -\frac{3}{5} & 0 \\ \frac{3}{5} & \frac{4}{5} & 0 \\ 0 & 2 & 1 \end{bmatrix}}$$

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

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} \frac{7}{25} & -\frac{24}{25} & 0 \\ -\frac{24}{25} & -\frac{7}{25} & 0 \\ 0 & 2 & 1 \end{bmatrix} = \begin{bmatrix} \frac{7}{25} & -\frac{24}{25} & 0 \\ -\frac{24}{25} & -\frac{7}{25} & 0 \\ \frac{48}{25} & \frac{64}{25} & 1 \end{bmatrix}$$

Matrix for triangle ABC can be written as

$$\begin{array}{c} A \\ B \\ C \end{array} \begin{bmatrix} 4 & 1 & 1 \\ 5 & 2 & 1 \\ 4 & 3 & 1 \end{bmatrix}$$

The reflected co-ordinates can be calculated as follows.

$$\begin{bmatrix} 4 & 1 & 1 \\ 5 & 2 & 1 \\ 4 & 3 & 1 \end{bmatrix} \begin{bmatrix} \frac{7}{25} & -\frac{24}{25} & 0 \\ -\frac{24}{25} & -\frac{7}{25} & 0 \\ \frac{48}{25} & \frac{64}{25} & 1 \end{bmatrix} = \begin{bmatrix} \frac{52}{25} & -\frac{39}{25} & 1 \\ \frac{35}{25} & -\frac{70}{25} & 1 \\ \frac{4}{25} & \frac{53}{25} & 1 \end{bmatrix}$$

Thus the reflected co-ordinates are

$$A \rightarrow \left(\frac{52}{25}, -\frac{39}{25}\right)$$

$$B \rightarrow \left(\frac{35}{25}, -\frac{70}{25}\right)$$

$$C \rightarrow \left(\frac{4}{25}, \frac{53}{25}\right)$$

EXAMPLE 8.21.

Consider the line L and triangle ABC. The equation of line L is $y = 1/2\ (x + 4)$ and $A(2, 4)$, $B(4, 6)$, $C(2, 6)$. Reflect the triangle about L.

Solution: Equation of line $y = 1/2\ (x + 4)$ or $2y = x + 4$, or $x - 2y + 4 = 0$

$$m = -\frac{1}{2} \text{ so } \tan\theta = -\frac{1}{2}$$

$$\therefore \quad \sin\theta = -\frac{1}{\sqrt{5}} \quad \cos\theta = \frac{2}{\sqrt{5}}$$

The intersection of the line $x - 2y + 4 = 0$ with

$x = 0$ is $y = 2 \Rightarrow (0, 2)$

$y = 0$ is $x = -4 \Rightarrow (0, -4)$

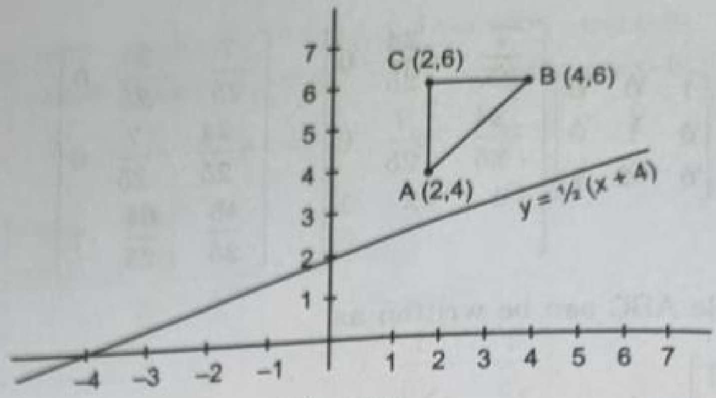


Fig. 8.20.

1. The line L passes through the origin by translation it 2 units in y-direction.
2. The resulting line can be made coincidents with the x-axis by rotating it θ about the origin.
3. Reflect the triangle through x-axis.
4. Inverse rotation
5. Translate back to original orientation.

i.e., $[T] = [T_R]\,[R_\theta]\,[R_{ref}]\,[R_\theta]^{-1}\,[T_R]^{-1}$

$$[T_R] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \quad [T_R]^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

$$R_\theta = \begin{bmatrix} \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad [R_\theta]^{-1} = \begin{bmatrix} \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 0 \\ -\frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[R_{ref}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Now
$$[T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 0 \\ -\frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 0 \\ -\frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & 0 \\ 0 & 0 & 1 \end{bmatrix}}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{5}} & 0 \\ 0 & 2 & 1 \end{bmatrix}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} \frac{3}{5} & \frac{4}{5} & 0 \\ \frac{4}{5} & -\frac{3}{5} & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

$$[T] = \begin{bmatrix} \frac{3}{5} & \frac{4}{5} & 0 \\ \frac{4}{5} & -\frac{3}{5} & 0 \\ -\frac{8}{5} & \frac{16}{5} & 1 \end{bmatrix}$$

The reflected co-ordinates can be calculated as follows

$$\begin{bmatrix} 2 & 4 & 1 \\ 4 & 6 & 1 \\ 2 & 6 & 1 \end{bmatrix} \begin{bmatrix} \frac{3}{5} & \frac{4}{5} & 0 \\ \frac{4}{5} & -\frac{3}{5} & 0 \\ -\frac{8}{5} & \frac{16}{5} & 1 \end{bmatrix} = \begin{bmatrix} \frac{14}{5} & \frac{12}{5} & 1 \\ \frac{28}{5} & \frac{14}{5} & 1 \\ \frac{22}{5} & \frac{6}{5} & 1 \end{bmatrix}$$

Thus the reflected co-ordinates are

$A \to (14/5,\ 12/5)$

$B \to (28/5,\ 14/5)$

$C \to (22/5,\ 6/5)$

EXAMPLE 8.22.

Prove that two 2-D rotations above the origin commutative i.e. $R_1R_2 = R_2R_1$.

(UPTU 2003)

Solution: Let R_1 be the rotation by the angle θ and R_2 be the rotation followed by R_1 in same direction by angle α.

$$R_1 = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad R_2 = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_1R_2 = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$$

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

$$= \begin{bmatrix} \cos(\theta+\alpha) & -\sin(\theta+\alpha) \\ \sin(\theta+\alpha) & \cos(\theta+\alpha) \end{bmatrix} \quad \text{—(A)}$$

Similarly we get,

$$R_2R_1 = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

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

$$= \begin{bmatrix} \cos(\theta+\alpha) & -\sin(\theta+\alpha) \\ \sin(\theta+\alpha) & \cos(\theta+\alpha) \end{bmatrix} \quad \text{—(B)}$$

From equation (A) and (B) $R_1R_2 = R_2R_1$.

EXAMPLE 8.23.

Prove that two scaling transformations are commutative i.e., $S_1S_2 = S_2S_1$.

Solution: Let S_1 is scaling by factor 'm' about origin and S_2 is scaling by factor 'n' after S_1.

$$S_1 = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$S_2 = \begin{bmatrix} n & 0 & 0 \\ 0 & n & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S_1S_2 = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n & 0 & 0 \\ 0 & n & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} mn & 0 & 0 \\ 0 & mn & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Now,

$$S_2S_1 = \begin{bmatrix} n & 0 & 0 \\ 0 & n & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} mn & 0 & 0 \\ 0 & mn & 0 \\ 0 & 0 & 1 \end{bmatrix} = S_1S_2$$

So,

$$S_1S_2 = S_2S_1$$

8.6. TRANSFORMATION BETWEEN CO-ORDINATE SYSTEMS

Suppose that we have two co-ordinate systems in the plane. The first system is located at origin O and has coordinate axis x y. The second co-ordinate system is located at origin O' and has coordinate axis $x'y'$ as shown in the Fig 8.21.

Now each point in the plane has two coordinate descriptions: (x, y) or (x', y') depending on which coordinate system is used. If we think of the second system $x'y'$ as arising from a transformation applied to the first system xy, we say that a co-ordinate transformation has been applied. We can describe this transformation by determining how the (x', y') coordinates of a point P are related to (x, y) coordinates of the same point. For example, a distance measured in inches can be converted to the same distance measured in centimeters by means of a scale. The actual operation of the transformation is the same as already described; only the interpretation changes. The co-ordinates of the point, when multiplied by the transformation, represent the same point, only measured in different coordinates.

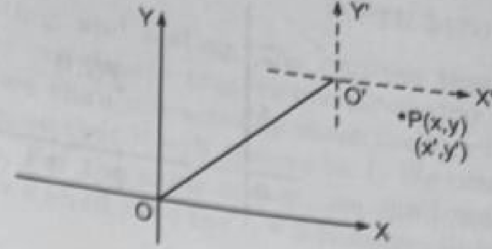


Fig. 8.21.

(a) **Translation:** Translations are useful coordinate transformations when the origins are not aligned. If the xy co-ordinate system is displaced to a new position, where the direction and distance of the displacement is given by the vector $V = t_x I + t_y J$, the coordinates of a point in both systems are related by the translation transformation $\overline{T_v}$.

$$(x', y') = \overline{T_v}\,(x, y) \quad \text{where} \quad x' = x - t_x \quad \text{and} \quad y' = y - t_y$$

or
$$\overline{T_v} = \begin{pmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{pmatrix}$$

(b) **Rotation about the Origin:** The xy system is rotated $\theta°$ about the origin. Then the co-ordinates of a point in both systems are related by the rotation transformation $\overline{R_\theta}$

$$(x', y') = \overline{R_\theta}\;(x, y)$$

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

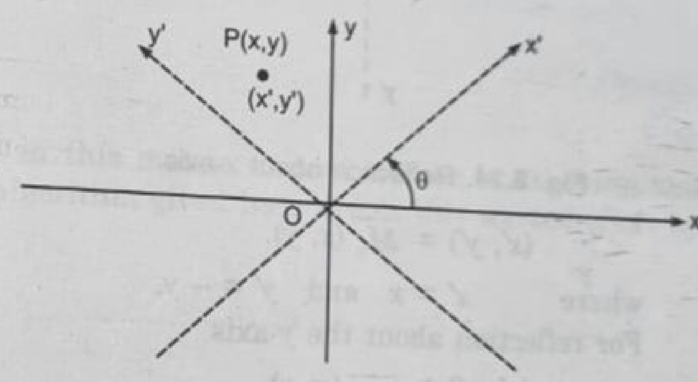


Fig. 8.22.

Then the matrix $\overline{R_\theta} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$

(e) **Scaling with respect to the origin:** Suppose that a new coordinate system is formed by leaving the origin and coordinate axis unchanged, but introducing different units of measurement along the x and y axis. If the new units are obtained from the old units by a scaling of S_x along the x axis and S_y along the y-axis, the coordinates in the new system are related to coordinates in the old system through the scaling transformation $\bar{S}_{s_x \cdot s_y}$

$$(x', y') = \bar{S}_{s_x, s_y}\,(x, y)$$

where $\quad x' = (1/S_x)x$ and $y' = (1/S_y)y$.

Figure shows co-ordinate scaling transformation using scaling factors $S_x = 2$ and $S_y = 1/3$.

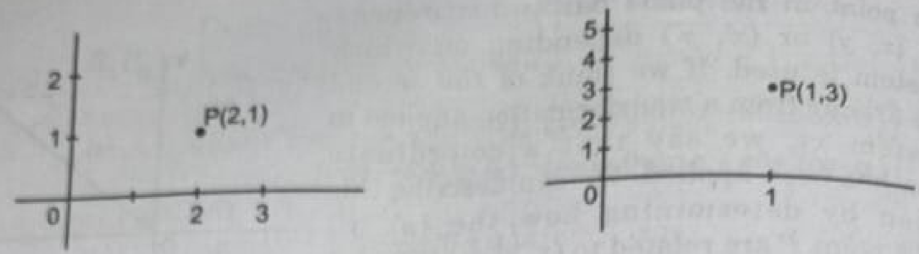


Fig. 8.23.

(*d*) **Mirror reflection about an axis:** If the new co-ordinate system is obtained by reflecting the old system about either x or y axis, then new co-ordinates is given by the coordinate transformations $\overline{M_x}$ and $\overline{M_y}$. For reflection about the x-axis.

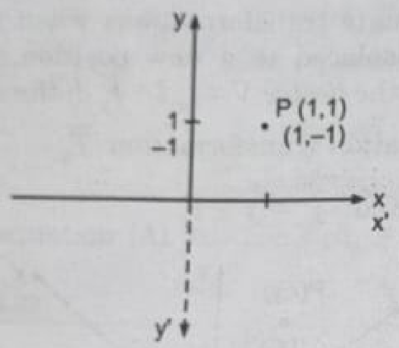


Fig. 8.24. Reflection about x-axis

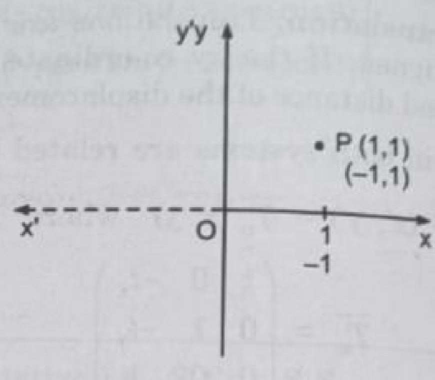


Fig. 8.25. Reflection about x-axis

$$(x', y') = \overline{M_x}\,(x, y).$$

where $\quad x' = x$ and $y' = -y$.

For reflection about the y-axis

$$(x', y') = \overline{M_y}\,(x, y).$$

where $\quad x' = -x$ and $y' = y$

8.7. AFFINE TRANSFORMATION

All the two dimensional transformations where each of the transformed co-ordinates x' and y' is a linear function of the original co-ordinates x and y as

$$x' = A_1 x + B_1\, y + C_1$$
$$y' = A_2\, x + B_2\, y + C_2$$

where A_i, B_i, C_i are parameters fixed for a given transformation type.

Affine transformations have the general properties that parallel lines all transformed into parallel lines and finite points map to finite points.

Translation, rotation, scaling, reflection and shear are examples of two-dimensional affine transformations. Any general two-dimensional affine transformation can always be expressed as a composition of these five transformations.

An affine transformation involving only rotation, translation and reflection preserves angles and lengths as well as parallel lines. For these three transformations, the lengths and angle between two lines remains the same after the transformation.

8.8. TRANSFORMATION ROUTINES

(UPTU 2009)

Now are shall construct routines for translating, rotating, and scaling. The routines that create the transformations will modify a homogeneous coordinate transformation matrix. While we use the notion of homogeneous coordinates, we shall not actually store the third coordinate instead, we shall arrange our transformation so that it will always be 1. We can therefore just use 1 whenever we need this coordinate. For the same reason, we shall not store the last column of the transformation matrix. So, we shall save the 3 × 3 homogeneous transformation matrix in the 3 × 2 array named H.

We begin with a routine to set the transformation matrix to the identity matrix.

(a) Routine to create an identity matrix

```
BEGIN
    FOR I = 1 to 3 Do
        For J = 1 to 2 Do
            IF I = J THEN H[I, J] ← 1
            ELSE H[I, J] ← 0;
    RETURN;
END;
```

(b) Routine to scaling transformation: We know the scaling matrix is

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

We notice that there are a lot of zero in this matrix and we will be multiplying and adding many noncontributing terms. The algorithm given here avoids this wasted effort by only considering with the non-zero terms.

```
BEGIN
    FOR I = 1 TO 3 DO
        BEGIN
            H[I, 1] ← H[I, 1] * Sx ;
            H[I, 2] ← H[I, 2] * Sy ;
        END;
    RETURN;
END;
```

Here S_x and S_y are the x-scale factor and y-scale factor respectively.

(c) Routine for translation: In case of translation, we again neglect the zero terms for simplification and get the following routine.

```
BEGIN
    H[3, 1] ← H[3, 1] + TX ;
    H[3, 2] ← H[3, 2] + TY ;
    RETURN;
END;
```

(d) Routine for rotation: This procedure takes the angle as an argument, calculates the sin and cos and then performs the matrix multiplication for non-zero terms.

Suppose A is the angle of counterwise rotation
S and C are the sine and cosine values
TEMP is the temporary storage of the first column

```
BEGIN
  C ← cos (A) ;
  S ← sin (A) ;
  FOR I = 1 TO 3 DO
    BEGIN
      TEMP ← H[I, 1] * C - H[I, 2] * S ;
      H[I, 2] ← H[I, 1] * S + H[I, 2] * C;
      H[I, 1] ← TEMP;
    END;
  RETURN;
END;
```

The above routines serve to create a transformation matrix, but we still need a routine to apply the resulting transformation. The following routine transforms a single point. The point coordinates are passed to the subroutine as arguments, and the transformed point is returned in the same variables.

Here, *X*, *Y* the co-ordinates of the point to be transformed

```
BEGIN
  TEMP ← X * H[1, 1] + Y * H[2, 1] + H[3, 1] ;
  Y ← X * H[1, 2] + Y * H[2, 2] + H[3, 2] ;
  X ← TEMP ;
  RETURN ;
END ;
```

To perform a transformation, we must form the appropriate transformation matrix and then apply it to the points in our display.

8.9. DISPLAY PROCEDURES

There may be times, when the image transformation is inadequate, when the user may wish to perform more than just a single scale, rotation and translation operation. For example, a rotation about an arbitrary center point is easily handled by a translation, followed by another translation. In these situations, the user should add one or more transformation levels to the system. A transformation matrix designed and created by the user may be as complex as necessary, involving many component transformations. Routines written according to this prescription will transform point values before they are entered into the display file.

The calls which involve establishment of a transformation are called **display procedure** calls and the subprograms which draw subpictures are known as display procedures. Because isplay procedure calls can be nested, there can be multiple transformations and the overall ransformation would be the product of these transformations. We can see that each time a ew display procedure is executed and the overall transformation matrix is multiplied on the ft by the transformation for that display procedure. Here is a case where pre-multiplication appropriate and each display procedure call must save the current overall transformation atrix. Because these can be nested calls, several transformation matrices may have to be

...red simultaneously. One possible data structure for storing these matrices is a stack. That is a display procedure call involves the following;

1. Saving the overall transformation matrix.
2. Multiplying the overall transformation matrix on the left by the transformation in the call to form a new overall transformation matrix.
3. Transferring control to the display procedure

and a return from a display procedure involves.

(i) Restoring the overall transformation matrix from the value saved.
(ii) Returning control to the calling program.

Summary

2D transformation means a change in either position or orientation or size or shape of graphics objects like line, circle, arc, ellipse, rectangle, polygon, polylines etc. A points coordinates can be expressed as elements of a matrix. Two matrix format are used: one is row matrix and other is column matrix format. A general procedure for applying translation, rotation and scaling parameters to reposition and resize 2D objects is called basic transformation. The process of calculating the product of matrices of a number of different transformations in a sequence is known as concatenation or composite transformation matrix.

Graphics applications often require the transformation of object require the transformation of object descriptions from one coordinate system to another. This is called transformation between co-ordinate systems.

Exercises

1. Why are homogeneous coordinates used for transformation computations in computer Graphics?
2. Determine a sequence of basic transformations that are equivalent to the x-direction shearing matrix.
3. Determine the homogeneous transformation matrix for reflection about the line $y = mx + b$ or $y = 2x - 6$.
4. Explain inverse transformation. Derive the matrix for inverse transformation.
5. Prove that two successive two-dimensional rotations are additive.
6. How do you perform shear in two dimensions.
7. Show that a reflection about the line $y = -x$ is equivalent to a reflection relative to y axis followed by a counterclockwise rotation of $90°$.
8. Find the transformation that converts a square with diagonal vertices (0, 3) and (–3, 6) into a unit square at the origin.
9. A triangle is located at $P(10, 40)$ $Q(40, 40)$, $R(40, 30)$. Work out the transformation matrix which would rotate the triangle by $90°$ counterclockwise about the point Q. Find the co-ordinates of the rotated triangle.
10. Reflect the diamond-shaped polygon whose vertices are $A(-1, 0)$, $B(0, -2)$, $C(1, 0)$ and $D(0, 2)$ about
    (i) The horizontal line $y = 2$
    (ii) The vertical line $x = 2$ and
    (iii) The line $y = x + 2$

current point to get the next point on the line. This is continued till the end of line.

5.4. DIGITAL DIFFERENTIAL ANALYZER (DDA) ALGORITHM

The vector generation algorithms (and curve generation algorithms) which step along the line (or curve) to determine the pixels which should be turned ON are sometimes called **digital differential analyzers** (DDAs). The name comes from the fact that we use the same technique as a numerical method for solving differential equations.

We know that the slope of a straight line is given as

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{...(a)}$$

This differential equation can be used to obtain a rasterized straight line.

For any given Δx (x interval) along a line, we can compute the Δy (y interval) as

$$\Delta y = m \cdot \Delta x$$

i.e.,
$$\Delta y = \frac{y_2 - y_1}{x_2 - x_1} \Delta x$$

Similarly, we can obtain the x interval Δx corresponding to Δy as $\Delta x = \dfrac{\Delta y}{m} = \dfrac{x_2 - x_1}{y_2 - y_1} \Delta y$

Once the intervals are known, the next values for x and y are obtained as

$$x_{i+1} = x_i + \Delta y$$

$$\boxed{x_{i+1} = x_i + \left(\frac{x_2 - x_1}{y_2 - y_1}\right) \Delta y} \qquad \text{...(i)}$$

and
$$y_{i+1} = y_i + \Delta x$$

$$\boxed{y_{i+1} = y_i + \frac{y_2 - y_1}{x_2 - x_1} \Delta x} \qquad \text{...(ii)}$$

For simple DDA either Δx or Δy, whichever is larger, is chosen as some raster unit, *i.e.*,

If $|\Delta x| \geq |\Delta y|$ then

$\Delta x = 1$

else $\Delta y = 1$

if $\Delta x = 1$ then

$$y_{i+1} = y_i + \frac{y_2 - y_1}{x_2 - x_1} \text{ and}$$

$$x_{i+1} = x_i + 1$$

If $\Delta y = 1$ then

$$y_{i+1} = y_i + 1$$

$$x_{i+1} = x_i + \frac{x_2 - x_1}{y_2 - y_1}$$

DDA algorithm

1. Read the line end points (x_1, y_1) and (x_2, y_2)
2. $\Delta x = |x_2 - x_1|$
   $\Delta y = |y_2 - y_1|$
3. If $(\Delta x \geq \Delta y)$ then
   length $= \Delta x$
   else
   length $= \Delta y$
4. Select the raster unit, *i.e.*,

$$\Delta x = \frac{(x_2 - x_1)}{length}$$

$$\Delta y = \frac{(y_2 - y_1)}{length}$$

> **Note:** Either Δx or Δy will be one because length is either $x_2 - x_1$ or $y_2 - y_1$. Thus the incremental value for x or y will be one.

5. $$x = x_1 + 0.5 * \text{Sign}(\Delta x)$$
$$y = y_1 + 0.5 * \text{Sign}(\Delta y)$$

Sign function makes the algorithm work in all quadrant. It returns – 1, 0, 1 depending on whether its agreement is < 0, = 0, > 0 respectively. The factor 0.5 makes it possible to round the values in the integer function rather than truncating them.

6. Now plot the points

```
i = 1
while (i ≤ length)
{
    Plot (integer (x), integer (y))
    x = x + Δx
    y = y + Δx
    i = i + 1
}
```

7. Stop.

Advantages of DDA algorithm

1. DDA algorithm is faster than the direct use of the line equation since it calculates points on the line without any floating point multiplication.
2. It is a simplest algorithm and does not require special skills for implementation.

Direct use of the line equation

A simple approach to scan converting a line is to first scan-convert P_1 and P_2 to pixel coordinates (x_1', y_1') and (x_2', y_2') respectively then $m = \dfrac{y_2' - y_1'}{x_2' - x_1'}$ and $c = y_1' - mx_1'$.

If $|m| \leq 1$, then for every integer value of x between and excluding x_1' and x_2' calculate the corresponding value of y using equation and scan convert (x, y).

If $|m| > 1$, then for every integer value y between and excluding y_1' and y_2' calculate the corresponding value of x using the equation and scan-convert (x, y).

Disadvantages of DDA algorithm

1. It is orientation dependent, due to this the end point accuracy is poor.
2. A floating-point addition is still needed in determining each successive point which is time consuming. Furthermore, cumulative error due to limited precision in the floating-point representation may cause calculated points to drift away from their true position when the line is relatively long.

EXAMPLE. 5.1

Consider a line from (0, 0) to (4, 6). Use the simple DDA algorithm to rasterize this line.

Solution:

$$(x_1, y_1) = (0, 0)$$
$$(x_2, y_2) = (4, 6)$$
$$\Delta x = 4 - 0 = 4$$
$$\Delta y = 6 - 0 = 6$$

$\Delta x < \Delta y$ so length $= \Delta y = 6$

$$\therefore \quad \Delta x = \frac{x_2 - x_1}{\text{length}} = \frac{4}{6} = 0.666 \approx 0.667$$

$$\Delta y = \frac{y_2 - y_1}{\text{length}} = \frac{6}{6} = 1$$

$$x = 0 + 0.5 \text{ sign } (4/6) = 0.5$$
$$y = 0 + 0.5 \text{ sign } (1) = 0.5$$

Plot (integer (0.5), integer (0.5)) *i.e.*, Plot (1, 1)

$$x = x + \Delta x$$
$$x = 0.5 + 0.667 = 1.167$$
$$y = 0.5 + 1 = 1.5$$

Plot (integer (1.167), integer (1.5)) *i.e.*, Plot (1, 2)

$$x = 1.167 + 0.667 = 1.833$$
$$y = 1.5 + 1 = 2.5$$

Plot (integer (1.833), integer (2.5)) *i.e.*, Plot (2, 3)

$$x = 1.833 + 0.667 = 2.5$$
$$y = 2.5 + 1 = 3.5$$

Plot (integer (2.5), integer (3.5)) *i.e.*, Plot (3, 4)

$$x = 2.5 + 0.667 = 3.167$$
$$y = 3.5 + 1 = 4.5$$

Plot (3, 5)

$$x = 3.167 + 0.667 = 4.5$$
$$y = 4.5 + 1 = 5.5$$

The results are shown in the figure below.

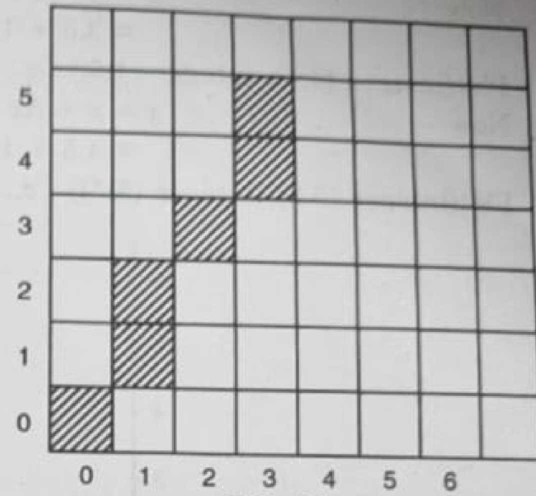


Fig. 5.4.

EXAMPLE 5.2

Implement the DDA algorithm to draw a line from (0, 0) to (6, 6)

Solution:

$$(x_1, y_1) = (0, 0)$$
$$(x_2, y_2) = (6, 6)$$
$$\Delta x = x_2 - x_1 = 6 - 0 = 6$$
$$\Delta y = y_2 - y_1 = 6 - 0 = 6$$
$$\Delta x = \Delta y \text{ so length } = \Delta x = 6$$

Now the raster unit *i.e.*, $\Delta x = \dfrac{(x_2 - x_1)}{\text{length}} = \dfrac{6}{6} = 1$

and

$$\Delta y = \frac{6}{6} = 1$$

$$x = x_1 + 0.5 \text{ sign } (1) = 0 + 0.05 \times 1 = 0.5$$
$$y = y_1 + 0.5 \text{ sign } (-1) = 0 + 0.5 \times 1 = 0.5$$

Plot (integer (0.5), integer (0.5)) *i.e.*, **plot (1,1)**

Now

$$x = x + \Delta x \quad \text{and} \quad y = y + \Delta y$$
$$= 0.5 + 1 = 1.5 \quad \text{and} \quad = 0.5 + 1 = 1.5$$

Plot (integer (1.5), integer (1.5)) *i.e.*, **plot (2, 2)**

Now

$$x = x + \Delta x \quad \text{and} \quad y = y + \Delta y$$

$1.5 + 1 = 2.5$ $\qquad\qquad = 1.5 + 1 = 2.5$

Plot(integer (2.5), integer (2.5)) *i.e.*, **plot (3, 3)**

Now $\quad x = x + \Delta x$ and $y = y + \Delta y$

$= 2.5 + 1 = 3.5 \qquad = 2.5 + 1 = 3.5$

Plot(integer (3.5), integer (3.5)) *i.e.*, **plot (4, 4)**

Now $\quad x = x + \Delta x$ and $y = y + \Delta y$

$= 3.5 + 1 = 4.5 \qquad = 3.5 + 1 = 4.5$

Plot(integer (4.5), integer (4.5)) *i.e.*, **plot (5, 5)**

Now $\quad x = x + \Delta x$ and $y = y + \Delta y$

$= 4.5 + 1 = 5.5 \qquad = 4.5 + 1 = 5.5$

Plot(integer (5.5), integer (5.5)) *i.e.*, **plot (6, 6)**

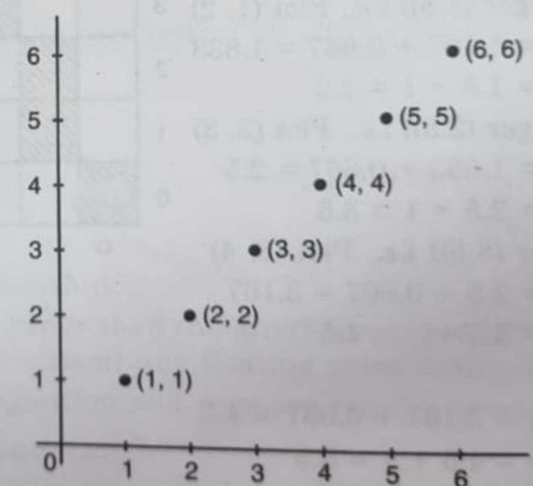


Fig. 5.5.

EXAMPLE 5.3.

Modify the simple straight li... ... lines.

5.5. BRESENHAM'S LINE ALGORITHM (UPTU, B.Tech, 2008-09; 2012-13)

One serious drawback of the DDA algorithm is that it is very time consuming as it deals with a rounding off operation and floating point arithmetic. The algorithm developed by Bresenham is more accurate and efficient compared to DDA algorithm because it cleverly avoids the "Round" function and scan converts line using only incremental integer calculation.

Bresenham's line algorithm uses only integer addition and subtraction and multiplication by 2, and we know that the compute can perform the operations of integer addition and subtraction very rapidly. The computer is also time-efficient when performing integer multiplication by powers of 2.

The basic principle of Bresenham's line algorithm is to find the optimum raster locations to represent the straight lines. To accomplish this the algorithm always increments either x or y by one unit depending on the slope of line. Once this is done, then increment in other variable is found on the basis of the distance between the actual line location and the nearest pixel. This distance is called **decision variable** or the **error term**.

Figure illustrated sections of a display screen where straight line segments are to be drawn. The vertical axis show scan-line, positions and the horizontal axis identify pixel columns.

Sampling at unit x intervals in these examples, we need to decide which of two possible pixel positions is closer to the line path at each sample step. Starting from the left endpoint shown in figure 5.6, we need to determine at the next sample position whether to plot the pixel at position (11, 11) or the one at (11, 12). Similarly, Figure 5.6(*b*) shown a negative slop line path starting from the left endpoint at pixel position (50, 50). In this one, do we select the next pixel position as (51, 50) or as (51, 49)? These questions are answered with Bresenham's line algorithm by testing the sign of an integer parameter, whose value is proportional to the difference between the separations of the two pixel positions from the actual line path.

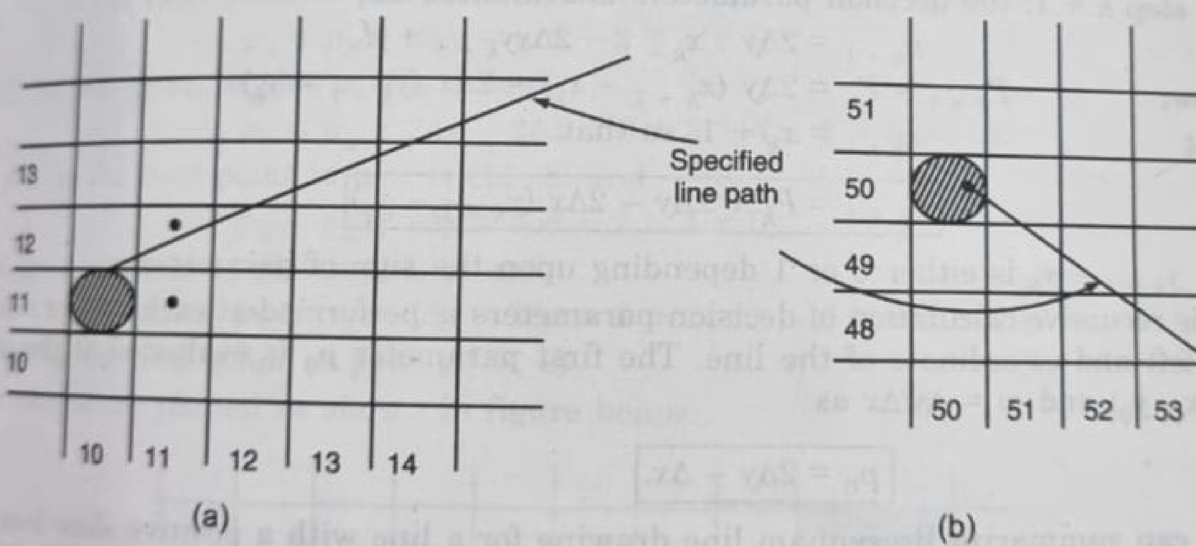


Fig. 5.6.

To illustrate Bresenham's approach, we first consider the scan-conversion process for lines with positive slope less than 1. Pixel positions along a line path are then determined by sampling at unit x intervals. Starting from the left end point (x_0, y_0) of a given line, we step to each successive column (x position) and plot the pixel whose scan-line y value is closest to the line path Figure demonstrates the kth step in this process. Assuming we have determined that the pixel at (x_k, y_k) is to be displayed, we next need to decide which pixel to plot in column x_{k+1}. Our choices are the pixels at positions $(x_k + 1, y_k)$ and $(x_k + 1, y_k + 1)$.

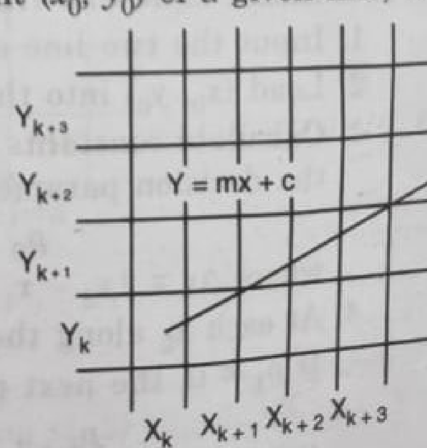


Fig. 5.7.

At sampling position $x_k + 1$, we label vertical pixel separations from the mathematical line path as d_1 and d_2. The y coordinate on the mathematical line at pixel column position $x_k + 1$ is calculated as

Then, $d_1 = y - y_k$
$= m(x_k + 1) + c - y_k$

and $d_2 = (y_k + 1) - y$
$= y_k + 1 - m(x_k + 1) - c$

The difference between these two separations is

$$d_1 - d_2 = 2m(x_k + 1) - 2y_k + 2c - 1$$

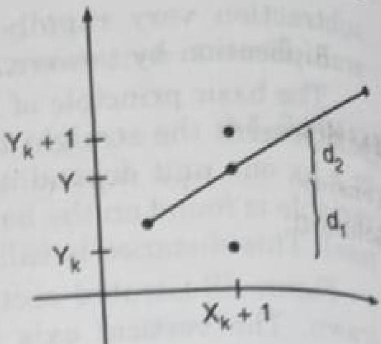


Fig. 5.8.

A decision parameter p_k for the kth step in the line algorithm can be obtained by rearranging equation so that it involves only integer calculations. We accomplish this by substituting $m = \Delta y/\Delta x$, where Δy and Δx are the vertical and horizontal separations of the endpoint positions, and defining :

$$p_k = \Delta x\ (d_1 - d_2) = 2\Delta y\ .\ x_k - 2\Delta x\ .\ y_k + d$$

The sign of p_k is the same as the sign of $d_1 - d_2$, since $\Delta x > 0$ for our example. Parameter d is constant and has the value $2\Delta y + \Delta x(2c - 1)$, which is independent of pixel position and will be eliminated in the recursive calculations for p_k.

If pixel at y_k is closer to the line path than the pixel at $y_k + 1$ that means $d_1 < d_2$ i.e., decisions parameters p_k is – ve. In that case we plot the lower pixel, otherwise, we plot the upper pixel.

At step $k + 1$, the decision parameters is evaluated as

$$p_{k+1} = 2\Delta y\ .\ x_{k+1} - 2\Delta x y_{k+1} + d.$$

Now, $P_{k+1} - P_k = 2\Delta y\ (x_{k+1} - x_k) - 2\Delta x\ (y_{k+1} - y_k)$

But $x_{k+1} = x_k + 1$, so that

$$\boxed{P_{k+1} = P_k + 2\Delta y - 2\Delta x\ (y_{k+1} - y_k)}$$

where $y_{k+1} - y_k$ is either 0 or 1 depending upon the sign of parameter p_k.

This recursive calculation of decision parameters is performed at each integer x starting at the left and co-ordinate of the line. The first parameter p_0 is evaluated at the starting point (x_0, y_0) and $m = \Delta y/\Delta x$ as

$$\boxed{p_0 = 2\Delta y - \Delta x.}$$

We can summarize Bresenham line drawing for a line with a positive slope less than 1 in the following listed steps. The constants $2\Delta y$ and $2\Delta y - 2\Delta x$ are calculated once for each line to be scan converted, so the arithmetic involves only integer addition and substraction of these two constants.

Bresenham's Line-drawing algorithm for $|m| < 1$

1. Input the two line endpoints store that left endpoint in (x_0, y_0).
2. Load (x_0, y_0) into the frame buffer; that is, plot the first point.
3. Calculate constants Δx, Δy, $2\Delta y$ and $2\Delta y - 2\Delta x$, and obtain the starting value for the decision parameter as

   $$p_0 = 2\Delta y - \Delta x$$

   where $\Delta x = |x_2 - x_1|$ and $\Delta y = |y_2 - y_1|$
4. At each x_k along the line, starting at $k = 0$, perform the following test :
   If $p_k < 0$, the next point to plot is (x_{k+1}, y_k) and

   $$p_{k+1} = p_k + 2\Delta y$$

   Otherwise, the next point to plot is $(x_k + 1, y_k + 1)$ and

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$

5. Repeat step 4 Δx times.

EXAMPLE. 5.4.

Consider the line from (5, 5) to (13, 9). Use the Bresenham algorithm to rasterize the line.

Solution:

$$(x_1, y_1) = (5, 5)\ (x_2, y_2) = (13, 9)$$

$$\Delta x = 13 - 5 = 8$$

$$\Delta y = 9 - 5 = 4$$

$$|m| = \frac{4}{8} = \frac{1}{2} < 1$$

$$p_0 = 2\Delta y - \Delta x = 2 \times 4 - 8 = 0$$

So, the next point is (6, 6) and

$$p_1 = p_0 + 2\Delta y - 2\Delta x = 0 + 2 \times 4 - 2 \times 8 = 8 - 16 = -8$$

$p_1 < 0$, so the next point to plot is (7, 6)

$$p_2 = p_1 + 2\Delta y = -8 + 2 \times 4 = 0$$

and

So, the next point to plot is (8, 7) and

$$p_3 = p_2 + 2\Delta y - 2\Delta x = 0 + 2 \times 4 - 2 \times 8 = 8 - 16 = -8$$

$p_3 < 0$, so the next point to plot is (9, 7) and

$$p_4 = p_3 + 2\Delta y = -8 + 2 \times 4 = 0$$

and the next point to plot is (10, 8) and

$$p_5 = p_4 + 2\Delta y - 2\Delta x = 0 + 2 \times 4 - 2 \times 8 = -8$$

$p_5 < 0$, so the next point to plot is (11, 8) and

$$p_6 = p_5 + 2\Delta y = -8 + 2 \times 4 = 0$$

and the next point to plot is (12, 9) and

$$p_7 = p_6 + 2\Delta y - 2\Delta x = 0 + 2 \times 4 - 2 \times 8 = 8 - 16 = -8$$

$p_7 < 0$ so, the next point to plot is (13, 9)

The results are plotted as shown in figure below

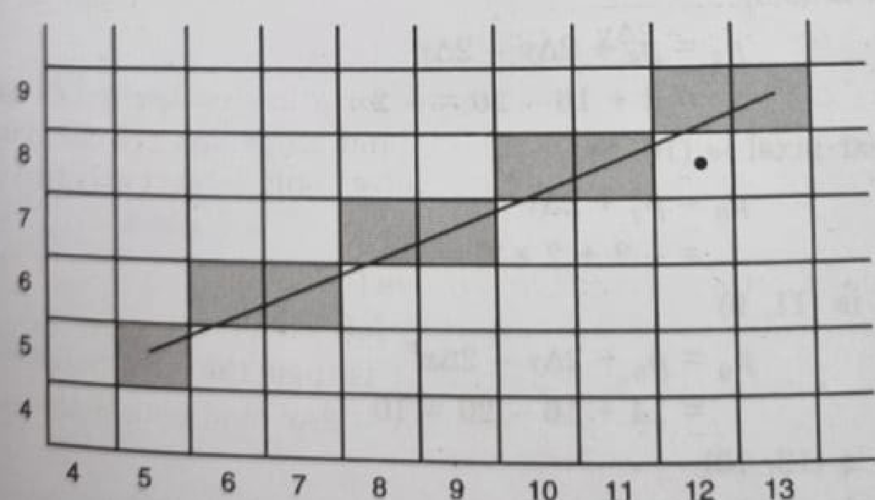


Fig. 5.9.

EXAMPLE. 5.5.

Using Bresenham's algorithm predict the pixels on the line from (2, 2) to (12, 10).

Solution:

$$(x_1, y_1) = (2, 2)\ (x_2, y_2) = (12, 10)$$

$$\Delta x = x_2 - x_1 = 12 - 2 = 10$$
$$\Delta y = y_2 - y_1 = 10 - 2 = 8$$
$$|m| = \frac{\Delta y}{\Delta x} = \frac{8}{10} < 1$$

Now
$$p_o = 2\Delta y - \Delta x$$
$$= 2 \times 8 - 10 = 16 - 10 = 6$$

Since $p_o > 0$ so next pixel is (3, 3)

and
$$p_1 = p_o + 2\Delta y - 2\Delta x$$
$$= 6 + 2 \times 8 - 2 \times 10 = 6 + 16 - 20 = 2$$

So, the next pixel is (4, 4)

Now,
$$p_2 = p_1 + 2\Delta y - 2\Delta x$$
$$= 2 + 16 - 20 = -2$$

Since $p_2 < 0$, so the next pixel is (5, 4)

Now,
$$p_3 = p_2 + 2\Delta y$$
$$= -2 + 2 \times 8 = 14$$

So, next pixel is (6, 5)

and
$$p_4 = p_3 + 2\Delta y - 2\Delta x$$
$$= 14 + 16 - 20 = 10$$

So, next pixel is (7, 6)

and
$$p_5 = p_4 + 2\Delta y - 2\Delta x$$
$$= 10 + 16 - 20 = 6$$

So, next pixel is (8, 7)

and
$$p_6 = p_5 + 2\Delta y - 2\Delta x$$
$$= 6 + 16 - 20 = 2$$

So, next pixel is (9,8)

and
$$p_7 = p_6 + 2\Delta y - 2\Delta x$$
$$= 2 + 16 - 20 = -2$$

$p_7 < 0$ so, next pixel is (10, 8)

and,
$$p_8 = p_7 + 2\Delta y$$
$$= -2 + 2 \times 8 = 14$$

So, next pixel is (11, 9)

and
$$p_9 = p_8 + 2\Delta y - 2\Delta x$$
$$= 14 + 16 - 20 = 10$$

So, next pixel is (12, 10)

Thus, the pixels on the line are

(2, 2) (3, 3) (4, 4) (5, 4) (6, 5) (7, 6) (8, 7) (9, 8) (10, 8) (11, 9) and (12, 10)

This method is well suited for the line having the slope less than 45° *i.e.*, first octant. We can have the modification in these steps to draw the line in any quadrant. Such algorithm can easily developed by considering the quadrant in which the line lies and the stope. When the magnitude of the slope is greater than 1, y increment by one and the decision variable is use to determine when to increment x. The x and y incremental values depend in the quadrant in which the line exists. This is shown in Fig. 5.10.

Generalized Bresenham's Algorithm

1. Read the line end point (x_1, y_1) and (x_2, y_2) such that they are not equal
$$\Delta x = |x_2 - x_1|$$
2. $$\Delta y = |y_2 - y_1|$$
3. Initialize starting point $x = x_1$
$$y = y_1$$
$$S_1 = \text{Sign}\,(x_2 - x_1)$$
4. $$S_2 = \text{Sign}\,(y_2 - y_1)$$

[Sign function returns $-1, 0, 1$ depending on whether its agreement $< 0, = 0, > 0$ respectively].

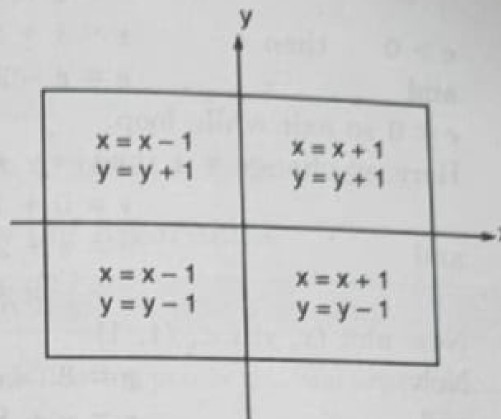


Fig. 5.10.

5. If $\Delta y > \Delta x$ then
   exchange Δx & Δy
   ex_change = 1

   else
   ex_change = 0
6. $e = 2 * \Delta y - \Delta x$
7. $i = 1$
8. plot (x, y)
9. while $(e \geq 0)$
```
{
    if (ex_change = 1) then
        x = x + s1
    else
        y = y + s2
    e = e - 2 * Δx
}
```
10. If ex_change = 1 then
    $y = y + s_2$
    else
    $x = x + s_1$
    $e = e + 2 * \Delta y$
11. $i = i + 1$
12. If $(i \leq \Delta x)$ then go to step (8)
13. stop.

EXAMPLE. 5.6.

Consider a line from (0, 0) to (6, 7). Use Bresenham's algorithm to rasterize the line.

Solution:

$$(x_1, y_1) = (0, 0) \quad (x_2\ y_2) = (6, 7)$$
$$x = 0 \quad \Delta x = |x_2 - x_1| = 6$$
$$y = 0 \quad \Delta y = |y_2 - y_1| = 7$$

$\Delta y > \Delta x$ so exchange Δx and Δy and

Now $\quad \Delta x = 7$

$\Delta y = 6$

plot (0, 0) $\quad e = 2 * \Delta y - \Delta x$ $\quad \begin{cases} S_1 = 1 \\ S_2 = 1 \end{cases}$

$= 2 \times 6 - 7 = 5 \quad$ *i.e.*, $e = 5$

$e > 0$ then $\quad x = x + S_1 = 0 + 1 = 1$

and $\quad e = e - 2 * \Delta x = 5 - 2 * 7 = 5 - 14 = -9$

$e < 0$ so exit while loop.

Here ex_change = 1 then $y = y + S_2$

$y = 0 + 1 = 1 \quad y = 1$

and $\quad e = e + 2 * \Delta y = -9 + 2 * 6$

$= -9 + 12 = 3, \quad$ *i.e.*, $\quad e = 3$

Now plot (x, y) *i.e.*, (1, 1)

Now $\quad e = 3$ *i.e.*, $e \geq 0$

and $\quad x = x + S_1 = 1 + 1$

$x = 2$

$e = e - 2 * \Delta x = 3 - 2 * 7 = 3 - 14 = -11$

$y = y + S_2 = 1 + 1 = 2$

$e = e + 2 * \Delta y = -11 + 2 * 6 = 1$

$e = 1$

Plot (x, y) *i.e.*, (2, 2).

Now $\quad e = 1$ i.e., $e \geq 0$

$x = 2 + 1 = 3$

$e = e - 2 * \Delta x = 1 - 2 * 7 = -13$

$y = 2 + 1 = 3$

$e = e + 2 * \Delta y = -13 + 2 * 6 = -1$

$e = -1$

Plot (x, y) *i.e.*, (3, 3)

Now $\quad e = -1$ *i.e.*, $e < 0$

$y = y + S_2$

$y = 3 + 1 = 4$

$e = e + 2 * \Delta y = -1 + 2 * 6 = 11$

Plot (x, y) *i.e.*, (3, 4)

Now $\quad e = 11$ *i.e.*, $e \geq 0$

$x = 3 + 1 = 4$

$e = e - 2 * \Delta x = 11 - 2 * 7 = -3$

$y = 4 + 1 = 5$

$e = e + 2 * \Delta y = -3 + 2 * 6 = 9$

Plot (x, y) *i.e.*, (4, 5)

Now $\quad e = 9$ *i.e.*, $e \geq 0$

$x = 4 + 1 = 5$

$e = e - 2 * \Delta x = 9 - 2 * 7 = -5$

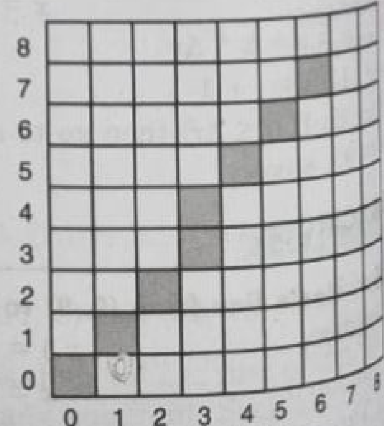


Fig. 5.11.

$$y = 5 + 1 = 6$$
$$e = e + 2 * \Delta y = - 5 + 2 * 6 = 7$$

Plot (x, y) i.e., (5, 6)

Now

$$e = 7 \quad i.e., \quad e \geq 0$$
$$x = 5 + 1 = 6$$
$$e = e - 2 * \Delta x = 7 - 2 * 7 = - 7$$
$$y = 6 + 1 = 7$$
$$e = e + 2 * \Delta y = - 7 + 2 * 6 = 5$$

Plot (6, 7)

Thus the pixels to be plotted are (0, 0) (1, 1) (2, 2) (3, 3) (3,4) (4, 5) (5, 6) (6, 7).

Difference between Bresenham and DDA Line Drawing Algorithms

Bresenham line drawing algorithm	DDA line drawing algorithm
1. Bresenham algorithm uses fixed points i.e., integer arithmetic.	1. DDA uses floating points i.e., real arithmatic.
2. Bresenham uses only subtraction and addition in its operators.	2. DDA uses multiplication and division in its operators.
3. Bresenham is faster than DDA.	3. DDA is rather slower than Bresenham algorithm.
4. Bresenham algorithm is more efficient and much accurate than DDA algorithm.	4. DDA algorithm is not as accurate and efficient as Bresenham algorithm.
5. Bresenham algorithm is less expensive than DDA algorithm.	5. DDA algorithm uses an enormous number of floating point multiplications so it is expensive.
6. Bresenham algorithm does not round off but takes the incremental value in its operation.	6. DDA algorithm round off the coordinates to integer that is nearest to the line. Rounding off of the pixel position obtained by multi plication or division causes an accumulation of error in the proceeding pixels.

One note concerning efficiency, fixed point DDA algorithm is generally superior to Bresenham's algorithm on modern computers. The reason is that Bresenhams algorithm uses a conditional branch in the loop and this result in frequent branch misprediction in the CPU. Fixed point DDA also has fewer instructions in the loop body (one bit shift, one increment and one addition to be exact. In addition to the loop instruction and the actual plotting). Since fixed point DDA does not require conditional jumps, we can compute several lines in parallel with SIMD (Single Instruction Multiple Data) techniques.

5.6. CIRCLE-GENERATING ALGORITHMS

5.6.1. Basic Concepts in Circle Drawing

A circle is a symmetrical figure. It has eight-way symmetry *i.e.*, the shape of the circle is similar in each quadrant as shown in the figure.

Thus, any circle generating algorithm can take advantage of the circle symmetry to plot eight points by calculating the co-ordinates of any one point. For example, calculation of a

Therefore now ... $_{n+1}$ in the equation of y_{n+1} ... to make one correction in the equation. We

$$x_{n+1} = x_n + \in y_n$$

$$y_{n+1} = y_n - \in x_{n+1}$$

i.e.,

$$y_{n+1} = (1 - \in^2)\, y_n - \in x_n$$

so,

$$[x_{n+1} \ \ y_{n+1}] = [x_n \ y_n] \begin{bmatrix} 1 & -\in \\ \in & 1-\in^2 \end{bmatrix}$$

and the value of $\begin{bmatrix} 1 & -\in \\ \in & 1-\in^2 \end{bmatrix}$ is 1.

5.7.2. Midpoint Circle Algorithm

As in the raster line algorithm, we sample at unit intervals and determine the closest pixel position to the specified circle path at each step. We can set up our algorithm to calculate pixel positions around a circle path centered at the co-ordinate origin (0, 0). Then each calculated position (x, y) is moved to its proper screen position by adding x_c to x and y_c to y, where (x_c, y_c) is the actual centre of the circle with radius r.

Only one octant of the circle needs to be generated. The other parts can be obtained by successive reflections. If the first octant (0 to 45°) is generated the second octant can be obtained by reflection through the line $y = x$ to yield the first quadrant. The results in the first quadrant are reflected through line $x = 0$, to obtain those in the second quadrant. The combined results on the upper semicircle are reflected through the line $y = 0$ to complete the circle.

To apply the midpoint method, we define the equation of a circle with (0, 0) as its centre as

$$F_{circle}(x, y) = x^2 + y^2 - r^2$$

The relative position of any point (x, y) can be determined by checking the sign of the circle function

$$F_{circle}(x, y) = \begin{cases} <0 & \text{if } (x,y) \text{ is inside the circle boundary} \\ =0 & \text{if } (x,y) \text{ is on circle boundary} \\ >0 & \text{if } (x,y) \text{ is outside the circle boundary} \end{cases}$$

Thus, the circle function is the decision parameter in the midpoint algorithm and we can set up incremental calculations for this function as we did in the line algorithm.

In Fig. 5.15 we have plotted the pixel at (x_k, y_k), we next need to determine whether the pixel at position (x_{k+1}, y_k) or one at position (x_{k+1}, y_{k-1}) or at position (x_k, y_{k-1}) is closer to the circle.

Firstly, we check that if the diagonal point is inside or outside the circle *i.e.*,

$\Delta = (x_{k+1})^2 + (y_{k-1})^2 - r^2$ is < 0, $= 0$ or > 0

If $\boxed{\Delta < 0}$, then the diagonal point is inside the circle *i.e.*, we have to select one of the two point (x_{k+1}, y_k) or (x_{k+1}, y_{k-1}), the one which is more closer to the circle boundary. For this, we check for the midpoint between these two points *i.e.*,

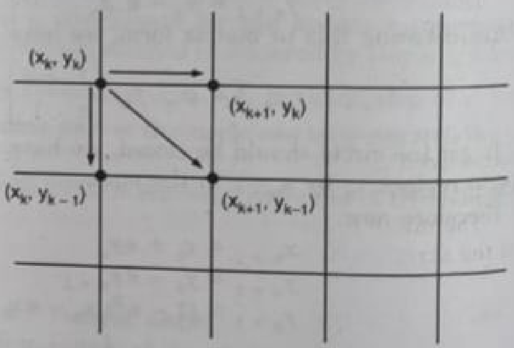


Fig. 5.15.

$$p_k = F_{circle}\left(x_k + 1, y_k - \frac{1}{2}\right)$$

$$= (x_k + 1)^2 + \left(y_k - \frac{1}{2}\right)^2 - r^2$$

If $p_k > 0$, this midpoint is outside the circle and the pixel on the scan line y_{k-1} is closer to the circle boundary otherwise the mid position is inside or on the circle boundary and we select the pixel on scan line y_k.

Similarly, if $\boxed{\Delta > 0}$, then the diagonal point is outside the circles, *i.e.*, we have to select one of the two points (x_{k+1}, y_{k-1}) and (x_k, y_{k-1}) the one which is more closer to the circle boundary. For this, once again we check for the midpoint between these two points *i.e.*,

$$p_k = \left(x_k + \frac{1}{2}\right)^2 + (y_{k-1})^2 - r^2$$

If $p_k > 0$, this midpoint is outside the circle and the pixel (x_k, y_{k-1}) is selected, otherwise the midpoint is inside or on the circle boundary $(p_k < 0)$ and we select the pixel (x_{k+1}, y_{k-1})

If $\boxed{\Delta = 0}$, pixel (x_{k+1}, y_{k-1}) is selected.

The initial decision parameter is obtained by evaluating the circle function at the start position $(x_0, y_0) = (0, r)$

$$p_0 = F_{circle}\left(1, r - \frac{1}{2}\right) = 1 + \left(r - \frac{1}{2}\right)^2 - r^2 = \frac{5}{4} - r$$

if the radius r is specified as an integer, we can simply round p_0 as $p_0 = 1 - r$.

Successive decision parameters are obtained using incremental calculations. We obtain a recursive expression for the next decision parameter by evaluating the circle function at sampling position $x_{k+1} + 1 = x_k + 2$.

$$p_{k+1} = F_{circle}\left(x_{k+1} + 1, y_{k+1} - \frac{1}{2}\right) = [x_k + 1 + 1]^2 + \left(y_{k+1} - \frac{1}{2}\right)^2 - r^2$$

$$p_{k+1} = p_k + 2\,(x_k + 1) + \left(y_{k+1}^2 - y_k^2\right) + (y_{k+1} - y_k) + 1$$

where y_{k+1} is either y_k or y_{k-1}, depending on the sign of p_k

If p_k is negative (– ve)

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

$$p_{k+1} = p_k + 2(x_k + 1) + 1 = p_k + 2x_{k+1} + 1$$

If p_k is positive (+ ve)

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

$$\therefore \quad p_{k+1} = p_k + 2\,(x_k + 1) + 1 - 2y_{k+1}$$

Thus, increments for obtaining p_{k+1} are either $2x_{k+1} + 1$ (if p_k is – ve) or $2x_{k+1} + 1 - 2y_{k+1}$ (if p_k is + ve) and the evaluation of the terms $2x_{k+1}$ and $2y_{k+1}$ can be done as

$$2x_{k+1} = 2x_k + 2$$

$$2y_{k+1} = 2y_k - 2$$

At the start position $(0, r)$, these two terms have the values 0 and $2r$ respectively. Each successive value is obtained by adding 2 to the previous value of $2x$ and subtracting 2 from the previous value of $2y$.

The initial value of decision parameter can be obtained by evaluating the circle function at the start position $(x_0, y_0) = (0, r)$

$$p_0 = f_{circle}\left(1, r - \frac{1}{2}\right) = 1 + \left(r - \frac{1}{2}\right)^2 - r^2 = \frac{5}{4} - r = 1.25 - r$$

If radius r is an integer, we can simply round p_0 to $p_0 = 1 - r$.

> **Note:** As in Bresenham's line algorithm, the midpoint method calculates pixel positions along the circumference of a circle using integer additions and subtractions, assuming that the circle parameters are specified in integer screen co-ordinates.

Midpoint circle Algorithm

1. Input radius r and centre (x_c, y_c) and obtain the first point on the circumference of a circle centered on the origin as $(x_0, y_0) = (0, r)$ *i.e.*, initialize starting position as

$$x = 0$$

$$y = r$$

2. Calculate the initial value of the decision parameter as

$$p = 1.25 - r$$

3. do

```
{
    plot (x, y)
    if (p < 0)
        {
            x = x + 1
            y = y
            p = p + 2x + 1
        }
    else
        {
            x = x + 1
            y = y − 1
            p = p + 2x − 2y + 1
        }
}
while (x < y)
```

4. Determine symmetry points in the other seven octants.
5. Stop.

EXAMPLE 5.7.

Using midpoint circle algorithm plot a circle whose radius = 10 units.

Solution:

$$r = 10$$

The initial point $(x, y) = (0, 10)$

i.e.,

$$x = 0$$
$$y = 10$$

Calculate initial decision parameters $p = 1 - r = 1 - 10 = -9$

$$p = -9\ (< 0)$$

First **plot (0, 10)**

Here $p < 0$ so

$$x = 0 + 1 = 1$$

and

$$y = 10$$
$$p = -9 + 2 \times 1 + 1 = -9 + 3 = -6$$

Now $x < y$ *i.e.,* $1 < 10$. So condition is true and **plot (1, 10)**

so

$$p = -6\ (< 0)$$
$$x = 1 + 1 = 2$$

and

$$y = 10$$
$$p = -6 + 2 \times 2 + 1 = -6 + 5 = -1$$

Now $x < y$ *i.e.,* $2 < 10$, so **plot (2, 10)**

So

$$p = -1\ (< 0)$$
$$x = 2 + 1 = 3$$
$$y = 10$$
$$p = -1 + 2 \times 3 + 1 = 6$$

$x < y$ so plot (3, 10)

Now $p = 6(> 0)$

so

$$x = x + 1 = 3 + 1 = 4$$
$$y = y - 1 = 10 - 1 = 9$$
$$p = p + 2x - 2y + 1 = 6 + 8 - 18 + 1 = -3$$

$x < y$, so the next point to plot is (4, 9)

Now $p = -3\ (< 0)$

So

$$x = x + 1 = 4 + 1 = 5$$
$$y = y = 9$$
$$p = p + 2x + 1 = -3 + 10 + 1 = 8$$

$x < y$, So next point *plot is (5, 9).*

Now $p = 8\ (> 0)$

So

$$x = x + 1 = 5 + 1 = 6$$
$$y = y - 1 = 9 - 1 = 8$$
$$p = p + 2x - 2y + 1 = 8 + 12 - 16 + 1 = 5$$

$x < y$, So *plot (6, 8)*

Now $p = 5\ (> 0)$

So,

$$x = x + 1 = 6 + 1 = 7$$
$$y = y - 1 = 8 - 1 = 7$$
$$p = p + 2x - 2y + 1 = 5 + 14 - 14 + 1 = 6$$

Plot (7, 7), now stop as x equal to y.

Note: We have to continue the iteration until $x \geq y$ *i.e.*, either $x > y$ or $x = y$.

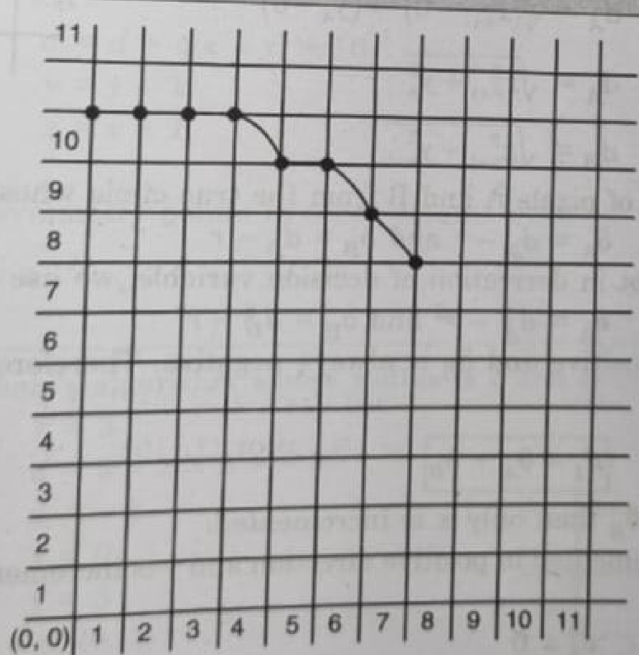


Fig. 5.16.

5.7.3. Bresenham's Circle Drawing Algorithm

Bresenham's method of drawing the circle is an efficient method because it avoids trigonometric and square root calculation by adopting only integer operation involving squares of the pixel separation distances.

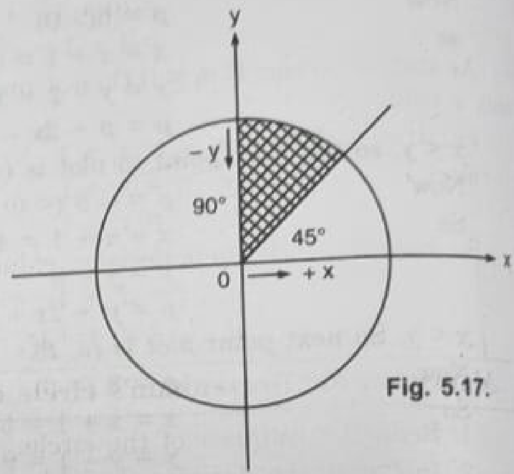


Fig. 5.17.

The Bresenham's circle drawing algorithm considers the eight way symmetry of the circle. It plots 1/8th part of the circle from 90° to 45°. As circle is drawn from 90° to 45°, the x-moves in + ve direction and y moves in the − ve direction.

To achieve best approximation to the true circle we have to select those pixels in the raster that falls the least distance from the true circle. Let us observe the 90° to 45° portion of the circle, each new point closest to the true circle can be found by applying either of two options :

(a) Increment in positive x direction by one unit or

(b) Increment in positive x direction and negative y direction both by one unit.

If P_n is a current point with co-ordinates (x_n, y_n) then the next point could be either A or B.

We have to select A or B, depending on which is close to the circle, and for that we have to perform some test.

The closer pixel amongst these two can be determined as follows.

The distances of pixels A and B from the origin (0, 0) are given by

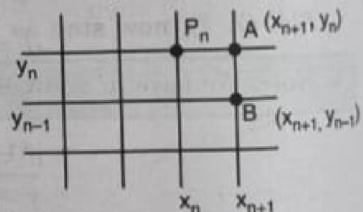


Fig. 5.18.

$$d_A = \sqrt{(x_{n+1} - 0)^2 + (y_n - 0)^2}$$

$$d_A = \sqrt{x_{n+1}^2 + y_n^2}$$

and
$$d_B = \sqrt{x_{n+1}^2 + y_{n-1}^2}$$

Now, the distances of pixels A and B from the true circle whose radius r are given as

$$\delta_A = d_A - r \text{ and } \delta_B = d_B - r$$

To avoid square root in derivation of decision variable, we use

$$\delta_A = d_A^2 - r^2 \text{ and } \delta_B = d_B^2 - r^2$$

But δ_A is always positive and δ_B is always negative. Therefore we can define decision variable d_i as

$$\boxed{d_i = \delta_A + \delta_B}$$

If $d_i < 0$ *i.e.*, $\delta_A < \delta_B$ then only x is incremented.

Otherwise x is incremented in positive direction and y is incremented in negative direction *i.e.*,

For $\quad d_i < 0$

$$x_{i+1} = x_i + 1$$

else

$$x_{i+1} = x_i + 1$$
$$y_{i+1} = y_i - 1$$

The equation for d_i at starting point *i.e.*, $x = 0$ and $y = r$ is as follows

$$d_i = \delta_A + \delta_B = x_{n+1}^2 + y_n^2 - r^2 + x_{n+1}^2 + y_{n-1}^2 - r^2$$

$$= (0 + 1)^2 + r^2 - r^2 + (0 + 1)^2 + (r - 1)^2 - r^2$$
$$= 1 + 1 + r^2 + 1 - 2r - r^2 = 3 - 2r$$

As stated earlier, if $d_i < 0$ then it implies A is closer to the true circle, hence increment only x value.

i.e., If $d_i < 0$ then $x_{i+1} = x_i + 1$

else $x_{i+1} = x_i + 1$

$y_{i+1} = y_i - 1$

So, recompute the new decision value, by substituting new value of x_i we get

$$d_{i+1} = d_i + 4x_i + 6 \qquad \text{if } d_i < 0$$
$$d_{i+1} = d_i + 4(x_i - y_i) + 10 \qquad \text{if } d_i \geq 0$$

Algorithm for Bresenham's circle drawing

1. Read the radius r of the circle.
2. Initialize the decision variable

   $d = 3 - 2 * r$
3. $x = 0$

   $y = r$
4. If we are using octant symmetry to plot the pixels then until $(x < y)$ we have to perform following steps.

   if $(d < 0)$ then

   $d = d + 4x + 6$

   $x = x + 1$

   else

   $d = d + 4(x - y) + 10$

   $y = y - 1$

   $x = x + 1$
5. Plot (x, y)
6. Determine the symmetry points in other octants also.
7. Stop.

EXAMPLE. 5.8.

Plot a circle by Bresenham's algorithm whose radius is 3 and center is (0, 0).

Solution:

$r = 3$

$d = 3 - 2 \times 3 = -3$

$d = -3$

Here $x = 0$

$y = 3$

Plot first point (x, y) i.e., $(0, 3)$

Now $d = -3\ (< 0)$ then

$d = -3 + 4 \times 0 + 6$

$= 3$

$x = x + 1$

$= 0 + 1 = 1$

$y = y = 3$

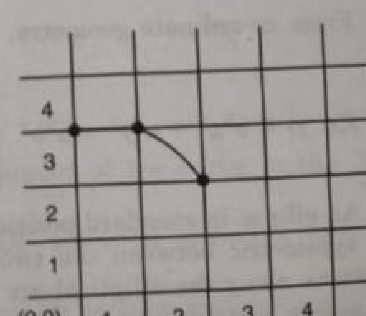


Fig. 5.19.

Plot (1, 3).

Now $d = 3 (> 0)$ then

$$\begin{aligned} d &= d + 4(x - y) + 10 \\ &= 3 + 4\,(1 - 3) + 10 \\ &= 3 - 8 + 10 = 5 \\ x &= x + 1 = 1 + 1 = 2 \\ y &= y - 1 = 3 - 1 = 2 \end{aligned}$$

Plot (2, 2)

After plotting these points, we will get the circle.

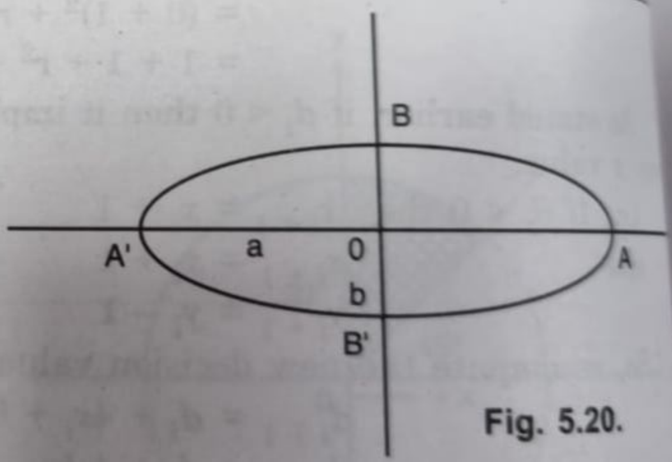


Fig. 5.20.

3. BASIC CONCEPTS IN ELLIPSE

The general equation for an ellipse with major axis $2a$ and minor axis $2b$, centered at the origin is

$$p2_{k+1} = p2_k - 2r_x^2 (y_k - 1) + r_x^2 + r_y^2 \left[\left(x_{k+1} + \frac{1}{2}\right)^2 - \left(x_k + \frac{1}{2}\right)\right]$$

where x_{k+1} set either to x_k or to x_{k+1} depending on the sign of $p2_k$.

In region 2, the initial value of the decision parameter can by obtained by eval ellipse function at the last position in the region 1.

$$p2_0 = f_{\text{ellipse}}\left(x_0 + \frac{1}{2}, y_0 - 1\right) = r_y^2 \left(x_0 + \frac{1}{2}\right)^2 + r_x^2 (y_0 - 1)^2$$

Midpoint Ellipse Algorithm

1. Read radii r_x and r_y
2. Initialise starting point as

$$x = 0$$
$$y = r_y$$

Plot (x, y)

3. Calculate the initial value of decision parameter in region 1 as

$$p_1 = r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^2$$

4. Initialize dx and dy as

$$dx = 2r_y^2 x$$
$$dy = 2r_x^2 y$$

5. do

{

plot (x, y)

if $(p1 < 0)$

{

$x = x + 1$

$y = y$

$dx = dx + 2r_y^2$

$p1 = p1 + dx + r_y^2$

}

else

{

$x = x + 1$

$y = y - 1$

```
            dx = dx + 2r_y^2
            dy = dy - 2r_x^2
            p1 = p1 + dx - dy + r_y^2
        }
    }
    while (dx < dy)
```

6. Calculate the initial value of decision parameter in region 2 as

$$p2 = r_y^2 \left(x + \frac{1}{2}\right)^2 + r_x^2 (y-1)^2 - r_x^2 r_y^2$$

7. do

```
    {
        plot (x, y)
        if (p2 > 0)
            {
                x = x
                y = y - 1
                dy = dy - 2r_x^2
                p2 = p2 - dy + r_x^2
            }
    else
        {
            x = x + 1
            y = y - 1
            dy = dy - 2r_x^2
            dx = dx + 2r_y^2
            p2 = p2 + dx - dy + r_x^2
        }
    }
    while (y > 0)
```

8. Determine symmetrical points in other three quadrants.
9. Stop.

5.10. PARALLEL LINE ALGORITHMS

With a parallel computer, we can calculate ...

2. The display file must be interpreted.
3. An explicit action is required to show the contents of frame buffer on some devices.

EXAMPLE 2.1.

Consider three different raster systems with resolutions of 640 × 480, 1280 × 1024 and 2560 × 2048. What size of frame buffer (in bytes) is needed for each of these systems to store 12 bits per pixel ? How much storage is required for each system if 24 bits per pixel are to be stored ? *(UPTU, MCA 2003)*

Solution: Case 1: For 640 × 480,

Total number of pixel required for 640 × 480 resolution = 640 × 480 pixels

So, size of frame-buffer required = 640 × 480 pixels

because 1 pixel can store 12 bits.

Therefore size of frame buffer (in bits) = 640 × 480 × 12 bit

$$= \frac{640 \times 480 \times 12}{8} \text{ bytes (1 byte = 8 bit)}$$

$$= 460800 \text{ bytes} = 450 \text{ KB (1024 bytes = 1 KB)}$$

For 1280 × 1024: Size of frame buffer (in bits)

$$= 1280 \times 1024 \times 12 \text{ bits} = \frac{1280 \times 1024 \times 12}{8} \text{ bytes}$$

$$= 1920 \text{ KB}$$

For 2560 × 2048:

$$\text{Size of frame-buffer} = \frac{2560 \times 2048 \times 12}{8} \text{ bytes} = 7.5 \text{ MB}$$

Case II: When one pixel can store 24 bits then frame-buffer size will be twice.

For 640 × 480, Frame-Buffer size

$$= \frac{640 \times 480 \times 24}{8} \text{ byte} = 900 \text{ KB}$$

For 1280 × 1024, Frame-buffer size

$$= 1920 \times 2 \text{ KB} = 3840 \text{ kB} = 3.75 \text{ MB}$$

For 2560 × 2048, Frame-buffer size

$$= 7.5 \times 2 \text{ MB} = 15 \text{ MB.}$$

EXAMPLE 2.2.

Consider a raster system with a resolution of 1024 × 1024. What is the size of the raster (in bytes) needed to storestire 4 bits per pixel ? How much storage is required if 8 bits per pixel are to be stored ? *(UPTU, B. Tech, 2002)*

Solution: Resolution = 1024 × 1024

Case-1: When 1 pixel = 4 bits

Size of raster = 1024 × 1024 × 4 bits

$$= \frac{1024 \times 1024 \times 4}{8} \text{ bytes} \qquad (\because 1 \text{ bytes} = 8 \text{ bits})$$

$$= 512 \times 1024 = 1 \times 2^{19} \text{ bytes}$$

Case-2: When 1 pixel = 8 bits

$$\text{Size of raster} = 1024 \times 1024 \times 8 \text{ bits}$$
$$= \frac{1024 \times 1024 \times 8}{8} \text{ bytes} = 1 \times 2^{20} \text{ bytes}$$

EXAMPLE 2.3.

Consider two raster systems with resolution of 640 by 480 and 1280 by 1024. How many pixels could be accessed per second in each of these systems by a display controller that refreshes the screen at a rate of 60 frames per second ? What is the access time per pixel in each system ?

Solution: Case-1. Resolution = 640×480

N. of pixels in one frame = $640 \times 480 = 307200$

Since controller can access 60 frames in one second

Therefore, total no. of pixels accessed = $60 \times 307200 = 18432000$ per sec.

Access time/pixel = 1/total pixels accessed per sec.

= 5.4×10^{-8} sec/pixel.

Case-2. When resolution = 1280×1024

Total no. of pixels accessed by the raster system

= $60 \times 1280 \times 1024 = 78643200$ per sec.

Access time per pixel = $1/78643200 = 1.2 \times 10^{-8}$ sec/pixel

EXAMPLE 2.4.

A laser printer is capable of printing two pages (size 9×11 inch) per second at resolution of 600 pixels per inch. How may be bits per second does such device require ?

(UPTU, MCA, 2002)

Solution: Storage required per page (in pixel)

= 9×11 inch/sec

= $9 \times 11 \times 600 \times 600$ pixel sec (∵ 1 inch = 600 pixels)

Since laser printer is capable of printing two pages per second bits per second required by the printer.

= $2 \times 9 \times 11 \times 600 \times 600$ pixels per second

= 7.128×10^7 pixel/sec

= 7.128×10^7 n bits/sec

(Assume 1 pixel = n bits)

EXAMPLE 2.5.

How many k bytes does a frame buffer need in a 600×400 pixel ?

Solution: Given resolution is 600×400

Suppose 1 pixel can store n bits then the size of frame buffer

= Resolution × bits per pixel

= $(600 \times 400) \times n$ bits

= $240000\, n$ bits

$= \frac{240000n}{1024 \times 8}$ k bytes

= $29.30\, n$ k bytes. (∵ 1kb = 1024 bits)

EXAMPLE 2.6.

Find out the aspect ratio of the raster system using 8 × 10 inches screen and 100 pixel/inch.

Solution: We know that,

$$\text{Aspect ratio} = \frac{\text{width}}{\text{height}} = \frac{8 \times 100}{10 \times 100} = \frac{4}{5}$$

$$\therefore \quad \text{Aspect ratio} = 4 : 5$$

EXAMPLE 2.7.

How much time is spent scanning across each row of pixels during screen refresh on a raster system with resolution of 1280 × 1024 and a refresh rate of 60 frames per second?

(U.P. Tech 2009-10)

Solution: Here, resolution, 1280 × 1024

That means system contains 1024 scan lines and each scan line contains of 1280 pixels

and refresh rate = 60 frames/sec.

that means 1 frame takes $\frac{1}{60}$ sec

Since, resolution = 1280 × 1024

∴ 1 frame consists of 1024 scan lines.

∴ 1024 scan lines takes to $\frac{1}{60}$ sec

∴ 1 scan lines take $\frac{1}{60 \times 1024}$ sec = 0.058 sec or 58 m sec.

EXAMPLE 2.8.

Suppose RGB raster system is to be designed using on 8 inch × 10 inch screen with a resolution of 100 pixels per inch in each direction. If we want to store 6 bits per pixel in the frame buffer, how much storage (in bytes) do we need for frame buffer?

Solution: Here, resolution = 8 inch × 10 inch

First we convert it in pixel, then

now resolution = 8 × 100 by 10 × 100 pixel = 800 × 1000 pixel

1 pixel can store 6 bits

so, frame buffer size required = 800 × 1000 × 6 bits

$$= \frac{800 \times 1000 \times 6}{8} \text{ bytes} = 6 \times 10^5 \text{ bytes.}$$

EXAMPLE 2.9.

How long would it take to load a 640 × 480 frame buffer with 12 bits per pixel, if 10^5 bits can be transferred per second? How long would it take to load a 24 bits per pixel frame buffer with a resolutions of 1280 by 1024 using this same transfer rate?

Solution: Frame buffer size = 640 × 480 pixels = 640 × 480 × 12 bits [1 pixel = 12 bits]

Since 10^5 bits takes 1 second

So, 640 × 480 × 12 bits takes $\frac{640 \times 480 \times 12}{10^5}$ = 36.864 sec.

EXAMPLE 2.10.

Suppose we have a computer with 32 bits per word and a transfer rate of 1 mips (million instructions per second). How long would it take to fill the frame buffer of a 300 dpi (dot per inch) laser printer with a page size of $8\frac{1}{2}$ inch by 11 inches?

Solution: Frame buffer size $= 8\frac{1}{2}\text{ inch} \times 11\text{ inch} = 8\frac{1}{2} \times 11\,(\text{inches})^2$

$$= \frac{17}{2} \times 11 \times (300)^2 \qquad [1\text{ inch} = 300\text{ dpi}]$$

$= 8415000$ dotes.

Suppose 1 dot = n bits

so, frame buffer size = 8415000 n bits.

Transfer rate = 1 mip = 1×10^6 word per second

$= 32 \times 10^6$ bits per second [1 word = 32 bits]

$$\text{Time required to fill the frame buffer} = \frac{841500\,n}{32 \times 10^6} = 0.263\,n\text{ sec.}$$

if $n = 4$ then time taken $= 0.263 \times 4 = 1.052$ sec.

EXAMPLE 2.11.

Consider a non-interlaced raster monitor with a resolution of $n \times m$, a refresh rate of r frames per second, a horizontal ratrace time of t_h and a vertical retrace time of t_v. What is the fraction of the totalm refresh time per frame spent in retrace of the electron beam?

Solution: Since resolution is $n \times m$. So, raster system contains m scan lines and system takes $(m - 1)$ horizontal retraces.

Total time taken for horizontal retrace $= (m - 1) \times$ horizontal retrace time

$= (m - 1) \times t_h$

If there are are r frames then total time $= r \times (m - 1) \times t_h$.

Since, there are r frames then there will be $(r - 1)$ vertical retraces and total time for vertical retrace $= (r - 1) \times$ vertical retrace time $= (r - 1) \times t_v$

$\therefore$ Total time for retrace $= r(m - 1)\,t_h + (r - 1) \times t_v$

$$\text{Total time taken for one frame} = \frac{r(m-1)\,t_h + (r-1)\,t_v}{r} \qquad [\because \text{for } r \text{ frames}]$$

$$\text{Total refresh time for one frame} = \frac{1}{r}$$

The fraction of total refresh time per frame spent in retrace of the electron beam

$$= \frac{\left[\dfrac{r(m-1)t_h + (r-1)\,t_v}{r}\right]}{\left(\dfrac{1}{r}\right)} = r(m-1)t_h + (r-1)\,t_v$$

EXAMPLE 2.12.

How much memory is needed for the frame buffer to store a 640 × 400 display 16 gray levels?

(UPTU 2005-0