Transformation

Changing Position, shape, size, or orientation of an object on display is known as transformation.

Basic Transformation

- Basic transformation includes three transformations **Translation**, **Rotation**, and **Scaling**.
- These three transformations are known as basic transformation because with combination of these three transformations we can obtain any transformation.

Translation

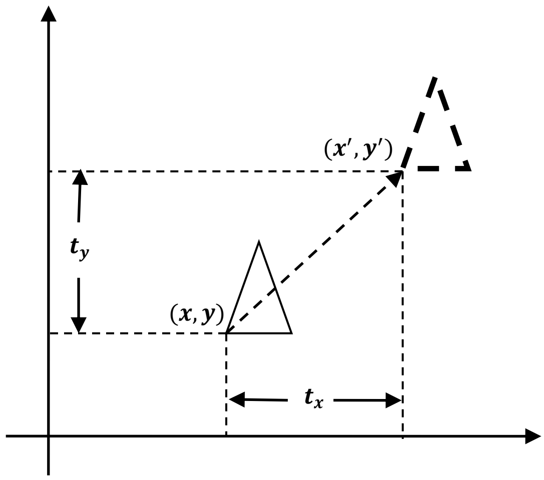


Fig. 3.1: - Translation.

- It is a transformation that used to reposition the object along the straight line path from one coordinate location to another.
- It is rigid body transformation so we need to translate whole object.
- We translate two dimensional point by adding translation distance t_x and t_y to the original coordinate position (x, y) to move at new position (x', y') as:
  $x' = x + t_x$ & $y' = y + t_y$
- Translation distance pair (t_x, t_y) is called a **Translation Vector** or **Shift Vector**.
- We can represent it into single matrix equation in column vector as;
  $P' = P + T$
  $$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$
- We can also represent it in row vector form as:
  $P' = P + T$
  $$[x' \quad y'] = [x \quad y] + [t_x \quad t_y]$$
- Since column vector representation is standard mathematical notation and since many graphics package like **GKS** and **PHIGS** uses column vector we will also follow column vector representation.
- **Example**: - Translate the triangle [A (10, 10), B (15, 15), C (20, 10)] 2 unit in x direction and 1 unit in y direction.
  We know that
  $P' = P + T$
  $$P' = [P] + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

For point (10, 10)
$A' = \begin{bmatrix}10\\10\end{bmatrix} + \begin{bmatrix}2\\1\end{bmatrix}$
$A' = \begin{bmatrix}12\\11\end{bmatrix}$
For point (15, 15)
$B' = \begin{bmatrix}15\\15\end{bmatrix} + \begin{bmatrix}2\\1\end{bmatrix}$
$B' = \begin{bmatrix}17\\16\end{bmatrix}$
For point (10, 10)
$C' = \begin{bmatrix}20\\10\end{bmatrix} + \begin{bmatrix}2\\1\end{bmatrix}$
$C' = \begin{bmatrix}22\\11\end{bmatrix}$

- Final coordinates after translation are [A′ (12, 11), B′ (17, 16), C′ (22, 11)].

Rotation

- It is a transformation that used to reposition the object along the circular path in the XY - plane.
- To generate a rotation we specify a rotation angle $\boldsymbol{\theta}$ and the position of the **Rotation Point** (**Pivot Point**) $(\boldsymbol{x_r}, \boldsymbol{y_r})$ about which the object is to be rotated.
- Positive value of rotation angle defines counter clockwise rotation and negative value of rotation angle defines clockwise rotation.
- We first find the equation of rotation when pivot point is at coordinate origin$(\mathbf{0}, \mathbf{0})$.

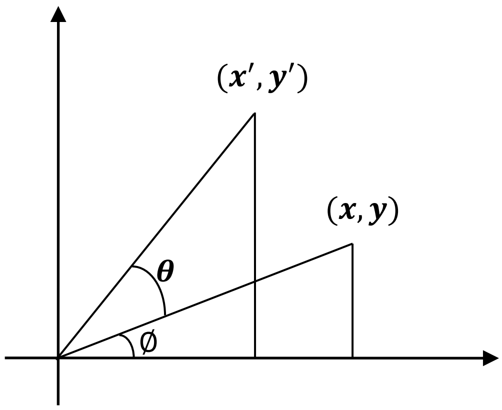


Fig. 3.2: - Rotation.

- From figure we can write.
$\boldsymbol{x = r\cos\emptyset}$
$\boldsymbol{y = r\sin\emptyset}$
and
$\boldsymbol{x' = r\cos(\theta + \emptyset) = r\cos\emptyset\cos\theta - r\sin\emptyset\sin\theta}$
$\boldsymbol{y' = r\sin(\emptyset + \theta) = r\cos\emptyset\sin\theta + r\sin\emptyset\cos\theta}$
- Now replace $\boldsymbol{r\cos\emptyset}$ with $\boldsymbol{x}$ and $\boldsymbol{r\sin\emptyset}$ with $\boldsymbol{y}$ in above equation.
$\boldsymbol{x' = x\cos\theta - y\sin\theta}$
$\boldsymbol{y' = x\sin\theta + y\cos\theta}$
- We can write it in the form of column vector matrix equation as;
$\boldsymbol{P' = R \cdot P}$

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

- Rotation about arbitrary point is illustrated in below figure.

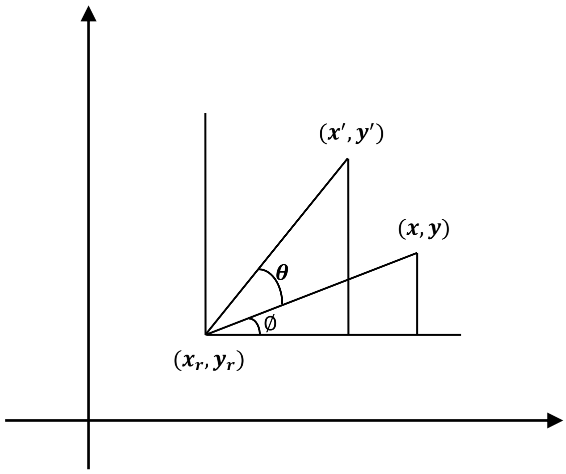


Fig. 3.3: - Rotation about pivot point.

- Transformation equation for rotation of a point about pivot point (x_r, y_r) is:
  $$x' = x_r + (x - x_r)\cos\theta - (y - y_r)\sin\theta$$
  $$y' = y_r + (x - x_r)\sin\theta + (y - y_r)\cos\theta$$
- These equations are differing from rotation about origin and its matrix representation is also different.
- Its matrix equation can be obtained by simple method that we will discuss later in this chapter.
- Rotation is also rigid body transformation so we need to rotate each point of object.
- **Example**: - Locate the new position of the triangle [A (5, 4), B (8, 3), C (8, 8)] after its rotation by 90° clockwise about the origin.
  As rotation is clockwise we will take $\theta = -90^\circ$.
  $$P' = R \cdot P$$
  $$P' = \begin{bmatrix} \cos(-90) & -\sin(-90) \\ \sin(-90) & \cos(-90) \end{bmatrix} \begin{bmatrix} 5 & 8 & 8 \\ 4 & 3 & 8 \end{bmatrix}$$
  $$P' = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 5 & 8 & 8 \\ 4 & 3 & 8 \end{bmatrix}$$
  $$P' = \begin{bmatrix} 4 & 3 & 8 \\ -5 & -8 & -8 \end{bmatrix}$$
- Final coordinates after rotation are [A′ (4, -5), B′ (3, -8), C′ (8, -8)].

Scaling

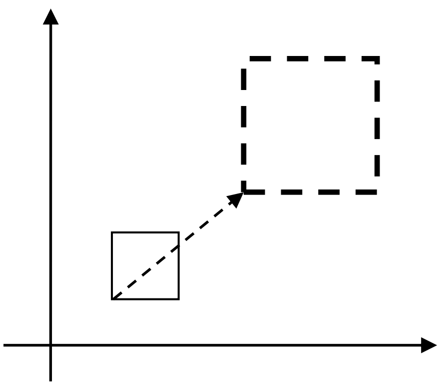


Fig. 3.4: - Scaling.

- It is a transformation that used to alter the size of an object.
- This operation is carried out by multiplying coordinate value (x, y) with scaling factor (s_x, s_y) respectively.
- So equation for scaling is given by:
  $x' = x \cdot s_x$
  $y' = y \cdot s_y$
- These equation can be represented in column vector matrix equation as:
  $P' = S \cdot P$
  $$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$
- Any positive value can be assigned to(s_x, s_y)**.**
- Values less than 1 reduce the size while values greater than 1 enlarge the size of object, and object remains unchanged when values of both factor is 1.
- Same values of s_x and s_y will produce **Uniform Scaling**. And different values of s_x and s_y will produce **Differential Scaling**.
- Objects transformed with above equation are both scale and repositioned.
- Scaling factor with value less than 1 will move object closer to origin, while scaling factor with value greater than 1 will move object away from origin.
- We can control the position of object after scaling by keeping one position fixed called **Fix point** (x_f, y_f) that point will remain unchanged after the scaling transformation.

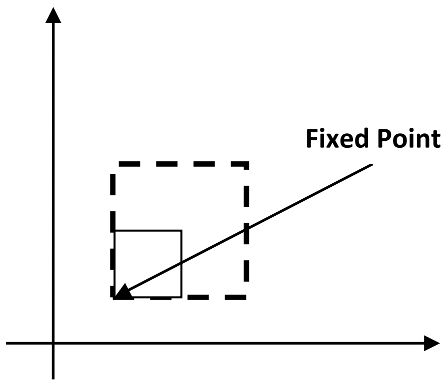


Fig. 3.5: - Fixed point scaling.

- Equation for scaling with fixed point position as (x_f, y_f) is:
  $x' = x_f + (x - x_f)s_x$ $\qquad$ $y' = y_f + (y - y_f)s_y$
  $x' = x_f + xs_x - x_f s_x$ $\qquad$ $y' = y_f + ys_y - y_f s_y$
  $x' = xs_x + x_f(1 - s_x)$ $\qquad$ $y' = ys_y + y_f(1 - s_y)$
- Matrix equation for the same will discuss in later section.
- Polygons are scaled by applying scaling at coordinates and redrawing while other body like circle and ellipse will scale using its defining parameters. For example ellipse will scale using its semi major axis, semi minor axis and center point scaling and redrawing at that position.
- **Example**: - Consider square with left-bottom corner at (2, 2) and right-top corner at (6, 6) apply the transformation which makes its size half.
  As we want size half so value of scale factor are $s_x = 0.5, s_y = 0.5$ and Coordinates of square are [A (2, 2), B (6, 2), C (6, 6), D (2, 6)].
  $P' = S \cdot P$

$P' = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} 2 & 6 & 6 & 2 \\ 2 & 2 & 6 & 6 \end{bmatrix}$

$P' = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} 2 & 6 & 6 & 2 \\ 2 & 2 & 6 & 6 \end{bmatrix}$

$P' = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 1 & 1 & 3 & 3 \end{bmatrix}$

- Final coordinate after scaling are [A′ (1, 1), B′ (3, 1), C′ (3, 3), D′ (1, 3)].

Matrix Representation and homogeneous coordinates

- Many graphics application involves sequence of geometric transformations.
- For example in design and picture construction application we perform Translation, Rotation, and scaling to fit the picture components into their proper positions.
- For efficient processing we will reformulate transformation sequences.
- We have matrix representation of basic transformation and we can express it in the general matrix form as:
  $\boldsymbol{P' = M_1 \cdot P + M_2}$
  Where $\boldsymbol{P}$ and $\boldsymbol{P'}$ are initial and final point position, $\boldsymbol{M_1}$ contains rotation and scaling terms and $\boldsymbol{M_2}$ contains translation al terms associated with pivot point, fixed point and reposition.
- For efficient utilization we must calculate all sequence of transformation in one step and for that reason we reformulate above equation to eliminate the matrix addition associated with translation terms in matrix $\boldsymbol{M_2}$.
- We can combine that thing by expanding 2X2 matrix representation into 3X3 matrices.
- It will allows us to convert all transformation into matrix multiplication but we need to represent vertex position $(\boldsymbol{x}, \boldsymbol{y})$ with homogeneous coordinate triple $(\boldsymbol{x_h}, \boldsymbol{y_h}, \boldsymbol{h})$ Where $\boldsymbol{x = \frac{x_h}{h}}$, $\boldsymbol{y = \frac{y_h}{h}}$ thus we can also write triple as $(\boldsymbol{h \cdot x}, \boldsymbol{h \cdot y}, \boldsymbol{h})$.
- For two dimensional geometric transformation we can take value of $\boldsymbol{h}$ is any positive number so we can get infinite homogeneous representation for coordinate value $(\boldsymbol{x}, \boldsymbol{y})$.
- But convenient choice is set $\boldsymbol{h = 1}$ as it is multiplicative identity, than $(\boldsymbol{x}, \boldsymbol{y})$ is represented as $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{1})$.
- Expressing coordinates in homogeneous coordinates form allows us to represent all geometric transformation equations as matrix multiplication.
- Let's see each representation with $\boldsymbol{h = 1}$

Translation

$$\boldsymbol{P' = T_{(t_x, t_y)} \cdot P}$$

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

NOTE: - Inverse of translation matrix is obtain by putting $-\boldsymbol{t_x}$ & $-\boldsymbol{t_y}$ instead of $\boldsymbol{t_x}$ & $\boldsymbol{t_y}$.

Rotation

$$\boldsymbol{P' = R_{(\theta)} \cdot P}$$

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

NOTE: - Inverse of rotation matrix is obtained by replacing $\boldsymbol{\theta}$ by $-\boldsymbol{\theta}$.

Scaling

$$\boldsymbol{P' = S_{(s_x, s_y)} \cdot P}$$

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

NOTE: - Inverse of scaling matrix is obtained by replacing s_x & s_y by $\frac{1}{s_x}$ & $\frac{1}{s_y}$ respectively.

Composite Transformation

- We can set up a matrix for any sequence of transformations as a **composite transformation matrix** by calculating the matrix product of individual transformation.
- For column matrix representation of coordinate positions, we form composite transformations by multiplying matrices in order from right to left.

Translations

- Two successive translations are performed as:

$$P' = T(t_{x2}, t_{y2}) \cdot \{T(t_{x1}, t_{y1}) \cdot P\}$$
$$P' = \{T(t_{x2}, t_{y2}) \cdot T(t_{x1}, t_{y1})\} \cdot P$$
$$P' = \begin{bmatrix} 1 & 0 & t_{x2} \\ 0 & 1 & t_{y2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_{x1} \\ 0 & 1 & t_{y1} \\ 0 & 0 & 1 \end{bmatrix} \cdot P$$
$$P' = \begin{bmatrix} 1 & 0 & t_{x1} + t_{x2} \\ 0 & 1 & t_{y1} + t_{y2} \\ 0 & 0 & 1 \end{bmatrix} \cdot P$$
$$P' = T(t_{x1} + t_{x2}, t_{y1} + t_{y2}) \cdot P\}$$

Here P' and P are column vector of final and initial point coordinate respectively.

- This concept can be extended for any number of successive translations.

Example: Obtain the final coordinates after two translations on point $p(2,3)$ with translation vector $(4,\ 3)$ and $(-1,\ 2)$ respectively.

$$P' = T(t_{x1} + t_{x2}, t_{y1} + t_{y2}) \cdot P$$

$$P' = \begin{bmatrix} 1 & 0 & t_{x1} + t_{x2} \\ 0 & 1 & t_{y1} + t_{y2} \\ 0 & 0 & 1 \end{bmatrix} \cdot P = \begin{bmatrix} 1 & 0 & 4 + (-1) \\ 0 & 1 & 3 + 2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

$$P' = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \\ 1 \end{bmatrix}$$

Final Coordinates after translations are $p'(5,\ 8)$.

Rotations

- Two successive Rotations are performed as:

$$P' = R(\theta_2) \cdot \{R(\theta_1) \cdot P\}$$
$$P' = \{R(\theta_2) \cdot R(\theta_1)\} \cdot P$$
$$P' = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot P$$
$$P' = \begin{bmatrix} \cos\theta_2 \cos\theta_1 - \sin\theta_2 \sin\theta_1 & -\sin\theta_1 \cos\theta_2 - \sin\theta_2 \cos\theta_1 & 0 \\ \sin\theta_1 \cos\theta_2 + \sin\theta_2 \cos\theta_1 & \cos\theta_2 \cos\theta_1 - \sin\theta_2 \sin\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot P$$

$$\boldsymbol{P'} = \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \boldsymbol{P}$$

$$\boldsymbol{P'} = \boldsymbol{R}(\theta_1+\theta_2) \cdot \boldsymbol{P}$$

Here $\boldsymbol{P'}$ and $\boldsymbol{P}$ are column vector of final and initial point coordinate respectively.

- This concept can be extended for any number of successive rotations.

Example: Obtain the final coordinates after two rotations on point $p(6,9)$ with rotation angles are 30^o and 60^o respectively.

$$P' = R(\theta_1+\theta_2) \cdot P$$

$$P' = \begin{bmatrix} cos(\theta_1+\theta_2) & -sin(\theta_1+\theta_2) & 0 \\ sin(\theta_1+\theta_2) & cos(\theta_1+\theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot P$$

$$P' = \begin{bmatrix} cos(30+60) & -sin(30+60) & 0 \\ sin(30+60) & cos(30+60) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot P$$

$$P' = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 6 \\ 9 \\ 1 \end{bmatrix} = \begin{bmatrix} -9 \\ 6 \\ 1 \end{bmatrix}$$

Final Coordinates after rotations are $p'(-9,\ 6)$.

Scaling

- Two successive scaling are performed as:

$$\boldsymbol{P'} = \boldsymbol{S}(s_{x2}, s_{y2}) \cdot \{\boldsymbol{S}(s_{x1}, s_{y1}) \cdot \boldsymbol{P}\}$$

$$\boldsymbol{P'} = \{\boldsymbol{S}(s_{x2}, s_{y2}) \cdot \boldsymbol{S}(s_{x1}, s_{y1})\} \cdot \boldsymbol{P}$$

$$\boldsymbol{P'} = \begin{bmatrix} s_{x2} & 0 & 0 \\ 0 & s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_{x1} & 0 & 0 \\ 0 & s_{y1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \boldsymbol{P}$$

$$\boldsymbol{P'} = \begin{bmatrix} s_{x1} \cdot s_{x2} & 0 & 0 \\ 0 & s_{y1} \cdot s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \boldsymbol{P}$$

$$\boldsymbol{P'} = \boldsymbol{S}(s_{x1} \cdot s_{x2}, s_{y1} \cdot s_{y2}) \cdot \boldsymbol{P}$$

Here $\boldsymbol{P'}$ and $\boldsymbol{P}$ are column vector of final and initial point coordinate respectively.

- This concept can be extended for any number of successive scaling.

Example: Obtain the final coordinates after two scaling on line pq $[p(2,2),\ q(8,\ 8)]$ with scaling factors are $(2,\ 2)$ and $(3,\ 3)$ respectively.

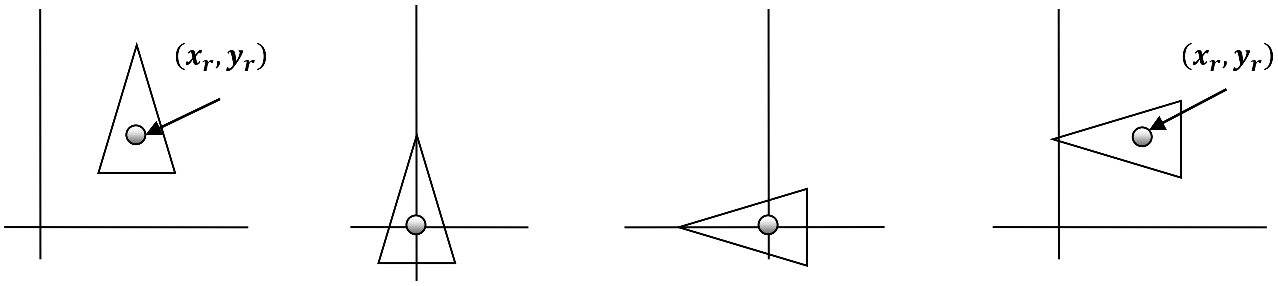
$$P' = S(s_{x1} \cdot s_{x2}, s_{y1} \cdot s_{y2}) \cdot P$$

$$P' = \begin{bmatrix} s_{x1} \cdot s_{x2} & 0 & 0 \\ 0 & s_{y1} \cdot s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot P = \begin{bmatrix} 2 \cdot 3 & 0 & 0 \\ 0 & 2 \cdot 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot P$$

$$P' = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 8 \\ 2 & 8 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 12 & 48 \\ 12 & 48 \\ 1 & 1 \end{bmatrix}$$

Final Coordinates after rotations are $p'(12,\ 12)$ and $q'(48,\ 48)$.

General Pivot-Point Rotation



(a) Original Position of Object and Pivot Point. (b) Translation of Object so that Pivot Point (x_r, y_r) is at Origin. (c) Rotation about Origin. (d) Translation of Object so that Pivot Point is Return to Position (x_r, y_r) .

Fig. 3.6: - General pivot point rotation.

- For rotating object about arbitrary point called pivot point we need to apply following sequence of transformation.
  1. Translate the object so that the pivot-point coincides with the coordinate origin.
  2. Rotate the object about the coordinate origin with specified angle.
  3. Translate the object so that the pivot-point is returned to its original position (i.e. Inverse of step-1).
- Let's find matrix equation for this

$$\boldsymbol{P'} = \boldsymbol{T}(\boldsymbol{x_r}, \boldsymbol{y_r}) \cdot [\boldsymbol{R}(\boldsymbol{\theta}) \cdot \{\boldsymbol{T}(-\boldsymbol{x_r}, -\boldsymbol{y_r}) \cdot \boldsymbol{P}\}]$$

$$\boldsymbol{P'} = \{\boldsymbol{T}(\boldsymbol{x_r}, \boldsymbol{y_r}) \cdot \boldsymbol{R}(\boldsymbol{\theta}) \cdot \boldsymbol{T}(-\boldsymbol{x_r}, -\boldsymbol{y_r})\} \cdot \boldsymbol{P}$$

$$\boldsymbol{P'} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \boldsymbol{x_r} \\ \mathbf{0} & \mathbf{1} & \boldsymbol{y_r} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{cos}\,\boldsymbol{\theta} & -\mathbf{sin}\,\boldsymbol{\theta} & \mathbf{0} \\ \mathbf{sin}\,\boldsymbol{\theta} & \mathbf{cos}\,\boldsymbol{\theta} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} & -\boldsymbol{x_r} \\ \mathbf{0} & \mathbf{1} & -\boldsymbol{y_r} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \boldsymbol{P}$$

$$\boldsymbol{P'} = \begin{bmatrix} \mathbf{cos}\,\boldsymbol{\theta} & -\mathbf{sin}\,\boldsymbol{\theta} & \boldsymbol{x_r}(\mathbf{1} - \mathbf{cos}\,\boldsymbol{\theta}) + \boldsymbol{y_r}\,\mathbf{sin}\,\boldsymbol{\theta} \\ \mathbf{sin}\,\boldsymbol{\theta} & \mathbf{cos}\,\boldsymbol{\theta} & \boldsymbol{y_r}(\mathbf{1} - \mathbf{cos}\,\boldsymbol{\theta}) - \boldsymbol{x_r}\,\mathbf{sin}\,\boldsymbol{\theta} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \boldsymbol{P}$$

$$\boldsymbol{P'} = \boldsymbol{R}(\boldsymbol{x_r}, \boldsymbol{y_r}\boldsymbol{\theta}) \cdot \boldsymbol{P}$$

Here $\boldsymbol{P'}$ and $\boldsymbol{P}$ are column vector of final and initial point coordinate respectively and $(\boldsymbol{x_r}, \boldsymbol{y_r})$ are the coordinates of pivot-point.

- **Example**: - Locate the new position of the triangle [A (5, 4), B (8, 3), C (8, 8)] after its rotation by 90° clockwise about the centroid.

Pivot point is centroid of the triangle so:

$$x_r = \frac{5+8+8}{3} = 7\,, \qquad y_r = \frac{4+3+8}{3} = 5$$

As rotation is clockwise we will take $\theta = -90^\circ$.

$$P' = R_{(x_r, y_r, \theta)} \cdot P$$

$$P' = \begin{bmatrix} \cos\theta & -\sin\theta & x_r(1-\cos\theta) + y_r\sin\theta \\ \sin\theta & \cos\theta & y_r(1-\cos\theta) - x_r\sin\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 & 8 & 8 \\ 4 & 3 & 8 \\ 1 & 1 & 1 \end{bmatrix}$$

$$P' = \begin{bmatrix} \cos(-90) & -\sin(-90) & 7(1-\cos(-90)) + 5\sin(-90) \\ \sin(-90) & \cos(-90) & 5(1-\cos(-90)) - 7\sin(-90) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 & 8 & 8 \\ 4 & 3 & 8 \\ 1 & 1 & 1 \end{bmatrix}$$

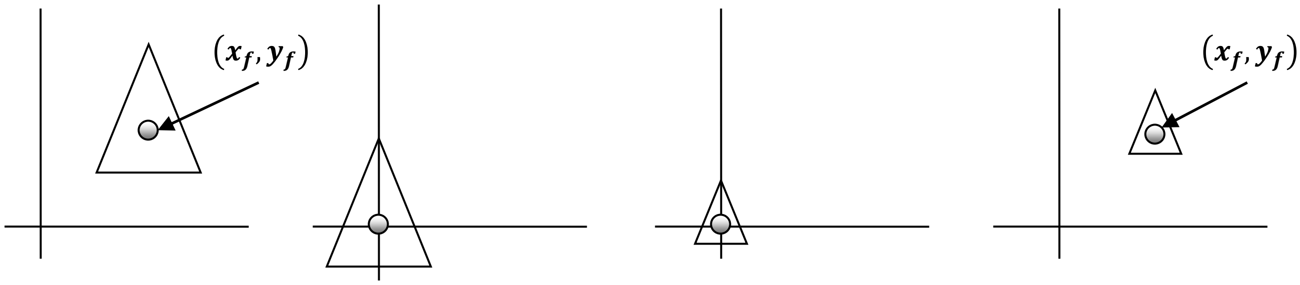
$$P' = \begin{bmatrix} 0 & 1 & 7(1-0) - 5(1) \\ -1 & 0 & 5(1-0) + 7(1) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 & 8 & 8 \\ 4 & 3 & 8 \\ 1 & 1 & 1 \end{bmatrix}$$

$$P' = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 12 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 & 8 & 8 \\ 4 & 3 & 8 \\ 1 & 1 & 1 \end{bmatrix}$$

$$P' = \begin{bmatrix} 11 & 13 & 18 \\ 7 & 4 & 4 \\ 1 & 1 & 1 \end{bmatrix}$$

- Final coordinates after rotation are [A′ (11, 7), B′ (13, 4), C′ (18, 4)].

General Fixed-Point Scaling



(a) Original Position of Object and Fixed Point

(b) Translate Object so that Fixed Point (x_f, y_f) is at Origin

(c) Scale Object with Respect to Origin

(d) Translate Object so that Fixed Point is Return to Position (x_f, y_f) .

Fig. 3.7: - General fixed point scaling.

- For scaling object with position of one point called fixed point will remains same, we need to apply following sequence of transformation.
  1. Translate the object so that the fixed-point coincides with the coordinate origin.
  2. Scale the object with respect to the coordinate origin with specified scale factors.
  3. Translate the object so that the fixed-point is returned to its original position (i.e. Inverse of step-1).
- Let's find matrix equation for this

$$\boldsymbol{P' = T(x_f, y_f) \cdot [S(s_x, s_y) \cdot \{T(-x_f, -y_f) \cdot P\}]}$$
$$\boldsymbol{P' = \{T(x_f, y_f) \cdot S(s_x, s_y) \cdot T(-x_f, -y_f)\} \cdot P}$$
$$\boldsymbol{P' = \begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix} \cdot P}$$
$$\boldsymbol{P' = \begin{bmatrix} s_x & 0 & x_f(1-s_x) \\ 0 & s_y & y_f(1-s_y) \\ 0 & 0 & 1 \end{bmatrix} \cdot P}$$
$$\boldsymbol{P' = S(x_f, y_f, s_x, s_y) \cdot P}$$

Here $\boldsymbol{P'}$ and $\boldsymbol{P}$ are column vector of final and initial point coordinate respectively and $(\boldsymbol{x_f}, \boldsymbol{y_f})$ are the coordinates of fixed-point.

- **Example**: - Consider square with left-bottom corner at (2, 2) and right-top corner at (6, 6) apply the transformation which makes its size half such that its center remains same.

Fixed point is center of square so:

$$x_f = 2 + \frac{6-2}{2}, \quad y_f = 2 + \frac{6-2}{2}$$

As we want size half so value of scale factor are $s_x = 0.5, s_y = 0.5$ and Coordinates of square are [A (2, 2), B (6, 2), C (6, 6), D (2, 6)].

$$P' = S(x_f, y_f, s_x, s_y) \cdot P$$
$$P' = \begin{bmatrix} s_x & 0 & x_f(1-s_x) \\ 0 & s_y & y_f(1-s_y) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 6 & 6 & 2 \\ 2 & 2 & 6 & 6 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$P' = \begin{bmatrix} 0.5 & 0 & 4(1-0.5) \\ 0 & 0.5 & 4(1-0.5) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 6 & 6 & 2 \\ 2 & 2 & 6 & 6 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$
$$P' = \begin{bmatrix} 0.5 & 0 & 2 \\ 0 & 0.5 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 6 & 6 & 2 \\ 2 & 2 & 6 & 6 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$
$$P' = \begin{bmatrix} 3 & 5 & 5 & 3 \\ 3 & 3 & 5 & 5 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

- Final coordinate after scaling are [A′ (3, 3), B′ (5, 3), C′ (5, 5), D′ (3, 5)]

General Scaling Directions

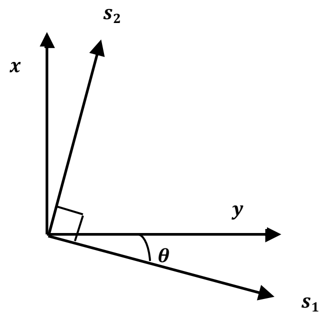


Fig. 3.8: - General scaling direction.

- Parameter s_x and s_y scale the object along x and y directions. We can scale an object in other directions by rotating the object to align the desired scaling directions with the coordinate axes before applying the scaling transformation.
- Suppose we apply scaling factor s_1 and s_2 in direction shown in figure than we will apply following transformations.
  1. Perform a rotation so that the direction for s_1 and s_2 coincide with x and y axes.
  2. Scale the object with specified scale factors.
  3. Perform opposite rotation to return points to their original orientations. (i.e. Inverse of step-1).
- Let's find matrix equation for this

$$\boldsymbol{P' = R^{-1}(\theta) \cdot [S(s_1, s_2) \cdot \{R(\theta) \cdot P\}]}$$
$$\boldsymbol{P' = \{R^{-1}(\theta) \cdot S(s_1, s_2) \cdot R(\theta)\} \cdot P}$$
$$\boldsymbol{P'} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \boldsymbol{P}$$
$$\boldsymbol{P'} = \begin{bmatrix} s_1\cos^2\theta + s_2\sin^2\theta & (s_2 - s_1)\cos\theta\sin\theta & 0 \\ (s_2 - s_1)\cos\theta\sin\theta & s_1\sin^2\theta + s_2\cos^2\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \boldsymbol{P}$$

Here P' and P are column vector of final and initial point coordinate respectively and θ is the angle between actual scaling direction and our standard coordinate axes.

Other Transformation

- Some package provides few additional transformations which are useful in certain applications. Two such transformations are reflection and shear.

Reflection

- A reflection is a transformation that produces a mirror image of an object.

- The mirror image for a two –dimensional reflection is generated relative to an **axis of reflection** by rotating the object 180^o about the reflection axis.
- Reflection gives image based on position of axis of reflection. Transformation matrix for few positions are discussed here.

Transformation matrix for reflection about the line $y = 0$, $the\ x\ axis$.

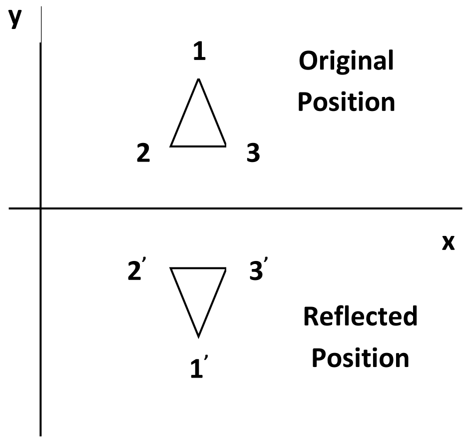


Fig. 3.9: - Reflection about x - axis.

- This transformation keeps x values are same, but flips (Change the sign) y values of coordinate positions.

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

Transformation matrix for reflection about the line $x = 0$, $the\ y\ axis$.

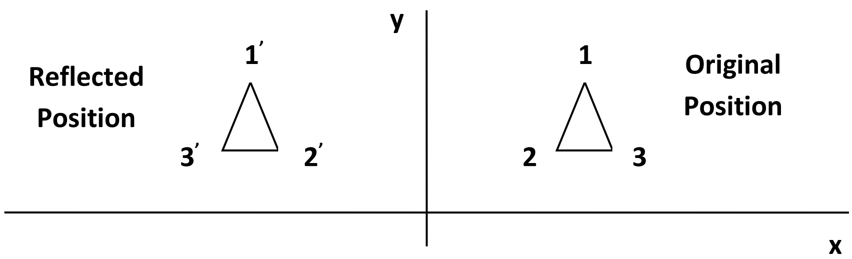


Fig. 3.10: - Reflection about y - axis.

- This transformation keeps y values are same, but flips (Change the sign) x values of coordinate positions.

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

Transformation matrix for reflection about the $Origin$.

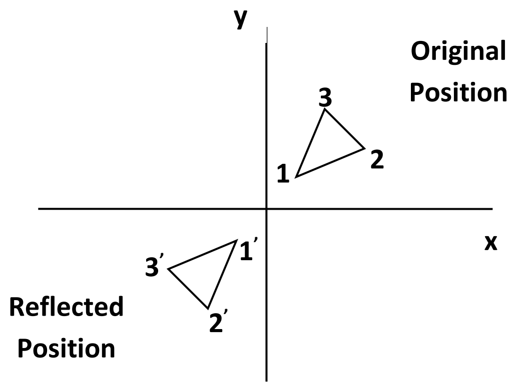


Fig. 3.11: - Reflection about origin.

- This transformation flips (Change the sign) x and y both values of coordinate positions.

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

Transformation matrix for reflection about the line $\boldsymbol{x = y}$.

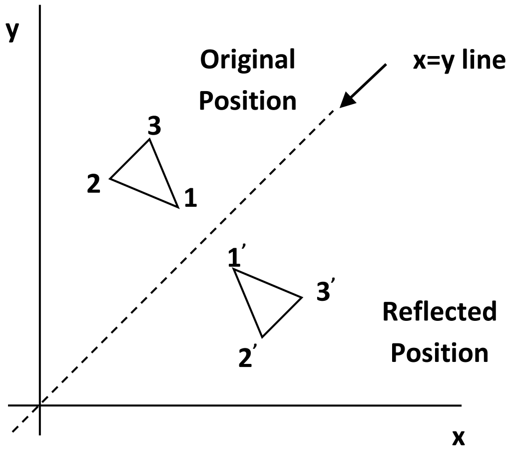


Fig. 3.12: - Reflection about x=y line.

- This transformation interchange x and y values of coordinate positions.

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

Transformation matrix for reflection about the line $\boldsymbol{x = -y}$.

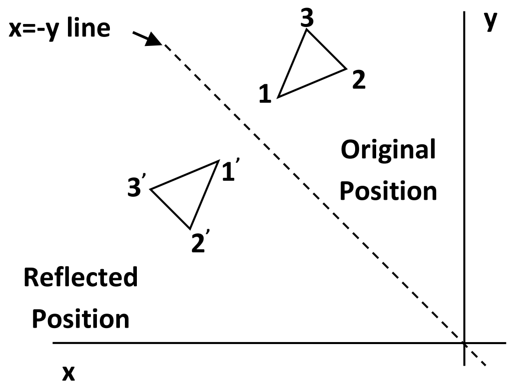


Fig. 3.12: - Reflection about x=-y line.

- This transformation interchange x and y values of coordinate positions.

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

- **Example**: - Find the coordinates after reflection of the triangle [A (10, 10), B (15, 15), C (20, 10)] about x axis.

$$P' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 10 & 15 & 20 \\ 10 & 15 & 10 \\ 1 & 1 & 1 \end{bmatrix}$$

$$P' = \begin{bmatrix} 10 & 15 & 20 \\ -10 & -15 & -10 \\ 1 & 1 & 1 \end{bmatrix}$$

- Final coordinate after reflection are [A′ (10, -10), B′ (15, -15), C′ (20, -10)]

Shear

- A transformation that distorts the shape of an object such that the transformed shape appears as if the object were composed of internal layers that had been caused to slide over each other is called **shear**.
- Two common shearing transformations are those that shift coordinate x values and those that shift y values.

Shear in $\boldsymbol{x - direction}$.

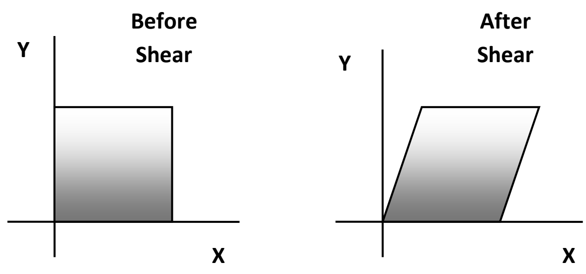


Fig. 3.13: - Shear in x-direction.

- Shear relative to $x - axis$ that is $y = 0$ line can be produced by following equation:
  $\boldsymbol{x' = x + sh_x \cdot y}$, $\boldsymbol{y' = y}$
- Transformation matrix for that is:
  $$\begin{bmatrix} \mathbf{1} & \boldsymbol{sh_x} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$
  Here $\boldsymbol{sh_x}$ is shear parameter. We can assign any real value to $\boldsymbol{sh_x}$.
- We can generate $x - direction$ shear relative to other reference line $y = y_{ref}$ with following equation:
  $\boldsymbol{x' = x + sh_x \cdot (y - y_{ref})}$, $\boldsymbol{y' = y}$
- Transformation matrix for that is:
  $$\begin{bmatrix} \mathbf{1} & \boldsymbol{sh_x} & \boldsymbol{-sh_x \cdot y_{ref}} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$
- **Example**: - Shear the unit square in x direction with shear parameter ½ relative to line $y = -1$.
  Here $y_{ref} = -1$ and $sh_x = 0.5$
  Coordinates of unit square are [A (0, 0), B (1, 0), C (1, 1), D (0, 1)].
  $$P' = \begin{bmatrix} 1 & sh_x & -sh_x \cdot y_{ref} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$
  $$P' = \begin{bmatrix} 1 & 0.5 & -0.5 \cdot (-1) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$
  $$P' = \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$
  $$P' = \begin{bmatrix} 0.5 & 1.5 & 2 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$
- Final coordinate after shear are [A′ (0.5, 0), B′ (1.5, 0), C′ (2, 1), D′ (1, 1)]

<u>Shear in $\boldsymbol{y - direction}$.</u>

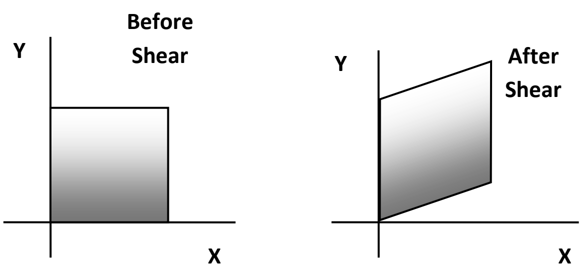


Fig. 3.14: - Shear in y-direction.

- Shear relative to $y - axis$ that is $x = 0$ line can be produced by following equation:
  $\boldsymbol{x' = x}$, $\boldsymbol{y' = y + sh_y \cdot x}$
- Transformation matrix for that is:
  $$\begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{sh_y} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$
  Here $\boldsymbol{sh_y}$ is shear parameter. We can assign any real value to $\boldsymbol{sh_y}$.

- We can generate $y - direction$ shear relative to other reference line $x = x_{ref}$ with following equation:
  $\boldsymbol{x' = x}$, $\quad \boldsymbol{y' = y + sh_y \cdot (x - x_{ref})}$
- Transformation matrix for that is:
  $$\begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{sh_y} & \mathbf{1} & \boldsymbol{-sh_y \cdot x_{ref}} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$
- **Example**: - Shear the unit square in y direction with shear parameter ½ relative to line $x = -1$.
  Here $x_{ref} = -1$ and $sh_y = 0.5$
  Coordinates of unit square are [A (0, 0), B (1, 0), C (1, 1), D (0, 1)].
  $$P' = \begin{bmatrix} 1 & 0 & 0 \\ sh_y & 1 & -sh_y \cdot x_{ref} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$
  $$P' = \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & -0.5 \cdot (-1) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$
  $$P' = \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0.5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$
  $$P' = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0.5 & 1 & 2 & 1.5 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$
- Final coordinate after shear are [A′ (0, 0.5), B′ (1, 1), C′ (1, 2), D′ (0, 1.5)]

The Viewing Pipeline

- **Window:** Area selected in world-coordinate for display is called window. It defines what is to be viewed.
- **Viewport:** Area on a display device in which window image is display (mapped) is called viewport. It defines where to display.
- In many case window and viewport are rectangle, also other shape may be used as window and viewport.
- In general finding device coordinates of viewport from word coordinates of window is called as **viewing transformation.**
- Sometimes we consider this viewing transformation as window-to-viewport transformation but in general it involves more steps.

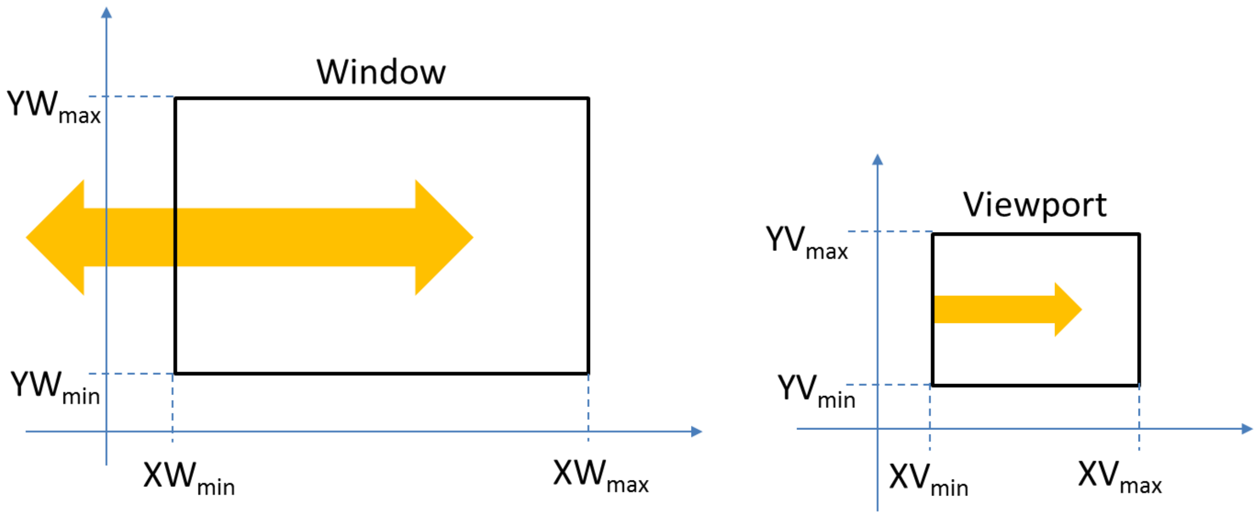


Fig. 3.1: - A viewing transformation using standard rectangles for the window and viewport.

- Now we see steps involved in viewing pipeline.

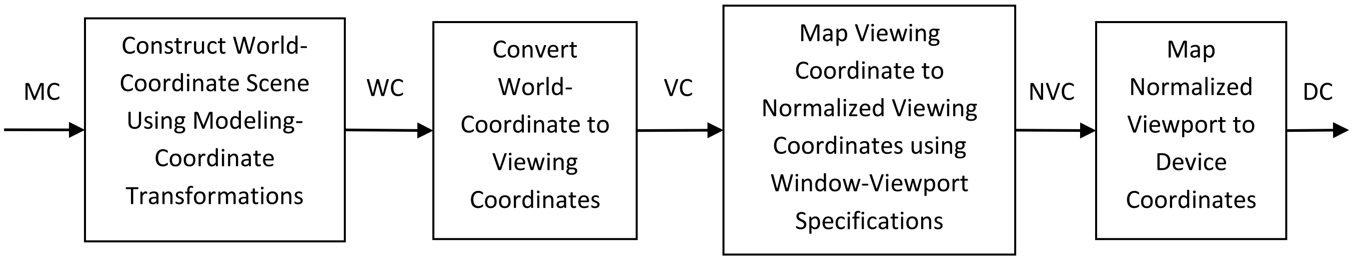


Fig. 3.2: - 2D viewing pipeline.

- As shown in figure above first of all we construct world coordinate scene using modeling coordinate transformation.
- After this we convert viewing coordinates from world coordinates using window to viewport transformation.
- Then we map viewing coordinate to normalized viewing coordinate in which we obtain values in between 0 to 1.
- At last we convert normalized viewing coordinate to device coordinate using device driver software which provide device specification.
- Finally device coordinate is used to display image on display screen.
- By changing the viewport position on screen we can see image at different place on the screen.
- By changing the size of the window and viewport we can obtain zoom in and zoom out effect as per requirement.
- Fixed size viewport and small size window gives zoom in effect, and fixed size viewport and larger window gives zoom out effect.
- View ports are generally defines with the unit square so that graphics package are more device independent which we call as normalized viewing coordinate.

Viewing Coordinate Reference Frame

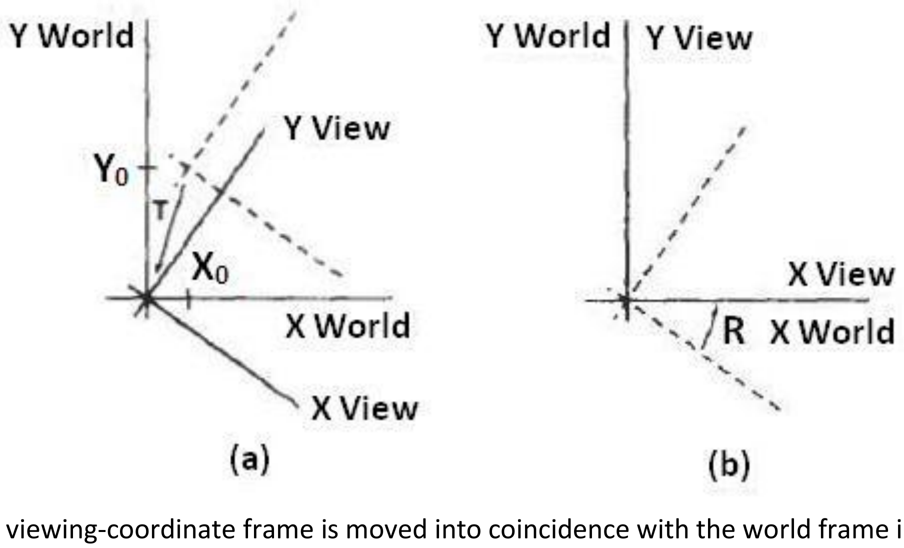


Fig. 3.3: - A viewing-coordinate frame is moved into coincidence with the world frame in two steps: (a) translate the viewing origin to the world origin, and then (b) rotate to align the axes of the two systems.

- We can obtain reference frame in any direction and at any position.
- For handling such condition first of all we translate reference frame origin to standard reference frame origin and then we rotate it to align it to standard axis.
- In this way we can adjust window in any reference frame.
- this is illustrate by following transformation matrix:

$$\mathbf{M_{wc,vc} = RT}$$

- Where T is translation matrix and R is rotation matrix.

Window-To-Viewport Coordinate Transformation

- Mapping of window coordinate to viewport is called window to viewport transformation.
- We do this using transformation that maintains relative position of window coordinate into viewport.
- That means center coordinates in window must be remains at center position in viewport.
- We find relative position by equation as follow:

$$\frac{\mathbf{x_v - x_{vmin}}}{\mathbf{x_{vmax} - x_{vmin}}} = \frac{\mathbf{x_w - x_{wmin}}}{\mathbf{x_{wmax} - x_{wmin}}}$$

$$\frac{\mathbf{y_v - y_{vmin}}}{\mathbf{y_{vmax} - y_{vmin}}} = \frac{\mathbf{y_w - y_{wmin}}}{\mathbf{y_{wmax} - y_{wmin}}}$$

- Solving by making viewport position as subject we obtain:

$$\mathbf{x_v = x_{vmin} + (x_w - x_{wmin})s_x}$$
$$\mathbf{y_v = y_{vmin} + (y_w - y_{wmin})s_y}$$

- Where scaling factor are :

$$\mathbf{s_x = \frac{x_{vmax} - x_{vmin}}{x_{wmax} - x_{wmin}}}$$
$$\mathbf{s_y = \frac{y_{vmax} - y_{vmin}}{y_{wmax} - y_{wmin}}}$$

- We can also map window to viewport with the set of transformation, which include following sequence of transformations:
  1. Perform a scaling transformation using a fixed-point position of (x_{Wmin},y_{wmin}) that scales the window area to the size of the viewport.
  2. Translate the scaled window area to the position of the viewport.
- For maintaining relative proportions we take ($s_x = s_y$). in case if both are not equal then we get stretched or contracted in either the x or y direction when displayed on the output device.
- Characters are handle in two different way one way is simply maintain relative position like other primitive and other is to maintain standard character size even though viewport size is enlarged or reduce.
- Number of display device can be used in application and for each we can use different window-to-viewport transformation. This mapping is called the **workstation transformation.**

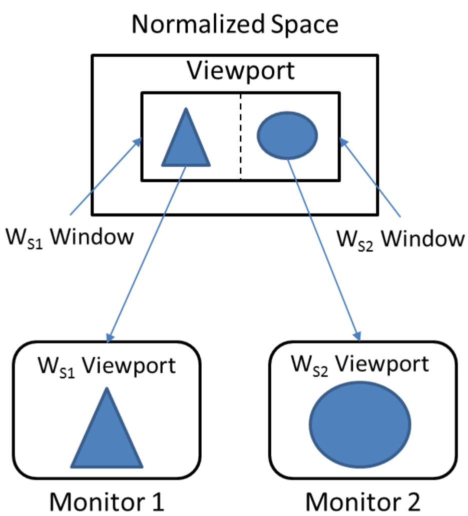


Fig. 3.4: - workstation transformation.