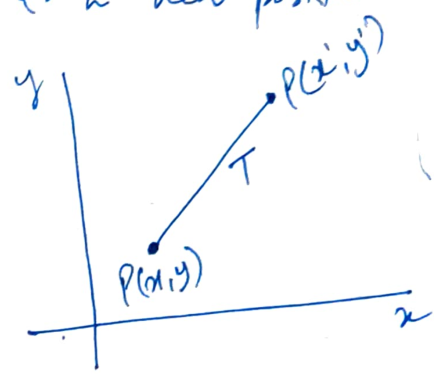
4. Transformation

Transformation is a process to accomplish the changes in orientation, size, & shape of the objects. The basic geometric transformation techniques include translation, rotation, & scaling. Some other advanced transformation techniques include reflection & shear.

4.1. Translation

It is a process of repositioning an object from one coordinate location to another along a straight line path. A 2-D point is translated by adding translation distance (t_x & t_y) to the original coordinate (x, y) to move to a new position (x', y').



$$x' = x + t_x, \quad y' = y + t_y$$

(t_x, t_y) is called a translation vector or shift vector. The above translation eqns can be represented in the matrix form as:

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

Then, $P' = P + T$

or

$$P = [x, y], \quad P' = [x' \; y'], \quad \& \quad T = [t_x \; t_y]$$

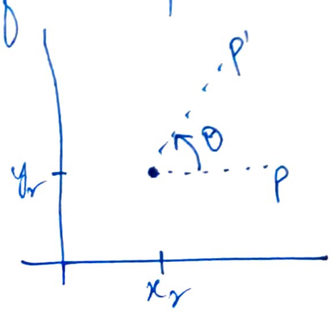
Then, $P' = P + T$

Column-vector representation is generally considered as the standard notation as it is used in many graphics packages like, GKS, PHIGS, etc.

Translation is a rigid-body transformation that moves objects without deformation, i.e., every point is translated equally. A straight line segment is translated by applying the translation eqns on the two endpoints and then redrawing the line thru the new endpoints. Polygons are translated by adding the translation vector to each vertex and regenerating the polygon using the new set of vertices. To change the position of an ellipse or a circle, the center is translated and the figure is redrawn. Other curves like splines are translated by displacing the coordinates defining the object.

4.2. Rotation

It is a process of repositioning of an object along a circular path in the xy plane. Rotation is specified by a rotation angle θ and a rotation point (x_r, y_r). Positive θ represents anti-clockwise rotation while negative θ represents clockwise rotation. Rotation can also be described as the repositioning of an object about an axis perpendicular to the xy plane.



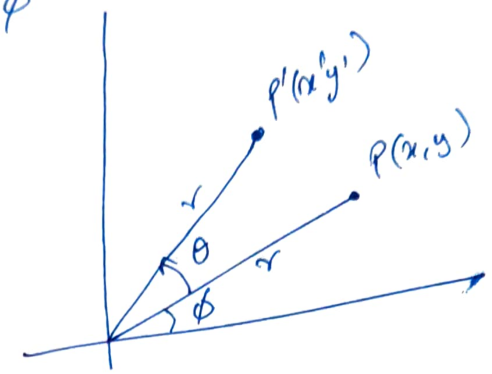
Considering the rotation point at the origin, the equations for rotation of a point P are:

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

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

where, r is the constant distance of the point from the origin (pivot point), ϕ is the original angular position of the point from the horizontal, and θ is the rotation angle. The original coordinates of the point in polar form are:

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



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

In the matrix form:

$$P' = R \cdot P$$

where, $R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$, $P = \begin{bmatrix} x \\ y \end{bmatrix}$

If $P = [x \; y]$ is represented as a row matrix, ~~the~~ then

~~we take transpose of R as:~~

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

$$P' = (R \cdot P)^T$$

or $P' = P^T \cdot R^T$

Now, rotation of a point about an arbitrary pivot position (x_r, y_r) will be given as:

$$x' = x_r + (x - x_r)\cos\theta - (y - y_r)\sin\theta$$
$$y' = y_r + (x - x_r)\sin\theta + (y - y_r)\cos\theta$$

4.3. Scaling

It transforms the size of an object by multiplying the coordinate values (x, y) of each vertex by scaling factors S_x & S_y to produce the transformed coordinates (x', y').

$$x' = x \cdot S_x \quad \& \quad y' = y \cdot S_y$$

In the matrix form :

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

$$\Rightarrow \quad P' = S \cdot P$$

Any the numeric values can be assigned to the scaling factor S_x & S_y. Values less than 1 reduce the size and values greater than 1 enlarge the size of the objects.

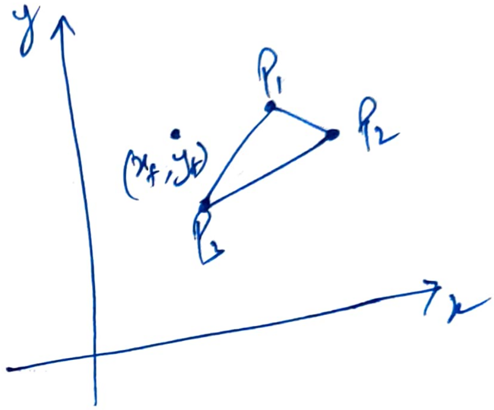
Broadly, there are two types of scaling transformations:

(a) Uniform Scaling - In this type of scaling, S_x & S_y are assigned the same value to maintain relative object proportions.

(b) Differential Scaling - In differential scaling, S_x & S_y are assigned unequal values. It is often used in design applications, where pictures are constructed from a few basic shapes that can be adjusted by scaling and positioning transformations.

Objects transformed with the above equation are both scaled and repositioned. Scaling factors with values less than 1 move objects closer to the origin, while values greater than 1 move objects farther from the origin.

The location of a scaled object can be controlled by choosing a position, called the fixed point. This fixed point is to remain unchanged after the scaling transformation. Coordinates of the fixed point (x_f, y_f) can be chosen as one of the vertices, the object centroid, or any other position. A polygon is then scaled relative to the fixed point by scaling the distance from each vertex to the fixed point. For a vertex (x, y), the scaled coordinates (x', y') are calculated as:



$$x' = x_f + (x - x_f) S_x$$

$$y' = y_f + (y - y_f) S_y$$

OR

$$x' = x \cdot S_x + x_f (1 - S_x)$$

$$y' = y \cdot S_y + y_f (1 - S_y)$$

4.4. Matrix Representations and Homogeneous coordinates

Graphics apps may involve sequences of geometric transformations. We need to reformulate the earlier matrix representations so that sequential transformations can be processed. In the general form:

$$P' = M_1 \cdot P + M_2$$

Where M_1 and M_2 are the matrices containing multiplicative and additive factors respectively. For translation, M_1 is the identity matrix. For rotation or scaling, M_2 contains the translational terms associated with pivot point or scaling fixed point. For sequential transformations, following order should be followed:

Scaling → Rotation → Translation

However, instead to this step-by-step transformation, it would be efficient to combine the transformations so that the final coordinates are obtained directly from the initial coordinates. To do so, we need to represent each cartesian coordinate position (x, y) as a homogeneous coordinate triple (x_h, y_h, h) where $x_h = h \cdot x$ & $y_h = h \cdot y$. For 2-D transformations, h can take any non-zero value. For convinience, h is set to 1 resulting in the homogeneous coordinates $(x, y, 1)$.

, for translation:

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

For rotation:

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

For scaling:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow P' = S(S_x, S_y) \cdot P$$

4.5. Composite Transformations

Using the matrix representations, any sequence of transformations can be obtained to form a composite transformation matrix by calculating the matrix product (R to L) of the individual transformations. Formation of the composite transformation matrix is often referred as concatenation or composition of matrices.

(a) ~~Trans~~ Composite Translations

If two successive translation vectors (t_{x1}, t_{y1}) and (t_{x2}, t_{y2}) are applied to ~~coo~~ P, the final transformed location P' is calculated as

$$P' = T(t_{x_2}, t_{y_2}) \cdot \{T(t_{x_1}, t_{y_1}) \cdot P\}$$

or $$P' = \{T(t_{x_2}, t_{y_2}) \cdot T(t_{x_1}, t_{y_1})\} \cdot P$$

Composite transformation matrix for this sequence of translation is

$$\begin{bmatrix} 1 & 0 & tx_2 \\ 0 & 1 & ty_2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & tx_1 \\ 0 & 1 & ty_2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx_1 + tx_2 \\ 0 & 1 & ty_1 + ty_2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow T(tx_2, ty_2) \cdot T(tx_1, ty_1) = T(tx_1 + tx_2, ty_1 + ty_2)$$

which demonstrates that two successive translations are additive

(b) Composite Rotations

Two successive rotations applied to P produce the transformed position

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

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

Composite transformation matrix for this sequence of ~~translation~~ rotation is

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

$$\Rightarrow R(\theta_2) \cdot R(\theta_1) = R(\theta_1 + \theta_2)$$

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

which demonstrates that two successive rotations are additive

Composite Scalings.

Two successive scalings applied to P produce the transformed position

$$P' = S(Sx_2, Sy_2) \cdot \{S(Sx_1, Sy_1) \cdot P\}$$

or $$P' = \{S(Sx_2, Sy_2) \cdot S(Sx_1, Sy_1)\} \cdot P$$

Composite transformation matrix for this sequence of scaling is

$$\begin{bmatrix} Sx_2 & 0 & 0 \\ 0 & Sy_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} Sx_1 & 0 & 0 \\ 0 & Sy_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} Sx_1 \cdot Sx_2 & 0 & 0 \\ 0 & Sy_1 \cdot Sy_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow S(Sx_2, Sy_2) \cdot S(Sx_1, Sy_1) = S(Sx_1 \cdot Sx_2, Sy_1 \cdot Sy_2)$$

$$\Rightarrow P' = S(Sx_1 \cdot Sx_2, Sy_1 \cdot Sy_2) \cdot P$$

(d) General Pivot-Point Rotation.

We can generate rotations about any selected pivot point (x_r, y_r) by in the following sequence:

(i) Translate the object to move the pivot-point position to the origin.

(ii) Rotate the object about the coordinate origin.

(iii) Translate the object to move the pivot-point position to its original position.

The composite transformation matrix for this sequence is obtain...

संस्थान

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

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

$$\Rightarrow R(x_r, y_r, \theta) = T(x_r, y_r) \cdot R(\theta) \cdot T(-x_r, -y_r)$$

(e) General fixed-Point Scaling

We can ~~generate~~ scale an object about a fixed point (x_f, y_f) in the following sequence:

(i) Translate the object to move the fixed point position to the origin.

(ii) Scale the object w.r.t. origin.

(iii) Translate the object to move the fixed point position to its original position

The composite transformation matrix for this sequence is obtained as

$$\begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix}$$

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

$$\Rightarrow S(x_f, y_f, S_x, S_y) = T(x_f, y_f) \cdot S(S_x, S_y) \cdot T(-x_f, -y_f)$$

General Scaling Directions

Parameters S_x & S_y scale objects along the x & y directions. We can scale an object in other directions by using the following sequence:

(i) Rotate the object to allign the desired scaling directions with the coordinate axes

(ii) Apply the desired scaling transformations.

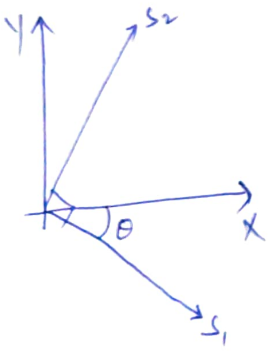
(iii) Rotate the object in the opposite direction to return the points to their original orientation.

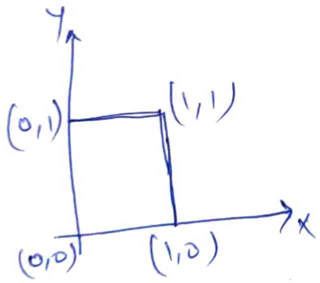
The composite matrix for this sequence is obtained as

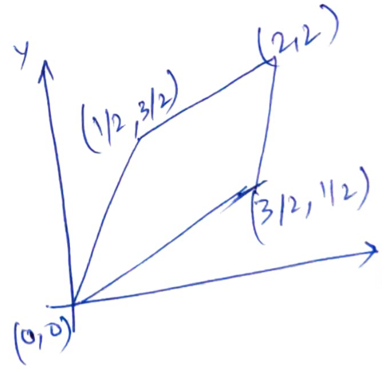
$$\begin{bmatrix} \cos(-\theta) & -\sin(-\theta) & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} S_1 & 0 & 0 \\ 0 & S_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_1\cos^2\theta + S_2\sin^2\theta & (S_2-S_1)\cos\theta\sin\theta & 0 \\ (S_2-S_1)\cos\theta\sin\theta & S_1\sin^2\theta + S_2\cos^2\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= R(-\theta)\cdot S(S_1,S_2)\cdot R(\theta)$$







We could take this scaling operation one step further and concatenate the matrix with translation operators, so that the composite matrix would include parameters for the specification of a scaling fixed position

(g) Concatenation Properties

Matrix multiplication is associative

$\Rightarrow A \cdot B \cdot C = (A \cdot B) \cdot C = A \cdot (B \cdot C)$

Matrix multiplication may not be commutative.

$\Rightarrow$ in general, $A \cdot B \neq B \cdot A$

In composite transformations, some special cases where matrix multiplication is commutative are:

- sequence transformations all of the same kind.
- Sequence transformation of rotation and uniform scaling.

(h) General Composite Transformations and computational Efficiency

A general 2-D transformation, representing a combination of translations, rotations and scaling, can be expressed as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} rS_{xx} & rS_{xy} & trS_x \\ rS_{yx} & rS_{yy} & trS_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

The four element rS_{ij} are the multiplicative rotation - scaling terms. trS_x & trS_y are the translation terms containg combinations of translation distances, pivot-point and fixed-point coordinates, and rotation angles and scaling parameters.