Introduction to Computer Graphics (CS602) Lecture 11

2D Transformations I

In the previous lectures so far we have discussed output primitive as well as filling primitives. With the help of them we can draw an attractive 2D drawing but that will be static whereas in most of the cases we require moving pictures for example games, animation, and different model; where we show certain objects moving or rotating or changing their size.

Therefore, changes in orientation that is displacement, rotation or change in size is called geometric transformation. Here, we have certain basic transformations and some special transformation. We start with basic transformation.

11.1 Basic Transformations

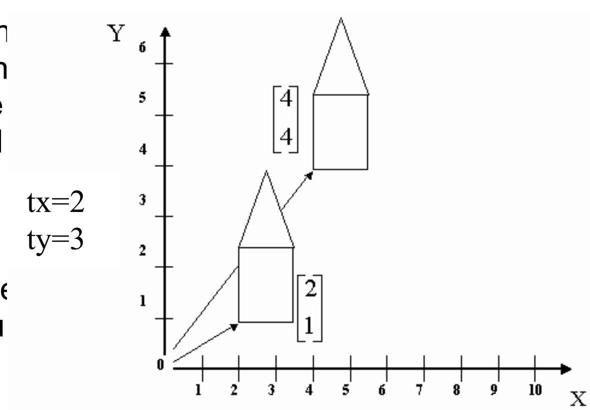
a) Translation
b) Rotation
c) Scaling

Above are three basic transformations. Where translation is independent of others whereas rotation and scaling depends on translation in most of cases. We will see how in their respective sections but here we will start with translation.

a) Translation

A translation is displacement from origin be along a straingt line; where two distan and second is along y-axis that is t_y. The express it with following equation as well

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



Here (t_x, t_y) is translation vector or shift ve as a single matrix equation by using colu positions and the translation vector:

$$P' = P + T$$

Where P = $\begin{bmatrix} x \\ y \end{bmatrix}$ P′= $\begin{bmatrix} x' \\ y' \end{bmatrix}$ T = $\begin{bmatrix} t_x \\ t_y \end{bmatrix}$

Translation is a rigid-body tansofrmation that moves objects without deformation. That is, every point on the object is translated by the same amount.

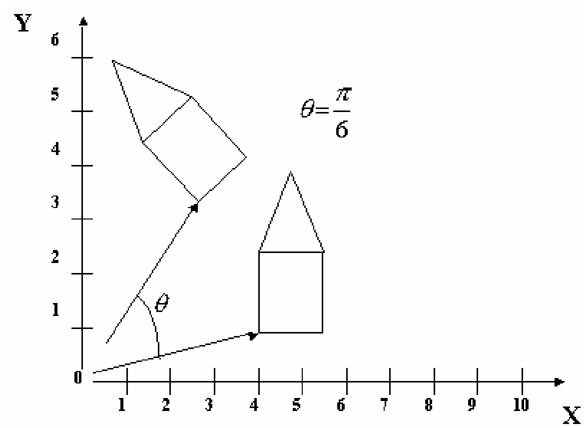
A **straight line** can be translated by applying the above transformation equation to each of the line endpoints and redrawing the line between the new coordinates. Similarly a **polygon** can be translated by applying the above transformation equation to each verticies of the polygon and redrawing the polygon with new coordinates. Similarly **curved objects** can be translated. For example to translate **circle or ellipse**, we translate the center point and redraw the same using new center point.

b) Rotation

A two dimensional rotation is applied to an object by repositinong it along a circular path in the xy plane. To rotate a point, its coordinates and rotation angle is required. Rotation is performed around a fixed point called pivot point. In start we will assume pivot point to be the origin or in other words we will find rotation equations for the rotation of object with respect to origin, however later we will see if we change our pivot point what should be done with the same equations.

Another thing is to be noted that for a positive angle the rotation will be anti-clockwise where for negative angle rotation will be clockwise.

Now for the rotation around the origin as shown in the above figure we required original position/ coordinates which in our case is P(x,y) and rotation angle θ. Now using polar coordinates assume point is already making angle Φ from origin and distance of point from origin is r, therefore we can represent x and y in the form:



$$x = r\cos\Phi \text{ and } y = r\sin\Phi$$

Now if we want to rotate point by an angle θ, we have new angle that is (Φ+ θ), therefore now point P′(x′,y′) can be represented as:

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

and

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

Now replacing r cosΦ = x and r sinΦ = y in above equations we get:

$$x' = x\cos\theta - y\sin\theta \text{ and } y' = x\sin\theta + y\cos\theta$$

Again we can represent above equations with the help of column vectors:

$$P' = R \cdot P$$

- Where

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

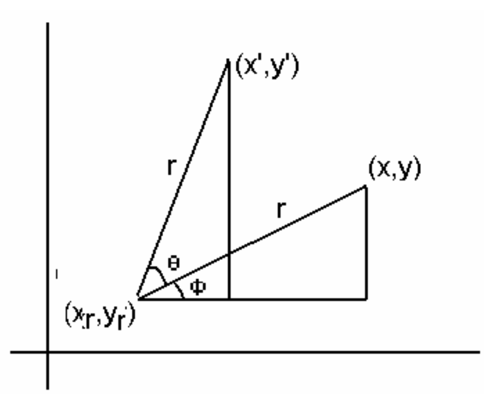
When coordinate positions are represented as row vectors instead of column vectors, the matrix product in rotation equation is transposed so that the transformed row coordinate vector [x′,y′] is calculated as:

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

Where P^T and the other transpose matrix can be obtained by interchanging rows and columns. Also, for rotation matrix, the transpose is obtained by simply changing the sign of the sine terms.

Rotation about an Arbitrary Pivot Point:

As we discussed above that pivot point may be any point as shown in the above figure, however for the sake of simplicity we assume above that pivot point is at origin.



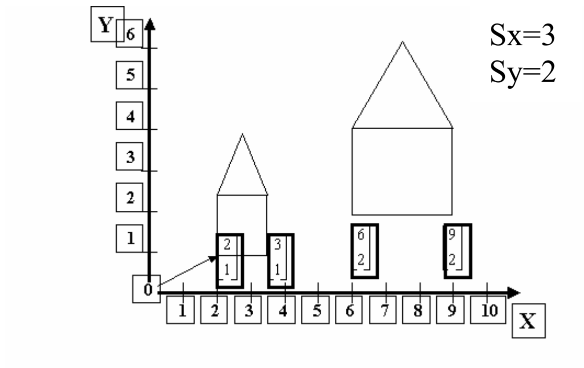
Anyhow, the situation can be dealt easily as we have equations of rotation with respect to origin. We can simply involve another transformation already read that is tranlation so simply translate pivot point to origin. By translation, now points will make angle with origin, therefore apply the same rotation equations and what next? Simply retranslate the pivot point to its original position that is if we subtract x_r,y_r now add them therefore we get following equations:

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

As it is discussed in translation rotation is also rigid-body transformation that moves the object along a circular path. Now if we want to rotate a **point** we already achieved it. But what if we want to move a **line** along its one end point very simple treat that end point as pivot point and perform rotation on the other end point as discussed above. Similarly we can rotate any **polygon** with taking some pivot point and recalculating verticies and then redrawing the polygon.

c) Scaling

A scaling transformation changes the size of an object. Scaling may be in any terms means either increasing the original size or decreasing the original size. An exemplary scaling is shown in the above figure where scaling factors used Sx=3 and Sy=2. So, what are these scaling factors and how they work very simple, simply we multiply each coordinate with its respective scaling factor.



Therefore, scaling with respect to origin is achieved by multiplying x coordinate with factor S_x and y coordinate with factor S_y. Therefore, following equations can be expressed:

$$x' = x.S_x \qquad y' = y.S_y$$

In matrix form it can be expressed as:

$$P' = S.P$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix}$$

Now we may have different values for scaling factor. Therefore, as it is multiplying factor therefore, if we have scaling factor > 1 then the object size will be increased than original size; whereas; in reverse case that is scaling factor < 1 the object size will be decreased than original size and obviously there will be no change occur in size for scaling factor equal 1.

Two variations are possible in scaling that is having scaling factors to be kept same that is to keep original shape; which is called uniform scaling having Sx factor equal Sy factor. Other possibility is to keep Sx and Sy factor unequal that is called differential scaling and that will alter the original shape that is a square will no more remain square.

Now above equation of scaling can be applied to any line, circle and polygon etc. However, as in case of line and polygon we will scale ending points or vertices then redraw the object but in circle or ellipse we will scale the radius.

Now coming to the point when scaling with respect to any point other then origin, then same methodology will work that is to apply translation before scaling and retranslation after scaling. So here if we consider fixed/ pivot point (x_f,y_f), then following equations will be achieved:

$$x' = x_f + (x - x_f)S_x$$
$$y' = y_f + (y - y_f)S_y$$

These can be rewritten as:

$$x' = x.\ S_x + x_f\,(1 - S_x)$$
$$y' = y.\ S_y + y_f\,(1 - S_y)$$

Where the terms $x_f\,(1 - S_x)$ and $y_f\,(1 - S_y)$ are constant for all points in the object.