Computer Graphics
Third stage

2018-2019

Al-Mustansiriyah university
Education college
Computer science department

Chapter Six

6.1 Two Dimension(2D)- Transformations

One of the most common and important tasks in computer graphics is to transform the coordinates (position, orientation, size and shape) of either objects within the graphical scene or the camera that is viewing the scene. It is also frequently necessary to transform coordinates from one coordinate system to another, (e.g. world coordinates to viewpoint coordinates to screen coordinates). All of these transformations can be efficiently and sufficiently handled using some simple matrix representations, which we will see can be particularly useful for combining multiple transformations into a single composite transform matrix.

6.2 Fundamental Transformation

Three basic types of transformations that can perform in two dimensions::
  A. Translation(shift OR move).
  B. Scaling.
  C. Rotation

These basic transformations can also be combined to obtain more complex transformations.

Representation of Points/Objects

A point p in 2D is represented as a pair of numbers: p= (x, y) where x is the x-coordinate of the point p and y is the y-coordinate of p . 2D objects are often represented as a set of points (vertices), $\{p_1, p_2,..., p_n\}$. We can also write points in vector/matrix as : p=(xy)

6.2.A) Translation:
A translation moves an object to a different position on the screen. You can translate a point in 2D by adding translation coordinate (t_x, t_y) to the original coordinate (X, Y) to get the new coordinate (X', Y').

Computer Graphics
Third stage

2018-2019

Al-Mustansiriyah university
Education college
Computer science department
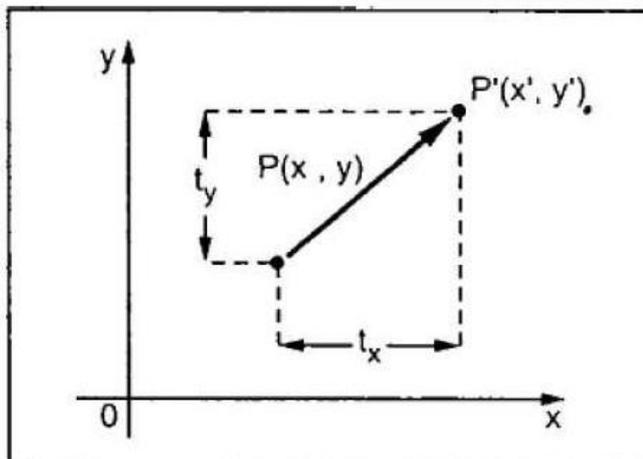
Consider a point P(x, y). we can translated it means shift it to new position p`(x`, y`) by adding tx and ty in y where Tx and Ty are translating factor.

Mathematically this can be represented as:

x`= x + Tx

y`= y + Ty

Note1:Using coordinate system the translating factor are

If Tx>0 then point moves to the right.

If Tx<0 then point moves to the left.

If Ty>0 then point moves to the up.

If Ty<0 then point moves to the down.

Note2*:* using coordinate system in screen then translating factor are

If TX>0 then point moves to the right.

If TX <0 then point moves to the left.

If Ty >0 then point moves to the down.

If Ty <0 then point moves to the up.

We can represent the translation transformation using column vectors as follows:

P′ = P+T, $P = \begin{bmatrix} x \\ y \end{bmatrix}$, $T = \begin{bmatrix} tx \\ ty \end{bmatrix}$, $P' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} tx \\ ty \end{bmatrix}$

$$P' = \begin{bmatrix} x + tx \\ y + ty \end{bmatrix}$$

Example : Consider a triangle defined by its three vertices (20,0), (60,0), (40,100) been moved 100 units to the right and 10 units up. Find the new coordinates of this triangle after translation.

So, the new coordinated of the triangle are :

$T = \begin{bmatrix} 100 \\ 10 \end{bmatrix}$, $P = \begin{bmatrix} 20 & 60 & 40 \\ 0 & 0 & 100 \end{bmatrix}$, $P' = \begin{bmatrix} 20 + tx & 60 + tx & 40 + tx \\ 0 + ty & 0 + ty & 100 + ty \end{bmatrix}$

So, the new coordinated of the triangle are : $P' = \begin{bmatrix} 120 & 160 & 140 \\ 10 & 10 & 110 \end{bmatrix}$

Computer Graphics
Third stage

2018-2019

Al-Mustansiriyah university
Education college
Computer science department
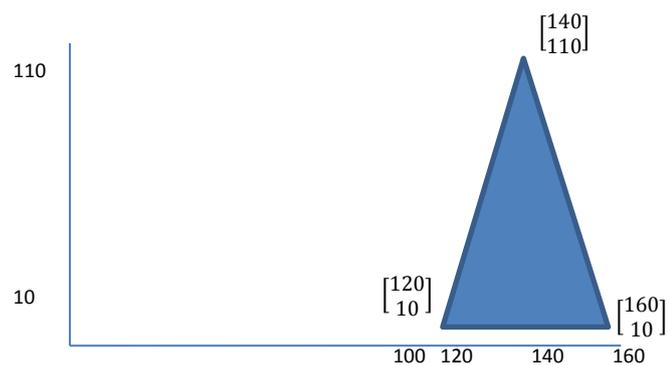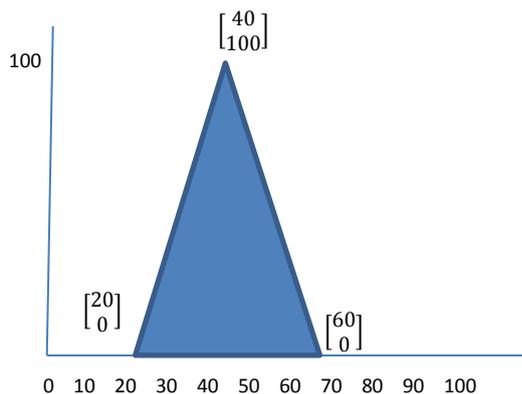
6.2.B) Scaling

: To change the size of an object, scaling transformation is used. In the scaling process, you either expand or compress the dimensions of the object. Scaling can be achieved by multiplying the original coordinates of the object by the scaling factor (S) to get the desired result.

Notes:

✓ If the scaling factor (S < 1) ; then we can reduce the size of the object.

✓ If the scaling factor (S > 1) ; then we can increase the size of the object.

✓ If the scaling factor (S = 1) ; then no change.

Let us assume that the original coordinates are (x, y), the scaling factors are (S_x, S_y), and the produced coordinates are (x', y'). This can be mathematically represented as shown below :

$$\mathbf{x' = x \cdot S_x} \qquad \textbf{and} \qquad \mathbf{y' = y \cdot S_y}$$

The scaling factor S_x, S_y scales the object in x and y direction respectively. The above equations can also be represented in matrix form as below.

$$P' = S \cdot P \qquad\qquad OR \qquad\qquad \binom{X'}{Y'} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \cdot \binom{x}{y}$$

Where S is the scaling matrix.

If $S_x = S_y \rightarrow$ No change in the shape the object (*uniform scaling*)

If $S_x \neq S_y$ change in the shape the object (distortion *in the original object*)

The scaling process is shown in the following figure.

3
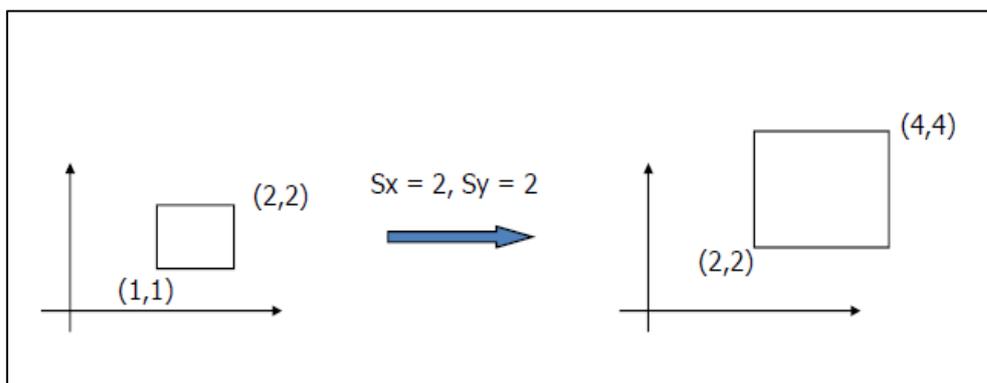
Whenever the scaling process is performed there is one point still no change (same position), this point is called fixed point of the scaling transformation. If the fixed point at the origin, then the point (x,y) can be scaled by a scale factor S_x, S_y in the x-axis and y-axis direction respectively to the new point (x', y').

$$x' = x \cdot S_x \quad ; \quad y' = y \cdot S_y \quad ; \quad \textit{(apply in fixed point in origin point)}$$

6.2.B.1) Scaling about Arbitory Point
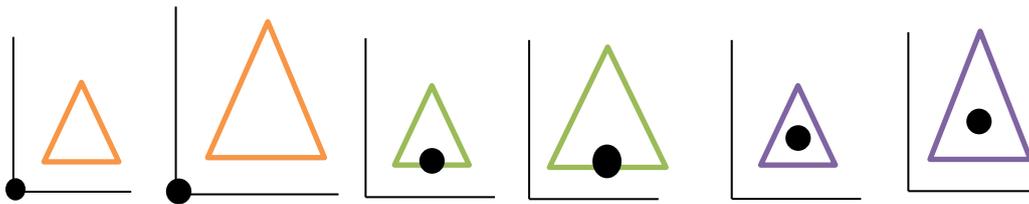
The scaling is performed with respect to any point (x_f, y_f) as fixed point according to the three steps:

1. We must first **translate** the object so that the fixed point is coincide with the origin as follows: every object point (x, y) is moved to the new position (x', y') such as:

$$x' = x - x_f, \qquad y' = y - y_f$$

2. We then scaled these translated points by scale factors S_x and S_y ,so that :

$$x'' = x' \cdot S_x, \qquad y'' = y' \cdot S_y$$

3. Then perform the inverse of the original translation to move the fixed point back to its original position.

$$x''' = x'' + x_f, \qquad y''' = y'' + y_f$$

These three steps can be combined in the following equation that scales a point (xf, yf). $i.e$ the $final$ $equations$ of $scaling$ $object$ $about$ $arbitory$ $point$ are:

$$x''' = \left(x - x_f\right) \cdot S_x + x_f, \qquad y''' = \left(y - y_f\right) \cdot S_y + y_f$$

Computer Graphics
Third stage

2018-2019

Al-Mustansiriyah university
Education college
Computer science department

6.2.C) Rotations

- You can rotate an object about the origin or about a pivot point.
- It is possible to rotate one or more objects or the entire image about any point in world space in either negative oriented a (clockwise) where angle is negative oriented or positive oriented (Anti-clockwise) where angle is positive.

6.2.C.1) Rotate about the origin :

- Any point (x,y) can be represented by its radial distance (r) from the origin and its angle Ø of the x-axis.

    x= r*cos(Ø)
    y=r*sin(Ø)   …….. (1)

- If (x,y) is rotated an angle θ in the **Anti-clockwise** direction. The transformed point $(\bar{x}, \bar{y})$ is represented as:

$$\bar{x} = r * \cos(\emptyset + \theta)$$

$$\bar{y} = r * \sin(\emptyset + \theta) ……. (2)$$

Rotation about the origin
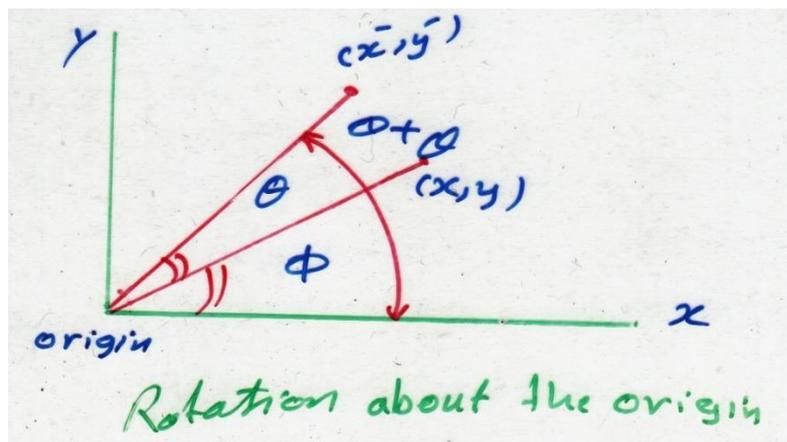
Using the laws of sines and cosines we get:

$$\bar{x} = x\cos(\theta) - y\sin(\theta)$$
$$\bar{y} = x\sin(\theta) + y\cos(\theta) ……. (3)$$

Equation (3) are the transformation that rotate a point an angle (θ) about the origin in the ___Anti-clockwise___ direction.

To rotate an object each point defining that object must be transformed using equation (3). The object is then drawn using the list of transformed points.

To rotate in **_clockwise_** change the angle θ to - θ where:

$\cos(-\theta) = \cos(\theta)$

$\sin(-\theta) = -\sin(\theta)$

So to rotate a point (x,y) through a **_clockwise_** angle θ about the origin of the coordinate system we write:

$$\bar{x} = x\cos(\theta) + y\sin(\theta)$$
$$\bar{y} = -x\sin(\theta) + y\cos(\theta) \quad ....(4)$$

Equation (4) are the transformation that rotate a point an angle (θ) about the origin in the **_clockwise_** direction.

6.2.C.2) Rotate about a pivot point:

After an object is rotated about a specified pivot point, it is still the same distance away from the pivot point but its orientation has been changed:

Pivot Point	Inside object	Pivot Point	Outside object
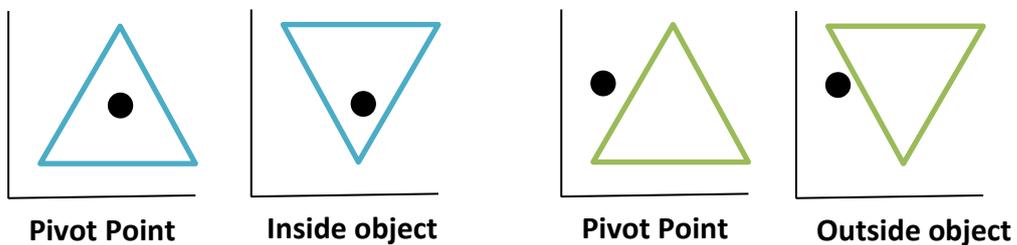
To rotate an object an angle (θ) about a pivot point three steps are required:

1. Translate the pivot point (x_p , y_p) to the origin. Every point (x,y) defining the object is translated to a new point ($\bar{x}, \bar{y}$) where:

$$\bar{x} = x - x_p$$
$$\bar{y} = y - y_p$$

Computer Graphics
Third stage

2018-2019

Al-Mustansiriyah university
Education college
Computer science department

2. Rotate these translated points $(\bar{x}, \bar{y})$ θ degree about the origin to obtain the new point $(\bar{\bar{x}}, \bar{\bar{y}})$

$$\bar{\bar{x}} = \bar{x} * \cos(\theta) - \bar{y} * \sin(\theta)$$
$$\bar{\bar{y}} = \bar{y} * \cos(\theta) + \bar{x} * \sin(\theta)$$

Substituting for $\bar{x}$ and $\bar{y}$

$$\bar{\bar{x}} = (x - x_p) * \cos(\theta) - (y - y_p) * \sin(\theta)$$
$$\bar{\bar{y}} = (y - y_p) * \cos(\theta) + (x - x_p) * \sin(\theta)$$

3. Translate the center of rotation back to the pivot point (x_p, y_p)

$$\bar{\bar{\bar{x}}} = \bar{\bar{x}} + x_p$$
$$\bar{\bar{\bar{y}}} = \bar{\bar{y}} + y_p$$

$i.e$

$$\bar{\bar{\bar{x}}} = (x - x_p) * \cos(\theta) - (y - y_p) * \sin(\theta) + x_p$$
$$\bar{\bar{\bar{y}}} = (y - y_p) * \cos(\theta) + (x - x_p) * \sin(\theta) + y_p$$