UNIT-3

CHAPTER 5

TWO DIMENSIONAL TRANSFORMATIONS

CONTENTS

5.0 Introduction
5.1 Fundamental transformations
  5.1.1 Translation
  5.1.2 Rotation
    5.1.2.1 Rotation About Origin
    5.1.2.2 Rotation About Arbitrary Pivot Point
  5.1.3 Scaling
    5.1.3.1 Uniform Scaling
    5.1.3.2 Reposition Problem
    5.1.3.3 Fixed Point and Repositioning Control
5.2 Other Transformation
  5.2.1 Mirror Reflection
  5.2.2 Shearing
5.3 Homogeneous Coordinates
5.4 Composite Transformations
  5.4.1 Multiple Translation
  5.4.2 Successive Rotation
  5.4.3 Successive Scaling
5.5 Programs for Transformations
  5.5.1 Program For Translation
  5.5.2 Program for Rotation
  5.5.3 Program for Scaling
5.6 Key Words
5.7 Sample Questions

5.0 INTRODUCTION:

The fundamental to all computer graphics is the manipulation of objects. This manipulation can be of several types. Orientation, size, shape are main. This can be done by changing the co-ordinate description of the object. This alteration of the co-ordinate description is known as geometric transformation. Geometric transformation is very important in animation also. Animations are created by moving the object in the specified animation path. In graphic design package, the facilities like changing the shape and orientation etc. should be available for better utilization of the package. There are three basic transformations. These are translation, rotation and scaling. Other popular transformations are, reflection and shear transformation. We shall discuss all of these transformations along with the methods for performing these transformations in this chapter.

5.1 FUNDAMENTAL TRANSFORMATION:

First of all we shall discuss the three fundamental transformations translation, rotation and scaling. Composite and other transformations will be discussed letter.

5.1.1 TRANSLATION:

A translation is moving the object along a straight line from one location to another without changing the shape or direction. The distance between the old position of the object and new position of the object is called the translation distance. Let (x, y) is the original coordinate of a point of the object. Let the translation along the x-axis is tx and translation along y-axis is ty respectively. Then the new co-ordinate of that point (x^1, y^1) is

$$x^1 = x + tx \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots 5.1$$

$$y^1 = y + ty \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots 5.2$$

The translation distance pair (tx, ty) is called translation vector or shift vector and is denoted by T_V.

So the formula stated above can be rewritten in vector addition form as

$$P^1 = T_V + P \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots 5.3$$

where $P^1 = (x^1, y^1)$ and $P = (x, y)$.

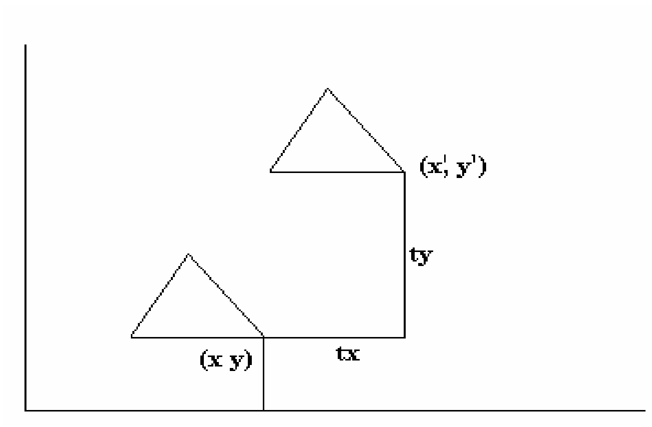


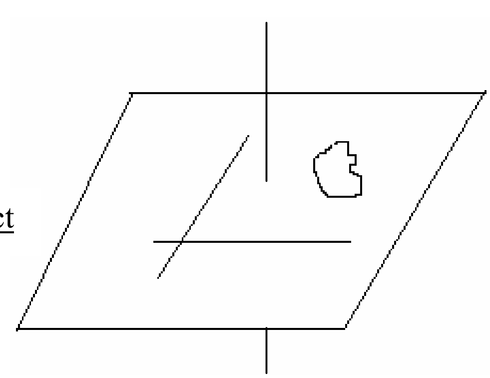
Fig-5.1: Translation of object

Translation is also called rigid body transformations as the shape of the figure do not change with this type of transformation. For an irregular shaped object in computer graphics, the translation is done pixel by pixel. For regular shaped object, the translation is done by shifting the main pixels and then redrawing the object. For example in case of straight line, the end points are shifted and the straight line is redrawn between the new endpoints. For circle, the centre is shifted and then the circle is redrawn with respect to the new centre.

5.1.2 ROTATION:

Rotation means rotation along a circular path i.e., the object will be rotated along a circular path without changing the shape of the object. One natural question that comes is what will be the centre of the circle? For rotation, this is a special point and is user defined. The point about which the rotation will be done is called the rotation point or pivot point. The angle that any point of the object in its old and new position will make at the pivot point is called the angle of rotation. The angle with positive value means rotation in anti-clockwise direction and negative angle means rotation in clockwise direction. The rotation can also be described as follows. Suppose we have placed a two dimensional object on a plane

Fig-5.2: Rotation of an Object



(xy-plane) and we are rotating the plane about one axis perpendicular to xy plane passing through the pivot point. The position of the object will be the rotation about the pivot point.

5.1.2.1 ROTATION ABOUT ORIGIN:

A natural pivot point is the origin. So, we shall first take the origin as the pivot point and perform the transformation. Let the angle of rotation of a point is θ and the originally made an angle φ with the pivot point i.e., with origin.

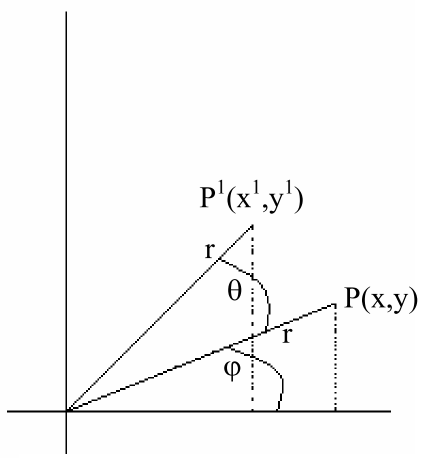


Fig-5.3: Rotation about Origin

Let the old or original point is p(x,y) and the changed point is p^1x^1,y^1), also let the distance of the point from the pivot point is r.

Clearly,

$$x^1 = r \cos(\theta + \varphi) \quad \ldots\ldots 5.4$$

$$y^1 = r \sin(\theta + \varphi) \quad \ldots\ldots 5.5$$

i.e., $$x^1 = r \cos\theta \cos\varphi - r \,\mathrm{Sir}\,\theta \,\mathrm{Sir}\,\varphi \quad \ldots\ldots 5.6$$

$$y^1 = r \cos\theta \sin\varphi) + r \,\mathrm{Sir}\,\theta \cos\varphi \quad \ldots\ldots 5.7$$

It is clear from the above picture, that

$$x = r \cos\varphi \quad \ldots\ldots 5.8$$

$$y = r \sin\varphi \quad \ldots\ldots 5.9$$

Putting those in the equations 5.8 and 5.9 in 5.6 and 5.7 we get

$$x^1 = x \cos\theta - y \sin\theta \quad \ldots\ldots 5.10$$

and $\quad Y^1 = x \, Sin\theta + y \, Cos\theta$ ……………………………………5.11

We can represent the same in matrix form.

Let us define rotation matrix R for the angel θ as

$$R(\theta) = \begin{bmatrix} Cos\,\theta & -Sin\,\theta \\ Sin\,\theta & Cos\theta \end{bmatrix} \qquad \text{……………………………………5.12}$$

$$\text{Let } P = \begin{bmatrix} x \\ y \end{bmatrix} \text{ and } P^1 = \begin{bmatrix} x^1 \\ Y^1 \end{bmatrix}$$

So, we can write,

$$P^1 = R.\,P \qquad \text{……………………………………………5.13}$$

If the point representation in row vector form,

i.e., $\quad P = [\, x, y]$ and $P^1 = [x^1 y^1]$ then

$$P^1 = P.\,R^T \qquad \text{……………………………………………5.14}$$

where R^T is the transpose of the matrix R.

5.1.2.2 ROTATION ABOUT AN ARBITRARY PIVOT POINT:

Instead of the origin, in this case, we shall take an arbitrary point (h, k) as the pivot point.

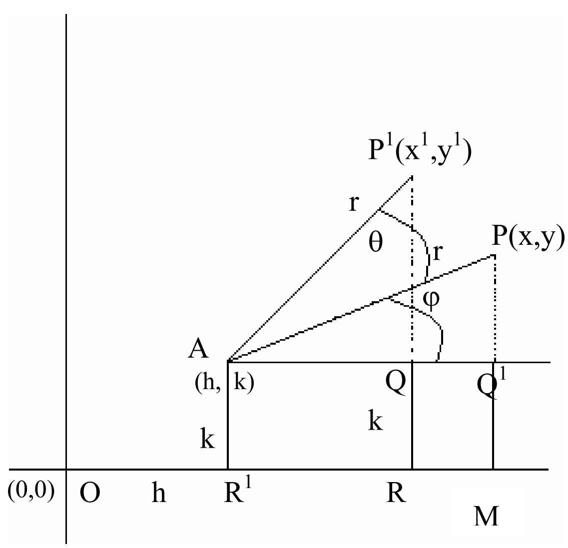


Fig-5.4: Rotation about Arbitrary Pivot

Clearly from the figure 5.4

$$AQ = AP^1 \, Cos\,(\theta + \varphi)$$

$$= r \cos(\theta + \varphi) \quad \ldots\ldots 5.15$$

So, $OR = h + r \cos\varphi \cos\theta - r \sin\varphi \sin\theta$

Or, $x^1 = h + r \cos\varphi \cos\theta - r\sin\varphi \sin\theta \quad \ldots\ldots 5.16$

Similarly

$$P^1Q = r \sin(\theta + \varphi)$$

$\therefore \quad RP^1 = k + r\cos\varphi \sin\theta + r\sin\varphi \cos\theta$

or $\quad y^1 = k + r \cos\varphi \sin\theta + r\sin\varphi \cos\theta \quad \ldots\ldots 5.17$

Now from APQ^1 triangle, we get,

$$AQ^1 = r \cos\varphi$$

i.e., $OM - OR^1 = r \cos\varphi$

$\Rightarrow \quad (x - h) = r \cos\varphi \quad \ldots\ldots 5.18$

Similarly,

$$PQ^1 = r \sin\varphi$$

Or, $PM - Q^1M = r \sin\varphi$

Or, $(y - k) = r \sin\varphi \quad \ldots\ldots 5.19$

Putting 5.18 and 5.19 in 5.17 and 5.18 we get,

$$x^1 = h + (x - h) \sin\theta + (y - k) \sin\theta \quad \ldots\ldots 5.20$$

$$y^1 = k + (x - h) \cos\theta + (y - k) \cos\theta \quad \ldots\ldots 5.21$$

If we take $P = \begin{bmatrix} x \\ y \end{bmatrix}$ $P^1 = \begin{bmatrix} x^1 \\ y^1 \end{bmatrix}$

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

And shift point $S = \begin{bmatrix} h \\ k \end{bmatrix}$

We have

$$P^1 = S + R.(P - S) \qquad \ldots\ldots 5.22$$

As in case of translation, rotation is also called the rigid body transformation because in this case also, the object does not deform.

5. 1.3 SCALING:

Scaling means changing the size of the object i.e., scaling means expanding or compressing the dimensions of an object. The constants that decide the changes in dimensions are called scaling factors. Scaling factor in x and y directions are known as S_x and S_y respectively. If the scaling factor is less than one, then the object will be compressed and if the scaling factor is greater than one the object will be expanded.

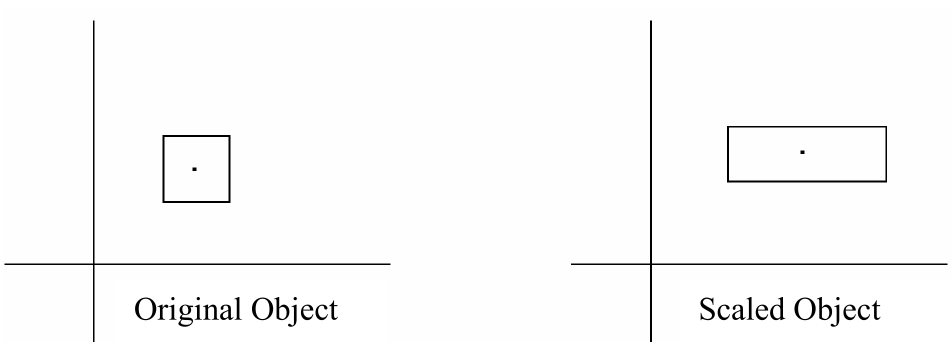


Fig-5.5: Scaling of Object with scale factor greater than one

Let P(x, y) is the original point and the scaling factor along x and y are S_x and S_y. The point will be transformed to $P^1 = (x^1, y^1)$

Where $\qquad x^1 = S_x . x \qquad \ldots\ldots 5.23$

And $\qquad y^1 = S_y . y \qquad \ldots\ldots 5.24$

The transformation equations stated above can also be written as

$$\begin{bmatrix} x^1 \\ y^1 \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix}$$

Or, $\qquad P^1 = S.P \qquad \ldots\ldots 5.25$

where S is the scaling matrix.

5.1.3.3 UNIFORM AND DIFFERENTIAL SCALING:

If both S_x and S_y have the same value, the transformed images shape will remain similar to the original one. Only the size of the whole image will either increase or decrease. This is called uniform scaling. In this case the relative properties remain unaltered.

If the value of S_x and S_y are unequal the corresponding transformation is called differential scaling.

5.1.3.2 REPOSITION PROBLEM:

Consider the case of a straight line scaling with scaling factors S_x = ½ and S_y = ½ where the straight line is vertical to x-axis. Let the coordinates of the end points one P(2, 2) and Q(2,4).

The transformed point for P i.e.,

$$P^1 = \begin{bmatrix} x^1 \\ y^1 \end{bmatrix} = \begin{bmatrix} ½ & 0 \\ 0 & ½ \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Similarly for Q , the changed location is

$$Q^1 = \begin{bmatrix} x^1 \\ y^1 \end{bmatrix} = \begin{bmatrix} ½ & 0 \\ 0 & ½ \end{bmatrix} \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

So the new lines end points are (1,1) and (1,2).

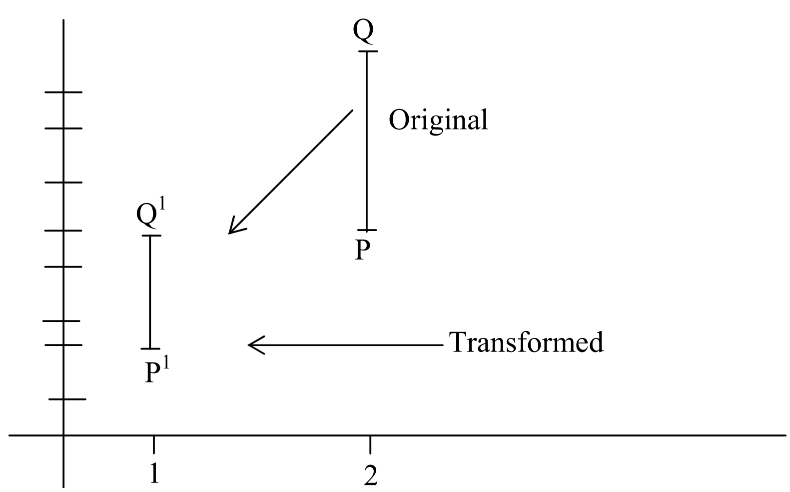


Fig-5.6: Reposition Problem in case of a straight line.

From the figure 5.6 it is clear that not only the size of the straight line is reduced, but also it is shifted towards the origin.

In general if the scaling factors are less than 1 then the object will be shifted towards origin and if the scaling factors are greater than 1 then the object will go away from the origin. This is known as repositioned problem.

5.1.3.3 FIXED POINT AND REPOSITIONING CONTROL:

The repositioning problem can be controlled by introducing fixed point. A fixed point in a scaling a point whose coordinates do not alter during the transformation process.

Let F = (x_f, y_f) be the fixed point. Then the formula of scaling will be

$$x^1 = x_f + (x - x_f)S_x \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 5.26$$

and $$y^1 = y_f + (y - y_f).\, S_y \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 5.27$$

We can rewrite the equations 5.26.and 5.27 as

$$x^1 = x.S_x + x_f(1 - S_x) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 5.28$$

$$y^1 = y.S_y + y_f(1 - S_y) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 5.29$$

Where $X_f(1-S_x)$ and $y_f(1-S_y)$ are constants.

Let us again consider the earlier example of straight line scaling.

Let the fixed point is the middle point of the straight line i.e., F = (2, 3).

Now, in early problem, P^1 will be

$$x^1 = 2 \text{ x } \tfrac{1}{2} + 2(1 - \tfrac{1}{2}) = 2$$

$$y^1 = 2 \text{ x } \tfrac{1}{2} + 3(1 - \tfrac{1}{2}) = 2.5$$

Similarly, Q^1 will be

$$x^1 = 2 \text{ x } \tfrac{1}{2} + 2(1 - \tfrac{1}{2}) = 2$$

$$y^1 = 4 \text{ x } \tfrac{1}{2} + 3(1 - \tfrac{1}{2}) = 3.5$$

So the scaling will be in the same place

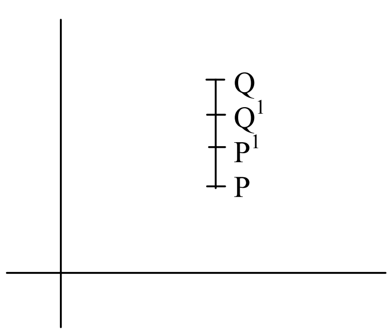


Fig-5.7: Reposition is controlled

5.2: OTHER TRANSFORMATIONS:

Though translation, rotation and scaling are considered as fundamental transformations, there are other transformations which are also very important in graphic design. Mirror reflection and shear transformation.

5. 2.1 MIRROR TRANSFORMATION:

Mirror transformation is a transformation which produces the mirror image

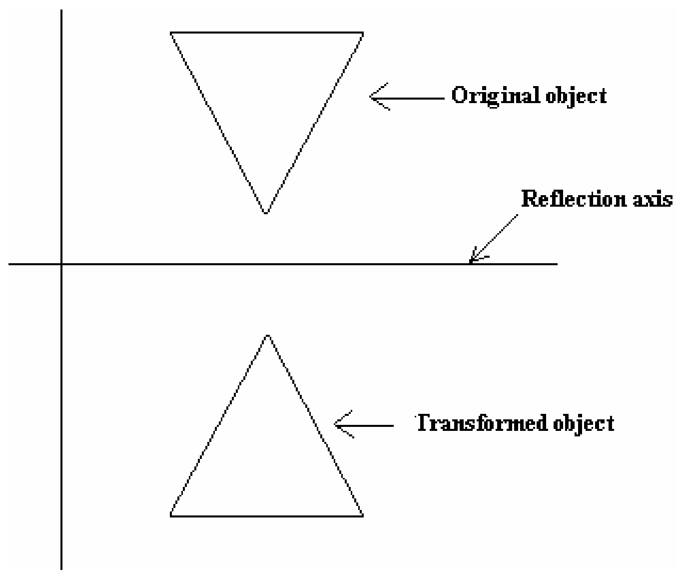


Fig-5.8: Mirror Reflection of an object

of an object with respect to a given mirror axis. Reflection means rotating the object 180^0 with respect to the mirror axis.

Reflection about x-axis i.e., y = 0, can be determined by transformation matrix.

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

Let P(x, y) is the original point, then the transformed image $P^1(x^1, y^1)$ can be given by $P^1 = M. P.$

$$P^1 = \begin{bmatrix} x^1 \\ y^1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ -y \end{bmatrix}$$

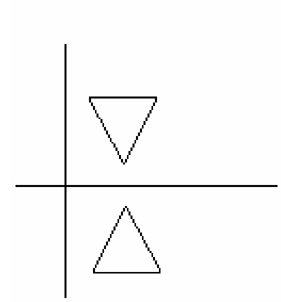


Fig-5.9: Reflection about x-axis

$\therefore x^1 = x$ and $y^1 = -y$.

Similarly mirror reflection about y-axis, i.e., x = 0 can be given by the transformation matrix

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

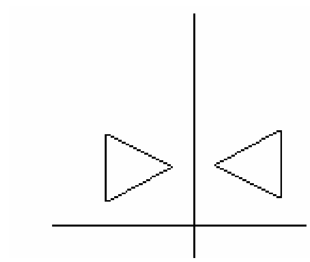


Fig-5.10: Reflection about y-axis

and in that can value of the transformed co-ordinates will be $x^1 = -x$ and $y^1 = y$.

5.2.2 SHEAR TRANSFORMATION:

A transformation that distorts the shape of the object and makes the object slant is called a shear transformation.

There are two types of shear transformations namely x-shear and y-shear.

x-shear transformation : x-shear means slanting in the right hand or left hand side.

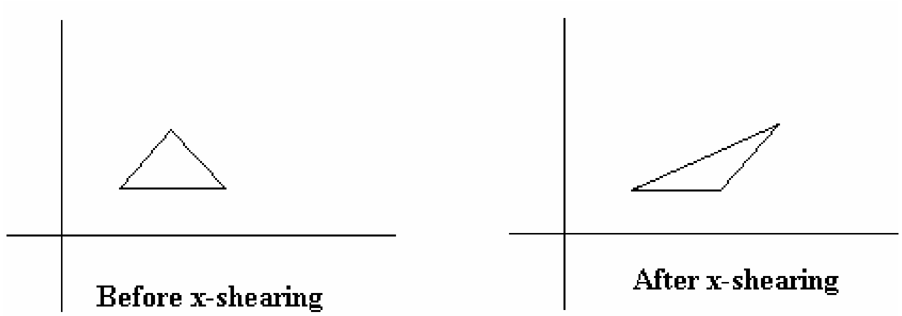


Fig-5.11: Shearing Transformation

x-shearing maintain the y-coordinates but transforms the x-coordinates of the object. x-shearing relative to x-axis can be obtained from the matrix

$$Sh_x = \begin{pmatrix} 1 & Sf_x \\ 0 & 1 \end{pmatrix}$$ Where Sf_x is the x-shearing factor.

So, P(x, y) will be changed to $P^1(x^1, y^1)$

Where $x^1 = x + Sf_x y$

and $y^1 = y$.

Setting Sf_x to 2 will cause the following changes in case of the following object

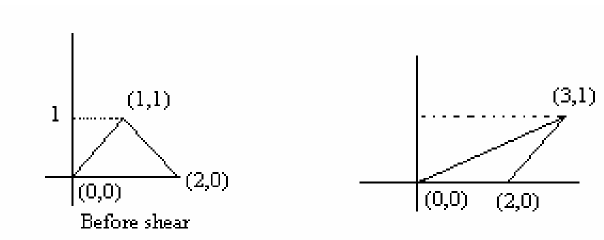


Fig-5.12: x-shearing with shearing factor 2

Similarly y-shear with respect to y-axis can be given by the matrix

$$Sh_y = \begin{pmatrix} 1 & 0 \\ Sf_y & 1 \end{pmatrix}$$

5.3 HOMOGENEOUS COORDINATES:

In reality, many graphics application involves series of transformation at a time. For example, an animation might require translation, rotation and scaling at a time to bring the realistic effect.

One way to do this is to transform the object sequentially one after another, i.e., first translation, then rotation, then scaling. But this will take more time. In an animation, where illusion of motion is a factor, the time complexity plays a central role. In the following section we shall discuss how to perform the sequence of transformations efficiently.

Every basic transformation can be expressed by the general matrices by the following equation.

$$P^1 = M_1 P + M_2 \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 5.30$$

When P is the original point, P^1 is the transformed point, M_1 is a 2 x 2 matrix and M_2 is a 2 x 1 column matrix. P and P^1 are also represented as 2 x 1 column matrices.

For translation, M_1 is a unit matrix and M_2 contains shift vector values. For relation, M_1 contains the rotation matrix value and M_2 contains the shift point adjustment value.

For scaling, M_1 contains the scaling factor matrix and M_2 contains pivot point adjustment. To calculate series of transformations, we have to calculate one step at a time.

A better approach is to obtain the pixel coordinate form the original coordinate directly and thereby elimination of intermediate coordinate values.

To do this, first we shall transform the M_1 and M_2 into a single Matrix of order 3x3. This will allow us to perform all transformations with matrix multiplication only. We shall-in this case-represent the Cartesian coordinates P(x, y) by homogeneous coordinates (x_h, y_h, h) where $x = x_h/h$ and $y = y_h/h$. We shall choose any non-zero h. As there are infinitely many non-zero values for h, there are infinitely many representation of homogeneous coordinate of a Cartesian coordinate. But the most general representation is by taking h = 1.

The representation stated above can perform any basic transformation. For example translation with shift factor (t_x, t_y) can be given by

$$\begin{pmatrix} x^1 \\ y^1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 5.31$$

i.e., $(x^1, y^1, 1) = (x + t_x, y + t_{y,} 1)$

or $x^1 = x + t_x, y^1 = y + t_y$

$p^1 = T. (t_x, t_y).P$

Rotation can be given by

$$\begin{pmatrix} x^1 \\ y^1 \\ 1 \end{pmatrix} = \begin{pmatrix} Cos\theta & -Sin\theta & 0 \\ Sin\theta & Cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ Y \\ 1 \end{pmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 5.32$$

And scaling can be given by

$$\begin{pmatrix} x^1 \\ y^1 \\ 1 \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ Y \\ 1 \end{pmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 5.33$$

$P^1 = S(S_x, S_y) . P$

To get inverse transformation in the above case transformations in the above cases, we shall transform

t_x, t_y to $-t_x$ and $-t_y$ in case of equation 5.31

θ to $-\theta$ in case of 5.32 and S_x, S_y to $1/ S_x$ and $1/ S_y$ in case of 5.33.

5.4 COMPOSITE TRANSFORMATIONS:

Composite Transformation can easily be performed by using homogeneous coordinates. Composite transformation means multiple translations, rotations or scaling or mixing of them.

5.4.1 MULTIPLE TRANSLATIONS:

Let there be two translations with translation vectors (tx_1, ty_1) and (tx_2, ty_2)

$P^1 = T (tx_2, ty_2) [T (tx_1, ty_1).P]$

$= T(tx_2, ty_2). T\ tx_1, ty_1).P$

(As matrix multiplication is associative).

$$P^1 = \begin{pmatrix} x^1 \\ y^1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & tx2 \\ 0 & 1 & ty2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & tx_1 \\ 0 & 1 & ty_1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ Y \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x^1 \\ y^1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & tx_1 + tx_2 \\ 0 & 1 & ty_1 + ty_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$\therefore\ P^1 = T(tx_1 + tx_2, ty_1 + ty_2) . P$

$\therefore\ T (tx_1, tx_2) . T (tx_2, ty_2) = T (tx_1 + tx_2)$ ……………………………5.34

5.4.2 SUCCESSIVE ROTATION:

Two successive rotations can be formulated as follows.

Let $R(\theta_1)$ is the first rotation and then second is $R(\theta_2)$.

$P_1 = R(\theta_2) [R(\theta_1).P]$

$= R(\theta_2). R(\theta_1).P$

It can be shown that $R(\theta_2). R(\theta_1). R(\theta_1) = R(\theta_1 + \theta_2)$

$\therefore\ P' = R(\theta_1+\theta_2).P$

and $R(\theta_1). R(\theta_2) = R(\theta_1+\theta_2)$ ……………………………………5.35

5.4.3 SUCCESSIVE SCALING:

In case of scaling, by the similar multiplication of matrix, it can be shown that $S(Sx_2, Sy_2) . S(Sx_1, Sy_1) = S(Sx_1.Sx_2, Sy_1.Sy_2)$

5.5 PROGRAMS FOR TRANSFORMATIONS:

Following are some C-programs for fundamental transformations. These are program for translation, rotation and scaling. These assumes that all the graphics related files are in f:\tc\bgi folder. If the reader has a different path setting then he/she has to change the initgraph() function accordingly.

5.5.1 PROGRAM FOR TRANSLATION:

```
#include<graphics.h>
#include<conio.h>
struct pt
  {
    int x;
    int y;
  };
typedef struct pt point;
typedef struct pt factor;
void translatePoint(point p1,factor f,point* p2)
  {
    p2->x=p1.x+f.x;
    p2->y=p1.y+f.y;
  }
int main()
  {
    int gd=DETECT,gm;
    point p1,p2,p3,p4;
    factor f;
    f.x=300,f.y=-100,p1.x=100,p1.y=400,p2.x=200,p2.y=200;
    initgraph(&gd,&gm,"F:\\tc\\bgi");
    setcolor(GREEN);
    line(p1.x,p1.y,p2.x,p2.y);
    setcolor(YELLOW);
    outtextxy(50,410,"Before Translation");
    outtextxy(400,410,"Press Enter");
    getch();
    setcolor(GREEN);
    translatePoint(p1,f,&p3);
    translatePoint(p2,f,&p4);
    line(p3.x,p3.y,p4.x,p4.y);
    setcolor(YELLOW);
```

```
   outtextxy(350,310,"After Translation");
   getch();
   closegraph();
   restorecrtmode();
   return(0);
 }/* End of the program*/
```

5.5.2 PROGRAM FOR ROTATION:

```
#include<graphics.h>
#include<conio.h>
#include<math.h>
struct pt
 {
   int x;
   int y;
 };
typedef float angle;
typedef struct pt point;
void rotatePoint(point p1,angle theta,point* p2)
 {
   p2->x=(float)p1.x*cos(theta)-(float)p1.y*sin(theta);
   p2->y=(float)p1.x*sin(theta)+(float)p1.y*cos(theta);
 }
int main()
 {
   int gd=DETECT,gm;
   point p1,p2,p3,p4,p5,p6;
   angle theta=3.14/8.0;
   p1.x=300,p1.y=100,p2.x=300,p2.y=200,p3.x=250,p3.y=150;
   initgraph(&gd,&gm,"F:\\tc\\bgi");
   setcolor(GREEN);
   line(p1.x,p1.y,p2.x,p2.y);
```

```
    line(p2.x,p2.y,p3.x,p3.y);
    line(p3.x,p3.y,p1.x,p1.y);
    setcolor(YELLOW);
    outtextxy(250,210,"Before Rotation");
    outtextxy(400,410,"Press Enter");
    getch();
    setcolor(GREEN);
    rotatePoint(p1,theta,&p4);
    rotatePoint(p2,theta,&p5);
    rotatePoint(p3,theta,&p6);
    line(p4.x,p4.y,p5.x,p5.y);
    line(p5.x,p5.y,p6.x,p6.y);
    line(p6.x,p6.y,p4.x,p4.y);
    setcolor(YELLOW);
    outtextxy(150,310,"After Rotation");
    getch();
    closegraph();
    restorecrtmode();
    return(0);
  }
```

5.5.3 PROGRAM FOR SCALING:

```
#include<graphics.h>
#include<conio.h>
struct pt
  {
    int x;
    int y;
  };
struct fctr
  {
    float Sx;
    float Sy;
```

```
  };
typedef struct pt point;
typedef struct fctr scaleFactor;
void scalePoint(point p1,scaleFactor s,point* p2)
  {
    p2->x=(float)p1.x*s.Sx;
    p2->y=(float)p1.y*s.Sy;
  }
int main()
  {
    int gd=DETECT,gm;
    point p1,p2,p3,p4,p5,p6;
    scaleFactor f;
    f.Sx=0.5,f.Sy=0.5;
    p1.x=300,p1.y=100,p2.x=300,p2.y=200,p3.x=250,p3.y=150;
    initgraph(&gd,&gm,"F:\\tc\\bgi");
    setcolor(GREEN);
    line(p1.x,p1.y,p2.x,p2.y);
    line(p2.x,p2.y,p3.x,p3.y);
    line(p3.x,p3.y,p1.x,p1.y);
    setcolor(YELLOW);
    outtextxy(250,210,"Before Scaling");
    outtextxy(400,410,"Press Enter");
    getch();
    setcolor(GREEN);
    scalePoint(p1,f,&p4);
    scalePoint(p2,f,&p5);
    scalePoint(p3,f,&p6);
    line(p4.x,p4.y,p5.x,p5.y);
    line(p5.x,p5.y,p6.x,p6.y);
    line(p6.x,p6.y,p4.x,p4.y);
    setcolor(YELLOW);
    outtextxy(100,110,"After Scaling");
```

```
  getch();
  closegraph();
  restorecrtmode();
  return(0);
}
```

5.6 KEY WORDS:

- Composite Transformation
- Homogeneous Coordinates
- Reposition Problem
- Rotation
- Scaling
- Scaling Factor
- Shearing
- Transformation Matrix
- Translation
- Uniform Scaling

5.7 SAMPLE QUESTIONS:

5.7.1 What do you mean by rigid body transformation? Explain the translation method along with its mathematical equations.

5.7.2 What is rotation? Establish the mathematical formula for rotation.

5.7.3 What is scale factor? Explain with suitable picture the scaling with x factor and y factor.

5.7.4 What is the effect of scaling if scaling factor is greater or less than one?

5.7.5 Establish the mathematical equation for scaling transformation.

5.7.6 What is uniform scaling?

5.7.7 What is reposition problem? How can it be removed?

5.7.8 Describe what you know about mirror transformation.

5.7.9 What is shear transformation? What is x shearing and y shearing? Explain them with suitable example.

5.7.10 What do you mean by homogeneous coordinates? What are the advantages of homogenous coordinates?

5.7.11 What is composite transformation? Explain multiple translation, successive rotation and successive scaling with their mathematical equations.