ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

A Descriptive Study of 2D Graphics Transformations

Kartik Nagpal

B. Tech Student, Department of Information Technology, Seth Jai Parkash Mukand Lal Institute of Engineering and Technology, Radaur, Yamunanagar, India

ABSTRACT: Computer Graphics is transforming our world significantly and rapidly. Computer Graphics is itself a very broad term and describes everything on the computer which is not text or sound. It can be a video, image, animation or anything else. It can be anything but everything is made up of ‘pixel’. Pixel is the smallest unit of computer graphics. Computer Graphics is an art of drawing line, object, pictures etc. with the help of computer programming. There are two types of computer graphics: Interactive Graphics and Non- Interactive Graphics. Interactive graphics are those which have a two-way communication and the operator can control the operations like playing a video game on a computer. Non- interactive graphics are those in which where the user doesn’t have any kind of control over the graphics.

KEYWORDS: Transformations, Translation, Homogeneous Coordinates, Scaling, Rotation, Reflection, Shearing, Computer Graphics, Matrix representations.

I. Introduction

Computer Graphics definitely looks stunning and are often very complicated and takes a lot of time and effort to make. The techniques and hardware used today are very advanced and has automated most of the things. But this was not the case always. The development of Computer Graphics started in 1950’s when it was designed using the vector graphics. Times changed and the methods also changed. Raster graphics comes into play and make things interesting and easy. But the problem was it doesn’t look real and that’s what stopped it being the mainstream technology.

Nowadays, 3D graphics are used to design everything due to extra features of depth and originality. They are not very much different than 2D graphics apart from the fact that they have an extra dimension. 3D graphics rely on similar algorithms which are used to design 2D graphics. Even 2D graphics can also achieve 3D effects by using different techniques like lightening. The animation is one step ahead 3D graphics where moving objects or graphics are used. No matter how much complicated and advanced the graphic is, it will be using the same basic algorithms and transformation techniques to get the work done.

To design and work on 2D graphics, various techniques are used. The most important techniques are transformations. It simply means changing some graphics into something else by applying rules. We have various types of transformations which are translation, rotation, scaling, reflection and shearing. When the transformation takes place in the 2D plane, then it is called 2D transformations. It plays and important role to reposition the graphic on the screen and change their orientation and size to get the desired result.

II. BASIC TRANSFORMATIONS

2D graphics may combine images, text, equations and more. These components can be modified and manipulated by using two-dimensional geometric transformations such as translation, rotation, and scaling.

A. *TRANSLATION*

A translation is applied to an object to move every pixel of that object a constant distance in a specified direction. It can be described as rigid motion due to the fact that the shape of the object doesn’t change at all. To translate a two-dimensional point, we need to add translation distances t_x and t_y, to the original coordinate position (X,Y) to move the point to a new position (X’,Y’).

DOI: 10.15680/IJIRCCE.2016. 0409028

$$\mathbf{X' = X + t_xY' = Y + t_y}$$

The translation distance pair (t_x,t_y) is called a translation vector. The above equations can also be represented using column vectors.

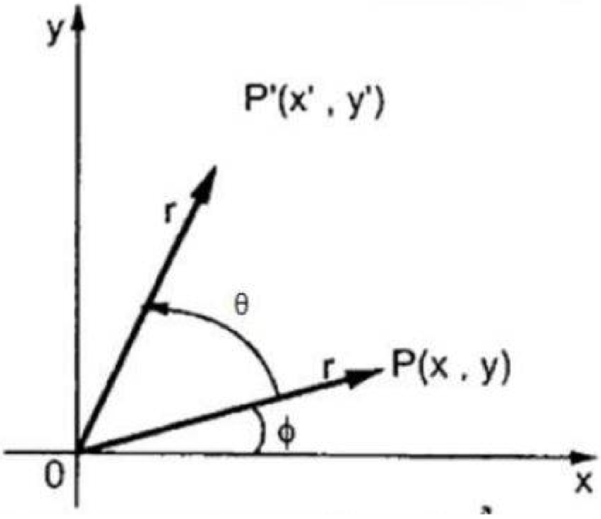
$$P = \frac{[X]}{[Y]} \qquad p' = \frac{[X']}{[Y']} \qquad T = \frac{[t_x]}{[t_y]}$$

We can write it as:

$$\mathbf{P' = P + T}$$

III. Rotation

Rotation is applied to an object by repositioning it along a circular path in atwo-dimensional plane. In order to generate a rotation, we have to specify a rotational angle θ (theta) and the position (x_r,y_r) of the rotation point about which the object is to be rotated. Rotation can be done either in aclockwise direction or anti-clockwise direction. This transformation is also rigid motion because the shape of the object will be the same.



The original coordinate of point P(X,Y) can be represented as –

$$X = r\cos\phi \ldots\ldots(1)$$

$$Y = r\sin\phi \ldots\ldots(2)$$

The point P' can be represented as –

$$x' = r\cos\left(\phi + \theta\right) = r\cos\phi\cos\theta - r\sin\phi\sin\theta \ldots\ldots.(3)$$

$$y' = r\sin\left(\phi + \theta\right) = r\cos\phi\sin\theta + r\sin\phi\cos\theta \ldots\ldots(4)$$

Substituting the above 4 equations, we get-

$$[X'Y'] = [X'Y'] \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} OR$$

$$P' = P \,.\, R$$

Where R is the Rotation matrix.

IV. Scaling

This transformation is applied to resize the object. In the scaling process, we can either expand or compress the dimensions of the object. It can be achieved by multiplying the original coordinates of the object with the scaling factor to get the desired size. There are, further, two types of scaling transformations: uniform and non-uniform. In the uniform scaling, the coordinate values change uniformly which means all three coordinates have the same scaling factor. In the non-uniform scaling, it is not necessarily the same in all the coordinate directions.

To scale a 2D-object with original coordinates (X,Y), the scaling factors are (S_x,S_y) and the produced coordinates are (X’,Y’). This can be represented as shown below –

$$\mathbf{X' = X \,.\, S_X} \qquad \mathbf{and} \qquad \mathbf{Y' = Y \,.\, S_Y}$$

Where S_x and S_y are the scaling factors for x and y coordinates respectively. Representing this relation in the matrix form –

$$\begin{pmatrix} X' \\ Y' \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

or

$$\mathbf{P' = P \,.\, S}$$

Where S is the Scaling matrix.

Scaling is a non-rigid motion transformation due to the fact that the shape and size of the original object changes after the transformation. When the scaling factor is 1, then the size does not change. If both factors S_x and S_y are equal to 1, then there will be no change on the object at all.

The transformations are not used directly. Most of the times, we need to apply multiple transformations to the single object. To tackle this problem, matrix representations, and homogeneous coordinates are used.

V. MATRIX REPRESENTATIONS AND HOMOGENEOUS COORDINATES

Most of the graphics applications involve asequence of geometric representations. We need to apply each set of transformations in the exact sequence to get the desired result. A simple animation may require translation and scaling at each increment of the motion. When using these transformations collectively, they can create a little difficulty. Scaling and Rotation transformations both are handled using the matrix multiplication but Translation is handled using the addition. To use all the transformations, in the same way, we give each point a third coordinate which means instead of using (X,Y), we will be using (X,Y,W). This allows translations to be handled using the matrix multiplication.

To represent a two-dimensional transformation as a matrix multiplication, we represent each Cartesian coordinate position (x,y) with the homogeneous coordinate triple (X_h,Y_h,h) where

$$X=X_h/h \qquad \text{and} \qquad Y=Y_h/h$$

Thus, a general homogeneous coordinate representation can also be written as (h*X,h*Y,h).

When a point (x,y) is converted into a homogeneous representation (x_h,y_h,h), equations containing x and y, such as f(x,y)=0, become homogeneous equations in the three parameters. This means that if each of the parameters is replaced by any value u times that parameter, the value u can be factored out of the equation.

The matrix representations are standard methods for implementing the transformations in computer graphics. In many systems, the transformations take place with respect to the coordinate origin. If we want to apply transformations with respect to some other coordinate point, then the successive transformations will take place. In doing so, the matrix representations are used and this method makes things clear and easy. The sequence of the transformations will take effect on the result and it will not be the same correct result.

VI. OTHER TRANSFORMATIONS

Basic transformations such as translation, scaling and rotation are included in the graphics package. Some packages also include other transformations which are not used that much but are useful. Two such transformations are shearing and reflection. Reflection is a rigid motion transformation but shearing is a not-rigid motion transformation.

A. *SHEARING*

It is a transformation which distorts the shape of the object such that the object was made of a number of layers and these layers have slid over each other. It is called a shear. There are two shear transformations X-Shear and Y-Shear.

In X-Shear transformation, the Y coordinates are preserved and changes are made to the X coordinates, which causes the vertical line to tilt either left or right. The transformation matrix for X-Shear can be represented as –

$$X_{sh} = \begin{bmatrix} 1 & 0 & 0 \\ shx & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$X' = X + Sh_X \,.\, Y$$

$$Y' = Y$$

In Y-Shear transformation, the X coordinates are left unchanged and Y coordinates are changed, which causes the horizontal line to tilt. The transformation matrix for Y-Shear can be represented as –

$$Y_{sh} \begin{bmatrix} 1 & shy & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Y' = Y + Sh_Y \,.\, X$$

$$X' = X$$

VII. REFLECTION

It is a transformation that produces the mirror image of the object. The mirror image of the object will be produced relative to an axis of reflection by rotating the object at 180 degrees along the reflection axis. The axis of reflection can be chosen anywhere in the x-y plane.

In Reflection about the line y=0 or the x-axis, this transformation keeps the x values the same but flips the y values of the coordinate positions. Similarly, in the reflection about the y-axis, the values of y coordinates remain the same but flips the x values of the coordinate positions.

VIII. COMPOSITE TRANSFORMATIONS

We can set up a matrix for any number or sequence of transformations as a composite transformation matrix by calculating the matrix product for the individual transformations. It is often referred as concatenation or composition of matrices. It is a combination of two or more transformations performed after each other. Sometimes, this results in a single transformation.

If a transformation of a plane T1 is performed and followed by a transformation T2 of an another plane, then the result can be expressed as a single transformation T which is the combination of T1 and T2 in the same order. This can be written as T=T1*T2. Similarly, we can apply a number of transformations, one after another to get the desired result. A combined matrix –

$$[T][X] = [X][T1][T2][T3][T4][T5]\ldots\ldots$$

Where $[T_i]$ is any combination of any kind of transformation.

In composite transformation, the sequence of the transformations is important. The change in the sequence will result in a different solution. [T1]*[T2] is not equal to the [T2]*[T1]. The purpose of the composite transformations is to gain efficiency by applying a single composed transformation to a point instead of applying a number of transformations.

IX. CONCLUSION AND FUTURE WORK

In this paper, we have reviewed the 2D transformations in computer graphics and how these transformations are done. We have discussed the difference between these transformations and how they are applied. These transformations are the basis of computer graphics and also used in 3D graphics, modeling, and animation. We have discussed the methods and techniques of implementing these transformations in an efficient way. In future, other transformations which are not much used will be discussed. There is always a scope of finding a better method of doing things and we will try to find a more efficient and faster method to solve these transformations with the use of least resources than the existing ones.

REFERENCES

1. Durand Cutler, 'Transformations', Massachusetts Institute of Technology, September 2008.
2. John Pile Jr., '2D Graphics Programming for Games', New York, NY: CRC Press. ISBN 1466501898, May 2013.
3. John E. Howland, 'Representing 2D Transformations as Matrices', Department of Computer Science, Trinity University, 2001.
4. Wayne Carlson, 'A Critical History of Computer Graphics and Animation, Ohio State University, 2003.
5. John Peddie, 'The History of Visual Magic in Computers', 'How Beautiful Images Are Made in CAD, 3D, VR, and AR,Springer, ISBN 978-1447149316, pp. 101, 2013.
6. Marc Levoy, '2D transformations', Introduction to Computer Graphics, Stanford University, 1996.
7. B. Stenseth, '2D transformations', Department of Information Technology, Ostfold University College, 1998.
8. David Blythe, 'Advanced Graphics Programming Techniques Using OpenGL', Siggraph 1999.
9. David Rogers, 'Procedural Elements for Computer Graphics', McGraw Hill, ISBN-13: 978-0070535480, 1998.
10. Donald Hearn and M. Pauline Baker, 'Computer Graphics', Prentice-Hall, ISBN 978-81-775-8765-4, 1994.
11. Anon., 'What is Computer Graphics?', Cornell University Program of Computer Graphics, April 1998.

DOI: 10.15680/IJIRCCE.2016. 0409028