# Technical Report Writing On

## Image Transformation Tools

**Submitted By**

Name: ARPAN SEAL
Department: Computer Science & Engineering
Semester: 6th
Roll Number: 16900121180

Department of Computer Science Engineering
Academy of Technology
Aedconagar, Hooghly-712121
West Bengal, India

- ## Abstract:

Digital image transformation is a subcategory or field of digital image processing having few real-life applications. It allows a much wider range of algorithms to be applied to the input data and can avoid problems. Since images are defined over two dimensions (perhaps more) image transformation may be modelled in the form of multidimensional systems.

- ## Introduction:

Transformation is a function. A function that maps one set to another set after performing some operations.

Consider this equation,

$$G(x,y) = T\{f(x,y)\}$$

In this equation,

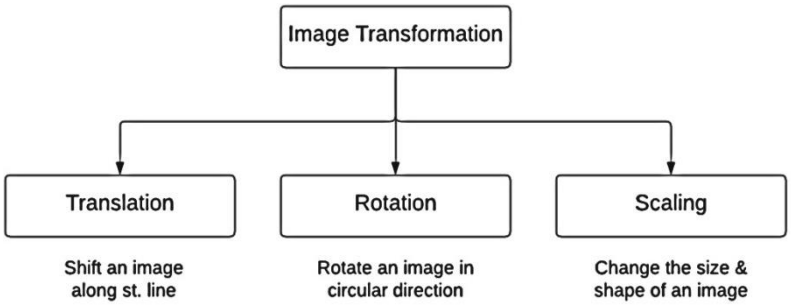
- F(x,y) = input image
- G(x,y) = processed image
- T = transformation function

This relation between input image and the processed output image can also be represented as,

$$s = T(r)$$

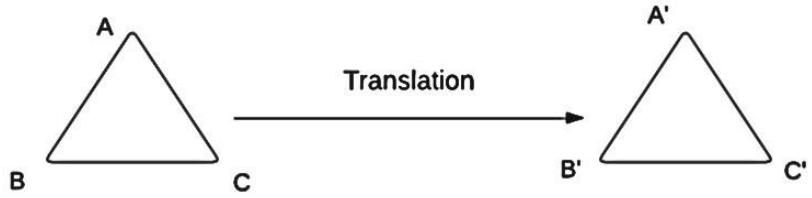
Where r is actually the pixel value or gray level intensity of f(x,y) at any point. And s is the pixel value or gray level intensity of g(x,y) at any point.

- ## Theory:



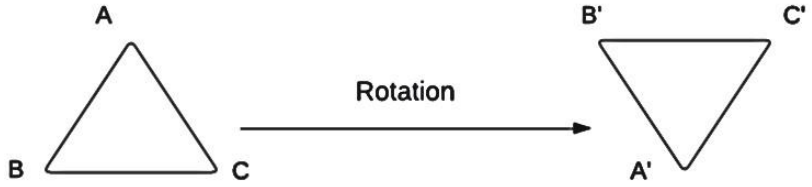

**Translation:**

**Image Translation** refers to the conversion of one image representation to another image representation. The goal is to learn the transformation between the input and output images.




**Rotation:**

**Image Rotation** is a basic transform of Euclidean geometry. It preserves distances and angle between points. If we rotate a continuous figure by α degree, and then by –α degrees, the result is identical to the initial figure in the continuous space.



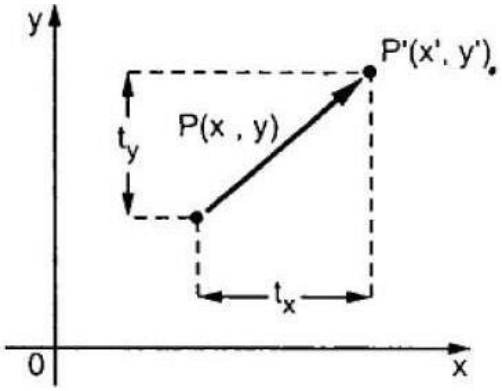

**Scaling:**

**Image Scaling** performs a geometric transformation which can be used to shrink or zoom the size of an image. Scaling is used to change the visual appearance of an image, to alter the quantity of information stored in a scene representation.


## ➢ Methodology:

**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').




From the above figure, you can write that − X'

= X + $t_x$

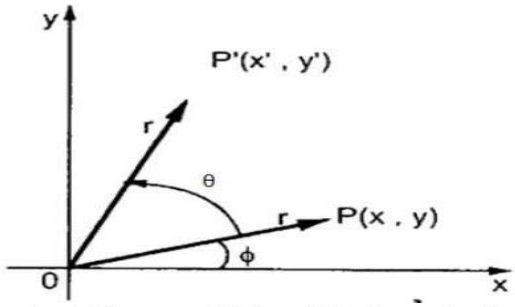
$Y' = Y + t_y$

$P' = P + T$

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

**Rotation:**

In rotation, we rotate the object at a particular angle $\theta$ from its origin. From the following figure, we can see that the point P(X,Y) is located at angle Ø from the horizontal X coordinate with distance r from the origin. Let us suppose you want to rotate it at the angle θ. After rotating it to a new location, you will get a new point P'(X',Y').




Using standard trigonometric the original coordinate of point P $X, Y$ can be represented as –

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

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

Same way we can represent the point P' $X', Y'$ as –

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

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

Substituting equation 1 & 2 in 3 & 4 respectively, we will get

$$x' = x\cos\theta - y\sin\theta$$

$$y' = x\sin\theta + y\cos\theta$$

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

P' = P . R

Where R is the rotation matrix

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

The rotation angle can be positive and negative.

For positive rotation angle, we can use the above rotation matrix. However, for negative angle rotation, the matrix will change as shown below −

$$R = \begin{bmatrix} cos(-\theta) & sin(-\theta) \\ -sin(-\theta) & cos(-\theta) \end{bmatrix}$$

$$= \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} (\because cos(-\theta) = cos\theta \; and \; sin(-\theta) = -sin\theta)$$

**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 with the scaling factor to get the desired result.

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 –

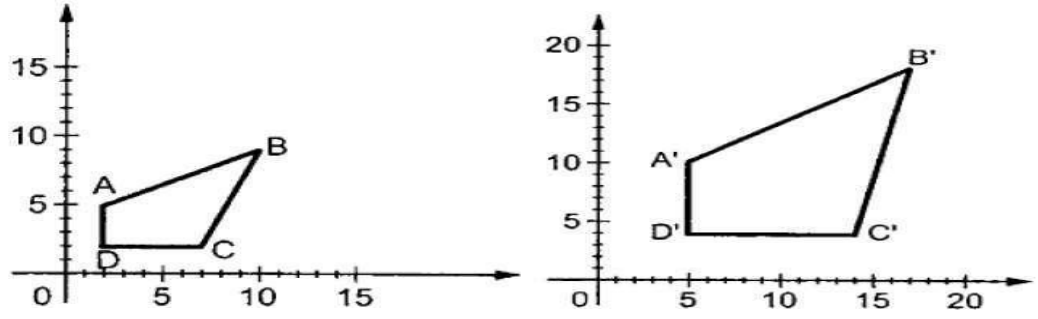
$$X' = X.S_X Y' = Y.S_Y$$

The scaling factor SX, SY scales the object in X and Y direction respectively. The above equations can also be represented in matrix form as below –

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

$$P' = P.S$$

Where S is the scaling matrix. The scaling process is shown in the following figure.



If we provide values less than 1 to the scaling factor S, then we can reduce the size of the object. If we provide values greater than 1, then we can increase the size of the object.

## Application:

Transformation means changing some graphics into something else by applying rules. We can have various types of transformations such as translation, scaling up or down, rotation, shearing, etc. When a transformation takes place on a 2D plane, it is called 2D transformation.

Transformations play an important role in computer graphics to reposition the graphics on the screen and change their size or orientation.

In the 2D system, we use only two coordinates X and Y but in 3D, an extra coordinate Z is added. 3D graphics techniques and their application are fundamental to the entertainment, games, and computer-aided design industries. It is a continuing area of research in scientific visualization.

Furthermore, 3D graphics components are now a part of almost every personal computer and, although traditionally intended for graphics-intensive software such as games, they are increasingly being used by other applications.

## Conclusion:

To perform a sequence of transformation such as translation followed by rotation and scaling, we need to follow a sequential process−

- Translate the coordinates
- Rotate the translated coordinates
- Scale the rotated coordinates

To shorten this process, we must use 3×3 transformation matrix instead of 2×2 transformation matrix. To convert a 2×2 matrix to 3×3 matrix, we must add an extra dummy coordinate W.

In this way, we can represent the point by 3 numbers instead of 2 numbers, which is called **Homogenous Coordinate system**. In this system, we can represent all the transformation equations in matrix multiplication. Any Cartesian point P(X,Y) can be converted to homogenous coordinates by P'($X_h$,$Y_h$,h).

- **References:**


0. Digital Image Processing, Fourth Edition, R. C. Gonzalez, R. E.Woods
1. https://www.tutorialspoint.com/dip/image_transformations.htm
2. https://www.tutorialspoint.com/computer_graphics/2d_transformation.htm
3. https://www.tutorialspoint.com/computer_graphics/3d_computer_graphics.htm