Chapter 3

Introduction to Graphics

Transformations

3.1 Introduction

The most important 2D graphics transforms are translation, rotation, scaling. Other transforms that are often applied to objects include reflection and shear. Graphics transforms using 2D cellular automata are discussed in chapter 6. In this chapter we introduce various methodolgies for performing these transformations which are generally incorporated into graphics packages [Roger90, Hearn95, Plast86].

There are two complementary points of view for describing object movements. 1) Geometric transformation, in which the object itself is moved relative to a stationary coordinate system or background and 2) Coordinate transformation, in which the object is held stationary while the coordinate system is moved relative to the object. In some situations both the methods are employed. Let us describe these systems briefly.
3.2 Geometric Transformation

An object in the plane is considered as a set of points. Every point \( P \) has coordinates \((x, y)\) and so the object is the sum total of all its coordinate points. If the object is manipulated in its orientation, size or shape, it can be regarded as a new object, all of whose coordinate points \( P' \) can be obtained from the original point \( P \) by the application of geometric transformation.

3.2.1 Translation

A translation is applied to an object by repositioning it along a straight path from one coordinate location to another. A two-dimensional point is translated by adding translation distances, \( t_x \) and \( t_y \), to the original coordinate position \((x, y)\) to move the point to a new position \((x', y')\). The new object point \( P'(x', y') \) is found by applying the transformation \( T_v \) to \( P(x, y) \) (Fig: 3.1)

\[
P' = T_v(P)
\]

and is given by

![Fig: 3.1](image-url)
CHAPTER 3. INTRODUCTION TO GRAPHICS TRANSFORMS

where \( x' = x + t_x \) and \( y' = y + t_y \).

3.2.2 Rotation

In rotation, the object is rotated by angle \( \theta \) about the origin. (Fig: 3.2)

The convention is that the direction of rotation is counterclockwise if \( \theta \) is positive and clockwise if \( \theta \) is negative. The transformation of rotation \( R_\theta \) is

\[
P' = R_\theta(P)
\]

where \( x' = x \cos(\theta) - y \sin(\theta) \)

and \( y' = x \cos(\theta) + y \cos(\theta) \)

3.2.3 Scaling

A scaling transformation alters the size of an object. Positive constants \( S_x \) and \( S_y \) are used to describe changes in length with respect to the \( x \) direction and \( y \) direction, respectively. The scaling transformation \( S_{S_x, S_y} \) is given by

\[
P' = S_{S_x, S_y}(P)
\]
where \( x' = S_x x \) and \( y' = S_y y \). Fig: 3.3 shows scaling transformation with scaling factors \( S_x = 2 \). If both scaling constants have the same value \( S \), the scaling transformation is said to be homogeneous. Further, if \( S > 1 \), it is called Zoom Out (magnification) and for \( S < 1 \), Zoom In (reduction).

![Fig: 3.3](image)

### 3.2.4 Inverse Geometric Transformations

Each geometric transformation has an inverse, which is described by the opposite operation performed by the transformation.

- Translation: \( T_v^{-1} = T_{-v} \), or translation in the opposite direction.
- Rotation: \( R_{\theta}^{-1} = R_{-\theta} \), or rotation in the opposite direction.
- Scaling: \( S_{S_x, S_y}^{-1} = S_{1/S_x, 1/S_y} \).

### 3.3 Co-ordinate Transformations

Suppose there are two coordinate systems in the plane. The first system is located at origin \( o \) and has coordinate axes \( xy \). The second coordinate system is located at origin \( o' \) and has coordinate axes \( x'y' \) (Fig: 3.4). Now each point in the plane has two coordinate descriptions: \((x, y)\) or \((x', y')\), depending
on which coordinate system is used. If the second system is thought of as arising from a transformation applied to the first system, then it is said that a coordinate transformation has been applied.

3.3.1 Translation

If the $xy$ coordinate system is displaced to a new position, the coordinates of a point in both systems are related by the translation transformation $T_v$:

$$(x', y') = T_v(x, y)$$

where $x' = x - t_x$ and $y' = y - t_y$.

3.3.2 Rotation

The $xy$ system is rotated by an angle $\theta$ about the origin (Fig: 3.5). Then the coordinates of a point in both systems are related by the rotation transformation $R_\theta(x, y)$:
\[(x', y') = \bar{R}_\theta(x, y)\]

where \(x' = x \cos(\theta) + y \cos(\theta)\) and \(y' = -x \sin(\theta) + y \cos(\theta)\).

### 3.3.3 Scaling

Suppose that a new coordinate system is formed by leaving the origin and coordinate axes unchanged, but introducing different units of measurement along \(x\) and \(y\) axes. If the new units are obtained from the old units by a scaling of \(S_x\) units along the \(x\) axis and \(S_y\) units along the \(y\) axis, the coordinates in the new system are related to coordinates in the old system through the scaling transformation \(\bar{S}_{S_x, S_y}\):

\[(x', y') = \bar{S}_{S_x, S_y}(x, y)\]

where \(x' = (1/S_x)x\) and \(y' = (1/S_y)y\). Fig: 3.6 shows coordinate scaling transformation using scaling factors \(S_x = 2\) and \(S_y = 1/2\).
3.3.4 Inverse Coordinate Transformations

Each coordinate transformation has an inverse which can be found by applying the opposite transformation.

Translation: \( T_v^{-1} = T_{-v} \), translation in the opposite direction.

Rotation: \( R_\theta^{-1} = R_{-\theta} \), rotation in the opposite direction.

Scaling: \( S_{x',y'}^{1/s_x,1/s_y} \).

3.4 Composite Transformations

More complex geometric and coordinate transformations, than what is described above, can be obtained by using the process of composition of functions. For example, such operations as rotation about a point other than the origin or reflection about lines other than the axes can be constructed from the basic transformations.

**Example 3.1:** Magnification of an object while keeping its center fixed. Let the geometric center be located at \( C(h,k) \). Choosing a magnification factor \( S > 1 \), we construct the transformation by performing the following sequence of the transformations.
1. translate the object so that its center coincides with the origin.

2. scale the object with respect to the origin.

3. translate the scaled object back to the original position.

The required transformation $S_{S,C}$ can be formed by composition

$$S_{S,C} = T_{v}^{-1} \cdot S_{S,S} \cdot T_{v}$$

By using composition, one can build more general scaling, rotation and reflection transformations.

### 3.4.1 Matrix description of the basic transformations

The use of homogeneous coordinates and homogeneous transformations ensure that all the above mentioned operations can be represented in matrix form. The transformation is of the form

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$$

It is well known that any 2D transforms can be implemented as a combination of the following transforms.

* $\text{translate}_x$: translate in the $x$ direction by a factor $T_x$

$$G_{T_x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & 0 & 1 \end{bmatrix}$$

* $\text{translate}_y$: translate in the $y$ direction by a factor $T_y$
CHAPTER 3. INTRODUCTION TO GRAPHICS TRANSFORMS

\[ G_{T_y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & T_y & 1 \end{bmatrix} \]

\textit{scale}_x: \text{ scale in the } x \text{ direction by a factor } S_x

\[ G_{S_x} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

\textit{scale}_y: \text{ scale in the } y \text{ direction by a factor } S_y

\[ G_{S_y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

\textit{rotate}_\theta: \text{ rotate through an angle } \theta

\[ G_{R_\theta} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

3.5 Conclusion

In this chapter, we briefly discussed the methodologies for performing the basic graphics transformations. All these operations are in general incorporated in software packages. We have made an attempt to build some hardware components to make these basic operations effective, using 2D cellular automata as a vehicle. Thus, as can be seen in chapter 6, we have been successful in applying certain 2D CA algorithms for getting certain fundamental graphics
operations and we can equip ourselves for the design of an application specific graphics processor.