3D GEOMETRIC TRANSFORMATION AND VISUALIZATION

1. Objectives

1. To understand the concept of **3D geometric transformations**.
2. To implement translation, scaling, and rotation in 3D space.
3. To represent 3D objects using coordinate systems.
4. To apply homogeneous coordinate representation in 3D transformations.
5. To visualize original and transformed 3D objects using graphical tools (e.g., Python/Matplotlib).
6. To observe the effect of different viewing angles on 3D projections.

2. Theory

Introduction

3D Geometric Transformation refers to the mathematical operations applied to three-dimensional objects to change their position, size, or orientation in space. These transformations are widely used in computer graphics, animation, CAD, gaming, and simulations.

In 3D space, a point is represented as:

P(x, y, z)

Using **homogeneous coordinates**, it is represented as:

P(x, y, z, 1)

Homogeneous coordinates allow translation, scaling, and rotation to be expressed using matrix multiplication.

Types of 3D Transformations

1. Translation

Translation moves an object from one position to another without changing its shape or orientation.

If a point P(x,y,z)P(x,y,z) is translated by (tx,ty,tz)(tx,ty,tz), the new point becomes:

$x' = x + t_x$
$y' = y + t_y$
$z' = z + t_z$

Translation matrix:

$$T = \begin{bmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Scaling

Scaling changes the size of the object.

If scaling factors are (Sx,Sy,Sz)(Sx,Sy,Sz):

$x' = S_x x$
$y' = S_y y$
$z' = S_z z$

Scaling matrix:

$$S = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Rotation

Rotation changes the orientation of an object around an axis.

Rotation about X-axis:

$$Rx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathbf{cos}\theta & -\mathbf{sin}\theta & 0 \\ 0 & \mathbf{sin}\theta & \mathbf{cos}\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about Y-axis:

$$Ry = \begin{bmatrix} \mathbf{cos}\theta & 0 & \mathbf{sin}\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\mathbf{sin}\theta & 0 & \mathbf{cos}\theta & 0 \\ 0 & & 0 & 1 \end{bmatrix}$$

Rotation about Z-axis:

$$Rz = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Visualization in 3D

Visualization is the graphical representation of 3D objects on a 2D screen. It involves:

- Projection (Perspective or Orthographic)
- Viewing transformation
- Rendering

In Python, libraries like Matplotlib (mpl_toolkits.mplot3d) are commonly used to visualize 3D objects and transformations.

3. Algorithm

Algorithm for 3D Translation

1. Start.
2. Define the 3D object vertices.
3. Input translation values (tx, ty, tz).
4. Construct the translation matrix.
5. Multiply each vertex by the translation matrix.
6. Obtain new transformed coordinates.
7. Display original and translated object.
8. Stop.

Algorithm for 3D Scaling

1. Start.
2. Define object vertices.
3. Input scaling factors (Sx, Sy, Sz).
4. Construct scaling matrix.
5. Multiply vertices by scaling matrix.
6. Get scaled coordinates.
7. Display original and scaled object.
8. Stop.

Algorithm for 3D Rotation

1. Start.
2. Define object vertices.
3. Choose axis of rotation (X, Y, or Z).
4. Input rotation angle θ.
5. Construct corresponding rotation matrix.
6. Multiply vertices by rotation matrix.
7. Obtain rotated coordinates.
8. Display original and rotated object.
9. Stop.