

Algorithms

1. Construct a Cube and Draw it Using Matplotlib

Algorithm: Cube Construction and Plotting

Step 1: Define the 8 vertices of a unit cube in 3D space using homogeneous coordinates $(x, y, z, 1)$

Step 2: Store the vertices in a matrix for easy manipulation

Step 3: Define cube edges by pairing appropriate vertices

Step 4: Use matplotlib 3D plotting (Axes3D) to draw lines connecting vertices

Step 5: Display the cube

2. Apply Scaling, Rotation, and Translation and Display Both Original and Transformed Cubes

Algorithm: 3D Cube Transformation

Step 1: Input scaling factors S_x, S_y, S_z

Step 2: Form scaling matrix S

Step 3: Multiply S with cube vertices \rightarrow get scaled cube

Step 4: Input rotation angle θ (degrees \rightarrow radians)

Step 5: Form rotation matrix R for Z-axis rotation

Step 6: Multiply R with scaled cube \rightarrow get rotated cube

Step 7: Input translation values t_x, t_y, t_z

Step 8: Form translation matrix T and apply to rotated cube \rightarrow get final transformed cube

Step 9: Plot both original and transformed cubes together

3. Try Different Viewing Angles Using `ax.view_init()`

Algorithm: Changing Cube Viewing Angles

Step 1: Set elevation angle $elev$ using `ax.view_init(elev, azim)`

Step 2: Set azimuth angle $azim$

Step 3: Render cube after changing viewing angles

Step 4: Observe how the cube's projection changes without modifying the cube itself

Step 5: Repeat for multiple angle combinations to understand perspective

4. Implement Rotations About X-axis, Y-axis, and Compare with Z-axis

Algorithm: Rotation Along Different Axes

Step 1: Input rotation angle θ (degrees \rightarrow radians)

Step 2: Create rotation matrices:

- R_x for rotation about X-axis
- R_y for rotation about Y-axis
- R_z for rotation about Z-axis

Step 3: Apply each rotation separately to cube vertices

Step 4: Plot cubes rotated about X, Y, and Z axes side by side

Step 5: Compare orientations of cubes after rotation

Step 6: Observe the effect of each axis on cube geometry