Rotaciones y Traslaciones en el Espacio 3D

ALDAIR SIERRA FONTALVO,ELIMELEC RUIZ QUINTERO,LEISER ANGARITA MEL

March 20, 2025

1 Introducción

Este documento presenta un análisis de las rotaciones y traslaciones en el espacio tridimensional. Se utiliza Python y la librería SymPy para representar las rotaciones de un punto en el espacio. El propósito de este ejercicio es ilustrar cómo la rotación de un punto cambia dependiendo del orden de las matrices de rotación. Además, se incluye una traslación de un punto para visualizar cómo se mueve en el espacio.

2 Matrices de Rotación

Las matrices de rotación son fundamentales para describir cómo un punto en el espacio 3D se rota alrededor de los ejes coordenados. Para una rotación en torno al eje X, la matriz de rotación es:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

De forma similar, la rotación sobre el eje Y es representada por la matriz:

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

Finalmente, la rotación sobre el eje Z se define como:

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Estas matrices permiten rotar cualquier vector en el espacio tridimensional.

3 Rotaciones en el Espacio

Supongamos que tenemos un punto $P_0 = [1, 1, 0]$ en el espacio. Al aplicar una rotación sobre el eje X con un ángulo de $\theta = 180° = \pi$ radianes, la nueva posición del punto es calculada como:

$$P_0' = R_x(\pi) \cdot P_0$$

Después de realizar la rotación, el punto resultante será:

$$P_0' = [1, -1, 0]$$

Si se aplica una rotación de $45° = \frac{\pi}{4}$ radianes, obtenemos:

$$P_0'' = R_x\left(\frac{\pi}{4}\right) \cdot P_0 = [1, 1, -0.707]$$

Esto demuestra cómo las rotaciones afectan la posición de un punto en el espacio.

4 Evaluación del Orden de las Rotaciones

El orden en que aplicamos las rotaciones afecta el resultado final. Para evaluar este efecto, consideremos tres diferentes órdenes de rotación para el punto P_0:

1. Rotación en el orden $X \to Y \to Z$
2. Rotación en el orden $Z \to Y \to X$
3. Rotación en el orden $Y \to Z \to X$

Para cada caso, calculamos el nuevo punto después de aplicar las rotaciones en el orden especificado. El resultado final puede ser diferente dependiendo del orden en que aplicamos las rotaciones.

5 Traslación

Una traslación de un punto en el espacio consiste en moverlo en una dirección determinada por un vector de traslación. Si tenemos un punto $P_0 = [1, 1, 0]$ y un vector de traslación $v = [0.5, -0.5, 0.5]$, el nuevo punto P_{traslado} se obtiene como:

$$P_{\text{traslado}} = P_0 + v$$

El resultado de la traslación será:

$$P_{\text{traslado}} = [1.5, 0.5, 0.5]$$

6 Código Python

A continuación se muestra el código Python utilizado para realizar las rotaciones y traslaciones descritas anteriormente.

```
import numpy as np
from sympy import *
import matplotlib.pyplot as plt

# Función para dibujar un punto en 3D
def dibujar_punto(punto, label, x_lim=[0, 1], y_lim=[0, 1], z_lim=[0, 1]):
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    ax.scatter(punto[0], punto[1], punto[2], color='b', label=label)
    ax.set_xlabel('X')
    ax.set_ylabel('Y')
    ax.set_zlabel('Z')
    ax.set_title('Espacio 3D')
    ax.set_xlim(x_lim)
    ax.set_ylim(y_lim)
    ax.set_zlim(z_lim)
    ax.legend()
    plt.show()

# Matrices de rotación
theta = symbols("theta", real=True)
R_x = Matrix([[1, 0, 0], [0, cos(theta), -sin(theta)], [0, sin(theta), cos(theta)]])
R_y = Matrix([[cos(theta), 0, sin(theta)], [0, 1, 0], [-sin(theta), 0, cos(theta)]])
R_z = Matrix([[cos(theta), -sin(theta), 0], [sin(theta), cos(theta), 0], [0, 0, 1]])

# Punto inicial
P0 = np.array([1, 1, 0])

# Rotaciones y traslaciones
theta_val = np.pi  # 180 grados
R_p0 = np.array(R_x.subs({theta: theta_val})).astype(np.float64)
P0_rotado = np.dot(R_p0, P0)
dibujar_punto(P0_rotado, "P0 rotado", y_lim=[-1, 1])

# Traslación
vector_traslacion = np.array([0.5, -0.5, 0.5])
P_traslado = P0 + vector_traslacion
dibujar_punto(P_traslado, "P0 trasladado", x_lim=[-1, 1], y_lim=[-1, 1])
```

7 Conclusiones

Las rotaciones y traslaciones son operaciones fundamentales en geometría computacional y gráficos 3D. Al estudiar el orden de las rotaciones, podemos entender cómo estas afectan la orientación de un punto en el espacio. Además, las traslaciones permiten mover puntos de un lugar a otro sin cambiar su orientación.