VR&AR Assignment1

- yn2273 Yuhe NIE

1. Code

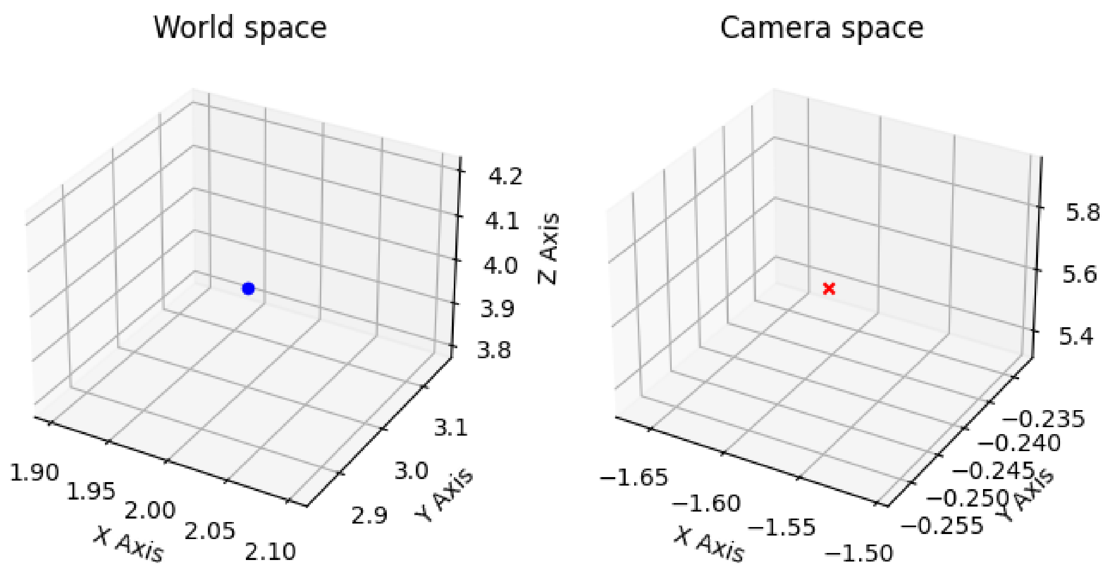
> https://colab.research.google.com/drive/1B6YL-KKUPgVcnjz3yKPiBPbMagt27SRf?usp=drive_link

```
import matplotlib.pyplot as plt
import numpy as np
import math

# In the beginning, the camera's local coordinate system coincides with the world coordinate system.
initial_camera_transformation = np.identity(4, dtype=np.float32)

# First rotate the camera around the z axis by 130 degrees, then move it by a translation of (dx, dy, dz) = (2.0, 3.0, 1.0), finally rotate it around the x axis by 30 degrees.
# TODO: compute the 4 x 4 matrix corresponding to each transformation described above and composite them into a single transformation matrix, i.e., the camera transformation matrix.
radius130 = math.pi*13/18
radius30 = math.pi/6
rotation1 = np.array([[math.cos(radius130), -1*math.sin(radius130), 0, 0],
            [math.sin(radius130), math.cos(radius130), 0, 0],
            [0, 0, 1, 0],
            [0, 0, 0, 1]
            ], dtype=np.float32)
translation = np.array([[1, 0, 0, 2],
            [0, 1, 0, 3],
            [0, 0, 1, 1],
            [0, 0, 0, 1]
            ], dtype=np.float32)
rotation2 = np.array([[1, 0, 0, 0],
            [0, math.cos(radius30), -1*math.sin(radius30), 0],
            [0, math.sin(radius30), math.cos(radius30), 0],
            [0, 0, 0, 1]
            ], dtype=np.float32)
current_camera_transformation = np.dot(rotation2, np.dot(translation, rotation1))

# Object position in the world space.
# TODO: compute the object's position in the camera space using the camera transformation matrix.
object_position_world = np.array([2, 3, 4], dtype=np.float32)
object_position_Cartesian = np.array([2, 3, 4, 1], dtype=np.float32)
object_Cartesian = np.dot(current_camera_transformation, object_position_Cartesian.T)
object_position_camera = np.array([object_Cartesian[0], object_Cartesian[1], object_Cartesian[2]], dtype=np.float32)

# TODO: visualize the object's position in the world space and the camera space.
fig = plt.figure(figsize=(8, 4))
```

```
ax1 = fig.add_subplot(121, projection='3d')
ax1.set_title("World space")
ax2 = fig.add_subplot(122, projection='3d')
ax2.set_title("Camera space")
ax1.scatter(*object_position_world, color='blue', marker='o')
ax2.scatter(*object_position_camera, color='red', marker='x')
ax1.set_xlabel("X Axis")
ax1.set_ylabel("Y Axis")
ax1.set_zlabel("Z Axis")
ax2.set_xlabel("X Axis")
ax2.set_ylabel("Y Axis")
ax2.set_zlabel("Z Axis")
plt.show()
```

2. Visualization Results and Calculation

The 3 coordinates f the given object in the camera space is $(-1.584, -0.245, 5.632)$



Assignment 1 Calculation Process

$$R_z(120) = \begin{vmatrix} \cos(130) & -\sin(130) & 0 & 0 \\ \sin(130) & \cos(130) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad T(2,3,1) = \begin{vmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$120 = \tfrac{2}{3}\pi$$

$$R_x(30) = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(30) & -\sin(30) & 0 \\ 0 & \sin(30) & \cos(30) & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$30 = \tfrac{\pi}{6}$$

$$M = R_x(30)\, T(2,3,1)\, R_z(120)$$

$$= \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} \cos(130) & -\sin(130) & 0 & 0 \\ \sin(130) & \cos(130) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$= \begin{vmatrix} -0.643 & -0.766 & 0 & 2 \\ 0.663 & -0.557 & -0.5 & 2.098 \\ 0.383 & -0.321 & 0.866 & 2.366 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad \text{point world} = \begin{vmatrix} 2 \\ 3 \\ 4 \\ 1 \end{vmatrix}$$

$$\text{point camera} = \begin{vmatrix} -0.643 & -0.766 & 0 & 2 \\ 0.663 & -0.557 & -0.5 & 2.098 \\ 0.383 & -0.321 & 0.866 & 2.366 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 2 \\ 3 \\ 4 \\ 1 \end{vmatrix} = \begin{vmatrix} -1.584 \\ -0.245 \\ 5.632 \\ 1 \end{vmatrix}$$