Enhanced visualization: assignment 2

Transformations in 2D/3D

I. Theoretical part

In the first part of this assignment you are asked a few theoretical questions about transformations in 2D/3D. Each time you are asked to give the matrix representing a transformation, you might consider that we use homogeneous coordinates.

A. Linear and affine transformations.

1. Is the origin invariant under linear transformations?
2. Is the origin invariant under affine transformations?
3. Show that affine transformations preserve parallelism between lines.

B. Rigid body transformations.

1. Write the general matrix form of a rigid body transformation.
2. Are rigid body transformations invertible? If yes, show why this is true.
3. What is the inverse in matrix form of a general rigid body transformation?[1]

[1] Hint: you can use the fact that the inverse of a block partitioned matrix $M = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}$ is $M^{-1} = \begin{bmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{bmatrix}$

where

$\mathbf{E} = (\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}$
$\mathbf{F} = -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}$
$\mathbf{G} = -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}$
$\mathbf{H} = (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}$

C. Scaling. Let us consider a scaling transformation in 3D:

1. Show that scaling is not a rigid body transformation.
2. Show that uniform scaling, *i.e.* same scaling in each axis, preserves angles between vectors?
3. Is the scaling transformation invertible? If yes, give its inverse.

D. Rotation. Let us consider a rotation transformation in 3D:

1. Show that rotations around a point (2D) and rotations around an axis (3D) are rigid body transformations.
2. What can you say about its invertibility from the results of the previous exercises?
3. Suppose that the center of a 3D object is given by a vector $\mathbf{u}$, how do you change the orientation of the object without changing its position?

II. Practical part

In this part you are going to code some transformations in homogeneous coordinates. Download the Jupyter notebook "assignment_2.ipynb" from *moodle*.

A. Transformation matrices. Code the corresponding matrices for the three following transformations in homogeneous coordinates:

1. Non uniform scaling.
2. Translation.
3. Rotation around a given axis.

Insert the code in the notebook in the places indicated by " ******* ".

You can test your library on the cloud of points given in the file "teapot.txt"[2] which you can find in *moodle*. The coordinates of the points can be loaded with the *numpy* command

```
tp = np.loadtxt('teapot.txt')
```

as you can find in the notebook. If you use the 3D version of scatter plot you should see the same as in Figure 1.

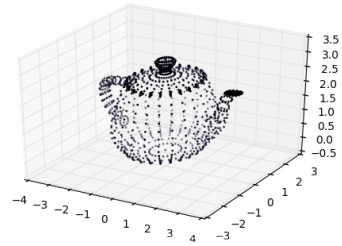

Figure 1: 3D scatter plot of a teapot.

[2] Source: `https://people.sc.fsu.edu/~jburkardt/data/ply/teapot.ply`

B. Animation.

1. Generate a first simple animation where the teapot is scaled with unitary scaling and it is rotating horizontally[3] above the position $[6.0\,0.0\,0.0]^T$. You need just to fill the missing code lines within "update_graph".
2. Generate an animation with two teapots of different colors in two different positions on the x axis, rotating horizontally around their respective centers in opposite directions.
3. You can also test your animation code with the Stanford bunny, which has a much larger number of points. For this you can load the points from the file "bunny.txt"[4] which is also given in *moodle*. Use small markers for the scatter plot ($s = 0.001$).

[3] The horizontal plane is the xy plane in *matplotlib*.

[4] Source: `https://graphics.stanford.edu/data/3Dscanrep/`