Homework 1

Instructions

This homework covers the basics of rigid $SE(3)$ transforms and coordinate systems. It contains a set of written problems (Problem 1) and programming problems (Problem 2). It has a total of 100 points. Please refer to the course website for the homework due date.

For Problem 1, compile your answers in PDF and title it `hw1.pdf`. For Problem 2, you will directly be editing the Python file `transforms.py` provided by us.

Submission: 1) hw1.pdf 2) transforms.py. **Please do not include any other files.** When you are done, zip the directory, rename it to `SUID_hw1.zip`, and upload it to Canvas before the homework deadline.

Problem 1 *(50 points)*

1.1 *(10 points)*

Given the following matrices:

$$A = \begin{bmatrix} 1 & 0 & 0 & 20 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 30 \\ 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 3 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 3 \\ 0 & -1 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 3 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

1. (5 *point*) Determine if each matrix belongs to the $SE(3)$ group of valid homogeneous transforms. Justify your answers and show any relevant calculations.
2. (5 *point*) For all transform $T \in SE(3) \cap \{A, B, C, D\}$ compute the inverse transform T^{-1}. Verify $T^{-1}T = I$, $TT^{-1} = I$. Show any relevant calculations.

1.2 *(10 points)*

For this problem, no need to show your work, but give a list of the steps or expressions you evaluated to get the answer.
Given the following $SE(3)$ poses:

$$
{}^0T_1 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1T_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -3 \\ 0 & 0 & 1 & -10 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

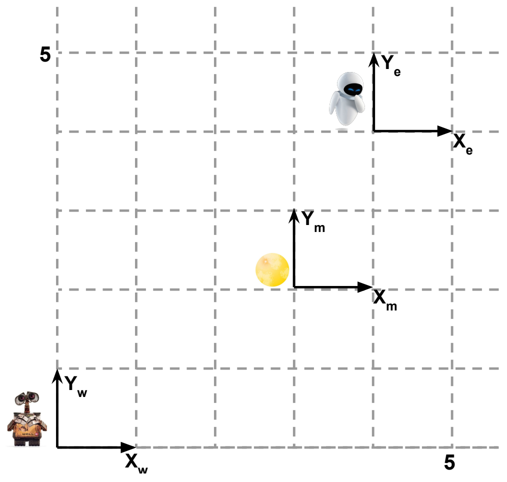
1. (5 *point*) The coordinate of a 3D point p, is given as ${}^2p = [2, 4, 6]^T$ in the coordinate space of frame 2. Compute ${}^1p$, the coordinate of p in frame 1.

2. (5 *point*) Given the transformation matrix ${}^0T_2$, which describes the transformation of a point from the coordinate frame of frame 2 to that of frame 0, do the following:

   - Decompose the transformation into its corresponding Euler angles (rotations about specific axes) and translation components.
   - Write a few sentences explaining this transformation in terms of the rotations (about which axes the rotations occur) and translations (along which axes the translations occur). Be sure to specify the order of the rotations and the direction of the translations.

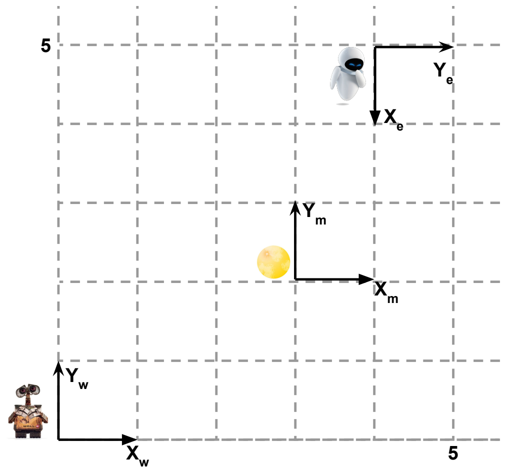
1.3 *(30 points)*

1.3.1 Here we have three coordinate frames. Eve's reference frame e, Wall-e's reference frame w, and the moon's reference frame m.



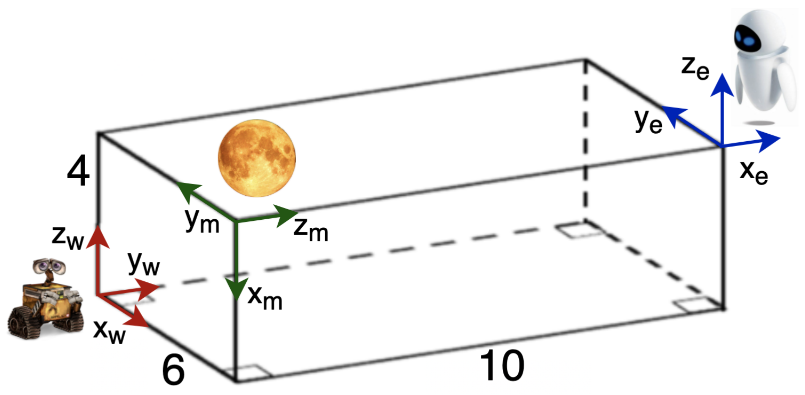
1. (3 points) What is the moon's position in Wall-e's reference frame? ${}^{w}P_m$
2. (3 points) What is the moon's position in Eve's reference frame ${}^{e}P_m$?
3. (2 points) Write out the transformation matrix that changes the coordinate frame from Wall-e to Eve ${}^{e}T_w$? To check your work, please verify that ${}^{e}P_m = {}^{e}T_w {}^{w}P_m$

1.3.2 Now Eve's reference frame is changed as shown in the Figure below. Let's repeat the above steps:



1. (3 points) What is the moon's position in Wall-e's reference frame? ${}^{w}P_m$
2. (3 points) What is the moon's position in Eve's reference frame ${}^{e}P_m$?
3. (2 points) Write out the transformation matrix that changes the coordinate frame from Wall-e to Eve ${}^{e}T_w$? To check your work, please verify ${}^{e}P_m = {}^{e}T_w {}^{w}P_m$
4. (2 points) What is Eve's pose in the Wall-E reference frame? Write out both symbol ${}^{??}T_{??}$ and numerical representation.

1.3.3 Now let's move to the 3D! Wall-E and Eve's coordinate is shown below



1. (2 points) What is the moon's position in Wall-e's reference frame?
2. What is Eve's position in Wall-e's coordinate frame?
3. (2 points) Write out the transformation matrix that changes the coordinate frame from Wall-e to Eve ${}^{e}T_{w}$?
4. (4 points) Write the following poses ${}^{e}T_{m}$, ${}^{m}T_{w}$.
5. (2 points) Show that ${}^{e}T_{m}\ {}^{m}T_{w} ={}^{e} T_{w}$.

Problem 2

Getting started

In this problem, you will implement lightweight transform functions in Python.

To start we will install Python interpreter and dependencies using miniforge to avoid any version issues. Please follow the installation instructions for your system (Unix-like or Windows). You can also download the corresponding Mambaforge-OS-arch.sh/exe file at miniforge and execute the downloaded script. After installation and initialization, launch a **new** terminal and run the following command **inside the unzipped homework zip file**.

```
mamba env create -f environment.yaml
```

This will create a new environment named "hw1", which can be activated by running

```
mamba activate hw1
```

To deactivate an active environment, use

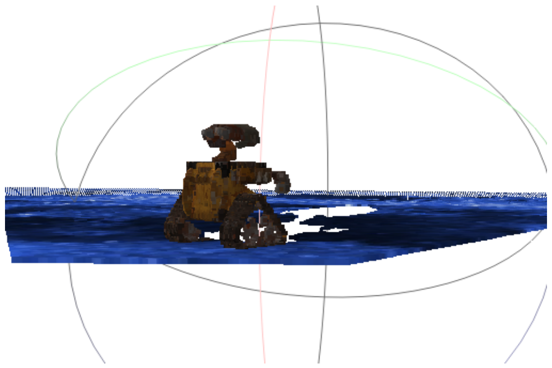
```
mamba deactivate
```

Implementations (50 points)

In the first part of this problem, you will implement parts of `transforms.py`. This file contains some generic transform definitions. Detailed descriptions of function inputs and outputs are given in `transforms.py`. Check out the comments for each function. Please implement the following functions:

```
    transform_concat(...) -- 10 pt
    transform_point3d(...) -- 10 pt
    transform_inverse(...) -- 10 pt
    depth_to_point_cloud(...) -- 20 pt
```

We have provided the function `test_rgbd_image_to_points` that will read an RGB-D image convert it into point cloud and save it as `test.ply`. Downloading a viewer like Meshlab to make sure that your *point clouds* look as expected. Expected output:



Testing and Debugging

To help you validate your implementation, we have provided `transforms_test.py` that include basic unit tests for your transform functions. To run unit tests, execute the command:

```
python transforms_test.py
```

Note that these tests are meant to help you and are not necessarily exhaustive. You are encouraged to do further testing if you feel certain cases are not properly tested. Expected output for passing all the tests:

```
---------
Ran 6 tests in 0.317s
OK
```