Middle East Technical University Department of Computer Engineering

CENG 460

Introduction to Robotics for Computer Engineering

Spring 2021-2022

Assignment 1 - Pose Representations

Due date: April 16th 2022, Saturday, 23:55

1 Theory (65pts)

1. An arrangement of various objects (a cube, a prism of an equilateral right triangle, a rectangular plane folded at half of its length by 30 degrees) are shown in Figure 1,2. Calculate the following homogeneous 3D relative poses by hand (^iT_j represents pose of j w.r.t i):

   (a) (5pts) ^AT_C.

   (b) (5pts) ^BT_D.

   (c) (7.5pts) ^CT_D.

2. (a) (7.5pts) Find the 3d homogeneous transformation T that aligns the line $[a,\ b,\ c]^\top t + [d,\ e,\ f]^\top,\ t \in \mathcal{R}$ to the global z axis.

   (b) (5pts) Find the transformation Q that represents a rotation of a point p_0 by θ around this line (*Hint & Simplification: You do not have to explicitly calculate Q. Just leave it as a function of T*).

   (c) (5pts) Write down the set of homogeneous vectors with $Qp = p$ without explicitly calculating them. Show that the set you proposed satisfies this equation.

   (d) (5pts) Using your answers and ideas in 2-(a, b), show that if two transformations $T_1,\ T_2$ represent a rotation around the same line, $T_1T_2 = T_2T_1$.

3. Write down the shortest angle difference (see Figure 3) function for:

   (a) (5pts) For two angles $\theta_1,\ \theta_2 \in [-\pi, \pi)$ (measured from the positive x axis, counter-clockwise) in a piecewise form.

   (b) (5pts) Two quaternions that represent a rotation around global z axis by $\theta_1,\ \theta_2$ in a non-piecewise form.

4. For a 2D rotation matrix $R(\theta)$ prove that $R^n(\theta) = R(n\theta)$ for:

   (a) (5pts) $n \in \mathcal{N}$ (*Hint: Prove it for $n = 0$ ($R^0 = R^{-1}R$). Then assume it is true for R^n and show it would be true for R^{n+1} if your assumption held*).

   (b) (bonus - 2.5pts) $n \in \mathcal{R}$ (*Hint: Apply eigen-decomposition to R, remember that $exp(PDP^{-1}) = Pe^DP^{-1}$ and review this link*).

5. Prove that any 3D rotation matrix R can be represented with

(a) (5pts) some invertible matrix A and $\theta \in [0, 2\pi)$, s.t. $R = AR_z(\theta)A^{-1}$ (*Hint: Think of the vector R rotates about*).

(b) (bonus - 2.5pts) an arbitrary rotation matrix R' and some invertible matrix A s.t. $R = AR'A^{-1}$.

6. (5pts) Prove that quaternion multiplication is commutative if and only if their vector parts are linearly dependent (*Hint: What does linear dependence between 3D vectors entail in terms of cross and dot product?*).

2 Programming (35pts)

In this part, you will design a transformation query system similar to the ROS tf2 package, that will take some input 3D homogeneous transformations ${}^iT_j$,with indices $i, j \in \{0..10\}$ which represents the transform of j w.r.t i. The system will also incorporate a query functionality, in which the user supplies the indices for the system to calculate the transform from existing data. Note that the previously inserted transforms might not be sufficient to calculate the transform the user requested. In this case, the function should throw an error.

Fill in the MATLAB function stubs provided with the assignment. There is also an example script executing these functions. Do not rely on the example and hardcode transformations to your code. The example will be switched during grading.

The example file uses the Corke's Robotics toolbox. To install it, simply download the `rvctools` directory from MATLAB drive using this link. Then, add the following to your startup script `startup.m` in your MATLAB startup folder:

```
addpath(genpath(<absolute_path_to_rvctools>));
startup_rvc;
```

3 Other Specifications

- All of the rotation matrices are to be assumed right handed.
- You can complete the programming assignment without relying on any toolbox, including the robotics toolbox. If you require a toolbox utility, please publicly ask in ODTUCLASS. We may or may not allow it.
- Your programming assignment will be reviewed in a glass-box fashion, therefore please be diligent with your comments in your implementation (you do not have to put comments in every line, just put comments to places that you think it is complicated for the reader to understand. Your comments should provide information about *why* rather than *what*. Avoid "`a = a.*b % element-wise multiply a and b`" sort of comments).
- Please ask your questions regarding the assignment publicly in ODTUCLASS, unless your question reveals part of your solution. In that case, send a mail to `onem@ceng.metu.edu.tr`.
- **This is an individual assignment. Using any piece of code that is not your own is strictly forbidden and constitutes as cheating. This includes friends, previous homeworks, or the Internet. The violators will be punished according to the department regulations.**

4 Submission

Submit your solution as a compressed zip file name `hw1_eXXXXXXX.zip` to ODTUCLASS, containing:

- a pdf scan of your hand-written answers or a latex formatted pdf document named `theory_hw1_eXXXXXXX.pdf`.
- .m function files comprising the filled in stubs and additional helper functions.

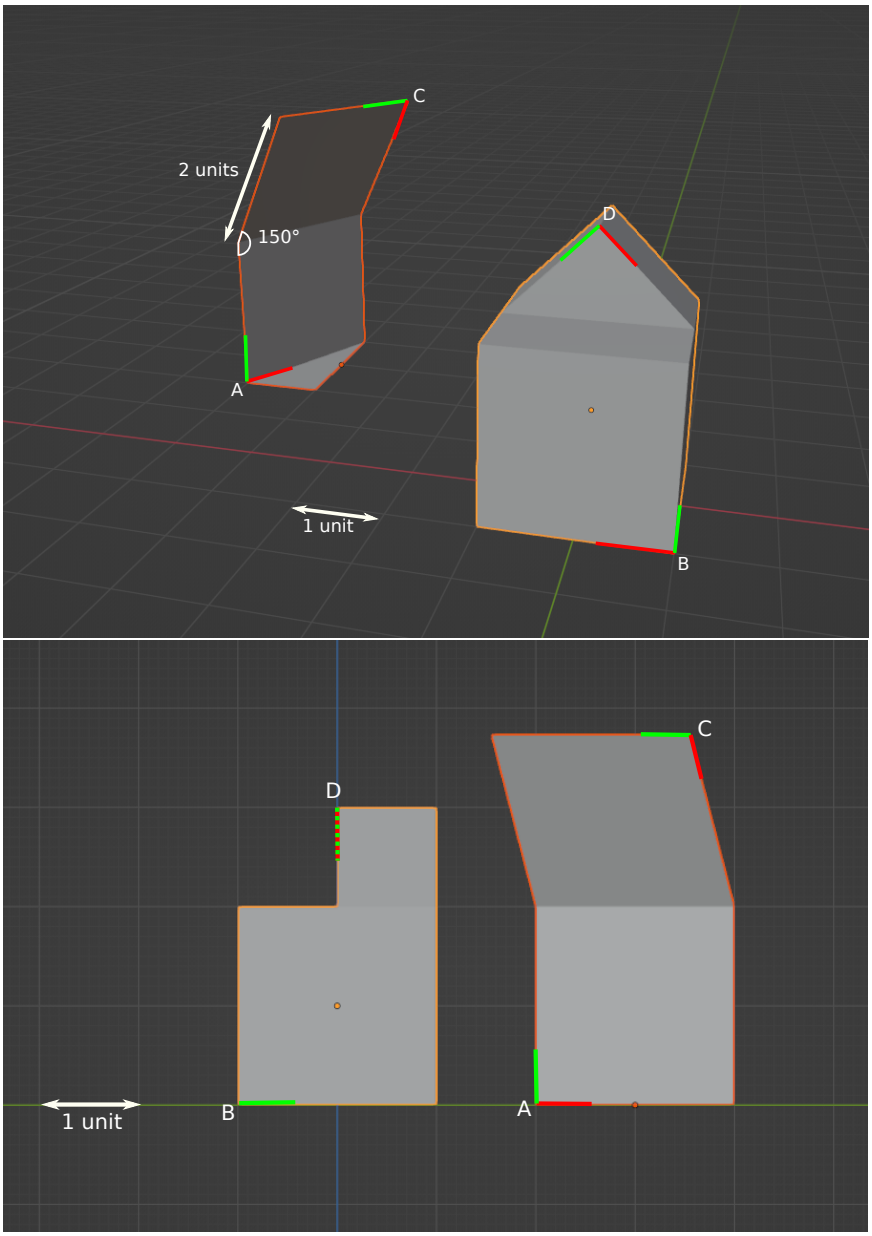

Figure 1: Arrangement in objects in theory Q-1, from perspective and right side view. Red, green represent local x, y directions respectively.

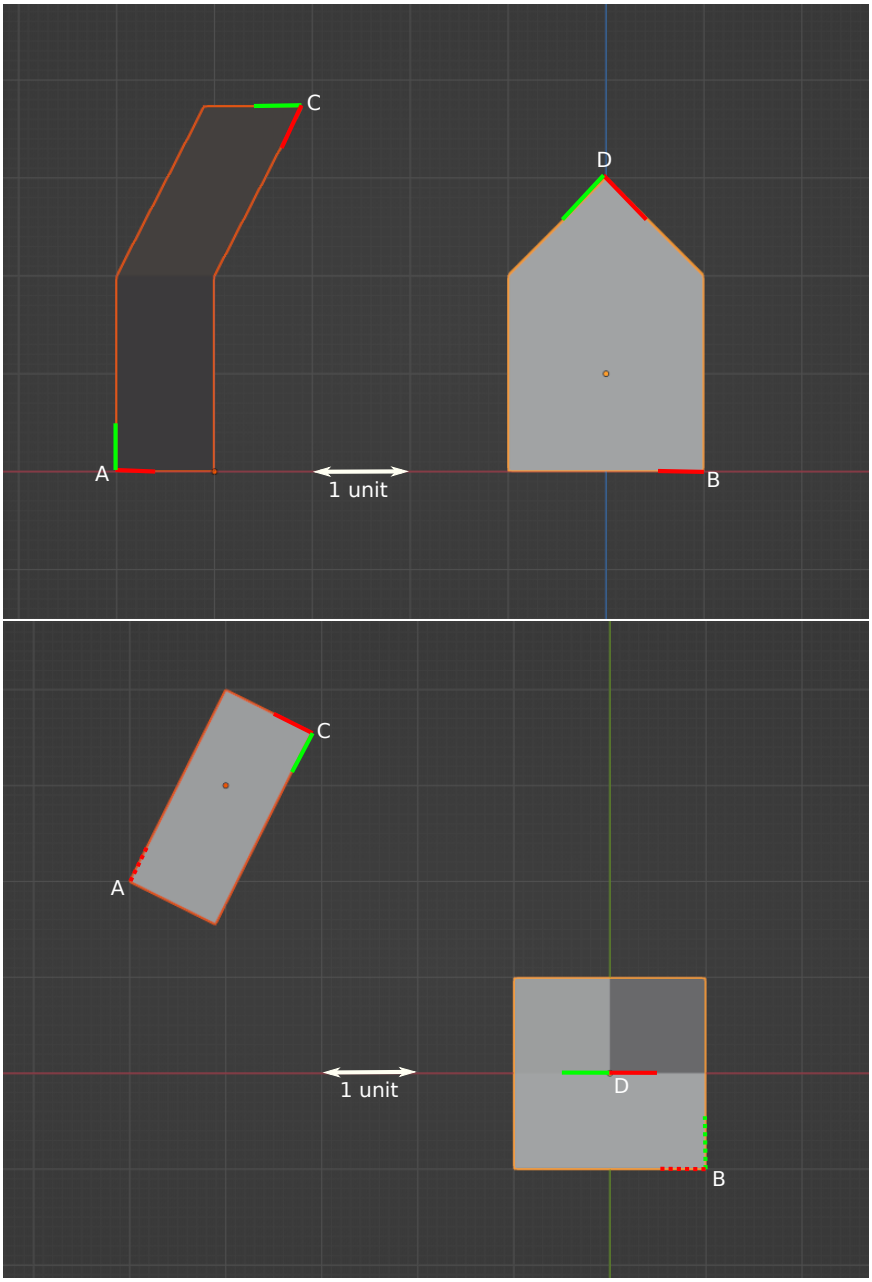

Figure 2: Arrangement in objects in theory Q-1, from front and top view. Red, green represent local x, y directions respectively.

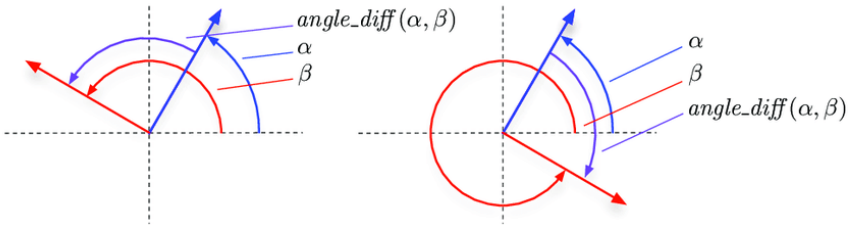

Figure 3: Shortest angle difference function.