SLAM Homework 1:
Pose Graph Optimization

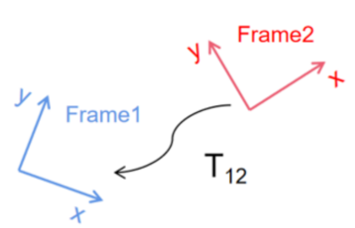
Prof. Laurent Kneip

March 11th 2024

Acknowledgements:

1. Deadline: **2024-3-26 23:59:59**
2. No handwritten homework is accepted.
3. Your homework should be submitted in PDF format and packed with your code, and the naming format of the file is *studentID-name-hw1.*
4. Please submit your homework through email to *daizj2022@shanghaitech.edu.cn* with the subject line "studentID-name-hw1"

This homework extends the discussion on pose graph optimization. Assume we have a robot that is moving on a circular trajectory of 1m radius. In this homework, the relative pose estimations are already given, and your task will be to optimize the overall trajectory of the robot. The robot measures one scan every 30 degrees and makes a full turn, which is why there are 13 poses and 12 relative pose measurements (the last frame is at an identical position than the first one). The definition of the relative pose is shown in Figure 1. The parameters are added in the below form to a transformation matrix that then can help to transform points from the next to the previous frame.



$$T_{12} = \begin{bmatrix} cos(\Delta\theta_{12}) & -\sin(\Delta\theta_{12}) & \Delta x_{12} \\ \sin(\Delta\theta_{12}) & cos(\Delta\theta_{12}) & \Delta y_{12} \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 1: Definition of relative pose.

There are five tasks for you, the last one is optional:

1) Install evo library. The evo library is a powerful tool for evaluating and analyzing SLAM algorithms. It provides an easy-to-use interface for computing various metrics (ATE, RPE). To make use of it, you can check the evo wiki or directly append **-h/–help** to your command to seek available arguments. This tool will be utilized frequently in forthcoming assignments, so it is worthy to pay a bit of efforts on it at the beginning of your SLAM journey. (10%)

2) Complete dead reckoning using the relative poses. You may assume that the first frame has identity rotation and zero translation. What can you observe? In particular, is there anything to say about the thirteenth pose? (10%)

3) Now drop the last frame and interpret the final relative pose (i.e. column 12) as the geometrically verified relative pose between the 12th and the first frame (found through loop closure), i.e. $T_{12,1}$. The relative pose constraints may be considered as edges in a graph where each node corresponds to one of the absolute poses (the newly added edge is illustrated in orange in the Figure below). Use pose graph optimization to optimize the absolute poses. Record your time and accuracy (by using evo). Describe your observations and try to exploit the sparsity pattern for improved computational efficiency. (50%)

4) There are many third-party libraries available for implementing pose graph optimization. Such as g2o for C++ and PyPose for Python. Please read one of their official document and do pose graph optimization again. Meanwhile, compare the accuracy (by using evo) and time efficiency with yours in last Task. (30%)

5) Visualize and explain what the information matrix looks like. (bonus 20%)

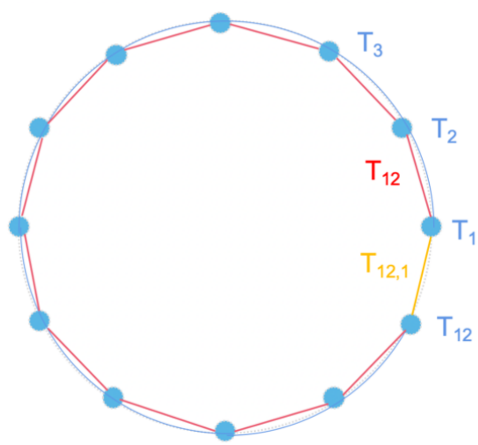


Figure 2: Constraints between adjacent frames.

Notes:

1) The data is given to you in the form of a 3x12 matrix of relative poses in which each column is given by the parameters $(\Delta\mathrm{x}, \Delta\mathrm{y}, \Delta\theta)^{\mathrm{T}}$ that define the relative transformation between adjacent frames (column 1 is the transformation between frame 1 and 2 etc.).

2) Both C++ and Python are acceptable. You can only use third-party libraries for matrix computation (i.e. NumPy and Eigen) and visualization (i.e. evo or Open3D or Matplotlib).

3) Attach your implementation with pdf in the zip. In the package, you also need to include a file named README.txt/md to identify the function of each file. Make sure that your codes can run and are consistent with your homework. We would then arrange a meeting after the deadline in which we would ask each one of you to come in for 10 minutes to demonstrate your solution on your own computer.

4) If submitted after the deadline but still within 24hrs, a 50% penalty is applied. If submitted more than 24hrs after the deadline, a zero score will be given. In special case, please contact Prof Kneip.