SLAM Homework 3:
Scale-aware Pose Graph Optimization

Prof. Laurent Kneip

March 11th 2024

Instructions:

1. Deadline: **2024-5-5 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-hw3*.
4. Please submit your homework through email to *daizj2022@shanghaitech.edu.cn* with the subject line "studentID-name-hw3"

Unlike stereo or depth sensors, which directly provide depth information, using a monocular camera necessitates the triangulation operation to get depth from pairs of images, making it challenging to determine the true scale of the scene. In homework1, we have discussed the pose graph optimization, which typically does not explicitly address scale drift. It optimizes relative poses between robot poses (nodes) and constraints (edges) based on observed measurements without considering scale consistency. In this assignment, we'll incorporate scale drift-aware into pose graph optimization[1] and further explore the situation of scale jump-aware pose graph optimization[2].

The sequence that provided to you only consider the circle scenario. For the scale jump part, we made different segments on the circle. The scale between adjacent segments is unknown, which is caused by the scale jump.

And we also provide the Lie.h head file for you, so that you don't need to make wheels of lie group calculus and could be able to focus on the essential part of pose graph optimization.

There are two tasks for you:

- We assume that the relative transformation and scale between last pose and initial pose is accurate (it can be achieved by loop-closure). Besides this, there is a small scale drift between each camera poses. Please use the scale drift-aware pose graph optimization to refine those poses. (60%)

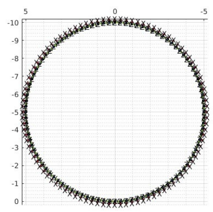

Figure 1: gt

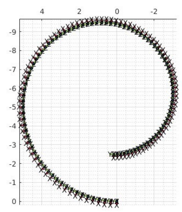

Figure 2: scale drift

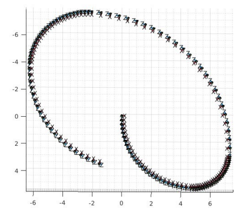

Figure 3: scale jump

- Based on task1, we assume that there is a scale jump in some relative pose transformations. Within our data, the scale jump happens with 3 and 4 times respectively. This issue commonly arises when a monocular camera loses tracking and needs to reinitialize, resulting in a discrepancy in scale magnitude between the current and previous movements. Please consider the scale jump situation into the pose graph optimization and verify through the rank calculation numerically that globally consistent scale can or cannot be recovered with different segments. (we recommend you to read [2] before starting it). (40%)

Notes:

1) The data description is written in data.md.

2) Both C++ and Python are acceptable. You can use the off-the-shelf libraries (e.g. ceres, g2o) to help you doing the pose graph optimization.

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.

References

[1] Hauke Strasdat, JoséMaria Montiel, and AndrewJ. Davison. *Scale Drift-Aware Large Scale Monocular SLAM.* Jan 2011. 1

[2] Runze Yuan, Ran Cheng, Lige Liu, Tao Sun, and Laurent Kneip. Scale jump-aware pose graph relaxation for monocular slam with re-initializations. 1, 2