Computer Vision Group
TUM Department of Informatics

Exercise Sheet 5

Topic: Pose Graph Optimization

Exercise 5.1: Pose Graph Optimization

In this exercise, you will implement a pose graph optimization for a direct RGBD-SLAM approach.

a) Extract the exercise archive to obtain the provided code. Download the fr2/desk sequence from the TUM RGB-D benchmark from the following website:
https://vision.in.tum.de/data/datasets/rgbd-dataset/download
The file formats are described here:
https://vision.in.tum.de/data/datasets/rgbd-dataset/file_formats

b) The file **`runDirectVO.m`** implements a keyframe based visual odometry front-end which is tracking the camera based on direct image alignment (as implemented in Exercise 3.2). Run the code on the provided dataset to estimate the camera trajectory. Check the code for the provided keyframe selection scheme.
Process the complete sequence and evaluate the relative pose error (RPE) using the provided evaluation tool in the TUM RGB-D benchmark (https://vision.in.tum.de/data/datasets/rgbd-dataset/tools). Plot the resulting trajectory using the absolute trajectory error (ATE) tool.

Note: Running the entire sequence will take some time.

The file **`runPGO.m`** implements a pose graph optimization based on the keyframe poses using the Gauss-Newton algorithm and left-multiplied increments on the poses.

c) Therefore, detect manually a loop closure between a keyframe at the end of the trajectory and the very first keyframe of the trajectory. Implement the estimation of the relative pose between the loop closure pair using the direct image alignment code in the **`runPGO.m`** script.

d) Initialize the state vector using the poses obtained from the VO front-end. Furthermore, create relative pose constraints between keyframes that are in successive order through direct image alignment, as well as the loop closure pair by implementing the corresponding residual calculations.

e) Implement the Gauss-Newton update set in the **`runPGO.m`** script, while keeping the pose of the first keyframe fixed. Make sure to use a left-multiplied pose increment.

f) Run the PGO implementation and evaluate the relative pose error (RPE) and absolute trajectory error (ATE) using the provided evaluation tool in the TUM RGB-D benchmark (https://vision.in.tum.de/data/datasets/rgbd-dataset/tools). Plot the resulting trajectory using the absolute trajectory error (ATE) tool. Compare your result with the direct visual odometry result from the previous exercise part.