SLAM Homework 2 - Pose graph optimization

Runze Yuan (yuanrz@shanghaitech.edu.cn)

October 19, 2021

1. Algorithm Pipeline

The goal of the homework is to implement graph optimization in trajectory and compare the result. Due to the noise of measurement or inaccurate estimate of poses, the error of our global pose will accumulate which causing drift. Graph optimization can distribute the error to every node and achieve an better result.

The algorithm is as below

Q1:

1. Given the relative poses between each two frames. Applying dead reckoning, the equation is

$$T_i = T_{i-1} * T_{i-1,i}$$

2. Define the state vector is s = $[x, y, \theta]$, the transformation from matrix to state is as below

$$x = T_{02}$$
$$y = T_{12}$$
$$\theta = atan2(T_{10}, T_{00})$$

The transformation from matrix to vector is defined as **t2v**(), then the global poses are **t2v**(T_i)

Q2:

Simply applying dead reckoning will causes large drift. Since we know the constraint that the robot will return to the starting point in the end, we can perform loop closure to optimize the poses.

Here we use g2o graph optimization library, we add totally 12 poses(nodes) and 12 relative poses measurement (edges) into the graph. The graph is shown as below.

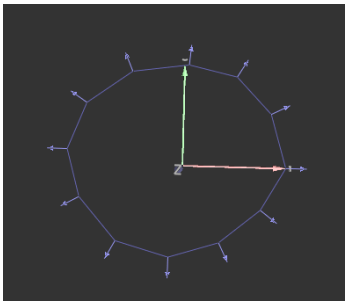

Figure 1: Loop closure

Q3:

In this Q3, given 10 more constraints, we add them into our graph as edges. The graph is show as above.

2. Usage

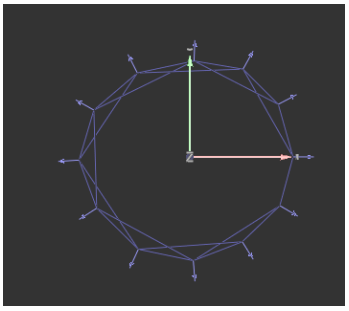

Figure 2: More constraints

1. Dependency:
   (a) ROS (version:noetic) (for visualization)
   (b) Eigen (for computation)
   (c) g2o (for graph optimization)
2. Compile: Create a directory named **icp_ws**, put the src directory into it. Then run **catkin_make -DCMAKE_BUILD_TYPE=Release**
3. Run: **source devel/setup.bash** and **rosrun posegragh posegragh_node path_to_data.txt path_to_T_everytwo.txt**

   For visualization, open rviz and add three path visualization ,
   (a) change the name of the topic to **"nonOptimizedPose"**, you will see the trajectory not optimized.
   (b) change the name of the topic to **"optimizedPose"**, you will see the trajectory only performed loop closure.
   (c) change the name of the topic to **"finalPose"**, you will see the trajectory optimized by more constraints.

   Also, you can ran the python script to visualize the trajectory and the errors.

3. Evaluation

In this section, we implement the motion model of the robot and get the ground truth trajectory. The comparison between each and the ground truth is shown as below.

Table 1: performance table

	rotation error(radian)	translation norm error
dead_reckoning	0.6175	1.3792
loop_closure	0.4847	0.4949
more_constraints	0.2359	0.6457

We can see, from the table, that the drift error decreases significantly after doing graph optimization. However, the translation error increases when we add more constraints to our model which indicates that two many inaccurate observation may have negative contribution to the model.

The total time cost is 0.002119 seconds, which is very fast.

Here is the 4 trajectories visualized. (dead_reckoning, loop_closure, more_constraints and ground truth)

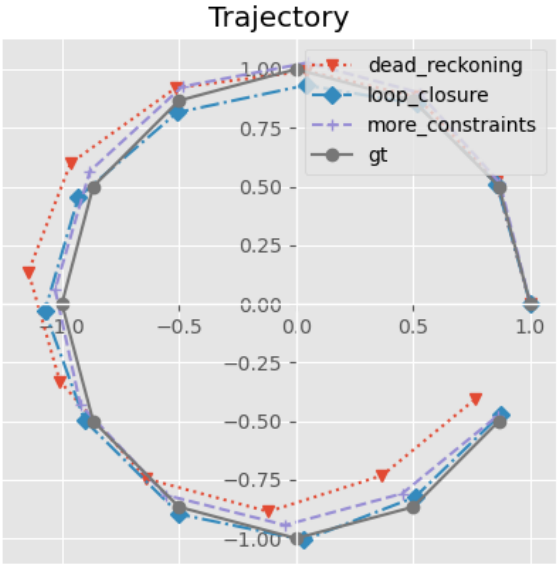

Figure 3: data

The information matrix is shown as below

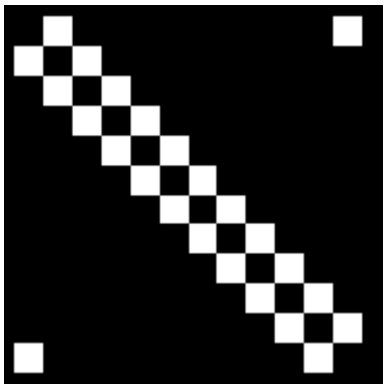

Figure 4: loop closure (12 measurements)

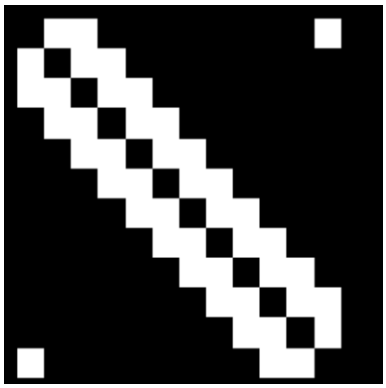

Figure 5: add more constraints (22 measurements)

The information matrix is a 36x36 matrix. The white space indicates a 3x3 identity matrix, and the black space indicates zeros matrix.