CS284 HW2 Report: Pose Graph Optimization

Hongchen Cao
2019533114
caohch1@shanghaitech.edu.cn

***Abstract*—This report covers a basic description of the implementation, performance, and Q&A of the pose graph opyimization in HW2.**

I. Project

A. Description of the pose graph optimization algorithm

The pose graph optimization algorithm can be summarized as follow:

ALGORITHM 1: Graph Optimization Algorithm

while $!converged$ **do**
  $(H, b) = buildLinearSystem(x)$
  $\Delta x = solveSparse(H\Delta x = -b)$
  $x = x + \Delta x$
end while
return x

I use *python* to implement the algorithm and visualize the result.

B. Structure of the project

CS284_hw2_data contains the raw data.
g2o_data contains the data in g2o format.
g2opy contains the 3rd library.
main.py is the implementation and visualization of the pose graph optimization algorithm.

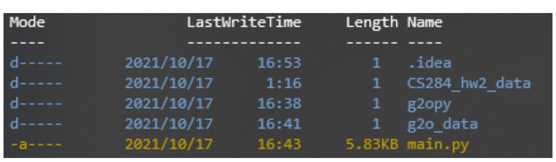


Fig. 1: Structure of the project

C. Dependencies

python $\geq$ 3.8.11
numpy $\geq$ 1.20.3

D. Instructions

For task k, run `python main.py --task_index=k`

II. Performance

All the results can be Visualized by run *main.py*.

A. Efficieny

The time consumed for Task1 is $0.51043s$;

The time consumed for Task2 is $0.51268s$;

The time consumed for Task3 is $0.53842s$;

The time calculated includes the process of reading raw data, the main optimization part, printing, ploting and so on.

B. Accuracy and Visualization

Accuracy

I take the ideal pose positions as reference(i.e., $[0, 0, 0], [0, 0, \pi/6], [0, 0, \pi/3], ..., [0, 0, 2\pi]$) and calculate the sum of error for $t_x, t_y, theta$ independently.

	t_x	t_y	$theta$
original	11.73150	7.28932	0.61755
task2	12.84017	7.15501	0.61756
task3	12.16079	7.46702	0.23592

TABLE I: Error for all the tasks

Visualization

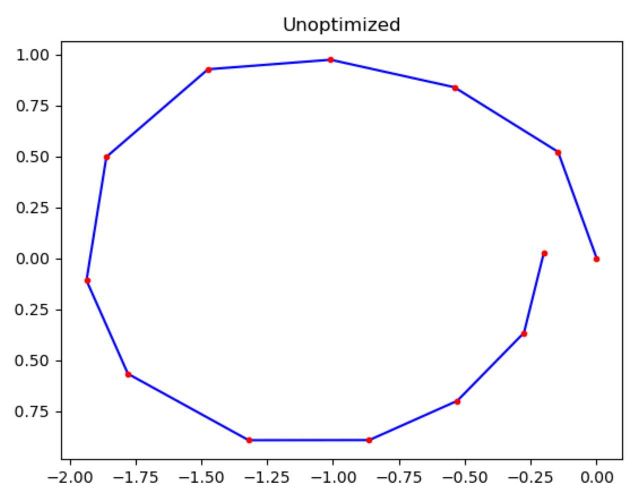
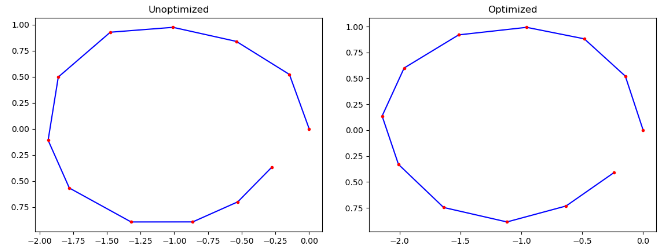
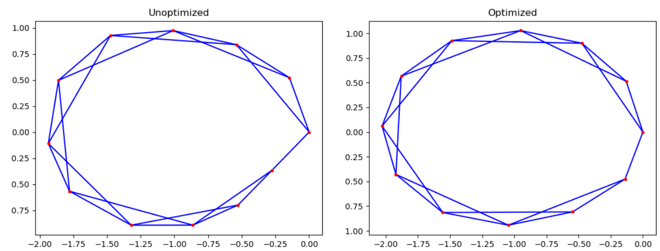


Fig. 2: Results for Task1



(a) Unoptimized (b) Optimized

Fig. 3: Results for Task2



(a) Unoptimized (b) Optimized

Fig. 4: Results for Task3

III. Q&A

Q1: For task1, what can you observe? In particular, is there anything to say about the thirteenth pose?

A1: As shown in Fig. 2, the distance between 13^{th} and 1^{st} pose is obvious (i.e., The Euclidean diatance is 0.20182). In other word, the cumulative error is large.

Q2: For task2, Describe your observations.

A2: Without the edge between 13^{th} and 1^{st} pose, the effect of pose graph optimization is weak. The interpret $T_{12,1}$ is $[-0.00247, 0.45857, 0.64978]$.

Q3: For task3, Describe your observations.

A3: The effect of pose graph optimization is obvious. The 12 absolute poses are close to the ideal circular trajectory of 1m radius.

Q4: Bonus questions: Visualize and explain what the information matrix looks like.

A4: I set the information matrix as identity in this project. The information matrix is the inverse of covariance matrix, so the visualization of it is Symmetrical.