CS284 HW3 Report: Bundle Adjustment

Hongchen Cao
2019533114
caohch1@shanghaitech.edu.cn

***Abstract*—This report covers a basic description of the implementation, performance, and observations on the results of 3 sub-tasks in HW4.**

I. Project

A. Description of the algorithm

The algorithm for task1 can be summarized as follow:

ALGORITHM 1: Gauss-Newton Bundle Adjustment

X = measured (the points in the image plane)
P = unknowns (the minimal pose and point corrections)
while reach iter num **do**
  $\Delta = (J^T J)^{-1} J^T e_0$
  $P = P + \Delta$
end while
return P

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

B. Structure of the project

CS284_hw4.pdf is the introduction for HW3.
utils.py contains some helper functions.
main.py is the implementation of 3 sub-tasks.

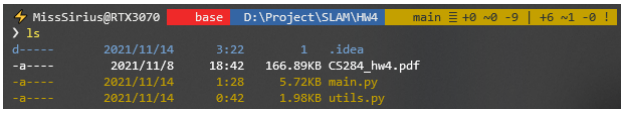


Fig. 1: Structure of the project

C. Dependencies

python == 3.6.13
matplotlib == 3.3.4
numpy == 1.19.5
scipy == 1.5.4

D. Instructions

Run `python main.py`

II. Performance

A. Efficieny

The time consumed for Task1 is $0.00099s$;
The time consumed for Task2 is $1.04320s$;
The time consumed for Task3 is $4.66053s$;
The time calculated includes the process of creating data, the main calculating part, printing, saving and so on.

B. Accuracy and Visualization

Accuracy

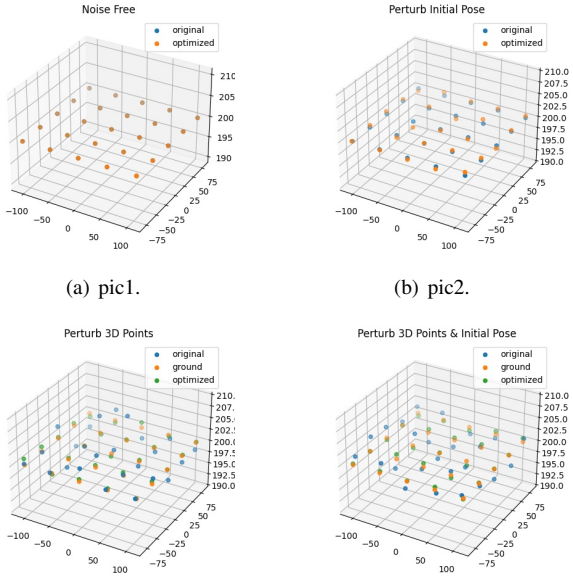
Table. I shows the results of different noise.

	Perturb Initial Pose	Perturb 3D Points	Perturb Both
3d-points	2.54823	6.13423	9.77302
Rotation	0.43352	0.66003	0.72231
Translation	4.79652	7.66132	11.2458

TABLE I: Error

Visualization

Fig. 2 shows the results of different noise.



(a) pic1. (b) pic2.

(c) pic3. (d) pic4.

Fig. 2: Visualization

III. Q&A

A. How to improve the results

To improve efficieny, we can use sparse bundle adjustment since Jacobian matrix has sparse block structure. To improve accuracy, we can use robust kernel function.