

CS284: Simultaneous Localization and Mapping

Homework 4, Fall Semester 2021

- Contact: lkneip@shanghaitech.edu.cn
- TA: cuili@shanghaitech.edu.cn
- Submission deadline: **23:59, 14th of November 2021** (please submit a zip via email)
- For general questions not directly related to the homework material, please first use Piazza.
- **Please reread our policies on academic integrity. It is a very serious matter, and any violations will be prosecuted to the fullest extent.**

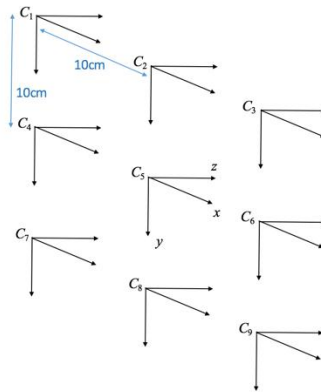
Task Description

➤ Task 1

Define a toy example for bundle adjustment. We assume the existence of a perspective camera with intrinsic parameters (no distortion)

$$\mathbf{K} = \begin{bmatrix} 500 & 0 & 320 \\ 0 & 500 & 240 \\ 0 & 0 & 1 \end{bmatrix}$$

Define 9 camera poses arranged in a 3x3 square grid with 10 cm baseline between neighboring views. All cameras are having identity orientation with respect to the reference view C_5 , and are shifted along the x and y directions w.r.t. the reference view. The arrangement of the views for the toy example is illustrated in the below figure.



Define 25 3D world points such that their image coordinates in the reference view C_5 are

x	50	185	320	455	590	50	185	320	455	590	50	185	320	455	590	50	185	320	455	590	50	185	320	455	590
y	40	40	40	40	40	140	140	140	140	140	240	240	240	240	240	340	340	340	340	340	440	440	440	440	440

and their depth along the principal axis in the reference view is 2m. Generate the noise-free measurements of these 25 3D points in all views.

➤ Task 2

We are aiming at an implementation and test of the bundle adjustment algorithm, which is why we are going to add noise to the data. Bundle adjustment requires two types of inputs. The first one is given by the 2D measurements, and the second one is given by an initial guess for the camera poses. Noise is added as follows:

- Start with the 2D measurements of the points in all views. Add noise to the x and y coordinates of all 2D measurements in the image plane by sampling from a normal distribution with a standard deviation of 0.25 pixels.
- Perturb the 3D points to generate an initial guess for the 3D structure by adding noise to all 3 coordinates. Again, this is done by sampling from a normal distribution, this time with 2cm standard deviation.
- Perturb the camera poses to generate an initial guess for the views. Once more, perturb each coordinate of each camera pose by sampling from a normal distribution of 2mm standard deviation. The orientation is perturbed by multiplying each view with a small rotation derived from a random Euler axis angle vector for which each coordinate is sampled from a normal distribution with a standard deviation of 0.002.

➤ Task 3

Make an implementation of bundle adjustment as taught in class. The following should be respected:

- You should make your own implementation of Gauss-Newton updates using the normal equations rather than using an off-the-shelf toolbox.
- You may work in normalized coordinates such that the camera matrix K does not appear in the optimization problem.
- You should optimize local changes of 3D world point positions and camera pose parameters rather than absolute parameters. You may choose the Euler axis-angle parameterization for the camera orientations.
- The core of the algorithm consists of implementing the Jacobian evaluation for each step of the optimization. You are free to choose numerical or analytical differentiation.
- You do not have to worry about robust kernels since the data contains no outliers. Furthermore, simple Gauss-Newton updates should be sufficient to approach the solution, no damping should be required.

- You do not need to implement a termination condition; you may simply run a fixed number of iterations.
- You may first try with zero noise to confirm that your algorithm is working properly. You may also try to only perturb the initial poses and 3D points, but add no noise to the measurements (again to confirm that the algorithm is running properly).

➤ Task 4 (Bonus): Implementation of the Schur complement trick.

Submission

Please submit a PDF report (not more than 3 pages!) along with the code to the TA email address before the deadline (put professor in CC). Your code and results should be the same with what you submitted in the email, then arrange for a time with the TA to demonstrate your code. Your report must include:

- A description of your implementation. This should cover a basic description of the algorithm. It should also include the physical structure of the program (i.e. file description, dependencies), and instructions on how to compile and use it (i.e. how to interpret the result).
- Add comments on the algorithm's ability to compensate for noise and reduce the errors of the initial guess with respect to ground truth. Add a figure that displays the initial arrangement as well as the result after optimization. Also indicate the reprojection errors before and after the optimization. You may try out different amounts of noise added to the data.
- Observations on the performance of your algorithm, both in terms of efficiency and in terms of accuracy. And thoughts on how to improve the results.

Please use "SLAM HW4 – Your name (Chinese Pinyin, e.g. Wang Xiaoming)" as the email-header when sending the email. Please pack all the files into a zip file, the structure should be:

[Your name] folder
 – Your name.pdf
 – [code] folder