Visual Navigation for Flying Robots
D. Cremers, X. Gao, V. Usenko
Summer Semester 2018

Computer Vision Group
Department of Informatics
Technical University of Munich

Exercise Sheet 5

Topic: Backend

Submission deadline: Sunday, 27.05.2018, 23:59 pm
Hand-in via email to visnav_ss2018@vision.in.tum.de

General Notice

The exercises should be done by yourself, but the final project should be done in teams of two to three students. We will use Ubuntu 16.04 in this lab course. It is already installed on the lab computers. If you want to use your own laptop, you will need to install Ubuntu by yourself.

Files related with the exercise will be placed in directories like 'paper/' or 'doc/'. Please read these materials before start answering the questions.

Exercise 1: Bundle Adjustment

The BAL (Bundle Adjustment in large) data set (`http://grail.cs.washington.edu/projects/bal/`) is a large BA dataset that provides camera and point initial values as well as observations. In this exercise you will implement a Bundle Adjustment program to optimize the data set. You should use Google Ceres or g2o to define the reprojection error (or your own vertex and edge if you use g2o), write the BA program on BAL. You can pick one sequence of BAL, run your BA, and give an optimized point cloud after the optimization converges. Please visit the BAL webpage in order to know the data format.

We don't provide the code framework for this exercise. You can find a Ceres tutorial regarding of how to implement the BA with Ceres, and for g2o there is also an example in the example/ directory.

Exercise 2: Photometric Bundle Adjustment

The BA used for the feature-based methods minimizes the reprojection error as an optimization target. In contrast, if we want to minimize the photometric error, we must formulate the direct BA or photometric BA. We have talked about how to use the direct method to estimate the camera pose in the previous lectures. Furthermore, the direct method can also be used to handle the entire Bundle Adjustment. Please derive the mathematical model of the direct BA and complete its Ceres or g2o implementation. Note that the parameterization used in this exercise is a bit different from DSO [1]. We parameterize each 3D point with x, y, z, whereas DSO

uses the inverse depth parameterization. The inverse depth model will lead to a more complicated graph model so we just start with a simpler one.

In this exercise we provide 7 images, saved as 0.png to 6.png. The initial position of each camera pose $\mathbf{T}_i$, in the form of $\mathbf{T}_{cw}$ is stored in the poses.txt file, where each line represents the pose of the camera in the same format as the previous exercises:

$$\text{time}, t_x, t_y, t_z, q_x, q_y, q_z, q_w.$$

We also provide a set of 3D points P for a total of N points. The initial coordinates of these points are $\mathbf{p}_i = [x, y, z]_i^{\mathrm{T}}$. Each point also has its own fixed gray value, which we describe with 16 numbers. The 16 numbers are the intensity values in 4x4 a small patch around this point, which we denote as $I(p)_i$. In other words, the intensity values are taken from $u-2, v-2$ to $u+1, v+1$ first iterating through v. We can assume that every point can be projected into every image, and we can compute the difference between the original patch and the one after projection. Then, the overall optimization objective function is:

$$\min \sum_{j=1}^{7} \sum_{i=1}^{N} \sum_{W} \|I(\pi(\mathbf{p}_i)) - I_j(\pi \mathbf{T}_j \mathbf{p}_i))\|_2^2, \tag{1}$$

which is minimizes the difference between the projection of every point in every image, where π is the projection function and W refers to the entire patch. Now please implement the PBA using the code framework in PBA.cpp. We only provide the g2o version, if you want to use Ceres, you can modify some codes in this file.

Submission instructions

A complete submission consists both of a PDF file with the solutions/answers to the questions on the exercise sheet and a ZIP file containing the source code that you used to solve the given problems. Note all names of your team members in the PDF file. Make sure that your ZIP file contains all files necessary to compile and run your code, but it should not contain any build files or binaries. Please submit your solution via email to `visnav_ss2018@vision.in.tum.de`.

References

[1] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *arXiv preprint arXiv:1607.02565*, 2016.