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

Computer Vision Group
Department of Informatics
Technical University of Munich

Instructions on Final Project

Submission deadline: Monday, 02.07.2018, 23:59 pm
Hand-in via email to visnav_ss2018@vision.in.tum.de

General Notice

The final project should be completed by a team of up to 3 students. You can choose one research topic you are interested in, which should be related to visual SLAM and navigation. The topic of the final project is open so there is no standard answer, and the score is based on the degree of completion of the submitted report. If a team has more than one student, the division of work must be stated in the report. Along with the submitted report, you also need to make a 15 minutes presentation about the results.

The final project has four weeks to complete. We provide an example of research topic here, and if you want to choose you own topic, please send us an email stating which problem you want to solve in the project.

Example topic: stereo visual odometry in Kitti dataset

Kitti's visual Odometry dataset provides left and right camera images as well as calibration information. It contains several sequences (also with lasers but in this job we do not use the laser data). Part of the sequence's ground-truth trajectory is available (called training sequences), and the other is hidden and used as a test dataset to evaluate the results. You can upload your trajectory estimation for the test part to the Kitti website. The system calculates the difference with the ground-truth trajectory and gives the score and rank.

Since we have introduced all the aspects of visual SLAM, you can implement a stereo visual odometery based on the existing algorithms (in OpenCV or in our exercises). Please implement a stereo visual odometry based on what you have learned, then run it on the provided sequence and compare the trajectory with the ground-truth.

The following are some hints about the implementation. This is only for instruction, and does not necessarily to be used in your work.

1. You can first implement a frame-by-frame odometry, which only estimates the motion between the current image and the previous image, and then compose

them into a complete camera trajectory. This is a simple and straightforward approach. You may find the error will accumulate very quickly and make the trajectory not accurate after a short time.

2. Next, you can extract the keyframes from the trajectory and perform a bundle adjustment on the keyframe cameras. Because BA reduces the reprojection error, it can effectively control the drift. Meanwhile, you need a mechanism to manage all map points and keyframes. You may define part the cameras and map points as active, and leave the others fixed in order to make your program run in real-time.

3. Finally, test your program's performance on Kitti, such as the trajectory accuracy, run time and so on.

You may find some interesting questions during the implementation. Actually most the SLAM systems use the same theory, i.e., the nonlinear optimization or filters. What makes them different is the engineering part like how to choose the keyframes and build the BA graph, how to set the parameters in feature matching or optical flow, etc. The discussion on these engineering problem is meaningful. If you have any interesting findings in the implementation process, please explain in the report.

Submission instructions

A complete submission of the final project consists of a PDF file with the solutions/answers to the questions, and also the slides of the presentation. 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`.