Answer Sheet 5

Topic: Real-Time Visual Odometry

Yihao Wang 03740227

December 6, 2022

Part 1: Skeleton Code

The workflow for the implemented visual odometry can be directly described as follows:

- Project the landmarks
- Detect keypoints and descriptors
- (Compute the essential matrix)
- (Match descriptors)
- (Find the essential matrix in inliers)
- Find matches landmarks
- Localize cameras
- (Add new landmarks)
- (Remove old keyframes)
- (Optimization)
- Update image views
- (Compute projections)

The items with brackets are only implemented when current frame is keyframe.
In conclusion, it chooses the first frame as keyframe (bool keyframe is true). Then it projects and matches until find matches landmarks. After that, it adds new landmarks and remove old keyframes, so that the number of keyframes keep constant. Then it optimizes the camera pose and position of landmarks with bundle adjustment. If the chosen frame is not keyframe, it just matches landmarks and localizes cameras, without optimization. In this way, the whole pipeline can run rapidly and real-time.

Part 3: Optimization

- Difference of optimize function between odometry.cpp and sfm.cpp:
  The main difference is that the optimization in visual odometry is real-time, while in sfm is not. In VO, the optimization only focuses on the frames after the last key frame. The gray landmarks are out of optimization window and are not optimized anymore. However, in sfm, all landmarks will be taken into consideration. Therefore, the optimization in VO is much faster than in sfm. Another difference is that the optimization in VO uses threads, which can run without affecting other processes.

- Functionality of the variables opt_finished and opt_running:
  This is to control the optimization threads and optimization variables. They are edited in next_step and optimize(). If optimization is done, the flag opt_finished turns true, and then it saves the correspondent landmarks, cameras and calib_cam. Then new optimization starts. If we remove them, we cannot optimize in a correct order, and the thread will become a mess.