Exercise Sheet 5

Topic: Real-Time Visual Odometry

Yun-Jin Li

November 30, 2022

Part 1: Skeleton Code

The workflow of the odometry method is as follow:

- If take_keyframe is true
  1. Project the landmarks into the left camera.
  2. Detect the keypoints and compute them into the descriptors for both left and right cameras.
  3. Compute the essential matrix for the left and the right camera (stereo pair).
  4. Find the matches between the keypoints in both cameras.
  5. Using the epipolar constraint to determine the inliers.
  6. Find the matches between the projected landmarks and the keypoints in the left camera.
  7. Localize the left camera using 2D-3D correspondences and use the stereo pair transformation to compute the pose of the right camera.
  8. Add new landmarks based on the 2D-2D correspondences between the stereo pair.
  9. Remove the old keyframe.
  10. Do bundle adjustment optimization with only the keyframe.
- If take_keyframe is false
  1. Project the landmarks into the left camera.
  2. Detect the keypoints and compute them into the descriptors for the left camera.
  3. Find the matches between the projected landmarks and the keypoints in the left camera.
  4. Localize the left camera using 2D-3D correspondences.
  5. Add new landmarks based on the 2D-2D correspondences between the stereo pair.
  6. If the length of inliers of the matches is bigger than new_kf_min_inliers and the optimization is neither finished nor running, set take_keyframe true.
  7. Check if the optimization is finished, if yes, update the optimized camera poses and the map.

Part 2: Setting Up the Problem

Please refer to the implementation on GitLab.

Part 3: Optimization

I think the optimize() method in this exercise is able to run the bundle adjustment in the background because it opens a new thread to run it. Since we run the bundle adjustment in the background, the other programs are still running and would have no idea about the status of the bundle adjustment. Therefore, these two flags opt_finished and opt_running can help us tell the other programs the status of the bundle adjustment so that we can access the result from it more safely. If we remove them, the other programs would have no idea how the bundle adjustment goes, and this would possibly lead to some issues (e.g when we try to modify the memory locations of the parameters that are used in the bundle adjustment).

References

As suggested in the exercise sheet.