visual navigation exercise 05

Name: Chenguang Huang Matriculate number: 03709255

June 2019

1 Exercise 1

The work flow of the implemented odometry is as follows: First judge whether this frame is key frame. If it is, the work flow is as below

1. Project existing landmarks to the current camera pose(left).
2. Detect key points and descriptors of left and right camera in the next frame.
3. Match key points of the left and right camera in the next frame and find the inliers.
4. Match key points of left camera in the next frame and projected key points of the current frame. Link projected key points with landmarks. Therefore the final result of the matching is the match between key points of left camera in the next frame and the landmarks.
5. Use the match between key points of left camera in the next frame and the landmarks to localize the camera pose of the next frame. Then update the current pose of the camera.
6. add new landmark with match data between stereo pair in the next frame using triangulation. In addition, add new observations to existing landmarks.
7. remove keyframes from **kf_frames** and cameras from **cameras** if the number of key frames stored in **kf_frames** is larger than **max_num_kfs**.
8. Optimize bundle adjustment problem within key frames.

If the frame is not key frame, the difference of the work flow is as below:

1. In step 2, only detect key points and descriptors in the left camera of the next frame.
2. Do not compute matches between stereo pair of the next frame as in step 3.
3. There are no steps 6, 7, 8.
4. If the inliers number is smaller than /textbfnew_kf_min_inliers, optimization is not running and optimization is not finished, then change the flag **take_keyframe** to true, so next frame will be key frame.

2 Exercise 3

2.1 What is the difference between optimize function in exercise 4 and exerciese 5?

1. In sfm, the optimization problem optimizes all landmarks. In slam, it optimizes only a fraction of landmarks.
2. In sfm, it fixes the first two camera and sets them as reference. In slam, because the cameras are removed and added frequently, there is no long living camera stored in **cameras** variable, so it uses the oldest two cameras stored in **cameras** as reference.

2.2 What is the function of variables opt_finished and opt_running?

These two variables are used to make the pipeline real time solvable. They are the criterion of deciding when it is time to add another key frame. Because optimizing the bundle adjustment problem might take some time, we can not add another key frame and launch another optimization problem when an optimization problem is already running. Thus these variables can prevent the pipeline from adding new key frame and launching another optimization problem. **opt_finished** adds the buffer function to this mechanism. When optimization ends, the variable let the pipeline have chance to update the optimization result.

2.3 What will happen if we remove them?

If we remove these two variables, the pipeline will not be real time because the pipeline will launch new optimization more frequently.