Exercise Sheet 5

Anselm Coogan

December 4, 2019

1 Exercise 1: Skeleton code

1.1 Workflow for implemented odometry method in next step in odometry.cpp

- if we're at the last camera: stop
- if there is currently no optimization running, do:
  - make sure that this loop doesn't run again before it has finished
  - project all landmarks onto image plane of left camera
  - detect keypoints for left and right image
  - load transformation between cameras from calibration results and compute the essential matrix
  - match keypoints and find inlier matches
  - match projected landmarks and keypoints
  - localize left camera (i.e. T w c) using the landmark - keypoint matches via PnP; and localize right camera through left T w c and transformation between cameras
  - add observations to existing landmarks from matched keypoints and add new landmarks by triangulating matched stereo keypoints that aren't already landmarks
  - remove old keyframes (i.e. cameras and associated observations)
  - optimize landmarks using bundle adjustment in separate thread
  - recompute landmark projections and remove outlier landmarks
  - go to next frame
- else, do:

  - above steps for only left camera and without optimization
  - before we go to next frame, we check whether optimization is running and has finished
    * if it's finished but the thread hasn't joined, we join the thread and assign the variables
    * if optimization is completed and the optimized variables have been assigned, we assure that we optimize both cameras in the next step
  - go to next frame

2 Exercise 3: Optimization

2.1 What is the difference in optimize in odometry and sfm?

- in odometry we only optimize over a moving window of keyframes in a separate thread
- in sfm we optimize over an evergrowing amount of keyframes/cameras

2.2 Opt finished and opt running

- what do these variables do?
  - opt running acts as a lock for the optimization thread and opt finished as a lock for the landmarks, cameras, and intrinsics which are being optimized
  - before we start to optimize in a new thread we set opt running to true which acts as a lock for optimization
  - after the bundle adjustment of the optimization thread is finished it releases the thread lock
  - after the new variables have been assigned, the variable lock is released
- what happens if we remove these variables?
  - if we remove opt running we will start concurrent optimization threads which will result in CPU overloading and race conditions
  - if we remove opt finished we might start a new optimization without using the optimized landmarks/cameras/intrinsics from the previous optimization