CMPUT 631: Autonomous Robot Navigation
Fall 2019
Assignment No. 3
Due 23:59, October 28, 2019 (via email of a PDF file)

1. Refer to the INTEL SLAM dataset we studied in Q6 of Assignment No. 1. Recall that the format of the data input to g2o solver is defined in the document titled g2o versus Toro, which also describes the cost function being used in g2o (see its Equation (3)). Please answer the following questions.
   (a) In your own words, explain how Equation (3) in the document "g2o versus Toro" is related to exponent in Equation (3) of [Cadena 2016].
   (b) Line 2000 and 3000 of the file "input_INTEL_g2o.g2o" reads:

   EDGE_SE2 771 772 0.006744 0.005112 0.865896 64935.054362 294012.600872 0.000000

   1331467.660768 0.000000 718.067286

   …

   EDGE_SE2 1158 1175 0.586855 -0.249767 2.829060 13.000858 20.975969 0.000000 243.941877

   0.000000 170.512281

   Show how the corresponding constraints are defined in terms of Equation (3) of "g2o versus Toro".
   (c) Use either the stand-alone version of g2o or the one in ROS installed with ORB-SLAM/RTAB-Map to optimize the pose graph defined by "input_INTEL_g2o.g2o". Upon completion, how much has pose or VERTEX 771 changed after optimization?

2. This OpenCV page contains examples of how ORB and SIFT keypoints can be detected and matched. "crossCheck" (mutual consistency) and distance ratio are used as ways of rejecting outliers, respectively. Debug the code if you have to so that you can replicate the outputs of the two examples. Then study this page that shows how to use the epipolar constraint on planar objects (e.g., one side of a biscuit box) to verify SIFT keypoint matches and reject outliers reliably. Debug the code if you have to so that you replicate the result on this page as well.
   (a) Combine the examples above into a script to match ORB keypoints reliably by imposing the homographic constraint, on the cookie box example (box.png and box_in_scene.png). Submit the matching result as an image.
   (b) Since homography applies only to planar objects (or special camera motion), epipolar constraint in feature matching in general is enforced through the fundamental matrix, as demonstrated on this page. Combine all the examples so you can make ORB keypoints in two images (left.jpg/right.jpg) by imposing the epipolar constraint in terms of the fundamental matrix. Submit the matching result as an image.

3. Visit the GitHub page DBoW2, the repository for the hierarchical (tree) BoW implementation with ORB features. Download the code and build and test the executable "demo", to make sure that it runs properly. Study demo.cpp and examine the images in the "images" directory to see how image matching is performed with the BoW index.

   Download map.zip and queries.zip. First build the BoW visual vocabulary using the images in map.zip as the training set using the default branch factor k (=9) with tree depth L =1,2,3 and 4. For each tree depth, find the top matching image in map.zip for each of the 4 images in queries.zip. Submit the matching result in the form of a table whose entries are the map image indices matching the query images.

Materials to Submit

Problem 1: (a) a description (b) two expressions (c) a transformation
Problem 2: (a) an image (b) an image
Problem 3: a table

In addition, for Problems 2(a), 2(b) and 3, zip the three program files called p2a, p2b, and p3, and submit the zip file as an attachment to the email.