Visual-Based Navigation
Solution Exercise Sheet 2
Topic: Feature Detectors, Descriptors, Epipolar Geometry, RANSAC

Kamysek, Josef-Maria

May 16, 2021

Part 2: Epipolar Constraint

We know that when X is observed by two cameras, this point and two focal points O_L and O_R of the cameras form a plane. In particular, the epipolar constraint states that the vectors O_LX (x_L), O_RX (x_R), O_LO_R should lie in the same plane. This constraint can be formulated using the essential matrix as follows: $x_L^TE_R = 0$. We know that the projection matrices M and M' map 3D points to their respective 2D image planes look the following way: $M = K[I \quad 0]$ and $M' = K'[R^T \quad -R^TT]$. Assuming canonical cameras with $K = K' = I$ we obtain the following reduced equations: $M = [I \quad 0]$ and $M' = [R^T \quad -R^TT]$. This means that for the location of our x_R we have $Rx_R + T$ (1). Now we need to find a way to construct a dot product such that the equation shown is zero. This condition is satisfied by taking the cross-product of the above equation (1) and T. Since both vectors lie in the epipolar plane we obtain a vector that is normal to the epipolar plane. This means that x_L is normal to $T_x(Rx_R)$ and results in $x_L^T \cdot [T \times (Rx_R)] = 0$. This cross-product can be converted into matrix multiplication representation and gives us the term: $x_L^T \cdot [T_X]Rx_R = 0$. Meaning our essential matrix $E = T_xR$.

The solution was developed using the following source: Epipolar Geometry Stanford

Part 4: Bag-of-Words for Place Recognition

1. What is the main difference between the match_all() and match_bow() functions in src/sfm.cpp?
   (a) The main difference is that the match_bow function loads the bow vocabulary and performs matching using a location recognition approach that allows finding candidate pairs using bag-of-words descriptors. In particular a bow database is built using feature corners. The database allows to insert new bow vectors for frame cam ids as well as querying bow vectors. Match_all() simply performs a brute force matching.

2. What does the num_bow_candidates parameter control?
   (a) The num_bow_candidates define the result size of the query operation. In particular, it specifies how many pairs of FrameCamId and WordValue should be returned.

3. Comparison of the number of candidate pairs and inliers when using the match_all() and match_bow() functions of Frame1 = 1 and Frame2 = 0.

   Brute Force Matching:

   (a) Brute-force matching 13284 image pairs...
   Successfully matched 978 out of 13284 image pairs with a total of 42067 inlier feature matches (109579 total). New total of matched image pairs is 13284.
   (b) Detected 281 corners, Detected 44 matches, Detected 37 inliers

   BoW Matching:

   (a) Matching 3649 image pairs using BoW...
   Successfully matched 448 out of 3649 image pairs with a total of 22925 inlier feature matches (43848 total). New total of matched image pairs is 3649.

(b) Detected 281 corners, Detected 44 matches, Detected 39 inliers