Feature Matching

Goal

In this chapter

- We will see how to match features in one image with others.
- We will use the Brute-Force matcher and FLANN Matcher in OpenCV

Basics of Brute-Force Matcher

Brute-Force matcher is simple. It takes the descriptor of one feature in first set and is matched with all other features in second set using some distance calculation. And the closest one is returned.

For BF matcher, first we have to create the BFMatcher object using **cv.BFMatcher()**. It takes two optional params. First one is normType. It specifies the distance measurement to be used. By default, it is **cv.NORM_L2**. It is good for SIFT, SURF etc (**cv.NORM_L1** is also there). For binary string based descriptors like ORB, BRIEF, BRISK etc, **cv.NORM_HAMMING** should be used, which used Hamming distance as measurement. If ORB is using WTA_K == 3 or 4, **cv.NORM_HAMMING2** should be used.

Second param is boolean variable, crossCheck which is false by default. If it is true, Matcher returns only those matches with value (i,j) such that i-th descriptor in set A has j-th descriptor in set B as the best match and vice-versa. That is, the two features in both sets should match each other. It provides consistent result, and is a good alternative to ratio test proposed by D.Lowe in SIFT paper.

Once it is created, two important methods are *BFMatcher.match()* and *BFMatcher.knnMatch()*. First one returns the best match. Second method returns k best matches where k is specified by the user. It may be useful when we need to do additional work on that.

Like we used **cv.drawKeypoints()** to draw keypoints, **cv.drawMatches()** helps us to draw the matches. It stacks two images horizontally and draw lines from first image to second image showing best matches. There is also **cv.drawMatchesKnn** which draws all the k best matches. If k=2, it will draw two match-lines for each keypoint. So we have to pass a mask if we want to selectively draw it.

Let's see one example for each of SIFT and ORB (Both use different distance measurements).

Brute-Force Matching with ORB Descriptors

Here, we will see a simple example on how to match features between two images. In this case, I have a queryImage and a trainImage. We will try to find the