Lab 5 report : panoramic image

In this lab we want to develop an algorithm which, starting from a set of images taken from the same viewpoint but at gradually rotated horizontal angles, is able to produce a panoramic view. I obtained this by first applying feature detection, description (ORB) and matching between the consecutive images. Then the horizontal shift between successive frames was extracted from the interpolated homography linking the coordinates of keypoints whose feature descriptors matched better.

Practical implementation difficulties and lessons learned:

- You cannot just declare a OutputArray variable and pass it to a method that asks for a an OutputArray to write on. For instance in detectAndCompute (InputArray image, InputArray mask, std::vector< KeyPoint > &keypoints, OutputArray descriptors, bool useProvidedKeypoints=false) I have finally understood that I had to pass a cv::Mat as descriptors.

- References must always be initialized when declared. Initialization is also necessary for all the variables inside the definition of "struct", which is very helpful to group outputs of a method into a single object.
- Printing variables values to standard output can be very helpful in debugging. I noticed in this way that my matches looked so bad because I had actually passed by errors the descriptors of the same image's features to the matcher twice... As keypoints were different it was not straightforward to notice by looking at drawMatches() output image, but by printing DMatch.trainIdx, DMatch.queryIdx and DMatch.distance the first two appeared to be always identical and the third always equal to zero.
- When fitering the matches, it may happen that we exclude all of them, especially if min_dist = 0 which is not so unlikely when using the Hamming distance between ORB descriptors.
- The visualization of the filtered matches is not stable for reasons yet unknown to me so I avoided to display the filtered matches but I still use them.
- The method warpPerspective() allow to apply an homography to an image, by passing the inverse of the previously identified homography matrix we can thus reharmonize the viewpoint of two different images.

Additional notes:

The C++ algorithm used to obtain all the previous results can be found in the folder “CV_Lab5_code”, the main file that need to be compiled and executed run is “LAB3.cpp”, it is essential that the main file is kept together with all its dependencies contained in the other files and to pass as a parameters : the glob() pattern indicating the location of images, the field of view angle of the camera, the ratio to use in match filtering. For instance to execute the compilation output a.out over pictures in format .bmp of folder kitchen, with an angle of 66° and ratio = 5 use: *LD_LIBRARY_PATH=/workspace/opencv/lib/ ./a.out "/home/francesco/Desktop/Computer Vision/CV Lab 5/kitchen/*.bmp" 66 5* figures will appear quickly, just press any keystroke to pass to the successive.