Assignment 6

This exercise is part of the course assignment. **Deadline for the assignment 30.11.2022 at 23:59**

The topic of this assignment is features matching using SURF descriptors. For this assignment you should return

- The files surf_nn.m and surf_nndr.m. Each file should contain your name and student number (of both students if you work in pairs).
- Your answers to the questions in the analysis part. At the end of the course, you should return a single pdf containing the answers to all questions of the assignments. The report should contain also your name and student number (of both students if you work in pairs).

Coding part (5pt)

First run the demo demo6.p to get a taste of the assignment. Your first task is to complete the function [y1, y2] = surf_nn(surf1, surf2, x1, x2). The variables x_1, x_2 are $m \times 2$ and $n \times 2$ matrices containing the coordinates of the interest points in the first and second image, respectively. The matrices $surf_1$ and $surf_2$ contain the surf descriptor (one per row) of each keypoint.

Your first task is to find the coordinates of the *mutually nearest* keypoints and return their coordinates sorted from the most similar pair to the most different. The similarity is measured using the standard Euclidean distance between descriptors. The smaller the distance, the more similar the descriptors are. Particularly you should

1. Compute the $m \times n$ matrix distmat, each element at row i and column j should contain the euclidean distance between the ith descriptor of the first image and the jth descriptor of the second image.

2. Let f_1 a descriptor from the first image and f_2 a descriptor from the second image. The two descriptors are said to be mutually nearest if f_1 is the most similar descriptor to f_2 among all descriptors in the first image **and** f_2 is the most similar descriptor to f_1 among all descriptors in the second image. Construct a matrix pairs which contains the mutually nearest descriptors. The first column should contain the index of the descriptor of the first image, the second column the index of the corresponding closest descriptor in the second image and the third column the distance between the two descriptors.
   Example 1: The third row from surf1 and the fifth row from surf2

form a mutually nearest pair and their distance is 0.4, the matrix pairs should then contain the row $[3, 5, 0.4]$.
Example 2: You pick the fourth row from surf1 and find that the closest descriptor from surf2 is the second row. However, when you consider the second row of surf2, you find that closest row from surf1 is the fifth, hence the fourth and second descriptor from surf1 and surf2 do not form a mutually nearest pair.
Hint! Read carefully the documentation of the matlab function min. What does min(A, [], 2) do? What happens if you call min with two outputs? You may want to loop through each descriptor in surf1, find its closest neighborhood from surf2 and check if they form a mutually nearest pair.

3. Sort the rows of pairs in increasing order of distances.

4. Extract the coordinates of the sorted mutually nearest pairs from x_1 and x_2.

Your second task is to perform feature matching using nearest neighborhood distances ratios. Let f_1 and f_2 two mutually nearest descriptors as before and let f_3 the second closest descriptor to f_1 from the second image. Now we define the distances ratio as

$$dr = \frac{||f_1 - f_2||}{||f_1 - f_3||} \tag{1}$$

and use this distances ratio to measure the similarity between descriptors instead of plain Euclidean distance, i.e. the smaller the distances ratio, the more similar the descriptors are. Here you will need to complete the function [y1, y2] = surf_nndr(surf1, surf2, x1, x2). Particularly,

1. Repeat steps 1-2 from the previous task

2. For each row in pairs (i.e. each mutually nearest pair), compute the distances ratio as defined in eq. (1). Store the distances ratios in an array called nndr. **Hint:** you may want to create a copy of distmat where the columns are sorted in increasing order.

3. Sort the rows in pairs in increasing order with respect to nndr.

4. Extract the coordinates of the sorted mutually nearest pairs from x_1 and x_2.

When you are done, run the script main6.m to verify your functions work correctly.

Analysis part (5pt)

Answer the following questions in your report

- Consider the 5 best matches for each algorithm. How many matches are correct using plain distances? How many matches are correct using ratios of distances? Why does using distance ratios lead to a more robust result? I.e. what is the idea behind nearest neighborhood distance ratio? (3pt)
- What are the advantages of using descriptors like SIFT or SURF to perform feature matching over comparing the raw RGB values of the pixels? (2pt)