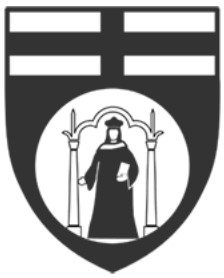

Università degli studi di Genova

Computer Vision

Assignment 7: Image Matching and Retrieval .

Aurora Bertino, Chiara Saporetti, Silvana Andrea Civiletto

June 14, 2020

Introduction

0.1 Image Matching

Feature detection is the process of computing the abstraction of an image information and making a local decision at every image point to see if there is an image feature of the given type existing in that point.
The concept of estimating the similarity between image pairs is very important and it is usually estimated with an image matching process. In this Lab, we are going to compare the results from different feature descriptors: SIFT and NCC (keypoints feature).
SIFT is a typical approach to matching features across different images, when all images are similar in nature (same scale, orientation, etc) simple corner detectors can work. But when there are images of different scales and rotations, the latter don't work.
The idea behind SIFT algorithm is to detect points on a multi scale image representation and then associate them with a vectorial description invariant to translation, rotation and scale. The SIFT algorithm has 4 basic steps. First: estimate a scale space extrema using the Difference of Gaussian (DoG).
Second: Apply key point localization where the key point candidates are localized and refined by eliminating the low contrast points.
Third: Assign key point orientation based on local image gradient.
Fourth:Create a keypoint descriptor generator to compute the local image descriptor for each key point based on image gradient magnitude and orientation. Major advantages of SIFT are:

- ***Locality:*** *features are local, so robust to occlusion and clutter (no prior segmentation).*
- ***Distinctiveness:*** *individual features can be matched to a large database of objects.*
- ***Quantity:*** *many features can be generated for even small objects.*
- ***Efficiency:*** *close to real-time performance.*
- ***Extensibility:*** *can easily be extended to a wide range of different feature types, with each adding robustness.*

Another way to apply template matching, which is a simple intesity matching is through normalized cross correlation (NCC). Given an image f and a template g:

- *Substract from both of them the mean value of the template intensity. Without this step, the correlation score would increase with the brightness of the patch.*
- *Divide for the standard deviation. Without this step, the correlation score would be highest in the local neighborhood of the object.*

The result is a matrix that has the maximum value in the element corresponding to the pixel that has the highest correspondence score.

0.2 Image Retrieval

The idea behind the **Bag of Words image representation** *is to first build a visual dictionary. The elements of the dictionary are the visual words. The steps to create a BoW representation are:*

• *Build a gallery, which is the set of images that our algorithm has and through which it gains information by looking for keypoints in each image. The descriptors for each keypoint are usually NCC or SIFT ones.*

• *Build the dictionary, in which words are derived by taking the most frequent descriptors and then grouping similar ones together (via coherency measures). This is called the clustering method.*

• *Represent all images by quantizing their keypoints with respect to the gallery: each image is represented by computing an histogram of the frequencies of all visual words appearing in it. Image retrieval is achieved by representing a query image according to the dictionary D, comparing it with the image representations in the gallery according to a certain metric and then retrieving the k images most similar to the query one.*

0.3 Matlab Resolution

0.4 Image Matching

The goal of this Lab is to observe the effect of changing the similarity function and its parameters. The matching is based on an affinity matrix including the similarities between keypoints comparing their position and appearance (using euclidean distance and Normalized-Cross Correlation) or their SIFT descriptors. In

the provided main file "Labo8_part1" we modify the code by adding a for loop. In this way, starting from a sigma=1 (both for NCC and SIFT) and by increasing it by 0.5 at each loop we obtain different results for different sigma values. Basically, we have a loop with an i=1 :5 in which are applied the provided functions:

$$list\_ncc = findMatches(img1, img2, 'NCC', threshold, sigmaNCC, sigmaSIFT); \quad (1)$$

which extract CORNERS and SIFTs from both images (keypoints positions and sift descriptors)

show_matches(img1, img2, list_ncc, 0, 1, 'NCC');

Likewise, starting from threshold=0.1, we in crease the threshold values by 0.1 at each loop in order to compare the results with different threshold values.

Results

0.5 Image Matching

We tested the provided code, with different sigma values (sigmaNCC and sigmaSIFT) which are input values for the function "findMatches" and we noticed that increasing sigma we decrease "affinity". This is because in the formula of "affinity matrix" sigma value is at the denominator. So, sigma and affinity are inversely proportional.
From the different results obtained by NCC (keypoints descriptors) and SIFT, we point out that at the beginning of the for loop (with the smaller sigma value) the features matched with NCC are 65 and that this number will be lower increasing sigma. Hence, with a sigmaNCC= 3.5, the value which represents the feature matches with NCC is 44. Also, regarding image matching with SIFT descriptors, we have different results applying the same sigma values.
Infact, setting sigmaSIFT=1 the features matched with SIFT are 85, increasing sigma they arrive to the value 79. So, we can affirm that using the SIFT descriptors the number of matches are always bigger with respect to the ones found using the NCC descriptors.
This is shown in the figure 1-2. We can also notice that variation of the Threshold influences the number of features matched with NCC and with SIFT increasing the accuracy of the results.
At the beginning, we setted a threshold=0.1 and we increase it by 0.1 at each loop. We also made these steps for image In Figure 3. We compare the two methods showing the results with sigma=3.5. In this image there are more features matched, infact we start with a number of 412 (for NCC) and 523 (for SIFT).
Obviously, as in the previous steps, increasing sigma values we reduce the features matched respectively into 79 and 332.

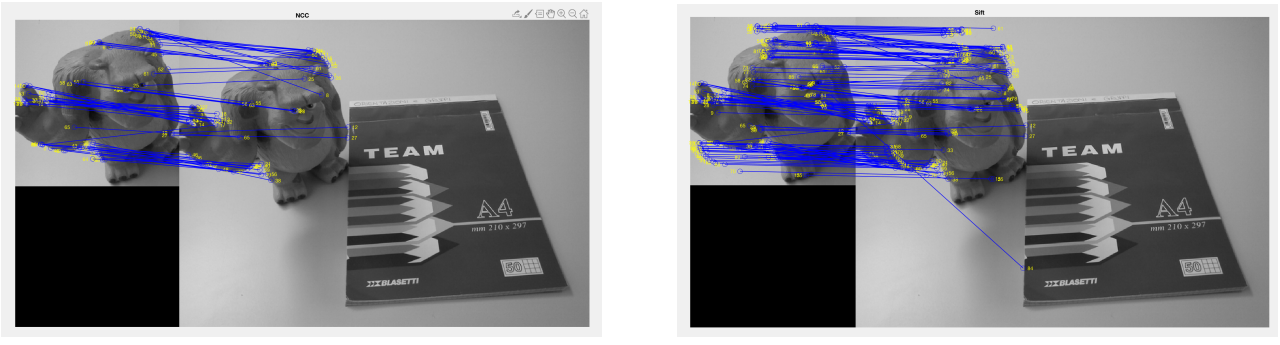

Figure 1: Ncc,sigma=1 and SIFT, sigma=1

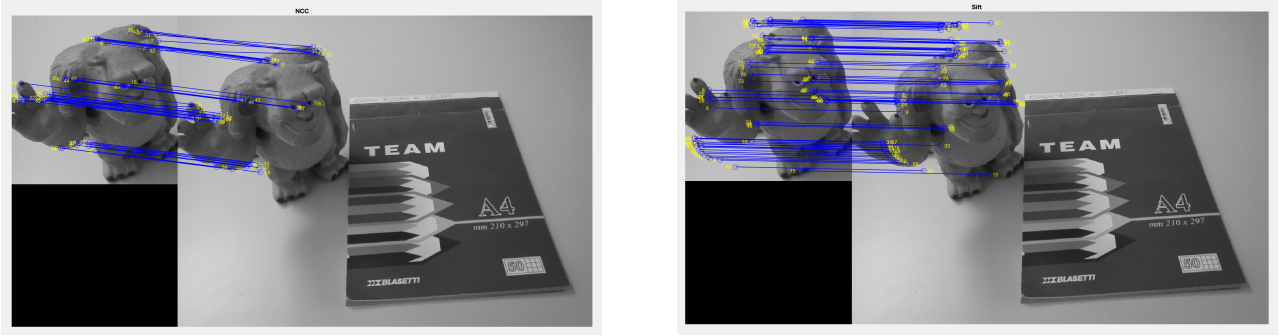

Figure 2: NCC,sigma=3.5 and SIFT, sigma=3.5

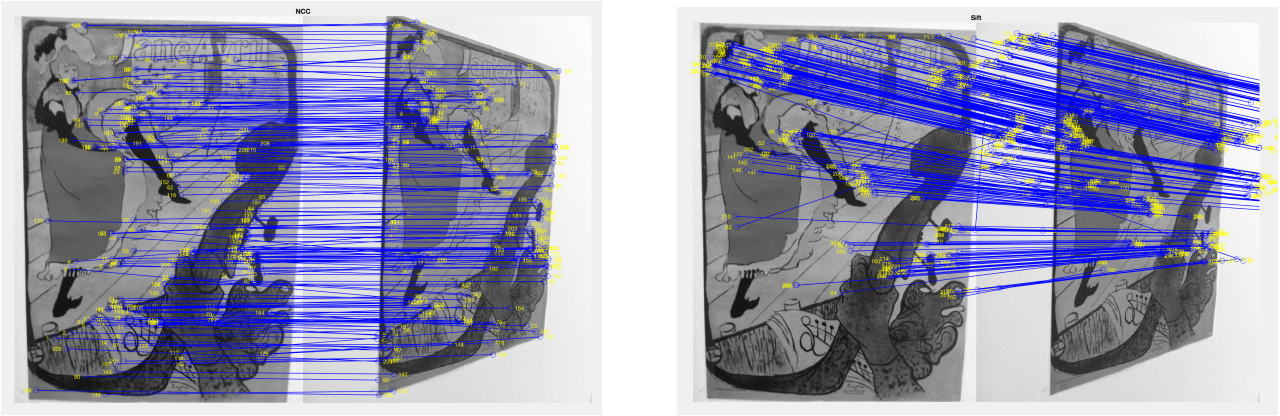

Figure 3: NCC,sigma=3.5 and SIFT, sigma=3.5

0.6 Image Retrieval

In the image retrieval code we have some dictionaries of different dimensions. We tried all of them to understand the differences in the results. We also tried the code on different query images. The results improve proportionally to the size of the dictionary, even though the results obtained are in general very good since in the gallery we have groups of similar images. However, also the time taken to apply the code increases proportionally. We tested the code on various different images and sizes of dictionaries. The following pictures illustrate some of the results we obtained, by using two pictures from the gallery and two sizes of the dictionary.

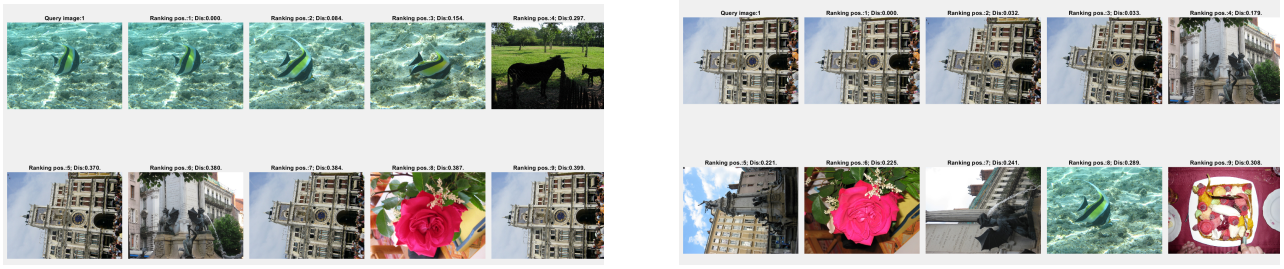

Figure 4: D=100 applied to two images

Figure 5: D=1000 applied to two images