Computer Vision Laboratory Report n.7

IMAGE MATCHING AND RETRIEVAL

June 14, 2020

Chiara Terrile ID: 4337786
Giulia Scorza Azzarà ID: 4376318
University of Genoa

Contents

1 Introduction **2**
1.1 Defining the objective 2

2 Implementation **3**
2.1 Image Matching 3
2.2 Image Retrieval 3

3 Results **4**
3.1 Analysis 1 - Image Matching 4
3.1.1 NCC 4
3.1.2 SIFT 6
3.2 Analysis 2 - Image Retrieval 8

Chapter 1

Introduction

1.1 Defining the objective

The goal of this laboratory was to understand the meaning and use of different strategies for image matching and image retrieval by practicing with some provided MATLAB code. In order to evaluate the results with respect to the variation of some key parameters, the code has been run on different image pairs and on the provided gallery of photos.

Chapter 2

Implementation

The provided MATLAB code is divided in two main folders. Each folder has a main script (respectively **Labo8_part1** and **Labo8_part2**) and some functions.
In the following sections the files of both folders are reported and described; the first referring to Image Matching, while the second referring to Image Retrieval.

2.1 Image Matching

In this section all the files contained in the first folder are briefly described:

- **Labo8_part1**: which loads the image pairs and then finds and displays the located matches.
- **findMatches**: which locates the matches using two different methods. The method used is specified as a parameter, whether NCC or SIFT.
- **similarity**: which computes the similarities between a given pair of features, according to the different inputs parameters.
- **show_matches**: which visualizes the two images and the list of matching features.

2.2 Image Retrieval

Here below all the files contained in the second folder are briefly described:

- **Labo8_part2**: which loads the images, extracts the features for dictionary learning, constructs the dictionary, constructs the Bag of Features, performs the image retrieval and ranks the images.
- **load_images**: which simply loads the set of image descriptors.
- **showranking**: which simply displays the image ranking.

Chapter 3

Results

Since the MATLAB code was entirely provided as the starting material, the work required to us was to analyze it by focusing on two different experiments:

- **Analysis 1**: in which the focus is on Image Matching and it requires to evaluate the effects of varying different parameters.
- **Analysis 2**: in which the focus is on Image Retrieval and it requires to evaluate the performance when varying the size of the dictionary.

3.1 Analysis 1 - Image Matching

In this first analysis the Image Matching has been performed applying two different methods (NCC and SIFT). In both cases, at the beginning, features are extracted using Multi-Scale Harris, then a similarity measure is performed. In the first case, by mixing the Normalized Cross Correlation (NCC) applied to image patches of size $2 * delta + 1$ and a measure of the euclidean distance of the detected feature points; in the second case, by comparing the distance of the SIFT descriptors of the found features.
In general, it is possible to notice that the correspondences provided by SIFT are more accurate than those provided by NCC.

3.1.1 NCC

As far as it concerns the Normalized Cross Correlation method, the parameters to be varied are sigma (σ), which controls the maximum distance between the positions of the features considered, and a minimum threshold (Th) of the correspondence value in the affinity matrix such that one correspondence can be considered good.
These two parameters are set in **findMatches**; *sigma* was already present in the code, while the threshold has been added to see how it affects the number of matches.
By increasing the value of sigma, the correspondences seem to decrease, since less one-to-one matches happen. When increasing the threshold for higher values of σ (e.g. $\sigma = 0.1$), the correspondences with low similarity scores are discarded and therefore some incorrect

correspondences disappear, with the result of seeing a decrease of corresponding matches. The table below reports the values assigned to the parameters σ and Th in different combinations and each case is associated with a corresponding figure.
All the figures are reported in the following pages (from Figure 3.1 to Figure 3.4).

N° Figure	Sigma	Threshold	N° Matches
Figure 3.1	$\sigma = 0.01$	Th $= 0$	85
Figure 3.2	$\sigma = 0.01$	Th $= 0.75$	85
Figure 3.3	$\sigma = 0.1$	Th $= 0$	47
Figure 3.4	$\sigma = 0.1$	Th $= 0.75$	41

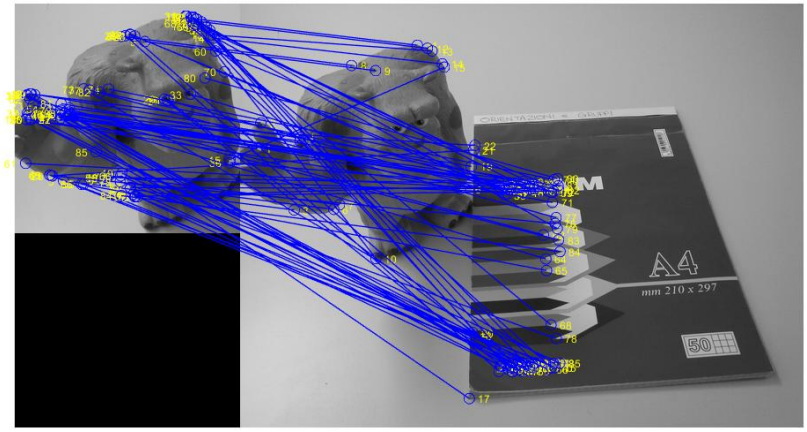

Figure 3.1: 85 matches found

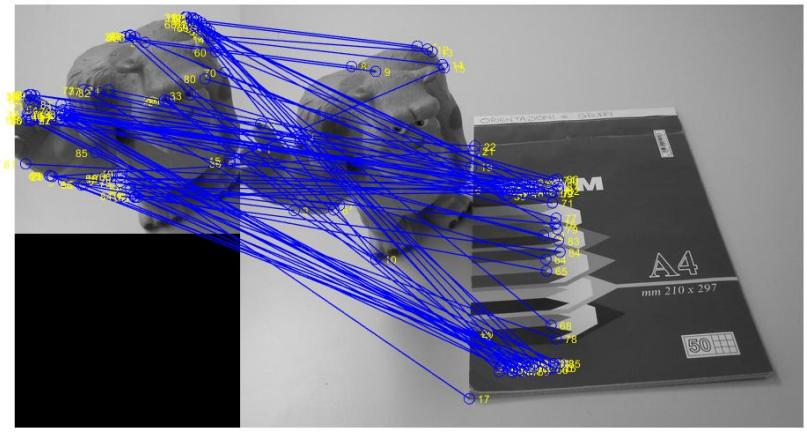

Figure 3.2: 85 matches found

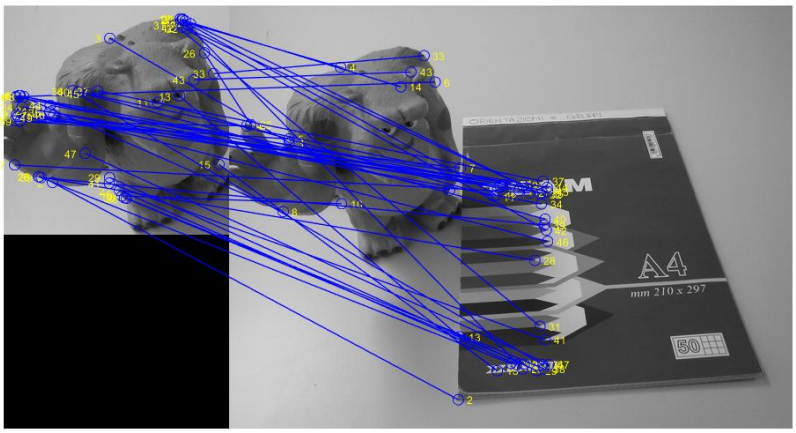

Figure 3.3: 47 matches found

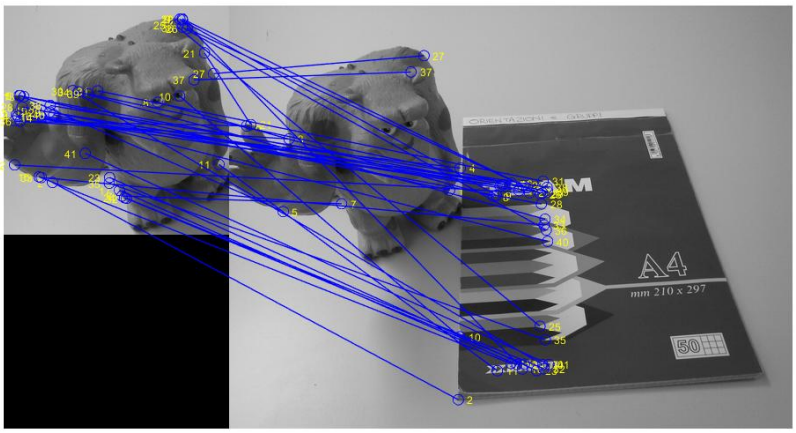

Figure 3.4: 41 matches found

3.1.2 SIFT

Regarding the SIFT method, the parameter to be varied is just sigma (σ), which in this case has a different meaning than before and a less immediate interpretation, while the minimum threshold (Th) of the correspondence value in the affinity matrix is considered as fixed to 0.75. In this case, when increasing the value of sigma, the number of correspondences decreases; in particular, those that disappear seem to be false correspondences. The table below refers each figure (from Figure 3.5 to Figure 3.7) to its value of σ.

N° Figure	Sigma	N° Matches
Figure 3.5	$\sigma = 0.1$	85
Figure 3.6	$\sigma = 0.5$	82
Figure 3.7	$\sigma = 1$	70

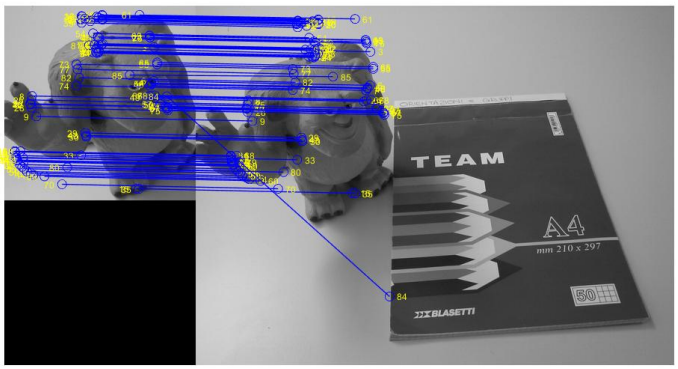

Figure 3.5: 85 matches found

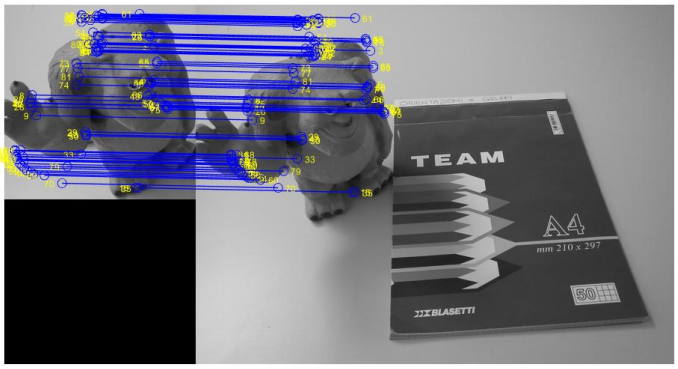

Figure 3.6: 82 matches found

Figure 3.7: 70 matches found

3.2 Analysis 2 - Image Retrieval

In this second analysis the Image Retrieval has been performed by extracting features from each image to build the bag of words, which are sorts of histograms used to measure the similarities between images. This happens for each image by comparing the SIFT description of the features to those of the code words of a code book provided in the files. Then a ranking based on these measurements is built. The main goal of this part was to explain the differences in the results when changing the size of the dictionary within the possible values (30, 100, 200, 300, 500, 1000). Given a query image (Q1 or Q2), the code estimates a rank of the images which are more similar to the one provided.
Therefore the number of code words used (the size of the dictionary) influences the precision of the image description as a sort of resolution of the image description.
As a consequence, by increasing its size, the precision of the ranking should improve.
The table below refers each figure (from Figure 3.8 to Figure 3.13) to its image query (Q1 or Q2) and threshold (Th) value. All the images are shown in the following pages.

N° Figure	N° Query	Threshold
Figure 3.8	Q1	Th = 30
Figure 3.9	Q1	Th = 300
Figure 3.10	Q1	Th = 1000
Figure 3.11	Q2	Th = 30
Figure 3.12	Q2	Th = 300
Figure 3.13	Q2	Th = 1000

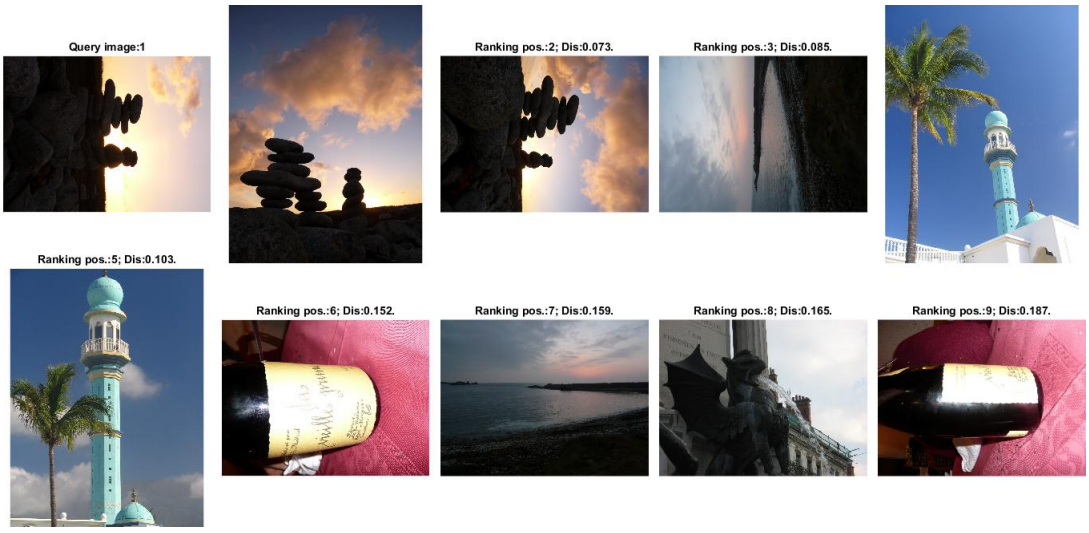

Figure 3.8: Image query Q1, Th = 30

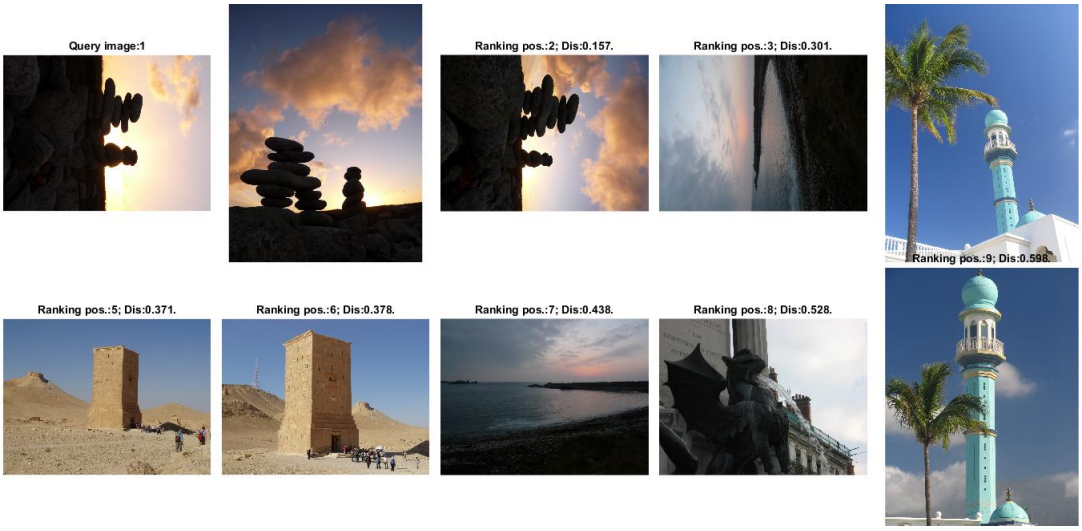

Figure 3.9: Image query Q1, Th = 300

Figure 3.10: Image query Q1, Th = 1000

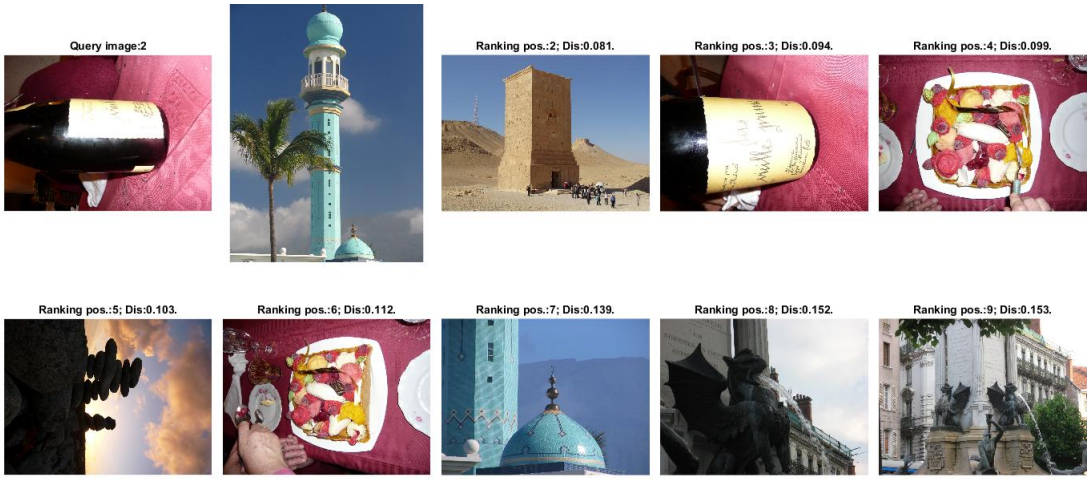

Figure 3.11: Image query Q2, Th = 30

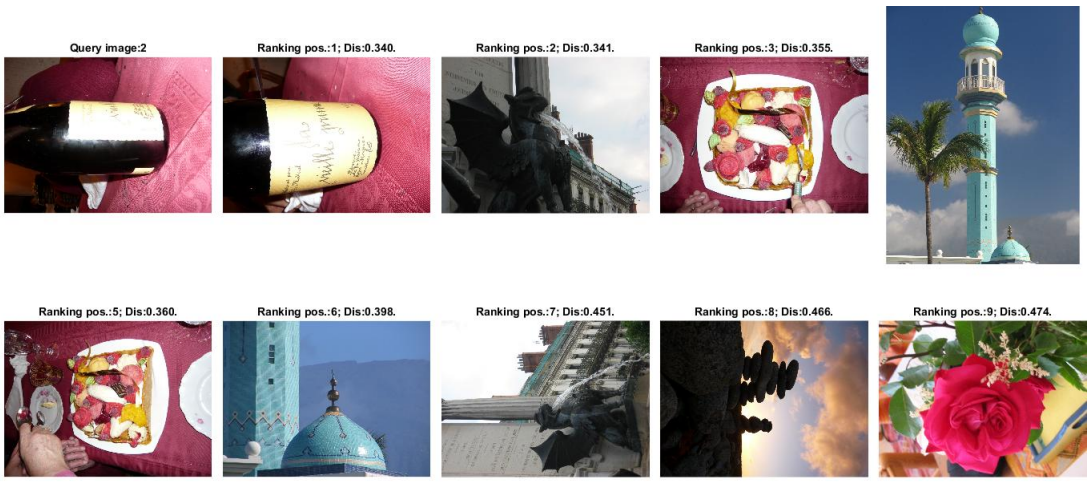

Figure 3.12: Image query Q2, Th = 300

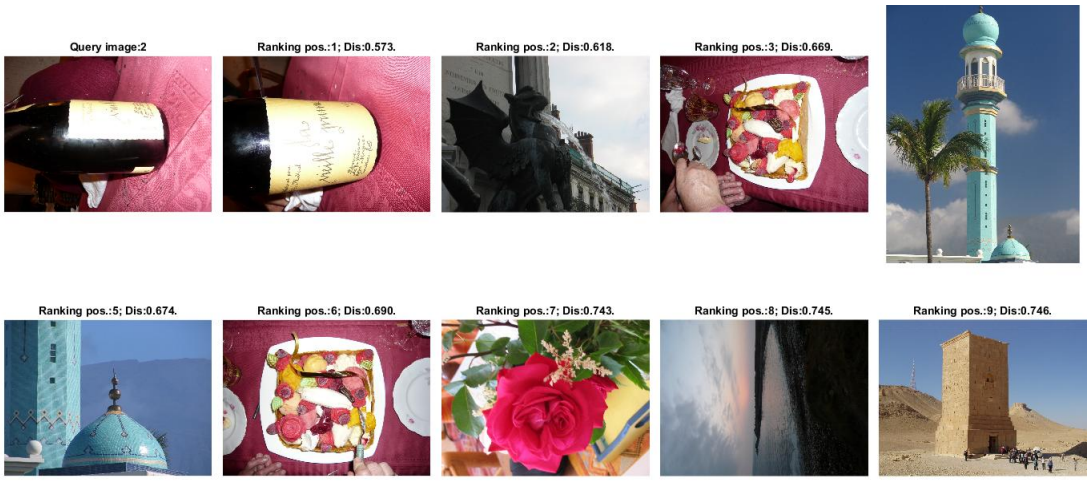

Figure 3.13: Image query Q2, Th = 1000