LAB 7

Image Matching and Retrieval

Matteo Dicenzi
Marco Demutti

1. Introduction

The aim of this experiment is to perform **Image Matching** and **Image Retrieval** with different images.
In general, Image Matching is a procedure that allows to estimate the similarity between image pairs. In this lab we are going to perform a **local matching**, i.e. we are going to extract local features (corner keypoints along with their SIFT descriptors) from both images and then solve a feature correspondence problem.
In particular, the affinity matrix will be created using two different similarity measures: at first features will be compared considering their position and appearance (representing features with image patches centered in the feature points and then using *Euclidean distance* and *Normalized-Cross Correlation*); then features will be compared using their SIFT descriptors. While the first representation is effective if the differences between the two images are mainly due to translation, the second one is a vectorial representation which is also invariant to scale and rotation changes.
In the second part of the experiment, starting from one given image called *query image*, a set of images (the *gallery images*) and a set of correspondences between the query image and each one of the gallery images (matching points), we will retrieve a sorting based on the similarity between the query image and the images in the gallery. This process is known as **Image Retrieval**.
Our goal will be to assess different strategies for image retrieval, and try to observe the differences among them. In particular, the first procedure will be based on SIFT descriptors and the bag-of-keypoints paradigm, while the second procedure, that we are going to implement, will sort images in decreasing order with respect to the number of detected matches with the query image.

2. Procedure

Starting from some provided code, our first goal was to observe different results obtained by changing some characteristic parameters in both Image matching and image retrieval procedures (see *Results* section).
As a next step, we set up a new procedure (*Labo8_part3.m*) to perform image retrieval with a different strategy: this time, the aim was to sort the images in the gallery by decreasing order with respect to the number of detected matches with the query image (analyzing the similarities using *SIFT descriptors*). The procedure is the following: at first, we set up the environment by loading the query image (we chose *Monster.jpg*) and the gallery images. Then, through the provided function *findMatches ('SIFT'),* we computed and stored the number of the matching points between the query image and the *i-th* gallery image.

Finally, we were able to sort the image gallery by a decreasing order and display the results obtained, using the provided function *showranking.m*.

3. Results

In the first part of the experiment, we focused our attention on image matching. In particular, we observed the effects of changing the features description and some other parameters of the affinity matrix (i.e. $\boldsymbol{\sigma}$ and the threshold).
The first analysis that was performed was related to the image matching between the two images that portray the "monster" (i.e. '*Ex01_01.jpg* and *Ex01_02.jpg*'). It was performed by considering *Euclidean distance* between positions and *patches similarity* using *NCC*. When computing the affinity matrix, which is described by the equation $E(i,j) = e^{-\frac{\|p_i - p_j\|^2}{22}}$, we chose $\boldsymbol{\sigma}$ = 0.05. This parameter controls the maximum distance we are willing to consider between two points. In addition, we added a threshold equal to 0.6. This means that, when a maximum in the affinity matrix E(i,j) is detected (i.e. the value is greater than any other value both in its specific row and in its column), it is considered *only if* it is below the threshold. This allowed us not to take into account the weakest similarities between the two images.
Figure 3.1 shows the result of the matching and *Figure 3.2* the corresponding number of matches. It is clear that this is not a satisfactory result. This is mainly due to the bad choice of $\boldsymbol{\sigma}$, which is too low.
In fact, if we apply the same procedure to the same two images, this time setting the value of the standard deviation $\boldsymbol{\sigma}$ equal to 0.5 (and the threshold equal to 0.9), we can see that the results are far more satisfactory (*Figure 3.3*). Choosing a higher value of $\boldsymbol{\sigma}$ permitted to consider, as neighbors, points placed at a higher distance. In general we observed that, for a fixed value of the threshold, the number of matches decreases (*Figure 3.4* for the shown case) as $\boldsymbol{\sigma}$ increases. This can be justified by the fact that, in this case, local correspondences are less enforced (the Euclidean distance has less influence on the value of the *affinity matrix*) and, as a consequence, local maxima (in the row and in the column of the matrix at the same time) are more difficult to be found.

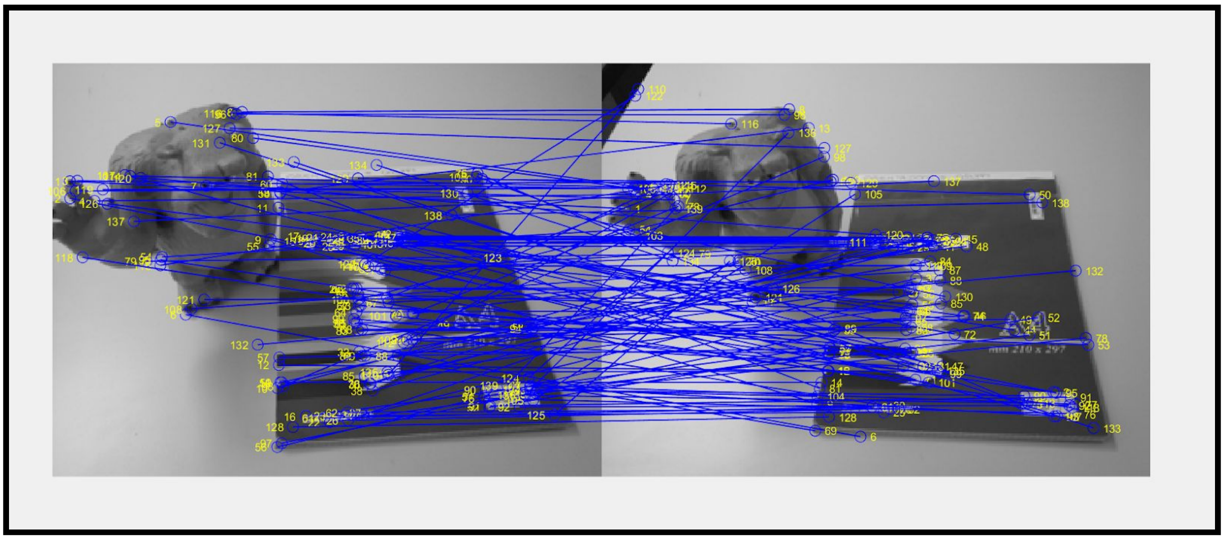

*Figure 3.1: results of matching using NCC (**σ** = 0.05, threshold = 0.6)*

```
Features matched: 146
```

Figure 3.2: total number of feature matched

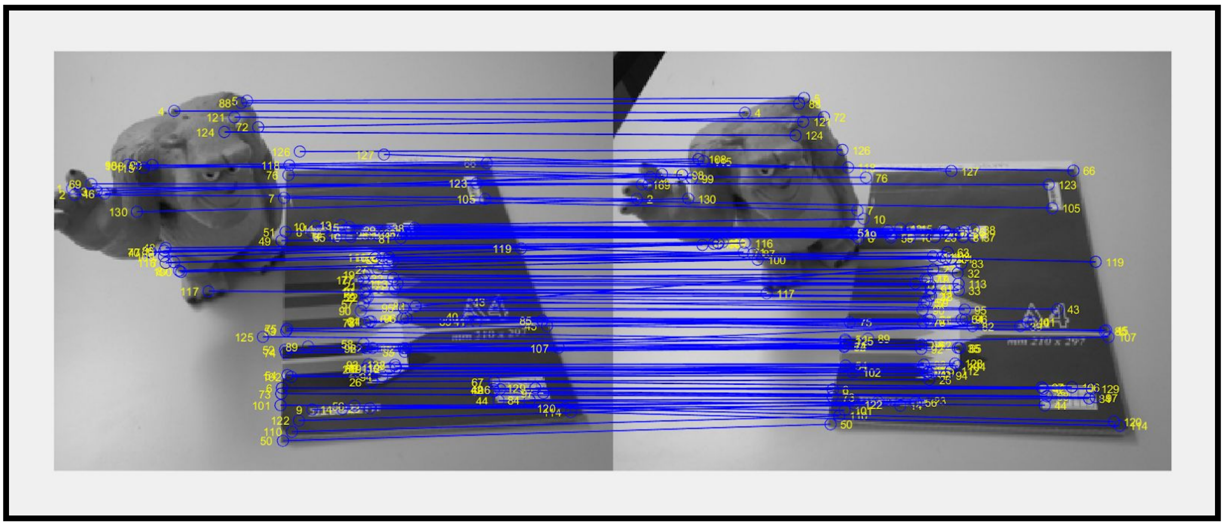

*Figure 3.3: results of matching using NCC (**σ** = 0.5, threshold = 0.9)*

```
Features matched: 130
```

Figure 3.4: total number of feature matched

In *Figures 3.5-3.6* we show the result of the image matching between the two images '*Ex02_01.jpg*' and '*Ex02_02.jpg*' and the total number of matched features. This time the matching was performed using *SIFT descriptors*, due to the characteristics of the images. In fact, we can notice how these two images represent the same subject seen from two different points of view. (i.e. change in the orientation). For this reason, and for the nature of SIFT descriptors, we thought that this was a good way to carry out the matching.

In addition, as shown in *Figure 3.7*, we provide the matching by comparing the features using NCC. The result is clearly not satisfactory as the previous one, due to the nature of this kind of method (as explained in the Introduction).

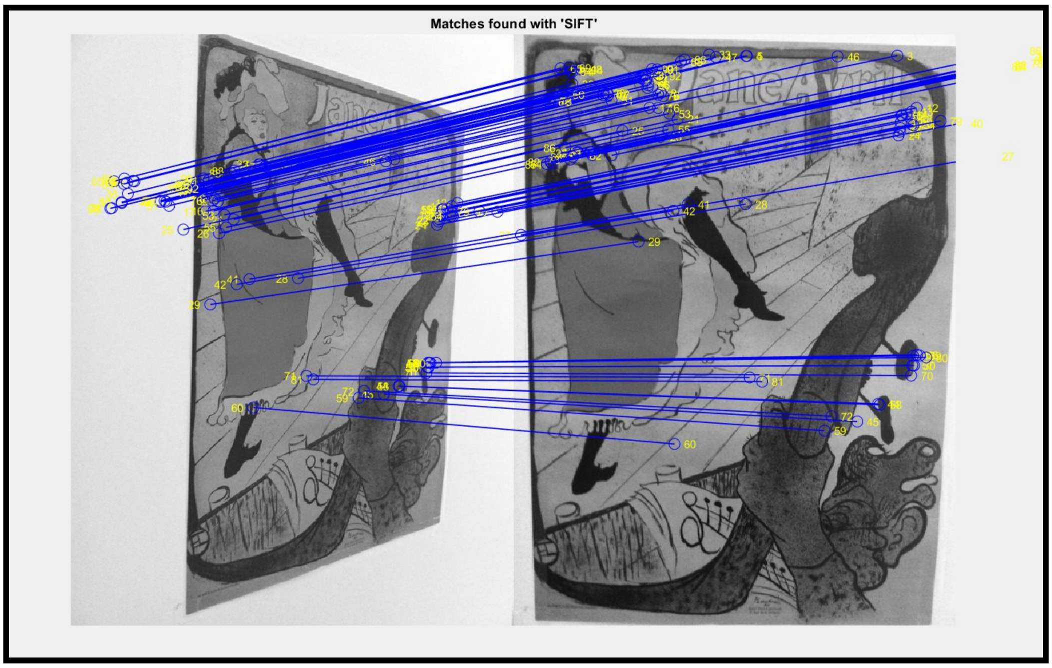

Figure 3.5: results of matching using SIFT descriptors (sigma=1.5, threshold =0.85)

```
Features matched: 45
```

Figure 3.6: total number of feature matched

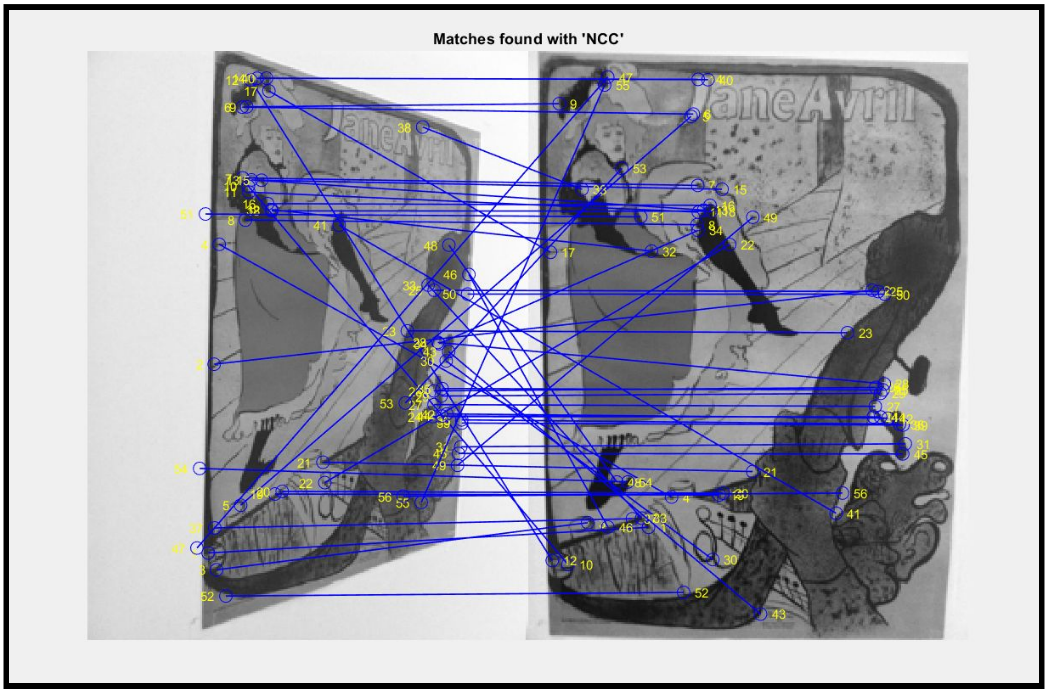

Figure 3.7: results of matching using NCC descriptor (sigma=0.1, threshold =0.8)

The second part of the analysis was carried out on the Image Retrieval procedure.
As already said, the provided functions allow to perform Image Retrieval using a representation based on SIFT descriptors and the bag-of-keypoints paradigm. Each image in the gallery is represented using a dictionary, after extracting its keypoints and computing the SIFT descriptors. The result consists in a histogram of the frequencies of all visual words appearing in it. The same representation is then obtained for a query image. This allows to compare, according to a certain metric, the query with the image representations in the gallery, thus retrieving the 10 images most similar to it.
Our goal was to observe the different results by changing the size of the dictionary: *Figure 3.8*, *Figure 3.9*, *Figure 3.10* show what happens when the size is changed from 30 to 500. We can observe that the distances between the query image and the first 9 images of the gallery generically increase with the size of the dictionary. This result is coherent with the fact that, by doing so, we obtain a more and more sparse representation of the images. This increasing distance allows to correctly highlight the fact that the query image is quite different with respect to the gallery images.
In addition we can observe that, when choosing a higher number of clusters, the closest image to the query actually becomes the most similar.
As a final step, these latter results are compared with the ones of the Image Retrieval based on the number of detected matches with the query image (*Figure 3.11*), for which we chose to compare SIFT descriptors with $\sigma = 0.1$.
We can observe that, although the two different strategies led to quite different results, some of the subjects in the 9 closest images are the same in both cases. In particular, the subject in the closest image remains the same.

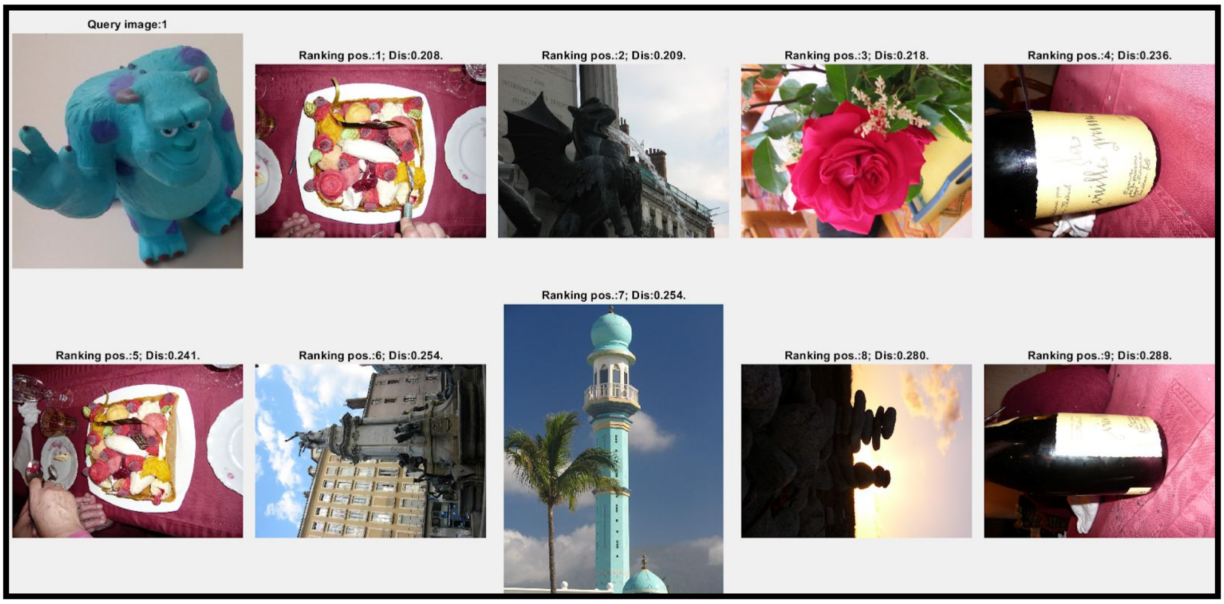

Figure 3.8: Image retrieval (number of clusters equals to 30)

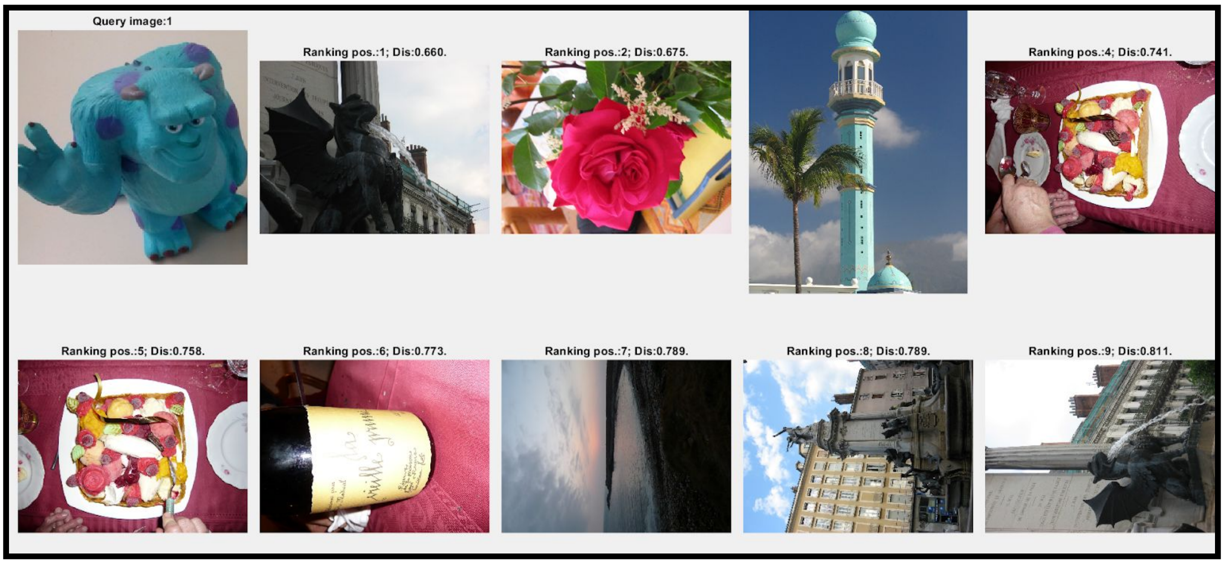

Figure 3.9: Image retrieval (number of clusters equals to 200)

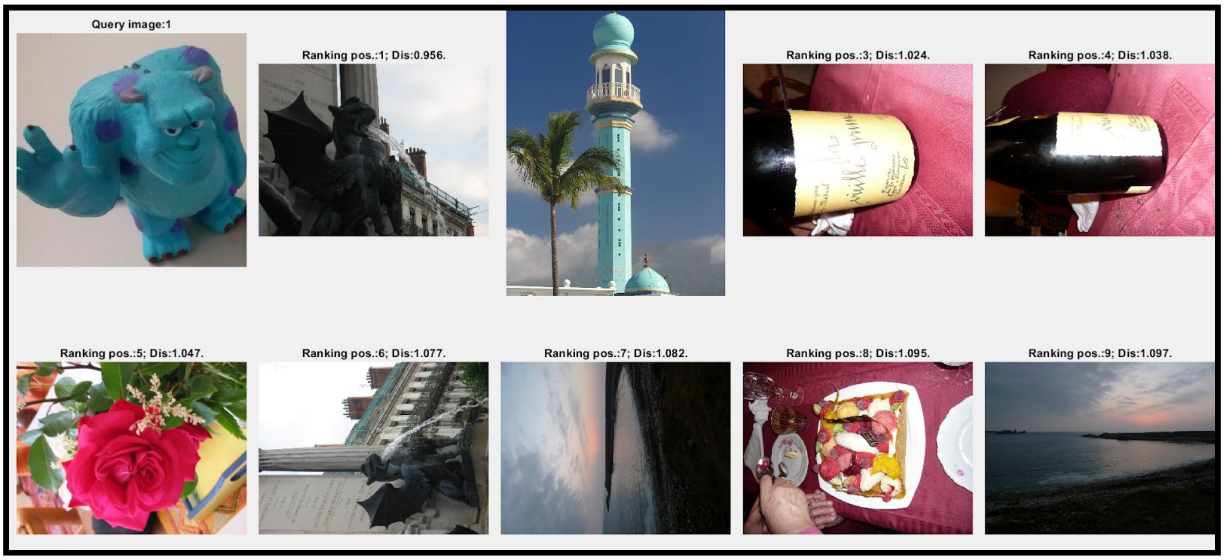

Figure 3.10: Image retrieval (number of cluster equals to 500)

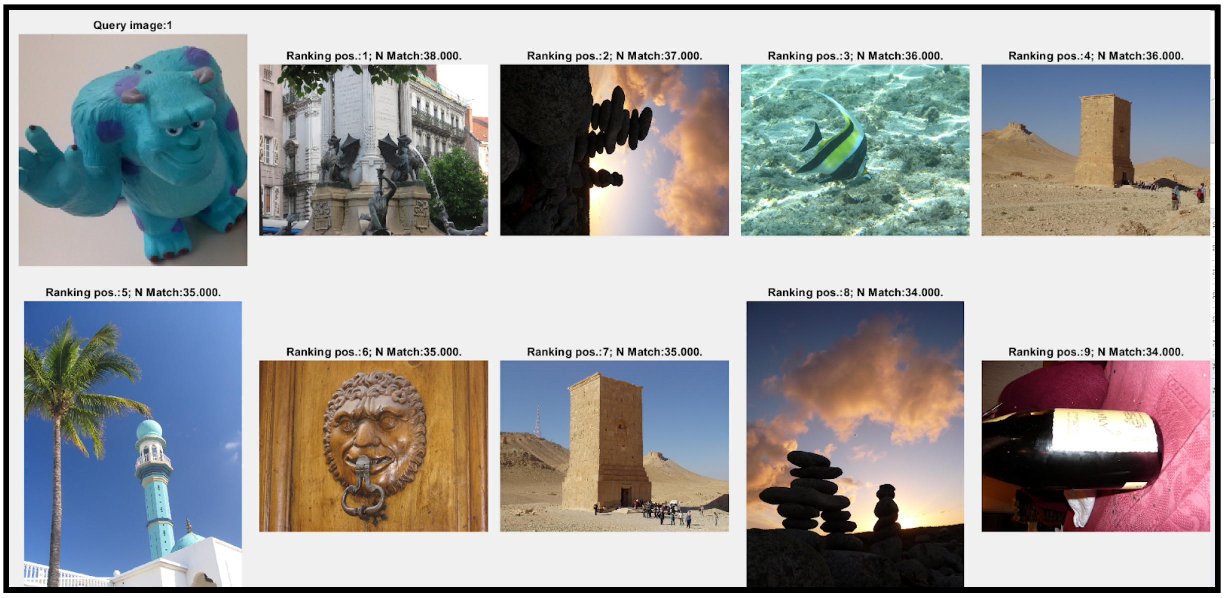

Figure 3.11: Monster with numMatches $\sigma = 0.1$)

4. Conclusions

The first image matching was carried out by combining together two similarity measures: the *position* and the *appearance* (using *Euclidean distance* and *Normalized-Cross Correlation*). This produced really satisfactory results, when it involved two image pairs whose differences only regarded a *translational* transformation. But then, when we considered a more general problem, that involved also changes in the rotation point of view and in the scale, we had to explore a different solution. This is the case of the *Scale Invariant Feature Transform* (*SIFT*). In the results we verified how it is fairly insensitive to view-point changes, and for this reason it led to far better performances, even though some inaccuracies are still visible.

In both cases, though, we experimented that the results are very sensitive to the value of the characteristic parameters (**σ** and threshold): a correct choice is what often determines the success of the procedure. These values also appear to be fairly related to the specific pair of images chosen.

In the second part of this experiment we represented a given image by quantizing its key points with respect to the dictionary. Doing so, we changed the size of the dictionary (by changing the *number of clusters*) in order to understand how this influences the goodness of the representation and of the sorting. We can state that with a low number of clusters, the result is less precise. However, since the query image was quite 'different' from all the gallery images, the visual result was not not exactly straightforward to analyse.

In the last part of the lab, a different strategy was adopted to perform *Image Retrieval*. Although the results (the 9 closest images found by the algorithm) were quite different, some analogies can be found in the two methods.