Report: Ninth Assignment
Image Matching and Retrieval

Giovanni Battista Borrè, Mario Ciranni, Tommaso Gruppi, Federico Tomat

June 24, 2019

1 Introduction

A feature is a piece of information which is relevant for solving the computational task related to a certain application. Features may be specific structures or set of distinctive keypoints in the image such as edges or objects. Features may also be the result of a general neighborhood operation or feature detection applied to the image. The features can be classified into two main categories:

- features that are in specific locations of the images, such as mountain peaks, building corners, doorways, or interestingly shaped patches of snow. These kinds of localized features are often called keypoint features (or even corners) and are often described by the appearance of patches of pixels surrounding the point location;
- features that can be matched based on their orientation and local appearance (edge profiles) are called edges and they can also be good indicators of object boundaries and occlusion events in the image sequence.

1.1 Image Matching

Features matching or generally image matching, a part of many computer vision applications such as image registration, camera calibration and object recognition, is the task of establishing correspondences between two images of the same scene/object. A common approach to image matching consists of detecting a set of interest points each one associated with image descriptors from image data. Once the features and their descriptors have been extracted from two or more images, the next step is to establish some preliminary feature matches between these images. Generally, the performance of matching methods based on interest points depends on both the properties of the underlying interest points and the choice of associated image descriptors. Thus, detectors and descriptors appropriate for images contents shall be used in applications. For instance, if an image contains bacteria cells, the blob detector should be used rather than the corner detector. But, if the image is an aerial view of a city, the corner detector is suitable to find man-made structures. Furthermore, selecting a detector and a descriptor that addresses the image degradation is very important.

1.2 Image Retrieval

Image retrieval is a task of searching similar images of a certain type from the image datasets. In recent years real-life applications of image retrieval has gained a great interest in the research area. Content-based image retrieval (CBIR) is a widespread technique gradually applied in retrieval systems. In CBIR, images retrieval is done by using visual characteristics also known as features, extracted from the database. CBIR system's retrieval accuracy and efficiency rely greatly on the adopted visual feature. Visual features may describe numerous properties of either low-level features include shape, color, spatial relationship and texture, or high-level features also called semantic features. Earlier research in CBIR motivate encoding the spatial arrangement of colors due to the problem of having several diverse images with identical or similar color histograms. Now this problem is being reconsidered with the visual dictionary model which takes documents as bag-of-words. This model has numerous important advantages including compactness and invariance to the image or the scene transformations and is one of the most famous feature representations in the CBIR framework. The image retrieval algorithm receives as input a query image and many gallery images, then it orders the gallery images by similarity with respect to the query image. So its goal is to retrieve images in a database based on their visual similarities with a query image, this algorithm is often used in object recognition

2 Result

2.1 Image Matching Evaluation

For the evaluation of the results coming from the image matching algorithm we have decided to plot the number of matches with respect to the variations of the sigma's and the threshold's values as shown in Figure 1-2. It can be noted that using the SIFT descriptors the number of resulting matches are always bigger with respect to the ones found using the NCC descriptors. Another relevant aspect is that matching using the NCC descriptors is really more affected by the variations of the threshold with respect to the ones deriving from the usage of the SIFT descriptors: in this last case in fact the matches are always more than 200 also if the threshold is equal to 0.9 (a very high value) while with the NCC descriptors the matches decrease hugely at each decrementation of the threshold' s value as reported in Figure 1, in particular as soon as this one exceeds the 0.5 value. Moreover if the threshold is equal to 1 using the NCC descriptors no match will be found while using the SIFT descriptors about 50 matches are found, this is not a good result overall but in this case the threshold is very high and so also this one can be considered as an appreciable result, so also in this case the usage of SIFT descriptor is the best choice. Therefore for values of the threshold lower than 0.5 the NCC descriptors can be used and the results will be acceptable and just a little bit less good than using the SIFT descriptors, while if the threshold increases more and more the validity of the results decrease hugely with the NCC descriptors and so using the SIFT ones will be the best choice in order to find more matches between the two input images.

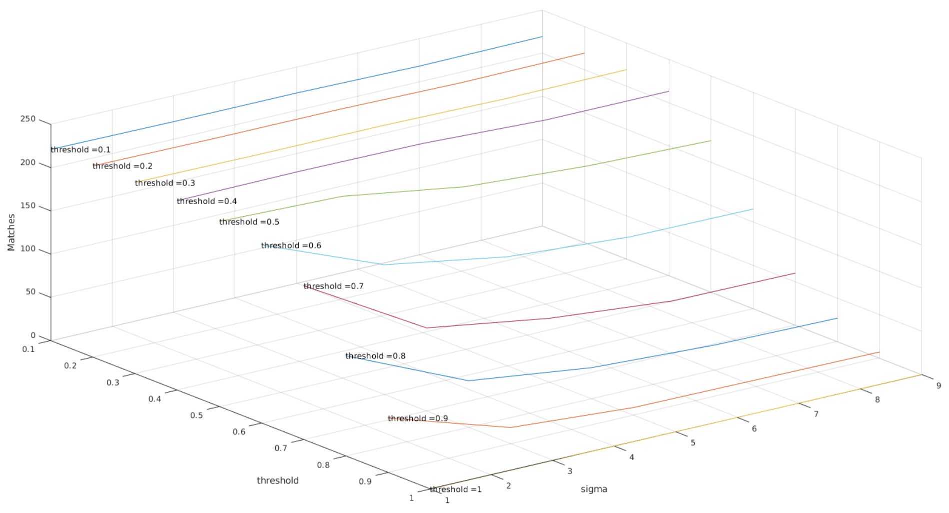


Figure 1: Features matched with NCC descriptors

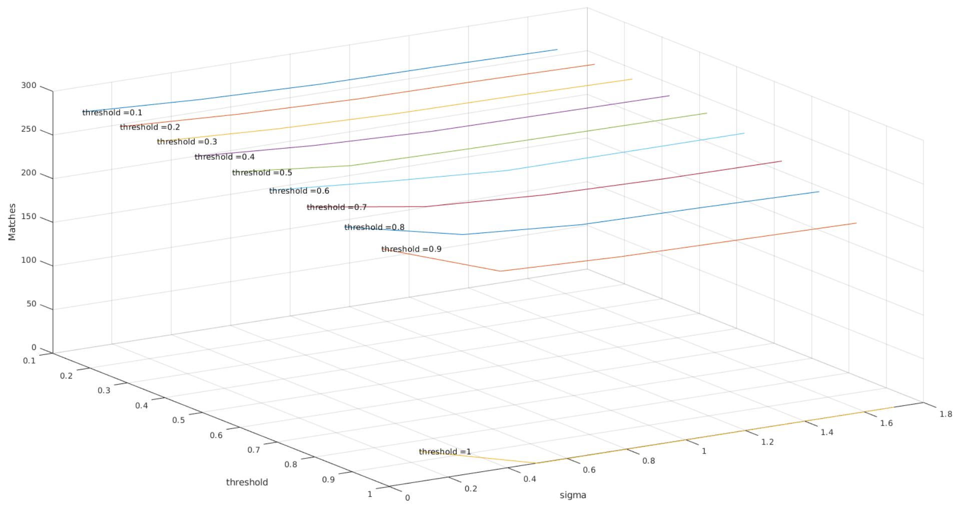
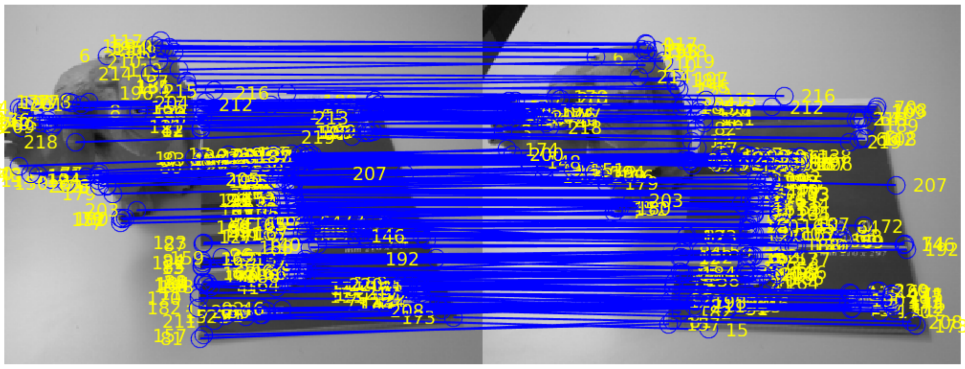


Figure 2: Features matched with SIFT descriptors

The images representing the matched features from which our analysis come out are more than 100 and are collected inside the repository located here. Some significant results retrieved from that repository are shown in the following figures (Figure 3-4) in order to support our evaluations.



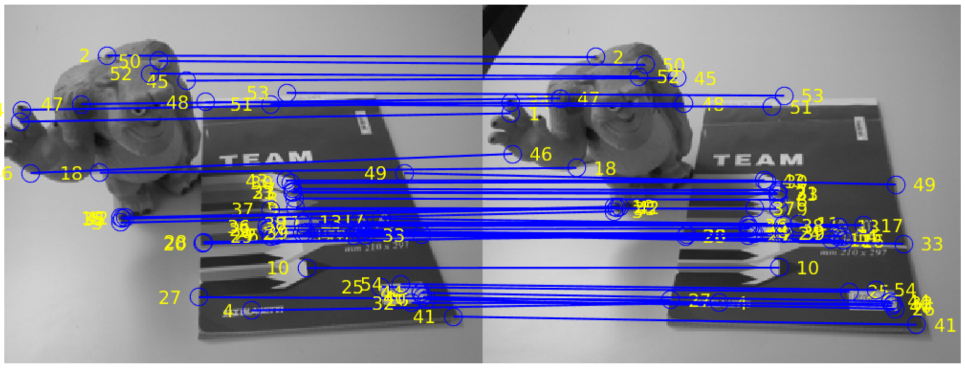
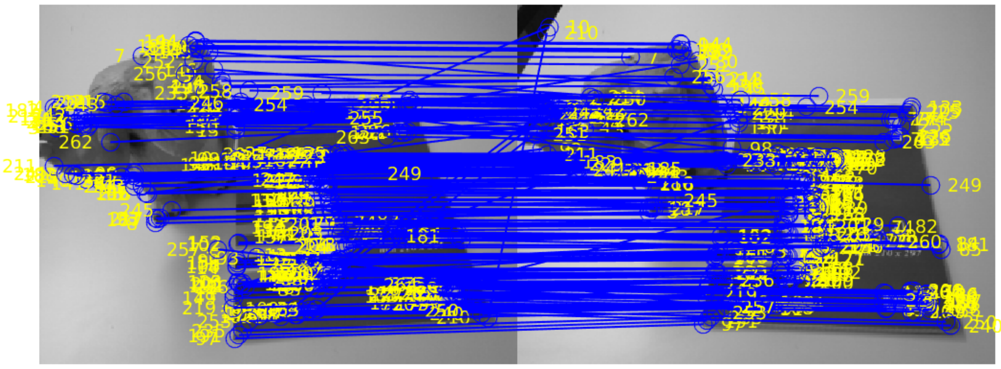


Figure 3: matches using SIFT descriptors with sigma=3 and thresholds respectively equal to 0.4 and 0.8



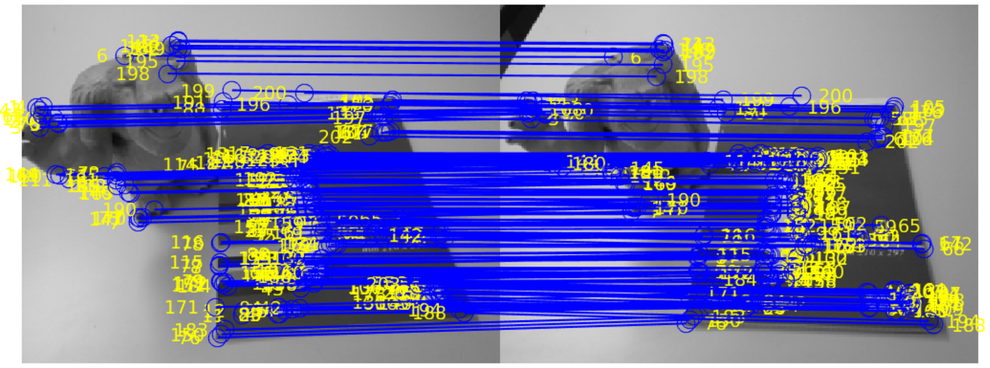


Figure 4: matches using SIFT descriptors with sigma=0.5 and thresholds respectively equal to 0.1 and 0.9

2.2 Image Retrieval Evaluation

In order to evaluate the performances of the image retrieval algorithm we have decided to use the average error ϵ such defined:

$$\epsilon = Avg(I(y_i \neq \hat{y}_i))$$

Where $\hat{y}_i$ is the the first ranked image estimated by the algorithm and y_i the true better matching image. As shown in the following graphics (also stored here) the variation of the dictionary has a greater influence, on the time taken for the computation as it can be noted from Figure 5 and in the precision of the results as shown in Figure 6. From these figures it can be noted that the time needed for the computations grows in a linear way, and also the precision of the results grows while increasing the dimension of the dictionary with an exception for a value equal to 300.

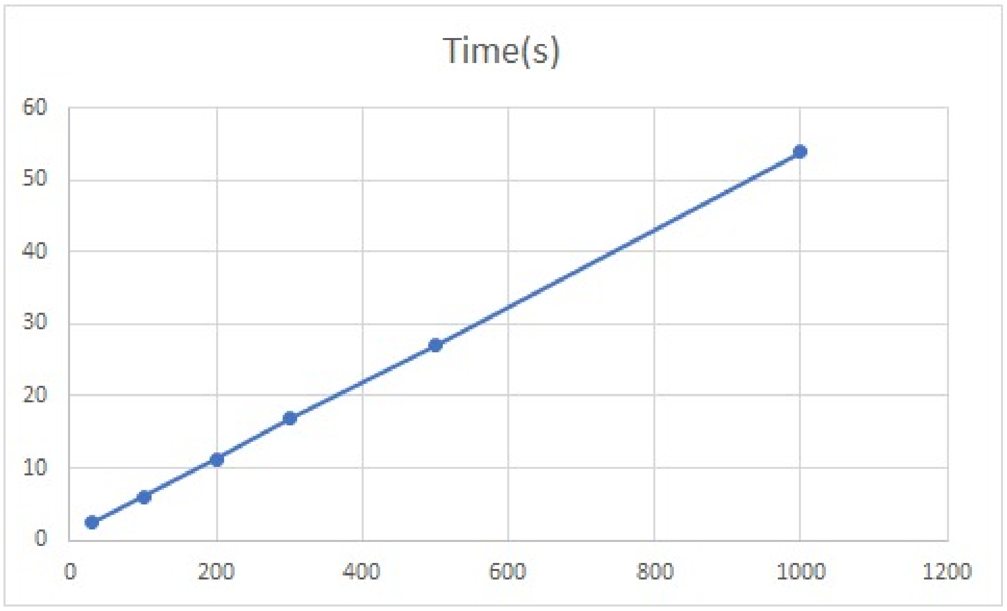


Figure 5: Computational time needed by varying the dimension of the dictionary

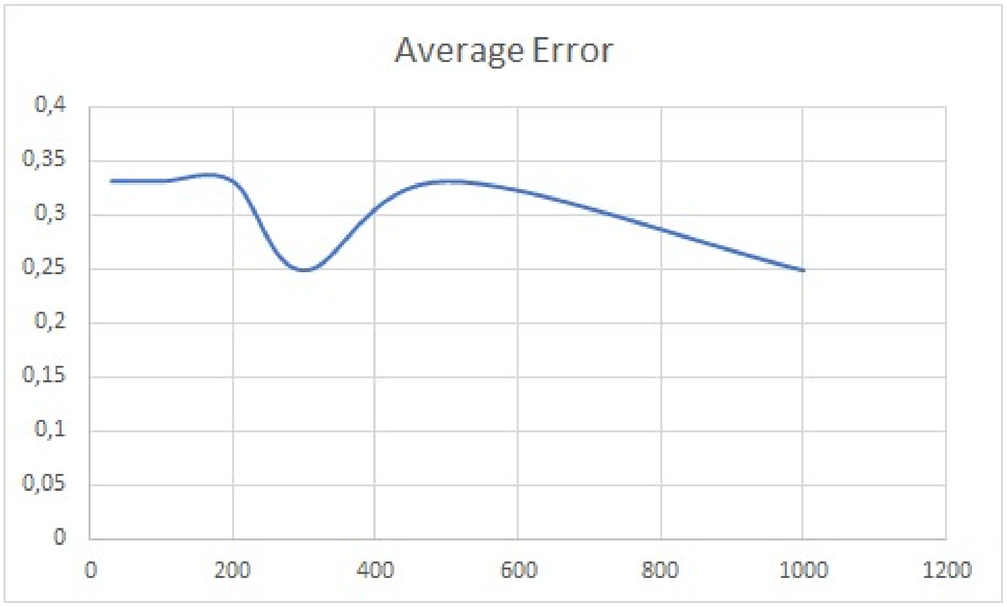


Figure 6: Average error resulting by varying the dimension of the dictionary

2.3 Image Retrieval through Image Matching Evaluation

In order to accomplish this we setup a procedure to perform image retrieval using the number of detected matches (using the less time comsuming descriptr: SIFT descriptor): so given a query image to compare it with all the images in the gallery, and sort them in decreasing order with respect to the number of detected matches. This is always done correctly by the algorithm that shows the image matching between the query considered and the images belonging to the gallery and ordering them in a decreasing way as requested. We use a threshold set to 0.75 and $\sigma = 1.2$.

Then it is requested to compare the results of image retrieval based on the procedure descripted above and the one provided and based on the bag of key points. In many cases the order of the images changes, mostly for images below the fourth position, and it seems to have better results as shown in the following figures stored here.

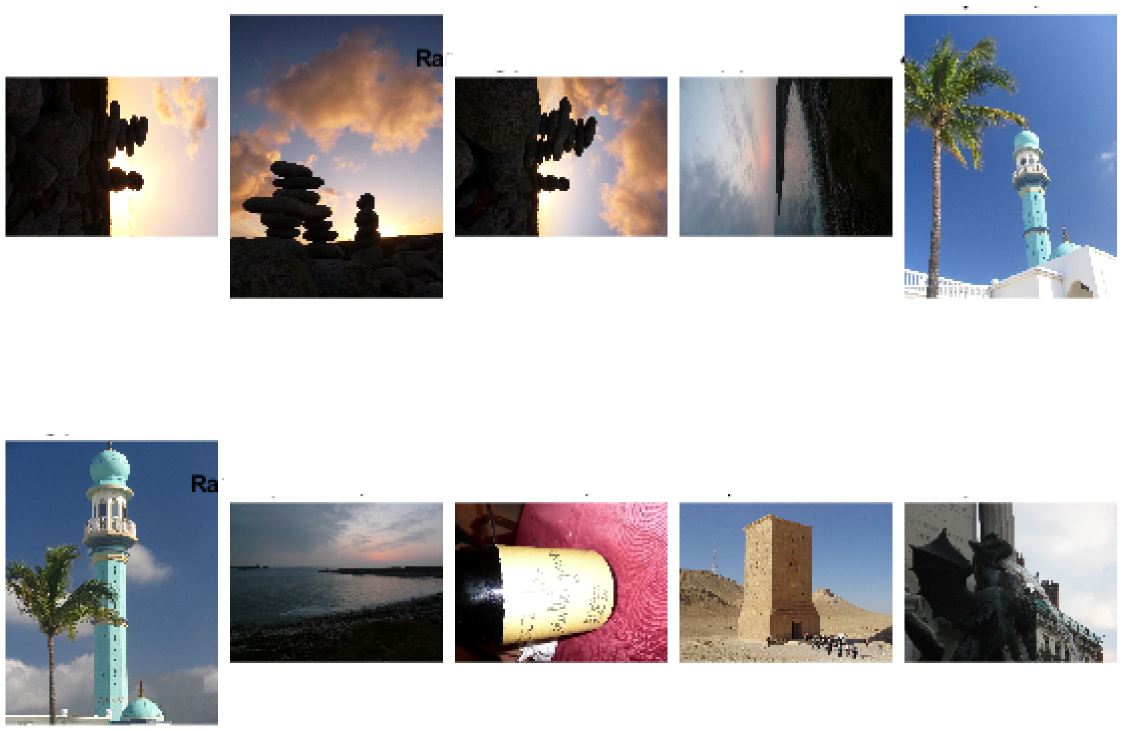


Figure 7: results from the image retrieval version provided

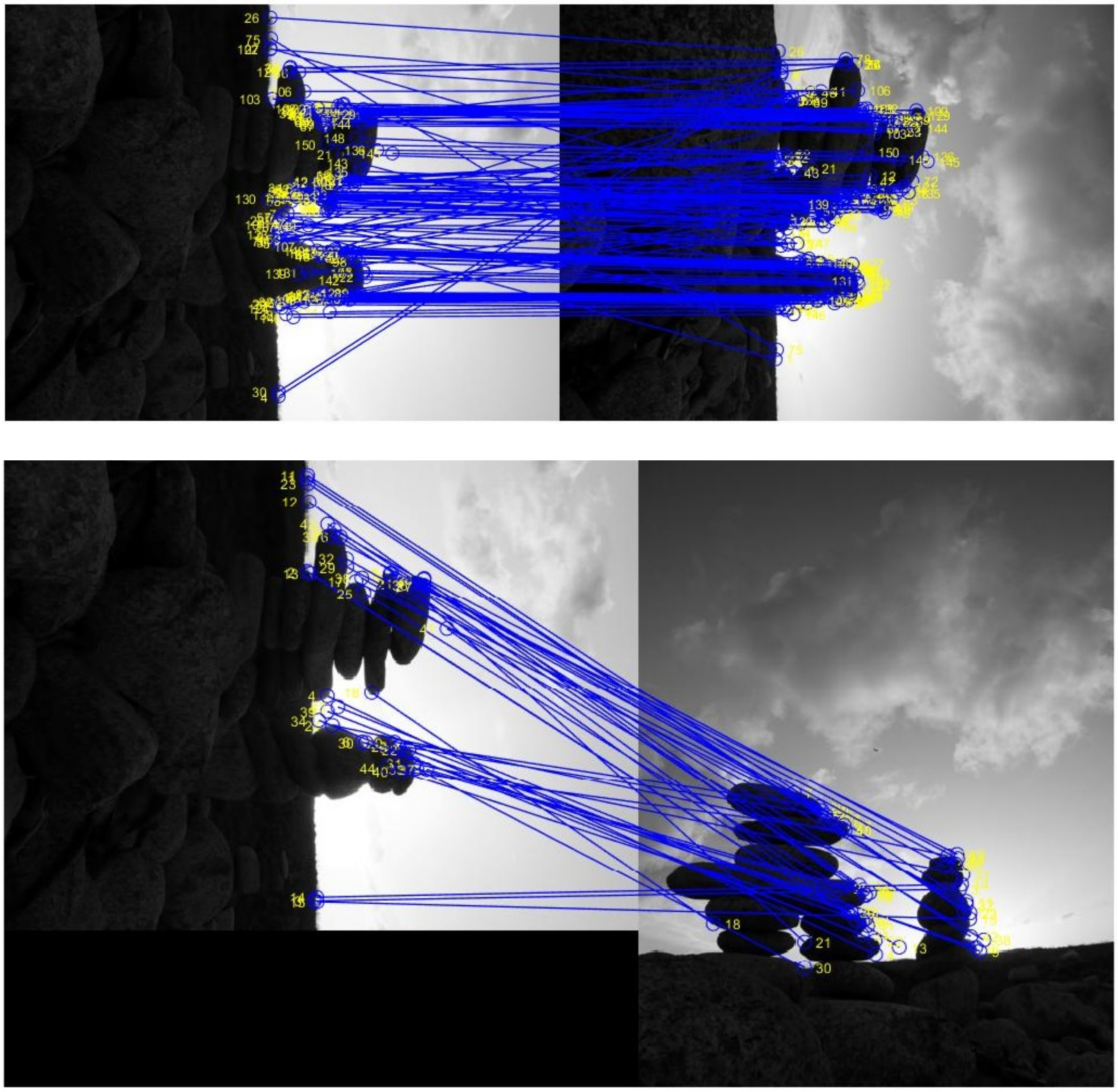


Figure 8: first and second output images deriving from other implementation

In these images it can be noted that the first output image is the one oriented as the considered one instead of the one in vertical position. From this fact (that happens also in other cases) we can conclude that the new implementation works better than the previous one but a more detailed analysis should be done in order to confirm this fact.

3 External links

https://drive.google.com/drive/u/1/folders/19d-GusAg4qS2nwuUu6-lvdGQb1jzjcFl

https://drive.google.com/open?id=19q9b9tVuD5kCUkpQagUKdohyQ3KzY7lY

https://drive.google.com/open?id=1gy-Q4JCWlRWH3kXlx8fFN5GLRVOM4fC7

4 GitHub Repository

https://github.com/federicotomat/Computer-Vision

References

[1] Richard Szeliski, *Computer Vision: Algorithms and Applications*, (University of Washington, 2010).

[2] Francesca Odone & Fabio Solari, *Computer Vision Course: Image Foundamental*, (University of Study of Genoa, 2019).