Question 3

(a) The images were taken

(b) Using the provided vlfeat library features and descriptors have been extracted.

- Each of the feature vectors is a 4-row vector: $\begin{cases} (f(1:2)) - location\ of\ the\ feature \\ f(3) - scale \\ f(4) - orientation \end{cases}$
- Each of the descriptor vectors is a 128 elements vectors, describing orientation histogram (for 8 orientations) for each of the 16 cells in the vicinity of the feature (16 * 8 = 128).

The vlfeat library allows the usage of customized parameters for feature extraction:

- Peak threshold
- Edge threshold

They define the threshold for a potential feature to be declared a feature. In our example, 57 features were extracted for #1 and 64 features were extracted for Image #2:

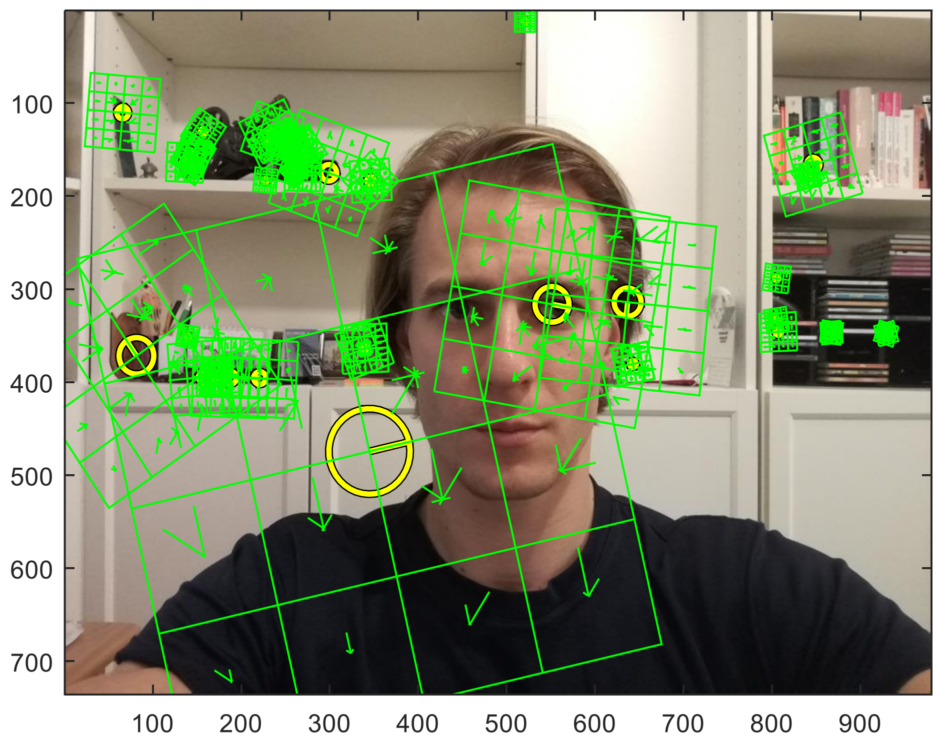


Figure 1 Image 1: 57 features detected

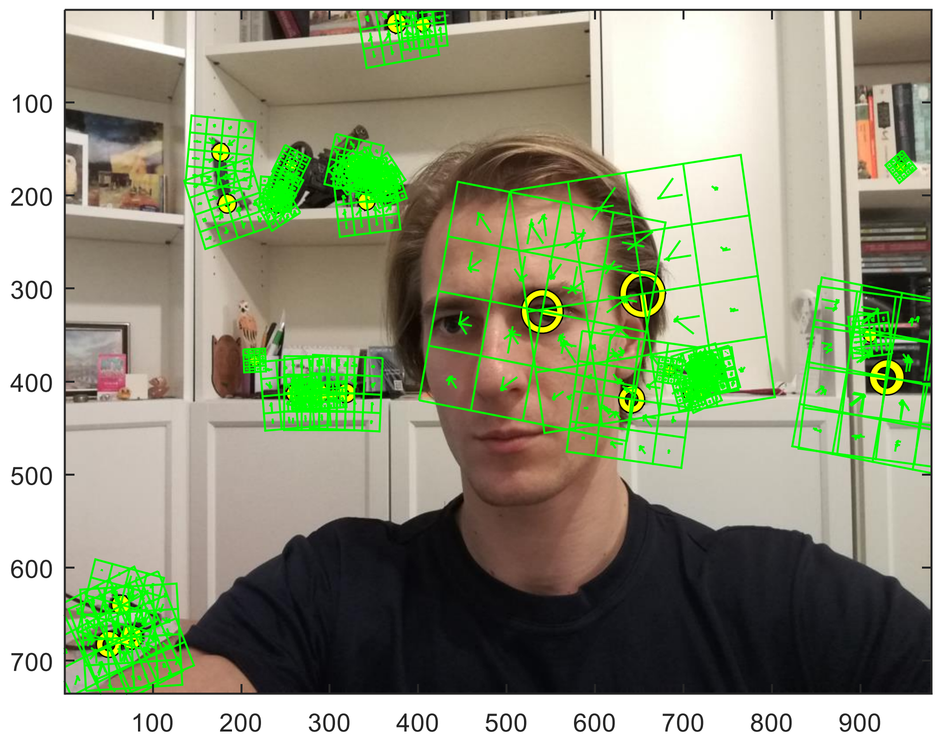
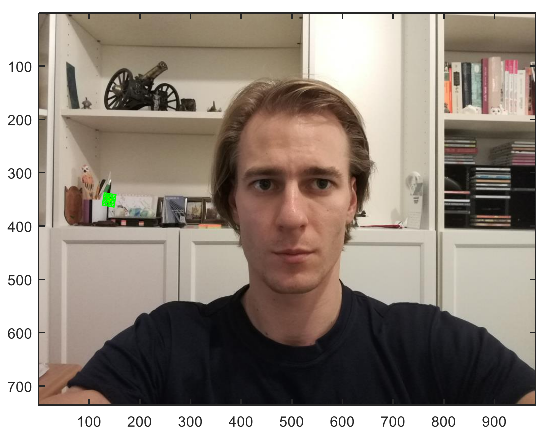
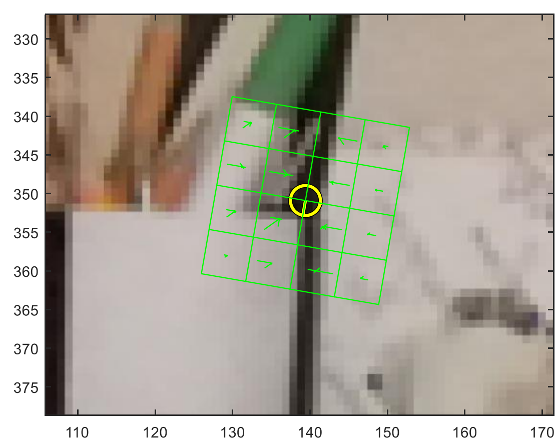


Figure 2   Image2: 64 features detected

Scale and orientation for a representative feature: select arbitrary index 1. The feature 1 on Image1 is the following: (right image is the close-up)



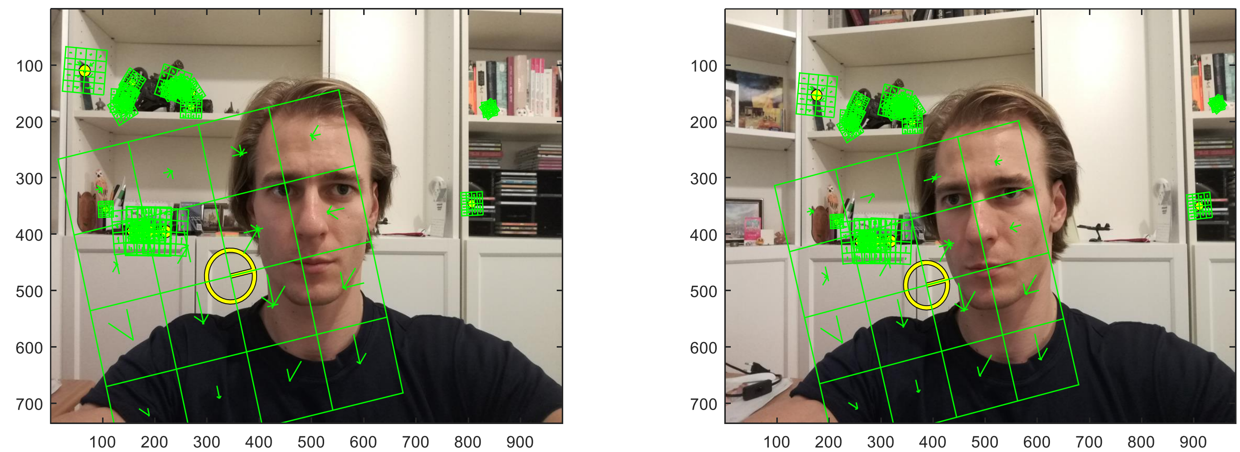


The representative feature is then : $\begin{cases} x \\ y \\ scale \\ orientation \end{cases} = \begin{cases} 139.4 \\ 350.91 \\ 1.937 \\ 0.172 \end{cases}$; with scale =1.937 & orientation = 0.172

(c) The vlfeat library does possess the function for the feature matching, but since the question asks to perform the calculation as explained in class, the algorithm is written manually, while the library function helps us to validate the results. The algorithm for feature matching is the following:

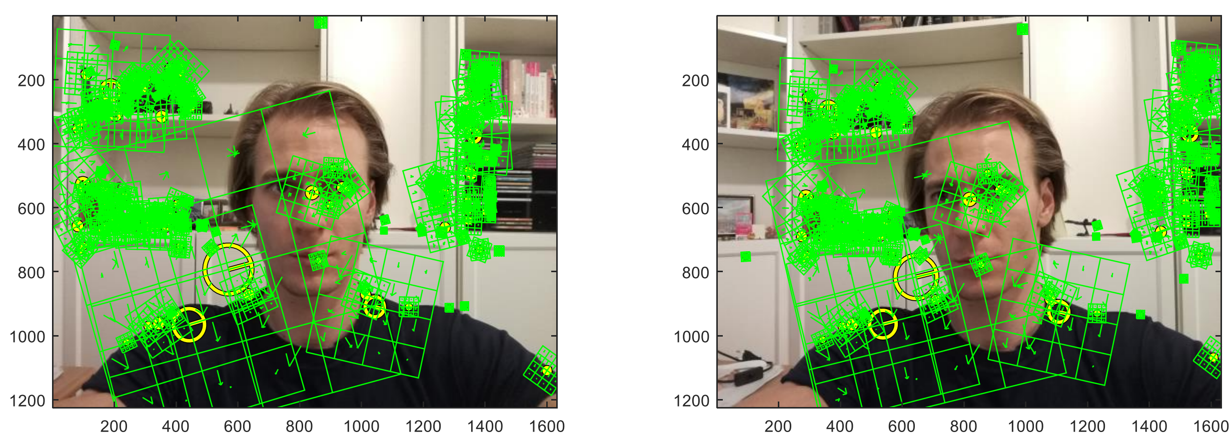
1. Calculate the Euclidean distance from every feature descriptor (128 row vector) in Image 1 to every feature descriptor in Image 2.
2. For every feature in Image 1, find the minimum Euclidean distance out of Euclidean distances calculated to every feature in Image 2 (calculated in step 1)
3. Use the threshold parameter to validate this feature is “unique” enough. Meaning, the minimal Euclidean distance multiplied by threshold is still smaller than any other Euclidean distance to other features. Default value in vlfeat library is 1.5
4. If the condition in Step 3 holds, add this couple of matched features (their indexes) to the matched array.

The features matched by this way using a certain threshold (1.5) may be observed in the following image:



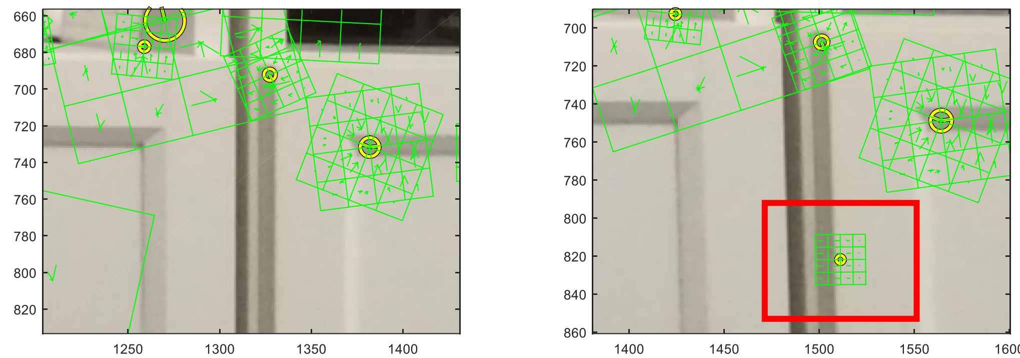
By the first look, it is indeed observable that those features are corresponding to each other. No outliers were found on this image, since the threshold parameters were set quiet to make the features amount small (for easier visual inspection).

In order to find the outliers, the higher resolution image was used, with smaller threshold parameters for feature extraction. As a result, in the following specific example, for Image 1 and 2 : 1188 and 1210 features were extracted, with 237 matches:

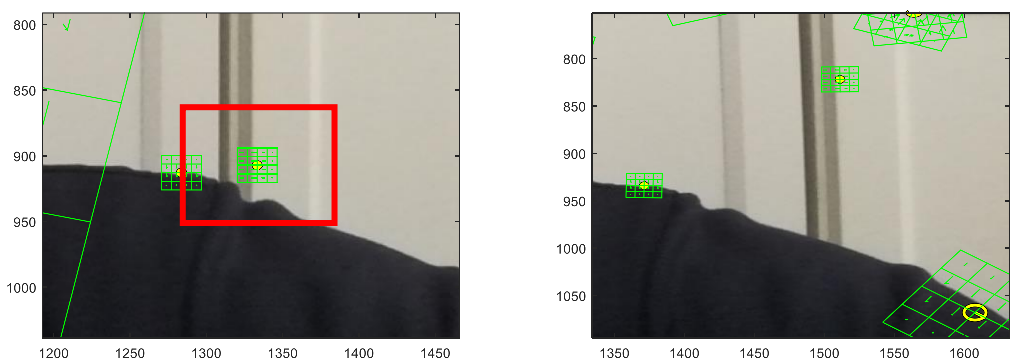


After the fast inspection, the following are the two examples of the outliers: (marked with red boxes) – meaning this feature is identified as match with some other feature from another image, but as we can see, it does not appear in the second image where it should be.

Outlier 1 :

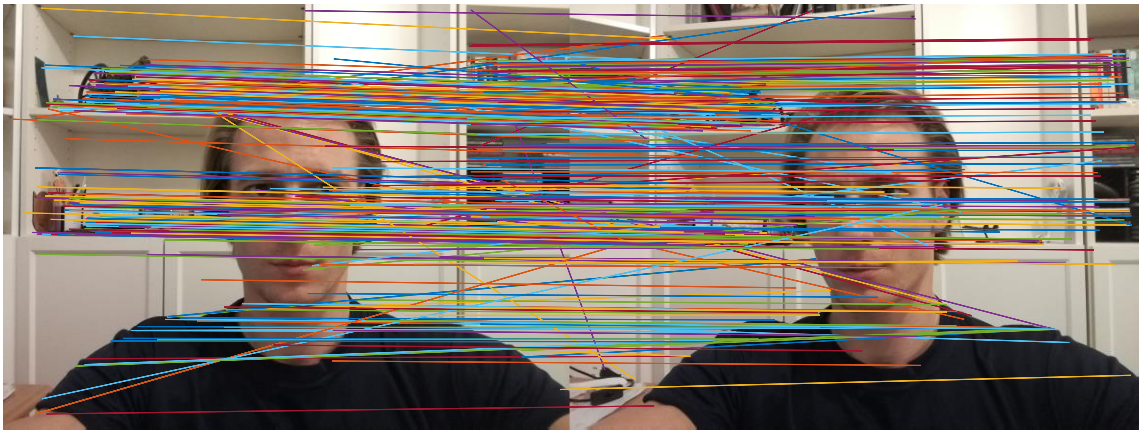


Outlier 2 :

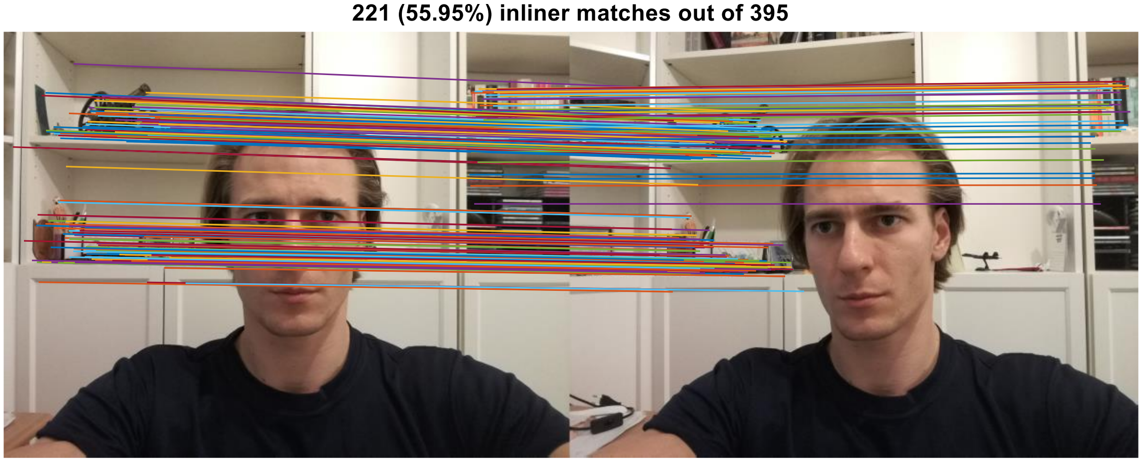


Using another function, provided by the vlfeat library (although which does not appear in the standard library, but its code is public on their website), the “SIFT_MOSAIC”, we may witness the outliers more clearly. In the following image it may be visible how some tentative matches are found, but later declared “outliers” and do not appear in the “inliers” section. The percentage ratio of the inliers also shown, allowing to adjust the parameters:

395 tentative matches

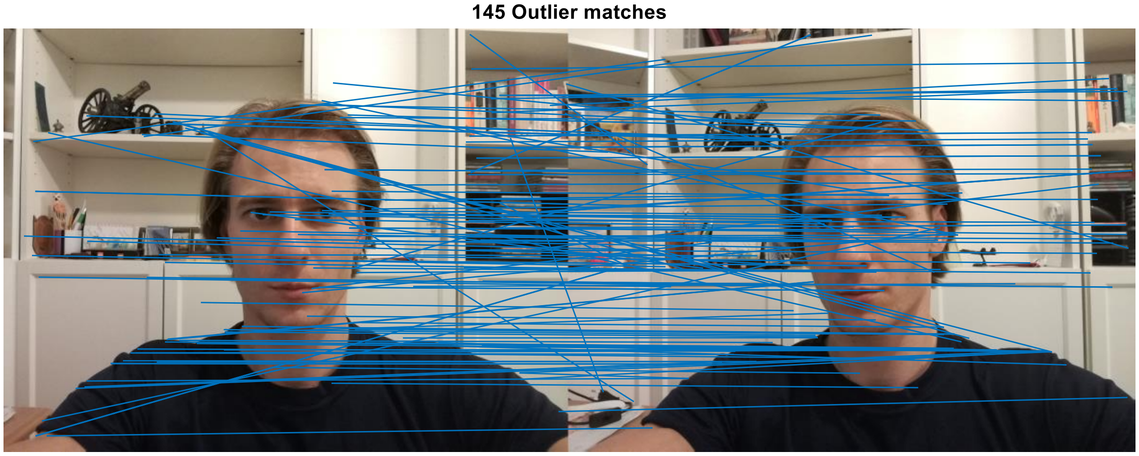


221 (55.95%) inliner matches out of 395



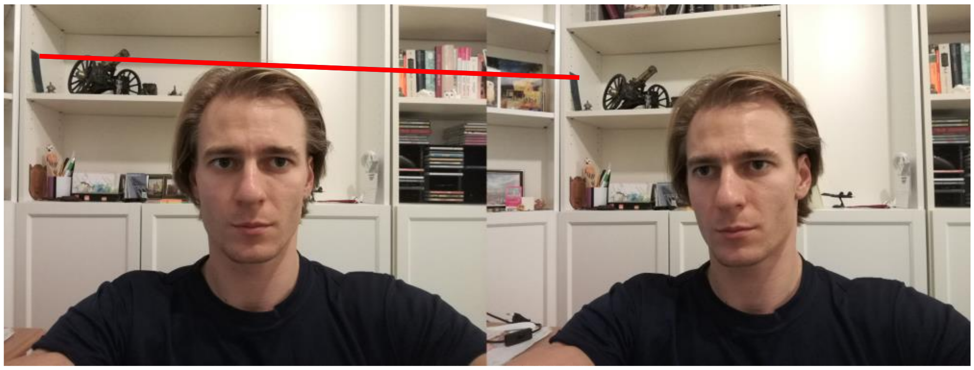
Plotting just the outliers gives the following figure.

145 Outlier matches



Answering the question, we represent some representative inlier and outlier matches, and discuss possible reasons for their appearance (the outliers):

Inlier representative match: (1st figure shows the general picture, 2nd picture the close-up)



Close-up (looks almost same on both pictures, surely):

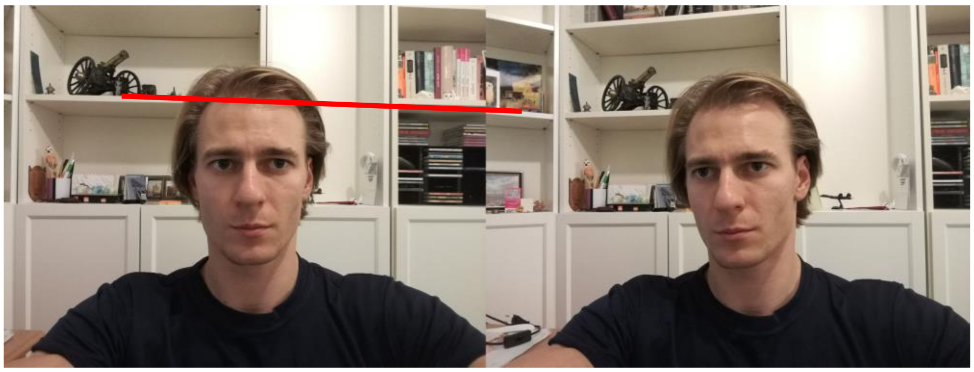
Outliers' representative matches: although apparently Euclidean Distance between 2 points represented here was small enough for this couple to be declared a "match", we can see that this is certainly not the case, and the point is not the same.

Example #1:

Close-up Left Image:

Close-up Right Image:

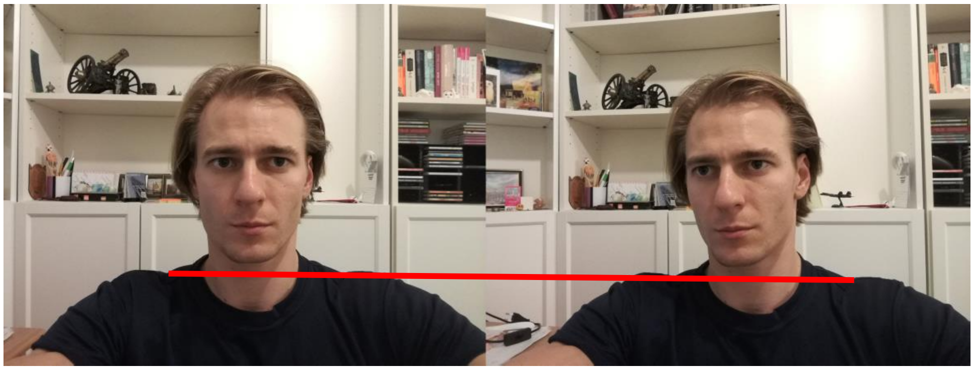
Example #2:



Close-up left image:

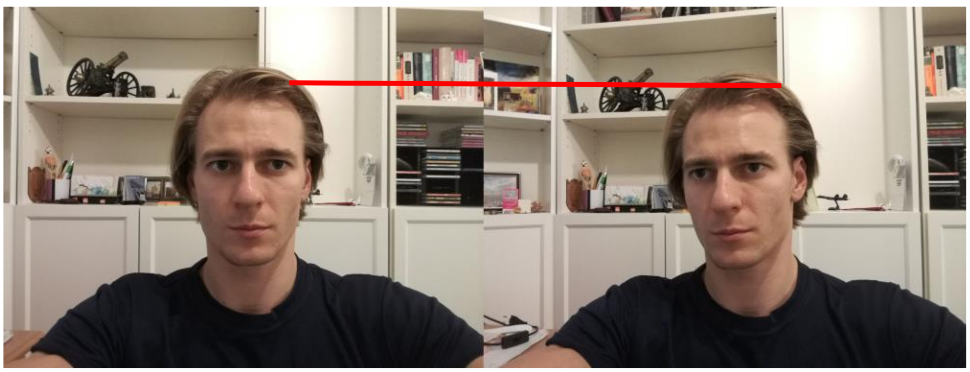
Close-up right image:

Example #3



I think, no close-up image needed here.

Example #4:



Explanation for possible outliers appearance: several reasons may lead to the creation of the outliers. Here are presented some we have come up with:

1. Since Feature description use the SIFT technique, which is based on the edge orientation, same features may have similar descriptions, although are in fact representing completely different objects. (may be seen in Example 1&2).
2. For the same reason as in 1, but the objects are indeed similar in the image, or appear several times in the same image – both having the similar edges. For example, the shelves behind me, or the eyes. Example 3 shows
3. The environment is dynamic. Each frame is taken at different camera pose, and at different time. A similar feature may appear in a certain image, while it wasn't present in the previous image, but a feature with similar description was. This may be the reason for Example 4 (but not 100%)