MAI-OR Practical Block 1: **Landmark Detection and Descriptors**

Gonzalo Recio and Jana Reventós— March 2020

1 Introduction

Robust image matching is a very important task in computer vision and robotics applications. In this practical session we compare the performance of two different image matching techniques (in our case SURF and BRISK) against different kind of transformations such as scaling, rotation, blurring, change in 3D viewpoint and contrast changes. With this purpose we aim to study which are the landmark detectors most suited depending on the problem and image of interest. We used two different images, a corner and a blob based image, and we manually apply the different types of transformations on original images and compute the matching evaluation parameters such as the number of key points, the execution time required for each method to show which algorithm is the best more robust against each kind of distortion.

1.1 Test data-set and image transformations

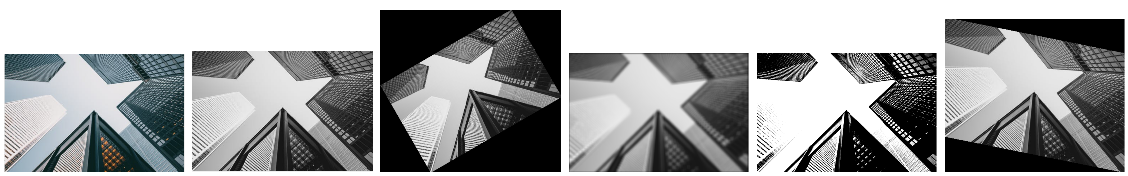


Figure 1: Original corner image and transformations (scale, rotation, blurring, contrast and 3D viewpoint)

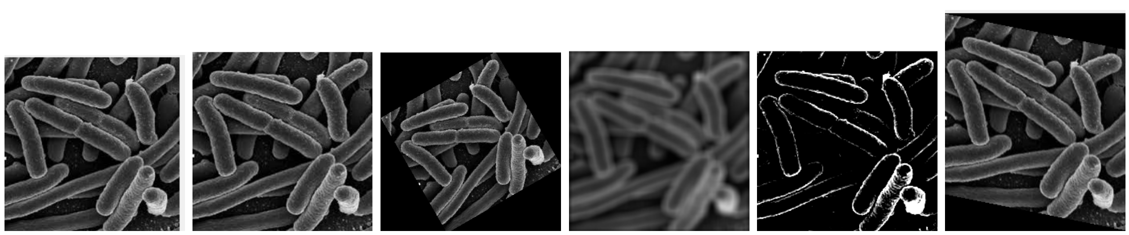


Figure 2: Original blob image and transformations (scale, rotation, blurring, contrast and 3D viewpoint)

2 Landmark Detection and Descriptors

Keypoint detectors are in charge of identifying driscriminative and characteristic points on images. Overviewing the feature detectors, we can observe that there are two big groups based on their type (corner based or blob based). Depending on the image we are analyzing a feature detector might be more suitable than an other.

- **Corner based**: They are more suitable for images containing corners. Methods like FAST, ORB, Harris corner detection, BRISK.
- **Blob based**: Suitable for images containing circular or blob shapes. Methods such as SURF, KAZE.

The task of descriptors is to transform a local pixel (keypoint) neighborhood into a consistent vector representation which allows comparison between neighborhoods invariant to changes (scale, orientation, illumination, etc.). There are two main types of descriptors robust to changes:

- **Gradient descriptors**: based on local gradient computations such as SIFT or SURF.
- **Binary descriptors**: based on pairs of local intensity differences and encoded into binary vectors, such as BRISK, ORB or FREAK.

3 Experiments

Our idea is to take a method of each type (corner/blob and gradient/binary) and compare their robustness in different image changes and distortions. In our case, we are going to use two well known pipelines: SURF and BRISK. Both methods are a complete pipeline that involve feature detection and description. However, they are both quite different in terms of landmark detection and description: SURF has a blob detector and gradient descriptor while BRISK involves a corner detector and a binary descriptor.

3.1 SURF

SURF (Speeded Up Robust Feature) is a feature detection and descriptor based on the SIFT pipeline, but with with faster performance [1]. It uses a blob based feature detection and a gradient based feature descriptor, obtaining scale invariant features.

3.2 BRISK

BRISK (Binary Robust Invariant Scalable Keypoints) is a uses a corner based feature detection and a binary based scale-invariant feature description. From 2011, authors claim to be a faster method than SURF with same quality in results [2].

4 Comparison between descriptors

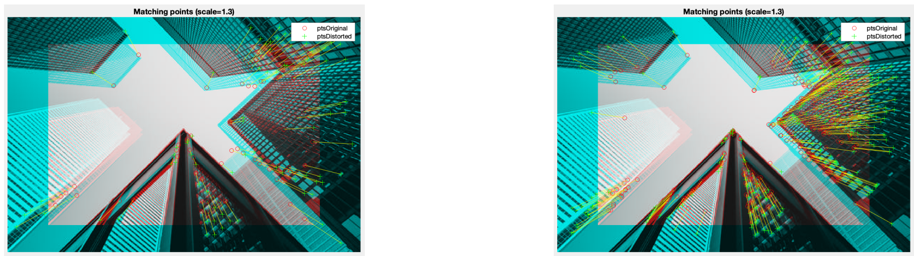
In order to do a comparison of the descriptors performance in different image distortions (scale, rotation, blurring, illumination and 3D viewpoint) we analyzed the results when using a corner based image and a blob based image. Furthermore, we wanted to study the image matching results for different levels of each distortions. In the following tables we present the results obtained for each image distortion. Each table contain the descriptor used, type of image, computational time, key points in the original image (Kpoints1), key points in the disorted image (Kpoints2), number of matches and a Match Rate (%) to indicate the ratio of useful keypoints detected by an algorithm that the end are being matched, which is computed using the following formula:

$$\%\ MatchRate = \frac{\#matches}{(Kpoints1 + Kpoints2)/2} \times 100$$

4.1 Scale

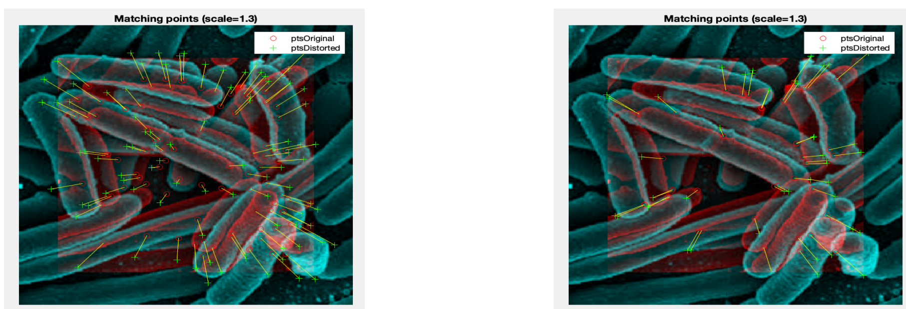
The images with varying scale dimensions were used to compare both algorithms, SURF and BRISK, and results are presented in table 1 and figures 3, 4. In the table we show the results using the corner and blobs based images scaled by 0.7 and 1.3 times with respect the original. However, figures only show the result for the 1.3 scale change.

For corner-based images with a different scale with respect to the original image, BRISK provides the highest number of matching points for the two scale cases. For blob-based images is the SURF descriptor is most robust. Computational time requirement for SURF is the least for the two types of images. As we can observe in figure 3 the BRISK detector identifies much better the corners of the buildings and widows in comparison with the SURF detector. On the other hand, when it comes the case of analyzing a blob based image, figure 4, we can see that the SURF descriptor detects easily the shape of the rounded bacteria's.



(a) (b)

Figure 3: The final matched points of varying scale (1.3) images using SURF (a) and BRISK (b) in a corner based image.



(a) (b)

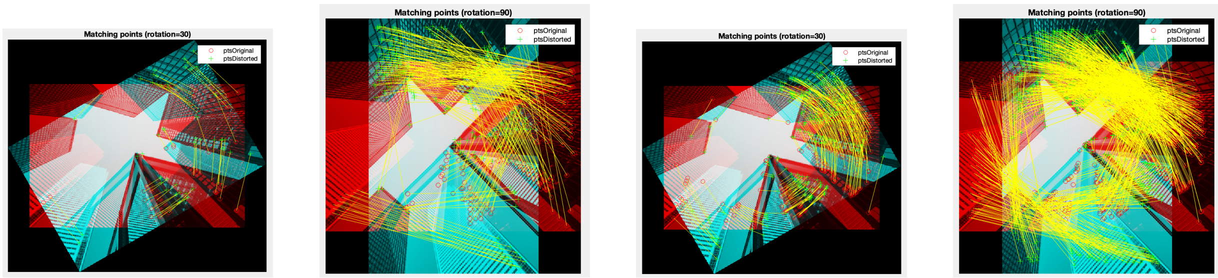
Figure 4: The final matched points of varying scale (1.3) images using SURF (a) and BRISK (b) in a blob based image.

Descriptor	Image	Scale	Time (sec)	Kpoints1	Kpoints2	Matches	Match rate (%)
SURF	corners	0.7	0.612	244	110	127	71.8
BRISK	corners	0.7	1.291	1406	488	252	26.6
SURF	corners	1.3	0.544	244	413	163	49.6
BRISK	corners	1.3	1.210	1406	2089	437	25.0
SURF	blobs	0.7	0.512	173	103	127	92.0
BRISK	blobs	0.7	1.114	132	70	58	57.4
SURF	blobs	1.3	0.450	173	241	133	64.3
BRISK	blobs	1.3	0.906	132	145	60	43.3

Table 1: Results of comparing different parameters depending on scale changes and descriptors.

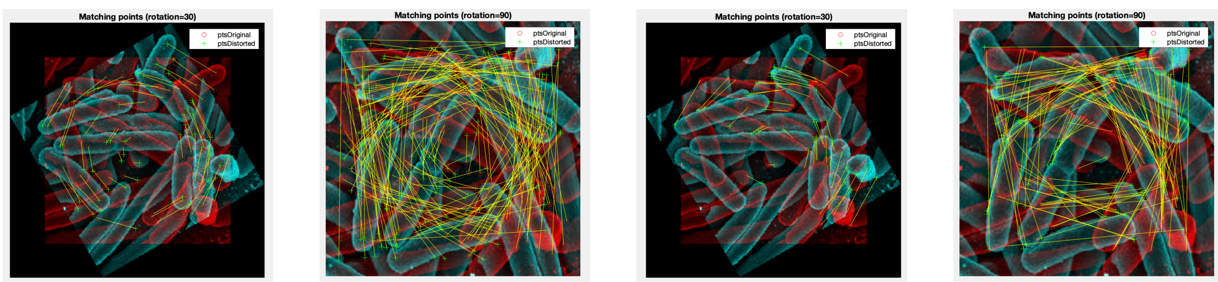
4.2 Rotation

We consider here rotations of 30^o and 90^o to the image to be matched with respect the original. The results are given in the table 2 and figures 5, 6. When comparing the matches between corner and blob based images we can observe the same behavior as before. SURF descriptor is more robust for blob-based images whereas BRISK is more robust for corner based images. The same happens with the computational time cost, SURF is always the fastest. In addition, we can see that with rotation angles proportional to 90 degrees the both descriptors identify much more matches than when a rotation of 30 degrees is done. Furthermore, the good performance achieved by SURF method may be attributed to the gradient-based descriptor that is more robust to rotation deformations (rotation invariant).



(a - 30°) (a - 90°) (b - 30°) (b - 90°)

Figure 5: The final matched points of varying rotation images using SURF (a) and BRISK (b) in a corner based image.



(a - 30°) (a - 90°) (b - 30°) (b - 90°)

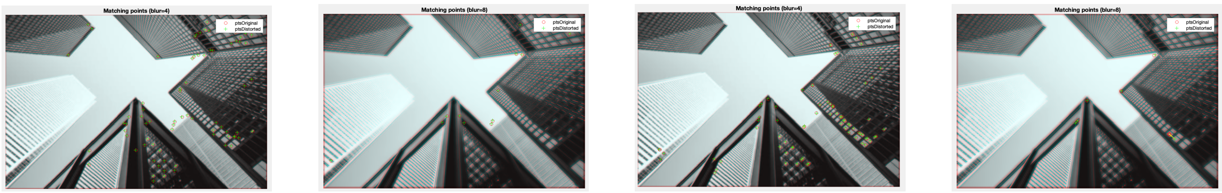
Figure 6: The final matched points of varying rotation images using SURF (a) and BRISK (b) in a blob-based image.

Descriptor	Image	Rotation (°)	Time (sec)	Kpoints1	Kpoints2	Matches	Match rate (%)
SURF	corners	30	0.795	244	490	138	40.1
BRISK	corners	30	1.217	1406	2213	478	26.4
SURF	corners	90	0.588	244	244	244	100
BRISK	corners	90	1.054	1406	1352	854	61.93
SURF	blobs	30	0.631	173	243	105	50.5
BRISK	blobs	30	0.821	132	447	69	23.8
SURF	blobs	90	0.437	173	180	170	96.3
BRISK	blobs	90	0.931	132	133	123	88.8

Table 2: Results of comparing different parameters depending on rotation changes and descriptors.

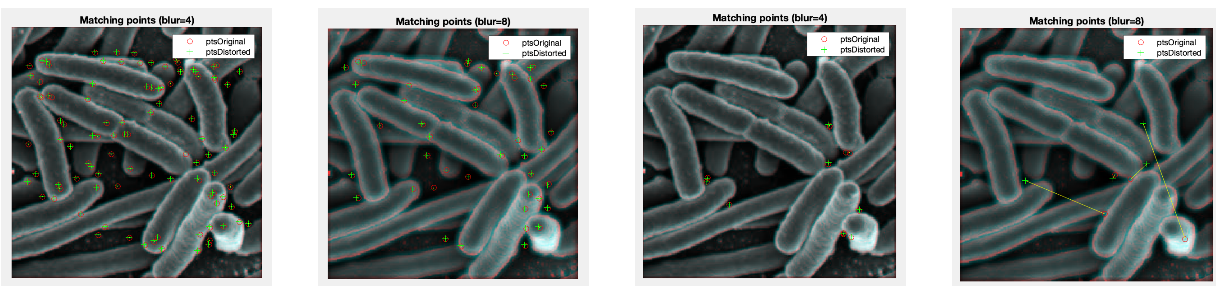
4.3 Blurring

In this case, we filtered the original image using a filter with kernels size of 4x4 and 8x8 to see the blurring effect on the number of matches. From table 3 and figures 7, 8 one can see that the highest number of matches is acquired with a BRISK descriptor and a kernel size of 4x4 for the corner-based image. However, with a kernel size of 8x8, where the corners are more smoothed, the SURF descriptor works better. This effect can be also observed in the recovered image shown in figure 13. For the buildings image, the reconstruction of the blurred image with a kernel size of 8x8 is better when using the SURF method than using the BRISK method. This means that when we increase the blurring effect, corner detectors like BRISK might perform worse as corners in the image are being removed with the smoothing.



(a - windowWidth=4) (a - windowWidth=8) (b - windowWidth=4) (b - windowWidth=4)

Figure 7: The final matched points of varying window widths (blur) images using SURF (a) and BRISK (b) in a corner based image.



(a - kernel=4x4) (a - kernel=8x8) (b - kernel=4x4) (b - kernel=4x4)

Figure 8: The final matched points of varying window widths (blur) images using SURF (a) and BRISK (b) in a blob-based image.

Figure 9: (a) Figure 10: (b) Figure 11: (a) Figure 12: (b)

Figure 13: The corner-based recovered images of varying blurring (kernel size = 8x8) images using SURF (a) and BRISK (b).

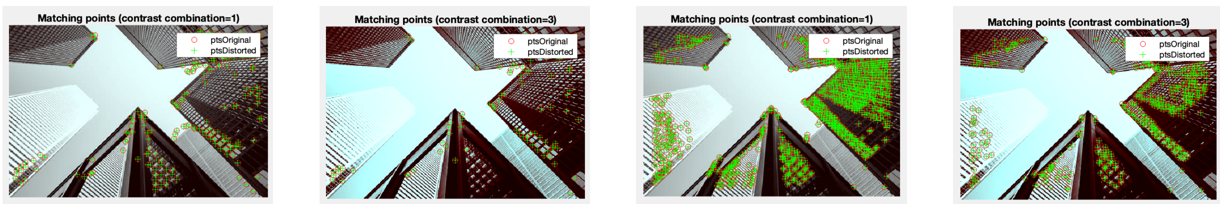
Descriptor	Image	kernel size	Time (sec)	Kpoints1	Kpoints2	Matches	Match rate(%)
SURF	corners	4x4	0.647	244	200	136	61.3
BRISK	corners	4x4	0.950	1406	163	159	20.27
SURF	corners	8x8	0.658	244	104	103	59.2
BRISK	corners	8x8	1.459	1406	17	60	8.43
SURF	blobs	4x4	0.389	173	147	129	80.6
BRISK	blobs	4x4	0.850	132	19	23	30.4
SURF	blobs	8x8	0.522	173	76	83	66.7
BRISK	blobs	8x8	1.048	132	4	14	20.7

Table 3: Results of comparing different parameters depending on blurring changes and descriptors.

4.4 Illumination

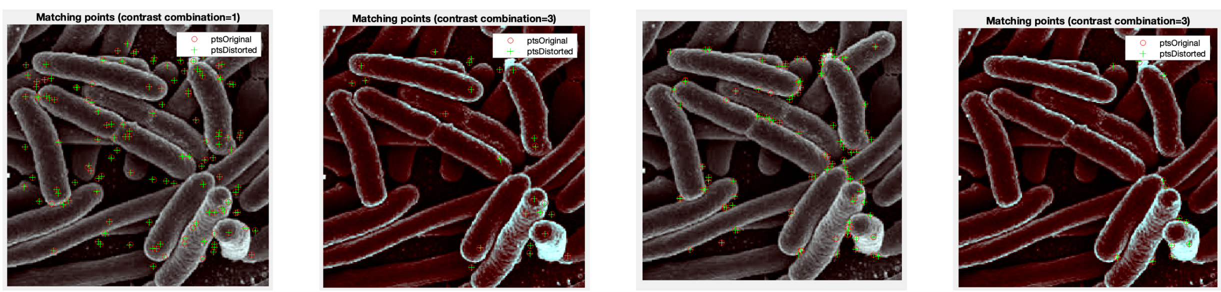
We used different ranges contrast limits to study the descriptors performance. In the table 4 and figures 14, 15 we show the results of the contrast limits ranged between [0.1 0.9] and [0.3 0.7], names combination 1 and combination 3 respectively. We can clearly see that as the contrast limit shrinks SURF and BRISK descriptors have more difficulties in detecting matches. However, even we already know that BRISK descriptor works better in corner-based images and SURF in blob-based images, we can see that BRISK algorithm outperforms in the case where the contrast changes are not that noticeable (combination 1). The main drawback is that its computational time is much higher.

For the bacteria images, figure 15, an strange fact is that with the increment of intensity (combination 3) where the edges are more pronounced we expected to detect more matches with the SURF method but it is not the result obtained.



(a - [low high]=[0.1 0.9]) (a - [low high]=[0.3 0.7]) (b - [low high]=[0.1 0.9]) (b - [low high]=[0.3 0.7])

Figure 14: The final matched points of varying intensity and contrast images using SURF (a) and BRISK (b) in a corner based image.



(a - [low high]=[0.1 0.9]) (a - [low high]=[0.3 0.7]) (b - [low high]=[0.1 0.9]) (b - [low high]=[0.3 0.7])

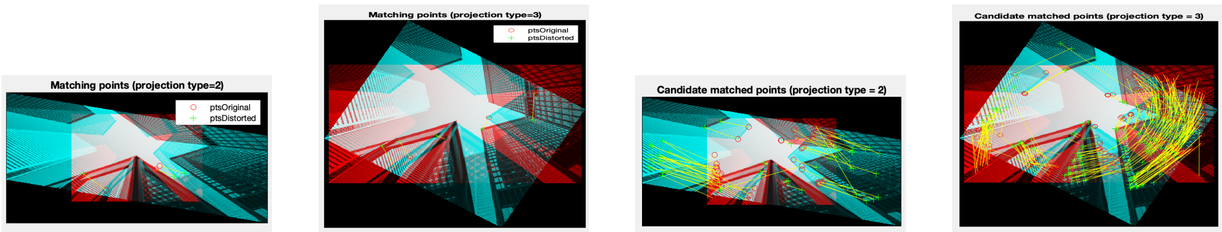
Figure 15: The final matched points of varying intensity and contrast images using SURF (a) and BRISK (b) in a blob-based image.

Descriptor	Image	[low high]	Time (sec)	Kpoints1	Kpoints2	Matches	Match rate(%)
SURF	corners	[0.1 0.9]	0.632	244	343	233	79.4
BRISK	corners	[0.1 0.9]	1.240	1406	1888	1323	80.3
SURF	corners	[0.3 0.7]	0.448	244	582	149	36.1
BRISK	corners	[0.3 0.7]	0.938	1406	2821	618	29.2
SURF	blobs	[0.1 0.9]	0.524	173	230	148	73.5
BRISK	blobs	[0.1 0.9]	1.104	132	222	107	60.5
SURF	blobs	[0.3 0.7]	0.448	173	267	58	26.4
BRISK	blobs	[0.3 0.7]	0.864	132	609	19	5.13

Table 4: Results of comparing different parameters depending on contrast changes and descriptors.

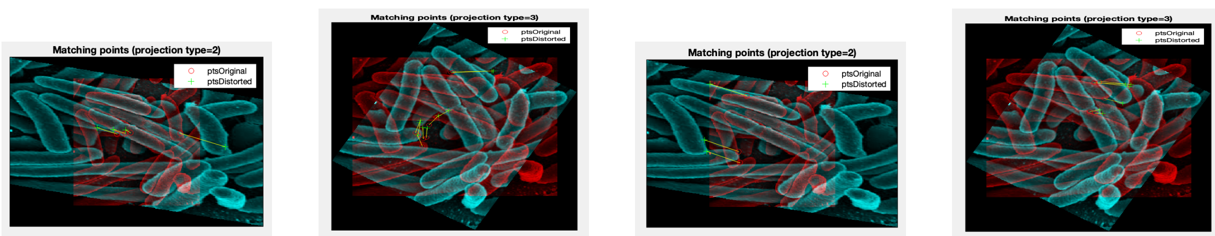
4.5 Projections

In this scenario, the original image was projected in different points of view. In table 5 and figure 16, 17 we show the results of the matching obtained according two types of projections. The type 2 does 3D transformation following the non singular matrix [2 0.33 0; 0 1 0; 0 0 1], whereas for type 3 the transformations performed follow the matrix [0.7 0.5 0; -0.5 1 0; 0 0 1]. Here we can observe that the SURF descriptor works poorly for both types of images whereas BRISK algorithm has a good performance for corner-based images for type 3 projection. Furthermore, we can clearly observe that the reconstruction of the images where each algorithm is stronger is not successfully achieved given the poor number of matches found.



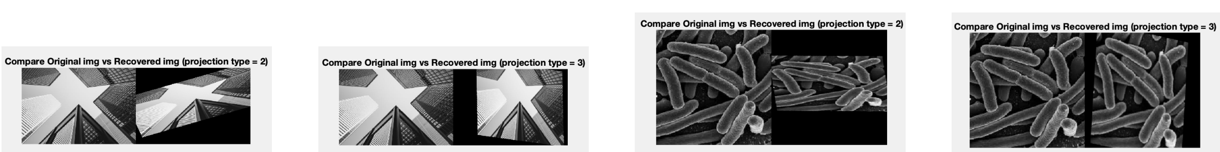
(a - proj. type = 2) (a - proj. type = 3) (b - proj. type = 2) (b - proj. type =3

Figure 16: The final matched points of varying point of view images using SURF (a) and BRISK (b) in a corner based image.



(a - proj. type = 2) (a - proj. type =3) (b - proj. type = 2) (b - proj. type = 3)

Figure 17: The final matched points of varying point of view images using SURF (a) and BRISK (b) in a blob-based image.



(a - proj. type = 2) (a - proj. type =3) (b - proj. type = 2) (b - proj. type = 3)

Figure 18: The recovered images from different point of view using BRISK (a) and SURF (b) in corner-based and blob-based images respectively.

Descriptor	Image	viewpoint	Time (sec)	Kpoints1	Kpoints2	Matches	Match rate(%)
SURF	corners	2	0.736	244	726	66	13.6
BRISK	corners	2	1.154	1406	1681	64	4.15
SURF	corners	3	0.592	244	383	100	31.9
BRISK	corners	3	1.073	1406	1323	302	36.38
SURF	blobs	2	0.681	173	268	56	25.4
BRISK	blobs	2	0.883	132	128	14	10.77
SURF	blobs	3	0.590	173	234	83	40.8
BRISK	blobs	3	1.684	132	218	47	26.9

Table 5: Results of comparing different parameters depending on 3D view points changes and descriptors.

5 Conclusions

The main conclusion that we can extract is that in most of the deformations (scaling, rotation, illumination and projection) it is clear that the most suitable feature extractor and descriptor is the one that matches with the shape typology of the image. That is, BRISK method (corner detector) worked better for the building image (corners) and SURF method (blob detector) for the bacteria image (blobs). It can be seen in tables 1, 2, 4 and 5 that the number of matches is higher for when the image type and the detector type coincide (corner/blob). This shows that is important to take into account the content shapes of the image and choose the appropriate keypoint detector (corner or blobs) in order to ensure robust matches.

With respect to the blur deformation, in table 3, we can observe that BRISK method performs badly as we increase the blurring intensity. However, SURF can handle blurring deformations much better. This means that the corner detector does not perform well when corners are blurred. We note that on table 3 some experiments have more matches than keypoints extracted in the disturbed image. This is due to having one-to-many matches (e.g. one keypoint of image 1 can match several keypoints in image 2, and viceversa).

In general, SURF method has a higher matching rate generally in most of the executions. That means that the keypoints that SURF detects are really discriminative landmarks and easier to match. On the other hand, BRISK generates a lot of keypoints, specially in corner images, but then the matching rate if quite poor (generally under a 50%). This may indicate that SURF algorithm is more *efficient* because the keypoints it detects are more robust to matching while BRISK extracts too many noisy keypoints that then are not being matched.

Regarding the execution time, SURF clearly outperforms BRISK. In all the executions we carried out, SURF execution times are, on average, the half of the BRISK ones. From the experiments, we can notice that BRISK extracts a lot of keypoints (specially on corner images), which can explain why the performance is slower, but the ratio of matched keypoints then is lower, as mentioned before. Hence, SURF features are more suitable for real-time matching applications.

References

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[2] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555, Nov 2011.