Assignment 1: Title of the assignment

Vincent Matthys

vincent.matthys@ens-paris-saclay.fr

QIA.1: (i) Why is it important to have a similarity co-variant feature detector? (ii) How does this affect the descriptors computed at these detections? (iii) How does this affect the matching process?

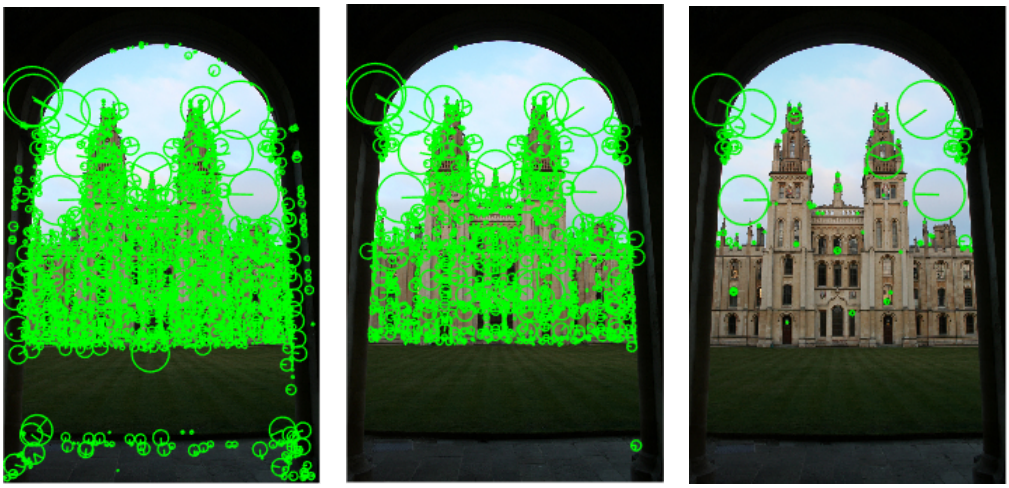
(i) It is important to have a similarity co-variant feature detector in order to extract the same regions of interest regardless of translations, rotations or scales, *i.e.* regardless of viewpoints.
(ii) The fact that the feature detector is co-variant causes the descriptors computed at these detections to be invariant after renormalisation step.
(iii) Having feature detectors with similarity co-variance and feature descriptors allows the matching process to be independant of the viewpoints of the images.

QIA.2: Show the detected features in the two images for three different values of the peakThreshold option

The detected features for $peakThreshold \in \{0.0001, 0.001, 0.01\}$ are shown in Figure 1 for the all_souls_000002.jpg image, and in Figure 2 for the all_souls_000015.jpg image. As expected, the number of keypoints decreses with the threshold. Moreover the keypoints are fewer in the darker regions, wich leads to asymetric density in the Figure 2, especially because of the building shadow.

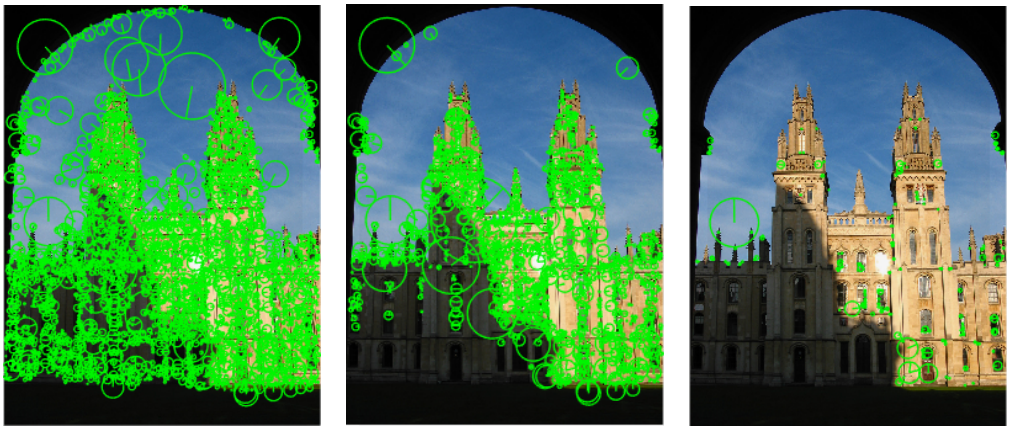
QIA.3: Note the change in spatial density of detections across images, for a given value of peakThreshold. (i) Is the density uniform? If not, why? (ii) Which implications for image matching can this have? (iii) How can it be avoided?

(i) In Figures 1a and 2a. The density is clearly not uniform between the two images. In both, the top-center of the image is very dense, but, in the first one, the grass admits no detection, which leads to a gap in the density, with some detections in the bottom limit of the grass.
(ii) This can lead to error matching the correspondances with local features
(iii) To avoid it we can proceed to a spatial verification relying on global geometric relations.



(a) $peakThreshold = 0.0001$ (b) $peakThreshold = 0.001$ (c) $peakThreshold = 0.01$

Figure 1: Feature detectors with three different values of $peakThreshold$ for all_souls_000002.jpg

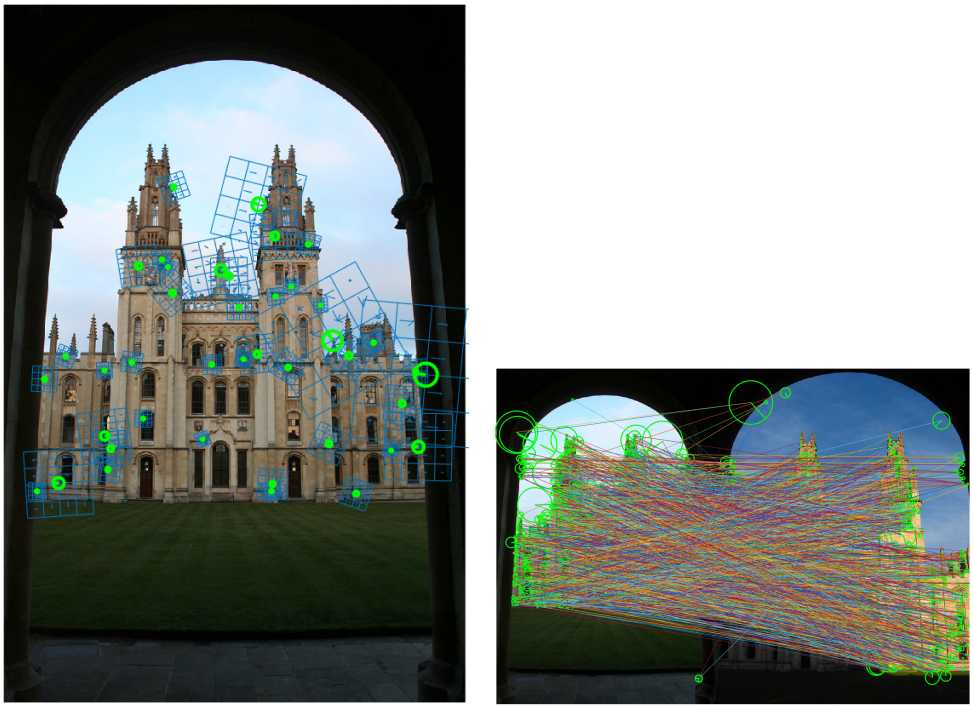


(a) $peakThreshold = 0.0001$ (b) $peakThreshold = 0.001$ (c) $peakThreshold = 0.01$

Figure 2: Feature detectors with three different values of $peakThreshold$ for all_souls_000015.jpg

QIB.1: Note the descriptors are computed over a much larger region (shown in blue) than the detection (shown in green). Why is this a good strategy?

The larger regions are shown in Figure 3a. It is a good strategy to be insensitive to small geometric deformations, and small variations of illumination, by taking the average over a larger region



(a) 4x4 grid arround the keypoint (b) first NN match

Figure 3: Computation of SIFT descriptors and matching

QIB.2: Examine carefully the mismatches and try to understand the different causes of mismatch. (i) In your report, present at least 3 of them and explain why the mistakes are being made. For example, is the change in lighting a problem? (ii) What additional constraints can be applied to remove the mismatches?

(i) Three types of mismatch are presentend in Figure 4, and are extracted from Figure 3b

- repetitive pattern (in blue)
- constant local change in lighting leading to similarity between two different keypoints (in red)
- ambiguity between a keypoint his first two nearest neighbors (in yellow)

A global change in lighting is irrelevant for matching, because of the gradient nullifying the constant component

(ii) Geometric verification with global constraints can discard those mismatches. Another method, presented below, is to look at the second nearest neigbour to remove ambiguous matches.

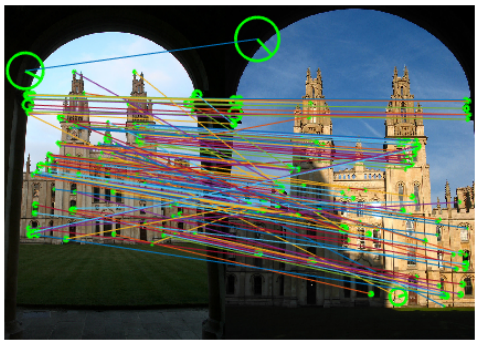
Figure 4: Three mismatches with SIFT features descriptors, subgroup of matches extracted from 3b

QIC.1: Illustrate and comment on improvements that you can achieve with this step. Include in your report figures showing the varying number of tentative matches when you change the nnThreshold parameter.

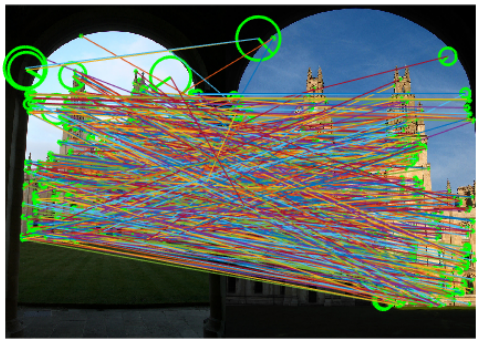
As expected, the percentage of matches kept increases with $nnThreshold$. The results obtained are shown in Figures 5 6 7. Until $nnThreshold = 0.7$, there is no substantial mismatches, but the number of matches kept is below 10 %. If more matches are kept, a non-negligeable part of new matches is missmatches, because the ambiguity between the first and the second nearest neighbour is no longer removed. In Figure 7b we can see the exponential growth of the number of matches with $nnThreshold$. A good compromise would be to take the threshold before this exponential growth.

(a) $nnThreshold = 0.3$ (b) $nnThreshold = 0.5$

Figure 5: Matching using Lowe's second nearest neighbour - part1

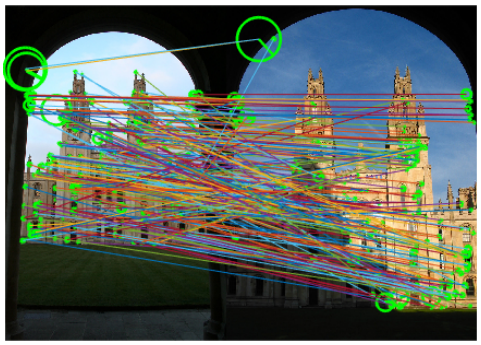


(a) $nnThreshold = 0.7$

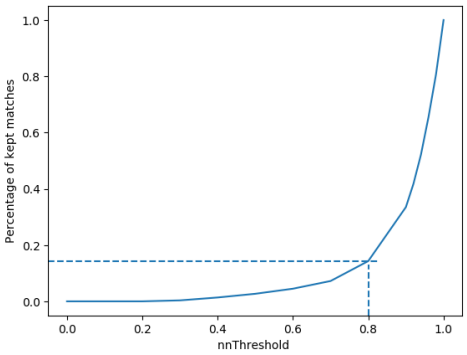


(b) $nnThreshold = 0.9$

Figure 6: Matching using Lowe's second nearest neighbour - part2



(a) $nnThreshold = 0.8$



(b) Percentage of matches in function of the $nnThreshold$

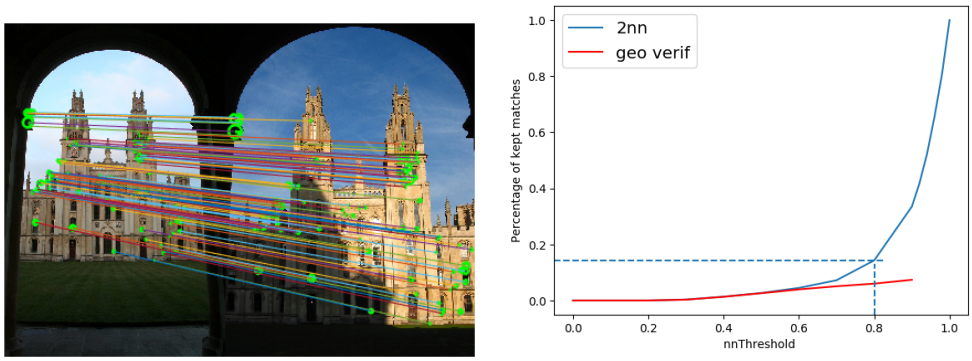
Figure 7: Matching using Lowe's second nearest neighbour - part3

QID.1: Work out how to compute this transformation from a single correspondence. Hint: recall from Stage I.A that a SIFT feature frame is an oriented circle and map one onto the other.

We need four equations to compute the transformation. With a signle correspondance, we can have 2 equations given by the coordinates of the keypoint in image 1 and the second one in image 2. We have the direction of the gradient, given by the segment, in the both images, giving a third equation. Finaly, we have the size of the blobs in both equations, giving the final equation to compute the 4-parameters transformation.

QID.2: Illustrate improvements that you can achieve with this step.

In Figure 8a, it's clear there is no mismatch anymore: every match corresponds to a unique geometric tranformation. The number of matches dropped from 293 to 123. It means only 42% of the matches given by the Lowe's method were inliers. In Figure 8b we can observe



(a) $nnThreshold = 0.8$

(b) Percentage of matches in function of the $nnThreshold$

Figure 8: Improving SIFT matching with Lowe's 2nd nn and geometric verification

the almost linear growth of matches with $nnThreshold$ with the geometric verification, in comparaison with the exponential growth of the Lowe's method alone.

QII.1: Include in your report images and a graph showing the number of verified matches with changing viewpoint. At first there are more similarity detector matches than affine. Why?

At first, introducing affine detectors is introducing a second spatial dimension for each blob, which leads to one more constraint to verify in the geometric transformation. This results discarding some outliers at the very beginning.

QIIIA.1: In the above procedure the time required to convert the descriptors into visual words was not accounted for. Why the speed of this step is less important in practice?

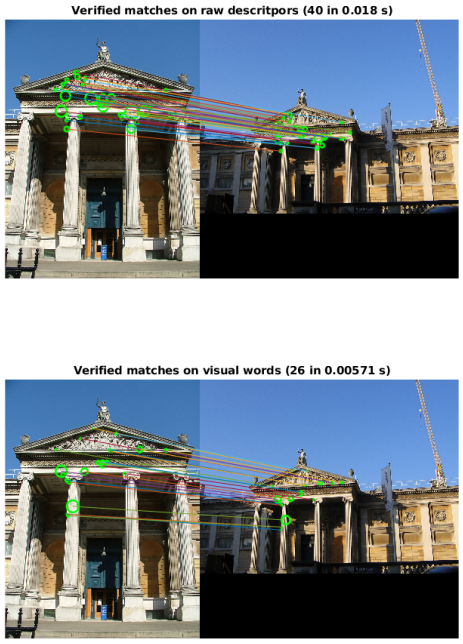
This step is less important in practice because it's an individual step, which can be done seperatly from the query process, and the result of the translation into visual words can be easily stored with the image, as it represents nothing in comparaison with the size of the image.

QIIIA.2: What is the speedup in seconds in searching a large, fixed database of 10, 100, 1000 images? Measure and report the speedup in seconds for 10, 100 and 1000 images.

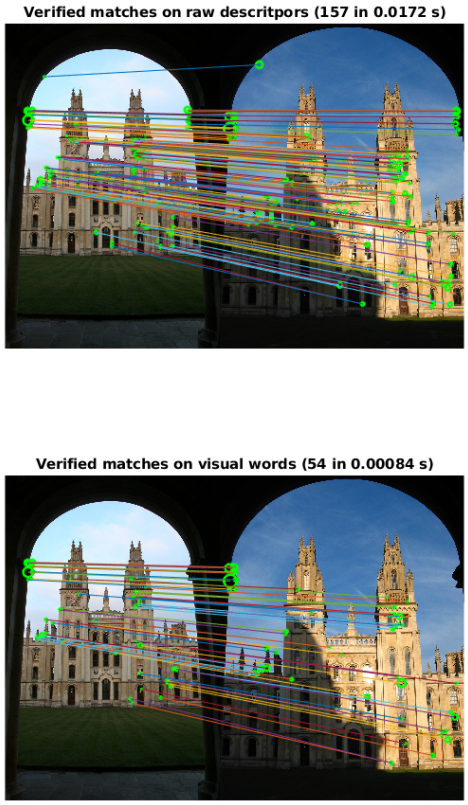
In Figure 9, the speedup in seconds per image in the databse is $\approx 1.5\ 10^{-2}\ s$, so, if the conjecture holds, for a database of 10, 100 and 1000 images respectively, the speedup would be 0.15 s, 1.5 s, 15 s.

QIIIB.1: Why does the top image have a score of 1?

The top image have such a score because it comes from the image database. So the histograms of visual words between the query and the same image in the databae are the same,



(a) 3.1x faster, with $nnThreshold = 0.8$ and geometric verification



(b) 20x faster, with $nnThreshold = 0.8$ and geometric verification

Figure 9: Accelerating descriptor matching with visual words between second neighbour test and geometric verification

hence the score of 1.

QIIIB.2: Show the first 25 matching results, indicating the correct and incorrect matches. How many erroneously matched images do you count in the top results?

The results of the search are presented in Figure 10. The search took 7.51 10^{-6} $s/image$ and gives 9 correct answers over the 25 top results.

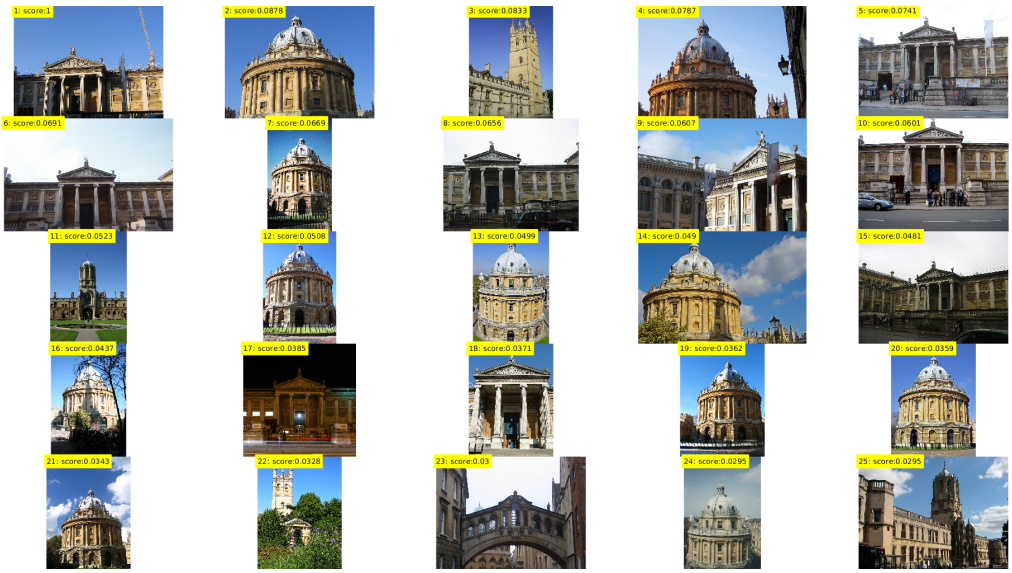


Figure 10: Searching ashmolean_000028 image in 660 images of the Oxford image database lite_imdb_100k

QIIIC.1: Why is the top score much larger than 1 now?

The score is now the number of inliers words between the query and the database. It's larger than 1 for the top result which shares every inliers words with himself.

QIIIC.2: Illustrate improvements of retrieval results after geometric verification.

The result of the rescoring of the top 25 previous results is shown in Figure 12. On the top 25 results, the 9 correct images are now ranked from 1 to 9,, with a score at least 4 times bigger than the 10th result.

QIII.D: Full system

The query has yet not been shown to the database. The result of the search in the 660 images of the Oxford building database are shown in Figure 13. Over the top 25 results, only correct answers appear. The search took 2.2 10^{-3} $s/image$ globally, but the index time was only 6.1 10^{-3} $s/image$ as already computed in partIII.B.

Figure 11: Searching ashmolean_000028 image in 660 images of the Oxford image database lite_imdb_100k. Rescoring with number of inliers words

Figure 12: Searching ashmolean_000028 image in 660 images of the Oxford image database lite_imdb_100k. Rescoring with number of inliers words

QIV.1: How many features are there in the painting database?

QIV.2: How much memory does the image database take?

QIV.3: What are the stages of the search? And how long does each of the stages take for one of the query images?

(a) Quert image

(b) Top 25 results, rescoring with number of common visual words

Figure 13: Searching mistery-building1 image in 660 images of the Oxford image database lite_imdb_100k with geometric rescoring