Image denoising: cours 3

Experimental report

MVA 2021-2022
ENS Paris-Saclay
Elías Masquil
eliasmasquil@gmail.com - elias.masquil@ens-paris-saclay.fr

Exploring Patch Similarity in an Image

With the tools and methods explained in this article, I'm going to do some experiments to test some of the most common hypotheses about image patches used in image denoising, and image processing in general.
For three different patches, each of them from a different type (natural texture, edge, and flat patch), I'm going to observe the following hypothesis:

- Self-similarity
- Sparsity
- Gaussianity

Natural texture

Original image

Center of the reference patch

Figure 1: Natural texture

Self-similarity

By observing the different sets of patches closest to the closest patches to the reference patch, we see that there's a lot of self-similarity on this image. The algorithm was able to find 255 patches which form a self-similar group.

Closest patches to the reference patch Closest patches to the patch 255 from the original set of closest patches

Figure 2: Closest patches are a self-coherent set

By looking at the figure 2, we can see that the patches similar to the reference one, are also similar to the last patch closest to the reference. Simply looking at the set of all the closest patches, we can also see that all of them are really similar.

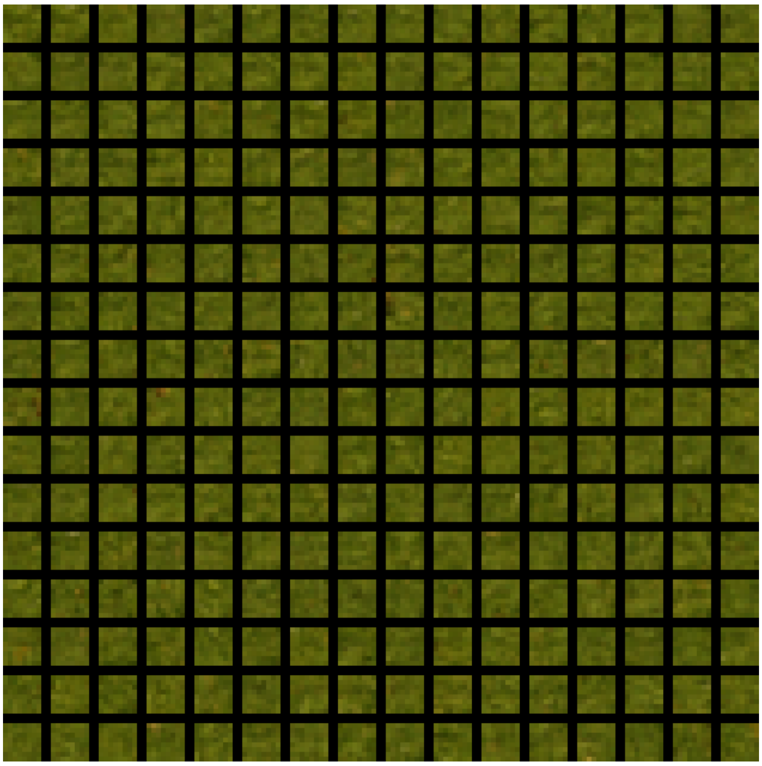


Figure 3: Closest patches to the reference

Sparsity

The sparsity hypothesis doesn't hold in this case. If we observe the histogram of the eigenvalues associated with the principal components, we don't see a rapid decay in the coefficients. Conversely, for representing such a complex texture, we need to keep many coefficients.

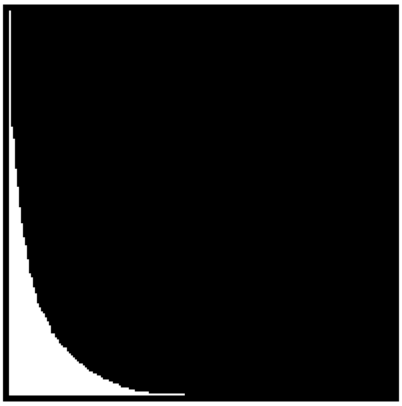
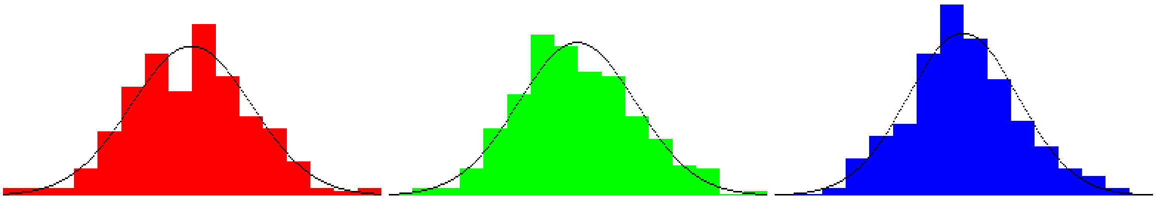


Figure 4: Absolute value of the PCA coefficients of the cluster of Canny patches of natural texture

Gaussianity

Let's observe the projection of the three principal components of the cluster and apply the Anderson-Darling normality test to each component.



p-values: 0.5187, 0.1239, 0.3773

Figure 5: Histograms of PCA projections of the cluster of natural texture patches

All three p-values>0.05, so we can't reject the hypothesis that data follow a Gaussian distribution. This can also be seen by just looking at the projections, which look pretty Gaussian.

For this natural texture, we can safely say that the patches follow a Gaussian distribution

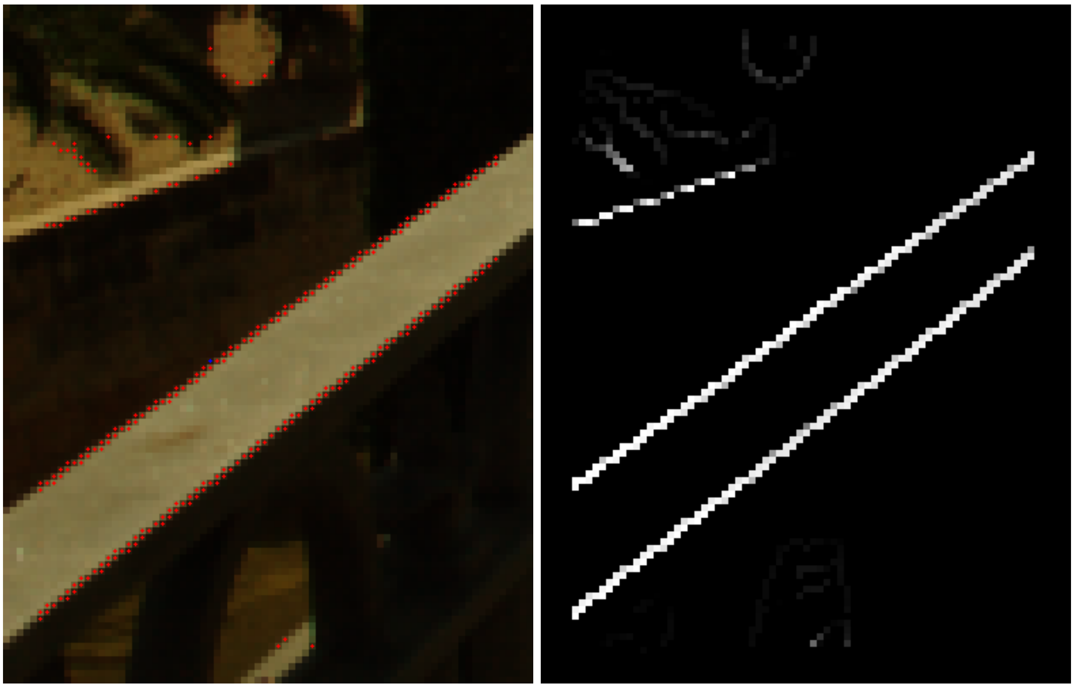
Edge

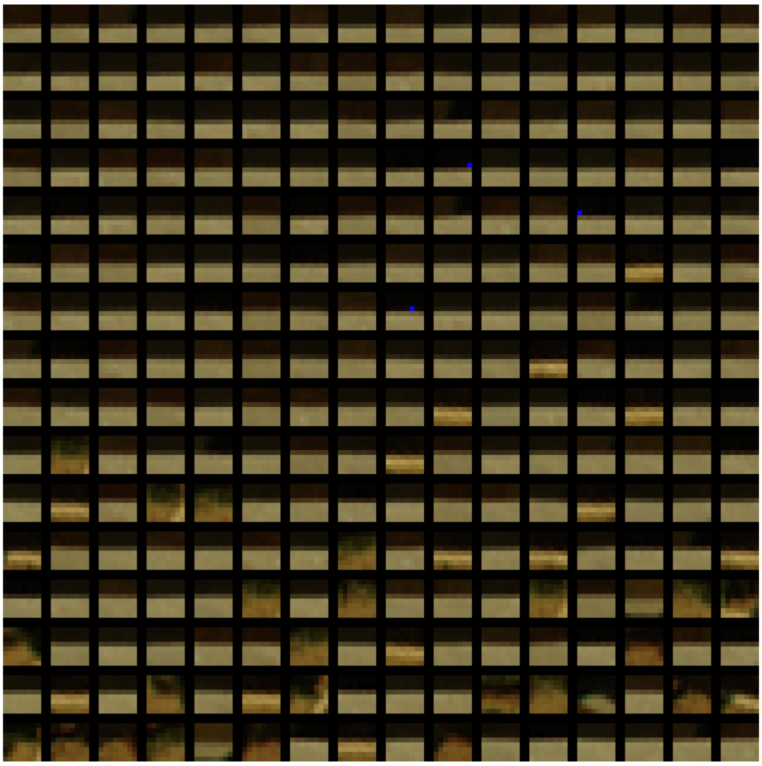
Center of the reference patch in red

Figure 6: Edge

Self-similarity

For this image, if we run the demo with the default parameters, we can see that the default window size of 100 is not big enough for finding a self-coherent set of patches. Although the first matches are similar to the reference patch, the least similar ones don't seem to belong to the self-similar set. This can be seen in the following figure. Note the circular pattern in the histogram and the centers of the closest patches.

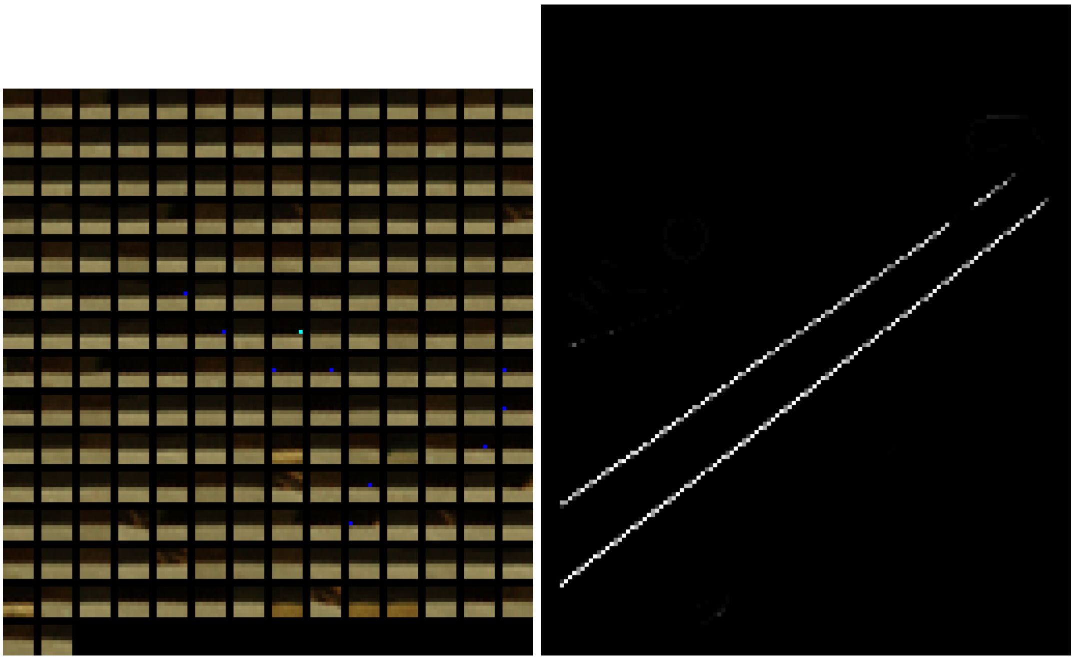




(Top left) Closest patches to the reference, (top right) histogram of positions of closest patches, and (bottom) set of all similar patches

Figure 7: Some of the closest patches are not the same type of edges

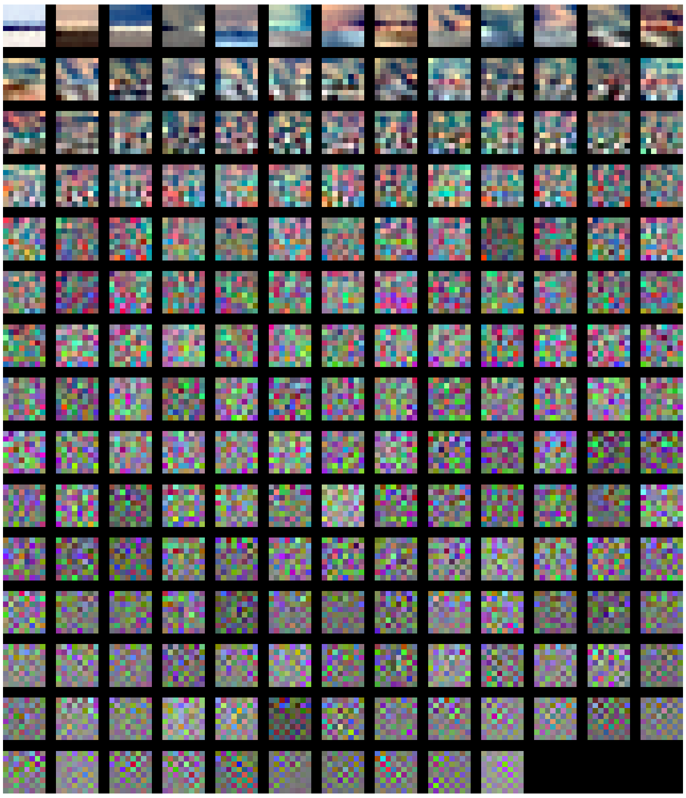
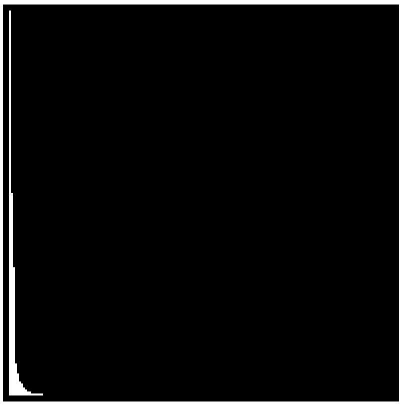
Reducing the number of patches to 198 and expanding the window size to 200, allow us to find a set with more self-coherence. Now only patches on the edge of the railing remain. By playing a little with the parameters, we can also validate the self-similarity hypothesis for the edge pattern on this image.



(Left) Closest patches to the reference, (right) histogram of positions of closest patches

Figure 8: Similar patches for the edge

Sparsity

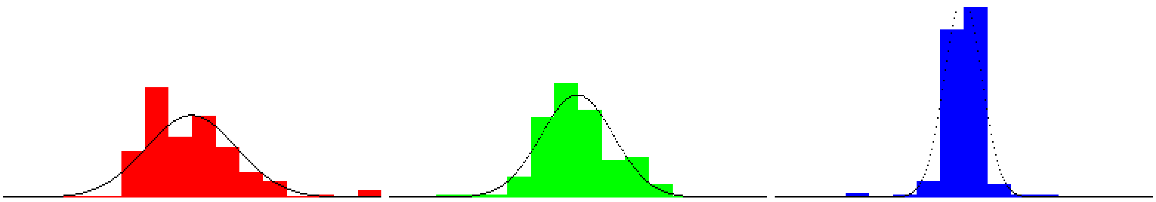
 

(Left) Basis of the image patches cluster obtained with PCA
(right) Absolute value of the PCA coefficients of the cluster of Canny patches of an edge
Figure 9: PCA of the cluster of patches of the edge

In this case, the representation of this patch is much simpler, so the sparsity hypothesis holds. Only about 7 coefficients are different from zero, and are enough to describe this cluster of patches. Note how the vectors from the basis, associated to the zero coefficients, don't actually have any structure and only exist for completing the basis.

Gaussianity

It is clear that the edge patches are not Gaussian, both by looking at the projections of the principal components and applying the Anderson-Darling normality test.



p-values: 0.0000, 0.0053, 0.0000
Figure 10: Histograms of PCA projections of the cluster of edge patches

Flat

To find flat patches I needed to change the extraction method to allow all possible points, instead of only Canny points.

Figure 11: Center of the flat patch

Self-similarity

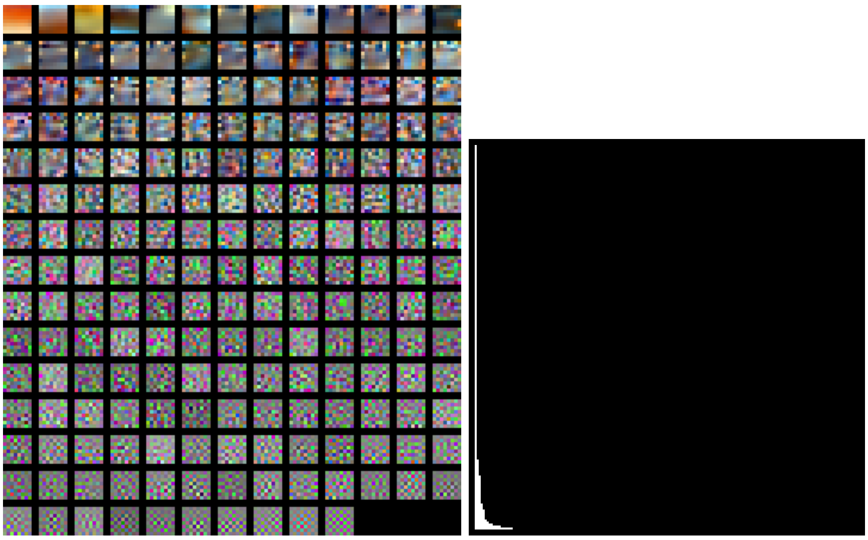
As expected, there are hundreds of similar patches in this image. This can be seen in the results of running the demo with a window of size 200 and looking for 350 similar patches.

(Left) center of the 350 similar patches, (right) histogram of positions of closes patches

Figure 12: Similar flat patches

Sparsity

Again, due to the flat structure of these patches, its representation in the orthogonal basis obtained with PCA is sparse.



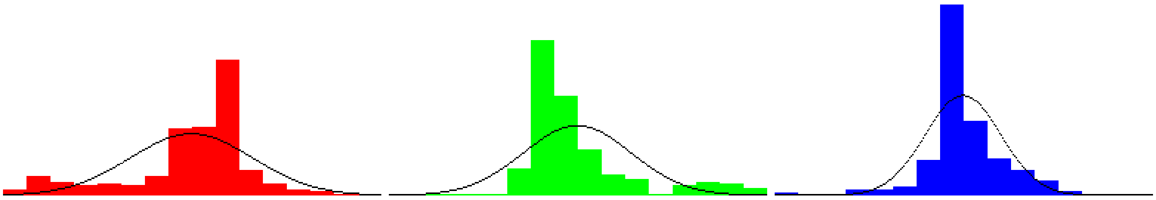
(Left) Basis of the image patches cluster obtained with PCA

(right) Absolute value of the PCA coefficients of the cluster of flat patches

Figure 13: PCA of the cluster of flat patches

Gaussianity

Flat patches are not Gaussian; this can be concluded by looking at the projections of the principal components and applying the Anderson-Darling normality test.



p-values: 0.0000, 0.0000, 0.0000

Figure 14: Histograms of PCA projections of the cluster of flat patches

Non-Local Bayes

Theoretical comparison with BM3D

Non-Local Bayes can be understood as an extension of the Non-Local Means method. Instead of taking a weighted average of the self-similar patches, we add a Gaussian prior to the self-similar set of patches, estimate its mean and covariance, and find the MAP estimator of the noise-free patch.

At the same time, it's very interesting to compare this method to BM3D. Although at first sight both methods can seem pretty different, they are indeed quite similar. Both of the methods start by grouping similar patches, then apply a collaborative filtering, and finally aggregate the estimation of the patches. Both algorithms use the oracle trick in which a first estimate is then used for getting a better, and final, estimate of the noiseless image. In Non-Local Bayes the collaborative filtering consists in estimating the mean and covariance of the cluster containing the noise-free patch, and the aggregation is just an average without weights. If one wants to compare the filtering on the frequency domain, it's possible to frame NL-Bayes as a Wiener filter using as an orthogonal basis the eigenvectors of the covariance matrix instead of a fixed basis like DCT, etc.

Experimental comparison with BM3D

Figure 15: Noisy image (std=50)

Figure 16: BM3D (PSNR=26.453)

Figure 17: Non-Local Bayes (PSNR=26.8357)

Both methods achieve good results despite the high level of noise. In terms of the PSNR, Non-Local Bayes has a slightly better performance. Comparing the images visually, I think that BM3D outperforms Non-Local Bayes over most of the flat areas of the image (such as the road, and especially over the cars and the bus). On the other hand, over the texture of the trees, Non-Local Bayes returned a smoother and more realistic estimate. Finally, the artifacts produced in the sky are quite different in both methods. With BM3D some minor artifacts related to the DCT seem to appear, with Non-Local Bayes some green artifacts seem to happen due to patches with tree leaves.