Image denoising
Experimental assignment Course 3

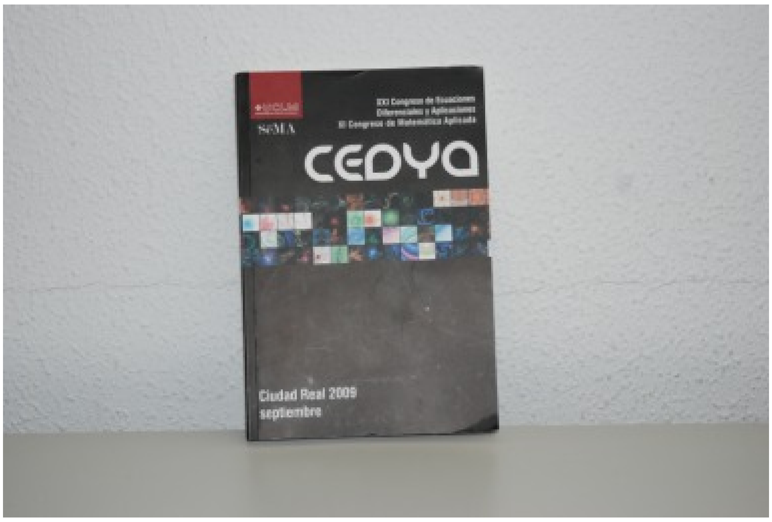
Denys Sikorskyi

1. Exploring Patch Similarity in an Image

1.1. Introduction

In the paper, we could see the methods to test self-similarity based on the fact that repeated patterns, edges, or texture are good characteristics of the image. We could also test the gaussianity hypothesis for image analysis and restoration. In this work, we will test these methods by changing some parameters and exploring different images.

1.2. Change of parameters



Original image

At first, we will change the number of patches and see how it affects the results of the algorithm.

Patches closest to closest 1 (16 patches) *Patches closest to closest 15* *Patches closest to closest (histogram)*

Patches closest to closest 1 (64 patches)

Patches closest to closest 63

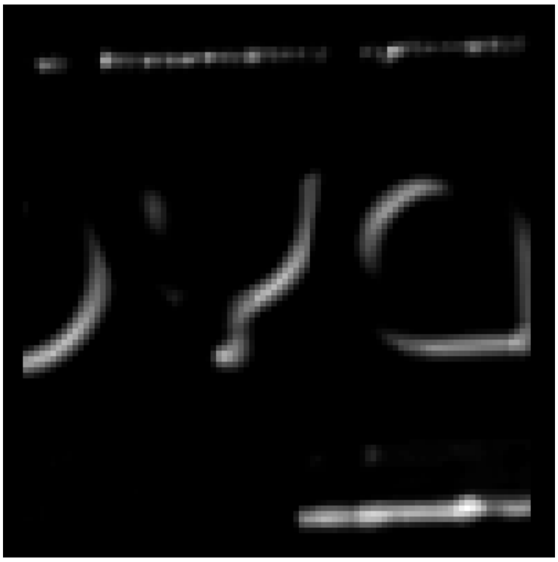

Patches closest to closest (histogram)

Patches closest to closest 1 (256 patches)

Patches closest to closest 256


Patches closest to closest (histogram)

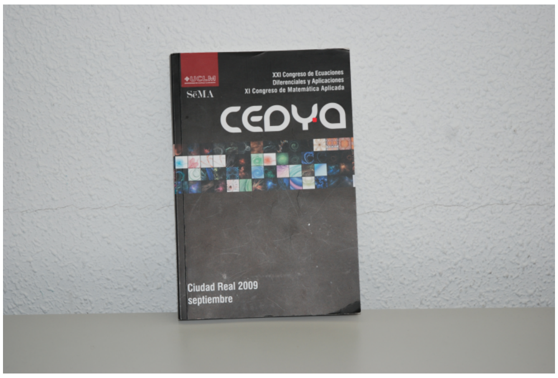
Starting from the 16 patches, we can see that patches don't have the same closest patches. The histogram doesn't seem to capture all boundaries. With the increase in the number of patches, the histogram starts to have the visible boundaries of the letters on the image. And finally, when the amount of patches is equal to 256 we get almost all edges of the letters on the image, and we can see this even in the histogram image.



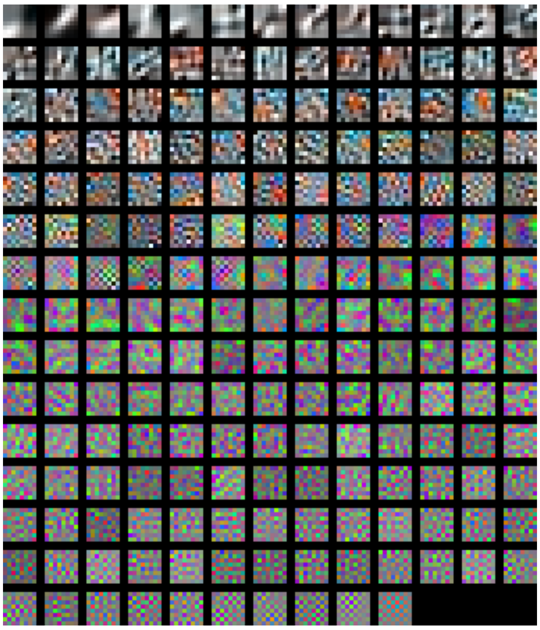

Patches closest to closest (histogram) without(left) and with(right) normalization of orientation

The histogram image when we don't apply normalization of orientation shows that in this case, the algorithms focus on other structures and only on some parts of the letter edges. As we can see the algorithm catches other letters at the top of the image and the lower part of the bottom line of images. On the histogram image with normalization of orientation, we can clearly see the boundaries of big letters. Thus, when we use this algorithm in future experiments, we will use a big number of patches and apply normalization.

1.3. Test of self-similarity and gaussianity hypothesis



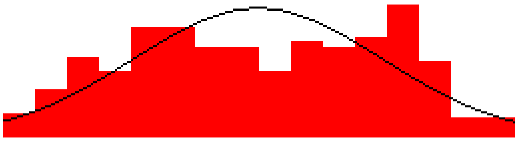
Selection of patch center



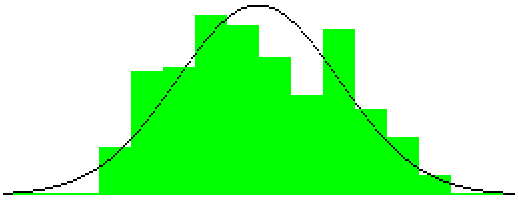
The eigenvectors of the PCA

The spectrum of the PCA

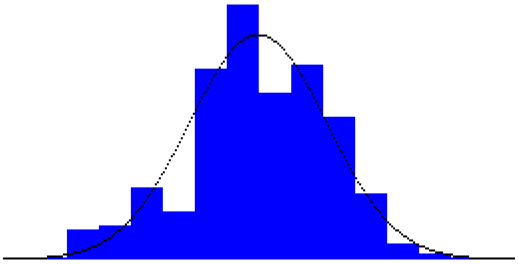
From the spectrum of the PCA, we can make a conclusion that most of the eigenvectors can be neglected and only a small percentage of them contain the information. Therefore, we have low dimensional characteristics of edge-based patches and we can say clusters of canny patches on edges are low dimensional and therefore sparse.



p-value = 0.0000



p-value = 0.0005



p-value = 0.0070

Histograms of PCA projections and their p-value of Anderson-Darling normality test

The histograms of PCA projects clearly show that distributions are not normal which is confirmed by p-values of the Anderson-Darling normality test (smaller than 0.05). We can make a conclusion that clusters for edge patches are not Gaussian and the set of similar patches is sparse.

1.4. Texture Example

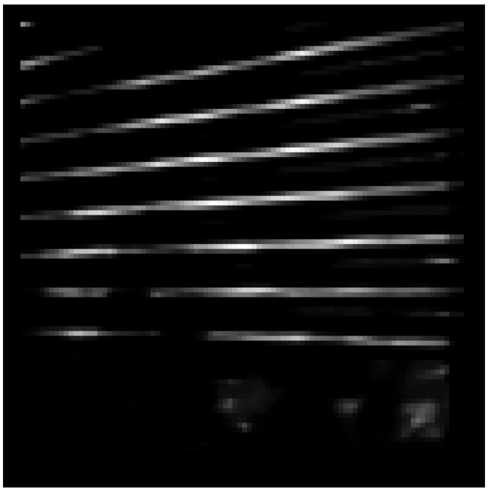
In the next example, we will use texture patches and make some experiments with them.

Original image with initial patch

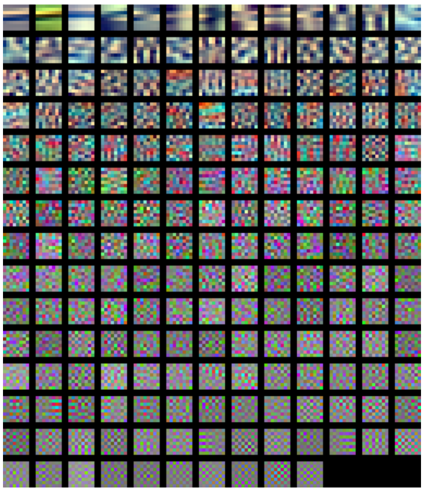
We will work with a patch on the wall of the building.

Patches closest to closest 1 (512 patches)

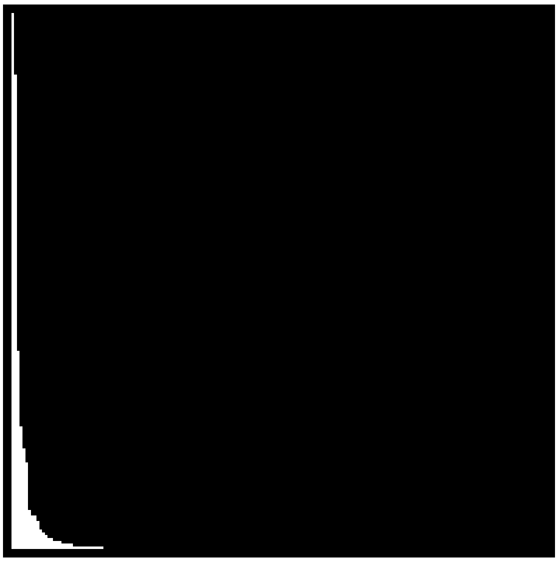
Patches closest to closest 511



Patches closest to closest (histogram)



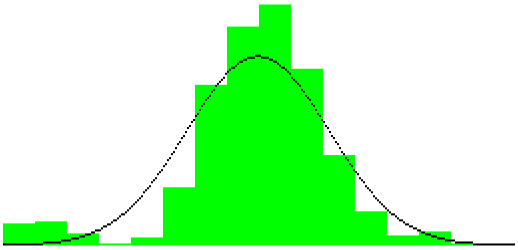
The eigenvectors of the PCA



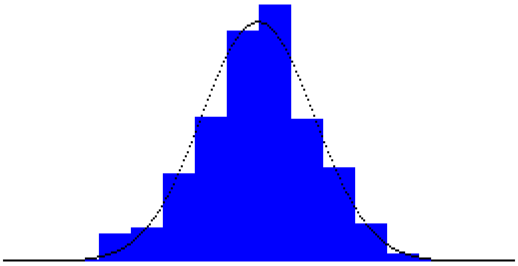
The spectrum of the PCA

From the first and the last patch cluster pictures we can say that images have lots of similar patches, so we have a stable cluster. The histogram clearly shows the lines, so the algorithm works well. The PCA spectrum shows that there are some vectors that provide a decent amount of information. Therefore, we have a high dimensional characteristic of texture-based patches and the structure of texture patches is not sparse.

p-value = 0.0000



p-value = 0.0000



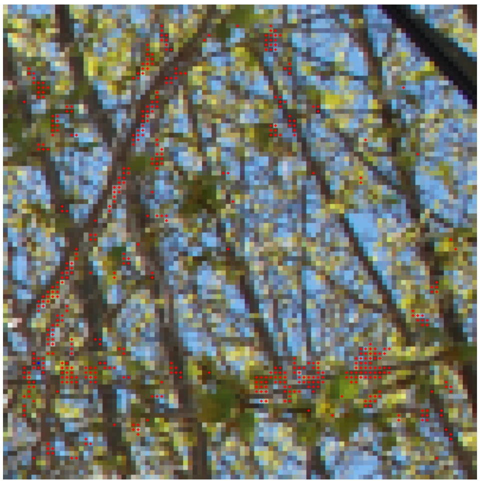
p-value = 0.0289

All histograms of the PCA projections clearly show that they are not Gaussian and this fact is confirmed not only visually but by the p-value of the Anderson-Darling test which is all less than 0.05. Therefore, texture patches are high-dimensional and not Gaussian.

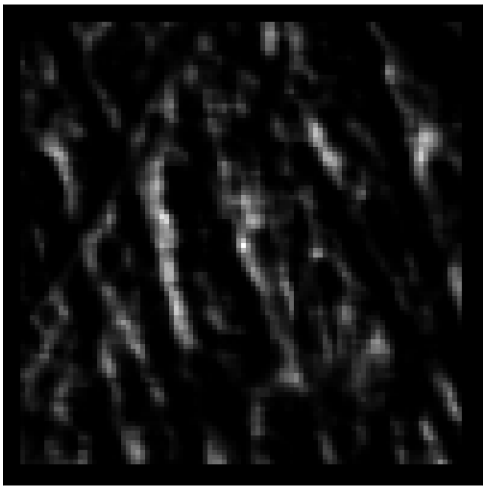
1.5. Natural Texture Example

Original image with the center

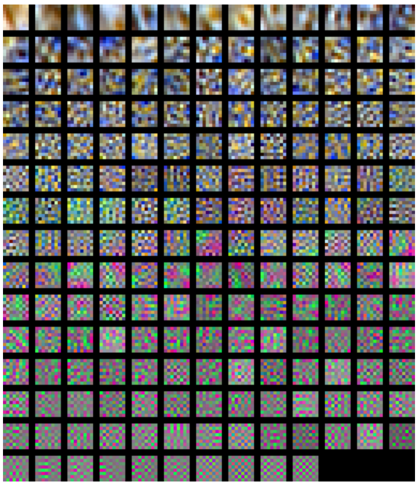
Patches closest to closest 1 (512 patches)



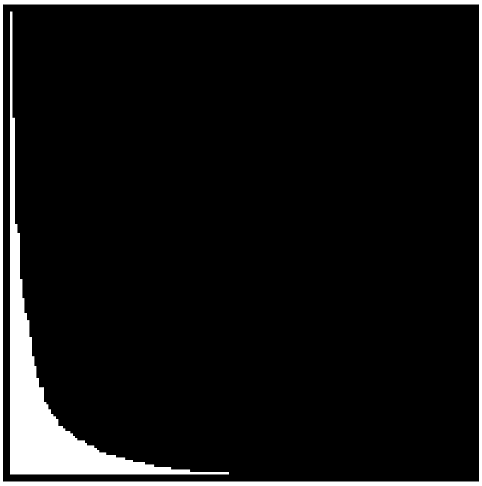
Patches closest to closest 511 (512 patches)



Patches closest to closest(histogram)

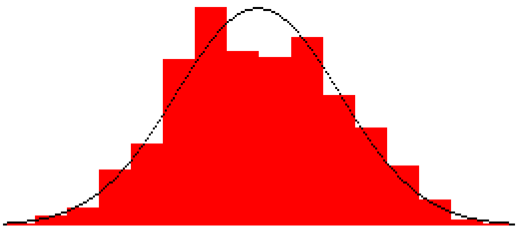


The eigenvectors of the PCA

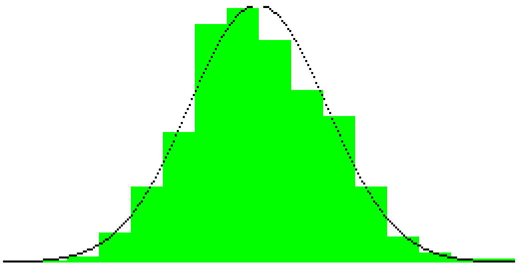


The spectrum of the PCA

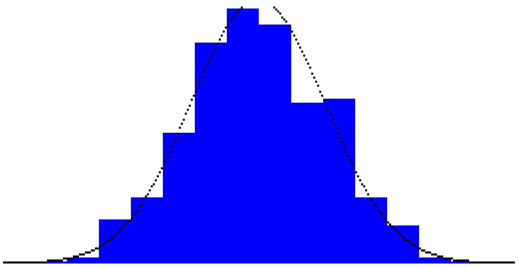
Again from the spectrum of PCA, we can make a conclusion that there is a big number of significant eigenvectors, so we have high dimensional characteristics. Therefore, the natural texture is not sparse.



p-value = 0.0087



p-value = 0.1158



p-value = 0.3574

Two out of three histograms of the PCA projections clearly show that they are Gaussian and this fact is confirmed not only visually but by the p-value of the Anderson-Darling test which is greater than 0.05. Therefore, nature texture patches are high-dimensional and close to Gaussian.
We can make a final conclusion about patch similarity. The clusters on the edges are sparse and not Gaussian. The clusters of texture patches are not sparse and not Gaussian, but the clusters of natural texture patches are not sparse, but Gaussian.

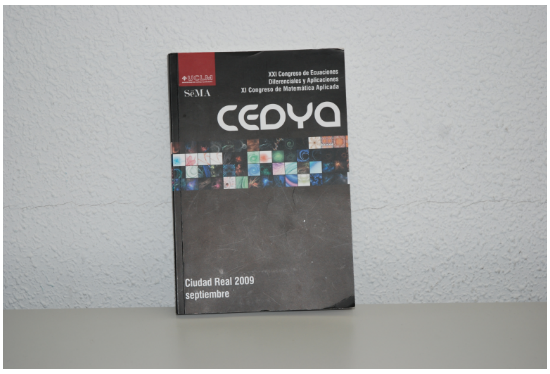
2. NL-Bayes Image Denoising Algorithm

2.1. Introduction of the algorithm

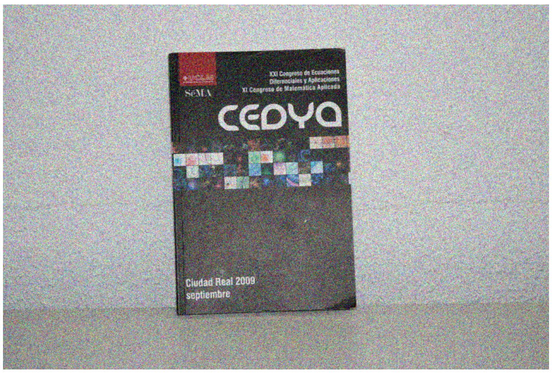
The Non-Local Bayes is the algorithm that was created by using two from previous assignments, which can be guessed from its name. In short, it is made from NL-means and BM3D. The NL-Bayes uses average averaging of similar patches and then evaluates for each group of similar patches a Gaussian vector model. Next, we use something like BM3D where the second one uses the denoised image of the first iteration as an "oracle". The first step of NL-Bayes forms 3D blocks of similar patches as in BM3D. Next, we have a criterion for homogeneous areas to evaluate the estimate as the average of all similar patches of each block. When we don't have homogeneous areas we have a Bayesian framework for the shrinkage of 3D groups of sismilar patches. We compute the mean and variance of each block as an empirical mean of similar patches. Then we get the restored patch as the MAP estimator, then aggregation, but unlike BM3D, this aggregation is not weighted. Next, we use all denoised patches from the first step and repeat the estimate of mean and variance, then we use oracle and get the denoised patch. In the end, we use an unweighted aggregation to get the final denoised patch.

2.2. Parameters testing

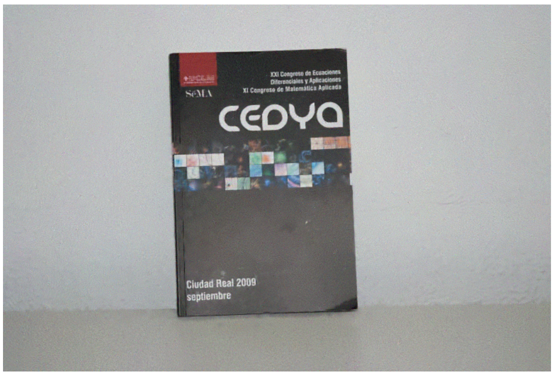
In this part, we will make some experiments in order to estimate its effectiveness. Also, the NL-Bayes have a comparable structure with BM3D, so it is logical to compare them with each other. But first, we will check the effectiveness of the algorithm and the importance of each parameter.



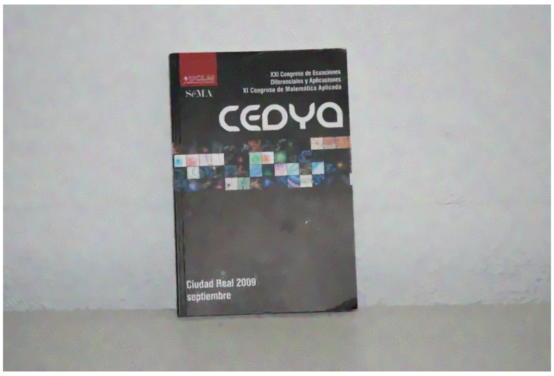
Original image



Noisy image with sigma = 30



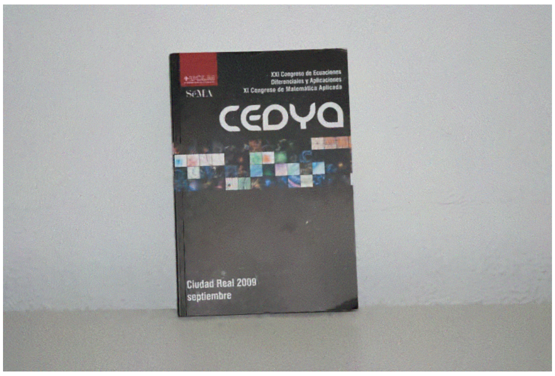
Denoised image at 1st step PSNR = 30.5548



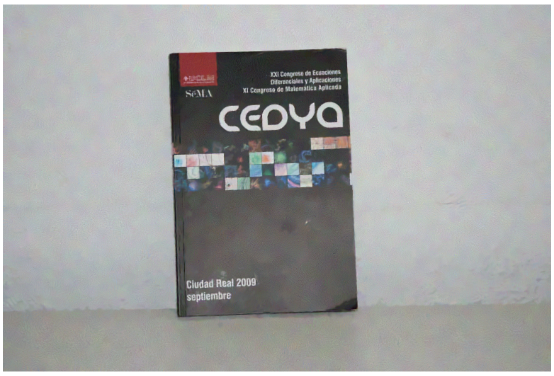
Denoised image at 2nd step PSNR = 32.7503

The PSNR of the second step is 32.7503 which is greater than the PSNR of the first step. Visually, the first image looks better, especially in homogeneous black areas which can be concluded by noticing some artifacts in the black areas of the book. There are different ways to apply the homogeneous area criterion, we will combine them into all possible combinations and compare them.

At first, we don't apply any criterion at any step.

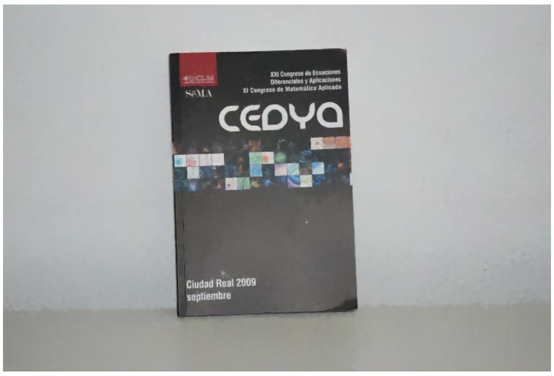


Denoised image at 1st step PSNR = 30.5548

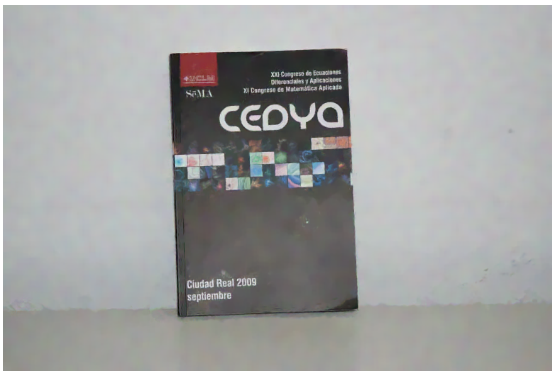


Denoised image at 2nd step PSNR = 32.7503

Next we apply it only at the first step.

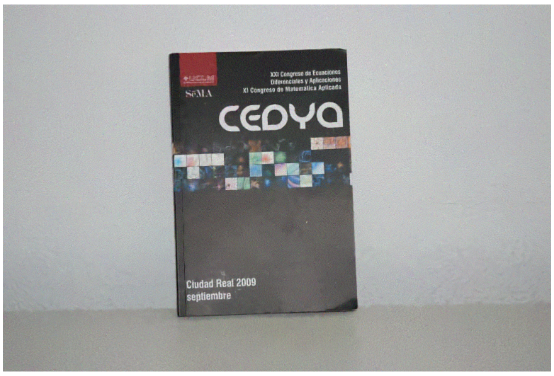


Denoised image at 1st step PSNR = 34.4946

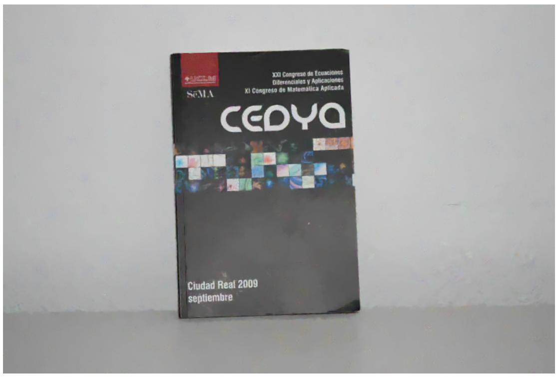


Denoised image at 2nd step PSNR = 34.6787

Only at the second step.

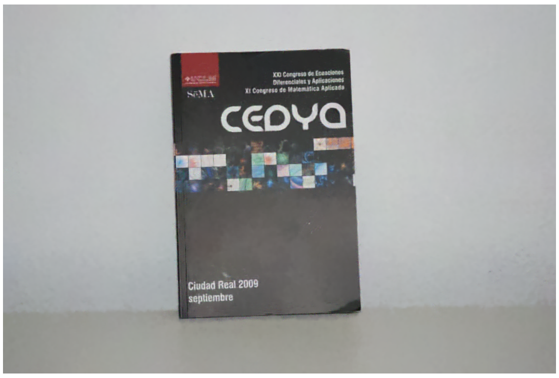


Denoised image at 1st step PSNR = 30.5292

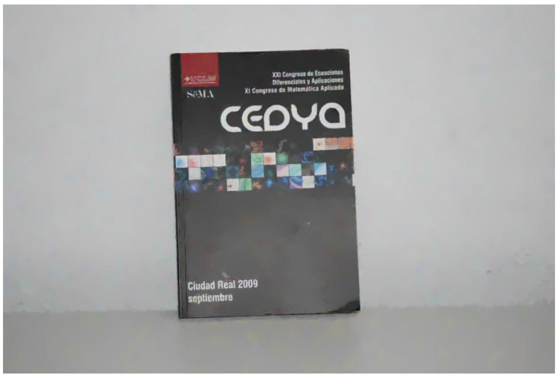


Denoised image at 2nd step PSNR = 34.3985

And finally, at the first and at the second.



Denoised image at 1st step PSNR = 34.449



Denoised image at 2nd step PSNR = 34.5625

As we can see, the most effective usage of the homogeneous area criterion is only at the first step. Therefore, in the future, we will use only at the first step.

2.3. Comparison with BM3D

Let's compare with BM3D but using different levels of noise.

Original image

Noisy image with sigma = 10

Denoised image with NL-Bayes PSNR = 35.1448

Denoised image with BM3D PSNR = 34.5339

Difference image with NL-Bayes

Difference image with BM3D

In terms of PSNR NL-Bayes showed better performance at low noise, but visually BM3D ahs better edges than NL-Bayes. The difference images don't show anything important all shapes are hard to distinguish, but for NL-Bayes it is easier. Finally, the difference is not that big and visually both algorithms performed well. Now we will use sigma = 30.

Denoised image with NL-Bayes PSNR = 28.9873

Denoised image with BM3D PSNR = 28.8618

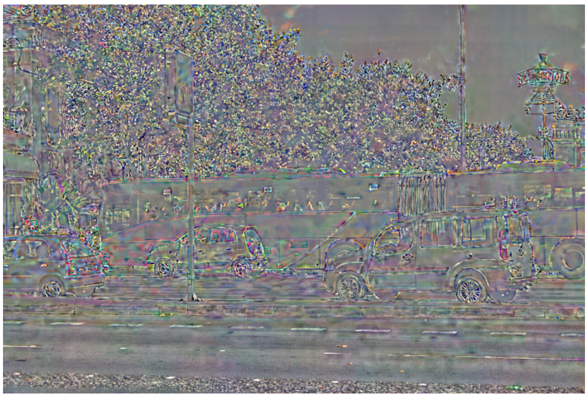
Difference image with NL-Bayes

Difference image with BM3D

Again, we have better PSNR value for NL-Bayes and now some artifacts are present for the BM3D, especially at sky area. At difference images of the NL-Bayes algorithm, we have a huge amount of randomness in their pixel values, mainly in the textured regions and regions containing edges and sharp transitions. However, the difference image given by the BM3D algorithm shows much less randomness. And finally, we will use a noisy image with sigma = 50.

Denoised image with NL-Bayes PSNR = 26.8538

Denoised image with BM3D PSNR = 26.4329



Difference image with NL-Bayes

Difference image with BM3D

So once again, there is better PSNR value for NL-Bayes and visually not so much difference. Here, we have the same observations as for sigma = 30.

2.4. Conclusion

The NL-Bayes and BM3D have similar structures, but some results are different. For example, for low noise values, BM3D visually performed better, even though PSNR was higher for NL-Bayes. For some parts of an image like texture and edges, the BM3D performed worse, especially at high noise, and the artifacts were visible. However, both of the algorithms have shown decent performance, and the main difference is that for lower noise we should use BM3D, and for higher - NL-Bayes.