Master MVA
Image Denoising

Experimental Report

Realized By: Eya Ghamgui

An Analysis and Implementation of the BM3D Image Denoising Method

The BM3D grouping and filtering procedure is a method used in image denoising. It is an extension of the DCT and the NL-means denoising methods. It differs from the latter method by its 3D group processing which includes spatial dimensions. In addition, it takes into account the sparsity of the image to improve the filtering procedure. The BM3D method is based on two steps. In the first step, the algorithm computes an initial estimate of the denoised image using the hard thresholding. In the second step, the algorithm uses this estimate and the original noisy image to compute a new estimate using Wiener filtering.

More precisely, the first step is decomposed into three parts: the grouping, the collaborative filtering, and the aggregation. In the grouping part, for each pixel "p", all patches similar to the patch centered in this pixel are grouped together. The similarity between patches is considered as the distance threshold. The patches with the minimum distances including the patch centered in "p" are grouped and sorted into a 3D block. The collaborative filtering is illustrated by applying two transformations on the 3D block. The first one is a 3D isometric linear transform. Then, it is followed by a shrinkage method which is based on the hard thresholding. Finally, it ends with the inverse of the linear transformation. The aggregation step is necessary to combine all these redundancies in order to estimate the value in pixel "p". It is based on a particular averaging procedure to compute the weighted sum. The role of the calculated weights is to give importance to homogeneous patches and to reduce the effect of those containing edges.

The second step of the algorithm is added to improve the result of the first step. This step is decomposed into three parts similar to the previous step. However, the difference is that in the grouping part, the patches from the previous estimation are grouped with ones having the same order from the noisy image. The second difference is in the collaborative filtering. The algorithm uses Wiener filtering instead of hard thresholding. Finally, the aggregation is identical to the first step, the only change is the use of Wiener's coefficients.

This algorithm is not only used for grayscale images but also for color images. A few additional steps are performed to solve this issue. First, the image must be transformed into luminance-chrominance space. Then, the 3D block is constructed using only the luminance channel. This block is next applied to the other channels. In addition, collaborative filtering and weighted aggregation are performed separately for each channel. Finally, the denoised image is returned to the original channel space (RGB).

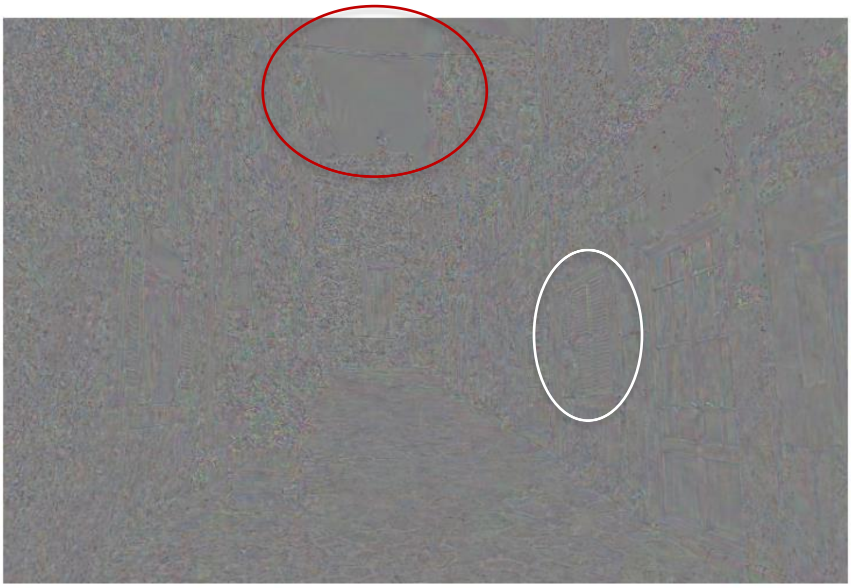
1. **For σ = 30**

Original Image

Noisy Image

First Denoising ; PSNR = 28.93 dB

Final Denoising ; PSNR = 29.3 dB



Difference Image

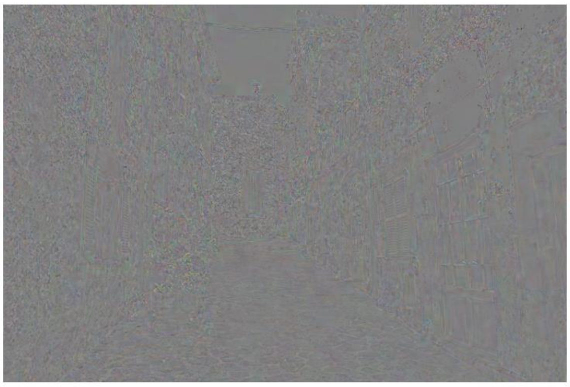
- Interpretations:

From the previous images, we can notice that this method is very efficient as it succeeded in denoising a multi-structured image. We can remark from the difference image that this method is outperforming in homogenous areas and flat regions such as the sky region which is circled in red. Moreover, the method preserves well the contours and the edges, this is due to the weighting of the patches which gives more importance to the homogeneous patches and give less importance to the patches containing edges. However, we still notice some original aspect of the image in the difference image. These forms correspond to the highly textured areas in the original image, for example, the shape of the window which is circled in white. In fact, the algorithm failed to differentiate between these edges and the noise. That is, we can say that even though we managed to denoise the image and reconstruct the original one, we still have a loss of information in the images due to the loss of fine details.

Moreover, we notice that the second step has remarkably improved the results. Indeed, the PSNR has increased from 28.93 dB to 29.3 dB. We can say that this step restores additional details lost in the first step and improves the denoising performance by taking the grouping of the estimated image from the first step with the noisy image. In addition, this method improves the contrast of the edges and acts properly to eliminate artifacts. This result is due to the use of Wiener filtering.

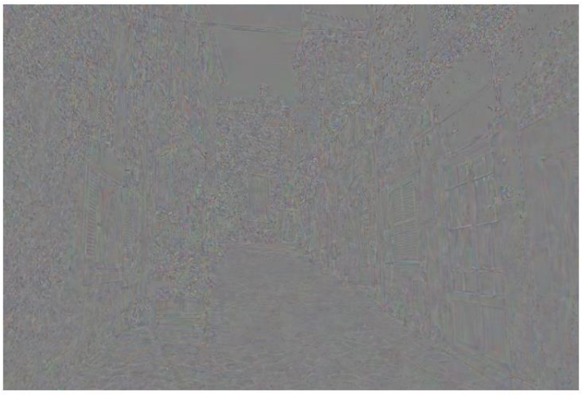
2. For different values of σ

Final Denoising ; σ = 40



Difference Image

Final Denoising ; σ = 50



Difference Image

	σ = 30	σ = 40	σ = 50
PSNR of the first filtering	28,9549	27,6245	26,5303
PSNR of the final filtering	29,3333	28,0676	27,0973

- Interpretations:

By increasing the variance of the noise, we notice that the algorithm makes more errors in the denoising part. Also, more artifacts are created in the denoised images. This is noticed from the difference image. In fact, more details of the original image become visible. That is, the more the noise value increases, the worse the denoising is.

From the above table, we can notice that the denoising results (PSNR) decrease as σ increases. Moreover, we notice that the second step improves the denoising results for all noise values. However, the increase in PSNR value is slightly significant between the first and second step. In addition, this method takes a lot of computational time. Thus, the result of the first step is considered sufficient and acceptable for the denoising task since the second step takes more computational time and slightly improves the results. We can conclude that this method is more complex, less flexible, and slower than basic methods such as DCT threshold denoising.

Non-Local Means Denoising

Non-local averaging is one of the denoising methods. This algorithm uses redundant information that is common in natural images to remove noise. For a given pixel, denoising consists in replacing its value by the weighted average of intensities in similar pixels. The similarity between pixels is determined by a search in the entire image, as indicated by the name of the method. It is implemented in two versions: pixelwise implementation and the patchwise implementation.

In the first version, the algorithm searches for similar pixels. For a given pixel, the similarity is computed using the Euclidean distance. Then, the value of each pixel is stored as a weighted average of intensities of the most similar ones. These weights are computed using an exponential kernel and depend on the squared Euclidean distance. In fact, they are introduced to reduce the effect of noise by exponentially reducing the effect of pixels whose distance is greater than the noise value.

In the second implementation, for a given pixel, the algorithm searches for patches similar to the patch centered in that pixel. Then, an estimate for the patch is computed using these similarities. Several estimates for the pixel are computed based on different patches. Then, these estimates are averaged at each pixel location to construct the final denoised image. This version therefore reduces noise and eliminates artifacts around the edges.

1. **For σ = 30**

Original Image

Noisy Image

Denoised Image ; PSNR = 27.82 dB

Difference Image

- Interpretations:

From the previous results, we can notice that this method provides very good denoising performance. Overall, this method reconstructs all parts of the image very well and preserves the appearance of the image. However, many fine details are removed. For example, in the alley, small details become blurred. This result can also be noticed in the difference image.

In addition, we can notice that lines and highly textured elements are removed by the denoising procedure. This is due to the fact that denoising consists of taking the average of pixels that are similar to neighboring pixels. This action did not take into account the presence of edges. In the previous paper, this result is improved by giving each patch a weight that indicates whether it is homogeneous or contains edges.

2. For different values of σ

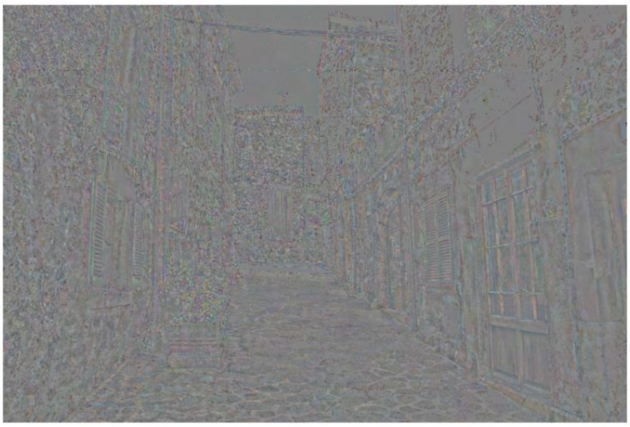
- σ = 50

Denoised Image ; PSNR = 25.42 dB

Difference Image

➢ σ = 20

Denoised Image ; PSNR = 29.87 dB



Difference Image

- Interpretations:

From the previous results, we can notice that the increase of noise values leads the algorithm not to reconstruct the fine structures of the image. Moreover, the denoised image becomes more blurred. Add to that, the difference image becomes globally similar to the original one. Indeed, the PSNR decreased to 25.42 dB for σ = 50. However, with small noise values, the algorithm gives very high performance with PSNR equal to 29.87 dB.

Comparison between the algorithms

Original Image

Noisy Image

BM3D

NL-Means

	BM3D	***NL-Means***
PSNR	37.6	36.83
RMSE	3.36	3.67

After all these experiments, we can say that these two algorithms are among the best denoising algorithms. Indeed, both algorithms are able to remove a considerable amount of noise and reconstruct the image structures.

Moreover, we can notice that BM3D obtains the highest PSNR and the lowest RMSE. The denoised images also show that the effect of BM3D is better than NL-Means, with less noise and better image details. Indeed, the BM3D takes into account the presence of edges in patches by giving low weights to patches containing edges and high weights for homogeneous patches.

From a visual perspective, the image denoised by the NL-Means method seems more denoised and closer to the original image. In fact, this image contains a lot of flat regions that make NL-Means respond better in denoising. However, if we focus on the details and edges, this algorithm cannot reconstruct them, for example, the line on the edge of the cube. But the MB3D is able to recognize this small details.