Block Matching and 3D-Collaborative filtering based image denoising

Ajinkya Ambatwar
EE16B104
Dept. Of Electrical Engineering

April 23, 2020

1 Introduction

BM3D is a recent denoising method based on the fact that an image has a locally sparse representation in transform domain. This sparsity is enhanced by grouping similar 2D image patches into 3D groups. In this report we look at implementation of the method. The methodology is taken from original paper and tested for a specific/suitable value of noise standard deviation. It can be easily tested with other standard deviation by tweaking the corresponding parameter.

Collaborative filtering is the name of the BM3D grouping and filtering procedure. It is realized in four major steps: 1) finding the image patches similar to a given image patch and grouping them in a 3D block 2) 3D linear transform of the 3D block; 3) shrinkage of the transform spectrum coefficients; 4) inverse 3D transformation. This 3D filter therefore filters out simultaneously all 2D image patches in the 3D block. By attenuating the noise, collaborative filtering reveals even the finest details shared by the grouped patches. The filtered patches are then returned to their original positions. Since these patches overlap, many estimates are obtained which need to be combined for each pixel. Aggregation is a particular averaging procedure used to take advantage of this redundancy.

Generally this first step is followed by weiner filtering instead of hard thresholding. But for this report, we are not going to implement thi because of two reasons 1) the time consumption for running each step 2) The output was doing better compared to another competitive algorithm Non-Local means(NLM) even after 1st stage itself.

2 Algorithm Details

2.1 Algorithm Architecture

The algorithm is divided in two major steps:

1. The first step estimates the denoised image using hard thresholding during the collaborative filtering.

2. The second step is based both on the original noisy image, and on the basic estimate obtained in the first step. It uses Wiener filtering.

2.2 First denoising step

We denote by p the reference current patch whose size is $k^{hard} \times k^{hard}$ of the image loop.

2.2.1 Grouping

The original noisy image is searched over a $n_{hard} \times n_{hard}$ patch centred at p for the similar patches q that satisfy

$$P(p) = \{Q : d(p,q) = \leq \tau^{hard}\}$$

where:

- τ^{hard} is the distance threshold for d under which two patches are assumed similar;
- $d(p,q) = \frac{||\gamma'(p)-\gamma'(q)||_2^2}{(k^{hard})^2}$ is the normalized distance between the patches
- γ' is the hardthreshold operator with threshold $\lambda_{2D}^{hard}\sigma$. It simply puts to zero the coefficients of the patch with an absolute value below the threshold$\lambda_{2D}^{hard}\sigma$. With experiments performed by various researchers it was found that for $\sigma \leq 40$, the threshold is 0.
- σ^2 is the variance of the zero-mean Gaussian noise.

For computational efficieny we earch over every 5th($N_{step} + 1$) $k^{hard} \times k^{hard}$ patch in the $n^{hard} \times n^{hard}$ search window.

The 3D group is obtained by stacking the patched in P one over the other. We select top N_{hard} patches for reconstruction fron the transformed domain.

2.2.2 Collaborative Filtering

Once the 3D block is obtained collaborative filtering is applied. A 3D isometric linear transform is applied to the grou, followd by shrinkage in the transform domain. Finally inverse transform is taken in order to reconstruct the image.

$$P^{stacked}(P)^{hard} = \tau_{3D}^{hard^{-1}}(\gamma(\tau_{3D}^{hard}(P^{stacked}(P))))$$

where γ is a hard thresholding operator with threshold $\lambda_{3D}^{hard}\sigma$ which zeros out the values with absolutes less than the threshold.

2.2.3 Aggregation

A given patch can be grouped for multiple refernce patch and hence maybe present in multiple 3D blocks. Hence all blocks are weighted added in order to get the reconstructed val. This is Aggreation. The weight given to each block $P^{stacked}(P)^{hard}$ in which the given patch is present is the inverse of number of non zero(and positive) elements(N_p^{-1}) in the block. The interest of this weighting is that it gives a priority to homogeneous patches (where there are many canceled coefficients). Patches containing an edge will be less taken into account than homogeneous ones on the border of the edge. The figure below illustrates this

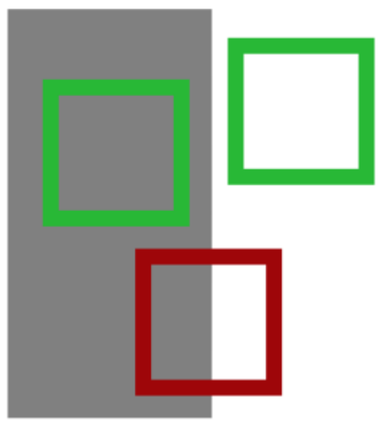


Figure 1: Green patches will have more weightage than the red one because they are more sparse(have less non-zero transform corfficients)

The basic estimate will be

$$\forall q \in P(p), \forall x \in q, \begin{cases} v(x) = v(x) + w_p^{hard} u_{Q,P}^{hard}(x) \\ \delta(x) = \delta(x) + w_p^{hard} \end{cases}$$

where $w_p(x)$ is the weight given by

$$w_p^{hard} = \begin{cases} \frac{1}{Np} & , N_p \geq 1 \\ 1 & , else \end{cases}$$

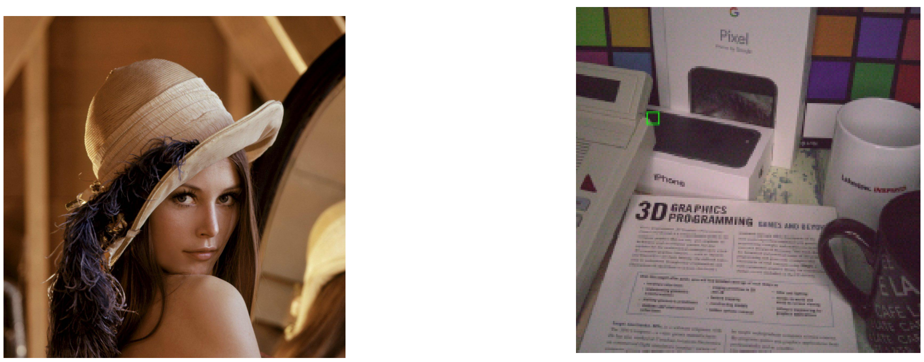
where N_p is the number of non-zero elements after hard thresholding. v is the aggregated value while δ is the nomalization factor.

3 Experiment

For our experiment, we choose the parameters as mentioned in the table 1(taken from the original paper)

Parameter	Value
k^{hard}	8
n^{hard}	39
λ_{2D}^{hard}	0
λ_{3D}^{hard}	2.7σ
N_{hard}	8
τ_{hard}^{2D}	2D-DCT
τ_{hard}^{3D}	3D-DCT
N_{step}	4

Table 1: Parameter values‘



(a) Lena

(b) From Smartphone Image Denoising Dataset

Figure 2: Test Images

Two images were tested out for the denoising task and were compared with Non-local means[1]. The images that were tested are shown in the figure 2.

Both the images were cropped and resized to $256 - 2 \times n^{hard}$ and then padded with n^{hard} mirror imaged samples on all sides to make the image size (256,256). A zero mean gaussian noise is added to the image with (matlab imnoise) variance of σ^2 where we choose $\sigma = 0.1225$ and 0.2000. Now we process the image channels independently and get the results shown in figure 3 and 4.

For the given setup following results were obtained

(a) Lena

(b) From Smartphone Image Denoising Dataset

Figure 3: Test Images

Lena $\sigma = 0.1225$

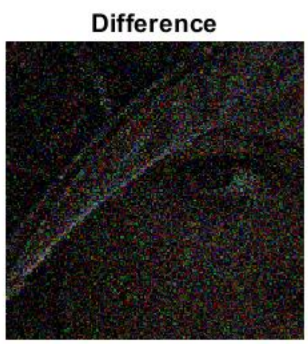
(a) Lena noised

(b) Lena Denoised

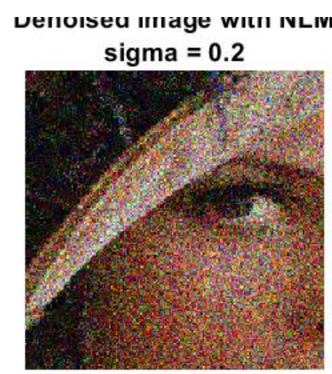
Lena $\sigma = 0.2$

(a) Lena noised

(b) Lena Denoised



(c) Difference



(d) Lena NLM denoised

Figure 5: Lena denoised at 0.2(imnoise sigma)

Dataset $\sigma = 0.1225$

(a) Dataset noised

(b) Denoised

(c) Difference

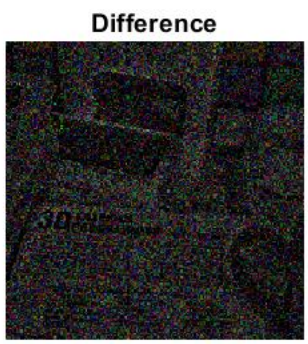
(d) NLM denoised

Figure 6: Dataset denoised at 0.1225(imnoise sigma)

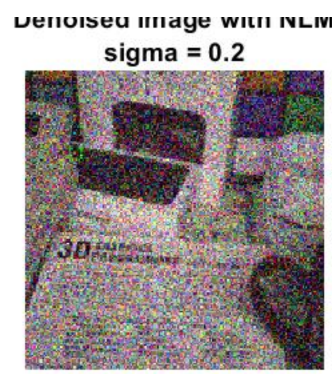
Dataset $\sigma = 0.2$


(a) Dataset noised


(b) Denoised



(c) Difference



(d) NLM denoised

Figure 7: Dataset denoised at 0.2(imnoise sigma)

The PSNR and SSIM values obtained are shown in table 2-5.

Sample	PSNR(dB)	SSIM
Lena 0.1225 Noisy	18.9578	0.4331
Lena 0.1225 BM3D P1	23.0265	0.7407
Lena 0.1225 NLM	22.5884	0.6894

Table 2: Lena @0.1225

Sample	PSNR(dB)	SSIM
Lena 0.2 Noisy	14.99	0.3282
Lena 0.2 BM3D P1	20.5716	0.6973
Lena 0.2 NLM	15.2685	0.3376

Table 3: Lena @0.2

Sample	PSNR(dB)	SSIM
Dataset 0.1225 Noisy	18.41	0.3451
Dataset 0.1225 BM3D P1	20.1010	0.6631
Dataset 0.1225 NLM	21.33	0.4785

Table 4: Dataset @0.1225

Sample	PSNR(dB)	SSIM
Dataset 0.2 Noisy	14.448	0.1845
Dataset 0.2 BM3D P1	19.104	0.5167
Dataset 0.2 NLM	14.4866	0.1850

Table 5: Dataset @0.2

4 Conclusion

We found that BM3D beats NLM in terms of PSNR and SSIM values of the ouput obtained. If the weiner filtering ie.the 2nd step is added to BM3D the output is going to get better in visual as well as theoretical aspect.

Hence we studied the BM3D denoising algorithm and compared it against a standard NLM algorithm

References

[1] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, pp. 60–65, IEEE, 2005.