

# An Approach for Modification of Block Match Three Dimension De-Noiseing Algorithm with Pre-Filtering In Nonlocal Domain

Kamalakshi N<sup>1#</sup>, ,Dr M N Shanmukhaswamy<sup>2</sup>

<sup>1</sup> Research Scholar JSSRF,Mysore ,UOM

#Dept of Computer Science &Engg Sapthagiri College of Engg ,Bangalore VTU,Belagaum,India

<sup>2</sup>Professor and Head Dept of Electronics & Communication SJCE Mysore , India

<sup>1</sup> [kamalnags@gmail.com](mailto:kamalnags@gmail.com), , <sup>2</sup>[mnsjce11@gmail.com](mailto:mnsjce11@gmail.com)

**Abstract.**BM3D (Block Matching three dimension)is one of the popular de-noising state of art algorithm which works on transform domain .It has lot of engineering in it. Even though BM3D outperforms all other de-noising algorithms, the performance degrades as sigma value increases ,also some artifacts are introduced as the noise level increases. To overcome this drawbacks a new modified approach is proposed by adding an additional step for pre-filtering & removing noise by edge detection. Experimental results depicts that proposed approach gives better results in terms of PSNR (Peak signal to Noise Ratio) and visual quality with a reasonable computational burden

Keywords:

## 1. Introduction

Image processing is a very renowned area of research. But the de-noising is quite a old area of interest which has lot of real time application. With the advent of various technology and well equipped devices for capturing an image but some noise is introduced always and de-noising which is a preprocessing step is one of the challenging task in the area of image processing. Vast Literature exists which depicts emerging work in the de-noising so depends on some assumptions about the removal of random noise.

Presently, patch based de-noising image processing has become very eminent using which a local patches (Patch(block or Fragment ):is a sub-image of an image of size  $N \times N$ ) has been adopted by most of the researchers and hence very popular due to its highly effective .Taking advantage of the redundancy of small sub-images inside the image of interest. The core idea behind these and many other contributions is the same, for given the image to be processed, extract all possible patches with overlaps; these patches are very minute compared to the image size (a typical patch size would be  $8 \times 8$  pixels). The processing takes by operating on these patches and furnishing interrelations between them. The final image is formed by the modification of patches (or sometimes only their center pixels) and later put them back into the original position . The paper is organized as section II discuss about related work, section III about proposed work and section IV about the results & Discussion & Finally conclusion

## 2. Related Work

BM3D is currently known as the state of the art method for image de-noising and outperforms all other algorithm when it comes to de-noise AWGN at a reasonable computational cost[1]. BM3D depends on the presumption that an image has a locally sparse representation in the transform domain. In BM3D the sparsity is enhanced by grouping similar 2D fragments into 3D data array which the authors called “groups”. Because each block in the group was chosen according to some similarity measure with respect to a reference block, the use of a higher dimensional filtering of each group was possible. This exploits the potential similarity between grouped blocks to estimate the true signal in each of them by producing a highly sparse representation in 3D transform domain, so that the noise can be removed by wavelet shrinkage. This approach of exploiting similarity and estimating the original signal is called as collaborative filtering.

### Issues in BM3D

BM3D has few lacunae and modification are still possible, especially for higher noise levels. BM3D introduces many artifacts when the noise level is high ( $\sigma > 40$ ). The performance of noise reduction also significantly drops as the noise level increases. When the noise level is large, block matching is not reliable any more, as blocks which are not similar to the referenced block can easily be grouped together into the 3D array, resulting in less sparser representation in transform domain. It also tends to give poor visual results when exposed to micro-textured zones in natural images. Another important drawback of BM3D is that it blurs sharp edges, as it uses a weighted averaging at the end of each step (i.e. first step and second step), which works more like a low pass filter. BM3D also reduces the contrast of images which can easily be observed in almost all of its denoised images. Also, if the image is highly contaminated by noise, the image features becomes inseparable from the noise itself. In these situations, though it preserves the fine image details after denoising, it can blur the edges after the collaborative filtering and aggregation step I. In the transform domain the edges and the noises are not distinguishable and this forces the filtering process to remove some of the edge information making edges blurry.

## 3. Proposed Algorithm

The unique feature of this algorithm compared to BM3D is that it does a pre-filtering that is the first step where in it identifies whether a pixel is corrupted or no and the remaining two steps are similar to BM3D. The algorithm consists of three steps

**Step1: Identifying a noisy pixel (based on current pixel & its neighborhood values)**

**Step 2: To determine the basic estimate**

**Step 3: Estimation of the true image from the basic estimate and the noisy image**

### IV Noise Mathematical Model

Assume a noise model with true image  $y$  being corrupted by AWGN as

$$z(x) = y(x) + \eta(x), \quad x \in X \tag{1}$$

where  $x$  denotes a 2D index or location or spatial coordinate belonging to the image domain  $X \subset Z^2$ ,  $y$  is the true image, and  $\eta$  is i.i.d. zero-mean Gaussian noise with variance  $(0, \sigma^2)$ .

### Modified BM3D Proposed Algorithm

The algorithm consists of three steps

#### Step1: Identifying a noisy pixel (based on current pixel & its neighborhood values)

- a. Use a sliding window to identify the noisy pixel and restore it
- b. Apply a median filter to reduce the noisy pixels from image before applying the actual denoising method
- c. Designate current pixels as  $P_x$ . Compare each pixel with its neighboring pixels
- d. If  $P_x$  is corrupted
  - i.  $P_x$  value is maximum or minimum of all other pixel values in the corresponding neighbor
  - ii. Sort the pixel value except  $P_x$  of the filter window  $w_x^n$  in ascending order
  - iii.  $s(x) = (s_1(x), s_2(x), \dots, s_8(x))$

$$\partial \square(x) = \begin{cases} 1, & \{P_x < \min\{s(x)\} - \tau_{pre}\} \vee \{P_x \geq (\max\{s(x)\} - \tau_{pre})\} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Where  $\tau_{pre}$  is a threshold value  
 if  $(\partial \square(x) = 1)$

then apply median filter on the window  $w_x^n$  to determine the value of the pixel and restore it

$$\text{med} \square f(x) = \begin{cases} \{s(x) \in w_x^n, s(x) \neq p_x\}, & \text{if } \partial \square(x) = 1 \\ P_x, & \text{otherwise} \end{cases} \tag{3}$$

else restore the original pixel

#### Step 2: To determine the basic estimate

Here, we get the basic estimate using the pre-filtered version of the noisy image. We follow the following procedure for two different block sizes of  $l$  and  $l'$  depending on the noise level present in the examined image.

- i) To identify the patches similar to the reference patch or noise standard deviation crosses 39 denoising performance has a sharp drop. To overcome this issue, it is advised to use coarse prefiltering to measure the block-distance. This prefiltering is realized by applying a normalized 2-D linear transform on both blocks and then hard-thresholding the obtained coefficients, which results in

$$d(Z_{xR}, Z_x) = \frac{\|\gamma'(\tau_{2D}^{ht}(Z_{xR})) - \gamma'(\tau_{2D}^{ht}(Z_x))\|_2^2}{(N_1^{ht^2})} \tag{4}$$

where  $\gamma'$  is a hard threshold operator and  $\tau_{2D}^{ht}$  is a 2-D linear unitary transform operator. Each group is made by stacking together maximum  $N_1^{ht^2}$  similar noisy blocks, with similarity distances less than a predefined threshold  $\tau_{match}^{ht}$  match in a 3-D array form, R To stack the similar patches to form a 3D group

$$S_{xR}^{ht} = \{x \in X : d(Z_{xR}, Z_x) \leq \tau_{match}^{ht}\} \tag{5}$$

Using the  $d$ -distance (4) is a set  $S_{xR}^{ht}$  which contains the coordinates of blocks similar to  $Z_{xR}$ ,  $\tau_{match}^{ht}$  is the maximum for which two blocks are similar

- ii) An effective collaborative filtering can be asumed as shrinkage in transform domain. Suppose  $d+1$ -dimension groups of similar signal fragments are already grouped, the collaborative shrinkage comprises of the following steps.
  - For each group Introduce a  $d+1$ -dimensional linear transform.
  - To attenuate noise shrink (by hard-thresholding in step 1| Wiener filtering in step 2) the transform coefficients
  - To get estimates of all grouped fragments invert the linear transform to produce.
- iii) Three dimension Transform of a group is done by hard thresholding of the transform coefficients, then inverse the transform and return the estimates of the block to their original positions

$$\hat{Y}_{S_{xR}^{ht}}^{ht} = \tau_{3D}^{ht^{-1}}(\gamma(\tau_{3D}^{ht}(Z_{S_{xR}^{ht}}))) \tag{6}$$

$$\gamma(x) = \left\{ \begin{array}{l} 0, \text{ if } |x| \leq \lambda_{3D}^{hard} \sigma \\ x, \text{ otherwise} \end{array} \right\} \tag{7}$$

- iv) Use weighted average of all the obtained overlapping estimates to obtain the basic estimate of the image

$$\omega_X^{ht} = \begin{cases} \frac{1}{\sigma^2 N_{hat}^{X_R}}, \text{ if } N_{hat}^{X_R} \geq 1 \\ 1, \text{ otherwise} \end{cases} \tag{8}$$

Once the 3D block is done collaborative filtering of  $Z_{S_{xR}^{ht}} (N_1 \times N_1 \times |S_{xR}^{ht}|)$ , estimate aggregation After processing all reference blocks, we have a set of local block estimates  $\hat{Y}_{S_{xR}^{ht}}^{ht}$ , ; $x_R \in X$  (and their corresponding weights  $S_{xR}$ , ; $x_R \in X$ ), which forms an overcomplete representation of the estimated image due to the overlap between the blocks. The final estimate is given by

$$\hat{Y}^{basic}(x) = \frac{\sum_{X_R \in X} \sum_{x_m \in S_{X_R}^{ht}} \omega_{X_R}^{ht} Y_{X_m}^{ht}(X)}{\sum_{X_R \in X} \sum_{x_m \in S_{X_R}^{ht}} \omega_{X_R}^{ht} X_{X_m}(X)}, \forall x \in X \quad \dots \dots \dots (9)$$

**Edge map building**

For the preliminary basic estimation apply the Canny edge detector to build a edge map. The Canny edge detector prominent the regions with high spatial derivatives by applying a Gaussian smoothing with a kernel size to reduce the level of noise in the input image, followed by the processing of derivative in both x and y direction to get the magnitude and direction of gradient,

$$d = \sqrt{(D^2_x(x, y) + (D^2_y(x, y))} \tag{11}$$

Magnitude

$$\theta = \arctan\left(\frac{D_x}{D_y}\right) \tag{12}$$

where D is the magnitude of gradient and θ is angle of gradient. Given the values of the gradient, the algorithm applies non-maximum suppression, which suppresses any pixel that is not at the maximum. This is done by preserving all local maxima in the gradient image, and deleting everything else. Furthermore, the gradient array is reduced by hysteresis that uses two thresholds, and , determined empirically. Pixels with gradient magnitude are discarded, while pixels with gradient magnitude are kept as edges. If the magnitude is between the thresholds (i.e. ), it is only kept as an edge if and only if there is a path from this pixel to a pixel with gradient. Once we get the edge map by applying the Canny, we binarize the array values, where, represents the Canny edge detection process and represents the binarization method. The obtained edge guide is then used to combine the values from the two preliminary basic estimates and to form the final basic estimate. This is done by processing each pixel in a raster scan fashion and obtaining the edge strength for that location based on its neighboring pixels.

The PDF of noisy (Z<sub>x</sub>,Z<sub>x</sub>), are likely to overlap heavily and this results in error grouping ie., blocks with greater distances than the threshold are matched as similar, where as blocks with smaller distances are left out.

**Step 3: Estimation of the the true image from the basic estimate and the noisy image**

- The third step is similar to the original BM3D algorithm except instead of original noisy image we use the pre-filtered image as the input to this step.
- In this step, both the basic estimate and the pre-filtered noisy image are used to improve the denoising.

- We take advantage of the basic estimate by processing the grouping within the basic estimate and using collaborative Wiener filtering.
- The significant reduction of noise in the basic estimate allows us to use the normalized distance for grouping the similar blocks for reference blocks

Find out the locations of the blocks close to the reference one by block matching in the basic estimate. From the obtained locations, form two 3D groups which are from the noisy image and the basic estimate.

$$S_{X_R}^{wie} = \left\{ x \in X : \frac{\|\hat{Y}_{X_R}^{basic} - \tilde{Y}_{X_R}^{basic}\|_2^2}{(N_1^{wie})^2} \leq \tau_{match}^{wie} \right\} \quad (12)$$

Perform 3D transform to both the groups. And later perform wiener filtering on the noisy one with the energy spectrum of the basic estimate. The 3D transform domain collaborative Weiner filtering is done by element by element multiplication of Weiner shrinkage coefficient  $W_{S_{X_R}^{wie}}$  and the 3D transform domain coefficients of noisy data  $T_{3D}^{wie}(Z_{S_{X_R}^{wie}})$

$$W_{S_{X_R}^{wie}} = \frac{|T_{3D}^{wie}(\hat{Y}_{S_{X_R}^{wie}}^{basic})|^2}{|T_{3D}^{wie}(\hat{Y}_{S_{X_R}^{wie}}^{basic})|^2 + \sigma^2} \quad (13)$$

The 3D array  $\hat{Y}_{S_{X_R}^{wie}}^{wie}$  is made of the block estimates  $\hat{Y}_x^{wie, xR}$ ,  $\forall x \in S_{X_R}^{wie}$

$$\hat{Y}_{S_{X_R}^{wie}}^{wie} = T_{3D}^{wie^{-1}}(W_{S_{X_R}^{wie}}(T_{3D}^{wie}(Z_{S_{X_R}^{wie}}))) \quad (14)$$

The aggregation of the block wise estimate weights are computed as follows

$$\omega_{X_R}^{wie} = \sigma^{-2} |W_{S_{X_R}^{wie}}|^{-2} \quad (15)$$

The final estimate after wiener filtering is as follows

$$\hat{Y}^{final}(x) = \frac{\sum_{X_R \in X} \sum_{x_m \in S_{X_R}^{wie}} \omega_{X_R}^{wie} Y_{X_m}^{wie, X_R}(X)}{\sum_{X_R \in X} \sum_{x_m \in S_{X_R}^{wie}} \omega_{X_R}^{wie} X_{X_m}(X)}, \forall x \in X \quad (16)$$

This is the new algorithm devised by modifying BM3D.

#### 4. Results & Discussion

This section is contributed to the performance evaluation of proposed method. The experiments were conducted on images such as Barbara, Pepper, house, Lena, cameraman and compared with original BM3D[02], KSVD[09], NLMeans[08] & NLBayes[10]. The metrics used

for the performance evaluation is PSNR The analysis was done for images with noise levels (i.e. ) 5 to 100. Results presented here are acquired on exactly the same noisy image for both the BM3D and our proposed method. All of our images are 8-bit gray scale images and of dimension 512x512 , except House and Cameraman images, which have 256x256 pixels

The result obtained from the experiments proves that our proposed method shows better result than BM3D in most of images. The PSNR improvement is well for higher noise levels. During subjective evaluation we have shown that the images de-noised by our proposed method preserves edges and details in better manner .In terms of PSNR, our method demonstrates the best performance and outperforms other 4 methods in almost all noise levels for less textured images except in few cases. In particular, the addition of pre-filtering step produce PSNR values almost more than one decibel higher than the original BM3D algorithm for few sigma value..The results (PSNR in db )are tabulated in the following tables for different images

**Table 1:Output PSNR of the proposed BM3D algorithm for image Barbara & peppers**

Sigma	BM3D	KSV D	NLBayes	NL Means	Proposed	Sigma	BM3D	KSV D	NL Bayes	NL Means	Proposed
5	38.31	37.73	38.37	37.07	<b>38.81</b>	5	38.12	37.56	38.16	37.27	<b>38.81</b>
10	34.98	33.64	35.41	33.18	<b>35.88</b>	10	34.68	33.82	34.71	33.47	<b>35.88</b>
15	33.11	31.13	32.71	30.74	<b>34.81</b>	15	32.7	31.4	32.56	31.1	<b>35.01</b>
20	31.78	29.21	31.53	30.21	<b>33.01</b>	20	31.29	29.47	31.17	30.11	<b>33.87</b>
25	30.72	29.14	30.31	29.08	<b>32.16</b>	25	30.16	29.21	30.14	28.75	<b>33.08</b>
30	29.81	27.98	29.66	27.97	<b>31.15</b>	30	29.28	28.28	29.12	27.68	<b>33.01</b>
35	28.98	27.04	28.12	27.44	<b>31.27</b>	35	28.52	27.51	28.41	26.61	<b>31.08</b>
40	27.99	26.17	27.232	26.64	<b>29.79</b>	40	27.7	26.65	27.71	25.77	<b>29.8</b>
50	27.17	24.79	26.92	25.65	<b>28.80</b>	50	26.41	26.04	26.66	24.14	<b>28.8</b>
60	24.96	23.55	26.03	24.58	<b>27.72</b>	60	25.55	25.43	25.46	22.96	<b>27.72</b>
70	24.76	23.64	24.99	23.56	<b>27.07</b>	70	24.8	24.11	24.62	22.21	<b>27.07</b>
80	24.25	22.91	24.06	23.07	<b>25.5</b>	80	24.1	23.23	24.04	21.27	<b>25.94</b>
90	23.49	22.31	23.49	22.47	<b>24.58</b>	90	23.61	22.42	23.27	20.7	<b>24.58</b>
100	22.38	21.72	22.65	21.83	<b>23</b>	100	22.91	22.01	22.46	20.4	<b>23.05</b>

The table 1 depicts that PSNR of the proposed method outperformance all other method for image Barbara & pepper

**Table 2:Output PSNR of the proposed BM3D algorithm for image House & cameraman**

Sigma	BM3D	KSV D	NL Bayes	NL Means	Proposed	Sigma	BM3D	KSV D	NL Bayes	NL Means	Proposed
5	39.83	38.93	39.55	38.69	<b>38.9</b>	5	39.83	38.93	39.55	38.69	38.9
10	36.71	34.9	36.2	34.98	<b>35.88</b>	10	36.71	34.9	36.2	34.98	35.88
15	34.94	32.47	34.35	32.83	<b>34.89</b>	15	<b>34.94</b>	32.47	34.35	32.83	<b>34.89</b>
20	33.77	30.43	33.3	32.3	<b>34.01</b>	20	<b>33.77</b>	30.43	33.3	32.3	<b>33</b>
25	32.86	31.07	33.49	31.24	<b>33.1</b>	25	<b>32.86</b>	31.07	33.49	31.24	<b>33.1</b>
30	32.09	30	31.78	30.3	<b>32.7</b>	30	<b>32.09</b>	30	31.78	30.3	<b>32.7</b>



<b>35</b>	<b>31.38</b>	28.82	31.1	29.78	31.12	<b>35</b>	<b>31.38</b>	28.82	31.1	29.78	<b>31.12</b>
<b>40</b>	<b>30.65</b>	28.01	30.37	28.69	29.8	<b>40</b>	<b>30.65</b>	28.01	<b>30.37</b>	28.69	<b>29.8</b>
<b>50</b>	<b>29.69</b>	26.53	29.13	27.44	29.4	<b>50</b>	<b>29.69</b>	26.53	29.13	27.44	<b>29.4</b>
<b>60</b>	<b>28.74</b>	25.07	28.23	26.1	27.7	<b>60</b>	<b>28.74</b>	25.07	28.23	26.1	<b>28.7</b>
<b>70</b>	<b>27.51</b>	25.65	26.95	25.13	27.01	<b>70</b>	<b>27.51</b>	25.65	26.95	25.13	<b>27.01</b>
<b>80</b>	<b>27.16</b>	24.6	<b>26.51</b>	24.37	25.5	<b>80</b>	<b>27.16</b>	24.6	26.51	24.37	<b>25.5</b>
<b>90</b>	<b>26.48</b>	23.94	<b>25.99</b>	23.8	24.58	<b>90</b>	<b>26.48</b>	23.94	25.99	23.8	<b>25</b>
<b>100</b>	<b>25.87</b>	23.07	<b>25.12</b>	23.1	23.75	<b>100</b>	<b>25.87</b>	23.07	25.12	23.1	<b>24.75</b>

The table 2 depicts that PSNR of the proposed method is higher for few sigma values and lesser than BM3D and NLBayes

Table 3:Output PSNR of the proposed BM3D algorithm for image Lena

<b>Sigma</b>	<b>BM3D</b>	<b>KSVD</b>	<b>NLBayes</b>	<b>NLMeans</b>	<b>Proposed</b>
<b>5</b>	37.82	37.43	37.89	37.02	38.81
<b>10</b>	33.98	33.36	34.11	33.1	<b>35.88</b>
<b>15</b>	31.93	30.99	31.95	30.78	<b>34.08</b>
<b>20</b>	30.59	29.28	30.67	29.68	<b>33.71</b>
<b>25</b>	29.62	28.97	29.61	28.53	<b>33.01</b>
<b>30</b>	28.86	28.07	28.79	27.71	<b>31.93</b>
<b>35</b>	28.22	27.26	28.12	26.97	<b>29.94</b>
<b>40</b>	27.65	26.56	27.54	26.33	<b>28.44</b>
<b>50</b>	26.81	25.42	26.66	25.27	<b>27.8</b>
<b>60</b>	26.14	24.39	25.97	24.47	<b>27.42</b>
<b>70</b>	25.56	24.6	25.27	23.8	<b>26.94</b>
<b>80</b>	25.06	24.59	24.69	23.34	<b>25.94</b>
<b>90</b>	24.63	23.95	24.02	22.84	<b>25.3</b>
<b>100</b>	24.22	22.82	23.82	22.36	<b>24.67</b>

The table 3 depicts that PSNR of the proposed method is higher for few sigma values and lesser than BM3D and NLBayes for image lena



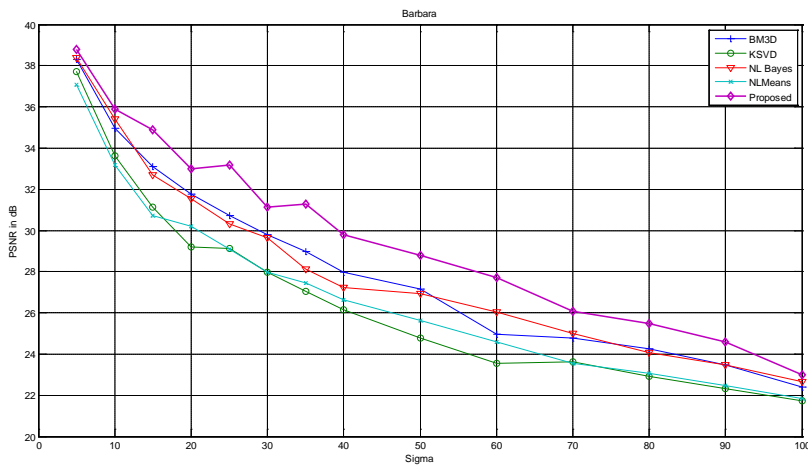
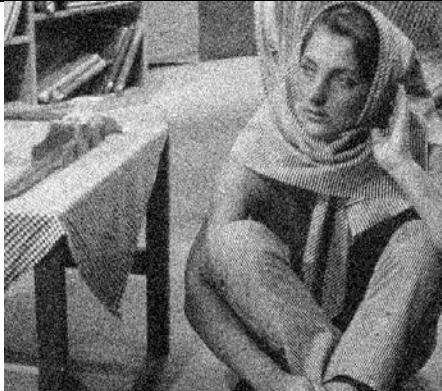





Figure 1: Comparison of BM3D,KSVD,NLBayes,NLMeans and Proposed method for Barbara Image

Table 3 Noisy & Denoised Image of Barabara for Sigma80,90 & 100

Sigma	Noisy Image	Denoised Image
80		
90		



**Computational Time:**The BM3D ,NLBayes, NLMeans and KSVD algorithms were taken & demonstrated from Image Processing Online on Intel Core(I3) CPU 2.13 MHz RAM 3.GB as in table 4

**Table 5:**Computational Time in seconds of BM3D ,KSVD ,NLMeans, NLBayes & Proposed Method

Images	Sigma	BM3D	KSVD	NLMeans	NLBayes	Proposed
Barbara	100	4.92	10.45	14.66	1.62	6.6
Peppers	100	2.12	3.02	3.54	1.02	6.2
Lena	100	4.83	11.34	15.46	2.12	6.1
House	100	2.02	2.42	4.35	1.02	5.8
Cameraman	100	2.02	2.72	3.74	1.22	4.81

From the above table it can be seen that proposed method imposes computational burden compared all other methods except for Barbara image due to the addition of pre-filtering step in the algorithm.

## Conclusion

From the experimental results it is found that the proposed method outperforms the state of art algorithm in terms of objective & subjective visual quality for AGWN noise with Reasonable computational cost compared to BM3D that is due to additional pre-filtering step at initial stage

## References

- 1.A. Buades, B. Coll, and J. M. Morel, .A review of image denoising algorithms, with a new *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490-530, 2005.
2. K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, .Image denoising by sparse 3D transform-domain collaborative filtering ., *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080-2095, August 2007.
3. M.C. Motwani, M.C. Gadiya and R.C. Motwani, "Survey of Image Denoising Techniques", proceedings of GSPx, Santa Clara, CA., Sep., 2004.
4. A. Buades, B. Coll, and J Morel. On image denoising methods. Technical Report 2004-15, CMLA, 2004.
- 5.A. Buades, B. Coll, and J Morel. A non-local algorithm for image denoising. IEEE International Conference on Computer Vision and Pattern Recognition, 2005.
- 6.M LEBRUN, *An Analysis and Implementation of the BM3D Image Denoising Method*, Image Processing On Line, 2 (2012), pp. 175–213. <http://dx.doi.org/10.5201/ipol.2012.1-bm3d>
7. Antoni Buades, Bartomeu Coll and Jean Michel Morel “An Implementation of Non Local (NL Bayes) Image denoising algorithm “IPOL 2011 pp 1-42 <http://dx.doi.org/10.5201/ipol.2011.16>
- 8.Marc Lebrun, Arthur Leclaire “An Implementation and Detailed Analysis of the K-SVD Image Denoising Algorithm “ IPOL 2012 pp. 175–213. <http://dx.doi.org/10.5201/ipol.2012.1-bm3d>
- 9.Antoni Buades, Bartomeu Coll and Jean Michel Morel “Implementation of Non Local Bayes”(NL Bayes) Image denoising algorithm IPOL 2013 pp 1-42 <http://dx.doi.org/10.5201/ipol.2013.16>