

## A MODIFIED BLOCK MATCHING 3D ALGORITHM FOR ADDITIVE NOISE REDUCTION

MONAGI H. ALKINANI AND MAHMOUD R. EL-SAKKA

*Abstract.* This paper presents a patch-based image filtering algorithm for additive noise reduction. Our algorithm is a modification to the block matching 3D algorithm, where an adaptive thresholding was used for the collaborative hard-thresholding step. The collaborative Wiener filtering step was also modified by assigning more weights for similar patches. Experimental results show that our algorithm outperforms the original block matching 3D algorithm at various noise levels.

### 1. INTRODUCTION

During images acquisition, compression or transmission phases, digital images are often contaminated with undesired random additive noise. This noise is generally modelled as:

$$v(x) = u(x) + n(x), \quad x \in \Omega$$

where  $v(x)$  is the noisy image,  $u(x)$  is the noise-free image,  $n(x)$  is the additive noise, and  $\Omega$  denotes the set of all pixels in the image. If  $n(x)$  is a Gaussian random process, then the noise is recognized as an additive Gaussian noise. Such a noise varies from being unnoticeable to being very visible. Image denoising schemes attempt to estimate a new image that is closer to the noise-free image.

Patch-based image denoising schemes segregate the noisy image into patches, or “blocks” using searching windows, and then manipulate those patches separately in order to provide an estimate of the true pixel values. Patch-based methods include Non-local Means filtering [2], Dictionary Learning filtering [1], Block Matching 3D filtering [3], and Image Denoising with Patch-based PCA filtering [4].

The Block Matching 3D (BM3D) filter is considered to be the state-of-the-art patch-based denoising method. It preserves edges and blurs homogeneous areas by exploiting similarities among the various parts of the input image by using a 3D transform-domain collaborative filtering.

---

*MSC (2010):* primary 34K06, 34K25.

*Keywords:* Block Matching 3D (BM3D) algorithm, patch-based image filtering, additive noise reduction, PSNR.

This research is partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC). This research is also partially funded by Jeddah University, Jeddah, Saudi Arabia. These supports are greatly appreciated.

In this work, the BM3D scheme is modified in order to improve its performance. Our proposed method uses an adaptive thresholding for the collaborative hard-thresholding step. In addition, it utilizes modified weights for similar patches in the collaborative Wiener filtering step.

The rest of the paper is organized as follows. Section 2 describes the methodology and the proposed method. In Section 3, the performance of the proposed method is compared with the performance of the original BM3D. Section 4 offers concluding comments.

## 2. METHODOLOGY

### 2.1. Block Matching 3D filtering

Block Matching 3D filtering is a sparse representation in the frequency domain. BM3D groups the patches into 3D data arrays instead of into 2D arrays, then it applies a transformation in the frequency domain. Collaborative filtering is used for dealing with the 3D arrays. BM3D's algorithm depends on two steps: (1) collaborative hard thresholding and (2) the collaborative Wiener filtering. The two steps allow the BM3D to suppress noise and to preserve more details. The noise is suppressed in the thresholding step, and the details are restored in the second step. The two steps have three functions: (1) 3D transformation, (2) shrinkage and (3) 3D inverse transformation. The patches in the 3D arrays are overlapped, so a weighted average is used for obtaining one estimate for each pixel. Aggregation is an averaging procedure. In the following subsections, the two steps of the BM3D algorithm are described. First, the collaborative hard thresholding step is explained. Then, the collaborative Wiener filtering is discussed.

#### 2.1.1. Step One: Collaborative Hard Thresholding.

Grouping: The grouping stage gathers similar patches to form 3D arrays. By benefiting from the high redundancy among the neighbouring patches, a searching window is used for gathering similar patches. There are several different patch grouping techniques; such as, vector quantization [6],  $k$ -means clustering [9], self-organizing maps [8] and others discussed in Jain et al. survey [7]. It is worth mentioning that the grouping in BM3D is based on the similarity between patches – the luminance distance between patches. Patches with a distance that is below a fixed threshold are considered to be similar and are grouped into the 3D array. Before measuring the distance, a coarse pre-filtering is used for linearly transforming the patches using a 2D linear transformation such as: multiple wavelet transforms [5, 10]. The formula in Equation (2.1) is used for computing the similarity distance between patches,

$$Dst([v]_i, [v]_j) = \frac{\left\| \gamma^{2D} (T_{hard}^{2D}([v]_i)) - \gamma^{2D} (T_{hard}^{2D}([v]_j)) \right\|_2^2}{(N_1^{hard})^2} \quad (2.1)$$

where  $[v]_i, [v]_j$  are the reference patches at  $i$  and its neighbour at  $j$  respectively,  $T_{hard}^{2D}$  is the 2D linear transform,  $\gamma^{2D}$  is a hard-thresholding operator equal to  $\lambda_{2D} \times \sigma$  and  $(N_1^{hard})^2$  is the patch size  $N \times N$ .  $\lambda_{2D} = 2.7$  when  $\sigma < 40$ , or  $\lambda_{2D} = 2.8$  when  $\sigma \geq 40$ . The  $\sigma$  is the estimated noise standard deviation. The

2D thresholding operator  $\gamma^{2D}$  makes all coefficients with absolute value less than the threshold ( $\lambda_{2D} \times \sigma$ ) equal to zero, and leaves the other coefficients unchanged. After computing the Euclidian distance, the formula in Equation (2.2) is used for gathering similar patches.

$$3D S_i^{hard} = \left\{ j \in \Omega : Dst([v]_i, [v]_j) \leq T_{match}^{hard} \right\} \quad (2.2)$$

where  $3D S_i^{hard}$  is the constructed 3D array containing similar patches, and  $T_{match}^{hard}$  is the maximum allowed similarity distance between two patches. The maximum grouped patch sizes are restricted to  $N_2^{hard}$ . The next stage is to apply the collaborative filter by: (1) performing a 2D linear transform then a 1D linear transform, (2) shrinkage, and (3) inverting the 1D transform and the 2D linear transform.

Collaborative Filtering: Once the 3D array is built, a collaborative filter is used for suppressing the noise. A 3D transform is applied to the 3D array, before the shrinkage of the transforming coefficients. The 2D transformation in the *Grouping* stage is applied along both horizontal and vertical lines for each patch, and here a third transformation is conducted along the third dimension of the 3D array for the 3D transform. The formulation of the collaborative filter is:

$$3D \hat{u}_{S_i^{hard}} = T_{hard}^{3D^{-1}} (\gamma^{3D} (T_{hard}^{3D} (3D S_i^{hard})))$$

where  $T_{hard}^{3D}$  is the 3D linear transform of the first (hard) step,  $T_{hard}^{3D^{-1}}$  is the inverse of 3D transformation, and  $\gamma^{3D}$  is a hard-thresholding operator equal to  $\lambda_{3D} \times \sigma$ .  $\lambda_{3D} = 2.7$  when  $\sigma < 40$ , or  $\lambda_{3D} = 2.8$  when  $\sigma \geq 40$ . The 3D thresholding operator  $\gamma^{3D}$  makes all coefficients with absolute value less than the threshold ( $\lambda_{3D} \times \sigma$ ) equal to zero.

Aggregation Weights: The overlapped patches of the 3D array ( $3D \hat{u}_{S_i^{hard}}$ ) have multiple estimates for each pixel in the reference patch at the location  $i$ . A weighted averaging procedure is required to provide an estimate for each pixel. Weights in BM3D are inversely proportional to the total variance of the patches in the  $3D \hat{u}_{S_i^{hard}}$  array. When the total variance is high, a small weight is assigned to the patch.

The amount of the additive noise is independent when processing the collaborative filter in *Step One* and *Step Two*. Thus, the total variance is not the same after applying the collaborative filter in the first and second steps. In *Step One*, the total variance is computed by  $\sigma^2 \times N_{non-zero}^{hard}$ , where  $N_{non-zero}^{hard}$  is the number of non-zero coefficients after the hard-thresholding. The total variance calculated in *Step Two* depends on the results of the Wiener filter coefficients. The weights of *Step Two* is explained 2.1.2. However, the weights for *Step One* is equal to:

$$\omega_i^{hard} = \begin{cases} \frac{1}{\sigma^2 \times N_{non-zero}^{hard}}, & \text{if } N_{non-zero}^{hard} \geq 1, \\ 1, & \text{otherwise.} \end{cases}$$

### 2.1.2. Step Two: Collaborative Wiener filtering.

Grouping: Grouping here is, in some ways, similar to the grouping in *Step One*; but here the power spectrums of *Step One* are grouped, not just the patches from the noisy image. The same formula is used:

$$3D S_i^{Wiener} = \left\{ j \in \Omega : \frac{\left\| \hat{u}_{S_{[v]_i}^{hard}} - \hat{u}_{S_{[v]_j}^{hard}} \right\|_2^2}{(N_1^{Wiener})^2} \leq T_{match}^{Wiener} \right\} \quad (2.3)$$

where  $\hat{u}_{S_{[v]_i}^{hard}}$  and  $\hat{u}_{S_{[v]_j}^{hard}}$  are the estimated patches from *Step One* at locations  $i$  and  $j$ , respectively. At this stage, there are two groups: (1) a group of similar patches derived from the noisy image and (2) a group of similar patches derived from *Step One*.

Collaborative filtering: After grouping the patches, a 3D transform is applied to the 3D array of the grouped patches. A Wiener shrinkage is applied to the transform coefficients of the 3D array. The definition of the Wiener shrinkage coefficients is shown in Equation (2.4):

$$3D W_{S_i^{Wiener}} = \frac{|T_{3D}^{Wiener} (3D S_i^{Wiener})|^2}{|T_{3D}^{Wiener} (3D S_i^{Wiener})|^2 + \sigma^2} \quad (2.4)$$

where  $T_{3D}^{Wiener}$  is the 3D linear transform and  $3D S_i^{Wiener}$  is the result of Equation (2.3). The final stage in the collaborative Wiener filtering is to multiply the Wiener shrinkage coefficients element-by-element by the 3D transform coefficients of the noisy image. The inverse of the 3D transform is applied. Multiplication and the inverse of the 3D transform are shown in Equation (2.5):

$$3D \hat{u}_{S_i^{Wiener}} = T_{3D}^{Wiener^{-1}} \left( 3D W_{S_i^{Wiener}} \times (T_{3D}^{Wiener} (3D \nu_i)) \right) \quad (2.5)$$

where  $3D \nu_i$  are the 3D transform coefficients of the noisy data.

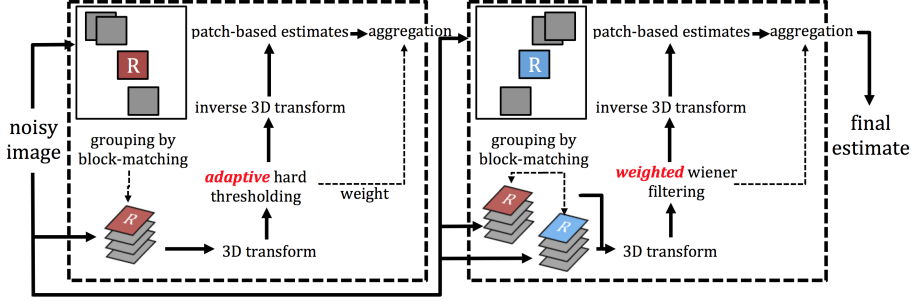
Aggregation Weights: Adjusting the weights, here, is not like the one in *Step One*, which depends on the number of non-zero coefficients reached after the hard-thresholding. The weights of this step depend on the Wiener shrinkage coefficients; assigned as in Equation (2.6):

$$\omega_i^{Wiener} = \sigma^{-2} \left\| W_{S_i^{Wiener}} \right\|_2^{-2}. \quad (2.6)$$

## 2.2. The proposed method

In this section, we describe an algorithm for solving the problem of denoising additive noisy images. The novelty of this algorithm is the use of an adaptive hard-thresholding operator for the shrinkage of the transforming coefficients, and the use of weighted average for the collaborative Wiener filtering step by using PSNR. The scheme in Figure 1 shows the proposed filtering scheme. In the following subsections, the two steps of the proposed algorithm are described. First, the collaborative hard thresholding step is explained. Then, using the collaborative Wiener filtering is discussed.

**2.2.1. Step One modification.** Collaborative filtering is the second stage of *Step One*, where noise is suppressed via using a shrinkage function after applying a 3D transform for the built 3D array. The original BM3D uses a static



**Figure 1.** The two steps of the proposed filtering scheme.

hard-thresholding operator as a shrinkage function for similar patches with low luminance distance. This function suppresses blindly all the transformed coefficients with absolute value from all patches of the 3D array under a threshold whether the coefficients are noise or not.

In order to improve the thresholding process, an adaptive hard-thresholding operator as a shrinkage function is proposed. We adapt the basic idea of the bilateral filter [11] to be used for the thresholding in BM3D algorithm. Similarity weights of the bilateral filter are based on geometric and luminance distances. We propose to use an adaptive hard-thresholding operator, where the operator's values depend on geometric and luminance distance similarities between patches. When a patch is geometrically far from to the reference patch, we enforce more thresholding to the patch transforming coefficients. The formulation of the proposed collaborative filter is:

$$3D \hat{u}_{S_i^{hard}} = T_{hard}^{3D^{-1}} (\gamma_{\Gamma}^{3D} (T_{hard}^{3D} (3D S_i^{hard})))$$

where  $T_{hard}^{3D}$  is the 3D linear transform of the first (hard) step,  $T_{hard}^{3D^{-1}}$  is the inverse of 3D transformation, and  $\gamma_{\Gamma}^{3D}$  is an adaptive hard-thresholding operator equal to  $\lambda_{3D} \times \sigma$ .  $\lambda_{3D}$ 's value is chosen adaptively based on the geometric and luminance distance similarities. When  $\sigma < 40$ ,  $\lambda_{3D}$ 's value is equal to:

$$\lambda_{3D} = \begin{cases} 2.7, & \text{if } \rightarrow \text{patch location} < \frac{N_2^{hard}}{2}, \\ 3.0, & \text{otherwise,} \end{cases}$$

and when  $\sigma \geq 40$ ,  $\lambda_{3D}$ 's value is equal to:

$$\lambda_{3D} = \begin{cases} 2.8, & \text{if } \rightarrow \text{patch location} < \frac{N_2^{hard}}{2}, \\ 3.1, & \text{otherwise} \end{cases}$$

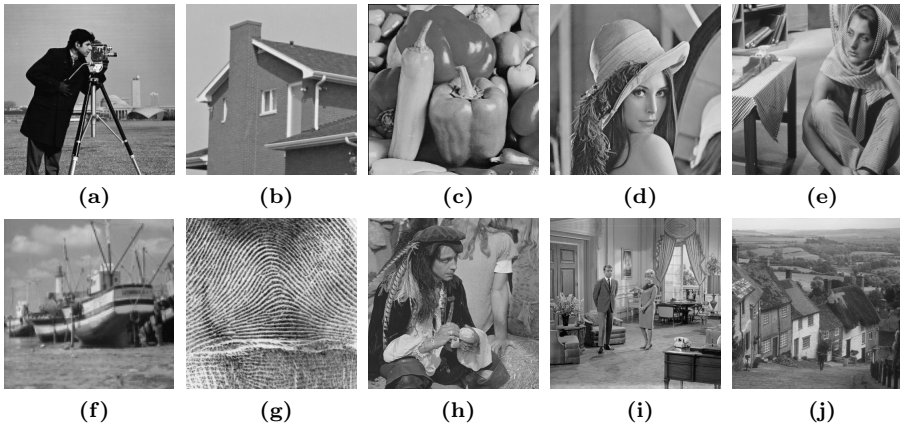
where  $N_2^{hard}$  is the number of the collected similar patches. The 3D thresholding operator  $\gamma_{\Gamma}^{3D}$  makes all coefficients with absolute value less than the threshold ( $\lambda_{3D} \times \sigma$ ) equal to zero.

**2.2.2. Step Two modification.** As a final stage in the BM3D collaborative Wiener filtering, the Wiener shrinkage coefficients of the 3D array and the 3D transform coefficients of the noisy image are multiplied element-by-element. Before

the multiplication process, Wiener shrinkage coefficients of the 3D array patches must be averaged. BM3D averages those coefficients arithmetically. We proposed to use weights for the 3D array patches; similar patches will be assigned more weights. PSNR is utilized as a patch similarity measure for assigning those weights. The inverse of the 3D transform is applied then.

### 3. EXPERIMENTAL RESULTS

The objective of this section is to experimentally study the performance of the proposed method at various noise levels. The original BM3D parameters are used for our method. We use  $8 \times 8$  patch size for low noise level ( $\sigma < 40$ ) and  $12 \times 12$  patch size for high noise level ( $\sigma \geq 40$ ). A fixed  $32 \times 32$  searching window size is used. The step size is 3, and the numbers of similar patches for the 3D arrays are 16 and 32 for *Step One* and *Step Two*, respectively. Ten gray-scale images are used during this experiment. The images are shown in Figure 2. The code is implemented in C++. The computer's processor is Intel® Core™ i7 (2.5 GHz). In Subsection 3.1 and Subsection 3.2, the methods are evaluated both quantitatively and qualitatively.



**Figure 2.** The ten images used in the experiment: (a) *Cameraman* image  $256 \times 256$ , (b) *House* image  $256 \times 256$ , (c) *Peppers* image  $256 \times 256$ , (d) *Lena* image  $512 \times 512$ , (e) *Barbara* image  $512 \times 512$ , (f) *Boats* image  $512 \times 512$ , (g) *Fingerprint* image  $512 \times 512$ , (h) *Man* image  $512 \times 512$ , (i) *Couple* image  $512 \times 512$ , and (j) *Hill* image  $512 \times 512$ .

#### 3.1. Quantitative evaluation

**3.1.1. Image similarity metric.** Peak Signal-to-Noise Ratio (PSNR) is used as a similarity metric to objectively assist the difference between the original and denoised images. The higher the PSNR values, the better the results are. The Peak Signal-to-Noise ratio is defined as:

$$PSNR = 10 \log_{10} \left( \frac{(2^n - 1)^2}{MSE} \right)$$

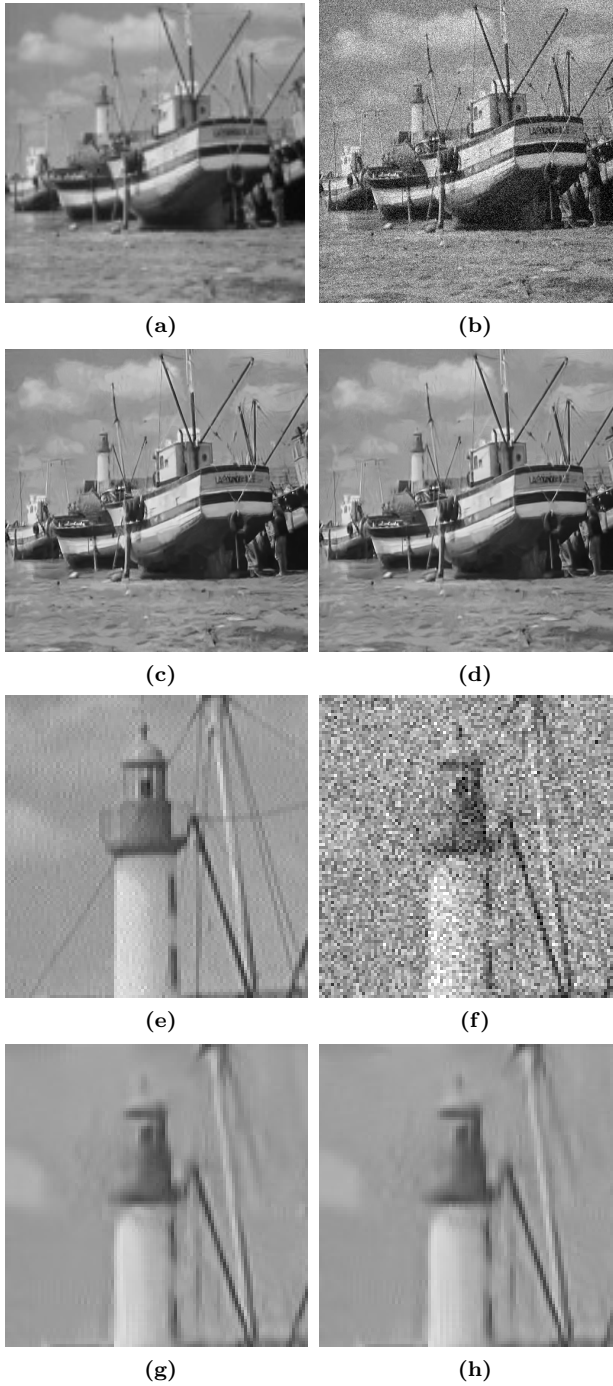
where  $MSE$  is the Mean Squared Error value, and  $n$  is an integer number representing the number of bits per pixel,  $n = 8$  in case of gray-scale images.

**3.1.2. Results.** The results of our proposed method are shown in Table 1, which compares the performance of our method with the original BM3D. Each noise level ( $\sigma$ ) has two rows: the first row shows the results of the original BM3D, and the second row shows the performance of our method. The higher PSNR values are highlighted with a bold font. The results are computed by measuring the differences between the noise-free images and the denoised ones.

**Table 1.** The performance of the denoising algorithms at various noise levels ( $\sigma$ ). At each noise level ( $\sigma$ ): the first row shows the results of using BM3D, and the second row shows the performance of our method.

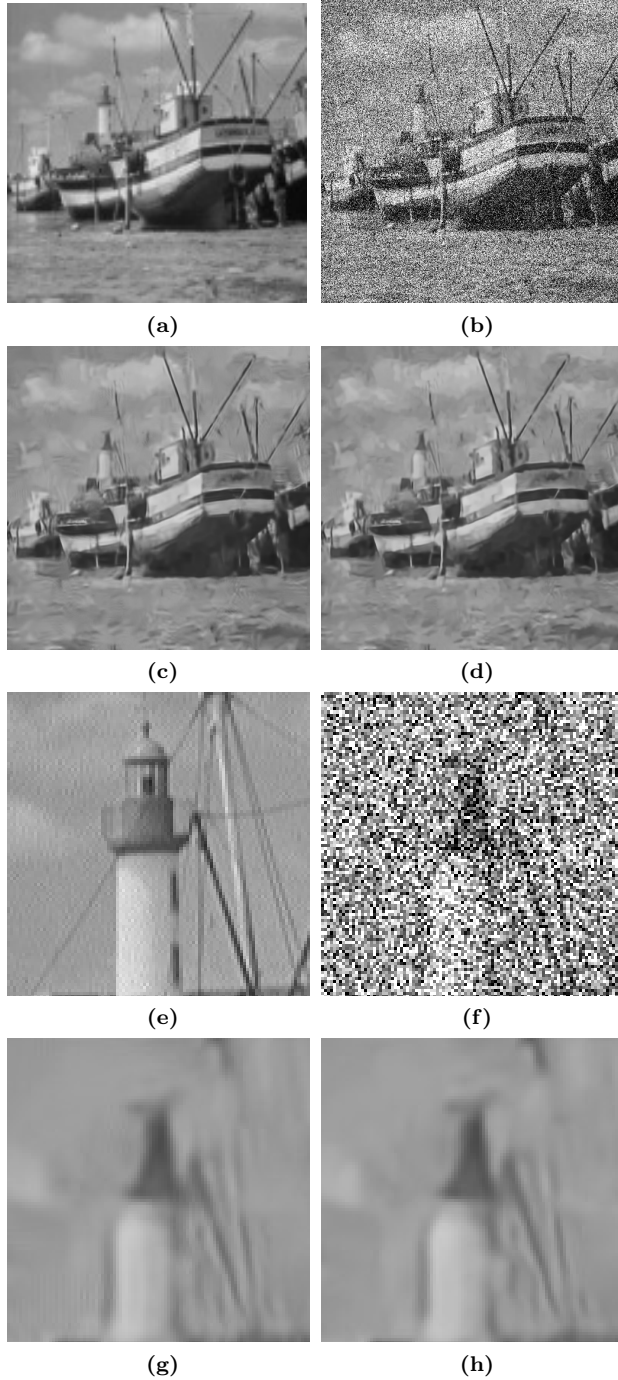
$\sigma$	C.man	House	Pep.	Lena	Barba.	Boats	F.print	Man	Couple	Hill	Average
10	33.99	36.48	34.52	35.87	34.78	33.84	32.40	33.93	33.93	<b>33.62</b>	34.33
	<b>34.07</b>	<b>36.52</b>	<b>34.61</b>	<b>35.94</b>	<b>34.90</b>	<b>33.89</b>	<b>32.55</b>	<b>33.95</b>	<b>33.97</b>	<b>33.62</b>	<b>34.40</b>
20	30.21	33.60	31.14	33.01	31.59	30.77	28.82	30.59	30.63	30.68	31.10
	<b>30.29</b>	<b>33.70</b>	<b>31.23</b>	<b>33.08</b>	<b>31.71</b>	<b>30.83</b>	<b>28.91</b>	<b>30.60</b>	<b>30.67</b>	<b>30.69</b>	<b>31.17</b>
30	28.22	31.96	29.10	31.24	29.60	28.93	26.84	28.87	28.68	29.03	29.25
	<b>28.25</b>	<b>32.04</b>	<b>29.18</b>	<b>31.30</b>	<b>29.70</b>	<b>29.00</b>	<b>26.91</b>	<b>28.89</b>	<b>28.73</b>	<b>29.06</b>	<b>29.31</b>
40	26.38	30.57	27.42	<b>29.93</b>	<b>28.14</b>	27.54	25.52	27.60	27.24	27.86	27.82
	<b>26.58</b>	<b>30.60</b>	<b>27.57</b>	29.92	28.12	<b>27.62</b>	<b>25.54</b>	<b>27.68</b>	<b>27.29</b>	<b>27.87</b>	<b>27.88</b>
50	25.10	<b>29.37</b>	26.08	<b>28.80</b>	<b>26.81</b>	26.42	24.40	26.62	26.13	26.85	26.66
	<b>25.34</b>	29.34	<b>26.23</b>	28.77	26.80	<b>26.50</b>	<b>24.44</b>	<b>26.70</b>	<b>26.16</b>	<b>26.88</b>	<b>26.71</b>
60	23.86	28.15	24.88	27.71	<b>25.58</b>	25.46	23.44	25.73	25.18	25.95	25.59
	<b>24.08</b>	<b>28.19</b>	<b>25.04</b>	<b>27.73</b>	<b>25.58</b>	<b>25.53</b>	<b>23.49</b>	<b>25.82</b>	<b>25.20</b>	<b>26.00</b>	<b>25.67</b>
70	22.79	27.01	23.78	26.67	<b>24.44</b>	24.59	22.56	24.87	24.32	25.09	24.61
	<b>23.00</b>	<b>27.06</b>	<b>23.95</b>	<b>26.73</b>	<b>24.44</b>	<b>24.68</b>	<b>22.63</b>	<b>24.98</b>	<b>24.37</b>	<b>25.16</b>	<b>24.70</b>
80	21.87	25.94	22.74	25.66	23.36	23.79	21.74	24.05	23.56	24.27	23.70
	<b>22.11</b>	<b>25.98</b>	<b>22.94</b>	<b>25.75</b>	<b>23.38</b>	<b>23.88</b>	<b>21.81</b>	<b>24.17</b>	<b>23.63</b>	<b>24.38</b>	<b>23.80</b>
90	21.12	24.93	21.81	24.70	22.35	23.04	20.96	23.26	22.88	23.47	22.85
	<b>21.33</b>	<b>24.95</b>	<b>22.03</b>	<b>24.81</b>	<b>22.41</b>	<b>23.14</b>	<b>21.05</b>	<b>23.40</b>	<b>22.95</b>	<b>23.61</b>	<b>22.97</b>
100	20.29	23.92	20.99	23.78	21.45	22.36	20.23	22.52	22.26	22.72	22.05
	<b>20.50</b>	<b>23.98</b>	<b>21.15</b>	<b>23.91</b>	<b>21.53</b>	<b>22.45</b>	<b>20.31</b>	<b>22.67</b>	<b>22.33</b>	<b>22.87</b>	<b>22.17</b>

Our method performs slightly better than the original BM3D at various noise levels. BM3D is the state-of-the-art denoising method, so we believe that any improvement would contribute to developing a better denoising method. BM3D has two denoising steps and more than twenty parameters, we improved just one parameter. Improving the twenty parameters is needed in order to achieve a bigger improvement.



**Figure 3.** The results of denoising the *boat* image at noise levels ( $\sigma = 30$ ): (a) noise-free *boat* image  $512 \times 512$ , (b) AWGN image, ( $\sigma = 30$ ), (c) BM3D, (d) Our method, and (e), (f), (g) and (h) fragments images of the first row.





**Figure 4.** The results of denoising the *boat* image at noise levels ( $\sigma = 80$ ): (a) noise-free *boat* image  $512 \times 512$ , (b) AWGN image, ( $\sigma = 80$ ), (c) BM3D, (d) Our method, and (e), (f), (g) and (h) fragments images of the first row.

### 3.2. Qualitative evaluation

The evaluation in this section is subjective, where the quality of the denoised images is addressed via the visual perception. Additive White Gaussian Noisy (AWGN) images with the noise levels ( $\sigma = 30, 80$ ) are chosen to perform this evaluation. Fragments of the noisy gray-scale boat images and the corresponding estimates are shown in Figure 3 and Figure 4.

The fragment images in Figure 3h and Figure 4h show that our method outperforms the BM3D method. Our method preserves sharp edges; e.g., the mast and ropes of the boat image in Figure 3h. Homogeneous regions are smoothed properly by our method; i.e., lighthouse in the boat image in Figure 4h.

## 4. CONCLUSION

In this work, we have explored the problem of gray-scale image denoising process. We restored the noisy images by using BM3D approach with adaptive thresholding and weighted Wiener filtering. The results show that our method achieved better denoising than the original BM3D at various noise level.

## REFERENCES

- [1] M. Aharon, M. Elad and A Bruckstein, *K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation*, IEEE Trans. Signal Process. **54** (2006), 4311–4322.
- [2] A. Buades, B. Coll and J. Morel, *A non-local algorithm for image denoising*, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) **2**, 2005, 60–65.
- [3] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, *Image denoising by sparse 3-D transform-domain collaborative filtering*, IEEE Trans. Image Process. **16** (2007), 2080–2095.
- [4] C. Deledalle, J. Salmon and A. Dalalyan, *Image denoising with patch based PCA: Local versus global*, in: Jesse Hoey, Stephen McKenna and Emanuele Trucco, Proceedings of the British Machine Vision Conference, BMVA Press, 2011, 25.1–25.10.
- [5] D. L. Donoho and I. M. Johnstone, *Ideal spatial adaptation by wavelet shrinkage*, Biometrika **81** (1994), 425–455.
- [6] A. Gersho, *On the structure of vector quantizers*, IEEE Trans. Inf. Theory **28** (1982), 157–166.
- [7] A. K. Jain, M. N. Murty and P. J. Flynn, *Data clustering: A review*, ACM Comput. Surv. **31** (1999), 264–323.
- [8] T. Kohonen, M. R. Schroeder and T. S. Huang, *Self-Organizing Maps*, 3rd ed., Springer-Verlag, New York, Secaucus, NJ, USA, 2001.
- [9] J. MacQueen, *Some methods for classification and analysis of multivariate observations*, Proc. Fifth Berkeley Symp. on Math. Statist. Prob. **1** (1967), 281–297.
- [10] O. Rioul and M. Vetterli, *Wavelets and signal processing*, IEEE Signal Process. Mag. **8** (1991), 14–38.
- [11] C. Tomasi and R. Manduchi, *Bilateral filtering for gray and color images*, Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), 1998, 839–846.

Monagi H. Alkinani, Computer Science Department, University of Western Ontario, London, Ontario, Canada, N6A 5B7

*e-mail:* [malkinan@uwo.ca](mailto:malkinan@uwo.ca)

Mahmoud R. El-Sakka, Computer Science Department, University of Western Ontario, London, Ontario, Canada, N6A 5B7

*e-mail:* [melsakka@uwo.ca](mailto:melsakka@uwo.ca)

