Hybrid Filter for Gaussian Noise Removal with Edge Preservation

Ali S. Awad
Faculty of Engineering and Information Technology
AL-Azhar University-Gaza,Palestine
aawad@alumni.stevens.edu

ABSTRACT
This paper proposes a new algorithm to remove Gaussian noise. The new method introduces two filters. The first one is linear filter that modifies the noisy and noisy-free pixels uniformly and regardless of the pixel location. The second one is non-linear filter, a direction-based filter used to re-estimate the first output, particularly the values of the edge pixels. Simulation results indicate that the proposed method restores images corrupted at different degrees of Gaussian noise and demonstrates the best performance compared to other methods, particularly for highly corrupted images in terms of PSNR or visual quality.

Keywords
De-noising, Gaussian noise, triangular filter, non-linear filter

1. INTRODUCTION
Images are degraded by Gaussian noise for a variety of reasons, including electronic circuit noise and sensor noise of a scanner or digital camera. Unlike impulse noise, Gaussian noise affects every pixel in the image. Therefore, it is difficult to accurately identify the noisy pixel. Using specific technique or cascade of several filters is rather efficient in Gaussian noise removal. Gaussian noise removal approach is based on linear and non-linear filters. Linear filters are arithmetic mean filter, Gaussian filter, and Wiener filter. The major drawback of linear filters is a uniform modification of every pixel in the image regardless of the location of the tested pixel. In other words, the application of linear filters may lead to the loss of fine details and produce blurry edges. As a result, there is a need for a more efficient filter that deals non-linearly with the corrupted image pixels. The secondary literature proposes diverse non-linear filters for Gaussian noise removal, namely anisotropic diffusion algorithms [1]-[3]. Even though anisotropic diffusion algorithms deliver positive results, they still cause a loss in the edge information. In addition, anisotropic diffusion algorithms represent a parametric method: several parameters such as number of iteration and standard deviation must be justified repeatedly based on the noise intensity. Finally, anisotropic diffusion algorithms commonly lead to a stair- case effect around smooth edges [4]-[5]. Another type of non-linear filters is alpha-trimmed mean filters [6]-[9] that eliminate the noise of extreme values. Alpha-trimmed mean filters are an effective compromise between the median filters and moving average filters. Thus, they are the best fit to the Gaussian noise removal.

Furthermore, a number of techniques, such as Wavelet transform [10], Curvelet [11], and contourlet transforms [12] are also applied in the removal Gaussian noise. The standard bilateral filtering was first proposed by Tomasi and Manduchi in 1998 [13]. Significant efforts have been devoted to improve the classical bilateral filtering algorithm, as shown in [14]-[17]. Garnett et al. [18], for instance, developed a trilateral filter to remove both impulse and Gaussian noise. Nevertheless, all of the above mentioned filters tend to show an overall poor performance when the noise in the image is relatively high. This paper introduces an efficient method consisting of two cascade filters to remove low to high density of Gaussian noise. Due to a correlation between the neighboring pixels, the first filter is used to estimate each pixel from its neighboring pixels based on a triangular function. In the second filter, a restoration process is performed specifically to re-estimate the edge pixels. Compared to other existing linear and non-linear methods, the proposed technique has effective performance at low density noise, where the standard deviation is relatively small and superior at highly corrupted images. Section 2 of this paper describes the first and the second filters of the proposed method. Sections 3 and 4 discuss simulation results and present conclusion, respectively.

2. ALGORITHM DESCRIPTION
Pixels in original images are splitted into two regions: (1) similar pixels in smoothing areas and (2) edge pixels in abrupt areas. Edge pixels tend to have higher or lower values than other pixels. In addition, edge pixels are small in number compared to similar pixels. There is a specific correlation among the pixels in both regions, particularly between the neighboring pixels. Most of the Gaussian noise values are distributed around the mean and other have values significantly higher than the mean. Thus, cascade of filters is necessary to tackle different values of the Gaussian noise. The proposed algorithm integrates two filters based on the structure and the characteristics of the image pixels, as mentioned before. The first filter replaces the image pixels by the weighted average of their surrounding pixels. The purpose of this filter is to estimate the original values for all noisy pixels. One of the drawbacks of the first filter is the resulting linear modification of all image pixels, regardless of the location of every pixel. Due to this drawback, the second filter is proposed to replace pixel by the median of the pixels in the optimum direction. Optimum direction crosses the tested pixel and has the most similar pixels. The second filter allows enhancing the edge pixels that are usually aligned in lines of edges.

2.1 Triangular Filter
Considering the correlation between the neighboring pixels in the image, it is possible to estimate the tested pixel from its neighbors in a $n \times n$ window. Furthermore, as the distance between the central pixel(tested pixel) and any other pixel in the window is increasing, the similarity between pixels is decreasing, particularly in large windows. The estimation can

be easily performed by using a triangular filter. In triangular filter, the estimated pixel $\hat{p}(i,j)$ is calculated as:

$$\hat{P}(i,j)=\sum_{k}\sum_{m}w(k,m)P(i+k,j+m) \qquad (1)$$

where ***w(k,m)*** represents the space weights of the pixels around the tested pixel ***P(i,j),*** while ***k*** and ***m*** are bounded as: $\{-(n-1)/2\le k\,\&\,m\le(n-1)/2\},\{k,m\}\neq\{0,0\}$.

Based on the inverse relationship between the similarity and the distance from the origin, the following triangular function ***f (k,m)*** is used to calculate the space weight ***w(k,m)*** of each pixel in the window:

$$f(k,m)=a-d(k,m) \qquad (2)$$

$$d(k,m)=\sqrt{k^{2}+m^{2}} \qquad (3)$$

$$w(k,m)=\frac{f(k,m)}{\sum_{k}\sum_{m}f(k,m)} \qquad (4)$$

d(i,j) is the Euclidean distance between the location ***(0,0)*** of the central pixel and the indices ***(k,m)*** in the window. The value of ***a*** should be positive and greater than the largest Euclidean distance:

$$a>Max\{d_{k,m}\} \qquad (5)$$

The output results are consistent for any value of ***a*** > ***max***($d_{k,m}$). For example, if ***a***=***3***, then the deduced space weights around the tested pixel in ***3x3*** window and based on Eq.4 are shown in Figure 1.

w=1/14.4 ×

m =	-1	0	1	***k*** =
	1.6	2	1.6	-1
	2		2	0
	1.6	2	1.6	1

Fig 1: Weights of the different locations in 3×3 window around the central pixel

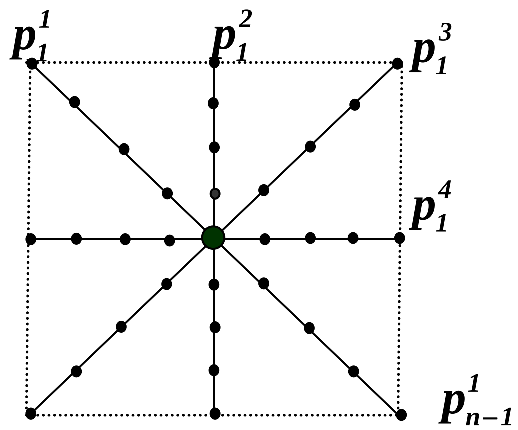
Therefore, the estimated pixel $\hat{p}(i,j)$ is defined as:

$$\hat{p}(i,j)=\frac{\sum_{k}\sum_{m}[\{a-d(k,m)\}P(i+k,j+m)]}{\sum_{k}\sum_{m}\{a-d(k,m)\}} \qquad (6)$$

If noisy pixels of high or low values are compared to their surrounding pixels, then the surrounding pixels will be estimated inaccurately. The Triangle filter is a low pass filter that suppresses the fine details and smoothens the edge pixels and also it is easy to construct and has simple equation as depicted in (2). In some instances, the difference between the original and the estimated value is rather high, particularly for the edge pixels. To tackle this problem and to restore the edge values, a new estimation of pixels should be performed with a non-linear filter. The new filtering process is aimed on finding the optimum direction that has the most similar pixels.

2.2 Direction-based filter

In this section, we describe the process of restoring the degraded edge pixels. To achieve this goal, the pixels around the central pixel in ***nxn*** window are divided into four directions $d^{i'}$ where ***i'=1:4***, as shown below. Central pixel is excluded from each direction. Thus, each direction has ***n-1*** pixels. The direction with the largest similarity factor ***SF*** is used to re-estimate the tested pixel $\hat{p}(i,j)$ as:



$$d^{i'}=\{p_{1}^{i'},p_{2}^{i'},p_{3}^{i'},p_{4}^{i'},....p_{n-1}^{i'}\} \qquad (7)$$

where pixel $p^{i}_{(n+1)/2}$ is excluded . The similarity factor is defined as:

$$SF_{j'}^{i'}=1-\frac{\delta_{j'}^{i'}}{L_{\max}}, \qquad \text{where} \qquad j'=1:n-2$$

(8) L_{max} is the maximum value in the image and $\delta_{j'}^{i'}$ defined as

$$\delta_{j'}^{i'}=|\,p_{j'}^{i'}-p_{j'+1}^{i'}\,| \qquad (9)$$

a new vector $\delta^{i'}$ for each direction is defined as:

$$\delta^{i'}=\{\delta_{1}^{i'},\delta_{2}^{i'},\delta_{3}^{i'},\delta_{4}^{i'},....\delta_{n-2}^{i'}\}$$

(10) If the summation of the vector $\delta_{j'}^{i'}$ is defined as $\zeta^{i'}=\sum_{j'}\delta_{j'}^{i'}$ then the similarity factor of the direction $d^{i'}$ is calculated in the following way:

$$SF^{i'}=\sum_{j'=1}^{n-2}\left(1-\frac{\delta_{j'}^{i'}}{L_{\max}}\right)=\left(n+\frac{\zeta^{i'}}{L_{\max}}\right)-2 \qquad (11)$$

To identify the direction with the maximum similarity factor, it is necessary to search over all of the pre-determined directions:

$$d^{op} = \underset{i'}{argmax}\{SF^{i'} \textbf{ where } i' = 1:4\} \quad (12)$$

where d^{op} is the optimum direction used for re-estimating pixel $\hat{p}(i,j)$, as described below:

$$\hat{p}(i,j) = Median\ \{d^{op}\} \quad (13)$$

Figure 2 shows the results after the application of the first and second filters on the pixels 48, 92, 130, 84, 127, and 154. After the first filter was applied, the attained results for the edge values 130, 154, and 127 were not accurate (see part b) However, after the use of second filter, the edge pixels became closer to the original ones (see part c). For both filters, we slided 3×3 window over each pixel in the image with L_{max}=255.

31	35	59	87	149
27	**48**	**92**	**130**	164
46	**84**	**127**	**154**	151
73	117	149	160	137

(a)

31	35	59	87	149
27	**62**	**92**	**123**	164
46	**86**	**118**	**137**	151
73	117	149	160	137

(b)

31	35	59	87	149
27	**52**	**86**	**133**	164
46	**79**	**125**	**156**	151
73	117	149	160	137

(C)

Fig. 2.Implementing the proposed filter on six values in Lena image: (a) part of original values in Lena image, (b) the results after implementing the first filter, (c) the results after implementing the second filter.

3. SIMULATION RESULTS

Different images are tested during the simulation experiments. Original images were corrupted artificially by addition of Gaussian noise of zero mean, $m=0$, and variance σ^2. Results of all the experiments are obtained by the end of the first iteration. ***3×3*** window and $a=3$ are used in all the simulation experiments.

The performance of the proposed method is compared with well-known methods, namely Perona and Malik, bilateral method, trilateral, and alpha-trimmed mean filters. The comparison is done subjectively in terms of the visual image quality and objectively in terms of the ***PSNR***. One of the advantages of the newly proposed method is its non-parametric approach. In particular, there is no need to justify any parameter at different levels of the noise and even the number of iterations. Table 1 reveals that the proposed method outperforms other approaches in restoring corrupted version of the Lena image at $\sigma \geq 30$ (Figure 3). The positive results are achieved due to the second stage of the proposed method, in which an enhancing to the edge values is carried out. For example, an edge pixel in the Lena image has an original value of ***79*** and is corrupted to ***107***. After the application first filter, its value becomes ***91.3*** and after the second filter application its value is further reduced to ***81.3***. Similarly, the original value of another edge pixel is ***94***, and becomes ***114*** and ***109*** after the first and second filters, respectively. The improvement is occurred thanks to the second estimation, where every pixel is replaced by a new value obtained from the optimal direction that include the most similar pixels.

Table 1: Comparison for LENA image in terms of PSNR (dB)

Method	PSNR(dB)- Lena image					
	σ =10	σ =20	σ =30	σ =40	σ =50	σ =60
Bilateral[13]	33.30	30.02	26.36	22.72	19.59	17.12
Trilateral[18]	29.73	28.74	27.32	25.71	23.99	22.36
Malik [1]	31.7	29.13	27.43	25.92	25.69	24.9
α-mean[6]	32.43	29.97	27.75	25.85	24.24	22.89
NEW	31.33	29.27	28.19	27.18	26.16	25.25

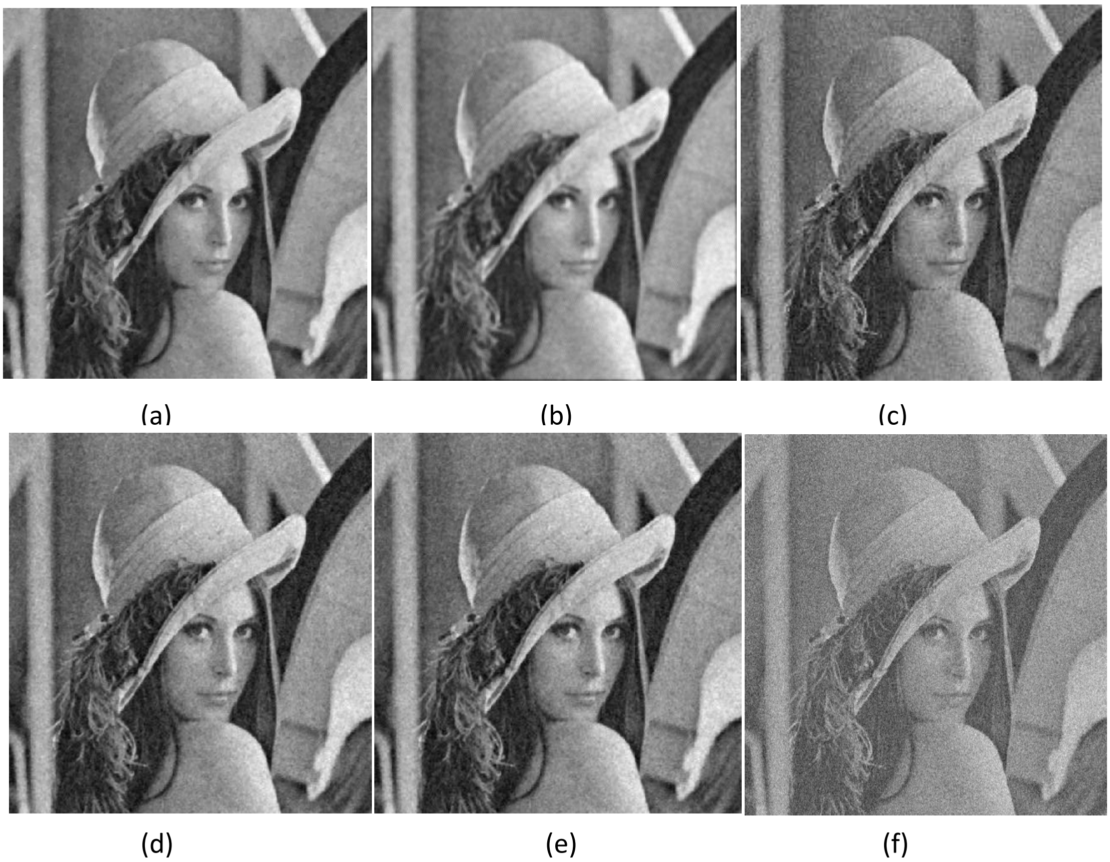


Fig.3. Comparison between existing methods and the proposed one for restoring corrupted Lena image at σ=30:(a) Proposed, (b) Malik[1],(c) bilateral[13],(d) alpha-mean[6],(e) trilateral[18],(f) corrupted image

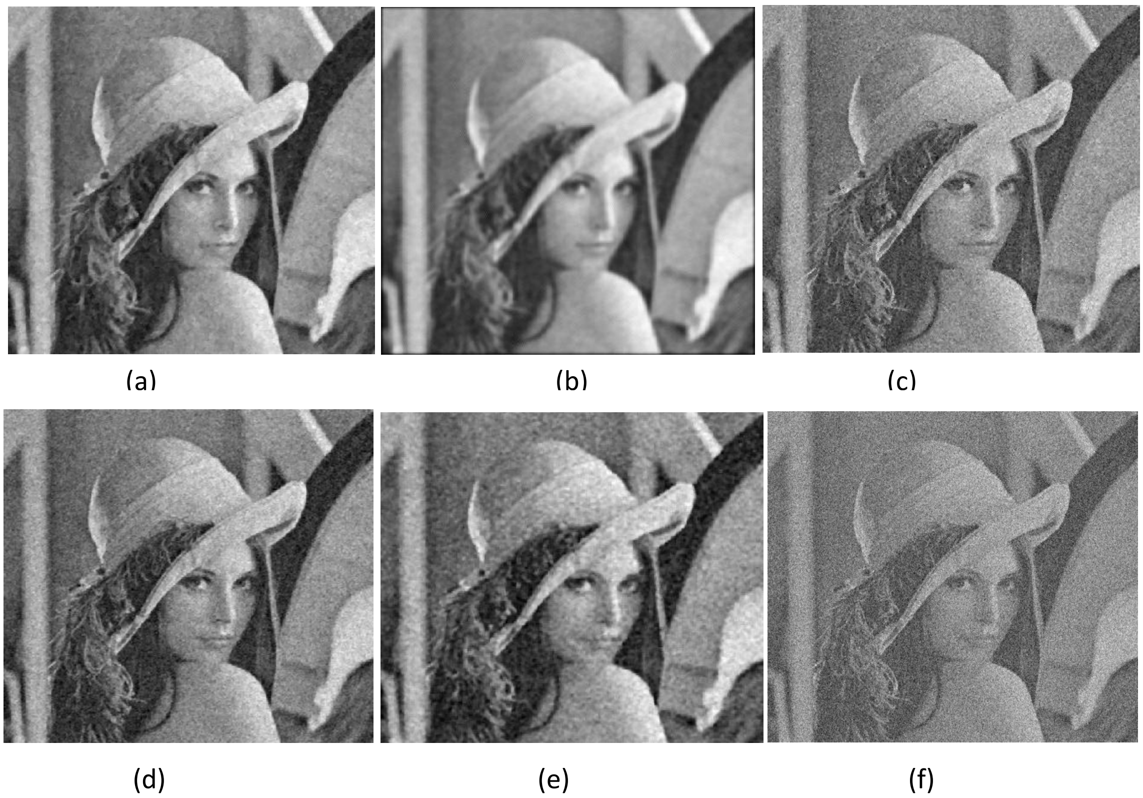


Fig.4. Comparison between existing methods and the proposed one for restoring corrupted Lena image at σ=40:(a) Proposed, (b) Malik[1],(c) bilateral[13],(d) alpha-mean[6],(e) trilateral[18],(f) corrupted image

Figure 3 shows the results of applying different filters in restoring Lena image corrupted with Gaussian noise at standard deviation of ***σ=30***. It is obvious that the proposed method preserves the fine details of the image, as indicated in the lines of the hair and hat in particular. In addition, it is apparent that the anisotropic method results in blurred edge lines. Figure 4 illustrates the restoration performance of different methods in removing Gaussian noise of ***σ = 40*** in Lena image. Similarly, the proposed method generates the best results since it preserves the image details, while other methods produce blurry and noisy images. Table 2 demonstrates that the proposed method delivers comparable results for pepper image at low standard deviation values and shows the best values of PSNR at ***σ ≥30***. Figure 5 presents a new experiment at ***σ=50***, in which the visual quality of the restored pepper image is the best compared to the other methods. Figure 6 shows that the restored boat image of the proposed method has the best visual quality and less blur even at ***σ=50***. This result is achieved thanks to the enhancement of the edge pixels during the second stage. Table 3 demonstrates the outcomes of applying different filters in terms of ***PSNR*** obtained in the result of restoration of a corrupted version of boat image which has many fine details. The proposed method outperforms others filters in terms of ***PSNR***, when the standard deviation ***σ ≥50***.

Table 2: **Comparison for PEPPER IMAGE in terms of PSNR (dB)**

Method	**PSNR(dB)- pepper image**					
	σ =10	σ =20	σ =30	σ =40	σ =50	σ =60
Bilateral[13]	33.97	30.27	26.42	22.75	19.64	17.08
Trilateral[18]	30.73	29.50	27.91	26.08	24.27	22.52
Malik [1]	32.44	29.70	27.92	26.13	25.87	24.95
α-mean[6]	33.10	30.34	28	25.98	24.31	22.89
NEW	31.09	30.06	28.86	27.75	26.60	25.46

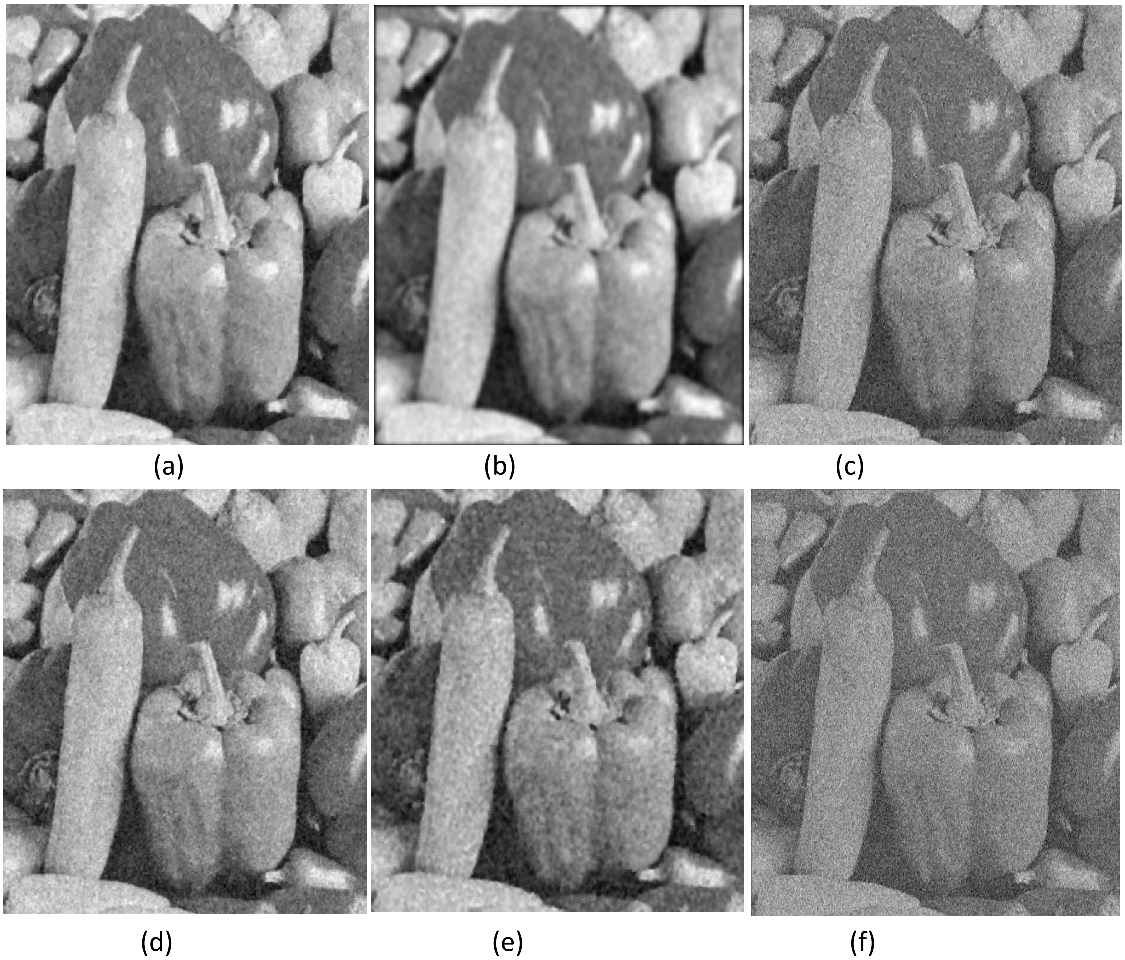


Fig.5. Comparison between existing methods and the proposed one for restoring corrupted Pepper image at σ=50:(a) Proposed, (b) Malik[1],(c) bilateral[13],(d) alpha-mean[6],(e) trilateral[18],(f) corrupted image

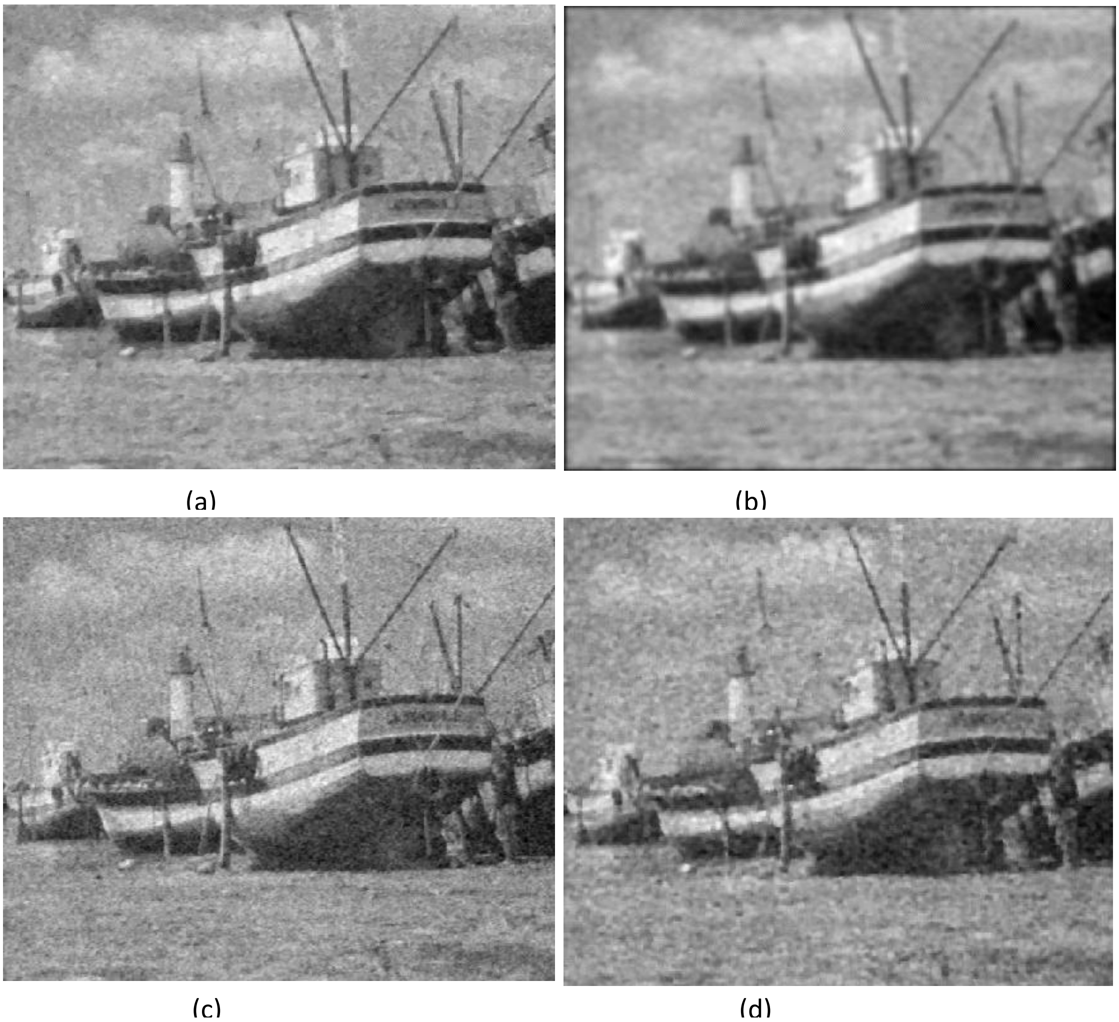


Fig.6. Comparison between existing methods and the proposed one for restoring corrupted Boat image at σ=50 :(a) Proposed, (b)Malik[1],(c) alpha-mean[6],(d)trilateral[18].

4. CONCLUSION

In the proposed paper, the weighted average filter based on a triangular function was applied to estimate the noisy pixels. The second filter was used to re-estimate the formerly restored pixels with the purpose to improve the visual quality of the corrupted image. In the second filter, each pixel was replaced by a new value calculated from the most similar values included in the optimal direction. Simulation experiments demonstrate the potential of the proposed hyper-filter to exhibit superior results, particularly for highly corrupted images.

5. REFERENCES

[1] P. Perona and J. Malik, "Scale -space and edge detection using anisotropic diffusion," IEEE Trans. Pattern Anal. Machine Intell., Vol. 12, pp. 629–639, 1990

[2] M.J. Black , Guillermo Sapiro, David Marimont, and David Heeger, 'Robust Anisotropic Diffusion', IEEE Trans. on Image Processing, Vol. 7, No.3, pp. 4 21-432,1998

[3] M.J. Black , D. Fleet D., and Y. Yacoob , 'Robustly estimating changes in image appearance', Computer Vision and Image Understand, Vol. 78, pp. 8 -31 2000

[4] J. Tumblin and G. Turk, "LCIS: A Boundary Hierarchy for Detail-Preserving Contrast Reduction," in Proc. ACM SIGGRAPH 99.pp. 83-90, 1999..

[5] Y. You, W. Xu, A. Tannenbaum, and M. Kaveh, "Behavioral Analysis of Anisotropic Diffusion in Image Processing," IEEE Tran. Image Processing. Vol.5,No.11, pp.1539-1553 ,1996.

[6] J. B. Bednar and T. L. Watt, "Alpha-trimmed means and their relationship to median filters," IEEE Trans. on Acous.,Speech and Signal Processing, vol. 32, pp. 145–153, 1987

[7] Y. B. Rytsar and I. B. Ivanesko, "Application of (alpha, beta)-trimmed mean filtering for removal of additive noise from images," Proc. of SPIE, vol. 3238, 1997.

[8] S. R. Peterson, Y. H. Lee, and S. A. Kassam, "Some statistical properties of alpha-trimmed mean and standard type M filters," IEEE Trans. On Signal Proc., vol. 36, no. 5, pp. 707–713, 1988.

[9] A. Taguchi, "An adaptive _ -trimmed mean filter with excellent detail preservation and evaluation of its performance," Elec. Comm. in Japan Part III, vol. 78, no. 10, pp. 46–56, 1995

[10] L. Kaur, S. Gupta and R.C. Chauha., "Image Denoising using Wavelet Thresholding , ICVGIP, 2002.

[11] E. J. Candes, and D. L. Donoho,(1999) "Curvelets—A surprisingly effective nonadaptive representation for objects with edges," in Curves and Surfaces Fitting, L. L. Schumaker, Ed. et al. Nashville, TN: Vanderbilt Univ. Press**.**

[12] R. Eslami and H. Radha (2003),"The Contourlet Transform for Image De-noising Using Cycle Spinning", IEEE Trans. Image Processing pp. 1982-1986.

[13] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," IEEE Int. Conf. Computer Vision, pp. 839- 846, 1998.

[14] S. Morillas, V. Gregori, and A. Sapena. Fuzzy bilateral filtering for color images. Lecture Notes in Computer Science, pages 138–145, 2006.

[15] K. J. Overton and T. E. Weymouth. A noise reducing preprocessing algorithm. IEEE Proc . Comput. Sci. Conf. Pattern Recoginition and Image Processing, pp. 498–507, 1979.

[16] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand. A gentle introduction to bilateral filtering and its applications. SIGGRAPH, presentation, 2007.

[17] B. Zhang and J. P. Allebach. Adaptive bilateral filter for sharpness enhancement and noise removal. IEEE Trans. Image Processing, Vol.17,No.5, pp.664–678, 2008.

[18] R. Garnett , Timothy Huegerich and Charles Chui, 'A Universal Noise Removal Algorithm with an Impulse Detector' IEEE Trans. on Image Processing, Vol. 14, No.11, pp.1747-1754, 2005.

Table 3: Comparison for BOAT image in terms of PSNR (dB)

Method	PSNR(dB)- boat image					
	σ =10	σ =20	σ =30	σ =40	σ =50	σ =60
Bilateral[13]	30.79	28.68	25.71	22.47	19.47	17.02
Trilateral[18]	29.73	28.73	27.34	25.63	23.99	22.31
Malik [1]	30.33	26.66	25.02	23.65	23.55	22.93
α-mean[6]	29.06	27.72	26.22	24.81	23.45	22.21
NEW	26.59/28	26.14/26.9	25.63	25.05	24.32	23.65