ISSN 2278-3091
Volume 9 No.2, March - April 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse212922020.pdf
https://doi.org/10.30534/ijatcse/2020/212922020

A New Faster, Better Pixels Weighted Don't Care Filter for Image Denoising and Deblurring

Rachana Dhannawat[1], Dr. Archana B. Patankar[2]
[1]Research Scholar, University of Mumbai, India, rachanadhannawat82@gmail.com
[2]Professor, Thadomal Shahani Engineering College, University of Mumbai, India, Archana.patankar@thadomal.org

ABSTRACT

This paper proposes a novel filter, which assigns weight selectively and considers only 4 neighbors for calculation. This filter has reduced complexity approximately by 50 % to that of averaging filter and median filter. The proposed technique has improved PSNR by 13% and SSIM by 40% as compared to noisy images. The execution time is reduced by 70% as compared to averaging technique and 93% as compared to median filter. This filter is used for image deblurring as well and the results are improved in terms of PSNR and SSIM by 11% and 1 % respectively. As this filter has improved results for denoising as well as deblurring, it is called as a dual purpose filter. The filter is tested for both gray and colour images and improves results for both.

Key words: Don't care filter, diamond, denoising, deblurring, plus, 4 neighbours.

1. INTRODUCTION

Image degradation is an unavoidable process. The image can get degraded even while capturing the image. Degradation can be due to sensor noise, camera misfocus, object or camera motion, atmospheric conditions, vibrations of atoms in receiver devices etc [1]. Image denoising [2] and deblurring [3] are very important research areas in image processing. Many filters are developed for image denoising [4].

Filters used in the literature are averaging, median [5], high boost, Wiener filter [6], Inverse filter, etc. The best results are obtained by nonlocal means filters (NLM) [7]. Many state of art advanced algorithms such as KSVD [8] K-means clustering with Singular Value Decomposition, clustering based dictionaries with locally learned dictionaries KLLD [9], Clustering based Sparse Representation CSR [10], Local Pixel grouping- Principal Components analysis LPG-PCA [11], Nonlocally Centralized Sparse Representation NCSR [12], Block Matching and 3D filtering BM3D [13] are also developed for image denoising. NCSR [12], Iterative Shrinkage Thresholding algorithm (ISTA) [14], Fast Iterative Shrinkage Thresholding algorithm (FISTA) [15] [16], Two Step Iterative Shrinkage Thresholding algorithm (TwIST) [17], Total Variation TV- based [18] iterative algorithms are used frequently for image deblurring. These all algorithms give good results for image denoising however they are iterative, slow as well as complex as compared to basic filters. This paper proposes a new filter which considers only 4 adjacent neighbors of the central pixel in the calculation and assigns more weight to the central pixel. Remaining 4 neighbors are not part of computation hence obviously the filter is fast and less complex. The main objective of this filter is to denoise and deblur the image with minimum complexity and still maintain the quality of the restored image. A basic filter and its different variations are discussed in this paper. Results are compared in terms of PSNR and SSIM [19] and it is observed that PSNR is improved by 13% as compared to noisy image and SSIM is improved by 40% as compared to noisy image. Complexity is reduced nearby about 50% as compared to any basic filters [20]. This filter gives better results for both denoising as well as deblurring.

2. PROPOSED FILTER

In most of the filters, equal importance or weight is given to all the pixels under consideration. For example, Averaging filter [4] takes an average of all pixels in the neighborhood. As a consequence performance of the filter degrades as the redundant pixel values are part of the computation and calculations are also more as all pixels are involved. Basic concept of averaging filter is that as it takes the average of neighbouring pixels so the image gets blurred and the effect of noise is reduced. Problem with this filter is that if any one value is out of the range then average will distribute its effect everywhere in the neighbourhood.

In this new filter, this problem is solved by selecting only some important pixels and assigning a higher weight to the central pixel. Pixel values selected are those which majorly affect the central pixel. Rests of the neighboring pixels are ignored. Therefore, the number of calculations obviously decreases. Because of selecting the most important responsible pixel values, accuracy increases. This higher weight is denoted as 'a' and is assigned the value as 1 percent of standard deviation of noise. If a=0, the filter is equivalent to an averaging filter with only 5 pixels involved in the calculation. The new filter is as given below.

x	1/5	x
1/5	(1+a)/5	1/5
x	1/5	x

Figure 1: The new filter with dimension 3*3

Figure 1 gives a 3 *3 new filter. Here, 'a' is a weighing factor and x denotes don't care. In this filter, we consider only 5 central pixels for calculations and ignore other pixels as don't care. Extending this further, higher dimension masks can be developed in two ways: plus and diamond as shown in Figure 2 for dimension 5.

x	x	1/9	x	x
x	x	1/9	x	x
1/9	1/9	(1+a)/9	1/9	1/9
x	x	1/9	x	x
x	x	1/9	x	x

a)Plus

x	x	1/13	x	x
x	1/13	1/13	1/13	x
1/13	1/13	(1+a)/13	1/13	1/13
x	1/13	1/13	1/13	x
x	x	1/13	x	x

b)Diamond

Figure 2: Extensions of filter for size 5*5

Diamond is an improved variation of technique 1 (plus) which has values as shown in Figure 6.2 (b). It considers immediate 8 neighbors of a central pixel along with horizontal and vertical pixels.

The odd size filters, being symmetric in nature are considered for further processing such as 3*3, 5*5, 7*7, 9*9,......., etc. The generalized plus filter shown in Figure 3.

x	x	x	1/(2n-1) till n/2 . . .	x	x	x
x	x	x	1/(2n-1)	x	x	x
x	x	x	1/(2n-1)	x	x	x
1/(2n-1) …till n/2	1/(2n-1)	1/(2n-1)	(1+a)/(2n-1)	1/(2n-1)	1/(2n-1)	… till n/2 1/(2n -1)
x	x	x	1/(2n-1)	x	x	x
x	x	x	1/(2n-1)	x	x	x
x	x	x	. . . till n/2 1/(2n-1)	x	x	x

Figure 3: Generalized plus filter

Generalized plus filter is developed based on equations as explained here. If n is the size of the filter then ⌊ (n/2) ⌋ pixels from the central pixel are selected in left, right, top and bottom directions. The total number of pixels will be 4* ⌊ (n/2) ⌋ + 1.This calculation 4* ⌊ (n/2) ⌋ + 1 is equal to (2*n)-1. Therefore, all pixels are divided by (2*n)-1 for the average purpose. E.g. if filter size is entered as 5, then ⌊ n/2 ⌋ = 2, so 2 pixels each from all 4 directions and 1 central pixel i.e. 9 pixels = (2*5-1) are selected. Similarly, if n=7, ⌊ n/2 ⌋ = 3, then (2*7-1) = 13 pixels are selected.

For diamond shape filter the generalized equation is written as: A total number of pixels = ⌈ n2/2 ⌉. For example consider n = 5 then total pixels = ⌈ 25/2⌉ =13, for n =7, ⌈ 49/2⌉ =25, for n = 9, total pixels = ⌈ 81/2⌉ = 41.

This value obtained for a total number of pixels is the factor, which divides all pixels considered as coefficients of the filter.

3. COMPLEXITY ANALYSIS

In existing filters such as averaging, highboost the number of multiplications required is n^2 and number of additions required is (n^2-1) for a pixel. Therefore, if the size of the image is x*y then $x*y*n^2$ additions and multiplications are required. Complexity is O (n^2). Median filter has an additional step of sorting the values so its complexity is O (n^4). Proposed technique requires only (2n-2) additions and (2n-1) multiplications for a single pixel. A total number of additions and multiplications required for an image are x*y*(2n-1) and total complexity is O (n) for plus extension. The calculations are reduced by a factor of (n*(n-2) +1). This value is near about 50 % for 3*3 filter and always greater than 50% for higher dimensions. As we increase the dimension, the value goes on increasing. Therefore, complexity reduces by more than 50% in this filter.

For the diamond type of extension, the complexity is reduced by a factor of $\lfloor n^2/2 \rfloor$. This means these many multiplications and additions are reduced as compared to n^2. Hence, the total pixels in the calculation are n^2 - ($\lfloor n^2/2 \rfloor$). This comes near about 50% pixels on an average. For example, for n=9, out of 81 only 41 pixels chosen, for n=7, out of 49 only 25 chosen, for 5 out of 25, 13 are chosen and so on.

4. RESULTS AND ANALYSIS

Figure 4 gives resultant denoised images of butterfly image by all filters discussed for denoising.

Table 1 displays result in terms of PSNR and SSIM [19] of all filters of size 3*3 for 9 gray images for image denoising. It is observed in Table 1 that for size 3*3 PSNR is best by proposed technique and SSIM is best for the average filter. The PSNR of the proposed diamond filter is 7% more than the average filter, 8 % more than median filter [21] and 39% more than high boost filter. SSIM of the proposed technique is 3% less than average filter, 150% more than high boost filter, and 3 % more than median filter. The complexity is reduced by

more than 50% as compared to other filters as observed from the complexity table.
For dimension 5*5 and higher, there is the difference in plus and diamond proposed techniques. Therefore, in Table 6.2 two separate columns for proposed techniques are presented and it is observed from Table 1 that high boost filtering is not improving the results, so omitted from further discussion. Table 2 displays PSNR and SSIM for 5*5 filters for gray images.

Table 1: Results in terms of PSNR and SSIM [19] of all filters of size 3*3 for 9 gray images for image denoising

Technique	**Noisy**		**Average**		**High Boost**		**Median [5]**		**Proposed (plus & diamond**	
Images\ Metric	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
House	29.308	0.346	33.066	**0.681**	24.363	0.175	32.276	0.618	**35.574**	0.608
Butterfly	29.283	0.515	32.351	**0.796**	24.615	0.273	31.675	0.759	**34.043**	0.740
Lena	29.307	0.434	32.736	**0.752**	24.480	0.221	32.043	0.702	**34.663**	0.682
Barbara	29.307	0.518	31.565	0.713	24.492	0.286	31.028	0.676	**33.802**	**0.721**
Baboon	29.274	**0.679**	29.624	0.475	24.399	0.433	29.501	0.454	**31.107**	0.586
Cameraman	29.514	0.399	31.861	**0.657**	24.896	0.223	31.603	0.609	**33.864**	0.603
Boat	29.307	0.476	32.301	**0.747**	24.468	0.248	31.654	0.697	**34.888**	0.711
Peppers	29.331	0.450	32.396	**0.737**	24.628	0.229	31.785	0.693	**34.173**	0.677
Pentagon	29.305	0.617	31.190	0.739	24.287	0.349	30.943	0.718	**33.816**	**0.766**
Average 3*3	29.326	0.493	31.899	**0.700**	24.514	0.271	31.390	0.658	**33.992**	0.677

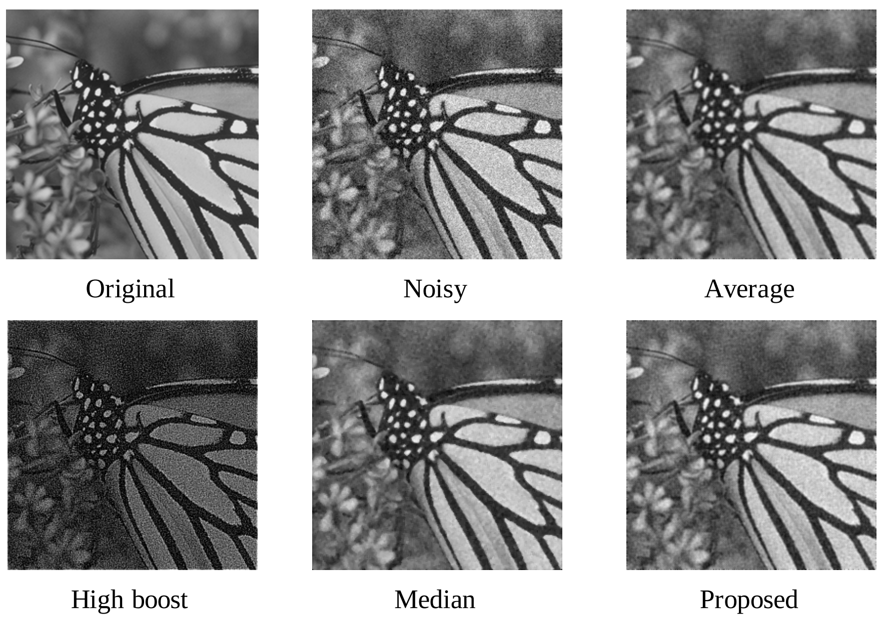


Figure 4: Resultant images of a butterfly for a 3*3 filter with a=0.2

Table 2: Results in terms of PSNR and SSIM [19] of all filters of size 5*5 for 9 gray images for image denoising

Technique	**Noisy**		**Average**		**Proposed diamond**		**Median [5]**		**Proposed plus**	
Images\ Metric	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
House	29.314	0.344	32.742	**0.739**	34.883	0.731	33.101	0.724	**34.950**	0.687
Butterfly	29.299	0.516	31.594	0.763	**33.255**	**0.795**	32.039	0.792	33.148	0.756
Lena	29.307	0.434	32.246	0.748	**33.935**	**0.759**	32.579	0.746	33.918	0.715
Barbara	29.267	0.517	30.796	0.585	32.171	0.674	30.864	0.554	**32.293**	0.662
Baboon	29.315	**0.678**	29.316	0.359	30.194	0.452	29.424	0.358	**30.416**	0.490
Cameraman	29.519	0.402	31.874	0.673	33.230	**0.688**	32.111	0.669	**33.255**	0.652
Boat	29.304	0.477	31.443	0.708	33.645	**0.768**	31.826	0.704	**33.943**	0.746
Peppers	29.349	0.451	31.863	0.745	**33.636**	**0.757**	32.289	0.754	33.555	0.719
Pentagon	29.269	0.616	30.083	0.561	31.650	**0.687**	30.383	0.586	**31.875**	0.683
Average	29.327	0.493	31.328	0.653	32.955	**0.701**	31.624	0.654	**33.039**	0.679

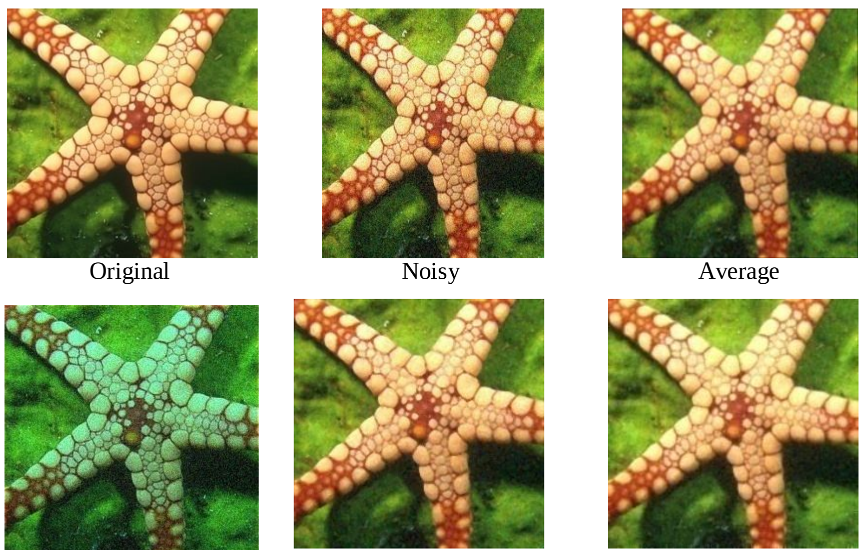
The PSNR of proposed diamond and plus filter is 5% more than the average filter and 4% more than median filter. The SSIM of the proposed diamond filter is 7% more than the average filter and median filter, while the SSIM of the proposed plus filter is 4% more than the average filter and median filter.

Table 3 displays results in terms of PSNR and SSIM [19] of all filters for 7 color images for image denoising using a 3*3 filter and resultant images for starfish image are shown in Figure 5.

It is observed from Table 3 that for color images PSNR is improved 14% than averaging and median filter by the newly proposed technique and SSIM is also improved 24% than the noisy image but best SSIM is obtained by averaging filter. The proposed technique has 3% less SSIM as compared to averaging technique.

Table 3: Results in terms of PSNR and SSIM of all filters for 7 color images for image denoising using a 3*3 filter

Technique	**Noisy**		**Average**		**High Boost**		**Median**		**Proposed**	
Images\ Metric	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
Parrot	29.378	0.365	32.667	**0.726**	27.148	0.303	32.18	0.665	**36.540**	0.665
Butterfly	29.310	0.582	31.513	**0.789**	27.016	0.497	31.35	0.771	**34.666**	0.766
House	29.299	0.355	32.443	**0.665**	26.906	0.299	31.99	0.608	**37.848**	0.624
Barbara	29.280	0.470	32.076	**0.744**	27.024	0.391	31.62	0.694	**35.957**	0.709
Starfish	29.290	0.520	31.840	**0.772**	27.140	0.440	31.52	0.733	**35.646**	0.7557
Leaves	29.302	0.697	30.610	0.844	26.988	0.6004	31.06	0.840	**36.078**	**0.881**
Tower	29.324	0.309	32.662	**0.668**	26.998	0.2569	32.16	0.602	**37.442**	0.594
Average 3*3	29.312	0.471	31.858	**0.727**	27.039	0.386	31.86	0.719	**36.218**	0.707



Original Noisy Average

High boost Median Proposed

Figure 5: Resultant images of starfish color image for a 3*3 filter with a=0.2

5. EFFECT OF VARYING FILTER SIZE

As filter size is increased PSNR is reduced. Even SSIM [19] is decreased for averaging and median for 5*5, 7*7 and 9*9 sizes. For newly proposed techniques SSIM is increased by 5*5 size but again decreased by 7*7 and 9*9 size. For 7*7 and 9*9 size filters, both PSNR and SSIM are reduced [Table 4]. Therefore, we do not go on increasing filter size further. For color images, we work with only 3*3 sizes. High boost filter results are not promising for denoising so not considered for further extended dimensions. It is also observed from Table 4 that diamond filter gives best SSIM and plus filter gives best PSNR. Results are best by 3*3 size filter as compared to higher size filters.

Table 4: PSNR and SSIM for different filter sizes on gray images denoising

Technique	**Noisy**		**Average**		**Median**		**Proposed diamond**		**Proposed plus**	
Filter Size \ Metric	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
3*3	29.326	0.493	**31.899**	**0.700**	31.390	**0.658**	**33.992**	0.677	**33.992**	0.677
5*5	29.327	0.493	31.328	0.653	**31.624**	0.654	32.955	**0.701**	33.039	**0.679**
7*7	29.328	0.493	30.699	0.595	31.289	0.624	32.167	0.665	32.351	0.656
9*9	29.322	0.492	30.162	0.541	30.865	0.587	31.587	0.621	31.795	0.627

6. EFFECT OF VARYING VALUE OF 'a'

If the value of 'a' varied it is observed that as we go on increasing the value of 'a' from 0.1 to higher; the value of PSNR goes on increasing and the value of SSIM is decreased. Value of weight 'a' is taken as 1 % of standard deviation of noise.

7. EFFECT OF VARYING NOISE VARIANCE

Noise Variance is varied as 10, 15, 20,30,40,50,100 and the effect is analyzed for 9 gray images considered and average results for each variance are compared in tables below. Table 5 compares average PSNR for different noise variance for 9 gray images. Figure 6 displays the comparative chart for average PSNR for different noise variance for 9 gray images. Similarly, Table 6 compares average SSIM for different noise variance and Figure 7 displays the comparison chart for SSIM. Table 7 compares average execution time for different noise variance by different techniques.

Table 5: Comparison of all techniques average PSNR for different noise variance for 9 gray images

Technique\ Variance	Noisy	Average	Highboost	Median	Proposed
10	32.041	33.417	24.309	33.754	**35.680**
15	30.200	32.673	24.397	32.458	**34.851**
20	29.331	31.939	24.515	31.482	**33.987**
30	28.518	30.748	24.804	30.211	**32.687**
40	28.137	29.944	25.090	29.465	**31.808**
50	27.922	29.419	25.337	29.020	**31.230**
100	27.503	28.275	26.046	28.074	**29.798**
Average	29.093	30.916	24.928	30.638	**32.863**

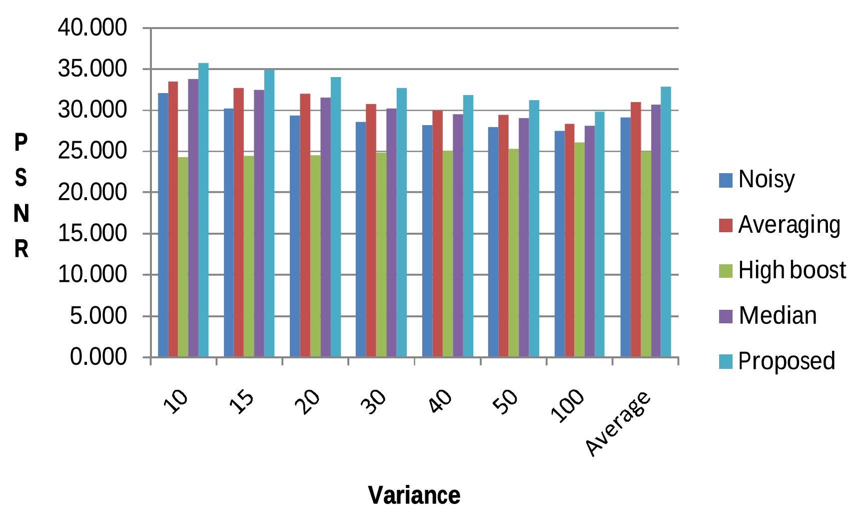


Figure 6: A chart comparing Average PSNR of all techniques for different noise variance for 9 gray images

Table 5 and Figure 6 indicate that the average PSNR of proposed technique is greater than all other techniques. This average increase by proposed technique is 13 % as compared to noisy image while averaging and median filter have increase of 6 % and 5 % respectively.

Table 6: Comparison of all techniques Average SSIM for different noise variance for 9 gray images

Technique\ Variance	Noisy	Average	Highboost	Median	Proposed
10	0.726	0.790	0.447	0.785	**0.821**
15	0.592	0.747	0.342	0.722	**0.749**
20	0.493	**0.700**	0.271	0.660	0.676
30	0.356	**0.606**	0.181	0.547	0.552
40	0.269	**0.525**	0.128	0.462	0.455
50	0.208	**0.455**	0.095	0.390	0.378
100	0.077	**0.253**	0.030	0.201	0.168
Average	0.389	**0.582**	0.213	0.538	0.543

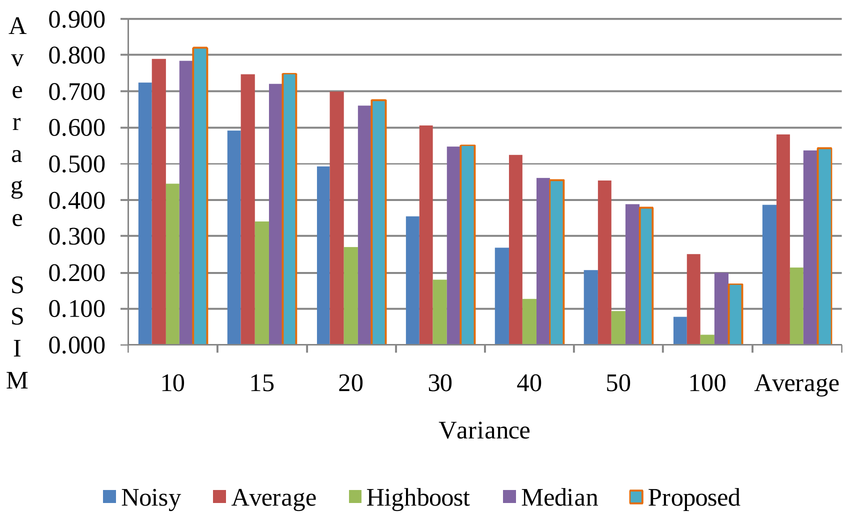


Figure 7: A chart comparing Average SSIM of all techniques for different noise variance for 9 gray images

Table 6 and Figure 7 indicate that the average SSIM of averaging technique is greater than the proposed technique but difference is not much. This average increase by averaging technique is 50 % as compared to noisy image while proposed technique and median filter have increase of 40 % and 38% respectively. The high boost filter has not improved results at all.

Table 7: Comparison of all techniques Average Time of execution for different noise variance for 9 gray images

Technique\ Variance	Average	Highboost	Median	Proposed
10	0.022	0.023	0.093	0.006
15	0.022	0.022	0.094	0.006
20	0.022	0.022	0.095	0.006
30	0.022	0.022	0.092	0.006
40	0.024	0.023	0.109	0.008
50	0.022	0.022	0.094	0.006
100	0.023	0.023	0.090	0.006
Average	0.023	0.022	0.095	0.007

Table 7 indicates that the average execution time of the proposed technique is very much less than all other techniques. This average decrease by proposed technique is of 70% as compared to averaging technique, 68% than high boost filter and 93 % than median filter.

8 FILTER FOR DEBLURRING

This proposed filter can also be used for image deblurring. The results obtained are as shown below in Table 8. It is observed that for all images considered the proposed don't care method has given best results in terms of PSNR and SSIM [19]. The proposed technique has increased PSNR by 11% and SSIM by 1% in deblurring as compared with noisy, averaging and median filter. It is also observed that in deblurring operation too as the value of 'a' is increased, PSNR is increased and SSIM is decreased. If 'a' is varied as 0.2, 0.5, 0.7 then, for the value of a=0.5, optimum results for both SSIM and PSNR are obtained. The resultant images for boat image are shown in Figure 8.

Table 8: PSNR and SSIM of deblurring for 9 gray images with filter size 3*3 and a=0.5 and sd =3

Technique	**Noisy**		**Average**		**High Boost**		**Median**		**Proposed**	
Images\ Metric	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
House	32.571	0.702	32.562	0.710	24.160	0.528	32.613	0.713	36.279	0.711
Butterfly	30.254	0.628	30.178	0.625	24.624	0.444	30.235	0.631	32.899	0.635
Lena	31.390	0.660	31.330	0.662	24.303	0.478	31.401	0.667	35.088	0.667
Barbara	30.528	0.554	30.483	0.555	24.306	0.389	30.512	0.557	34.181	0.561
Baboon	29.085	0.278	29.065	0.275	24.370	0.189	29.074	0.276	31.761	0.283
Cameraman	31.862	0.632	31.844	0.640	25.087	0.487	31.888	0.642	34.519	0.640
Boat	30.322	0.594	30.258	0.594	24.382	0.428	30.310	0.598	34.656	0.600
Peppers	30.772	0.650	30.709	0.650	24.498	0.468	30.756	0.653	34.440	0.656
Pentagon	29.628	0.380	29.601	0.373	24.179	0.255	29.617	0.376	33.061	0.386
Average	30.712	0.564	30.670	0.565	24.434	0.407	30.712	0.568	34.098	0.571

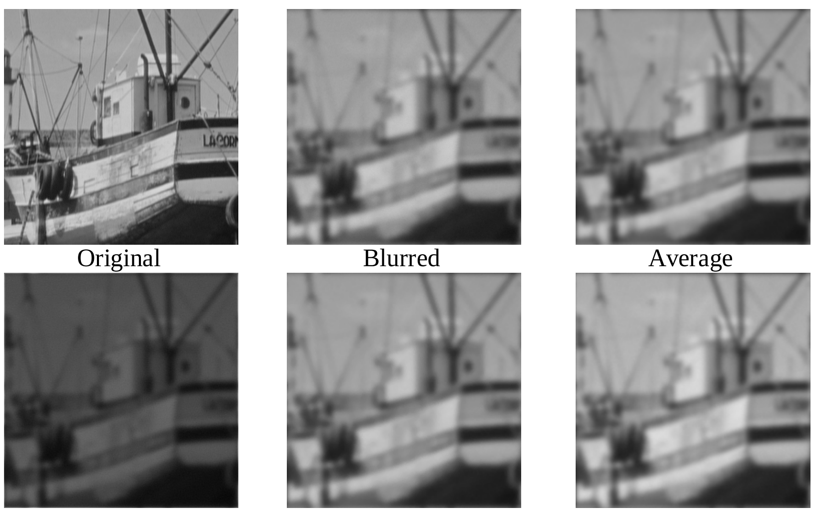


Figure 8: Resultant images of boat image for a 3*3 filter with a=0.5

9 CONCLUSION

This paper proposed a new weighted filter. For calculations, this filter considers only 4 neighboring pixels and therefore reduces the complexity of filtering operation. The results are compared for grey and color images and are found better in terms of PSNR and SSIM [19]. The proposed technique has improved PSNR by 13% and SSIM by 40% as compared to noisy images. The execution time is reduced by 70% as compared to averaging technique and 93% as compared to median filter. The analysis is done for varying the filter size, different noise variance and weight 'a'. As expected, the increase in filter size reduces the quality of the image. However, regarding 'a' factor, with an increase in the value of 'a', PSNR increases and SSIM decreases. For all noise variances the proposed filter has proved best in terms of PSNR but for variance 20 and higher averaging filter has better SSIM than proposed filter. Since the number of calculations is less, this proposed weighted filter is very fast and also an effective way for image denoising. The complexity is reduced by 50%. This filter can also be used for deblurring the image. The results are improved in terms of PSNR and SSIM by 11% and 1 % respectively. As this filter is also an efficient deblurring filter, it can be used as a dual purpose filter for denoising as well as deblurring. The results are better for denoising than deblurring. Further improvements can be done for the better quality deblurred image.

REFERENCES

[1] Anamika Maurya, Rajinder Tiwari," **A novel method of Image Restoration by using different types of filtering techniques",** *International Journal of Engineering Science and Innovative Technology* (IJESIT) Vol. 3, Issue 4, pp.124-129, July 2014.

[2] Antoni Buades, Bartomeu Coll, Jean-Michel Morel,"**A non-local algorithm for image denoising,**" *IEEE International Conference on Computer Vision and Pattern Recognition*, Vol. 2, 2005, pp- 60-65.

[3] Chao Wang, Yong Yue, Feng Dong, Yubo Tao, Xiangyin Ma, Gordon Clapworthy, Hai Lin, Xujiong Ye," **Nonedge-Specific Adaptive Scheme for Highly Robust Blind Motion Deblurring of Natural Images"**, *IEEE Trans. on Image Processing*, Vol. 22, No. 3, pp-884-897, March 2013.

[4] Peyman Milanfar," **A tour of Modern Image Filtering**", *IEEE Signal Processing Magazine*, 2013, pp-106-128.

[5] Rakesh M.R, Ajeya B, Mohan A.R., "**Hybrid Median filter For Impulse Noise Removal of an Image in Image Restoration",** *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, Vol. 2, No. 10, , pp-5117-5124, October 2013.

[6] Jae-Chern Yoo, Chang Wook Ahn," **Image restoration by blind-Wiener filter**," *IET Image Processing*, Vol. 8, Iss. 12, pp. 815–823, 2014,.

[7] Lu Lu, Weiqi Jin, Xia Wang,"**Non Local Means Image Denoising with a Soft Threshold",** *IEEE Signal Processing Letters*, Vol. 22, No.7, pp-833-837, July 2015,.

[8] Michal Aharon, Michael Elad, and Alfred Bruckstein, **"K-SVD: An Algorithm for Designing Overcomlete Dictionaries for Sparse Representation",** *IEEE Trans. on Signal Processing*, vol. 54, No. 11, pp-4311-4322 ,November 2006.

[9] Priyam Chatterjee, Peyman Milanfar," **Clustering-Based Denoising With Locally Learned Dictionaries",** *IEEE Trans. on Image Processing*, Vol. 18, No. 7, pp-1438-1451, July 2009.

[10] Michael Elad and Michal Aharon, "**Image Denoising via Sparse and Redundant Representations over Learned Dictionaries"**, *IEEE Trans. on Image Processing*, Vol. 15, No. 12, pp-3736-3745, December 2006.

[11] Lei Zhang, WeishengDong, DavidZhang, GuangmingShi," **Two-stage image denoising by principal component analysis with local pixel grouping"**, *Elsevier journal on Pattern Recognition*, Vol. 43, pp-1531–1549, 2010.

[12] Weisheng Dong, Lei Zhang,Guangming Shi, Xin Li, "**Nonlocally Centralized Sparse Representation for Image Restoration",** *IEEE Trans. on Image Processing*, Vol. 22, No. 4, pp-1620-1630, April 2013.

[13] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, Karen Egiazarian, **"Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering",** *IEEE Transactions on Image Processing*, Vol. 16, No. 8, pp-2080-2095, August 2007.

[14] Daubechies, M. Defriese, and C. DeMol, "**An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,"** *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004

[15] Amir Beck, Marc Teboulle,"**A Fast Iterative Shrinkage-Thresholding Algorithm With Application ToWavelet-Based Image Deblurring",** *IEEE International Conference on Acoustics, speech and signal processing*, April 2009, pp-693-696.

[16] Amir Beck, Marc Teboulle,"**A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems"**,*SIAM Journal on Imaging Sciences,Society for Industrial and Applied Mathematics*, Vol. 2, No. 1, pp. 183–202, 2009.

[17] J. M. Bioucas-Dias and M. A. T. Figueiredo, "**A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration,**" *IEEE Trans. on Image Processing*, vol. 16, no. 12, pp. 2992–3004, Dec. 2007.

[18] Jose M. Bioucas-Dias, Mario A. T. Figueiredo, and Joao P. Oliveira,"**Total Variation-Based Image deconvolution: A Majorization-Minimization Approach**",*IEEE International Conference on acoustics, speech and signal processing*, France, 2006.

[19] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, **"Image quality assessment: from error visibility to structural similarity**", *IEEE Trans. on Image Processing*, vol. 13, No. 4, pp.600-611, April 2004.

[20] Asem Khmag , Abd Rahman Ramli ,Shaiful H. , Syed "**Review of Image Denoising Algorithms Based on the Wavelet Transformation",** *International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE)*, Vol.2 , No.5, pp- 01-08, 2013.

[21] Vishnu Praksh.M, Shreedarshan.K, "**An Efficient Noise Removal Algorithm Based on the Noise Density",** *International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE)*, Vol. 4 No.1, pp- 01 – 05, 2015.