International Journal of Electronics Engineering Research.
ISSN 0975-6450 Volume 9, Number 7 (2017) pp. 1055-1070

http://www.ripublication.com

A Novel and Efficient Black and White Area Preserving Algorithm for Removal of Salt and Pepper Noise

Abhishek Jain and Richa Gupta
ECE Department, Jaypee Institute of Information Technology, Noida-201309, India.

Abstract

In this paper, we are proposing a novel black and white area preserving algorithm for the removal of salt and pepper noise in grayscale images that are infected by salt and pepper impulse noise. The proposed algorithm replaces the noisy central pixel by trimmed mean value when 0’s, 255’s and other pixel values are present in the selected window. When all the pixel values are 0’s and 255’s then the central pixel is replaced by either 0 or 255 depending upon higher number of pixel values (0 or 255) present in the selected window. This algorithm is based on the logic that for different noise density values (up to 0.9), all the pixels values in the selected window can’t become 0’s and 255’s and if all the pixels in the selected window of noisy image are 0’s and 255’s then central pixel in selected window can be actual image pixel not salt and pepper noise pixel. This proposed algorithm shows better results than other existing algorithms like Standard Median Filter (MF), Decision Based Algorithm (DBA), Modified Decision Based Algorithm (MDBA), Progressive Switched Median Filter (PSMF), Edge Preserving Filter etc. The proposed algorithm is verified against different grayscale images and it gives better Peak Signal-to-Noise Ratio (PSNR) value.

Keywords: Salt and Pepper Noise, Grayscale Images, Noisy Pixel, Central Pixel, Noise Density, Peak Signal-to-Noise Ratio (PSNR), Black and White Area Preserving Algorithm.

1. INTRODUCTION

Salt and pepper noise is mainly originated in images during signal addition stage. It is caused from presence of dust particles in image acquisition source or due to overheated faulty components. It is also caused due to bit errors in transmission. Salt and pepper noise corrupted pixel takes either minimum or maximum gray level. As images can be corrupted by different types of noises such as impulse, additive or signal dependent noise thus, noise removal is difficult task. Linear filtering, which can be used to remove additive noise in an image, blur edges and it fails to reduce impulse noise. This disadvantage leads to the use of non-linear filtering in salt and pepper noise reduction [1]. Due to its good noise removal power and efficient computation features, the median filter was the most commonly used nonlinear filter for removing salt and pepper noise. However, when the noise density value is over 0.5, some details and edges of the original image are spread by the median filter. Different types of median filter have been used for removing the salt and pepper noise, e.g., the Adaptive Median filter [2], the switching median filter (SMF) [3, 4], Decision Based Algorithm (DBA), Modified Decision Based Unsymmetric Trimmed Median Filter (MDBUTMF) and High Density Bilateral Filter (HDBF).

Adaptive Median Filter (AMF) is efficient to reduce salt and pepper noise at low noise densities [2]. But at high noise densities (noise density value > 0.5), the kernel window size need to increase which may result in blurring the image using adaptive median filter. In order to avoid this drawback, Decision Based Algorithm (DBA) [5] is proposed where a 3×3 window is used to reduce salt and pepper noise. If the processing central pixel value is 0 or 255 it is processed or else it is left unmodified. At high noise density values, the calculated median value can be 0 or 255 which are noisy pixel values and can’t be used for replacement. Thus, neighboring pixel is used for replacement of the noisy pixel. But, repeated replacement of noisy pixel by neighboring pixel causes streaking effect in the output image. To overcome the above problem, Decision Based Unsymmetric Trimmed Median Filter (DBUTMF) is proposed. At high noise densities, if the selected window contains all 0’s or 255’s or both then, trimmed median value cannot be obtained. So this algorithm does not give better results at very high noise density values (0.8 to 0.9). In order to avoid this drawback, The Modified Decision Based Unsymmetric Trimmed Median Filter (MDBUTMF) [6] algorithm is proposed. Edge-Preserving Algorithm (EPA) [7] is using a directional correlation dependent filtering technique to remove salt and pepper noise.

A Bilateral filter is also an efficient salt and pepper noise reducing filter that uses weights based upon spatial and radiometric similarity [9]. The bilateral filter has good results in removing salt and pepper noise while preserving image details. This method is also simple, local and non-iterative. The bilateral filter is based upon the “detect and replace” methodology. Noise detection is based on the absolute difference between a current pixel value and the reference median. The reference median is obtained from sorted quadrant median vector (SQMV).

All the currently available algorithms for removal of salt and pepper noise are not efficiently preserving the black and white area of the image. In this paper, we are proposing the new improved algorithm which will efficiently remove the salt and pepper noise while preserving the black and white area of the image. Proposed algorithm gives better Peak Signal-to-Noise Ratio (PSNR) as compared to previous algorithms.

The rest of the paper is structured as follows. A brief introduction of existing Modified decision based Unsymmetric trimmed median filter is given in Section 2. The detailed description of the proposed algorithm with an example is presented in Section 3. Experimental results are presented in Section 4. Finally conclusions are drawn in Section 5.

2. MODIFIED UNSYMMETRIC TRIMMED MEDIAN FILTER [6]

Trimmed median filter is used to remove the salt and pepper noise corrupted pixel from the selected 3×3 window in the corrupted image. Alpha Trimmed Mean Filtering (ATMF) is symmetric at either end of the window in the trimming process. Thus, even the uncorrupted pixels are trimmed which results in loss of image details and blurring of the image. To overcome the above problem, an Unsymmetric Trimmed Median Filter (UTMF) is used. In UTMF, the selected 3×3 window elements are arranged in either decreasing or increasing order. If the selected window contains all the elements as 0’s and 255’s, then, the noisy pixel is replaced with the mean of the elements of window. Otherwise, the pixel values 0’s and 255’s in the image (i.e., the pixel values responsible for the salt and pepper noise) are removed from the image. Then the median value of the remaining pixels is taken. This median value is used to replace the noisy pixel. This filter is called trimmed median filter because the pixel values 0’s and 255’s are removed from the selected window.

3. PROPOSED ALGORITHM

The proposed black and white area preserving algorithm for removal of salt and pepper noise processes the corrupted images by first detecting the salt and pepper noise. Proposed algorithm also considers the situation that black (0) and white (255) pixels can be part of the uncorrupted image. Thus, proposed salt and pepper noise removal algorithm also detects the black and white area of the image while processing the corrupted image. This algorithm is based on the logic that for different noise density values (up to 0.9), all the pixels values in the selected window of image can't become 0's and 255's and if all the pixels in the selected window are 0's and 255's then central pixel in selected window can be actual image pixel not salt and pepper noise pixel. The processing pixel is checked whether it is noisy or noise free. That is, if the processing pixel lies between minimum and maximum gray level values then it is noise free pixel, it is left unmodified. If the processing pixel takes the minimum or maximum gray level then it can be noisy pixel or actual image pixel which is processed by the proposed algorithm. The steps of the proposed algorithm are described below.

ALGORITHM

```
for iter 1:N
    Step 1: Select 2-D window of size 3x3 (if Noise Density < 0.5) or of size 5x5 (for Noise Density >= 0.5). Assume that the pixel being processed is $P_{ij}$.
    Step 2: If $0 < P_{ij} < 255$ then $P_{ij}$ is an uncorrupted pixel and its value is left unchanged.
    Step 3: If $P_{ij} == 0$ or $P_{ij} == 255$ then $P_{ij}$ can be corrupted pixel or actual image pixel and two cases are possible as given in Case i) and ii).
        Case i): If the selected window contains all the elements as 0's and 255's. Then center pixel can be actual image pixel and replace center pixel value by either 0 or 255 depending upon higher number of pixel values (0 or 255) present in the selected window.
        Case ii): If the selected window not contains all elements as 0's and 255's. Then remove elements having 255's and 0's values and find the mean value of the remaining elements. Then replace center pixel value with the mean value.
    Step 4: Repeat steps 1 to 3 until all the pixels in the entire image are processed.
    Step 5: Calculate PSNR value of current output denoised image.
    if iter > 1
        i) If PSNR value of current denoised output image is greater than PSNR value of previous denoised output image. Then, send current output denoised image for further processing in next iteration.
```

```
      ii) If PSNR value of current denoised image is less than or equal to PSNR
    value of previous denoised image. Then, send previous output denoised image as
    final output image.
     else
        Go to next iteration.
     end
  end
```

The pictorial representation of each case of the proposed algorithm is shown in Fig. 1.

The detailed description of each case of the flow chart shown in Fig. 1 is illustrated through an example in Section 4.

4. ILLUSTRATION OF PROPOSED ALGORITHM

In proposed algorithm, each and every image pixel of the corrupted image is verified for the presence of salt and pepper noise. Different cases are illustrated in this Section. If the processing image pixel value is 255/0 and all other pixel values in the window are either 0's or 255's is illustrated in Case i). If the processing image pixel value in the window is 0 or 255 and other pixels in the window also contain values other than 0's or 255's is illustrated in Case ii). If the processing image pixel in the window is not 255/0 and its value lies between 0 and 255 is illustrated in Case iii).

Case i): If the selected window contains 255/0 pixel value for processing pixel and all other pixel values in the window are either 0's or 255's. Suppose noise density value is 0.4.

$$\begin{bmatrix} 0 & 0 & 255 \\ 255 & <255> & 0 \\ 0 & 255 & 255 \end{bmatrix}$$

In above example, central processing pixel (P_{ij}) value is "255".

Since all the window elements surrounding central processing pixel (P_{ij}) are 0's and 255's and noise density value is 0.4, all the pixels values in the selected window can't become 0's and 255's due to salt and pepper noise. Central pixel in selected window can be actual image pixel not salt and pepper noise pixel. To solve this problem, central pixel is replaced by either 0 or 255 depending upon higher number of pixel values (0 or 255) present in the selected window. As number of 0's in the selected window is 4 and number of 255's in the selected window is 5. Thus, the central processing pixel value (255) remains unchanged.

Case ii): If the selected window contains 255/0 pixel value for processing pixel and neighboring pixel values contains some pixels that adds salt ("255") and pepper ("0") noise to the image:

$$\begin{bmatrix} 56 & 0 & 72 \\ 84 & <0> & 255 \\ 31 & 255 & 42 \end{bmatrix}$$

In above example, central processing pixel (P_{ij}) value is "0".

Now eliminate the salt and pepper noise from the selected window. That is, removal of 0's and 255's. The 1-Dimensional array of the above matrix is [56 0 72 84 0 255 31 255 42]. After removal of 0's and 255's, the pixel values in the selected window will be [56 72 84 31 42]. Here the mean value is 57. Hence replace the central processing pixel (P_{ij}) value by 57.

Case iii): If the selected window contains a noise free pixel as a central processing pixel (P_{ij}), it does not require any further processing.

$$\begin{bmatrix} 43 & 3 & 61 \\ 80 & <50> & 0 \\ 34 & 255 & 38 \end{bmatrix}$$

In above example, central processing pixel (P_{ij}) value is "50".

Since "50" is a noise free pixel thus, it does not require any further processing. Thus, value of central processing pixel (P_{ij}) will remain the same ($P_{ij} == 50$).

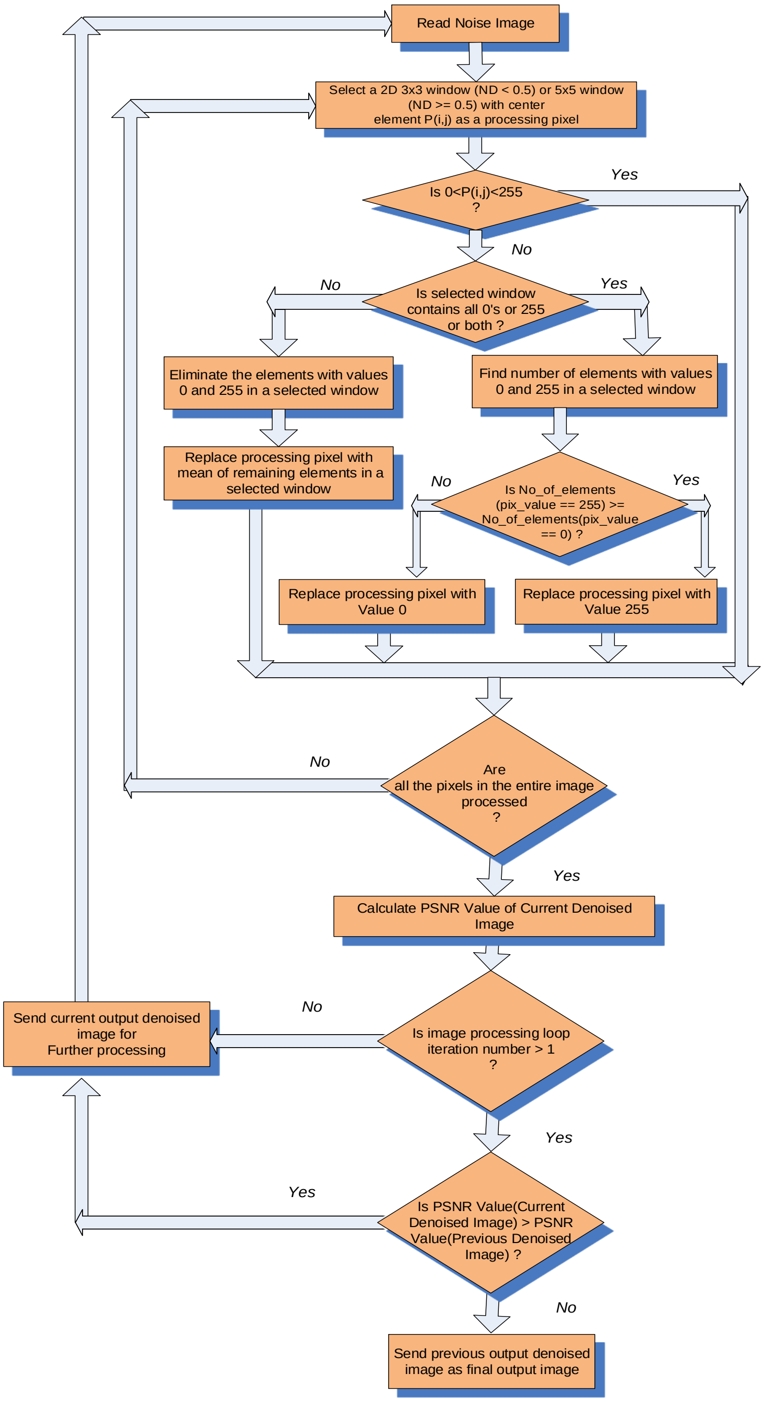


Fig. 1. Flow Chart of Proposed Algorithm

5. RESULTS

The performance of the proposed algorithm is tested with different 512 × 512 grayscale test images as shown in Fig. 2. The noise density value is varied from 0.1 to

0.9. Denoising performances of different algorithms are quantitatively measured by the PSNR as defined in equation (1) below:

$$PSNR\ in\ dB = 10\log_{10}(\frac{255^2}{MSE}) \qquad (1)$$

$$MSE = \frac{\sum_i \sum_j \big(Y(i,j) - D(i,j)\big)^2}{M\ X\ N} \qquad (2)$$

Where MSE stands for mean square error, $M\ X\ N$ is size of the image, Y denotes the original image and D represents the denoised image.

The PSNR value of the proposed algorithm is compared against the existing algorithms for different images by varying the noise density from 10% to 90% and is shown in Tables I-V. Plots of PSNR against noise densities for different images are shown in Fig. 5. From the Tables I-V and PSNR plots in Fig. 5, it is observed that the performance of the proposed algorithm is better than the existing algorithms at both low and high noise densities.

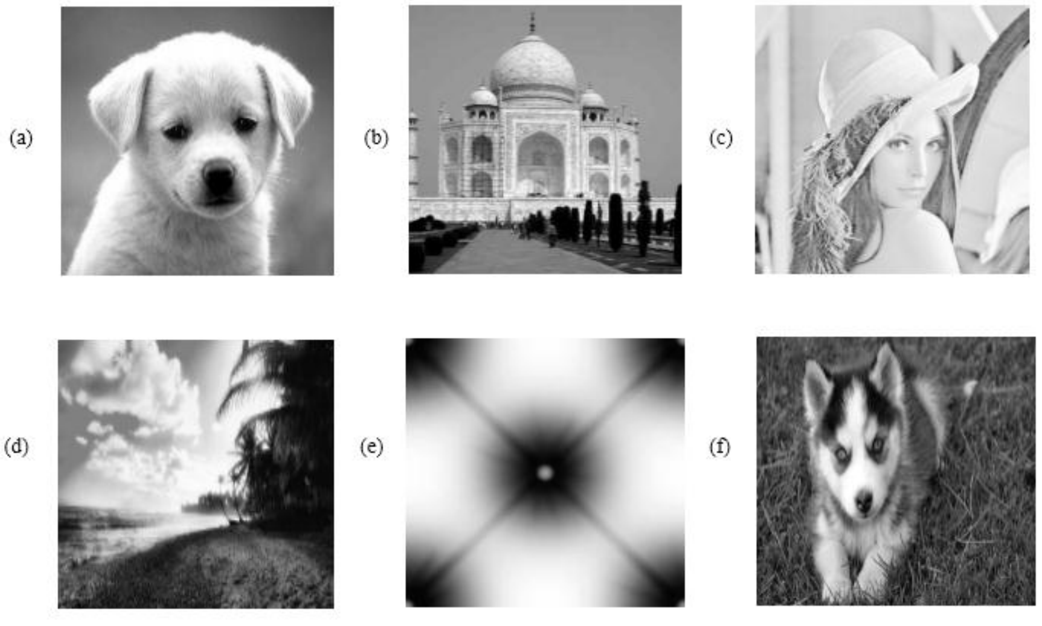


Fig. 2. Test Images. (a) Dog, (b) Taj, (c) Bright Lena, (d) Palms, (e) Pattern, (f) Dog in Grass.

Fig. 2. Test Images. (a) Dog, (b) Taj, (c) Bright Lena, (d) Palms, (e) Pattern, (f) Dog in Grass.

TABLE I. COMPARISON OF PSNR VALUES OF DIFFERENT ALGORITHMS FOR DOG IMAGE AT DIFFERENT NOISE DENSITIES

PSNR in dB	**Noise Density in %**								
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
MF	38.22	34.04	30.92	27.90	24.39	19.54	14.66	10.85	7.44
DBA	39.92	36.79	34.25	32.36	30.39	28.32	26.04	23.31	19.59
EP	32.64	30.62	27.98	25.53	24.33	22.16	19.88	18.45	16.30
PSMF	35.23	31.31	27.96	24.60	20.81	11.83	9.47	7.84	6.23
UNTRIMMED	35.51	32.08	29.73	27.92	26.03	23.76	20.94	17.85	14.06
MDBUTMFG	38.45	34.93	32.40	30.63	29.01	27.66	26.38	24.59	20.00
Bilateral	37.74	33.64	31.17	28.99	26.68	23.98	20.98	17.79	14.02
Proposed filter	**40.21**	**37.43**	**35.42**	**34.17**	**32.34**	**31.33**	**30.21**	**29.06**	**26.99**

TABLE II. COMPARISON OF PSNR VALUES OF DIFFERENT ALGORITHMS FOR BRIGHT LENA IMAGE AT DIFFERENT NOISE DENSITIES

PSNR in dB	**Noise Density in %**								
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
MF	34.22	31.12	28.07	25.39	22.70	18.08	13.98	10.01	7.01
DBA	36.20	33.85	31.64	29.79	28.17	26.24	24.47	22.06	19.36
EP	35.47	33.53	31.45	29.44	27.11	24.89	22.33	19.93	17.56
PSMF	29.75	26.91	24.51	22.07	19.44	15.70	9.02	7.13	5.75
UNTRIMMED	34.96	32.91	31.19	29.78	28.06	25.35	21.45	17.09	12.90
MDBUTMFG	34.80	33.00	31.34	30.22	29.02	28.21	27.03	25.27	19.40
Bilateral	36.39	34.33	32.04	29.98	27.58	24.35	20.65	16.61	12.71
Proposed filter	**36.46**	**34.77**	**33.05**	**31.76**	**29.11**	**28.23**	**27.27**	**26.18**	**24.38**

Table III. Comparison Of PSNR Values Of Different Algorithms For Taj Image At Different Noise Densities

PSNR in dB	**Noise Density in %**								
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
MF	33.20	29.29	26.91	24.79	21.77	18.26	14.63	10.59	7.57
DBA	33.66	30.00	27.39	25.57	24.16	22.89	20.77	19.20	16.57
EP	31.92	28.89	26.80	24.61	22.88	21.52	20.34	18.52	16.54
PSMF	29.33	26.48	24.31	21.87	14.45	11.81	9.75	7.78	6.38
UNTRIMMED	32.71	29.77	27.60	25.84	23.99	22.37	20.36	17.40	14.30
MDBUTMFG	32.22	29.82	27.98	26.54	25.06	24.08	22.82	21.40	18.46
Bilateral	33.99	31.10	28.68	26.55	24.44	22.52	20.29	17.24	14.20
Proposed filter	**34.05**	**31.73**	**29.95**	**28.67**	**25.85**	**24.98**	**24.16**	**23.04**	**21.18**

Table IV. Comparison Of PSNR Values Of Different Algorithms For Palms Image At Different Noise Densities

PSNR in dB	**Noise Density in %**								
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
MF	32.92	29.67	27.40	25.32	22.68	18.37	13.83	10.09	6.96
DBA	32.86	29.76	27.79	26.19	24.57	23.24	21.61	20.05	17.28
EP	31.96	29.05	27.26	25.29	23.48	21.34	19.31	17.19	14.84
PSMF	26.13	24.93	23.49	21.57	19.14	15.70	8.89	7.12	5.67
UNTRIMMED	32.88	29.62	27.75	26.31	24.84	22.77	19.99	16.37	12.48
MDBUTMFG	33.25	30.13	28.38	27.01	25.89	24.82	23.92	22.48	18.17
Bilateral	33.80	30.52	28.47	26.70	24.84	22.46	19.59	16.06	12.35
Proposed filter	**33.93**	**30.79**	**28.93**	**27.49**	**26.15**	**25.18**	**24.34**	**23.23**	**21.68**

TABLE V. COMPARISON OF PSNR VALUES OF DIFFERENT ALGORITHMS FOR DOG IN GRASS IMAGE AT DIFFERENT NOISE DENSITIES

PSNR in dB	**Noise Density in %**								
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
MF	39.43	35.73	33.25	30.48	25.74	20.20	14.92	10.85	7.55
DBA	41.23	37.37	34.68	32.51	30.34	28.21	26.09	23.71	20.72
EP	37.66	34.35	32.19	30.50	28.52	26.59	24.62	22.57	20.03
PSMF	37.03	33.23	29.79	25.85	21.13	16.64	9.61	7.80	6.26
UNTRIMMED	39.05	35.71	33.40	31.47	29.49	26.67	22.84	18.66	14.43
MDBUTMFG	39.26	35.87	33.81	32.14	30.74	29.40	28.09	26.33	20.96
Bilateral	41.05	37.21	34.59	32.03	29.29	26.04	22.18	18.22	14.25
Proposed filter	**42.62**	**39.32**	**37.27**	**35.57**	**31.77**	**30.81**	**29.72**	**28.33**	**25.87**

The qualitative analysis of the proposed algorithm against the already existing algorithms at different noise densities for dog and bright Lena images is shown in Fig. 3 and Fig. 4 respectively. In these figures, the first column represents the processed image using median filter at 70%, 80% and 90% noise densities. Subsequent columns represent the processed images for PSMF [8], DBA, MDBUTMF, Bilateral Filter,

Edge Preserving Filter and Proposed Filter. From visual inspection, it is clear that proposed algorithm gives better results than other existing algorithms. The PSNR values of images using different algorithms are given in Tables I-V. From the tables, it is clear that the proposed algorithm gives better PSNR values for the different images. Time taken by proposed algorithm is not less than other existing algorithms.

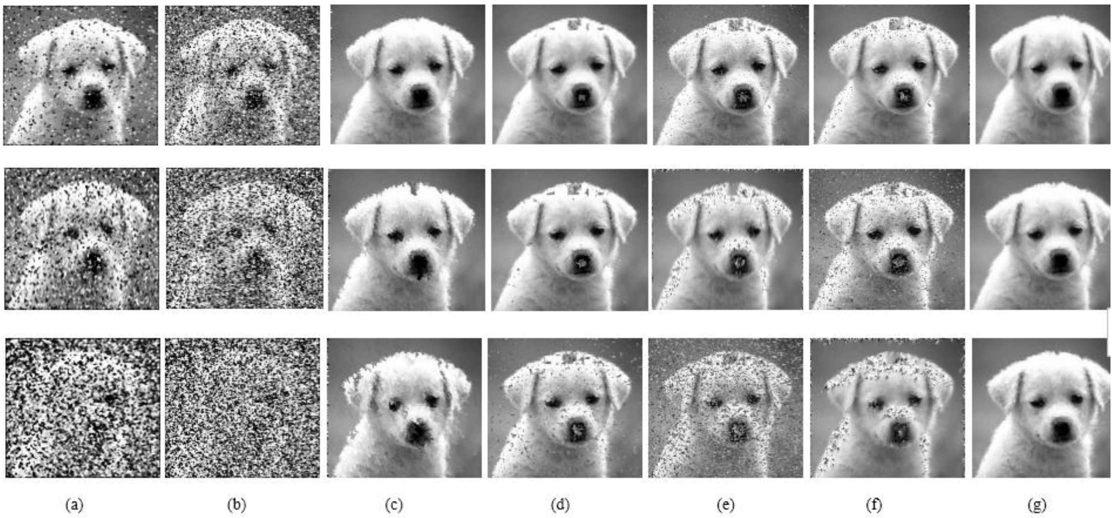


Fig. 3. Results of different algorithms for Dog image. (a) Output of MF. (b) Output of PSMF. (c) Output of DBA. (d) Output of MDBUTMF. (e) Output of Bilateral Filter. (f) Output of Edge Preserving Filter. (g) Output of Proposed Filter. Row 1 , Row 2 and Row 3 show processed results of various algorithms for image corrupted by 70%, 80% and 90% noise densities, respectively.

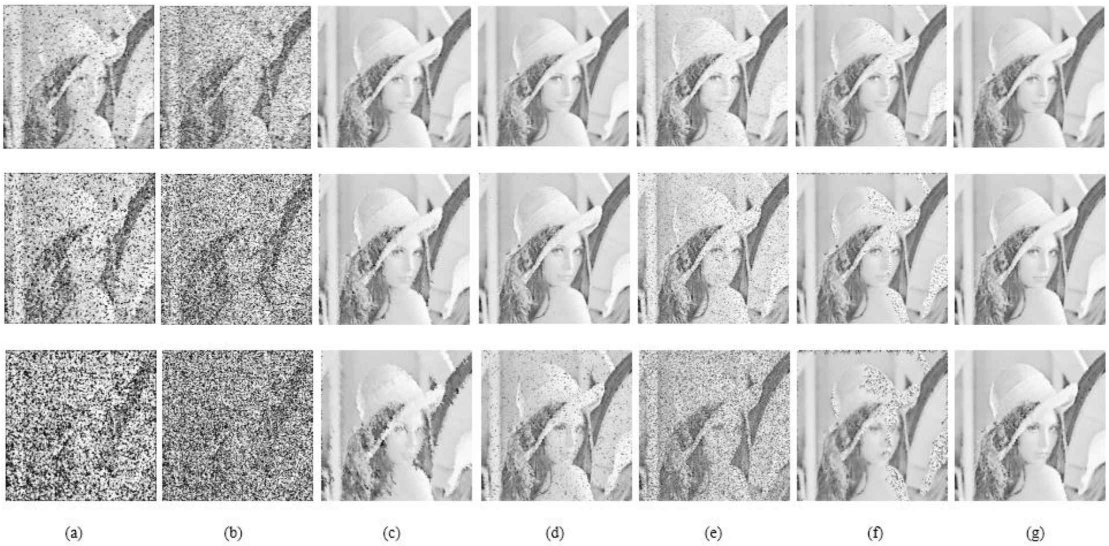
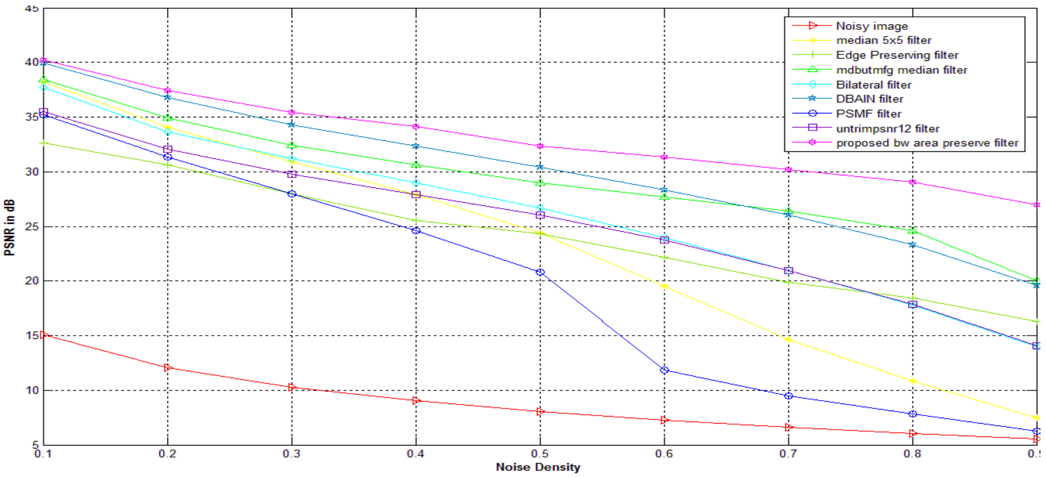
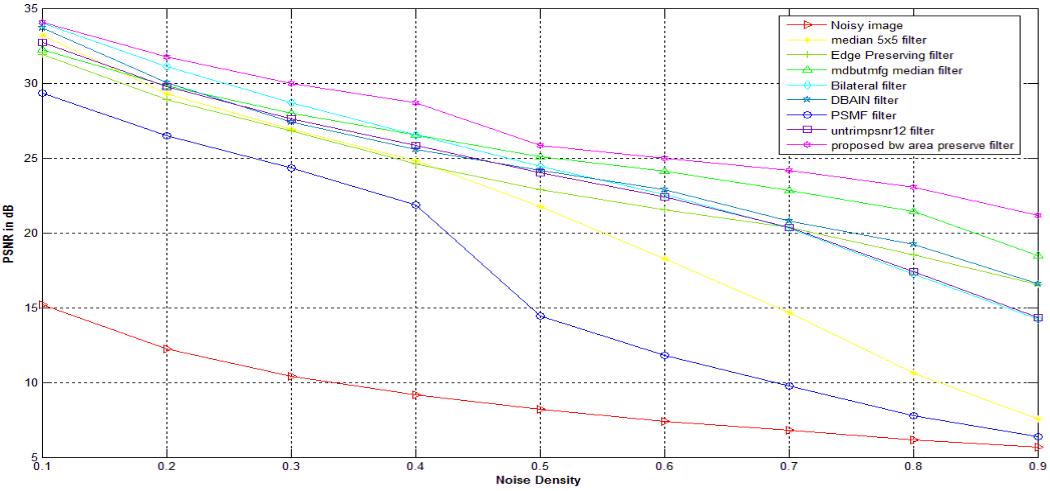


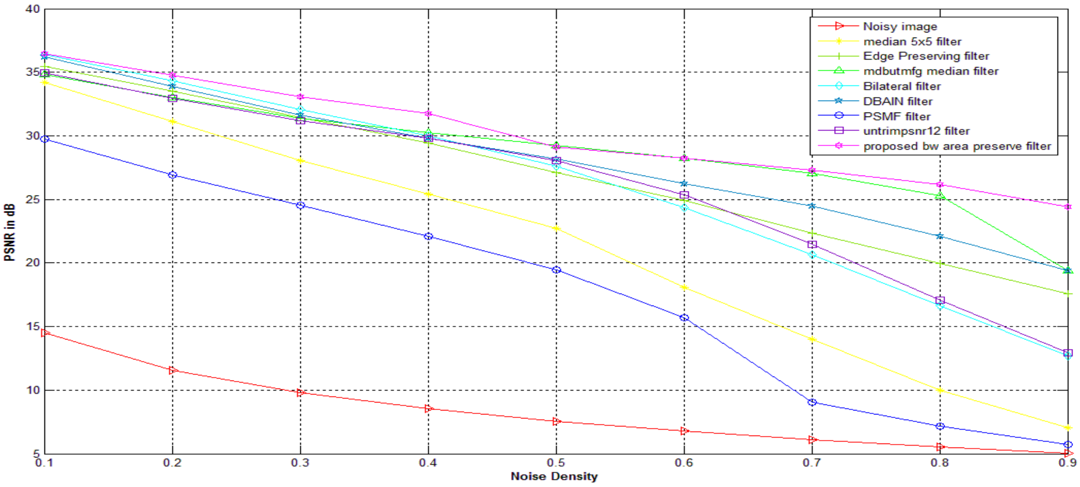
Fig. 4. Results of different algorithms for Bright Lena image. (a) Output of MF. (b) Output of PSMF. (c) Output of DBA. (d) Output of MDBUTMF. (e) Output of Bilateral Filter. (f) Output of Edge Preserving Filter. (g) Output of Proposed Filter. Row 1 , Row 2 and Row 3 show processed results of various algorithms for image corrupted by 70%, 80% and 90% noise densities, respectively.


PSNR in dB
Noise Density
5
10
15
20
25
30
35
40
45
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Noisy image
median 5x5 filter
Edge Preserving filter
mdbutmfg median filter
Bilateral filter
DBAIN filter
PSMF filter
untrimpsnr12 filter
proposed bw area preserve filter

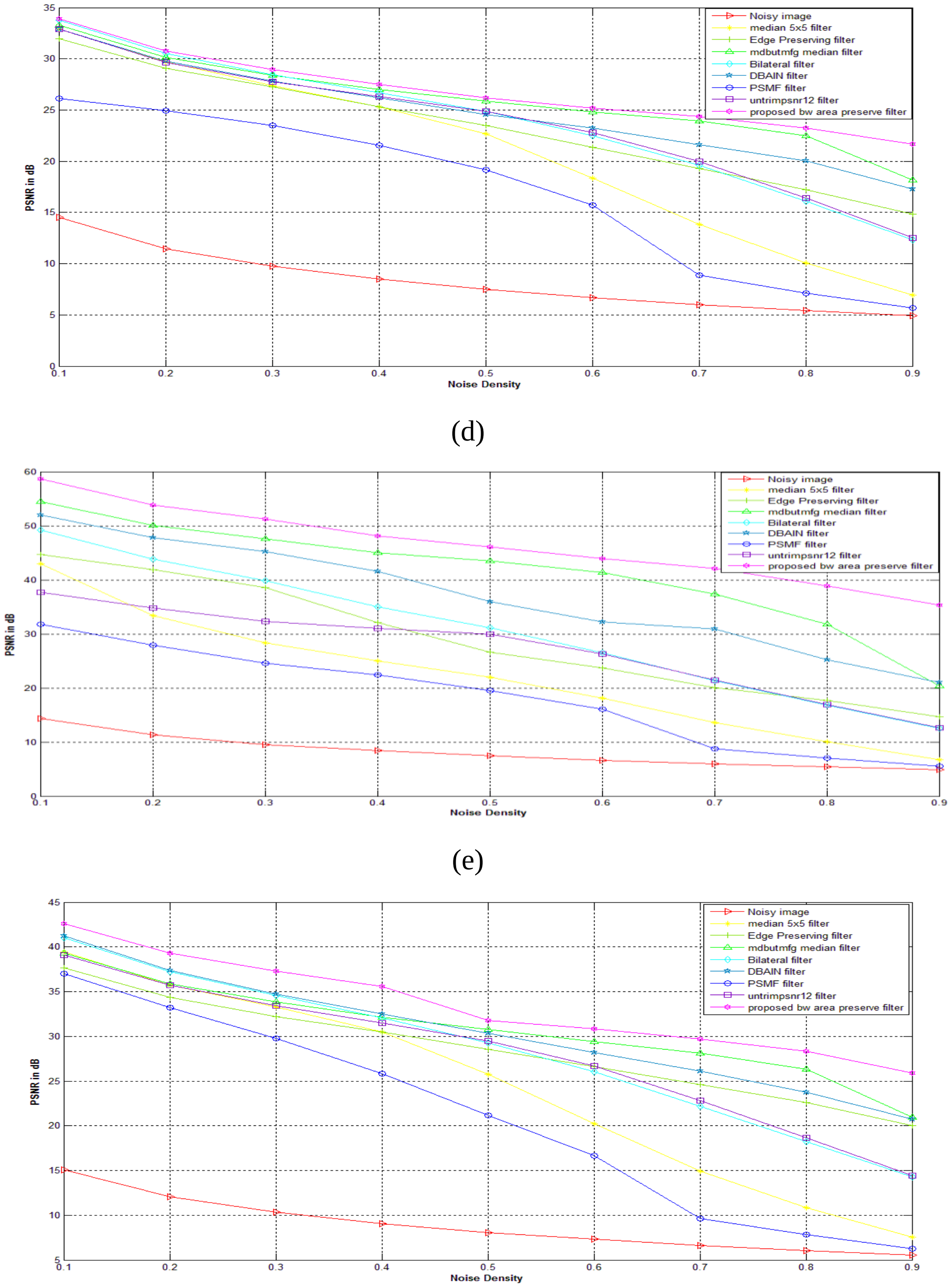
(a)


PSNR in dB
Noise Density
5
10
15
20
25
30
35
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Noisy image
median 5x5 filter
Edge Preserving filter
mdbutmfg median filter
Bilateral filter
DBAIN filter
PSMF filter
untrimpsnr12 filter
proposed bw area preserve filter

(b)


PSNR in dB
Noise Density
5
10
15
20
25
30
35
40
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Noisy image
median 5x5 filter
Edge Preserving filter
mdbutmfg median filter
Bilateral filter
DBAIN filter
PSMF filter
untrimpsnr12 filter
proposed bw area preserve filter

(c)



(d)

(e)

(f)

Fig. 5. Comparison of restoration results in PSNR (dB) for different images corrupted by various densities of Salt and Pepper Noise. (a) Dog, (b) Taj, (c) Bright Lena, (d) Palms, (e) Pattern, (f) Dog_in_grass.

6. CONCLUSION

This paper proposed a new algorithm which gives better performance in comparison with other existing noise removal algorithms in terms of PSNR. The performance of the algorithm has been tested at low, medium and high noise densities on gray-scale images. Even at high noise density levels the proposed algorithm gives better results in comparison with other existing algorithms. Both visual and quantitative results are demonstrated. The proposed algorithm is effective for salt and pepper noise removal in different images at different noise densities.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

ACKNOWLEDGMENT

The authors would like to thank Dr. Vineet Khandelwal (JIIT, Noida) and Prof. Alka Tripathi (JIIT, Noida) for their guidance and support. We would also like to thank management and team members of Imaging Division, STMicroelectronics; Faculty members and peer scholars of ECE Department, Jaypee Institute of Information Technology for their support and guidance.

REFERENCES

[1] Astola, J., and Kuosmaneen, P., 1997, Fundamentals of Nonlinear Digital Filtering. Boca Raton, FL: CRC.

[2] Hwang, H., and Hadded, R. A., 1995, “Adaptive median filter: New algorithms and results,” IEEE Trans. Image Process., vol. 4, no. 4, pp. 499–502.

[3] Zhang, S., and Karim, M. A., 2002, “A new impulse detector for switching median filters,” IEEE Signal Process. Letters, vol. 9, no. 11, pp. 360–363.

[4] Ng, P. E., and Ma, K. K., 2006, “A switching median filter with boundary discriminative noise detection for extremely corrupted images,” IEEE Trans. Image Process., vol. 15, no. 6, pp. 1506–1516.

[5] Srinivasan, K. S., and Ebenezer, D., 2007, “A new fast and efficient decision based algorithm for removal of high density impulse noise,” IEEE Signal Process. Letters, vol. 14, no. 3, pp. 189–192.

[6] Esakkirajan, S., Veerakumar, T., Subramanyam, A. N., and PremChand, C. H., 2011, “Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter,” IEEE Signal Processing Letters, vol. 18, no. 5, pp. 287-290.

[7] Jayanthi Sree, S., Ashwin, S., and Aravind Kumar, S., 2012, "Edge preserving algorithm for impulse noise removal using FPGA," In Machine Vision and Image Processing (MVIP), 2012 IEEE International Conference on, pp. 69-72.

[8] Wang, Z., and Zhang, D., 1999, “Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images,” IEEE Transactions On Circuits and Systems—II: Analog And Digital Signal Processing, Vol. 46, No. 1, pp. 78-80.

[9] Veerakumar, T., Esakkirajan, S., and Vennila, I., 2012, “High density impulse noise removal using modified switching bilateral filter,” International Journal Of Circuits, Systems And Signal Processing, vol. 6, no. 3, pp. 189-196.

Abhishek Jain received the M.Sc. degree in electronics from Delhi University, Delhi, India, in 2002 and the M.Tech. degree in computer science from IETE, Delhi, India, in 2006. He is currently pursuing the Ph.D. degree in electronics & communication engineering at JIIT, Noida, India.

He has more than 14 years of experience in industry and currently working as Senior Staff Manager at Qualcomm. He is driving key activities on functional verification flow. His current research interests include Digital Image Processing, Advanced Functional Verification Methods and System Design and Verification especially UVM based Verification, Emulation/Acceleration and Virtual System Platform. He is a member of IETE (MIETE).

Dr. Richa Gupta is a post graduate in Information Systems from Indian Institute of Technology, Kanpur and has subsequently earned her Ph.D. degree in the area of Information Theory and Coding.

She is primarily associated with JIIT as a faculty in the area of ECE. Her research interests include Image Processing, Image Verification, Speech Processing, Joint Source Channel Coding and Mathematical Modelling of Signals and Systems. She has published many papers in reputed journals. She has served as the secretory of the board of trustees of Forum for Interdisciplinary Mathematics for 2 years.