Indian Journal of Science and Technology, *Vol 8(32), DOI: 10.17485/ijst/2015/v8i32/59797, November 2015*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

An Efficient Denoising Algorithm for Detection and Removal of High-Density Impulse Noise from Digital Images

P. Rajalakshmi* and I. Daphne

Department of ECE, Velammal Institute of Technology, Chennai – 601204, Tamil Nadu, India;
rajalakshmi0126@gmail.com, daphne.issac@gmail.com

Abstract

Background/Objectives: Images that are affected by noise should be denoised in order to obtain information from it. In this paper an efficient denoising algorithm based on adaptive and unsymmetric trimmed median filter is proposed which effectively gets rid off the high-density salt and pepper noise. **Methods/Stastical Analysis:** The proposed algorithm restores the noisy pixel with the trimmed median value when, 0's and 255's are found in the neighborhood and also adaptively controls the neighborhood size based on the relative amount of noisy pixels in the selected neighborhood. Many of the existing algorithms do not preserve fine details of the image while removing noise at high noise densities. **Results/Findings:** The Proposed Algorithm (PA) is tested against grayscale and color images and it gives an exceptional Peak Signal to Noise Ratio (PSNR) and Image Enhancement Factor (IEF) especially at higher noise densities. **Conclusion/Application:** This algorithm effectively removes high density noise caused by bit error transmission or introduced during the image acquisition stage.

Keywords: Image Denoising, Impulse Noise, Median Filter, Mid Point Filter, Salt and Pepper Noise

1. Introduction

An image may be affected by noise due to a number of sources, such as fluctuation in the detector sensitivity, changes in the environment, transmission errors, quantization errors, etc. The characteristics of noise depends on its source. Noise can be broadly grouped into two classes, independent noise, and dependent noise. Image independent noise is generally described by an additive noise model and Image dependent noise can be described by a multiplicative noise model. We adopt the use of salt and pepper noise for the analysis of image dependant noise. Salt and pepper noise is commonly referred to as *intensity spikes* or *impulse noise*.

If an Image is degraded by salt and pepper noise it will have dark picture elements in bright sections and bright picture elements in dark sections[2]. Salt and Pepper noise is caused due to errors in the analog to digtal convertors, bit errors in transmission, etc. Salt and Pepper noise is characterized by the pixels whose values are either 0 (White) or 255 (Black). This looks as if Salt and Pepper has been sprinkled on the image hence the name. Noise degrades the image and should be removed by a suitable algorithm to render the image suitable for usage. The noise is usually specified by the proportion of pixels which are corrupted.

Image De-noising can be done either in spatial domain or frequency domain. Algorithms that are based on spatial domain filtering act directly on the pixel values. Neighborhoods are defined using masks or windows or kernels. Several non-linear filters have been introduced to de-noise the images corrupted by Salt and Pepper Noise. The basic and standard filter is the median filter. The median filter preserved the edges of the images only at low noise densities[3]. Hence, adaptive median filter was introduced which increased its window size when the noise density increased. This caused the images to blur. Switching median filters were based on threshold values, but defining

**Author for correspondence*

these threshold values was a problem. Also at higher noise density the edge details were lost. Decision Based Algorithm (DBA)[4] was proposed which used 3 x 3 Window. The value of the pixel was replaced with the median only if the pixel is deemed noisy. But at high noise density the median value itself will be noisy in which case the neighborhood pixel is used for replacement which caused the streaking effect.

In order to overcome this drawback, Decision Based Algorithm Unsymmetric Trimmed Median Filter was proposed. This algorithm had two drawbacks.It did not give good result at very high noise density and trimmed median value could not be obtained if the selected window contained all 0's or 255's or both. To avoid this Modified Decision Based Unsymmetric Trimmed Median Filter (MDBUTMF) algorithm was proposed[1]. This algorithm is good at removing noise, even at high noise density, but the drawback is that at high noise densities, if all the pixel values within the window are '0' or '255' or both then the processing pixel is restored by the mean value of the selected window which is again noisy. In this algorithm, the noisy pixels are not efficiently processed. To overcome this drawback, at high noise densities the processing pixel is replaced by using different partial trimmed global mean value of the intensity of the image[5]. Here when the noise level is high details and edges are not recovered satisfactorily.

To overcome the drawbacks mentioned above we propose the use of partial trimmed global Midpoint value at neighborhoods having high noise density and using Unsymmetric Trimmed Median Filter at neighborhoods having low noise density. It gives exceptional Peak Signal to Noise Ratio (PSNR) and Image Enhancement Factor (IEF) than the existing algorithms.

The rest of the paper is ordered as follows. Section II explains the proposed algorithm and the steps and conditions involved in the denoising process. Section III gives a brief idea about the two types of filters, that is, Partially Trimmed Global Mid-point filter and Unsymmetric Trimmed Median Filter which are used in the proposed algorithm. Section IV provides an illustration of the proposed algorithm. Section V explains the performance analysis and comparison of the parameters with existing algorithms. Conclusions are shown in Section VI.

2. Proposed Algorithm

The first step in image de-noising is to determine whether the pixel that is being processed is noisy or not. Since we are concerned with salt and pepper noise, a pixel value of 0 or 255 is considered or treated as noise. The next step is to know the neighborhood. If the entire neighborhood consists of noisy pixels, i.e. high noise density, then partially trimmed global midpoint value is used, else the unsymmetric trimmed median filter is used. If the pixel that is being processed is neither 0 nor 255 it is left unchanged. The procedure is as follows:

Step 1: Select a 2-D 3 x 3 sliding neighborhood. Let the center pixel be $L_{i,j}$. It is the pixel that is being processed.

Step 2: If 0<Li,j<255 the pixel is left unchanged.

Step 3: If Li,j = 0 or 255 then the neighborhood is checked.

Step 4: If 3/4th or more pixels in the neighborhood are noisy then increase the window size to 5*5.

Step 5: If the selected neighborhood contains all pixels as only '0'. Then restore the pixel Li,j with the salt noise (255)trimmed global midpoint value of the image.

Step 6: If the selected neighborhood contains all pixels as only '255'. Then restore the pixel Li,j with the pepper noise(0) trimmed global midpoint value of the image.

Step 7: If the selected neighborhood contains all pixels as '0' and '255'. Then restore the pixel Li,j with the salt and pepper noise trimmed global midpoint value of the image.

Step 8: Else restore the pixel Li,j with the median of the neighborhood.

Step 9: Repeat steps 1 through 8 till the entire image is processed.

The proposed algorithm is shown as a flowchart in Figure 1.

3. Partial Trimmed Global Midpoint and Unsymmetric Trimmed Median Filter

3.1 Partial Trimmed Global Midpoint

To calculate partial trimmed global midpoint value three cases have to be considered. The global midpoint value with only salt noise trimmed (i.e. Remove all the '0' value from the whole image and calculate the midpoint for the rest of the pixels), the global midpoint value with only pepper noise trimmed (i.e. Remove all the '255' value from the whole image and calculate the midpoint for the rest of the pixels), global midpoint value with only salt and

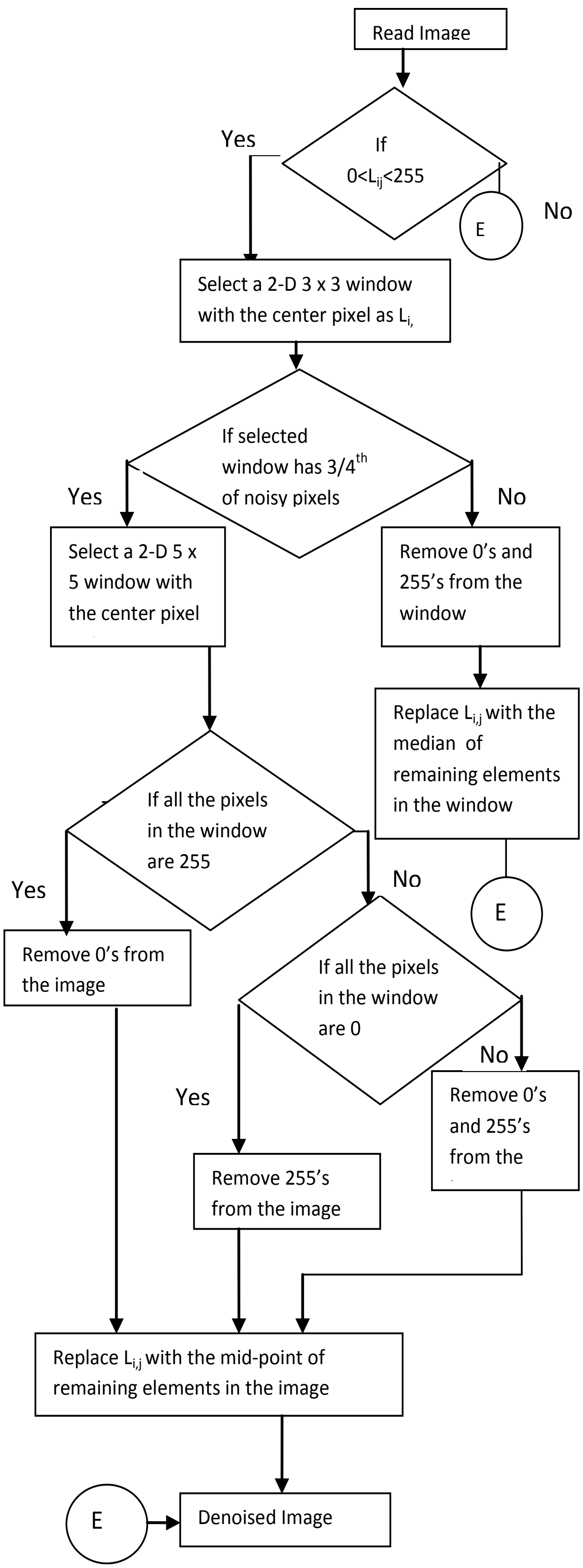


Figure 1. Flow chart of PA.

pepper noise trimmed (i.e. Remove all the '0' and '255' value from the whole image and calculate the midpoint for the rest of the pixels). Taking these different values preserves the edges and performance is also improved.

This computes the midpoint between the maximum and minimum values in the area enclosed by the filter. It works best for noise that is randomly distributed.

$$\hat{f}(x,y)=\frac{1}{2}\left[\max_{(s,t)\in S_{xy}} g(s,t)\} + \min_{(s,t)\in S_{xy}}\{g(s,t)\}\right]$$

3.2 Unsymmetric Trimmed Median Filter

The concept of a trimmed filter is to eliminate the noisy pixel from the selected window. In Alpha Trimmed Mean Filtering (ATMF), even the uncorrupted pixels are also trimmed as trimming is symmetric at one of the two ends.This causes image blurring which eventually leads to loss of image details. In order to overcome this shortcoming, we propose an Unsymmetric Trimmed Median Filter (UTMF). In UTMF, the pixel elements of the selected window are aligned in each of the two increasing or decreasing order. The pixel values concerning the salt and pepper noise are eliminated i.e., 0's and 255's are removed and the median value of the other pixels is computed. This value is then used to restore the noisy pixel. Since the pixel values 0's and 255's are eliminated from the selected window this filter is called trimmed median filter. This method removes noise in a preferred way.

4. Illustration of the Proposed Algorithm

Check each pixel and identify whether it is a noise or noise free pixel. All pixels having the value either 0 or 255 are deemed as noisy pixels. If more than $3/4^{th}$ of the pixels in a 3 x 3 window are noisy then expand to 5 x 5 window and check for the following conditions.

Case 1: If all the pixels in the window are 0's. Eliminate 255's from the entire image and restore the processing pixel with the midpoint of the remaining pixels in the image.

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Case 2: If all the pixels in the window are 255's. Eliminate 0's from the entire image and restore the processing pixel with the midpoint of the remaining pixels in the image.

255	255	255	255	255
255	255	255	255	255
255	255	255	255	255
255	255	255	255	255
255	255	255	255	255

Case 3: If all the pixels in the window are either 0 or 255. Eliminate 0's and 255's from the entire image and restore the processing pixel with the midpoint of the remaining pixels in the image.

0	0	255	255	0
0	255	0	255	0
255	0	0	0	255
0	0	0	255	255
255	255	0	0	0

Case 4: If the window contains less than $3/4^{th}$ noisy pixels. Remove the 0's and 255's and restore the processing pixel with the median of the selected window.

92	87	121
68	0	65
95	75	55

If the processing pixel is neither 0 nor 255, it remains unchanged.

5. Simulation Result

The proposed algorithm is compared with existing algorithms for various noise densities varying from 10% to 90%. A comparison of PSNR and IEF for the proposed algorithm and existing algorithms for different noise densities is shown in Table 1 and Table 2. From the following tables, it is observed that the proposed algorithm produces exceptional results when compared with other algorithms particularly at higher noise densities.

Figure 2 shows a graph of MSE, PSNR and IEF values at various noise densities. In Figure 3 and 4 shows the color and grayscale output images when various algorithms are applied on Lena image at 70% and 80% noise density. The first row shows the original image followed by the image affected by 70% noise and de-noised images when

Table 1. Performance comparison table of PSNR values for Lena image

Noise in %	PSNR in dB				
	MF	DBA	PTGMF	MDBUTMF	PA
10	29.46	39.39	42.20	42.42	42.73
20	27.08	35.49	38.27	38.87	38.29
30	25.26	33.08	35.79	35.68	35.74
40	23.18	30.56	33.45	33.26	33.74
50	20.85	28.33	31.32	30.91	31.70
60	18.7	25.67	28.57	27.83	30.33
70	15.16	23.4	25.58	23.59	28.59
80	12.29	21.11	21.60	19.9	26.35
90	8.59	17.06	17.60	15.31	22.01

Table 2. Performance comparison table of IEF values for Lena image

Noise in %	IEF				
	MF	DBA	PTGMF	MDBUTMF	PA
10	25.71	255.92	488.45	511.33	554.48
20	29.96	209.53	401.14	405.88	399.70
30	29.62	177.43	333.53	326.35	329.59
40	24.53	133.58	259.16	248.54	280.17
50	17.88	100.12	200.52	182.99	218.33
60	13.07	63.40	127.87	110.5	193.05
70	6.8	45.42	74.73	47.39	149.39
80	3.97	30.38	34.08	22.85	101.38
90	1.91	13.50	15.30	8.99	42.31

Median Filter, DBA, PTGMF, MDBUTMF and proposed algorithm are applied respectively. The second row shows similar images for 80% noise density.

The Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) is given by:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[I(i,j)-K(i,j)\right]^2$$

$$\begin{aligned} PSNR &= 10\cdot\log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20\cdot\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20\cdot\log_{10}\left(MAX_I\right)-10\cdot\log_{10}\left(MSE\right) \end{aligned}$$

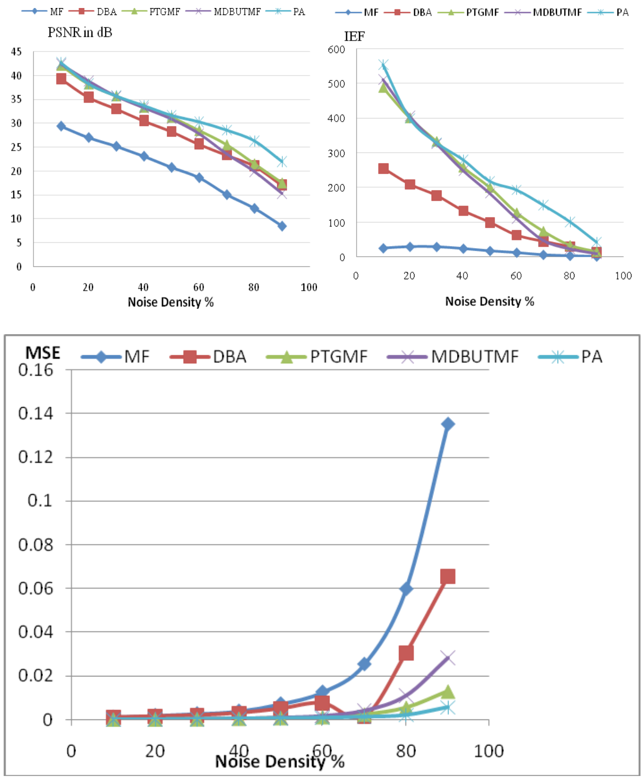
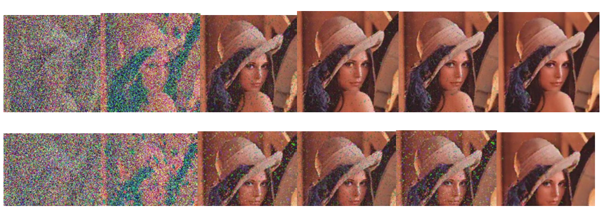


Figure 2. Graphical comparison of MSE, PSNR and IEF for Lena image.



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 3. Outcome of different algorithms for color Lena image. **(a)** Image affected by Noise. **(b)** MF Output Image. **(c)** DBA Output Image. **(d)** MDBUTMF Output Image. **(e)** PTGMF Output Image. **(f)** PA Output Image. Rows 1 and 2 show processed outcome of various algorithms for color image distorted by 70% and 80% noise densities respectively.

The Image Enhancement factor (IEF) is given by:

$$IEF = \frac{\sum_i \sum_j \left(\eta(i,j) - Y(i,j)\right)^2}{\sum \sum \left(\hat{Y}(i,j) - Y(i,j)\right)^2}$$

$\hat{Y}$ represents the denoised image, η represents the noisy image and Y denotes the original image.

(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 4. Outcome of different algorithms for grayscale Lena image. **(a)** Image affected by Noise **(b)** MF Output Image. **(c)** DBA Output Image. **(d)** MDBUTMF Output Image. **(e)** PTGMF Output Image. **(f)** PA Output Image. Rows 1 and 2 show processed outcome of various algorithms for the grayscale image distorted by 70% and 80% noise densities, respectively.

6. Conclusion

An efficient algorithm has been proposed, the performance of which is tested under different noise densities ranging from 10% to 90% on both grayscale and color images. The proposed algorithm has shown exceptional and satisfactory results in terms of MSE, IEF and PSNR than the other existing algorithms especially at higher noise densities. Both visual and quantitative results have been demonstrated. These techniques play an important role in a variety of applications such as medical imaging, Satellite imaging, web browsing, archeology, etc. The algorithm can also be extended to remove noise from video.

7. References

1. Esakkirajan S, Veerakumar T, Subramanyam AN, PremChand CH. Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter. IEEE Signal Processing Letters. 2011 May; 18(5):287–90.
2. Gonzalez, Woods. Digital image processing. Pearson Education. 2005.
3. Astola J, Kuosmaneen P. Fundamentals of nonlinear digital filtering. Boca Raton, Florida: CRC; 1997.
4. Srinivasan KS, Ebenezer D. A new fast and efficient decision based algorithm for removal of high density impulse noise. IEEE Signal Processing Letters. 2001 March; 14(3):189–92.
5. Raza MT, Sawant S. High density salt and pepper noise removal through decision based partial trimmed global mean filter. Nirma University International Conference on Engineering (NUiCONE). 2012 Dec 6-8. p. 1–5.

6. Hwang H, Hadded RA. Adaptive median filter: New algorithms and results. IEEE Trans Image Process. 1995 Apr; 4(4):499–502.
7. Chandra V, Deokar S, Badhe S, Yawle R. Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed adaptive median filter. Int J Eng Adv Tech. 2013 Feb; 2(3):495–9.
8. Santhosh Kumar K, Gopi M. Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter. Int J Sci Res. 2013 Mar; 2(3).
9. Usharani LS, Thiruvalar Selvan P, Jagajothi G. An improved de-noising algorithm for highly corrupted color videos using FPGA based impulse noise detection and correction techniques. Indian Journal of Science and Technology. 2015 Apr; 8(7):664–9.
10. Kadali KS, Rajaji L. FPGA and ASIC implementation of systolic arrays for the design of optimized median filter in digital image processing applications. Indian Journal of Science and Technology. 2014 Nov; 7(7):99–103.
11. Karthickmanoj S, Sinthuja N, Manoharan. Removal of impulse noise using adaptive weighted median filter. Indian Journal of Science and Technology. 2014 Oct; 7(6):61–3.
12. Kheirkhah E, Tabatabaie ZS. A hybrid face detection approach in color images with complex background. Indian Journal of Science and Technology. 2015; 8(1):49–60.