

A Decision Based Unsymmetrical Trimmed Variants for the Removal of High Density Salt and Pepper Noise

K.Vasanth
Research scholar
Sathyabama University
OMR, Chennai, Tamilnadu, India

V.Jawahar senthilkumar
Research Guide
Anna University
Chennai, Tamilnadu, India

Rajesh.V
Faculty
Sathyabama University
OMR, Chennai, Tamilnadu, India

ABSTRACT

A Decision based scheme using a improved mesh based snake like sorting is proposed for the restoration of gray scale images that are heavily corrupted by salt and pepper noise. The proposed algorithm uses modified mean or median for image restoration. The processed pixel is examined for 0 or 255; if checked pixel is equal to 0 or 255, then it is considered as noisy pixel else not noisy. The noisy pixel is replaced by median of unsymmetrical trimmed output, if the current window has at least three noisy pixels or mean of first and last values of the un-symmetrical trimmed output. The uncorrupted pixel is left unchanged. The proposed algorithm shows excellent noise suppression capability with good edge preservation in heavy noisy conditions both qualitative and quantitatively. The proposed algorithm was applied on various grayscale images and found to have excellent PSNR and high IEF, low MSE and consumes less time even at very high noise densities with edge preservation.

Keywords

Decision based filter, Unsymmetrical trimmed mean filter, salt and pepper noise

1. INTRODUCTION

Images are often corrupted by salt and pepper noise due to error in transmission of digital images. In yester years removal of noise is facilitated, by using linear filtering techniques. These linear approaches were very popular because of its mathematical simplicity. The objective of noise removal is to eliminate the salt and pepper noise with minimum deformation caused to the image. The linear filters were not effective in removing non Gaussian noise. The removal of impulse noise leads to blurred and defamed features of the image. Hence nonlinear filters were introduced. The most widely used nonlinear filters are median filters. Median filters would eliminate impulse and preserves edges in the image. For increasing noise densities the standard median filter (SMF) flatters also irrespective of pixel is noisy or not, median is applied to entire image [1]. An adaptive median filter (AMF) removed impulse noise at high noise densities, owing to its increasing window size caused blurring of images [2]. Many switched median filters were proposed to detect and correct only the corrupted pixel [3]-[4]. The flaw of the switched median filters is that it detected and corrected the impulse, but did not take local feature such as edges into account. For the removal of high density impulse noise (DBA) Decision based filters [5] were proposed. Due to the replacement of neighborhood pixel the performance of the image tend to flatter above 60% by exhibiting "Streaks" in images. Hence the edges of the images are destroyed. To avoid streaking in images at heavy noise conditions a two

stage cascaded approach [6] was proposed. The first stage of the filter performed impulse detection by comparing the processed pixel with minimum and maximum grayscale value of an image. If the processed pixel is noisy then median is replaced else left unaltered. The second stage of the cascaded algorithm is a mere replacement of unsymmetrical trimmed midpoint filter; this could damage the fine details of the image. The flaw of streaking encountered in DBA was rectified in Modified decision based filter (MDBF)[7]. This algorithm replaced the noisy pixel by finding median of non noisy pixels. Even at high noise densities MDBF fails to preserve edges. Modified decision based unsymmetrical trimmed median filter (MDBUTMF) [8] is proposed to eliminate the flaw of streaking and edge preservation. This algorithm replaced the corrupted pixel with median of the trimmed output. At very high noise densities the filter smudges the edge. To overcome the above flaws decision based unsymmetrical trimmed variants filter (DBUTVF) is proposed. This paper is organized as follows. Section II deals with decision based unsymmetrical trimmed variants. Section III gives the qualitative and quantitative comparison of proposed filter with existing filters. Section IV gives the concluding remarks for this letter.

2. DECISION UNSYMMETRICAL TRIMMED FILTER

2.1 Un-symmetrical trimmed filter

The crux behind the above filter is to eliminate the outliers inside the current window. Certain type of non linear filters such as Alpha trimmed mean filter (ATMF), Alpha trimmed midpoint (ATMP) etc., works on the above principle. These filters use a parameter called " α " which decides the number of pixels to be eliminated. It was found that when " α " is increased, the filter fared well. For high noise densities it does not preserve the image information due to the elimination of outlier values. Unsymmetrical trimmed filter replaced the symmetrical counterpart. The trimming was not uniform as in the previous case; it depends on pixels inside the current processing window. So for the removal of salt and pepper noise at high noise densities with edge preservation (DBUTVF) decision based unsymmetrical trimmed variants filter is proposed [9].

2.2 Snake like improved Shear Sorting

Over the years sorting algorithm is a basic operation behind all the median filters. All the existing sorting algorithms require more comparators. In this paper a new snake like improved shear sorting algorithm is proposed for ordering the entire array of processed pixels as shown in figure 1.

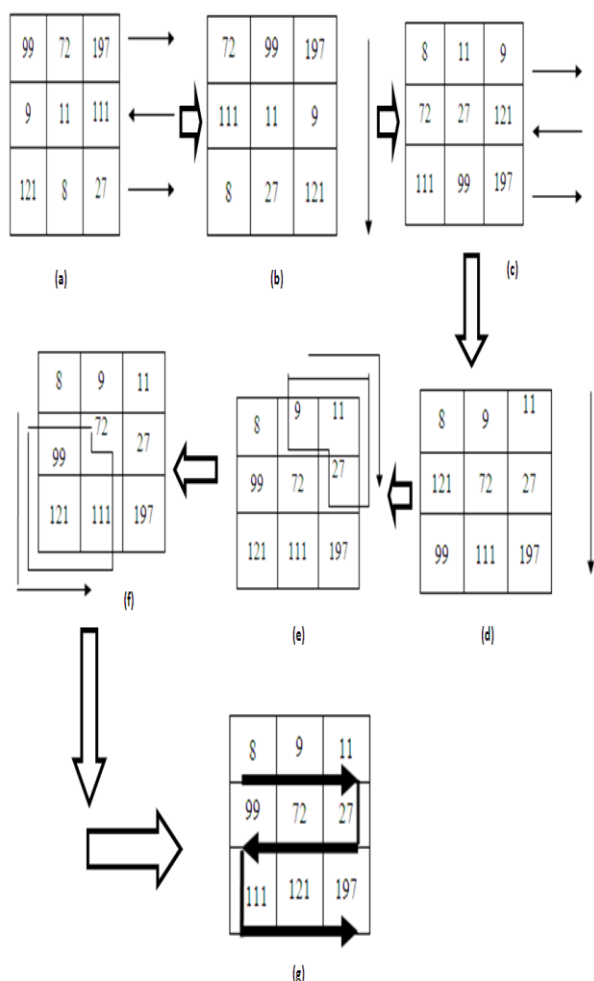


Figure 1 illustration of the proposed sorting methodology

Let D be an $m \times n$ matrix which is mapped with linear integer sequence W . Sorting the sequence W is then equivalent to sorting the elements of D in some Pre determined indexing scheme. The proposed Snake like modified algorithm consists of three basic operations row sorting, column sorting and semi diagonal sorting. The algorithm of the proposed snake like improved shear sorting algorithm is as follows.

- Step1: The considered 2D processing window as shown in figure 1.a
- Step2: Sort the 1th and 3rd rows of the 2D array in ascending order and 2nd row in descending order independently .The sorted sequence is fed to step3 as shown in figure1.b.
- Step3: Sort the three columns of the 2D array in ascending order .The sorted sequence is fed to step4 as shown in figure 1.c.
- Step4: Repeat step 2 and 3 once again as shown in figure1.d and e.
- Step5: Now Sort the upper semi diagonal of the semi sorted 2D array in ascending order as shown in figure1.e.
- Step6: Sort the Lower semi diagonal sorted array in ascending order as shown in figure1. Resulting array is sorted in a snake like order. The procedure is repeated for the other windows of the image [9].

2.3 Proposed algorithm

The Decision based unsymmetrical trimmed variants filter (DBUTVF) initially detects impulse and corrects it subsequently. All the pixels of an image lie between the

dynamic ranges [0,255]. If the processed pixel holds minimum (0) or maximum (255), pixel is considered as noisy and processed by DBUTVF else as not noisy and the pixel is unaltered. The brief illustration of the algorithm is as follows.

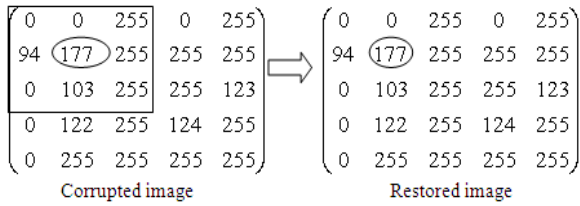
TABLE I Complexities of various sorting algorithm

SORTING TECHNIQUES	Number of Comparisons required to sort the entire 3x3 window (worst case)
Bubble sorting $o(n^2)$	81
Insertion sorting $o(n^2)$	81
Selection sort $o(n^2)$	81
Merge Sort $o(n \log n)$	25
Heap Sort $o(n \log n)$	28
Quick Sort $o(n^2)$	81
Shear sorting $o(\sqrt{n}(2n))$	54
Snake like Modified shear sorting (proposed Algorithm) $o(4n+2\sqrt{n})$	42

- Step 1: Choose 2-D window of size 3x3. The processed pixel in current window is assumed as p_{xy} .
- Step 2: Check for the condition $0 < p_{xy} < 255$, if the condition is true then pixel is considered as not noisy and left unaltered.
- Step 3: If the processed pixel p_{xy} holds 0 or 255 i.e. ($p_{xy}=0$ or $p_{xy} =255$) then pixel p_{xy} is considered as corrupted pixel. Convert 2D array into 1D array. Sort the 1D array which is assumed as S_{xy} .
- Step 4: Initialize two counters, forward counter (F) and reverse counter (L) with 1 and 9 respectively. When a 0 or 255 is encountered inside the window F is increased by 1 or L is decremented by 1 respectively. When pixel is noisy there happens to be two possible cases. Check for the number of corrupted pixel inside the current processing window. It is denoted as count.
 - Case 1) if the count value ≤ 3 inside the current processing window then the corrupted pixel is replaced with median of unsymmetrical trimmed output.
 - Case 2) if the count value is greater than 3 then the noisy pixel is replaced with mean of Fth and Lth value of the rank ordered unsymmetrical trimmed output. If entire pixels inside the processing window are 0 or 255 then pixel value is retained considering it as texture.
- Step 5: Steps 1 to 4 is repeated until all pixels of the entire image is processed.

2.4 Insight of DBUTVF

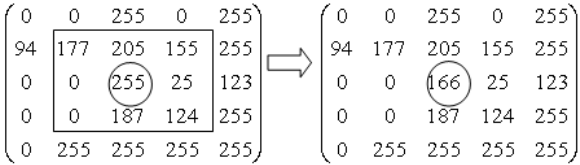
The processed pixel is checked for low (0) or high(255) values of the gray level values. This process is done on entire pixels in the image. The large matrix refers to image and values enclosed inside a rectangle is considered to be the current processing window. The element encircled refers to processed pixel. Step 2 is illustrated in case (a). Step 3 and 4 are visualized along with the case 1) in case (b) .Case II is briefed in case (c.1) and (c.2).



Corrupted image

Restored image

Case (a): In the above illustration the processed pixel is checked for $0 < p_{xy} < 255$. Here in the discussed example processed pixel is 106. Hence processed pixel is not 0 or 255. So pixel is considered as noise free and pixel is unaltered.



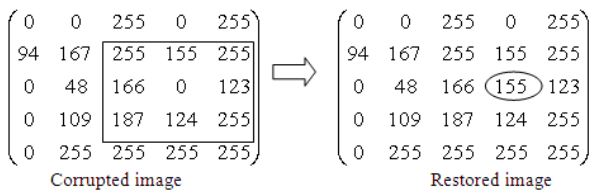
Corrupted image

Restored image

Case (b): In the selected window the processed pixel holds 255 (or 0). So the processed pixel is considered as noisy. Initialize forward counter F=1 and reverse counter L=9. Convert the 2D array into 1D array and sort the converted array. F and L counter moves in forward and reverse directions respectively. Check for number of noisy pixel inside the window. Every time a 0 or 255 is encountered inside the window the counter value increments by 1.

Unsorted array: 177 0 0 205 255 187 155 25 124
Check for Count=3 (so case 1 is used)
Sorted array S_{xy} 0 0 25 124 155 177 187 205 255

Here the count value is 3 so replace the corrupted pixel with median of rank ordered symmetrical trimmed output. The case is illustrated as follows. Now check for the presence of 0 or 255 in the sorted array. Every time a 0 is detected F is incremented by 1 and if 255 is detected L is decremented by 1. In the above example there is two 0 and one 255. Hence F is incremented by two times and L is decremented by one time. Now finally F is holding 3 and L is holding 8. Now the corrupted pixel is replaced with median of the rank ordered unsymmetrical trimmed output i.e. corrupted pixel is replaced by median (25,124,155,177,187,205) = 166



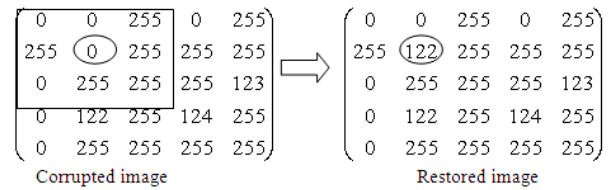
Corrupted image

Restored image

Case (c.1): This case works if the entire pixel inside the current window is either pepper (0) or salt (255). Initialize F=1 and L=9 and convert the elements of 2D window into 1D. Sort the 1D array. Check for count values by comparing all the pixel elements with 0 or 255.

Unsorted 1D array: 255 166 187 155 0 124 255 123 255
Check for Count=4 (so case 2 is used)
Sorted 1D array S_{xy} : 0 123 124 155 166 187 255 255 255

Now the F counters propagates forward and L in reverse direction. Finally F and L hold 2 and 6 respectively. Hence the noisy pixel is replaced by $(S(2) + S(6))/2 = (123+187)/2 = 155$.



Corrupted image

Restored image

Case (c.2): If the current processing window has combination of 0 and 255 only. The counter count will give the number of noisy pixels as 9. Hence output pixel is replaced by mean of Fth and Lth value of the rank ordered unsymmetric trimmed output. Initialize F=1 and L=9 and convert the elements of 2D window into 1D. Sort the 1D array.

Unsorted 1D array: 0 255 0 0 0 255 255 255 255
Count=9 (so case 2 is used).
Sorted 1D array S_{xy} : 0 0 0 0 255 255 255 255 255

Now the F counters propagates forward and L in reverse direction. Finally F and L hold 5 and 4 respectively. Hence the noisy pixel is replaced by $(S(5) + S(4))/2 = (255+0)/2 = 122$.

3. SIMULATION RESULTS

The Quantitative performance of the proposed algorithm is evaluated based on Peak signal to noise ratio (PSNR), Mean Square Error (MSE) and Image Enhancement Factor (IEF) which is given in equations 1, 2, 3 respectively.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (1)$$

$$MSE = \frac{\sum_i \sum_j (r_{ij} - x_{ij})^2}{M \times N} \quad (2)$$

$$IEF = \frac{\left(\sum_i \sum_j (n_{ij} - r_{ij})^2 \right)}{\left(\sum_i \sum_j (x_{ij} - r_{ij})^2 \right)} \quad (3)$$

Where r refers to Original image, n gives the corrupted image x is denotes restored image, $M \times N$ is the size of Processed image. The existing algorithms used for the comparison are SMF, AMF, PSMF, DBA, CUDBMPF, and MDBUTMF. The qualitative performance of the proposed algorithm is tested on various images such as Lena, Cameraman, Baboon, Barbara, girl, pepper etc (Images are chosen as per the details of the image). Quantitative analysis is made by varying noise densities in steps of ten from 10% to 90% on low detail, given in Table II, III, IV, V figure 2,5 respectively. All the simulation is done in dual CPU E2140@1.6Ghz with 1GB RAM capacity. It is vivid from the tables and graphs that the proposed algorithm suppresses noise at low and high noise densities. Hence the DBUTVF has high PSNR and IEF when compared to other algorithms. The qualitative aspect of the DBUTVF against various algorithms for noise densities (10% to 90%) for Lena (low detail image) and edge preservation of Baboon (high detail image) at 85% is shown in figure 3 and 4 respectively. The canny edge detection method is used to detect the edges. It was found that DBUTVF preserved the global and local edges up to 85% noise where filters such as SMF, AMF, and PSMF fail. The cascaded algorithm too fails to preserve local edge. The DBA algorithm induces streaking

TABLE II
PERFORMANCE OF VARIOUS ALGORITHMS AT DIFFERENT NOISE DENSITIES FOR PSNR

Noise in %	PSNR IN DB						
	SMF	AMF	PSM	DBA	CUD BMP F	MDB UTM F	PA
10	34.9	39.3	38.8	39	32.3	43.12	45.3
20	30.3	36.9	33.4	36.8	32.1	41.22	41.6
30	23.9	34.6	29.4	35.8	31.83	37.9	38.9
40	19.0	32.2	25.4	33.2	31.48	36.4	37.2
50	15.9	27.3	25.3	31.4	31.11	34.3	35.3
60	12.3	21.6	21.2	29.6	30.38	32.15	33.9
70	10.0	16.6	9.94	27.8	30.21	29.61	33.2
80	8.1	12.7	8.1	25.5	29.35	26.83	30.3
90	6.6	9.86	6.68	21.8	27.4	22.45	27.6

TABLE III
PERFORMANCE OF VARIOUS ALGORITHMS AT DIFFERENT NOISE DENSITIES FOR IEF

Noise in %	IEF						
	SMF	AMF	PSM	DBA	CUD BMP F	MDB UTM F	PA
10	89.0	246.8	219	230	48.9	630	969
20	61.0	281.3	124	276	40.0	552	818
30	21.4	254.4	74.5	331	130.7	565	674
40	9.18	192.9	40.1	242	160.3	509	600
50	4.95	78.3	39.6	199	185.1	384	496
60	2.95	25.07	19.1	157	187.8	282	423
70	2.03	9.17	1.9	123	210.3	183	333
80	1.49	4.37	1.4	81.5	196.5	110	248
90	1.18	2.5	1.1	39.1	141.5	45	150

TABLE IV
PERFORMANCE OF VARIOUS ALGORITHMS AT DIFFERENT NOISE DENSITIES FOR MSE

Noise in %	MSE						
	SMF	AMF	PSM	DBA	CUD BMP F	MDB UTM F	PA
10	20	7	8	8	38	2	1
20	60	13	29	13	92	4	4
30	259	22	74	16	42	8	8
40	814	38	185	30	46	14	12
50	1877	118	187	46	50	23	18
60	3776	443	484	70	59	39	26
70	637	1421	600	105	61	69	39
80	9945	3413	1000	182	75	134	59
90	14179	6708	1396	427	118	369	111

TABLE V
PERFORMANCE OF VARIOUS ALGORITHMS AT DIFFERENT NOISE DENSITIES FOR TIME

Noise in %	TIME IN SECONDS						
	SMF	AMF	PSM	DBA	CUD BMP F	MDB UTM F	PA
10	2	23	3.4	27	18.4	12	10
20	2	22	2.1	27	18.9	10	9
30	1.6	22	2.1	27	18.1	8	9
40	1.5	22	2.3	28	18	9	8
50	1.4	22	2.4	28	18	10	9
60	1.5	21	1.7	29	17.9	11	8
70	2.5	21	1.7	29	18.1	11	9
80	2.1	20	1.6	30	18	12	10
90	1.4	19	1.7	30	18.3	14	10

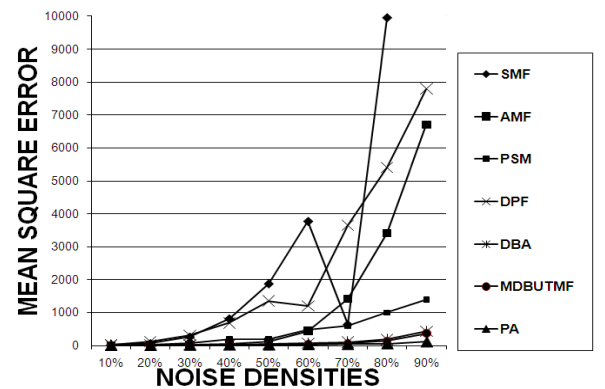
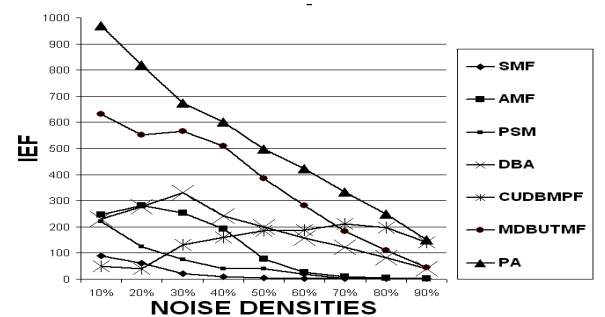
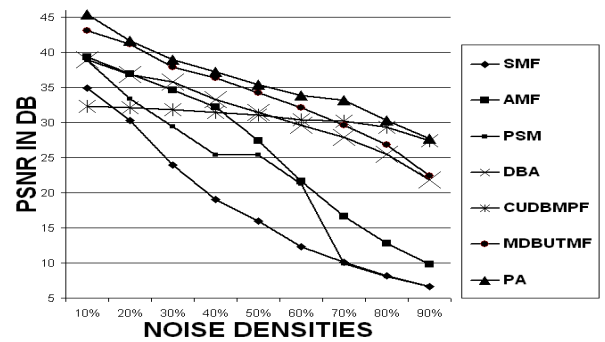


Figure 2. Performance comparison of PSNR, IEF, MSE (from the top) at various noise densities for low detail Lena image

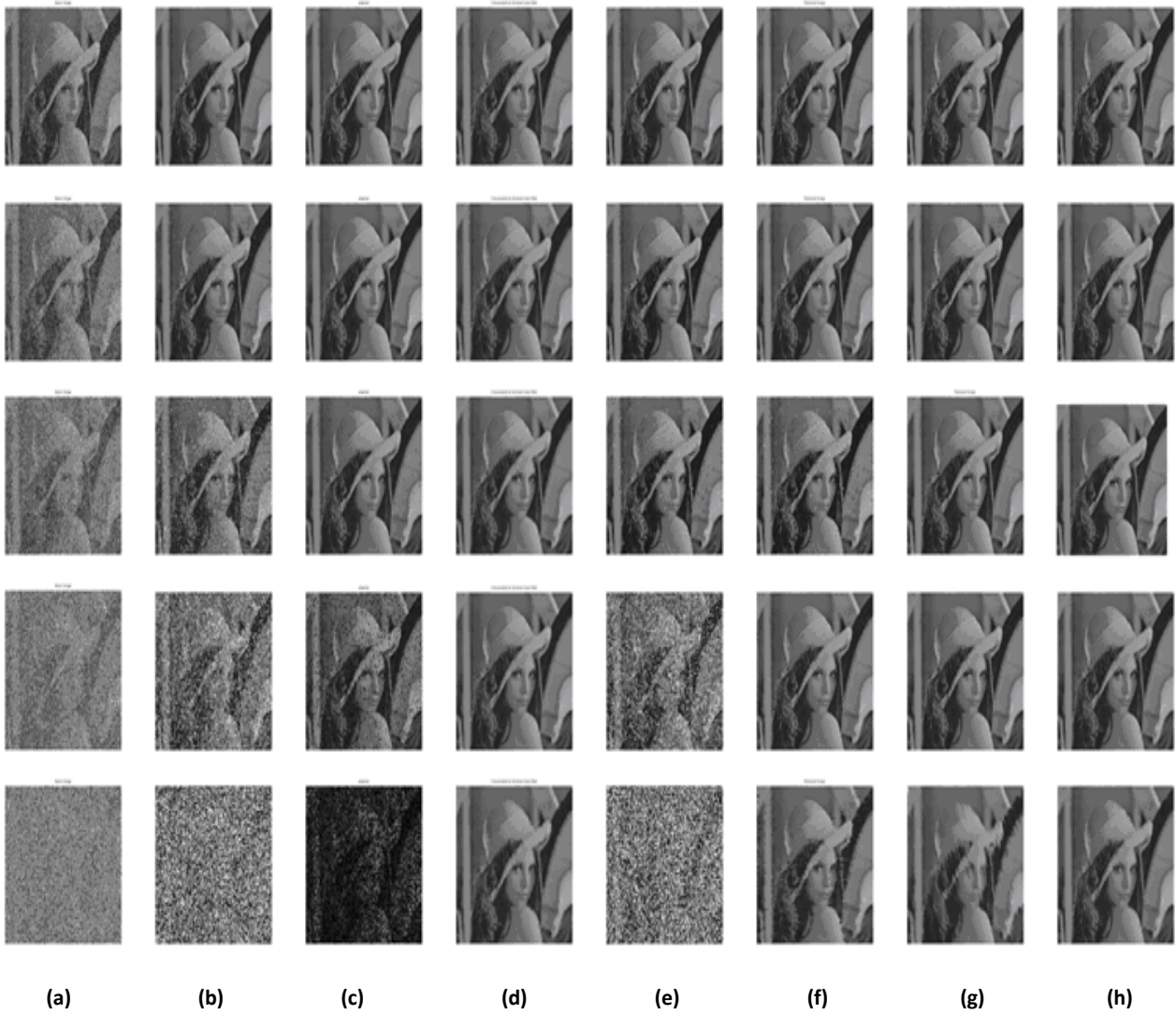


Figure 3. Performance of various filter of Lena image corrupted by 10%,30%,50%,70%,90% Salt and pepper noise in row1,2,3,4,5 respectively.(a) fixed value salt and pepper noise(b) output of SMF (c) output of AMF (d) output of CUDBMPF (e) output of PSMF (f)output of DBA (g) output of MDBUTMF (h) output of proposed algorithm

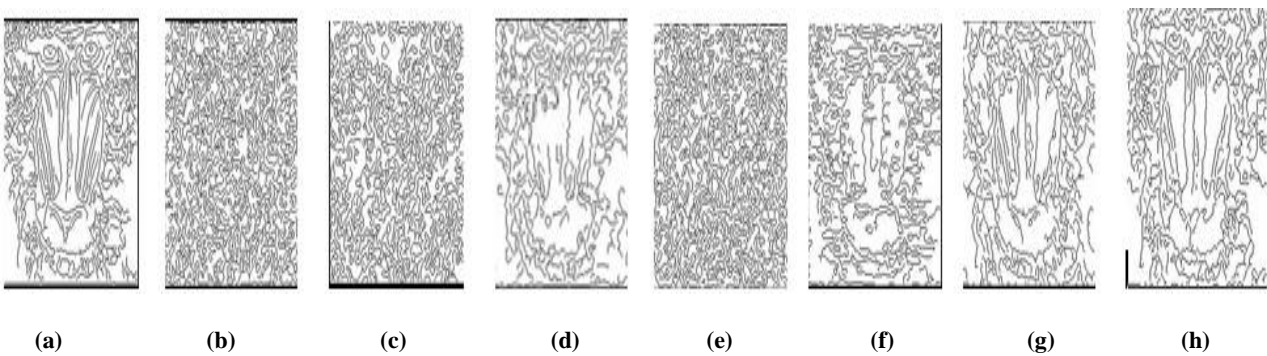


Figure 4. Edge preserving capability of various algorithms on baboon image for 85% salt and pepper noise (a) edges of the original image (b) output of SMF (c) output of AMF (d) output of CUDBMPF (e) output of PSMF (f)output of DBA (g) output of MDBUTMF (h) output of Proposed algorithm

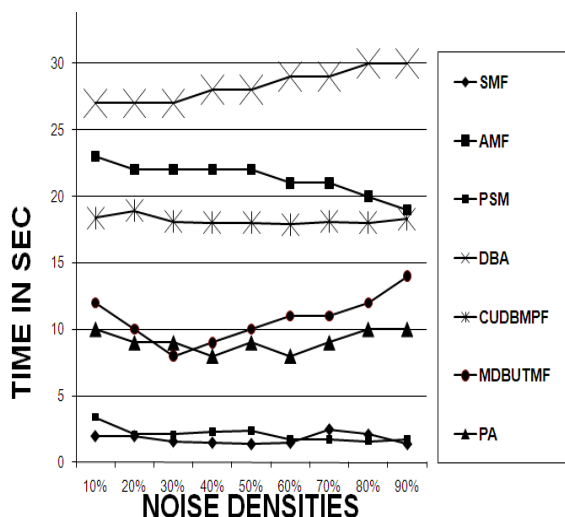


Figure 5. Performance comparison of Time (from the top) at various noise densities for low detail Lena image

as shown in sixth column of figure 3. Hence the fine details are not preserved. Seventh column of figure 3 gives the edge preservation capability of MDBUTMF. For the baboon image MDBUTMF smudges the edge poor edge preservation. The cascaded algorithm used two level algorithms to achieve edge preservation. The proposed algorithm preserves global edge at very high noise densities. From figure 2 it was vivid that the proposed algorithm has a low mean square error value. The proposed algorithm that employs Snake like modified shear sorting is faster on comparison with other existing and other conventional algorithms. The other algorithms that are employed such as SMF, AMF, PSMF, CUDBMPF, MDBUTMF uses inbuilt sorting function of mat lab. The Decision based algorithm uses median finding algorithm. Hence the window is not completely sorted. Hence from figure 5 we find the mesh based sorting proposed is faster. The proposed snake like algorithm requires only three comparators for arranging the three elements in increasing order and hence the time complexity of the algorithm is $4n+2\sqrt{n}$. This snake like algorithm is intended to rank order the pixel inside the current processing window.

4. CONCLUSION

A simple decision based algorithm (DBUTVF) using a Mesh based snake like modified shear sorting, having a fixed 3x3 window is proposed which gives excellent noise suppression capabilities by preserving edges in images corrupted by salt

and pepper noise as high as 85%. The Proposed algorithm uses 3x3 fixed window for increasing noise densities. The proposed algorithm outclasses other existing filters when compared to existing filters both in quantitative and qualitative aspects. The proposed algorithm would also require minimal hardware for implementation.

5. REFERENCES

- [1] J. Astola and P. Kuosmaneen, Fundamentals of non linear Digital Filtering. Boca Raton, FL: CRC, 1997.
- [2] H. Hwang and R. A.Hadded, "Adaptive median filter: New algorithms and results," IEEE transaction on image processing., vol 4,no 4,pp 499-502
- [3] P. E. Ng and K. K. Ma, "A switching median filter with boundary discriminative noise detection for extremely corrupted images," IEEE Transactions on image processing, vol.15, no.6, pp. 1506-1516,June 2006.
- [4] S. Zhang and M.A. Karim, "A new impulse detector for switching median filters," IEEE Signal processing letters, vol.9, no.11, pp. 360-363, November 2002.
- [5] K.S. Srinivasan and D.Ebenezer, "A new fast and efficient decision based algorithm for the removal of high density impulse noise," IEEE Signal processing letters, vol.14, no.3, pp.189-192, March 2007.
- [6] S.Balasubramanian, S.Kalishwaran, R.Muthuraj, D.Ebenezer, V.Jayaraj "An efficient Non linear cascade filtering algorithm for removal of high density salt and pepper noise in image and video sequence", Intl Conf on control, Automation, communication and Energy Conservation,2009,
- [7] K.Aiswarya, V .Jayaraj, and D.Ebenezer, "A new and efficient algorithm for the removal of high density salt and pepper noise in images and videos," in second international conference on computer modeling and simulation, 2010, pp.409-413.
- [8] S. Esakkirajan, T. Veerakumar, Adabala N. Subramanyam and C.H. Prem Chand, "Removal of high density Salt and pepper noise through modified decision based Unsymmetrical trimmed median filter." IEEE Signal processing letters, Vol. 18,no.5, May 2011.
- [9] K.Vasanth, S.Karthik, "Unsymmetrical Trimmed median as Detectors for salt and pepper noise Removal", National Conference on signal and image processing-NCSIP2012, Gandhi gram rural university, February 2012, pages 31-35.
- [10] K. K.Vasanth, S.Karthik, Sindu Divakaran "Removal of Salt & Pepper Noise Using Unsymmetrical Trimmed Variants as Detector", European journal of Scientific Research, Vol.70 No.3 (2012), pp. 468-478.