ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2010.0009

A SWITCHING ALGORITHM USING MODIFIED SELECTION SORT FOR THE REDUCTION OF IMPULSE NOISE

K. Vasanth[1] and S. Karthik[2]
[1]*Sathyabama University, Chennai, Tamilnadu, India*
Email: vasanthece_k@yahoo.co.in
[2]*Cognizant Technology solutions, Chennai, Tamilnadu, India*
Email: skarthick76@gmail.com

Abstract
A novel algorithm is proposed for the restoration of images corrupted by impulse noise; this work aims at a novel filter whose window size is fixed (3X3) for all noise densities. The new algorithm significantly produces better image quality than standard median filter (SMF), adaptive median filters (AMF), and center weighted median filter (CWMF) and threshold decomposition filter (TDF). Unlike the other filters, the proposed algorithm computes median if and only if there is a corrupted pixel and replaces it by the median value, if the median is corrupted then average of uncorrupted pixels in the current processing window is replaced else preprocessed pixel value is replaced. The proposed method removes the noise effectively even at noise level as high as 85% and preserves the edges without any loss up to 80% of noise level. The proposed algorithm (PA) is tested on different images and is found to produce better results in terms of the qualitative and quantitative measures.

Keywords:
Impulse Noise, Median Filter, Adaptive Median Filter, Decision Based Filter, Weighted Filter

1. INTRODUCTION

Visual information transmitted in the form of digital images is becoming a major method of communication in the modern age, but the image obtained after transmission is often corrupted with impulse noise due to a noisy sensor or channel transmission errors. Impulse noise randomly and sparsely corrupts pixels to two intensity levels—relatively high or relatively low, when compared to its neighboring pixels. The goal of impulse noise removal is to suppress the noise while preserving the integrity of edge and detail information. To this end, nonlinear techniques have been found to provide more satisfactory results in comparison with linear methods. Most of the classical linear digital image filters removes the noise but degrades the quality of an image. This has led researchers to use non-linear filters. A class of widely used nonlinear digital filters is median filters. Median filters are known for their capability to remove impulse noise as well as preserve the edges. Standard median (SM) filter was used by the researchers to remove impulse noise and it achieved reasonably good performance for lower noise densities. SM filter exploits the rank-order information (i.e., order statistics) [3] of the input data by replacing the processed pixel with median of the re-ordered input to remove impulse noise. Since its introduction by the researches, SM filter has been widely studied and extended to various approaches such as weighted median (WM) and center weighted median (CWM) filters. The WM filter used a set of weights to control the filtering performance in order to preserve more signal details than existing SM filtering can accomplish. CWM filter is a special case of the WM filter, where only the center pixel of the filtering window has a weighting factor. In all of the above discussed method due to the application of median values to all the pixels irrespective of pixel is noisy or not noisy, filters are effective only for low noise densities. At high noise densities, the SMF exhibits blurring when the window sizes are increased and not capable of suppressing noise for small window sizes [1], [2]. In addition, when the percentage of noise is large these filters are prone to edge jitter [2],[6],[15]. Consequently, the effective removal of impulses is often at the expense of blurred and distorted features. Ideally, the filtering should be applied only to corrupted pixels while leaving uncorrupted pixels intact. Applying median filter unconditionally across the entire image as practiced in the conventional schemes would inevitably alter the intensities and remove the signal details of uncorrupted pixels. Therefore, a noise-detection process to discriminate between uncorrupted pixels and the corrupted pixels prior to applying nonlinear filtering is highly desirable. Filters such as AMF, decision-based, or switching median filters have been proposed with this objective. The idea is to identify possible noisy pixels and replaced by median value or it's variant while uncorrupted pixels left unchanged [3]. The performance of AMF is good at lower noise density levels, due to less number of corrupted pixels that are replaced by the median values with small window sizes. At higher noise densities, number of noisy pixels increases hence the number of replacements of corrupted pixel increases considerably. In AMF depending upon the noise densities the size of the window increases and will provide better noise removal performance; however, the similarities of corrupted pixel values and replaced median pixel values are less. As a result, the edges are blurred significantly. The main drawback of decision-based or switching median filter is that defining a good decision measure is difficult, because the decision is usually based on a predefined threshold value. An additional drawback is that the noisy pixels are replaced by some median value in their neighborhood without taking into account local features such as possible presence of edges. Hence, details and edges are not recovered satisfactorily, especially when the noise level is high.[12]-[13]. Decision based adaptive algorithm, such as adaptive filter [8], Tri-state median filter [7], Progressive switched median filter [9] noise adaptive soft switching median filter [5], [10], Detail preserving filter showed great deal of noise removal for low density and medium density noises with increase in its window size. Decision based modified sorting algorithm [8] degrades the image quality as the noise density increases. Since the neighborhood value is used as a replacement for the median under the condition median being noisy. This leads to streaks in images. The drawback of Chan & Nikolova method [8] is that the size of the window mask is very large which considerably requires large computational time and hence a complex hardware.

2. PROPOSED TECHNIQUE

The Proposed algorithm performs efficiently even for images corrupted by noise densities as high as 85% and shows significantly better image quality than the existing decision based algorithms with a fixed window size of 3X3. The proposed algorithm is given as follows:

Step 1: A 2-D window of size 3*3 is selected. Assume the pixel to be processed is P(X, Y).

Step 2: The pixel intensities of the window considered are converted into an 1D array of size 9.

Step 3: The pixel with maximum intensity is propagated to the final array position of the input data by the process of swapping as shown in figure 1. This gives P_{max}.

Step 4: The pixel with minimum intensity is propagated to the last but one position just next to P_{max} of the array by the process of swapping the array elements, excluding P_{max}. This gives P_{min} as shown in Fig.1.

Step 5:

Case 1: P(X, Y) is an uncorrupted pixel,
if P_{min} < P(X, Y) <P_{max}; the pixel being processed is left unchanged. This case does not involve the computation of the median.
Otherwise,
P(X, Y) is a corrupted Pixel.
The median is computed only when the processed pixel is noisy.

Case 2: If P(X, Y) is a corrupted pixel, the median is computed as follows;
To find the median P_{med}, swap the remaining unsorted array elements obtained from step 4, excluding P_{max} and P_{min} for four passes as shown in Fig.2. After each pass, the smallest element encountered in the current pass will reside in the last position traversed. So each pass can be one step shorter than the previous pass, instead of every pass continuing to traverse all the elements at the end, which are already in their final positions and will not move in any case. After the 4th pass, the pixel in the 4th position will give the median of the window as illustrated in the Fig.2. The corrupted pixel is replaced by its median value. For high noise densities the manipulated median may also be noisy. So check the calculated median is noisy or not.
If Pmin < Pmed < Pmax and 0 < Pmed < 255,
then Pmed is a uncorrupted pixel, replace on the processed pixel.

Case 3: If Pmin < Pmed < Pmax is not satisfied or 255<Pmed<0, then Pmed is a noisy pixel. In this case, the P(X, Y) is replaced by the average of the non- noisy pixels in the window considered. These pixels must satisfy the condition, min<pixel intensity<max. Only those pixels satisfying the above condition are considered as non noisy or noise free pixel of the current processing window. When no non noisy pixel is presented then go to Case 4.

Case 4: If there are no uncorrupted pixels in the window, replace the corrupted pixel with the neighborhood pixel.

Step 5: Steps 1 to 5 are repeated until the processing is completed for the entire image. The proposed algorithm is illustrated in figure 1, 2, 3.

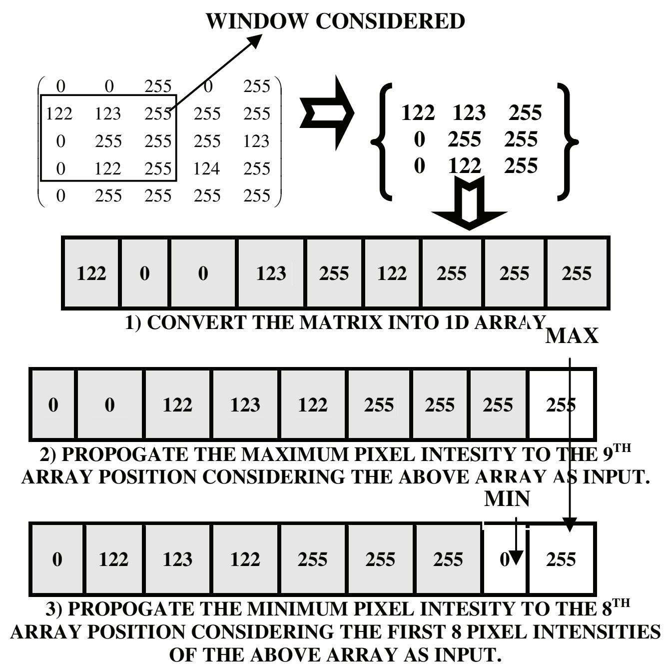


Fig.1. Illustration of the proposed algorithm to find P_{min} and P_{max}

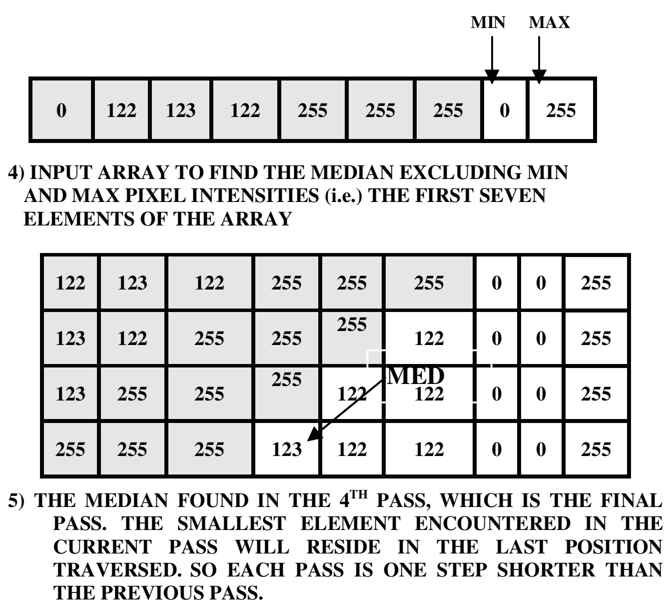


Fig.2. Illustration of the proposed algorithm to find P_{med}.

Case 1:

$$\begin{pmatrix} 0 & 0 & 255 & 0 & 255 \\ 122 & 123 & 255 & 255 & 255 \\ 0 & 255 & 255 & 255 & 123 \\ 0 & 122 & 255 & 124 & 255 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix}$$

(0 < 123 < 255) center pixel is not noisy so the same value is retained

Case 2:

$$\begin{pmatrix} 0 & 0 & 255 & 0 & 255 \\ 122 & 123 & 255 & 255 & 255 \\ 0 & 255 & 255 & 255 & 123 \\ 0 & 122 & 255 & 124 & 255 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix}$$

(0 < 255 < 255) center pixel is noisy, so the median is found 123 (0 < 123 < 255). Median is not noisy. Replace processed pixel by median value

Case 3:

$$\begin{pmatrix} 0 & 0 & 255 & 0 & 255 \\ 122 & 255 & 255 & 255 & 255 \\ 0 & 255 & 255 & 123 & 123 \\ 0 & 122 & 255 & 124 & 255 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix}$$

(0 < 255 < 255) center pixel is noisy, so the median is found 255 (0 < 255< 255). Median is also noisy. Replace processed pixel by arithmetic mean of non noisy pixel inside the window. Here only one non noisy pixel i.e 123

Case 4:

$$\begin{pmatrix} 0 & 0 & 255 & 0 & 255 \\ 122 & 123 & 123 & 255 & 255 \\ 0 & 255 & 255 & 255 & 255 \\ 0 & 122 & 255 & 124 & 255 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix}$$

(0 < 255 < 255) center pixel is noisy, so the median is found 255 (0 < 255< 255). Median is also noisy. There is no non noisy pixel in the window. Replace by left neighborhood from the last processed value

Fig.3. Methodology of Proposed Algorithm

3. SIMULATION RESULTS

The performance of the algorithm is tested with the gray scale image lena.gif, with their dynamic range of values [0,255]. In the simulation, images will be corrupted by fixed impulse noise (salt and pepper noise), where 255 represents "salt" and "0" represents the "pepper" noise with equal probability. To perform this MATLAB inbuilt function was used to model fixed value impulse noise [4]. The noise levels are varied from 10% to 90% with increments of 10%, and the results are tabulated in Table's I-III. Similar performance was achieved for images corrupted by random impulse noise. The random impulses were added to the Lena.gif image and the restoration performances are measured quantitatively and qualitatively by peak signal-to-noise ratio (PSNR) and image enhancement factor (IEF) and the results are tabulated in Table's I-VI. The results depict the significant performance of the algorithm. The mean squared error (MSE) of the filtered image is given by

$$\text{MSE}= \frac{\left(\sum_{i,j=1}^{MXN} S_{ij} - X_{ij}\right)^2}{MXN}$$

The PSNR in decibels (db) is computed by using

$$\text{PSNR} = \frac{\left[\left(10 * \log(255)^2\right)\right]}{MSE}$$

IEF is defined as the ratio of Mean Square Error before and after filtering the signal.

$$\text{IEF}= \frac{\left(\sum_{i,j=1}^{256} r_{ij} - s_{ij}\right)^2}{\left(\sum_{i,j=1}^{256} x_{ij} - r_{ij}\right)^2}$$

Where, s_{ij} is the original image
r_{ij} is the corrupted image
x_{ij} is the restored image
MXN is the size of the image

The PSNR, IEF, and CPU computation time in seconds are calculated for the proposed algorithm and a comparison of performance with various filters such as SMF, AMF, TDF, CWMF and modified decision based filter are shown in Table's I-VI. The Proposed algorithm (PA) has superior performance in comparison with other decision- based median and switching filters. The important aspect of the PA is that it uses a fixed 3X3 window for processing. The PA leads to simple physical realization as well as much smaller computation time. MATLAB 7.0(R14) on a PC equipped with 2-GHz CPU and 3GB of RAM memory has been employed for the evaluation of computation time of all algorithms. The Plot for PSNR of various algorithm Vs various noise densities is given in figure 4. Fig. 5 and 6 gives the plot between IEF and computation time of various algorithm for increasing noise densities. All the test results were performed on the lena.gif image corrupted by fixed impulse noise. Fig. 6-8 shows that the PA performs significantly better when compared with SMF, AMF, TDF, CWMF and modified decision based filter for various values of noise density. The table V and VI illustrates the performance of proposed algorithm on random impulse noise. Fig. 7 & 8 gives the performance of various algorithms for different fixed and random impulse noises.

Table.1. PSNR for various filters for lena.gif (512X512) image corrupted by fixed impulse noise at different noise densities

ND	SMF	TDF	AMF	CWF	MSSD	PA
10%	35.0929	29.744	29.48	36.2179	38.6757	39.3
20%	29.8711	27.9068	28.3	34.6297	37.1764	38.3
30%	23.9786	23.4899	27.1	32.8655	35.1881	37.4586
40%	19.1703	19.119	25.55	31.4116	33.5286	34.6654
50%	15.2982	15.2846	24.04	30.2603	31.4596	32.8978
60%	12.3881	12.4011	21.07	29.0322	29.8461	31.6531
70%	10.1001	10.0593	16.1	27.8673	27.753	30.3099
80%	8.1631	8.1655	11.6	25.8155	25.2832	28.0041
90%	6.6517	6.6596	8.002	22.2188	21.8216	24.2658

Table.2. IEF for various filters for lena.gif (512X512) image corrupted by fixed impulse noise at different noise densities

ND	SMF	TDF	AMF	CWF	MSSD	PA
10%	91.7773	35.1923	25.44	230.5137	210.3562	252.878
20%	55.3248	41.4499	38.89	297.6942	299.3415	334.7307
30%	21.5101	20.1159	43.32	293.8959	282.4063	303.6001
40%	9.4291	9.4188	41.24	261.9374	257.7399	290.7325
50%	4.8583	4.8634	36.69	240.0405	199.5982	279.0028
60%	2.9717	2.9837	21.9	211.2649	165.8077	250.5573
70%	2.0429	2.0253	8.14	176.2735	119.7057	215.026
80%	1.4992	1.4984	3.289	118.8092	77.3397	144.4504
90%	1.1899	1.1903	1.622	52.9698	39.1291	68.8089

Table.3. COMPUTATION TIME for various filters for lena.gif (512X512) image corrupted by fixed impulse noise at different noise densities

ND	SMF	TDF	AMF	CWF	MSSD	PA
10%	1.031	115.969	90.44	13.031	7.906	6.312
20%	1.078	142.531	89.9	13.063	7.594	6.328
30%	1.031	160.094	90.3	13.469	7.594	6.332
40%	1.047	168.547	91.2	13.469	7.672	6.343
50%	1.047	177.578	91.2	13.656	7.718	6.437
60%	1.047	180.813	90.9	13.953	7.718	6.61
70%	1.063	183.187	90.7	14.313	7.703	6.656
80%	1.078	186.187	90.7	14.407	7.61	6.687
90%	1.11	185.078	90.1	14.594	7.719	6.721

Table.4. PSNR for various filters for lena.gif (512X512) image corrupted by random impulse noise at different noise densities

SALT	PEPPER	SMF	AMF	CWF	TDF	MSSD	PROPOSED
3	6.7	23.4378	28.966	22.9664	22.7609	36.7614	37.6994
6.7	3	23.2438	27.7259	22.891	22.5638	36.2456	37.6626
12	7	18.0719	21.3396	17.979	17.8402	34.2459	36.0823
4	15.3	18.4597	21.8632	18.3143	18.2458	33.8038	35.5537
21	7.2	15.494	17.3053	15.5174	15.3754	31.2573	34.0771
18	9.8	15.0994	15.9875	15.2097	15.0067	29.7089	33.0908
32	5.4	14.0134	15.4691	14.0591	13.9244	30.2713	33.3652
11.6	28	14.2141	15.6041	14.2952	14.1456	30.2284	33.3254
30	14	14.0016	15.783	13.9949	13.9114	29.6584	33.025
25	18.7	13.8874	15.5575	13.9106	13.8103	29.6573	32.8097
43	9.7	13.5794	14.7935	13.6483	13.5085	29.3586	32.5899
27	24	31.9389	15.2472	13.6405	13.6	28.8704	31.9389
19	41.3	34.5503	18.6608	16.4581	16.3995	32.489	34.5503
63	2.5	36.9909	24.4849	21.1283	20.8109	35.3786	36.9909
34	36.9	29.3033	11.6442	11.0979	10.9132	25.6572	29.3033
42	22.03	30.8287	14.0993	12.9936	12.8544	27.7515	30.8287

Table.5. IEF for various filters for lena.gif (512X512) image corrupted by random impulse noise at different noise densities

SALT	PEPPER	SMF	AMF	CWF	TDF	MSSD	PA
3	6.7	2.6619	3.0541	2.3432	2.3386	56.1448	69.6807
6.7	3	2.3222	2.5531	2.1411	2.0907	46.354	64.2376
12	7	1.7296	1.9158	1.693	1.6742	71.6723	109.3949
4	15.3	1.8097	1.9906	1.7502	1.7465	61.9483	92.6879
21	7.2	1.4616	1.4893	1.4695	1.4348	55.1006	1054715
18	9.8	1.2564	1.2416	1.2888	1.2358	36.3163	79.12
32	5.4	1.4032	1.4007	1.4181	1.3839	59.2812	120.8676
11.6	28	1.3992	1.3882	1.4256	1.3836	55.8887	114.0316

30	14	1.5055	1.5063	1.5032	1.4861	55.3805	120.2306
25	18.7	1.4704	1.4697	1.4783	1.4528	55.5182	114.7284
43	9.7	1.4411	1.3805	1.4641	1.4262	54.5259	114.7423
27	24	106.5867	1.511	1.577	1.5638	52.5808	106.5867
19	41.3	95.2027	1.5662	1.5081	1.4778	59.2266	95.2027
63	2.5	66.1443	1.9273	1.7149	1.6424	45.6317	66.1443
34	36.9	85.3627	1.2097	1.2904	1.2391	36.8694	85.3627
42	22.03	86.3374	1.354	1.4213	1.3821	42.5091	86.3374

Table.6. COMPUTATION TIME for various filters for lena.gif (512X512) image corrupted by random impulse noise at different noise densities

SALT	PEPPER	SMF	AMF	CWF	TDF	MSSD	PA
3	6.7	1.453	347.422	19.671	80.799	7.531	6.344
6.7	3	1.203	347.75	24.454	85.781	7.578	6.359
12	7	1.203	348.672	18.109	95.485	7.657	6.391
4	15.3	1.234	416.609	14.312	82.203	7.578	6.375
21	7.2	1.172	449.625	16.625	106.678	7.703	6.5
18	9.8	1.172	497.703	13.703	100.953	7.61	6.453
32	5.4	1.203	532.484	11.328	120.672	7.625	6.485
11.6	28	1.219	377.578	12.062	97	7.594	6.532
30	14	1.188	354.688	14.515	116.219	7.562	6.469
25	18.7	1.172	379.688	11.266	119.828	7.562	6.484
43	9.7	1.219	372.813	11.283	124.343	7.531	6.469
27	24	6.484	348.688	11.5407	120.407	7.563	6.484
19	41.3	6,375	361.875	11.141	104.282	7.576	6.375
63	2.5	6.391	353.313	11.14	86.875	7.641	6.391
34	36.9	6.718	350.234	12.047	124.422	7.516	6.718
23	51.59	6.625	379	11.797	107.75	7.625	6.625

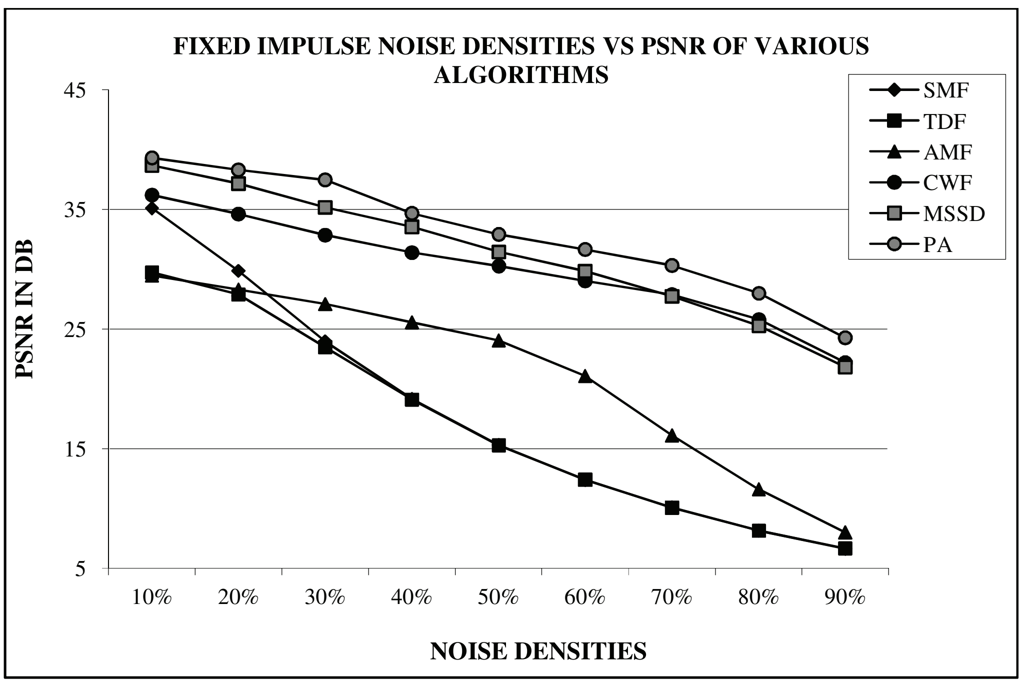


Fig.4. Various noise densities versus PSNR

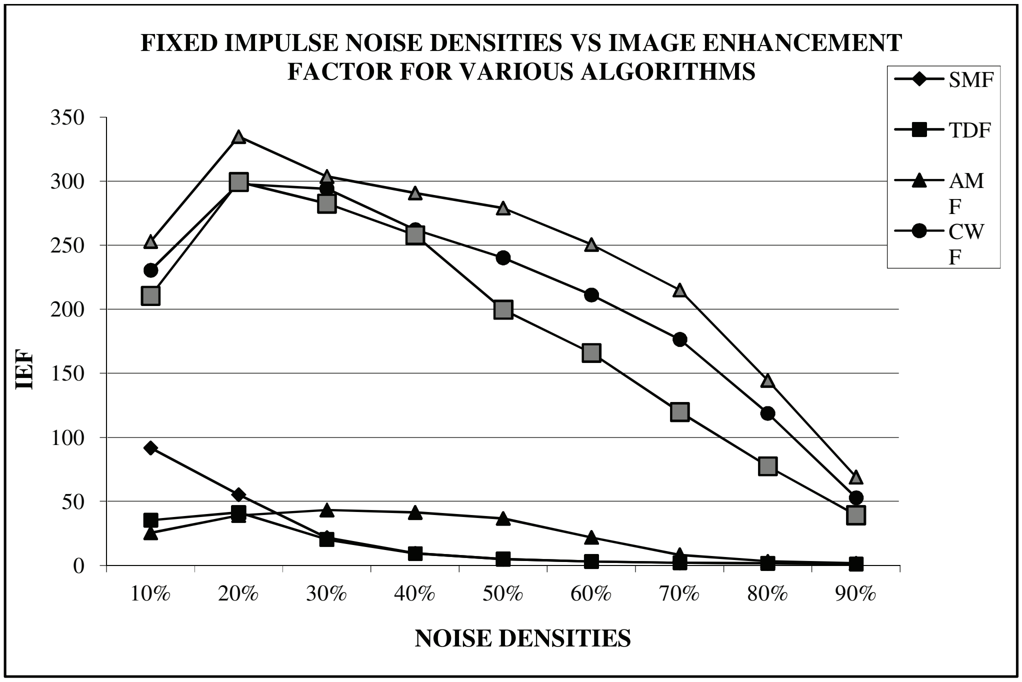


Fig.5. Various noise densities versus IEF

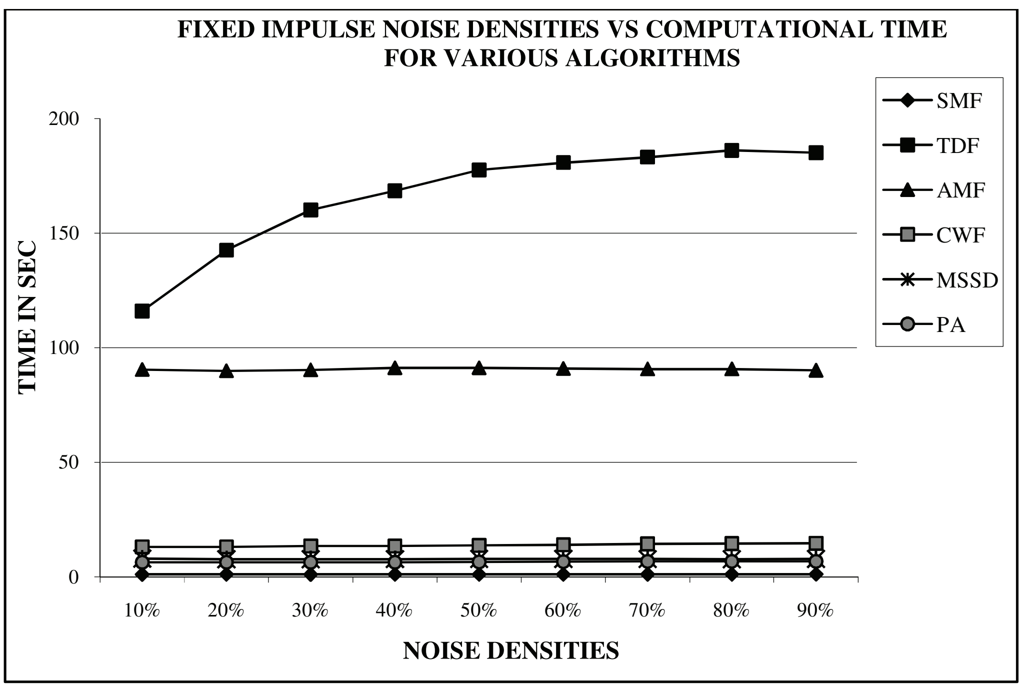
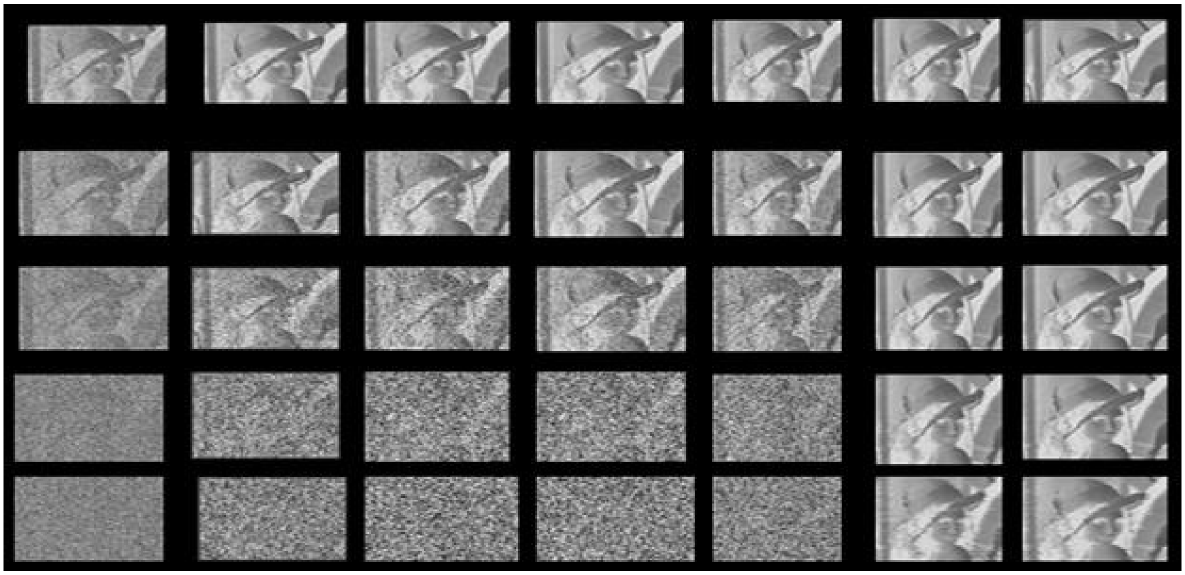
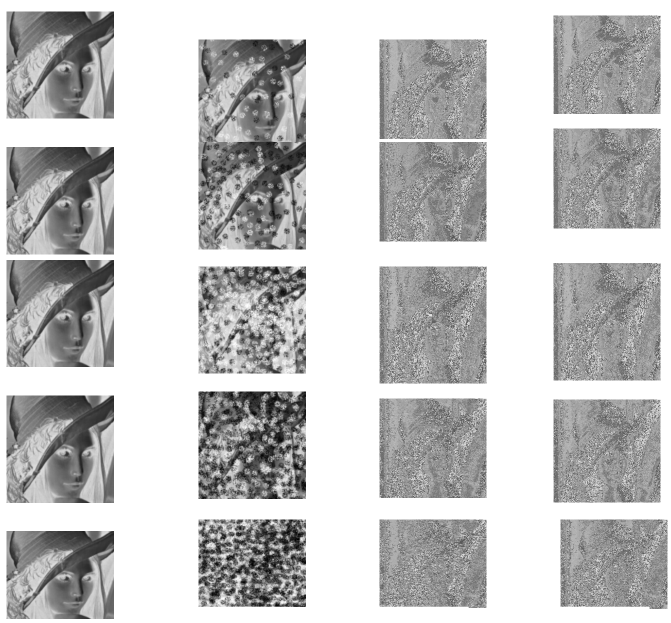


Fig.6. Various noise densities versus computation time



a) b) c) d) e) f) g)

Fig.7. Simulation results of different filters for Lena image. Column: (a) Noise corrupted image. (b) Output for SMF. (c) Output for TDF. (d) Output for AMF. (e) Output for CWMF (f) Output for Modified Decision based filter (MSSD). (g) Output for Proposed Algorithm. Row 1 shows the Lena image corrupted by 30% noise. Row 2 shows the Lena image corrupted by 50% noise. Row 3 shows the Lena image corrupted by 70% noise. Row 4 shows the Lena image corrupted by 90% noise. Row 5 shows the Lena image corrupted by 95% noise



(a) (b) (c) (d)

Fig.8. Simulation results for Lena image corrupted by random impulse noise. Column (a) original image. Column (b) images corrupted by random impulse noise. Column (c) images restored by the modified decision based filter algorithm. Column (d) images restored by the proposed algorithm. Row 1 image corrupted by random impulse noise with the ratio of salt and pepper being 3% and 6.7% respectively. Row 2 image corrupted by random impulse noise with the ratio of salt and pepper being 12% and 7% respectively. Row 3 image corrupted by random impulse noise with the ratio of salt and pepper being 11.6% and 28% respectively. Row4 image corrupted by random impulse noise with the ratio of salt and pepper being 43% and 9.7% respectively. Row 5 image corrupted by random impulse noise with the ratio of salt and pepper being 34% and 36.9% respectively

4. CONCLUSION

A novel algorithm has been proposed to eliminate blurring of images for large window sizes and poor impulse noise removal for small window sizes which are commonly encountered in SMF. The proposed algorithm makes use of 3X3 window for all noise densities using the neighborhood pixels to be processed in the current window considered for processing. This eliminates the complexity of existing adaptive median filter, progressive switched median filter and Chan-Nikolova method. The Srinivasan and Ebenezar method makes use of all nine inputs for the evaluation of median values this hampers the processing speed of the existing algorithms, as the evaluating procedure has to wait for the previous stage comparison output. This drawback is overcome by the proposed algorithm since the median value is computed only if the pixel to be processed is noisy. The proposed algorithm eliminates the need for nine inputs by replacing six, five, four, three inputs in successive stages for the computation of the median. This makes the proposed algorithm much faster when compared the existing decision based algorithms. In the case of existing decision based filters the process of decision making becomes quite complex when the evaluated median is found to be noisy. This complexity is eliminated in the proposed algorithm by finding out the uncorrupted pixel in the current window considered and replacing the current pixel to be processed with the mean of the uncorrupted pixels in the given window. The use of this linear operation does not hamper the non linearity of the proposed algorithm, which is the phenomenon for effective removal of impulse noise. All these advantages make the proposed filter to perform consistently for varying noise densities from 5% to 95% with fixed window of size 3X3. The novel sorting technique used in the proposed algorithm reduces the computational time, which is 1.5 times less than the existing decision based algorithm and other adaptive algorithm and reduced by the factor of 150 to 200 compared with the two-phase algorithm. The algorithm was implemented on images corrupted by both, fixed impulse noise (MATLAB inbuilt function) and random impulse noise (which are added manually). The proposed method removes the noise effectively even at noise level as high as 85% and preserves the edges without any loss up to 80% noise levels. The proposed algorithm is tested on different images and is found to produce better results in terms of the qualitative and quantitative measures of the image, as compared to SMF, AMF, TDF, CWMF and modified decision based filter, even at noise densities as high as 85%.

REFERENCES

[1] K.S.Srinivasan and D. Ebenezer, 2007, “A New Fast and Efficient Decision- Based Algorithm for Removal of High-Density Impulse Noises,” IEEE Signal Processing Letter, Vol.14, No.3, pp.189-192.

[2] How-Lung Eng, Kai-Kuang Ma, February 2001, “Noise Adaptive Soft-Switching Median Filter”, IEEE Transaction on Image Processing”, Vol.10, No 2.

[3] Abreu, E. Lightstone, M. Mitra, S.K. Arakawa, K.A, June 1996, “New Efficient Approach for the Removal of Impulse Noise from Highly Corrupted Images”, IEEE Transactions on Image Processing, Vol. 5, No. 6, pp.1012 – 1025.

[4] K. Vasanth, S. Karthik., “A New Class of Decomposition Algorithm for the Reduction of Low Density Impulse Noise”, IEEE International Conference on ARTCOM2009, Kerala, India, pp.203- 207.

[5] Fei Duan and Yu-Jin Zhang, July 2010, “A Highly Effective Impulse Noise Detection Algorithm for Switching Median Filters”, IEEE Signal Processing Letters, Vol.17, No.7, pp.647–650.

[6] K. Aiswarya, V. Jayaraj and D. Ebenezer, 2010, “A New and Efficient Algorithm for the Removal of High Density Salt and Pepper Noises in Images and Videos”, Second International Conference on Computer Modeling and Simulation, China, pp.409 – 413.

[7] Tao Chen and K. Ma, December 1999, “Tri-state Median Filter for Image De-noising”, IEEE Trans Image processing, Vol.8, No.12, pp.1834-1837.

[8] Raymond H. Chan, Chung-Waa Ho, and Mila Nikolova, October 2005, “Salt and Pepper Noise Removal by Median – Type Noise Detectors and Detail Preserving Regularization,” IEEE Transactions on Image Processing, Vol. 14, No. 10, pp.1479-1485.

[9] Z. Wang and D. Zhang, January 1999, “Progressive Switching Median Filter for Removal of Impulse Noise from Highly Corrupted Images”, IEEE Transactions on Circuits Systems II, Vol.46, No.1, pp.78-80.

[10] How-Lung Eng and Kai-Kuang Ma, February 2001, “Noise Adaptive Soft-Switching Median Filter”, IEEE Trans image Processing, Vol.10, No.2, pp.242-251.

[11] H. Hwang and R.A. Haddad, April 1995, “Adaptive median filters: new algorithms and results”, IEEE transaction on Image processing, Vol.4, No.4, pp 499-502.

[12] R.C. Gonzalez, and R.E. Woods, 2002, Digital Image Processing (2nd Edition), Prentice Hall.

[13] I. Pitas and A. N. Venetsanopoulos, 1990, Nonlinear Digital Filters: Principles and Applications, Boston, Kluwer.

[14] W.K. Pratt, 2007, Digital Image Processing, John Wiley & Sons

[15] J. Astola and P.Kuosmanen, 1997, Fundamentals of Nonlinear Digital Filtering, CRC Press.