A Novel Filter for Salt and Pepper Noise using Adaptive Interpolation Technique

R. Vishnu Priya
PG Scholar
St. Xavier's Catholic College of Engineering
Chunkankadai
India

C. Seldev Christopher, PhD.
Professor
St. Xavier's Catholic College of Engineering
Chunkankadai
India

ABSTRACT
An interpolation technique with edge sensing and adaptive window size is proposed for restoration of images that are highly corrupted by salt and pepper noise. The new technique shows significantly better image quality than Standard Median Filter, Adaptive Median Filter, Decision Based Algorithm, Progressive Switched Median Filter, Decision Based Unsymmetric Trimmed Median Filter and Modified Decision Based Unsymmetric Trimmed Median Filter. The Proposed technique is tested against different images and it gives better Peak Signal-to-Noise Ratio.

General Terms
Impulse Noise, Interpolation.

Keywords
Median Filter, salt and pepper noise, Adaptive Interpolation.

1. INTRODUCTION
Impulse noise in images is present due to channel transmission errors or introduced during the signal acquisition stage. There are two types of impulse noise, namely salt and pepper noise and random valued noise. Salt and pepper noise represents itself as randomly occurring white and black pixels. Number of nonlinear filters has been proposed for restoration of images contaminated by salt and pepper noise. Among these standard median filter is introduced to remove the salt and pepper noise. In median filter, the corrupted pixel is replaced by its median value. The major drawback of standard Median Filter is that the filter is not effective for high noise densities [1], [2]. That is, when the noise level is too high the edge details of the original image will get lost and also the standard median introduced too much distortion in the processed image. So Adaptive Median Filter [3], [4] is introduced. But at higher noise densities, both the corrupted and uncorrupted pixels in the neighborhood of an impulse corrupted pixel are considered to determine the noise replacing value, so the window size has to be increased which may lead to blurring the image.

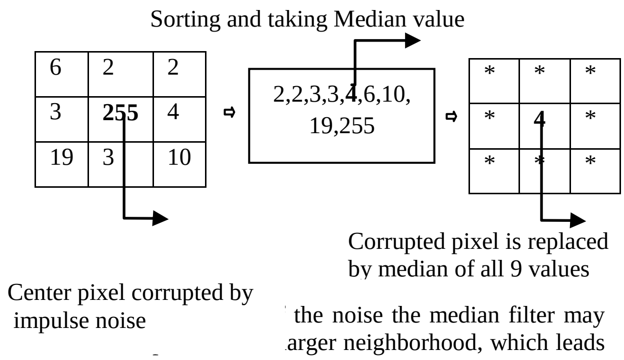
In progressive switching median filter [5] the difference of the median value in the filtering window and the current pixel value is compare with a pre-defined threshold to decide about the presence of impulse in an image. The major drawback of this method is taking a robust decision is difficult. To overcome the above drawback, Decision Based Algorithm is proposed [6], [7]. In this, at high noise density, the neighboring pixel is used repeatedly for replacing the noisy pixel which degrades the quality of restored images, called streaking effect. In order to avoid this drawback, Decision Based Unsymmetric Trimmed Median Filter is proposed [8]. If the processing pixel in the window takes either the value of 0 or 255, then the corresponding pixel values are trimmed from the window & the median of the remaining values are calculated. At high noise densities, the trimmed median value cannot be obtained. So this algorithm can be also not suitable for high noise density. The Modified Decision Based Unsymmetric Trimmed Median Filter algorithm is introduced to remove this drawback. At high noise density the mean value is taken into account if the selected window contains all 0's or 255's or both [9]. The proposed interpolation technique with edge sensing and adaptive window size remove this drawback at high noise density and gives better Peak Signal-to-Noise Ratio.

The rest of the paper is structured as follows. Section 2 describes about the existing system. Section 3 describes about the proposed algorithm. Section 4 & 5 describes about the noise detection and the determination of replacing the noisy pixel. Section 6 describes about the results and discussion, and Section 7 describes the conclusion.

2. EXISTING SYSTEMS

2.1 Standard Median Filter
The standard median filter [1], [2], [12] is a reliable method to remove salt and pepper noise without damaging the edge details.

This filtering requires arranging the pixel values either in increasing or decreasing order and replaces the noisy pixel by the median of its neighborhood values.



... the noise the median filter may ... larger neighborhood, which leads to a loss of edge details.

2.2 Adaptive Median Filter
In adaptive median filter [3], [4] the pixel of the corrupted image is retained if the median and the processing pixel itself are strictly between the minimum and the maximum value

of the filtering window. If the median lies between that minimum and the maximum value of that window and the processing pixel does not lies between that minimum and the maximum value, the pixel is replaced by the median value;

otherwise when the median does not lies strictly in between that minimum and the maximum value of that window, the size of the window is increased and the pixel is replaced by the median of the increased size window, provided, this new median is also strictly between the minimum and the maximum value of the window, otherwise, the window size is again increased by without changing the central pixel.

The main drawback of the adaptive median filter is that at high noise density the adaptive nature increases the window larger and larger to obtain a valid median value. Also, there is a possible that the median value obtained from larger window will not be accurate one.

2.3 Progressive Switching Median Filter

In Progressive Switching Median Filter [5], [10] to determine the presence of impulse in an image, the difference of the median value, M(i,j) of the pixels in the filtering window and the current processing pixel, P(i,j) is calculated and then it is compared with a threshold. If the difference is greater than the threshold, then the noisy pixels are replaced by the median of the noise free pixels.

$$X(i,j) = \begin{cases} M(i,j) & \text{if } \; |M(i,j) - P(i,j)| > \text{Threshold} \\ P(i,j) & \text{Otherwise} \end{cases}$$

The drawback of this process is that the estimation of the optimum threshold is very difficult. Also some time the impulse may be wrongly identified as original pixel and the original pixel may be identified as impulse.

2.4 Decision Based Algorithm

In decision based algorithm [6], [7] the pixel value inside the window are sorted in ascending order and the middle element of the window is consider as median value of that window. If the processing pixel takes the value of minimum and maximum gray levels, then it is consider as corrupted pixel and it is replaced by median of its neighborhood pixels. If the median of the neighborhood pixels also takes the value of 0 or 255, then the processing pixel is replaced by the value of its neighborhood pixel value.

Window selected

Row Sorting

$$\begin{bmatrix} 255 & 255 & 255 & 0 & 220 \\ 122 & 223 & 255 & 255 & 255 \\ 132 & \mathbf{255} & 121 & 124 & 255 \\ 120 & 250 & 255 & 255 & 255 \\ 250 & 255 & 200 & 125 & 120 \end{bmatrix} \qquad \begin{bmatrix} 122 & 223 & 255 \\ 121 & 132 & 255 \\ 120 & 250 & 255 \end{bmatrix}$$

(0<255≤255) processing pixel is noisy so replaced by median value.

Column Sorting

Right diagonal Sorting

$$\begin{bmatrix} 120 & 132 & 255 \\ 121 & 223 & 255 \\ 122 & 250 & 255 \end{bmatrix} \Rightarrow \begin{bmatrix} 120 & 132 & 255 \\ 121 & \mathbf{223} & 255 \\ 122 & 250 & 255 \end{bmatrix}$$

Median, which is used for replacing the processing pixel

Window selected

Row Sorting

$$\begin{bmatrix} 255 & 255 & 255 & 0 & 220 \\ 122 & 223 & \mathbf{255} & 255 & 255 \\ 132 & 255 & 121 & 124 & 255 \\ 120 & 250 & 255 & 255 & 255 \\ 250 & 255 & 200 & 125 & 120 \end{bmatrix} \qquad \begin{bmatrix} 0 & 255 & 255 \\ 223 & 255 & 255 \\ 121 & 124 & 255 \end{bmatrix}$$

(0<255≤255) both processing pixel and median are noisy. So processing pixel is replaced by left neighborhood from the last processed value

Column Sorting

Right diagonal Sorting

$$\begin{bmatrix} 0 & 124 & 255 \\ 121 & 255 & 255 \\ 223 & 255 & 255 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 124 & 255 \\ 121 & \mathbf{255} & 255 \\ 223 & 255 & 255 \end{bmatrix}$$

Median is noisy so processing pixel is replaced by using the neighborhood pixel

The drawback of this algorithm is at high noise density the neighboring pixel is used repeatedly for replacing the noisy pixel which degrades the quality of restored images, called streaking effect.

2.5 Decision Based Unsymmetric Trimmed Median Filter

In decision based unsymmetric trimmed median filter [8], the pixel values inside the window are sorted and if the processing pixel takes either the value of 0 or 255, then the corresponding pixel values in the window are trimmed from the window & the median of the remaining values are calculated, which is then used for replacing the noisy pixel.

Window Selected

median = (120 +158) / 2 = 139

$$\begin{bmatrix} 121 & 221 & 45 & 149 & 75 \\ 222 & 255 & 224 & 201 & 105 \\ 0 & 120 & \mathbf{255} & 74 & 195 \\ 167 & 158 & 0 & 109 & 163 \\ 224 & 0 & 255 & 123 & 0 \end{bmatrix} \Rightarrow \boxed{74, 109, \mathbf{120}, \mathbf{158}, 201, 224}$$

(0<255≤255) processing pixel is noisy.

$$\begin{bmatrix} 121 & 221 & 45 & 149 & 75 \\ 222 & 255 & 224 & 201 & 105 \\ 0 & 120 & \mathbf{139} & 74 & 195 \\ 167 & 158 & 0 & 109 & 163 \\ 224 & 0 & 255 & 123 & 0 \end{bmatrix}$$

Current noisy pixel in the window is replaced by median value calculated.

The main drawback of the decision based unsymmetric trimmed median filter is, at high noise density, the trimmed median value cannot be obtained because the selected window contains all 0's or 255's or both.

2.6 Modified Decision Based Unsymmetric Trimmed Median Filter

In modified decision based unsymmetric trimmed median filter [9], if the processing pixels takes either the value of 0 or 255, then two cases are possible. If the selected window contains all the element as 0's or 255's, then replace the processing pixel with the mean of the element of the window

.

$$\begin{bmatrix} 0 & 255 & 0 \\ 255 & \mathbf{255} & 0 \\ 0 & 0 & 255 \end{bmatrix}$$

(0<255≤255) processing pixel is noisy, also all the remaining elements are 0's & 255's, so calculate mean and replace the processing pixel with the mean value.

If the selected window contains not all the element as 0's and 255's, then eliminate 255's & 0's and find the median value of the remaining elements. Replace the processing pixel with the median value.

$$\begin{bmatrix} 120 & 255 & 110 \\ 0 & \mathbf{0} & 220 \\ 150 & 0 & 255 \end{bmatrix}$$

(0≤0<255) processing pixel is noisy but not all the elements are 0's and 255's, so calculate median and replace the processing pixel with the median value calculated.

3. PROPOSED ALGORITHM

The proposed interpolation technique with edge sensing and adaptive window size algorithm processes the corrupted images by first detecting the impulse noise. The processing pixel is checked whether it is noisy or noisy free. That is, if the processing pixel lies between maximum and minimum gray level values then it is noise free pixel, it is left unchanged. If the processing pixel takes the maximum and minimum gray level values then it is noisy pixel and the processing pixel is tested for noise detection. After noise detection, if the processing pixel is detected as noise, it is processed by using the proposed algorithm.

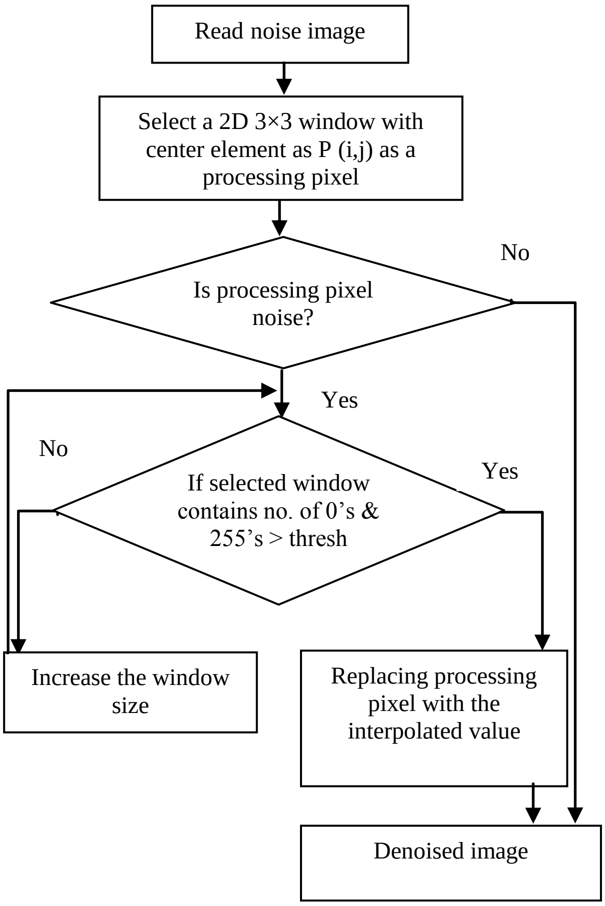


Fig 1: Architecture of proposed system

The step is as follows

1) A 2-D window of size 3×3 is selected. Assume the pixel to be processed is P(i,j).
2) The P(i,j) is an uncorrupted pixel if P(i,j) lies between maximum and minimum grey level values (i.e.) 0< P(i,j)<255; the pixel being processed is left unchanged. Otherwise, P(i,j) is a corrupted pixel.
3) If P(i,j) is a corrupted pixel, then the neighboring pixel values are taken into account and compared with the threshold value.
   a) If the value is less than threshold then increase the window size.
   b) Otherwise, the processing pixel is replaced with the interpolated value.

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

4. NOISE DETECTION

In order to distinguish the salt and pepper noise from an original image first the corrupted pixels should be detected from the uncorrupted pixels. For that a simple algorithm [11] is used as follows:

The algorithm consists of two phases namely phase I and phase II.

Phase I:

1) Consider a 5×5 window with center processing pixel as $P_{i,\,j}$.
2) Sort all the pixels inside the window in ascending order (S), so that the minimum values lies at the beginning of S and the maximum value lies at the end of S, then find the median (med) of S.
3) Calculate the intensity difference (D) between each adjacent pixel in S.
4) Now, from the first value to the med in S, find the maximum intensity difference in D of the same range & mark its corresponding pixel in S.
5) Also, from the med to the last value in S, find the maximum intensity difference in D of the same range & mark its corresponding pixel in S.
6) Now, three ranges namely minimum, medium and maximum ranges are obtained.
7) If the processing pixel belongs to the medium range, then it is consider as uncorrupted pixel & hence no further processing is required.
8) Otherwise, if it lies in minimum or maximum range, the phase II will be starts.

Phase II:

1) Consider a 3×3 window, with processing pixel Pi,j as center & proceed the step 2 to step 6.
2) If the processing pixel belongs to the medium range, then it is consider as uncorrupted pixel or else it is consider as corrupted.

Consider an example of 5×5 window with center processing pixel as 255.

$$W_{5\times5} = \begin{bmatrix} 255 & 150 & 255 & 255 & 255 \\ 155 & 0 & 0 & 220 & 255 \\ 0 & 22 & \mathbf{255} & 0 & 126 \\ 12 & 0 & 45 & 134 & 255 \\ 255 & 160 & 255 & 0 & 0 \end{bmatrix}$$

- Sort the window in ascending order & find the median (med) as 150. i.e., S = [0 0 0 0 0 0 0 12 22 45 126 134 **150** 155 160 220 255 255 255 255 255 255 255 255 255].
- The intensity difference between each adjacent pixel in S is calculated as D = [0 0 0 0 0 0 12 10 23 **81** 8 16 5 5 50 **60** 35 0 0 0 0 0 0 0].
- For the pixel with intensities between 0 & med in S, the corresponding maximum difference in D is 81, which is obtained from taking the difference of 45 & 126 in S.
- For the pixel with intensities between med & 255 in S, the corresponding maximum difference in D is 60, which is obtained from taking the difference of 160 & 220 in S.
- Thus the minimum, medium & the maximum ranges are {0, 0, 0, 0, 0, 0, 0, 12, 22, 45}, {126, 134, 150, 155, 160} & {220, 250, 255, 255, 255, 255, 255, 255, 255, 255} respectively.

Since the processing pixel **255** lies in maximum range, the phase II starts. For that a 3×3 window is selected with 255 as center processing pixel.

$$W_{3\times3} = \begin{bmatrix} 0 & 0 & 220 \\ 22 & \mathbf{255} & 0 \\ 0 & 45 & 134 \end{bmatrix}$$

- Sort the window in ascending order & find the median (med) as 22. i.e., S = [0 0 0 0 **22** 45 134 220 255].
- The intensity difference between each adjacent pixel in S is calculated as D = [0 0 0 **22** 23 **89** 86 35].
- For the pixel with intensities between 0 & med in S, the corresponding maximum difference in D is 22, which is obtained from taking the difference of 0 & 22 in S.
- For the pixel with intensities between med & 255 in S, the corresponding maximum difference in D is 89, which is obtained from taking the difference of 45 & 134 in S.
- Thus the minimum, medium & the maximum ranges are {0, 0, 0, 0}, {22, 45} & {134, 220, 255} respectively.

At the end of phase II, the center processing pixel 255 is detected as "corrupted" pixel, since it belongs to the maximum range.

5. DETERMINATION OF REPLACING THE NOISY PIXEL

In the proposed method, the corrupted pixel is processed by using interpolated value. For that first the horizontal and vertical gradient is calculated as shown below

P_1	P_{up}	P_2
P_{left}	$P_{i,j}$	P_{right}
P_3	P_{down}	P_4

Horizontal gradient $L^H = (P_{left} - P_{right})$

Vertical gradient $L^V = (P_{up} - P_{down})$

If L^H and L^V < ethresh, then there exists smooth region

Calculate $P(i,j) = (P_{left} + P_{up} + P_{right} + P_{down}) / 4$ (4.1)

If $L^H > L^V$, then there exists a sharp horizontal edge

Calculate $P(i,j) = W1\,((P_{left} + P_{right})/2) + W2\,((P_{up} + P_{down})/2)$, $W1 > W2$ & $W1 + W2 = 1$ (4.2)

If $L^H < L^V$, then there exists a sharp vertical edge

Calculate $P(i,j) = W1\,((P_{up} + P_{down})/2) + W2\,((P_{left} + P_{right})/2)$, $W2 > W1$ & $W1 + W2 = 1$ (4.3)

If any two consecutive pixels P_{up}, P_{down}, P_{left}, P_{right} takes minimum or maximum grey level values, then replace P_{left}, P_{up}, P_{down}, P_{right} by P_1, P_2, P_3, P_4 respectively.

Where

W1 & W2 − weights and based on various analysis the best values chosen for weights are 0.13 and 0.87 respectively.

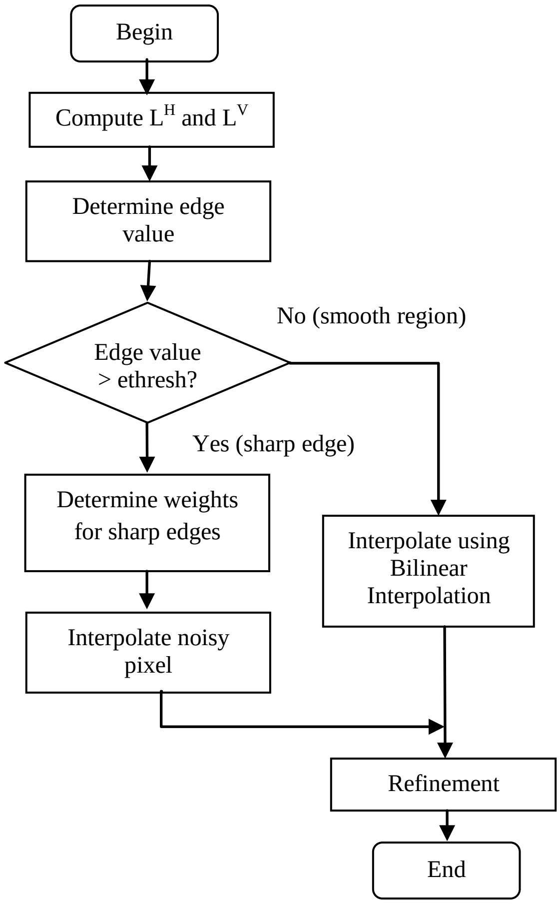


Fig 2: Block diagram for replacing the Noisy pixel

The step is as follows

1) First, the horizontal and vertical gradients are calculated to determine the existence of sharp horizontal and vertical edges.
2) If the horizontal and vertical gradient values are lesser than the threshold, then there is no edge exists and the region is consider to be smooth. The processing pixel P(i,j) is processed by bilinear interpolation using the equation (4.1).

3) If the horizontal and vertical gradient values are greater than the threshold, then there exists an edge.

Two cases are possible:

a) If the horizontal gradient L^H is greater than the vertical gradient L^V, then there is a sharp horizontal edge and the processing pixel P(i,j) is processed by using the equation (4.2).

b) If the vertical gradient L^V is greater than the horizontal gradient L^H, then there is a sharp vertical edge and the processing pixel P(i,j) is processed by using the equation (4.3).

4) At higher noise density, the interpolation doesn't perform well; as a result some noises are seen in the restored image. In that case after getting the restored image, smoothening technique are applied to that particular noisy positions.

The pictorial representation of replacing the noisy pixel is shown in Fig. 2.

6. RESULTS AND DISCUSSION

The performance of the proposed algorithm is tested with different grayscale images. The noise density is varied from 10% to 90%. Denoising performances are quantitatively measured by the PSNR as defined in (1)

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

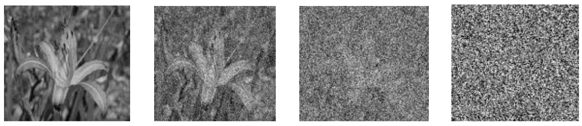
Here 255 is the maximum pixel (signal) value of the image.

$$\mathrm{MSE} = \frac{1}{M \times N}\sum_{i-1}^{M}\sum_{j-1}^{N}|u(i,j) - I(i,j)|^2$$

where PSNR stands for peak signal to noise ratio, MSE stands for mean square error, M×N is the size of the image, u(i,j) represents the original image, I(i,j) represents the denoised image. Fig.3 displays the original and corrupted images of Flower.jpg image. Fig.5 displays the original and corrupted images of Cameraman.jpg image.

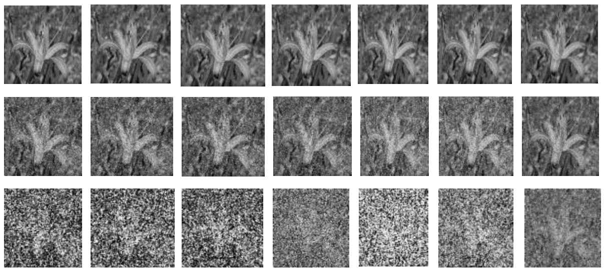
In Fig.4, Fig.6, the first column represents the output of standard median filter (SMF) [2], second column represents the output of Adaptive median filter (AMF) [3], third column represents the output of Progressive switching median filter (PSMF) [4], fourth column represents the output of Decision based algorithm (DBA) [5], fifth column represents the output of decision based unsymmetric trimmed median filter (DBUTMF) [6], sixth column represents the output of Modified decision based unsymmetric trimmed median filter (MDBUTMF) [7] and seventh column represents the output of the Proposed Algorithm (PA).

The MSE, PSNR values of the proposed algorithm are compared against the existing algorithms by varying the noise density as shown in Table 1, Table 2, Fig 7 & Fig 8. From the Fig.7 & Fig.8 it is observed that the proposed algorithm provides better PSNR when compared with the existing algorithms at both low and high noise densities.



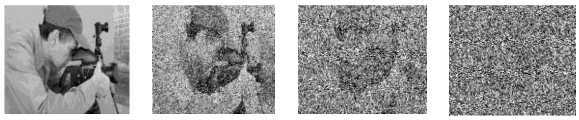
a b c d

Fig 3: (a) Original Flower image. (b) Image corrupted by 30% noise density. (c) Image corrupted by 70% noise density. (d) Image corrupted by 90% noise density.



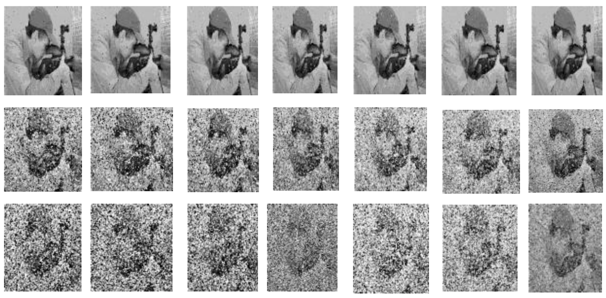
a b c d e f g

Fig 4: Results of different filters for Flower image. (a) Output of SMF. (b) Output of AMF. (c) Output of PSMF. (d) Output of DBA. (e) Output of DBUTMF. (f) Output of MDBUTMF. (g) Output of PA. Row 1, Row 2 & Row 3 shows processed results of various filters for Flower.jpg image corrupted by 30%, 70% & 90% noise densities.



a b c d

Fig 5: (a) Original Cameraman image. (b) Image corrupted by 30% noise density. (c) Image corrupted by 70% noise density. (d) Image corrupted by 90% noise density.



a b c d e f g

Fig 6: Results of different filters for Cameraman image. (a) Output of SMF. (b) Output of AMF. (c) Output of PSMF. (d) Output of DBA. (e) Output of DBUTMF. (f) Output of MDBUTMF. (g) Output of PA. Row 1, Row 2 & Row 3 shows processed results of various filters for Cameraman.jpg image corrupted by 30%, 70% & 90% noise densities.

Table 1. MSE of the restored flower image corrupted by various noise densities

Noise Ratio %	Mean Square Error (MSE)						
	SMF	AMF	PSMF	DBA	DBU TMF	MDB UTMF	PA
10	5.87	4.36	4.40	4.34	4.29	4.24	4.13
20	39.22	37.32	34.95	30.32	29.74	28.36	17.81
30	158.1	151.6	156.9	140.3	140.2	144.56	57.13
40	587.3	574.5	519.6	484.1	474.4	460.51	138.6
50	1462	1452	1394	1351	1299	1067.8	303.2
60	2908	2828	2810	2151	2648	2597.4	506.2
70	5166	5078	5020.1	4207	5274	4834.7	919.88
80	8800.3	84901	8474.1	8170.9	7294.1	6956	1575
90	12858	12838	12781	11939	11992	10758	2395

Table 2. MSE of the restored cameraman image corrupted by various noise densities

Noise Ratio %	Mean Square Error (MSE)						
	SMF	AMF	PSMF	DBA	DBU TMF	MDB UTMF	PA
10	14.53	13.86	13.55	13.67	12.89	12.80	6.97
20	47.09	43.42	43.63	38.87	39.73	39.68	25.42
30	192.1	185.3	174.7	174.7	167.5	154.47	65.01
40	574.0	569.6	547.5	504.8	504.7	491.46	170.4
50	1478	1461	1383	1187	1167	1141	325.9
60	2936	2894	2883	2465	2389	2383.9	612.3
70	5791.7	5431.7	5337.4	4398.9	4336.7	3875.4	1058
80	8952.3	8860.3	8719.5	76088	6796.7	5908.5	1747
90	13320	13281	13256	12288	10021	99571	2677

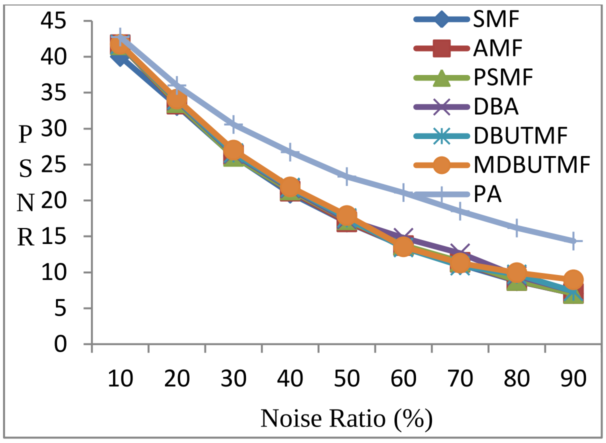


Fig 7: PSNR analysis of the various filters on Flower image for various noise levels

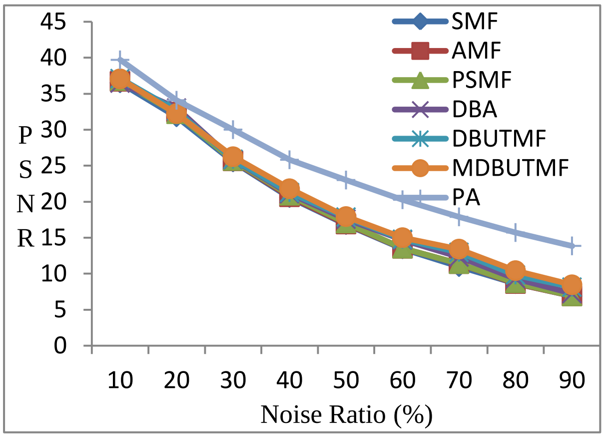


Fig 8: PSNR analysis of the various filters on Cameraman image for various noise levels

7. CONCLUSION

A new algorithm is proposed to restore the image corrupted by salt and pepper noise, which employs the sharp edge and smooth region detection and the adaptive interpolation technique. From the results and discussion, it is seen that the proposed algorithm preserve details at high noise densities and has a better performance than the existing algorithms.

8. ACKNOWLEDGMENTS

We would like to thank Dr. S. Joseph Sekar, Principal, St. Xavier's Catholic College of Engineering and Dr. J. Mohanalin, Professor, St. Xavier's Catholic College of Engineering for their constant encouragement and support.

9. REFERENCES

[1] W. K. Pratt, Digital Image Processing, John Wiley &Sons, Inc, New York, NY, 1978.

[2] RC. Gonzalez and R. E. Woods, "Digital Image Processing", Prentice Hall, 2002.

[3] H. Hwang and R. A. Hadded, "Adaptive median filter: New algorithms and results," IEEE Transactions on Image Processing, vol. 4, no. 4, Apr. 1995, pp. 499–502.

[4] S.Manikandan, O.Uma Maheswari, D.Ebenezer, "An Adaptive Recursive Weighted Median Filter with Improved Performance in Impulsive Noisy Environment", WSEAS Transactions on Electronics, Issue 3, Vol.1, July 2004.

[5] Zhou Wang and David Zhang, "Progressive switching median filter for the Removal of impulse noise from highly corrupted images," IEEE Trans Circuits and Systems-II: Analog and Digital Signal Processing, vol. 46, no.1, Jan.1999.

[6] Srinivasan K. S. and Ebenezer D, "A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noises", IEEE signal process. Lett, Vol. 14, No. 3, pp.189 -192, 2007.

[7] Pok, G. Jyh-Charn Liu, "Decision based median filter improved by predictions", ICIP 99(1999) proceedings vol. 2 pp 410-413.

[8] 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 Int. Conf. Computer Modeling and Simulation, 2010, pp. 409–413.

[9] S.Esakkirajan, T. kumar, Adabala N. Subramanyam, and C. H. PremChand, " Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter," IEEE Signal Process. Lett, vol. 18, no. 5, May. 2011.

[10] H.-L. Eng and K.-K. Ma, "Noise adaptive soft-switching median filter," IEEE Transactions on Image Processing, vol. 10, no. 2, Feb. 2001, pp. 242–251.

[11] Pei-Eng Ng and Kai-Kuang Ma, "A Switching median filter with boundary noise detection for extremely corrupted images," IEEE Transactions on Image Processing, vol. 15, no.6, June.2006.

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