An Efficient Adaptive Weighted Switching Median Filter for Removing High Density Impulse Noise

Madhu S. Nair · P. M. Ameera Mol

Received: 28 November 2012 / Accepted: 9 January 2014 / Published online: 17 June 2014
© The Institution of Engineers (India) 2014

Abstract Restoration of images corrupted by impulse noise is a very active research area in image processing. In this paper, an Efficient Adaptive Weighted Switching Median filter for restoration of images that are corrupted by high density impulse noise is proposed. The filtering is performed as a two phase process—a detection phase followed by a filtering phase. In the proposed method, noise detection is done by HEIND algorithm proposed by Duan et al. The filtering algorithm is then applied to the pixels which are detected as noisy by the detection algorithm. All uncorrupted pixels in the image are left unchanged. The filtering window size is chosen adaptively depending on the local noise distribution around each corrupted pixels. Noisy pixels are replaced by a weighted median value of uncorrupted pixels in the filtering window. The weight value assigned to each uncorrupted pixels depends on its closeness to the central pixel.

Keywords Image denoising · Efficient adaptive weighted switching median filter · Impulse noise · Salt and pepper noise

Introduction

Digital Images are often contaminated by noise during acquisition, storage and transmission yielding degradation at the quality of images; therefore a common problem in

many applied science and engineering is the restoration of the corrupted images. Impulse noise is a specific type of noise which causes alteration of the pixels in the images so that their gray values do not exhibit compatibility with their local neighborhood. Usually impulse noises are degraded by impulse noise of short duration and high energy because of the errors caused by noisy sensors or transmission channels [1].

There are many methods used for removal of impulse noises from the images. Usage of linear filters such as averaging filters produces blurring of the images. Non linear filters such as median filters are most popular technique for removing impulse noise because of its good denoising power and computational efficiency [2, 3]. Median filters (MED) replace the value of a pixel by a median of the intensity levels in the neighborhood of that pixel. Median filters use a fixed filtering window size for finding out neighborhood pixels. However most of the median filters are implemented uniformly across the image and thus tend to modify both noisy and noise free pixels. Consequently the effective removal of impulse noise is often accomplished at the expense of blurred and distorted features thus removing fine details in the image. There are different variations of Median filters such as weighted median filters (WM) [4], centre weighted median filters (CWM) [5] and adaptive median filters (AMED) [6].

An adaptive median filter discriminates pixels in the filtering window as corrupted and uncorrupted and then a filtering technique is applied to corrupted pixels in the window. In this method noisy pixels are replaced by the median value of the pixels in the filtering window. AMED [6] performs well at low noise densities since the corrupted pixels that are replaced by the median values are very few. At higher noise densities, window size has to be increased to get better noise removal which will lead to less

M. S. Nair (✉) · P. M. Ameera Mol
Department of Computer Science, University of Kerala,
Kariavattom, Thiruvananthapuram 695581, Kerala, India
e-mail: madhu_s_nair2001@yahoo.com

correlation between corrupted pixel values and replaced median pixel values.

In efficient decision-based algorithm (EDBA) [7], the corrupted pixels are replaced by either the median pixel or neighbourhood pixel by using a fixed window size of 3×3 , resulting in lower processing time and good edge preservation. EDBA filter [7] performs well only for salt and pepper noises up to 70 %. In this method, a smooth transition between the pixels is lost leading to degradation in the visual quality of the image in the form of line artifacts, since it only considers the left neighborhood from the last processed value. For other impulse noise models such as Noise Model 3 and Noise Model 4 this method is found to give poor performance.

Switching median filters [8] are shown to be simple and yet more effective than uniformly applied methods such as median filters [9]. In the case of switching median filters, it will identify the possible noisy pixels in the image and then replace them by using median filters and its variants while leaving all other uncorrupted pixels unchanged. Impulse detection became therefore crucial to subsequent filtering in switching median filters.

The earlier developed switching median filters were commonly found, being non adaptive to a given, but unknown, noise density and prone to yielding pixel misclassifications especially at higher noise density interference. There are different methods for impulse noise detection fuzzy approaches as in [10–12], neural approaches [13] and boundary based approaches [14–18]. Among the three categories, boundary based approach [14] is preferred due to its simplicity compared to computational complexity and system structure of other two categories. To address pixel misclassification issue in switching median filters at high density noise, a noise adaptive soft switching median filter (NASM) was proposed [15], which consists of a three level hierarchical soft switching process. The boundary based approach called boundary discriminative noise detection [14] is very good in detecting impulse noises of various densities. BDND [14] can handle image corruption even up to 90 % of noise density. Highly Effective Impulse Noise Reduction algorithm [18] is providing efficient method for noise detection. This method uses both boundary based information and directional based information for detection purpose. This method is more efficient in noise detection than BDND method in terms of false alarms and misdetection produced.

In noisy pixel replacement stage many median based schemes have been proposed. The NASM [15] noise replacement strategy is giving robust performance in removing impulse noise while preserving signal details across a wide range of noise densities, ranging from 10 %

to 50 %. However, for those corrupted images with noise density greater than 50 %, the quality of the recovered images become significantly degraded, due to the sharply increased number of misclassified pixels. The modified NASM [14] along with BDND detector is giving better performance for high density impulse noises compared to filtering in NASM [14], but causes blurring of edges and loss of finer details in the image. NAFSMF [16] uses fuzzy membership value of luminance difference value and median value in the filtering window for noisy pixel replacement. The noisy pixels are replaced either partially or fully depending on fuzzy membership value. Although NAWSM [17] filter is found to be effective in removing high density impulse noises from images, there occurs some misdetection and false alarms during the detection stage. To improve the visual quality of the restored image an Efficient Adaptive Weighted Switching Median (EAWSM) filter is proposed in this paper.

The proposed method uses HEIND [18] detection method for determining the corrupted pixels in the noisy image and then an efficient noise adaptive weighted switching median filter [17] is applied on each corrupted pixels. The filtering window size is chosen adaptively and depends on the percentage of noise that corrupts the image. The initial filtering window size is taken as 3×3 and maximum window size is chosen depending on the percentage of noise that corrupts a local region around the detected noisy pixel in the image. The noisy pixels are replaced by the weighted median value of uncorrupted pixels in the filtering window. The proposed method is giving significant improvement in visual quality of the image especially in the case of high density impulse noises and the thinner edges and finer details in the images are restored more effectively than previous methods.

Noise Models

In this section four different random impulse noise models are discussed. These models [14] are considered for extensively examining the performance of the proposed filter considering practical situations.

Noise Model 1

Noise is modeled as salt-and-pepper impulse noise, pixels are randomly corrupted by two fixed extreme values, 0 and 255 (for gray level image), generated with the same probability, that is, if P is noise density, then P_1 , the noise density of salt is $P/2$ and P_2 . The noise density of pepper is also $P/2$.

Noise Model 2

This is similar to Noise Model 1, but here each pixel might be corrupted by either pepper noise or salt noise with unequal probabilities, that is $P1 \neq P2$.

Noise Model 3

Instead of two fixed values, impulse noise could be more realistically modeled by two fixed ranges that appear at both ends with a length of m each, respectively, that is, $[0, m]$ denotes salt and $[255-m, 255]$ denotes pepper. Here for noise density P , $P1 = P2 = P/2$.

Noise Model 4

This is similar to Noise Model 3 but here probability densities of low intensity impulse noise and high density impulse noise are different, that is, $P1 \neq P2$.

Efficient Adaptive Weighted Switching Median (EAWSM) Filter

The proposed method EAWSM filter is a double stage filter, where initially it will perform a detection stage for identifying the noisy pixel positions. When a noisy pixel is detected it is subjected to the second stage known as filtering stage. For detection purpose, the Highly Effective Impulse Noise Detection Algorithm proposed by Fei Duan et al. [18] is used.

Noise Detection Stage

The basic strategy of HEIND [18] is to examine each pixel in its neighborhood from coarse to fine. During examination of pixels in the coarse stage a neighborhood of 21×21 is considered around each pixel. This stage is just like the first stage in BDND [14] detection method but with an improved classification criterion. The examination of pixels in the fine stage takes a neighborhood of 7×7 . This stage uses four directional convolutionary kernels to decide whether a pixel is noisy or not. Only the pixels which are classified as ‘corrupted’ in the first stage will be piped into second stage for examination. If current pixel is categorized into ‘corrupted’ in both stages, then it is considered to be contaminated.

Noise Detection—1st Stage

The most critical part of this stage is the determination of the decision boundaries. Here by considering a window size of 21×21 , more than 400 pixels are taken into account to calculate the local histogram and subsequently, the decision boundaries $b1$ and $b2$. Obviously, this order of magnitude is statistically meaningful and hence, the reliability of the two decision boundaries is acceptable. Depending on the decision boundaries, pixels are classified into three clusters—low intensity impulse noise, uncorrupted pixels and high density impulse noise. If $x(i, j)$ is the intensity of noisy image pixel under consideration and if $x(i, j) \leq \min(b1, m)$ $\{m$ denotes the length of interval of uniform distribution of impulse noises $\}$, it is classified as low intensity impulse noise, if $x(i, j) \geq \max(b2, m)$, it is classified as high intensity impulse noise and otherwise it is classified as uncorrupted pixel.

The following steps are used in the detection stage [18].

- (i) For each pixel $x(i, j)$ in the image, impose a 21×21 window centres on $x(i, j)$.
- (ii) Sort the pixels lie in the window region to an ordered vector v_o and find the median ‘med’
- (iii) Compute the difference vector of v_o

$$\begin{array}{cc}
 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 6 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \\
 \text{(a) } Ker^1 & \text{(b) } Ker^2 \\
 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \\
 \text{(c) } Ker^3 & \text{(d) } Ker^4
 \end{array}$$

Fig. 1 Four directional convolutionary kernels

Table 1 Window size for different noise densities

Noise Density	$W_D \times W_D$
$0\% < p \leq 20\%$	3×3
$20\% < p \leq 40\%$	5×5
$p > 40\%$	7×7

(iv) Find the pixels which correspond to the maximum differences in the intervals of $[0, \text{med}]$ and $(\text{med}, 255]$. And set these two pixel intensities as the decision boundaries $b1$ and $b2$, respectively

(v) Classify the pixels in current window into three clusters according to the rule given below

$$\text{class}(x(i, j)) = \begin{cases} \text{Pepper noise} & x(i, j) \leq \min(b1, m) \\ \text{Salt noise} & x(i, j) \geq \max(b2, 255 - m) \\ \text{Uncorrupted} & \text{Otherwise} \end{cases} \quad (1)$$

After such an application to the entire image, a two-dimensional binary decision map is formed at the end of the noise detection stage, with 1's indicating the positions of 'uncorrupted' pixels and 0's for those of 'corrupted' ones.

$$N1(i, j) = \begin{cases} 0 & x(i, j) \text{ noisy pixel} \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

Fig. 2 Column **a** represents original boat, Lena and circuit images. Columns **b**, **c** and **d** represent boat, Lena, and circuit images corrupted with 30, 60 and 90 % noise, respectively, based on Noise Model 1

Noise Detection—2nd Stage

The intensity differences between the current pixel and its neighbours in four representative directions are accurately quantified in this stage. For that, around each pixel, a 7×7 neighbourhood in four directions $0, \pi/4, \pi/2$ and $3\pi/4$ are considered. This window size helps to accurately quantify the pixel differences even in the case of high noise density and hence increase the efficiency and accuracy of second detection stage. The four different directional kernels used for this purpose is shown in Fig. 1. By convolving the group of kernels with the noise image, every pixel gets four convolutionary results. The absolute values of the convolutionary values are taken as the metrics of the intensity homogeneity of the neighbourhood of the current pixel. In order to provide reliable decisions on noise, this stage takes into account both minimum descriptor ($R_{\min}$) and maximum descriptor ($R_{\max}$). This stage also uses two predefined threshold values T_1 and T_2 during classification.

The following steps are used in the second stage of detection [17].

- Convolve the noisy image X , with four directional kernels given in Fig. 1.
- For each kernel k , find $V(i, j)^k = |X \otimes \text{Ker}^k|_{i,j}$
- Find $R(i, j)^{\min} = \min\{V(i, j)^1, V(i, j)^2, V(i, j)^3, V(i, j)^4\}$ and $R(i, j)^{\max} = \max\{V(i, j)^1, V(i, j)^2, V(i, j)^3, V(i, j)^4\}$

Fig. 3 Column **a** represents edge maps of original boat, Lena and circuit images. Columns **b**, **c** and **d** represent boat, Lena, and circuit images corrupted with 30, 60 and 90 % noise, respectively, based on Noise Model 1

Table 2 Performance parameters of Lena image corrupted by 30 % and 60 % noise based on Noise Model 1

	30 % Noise				60 % Noise			
	PSNR	IEF	SSIM	IQI	PSNR	IEF	SSIM	IQI
MED	25.57743	31.05598	0.960761	0.55217	22.00929	27.21838	0.913792	0.464371
AMED	33.84174	206.9121	0.99407	0.86349	27.43641	94.59734	0.974433	0.704765
MNASM	36.01521	344.4725	0.99638	0.915895	29.46695	152.1261	0.983695	0.750613
EDBA	34.71787	254.2646	0.995143	0.907821	27.7892	103.3891	0.976275	0.72064
NAFSMF	33.60344	618.5353	0.993717	0.899599	29.97606	268.3697	0.98551	0.76478
HMNASM	36.35323	370.1413	0.996647	0.921494	29.67978	158.8364	0.984466	0.760369
EAWSM	36.50742	383.5193	0.996762	0.922268	31.30859	231.1168	0.989373	0.795774

Table 3 Performance parameters of circuit image corrupted by 30 % and 60 % noise based on Noise Model 1

	30 % Noise				60 % Noise			
	PSNR	IEF	SSIM	IQI	PSNR	IEF	SSIM	IQI
MED	25.76302	37.19159	0.959928	0.545579	21.92215	30.42123	0.905493	0.450517
AMED	34.59798	282.6367	0.994764	0.927464	27.1684	101.3775	0.971484	0.779791
MNASM	37.18978	498.6319	0.997087	0.954637	28.83927	148.3144	0.980184	0.733033
EDBA	34.89471	297.174	0.995104	0.940453	27.87887	119.8241	0.975655	0.772546
NAFSMF	33.30487	261.0643	0.992951	0.88901	29.41089	106.4496	0.982678	0.749605
HMNASM	37.42248	539.0515	0.997236	0.956183	28.86856	147.807	0.980317	0.739655
EAWSM	37.68981	573.2746	0.997398	0.95857	31.51976	272.1543	0.989354	0.846654

Table 4 Performance parameters of Lena image and circuit image corrupted by 90 % noise based on Noise Model 1

Filters	Lena				Circuit			
	PSNR	IEF	SSIM	IQI	PSNR	IEF	SSIM	IQI
MED	8.563264	1.840969	0.202061	0.022447	8.094548	1.879931	0.179843	0.02868
AMED	10.45888	2.846429	0.349906	0.062397	9.903912	2.841197	0.305661	0.087942
MNASM	25.03471	81.82447	0.954504	0.477606	24.12532	75.64358	0.941065	0.387931
EDBA	19.51096	22.9719	0.84106	0.290314	19.04521	23.4329	0.815401	0.291309
NAFSMF	25.59571	97.90582	0.960257	0.510374	25.08875	39.32797	0.953116	0.484606
HMNASM	25.21851	85.37265	0.95632	0.484137	24.48525	81.98875	0.94577	0.402526
EAWSM	25.91257	100.1668	0.963175	0.534997	25.89555	113.4447	0.96122	0.579701

Table 5 PSNR values obtained for boat image based on Noise Model 1

Noise (%)	MED	AMED	MNASM	EDBA	NAFSMF	HMNASM	EAWSM
90	8.381632	10.3233	22.4115	18.37734	22.96581	22.69966	23.27274
80	12.71335	15.46635	23.6205	21.12903	24.45306	23.91824	25.22579
70	17.37563	20.96976	24.6718	23.53234	25.51336	24.85212	26.57445
60	20.43905	25.03024	25.9988	25.36471	26.5106	26.12361	27.88361
50	21.72098	27.13025	27.5804	27.26044	27.5182	27.82047	29.12123
40	22.60334	28.52717	29.6215	29.03983	28.65529	30.3083	30.76089
30	23.21251	30.40575	31.903	31.36166	29.93714	32.58421	32.69288
20	24.06188	32.21128	34.4342	33.76107	31.89992	35.27939	35.31241
10	24.9179	33.9189	37.4059	37.47399	34.79604	38.59112	38.59492

Table 6 IEF values obtained for boat image based on Noise Model 1

Noise (%)	MED	AMED	MNASM	EDBA	NAFSMF	HMNASM	EAWSM
90	1.758522	2.744471	44.3593	17.56312	51.56183	47.56183	54.27086
80	4.237549	7.994116	52.3802	29.46381	65.82606	55.82606	75.43841
70	10.84806	24.84822	58.2167	44.86728	85.63398	60.63398	90.1464
60	18.86463	54.00781	67.6308	58.43937	99.56359	69.56359	104.3235
50	21.08041	73.23791	80.9163	75.69246	105.81139	85.81139	115.7765
40	20.73124	80.75526	104.0705	91.2369	128.9278	121.9278	135.32
30	17.85578	93.28791	131.5206	116.7452	155.9128	154.9128	158.8381
20	14.41456	94.59943	156.6994	134.2007	190.1912	191.1912	192.6505
10	8.776051	69.16714	157.2348	158.9357	205.2426	206.2426	206.4229

Table 7 SSIM values obtained for boat image based on Noise Model 1

Noise (%)	MED	AMED	MNASM	EDBA	NAFSMF	HMNASM	EAWSM
90	0.167402	0.316213	0.9101	0.781848	0.922125	0.916557	0.928187
80	0.482357	0.666083	0.9322	0.884206	0.944692	0.937036	0.954112
70	0.758399	0.885844	0.9471	0.933216	0.956838	0.949462	0.966417
60	0.868675	0.953225	0.9614	0.956157	0.965788	0.962566	0.975239
50	0.899421	0.970935	0.9733	0.971658	0.972968	0.974849	0.981419
40	0.916667	0.978899	0.9834	0.981212	0.979243	0.98587	0.987278
30	0.927117	0.986272	0.9902	0.98898	0.984602	0.991659	0.991864
20	0.939373	0.990914	0.9945	0.993642	0.990226	0.995496	0.99553
10	0.949805	0.993843	0.9972	0.997266	0.994975	0.997871	0.997873

- (iv). If $R(i, j)^{\min} > T_1$ and $R(i, j)^{\max} - R(i, j)^{\min} > T_2$ classify the pixel $X(i, j)$ as noisy, otherwise classify it as uncorrupted (where $T_1 = 5, T_2 = 1$ which are empirically estimated from several test images)
- (v). Form a noise mask $N2$ as shown below

$$N2(i, j) = \begin{cases} 0, & x(i, j) \text{ noisy pixel} \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

The binary decision mask N is created from $N1$ and $N2$ as given below

$$N(i, j) = \begin{cases} 0, & N1(i, j) = 0 \wedge N2(i, j) = 0 \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

This noise mask generated is used in the filtering stage by the proposed EAWSM filter.

Table 8 IQI values obtained for the boat image based on Noise Model 1

Noise (%)	MED	AMED	MNASM	EDBA	NAFSMF	HMNASM	EAWSM
90	0.023058	0.064219	0.3584	0.4391	0.404037	0.368446	0.448815
80	0.104396	0.258232	0.48	0.5698	0.527174	0.49464	0.582476
70	0.251595	0.505563	0.5854	0.6661	0.621156	0.595271	0.673466
60	0.347314	0.659609	0.6835	0.7413	0.697754	0.69032	0.747222
50	0.389543	0.736445	0.7728	0.8021	0.760415	0.779574	0.806769
40	0.406506	0.78878	0.8439	0.8523	0.818791	0.852069	0.858244
30	0.416929	0.835577	0.8955	0.897	0.867313	0.899461	0.900563
20	0.426314	0.865149	0.9362	0.9364	0.914813	0.939583	0.939727
10	0.433707	0.875455	0.9683	0.9683	0.957943	0.971071	0.971074

Fig. 4 Column **a** represents restored images of boat by corrupted image (30 %). Column **b** represents restored images of Lena corrupted image (30 %). Column **c** represents restored images of circuit corrupted image (30 %). Rows 1, 2, 3, 4, 5 and 6 represent the restored images after applying the median filter (MED-7 × 7), Adaptive Median Filter (AMED), MNASM, NAFSMF, MNASM (with HEIND) and EAWSM filters, respectively. Column **d** represents the image quality map of the restored images of boat shown in the column **a**. Column **e** represents the image quality map of the restored images of Lena shown in the column **b**. Column **f** represents the image quality map of the restored images of circuit shown in column **c**

Fig. 5 Column **a** represents restored images of boat by corrupted image (60 %). Column **b** represents restored images of Lena corrupted image (60 %). Column **c** represents restored images of circuit corrupted image (60 %). Column **d** represents the image quality map of the restored images of boat shown in the column **a**. Column **e** represents the image quality map of the restored images of Lena shown in the column **b**. Column **f** represents the image quality map of the restored images of circuit shown in column **c**. Rows 1, 2, 3, 4, 5 and 6 represent the restored images after applying the median filter (MED-7 $\times$ 7), Adaptive Median Filter (AMED), MNASM, NAFSMF, MNASM (with HEIND) and EAWSM filters, respectively

Noise Filtering Stage

The proposed EAWSM filter is an improvement over NAWSM filter [17]. The proposed method uses an efficient noise detection method along with the adaptive weighted switching median filter. The restored images were with more prominent edges and retained much finer details in the image.

The EAWSM filter is applied to each pixel identified as noisy during detection stage. The noise mask or the binary decision mask generated by the HEIND [18] detection scheme will contain a value '0' in the noisy pixel position and a value '1' in the uncorrupted pixel position. The proposed filter is then applied to all the pixels in the image, for which the corresponding pixel in the binary decision mask is set as '0'.

First step in the filtering stage is to select appropriate window size for the filtering to achieve better results. The proposed EAWSM uses a square filtering window $W_F \times W_F$ with odd dimension as follows

$$W_F(i, j) = W_{2s+1}(i, j) = \{X(i + m, j + n)\} \quad (5)$$

where $m, n \in (-s, \dots, 0, \dots, s)$.

In order to determine the filter's window size $W_F \times W_F$, the limit of the maximum filtering window size $W_D \times W_D$ is to be determined first. It is determined by considering local noise distribution around each corrupted pixel. In the case of MNASM filter it was determined globally from the binary decision map obtained from the detection stage. Usually the noise distribution in an image is random. So the local estimation of noise density may be different from the global estimation. It is found that the local estimation of noise density is giving an optimum value for $W_D \times W_D$.

For determining the limit of the maximum filtering window size $W_D \times W_D$, a window of size $K \times K$ around each corrupted pixel is considered and percentage of noise that corrupts the image in that local window is calculated. This can be calculated by counting the number of zeros in the local region of the binary decision map that corresponds to noisy image and dividing it by total pixels in the

Fig. 6 Column **a** represents restored images of boat by corrupted image (90 %). Column **b** represents restored images of Lena corrupted image (90 %). Column **c** represents restored images of circuit corrupted image (90 %). Rows 1, 2, 3, 4, 5 and 6 represent the restored images after applying the median filter (MED-7 × 7), Adaptive Median Filter (AMED), MNASM, NAFSMF, MNASM (with HEIND) and EAWSM filters, respectively. Column **d** represents the image quality map of the restored images of boat shown in the column **a**. Column **e** represents the image quality map of the restored images of Lena shown in the column **b**. Column **f** represents the image quality map of the restored images of circuit shown in column **c**

$K \times K$ local window. The calculation of noise density p , for a local window $K \times K$ is given as shown below.

$$p = \frac{\sum_{i,j=1}^k Zc_k(i, j)}{K^2} \quad (6)$$

where Zc_k is the zero count in $K \times K$ window and is given as

$$Zc_k(i, j) = \begin{cases} 1, & N_k(i, j) = 0 \\ 0, & \text{Otherwise} \end{cases} \quad (7)$$

where N_k is the binary decision mask that corresponds to $K \times K$ window in noisy image.

After calculating noise density p , the limit of the maximum filtering window size, $W_D \times W_D$ is determined according to Table 1. Table 1 is empirically established based on multiple test images, in which different window sizes are suggested for different noise-density levels of corruption estimated.

After finding out the maximum window size $W_D \times W_D$, filtering is done. Initial filtering window size $W_F \times W_F$ is

set as $L \times L$. The filtering window is then iteratively extended outwards by one pixel in all the four sides of the window, provided that the number of uncorrupted pixels (denoted by N_c) is less than half of the total number of pixels (denoted by $S_{in} = 1/2[W_F \times W_F]$) within the filtering window, while $W_F \leq W_D$. Filtering window will also be extended when the number of uncorrupted pixels is equal to zero in that window. That is, while $((N_c < S_{in})$ and $(W_F \leq W_D))$ or $N_c = 0$, window will be extended outwards in all the four sides of the window. Then noisy pixels are replaced by a weighted median value of the uncorrupted pixels in the filtering window. Weight value assigned to each uncorrupted pixel depends on its closeness to the central pixel.

The weight value $w_{s,t}$ assigned to each uncorrupted pixel, that corresponds to the position (s, t) of the filtering window $W_F \times W_F$ is defined as follows.

$$w_{s,t} = \begin{cases} 3, & (s, t) \in \Omega^3 \\ 2, & (s, t) \in \Omega^5 \wedge (s, t) \notin \Omega^3 \\ 1, & \text{Otherwise} \end{cases} \quad (8)$$

Fig. 7 Graph representing the performance measures obtained after the applying the different filters on the corrupted boat image at difference noise densities (based on Noise Model 1)

where Ω^n denotes the $n \times n$ neighbourhood around the central pixel and is defined as

$$\Omega^n = \{(s, t) / -n \leq s, t \leq n\} \quad (9)$$

If the filtering window is reached to its maximum size and N_c , the number of uncorrupted pixels in the filtering window is equal to zero ($N_c = 0$), then the central noisy pixel is replaced by performing a conventional median filtering on already restored pixels in the filtering window $W_F \times W_F$.

So each noisy pixel $X(i, j)$ in the image is restored by a value $Y(i, j)$ defined as follows.

$$Y(i, j) = \begin{cases} \text{median}\{w_{s, t} \diamond X(i - s, j - t) / (s, t) \in W\} & \text{with} \\ N(i - s, j - t) = 1, & N_c > 0 \\ \text{median}\{X'(i - s, j - t) / (s, t) \in W\}, & N_c = 0 \end{cases} \quad (10)$$

where $W \{(s, t) / -(W_F - 1)/2 \leq s, t \leq (W_F - 1)/2\}$ and $X'(i, j)$ represents restored pixel at (i, j) th location in the filtering window

The algorithm for filtering can be summarized as below

- (i). Set the initial size of filtering window $W_F \times W_F$
- (ii). Calculate the local noise density p around each corrupted pixel [using Eq. (6)].
- (iii). Find the limit of the maximum window size $W_D \times W_D$ according to Table 1.
- (iv). While $((N_c > \text{Sin} \quad \text{and} \quad W_F \leq W_D) \text{ or } (N_c = 0))$, iteratively extends the filtering window size by 1 pixel outwards in all the four directions (N_c is the number of uncorrupted pixels in the filtering window and sin is half the number of pixels i the filtering window).

Fig. 8 Column **a** shows the edge maps of restored boat images corrupted with 30 % impulse noise. Column **b** shows the edge maps of restored boat images corrupted with 60 % impulse noise. Column **c** shows the edge maps of restored boat images corrupted with 90 % impulse noise. Column **a** shows the edge maps of restored Lena images corrupted with 30 % impulse noise. Column **b** shows the edge maps of restored Lena images corrupted with 60 % impulse noise. Column **c** shows the edge maps of restored Lena images corrupted with 90 % impulse noise. Rows 1, 2, 3, 4, 5 and 6 represent the restored images after applying the median filter (MED-7 × 7), Adaptive Median Filter (AMED), MNASM, NAFSMF, MNASM (with HEIND) and EAWSM filters, respectively

Fig. 9 Column **a** represents 200th row of original circuit image. Columns **b**, **c** and **d** represents 200th row of circuit image corrupted by 30 %, 60 % and 90 % noise (Noise Model 1)

- (v). Replace each noisy pixel $X(i, j)$ by a value $Y(i, j)$ defined as in Eq. (10)

Performance Measures

The performance of the restoration process is quantified using peak signal-to-noise ratio (PSNR), structured similarity index (SSIM) and image enhancement factor (IEF), is defined as follows.

$$\text{PSNR} = 10 \log_{10}(255^2 / \text{MSE})$$

$$\text{MSE} = \sum_{m,n} [O(m, n) - R(m, n)]^2 / (M \times N)$$

$$\text{SSIM} = L(O, R)C(O, R)S(O, R)$$

$$L(O, R) = (2\mu_O\mu_R + C_1) / (\mu_O^2 + \mu_R^2 + C_1)$$

$$C(O, R) = (2\sigma_O\sigma_R + C_2) / (\sigma_O^2 + \sigma_R^2 + C_2)$$

$$S(O, R) = (\sigma_{OR} + C_3) / (\sigma_O\sigma_R + C_3)$$

$$C_1 = (K_1G)^2, C_2 = (K_2G)^2, C_3 = C_2/2$$

$$G = 255; K_1, K_2 > 1, (K_1 = 0.001, K_2 = 0.002)$$

$$\text{IEF} = \left(\sum_{m,n} [P(m, n) O(m, n)]^2 \right) / \left(\sum_{m,n} [R(m, n) O(m, n)]^2 \right)$$

Fig. 10 Column **a** shows the edge maps of restored circuit images corrupted with 30 % impulse noise. Column **b** shows the edge maps of restored circuit images corrupted with 60 % impulse noise. Column **c** shows the edge maps of restored circuit images corrupted with 90 % impulse noise. Column **d** shows the 200th row of restored circuit images corrupted with 30 % impulse noise. Column **e** shows the 200th row of restored circuit images corrupted with 60 % impulse noise. Column **f** shows the 200th row of restored circuit images corrupted with 90 % impulse noise. Rows 1, 2, 3, 4, 5 and 6 represent the restored images after applying the median filter (MED-7 × 7), Adaptive Median Filter (AMED), MNASM, NAFSMF, MNASM (with HEIND) and EAWSM filters, respectively

where O is the original image; R , the restored image; P , the corrupted image; MSE, the mean square error; $M \times N$, the size of the image; L , the luminance comparison; C , the contrast comparison; S , the structure comparison; μ , the mean; and σ is the standard deviation.

In this paper, a qualitative-based performance measure named image quality index (IQI) is also used to prove the efficiency of our proposed EAWSM algorithm. This universal objective image quality index was proposed by Wang and Bovik [19], which is easy to calculate and applicable to various image processing applications. IQI is designed by modelling any image distortion as a combination of three factors: loss of correlation, luminance distortion and contrast distortion.

$$IQI_j = \text{Corr}(Ow, Rw) \times \text{Lum}(Ow, Rw) \times \text{Cont}(Ow, Rw)$$

IQI is first applied to local regions using a sliding window approach with size 8×8 . At the j th step, the local quality index IQI_j is computed within the sliding window using the formula given earlier. Ow and Rw represent the

sliding window of original and restored images, respectively. If there are a total of M steps, then the overall image quality index is given by

$$IQI = \frac{1}{M} \sum_j IQI_j$$

where j varies from 1 to M . The dynamic range of IQI is $[-1, 1]$, and the best value 1 is achieved if and only if restored image R is equal to the original image O .

Results and Discussion

The performance of EAWSM filter has been evaluated qualitatively and quantitatively through experimental analysis. Although extensive simulations were carried out using standard test images, only performance evaluation using images such as Lena image of size 512×512 , Peppers image of size 384×512 Goldhill image of size 512×512 , Circuit image of size 280×272 and Boat image of size 512×512 are explained here. To evaluate

the performance of the proposed EAWSM algorithm, edge maps of restored images are also computed, which has been compared with the edge map of the original image. The Canny operator was used for calculating edge maps. The superiority of the EAWSM algorithm over other famous denoising algorithms has been proved through performance measure graphs. To further convince the performance of proposed algorithm, a particular row (one dimensional signal) of original, corrupted and restored image has been shown in the form of a graph for each of the impulse noise models. The proposed algorithm along with HEIND detector [16] (EAWSM) was compared with median filter (MED) [2], adaptive median filter (AMED) [6], MNASM filter with BDND detector [14] (B-MNASM) and HEIND detector [18] (H-MNASM). In the case of Noise Models 1 and 2, NAFSMF [16] and EDBA [7] filters are also used for comparisons.

All the noise models described are used for measuring the performance of the proposed algorithm. For Noise Model 1 and Noise Model 3 images are corrupted with salt and pepper noise with different noise densities ranging from 10 % to 90 %. For Noise Model 2 and Noise Model 4, different noise densities for salt and pepper are considered.

For Noise Model 3 and Noise Model 4 the low intensity values in the range 0 to 9 and high intensity values in the range 246 to 255 is used. In our experiment, value of K as 7, value of L as 3 and maximum value of W_F as 21 are taken.

The original Boat, Lena and Circuit images and the corresponding corrupted images with 30, 60 and 90 % impulse noise based on the Noise Model 1 are shown in Fig. 2. Fig. 3 shows the edge maps (obtained using Canny operator) of original and corrupted images shown in Fig. 2. Tables 2, 3, 4 show the performance measure values such as PSNR, IEF, SSIM and IQI obtained after applying various filters on the above mentioned corrupted images (Lena and Circuit). Table 5 shows the PSNR performance measures of Boat image corrupted by noise densities varying from 10 % to 90 % (using Noise Model 1). Tables 6, 7, 8 show performance parameter values IEF, SSIM and IQI of Boat image corrupted by Noise Model 1. From the quantitative values shown in the tables, it is very clear that proposed EAWSM algorithm outperforms all other noise removal filters with enhanced edges and improved visual quality. Figures 4, 5, 6 shows the restored images obtained after applying the various filters, and their

Fig. 11 Column **a** represents original Peppers image, Peppers image corrupted by 80 % noise (30 % salt, 50 % pepper) and 60 % noise (20 % salt, 40 % pepper) based on Noise Model 2. Column **b** represents edge maps and column **c** represents 278th row of images in column **a**

corresponding image quality maps. It can be seen that the quality map of EAWSM is brighter compared to others, especially for high-density impulse noises such as 60 and 90 %. Also the images restored by our proposed method are with more prominent edges. Figure 7 graphically demonstrates supremacy of proposed algorithm over other filtering algorithm. Figure 8 shows the edge map of the restored images (Boat and Lena images) after applying the EAWSM and other filters on the corrupted test images.

Figure 9 shows the 200th row of original circuit image and the corrupted circuit image with noise densities 30, 60 and 90 % (using Noise Model 1). Figure 10 shows edge maps and 200th row of restored circuit image using various filtering algorithm mentioned earlier. It can be seen that there is a close resemblance between the row vectors of original circuit image and restored image obtained using our proposed algorithm. From this analysis it is clear that restored image obtained using proposed algorithm is very much closer to the original one.

Fig. 12 Column **a** represents restored images of corrupted peppers image (80 %) based on Noise Model 2 (30 % salt, 50 % pepper). Column **b** represents the corresponding image quality maps. Column **c** represents edge maps of the restored image. Column **d** represents the 278th row of the restored images. Rows 1, 2, 3, 4, 5 and 6 represent the restored images after applying the median filter (MED-7 $\times$ 7), Adaptive Median Filter (AMED), MNASM, EDBA, MNASM (with HEIND) and EAWSM filters, respectively

Fig. 13 Column **a** represents restored images of corrupted peppers image (60 %) based on Noise Model 2 (20 % salt, 40 % pepper). Column **b** represents the corresponding image quality maps. Column **c** represents edge maps of the restored image. Column **d** represents the 278th row of the restored images. Rows 1, 2, 3, 4, 5 and 6 represent the restored images after applying the median filter (MED-7 $\times$ 7), Adaptive Median Filter (AMED), MNASM, EDBA, MNASM (with HEIND) and EAWSM filters, respectively

For Noise Model 2, the performance of the algorithm was analysed using peppers image. Fig. 11 shows the original and noisy Peppers image (corrupted by 30 % salt and 50 % pepper noise and 20 % salt and 40 % pepper noise), corresponding edge maps (using Canny operator) and 278th row of corresponding images. Figs. 12 and 13 show the restored images, corresponding edge maps, quality

maps and 278th row of restored images obtained using different filters. Tables 9 and 10 show various performance measures obtained using different filters mentioned earlier.

The performance analysis for images corrupted by Noise Model 3 was performed using Lena image. Fig. 14 shows the original and corrupted Lena image (corrupted by 30, 60 and 90 % noise densities), corresponding edge maps (using

Table 9 PSNR and IEF values obtained for peppers image based on Noise Model 2

	20 % Salt, 60 % Pepper		15 % Salt, 45 % Pepper		10 % Salt, 30 % Pepper		5 % Salt, 15 % Pepper	
	PSNR	IEF	PSNR	IEF	PSNR	IEF	PSNR	IEF
MED	6.111045	1.181735	13.80451	5.412807	25.31103	55.2902	31.40361	112.1336
AMED	8.246893	1.928821	16.90648	11.08367	30.34671	174.4486	38.66351	597.5836
MNASM	27.778	173.6171	32.22679	376.3697	36.57075	734.0402	41.02567	1024.833
BNAWSM	28.98222	229.0946	34.00419	566.7022	37.22748	853.872	41.54025	1153.749
HMNASM	29.63024	265.2852	32.85442	435.0997	37.48233	912.5726	42.96026	1610.408
EAWSM	30.71888	340.8612	34.22506	596.5581	37.96694	1020.302	42.98272	1618.757

Table 10 SSIM and IQI values obtained for peppers image based on Noise Model 2

	20 % Salt, 60 % Pepper		15 % Salt, 45 % Pepper		10 % Salt, 30 % Pepper		5 % Salt, 15 % Pepper	
	SSIM	IQI	SSIM	IQI	SSIM	IQI	SSIM	IQI
MED	0.168421	0.02485	0.61879	0.280937	0.959419	0.609556	0.989228	0.69497
AMED	0.272006	0.072876	0.764778	0.510601	0.986537	0.852811	0.997922	0.930023
MNASM	0.974566	0.579686	0.990835	0.804488	0.996619	0.918587	0.998753	0.967452
BNAWSM	0.980913	0.654722	0.993939	0.857003	0.99709	0.92522	0.998886	0.968133
HMNASM	0.983268	0.680107	0.992076	0.816491	0.997252	0.922727	0.999181	0.971741
EAWSM	0.987131	0.755085	0.99424	0.863543	0.997538	0.929067	0.999185	0.971877

Fig. 14 **a, d, g** and **j** represents original Lena image. Lena image corrupted by 30, 60 and 90 % noise (Noise Model 3 with $m = 10$). **b, e, h** and **k** represents edge maps and **c, f, i** and **l** represents 278th row of the images **a, d, g** and **j** respectively

Canny operator) and 278th row of corresponding images. Figs. 15, 16, 17 show the restored images, edge maps, quality maps and 278th row vector obtained using different filters mentioned earlier. Tables 11 and 12 show various performance measures obtained using different filters mentioned earlier. Figs. 18 and 19 graphically demonstrate performances of different filtering methods.

For Noise Model 4, the performance of the algorithm was analysed using Goldhill image. The original and corrupted Goldhill image (corrupted by 30 % salt and 50 % pepper noise and 20 % salt and 40 % pepper noise),

corresponding edge maps (using Canny operator) and 278th row of corresponding images are shown in Figs. 20 and 21, respectively. The restored images, corresponding edge maps, quality map and 278th row of restored images obtained using different filters are shown in Figs. 22 and 23, respectively. Various performance measures obtained using different filters mentioned earlier are shown in Tables 13 and 14, respectively.

By analyzing the restored images, edge maps, quality maps and row vectors it is clearly evident that images closer to the original image are produced by the proposed

Fig. 15 Column **a** represents restored images of corrupted Lena image (30 %) based on Noise Model 3 (with $m = 10$). Column **b** represents the corresponding image quality maps. Column **c** represents edge maps of the restored image. Column **d** represents the 278th row of the restored images. Rows 1, 2, 3, 4, and 5 represent the restored images after applying the median filter (MED-7 $\times$ 7), Adaptive Median Filter (AMED), MNASM, MNASM (with HEIND) and EAWSM filters, respectively

method. From the tabular representation of performance measures and graphical representations, it can be seen that the proposed method outperforms other existing methods.

A comparison between the BDND [14] algorithm and HEIND [18] algorithm was also made by considering two evaluation indices, miss detection and false alarms. Miss detection represents number of noise being misdetected and false alarm represents number of noise free pixels that are misclassified as noisy. Table 15 represents the values obtained for a Lena image corrupted by various noise densities ranging from 20 % to 90 %. From the table it is clearly evident that the HEIND [18] detection method outperforms the BDND [14] method. Hence the HEIND

[18] increases the efficiency of EAWSM. A time complexity analysis was also performed for the proposed method and it is found that for an image of size $M \times N$ and filtering window of size W , time complexity of detection stage and detection stage is calculated as $O(MN)$ and $O(MN + W)$, respectively. Hence time complexity of proposed filter is estimated as $O(MN + W)$.

The proposed algorithm can be extended to denoise color images using the scalar median filtering approach. In scalar approach, each color component is taken as an independent entity, that is, here same filtering method will be applied to R-, G- and B-planes independently, as each plane is a separate monochrome image. The filtered

Fig. 16 Column **a** represents restored images of corrupted Lena image (60 %) based on Noise Model 3 (with $m = 10$). Column **b** represents the corresponding image quality maps. Column **c** represents edge maps of the restored image. Column **d** represents the 278th row of the restored images. Rows 1, 2, 3, 4, and 5 represent the restored images after applying the median filter (MED-7 $\times$ 7), Adaptive Median Filter (AMED), MNASM, MNASM (with HEIND) and EAWSM filters, respectively

R-, G- and B-planes will be then combined to form the restored color image. In order to explain the efficiency of filtering algorithm, color image of Lena and Peppers corrupted by Noise Model 4 is used. The original Lena and peppers color images and their corrupted versions are shown in Fig. 24. In this case, images are corrupted with 75 % of impulse noise based on model 4 ($m = 10$), in which 25 % is of pepper noise and 50 % is of salt noise. The restored images of the corrupted Lena and Peppers color images by different filters are shown in Fig. 25. Various performance measures obtained are shown in Table 16. From these results, it is very clear that EAWSM gives a better restored image in the case of color images also.

Conclusion

In this paper, an EAWSM filter is proposed for the restoration of images that are corrupted with high density of impulse noises based on different noise models. EAWSM is an improved switching median filter in which the filtering is applied only to corrupted pixels in the image while the uncorrupted pixels are left unchanged. The corrupted pixels are identified using HEIND detection. The advantage of EAWSM filter is that it is retaining the edge information in the case of high density impulse noises. The proposed method is found to be retaining finer details in the image and the images restored have an improved visual quality.

Fig. 17 Column **a** represents restored images of corrupted Lena image (90 %) based on Noise Model 3 (with $m = 10$). Column **b** represents the corresponding image quality maps. Column **c** represents edge maps of the restored image. Column **d** represents the 278th row of the restored images. Rows 1, 2, 3, 4, and 5 represent the restored images after applying the median filter (MED-7 $\times$ 7), Adaptive Median Filter (AMED), MNASM, MNASM (with HEIND) and EAWSM filters, respectively

Table 11 PSNR and IEF values obtained for Lena image based on Noise Model 3

Noise (%)	PSNR					IEF				
	MED	AMED	MNASM	HMNASM	EAWSM	MED	AMED	MNASM	HMNASM	EAWSM
90	8.766	10.647	24.716	25.218	25.831	1.816	2.705	70.88	80.834	93.086
80	13.33	16.053	26.697	26.981	28.114	4.358	8.706	98.751	108.636	141.028
70	18.53	21.924	28.065	28.334	29.806	12.82	30.074	119.38	127.013	178.251
60	22.14	27.23	29.506	29.762	31.392	26.401	86.586	146.694	152.886	222.52
50	23.65	30.09	31.355	31.708	32.897	29.967	139.371	187.534	196.42	258.248
40	24.51	31.793	33.847	34.117	34.642	29.86	162.769	269.099	278.766	314.577
30	25.48	33.85	36.142	36.5	36.656	29.834	201.902	329.521	355.627	368.634
20	27	35.97	38.701	39.225	39.26	27.836	207.947	405.128	440.657	444.206
10	28.47	38.18	42.586	42.942	42.941	19.055	166.142	470.841	548.293	548.18

Table 12 SSIM and IQI values obtained for Lena image based on Noise Model 3

Noise (%)	SSIM					IQI				
	MED	AMED	MNASM	HMNASM	EAWSM	MED	AMED	MNASM	HMNASM	EAWSM
90	0.204	0.343	0.951	0.956	0.963	0.022	0.057	0.472	0.484	0.534
80	0.541	0.713	0.969	0.971	0.978	0.123	0.271	0.591	0.599	0.656
70	0.820	0.913	0.977	0.979	0.985	0.325	0.549	0.677	0.684	0.736
60	0.916	0.973	0.984	0.985	0.990	0.463	0.703	0.751	0.759	0.796
50	0.940	0.986	0.990	0.990	0.992	0.518	0.777	0.820	0.826	0.845
40	0.950	0.991	0.994	0.994	0.995	0.538	0.827	0.876	0.881	0.885
30	0.960	0.994	0.996	0.997	0.997	0.553	0.865	0.916	0.920	0.921
20	0.971	0.996	0.998	0.998	0.998	0.563	0.887	0.948	0.951	0.951
10	0.979	0.998	0.999	0.999	0.999	0.569	0.894	0.974	0.977	0.977

Fig. 18 Graph representing the performance measures PSNR and IEF obtained after the applying the different filters on the corrupted Lena image at difference noise densities (based on Noise Model 3) **Fig. 19** Graph representing the performance measures SSIM and IQI obtained after the applying the different filters on the corrupted Lena image at difference noise densities (based on Noise Model 3)

Fig. 20 Column **a** represents original Goldhill image, **b** and **c** represents edge maps and 278th row of images in **a**

Fig. 21 **a** and **d** represents Goldhill image corrupted by 80 % noise (30 % salt 50 % pepper) and 60 % noise (20 % salt, 40 % pepper) based on Noise Model 4 (with $m = 10$), **b** and **e** represents the edge maps and **c** and **f** shows 278th row of **a** and **c**

Fig. 22 First column represents restored images of corrupted Goldhill image (80 %) based on Noise Model 4 (30 % salt, 50 % pepper). Second column represents the corresponding image quality maps. Third column represents edge maps of the restored image. Fourth represents the 278th row of the restored images. Rows 1, 2, 3, 4, and 5 represent the restored images after applying the median filter (MED-7 $\times$ 7), Adaptive Median Filter (AMED), MNASM, MNASM (with HEIND) and EAWSM filters, respectively

Fig. 23 Column **a** represents restored images of corrupted Goldhill image (60 %) based on Noise Model 4 (20 % salt, 30 % pepper). Column **b** represents the corresponding image quality maps. Column **c** represents edge maps of the restored image. Column **d** represents the 278th row of the restored images. Rows 1, 2, 3, 4, and 5 represent the restored images after applying the median filter (MED-7 $\times$ 7), Adaptive Median Filter (AMED), MNASM, MNASM (with HEIND) and EAWSM filters, respectively

Table 13 PSNR and IEF values of Goldhill image based on Noise Model 4

	20 % Salt, 60 % Pepper		15 % Salt, 45 % Pepper		10 % Salt, 30 % Pepper		5 % Salt, 15 % Pepper	
	PSNR	IEF	PSNR	IEF	PSNR	IEF	PSNR	IEF
MED	7.700285	1.341683	14.98689	5.502156	23.24984	25.51234	26.50721	27.00208
AMED	10.11358	2.311701	18.49932	11.8256	28.1278	78.49726	32.98226	124.2058
MNASM	25.30208	79.25974	28.2371	115.5824	31.82888	179.5029	36.19072	262.3007
HMNASM	26.21738	97.0212	28.39536	122.1757	32.00533	193.2769	36.51104	270.7033
EAWSM	27.16348	120.636	29.76214	167.3643	32.38997	211.1756	36.53296	272.0728

Table 14 SSIM and IQI values of Goldhill image based on Noise Model 4

	20 % Salt, 60 % Pepper		15 % Salt, 45 % Pepper		10 % Salt, 30 % Pepper		5 % Salt, 15 % Pepper	
	SSIM	IQI	SSIM	IQI	SSIM	IQI	SSIM	IQI
MED	0.232305	0.025062	0.690678	0.217595	0.940467	0.425555	0.969641	0.48709
AMED	0.392254	0.090722	0.838759	0.491506	0.980205	0.787843	0.993277	0.88556
MNASM	0.959947	0.459025	0.979505	0.709547	0.991082	0.866923	0.99671	0.947449
HMNASM	0.967175	0.531349	0.980237	0.717767	0.991444	0.871488	0.996944	0.951239
EAWSM	0.973889	0.627618	0.985648	0.776954	0.99217	0.878948	0.996959	0.951418

Table 15 Number of miss detection and false alarms

Noise (%)	Lena				Peppers			
	Miss Detection		False Alarm		Miss Detection		False Alarm	
	BDND	HEIND	BDND	HEIND	BDND	HEIND	BDND	HEIND
20	0	0	2	0	0	0	370	173
40	0	0	5	0	0	0	370	126
60	0	0	6	0	0	0	290	83
80	21	0	10	0	12	0	127	39
90	190	0	5	0	51	0	336	18

Fig. 24 **a** and **c** shows the original color Lena image and peppers image. **b** and **d** shows 75 % (50 % salt, 25 % pepper) corrupted versions of **a** and **c** based on Noise Model 4 **Fig. 25** Columns **a**, **b**, **c**, **d**, and **e** represent the restored images after applying the median filter (MED- 7×7), Adaptive Median Filter (AMED), MNASM, MNASM (with HEIND) and EAWSM filters, respectively

Table 16 Performance measures of color Lena image and peppers image based on Noise Model 4

	Lena				Peppers			
	PSNR	IEF	SSIM	IQI	PSNR	IEF	SSIM	IQI
MED	8.533679	1.447702	0.321224	0.0377	10.90196	2.543459	0.374967	0.050095
AMED	13.55061	4.523463	0.650825	0.157631	10.74817	2.574992	0.514584	0.119821
MNASM	24.23487	55.45756	0.964783	0.537614	24.4014	53.55964	0.967646	0.588941
HMNASM	26.75229	102.6685	0.980033	0.60054	24.85759	60.26638	0.971101	0.591561
EAWSM	27.10319	111.308	0.981671	0.619554	25.37271	67.85586	0.974537	0.619032

References

1. P. Civicioglu, Using uncorrupted neighborhoods of the pixels for impulsive noise suppression with ANFIS, *IEEE Trans. Image Process.*, **16**(3), 759–773 (2007)
2. R.C. Gonzalez, R.E. Woods, *Digital image processing*, 3rd edn. (Prentice-Hall, Englewood Cliffs NJ, 2008)
3. T.S. Huang, G.J. Yang, G.Y. Tang, Fast two-dimensional median filtering algorithm, *IEEE Transactions on Acoust. Speech Signal Process.*, **ASSP-1**(1), 13–18 (1979)
4. A. Bovik, *Handbook of image and video processing* (Academic, New York, 2000)
5. S.-J. Ko, Y.H. Lee, Center weighted median filters and their applications to image enhancement. *IEEE Trans. Circ. Syst.* **38**(9), 984–993 (1991)
6. H. Hwang, R.A. Haddad, Adaptive median filters: new algorithms and results. *IEEE Trans. Image Process.* **4**(4), 499–502 (1995)
7. K.S. Srinivasan, D. Ebenezer, A new fast and efficient decision based algorithm for removal of high-density impulse noises. *IEEE Signal Process. Lett.* **14**(3), 189–192 (2007)
8. M.S. Nair, P.M.A. Mol, Direction based adaptive weighted switching median filter for removing high density impulse noise. *Comput. Electr. Eng.* Elsevier **39**(2), 663–689 (2013)
9. W. Luo, Efficient removal of impulse noise from digital images. *IEEE Trans. Consumer Electron.* **52**(2), 523–527 (2006)
10. S. Schulte, M. Nachtegaal, V.D. Witte, D.V. Weken, E.E. Kerre, A fuzzy impulse noise detection and reduction method. *IEEE Trans. Image Process.* **15**(5), 1153–1162 (2006)
11. S. Schulte, V.D. Witte, M. Nachtegaal, D.V. Weken, E.E. Kerre, Fuzzy two-step filter for impulse noise reduction from color images. *IEEE Trans. Image Process.* **15**(11), 3567–3578 (2006)
12. M.S. Nair, G. Raju, A new fuzzy-based decision algorithm for high-density impulse noise removal, Springer-Verlag London Limited 2010, *Signal Image and Video processing*, DOI [10.1007/s11760-010-0186-4](https://doi.org/10.1007/s11760-010-0186-4)
13. Y.-T. Zhou, R. Chellappa, A. Vaid, B.K. Jenkins, Image restoration using a neural network. *IEEE Trans. Acoust. Speech Signal Process.* **36**(7), 1141–1151 (1989)
14. P.-E. Ng, K.-K. Ma, A switching median filter with boundary discriminative noise detection for extremely corrupted images. *IEEE Trans. Image Process.* **15**(6), 1506–1516 (2006)
15. H.-L. Eng, K.-K. Ma, Noise adaptive soft-switching median filter. *IEEE Trans. Image Process.* **10**(2), 242–251 (2001)
16. K.K.V. Toh, N.A.M. Isa, Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction. *IEEE Signal Process. Lett.* **17**(3), 281–284 (2010)
17. M.S. Nair, P.M. Ameera Mol, Noise adaptive weighted switching median filter for removing high density impulse noise, First International Conference on Advances in Computing and Communications-ACC 2011 (Kochi), Communications in Computer and Information Science (CCIS), Springer, Part III, 192, 193–204, July 2011
18. Duan F, Zhang Y-J (2010) A Highly effective impulse noise detection algorithm for switching median filters, *IEEE Signal Process. Lett.*, **17**(7), 647–650
19. Z. Wang, A.C. Bovik, A universal image quality index. *IEEE Signal Process. Lett.* **9**(3), 81–84 (2002)