EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

An adaptive dynamically weighted median filter for impulse noise removal

Sajid Khan[1] and Dong-Ho Lee[2*]

Abstract

A new impulsive noise removal filter, adaptive dynamically weighted median filter (ADWMF), is proposed. A popular method for removing impulsive noise is a median filter whereas the weighted median filter and center weighted median filter were also investigated. ADWMF is based on weighted median filter. In ADWMF, instead of fixed weights, weightages of the filter are dynamically assigned with the results of noise detection. A simple and efficient noise detection method is also used to detect noise candidates and dynamically assign zero or small weights to the noise candidates in the window. This paper proposes an adaptive method which increases the window size according to the amounts of impulsive noise. Simulation results show that the AMWMF works better for both images with low and high density of impulsive noise than existing methods work.

Keywords: Impulsive noise, Median filter, Weighted median filter, Noise detection, Adaptive filter

1 Introduction

Image processing is used in many scientific fields such as astronomy, aerospace, photogrammetry, particle physics, biology, medical sciences, geology, and science of materials [1]. Removing noise in an image is important for improving image quality. Some filters like H-infinity filter have been applied to practical industrial systems [2–4]. Usually, noise originates from taking pictures through a defective sensor or transmitting images through a noisy channel [5]. Such noise can be categorized into many different types by probability distributions, such types being impulsive noise, Gaussian noise, Rayleigh noise, and Laplacian noise, to name a few [4, 6, 7].

Impulse noise is a random variation of image intensity. The presence of impulse noise in image replaces intensities of corrupted pixels by extreme values. Mathematical model of impulse noise is given in (1)

$$Y = \begin{cases} 0, & \text{with probability } p_1 \\ L{-}1, & \text{with probability } p_2 \\ x, & \text{with probability } 1{-}p_1{-}p_2 \end{cases} \tag{1}$$

where x denotes noise-free pixels of corrupted image.

Addition of impulse noise to images or frames of a video is a problem that occurs in TV [8], CCTV, and MRI [6] due to many reasons like unreliable sensor, power, and channel. Because impulsive noise seriously deteriorates picture quality and disrupts image processing, an efficient noise-removal algorithm is a core image processing technology.

Applying linear filters to image corrupted by impulsive noise leads to image blurring [9, 10]. To address the problem of impulsive noise, researchers focused on applying nonlinear filters. Tukey [11] introduced a powerful nonlinear filter called median filter, and many improved filters have been proposed by other researchers. Huang et al. [12] described an efficient method to apply Median to 2-D images. Yin and Yang [13], Ko and Lee [14] proposed weighted median, and center weighted median filters respectively that apply weights to the window. Median filter with adaptive window size was used by Hwang and Hadded [15], Chang et al. [16], Juneja and Mohana [17] and Ibrahim et al. [18] to propose adaptive median filter (AMF) and its different types.

Toh and Isa [19] proposed a modified adaptive median filter that extends window size when current window does not have any noise-free pixel. Bhateja et al. [20] proposed a method similar to [19]; however, the maximum allowable size of window became 9 × 9. If 9 × 9 window surrounding noisy pixel does not have any known noise-free pixel, it replaces the noisy pixel with previously processed

* Correspondence: dhlee77@hanyang.ac.kr
[2]Department of Electronic and Communication Engineering, Hanyang University ERICA Campus, Ansan, South Korea
Full list of author information is available at the end of the article

or previous noise-free pixel. Bhadouria et al. [21] proposed a median-based method that extends the window up to maximum allowable size of 9 × 9 when there is no noise-free pixel in the window. If 9 × 9 window does not have noise-free pixels, it replaces the corrupted pixel by mean of all pixels in 3 × 3 window. Afzal et al. [22] simply replaces the corrupted pixel with the next noise-free pixel.

Many researches used two-stage approach that first detects location of impulse noise and then apply median or other filters derived from median on those detected locations only to avoid blurring. To detect impulsive noise, Vijaykumar et al. [23] used absolute difference of all the pixels of 3 × 3 window with the center pixel and decides the center pixel to be noisy if less than two differences are greater than threshold. This algorithm fails in regions with high density of noise as more than two differences will be greater that threshold. Jampur et al. [24] and Luo [25] used fuzzy-based techniques to detect impulse noise and then applies heuristic median and median filters respectively on those detected pixels. Heuristic median filter uses average of non-noisy pixels that results in an average filter which causes blurring. Luo [25] uses degree of membership in replacing noisy pixel that may result in a value that does not exist in that region.

Duan and Zhang [26] divides the sorted window across every pixel in two intervals [0, med] and [med, 255] and compare the pixel intensity with locations corresponding to maximum difference in those two intervals to select noise candidates. Noise candidates are further processed with four convolutional kernels as used by Zhang and Karim [27] along the four directions. In algorithm proposed by Zhang and Karim [27], if a particular direction have excess of noise, the absolute difference will be smaller in that direction which will results the value to be smaller than the threshold, resulting in detection as non-noisy pixel. Ng and Ma [28] first detects noise candidate the same as Duan and Zhang [26] did; it then applies the same technique to noise candidates but using 3 × 3 window this time to have final selection of noisy pixels. Lee [29] uses absolute difference of median of 5 × 5 window and average of 3 × 3 window with center pixel and compares them with a threshold. It is a time efficient algorithm for low-density noise. Ghanekar [30] detects candidates by comparing pixel with extreme values in a 3 × 3 window, then performs the final detection by choosing minimum and maximum locations among locations of four maximum differences, and compares pixel with intensity values at those minimum and maximum locations.

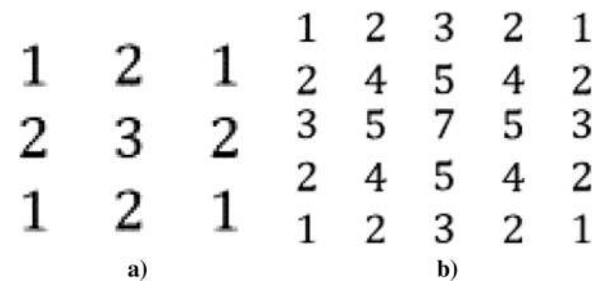
Some researchers used techniques other than median filter for removing impulse noise. Deng and An [31] and Sadri et al. [32] used wavelet. Wang et al. [33] and Liu et al. [34] used dictionary learning-based impulse noise removal techniques. Dictionary learning-based techniques are very complex and result in changing properties of input image as they replace patches of input image with patches stored in predefined dictionary. Wavelet-based methods, on the other hand, are far more complex. Median-based methods are simple methods that are powerful for removal of impulse noise and preserve edges well.

In this paper, adaptive dynamically weighted median filter (ADWMF) is proposed which combines weighted median filter with an impulse noise detection algorithm. Previous switching filters use static window size for a noisy image whose size depends on the total noise density of the image. Such techniques fail to remove noise when certain regions have comparatively excessive amount of noise due to uneven distribution of noisy pixels in the corrupted image. Conventional adaptive filters, on the other hand, extend window size when there is no noise-free pixel in window or median of window is a noisy pixel. Such algorithms do not consider performance when extending the window size. Instead of using fixed window size for an image, the proposed method changes window size according to the amount of noisy pixels present in a region. The proposed method uses an efficiently modified version of simple noise detection method, and it is used to propose the ADWMF. After detecting impulsive noise candidates, the weighted median filter is modified by dynamically assigning zero weights to the noise candidates in the window. Finally, this paper proposes an adaptive filter which increases the window size according to the amounts of impulsive noise candidates in a region. Window size in each region is determined by using number of noisy pixels in that region. Thresholds for changing window size according to number of noisy pixels in a region are defined after extensive simulations that ensure best possible results on the basis of minimum mean square error. Simulation results show that ADWMF works better for both images with low and high density of impulsive noise than existing methods work.

The rest of this paper is organized as follows: Section 2 describes some of the conventional methods. Section 3 describes proposed impulse noise detection algorithm, proposed dynamically weighted median filter, and proposed adaptive dynamically weighted median filter (AMWMF). Section 4 shows the results and discussion of the proposed filter followed by conclusion in Section 5.

2 Problem statement

Before introducing some conventional and proposed filters, it is suitable to introduce some notations that will be used in this paper. Image will be represented by notation I. $X(i,j)$ will be used to represent an input pixel on which a particular impulse noise removal filter will be applied; notation $Y(i,j)$ will be used to represent the output of a particular filter to replace $X(i,j)$. A $(2N+1)\times(2N+1)$ window that is used across $X(i,j)$ for a particular filter is represented by W and is given in (2)



1	2	1
2	3	2
1	2	1

a)

1	2	3	2	1
2	4	5	4	2
3	5	7	5	3
2	4	5	4	2
1	2	3	2	1

b)

Fig. 1 Weighting windows used in the WM filter. **a** 3 × 3 window. **b** 5 × 5 window

$$W = \{(q,t)| -N \le q \le N, -N \le q \le N\} \tag{2}$$

So the pixels in window W can be represented by $X(i+q, j+t)$. Median filter proposed by Tukey [8] is given in (3)

$$Y(i,j) = \mathrm{MED}\{X(i+q, j+q)|(q,t) \in W\} \tag{3}$$

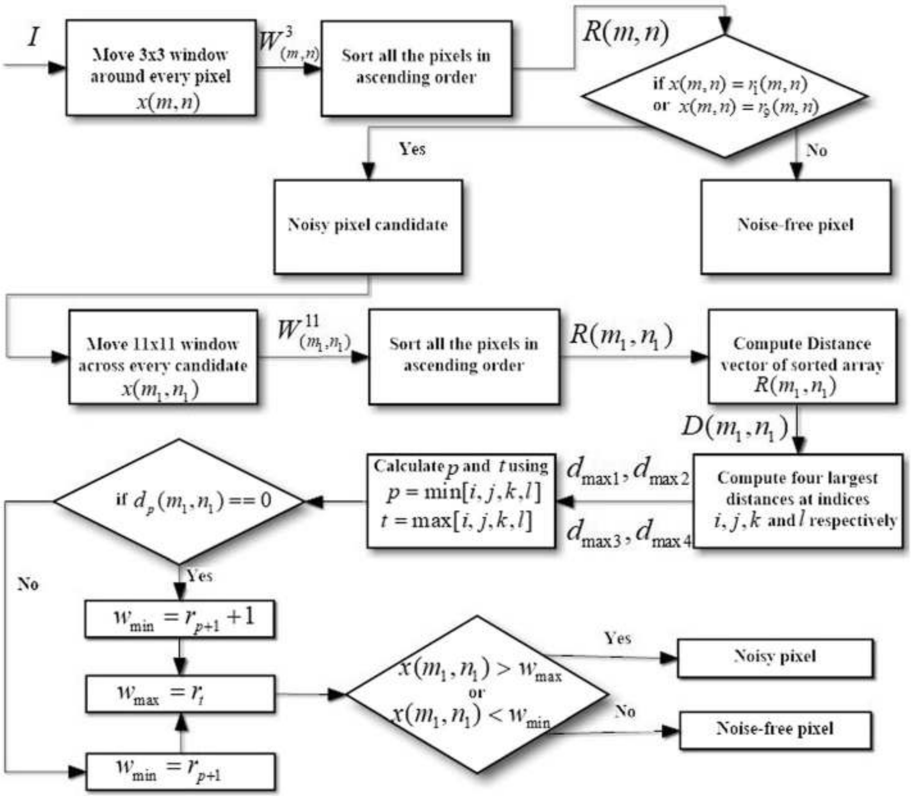
Adaptive median filter (AMF) [12, 13, 17, 18] uses median filters adaptively. It mostly increases the window size by comparing median value with extreme values of image. The problem with AMF is that there might have background pixels which equal the extreme values of image. In such a case, AMF will keep extending its window size until the window contains a median value that is not equal to one of two extreme values. To improve the performance, weighted median (WM) filter [13] was proposed by assigning weightage to each location with the help of weighted window. WM value of a pixel $X(i,j)$ is

$$WM(i,j) = \mathrm{MED}[X(i+q, j+t) \diamond W(i+q, j+t)|(q,t) \in W] \tag{4}$$

where ◊ is the repetition operator such that

$$K \diamond X = \overbrace{X, X, X, \ldots\ldots, X}^{K \text{ times}} \tag{5}$$

Figure 1 shows the weightage windows which are used for cases of 3 × 3 and 5 × 5 WM filters. It can be observed that locations near to center have high amount of weights compared to the locations that are away from the center. The sum of all weights in window should be odd. The center weighted median (CWM) [14] is a special case of WM which assigns weight to center pixel only. Duan et al. [35] proposed a method which uses a rank ordered mean (ROM) filter based on a noise detection. Steps for noise removal are as follows:



I
Move 3x3 window around every pixel $x(m,n)$
$W^3_{(m,n)}$
Sort all the pixels in ascending order
$R(m,n)$
if $x(m,n) = r_1(m,n)$ or $x(m,n) = r_9(m,n)$
Yes
No
Noisy pixel candidate
Noise-free pixel
Move 11x11 window across every candidate $x(m_1,n_1)$
$W^{11}_{(m_1,n_1)}$
Sort all the pixels in ascending order
$R(m_1,n_1)$
Compute Distance vector of sorted array $R(m_1,n_1)$
$D(m_1,n_1)$
Compute four largest distances at indices i, j, k and l respectively
$d_{\max1}, d_{\max2}$
$d_{\max3}, d_{\max4}$
Calculate p and t using $p = \min[i,j,k,l]$ $t = \max[i,j,k,l]$
if $d_p(m_1,n_1) == 0$
Yes
No
$w_{\min} = r_{p+1} + 1$
$w_{\max} = r_t$
$w_{\min} = r_{p+1}$
$x(m_1,n_1) > w_{\max}$ or $x(m_1,n_1) < w_{\min}$
Yes
No
Noisy pixel
Noise-free pixel

Fig. 2 Flowchart for impulsive noise detection

Table 1 False caught (FC) and true left (TL) comparison

	30%						50%						90%					
	Ghanekar		BDND		Proposed		Ghanekar		BDND		Proposed		Ghanekar		BDND		Proposed	
	FC	TL	FC	TL	FC	TL	FC	TL	FC	TL	FC	TL	FC	TL	FC	TL	FC	TL
Banana	0	0	0.75	0	0	0	0	0	0.41	0	0	0	0	0.15	0.19	1.01	0	0
Briefcase	2.7	1.75	4.23	0	3.1	0	1.34	2.54	2.14	0	1.33	0	1.14	3.43	0.74	1.34	1.12	0
Brush	13.78	3.98	15.1	0	13.3	0	7.34	4.38	7.76	0	7.04	0	6.14	4.84	1.4	1.53	5.86	0
Coffee	3.08	0.26	4.43	0	3.07	0	1.56	0.38	2.29	0	1.55	0	1.25	0.61	1.13	1.12	1.25	0
Egg	0.35	0	1.14	0	0.35	0	0.18	0.02	0.59	0	0.18	0	0.14	0.15	0.35	1.12	0.14	0
Flower	0.76	0.07	1.4	0	0.76	0	0.4	0.1	0.73	0	0.4	0	0.31	0.22	0.48	1.15	0.31	0
Golf cart	1.46	0.04	2.6	0	1.46	0	0.74	0.08	1.3	0	0.74	0	0.57	0.18	0.83	1.14	0.57	0
Grater	0.26	0	1.13	0	0.26	0	0.13	0	0.57	0	0.13	0	0.1	0.14	0.35	0.96	0.11	0
Kettle	0.09	2.41	2.82	0	0.98	0	0.01	3.03	1.33	0	0.07	0	0	4.11	0.42	0.73	0.02	0
Lena	0	0	0.55	0	0	0	0	0	0.34	0	0	0	0	0.06	0.27	0.97	0	0
Camera	0.1	0	0.74	0	0.1	0	0.04	0.01	0.39	0	0.04	0	0.03	0.16	0.32	1.29	0.03	0
Average	2.05	0.77	3.18	0	2.13	0	1.07	0.96	1.62	0	1.04	0	0	0.15	0.19	1.01	0	0

1) Sort all the pixels $X(i+q, j+t)$ except $X(i,j)$ itself to get $r = r_1, r_2, ..., r_{(2N+1)^2-1}$. If $X(i,j) > r_{(2N+1)^2-1}$ or $X(i,j) < r_1$, then select it as noisy pixel and apply ROM filter to it by skipping remaining steps for noise detection. If $X(i,j) = r_{(2N+1)^2-1}$ or $X(i,j) = r_1$, then select $X(i,j)$ as noise candidate and move to the second step. If none of the above conditions are satisfied, select the pixel as noise-free pixel.
2) Take the sum of absolute difference of $X(i,j)$ with elements in horizontal, vertical, main diagonal, and auxiliary diagonal with respect to $X(i, j)$ and store them in m_1 , m_2 , m_3, and m_4 respectively. Calculate $m = \min[m_1, m_2, m_3, m_4]$.
3) If $m > T$, then pixel $X(i,j)$ is a noisy pixel, and else, it is a noise-free pixel.
4) Apply ROM filter to pixels that are detected as a noise.

Since ROM filter uses average of two values in window, it blurs the image. Also, if a certain direction among four directions have excess of noise, the sum of absolute difference along that direction will be less than T. Noise detection algorithm will select $X(i,j)$ as noise-free pixel no matter it is noisy or not.

Ibrahim et al. [18] modified AMF to propose an adaptive filter named simple adaptive median filter (SAMF). SAMF uses non-noisy pixels for calculating median value of selected window. It first detects noisy pixels, extends the window if numbers of non-noisy pixels are less than 8, and replace the detected noisy pixel with median of non-noisy pixels in the window. Another modification is the automatic evaluation of minimum window size like

$$W_{\min} = 2R_{\min} + 1 \tag{6}$$

where

$$R_{\min} = \left\lfloor \frac{1}{2}\sqrt{\frac{7}{1-\sigma}} \right\rfloor \tag{7}$$

(M, N) = (row, column) in image, $\sigma = \frac{K}{MN}$, K is the number of detected noisy pixels in image, and $\lfloor\,\rfloor$ corresponds to rounding off to closest integer value. If number of "noise-free pixels" is even, median value will be

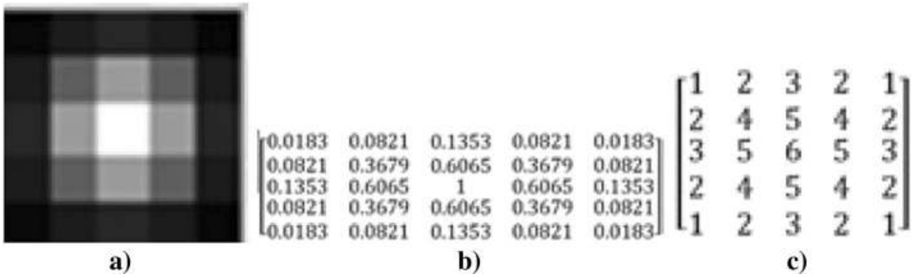


Fig. 3 A weighting window used in the WM filter. **a** 5 × 5 Gaussian surface. **b** Intensity values of a Gaussian surface. **c** Weighting window after normalization

replaced by the average of two values that does not exist in its neighborhood region.

3 Proposed work

Proposed filter consists of two steps, noise detection and noise removal. Both stages are incorporated to each other. Firstly, impulse noise is detected by the proposed efficient method, and one of 3 × 3, 5 × 5, and 7 × 7 dynamically weighted median filters (DWMs) is applied according to the number of noisy pixels in windows.

3.1 Proposed impulse noise detection

A new simple impulse noise detection technique is proposed by modifying impulse detection method proposed

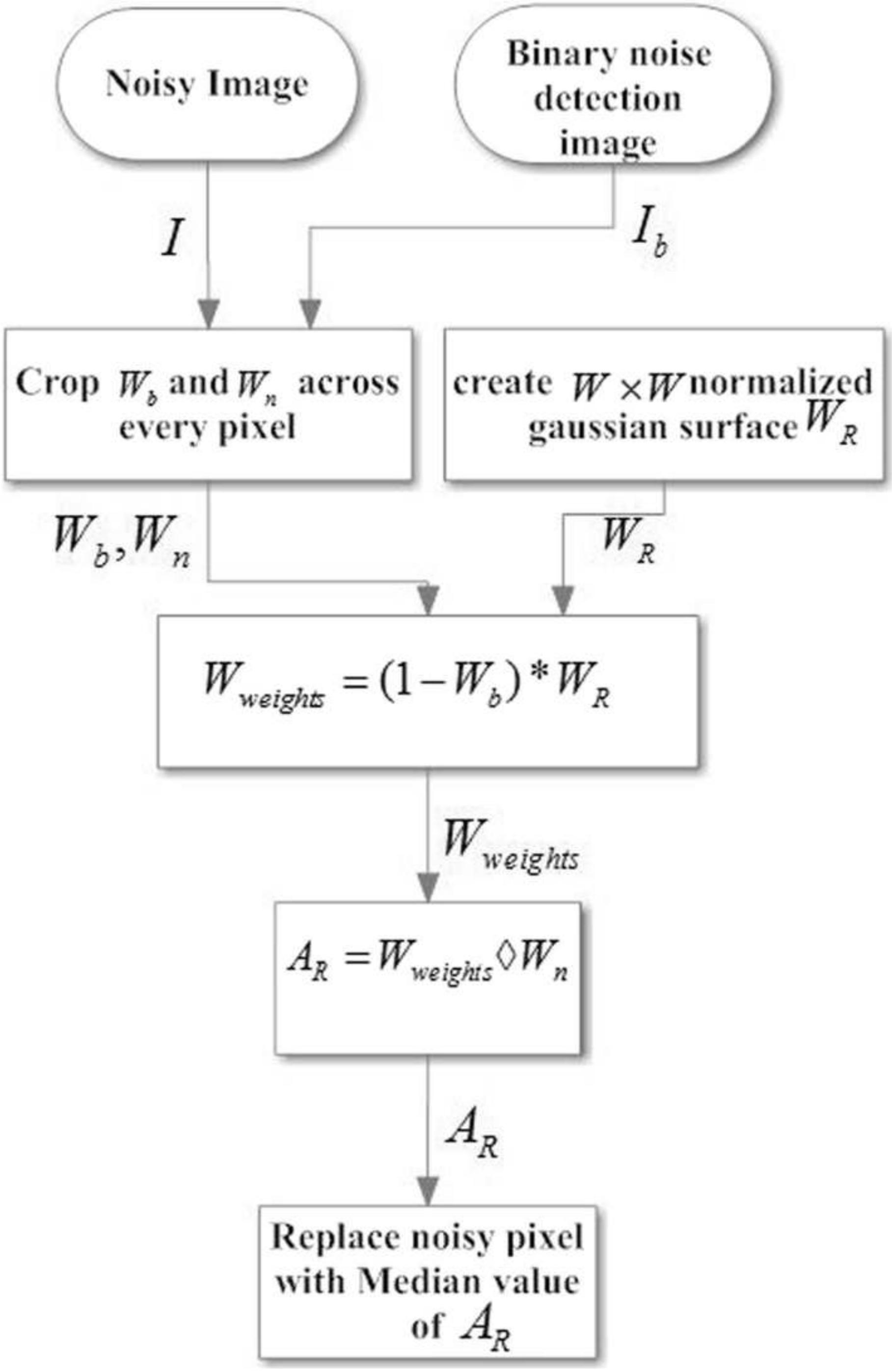

Noisy Image
Binary noise detection image
I
I_b
Crop W_b and W_n across every pixel
create $W \times W$ normalized gaussian surface W_R
W_b, W_n
W_R
$W_{weights} = (1 - W_b) * W_R$
$W_{weights}$
$A_R = W_{weights} \Diamond W_n$
A_R
Replace noisy pixel with Median value of A_R

Fig. 4 Flowchart of the proposed DWMF

by Ghanekar [30]. Figure 2 shows the whole flowchart of the proposed impulse noise detection method. The method in [30] is divided in two stages. First stage is based on noise candidate detection, and its steps are given as:

1) Move 3 × 3 window across every pixel $x(m, n)$ at location (m, n).
2) For each window $W^3_{(m,n)}$, order the pixels in ascending order to get $R(m, n) = [r_1(m, n), r_2(m, n), \ldots\ldots, r_9(m, n)]$.
3) If $x(m, n) = r_1(m, n)$ or $x(m, n) = r_9(m, n)$, then pixel $x(m, n)$ is a noisy candidate, and else, it is a noise free pixel.

Second stage is based on final selection of noisy pixels by processing noise candidates using steps given below

1) Move 11 × 11 window $W^{11}_{(m_1,;n_1)}$ across all noisy candidates at locations (m_1, n_1), and sort the window to get (8) as

$$R(m1, n1) = [r_1(m_1, , n_1), r_2(m_1, , n_1), \ldots\ldots, r_{121}(m_1, ; n_1)] \tag{8}$$

2) Compute the distance vector $D(m_1, n_1) = [d_1(m_1, n_1), d_2(m_1, n_1), \ldots\ldots, d_{120}(m_1, n_1)]$ where $d_i(m_1, n_1) = r_{(i+1)}(m_1, n_1) - r_i(m_1, n_1)$.
3) Find first the four largest distances $d_{\max1} = d_i(m_1, n_1)$, $d_{\max2} = d_j(m_1, n_1)$, $d_{\max3} = d_k(m_1, n_1)$, and $d_{\max4} = d_l(m_1, n_1)$ where $d_{\max1} > d_{\max2} > d_{\max3} > d_{\max4}$.
4) If $(x(m_1, n_1) < w_{min}\ or\ x(m_1, n_1) > w_{max})$, select $x(m_1, n_1)$ as noisy pixel, where $w_{\min} = r_{(p+1)}$ and $w_{\max} = r_t$. $p = \min[i, j, k, l]$ and $t = \max[i, j, k, l]$.

Method in [30] is an efficient algorithm, but its second stage has a minor problem that diverts true detection from 100% accuracy. Let us consider an example shown below. A 5 × 5 window is used instead of 11 × 11 for better understanding.

$$W^x_{(m_1,,n_1)} = \begin{vmatrix} 18 & 20 & 18 & 20 & 18 \\ 255 & 0 & 18 & 18 & 18 \\ 18 & 18 & 0 & 20 & 18 \\ 20 & 20 & 18 & 20 & 18 \\ 20 & 18 & 20 & 20 & 18 \end{vmatrix}.$$

If pixel $x(m_1, n_1)$ in a region with uniform values is affected by pepper noise, then $D(m_1, n_1)$ will contain less than four non-zero values as shown below.

$$\begin{aligned} &D(m_1, , n_1) \\ &\quad = [0, 0, 18, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 255] \end{aligned}$$

In this case, $d_{\max1} = 255$, $d_{\max2} = 18$, $d_{\max3} = 2$, and $d_{\max4} = 0$ with $i = 24$, $j = 3$, $k = 16$, and $l = 1$. Hence, $p = 1$ will results in $w_{\min} = r_2 = 0$. Since the condition $x(m_1, n_1) < w_{min}$ is not satisfied, the algorithm will results in detection of $x(m_1, n_1)$ as noise-free pixel. To avoid this drawback of detection method in [30], an extra step was added to algorithm in [30] after application of fourth step of stage.

$$\text{if } (dp(m_1, n_1) = 0) \text{ then} \atop \boldsymbol{w}_{\min} = \boldsymbol{r}_{p+1} \quad +1 \qquad\qquad \text{end if}$$

The results of proposed impulse noise detection were compared with boundary discriminative noise detection (BDND) [28] algorithm and Ghanekar method [30]. For comparison purpose, BDND [28] algorithm was selected because it provides 100% true detection whereas Ghanekar method [30] was selected to show improvement in results. Performance of an impulse noise detection algorithm depends on its capability to detect all the true positives effectively in the presence of false positives as less as possible. Because if a noise detector detects too much false positives, non-noisy pixels at the locations of false positives will be treated as noisy pixels; hence, their intensity values will be replaced by a new value causing the distortion of information along with increase in image blurring.

Table 1 shows false caught (FC) and true left (TL) comparison of BDND [28], Ghanekar [30], and proposed

Table 2 Average MSE (× 10^3) comparison for different nos. of noisy pixels

Filters	1	2	3	4	8	11	12	13	19	20	21
Median 3 × 3	0.66	0.93	1.20	1.58	201.17						
Median 5 × 5	0.99	1.05	1.11	1.24	1.39	1.62	1.82	2.11	47.03	53.31	50.43
Median 7 × 7	1.17	1.23	1.45	1.42	1.75	1.50	1.49	1.63	1.90	1.8	1.85
W median 3 × 3	0.75	1.10	1.45	20.131	219.25						
W median 5 × 5	0.83	0.90	0.95	1.08	1.25	1.92	2.95	5.15	74.12	73.78	64.52
W median 7 × 7	0.95	0.98	1.12	1.09	1.36	1.19	1.18	1.31	1.54	1.49	1.56
Proposed 3 × 3	0.63	0.64	0.78	0.95	297.3						
Proposed 5 × 5	0.79	0.8	0.86	0.91	0.94	1.02	1.07	1.14	2.14	1.90	2.55
Proposed 7 × 7	1.25	1.12	0.99	1.08	1.33	1.09	1.12	1.13	1.35	1.18	1.15

Table 3 Final selected filters for specific range of number of noisy pixels

No. of noisy pixels	1–3	4–12	13–onward
Best filter	DWMF 3 × 3	DWMF 5 × 5	DWMF 7 × 7

algorithm for 11 test images. It can be seen from Table 1 that false caught are minimum for proposed algorithm as compared to BDND and is nearly equal to Ghanekar. True left is 0% for BDND and proposed algorithm whereas 0–4.38% true left for Ghanekar as can be seen from Table 1.

3.2 Adaptive dynamically weighted median filter

Weighted median (WM) filter discussed in Section 2 is followed in this paper. Proposed algorithm addresses a problem that is faced by WM filter in case of presence of noise in nearest neighborhood of noisy pixel. In weighted median filter, if center pixel is noisy or its nearest neighbors are noisy pixels, those noisy pixels will get high weightage and hence will be repeated more. In such case, probability of selection of noisy pixel as median value becomes higher. Even if size of window increases in case of selection of 0 (lowest intensity) or $L - 1$(highest intensity) as median value, window will keep increasing until the sum of weightage for non-noisy pixels exceeds the sum of weightage of noisy pixels. A new dynamically weighted median filter (DWMF) assigns weightage of 0 to those locations in a W*W window that are detected as noisy pixels using proposed impulse noise detection algorithm. Before discussing steps for proposed adaptive dynamically weighted median filter (ADWMF), some statistics and

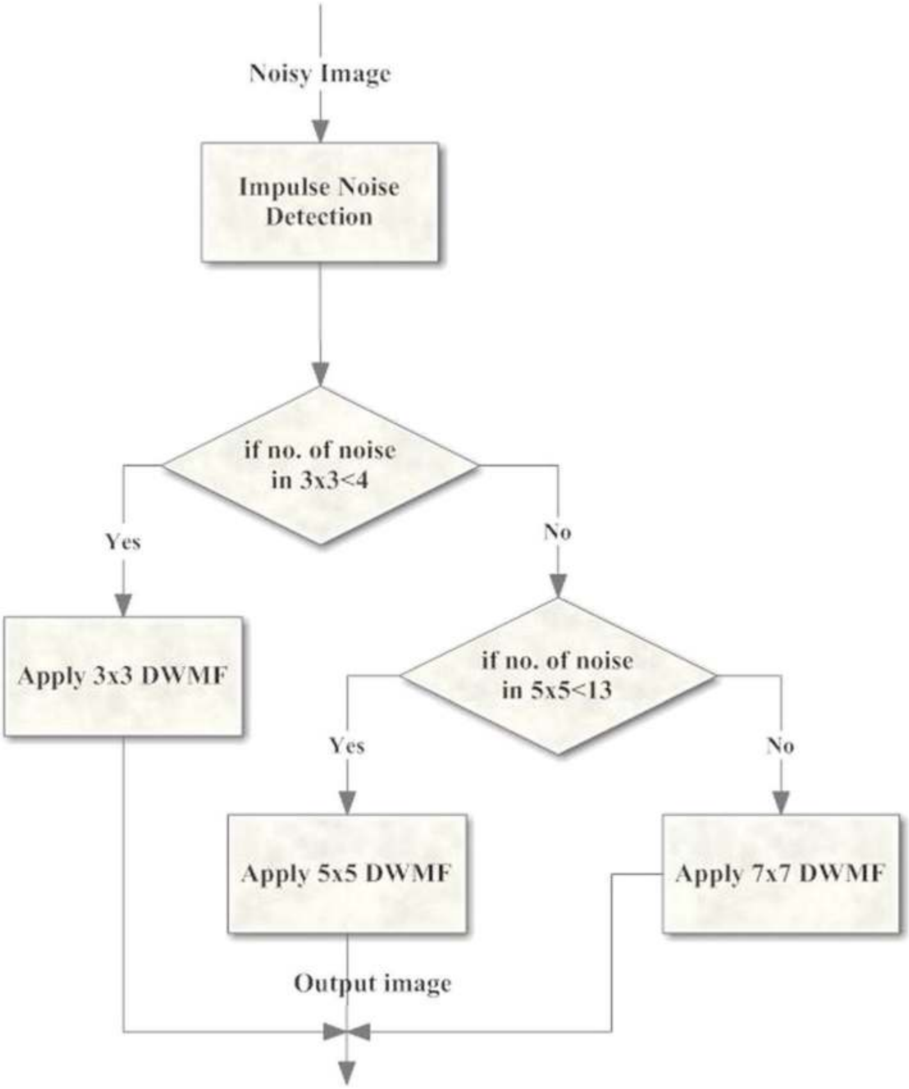


Fig. 5 Flowchart of the proposed ADWMF

Fig. 6 Test images used for performance evaluation

calculations that were helpful in proposing adaptive weighted median filter are discussed.

3.2.1 Window selection

Automatic selection of weightage window can be made using 2D Gaussian surface. Gaussian function has the property that its intensity increases as move towards center. Figure 3 shows the process of obtaining 5 × 5 weightage window using 5 × 5 Gaussian surface. Figure 3a shows the intensity values of 5 × 5 Gaussian surface. Intensity values (X) in Fig. 3b were sorted in ascending order as

$$S = \left\{s_{(1,r)} | s_{(1,r-1)} \le s(1,r), r = 1, 2, \dots ..(2N+1)^2, s_{(1,r)} \in X\right\} \tag{9}$$

Weights were assigned to those locations using

$$H = \left\{h_{(1,l)} \, | h_{(1,l+1)} = h_{(1,l)} + 1 \forall s_{(1,l+1)} > s_{(1,l)}, h_{(1,1)} = 1\right\} \tag{10}$$

All those weights assigned to sorted elements were placed in corresponding locations of weightage window according to original locations of sorted elements of 5 × 5 Gaussian surface to get weightage window shown in Fig. 3c. Weightage windows for other window sizes were also obtained using Gaussian surface in the same way.

3.2.2 Dynamically weighted median filter

Figure 4 shows the flowchart of dynamically weighted median filter (DWMF). Weighted median filter is modified with the help of impulse noise detection algorithm to propose DWMF. Binary image I_b and noisy image I are input of DWMF. Binary image I_b can be obtained using proposed impulse noise detection algorithm. W*W patches, W_n and W_b, are selected across all detected noisy pixels in both noisy image I and binary image I_b respectively. Weightage window W_{weights} of size W*W is calculated, but those locations of W_{weights} are removed where W_b have the values of 1 (if all entries of W_b are 1, W_{weights} is replaced by W_b to avoid exception). In such way, detected noisy pixels get weightage of 0. Assigned weights in W_{weights} are shifted if gaps are observed in W_{weights} due to removal of elements at noisy locations. For example, if all elements that correspond to weightage of 4 are detected as noisy pixels, weightage of 4 is removed from W_{weights}. Hence, there is a jump of 2 is observed when moving from weightage of 3 to 5 in W_{weights}. In such case, weights are reassigned to decrease the number of repetitions. The modified window is summed, and weight of one of the highest weights is incremented in case if the sum is even. The condition of odd sum is imposed in order to avoid having averaged value of two pixels as a result of DWMF. Final weightage window W_R is created after confirming the odd sum of

Table 4 PSNR comparison for 30% noise (dB)

	AMF	AWMF	SAMF	[21]	[20]	Proposed
Banana	38.54	37.83	39.07	40.12	38.32	40.19
Briefcase	31.91	31.15	32.59	33.5	31.8	33.36
Brush	33.85	33.41	21.91	26.39	33.74	35.18
Coffee	32.86	32.61	33.19	33.61	32.7	35.35
Egg	39.28	38.52	40.22	41.05	39.08	41.23
Flower	35.43	34.69	36.06	37.06	34.98	37.03
Golf cart	31.37	31.07	30.53	32.2	31.31	31.53
Grater	34.17	33.69	34.57	35.68	34.1	35.79
Kettle	32.9	32.46	34.27	35.4	32.77	35.64
Lena	34	33.31	34.78	36.41	33.89	36.12
Camera	33.7	33.15	33.95	35.93	33.66	35.57
Average	34.3	33.8	33.7	35.2	34.2	36.1

Table 5 PSNR comparison for 50% noise (dB)

	AMF	AWMF	SAMF	[21]	[20]	Proposed
Banana	34.92	34.36	36.47	36.56	34.78	36.57
Briefcase	27.94	27.42	29.69	30.14	27.81	29.91
Brush	30.33	29.99	19.2	22.82	30.28	31.75
Coffee	29.05	28.75	30.4	30.23	28.91	31.56
Egg	35.2	34.5	36.67	37.11	34.87	37.47
Flower	31.42	30.69	33.19	33.37	31.1	33.59
Golf cart	27.75	27.38	28.38	29.43	27.7	28.92
Grater	30.46	29.98	31.72	32.59	30.42	32.25
Kettle	29.16	28.69	31.05	31.72	29.05	31.76
Lena	29.92	29.23	31.71	32.78	29.76	31.39
Camera	29.91	29.46	31.02	32.46	29.76	32.55
Average	30.5	30	30.8	31.7	30.4	32.5

repeated window to create repetition array A_R. Noisy pixel is replaced by median value of A_R.

3.2.3 Statistics calculation for adaptive criteria

As it is discussed in Section 1, using the same window for an image on the basis of total noise density of image results in failure of removal of noise in case when some regions of image contain more noisy pixels compared to other. However, extending window adaptively without considering performance might not be a good option as well. Considering these problems, the proposed ADWMF uses windows of different sizes according to the number of noisy pixels present in a region. Window is extended when numbers of noisy pixels are greater than threshold set for the current window. These thresholds are defined on the basis of minimum mean square error.

Table 6 PSNR comparison for 90% noise (dB)

	AMF	AWMF	SAMF	[21]	[20]	Proposed
Banana	23.9	18.93	30.54	30.65	28.13	30.71
Briefcase	19.88	16.78	21.88	22.97	21.16	23.6
Brush	20.97	17.68	14.77	17.43	24.74	27.16
Coffee	20.02	16.98	22.56	23.41	21.7	24.23
Egg	24.06	18.98	28.81	29.63	26.75	29.32
Flower	21.18	17.52	25.03	25.85	23.05	25.85
Golf cart	20.3	17.59	23.18	23.21	21.77	23.54
Grater	21.66	17.9	26.25	26.36	24.1	26.63
Kettle	20.14	16.8	23.97	24.86	22.21	24.69
Lena	20.44	16.99	24.9	25.26	22.38	25.21
Camera	20.31	17.03	24.76	25.24	22.98	25.09
Average	21.2	17.5	24.2	24.9	23.53	26

Table 7 MSSIM comparison for 30% noise

	AMF	AWMF	SAMF	[21]	[20]	Proposed
Banana	0.966	0.962	0.968	0.968	0.966	0.975
Briefcase	0.927	0.918	0.933	0.933	0.926	0.941
Brush	0.949	0.945	0.897	0.893	0.949	0.963
Coffee	0.96	0.958	0.963	0.958	0.96	0.974
Egg	0.971	0.967	0.972	0.972	0.971	0.98
Flower	0.964	0.958	0.97	0.968	0.964	0.976
Golf cart	0.916	0.91	0.91	0.926	0.916	0.936
Grater	0.955	0.951	0.956	0.96	0.955	0.969
Kettle	0.948	0.944	0.957	0.956	0.948	0.965
Lena	0.953	0.947	0.958	0.958	0.953	0.966
Camera	0.958	0.953	0.961	0.963	0.958	0.971
Average	0.952	0.947	0.95	0.95	0.951	0.965

Binary image containing locations of detected noisy pixels was obtained using the proposed impulse noise detection. Using that binary image, numbers of detected noisy pixels in 3 × 3, 5 × 5, and 7 × 7 windows were counted to apply the corresponding DWMF filter. The formula for mean square error used is given as

$$\mathrm{MSE} = \sum_{k=1}^{n} \left(I_k - \hat{I}_k\right)^2 / n \qquad (11)$$

where

I = Image

n = Number of pixels

k = kth noisy pixel

I_k = Original kth pixel value.

$\hat{I}_k$ = Recovered kth pixel value.

Equation (11) was applied to all 11 test images. Average MSEs were calculated for median, weighted median, and DWMF with 3 × 3, 5 × 5, and 7 × 7 windows to see the possibility of achieving low MSE with combination of hybrid filter of more than one filter or switching between different window sizes, but MSE for DWMF was smaller as compared to median

Table 8 MSSIM comparison for 50% noise

	AMF	AWMF	SAMF	[21]	[20]	Proposed
Banana	0.927	0.921	0.941	0.942	0.927	0.948
Briefcase	0.844	0.832	0.876	0.876	0.844	0.885
Brush	0.893	0.888	0.846	0.847	0.893	0.922
Coffee	0.914	0.908	0.932	0.928	0.914	0.947
Egg	0.934	0.927	0.948	0.95	0.934	0.956
Flower	0.915	0.903	0.942	0.94	0.915	0.948
Golf cart	0.819	0.808	0.836	0.86	0.819	0.866
Grater	0.9	0.892	0.916	0.925	0.9	0.931
Kettle	0.891	0.881	0.917	0.916	0.891	0.926
Lena	0.896	0.886	0.919	0.922	0.896	0.928
Camera	0.907	0.899	0.926	0.931	0.907	0.937
Average	0.895	0.886	0.909	0.912	0.895	0.927

Table 9 MSSIM comparison for 90% noise

	AMF	AWMF	SAMF	[21]	[20]	Proposed
Banana	0.718	0.619	0.827	0.836	0.768	0.834
Briefcase	0.528	0.464	0.652	0.656	0.565	0.674
Brush	0.64	0.576	0.672	0.676	0.692	0.767
Coffee	0.657	0.58	0.786	0.791	0.703	0.811
Egg	0.713	0.625	0.82	0.843	0.758	0.833
Flower	0.582	0.496	0.757	0.785	0.641	0.779
Golf cart	0.455	0.405	0.566	0.599	0.49	0.591
Grater	0.627	0.543	0.75	0.77	0.674	0.768
Kettle	0.61	0.538	0.735	0.753	0.659	0.758
Lena	0.6	0.516	0.74	0.765	0.65	0.754
Camera	0.635	0.562	0.751	0.786	0.684	0.771
Average	0.615	0.539	0.732	0.751	0.662	0.758

Table 11 CPU elapsed time comparison for 50% noise

	AMF	AWMF	SAMF	[21]	[20]	Proposed
Banana	2.7	3.1	2.4	3.3	1.5	4.5
Briefcase	3.2	1.7	2.4	3.2	1.6	4.2
Brush	8	1.9	4.8	5.4	4.6	4.6
Coffee	4.4	3.1	3.2	4.3	2.1	5.1
Egg	3	3.8	2.6	3.5	1.6	4.3
Flower	3.2	5.4	2.7	3.7	1.7	4.5
Golf cart	3.4	2.7	2.8	3.9	1.8	4.6
Grater	2.5	2.8	2.2	3	1.3	3.7
Kettle	2.5	2.8	2.2	3.1	1.4	3.9
Lena	2.8	3.1	2.5	3.4	1.5	4.5
camera	2.7	3	2.4	3.3	1.5	4.2
Average	3.5	3	2.7	3.6	1.9	4.4

and weighted median filters. For example, to calculate statistics for DWMF 3 × 3, a 3 × 3 window was moved across every detected noisy pixel and numbers of detected noisy pixels were calculated in that window. All pixels which have the same number of noisy pixels in a 3 × 3 window were grouped together. Similarly, other cases which have different numbers of noisy pixels from 2 to 9 were also calculated and grouped to calculate MSEs together. Similarly for 5 × 5 and 7 × 7 windows, the cases which have different numbers of noisy pixels were in group.

Table 2 shows the average MSEs with applications of median, weighted median, and DWMF using 3 × 3, 5 × 5, and 7 × 7 windows for selected number of noisy pixels. From the table, we can find that the DWMF provides better performances for any window size than the other filters. Since the number of noisy pixel cannot exceed the total number of pixels of a window, the maximum number of noisy pixels in a 3 × 3 window becomes 8. MSE for all filters with all three window sizes were examined to see the possibility of proposing a hybrid filter, but it can be seen from the table that proposed DWMF have smaller MSE for all cases. Salt and pepper noise with 30, 50, and 90% densities were added to all 11 test images to obtain statistics for improving. Table 3 shows the selected filters which provide the best performances according to the different numbers of noisy pixels.

3.3 Proposed adaptive filter

Figure 5 shows the flowchart of the proposed algorithm. Proposed filter consists of two steps, noise detection and noise removal. Both stages are incorporated to each other. Result of impulse noise detection is a binary image with values of 0 for non-noisy region and values of 1 for noisy pixel region. Numbers of noisy pixels are counted across all detected pixels using 3 × 3 and 5 × 5 windows, and then, a dynamically weighted median filter (DWM) with a specific window size is adaptively applied according to the presence of number of noisy pixels in a region. Impulse noise detection is not only used to count

Table 10 CPU elapsed time comparison for 30% noise

	AMF	AWMF	SAMF	[21]	[20]	Proposed
Banana	1.6	2.2	1.5	2.1	1.1	2.19
Briefcase	2	2.6	1.5	2	1.2	2.29
Brush	6.6	3.1	2.5	3.2	2.9	4.61
Coffee	2.9	1.9	1.9	2.7	1.8	3.01
Egg	1.6	3	1.5	2.2	1.2	2.3
Flower	1.8	3.1	1.6	2.2	1.3	2.39
Golf cart	1.9	2.8	1.7	2.4	1.3	2.45
Grater	1.3	2.1	1.3	1.8	0.9	1.95
Kettle	1.4	1.8	1.3	1.9	0.9	2.05
Lena	1.5	2	1.5	2.1	1	2.38
Camera	1.5	1.9	1.4	1.9	1	2.24
Average	2.2	2.4	1.6	2.2	1.3	2.5

Table 12 CPU elapsed time comparison for 90% noise

	AMF	AWMF	SAMF	[21]	[20]	Proposed
Banana	5.9	6.1	4.9	6.4	2.3	8.4
Briefcase	5.7	6	4.7	6	2.2	8.5
Brush	9.2	11	6.3	8.5	3	9.8
Coffee	7.4	7.5	6	7.7	2.8	10.4
Egg	6.5	6.3	5.1	6.6	2.4	9
Flower	6.5	7.1	6.6	7.7	2.6	9.4
Golf cart	7.6	7.2	5.4	7	2.5	9.7
Grater	5.2	5.4	4.3	5.5	2	7.8
Kettle	5.4	5.5	4.5	5.9	2.1	8.5
Lena	6	6.3	5	6.4	2.3	9.5
Camera	5.7	5.9	4.7	6.1	2.2	9
Average	6.5	6.8	5.2	6.7	2.4	9.1

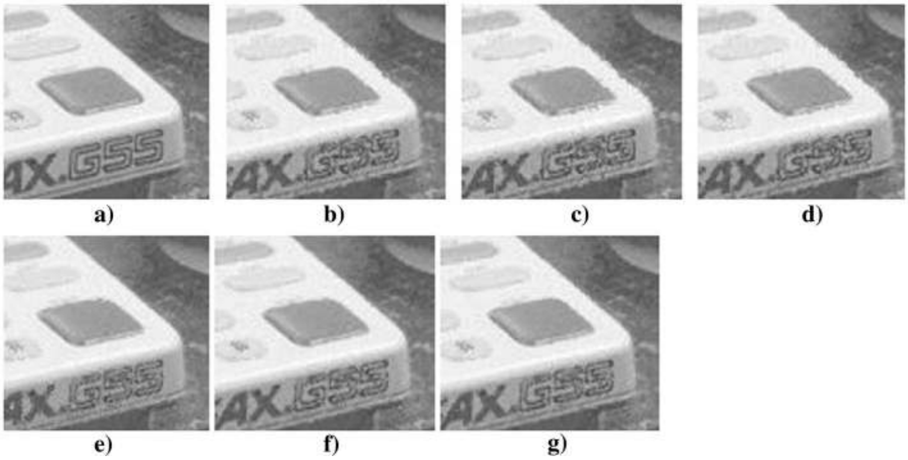


Fig. 7 Zoomed results with 30% impulsive noise. **a** Original image. **b** AMF. **c** AWMF. **d** Bhateja [20]. **e** SAMF. **f** Bhadouria [21]. **g** Proposed ADWMF

the number of noisy pixels but also used in assigning weightage of 0 to detected noisy locations in 3 × 3, 5 × 5, and 7 × 7 windows.

Thresholds given in Table 3 are used to propose ADWMF. Steps for ADWMF are given as:

1) Get binary image I_b by applying proposed noise detection on noisy image I.
2) Calculate the number of detected noisy pixels per $N \times N$ window by centering every detected noisy pixel using 3 × 3, 5 × 5 and 7 × 7 windows on I_b.
3) Create binary image I^3_{bDWM} showing locations where numbers of detected noisy pixels are 1, 2, or 3 in 3 × 3 window calculated using step 2. In those locations where I^3_{bDWM} is 1, apply 3 × 3 DWMF. Subtract I^3_{bDWM} from I_b to know about remaining detected locations.
4) Repeat step 3 by creating I^5_{bDWM} and I^7_{bDWM} using Table 3 and apply 5 × 5 DWMF and 7 × 7 DWMF respectively. Subtract I^5_{bDWM} from I_b after applying 5 × 5 DWMF so that 7 × 7 DWMF does not reprocess pixels processed by 5 × 5 DWMF.

4 Results and discussion

Figure 6 shows the set of images used for comparisons of algorithms. Performance evaluation was made on the basis of comparison of peak signal-to-noise ratio (PSNR) and mean structural similarity (MSSIM) of adaptive DWMF with methods proposed by adaptive median filter (AMF), adaptive weighted median filter (AWMF), Ibrahim et al. [19] (AdFuz), Bhadouria [21], and Bhateja et al. [20].

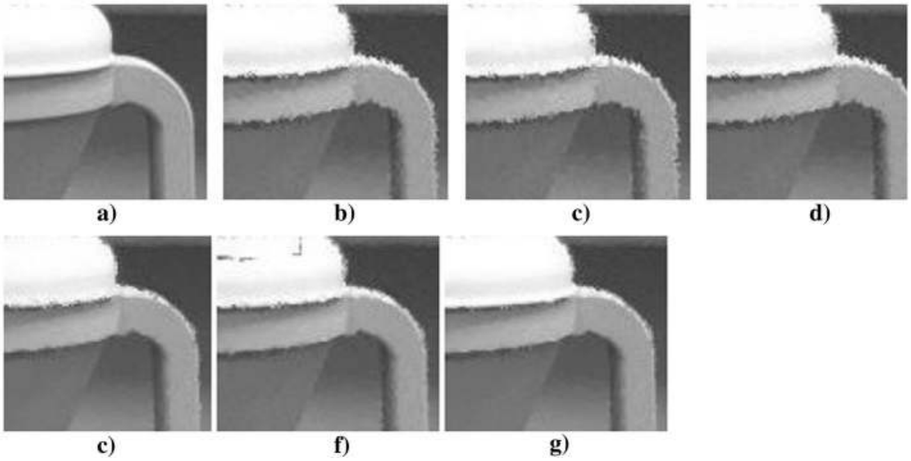


Fig. 8 Zoomed results with 50% impulsive noise. **a** Original image. **b** AMF. **c** AWMF. **d** Bhateja [20]. **e** SAMF. **f** Bhadouria [21]. **g** Proposed ADWMF

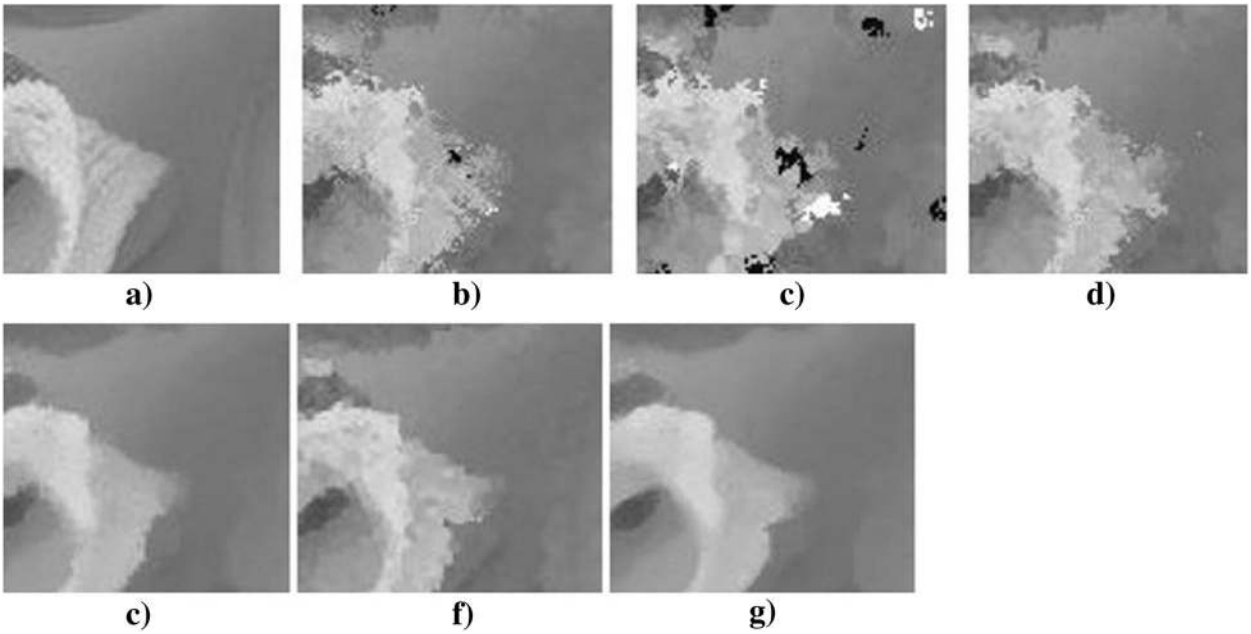


Fig. 9 Zoomed results with 90% impulsive noise. **a** Original image. **b** AMF. **c** AWMF. **d** Bhateja [20]. **e** SAMF. **f** Bhadouria [21]. **g** Proposed ADWMF

PSNR calculation was done using (12) as given below

$$\mathrm{PSNR}_I = 10\log_{10}\left(\mathrm{Max}_I^2/\mathrm{MSE}_I\right) \tag{12}$$

Tables 4, 5, and 6 show the PSNR comparisons of proposed ADWMF with conventional methods for 30, 50, and 90% noise densities respectively. Adaptive weighted median filter (AWMF) is an adaptive form of simple weighted median filter. PSNR comparisons show that the proposed method outperforms all other methods for both low- and high-density noise. Tables 7, 8, and 9 show the MSSIM comparisons of all methods. The proposed method outperformed all other methods in both PSNR and MSSIM comparisons.

Tables 10, 11, and 12 show the CPU elapsed time comparisons to show the complexity of all methods. For 30 and 50% noise densities, the proposed method provides slightly higher complexity compared to other methods; however, for 90% noise, the proposed method needs more time since 7 × 7 window is used too often.

Figures 7, 8, and 9 show the zoomed results of all methods. SAMF among conventional methods performs well for low-density noise; however, for higher densities, it provides artifacts along edges. For 90% noise density, AMF and AWMF fail to remove noise, whereas SAMF, Bhadouria [20], and Bhateja [21] restore all noisy pixels, however, at the cost of blurring along edges. The proposed method not only preserves edges for low density of noise but also performs very well for noise density as high as 90%. Conventional method totally fails to preserve regions that have pure white or pure black background as in Fig. 8, because pure white or pure black regions provide confusion as they have same intensity as of the impulse noise; however, the proposed method, irrespective of intensity values, provides edge and detail preservation in all cases.

5 Conclusions

A new impulse noise removal filter, dynamically weighted median filter (DWMF) along with modified impulse noise detection method, is proposed in this paper. Conventional modified versions of switching median filters extend window size either according to the total noise density of image that results in failure to remove noise from regions that have excessive noise in a region due to uneven distribution of noise in different regions. Conventional versions of adaptive median filters extend its window size either when none of noise-free pixel is available or when median filter of a window is a noisy pixel. Such approaches does not consider the performance analysis. Proposed method overcomes problems mentioned above by considering both different amounts of noisy pixels in each region as well as ensuring the best performance. The proposed method first detects locations of noisy pixels in image followed by noise removal stage. Proposed impulse noise detection algorithm is a modified version of impulse noise detection algorithm proposed by Ghanekar [30]. Weighted median filter is modified by assigning weights dynamically by combining it with proposed impulse noise detection technique to propose DWMF. Impulse noise detection algorithm is not only used for assigning weights dynamically but also used with DWMF to propose a new adaptive filter named adaptive dynamically weighted median filter (ADWMF). Experiments were performed to evaluate the thresholds at which DWMF should switch to a larger window to provide better performance. Statistics of those experiments were used to propose ADWMF. Switching of window for ADWMF is made on the basis of number of detected noisy pixels in a specific region instead of applying DWMF of a specific size for whole image on the basis of total noise density present in it. ADWMF can be applied to any kind of image whether effected by low or high density of noise. It is not necessary for noise to be evenly distributed in image; ADWMF can restore the image even if some portions have access of noise density. Both objective and subjective simulations were performed which show that ADWMF outperforms conventional methods.

In future, after researching about the possibility where the proposed noise detection and filtering approach can be applied to some practical industrial processes [36, 37], it might be investigated.

Acknowledgements
Not applicable (no contributors for acknowledgements)

Funding
Not applicable (no sources of funding for the research).

Availability of data and materials
Not applicable.

Authors' contributions
SK carried out most of implementations and simulations for this manuscript. DHL provided core concepts and drafted the manuscript. All authors read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details
[1]Department of Computer Science, Sukkur IBA University, Sukkur, Pakistan.
[2]Department of Electronic and Communication Engineering, Hanyang University ERICA Campus, Ansan, South Korea.

Received: 12 April 2017 Accepted: 5 September 2017
Published online: 18 September 2017

References

1. F. Russo, Edge detection in noisy images using fuzzy reasoning. IEEE Trans. Instr. and Meas. **47**(5), 1102-1105 (1998)
2. Y. Wei, J. Qiu, H.R. Karimi, Quantized H_∞filtering for continuous-time Markovian jump systems with deficient mode information. Asian J. Control **17**(5), 1914–1923 (2015)
3. Y. Wei, J. Qiu, P. Shi, H.K. Lam, A new design of H-infinity piecewise filtering for discrete-time nonlinear time-varying delay systems via T-S fuzzy affine models. IEEE Trans. Syst. Man Cybern. Syst. **47**(8), 2034–2047 (2017)
4. Wang, T., Qiu J., Fu, S., Ji, W.: Distributed fuzzy H_∞ filtering for nonlinear multirate networked double-layer industrial processes. IEEE Trans. Industrial Electronics **64**(6), 5203-5211 (2017)
5. E. Abreu, M. Lightstone, S.K. Mitra, K. Arakawa, A new efficient approach for the removal of impulse noise from highly corrupted image. IEEE Trans. Image Process. **5**(6), 1012–1025 (1996)
6. H.L. Eng, K.K. Ma, Noise adaptive soft-switching median filter. IEEE Trans. Image Process. **10**(2), 242–251 (2001)
7. P. Civicioglu, Using uncorrupted neighborhoods of the pixels for impulsive noise suppression with ANFIS. IEEE Trans. Image Process. **16**(3), 759–773 (2007)
8. Q. Zhang, R.K. Ward, J. Du, Impulse noise correction in TV transmission. IEEE Trans. Consum. Electron. **41**(3), 731–737 (1995)
9. B.Y.M. Kwan, H.K. Kwan, Impulse noise reduction in brain magnetic resonance imaging using fuzzy filters. International Journal of Medical, Health, Biomedical, Bioengineering and Pharmaceutical Engineering **5**(12), 688–691 (2011)
10. P.S. Windyga, Fast impulse noise removal. IEEE Trans. Image Process. **10**(1), 173–179 (2001)
11. J.W. Tukey, Exploratory data analysis, reading, mass. Preliminary edition (Addison-Wesley, Boston, MA, USA, 1971)
12. T.S. Huang, G.J. Yang, G.Y. Tang, A fast two-dimensional median filtering algorithm. IEEE Trans. Acoust. Speech Signal Process. **27**(1), 13–18 (1979)
13. L. Yin, R. Yang, Weighted median filters: a tutorial. IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. **43**(3), 157–192 (1996)
14. S.J. Ko, Y.H. Lee, Center weighted median filters and their applications to image enhancement. IEEE Trans. Circuits Syst. **38**(9), 984–993 (1991)
15. H. Hwang, R.A. Hadded, Adaptive median filters: new algorithms and results. IEEE Trans. Image Process. **4**(4), 499–502 (1995)
16. C.C. Chang, J.Y. Hsiao, C.P. Hsieh, An adaptive median filter for image denoising. International Symposium on Intelligent Information Technology Application **2**, 346–350 (2008)
17. M. Juneja, R. Mohana, An improved adaptive median filtering method for impulse noise detection. International Journal of Recent Trends in Engineering **1**(1), 274–278 (2009)
18. H. Ibrahim, N.S.P. Kong, T.F. Ng, Simple adaptive median filter for the removal of impulse noise from highly corrupted images. IEEE Trans. Consum. Electron. **54**(4), 1920–1927 (2008)
19. K.K.V. Toh, N.Q.M. Isa, Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction. IEEE Signal Processing Letters **17**(3), 281–284 (2010)
20. V. Bhateja, K. Rastogi, A. Verma, C. Malhotra, Non-iterative adaptive median filter for image denoising. IEEE International Conference on Signal Processing and Integrated Networks, 113–118 (2014)
21. V.S. Bhadouria, D. Ghoshal, A.H. Siddiqui, A new approach for high density saturated impulse noise removal using decision-based coupled window median filter. SIViP **8**(1), 71–84 (2014)
22. H.M.R. Afzal, J. Yu, Y. Kang, Impulse noise removal using fuzzy logics. IEEE Annual Academnic Conference of Chinese Association of Automation **32**, 413–418 (2017)
23. V.R. Vijaykumar, D. Ebenezer, P.T. Vanathi, Detail preserving median based filter for impulse noise removal in digital images. Int. Conf. Signal Process., 793–796 (2008)
24. M. Jampur, M. Ziari, R.E. Zadeh, M. Ashourzadeh, Impulse noise detection and reduction using fuzzy logic and median heuristic filter. Int. Conf. Netw. Inf. Technol., 19–23 (2010)
25. W. Luo, Efficient removal of impulse noise from digital images. IEEE Trans. Consum. Electron. **52**(2), 523–527 (2006)
26. F. Duan, Y.J. Zhang, A highly effective impulse noise detection algorithm for switching median filters. IEEE Signal Processing Letters **17**(7), 647–650 (2010)
27. S. Zhang, M.A. Karim, A new impulse detector for switching median filters. IEEE Signal Process Lett. **9**(11), 360–363 (2002)
28. P.E. Ng, K.K. Ma, A switching median filter with boundary discriminative noise detection for extremely corrupt images. IEEE Trans. Image Process. **15**(6), 1506–1516 (2006)
29. D.H. Lee, An edge-based adaptive method for removing high-density impulsive noise from an image while preserving edges. ETRI J. **34**(4), 564–571 (2012)
30. U. Ghanekar, A novel impulse detector for filtering of highly corrupted images. World Acad. Sci. Eng. Technol. **14**, 353–355 (2008)
31. C. Deng, J.Y. An, An impulse noise removal based on a wavelet neural network. Inf. Comput. Sci. **2**, 71–74 (2009)

32. A.R. Sadri, M. Zekrii, S. Sadrii, N. Gheissari, Impulse noise cancellation of medical images using wavelet networks and median filters. Journal of Medical Signals and Sensors **2**(1), 25–37 (2012)
33. S. Wang, Q. Liu, Y. Xia, P. Dong, J. Luo, Q. Huang, D.D. Feng, Dictionary learning based impulse noise removal via l1-l1 minimization. Signal Process. **93**(9), 2696-2708 (2013)
34. J. Liu, X.C. Tai, H. Huang, Z. Huan, A weighted dictionary learning model for denoising images corrupted by mixed noise. IEEE Trans. Image Process. **22**(3), 2696–2708 (2013)
35. D. Duan, Q. Mo, Y. Wan, Z. Han, A detail preserving filter for impulse noise removal. International Conference on Computer Application and System Modeling **2**(11), 265–268 (2010)
36. T. Wang, H. Gao, J. Qiu, A combined fault-tolerant and predictive control for network-based industrial processes. IEEE Trans. Ind. Electron. **63**(4), 2529–2536 (2016)
37. Y. Wei, J. Qiu, H.K. Lam, L. Wu, Approaches to T-S Fuzzy-affine-model-based reliable output feedback control for nonlinear Ito stochastic systems. IEEE Trans. Fuzzy Syst. **25**(3), 569–583 (2017)