

RESEARCH

Open Access



CrossMark

Triple Threshold Statistical Detection filter for removing high density random-valued impulse noise in images

Neeti Singh* and Umamaheswari Oorkavalan

Abstract

This study presents a novel noise detection algorithm which satisfactorily detects noisy pixels in images corrupted by random-valued impulse noise of high levels up to 80% noise density. Three levels of adaptive thresholds along with an auxiliary condition are used in this method which adequately addresses the drawbacks of existing methods, especially the miss detection of noise-free pixels as noisy pixels and vice versa. A noise signature is calculated for every pixel and compared with the first threshold to identify noise followed by the comparison of the central pixel with the second and third levels of thresholds. In addition to the standard deviation and mean, the concept of quartile has been used as another measure of dispersion. After detection, a fuzzy switching weighted median filter is applied to restore the corrupted image. The simulation results demonstrate that the proposed method is able to outperform the existing methods in both the detection and filtering of random-valued impulse noise in images.

Keywords: Triple Threshold Statistical Detection, Random-valued impulse noise, Quartile, High density noise, Fuzzy switching weighted median filter and noise signature

1 Introduction

This research is aimed at designing and proposing a new algorithm for detection and filtering of high density random-valued impulse noise (RVIN) in images. Removing salt and pepper impulse noise (SPN) is an easy task, but removing RVIN of high densities is a challenging task because of the nature of RVIN. Impulse noise is primarily caused by malfunctioning pixels in camera sensors, faulty memory locations, and transmission in noisy channel [1].

In RVIN, noisy pixels are randomly located between 0 and 255, and hence, it is very difficult to detect the noise and restore the image [2]. Since the difference between a noisy pixel and a noise-free pixel in RVIN may not be huge, the image appears hazy and blurred. Impulsive noise removal consists of, first, the detection of the noisy pixels by taking into account the edges and then substitution of the noisy pixels with the best approximation of a value based on the neighborhood. Removal of RVIN at higher noise densities is still more tedious because the

noise-free pixels available for restoration of image are too few at high noise densities. In addition to the removal of impulsive noise, preservation of the image details is also a desired result of the proposed work.

Some methods only employ filtering without the detection of impulse noise whereas certain other methods first employ the detection of the impulse noise followed by the filtering. Over the past two decades, performance of the nonlinear filters particularly the two-stage median-based filters has improved multifold for the images corrupted by RVIN [1]. The existing state-of-the-art methods have attempted to remove the RVIN in images but are successful only at lower noise densities. At higher noise densities of RVIN, the image restoration by these methods is not satisfactory. Hence, various applications which are based on image denoising like medical imaging, face recognition, satellite imaging, and fisheries are rendered ineffective. The main focus areas of this research are quality image restoration at higher noise densities of RVIN and achievement of lower computational cost of the proposed detection and filtering technique.

* Correspondence: neetisingh83@rediffmail.com

Department of Electronics and Communication Engineering, College of Engineering, Guindy, Anna University, Chennai, Tamil Nadu, India

1.1 Related work

Various algorithms have been proposed to remove RVIN from corrupted images with different architects [3–28]. However, these algorithms have their own assumptions, advantages, and limitations. While some algorithms modify all the pixels (whether noisy or not), other advanced algorithms first detect the noisy pixels and then replace them with an estimated value. Some of these algorithms which have been considered for study and comparison with the proposed algorithm are noise adaptive fuzzy switching weighted median (NAFSWM) filter [14], rank order absolute difference (ROAD) filter [25], rank order logarithmic difference (ROLD) filter [23], triangle-based linear interpolation detection (TBLI) filter [19], adaptive switching median (ASM) filter [10], adaptive fuzzy inference system-based directional median (AFIDM) filter [3], and condition-based detection (CBD) filter [9].

The NAFSWM filter [14] is a combination of the simple adaptive weighted median filter [20] and the fuzzy switching median filter. In ROAD, the detection stage utilizes the absolute difference between the central pixel and other pixels and then a predefined threshold is employed to detect the pixel as noisy or noise-free [25]. ROLD is similar to ROAD except that the ROLD exploits the logarithmic function of the absolute difference which results in a better detection as compared to ROAD [23]. TBLI method uses triangle-based linear interpolation to detect noisy pixels and Differential Evolution algorithm for restoring the image [19]. In ASM, detection is based on the absolute deviation between the mean value and the central pixel and its comparison with threshold value [10]. The CBD filter uses two conditions to differentiate between corrupted and uncorrupted pixels; first one is based on the noise intensity level and second is based on two predetermined threshold values [9]. The AFIDM filter consists of two major processes namely fuzzy inference system-based noise detection and noise filtering [3].

When these methods are applied to the images corrupted by high density RVIN, some of the noise-free pixels are detected as noisy pixels and vice versa. Hence, the filtering stage of these methods not only fails to correct all the noisy pixels but also alters some noise-free pixels which were detected as noisy pixels. This results in lower values of peak signal to noise ratio (*PSNR*) and structural similarity index (*SSIM*) corroborated by poor picture restoration. This is further elaborated in the Section 3.

1.2 Our contribution

In order to overcome the miss detection, in this paper, a new algorithm has been designed and proposed in the detection stage. This new detection algorithm is designated

as Triple Threshold Statistical Detection (TTSD). Noise signature (*NS*) used in this method gives it a distinct advantage over other methods in detection of noisy pixels. The proposed filter is compared with other filters in terms of *PSNR* and *SSIM*, and the comparison results show that the TTSD filter performs better than other filtering techniques. The detection stage of TTSD involves three main conditions (three levels of thresholds) and one auxiliary condition [9]. These conditions must be verified to accurately detect noisy pixels. Statistical tools such as standard deviation, mean, and quartile help in detecting outliers. The filtering stage of TTSD utilizes fuzzy switching weighted median (FSWM).

The main contributions as compared to the existing methods can be summarized as follows:

- Devising a completely new method of image processing wherein simple statistical parameters have been utilized for detection of noisy pixels
- Use of three levels of thresholds to detect noisy pixels with higher accuracy than the existing methods
- Achieving quality picture restoration at higher noise densities including up to 80%
- Use of adaptive thresholds which depend upon the values of noise density of corruption and neighboring pixels
- Devising a new parameter called *NS* which can be used to have comparisons among different methods of image processing

2 Proposed method

2.1 Noise model

The RVIN model of equal probability is used in this filter. The probability density function, $f(C_{i,j})$, can be expressed as

$$f(C_{i,j}) = \begin{cases} \frac{N_D}{2} & 0 \leq C_{i,j} < m, \\ 1 - N_D & C_{i,j} = S_{i,j}, \\ \frac{N_D}{2} & (255-m) < C_{i,j} < 255 \end{cases} \quad (1)$$

where $C_{i,j}$ is the (i,j) th pixel in the corrupted image, $S_{i,j}$ is the (i,j) th pixel in the original image, N_D is the noise density, and m is the noise intensity level. The dynamic range of the image intensity values is $[0, R-1]$, where $R = 2^n$ and n is the number of bits per pixel. An 8-bit gray image is assumed and hence, $n = 8$ and $R = 256$.

As mentioned earlier, statistical parameters such as measures of dispersion have been utilized in setting of the three thresholds in TTSD. The same is discussed below.

2.2 Basic concept

The basic aim of any detection process is to separate the noisy pixels from the noise-free pixels so that they can

be replaced with a suitable pixel value in the filtering stage. Various existing methods for image processing have utilized some or the other measures of dispersion to identify the outliers. Awad [2] has demonstrated that the detection process can be based on finding the optimum direction by calculating the standard deviation of different directions in the filtering window. The measures of dispersion used in this proposed algorithm for setting different thresholds are the mean, the standard deviation, and the quartile. The reason for choosing three levels of thresholds is explained below.

The standard deviation is a measure which represents the extent by which each value within a set of data varies from the mean value. In effect, it shows how closely all the values in the sample are bunched around the mean. It is the best and most widely used measure of dispersion since it takes into account every variable in the dataset. When the values are very closely bunched together, the standard deviation is smaller. When the values are widespread, the standard deviation will be relatively large. The standard deviation is usually applied along with the mean, and its unit is same as that of mean.

The standard deviation of a population is denoted as σ (sigma) and is calculated as follows:

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \mu)^2}{N}} \quad (2)$$

where x_i represents each value in the population, μ is the mean of the population, and N is the number of values in the population. Mean μ is calculated as follows:

$$\mu = \frac{\sum_{i=1}^N x_i}{N} \quad (3)$$

Many natural datasets follow a normal distribution. In a normal distribution, most of the values of the dataset are clustered around the mean while only few values are very high or very low. For normally distributed dataset, 68% of the values lie within one standard deviation (1σ) from the mean value, 95% of the values lie within two standard deviations (2σ) from the mean, and 99% of the values lie within three standard deviations (3σ) from the mean. Figure 1 clearly depicts the standard deviation as a measure of dispersion.

The quartiles represent the milestones in a sorted dataset which along with the median divide the dataset into four quarters. The first quartile is the median of the lower half of the data set and is denoted by Q_1 . This also means that the 25% of the values in the dataset lie below Q_1 and the rest 75% lie above Q_1 . The third quartile is the median of the upper half of the dataset and is denoted by Q_3 . This would also mean that the 75% of

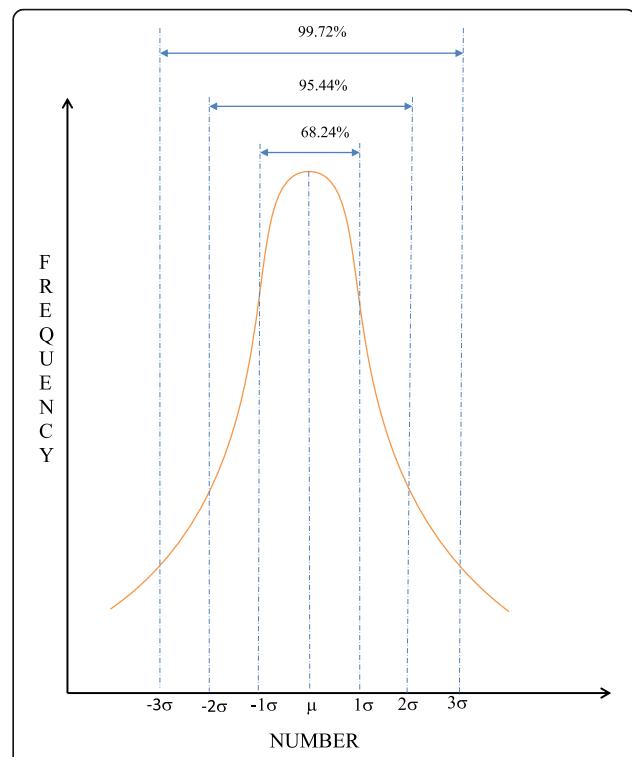


Fig. 1 Standard deviation as a measure of dispersion. Legends: σ , standard deviation; μ , mean

the values in the dataset lie below Q_3 and the rest 25% lie above Q_3 . The first and third quartiles can be calculated as

$$Q_1 = \left(\frac{N+1}{4} \right)^{\text{th}} \text{ value} \quad (4)$$

$$Q_3 = \left(\frac{3(N+1)}{4} \right)^{\text{th}} \text{ value} \quad (5)$$

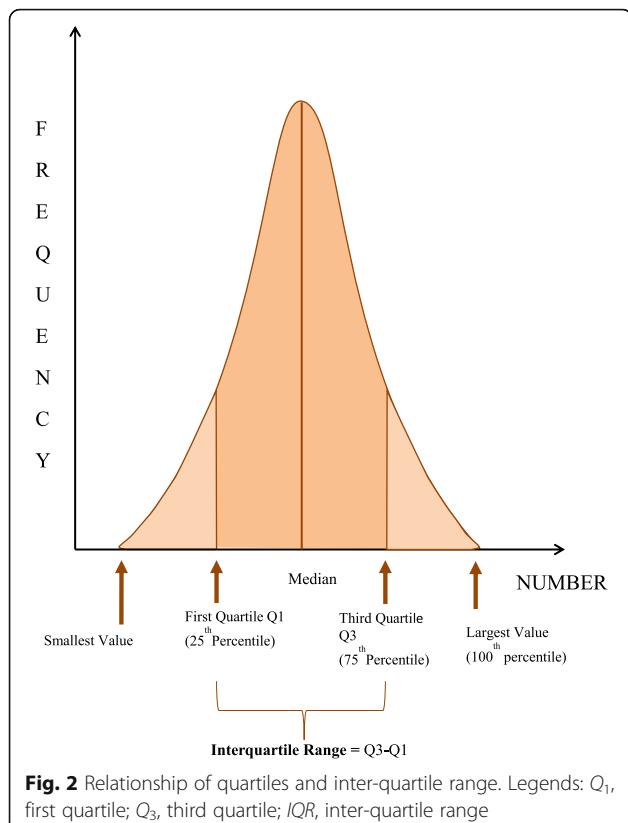
where N is the number of values in the sorted dataset.

The inter-quartile range (IQR) in a dataset indicates the extent to which the central 50% of its values are dispersed. IQR is calculated by subtracting the lower quartile from the upper quartile as follows

$$IQR = Q_3 - Q_1 \quad (6)$$

Figure 2 shows the relationship of the quartiles and IQR . The IQR provides a robust basis to identify the outlying values. Moreover, like the range, it is also a measure of dispersion based on only two values from the dataset and hence it can only be used in addition to the other conditions to detect outliers.

The three levels of thresholds are presented and discussed as follows.



2.3 Three threshold levels

The first threshold used in this method is based on the concept of incongruity of the noisy pixel. Owing to its incongruity or abnormality, the noisy pixel would leave a signature by which it can be detected. First of all, absolute differences of all pixels with their mean value except the central pixel (*CP*) are calculated. Subsequently, mean and standard deviation of all these absolute differences are calculated. The sum of this mean and standard deviation represents the first threshold. It is being used as a threshold because the absolute differences of the pixels with their mean value would show a less deviant behavior as compared to the absolute differences of the pixels with the *CP*, provided that *CP* is a noisy pixel. Considering mean and standard deviation of such absolute differences and adding them further eases up the deviant behavior of the threshold and makes it comparable with the noise-free pixel values. Now, if *NS* is equal to or more than the first threshold, the *CP* is termed noisy; otherwise, we move to the next level of thresholds, i.e., the second level of thresholds.

After the first threshold is applied, a majority of noisy pixels get segregated from the rest of the pixels but it is still unknown whether the rest of the pixels are noise-free. For ascertaining that, the second level

of thresholds is employed. The second level of thresholds is defined as follows:

$$T_{2 \min} = \mu - K \times \sigma \quad (7)$$

$$T_{2 \max} = \mu + K \times \sigma \quad (8)$$

where, μ and σ are the mean and standard deviation of all pixels of the detection window and K is the multiplication factor of σ in the above equations. Optimum value of K is derived empirically for various random sets of pixels through simulations. A discussion on the test of optimality of K has been presented in Section 3.1. From that discussion, we find that for the given experimental setup, optimum value of K is 0.5. If a pixel falls beyond these thresholds, it is considered as noisy.

The reason behind taking mean and standard deviation for setting these thresholds is that most of the noise-free pixels would fall within the above thresholds since they smoothen out the pixel values and generally tend towards a value resembling a noise-free pixel. The noisy pixels (including the higher values and the lower values) would normally tend to fall beyond these thresholds because of their incongruity with the remaining pixels of the matrix. Figure 3 depicts the second level of thresholds for $K = 0.5$ and $K = 0.7$ and the corresponding bands available for identifying the noisy pixels in the image.

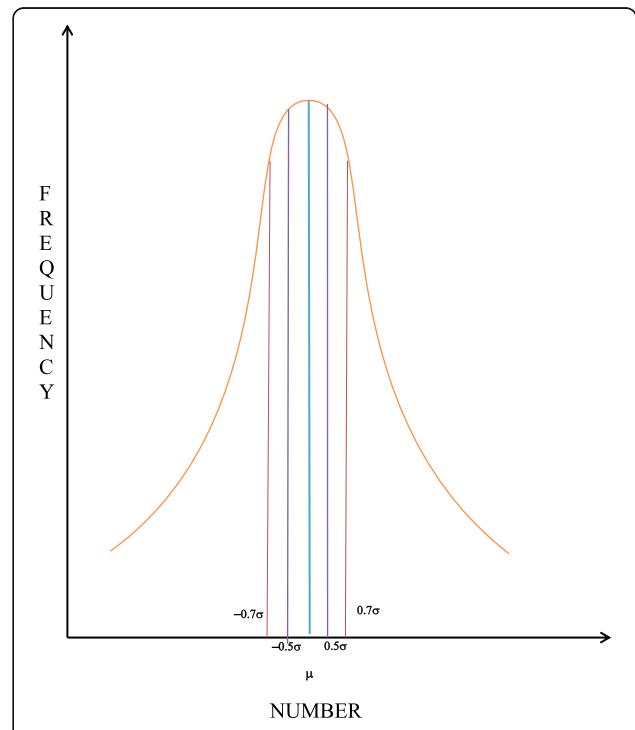


Fig. 3 Depiction of second level of thresholds for TTSD algorithm. Legends: σ , standard deviation; μ , mean

After the second level of thresholds is applied, few more noisy pixels are segregated from the rest of pixels but still it is not known whether the remaining pixels are all noise-free. For ascertaining that, the third level of thresholds is employed. The first quartile Q_1 is taken as the lower threshold, i.e., $T_{3\min}$, and the third quartile Q_3 is taken as the higher threshold, i.e., $T_{3\max}$ [29, 30]. When a CP is beyond these thresholds, the pixel is termed as noisy otherwise noise-free.

2.4 Noise signature

The NS of a pixel denotes the possibility and tendency of the pixel to be noisy. It is basically a measure of noisiness of a pixel. For a pixel, if the NS is more than a certain threshold, the pixel may be considered as noisy. Although it is not a necessary condition for a pixel to be noisy, yet it is a sufficient condition. The NS -based noise detection proposed in this work segregates a large number of noisy pixels by comparing the NS of each pixel in the image with the first threshold itself.

To calculate the NS of a pixel in a given window, absolute differences of CP with all the remaining pixels are calculated and mean of all these absolute differences is considered. This mean represents the NS . Since all the absolute differences of the CP with the remaining pixels are considered in the calculation of the NS , the probability of correct detection of noisiness in this method increases. If the CP is noisy, the value of the NS would be expected to be more than the first threshold, but if the CP is noise-free, the value of NS would be expected to be less than the first threshold. However, the occurrence of the latter does not guarantee the CP to be noise-free. In that case, the CP is to be further tested against two more thresholds as explained later.

2.5 Proposed algorithm

The TTSD method uses a 5×5 sliding window to estimate whether the CP is a noisy pixel or noise-free. As the name implies, TTSD uses three levels of thresholds as explained above. In addition to these thresholds, an auxiliary condition based on the noise intensity level (m) is also used in the detection stage of the proposed algorithm. Firstly, the NS is compared with the first threshold to segregate noisy pixels from the rest. This results in separation of a majority of noisy pixels from noise-free pixels. However, as the noise density increases, the accuracy of detection by use of first threshold decreases. Therefore, in subsequent levels of check in the proposed algorithm, the CP is compared with the second and third levels of thresholds to further segregate noisy pixels. This enhances the accuracy of detection for the algorithm as a whole. After the detection stage, the FSWM filter is employed in the filtering stage, which replaces only the

noisy pixels with an estimated value close to the original value while leaving noise-free pixels unaltered [14, 15]. Hence, the combination of TTSD in detection stage with FSWM filter in the filtering stage provides better picture restoration.

The algorithm of detection stage is summarized in next subsection.

2.5.1 Noise detection

The detection stage is used to find out whether a pixel is noisy or noise-free. In this paper, the detection stage employs three main conditions and one auxiliary condition. The three levels of thresholds used in TTSD are based on three conditions explained above. In combination with the above thresholds, another auxiliary condition should also be satisfied along with both conditions for a pixel to be noisy, i.e., $(0 \leq CP \leq m)$ or $(255 - m \leq CP \leq 255)$. This auxiliary condition is simultaneously applied with the above three conditions [9]. Only the pixels that are found to be noisy are processed in the filtering stage. The algorithm for detecting RVIN is as follows:

Step I: Take a 5×5 window A . Then, calculate mean (μ_A) and standard deviation (σ_A) of all pixels of matrix A except the CP .

Step II: Calculate p_{ij} as absolute differences of μ_A with all pixels of matrix A except CP and obtain 24 such values.

Step III: Calculate μ_p and σ_p of all the above values (i.e., p_{ij}) and define first threshold T_1 as

$$T_1 = \mu_p + \sigma_p \quad (9)$$

Step IV: Now, calculate q_{ij} as absolute difference of CP with rest of all pixels of A and obtain 24 values.

Step V: Calculate μ_q of all the above values (i.e., q_{ij}) and define the NS as

$$NS = \mu_q \quad (10)$$

Step VI: Now, check if $NS \geq T_1$

and $(0 \leq CP \leq m)$ or $(255 - m \leq CP \leq 255)$, then the CP is noisy.

Step VII: But, if $NS < T_1$, define second level of thresholds

$$T_{2\min} = \mu_A - 0.5 \times \sigma_A \quad (11)$$

$$T_{2\max} = \mu_A + 0.5 \times \sigma_A \quad (12)$$

Step VIII: Now check if ($CP \leq T_{2\min}$ or $CP \geq T_{2\max}$) and ($0 \leq CP \leq m$) or ($255 - m \leq CP \leq 255$), then the CP is noisy.

Step IX: If both conditions are not satisfied, define third thresholds

$$T_{3\min} = Q_1 \quad (13)$$

$$T_{3\max} = Q_3 \quad (14)$$

where Q_1 and Q_3 are the first and third quartiles of the set of all pixels of A except CP .

Step X: Now, check if ($CP \leq T_{3\min}$ or $CP \geq T_{3\max}$) and ($0 \leq CP \leq m$) or ($255 - m \leq CP \leq 255$), then the CP is noisy.

Otherwise, the CP is noise-free.

The flowchart of detection process is depicted in Fig. 4.

2.5.2 Filtering

Now, the filtering stage will consider only the noisy pixels leaving the noise-free pixels untouched [17]. The filtering stage employs the FSWM filter [14, 15]. The FSWM filter provides better results as compared to other filters because the detected noisy pixels are replaced by the fuzzy switching weighted median value of the noise-free pixels in its neighborhood. The weights assigned to different pixels contribute in achieving an appropriate value of median to replace the noisy pixels. The restoration term is defined as a linear combination of original pixel value and median value. The steps involved are as follows:

Step I: First, extract local information by calculating the absolute luminance difference $l_{x,y}$ in a 3×3 window W . $l_{x,y}$ is calculated as follows:

$$l_{x+k,y+l} = |W_{x+k,y+l} - W_{x,y}| \quad (15)$$

where $(x+k, y+l) \neq (x, y)$ and $-N \leq k, l \leq N$

Step II: Then, calculate the maximum absolute luminance difference in the filtering window as follows:

$$L_{x,y} = \max\{l_{x+k,y+l}\} \quad (16)$$

where $L_{x,y}$ denotes the local information of the sliding window considered.

Step III: Then apply Fuzzy Reasoning to the defined local information $L_{x,y}$. The fuzzy membership function $\mathbf{f}f_{x,y}$ is defined as follows:

$$\mathbf{f}f_{x,y} = \begin{cases} 0 & L_{x,y} < T1, \\ \frac{(L_{x,y}-T1)}{(T2-T1)} & T1 \leq L_{x,y} < T2 \\ 1 & L_{x,y} \geq T2, \end{cases} \quad (17)$$

where $T1$ and $T2$ are constants taken as 10 and 30 respectively in this paper.

Step IV: The next step searches for noise-free pixels and if not even a minimum of one noise-free pixel is identified in the current filtering window W , then the filtering window is expanded by one pixel at each of its four sides. This procedure is repeated until minimum of one noise-free pixel is identified.

Step V: Then, find the median (MED) by using the noise-free pixels. For filtering the image, a weighted median filter of 3×3 window is employed. The weight of a pixel is decided on the basis of the gradient of the surrounding pixels.

$$w_{s,t} = \begin{cases} 3 & \text{if } l < 5, \\ 2 & \text{if } 5 < l < 10, \\ 1 & \text{if } l > 10 \end{cases} \quad (18)$$

and

$$MED = \text{median}\{w_{s,t} \times W_{x+s,y+t}\} \quad (19)$$

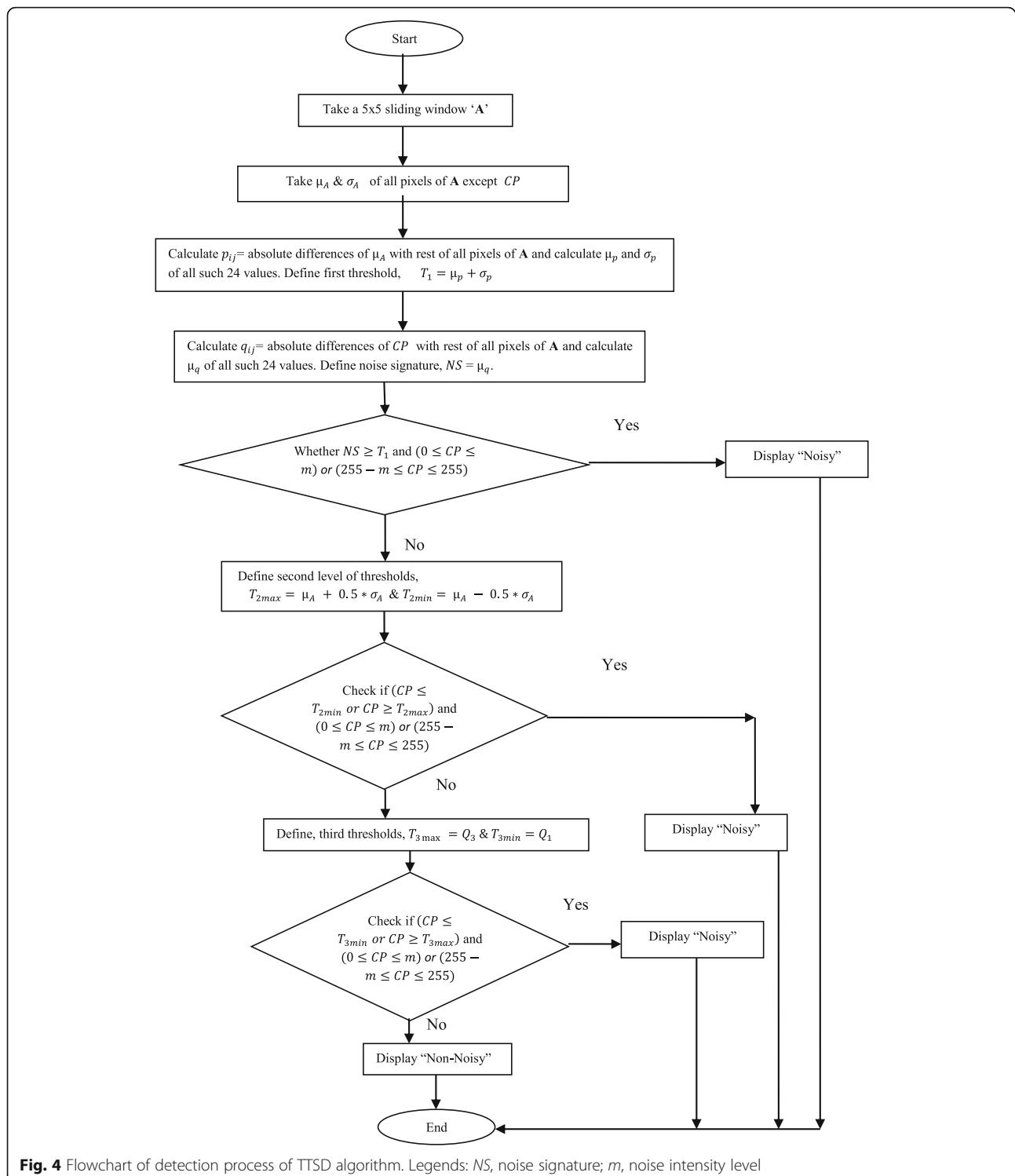
with $W_{x+s, y+t}$ as noise-free pixel.

The idea behind choosing only the noise-free pixels is to avoid selection of a noisy pixel as the median pixel.

Step VI: Now, at uniform image, regions having same intensities as noisy pixels the noise-free pixels might get detected as noisy pixels. Consequently, the filtering window is expanded continuously and the selected median pixel may not be appropriate to be used as a correction term. Considering this probability, the search for noise-free pixels is stopped when the filtering window has reached a size of 7×7 even though no noise-free pixel is detected. In such case, the first four pixels in the 3×3 filtering window are used to compute the median pixel as follows:

$$MED = \text{median}\{w_{s,t} \times (W_{x-1,y-1} + W_{x,y-1} + W_{x-1,y} + W_{x,y})\} \quad (20)$$

The first four pixels chosen, which made up the upper-left diagonal of the 3×3 filtering window, can be justified by the recursive nature of the FSWM filter [14].



Step VII: For all the noisy pixels, the median value is computed by using only the noise-free pixels surrounding it.

The restoration term is thus defined as linear combination of original pixel value and median value, i.e.,

$$r_{x,y} = (1 - f_{x,y}) \times W_{x,y} + f_{x,y} \times MED \quad (21)$$

In this process, the pixels which are marked noisy are replaced in the filtering stage and the noise-free pixels are retained the same without any modification.

Table 1 Variation in average values of *PSNR* and *SSIM* with variation in multiplication factor *K* for TID2008 database at 80% noise density

Multiplication factor <i>K</i>	<i>PSNR</i>	<i>SSIM</i>
0.1	31.32	0.291
0.2	31.95	0.420
0.3	32.55	0.516
0.4	33.08	0.611
0.5	33.71	0.784
0.6	33.19	0.658
0.7	32.09	0.534
0.8	31.21	0.383
0.9	30.27	0.188
1.0	29.30	0.104

3 Results and discussions

3.1 Experimental setup

In order to demonstrate the effectiveness of the proposed method, we initially considered three standard test images (Lena, Boat, and Cameraman) of size 256×256 and JPEG format which are extensively used in literature to measure the performance of the existing methods. Then, we considered TID2008 database of 25 images of size 256×256 and BMP format to further demonstrate the effectiveness of the TTSD algorithm over a variety of images.

These images are corrupted with RVIN of equal probability with a noise intensity level (*m*) as 4. Simulations have been carried out in MATLAB R2013a. We evaluated the performance of the proposed method for a noise density varying from 75 to 95% RVIN and compared it with that of different filters like NAFSWM [14], ROAD [23], ROLD [21], TBLI [17], ASM [8], AFIDM [3], and CBD [9].

Restoration performances are quantitatively measured by the mean square error (*MSE*) and *PSNR* in decibels which are defined as:

$$MSE = \left(\frac{1}{MN} \right) \left[\sum_{i=1}^M \sum_{j=1}^N (Y(i,j) - S(i,j))^2 \right] \quad (22)$$

$$PSNR = 10 \times \log_{10} \left(\frac{255^2}{MSE} \right) \text{ dB} \quad (23)$$

where *M* and *N* are the total number of pixels in the horizontal and vertical dimensions of the image and $Y(i,j)$ and $S(i,j)$ are the pixel values in the $(i,j)^{\text{th}}$ locations of the restored image and the uncorrupted image, respectively.

Another performance index, widely known as *SSIM* index is used for measuring the similarity between the original and restored images. It is designed to improve on traditional methods like *MSE* and *PSNR* which provide results in mere numbers and fail to represent visual performance. For two images, *x* and *y*, the *SSIM* index is defined as

$$SSIM(x,y) = [l(x,y)]^\alpha \cdot [s(x,y)]^\beta \cdot [c(x,y)]^\gamma \quad (24)$$

where $l(x,y)$, $c(x,y)$, and $s(x,y)$ are the luminance, contrast, and structure components of the index respectively. Typical values of the constants $\alpha = \beta = \gamma = 1$. MATLAB implementation of Eq. (24) has been used to calculate the *SSIM*.

Main parameters used in the proposed algorithm are mean, standard deviation, absolute difference, and quartiles. As explained already in Section 2.3, these parameters are able to detect the outliers when employed in a particular combination, i.e., the three levels of thresholds applied one after the other along with the auxiliary condition. These thresholds form a band of upper and lower limits against which the *CP* is tested for noisiness. We carried out an analysis of the optimality of multiplication factor *K* used in the second level of threshold of the proposed algorithm for TID2008 database at various

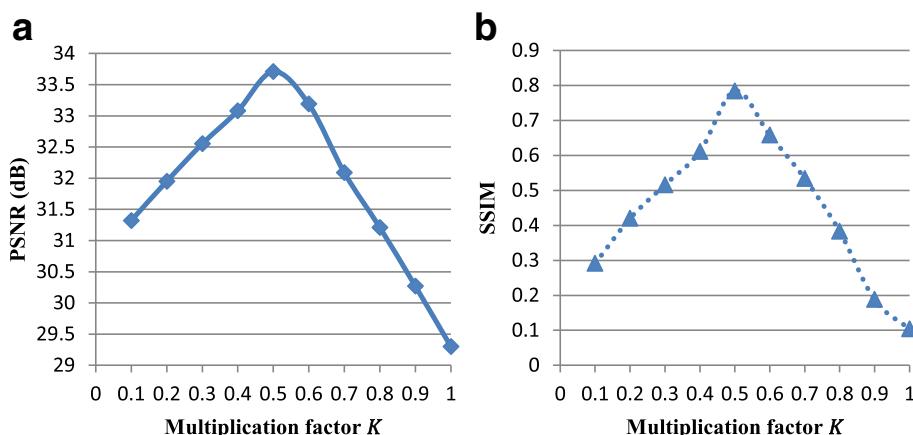


Fig. 5 Graph of variation of *PSNR* and *SSIM* with variation in multiplication factor *K*. Legends: **a** *PSNR* vs. *K* and **b** *SSIM* vs. *K*

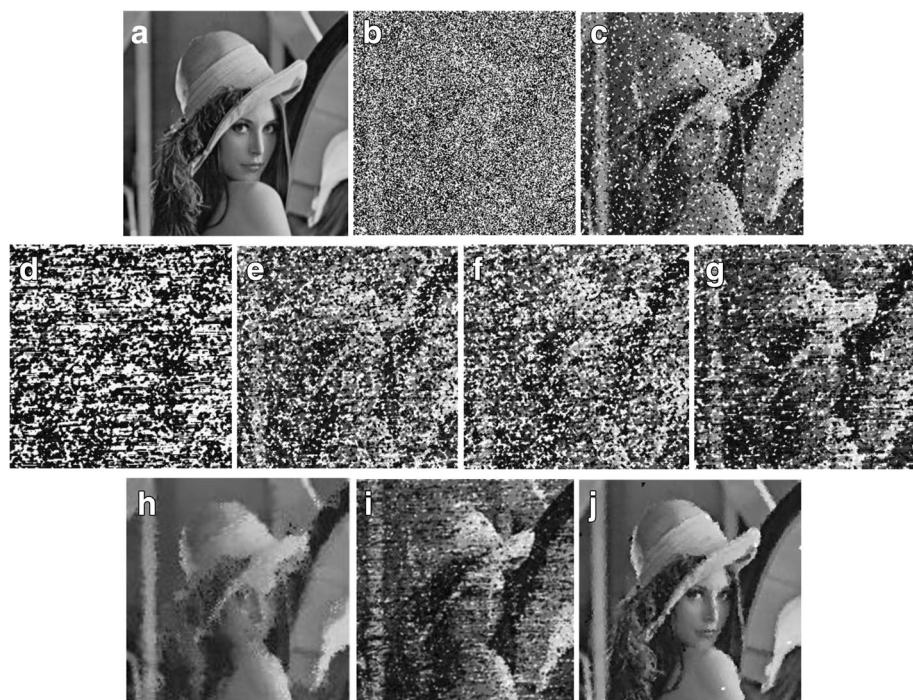


Fig. 6 Restoration results of different methods for Lena image corrupted with 80% noise density. Legends: **a** Lena (original image); **b** noisy image (80%); **c** NAFSWM; **d** ROAD; **e** ROLD; **f** TBLI; **g** ASM; **h** CBD; **i** AFIDM; and **j** TTSD

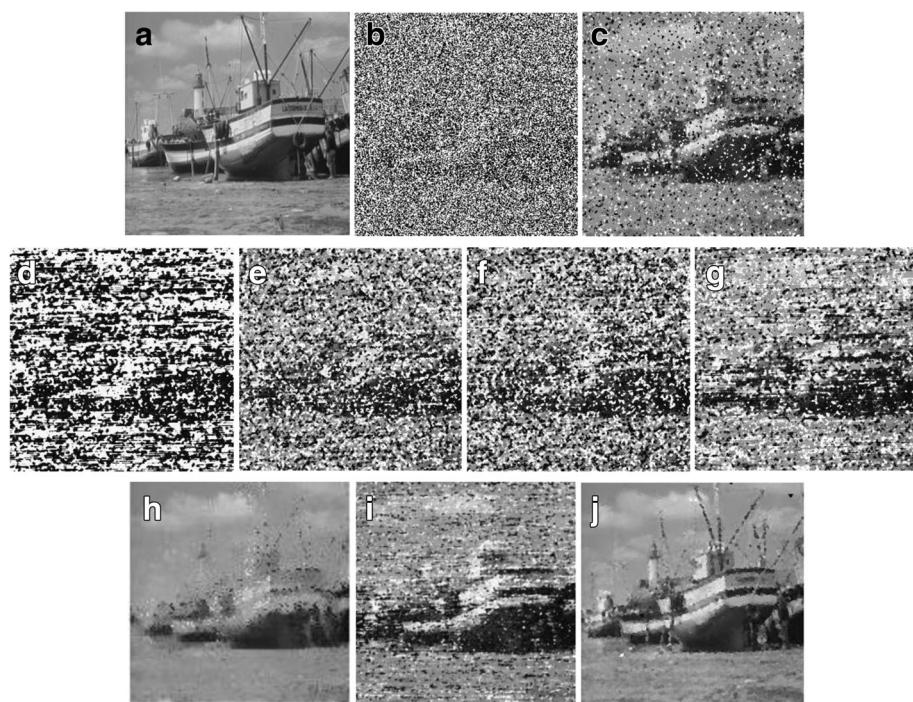


Fig. 7 Restoration results of different methods for Boat image corrupted with 80% noise density. Legends: **a** Boat (original image); **b** noisy image (80%); **c** NAFSWM; **d** ROAD; **e** ROLD; **f** TBLI; **g** ASM; **h** CBD; **i** AFIDM; and **j** TTSD

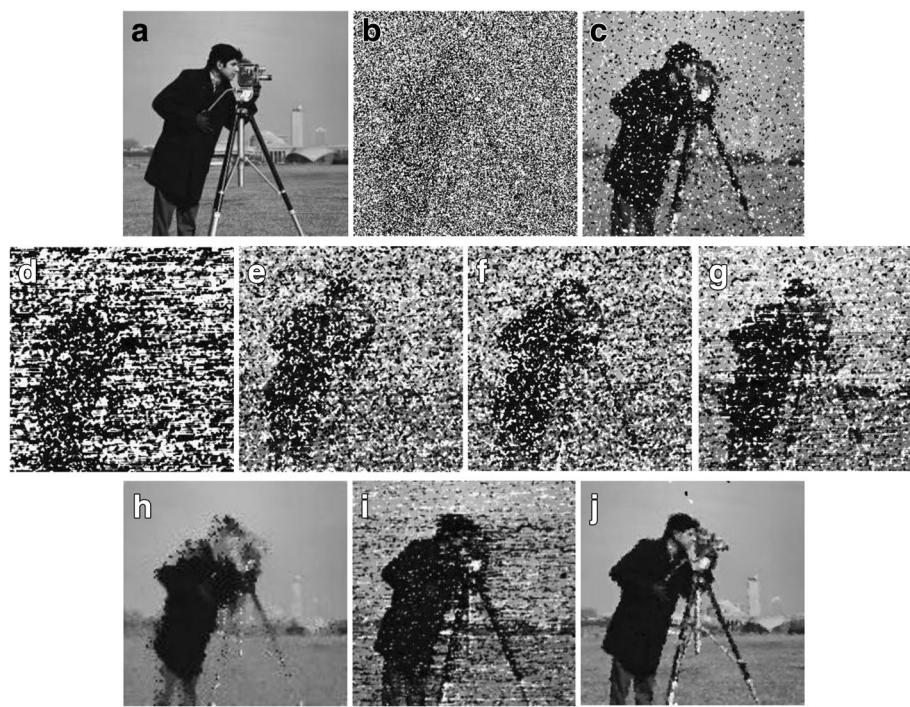


Fig. 8 Restoration results of different methods for Cameraman image corrupted with 80% noise density. Legends: **a** Cameraman (original image); **b** noisy image (80%); **c** NAFSWM; **d** ROAD; **e** ROLD; **f** TBLI; **g** ASM; **h** CBD; **i** AFIDM; and **j** TTSD

noise densities. The results of this analysis at 80% noise density are presented in the Table 1 and Fig. 5.

From Table 1 and Fig. 5, we see that the highest values of performance parameters *PSNR* and *SSIM* (in italics) are achieved when value of *K* is 0.5. For values below and above 0.5, the values of *PSNR* and *SSIM* are lower than what they are at *K* = 0.5. Hence, the value of *K* has been

considered as 0.5 in the proposed algorithm. The rest of the parameters used in the proposed algorithm assume different values depending upon the value of noise density.

3.2 Results on simulated images

Figures. 6, 7, and 8 show the restoration results of different filters, i.e., NAFSWM, ROAD, ROLD, TBLI, ASM,

Table 2 Restoration results in *PSNR* (dB) for Lena, Boat, and Cameraman images corrupted by RVIN

Noise density (%)	Image	NAFSWM [14]	ROAD [25]	ROLD [23]	TBLI [19]	ASM [10]	CBD [9]	AFIDM [3]	TTSD
75	Lena	30.82	28.65	29.81	30.05	31.57	31.33	31.12	34.87
	Boat	30.39	27.70	29.61	29.18	29.57	31.68	30.41	33.52
	Cameraman	31.27	28.68	30.27	30.61	30.84	33.12	31.32	35.06
80	Lena	30.66	27.96	29.12	29.13	30.26	31.57	30.63	34.15
	Boat	30.23	27.91	29.15	29.96	29.15	30.81	29.81	32.99
	Cameraman	31.40	28.30	29.46	29.43	29.45	32.26	31.13	34.63
85	Lena	30.70	27.63	28.33	28.48	29.51	30.31	29.85	33.46
	Boat	30.21	27.56	28.45	28.41	28.17	29.76	29.11	32.68
	Cameraman	31.13	27.97	28.63	28.65	28.50	31.13	30.43	34.03
90	Lena	30.47	27.27	27.85	27.84	28.26	29.70	29.14	32.77
	Boat	30.08	27.43	27.93	27.88	27.72	29.63	28.27	31.99
	Cameraman	30.97	27.54	27.94	28.05	27.96	29.92	29.94	33.46
95	Lena	30.29	27.17	27.36	27.33	27.72	29.49	28.42	31.98
	Boat	29.90	27.35	27.44	27.48	27.37	29.40	27.61	31.14
	Cameraman	30.69	27.36	27.45	27.52	27.65	28.38	28.84	32.69

Table 3 Restoration results in SSIM for Lena, Boat, and Cameraman images corrupted by RVIN

Noise density (%)	Image	NAFSWM [14]	ROAD [25]	ROLD [23]	TBLI [19]	ASM [10]	CBD [9]	AFIDM [3]	TTSD
75	Lena	0.196	0.030	0.074	0.076	0.118	0.661	0.563	0.861
	Boat	0.161	0.023	0.078	0.050	0.125	0.601	0.531	0.789
	Cameraman	0.159	0.027	0.074	0.067	0.112	0.723	0.571	0.859
80	Lena	0.161	0.023	0.052	0.043	0.083	0.589	0.502	0.829
	Boat	0.136	0.019	0.056	0.079	0.088	0.559	0.488	0.748
	Cameraman	0.138	0.024	0.048	0.049	0.076	0.692	0.529	0.832
85	Lena	0.129	0.016	0.031	0.030	0.050	0.497	0.474	0.782
	Boat	0.116	0.015	0.034	0.033	0.059	0.471	0.453	0.701
	Cameraman	0.107	0.014	0.030	0.034	0.046	0.609	0.483	0.792
90	Lena	0.108	0.015	0.022	0.018	0.032	0.382	0.314	0.718
	Boat	0.088	0.016	0.020	0.020	0.042	0.401	0.294	0.628
	Cameraman	0.087	0.012	0.021	0.021	0.031	0.534	0.318	0.746
95	Lena	0.071	0.009	0.012	0.012	0.018	0.330	0.263	0.656
	Boat	0.074	0.006	0.008	0.010	0.017	0.357	0.244	0.579
	Cameraman	0.059	0.011	0.010	0.012	0.021	0.482	0.274	0.685

CBD, AFIDM, and TTSD for “Lena,” “Boat,” and “Cameraman” images corrupted by 80% of RVIN respectively. Tables 2 and 3 show the PSNR and SSIM values of various filters for these images corrupted with different levels of RVIN. The highest values are given in italics.

In addition to the three standard test images, i.e., Lena, Boat, and Cameraman, simulations have also been carried out on the TID2008 database of 25 images of size 256×256 for validating the proposed method. The average values of PSNR and SSIM have been calculated for all methods and compared for the TID2008 database in Tables 4 and 5 below. The highest values are given in italics. Figures 9 and 10 show the graphs of comparison of PSNR and SSIM of various methods for TID2008 database, respectively. The image restoration results of TID2008 image database for 80% noise density are shown in the Fig. 11.

3.3 Discussions

From the results of existing methods, we observe that there is miss detection of noisy pixels as noise-free and vice versa. However, in case of the NS-based TTSD, miss detection is substantially minimized till 80% noise density and some miss detection occurs in TTSD at high

noise densities of more than 80%. As already mentioned, the TTSD method uses the concept of noise signature and multiple thresholds to detect noisy pixels. A level-by-level verification is performed to avoid miss detection of noise-free pixels as noisy pixels.

From Figs. 6, 7, 8, 9, 10, and 11 and Tables 2, 3, 4, and 5, we observe that the NAFSWM [14] filter does not yield good restoration results at 80% noise density because it utilizes the histogram of the corrupted image to identify noise pixels as the noise detection algorithm which produces satisfactory results till 50% noise density only. Similarly, ROAD [25] is also good only up to 50% noise density beyond which miss detection is very high. The performance of ROLD [23] is better than that of the ROAD because it uses the logarithmic difference for detection, but after 60% noise density, it also does not provide good results. TBLI [19] detection algorithm is based on triangle-based linear interpolation which is used to detect noisy and noise-free pixels. TBLI [19] is also good only up to 60% noise density beyond which miss detection increases and picture quality worsens. In ASM’s detection stage, noisy pixels are detected using the absolute deviation between the mean value and

Table 4 Restoration results in average PSNR (dB) for TID2008 database corrupted by RVIN

Noise density (%)	NAFSWM [14]	ROAD [25]	ROLD [23]	TBLI [19]	ASM [10]	CBD [9]	AFIDM [3]	TTSD
75	30.98	27.64	29.96	29.85	30.07	32.02	30.58	34.18
80	30.84	27.47	29.21	29.16	29.43	31.17	30.12	33.71
85	30.68	27.28	28.83	28.76	28.75	30.65	29.47	33.12
90	30.35	27.14	28.13	28.09	28.11	29.86	28.78	32.54
95	30.11	27.05	27.52	27.52	27.59	29.42	27.83	31.80

Table 5 Restoration results in average SSIM for TID2008 database corrupted by RVIN

Noise density (%)	NAFSWM [14]	ROAD [25]	ROLD [23]	TBLI [19]	ASM [10]	CBD [9]	AFIDM [3]	TTSD
75	0.156	0.022	0.071	0.068	0.115	0.645	0.545	0.821
80	0.127	0.018	0.046	0.044	0.096	0.582	0.493	0.784
85	0.105	0.015	0.034	0.031	0.061	0.502	0.443	0.748
90	0.086	0.013	0.024	0.022	0.039	0.428	0.332	0.675
95	0.062	0.011	0.011	0.011	0.016	0.341	0.252	0.621

the *CP* which is then compared with threshold value. ASM [10] though provides better value of *PSNR*; it does not give good quality picture restoration because of miss detection at higher noise densities.

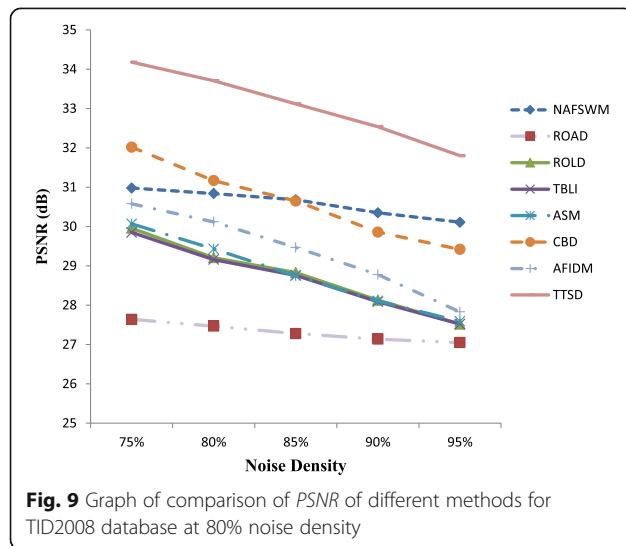
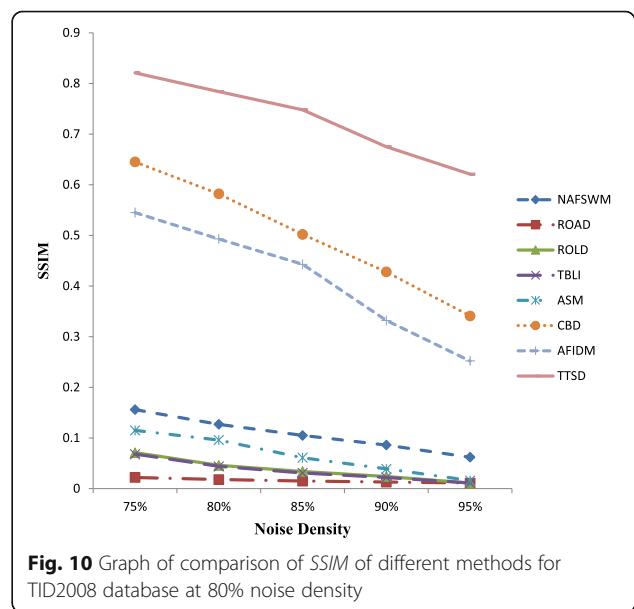
The CBD [9] and AFIDM [3] are two methods among the existing methods which provide relatively better results till high noise densities. In these methods also, the detection is satisfactory up to 60% and miss detection occurs beyond that which results in blurring and haziness. In contrast to CBD, the NS-based TTSD is based on adaptive thresholds and hence the algorithm used is very effective. Further, in contrast to AFIDM, the TTSD algorithm uses three levels of thresholds which provide high quality of detection. The detection in TTSD is satisfactory up to 80% with some miss detection. This results in better restoration of image and better contrast as compared to the CBD.

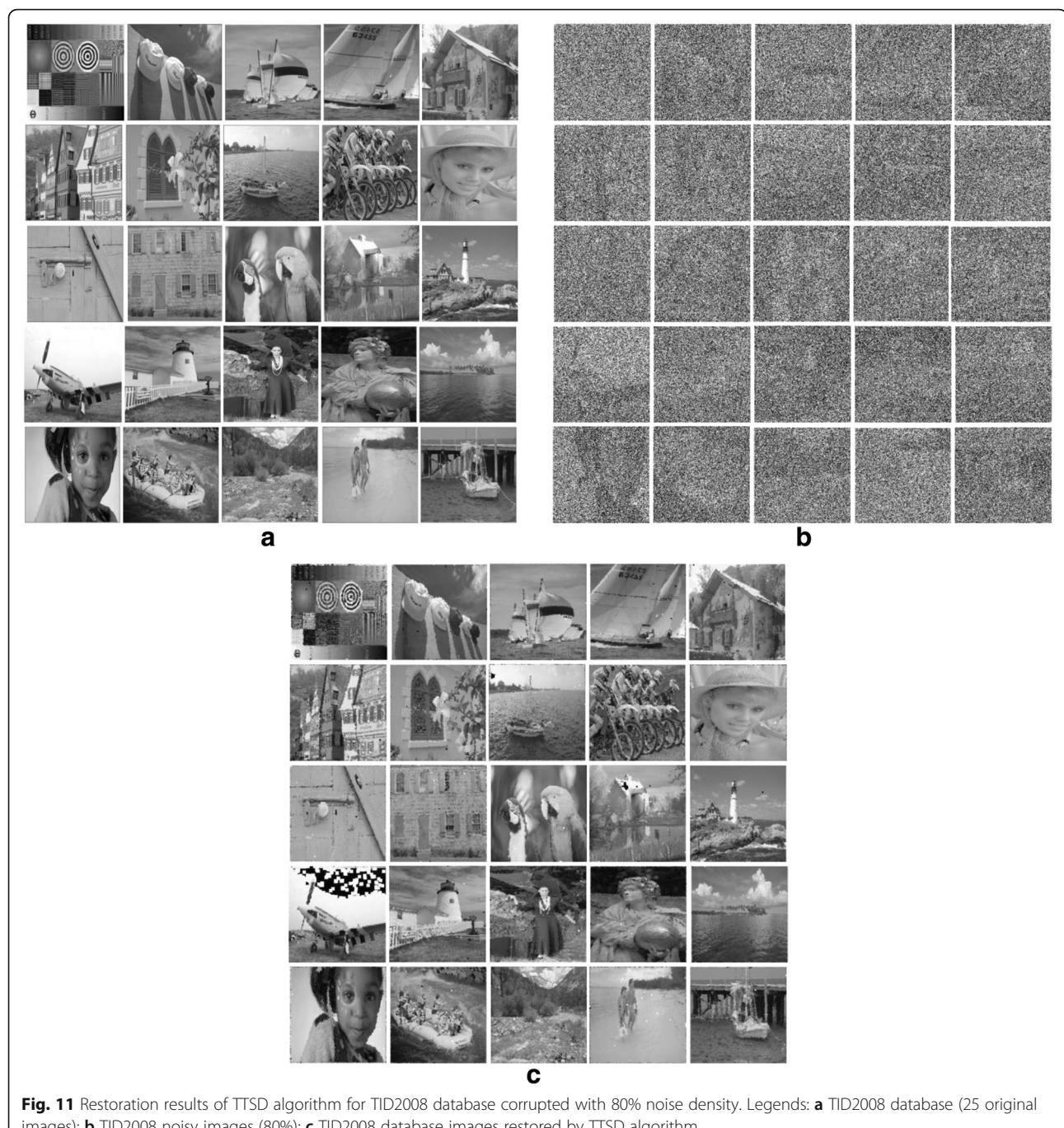
From Table 4 and Fig. 9, we see that the TTSD filter achieves higher *PSNR* values even when the noise level is very high, e.g., at 95%. This is mainly because of the relatively better noise detection and efficient fuzzy switching weighted median filtering compared to the different kinds of median filtering used in other methods. Further, from Table 5 and Fig. 10, it is evident that the restoration performance of the TTSD filter is better in terms of *SSIM* than that of the other filters even when the noise density is higher than 80%. The higher values

of *SSIM* produced by TTSD are also in consonance with the visual performance of the three restored images in Figs. 6, 7, and 8, whereby it is evident that the TTSD filter is able to preserve the contrast when compared with the other filters.

We also observed consistency in effectiveness of the proposed algorithm at high noise densities as compared to that of existing algorithms. For high noise densities beyond 75%, the proposed method TTSD consistently provides higher *PSNR* and *SSIM* values as compared to those of NAFSWM, ROAD, ROLD, TBLI, ASM, CBD, and AFIDM. As is evident from Fig. 9, the rate of decrease in *PSNR* is lower for TTSD as compared to those of other existing methods except NAFSWM and ROAD. Further, we observe from Fig. 10 that the rate of decrease in *SSIM* for TTSD is lower than those of CBD and AFIDM which are the other two significant algorithms so far as *SSIM* is concerned. This shows that the deterioration in the effectiveness of the proposed method with increase in noise density is not as steep as it is in case of other existing methods.

Further, the picture quality of TTSD is better than all other methods for high noise densities. This is because the TTSD filter satisfactorily addresses miss detection

**Fig. 9** Graph of comparison of *PSNR* of different methods for TID2008 database at 80% noise density**Fig. 10** Graph of comparison of *SSIM* of different methods for TID2008 database at 80% noise density



80%, whereas the miss detection in other methods increases very sharply with the increase of noise density. Beyond 80%, miss detection also increases in TTSD at a slow rate.

3.4 Rational for using three levels of thresholds

There are various existing algorithms which use one or two thresholds for detection of noisy pixels. The proposed method TTSD uses three levels of thresholds to

further refine the detection process. To improve the accuracy of the detection process, one can go on adding the levels of thresholds; however, it will result in high computational complexity. By carrying out extensive simulations on different images, the TTSD could arrive at the optimal level of three thresholds which is proposed in this study. It not only yields higher level of accuracy in detection but also provides the low computational complexity.

Table 6 Average computation time of various algorithms for TID2008 database corrupted by RVIN (unit: seconds)

Noise density (%)	NAFSWM [14]	ROAD [25]	ROLD [23]	TBLI [19]	ASM [10]	CBD [9]	AFIDM [3]	TTSD
75	5.87	1.93	1.69	17.79	8.82	1.56	9.12	5.62
80	5.98	1.99	1.76	18.18	8.90	1.60	9.54	5.81
85	6.28	2.02	1.90	18.54	9.06	1.64	9.89	6.12
90	6.83	2.09	2.04	19.04	9.17	1.72	10.21	6.38
95	7.39	2.32	2.15	19.72	9.33	1.97	10.65	6.64

Each level of threshold in TTSD acts as a filter to separate noisy pixels from the rest and passes only those pixels to the next levels which are not detected as noisy pixels. Therefore, TTSD keeps on separating the noisy pixels after each level and probes the noise-free pixels level by level unless the three proposed statistical conditions of noisiness are verified. For every central pixel in a sliding window, unique threshold value is determined based on the surrounding pixel values. Therefore, the performance of detection in TTSD enhances.

3.5 Speed comparison

To carry out speed comparison of the proposed method with the existing methods and to study their computational complexities, we calculated the computation time taken by these methods for noise densities varying from 75 to 95%. Table 6 and Fig. 12 show the comparison of average computation time taken by various algorithms for the TID2008 database from 75 to 95% noise densities. The highest values are given in italics.

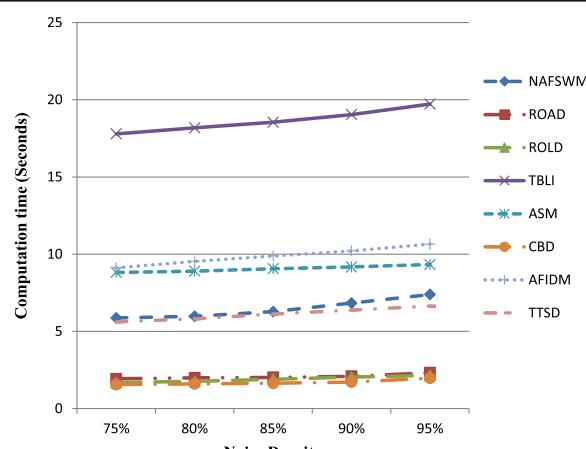
From Table 6 and Fig. 12, it is evident that the TTSD takes less computation time than NAFSWM, TBLI, ASM, and AFIDM and more computation time than ROAD, ROLD, and CBD. We also see that the computation time increases with the increase in the noise density in all methods. However, the rate of increase in computation time for TTSD is lower than the existing methods.

To avoid miss detection, TTSD uses three levels of thresholds along with an auxiliary condition which involve various calculations. Therefore, the *PSNR* and *SSIM* values of TTSD are better than those of the existing methods and the computation time taken is higher than ROAD, ROLD, and CBD. This again demonstrates the effectiveness of the TTSD algorithm as against the existing methods. Our future work will concentrate on further reducing the computation time as well as reduction of the miss detection above 80% RVIN. This can be achieved by use of parallel processing architect in place of series processing [31, 32].

These simulations demonstrate that the proposed method achieves better results at higher noise densities than the existing methods for a wide range of images. Since the restored picture quality is comparably good in TTSD, its average SSIM values at higher noise densities are far better than those of NAFSWM, ROAD, ROLD, TBLI, ASM, CBD, and AFIDM.

4 Conclusions

In this paper, a new method based on noise signature is presented for effective removal of random-valued impulse noise in images. It is extremely difficult to restore images corrupted with very high noise densities. Most of the existing algorithms provide good results only up to low noise densities. Some of them use pre-defined thresholds for detection which may not be

**Fig. 12** Average computation time of various algorithms for TID2008 database corrupted by RVIN

effective in case of randomly picked up images at higher noise densities. In contrast, the proposed noise signature-based triple threshold statistical detection method promises better results in the case of highly corrupted random images too. This is possible because it does not use any predefined threshold values. Simulation results demonstrate the efficacy of this method in removing high density random-valued impulse noise while preserving image details. The proposed method is able to suppress up to 80% of noise with a considerable level of satisfaction due to effective use of noise signature. Further, the results reveal that the method performs better than the existing state-of-the-art filtering approaches. Thus, the proposed triple threshold statistical detection method is an adaptive filtering approach which can be considered as a high-performance method in its class.

Abbreviations

AFIDM: Adaptive fuzzy inference system-based directional median filter; ASM: Adaptive switching median filter; BMP: Bitmap; CBD: Condition-based detection filter; CP: Central pixel; FSWM: Fuzzy switching weighted median filter; IQR: Inter-quartile range; JPEG: Joint photographic experts group; MSE: Mean square error; NAFSWM: Noise adaptive fuzzy switching weighted median filter; NS: Noise signature; PSNR: Peak signal to noise ratio; ROAD: Rank order absolute difference; ROLD: Rank order logarithmic difference; RVIN: Random-valued impulse noise; SPN: Salt and pepper noise; SSIM: Structural similarity index; TBLI: Triangle-based linear interpolation detection filter; TID2008: Tampere image database 2008; TTSD: Triple Threshold Statistical Detection

Acknowledgements

Not applicable.

Funding

Not applicable.

Availability of data and materials

The datasets supporting the conclusions of this article are included within the article and its additional files.

Authors' contributions

NS has made substantial contributions to the conception and design of the algorithm, analysis and interpretation of data, and drafting of the manuscript. OU has revised it critically for important intellectual content and has given final approval of the version to be published. Both authors read and approved the final manuscript.

Authors' information

Corresponding author, Neeti Singh, is a graduate in Electronics and Telecommunication and post graduate in Applied Electronics; she started her career as Assistant Executive Engineer, GSM project in ITI limited, Raipur, India, in 2005. She has also worked as Switching Engineer in Alcatel-lucent, Chennai, India. She presented papers in IEEE International conference ICECCN-13, India; ADELCO 9th National conference, India; NCCS-15, India; and IEEE International conference ICSCN-17, India. Presently, she is pursuing her PhD from Anna University, Chennai, India.
Second author, O. Umamaheswari, received her B.E. Degree in Electronics and Communication Engineering and M.E. Degree in Communication Engineering from Thiagarajar College of Engineering, Madurai, India, in the year 1998 and 1999 respectively. She obtained her Ph.D. Degree in the area "Nonlinear signal processing" in the year 2009 from Anna University, Chennai, India. She is currently an Assistant Professor (Senior Grade) in the Department of Electronics and Communication Engineering, College of Engineering, Guindy, Anna University, Chennai, India. Her research includes

nonlinear signal processing, optimization techniques, and analysis of FMRI. She is a member of IEEE and life member of Computer Society of India.

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.

Received: 9 February 2017 Accepted: 20 March 2018

Published online: 04 April 2018

References

1. RC Gonzalez, RE Woods, *Digital Image Processing*, 2nd edn. (Prentice Hall, New Jersey, 2002)
2. AS Awad, Standard deviation for obtaining the optimal direction in the removal of impulse noise. *IEEE Signal Process. Lett.* **18**(7), 407–410 (2011)
3. M Habib, A Hussain, S Rasheed, M Ali, Adaptive fuzzy inference system based directional median filter for impulse noise removal. *Int. J. Electron. Commun. (AEU)* **70**(5), 689–697 (2016)
4. P Shrivastav, S Changlani, Advance impulse noise filtering using double threshold with edge preservation. *Int. J. Comput. Appl.* **115**(3), 24–27 (2015)
5. G Hanji, Adaptive tolerance filter for RVIN suppression. *J. Adv. Comput. Sci.* **3**(1), 1–11 (2015)
6. C Yan, Y Zhang, J Xu, F Dai, J Zhang, Q Dai, F Wu, Efficient parallel framework for HEVC motion estimation on many-core processors. *IEEE Trans. Circuits Syst. Video Technol.* **24**(12), 2077–2089 (2014)
7. C Yan, Y Zhang, J Xu, F Dai, L Li, Q Dai, F Wu, A Highly, Parallel framework for HEVC coding unit partitioning tree decision on many-core processors. *IEEE Signal Process. Lett.* **21**(5), 573–576 (2014)
8. H Dawood, P Guo, Removal of random valued impulse noise by local statistics. *Multimed. Tools Appl. - Int. J.* **74**(24), 11485–11498 (2014)
9. ZM Ramadan, A New, Method for impulse noise elimination and edge preservation. *Can. J. Electr. Comput. Eng.* **37**(1), 2–10 (2014)
10. X Lan, Z Zuo, Random valued impulse noise removal by the adaptive switching median detector and detail-preserving regularization. *Optik-Int. J. Light Electron. Opt.* **125**(3), 1101–1105 (2014)
11. CY Lien, CC Huang, PY Chen, An efficient denoising architecture for removal of impulse noise in images. *IEEE Trans. Comput.* **62**(4), 631–643 (2013)
12. R Pushpavalli, G Sivaradjie, Asymmetric trimmed median filter for images highly corrupted with random valued impulse noise. *Int. J. Comput. Appl.* **44**(7), 1506–1516 (2012)
13. R Seetharaman, R Vijayaraghavan, Removal of random valued impulse noise by directional mean filter using statistical noise based detection. *Int. J. Comput. Appl.* **46**(10), 14–18 (2012)
14. C Suganya, O Umamaheswari, in *Proc. Defense Science Research Conference and Expo (DSR)*. Image restoration using noise adaptive fuzzy switching weighted median filter for the removal of impulse noise (IEEE, Singapore, 2011), pp. 1–4
15. KKV Toh, NAM Isa, Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction. *IEEE Signal Process. Lett.* **17**(3), 281–284 (2010)
16. VR Vijaykumar, P Jothibasu, in *Proc. IEEE 17th International Conference on Image Processing (ICIP)*. Decision based adaptive median filter to remove blotches, scratches, streaks, stripes and impulse noise in image (IEEE, Hong Kong, 2010), pp. 117–120
17. V Jayaraj, D Ebenezer, A new switching based median filtering scheme and algorithm for removal of high density salt and pepper noise in image. *EURASIP J. Adv. Signal Process.* **2010**, 1–11 (2010)
18. PK Sa, R Dash, B Majhi, in *Proc. Fourth International Conference on Industrial and Information Systems (ICIIS)*. Second order difference based detection and directional weighted median filter for removal of random valued impulsive noise (IEEE, Sri Lanka, 2009), pp. 362–364
19. P Civicioglu, Removal of random-valued impulsive noise from corrupted images. *IEEE Trans. Consum. Electron.* **55**(4), 2097–2104 (2009)

20. H Ibrahim, NSP Kong, TF Foo, Simple adaptive median filter for the removal of impulse noise from highly corrupted images. *IEEE Trans. Consum. Electron.* **54**(4), 1920–1927 (2008)
21. KS 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)
22. Y Dong, A new directional weighted median filter to removal of random valued impulse noise. *IEEE Signal Process. Lett.* **14**(3), 193–196 (2007)
23. Y Dong, RH Chan, S Xu, A Detection, Statistic for random-valued impulse noise. *IEEE Trans. Image Process.* **16**(4), 1112–1120 (2007)
24. W Luo, A new efficient impulse detection algorithm for the removal of impulse noise. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **E88C**(10), 2579–2586 (2005)
25. R Garnett, T Huegerich, C Chui, WJ He, A universal noise removal algorithm with an impulse detector. *IEEE Trans. Image Process.* **14**(11), 1747–1754 (2005)
26. T Chen, HR Wu, Adaptive impulse detection using center weighted median filters. *IEEE Signal Process. Lett.* **8**(1), 1–3 (2001)
27. MS Moore, M Gabbouj, SK Mitra, in *Proc. Second EURASIP Conference on DSP for Multimedia Communications and Services (ECMCS)*. Vector SD-ROM filter for removal of impulse noise from color images (EURASIP, Poland, 1999)
28. Z Wang, D Zhang, Progressive switching median filter for the removal of impulse noise from highly corrupted images. *IEEE Trans Circuits Syst* **46**(1), 78–80 (1999)
29. N Singh, O Umamaheswari, in *Proc. Fourth International Conference on Signal Processing, Communication and Networking (ICSCN-17)*. A new denoising algorithm for random valued impulse noise in images using measure of dispersion (IEEE, India, 2017), pp. 1–6
30. N Singh, O Umamaheswari, Some studies on detection and filtering algorithm for the removal of random valued impulse noise. *IET Image Process.* **11**(11), 953–963 (2017)
31. C Yan, H Xie, S Liu, J Yin, Y Zhang, Q Dai, Effective Uyghur language text detection in complex background images for traffic prompt identification. *IEEE Trans. Intell. Transport. Syst.* **19**(1), 220–229 (2017)
32. C Yan, H Xie, D Yang, J Yin, Y Zhang, Q Dai, Supervised hash coding with deep neural network for environment perception of intelligent vehicles. *IEEE Trans. Intell. Transport. Syst.* **19**(1), 284–295 (2017)

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com