Information Sciences 315 (2015) 1–16

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A new weighted mean filter with a two-phase detector for removing impulse noise

Licheng Liu [a], C.L. Philip Chen [a,*], Yicong Zhou [a], Xinge You [b]

[a] Department of Computer and Information Science, University of Macau, 999078, Macau
[b] Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

ARTICLE INFO

Article history:
Received 27 August 2014
Received in revised form 24 March 2015
Accepted 29 March 2015
Available online 3 April 2015

Keywords:
Impulse noise
Fuzzy noise detector
Edge pixels identification
Weighted mean filter
Image denoising

ABSTRACT

This paper proposes a new weighted mean filter with a two-phase noise detector for image denoising. Operations are carried out by the detection followed by filtering strategy. For detection, a two-phase noise detector is presented to detect impulse noise (IN). In the first phase, a rank-ordered difference of ROAD (ROD-ROAD) scheme is introduced for detecting noise candidates. Different from most of the existing IN detectors, the proposed detector identifies a pixel by a fuzzy rule that matches the stochastic nature of IN and greatly improves the denoising performance. In the second phase, a local image statistic minimum edge pixels difference (MEPD) is proposed to identify edge pixels from noise candidates. This preserves edges from being wrongly detected as noise; therefore, improves the detection accuracy. For filtering, we design a new weighted mean filter (WMF) that is more suitable for IN to suppress the detected noisy pixels. Finally, an iterative denoising algorithm is presented by combining the proposed two-phase noise detector and the new WMF. The proposed method is accessible and easy to implement. Experimental results show that the proposed method outperforms all the tested state-of-the-art denoising methods with respect to the visual effects and quantitative measure results.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Digital images are frequently corrupted by IN due to a noisy camera sensor and faulty analog to digital conversion [20]. Images with noise can severely hamper the subsequent image processing such as segmentation [10], target detection [9], and classification [21,24]. Therefore, removing noise from the corrupted images is absolutely obligatory and very important. The goal of image denoising is removing noise as much as possible while preserving more image details.

Unlike Gaussian noise [35,42], the characteristic of IN is that, for images contaminated by IN, not all the pixel values are changed, but only a portion of the pixels are replaced by noise. For simplicity, let $x_{i,j}$ and $c_{i,j}$ be the pixel intensities at (i,j)th position in the noisy and clean images respectively, and the pixel values be bounded by n_{min} and n_{max} ($n_{max} > n_{min}$). Then, the IN model can be described as,

$$x_{i,j} = \begin{cases} c_{i,j} & p = 1 - p_0 \\ n_{i,j} & p = p_0 \end{cases} \tag{1}$$

* Corresponding author. Tel.: +853 83974950.
E-mail address: Philip.Chen@ieee.org (C.L.P. Chen).

http://dx.doi.org/10.1016/j.ins.2015.03.067
0020-0255/© 2015 Elsevier Inc. All rights reserved.

where $n_{i,j}$ denotes the gray-value of noise, p_0 is the noisy probability. Generally, there are two kinds of IN, namely, salt & pepper noise (SPN) and random-valued impulse noise (RVIN). For SPN, the values of noisy pixels are chosen from n_{min} and n_{max}. While for RVIN, the intensities of corrupted pixels can be changed into any value in $[n_{min}, n_{max}]$. Compared with SPN, RVIN is more difficult to be removed.

In order to suppress IN, various filters and techniques have been employed. The median (MED) filter [31], famous for its simplicity, is one of the most common tools. However, the MED may remove some desirable details in images, especially when the noise density is higher. Therefore, several MED extensions were then proposed to obtain better denoising performance. Such as the weighted median (WM) filter [7], the center weighted median (CWM) filter [23], and the recursive weighted median filter (RWMF) [4]. By assigning weights to emphasize the desirable pixels, these filters can achieve better filtering performance. Nevertheless, they still degrade the image quality because they just process each pixel in the noisy image without considering whether it is noise or not.

To solve the above problem, several filtering techniques integrated with noise detectors have been developed. The noise detector, prior to filtering, is used to distinguish the corrupted and clean pixels. Then the detected noise pixels are filtered while the clean ones remain unchanged. Many of these techniques are based on the median- or mean-type filters, examples including the adaptive center-weighted median (ACWM) filter [11], Luo-iterative method [29], the directional weighted median (DWM) filter [16], the adaptive switching median (ASWM) filter [3], the optimal direction median filter [5], the two-pass switching rank-ordered arithmetic mean (TSRAM) filter [27], and the ROR non-local mean (ROR-NLM) [37].

In [18], based on the assumption that in natural images, the information pixels always have strong relationship with their neighbors, Garnett et al. introduced a local statistic rank-ordered absolute difference (ROAD) to describe noisy pixels. Later, Dong et al. [15] found that the detection accuracy can be improved by using a logarithmic function to amplify the differences between the noisy and clean pixels. Hence, based on the ROAD, they proposed a new statistic ROLD and united it with edge-preserving regularization (EPR) [30] for suppressing RVIN. Though the ROAD and ROLD can detect most of the IN, they may be failed in edges. To overcome this shortcoming, generating a relative difference image, Yu et al. [38] proposed a rank-ordered relative differences(RORD) to identify noise. Using a reference image, the RORD can preserve most edge pixels. However, it excessively depends on the reference image which is not guaranteed to be entirely clean. Hence it still damages some image details, especially when the noise density is high. In [28], combined with noise detector, the sparse representation technique was also extended for IN removal.

Recently, some techniques using the fuzzy rule for denoising have been developed [33]. The fuzzy rule is suitable for IN due to its inherent uncertainty feature [43]. It has been verified that the fuzzy detectors or fuzzy filters could achieve good performance in removing SPN even the noise density is as high as 90% [2]. Unfortunately, most of these fuzzy techniques are proposed for SPN, few are suitable for RVIN.

In this paper, we propose a new image denoising algorithm to remove RVIN. The proposed algorithm is also based on the "detecting then filtering" trick. For detecting noise, a two-phase noise detector is proposed. In the first phase, a ROD-ROAD scheme is presented to distinguish the noise-like and clean-like pixels. The absolute deviation to the median of clean-like pixels in the sliding window is calculated, and a membership function is introduced to describe how noise-like a pixel is. Since edge pixels are always wrongly detected as noise, in the second phase, we propose a statistic MEPD to identify the edge pixels which are falsely detected as noise. This process is necessary to preserve edges from being damaged. For filtering, a new weighted mean filter (WMF), which takes into account not only the information of image features but also the IN characteristics, is designed to remove the detected noisy pixels. Our main contributions are summarized as follows,

- Introduce a new ROD-ROAD scheme to generate a cleaner reference variable for detecting noise.
- The possibility of a pixel corrupted by IN is described in a fuzzy rule by using a membership function.
- Propose a local statistic MEPD to select out edge pixels. To the best of our knowledge, this is the first time to identify edge pixels from the noise candidates, which helps to preserve the image details.
- Integrating the image features and IN characteristics, design a new WMF which is more suitable for impulse noise removal.

The rest of this paper is organized as follows. Section 2 shows some related works. Section 3 introduces the two-phase noise detector. The new weighted mean filter is introduced in Section 4. The iterative denoising algorithm is presented in Section 5. Simulation results are shown in Section 6 and Section 7 reaches a conclusion.

2. Related works

In this section, we review some studies related to our work that focus on the noise detectors and filters design for IN removal. We first briefly review some noise detection schemes and then introduce some switching filters.

2.1. Noise detectors

One crucial problem for IN removal is the noise detection. Most existing IN detectors can be classified into two types, one is based on the absolute deviation defined as,

$$d(x_{i,j}) = |x_{i,j} - \Omega_{i,j}| \tag{2}$$

where $\Omega_{i,j}$ denotes the reference variable calculated from the local information. This absolute deviation is further compared with an appropriate threshold T. Then a binary matrix f is chosen to record the compared results.

$$f(i,j) = \begin{cases} 1; & d(x_{i,j}) \geqslant T \\ 0; & d(x_{i,j}) < T \end{cases} \tag{3}$$

where the value "1" means that the current pixel $x_{i,j}$ is a noisy pixel, otherwise $x_{i,j}$ is a clean pixel.

Over the years, various local statistics are used as the reference variables. For example, $\Omega_{i,j}$ is replaced by the median or weighted median in [29,32,37], normalized mean in [17], rank order in [1], center-weighted median in [12], median of the absolute deviations from the median (MAD) in [13], directional weighted median in [16], weighted mean in [3], and median of sorted quadrant median vector (SQMV) in [25].

The other one is based on the absolute differences between the center pixel and its neighbors. Denote $x_{k,l}$ be the neighbor pixels of $x_{i,j}$ within a local window, then the absolute difference is defined by,

$$d_{i,j}(k,l) = |x_{i,j} - x_{k,l}| \tag{4}$$

A typical representative of such detection scheme is the rank-ordered absolute difference (ROAD) [18],

$$\mathrm{ROAD}_m(x_{i,j}) = \sum_{n=1}^{m} d_{i,j}^s(n) \tag{5}$$

where $d_{i,j}^s(n)$ is the n-th smallest one in the $d_{i,j}(k,l)$ defined in Eq. (4). The ROAD value, which may be used to compared with a threshold, provides a proximity measurement between a pixel and its m most similar neighbors. Later, taking the logarithm of the absolute difference, Dong et al. [15] proposed a statistic ROLD to improve the detection accuracy. Employing a reference image, Yu et al. [38] introduced a rank-ordered relative differences (RORD) which can preserve more image edges than ROAD and ROLD. In [19], Ghanehar et al. used an exponential function to enlarge the absolute difference $d_{i,j}(k,l)$ in (4), and identify noise in a similar way with ROAD scheme.

2.2. Switching filters

Switching filters, which first utilize some detectors to identify noisy pixels then use some filters to remove the noisy pixels, are widely used to address the IN removal problems. One commonly used filter is the median-type filter. Chen et al. [11] presented the adaptive center weighted median (ACWM) filter, in which the median value was used to verdict if the center pixel is noisy, then the noisy pixels were suppressed by the center weighted median filter. Such strategy is further improved by the new ACWM in [26] and the adaptive weighted mean filter (AWMF) in [41]. In [19], a contrast enhancement-based filter (CEF) is presented, where the absolute differences are first enlarged by an exponential function and then summed to identify noisy pixels. The noisy pixels were further filtered by the weighted median filter. Tsai et al. [34] employed ten existing IN detectors to construct a two-level tree for noisy pixels detection, and the noisy pixels were restored by a median-type filter associated with the support vector regression method. Recent years, some mean filters were also incorporated into the switching scheme for IN removal as they can capture more image detailed information. In [18], the statistic ROAD was incorporated into the bilateral filter and a trilateral filter was designed for IN reduction. In [27], a rank-ordered arithmetic mean filter was combined with a detector to remove IN. Lin et al. [25] presented a switching bilateral filter to suppress IN. Recently, united with detectors, the non-local mean (NLM) filer [8] was also extended for IN removal due to its fantastic denoising performance [37,22]. In [37], the weights of NLM were calculated on an initial denoised image, while these in [22] were computed on the noisy image by diminishing the contributions of noisy pixels offered in the similarity measurement calculation. Combining the noise detector into distance learning, Delon et al. [14] proposed a patch based method for IN removal.

Though the techniques mentioned above could suppress most of the IN, the median-type filters cannot preserve the image details well because they just use median-type values to replace the noisy pixels and do not take full use of all the pixels in the local window. In contrast, the mean-type filters have shown much better denoising performance. However, they still do not catch the characteristic of IN well and bring in some artifacts because those mean filters are originally developed for GN removal. All these show that it is very critical to design better filters for IN reduction.

3. Two-phase noise detector

We present an efficient noise detector for detecting noisy pixels in this section. The proposed detector consists of two parts, the first part is a noise candidates detection phase based on the ROD-ROAD scheme. The second part is an edge pixels identification phase, which is used to identify edge pixels from the noise candidates. Fig. 1 shows the flow chart of the two-phase detector.

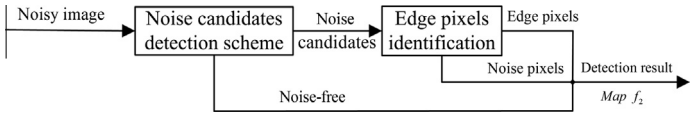

Fig. 1. Details of the two-phase detection mechanism.

3.1. Motivation of ROD-ROAD

The above two mentioned detection schemes in Section 2.1 both have their own deficiencies. For example, in the first scheme, the reference variable $\Omega_{i,j}$ with noise information may cause extremely inaccurate detection results. The ROAD in (5) and its variants, typical representative of the second detection scheme, will be failed in different image patterns. This will be demonstrated in the following example.

From the definition in Eq. (5), one can see that each ROAD value is associated with a pixel. The basic assumption of ROAD scheme is that larger ROAD values indicate noisy pixels, while smaller ones suggest clean pixels. The noisy pixels are detected by comparing their ROAD values with a pre-defined threshold. Nevertheless, the ROAD values are easily affected by different patterns in images as shown in Fig. 2. In this figure, two 5×5 patches A and B containing different patterns are extracted from the 'lena' image with 5% IN. Patch A is chosen from a smooth area, while patch B is selected from a detailed region containing an edge. The noisy pixels in both patches are marked out by ellipses. Fig. 3 shows the corresponding ROAD values for these two patches. The threshold T for the ROAD values of the two patches can be analyzed from these three cases shown in Fig. 4. It can been seen that, because the ROAD values of noisy pixels in patch A are approach to those of some clean pixels in patch B, there exists no such an appropriate threshold that can distinguish the noisy pixels and clean pixels correctly in these two patches simultaneously. The ROD-ROAD is hence presented to address such concern.

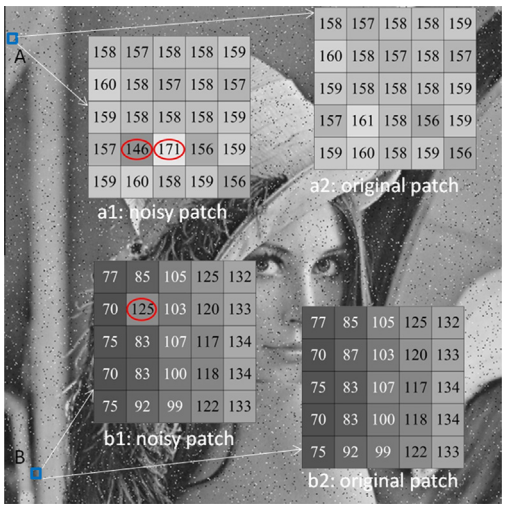

Fig. 2. Two patches in lena image with 5% IN. (a1) noisy data of patch A with the noisy pixels marked by the ellipses, (a2) original data of patch A, (b1) noisy data of patch B with the noisy pixel marked by the ellipse, (b2) original data of patch B.

6	12	4	5	8
18	5	9	6	12
13	4	4	4	6
14	132	142	19	8
9	16	4	6	19

(a)

110	147	158	85	96
98	204	156	109	94
106	131	143	125	89
95	132	144	123	86
112	119	124	136	102

(b)

Fig. 3. ROAD values of patch A and B in Fig. 2: (a) ROAD values of patch A, (b) ROAD values of patch B.

3.2. ROD-ROAD scheme

The basic assumption of ROAD is it assumes that the ROAD values of noisy pixels always larger than those of clean ones. However, this is not quite right because pixels in edges usually also have large ROAD values (see Figs. 2 and 3), leading to inaccurate detection results. Based on the observation that, though the ROAD values will vary a lot for image patches with different patterns (e.g., edges, textures, or smooth areas), the deviations of them do not have such large changes. This is because pixels in one patch belong to the same pattern, then the effects of different patterns can be eliminated by calculating the deviations of ROAD values for each patch, leading to more accurate detection results. Inspired by this, we present a novel criterion called Rank-Ordered Differences of ROAD (ROD-ROAD) which enforces to detect the possible noisy pixels based on the deviation of ROAD values. The ROD-ROAD scheme is described as: First, compute the ROAD value for each pixel in the sliding window, and sort them in ascending order. Then each ROAD value is subtracted by its next adjacent element, and the ROD-ROAD is obtained. Finally, a threshold T is utilized to distinguish the noise-like and clean-like pixels.

Let us still take patches A and B in Fig. 2 for example. Operations are described as, (a) Sort the ROAD values of two patches in Fig. 3, patch A: $4,4,4,4,4,5,5,6,6,6,6,8,8,9,9,12,12,13,14,16,18,19,19,132,142$; and patch B: $85,86,89,94,95,96,98,102,106,109,110,112,119,123,124,125,131,132,136,143,144,147,156,158,204$. (b) Calculate the difference between two adjacent elements, $D_A = (0,0,0,0,1,0,1,0,0,0,2,0,1,0,3,0,1,1,2,2,1,0,113,10)$; and $D_B = (1,3,5,1,1,2,4,4,3,1,2,7,4,1,1,6,1,4,7,1,3,9,2,46)$. (c) Choose a threshold T (for example $T = 12$), and find the first element in D_A and D_B which is larger than this threshold. This ROD-ROAD value is generated by two adjacent ROAD values, and the latter indicates a noisy pixel. Moreover, the ROAD values larger than this found one are also treated to be corresponding to noisy pixels. For example, in patch A, 113 produced by 19 and 132 is the first value larger than the threshold, so 132 associates a noisy pixel (i.e., 146). Moreover, the next one 142 is also considered to be corresponding to a noisy pixel. Therefore, in patch A, 146 and 171 are identified as noisy pixels, the rest are identified as clean-like ones. Similarly, in patch B, using the same threshold, 125 is detected as noisy pixels, and the rest are clean-like ones. By using the ROD-ROAD scheme, the detection results are more accurate.

3.3. Noise candidates detection

The ROD-ROAD scheme can detect noise, nevertheless, its detection performance is till fragile, since it is just a simple improvement of ROAD. Actually, the differences between them is that the thresholds in the ROD-ROAD is adaptively chosen (i.e., based on the deviation) while these in the ROAD is manually tuned. Due to the poor performance of ROAD, the ROD-ROAD performance also needs to be improved. (This analysis is further verified by the experiments in subSection 6.3). To address such concern, instead of directly using the ROD-ROAD scheme to detect noise, we combine it with the absolute deviation scheme in Eq. (2) to detect IN.

As previously analyzed, the reference variable $\Omega_{i,j}$ in Eq. (2) greatly affects the detection results. A clean $\Omega_{i,j}$ leads to good results, while a noisy $\Omega_{i,j}$ causes bad detection results. With this in mind, we first employ the ROD-ROAD scheme to select out the clean-like pixels, then a more cleaner reference variable named MSP (median of selected pixels) is calculated for Eq. (2). This leads to more accurate detection results. Let S_{ij} be a set of pixel coordinates within a $(2N+1)\times(2N+1)$ sliding window, centered at point (i,j). The new noise candidates detection mechanism is described as,

1. Compute the ROAD map R via Eq. (5) [18] for the noisy image by using a 5×5 window and $m = 8$. For each (i,j)-th pixel $x_{i,j}$ in the noisy image, initialize $N = 1$.
2. Set a $(2N+1)\times(2N+1)$ sliding window centering at $x_{i,j}$. Let $R_{i,j} = \{r_{k,l}|(k,l)\in S_{i,j}\}$ be the set that contains the ROAD values corresponding to all the pixels in the window.
3. Sort $R_{i,j}$ in ascending order, denoted as $R^t_{i,j}=\left\{r^t_1, r^t_2, \ldots, r^t_{(2N+1)^2}\right\}$, where r^t_i is the sorted value of $r_{k,l}$. Suppose that $X_l = \left\{\widetilde{x}_i|ROAD(\widetilde{x}_i) = r^t_i, (1\leqslant i\leqslant(2N+1)^2)\right\}$ are the corresponding pixel values.
4. ROD-ROAD scheme: calculate the absolute difference between the two adjacent elements in $R^t_{i,j}$, and a set $D = \left\{d_i|d_i = r^t_{i+1} - r^t_i, i = 1,2,\ldots,(2N+1)^2-1\right\}$ containing all the difference values is obtained. Find a d_k from D such that $d_j\leqslant T$ for $j<k$, and $d_k > T$. Then $R^{t,c}_{i,j} = \{r^t_1, r^t_2, \ldots, r^t_{k+1}\}$ is the set that contains the $k+1$ minimum ROAD values, where T is a threshold, chosen as the mean value of D. If $card(R^{t,c}_{i,j}) < 5$, then $N = N+1$, and go to step 2.
5. These pixels with ROAD values corresponding to $R^{t,c}_{i,j}$ are selected as clean-like pixels, denoted by $X^c_l = \left\{\widetilde{x}_i|ROAD(\widetilde{x}_i) = r^t_i, r^t_i\in R^{t,c}_{i,j}\right\}$.
6. The weighted median value of X^c_l is calculated, $x_{msp} = \text{median}\{w_1\Diamond\widetilde{x}_1, w_2\Diamond\widetilde{x}_2, \ldots, w_{k+1}\Diamond\widetilde{x}_{k+1}\}$, where $w\Diamond\widetilde{x}$ means repeat the pixel $\widetilde{x}$ w times. For the (k,l)-th pixel in the sliding window, the weight is defined as,
$$w(k,l) = \begin{cases} 2; & \min(|k-i|,|l-j|)\leqslant 1\wedge x_{k,l}\in X^c_l \\ 1; & \min(|k-i|,|l-j|) = 2\wedge x_{k,l}\in X^c_l \end{cases} \tag{6}$$
7. Compute the absolute deviation between the pixel $x_{i,j}$ and x_{msp}, e.g., $d^{ref}_{i,j} = |x_{i,j} - x_{msp}|$.

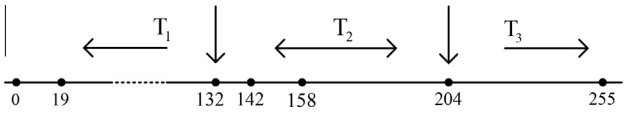

Fig. 4. Illustration of threshold chosen. T_1 cannot correctly identify some clean pixels in patch B; T_2 is not able to accurately detect the noisy pixels in patch A; T_3 cannot identify noisy pixels in both patches.

A fuzzy criterion is proposed to describe the noisy pixels in the following. IN may be more correctly described by fuzzy rules since it is stochastic. The membership function f_1 describes how likely a pixel to be corrupted by IN,

$$f_1(i,j)=\begin{cases}0; & d_{i,j}^{ref}\leqslant T_{min}\\ \left(\frac{d_{i,j}^{ref}-T_{min}}{T_{max}-T_{min}}\right)^{\alpha}; & T_{min}<d_{i,j}^{ref}<T_{max}\\ 1; & d_{i,j}^{ref}\geqslant T_{max}\end{cases} \tag{7}$$

where the threshold T_{max} is larger than T_{min}. The parameter α is set as 0.2 in our method, and by our extension simulations, any $0.1<\alpha<0.5$ can achieve satisfactory results.

3.4. Edge pixels identification

Although our proposed noise candidates detector can significantly reduce the probability of falsely detecting information pixels as noisy ones, some stubborn pixels in edges are still easy to be wrongly judged. Hence, based on the first detection results, this subsection presents an edge pixels detector to pick out edge pixels from noise candidates. Such an edge pixel identification strategy is inspired by the observation that, for an edge pixel not corrupted by IN, the derivative of this pixel on the edge should be small, while the derivative of a noisy pixel be large. It is worth noting that this process is just designed for noisy pixel candidates. The following shows the steps:

1. Scanning the first detection result map f_1. If $f_1(i,j)>0.5$, then set a $(2N+1)\times(2N+1)$ sliding window centered at $x_{i,j}$ in the noisy image X.
2. Calculate these four sets: $s_{i,j}^h=\left\{d_n^h|d_n^h=|x_{i,j}-x_{i,j-N+n}|\right\}, s_{i,j}^v=\left\{d_n^v|d_n^v=|x_{i,j}-x_{i-N+n,j}|\right\}, s_{i,j}^l=\left\{d_n^l|d_{k,l}^l=|x_{i,j}-x_{i-N+n,j-N+n}|\right\}$, and $s_{i,j}^r=\left\{d_n^r|d_n^r=|x_{i,j}-x_{i-N+n,j+N-n}|\right\}$, where $0\leqslant n\leqslant 2N, n\neq N$. These four sets contain the absolute differences between the center pixel and its neighbors in four directions, namely, horizontal, vertical, left diagonal, and right diagonal, representing four edges.
3. To further broaden the relative gap between noisy pixels and other pixels in the four directions. The absolute differences in the four directions are weighted as,

$$S_{i,j}^h=\left\{D_n^h|D_n^h=\frac{w_n^h}{W_N^h}\cdot d_n^h, w_n^h=\left(d_n^h\right)^{\beta}\right\} \tag{8}$$

$$\mathcal{S}_{i,j}^v=\left\{D_n^v|D_n^v=\frac{w_n^v}{W_N^v}\cdot d_n^v, w_n^v=\left(d_n^v\right)^{\beta}\right\} \tag{9}$$

$$\mathcal{S}_{i,j}^l=\left\{D_n^l|D_n^{ld}=\frac{w_n^l}{W_N^l}\cdot d_n^l, w_n^l=\left(d_n^l\right)^{\beta}\right\} \tag{10}$$

$$\mathcal{S}_{i,j}^r=\left\{D_n^r|D_n^r=\frac{w_n^r}{W_N^r}\cdot d_n^r, w_n^r=\left(d_n^r\right)^{\beta}\right\} \tag{11}$$

where $W_N^h=\sum_{n=0,n\neq N}^{2N}w_n^h$, $W_N^v=\sum_{n=0,n\neq N}^{2N}w_n^v$, $W_N^l=\sum_{n=0,n\neq N}^{2N}w_n^l$, and $W_N^r=\sum_{n=0,n\neq N}^{2N}w_n^r$ are normalization factors, and D_n denotes the weighted absolute difference.
4. For each direction, sort the $d_{k,l}$ in ascending order, then m_1 ($m_1=2N+1<2N$) minimum values are summed, defined as minimum orientation difference (MOD), e.g., $\mathrm{MOD}_{i,j}^h=\sum_{i=1}^{m_1}D_{(i)}^h$, $\mathrm{MOD}_{i,j}^v=\sum_{i=1}^{m_1}D_{(i)}^v$, $\mathrm{MOD}_{i,j}^l=\sum_{i=1}^{m_1}D_{(i)}^l$, and $\mathrm{MOD}_{i,j}^r=\sum_{i=1}^{m_1}D_{(i)}^r$, in which $D_{(i)}^h, D_{(i)}^v, D_{(i)}^l$, and $D_{(i)}^r$ are the ith smallest values in the sets $S_{i,j}^h, S_{i,j}^v, S_{i,j}^l$, and $S_{i,j}^r$, respectively.

In the following discussions, the subscript h, v, l, and r are omitted without confusion. We further denote $d_{(n)}, D_{(n)}$ as the n-th smallest elements in $s_{i,j}$ and $S_{i,j}$, respectively. Note that the target of adding weights for each d_n is to broaden the relative gap of two pixels, therefore, it is expected that $\frac{D_(n)}{D_{(n+1)}}<\frac{d_{(n)}}{d_{(n+1)}}$, which leads to $\beta>0$. On the other hand, the intuitive way to set

the weights is based on their contributions to the summation; that is $w_n = d_n$ and $D_n = \frac{(d_n)^2}{\sum d_n}$. However, due to the presence of IN, the minimum value $d_{(1)}$ in $S_{i,j}$ may be far less than the maximum value $d_{(2N)}$; therefore, the minimum weight $w_{(1)} = \frac{d_{(1)}}{\sum_n d_n}$ may close to 0, which is not required. To avoid such problem, we expect that the relative difference between $D_{(n)}$ and $D_{(n+1)}$ with weight β is not smaller than the relative difference between $\frac{(d_{(n)})^2}{\sum_n d_n}$ and $\frac{(d_{(n+1)})^2}{\sum_n d_n}$. That is $\frac{D_{(n)}}{D_{(n+1)}} \geqslant \frac{(d_{(n)})^2}{\sum_n d_{(n)}} / \frac{(d_{(n+1)})^2}{\sum_n d_{(n)}}$, which leading to $\beta \leq 1$.

According to the above discussion, we observe that $0 < \beta \leq 1$ is suitable for (8)–(11). To further determine the β, we implement the proposed Algorithm 1 (see Section 5) to process the noisy images (the six tested images in Fig. 5 with 40% IN), and the average PSNR value of the denoised results is plotted as a function of β. The results are shown in Fig. 6. From this figure, one can see that the average PSNR value first increases then decreases along with β increasing, and the largest value reaches when $\beta = 0.2$. Therefore, we choose $\beta = 0.2$ for our method. Note that though $\beta = 0.2$ achieves the best result, the gap between the largest PSNR value and the smallest one is so small, which indicates that any $\beta \in (0, 1]$ can achieve satisfactory results.

Finally, we define the minimum edge pixels difference as $\mathrm{MEPD}_{i,j} = \min\left\{\mathrm{MOD}^h_{i,j}, \mathrm{MOD}^v_{i,j}, \mathrm{MOD}^l_{i,j}, \mathrm{MOD}^r_{i,j}\right\}$. In the following, we will discuss why this local statistic can be used to pick out the edge pixels from noise candidates. The pixel $x_{i,j}$ in the noise candidates can be analyzed according to the following four cases,

I. When $x_{i,j}$ is an edge pixel, and suppose its neighbors are all clean. Though for some directions, the $\mathrm{MOD}_{i,j}$ may be large, the intensities of neighbor pixels in the same edge are very close to that of the center pixel, so $\mathrm{MEPD}_{i,j}$ value for this current pixel is small.
II. When $x_{i,j}$ is noise, and suppose its neighbors are all clean. Because $\mathrm{MOD}_{i,j}$ in the four direction indexes are large, then $\mathrm{MEPD}_{i,j}$ is large.
III. When $x_{i,j}$ is an edge pixel with some neighbor pixels corrupted by IN. Because in the same edge, some neighbors are not contaminated by noise, $\mathrm{MEPD}_{i,j}$ is still small.
IV. When $x_{i,j}$ is noise, and suppose some of its neighbors are corrupted by IN too, clearly, $\mathrm{MEPD}_{i,j}$ is large.

It is worth noting that, in case IV, the MEPD may be small if one or more pixels in the same edge are corrupted by the similar noise values with those of the center pixel. However, such case may be happened with a very small probability since the RVIN can choose any value in [0,255]. In general, for a grayscale image, it is not noticeable if the absolute difference

Fig. 5. Test images. From left to right: lena, bridge, boat, pepper, house, and pentagon.

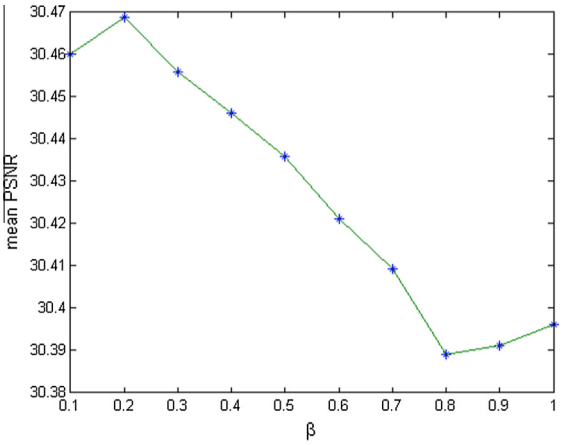

Fig. 6. The mean PSNR values associated with different β values.

between a pixel value and that of its neighbors is less than 8 [6,15]. More specifically, the probability for one pixel corrupted with the similar noise value of the center pixel is $\frac{16}{256^2} < 0.05$, which is a small probability event and can be ignored. According to the definition of MEPD, it is obvious that we take full advantage of the information in the four directions no matter they are corrupted or not. These directions can be treated as potential edge directions in images. The statistic MEPD is still efficient when some of the neighbor pixels are corrupted.

From the above analysis, it can be find that the clean edge pixels have small MEPD values while the noisy pixels have large ones. Therefore, by employing a threshold T, the noise candidates of the first detection results are reclassified, and the edge pixels are identified. Hence, the membership function of these noise candidates is updated as,

$$f_2(i,j) = \begin{cases} f_1(i,j); & \text{MEPD}_{i,j} > T \\ 0; & \text{MEPD}_{i,j} \leqslant T \end{cases} \tag{12}$$

To calculate the statistic MEPD, it is necessary to determine the values of window size N and threshold T. Here, we choose $N = 2$, which means a 5×5 window, and $T = 5$ for simplicity.

4. New weighted mean filter

After detection, the next problem is how to choose an appropriate filter to remove these detected noisy pixels. Instead of using the existing median or mean filters [26,37], in this section we design a more robust weighted mean filter (WMF) for image denoising. The weight in the proposed WMF contains three components which take into account both the image features and IN characteristics. Hence it is more suitable for IN removal.

Let $S_{i,j}$ be a set of pixel coordinates within a $(2N+1) \times (2N+1)$ sliding window, centered at the (i,j)-th pixel. Suppose $x_{i,j}$ is the (i,j)-th pixel in the noisy image, and $\widehat{x}_{i,j}$ is the output of the filter, then

$$\widehat{x}_{i,j} = \frac{\sum_{k,l \in S_{i,j}} W^d_{k,l} W^c_{k,l} W^s_{k,l} x_{k,l}}{\sum_{k,l \in S_{i,j}} W^d_{k,l} W^c_{k,l} W^s_{k,l}} \tag{13}$$

where $W^d_{k,l}$ is the distance weight inverse to the spatial distance between the neighbor pixel (i.e., $x_{k,l}$) and the center one (i.e., $x_{i,j}$). It is expected that the larger the distance between $x_{k,l}$ and $x_{i,j}$ is, the smaller the distance weight should be, and vice versa. Here, $W^d_{k,l}$ is simply defined as the inverse function of the spatial distance,

$$W^d_{k,l} = \frac{1}{(k-i)^2 + (l-j)^2} \tag{14}$$

$W^c_{k,l}$ is called clean-like weight, which means that if a pixel is more likely to be a clean pixel, then it should be matched with a higher weight. On the contrast, if a pixel has a larger probability to be noise, the clean-like weight for it should be lower. Extremely, the noise free pixel has the largest weight, and the completely noisy pixel (the pixel with f_2 value equals to 1) has no weight. Therefore, the clean-like weight is defined as,

$$W^c_{k,l} = e^{|f_2(k,l)-1|} - 1 \tag{15}$$

and $W^s_{k,l}$ is the median-similarity weight, which indicates that if the luminance intensity of $x_{k,l}$ is closer to the median value of the relative-clean pixels (pixels with membership function $f_2 < 1$), the median-similarity weight for $x_{k,j}$ should be higher. The mathematical formula is defined as follows,

$$W^s_{k,l} = e^{-\left(\frac{|x_{k,l} - x^*_m|}{d_{max}}\right)^2} \tag{16}$$

in which $x_{k,l}((k,l) \in S_{i,j}/(i,j))$ is the neighbor pixel of $x_{i,j}$ in the sliding window, x^*_m is the median value of the neighbor pixels excluded the noisy ones, i.e., $x^*_m = median\left\{X^*_{i,j}\right\}$. $X^*_{i,j}$ denotes the set containing the relative clean pixels in the window,

$$X^*_{i,j} = \left\{x_{k,l} | (k,l) \in S_{i,j}, 0 \leqslant f_2(k,l) < 1\right\} \tag{17}$$

and $d_{max} = \max\left\{| x_{k,l} - x^*_m | \,|(k,l) \in S_{i,j}/(i,j)\right\}$ is the maximum difference.

The idea for designing such three components for the weight is quite simple. Firstly, it is well known that, in natural images, the closer the distance of two pixels is, the closer the relationship will be. Hence setting larger weights for these pixels that near to the current pixel is reasonable. Secondly, it is sensible just using these information pixels to filtering the noisy ones. Therefore, in the proposed weighted mean filter, the clean-like weight W^c is inversely proportional to the membership function f_2. Finally, since median value is a good estimator for the IN, it is advisable to design large weights for these pixels whose luminance intensities are approach to the median value of the relative-clean ones.

5. Proposed denoising algorithm

Combining the two phase noise detector described in Section 3 with the proposed weighted mean filter in Section 4, we summarize our denoising method in Algorithm 1.

Algorithm 1. The proposed denoising algorithm for IN removal

Input: The noisy image X, with size of $K_1 \times K_2$; Maximum window size N_{max}; Thresholds T_{min} and T_{max} in (7).
1: Iterative on k until the stop criterion is met.
2: Compute the noise detection map f_2 for image X based on the two-phase detection mechanism.
Outer loop: for $i = 1, \cdots K_1$
Inner loop: for $j = 1, \cdots K_2$
1. Initialize the window size $N = 1$.
2. Locate a sliding window of size $(2N+1) \times (2N+1)$ centered at $x_{i,j}$.
3. If $|X^*_{i,j}| < 3$ ($X^*_{i,j}$ is defined in (17), $|\cdot|$ denotes the cardinality operator), and $N < N_{max}$, then $N \leftarrow N+1$; Return to step (2).
4. Calculate $\widehat{x}_{i,j}$ using (13).
5. The estimator of $x_{i,j}$ is computed as, $y_{i,j} = \widehat{x}_{i,j} \cdot f_2(i,j) + x_{i,j} \cdot (1 - f_2(i,j))$.
6. If $k > k_{max}$, stop; Else $X \leftarrow Y$, the next iteration.
Output: The restored image Y

Suppose $\hat{X}$ is the estimation of the noisy image X by the weighted mean filter combined with the two-phase noise detector. We then add back part of the noisy image into the estimation to form the recovered image, which can avoid over filtering and preserve more image details. Hence, the final recovered image is,

$$Y = \hat{X} \otimes f_2 + X \otimes (1 - f_2) \tag{18}$$

where f_2 is the weight matrix generated by the two-phase detector, and $\otimes$ denotes the Hadamard product.

In the proposed denoising algorithm, we adopt the iterative way that many methods used to improve the quality of outputs. That is, the input of current iteration is the output of the last iteration.

6. Experimental results

In this section, to assess the noise removal capability of the proposed algorithm, we compare the noise detection results and denoising results with several existing methods. Many standard test images are first corrupted by RVIN with various noise densities. Then they are filtered by all the tested algorithms. Fig. 5 shows some examples of tested images.

6.1. Implementations and parameters setting

We implement the proposed algorithm based on Algorithm 1 for RVIN removal. Before implementation, three parameters should be predefined: two thresholds T_{min} and T_{max}, and the maximum window size N_{max}.

We adopt the experimental methods to set values for the two thresholds. Fig. 8 shows the denoised results of 'lena' image with 40% IN in PSNR for different thresholds. As shown in Fig. 8, the two thresholds chosen as $3 \leqslant T_{min} \leqslant 12$ and $35 \leqslant T_{max} \leqslant 75$ can generate satisfied restoration results, which indicates that the proposed method is very robust to these two thresholds. In this paper, we set $T_{min} = 3$ and $T_{max} = 55$ for all the tested images.

For the maximum window size, N_{max} should not to be set too small; otherwise, the number of clean pixels in the sliding window will be not enough to reconstruct the centering noisy pixel. On the other hand, N_{max} should be not too large, so as to preserve the local information of images. In our simulations, N_{max} is set as 3, which indicates a 7×7 window.

The stopping criterion is sill an elusive problem. We choose to terminate our method when the residual error of two iterations is smaller than a threshold,

$$\frac{\|Y_{pre} - Y_{cur}\|_F}{\|Y_{pre}\|_F} < 0.008 \tag{19}$$

where Y_{pre} and Y_{cur} represent the denoised images of previous and current iterations, respectively. $\|\cdot\|_F$ denotes the Frobenius norm. Our method is also terminated when the maximum iteration number is reached. The maximum iterations for 40%, 50%, and 60% noise densities are 3, 3, and 6, respectively.

6.2. Demonstration of edge pixels identification

To demonstrate that our two phase detector has the capability of identifying edge pixels in the noisy environment, Fig. 7 shows the detection results of 'lena' and 'house' images corrupted by 20% RVIN, where (a) mean the first detection mapping

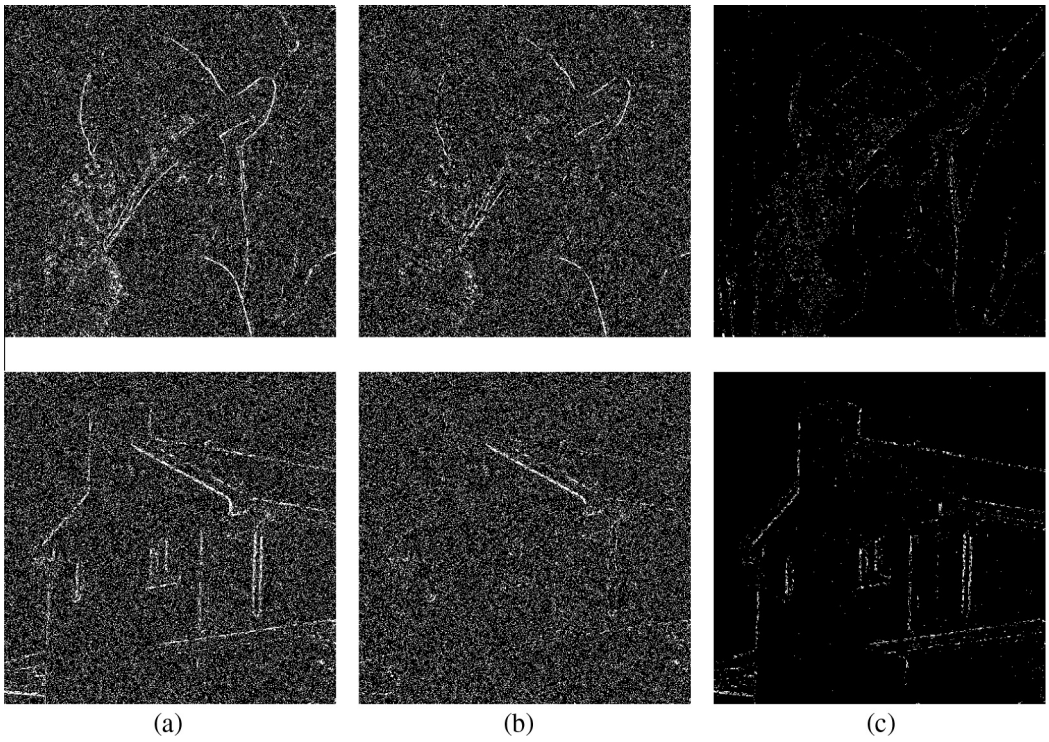

Fig. 7. Noise candidates detection results: (a) noise candidates after the first detection, (b) noise candidates after the second detection, (c) edge pixels identified by the second detection (the difference between (a) and (b)). Zoom into pdf file for a detailed view.

matrices (f_1), (b) denote the second detection results (mapping matrices f_2), and (c) display the difference between these two mapping matrices. In (a) and (b), white dots mean corresponding pixels in the noisy image are detected as noise, while black dots denote corresponding pixels are recognized as clean ones. In (c), bright dots indicate corresponding pixels are identified as edges based on the first detection results.

From Fig. 7(a), it can be easily observed that in the first detection results, there are still many edge pixels which are false detected as noise. On the contrast, fewer edge pixels exist in the Fig. 7(b) though there are still some. This is because that the values of RVIN pixels always do not have a large difference from those of their neighbors, it is impossible for a noise detector to detect all the noisy pixels accurately without any false detection. Fig. 7(c) contains most of the edges but few noisy pixels, which means that, after the second detection, most of the edge pixels are identified from noise candidates.

6.3. Comparison of noise detection

For IN removal, the capability of noise detector affects the recovered results significantly. In this subsection, we compare our detection method with several state-of-the-art RVIN detection methods, such as ACWM [11], Luo-iterative [29], CEF [19], ASWM [3], DWM [16], SD-OOD [5], ROR-NLM[1] [37], and ROLD-EPR[2] [15].

One may realize that it is meaningless to compare the detection capability of our method with other methods, since our proposed method detects the noise in a fuzzy rule which none of the above mentioned methods adopted. For a fair comparison, we set the T_{min} equal to T_{max} in (7). Note that when $T_{min} = T_{max}$, the proposed fuzzy detection method returns to the explicit rule which can be compared with the above mentioned detection methods. In our simulations, we apply our detection method iteratively to improve the detection accuracy. By our extension experiments, two iterations is enough. We also borrow the idea from [16] that the thresholds decrease with iterations to keep a good tradeoff between the undetected and wrongly detected pixels. Therefore, the values for the two thresholds in the first iteration is chosen as 13, and 6 in the second iteration. Also for a fair comparison, we implement others' methods with the optimal parameters and iterations as the original papers suggested to obtain the best detection results. To verify the superiority of the first detection phase (the ROD-ROAD combined with the absolute deviation scheme) to the ROD-ROAD scheme. We choose to use the single ROD-ROAD scheme to detect the noisy pixels and named it as "Proposed 0".

[1] The matlab codes are provided by one of the authors, Dr. Bo Xiong.
[2] The source codes are provided by one of the authors, Prof. Yiqiu Dong.

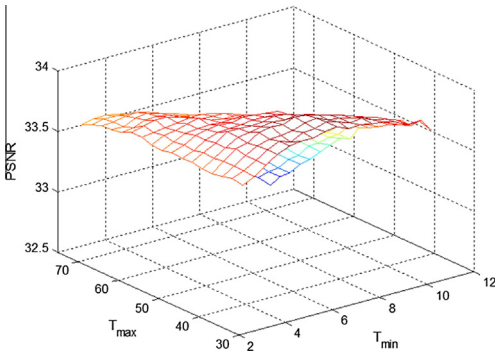

Fig. 8. Restoration results of 'lena' image with 40% random value impulse noise generated by different parameter values.

Table 1
Comparison of noise detection results for 'lena' image corrupted by RVIN.

Method	40%			50%			60%		
	Miss	False-hit	Total	Miss	False-hit	Total	Miss	False-hit	Total
ACWM [11]	14,249	1928	16,177	20,596	3602	24,198	31,165	6668	37,833
Luo's [29]	14,365	1713	16,078	20,236	2135	22,371	33,374	2886	36,260
CEF [19]	14,727	6141	20,868	17,490	7745	25,235	21,314	8657	29,971
ASWM [3]	7381	11,042	18,423	10,614	12,050	22,664	19,577	16,845	36,422
DWM [16]	11,600	7937	19,537	15,035	8652	23,687	15,373	14,215	29,588
SD-OOD [5]	13,299	10,326	23,625	11,741	15,588	27,329	16,993	18,243	35,236
ROR-NLM [37]	12,443	3056	**15,499**	15,778	3655	**19,433**	21,601	5917	27,518
ROLD-EPR [15]	13,987	7471	21,458	16,331	7875	24,206	17,245	9223	26,468
Proposed 0[a]	11,693	9303	20,996	14,211	9661	23,872	19,843	10,117	29,960
Proposed 1[a]	8447	8389	16,836	9115	11,302	20,417	12,133	11,918	**24,051**
Proposed 2[a]	10,138	5203	**15,341**	11,584	6946	**18,530**	15,451	7455	**22,906**

[a] "Proposed 0" denotes the the detect results are produced by the single ROD-ROAD scheme, "Proposed 1" means the results are detected by our method just using the proposed first detection phase (i.e., the ROD-ROAD combined with the absolute deviation scheme), while "Proposed 2" denotes the results are produced by both the two detection phases.

Table 2
Comparison of restoration results in PSNR for images corrupted by RVIN.

Method	ACWM [11]	Luo's [29]	CEF [19]	ASWM [3]	DWM [16]	Trilateral [18]	SD-OOD [5]	SBF [25]	ROR-NLM [37]	ROLD-EPR [15]	Proposed
Lena											
40%	29.5842	30.7707	32.1148	32.2911	32.3415	32.0734	29.5985	30.0405	32.9657	32.7150	**33.7277**
50%	24.6324	27.1577	29.7564	29.2315	29.3169	30.2398	27.8896	27.1795	30.0189	31.1240	**31.7281**
60%	20.3994	22.6151	25.9032	25.0360	25.4869	27.4184	26.5063	23.4089	25.5969	28.9796	**29.6646**
Bridge											
40%	23.5194	23.5930	23.8539	23.9678	24.0687	23.7333	22.6622	23.0450	24.1837	24.5132	**24.7997**
50%	21.4062	21.6216	22.7932	22.5755	22.5717	23.0881	21.7140	21.9910	22.8426	23.5096	**23.7190**
60%	19.1227	19.1737	21.4059	21.1075	21.1289	21.8844	21.1949	20.4935	21.1945	22.5179	**22.7028**
Boat											
40%	26.7405	26.8804	27.0804	27.2710	27.0605	26.7729	24.6639	25.9143	27.5882	27.0475	**27.6631**
50%	24.5015	24.8246	25.6531	25.5612	25.4535	25.7874	23.8144	24.6599	25.8695	25.9592	**26.3533**
60%	21.4506	21.6170	24.1626	23.7997	23.8641	24.4383	23.3654	22.8370	24.1741	25.0065	**25.1785**
Pepper											
40%	28.8938	29.5329	30.4169	30.2760	30.6171	30.1142	29.1236	29.1679	31.2425	31.1144	**31.5458**
50%	25.4002	26.7129	28.7790	28.3882	28.5211	29.3005	27.5441	27.0299	28.9384	29.9015	**30.3631**
60%	21.5320	23.5268	26.2016	25.3548	25.5886	27.1517	26.0899	24.0161	25.8514	28.1943	**28.7147**
House											
40%	31.9587	33.4320	35.2325	36.0844	36.2694	35.3648	33.9365	33.6728	36.4996	36.5677	**37.3145**
50%	27.2731	29.7165	33.3614	34.6239	33.2377	33.1997	31.2151	29.9722	33.3515	34.4370	**35.3826**
60%	22.4235	24.2715	28.2534	28.0607	28.5279	29.8992	29.0579	25.7338	28.7264	32.0245	**32.8893**
Pentagon											
40%	27.0890	26.9968	27.1645	27.2907	27.2324	26.6126	25.7515	26.3351	27.6785	27.5863	**27.8970**
50%	25.4749	25.3330	26.2436	26.1936	26.0746	25.9229	25.0935	25.6544	26.5610	26.6542	**26.9565**
60%	23.4055	22.7786	25.1150	24.9757	25.0295	24.8232	24.6009	24.6261	25.3569	25.6124	**25.9037**

Table 3
Comparison of restoration results in SSIM and FSIM for images corrupted by RVIN.

Method	Lena			Bridge			Boat			Pepper			House			Pentagon		
	40%	50%	60%	40%	50%	60%	40%	50%	60%	40%	50%	60%	40%	50%	60%	40%	50%	60%
SSIM																		
ACWM [11]	0.8771	0.7648	0.5835	0.7270	0.6262	0.4824	0.8089	0.7062	0.5691	0.8126	0.7069	0.5484	0.9098	0.7975	0.5938	0.7837	0.7057	0.5898
Luo's [29]	0.8807	0.7957	0.0.6302	0.7388	0.6388	0.5005	0.8025	0.7189	0.5675	0.8137	0.7227	0.6113	0.9095	0.8359	0.6637	0.7829	0.7047	0.5812
CEF [19]	0.9162	0.8850	0.7634	0.7162	0.6419	0.5616	0.8223	0.7658	0.6849	0.8497	0.8101	0.7304	0.9380	0.9209	0.7991	0.7801	0.7202	0.6572
ASWM [3]	0.9319	0.8823	0.7913	0.6953	0.5830	0.4936	0.8056	0.7258	0.6515	0.8134	0.7709	0.7068	0.9676	0.9411	0.8493	0.7640	0.6852	0.6178
DWM [16]	0.9280	0.8905	0.8063	0.6961	0.5851	0.4981	0.8186	0.7334	0.6574	0.8505	0.7819	0.7126	0.9537	0.9450	0.8761	0.7711	0.6868	0.6227
Trilateral [18]	0.9033	0.8655	0.8059	0.6290	0.5871	0.4915	0.7365	0.7208	0.6348	0.8099	0.7832	0.7150	0.9345	0.9036	0.8565	0.6825	0.6459	0.5697
SD-OOD [5]	0.8822	0.8404	0.7425	0.6058	0.5447	0.4877	0.7289	0.6741	0.6283	0.8432	0.7644	0.7138	0.9188	0.8873	0.7798	0.6880	0.6480	0.5957
SBF [25]	0.8817	0.8119	0.7046	0.5903	0.5320	0.4504	0.7310	0.6754	0.5983	0.7887	0.7347	0.6536	0.9199	0.8594	0.7557	0.6813	0.6416	0.5830
ROR-NLM [37]	0.9359	0.8993	0.8102	0.7254	0.6459	0.5462	0.8328	0.7806	0.7083	0.8641	0.8202	0.7452	0.9555	0.9327	0.8725	0.7962	0.7354	0.6725
ROLD-EPR [15]	0.9292	0.9015	0.8529	0.7563	0.6873	0.6083	0.8297	0.7807	0.7213	**0.8671**	0.8271	0.7764	0.9586	0.9359	0.8985	0.7940	0.7435	0.6762
Proposed	**0.9453**	**0.9189**	**0.8785**	**0.7689**	**0.7050**	**0.6301**	**0.8353**	**0.7913**	**0.7323**	0.8605	**0.8293**	**0.7887**	**0.9661**	**0.9467**	**0.9168**	**0.8031**	**0.7522**	**0.6916**
FSIM																		
ACWM [11]	0.9694	0.9167	0.9029	0.9543	0.9004	0.8898	0.9673	0.9107	0.9050	0.9499	0.9053	0.8648	0.9330	0.8873	0.8379	0.9506	0.9240	0.8821
Luo's [29]	0.9790	0.9542	0.8882	0.9654	0.9329	0.8664	0.9814	0.9552	0.8715	0.9469	0.9252	0.8657	0.9386	0.9095	0.8545	0.9451	0.9318	0.8924
CEF [19]	0.9818	0.9674	0.9224	0.9710	0.9545	0.9150	0.9848	0.9738	0.9204	0.9451	0.9273	0.8936	0.9353	0.9059	0.8723	0.9493	0.9285	0.9034
ASWM [3]	0.9815	0.9560	0.9029	0.9629	0.9374	0.8898	0.9860	0.9649	0.9051	0.9485	0.9184	0.8648	0.9337	0.8874	0.8379	0.9478	0.9197	0.8821
DWM [16]	0.9813	0.9598	0.9103	0.9702	0.9438	0.8965	0.9856	0.9701	0.9162	0.9500	0.9158	0.8702	0.9286	0.8874	0.8423	0.9449	0.9184	0.8824
Trilateral [18]	0.9710	0.9528	0.9194	0.9689	0.9527	0.9151	0.9692	0.9492	0.9106	0.9436	0.9206	0.8747	0.9253	0.8989	0.8580	0.9318	0.9093	0.8614
SD-OOD [5]	0.9617	0.9500	0.9288	0.9619	0.9479	0.9241	0.9691	0.9559	0.8968	0.9011	0.8859	0.8697	0.8747	0.8561	0.8372	0.9073	0.8951	0.8765
SBF [25]	0.9559	0.9193	0.8608	0.9481	0.9111	0.8522	0.9551	0.9103	0.8310	0.9164	0.8793	0.8239	0.8942	0.8583	0.8107	0.9174	0.8949	0.8564
ROR-NLM [37]	0.9842	0.9678	0.9229	0.9736	0.9530	0.9067	0.9876	0.9745	0.9308	**0.9532**	0.9312	0.8885	0.9364	0.8983	0.8556	0.9504	0.9294	0.8986
ROLD-EPR [15]	0.9824	0.9697	0.9501	0.9756	0.9640	0.9397	0.9872	0.9760	0.9536	0.9525	0.9357	0.9084	0.9436	**0.9211**	0.8938	0.9526	0.9345	0.9091
Proposed	**0.9847**	**0.9759**	**0.9574**	**0.9769**	**0.9661**	**0.9487**	**0.9887**	**0.9814**	**0.9655**	0.9496	**0.9377**	**0.9122**	**0.9446**	0.9209	**0.8959**	**0.9538**	**0.9371**	**0.9115**

Table 1 lists the detection comparison results for the 'lena' image contaminated by RVIN with different densities. The detection results of each method consist of three parts, namely, the number of undetected noisy pixels ("miss" term), the false detected pixels ("false-hit" term), and the total number. A good noise detector should reduce both the number of "miss" and "false-hit" pixels. The two lowest "total" numbers are marked in bold for an intuitive comparison.

From Table 1, one can see that the "Proposed 0" generates more "total" numbers than "proposed 1", which indicates that the detection performance is indeed improved by jointing the ROD-ROAD into the absolute deviation scheme than the single ROD-ROAD scheme. Besides, the miss pixels of proposed 1 is less than those of proposed 2, suggesting that some noisy pixels are also identified as edges by the edge pixels identification stage. This is because that both the noisy and edge pixels exhibit the similar properties, and our second detection phase is conducted in the noisy environment, hence it is inevitable to false identify noisy pixels as edge pixels. However, the "total" numbers generated by proposed 2 are lower than those of proposed 1, which indicates the second detection phase indeed can identify the edge pixels from noise candidates. Though with the cost of missing some noisy pixels, the second edge identification phase is necessary and meaningful.

Though some methods such as ACWM, Luo's, and ROR-NLM generate less "false-hit" than ours, there are too many missed noisy pixels which will lead to the presence of noticeable noisy patches. Actually, for the "total" term, the proposed noise detection method has the lowest "total" numbers among all the methods for various noise densities. Comparing with other methods, our method can identify more noisy pixels with fewer mistakes. In addition, one can also observed that, when the noise density is higher, the superiority of our method becomes more obvious. This means that our method is very robust and still efficient when the noise density becomes high.

6.4. Comparison of image restoration

The denoising performance of our proposed method is accessed by comparing with those of other well known IN removal methods, such as ACWM [11], Luo-iterative [29], CEF [19], ASWM [3], DWM [16], Trilateral [18], SD-OOD [5], SBF [25], ROR-NLM [37], and ROLD-EPR [15]. There are lots of models for quantitative image quality assessment, such as the peak signal to noise ratio (PSNR), structural similarity (SSIM) [36], feature similarity (FSIM) [40], and visual saliency-based index (VSI) [39].

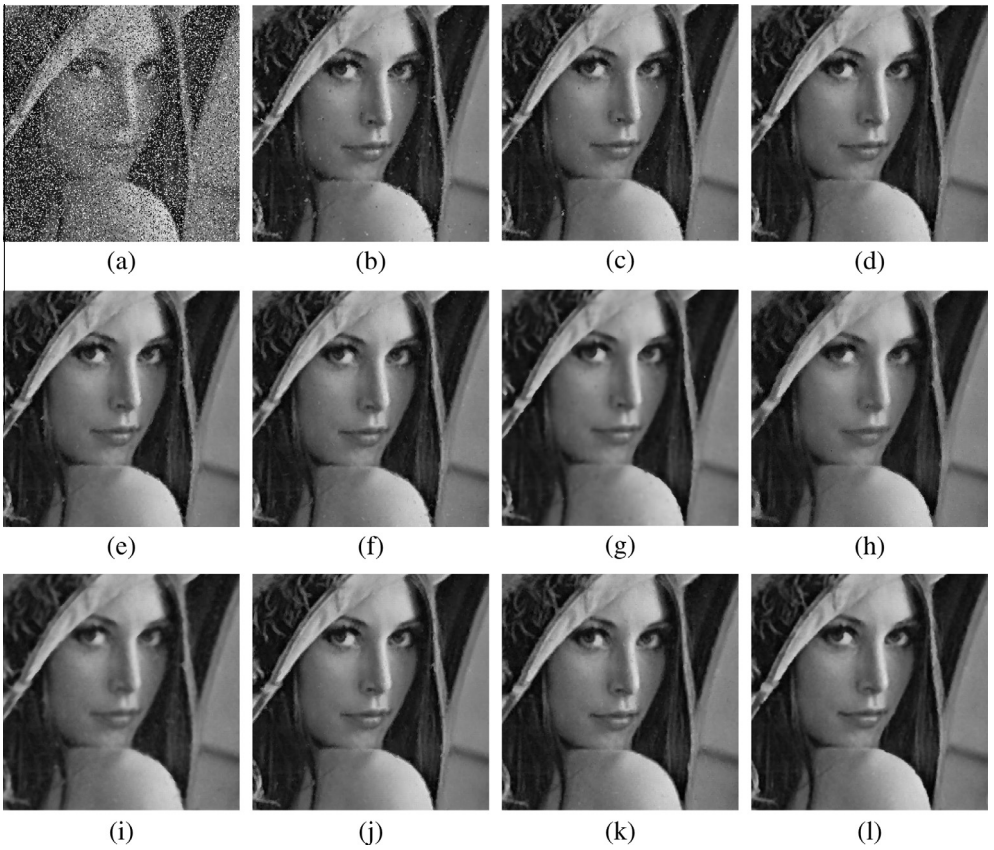

Fig. 9. Results of different algorithms in restoring the test 'lena' image corrupted by RVIN with 40% noise density. (a) Noisy image, (b) ACWM, (c) Luo's method, (d) CEF, (e) ASWM, (f) DWM, (g) ROAD-Trilateral, (h) SD-OOD, (i) SBF, (j) ROR-NLM, (k) ROLD-EPR, (l) Proposed Method. Zoom into pdf file for a detailed view.

Consider that VSI is mainly designed for color image assessment and our experiments are conducted on grayscale images, hence we choose PSNR, SSIM, and FSIM to measure the restored results. Generally speaking, the larger PSNR, SSIM, and FSIM values are, the better quality of the restored image will be.

For a fair comparison, all the other methods are set with the optimal parameters and the optimal iteration numbers suggested by the original papers to achieve the best denoising performance, e.g., four iterations, $[\delta_0, \delta_1, \delta_2, \delta_3] = [40, 25, 10, 5]$, and $s = 0.5$ for ACWM. $N_d = 3, T_d = 14$, and $K_d = 3$ for Luo' method. For CEF, the initial K is set as 5, and for each iteration $K = K + t$, where $t = 10, 15, 20, 25, 30$. $\delta = 0.1, \varepsilon = 0.01$, and iteration number changes form 3 to 10 based on the noise density for ASWM. $T_0 = 510$, iterations = 11 to 18 for DWM. $\sigma_S = 0.5, \sigma_R = 100, \sigma_I = 45$, and $\sigma_J = 0$ for Trilateral filter. A 5×5 window, $T_1 = 0.48$, and $T_2 = 0.24$ for SD-OOD. $\sigma_R = 120, Tk_1 = 25$,and $Tk_2 = 5$ for SBF. For ROR-NLM, $[T_1^c, T_2^c, T_3^c] = [30, 40, 50]$, $[T_1^f, T_2^f, T_3^f, T_4^f] = [7, 10, 20, 25]$, and the iteration number = 3, 4, and 9 for 40%, 50%, and 60% noise density, respectively. The window size is set as $5 \times 5, m = 12$, and the initial threshold $T_e = 5.4$ for ROLD-EPR.

Tables 2 and 3 list the PSNR, SSIM and FSIM values from the methods for all the tested images corrupted by RVIN with noise densities from 40% to 60%, respectively. In both tables, the best values are marked for easy comparisons. From Table 2, it is obvious that for all the tested images and noise densities, our proposed method produces the best results in terms of PSNR. For SSIM and FSIM measurements, as shown in Table 3, our method also generates the best results for almost all the tested images except for the 'pepper' image with 40% noise and 'house' image with 50% noise, for which the SSIM or FSIM scores of our method are slightly less than the best ones.

For subjective comparisons, we show the deoised results of 'lena' and 'pepper' images restored by these tested methods. Fig. 9 shows the output results of 'lena' image corrupted by 40% IN. In order to give a better visual impression, only enlarged portion of the restored images are displayed. To further compare the ability of removing high density noise, the restoration results of 'pepper' image contaminated by 60% IN are shown in Fig. 10. From these two figures, one can find that, other methods except for the ROLD-EPR, not only cannot suppress IN thoroughly, but also bring many artifacts and damage the image details. Although some methods such as Trilateral filter and ROR-NLM can remove most of the noise when noise density is low (i.e., 40%), they are severely failed in the images corrupted by heavy density noise (i.e., 60%). The results produced by ROLD-EPR verify that this method indeed has a good capability of preserving the image edges, however, there still exist some

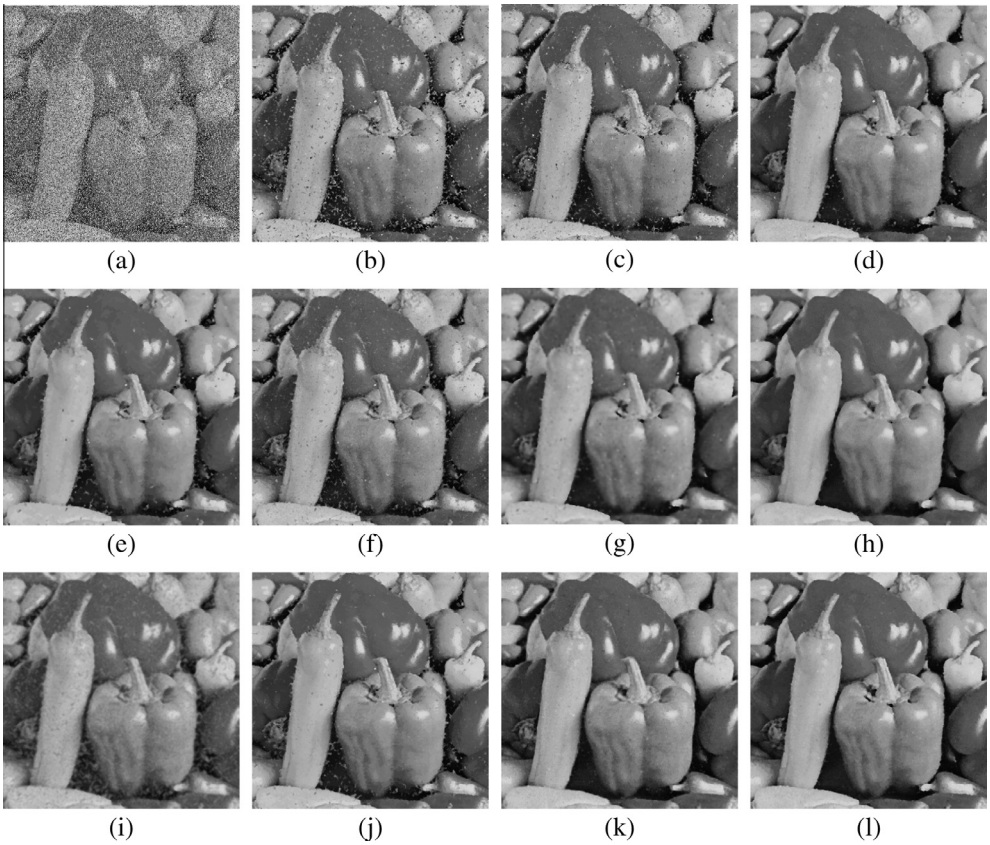

Fig. 10. Results of different algorithms in restoring the test 'pepper' image corrupted by RVIN with 60% noise density. (a) Noisy image, (b) ACWM, (c) Luo's method, (d) CEF, (e) ASWM, (f) DWM, (g) ROAD-Trilateral, (h) SD-OOD, (i) SBF, (j) ROR-NLM, (k) ROLD-EPR, (l) Proposed Method. Zoom into pdf file for a detailed view.

noticeable noise unremoved due to the imperfect detector it used. In contrast, the results generated by our method have very good visual qualities. Even though the noise density is 60%, our method still can remove almost all the IN, and preserve most of the image details.

7. Conclusion

In this paper, we present a new algorithm for removing IN. In summary, a new statistical mechanism ROD-ROAD as well as a fuzzy rule is introduced for describing how likely a pixel to be noisy. Besides, a novel statistic MEPD is proposed to pick out the edge pixels from the noisy image based on the first detection results, which improves the detection accuracy. Finally, a weighted mean filter which takes into account both the information of image features and IN characteristics is designed for IN reduction.

The proposed method uses no complexity optimization algorithm, hence is well-understood and easy to implement. In comparison with previous researches, experimental results demonstrate that our method has a very good capability of suppressing RVIN, even though the images are corrupted by high density noise. With respect to both the visual effects and quantitative measure results, our proposed method outperforms all the compared state-of-the-art denoising methods.

Acknowledgements

The authors would like to thank Prof. Dong and Prof. Chan for supplying the codes of their algorithm in [15], and Dr. Xiong for providing the MATLAB implementation of the method in [37].

This work was supported in part by the Multiyear Research Grants, Macau Science and Technology Development Fund under Grant No.008/2010/A1, and Macau Science and Technology Development Fund under Grant 017/2012/A1.

References

[1] E. Abreu, M. Lightstone, S.K. Mitra, K. Arakawa, A new efficient approach for the removal of impulse noise from highly corrupted images, IEEE Trans. Image Process. 5 (6) (1996) 012–1025.
[2] F. Ahmed, S. Das, Removal of high-density salt-and-pepper noise in images with an iterative adaptive fuzzy filter using alpha-trimmed mean, IEEE Trans. Fuzzy Syst. 22 (5) (2014) 1352–1358.
[3] S. Akkoul, R. Ledee, R. Leconge, R. Harba, A new adaptive switching median filter, IEEE Signal Process. Lett. 17 (6) (2010) 587–590.
[4] G.R. Arce, J.L. Paredes, Recursive weighted median filters admitting negative weights and their optimization, IEEE Trans. Signal Process. 48 (3) (2000) 768–779.
[5] A.S. Awad, Standard deviation for obtaining the optimal direction in the removal of impulse noise, IEEE Signal Process. Lett. 18 (7) (2011) 407–410.
[6] A. Bovik, Handbook of Image and Video Processing, New York Academic, 2000.
[7] D.R.K. Brownrigg, The weighted median filter, ACM Commun. 27 (8) (1984) 807–818.
[8] A. Buades, B. Coll, J.M. Morel, A non-local algorithm for image denoising, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, IEEE, 2005, pp. 60–65.
[9] C.L.P. Chen, H. Li, Y. Wei, T. Xia, Y.Y. Tang, A local contrast method for small infrared target detection, IEEE Trans. Geosci. Remote Sens. 52 (1) (2014) 574–581.
[10] L. Chen, C.L.P. Chen, M. Lu, A multiple-kernel fuzzy c-means algorithm for image segmentation, IEEE Trans. Syst. Man Cybern. B: Cybern. 41 (5) (2011) 1263–1274.
[11] T. Chen, H. Wu, Adaptive impulse detection using center-weighted median filters, IEEE Signal Process. Lett. 8 (1) (2001) 1–3.
[12] T. Chen, H. Wu, Space variant median filters for the restoration of impulse noise corrupted images, IEEE Trans. Circ. Syst. II: Analog Digital Signal Process. 48 (8) (2001) 784–789.
[13] X. Crnojevic, V. Senk, Z. Trpovski, Advanced impulse detection based on pixel-wise MAD, IEEE Signal Process. Lett. 11 (7) (2004) 589–592.
[14] J. Delon, A. Desolneux, A patch-based approach for removing impulse or mixed gaussian-impulse noise, SIAM J. Imaging Sci. 6 (2) (2013) 1140–1174.
[15] Y. Dong, R.H. Chan, S. Xu, A detection statistic for random-valued impulse noise, IEEE Trans. Image Process. 16 (4) (2007) 1112–1120.
[16] Y. Dong, S. Xu, A new directional weighted median filter for removal of random-valued impulse noise, IEEE Signal Process. Lett. 14 (3) (2007) 193–196.
[17] D.A.F. Florencio, R.W. Schafer, Decision-based median filter using local signal statistic, Proc. SPIE 2308 (1994) 268–275.
[18] R. Garnett, T. Huegerich, C. Chui, W. He, A universal noise removal algorithm with an impulse detector, IEEE Trans. Image Process. 14 (11) (2005) 1747–1754.
[19] U. Ghanekar, A.K. Singh, R. Pandey, A contrast enhancement-based filter for removal of random valued impulse noise, IEEE Signal Process. Lett. 17 (1) (2010) 47–50.
[20] R.C. Gonzalez, R.E. Woods, Digital Image Processing, third ed., Pearson Prentice Hall, 2008.
[21] J. Gui, D. Tao, Z. Sun, Y. Luo, X. You, Y.Y. Tang, Group sparse multiview patch alignment framework with view consistency for image classification, IEEE Trans. Image Process. 23 (7) (2014) 3126–3137.
[22] H. Hu, B. Li, Q. Liu, Removing mixture of gaussian and impulse noise by patch-based weighted means, arXiv preprint arXiv:1403.2482, 2014.
[23] S.J. Ko, Y.H. Lee, Center weighted median filters and their applications to image enhancement, IEEE Trans. Circ. Syst. 38 (9) (1991) 984–993.
[24] H. Li, Y. Wei, L. Li, C.L.P. Chen, Hierarchical feature extraction with local neural response for image recognition, IEEE Trans. Cybern. 43 (2) (2013) 412–424.
[25] C.H. Lin, J.S. Tsai, C.T. Chiu, Switching bilateral filter with a texture/noise detector for universal noise removal, IEEE Trans. Image Process. 19 (9) (2010) 2307–2320.
[26] T.C. Lin, A new adaptive center weighted median filter for suppressing impulsive noise in images, Inf. Sci. 177 (4) (2007) 1073–1087.
[27] T.C. Lin, Switching-based filter based on dempsters combination rule for image processing, Inf. Sci. 180 (24) (2010) 4892–4908.
[28] L. Liu, C.L.P. Chen, Y. Zhou, Y.Y. Tang, Impulse noise removal using sparse representation with fuzzy weights, in: 2014 IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 4052–4057.
[29] W. Luo, A new efficient impulse detection algorithm for the removal of impulse noise, IEICE Trans. Fund. Electron. Commun. Comput. Sci. 88 (10) (2005) 2579–2586.
[30] M. Nikolova, A variational approach to remove outliers and impulse noise, J. Math. Imaging Vis. 20 (1–2) (2004) 99–120.
[31] T. Nodes, N.C. Gallagher Jr., Median filters: some modifications and their properties, IEEE Trans. Acoust. Speech Signal Process. 30 (5) (1982) 739–746.
[32] T. Sun, Y. Neuvo, Detail-preserving median based filters in image processing, Pattern Recogn. Lett. 15 (1994) 341–347.

[33] K.K.V. Toh, N.A.M. Isa, Cluster-based adaptive fuzzy switching median filter for universal impulse noise reduction, IEEE Trans. Consum. Electron. 56 (4) (2010) 2560–2568.
[34] H.H. Tsai, B.M. Chang, X.P. Lin, Using decision tree, particle swarm optimization, and support vector regression to design a median-type filter with a 2-level impulse detector for image enhancement, Inf. Sci. 195 (2012) 103–123.
[35] X.Y. Wang, H.Y. Yang, Y. Zhang, Z.K. Fu, Image denoising using SVM classification in nonsubsampled contourlet transform domain, Inf. Sci. 246 (2013) 155–176.
[36] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[37] B. Xiong, Z. Yin, A universal denoising framework with a new impulse detector and nonlocal means, IEEE Trans. Image Process. 21 (4) (2012) 1663–1675.
[38] H. Yu, L. Zhao, H. Wang, An efficient procedure for removing random-valued impulse noise in images, IEEE Signal Process. Lett. 15 (2008) 922–925.
[39] L. Zhang, Y. Shen, H. Li, VSI: a visual saliency-induced index for perceptual image quality assessment, IEEE Trans. Image Process. 23 (10) (2014) 4270–4281.
[40] L. Zhang, L. zhang, X. Mou, D. Zhang, FSIM: a feature similarity index for image quality assessment, IEEE Trans. Image Process. 20 (8) (2011) 2378–2386.
[41] P. Zhang, F. Li, A new adaptive weighted mean filter for removing salt-and-pepper noise, IEEE Signal Process. Lett. 21 (10) (2014) 1280–1283.
[42] Y. Zhang, J. Liu, M. Li, Z. Guo, Joint image denoising using adaptive principal component analysis and self-similarity, Inf. Sci. 259 (2014) 128–141.
[43] Z. Zhou, Cognition and removal of impulse noise with uncertainty, IEEE Trans. Image Process. 21 (7) (2012) 3157–3167.