



# Article An Iterative Weighted-Mean Filter for Removal of High-Density Salt-and-Pepper Noise

Fengyu Chen<sup>1</sup>, Minghui Huang<sup>1,2</sup>, Zhuxi Ma<sup>1</sup>, Yibo Li<sup>1,2,\*</sup> and Qianbin Huang<sup>1,3</sup>

- <sup>1</sup> Light Alloy Research Institute, School of Mechanical and Electrical Engineering, Central South University, Changsha 410083, China; 183812050@csu.edu.cn (F.C.); meeh@mail.csu.edu.cn (M.H.); mazhx@csu.edu.cn (Z.M.); donglei666@csu.edu.cn (Q.H.)
- <sup>2</sup> State Key Laboratory of High-Performance Manufacturing, Central South University, Changsha 410083, China
- <sup>3</sup> Guangxi Liuzhou Yinhai Aluminum Company Limited, Liuzhou 545006, China
- \* Correspondence: liyibo@csu.edu.cn or csuemail@163.com

Received: 26 October 2020; Accepted: 30 November 2020; Published: 2 December 2020



**Abstract:** Salt-and-pepper noise, which is often introduced by sharp and sudden disturbances in the image signal, greatly reduces the quality of images. Great progress has been made for the salt-and-pepper noise removal; however, the problem of image blur and distortion still exists, and the efficiency of denoising requires improvement. This paper proposes an iterative weighted-mean filter (IWMF) algorithm in detecting and removing high-density salt-and-pepper noise. Three steps are required to implement this algorithm: First, the noise value and distribution characteristics were used to identify the noise pixels, effectively improving the accuracy of noise detection. Second, a weighted-mean filter was applied to the noisy pixels. We adopted an un-fixed shape symmetrical window with better detail preservation ability. Third, this method was performed iteratively, avoiding the streak effect and artifacts in high noise density. The experimental results showed that IWMF outperformed other state-of-the-art filters at various noise densities, both in subjective visualization and objective digital measures. The extremely fast execution speed of this method is quite suitable for real-time processing.

Keywords: iterative weighted-mean filter; salt-and-pepper noise; noise detection and removal

# 1. Introduction

Images are often contaminated by salt-and-pepper noise, due to various reasons, including transmitting images in defective channels or taking pictures with a faulty sensor [1]. The salt-and-pepper noise is classified as the impulse noise and comprises a random set of pixels with extreme intensity and usually in significant contrast with the neighboring noise-free pixels [2]. The existence of salt-and-pepper noise, even with a low noise density, can seriously affect visual perception and image analysis, and thus, needs to be detected and removed.

Nonlinear filters, particularly the standard median filter (MF) [3] and the modified median filters, are widely used to remove salt-and-pepper noises. However, the standard MF alters each pixel by a median value of pixels in the predefined window, which destroys many noise-free pixels, and cannot effectively retain the high-frequency components. To improve the denoising performance and efficiency, extended versions of MF have been proposed [2,4–23]. The most commonly used modified MFs are the weighted filter [5,8,18] and the adaptive median filter [14,22].

By giving higher predefined weights to the selected pixels in the filtering window before calculating the median, the weighted filter can retain more details than MF. Through automatically adjusting the filtering window size based on the local noise content, the adaptive median filter can prevent image

over-smoothing and help to keep the image details. The methods mentioned above can improve the denoising performance to some extent, but the destruction of many noise-free pixels cannot be avoided.

A decision-based algorithm (DBA) was proposed in Reference [19] by Srinivasan and Ebenezer, in which the pixel value of 0 or 255 was regarded as the "noisy pixels" and the others were treated as "noise-free pixels". Only "noise pixels" were replaced by the median value. This method avoids the destruction of many noise-free pixels and effectively improves the capability of the image edges and detail preservation. However, when dealing with the images under high noise density (ND > 30%) with DBA, artifacts will be introduced. To overcome this drawback, Esakkirajan et al. [11] proposed an un-symmetry trimmed median method. This replaced noisy pixels with the trimmed median value of all the elements present in the selected window to avoid the interference of noisy pixels on the sorting result. The proposed algorithm improved the effect of salt-and-pepper noise removal in images at high noise densities (ND > 30%); however, it often introduced dark patch-like surface in the reconstructed image, and thus, requires further improvement. Decision-based or adaptive switching hybrid filters were proposed, such as the two-stage filter (TSF) [20], different applied median filter (DAMF) [10], adaptive switching weighted median filter (AFMF) [9], and modified decision based median filter (MDBMF) [17].

The TSF and DAMF are adaptive-switching-trimmed-median filters that replace noisy pixels with the trimmed median value in the adaptive window. Both are adaptive for a wide range of noise density, but too large of an adaptive window is required when processing images with high-density noise. The replacement of the noisy pixels under such a large window causes the images to be blurred. The BPDF is a useful filter below medium-density noise; however, it will create a raindrop effect in the case of high-density noise. With combinations of different weighted-median and switching technologies, the switching-trimmed-weighted-median filters, such as ASWMF, have a good capability for detail preservation, but the judgment and switching processes increase the computational complexity and lead to a long processing time. An effective and efficient filter, with the considerations of high-density noise, avoiding over-smoothing and artifacts, and time-saving is further required.

In addition to median-based filters, great attention has also been paid to improving mean-based filters for denoising [24–30]. The mean-based filters typically incorporate important filtering technologies, which are called the trimmed mean, switching mean, weighted mean, etc. Two typical mean-based filters, namely, efficient restoration method for impulse noise (ERMI) [28] and adaptive Gaussian filter (AGF) [27] were commonly used. The ERMI is a switching-trimmed-mean filter that adjusts the window size automatically according to the noise density, and then uses the mean value of the noise-free pixels in the filtering window as the restored pixel value. This process is simple and has a high processing speed. Inspired by the simplest and high efficiency of the ERMI, a switching-trimmed-weighted-mean filter, called adaptive Gaussian filter (AGF) was further proposed by Mehdi et al. [27], which performs noise removal by using a Gaussian filter with adaptive variances based on the density of noise. This retains the advantages of the ERMI, and improves the edges and details preservation capability. These filters have the advantages of simplicity and good real-time performance, but are still subject to the drawback of low pass filtering, which leads to the loss of many high-frequency components in the image. Therefore, further optimization of the mean-based method is required.

Another kind of hybrid filter, integrating the median-based and mean-based filtering technology, has been proposed, such as the fast switching based median-mean filter (FSMMF) [31], adaptive weighted mean filter (AWMF) [30], and switching median-mean filter (SMMF) [32]. The FSMMF selects the median value or mean value based on the number of noise-free pixels in the window, and the SMMF removes noises at different times and orders of the median filter and mean filter. Both methods show better robustness at high noise densities; however, the denoising effect of these filters depends on the accuracy of noise detection and the rationality of the switching conditions.

From the discussions above, the current filters usually perform well on images with the noise density within a specific range, but some filtering distortion problems, such as streak effects and over-smoothing,

are still common, especially under high noise density conditions. In addition, the discussed filters achieve the denoising effect by increasing the complexity of the algorithm, which inevitably decreases the efficiency and increases the time of the process. In real-time applications, the denoising speed of the filters is required to be largely improved.

To overcome such problems, a robust noise removal method, called iterative weighted mean filter (IWMF) is proposed in this paper. IWMF is based on the effective weighted-mean framework and adopts an un-fixed shape symmetrical window, which has better detail protection ability. With the combination of such algorithms, both the denoising effect and real-time performance can be greatly improved.

The remaining parts are as follows: Section 2 introduces the scheme of the iterative weighted-mean filter. Section 3 presents the experimental results of the proposed method compared with other state-of-the-art methods. Finally, our conclusions are presented in Section 4.

## 2. Scheme of the Iterative Weighted-Mean Filter

Three stages are involved in the proposed model: The noise detection stage, noise removal stage, and iterative denoising stage. Figure 1 is the flow chart for the proposed model, with details about the flow chart. These three stages are described in detail in the following sections.



Figure 1. Flow chart for the proposed method.

#### 2.1. Stage 1: Noise Detection

At this stage, the pixels are divided into two categories—there are noisy pixels and noise-free pixels. It is reasonable to assume that the pixels with the extreme maximum and the extreme minimum values are noises [2,15,19,20].

For a pixel g in the image I, let I(g) be its pixel value, and R be its noise recognition matrix. The noise recognition matrix R can be written as:

$$R(g) = \begin{cases} 0, \ 0 < I(g) < 255 \\ 1, \ I(g) = 0 \text{ or } I(g) = 255 \end{cases}$$
(1)

where R(g) = 0 represents the "noise-free pixel" and R(g) = 1 represents the "noise pixel".

This direct noise detection algorithm cannot accurately recognize the noise pixels in certain images, such as images with high-brightness and artificial pictures. Images with such characteristics often have special regions called extreme intensity flat regions, in which all pixels have extreme intensity. Noise-free pixels in these regions will also be mistakenly detected as noise and removed, and then be severely distorted; artifacts and dark patch-like surfaces on the images cannot be avoided. Therefore, a robust detection algorithm is required to improve the denoising performance.

Take a white extreme intensity flat region (WFR) that is corrupted by salt-and-pepper noises as an example. The fundamental characteristics of the pixels include:

- 1. All pixels in this region have extreme intensity.
- 2. About half of the noise pixels take the intensity 255, so the total number of pixels with an intensity of 255 is greater than the pixels with an intensity of 0. In other words, pixels with an intensity of 255 are the majority.

For a pixel *g* in the image I, let I(*g*) be its pixel value and  $W_5(g)$  be its neighborhood window of size 5 × 5. If I(*g*) = 255 or I(*g*) = 0, label *g* as a noise candidate pixel; otherwise, label *g* as a noise-free pixel. For a noise candidate pixel  $g_{candidate}$ , if all pixels in  $W_5(g_{candidate})$  with intensity 0 or 255, and  $N_{255} > T$ , where  $N_{255}$  is the number of pixels with an intensity of 255, and the optimal value of T is 20, then  $g_{candidate}$  is in the white extreme intensity flat regions. If  $I(g_{candidate}) = 255$ , it is considered a noise-free pixel, otherwise it is a noise pixel.

Such a procedure is also applicable to black extreme intensity flat regions. If a pixel g is in a black extreme intensity flat region and I(g) = 255, then it is considered a noise-free pixel; otherwise, it is a noise pixel. Therefore, mathematically, the improved detection algorithm can be given as

$$R(g) = \begin{cases} 0, & 0 < I(g) < 255 \\ 0, & g \text{ is in WFR and } I(g) = 255 \\ 0, & g \text{ is in BFR and } I(g) = 0 \\ 1, & \text{otherwise} \end{cases}$$
(2)

where WFR is white extreme intensity flat regions, and BFR is black extreme intensity flat regions. R(g) = 0 represents the "noise-free pixel" and R(g) = 1 represents the "noise pixel".

## 2.2. Stage 2: Noise Removal

In this stage, a decision-based weighted mean filter with an adaptive window is used for denoising. The pixels that are labeled noise-free (R(g) = 1) in the noise detection stage remain unchanged, and the noises (R(g) = 0) are replaced by the restored intensity. The steps for noise removal are as follows.

#### 2.2.1. Selection of Filtering Window

Generally, selecting a small filtering window can preserve image details better, while a large filtering window can adapt to a higher noise density [11,22]. To adaptively change the filtering windows according to the noise density, and thus, improve the process efficiency, a symmetrical window with an unfixed shape was adopted in this paper. The size was adaptively changed according to the noise density. This kind of window can better reflect the local correlation between pixels, and has a stronger detail protection ability than the traditional square window.

Let *r* be the Euclidean distance between *g* and the other pixels in  $W_5(g)$ , and we define five different windows  $r_i$  ( $r_1$ ,  $r_2$ , ...,  $r_5$ ) according to the value of *r*, as shown in Figure 2a. Let W be the filtering window, and we choose W according to the following three rules:

- 1. If the number of noise-free pixels in  $W_5(g)$  is greater than 3, then set  $W = r_1$  as the candidate filtering window. If the number of noise-free pixels in W is less than 3, then let  $W = r_1 + r_2$ . By analogy, increase W by  $r_i$  ( $r_1$ ,  $r_2$ , ...,  $r_5$ ) until the number of noise-free pixels selected exceeds 2.
- 2. If the number of noise-free pixels in  $W_5(g)$  is 1 or 2, then let  $W = W_5(g)$ .
- 3. If all pixels in  $W_5(g)$  are noise, then a suitable filtering window cannot be obtained. In this case, the pixel needs to be further detected by method 2 in Section 2.2.2.

$r_5$	$r_4$	$r_3$	$r_4$	$r_5$	
$r_4$	$r_2$	$r_1$	$r_2$	$r_4$	
$r_3$	<i>r</i> <sub>1</sub>	g	$r_1$	$r_3$	
$r_4$	$r_2$	$r_1$	$r_2$	$r_4$	
$r_5$	$r_4$	$r_3$	<i>r</i> <sub>4</sub>	$r_5$	
(a)					

**Figure 2.** Illustrated images: (a) Distribution of windows  $r_i$  ( $r_1$ ,  $r_2$ , ...,  $r_5$ ), (b) Weighted matrix X.

#### 2.2.2. Calculation of Noise Pixels Restored Value

The calculation of noise restored value is based on weighted mean filtering and the distribution characteristics of the pixels. For each noise, the pixels g, if the number of noise-free pixels in  $W_5(g)$  is greater than 0, then the method in 1 is used for denoising; otherwise, the method in 2 is used.

1. In the spatial filtering theory, corrupted pixels can be restored using the normalized weighted mean of all pixels in the neighborhood. The noise restored value can be calculated as (3). Replace the noise pixel value with the restored value, and set R(g) = 0.

$$F(i,j) = \frac{\sum_{(r,s)\in D} w(r,s) I(i+r,j+s)}{\sum_{(r,s)\in D} w(r,s)}$$
(3)

Here, *D* is the noise-free pixels group in window *W*, (i, j) is the coordinate of pixel *g*, (r, s) is the relative coordinate with (i, j) as the center, and w(r, s) is the weighted function (4), which can be obtained from

$$w(r,s) = 1/\sqrt{r^2 + s^2}$$
(4)

In general, the value of a pixel is closer to the value of its neighboring pixels than the values of far pixels. The median filter takes the median value of pixels in the window as the restoration value of noise pixels, and cannot reflect the difference of the spatial position of pixels. To overcome such a defect, a distanced weighted method is proposed for denoising in this paper. The weighted coefficients are formed to be a center symmetric weight matrix whose values gradually decrease with the increase of distance, as shown in Figure 2b. Such a treatment can well reflect the local correlation between pixels, thereby reducing the loss of detail and avoiding image blur.

2. If *g* is in the extreme intensity flat regions, then the recovery step is performed according to the formula (5) and set R(g) = 0; otherwise, the pixel is processed in Stage 3.

$$F(g) = \begin{cases} 0, & \text{if } g \text{ is in BFR} \\ 255, & \text{if } g \text{ is in WFR} \end{cases}$$
(5)

#### 2.3. Stage 3: Noise Removal by Iterative Approach

For the images with high-density noises, as the pixels of a small neighborhood may all be destroyed, it is difficult to calculate the restoration value based on noise-free pixels in a small neighborhood, and thus, a wider window of the pixels is required. However, too large of a filtering window may result in blurring and unnecessary distortion. If the large window does not contain noise-free pixels, some filters replace the intensity of noisy pixels with the previously processed pixel or the mean of processed pixels in the neighborhood, such as References [17,31], which will lead the restoration image to have a streaking effect or artifacts. Such problems can be solved by using an iterative filter to produce higher quality images.

For each pixel *g* in image I, the iterative procedure can be described as follows.

- 1. For each pixel g with R(g) = 1, process g by the method proposed in stage 2.
- 2. If R is not a zero matrix, repeat 1 until R becomes a zero matrix, but use the last reconstruct image as the input image. Otherwise, leave it unchanged. If all pixels in the image are noisy pixels, then the procedure should stop.

Taking a noise image, for example, the denoising procedure can be described, as shown in Figure 3. The image segment obtained from "Lena. jpg" is chosen for illustration, and it is corrupted by salt-and-pepper noise with a density of 70%.



Figure 3. A specific illustration of the proposed method.

## 3. Simulation Results

To evaluate the performance of the proposed algorithm, we used several pictures (Figure 4) and 100 natural images (Figure 5) from the UC Berkeley dataset [33]. All experiments were performed using a PC with Intel) I(TM) i5-9893 CPU @ 3.10 GHZ, 16GB RAM. The program codes were written in C++. In experiments, the performance of the proposed filter was compared with several state-of-the-art filters, such as ASWMF [12], AFMF [9], DAMF [10], DBA [19], AWMF [26], TSF [20], FSMMF [31], MDBMF [17], and ERMI [28], in terms of the PSNR (peak signal to noise ratio), the SSIM (structural similarity index), visual perception, and the average processing time. The PSNR and SSIM are defined as follows:

$$PSNR = 10 \times lg \frac{m \times n \times 255^2}{\sum_{i=1}^{m} \sum_{j=1}^{n} (f(i,j) - g(i,j))^2}$$
(6)

$$SSIM = \frac{(2u_f u_g + C_1)(2\delta_{fg} + C_2)}{(u_e^2 + u_e^2 + C_1)(\delta_e^2 + \delta_e^2 + C_2)}$$
(7)

$$C_1 = (K_1 L)^2, \ C_2 = (K_2 L)^2$$
(8)

where *f* is the original image, *g* is the denoising image, and m and *n* are the length and width of the image, respectively;  $u_f$  and  $u_g$  are the mean of *f* and *g*, respectively;  $\sigma_f$  and  $\sigma_g$  are the standard deviation of *f* and *g*, respectively;  $\sigma_{fg}$  is the covariance between  $u_f$  and  $u_g$ ;  $C_1$  and  $C_2$  are constants used to maintain stability, where L = 255 is the dynamic range of pixel values, and  $K_1 = 0.01$ ,  $K_2 = 0.03$ .



Figure 4. Several test images: (a) Lena, (b) Peppers, (c) Barbara, and (d) Boat.



Figure 5. Several images of the UC Berkeley dataset: (a) Test018, (b) Test006, and (c) Training098.

#### 3.1. Evaluate by Visual Perception and Quantitative Measurements

Figures 6–8 show the restored images using IWMF and several filters for the images Lena and Test018 corrupted by noise density 0.9 and Test006 corrupted by noise density 0.8. It can obviously be seen from the reconstructed images that the ASWMF and AFMF failed to remove noises thoroughly. The DAMF demonstrated good noise removal capabilities; however, the restored images still contain some subtle noisy dots. The AWMF, TSF, and FSMMF outperformed the previous filters in removing noise thoroughly, but introduced a blur effect on the restored image. The ERMI over smoothed the images and lost some details for the high-density noise images. The DBA and MDBMF were affected by a streaking effect to an extent, which degraded the quality of the restored images. Comparatively, the IWMF showed superior performance in terms of noise removal and detail preservation.

In Figure 7, due to a certain number of pixels with extreme intensity values in the original image, the results generated by SAMF, ASWMFF, FSMMF, and ERMI were highly unsatisfactory. The MDBMF replaces a pixel with the previous pixel when there is still no noise-free pixel in  $5 \times 5$  neighborhoods, which caused the restored image to suffer from the streaking effect. The TSF caused a large number of pixels with an intensity of 255 to be destroyed. The IWMF performed well in removing the noise in extreme intensity flat regions and was able to generate the highest quality images.

Tables 1–3 provide the comparative results of the IWMF and other filters in terms of the PSNR and SSIM using the Lena, Test018, Boat, and Training098 images corrupted with the noise densities ranging from 0.1 to 0.9. The proposed IWMF showed the best noise removal performance of all the filters, according to the values of PSNR and SSIM. The FSMMF and DBA were barely satisfactory, and the DBA produced the lowest PSNR and SSIM in most ranges of the noise density. The ERMI

was effective in removing low-density salt-and-pepper noises but was not suitable for high-density noises removal. The values of PSNR and SSIM for the TSF, MDBMF, DAMF, and ERMI filters were close to each other but lower than the values of IWMF. We further inferred from Figure 9 that the IWMF produced the highest values of PSNR and SSIM at noise densities ranging from 0.1 to 0.9, which indicates that the IWMF gave the best performances in preserving image details and generating the highest quality images.



**Figure 6.** Visual perception of IWMF versus different methods on Test018: (**a**) Original image, (**b**) Noisy image of density 0.9, (**c**) AFMF, (**d**) ASWMF, (**e**) TSF, (**f**) DBA, (**g**) FSMMF, (**h**) ERMI, (**i**) AWMF, (**j**) DAMF, (**k**) MDBMF, and (**l**) IWMF.



**Figure 7.** Visual perception of IWMF versus different methods on Test006: (a) Original image, (b) Noisy image of density 0.8, (c) AFMF, (d) ASWMF, (e) TSF, (f) DBA, (g) FSMMF, (h) ERMI, (i) AWMF, (j) DAMF, (k) MDBMF, (l) IWMF.



**Figure 8.** Visual perception of IWMF versus different methods on Lena: (a) Original image, (b) Noisy image of density 0.9, (c) AFMF, (d) ASWMF, (e) TSF, (f) DBA, (g) FSMMF, (h) ERMI, (i) AWMF, (j) DAMF, (k) MDBMF, and (l) IWMF.

Table 1. The PSNR and SSIM of IGMF versus different methods on Lena.

Noise Density, %	10	30	50	70	90	10	30	50	70	90	
		P	SNR (d	B)		SSIM (%)					
AFMF	37.9	34.9	31.8	29.0	23.1	97.8	94.9	91.6	85.8	70.2	
DBA	40.5	34.8	30.4	26.1	19.9	98.6	94.8	90.5	80.1	56.5	
ASWMF	42.2	36.1	32.3	29.0	23.8	98.9	96.3	92.3	85.7	67.2	
TSF	43.2	36.8	33.0	30.2	27.1	98.9	96.5	93.1	87.9	80.2	
AWMF	39.6	36.7	32.3	28.2	24.5	98.9	96.0	91.9	84.1	76.4	
DAMF	43.2	36.9	33.1	30.1	27.0	99.1	96.5	93.0	87.8	80.1	
FSMMF	40.9	34.3	30.5	27.9	23.9	98.7	95.2	89.6	83.7	73.2	
ERMI	42.2	36.9	31.7	29.5	25.7	99.0	96.6	91.7	86.7	75.3	
MDBMF	42.7	36.6	33.0	30.1	26.1	99.0	96.5	93.0	87.7	77.8	
IWMF	43.3	37.6	34.0	31.0	27.1	99.1	96.9	93.8	89.3	80.3	

Table 2. The PSNR and SSIM of Test018 versus different methods on Test018.

Noise Density, %	10	30	50	70	90	10	30	50	70	90		
	PSNR (dB)						SSIM (%)					
AFMF	37.5	36.5	32.8	30.5	24.9	96.2	97.2	94.7	89.3	78.6		
DBA	40.2	34.6	31.9	27.2	20.4	98.5	95.3	86.6	80.2	70.6		
ASWMF	43.0	37.0	33.3	30.1	25.4	99.3	97.5	94.5	89.2	71.9		
TSF	43.5	36.9	33.7	30.6	27.5	99.2	97.3	94.5	89.4	84.5		
AWMF	40.5	35.6	31.5	26.9	25.0	99.0	96.8	94.3	89.2	71.8		
DAMF	43.5	36.9	33.8	30.4	27.3	99.3	97.3	94.7	89.2	84.3		
FSMMF	41.3	34.9	31.8	29.5	26.8	99.1	96.2	92.2	87.2	78.5		
ERMI	42.1	36.7	32.5	30.1	27.8	99.2	97.4	92.9	88.6	80.7		
MDBMF	42.7	36.8	33.6	30.9	27.8	99.3	97.4	94.6	90.6	83.3		
IWMF	44.0	37.9	34.8	31.7	28.5	99.5	98.0	95.8	91.9	84.5		

Noise Density, %	10	30	50	70	90	10	30	50	70	90	
		Р	SNR (d	B)		SSIM (%)					
AFMF	34.3	32.1	30.7	27.1	23.5	95.3	92.9	88.0	79.8	68.2	
DBA	36.1	31.2	28.1	23.5	18.9	97.2	90.1	80.6	73.2	59.9	
ASWMF	38.9	33.4	30.1	27.3	23.1	98.6	95.0	89.5	81.2	60.8	
TSF	39.1	33.3	30.5	28.3	25.2	98.5	95.1	89.9	83.4	71.5	
AWMF	38.0	33.2	30.1	27.4	22.5	98.4	94.5	88.5	80.2	60.0	
DAMF	39.1	33.5	30.6	28.1	25.0	98.5	95.2	90.1	82.9	71.4	
FSMMF	38.1	32.0	28.8	26.6	24.4	98.3	93.4	86.3	78.1	65.5	
ERMI	39.0	33.9	29.4	27.4	25.2	98.6	95.3	86.7	79.1	66.5	
MDBMF	39.1	33.6	30.5	28.0	25.1	98.6	95.1	90.0	82.9	70.6	
IWMF	40.3	34.7	31.4	28.6	25.6	98.9	96.1	91.6	84.3	71.6	

Table 3. The PSNR and SSIM of IWMF versus different methods on Training098.



**Figure 9.** Versus different methods on 100 natural images in terms of the average values of (**a**) PSNR, and (**b**) SSIM.

## 3.2. Evaluate by Computational Time

The CPU elapsed time is an important index to evaluate the performance of the filter. Table 4 illustrates the average CPU processing time of the discussed seven filters for 100 natural images with size  $481 \times 321$ . These images were corrupted by salt-and-pepper noise with densities ranging from 0.1 to 0.9. For each method, the execution time increased with the increasing noise density, except for AWMF. The AFMA had the longest processing time, due to the complexity of denoising strategy. The denoising strategies of the ASWMF and AFMF meant they processed more pixels than the other methods, which prolonged the execution time.

Table 4. The average processing times were obtained by the denoising methods for 100 natural images.

Noise Density, %	10	20	30	40	50	60	70	80	90
	Time (ms)								
AFMF	22.44	23.61	28.97	36.16	50.77	60.25	70.97	94.59	116.6
DBA	12.07	12.66	13.25	13.25	13.31	13.25	13.31	13.42	12.84
ASWMF	7.95	19.79	21.91	22.49	25.15	28.74	33.04	34.51	29.68
TSF	2.53	6.07	7.71	7.77	8.18	8.95	9.37	7.25	7.18
AWMF	28.38	26.62	26.21	25.79	24.79	24.85	24.44	24.03	23.67
DAMF	2.54	6.05	7.68	7.72	8.07	8.99	9.18	7.19	7.21
FSMMF	3.59	6.12	9.54	13.61	16.37	16.93	19.14	19.91	19.96
ERMI	0.94	1.59	2.29	3.29	9.77	10.36	10.48	11.36	23.09
MDBMF	2.29	5.59	7.42	7.01	7.95	8.54	8.66	6.18	6.01
IWMF	0.76	1.06	1.35	1.82	2.29	3.01	3.59	4.24	4.59

The execution time of the MDBWF and FSMMF were close to each other, and at a medium level. The ERMI and IWMF, by using the mean-based filtering instead of median-based filtering and without sorting pixel values, showed the best performance in the computing and processing time. In addition, the use of small windows also played an important role in improving efficiency.

## 4. Discussion

In this paper, we proposed an efficient salt-and-pepper noise removal filter, the iterative weighted mean filter (IWMF). In the noise detection stage, the extreme value and the distribution characteristics of the noisy image were used to identify the noise pixels, which overcame the problem of the inaccurate noise detection of traditional switching filters. In the noise removal stage, a weighted mean filter was applied to restore the noisy image, and this was performed iteratively. Comparatively, this noise removal technique effectively avoided the problem of detail blur and streaking effects, especially with high-density noise. Extensive simulation results demonstrate that the proposed filter outperformed other existing state-of-the-art filters in visual perception and quantitative measurements. Due to the efficient filter framework and low computational complexity, IWMF is very suitable for real-time implementation.

Author Contributions: Conceptualization, F.C. and Y.L.; methodology, F.C.; software, F.C.; validation, M.H.; formal analysis, Z.M.; investigation, Z.M. and F.C.; resources, M.H.; writing—original draft preparation, F.C.; writing—review and editing, Y.L.; visualization, Y.L.; supervision, M.H.; project administration, M.H. and Q.H.; funding acquisition, M.H. and Q.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** Guangxi Specially-invited Experts Foundation of Guangxi Zhuang Autonomous Region, China (GuiRenzi2019(13)).

Acknowledgments: This work was supported by Guangxi Specially-invited Experts Foundation of Guangxi Zhuang Autonomous Region, China (GuiRenzi2019(13)).

Conflicts of Interest: The authors declare no conflict of interest.

## References

- 1. Sundaramurthy, S. Digital Image Processing, 2nd ed.; Oxford University Press: Oxford, UK, 2016.
- 2. Chen, J.Y.; Zhan, Y.W.; Cao, H.Y.; Xiong, G.Q. Iterative grouping median filter for removal of fixed value impulse noise. *IET Image Proc.* **2019**, *13*, 946–953. [CrossRef]
- 3. Tukey, J.W. Exploratory Data Analysis. J. Am. Stat. Assoc. 1977, 28. [CrossRef]
- 4. Bhadouria, V.S.; Ghoshal, D.; Siddiqi, A.H. A new approach for high density saturated impulse noise removal using decision-based coupled window median filter. *Signal Image Video Process.* **2014**, *8*, 71–84. [CrossRef]
- Charalampidis, D. Steerable Weighted Median Filters. *IEEE Trans. Image Process. IEEE Signal Process. Soc.* 2010, 19, 882–894. [CrossRef]
- Chen, J.; Zhan, Y.; Cao, H. Iterative deviation filter for fixed-valued impulse noise removal. *Multimed. Tools Appl.* 2020, 79, 23695–23710. [CrossRef]
- Chen, J.; Zhan, Y.; Cao, H.; Wu, X. Adaptive probability filter for removing salt and pepper noise. *IET Image Proc.* 2018, 12, 863–871. [CrossRef]
- 8. Dong, Y.; Xu, S. A New Directional Weighted Median Filter for Removal of Random-Valued Impulse Noise. *IEEE Signal Process. Lett.* **2007**, *14*, 193–196. [CrossRef]
- 9. Erkan, U.; Enginolu, S.; Thanh, D.N.H.; Minh, H.L. Adaptive Frequency Median Filter for the Salt and Pepper Denoising Problem. *IET Image Proc.* **2020**, *14*, 1291–1302. [CrossRef]
- Erkan, U.; Gokrem, L.; Enginolu, S. Different applied median filter in salt and pepper noise. *Comput. Electr. Eng.* 2018, 70, 789–798. [CrossRef]
- Esakkirajan, S.; Veerakumar, T.; Subramanyam, A.N.; Premchand, C.H. Removal of High Density Salt and Pepper Noise through Modified Decision Based Unsymmetric Trimmed Median Filter. *IEEE Signal Process. Lett.* 2011, *18*, 287–290. [CrossRef]
- 12. Faragallah, O.S.; Ibrahem, H.M. Adaptive switching weighted median filter framework for suppressing salt-and-pepper noise. *AEU Int. J. Electron. Commun.* **2016**, *70*, 1034–1040. [CrossRef]

- 13. Hsieh, M.H.; Cheng, F.C.; Shie, M.C.; Ruan, S.-J. Fast and efficient median filter for removing 1–99% levels of salt-and-pepper noise in images. *Eng. Appl. Artif. Intell.* **2013**, *26*, 1333–1338. [CrossRef]
- Hwang, H.; Haddad, R.A. Adaptive median filters: New algorithms and results. *IEEE Trans. Image Process.* 1995, 4, 499–502. [CrossRef]
- 15. Ibrahim, H.; Kong, N.S.P.; Ng, T.F. Simple adaptive median filter for the removal of impulse noise from highly corrupted images. *IEEE Trans. Consum. Electron.* **2008**, *54*, 1920–1927. [CrossRef]
- Khan, S.; Lee, D.H. An adaptive dynamically weighted median filter for impulse noise removal. *EURASIP J. Adv. Signal. Process.* 2017, 1–4. [CrossRef]
- Kunsoth, R.; Biswas, M. Modified decision based median filter for impulse noise removal. In Proceedings of the IEEE International Conference on Wireless Communications, Signal Processing and Networking, Chennai, India, 23–25 March 2016; pp. 1316–1319.
- Park, H. Making Joint-Histogram-Based Weighted Median Filter Much Faster. *IEICE Trans. Inf. Syst.* 2015, 98, 721–725. [CrossRef]
- 19. Srinivasan, K.S.; Ebenezer, D. A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noises. *IEEE Signal Process. Lett.* **2007**, *14*, 189–192. [CrossRef]
- 20. Thanh, D.N.H.; Hai, N.H.; Prasath, S.; Minh, H.L.; Tavares, J.M.R.S. A Two-Stage Filter for High Density Salt and Pepper Denoising. *Multimed. Tools Appl.* **2020**, *79*, 21013–21035. [CrossRef]
- 21. Erkan, U.; Gokrem, L. A new method based on pixel density in salt and pepper noise removal. *Turk. J. Electr. Eng. Comput. Sci.* **2018**, *26*, 162–171. [CrossRef]
- 22. Yuan, S.; Tan, Y. Impulse noise removal by a global–local noise detector and adaptive median filter. *Signal Process.* **2006**, *86*, 2123–2128. [CrossRef]
- 23. Zhang, Z.; Han, D.; Dezert, J.; Yang, Y. A new adaptive switching median filter for impulse noise reduction with pre-detection based on evidential reasoning. *Signal Process.* **2018**, *147*, 173–189. [CrossRef]
- Goel, N.; Aggarwal, P. Enhanced Modified Decision-Based Unsymmetric Trimmed Adaptive Neighborhood Winsorized Mean Filter for Removing 1–99% Levels of Salt-and-Pepper Noise. *Iran. J. Sci. Technol. Trans. Electr. Eng.* 2019, 43, 459–468. [CrossRef]
- 25. Iqbal, N.; Ali, S.; Khan, I.; Lee, B. Adaptive Edge Preserving Weighted Mean Filter for Removing Random-Valued Impulse Noise. *Symmetry* **2019**, *11*. [CrossRef]
- 26. Chen, J.; Huang, N.; Xiong, G.; Cao, H.; Xu, Q. Adaptive weighted mean filtering algorithm based on confidence interval. *J. Nanjing Univ. Sci. Technol.* **2017**, *22*, 1538–1575.
- 27. Nasri, M.; Saryazdi, S.; Nezamabadi-Pour, H. A Fast Adaptive Salt and Pepper Noise Reduction Method in Images. *Circuits Syst. Signal Process.* 2013, *32*, 1839–1875. [CrossRef]
- Ramadan, Z.M. Efficient Restoration Method for Images Corrupted with Impulse Noise. *Circuits Syst. Signal Process.* 2012, 31, 1397–1406. [CrossRef]
- 29. Wang, X.; Shen, S.; Shi, G.; Xu, Y.; Zhang, P. Iterative non-local means filter for salt and pepper noise removal. *J. Vis. Commun. Image Represent.* **2016**, *34*, 440–450. [CrossRef]
- 30. Zhang, P.; Li, F. A New Adaptive Weighted Mean Filter for Removing Salt-and-Pepper Noise. *IEEE Signal Process. Lett.* **2014**, *21*, 1280–1283. [CrossRef]
- 31. Vijaykumar, V.R.; Santhana Mari, G.; Ebenezer, D. Fast switching based median-mean filter for high density salt and pepper noise removal. *EURASIP J. Adv. Signal Process.* **2014**, *68*, 1145–1155. [CrossRef]
- 32. Zhang, C.; Wang, K. A switching median–mean filter for removal of high-density impulse noise from digital images. *Optik* **2015**, *126*, 956–961. [CrossRef]
- The UC Berkeley Dataset. Available online: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/ bsds/BSDS300/html/dataset/images.html (accessed on 31 October 2020).

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).