Proc. of the 2nd International Conference on Electrical, Communication and Computer Engineering (ICECCE)
12-13 June 2020, Istanbul, Turkey

Improved Adaptive Weighted Mean Filter for Salt-and-Pepper Noise Removal

Uğur Erkan
Department of Computer Engineering
Karamanoğlu Mehmetbey University
Karaman, Turkey
ugurerkan@kmu.edu.tr

Dang N. H. Thanh
Department of Information Technology, School of Business Information Technology,
University of Economics Ho Chi Minh City, Vietnam
thanhdnh@ueh.edu.vn

Serdar Enginoğlu
Department of Mathematics
Çanakkale Onsekiz Mart University
Çanakkale, Turkey
serdarenginoglu@comu.edu.tr

Samet Memiş
Department of Mathematics
Çanakkale Onsekiz Mart University
Çanakkale, Turkey
sametmemis@stu.comu.edu.tr

Abstract— **In this study, we propose an improved adaptive weighted mean filter (IAWMF) to remove salt-and-pepper noise. The most prominent advantage of IAWMF is its ability to take into account the weights of noise-free pixels in the adaptive window. Hence, the new grey value occurs closer to the original grey value of the centre pixel than the grey value computed by the adaptive weighted mean filter (AWMF). Moreover, the proposed method utilises the advantage of AWMF to reduce the error of detecting noisy pixels. In the experiments, we compare the denoising results of the proposed method with other state-of-the-art image denoising methods. The results confirm that IAWMF outperforms other methods.**

Keywords—Image denoising, salt-and-pepper noise, non-linear function, image processing, Euclidean pixel similarity

I. INTRODUCTION

Image denoising is a crucial pre-processing task to improve image quality. Depending on various image formation methods, an image can be affected by noise. There are several types of noise, such as Gaussian noise [1], Poisson noise [2], Speckle noise [3], and impulse noise [4]. The salt-and-pepper noise (SPN) is a simple type of impulse noise [5, 6]. Noisy pixels of SPN have only two values: 255 white/salt pixels and 0 black/pepper pixels for an 8-bit greyscale image.

To remove SPN, there are several approaches, e.g. the ones based on filtering [7-15], regularization [16], wavelet transform [17], and machine learning [18]. In this study, we only consider denoising methods relying on nonlinear filters. Because nonlinear filters are non-iterative, their implementing speed does not depend on tolerance. Nonlinear filters are usually developed based on median or mean. In recent years, among the nonlinear filters proposed to remove SPN are the adaptive median filter (AMF) [7], the noise adaptive fuzzy switching median filter (NAFSMF) [8], the decision based unsymmetric trimmed median filter (DBUTMF) [19, 4], the modified DBUTMF (MDBUTMF) [20], the adaptive weighted mean filter (AWMF) [21], the different applied median filter (DAMF) [22], the adaptive type-2 fuzzy filter (FDS, fuzzy denoising for SPN) [23], the adaptive Riesz mean filter (ARmF) [24], and the iterative mean filter (IMF) [25]. NAFSMF, MDBUTMF, AWMF, DAMF, FDS, ARmF, and IMF are the state-of-the-art filters for high-density noise. NAFSMF is a recursive filter. It removes noise employing a fuzzy function. Both DBUTMF and MDBUTMF work well in the occurrence of high-density noise. However, if the considered window incorporates only salt pixel or pepper pixel, the trimmed median of DBUTMF cannot be evaluated. MDBUTMF overcomes this drawback and is more effective than DBUTMF. DAMF and FDS are other effective filters intended for high-density noise. FDS is developed based on type-2 fuzzy sets. ARmF and IMF are effective too. ARmF is an adaptive method. Its essential success comes from the fact that it uses the pixel similarity-based Riesz mean. IMF is an iterative method; therefore, its performance depends on tolerance.

AMF is an improved version of the median filter (MF) to remove medium-density and high-density noise. AMF works more effectively than MF because it uses an adaptive window. Although AMF works well in the presence of low-density noise and even of medium-density and high-density noise, it may fail to detect some noisy pixels. AWMF substantially overcomes this drawback. AWMF sets the mean of the noise-free pixels of a considered adaptive window as the new value of its centre pixel. Yet the grey value computed by AWMF is not close enough to the possible value of the pixel. Our goal is to improve the method by a weighted mean based on Euclidean pixel similarity (eps-mean). We evaluate various weights for noise-free pixels based on their distance to the centre pixel.

The rest of the study is organised as follows: Section II describes the SPN and provides some needed notions. Then, a new method called "Improved Adaptive Weighted Mean Filter (IAWMF)" and its motivation are presented. Section III presents the experimental results and their discussion. Finally, Section IV concludes the study.

II. PROPOSED SALT-AND-PEPPER DENOISING METHOD

A. Definitions and Notions

Throughout this paper, let $X = \left[x_{ij}\right]_{m\times n}$, $Y = \left[y_{ij}\right]_{m\times n}$, and $Z = \left[z_{ij}\right]_{m\times n}$ be an original image, a corrupted (noisy) image by SPN, and a restored image, respectively, where m and n are the numbers of pixels by the image height and by the image width, respectively.

978-1-7281-7116-6/20/$31.00 ©2020 IEEE

Definition 1. $[s_{min}, s_{max}]$ denotes the dynamic grey value of an image. For an 8-bit greyscale image, $s_{min} = 0$ and $s_{max} = 255$. The SPN model can be presented as follows:

$$y_{ij} = \begin{cases} s_{min}, & with\ probability\ p \\ s_{max}, & with\ propability\ q \\ x_{ij}, & with\ probability\ 1-(p+q) \end{cases}$$

where $0 \le p + q \le 1$ is called the SPN level/density/ratio. If the noise level is an even number, then $p = q$.

Hereinafter, let $S_{ij}(w)$ be a window with a size of $(2w+1) \times (2w+1)$ centred at a pixel location (i,j) and its indices set be $\{(i^*, j^*): |i^* - i| \le w, |j^* - j| \le w\}$. Moreover, we denote $S_{ij}^{min}(w), S_{ij}^{max}(w)$, and $S_{ij}^{mean}(w)$ for the minimum value, the maximum value, and the weighted mean value used in AWMF of $S_{ij}(w)$, respectively.

Definition 2. Let $S_{ij}(w)$ be a window in $[y_{ij}]_{m\times n}$. The weighted mean based on Euclidean pixel similarity (eps-mean) of $S_{ij}(w)$ is defined as follows:

$$S_{ij}^{epsmean}(w) = \begin{cases} \mathcal{R}_{ij}^{\delta}(w), & \sum_{(i^*,j^*)\in S_{ij}(w)} \delta_{i^*j^*} \neq 0 \\ -1, & otherwise \end{cases}$$

where

$$\mathcal{R}_{ij}^{\delta}(w) = \frac{\sum_{(i^*,j^*)\in S_{ij}(w)} \delta_{i^*j^*}\mathcal{D}_{i^*j^*}y_{i^*j^*}}{\sum_{(i^*,j^*)\in S_{ij}(w)} \delta_{i^*j^*}\mathcal{D}_{i^*j^*}},$$

$$\delta_{i^*j^*} = \begin{cases} 1, & if\ S_{ij}^{min}(w) < y_{i^*j^*} < S_{ij}^{max}(w) \\ 0, & otherwise \end{cases},$$

and

$$\mathcal{D}_{i^*j^*} = \frac{1}{\left(\varepsilon + \sqrt{(i-i^*)^2 + (j-j^*)^2}\right)^4}, \quad 0 < \varepsilon \ll 1$$

Here, ε is the stabilization parameter, typically, $\varepsilon = 0.001$.

B. Motivation of the proposed method

AWMF was developed based on AMF. This development is built on decreasing the number of errors in detecting noisy pixels; thus, AWMF exhibits a better denoising performance than AMF. However, the new grey value of the centre pixel of the considered window is computed based on grey values of noise-free pixels of the window without considering the distance between the pixels and the centre pixel. Since this evaluation can result in artefacts and sharpen edges, image details will look unnatural. IAWMF focuses on reducing artefacts and smoothing edges by placing weights.

C. AWMF and the proposed method

In this subsection, firstly, AWMF and IAWMF are presented in Algorithm 1 and 2, respectively. Note that IAWMF uses the $S_{ij}^{epsmean}(w)$ instead of the $S_{ij}^{mean}(w)$ used in AWMF. $S_{ij}^{mean}(w)$ is computed as an average value of grey values of all noise-free pixels in $S_{ij}(w)$.

Let us consider how IAWMF works. For each pixel of a noisy image, we determine a window size by gradually enlarging the window from 3×3 to the maximum size. The centre pixel of the window is considered noisy if its grey value is equal to the maximum or the minimum value; otherwise, it is a noise-free pixel. If the centre pixel is noisy, then the eps-mean of the windows is assigned as its new grey value.

Algorithm 1. AWMF (Adaptive Weighted Mean Filter)

```
Input: [y_ij]_{m×n}
Output: [z_ij]_{m×n}
Initialize h = 1, w_max = 39
For each pixel (i,j) of the noisy image [y_ij]_{m×n}
Initialize w = 1
Repeat
  Compute S_ij^min(w), S_ij^max(w), S_ij^mean(w), S_ij^min(w+h), S_ij^max(w+h)
  If S_ij^min(w) ≠ S_ij^min(w+h) & S_ij^max(w) ≠ S_ij^max(w+h) & S_ij^mean(w) = −1
    If S_ij^min(w) < y_ij < S_ij^max(w)
      z_ij = y_ij
    Else
      z_ij = S_ij^mean(w)
    End
  Break
  Else
    w = w + h
    If w > w_max
      z_ij = S_ij^mean(w)
    End
  End
Until w > w_max
End
```

Algorithm 2. IAWMF (Improved Adaptive Weighted Mean Filter)

```
Input: [y_ij]_{m×n}
Output: [z_ij]_{m×n}
Initialize h = 1, w_max = 39, ε = 0.001
For each pixel (i,j) of the noisy image [y_ij]_{m×n}
Initialize w = 1
Repeat
  Compute S_ij^min(w), S_ij^max(w), S_ij^mean(w), S_ij^min(w+h), S_ij^max(w+h)
  If S_ij^min(w) ≠ S_ij^min(w+h) & S_ij^max(w) ≠ S_ij^max(w+h) & S_ij^mean(w) = −1
    If S_ij^min(w) < y_ij < S_ij^max(w)
      z_ij = y_ij
    Else
      z_ij = S_ij^epsmean(w)
    End
  Break
  Else
    w = w + h
    If w > w_max
      z_ij = S_ij^epsmean(w)
    End
  End
Until w > w_max
End
```

Secondly, let us see the following example to understand the difference between AWMF and IAWMF. We consider a window of 5×5 from the location (85,198) of the Cameraman image. The left matrix stands for the noise-free image and the right matrix for the noisy image with a noise level of 60%.

183	174	146	107	59	255	255	**146**	107	59
184	172	140	100	56	255	172	255	0	255
187	178	149	110	60	0	178	255	110	255
192	187	163	127	69	192	187	0	255	255
193	191	176	146	78	0	0	176	146	78

For AWMF, the new grey value assigned to the centre pixel is the average value of grey values of all noise-free pixels:

$$mean\{146, 107, 59, 172, 178, 110, 192, 187, 176, 146, 78\} = 141$$

We consider the noisy image on the right side. Since the value of the centre pixel is 255 (red), the centre pixel located in (3,3) is considered noisy. Let us evaluate the value

$\delta_{13}\mathcal{D}_{13}y_{13}$ for the noise-free pixel, 146 (in bold), located in (1,3). Here,

$$\mathcal{D}_{13} = \frac{1}{\left(0.001 + \sqrt{(|3-1|^2 + |3-3|^2)}\right)^4} = 0.0624$$

$\delta_{13} = 1$, and $y_{13} = 146$. Therefore, $\delta_{13}\mathcal{D}_{13}y_{13} = 9.1068$. Thus, the eps-mean of the right window, the new value of the centre pixel, is calculated as follows for noise-free pixels whose $\delta_{i^*j^*} = 1$.

i^*	j^*	$y_{i^*j^*}$	$\mathcal{D}_{i^*j^*}$	$\mathcal{D}_{i^*j^*}y_{i^*j^*}$
1	3	146	0.0624	9.1068
1	4	107	0.0339	4.2724
1	5	59	0.0156	0.9206
2	2	172	0.2493	42.8786
3	2	178	0.9960	177.2898
3	4	110	0.9960	109.5611
4	1	192	0.0399	7.6663
4	2	187	0.2493	46.6180
5	3	176	0.0624	10.9780
5	4	146	0.0399	5.8296
5	5	78	0.0156	1.2170
		SUM:	2.7663	416.3381
		New Grey Value:	416.3381/2.7663≈ **151**	

Hence, IAWMF assigns 151 to the centre pixel. As can be observed, the new grey value evaluated by IAWMF (i.e. 151) is closer to the original grey value (i.e. 149) than to the grey value evaluated by AWMF (i.e. 141).

III. Experimental Results and Discussion

We conduct the experiments of the proposed method for SPN denoising on MATLAB R2019b. The configuration of the computing system is the workstation with I(R) Xeon(R) CPU E5 1620 v4 @ 3.5 GHz and 64 GB RAM, and Windows 10 Pro.

A. *Image Quality Assessment Metrics*

To evaluate image quality, we use Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) [26]:

$$\mathrm{PSNR}(X,Z) = 10\log_{10}\left(\frac{L^2}{\mathrm{MSE}(X,Z)}\right)$$

where L denotes the maximum grey level, e.g. $L = 255$ for 8-bit images, and MSE is the mean squared error defined by

$$\mathrm{MSE}(X,Z) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{ij} - z_{ij}\right)^2$$

Here, x_{ij} and z_{ij} are grey values at a pixel located in (i,j) of the original image X and of the restored image Z, respectively; m and n are the numbers of pixels by image width and height, respectively.

Structural similarity (SSIM) is proven to be a suitable error metric for assessing image quality and gives a value in the range of [0, 1], where a value closer to 1 (one) indicates better structure preservation. SSIM between two images X and Z with the same size of $m \times n$ is computed as follows:

$$SSIM(X,Z) = \frac{(2\mu_X\mu_Z + C_1)(2\sigma_{XZ} + C_2)}{(\mu_X^2 + \mu_Z^2 + C_1)(\sigma_X^2 + \sigma_Z^2 + C_2)}$$

where $\mu_X, \mu_Z, \sigma_X^2, \sigma_Z^2$, and σ_{XZ} are the average of X, the average of Z, the variance of X, the variance of Z, and the covariance of X and Z, respectively. Here, $C_1 = (K_1L)^2$ and $C_2 = (K_2L)^2$ are two constants such that $K_1 = 0.01$ and $K_1 = 0.03$ and $L = 255$ for 8-bit images.

B. *Datasets, Test Cases, and Discussion*

We consider three test cases: the first test case is for the Cameraman image, the second for the Lena image, and the third for 40 images of the TESTIMAGES dataset [27] and 200 images of the BSDS dataset [28]. The Cameraman and Lena images are stored in greyscale and have the same size, i.e. 512×512 pixels. The TESTIMAGES dataset and the BSDS dataset contain greyscale images in PNG format (TESTIMAGES) and JPEG format (BSDS) with various sizes.

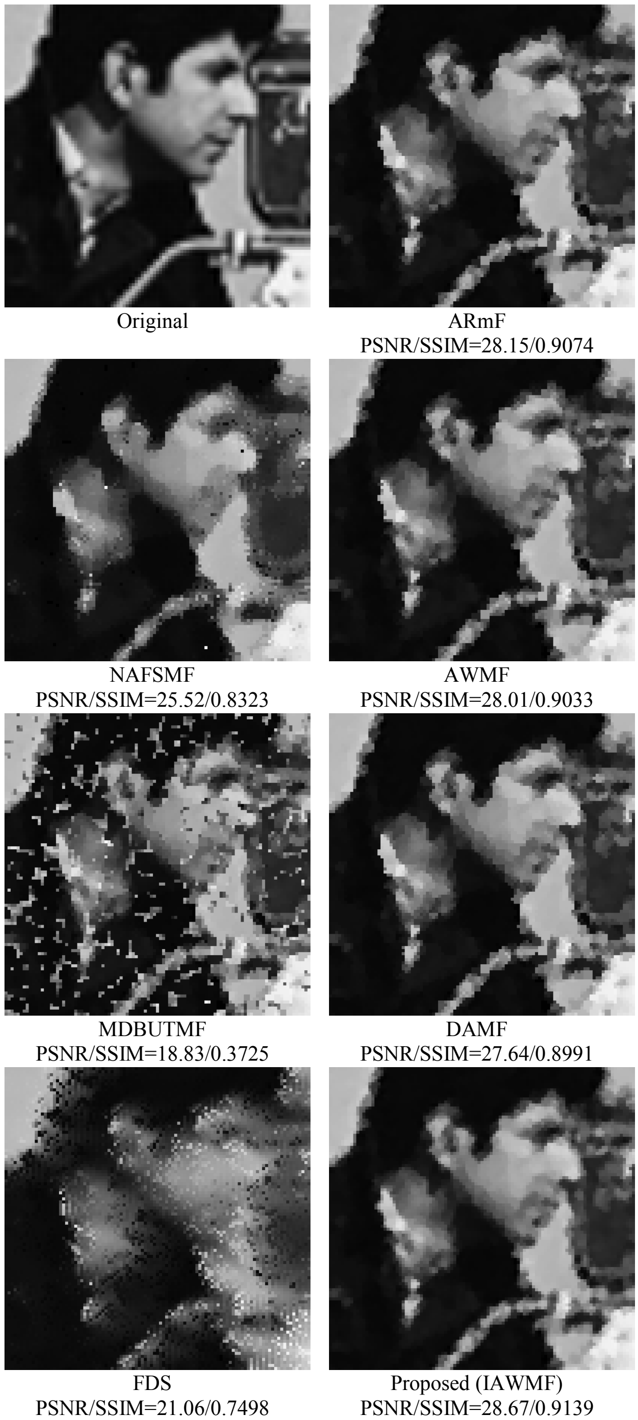
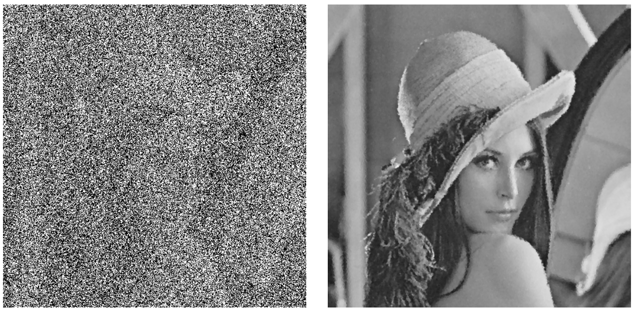


Fig. 1. Denoising results of the methods with the noise level of 80% (a part of 100×100 pixels) of the Cameraman image (512×512 pixels).

Noise of 20% PSNR/SSIM= 12.42/0.0836 | Denoising of 20% PSNR/SSIM=39.80/0.9812

Noise of 40% PSNR/SSIM= 9.42/0.0359 | Denoising of 40% PSNR/SSIM=35.84/0.9559

Noise of 60% PSNR/SSIM= 7.65/0.0180 | Denoising of 60% PSNR/SSIM=32.62/0.9186



Noise of 80% PSNR/SSIM= 6.42/0.0091 | Denoising of 80% PSNR/SSIM=29.32/0.8537

Fig. 2. Denoising results of the proposed method with various noise levels of the Lena image (512×512 pixels).

The Cameraman image as the first test case: We generate SPN (80%) over the Cameraman image and implement ARmF, NAFSMF, AWMF, MDBUTMF, DAMF, FDS, and the proposed method to this noisy image. In Figure 1, we then provide a zoomed-in part of the images with the size of 100×100 pixels. MDBUTMF works ineffectively on the zoomed-in images and causes many defects. Much noise remains on the result of FDS. NAFSMF removes noise more effectively than MDBUTMF does. However, though NAFSMF leads to noise on the resultant image less than MDBUTMF, small details such as the eye and the mouth are lost. Although AWMF, DAMF, and the proposed method work quite efficaciously and the noise is removed to a greater extent, there is a substantial difference in their denoising results. For the result of DAMF, the nose is lost and the edges are excessively sharpened. Further, there are many artefacts in the result of DAMF, especially on the face and the neck of the man. The denoising results by AWMF and ARmF look better than those by DAMF. In AWMF and ARmF, the artefacts are reduced and details such as the eye and the mouth are preserved well, but the details are sharpened.

On the other hand, the denoising results by the proposed method, i.e. IAWMF, are impressive, in which noise and artefacts are removed to a greater extent and small details such as the eye and the mouth are preserved well. Moreover, the edges are smoothened effectively. The face and the neck are smoothened and exhibit a better visual quality than the results by DAMF and AWMF. By intuition, we can observe that the proposed method yields the best denoising results. The PSNR and SSIM results of ARmF, NAFSMF, AWMF, MDBUTMF, DAMF, FDS, and IAWMF are 28.15/0.9074, 25.52/0.8323, 28.01/0.9033, 18.83/0.3725, 27.64/0.8991, 21.06/0.7498, and **28.67/0.9139**, respectively. As understandable, the results of the denoising result by the proposed method are the highest.

The Lena image as the second test case: We generate SPN (20%, 40%, 60%, and 80%) over the Lena image and implement the proposed method to these noisy images. The denoising results are presented in Figure 2. We can observe that our method ably removes noise at various noise levels. The PSNR and SSIM results for the noise levels of 20%, 40%, 60%, and 80% are 39.80/0.9812, 35.84/0.9559, 32.62/0.9186, and **29.32/0.8537**, respectively.

TABLE I. Average PSNR/SSIM value of the denoising results of 40 images of the TESTIMAGES dataset.

Method	20%	40%	60%	80%	Average
NAFSMF	34.14/0.9602	30.56/0.9207	27.90/0.8719	24.83/0.7882	29.34/0.8734
MDBUTMF	36.57/0.9773	31.45/0.9457	25.77/0.8151	17.82/0.3750	27.90/0.7520
AWMF	36.53/0.9749	34.26/0.9570	31.36/0.9234	27.68/0.8512	32.46/0.9177
DAMF	37.24/0.9815	32.82/0.9554	29.71/0.9160	26.41/0.8430	31.55/0.9154
FDS	35.32/0.9766	30.37 /0.9327	25.63/0.8342	20.66/0.6501	28.00/0.8339
ARmF	39.51/0.9852	35.39/0.9653	31.78/0.9300	27.79/0.8558	33.62/0.9341
IAWMF	39.52/0.9853	35.42/0.9658	31.98/0.9328	28.27/0.8658	33.80/0.9374

TABLE II. Average PSNR/SSIM value of the denoising results of 200 images of the BSDS dataset.

Method	20%	40%	60%	80%	Average
NAFSMF	30.91/0.9391	27.78/0.8757	25.73/0.7994	23.52/0.6888	26.99/0.8258
MDBUTMF	33.54/0.9634	29.59/0.9147	25.21/0.7811	18.42/0.3757	26.69/0.7587
AWMF	32.63/0.9573	30.25/0.9217	27.84/0.8649	24.97/0.762	28.92/0.8765
DAMF	33.39/0.9648	29.66/0.9197	26.99/0.8563	24.33/0.7538	28.59/0.8737
FDS	32.41/0.9624	28.22/0.9033	24.67/0.7953	21.13/0.6211	26.61/0.8205
ARmF	34.48/0.9714	30.95/0.9345	28.07/0.8757	24.96/0.7679	29.62/0.8874
IAWMF	34.68/0.9726	31.17/0.9370	28.37/0.8811	25.43/0.7805	29.91/0.8928

The TESTIMAGES dataset and the BSDS dataset as third test cases: We add SPN (20%, 40%, 60%, and 80%) to the images of the datasets. After implementing the denoising methods, we evaluate the average PSNR/SSIM results of all the images by each noise level. The results for the TESTIMAGES dataset and the BSDS dataset are presented in Table I and II, respectively.

From Table I, we can infer that the results of the proposed method are higher than the ones of the other methods. It holds for Table II. Moreover, the average PSNR/SSIM values of all the noise levels of the proposed method are observed to be the highest for the TESTIMAGES dataset as well as for the BSDS dataset (the last columns of Table I and II).

Besides, note that the proposed method focuses on improving the noise removal performance of AWMF. In all the cases, we can easily observe that the proposed method remove noise much more effectively than AWMF in terms of both visual results and the PSNR/SSIM results.

IV. CONCLUSIONS

In this paper, we propose a two-step improved adaptive weighted mean filter (IAWMF) to remove SPN. Firstly, IAWMF detects noisy pixels as AWMF does. In the second step, differently from AWMF, it uses EPS-based weights to evaluate a new grey value. IAWMF can effectively remove noise with various noise levels without creating artefacts. Moreover, IAWMF naturally smoothens edges and details. Based on a variety of test cases, we can confirm that IAWMF outperforms the compared denoising methods.

REFERENCES

[1] W. Lu, J. Duan, Z. Qiu, Z. Pan, R. W. Liu, and L. Bai, "Implementation of high□order variational models made easy for image processing", *Math. Methods Appl. Sci.,* vol. 39, no. 14, pp. 4208-4233, 2016.

[2] D. N. H. Thanh, V. B. S. Prasath, and L. M. Hieu, "A Review on CT and X-Ray Images Denoising Methods", *Informatica,* vol. 43, no. 2, pp. 151-159, 2019.

[3] A. Saadia and A. Rashdi, "A Speckle Noise Removal Method", *Circuits Syst. Signal Process.,* vol. 37, no. 6, p. 2639–2650, 2018.

[4] T. Bai and J. Tan, "Automatic Detection and Removal of High-Density Impulse Noises", *IET Image Process.,* vol. 9, no. 2, pp. 162-172, 2015.

[5] R. Rojas and P. Rodriguez, "Spatially Adaptive Total Variation Image Denoising Under Salt and Pepper Noise", in *19th European Signal Processing Conference EUSIPCO,* pp 278-282, Barcelona, 2011.

[6] M. S. and G. M. Arash, "Impulse Noise Reduction Based on Improved Median ϵ-Filter", *Int. J. Electron.,* vol. 99, no. 11, pp. 1489-1496, 2012.

[7] H. Hwang and R. A. Haddad, "Adaptive Median Filters: New Algorithms and Results", *IEEE Trans. Image Process.,* vol. 4, no. 4, p. 499–502, 1995.

[8] K. K. V. Toh and N. A. M. Isa, "Noise Adaptive Fuzzy Switching Median Filter for Salt-and-Pepper Noise Reduction", *IEEE Signal Process Lett.,* vol. 17, no. 3, pp. 281-284, 2010.

[9] U. Erkan and L. Gökrem, "A New Method Based on Pixel Density in Salt and Pepper Noise Removal", *Turk. J. Electr. Eng. Comput. Sci.,* vol. 26, no. 1, p. 162–171, 2018.

[10] U. Erkan, L. Gökrem, and S. Enginoğlu, "Adaptive Right Median Filter for Salt-and-Pepper Noise Removal", *Int. J. Eng. Res. Dev.*, vol. 11, no. 2, 542-550, 2019.

[11] D. N. H. Thanh, V. B. S. Prasath, and U. Erkan, "An Improved BPDF Filter for High Density Salt and Pepper Denoising", in *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF 2019),* pp. 1-5, Danang, Vietnam, 2019.

[12] U. Erkan, S. Enginoğlu, and D. N. H. Thanh, "A Recursive Mean Filter for Image Denoising", in *2019 International Artificial Intelligence and Data Processing Symposium (IDAP 2019),* pp. 1-5, Malatya, Turkey, 2019.

[13] U. Erkan, S. Enginoğlu, and D. N. H. Thanh and L. M. Hieu, "Adaptive Frequency Median Filter for the Salt-and-Pepper Denoising Problem", *IET Image Process.*, vol. 14, no. 7, 1291-1302, 2020.

[14] S. Enginoğlu, U. Erkan, and S. Memiş, "Adaptive Cesáro Mean Filter for Salt-and-Pepper Noise Removal", *El-Cezeri J. Sci. Eng.,* vol. 7, no. 1, 304-314, 2020.

[15] B. Arpacı, E. Kurt, K. Çelik, and B. Ciylan, "Colored Image Encryption and Decryption with a New Algorithm and a Hyperchaotic Electrical Circuit". *J. Electr. Eng. Technol.,* vol. 15, 1413-1429, 2020.

[16] D. N. H. Thanh, L. T. Thanh, N. N. Hien, and V. B. S. Prasath, "Adaptive Total Variation L1 Regularization for Salt and Pepper Image Denoising", *Optik,* vol. 208, 163677, 1-10, 2020.

[17] S. Deivalakshmi and P. Palanisamy, "Removal of High Density Salt and Pepper Noise Through Improved Tolerance Based Selective Arithmetic Mean Filtering with Wavelet Thresholding", *AEU Int. J. Electron. Commun.,* vol. 70, no. 6, pp. 757-776, 2016.

[18] L. Jin, W. Zhang, G. Ma, and E. Song, "Learning Deep CNNs for Impulse Noise Removal in Images", *J. Visual Commun. Image Represent.,* vol. 62, pp. 193-205, 2019.

[19] K. Aiswarya, V. Jayaraj, and D. Ebenezer, "A New and Efficient Algorithm for the Removal of High Density Salt and Pepper Noise in Images and Videos", in *IEEE 2nd International Conference on Computer Modeling and Simulation*, Sanya, 2010.

[20] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, and C. H. PremChand, "Removal of High Density Salt and Pepper Noise Through Modified Decision Based Unsymmetric Trimmed Median Filter", *IEEE Signal Process Lett.,* vol. 18, no. 5, pp. 287-290, 2011.

[21] P. Zhang and F. Li, "A New Adaptive Weighted Mean Filter for Removing Salt-and-Pepper Noise", *IEEE Signal Process Lett.,* vol. 21, no. 10, pp. 1280-1283, 2014.

[22] U. Erkan, L. Gokrem, and S. Enginoglu, "Different Applied Median Filter in Salt and Pepper Noise", *Comput. Electr. Eng.,* vol. 70, pp. 789-798, 2018.

[23] V. Singh, R. Dev, N. K. Dhar, P. Agrawal, and N. K. Verma, "Adaptive type-2 fuzzy Approach for Filtering Salt and Pepper Noise in Greyscale Images", *IEEE Trans. Fuzzy Syst.,* vol. 26, no. 5, pp. 3170 - 3176, 2018.

[24] S. Enginoglu, U. Erkan, and S. Memiş, "Pixel Similarity-Based Adaptive Riesz Mean Filter for Salt-and-Pepper Noise Removal", *Multimedia Tools Appl.,* vol. 78, no. 24, p. 35401–35418, 2019.

[25] U. Erkan, D. N. H. Thanh, L. M. Hieu, and S. Enginoglu, "An Iterative Mean Filter for Image Denoising", *IEEE Access,* vol. 7, pp. 167847 - 167859, 2019.

[26] Z. Wang, A. Bovik, H. Sheikh, Simoncelli, and Eero, "Image Quality Assessment: From Error Visibility to Structural Similarity", *IEEE Trans. Image Process.,* vol. 13, no. 4, pp. 600-612, 2004.

[26] N. Asuni and A. Giachetti, "TESTIMAGES: A Large Data Archive for Display and Algorithm Testing", *J. Graphics Tools,* vol. 17, no. 4, pp. 113-125, 2015.

[28] P. Arbelaez, C. Fowlkes, and D. Martin, "The Berkeley Segmentation Dataset and Benchmark", UC Berkeley, 2007. [Online]. Available: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/. [Accessed 6 January 2020].