Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi
International Journal of Engineering Research and Development

UMAGD, (2019) 11(2), 542-550.

10.29137/umagd.495904

Cilt/Volume:11 Sayı/Issue:2 Haziran/June 2019

Araştırma Makalesi / Research Article

Adaptive Right Median Filter for Salt-and-Pepper Noise Removal

Uğur Erkan*[,1], Levent Gökrem[2], Serdar Enginoğlu[3]

[1]*Computer Engineering, Engineering Faculty, Karamanoğlu Mehmetbey University, Karaman, TURKEY*
[2]*Department of Mechatronics Engineering, Faculty of Engineering and Natural Sciences, Tokat Gaziosmanpaşa University, Tokat,TURKEY*
[3]*Department of Mathematics, Faculty of Arts and Sciences, Çanakkale Onsekiz Mart University, Çanakkale, TURKEY*

***Başvuru/Received**: 12/11/2018* ***Kabul/Accepted**: 14/05/2019* ***Son Versiyon/Final Version**: 30/06/2019*

Abstract
In image processing, nonlinear filters are commonly used as a pre-process for noise removal before applying any advanced processing such as classification and clustering to an image. The adaptive filters being a kind of the nonlinear filters mainly perform better than the others in salt-and-pepper noise. In this paper, we first define a new median method, i.e. right median (rm). We then define a new adaptive nonlinear filter developed via rm, namely Adaptive Right Median Filter (ARMF), for salt-and-pepper noise removal. Afterwards, we compare the results of ARMF with some of the known filters by using 12 test images and two image quality metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM). The results show that ARMF outperforms the other methods at all the noise density except 80% and 90% in the mean percentages. Finally, we discuss the need for further research.

Key Words
"Image denoising, Noise removal, Nonlinear filters, Nonlinear functions, Matrix algebra"

***Sorumlu Yazar/Corresponding Author:** uerkan80@gmail.com*

1. Introduction

The image processing is an area applied to many sectors from military to movie industries (Tomasi & Manduchi, 1998; Xiong et al., 2016). With image processing methods, problems in these areas are tried to be troubleshot. These methods are realised in spatial and frequency domains (Han et al., 2015). The filter is the most basic operation of image processing and computer vision (Lee et al., 2016). One of the essential processes in the spatial domain is low or high pass filters. Before the advanced processing of the images such as classification and clustering, they are treated by these filters. What is aimed at these processes is to eliminate the undesirable features of the image as much as possible. The success of these processes directly affects other operations in obtaining quality images.

Noise removal and keeping image information such as edges, textures, and other details are also essential topics for image denoising (Erkan & Gökrem, 2018; Jiang et al., 2014; Li et al., 2015; Liu et al., 2017; Nguyen & Chun, 2017; Rafsanjani et al., 2017; Xu et al., 2017). In camera-sensors, faulty memory locations in hardware, or transmissions in a noisy channel lead to some kinds of noise such as salt-and-pepper noise (SPN) and Gaussian noise (Bai et al., 2014; Chan et al., 2005; Gellert & Brad, 2016; Xu et al., 2014). SPN substantially lowers the quality of the image and randomly sets certain pixel values in the image to the maximum or minimum value (Erkan & Kilicman, 2016; Lin et al., 2010; Sun et al., 2015; Wang et al., 2016). One of the frequently used methods to remove SPN is nonlinear filters such as Standard Median Filter (SMF) (Pratt, 1975; Tukey, 1977), Adaptive Median Filter (AMF) (Hwang & Haddad, 1995), Median Filter without Repetition (MFWR) (Erkan & Gökrem, 2017), Progressive Switching Median Filter (PSMF) (Wang & Zhang, 1999), Decision Based Filtering Algorithm (DBA) (Pattnaik et al., 2012) Modified Decision-Based Unsymmetric Trimmed Median Filter (MDBUTMF) (Esakkirajan et al., 2011), and Noise Adaptive Fuzzy Switching Median Filter (NAFSMF) (Toh & Isa, 2010).

In this paper, in Section 2, we define a new method, i.e. Adaptive Right Median Filter (ARMF), which is improved via the right median (rm), for SPN removal. In Section 3, we compare ARMF with DBA, MDBUTMF, and NAFSMF by using 12 test images via two image quality metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM). Finally, we discuss the need for further research.

2. Preliminaries and ARMF Algorithm

In this section, firstly, we give some basic notions provided in (Erkan et al., 2018). Throughout this paper, let $A := [a_{ij}]_{m\times n}$ be an image matrix (IM) such that a_{ij} is an unsigned integer number, $0 \leq a_{ij} \leq 255$, and for at least one i and j, $a_{ij} \neq 0$ and $a_{ij} \neq 255$.

Definition 2.1 Let A be an IM. Then, a_{ij} is called a noisy entry of A if $a_{ij} = 0$ or $a_{ij} = 255$; otherwise, a_{ij} is called a regular entry of A.

Definition 2.2 Let A be an IM. Then, A is called a noise image matrix (NIM) if for some i and j, a_{ij} is a noisy entry of A.

Definition 2.3 Let A be an NIM. Then, the matrix $B := [b_{ij}]_{m\times n}$ is called the binary matrix of A where

$$b_{ij} := \begin{cases} 0, & a_{ij} \text{ is a noisy entry of } A \\ 1, & \text{otherwise} \end{cases}$$

Definition 2.4 Let $A := [a_{ij}]_{m\times n}$ and $t \in \{1,2,\dots,\min\{m,n\}\}$, then the matrix $\bar{\bar{A}}_{tsim} := [\bar{\bar{a}}_{rs}]_{(m+2t)\times(n+2t)}$ called t-symmetric pad matrix of A is defined as follows:

$$\begin{bmatrix}
a_{tt} & \cdots & a_{t1} & a_{t1} & a_{t2} & \cdots & a_{tn} & a_{tn} & \cdots & a_{t(n-t+1)} \\
\vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
a_{1t} & \cdots & a_{11} & a_{11} & a_{12} & \cdots & a_{1n} & a_{1n} & \cdots & a_{1(n-t+1)} \\
a_{1t} & \cdots & a_{11} & \boldsymbol{a_{11}} & \boldsymbol{a_{12}} & \cdots & \boldsymbol{a_{1n}} & a_{1n} & \cdots & a_{1(n-t+1)} \\
a_{2t} & \cdots & a_{21} & \boldsymbol{a_{21}} & \boldsymbol{a_{22}} & \cdots & \boldsymbol{a_{2n}} & a_{2n} & \cdots & a_{2(n-t+1)} \\
a_{3t} & \cdots & a_{31} & \boldsymbol{a_{31}} & \boldsymbol{a_{32}} & \cdots & \boldsymbol{a_{3n}} & a_{3n} & \cdots & a_{3(n-t+1)} \\
\vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
a_{mt} & \cdots & a_{m1} & \boldsymbol{a_{m1}} & \boldsymbol{a_{m2}} & \cdots & \boldsymbol{a_{mn}} & a_{mn} & \cdots & a_{m(n-t+1)} \\
a_{mt} & \cdots & a_{m1} & a_{m1} & a_{m2} & \cdots & a_{mn} & a_{mn} & \cdots & a_{m(n-t+1)} \\
\vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
a_{(m-t+1)t} & \cdots & a_{(m-t+1)1} & a_{(m-t+1)1} & a_{(m-t+1)2} & \cdots & a_{(m-t+1)n} & a_{(m-t+1)n} & \cdots & a_{(m-t+1)(n-t+1)}
\end{bmatrix}$$

Example 2.1 Let $A = \begin{bmatrix} 11 & 0 & 13 \\ 255 & 22 & 23 \\ 31 & 32 & 0 \end{bmatrix}$. Then,

$$\bar{\bar{A}}_{2sim} = \begin{bmatrix} 22 & 255 & 255 & 22 & 23 & 23 & 22 \\ 0 & 11 & 11 & 0 & 13 & 13 & 0 \\ 0 & 11 & \mathbf{11} & \mathbf{0} & \mathbf{13} & 13 & 0 \\ 22 & 255 & \mathbf{255} & \mathbf{22} & \mathbf{23} & 23 & 22 \\ 32 & 31 & \mathbf{31} & \mathbf{32} & \mathbf{0} & 0 & 32 \\ 32 & 31 & 31 & 32 & 0 & 0 & 32 \\ 22 & 255 & 255 & 22 & 23 & 23 & 22 \end{bmatrix}_{7\times 7}$$

Definition 2.5 Let $A := [a_{ij}]_{m\times n}$, $\bar{\bar{A}}_{tsim}$ be a $tsim$-pad matrix of A and $k \in \{1,2,\dots,t\}$. Then, the matrix, denoted by A_{ij}^k,

$$\begin{bmatrix} \bar{\bar{a}}_{(i+t-k)(j+t-k)} & \cdots & \bar{\bar{a}}_{(i+t-k)(j+t+k)} \\ \vdots & \bar{\bar{a}}_{(i+t)(j+t)} & \vdots \\ \bar{\bar{a}}_{(i+t+k)(j+t-k)} & \cdots & \bar{\bar{a}}_{(i+t+k)(j+t+k)} \end{bmatrix}_{(2k+1)\times(2k+1)}$$

is called k-approximate matrix of a_{ij} in $\bar{\bar{A}}_{tsim}$.

Example 2.2 Let us consider Example 2.1. Then,

$$A_{21}^1 = \begin{bmatrix} 11 & 11 & 0 \\ 255 & 255 & 22 \\ 31 & 31 & 32 \end{bmatrix}$$

Definition 2.6 If all entries of a matrix are zero, then it is called a zero matrix and is denoted by $[0]$.

Secondly, we give two basic notions needed for ARMF.

Definition 2.7 Let A be an NIM, R be the nonempty set of all regular entries of A, and $|R|$ denote the cardinality of R. Then, the matrix $\vec{A} := [\vec{a}_{1w}]_{1\times|R|}$ is called strictly increasing regular entry matrix (SIREM) of A, where

$$\vec{a}_{1w} := \begin{cases} \min R, & w = 1 \\ \min\left(R\setminus\{\vec{a}_{11}, \vec{a}_{12}, \dots, \vec{a}_{1(w-1)}\}\right), & 1 < w \le |R| \end{cases}$$

Definition 2.8 Let $\vec{A} := [\vec{a}_{1w}]_{1\times|R|}$ be SIREM of A. Then, the value

$$rm\vec{A} := \begin{cases} \vec{a}_{1\left(\frac{|R|+1}{2}\right)}, & |R| \text{ is odd} \\ \vec{a}_{1\left(\frac{|R|+2}{2}\right)}, & |R| \text{ is even} \end{cases}$$

is called right median of $\vec{A}$.

Example 2.3 Let us consider Example 2.1. Then, $\vec{A}_{21}^1 = [11 \quad 22 \quad 31 \quad 32]$ and $rm\vec{A} = 31$.

Finally, we give ARMF which is a new adaptive method for salt-and-pepper noise removal. In this method, a NIM is considered, and its binary matrix is obtained. After that, t-symmetric pad matrices of these matrices are constructed. If an entry of the binary matrix is equal to zero and its 1-approximate matrix differs from zero matrices, then SIREM of the 1-approximate matrix is obtained, and the right median is evaluated. Afterwards, this right median is overwritten to the entry. If an entry of the binary matrix is equal to zero and its 1-approximate matrix is zero matrices, but the 2-approximate matrix of this entry differs from zero matrices, then SIREM of 2-approximate matrix is obtained, and the right median is evaluated. Afterwards, this right median is overwritten to the entry. Similarly, if an entry of the binary matrix is equal to zero and its $(k-1)$-approximate matrix is zero matrix but the k-approximate matrix of this entry differs from zero matrix, then SIREM of the k-approximate matrix is obtained, and the right median is evaluated. Afterwards, this right median is overwritten to the entry. Algorithm steps of this method are as follows:

ARMF Algorithm Steps

Step 1. Let $A \coloneqq [a_{ij}]_{m\times n}$ be a NIM such that $\min\{m,n\} \geq 3$.

Step 2. Write the binary matrix $B \coloneqq [b_{ij}]_{m\times n}$ of A.

Step 3. Write $\bar{\bar{A}}_{tsim}$ and $\bar{\bar{B}}_{tsim}$ such that $t = \min\{m,n\}$.

Step 4. For all i and j,

If $b_{ij} = 0$

$k \coloneqq 1$

While $k > 0$

If $[b_{ij}^k] \neq [0]$, then

a. Obtain $\vec{A}_{ij}^k$

b. Evaluate $rm\vec{A}_{ij}^k$

c. Set $a_{ij} \leftarrow rm\vec{A}_{ij}^k$

d. Break

Else

$k \coloneqq k+1$

Else

Keep the value of a_{ij}

3. Algorithms Results

In this section, we first determine 12 test images shown in Fig 1. The first four of these images are among the most popular images. The second four of them are from TID2013 (Ponomarenko et al., 2015), and the last four of them are randomly extracted from TESTIMAGES database (Asuni & Giachetti, 2014). We then give the PSNR and SSIM results of DBA, MDBUTMF, NAFSMF, and ARMF shown in Fig 2 for Baboon, Motocross, and Billiard-Balls images with 30%, 50%, and 70% SPN densities, respectively. The results show that ARMF performs better than other methods.

Afterwards, in Table 1 and 2, we give the results of the methods, for Cameraman, Lena, Baboon, and Peppers images ranging in noise densities from 10% to 90%. Moreover, in Table 3 and 4, we give the mean results of the methods for 12 test images. The results show that ARMF performs better than the others at all the noise density except 80% and 90% in the mean percentages. Here, PSNR is defined by

$$PSNR \coloneqq 10\log\left(\frac{255^2}{MSE}\right)$$

where MSE stands for the Mean Square Error and is defined by

$$MSE \coloneqq \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(e_{ij} - f_{ij}\right)^2$$

such that $E := [e_{ij}]$ is the earliest form/original image and $F := [f_{ij}]$ is the final form/corrupted image. And Structural Similarity (SSIM) (Wang et al., 2004) is defined by

$$SSIM(x,y) \coloneqq \frac{\left(2\mu_x\mu_y + C_1\right) + \left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right) + \left(\sigma_x^2 + \sigma_y^2 + C_2\right)}$$

where $\mu_x, \mu_y, \sigma_x, \sigma_y$, and σ_{xy} are the average intensities, standard deviations and cross-covariance for images x and y, respectively. In addition, $C_1 \coloneqq (K_1L)^2$ and $C_2 \coloneqq (K_2L)^2$ are two constants such that $K_1 = 0.01$, $K_2 = 0.03$, and $L = 255$ for 8-bit grayscale images.

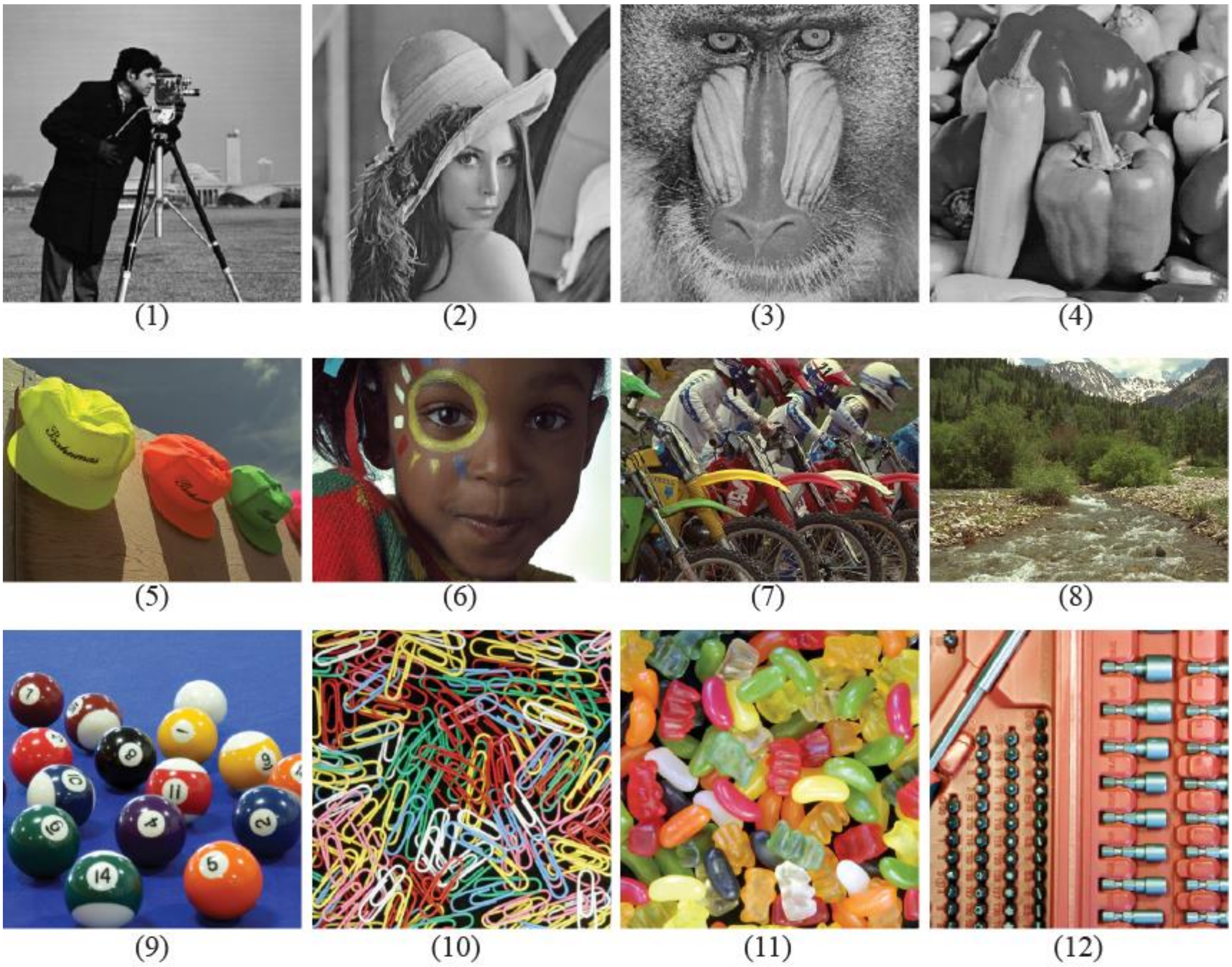

Fig 1. 12 test images: (1-4) from classic test images, (5-8) from TID2013, (9–l2) from TESTIMAGES Database.

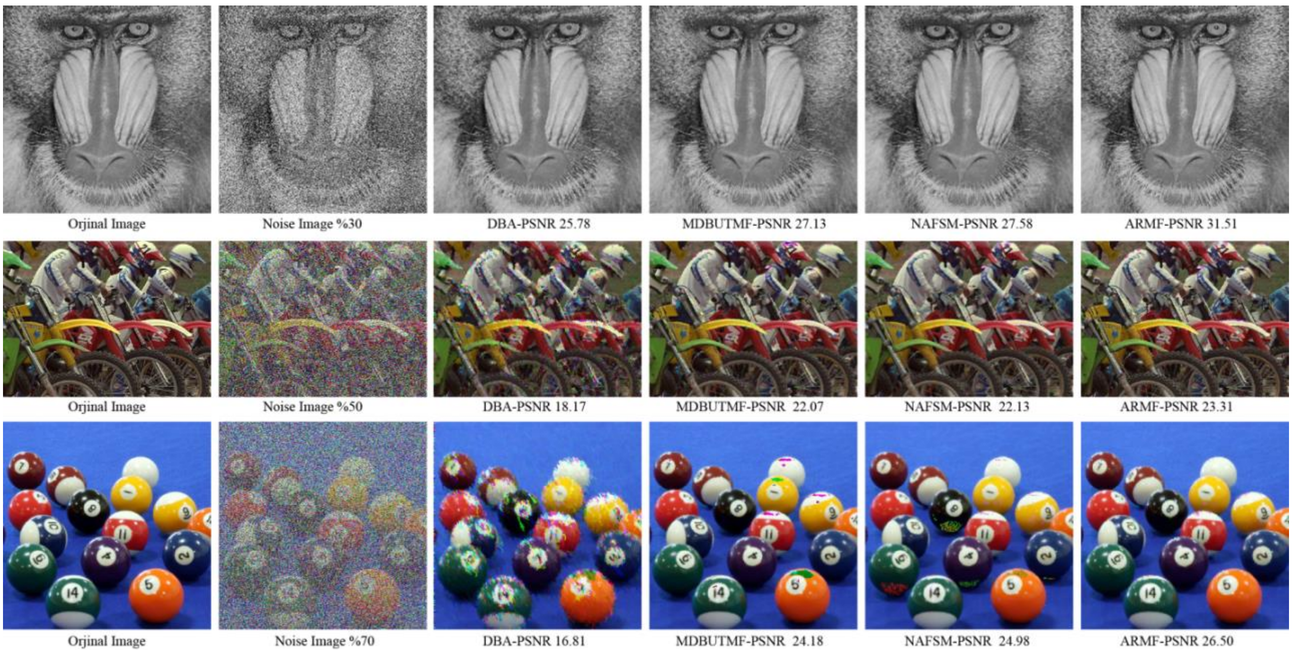

Fig. 2. PSNR results of the methods for Baboon, Motocross, and Billiard-Ball images with 30%, 50%, and 70% SPN densities, respectively.

Table 1. PSNR results the methods for some images

Image	Filter	10%	20%	30%	40%	50%	60%	70%	80%	90%
Cameraman	DBA	37.87	32.79	28.81	25.98	23.07	20.85	18.26	15.60	13.17
	MDBUTMF	35.30	30.99	29.27	29.77	30.25	29.39	27.81	23.59	15.10
	NAFSMF	36.91	33.96	31.84	30.50	29.49	28.33	27.00	25.57	**22.36**
	ARMF	**42.78**	**38.76**	**35.89**	**33.78**	**31.88**	**29.55**	**27.49**	**24.68**	20.48
Lena	DBA	37.78	33.51	29.84	27.12	24.45	21.87	19.31	16.12	13.21
	MDBUTMF	36.13	32.04	30.35	30.86	31.05	30.36	28.74	24.29	15.54
	NAFSMF	38.89	35.74	33.64	32.33	31.04	29.87	28.81	27.12	**23.53**
	ARMF	**42.10**	**38.68**	**36.26**	**34.25**	**32.59**	**30.62**	**28.54**	**25.99**	21.78
Baboon	DBA	33.29	29.01	25.78	23.52	21.56	19.82	18.21	16.54	13.76
	MDBUTMF	30.87	28.61	27.13	26.62	26.00	25.14	24.02	21.61	14.84
	NAFSMF	32.43	29.39	27.58	26.35	25.29	24.34	23.46	22.49	**20.52**
	ARMF	**37.60**	**33.96**	**31.51**	**29.70**	**27.85**	**26.17**	**24.39**	**22.51**	20.04
Peppers	DBA	36.63	32.64	29.51	26.75	24.06	21.11	18.65	15.52	11.94
	MDBUTMF	35.77	31.68	30.13	30.66	31.03	30.59	28.70	24.70	15.61
	NAFSMF	39.48	36.39	34.49	32.90	31.53	**30.46**	**29.04**	**27.28**	**23.68**
	ARMF	**40.40**	**37.06**	**34.83**	**33.45**	**31.74**	30.13	27.97	25.29	20.35
Mean	DBA	36.39	31.98	28.48	25.84	23.28	20.91	18.60	15.94	13.01
	MDBUTMF	34.51	30.82	29.21	29.47	29.58	28.87	27.31	23.54	15.27
	NAFSMF	36.92	33.87	31.88	30.52	29.33	28.25	27.07	**25.61**	**22.52**
	ARMF	**40.72**	**37.11**	**34.62**	**32.79**	**31.01**	**29.11**	**27.09**	24.61	20.66

Table 2. SSIM results the methods for some images

Image	Filter	10%	20%	30%	40%	50%	60%	70%	80%	90%
Cameraman	DBA	0.9881	0.9656	0.9309	0.8808	0.8123	0.7381	0.6589	0.5771	0.4738
	MDBUTMF	0.9488	0.8355	0.7749	0.8268	0.9012	0.9179	0.8959	0.7904	0.4078
	NAFSMF	0.9798	0.9637	0.9496	0.9346	0.9178	0.8991	0.8754	0.8326	0.7123
	ARMF	**0.9955**	**0.9896**	**0.9821**	**0.9720**	**0.9562**	**0.9345**	**0.9018**	**0.8413**	**0.7323**
Lena	DBA	0.9758	0.9414	0.8937	0.8308	0.7530	0.6625	0.5615	0.4486	0.3567
	MDBUTMF	0.9541	0.8691	0.8132	0.8442	0.8834	0.8834	0.8516	0.7386	0.3263
	NAFSMF	0.9836	0.9664	0.9484	0.9278	0.9058	0.8804	0.8486	0.8041	0.6813
	ARMF	**0.9894**	**0.9770**	**0.9630**	**0.9459**	**0.9259**	**0.8991**	**0.8607**	**0.8041**	**0.6843**
Baboon	DBA	0.9678	0.9148	0.8291	0.7234	0.6024	0.4672	0.3538	0.2559	0.1882
	MDBUTMF	0.9382	0.8795	0.8298	0.8186	0.8048	0.7682	0.7095	0.5964	0.2871
	NAFSMF	0.9618	0.9208	0.8767	0.8311	0.7794	0.7212	0.6541	0.5720	0.4443
	ARMF	**0.9876**	**0.9714**	**0.9504**	**0.9236**	**0.8845**	**0.8315**	**0.7518**	**0.6337**	**0.4554**
Peppers	DBA	0.9578	0.9098	0.8523	0.7853	0.7032	0.6018	0.5072	0.3915	0.2747
	MDBUTMF	0.9411	0.8457	0.7862	0.8121	0.8481	0.8448	0.8070	0.7070	0.3431
	NAFSMF	0.9783	0.9558	0.9337	0.9094	0.8817	**0.8542**	**0.8177**	**0.7668**	**0.6519**
	ARMF	**0.9802**	**0.9586**	**0.9358**	**0.9125**	**0.8841**	0.8514	0.8074	0.7422	0.6006
Mean	DBA	0.9724	0.9329	0.8765	0.8051	0.7177	0.6174	0.5203	0.4183	0.3234
	MDBUTMF	0.9455	0.8574	0.8010	0.8254	0.8594	0.8536	0.8160	0.7081	0.3411
	NAFSMF	0.9759	0.9517	0.9271	0.9007	0.8712	0.8387	0.7990	0.7439	**0.6225**
	ARMF	**0.9882**	**0.9741**	**0.9578**	**0.9385**	**0.9127**	**0.8791**	**0.8304**	**0.7553**	0.6181

Table 3. Mean PSNR results of the methods for 12 test images

Noise Density	DBA	MDBUTMF	NAFSMF	ARMF
10%	34.02	27.96	33.20	**36.82**
20%	29.94	25.87	30.70	**33.97**
30%	26.73	24.83	28.98	**31.72**
40%	24.02	24.82	27.63	**29.95**
50%	21.51	24.68	26.44	**28.30**
60%	19.10	24.05	25.39	**26.60**
70%	16.86	22.93	24.24	**24.51**
80%	14.35	20.45	**22.88**	22.16
90%	11.82	14.50	**20.19**	18.42

Table 4. Mean SSIM results of the methods for 12 test images

Noise Density	DBA	MDBUTMF	NAFSMF	ARMF
10%	0.9704	0.9338	0.9697	**0.9836**
20%	0.9285	0.8367	0.9455	**0.9692**
30%	0.8691	0.7782	0.9203	**0.9515**
40%	0.7911	0.8063	0.8947	**0.9306**
50%	0.6963	0.8458	0.8652	**0.9026**
60%	0.5911	0.8402	0.8324	**0.8656**
70%	0.4810	0.8003	0.7917	**0.8121**
80%	0.3736	0.6924	**0.7347**	0.7281
90%	0.2926	0.3455	**0.6099**	0.5752

4. Conclusion

In this study, we have proposed a new filter ARMF to remove the SPN. We then have shown that ARMF outperforms DBA, MDBUTMF, and NAFSMF methods at all the noise densities except 80% and 90% noise density in the mean percentages. ARMF uses an adaptive window size and enlarges the window size until the conditions in the algorithm are satisfied. In high-noise density, the regular entries in the windows with high-size may be too far away from the centre pixel. Therefore, ARMF has a drawback in the event that the noise densities bigger than 80%. It is an open question whether it is possible to improve the noise removal success of ARMF in SPNs with the high-density by limiting the window size like in NAFSMF.

References

Asuni, N., & Giachetti, A. (2014). TESTIMAGES: a large-scale archive for testing visual devices and basic image processing algorithms. In A. Giachetti (Ed.), *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference*. The Eurographics Association. https://doi.org/10.2312/stag.20141242

Bai, T., Tan, J., Hu, M., & Wang, Y. (2014). A novel algorithm for removal of salt and pepper noise using continued fractions interpolation. *Signal Processing*. https://doi.org/10.1016/j.sigpro.2014.03.023

Chan, R. H., Ho, C. W., & Nikolova, M. (2005). Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE Transactions on Image Processing*. https://doi.org/10.1109/TIP.2005.852196

Erkan, U., & Gökrem, L. (2017). Median filter without repetition in salt and pepper noise. *Gaziosmanpasa Journal of Scientific Research,* 6(2), 11-19. Retrieved from https://dergipark.org.tr/download/article-file/326397

Erkan, U., & Gökrem, L. (2018). A new method based on pixel density in salt and pepper noise removal. *Turkish Journal of Electrical Engineering and Computer Sciences*. https://doi.org/10.3906/elk-1705-256

Erkan, U., Gökrem, L., & Enginoğlu, S. (2018). Different applied median filter in salt and pepper noise. *Computers and Electrical Engineering*. https://doi.org/10.1016/j.compeleceng.2018.01.019

Erkan, U., & Kilicman, A. (2016). Two new methods for removing salt-and-pepper noise from digital images. *ScienceAsia*. https://doi.org/10.2306/scienceasia1513-1874.2016.42.028

Esakkirajan, S., Veerakumar, T., Subramanyam, A. N., & PremChand, C. H. (2011). Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter. *IEEE Signal Processing Letters*. https://doi.org/10.1109/LSP.2011.2122333

Gellert, A., & Brad, R. (2016). Context-based prediction filtering of impulse noise images. *IET Image Processing*, *10*(6), 429–437. https://doi.org/10.1049/iet-ipr.2015.0702

Han, J., Yue, J., Zhang, Y., & Bai, L. (2015). Local sparse structure denoising for low-light-level image. *IEEE Transactions on Image Processing*. https://doi.org/10.1109/TIP.2015.2447735

Hwang, H., & Haddad, R. A. (1995). Adaptive Median Filters: New algorithms and results. *IEEE Transactions on Image Processing*. https://doi.org/10.1109/83.370679

Jiang, J., Zhang, L., & Yang, J. (2014). Mixed noise removal by weighted encoding with sparse nonlocal regularization. *IEEE Transactions on Image Processing*. https://doi.org/10.1109/TIP.2014.2317985

Lee, G. S., Lee, S. H., Kim, G. O., Park, J. H., & Park, Y. S. (2016). A modified GrabCut using a clustering technique to reduce image noise. *Symmetry*. https://doi.org/10.3390/sym8070064

Li, Z., Zheng, J., Zhu, Z., Yao, W., & Wu, S. (2015). Weighted guided image filtering. *IEEE Transactions on Image Processing*. https://doi.org/10.1109/TIP.2014.2371234

Lin, C. H., Tsai, J. S., & Chiu, C. Te. (2010). Switching bilateral filter with a texture/noise detector for universal noise removal. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. https://doi.org/10.1109/ICASSP.2010.5495475

Liu, X., Jing, X. Y., Tang, G., Wu, F., & Ge, Q. (2017). Image denoising using weighted nuclear norm minimization with multiple strategies. *Signal Processing*. https://doi.org/10.1016/j.sigpro.2017.01.003

Nguyen, M. P., & Chun, S. Y. (2017). Bounded self-weights estimation method for non-local means image denoising using minimax estimators. *IEEE Transactions on Image Processing*. https://doi.org/10.1109/TIP.2017.2658941

Pattnaik, A., Agarwal, S., & Chand, S. (2012). A new and efficient method for removal of high density salt and pepper noise through cascade decision based filtering algorithm. *Procedia Technology*. https://doi.org/10.1016/j.protcy.2012.10.014

Ponomarenko, N., Jin, L., Ieremeiev, O., Lukin, V., Egiazarian, K., Astola, J., …, & Jay Kuo, C. C. (2015). Image database TID2013: Peculiarities, results and perspectives. *Signal Processing: Image Communication*. https://doi.org/10.1016/j.image.2014.10. 009

Pratt, W. K., (1975). *Semiannual Technical Report*. Image Processing Institute, University of Southern California.

Rafsanjani, H. K., Sedaaghi, M. H., & Saryazdi, S. (2017). An adaptive diffusion coefficient selection for image denoising. *Digital Signal Processing: A Review Journal*. https://doi.org/10.1016/j.dsp.2017.02.004

Sun, C., Tang, C., Zhu, X., Li, X., & Wang, L. (2015). An efficient method for salt-and-pepper noise removal based on shearlet transform and noise detection. *AEU - International Journal of Electronics and Communications*. https://doi.org/10.1016/ j.aeue.2015.09.007

Toh, K. K. V., & Isa, N. A. M. (2010). Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction. *IEEE Signal Processing Letters*. https://doi.org/10.1109/LSP.2009.2038769

Tomasi, C., & Manduchi, R. (1998). Bilateral filtering for gray and color images. *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 839–846. https://doi.org/10.1109/ICCV.1998.710815

Tukey, J. W. (1977). *Exploratory Data Analysis*. https://doi.org/10.1007/978-1-4419-7976-6

Wang, Y., Wang, J., Song, X., & Han, L. (2016). An efficient adaptive fuzzy switching weighted mean filter for salt-and-pepper noise removal. *IEEE Signal Processing Letters*. https://doi.org/10.1109/LSP.2016.2607785

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*. https://doi.org/10.1109/TIP.2003.819861

Wang, Z., & Zhang, D. (1999). Progressive switching median filter for the removal of impulse noise from highly corrupted images. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*. https://doi.org/10.1109/82.749102

Xiong, R., Liu, H., Zhang, X., Zhang, J., Ma, S., Wu, F., & Gao, W. (2016). Image denoising via bandwise adaptive modeling and regularization exploiting nonlocal similarity. *IEEE Transactions on Image Processing*. https://doi.org/10.1109/TIP.2016. 2614160

Xu, J., Wang, L., & Shi, Z. (2014). A switching weighted vector median filter based on edge detection. *Signal Processing*. https://doi.org/10.1016/j.sigpro.2013.11.035

Xu, S., Yang, X., & Jiang, S. (2017). A fast nonlocally centralized sparse representation algorithm for image denoising. *Signal Processing*. https://doi.org/10.1016/j.sigpro.2016.08.006