Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2018) 26: 162 – 171

doi:10.3906/elk-1705-256

A new method based on pixel density in salt and pepper noise removal

Uğur ERKAN[1,*], **Levent GÖKREM**[2]
[1]Department of Computer Programing, Erbaa Vocational School, Gaziosmanpaşa University, Erbaa, Tokat, Turkey
[2]Department of Mechatronics Engineering, Faculty of Engineering and Natural Sciences, Gaziosmanpaşa University, Tokat, Turkey

Received: 22.05.2017 • **Accepted/Published Online:** 14.11.2017 • **Final Version:** 26.01.2018

Abstract: In this paper, we deliver a new method to remove salt and pepper noise, which we refer to as based on pixel density filter (BPDF). The first step of the method is to determine whether or not a pixel is noisy, and then we decide on an adaptive window size that accepts the noisy pixel as the center. The most repetitive noiseless pixel value within the window is set as the new pixel value. By using 18 test images, we give the results of peak signal-to-noise ratio (PSNR), structural similarity (SSIM), image enhancement factor (IEF), standard median filter (SMF), adaptive median filter (AMF), adaptive fuzzy filter (AFM), progressive switching median filer (PSMF), decision-based algorithm (DBA), modified decision-based unsymmetrical trimmed median filter (MDBUTMF), noise adaptive fuzzy switching median filter (NAFSM), and BPDF. The results show that BPDF produces better results than the above-mentioned methods at low and medium noise density.

Key words: Noise removal, salt and pepper noise, image denoising, impulse noise

1. Introduction

One of the most important issues in image processing is removing noise from images by preserving their details and features such as edges, textures, and colors [1–7]. Hence, there has been much research on this subject [8]. The success of image denoising affects the success rate of segmentation, classification, and similar procedures.

When the images are captured, some disruption occurs in the pixels during the digitalization process of the image. Additionally, vibrations may occur on the sensors during the imaging process [2–6,9,10]. This deterioration is classified as salt and pepper noise (SPN) and random valued impulse noise [11,12].

SPN degrades the image quality to a great extent [13]. Therefore, many linear/nonlinear filters have been developed to solve the problem. Several of these filters can only be applied to noisy pixels [3,10,14,15], whereas others work on all pixels [16]. In general, the nonlinear filters with fixed/adaptive window size, such as median, average, mean, and adaptive Wiener filters, are stronger than linear filters. To set the new value of a pixel, these filters use a window consisting of the neighboring pixels of the noisy pixel accepted as the center pixel. The most familiar filter is the standard median filter (SMF) [3,17,18]. SMF works on all pixels, not only on noisy ones. Applying the filter in this way, however, blurs the image and distorts the original pixel values. Although SMF works well in low-intensity noise by using the small window size, it does not work well in other environments, which is a disadvantage for SMF [19,20].

*Correspondence: uerkan80@gmail.com

Recently, certain filters, such as adaptive median filter (AMF) and adaptive fuzzy filter combined with median filter (AFM), have been using the adaptive window size instead of the fixed one. These filters adjust the window size adaptively according to the noise density [21–25]. In this filter, the window size is not fixed as in SMF; instead, it changes according to changing conditions [26]. A disadvantage of the AMF method is that when the window size is enlarged, it moves away from the original pixel information. AFM is a three-step method: firstly, the noisy pixel of the image is determined, secondly, the noise is removed by using the adaptive fuzzy filter, and finally, SMF is applied to the remaining noisy pixels from the second step [27].

Several other popular methods are progressive switching median filter (PSMF), decision-based algorithm (DBA), modified decision-based unsymmetrical trimmed median filter (MDBUTMF), and noise adaptive fuzzy switching median filter (NAFSM). PSMF is one of the most successful median filter types in removing noise. The image is made locally smooth and the edges are removed. The location of the noise is determined and then SMF is applied locally to the image [28]. PSMF produces better results than SMF. However, because of PSMF's recursive feature, high computational complexity becomes a disadvantage [29]. DBA emerges from the changes to SMF. The filter determines the noisy pixels and applies SMF to them [30]. In MDBUTMF, the noisy pixel value is replaced with the average value of all pixels in the window accepting this pixel as the center pixel. In this case, even if all pixel values in the window are 0 or 255, the filter assumes that these pixels are not corrupted. This is an inefficient way of reaching the value close to the original pixel value [31,32]. NAFSM removes noise in two stages. Firstly, the noisy pixels are identified by the histogram of the corrupted image. Secondly, the filter is applied to the noisy pixels [33]. Fuzzy reasoning is employed to handle uncertainty in the extracted local information about the noisy pixels, and the values of the corrupted pixels are replaced with a median filter or are estimated via the values of the neighboring pixels [34,35].

In this study, we developed a method called based on pixel density filter (BPDF) to remove SPN. The basic idea in this new method is that the most repetitive pixel value among the uncorrupted neighboring pixels of the noisy pixel is set as the new value.

2. Proposed algorithm

In this section, we develop the BPDF method to remove SPN. Determining whether a pixel with value 0 (or 255) is noisy or not plays an important role in noise removal. To that end, BPDF assumes that all pixels with value 0 (or 255) are not noisy, on the condition that the uncorrupted pixel values, if any, are less than 10 (or higher than 245) in the window. Of course, the threshold value determined as 10 can change according to the noise density of the images. Furthermore, the most crucial feature of this method is that if all the pixels in the window are noisy, this increases the window size until an uncorrupted pixel is reached.

The BPDF algorithm is as follows:

Let $X := [x(i,j)]$ be a noisy image consisting of pixels $x(i,j)$, where i and j range from 1 to m and n, respectively.

Step 1. For all i and j,

Step 1.1. If x(i,j) is noisy, and at least one pixel in the window with a 3 $\times$ 3 size that accepts this pixel as the center pixel is not noisy, then

For all i and j in the window, if there is at least one $x(i,j)$ such that $0 < x(i,j) < 10$ or $245 < x(i,j) < 255$, then

a. Find the maximum repetitive pixel values in the window;

b. Evaluate the median of the values;

c. Overwrite this value to the $x(i,j)$.

Step 1.2. If $x(i,j)$ is noisy, and at least one pixel in the window with 5 $\times$ 5 size that accepts this pixel as the center pixel is not noisy, then

For all i and j in the window, if there exists at least one $x(i,j)$ such that $0 < x(i,j) < 10$ or $245 < x(i,j) < 255$, then

a. Find the maximum repetitive pixel values in the window;

b. Evaluate the median of the values;

c. Overwrite this value to $x(i,j)$.

$\vdots$

Step 1.3. If $x(i,j)$ is noisy, and at least one pixel in the window with (2k +1) $\times$ (2k + 1) size that accepts this pixel as the center pixel is not noisy, then

For all i and j in the window, if there exists at least one $x(i,j)$ such that $0 < x(i,j) < 10$ or $245 < x(i,j) < 255$, then

a. Find the maximum repetitive pixel values in the window;

b. Evaluate the median of the values;

c. Overwrite this value to the $x(i,j)$.

where $0 < \mathrm{k} < \min\{\mathrm{m,n}\}$.

Step 2. Otherwise, keep the value of $x(i,j)$.

Example 2.1 *Assume that X is a noisy image with a size of 512 $\times$ 512. Let x (40,41) = 0, and the window with the 3 $\times$ 3 size shown in Figure 1a be the window that accepts this pixel as the center pixel. At least one pixel in this window is not noisy; for example, x (39,41) $\neq$ 0 (or x (39,41) $\neq$ 255) since x (39,41) = 236. Similarly, at least one pixel value in this window is between 245 and 255; for example, 245 < x (40,40) = 252 < 255. Therefore, the window satisfies the conditions given in Step 1.1. In that case, the maximum repetitive pixel values in the window are found as 240 and 244, and the median value of these values is evaluated as 242. Therefore, the value of 242 is set to the noisy pixel, and the window becomes as in Figure 1b.*

Example 2.2 *Assume that X is a noisy image with a size of 512 $\times$ 512. Let x (314,350) = 255 and the window with the 3 $\times$ 3 size shown in Figure 2a be the window that accepts this pixel as the center pixel. All pixels in this window are noisy. Therefore, the window does not satisfy the conditions given in Step 1.1. Let the window with the 5 $\times$ 5 size, shown in Figure 2b, accept the pixel x (314,350) as the center pixel. Clearly, at least one pixel in this window is not noisy; for example, x (312,350) $\neq$ 0 (or x (312,350) $\neq$ 255), since x (312,350) = 8. Similarly, at least one pixel value in this window is between 0 and 10; for example, 0 < x (314,348) = 8*

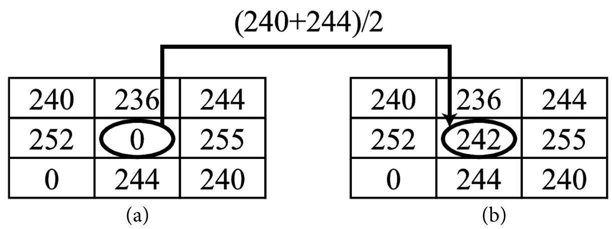

Figure 1. Illustration of Example 2.1.

$<$ 10. That is, the window satisfies the conditions given in Step 1.2. In that case, the maximum repetitive pixel values in the window are found as 0 and 8, and the median value of these is evaluated as 4. Therefore, value 4 is set to the noisy pixel, and the window becomes as is shown in Figure 2c.

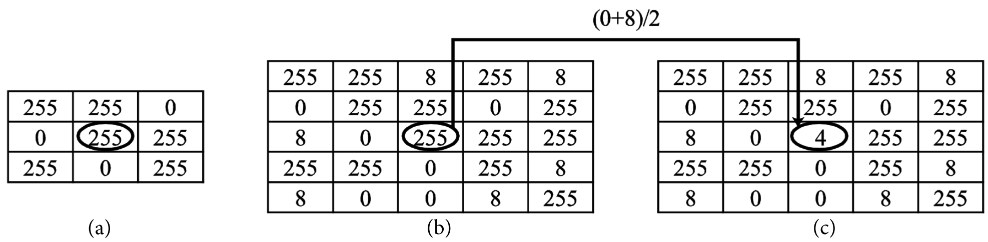

Figure 2. Illustration of Example 2.2

3. Algorithm results

3.1. Algorithm evaluation criteria

In this subsection, we compare the results of AMF, SMF, AFM, PSMF, DBA, MDBUTMF, NAFSM, and BPDF methods, by using 18 test images with a size of 512 $\times$ 512 and three metrics. The images are "Lena", "Cameraman", "Barbara", "Peppers", "Plane", "Baboon", "Bridge", "Pirate", "Elaine", "Boat", "Lake", "Flintstones", "Living Room", "Blonde Woman", "Dark-haired Woman", "House", "Parrot", and "Flower".

The first of these metrics, PSNR, is commonly used in literature and is defined as

$$PSNR := 10 \log \left(\frac{255^2}{MSE} \right), \tag{1}$$

where MSE stands for the mean square error, and is defined as

$$MSE := \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(e(i,j) - f(i,j)\right)^2 \tag{2}$$

Here $E := [e(i,j)]$ is the earliest form/original image and $F := [f(i,j)]$ is the final form/corrupted image.

The second of these metrics, SSIM, which is given in [36], is simplified and defined as

$$SSIM := \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right) + \left(\sigma_x^2 + \sigma_y^2 + C_2\right)}, \tag{3}$$

where μ_x, μ_y, σ_x, σ_y, and σ_{xy} are the average intensities, standard deviations and cross-covariance for images x and y, respectively. In addition, $C_1 := (K_1L)^2$ and $C_2 := (K_2L)^2$ are two constants, such that $K_1K_2 \ll 1$ are small constants and $L = 255$ for eight-bit grayscale images.

The last of these metrics, simplified. image enhancement factor (IEF), given in [32], is defined as

$$IEF := \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(x(i,j)-e(i,j))^2}{\sum_{i=1}^{m}\sum_{j=1}^{n}(f(i,j)-e(i,j))^2} \tag{4}$$

Here $E := [e(i,j)]$ is the earliest form/original image, $F := [f(i,j)]$ is the final form/corrupted image, and $X := [x(i,j)]$ is the noise image.

3.2. Algorithm evaluation results

Table 1 shows the mean PSNR-SSIM-IEF results of 18 test images. In Table 1, the noise densities (NDs) range from 10% to 50%. According to Table 1, BPDF is the most successful method in the 10% and 20% ND, whereas it is the second most successful method in the 30% and 40% ND. As a result, it is safe to state that BPDF is the most successful method when the ND is low.

Table 1. Mean results of the methods for the eighteen test images.

ND	Evaluation	AMF	SMF	AFM	PSMF	DBA	MDBUTMF	NAFSM	BPDF
10%	PSNR	29.16	31.09	29.48	33.99	35.46	33.78	36.46	37.55
	SSIM	0.9503	0.8936	0.8527	0.9664	0.9709	0.9458	0.9768	0.9815
	IEF	25	49	37	92	159	89	240	251
20%	PSNR	26.35	27.30	28.81	30.22	31.23	30.20	33.47	33.76
	SSIM	0.9374	0.8481	0.8382	0.9343	0.9287	0.8574	0.9532	0.9585
	IEF	26	34	62	69	113	68	230	201
30%	PSNR	24.55	22.72	28.52	26.76	28.05	28.68	31.62	31.06
	SSIM	0.9173	0.7132	0.8336	0.8615	0.8696	0.7998	0.9282	0.9289
	IEF	26	17	84	45	78	72	220	158
40%	PSNR	23.22	18.47	28.12	23.58	25.24	28.84	30.21	28.71
	SSIM	0.8907	0.4900	0.8281	0.7501	0.7914	0.8243	0.9014	0.8897
	IEF	25	8	97	28	51	106	207	119
50%	PSNR	22.06	14.93	27.08	19.24	22.67	29.06	29.01	26.40
	SSIM	0.8559	0.2750	0.8036	0.5294	0.6957	0.8569	0.8715	0.8354
	IEF	24	5	87	13	33	164	191	85

In Table 2, the PSNR-SSIM-IEF results of the methods for 18 test images with 20% ND are shown. According to Table 2, BPDF produced the best results in 13 images for each metric.

Figure 3 shows the cameraman images of DBA, MDBUTMF, NAFSM, and BPDF in 10% noise density as well as the PSNR-SSIM-IEF results. BPDF gives the best results in 10% ND.

Figure 4 shows the Lena images, where BPDF is applied to various NDs. BPDF produces outstanding results in low and medium ND.

Table 2. Results of the methods for the eighteen test images in 20% ND.

	Evalution	AMF	SMF	AFM	PSMF	DBA	MDBUTMF	NAFSM	BPDF
Lena	PSNR	27.04	28.95	30.91	32.64	33.38	32.17	35.66	36.10
	SSIM	0.9439	0.8656	0.8734	0.9627	0.9422	0.8693	0.9667	0.9659
	IEF	29	44	70	104	123	93	208	230
Cameraman	PSNR	26.96	28.36	30.77	31.21	32.58	30.94	33.91	35.07
	SSIM	0.9697	0.9175	0.9351	0.9443	0.9659	0.8391	0.9643	0.9798
	IEF	31	43	74	82	113	77	153	199
Barbara	PSNR	24.54	23.37	24.03	24.93	26.62	27.24	30.18	29.49
	SSIM	0.9073	0.7495	0.7425	0.8466	0.8944	0.8454	0.9482	0.9430
	IEF	17	13	15	18	27	31	61	52
Peppers	PSNR	26.35	26.15	27.15	29.32	28.92	28.58	29.47	31.56
	SSIM	0.9430	0.8469	0.8443	0.9421	0.9134	0.8785	0.9212	0.9511
	IEF	23	22	28	46	42	39	48	78
Plane	PSNR	26.56	28.01	29.73	31.08	32.71	31.66	36.36	34.78
	SSIM	0.8962	0.7839	0.7720	0.9209	0.9098	0.8458	0.9553	0.9467
	IEF	27	37	55	75	109	86	254	176
Baboon	PSNR	25.37	24.58	24.96	27.06	27.39	27.91	29.63	30.25
	SSIM	0.9058	0.7581	0.7311	0.8911	0.8820	0.8614	0.9230	0.9350
	IEF	20	17	19	30	32	37	54	63
Bridge	PSNR	26.62	27.60	28.41	31.18	31.18	30.70	33.60	33.84
	SSIM	0.9338	0.8324	0.8209	0.9495	0.9220	0.8602	0.9512	0.9548
	IEF	26	33	40	75	75	67	131	139
Pirate	PSNR	26.77	28.07	30.78	32.25	33.60	32.55	37.56	36.05
	SSIM	0.8884	0.7351	0.7449	0.9250	0.8969	0.8554	0.9542	0.9402
	IEF	27	37	68	96	130	102	325	229
Elaine	PSNR	26.27	26.48	27.14	29.32	29.86	29.79	31.52	32.43
	SSIM	0.9175	0.7997	0.7811	0.9249	0.9052	0.8714	0.9384	0.9448
	IEF	24	25	29	48	54	53	79	98
Boat	PSNR	26.47	27.64	30.00	29.48	31.91	30.32	33.58	34.63
	SSIM	0.9616	0.9052	0.9124	0.9130	0.9558	0.8177	0.9685	0.9736
	IEF	28	37	63	56	98	68	144	184
Lake	PSNR	26.98	29.74	30.92	32.06	36.87	33.15	41.44	39.75
	SSIM	0.9770	0.9357	0.8656	0.9561	0.9777	0.8444	0.9828	0.9854
	IEF	30	56	73	96	289	123	828	562
Flintstones	PSNR	26.23	27.02	27.95	29.64	29.36	29.23	31.54	32.56
	SSIM	0.9307	0.8292	0.8302	0.9136	0.9195	0.8346	0.9501	0.9542
	IEF	26	32	39	58	54	53	89	113
Living room	PSNR	26.27	26.75	27.58	29.50	29.99	29.97	31.81	32.51
	SSIM	0.9121	0.7891	0.7878	0.9152	0.9057	0.8624	0.9417	0.9447
	IEF	24	27	33	51	57	57	87	102
Blonde woman	PSNR	24.31	23.45	22.78	25.18	25.00	25.88	26.39	27.45
	SSIM	0.9101	0.8100	0.7612	0.9070	0.8988	0.8523	0.9330	0.9414
	IEF	18	15	13	22	21	26	29	38
Dark-haired woman	PSNR	25.72	25.39	25.76	28.46	27.14	28.06	28.72	29.45
	SSIM	0.9612	0.9025	0.8876	0.9592	0.9303	0.9288	0.9442	0.9574
	IEF	22	20	22	41	30	37	43	51
House	PSNR	26.54	28.18	28.12	31.19	29.83	29.54	30.68	31.58
	SSIM	0.9553	0.8882	0.8762	0.9537	0.9423	0.8858	0.9632	0.9671
	IEF	25	36	36	73	53	50	65	80
Parrot	PSNR	26.06	27.13	27.06	30.45	28.95	28.55	29.47	30.57
	SSIM	0.9126	0.7940	0.7897	0.9418	0.9132	0.8605	0.9464	0.9470
	IEF	22	28	28	61	43	39	48	62
Flower	PSNR	27.05	29.83	36.74	33.26	38.77	32.53	42.04	40.96
	SSIM	0.9660	0.9117	0.9365	0.9517	0.9692	0.8166	0.9814	0.9802
	IEF	32	60	295	133	472	112	1000	780

Figure 3. Cameraman image filter results: (a) original image, (b) image with 10% ND, (c) image after DBA, (d) image after MDBUTMF, (e) image after NAFSM, (h) image after BPDF.

Figure 4. Lena images with noise, and Lena images after noise removal: (a) 1% cleaned, (b) 10% cleaned, (c) 30% cleaned.

For the NDs that range from 1% to 9%, Table 3 shows the mean results of the methods for all images. The results show that BPDF is more successful than other methods at very low NDs.

The time performances of the methods have been measured in seconds and are shown in Table 4. In comparison, the computer that has been used has Windows 7, 64-bit, Intel Core i7-4720 CPU @ 2.60 GHz, 16 GB memory, and MATLAB R2016b. The mean results show that BPDF is faster than its most powerful competitors, MDBUTMF and NAFSM.

Table 3. Mean results of all methods at very low ND.

ND	Evaluation	AMF	SMF	AFM	PSMF	DBA	MDBUTMF	NAFSM	BPDF
1%	PSNR	34.98	33.51	31.56	38.49	44.28	45.13	45.62	47.41
	SSIM	0.9527	0.9064	0.8899	0.9861	0.9976	0.9967	0.9975	0.9982
	IEF	11	13	7	54	195	208	222	274
2%	PSNR	33.70	33.30	31.17	37.80	42.29	42.24	43.03	44.66
	SSIM	0.9517	0.9054	0.8840	0.9841	0.9951	0.9935	0.9954	0.9966
	IEF	15	25	12	70	201	203	230	283
3%	PSNR	32.77	33.04	30.82	37.34	40.90	40.54	41.39	43.01
	SSIM	0.9513	0.9042	0.8786	0.9821	0.9924	0.9901	0.9930	0.9947
	IEF	18	31	16	90	208	206	236	288
4%	PSNR	32.03	32.88	30.55	36.81	39.78	39.36	40.22	41.81
	SSIM	0.9516	0.9031	0.8738	0.9801	0.9898	0.9868	0.9907	0.9930
	IEF	20	40	20	93	202	203	239	294
5%	PSNR	31.31	32.55	30.31	36.39	38.77	38.40	39.38	40.76
	SSIM	0.9514	0.9018	0.8692	0.9781	0.9869	0.9833	0.9884	0.9911
	IEF	21	42	23	102	184	192	241	278
6%	PSNR	30.83	32.32	30.08	35.81	37.86	37.47	38.58	39.98
	SSIM	0.9514	0.9003	0.8652	0.9759	0.9839	0.9793	0.9861	0.9894
	IEF	22	48	26	102	169	176	244	273
7%	PSNR	30.45	31.86	29.88	35.31	37.29	36.54	37.93	39.29
	SSIM	0.9518	0.8987	0.8612	0.9737	0.9809	0.9739	0.9837	0.9874
	IEF	24	44	29	100	177	149	238	274
8%	PSNR	29.94	31.67	29.73	34.86	36.63	35.72	37.42	38.66
	SSIM	0.9514	0.8971	0.8582	0.9712	0.9778	0.9674	0.9816	0.9855
	IEF	24	47	31	96	171	131	245	265
9%	PSNR	29.53	31.40	29.59	34.37	36.08	34.74	36.86	38.02
	SSIM	0.9506	0.8951	0.8554	0.9686	0.9745	0.9579	0.9791	0.9836
	IEF	24	50	34	93	170	106	236	255

Table 4. Time comparisons of the methods (in seconds).

ND	AMF	SMF	AFM	PSMF	DBA	MDBUTMF	NAFSM	BPDF
10%	4.09	3.88	16.31	1.58	12.04	12.42	4.35	3.33
20%	4.03	4.14	15.54	1.81	10.57	13.85	8.97	6.73
30%	4.12	4.17	16.02	2.02	12.39	18.98	13.53	9.84
40%	3.90	4.19	10.42	1.38	11.10	22.22	16.45	12.93
50%	5.13	4.63	15.56	2.48	12.20	29.98	24.54	18.51
60%	4.52	4.15	17.17	0.68	11.75	29.86	25.05	19.83
70%	8.23	4.30	15.06	0.87	11.85	36.64	33.46	23.76
80%	16.22	4.41	15.13	0.85	11.99	36.02	37.49	27.26
90%	35.73	4.35	14.90	0.55	11.50	37.81	39.56	30.24
Mean	9.55	4.25	15.12	1.36	11.71	26.42	22.60	16.94

4. Conclusion

BPDF removed the salt and pepper noise by looking at the repeat numbers of the pixels, and succeeded more than the others in ND up to 50%. This success is more distinct when ND is low. When processing noisy pixels, BPDF does not remove those pixels from the window via a threshold value, but takes them into account if there is any doubt about any neighboring pixels of the processed pixel being noiseless. Each filter has its unique difficulties and successes. Perhaps it is best is to define a hybrid filter that determines the NDs of images and uses the most successful filter for each ND slice.

References

[1] Zhang X, Ding F, Tang Z, Yu C. SPN removal with image in painting. Int J Electron Commun 2015; 69: 307-313.

[2] González-Hidalgo M, Massanet S, Mir A, Ruiz-Aguilera D. A fuzzy filter for high-density SPN removal. Lect Notes Comp Sci 2013; 8109: 70-79.

[3] Xiao L, Li C, Wu Z, Wang T. An enhancement method for X-ray image via fuzzy noise removal and homomorphic filtering. Neurocomputing 2016; 64: 195: 56-64.

[4] Coupe P, Manjon JV, Robles M, Collins DL. Adaptive multiresolution non-local means filter for three-dimensional magnetic resonance image denoising. IET Image Process 2012; 6: 558-568.

[5] Baljozovic D, Kovacevic B, Baljozovic A. Mixed noise removal filter for multi-channel images based on halfspace deepest location. IET Imag Proc 2013; 7: 310-323.

[6] Vijendran AS, Lukose B. Fast and efficient method for image denoising. IJEIT 2013; 3: 200-208.

[7] Sakthidasan K, Sankaran A, Nagappan VN. Noise free image restoration using hybrid filter with adaptive genetic algorithm. Comput Elec Eng 2016; 54: 382-392.

[8] Thanha DNH, Dvoenkoa SD. A method of total variation to remove the mixed Poisson–Gaussian noise. Pattern Recogn 2016; 26: 285-293.

[9] Zhang C, Wang K. A switching median–mean filter for removal of high-density impulse noise from digital images. Optik 2015; 126: 956-961.

[10] Gellert A, Brad R. Context-based prediction filtering of impulse noise images. IET Image Process 2016; 10: 429-437.

[11] Vasanth K, Kumar VJS. Decision-based neighborhood-referred unsymmetrical trimmed variants filter for the removal of high-density salt-and-pepper noise in images and videos. Signal Image Video P 2015; 9: 1833-1841.

[12] Chan RH, Ho CW, Nikolova M. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. IEEE T Image Process 2005; 14: 1479-1485.

[13] Zhao F, Ma RC, Ma JQ. An algorithm for SPN removal based on information entropy. Appl Mech Mater 2012; 220–223: 2273-2279.

[14] Erkan U, Kilicman A. Two new methods for removing salt-and-pepper noise from digital images. ScienceAsia 2016; 42: 28-32.

[15] Roig B, Estruch VD. Localised rank-ordered differences vector filter for suppression of high-density impulse noise in colour images. IET Image Process 2016; 10: 24-33.

[16] Hwang H, Haddad RA. Adaptive median filters: new algorithms and results. IEEE T Image Process 1995; 4: 499-502.

[17] Jin L, Xiong C, Liu H. Improved bilateral filter for suppressing mixed noise in color images. Digit Signal Process 2012; 22: 903-912.

[18] Sreenivasulu P, Chaitanya NK. Removal of SPN for various images using median filters: a comparative study. IUP J Telecommunications 2014; VI.

[19] Sun C, Tang C, Zhu X, Li X, Wang L. An efficient method for salt-and-pepper noise removal based on shearlet transform and noise detection. Int J Electron Commun. 2015; 69: 1823-1832.

[20] Xiao Y, Zeng T, Yu J, Ng MK. Restoration of images corrupted by mixed Gaussian-impulse noise via l1–l0 minimization. Pattern Recogn 2011; 44: 1708-1720.

[21] Astola J, Kuosmanen P. Fundamentals of Nonlinear Digital Filtering. Boca Raton, FL, USA: CRC Press, 1997.

[22] Tsirikolias K. Low level image processing and analysis using radius filters. Digit Signal Process 2016; 50: 72-83.

[23] Yin L, Yang R, Gabbouj M, Neuvo Y. Weighted median filters: a tutorial. IEEE T Circuits Syst 1996; 43: 157-192.

[24] Arce G. A general weighted median filter structure admitting negative weights. IEEE T Signal Proc 1998; 46: 3195-3205.

[25] Arce G, Paredes J. Recursive weighted median filters admitting negative weights and their optimization. IEEE T Signal Proc 2000; 48: 768-779.

[26] Pattnaik A, Agarwal S, Chand S. A new and efficient method for removal of high density SPN through cascade decision-based filtering algorithm. Procedia Technol 2012; 6: 108-117.

[27] Palabaş T, Gangal A. Adaptive fuzzy filter combined with median filter for reducing intensive SPN in gray level images. In: IEEE 2012 International Symposium on Innovations in Intelligent Systems and Applications; 2–4 July 2012; Trabzon, Turkey. New York, NY, USA: IEEE. pp. 1-4.

[28] Shrestha S. Image denoising using new adaptive based medıan filter. Signal &. Image Processing: An International Journal 2014; 5: 1-13.

[29] Wang Z, Zhang D. Progressive switching median filter for the removal of impulse noise from highly corrupted images. IEEE T Circuit Syst 1999; 46: 78-80.

[30] Srinivasan KS, Ebenezer D. A new fast and efficient decision-based algorithm for removal of high density impulse noises. IEEE Signal Process 2007; 14: 189-192.

[31] Kesharwani A, Agrawal S, Dhariwal MK. An improved decision based asymmetric trimmed median filter for removal of high density SPN. Int J Comput Appl 2013; 84: 37-43.

[32] Esakkirajan S, Veerakumar T, Subramanyam AN, Prem Chand CH. Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter. IEEE Signal Process 2011; 18: 287-290.

[33] Vasanth K, Senthilkumar VJ. A decision based unsymmetrical trimmed midpoint algorithm for the removal of high density SPN. J Theor Appl Inf Technol 2012; 42: 245-252.

[34] Toh KKV, Isa NAM. Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction. IEEE Signal Process 2010; 17: 281-284.

[35] Zhang P, Li F. A new adaptive weighted mean filter for removing salt-and-pepper noise. IEEE Signal Process 2014; 21: 1280-1283.

[36] Zhou W, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE T Image Process 2004; 13: 600-612.