A Decision Based Unsymmetrical Trimmed Modified Winsorized Geometric Mean for the Removal of High Density Salt and Pepper Noise in Images

Sahithi K[1] , Ravalika B[1] , Triveni A[1] , Rishitha K [1], Vasanth K[1] , Marish Kumar P[2], Srinivasan K S [3]

[1]VIdya Jyothi Institute of Technology, Aziz Nagar, Chilukur Road, Hyderabad – 500075, India
[2]Easwari Engineering College, Chennai, India.
[3]Mohamed Sathak AJ college of Engineering, Chennai, India

Abstract. A Decision Based Unsymmetrical trimmed modified Winsorized Geometric mean is proposed for heavily corrupted images. The proposed Algorithm uses a Fixed 3x3 window for increasing noise densities. The algorithm replaces the corrupted pixel with mean of the 4 neighbours or Unsymmetrical trimmed Modified Geometric mean or mean of the window. The use of modified Winsorized Geometric mean provided a better noise elimination characteristic than many Existing algorithms. The Algorithm Exhibits very good noise eliminating capability with improved edge preservation even at very high noise densities.

1. Introduction

Salt and Pepper noise (SPN) appears as randomly scattered pixel value that hold white (255) or black (0) or both pixels over the image. The SPN gets corrupted due to error in transmission. A use of linear filters blurs the image. But the use of Non Linear image removes SPN and preserves information of the image. At High Noise densities filters such as Simple median filter fails to remove the faulty noise pixel. The Noise Model of Salt and Pepper noise is given below. The noise model for salt and pepper noise of Fixed Valued Impulse noise for an 8 bit image is given as follows. Consider [0; 255] denote the dynamic range of the corrupted image y', i.e., 0 <= y'ij <= 255 for all (i,j), then they are denoted by Salt-and-pepper noise: the gray level of y at pixel location (i j) is yij = 0; with probability p;

255; with probability q;

y'ij; with probability 1 - p - q;

Where s = p + q denotes the salt-and-pepper noise level and made 0 if uncorrupted. I(x, y) is the observed intensity in the corrupted region. Many Researchers had contributed work that eliminates SPN few are discussed here. Vasanth et al [1] proposed a modified Winsorized mean that eliminated SPN at very high noise densities. The Algorithm produces Image artifact at higher noise densities. Fang et al [2] used L0 Norm for finding the closeness of image after restoring. Pulse Coupled Neural Network [3] was used to identify uncorrupted pixels from the corrupted image. Later median of uncorrupted pixel is used to replace the faulty pixel. An adaptive Neutrosophic filter was used to eliminate the indeterminacy of pixel at high noise densities. The Filter used similarity of the pixels. An Adaptive filter [4] that used distribution ratios of three groups to weight the non extreme pixels using the maximum, middle, and minimum pixel values of a closed vicinity. The faulty pixel with an extreme value is replaced by the weighted value, thus enabling the noisy pixels to be restored. An Adaptive algorithm that uses local edge-preserving function solved using Poly-Ribiere-Polak (PRP) method [5]. An Algorithm that modified the conventional Winsorized operation was proposed. The algorithm used Modified Winsorized median [6] to eliminate high density SPN in images. An Improved Algorithm that uses asymmetrically trimmed modified Winsorized Median or Mean (Winsorized variants) depends on the pixel information of a local window [7]. An Iterative two-stage algorithm that used multilevel weighted graphs model for image representation followed by an operator to determine the order-inducing variables and weighted vectors of the Induced Generalized Order Weighted Average (IGOWA) to restore the detected noise candidate [8]. A Novel methodology for the detection of Random valued impulse noise was introduced using three levels of thresholds. A noise signature is calculated for every pixel and compared with the threshold to identify noise followed by the comparison of the central pixel with the second and third levels of thresholds [9]. An Adaptive Riesz mean filter used pixel similarity for removing the faulty pixels of an image [10]. Sheperds interpolation algorithm was used to remove high density salt and pepper noise [11]. Most algorithms fails to remove noise at very high noise densities, a few induces image artifacts while removing. Hence a suitable filter that removes the faulty pixels with high degree of information preservation is desired for subsequent stages of image processing. The paper is organized as follows Section 2 gives the Decision based Unsymmetrical Trimmed Modified Winsorized Geometric Mean proposed algorithm (PA). Section 3 deals with the Simulation Results and Discussions. Section 4 gives the concluding remark of the work.

2. Unsymmetrical Trimmed Modified Winsorized Operation

2.1. t Modified Winsorized Operation

A Conventional Winsorized operation on a sorted Data replaces “Z” smaller and largest value of a sorted Array. This is mainly used to eliminate outliers in a given data set. An Unsymmetrical trimmed Modified Winsorized Operation of a Sorted Data trims the outliers in an unsymmetrical way and replaces the small and large value of the sorted Array. Even if the outliers are more only the smaller and larger value of the trimmed array is replaced. Using the above operation, series of operations were derived such as Modified Winsorized Mean, Median and Mode respectively. These statistics would take a set of finite numbers from image vicinity and eliminate the outliers and perform Modified Winsorized Mean, Median and Mode respectively.

Why Unsymmetrical Trimmed Modified Winsorized variants (Mean and Median) perform better on Natural images?

The Pixel values of the images have a slim distribution (i.e., they lie closer to each other). Only when there is a bigger distribution then the conventional Winsorized operation fails. Hence to make a statistic that is robust to outliers a modified Winsorized operation is proposed. This Modified operation uses only non outlier values to compute Mean, Median or Mode. Hence the proposed modified Winsorized operation was found to work better in Natural Images.

2.2. Modified Winsorized Geometric Mean Operation

The Modified Winsorized Geometric mean is evaluated by sorting the data acquired from small vicinity from different parts of the images. The outliers were eliminated and Geometric mean of the trimmed array is evaluated. The value of the Geometric mean is better than the Arithmetic mean for a small slim distribution. The proposed static has better robust characteristic than conventional Arithmetic Mean. Hence it is decided to apply this statistic on the image with few changes.

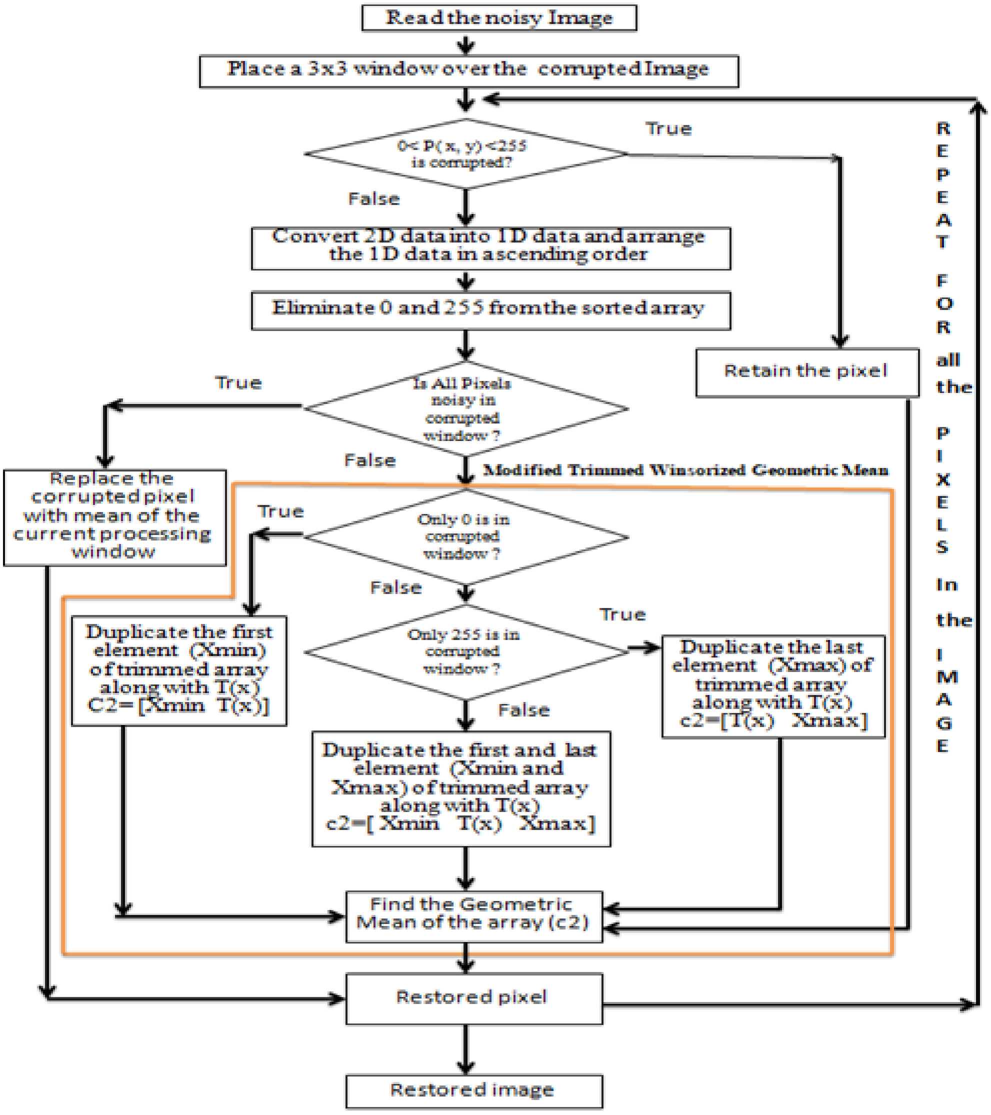


Figure 1 Flowchart of the Proposed Algorithm

2.3. Decision based Unsymmetrical Trimmed Modified Winsorized Geometric Mean proposed algorithm (Proposed Algorithm)

The Unsymmetrical trimmed Modified Geometric mean will not eliminate high density salt and pepper noise. Few conditions have been added to make it usable for the removal of high density salt and pepper noise. The Flowchart of the Decision based Unsymmetrical trimmed Modified Geometric mean is given in figure 1.

2.4. Methodology of the Proposed Algorithm

This Section deals with the insight of the Decision based modified trimmed Winsorized geometric mean (proposed algorithm). The Bigger matrix is the pixel values derived from a corrupted image. The algorithm operates with 3x3 windows for increasing noise densities. A Smaller rectangle refers to this fixed 3x3 window that moves over every pixel of the image. An oval shape represents the pixel that is currently processed. The Algorithm operates in six different cases.

Case 1: The processed pixel is not noisy and 4 neighbors are not noisy:

$$\begin{pmatrix} 159 & 161 & 160 & 0 & 255 \\ 160 & 161 & 160 & 255 & 73 \\ 159 & 160 & 159 & 160 & 80 \\ 46 & 255 & 158 & 160 & 81 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix} \Rightarrow \begin{pmatrix} 159 & 161 & 160 & 0 & 255 \\ 160 & 161 & 160 & 255 & 73 \\ 159 & 160 & 159 & 160 & 80 \\ 46 & 255 & 158 & 160 & 81 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix}$$

Faulty image segment De-noised Image segment

Figure 2 Illustration of Case 1

In this case the processed pixel is between 0 and 255 and hence the processed pixel is considered as not corrupted as shown in figure 2.

Case 2: The processed pixel is noisy and the 4 neighbors are noisy: In this case the current processing pixel is 255.

The detector of the algorithm will compare the current functioning pixel with 0 and 255. The operation pixel is named altered. Now detectors check the 4 neighbors of the current processing pixel for 0 and 255. In this case all the 4 neighbors are either 0 or 255. Hence it is considered as noisy. Now find the mean of four neighbors and replace in the place of the current processing pixel.

4 Neighbors of current processing pixel: 255, 255, 255, 0 (All the values are clattered)

Observe the mean of the 4 Neighbors: (255+255+255+0)/4 =191.25=192. The value 192 is replaced in the place of the corrupted pixel as shown in figure 3.

Case 3: The processed pixel is noisy; 4 neighbors are not noisy; both 0 and 255 pixels are present in the current processing window:

In this case the Current processing pixel is 0. Hence the detector deems it as a corrupted pixel. Now the detector checks for the 4 neighbors as noisy. It was found that all 4 neighbors are not noisy. Now convert the 2D data into 1D data of the current functioning window. Classify

$$\begin{pmatrix} 159 & 161 & 160 & 0 & 255 \\ 160 & 161 & 160 & 255 & 73 \\ 159 & 160 & 159 & 255 & 80 \\ 46 & 255 & 255 & 255 & 0 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix} \Rightarrow \begin{pmatrix} 159 & 161 & 160 & 0 & 255 \\ 160 & 161 & 160 & 255 & 73 \\ 159 & 160 & 159 & 255 & 80 \\ 46 & 255 & 255 & 192 & 0 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix}$$

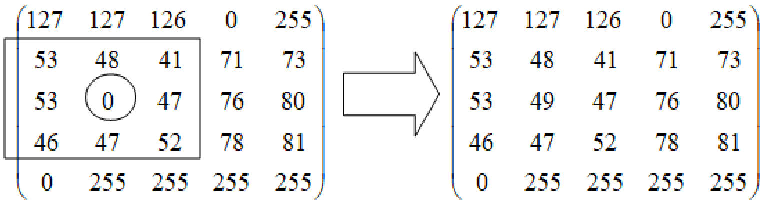
Faulty image segment De-noised Image segment

Figure 3 Illustration of Case 2

$$\begin{pmatrix} 127 & 127 & 126 & 0 & 255 \\ 127 & 0 & 255 & 71 & 73 \\ 127 & 0 & 73 & 0 & 80 \\ 0 & 255 & 75 & 78 & 81 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix} \Rightarrow \begin{pmatrix} 127 & 127 & 126 & 0 & 255 \\ 127 & 0 & 255 & 71 & 73 \\ 127 & 0 & 73 & 76 & 80 \\ 0 & 255 & 75 & 78 & 81 \\ 0 & 255 & 255 & 255 & 255 \end{pmatrix}$$

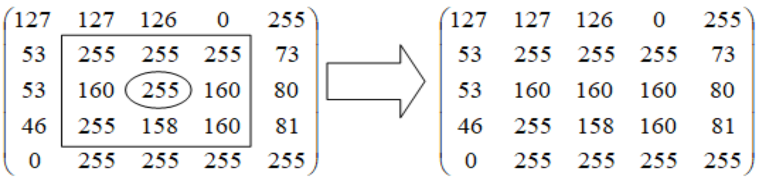
Faulty image segment De-noised Image segment

Figure 4 Illustration of Case 3



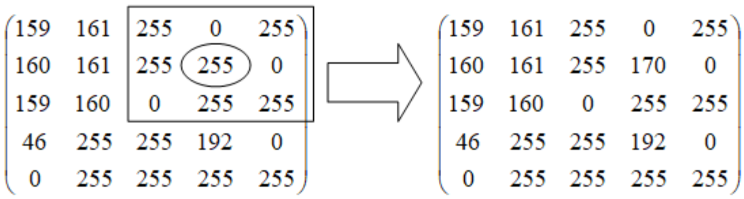
Faulty image segment De-noised Image segment

Figure 5 Illustration of Case 4



Faulty image segment De-noised Image segment

Figure 6 Illustration of Case 5



Faulty image segment De-noised Image segment

Figure 7 Illustration of Case 6

the data of the current functioning window in rise order. Now remove 0 and 255 from the pixels. Restore the least value and topmost value of the trimmed array as the first element and the last element. Now find the geometric mean of the data replaced array.

Converting 2D data into 1D data: 255 71 73 73 0 80 75 78 81

Arrange the Data In Increasing order:

0 71 73 73 75 78 80 81 255

Trimmed Array (eliminate 0 and 255 from the array):

71 73 73 75 78 80 81

Restore the least value and topmost value of the trimmed array: 71 71 73 73 75 78 80 81 81

observe the Geometric mean of the array (71, 71, 73, 73, 75, 78, 80, 81, 81) =75.78 =76. The calculated value 76 restores the corrupted pixel as shown in figure 4.

Case 4: The processed pixel is noisy; 4 neighbors are not noisy; only 0 is present in the current processing window:

In this case the Current processing pixel is 0. Hence the current functioning pixel is named altered. The 4 Nearest are examine for noise. In this case not all 4 nearest are noisy. In this case only 0 is establish in the array. Now change the 2D details into 1D details of the current functioning window. Sort the data of the current processing window in increasing order. Now eliminate 0 from the pixels. Replace the minimum value of the trimmed array as the first element. Now find the geometric mean of the data replaced array.

Geometric mean (41,41, 46,47,47,48,52,53,53) =48.21=49

Arrange the Data in Increasing order:

0 41 46 47 47 48 52 53 53

Trimmed Array (eliminate 0 from the array): 41 46 47 47 48 52 53 53

Restore the least value and topmost value of the trimmed array: 41 41 46 47 47 48 52 53 53

Observe the Geometric mean of the array (41 41 46 47 47 48 52 53 53) =48.21=49. The computed value 49 replaces the corrupted pixel as shown in figure 5.

Case 5: The processed pixel is noisy; 4 neighbors are not noisy; only 255 pixels are present in the current processing window:
In this case the Current processing pixel is 255. Hence the current processing pixel is termed corrupted. The 4 Nearest are examine for noise. In this case not all 4 neighbors are noisy and only a few pixels are considered noisy. In this case only 255 are found in the array. Now convert the 2D data into 1D data of the current processing window. Sort the data of the current processing window in increasing order. Now eliminate 255 from the pixels. Replace the maximum value of the trimmed array as the last element. Now find the geometric mean of the data replaced array.
Converting 2D data into 1D data:
255 255 255 160 255 160 255 158 160
Arrange the Data In Increasing order:
158 160 160 160 255 255 255 255 255
Trimmed Array (eliminate 0 from the array):
158 160 160 160
Replace the maximum value of the trimmed array:
158 160 160 160 160
Find the Geometric mean of the array (158 160 160 160 160) =159.59=160. The computed value 160 replaces the corrupted pixel as shown in figure 6.
Case 6: All the pixels of the current processing window is noisy: In this case the Current processing pixel is 255. Hence the current processing pixel is termed corrupted. All the elements of the current processing window are termed noisy. Find the mean of the current processing window and replace it in the place of the current processing window. Find the mean of the current processing window: (255+0+255+255+255+0+0+255+255) / 9 =170. The computed value 170 replaces the Corrupted pixel as shown in figure 7.

3. Simulation Results and Discussions

All the algorithms used in the paper were developed in MATLAB software on a 7 generation Intel i7processor. Images used in the paper are hosted in University of South California website [12]. Experiments were done by adding SPN to the image in increments of 10% and algorithm was applied to it . The Results were tabulated for different quantitative measures.

The Quantitative measures used in the work are Peak signal to noise ratio(PSNR), Image Enhancement Factor(IEF), Structural Similarity Index Metric (SSIM)[13] , Error rate (ER) and Pratts FOM [14] as given in equation 1,2,3,4,5,6.

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2}{MSE}\right) \qquad (1)$$

$$\text{MSE} = \frac{\sum_i \sum_j (a_{ij} - r_{ij})2}{row \times col} \qquad (2)$$

$$\text{IEF} = \frac{\left(\sum_i \sum_j f_{ij} - a_{ij}\right)^2}{\left(\sum_i \sum_j r_{ij} - a_{ij}\right)^2} \qquad (3)$$

$$\text{Error Rate} = \frac{[\#nos]}{row * col} * 100\% \qquad (4)$$

Table 1 Evaluation (PSNR) of algorithms on Lena Image corrupted by SPN

Noise In %	DBA	IDBA	EDBA	MDB MF	CUD BMPF	MD BUTMF	MDBUMF_ GM	ACBSA	DMF	DBATM WMF	DBATM WMeanF	DBATM WVF	DBATM WGM(PA)
10	39	39.45	41.29	45.2	32.3	43.1	45.3	41.9	37.42	44.5	44.4	44.56	44.27
20	36.8	37.26	39.83	41.5	32.1	41.2	41.6	38.8	33.73	40.92	41.1	40.98	40.58
30	35.8	35.56	38.30	38.8	31.8	37.9	38.8	37.1	31.08	38.47	38.8	38.49	38.61
40	33.2	34.05	36.48	36.5	31.4	36.4	36.5	35.5	28.81	36.25	37	36.86	36.77
50	31.4	32.49	34.70	34.4	31.1	34.3	34.53	33.8	26.13	34.37	35.5	35.30	35.40
60	29.6	31.56	33.40	32.1	30.3	32.1	32.1	30.3	23.18	32.05	33.9	33.76	33.82
70	27.8	30.28	31.62	29.6	30.2	29.6	29.73	29.8	20	29.63	32.1	32.26	31.927
80	25.5	28.03	28.84	26.5	29.3	26.8	28.78	27.2	14.37	26.86	29.9	30.44	29.853
90	21.8	24.02	24.60	22.1	27.4	22.4	22.36	26.6	12.52	22.37	26.7	27.73	26.642

Table 2. Evaluation (IEF) of algorithms on Lena Image corrupted by SPN

Noise in %	DBA	IDBA	EDBA	MDBMF	CUDBMPF	MDBUTMF	MDBUMF_GM	ACBSA	DMF	DBATMWMF	DBATMWMeanF	DBATMWVF	DBATMWGM(PA)
10	230.3	254.5	386	932	49.6	630.8	928	447	159.2	812.7	797.6	819.4	753.07
20	276.3	304	544	694.8	92.9	552.6	820	434	136.8	662.5	733.2	715.3	655.31
30	331.1	308	574	568.8	129.8	565.4	698	446	123.4	630.6	654.9	608.3	621.53
40	242.3	290.1	509	439.5	160.1	489.1	514	406	118.8	483.0	584.5	555.8	545.03
50	199.9	253.2	421	322.1	180.9	384.8	404	345	115.3	389.9	514	482.8	496.92
60	157.8	247.5	375	217	205.8	282.1	277	184	131.5	274.8	421.2	406.9	413.81
70	123.0	213.6	290	144.8	201.3	183.4	188	194	147.8	183.9	327.3	338.0	312.06
80	81.5	145.7	175	90.6	200.8	110.5	109	120	157.9	111.2	227.2	253.3	220.50
90	39.1	65	74.3	40.2	114.2	45.5	44	116	100.9	44.53	122.9	152.6	118.46

Table 3 Evaluation (SSIM) of algorithms on Lena Image corrupted by SPN

Noise in %	DBA	IDBA	EDBA	MDBMF	CUDBMPF	MDBUTMF	MDBUMF_GM	ACBSA	DMF	DBATMWMF	DBATMWMeanF	DBATMWVF	DBATMWGM(PA)
10	0.970	0.97	0.97	0.992	0.895	0.992	0.922	0.987	0.976	0.991	0.992	0.991	0.992
20	0.962	0.96	0.97	0.983	0.893	0.982	0.983	0.975	0.949	0.982	0.983	0.983	0.982
30	0.950	0.95	0.96	0.971	0.888	0.971	0.972	0.963	0.913	0.971	0.973	0.970	0.972
40	0.930	0.93	0.95	0.955	0.881	0.957	0.957	0.948	0.871	0.955	0.961	0.958	0.960
50	0.903	0.91	0.94	0.931	0.872	0.938	0.938	0.927	0.810	0.937	0.947	0.944	0.946
60	0.866	0.89	0.92	0.897	0.862	0.910	0.910	0.866	0.725	0.910	0.928	0.925	0.928
70	0.814	0.86	0.89	0.846	0.85	0.870	0.870	0.850	0.597	0.869	0.903	0.901	0.902
80	0.735	0.81	0.83	0.764	0.831	0.800	0.803	0.769	0.333	0.802	0.862	0.867	0.862
90	0.592	0.68	0.69	0.60	0.789	0.676	0.673	0.749	0.311	0.674	0.793	0.808	0.791

Table 4 Evaluation (Pratts FOM) of algorithms on Lena Image corrupted by SPN

ND in %	DBA	IDBA	MDBMF	CUDMPF	MDBUTMF	MDBUMF_GM	ACBSA	DMF	DBATM WMF	DBATM WMeanF	DBATM WVF	DBATM WGM(PA)
10	0.885	0.892	0.942	0.733	0.940	0.942	0.929	0.88	0.950	0.946	0.956	0.946
20	0.871	0.856	0.896	0.688	0.890	0.904	0.894	0.82	0.919	0.913	0.931	0.903
30	0.823	0.831	0.861	0.670	0.852	0.853	0.859	0.75	0.878	0.882	0.892	0.867
40	0.787	0.797	0.813	0.647	0.807	0.805	0.825	0.65	0.833	0.846	0.839	0.828
50	0.743	0.758	0.763	0.641	0.735	0.741	0.791	0.56	0.767	0.796	0.802	0.793
60	0.699	0.727	0.651	0.621	0.664	0.659	0.626	0.46	0.695	0.739	0.764	0.731
70	0.582	0.670	0.528	0.558	0.587	0.583	0.594	0.37	0.597	0.679	0.709	0.668
80	0.467	0.580	0.441	0.477	0.468	0.468	0.468	0.24	0.493	0.575	0.628	0.575
90	0.332	0.425	0.306	0.392	0.326	0.339	0.441	0.19	0.363	0.432	0.505	0.457

Table 5 Evaluation (ER) of algorithms on Lena Image corrupted by SPN

ND in %	DBA	IDBA	MDBMF	CUD MPF	MDB UTMF	MDBUMF _GM	ACBSA	AWMF	DBAT MWMF	DBATM WMeanF	DBATM WVF	DBATM WGM(PA)
10	20.85	21.01	8.78	21.34	8.84	8.9	8.72	10.39	8.53	9.8	8.55	9.9915
20	24.75	24.86	17.5	25	17.53	17.67	17.36	18.35	17.28	19.79	17.36	19.8673
30	30.5	30.64	26.6	31.13	26.69	26.42	26.19	26.28	26.36	29.82	26.76	29.8157
40	37.75	37.57	35.81	38.96	35.75	35.68	35.12	34.54	35.83	39.91	36.55	39.9666
50	45.7	45.91	45.24	48.74	45.27	45.22	43.66	43.08	45.32	49.75	46.34	49.9386
60	53.96	54.49	55.02	60.04	55.16	55.12	53.57	52.07	55.04	59.86	56.93	59.8949
70	63.08	63.31	65.63	71.6	65.55	65.7	62.48	61.31	65.49	70	67.75	69.9665
80	72.83	72.56	76.93	83.1	77.05	76.98	71.47	71.21	76.87	80	78.68	79.9984
90	83.29	82.46	88.97	93.1	87.97	89.02	81.38	82.32	88.96	90.03	89.61	90.0486

proposed algorithm employs a fixed 3 x 3 window filter for increasing noise densities. The initial challenge was to extract the limited resource from a small neighborhood. This was achieved using a Modified Winsorized geometric mean of a small neighborhood. The algorithm exhibits excellent noise suppression characteristics with vital information preservation on images which are heavily corrupted. The proposed method exhibits high PSNR, MSE, IEF with excellent SSIM and FOM with a reduced error rate in comparison with 10%to 90% noise densities. The Modified Winsor zed geometric mean algorithm is relatively simple and exhibits very good results.

References

[1] Vasanth K, Nirmal raj S, Manjunath T G, " A Decision based unsymmetrical trimmed modified winsorized mean filter for the removal of high density salt and pepper noise in images and videos", eleventh international multi-conference on information processing, procedia computer science 54, 595-604, (2015)

[2] Fangfang dong , Yunmei chen, De-xing kong, Bailin yang, " Salt and pepper noise removal based on an approximation of L0 norm" computers and mathematics with applications, 70, 5, 789-804, (2015)

[3] Xiangyu deng, Yide ma, Min dong ," A New adaptive filtering method for removing salt and pepper noise based on multi layered PCNN" , Pattern recognition letters 2016, 79, 1, 8-17,(2016)

[4] Xianying qi , boqiang liu , jianwei xu, "a neutrosophic filter for high- density salt and pepper noise based on pixel-wise adaptive smoothing parameter" Journal of Visual communication .image , 36, 1-10, (2016).

[5] Ching-ta lu , yung-yue –chen, ling-ling wang, chun-fan chang," removal of salt-and- pepper noise in corrupted image using three-values-weighted approach with variable -size window", pattern recognition letters, 80, 188-199,(2016)

[6] Changhong Wang ; Taoyi Chen ; Zhenshen Qu, "A novel improved median filter for salt-and pepper noise from highly corrupted images",International Symposium on Systems and Control in Aeronautics and Astronautic, 718 - 722, (2010).

[7] Christo MS, Vasanth K, Varatharajan R,"A decision based asymmetrically trimmed modified winsorized median filter for the removal of salt and pepper noise in images and videos" ,Multimedia Tools and Applications, 79, 1, 415-432, (2020)

[8] Vasanth K, Varatharajan R, "A decision based unsymmetrical trimmed modified Winsorized variants for the removal of high density salt and pepper noise in images and videos" Computer Communications 154, 433-441, (2020)

[9] Qin xu, qiang zhang, duo hu , jinpei liu," Removal of salt and pepper noise in corrupted image based on multilevel weighted graphs and igowa operator" hindawi, mathematical problems in engineering, 2018, Article ID 7975248,1-11, (2018).

[10] M. Nadeem , Ayyaz Hussain , Asim Munir , M. Habib , M. Tahir Naseem , "Removal of Random Valued Impulse Noise from Grayscale Images using Quadrant based Spatially Adaptive Fuzzy Filter", Signal Processing Journal(Elsevier), 169, 1, 1-32, (2019).

[11] Serdar enginoglu, Ugur Erkan, Samet Memis, "Pixel similarity-based adaptive Riesz mean filter for salt and pepper noise removal", Multimedia Tools and Applications, 78, 24, 35401-35418, (2019).

[12] Chaipichit Cumpim ; Rachu Punchalard ; Kanok Janchitrapongvej ; Chom Kimpan, "Salt-and pepper noise removing by Shepard interpolation method", International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 1 - 5,(2016)

[13] http://sipi.usc.edu/database/ (Accessed on 23 May 2020).

[14] Z. Wang, A. C.Bovik, H. R. Sheikh, and E. P. Simoncelli , "Image quality assessment: From error measurement to structural similarity" IEEE Transactions on Image Processing, 13, 4, 102-109, (2004)

[15] Abdou, I.A., Pratt, W. (1979): Quantitative design and evaluation of enhancement/thresholding edge detectors. Proc. IEEE 67,5, 753–766,(1979)