International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 4, April 2012)

Modified Boundary Discriminative Noise Detection and Removal Technique for Salt and Pepper Noise Removal

P. H. Sangave[1], Prof. G.P. Jain[2]
[1,2]WIT Solapur
[1]Piyu_s07@yahoo.com
[2]gpjain_bv@rediffmail.com

Abstract: **In this paper an efficient nonlinear decision based filter is proposed to remove salt and pepper impulse noise. Propose filter is a two stage filter that incorporates a powerful impulse noise detection method, called the modified boundary discriminative noise detection (MBDND) [1] to determine whether the current pixel is corrupted or not. In the second stage a Euclidean distance algorithm [2] is used to restore the corrupted pixels.**

Extensive experimental results demonstrate that proposed filters performs significantly better than many existing, well accepted and recently proposed median and decision based filters for both gray scale and color images corrupted up to 70% of salt and pepper noise.

Keywords: image restoration, impulse noise, decision based filters, discriminative noise detection, Euclidean distance.

I. Introduction

Noise is any undesired information that contaminates an image. Impulse noise is a special type of noise, which have many different origins. The Salt and Pepper type impulse noise is typically caused by malfunctioning of the pixel elements in the camera sensors, faulty memory locations, or timing errors in the digitization process. For the images corrupted by Salt and Pepper noise, the noisy pixels can take only the maximum or the minimum values in the dynamic range.

An important non linear filter that will preserve the edges and remove impulse noise is standard median filter Median (SMF) [3]. It replaces every pixel by its median value from its neighborhood and often removes desirable details in the image. Specialized median filter such as weighted median filter (WMF) [4] were proposed to improve the performance of median filter by giving more weight to some selected pixel in the filtering window. But they still implemented uniformly across the images without considering the current pixel whether is noisy or noise free. Therefore, a noise-detection process to discriminate between uncorrupted pixels and corrupted pixels is highly desirable. Some of decision based algorithms, such as Adaptive Median filter (AMF) [5], These algorithms first detect the noisy pixels and remove it by applying either standard median filter or its variants. These filters are effective in removing low to medium density impulse noise only.

The outline of the paper is as follows. Boundary discriminative noise detection (BDND) scheme is reviewed in Section 2. Our denoising scheme is presented in Section 3. Experimental results and conclusions are presented in Sections 4 and 5, respectively.

II. Boundary Discriminative Noise Detection (BDND)

The BDND algorithm first classifies the pixels of a localized window, centering on the current pixel, into three groups—*lower intensity impulse noise*, *uncorrupted pixels*, and *higher intensity impulse noise*. The center pixel will then be considered as "uncorrupted," provided that it belongs to the "uncorrupted" pixel group, or "corrupted." For that, two boundaries that discriminate these three groups require to be accurately determined for yielding very high noise detection accuracy.

The BDND algorithm is applied to each pixel of the noisy image in order to identify whether it is "uncorrupted" or corrupted." After such an application to the entire image, a two-dimensional binary decision map is formed at the end of the noise detection stage, with "0s" indicating the positions of "uncorrupted" pixels, and "1s" for those "corrupted" ones.

To accomplish this objective, all the pixels within a pre-defined window that center on the considered pixel will be grouped into three clusters; hence, two boundaries $b_1$and b_2 are required to be determined. For each pixel being considered, if $0 \le x_{i,j} \le b_1$, the pixel will be assigned to the *lower*-intensity cluster; otherwise, to the *medium*-intensity cluster for $b1 < x_{i,j} \le b_2$ or to the *high*-intensity cluster for $b_2 < x_{i,j} \le 255$. Obviously, if the center pixel being considered falls onto the middle cluster, it will be treated as "uncorrupted," since its intensity value is neither relatively low nor relatively high. Otherwise, it is very likely that the pixel has been corrupted by impulse noise. Clearly, the accuracy of clustering results (hence, the accuracy of noise detection) ultimately depends on how accurate the identified boundaries $b_1$and b_2 are.

The *boundary discriminative* process consists of two iterations, in which the second iteration will only be invoked conditionally. In the first iteration, an enlarged local window with a size of 21x21 (empirically determined) is used to examine whether the considered pixel is an uncorrupted one.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 4, April 2012)

If the pixel fails to meet the condition to be classified as "uncorrupted" (i.e., not falling onto the middle cluster), the second iteration will be invoked to further examine the pixel based on a more confined local statistics by using a 3x3 window. In summary, the steps of the BDND algorithm are:

Step 1) Impose a 21x21 window, which is centered on the current pixel.
Step 2) Sort the pixels in the window according to the ascending order and find the median, *med*, of the sorted vector V_o.
Step 3) Compute the intensity difference between each pair of adjacent pixels across the sorted vector V_o and obtain the difference vector V_D .
Step 4) for the pixel intensities between 0 and *med* in the V_o, find the maximum intensity difference in the V_D of the same range and mark its corresponding pixel in the V_o as the boundary b_1.
Step 5) Likewise, the boundary b_2 is identified for pixel intensities between *med* and 255; three clusters are, thus, formed.
Step 6) If the pixel belongs to the middle cluster, it is classified as "uncorrupted" pixel, and the classification process stops; else, the second iteration will be invoked in the following.
Step 7) Impose a 3x3 window, being centered on the concerned pixel and repeat Steps 2)–5).
Step 8) If the pixel under consideration belongs to the middle cluster, it is classified as "uncorrupted"
Pixel; otherwise, "corrupted."

III. Proposed Algorithm (PA)

Many denoising schemes are "*decision-based*" median filters this means that the noise candidates are first detected by some rules and are replaced by the median output or its variants. For instance, in decision based filter (DBA) [6] & improved decision based filter (IDBA) [7], the noise candidate is replaced by median or mean of neighboring pixels. These schemes are good because the uncorrupted pixels will not be modified. However, the replacement methods in these denoising schemes cannot preserve the features of the images; in particular the edges are smeared. In contrast, Boundary discriminative Noise detection can preserve edges during denoising but it uses 21x21 huge window size followed by 3x3 window size for noise detection that require more processing time & its filtering process is complicated.

Combining Modified Boundary Discriminative Noise Detection followed by Euclidean distance replacement of noisy pixel only with adaptive window size to noise density will provide the desire results. Working window is Decided based on the estimated noise density adaptively as given in table below.

Suggested Window size for the estimated noise density

Noise density (ND)	Working window (W×W)
P < 50%	3 × 3
50% ≤ P ≤ 70%	5 × 5
P > 70%	7 × 7

The steps of proposed algorithm are:

Step 1) Impose a working window of W × W centered on the current pixel.
Step 2) Let $X_{i,j}$ be the current pixel. Check whether the $X_{i,j}$ lies in between '0' and '255' if so declare it as "uncorrupted" pixel and go to step 8
Step 3) Sort the pixels in the working window according to the ascending order and find the median, med, of the sorted vector V_o.
Step 4) Compute the intensity difference between each pair of adjacent pixels across the sorted vector V_o, and obtain the difference vector V_D.
Step 5) for the pixel intensities between 0 and med in the V_o, find the maximum intensity difference in the same range and mark its corresponding pixel as the boundary B_1.
Step 6) Likewise, the boundary B_2 is identified for pixel intensities between med and 255; three clusters are, thus, formed.
Step 7) If the pixel belongs to the middle cluster, it is classified as "uncorrupted" pixel; otherwise it is classified as corrupted one.
Step 8) Prepare a binary flag map B(i, j) with '0' indicating the uncorrupted pixels and '1' indicating the corrupted one.
Step 9) Start the iterative computation process to restore the corrupted pixels within a corrupted image as follows.
a) Let W × W be a working sliding window whose centre pixel is a corrupted pixel, find the number of uncorrupted pixels within the current window W × W. Perform the following steps if the number of the uncorrupted pixels that exists within the current window is greater than zero.

(i) For the current window, compute the Euclidean distances, D_t, between the centre pixel and the uncorrupted pixels by using the formula

$$D_t = \left|\sqrt{k_t^2 + L_t^2}\right| , \quad t = 1,2,3,\ldots.s, \tag{i}$$

Where s denotes the number of uncorrupted pixels that exist within the current window. (K,L) are integers ($-1 \le K \le 1$, $-1 \le L \le 1$), which denote the spatial coordinates of uncorrupted pixels within the window. The spatial coordinates of centre pixel of window is (K = 0, L = 0).

(ii) Convert the computed distance D_t values to distance weight , H_t, by using the equation.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 4, April 2012)

$$H_t = \left(\frac{D_t}{\sum_{t=1}^{S} D_t} \right)$$

(iii) Restore the intensity value of the centre pixel in current window with the value of V_t, which is computed by using equation.

$$V_t = \sum_{t=1}^{S} H_t P_t \qquad \text{(iii)}$$

Where P_t denotes the intensity values of the uncorrupted pixels within the current window.

b)If the number of uncorrupted pixels in current window is equal to zero, then don't replace the intensity value of centre pixel.

3.1 COLOR IMAGE DENOISING

Generally there are two approaches for color image denoising, Scalar Median filtering approach and vector Median filtering approach. The scalar median filtering approach has been used in this paper. The RGB color space is used in this project to represent the color images. In the RGB color space, each pixel at the location (i,j) can be represented as color vector $O_j = [\, O_j^R \; O_j^G \; O_j^B]$, Where O_j^R O_j^G and O_j^B are the red(R), green (G), and blue(B) components, respectively. The noisy color images are modeled by injecting the salt and pepper noise to each of these color components.

IV. RESULTS & SIMULATIONS

Among the commonly tested 512x512 8-bit grayscale Lena and Lena color images, will be selected for our simulations. In the simulations, images will be corrupted by "salt" (with value 255) and "pepper" (with value 0) noise with equal probability. Also a wide range of noise levels varied from 10% to 70% with increments of 10% will be tested. Restoration performances are quantitatively measured by the peak signal-to-noise ratio (PSNR) and the mean absolute error (MAE).

$$PSNR = 10 \log_{10} \frac{255^2}{\frac{1}{MN} \sum_{i,j} (r_{i,j} - x_{i,j})^2} \qquad \text{(iv)}$$

$$MAE = \frac{1}{MN} \sum_{i,j} | r_{i,j} - x_{i,j} | \qquad \text{(v)}$$

Where $r_{i,j}$ and $x_{i,j}$ denote the pixel values of the restored image and the original image, respectively.

4.1 Denoising Performance:

The denoising performance of proposed algorithm and other standard methods are tested for grayscale image and color image. The visual quality results are presented in tables below for different noise densities of salt and pepper noise ranging from 10% to 70%.

Fig. 1 a) Original Lena image[1] b) Noisy image with 70% Noise Density & PSNR of 6.99 c) DBA with PSNR of 25.69 d) IDBA with PSNR of 24.46 e) PA with PSNR of 28.97.

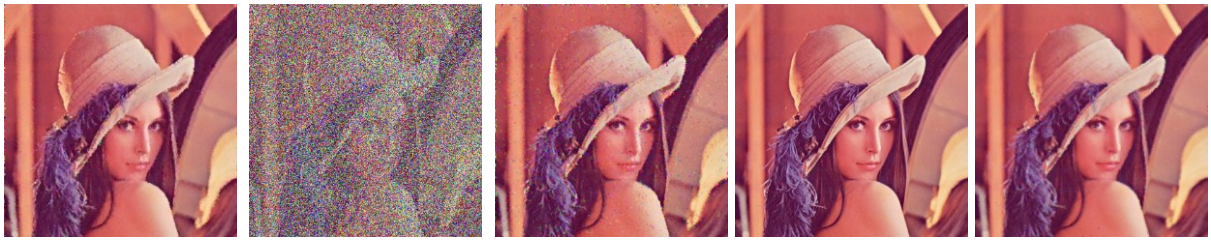

Fig.2 a)Original Lena color image[1] b)Noisy image with 70% Noise Density & PSNR of 31.67 c) DBA with PSNR of 43.72 d) IDBA with PSNR of 42.87 e) PA with PSNR of 44.06.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 4, April 2012)

ND	SMF	DBA	IDBA	PA
10%	33.12	41.58	41.39	42.29
20%	28.76	37.44	37.32	38.91
30%	23.64	34.69	34.50	36.54
40%	19.08	32.18	32.22	34.18
50%	15.29	30.18	29.72	31.12
60%	12.35	27.92	27.18	30.03
70%	10.0	25.69	24.46	28.97

Table 1. Peak Signal to Noise ratio of "Lena"
Test image for different noise removal techniques

ND	SMF	DBA	IDBA	PA
10%	2.79	0.40	0.40	0.37
20%	3.57	0.86	0.88	0.78
30%	5.32	1.43	1.44	1.22
40%	9.21	2.10	2.12	1.71
50%	16.81	2.92	3.08	2.84
60%	29.77	4.03	4.48	3.50
70%	47.27	5.56	6.75	4.22

Table 2. Mean Absolute Error of "Lena"
Test image for different noise removal techniques.

ND	SMF	DBA	IDBA	PA
10%	46.72	55.00	55.01	55.16
20%	45.79	51.67	51.66	52.07
30%	44.12	49.47	49.51	50.20
40%	41.70	47.86	47.75	48.77
50%	38.95	46.42	46.21	45.66
60%	36.52	45.12	44.60	44.79
70%	34.47	43.72	42.87	44.06

Table 3 Peak Signal to noise ratio of "Lena" color image for different noise removal techniques

ND	SMF	DBA	IDBA	PA
10%	1.38	0.20	0.20	0.19
20%	1.71	0.44	0.44	0.40
30%	2.51	0.73	0.72	0.61
40%	4.38	1.06	1.08	0.86
50%	8.26	1.48	1.55	1.76
60%	14.46	1.99	2.25	2.15
70%	23.22	2.75	3.35	2.55

Table 4. Mean Absolute Error of "Lena" color image for different noise removal techniques

The quantitative performances in terms of PSNR, and MAE for all the algorithms are given in table 1,2 3 4 below the same is plotted in figures 1& 2. For lower noise density up to 30% almost all algorithms performs equally well in removing the salt and pepper noise completely with edge preservation as show in table. For the case of noise density 40% and above the standard algorithms such as SMF, WMF fails to remove the salt

In the case of high noise density such as 70% of salt and pepper noise the standard methods are very poor in noise cleaning and details preservation for the case of 70% of noise density AMF, is slightly less than that of proposed filter in terms of noise removal and edge preservation as shown in figures. The maximum window size of 17x17 is selected for AMF to give better at high noise density level. The visual quality, PSNR, MAE and MSE results clearly show that the proposed filter out performs the many of standard and recently proposed filters.

4.2 Computational time

The CPU time of proposed algorithms is compared with standard filters in the table for Lena gray and Lena color image respectively. For the case of comparison Lena grayscale and Lena color image is corrupted with salt and pepper noise of density 70% and applied to all filters. All the algorithms are implemented in MATLAB 7.1 on a PC equipped with AMD Athlon 64x2 Dual core processor of 2.71 GHZ Capacity and 1.75 GHZ RAM memory. The computation time of proposed filter is slightly higher than the decision based algorithms since the proposed algorithms uses adaptive filtering window instead of fixed window as in case of DBA.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 4, April 2012)

FILTER	TIME
Standard Median Filter	0.08
Decision Based Filter	2.83
Improve Decision Based Filter	3.08
Proposed Algorithm	76.82

Table 5.Computational Time in Seconds for Denoising 70% Salt & Pepper Noise Corrupted LENA (512x512 Grayscale) Image for Different Denoising Techniques.

FILTER	TIME
Standard Median Filter	0.18
Decision Based Filter	8.55
Improve Decision Based Filter	9.28
Proposed Algorithm	233.65

Table 6.Computational time in seconds for denoising 70% salt & pepper noise corrupted LENA (512x 512 colors) Image for different denoising techniques.

V CONCLUSIONS

In this paper an efficient decision based filter are proposed to remove low to high value of salt and pepper noise with edge preservation. The proposed decision based filter performs well for both gray scale and color image with different noise model of salt and pepper noise. In proposed algorithm modified boundary discriminator can accurately tell where the noise is, only noise affected pixels are replaced by algebraic sum of product of Euclidean distance and intensity of unaffected pixels within the working window.

REFERENCES

[1] Pei-Eng Ng and Kai-Kung Ma, "A switching Median Filter with Boundary Discriminative Noise Detection for Extremely Corrupted Images", IEEE Trans on Image Processing, Vol 15, No 6 June 2006.

[2] Pinar Civicioglu, Mustafa Alci and Erkan Besdok, "Impulse Noise Suppression from Images with the Noise Exclusive Filter", EURASIP Journal on Applied signal Processing 2004 Vol 16 pp 2434-2440.

[3] Astola J and P. Kuosamanen, "Fundamentals of Nonlinear digital Filtering", CRC Press 1997.

[4] Brownrigg D.R.K, " The Weighted Median Filter," Communication, ACM Vol 27, No 3 pp 807-818 1984.

[5] Hwang H and Haddad R.A, "Adaptive Median Filters new and results", IEEE Trans on Image Processing Vol 4 No 4 pp 499-502, 1995.

[6] K.S Srinivasan, D.Ebenezer, "A New Fast and Efficient Decision-Based Algorithm for Removal of high Density Impulse Noises," IEEE Signal Processing Letters Vol 14 No 3 pp 189-192 March 2007.

[7] Madhu S Nair, K Revathy and Rao Tatavarti, "An Improved Decision-Based Algorithm for Impulse Noise Removal", 2008 Congress on Image and Signal Processing pp No 426-431.