A modified approach for Removal of High density Salt and Pepper Noise in Color Images

Gaganpreet Kaur Bhullar[1], Vinod Singla[2]
M.tech student[1], Department of Computer Science Engineering
Assistant Professor[2], Department of Computer Science Engineering
Guru Kashi University, Talwandi Sabo, Bathinda, Punjab, India.
Email ID-gaganbhullar9@gmail.com

ABSTRACT

This paper proposed a simple edge preserved denoising algorithm to remove high density salt and pepper noise in digital color images. The proposal has three steps: noisy pixel detection, replacement of noisy pixels and edge preservation using Kuwahara filter. The noisy pixel is replaced by trimmed median value when some of the elements with value 0's and 255's are present in the selected window. If all the pixel values in the selected window are 0's and 255's then the noisy pixel is replaced by mean value of all the elements present in that selected window. The color images are denoised by extracting the R, G and B planes from the noisy image, denoised separately and are merged together to form the color image, rather than converting to gray for denoising and then reconstructing from the denoised gray image. The proposed algorithm will yields visually pleasing images in terms of qualitative and quantitative evaluation compared to other standard algorithms.

Keyword -Color image de-noising, salt and pepper noise, Kuwahara filter.

I. INTRODUCTION

Salt and pepper noise is characterized by the appearance of black and white dots in the image. The characteristic of this noise is that some of the original pixel values are retained in the noisy image and only some of the pixels are affected by the salt and pepper noise. The affected pixel value jumps either to a value close to 255 (salt noise) or to a value nearby 0 (pepper noise). Since there is an unusual jump in the value of the pixel, this noise is also called as impulse noise [1].

Many researches have been conducted and numerous algorithms were proposed to remove salt and pepper noise. Among these noise reduction techniques, majority splits the noise removal procedures intopreliminarily detection of pixels corrupted by impulse noise followed by filtering the noise detected on the previous phase [2]. Standard median filter (SMF) [2] Was one of famous among others due to its great denoising performance and computational efficiency. But since the conventional median filter applies the median operation to each pixel whether it is corrupted or not, it suffers from Preserving details of the image as the noise density increases. More improved algorithms such as adaptive median filter (AMF) [3], decision-based algorithm (DBA) [4], Decision Based Unsymmetric Trimmed Median Filter (DBUTMF) [5] and Modified Decision Based Unsymmetric Trimmed Median Filter (MDBUTMF) along with Fuzzy noise reduction method (FNRM) [6] mainly focus on noise detector and noise removal only. Pixels detected by the noise detector will be considered as noise and shall further be processed in their respective noise reduction scheme. Having such mechanism with phases would highly preserve the details of the image and save the restored image from having blurred and distorted feature.

The proposed algorithm includes Modified Decision Based Unsymmetric Trimmed Median Filter (MDBUTMF) followed by edge preserving Kuwahara filter and Fuzzy noise reduction method (FNRM) works effectively on those images which have been corrupted with high density salt and pepper noise to preserve edges and details of the restored image. This makes the proposed algorithm to have an outstanding performance even at noise density as high as 90%.

Denoising of a color image is done by converting it into gray, denoised and get back to color. In this process while reconverting gray to color image, the original information might get lost. Therefore in this paper, the originality of the color images are maintained by extracting the R, G and B planes from the noisy image, denoised separately and are merged together to form the color image. The rest of the paper is organized as follows; in section II the proposed algorithm is explained. Section III presents the experimental results. Finally conclusions are discussed in section IV.

II. PROPOSED ALGORITHM

The proposed algorithm is a windowing technique and so the least window size 3×3 is chosen to reduce complexity. In this algorithm, the pixel of interest is the centre pixel $P_{i,j}$. For the pixel $P_{i,j}$ to be the very first pixel in the image, the noisy image is surrounded by a border copying one complete row/column from the four edges of the noisy image.

1	**1**	**2**	**3**	**4**	**5**	**6**	**6**
1	**1**	**2**	**3**	**4**	**5**	**6**	**6**
11	**11**	**12**	**13**	**14**	**15**	**16**	**16**
21	**21**	**22**	**23**	**24**	**25**	**26**	**26**
31	**31**	**32**	**33**	**34**	**35**	**36**	**36**
41	**41**	**42**	**43**	**44**	**45**	**46**	**47**
51	**51**	**52**	**53**	**54**	**55**	**56**	**56**
51	**51**	**52**	**53**	**54**	**55**	**56**	**56**

Fig. 1. Adding border to an image.

In the Fig. 1 the pixel values within the box represents the image. By the method shown in "Fig.1", the first pixel from the image is selected as the centre pixel value in the first window. The window then traverses by one column in the right leaving one column in the left, now the second pixel value will be the centre pixel value and so on until the last column in the noisy image is reached, then the row is incremented by one and column is reset to one and this process continues till the last pixel in the image is reached. Each window has nine pixel values consisting of 3 rows and 3 columns.

	j-1	**j**	**j+1**
i-1	$P_{i-1,j-1}$	$P_{i-1,j}$	$P_{i-1,j+1}$
i	$P_{i,j-1}$	$P_{i,j}$	$P_{i,j+1}$
i+1	$P_{i+1,j-1}$	$P_{i+1,j}$	$P_{i+1,j+1}$

w: 3×3 mask

Fig. 2. 3×3 window centered on $P_{i,j}$

Except the first row of the pixels in the image, while denoising the remaining pixel values the first row in the window is already denoised. Fig.3 shows the block diagram of the algorithm and is explained below

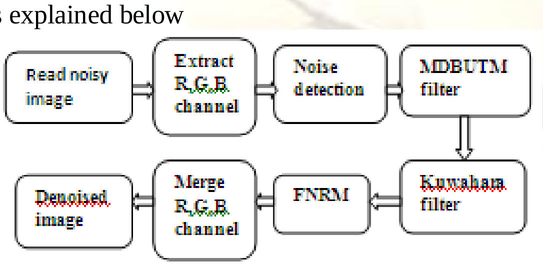

Fig.3. Block diagram.

1. Noise Detection and correction using MDBUTMF

Step 1: Select 2-D window of size 3X 3. Assume that the pixel being processed is $P_{i,j}$.

Step 2: If $0 < P_{i,j} < 255$ then is an uncorrupted pixel and its value is left unchanged.

Step 3: if $P_{i,j}$ is a corrupted pixel then two cases are possible as given in Case i) and ii).

Case i): If the selected window contains all the elements as 0's and 255's. Then replace with the mean of the element of window.

$$\begin{bmatrix} 255 & 0 & 0 \\ 0 & (0) & 255 \\ 255 & 255 & 0 \end{bmatrix}$$

where '0' is the processing pixel.

Since all the elements surrounding are 0's and 255's. If one takes the median value it will be either 0 or 255 which is again noisy.

Case ii): If the selected window contains not all elements as 0's and 255's. Then eliminate 255's and

$$\begin{bmatrix} 66 & 71 & 0 \\ 92 & (0) & 255 \\ 77 & 255 & 104 \end{bmatrix}$$

0's and find the median value of the remaining elements. Replace with the median value. Step 4: Repeat steps 1 to 3 until all the pixels in the entire image are processed.

Where '0' is processing pixel, i.e., $P_{i,j}$. Now eliminate the salt and pepper noise from the selected window. That is, elimination of 0's and 255's. The 1-D array of the above matrix is [66 71 0 92 0 255 77 255 104]. After elimination of 0's and 255's the pixel values in
the selected window will be [66 71 92 77 104]. Here the median value is 82. Hence replace the processing pixel by 82.

2. Flow chart

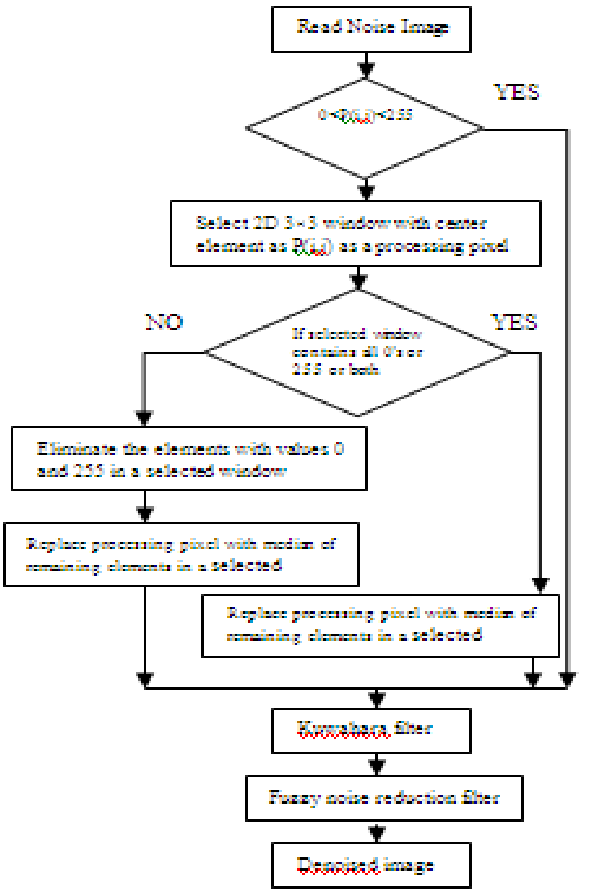

Fig. 4 Flow chart

3. Edge Preservation using Kuwahara filter

To preserve the details and edges of the restored image we apply Kuwahara filtering which will enhance the image quality. The basic process of Kuwahara filter is described as follows: Firstly, some different templates are made based on the center pixel. Secondly, the mean value and the standard deviation of the pixels in different templates are calculated. Finally, the gray value of the center pixel is defined as the mean value in the template where the standard deviation is the least. The Kuwahara filtering selects one template from 4 square windows. Through the analysis of this algorithm, we find that the image details can be preserved from selecting a suitable template according to the rule of minimizing the standard deviation.

4. Fuzzy Noise Reduction Methods

The output of MDBUTMF followed by Kuwahara filter is a partially noise removed one. So it is get further processed using Fuzzy Filter. The Fuzzy Filter consists of two sub-filters namely Fuzzy First Sub-Filter and Fuzzy Second Sub-Filter. The Fuzzy First Sub-Filter computes fuzzy distances between the color components of the processing pixel and its neighborhood. These distances determine in what degree each component should be corrected. The aim of the Fuzzy Second Sub-Filter is to correct the corrupted pixels. The detailed explanations of fuzzy filters are given below.

4.1. Fuzzy First Sub-Filter

The main aim of the Fuzzy First Sub-Filter is to average the processing pixel by using its neighborhood pixel values. Meanwhile care should be taken in order to avoid the damage of the image features such as edges and color component distances. The color component distances are used to differentiate between the local variations caused by noise and the image structures such as edges. This is one of the main concerns of the Fuzzy First Sub-Filter.

In order to find the color component distances, the following couples are used. They are rg(i, j), rb(i, j) and gb(i,j). The couple red and green is given by rg(i, j)=(N(i,j,1), N(i,j,2)), the couple red and blue is denoted as rb(i, j)=(N(i,j,1),N(i,j,3)) and the couple green and blue is denoted by gb(i, j)=(N(i,j,2),N(i,j,3)). Where N (i, j, 1), N (i, j, 2) and N (i, j, 3) are the certain pixel positions of a noisy input image N for the red, green and blue components. The size of the window considered here is (2K + 1) × (2K + 1) and it is get centered at (i, j) that is the processing pixel. For each pixel in the selected window, certain weights are get assigned based on the color components. The weights that are get assigned for red, green and blue at position (i+k, j+l) are w(i+k, j+l, 1), w(i+k, j+l, 2) and w (i+k, j+l, 3) respectively with parameters k,l ε{-K…K}. The distance between the couple rg(i,j) and rg(i+k, j+l) is calculated according to the Minkowski's distances. And it is given below in expression (1).

$$D(rg(i, j), rg(i+k, j+l)) = [(N(i+k, j+l, 1) - N(i, j, 1))^{\delta} + (N(i+k, j+l,2) - N(i, j, 2))\delta]^{1} \qquad (1)$$

In order to find at what degree the distance between two couples be small, Fuzzy set small is used. These fuzzy sets are commonly represented by membership functions. It is denoted by 'µ'. By using such membership functions the membership degree is get derived. If the distance between two couples has a membership degree of one in the Fuzzy Set Small means then the distance is considered as small for sure. But if the distance between them has a membership degree of zero in the Fuzzy Set Small means then the distance is not consider as small. So here there is a kind of uncertainty is present, for the membership degrees occur between zero and one. The membership function small is denoted by 'μ_s'. If $x \leq p$ then the value of $\mu_s(x)$ is $(p-x/p)^2$ and for $x>p$ the value of $\mu_s(x)$ is zero. The output of the Fuzzy First Sub-

Filter for red component is given below in expression (2). It contains the weight function for red component that is w(i+k,j+l,1) at position (i+k, j+l). The weight function contains numerical number one that indicates that it belongs to red component.

$$F(i,j,1)=\frac{\sum_{k=-k}^{k}\sum_{i=-k}^{k} w(i+k,j+l,1)\times N(i+k,j+l,1)}{\sum_{k=-k}^{k}\sum_{l=-k}^{k} w(i+k,j+l,1)} \quad (2)$$

This expression for w(i+k, j+l,1) that is present in above expression(2) is given below in (3)

$$w(i+k, j+l,1)=\mu_{s1}(\gamma_{rg}(i, j, k, l))\times\mu_{s2}(\gamma_{rb}(I, j, k, l)) \quad (3)$$

Where F(i,j,1) is the output image for the red component at position (i,j) and numerical number one indicates that it belongs to the red component. The numerical numbers two and three are used for green and blue components respectively. The above filtering method is similar for both green and blue components.

4.2. Fuzzy Second Sub-Filter

The window size considers here is (2L+1)(2L+1), but it is not necessary that the value of 'L' must be equal to 'K' of the Fuzzy First Sub-Filter. To filter the current image pixel at position (i,j) the window is placed centered at (i,j). The local differences (LD) for red, green and blue components are denoted by LD_R, LD_G and LD_B . These are calculated as follows and given in expression (4).

$$LD_R(k, l) = F(i+k,j+l,1)-F(I,j,1)$$
$$LD_G(k, l) = F(i+k,j+l,2)-F(I,j,2)$$
$$LD_B(k,l) = F(i+k,j+l,3)-F(I,j,3) \quad (4)$$

These differences are finally used to calculate the following correction term given in expression (5)

$$\varepsilon(k,l) = ⅓(LD_R(k,l) + LD_G(k,l) + LD_B(k,l)) \quad (5)$$

Where ε(k,l) is the correction term for the neighboring pixel F(i+k,j+l,1), F(i+k,j+l,2) and F(i+k,j+l,3) of the processing pixel. The output of the Fuzzy Second Sub-Filter for the red component is given below in expression (6). It is denoted as Out (i,j,1) that is it represent the output image of Fuzzy Second Sub-Filter for red component at position (i,j). The numerical number one indicates that it belongs to red component. The numerical number two and three are used to represent green and blue.

$$Out(i,j,1)=\frac{\sum_{k=-L}^{+L}\sum_{i=-L}^{+L}(F(i+k,j+l,1)+\varepsilon(k,l)}{(2L+1)^2} \quad (6)$$

The above procedure is similar for the case of green and blue components. The correction term is similar for both green and blue components. Where, $(2L+1)^2$ is the size of the window that is selected for the Fuzzy Second Sub-Filter and F (i+k,j+l,1) is the output of Fuzzy First Sub-Filter for red component.

III. METHODOLOGY AND RESULTS

i. The noisy image is read through MATLAB and its corresponding pixel values are written into a file.
ii. The proposed algorithm is processed and the output is written into another file From this file
iii. The denoised image is reconstructed in MATLAB from the output file.

Denoising a color image, by converting it into its equivalent gray image, the original information might get lost during reconstruction. The RGB color image is converted into its gray by 0.2989*R+0.5870*G+0.1140*B. during the reconstruction process the gray image consists of only the intensity values so the reverse process of reconstruction does not give the original color values. Therefore for color images, the three planes R, G and B are extracted from the image and are written separately into 3 files and the denoising algorithm is executed for three times for a single color image.

Table 1: comparison of restoration results for noise free image

DENSITY IN %	DBA	MDBA	MDBUTMF	PROPOSED
10	36.40	36.94	39.91	49.52
20	32.90	32.69	35.98	47.23
30	30.15	30.41	34.99	43.54
40	28.49	28.49	32.40	42.51
50	26.41	26.52	30.93	41.31
60	24.83	24.41	29.43	39.25
70	22.64	22.47	25.97	36.84
80	20.32	20.44	24.41	33.62
90	17.14	17.56	22.98	29.81

"Fig. 5" shows the method of denoising a color image affected by 10% impulse noise by extracting the three planes separately. "Fig. 6" shows the image affected by various densities of noise and its reconstructed images. To explore the visual quality the proposed algorithm is compared with standard algorithms and the results are displayed in "Fig. 7".

In addition to the visual quality, the performance of the proposed algorithm and standard algorithms are quantitatively measured by peak signal-to-noise ratio (PSNR). The quantitative performance in terms of PSNR for image affected by 10% to 90% of impulse noise is given in “table 1”.

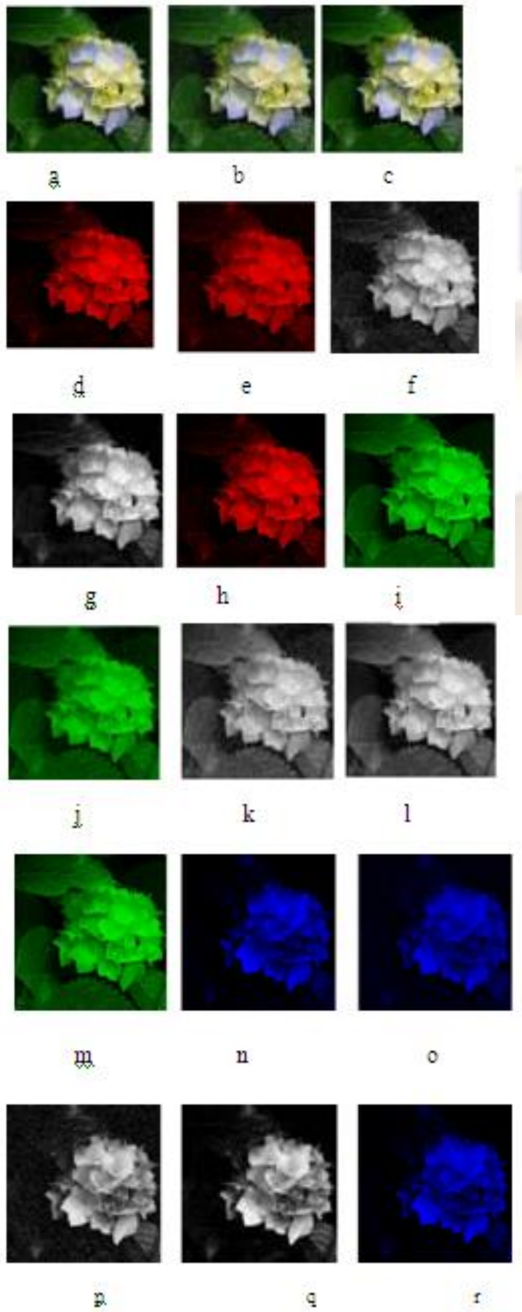

Fig. 5. (a) Original image (b) Noisy image affected by 10% salt and pepper noise image (c) Denoised image by proposed algorithm (d) Original red plane image (e) Noisy red plane image (f) Noisy gray red plane image (g) Denoised gray red plane image (h) Denoised red plane image (i) Original green plane image (j) Noisy green plane image (k) noisy gray green plane image (l) Denoised gray green plane image (m) Denoised green plane image (n) Original blue plane image (o) Noisy blue plane image (p) Noisy gray blue plane image (q) Denoised gray blue plane image (r) Denoised blue plane image.

Fig. 6. (a),(b) image corrupted by 30% of salt and pepper noise and its denoised image (c),(d) image corrupted by 50% of noise and its denoised image (e),(f) image corrupted by 70% of noise and its denoised image.

Fig.7. (a) Original image, (b) Noisy image

(density 90%), (c) DBF outputs (d) MDBA output (e) MDBUTMF output (f) Proposed algorithm output.

IV. CONCLUSION

In this paper a modified simple edge preserved denoising algorithm for impulse noise removal in color images is presented. The extensive experimental results show that the proposed algorithm achieves excellent performance in terms of quantitative evaluation and visual quality, even the noise ratio is high.

REFERENCES

[1] T. Nodes and N. Gallagher, "Median filters: Some modifications and their properties," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-30, no. 5, pp. 739–746, Oct.1982.

[2] Ho-Ming Lin and Alan"Median filters with Adaptive Length", IEEE transactions of the circuits and systems, vol..35, no.6, june 1988.

[3] W. Luo, "Efficient removal of impulse noise from digital images,"*IEEE Trans. Consum. Electron.*,vol. 52, no. 2, pp. 523–527, May 2006..

[4] W. Luo, "An efficient detail-preserving approach for removing impulse noise in images," *IEEE Signal Process. Lett.*,vol. 13, no. 7, pp. 413–416, Jul. 2006.

[5] K. S. Srinivasan and D. Ebenezer, "A new fast and efficient decision based algorithm for removal of high-density impulse noises," *IEEE Signal Process. Lett.*, vol. 14, no. 3, pp. 189–192, March 2007.

[6] Stefan Schulte , Valerie De Witte, "A Fuzzy Noise Reduction method for color images," *IEEE Trans. Image processing.*, vol. 16, no. 5, May 2007.

[7] Qin Zhiyuan, Zhang Weiqiang , Zhang Zhanmu , Wu Bing , Rui Jie, Zhu Baoshan "A robust adaptive image smoothing algorithm".

[9] GOPI KRISHNA,T. SREENIVASULU REDDY , "Removal of High Density Salt and Pepper Noise Through Modified Decision Based Unsymmetric Trimmed Median Filter," (*IJERA*) Vol. 2, Issue 1, Jan-Feb 2012, pp.090-094

S.Manikandan, O.Uma Maheswari, D.Ebenezer "Adaptive length Recursive weighted median filter with improved performance in impulsive noisy environment" WSEAS transaction on Electronics, issue 3, Vol..1,july 2004.