International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue 3, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

A Fuzzy Approach to Detect and Reduce Impulse Noise - A Review

Swati*, Suresh Kumar
Department of Electronics & Communication Engineering
University Institute of Engineering &Technology
Maharshi Dayanand University, Rohtak, Haryana, INDIA
*Corresponding Author's Email: swatichaudhary7@gmail.com; Tel +91-1662-250080

***Abstract* - Image restoration is an important branch of image processing, which deals with the reconstruction of images by removing noise and blurriness , and making them suitable for human perception.In this paper we describe a new algorithm that is especially developed for reducing all kinds of impulse noise: FIDRM (Fuzzy Impulse noise Detection and Reduction Method). It can also be applied to images having a mixture of impulse noise and other types of noise. The result is an image with very little or no impulse noise so that other filters can be used afterwards. Here nonlinear filtering technique is applied usingtwo steps: an impulse noise detection step and a reduction step that preserves edge sharpness. A fuzzy set impulse noise is constructed by our detection method based on the concept of fuzzy gradient values.After that the filtering method (which is a fuzzy averaging of neighbouring pixels) will use this fuzzy set represented by a membership functions.Basedon the criteria of peak-signal-to-noise-ratio (PSNR) and subjective evaluations we have found experimentally, that the proposedmethod provides a significant improvement on other state-of-the-art methods. FIDRM is not only very fast but also very effective for reducing little as well as very high impulse noise.**

***Keywords* – Image Processing, Fuzzy Filter, Membership Functions, Impulse Noise, Noise Reduction.**

I. Introduction

Image processing is one vital part of signal processing [1], which takes an image as input and produces processed output image or image information. Now-a-days image processing is used in various fields as essential features of technology. Where image is used one major source of information then finding the accurate data is very essential. So here comes the need of image enhancement or improvement. Images can become corrupted during any of the acquisition, pre – processing, compression, transmission, storage and/ or reproduction phases of processing [2] – [5]. Several (fuzzy and non-fuzzy) filters have been studied for impulse noise reduction [6] - [10] in the literature. Noise is usually quantified by the percentage of pixels which are corrupted. Corrupted pixels are either set to the maximum value or have single bits flipped over. In some cases, single pixels are set alternatively to zero or to the maximum value. This is the most common form of impulse noise and is called salt and pepper noise. Nevertheless other types of impulse noise are possible as well.

The paper is organised as follows: The impulse noise model is presented in Section 2. In Section 3 the detection andis discussed. Section 4 presents the filtering step. Experimental results are presented in Section 5 and Section 6 gives the final conclusion.

II. Impulse Noise Model

The two common types of noise in images are impulse (or salt and pepper) noise, and random (or Gaussian) noise. Impulse noise can be expressed by noise density. Random noise can be expressed in terms of its mean and variance values. Noise reduction in images has been one of the common tasks in image processing. Impulse noise is caused by errors in the data transmission generated in noisy sensors or communication channels, or by errors during the data capture from digital cameras. Noise is usually quantified by the percentage of pixels which are corrupted. Corrupted pixels are either set to the maximum value or have single bits flipped over. In some cases, single pixels are set alternatively to zero or to the maximum value. This is the most common form of impulse noise and is called salt and pepper noise. Nevertheless, other types of impulse noise are possible as well. This thesis work is going to provide a new, faster, and more efficient noise reduction method for images corrupted with impulse noise.For the case of impulse noise, most part of an original image is unaltered, and the image is characterized by some corrupted samples that vary drastically. Compared to impulse noise, random noise is a more challenging type of noise, it is important to be able to reduce random noise effectively in images. In image processing [11], various linear and nonlinear filtering methods have been proposed. Linear filtering techniques used for noise reduction in images are characterized by mathematical simplicity and can effectively reduce noise with spectral components that do not overlap with those of an image. However, linear filters cannot effectively reduce impulse noise and have a tendency to blur the edges of an image. In this case nonlinear filter is the better option.

A Monochrome (grayscale) digital image 'O' is often represented by a two-dimensional array where an address (i, j) defines a position in 'O', called a pixel or picture element [12]. In a grayscale (or gray level) image, the only colors are shades of gray. A "gray" color is one in which the red, green, and blue components all have equal intensity in the RGB space, so it is only necessary to specify one single intensity value for each pixel, as opposed to the three intensities needed to specify a pixel in a full color image. Often, the (grayscale) intensity is stored as an 8-bit integer giving 256 possible different shades of

International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue 3, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

gray going from black to white, which can be represented as a [0,255] integer interval. In this interval, we consider several integer values $p_1, p_2, p_3 \ldots\ldots p_n$ with $p_k \neq p_l$ and $n \leq 255$. If O(i, j) denotes the pixel value of the (two-dimensional) image 'O' at position (i, j) , then we can model the occurrence of impulse noise for grayscale image[13] , as

$$A(i,j) = \begin{cases} O(i,j) & \text{with probability} 1 - P_r \\ p_1 & \text{with probability } P_{r1} \\ p_2 & \text{with probability } P_{r2} \\ . & \\ . & \ldots\ldots\ldots\ldots\ldots\ldots \\ p_n & \text{with probability } P_{rn} \end{cases}$$

Where p_r is the probability that a pixel is corrupted, and A is the corrupted image. The value p_{rk} with($k \in \{1,\ldots\ldots,n\}$) indicates the probability [19] that an original image pixel becomes p_k .Of course, the following restriction must be valid: $\sum_{k=1}^{n} p_{rk} = p_r$. In the case of saturated impulse noise (salt and pepper noise), there are only two values p_1 and p_2 , which are the maximum and the minimum pixel value of the considered integer interval (in our case respectively 255 and 0). This definition of impulse noise is a simplification of a more general noise model in which a noisy pixel can take on arbitrary values in the dynamic range according to some underlying probability distribution. Let $O(i, j)$ and $\eta(i, j)$ denote the luminance values of the original and the noisy image , respectively, at position (i, j). Then, we have $A(i,j) = \begin{cases} O(i,j) & \text{with probability } 1 - p_r \\ \eta(i,j) & \text{with probability } p_r \end{cases}$

where $\eta(i, j)$ is an identically distributed, independent random process with an arbitrary underlying probability density function.

III. Detection Method

To determine the impulse noise fuzzy [20] rules are defined. The detection method mainly depends on fuzzy gradient values [15]. When the impulse noise is detected the $p_k(k \in 1\{1,2,\ldots\ldots,n\}$ with $1 \leq n \leq 255)$ values are indicated. If a certain pixel is found noisy then the filtering process is to be done by defining some parameters over it.

For each pixel (i, j) of the image (that is not a border pixel), we use a 3x3 neighborhood window as shown in .1. Each neighbor with respect to (i, j) corresponds to one direction {NW = north west, N = north, NE = north east, W = west, E = east, SW = south west, S = south, SE = south east}. Each such direction with respect to (i, j) can also be linked to a certain position (also indicated in Fig.1).

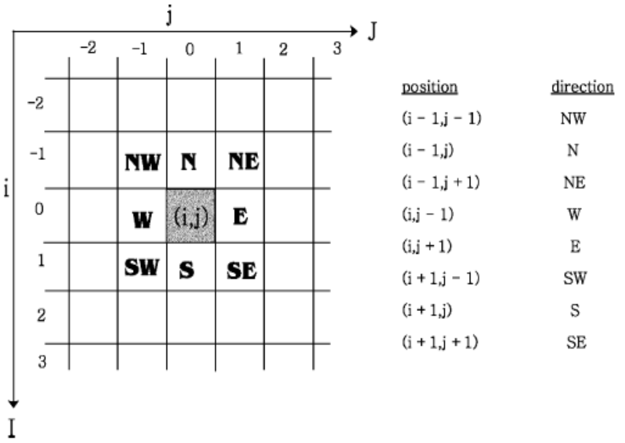


Fig.1. Neighborhood of a central pixel (i, j)

If we denote A as the input image, then the gradient $\nabla_{(k,l)}(i, j)$ is defined as the difference.

$$\nabla_{(k,l)} A(i,j) = a(i+k, j+l) - A(i,j) with\, k,l \in \{-1,0,1\}$$

Where the pair (k,l) corresponds to one of the eight directions and (i, j) is called the center of the gradient. The eight gradient values (according to the eight different directions or neighbors) are called the basic gradient values. One such gradient value with respect to (i, j) can be used to determine if a central pixel is corrupted [16] with impulse noise or not, because if this gradient is quite large then it is a good indication that some noise is present in the central pixel (i, j) , but there are two cases in which this conclusion is wrong.

a) If the central pixel is not noisy, but one of the neighbors is, then this can also cause large gradient values.
b) An edge in an image causes some kind of natural large gradient values.

To handle the first case, we use not only one gradient value, but eight different gradient values[17] (according to the eight different directions) to make a conclusion; to solve the second case, we use not one basic gradient for each direction, but one basic and two related gradient values for each direction. The two related gradient values in the same directionare determined by the centersmaking a right angle with the direction of the first gradient [18].

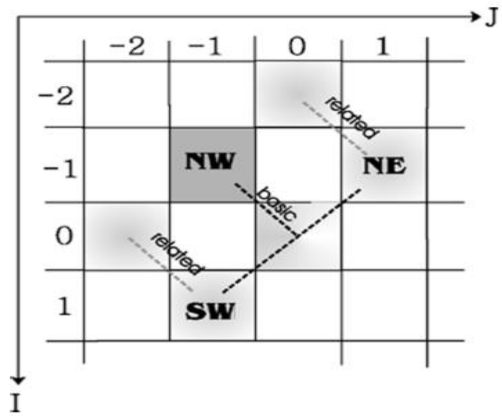


Fig.2. Involved centers for the calculation of the related gradient values in the NW direction.

International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue 3, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Table 1: Related Gradient Values to calculate the Fuzzy Gradient

R	basic gradient	related gradients
NW	$\nabla_{NW}A(i,j)$	$\nabla_{NW}A(i+1,j-1), \nabla_{NW}A(i-1,j+1)$
N	$\nabla_{N}A(i,j)$	$\nabla_{N}A(i,j-1), \nabla_{N}A(i,j+1)$
NE	$\nabla_{NE}A(i,j)$	$\nabla_{NE}A(i-1,j-1), \nabla_{NE}A(i+1,j+1)$
E	$\nabla_{E}A(i,j)$	$\nabla_{E}A(i-1,j), \nabla_{E}A(i+1,j)$
SE	$\nabla_{SE}A(i,j)$	$\nabla_{SE}A(i-1,j+1), \nabla_{SE}A(i+1,j-1)$
S	$\nabla_{S}A(i,j)$	$\nabla_{S}A(i,j-1), \nabla_{S}A(i,j+1)$
SW	$\nabla_{SW}A(i,j)$	$\nabla_{SW}A(i-1,j-1), \nabla_{SW}A(i+1,j+1)$
W	$\nabla_{W}A(i,j)$	$\nabla_{W}A(i-1,j), \nabla_{W}A(i+1,j)$

In the NW-direction [i.e., for $(k,l)=(-1,-1)$] we calculate the basic gradient value $\nabla_{(-1,-1)}A(i,j)$ plus the two related gradient values $\nabla_{(-1,-1)}A(i-1,j+1)$ and $\nabla_{(-1,-1)}\ A(i+1,j-1)$. The two extra gradient values are used for making the separation between noisy pixels and edge pixels: when all these gradients are large, then (i,j) is considered to be not a noisy but an edge pixel. There is an overview of the involved gradient values: each direction R (column 1) corresponds to a position (Fig.2) with respect to a central position. Column two gives the basic gradient for each direction; column three gives the two related gradients.

Finally, eight gradient values are defined for each of the eight directions. These values indicate in which degree the central pixel can be seen as an impulse noise pixel. The fuzzy gradient value $\nabla_R^F A(i,j)$ for direction $R(R\in$ {NW, N, NE, E, SE, S, SW, W}, is calculated by the following fuzzy rule:

If $|\nabla_R A(i,j)|$ *is* large AND $|\nabla'_R A(i,j)|$ *is small*

Or

If $|\nabla'_R A(i,j)$ is large AND $|\nabla''_R A(i,j)$ is small

Or

$\nabla_R A(i,j)$ is big positive AND $\nabla'_R A(i,j)$ *AND* $\nabla''_R A(i,j)$ are big negative

Or

$\nabla_R A(i,j)$ is big negative AND $\nabla'_R A(i,j)$ *AND* $\nabla''_R A(i,j)$ are big positive

Then $\nabla_R^F A(i,j)$ is large

Where $\nabla_R A(i,j)$ is the basic gradient value and $\nabla'_R A(i,j)$ and $\nabla''_R A(i,j)$ are the two related gradient values for the direction. "large", "small," "big negative," and "big positive" are nondeterministic features, these terms can be represented as fuzzy sets. Fuzzy sets can be represented by a membership function. Examples of the membership functions LARGE (for the fuzzy set *large*), SMALL (for the fuzzy set *small*), BIG POSITIVE (for the fuzzy set *big positive*), and BIG NEGATIVE (for the fuzzy set *big negative*) are shown in Fig. 3.

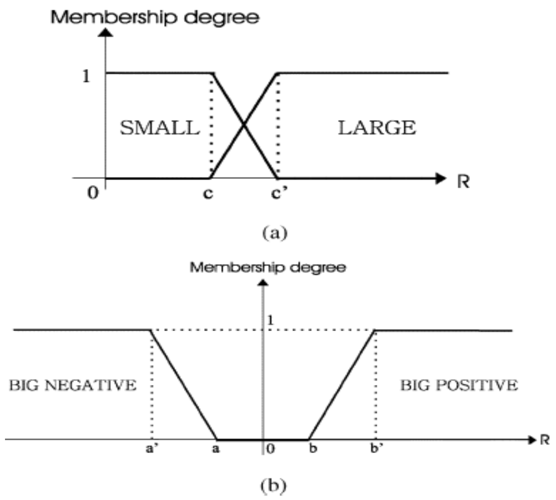


Fig.3. Membership functions (a) SMALL, respectively, LARGE

(b) BIG NEGATIVE, respectively, BIG POSITIVE

The horizontal axis represents the UOD(Universe of Discourse[-255,255] and the vertical axis represents the membership degree[0 to 1].If the value of the membership degree for the fuzzy set large is one ,it means it is large for sure.

These above mentioned fuzzy rule can be transformed like this:

$$\min(LARGE \quad \nabla'_R A(i,j), SMALL(\nabla'_R A(i,j)))$$

Where LARGE and SMALL are membership functions which depend on the two parameters c and c'. The value of these parameters can be determined on basis of some observations.

If Gradient value for a given direction R lies in the range of 0 to 40,the pixels are non-noisy and non-edge pixels and it is considered as the zero membership degree, means noise free pixels. If gradient values lie between [40,125] the pixels are considered to be noise pixels most likely. The membership degree for this range of gradient value lies between zero and one. If the gradient values lies in interval [125,255], the value of membership degree for this range of pixel is one and the pixels are noisy.

To implement this method, the value of c lies in interval 50 to 80 and c' lies in interval 100 to 150.If the basic gradient and the related gradients both are large but have different signs then it indicates that noise is present and the fuzzy set used for this method is big positive and big negative as shown in fig. 3.Gradient values around zero are seen as more or less unsigned and gradient values above 15 or under -15 become significant to matching the feature big positive, respectively, big negative. Therefore the parameters for this method are as $a\in[-15,-10]$, $b\in[10,15]$, $a'\in[-15,-10]$ and $b'\in[15,25]$. After that choosing a T-norm (triangular norm) and a S-norm (triangular co-norm), the activation degree can be calculated by an "IF-THEN" rule. This activation degree also indicates the degree in which ∇_R^F can be considered as large. So the calculated activation degree will be used as a membership degree for ∇_R^F in the fuzzy set *large*.

International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue 3, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

To decide if a central pixel (a nonborder pixel of course) is an impulse noise pixel, The following (fuzzy) rule is used:

If most of the eight $\nabla_R^F A(i,j)$ are *large*

Then the central pixel $A(i,j)$ is an impulse noise pixel. This rule can be translate by: if for a certain central pixel more than half of the fuzzy gradient values (thus more than four) are part of the support of the fuzzy set large, then conclusion is that this pixel is an impulse noise pixel. The support of a certain fuzzy set F is the crisp set of all points in the Universe of Discourse U such that the membership function of F is nonzero. If a pixel (i,j) is detected as an impulse noise pixel, then the corresponding grayscale value is stored in a histogram as shown in result section. This histogram indicates the amount of noise detections (vertical axis) for each possible grayscale value (horizontal axis).This histogram (which is called the noise histogram) is used to investigate the presence of impulse noise. If the noise histogram contains some peaks, then we conclude that the image contains impulse noise pixels, otherwise the image is noise free If the noise is of mixed type (Gaussian and Impulse both) then the histogram contains the two extreme values or two peaks, otherwise in case of impulse noise histogram contains a single peak.

Finally by this procedure, it is decided that an image A is impulse noise-free if the maximum value (the maximal y value) of the noise histogram (for image A) contains less than threshold value of the total detections. The threshold value should be situated in the interval [2.5;10] because when the maximum value is lower than 2.5% the peaks cannot be detected. On the other hand, in images corrupted with mixed types of noise (including impulse noise) a threshold value above 10% could lead to the wrong conclusion that there is no impulse noise present.

Algorithm for the “Impulse Noise” function is as follows :

Function Impulse Noise (real TOT, matrix HIST)
cum = 0
FOR k =1 to 5
kmax = select the Kth greatest y-value of the noise histogram
kpos = the corresponding grayscale value where the maximum is reached
IF (kmax/TOT)+cum $\geq$THR2
p_k = kpos
set the 4 parameters (a_k, b_k, c_k, d_k) to (p_k, p_k, p_k, p_k)
cum = cum+(kmax/TOT)
ELSE
break
END IF
END FOR

END Function

The function “ImpulseNoise” determines the integer values $p_k(k \in \{1,..,n\})$ [used in (1)]. There is need to determine maximal five such integer values to make the filtering step more robust against over filtering. If an image is corrupted with impulse noise, with more than five such p_k values than there is need to restart the complete method when the filtering method has finished. The threshold value THR2 , which is related to THR1(THR2 $\leq$/100 , and THR2 $\in$ [0.025,0.1]). This threshold value is used to select the quantity of noise integers p_k , that agrees with the amount of peaks in the histogram.

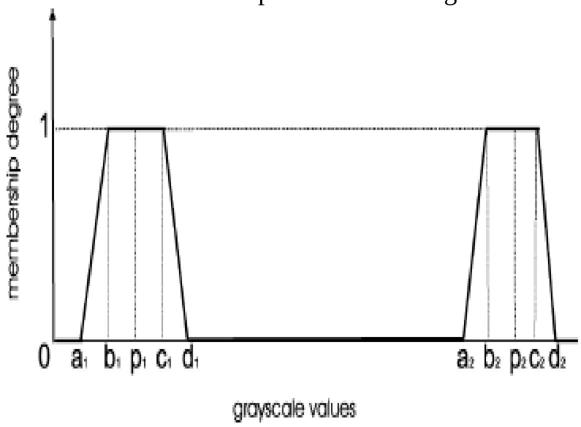


Fig.4. Membership function representing the fuzzy set “more or less impulse noise”.

IV. Filteration Method

After the detection of impulse noise by above procedure the filtering is done, otherwise the image is left unchanged. For this there is need to calculate first the 4 parameters (a_k, b_k, c_k, d_k) which are used to construct the fuzzy set more or less impulse noise. Afterward the iteration process is done for the filtering phase based on the membership functions that represents the fuzzy set. The obtained membership functions is a simplification of the obtained noise histogram. The rules for calculating the parameters are as follows :

$$a_k = p_k - THR_a, \quad b_k = p_k - THR_b$$
$$c_k = p_k + THR_c, \quad d_k = p_k + THR_d$$
$$THR_b = \frac{2}{3} THR_a, \quad THR_c = \frac{2}{3} THR_d$$

with $THR_a = THR_d = \min(25,[\sigma])$ *where* $[\sigma]$ is the largest integer value smaller than the variance σ . These rules are used to approximate the noisy histogram. As the value of variance is unknown ,at first the variance is accurately estimated. when $[\sigma]$ >25,then THRa and THRd is restricted to be 25 to prevent overfiltering, the extreme large value cause wide membership function that results some kind of blurring of the image.

(a) Filtering Step of First Iteration

The filtering step of first iteration is given in the algorithm. This method is based on the membership function “more or less impulse noise” (μ) . The corresponding degree of a certain intensity value $A(i,j)$ is denoted as $\mu(A(i,j))$. A degree one indicates that the intensity value is noisy for sure and the value zero indicates not noisy image for sure. When the degree is between one and zero, then there is some kind of uncertainty. The pixels which are the part of the support of the fuzzy set more or less impulse noise have to be filtered, otherwise leave pixels unchanged. A 3x3 window around the filtered pixel is used. If the output image is the same as the input image (A) then the filter method is called recursive; otherwise non-recursive.

(b) Algorithm for the first iteration

INPUT: A: The noisy image with impulse noise $\mu(A(i,j))$: the membership degree for the fuzzy set more or less impulse noise.

F: the output image.

Steps:

(1) FOR each border pixel $(i,j)\in A$

(2) IF $A(i,j)\in \text{supp}(\text{"more or less impulse noise"})$

(3) $$F(i,j)=\frac{\sum_{h=-1}^{1}\sum_{l=-1}^{1} 1-\mu(A(i+h,j+l))A(i+h,j+l)}{\sum_{h=-1}^{1}\sum_{l=-1}^{1} 1-\mu(A(i+h,j+l))}$$

(4) ELSE

(5) $F(i,j)=A(i,j)$

(6) END IF

(7) END FOR

After the first iteration, it is possible as a side effect (especially with high initial impulse noise) that the impulse noise is clustered around one or more pixels. To reduce these noisy pixels, some more (recursive) iterations are provided that are quite similar to the first one. In each iteration, we use the modified image of the previous performed iteration and a different window as shown in Fig. 5 around a given pixel. Fig.5 shows the neighborhood windows used in the first, second, third, and fourth iteration. The changing window is used to avoid future clustering and also speeds up the execution time. In addition to the different window the modification of the membership function "more or less impulse noise" (for the mth iteration) is also done by changing the parameters.

$$a_k^m=\frac{1}{2}(a_k^{m-1}+p_k),\qquad b_k^c=\frac{1}{2}b_k^{m-1}+p_k$$

$$c_k^m=\frac{1}{2}(c_k^{m-1}+p_k),\qquad d_k^c=\frac{1}{2}d_k^{m-1}+p_k$$

This change is going to reduce the slope of the membership functions, and therefore the amount if investigated pixels for an image A, It will speed up the execution time cause the amount of noisy pixels was already reduced in the previous iteration.

(c) Algorithms for next filtering iteration ($m\geq 2$)

INPUT: F : The output image of the previous iteration $\mu_m(F(i,j))$: the membership degree for the fuzzy set more or less impulse noise for the mth iteration(m≥ 2)

Steps:

(1) FOR each border pixel $(i,j)\in A$

(2) IF $F(i,j)\in \text{supp}(\text{"more or less impulse noise"})$

(3) $$F(i,j)=\frac{\sum_{h=-1}^{1}\sum_{l=-1}^{1} 1-\mu(A(i+h,j+l))A(i+h,j+l)}{\sum_{h=-1}^{1}\sum_{l=-1}^{1} 1-\mu(A(i+h,j+l))}$$

(4) ELSE

(5) $F(i,j)=A(i,j)$

(6) END IF

(7) END FOR

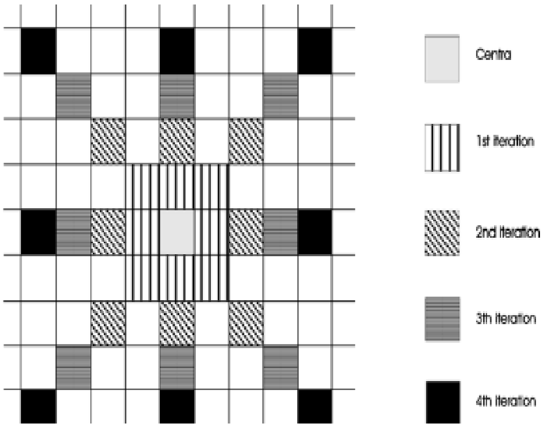


Fig.5. Used neighborhood window for the first, second, third, and fourth iteration.

To stop the iteration procedure there should be some criteria. During the first iteration every pixel is checked one by one. If the pixel value does not depend belong to the support of the fuzzy set more or less impulse noise then this pixel value is not going to changed, not only in this iteration neither in other ones. By remembering only the positions of pixels, whose pixel value is an element of the support of the fuzzy set *more or less impulse noise*, one can drastically reduce the scanning amount in the next iterations.

Suppose the #m is the amount of pixel values which belong to the support of the fuzzy set more or less impulse noise in the mthiteration, thenthe stop criteria is like this :

1. There are no pixel values in the support of the fuzzy set (#m=0) in any iteration m(m≥ 2)
2. #m is equal to #m-1.This means that the resulting pixels are not noisy even with nonzero membership degrees.
3. Since #1 $\geq$#2 ……..$\geq$ #m-1$\geq$ #m hold ∇_m can define as ∇_m = (#m-1) - (#m).If ∇_m is very small then iteration can be stopped too.

By applying these rules the iteration may stop and the filtered result can be obtained.

V. Results of FIDRM

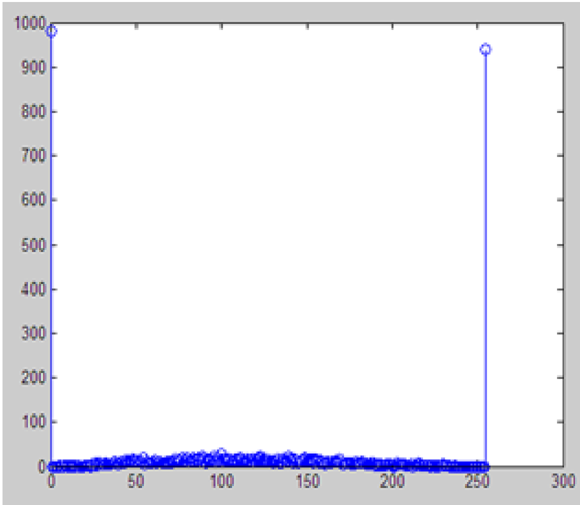
To show the result of this method two figures have been taken. One is Lena image and second one is mandrill baboon as shown in fig. 6. For the sake of easy calculation of corrupted pixels the image is transformed in form of 200x200 pixels.

Fig.6. (a) Lena image (noise free)

Fig.6. (b) Mandrill image(noise free)

After adding some randomly generated noise (5%) the images look like as under:

Histogram produced by Lena image (having a single peak showing presence of impulse noise) for 5% noise

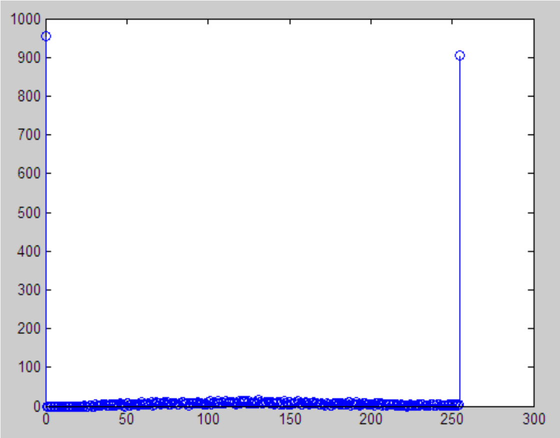


The Lena image after filteration process using FIDRM method (with 5% noise) is as:

PSNR=26.789(dB)

The histogram produced by Mandrill image (extreme peak showing impulse noise) for 5% noise

International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue 3, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

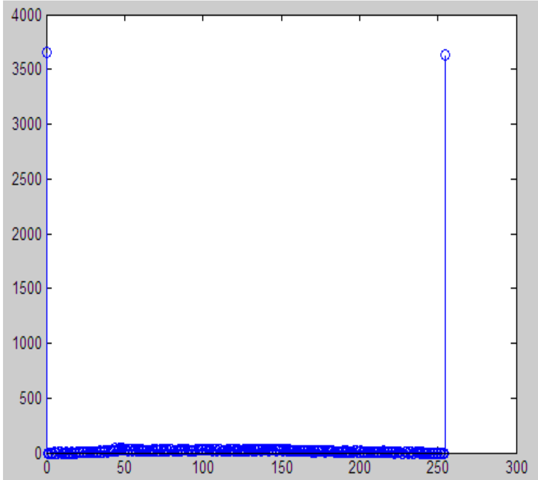


Mandrill image after filtering (for 5% noise) is as :

PSNR=27.019(dB)

When 20% noise is added on Lena image(256x256) and Mandrill image the image looks like as and the results are as under :

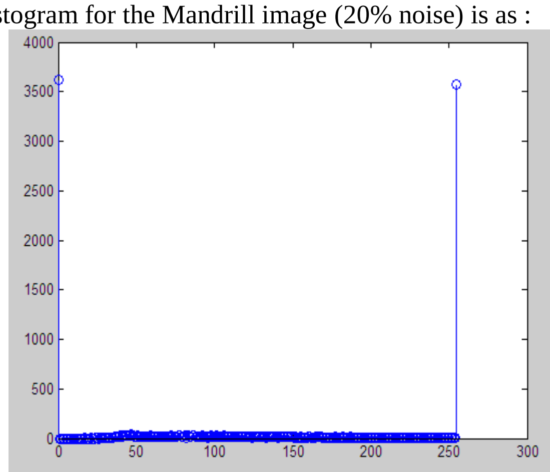
Lena and Mandrill after addition of 20% noise



Histogram for Lena image (for case of 20% noise)

Filtered Lena image (20% noise)
PSNR= 22.35

Histogram for the Mandrill image (20% noise) is as :

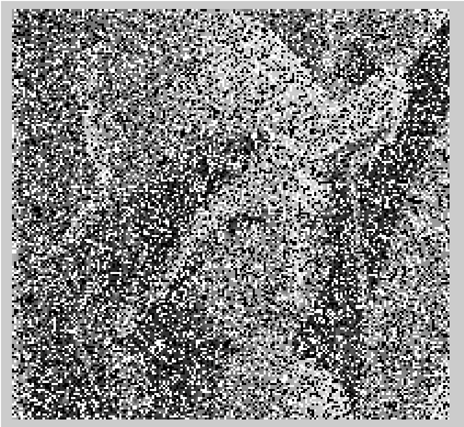


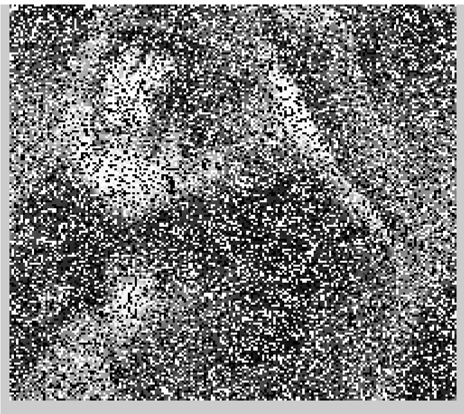
The Mandrill image after filteration (20% noise) is

PSNR =23.06(dB)

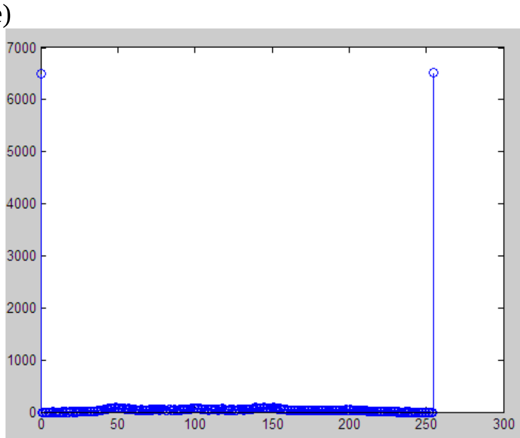
International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue 3, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Lena image and Mandrill image after addition of 50% noise looks like as :



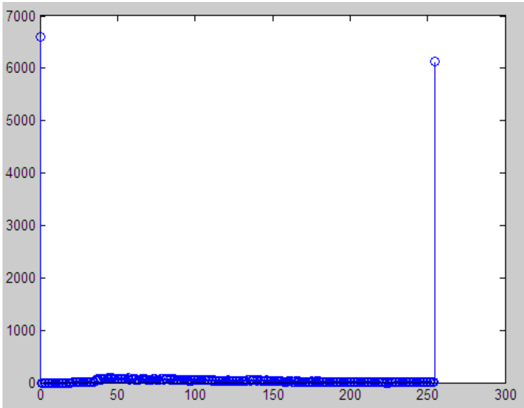


Histogram produced by the Lena Image(in case of 50% noise)

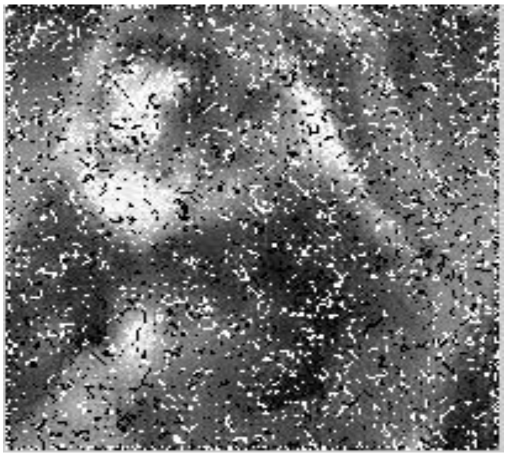


After filteration the Lena image (in case of 50% noise) looks like as under :

Histogram produced by the Mandrill image (in case of 50% noise)



After filteration the Mandrill image (for 50% noise) looks like as under :



5.1 Comparison with existing filters

When the performance of the various filter is compared ,it is found that the PSNR value is increased in this proposed method as compared to the other existing filters, The filtered image shows the difference too. The 2 case is compared (1) For Mandrill image (512x512) and (2) For Lena image(256x256)

Case 1: Mandrill image (512x512)

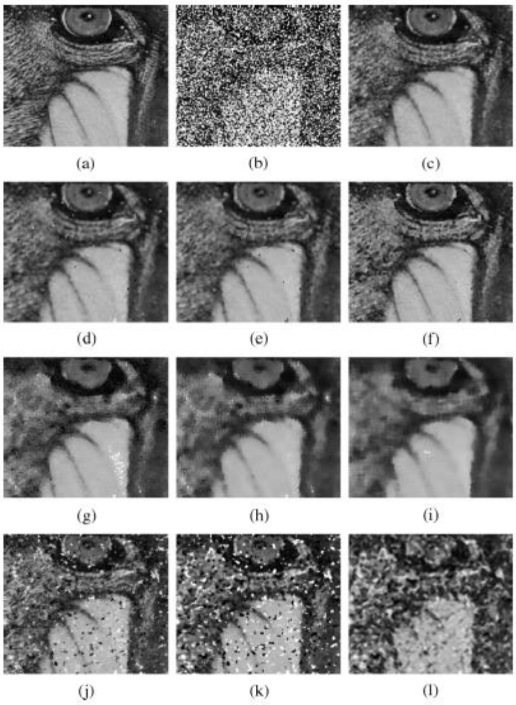


Fig.7: Restoration of a noisy (512x512) Mandrill image.(a) The increased noise-free Mandrill, (b) image contaminated with 50% impulse noise (salt and pepper noise), (c) FIDRM, (d) HAF, (e) ATMAV, (f) AWFM, (g) TSM (7x7), (h) CWM (7x7), (i) basic MEDIAN (7x7), (j) DSFIRE, (k) LUM, and (l) IFCF.

Comparison of PSNR(dB) for the Mandrill image (512x512) for different filters for different levels of impulse noises is as shown in table 2. The results show that this process is much more improved version than others existing filters.

Table 2: Comparison of PSNR for Mandrill (512x512)

	PSNR (dB)		
	5%	25%	50%
Original	18.5	11.5	8.5
CWM(3*3)	26.8	21.5	14.3
TSM(3*3)	29.0	21.7	14.3
LUM	23.2	20.4	14.3
MED(3*3)	20.2	21.0	14.4
ATMAV	22.8	21.2	18.9
FSB	22.1	20.1	19.5
HAF	22.4	22.1	20.7
AWFM	22.6	22.3	21.3
SFCF	22.8	19.7	14.1
EIFCF	23.5	20.0	15.4
MIFCF	23.6	20.0	15.3
IFCF	23.6	20.0	15.5
FIRE	24.3	18.8	12.8
FMF	27.7	20.9	14.4
DSFIRE	30.3	24.6	16.6
PWLFIRE	32.6	19.6	12.6
FIDRM	34.4	27.0	23.2

Case 2: Lena image (256x256)

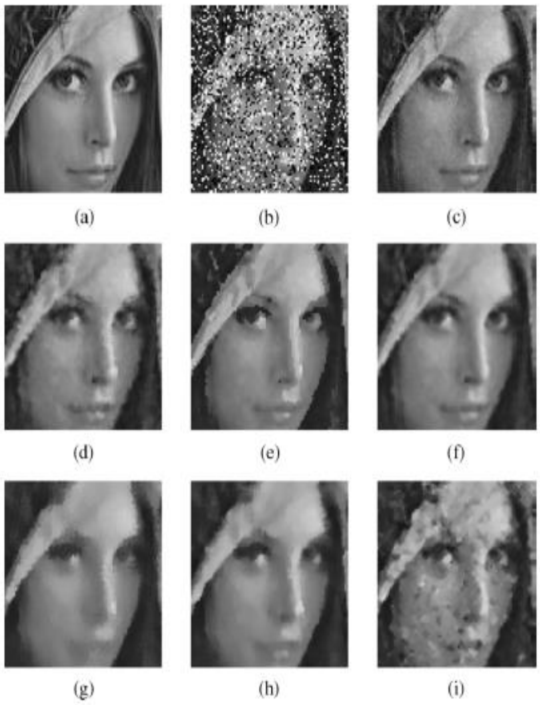


Fig.8: Restoration of a noisy (256x256) Lena image. (a) The increased noise-free Lena, (b) image contaminated 30% impulse noise (salt and pepper noise), (c) FIDRM, (d) ATMAV, (e) AWFM, (f) HAF, (g) CWM (7x7), (h) basic MED (5x5), and (i) EIFCF.

Comparison of PSNR values for Lena image of 256x256 for different filters while impulse noise is 5%, 25% and 50% shows that the result in case of FIDRM is much improved and better.

Table 3: Comparison of PSNR values for Lena image (256x256)

	PSNR (dB)		
	5%	25%	50%
Original	18.6	11.6	8.6
CWM(3*3)	29.4	26.8	24.3
TSM(3*3)	35.4	25.6	24.8
LUM	31.9	25.1	18.4
MED(3*3)	29.7	24.9	15.1
ATMAV	24.1	23.5	20.2
FSB	29.4	24.7	14.9
HAF	29.1	28.4	25.6
AWFM	30.3	29.5	26.9
SFCF	29.1	22.3	14.7
EIFCF	28.9	24.0	17.5
MIFCF	29.4	24.9	17.6
IFCF	29.2	24.1	17.6
FIRE	31.8	21.1	13.5
FMF	34.3	25.5	15.7
DSFIRE	35.7	28.5	17.9
PWLFIRE	36.5	20.4	12.6
FIDRM	40.7	33.4	29.0

VI. Conclusion

A new two step filter (FIDRM), which uses a fuzzy detection and an iterative filtering algorithm, has been presented. This filter is especially developed for reducing all kinds of impulse noise (not only salt and pepper noise). Its main feature is that it leaves the pixels which are noise-free unchanged. Experimental results show the feasibility of the new filter. A numerical measure, such as the PSNR, and visual observations show convincing results for greyscale images. Finally, this new method is easy to implement and has a very low execution time.

References

[1] T. S. Huang, Ed., Two Dimensional Digital Signal Processing II: Transforms and Median Filters. New York: Springer-Verlag, 1981.
[2] R.C.Gonzalez and R.E.Woods, Digital Image Processing, Addison- Wesley, 1992.
[3] A. M. Mirza, A. Chaudhary and B. Munir, Spatially adaptive image restoration using fuzzy punctual kriging, Jouranal of Computer Science and Technology, vol.22, no.4, pp.580-589, 2007.
[4] X.-H. Han, Y.-W.Chen and J.-M. Lei, A spatio—hromatic ICA based noise reduction in color images, International Journal of Innovative Computing, Information and Control, vol.4, no.3, pp.661-669, 2008.
[5] H. Zheng and O. Hellwich, VQ-based Bayesian estimation for blur identification and image selection in vedio sequences, International Journal of Innovative Computing, Information and Control, vol.2, no.2, pp.399-410, 2006.
[6] E. E. Kerre and M. Nachtegael, “Fuzzy Techniques in Image Processing,” 1st ed., vol. 52, Heidelberg, PhysicaVerlag, 2000, 429 pages.
[7] M. Nachtegael, D. Van der Weken, D. Van de Ville and E. E. Kerre, “Fuzzy Filters for Image Processing,” 1st ed., vol. 122, Heidelberg, PhysicaVerlag, 2003, 386 pages.
[8] S. J. Ko and Y. H. Lee, “Center weighted median filters and their applications to image enhancement,” IEEE Transactions on Circuits and Systems, vol. 38, pp. 984-993, Sep. 1991.
[9] T. Chen, K. K. Ma and L. H. Chen, “Tri-state median filter for image denoising,” IEEE Transactions on Image Processing, vol. 8, pp. 1834- 1838, Dec. 1999.
[10] R. C. Hardie and C. G. Boncelet, “LUM filters: a class of rank-orderbased filters for smoothing and sharpening,” IEEE Transactions on Signal Processing, vol. 41, pp. 1834-1838, Mar. 1993.
[11] A. K. Jain, “Fundamentals of Digital Image Processing”, Prentice Hall of India, First Edition, 1989
[12] Rafael C. Gonzalez and Richard E.Woods, “Digital Image Processing”, Pearson Education, Second Edition, 2005.
[13] Arthur R. Weeks, Jr., “Fundamentals of Electronic Image Processing”, Prentice Hall of India, First Edition, 1999.
[14] G. Pok, J. C. Liu and A. S. Nair, “Selective removal of impulse noise based on homogeneity level information”, IEEE Trans. Image Processing, vol. 12, no. 1, pp. 85-92, Jan. 2003
[15] M. Tulin YILDIRIM, Alper BASTURK and M. Emin YUKSEL, “A detail-preserving type-2 fuzzy logic filter for impulse noise removal from digital images”, Proc. of IEEE international conference on fuzzy systems, 2007
[16] WebinLuo, “An efficient algorithm for the of impulse noise from corrupted images”, Int. J. Electron. Commun. (AEU) vol. 61, pp. 551-555, Feb
[17] L. A. Zadeh, "Fuzzy sets," Information and Control, vol. 8, pp. 338-352, 1965.
[18] G. Voit, "Fuzzy Logic in C", Dr.DobbYs Journal, pp.40 -94, Feb. 1993.