IRACST – Engineering Science and Technology: An International Journal (ESTIJ), ISSN: 2250-3498, Vol.2, No. 3, June 2012

Retrieving approximately concrete image from the Blurred image using Fuzzy Logic

Sachin Garg[1], Sapna Aggarwal[2], Vishal Pahal[3]

Department of Computer Science & Engineering
Jind Institute of Engineering & technology.
Jind, Haryana, India.

Abstract— **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 paper work is going to provide a new, faster, and more efficient noise reduction method for images corrupted with impulse noise. So the main objectives of this paper work is to get the almost actual image from the corrupted impulse noised image using fuzzy logic i.e. image enchantment.**

Keywords— **Image processing, Fuzzy logic, noise, impulse, noise reduction method**

I. INTRODUCTION

The process of receiving and analyzing visual information by the human species is referred to as sight, perception or understanding. Similarly, the process of receiving and analyzing visual information by digital computer is called *digital image processing*. One of the first applications of digital images was in the newspaper industry, when pictures were first sent by submarine cable between London and New York. The term digital image processing refers to processing of a two dimensional picture by a digital computer. In other words, it implies digital processing of any two dimensional data. A digital image is an array of real or complex numbers represented by a finite number of bits.

An image given in the form of a transparency, slide, photograph, and chart is first digitized and stored as a matrix of binary digits in computer memory. The digitized image can then be processed on a high resolution television monitor. For display, the image is stored in a rapid access buffer memory which refreshes the monitor at 30 frames per second to produce a visibly continuous display.

The principal source of noise in digital images arise during image acquisition (digitization) or transmission. The performance of imaging sensors is affected by a variety of factors, such as the environmental conditions during image acquisition, and by the quality of the sensing elements themselves. For instance, in acquiring images with a camera, light levels and sensor temperature are major factors affecting the amount of noise in the resulting image. Images are also corrupted during transmission principally due to interference in the channel used for transmission. For example, an image transmitted using a wireless network might be corrupted as a result of lighting or other atmospheric disturbance. There are various types of noise can be added in image.

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.

This new filter has two separated steps or phases: the detection phase and the filtering phase. The detection phase uses fuzzy rules to determine whether a pixel is corrupted with impulse noise or not. When impulse noise is detected, some parameters will be determined which will be passed to the filtering phase. After this detection, the fuzzy filtering technique focuses only on the on the real noisy pixels.

A digital image $a[m, n]$ described in a 2D discrete space is derived from an analog image $a(x, y)$ in a 2D continuous space through a sampling process that is frequently referred to as digitization. The 2D continuous image $a(x, y)$ is divided into *N rows* and *M columns*. The intersection of a row and a column is termed a *pixel*. The value assigned to the integer coordinates $[m, n]$ with {m=0,1,2,...,M-1} and {n=0,1,2,...,N-1} is $a[m, n]$. In fact, in most cases $a(x, y)$ which we might consider to be the physical signal that impinges on the face of a 2D sensor is actually a function of many variables including depth (z), color (λ), and time (t). Unless otherwise stated, we will consider the case of 2D, monochromatic, static images.

Local operations produce an output pixel value $b[m=m_o, n=n_o]$ based upon the pixel values in the *neighborhood* of $a[m=m_o, n=n_o]$. Some of the most common neighborhoods are the 4-connected neighborhood and the 8-connected neighborhood in the case of rectangular sampling and the 6-connected neighborhood in the case of hexagonal sampling illustrated in figure.

In Image representation one is concerned with the characterization of the quantity that each picture element represents. An image could represent luminance of objects in a scene, the absorption characteristics of the body tissue, the radar cross section of the target, the temperature profile of the region or the gravitational field in an area. In general, any two dimensional function that bears information can be considered an image.

An important consideration in image representation is the fidelity or intelligibility criteria for measuring the quality of an image or the Performance of processing technique. Specification of such measures requires models of perception of contrast, spatial frequencies, and colors and so on. The fundamental requirement of digital processing is that images be sampled and quantized. The sampling rate has to be large enough to preserve the useful information in an image. It is determined by the bandwidth of the image.

II. Related work

E. Abreu, M. Lightstone, S. K. Mitra, and K Arakawa Rank-Ordered Mean [9]filter. This is an adaptive approach to solve the restoration problem in which filtering is conditioned on the current state of the algorithm. The state variable is defined as the output of a classifier that acts on the differences between the current pixel value and the remaining ordered pixel values inside a window centered around the pixel of interest. This scheme is undoubtedly one of the robust and simple scheme but it fails in preserving the finer details of the image.

Z. Wang and D. Zhang. proposed Progressive-Switching Median [10] filter. It is a median based filter, which works in two stages. In the first stage an impulse detection algorithm is used to generate a sequence of binary flag images. This binary flag image predicts the location of noise in the observed image. In the second stage noise filtering is applied progressively through several iterations. This filter is a very good filter for fixed valued impulsive noise but for random values the performance is abysmal.

T. Chen and H. R. Wu. proposed Adaptive Center Weighted Median Filter [11]. This work is an improvement of previously described Center Weighted Median (CWM) filter. It works on the estimates based on the differences between the current pixel and the outputs of the CWM filters with varied center weights. These estimates decide the switching between the current pixel and median of the window. This is a good filter and is robust for a wide variety of images. But it is inefficient in recovering the exact values of the corrupted pixels.

X. Xu and E. L. Miller. proposed Adaptive Two-Pass Median [12]filter. As the name suggests it employs median filter on the noisy image twice. This adaptive system tries to correct for false replacements generated by the first round of median filtering operation. Based on the estimated distribution of the noise, some pixels changed by first median filter are replaced by their original values and kept unchanged in the second median filtering. And in the second round it filters out the remaining impulses. Even though the filter gives some good results in terms of noise suppression but spoiling of good pixels is more and it results in overall poor performance.

K. Kondo, M. Haseyama, and H. Kitajima proposed Accurate Noise Detector [13]filter. This filter justifies its name by detecting noise to the perfection. Based on Progressive Switching Median Filter, it generates an edge flag image to classify the pixels of noisy image into ones in the flat regions and edge regions. The two types of pixels are processed by different noise detector. When noise is very high prevention of false-detection and non-detection becomes difficult. Therefore, another iteration is dedicated for verification of the noise flag image. This scheme exhibits good performance on images not only with low noise density but also with high percentage of corruption. But all these come at the cost of computational complexity which is very high and not at all suitable for real time applications.

S. Zhang and Md. A. Karim proposed Switching Median) [14] filter. This is also a two stage process, where in the first stage noise detection is carried out and in the second stage filtering is done. The noisy image is convolved with a set of convolution kernels. Each of the kernels are sensitive to edges in a different orientation. The minimum absolute value of these four convolutions is used for impulse detection by comparing with a threshold. By varying the size of kernel different variations of SM may be obtained. Three such variations of SM are reviewed here in this paper. Because of its four kernels it detects noise effectively even in those images where the edge density is more. But when the kernel size increases to 7 × 7 and 9 × 9 it fails in doing so. Also it fails in preserving finer details.

C. Butako and I. Aizenberg proposed Differential Ranked Impulse Detector [15]. This is another nonlinear technique which also works in two stages. It aims at filtering only corrupted pixels. Identification of such pixels is done by comparing signal samples within a narrow rank window by both rank and absolute value. The first estimate is based on the comparison between the rank of the pixel of interest and rank of the median. The second estimate is based on the brightness value which is analyzed using the median. It is a good filter in low noise conditions but the performance slightly degrades in beyond 20% of noise. It also leaves noise blotch without correcting.

III. Background

Fuzzy logic was first introduced in the 1965 as a new way to represent vagueness in everyday life [23, 25]. The definition of fuzzy logic as a superset of conventional(Boolean) logic that has been extended to handle the concept of partial truth values between "completely true" and "completely false". By this definition, fuzzy logic departs from classical two-valued set logic. It uses soft linguistic system variables and a continuous range of true values in the interval [0, 1], rather than strict binary values. It is basically a multivalued logic that allows intermediate values to be defined between conventional evaluations like yes/no or true/false**,** etc. Notions like rather warm or pretty cold can be formulated mathematically and processed by computers."

Fuzzy logic is also a structured, model-free estimator that approximates a function through linguistic input/output associations. Fuzzy logic is a powerful, yet straight forward, problem solving technique with wide spread applicability, especially in the areas of control and decision making[24]. Fuzzy Logic was first invented as a representation scheme and calculus for uncertain or vague notions. It allows more human-like interpretation and reasoning in machines by resolving intermediate categories between notations such as true/false, hot/cold etc used in Boolean logic In this context, Fuzzy Logic is a problem-solving control system methodology that lends itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or workstation-based data acquisition and control systems. It can be implemented in hardware, software, or a combination of both. FL provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. FL's approach to control problems mimics how a person would make decisions, only much faster

In general, fuzzy logic is most useful in handling problems not easily definable by rigorous mathematical models. FL is capable of additional benefits of Fuzzy Logic include its simplicity and its flexibility. Fuzzy Logic can handle problems with imprecise and incomplete data, and it can model nonlinear functions of arbitrary complexity. "If you don't have a good plant model, or if the system is changing, then fuzzy will produce a better solution than conventional control techniques." Fuzzy logic derives much of its power from its ability to draw conclusions and generate responses based on vague, ambiguous, qualitative, incomplete, or imprecise information. In this respect, fuzzy-based systems have a decision reaching similar to that of humans. In fact, the behavior of a fuzzy system is represented in a very simple and natural way which allows quick construction of an understandable, maintainable, and robust system. In addition, a fuzzy approach generally requires much less memory and computing power than conventional methods, thereby resulting in a smaller and less expensive system [26]. Fuzzy logic has many applications such as:

1. Control system (Robotics, Automation, Tracking, Consumer Electronics).
2. Information systems(DBMS, Information retrieval)
3. Pattern recognition (Image Processing, Machine Vision).
4. Decision support (Sensor Fusion).

FL offers several unique features that make it a particularly good choice for many control problems.

1) It is inherently robust since it does not require precise, noise-free inputs and can be programmed to fail safely if a feedback sensor quits or is destroyed. The output control is a smooth control function despite a wide range of input variations.

2) Since the FL controller processes user-defined rules governing the target control system, it can be modified and tweaked easily to improve or drastically alter system performance. New sensors can easily be incorporated into the system simply by generating appropriate governing rules.

3) FL is not limited to a few feedback inputs and one or two control outputs, nor is it necessary to measure or compute rate-of-change parameters in order for it to be implemented. Any sensor data that provides some indication of a system's actions and reactions is sufficient. This allows the sensors to be inexpensive and imprecise thus keeping the overall system cost and complexity low.

4) Because of the rule-based operation, any reasonable number of inputs can be processed (1-8 or more) and numerous outputs (1-4 or more) generated, although defining the rulebase quickly becomes complex if too many inputs and outputs are chosen for a single implementation since rules defining their interrelations must also be defined. It would be better to break the control system into smaller chunks and use several smaller FL controllers distributed on the system, each with more limited responsibilities.

5) FL can control nonlinear systems that would be difficult or impossible to model mathematically. This opens doors for control systems that would normally be deemed unfeasible for automation.

IV. WAY OF APPLYING FUZZY LOGIC

1) Define the control objectives and criteria: What am I trying to control? What do I have to do to control the system? What kind of response do I need? What are the possible (probable) system failure modes?

2) Determine the input and output relationships and choose a minimum number of variables for input to the FL engine (typically error and rate-of-change-of-error).

3) Using the rule-based structure of FL, break the control problem down into a series of IF X AND Y THEN Z rules that define the desired system output response for given system input conditions. The number and complexity of rules depends on the number of input parameters that are to be processed and the number fuzzy variables associated with each parameter. If possible, use at least one variable and its time derivative. Although it is possible to use a single, instantaneous error parameter without knowing its rate of change, this cripples the system's ability to minimize overshoot for a step inputs.

4) Create FL membership functions that define the meaning (values) of Input/Output terms used in the rules.

5) Create the necessary pre- and post-processing FL routines if implementing in S/W, otherwise program the rules into the FL H/W engine.

6) Test the system, evaluate the results, tune the rules and membership functions, and retest until satisfactory results are obtained.

FL requires some numerical parameters in order to operate such as what is considered significant error and significant rate-of-change-of-error, but exact values of these numbers are usually not critical unless very responsive performance is required in which case empirical tuning would determine them. For example, a simple temperature control system could use a single temperature feedback sensor whose data is subtracted from the command signal to compute "error" and then time-differentiated to yield the error slope or rate-of-change-of-error, hereafter called "error-dot". Error might have units of degrees F and a small error considered to be 2F while a large error is 5F. The "error-dot" might then have units of degree/min with a small error-dot being 5F/min and a large one being 15F/min. These values don't have to be symmetrical and can be "tweaked" once the system is operating in order to optimize performance. Generally, FL is so forgiving that the system will probably work the first time without any tweaking.

V. Impulse Noise Detection By Fuzzy Method

To determine the impulse noise fuzzy rules are defined. The detection method mainly depends on fuzzy gradient values. When the impulse noise is detected the $p_k(k \in 1\{1,2,.....,n\} \quad with\, 1 \le n \le 255)$ 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 Fig. 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.3.2.1).

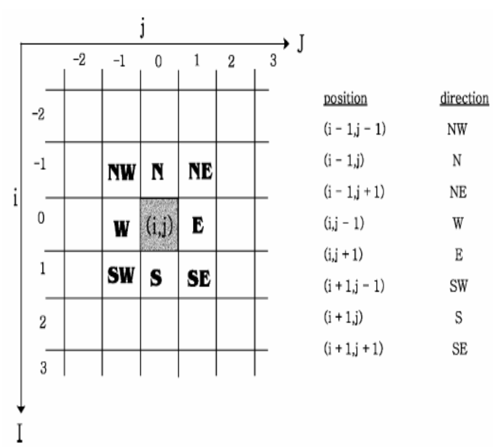

Fig. 1 Neighborhood of a central pixel (i, j)

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

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

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 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 (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 direction are determined by the centers making a right angle with the direction of the first 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	$\nabla_{W} A(i,j)$	$\nabla_{W} A(i-1,j), \nabla_{W} A(i+1,j)$

Table 1 Related Gradient Values to calculate the Fuzzy Gradient

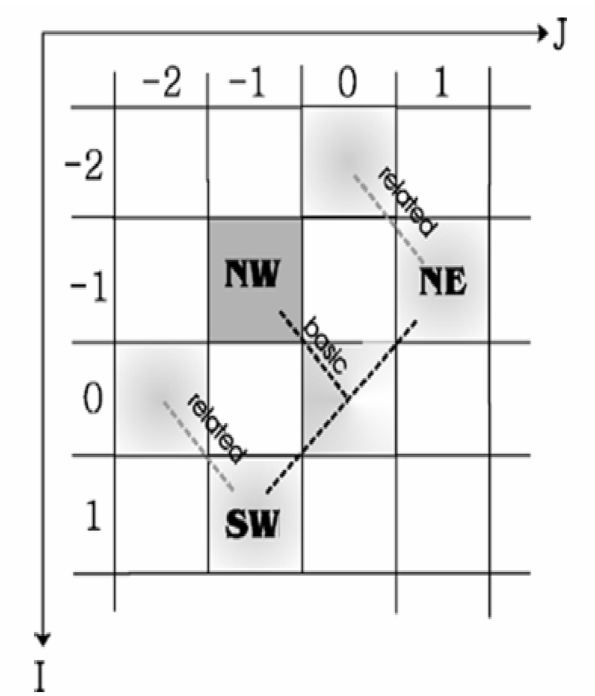

Fig.2 Involved centers for the calculation of the related gradient values in the NW direction.

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.5.1) 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:

$$\textit{If} \quad |\nabla_R A(i,j)| \quad \textit{is} \quad \text{large AND} \, |\nabla'_R A(i,j)| \quad \textit{is small}$$

OR

$$\textit{If} \quad |\nabla'_R A(i,j) \quad \text{is} \quad \text{large AND} \, |\nabla''_R A(i,j) \quad \text{is small}$$

OR

$$\nabla_R A(i,j) \quad \text{is big positive} \quad \text{AND} \quad \nabla'_R A(i,j) \quad \textit{AND} \; \nabla''_R A(i,j) \quad \text{are big negative}$$

OR

$$\nabla_R A(i,j) \quad \text{is big negative AND} \; \nabla'_R A(i,j) \quad \textit{AND} \; \nabla''_R A(i,j) \quad \text{are big positive}$$

$$\text{THEN} \; \nabla_R^F A(i,j) \; \text{is large}$$

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.

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.

Where LARGE and SMALL are membership functions which depends on the two parameters c and c'. The value of this 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 nonnoisy and non-edge pixels and it is considered as the zero membership degree, means noise free pixels.

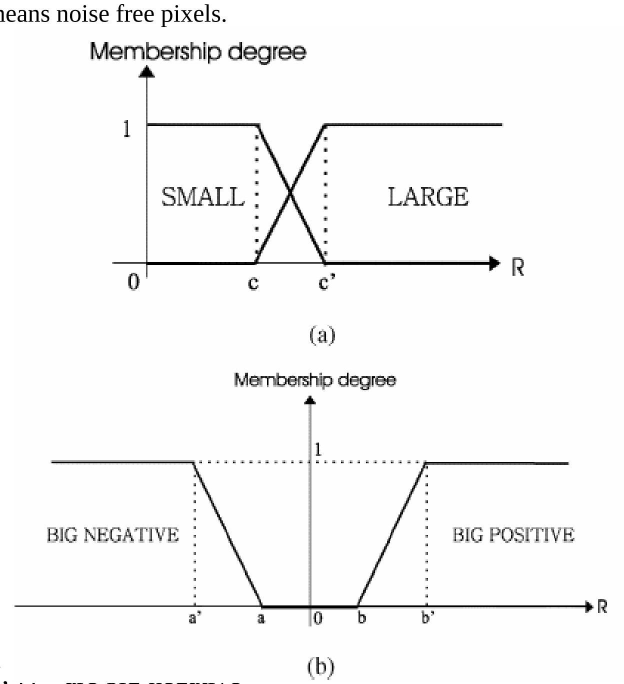

Fig.3 Membership functions (a) SMALL ,respectively, LARGE (b) BIG NEGATIVE, respectively, BIG POSITIVE

These above mentioned fuzzy rule can be transformed like this:

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

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.

VI CONCLUSION & FUTURE WORK

It primarily focuses on impulsive& Gaussian noise suppression from images. 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 grayscale images. But the filter does not gives good result for the Gaussian noise. Finally, this new method is easy to implement and has a very low execution time.

As it has been stated that the proposed technique is not good for Gaussian noise removal , investigation may be carried out in this direction. Development of parallel algorithms can also be done to counter attack the computational overhead.

In this thesis we have used fuzzy logic for noise detection. Investigation may be carried out to use neural network for detection of noisy pixels in the image and fuzzy logic to remove the detected noise from the image.

References

[1] A. K. Jain, "Fundamentals of Digital Image Processing", Prentice Hall of India, First Edition, 1989

[2] Rafael C. Gonzalez and Richard E.Woods, "Digital Image Processing", Pearson Education, Second Edition, 2005.

[3] Arthur R. Weeks, Jr., "Fundamentals of Electronic Image Processing", Prentice Hall of India, First Edition, 1999.

[4] 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

[5] Pei-Eng Ng and Kai-Kuang Ma, "A Switching Median Filter with Boundary Discriminative Noise Detection for Extremely Corrupted Images", IEEE Transactions on Image Processing, vol. 15, no. 6, June 2006.

[6] 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

[7] Webin Luo, "An efficient algorithm for the removal of impulse noise from corrupted images", Int. J. Electron. Commun. (AEU) vol. 61, pp. 551-555, Feb

[8] S. J. Ko and Y. H. Lee. Center Weighted Median Filters and Their Applications to Image Enhancement. IEEE Transactions on Circuits and Systems, 38(9):984 – 993, September 1991.

[9] Images. IEEE Transactions on Image Processing, 5(6):1012 – 1025, June 1996

[10] T. Chen and H. R. Wu. Adaptive Impulse Detection Using Center-Weighted Median Filters. IEEE Signal Processing Letters, 8(1):1 – 3, January 2001

[11] X. Xu and E. L. Miller. Adaptive Two-Pass Median Filter to Remove Impulsive Noise. In Proceedings of International Conference on Image Processing 2002, pages I–808 – I–811, September 2002.

[12] K. Kondo, M. Haseyama, and H. Kitajima. An Accurate Noise Detector for Image Restoration. In Proceedings of International Conference on Image Processing 2002, volume 1, pages I–321 – I–324, September 2002.

[13] S. Zhang and Md. A. Karim. A New Impulse Detector for Switching Median Filters. IEEE Signal Processing Letters, 9(11):360 – 363, November 2002.

[14] V. Crnojevic, V. Senk, and Z. Trpovski. Advanced Impulse Detection Based on Pixel-Wise MAD. IEEE Signal Processing Letters, 11(7):589 – 592, July 2004.

[15] W. Y. Han and J. C. Lin. Minimum-Maximum Exclusive Mean (MMEM) Filter to Remove Impulse Noise from Highly Corrupted Images. Electronics Letters, 33(2):124 – 125, January 1997.

[16] P. S. Windyga. Fast Impulsive Noise Removal. IEEE Transactions on Image Processing, 10(1):173 – 179, January 2001

[17] Naif Alajlan, Mohamed Kamel, and Ed Jernigan. Detail Preserving Impulsive Noise Removal. Signal Processing: Image Communication, 19:993 – 1003, 2004.

[18] E. Abreu and S. K. Mitra. A Signal-Dependent Rank Ordered Mean (SD- ROM) filter. A new approach for removal of impulses from highly corrupted images. In Proceedings of International Conference on Acoustics, Speech, and Signal Processing, volume 4 of ICASSP-95, pages 2371 – 2374, May 1995.