IMPULSE NOISE REMOVAL USING FCM ALGORITHM

Major Project submitted in partial fulfillment of the requirements for the award of degree of

Master of Technology
in
Information Systems

Submitted By:
Isha Singh
(05/IS/10)

Under the Guidance of:
Dr. O. P. Verma

Department of Information Technology
Delhi Technological University
Bawana Road, Delhi – 110042
(2010-2012)

CERTIFICATE

This is to certify that **Ms. Isha Singh (05/IS/10)** has carried out the major project titled "**Impulse Noise Removal using FCM Algorithm**" as a partial requirement for the award of Master of Technology degree in Information Systems by Delhi Technological University. The major project is a bonafide piece of work carried out and completed under my supervision and guidance during the academic session **2010-2012**. The matter contained in this report has not been submitted elsewhere for the award of any other degree.

(Project Guide)
Dr. O.P. Verma
Head of Department
Department of Information Technology
Delhi Technological University
Bawana Road, Delhi-110042.

ACKNOWLEDGEMENT

I express my gratitude to my major project guide **Dr. O.P. Verma**, HOD, IT Dept., Delhi Technological University, for the valuable support and guidance he provided in making this project. It is my pleasure to record my sincere thanks to my respected guide for his constructive criticism and insight without which the project would not have shaped as it has.

I humbly extend my words of gratitude to other faculty members of this department for providing their valuable help and time whenever it was required.

Isha Singh
05/IS/10
M.Tech (Information System)
Department of IT
E-mail: 222ishasingh@gmail.com

ABSTRACT

Image noise removal is an important aspect of image processing. Impulse noise is one such noise, which is frequently introduced into images while transmitting and acquiring them over an unsecure communication channel. Satellite or TV images can be corrupted by atmospheric disturbances. In other applications noise can be introduced by strong electromagnetic fields, transmission errors, etc. Human visual system is very sensitive to the high amplitude of noise signals, thus noise in an image can result in a subjective loss of information. Various techniques have been introduced for the removal of impulse noise based on the properties of their respective noise models. Performance of some recent filters is evaluated and compared to that of the proposed filter.

This study introduces an iterative filter for images highly corrupted with impulse noise, typically in the range 30-80%. It is a novel technique for detecting high density impulse noise and preserves more image details in high noise environment. The application of Fuzzy C-means (*FCM*) algorithm in the detection phase provides optimal results. By extensive simulation results and comparison with other filters, it is observed that the proposed algorithm outperforms several methods.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

LIST OF ACRONYMS

SPN	**Salt & Pepper noise**
p	**Probability**
S	**size of window**
MSE	**Mean Square Error**
PSNR	**Peak Signal to Noise Ratio**
E	**Termination criterion between 0 and 1**
FCM	**Fuzzy C-means**
SMF	**Standard median filter**
MMF	**Min-Max Median Filter**
CWMF	**Center Weighted Media Filter**
AMF	**Adaptive Median Filter**
PSMF	**Progressive Switching Median Filter**
TMF	**Tri-state Median Filter**
DBA	**Decision Based Algorithm**
WM	**Weighted median filter**
SWM	**Switching median filter**
DWM	**Directional weighted median filter**
NAFSM	**Noise Adaptive Fuzzy Switching Median Filter**
FSM	**Fuzzy Switching Median filter**
EEPA	**Efficient edge-preserving algorithm**

MDBUTMF	**Modified Decision Based Unsymmetrical Trimmed Median Filter**
BDND	**Boundary Discriminative Noise Detection**
HEIND	**Highly effective impulse noise detection algorithm**
DPF	**Detail-preserving filter**

Chapter 1

INTRODUCTION

1.1 PRELIMINARIES

An image is worth a thousand words. In the modern age, images are the most common and convenient means of conveying or transmitting information. Visual information in the form of digital images allows humans to perceive and understand the world surrounding them in a better manner. Hence, processing of images by computer has been drawing a very significant attention of the researchers over the last few decades. The process of receiving and analyzing visual information by digital computer is called digital image processing.

Mathematically, an image is a two-dimensional function, $f(x,y)$, where x and y are spatial (plane) coordinates and the amplitude of f at any coordinate (x,y) is called the intensity or gray level of the image at that point. When x, y and the amplitude values are all finite discrete quantities then image is known as a digital image [1]. A digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements, pels or pixels. Pixel is the term most widely used to denote the element of a digital image. [1, 2, 3] A rectangular array of pixels is known as bitmap.

Digital image processing is the use of computer algorithms to perform image processing on digital images. Digital Images can be of different types such as binary, gray-scale and color images [4].

1. **Binary images**: Binary images use only a single bit to represent each pixel. They are the simplest type of images and can take only two discrete values, black and white. Black is represented with the value '0' while white with '1'. This inability to represent intermediate shades of gray limits their usefulness in dealing with photographic images. It finds applications in computer vision areas where the general shape or outline information of the image is needed.
2. **Gray-scale images:** They are known as monochrome or one-color images. A black and white image is made up of pixels each of which holds a single number corresponding to the gray level of the image at a particular location. These gray levels span the full range from black to white in a series of very fine steps. For an 8-bit image there will be 256 gray levels where '0' represents black and '255' denotes white.

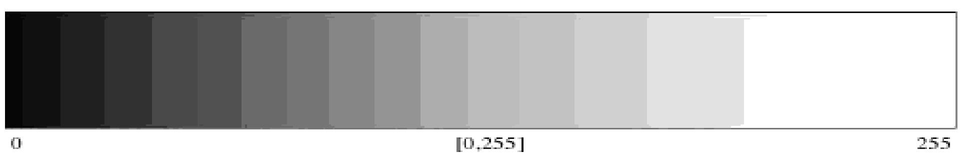


Figure 1.1: Gray scale levels for an 8-bit image

3. **Color Images:** A color image is made up of pixels each of which holds three numbers corresponding to the red, green, and blue levels of the image at a particular location. Red, green, and blue (RGB) are the primary colors for mixing light. Any color can be created by mixing the correct amounts of red, green, and blue light. Assuming 256 levels for each primary, each color pixel can be stored in three bytes (24 bits) of memory. This corresponds to roughly 16.7 million different possible colors.

For images of the same size, a gray scale image will use three times less memory than a color image. In this study, we have used several standard gray scale images for our experimental and simulation results.

For image processing, we need to convert the natural images into digital images by the process of digitization. A digitized image can be stored in a computer memory or on some form of storage media such as hard disk or CD-ROM. This digitization procedure can be done by scanner, or by video camera connected to frame grabber board in computer. Once the image has been digitized, it can be operated upon by various image processing operations. Digital image processing operations can be broadly divided into following classes:

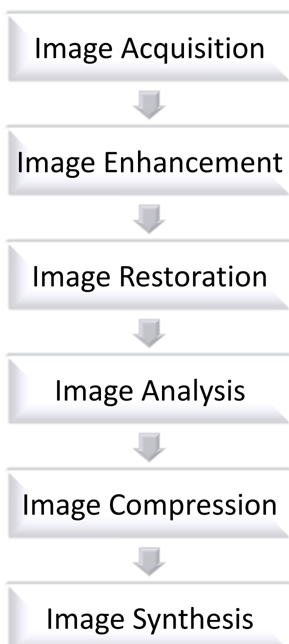


Figure 1.2: Digital image processing classes

Examples of operations within each class are as follows:

1) **Image Enhancement**: Brightness adjustment, contrast enhancement, image averaging, convolution, frequency domain filtering, and edge enhancement.

2) **Image restoration**: Photometric correction, inverse filtering, and noise removal

3) **Image analysis**: Segmentation feature extraction, object classification

4) **Image compression**: Lossless and lossy compression

5) **Image synthesis**: Topographic imaging, 3-D reconstruction

The fields that use digital image processing techniques can be divided into criminology, microscopy, photography, remote sensing, medical imaging, forensics, transportation and military applications.

Out of the five classes of digital image processing, cited above, this thesis deals with image restoration. To be precise, the thesis devotes on a part of the image restoration i.e. impulse noise removal from images, stated in the Problem Definition. Further, this thesis also discusses how image noise removal can be utilized for high quality image enhancement.

1.2 PROBLEM DEFINITION

The basic idea behind this thesis is the restoration of original image from the distorted image corrupted by impulse noise. It is also referred to as image “denoising”. Denoising is the process of removing unwanted noise from an image. A denoised image is an approximation to the underlying true image, before it was contaminated. Image denoising finds applications in fields such as astronomy, medical imaging and forensic science where the physical requirements for high quality imaging are needed for analyzing images of unique events. Therefore noise removal is one of the most important pre processing steps an image should undergo before further image analysis. A good denoising algorithm must preserve the structure and remove noise simultaneously.

Different types of noise frequently contaminate images. Impulse noise is one such noise, which is frequently introduced into images while transmitting and acquiring them due to channel errors or in storage media due to faulty hardware. There are various methods to help restore an image from impulse noise. Some of the techniques are discussed briefly in the following chapters. The main drawback of the several algorithms present in the literature is incapability to deal with high noise density. At high noise level, the edge details of the original image are not preserved which results in blurring of images. In order to overcome this problem, an impulse noise detection mechanism prior to filtering is employed in several algorithms , such as tri-state median filter (TSM) [5], an efficient edge-preserving algorithm (EEPA) [6], fuzzy switching median filter (FSM) [7], noise adaptive fuzzy switching median filter (NAFSM)[8], a highly effective impulse noise detection Algorithm (HEIND)[9], and Contrast Enhancement-Based Filter (CEF) [10], Modified Decision Based Unsymmetrical Trimmed Median Filter Algorithm (MDBUTMF) [11], A New Adaptive Switching Median (ASWM) Filter [12] etc.

This research work provides an efficient algorithm for impulse noise removal. It has a two phase scheme for the restoration of images corrupted by impulse noise. The detection method of the proposed algorithm efficiently identifies the location of noisy pixels so that only corrupted pixels are restored and the value of uncorrupted pixels is left intact. The application of Fuzzy C-means (FCM) algorithm in the detection phase provides optimal results. The derived algorithm is implemented in MATLAB and tested on some standard images. To show the effectiveness and efficiency of our algorithm, the outcome of the implementation is compared with the results of some existing algorithms.

1.3 ORGANIZATION OF THESIS

The rest of this thesis is organized as follows.

In Chapter 2 Impulse noise is discussed in detail. An overview of impulse noise detection and removal with literature survey is presented. Various noise models are also discussed briefly.

Chapter 3 introduces our proposed algorithm for the removal of high noise density using FCM algorithm. The design of the proposed algorithm is explained with the aid of flowchart. FCM algorithm is explained briefly. Further, an illustration of the operation of the algorithm in the form of example is presented.

Chapter 4 provides the comparative results with some recent as well as state of art techniques. The implemented code has been tested on various standard images. Results for varying noise level typically from 30% to 50% are shown for different test images. The quantitative results of comparison are also tabulated by calculating the Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE) of the output image.

Finally in Chapter 5 the conclusion is presented.

Chapter 2

IMPULSE NOISE

The digital image acquisition process converts an optical image into a continuous electrical signal that is, then, sampled [4]. At every step in the process there are fluctuations caused by natural phenomena that adds a random value to the exact brightness value for a given pixel. This process introduces noise in an in image. There are many types of noises that contaminate images. One of such noise is Impulse Noise.

Impulse noise is generally introduced into images while transmitting and acquiring them over an unsecure communication channel. Impulse noise affects images at the time of acquisition due to noisy sensors or at the time of transmission due to channel errors or in storage media due to faulty hardware. Sharp and sudden disturbances in the image signal introduce impulse noise. Its appearance is randomly scattered white or black (or both) pixels over the image.

(a) (b)

Figure 2.1: (a) Original image (b) Image corrupted with impulse noise

Let $Y(i, j)$: Gray level of an original image

$X(i, j)$: Gray level of noisy image X at a pixel location (i, j)

$[\text{Nmin Nmax}]$: Dynamic range of Y

Impulsive noise may be defined as:

$$X(i, j) = \begin{cases} Y(i, j) & \text{with} 1-p \\ R(i, j) & \text{with } p \end{cases} \qquad (1)$$

$R(i, j)$ is the substitute for the original gray scale value at the pixel location (i, j)

Impulse noise has the property of either leaving a pixel unmodified with probability 1- p or replacing it altogether with probability p. This is shown in Eq (1).

Two common types of impulse noise are:

- **Salt & Pepper Noise (SPN) :**

For images corrupted by salt-and-pepper noise, the noisy pixels can take only the maximum and the minimum values in the dynamic range i.e.

When $R(i, j) = \{ \text{Nmin}, \text{Nmax} \}$ (2)

Figure 2.3: Salt & Pepper Noise dynamic range

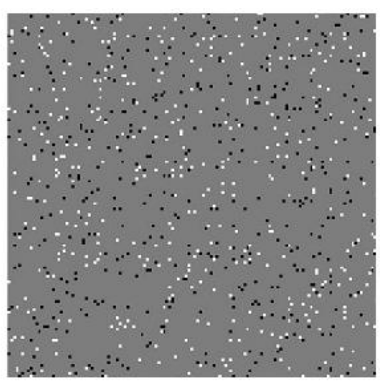


Figure 2.2: Salt & Pepper Noise

- **Random Valued Impulsive Noise (RVIN) :**

For images corrupted by Random-valued noise, the noisy pixels can take any random value in the dynamic range i.e. $R(\ i\ ,j\ )$ can vary between $\{\ Nmin\ ,\ Nmax\ \}$.

In this study our focus is to remove Salt & Pepper noise (Fixed valued impulse noise).

2.1 LITERATURE SURVEY

Noise removal from a contaminated image is still a prominent field of research. Large number of algorithms has been suggested by many researchers. Impulse Noise removal techniques are basically classified into two types:

- **Linear techniques**

In linear techniques noise reduction formula is applied for all the pixels of image linearly without classifying pixels into noisy and non-noisy pixels. Drawback of linear algorithms is blurring of the edges of image as they are not able to effectively eliminate the impulse noise. Examples for linear filters are average, mean, median filters etc.

- **Non linear techniques**

Non linear noise reduction is done in two steps

1) Noise detection

2) Noise restoration

In first step, location of noise is detected and in second step, detected noisy pixels are replaced by estimated value.

In the literature several algorithms are proposed but with low noise condition (10% to 40%). Such algorithms perform well but in high noise environment performance of these algorithms is degraded. To improve the range of noise reduction non linear techniques, MMF (Min-Max Median Filter) [13], CWMF (Center Weighted Media Filter)[14], AMF (Adaptive Median Filter) [15], PSMF (Progressive Switching Median Filter) [16], TMF(Tri-state Median Filter)[5] and DBA (Decision Based Algorithm) [17] algorithms are proposed. The drawback of these algorithms is that as soon as the noise ratio increases execution time required to process noise also increases. Large execution time is not suitable for real world applications.

Working of some linear and non linear filters is discussed briefly.

- **LINEAR FILTERS**

- **Average Filter:**

In an average filter, a square window of size *2s +1* is selected. Value of s varies from 1 to n. Window size (*2s+1*) must be selected as odd number so that the central pixel (*s+1*, *s+1*) is computed accurately. Using window, original image is scanned row wise as well as column wise. At each time of scan, value of central pixel of window is replaced by the average value of its neighbouring pixels computed within the window.

- **Mean Filter:**

Working of Mean Filter is same as Average filter but here central pixel value is replaced by the mean value of its neighboring pixels computed within the window.

- **Median Filter:**

Working of Median Filter is same as Average filter but here central pixel value is replaced by the median value of its neighboring pixels computed within the window.

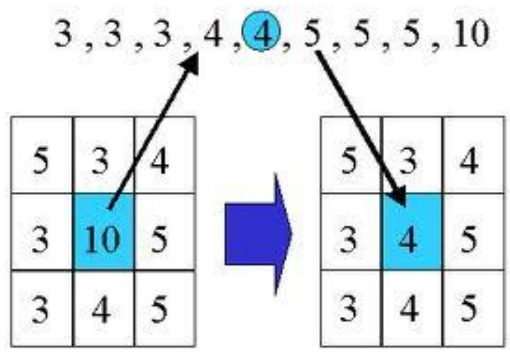


Figure 2.4: Working of Median filter

- **NON LINEAR FILTERS**

- **Min-Max Median Filter:**

Min-Max filter (MMF) [13] is conditional non linear filter. In this filter a window of size 3×3 is selected to scan the image left to right and top to bottom. The center pixel of window at the location (2, 2) is considered as a test pixel. Minimum value (*min*) and maximum value (*max*) of the pixel within the window is computed. If the test pixel is less than *min* and greater than *max*, then center pixel is treated as noisy pixel and its value is replaced by median value of the pixels present in a window. Value of Noise free pixels is left unchanged.

- **Adaptive Median Filter:**

The adaptive median filter (AMF) [15] uses varying window size for the removal of noise. Size of window increases until correct value of median is calculated and noise pixel is replaced with its calculated median value. In this filter two conditions are used one to detect corrupted pixels and second one is to check correctness of median value. If test pixel is less than minimum value and greater than maximum value of the pixel present within the window then center pixel is treated as noisy pixel. If calculated median value is less than minimum value and greater than maximum value present in the window then median value is treated as corrupted value. If calculated median is corrupted then increase the window size and recalculate the median value until we get correct median value or else window size reach maximum limit.

- **Progressive Switching Median Filter:**

The Progressive median filter (PMF) [16] is a two phase algorithm. In the first phase noisy pixels are identified using fixed size window of size 3×3. If test pixel is less than minimum value and greater than maximum value of the pixel present within the window present then center pixel is treated as corrupted pixel. In second phase prior knowledge of noisy pixels is used and noisy pixels are replaced by the estimated median value. Here median value is calculated in the same way as in AMF without considering the corrupted pixel present in window.

- **Tri-state Median Filter:**

The Tri-State Median filter (TSMF) [5] is a two phase algorithm. In phase one noisy pixel are identified using standard median filter. In second phase prior knowledge of noisy pixels is used and noise pixels are replaced by Center weighted median filter.

- **Decision Based Algorithm:**

The Decision-Based algorithm (DBA) [17] is a two phase algorithm. In phase one noise pixels are identified using fixed size window of 3×3. In second phase prior knowledge of noisy pixels is used and noise pixels are replaced by middle value of sorted window pixels. In this time complexity of algorithm is analyzed.

Many more modified forms of median filters have been proposed like weighted median (WM) filter [18], switching median [SWM] filter [19], directional weighted median (DWM) filter [20], modified switching median (MSWM) filter [21] etc.

Apart from classical techniques, recent progress in the fuzzy logic results in the development of new noise reduction methods. Fuzzy filters are easy to realize by means of simple fuzzy rules that characterize a particular noise. Already several fuzzy filters for noise reduction have been developed, such as the well-known Fuzzy Inference Rules by Else action (FIRE) filters by Russo [22]-[26].Due to good performance FIRE filters have been used by several authors to improve the efficiency of their work such as Ville [27]. Jiu [28] also proposed a multilevel filter in fuzzy domain.

Various fuzzy filters results from the modification of the classical median filter such as a new impulse detector for switching median filters [29]. To improve the efficiency, Arojawa [30-31] employed fuzzy rules to the classical median filter to develop a fuzzy median filter.

One common problem seen while processing of images is blurring of edges. It is due to linear filtering. To deal with blur Overton *et al.* [32] and Perona *et al.* [33] developed efficient filters. To preserve edges, an edge preserving fuzzy filter for color images developed by Verma *et al.* [34] provides efficient results. It is a novel technique to detect and remove impulse noise in color images. More sophisticated algorithms have been developed using fuzzy reasoning as well as non fuzzy mechanisms to provide better detection of noise resulting in accurate restoration.

Our goal here is not to give an exhaustive inventory of impulse noise removal algorithms but comparison of some popular ones with our proposed algorithm.

In our study the following filters are used for comparative analysis with our proposed scheme. Brief description of these filters is presented as follows:

- **Noise Adaptive Fuzzy Switching Median (NAFSM) Filter:**

NAFSM [8] developed by Kenny et al is a recursive double stage filter. It employs fuzzy reasoning for the removal of noise present in the image. Initially, for the detection of impulse noise the histogram of the noisy image is utilized. Only noisy pixels take part in the next filtering stage and the noise free pixels are left unchanged to avoid any alteration of fine details. However a major drawback of this filter is large computation time as well as inaccurate median term for restoration.

- **Fuzzy Switching Median (FSM) Filter:**

FSM [7] is also proposed by Kenny et al. It utilizes the separate noise detection and noise cancelation module. NAFSM is essentially adaptive FSM. Its working is same as FSM. The only difference is its adaptive nature which yields better results.

- **Modified Decision Based Unsymmetrical Trimmed Median Filter Algorithm (MDBUTMF) :**

Veerakumar et al. recently proposed MDBUTMF [11] to handle high noise density. In this algorithm noisy pixel is replaced by trimmed median value if pixel values are other than, 0’s and 255’s in the selected window. If all the pixel values are 0’s and 255’s then mean value of all elements in the current window is used. Firstly detection of the corrupted pixels in the image is performed. If the intensity value of the pixel being processed lies between maximum i.e. 255 and minimum i.e. 0 gray level values then it is considered as noise free pixel and it is not changed. If the pixel under consideration takes the maximum or minimum gray level then it is considered as noisy pixel which will be further processed. The quality of restored image using MDBUTMF is better than the several existing algorithms present in literature.

- **Boundary Discriminative Noise Detection (BDND):**

BDND [35] is one of state-of-art technique introduced in literature. It handles high noise density using a two stage scheme for restoration of impulse noise. It provides good visual and quantitative results and easy to implement for real-time image applications. However order of magnitude of the second window size lacks statistical significance and the effectiveness of the validation is weakened which leads to misclassification of pixels.

- **A Highly Effective Impulse Noise Detection Algorithm for Switching Median Filters (HEIND):**

Zhang and Duan developed HEIND [9] to remove high intensity impulse noise. Its strategy is same as BDND to compute the boundaries and consists of two iterations. First phase of detection is same as BDND but the in second stage the window size is varied followed by convolution. This results in less number of misclassification of pixels and maintains the fine details present in an image.

- **An Efficient Edge-Preserving Algorithm for Removal of Salt-and-Pepper Noise (EEPA):**

EEPA [6] is a two-stage scheme which is used for the removal of random as well as fixed valued impulse noise. This algorithm is based on the computation of directional differences. It can detect impulse noise efficiently while preserving the edges very well.

- **Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Detail-Preserving Regularization (DPF):**

Two phase scheme is also incorporated by Raymond et al in DPF [36]. An adaptive median filter is used in the noise detection stage. The window size used for detection is variable and is selected according to the noise level. For the noise removal, a regularization method is used. It restores only corrupted pixels. This algorithm provides better edge preservation and noise suppression. However when the image contains numerous edges like Baboon, Clown etc. this technique fails.

- **DRAW BACKS OF EXISTING SYSTEMS**

Existing systems use fixed or different window size for the restoration of images corrupted by impulse noise. This leads to misclassification of pixels and alters the fine details present in an image. Several algorithms already present in the literature do not provide consistent output in both low and high noise conditions. Only few algorithms efficiently handle high noise condition. They are not well suited for real time applications because of their large execution time.

Therefore we developed an efficient algorithm to handle high noise density of impulse noise. It is intuitive in nature and well suited for real time applications due to its simple structure.

2.2 PERFORMANCE MEASURES

To assess the performance of a filter in removing impulse noise from images many metrics are available. For performance analysis of different filters and comparison of our proposed work with existing techniques two metrics are used i.e.

- **MSE (Mean Square Error)**

Given an original image *f* and reconstructed image *I* of size M×N pixels.

MSE is calculated using:

$$MSE(f,I)=\frac{\sum_{z=1}^{3}\sum_{x=1}^{L}\sum_{y=1}^{M}\left[I\left(x,y,z\right)-f\left(x,y,z\right)\right]^{2}}{3\times L\times M} \quad (3)$$

Lower the value of MSE better the quality of reconstructed image. Ideally it should be zero but practically it's not feasible.

- **PSNR (Peak Signal to Noise Ratio)**

Using PSNR we can estimate the quality of the reconstructed image after noise removal. It is defined as the ratio of peak signal power to noise power. The basic idea is to compute a single number that reflects the quality of the reconstructed image.

PSNR is related to MSE by the following equation:

$$PSNR(f, I) = 10\log\left(\frac{1}{MSE(f, I)}\right) \tag{4}$$

Higher the value of PSNR better is the similarity between two images.

Chapter 3

PROPOSED ALGORITHM

3.1 FCM Algorithm

Fuzzy C-means (FCM) is one of the most popular fuzzy clustering algorithms. It is more tolerant to variations and noise in the input data [37]. It is widely used in different field's different engineering and scientific fields such as pattern recognition and image processing [38]. This method was developed by Dunn in 1973 and improved by Bezdek in 1981 and it is frequently used for image processing. Clustering is basically an unsupervised classification of data by which large sets of data are grouped into clusters of smaller sets of similar data.

Basic steps of the FCM algorithm are explained briefly as follows:

Main purpose of FCM is to minimize the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \left\| x_i - c_j \right\|^2 , 1 \leq m \leq \infty \tag{5}$$

Where,

m is the fuzziness index which is any real number greater than 1(usually 2)

u_{ij} is the degree of membership of x_i in the cluster j

x_i is the ith data of d-dimensional measured data

N is the number of data

C is the number of clusters

c_j is the d-dimension center of the cluster, and

$\left\|x_i - c_j\right\|^2$ is the Euclidean distance between i^{th} data and j^{th} cluster center.

This algorithm assigns membership value to each data point x_i corresponding to each cluster center c_j on the basis of distance between the cluster center and the data point. This distance is the Euclidean distance between i^{th} data and j^{th} cluster center.

More the data is near to the cluster center more is its membership towards the particular cluster center.

After each iteration membership u_{ij} and cluster centers c_j are updated by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\left\|x_i - c_j\right\|}{\left\|x_i - c_k\right\|}\right)^{\frac{2}{m-1}}} \quad (6)$$

Where,

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \quad (7)$$

This iteration will stop when $\max_{ij}\left\{\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right|\right\} < \varepsilon$ (8)

Where ε is a termination criterion between 0 and 1 and *k* is the iteration number.

The algorithm can be summarized as follows:

a) Initialize $U = \left[u_{ij}\right]$ matrix, $U^{(0)}$

b) At *k*-step: calculate the center vectors $C^{(k)} = \left[c_j\right]$ with $U^{(k)}$

c) Update $U^{(k)}$, $U^{(k+1)}$

d) If $\left\| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right\| < \varepsilon$ then STOP; otherwise return to step (b).

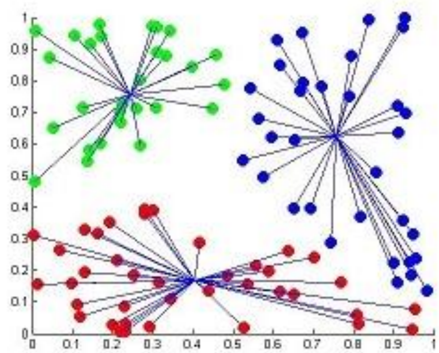


Figure 3.1: Result of Fuzzy C-Means Clustering

FCM (Fuzzy c-means) gives best result for overlapped data set and comparatively better than other existing algorithms such as k-means algorithm [39].

FCM requires the number of clusters to be specified in advance. In our approach we need to separate the low intensity pixels and high intensity pixels from the middle range intensity pixels therefore total number of clusters used is three. We have incorporated the FCM algorithm for clustering in our detection phase. It will be explained in detail in the following section of Impulse noise removal using FCM.

FCM has robust framework and basis of several clustering techniques. One more advantage of using FCM is that unlike k-means where data point must exclusively belong to one cluster center here data point is assigned membership to each cluster center as a result of which data point may belong to more than one cluster center [40-42] .This results in more efficient results even at high noise density.

3.2 IMPULSE NOISE REMOVAL USING FCM

The proposed algorithm named as "*Impulse Noise removal using FCM*" is a novel technique for detecting high density impulse noise from corrupted images. The algorithm is iterative in nature and preserves the fine details of an image in an efficient manner even in high noise environment. The application of Fuzzy C-means (*FCM*) algorithm in the detection phase provides optimal results.

From the literature survey it is clear that efficient removal of impulse noise mainly depends on the detection phase. The detection method of the proposed algorithm efficiently identifies the location of noisy pixels so that the fine details of the image are not altered. Proposed algorithm has two detection stages. Double stage detection efficiently locates the noisy pixels and does not alter the value of noise free pixels.

Algorithm basic steps are explained as follows:

Impulse noise detection stage I:

1. Select a window of size 21×21 which is centred on each pixel of an image.
2. Let the central pixel at which window is centred be $p(i, j)$ Using *FCM* algorithm, divide the neighbourhood values of the central pixel into three clusters.
3. Let the three clusters formed be A, B and C. Find the maximum value present in each cluster respectively.
4. The three maximum values from each cluster are sorted in ascending order.

In a 21×21 window there will be 441 values. FCM algorithm divides these values into three clusters. Let the three maximum values present in the three clusters be M1, M2 and M3.

These values are sorted such that M1 < M2 < M3.

5. Using the following equation, check whether the central pixel $p(i, j)$ is noisy or noise-free.

$$p(i, j) = \begin{cases} \text{pepper noise} : \text{ if } p(i, j) \le M1 \\ \text{noise free} \quad : \text{ if } M1 < p(i, j) \le M2 \\ \text{salt noise} \quad : \text{ if } M2 < p(i, j) \le M3 \end{cases} \tag{9}$$

Pixels in the cluster having the minimum maximum value i.e. M1 are lowest intensity pixels which contains Pepper noise. Cluster having the maximum value of pixel i.e. M3 are highest intensity pixels which contains Salt noise. The third maximum value M2 lies in the middle range intensity values. The cluster having the maximum value as M2 is considered as Noise free cluster.

6. If the pixel is noise free, it is left unaltered.
7. If the pixel is noisy it is processed again in the second detection stage.

Impulse noise detection stage II:

8. Now change the window size to 7 × 7.
9. Repeat Steps 2) to 6) in the same way.
10. If the pixel is detected as noisy pixel in the second detection stage also, it is marked as noisy pixel else noise free.

Restoration of noisy pixels is performed using the well known conventional median filter [1]. Median filter still remains an efficient filter for the restoration of corrupted pixels. In our study, median value for neighboring pixels within the window of size 7 × 7 is computed and is used to replace the noisy pixel value.

3.3 ILLUSTRATION OF WORK

For better understanding of our algorithm, an illustration of our work is presented as follows:

Instead of using a window size of 21×21 in the first detection stage, we are explaining the working of our algorithm using a window size of 7×7.

49	*87*	*155*	*255*	*54*	*64*	*81*
24	*255*	*132*	*163*	*0*	*255*	*0*
46	*39*	*145*	*0*	*156*	*119*	*0*
87	*46*	*141*	**0**	*155*	*255*	*117*
113	*104*	*77*	*125*	*0*	*136*	*116*
0	*0*	*99*	*72*	*119*	*255*	*255*
0	*0*	*255*	*0*	*97*	*54*	*29*

Central pixel

Figure 3.2: 7×7 window

- All the 49 values of the window of size 7×7 are sorted in ascending order and using *FCM* algorithm they are divided into three clusters.
  - *Cluster* A = {72, 77, 81, 87, 87, 97, 99, 104, 113, 116, 117, 119, 119, 125, 132, 136, 141, 145, 155, 155, 156, 163}
  - *Cluster* B = {255, 255, 255, 255, 255, 255, 255}
  - *Cluster* C = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 24, 29, 39, 46, 46, 49, 54, 54, 64}
- Maximum value for each cluster is computed and sorted in ascending order such that

  $M1 < M2 < M3$

  M1= 64, M2=163, M3= 255

- From figure 3.2, Pixel under consideration is $p(i, j) = 0$

- Using equation (9),

$$p(i, j) = \begin{cases} \text{pepper noise : if } p(i, j) \le 64 \\ \text{noise free : if } 64 < p(i, j) \le 163 \\ \text{salt noise : if } 163 < p(i, j) \le 255 \end{cases} \tag{10}$$

- From equation (10) it is clear that 0 belongs to the cluster having *Pepper noise.* Therefore it will be processed again in the second detection stage using a window of size 7×7 in the same manner as illustrated above.

- Noisy pixel will be restored using median filter.

  Median value from figure 3.2 is computed as follows:

{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 24, 29, 39, 46, 46, 49, 54, 54, 64, 72, 77, 81, 87, **87**, 97, 99 104, 113, 116, 117, 119, 119, 125, 132, 136, 141, 145, 155, 155, 156, 163, 255, 255, 255 255, 255, 255, 255}

Median value of the values arranged in ascending order

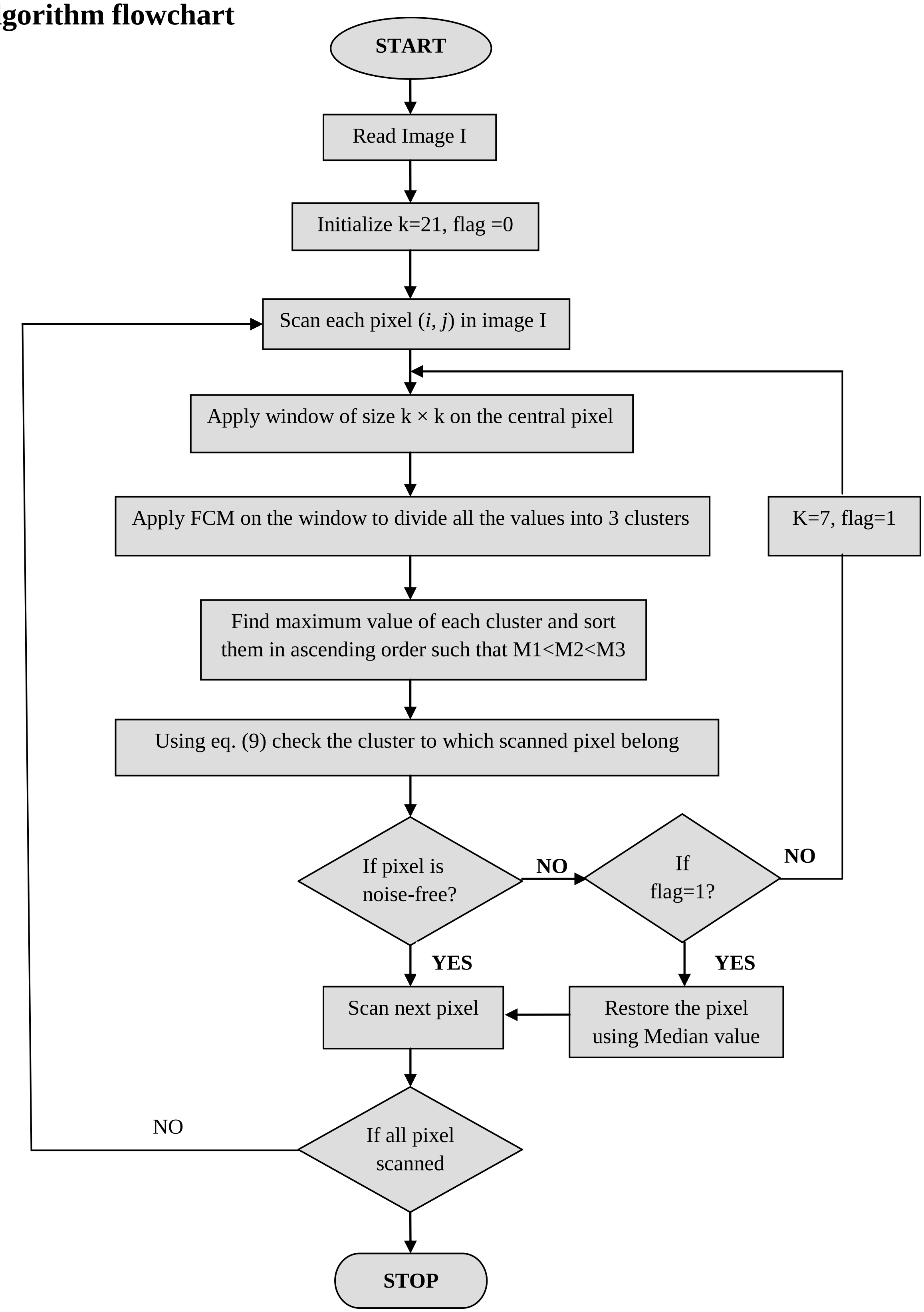
- Therefore the pixel value of *p(i , j)* = 0 is replaced by *p(i , j)* = 87.

49	*87*	*155*	*255*	*54*	*64*	*81*
24	*255*	*132*	*163*	*0*	*255*	*0*
46	*39*	*145*	*0*	*156*	*119*	*0*
87	*46*	*141*	**87**	*155*	*255*	*117*
113	*104*	*77*	*125*	*0*	*136*	*116*
0	*0*	*99*	*72*	*119*	*255*	*255*
0	*0*	*255*	*0*	*97*	*54*	*29*

Restored pixel

Figure 3.3: Restored value of noisy pixel

3.4 Algorithm flowchart



Chapter 4

EXPERIMENTAL RESULTS

4.1 Results using Proposed Approach

The proposed algorithm and all the techniques used for comparison with our approach have been implemented on Intel Core i3 at 2.40 GHz using MATLAB version 2009b.

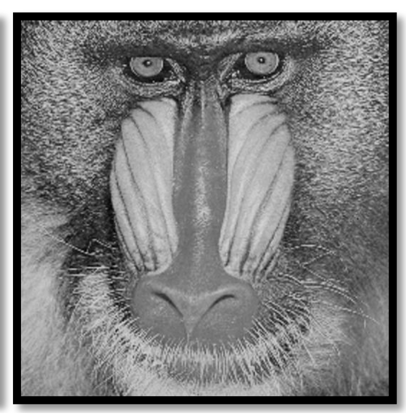
The proposed scheme is simulated on some standard images like *Lena*, *Mandrill*, *Living room*, *Woman blonde* and *Pirate*.

(a) Lena

(b)Living room

(c) Woman Blonde

(d) Pirate



(e) Mandrill

Figure 4.1: Original Standard Images used for simulation results

(a) (b)

Figure 4.2: Result for *Lena* image at 30% noise level (a) noisy image (b) restored image

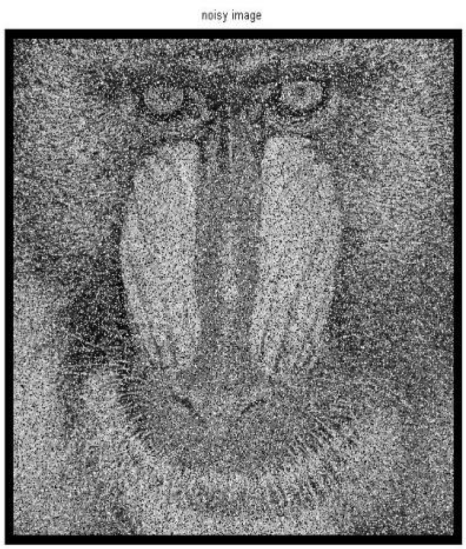
(a)

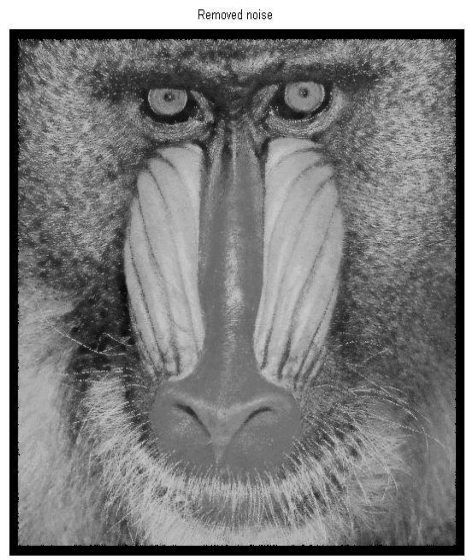
(b)

Figure 4.3: Result for *Lena* image at 40% noise level (a) noisy image (b) restored image

(a) **(b)**

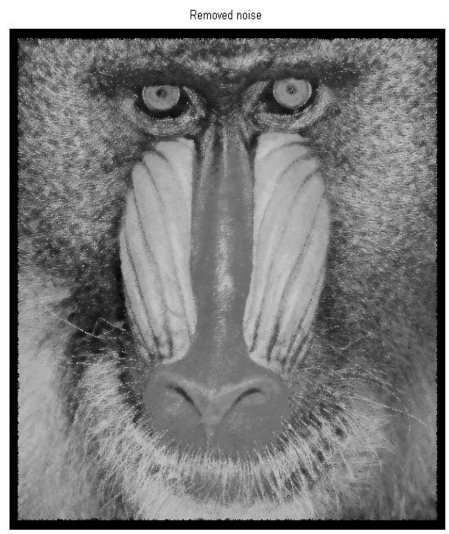
Figure 4.4: Result for *Lena* image at 50% noise level (a) noisy image (b) restored image





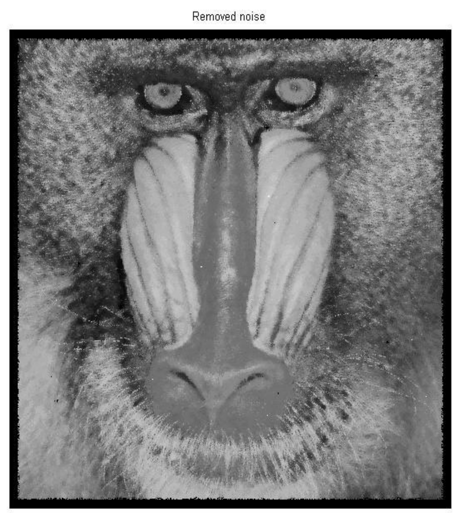
(a) (b)

Figure 4.5: Result for *Mandrill* image at 30% noise level (a) noisy image (b) restored image



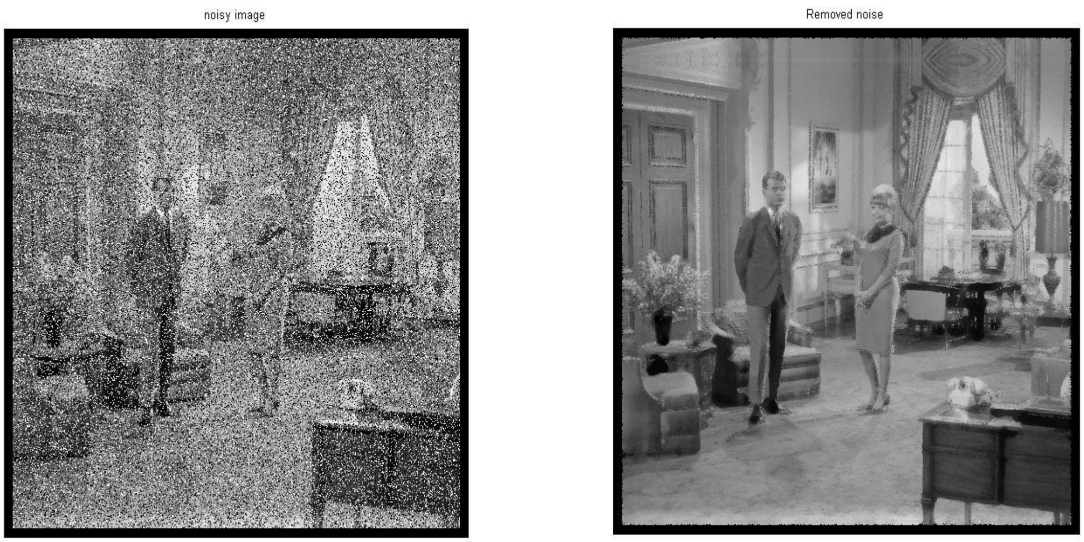
(a) (b)

Figure 4.6: Result for *Mandrill* image at 40% noise level (a) noisy image (b) restored image



(a) (b)

Figure 4.7: Result for *Mandrill* image at 50% noise level (a) noisy image (b) restored image



(a) (b)

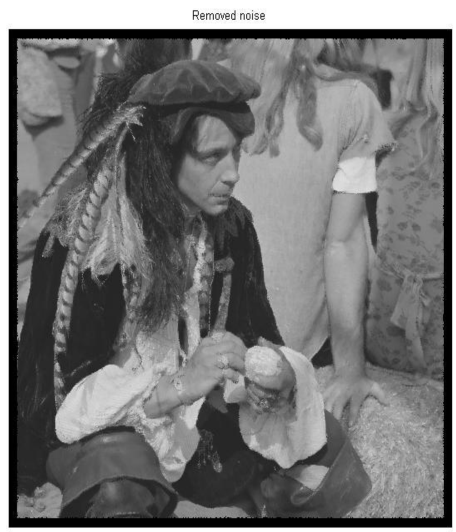
Figure 4.8: Result for *Living room* image at 30% noise level (a) noisy image (b) restored image

(a) (b)

Figure 4.9: Result for *Living room* image at 40% noise level (a) noisy image (b) restored image

(a)   (b)

Figure 4.10: Result for *Living room* image at 50% noise level (a) noisy image (b) restored image



(a) (b)

Figure 4.11: Result for *Pirate* image at 30% noise level (a) noisy image (b) restored image

(a) (b)

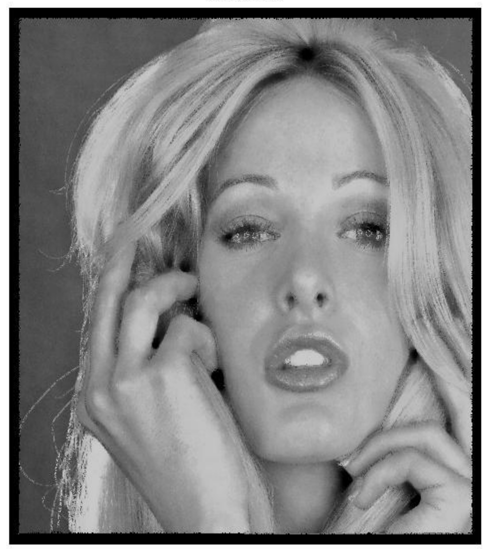
Figure 4.12: Result for *Pirate* image at 40% noise level (a) noisy image (b) restored image

(a)

(b)

Figure 4.13: Result for *Pirate* image at 50% noise level (a) noisy image (b) restored image

(a)



(b)

Figure 4.14: Result for *Woman Blonde* image at 30% noise level (a) noisy image (b) restored image

(a)

(b)

Figure 4.15: Result for *Woman Blonde* image at 40% noise level (a) noisy image (b) restored image

(a) (b)

Figure 4.16: Result for *Woman Blonde* image at 50% noise level (a) noisy image (b) restored image

Apart from the available standard images, we have simulated one test image from real world to test the feasibility of our work. This image was taken using the 2 megapixels camera of mobile phone from a moving airplane.

Figure 4.17: Original *Test Image* from real world

It is simulated for various noise levels. The results shown are from 30% to 50% noise level.

It is observed that the execution time is less for real world applications as well as our algorithm is capable to handle high noise densities in an effective manner. From table 4.1 and 4.2 *PSNR* and *MSE* values show that algorithm provides good restoration if corrupted images with impulse noise.

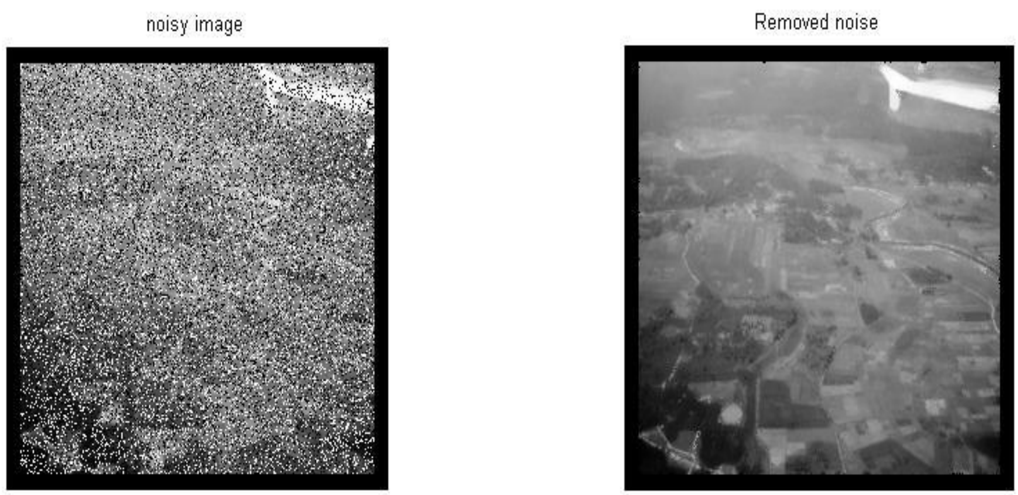


Figure 4.18: Result of *Test Image* at 30% noise level using proposed algorithm

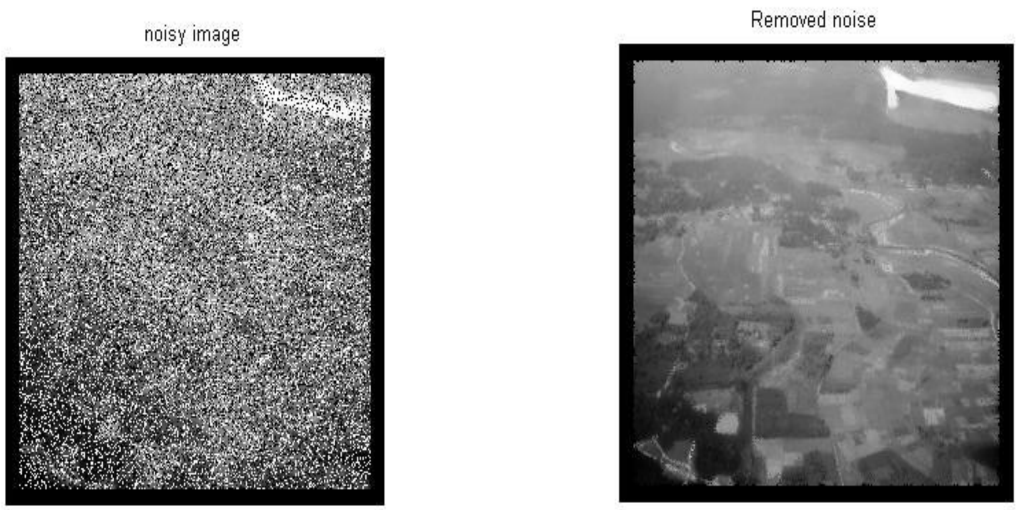


Figure 4.19: Result of *Test Image* at 40% noise level using proposed algorithm

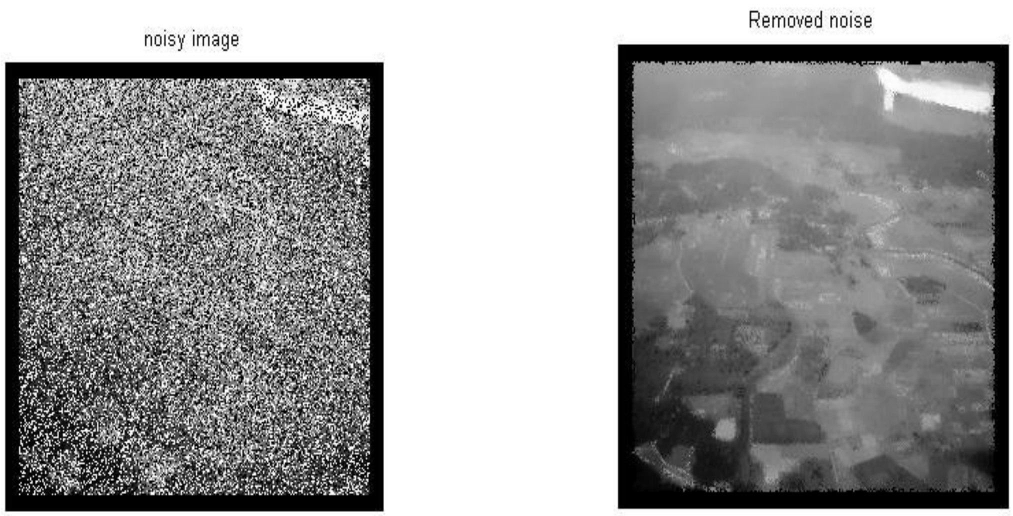


Figure 4.20: Result of *Test Image* at 50% noise level using proposed algorithm

TABLE 4.1

COMPARISON OF PSNR (db) VALUES FOR DIFFERENT IMAGES USING OUR PROPOSED ALGORITHM

	Noise percentage%		
Image used	30	40	50
Lena	39.14	37.90	36.05
Mandrill	34.68	33.52	32.48
Living room	36.59	35.46	33.96
Pirate	37.59	36.37	34.85
Woman blonde	37.91	36.93	35.89
Test image	39.87	38.44	37.55

TABLE 4.2

COMPARISON OF MSE VALUES FOR DIFFERENT IMAGES USING OUR PROPOSED ALGORITHM

	Noise percentage%		
Image used	30	40	50
Lena	7.91	10.53	16.56
Mandrill	12.11	18.89	20.66
Living room	14.24	18.49	26.09
Pirate	11.32	15.05	21.29
Woman blonde	10.34	13.20	16.89
Test image	11.41	17.69	20.60

4.1 Comparison with Other Techniques

Proposed algorithm has been compared with some existing techniques present in literature. For the comparison standard test image *Lena* corrupted with 30% noise density is used. The techniques used for the comparison have been implemented on Intel Core i3 at 2.40 GHz using MATLAB version 2009b.

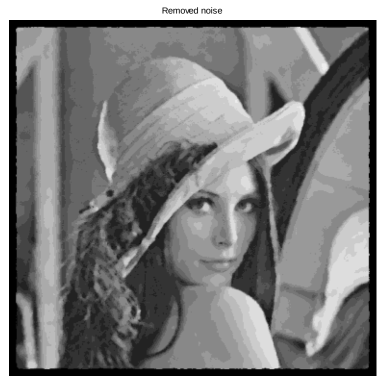
Techniques used for comparison in our work are as follows:

- Noise Adaptive Fuzzy Switching Median (NAFSM) Filter [8]
- Fuzzy Switching Median (FSM) Filter [7]
- Boundary discriminative noise detection (BDND) [35]
- A highly effective impulse noise detection algorithm for switching median filters (HEIND) [9]
- Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization (DPF) [36]
- Modified decision based unsymmetrical trimmed median filter algorithm (MDBU) [11]
- An efficient edge-preserving algorithm for removal of salt-and-pepper noise (EEPA) [6]

Performance analysis of each technique is done in terms of *PSNR* and *MSE* values as shown in Table 4.3 and Table 4.4 respectively.

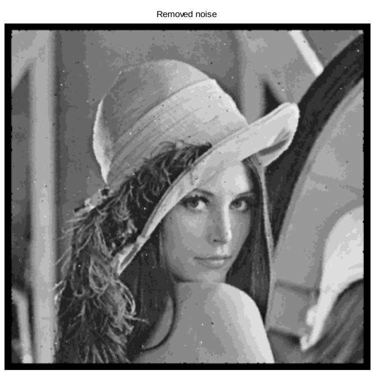
The comparison of our proposed algorithm with the above mentioned techniques show that our algorithm outperforms several techniques even at high noise levels. Both quantative and qualitative results are shown. Visual results are pleasing even at high noise density. The drawback with other methods introduced to handle high noise density fails as we increase the noise level. The processing time increases drastically which make them unsuitable for real world applications. Our algorithm provides consistent results at every noise level preserves more edge details and fine details present in the image. The main advantage of our algorithm is that its performance is not degraded with increasing noise level.

(a)



(e)

(b)



(f)

(c)

(g)

(d)

(h)

Figure 4.21: Comparison of Proposed Algorithm with other techniques on image *Lena* at 30% noise level (a) MDBU (b) BDND (c) FSM (d) NAFSM (e) HEIND (f) EEPA (g) DPF (h) OURS

TABLE 4.3
COMPARISON OF PSNR VALUES FOR "LENA" IMAGE

	Noise percentage%			
Method	10	20	30	40
DPF	33.95	34.06	34.24	34.08
FSM	36.08	34.11	32.41	31.30
NAFSM	37.09	34.52	32.49	31.41
EEPA	39.05	37.56	35.42	32.87
HEIND	39.65	38.57	37.41	35.87
BDND	40.26	38.95	37.39	35.03
MDBU	40.76	39.06	36.01	33.21
OURS	42.26	40.97	39.14	37.90

TABLE 4.4
COMPARISON OF MSE VALUES FOR "LENA" IMAGE

	Noise percentage%			
Method	10	20	30	40
DPF	16.26	18.53	23.40	30.96
FSM	12.30	12.56	14.25	15.02
NAFSM	12.60	25.25	37.13	47.83
EEPA	25.20	26.60	27.80	29.41
HEIND	38.60	37.90	37.17	36.15
BDND	30.60	32.55	37.42	45.67
MDBU	11.90	14.94	19.41	22.56
OURS	5.68	6.93	7.91	10.53

Chapter 5

CONCLUSIONS

Images are affected by impulse noise generally during image acquisition and image transmission. Analog to digital converters, errors due generated by noisy sensors and faulty equipments result in corrupted images by impulse noise. Many techniques have been introduced in the literature to remove noise. At low noise intensity of noise many algorithms perform well but as soon as the noise levels are increased, performance of the method degrades. Therefore, we developed a filter which provides consistent outputs.

Experimental results show that the proposed algorithm significantly outperforms existing well-known techniques. The main advantage of our algorithm is that its performance is not degraded with increasing noise level. It can easily handle high noise levels up to 80%.

It is easy to understand as it has uncomplicated structure and intuitive in nature .It provides good results on different images even in real world applications. The application of Fuzzy C-means (*FCM*) algorithm in the detection phase provides optimal results and makes it a novel technique.

In future, the current thesis work will be extended for RGB images as well as video images.

Our main focus has been the correct detection of noisy pixels so that the restoration provides optimal results. We will extend our work towards optimal restoration of images.

REFERENCES

[1] R.C.Gonzalez and R.E. Wood, Digital Image Processing, Prentice-Hall, India, Second Edition, 2007.

[2] Goutsias, J, Vincent, L., and Bloomberg, D. S. (eds.), Mathematical Morphology and Its Applications to Image and Signal Processing, Kluwer Academic Publishers, Boston, MA. 2000.

[3] Ritter, G.X. and Wilson, J.N., Handbook of Computer Vision Algorithms in Image Algebra, CRC Press, Boca Raton, FL., 2001.

[4] Scott E Umbaugh, Computer Vision and Image Processing, Prentice Hall PTR, New Jersey, 1998.

[5] T. Chen, K. K. Ma, and L. H. Chen, “Tri-state median filter for image denoising,” IEEE Trans. Image Process., vol. 8, pp. 1834–1838, Dec.1999

[6] P. Y. Chen and C. Y. Lien, “An efficient edge-preserving algorithm for removal of salt-and-pepper noise,” IEEE Signal Process. Lett, vol. 15, pp. 833–836, 2008.

[7] K. K. V. Toh, H. Ibrahim, and M. N. Mahyuddin, “Salt-and-pepper noise detection and reduction using fuzzy switching median filter,” IEEE Trans. Consumer Electron., vol. 54, no. 4, pp. 1956–1961, Nov. 2008.

[8] Kenny Kal Vin Toh,” Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction” IEEE signal processing letters, VOL. 17, NO. 3 pp 281-244, Mar. 2010

[9] Fei Duan and Yu-Jin Zhang," A Highly Effective Impulse Noise Detection Algorithm for Switching Median Filters" IEEE Signal Process. Lett. vol. 17, pp. 647–650, July 2010.

[10] Umesh Ghanekar, Awadhesh Kumar Singh, and Rajoo Pandey," A Contrast Enhancement-Based Filter for Removal of Random Valued Impulse Noise" IEEE Signal Process. Lett, vol. 17, no.1. pp. 47–50, Jan. 2010.

[11] S. Esakkirajan, T. Veerakumar, Adabala N. Subramanian, and C. H. PremChand" Removal of High Density Salt and Pepper Noise Through Modified Decision Based Unsymmetric Trimmed Median Filter" IEEE signal processing letters, VOL. 18, NO. 5 pp 287-290, May 2011.

[12] Smaïl Akkoul, Roger Lédée, Remy Leconge, and Rachid Harba, "A New Adaptive Switching Median Filter", IEEE Signal Processing letters, pp. 587-590. 2010.

[13] 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.

[14] S. J. KO and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," IEEE Trans. Circuits Syst., vol. 38, pp. 984–993, 1991.

[15] H.Hwang and R.A.Haddad, "Adaptive Median Filters: New Algorithms and Results," IEEE Trans. Image Processing, vol.4, no.4, pp.499-502, 1995.

[16] Z. Wang and D.Zhang, "Progressive Switching Median filter for the removal of Impulse Noise from Highly corrupted images," IEEE trans. on circuits and systems part II: Analog and Digital signal processing, vol.46, no.1, pp.78-80, 1999.

[17] K. S. Srinivasan and D. Ebenezer, “A new fast and efficient decision based algorithm for removal of high density impulse noise,” IEEE Signal Process. Lett. vol. 14, no. 3, pp. 189–192, Mar. 2007.

[18] D.R.K.Brownrigg “The Weighted median filter,” *Commun*.ACM, vol.27, no.8, pp.807-818, Aug 1984.

[19] T. Sun and Y. Nuevo, “Detail preserving median based filters in image processing,” Pattern Recognition. Lett, vol. 15, pp. 341–347, 1994.

[20] Y. Dong and S. Xu, “A new directional weighted median filter for removal of random-value impulse noise,” IEEE Signal Process. Lett, vol.14, pp. 193–196, Mar. 2007.

[21] C. C. Kang and W. J. Wang, “Modified switching median filter with one more noise detector for impulse noise removal,” Int.J.Electron.Commun.No. DOI: 10.1016/j.aeue.2008.08.009, 2008.

[22] Russo, F, “A FIRE filter for detail-preserving smoothing of images corrupted by mixed noise,” IEEE fuzzy syst., vol.2, pp 1051-1055, July 1997.

[23] Russo F., Noise Cancellation Using Nonlinear Fuzzy Filters, in: IEEE Instrumentation and Measurement Technology Conf., 1997,pp.772-777

[24] Russo F., FIRE operators for image processing, in: IEEE Fuzzy Sets and Systems, Vol. 103, 1999, pp. 265-275.

[25] Russo F. & Ramponi G., A fuzzy filter for images corrupted by impulse noise, in: IEEE Signal proceedings letters, Vol.3, No. 6, 1996, pp. 168- 170.

[26] Russo F. & Ramponi G., Removal of impulse noise using a FIRE filter, in: IEEE Proceedings, 1996, pp. 975-978.

[27] Kwan H.K., Fuzzy Filters for noise reduction in images, in: Fuzzy Filters for Image Processing (Nachtegael M., Van der Weken D., Van De Ville D. & Etienne E.E., editors), Springer-Verlag, 2002, pp. 25-53

[28] Jiu J.Y., Multilevel median filter based on fuzzy decision, DSP IC Design Lab E.E. NTU., 1996.

[29] S. Q. Zhang and M. A. Karim, “A new impulse detector for switching median filters,” IEEE Signal Process. Lett, vol. 9, no. 11, pp. 360–363, Nov. 2002

[30] K. Arojawa, “Median Filter based on Fuzzy Rules and its Application to Image Restoration”, Fuzzy Sets and Systems, Vol. 77, pp. 3-13, 1996.

[31] K. Arojawa, Edited by E.E. Kerre and M. Nachtegael, “Fuzzy Ruled-Based Image Processing with Optimization”, SpringerVerlag, pp.222-247, 2000.

[32] K.J. Overton and T.E. Weymouth, "A noise reducing preprocessing algorithm",Proceedings of the IEEE Computer Science Conference on Pattern Recognition and Image Processing, pages 498-507, Chicago, IL, 1979.

[33] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion", IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(7):629-639, 1990.

[34] Verma, O.P.; Parihar, A.S.; Hanmandlu, M.; “Edge Preserving Fuzzy Filter for Color Images”; 2010 International Conference on Computational Intelligence and Communication Networks (CICN), : 2010 , pp: 122 – 127.

[35] P.-E. Ng and K.-K. Ma, “A switching median filter with boundary discriminative noise detection for extremely corrupted images,” IEEE Trans. Image Process., vol. 15, no. 6, pp. 1506–1516, Jun. 2006.

[36] R. H. Chan, C.-W. Ho, and M. Nikolova, “Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization,” IEEE Trans. Image Process., vol. 14, no. 10, pp. 1479–1485, Oct. 2005

[37] R. N. Dave and R. Krishnapuram, “Robust clustering methods: A unified view,” IEEE Trans. Fuzzy Syst., vol. 5, no. 2,pp. 270–293, 1997.

[38] A. K. Jain, M. N. Murthy, and P. J. Flynn, “Data clustering: A review,” Department of Computer Science, Michigan State University, East Lansing, Michigan, Tech. Rep., August 2000.

[39] R.N.Dave, “Characterization and detection of noise in clustering,” Pattern Recognition Letter, vol. 12, no. 11, pp.657–664, 1991.

[40] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms. New York, NY: Plenum Press, 1981.

[41] Banshidhar Majhi. Soft Computing Techniques for Image Restoration. PhD thesis, Sambalpur University, 2000.

[42] J. G. Proakis and D. G. Manolakis. Digital Signal Processing: Principles, Algorithms and Applications. Prentice Hall of India, New Delhi, 3rd edition, 2002.