SOME NOVEL DIGITAL IMAGE FILTERS FOR SUPPRESSION OF IMPULSIVE NOISE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology
in
Telematics and Signal Processing

By

MANOJ KUMAR GUPTA
Roll No.-20507032

Department of Electronics and Communication Engineering

National Institute of Technology, Rourkela, Orissa
2007

SOME NOVEL DIGITAL IMAGE FILTERS FOR SUPPRESSION OF IMPULSIVE NOISE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

in

Telematics and Signal Processing

By

MANOJ KUMAR GUPTA

Roll No.-20507032

Under the Guidance of

Prof. S. MEHER

Department of Electronics and Communication Engineering

National Institute of Technology, Rourkela, Orissa
2007

National Institute of Technology
Rourkela

CERTIFICATE

This is to certify that the thesis titled, **“Some Novel Digital Image Filters for Suppression of Impulsive Noise”** submitted by Sri **Manoj Kumar Gupta** in partial fulfillment of the requirements for the award of Master of Technology Degree in Electronics and Communication Engineering with specialization in **“Telematics and Signal Processing”** at the National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Prof. S. Meher
Dept. of Electronics and Communication Engg.
National Institute of Technology
Rourkela-769008

Date:

ACKNOWLEDGEMENTS

I am grateful to numerous local and global peers who have contributed towards shaping this thesis. At the outset, I would like to express my sincere thanks to **Prof. S. Meher** for his advice during my thesis work. As my supervisor, he has constantly encouraged me to remain focused on achieving my goal. His observations and comments helped me to establish the overall direction of the research and to move forward with investigation in depth. He has helped me greatly and been a source of knowledge.

I thank **Prof. G. Panda** (Head of Department) for his constant encouragement and support during my Post Graduate studies. I would like to thank, **Prof. G. S. Rath, Prof. K. K. Mahapatra,** and **Prof. S.K. Patra** for sharing his time and knowledge. They have been great sources of inspiration to me and I thank them from the bottom of my heart.

I would like to thank administrative and technical staff members of the Department who have been kind enough to advise and help in their respective roles. I would also like to thank N. Bhoi, R. K. Kulakarni, C. S. Rawat**,** K. Ayyanna, M. Kamal Kumar**,** and Nirulata for helping me a lot during the thesis period.

Throughout Post Graduate studies I have been fortunate to have wonderful support structure among the Post Graduate students. I am really thankful to my friends. My sincere thanks to everyone who has provided me with kind words, a welcome ear, new ideas, useful criticism, or their invaluable time, I am truly indebted.

Last, but not least, I would like to dedicate this thesis to my family, for their love, patience, and understanding.

Manoj Kumar Gupta
Roll No. - 20507032

Contents

ABSTRACT

In digital imaging, quality of image degrades due to contamination of various types of noise during the process of acquisition, transmission and storage. Especially impulse noise appears during image acquisition and transmission, which severely degrades the image quality and cause a great loss of information details in an image. Various filtering technique are found in literature for removal of impulse noise. Nonlinear filter such as standard median, weight median filter, center weight median and switching based median filter out perform the linear filters.

This thesis investigates the performance analysis of different nonlinear filtering schemes. The performance of these filters can be improved by incorporating the mechanism of noise detection and then applying switching based adaptive filtering approach. Three novel filtering approaches that incorporate the above principles are proposed. It is found that all three approaches give noticeable performance improvement of over many filters reported in literature.

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Preview

Today digital imaging is required in many applications e.g., object recognition, satellite imaginary, biomedical instrumentation, digital entertainment media, internet etc. The quality of image degrades due to contamination of various types of noise. Noise corrupts the image during the process of acquisition, transmission, storage etc. For a meaningful and useful processing such as image segmentation and object recognition, and to have very good visual display in applications like television, photo-phone, etc., the acquired image signal must be noise free and made deblurre. Both the *noise suppression* (filtering) and the *deblurring* come under a common class of image processing tasks known as *image restoration*.

Amongst the various types of noise, the impulse noise may appear during image acquisition and transmission. Two types of impulse noise can be modeled: (i) Fixed valued impulse noise, also called, salt & pepper noise (SPN) and (ii) Random-valued impulse noise (RVIN). The absolute-average intensity of impulse noise could be very high for an RVIN under some circumstances. Thus, it could severely degrade the image quality and cause a great loss of information details in an image. For both SPN and RVIN, impulse noise density plays a great role. If the density is very high (normally > 50%), then it is very difficult to estimate the original pixel value from the neighborhood pixels.

For this dissertation, the following research activities are taken up:

(a) Study of various impulse noise types and their effect on digital images;

(b) Study and implementation of various efficient nonlinear and adaptive digital image filters available in the literature and their relative performance comparison;

(c) Development and implementation of various novel efficient nonlinear and adaptive digital image filters and their relative performance comparison.

1.2 Literature Review

Noise in an image is a serious problem. The noise could be Additive White Gaussian Noise (AWGN), Salt & Pepper Impulse Noise (SPIN), Random Value Impulse Noise (RVIN), or a mixed noise. Efficient suppression of noise in an image is a very important issue. Denoising finds extensive applications in many fields of image processing. Conventional techniques of image denoising using linear and nonlinear techniques have already been reported and sufficient literature is available in this area and has been reviewed in the next paragraph. Recently, various nonlinear and adaptive filters have been suggested for the purpose. The objectives of these schemes are to reduce noise as well as to retain the edges and fine details of the original image in the restored image as much as possible. However, both the objectives conflict each other and the reported schemes are not able to perform satisfactorily in both aspects. Hence, still various research workers are actively engaged in developing better filtering schemes using latest signal processing techniques. In the present thesis, efforts have been made in developing some efficient noise removal schemes.

Most of the classical linear digital image filters, such as averaging low pass filters have low pass characteristics and they tend to blur edges and to destroy lines, edges and other fine image details. One solution to this problem is the use of the median (MED) filter, which is the most popular order statistics filter [1,2] under the nonlinear filter classes. This filter has been recognized as a useful filter due to its edge preserving characteristics and its simplicity in implementation. The median filter, especially with larger window size destroys the fine image details due to its rank ordering process. Applications of the median filter require caution because median filtering tends to remove image details such as thin lines and corners while reducing noise. One way to improve this situation is the weighted median WM filter [3,4,5,6], which is an extension of the median filter that gives more weight to some values within the window. It emphasizes or de-emphasizes specific input samples, because in most applications, not all samples are equally important. The special case of the median filter is the center-weighted median (CWM) filter [7], which gives more weight only to the central value of the window. It is also reasonable to give emphasis to the central sample, because it is one that is the most correlated with the desired estimate. The median filter, as well as its modifications and generalizations [8] are typically implemented invariantly across an image. They tend to alter pixels undisturbed by noise. Additionally, they are prone to edge jitter in cases where the noise ratio is high. As a result, their effectiveness in noise suppression is often at the expense of blurred and distorted image features. Another way to circumvent this situation is to incorporate some decision making process in the filtering framework.

Conventional median filtering approaches apply the median operation to each pixel unconditionally, that is, without considering whether it is uncorrupted or corrupted. As a result, the image details contributed from the uncorrupted pixels are still subject to be filtered, and this causes image quality degradation. An intuitive solution to overcome this problem is to implement an impulse-noise detection mechanism prior to filtering; hence, only those pixels identified as “corrupted” would undergo the filtering process, while those identified as “uncorrupted” would remain intact. By incorporating such noise detection mechanism or “intelligence” into the median filtering framework, the so-called *switching median filters* [11]–[16] had shown significant performance improvement. To address this drawback, a number of modified median filters have been proposed, e.g., *minimum–maximum exclusive mean* (MMEM) filter [17], *prescanned minmax center-weighted* (PMCW) filter [18], and decision-based median filters [19], [20], [21], [22]. In these methods, the filtering operation adapts to the local properties and structures in the image. In the decision-based filtering, for example, image pixels are first classified as *corrupted* and *uncorrupted*, and then passed through the median and identity filters, respectively. The main issue of the decision-based filter lies in building a decision rule, or a noise measure, that can discriminate the uncorrupted pixels from the corrupted ones as precisely as possible. In the method proposed by Han MMEM [17], in these pixels that have values close to the maximum and minimum in a filter window are discarded, and the averages of remaining pixels in the window are computed. If the difference between the center pixel and average exceeds a threshold, the center pixel is replaced by average; otherwise, unchanged. In ACWM [20], CWM [7] has used to detect noisy pixels. The objective is to utilize the *center-weighted median* (CWM) [2] filters that have varied center weights to define a more general operator, which realizes the impulse detection by using the differences defined between the outputs of CWM filters and the current pixel of concern. The ultimate output is switched between the median and the current pixel itself. While still using a simple thresholding operation, the proposed filter yields superior results to other switching schemes in suppressing both types of impulses with different noise ratios. Florencio *et al.* [10] proposed a decision measure, based on a second order statistic called normalized deviation.

The *tri-state median filter* [11] further improved switching median filters that are constructed by including an appropriate number of center-weighted median filters into the basic switching median filter structure. These filters exhibit better performance than the standard and the switching median filters at the expense of increased computational complexity. In *progressive switching median filter* (PSM) [12] for the removal of impulse noise from highly corrupted images has proposed, where both the impulse detector and the noise filter are applied progressively in iterative manners. The noise pixels processed in the current iteration are used to help the process of the other pixels in the subsequent iterations. A main advantage of such a method is that some impulse pixels located in the middle of large noise blotches can also be properly detected and filtered. Therefore, better restoration results are expected, especially for the cases where the images are highly corrupted. A new impulse noise detection technique [13] for switching median filters, which is based on the minimum absolute value of four convolutions, obtained using one-dimensional Laplacian operators. It provides better performance than many of the existing switching median filters with comparable computational complexity.

The *signal-dependent rank-ordered mean filter* [23] is a switching mean filter that exploits rank order information for impulse noise detection and removal. The structure of this filter is similar to that of the switching median filter except that the median filter is replaced with a *rank-ordered mean filter*. This filter has been shown to exhibit better noise suppression and detail preservation performance than some conventional and state-of-the-art impulse noise cancellation filters for both grey scale [23] and color [24] images.

The *peak and valley filter* [25] is a highly efficient recursive nonlinear filter. It identifies noisy pixels by inspecting their neighborhood, and then replaces their values with the most conservative ones out of the values of their neighbors. In this way, no new values are introduced into the neighborhood and the histogram distribution range is conserved. The main advantage of this filter is its simplicity and speed, which make it very attractive for real time applications. A *modified peak and valley filter* [26] has also been proposed. This filter provides very good detail preservation performance but it is slower than the original peak and valley filter.

The *adaptive two-pass rank order filter* [27] has been proposed to remove impulse noise from highly corrupted images. Between the passes of filtering, an adaptive process detects irregularities in the spatial distribution of the estimated noise and selectively replaces some pixels changed by the first pass with their original values. These pixels are kept unchanged during the second filtering. Consequently, the reconstructed image maintains a higher degree of fidelity and has a smaller amount of noise. In [28], a detail-preserving variational method has been proposed to restore impulse noise. It uses a nonsmooth data fitting term together with edge-preserving regularization functions. A combination of this variational method [28] with an impulse detector [20] has also been presented in [29] for the removal of random valued impulse noise. The filter offers good filtering performance but its implementation complexity is higher than most of the previously mentioned filters.

The *two-output nonlinear filter* [30] is another rank order filter based on the subsequent activation of two recursive filtering algorithms that operate on different subsets of input data. Two pixel values are updated at each processing step. A nonlinear mechanism for error correction is also provided for avoiding detail blur. The filter provides very good detail preservation performance.

The *threshold boolean filter* [31] employs boolean functions for impulse noise removal. In this approach, the gray level noisy input image is decomposed into a number of binary images by gray level thresholding. Detection and removal of impulse noise are then performed on these binary images by utilizing specially designed boolean functions. Finally, the resulting boolean images are combined back to obtain a restored grey level image. A number of filters utilize the histogram information of the input image. In [28], histograms of homogenous image regions are used to characterize and classify the corrupting noise. In [32] and [33], the histogram information of the input image is used to determine the parameters of the membership functions of an adaptive fuzzy filter. The filter is then used for the restoration of noisy images. An adaptive vector filter exploiting histogram information is also proposed for the restoration of color images [34].

In addition to the median and the mean based filtering methods discussed above, a number of nonlinear impulse noise filtering operators based on soft computing methodologies have also been presented [35]–[40]. These filters offer relatively better noise removal and detail preservation performance than the median and the mean based operators. However, the implementation complexities of these filters are generally higher and the required filtering window size is usually larger than the other methods. In the last few years, there has been a growing research interest in the applications of soft computing techniques, such as neural networks and fuzzy systems, to the problems in digital image processing [32]–[40]. Indeed, neuro-fuzzy (NF) systems offer the ability of neural networks to learn from examples and the capability of fuzzy systems to model the uncertainty which is inevitably encountered in noisy environments. Therefore, neuro-fuzzy systems may be utilized to design line, edge, and detail preserving impulse noise removal operators provided that the appropriate network topologies and processing strategies are employed.

Early-developed switching median filters are commonly found being non adaptive to a given, but unknown, noise density and prone to yielding pixel misclassifications especially at higher noise density interference. To address this issue, the *noise adaptive soft-switching median* (NASM) filter was proposed in [14], which consists of a three-level hierarchical soft-switching noise detection process. The NASM achieves a fairly robust performance in removing impulse noise, while preserving signal details across a wide range of noise densities, ranging from 10% to 50%. However, for those corrupted images with noise density greater than 50%, the quality of the recovered images become significantly degraded, due to the sharply increased number of misclassified pixels. In BDND [41] highly-accurate noise detection algorithm, called the *boundary discriminative noise detection*, which can handle image corruption even up to 80% noise density. Together with the modified NASM median filtering scheme, this BDND has shown far superior performance in terms of subjective quality in the filtered image as well as objective quality in the *peak signal-to-noise ratio* (PSNR) measurement to that of the NASM filter. For denoising color images, BDND filter consistently shows impressive results as well.

1.3 The Problem Statement

In the last two decades, many researchers have attempted to develop filters to suppress the impulse noise. But that are not adaptive in nature, so the performance of that filter is not good in many occasion. Some filters are not able to preserve image detail and also many filters are quite efficient at high noise levels but don’t perform so well at low noise levels. Therefore, it is very important to design and develop highly efficient adaptive nonlinear image filters that suppress impulse noise quite effectively and preserve image detail.

Therefore, the problem taken for this thesis work is to develop *some novel nonlinear and adaptive digital image filters for efficient impulse noise suppression.*

1.4 Performance Metrics

The quality of an image is examined by objective evaluation as well as subjective evaluation. There are various metrics that can be used for objective evaluation of an image. Some of them are mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE) and peak signal to noise ratio (PSNR).In this thesis work only two metrics has used to evaluate the objective quality of filtered images i.e. PSNR and MSE.

The performance evaluation of the filtering operation is quantified by the PSNR calculated using the following standard formula:

$$\text{PSNR} = 10 \log_{10}\left(\frac{255^2}{\text{MSE}}\right) \text{ dB}$$

And

$$\text{MSE} = \frac{1}{\text{MN}} \sum_{i=1}^{M} \sum_{j=1}^{N} [I(i,j) - \hat{I}(i,j)]^2$$

where M and N are the total number of pixels in the horizontal and the vertical direction in the image respectively . $I(i,j)$ and $\hat{I}(i,j)$ denotes the original and filtered image pixels, respectively.

For subjective evaluation, the image has to be observed by a human expert. The human visual system (HVS) is so complicated that it is not yet modeled properly. Therefore, in addition to objective evaluation, the image must be observed by a human expert to judge its quality.

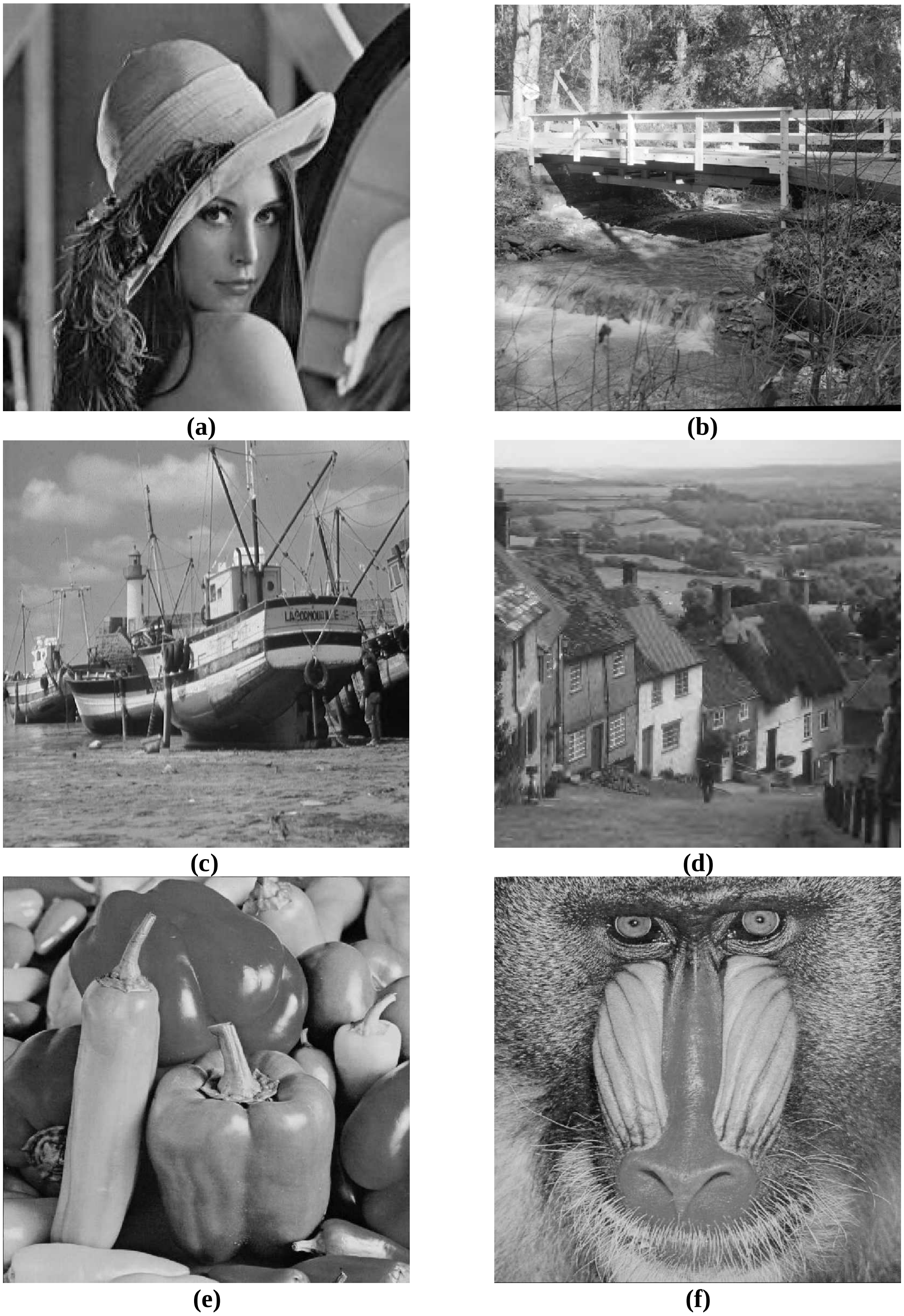
1.5 Standard Test Images

There are various standard test images, used extensively in literature, for this purpose. They are ‘Lena’, ‘Bridge’, ‘Boats’, ‘Goldhill’, ‘Pepper’, ‘Baboon’ etc. Here original image has shown in fig. 1.1.

1.6 Organization of the Thesis

Following the introduction, the rest of the thesis is organized as follows. Chapter 2 gives basics of image processing and noise filtering. Chapter 3 introduces Boundary Discriminative Noise Detection [41] for switching median filter. Chapter 4 presents Progressive Switching Median filter. Chapter 5 describes an Efficient Approach for Removing Impulse Noise [45].

Chapter 6 presents an Impulse Detection Method SM [13] for switching median filter. In Chapter 7 a novel Adaptive Noise Detection and Suppression ANDS [P2] has described. In Chapter 8 a novel Impulse Noise Detection has described .Then finally thesis has concluded in Chapter 9.



(a) **(b)**

(c) **(d)**

(e) **(f)**

Fig.1.1: Original test image of Lena, Bridge, Boat, Goldhill, Pepper, Baboon in a, b, c, d, e, f respectively.

Chapter 2

BASICS OF IMAGE PROCESSING AND NOISE FILTERING

2.1 Fundamentals of Digital Image Processing

Digital image processing generally refers to the processing of a 2-dimensional (2-D) picture signal by a digital hardware. An image is a 2-D function (signal), $\mathbf{X}(m,n)$, where m and n are the spatial (plane) coordinates. The magnitude of $\mathbf{X}$ at any pair of coordinates (m, n) is the *intensity* or *gray level* of the image at that point. In a digital image m, n and the magnitude of $\mathbf{X}$ are all finite and discrete quantities. Each element of this matrix (2-D array) is called a *picture element* or *pixel*.

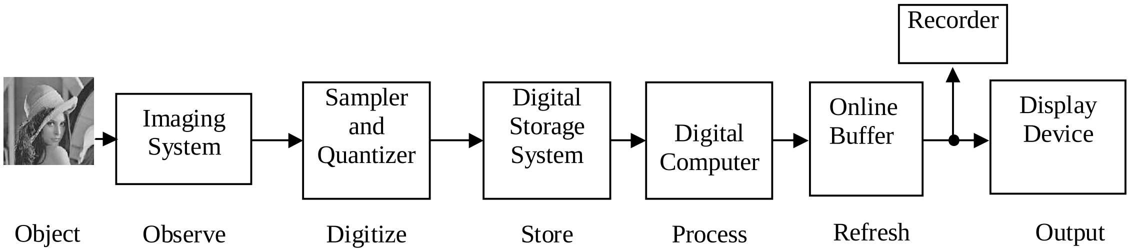


Fig.2.1 A typical digital image processing system

It is a hard task to distinguish between the domains of image processing and any other related area such as computer vision. Though, essentially not correct, image processing may be defined as a process where both input and output are images. At the high level of processing and after some preliminary processing, it is very common to perform some analysis, judgment or decision making or perform some mechanical operation (robot motion). These areas are the domains of artificial intelligence (AI), computer vision, robotics, etc.

Digital image processing has a broad spectrum of applications, such as digital television, photo-phone, remote sensing, image transmission and storage for business applications, medical processing, radar, sonar, and acoustic image processing, robotics, and computer aided manufacturing (CAM) and automated quality control in industries. Fig.2.1 depicts a typical image processing system. Except image acquisition and display, most of the images processing functions are implemented in software. A significant amount of basic image processing software is obtained commercially.

Fundamental steps in image processing are:

(a) Image acquisition

(b) Image Enhancement

(c) Image Transform

(d) Image Filtering and Restoration

(e) Color Image Processing

(f) Image Data Compression

(g) Morphological Processing

(h) Image segmentation

(i) Representation and description

(j) Image Analysis and Recognition, etc.

Image processing may be performed in the spatial domain or in a transform domain. Depending on the application, a suitable transform is used that may be discrete Fourier transform (DFT), discrete cosine transform (DCT), discrete wavelet transform (DWT), etc.

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. A familiar example of enhancement is when we increase the contrast of an image because "it looks better." It is important to keep in mind that enhancement is a subjective area of image processing. On the other hand, image restoration is very much objective. The restoration techniques are based on mathematical and statistical models of image degradation. Denoising (filtering) and deblurring tasks come under this category.

'Image restoration and filtering' is one of the prime areas of image processing and its objective is to recover the images from degraded observations. The techniques involved in image restoration and filtering are oriented towards modeling the degradations and then applying an inverse procedure to obtain an approximation of the original image.The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from scene. Second, humans can discern thousands of color shades and intensities, compared to shades of gray.

Compression, as the name implies, deals with techniques for reducing the storage required to save an image, or the bandwidth required to transmit it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar (perhaps inadvertently) to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually. On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

Representation and description almost always follow the output of segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections. Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. *Description*, also called *feature selection*, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another. *Recognition* is the process that assigns a label to an object based on its descriptors.

There are various types of imaging systems. X-ray, Gamma ray, ultraviolet, and ultrasonic imaging systems are used in biomedical instrumentation. In astronomy, the ultraviolet, infrared and radio imaging systems are used. Sonic imaging is performed for geological exploration. Microwave imaging is employed for radar applications. But, the most commonly known imaging systems are visible light imaging. Such systems are employed for applications like remote sensing, microscopy, measurements, consumer electronics, entertainment electronics, etc.

An image acquired by optical, electro-optical or electronic means is likely to be degraded by the sensing environment. The degradation may be in the form of sensor noise, blur due to camera miss focus, relative object camera motion, random atmospheric turbulence, and so on. The noise in an image may be due to a noisy channel if the image is transmitted through a medium. It may also be due to electronic noise associated with a storage-retrieval system.

2.2 Noise in Digital Images

In common use the word noise means unwanted signal. In electronics noise can refer to the electronic signal corresponding to acoustic noise (in an audio system) or the electronic signal corresponding to the (visual) noise commonly seen as 'snow' on a degraded television or video image. In signal processing or computing it can be considered data without meaning; that is, data that is not being used to transmit a signal, but is simply produced as an unwanted by-product of other activities. In Information Theory, however, noise is still considered to be information. In a broader sense, film grain or even advertisements in web pages can be considered noise.

The principle source of noise in digital images arises during image acquisition and or transmission. The performance of image sensors is affected by variety of factors, such as environmental conditions during image acquisitions, and by quality of sensing elements themselves. Images are 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 because of lighting or other atmospheric disturbances.

2.2.1 Some Important Noise Probability Density Function

Noise in imaging systems is usually either additive or multiplicative. This thesis deals only with additive noise which is zero-mean and *white*. White noise is spatially uncorrelated: the noise for each pixel is independent and identically distributed .Common noise models are:

a) Gaussian Noise

Because of the mathematical tractability in both the frequency and spatial domains, Gaussian noise models are used frequently in practice. In fact , this tractability is so convenient that it often results in Gaussian models being used in situations in which they are marginally applicable at best.

The PDF of Gaussian random variable, z, is given by

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}}\pi e^{-(z-\mu)^2 / 2\sigma^2} \tag{2.1}$$

Where z is gray level; μ= mean of average value of z; σ = the standard deviation. The standard deviation squared, σ^2, is called the variance of z.

The Gaussian distribution has an important property: to estimate the mean of a stationary Gaussian random variable, one can't do any better than the linear average. This makes Gaussian noise a worst-case scenario for nonlinear image restoration filters, in the sense that the improvement over linear filters is least for Gaussian noise. To improve on linear filtering results, nonlinear filters can exploit only the non-Gaussianity of the signal distribution.

b) Rayleigh noise

The PDF of Raleigh noise is given by

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z \geq 0 \\ 0 & \text{for } z < a \end{cases} \tag{2.2}$$

The mean and variance of this density are given by

$$\mu = a + \sqrt{\pi b / 4}$$

and

$$\sigma^2 = \frac{b(4-\pi)}{4}$$

Note the displacement from the origin and the fact that the basic shape of the density is skewed to the right. The Rayleigh density can be quite useful for approximating skewed histograms.

c) Erlang (Gamma) noise

The PDF of Erlang noise is given by

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!}e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases} \tag{2.3}$$

Where parameters are such that a>0, b is a positive integer, and "!" indicates factorial. The mean and variance of the density given by

$$\mu = \frac{b}{a}$$

$$\sigma^2 = \frac{b}{a^2}$$

Often the equation of gamma function is referred as gamma density; strictly speaking this is true when the denominator is gamma function, Γ (b). When the denominator is as shown in, the density is called Erlang density.

d) Exponential noise:

The PDF of exponential is given by

$$p(z) = \begin{cases} ae^{-az} & for\ z \geq 0 \\ 0 & for\ z < 0 \end{cases} \tag{2.4}$$

Where a > 0; the mean and variance density functions are

$$\mu = \frac{1}{a}$$

and $$\sigma^2 = \frac{1}{a^2}$$

Note that this PDF is a special case of Erlang PDF, with b = 1.

e) Uniform Noise

The PDF of uniform noise is given by

$$p(z) = \begin{cases} \frac{1}{(b-a)} & if\ a \leq z \leq b \\ 0 & otherwise \end{cases} \tag{2.5}$$

The mean of this density function and variance is given by

$$\mu = \frac{a+b}{2}$$

and $$\sigma^2 = \frac{(b-a)^2}{12}$$

f) Impulse (Salt and pepper) Noise :

The PDF of a bipolar impulse noise is given as

$$p(z)=\begin{cases} P_a & for\ z=a \\ P_b & for\ z=b \\ 0 & otherwise \end{cases} \tag{2.6}$$

If $b > a$, gray level b will appear as light dot in the image. Conversely level a will appear as a dark dot. If either P_a or P_b is zero, the impulse noise is called unipolar. If neither probability is zero, and if they are approximately equal, impulse noise value will resemble salt and pepper granules randomly distributed over the image. Hence, bipolar impulse noise is also called salt and pepper noise.

Let a digital image $\mathbf{X}(m,n)$, after being corrupted with SPN of density d, be represented by $X_{SPN}(m,n)$. Then, the noisy image $\mathbf{X}_{SPN}(m,n)$ is :

$$X_{SPN}(m,n)=\begin{cases} \mathbf{X}(m,n) & \text{with probability}, & p=1-d \\ 0 & & p=d/2 \\ 1 & & p=d/2 \end{cases} \tag{2.7}$$

The impulse noise occurs at random locations (m,n) with a probability of d. The SPN and RVIN are substitutive in nature. A digital image corrupted with RVIN of density d, $\mathbf{X}_{RVIN}(m,n)$:

$$X_{RVIN}(m,n)=\begin{cases} X(m,n) & \text{with probability}, & p=1-d \\ \eta(m,n) & \text{with probability}, & p=d \end{cases} \tag{2.8}$$

Here, $\eta(m,n)$ represents a uniformly distributed random variable, ranging from 0 to 1, that replaces the original pixel value $\mathbf{X}(m,n)$. The noise magnitude at any noisy pixel location (m,n) is independent of the original pixel magnitude.

2.3 Fundamental Noise Reduction Spatial Filters

As the fig.1.2 shows, the degradation process is modeled in this chapter as a degradation function that, together with an additive noise term , operates on an input image *f(x, y)* to produce a degraded image *g(x, y)*.

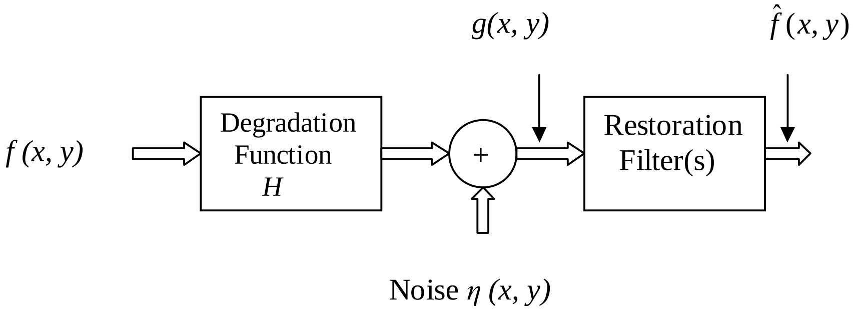


Fig.2.2: Degradation and Restoration model

Given *g(x, y)*, some knowledge of degradation function *H*, and some knowledge of additive noise term *η (x, y)*, the objective of noise reduction is to estimate $\hat{f} = (x, y)$ of the original image. If '*H*' is linear, position invariant process, then the degraded image is given in the spatial domain by

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y) \qquad (2.7)$$

Where *h (x, y)* = spatial representation of the degradation function; '*'represents spatial convolution. We know that the convolution in the spatial domain is equal to the multiplication in the frequency domain and is given as:

$$G(u, v) = H(u, v)\, F(u, v) + N(u, v) \qquad (2.8)$$

Where the terms in capital letters are the Fourier Transforms of the corresponding terms in equation (2.7).

2.3.1 Mean Filters

Mean filter is a linear filter. In this section we discuss types of mean filters

a) Arithmetic mean filter

This is the simplest of the mean filters. Let S_{xy} represent the set of coordinates in a rectangular subimage window of size $m \times n$, centered at point *(x, y)*. The arithmetic mean filtering process computes the average value of the corrupted image *g(x, y)* in the area defined by S_{xy}. The value of the restored image $\hat{f}$ at any point *(x, y)* is simply the arithmetic mean computed using the pixels in the region defined by S_{xy}. In other words,

$$\hat{f}(x,y)=\frac{1}{mn}\sum_{(s,t)\in Sxy} g(s,t) \qquad (2.9)$$

Where s and t are neighborhoods of x and y. This operation can be implemented using a convolution mask in which all coefficients have $1/mn$. A mean filter simply smoothes local variations in an image. Noise is reduced as a result of blurring.

b) Geometric mean filter

An image restored using a geometric mean filter is given by the expression

$$\hat{f}(x,y)=\left[\prod_{(s,t)\in S_{xy}} g(s,t)\right]^{\frac{1}{mn}} \qquad (2.10)$$

Here each restored pixel is given by the product of the pixels in the subimage window, raised to the power $1/mn$. A geometric mean filter achieves smoothing comparable to the arithmetic mean filter, but it tends to loose less image detail in the process.

c) Harmonic Mean Filter

The harmonic mean filtering operation is given by the expression

$$\hat{f}(x,y)=\frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}} \qquad (2.11)$$

The harmonic mean filters woks well for salt noise, but fails for pepper noise. It does well also with other types of noise like Gaussian noise.

d) Contraharmonic Mean Filter

The Contraharmonic mean filtering operation yields a restored image based on the expression

$$\hat{f}(x,y)=\frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}} \qquad (2.12)$$

Where Q is order of the filter. For positive value of Q, the filter eliminates pepper noise and for negative values of Q the filter eliminates salt noise. Both cannot eliminate noise simultaneously. The Contraharmonic filter reduces to the arithmetic mean filter if Q = 0, and to the harmonic mean filter if Q = -1.

2.3.2 Order Statistics Filter

Order statistic filter are nonlinear spatial filters whose response is based on ordering (ranking the pixels) contained in the image area encompassed by the filter, than replacing the value of the center pixel with the value determined by the ranking result.

a) Median Filter

It is the best known order-statistics filter which as its name suggests replaces the value of the pixel by the median of the gray levels in the neighborhood of that pixel:

$$\hat{f}(x,y) = median\{g_{(s,t)\in S_{xy}}(s,t)\} \qquad (2.13)$$

The original value of the pixel is included in the computation of the median. Median filters are particularly effective in the presence of both bipolar and unipolar impulse noise. Median filter provides excellent noise reduction capabilities, with considerably less blurring than linear smoothing filter of similar size.

The median, ξ, of a set of values is such that half the values in the set are less than or equal to ξ, and half are greater than equal to ξ. In order to perform median filtering at a point in an image we first sort the values of pixels in question and its neighbors, determine their median and assign the to the pixel. Thus the principle function of the median filter is to force points with distinct gray levels to be more like their neighbors. In fact isolated clusters of pixels that are light or dark with respect to there neighbors, and whose area is less than $n^2/2$ (one half of the filter area), are eliminated by an n×n median filter. The median represents the 50th percentile of a ranked set of numbers.

b) Max and Min filters

Median filter discussed above is by-far most used filter but by no means the only one. The median represents the 50th percentile of ranked set of the numbers. But using the 100th percentile result we get so called the Max filter, given by

$$\hat{f}(x,y) = \max\{g_{(s,t)\in S_{xy}}(s,t)\} \qquad (2.14)$$

This filter is useful for finding the brightest points in the image. Also, because the perpendicular noise has very low values, it is reduced by this filter as a result of max selection process in the sub image area S xy.

The 0th percentile filter is the min filter:

$$\hat{f}(x,y) = \min\{g_{(s,t)\in S_{xy}}(s,t)\} \qquad (2.15)$$

This filter is useful for finding the darkest points in the image. Also, it reduces the salt noise as a result of the min operation.

c) Midpoint filter

The midpoint filter simply computes the midpoint between the maximum and minimum values in the area encompassed by the filter:

$$\hat{f}(x,y) = 1/2[\max\{g_{(s,t)\in S_{xy}}(s,t)\} + \min\{g_{(s,t)\in S_{xy}}(s,t)\}] \qquad (2.16)$$

Here the filter combines order statistics and averaging. This filter works best for randomly distributed noise, like Gaussian or uniform noise.

d) Alpha-trimmed mean filter:

If we delete the *d*/2 lowest and *d*/2 highest gray-level values of *g(s, t)* in the neighborhood S_{xy}. Let $g_r(s, t)$ represent the remaining "*m*×*n*-*d*" pixels. A filter formed by the averaging of these remaining pixels is called alpha trimmed mean pixels.

$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t) \qquad (2.17)$$

Where the value of '*d*' can range from '0 to *mn*-1'. When *d*=0 the alpha trimmed filter reduces to the arithmetic mean filter. If we take d = (*mn*-1)/2, the filter becomes a median filter.

Chapter 3

SWITCHING MEDIAN FILTER WITH BOUNDARY DISCRIMINATIVE NOISE DETECTION

3.1 Introduction

To determine whether the current pixel is corrupted or not, the BDND [41] (Boundary Discriminative Noise Detection) algorithm first classifies the pixels of a localized window, centering on the current pixel, into three groups—*lower intensity impulse noise*, *uncorrupted pixels*, and *higher intensity impulse noise*. The center pixel will then be considered as "uncorrupted," provided that it belongs to the "uncorrupted" pixel group, or "corrupted." For that, two boundaries that discriminate these three groups require to be accurately determined for yielding very high noise detection accuracy even up to 70% noise corruption. Extensive simulation results conducted on both monochrome and color images under a wide range (from10% to 70%) of noise corruption clearly show that this switching median filter substantially outperforms existing median-based filters, in terms of suppressing impulse noise while preserving image details. BDND is algorithmically simple, suitable for real-time implementation and application.

3.2 Impulse-Noise Detection

3.2.1 Noise Models

Four impulse noise models are implemented, for extensively examining the performance of BDND filter with consideration of practical situations. Each model is described in detail as follows.

1) Noise Model 1: Noise is modeled as salt-and-pepper impulse noise as practiced (e.g., in [9]). Pixels are randomly corrupted by two fixed extremal values, 0 and 255 (for 8-bit monochrome image), generated with the same probability. That is, for each image pixel at location (i, j) with intensity value $s_{i,,j}$, the corresponding pixel of the noisy image will be $x_{i,j}$, in which the probability density function of $x_{i,j}$ is

$$f(x) = \begin{cases} \frac{p}{2}, & for\, x = 0 \\ 1 - p, & for\, x = s_{i,j} \quad [Model\, 1] \\ \frac{p}{2}, & for\ x = 255 \end{cases}$$

where p is the noise density.

2) Noise Model 2: For the Model 2, it is similar to Model 1, except that each pixel might be corrupted by either "pepper" noise (i.e., 0) or "salt" noise with *unequal* probabilities. That is

$$f(x)=\begin{cases}\frac{p_1}{2}, & for\ x=0\\ 1-p, & for\ x=s_{i,j} \quad [Model\ 2]\\ \frac{p_2}{2}, & for\ x=255\end{cases}$$

Where $p=p_1+p_2$ is the noise density and $p_1 \neq p_2$.

3) Noise Model 3: Instead of two fixed *values*, impulse noise could be more realistically modeled by two fixed *ranges* that appear at both ends with a length of each, respectively. For example, if *m* is 10, noise will equal likely be any values in the range of either [0, 9] or [246, 255] . That is

$$f(x)=\begin{cases}\frac{p}{2m}, & for\ \ 0\le x<m\\ 1-p, & for\ x=s_{i,j} \quad [Model\ 3]\\ \frac{p}{2m}, & for\ \ 255-m<x\le 255\end{cases}$$

Where p is the noise density.

4) Noise Model 4: Model 4 is similar to Model 3, except that the densities of low-intensity impulse noise and high-intensity impulse noise are *unequal.* That is

$$f(x)=\begin{cases}\frac{p_1}{m}, & for\ \ 0\le x<m\\ 1-p, & for\ x=s_{i,j} \quad [Model\ 4]\\ \frac{p_2}{m}, & for\ \ 255-m<x\le 255\end{cases}$$

where $p=p_1+p_2$ the noise density and $p_1 \neq p_2$.

3.2.2 Noise Detection

The BDND algorithm is applied to each pixel of the noisy image in order to identify whether it is “uncorrupted” or “corrupted.” After such an application to the entire image, a two-dimensional binary decision map is formed at the end of the noise detection stage, with “0s” indicating the positions of “uncorrupted” pixels, and “1s” for those “corrupted” ones. To accomplish this objective, all the pixels within a pre-defined window that center around the considered pixel will be grouped into three clusters; hence, two boundaries and are required to be determined. For each pixel $x_{i,j}$ being considered, if $0<= x_{i,j}<=b_1$, the pixel will be assigned to the *lower*-intensity cluster; otherwise, to the *medium*-intensity cluster for $b_1<x_{i,j}<=b_2$ or to the *high*-intensity cluster for $b_2<x_{i,j}<=255$.

Obviously, if the center pixel being considered falls onto the middle cluster, it will be treated as "uncorrupted," since its intensity value is neither relatively low nor relatively high. Otherwise, it is very likely that the pixel has been corrupted by impulse noise. Clearly, the accuracy of clustering results (hence, the accuracy of noise detection) ultimately depends on how accurate the identified boundaries b_1 and b_2 are.

First, we shed the light of our intuition that leads to the development of the BDND algorithm simply based on the histogram distribution of any subimage extracted from the simulated noisy image "Lena" corrupted by 70% impulse noise density based on the above-mentioned Noise Model 1. For illustrating a "typical" histogram distribution, the subimage chosen bears a "neutral" image content, meaning that the content is neither too "flat" (containing low frequency) nor too "busy" (containing high frequency). It could be observed that the distribution presented at the two ends of the distribution is most likely contributed by impulse noise. Furthermore, the locations of two distinct gaps (or valleys) mark the most possible positions of the two boundaries, respectively, that clearly separate the impulse noise regions (at the two ends) from the uncorrupted pixel region (a much wider region in between); thus, dividing all the pixels within the window into three groups—the lower intensity impulse noise, the uncorrupted pixels (in the middle) and the higher intensity impulse noise.

The *boundary discriminative* process consists of two iterations, in which the second iteration will only be invoked conditionally. In the first iteration, an enlarged local window with a size of 11×11 (empirically determined) is used to examine whether the considered pixel is an uncorrupted one. If the pixel fails to meet the condition to be classified as "uncorrupted" (i.e., not falling onto the middle cluster), the second iteration will be invoked to further examine the pixel based on a more confined local statistics by using a 3×3 window. In summary, the steps of the BDND are:

Step 1) Impose a 11×11 window, which is centered around the current pixel.

Step 2) Sort the pixels in the window according to the ascending order and find the median, *med*, of the sorted vector V_o .

Step 3) Compute the intensity difference between each pair of adjacent pixels across the sorted vector V_o and obtain the difference vector V_d.

Step 4) For the pixel intensities between 0 and *med* in the V_o, find the maximum intensity difference in the V_d of the same range and mark its corresponding pixel in the V_o as the boundary b_1 .

Step 5) Likewise, the boundary b_2 is identified for pixel intensities between *med* and 255; three clusters are, thus, formed.

Step 6) If the pixel belongs to the middle cluster, it is classified as “uncorrupted” pixel, and the classification process stops; else, the second iteration will be invoked in the following.

Step 7) Impose a 3×3 window, being centered around the concerned pixel and repeat Steps 2)–5).

Step 8) If the pixel under consideration belongs to the middle cluster, it is classified as “uncorrupted” pixel; otherwise, “corrupted.”

255	255	47	255	39
50	255	255	0	0
0	0	202	224	205
62	255	0	0	255
255	72	81	0	179

Fig. 3.1. Example for noise detection

For the understanding of the algorithmic steps mentioned above, a 5×5 (instead of 11×11) windowed subimage with the center pixel “202” (being boxed) is used as an example for illustrating the BDND process as follows:

• Pixel intensities are sorted in the ascending order and represented as a vector, where the median *med* is 81 ; i.e., V_o=[0 0 0 0 0 0 0 39 47 50 62 72 81 179 202 205 224 255 255 255 255 255 252 255 255 255].

• The vector of intensity differences between each pair of two adjacent pixels in Vo is computed as: V_d=[0 0 0 0 0 0 **39** 8 3 12 10 9 **98** 23 3 22 31 0 0 0 0 0 0 0].

• For the pixels with intensities between 0 and *med* in the , the corresponding maximum difference in the V_d is 39, which is the difference between the pixel intensities 0 and 39.

• For the pixels with intensities between *med* and 255 in the V_o, the maximum difference in the V_d is 98, which is the difference between the pixel intensities 81 and 179.

• Hence, b_1=0 and b_2=81. Thus, the lower intensity cluster is{0, 0, 0, 0, 0, 0, 0},, the medium-intensity cluster is{39, 47, 50, 62, 72, 81} and the higher intensity cluster is {179, 202, 205, 224, 255, 255, 255, 255, 255, 255, 255, 255}.

• Since the center pixel “202” belongs to the higher intensity cluster, hence, the second iteration needs to be invoked, and a 3×3 window is imposed and centered around it.

$W_{3\times3}$ =

255	255	0
0	202	224
255	0	0

- Now, the pixel intensities are sorted and represented in the vector form: Vo=[0 0 0 0 202 224 255 255 255].
- As before, the vector of intensity differences is computed: V_d=[0 0 0 202 22 31 0 0].
- The first maximum difference is 202, which is the difference between the pixel intensities 0 and 202. The second maximum difference is 31, which is the difference between the pixel intensities 224 and 255.
- Hence,b_1=0 and b_2=224. Thus, the lower intensity cluster is {0, 0, 0, 0}, the medium-intensity cluster is {202, 224}, and the higher intensity cluster is {255, 255, 255}.
- At the end of the discrimination process, the center pixel "202" is classified as an "uncorrupted" pixel, since it belongs to the middle cluster.

TABLE 3.1

Suggested window size for the estimated noise density level p

Noise Density	$\mathbf{W_d \times W_d}$
0%<p<=20%	3×3
20%<p<=40%	5×5
>40	11×11

3.2.3 Color Image Noise Detection

As the most directly used color space for digital image processing, the RGB color space is chosen in work to represent the color images. In the RGB color space, each pixel at the location *(i, j)* can be represented as color vector $s_{i,j} = \left(s_{i,j}^R, s_{i,j}^G, s_{i,j}^B\right)$, where $s_{i,j}^R$, $s_{i,j}^G$ and $s_{i,j}^B$ are the *red* (*R*), *green* (*G*), and *blue* (*B*) components, respectively. The noisy color images are modeled by injecting the salt-and-pepper noise randomly and independently to each of these color components. That is, when a color image is being corrupted by the noise density, it means that *each* color component is being corrupted by *p*. Thus, for each pixel $s_{i,j}$, the corresponding pixel of the noisy image will be denoted as $x_{i,j} = \left(x_{i,j}^R, x_{i,j}^G, x_{i,j}^B\right)$, in which the probability density functions of each color components can be one of the noise models described earlier. In this work, Noise Model 1 is used for performance demonstration.

The process of extending the noise detection algorithm to corrupted color images is straightforward. The BDND algorithm will be simply applied to *R*-, *G*-, and *B*-planes individually, and three binary decision maps are obtained.

3.3 Noise-Adaptive Filtering

Although the major contributions of making the entire switching median filter being *noise-adaptive* come from the impulse-noise detection as described in the previous section, the post-detection filtering to be discussed in this section also contributes to the overall denoising performance. Based on the filtering process described in [14], in the follow-up two subsections, we shall highlight this aspect and provide two additional improvements on the filtering stage.

3.3.1 Simplified Noise-Density Estimation

In order to determine the window size of the filtering window, the limit of the maximum window size requires to be determined first. For that, Table I is empirically established based on multiple test images, in which different window sizes are suggested for different noise-density levels of corruption estimated. To conduct the estimation of noise density, NASM involves a set of sophisticated procedures (such as quad-tree decomposition) [14]. On the contrary, the noise-density estimation performed in the BDND is much simpler, simply by counting the number of 1s on the binary decision map obtained in the impulse-noise detection stage conducted earlier. Based on the binary decision map, "*no filtering*" is applied to those "uncorrupted" pixels, while the SM filter with an adaptively determined window size is applied to each "corrupted" one.

3.3.2 Algorithmic Improvement on Filtering

First, *the maximum window size* is limited to 7×7 (instead of 11×11 as suggested in [14]) in order to avoid severe blurring of image details at high noise density cases (i.e., p>50%). After that, the filter's window size is obtained in a similar way as that in [42] with a slight modification as follows.

In the NASM [14], starting with W_F=3, the filtering window iteratively extends outward by one pixel in all the four sides of the window, provided that the number of uncorrupted pixels (denoted by N_c) is less than half of the total number of pixels (denoted by $S_{in}=1/2[W_F \times W_F]$)

within the filtering window, while . In this work, an additional condition is further imposed, such that the filtering window will also be extended when the number of uncorrupted pixels is equal to zero. Therefore, the second change for improvement is that while ($N_c < S_{in}$ and $W_F <= W_{D1}$) *or* (N_c=0) , window will be extended by one pixel outward in all the four sides of the window. In [14], the current pixel is included in the filtering (ranking) process. Note that the current pixel has already been identified as “corrupted;” thus, our third change in the BDND is *to exclude the current pixel* in the process of filtering; that is, only those “uncorrupted” pixels within the window are considered for the process of ranking. This will, in turn, yield a better filtering result with less distortion.

3.3.3 Color Image Denoising

The switching median filtering scheme can be extended to denoise corrupted color images via the *scalar median* filtering approach as well as the *vector median* filtering approach. The scalar approach treats each color component as an independent entity; that is, the same filtering scheme will be applied to *R*, *G*, and *B*-planes independently, as if each plane is a separate monochrome image. The filtered *R*, *G*, and *B*-planes will be then combined to form the recovered color image.

3.4 Simulation Results

Intensive simulations were carried out using several monochrome images, from which “Lena,” “Peppers,” and “Baboon” are chosen for demonstrations see fig.

Fig.3.2 Original image of Pepper (512×512), Lena (512×512), Baboon (512×512) taken for demonstration in 1, 2, 3 respectively.

Table-3.2. PSNR performance of the switching median filter with the BDND on Pepper image, corrupted by noise density (10-70%) of “salt and pepper” noise

Noise density (%)	Input PSNR(dB)	Output PSNR **BDND**	Output PSNR MED(3×3)	Output PSNR MED(5×5)	Output PSNR MED(7×7)
10	15.22	42.73	31.03	28.09	25.88
20	12.34	39.36	27.11	26.09	24.13
30	10.51	36.78	21.98	24.18	22.51
40	9.28	33.97	18.4	22.84	21.61
50	8.35	32.61	14.95	20.51	20.30
60	7.52	30.69	12.13	17.37	18.89
70	6.84	27.17	9.79	13.41	16.24

In table 3.2 PSNR performances of BDND for Pepper image are given, from that we can see the how BDND performance better than other median filters (MED(3×3), MED(5×5), MED(7×7)). BDND performance is good for higher noise density even up to 80%.Its giving less blurring effect compare to MED(5×5), MED(7×7) even for higher noise density. This PSNR performance table 3.2 is plotted in fig 3.4. In fig. 3.3 noisy Pepper images are taken with different noise density and corresponding restored image by different techniques are shown. Visual quality of restored image by BDND is very good comparing to MED (3×3), MED (5×5), and MED (7×7) even for higher noise density. See in fig 3.3 when noise density increased MED (3×3), MED (5×5), MED (7×7) are failed. But BDND is giving good visual quality and also good PSNR.

In table 3.3 PSNR comparisons of BDND with different exiting technique on Lena (512×512) image corrupted by “salt and pepper” noise density of (10-30%) are given. Fig.3.5. shows noisy images of Lena (512×512) corrupted by salt and pepper noise with noise density of 10, 20, 30, 50% respectively and corresponding restored image by BDND.

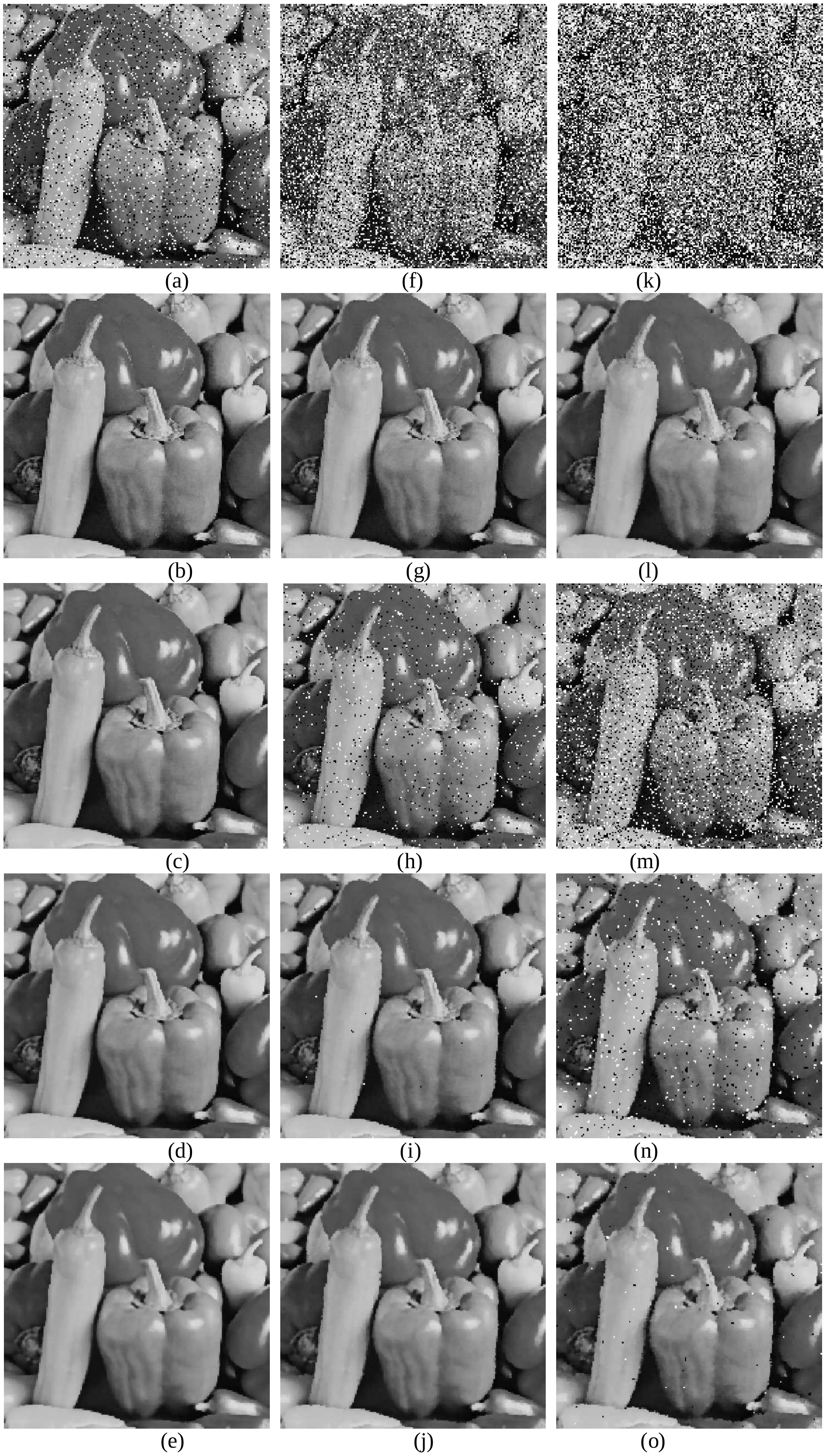


Fig.3.3: a, f, k are noisy (salt & pepper) Pepper image with noise density of 10, 40 and 60% respectively and corresponding filtered images by BDND are in b, g, l, by MED (3×3) in c, h, m, by MED(5×5) in d, i, n, and by MED(7×7) in e, j ,o respectively.

Algorithm	Lena Corrupted with Noise density		
	10%	20%	30%
MF(3×3)	31.19 dB	28.48 dB	25.45 dB
MF(5×5)	29.45 dB	28.91 dB	28.43 dB
MMEM[17]	30.28 dB	29.63 dB	29.05 dB
Florencio's[10]	33.69 dB	32.20 dB	30.95 dB
PMCWF [18]	35.70 dB	32.95 dB	31.86 dB
AMF(3×3) [22]	33.79 dB	30.65 dB	26.26 dB
AMF(5×5) [22]	30.11 dB	28.72 dB	27.84 dB
CMF(3×3) [21]	38.05 dB	31.79 dB	26.22 dB
CMF(5×5) [21]	36.32 dB	33.52 dB	30.33 dB
SDROM [23]	37.93 dB	34.10 dB	29.80 dB
BDND [41]	**42.08 dB**	**38.84 dB**	**36.20 dB**

Table 3.3 PSNR comparisons of BDND with different exiting technique on Lena (512×512) image corrupted by "salt and pepper" noise density of (10-30%).

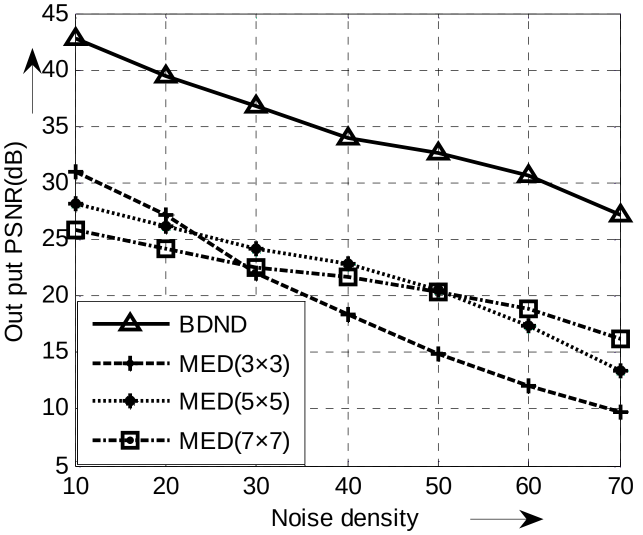
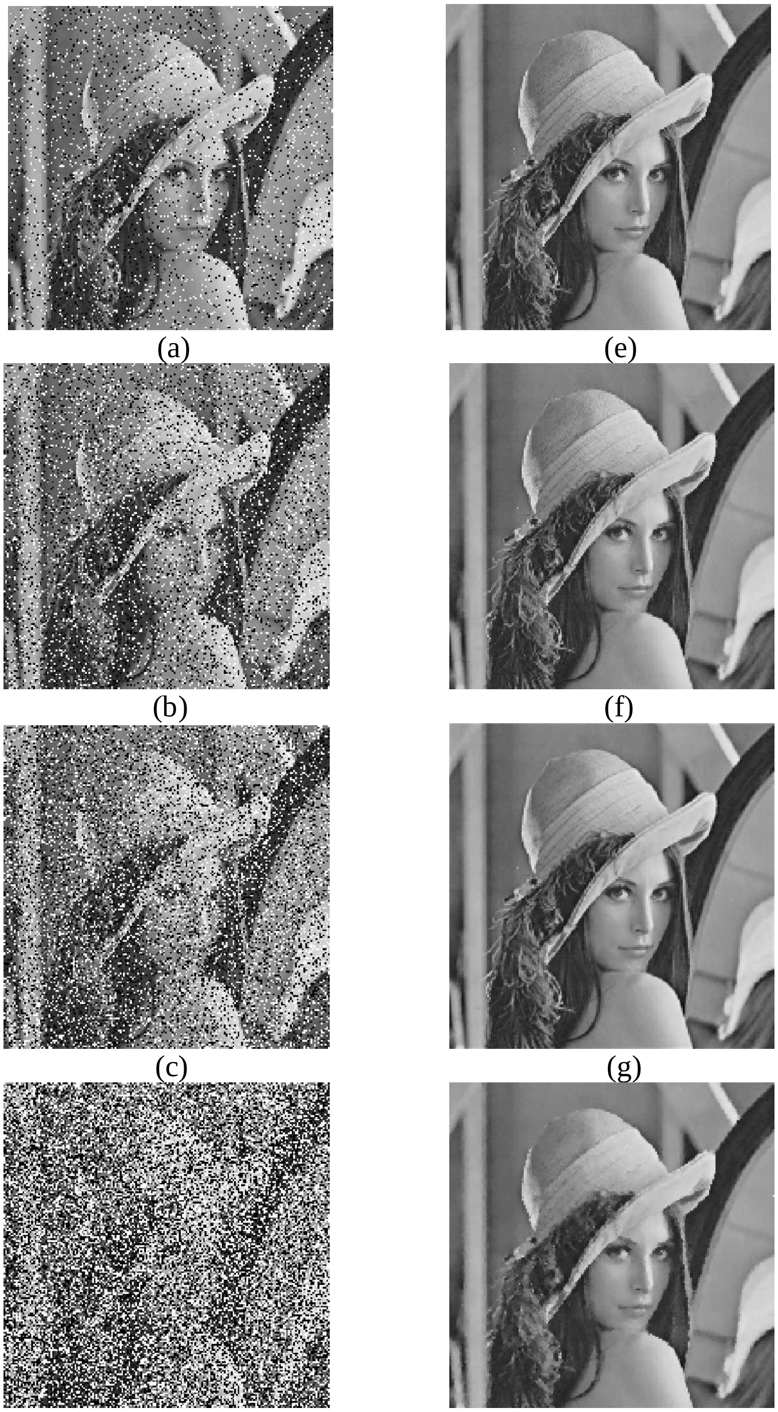


Fig 3.4: PSNR Plot for table 3.2

In this chapter only one noise model has taken for demonstration that is noise model 1. We can also apply this technique to other noise model. In fig 3.6 BDND performances for color images are given for 10% and 20% of salt and pepper noise. BDND work very well on color image also.

3.5 Conclusion

In this chapter BDND based noise detection has implemented, which has been further incorporated into the framework of switching median filter as a very powerful image denoising scheme. Extensive simulation results reveal that this filter consistently outperforms on the many existing filters (especially, with a large margin of improvement at extremely high noise density corruption) by attaining much higher PSNR across a wide range of noise densities, from 10% to 80%. Another tremendous advantage of BDND algorithm is fairly simple to implement for real-time image applications.



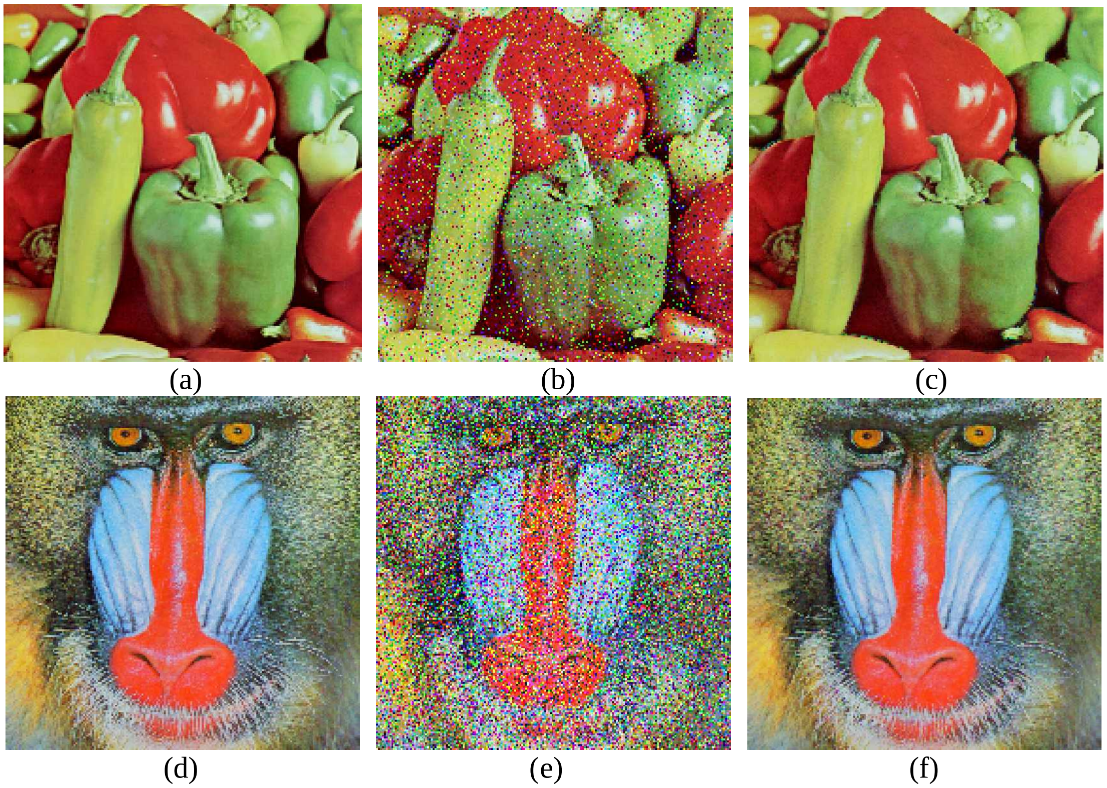
(a) (e)

(b) (f)

(c) (g)

(d) (h)

Fig.3.5 a, b, c, d are Lena (512×512) images corrupted by salt and pepper noise with noise density of 10, 20, 30, 50% respectively and corresponding restored image by BDND are in e, f, g, h respectively



(a) (b) (c)

(d) (e) (f)

Fig. 3.6: a is the original color image of Pepper, b is noisy Pepper image with 10 % salt and pepper noise and corresponding filtered image in c by BDND, d is the original color image of Baboon, e is noisy Baboon image 20 % salt and pepper noise and corresponding filtered image in f by BDND

Chapter 4

PROGRESSIVE SWITCHING MEDIAN FILTER

4.1 Introduction

In this chapter a median-based filter, progressive switching median (PSM) [12] filter, is implemented to restore images corrupted by salt–pepper impulse noise. The algorithm is developed by the following two main points: 1) switching scheme—an impulse detection algorithm is used before filtering, thus only a proportion of all the pixels will be filtered and 2) progressive methods—both the impulse detection and the noise filtering procedures are progressively applied through several iterations. The noise pixels processed in the current iteration are used to help the process of the other pixels in the subsequent iterations. A main advantage of such a method is that some impulse pixels located in the middle of large noise blotches can also be properly detected and filtered. Therefore, better restoration results are expected, especially for the cases where the images are highly corrupted.

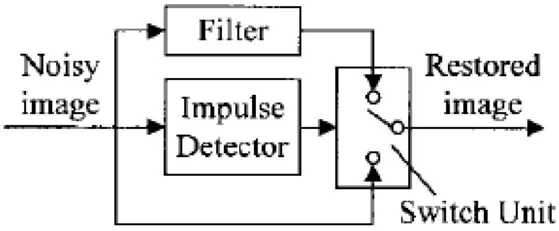


Fig. 4.1 A general framework of switching scheme-based image filters.

4.2 PSM Filter

4.2.1 Impulse Detection

Similar to other impulse detection algorithms, this impulse detector is implemented by prior information on natural images, i.e., a noise-free image should be locally smoothly varying, and is separated by edges [4]. The noise considered for this algorithm is only salt–pepper impulsive noise which means: 1) only a proportion of all the image pixels are corrupted while other pixels are noise-free and 2) a noise pixel takes either a very large value as a positive impulse or a very small value as a negative impulse. In this chapter, we use noise ratio $R(0 \le R \le 1)$ to represent how much an image is corrupted. For example, if an image is corrupted by R = 30% impulse noise, then 15% of the pixels in the image are corrupted by positive impulses and 15% of the pixels by negative impulses.

Two image sequences are generated during the impulse detection procedure. The first is a sequence of gray scale images, $\{x_{(i,j)}^{(0)}, x_{(i,j)}^{(1)}, x_{(i,j)}^{(2)}, \ldots. x_{(i,j)}^{(n)} \ldots..\}$, where the initial image $x_{(i,j)}^{(0)}$ is noisy image itself , (i , j) is position of pixel in image, it can be $1 \le i \le$ M, $1 \le j \le$ N where

M and N are the number of the pixel in horizontal and vertical direction respectively, and $x_{(i,j)}^{(n)}$ is image after n^{th} iteration. The second is a binary flag image sequence, $\{ f_{(i,j)}^{(0)}, f_{(i,j)}^{(1)}, f_{(i,j)}^{(2)}, ... f_{(i,j)}^{(n)} \}$ where the binary flag $f_{(i,j)}^{(n)}$ is used to indicate whether the pixel at *(i, j)* in noisy image detected as noisy or noise-free after n^{th} iteration. If $f_{(i,j)}^{(n)}$ =0 means pixel at *(i, j)* has been found as noise-free after n^{th} iteration and if $f_{(i,j)}^{(n)}$ =1 means pixel at *(i, j)* has been found as noisy after n^{th} iteration. Before the first iteration, we assume that all the image pixels are good, i.e. $f_{(i,j)}^{(0)}$ =0 for all *(i, j).*

In the nth iteration (n= 1, 2, 3…) for each pixel $x_{(i,j)}^{(n-1)}$ we first find out the median value of the samples in a $W_D \times W_D$ (W_D is an odd integer not smaller than 3) window centered about it. To represent the set of the pixels within a $W_D \times W_D$ window centered about $x_{(i,j)}^{(n-1)}$ is $x_{(i+k,j+l)}^{(n-1)}$ where $-W \le k \le W, -W \le l \le W$ k ≤ W, -W≤ l ≤ W and W≥1, then we have median value of this window $m_{(i,j)}^{(n-1)}$ is

$$m_{(i,j)}^{(n-1)} = \text{median} \, (x_{(i+k,j+l)}^{(n-1)}) \qquad (4.1)$$

The difference between $m_{(i,j)}^{(n-1)}$ and $x_{(i,j)}^{(n-1)}$ provides us with a simple measurement to detect impulses

$$f_{(i,j)}^{(n)} = \begin{cases} f_{(i,j)}^{(n-1)}, & if \left| x_{(i,j)}^{(n-1)} - m_{(i,j)}^{(n-1)} \right| < T \\ 1, & otherwise \end{cases} \qquad (4.2)$$

where *T* is a predefined threshold value. Once a pixel *(i, j)* is detected as an impulse, the value of

$x_{(i,j)}^{(n)}$ is subsequently modified

$$x_{(i,j)}^{(n)} = \begin{cases} m_{(i,j)}^{(n-1)}, & if \; f_{(i,j)}^{(n)} \neq f_{(i,j)}^{(n-1)} \\ x_{(i,j)}^{(n-1)}, & if \; f_{(i,j)}^{(n)} = f_{(i,j)}^{(n-1)} \end{cases} \qquad (4.3)$$

Suppose the impulse detection procedure is stopped after the N_Dth iteration, then two output images-$x_{(i,j)}^{(N_D)}$ and $f_{(i,j)}^{(N_D)}$ are obtained, but only $f_{(i,j)}^{(N_D)}$ is useful for our noise filtering algorithm.

4.2.2 Noise Filtering

Like the impulse detection procedure, the noise filtering procedure also generates a gray scale image sequence, $\{y_{(i,j)}^{(0)}, y_{(i,j)}^{(1)}, y_{(i,j)}^{(2)}, ... y_{(i,j)}^{(n)} .....\}$ and a binary flag image sequence

$\{g_{(i,j)}^{(0)}, g_{(i,j)}^{(1)}, ...g_{(i,j)}^{(n)}.....\}$. In the gray scale image sequence, we still use $y_{(i,j)}^{(0)}$ to denote the pixel value at position *(i, j)* in the noisy image to be filtered and use $y_{(i,j)}^{(n)}$ to represent the pixel value at position *(i, j)* in the image after the *n*th iteration. In a binary flag image $g_{(i,j)}^{(n)}$, the value $g_{(i,j)}^{(n)}$=0 means the pixel *(i, j)* is good and $g_{(i,j)}^{(n)}$ = 1 means it is an impulse that should be filtered. A difference between the impulse detection and noise-filtering procedures is that the initial flag image $g_{(i,j)}^{(0)}$ of the noise-filtering procedure is not a blank image, but the impulse detection result $f_{(i,j)}^{(N_D)}$, i.e., $g_{(i,j)}^{(0)} = f_{(i,j)}^{(N_D)}$.

In the *n*th iteration (n = 1; 2; ….), for each pixel $y_{(i,j)}^{(n-1)}$, we also first find its median value $m_{(i,j)}^{(n-1)}$ of a $W_F \times W_F$ (W_F is an odd integer and not smaller than 3) window centered about it. However, unlike that in the impulse detection procedure, the median value here is selected from only good pixels with $g_{(i,j)}^{(n-1)} = 0$ in the window.

Let *M* denote the number of all the pixels with $g_{(i,j)}^{(n-1)} = 0$ in the $W_F \times W_F$ window. If *M* is odd, then

$$m_{(i,j)}^{(n-1)} = median\{y_{(i,j)}^{(n-1)} \,\big|\, g_{(i,j)}^{(n-1)} = 0, (i, j) \in W_F \times W_F\} \qquad (4.5)$$

The value of $y_{(i,j)}^{(n)}$ is modified only when the pixel *(i, j)* is an impulse and *M* is greater than 0:

$$y_{(i,j)}^{(n)} = \begin{cases} m_{(i,j)}^{(n-1)}, & if\ g_{(i,j)}^{(n-1)} = 1; M > 0 \\ y_{(i,j)}^{(n-1)}, & else \end{cases} \qquad (4.6)$$

Once an impulse pixel is modified, it is considered as a good pixel in the subsequent iterations

$$g_{(i,j)}^{(n)} = \begin{cases} g_{(i,j)}^{(n-1)}, & if\ y_{(i,j)}^{(n)} = y_{(i,j)}^{(n-1)} \\ 0, & if\ y_{(i,j)}^{(n)} = m_{(i,j)}^{(n-1)} \end{cases} \qquad (4.7)$$

The procedure stops after the N_Fth iteration when all of the impulse pixels have been modified, i.e.,

$$\sum_{(i,j)} g_{(i,j)}^{N_F} = 0 \qquad (4.8)$$

Then we obtain the image $\{y_{(i,j)}^{(N_F)}\}$ which is our restored output image.

4.3 Simulation Results and Conclusion

In this experiment, the original test images are corrupted with fixed valued salt and pepper noise, where the corrupted pixels take on the values of either 0 or 255 with equal probability.

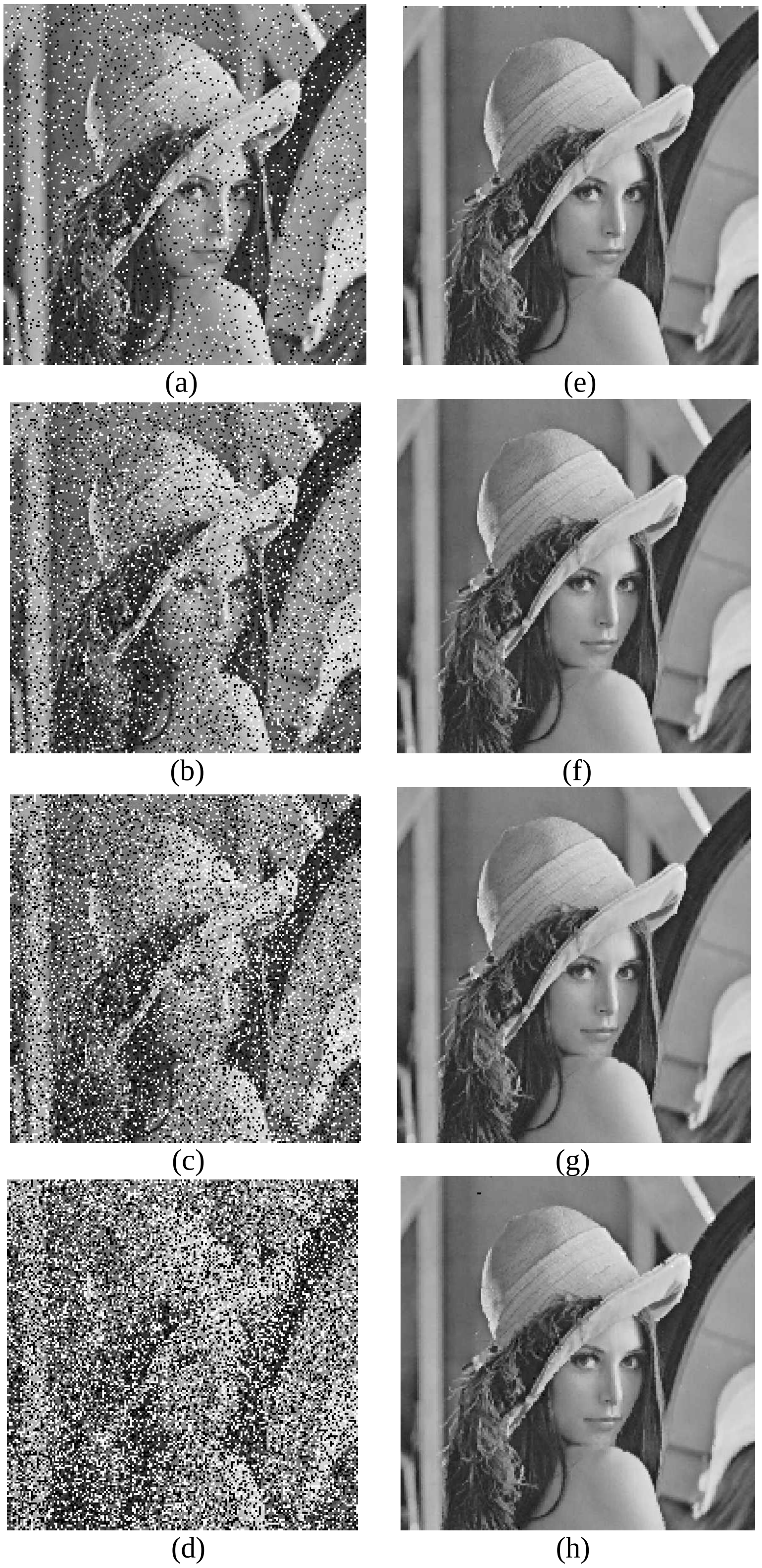
To implement the PSM algorithm, four parameters must be predetermined. They are the filtering window size W_F, the impulse detection window size W_D, the impulse detection iteration number N_D and the impulse detection threshold T. Experiments show that almost all the best restoration results are obtained when $W_F = 3$ and $N_D = 2$. In addition, these two parameters are not sensitive to noise rate and image type. Therefore, we simply set both W_F =3 and $N_D = 2$. From experiment we have seen that T will affect the restored image quality. T= 40 gives good PSNR and restored image quality.

Table 4.1. PSNR Performance of Different Algorithms for Lena image corrupted with salt and pepper noise

Algorithm	Lena Corrupted with Noise density		
	10%	20%	30%
MF(3×3)	31.19 dB	28.48 dB	25.45 dB
MF(5×5)	29.45 dB	28.91 dB	28.43 dB
MMEM [17]	30.28 dB	29.63 dB	29.05 dB
Florencio’s[10]	33.69 dB	32.20 dB	30.95 dB
PMCWF [18]	35.70 dB	32.95 dB	31.86 dB
AMF(3×3)[22]	33.79 dB	30.65 dB	26.26 dB
AMF(5×5)[22]	30.11 dB	28.72 dB	27.84 dB
CMF(3×3)[21]	38.05 dB	31.79 dB	26.22 dB
CMF(5×5)[21]	36.32 dB	33.52 dB	30.33 dB
SDROM [23]	37.93 dB	34.10 dB	29.80 dB
PSM [12]	**39.34 dB**	**35.53 dB**	**34.13 dB**

In table 4.1 PSNR comparisons of PSM with different exiting technique on Lena(512×512) image corrupted by “salt and pepper” noise density of (10-30%) are given. PSM performs better than other median-based methods, especially when noise ratios are high. Fig.4.2 shown noisy images of Lena (512×512) corrupted by salt and pepper noise with noise density of 10, 20, 30, 40% respectively and corresponding restored image by PSM . Similarly for Pepper image in fig. 4.3 Both the simple 3×3 median filter and the switch median filter can preserve image details but many noise pixels are remained in the image. The CWM filter performs better than simple median filter, but it still influences good pixels and misses many impulse pixels. The iterative median filter removes most of the impulses, but many good pixels are also modified, resulting in blurring of the image. Since the iterative switching filter does not modify good pixels in the image, it maintains image details better than the iterative median filter, but many noise blotches still remained in the image.

Dramatic restoration results are obtained by PSM filter. It can remove almost all of the noise pixels while preserve image details very well.



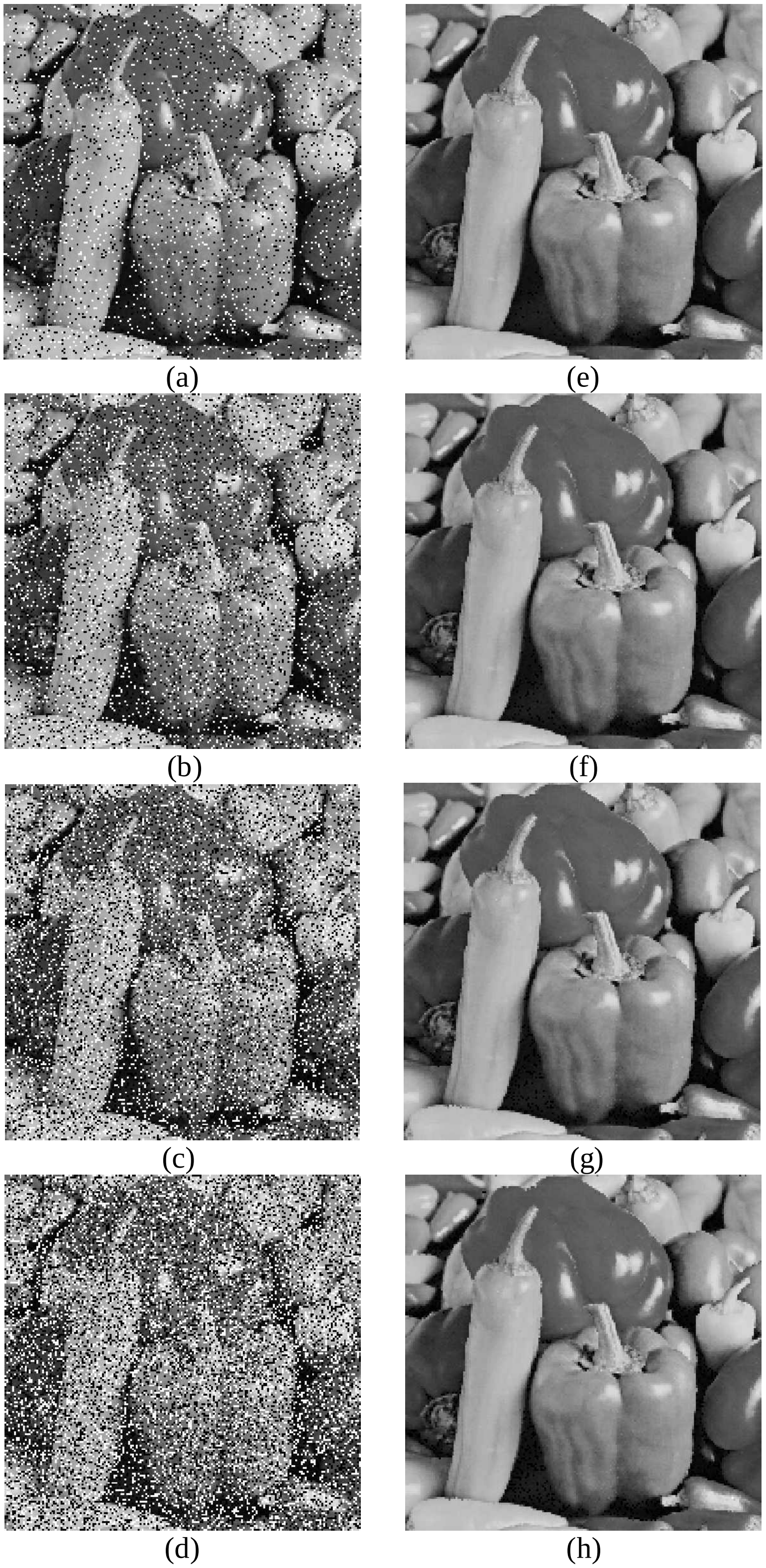
(a) (e)

(b) (f)

(c) (g)

(d) (h)

Fig. 4.2 a, b, c, d are noisy images of Lena (512×512) corrupted by salt and pepper noise with noise density of 10, 20, 30, 40% respectively and corresponding restored image by PSM are in e, f, g, h.



(a) (e)

(b) (f)

(c) (g)

(d) (h)

Fig. 4.3 a, b, c, d are noisy images of Pepper (512×512) corrupted by salt and pepper noise with noise density of 10, 20, 30, 40% respectively and corresponding restored image by PSM are in e, f, g, h.

Chapter 5

DETAIL-PRESERVING APPROACH FOR REMOVING IMPULSE NOISE IN IMAGES

5.1 Introduction

In this chapter, an efficient approach for removing impulse noise [45] from corrupted images while preserving image details, based on alpha-trimmed mean, is implemented. It is well known that the alpha-trimmed mean [43] see in equation 2.17 is a special case of the order-statistics filter [44]. The difference between this approach and [43],[44] is that this algorithm [45] uses the alpha-trimmed mean in impulse noise detection instead of pixel value estimation. Also, it applies the filtering process to only the identified noisy pixels instead of all image pixels. Extensive experimental results show that this algorithm performs significantly better than many other well-known techniques. This can be used for SPN and RVIN.

5.2 Alpha-Trimmed Mean-Based Approach

It is well known that simple mean might be inadequate in high-noise situations to represent the main body of the data but rather will be biased toward the "outliers" [43]. Since a noisy pixel is usually located near one of the two ends in the sorted sample [1], a robust statistical estimator, for example, *trimmed mean*, is more intuitively appealing than sample mean [43].

5.2.1 Impulse Noise Detection

Let I denote the corrupted, noisy image of size $l_1 \times l_2$, and x_{ij} is its pixel value at position (*i*, *j*), i.e., $I = \left\{ x_{ij} : 1 \le i \le l_1, 1 \le j \le l_2 \right\}$. Let $W_{ij}(I)$ denote the window of size $(2L_d + 1) \times (2L_d + 1)$ centered about x_{ij} , i.e., $W_{ij}(I) = \left\{ x_{i-u,j-v} \middle| -L_d \le u, v \le L_d \right\}$. The alpha-trimmed mean $M_{ij}(I)$ of the pixel values within window $W_{ij}(I)$ is defined as

$$M_{ij}(I) = \frac{1}{t - 2 * \lfloor \alpha t \rfloor} \sum_{i=\lfloor \alpha t \rfloor + 1}^{t - \lfloor \alpha t \rfloor} X_{(i)} \qquad (5.1)$$

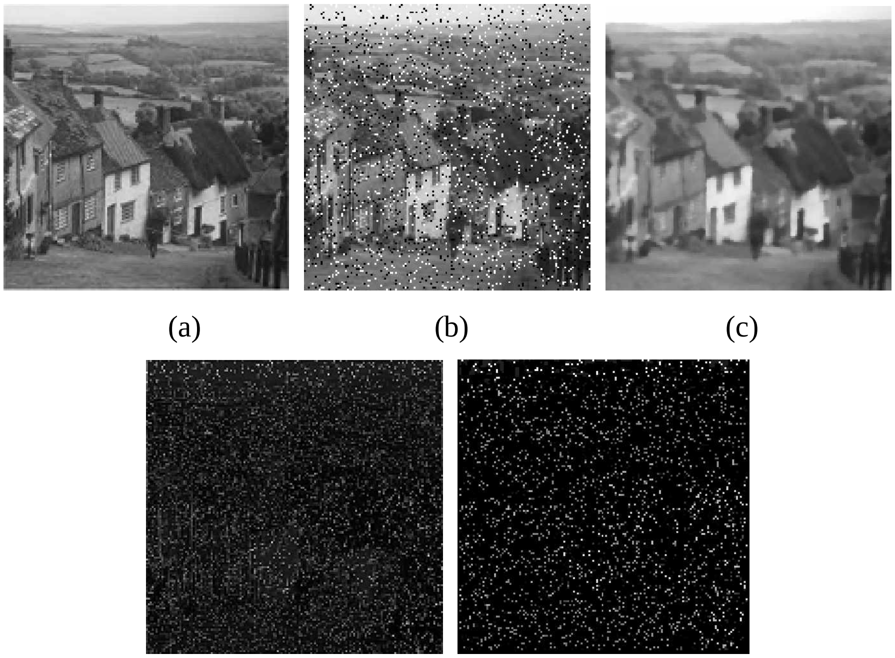
where $t = (2L_d + 1)^2, \alpha$ is the trimming parameter that assumes values between 0 and 0.5, $\lfloor . \rfloor$ is the floor function, and $X_{(i)}$ represents the *i*th data item in the increasingly ordered samples of $W_{ij}(I)$, i.e., $X_{(1)} \le X_{(2)} \le ...... \le X_{(t)}$ [43]. That is,

$$X_{(i)} = i\text{th smallest}\,(\,W_{ij}(I)\,)\,.$$

Since the alpha-trimmed mean $M_{ij}(I)$, with appropriately chosen α , represents approximately the average of the noise-free pixel values within window $W_{ij}(I)$, the absolute difference between x_{ij} and $M_{ij}(I)$

$$r_{ij} = \left| x_{ij} - M_{ij}(I) \right| \tag{5.2}$$

should be relatively large for noisy pixels and small for good, noise-free pixels. Let $R^{(0)} = \left\{ r_{ij} : 1 \le i \le l_1, 1 \le j \le l_2 \right\}$, and it is called the *residue* image of I . The residue image $R^{(0)}$ for image Goldhill of size 512×512 in Fig. 5.1(d), along with the original image [see Fig. 5.1(a)] of Goldhill and the noisy image [see Fig. 5.1(b)] of Goldhill corrupted by 10% fixed valued impulse noise, where the impulses take on the values of 0 or 255 with equal probabilities. For simplicity, in the experiment, we set $L_d = 1$ and α = 0.35. From Fig. 5.1(d), it is clear that the residue image $R^{(0)}$ contains not only noisy pixels but also some image details such as the sketches of the houses.



(a) (b) (c)

(d) (e)

Fig.5.1. Residue images for *Goldhill*. (a) Original image *Goldhill*. (b) *Goldhill* corrupted by 10% fixed-valued impulse noise.(c) Alpha trimed mean out put image (d) $R^{(0)}$. (e) $R^{(1)}$.

If we compare r_{ij} with a threshold value to determine whether or not the pixel x_{ij} is noisy, then it will detect many image details as noisy as well. In order to improve the impulse noise detection accuracy, ideally, we need to remove all the image details from the residue image $R^{(0)}$. Next, we implement a simple and efficient method to remove image details from $R^{(0)}$.

This method is based on the following observations. First, when the pixel x_{ij} is an impulse, it takes a value substantially larger than or smaller than those of its neighbors. Second, when the pixel x_{ij} is a noise-free pixel, which could belong to a flat region, an edge, or even a thin line, its value will be very similar to those of some of its neighbors. Therefore, we can detect image details from noisy pixels by counting the number of pixels whose values are similar to that of x_{ij} in its local window $W_{ij}(I)$. For $x_{i-u,j-v} \in W_{ij}(I)$ and, $(u,v) \neq (0,0)$, we define

$$\delta_{i-u,j-v} = \begin{cases} 1, & \left| x_{i-u,j-v} - x_{ij} \right| < T \\ 0, & otherwise \end{cases} \tag{5.3}$$

where T is a predetermined parameter. $\delta_{i-u,j-v} = 1$ indicates that the pixel $x_{i-u,j-v}$ is similar to the pixel in intensity. Also, let denote the total number of neighboring pixels in the window that are similar to the pixel x_{ij} in intensity.

That is

$$\xi_{ij} = \sum_{-L_d \le u,v \le L_d, (u,v) \neq (0,0)} \delta_{i-u,j-v} \tag{5.4}$$

Next, define as

$$\varphi_{ij} = \begin{cases} 0, & \xi_{ij} \ge N \\ 1, & otherwise \end{cases} \tag{5.5}$$

where N is a predetermined parameter. $\varphi_{ij} = 0$ indicates that most likely x_{ij} is a noise-free pixel instead of an impulse because it has at least N similar neighboring pixels in the window $W_{ij}(I)$. Let $R^{(1)} = \{ r_{ij} * \varphi_{ij} : 1 \le i \le l_1, 1 \le j \le l_2 \}$.It is clear that

$$r_{ij} * \varphi_{ij} = \begin{cases} 0, & \varphi_{ij} = 0 \\ 1, & \varphi_{ij} = 1 \end{cases} \tag{5.6}$$

Therefore, $R^{(1)}$ retains the impulse noises in $R^{(0)}$ while removing most of the image details, as shown in Fig. 1(e). From Fig. 1(e), it is clear that most image details disappear while the noise remains, thus improving the impulse noise detection accuracy in the subsequent steps. This is indeed very important for the good performance of this algorithm, as will be shown later. Since impulse noises can be detected more accurately from the residue image, next, we apply a fuzzy impulse detection technique to give each pixel a fuzzy flag indicating how much it looks like an impulse. The more it looks like a damaged pixel, the more it is modified later. This technique is very efficient in removing impulse noise, especially random-valued impulse noise where the impulse values are uniformly distributed within [0, 255], as will be shown later. The following two-parameter membership function, similar to that in [46], is used to generate a fuzzy flag for each pixel in the noisy image:

$$n_{ij} = \begin{cases} 1, & r_{ij} * \varphi_{ij} \geq W_u \\ \dfrac{r_{ij} * \varphi_{ij} - W_l}{W_u - W_l}, & W_l \leq r_{ij} * \varphi_{ij} \quad < W_u \\ 0, & r_{ij*}\varphi_{ij} < W_l \end{cases} \tag{5.7}$$

Where W_l and W_u are two predetermined parameters. Can be used to measure how much the pixel is corrupted.

5.2.2 Refinement

It is well known that a noisy pixel is usually located near one of the two ends in the ordered samples of $W_{ij}(I)$. In other words, if a pixel x_{ij} is not located near one of the two ends in the ordered samples, then most likely, it is not a noisy pixel. Based on this observation, we can refine the fuzzy flag n_{ij} as follows:

$$n_{ij} = \begin{cases} 0, & if \;\; X_{(s)} < x_{ij} < X_{(t-s+1)} \\ n_{ij}, & otherwise \end{cases} \tag{5.8}$$

Where s is constant, and $1 \leq s \leq (t-1)/2$.

5.2.3 Impulse Noise Cancellation

After we calculate the membership function n_{ij} for each pixel x_{ij}, the pixel value of x_{ij} is replaced by a linear combination of the median $m_{ij}(I)$ of $W_{ij}(I)$, i.e., $m_{ij}(I) = median(W_{ij}(I))$, and its original value x_{ij} . That is

$$y_{ij} = n_{ij} \times m_{ij}(I) + (1 - n_{ij}) \times x_{ij} \tag{5.9}$$

where y_{ij} is the restored value of x_{ij} .

5.3 Experimental Results

In this section, this algorithm is evaluated and compared with many other existing techniques. Extensive experiments are conducted on a variety of standard gray-scale test images with distinctly different features and different sizes, including *Lena*, *Bridge*, and *Goldhill*. For simplicity, we set L_d=1 and α = 0.35 in our computer simulations. For fixed-valued impulse noise, s=1, *T*=30, *N*=4, W_l=20 and *Wu* =40, while for random-valued impulse noise s=2, *T*=12, *N*=4, W_l =5 and *Wu* =30. This algorithm is implemented recursively. That is, the modified pixel values are immediately used in process of the following pixels. In order to further improve the restoration results, algorithm is applied iteratively. Usually, the best restoration results can be obtained after two to four iterations. Peak signal-to-noise ratio (PSNR) is used to give quantitative performance measures as in [7]-[23].

Table5.1 comparative results in PSNR for image Lena of size 512x512 corrupted by 20% fixed-valued impulse noise and random-valued impulse noise, respectively

Algorithm	Fixed Value Impulse noise	Random Value Impulse noise
Median filter	28.57	29.76
CWM [7]	30.39	32.42
AMF [22]	30.57	31.18
TSM filter[11]	31.84	34.13
DBM filter [10]	35.12	31.66
Fuzzy filter [48]	30.75	28.66
SD-ROM [23]	35.70	33.37
LRC Method [13]	36.95	33.43
ACWMF [20]	36.54	34.98
This Method [45]	**37.45**	**35.22**

First, the performance of this method for impulse noise suppression is compared with those of many other well-known algorithms, which include the standard median filter of size 3×3, the center-weighted median (CWM) filter of size 3×3 [7], the median filter with adaptive length (AMF) [22], the median filter based on fuzzy rules (FM) [48], the SD-ROM approach [23], and the adaptive center-weighted median filter (ACWMF) in [20]. For the CWM filter, center weights are appropriately tuned to obtain better performance for different noisy images. Unless otherwise mentioned, the SD-ROM approach used inside training set with *M*=1296 , as defined in [23]. The test image used for this comparison is Lena of size 512×512, which is corrupted by both 20% fixed-valued and random-valued impulse noises, as in [23]. Table 5.1 lists the restoration results of different algorithms. From Table 5.1, it is clear that for both fixed-valued and random-valued impulse noises, this method provides significant improvement over all the other approaches.

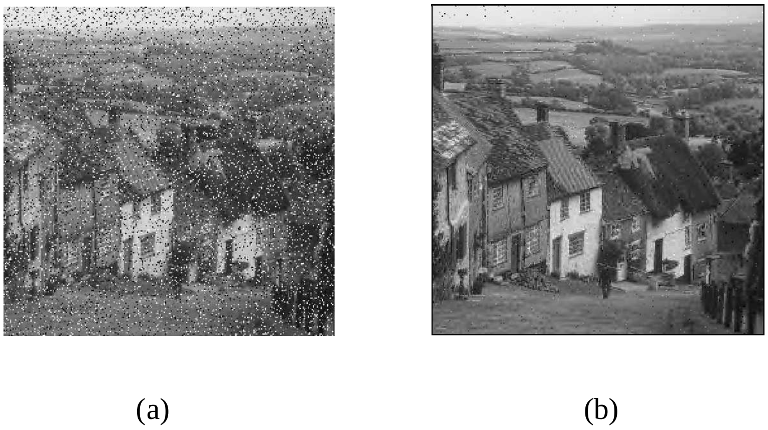

(a) (b)

Fig. 5.2. Goldhill (512 ×512) corrupted with random-valued impulse noise with 20% noise density in (a), and corresponding restored image in (b) .

Table 5.2 Comparative results in PSNR (dB) for standard images Lena, Bridge, and Goldhill of size (256 ×256) corrupted by 30% random-valued impulse noise.

Algorithm	**Lena**	**Bridge**	**Goldhill**
ACWMF [20]	27.18	22.21	26.57
DPVM [21]	27.29	22.44	27.13
This method	28.48	24.62	28.61

The restoration results of different algorithms are listed in Table 5.2, which also includes that of ACWMF in [20]. Table 5.2 shows clearly that this algorithm outperforms all other three techniques for Lena, Bridge, and Goldhill test images. In fig. 5.2 noisy images of Goldhill with noise density of 20% RVIN and corresponding restored images presented respectively. As a final remark, it should be mentioned that for all the experiments in this section, the parameters for this approach are fixed to show its robustness. Better performance could be achieved with more appropriately tuned parameters. In Fig. 5.3, noisy images of Lena with noise density of 10, 20, 30% and corresponding restored images presented respectively , which shows that the this algorithm yields superior subjective quality with respect to impulse noise suppression and image detail reservation.

5.4 Conclusion

In this chapter a detail-preserving algorithm is implemented for removing impulse noise efficiently from images. To demonstrate the superior performance of the method, extensive experiments have been conducted on a variety of standard test images to compare this method with many other well known techniques. Experimental results indicate that this method performs significantly better than many other existing techniques.

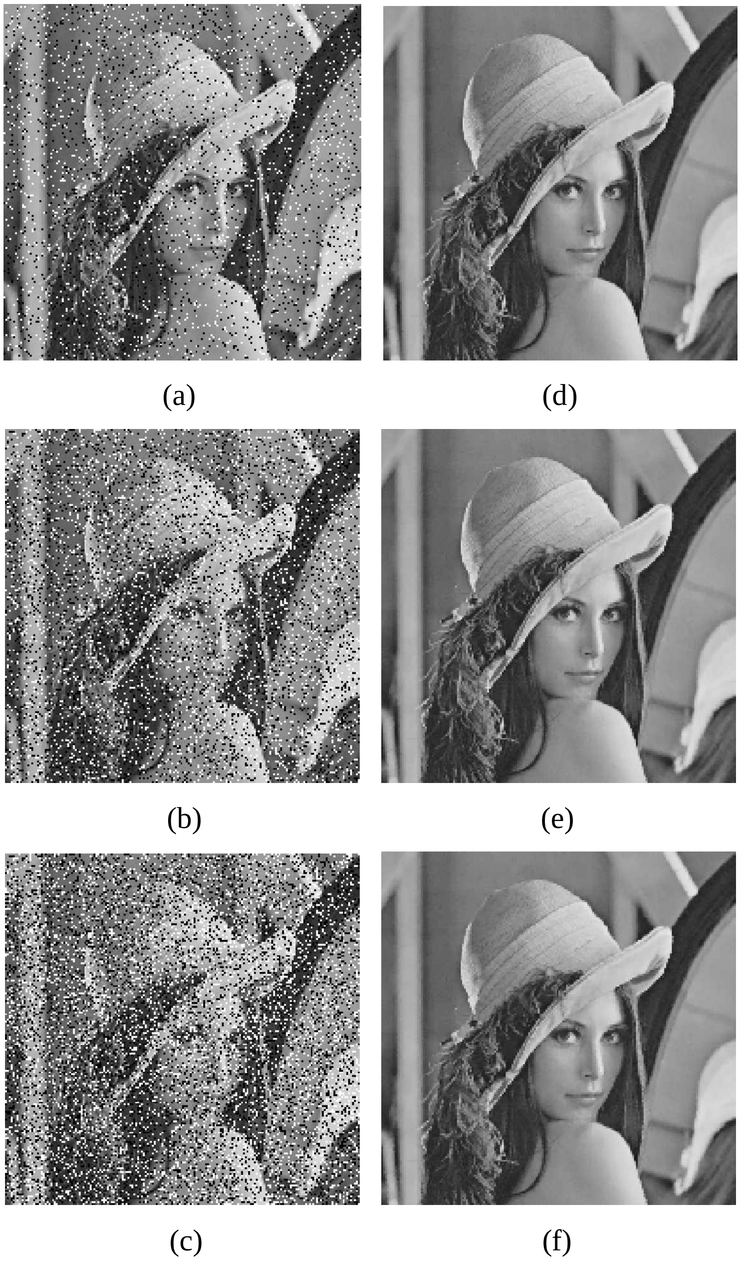


Fig. 5.3. Noisy images of Lena a, b, c with noise density of 10, 20, 30% and corresponding restored images d, e, f respectively

Chapter 6

AN IMPULSE DETECTOR FOR SWITCHING MEDIAN FILTERS

6.1 Introduction

In this chapter an impulse noise detection technique for switching median filters [13] is implemented, which is based on the minimum absolute value of four convolutions obtained using one-dimensional Laplacian operators. And after noise detection switching median filter has used. We have also proposed adaptive noise filtering technique at filtering end.Extensive simulations show that this proposed filter provides better performance than SM filter [13] .In particular, it can successfully preserve thin lines and other detail features.

6.2 Impulse Noise Detection

The impulse detection is usually based on the following two assumptions: 1) a noise-free image consists of locally smoothly varying areas separated by edges and 2) a noise pixel takes a gray value substantially larger or smaller than those of its neighbors. Let x_{ij} and y_{ij} represent the pixel values at position in the corrupted and restored images, respectively. The standard median filter outputs the median value of the samples in the $(2N+1)\times(2N+1)$ window centered at x_{ij}, i.e.,

$$m_{ij} = median\{x_{i-N,j-N},.....x_{ij}......x_{i+N,j+N}\} \tag{6.1}$$

To judge whether x_{ij} is an impulse, the median-based impulse detector [3] measures and compares it with a predefined threshold T_1

$$\alpha_{ij} = \begin{cases} 1, & if\ |x_{ij} - m_{ij}| > T_1 \\ 0, & otherwise \end{cases} \tag{6.2}$$

α_{ij} =1means x_{ij} is a corrupted pixel; otherwise x_{ij} is noise-free. The output of the SM filter is obtained by

$$y_{ij} = \alpha_{ij} \times m_{ij} + (1-\alpha_{ij}) \times x_{ij} \tag{6.3}$$

It is well known that the median filter cannot distinguish thin lines from impulses. Accordingly, the median-based impulse detector will interpret thin lines as impulses and lead to the removal of thin lines from images. Here a simple impulse detector is implemented to overcome this problem.

The input image is first convolved with a set of convolution kernels. Here, four one-dimensional Laplacian operators as shown in Fig. 6.1 are used, each of which is sensitive to edges in a different orientation. Then, the minimum absolute value of these four convolutions (denoted as r_{ij}) is used for impulse detection, which can be represented as

$$r_{ij} = \min\{|x_{ij} \otimes K_p| : p = 1\, to\, 4\} \qquad (6.4)$$

Where K_p is the p^{th} kernel, and $\otimes$ denotes a convolution operation.

The value of r_{ij} detects impulses due to the following reasons.

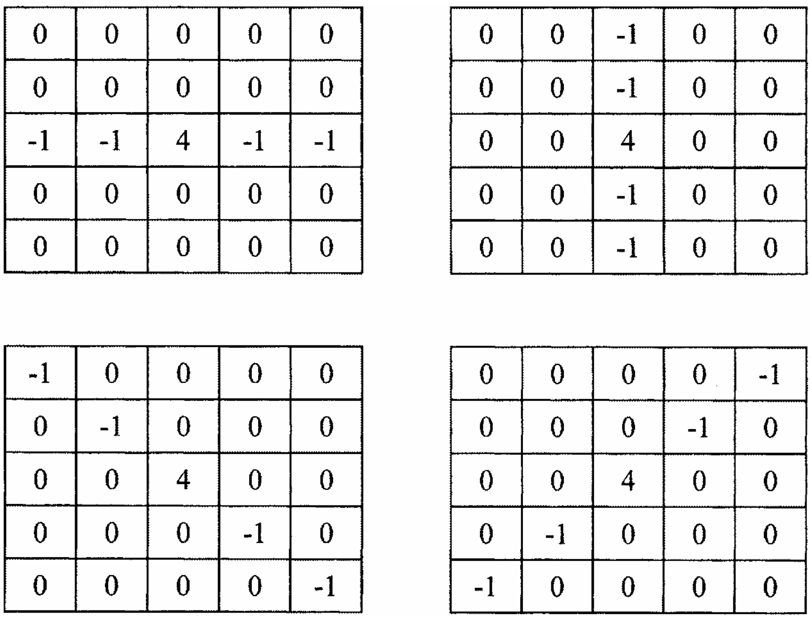
1) r_{ij} is large when the current pixel is an isolated impulse because the four convolutions are large and almost the same.

2) r_{ij} is small when the current pixel is a noise-free flat region pixel because the four convolutions are close to zero.

3) r_{ij} is small also when the current pixel is an edge (including thin line) pixel because one of the convolutions is very small (close to zero) although the other three might be large.

From the above analysis, r_{ij} is large when is corrupted by noise, and r_{ij} is small when is noise-free whether or not it is a flat-region, edge, or thin-line pixel. So, we can compare r_{ij} with a threshold to determine whether a pixel is corrupted, i.e.,

$$\alpha_{ij} = \begin{cases} 1, & r_{ij} > T \\ 0, & r_{ij} \le T \end{cases} \qquad (6.5)$$



0	0	0	0	0
0	0	0	0	0
-1	-1	4	-1	-1
0	0	0	0	0
0	0	0	0	0

0	0	-1	0	0
0	0	-1	0	0
0	0	4	0	0
0	0	-1	0	0
0	0	-1	0	0

-1	0	0	0	0
0	-1	0	0	0
0	0	4	0	0
0	0	0	-1	0
0	0	0	0	-1

0	0	0	0	-1
0	0	0	-1	0
0	0	4	0	0
0	-1	0	0	0
-1	0	0	0	0

Fig.6.1. Four 5×5 convolution kernels

Obviously, the threshold affects the performance of impulse detection. It is not easy to derive an optimal threshold through analytical formulation. But we can determine a reasonable threshold using computer simulations.

6.3 Switching Based Noise Filtering

After completion of noise detection process we have binary map α_{ij}. It gives information about whether corresponding pixel is corrupted in x_{ij} or not. If it is corrupted then it will replace by median of neighborhood pixels otherwise it will remain same. This step can be formulated by

$$y_{ij} = \alpha_{ij} \times m_{ij} + (1 - \alpha_{ij}) \times x_{ij} \qquad (6.6)$$

Where y_{ij} is restored pixel corresponding of x_{ij}

6.4 Modification in Filtering Process

After getting binary map α_{ij} we will apply noise adaptive filtering [41] technique at the filtering end. From the simulation we found its give much better than this method. The maximum filtering window size shouldn't be more than 7×7 to reduce blurring effect. Steps are given below for Adaptive Switching Filtering:

1. Start with (3×3) filtering window form x_{ij} and corresponding (3×3) window from binary map α_{ij}.
2. Find out how many pixels are detected as noise-free in current filtering window from corresponding binary flag window.
3. Iteratively extends window size outward by one pixel in all the four sides of the window, if the number of uncorrupted pixels is less than half of the total number of pixels (denoted by S_{in}=1/2[3×3]) within the filtering window .These all above three steps should be repeat again up to 7×7 filtering window if condition are not satisfy.
4. So since the current pixel has been marked noisy, it will not participate in filtering process. Only the pixels that are classified as noise free in filtering window will participate in median filtering process. This will, in turn, yield a better filtering result with less distortion.

6.5 Simulation Results

Computer simulations are carried out to assess the performance of this impulse noise detector using a variety of test images. Mean square error (MSE) is used to give a quantitative evaluation on the filtering results. The performance of the proposed filer is compared with SM(5x5),SM(7x7)[13] because detection process are same as in SM [13] but filtering process are noise adaptive[41] .Here some parameter has taken from SM[13],that is T=0.4 . MSE plot is given in fig.6.5 for SM(5x5),SM(7x7)[13] and Proposed modified SM for Lena image corrupted by different noise density(10-40%). Similarly MSE performance shown in fig 6.4 for Boat image. Fig.6.2, 6.3 shows the subjective visual qualities of the noisy and filtered images using proposed method for Lena and Boat image respectively with various noise density levels. From the experimental result we can say that proposed modified SM is giving better MSE performance and also good restored image quality.

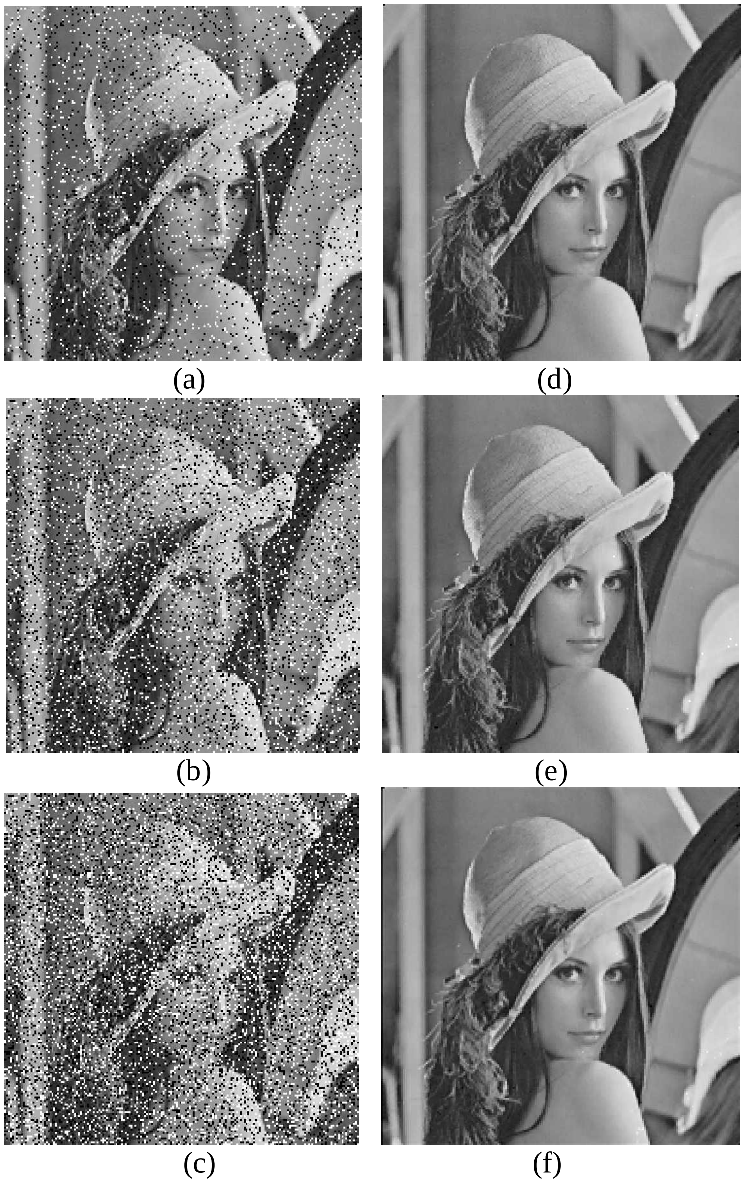
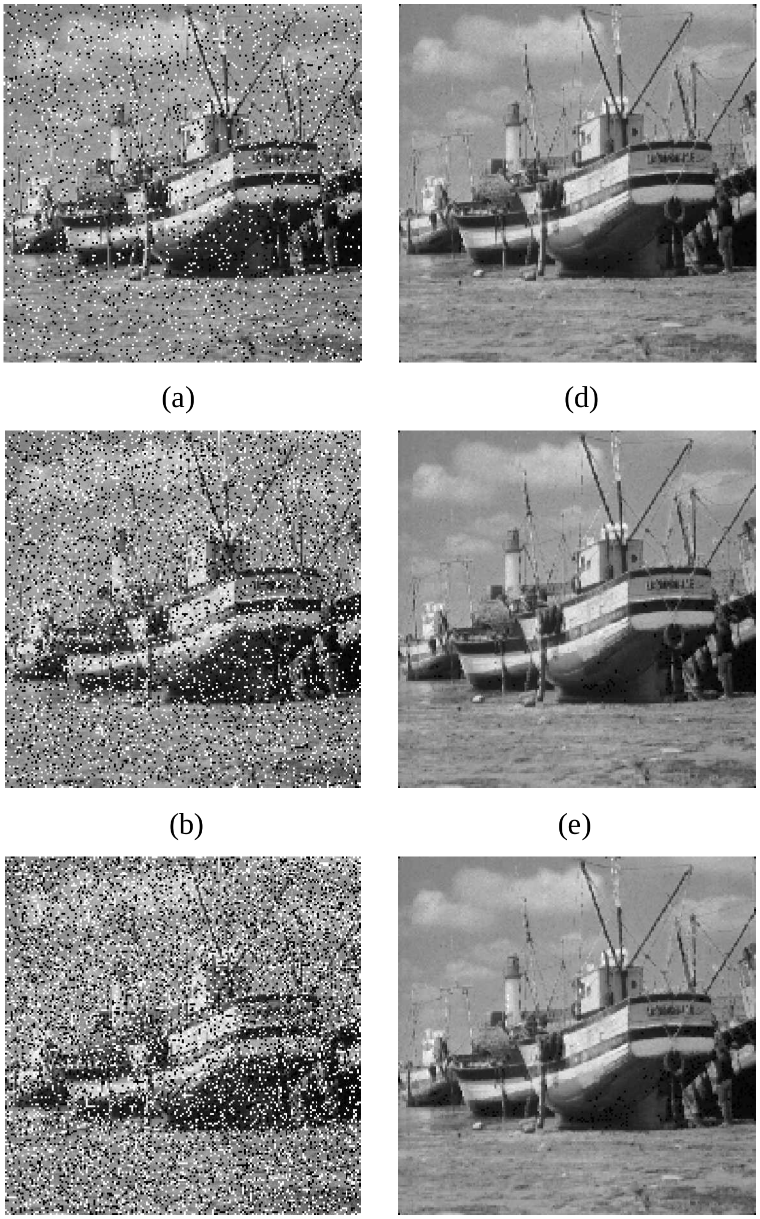


Fig. 6.2. a, b, c are noisy images of Lena (512×512) corrupted by salt and pepper noise with noise density of 10, 20, 30 respectively and corresponding restored image by proposed method are in d, e, f.

Simple median filter suppresses the impulses but introduces a blurring effect. The median-based and ROM based switching filters provide better results, but they remove some image details, especially thin lines. On the other hand, the WM-based and tri-state median filters can preserve image details, but many impulses remain in the image. It is seen that the proposed filter can remove most of the noise pixels while preserving image detail very well.



(a) (d)

(b) (e)

(c) (f)

Fig.6.3. a, b, c are noisy images of Boat (512×512) corrupted by salt and pepper noise with noise density of 10, 20, 30 respectively and corresponding restored image by proposed method are in d, e, f.

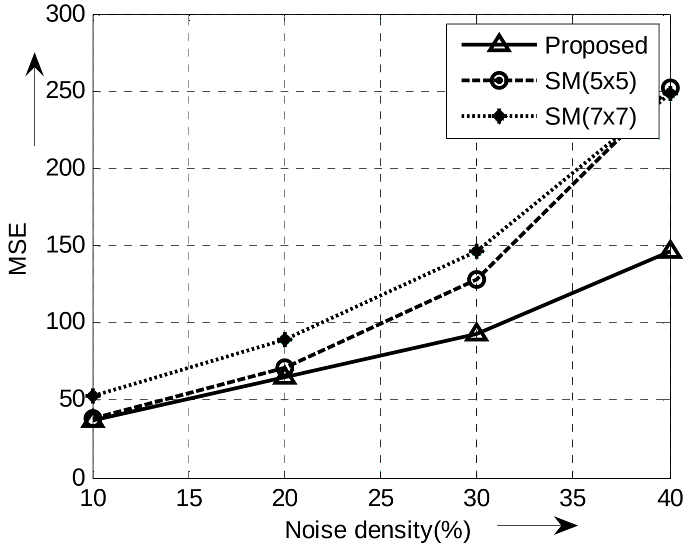


Fig. 6.4 : MSE Plot for Boat image

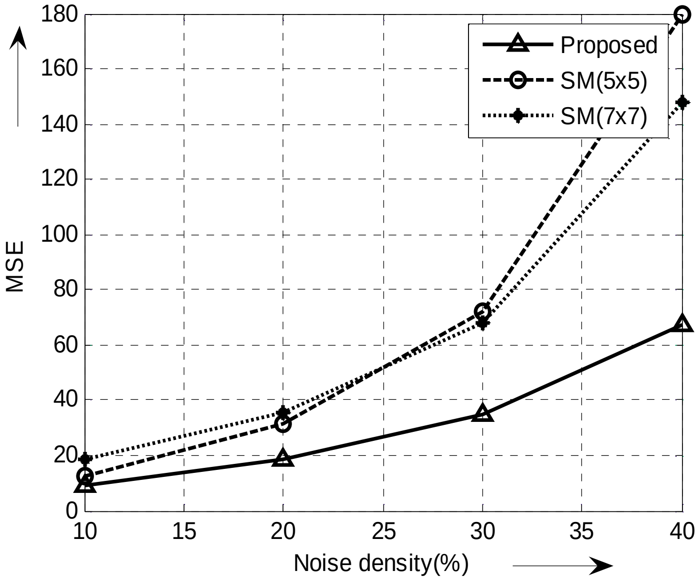


Fig. 6.5. MSE Plot for Lena image

6.6 Conclusion

We have proposed a modified SM [P3]that can effectively separate noise and noise-free pixels and also suppress the noise. In particular, it prevents the removal of fine details such as thin lines from the images and thus provides improved impulse detection ability. The simulation result shows that the proposed method is better than SM filters [13]. The MSE curves in Fig. 6.4 show that MSE increases significantly when the noise becomes heavy (after approximately 30%) for SM [13] filtering algorithm but modified SM is giving good MSE performance even for higher noise density.

Chapter 7

ADAPTIVE NOISE DETECTION AND SUPPRESSION FILTER FOR IMPULSE NOISE

7.1 Introduction

A new switching based median filter with adaptive noise detection and suppression (**ANDS**) method is proposed to restore images corrupted by salt & pepper impulse noise. The proposed algorithm works well for suppressing impulse noise with noise ratios from 5 to 60% while preserving image details. The algorithm is based on the following two schemes : (1) Adaptive noise detection scheme and (2) Adaptive filtering scheme. We begin by introducing neighborhood differentiation preprocessing step to quantify the increments in each local neighborhood of the noisy image. A correlation map is then derived by adaptive thresholding and used to designate pixels as noisy or noise free. Finally, the noise is attenuated by estimating the values of the noisy pixels with a switching based median filter applied exclusively to those neighborhood pixels not labeled as noisy. The size of filtering window is adaptive in nature, and it depends on the number of noise-free pixels in current filtering window.

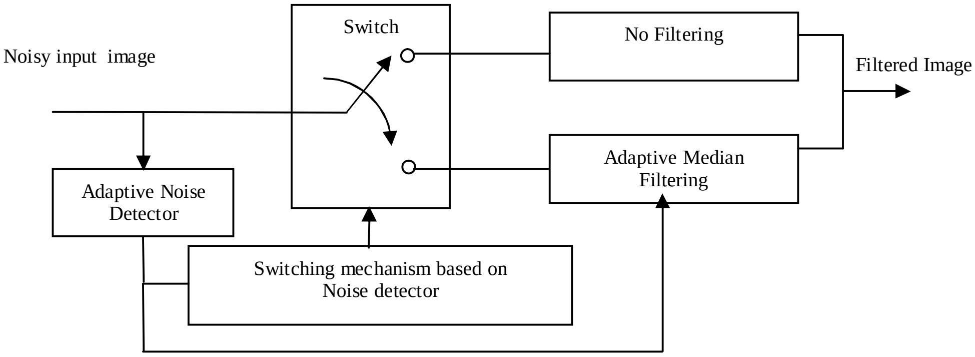


Fig. 7.1. Block diagram of **ANDS** Filter

7.2 Adaptive Noise Detection

ANDS is a nonlinear adaptive filtering algorithm consisting of two major components: corrupted pixel detection and adaptive spatially localized noise filtering, which are implemented in five processing steps as described in the following paragraphs.

Step 1: Neighborhood preprocessing.

Let us take a 3×3 window of image **A** (noisy image) center around A (i, j),that is $A_{i,j}$ =A(i + k, j +l) for $-1\leq(k, l)\leq+1$.The difference image for the window can be found according to

$D_{i,j}= A_{i,j} - A(i, j)$

Replace $D_{i,j}(2,2)=A(i, j)$

Example 1:

$$A_{i,j}=\begin{bmatrix}132 & 131 & 130\\ 131 & 132 & 129\\ 130 & 129 & 130\end{bmatrix}$$

Difference Image D is found to be

$$D_{i,j}=\begin{bmatrix}0 & -1 & -2\\ -1 & 132 & -3\\ -2 & -3 & -2\end{bmatrix}$$

Example 2:

$$A_{i,j}=\begin{bmatrix}132 & 131 & 130\\ 131 & 255 & 255\\ 130 & 129 & 130\end{bmatrix}$$

Difference Image D is

$$D_{i,j}=\begin{bmatrix}-123 & -124 & -125\\ -124 & 255 & 0\\ -125 & -126 & -125\end{bmatrix}$$

Step 2: Correlation map using adaptive thresholding

In this step correlation map to 8-neighborhood of $D_{i,j}(2, 2)$ is developed. Mapped image is formed according to the following rule-

$$Map_{i,j}(k,l)=\begin{cases}1, & D_{i,j}(k,l)<-\beta\\ 0, & -\beta\le D_{i,j}(k,l)\le\beta\\ 1, & D_{i,j}(k,l)>\beta\end{cases} \tag{7.1}$$

Where $1\le k\le 3$, and $1\le l\le 3$, $(k, l)\ne(2, 2)$. In (7.1), β [46] is a nonlinear adaptive threshold that will be designed as a function of A (i, j) based on the particular noise model. The thresholding parameter β is adaptive in nature so this is called correlation map by adaptive thresholding and this whole step becomes adaptive noise detection. β is given by

$$\beta=[41-0.00234(\,D_{i,j}(2,2)\text{ - }127.5\,)^2] \tag{7.2}$$

In case of salt and pepper noise maximum and minimum pixel values are 255, 0 respectively. When central pixel has maximum and minimum value then β value reaches to its minimum value. For Example-1 and Example- 2, β value will be 40.95 and 2.96 respectively.

So map image for Example 1 is

$$Map_{i,j} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 132 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and for Example 2 is

$$Map_{i,j} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 255 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

In Example 1, $Map_{i,j}(2,1) = 0$ indicates a “connectedness” condition in the sense that the observed pixel value $A(i, j{-}1)$ supports the hypothesis that $A(i, j)$ is noise free. For Example 2, $Map_{i,j}(2,1) = 1$ indicates that the observed value $A(i, j{-}1)$ supports the hypothesis that $A(i, j)$ has been corrupted by noise.

Step 3: Classification of pixel

Initially all pixel of **A** are labeled as noise-free pixels in a binary flag image ***B***, means all values are set to zeros initially. From the correlation map $Map_{i,j}$ central pixel will be classified as noisy or noise free, based on the number of zeros (Z) in the 8 neighborhood of $Map_{i,j}(2,2)$. If $Z \geq 3$ then current pixel A (i, j) is classified as a noise free and B(i, j)=0 otherwise B(i, j)=1. Z will be small when the noise density is high.

Step 4: Refinement

After classifying all pixels in **A** we have binary flag image **B**. Elements of **B** give information whether the pixel has been classified as noisy or noise-free. Since salt & pepper has minimum and maximum pixel values 0 and 255 respectively, so we will crosscheck the binary flag. If any pixel has classified as noisy but its value will be in the range of 10< A (i, j) <245, then corresponding flag will change from 1 to 0.

7.3 Adaptive Noise Filtering

The major contributions of making the entire switching median filter being *noise-adaptive* [14] come from the impulse-noise detection as described in the previous section. In order to determine the size of the filtering window, the limit of the maximum window size requires to be determined first. Based on the binary flag, "*no filtering*" is applied to those "uncorrupted" pixels, while the SM (switching median) with an adaptively determined window size is applied to each "corrupted" one.

The *maximum window size* is limited to (7×7) in order to avoid severe blurring of image details at high noise density cases (i.e., p>50%). Starting with (3×3) filtering window iteratively extends outward by one pixel in all the four sides of the window, provided that the number of uncorrupted pixels is less than half of the total number of pixels (denoted by S_{in}=1/2[3×3]) within the filtering window . Since the current pixel has been marked noisy, it will not participate in filtering process. Only the pixels that are classified as noise free in filtering window will participate in median filtering process. This will, in turn, yield a better filtering result with less distortion.

7.4 Simulation Result

Intensive simulations were carried out using several monochrome images, from which "Lena," "Peppers," and "Baboon" are chosen for demonstrations.

Table 7.1 PSNR Performance of Different Algorithms for Lena (512×512) image corrupted with salt and pepper noise

Algorithm	Lena Corrupted with Noise ratio		
	10%	20%	30%
MF(3×3)	31.19 dB	28.48 dB	25.45 dB
MF(5×5)	29.45 dB	28.91 dB	28.43 dB
MMEM [17]	30.28 dB	29.63 dB	29.05 dB
Florencio's[10]	33.69 dB	32.20 dB	30.95 dB
PMCWF [18]	35.70 dB	32.95 dB	31.86 dB
AMF(3×3)[22]	33.79 dB	30.65 dB	26.26 dB
AMF(5×5)[22]	30.11 dB	28.72 dB	27.84 dB
CMF(3×3)[21]	38.05 dB	31.79 dB	26.22 dB
CMF(5×5)[21]	36.32 dB	33.52 dB	30.33 dB
SDROM [23]	37.93 dB	34.10 dB	29.80 dB
CSAM [15]	39.23 dB	36.44 dB	34.32 dB
ACWMF [20]	40.60 dB	36.54 dB	33.68 dB
ANDS [P2]	**43.74 dB**	**39.64 dB**	**36.43 dB**

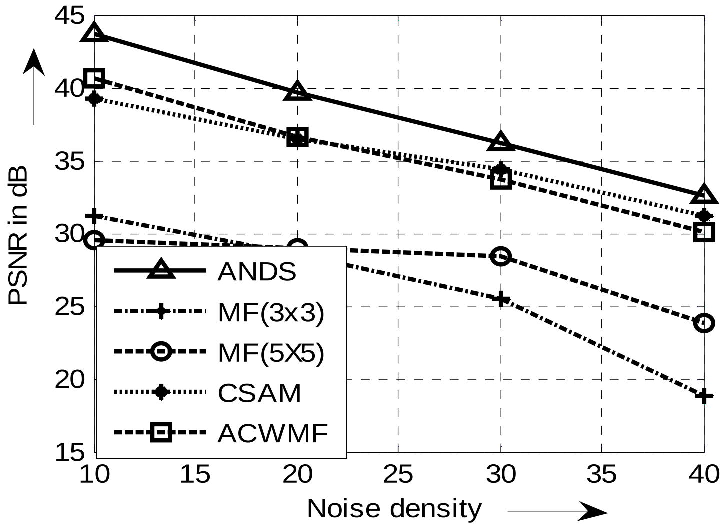


Fig.7.2. PSNR Plot for Lena image corrupted with
Different noise density

This proposed technique is compared with different standard methods that are standard median filters MF(3×3), MF(5×5), adaptive center weighted median filter (ACWMF) [20], minimum–maximum exclusive mean (MMEM) filter[17] , Florencio's[10], conditional median filtering (CMF)[21], Signal-dependent rank-order mean (SDROM) filter [23]. The proposed method ANDS has been applied on Lena, Baboon and Bridge gray images of size 512×512 corrupted by fixed-value impulse noise with different densities ranges from 5% to 60%. Comparatively PSNR performance has been given with different noise density in Table 7.1 for Lena image.

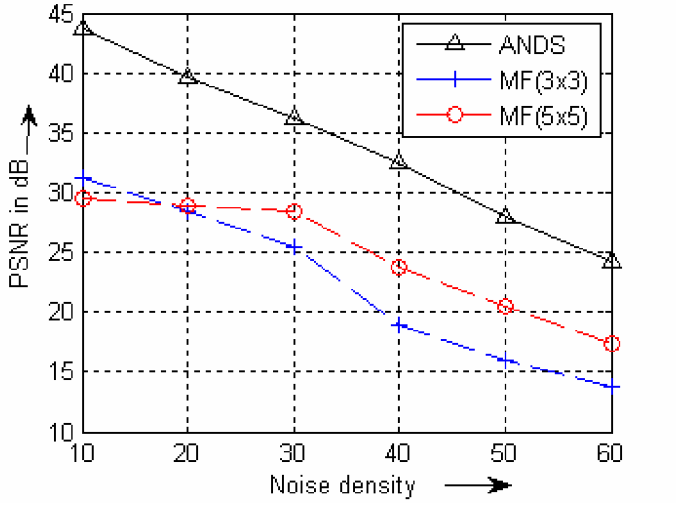
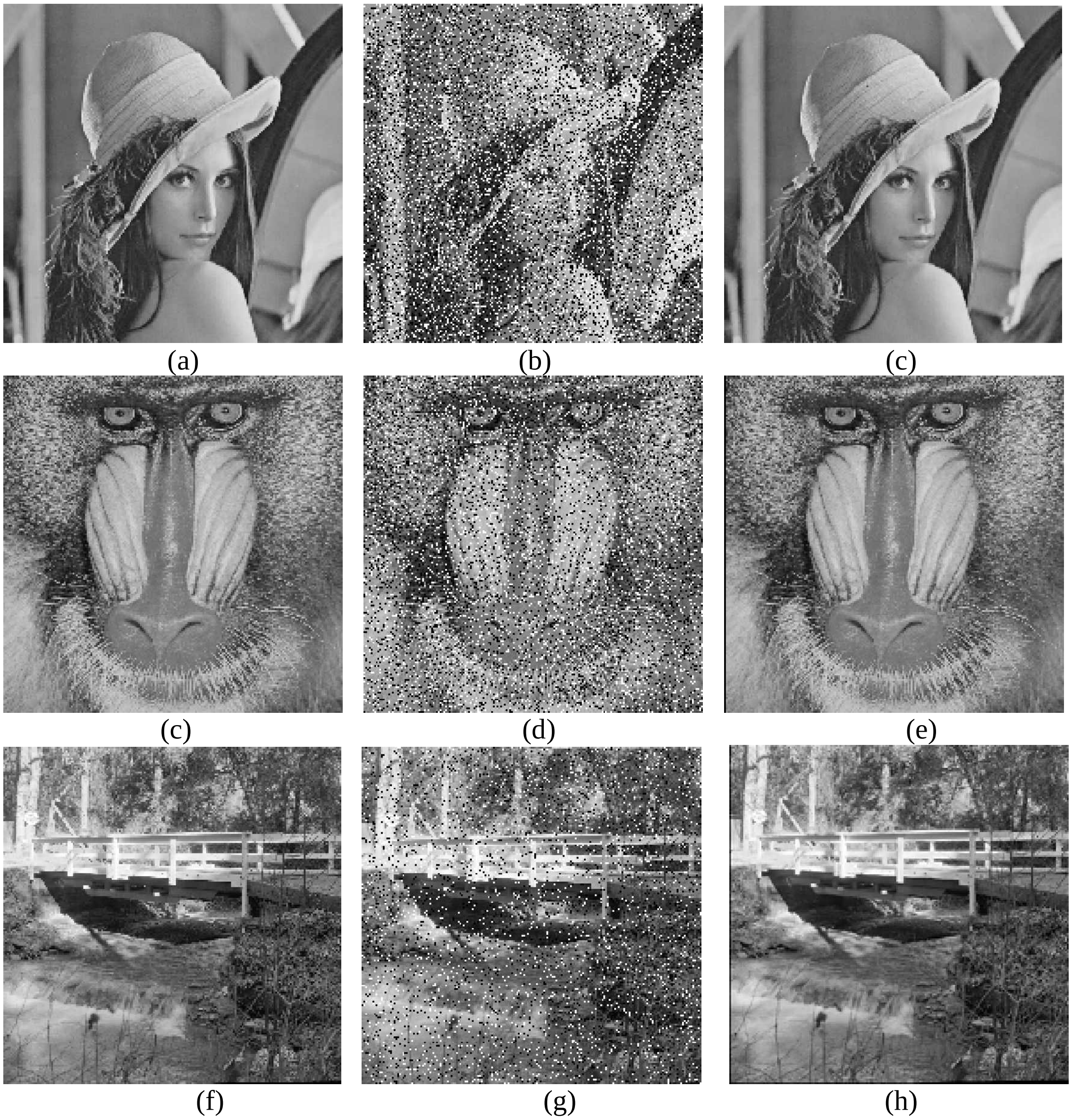


Fig.7.3. PSNR Plot for Lena image corrupted with 10 to 60%
noise density

This PSNR performance is also plotted in fig.7.2 and fig 7.3. In fig. 7.4 test images, noisy images and corresponding denoised images are shown. From all this simulation results we can say performance of the proposed ANDS scheme is better than other methods in terms of PSNR and visual aspect.


(a) (b) (c)

(c) (d) (e)

(f) (g) (h)

Fig.7.4 The original image (a)Lena, (c)Baboon, (f)Bridge and corresponding noisy image corrupted by 30%, 20%,10% fixed value impulse noise is (b),(d),(g)respectively and (c),(e),(h) are filtered image of (b),(d),(e) respectively.

7.5 Conclusion

In this chapter, a novel detail-preserving algorithm (ANDS) is simulated for removing impulse noise efficiently from images. To demonstrate the superior performance of the proposed method, extensive simulation experiments have been conducted on a variety of standard test images to compare our method with many other well known techniques.

Experimental results indicate that the proposed method performs significantly better than many other existing techniques.

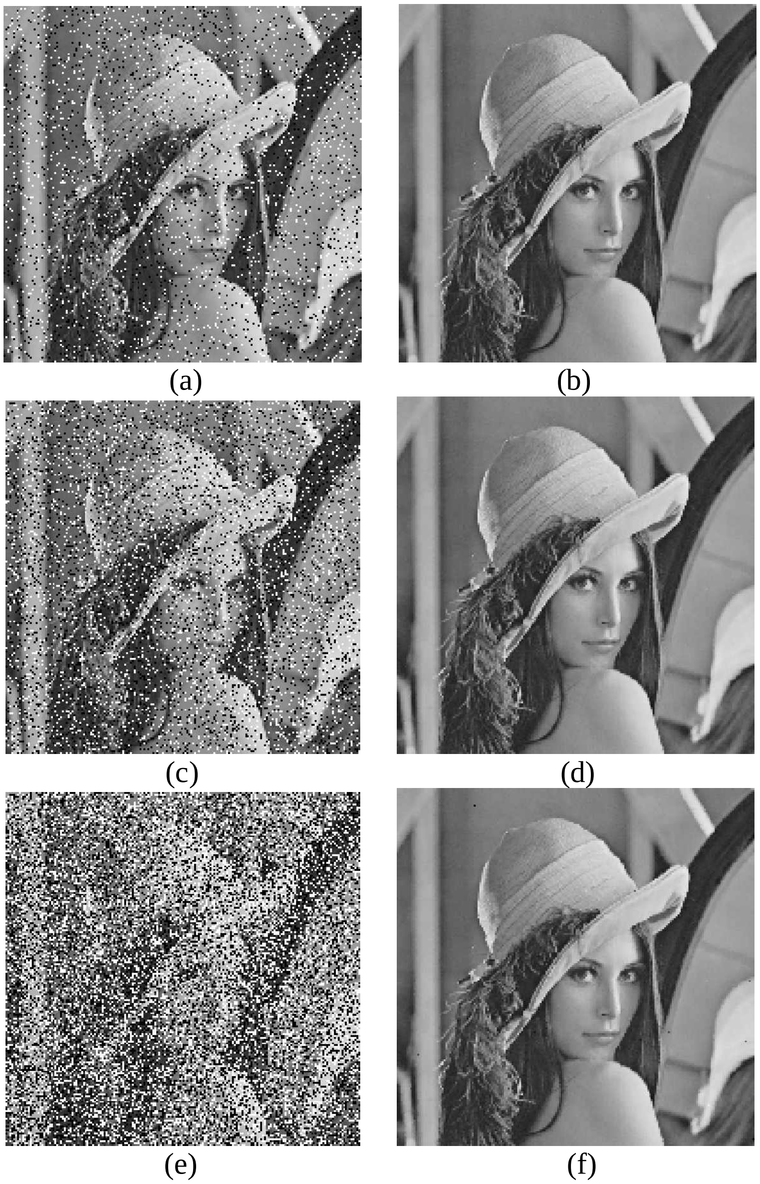


Fig.7.5 Lena image corrupted by salt and pepper noise with density of 10%, 20%,40% are in (a),(c),(e)respectively and corresponding filtered images by ANDS are in (b),(d),(f) respectively.

Chapter 8

IMPULSE NOISE DETECTION AND ADAPTIVE MEDIAN FILTER

8.1 Introduction

In this chapter a impulse noise detection & removal with adaptive filtering approach is proposed to restore images corrupted by salt & pepper noise. The proposed algorithm works well for suppressing impulse noise with noise density from 5 to 60% while preserving image details. The difference of current central pixel with median of local neighborhood pixels is used to classify the central pixel as noisy or noise-free. The noise is attenuated by estimating the values of the noisy pixels with a switching based median filter applied exclusively to those neighborhood pixels not labeled as noisy. The size of filtering window is adaptive in nature, and it depends on the number of noise-free pixels in current filtering window. Simulation results indicate that this filter is better able to preserve 2-D edge structures of the image and delivers better performance with less computational complexity as compared to other denoising algorithms existing in literature. The processing steps are shown in block diagram in Fig. 8.1.

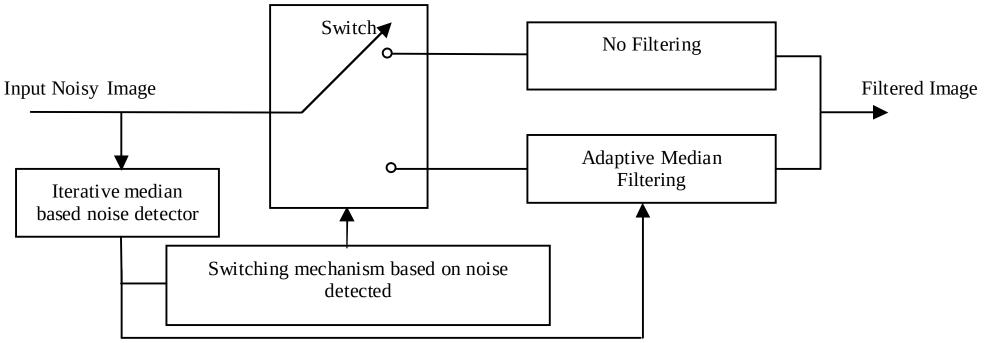

Switch
No Filtering
Input Noisy Image
Filtered Image
Adaptive Median Filtering
Iterative median based noise detector
Switching mechanism based on noise detected

Fig. 8.1.Block diagram of proposed Filter

8.2 Impulse Noise Detection

The impulse detection is usually based on the following two assumptions: 1) a noise-free image consists of locally smoothly varying areas separated by edges and 2) a noisy pixel has tendency of very high or very low gray value compare to its neighbors. Two image sequences are generated during the impulse detection procedure. The first is a sequence of gray scale images, $\{x^{(0)}_{(i,j)}, x^{(1)}_{(i,j)}, x^{(2)}_{(i,j)}, ...x^{(n)}_{(i,j)}\}$ where the initial image $x^{(0)}_{(i,j)}$ is noisy image itself , $(i\ ,\ j)$ is position of pixel in image, it can be $1\leq i \leq$ M, $1\leq j \leq$ N where M and N are the number of the pixel in horizontal and vertical direction respectively, and $x^{(n)}_{(i,j)}$ is image after n^{th} iteration. The second is a binary flag image sequence, $\{f^{(0)}_{(i,j)}, f^{(1)}_{(i,j)}, f^{(2)}_{(i,j)}, ...f^{(n)}_{(i,j)}\}$ where the binary flag $f^{(n)}_{(i,j)}$ is used to indicate whether the pixel at *(i, j)* in noisy image detected as noisy or noise-free after n^{th} iteration. If $f_{(i,j)}^{(n)}$ =0 means pixel at *(i, j)* has been found as noise-free after n^{th} iteration and if $f_{(i,j)}^{(n)}$ =1 means pixel at *(i, j)* has been found as noisy after n^{th} iteration. Before the first iteration, we assume that all the image pixels are good, i.e. $f_{(i,j)}^{(0)}$ =0 for all *(i, j)*.

Steps for noise detection:

1. Lets take a (2W+1)×(2W+1) window around $x_{(i,j)}^{(n-1)}$ means

   $x_{(i+k,j+l)}^{(n-1)}$ where -W$-W \le k \le W, -W \le l \le W$ k ≤ W, -W≤ l ≤ W and W≥1.

2. Find Median value of this window $m_{(i,j)}^{(n-1)}$

   $$m_{(i,j)}^{(n-1)} = \text{median}\ (x_{(i+k,j+l)}^{(n-1)}) \qquad (8.1)$$

3. Find absolute difference between $x_{(i,j)}^{(n-1)}$ and $m_{(i,j)}^{(n-1)}$,and assign

   $$f_{(i,j)}^{(n)} = \begin{cases} f_{(i,j)}^{(n-1)}, & if \left| x_{(i,j)}^{(n-1)} - m_{(i,j)}^{(n-1)} \right| < T \\ 1, & otherwise \end{cases} \qquad (8.2)$$

   Where *T* is predefined threshold value.1 indicate pixel detected as noisy after n^{th} iteration.

4. If $(i, j)^{th}$ is detected as noisy then the value of $x_{(i,j)}^{(n)}$ will be modified as

   $$x_{(i,j)}^{(n)} = \begin{cases} m_{(i,j)}^{(n-1)}, & if\ f_{(i,j)}^{(n)} \neq f_{(i,j)}^{(n-1)} \\ x_{(i,j)}^{(n-1)}, & if\ f_{(i,j)}^{(n)} = f_{(i,j)}^{(n-1)} \end{cases} \qquad (8.3)$$

This all steps will repeat for *t* times. This *t* can be 2,3,4…..After the t^{th} iteration we have two images $x_{(i,j)}^{(t)}$ and $f_{(i,j)}^{(t)}$ .But only $f_{(i,j)}^{(t)}$ binary flag image is required for noise filtering process. This median based noise detection had introduced by Wang and Zhang (PSM) [12] in progressive way. The difference between PSM and our approach is, we have applied adaptive filtering approach for improving filtering performance of the filter. From the simulation result we can see our approach is giving better performance in term of PSNR and visual aspect.

8.3 Adaptive Noise Filtering

From the last section we got binary flag image $f_{(i,j)}^{(t)}$ which elements give information about whether the pixel is corrupted or not corrupted at location (*i, j*) in noisy image $x_{(i,j)}^{(0)}$ .If (*i, j*)th pixel has detected as a noise then it will go through median filtering process other wise it will remain same. This is called Switching based median filter. Here the size of filtering window is adaptive in nature and its size is depending on the number of pixels which are noise free in current filtering window [41]. The maximum window size shouldn't be more than 7×7 to reduce blurring effect. Steps are given below for Adaptive Switching Filtering:

5. Start with (3×3) filtering window form $x_{(i,j)}^{(0)}$ and corresponding (3×3) window from binary flag image $f_{(i,j)}^{(t)}$.
6. Find out how many pixels are detected as noise-free in current filtering window from corresponding binary flag window.
7. Iteratively extends window size outward by one pixel in all the four sides of the window, if the number of uncorrupted pixels is less than half of the total number of pixels (denoted by S_{in}=1/2[3×3]) within the filtering window .These all above three steps should be repeat again if condition are not satisfy.
8. So since the current pixel has been marked noisy, it will not participate in filtering process. Only the pixels that are classified as noise free in filtering window will participate in median filtering process. This will, in turn, yield a better filtering result with less distortion.

8.4 Simulation Results

Intensive simulations were carried out using several monochrome images, from which "Lena," "Peppers," and "Bridge" are chosen for demonstrations. This proposed technique is compared with different standard methods that are standard median filters MF(3×3), MF(5×5), minimum–maximum exclusive mean (MMEM) filter[17] , Florencio's[10], conditional median filtering (CMF)[21], signal-dependent rank-order mean (SDROM) filter [23],progressive switching median filter [12]. The proposed method has been applied on Lena, Pepper and Bridge gray images of size 512×512 corrupted by fixed-value impulse noise with different densities ranges from 5% to 60%.For simulation we have taken T=40, and *t*=2. Comparatively PSNR performance has been given with different noise density in Table. 8.1 for Lena image. This PSNR performance is also plotted in fig.8.2.In fig. 8.3 test images, noisy images and corresponding denoised images are shown. For comparison with PSM [12] we have given MSE (mean square error) plot in fig.8.4 with different noise density for Bridge image. From all this simulation results we can say performance of the proposed method is better than other methods in terms of PSNR and visual aspect.

Algorithm	Lena Corrupted with Noise density		
	10%	20%	30%
MF(3×3)	31.19 dB	28.48 dB	25.45 dB
MF(5×5)	29.45 dB	28.91 dB	28.43 dB
MMEM [17]	30.28 dB	29.63 dB	29.05 dB
Florencio’s[10]	33.69 dB	32.20 dB	30.95 dB
PMCWF [18]	35.70 dB	32.95 dB	31.86 dB
AMF(3×3)[22]	33.79 dB	30.65 dB	26.26 dB
AMF(5×5)[22]	30.11 dB	28.72 dB	27.84 dB
CMF(3×3)[21]	38.05 dB	31.79 dB	26.22 dB
CMF(5×5)[21]	36.32 dB	33.52 dB	30.33 dB
SDROM [23]	37.93 dB	34.10 dB	29.80 dB
Proposed [P1]	**42.14 dB**	**38.66 dB**	**35.75 dB**

Table 8.1. PSNR Performance of Different Algorithms for Lena image corrupted with salt and pepper noise

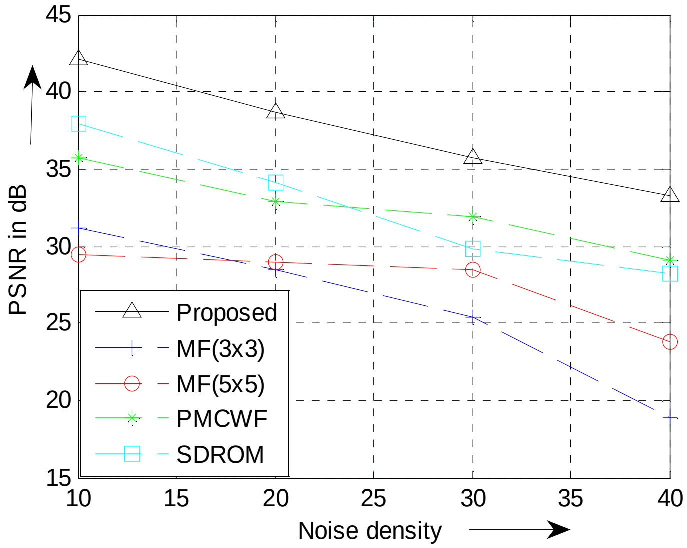
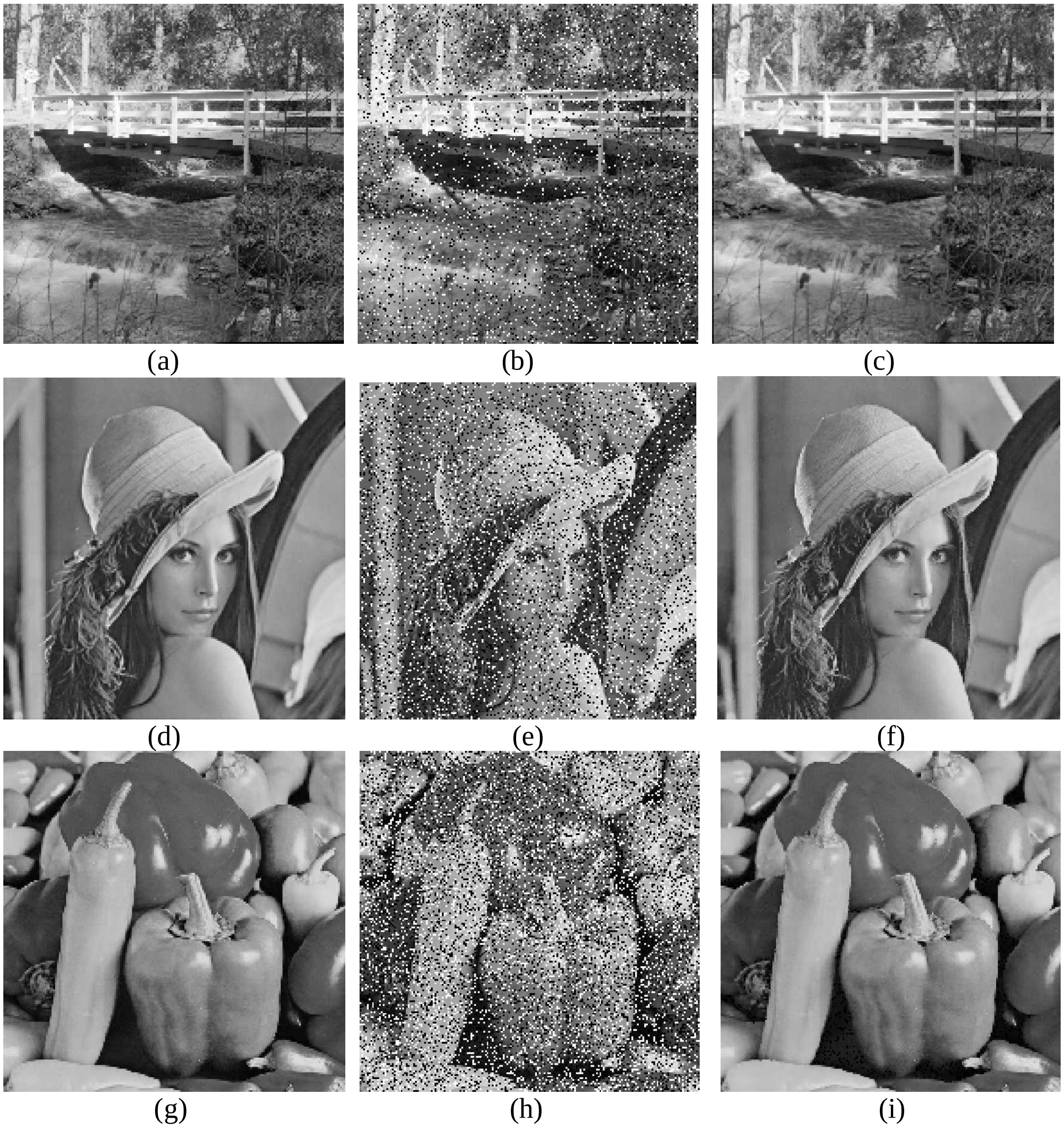


Fig.8.2. PSNR Plot for Lena image corrupted with different noise density



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig.8.3. The original image (a)Bridge, (d)Lena, (g)Pepper and corresponding noisy image corrupted by 10%, 20%, 30% fixed value impulse noise is (b),(e),(h) respectively and (c),(f),(i) are filtered image of (b),(e),(h) respectively.

8.5 Conclusion

To demonstrate the performance of the proposed method, extensive simulation experiments have been conducted on a variety of standard test images to compare our method with many other well known techniques. Experimental results indicate that the proposed method performs significantly better than many other existing techniques.

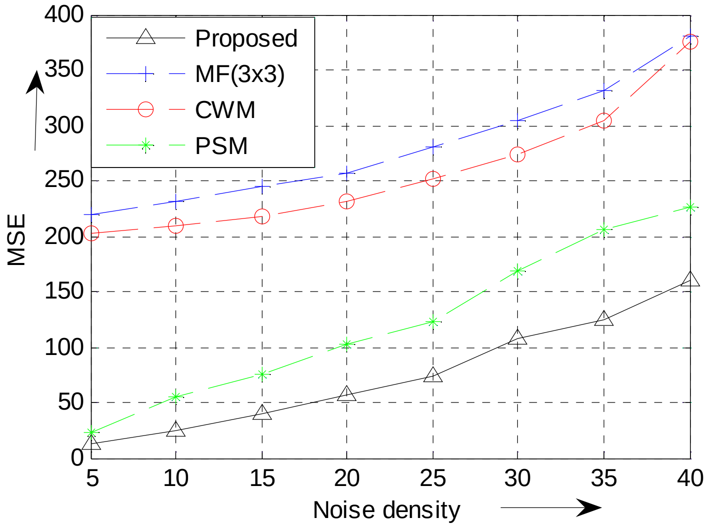


Fig.8.4. MSE Plot for Bridge image corrupted with different noise density

Chapter 9

DISCUSSION AND CONCLUSION

9.1 Comparative Study

In this section the performance of the proposed techniques has been compared with existing methods available in literature. Table 9.1 shows the PSNR performance of different existing and also proposed method for Lena image. The extensive simulation has carried out in this thesis work that shows the performance of the proposed filter gives better visual quality and objective quantity in term of PSNR. This PSNR performance also reported in graph (fig. 9.1).

Table 9.1 PSNR Performance of Different Algorithms for Lena (512×512) image corrupted with salt and pepper noise

Algorithm	Lena Corrupted with Noise ratio		
	10%	20%	30%
MF(3×3)	31.19 dB	28.48 dB	25.45 dB
MF(5×5)	29.45 dB	28.91 dB	28.43 dB
MMEM [17]	30.28 dB	29.63 dB	29.05 dB
Florencio’s[10]	33.69 dB	32.20 dB	30.95 dB
PMCWF [18]	35.70 dB	32.95 dB	31.86 dB
AMF(3×3)[22]	33.79 dB	30.65 dB	26.26 dB
AMF(5×5)[22]	30.11 dB	28.72 dB	27.84 dB
CMF(3×3)[21]	38.05 dB	31.79 dB	26.22 dB
CMF(5×5)[21]	36.32 dB	33.52 dB	30.33 dB
SDROM [23]	37.93 dB	34.10 dB	29.80 dB
SM(7×7)[13]	35.48 dB	32.49 dB	29.67 dB
CSAM [15]	39.23 dB	36.44 dB	34.32 dB
ACWMF [20]	40.60 dB	36.54 dB	33.68 dB
PSM[12]	39.34 dB	35.53 dB	34.13 dB
BDND [43]	42.08 dB	38.84 dB	36.20 dB
Proposed 1 [P3]	**38.35 dB**	**35.39 dB**	**32.71 dB**
Proposed 2 [P1]	**42.14 dB**	**38.66 dB**	**35.75 dB**
ANDS [P2]	**43.74 dB**	**39.64 dB**	**36.43 dB**

Performance of ACWMF is good for low noise density but poor for medium and high noise density. BDND gives good performance for low, medium, and high noise density. Proposed method [P1] also gives improved performance for low and medium noise density. ANDS [P2] filter gives appreciable performance for low and medium impulse noise density.

9.2 Conclusion

Many existing methods are studied and the BDND algorithm is found to be the best among them for removal of impulsive noise. In addition, some novel adaptive filtering schemes have been proposed for suppression of impulse noise from digital images. The performance of the proposed method: ANDS is found to be the best in terms of PSNR. Its performance in terms of visual quality is also found to be very nice for noise density upto 20 %. Since this algorithm outperforms almost all schemes existing in the literature, it is finally concluded that this filtering scheme be recommended for removal of low and medium density salt and pepper noise from digital images.

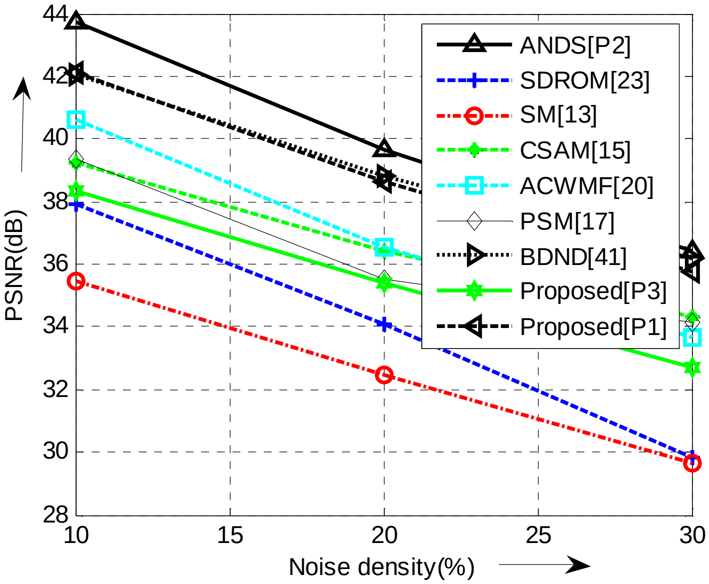


Fig.9.1 Comparative PSNR performance for Lena (512×512).

9.3 Future Scope

In this thesis work, many existing method are simulated and nonlinear adaptive methods have proposed for suppression of impulsive noise. The performance of proposed filter also can be improved by applying adaptive filtering technique through out the image in recursive way. Neuro-Fuzzy can be used for noise detection for switching based median filter.

REFERENCES:

[1] J. Astola and P. Kuosmanen, Fundamental of Nonlinear Filtering, Boca Raton, CRC Press, 1997.

[2] I. Pitas and A. N. Venetsanopoulos, Nonlinear Digital Filters: Principles and Applications, Boston, Kluwer, 1990.

[3] D. R. K. Brownrigg, "The weighted median filter," Comm. ACM, vol. 27, pp. 807-818, August 1984.

[4] B. I. Justusson, "Median filtering: Statistical properties," Two-Dimensional Digital Signal Processing II, T. S. Huang Ed., New York; Springer Verlag, 1981.

[5] T. Loupos, W. N. McDicken and P. L. Allan, "An adaptative weighted median filter for speckle suppression in medical ultrasonic images," IEEE Trans. Circuits Syst., vol. 36, Jan. 1989.

[6] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: A tutorial," IEEE Trans. Circuits and Syst. II : Analog and Digital Signal Processing, vol. 43, pp. 157-192, Mar. 1996.

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

[8] A. C. Bovik, T. Huang and D. C. Munson, "A generalization of median filtering using linear combinations of order statistics," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, pp. 1342-1350, Jun. 1983

[9] T. Sun and Y. Neuvo, "Detail-preserving median based filters in image processing," *Pattern Recognit. Lett.*, vol. 15, no. 4, pp. 341–347, Apr. 1994.

[10] D. A. Florencio and R. W. Schafer, "Decision-based median filter using local signal statistics," in *Proc. SPIE Vis. Commun. Image Process.*, vol. 2308, Sep. 1994, pp. 268–275.

[11] T. Chen, K.-K. Ma, and L.-H. Chen, "Tri-state median filter for image denoising," *IEEE Trans. Image Process.*, vol. 8, no. 12, pp. 1834–1838, Dec. 1999.

[12] Z. Wang and D. Zhang, "Progressive switching median filter for the removal of impulse noise from highly corrupted images," *IEEE Trans. Circuits Syst. II*, vol. 46, no. 1, pp. 78–80, Jan. 1999.

[13] S. Zhang and M. A. Karim, "A new impulse detector for switching median filters," *IEEE Signal Process. Lett.*, vol. 9, no. 4, pp. 360–363, Nov. 2002.

[14] H.-L. Eng and K.-K. Ma, "Noise adaptive soft-switching median filter," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 242–251, Feb. 2001.

[15] 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.

[16] H. Hwang and R. A. Haddad, “Adaptive median filters: New algorithms and results,” *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 499–502,Apr. 1995.

[17] W.-Y. Han and J.-C. Lin, “Minimum–maximum exclusive mean (MMEM) filter to remove impulse noise from highly corrupted images,” *Electron. Lett.*, vol. 33, no. 2, pp. 124–125, 1997.

[18] J.-H. Wang, “Prescanned minmax center-weighted filters for image restoration,” *Proc. Inst. Elect. Eng.*, vol. 146, no. 2, pp. 101–107, 1999.

[19] C.-T. Chen and L.-G. Chen, “A self-adjusting weighted median filter for removing impulse noise in image,” in *Proc. IEEE Int. Conf. on Image Processing*, 1998, pp. 419–422.

[20] T. Chen and H. R. Wu, “Adaptive impulse detection using center-weighted median filters,” *IEEE Signal Processing Lett.*, vol. 8, pp. 1-3, Jan. 2001.

[21] T. Kasparis, N. S. Tzannes, and Q. Chen, “Detail-preserving adaptive conditional median filters,” *J. Electron. Imag.*, vol. 1, no. 14, pp. 358–364, 1992.

[22] A. Sawant, H. Zeman, D. Muratore, S. Samant, and F. DiBianka, “An adaptive median filter algorithm to remove impulse noise in X-ray and CT images and speckle in ultrasound images,” *Proc. SPIE*, vol. 3661,pp. 1263–1274, Feb. 1999.

[23] E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, “A new efficient approach for the removal of impulse noise from highly corrupted images,” *IEEE Trans. Image Process.*, vol. 5, no. 6, pp. 1012–1025, Jun.1996.

[24] M. S. Moore, M. Gabbouj, and S. K. Mitra, “Vector SD-ROM filter for removal of impulse noise from color images,” presented at the PECMCS EURASIP Conf. DSP for Multimedia Communications and Services, Poland, 1999.

[25] P. S. Windyga, “Fast impulse noise removal,” *IEEE Trans. Image Process.*, vol. 10, no. 1, pp. 173–179, Jan. 2001.

[26] N. Alajlan, M. Kamel, and E. Jernigan, “Detail preserving impulsive noise removal,” *Signal Process. Image Commun.*, vol. 19, pp. 993–1003, 2004.

[27] X. Xu, E. L. Miller, D. Chen, and M. Sarhadi, “Adaptive two-pass rank order filter to remove impulse noise in highly corrupted images,” *IEEE Trans. Image Process.*, vol. 13, no. 2, pp. 238–247, Feb. 2004.

[28] M. Nikolova, “A variational approach to remove outliers and impulse noise,” *J. Math. Imag. Vis.*, vol. 20, pp. 99–120, 2004.

[29] R. H. Chan, C. Hu, and M. Nikolova, “An iterative procedure for removing random-valued impulse noise,” *IEEE Signal Process. Lett.*, vol. 11, no. 12, pp. 921–924, Dec. 2004.

[30] F. Russo, “Impulse noise cancellation in image data using a two-output nonlinear filters,” *Measurement*, vol. 36, pp. 205–213, 2004.

[31] I. Aizenberg, C. Butakoff, and D. Paliy, “Impulsive noise removal using threshold boolean filtering based on the impulse detecting functions,” *IEEE Signal Process. Lett.*, vol. 12, no. 1, pp. 63–66, Jan. 2005.

[32] J. H. Wang and L. D. Lin, “Image restoration using parametric adaptive fuzzy filter,” in *Proc. NAFIPS*, vol. 1, 1998, pp. 198–202.

[33] J. H.Wang and H. C. Chiu, “HAF: An adaptive fuzzy filter for restoring highly corrupted images by histogram estimation,” *Proc. Nat. Sci. Council*, vol. 23, no. 5, pp. 630–643, 1999.

[34] Z. Ma and H. R.Wu, “A histogram based adaptive vector filter for color image restoration,” in *Proc. 4th Pacific Rim Conf. Multimedia*, vol. 1, 2003, pp. 81–85.

[35] F. Russo and G. Ramponi, “A fuzzy filter for images corrupted by impulse noise,” *IEEE Signal Process. Lett.*, vol. 3, no. 6, pp. 168–170, Jun. 1996.

[36] D. Van De Ville, M. Nachtegael, D. Van der Weken, E. E. Kerre, W. Philips, and I. Lemahieu, “Noise reduction by fuzzy image filtering,” *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 4, pp. 429–436, Aug. 2003.

[37] H. Xu, G. Zhu, H. Peng, and D. Wang, “Adaptive fuzzy switching filter for images corrupted by impulse noise,” *Pattern Recognit. Lett.*, vol. 25, pp. 1657–1663, 2004.

[38] Y. S. Choi and R. Krishnapuram, “A robust approach to image enhancement based on fuzzy logic,” *IEEE Trans. Image Process.*, vol. 6, no. 6, pp. 808–825, Jun. 1997.

[39] M. E. Yüksel and A. Bas¸türk, “Efficient removal of impulse noise from highly corrupted digital images by a simple neuro-fuzzy operator,” *Int. J. Electron. Commun.*, vol. 57, no. 3, pp. 214–219, 2003.

[40] H. Xu, G. Zhu, H. Peng, and D. Wang, “Adaptive fuzzy switching filter for images corrupted by impulse noise,” *Pattern Recognit. Lett.*, vol. 25, pp. 1657–1663, 2004.

[41] Pei-Eng Ng and Kai-Kuang Ma, “A switching median filter with boundary discriminative noise detection for extremely corrupted images” *IEEE Trans. Image Processing*, vol. 15, no. 6, pp. 1506–1516, June 2006.

[42] H.-L. Eng and K.-K. Ma, “Noise adaptive soft-switching median filter,”*IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 242–251, Feb. 2001.

[43] J. B. Bednar and T. L.Watt, “Alpha-trimmed means and their relationship to median filters,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, no. 1, pp. 145–153, Feb. 1984.

[44] A. C. Bovik, T. S. Huang, and D. C. Munson, “A generalization of median filtering using linear combinations of order statistics,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-31, no. 6, pp. 1342–1350, Dec. 1983.

[45] W. Luo, “An efficient detail-preserving approach for removing impulse noise in image,” *IEEE Trans. Signal Process.*, vol. -13, no. 7, pp. 413–416, July 2006.

[46] Y. Wang, Linda S. DeBrunner, J. P. Havlicek, and D. Zhou, “Signal exclusive adaptive average filter for impulse noise suppression”, *2006 IEEE Southwest Symposium on Image Analysis and Interpretation, pp. 51–55, 2006.*

[47] D. Zhang and Z. Wang, “Impulse noise detection and removal using fuzzy techniques,” *Electron. Lett.*, vol. 33, no. 5, pp. 378–379, 1997.

[48] K. Arakawa, “Median filters based on fuzzy rules and its application to image restoration,” *Fuzzy Sets Syst.*, vol. 77, pp. 3–13, 1996.

PUBLICATIONS FROM THE THESIS

[P1] M. K. Gupta, S. Meher, “Impulse Noise Detection and Adaptive Median Filtering Method” Proceedings of National Conference on Trends in Computing (NCTC-07) at SVIET Chandigarh, May 2007.

[P2] M. K. Gupta, S. Meher, “An Efficient Detail Preserving Adaptive Noise Detection and Suppression Filter for Impulse Noise” communicated to International Conference on Pattern Recognition and Machine Intelligence-07 at Indian Statistical Institute Kolkata, Dec 2007.

[P3] M. K. Gupta, S. Meher, “Laplacian based Impulse Noise Detection for Adaptive Switching Median Filter” communicated to National Conference on Signal Processing and Automation (NCSPA-07) at D. Y. Patil Institute of Engineering and Technology, Pune, Sep 2007.