On the Development of Impulsive Noise Removal Schemes

Pankaj Kumar Sa

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India

On the Development of Impulsive Noise Removal Schemes

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Technology
(Research)

in

Computer Science and Engineering

by

Pankaj Kumar Sa
(Roll: 60406001)

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India
May 2006

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India.

Certificate

This is to certify that the work in the thesis entitled *On the Development of Impulsive Noise Removal Schemes* by *Pankaj Kumar Sa* is a record of an original research work carried out by him under our supervision and guidance in partial fulfillment of the requirements for the award of the degree of Master of Technology (Research) in Computer Science and Engineering during the session 2004–2006 in the department of Computer Science and Engineering, National Institute of Technology Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Rameswar Baliarsingh
Assistant Professor
CSE department of NIT Rourkela

Banshidhar Majhi
Assistant Professor
CSE department of NIT Rourkela

Place: NIT Rourkela
Date: 12 May 2006

Acknowledgment

It will be simple to name all those people who helped me to get this thesis done, however it will be tough to thank them enough. I will nevertheless try...

I would like to gratefully acknowledge the enthusiastic supervision and guidance of Prof. Banshidhar Majhi during this work. He is my source of inspiration.

I am grateful to my co-supervisor Prof. R. Baliarsingh for his valuable suggestions, and encouragements during this research period.

I am very much indebted to Prof. S. K. Jena, Head-CSE, for his continuous encouragement and support. My sincere thanks goes to Prof. S. K. Rath for motivating me to work harder.

Prof. A. K. Turuk and Prof. B. D. Sahoo were like two ceaseless source of power for me. Their help can never be penned with words.

My sincere thanks goes to Prof. G. Panda for his continuous encouragement. Special thanks goes to Prof. P. K. Nanda for appraising my work critically.

My overwhelming thanks goes to Prof. K. K. Mahapatra, Prof. S. K. Patra, Prof. S. Meher, Prof. G. K. Panda, Prof. D. P. Mahapatra and Prof. P. M. Khilar for being my knowledge resource.

I thank to all my friends for being there whenever I needed them. Thank you very much Aser, Manas, Raghunath, Saroj, Sunil, Sushant, Vamsi. I have enjoyed every moment I spent with you.

I must acknowledge the academic resource that I have got from NIT Rourkela.

Finally, I am forever indebted to my parents and my sister for their understanding and encouragement when it was most required.

Pankaj Kumar Sa

Abstract

Noise Suppression from images is one of the most important concerns in digital image processing. Impulsive noise is one such noise, which may corrupt images during their acquisition or transmission or storage etc. A variety of techniques are reported to remove this type of noise. It is observed that techniques which follow the two stage process of detection of noise and filtering of noisy pixels achieve better performance than others. In this thesis such schemes of impulsive noise detection and filtering thereof are proposed.

Two models of impulsive noise are considered in this thesis. The first one is *Salt & Pepper Noise* (SPN) model, where the noise value may be either the minimum or maximum of the dynamic gray scale range of the image. And, the second one is *Random Valued Impulsive Noise* (RVIN) model, where the noise pixel value is bounded by the range of the dynamic gray scale of the image.

Two of the proposed schemes deal with SPN model where as other four deal with RVIN model of noise. The first scheme is based on second order difference of pixels in order to identify noisy pixels. The second scheme for SPN model uses fuzzy technique to locate contaminated pixels. The contaminated pixels are then subjected to median filtering. This detection–filtration is done recursively so that filtered pixels take part in the detection of noise in the next pixel.

In the four proposed schemes for RVIN model adaptive threshold selection is emphasized. Incorporation of adaptive threshold into the noise detection process led to more reliable and more efficient detection of noise. Based on the noisy image characteristics and their statistics, threshold values are selected.

Extensive simulations and comparisons are done with competent schemes. It is observed, in general, that the proposed schemes are better in suppressing impulsive noise at different noise ratios than their counterparts.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A major portion of information received by a human from the environment is visual. Hence, processing visual information by computer has been drawing a very significant attention of the researchers over the last few decades. The process of receiving and analyzing visual information by the human species is referred to as sight, perception or understanding. Similarly, the process of receiving and analyzing visual information by digital computer is called *digital image processing* [1].

An image may be described as a two-dimensional function I.

$$I = f(x, y) \tag{1.1}$$

where x and y are spatial coordinates. Amplitude of f at any pair of coordinates (x, y) is called intensity I or gray value of the image. When spatial coordinates and amplitude values are all finite, discrete quantities, the image is called digital image [2].

Digital image processing may be classified into various subbranches based on methods whose: [2]

- input and output are images and
- inputs may be images where as outputs are attributes extracted from those images.

Following is the list of different image processing functions based on the above two classes.

- Image Acquisition

- Image Enhancement
- Image Restoration
- Color Image Processing
- Multi-resolution Processing
- Compression
- Morphological Processing
- Segmentation
- Representation and Description
- Object Recognition

For the first seven functions the inputs and outputs are images where as for the rest three the outputs are attributes from the input images. With the exception of image acquisition and display most image processing functions are implemented in software. Image processing is characterized by specific solutions, hence the technique that woks well in one area can be inadequate in another. The actual solution of a specific problem still requires a significant research and development [3].

Out of the ten subbranches of digital image processing, cited above, this thesis deals with image restoration. To be precise, the thesis devotes on a part of the image restoration i.e. noise removal from images, stated in the Problem Definition.

This is chapter is organized as follows. Image Restoration is discussed in Section 1.1 followed by a broad classification of filters in Section 1.2. The problem definition is described in Section 1.3. Different performance measures for comparison are described in Section 1.4. Review of different existing schemes and their performance analysis is done in Section 1.5. Motivation behind carrying out the work is stated in Section 1.6. Organization of the thesis is outlined in Section 1.7. Finally, Section 1.8 provides the chapter summary.

1.1 Image Restoration

The first encounter with digital image restoration in the engineering community were in the area of astronomical imaging. It primarily began with the efforts of scientists involved in the space programs of both the United States of America and the former Soviet Union in the 1950s and early 1960s. These programs were responsible for producing many incredible images of our solar system, which were at that time unimaginable. However, the images obtained from the various planetary missions of the time were subject to many photographic degradations. The 22 pictures produced during the Mariner IV flight to Mars in 1964 were later estimated to cost almost $10 million just in terms of the number of bits transmitted alone [4].

The degradations were as a result of substandard imaging environments, the vibration in machinery and spinning and tumbling of the spacecraft. Rapidly changing refractive index of the atmosphere was also one of the reasons. Pictures from the manned space mission were also blurred due to the inability of the astronaut to steady himself in a gravitation less environment while taking photographs. Extraterrestrial observations were degraded by motion blur as a result of slow camera shutter speed, relative to rapid spacecraft motions. The degradation of images was no small problem. Any loss of information due to image degradation is devastating as it reduces the scientific value of these images. Not surprisingly, astronomical imaging is still one of the primary applications of digital image restoration today.

In the area of medical imaging, image restoration has certainly played a very important part. Restoration has been used for filtering noise in X-rays, mammograms, and digital angiographic images [5]. It also finds its utility in Magnetic Resonance Imaging (MRI) [6]. Another application of image restoration in medicine is in the area of Quantitative Auto-Radiography (QAR). Images, in this area, are obtained by exposing X-ray sensitive film to a radioactive specimen. QAR is performed on post-mortem studies, and provides a higher resolution than techniques such as Positron Emission Tomography (PET), X-ray Computed Tomography (CAT), and MRI, but still needs to be improved in resolution in order to study drug diffusion and cellular uptake in the brain. Digital image restoration techniques can contribute significantly for this [7].

Another application of this field is the use of digital techniques to restore aging and deteriorated films. The idea of motion picture restoration is probably most often associated with the digital techniques used not only to eliminate scratches and dust from old movies, but also to colorize black-and-white films. Only a small subset of the vast amount of work being done in this area can be classified under the category of image restoration. Much of this work belongs to the field of computer graphics and enhancement. Nonetheless, some very important work has been done recently in the area of digital restoration of films. Digital restoration of the film "Snow White"and the "Seven Dwarfs"by Walt Disney, which originally premiered in 1937 [8] are few to cite.

Image restoration has also received some notoriety in the media, and particularly in the movies of the last decades. The climax of the 1987 movie "No Way Out"was based on the digital restoration of a blurry Polaroid negative image. The 1991 movie "JFK"made substantial use of a version of the famous Zapruder 8mm film of the assassination of the US President John F. Kennedy. It is no surprise that digital image restoration has been used in law enforcement and forensic science for a number of years. The restoration of poor quality security video tapes, blurry photographs of license plates and crime scenes are often needed when such photographs can provide the only link for solving a crime. Clearly, law enforcement agencies all over the world have made, and continue to make use of digital image restoration ideas in many forms.

One of the most exciting areas of application for digital image restoration is that in the field of image and video coding. Even though coding efficiency has improved and bit rates of coded images have reduced, there is another problem of blocking artifacts which needs significant research. These are as a result of the coarse quantization of transform coefficients used in typical image and video compression techniques. Usually, a Discrete Cosine Transform (DCT) will be applied to prediction errors on blocks of 8×8 pixels. Intensity transitions between these blocks become more and more apparent when the high-frequency data is eliminated due to heavy quantization. Already, much has been accomplished to model these types of artifacts, and develop ways of restoring coded images as a post-processing step to be performed after decompression [9–11].

Digital image restoration is being used in many other applications as well. Just to name a few, restoration has been used to restore blurry X-ray images of aircraft wings to improve aviation inspection procedures. It is used for restoring the motion induced effects present in still composite frames, and, more generally, for restoring uniformly blurred television pictures. Printing applications often require the use of restoration to ensure that halftone reproductions of continuous images are of high quality. In addition, restoration can improve the quality of continuous images generated from halftone images. Digital restoration is also used to restore images of electronic piece parts taken in assembly-line manufacturing environments. Many defense-oriented applications require restoration, such as that of guided missiles, which may obtained distorted images due to the effects of pressure differences around a camera mounted on the missile. All in all, it is clear that there is a very real and important place for image restoration technology today.

Image restoration is distinct from image enhancement techniques, which are designed to manipulate an image in order to produce results more pleasing to an observer, without making use of any particular degradation models. Image enhancement refers to the techniques of improvement of visual appearance of the image so as to be more appealing to the human. On the other hand restoration emphasizes on getting back the original image as far as possible from the degraded one. Thus the goal of image enhancement is very different from that of restoration. Better representation of image is obtained through image enhancement techniques, however, it would not be possible to define what enhancement exactly means, as an enhancement to one may be a noise to another [12].

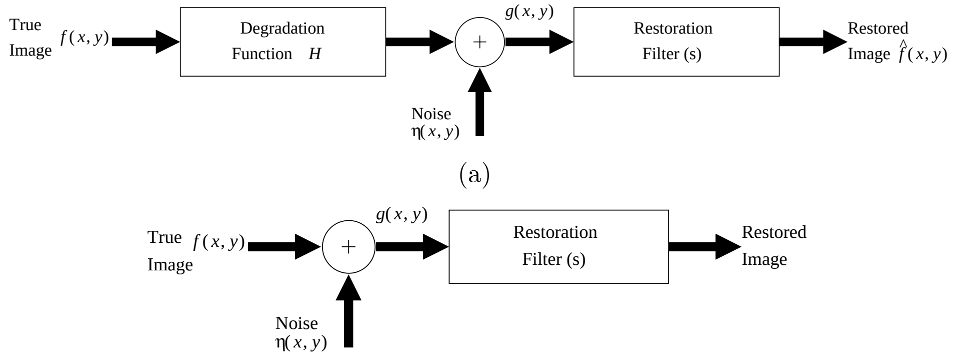
Image reconstruction techniques are also generally treated separately from restoration techniques, since they operate on a set of image projections and not on a full image. Restoration and reconstruction techniques do share the same objective, however, which is that of recovering the original image, and they end up solving the same mathematical problem, which is that of finding a solution to a set of linear or nonlinear equations.

Digital image restoration is a field of engineering that studies methods used to recover an original scene from degraded observations. Developing techniques to perform the image restoration task requires the use of models not only for the degradations, but also for the images themselves. Image restoration problem is a subset of Inverse Problem. In general, in inverse problems, the values of a certain set of functions are estimated from the known properties of other functions. Consider the following relationship

$$\mathrm{L}(\{f_i\}, \{g_j\}) = 0 \tag{1.2}$$

where L is an operator, the function, $\{f_i\}$, are sought, and the values of the functions, $\{g_j\}$, are known. When the problem is well poised, the existence of solution is assured. Also there exists a unique solution for a given problem. However, in the presence of noise, the uniqueness of solution is not assured.

The image degradation and subsequent restoration may be depicted as in Figure 1.1(a). In this thesis, however, only noise part of entire degradation is dealt with, which is shown in Figure 1.1(b). The following section provides a broad classification of restoration filters.



(a)

(b)

Figure 1.1: (a) Model of the image degradation/restoration process, (b) Model of the Noise Removal Process.

1.2 Filters

Image restoration, usually, employs different filtering techniques. Filtering may be done either in *spatial domain* or in *frequency domain.* In this thesis different spatial domain restoration techniques are studied and proposed. Broadly, filters may be classified into two categories: *Linear* and *Nonlinear.* The filtering methodologies are described below.

1.2.1 Linear Filters

In the early development of image processing, linear filters were the primary tools. Their mathematical simplicity with satisfactory performance in many applications made them easy to design and implement. However, in the presence of noise the performance of linear filters is poor. They tend to blur edges, do not remove impulsive noise effectively, and do not perform well in the presence of signal dependent noise [13].

Mathematically, a filter may be defined as an operator $L(\cdot)$, which maps a signal x into a signal y:

$$y = L(x) \tag{1.3}$$

When the operator $L(\cdot)$ satisfies both the superposition and proportionality principles, the filter is said to be linear. Two-dimensional and m-dimensional linear filtering is concerned with the extension of one-dimensional filtering techniques to two and more dimensions. If impulse response of a filter has only finite number of non-zero values, the filter is called a *finite impulse response* (FIR) filter. Otherwise, it is an *infinite impulse response* (IIR) filter [14].

If the filter evaluates the output image only with the input image, the filter is called *non-recursive.* On the other hand, if the evaluation process requires input image samples together with output image samples, it is called *recursive* filter [3, 13, 15]. Following are the few main types of filters:

- *Low-pass filter*: Smooths the image, reducing high spatial frequency noise components.

- *High-pass filter*: Enhances very low contrast features, when superimposed on a very dark or very light background.

- *Band-pass filter*: Tends to sharpen the edges and enhance the small details of the image.

1.2.2 Nonlinear Filters

Nonlinear filters also follow the same mathematical formulation as in (1.3). However, the operator $L(\cdot\,)$ is not linear in this case. Convolution of the input with its impulse response does not generate the output of a nonlinear filter. This is because of the non-satisfaction of the superposition or proportionality principles or both [13–15].

Gray scale transformations [1, 2, 16] are the simplest possible nonlinear transformations of the form (1.3). This corresponds to a memoryless nonlinearity that maps the signal x to y. The transformation

$$y = t(x) \tag{1.4}$$

may be used to transform one gray scale x to another y. *Histogram modification* is another form of intensity mapping where the relative frequency of gray level occurrence in the image is depicted. An image may be given a specified histogram by transforming the gray level of the image into another. *Histogram equalization* is one such methods that is used for this purpose. The need for it arises when comparing two images taken under different lighting conditions. The two images must be referred to the same base, if meaningful comparisons are to be made. The base that is used as standard has a uniformly distributed histogram [1, 2, 16]. Of course, a uniform histogram signifies maximum information content of the image [17]. Figure 1.2 gives a graphical representation of the various families of nonlinear filters [13].

Order statistic filters [13, 15] for noise removal are the most popular class nonlinear filters. A number of filters belongs to this class of filters, e.g., the median filter, the stack filter, the median hybrid filter etc. These filters have found numerous applications in digital image processing.

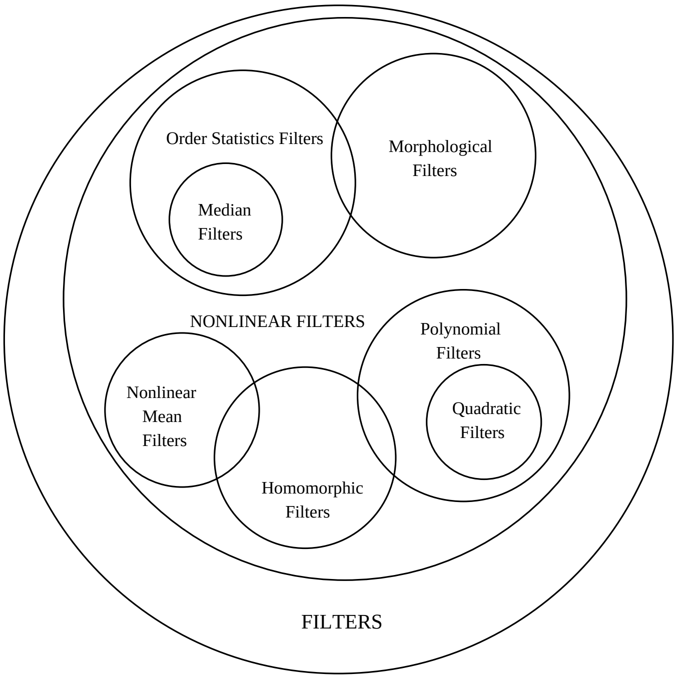


Figure 1.2: Nonlinear Filter Family

There exists some approaches that utilizes geometric features of signals. Their origin is basic set operations for image processing. These filters are called morphological filters and find applications in image processing and analysis. Biomedical image processing, shape recognition, edge detection, image enhancement are few other areas, where it is used extensively [1, 2, 13, 15, 16].

One of the oldest class of nonlinear filters, which have been used extensively in digital signal and image processing, are homomorphic filters and their extensions. These filter class find its applications in image enhancement, multiplicative and signal dependent noise removal, speech processing and also in seismic signal processing [1,2,13,15,16].

Adaptive filtering has also taken advantage of nonlinear filtering techniques. Non-adaptive nonlinear filters are usually optimized for a specific type of noise and signal. However, this is not always the case in many applications especially in image processing. Images can be modeled as two-dimensional stochastic processes, whose statistics vary in the various image regions and also from applications to applications. In such situations, adaptive filters become the natural choice and their performance depends on the accuracy of estimation of certain signal and noise statistics [1,2,13,15,16].

Considerable attenuation has been given nonlinear estimation of signals corrupted with noise. Despite impressive growth in last few decades, nonlinear filtering techniques still lack a unifying theory that encompasses existing nonlinear processing techniques.

1.3 Problem Definition

Different types of noise frequently contaminate images. Impulsive noise is one such noise, which may affect images at the time of acquisition due to noisy sensors or at the time of transmission due to channel errors or in storage media due to faulty hardware. Two types of impulsive noise models are described below.

Let $Y_{i,j}$ be the gray level of an original image Y at pixel location (i, j) and $[n_{min}, n_{max}]$ be the dynamic range of Y. Let $X_{i,j}$ be the gray level of the noisy image X at pixel (i, j) location. *Impulsive Noise* may then be defined as:

$$X_{i,j} = \begin{cases} Y_{i,j} & \text{with } 1 - p \\ R_{i,j} & \text{with } p \end{cases} \tag{1.5}$$

where, $R_{i,j}$ is the substitute for the original gray scale value at the pixel location (i, j). When $R_{i,j} \in [n_{min}, n_{max}]$, the image is said to be corrupted with *Random Valued Impulsive Noise* (RVIN) and when $R_{i,j} \in \{n_{min}, n_{max}\}$, it known as *Fixed Valued Impulsive Noise* or *Salt & Pepper Noise* (SPN). Pixels replaced with RVIN and their surroundings exhibit very similar behavior. These pixels differ less in intensity, making identification of noise in RVIN case far more difficult than in SPN.

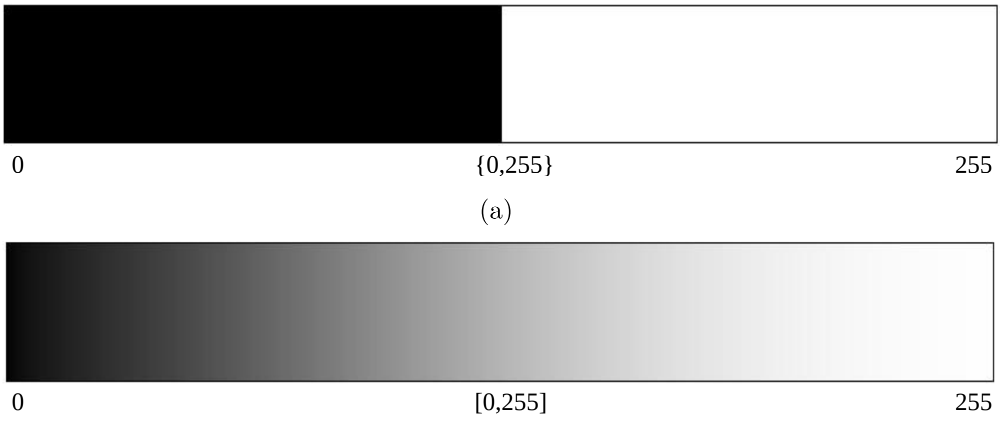
The difference between SPN and RVIN may be best described by Figure 1.3. In the case of SPN the pixel substitute in the form of noise may be either $n_{min}(0)$ or $n_{max}(255)$. Where as in RVIN situation it may range from n_{min} to n_{max}.

1.4 Performance Measures

The metrics used for performance comparison of different filters (exists and proposed) are defined below.

a. Peak Signal to Noise Ratio ($PSNR$)

   $PSNR$ analysis uses a standard mathematical model to measure an objective difference between two images. It estimates the quality of a reconstructed



(a)

(b)

Figure 1.3: Representation of (a) *Salt & Pepper Noise* with $R_{i,j} \in \{n_{min}, n_{max}\}$, (b) *Random Valued Impulsive Noise* with $R_{i,j} \in [n_{min}, n_{max}]$

image with respect to an original image. The basic idea is to compute a single number that reflects the quality of the reconstructed image. Reconstructed images with higher PSNR are judged better. Given an original image Y of size $(M \times N)$ pixels and a reconstructed image $\hat{Y}$, the $PSNR(dB)$ is defined as:

$$PSNR(dB) = 10\log_{10}\left(\frac{255^2}{\frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(Y_{i,j}-\hat{Y}_{i,j}\right)^2}\right) \tag{1.6}$$

b. Percentage of Spoiled Pixels (PSP)

PSP is a measure of percentage of non-noisy pixels change their gray scale values in the reconstructed image. In other words it measures the efficiency of noise detectors. Hence, lower the PSP value better is the detection, in turn better is the filter performance.

$$PSP = \frac{\text{number of non-noisy pixels changed their gray value}}{\text{total number of non-noisy pixels}} \times 100 \tag{1.7}$$

1.5 Literature Survey

Noise removal from a contaminated image signal is a prominent field of research and many researchers have suggested a large number of algorithms and compared their results. The main thrust on all such algorithms is to remove impulsive noise while preserving image details. These schemes differ in their basic methodologies applied to suppress noise. Some schemes utilize detection of impulsive noise followed by filtering whereas others filter all the pixels irrespective of corruption. In this section an attempt has been made for a detail literature review on the reported articles and study their performances through computer simulation. We have classified the schemes based on the characteristics of the filtering schemes and described are below.

A. Filtering without Detection

In this type of filtering a window mask is moved across the observed image. The mask is usually of size $(2N+1)^2$, where N is a positive integer. Generally the center element is the pixel of interest. When the mask is moved starting from the left-top corner of the image to the right-bottom corner, it performs some arithmetical operations without discriminating any pixel.

B. Detection followed by Filtering

This type of filtering involves two steps. In first step it identifies noisy pixels and in second step it filters those pixels. Here also a mask is moved across the image and some arithmetical operations is carried out to detect the noisy pixels. Then filtering operation is performed only on those pixels which are found to be noisy in the previous step, keeping the non-noisy intact.

C. Hybrid Filtering

In such filtering schemes, two or more filters are suggested to filter a corrupted location. The decision to apply a particular filter is based on the noise level at the test pixel location or performance of the filter on a filtering mask.

All those filtering schemes that are reviewed are described in this section under their respective head.

A. Filtering without Detection

As discussed in the previous section, this technique does not detect contaminated pixels. It applies the filtering mechanism through out the subject without discriminating any pixel.

A1. Moving Average [2]

This is a simple linear filter. Average of all pixels of a sliding window is replaced with the pixel of interest.

$$\hat{Y}_{i,j} = \frac{1}{m \times n} \sum_{(u,v)\in S_{mn}} X_{u,v} \tag{1.8}$$

where, X is the noisy image, $\hat{Y}$ is the restored image and S_{mn} is the sliding window of size $m \times n$ centered around (i, j). Its performance both in subjective as well as objective way is very poor.

A2. Median (3×3) A3. Median (5×5) [2]

The median filter (1.9) is one of the most popular nonlinear filters. It is very simple to implement and much efficient as well. But the cost is that it blurs the image and edges are not preserved. It acts like a low pass filter which blocks all high frequency components of the image like edges and noise, thus blurs the image.

$$\hat{Y}_{i,j} = \text{MEDIAN}_{(u,v)\in S_{mn}}\left(X_{u,v}\right) \tag{1.9}$$

Depending upon the sliding window mask there may be many variations of median filter. Here we have reviewed two such variations. Median (3×3) filter makes use of a 3×3 sliding window, whose center pixel is replaced with the median value of all the 9 pixels of the window. This kind of filter is helpful when noise is scattered throughout the image. Whereas median (5×5) filter replaces the pixel of interest i.e. the center pixel with the median value of all the 25 pixels of the sliding window. When noise appears in blotch, this type of filter works better. But for other situations it produces disappointing results.

A4. WM[1] $k = 1$ A5. WM $k = 2$ [18–21]

This is another nonlinear median filter, which favors the center pixel than others. Let the window size be $(2n+1)^2$ and $L = 2n(n+1)$. The filter is defined as:

$$\hat{Y}_{i,j}^{2k} = \text{MEDIAN}\{X_{i-u,j-v}, (2k) \diamond X_{i,j} | -h \leq u, v \leq h\} \tag{1.10}$$

where $2k$ is the weight given to pixel (i, j), and $\diamond$ represents the repetition operation. Hence in a 3×3 window $\hat{Y}_{i,j}^{2k}$ is the median of $(9 + 2k)$ gray values with the center value of the window repeated $(2k + 1)$ times. $\hat{Y}_{i,j}^{0}$ is the standard median filter, where as $\hat{Y}_{i,j}^{2k}$ becomes identity filter when $k \geq L$. Two variations of WM (with $k = 1$ and $k = 2$) have been simulated. When the noise percentage is low, both the filters work better but beyond 10% of noise the performance starts deteriorating. If noise appears as blotch in a window, it leaves the blotch as it is as if no filtering has been done.

B. Detection followed by Filtering

Such filtering schemes differentiate between noisy and non-noisy pixels. These filters, in general, consist of two steps. Detection of noisy pixels is followed by filtering. Filtering mechanism is applied only to the noisy pixels.

B1. Rank-Ordered Mean [22]

This is an adaptive approach to solve the restoration problem in which filtering is conditioned on the current state of the algorithm. The state variable is defined as the output of a classifier that acts on the differences between the current pixel value and the remaining ordered pixel values inside a window centered around the pixel of interest.

This scheme is undoubtedly one of the robust and simple scheme but it fails in preserving the finer details of the image.

[1] WM: Weighted Median

B2. Progressive-Switching Median [23]

It is a median based filter, which works in two stages. In the first stage an impulse detection algorithm is used to generate a sequence of binary flag images. This binary flag image predicts the location of noise in the observed image. In the second stage noise filtering is applied progressively through several iterations.

This filter is a very good filter for fixed valued impulsive noise but for random values the performance is abysmal.

B3. Adaptive Center Weighted Median Filter [19]

This work is an improvement of previously described Center Weighted Median (CWM) filter. It works on the estimates based on the differences between the current pixel and the outputs of the CWM filters with varied center weights. These estimates decide the switching between the current pixel and median of the window.

This is a good filter and is robust for a wide variety of images. But it is inefficient in recovering the exact values of the corrupted pixels.

B4. Adaptive Two-Pass Median [24]

As the name suggests it employs median filter on the noisy image twice. This adaptive system tries to correct for false replacements generated by the first round of median filtering operation. Based on the estimated distribution of the noise, some pixels changed by first median filter are replaced by their original values and kept unchanged in the second median filtering. And in the second round it filters out the remaining impulses.

Even though the filter gives some good results in terms of noise suppression but spoiling of good pixels is more and it results in overall poor performance.

B5. Accurate Noise Detector [25]

This filter justifies its name by detecting noise to the perfection. Based on Progressive Switching Median Filter, it generates an edge flag image to classify the pixels of noisy image into ones in the flat regions and edge regions. The two types of pixels are processed by different noise detector. When noise is very high prevention of false-detection and non-detection becomes difficult. Therefore, another iteration is dedicated for verification of the noise flag image.

This scheme exhibits good performance on images not only with low noise density but also with high percentage of corruption. But all these come at the cost of computational complexity which is very high and not at all suitable for real time applications.

B6. SM[2] (5 × 5) B7. SM (7 × 7) B8. SM (9 × 9) [26]

This is also a two stage process, where in the first stage noise detection is carried out and in the second stage filtering is done. The noisy image is convolved with a set of convolution kernels. Each of the kernels are sensitive to edges in a different orientation. The minimum absolute value of these four convolutions is used for impulse detection by comparing with a threshold. By varying the size of kernel different variations of SM may be obtained. Three such variations of SM are reviewed here in this paper.

Because of its four kernels it detects noise effectively even in those images where the edge density is more. But when the kernel size increases to 7 × 7 and 9 × 9 it fails in doing so. Also it fails in preserving finer details.

B9. Differential Ranked Impulse Detector [27]

This is another nonlinear technique which also works in two stages. It aims at filtering only corrupted pixels. Identification of such pixels is done by comparing signal samples within a narrow rank window by both rank and absolute value. The first estimate is based on the comparison between the rank of the pixel of interest and rank of the median. The second estimate is based on the brightness value which is analyzed using the median.

It is a good filter in low noise conditions but the performance slightly degrades in beyond 20% of noise. It also leaves noise blotch without correcting.

[2] SM: Switching Median

B10. Enhanced Ranked Impulse Detector [27]

This scheme is an alteration of the scheme described above. Here the brightness is analyzed by calculating the difference of pixel of interest with its closest neighbors in the variational series.

Its performance is very good at low noise but fails miserably at noise density more than 20%.

B11. Advanced Impulse Detection Based on Pixel-Wise MAD [28]

This scheme is based on modified median of absolute deviation from median (MAD). MAD is used to estimate the presence of image details. An iterative pixel wise modification of MAD is used here that provides a reliable removal of impulses. Its performance is more than average and fails when the edge density is more.

B12. Minimum-Maximum Exclusive Mean [29]

This is a simple nonlinear, robust filter that centers around two windows of size 3×3 and 5×5. It checks for a particular range of gray level in the 3×3 windows. If it fails it goes to 5×5 window. If average of all the pixels of that particular range is more than certain value then that pixel is replaced with the average, otherwise it is left intact. This is one of the good schemes because of its simplicity and easy implementation.

B13. Peak and Valley [30]

This recursive nonlinear filter is composed of two conditional rules. It compares the test pixel with surrounding neighbor pixels for some conditions. It then replaces the pixel of interest with the most conservative surrounding pixel. This scheme is computationally efficient over others but at the same time it spoils non-noisy pixels to a greater extent.

B14. Detail preserving impulsive noise removal [31]

Unlike thresholding techniques, it detects noisy pixels non-iteratively using the surrounding pixel values. It is based on a recursive minimum maximum method of Peak and Vally scheme. When the image contains numerous edges like *Babbon*,*Clown* etc. this technique totally fails.

B15. Signal-Dependent Rank Ordered Mean [32]

This is one of the most efficient nonlinear algorithms to suppress impulsive noise from highly corrupted images. Based on detection-estimation strategy, this algorithm replaces the identified noisy pixel with rank ordered mean of it surroundings.

C. Combined Filtering

Two or more filters are employed in this type of filtering mechanism. In addition to this a switch is used whose logic helps in switching among the employed filters. The switch may take output of individual filter into consideration or by some other means to decide which filter should be employed for a particular window such that the final output would be the best.

C1. Tri-State Median Filtering [33]

This combined filter comprises of standard median filter, identity filter, center weighted median filter and a switching logic. Noise detection is realized by an impulse detector, which takes the outputs from the standard median and center weighted median filters and compares them with the center pixel value in order to make a tri-state decision. The switching logic is controlled by a threshold value. Depending on this threshold value, the center pixel value is replaced by the output of either SM filter or CWM filter or identity filter. This is one of the good schemes reviewed in this paper.

C2. Two-Output Filter [34]

The two-output nonlinear filter is based on the subsequent activation of two recursive filtering algorithms that operates on different subsets of input data. One subset is the right-bottom 3 × 3 sub-window and the other one is left-top 3 × 3 sub-windows of a 4 × 4 sliding window. Two center pixels of both 3 × 3 sub-windows are updated at each step. Rank ordered filtering is used to remove impulsive noise. This is a good scheme and gives very good result under fixed valued impulsive noise conditions. But under random valued impulsive noise it fails miserably.

C3. MRHF[3]-1 C4. MRHF-2 C5. MRHF-3 [35]

This is a class of non-linear filters called Median Rational Hybrid Filters based on a rational function. The filter output is the result of a rational operation taking into account three sub function. In all the three operations the central operation is CWM.

In MRHF-1 the CWM gives ϕ_2 and two FIR sub filters give ϕ_1 and ϕ_3. The rational function on ϕ_1, ϕ_2 and ϕ_3 decides which of the filter is most suitable.

In MRHF-2 the sub-filters are four unidirectional median filters. Mean of two median filters gives ϕ_1 and mean of other two gives ϕ_3. And the CWM gives ϕ_2. The rational function decides based on these three ϕ values.

In MRHF-3 two bidirectional median filter give ϕ_1 and ϕ_3. Together with ϕ_2 from CWM the rational function takes the decision.

Spoiling of non-noisy pixels is high in all the three filters. When compared among the three, the MRHF-2 outperforms other two.

1.5.1 Results and Discussions

Lena image corrupted with RVIN (1% to 30% of noise) is subjected to the different filtering schemes discussed above and their performance is measured using metrics (1.6) and (1.7). Table-1.1 lists the $PSNR$ where as Table-1.2 lists the PSP of different filters. Figures 1.4– 1.6 depict the performance of each scheme in their respective groups.

The performance in $PSNR$ of Group-A schemes is depicted in Figure 1.4. The performance of A1 is very poor in comparison to others. A3's performance is steady, which is around $30dB$. A5 is in commanding position at very low noise

[3] MRHF: Median Rational Hybrid Filter

density but flunks at other situations. A2 is better in the upper half where as A4 is better in the lower half of noise density.

Group-B performance is depicted in Figure 1.5. B15 is one of the filters that outperforms rest all. When comparing other schemes it can be seen that, in the very low noise density (around 1%) B6 and B9 outperforms all others. When the density increases (low noise, 5%–15%) B3, B4, B10 and B11 performs equally good but B6 and B9 decline drastically. When the density further increases (medium noise, 20%–30%), all the schemes perform more or less same. But B3 and B4 are slightly better than others in this range of noise.

Figure 1.6 unequivocally depicts that C1 is not only the winner in Group-C but also outperforms all other schemes in the same group. Performances of C3, C4 and C5 are almost same where as C2 produces very very poor results.

Some of the schemes, whose performance is better in SPN model of noise are also compared. Figure 1.10 shows the PSNR (dB) variations and Figure 1.11 PSPs of such schemes.

1.6 Motivation

The last section reveals many things about the existing restoration schemes. Most of the reported schemes work well under SPN but fails under RVIN, which is more realistic when it comes to real world applications. Even though some of the reported methods [19, 24–28, 31, 32, 34–36] claim to be adaptive, they are not truly adaptive for the simple reason of not considering the image and noise characteristics. These schemes generally use a threshold value for the identification of noise. A predefined parameter is compared with this threshold value. If it exceeds, the pixel is marked as contaminated otherwise not. Usually the threshold value used is either a constant or a set of four/five values. A threshold, which is optimal in one environment may not be good at all in a different environment. By environment we mean, the type of image, characteristic and density of noise.

Further, there has been little or no usage of soft computing techniques in the reported schemes. Soft computing methodologies mimic the remarkable human capability of making decision in ambiguous environment. It embraces approximate reasoning, imprecision, uncertainty and partial truth. There exists scope for improving the detector's performance using softcomputing techniques.

These facts motivated us:

- to work towards improved and efficient detectors for identifying contaminated pixels.
- to devise adaptive thresholding techniques so that noise detection would be more reliable.
- to exploit the computational power of soft computing techniques in predicting the threshold value by adapting to the environment with a greater ease.

1.7 Thesis Organization

The rest of the thesis is organized as follows.

Chapter 2 proposes two restoration schemes for images contaminated with *Salt & Pepper Noise*. The first one is based on second order difference of pixels to identify noisy pixels which subsequently uses median filtering on the identified pixels only. The second scheme utilizes fuzzy technique to detect noise in a given window of an image. When compared with other existing techniques, the proposed schemes show good performance.

Techniques proposed in **Chapter 3** are based on second order difference of pixels with adaptive threshold value in order to identify *Random Valued Impulsive Noise* from images. Three different ways of selecting the threshold value are proposed. In the first method an equation is fitted to make the decision process adaptive to the noise and image characteristics. The second approach uses an artificial neural network trained with backpropagation algorithm to detect noisy pixels. And the third scheme uses an improved neural network i.e. functional link artificial neural network to estimate the threshold. Comparative analysis with most recent techniques reveal that our techniques are better in terms of noise suppression.

The last proposed scheme, presented in **Chapter 4**, also utilizes an artificial neural network, but with more emphasis on selection of proper input parameters to be used. Two different parameters are used in this scheme, which reduces the training time considerably and the noise detection becomes more accurate. Exhaustive simulations on different standard images and subsequent comparisons reveal that this proposed scheme outperforms existing schemes both qualitatively as well as quantitatively.

Finally **Chapter 5** presents the concluding remark, with scope for further research work.

1.8 Summary

The fundamentals of digital image processing, sources of noise and types of noise in an image, the existing filtering schemes and their merits and demerits and the various image metrics are studied in this chapter. Applications of soft computing techniques have been underutilized in the surveyed schemes. To derive the benefits of this paradigm, investigation has been made in this thesis to develop some novel schemes in the area of image restoration.

Table 1.1: Comparative Results in PSNR (dB) of different filters for *Lena* image corrupted with RVIN of varying strengths

$Noise \Rightarrow$	1%	5%	10%	15%	20%	25%	30%
$Filters \Downarrow$							
A1	28.16	26.76	24.19	22.44	21.07	20.01	19.11
A2	35.04	33.96	32.81	31.65	30.25	28.94	27.39
A3	30.95	30.4	29.82	29.22	28.48	27.96	27.32
A4	38.19	36.11	33.99	31.76	29.32	27.30	25.25
A5	41.16	36.05	31.26	27.73	24.87	22.70	20.88
B1	31.64	31.01	30.25	29.60	28.86	28.15	27.39
B2	31.95	31.28	30.81	30.05	29.27	28.54	27.84
B3	36.08	34.77	33.37	32.21	31.12	29.02	28.02
B4	35.32	34.15	32.94	31.88	30.59	29.40	28.19
B5	31.51	30.33	29.03	28.23	27.18	26.59	25.84
B6	40.55	35.01	31.84	29.71	27.97	26.66	25.43
B7	33.9	31.86	30.3	29.01	27.57	26.59	25.60
B8	30.48	29.31	28.09	27.11	26.03	25.18	24.22
B9	42.08	36.01	32.12	28.89	26.40	24.35	22.68
B10	39.21	36.06	33.85	31.64	30.80	28.91	27.22
B11	37.10	35.47	33.55	31.72	29.52	27.34	25.39
B12	38.93	33.47	30.06	27.63	25.67	24.13	22.73
B13	35.99	34.68	32.89	30.93	28.36	26.48	24.35
B14	36.87	33.34	29.53	26.62	24.37	22.64	21.07
B15	42.90	38.68	35.80	33.95	32.24	30.90	29.63
C1	39.49	36.06	34.01	31.61	29.09	28.46	27.88
C2	31.92	24.92	21.97	20.21	18.91	17.95	17.06
C3	30.97	29.38	27.03	24.89	23.02	21.46	20.13
C4	32.01	30.24	27.73	25.47	23.50	21.84	20.42
C5	31.59	30.02	27.73	25.56	23.62	22.02	20.60

Table 1.2: Comparative Results in PSP of different filters for *Lena* image corrupted with RVIN of varying strengths

$Noise \Rightarrow$ $\overline{Filters} \Downarrow$	1%	5%	10%	15%	20%	25%	30%
A1	98.99	99.08	99.13	99.18	99.32	99.45	99.56
A2	66.71	67.4	68.12	68.45	69.14	69.63	69.97
A3	78.23	78.84	79.43	79.73	80.34	80.77	80.98
A4	40.58	40.28	39.7	39.34	38.99	38.67	38.27
A5	22.84	21.77	20.60	19.52	18.40	17.61	16.92
B1	96.85	97.03	97.25	97.39	97.53	97.78	97.96
B2	00.00	00.05	00.09	00.05	00.11	00.00	00.22
B3	06.36	06.70	07.06	07.55	07.89	08.57	09.38
B4	59.01	59.52	60.05	60.56	61.14	61.74	62.29
B5	00.00	00.00	00.02	00.05	00.08	00.06	00.00
B6	00.13	00.17	00.21	00.26	00.32	00.44	00.58
B7	01.27	01.49	01.62	01.96	02.29	02.88	03.37
B8	03.63	04.06	04.49	05.22	06.03	07.13	08.47
B9	00.09	00.09	00.1	00.12	00.12	00.16	00.17
B10	01.17	01.21	01.33	01.49	01.79	02.21	02.75
B11	08.70	08.81	08.90	09.05	09.20	09.50	10.08
B12	00.33	00.60	01.24	02.21	03.62	05.37	07.68
B13	51.46	51.86	52.31	52.71	53.43	53.85	54.41
B14	25.63	23.87	21.98	19.93	18.25	17.13	15.63
B15	0.28	0.28	0.33	0.32	0.34	0.40	0.47
C1	00.74	00.79	00.89	00.99	01.16	01.14	01.59
C2	00.01	00.03	00.05	00.06	00.09	00.09	00.13
C3	99.12	99.15	99.18	99.19	99.2	99.21	99.22
C4	92.79	93.34	93.87	94.39	94.99	95.34	95.87
C5	92.28	92.72	93.16	93.62	94.14	94.45	94.91

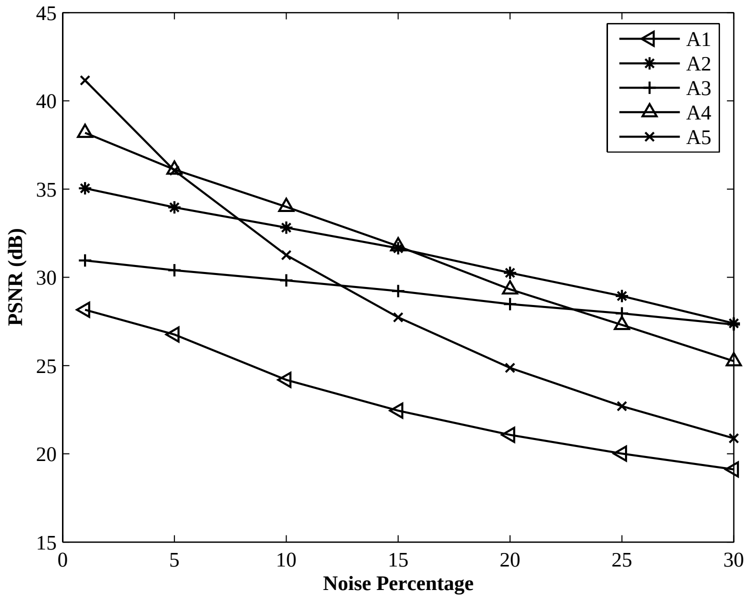


Figure 1.4: PSNR (dB) variations of *Lena* image corrupted with RVIN by Group-A schemes

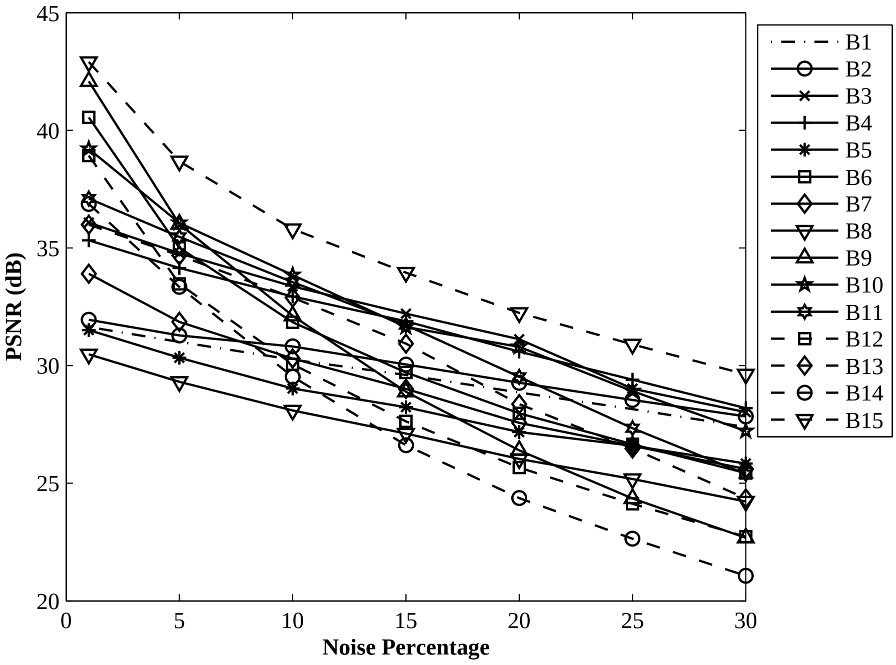


Figure 1.5: PSNR (dB) variations of *Lena* image corrupted with RVIN by Group-B schemes

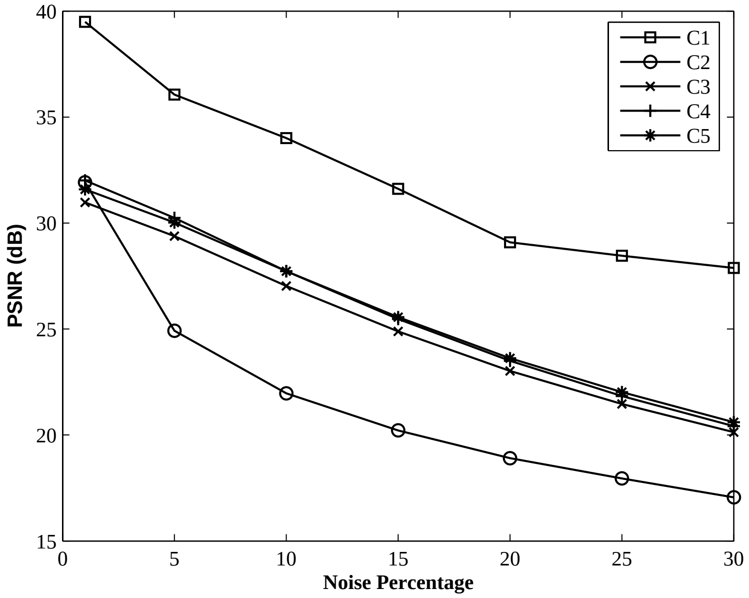


Figure 1.6: PSNR (dB) variations of *Lena* image corrupted with RVIN by Group-C schemes

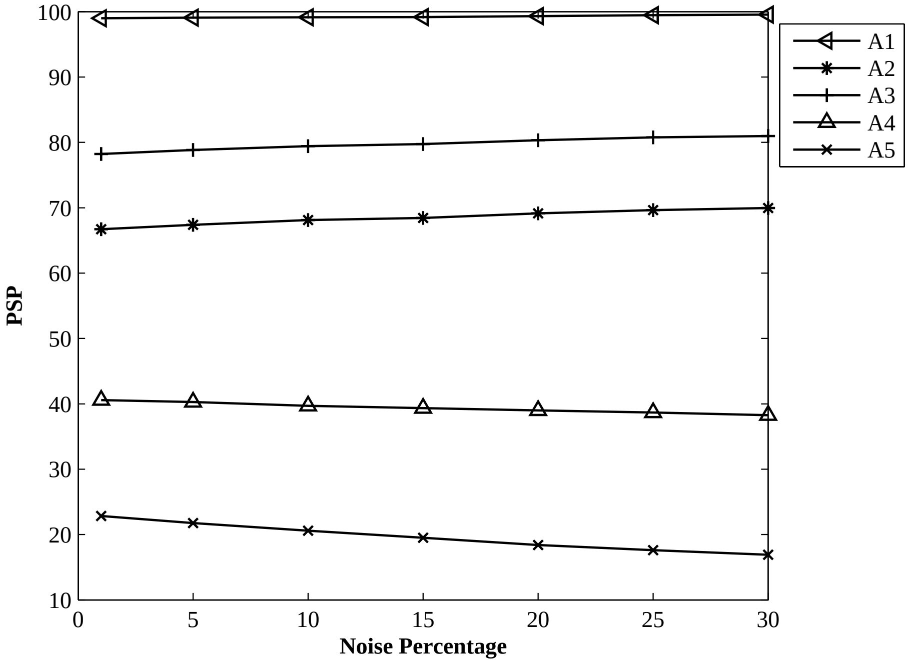


Figure 1.7: PSP variations of *Lena* image corrupted with RVIN by Group-A schemes

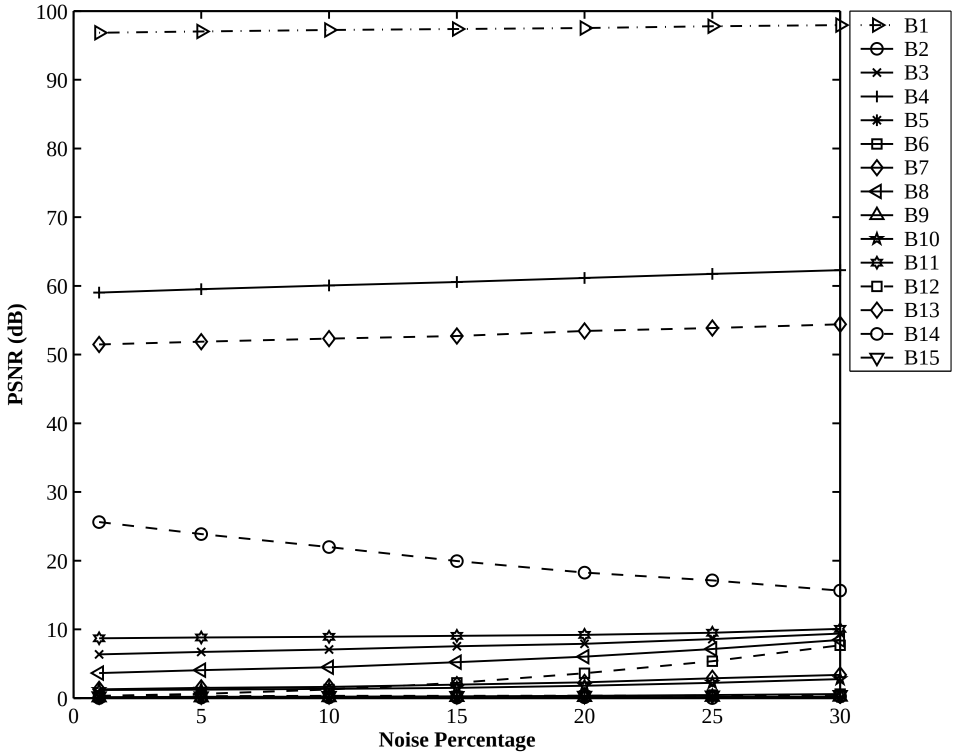


Figure 1.8: PSP variations of *Lena* image corrupted with RVIN by Group-B schemes

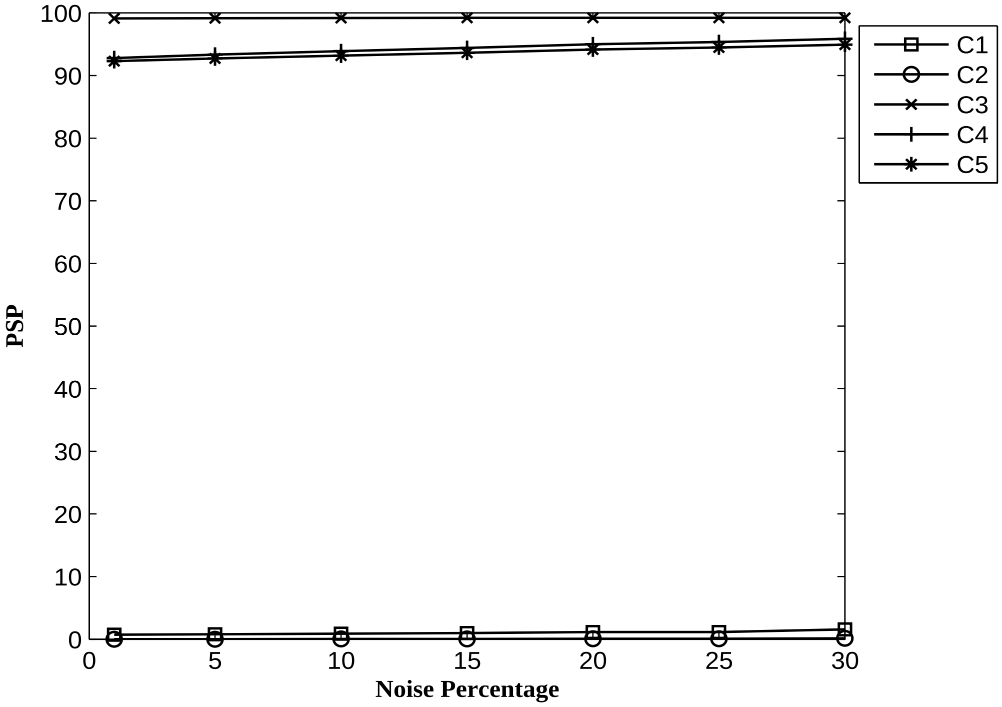


Figure 1.9: PSP variations of *Lena* image corrupted with RVIN by Group-C schemes

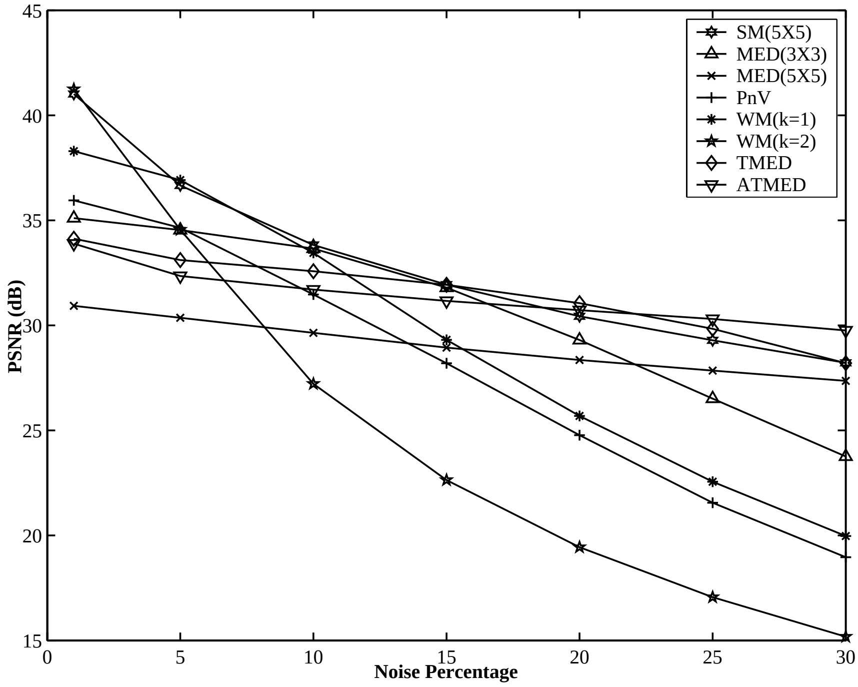


Figure 1.10: PSNR (dB) variations of *Lena* image corrupted with SPN

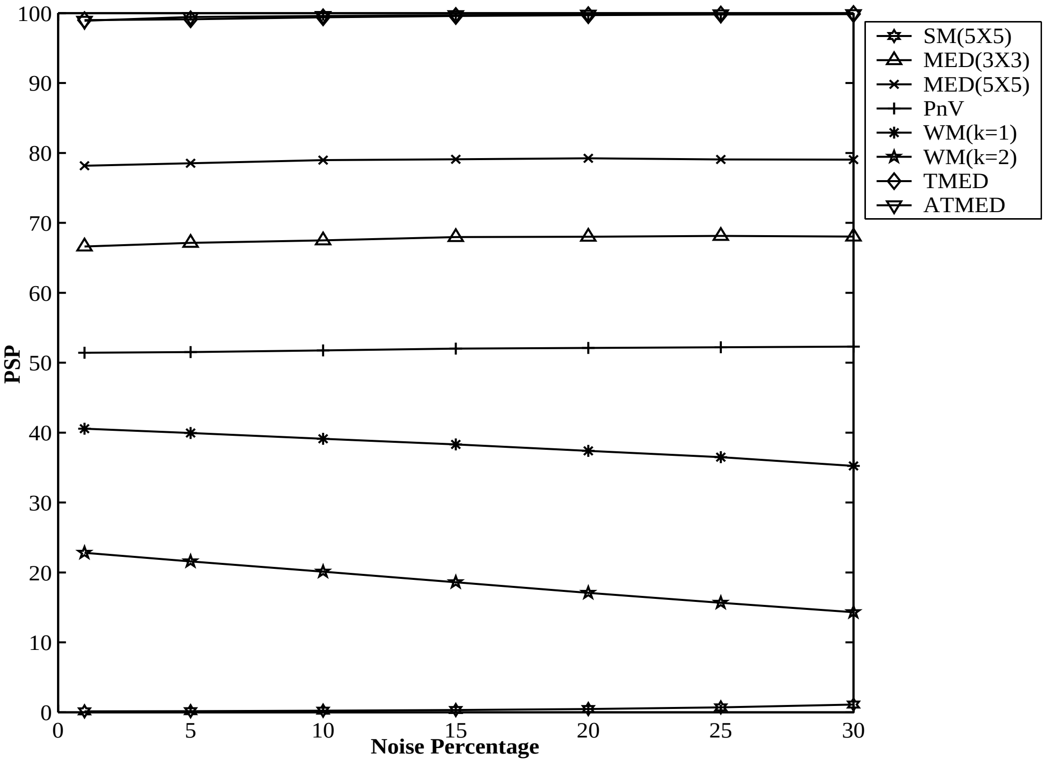


Figure 1.11: PSP variations of *Lena* image corrupted with SPN

Chapter 2

Efficient Impulsive Noise Removal Schemes

Two of the proposed schemes for eliminating impulsive noise from digital images are presented in this chapter. *Salt & Pepper Noise* (SPN) model of impulsive noise (Section 1.3) is considered here. The first scheme, *Decision Directed Median Filter*, is described in Section 2.1. The second scheme, *Fuzzy Impulsive Noise Detection*, is based on fuzzy logic which is discussed in Section 2.2. Comparison of the two proposed schemes with some of the existing schemes is presented in Section 2.3. Finally, the chapter is concluded with a brief summary in Section 2.4.

2.1 Decision Directed Median Filter

Usually the pixels located in the neighborhood of a test pixel are correlated to each other and they possess almost similar characteristics. Most of the reported impulse detection schemes exploit this feature of pixels. The scheme proposed here is one such novel technique of impulsive noise detection-suppression strategy from corrupted images. This scheme is simple but efficient and works alternatively in two phases: *detection of noisy pixels* followed by *median filtering*.

Methodology of the proposed scheme is outlined in Section 2.1.1. The algorithm is presented in Section 2.1.2 followed by the filtering algorithm in Section 2.1.3.

2.1.1 Methodology

In a practical situation, since the probability p (1.5) is less than 1, all the pixels of a digital image are not corrupted with the impulsive noise. In addition, when the probability of corruption is not cent percent, it is expected that the noisy pixel be surrounded by at least some healthy pixels. However, this assumption is not true as the noise density becomes very high. In any case, the total number of corrupted pixels is less than the total number of pixels in the image. Hence, it is not required to perform filtering operation on every pixel for eliminating the impulsive noise. Rather, it is computationally economical to filter only the corrupted pixels leaving the healthy pixels unchanged. This approach reduces the blurring effect in the restored image, as the magnitude of healthy pixels is not affected by filtering. Basically, the noise removal method proposed in this paper constitutes two tasks: identification of corrupted pixels and filtering operation on those corrupted pixels. Thus the effectiveness of this scheme lies on the accuracy and robustness of detection of noisy pixels and efficiency of the filtering methodology employed. Many researchers [19, 33, 37, 38] have suggested various methods for locating the distorted pixels as well as filtering techniques. Each of these methods has different shortcomings and hence fails to reproduce images very close to original ones. These are either over-filtering distortion, blurring effect or high computational involvement. In addition, as the density of the impulsive noise is gradually increased, the quality of the image recovered by the existing methods correspondingly degrades. The scheme proposed here, is an improved impulsive noise detection scheme followed by recursive median filtering to overcome many of the shortcomings observed in the existing methods.

To achieve this objective, it is necessary to devise an effective impulse detection scheme prior to filtering operation. The proposed scheme employs a second order difference based impulse detection mechanism at the location of a test pixel. The mathematical formulation of the proposed method is presented in (2.1),

$$\hat{Y}_{i,j} = \begin{cases} Z_{i,j} & \text{if } d_{i,j} = 0 \\ X_{i,j} & \text{if } d_{i,j} = 1 \end{cases} \tag{2.1}$$

where, $d_{i,j}$ is the decision index that controls the filtering operation and estimates the filtered output $\hat{Y}_{i,j}$ from the observed image $X_{i,j}$ and filtered pixel value $Z_{i,j}$. If the impulse detector determines that the center pixel of test window is noisy, then $d_{i,j} = 0$, otherwise $d_{i,j} = 1$. When $d_{i,j} = 0$, the corrupted pixel undergoes median filtering. On the other hand $d_{i,j} = 1$, the window is skipped and the process is repeated. Unlike in conventional methods, the filtering operation is performed selectively based on the decision of the impulse detector. Hence the proposed method is named as *Decision Directed Median Filter* (DDMF). The schematic diagram of the proposed filtering scheme is shown in Figure 2.1.

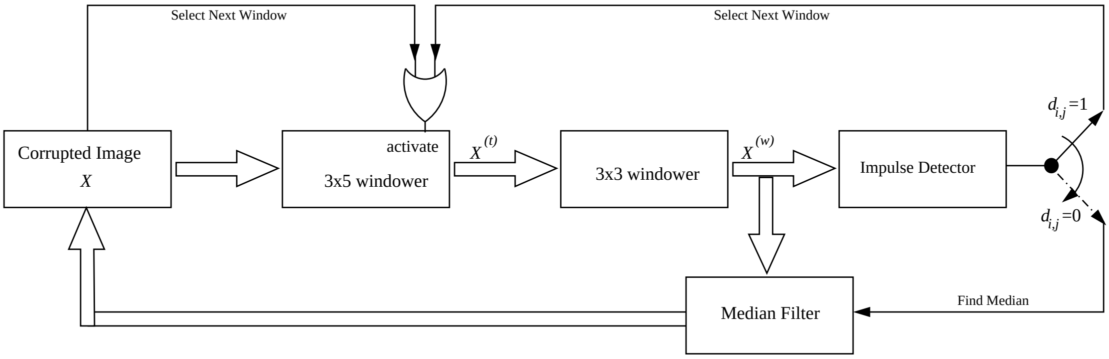


Figure 2.1: Schematic Diagram of the Proposed Filter

2.1.2 The DDMF Algorithm

The detailed algorithm for impulse detection and filtering is described below.

i. Choose a test window $X^{(t)}$ of size 3×5 located at the top-left corner of the observed image X.

$$X^{(t)} = \begin{pmatrix} X_{i-1,j-2} & X_{i-1,j-1} & X_{i-1,j} & X_{i-1,j+1} & X_{i-1,j+2} \\ X_{i,j-2} & X_{i,j-1} & X_{i,j} & X_{i,j+1} & X_{i,j+2} \\ X_{i+1,j-2} & X_{i+1,j-1} & X_{i+1,j} & X_{i+1,j+1} & X_{i+1,j+2} \end{pmatrix} \tag{2.2}$$

Consider a 3×3 sub-window $X^{(w)}$ from $X^{(t)}$ as:

$$X^{(w)} = \begin{pmatrix} X_{i-1,j-1} & X_{i-1,j} & X_{i-1,j+1} \\ X_{i,j-1} & X_{i,j} & X_{i,j+1} \\ X_{i+1,j-1} & X_{i+1,j} & X_{i+1,j+1} \end{pmatrix} \tag{2.3}$$

ii. Compute the first order 3 × 4 difference matrix $f^{(d)}$ from $X^{(t)}$ as:

$$f^{(d)} = \begin{pmatrix} f^{(d)}_{i-1,j-1} & f^{(d)}_{i-1,j} & f^{(d)}_{i-1,j+1} & f^{(d)}_{i-1,j+2} \\ f^{(d)}_{i,j-1} & f^{(d)}_{i,j} & f^{(d)}_{i,j+1} & f^{(d)}_{i,j+2} \\ f^{(d)}_{i+1,j-1} & f^{(d)}_{i+1,j} & f^{(d)}_{i+1,j+1} & f^{(d)}_{i+1,j+2} \end{pmatrix} \tag{2.4}$$

where $f^{(d)}_{i+k,j+l} = X^{(t)}_{i+k,j+l} - X^{(t)}_{i+k,j+l-1}$, $k = -1, 0, 1$ and $l = -1, 0, 1, 2$.

iii. Compute the second order 3 × 3 difference matrix $s^{(d)}$ from $f^{(d)}$ as:

$$s^{(d)} = \begin{pmatrix} s^{(d)}_{i-1,j-1} & s^{(d)}_{i-1,j} & s^{(d)}_{i-1,j+1} \\ s^{(d)}_{i,j-1} & s^{(d)}_{i,j} & s^{(d)}_{i,j+1} \\ s^{(d)}_{i+1,j-1} & s^{(d)}_{i+1,j} & s^{(d)}_{i+1,j+1} \end{pmatrix} \tag{2.5}$$

where $s^{(d)}_{i+p,j+q} = f^{(d)}_{i+p,j+q+1} - f^{(d)}_{i+p,j+q}$, $p = -1, 0, 1$ and $q = -1, 0, 1$.

iv. Apply the following rule for impulse detection at pixel $X_{i,j}$:

- If $s^{(d)}_{i,j}$ is a high magnitude negative quantity then $X_{i,j}$ is corrupted by a positive impulsive noise.
- If $s^{(d)}_{i,j}$ is a high magnitude positive quantity then $X_{i,j}$ is corrupted by a negative impulsive noise.

Since the objective is to detect the presence of an impulsive noise rather than its type, compute the absolute value of the second order difference, $|s^{(d)}|$.

v. A decision index d, for the test pixel at i, j is obtained by passing $s^{(d)}$ through a hard limiter H that saturates at a threshold value θ. The output of the hard limiter is given by:

$$d_{i,j} = \mathrm{H}(|s^{(d)}_{i,j}|) = \begin{cases} 0 & \text{if } |s^{(d)}_{i,j}| > \theta \\ 1 & \text{otherwise} \end{cases} \tag{2.6}$$

vi. Apply the binary decision rule for impulse detection at $X_{i,j}$ as:

- If $d_{i,j}$ is zero, then the test pixel $X_{i,j}$ is corrupted by impulsive noise and invoke the filtering operation to substitute the gray level of the test pixel with a filtered gray value. Then go to step (vii).

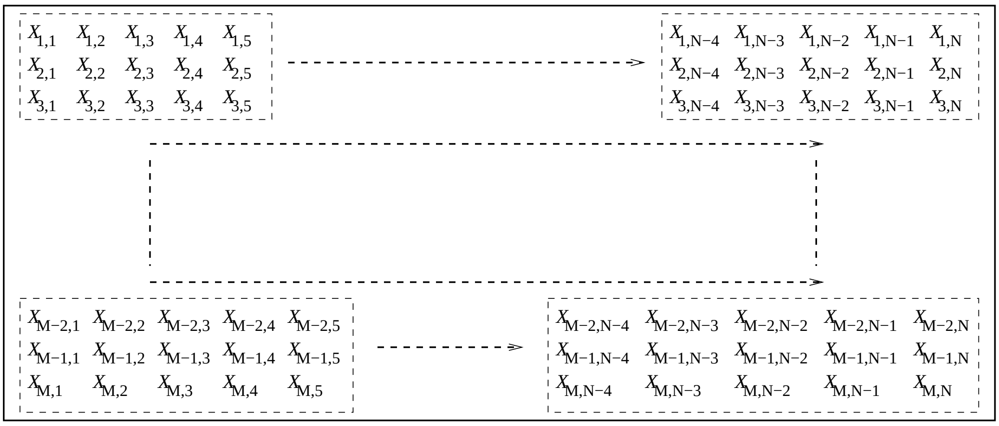

X1,1 X1,2 X1,3 X1,4 X1,5
X2,1 X2,2 X2,3 X2,4 X2,5
X3,1 X3,2 X3,3 X3,4 X3,5
X1,N−4 X1,N−3 X1,N−2 X1,N−1 X1,N
X2,N−4 X2,N−3 X2,N−2 X2,N−1 X2,N
X3,N−4 X3,N−3 X3,N−2 X3,N−1 X3,N
XM−2,1 XM−2,2 XM−2,3 XM−2,4 XM−2,5
XM−1,1 XM−1,2 XM−1,3 XM−1,4 XM−1,5
XM,1 XM,2 XM,3 XM,4 XM,5
XM−2,N−4 XM−2,N−3 XM−2,N−2 XM−2,N−1 XM−2,N
XM−1,N−4 XM−1,N−3 XM−1,N−2 XM−1,N−1 XM−1,N
XM,N−4 XM,N−3 XM,N−2 XM,N−1 XM,N

Figure 2.2: Typical window Selection for an $M \times N$ Image in DDMF

- If $d_{i,j}$ equals to one, then the test pixel is healthy. Skip the test window and go to step (vii).

vii. Shift the moving window $X^{(t)}$ by one column from left to right and top to bottom as shown in Figure 2.2.

viii. Repeat steps (ii) through (vi) for all the windows in the row.

ix. Obtain the next sliding window by shifting it down by one row.

x. Repeat the steps (ii) through (vii) till the complete image is covered.

2.1.3 Recursive Median Filtering Algorithm

Based on the algorithm corrupted pixels are identified across the image. Then the filtering operation is carried out only on those distorted pixels. The recursive filtering operation computes the median value of a 3×3 window $X^{(w)}$ surrounding the corrupted center pixel and substitutes this value at the location of the faulty pixel unlike the conventional median filter. In the next adjacent window, the healthiness of its center pixel is tested considering the gray level of the already filtered pixel rather than that of the original one. Mathematically,

$$\hat{Y}_{i,j} = \begin{cases} Y_{i,j} & \text{if } d_{i,j} = 1 \\ Z_{i,j} & \text{otherwise} \end{cases} \tag{2.7}$$

where,

$$Z_{i,j} = \text{median}\{X^{(w)}_{i-k,j-l}, (k,l) \in X^{(w)}\} \tag{2.8}$$

2.2 Fuzzy Impulsive Noise Detection

In this section, a fuzzy based filtering scheme namely Fuzzy Impulsive Noise Detection (FIND) is proposed. It employs a fuzzy detection scheme to identify pixels corrupted with impulsive noise and subsequently filter the noisy pixels using recursive median filter. The detector is responsible for ascertaining the healthiness of a pixel in a test window by utilizing the gray level information in its neighborhood. The median filtering is applied to corrupted locations only leaving the non-corrupted ones intact. Such selective filtering operation prevents from edge jittering and blurring of images.

Section 2.2.1 describes the methodology of the proposed scheme. The algorithm is outlined in Section 2.2.2.

2.2.1 Methodology

The proposed filtering scheme is a selective one and consists of two stages: decision-making regarding the presence of impulsive noise (*salt & pepper*) at a test pixel location and median filtering only of corrupted pixels. Hence, detection operation is carried out at all locations but filtering is performed only at selected locations. From the corrupted image a 3 × 3 window is selected and a fuzzy detection is employed to derive a decision regarding the presence of impulse at the center pixel. Accordingly, the center pixel is replaced with the median value of the pixels in its neighborhood prior to the selection of the next window. The overall block diagram of the combined filter structure is depicted in Figure 2.3.

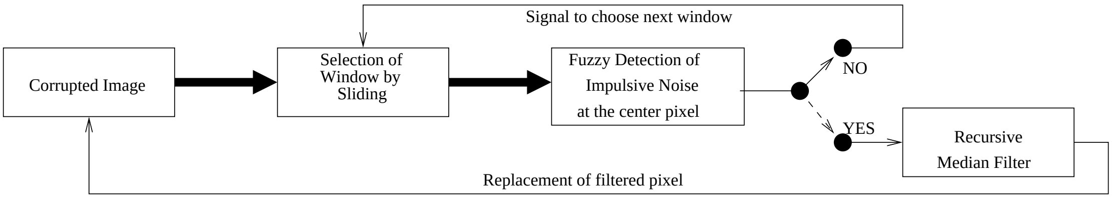

Signal to choose next window
Corrupted Image
Selection of Window by Sliding
Fuzzy Detection of Impulsive Noise at the center pixel
NO
YES
Recursive Median Filter
Replacement of filtered pixel

Figure 2.3: Block Diagram of the Proposed Filter

2.2.2 The FIND Algorithm

The impulse detection for the proposed filter is based on the fuzzy inference logic. In any fuzzy application, the challenge lies in fuzzification and defuzzification process [39]. In our case, we use a triangular fuzzy membership function that utilizes only two linguistic variables. In the following, the proposed FIND algorithm is outlined stepwise.

i. Select the first test window of size 3 × 3 from the corrupted image X and let $X_{i,j}$ represent the center pixel.

$$X_w = \begin{pmatrix} X_{i-1,j-1} & X_{i-1,j} & X_{i-1,j+1} \\ X_{i,j-1} & X_{i,j} & X_{i,j+1} \\ X_{i+1,j-1} & X_{i+1,j} & X_{i+1,j+1} \end{pmatrix} \tag{2.9}$$

ii. Convolve X_w with the two kernels (2.10) and (2.11) to obtained Δ_1 and Δ_2 respectively.

$$K_1 = \begin{pmatrix} \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & -1 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \end{pmatrix} \tag{2.10}$$

$$K_2 = \begin{pmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & -1 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{pmatrix} \tag{2.11}$$

iii. Apply Mamdani fuzzy model for two-input one-output with nine-rules as stated in Table 2.1.

Table 2.1: Fuzzy Rules for Impulse Detection

		Δ_2		
		Small	Medium	Large
Δ_1	Small	Less Noise	Less Noise	More Noise
	Medium	Less Noise	More Noise	More Noise
	Large	More Noise	More Noise	More Noise

iv. Compute the strength of each linguistic variable using triangular fuzzy membership functions as shown in Figure 2.4. Let the membership functions for Small, Medium and Large be denoted as μ_{small}, μ_{medium}, and μ_{large} respectively.

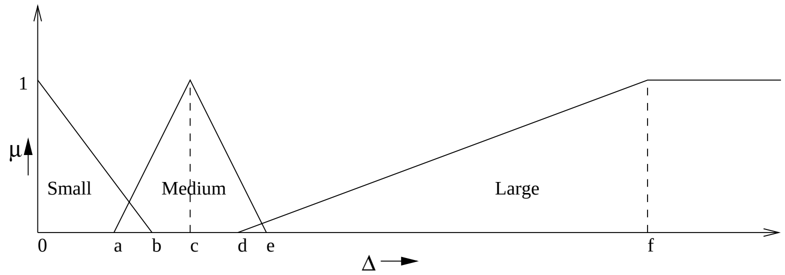


Figure 2.4: Fuzzy Membership Function

v. Evaluate the fuzzy rules using Zadeh logic for AND implication. The firing strength of membership functions to different rules are evaluated as:

$$\begin{aligned}
m_1 &= min\,[\mu_{small}(\Delta_1), \mu_{small}(\Delta_2)] \\
m_2 &= min\,[\mu_{small}(\Delta_1), \mu_{medium}(\Delta_2)] \\
&\vdots \\
m_9 &= min\,[\mu_{large}(\Delta_1), \mu_{large}(\Delta_2)]
\end{aligned} \tag{2.12}$$

vi. Construct the consequent membership function as shown in Figure 2.5 from nine active rules for a system with two inputs and one output.

vii. Obtain a crisp value (M) from the fuzzy set (Defuzzification) by using center-of-gravity method and apply to a decision process as given below.

$$d_{i,j} = \begin{cases} 0 & \text{if } M \text{ belongs to noisy class} \\ 1 & \text{otherwise} \end{cases} \tag{2.13}$$

viii. Invoke the median filtering on $X_{i,j}$ using X_w as described in Section 2.1.3, if $d_{i,j} = 0$ else go to step (viii).

ix. Shift the test window column wise and then row wise to cover the entire image pixels.

x. Repeat steps (ii) through (ix) for all windows.

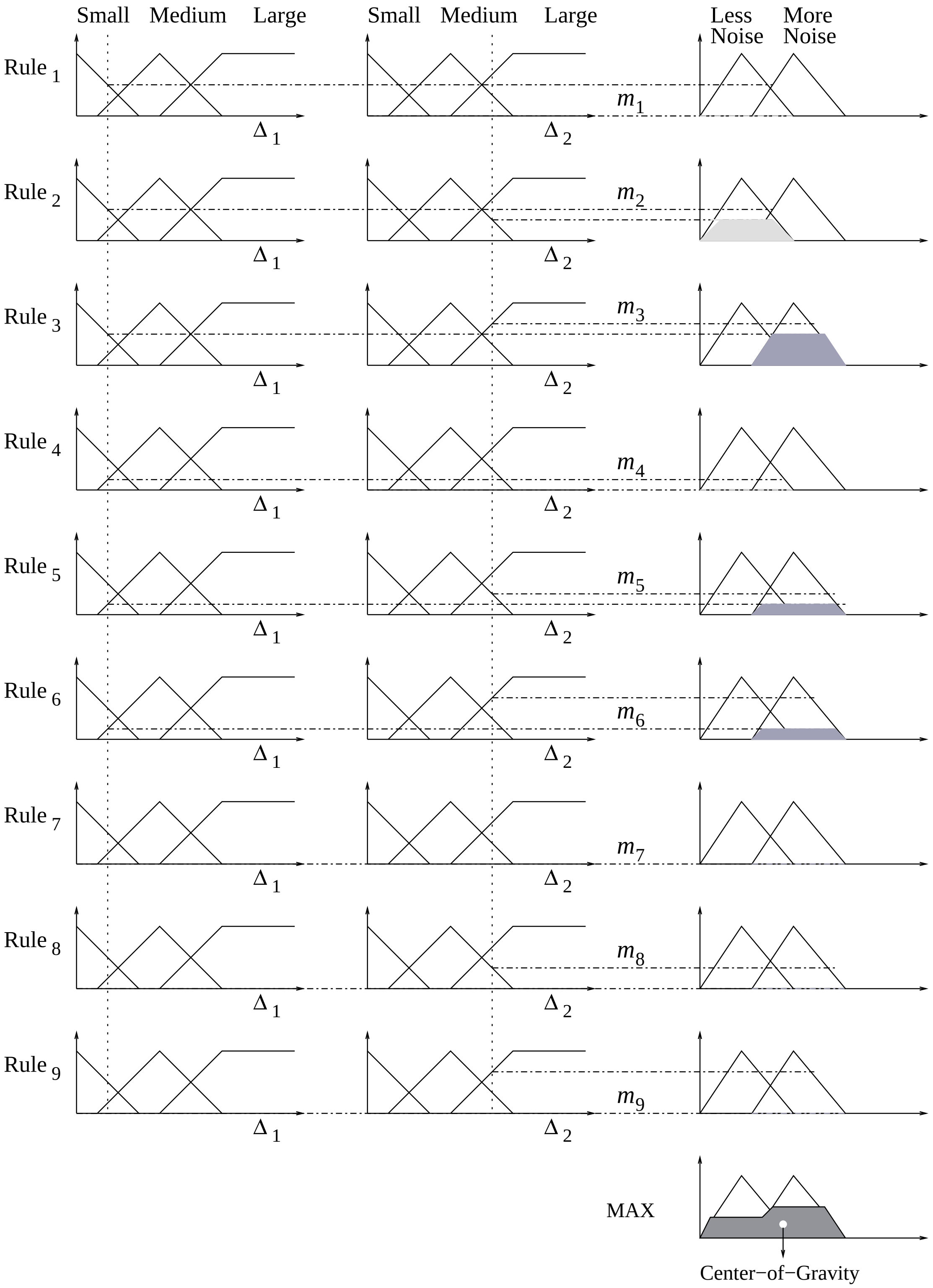


Figure 2.5: Construction of consequent membership function and defuzzification by center-of-gravity method

Table 2.2: PSNR (dB) of different schemes at 15% of noise on different images

	Lisa	*Girl*	*Clown*	*Gatlin*
SM5×5	28.66	28.43	24.78	30.70
MED3×3	34.19	30.63	22.75	31.19
MED5×5	27.71	27.41	21.75	28.45
PnV	28.97	29.47	21.79	28.32
WMed$k = 1$	29.87	28.63	23.47	28.43
WMed$k = 2$	22.85	21.76	20.70	21.84
TMED	28.26	27.29	20.45	29.84
ATMED	28.03	27.05	20.13	29.75
DDMF	40.70	38.00	34.00	42.40
FIND	30.25	32.46	25.85	35.05

2.3 Simulations and Results

The proposed schemes in this chapter are simulated and their performance is compared with some of the recently reported schemes. Median (MED(3×3)) and (MED(5×5)) [2], Switching-Median (SM(5×5)) [26], Weighted Median (WM($k = 1$)) and (WM($k = 2$)) [18], Peak and Valley (PnV) [30], TMED and ATMED [40] are the compared schemes. *Lena* image is corrupted with *Salt & Pepper Noise* with density ranging from 1% to 30%. These images are then subjected to filtering by the proposed schemes along with the above listed schemes. The PSNR (dB) (1.6) and PSP (1.7) thus obtained are plotted in Figures 2.6 and 2.7.

Similar experiments are conducted with four other standard images (*Lisa*,*Girl*, *Clown*, and *Gatlin*). Tables 2.2 and 2.3 lists the PSNR dB and PSP obtained at 15% of noise density.

Two more subjective comparisons are also done. Figure 2.8(a) is the true image of *Lena* and Figure 2.8(b) is the noisy version (20% SPN). Figures 2.8(c) to 2.8(l) shows the restored images. Similarly Figure 2.9 shows the restored images of *Peppers* corrupted with 20% of SPN.

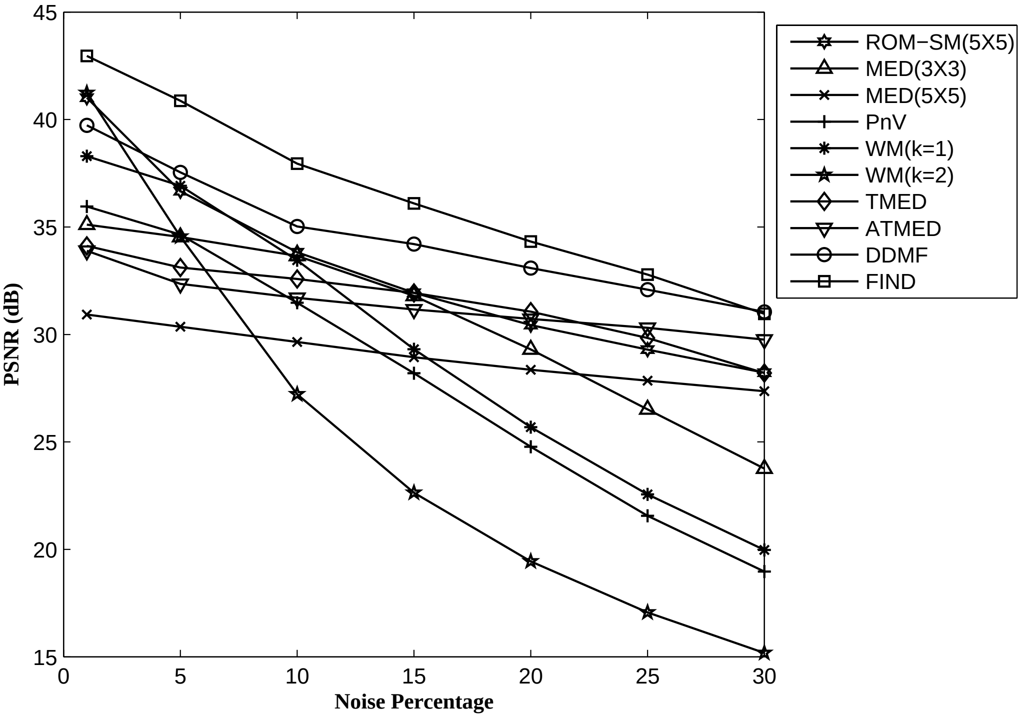


Figure 2.6: PSNR (dB) variations of Restored *Lena* image corrupted with SPN of varying strengths

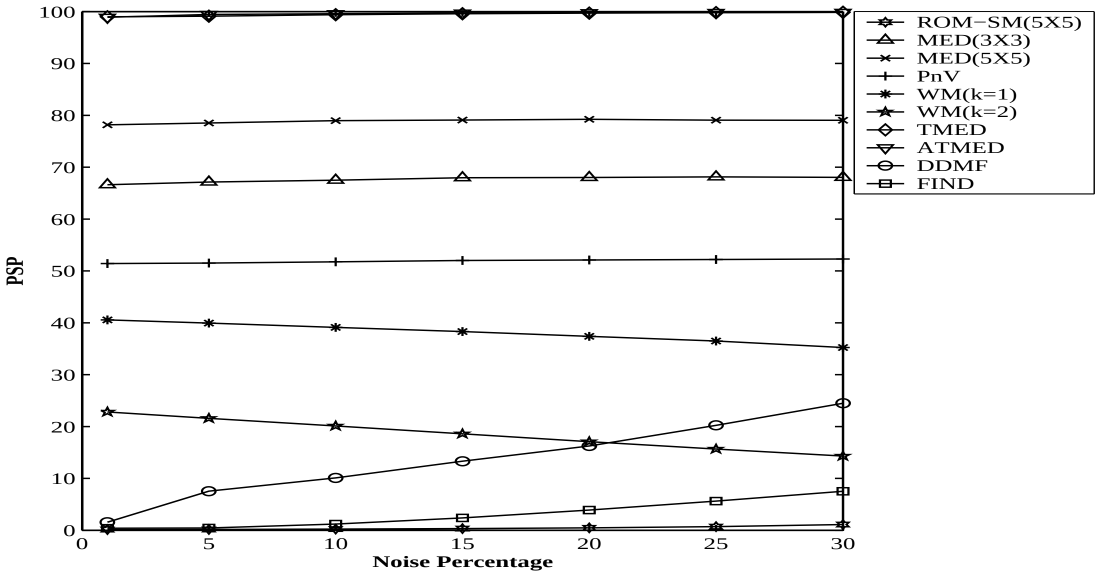
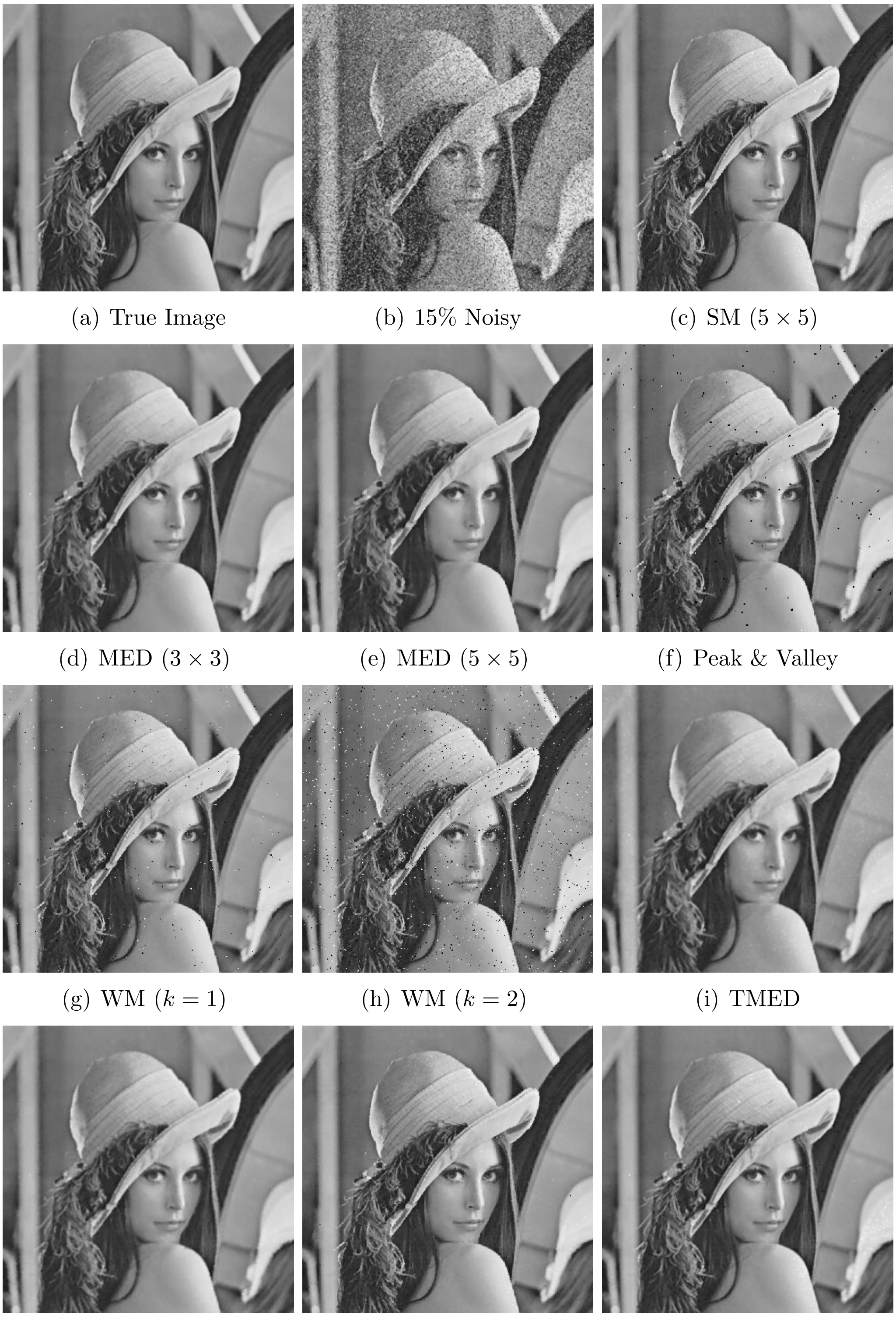


Figure 2.7: PSP variations of Restored *Lena* image corrupted with SPN of varying strengths



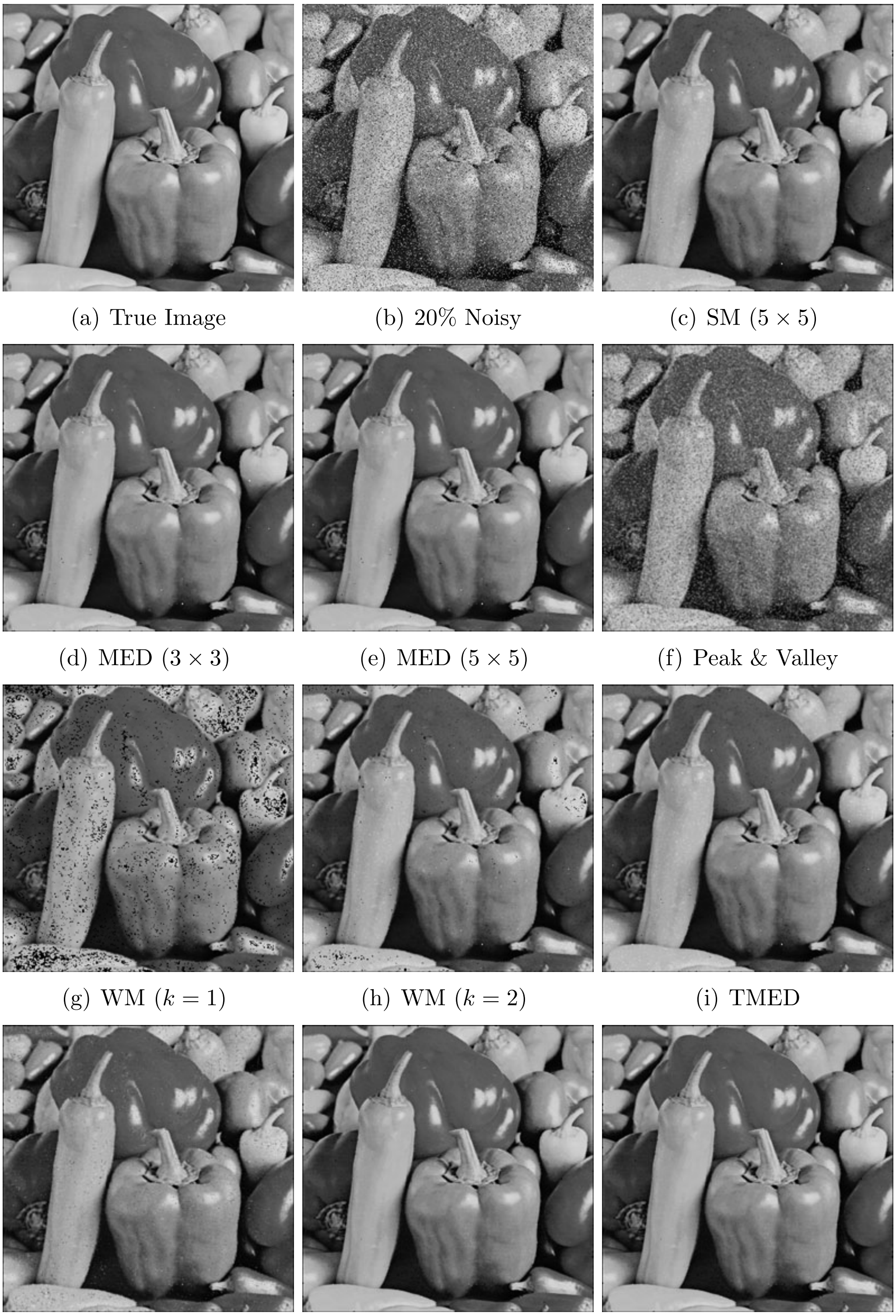
(a) True Image (b) 15% Noisy (c) SM (5×5)

(d) MED (3×3) (e) MED (5×5) (f) Peak & Valley

(g) WM $(k = 1)$ (h) WM $(k = 2)$ (i) TMED

(j) ATMED (k) DDMF (l) FIND

Figure 2.8: Impulsive Noise filtering of *Lena* image corrupted with 20% of SPN by different filters



(a) True Image (b) 20% Noisy (c) SM (5×5)

(d) MED (3×3) (e) MED (5×5) (f) Peak & Valley

(g) WM ($k = 1$) (h) WM ($k = 2$) (i) TMED

(j) ATMED (k) DDMF (l) FIND

Figure 2.9: Impulsive Noise filtering of *Peppers* image corrupted with 15% of SPN by different filters

Table 2.3: PSP of different schemes at 15% of noise on different images

	Lisa	*Girl*	*Clown*	*Gatlin*
SM5 × 5	0.12	0.41	2.67	30.70
MED3 × 3	40.00	56.28	51.98	31.19
MED5 × 5	48.85	65.02	56.46	28.45
PnV	29.36	43.84	45.98	28.32
WMed$k = 1$	18.80	32.11	31.85	28.43
WMed$k = 2$	8.19	15.42	17.85	21.84
TMED	99.59	99.46	99.57	29.84
ATMED	99.69	99.12	99.78	29.75
DDMF	6.20	11.10	10.00	42.40
FIND	1.98	2.03	2.46	35.05

2.4 Summary

Two different noise suppression approaches are proposed in this chapter. Based on the second order difference the first scheme uses a threshold to determine impulses. In the fuzzy approach, two parameters are generated from the image. These parameters are then subjected to different stages of fuzzy techniques to identify the noise location. As can be seen from the plots both the proposed schemes outperforms the existing schemes. One of limitations of these two techniques are that they use fixed value of threshold.

Chapter 3

Adaptive Threshold for Impulsive Noise Detection

This chapter and the chapter followed by this deal with *Random Valued Impulsive Noise* (RVIN) model of impulsive noise (Section 1.3). All the three schemes presented in this chapter are based on second order difference of pixels, which is described in Section 3.1. These schemes vary in the way threshold values are selected for impulse detection. The need for adaptive threshold is described in Section 3.2. The first scheme, *Second Order Differential Impulse Detector* (SODID) is discussed in Section 3.3. *ANN based Adaptive Thresholding for Impulse Detection* (ANNAT) is the second scheme presented in Section 3.4. In Section 3.5, the third scheme *FLANN based Adaptive Threshold Selection for Detecting Impulsive Noise in Images* (FLANNAT) is presented. Section 3.6 presents a comparison of the three proposed schemes with some of the existing schemes. Finally, Section 3.7 provides the summary of the chapter.

3.1 Second Order Difference of Pixels

For the sake of simplified explanation, one-dimensional derivative is focused here. The derivatives of a digital function are defined in terms of differences. The first derivative must be:

i. zero in the areas of constant gray level values i.e. flat segment,

ii. nonzero at the onset of a gray level step or ramp and along the ramp.

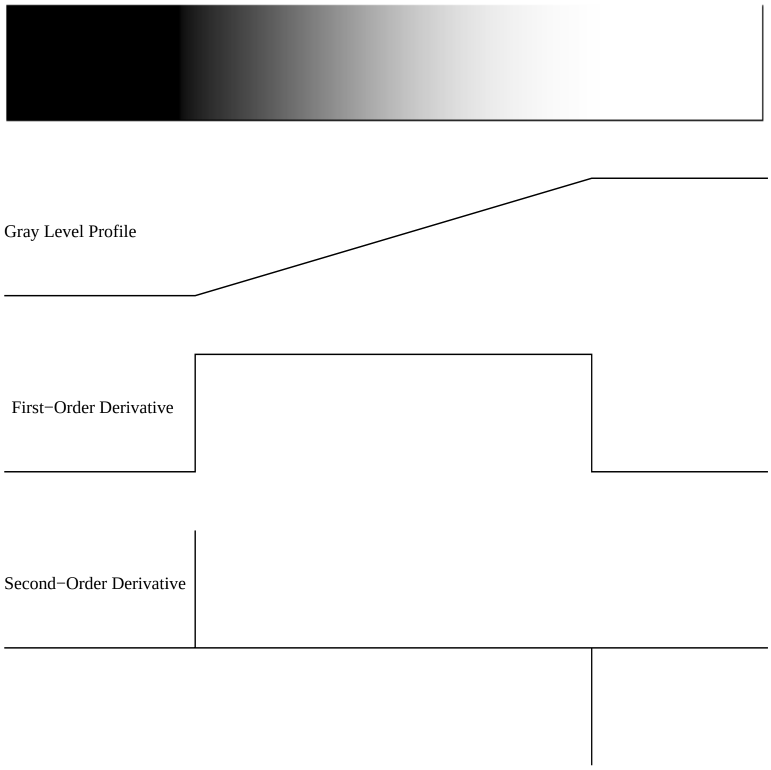


Figure 3.1: Gray level profile, first-order and second-order derivative of an image

Similarly, the second derivative must be:

i. zero in the flat areas and along ramps of constant slopes,

ii. nonzero at the onset and end of a gray level step or ramp.

As derivatives are found for digital quantities whose values are finite, the maximum possible gray level change is also is finite, and the shortest distance over which that change can occur is between adjacent pixels. First-order derivative of a one-dimensional function $f(x)$ may be defined as:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \tag{3.1}$$

Similarly, second-order derivative may be defined as:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x) \tag{3.2}$$

Figure 3.1 shows a horizontal gray level profile of the edge between two regions. Also the first and second derivatives of the gray level profile are shown in the figure.

From left to right along the profile, the first derivative is positive at the points of transition into and out of the ramp ; and is zero in the flat segment. The second derivative is zero except at the transition points [2].

This behavior of second difference is exploited in the proposed schemes. An impulse is nothing but the change in gray level profile of an image. The second difference of an impulse will result in a spike. Also there will be a spike for an edge. In order to differentiate between these two spikes, a threshold value is required. Selection of this threshold is an important task and is described in the next section.

3.1.1 Algorithm

The proposed algorithm consists of two passes described as below:

Pass One

i. Choose a window $X^{(t)}$ of size 3×5 located at the top-left corner of the observed image X.

$$X^{(t)} = \begin{pmatrix} X_{i-1,j-2} & X_{i-1,j-1} & X_{i-1,j} & X_{i-1,j+1} & X_{i-1,j+2} \\ X_{i,j-2} & X_{i,j-1} & X_{i,j} & X_{i,j+1} & X_{i,j+2} \\ X_{i+1,j-2} & X_{i+1,j-1} & X_{i+1,j} & X_{i+1,j+1} & X_{i+1,j+2} \end{pmatrix} \tag{3.3}$$

Consider a 3×3 sub-window $X^{(w)}$ from X as:

$$X^{(w)} = \begin{pmatrix} X_{i-1,j-1} & X_{i-1,j} & X_{i-1,j+1} \\ X_{i,j-1} & X_{i,j} & X_{i,j+1} \\ X_{i+1,j-1} & X_{i+1,j} & X_{i+1,j+1} \end{pmatrix} \tag{3.4}$$

ii. Compute the first order 3×4 difference matrix $f^{(d)}$ from $X^{(t)}$ as:

$$f^{(d)} = \begin{pmatrix} f^{(d)}_{i-1,j-1} & f^{(d)}_{i-1,j} & f^{(d)}_{i-1,j+1} & f^{(d)}_{i-1,j+2} \\ f^{(d)}_{i,j-1} & f^{(d)}_{i,j} & f^{(d)}_{i,j+1} & f^{(d)}_{i,j+2} \\ f^{(d)}_{i+1,j-1} & f^{(d)}_{i+1,j} & f^{(d)}_{i+1,j+1} & f^{(d)}_{i+1,j+2} \end{pmatrix} \tag{3.5}$$

where $f^{(d)}_{i+k,j+l} = X^{(t)}_{i+k,j+l} - X^{(t)}_{i+k,j+l-1}$, $k = -1, 0, 1$ and $l = -1, 0, 1, 2$.

iii. Compute the second order 3 × 3 difference matrix $s^{(d)}$ from $f^{(d)}$ as:

$$s^{(d)} = \begin{pmatrix} s^{(d)}_{i-1,j-1} & s^{(d)}_{i-1,j} & s^{(d)}_{i-1,j+1} \\ s^{(d)}_{i,j-1} & s^{(d)}_{i,j} & s^{(d)}_{i,j+1} \\ s^{(d)}_{i+1,j-1} & s^{(d)}_{i+1,j} & s^{(d)}_{i+1,j+1} \end{pmatrix} \tag{3.6}$$

where $s^{(d)}_{i+p,j+q} = f^{(d)}_{i+p,j+q+1} - f^{(d)}_{i+p,j+q}$, $p = -1, 0, 1$ and $q = -1, 0, 1$.

iv. The decision index $d_{i,j}$ at (i, j) is then computed as:

$$d_{i,j} = \begin{cases} 0 & \text{if } \left| s^{(d)}_{i,j} \right| > \theta_1 \\ 1 & \text{otherwise} \end{cases} \tag{3.7}$$

Select threshold θ_1 as described in Sections 3.2– 3.5.

v. Use median filter on the noisy pixels only to remove noise from the pixel (i, j) with the sub-window as in (3.4).

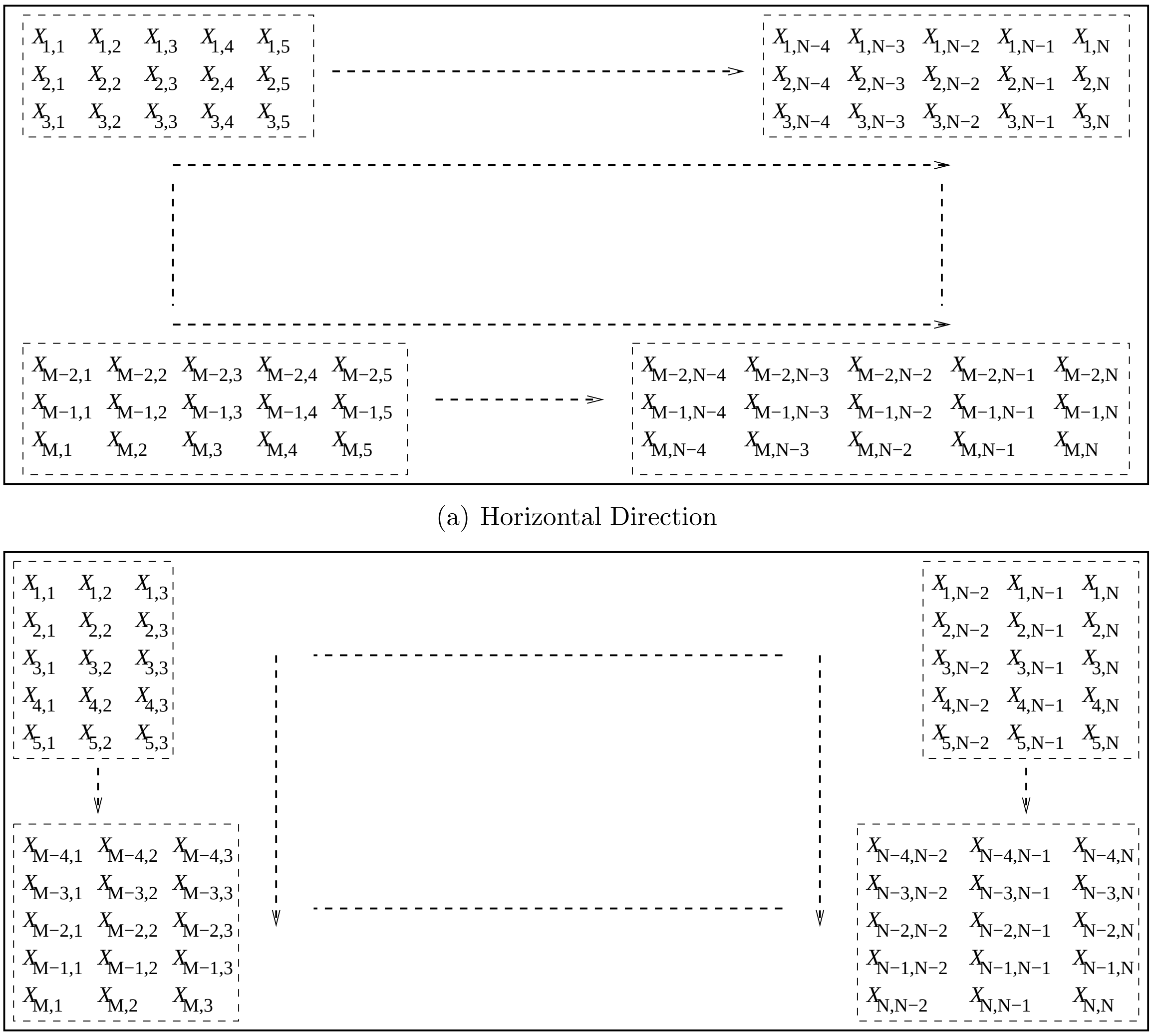
vi. Shift the window $X^{(t)}$ one by one column from left to right and top to bottom (as shown in Figure 3.2(a)) and for all windows repeat the steps (ii) through (vi).

Pass Two

i. Repeat steps (i) through (vi) of Pass One (as shown in Figure 3.2(b)) with $X^{(t)}$ order as 5×3, $f^{(d)}$ order as 4×3, and the threshold value as θ_2 in place of θ_1.

3.2 Adaptive Threshold Selection

Threshold plays an important role in deciding healthiness of a pixel. If a predefined parameter of a test pixel exceeds the threshold value, it is termed as contaminated. Further, the solution to image restoration problem depends very much on the type of image, characteristics and density of noise. Hence, there can not be one threshold value, which will be a panacea to all situations. A constant threshold value may not provide satisfactory performance for all circumstances. In other


$X_{1,1}$ $X_{1,2}$ $X_{1,3}$ $X_{1,4}$ $X_{1,5}$
$X_{2,1}$ $X_{2,2}$ $X_{2,3}$ $X_{2,4}$ $X_{2,5}$
$X_{3,1}$ $X_{3,2}$ $X_{3,3}$ $X_{3,4}$ $X_{3,5}$
$X_{1,N-4}$ $X_{1,N-3}$ $X_{1,N-2}$ $X_{1,N-1}$ $X_{1,N}$
$X_{2,N-4}$ $X_{2,N-3}$ $X_{2,N-2}$ $X_{2,N-1}$ $X_{2,N}$
$X_{3,N-4}$ $X_{3,N-3}$ $X_{3,N-2}$ $X_{3,N-1}$ $X_{3,N}$
$X_{M-2,1}$ $X_{M-2,2}$ $X_{M-2,3}$ $X_{M-2,4}$ $X_{M-2,5}$
$X_{M-1,1}$ $X_{M-1,2}$ $X_{M-1,3}$ $X_{M-1,4}$ $X_{M-1,5}$
$X_{M,1}$ $X_{M,2}$ $X_{M,3}$ $X_{M,4}$ $X_{M,5}$
$X_{M-2,N-4}$ $X_{M-2,N-3}$ $X_{M-2,N-2}$ $X_{M-2,N-1}$ $X_{M-2,N}$
$X_{M-1,N-4}$ $X_{M-1,N-3}$ $X_{M-1,N-2}$ $X_{M-1,N-1}$ $X_{M-1,N}$
$X_{M,N-4}$ $X_{M,N-3}$ $X_{M,N-2}$ $X_{M,N-1}$ $X_{M,N}$

(a) Horizontal Direction

$X_{1,1}$ $X_{1,2}$ $X_{1,3}$
$X_{2,1}$ $X_{2,2}$ $X_{2,3}$
$X_{3,1}$ $X_{3,2}$ $X_{3,3}$
$X_{4,1}$ $X_{4,2}$ $X_{4,3}$
$X_{5,1}$ $X_{5,2}$ $X_{5,3}$
$X_{1,N-2}$ $X_{1,N-1}$ $X_{1,N}$
$X_{2,N-2}$ $X_{2,N-1}$ $X_{2,N}$
$X_{3,N-2}$ $X_{3,N-1}$ $X_{3,N}$
$X_{4,N-2}$ $X_{4,N-1}$ $X_{4,N}$
$X_{5,N-2}$ $X_{5,N-1}$ $X_{5,N}$
$X_{M-4,1}$ $X_{M-4,2}$ $X_{M-4,3}$
$X_{M-3,1}$ $X_{M-3,2}$ $X_{M-3,3}$
$X_{M-2,1}$ $X_{M-2,2}$ $X_{M-2,3}$
$X_{M-1,1}$ $X_{M-1,2}$ $X_{M-1,3}$
$X_{M,1}$ $X_{M,2}$ $X_{M,3}$
$X_{N-4,N-2}$ $X_{N-4,N-1}$ $X_{N-4,N}$
$X_{N-3,N-2}$ $X_{N-3,N-1}$ $X_{N-3,N}$
$X_{N-2,N-2}$ $X_{N-2,N-1}$ $X_{N-2,N}$
$X_{N-1,N-2}$ $X_{N-1,N-1}$ $X_{N-1,N}$
$X_{N,N-2}$ $X_{N,N-1}$ $X_{N,N}$

(b) Vertical Direction

Figure 3.2: Window Selection for an $M \times N$ Image

words, threshold selection should be carefully done and preferably it should be an adaptive one. It should adapt to its environment dynamically and give an optimal value. In next three sections three different ways of selecting threshold are presented.

As said earlier, threshold depends on its environment. Environment means, the type of image, characteristic of noise and its density. In order to select a threshold, some of the image parameters are required. Steps for selecting image parameters that represents an image aptly are described below.

For any given image and at a particular noise condition the threshold value θ is varied in a wide range to obtain a set of mean squared error (MSE) values in order to establish a relation. For example:

i. An image (say *Lena*) is corrupted with impulsive noise of densities 1%, 5%, 10%, 15%, 20%, 25%, and 30%.

ii. The first noisy image $Lena_1$ (the subscript is for 1% of noise) is subjected to the proposed algorithm outlined in Section 3.1.1 by varying the threshold value θ between 0 and 1.

iii. Corresponding to each θ one mean squared error (MSE) is obtained. The minimum among those MSEs is recorded as $\text{MSE}_{min}^{(Lena_1)}$. Also the corresponding threshold value is recorded as optimal threshold value θ_{opt}.

iv. Steps (ii) and (iii) are repeated for other noisy *Lena*, i.e. $Lena_i$, $i \in \{5, 10, 15, 20, 25, 30\}$.

v. Repeat steps (i) to (iv) for other standard images like *Lisa*, *House*, *Peppers* etc.

Figures 3.3(a), 3.3(b), 3.3(c) and 3.3(d) show the relation between θ_{opt} and MSE_{min} for *Lena*, *Lisa*, *House* and *Peppers* images respectively.

From these plots (Figure 3.3) it is, in general, observed that the minimum MSE and the corresponding threshold bear an exponentially decaying relation. This is true for all other images. In a practical situation, the use of MSE or noise ratio to predict the threshold is ruled out as they need knowledge of the original image for computation. However, to alleviate this problem analysis have been made as follows. The minimum MSE is inversely proportional to optimal threshold value i.e.

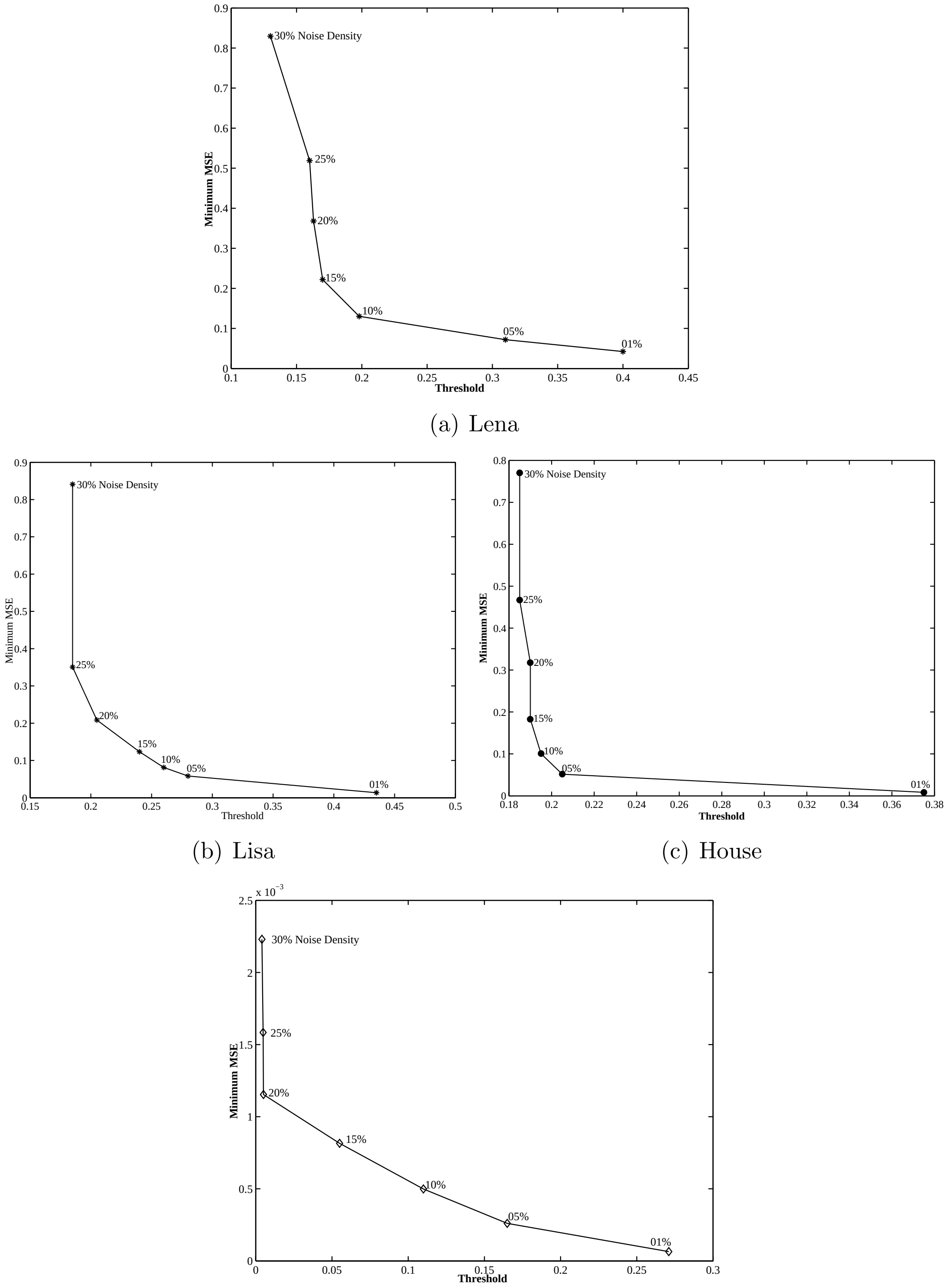
$$\text{MSE}_{min} \propto \frac{1}{\theta_{opt}} \tag{3.8}$$

also the noise percentage is inversely proportional to optimal threshold value, given as:

$$\eta \propto \frac{1}{\theta_{opt}} \tag{3.9}$$

where, η is the noise percentage. Also it is known that:

$$\eta \propto \sigma^2 \tag{3.10}$$



(a) Lena

(b) Lisa

(c) House

(d) Peppers

Figure 3.3: Variation of Minimum MSE at different Threshold values

and,

$$\eta \propto \mu \tag{3.11}$$

where, μ and σ^2 are the mean and variance of the noisy image respectively.

From the above four equations it may be established that both the mean and variance of a noisy image are proportional to the MSE$_{min}$ and hence these two parameters may be used for threshold selection.

3.3 Second Order Differential Impulse Detector

Using the assumptions in Section 3.2 let the plots of Figure 3.3 be represented as:

$$\theta = Ae^{-B\sigma^2} + C \tag{3.12}$$

where, θ is the optimum threshold value corresponding to MSE$_{min}$,
σ^2 is variance of the noisy image,
A, B and C are constants to be evaluated.

By the steps described in Section 3.2, we can have a set of parameters such as σ^2 and corresponding θ. Let x_i denote the σ^2 values and y_i denote the corresponding θ values.
The equation in 3.12 may otherwise be written as:

$$y = ae^{bx} + c \tag{3.13}$$

To get the value of c consider three pairs of (x_i, y_i) i.e. (x_1, y_1), (x_2, y_2) and (x_3, y_3) such that the three abscissas are in arithmetic progression.
By putting these values on (3.13)

$$y_1 = ae^{bx_1} + c \Rightarrow y_1 - c = ae^{bx_1} \Rightarrow y_1 - c = ak^{x_1}$$
$$y_2 = ae^{bx_2} + c \Rightarrow y_2 - c = ae^{bx_2} \Rightarrow y_2 - c = ak^{x_2}$$
$$y_3 = ae^{bx_3} + c \Rightarrow y_3 - c = ae^{bx_3} \Rightarrow y_3 - c = ak^{x_3}$$

where, $k = e^b$.

By multiplying the first and third terms:

$$\begin{aligned}(y_1-c)(y_3-c) &= ak^{x_1}\cdot ak^{x_3}\\ &= a^2\cdot k^{x_1+x_3}\\ &= a^2\cdot k^{2x_2}\\ &= (a\cdot k^{x_2})^2\\ &= (y_2-c)^2\\ \Rightarrow y_1y_3-cy_3-cy_1+c^2 &= y_2^2-2y_1y_3+c^2\\ \Rightarrow c &= \frac{y_1y_3-y_2^2}{y_1+y_3-2y_2} \qquad (3.14)\end{aligned}$$

Rewriting the Equation 3.13 as:

$$\begin{aligned}y &= ae^{bx}+c\\ \Rightarrow y-c &= ak^x\\ \Rightarrow \ln(y-c) &= \ln(ak^x)\\ \Rightarrow \ln(y-c) &= \ln(a)+x\ln(k)\\ \Rightarrow Y &= A+Bx \qquad (3.15)\end{aligned}$$

where, $Y=\ln(y-c)$, $A=\ln(a)$ and $B=\ln(k)$.

Now the nonlinear equation (3.13) is represented as a linear equation.

The abscissas y_i can be written in the form of Y_i.

Divide the set of values of (x_i, Y_i) into two halves.

Find $\bar{x}_1, \bar{x}_2, \bar{Y}_1, \bar{Y}_2$ and then fit them onto:

$$\begin{aligned}\bar{Y}_1 &= A+B\bar{x}_1\\ \bar{Y}_2 &= A+B\bar{x}_2 \qquad (3.16)\end{aligned}$$

Values of A and B can be found by solving the above two simultaneous equations. Values of a and b can be found by putting A and B values in the Equation 3.15.

$$\begin{aligned}A &= \ln(a)\\ \Rightarrow a &= e^A \qquad (3.17)\end{aligned}$$

$$\begin{aligned} B &= \ln(k) \\ &= \ln(e^b) \\ \Rightarrow b &= B \end{aligned} \tag{3.18}$$

Hence, by using Equations 3.14, 3.17 and 3.18 the curve (3.13) can be fitted.

The values of A, B and C are obtained and the curve for threshold θ may therefore be expressed as:

$$\theta = (-0.0036 - j0.1114)e^{(-23.3371+j77.0754)\sigma^2} + 0.2540 \tag{3.19}$$

Similarly the threshold value θ_1 used during the second pass of the proposed scheme for noise detection and filtering in vertical direction may be estimated by observing the minimum MSE of an image at different noise conditions. It is experimentally verified that θ_1 is directly proportional to θ and therefore may be expressed as:

$$\theta_1 = k\theta \tag{3.20}$$

where k is experimentally found to be 1.42.

3.4 ANN based Adaptive Thresholding for Impulse Detection

Artificial Neural Network (ANN) is a massively parallel distributed processor. It has a natural tendency to store knowledge and make them available for further use. ANN serves as a potential tool in numerous applications. The ANN based signal detection and filtering schemes are robust, accurate and work well under nonlinear situations [3].

An artificial neuron receives inputs from a number of other neurons or from external stimulus. A weighted sum of these inputs constitutes the arguments to a nonlinear activation function. The resulting value of the activation function is the output of the neuron. This output gets distributed along weighted connections to other neurons. The actual manner in which these connections are made defines the flow of information in the network and called architecture of the ANN. The method used to adjust the weights is the process of training the network is called the learning rule. The learning may be supervised or unsupervised [3]. Genuine neural networks are those with at least two layers of neurons—a hidden layer and an output layer. The hidden layer neurons should have nonlinear and differentiable activation functions. The nonlinear activation functions enable a neural network to be a universal approximator. The problem of representation is solved by the nonlinear activation functions [41].

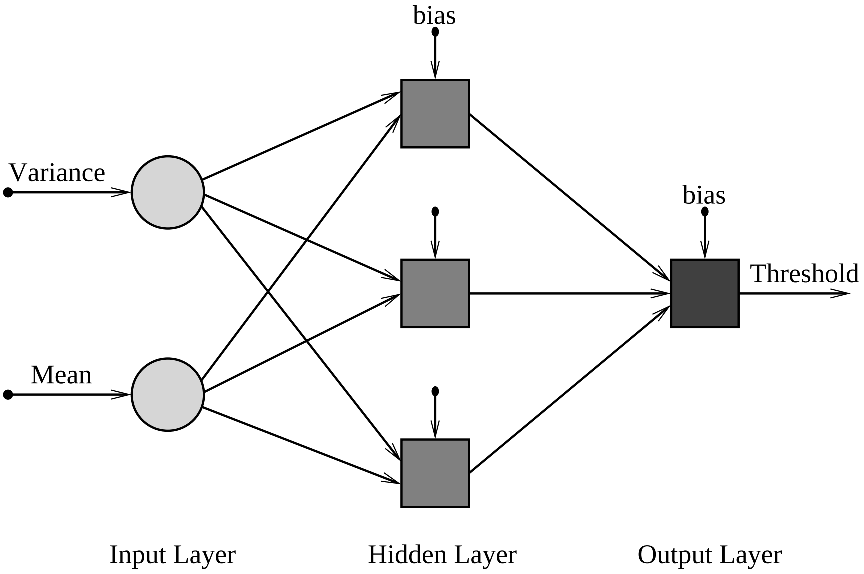


Figure 3.4: Multi-Layer Perceptron Structure of Threshold (θ_1) Estimator.

Here in this section a simple 2–3–1 ANN (Figure 3.4) is used to adapt the image environment and to provide an optimal threshold value for impulsive noise detection. Both the noisy image characteristics (Section 3.2) mean (μ) and variance (σ^2) of *Lisa*, *House*, *Gatlin* and *Peppers* images are obtained. These two parameters along with corresponding θ_{opt} of these four images are used here to train the suggested neural network using the conventional Backpropagation algorithm. μ and σ^2 of the noisy image are the two inputs to the network and θ_{opt} is the target output of the network. The training convergence characteristics obtained is shown in Figure 3.5. The neural network with trained weights are used to obtained threshold subsequently. It is seen that the network predicts an accurate threshold for images that are not used for training as well.

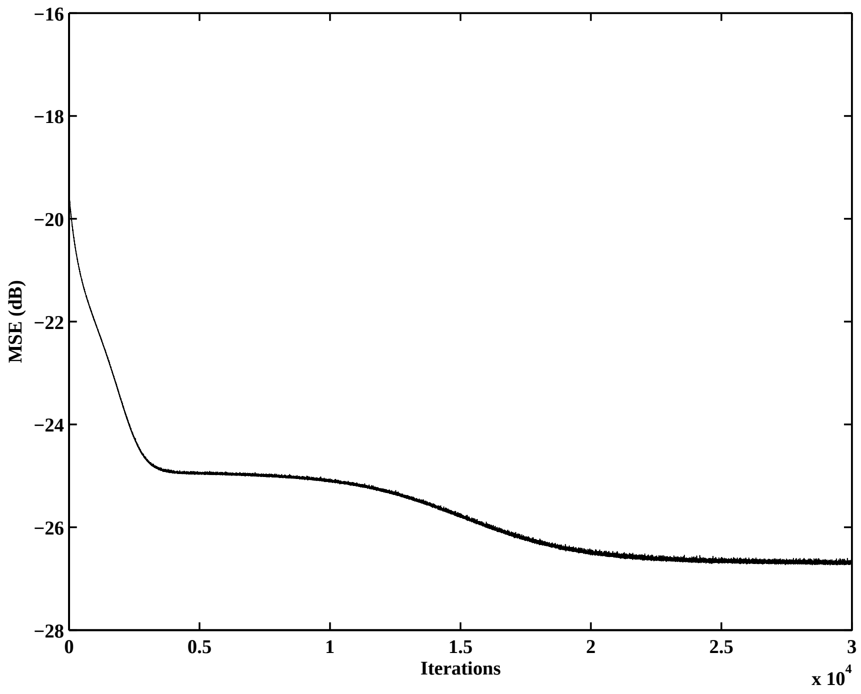


Figure 3.5: Convergence Characteristics of the Network

The threshold value thus obtained is used in the first pass of the algorithm. The output image of the first pass is then subjected to second pass of the algorithm. In the second pass a different θ is used. Mean and variance of the output image of the first pass is first calculated. These two values are then fed to the network to get the new threshold value.

3.5 FLANN based Adaptive Threshold Selection for Detecting Impulsive Noise in Images

Another variation of artificial neural network is the Functional Link Artificial Neural Network (FLANN). It is basically a flat net and the need of the hidden layer is removed. Training of FLANN by backpropagation algorithm becomes very simple. Also the network has lesser computational load and faster convergence rate than multilayer perceptrons. The functional expansion effectively increases the dimensionality of the input vector and hence the hyperplanes generated by the FLANN provides greater discrimination capability in the input pattern space [42, 43].

The proposed detector is shown in Figure 3.6. It is a two layers structure. The parameters used in this training are same as that of previous section. Mean and variance are the two inputs functionally expanded in the input layer with the trigonometric polynomial basis functions given by:

$$\{1, \mu, sin(\pi\mu), \cdots, sin(N\pi\mu), cos(\pi\mu), \cdots, cos(N\pi\mu),$$

$$\sigma^2, sin(\pi\sigma^2), \cdots, sin(N\pi\sigma^2), cos(\pi\sigma^2), \cdots, cos(N\pi\sigma^2)\}$$

In order to calculate the error, the actual output on the output layer is compared with the desired output. Depending on this error value, the weight matrix between the input–output layers is updated using backpropagation learning algorithm. The training convergence characteristics of the network is shown in Fig. 3.7.

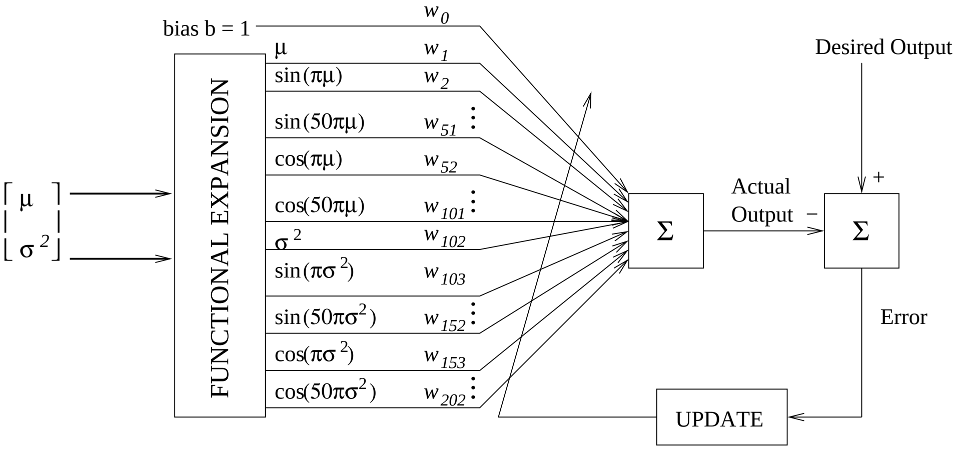


Figure 3.6: Functional Link Artificial Neural Network (FLANN) Structure for Threshold Estimation

This threshold value is used in the first pass of the algorithm to detect impulses in the horizontal direction. The filtered image obtained after the first pass is then subjected to second pass of the algorithm, where impulses are detected in vertical fashion. In the second pass a different θ is used. Using the mean and variance of the output image of the first pass new threshold value for the second pass is computed.

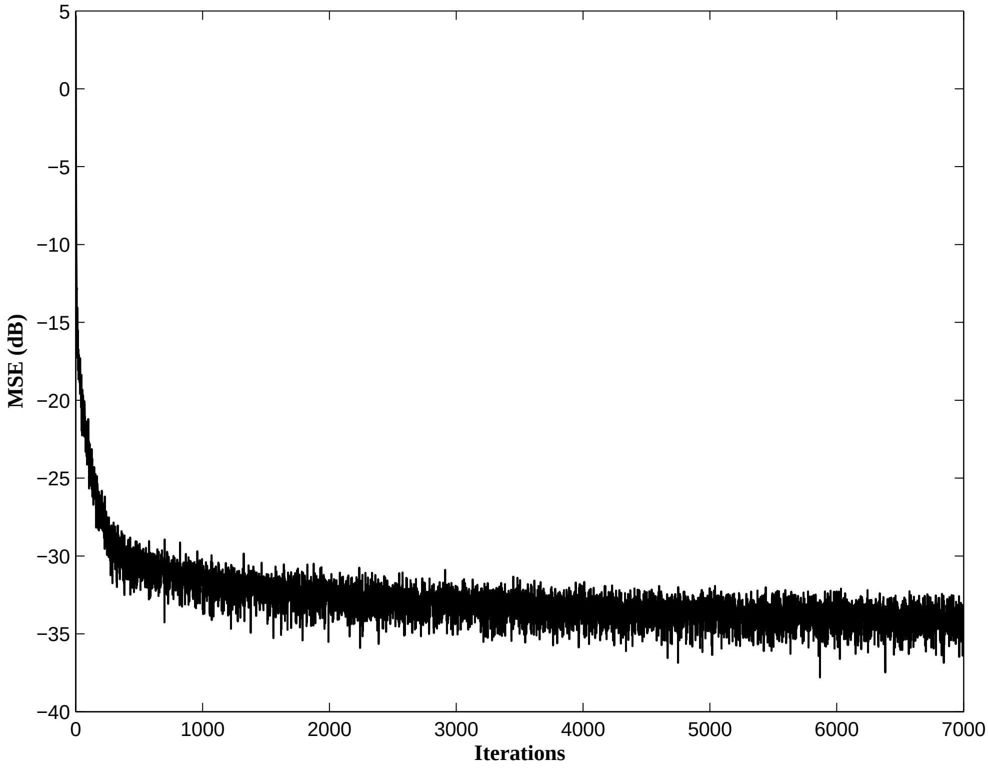


Figure 3.7: Convergence Characteristics of the Network

3.6 Simulations and Results

The three proposed schemes are simulated with some of the best performing schemes reviewed in Section 1.5. Adaptive Two-Pass Median filter (2-Pass) [24], Adaptive Center Weighted Median Filter (ACWMF) [19], Signal Dependent-Rank Ordered Mean (SD-ROM) [32], Tri-State Median (TSM) [33] and Pixel Wise MAD (PWMAD) [28] are used for comparison. *Lena* image is corrupted with *Random Valued Impulsive Noise* of 1% to 30% noise densities. These noisy images are subjected to filtering by the three proposed schemes (SODID, ANNAT and FLANNAT) along with the above five existing schemes. The PSNR (in *dB*) and PSP (in percentage) thus obtained are plotted in Figures 3.10 and 3.11.

Similarly, simulations are conducted with other standard images like *Lisa*,*Girl*, *Clown*, *Gatlin*, *Bridge*, *Boat* and *Peppers*. Table 3.1 lists the PSNR obtained at 15% and 20% of RVIN. Another listing is shown in Table 3.2 of PSP at the same noise densities.

Table 3.1: PSNR (dB) of different adaptive schemes at 15% and 20% of noise on different images

		Lisa	*Girl*	*Clown*	*Gatlin*	*Bridge*	*Boat*	*Peppers*
15% RVIN	2Pass	31.34	29.62	22.84	31.59	25.77	29.33	31.55
	ACWMF	31.78	30.04	22.56	31.62	25.36	28.87	32.98
	PWMAD	30.50	29.28	23.02	30.66	26.07	29.25	31.29
	SD-ROM	31.98	31.31	24.33	32.77	27.48	30.75	32.00
	TSM	32.05	31.05	23.88	32.46	27.08	30.51	33.04
	SODID	30.71	30.19	24.52	31.71	26.76	29.59	31.87
	ANNAT	29.86	30.20	22.97	31.67	26.68	29.98	32.04
	FLANNAT	31.79	30.17	23.59	31.83	17.42	28.93	32.15
20% RVIN	2Pass	30.46	28.41	22.25	30.45	25.09	28.32	30.14
	ACWMF	30.97	29.92	23.56	31.34	24.82	28.10	31.50
	PWMAD	28.26	27.73	22.16	28.77	24.99	27.57	29.04
	SD-ROM	30.86	29.92	23.59	31.51	26.55	29.50	31.40
	TSM	30.95	29.69	23.28	31.30	26.28	29.35	31.41
	SODID	28.89	28.66	22.82	30.14	25.82	28.41	30.24
	ANNAT	29.90	29.11	22.38	30.67	25.83	28.81	30.55
	FLANNAT	30.37	28.14	21.69	29.49	16.64	28.18	30.48

Two subjective comparisons are also made in Figures 3.8 and 3.9. The former figure shows the restored images of *Lena* corrupted with 15% of noise density and the later one shows restored images of *Peppers* corrupted with a noise density of 20%.

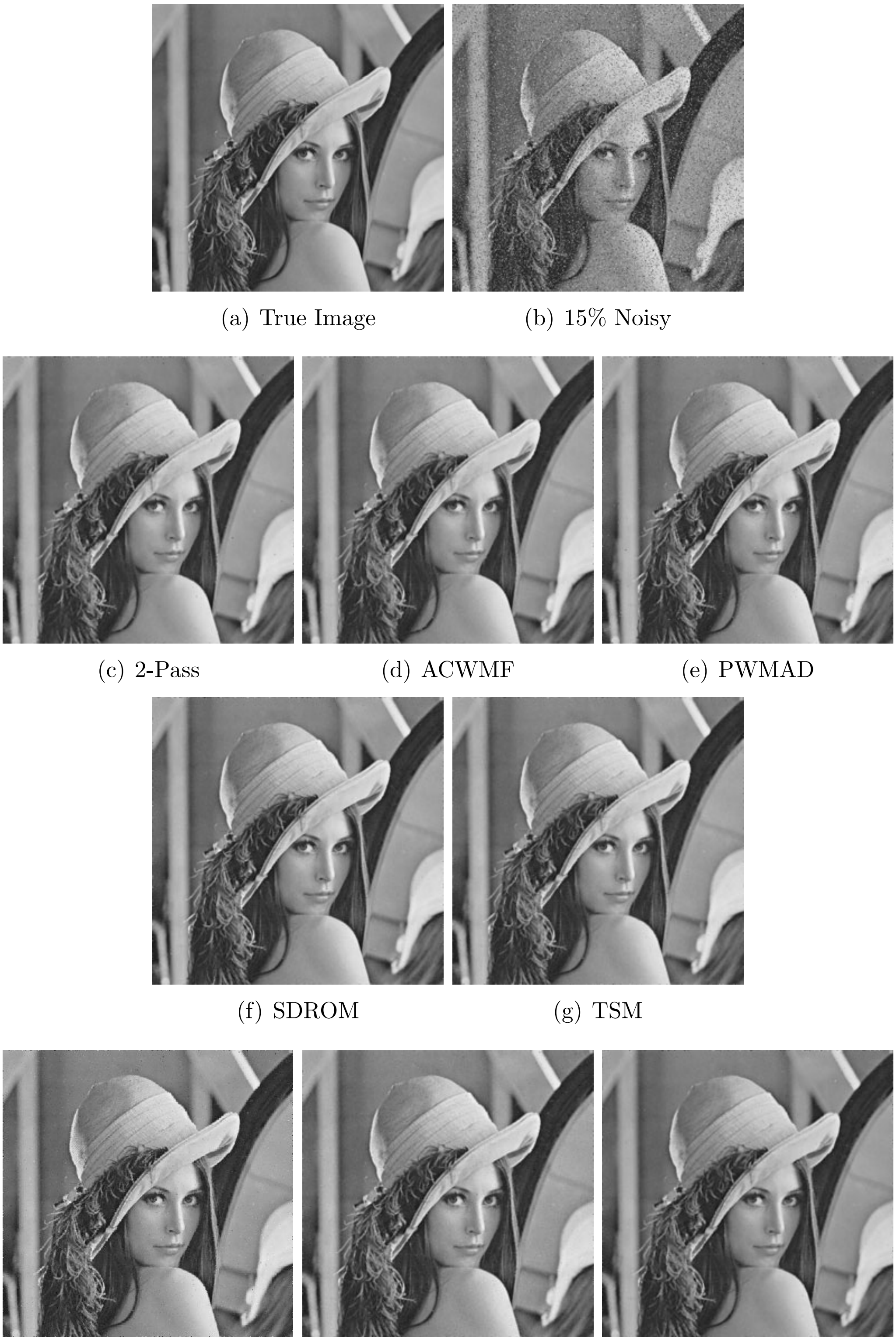
The performance of the proposed schemes in terms of $PSNR(dB)$ are better than most of the schemes except SDROM and TSM. However, the three proposed schemes are computationally better than the above two techniques (listed in Table 3.3). This is verified by simulating the schemes in MatLab 6.5, Microsoft Windows XP (SP2) Operating System and Intel Pentium IV–2.40GHz with 512MB of RAM.

Table 3.2: PSP of different adaptive schemes at 15% and 20% of noise on different images

		Lisa	*Girl*	*Clown*	*Gatlin*	*Bridge*	*Boat*	*Peppers*
15% RVIN	2Pass	35.23	51.76	51.95	31.75	54.73	67.30	67.41
	ACWMF	6.93	13.29	38.68	5.57	27.98	15.27	0.55
	PWMAD	7.35	11.05	24.85	4.00	15.83	13.95	10.53
	SD-ROM	0.14	0.93	10.29	0.39	3.06	1.04	0.34
	TSM	0.22	2.62	17.51	0.98	7.94	2.55	0.57
	SODID	4.84	11.56	19.94	8.34	15.54	13.00	12.16
	ANNAT	4.65	11.16	36.87	8.11	15.36	11.16	10.55
	FLANNAT	11.82	62.63	54.04	13.11	0.01	73.98	23.65
20% RVIN	2Pass	35.67	58.10	52.98	33.24	55.71	67.50	67.45
	ACWMF	0.60	0.43	12.60	0.41	28.69	15.80	0.57
	PWMAD	7.06	7.82	22.78	5.03	15.10	13.18	10.05
	SD-ROM	0.18	0.40	9.86	0.33	3.10	1.11	0.37
	TSM	0.31	1.21	21.07	1.00	8.2986	2.82	0.73
	SODID	6.52	16.74	24.37	10.10	18.75	16.85	15.92
	ANNAT	6.25	17.84	42.46	10.10	18.50	14.70	14.11
	FLANNAT	40.17	65.59	56.85	12.38	0.06	74.21	55.98

Table 3.3: Computational time consumed by different Schemes for removing impulsive noise from *Lena* image corrupted with 15% of RVIN

Scheme	Time (sec)
2-PASS	151.27
ACWMF	416.03
PWMAD	244.68
SDROM	11.20
TSM	77.14
SODID	10.86
ANNAT	11.97
FLANNAT	11.16



(a) True Image

(b) 15% Noisy

(c) 2-Pass

(d) ACWMF

(e) PWMAD

(f) SDROM

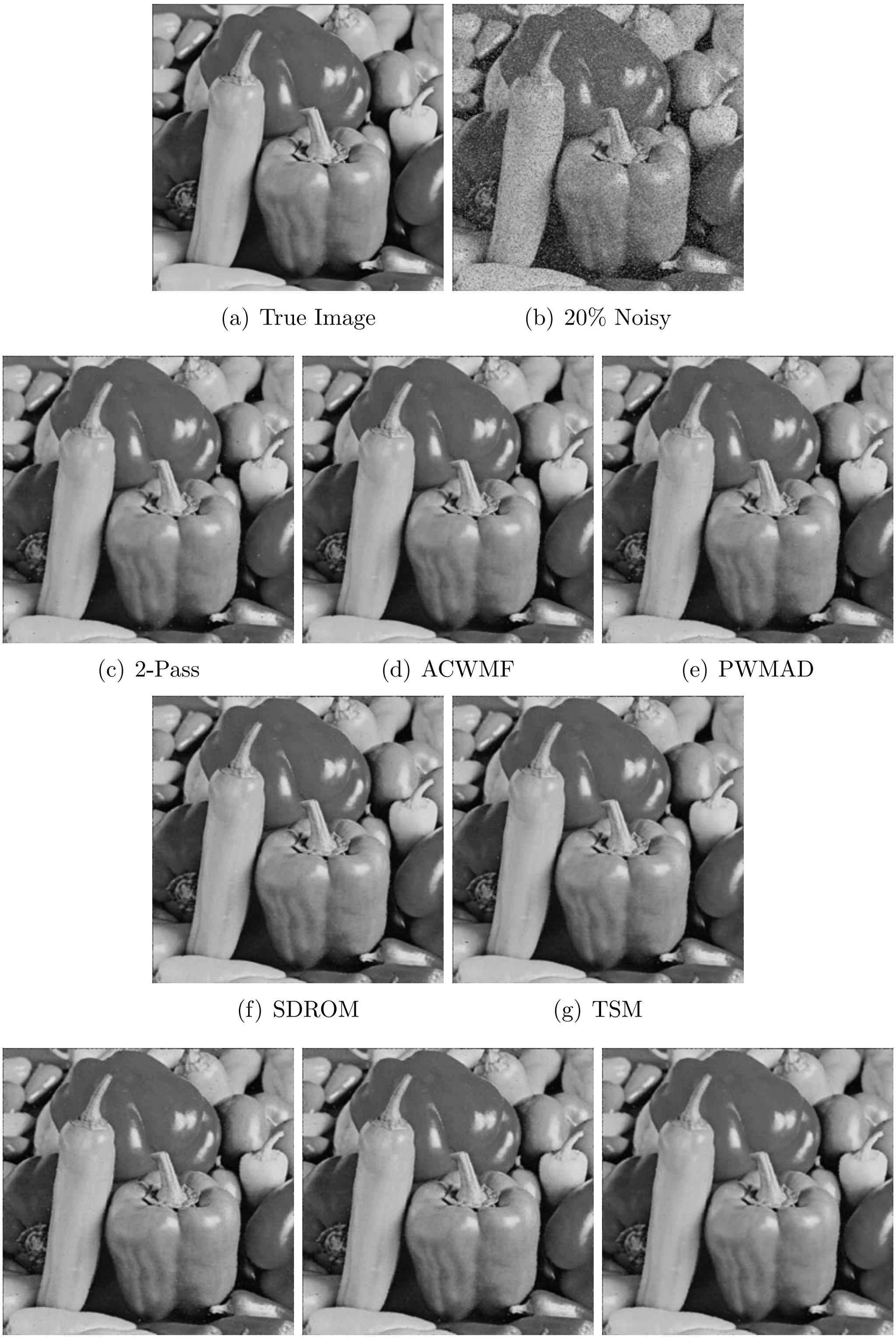
(g) TSM

(h) SODID

(i) ANNAT

(j) FLANNAT

Figure 3.8: Impulsive Noise filtering of *Lena* image corrupted with 15% of RVIN by different adaptive threshold schemes



(a) True Image (b) 20% Noisy

(c) 2-Pass (d) ACWMF (e) PWMAD

(f) SDROM (g) TSM

(h) SODID (i) ANNAT (j) FLANNAT

Figure 3.9: Impulsive Noise filtering of *Peppers* image corrupted with 20% of RVIN by different adaptive threshold schemes

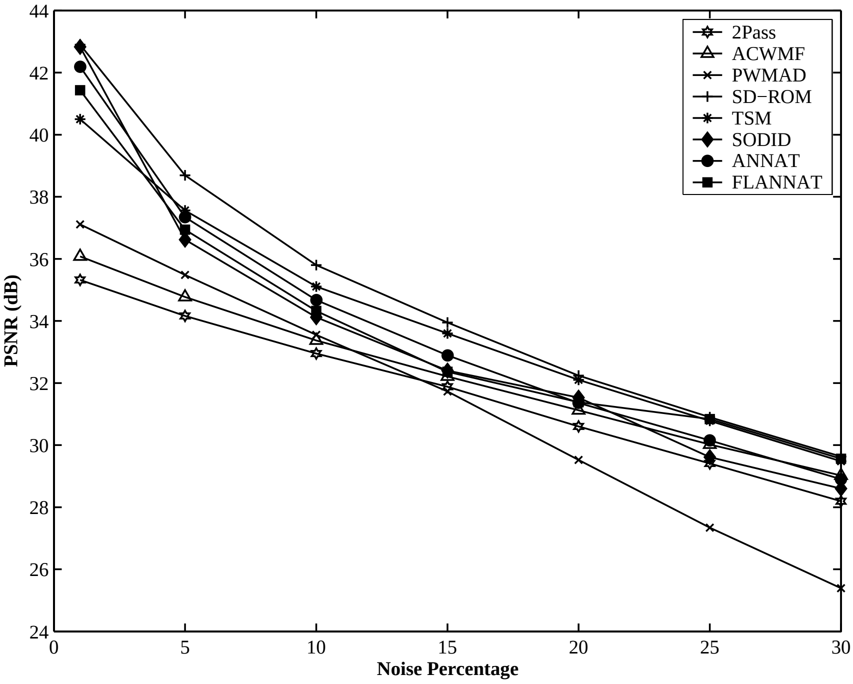


Figure 3.10: PSNR (*dB*) variations of Restored *Lena* image corrupted with RVIN of varying strengths by different adaptive threshold schemes

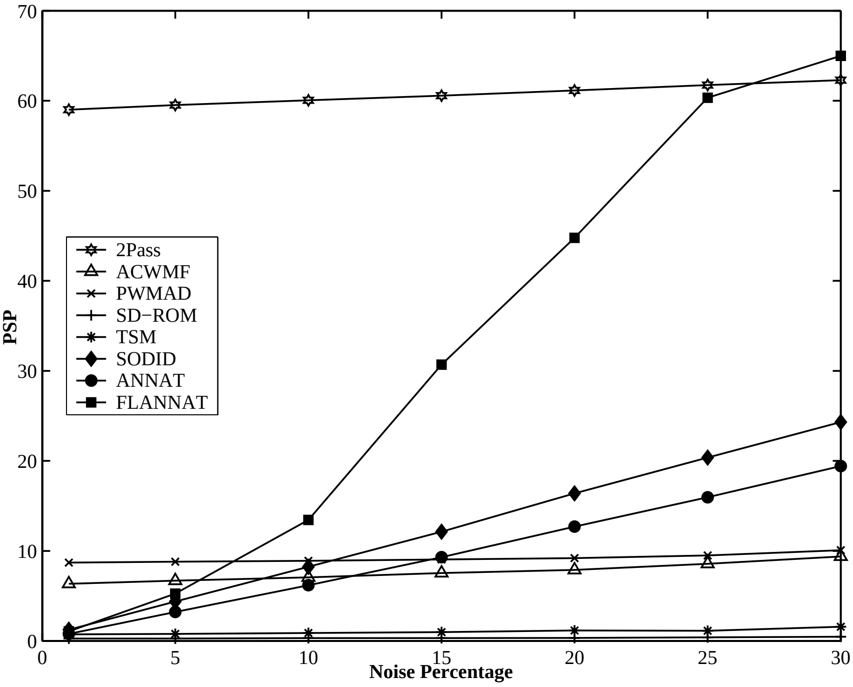


Figure 3.11: PSP variations of Restored *Lena* image corrupted with RVIN of varying strengths by different adaptive threshold schemes

3.7 Summary

Three different ways of determining the threshold values are presented. The first scheme calculate the threshold using an equation. Even though it provides good results, it needs the true image before hand, which is impractical. This problem is overcome easily by using neural network. A variation of neural network, FLANN is also used to determine the threshold value. The proposed schemes' performances are poor when compared with some of the schemes. However, computationally the proposed schemes are well off.

Chapter 4

Improved Adaptive Impulsive Noise Suppression

In this chapter an improved scheme for suppressing random valued impulsive noise of varying strengths from corrupted images is proposed. The threshold value suggested here for the detection of noisy pixels is an adaptive one, which is derived from an artificial neural network. Emphasis is put on right kind of input and the training patterns. With appropriate choice of patterns the assiduous task of training has become effortless and also the detection of noise has become reliable. The proposed scheme is presented in Section 4.1. Competent schemes are compared and analyzed with the proposed scheme with competent schemes in Section 4.2. Finally Section 4.3 summarizes the chapter.

4.1 Improved Adaptive Noise Suppression

Many different techniques are used to determine whether a given pixel is affected with impulses. Some of these techniques are relatively simple, on other hand some others are complex. Whatever may be the technique, they first identify the location of corruption and apply some filtering mechanism on those identified pixels.

The proposed scheme is also one such scheme that is based on estimation–suppression strategy. It decides the sanctity of a pixel based on the output of an artificial neural network (ANN). The network is first trained with a set of training patterns. Two different parameters are used for training. The first parameter is the pixel wise median of the absolute deviation from the median [28] and the second parameter is the rank ordered absolute difference [44]. The first parameter helps in separating the noisy image from the image details, where as the second parameter concentrates on detecting small impulses that are replaced with the true image during contamination.

Sections 4.1.1 and 4.1.2 describes the first and second parameters respectively. Section 4.1.3 presents the underlying algorithm.

4.1.1 Pixel-Wise MAD

Let x_{ij}, m_{ij} and d_{ij} represent pixels with coordinates (i, j) of noisy image, median image and absolute deviation image, respectively. Also, let x_{ij}, m_{ij} and d_{ij} denote matrices whose elements are pixels of the corresponding images contained within the $(2K + 1) \times (2K + 1)$ size window W, centered around at position (i, j). The median image and absolute deviation image may be defined as in (4.1) and (4.2) respectively.

$$m_{ij} = \mathrm{median}(\mathrm{x}_{ij}) \tag{4.1}$$

$$d_{ij} =\mid x_{ij} - m_{ij} \mid \tag{4.2}$$

The median of the absolute deviations from the median, MAD, is defined as in (4.3).

$$MAD_{ij} = \mathrm{median}(\mid \mathrm{x}_{ij} - \mathrm{median}(x_{ij}) \mid) \tag{4.3}$$

Pixel-Wise MAD is defined as in (4.4).

$$PWMAD_{ij} = \mathrm{median}(\mathrm{d}_{ij}) = \mathrm{median}(\mid \mathrm{x}_{ij} - \mathrm{m}_{ij} \mid) \tag{4.4}$$

Pixel-Wise MAD, described in an iterative manner in (4.5), is a modified version of MAD [mad].

$$d_{ij}^{(n+1)} =\mid d_{ij}^{(n)} - \mathrm{median}(\mathrm{d}_{ij}^{(n)}) \mid \tag{4.5}$$

Where $d_{ij}^{(0)}$ is defined in (4.2) and $n = 0, 1, 2, 3,$ and 4. However, in the proposed work only one iteration is performed.

4.1.2 Rank-Ordered Absolute Difference

Let $x = (x_1, x_2)$ be the location of the pixel under consideration, and let

$$\Omega_{\mathrm{x}}(N) := \{\mathrm{x} + (i, j) : -N \leq i, j \leq N\} \tag{4.6}$$

be the set of pixels in a $(2N + 1) \times (2N + 1)$ neighborhood centered at x for some non-negative integers N. With $N = 1$, $\Omega_{\mathrm{x}}(1)$ represents a (3×3) window centered at x.

Let the set of pixels excluding the center pixel be defined as:

$$\Omega_{\mathrm{x}}^{0} = \Omega_{\mathrm{x}}(1) - \{x\} \tag{4.7}$$

For each pixel $y \in \Omega_{\mathrm{x}}^{0}$, let the absolute difference between gray value of the pixels x and y be defined as:

$$d_{\mathrm{x,x}} = |\mathrm{I_x} - \mathrm{I_y}| \tag{4.8}$$

Then find rank of the eight $d_{\mathrm{x,y}}$ values such that $r_i(\mathrm{x}) \leq r_{i+1}(\mathrm{x})$, $r = 1, \cdots, 7$. Hence, *Rank-Ordered Absolute Differences* (ROAD) may then be defined as:

$$\mathrm{ROAD}_m(\mathrm{x}) = \sum_{i=1}^{m} r_i(\mathrm{x}) \tag{4.9}$$

where $2 \leq m \leq 7$. In the simulation m is taken as 4 to find $\mathrm{ROAD}_4(\mathrm{x})$.

The following is an example of ROAD statistic generation.

$$\text{Original Neighborhood} = \begin{pmatrix} 154 & 183 & 83 \\ 160 & 210 & 222 \\ 115 & 190 & 75 \end{pmatrix}$$

$$\text{Absolute Differences} = \begin{pmatrix} 56 & 27 & 127 \\ 50 & - & 12 \\ 95 & 20 & 135 \end{pmatrix}$$

Four smallest absolute differences: $r_1 = 12$, $r_2 = 20$, $r_3 = 27$, and $r_4 = 50$

$$\text{ROAD} = \Sigma_{i=1}^{4} r_i = 12 + 20 + 27 + 50 = 109$$

This statistic provides a measure of how close a pixel value is to its four similar neighbors.

4.1.3 The IANS Algorithm

The proposed Improved Adaptive Noise Suppression (IANS) algorithm is based on the two methods described above. It needs two parameters as inputs to a neural network to decide the healthiness of a pixel. At first training patterns are generated to train the network. *Lena* image of size 128 × 128 is corrupted with 15% of Random Valued Impulsive Noise. 300 number of noisy patterns and another 300 number of noise-free patterns are generated. Each pattern consists of statistics obtained from Section 4.1.1 and 4.1.2 and their noise status i.e. *noisy* or *noise-free*. These training pattern are then trained with a (2–4–4–1) artificial neural network (Figure 4.1) by feeding them at random. The convergence characteristics of the network is shown in Figure 4.2. It can be seen from the plot that the convergence rate is very fast.

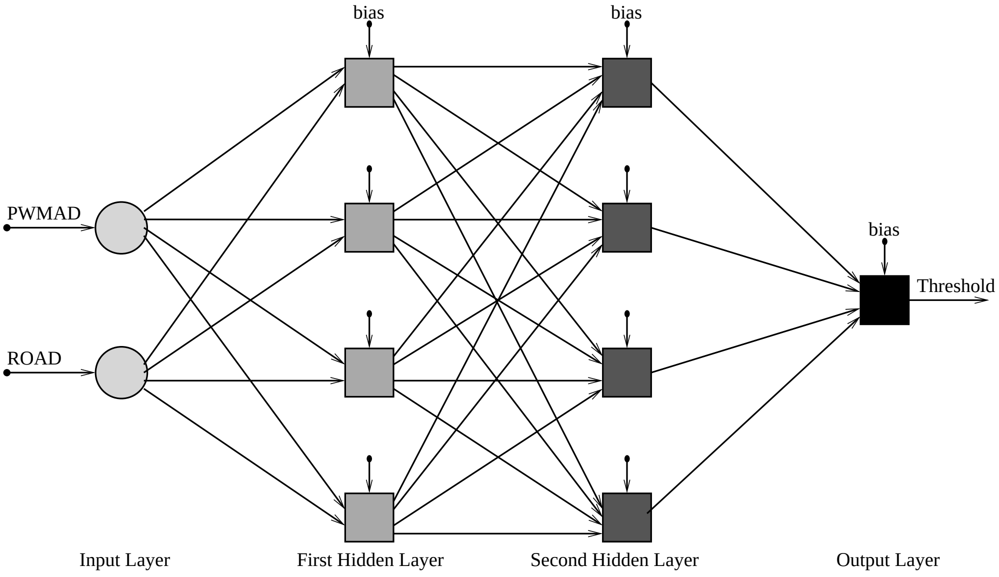


Figure 4.1: BPN structure for threshold estimation

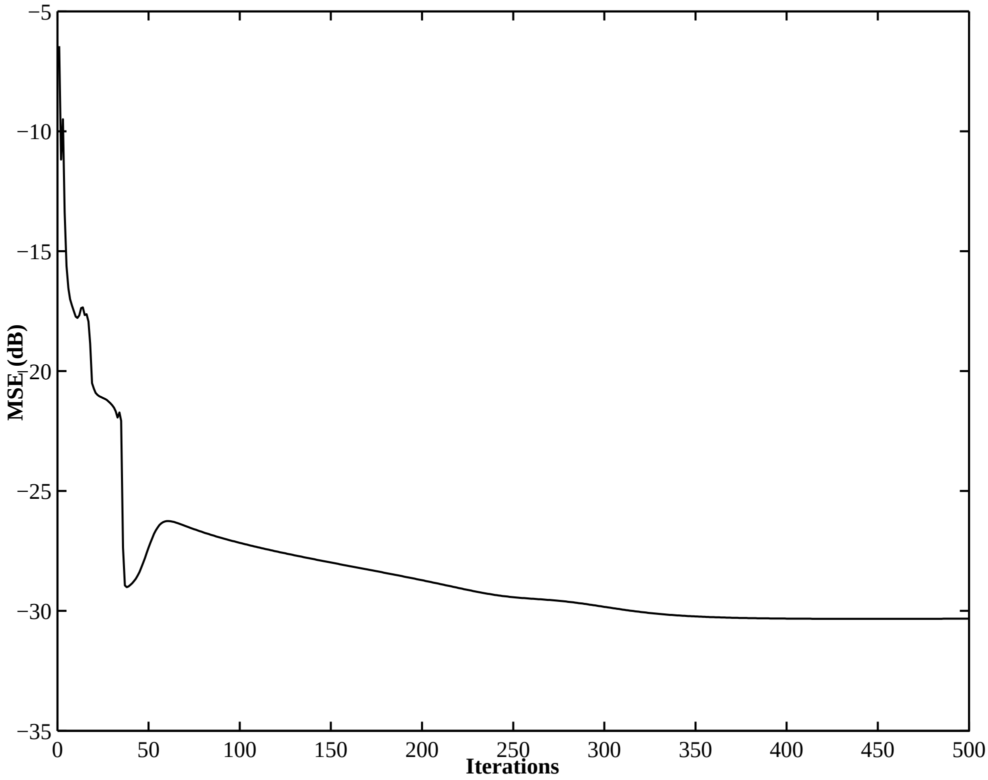


Figure 4.2: Convergence characteristics

4.2 Simulations and Results

The proposed scheme IANS is simulated on some standard images like *Lena*, *Lisa*,*Girl*, *Clown*, *Gatlin*, *Bridge*, *Boat* and *Peppers* etc. *Lena* image is corrupted with *Random Valued Impulsive Noise* of 1–30% noise densities. The seven noisy images thus generated are passed through the proposed IANS along with Signal Dependent-Rank Ordered Mean (SD-ROM) [32], Tri-State Median (TSM) [33] and Pixel Wise MAD (PWMAD) [28]. These are the few best performer in terms of noise suppression. The simulated result of PSNR (in *dB*) is plotted in Figure 4.5 and that of PSP (in Percentage) in Figure 4.6.

Few more comparisons are listed in the form of tables. Table 3.1 lists the PSNR of various images corrupted with 15% and 20% of noise. Similar observations of PSP are listed in Table 4.2.

The figures in 4.3 and 4.4 shows the images of restored *Lena* and restored *Peppers* corrupted with 15% and 20% of noise densities respectively.

Table 4.1: PSNR (dB) of different schemes at 15% and 20% of noise on different images

		Lisa	*Girl*	*Clown*	*Gatlin*	*Bridge*	*Boat*	*Peppers*
15% RVIN	PWMAD	30.50	29.28	23.02	30.66	26.07	29.25	31.29
	SD-ROM	31.98	31.31	24.33	32.77	27.48	30.75	32.00
	TSM	32.05	31.05	23.88	32.46	27.08	30.51	33.04
	IANS	38.65	34.94	24.30	34.71	27.42	31.06	35.14
20% RVIN	PWMAD	28.26	27.73	22.16	28.77	24.99	27.57	29.04
	SD-ROM	30.86	29.92	23.59	31.51	26.55	29.50	31.40
	TSM	30.95	29.69	23.28	31.30	26.28	29.35	31.41
	IANS	37.07	33.42	23.81	32.66	26.65	29.99	33.46

Table 4.2: PSP of different schemes at 15% and 20% of noise on different images

		Lisa	*Girl*	*Clown*	*Gatlin*	*Bridge*	*Boat*	*Peppers*
15% RVIN	PWMAD	7.35	11.05	24.85	4.00	15.83	13.95	10.53
	SD-ROM	0.14	0.93	10.29	0.39	3.06	1.04	0.34
	TSM	0.22	2.62	17.51	0.98	7.94	2.55	0.57
	IANS	0.31	1.11	15.34	1.26	7.53	2.56	0.95
20% RVIN	PWMAD	7.06	7.82	22.77	5.03	15.10	13.18	10.05
	SD-ROM	0.18	0.40	9.86	0.33	3.10	1.12	0.37
	TSM	0.31	1.21	21.07	1.00	8.30	2.82	0.73
	IANS	0.40	1.39	16.14	1.56	8.29	2.94	1.11

4.3 Summary

The proposed IANS is an efficient scheme in terms of noise detection and suppression. Two of the existing schemes are exploited here to make an efficient detector. When compared with the best reported schemes, it is found that the proposed one outperforms them.

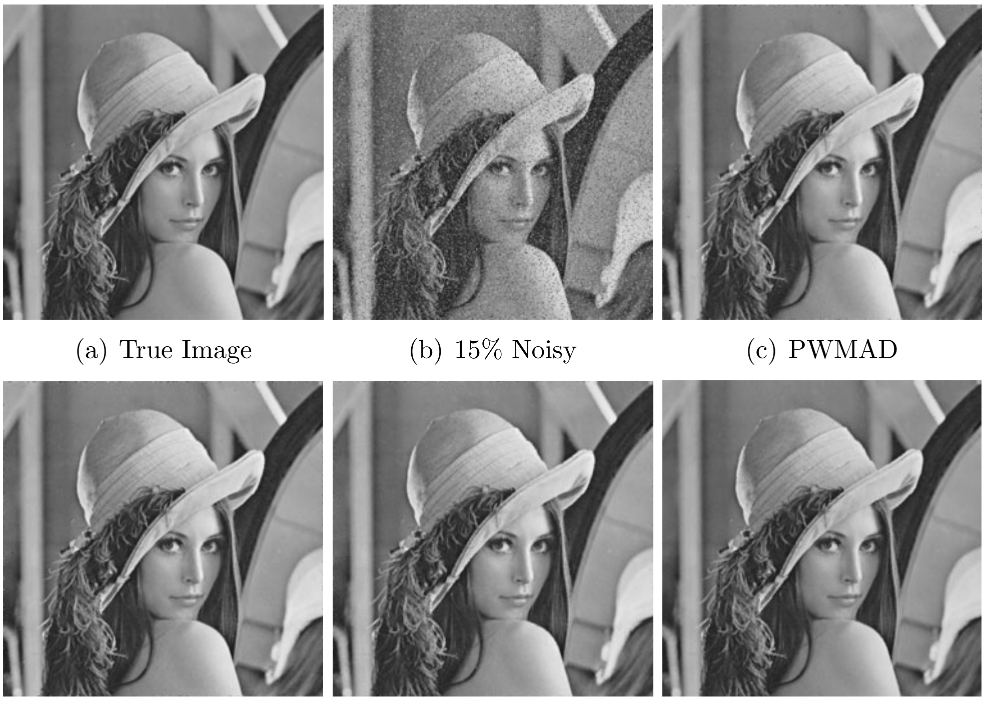

(a) True Image
(b) 15% Noisy
(c) PWMAD
(d) SDROM
(e) TSM
(f) IANS

Figure 4.3: Subjective comparison of impulsive noise removal of *Lena* image corrupted with 15% of RVIN by different filters

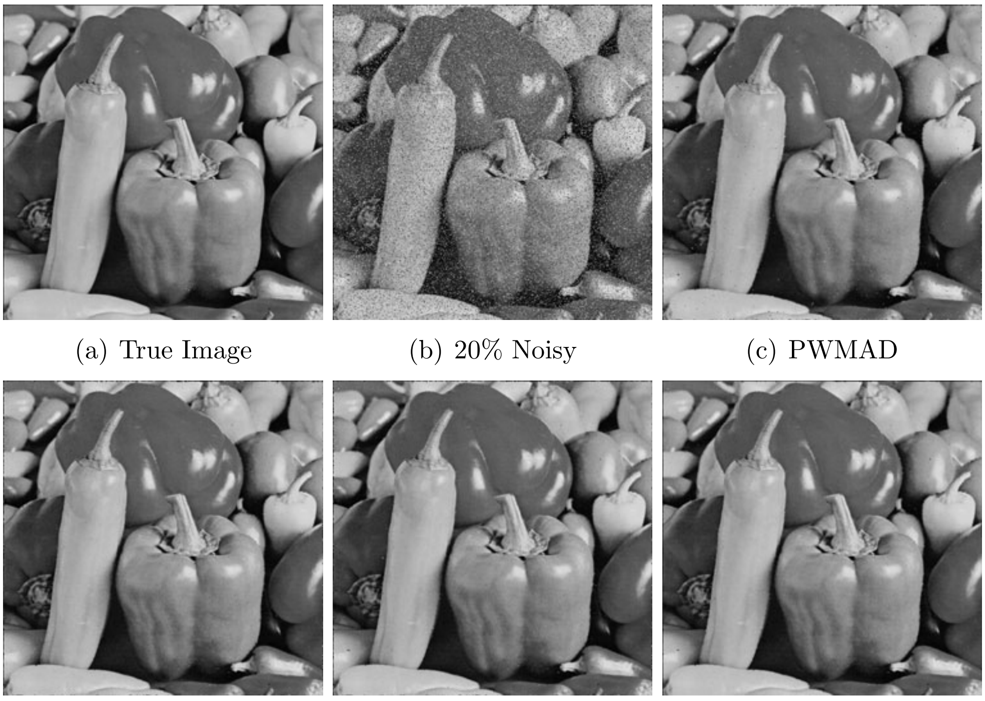

(a) True Image
(b) 20% Noisy
(c) PWMAD
(d) SDROM
(e) TSM
(f) IANS

Figure 4.4: Subjective comparison of impulsive noise removal of *Peppers* image corrupted with 20% of RVIN by different filters

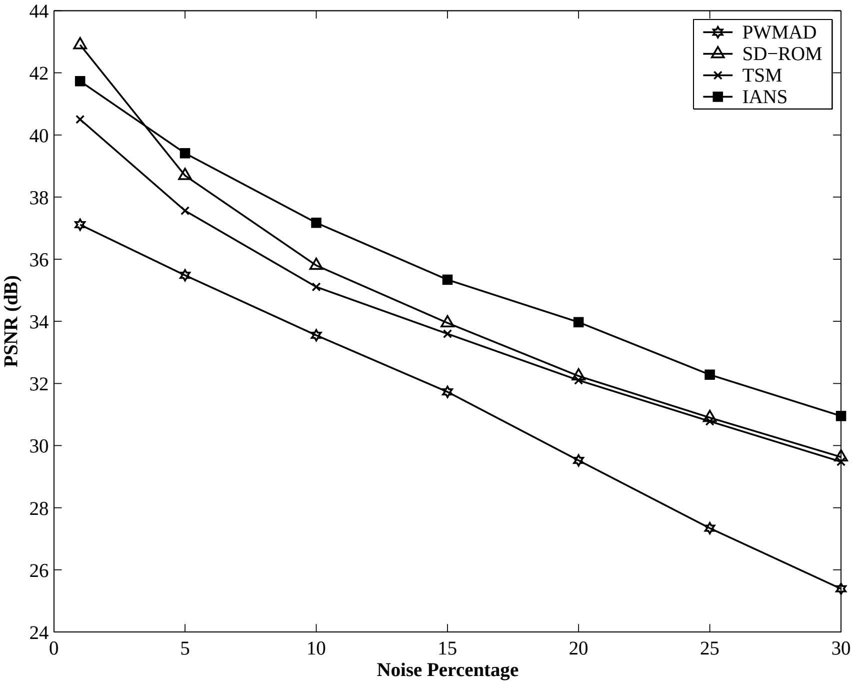


Figure 4.5: PSNR (dB) plot of Restored *Lena* image corrupted with RVIN of varying strengths

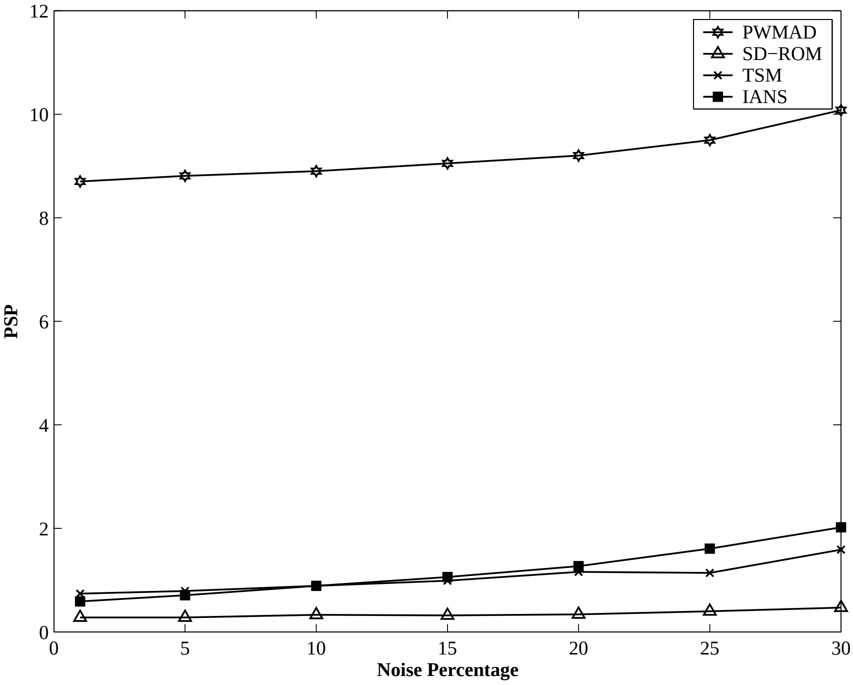


Figure 4.6: PSP plot of Restored *Lena* image corrupted with RVIN of varying strengths

Chapter 5

Conclusions

The work in this thesis, primarily focuses on impulsive noise suppression from images. Schemes for adaptive threshold selection for noise detection have also been devised. The work reported in this thesis is summarized in this chapter. Section 5.1 lists the pros and cons of the work. Section 5.2 provides some scope for further development.

5.1 Achievements and Limitations of the work

Two models of impulsive noise are considered in the thesis. One is *Salt & Pepper Noise* (SPN) and the other is *Random Valued Impulsive Noise* (RVIN). Then in subsequent chapters (Chapter 2–4) some novel schemes are proposed. Salient points of the thesis, highlighting the contribution at each stage, are presented below.

The first contribution is *Decision Directed Median Filter* (DDMF), which suppresses SPN from images. It is based on second order difference of pixels. The second difference of a test pixel is compared with a predefined threshold to determine its healthiness. The second contribution, *Fuzzy Impulsive Noise Detector* (FIND), also deal with SPN. Two parameters of a test window are supplied to a fuzzy membership function. Upon constructing the consequent membership function and subsequent defuzzification a decision is made on the noise status of the center pixel of the window. The restored images of these two schemes exhibit the desirable properties of edge and detail preservation. The inherent correlation among the pixels is exploited in these two schemes. However, it has a drawback of not making the threshold as adaptive.

Next three proposed schemes deal with RVIN and are based on second order difference of pixels. These three schemes primarily proposes three different techniques to select threshold in order to make noise detection process more reliable. In *Second Order Differential Impulse Detector* (SODID) an equation for threshold as function of variance of the noisy image pixels is fit to locate noise in the image. It needs a nonlinear equation to be fitted, which is a bit cumbersome as the true image statistics may not be available all the time.

ANN based Adaptive Thresholding for Impulse Detection (ANNAT) is another contribution that uses a simple artificial neural network (ANN) to determine the threshold value. A variation of ANN i.e. Functional Link ANN (FLANN) is used in another contribution namely *FLANN based Adaptive Threshold Selection for Detecting Impulsive Noise in Images* (FLANNAT). These two neural network approach use mean and variance of noisy image as input parameters of the network. Comparisons reveal that there are some better techniques in terms of PSNR. However, the proposed schemes computationally efficient.

The last contribution *Improved Adaptive Noise Suppression* also deals with removal of RVIN from images. Based on two different image statistics namely Pixel-Wise MAD and Rank Ordered Absolute Difference, this technique utilizes an ANN to predict the threshold value. In terms of PSNR this scheme outperforms its counterparts. Also it preserves image details and edges. These superiority in performance comes at the cost of computational overhead.

5.2 Further Development

To conclude this thesis, following are some points that may lead to some better and interesting results.

In this thesis, noise detection is mostly covered and for noise filtration median filter is used. Research may be undertaken to devise better filtration techniques. This technique together with a best detection technique can result in optimal restoration of degraded image.

As it has been stated that the existing as well as proposed techniques are computationally expensive, investigation may be carried out in this direction. Development of parallel algorithms can also be done to counter attack the computational overhead.

Bibliography

[1] B. Chanda and D. Dutta Majumder. *Digital Image Processing and Analysis.* Prentice-Hall of India, 1st edition, 2002.

[2] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Addison Wesley, 2nd edition, 1992.

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

[4] The NASA Website. http://history.nasa.gov.

[5] C. L. Chan, A. K. Katsaggelos, and A. V. Sahakian. Image Sequence Filtering in Quantum-Limited Noise with Applications to Low-Dose Fluoroscopy. *IEEE Transactions on Medical Imaging*, 12(3):610 – 621, September 1993.

[6] H. Soltanian-Zadeh, J.P. Windham, and A.E. Yagle. A Multidimensional Nonlinear Edge-Preserving Filter for Magnetic Resonance Image Restoration. *IEEE Transactions on Image Processing*, 4(2):147 – 161, February 1995.

[7] J. A. Goyette, G. D. Lapin, M. G. Kang, and A. K. Katsaggelos. Improving Autoradiograph Resolution Using Image Restoration Techniques. *IEEE Engineering in Medicine Biology*, pages 571 – 574, August/September 1994.

[8] The Walt Disney Company Website. http://disney.go.com.

[9] T.P. ORourke and R.L. Stevenson. Improved Image Decompression for Reduced Transform Coding Artifacts. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(6):490 – 499, December 1995.

[10] T. Ozcelik, J.C. Brailean, and A.K. Katsaggelos. Image and Video Compression Algorithms Based on Recovery Techniques Using Mean Field Annealing. In *IEEE Proceedings*, pages 304 – 316, February 1995.

[11] X. Lee, Y.Q. Zhang, and A. Leon-Garcia. Information Loss Recovery for Block-Based Image Coding Techniques-A Fuzzy Logic Approach. *IEEE Transactions on Image Processing*, 4(3):259 – 273, March 1995.

[12] S. Iyer and S. V. Gogawale. Image Enhancement and Restoration Techniques in Digital Image Processing. *Computer Society of India Communications*, pages 6 – 14, June 1996.

[13] I.Pitas and A.N.Venetsanopoulos. *Nonlinear Digital Filters: Principles and Applications*. Kluwer Academic Publishers, 1990.

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

[15] J.Astola and P.Kuosmanen. *Fundamentals of Nonlinear Filtering*. CRC Press, 1997.

[16] William K. Pratt. *Digital Image Processing*. John Wiley-Interscience Publication, 3rd edition, 2001.

[17] R. Bose. *Information Theory Coding and Cryptography*. TATA Mc-Graw Hill, India, 2003.

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

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

[20] D. R. K. Brownrigg. The Weighted Median Filter. *Communications ACM*, 27:807 – 818, August 1984.

[21] B. I. Justusson. *Median Filtering: Statistical Properties.* Two-Dimensional Signal Processing-II, T. S. Hwang Ed. New York: Springer Verlag, 1981.

[22] 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 Transactions on Image Processing*, 5(6):1012 – 1025, June 1996.

[23] Z. Wang and D. Zhang. Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images. *IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing*, 46(1):78 – 80, January 1999.

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

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

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

[27] C. Butakoff and I. Aizenberg. Effective Impulse Detector Based on Rank-Order Criteria. *IEEE Signal Processing Letters*, 11(3):363 – 366, March 2004.

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

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

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

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

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

[33] T. Chen, K. K. Ma, and L. H. Chen. Tri-State Median Filter for Image Denoising. *IEEE Transactions on Image Processing*, 8(12):1834 – 1838, December 1999.

[34] F. Russo. Impulse Noise Cancellation in Image Data Using A Two-Output Nonlinear Filter. *Measurement*, 36:205 – 213, 2004.

[35] L. Khriji and M. Gabbouj. Median-Rational Hybrid Filters. In *Proceedings of International Conference on Image Processing 1998*, volume 2, pages 853 – 857, October 1998.

[36] F. Russo and G. Ramponi. A Fuzzy Filter for Images Corrupted by Impulse Noise. *IEEE Signal Processing Letters*, 3(6):168 – 170, June 1996.

[37] T. Sun and Y. Neuvo. Detail Preserving Median Based Filters in Image Processing. *Pattern Recognition Letters*, 15:344 – 347, April 1994.

[38] D. A. P. Florencio and R. W. Shafer. Decision Based Median Filter using Local Signal Statistics. In *Proceedings of SPIE Symposium, Visual Comm. Image Processing*, volume 2038, pages 268 – 275, September 1994.

[39] J. S. R. Jang, C. T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing.* Prentice Hall International, USA, 1997.

[40] H. K. Kwan and Y. Cai. Fuzzy Filter for Image Filtering. In *Proceedings of Circuits and Systems, MWSCAS-2002, The 2002 45th Midwest Symposium*, volume 3, pages III–672 – III–675, August 2002.

[41] V. Kecman. *Learning and Soft Computing.* Pearson Education India, 1st edition, 2004.

[42] J. C. Patra, R. N. Pal, B. N. Chatterji, and G. Panda. Identification of Nonlinear Dynamic Systems Using Functional Link Artificial Neural Networks. *IEEE Transaction on Systems, Man, and Cybernatics*, 29(2):254 – 262, April 1999.

[43] J. C. Patra, G. Panda, and R. Baliarsingh. Artificial Neural Network-Based Nonlinearity Estimation of Pressure Sensors. *IEEE Transaction on Instrumentation and Measurement*, 43(6):874 – 881, December 1994.

[44] R. Garnett, T. Huegerich, C. Chui, and W. He. A Universal Noise Removal Algorithm With an Impulse Detector. *IEEE Transactions on Image Processing*, 14(11):1747 – 1754, November 2005.

[45] T. Loupas, W. N. McDicken, and P. L. Allan. An Adaptive Weighted Median Filter for Speckel Suppression in Medical Ultrasonic Images. *IEEE Transactions on Circuits and Systems*, 36(1):129 – 135, January 1989.

[46] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo. Weighted Median Filters: A Tutorial. *IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing*, 43:157 – 192, March 1996.

[47] A. C. Bovik, T. Hwang, and D. C. Munson. A Generalization of Median Filtering using Linear Combinations of Order Statistics. *IEEE Transactions on Accoustics, Speech and Signal Processing*, ASSP-31:1342 – 1350, June 1983.

[48] T. Loupas, W. N. McDicken, and P. L. Allan. An Adaptive Weighted Median Filter for Speckle Suppression in Medical Ultrasonic Images. *IEEE Transaction on Circuits and Systems*, 36(1):129 – 135, January 1989.

[49] Y. H. Pao and Y. Takefuji. Functional-Link Net Computing: Theory, System Architecture, and Functionalities. *Computer*, 25(5):76 – 79, May 1992.

[50] Simon Haykin. *Neural Networks.* Prentice Hall, 2nd edition, 1999.

[51] D. V. D. Ville, M. Nachtegael, D. V. D. Weken, E. E. Kerre, W. Phillips, and I. Lemahieu. Noise Reduction by Fuzzy Image Filtering. *IEEE Transaction on Fuzzy Systems*, 2(4):429 – 436, August 2003.

[52] F. Russo and G. Ramponi. Fuzzy Operator for Sharpening of Noisy Images. *IEEE Electronics Letters*, 28(18):1715 – 1717, August 1992.

[53] Sukadev Meher. *Development of Some Novel Nonlinear and Adaptive Digital Image Filters for Efficient Noise Suppression.* PhD thesis, National Institute of Technology Rourkela, 2005.

[54] Mark R. Banham and Aggleos K. Katsaggelos. Digital Image Restoration. *IEEE Signal Processing Magazine*, 14(2):24 – 41, March 1997.

[55] Lofti A. Zadeh. Fuzzy Logic. *IEEE Computer*, 21(4):83 – 93, April 1988.

[56] Lofti A. Zadeh. Soft Computing and Fuzzy Logic. *IEEE Software*, 11(6):48 – 56, November 1994.

[57] P. P. Bonissone. Soft computing and fuzzy logic. *Soft Computing, Springer-Verlag*, 1:6 – 18, 1997.

[58] P. Wang and S. Tan. Soft computing and fuzzy logic. *Soft Computing, Springer-Verlag*, 1:35 – 41, 1997.

[59] Lofti A. Zadeh. Fuzzy Logic = Computing with Words. *IEEE Transactions on Fuzzy Systems*, 4(2):103 – 111, May 1996.

[60] M. Gabbouj and J. Astola. Nonlinear Order Statistic Filter Design: Methodologies and Challenges. In *Proceedings of EUSIPCO 2000, Signal Processing X-Theories and Applications*, Tampere, Finland, September 2000.

Dissemination of Work

Published

1. B. Majhi, P. K. Sa and G. K. Panda, "ANN based Adaptive Thresholding for Impulse Detection", *IASTED International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA-2006)*, pp 294 - 297, 15 - 17 February, 2006, Innsbruck, Austria.

2. B. Majhi, P. K. Sa and G. Panda, "Decision Directed Median Filter", *Conference on Information Technology (CIT-2005)*, pp 233 - 235, 20 - 23 December, 2005, Bhubaneswar, India.

3. B. Majhi, P. K. Sa and G. Panda, "Second Order Differential Impulse Detector", *International Conference on Intelligent Systems (ICIS-2005)*, 1 - 3 December, 2005, Kuala Lumpur, Malaysia.

Communicated

1. B. Majhi and P. K. Sa, "FLANN based Adaptive Threshold Selection for Detection of Impulsive Noise in Images", *International Journal of Electronics and Communications.*