NEW DEVELOPMENTS IN MEDIAN FILTERS

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

By

KHUMANTHEM MANGLEM SINGH

to the

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING,

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

July 2006

CERTIFICATE

It is to certify that the work contained in this thesis entitled "NEW DEVELOPMENTS IN MEDIAN FILTERS" by Khumanthem Manglem Singh has been carried out under my supervision at Research and Development Laboratory, Department of Electronics and Communication Engineering, Indian Institute of Technology Guwahati, India and that this work has not been submitted elsewhere for a degree.

Prabin K. Bora
Professor
Department of Electronics and Communication Engineering
Indian Institute of Technology Guwahati, India

July 2006

ACKNOWLEDGEMENTS

I have great pleasure to express my deepest feelings of gratitude to Prof. Prabin K. Bora, Department of Electronics and Communication Engineering, Indian Institute of Technology, Guwahati, who as my supervisor has extensively helped in completion of this work with his valuable suggestions, cooperation, patience and love.

My sincere thanks go to Dr. S. Dandpat, Head of the Department, ECE, Indian Institute of Technology, Guwahati for providing me constant encouragement.

I thank Prof. A. Mahanta, Prof. A. Gogoi, and other teaching and non-teaching staff of the Department of Electronics and Communication Engineering, Indian Institute of Technology, Guwahati for their valuable advice to carry out this work.

Many thanks go to Dr. S. Birendra Singh, Director and other staff of DOEACC Centre, Imphal whose assistance was vital for the research.

Specially, I would like to give my special thanks to my wife Jibanlata, and my children Sanada, Segal and Chanchanbi whose patient love enabled me to complete this work.

Contents

List of Tables

List of Figures

Nomenclatures

BVDF	Basic vector directional filter
Chen and Wu	Chen and Wu's adaptive centre-weighted median
CLPF	Causal linear prediction filter
CLPVMF	Causal linear prediction vector median filter
CWM	Centre-weight median
CWVMF	Centre-weighted vector median filter
DDF	Directional distance filter
LPE	Linear prediction error
MAD	Median of the absolute deviations
MLE	Maximum likelihood estimate
MMF	Marginal median filter
NCLPF	Noncausal linear prediction error based filter
NCLPVMF	Noncausal linear prediction error based vector median filter
NSHP	Nonsymmetric half-plane
PSNR	Peak-signal-to-noise-ratio
RCM	Rank-conditioned median
RCT	Rank conditioned and threshold
RCTVMF	Rank-conditioned and threshold vector median filter
RCVMF	Rank-conditioned vector median filter
ROM	Rank-ordered mean
SD-ROM	Signal-dependent rank-ordered mean
SM	Standard median
SOD	Sum of distances
TSM	Tri-state median
VMF	Vector median filter
VMF_1	Vector median filter based on 1-norm
VMF_2	Vector median filter based on 2-norm
VSD-ROM	Vector signal-dependent rank-ordered mean

Mathematical Notations

$\boldsymbol{\beta}$	Location parameter
$\hat{\boldsymbol{\beta}}$	MLE of location parameter
δ_i	Sum of distances
δ_i^w	Weighted sum of distances
$\delta_{(i)}$	*i*th ordered sum of distances
$\phi(\boldsymbol{x}_i, \boldsymbol{x}_j)$	Angular distance between the *i*th and *j*th neighbouring vector pixels
γ_k	Constant of Chen and Wu's adaptive centre-weighted median filter
ψ_k	Rank-ordered difference of the SD-ROM filter
θ	Pre-defined threshold of the RCT-I filter
θ_k	Pre-defined threshold of the SD-ROM filter
η	Prediction order
∇	Gradient operator
$a(i, j)$	Prediction coefficient
$\boldsymbol{a}_{\boldsymbol{\eta}}$	Column vector for prediction coefficient
C	Number of channels
d	Difference between the centre pixel value and median
D	Distance between a healthy vector pixel and centre vector pixel
$d_1(\boldsymbol{x}_i, \boldsymbol{x}_j)$	City-block distance
$d_2(\boldsymbol{x}_i, \boldsymbol{x}_j)$	Euclidean distance
$d(\boldsymbol{x}_i, \boldsymbol{x}_j)$	Distance measure between the *i*th and *j*th neighbouring vector pixels

D_k	Difference between the centre pixel value and adaptive centre-weighted median
$d_M(\boldsymbol{x}_i, \boldsymbol{x}_j)$	L_M norm distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$
d_w	Difference between the centre pixel and CWM Output
$d_\infty(\boldsymbol{x}_i, \boldsymbol{x}_j)$	Chess-board distance
$e(m,n)$	Error of the linear prediction
$F(x)$	Probability density function
g	Fixed impulse noise
h	Positive impulse
I_{MAX}	Maximum of the three components of all the vector pixels
l	Negative impulse
$M_1 \times M_2$	Size of the image
N	Number of pixels in the observation window
$n(i)$	Impulse noise at the ith location
N_T	Pre-defined rank
p	Percentage of impulse noise ratio
$O(\mathrm{g}(n))$	Computational complexity of order $g(n)$
$\boldsymbol{r}_{x\eta}$	Cross-correlation vector of the pixels
$\boldsymbol{r}_{X\eta}$	Cross-correlation vector of the vector pixels
$\boldsymbol{R}_{x\eta}$	Autocorrelation matrix of the pixels
$\boldsymbol{R}_{X\eta}$	Autocorrelation matrix of the vector pixels
S	Mean-square Euclidean norm of the error vector
$\hat{s}_{\text{MLE}}$	Maximum likelihood estimate
$S^{\eta}_{(m,n)}$	Set of neighbours of site (m,n)

T	Pre-defined threshold of the TSM filter
T_k	Threshold of Chen and Wu's adaptive centre-weighted median filter
w	Weight of the centre pixel
W_1	Causal region of support
W_2	Noncausal region of support
x_1	Upper left pixel in the window
$\boldsymbol{x}_1$	Upper left vector pixel in the window
$x_{2-\text{NCLPF}_{\text{SM}}}$	Output of the second-order NCLPF based on standard median
$\boldsymbol{x}_{2\text{-NCLPVMF}}$	Output of the second-order NCLPVMF
x_{CHEN}	Output of Chen and Wu's adaptive centre-weighted median filter
x_{CWM}	Output of the centre-weighted median filter
$\boldsymbol{x}_{\text{CWVMF}}$	Output of the centre-weighted vector median filter
$\boldsymbol{X}_{\text{E}}$	Set of extended pixels
x_i	ith observation pixel
$\boldsymbol{x}_i$	ith vector observation pixel
$x_{(i)}$	ith order statistic
$\boldsymbol{x}_{(i)}$	ith rank-ordered vector pixel
$x(m,n)$	Scalar pixel at the (m,n) location
$\hat{x}(m,n)$	Predicted scalar pixel at the (m,n) location
$\boldsymbol{x}(m,n)$	Vector pixel at the (m,n) location
$\hat{\boldsymbol{x}}(m,n)$	Predicted vector pixel at the (m,n) location
x_N	Lower right pixel in the window
$\boldsymbol{x}_N$	Lower right vector pixel in the window
$\boldsymbol{x}_{\boldsymbol{\eta}}$	Row vector of pixels used for prediction

	pixel
$\boldsymbol{X}_{\mathrm{R}}$	Set of rank-ordered pixels
x_{RCM}	Output of the rank-conditioned median filter
$\boldsymbol{x}_{\mathrm{RCTVMF}}$	Output of rank-conditioned and threshold vector filter
$\boldsymbol{x}_{\mathrm{RCVMF}}$	Output of rank-conditioned vector median filter
$\boldsymbol{X}_{\mathrm{RE}}$	Set of extended rank-ordered pixels
x_{ROM}	Rank-ordered mean
$x_{\mathrm{SD\text{-}ROM}}$	Output of the SD-ROM filter
x_{SM}	Output of the standard median filter
x_{TSM}	Output of the tri-state median filter
$\boldsymbol{x}_{\mathrm{VMF}}$	Output of the vector median filter
x^{w}	Output of the CWM filter with varied centre weights
$\boldsymbol{X}_{\mathrm{W}}$	Set of pixels in the observation window

ABSTRACT

The median filter is a nonlinear filter that outputs the median of the data inside a moving window of predetermined length and has been traditionally used to remove impulse noise from images. The filter is known for its ability to preserve edges while smoothing a noisy signal. One of the major drawbacks of the median filter is that it tends to alter pixels undisturbed by noise and disturb features like thin lines, because it is implemented invariantly across the image. One way to circumvent this problem is to detect impulse noise prior to the filtering operation. In this work, two new types of median filters that employ impulse detection mechanisms prior to filtering operations are proposed. These are the *rank-conditioning and thresholding* (RCT) *median filters* and the *noncausal linear prediction error based median filters.*

The rank-conditioned and threshold median filters are based on simultaneous application of rank-conditioning and the thresholding of the deviation of the center pixel from healthy pixels in the window. Simultaneous application of these conditions is aimed at reducing the probability of a healthy pixel being detected as noisy. Rank-conditioning is kept fixed as one of the conditions - *the rank of the centre pixel inside the sliding window does not lie in the trimming range if it is an impulse*. Three different thresholding conditions are investigated. In RCT-I filter, the absolute difference of the centre pixel from the median is compared against a pre-defined threshold. The RCT-II filter employs the thresholding scheme of Chen and Wu's adaptive centre-weighted median filter. The thresholding scheme of signal-dependent rank-ordered mean filter is used in RCT-III filter. The corrupted pixel is replaced by the rank-ordered mean.

The RCT scheme is extended to the colour image by developing the *rank-conditioned and threshold vector median filter (*RCTVMF*).* The sum of distances (*SOD*) of each vector pixel from other vector pixels in the window is calculated and the vector pixels are ranked according to their *SOD* values. The vector median has the least *SOD* value. The

centre vector pixel is decided as corrupted if its rank does not lie in the trimming range and its distance from a healthy vector pixel is bigger than a pre-defined threshold. The corrupted centre vector pixel is replaced by the vector median.

Using linear-prediction error (*LPE*) to detect impulse noise in an image is investigated in another novel approach. The s*econd-order noncausal linear prediction* models are fitted in blocks of the image. The centre pixel in the window is predicted using the estimated prediction coefficients. The impulse noise is detected by thresholding the prediction error. The corrupted centre pixel is replaced by either the median or the rank-ordered mean. To control the effect of a corrupted pixel in the *LPE* of the neighbouring pixel, the median pre-filtered image is used in prediction. For a colour image, the *constrained multichannel noncausal linear prediction model* is applied for prediction, the marginal median filter is used for pre-filtering and the corrupted vector pixel is replaced by the vector median in the window.

The proposed algorithms are tested on a number of test images artificially corrupted with impulse noise. We will show that these new developments in median filtering have impressive overall performance comparable to or bettering the performance of some of the best median filtering methods.

Chapter 1

Introduction

Images are prone to be corrupted by noise during acquisition, transmission and storage. Noise affects the quality of visual perception and results of image processing operations. Filtering is one of the most fundamental operations of image and video processing with many enhancement applications like *sharpening the image*, *attenuating noise*, *highlighting the edge*, etc. Linear and nonlinear filters have proved to be useful in reducing noise from a corrupted image.

1.1 Linear Noise Filtering

Linear noise filtering techniques are used extensively and offer satisfactory performance for a variety of applications. Simple low-pass averaging filters and model-based Wiener filters are widely used for reduction of additive white Gaussian noise [1]. These filters are easily implemented and their properties are well-understood. In spite of the elegant theory, not all problems can be satisfactorily addressed through the use of linear filters. These techniques fail in reducing non-Gaussian and non-white noise. Such types of noise are encountered in a multitude of digital signal and image processing applications. An *impulse noise* is an example of such types of noise.

1.1.1 Limitations of Linear Noise Filtering

Image edges and details (e.g. corners, lines and ends of lines) have high frequency content and carry very important information for human perception. Filters having good edge and image detail preservation properties are highly suitable for digital image filtering. Most of the classical linear noise filters tend to blur edges and destroy lines, corners and other fine image details. Another example where linear techniques fail is the case of nonlinear image degradations. Such degradations occur during image formation and image transmission through nonlinear channels. The human visual perception mechanism has been shown to have nonlinear characteristics as well. Many researchers seem to have the view that it is difficult to obtain major breakthroughs in signal and image processing without resorting to nonlinear methods [1]. These reasons have led to the search for nonlinear filtering techniques.

A bulk of research on nonlinear image processing techniques has been presented in the past decades. Some examples on nonlinear image processing techniques are o*rder statistics-based filtering, homomorphic filtering, polynomial filtering, morphological and neural-network-based image processing* and *nonlinear image restoration*. Each class of techniques possesses its own mathematical tools that provide reasonably good analysis of its performance. One of the limitations of nonlinear techniques is the lack of a unifying theory. A good underlying theory covering a subclass of nonlinear filters is needed for the research to have a common foundation. Excellent covering on median filters and

related nonlinear filters may be found in [1,2]. Broader classes of nonlinear filters are considered in [3].

1.1.2 Impulse Noise and Median Filtering

An impulse noise is a short-duration noise of fixed or variable amplitudes. It is frequently encountered in digital transmission as a consequence of man-made noise sources or decoding errors. In the statistical sense, an impulse is an *outlier* that is inconsistent with the rest of the data in the neighbourhood. It may appear as isolated bright and dark points in the image. Speckle noise that appears in ultrasonic imaging and images formed by laser is also impulse noise. Linear filters are ineffective in reducing such noise.

A nonlinear filter class that has been proven very useful is the class of median based filters used for impulse noise suppression. Filters of this class have been subject to growing interest since the discovery of the standard median (SM) filter by Tukey [4] who applied it to the smoothing of statistical data. Pratt [5] was the first to use median filters in image processing. Later, median-based filters have shown their usefulness in many single- and multi-dimensional applications.

The median filter is a nonlinear filter that outputs the median of the data inside a moving window of pre-determined length. This filter is easily implemented and has some attractive properties. It performs well for heavy-tailed noise distributions (e.g. the Laplacian distribution), whereas its performance is poor for light-tailed noise distributions (e.g. the uniform distribution) [1]. The impulse noise is an example of very heavy-tailed noise. Thus the median filter is an appropriate filter to reduce it. The drawback of the median filter is the resulting blurring caused to the image. Especially for a larger window size, it destroys the fine image details such as thin lines and corners while reducing noise. [1,5]

1.2 Review on Modifications of Median Filtering

There has been a continuous effort to improve the performance of the median filter. One of the earliest contributions is the weighted median (WM) filter [6-9], which gives more weight to some pixels within the window than others. It emphasizes or de-emphasizes

specific input samples, because in most applications, not all samples are equally important. The special case of the WM filter is the centre-weighted median (CWM) filter [10], which gives more weight only to the central value of the window. It is reasonable to give emphasis to the central sample, because it is one that is the most correlated with the desired estimate. The adaptive median filters use different window size depending on the statistics of the pixels in the window [11].

1.2.1 Detection-based Median Filtering

The median filter and its modifications are generally implemented to all pixels in an image. They tend to alter pixels undisturbed by noise. As a result, their effectiveness in noise suppression is often at the expense of blurred and distorted image features. A better way to circumvent this drawback is to incorporate some decision-making processes in the filtering framework. At each pixel location, it is first determined whether the current pixel is contaminated. Filtering is applied on the pixel if it is detected to be corrupted by noise. The corrupted pixels are replaced by the median values, while the noise-free pixels are left unaltered. The simplest detection scheme [2] is based on the principle of outlier detection in statistical data. It thresholds the absolute value of the deviation of the centre pixel from the median of the pixels in the window.

The adaptive median filters [11] that use variable window sizes adaptively to filter noise have a detection step based on the statistics of the pixels in the window. A simple but effective impulse-detection-based filter is the rank-conditioned median (RCM) filter [12,13], in which pixels in the filtering window are ranked according to their magnitudes. The centre pixel is considered to be corrupted if it lies outside the *trimming range* [14], which is the subset obtained by excluding the extreme values in a data set. In the signal-dependent rank-ordered mean (SD-ROM) filtering [3,15,16], all but the centre pixel in a filtering window are considered for rank-ordering. An impulse is detected by thresholding the differences between the current pixel and each of one half of the ranks. The detected impulse is replaced by the mean of two central ranks. The SD-ROM filter has been shown to work well in removing fixed-valued impulse noise and random-valued impulse noise.

Recent work by Chen and Wu [1 7] has exploited the CWM filter to derive the scheme for the detection of impulses. In this approach, thresholds are selected from the absolute differences between the centre pixel and the CWM outputs. There are other decision based median filters and some of them are available in [21 - 28].

1.2.2 Median Filtering for Colour Images

All techniques so far mentioned are appropriate only for gray-scale images and cannot be directly applied in multichannel images without modification. Median-filtering based techniques cannot be applied to each channel separately, because this will disrupt the colour balance across the channels. The extension of the concept of the scalar median filter and its modifications to colour image processing is not straightforward. The colour vectors have to be sorted according to an order for median filtering, but there is difficulty in defining the natural concept of rank in the case of vectors. One of the most popular filtering methods is the vector median filter (VMF) [2,29-35]. Its basic principle is to map each vector into a scalar, rank the scalars and define the rank of the vectors according to the ranks of corresponding scalars. The scalar considered in the VMF is the sum of the distances between each vector pixel and other vector pixels in the window. The vector pixel with the smallest sum of distances is the vector median. Another class of the vector filter operates on the direction between vector pixels rather than on the distance between them. These filters aim at eliminating vector pixels with atypical direction in the colour space. The basic vector directional filter (BVDF) [36] calculates the sums of the angles between each vector pixel and other pixels in the filtering window and replaces the current vector pixel with one having the minimum sum of angles. The directional distance filter (DDF) combines the ordering principles of both the VMF and DDF [37]. Variants of the BVDF and DDF are also proposed [37].

The vector median filter and its modifications are generally implemented on all vector pixels in an image. It tends to alter vector pixels undisturbed by noise. It modifies edges and image details in the cases where the noise ratio is high and results in blurred and distorted image features. At each vector-pixel location, it is first determined whether the current vector pixel is corrupted. Filtering is applied on the corrupted vector pixels. The

corrupted vector pixels are replaced by the vector median values, while the noise-free vector pixels are left unaltered. Since all vector pixels are not filtered, undue distortion is avoided. A simple but effective impulse noise detection filter is the centre-weighted vector median filter [38,39], which emphasizes the centre vector pixel. It is an improvement over the weighted vector median filter [39,40]. In the vector signal-dependent rank-ordered mean (VSD-ROM) filtering [33], all but the centre vector pixel in a filtering window are considered for rank-ordering. An impulse is detected by thresholding the differences between the centre vector pixel and each of the one half ranks. The detected impulse is replaced by the vector median in the window excluding the centre vector pixel. The vector median filter and its modifications use colour impulse noise model [45-51] for simulation of impulse noise.

1.3 Motivation of the Present Work

Though median filtering is a classical problem in image processing, it continues to attract researchers because of the challenge involved in filtering out impulse noise without disturbing the finer details of the image. The current scenario leaves room for research in the detection and removal of impulse noise from both gray-scale and colour images. Motivated by this goal, the thesis develops new detection-based algorithms for filtering impulse noise. The major contributions of the work reported in this thesis are as follows:

- The scalar *rank-conditioned and threshold (RCT)* median filters are proposed by modifying the simple rank conditioned median filter.
- A new filter *rank-conditioned and threshold vector median filter* (RCTVMF*)* is proposed for the removal of impulse noise from colour images. It is a modification of the vector median filter that incorporates a detection mechanism to filter only corrupted pixels in the image.

A *linear-prediction error* based impulse noise detection technique is proposed. Hsue and Yagle have opined that a better estimate of the sample would be expected if linear prediction is based on both the past and future samples in the neighbourhood of the current sample [52-55]. It motivates us to design an impulse-noise detection technique

based on thresholding the error of noncausal linear prediction. Particularly, the second-order noncausal linear predictors are considered

1.4 Organization of the Thesis

The organization of the rest of the thesis is as follows.

Chapter 2 describes different types of impulse noise models for the gray-scale and the colour images. Mathematical formulations and important variants of median filters are discussed as a background to the developments of subsequent chapters.

Chapter 3 is devoted to the new class of detection-based filters called the *scalar* r*ank-conditioned and threshold median filters* for the removal of impulses from gray-scale images. Three different detection schemes are proposed.

Chapter 4 describes a new type of vector median filter called the *vector rank-conditioned and threshold median filter* with the ability to detect impulses in colour images.

Chapter 5 explores the use of linear prediction error for detection of impulse noise. A class of filters called *noncausal linear prediction error based median filters* is proposed.

Chapter 6 draws conclusions from the research work reported in this thesis and suggests a few directions of further developments.

Chapter 2

Impulse Noise and Median Filters

2.1 Introduction

Impulse noise is a short-duration noise of fixed or variable amplitudes. Images may be corrupted by impulse noise due to a variety of causes, such as saturation of the sensors or the adverse channel environment in a communication system. The data that are affected by impulse noise are perceptually visible, because their amplitudes are either very big or very small.

Impulse noise may be classified into two types – *fixed-valued impulse noise* and *random-valued impulse noise*. The first type of impulse noise appears as black and/or white spots in gray-scale images and mainly arises owing to the saturation of the imaging sensors. Extreme values are digitised as minimum and maximum allowed values. Such fixed-valued impulse noise is called the *salt-and-pepper noise*. Pure salt-and-pepper noise is an idealistic case. It is easy to detect such noise from images, because checking whether the pixel has a maximal or a minimal value can reveal it. The random-valued impulse noise has varying amplitudes and is more realistic. A typical source of such noise is the communication channel that is usually affected by lightning or other atmospheric disturbances.

Impulse noise belongs to the category of heavy-tailed noise for which the probability density approaches zero more slowly as values of the signal become larger. There are a number of models to represent impulse noise in images. Each of these models gives the probability of a pixel being corrupted by impulse noise. Two categories of impulse noise models are discussed in this chapter. The first category is basically meant for gray-scale images and the second, for colour images. The models for gray-scale images are widely discussed and analysed [2,56]. An impulse noise is also considered as an *outlier*. In statistics, an outlier is defined as an observation that appears to be inconsistent with the neighbouring data.

2.2 Bernoulli-state Impulse Noise Model

The state representing occurrence of the impulse noise at a location i is generally modelled by a binary Bernoulli process $b(i)$ that takes "1" with a probability p and "0" with a probability $1-p$. The corrupted signal values have the form:

$$x(i) = b(i)n(i) + (1-b(i))s(i) \tag{2.1}$$

where $n(i)$ is the impulse noise. $n(i)$ can take values in the dynamical range of the image according to some underlying distribution. Following cases are generally considered.

(a) If $n(i)$ is fixed at a value g, we have the one-sided impulse noise model [36,56]. Thus, the one-sided impulse noise model can be expressed by:

$$x(i) = \begin{cases} g & \text{with a probability } p \\ s(i) & \text{with a probability } (1-p) \end{cases} \tag{2.2}$$

(b) If $n(i)$ takes two fixed values l (negative impulse) and h (positive impulse) with probability q and $(1-q)$ respectively, we have the two-sided impulse noise model [36,56]. We are particularly interested in the case of fixed impulse with $q = \frac{1}{2}$. In the case of the two-sided impulse noise model, we have

$$x(i) = \begin{cases} h & \text{with probability } p/2 \\ l & \text{with a probability } p/2 \\ s(i) & \text{with a probability } (1-p) \end{cases} \tag{2.3}$$

(c) If the process $n(i)$ is modelled as a uniformly distributed random process, we have the Bernoulli-uniform impulse noise model. Thus

$$x(i) = \begin{cases} n(i) & \text{with a probability } p \\ s(i) & \text{with a probability } (1-p) \end{cases}$$

where $n(i)$ is an independent and identically distributed (*iid*) random process with the uniform probability density function in the interval $[l, h]$.

(d) The noise process $n(i)$ can also be considered to be an *iid* Gaussian process. In this case, we will have the Bernoulli-Gaussian impulse noise model.

2.3 Bit Errors Impulse Noise Model

Impulse noise normally does not take fixed values [2,56]. Its amplitude ranges from the minimum to maximum allowable range in the realistic case (between 0 and 2^B - 1, where B is the number of bits per pixel). It may occur due to the random change of bits in the signal.

Consider the signal $s(i)$ at point i given by

$$s(i) = k_1(i)2^{B-1} + k_2(i)2^{B-2} + ... + k_{B-1}(i)2 + k_B(i) \tag{2.4}$$

where $k_j(i) \in \{0,1\}$. Assume that the bit errors occur with probability p independent both of the errors at other signals and at the other bit locations of the current signal. Then the corrupted signal values are of the form:

$$x(i) = k_1^*(i)2^{B-1} + k_2^*(i)2^{B-2} + ... + k_{B-1}^*(i)2 + k_B^*(i), \tag{2.5}$$

where

$$k_j^*(i) = \begin{cases} k_j(i), & \text{with probability } 1-p \\ 1-k_j(i), & \text{with probability } p \end{cases} \tag{2.6}$$

Removing bit-errors impulse noise is a challenging task.

2.4 Mixed Impulse Noise Model

Noise may come from two or more separate sources [2,56]. One of the sources may impart impulse noise, while another source may impart continuously distributed noise like the Gaussian noise. In such a situation, the noise can be modelled to have both continuous and discrete characteristics. The corrupted signal in this case is given by

$$x(i) = s(i) + n(i) \tag{2.7}$$

where $n(i)$s are *iid* random variables. The probability density function (pdf) $f_n(u)$ of $n(i)$ can be expressed as

$$f_n(u) = (1-p)\,f(u) + \frac{p}{2}(\delta(h-u) + \delta(l-u)) \tag{2.8}$$

where $f(u)$ is a continuous density function, p is the probability of the pixel being corrupted by impulse noise and $\delta(\,)$ is the Dirac delta function. An image signal received over a noisy communication channel may be considered to be corrupted by a mixed type of noise.

2.5 Colour Impulse Noise Model

Modelling of impulse noise in a colour image presents a difficult task because of the interdependence of the channels. Many researchers consider each channel to be corrupted independently. A few alternate models that consider the interdependence between the channels are available. Pitas *et al*. have proposed a probability model for a two-channel case [46]. Viero *et al*. have used an impulse noise model for a three-channel signal [35,47-51]. Their model is given by

$$\boldsymbol{x}(i) = \begin{cases} \boldsymbol{s}(i) & \text{with probability } (1-p) \\ (n_1(i), s_2(i), s_3(i)) & \text{with probability } p_1 p \\ (s_1(i), n_2(i), s_3(i)) & \text{with probability } p_2 p \\ (s_1(i), s_2(i), n_3(i)) & \text{with probability } p_3 p \\ (n_1(i), n_2(i), n_3(i)) & \text{with probability } p_a p \end{cases} \tag{2.9}$$

where $\boldsymbol{x}(i) = (x_1(i), x_2(i), x_3(i))$ is the noisy vector signal, $\boldsymbol{s}(i) = (s_1(i), s_2(i), s_3(i))$ is the noise-free colour vector signal and $n_j(i)$, (where j = 1, 2 and 3) is the impulse noise that may be either fixed-valued or random-valued, p_1, p_2 and p_3 are the single interchannel noise factors, and p_a = 1 - p_1 - p_2 - p_3, and $p_1 + p_2 + p_3 \leq 1$ is the triple interchannel noise factor.

Thus the noise model in (2.9) assumes the channels to be corrupted in a single channel or all channels simultaneously.

2.5.1. General Colour Impulse Noise Model

We propose to generalise the impulse noise model of Equation 2.9. Suppose a vector pixel is corrupted by impulse noise with a probability p. Given that the vector pixel is corrupted, we assume it to be corrupted in a single channel, two channels or all three channels. Probabilities assigned to all the cases of corruptions are as follows:

$$\boldsymbol{x}(i) = \begin{cases} \boldsymbol{s}(i) & \text{with probability } (1-p) \\ (n_1(i), s_2(i), s_3(i)) & \text{with probability } p_1 p \\ (s_1(i), n_2(i), s_3(i)) & \text{with probability } p_2 p \\ (s_1(i), s_2(i), n_3(i)) & \text{with probability } p_3 p \\ (n_1(i), n_2(i), s_3(i)) & \text{with probability } p_4 p \\ (n_1(i), s_2(i), n_3(i)) & \text{with probability } p_5 p \\ (s_1(i), n_2(i), n_3(i)) & \text{with probability } p_6 p \\ (n_1(i), n_2(i), n_3(i)) & \text{with probability } p_a p \end{cases} \tag{2.10}$$

where $\boldsymbol{s}(i) = (s_1(i), s_2(i), s_3(i))$ is the noise-free colour vector signal and $n_j(i)$, (where j = 1, 2 and 3) is the impulse noise that may be either fixed-valued or random-valued, p_1, p_2 and p_3 are the single interchannel noise factors, p_4, p_5 and p_6 are the double interchannel noise factors and p_a = 1 - p_1 - p_2 - p_3 - p_4 - p_5 - p_6 with $p_1 + p_2 + p_3 + p_4 + p_5 + p_6 \leq 1$ is the triple interchannel noise factor. If we set $p_4 = p_5 = p_6 = 0$, we get the colour impulse noise model used by Viero *et al.*

2.6 Mathematical Formulation of the Median Filter

As pointed out earlier, the output of the standard median filter is the median of the data inside a moving window of pre-determined length. Consider a set of pixels $\boldsymbol{X}_{\mathrm{W}} = \{x_1, x_2, ..., x_N\}$ in a square window of length $N = 2L + 1$, $L \geq 0$. The pixels in the window are indexed from left to right and top to bottom with the centre pixel $x = x_{\frac{N+1}{2}}$.

The set of rank-ordered pixels including the central sample x is then given by

$$\boldsymbol{X}_{\mathrm{R}} = \{x_{(1)}, x_{(2)}, ..., x_{(N)}\} \tag{2.11}$$

such that $x_{(1)} \le x_{(2)} \le ... \le x_{(N-1)} \le x_{(N)}$. The output of the standard median (SM) filter is given by

$$x_{\mathrm{SM}} = x_{\left(\frac{N+1}{2}\right)} \tag{2.12}$$

The median filter has become very popular as a method of removing impulse noise from images while maintaining edges with considerably less blurring than linear filters of same size. Its implementation is very simple and the underlying principle of operation is easily understood. It is inherently nonlinear and so the powerful techniques that permit the efficient design of linear filters are not applicable to it. It is quite difficult to analyse the performance of the filter on different types of signal. Its performance as an image-processing tool has been studied from deterministic and statistical points of view. An overview of these techniques may be found in [1, 2, 29].

2.6.1 Median as a Signal Estimator

That the median is a robust estimator of the centre of a distribution has been extensively studied by statisticians. The median filter performs well as an estimator of a signal with heavy-tailed probability distributions, whereas its performance is poor for light-tailed distributions [36,57].

Consider a signal s corrupted by an additive noise v so that

$x = s + v$

Suppose $x_1, x_2, ..., x_N$ are N samples of x. If v is zero-mean Gaussian, *the maximum likelihood estimator* (*MLE*) of s is given by

$$\hat{s}_{\mathrm{MLE}} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

which is the sample mean of the sequence. Thus the sample mean is the *MLE* of a signal corrupted with zero-mean Gaussian noise.

On the other hand, if v is Laplace distributed [58], then the *MLE* of s is given by

$\hat{s}_{\text{MLE}} = \text{median}(x_1, x_2, ..., x_N)$

Thus the sample median is the *MLE* of a signal corrupted with of a Laplace probability distribution function (pdf). In general, the median filter works well in filtering noise of the pdf with a longer tail. Note that the Laplace pdf has a longer tail compared to the Gaussian pdf. A typical pdf of an impulse corrupted signal x is as shown in Figure 2.1. Because the tails of the pdf do not decay in this case, the median filter works better in suppressing the impulse noise.

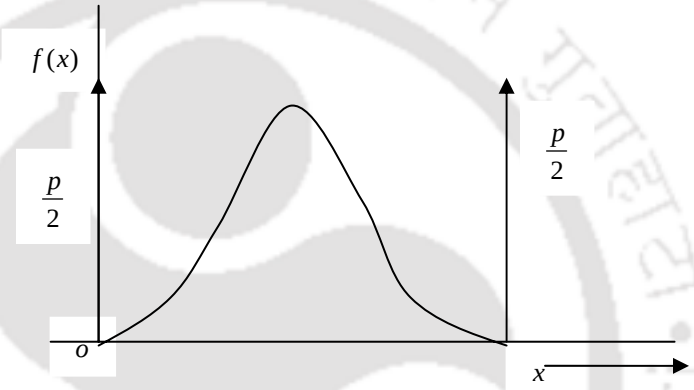


Figure 2.1 Probability density function of an impulse corrupted signal.

2.6.2 Statistical Analysis of Median Filter Performance

The good impulse-noise filtering and edge-preservation capabilities of the median filter can be analysed with the help of the models for the impulse noise and the image intensities. The median of N independent and identically distributed random variables $x(i), i = 1, 2, ..., N$ each having a distribution function $F(x)$ is known to have the probability distribution function $F_{(\frac{N+1}{2})}(x)$ given by [1]

$$F_{(\frac{N+1}{2})}(x) = \sum_{k=\frac{N+1}{2}}^{N} \binom{N}{k} F(x)^k (1-F(x))^{N-k} \qquad (2.13)$$

This expression can be used to evaluate the failure of the median filter in impulse noise.

Pitas *et al*. [1] have used this model to show numerically that the probability of impulse noise in the median filter output is very less for low percentages of noise. For example, consider an image consisting of a constant background c corrupted by the one-sided impulse in Equation 2.2. The probability of correct reconstruction in this case is given by

$$P\left(\left\{x_{(\frac{N+1}{2})}=c\right\}\right)=F_{(\frac{N+1}{2})}(c)$$
$$=\sum_{k=\frac{N+1}{2}}^{N}\binom{N}{k}(1-p)^k p^{N-k}$$
$$=\sum_{k=0}^{\frac{N+1}{2}}\binom{N}{k}p^k(1-p)^{N-k}$$

For $p=0.5,$ the probability of correct reconstruction is 50% irrespective of the size of the filter window.. Following results give insight into performance of the median filter:

(1) In the case of the Laplacian pdf, the sample median has a lower asymptotic variance compared to the sample mean.

(2) In the case of the Gaussian pdf, the sample mean has a lower asymptotic variance compared to the sample median.

Thus the median is a robust estimate of location with respect to outliers inside the filtering window.

2.6.3 Deterministic Analysis of Median Filter Performance

In the deterministic analysis of median filters, considerable importance is given to a class of signals called the *root* signals that are invariant under median filtering. The output of repeated median filtering converges to a root signal. Following definitions [56,60-65] are important for deterministic analysis of the median filter. We consider a sequence $\boldsymbol{X}_{\mathrm{W}}=\{x_1,x_2,...,x_N\}$ of length $N = 2L + 1,$ where L is a nonnegative integer.

A *constant neighbourhood* is at least $L + 1$ consecutive points all of which are identically valued.

A signal x_n is called *monotonic* if $x_i \le x_j$ or $x_i \ge x_j$ for every $i < j$.

A signal is *locally monotonic of degree* d ($LOMO-d$) if it is monotonic within each interval of length d.

An *edge* is a monotonic region between two constant neighbourhoods having different values.

An *impulse* is a set of points whose values are different from the surrounding regions and whose surrounding regions are identically valued constant neighbourhood.

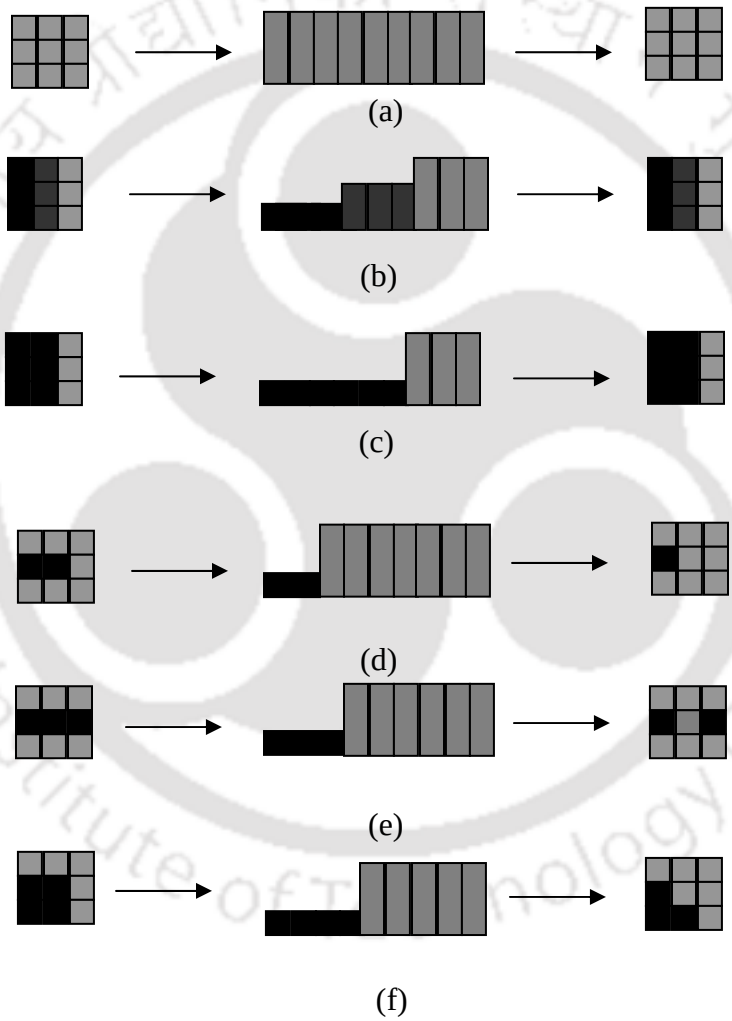


Fig. 2.2 Some important roots and non-roots: (a) Constant signal, (b) Ramp signal, (c) Step edge, (d) End of a thin line, (e) One pixel wide line and (f) Corner.

These definitions are extended to two-dimension for analysing root signals of two-dimensional median filtering. A number of theorems give the requirements for roots of a

two-dimensional signal. Figure 2.2 illustrates the roots and non-roots of the two-dimensional median filters as shown by Moore and Mitra [16]. Thus constant signals, ramp signals and step edges are passed unchanged by the median filter in the absence of impulse noise in the window, while ends of lines, one pixel wide lines and corners are removed by it.

From the statistical and deterministic analyses of median filters, some interesting properties [1, 2, 56, 57] may be stated as follows:

1. The median filter preserves edges and constant neighbourhoods.
2. Intensity oscillations with a period less than the window size are smoothed.
3. The median filter changes the image intensity mean value if the spatial noise distribution in the image is not symmetrical within the window.
4. Given a symmetrical window shape, the median filter preserves the location of edges.
5. The shape of the window chosen for a median filter may affect the processing results.
6. Common features such as end of lines, one-pixel wide lines and corners are destroyed by median filters.

The disadvantage of the median filter is that it destroys fine details in the image. Anything relatively small in size compared to the size of the window will have minimal affect on the value of the median, and so will be filtered out. In other words, the median filter cannot distinguish fine detail from noise. This is an obvious disadvantage if one is interested in preserving the fine details while smoothing noise.

2.7 Important Modifications of the Median Filter

To improve upon the performance of the standard median filter, a number of modifications have been suggested. This section gives a brief introduction on the mathematical formulation of these filters. The modifications considered are the centre-weighted median filter, the *rank-conditioned median filter*, the *signal-dependent rank-ordered mean filter*, the *tri-state mean filter* and *Chen and Wu's adaptive centre-*

weighted median filter respectively. These modifications will be needed in developing the algorithms in the subsequent chapters.

2.7.1 Centre-weighted Median Filter

A straightforward way of modifying the median filter is to bias the filter's output so that the pixel at the centre of the window is more strongly represented that is, the centre pixel is given more weight than the other pixels in the window before determining the median. This has the effect of preferentially preserving the centre pixel. The median filter with more weight to the centre pixel is called the centre-weighted median (CWM) filter [10]. It gives more weight to the centre pixel by replicating it by an integer number of times in the observation set. It is quite reasonable to give emphasis to the central sample, because it is one that is the most correlated with the desired estimate and it should not disturb if it is not corrupted.

Consider a set of pixels $\boldsymbol{X}_{\mathrm{W}} = \{x_1, x_2, ..., x_N\}$ in a square window of length $N = 2L + 1, L \geq 0$. Suppose the centre pixel is repeated w times so that w is the centre-weight, assumed to be odd. The extended set is given by

$$\boldsymbol{X}_{\mathrm{E}} = \{x_1, x_2, ..., x_{N_1}\} \tag{2.14}$$

where $N_1 = N + w - 1$. The elements of the extended set are arranged in ascending order of magnitudes to form an extended set of rank-ordered set

$$\boldsymbol{X}_{\mathrm{RE}} = \{x_{(1)}, x_{(2)}, ..., x_{(N_1)}\} \tag{2.15}$$

The output of the centre-weighted median filter is the median of the extended set of rank-ordered pixels and it is given by

$$x_{\mathrm{CWM}} = x_{\left(\frac{N_1+1}{2}\right)} \tag{2.16}$$

The centre-weighted median filter has the effect of preferentially preserving the centre pixel in the window, so both *fine detail and noise are more preserved*. More importance is always given to the centre pixel whether it is the healthy pixel or not.

2.7.2 Rank-conditioned Median Filter

A simple extension of the median filter that has the ability to preserve image features is the *rank-conditioned median* (RCM) filter. It passes the centre pixel unaltered if the pixel belongs to a set of healthy pixels in the window, known as the *trimming set* [12-14]. A trimming range is formed from the ranks of the elements of the rank-ordered set in Equation 2.11 by excluding the ranks of the extreme values in the set. Extreme values are assumed to be noisy pixels. The trimming range lies between i and $N+1-i$, where $1<i\le\frac{N+1}{2}$ and i represents the lower limit of the trimming range. The centre pixel is healthy if its rank lies in the trimming range. Thus, the output of the rank-conditioned median filter is given by

$$x_{\mathrm{RCM}}=\begin{cases}x & \text{if } \ i\le \mathrm{rank}(x)\le N+1-i\\ x_{\left(\frac{N+1}{2}\right)} & \text{otherwise}\end{cases} \tag{2.17}$$

It is usually true that extreme values are corrupted pixels, but these pixels may be healthy if there is no corrupted pixel in the window. No mechanism is included in the rank-conditioned median filter to check whether extreme values are healthy or not.

2.7.3 Signal-dependent Rank-ordered Mean Filter

An alternative filter effective in removal of fixed-valued and random-valued impulse noise is the signal-dependent rank-ordered mean (SD-ROM) filter [3,15,16]. It considers all pixels but the centre one in a filtering window for rank-ordering. An impulse is detected by thresholding the differences between the current pixel and each of the one half of the ranks. The detected impulse is replaced by the mean of two central ranks.

Let $\boldsymbol{X}_r=\{x_{(1)},x_{(2)},...,x_{(N-1)}\}$ be the set of the rank-ordered pixels excluding the centre pixel such that $x_{(1)}\le x_{(2)}\le ...\le x_{(N-1)}$. The rank-ordered mean is given by

$$x_{\mathrm{ROM}}=\frac{x_{\left(\frac{N-1}{2}\right)}+x_{\left(\frac{N+1}{2}\right)}}{2} \tag{2.18}$$

The rank-ordered mean is thus the average of two central ranks. The SD-ROM filter finds the rank-ordered differences by the relation

$$\psi_k = \begin{cases} x_{(k)} - x & \text{if } x \le x_{\text{ROM}} \\ x - x_{(N-k)} & \text{otherwise} \end{cases} \tag{2.19}$$

A set of thresholds in ascending of magnitudes is considered to compare with the rank-ordered differences. If any one order of these differences is greater than the corresponding threshold, the centre pixel is declared corrupted and replaced by the rank-ordered mean. The output of the SD-ROM filter for a 3×3 window is given by

$$x_{\text{SD-ROM}} = \begin{cases} x_{\text{ROM}} & \text{if } \psi_k > \theta_k \quad \text{for any } k = 1,2,3,4 \\ x & \text{otherwise} \end{cases} \tag{2.20}$$

where $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ is the set of thresholds. The SD-ROM filter has very good performance in removal of both fixed-valued and random-valued impulse noise. It replaces the centre pixel if one of the rank-ordered differences exceeds the corresponding pre-defined threshold. However, the presence of an impulse not at the centre may trigger the SD-ROM filter. This is illustrated below.

Example 2.1: let $\boldsymbol{X}_{\text{W}} = \{x_1, x_2, ..., x_9\}$ be the set of pixels in a 3×3 window, where $x_1 = 100$, $x_2 = 100$, $x_3 = 100$, $x_4 = 100$, $x_5 = 120$, $x_6 = 120$, $x_7 = 120$, $x_8 = 120$, $x_9 = 255$ and x_5 is the centre pixel. It is a case of the presence of an impulse not at the centre of the window. Values of differences d_1, d_2, d_3, and d_4 are 0, 0, 0 and 135 respectively. The centre pixel will be replaced by 110. It is an example of false impulse noise detection.

2.7.4 Tri-state Median Filter

It has been shown in [10] that the performance of the CWM filter with a larger centre weight is superior to the one with a smaller centre weight in detail preservation, but inferior in noise reduction. The standard median filter on the other hand is good in noise

removal but poor at detail preservation. The tri-state median (TSM) [22] filter tries to achieve a balance of the trade-off between detail preservation and impulse noise suppression. It replaces the corrupted centre pixel either by the median or centre-weighted median. The output of the TSM filter is given by

$$x_{\mathrm{TSM}} = \begin{cases} x & \text{if } d \le \theta \\ x_{\mathrm{CWM}} & \text{if } d_w \le \theta < d \\ x_{\mathrm{SM}} & \text{if } d_w > \theta \end{cases} \tag{2.21}$$

where θ is a predetermined threshold, $d = |x - x_{\mathrm{SM}}|$, $d_w = |x - x_{\mathrm{CWM}}|$ and $d \ge d_w$. The output is switched between those of the identity, standard median and centre-weighted median filters. The TSM filter gives very good performance for the random-valued impulse noise, but poor performance for the fixed-valued one [17,27].

2.7.5 Chen and Wu's Adaptive Centre-weighted Median Filter

The CWM filter gives more importance to the centre pixel. Particularly, if the centre weight $w \ge L$, the CWM filter becomes an *identity filter* and outputs the centre pixel as it is. One can devise a centre weighted median (CWM) filter that can be varied over the range from the median filter ($(w = 1)$ to the identity filter just by varying the central weight. The function of the CWM accordingly varies from strong removal of impulse noise and image details to leaving the centre pixel unchanged. Chen and Wu's *adaptive centre- weighted median filter* [17] varies the centre's weight from one point to the next by a technique designed to distinguish noise from fine details. It computes the differences between the centre pixel and the outputs of the CWM filter with varied centre weights as follow:

$$D_k = |\hat{x}^w - x| = |\hat{x}^{2k-1} - x| \tag{2.22}$$

where $\hat{x}^w$ is the output of the CWM filter for $w = 2k-1$ and $k = 1, 2..., L-1$. A set of thresholds is used to compare with these differences. If any one of these differences is bigger than the corresponding threshold, then the centre pixel is declared corrupted. The filter is given by

$$x_{\text{CHEN}} = \begin{cases} x_{\text{SM}} & \text{if } \exists k, D_k > T_k \\ x & \text{otherwise} \end{cases} \tag{2.23}$$

where T_k, $k = 1, 2..., L-1$ are the pre-decided thresholds. Chen and Wu propose to estimate the thresholds by the relation

$$T_k = s.MAD + \gamma_k \tag{2.24}$$

where MAD is the median of the absolute deviations from the median given by the relation

$$MAD = \text{median}(\{|x_i - x_{\text{SM}}| \;\; | i = 1, 2, ..., N\}) \tag{2.25}$$

and γ_k and s are the constants. It has been reported that good results could be obtained by using $[\gamma_1, \gamma_2, \gamma_3, \gamma_4] = [55, 40, 25, 15]$ for fixed-valued impulse noise and $[\gamma_1, \gamma_2, \gamma_3, \gamma_4] = [40, 25, 10, 5]$ for random-valued impulse noise and $0 \le s \le 0.6$ for both types of noise.

As a concluding remark on this chapter, we can say that the median filtering is still an active area of research. There is scope for improvement of the performance of the scalar and vector median filters. The following chapters will devote to new developments in both types of median filters.

Chapter 3

Rank-conditioned and Threshold Median Filters for Gray-scale Images

3.1 Introduction

An efficient impulse noise detection scheme is needed to filter out the impulse noise without damaging the healthy pixels. The scheme should have high rate of impulse noise detection (*true hit*). While wrong detection of the healthy pixel as a noise (*false hit*) and subsequent filtering results in unnecessary smoothing, wrong detection of an impulse to

be a healthy pixel (*miss hit*) leaves the noise unfiltered. An ideal detection mechanism should minimize both these types of errors.

The existing impulse noise detection techniques are based on the following broad principles:

(1) The rank of the centre pixel with respect to the other pixels in the filtering window is an indicator for the presence of an impulse. If this rank is one of the extreme ranks, the current pixel is likely to be corrupted by an impulse. This principle is utilized in the rank-conditioned median filtering [12, 13].

(2) An impulse noise may be considered as an outlier and techniques for outlier detection in statistical data may be applied to detect an impulse. The absolute deviation from the median has been used to detect outliers [2]. Thus the absolute difference between the centre pixel and the median of the pixels in the window is a measure of the corruption of the central pixel by an impulse. The switching scheme I [28] of Sun and Neuvo is based on this principle.

(3) The differences between the centre pixel and the centre weighted medians with varying centre weights can indicate the presence of an impulse. Thresholds on the absolute differences between the centre pixel and the centre-weighted medians have been utilized for better impulse noise detection in Chen and Wu's adaptive centre-weighted median filters [17].

(4) The difference between the centre pixel and the remaining rank-ordered pixels can indicate the presence of an impulse. Thresholds on these rank-ordered differences have been utilized for better impulse noise detection [3,15,16].

(5) The tri-state median filter [22] considers the outputs of both the SM filter and the CWM filter for detecting an impulse. Suppose d = absolute difference between the SM filter output and the centre pixel and d_w = absolute difference between the output of the CWM filter with weight w and centre pixel. If d is less than or equal to a user-specified threshold T, then the centre pixel is decided to be uncorrupted and left unaltered. If d_w is greater than T, then the centre pixel is decided to be corrupted and replaced by the SM output. A third state is

defined corresponding to $d_w \le T < d$. The pixel is decided likely to be a feature point in this case and replaced by the CWM output.

We propose to apply multiple detection schemes simultaneously in order to maximize the true hit and minimize the false hit rates. The centre pixel is declared corrupted if all the detection schemes decide it to be an impulse. We have to choose the detection schemes with very high individual rates of true hit. Another important concern is that such a combination of multiple detection schemes should not be computationally too expensive. This motivates us to simultaneous application of two detection schemes only. The following detection schemes are considered:

(a) Rank conditioning

(b) Thresholding the absolute difference of the centre pixel from the median

(c) Thresholding the absolute differences of the centre pixel from centre-weighted medians and

(d) Thresholding the absolute differences between the centre pixel and other rank-ordered pixels

The first detection scheme decides about the impulse on the basis of the rank of the current pixel. The second scheme uses only one threshold, whereas the last two require multiple thresholds. Once the pixel is detected as corrupted, the median of the pixels in the window can be used as an estimate of the true value of the central pixel. A better estimator should not include the noisy pixel in estimation of the median. Such an estimator is the rank-ordered mean, which is the average of middle ranks when the centre pixel is not used in ranking.

3.2 Formulation

Consider a pixel x at the centre of a square window W of size $n \times n$ containing $N = n^2$ pixels where n is odd. A pixel in the window is denoted by x_i, $i = 1, 2, ..N$ where x_1 is the upper-left and x_N is the lower-right pixels in the window. The notation is illustrated for a 3×3 window in the following figure.

Suppose the set $\boldsymbol{X}_{\mathrm{R}}$ of the rank-ordered pixels is given by

x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

Figure 3.1 Pixels in a 3×3 window.

$$\boldsymbol{X}_{\mathrm{R}} = \{x_{(1)}, x_{(2)}, ..., x_{(N)}\} \tag{3.1}$$

The standard median corresponds to the middle rank and is given by $x_{\mathrm{SM}} = x_{\left(\frac{N+1}{2}\right)}$. The order statistics $x_{(1)}$ and $x_{(N)}$ are the extreme values in the set and are probably the corrupted pixels if their values are either very small or big in comparison with other pixels in the window. The centre pixel that forms a part of the image feature usually appears at one of the extremes after ranking.

The *trimming range* [14] gives the range of ranks of the healthy pixels in the window. For the balanced signal and noise distributions, the trimming range may be assumed to be symmetrical and lie between i and $N+1-i$, where $1<i\le\frac{N+1}{2}$. The value of i in the present case depends on the percentage of impulse noise, and whether the impulse is fixed-valued or random-valued noise and is to be determined experimentally.

The pixels whose ranks are in the trimming range are likely to be healthy and others are more likely corrupted by impulse noise. The rank-conditioned median filter declares the centre pixel to be corrupted with the impulse noise when its rank does not belong to the trimming range. Thus

$$x \textbf{ is an impulse if } \textbf{rank}(x) < i \text{ or } \textbf{rank}(x) > N+1-i \tag{3.2}$$

It is interesting to find the efficiency of this scheme in detection of impulse. As shown experimentally in a later section, rank-conditioning has the remarkable percentage of true hits both for fixed-valued and random-valued impulse noise at a wide range of noise ratios. The number of false hits is also large because of feature points naturally

occupying the extreme ranks and partly because of improper selection of the trimming range in the case of random-valued impulse. Rank-conditioning is considered as the *primary* step for the detection of impulse noise from images in our proposed methods for its computational simplicity and high percentage of true hits.

As noted earlier, a healthy centre pixel may appear at one of extremes after ranking. Consider the pixels representing the end of a thin line in a uniform background in Figure 3.2. The darkened feature pixels are distinct from the rest of the pixels and appear at on extreme after ranking. In such a case, the above test may trigger to replace the centre pixel, which is a part of the feature, by a non-feature pixel.

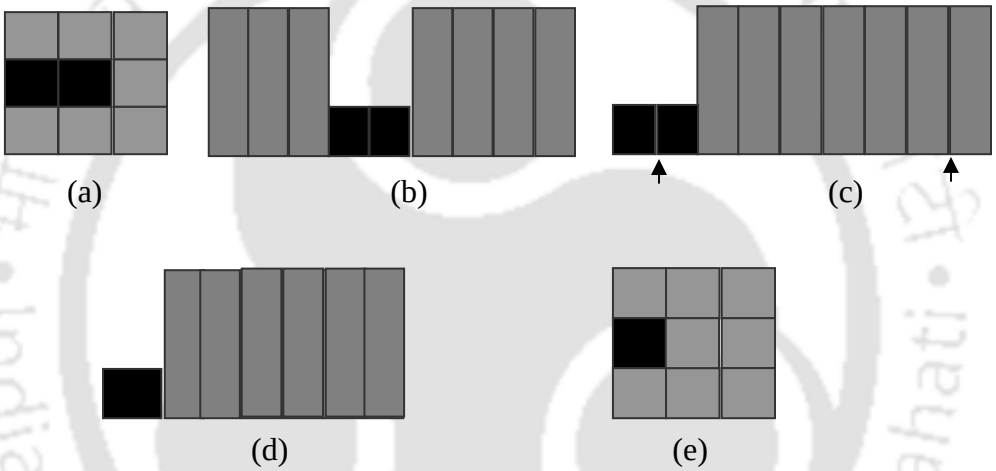


Figure 3.2 First stage of impulse detection in proposed methods: (a) end of a line in a 3×3 window, (b) set of pixels, (c) set of rank-ordered pixel, (d) trimming range and (e) result of filtering.

A *secondary step* is incorporated into the impulse detection procedure to reduce the problem of false hits. In this step, a secondary impulse detection procedure with very high true hit rate is applied. The *joint probability* that a healthy pixel will be wrongly detected to be an impulse by both the primary and secondary steps is considerably lower than the probability of being wrongly detected by each of the steps. In other words, the probability of false hit is reduced by simultaneous application of the primary and secondary steps. The pixel will be declared as an impulse if it is simultaneously detected to be an impulse both by the primary and secondary steps. As the secondary step has

very high true hit rate, the joint probability of correctly detecting the impulse by both these steps will be still high.

Three different methods are developed for the detection of impulse noise by integrating the detection steps of three of the popular impulse noise filtering techniques mentioned in Section 3.1. Each of these methods involves rank-conditioning and one of the thresholding conditions discussed earlier. The filter-class based on these three detection schemes will be referred to as rank-conditioning and thresholding (RCT) filters. The following three RCT filters are proposed:

RCT- I Rank-conditioning and thresholding the absolute difference of the centre pixel from the median,

RCT-II Rank-conditioning and adaptive thresholding of the absolute differences of the centre pixel from the CWM filter outputs for multiple weights and

RCT-III Rank-conditioning and thresholding of the absolute differences of the centre pixel from other rank-ordered pixels.

The detection scheme of RCT filters are explained in the following subsections.

3.2.1 RCT- I - Rank-conditioning and Thresholding the Absolute Difference from the Median

The median of the set of pixels in the window is a healthy pixel as long as the total number of the corrupted pixels in the window is less than 50% [5]. As pointed out, the impulse may be considered as an outlier and the absolute difference between the centre pixel and the median may be considered as an impulse detector. This difference is normally big for the pixel corrupted by impulse noise, while it is not so big for the pixel that forms the image feature and small for a healthy non-feature pixel. The detection scheme considers the absolute difference of the centre pixel from the median in the rank-ordered data set as follows:

Suppose

$$d = | x - x_{SM} | \tag{3.3}$$

For a pre-defined threshold θ,

$$x \textbf{ is an impulse if } d \geq \theta \tag{3.4}$$

Combining conditions in Equations 3.2 and 3.4, we have the following decision rule

$$x \textbf{ is an impulse if } (\textbf{rank}(x) < i \textbf{ or rank}(x) > N+1-i) \textbf{ and } d \geq \theta \tag{3.5}$$

The above conditions ensure that the corrupted pixel is different from all other pixels in the window by a large value. The efficiency of the impulse detection scheme depends on the selection of the threshold θ. An optimality criterion will be based on maximizing *PSNR* value that will ensure better subjective quality. The presence of a thin line, an end of a thin line or a corner in the window may trigger the RCT-I scheme to detect the centre pixel as an impulse if θ is small.

3.2.2 RCT- II - Rank-conditioning and Adaptive Thresholding of the Absolute Differences from the CWM Filter Outputs for Multiple Weights

Feature pixels in the image should be preserved as far as possible. One such attempt is to use the centre-weighted median (CWM) filter by giving more weight to the centre pixel. The approach tries to save the centre pixel if it is not an impulse. However, it produces adverse effect at high impulse noise ratios, because noise pixels are emphasised [22]. This results undesired blotches in the output. The tri-state median filter [22] and Chen and Wu's adaptive centre-weighted median filter [17] address this problem. We consider the impulse detection scheme proposed in the later. It uses multiple thresholds on the absolute differences of the centre pixel and the CWM filter outputs. The centre pixel is decided to be an impulse if any of these absolute differences is greater than the corresponding threshold.

Consider the absolute differences in Equations 2.21 and 2.22, given by

$$D_k = | \hat{x}^{2k-1} - x |$$

where $\hat{x}^{2k-1}$, $k = 1,2,..,\frac{N-1}{2}$ denotes the CWM filter output with centre weight $2k-1$.

x **is decided to be an impulse** if $\exists k \in \{1,2,...,\frac{N-1}{2}\}$ such that

$$D_k \geq T_k \tag{3.6}$$

where the threshold $T_k = s.MAD + \gamma_k$, MAD represents median$\{|x_i - x_{SM}|, i = 1,2,..,N\}$ as already shown in Equation 2.24 and γ_k and s are the constants. These constants are empirically chosen in [17] to be $[\gamma_1, \gamma_2, \gamma_3, \gamma_4] = [55, 40, 25, 15]$ for fixed-valued impulse noise and $[\gamma_1, \gamma_2, \gamma_3, \gamma_4] = [40, 25, 10, 5]$ for random-valued impulse noise and $0 \leq s \leq 0.6$ irrespective of the types of impulse noise.

Combining the conditions in Equations 3.2 and 3.6, the following rule is developed:

x **is decided to be an impulse**
if (**rank**$(x) < i$ **or rank**$(x) > N+1-i$) **and**
$\exists k \in \{1,2,...,\frac{N-1}{2}\}$ **such that** $D_k > T_k$ (3.7)

3.2.3 RCT-III - Rank-conditioning andThresholding of the Absolute Differences of the Centre Pixel from Rank-ordered Pixels

This RCT filter is based on the SD-ROM filter developed by Mitra *et al*. [3,15,16]. In this filter, the centre pixel is excluded from ranking and thus the remaining $N-1$ pixels are used. Consider the set $\boldsymbol{X}_r$ of $N-1$ rank-ordered pixels given by

$$\boldsymbol{X}_r = \{x_{(1)}, x_{(2)}, ..., x_{(N-1)}\} \tag{3.8}$$

An impulse is detected by thresholding the rank-ordered differences ψ_k for $k = 1,2,...,\frac{N-1}{2}$, between the current pixel and $\frac{N-1}{2}$ ranks in $\boldsymbol{X}_r$. The rank-ordered difference is given by

$$\psi_k = \begin{cases} x_{(k)} - x & \text{if } x \leq x_{ROM} \\ x - x_{(N-k)} & \text{otherwise} \end{cases} \tag{3.9}$$

Each rank-ordered difference ψ_k is compared with a pre-defined threshold θ_k. It is further assumed that $\theta_1 < \theta_2 < .... < \theta_{\frac{N-1}{2}}$. According to the SD-ROM filter decision rule,

$$x \textbf{ is an impulse if } \exists k \in \{1,2,...,\frac{N-1}{2}\} \textbf{ such that } \psi_k > \theta_k \tag{3.10}$$

Combining the decision rules in Equations 3.2 and 3.10, we have the following impulse-detection rule for the RCT-III filter:

$$x \textbf{ is an impulse if } (\textbf{rank}(x) < i \textbf{ or rank}(x) > N+1-i) \textbf{ and } \exists k \in \left\{1,2,...,\frac{N-1}{2}\right\} \textbf{ such that } \psi_k > \theta_k \tag{3.11}$$

3.3 Filtering of the Corrupted Pixels

Once a pixel is found to be corrupted by an impulse, a median filter can be applied to filter it. An effective way to do this is to replace the corrupted centre pixel by rank-ordered mean x_{ROM} [3,15,16] of the pixels in the window excluding the centre pixel. Inclusion of the centre pixel to obtain the median value will not affect the performance of the proposed filter in filtering out the corrupted pixels in a homogeneous region that contains more or less similar pixels. But the performance of the filter will be worse in the window that contains image features. This is because the corrupted pixel naturally occurs at one of the extremes after ranking and numbers of healthy pixels to the left of the median and to the right are different. This may lead biasing the filter output towards the undesired pixel. Therefore, the best choice will be the average of two central ranks in the window after excluding the centre pixel. The ROM filter has proved to be successful in restoring images contaminated with higher percentages of impulse noise [3,15,16]. It is reported that for the Lena image at 40% of fixed-valued impulse noise, replacing ROM by the median in the SD-ROM filtering scheme reduces the *PSNR* performance by about 2 dB.

The noisy pixel is replaced by the rank-ordered mean x_{ROM} given in Equation 2.18. Since the centre pixel lies outside the trimming range, we can find the ROM from the sorted N pixels. If x lies to the right of the trimming region, the ROM is found as follows:

$$x_{\text{ROM}} = \frac{1}{2}\left(x_{\left(\frac{N-1}{2}\right)} + x_{\left(\frac{N+1}{2}\right)} \right)$$

On the other hand, if x lies to the left of the trimming region, the ROM is given by

$$x_{\text{ROM}} = \frac{1}{2}\left(x_{\left(\frac{N+1}{2}\right)} + x_{\left(\frac{N+2}{2}\right)} \right)$$

Thus the proposed filters have the following replacement rule

$$\hat{x} = \begin{cases} x_{\text{ROM}} & \text{if } x \text{ is an impulse} \\ x & \text{otherwise} \end{cases} \tag{3.12}$$

where $\hat{x}$ is the final estimate of the current pixel x.

3.4 Detection Performance of Rank-conditioning and Thresholding

As we have noted earlier, rank-conditioning and thresholding based schemes have individually good impulse-noise detection performance. In the following, we show that simultaneous application of rank-conditioning and thresholding improves detection by reducing the probability of detecting a healthy pixel as a noisy pixel (*false hit).* We consider the thresholding scheme of the RCT-I filter in the analysis.

Suppose

$H_0 = \{x \mid x \text{ is an impulse}\},$
$H_1 = \{x \mid x \text{ is not an impulse}\},$
$A = \{x \mid \text{rank}(x) < i \text{ or } \text{rank}(x) > N + 1 - i\}$

and

$B = \{x \mid\mid x - x_{\text{SM}} \mid > \theta\}$

Then the *probability of false hit* in the RCT-I filter output is given by

$$\begin{aligned} P(A \cap B \mid H_1) &= \frac{P(A \cap B \cap H_1)}{P(H_1)} \\ &= \frac{P(A \cap H_1)}{P(H_1)} P(B \mid A \cap H_1) \\ &= P(A \mid H_1) P(B \mid A \cap H_1) \end{aligned}$$

where $P(A|H_1)$ represents the probability of false hit of the RCM filter based detection. Note that $P(B|A\cap H_1)$ represents the probability of B given that x is not an impulse and rank of x outside the trimming range. $P(B|A\cap H_1)$ should be small because a healthy pixel should not deviate much from the median of the pixels in the window. Therefore,

$$P(A\cap B|H_1) \ll P(A|H_1) \tag{3.13}$$

Similarly,

$$\begin{aligned} P(A\cap B|H_0) &= \frac{P(A\cap H_0)}{P(H_0)} P(B|A\cap H_0) \\ &= P(A|H_0)P(B|A\cap H_0) \end{aligned}$$

where $P(A|H_0)$ is *the probability of true hit* of the RCM filter based detection. The quantity $P(B|A\cap H_0)$ is close to unity as there is a high likelihood that the pixel will be distinct from the median under this condition. Therefore, we conclude that

$$P(A\cap B|H_0) \sim P(A|H_0) \tag{3.14}$$

Similar results can be derived for the other two RCT schemes justifying the claim that simultaneous application of rank-conditioning and thresholding improves impulse-noise detection performance.

3.5 Computational Complexity Analysis

An analysis of the computational complexities of the proposed rank-conditioned and threshold filters is carried out. The computational complexities of the SM, CWM, RCM, SD-ROM and Chen and Wu's adaptive centre-weighted median filters are included for comparison. The size of the filtering window is assumed to be $n\times n$ with $N=n^2$ elements in the window. The analysis does not consider the optimal implementation of the filters and is only indicative of the computational overhead associated with each filter.

Note that the computation of the standard median for $N=n^2$ data requires approximately $2n+10$ comparison operations per pixel using the fast histogram based algorithm [65]. In *quick sort* ordering, the number of comparison/swapping operations is

$2n^2 \log_2 n$. The CWM filter requires $2n^2 \log_2 n$ comparisons/swappings for sorting the pixels in the window and $n^2 - 1$ comparisons to find the rank of the centre pixel in the sorted pixels. It also requires $w - 1$ insertion operations, where w is the weight of the centre pixel.

The RCT-I filter involves the following computations:

(1) $2n^2 \log_2 n$ comparisons/swappings for sorting the pixels in the window,

(2) 2 comparisons to check whether the centre pixel belongs to the trimming range,

(3) 2 comparison and 1 subtraction operations to check that $|x - x_{SM}| > \theta$ and

(4) 1 addition and 1 multiplication to find out the ranked-order mean.

Thus the RCT-I filter requires $2n^2 \log_2 n + 4$ comparisons, 2 addition/subtraction and 1 multiplication operations.

The RCT-II filter involves computing $\frac{n^2 - 1}{2}$ numbers of CWM outputs for different weights of the centre pixel. This is efficiently implemented by first ranking n^2 elements in the window and replicating the centre pixel in the rank-ordered array in appropriate number of times. The RCT-II filter involves the following computations:

(1) $2n^2 \log_2 n$ comparisons/swappings for sorting the pixels in the window,

(2) 2 comparisons to check whether the centre pixel belongs to the trimming range,

(3) $n^2 - 1$ comparisons to find the rank of the centre pixel in the window and $w - 1$ insertion operations corresponding to the CWM filter of weight w,

(4) $\frac{n^2 - 1}{2}$ subtraction and $\frac{n^2 - 1}{2}$ comparison operations to compute each of $D_k = |\hat{x}^{2k-1} - x|$,

(5) n^2 subtraction and $2n^2 \log_2 n$ comparison/swapping operations to find the MAD in Equation 3.7,

(6) 1 multiplication and $\frac{n^2 - 1}{2}$ additions to compute each $T_k = s.MAD + \gamma_k$,

(7) $\frac{n^2-1}{2}$ comparisons with $\frac{n^2-1}{2}$ thresholds T_k, $k=1,2,..\frac{n^2-1}{2}$ and

(8) 1 addition and 1 multiplication to find out the ranked-order mean.

Thus RCT-II filter requires $4n^2\log_2 n+2n^2$ comparison/swapping, $2n^2$ addition/subtraction, 2 multiplication and $w-1$ insertion operations.

The RCT-III filter is efficiently implemented by performing the steps of the secondary stage of detection (that involves the SD-ROM filter) first. The computation involved in this stage includes ranking n^2-1 pixels in the window, finding the rank-ordered mean, evaluating the rank-ordered differences and thresholding these differences. To implement rank-conditioning, the centre pixel is inserted in the appropriate place to get n^2 rank-ordered pixels in the window. The RCT-III filter involves the following computations:

(1) $(n^2-1)\log_2(n^2-1)$ comparison/swapping operations to sort n^2-1 pixels in the window,

(2) 1 addition and 1 multiplication to find the rank-ordered mean,

(3) 1 comparison to decide whether to difference the left $\frac{n^2-1}{2}$ or the right $\frac{n^2-1}{2}$ rank-ordered elements,

(4) $\frac{n^2-1}{2}$ subtractions to find $\frac{n^2-1}{2}$ rank-ordered differences,

(5) $\frac{n^2-1}{2}$ comparisons with $\frac{n^2-1}{2}$ pre-decided thresholds,

(6) n^2-2 comparisons and 1 insertion to get n^2 rank-ordered pixels in the window

(7) 2 comparisons to check whether the centre pixel belongs to the trimming range.

Thus the RCT-III filter requires $(n^2-1)\log_2(n^2-1)+\frac{3n^2+1}{2}$ comparison operations, $\frac{n^2+1}{2}$ additions/subtractions, 1 multiplication and 1 insertion operation.

Table 3.1(a) shows the numbers of computations involved in these three RCT filters. It is interesting to compare the above figures against $2n+10$ comparisons involved in the SM filter. Table 3.1(b) shows the number of various operations required by different filters

under consideration. It is seen that the RCT-I filter involves less computation compared to each of the SD-ROM and Chen and Wu's adaptive centre-weighted median filters. The RCT-II filter is marginally costlier than the Chen and Wu's adaptive centre-weighted median filter, and also the RCT-III filter is slightly costlier than the SD-ROM filter. The comparative computational cost per pixel in a 3×3 window for the SM, CWM with $w=3$, RCM, SD-ROM, Chen and Wu's adaptive centre-weighted median, RCT-I, RCT-II and RCT-III filters is given in Table 3.1(c).

Table 3.1(a) Computational complexity of the RCT- I, RCT-II and RCT- III filters

Filters	Additions	Comparisons	Multiplications	Insertions
RCT-I	2	$2n^2\log_2 n+4$	1	---
RCT- II	$2n^2$	$4n^2\log_2 n+2n^2$	2	$w-1$
RCT- III	$\frac{n^2+1}{2}$	$(n^2-1)\log_2(n^2-1)+\frac{3n^2+1}{2}$	1	1

Table 3.1(b) Computational complexity of various filters

Filters	Additions	Comparisons	Multiplications	Insertions
SM	---	$2n+10$	---	---
CWM	---	$n^2+2n^2\log_2 n-1$	---	$w-1$
RCM	---	$2n^2\log_2 n+2$	---	---
SD-ROM	$\frac{n^2+1}{2}$	$(n^2-1)\log_2(n^2-1)+\frac{n^2+1}{2}$	1	---
Chen and Wu	$2n^2-1$	$2n^2+4n^2\log_2 n-2$	1	$w-1$
RCT- I	2	$2n^2\log_2 n+4$	1	---
RCT-II	$2n^2$	$4n^2\log_2 n+2n^2$	2	$w-1$
RCT-III	$\frac{n^2+1}{2}$	$(n^2-1)\log_2(n^2-1)+\frac{3n^2+1}{2}$	1	1

Table 3.1(c) Total number of operations per pixel in a 3×3 window size

Filters	Additions	Comparisons	Multiplications	Insertions
SM	---	16	---	---
CWM	---	36.53	---	2
RCM	---	30.53	---	---
SD-ROM	5	29	1	---
Chen and Wu	17	73.06	1	6
RCT-I	2	32.53	1	---
RCT-II	18	75.06	2	6
RCT-III	5	38	1	1

Table 3.1(d) Order of the computational complexity in a $n \times n$ window size

Filters	Additions	Comparisons	Multiplications	Insertions
SM	---	$O(n)$	---	---
CWM	---	$O(n^2 \log n)$	---	O(1)
RCM	---	$O(n^2 \log n)$	---	---
SD-ROM	$O(n^2)$	$O(n^2 \log n)$	O(1)	---
Chen and Wu	$O(n^2)$	$O(n^2 \log n)$	O(1)	O(1)
RCT-I	O(1)	$O(n^2 \log n)$	O(1)	---
RCT-II	$O(n^2)$	$O(n^2 \log n)$	O(1)	O(1)
RCT-III	$O(n^2)$	$O(n^2 \log n)$	O(1)	O(1)

The order of the computational complexity of the various filters is given in Table 3.1(d). The tables show that the RCT-I filter has lower computational complexity than the RCT-II and RCT-III filters. The asymptotic complexities of RCT filters are comparable with those of the SD-ROM filter and Chen and Wu's adaptive CWM filter.

3.6 Experimental Results

The proposed algorithms were tested on a number of 8-bit gray-scale images from a test image database. Each of the images is of size 512×512. The results of experiments on the Lena, Mandrill, Airplane, Lake, Pepper, House, Bridge, Goldhill, Miramar, Point Loma, Golden Gate, Terrain, Texmos1 and Texmos3 images are reported here. Fixed-valued and random-valued impulses were artificially injected into these images at various noise ratios. Fixed-valued impulse noise was generated by the two-sided impulse noise model with the noise values of 255 and 0. Random-valued impulse noise in the range 0-255 was generated by the Bernoulli-uniform impulse noise model. The quality of impulse detection was evaluated in terms of the number of corrupted pixels that are detected correctly (*true hit*), healthy pixels that are detected wrongly as corrupted pixels (*false hit*), corrupted pixels that are left undetected (*miss hit*) and the correctly detected percentage (% true hit) of the corrupted pixels. The performance of the proposed algorithms was evaluated in terms of the *visual quality*, the *peak-signal-to-noise-ratio*

(*PSNR*) and the *stability* of the performance of filters on different types of images. The *PSNR* is given by

$$PSNR = 10\log_{10}\left(\frac{I_{\text{MAX}}^2}{MSE}\right) \tag{3.15}$$

where I_{MAX} is the maximum gray level in the original image and *MSE* represents the mean square error between the filtered image and the original image.

Experiment 3.6.1 Detection Performance

The first set of experiments was carried out to study the performance of the detection schemes in identifying the noisy pixels in different test images at different impulse noise ratios. The experimental results on the Lena and Mandrill images in terms of true hit, false hit and miss hit are reported here. The *PSNR* of the corresponding detection-based median filter outputs is also included. In the first stage, each of the following detection schemes was separately applied on the noisy image:

(a) Rank conditioning

(b) Thresholding the absolute difference of the centre pixel from the median

(c) Adaptive thresholding the absolute differences of the centre pixel from centre weighted medians and

(d) SD-ROM filter based thresholding

The trimming range in rank-conditioning and thresholds in the other cases were fixed corresponding to the best performance. Tables 3.2(a) and 3.2(b) show the relative detection performance of the detection schemes on the Lena image at different noise ratios for fixed-valued and random-valued impulses respectively. Tables 3.3 (a) and 3.3 (b) show the same for the Mandrill image. From the tables following observations are made:

(1) Both true hit and false hit are the highest for the *ranking-conditioning* in case of both types of noise.

(2) The percentage of true hit is uniformly very high for all the detection schemes.

(3) False hit is highest for the *ranking-conditioning* and lowest for the *thresholding absolute differences from centre weighted medians*

(4) The *PSNR* is the lowest for rank-conditioning based filtering. Thresholding based filtering gives better *PSNR* performance with the highest value for the SDROM filter output.

In the second-stage, the detection performance of the proposed RCT filters was evaluated. Tables 3.4(a) and 3.4(b) show the detection performance for fixed-valued and random-valued impulse noise respectively on the Lena image. The same on the Mandrill image is shown in Table 3.5 (a) and 3.5(b) respectively. From the tables following observations are made:

Case I: For fixed-valued impulse noise

(1) True hit is high and almost same in all RCT filters.

(2) False hit is the highest for the RCT-I filter and lowest for the RCT-III filter on the Lena image. False hit is almost same in all RCT filters on the Mandrill image.

(3) Miss hit is the highest for the RCT-II filter.

(4) The *PSNR* is the highest for RCT- III filter and lowest for the RCT-I filter on the Lena image. The *PSNR* is almost same in all RCT filters on the Mandrill image.

Case II: For random-valued impulse noise

(1) True hit is the highest for the RCT-I filter and the lowest for the RCT- II filter on the Lena image. True hit is the highest for the RCT-III filter and lowest for the RCT-I filter on the Mandrill image.

(2) False hit is the highest for the RCT-I filter and lowest for the RCT-II filter on the Lena image. False hit is the highest for the RCT-III filter and lowest for the RCT- I filter on the Mandrill image.

(3) Miss hit is the highest for the RCT-II filter.

(4) *PSNR* is the highest for the RCT-III filter and lowest for the RCT-I filter on the Lena image.

(5) *PSNR* is the highest for the RCT-II filter and lowest for the RCT-I filter on the Mandrill image.

It is observed that the true hit percentage of the RCT filters is very high. False hit is considerably reduced because of the simultaneous application of rank conditioning and thresholding. These observations suggest that RCT filters have the ability to distinguish a healthy pixel from the corrupted one.

Impulse noise detection is affected by the presence of feature points in the window. Therefore true hit (and consequently miss hit) depends on the image. For an image with smoothed regions, true hit is higher compared to an image with micro-textured regions. This is illustrated by the results on the Lena and Mandrill images. The same also explains that false hit is lower for an image with fewer details.

Table 3.2 (a) Detection performance on the Lena image for fixed-valued impulse noise

% Impulse noise	**5**	**10**	**15**	**20**	**25**	**30**	**35**	**40**
Total number of corrupted pixels	13107	26214	39321	52429	65535	78643	91749	104860
(a) Rank-conditioning								
True hit	13107	26214	39321	52429	65535	78643	91749	104860
False hit	63416	54671	47275	41095	35235	29555	26111	22344
Miss hit	0	0	0	0	0	0	0	0
% True hit	100	100	100	100	100	100	100	100
PSNR	37.92	37.07	36.28	35.09	34.01	32.62	31.82	31.00
(b) Thresholding the absolute difference from the median								
True hit	13086	25944	38890	52329	65314	78381	91205	104106
False hit	398	631	778	1012	1246	1526	1803	1961
Miss hit	21	270	431	100	221	262	544	754
% True hit	99.73	99.70	99.58	99.59	99.53	99.50	99.38	99.19
PSNR	41.76	38.64	36.18	34.90	33.17	32.07	30.51	29.22
(c) Thresholding the absolute differences from centre weighted medians								
True hit	13107	26214	39321	52429	65535	78063	90685	104042
False hit	263	413	488	703	873	1122	1323	1462
Miss hit	0	0	0	0	0	544	1064	810
% True hit	100	100	100	100	100	99.50	99.23	98.90
PSNR	40.91	38.93	36.34	35.52	34.17	32.00	31.15	29.51
(d) SD-ROM filter								
True hit	13107	26014	38996	52388	65365	78254	90980	103565
False hit	163	219	280	343	410	540	671	764
Miss hit	0	200	325	41	170	389	769	1295
% True hit	100	99.67	99.85	99.71	99.61	99.34	99.13	98.67
PSNR	43.35	40.66	37.84	36.20	34.75	33.03	31.57	30.04

Table 3.2 (b) Detection performance on the Lena for random-valued impulse noise

% Impulse noise	**5**	**10**	**15**	**20**	**25**	**30**	**35**	**40**
Total number of corrupted pixels	13107	26214	39321	52429	65535	78643	91749	104860
(a) Rank-conditioning								
True hit	12391	24193	36405	47682	58753	68864	78588	87574
False hit	140681	130125	119856	110066	100295	91511	83214	75281
Miss hit	716	2021	2916	4747	6782	9779	10161	17286
% True hit	94.54	92.29	92.58	90.95	89.65	87.57	85.66	83.52
PSNR	35.77	34.61	33.46	32.57	31.74	30.62	29.87	28.81
(b) Thresholding the absolute difference from the median								
True hit	10458	20781	30969	41307	51253	61803	72231	81938
False hit	2712	5488	8151	11042	14012	16701	19488	22445
Miss hit	2649	5433	8352	11122	14282	16840	19518	22922
% True hit	79.41	79.11	79.10	78.91	78.53	78.73	78.75	78.50
PSNR	38.03	35.53	33.75	32.20	31.16	29.98	29.11	27.90
(c) Thresholding the absolute differences from centre weighted medians								
True hit	11001	21788	32521	42548	52347	62225	71430	80240
False hit	327	386	494	668	761	850	1043	1286
Miss hit	2106	4426	6800	9881	13188	16418	20319	24620
% True hit	83.93	82.86	82.33	81.24	80.19	87.97	77.79	76.53
PSNR	38.95	36.61	35.15	34.00	32.66	31.61	30.11	28.93
(d) SD-ROM filter								
True hit	11378	22603	33536	44698	54632	65096	75322	84482
False hit	906	950	934	993	1099	1139	1285	1417
Miss hit	1729	3611	5785	7731	10903	13547	16427	20378
% True hit	86.99	86.62	85.57	84.92	83.71	82.92	82.12	80.93
PSNR	39.60	38.03	36.05	34.53	33.20	32.18	30.80	29.72

Table 3.3 (a) Detection performance on the Mandrill for fixed-valued impulse noise

% Impulse noise	**5**	**10**	**15**	**20**	**25**	**30**	**35**	**40**
Total number of corrupted pixels	13107	26214	39321	52429	65535	78643	91749	104860
(a) Rank-conditioning								
True hit	13096	26200	39139	52096	65535	78643	91749	104860
False hit	57724	50246	43355	37510	32318	27888	24230	20772
Miss hit	11	14	182	333	0	0	0	0
% True hit	100	100	100	100	100	100	100	100
PSNR	25.83	25.56	25.24	24.88	24.40	23.84	23.25	22.56
(b) Thresholding the absolute difference from the median								
True hit	12926	25207	38385	51177	63843	75925	88737	100875
False hit	4635	4561	4557	4421	4411	4320	4391	4333
Miss hit	181	1007	936	1252	1692	2718	3012	3985
% True hit	97.53	97.38	97.39	97.14	9705	96.84	96.75	96.31
PSNR	29.16	28.24	27.13	26.30	25.20	24.27	23.41	22.74
(c) Thresholding the absolute differences from centre weighted medians								
True hit	12775	25687	38409	50670	62883	74795	86559	97974
False hit	2318	2412	2447	2627	2776	2675	2815	2936
Miss hit	332	527	912	1759	2652	3848	5190	6886
% True hit	98.14	97.50	97.11	96.61	96.20	95.19	9445	93.67
PSNR	29.48	27.92	26.50	25.48	24.29	23.46	22.77	21.92
(d) SD-ROM filter								
True hit	13107	25998	39005	51336	63848	76399	88638	100235
False hit	4287	4256	4144	4247	4126	4058	4119	3985
Miss hit	0	216	316	1093	1087	2244	3111	4625
% True hit	100	98.77	98.52	98.02	87.57	97.00	96.51	95.51
PSNR	29.26	28.07	27.88	26.12	25.18	24.32	23.44	22.70

Table 3.3 (b) Detection performance on the Mandrill image for random-valued impulse noise

% Impulse noise	5	10	15	20	25	30	35	40
Total number of corrupted pixels	13107	26214	39321	52429	65535	78643	91749	104860
(a) Rank-conditioning								
True hit	11798	23317	34780	46020	56996	67583	77653	87709
False hit	131425	123442	115251	107207	99488	91841	84660	77073
Miss hit	1309	2897	4541	6409	8539	11060	14096	17151
% True hit	90.20	89.36	88.75	87.84	87.03	86.09	8490	83.43
PSNR	23.74	23.47	23.13	22.83	22.40	21.91	21.43	21.17
(b) Thresholding the absolute difference from the median								
True hit	8033	15974	24151	32026	40091	47585	55562	63582
False hit	9258	9026	8948	8538	8356	8154	7223	7742
Miss hit	5074	10240	15170	20403	25444	31058	36187	41278
% True hit	61.26	61.09	61.44	61.42	61.18	60.69	60.64	60.48
PSNR	25.48	24.71	23.84	23.35	22.64	22.14	21.61	21.06
(c) Thresholding the absolute differences from centre weighted medians								
True hit	8494	16872	25474	33085	40368	48118	55054	61765
False hit	5215	5156	5059	4932	4917	4945	4878	4807
Miss hit	4613	9342	13847	19344	25167	30525	36695	43095
% True hit	65.00	64.52	64.18	63.03	62.04	61.21	60.23	59.14
PSNR	27.48	26.23	25.09	24.15	23.50	22.64	21.97	21.41
(d) SD-ROM filter								
True hit	9286	18551	27519	36384	44954	53324	61289	68874
False hit	10406	10084	9800	9428	9010	8714	8448	8007
Miss hit	3821	7663	11802	16045	20581	25319	30460	35986
% True hit	71.00	70.83	69.68	68.92	68.40	67.60	67.16	65.80
PSNR	25.87	25.07	24.43	23.74	23.17	22.61	22.08	21.56

Table 3.4(a) Detection performance of RCT filters the Lena image for fixed-valued impulse noise

% Impulse noise	**5**	**10**	**15**	**20**	**25**	**30**	**35**	**40**
Total number of corrupted pixels	13107	26214	39321	52429	65535	78643	91749	104860
RCT- I								
True hit	13066	26112	39001	52354	64925	78463	91559	104297
False hit	213	294	446	593	712	902	1067	1193
Miss hit	41	102	320	75	611	190	190	563
% True hit	99.69	99.61	99.19	99.86	99.07	99.77	99.79	99.47
PSNR	43.49	40.39	38.28	36.93	35.40	34.16	32.72	31.84
RCT- II								
True hit	13104	26112	39290	52404	65366	77777	90131	102832
False hit	203	338	414	447	562	675	701	756
Miss hit	3	0	31	25	169	866	1618	2028
% True hit	99.96	99.82	99.72	99.53	99.29	98.94	98.35	97.90
PSNR	44.50	41.02	38.81	37.56	35.76	34.13	32.97	31.61
RCT- III								
True hit	13106	26197	39263	52354	65418	78067	90561	103178
False hit	101	116	128	140	168	146	159	155
Miss hit	1	17	58	75	117	576	1188	1682
% True hit	99.99	99.94	99.88	99.80	99.54	99.48	99.00	98.66
PSNR	44.37	41.48	39.36	37.59	36.05	34.43	33.29	31.90

Table 3.4(b) Detection performance of RCT filters on the Lena image for random-valued impulse noise

% Impulse noise	**5**	**10**	**15**	**20**	**25**	**30**	**35**	**40**
Total Number of corrupted pixels	13107	26214	39321	52429	65535	78643	91749	104860
RCT- I								
True hit	12391	24193	36405	47682	58753	68864	78588	87574
False hit	717	1733	2950	4535	6771	9611	13139	17165
Miss hit	716	2021	2916	4747	6782	9779	13161	17286
% True hit	94.54	92.29	92.58	90.95	89.65	87.57	85.66	83.52
PSNR	38.90	36.61	35.19	33.93	32.81	31.72	30.49	29.40
RCT- II								
True hit	10907	21419	31278	41126	50179	58817	66672	74149
False hit	324	411	465	545	626	662	772	869
Miss hit	2200	4795	8043	11303	15356	19826	25077	30711
% True hit	82.80	81.49	79.82	78.58	76.86	74.97	72.56	70.63
PSNR	40.55	37.73	35.57	34.61	32.90	31.71	30.43	29.03
RCT- III								
True hit	11399	22852	33406	43635	53686	63170	72463	81036
False hit	908	889	936	956	965	1000	1057	1279
Miss hit	1708	3362	5915	8794	11849	15473	19286	23824
% True hit	86.76	86.22	84.91	83.70	81.87	80.60	79.38	77.24
PSNR	40.31	37.74	36.35	35.07	33.59	32.65	31.46	29.91

Table 3.5(a) Detection performance of RCT filters on the Mandrill image for fixed-valued impulse noise

% Impulse noise	**5**	**10**	**15**	**20**	**25**	**30**	**35**	**40**
Total number of corrupted pixels	13107	26214	39321	52429	65535	78643	91749	104860
RCT-I								
True hit	12823	25538	38153	50622	63879	76392	88634	100601
False hit	2874	2586	2463	2277	2090	1947	1660	1620
Miss hit	284	676	1168	1807	1656	2251	3115	4259
% True hit	97.44	97.57	97.29	97.19	97.13	96.91	96.65	96.22
PSNR	29.24	27.98	27.09	26.14	25.32	24.23	23.34	22.70
RCT-II								
True hit	12782	25616	38425	50617	62889	75068	86165	98396
False hit	2367	2277	2199	2205	2116	1963	1873	1821
Miss hit	325	598	896	1812	2446	3575	5584	6464
% True hit	97.92	97.52	97.03	96.55	96.07	95.27	94.49	93.50
PSNR	29.71	28.34	26.93	25.89	24.76	23.56	23.11	22.39
RCT-III								
True hit	12926	26056	38930	51673	64015	76544	88289	100686
False hit	3062	2891	2690	2492	2449	2230	2045	1869
Miss hit	181	158	391	756	1520	2099	3460	4174
% True hit	99.10	99.94	98.51	98.12	97.64	97.12	96.38	95.87
PSNR	29.27	28.19	27.14	26.23	25.28	24.33	23.34	22.76

Table 3.5(b) Detection performance of RCT filters on the Mandrill image for random-valued impulse noise

% Impulse noise	**5**	**10**	**15**	**20**	**25**	**30**	**35**	**40**
Total number of corrupted pixels	13107	26214	39321	52429	65535	78643	91749	104860
RCT-I								
True hit	7761	16112	23675	31674	3923	46498	53514	60806
False hit	8524	8027	7713	7537	7268	6967	6724	6492
Miss hit	5346	10102	15646	20755	2629	32145	38235	44054
% True hit	61.49	61.34	60.40	60.05	59.6	59.32	58.36	57.97
PSNR	25.82	24.99	24.08	23.39	22.7	22.24	21.70	21.17
RCT-II								
True hit	8962	17844	26291	34517	4256	49874	56755	63556
False hit	8736	8464	8137	7595	7330	7070	6869	6422
Miss hit	4145	8370	13030	17912	2296	28769	34994	41304
% True hit	67.86	67.87	66.50	65.48	64.6	63.47	61.92	60.67
PSNR	26.45	25.40	24.59	23.81	23.1	22.57	22.00	21.48
RCT-III								
True hit	9251	18580	27163	35627	4392	52062	59587	66792
False hit	10102	9786	9386	9016	8697	8091	7831	7606
Miss hit	3856	7634	12158	16802	2161	26581	32162	38066
% True hit	70.37	69.93	68.98	68.09	67.0	66.17	64.85	63.80
PSNR	26.02	25.20	24.50	23.83	23.1	22.67	22.10	21.60

Experiment 3.6.2: Selection of Trimming Range

This set of experiments was conducted to find out the most appropriate trimming set. Figure 3.3 shows the variation of the *PSNR* of the RCM-filtered Lena and Mandrill images with respect to the impulse-noise percentage for different values of the trimming range index i. It has been observed that the most appropriate value of i that optimises *PSNR* values is 2 for fixed-valued impulse noise. For random-valued noise, we get marginally better *PSNR* performance for $i = 3$. In the rest of the experiments, we shall take $i = 2$ for fixed-valued impulse and $i = 3$ for random-valued impulse. Experimental results on different images suggest that the above values of the trimming range index are independent of the choice of images.

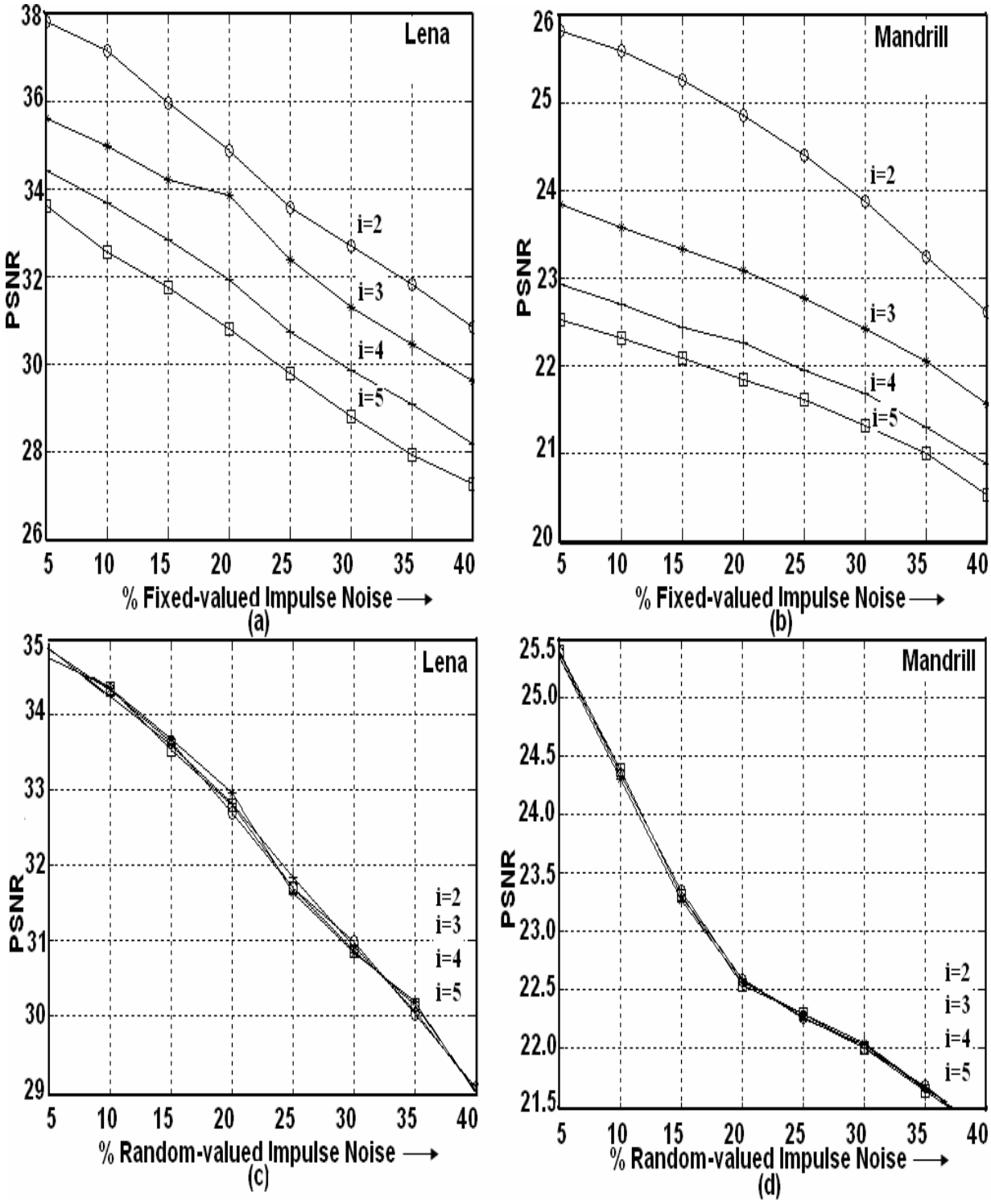


Figure 3.3 Variation of *PSNR* with respect to the impulse-noise percentage for different values of the trimming range index *i*: (a) on the Lena image and (b) on the Mandrill image for fixed-valued impulse noise; (c) on the Lena image and (d) on the Mandrill image for random-valued impulse noise.

Experiment 3.6.3: Selection of the Thresholds

Having selected the trimming range, we have to select the most appropriate values of thresholds in the case of the RCT-I, RCT-II and the RCT-III detection schemes. Table 3.6 shows the dependence of the *PSNR* on the threshold on the absolute difference of the current pixel from the median in the case of the Lena and Mandrill images. It has been seen from this table that the most appropriate values of threshold is image specific. In case of fixed-valued impulse noise, it is 35 on the Lena image and 55 on the Mandrill image. In case of random-valued impulse noise, the corresponding values are 25 and 45 respectively. Experimental results on different images suggest that the appropriate threshold lies between 30 and 60 for fixed-valued impulse and between 15 and 50 for random-valued impulse. The exact value of the threshold can be decided on the basis of gray-scale variation in a particular image. It has been observed that higher thresholds are required for texture images and images that contain more high frequency components. For the RCT-II and RCT-III filters, different values of multiple thresholds were chosen according to the suggestions in the original papers by Chen and Wu [17] and Mitra *et al*. [3,15,16] respectively.

Table 3.6 Dependence of RCT-I filter on threshold

Lena								
θ	**Fixed-valued impulse noise**				**Random-valued impulse noise**			
	10%	**20%**	**30%**	**40%**	**10%**	**20%**	**30%**	**40%**
5	38.04	35.70	33.13	31.00	34.53	32.46	30.81	28.98
15	38.76	35.99	33.42	31.30	35.74	33.67	31.69	29.49
25	40.02	36.16	34.07	31.90	36.61	33.93	31.72	29.40
35	40.39	36.93	34.16	31.84	35.06	32.07	29.80	27.74
45	37.77	34.35	31.69	29.84	33.20	29.91	27.97	25.95
Mandrill								
35	26.60	25.48	24.21	22.81	24.65	23.31	21.93	20.50
45	27.38	25.78	24.44	22.75	25.11	23.50	22.07	20.77
55	28.10	25.91	24.28	22.56	25.55	23.29	21.70	19.70
65	28.18	25.69	23.61	21.96	25.60	22.94	21.14	19.48
75	27.34	24.54	22.33	20.64	25.13	22.26	20.39	18.76

Experiment 3.6.4: Comparative performance of the proposed filters

This set of experiments investigates the improvement of performance achieved by the proposed filters over existing variants of median filters. The following filters were used for comparison with the proposed RCT- I, RCT- II and RCT-III filters:

(1) Standard median (SM) filter

(2) Centre-weighted median (CWM) filter

(3) Rank-conditioned median (RCM) filter

(4) Tri-state median (TSM) filter

(5) Signal-dependent rank-ordered mean (SD-ROM) filter

(6) Chen and Wu's adaptive centre-weighted (Chen and Wu) median filter

In all cases, a window of 3×3 size was used. All the algorithms were implemented recursively: the estimate of the current pixel being dependent on the new values of the previously processed pixels in the filtering window.

The results on the Lena image are shown in Table 3.7, where the noise ratios for the two types of impulses range from 5% to 40%. Figures 3.4(a) and 3.4(b) plot the same results. Note that the performance of the CWM filter is very good at low noise ratios, but it degrades at higher noise ratios. The performance of the RCM filter is better than that of the SM and CWM filters at different noise ratios. The performance of the TSM filter is very good for both types of impulses with exception at higher noise ratios for fixed-valued impulse noise. The SD-ROM filter gives uniformly better performance over the SM, CWM, RCM, and TSM and Chen and Wu's adaptive median filters.

It has been seen from the tables and the figures that filtering based on the RCT-I, RCT-II and RCT-III filters provides superior results to other methods under consideration in removing both types of impulse noise at different noise ratios. Of the three, the RCT-III filter gives the best results. The improved performance is more prominent in the case of fixed-valued impulses. The performance of the RCT-I-based filter is at par with the SD-

ROM filter, while it gives better performance than the SM, CWM, RCM, TSM and Chen and Wu's adaptive centre-weighted median filters.

Table 3.7(a) Comparative performance of different filters in filtering the Lena image corrupted with 5% to 40% fixed-valued impulse noise

% Noise	Filters								
	SM	CWM	RCM	TSM	SD-ROM	Chen & Wu	RCT I	RCT II	RCT III
5	33.58	36.37	37.92	41.97	43.35	40.91	43.49	44.50	44.37
10	32.61	34.47	37.07	38.15	40.66	38.93	40.39	41.02	41.48
15	31.85	32.71	36.28	34.60	37.84	36.34	38.28	41.02	39.36
20	30.39	31.12	35.09	32.44	36.20	35.52	36.93	37.56	37.59
25	29.76	28.72	34.01	29.47	34.75	34.17	35.40	35.76	36.05
30	28.91	26.65	32.62	26.63	33.03	32.00	34.16	34.13	34.43
35	27.90	24.46	31.82	24.59	31.57	31.15	32.72	32.97	33.29
40	26.90	23.52	31.00	22.31	30.07	29.51	31.84	31.61	31.90

Table 3.7(b) Comparative performance of different filters in filtering the Lena image corrupted with 5% to 40% random-valued impulse noise

% Noise	Filters								
	SM	CWM	RCM	TSM	SD-ROM	Chen & Wu	RCT I	RCT II	RCT III
5	33.64	36.47	35.77	39.06	39.60	38.95	38.90	40.55	40.31
10	32.82	34.88	34.61	36.82	38.03	37.08	36.61	37.73	37.74
15	31.78	33.40	33.46	35.33	36.05	35.38	35.19	35.57	36.35
20	30.93	32.29	32.57	33.95	34.53	34.00	33.93	34.61	35.07
25	29.88	30.68	31.74	32.45	33.20	32.66	32.81	32.90	33.59
30	28.69	29.39	30.62	31.25	32.18	31.61	31.72	31.71	32.65
35	27.69	28.10	29.87	29.88	30.80	30.11	30.49	30.43	31.46
40	26.91	26.44	28.81	28.56	29.72	28.93	29.40	29.03	29.91

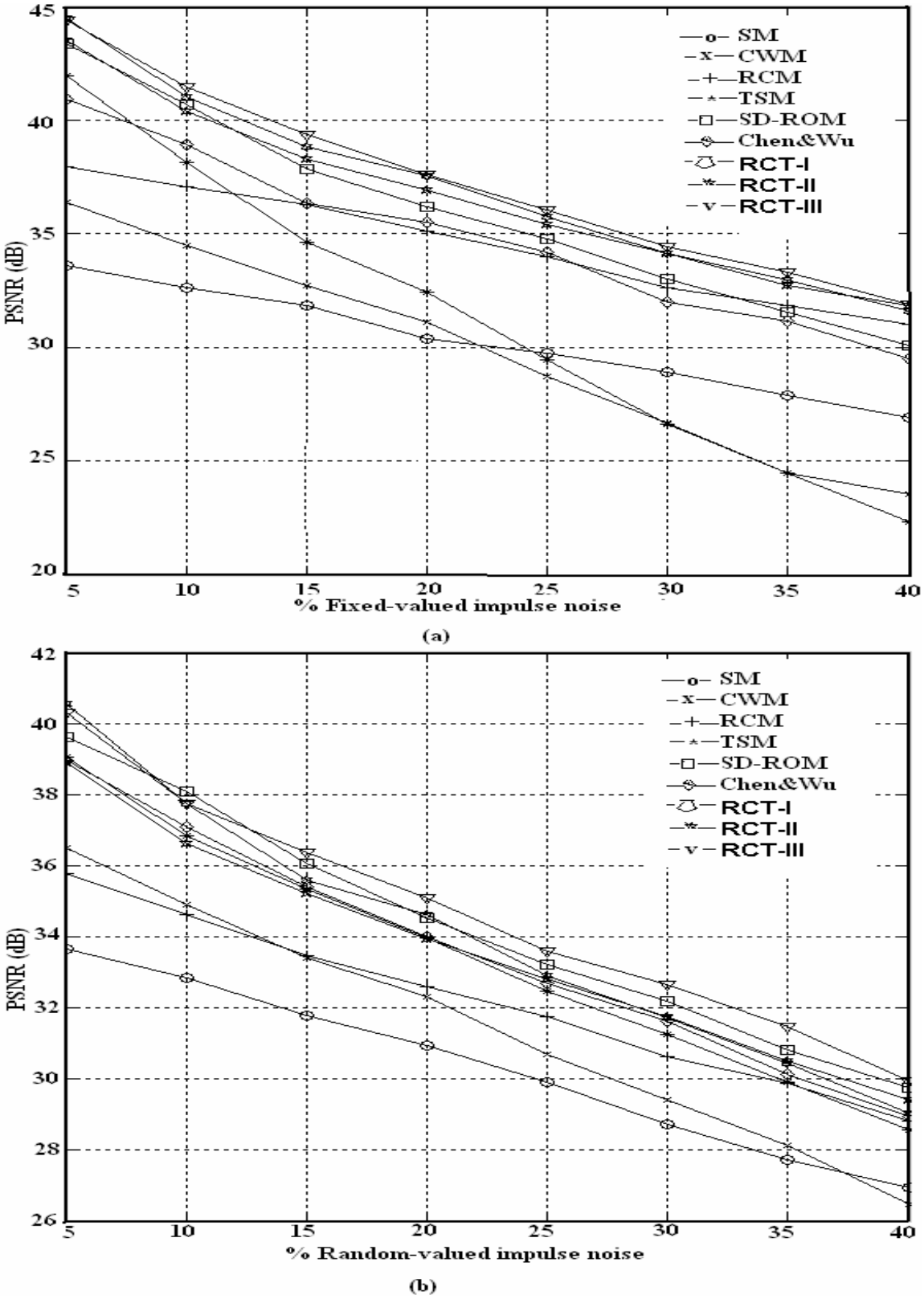


Figure 3.4 Noise ratio – *PSNR* plot for different filters on the Lena image: (a) fixed-valued impulse noise and (b) random-valued impulse noise.

Table 3.8(a) Comparative *PSNR* performance of filters in filtering different images corrupted with 20% fixed-valued impulse noise

Images	Filters								
	SM	**CWM**	**RCM**	**TSM**	**SD-ROM**	**Chen&Wu**	**RCT-I**	**RCT-II**	**RCT-III**
Lena	30.39	31.12	35.09	32.44	36.20	35.52	36.93	37.56	37.59
Mandrill	22.22	23.09	24.88	23.33	26.12	25.48	26.14	25.89	26.23
Airplane	26.86	29.46	33.11	30.88	31.99	31.99	33.37	33.78	33.83
Lake	26.65	27.91	31.05	29.48	30.40	30.38	31.75	32.22	32.32
Pepper	27.94	29.38	34.96	31.68	33.62	33.24	34.78	35.07	34.97
House	25.48	28.08	31.16	29.28	30.49	30.49	31.89	32.40	32.02
Bridge	25.51	25.49	27.60	26.63	27.65	27.54	28.47	29.00	29.15
Goldhill	27.94	30.83	32.64	30.86	33.26	32.90	33.85	34.25	34.09
Miramar	24.59	25.69	27.89	26.97	28.42	28.53	29.20	29.84	29.92
Point Loma	31.71	31.62	36.06	33.51	37.19	38.64	38.60	39.45	39.20
Golden	28.80	29.78	32.85	31.09	33.19	33.86	34.51	35.03	34.97
Terrain	22.87	22.97	24.71	23.71	24.29	24.75	25.59	25.95	26.10
Texmos1	17.63	17.62	19.73	17.46	16.37	17.10	19.76	19.62	19.64
Texmos3	17.71	17.85	19.66	17.52	16.58	15.41	19.76	19.72	19.63

Table 3.8(b) Comparative *PSNR* performance of filters in filtering different images corrupted with 20% random-valued impulse noise

Images	Filters								
	SM	**CWM**	**RCM**	**TSM**	**SD-ROM**	**Chen& Wu**	**RCT-I**	**RCT- II**	**RCT-III**
Lena	30.93	32.29	32.57	33.95	34.53	34.00	33.93	34.61	35.07
Mandrill	22.62	23.54	22.83	23.29	23.74	24.15	23.39	23.81	23.83
Airplane	27.51	30.33	30.77	31.80	30.51	31.16	31.45	32.01	31.85
Lake	27.20	28.77	28.73	29.93	28.63	29.52	29.83	30.25	30.03
Pepper	28.75	31.82	32.51	33.80	32.90	33.51	33.36	33.85	33.78
House	25.93	28.54	28.35	29.95	28.73	29.74	29.63	29.87	30.02
Bridge	23.94	24.02	23.48	24.50	24.00	24.47	23.86	24.11	24.61
Goldhill	28.93	30.11	30.34	31.99	31.30	31.85	29.83	32.37	32.15
Miramar	25.26	26.22	25.74	26.91	25.84	26.62	26.09	27.40	27.45
Point Loma	32.71	33.57	33.61	35.75	35.93	36.22	35.60	35.02	36.04
Golden	30.07	31.29	30.78	31.74	31.23	31.76	31.38	32.36	32.17
Terrain	23.13	23.30	22.58	23.55	22.39	23.18	22.62	24.13	24.31
Texmos1	17.65	17.67	17.18	16.43	15.38	16.34	17.15	17.47	17.70
Texmos3	17.03	17.12	16.63	16.37	15.12	15.84	16.60	17.08	17.24

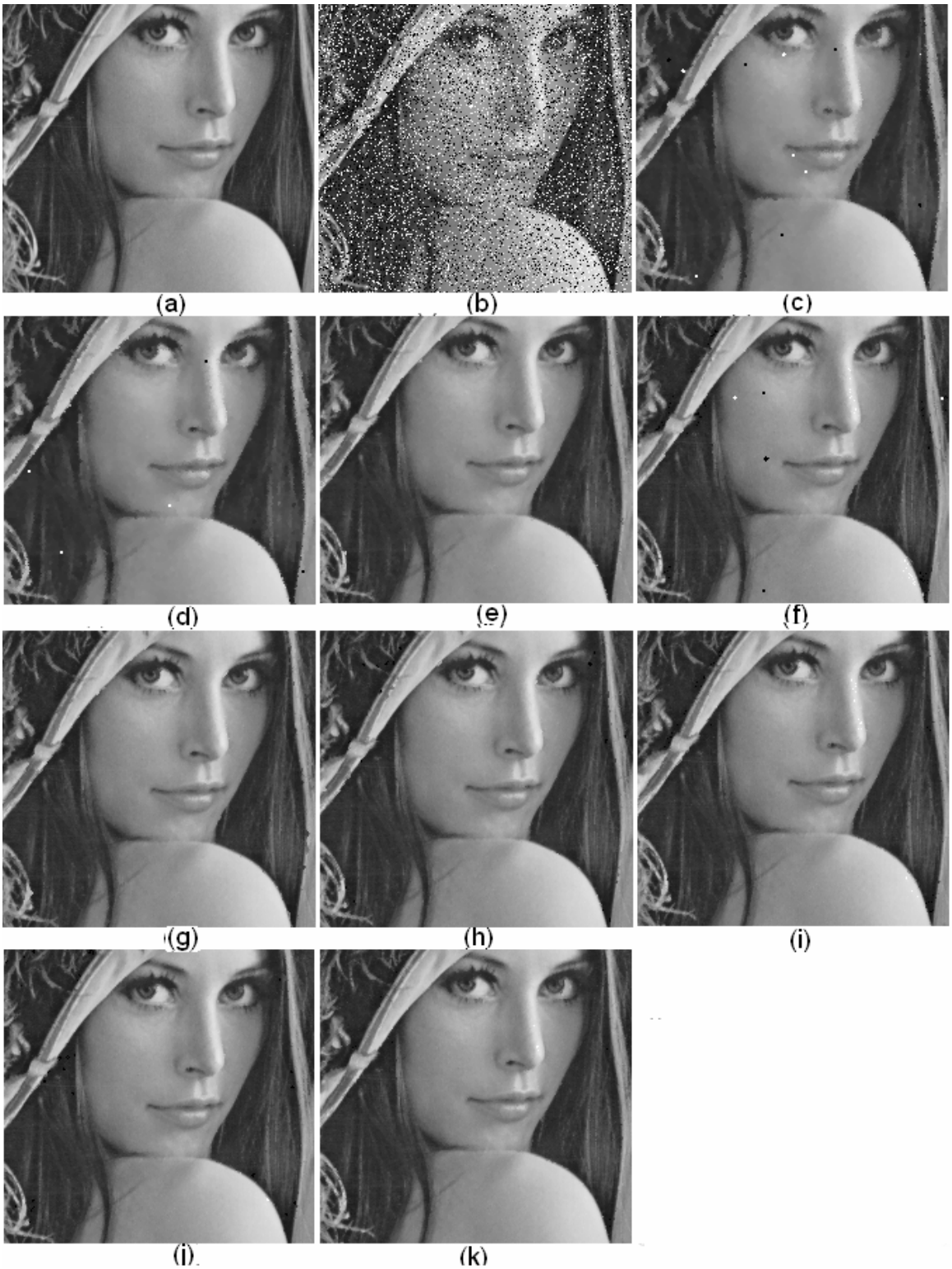


Figure 3.5 Outputs of various filters on the Lena image: (a) original image (part), (b) Image corrupted with 20% fixed-valued impulse noise, (c), (d), (e), (f), (g), (h), (i) (j) and (k) - outputs of the SM, CWM, RCM, TSM, SD-ROM, Chen and Wu, RCT-I, -II and -III filters.

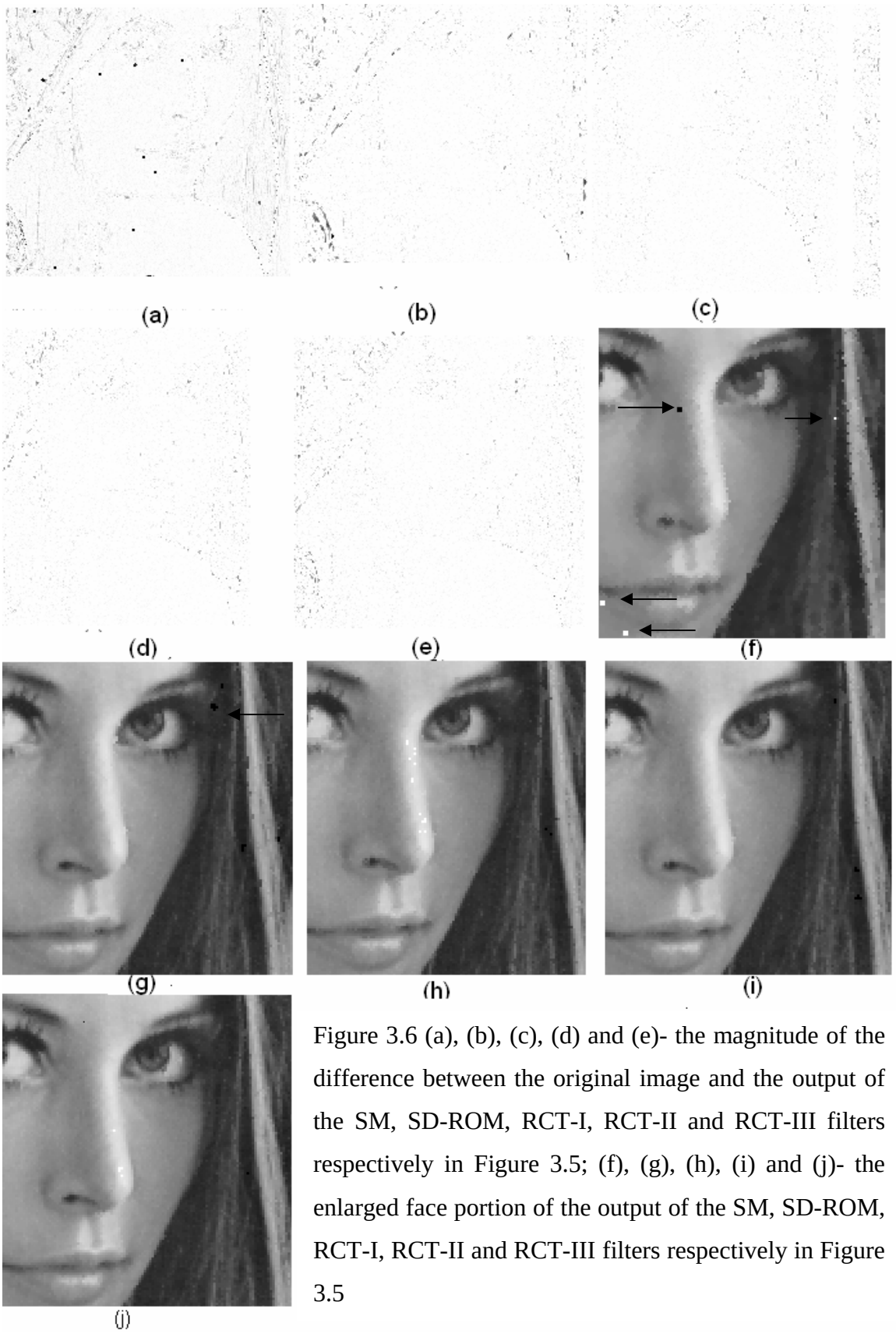


Figure 3.6 (a), (b), (c), (d) and (e)- the magnitude of the difference between the original image and the output of the SM, SD-ROM, RCT-I, RCT-II and RCT-III filters respectively in Figure 3.5; (f), (g), (h), (i) and (j)- the enlarged face portion of the output of the SM, SD-ROM, RCT-I, RCT-II and RCT-III filters respectively in Figure 3.5

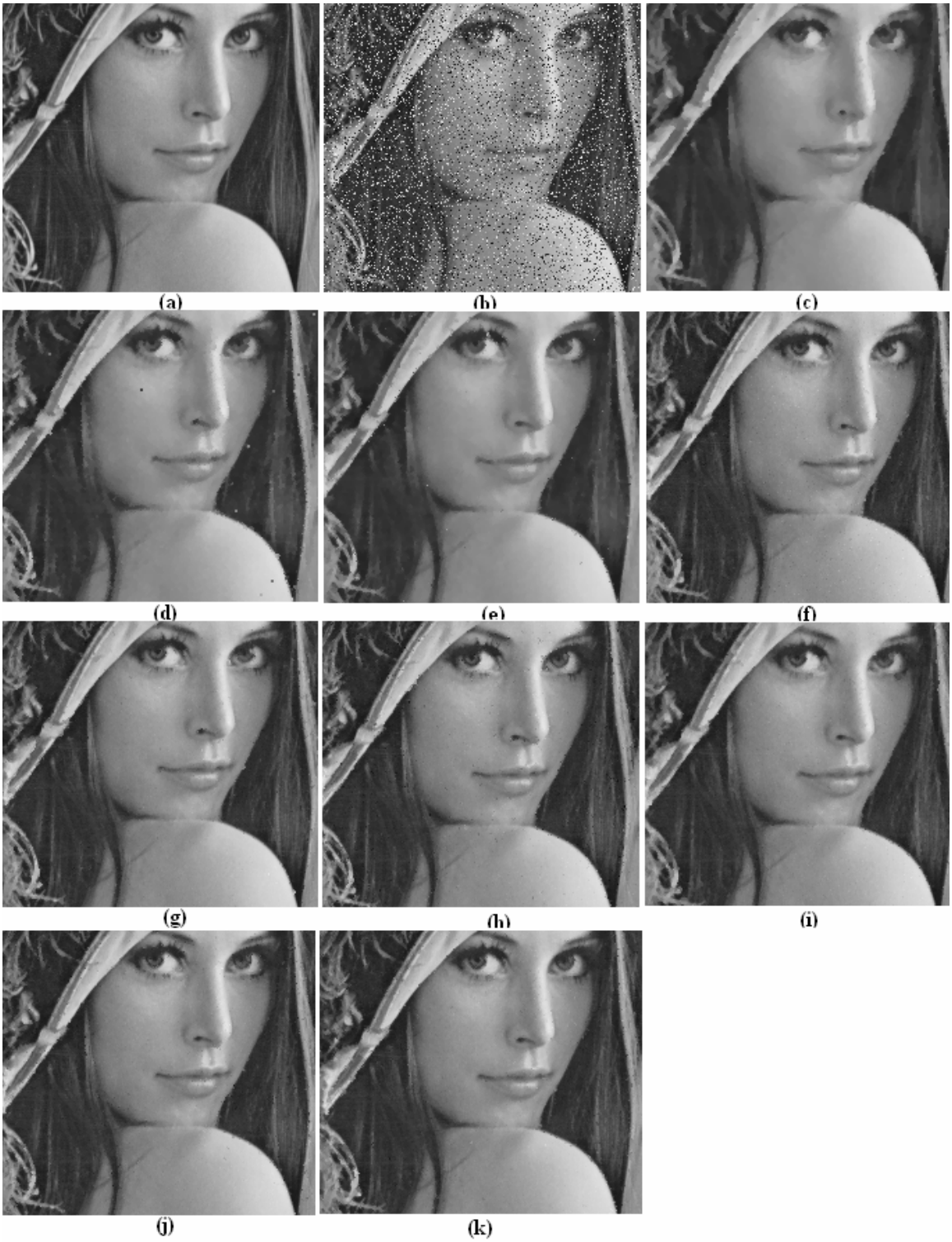


Figure 3.7 Outputs of various filters on the Lena image: (a) original image (part), (b) Image corrupted with 20% random-valued impulse noise, (c), (d), (e), (f), (g), (h), (i) (j) and (k) - outputs of the SM, CWM, RCM, TSM, SD-ROM, Chen and Wu, RCT-I, -II and -III filters.

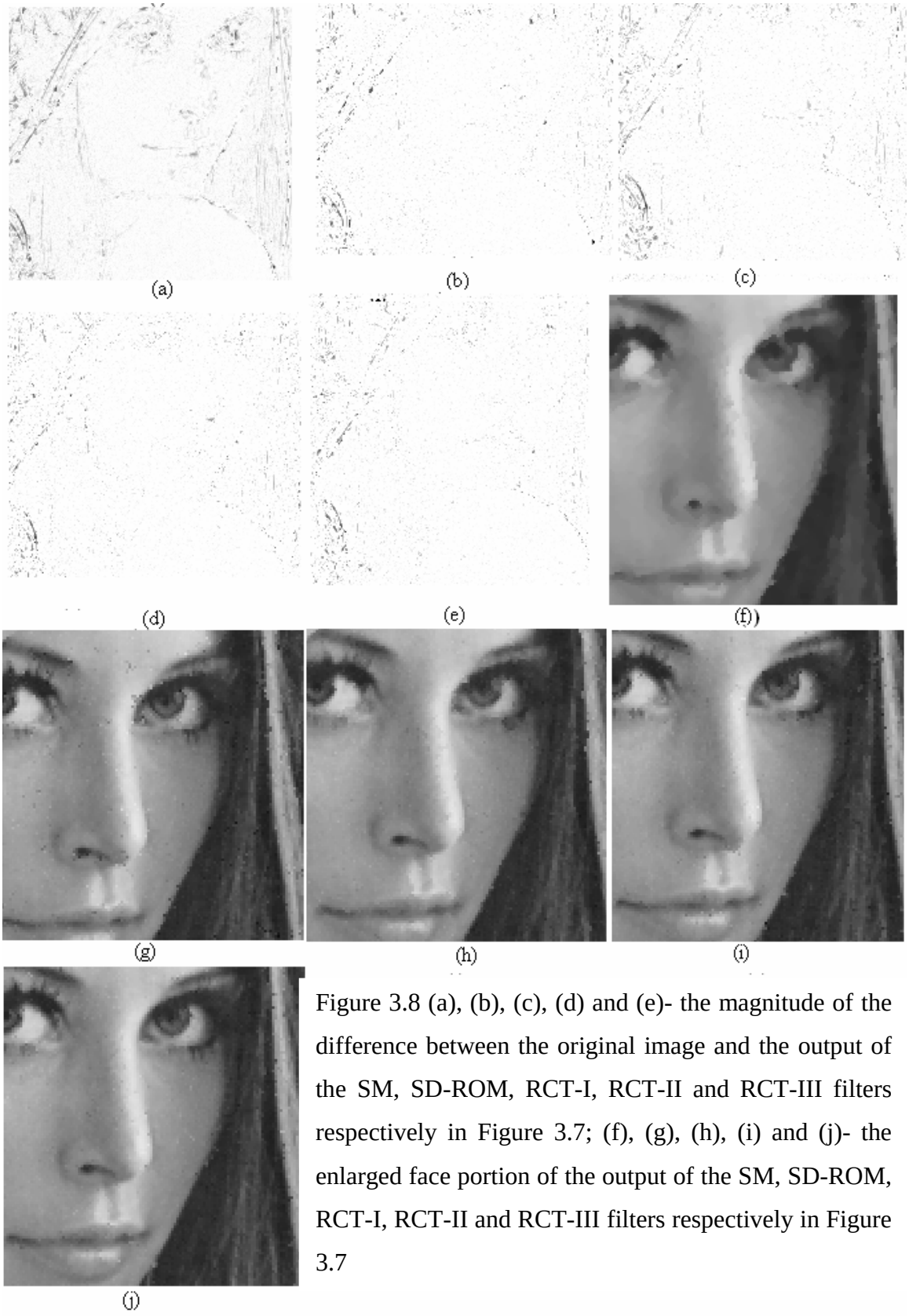


Figure 3.8 (a), (b), (c), (d) and (e)- the magnitude of the difference between the original image and the output of the SM, SD-ROM, RCT-I, RCT-II and RCT-III filters respectively in Figure 3.7; (f), (g), (h), (i) and (j)- the enlarged face portion of the output of the SM, SD-ROM, RCT-I, RCT-II and RCT-III filters respectively in Figure 3.7

Another set of experiments was carried out to study the comparative performance of the filters on different images. For this, each image was contaminated with 20% impulse noise ratio. Table 3.8 shows the comparative performance. The better performances of the proposed filters are observed from the table. Filters based on the RCT- II and RCT-III schemes achieved an improvement of over 1 dB of *PSNR* values over Chen and Wu's adaptive centre-weighted median filters and the SD-ROM filters.

Figures 3.5 and 3.7 show the Lena image restored by different filters for corruption with 20% of fixed-valued and random-valued impulse noise respectively. It has been seen from these figures that outputs of the SM filter are blurred. Outputs of the CWM filter contains many impulses left unfiltered, because the ability to remove impulses of this filter fails at higher noise ratios. It is clearly seen from Figures 3.5(d) and 3.7(d). Figure 3.5(f) shows that the TSM filter could not remove fixed-valued impulse noise. Performances of the SD-ROM and Chen and Wu's adaptive centre-weighted median filters are comparatively better than that of the SM, CWM, RCM and TSM filters, but edges could not be restored properly at certain regions. Besides, impulse remnants are seen in the outputs. Outputs of the proposed methods are shown in Figures 3.5 (i), 3.5 (j) and 3.5 (k) for fixed-valued impulse noise and in Figures 3.7(i), 3.7(j) and 3.7(k) for random-valued impulse noise. It has been seen that impulse remnant is very less. Blurring is less and edges are intact. The magnitudes of the differences of the original Lena image and the outputs of the SM, SD-ROM and proposed filters, for fixed- and random-valued impulse noise are shown in Figures 3.6(a), 3.6(b), 3.6(c), 3.6(d) and 3.6(e) and Figures 3.8(a), 3.8(b), 3.8(c), 3.8(d) and 3.8(e) respectively. The differences have been magnified two times for clarity. It has been seen from these differences that impulse remnants in case of proposed filters are less. Remnants are clearly visible in the case of the SM filter. The enlarged face parts of the outputs of the SM and the SD-ROM, the RCT-I, the RCT-II and the RCT-III filters for fixed- and random-valued impulse noise are shown in Figures 3.6(f), 3.6(g), 3.6(h), 3.6(i) and 3.6(j) and Figures 3.8(f), 3.8(g), 3.8(h), 3.8(i) and 3.8(j) respectively. These images show better visual performance of the proposed filters. Notice the remnant noise in SM filter output and the edge distortion in the SD-ROM filter output as indicated by arrows in Figure 3.6.

3.7 Conclusions

This chapter described the scalar rank-conditioned and threshold median filters for removal of fixed-valued and random-valued impulse noise from gray-scale images. The filters are based on the detection of impulse noise by simultaneous application of rank-conditioning and one of the following three thresholding conditions:

(1) Thresholding the absolute difference of the centre pixel from the median (RCT-I),

(2) Adaptive thresholding of the absolute differences of the centre pixel from the CWM filter outputs for multiple weights (RCT-II) and

(3) Thresholding of the absolute differences of the centre pixel from other rank-ordered pixels (RCT–III)

Simultaneous rank-conditioning and thresholding enhanced impulse detection as evidenced by the experimental results. The corrupted pixels were replaced by the rank-ordered mean (ROM) of the pixels in the window excluding the centre pixel. The performance of the proposed filters was compared with that of the SM, CWM, RCM, TSM, SD-ROM and Chen and Wu's adaptive median filters. Experimental results show that the RCT-II and the RCT-III filters outperformed the SM, CWM, RCM, TSM, SD-ROM and Chen and Wu's adaptive centre-weighted median filters in removal of fixed-valued and random-valued impulse noise at various impulse noise ratios from different types of images in the terms of *PSNR* values and better visual quality. The performance of the RCT-I filter, which is much simpler, is at per with the SD-ROM filter. The better performance of the RCT filters is due to the integration of the RCM filter with the threshold mechanism. The computational cost of the RCT-I filter is relatively low and that of the RCT-II filter is high. The proposed filters gave stable performance on different types of images.

The RCT filters promise to be better alternatives to the existing variants of median filters. The next chapter will explore the extension of the concepts developed in this chapter to colour images.

Chapter 4

Rank-conditioned and Threshold Vector Median Filters for Colour Images

4.1 Introduction

As discussed in the earlier chapters, the median filter is often applied to filter impulse noise from gray-scale images for its ability to preserve edges and fine image details besides removing noise. The extension of the concept of the scalar median filter and its modifications to colour image processing is not straightforward as the vector signal has strong inter-component correlation. The natural tendency will be to implement the scalar median filter separately on each channel of a multichannel image. Colour and edge distortions [35] may occur as a result of such independent filtering of separate channels. The better multichannel filters are based on the vector approach where simultaneous processing of all the components of the vector signal is done. In this chapter, the thesis introduces various vector median filters available in the literature and proposes two algorithms for removing impulse noise from the colour images.

4.2 Ordering Multichannel Data

The problem of ordering multichannel data should be addressed first before an order-statistics-based method is devised. An order is to be defined for ranking the sequence of colour vectors. Such ordering should be able to distinguish when a vector is either *larger* or *smaller* than another vector in some meaningful sense. There is no universally acceptable way to order vectors because of the difficulty in defining the natural concept of the rank in the case of vectors. Ordering methods involve the similarity between two vectors measured in terms of the distance measures and explained below. The existing ordering methods [29,31] are then are outlined.

4.2.1 Distance Measures

The generalized *distance* between two vector signals is the Minkowski metric (L_M-metric) [35]. Special cases of this metric are derived from it. For two C-dimensional vectors $\boldsymbol{x}_i = (x_i(1), x_i(2), ..., x_i(C))$ and $\boldsymbol{x}_j = (x_j(1), x_j(2), ..., x_j(C))$, the Minkowski metric $d_M(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is given by

$$d_M(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_{k=1}^{C} |\, \boldsymbol{x}_i(k) - \boldsymbol{x}_j(k) \,|^M\right)^{\frac{1}{M}} \tag{4.1}$$

Three special cases of the Minkowski metric are as follows:

City-block Distance (L_1-metric) **)**

$$d_1(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{k=1}^{C} |\, \boldsymbol{x}_i(k) - \boldsymbol{x}_j(k) \,| \tag{4.2}$$

Euclidean Distance (L_2-metric)

$$d_2(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\sum_{k=1}^{C} (\boldsymbol{x}_i(k) - \boldsymbol{x}_j(k))^2)^{\frac{1}{2}} \tag{4.3}$$

Chess-board distance (L_∞-metric)

$$d_\infty(\boldsymbol{x}_i, \boldsymbol{x}_j) = \max_k |\, \boldsymbol{x}_i(k) - \boldsymbol{x}_j(k) \,| \tag{4.4}$$

Other distance measures, like the Mahalanobis distance, the Canberra distance and the Czekanowski distance, which are distinct from the Minkowski metric, can also be used [35].

4.3 Sub-ordering Principles

Since there is no natural ordering of vector data, several restricted or *sub-ordering principles* are used in the literature [1,35,66]. Following four sub-ordering principles are suggested:

1. Marginal ordering or *M*-ordering
2. Conditional ordering or *C*-ordering
3. Partial ordering or *P*-ordering
4. Reduced ordering or *R*-ordering

Marginal ordering (M-ordering), is a scalar ordering [67] in which each channel of the multichannel signal is ordered independently and no correspondence among components is checked. The median filter based on this type of ordering is known as the *marginal median filter* (MMF). It is robust in the sense that it discards impulse noise while preserving important image features such as edges and details [67]. The main disadvantage of the marginal median filter is the resulting colour distortion and imbalance as there is no one-to-one correspondence between the original vector samples

and the samples obtained after ranking. The filter outputs can combine to produce vector pixels not present in the original image.

In *conditional ordering (C-ordering),,* multichannel ordering is reduced to a scalar one based on the marginal ordering of a single channel. Each multichannel sample is given a rank according the rank of the component for which ordering is done. Thus conditional ordering does not consider the vector nature of data.

In *partial ordering (P-ordering),* subsets of vector data are grouped together forming the minimum convex hulls. The first convex hull is formed such that the perimeter contains a minimum number of points and the resulting hull contains all other points in the given set. These points on the perimeter have the extreme rank. The perimeter points are discarded, the new perimeter points are formed and the process repeats. The problems with *P*-ordering are the multiplicity of data for a given rank and the complexity of convex hull determination.

In *reduced ordering (R-ordering),* each multichannel vector is reduced to single scalar by means of some combination of the components of the vector. As a consequence, multichannel ordering is reduced to scalar ordering. Ranking may be in terms of the *sum of distances* (*SOD*) from a fixed point and such ranking schemes [1,35,66] include *R*-ordering about the mean, *R*-ordering about the marginal median, *R*-ordering about the centre vector pixel etc. Reduced ordering is easy to implement and can provide clues about outlying observations. It is the sub-ordering principle that is the most natural for vector-valued observations like a colour image. The choice of the appropriate reference vector is crucial for the reduced ordering scheme.

4.4 Vector Median Filters

The basic vector median filter (VMF) [30] is based on a variation of the reduced ordering. As stated earlier, the principle of the reduced ordering is to map each vector into a scalar, rank the scalars and define the rank of the vector according to the rank of the corresponding scalar. The scalar is derived by a distance-based approach. The *SOD* of a vector from other vectors is used as the scalar for reduced-ordering. The data are ranked

from the vector with the least *SOD* from other vectors to the vector with the largest *SOD* [30,31,39,40]. Two popular distance measures that are frequently used in the VMF are the City-block distance and Euclidean distance.

Consider a C-channel vector pixel $\boldsymbol{x}$ at the centre of a square window W of size $n \times n$ containing $N = n^2$ vector pixels. Assume n to be odd. The vector pixels in the window are denoted by $\boldsymbol{x}_i, i = 1, 2..., N$, where $\boldsymbol{x}_1$ is the top left-most vector pixel and $\boldsymbol{x}_N$ is the bottom right-most vector pixel in the window. The remaining vector pixels are scanned from left to right and top to bottom with the centre vector pixel $\boldsymbol{x} = \boldsymbol{x}_{\frac{N+1}{2}}$. The set of all vector pixels inside the window is given by

$$\boldsymbol{X}_{\mathrm{W}} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\} \tag{4.5}$$

The vector median filter [30] is computed as follows:

(1) Find the sum of the distances δ_i of the ith $(1 \le i \le N)$ vector pixel from all other neighbouring vector pixels in the window given by

$$\delta_i = \sum_{j=1}^{N} d(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{4.6}$$

where $d(\boldsymbol{x_i}, \boldsymbol{x_j})$ represents an appropriate distance measure between the ith and jth neighbouring vector pixels.

(2) Arrange δ_is in the ascending order. Assign the vector pixel $\boldsymbol{x}_i$ a rank equal to that of δ_i. Thus, an ordering

$$\delta_{(1)} \le \delta_{(2)} \le .... \le \delta_{(N)} \tag{4.7}$$

implies the same ordering of the corresponding vectors given as

$$\boldsymbol{x}_{(1)} \le \boldsymbol{x}_{(2)} \le .... \le \boldsymbol{x}_{(N)} \tag{4.8}$$

where $\boldsymbol{x}_{(1)}, \boldsymbol{x}_{(2)}, ..., \boldsymbol{x}_{(N)}$ are the rank-ordered vector pixels with the number inside the parentheses denoting the corresponding rank. The set of rank-ordered vector pixels is given by

$$\boldsymbol{X}_{\mathrm{R}} = \{\boldsymbol{x}_{(1)}, \boldsymbol{x}_{(2)}, ..., \boldsymbol{x}_{(N)}\} \tag{4.9}$$

(3) Take $\boldsymbol{x}_{(1)}$ as the vector median. Thus the vector median is given by

$$\boldsymbol{x}_{\mathrm{VMF}} = \boldsymbol{x}_{(1)} \tag{4.10}$$

The vector median is defined as the vector that corresponds to the *minimum SOD* to all other vector pixels. This selection is due to the fact that a data vector differing from the data population usually appears in higher indexed locations in the ordered sequence. $\boldsymbol{x}_{(N)}$ is the most outlying vector pixel, because it has the largest *SOD* to all the other vector pixels in the window. The *healthiest vector pixel* in the window is the vector median.

The VMF processes all components jointly so that the cross-correlation between components is exploited. The vector median is the maximum likelihood estimate (*MLE*) of the location parameter of the vector data in a window when the underlying distribution is multivariate Laplace [58]. Thus, given $\boldsymbol{X}_{\mathrm{W}} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ of independent samples and the joint probability density $f(\boldsymbol{x_i}) = \frac{1}{\lambda} e^{-\lambda d(\boldsymbol{x}_i, \boldsymbol{\beta})}$, the *MLE* of the location parameter $\boldsymbol{\beta}$, denoted by $\hat{\beta}$, minimizes $\sum_{i=1}^{N} d(\boldsymbol{\beta}, \boldsymbol{x}_i)$.

The vector median filter is computationally very expensive. For an $N = n^2$ element processing window, the VMF_1 based on the City-block distance requires $\mathrm{O}(n^3)$ additions and comparisons per vector pixels in a running algorithm-based implementation. The

computational complexity and *PSNR*-performance of the filter leaves room for improvement in the filtering technique. Improved vector median filtering has been an area of research during the last two decades.

4.4.1 Basic Vector Directional Filters

A vector pixel is characterised by both its magnitude and direction. The angle between two vector pixels measures their similarity in terms of chromaticity. The vector pixel, which is different from other vector pixels in the window in terms of chromaticity, may be considered as an outlier. Thus the sum of the angles between each vector and other vectors in the window, rather than the sum of the distances, may be considered for ranking. The basic vector directional filter (BVDF) [36] works on this principle. The BVDF outputs $\boldsymbol{x}_k \in \boldsymbol{X}_{\mathrm{W}}$ such that

$$\sum_{j=1}^{N} \phi(\boldsymbol{x}_k, \boldsymbol{x}_j) < \sum_{j=1}^{N} \phi(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad \forall \boldsymbol{x}_i \neq \boldsymbol{x}_k \in \boldsymbol{X}_{\mathrm{W}} \tag{4.11}$$

where $\phi(\boldsymbol{x}_i, \boldsymbol{x}_j) = \cos^{-1} \dfrac{\boldsymbol{x}_i \bullet \boldsymbol{x}_j}{\| \boldsymbol{x}_i \| \ \| \boldsymbol{x}_j \|}$, $\bullet$ denotes the scalar product and $\| \ \|$ is the $L_2 - \text{norm}$.

The BVDF can reduce chromaticity errors and its output is the *MLE* of the location parameter of the underlying probability density function of the directions of the input vectors in the window [35].

The *directional distance filter* (DDF) [37] combines the properties of VMF and BVDF. It utilizes both the sum of the distances between vectors and the sum of the angles between vectors for ordering. The DDF outputs $x_k \in \boldsymbol{X}_{\mathrm{W}}$ such that

$$\sum_{j=1}^{N} d(\boldsymbol{x}_k, \boldsymbol{x}_j) \sum_{j=1}^{N} \phi(\boldsymbol{x}_k, \boldsymbol{x}_j) < \sum_{j=1}^{N} d(\boldsymbol{x}_i, \boldsymbol{x}_j) \sum_{j=1}^{N} \phi(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad \forall \boldsymbol{x}_i \neq \boldsymbol{x}_k \in \boldsymbol{X}_{\mathrm{W}} \tag{4.12}$$

The DDF reduces intensity outliers while preserving the chromaticity of the vector pixel.

4.5 Improved Vector Median Filters

Like in the case of scalar median filter, the only tunable parameter of the vector median filters is its sliding window shape or size and it may not be possible to tune its property for a given application. More flexible filters such as the weighted vector median filter (WVMF) [38] and the vector signal-dependent rank-ordered mean (VSD-ROM) filter [33] significantly improve the filter performance. We will be considering a special case of the weighted vector median filter, namely the Centre weighted vector median filter (CWVMF) and VSD-ROM filter as references for comparing the performance of the proposed vector filters.

4.5.1 Centre-weighted Vector Median Filter

The weighted vector median filter minimizes the sum of weighted distances $\delta_i^w = \sum_{j=1}^{N} w_j d(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where w_j is the non-negative integer weight associated with the vector pixel $\boldsymbol{x}_j$. The centre-weighted vector median filter gives a greater emphasis on the centre vector pixel by assigning a weight greater than unity and unity weight to each of the remaining vector pixels. This is equivalent to replicating the centre vector pixel an integer number of times. The output is, then selected as the vector median from the extended set of vector pixels. The extended set of rank-ordered vector pixels is given by

$$\boldsymbol{X}_{\mathrm{R}_1} = \{\boldsymbol{x}_{(1)}, \boldsymbol{x}_{(2)}, ..., \boldsymbol{x}_{(N_1)}\} \tag{4.13}$$

where $\boldsymbol{x}_{(1)} \le \boldsymbol{x}_{(2)} \le .... \le \boldsymbol{x}_{(N_1)}$, $N_1 = N + w - 1$ and w is the weight of the centre vector pixel. $\boldsymbol{x}_{(1)}$ is the centre-weighted vector median and it replaces the centre vector pixel.

4.5.2 Vector Signal-dependent Rank-ordered Filter

The vector signal-dependent rank-ordered mean (VSD-ROM) filter is based on a state-estimation approach, where the state of each vector pixel is estimated and an appropriate

replacement is determined, if required [33]. It sorts all vector pixels excluding the centre one in the window in the ascending order of magnitudes of the sums of distances to all other vector pixels, that is, $\boldsymbol{x}_{(1)} \le \boldsymbol{x}_{(2)} \le .... \le \boldsymbol{x}_{(N-1)}$. The vector pixel $\boldsymbol{x}_{(1)}$ with the smallest *SOD* is the vector median. The VSD-ROM filter, then compares the distances between the centre vector pixel and four smallest of the sorted vector pixels to a set of four increasing thresholds $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ such that $\theta_1 < \theta_2 < \theta_3 < \theta_4$. If any of the thresholds is exceeded, the centre vector pixel is decided as an impulse and replaced by the vector median. The vector signal-dependent rank-ordered mean filter is given by [33]

$$\boldsymbol{x}_{\text{VSD-ROM}} = \begin{cases} \boldsymbol{x}_{(1)} & \text{if } |\boldsymbol{x}_{(i)} - \boldsymbol{x}| > \theta_i \text{ for any } i = 1,2,3,4. \\ \boldsymbol{x} & \text{otherwise} \end{cases} \tag{4.14}$$

4.6 Proposed Rank-conditioned Vector Median Filter

One of the main drawbacks of the VMF is that the centre vector pixel in the window is always replaced by the vector median irrespective of whether it is corrupted with impulse noise or not. There are many vector pixels in the window and one of the vectors is the vector median. The application of the VMF on the image results in blurred and distorted outputs if the centre vector pixel is healthy and it is replaced by a new value. Example 4.1 explains it.

Example 4.1: Let $\boldsymbol{X}_{\text{W}} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_9\}$ be the set of vector pixels in a 3×3 window, where $\boldsymbol{x}_1 = (100,50,10)^{'}$, $\boldsymbol{x}_2 = (200,150,100)$, $\boldsymbol{x}_3 = (160,110,50)$, $\boldsymbol{x}_4 = (100,50,10)$, $\boldsymbol{x}_5 = (200,150,100)$, $\boldsymbol{x}_6 = (160,110,50)$, $\boldsymbol{x}_7 = (100,50,10)$, $\boldsymbol{x}_8 = (200,150,100)$ and $\boldsymbol{x}_9 = (160,110,50)$, and $\boldsymbol{x}_5$ is the centre vector pixel. The corresponding sums of distances are 1350, 1260, 870, 1350, 1260, 870, 1350, 1260 and 870. Then, the set of the rank-ordered vector pixels is given by $\boldsymbol{x}_{\text{R}} = \{\boldsymbol{x}_{(1)}, \boldsymbol{x}_{(2)}, ..., \boldsymbol{x}_{(9)}\}$ $= \{\boldsymbol{x}_3, \boldsymbol{x}_6, \boldsymbol{x}_9, \boldsymbol{x}_2, \boldsymbol{x}_5, \boldsymbol{x}_8, \boldsymbol{x}_1, \boldsymbol{x}_4, \boldsymbol{x}_7\}$ and the vector median $\boldsymbol{x}_{\text{VMF}} = \boldsymbol{x}_{(1)}$ is

$\boldsymbol{x}_3 = (160,110,50)$. The VMF_1 will replace the centre vector pixel $\boldsymbol{x}_5$ by $\boldsymbol{x}_3$. However, the centre vector pixel $\boldsymbol{x}_5$ is not a corrupted vector pixel, but a vector pixel on a thin line.

To avoid such an undesirable effect in gray-scale images, the rank-conditioned median filter [12,13] is used. It replaces the centre pixel by the median when the rank of the centre pixel does not belong to the trimming range [14]. We extend it into the rank-conditioned vector median filter (RCVMF) for multichannel images. The rank-conditioned vector median filter outputs the vector median when the rank of the centre vector pixel is bigger than a pre-defined rank of a healthy vector pixel inside the window. The rank-ordering given by Equation 4.9 is considered. We are motivated by the following two considerations.

(1) Those vector pixels that appear towards the extreme right in rank ordering have comparatively larger sums of distances. This means that these vector pixels may be different from other vector pixels in the window and may represent impulse noise.

(2) Those vector pixels that appear towards the extreme left in rank ordering have smaller sums of distances. They may be considered as healthy vector pixels.

On the basis of the above two considerations, a trimming range [14] of the healthy vector pixels may be defined by

$$\{i \mid 1 \le i \le N_T\} \tag{4.15}$$

where $N_T < N$.

The r*ank-conditioned vector median filter* (RCVMF) is now defined as

$$\boldsymbol{x}_{\text{RCVMF}} = \begin{cases} \boldsymbol{x}_{\text{VMF}} & \text{if } \text{rank}(\boldsymbol{x}) > N_T \\ \boldsymbol{x} & \text{otherwise} \end{cases} \tag{4.16}$$

The proposed rank-conditioned vector median filter is computed in the following steps:

(1) Find the sum of the distances δ_i of the ith $(1 \le i \le N)$ vector pixel from all other neighbouring vector pixels in the window.

(2) Arrange δ_is in the ascending order. Assign the vector pixel $\boldsymbol{x}_i$ a rank equal to that of δ_i.

(3) Form a trimming range according to Equation 4.15. Replace $\boldsymbol{x}$ by $\boldsymbol{x}_{\text{RCVMF}}$ according to the rule in Equation 4.16.

4.6.1 Selection of N_T

Selection of an appropriate value for N_T is very important to decide the compromise between the detail preservation in the vector median output and the removal of impulses from the corrupted images. Image details are preserved better with a larger value of N_T at the cost of not removing more impulses in the vector median output. On the other hand, impulses as well as image features are removed with a smaller value of N_T, resulting in blurring and distortion of image details in the vector median output.

In the above Example 4.1, let $N_T = 5$. The rank of the centre vector pixel is 5. The condition for filtering in Equation 4.16 of the proposed rank-conditioned vector median filter does not hold and hence, the centre vector pixel will not be replaced by a new value.

4.7 Proposed Rank-conditioned and Threshold Vector Median Filter

In the case of the scalar images, filtering performance was significantly improved by incorporating the *threshold mechanism* along with rank-conditioning for the detection of impulses. The filter structure in Equation 4.16 can be further enhanced by incorporating the *threshold mechanism* for the detection of impulses. Vector pixels in the window are arranged from the least rank to the highest rank. The centre vector pixel is considered healthy if its rank is not bigger than the pre-defined rank N_T. On the other hand, it may not be always true to consider the centre vector pixel as corrupted even if its rank is bigger than the pre-defined rank N_T. The vector pixel with an extreme rank may not be

sufficiently different from those vector pixels within the trimming range to qualify as impulse noise. Additional conditions are required to test whether the centre vector pixel is not a healthy pixel. *The corrupted pixel is dissimilar from a healthy pixel.* This means that the distance between the corrupted pixel and a healthy vector pixel is sufficiently large. This condition can be imposed by selecting a healthy vector pixel inside the trimming range and thresholding the distance of centre vector pixel from it.

Consider a *healthy vector pixel* $\boldsymbol{x}_{(k)}$ where $1 \le k \le N_T < N$. Suppose D is the distance of the centre vector pixel $\boldsymbol{x}$ from $\boldsymbol{x}_{(k)}$ and given by

$$D = d(\boldsymbol{x}, \boldsymbol{x}_{(k)}) \tag{4.17}$$

The vector pixel $\boldsymbol{x}$ is likely to be a corrupted vector pixel if its rank is bigger than the pre-defined rank N_T and its distance D from the *healthy vector pixel* is bigger than a pre-determined threshold θ. On the basis of the formulation, the proposed *rank-conditioned and threshold vector median filter (RCTVMF)* has the following form:

$$\boldsymbol{x}_{\text{RCTVMF}} = \begin{cases} \boldsymbol{x}_{\text{VMF}} & \text{if rank}(\boldsymbol{x}) > N_T \text{ and } D > \theta \\ \boldsymbol{x} & \text{otherwise} \end{cases} \tag{4.18}$$

The healthiest vector pixel is the vector median $\boldsymbol{x}_{(1)}$ and is considered for computing D. Incorporation of the threshold mechanism in the proposed filter benefits to differentiate between impulses and image details like edges, thin lines, ends of lines etc. The distance D is normally very big if an impulse is present at the centre of the window. N_T and θ ensure that an impulse is much different from other vector pixels in the neighbourhood. More examples are given below and the L_1-metric is used for measurement of D in these examples.

Example 4.2: Let $\boldsymbol{X}_{\text{W}} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_9\}$ be the set of vector pixels in a 3×3 window, where $\boldsymbol{x}_1 = (175, 50, 100)$, $\boldsymbol{x}_2 = (175, 50, 100)$, $\boldsymbol{x}_3 = (175, 50, 100)$, $\boldsymbol{x}_4 = (180, 50, 100)$, $\boldsymbol{x}_5 = (180, 255, 100)$, $\boldsymbol{x}_6 = (180, 50, 105)$, $\boldsymbol{x}_7 = (185, 50, 105)$, $\boldsymbol{x}_8 = (185, 55, 105)$ and

$\boldsymbol{x}_9 = (185, 55, 105)$. One component of the centre vector pixel is corrupted by impulse noise. The corresponding sums of distances are 280, 280, 280, 265, 1680, 270, 285, 300 and 300. The vector median is $\boldsymbol{x}_4 = (180, 50, 100)$ and considered as the healthy vector pixel for computing D. The rank of the centre vector pixel is 9. Let $N_T = 5$ and $\theta = 40$. As per the RCVMF, the centre vector pixel is corrupted, because its rank is bigger than 5. The distance D between the two vector pixels is 210. The centre vector pixel is declared corrupted by the proposed rank-conditioned and threshold vector median filter, because its rank is bigger than 5 and D is bigger than 40. Thus the VMF_1, RCVMF and RCTVMF replace the centre vector pixel by the vector median.

Example 4.3: Let $\boldsymbol{x}_1 = (181, 71, 81)$, $\boldsymbol{x}_2 = (181, 71, 81)$, $\boldsymbol{x}_3 = (181, 71, 81)$, $\boldsymbol{x}_4 = (181, 74, 81)$, $\boldsymbol{x}_5 = (188, 74, 81)$, $\boldsymbol{x}_6 = (185, 74, 81)$, $\boldsymbol{x}_7 = (185, 74, 81)$, $\boldsymbol{x}_8 = (185, 74, 81)$ and $\boldsymbol{x}_9 = (185, 74, 81)$ be the vector pixels. The centre vector pixel is healthy. The corresponding sums of distances are 41, 41, 41, 32, 49, 28, 28, 28 and 28. The vector median is $\boldsymbol{x}_6 = (185, 74, 81)$ and considered to be the healthy vector pixel. The rank of the centre vector pixel is 9. Let $N_T = 5$ be the rank of a healthy vector pixel that corresponds to $\boldsymbol{x}_3 = (181, 71, 81)$. As per the RCVMF, the centre vector pixel is corrupted, because its rank is bigger than 5. The distance D between the two vector pixels is 3. Let $\theta = 40$ be the threshold. The centre vector pixel is declared healthy by the proposed RCTVMF, though its rank is bigger than 5, because D is less than 40. On the other hand the VMF_1 and RCVMF replace the centre vector pixel by the vector median.

Like in the scalar case, simultaneous application of rank-conditioning and thresholding reduces the probability of detecting a healthy vector pixel as an impulse. However the rate of true hit is not affected, because an impulse simultaneously satisfies the conditions for rank-conditioning and thresholding.

4.8 Computational Complexity Analysis

An analysis of the computational complexities of the proposed RCVMF and RCTVMF is carried out. The computational complexities of the VMF_1 and the VSD-ROM filter are

included for comparison [68-70]. In the following, the number of channels and the size of the filtering window are assumed to be C and $N = n^2$ respectively and the L_1-metric is used for distance measurement. This analysis does not consider optimal implementation of the filters and is only indicative of comparative computational overheads.

Note that the computation of the vector median filter (VMF_1) requires $(2C-1)(n(n^2-n)+n(n-1)/2)+2n(n^2-n)+n^3-n$ additions/subtractions and $C(n(n^2-n)+n(n-1)/2))+n^2-1$ comparisons including n^2-1 comparisons/swappings to search the vector median. These numbers are based on a faster implementation, which calculates the distances relative only to those vector pixels that enter newly to the filtering window corresponding to the new location of the vector pixel [70]. The computation of the rank-conditioned vector median filter (RCVMF) involves the calculation of the same number of distances as the VMF_1. It involves additional n^2 comparisons/swappings to check whether the rank of the centre vector pixel is bigger than the pre-defined rank N_T. Thus the computation of the RCVMF requires $(2C-1)(n(n^2-n)+n(n-1)/2)+2n(n^2-n)+n^3-n$ additions and $C(n(n^2-n)+n(n-1)/2))+2n^2-1$ comparisons.

The RCTVMF requires the computation of an additional L_1-distance $D=d(\boldsymbol{x},\boldsymbol{x}_{(k)})$ involving $2C-1$ additions and C comparisons. Thus it requires $(2C-1)(n(n^2-n)+n(n-1)/2)+2n(n^2-n)+n^3-n+2C-1$ additions and $C(n(n^2-n)+n(n-1)/2))+2n^2+C$ comparisons including 1 comparison to compare D with the threshold.

The vector signal dependent rank-ordered mean (VSD-ROM) filter requires $(2C-1)(n(n^2-n-1)+n(n-1)/2)+2n(n^2-n-1)+n(n^2-2)+C(n^2-1)/2$ additions /subtractions including $C(n^2-1)/2$ additions/subtractions between the centre vector pixel and each of the first $(n^2-1)/2$ vector pixels. It also requires $C(n(n^2-n-1)+n(n-1)/2)+2(n^2-1)\log_2\sqrt{(n^2-1)}+(C+1)(n^2-1)/2$ comparisons including $(n^2-1)\log_2(n^2-1)$ comparisons/swappings to arrange the sums of distances in

ascending order of magnitudes, $C(n^2-1)/2$ comparisons to find the distances between the centre vector pixel and each of the first sorted $(n^2-1)/2$ vector pixels, and $(n^2-1)/2$ more comparisons to compare between these distances with $(n^2-1)/2$ thresholds.

Table 4.1(a) shows the number of various operations required by different filters in a window of size $N = n \times n$ for a *C*-channel image. Table 4.1(b) gives the total number of operations per vector pixel for a three-channel image in a 3×3 window. The asymptotic order of the computational complexity of each vector filter under consideration is of order $O(n^3)$ both for addition and comparison operations. From the analysis, it is observed that the computational complexity of the proposed two filters compares favorably with that of the VMF_1 and the VSD-ROM filter.

Table 4.1(a) Computational complexity of various vector median filters

Filters	**Additions/Subtractions**	**Comparisons**
VMF_1	$(2C-1)(n(n^2-n)+n(n-1)/2)$ $+2n(n^2-n)+n^3-n$	$C(n(n^2-n)+n(n-1)/2)$ $+n^2-1$
RCVMF	$(2C-1)(n(n^2-n)+n(n-1)/2)$ $+2n(n^2-n)+n^3-n$	$C(n(n^2-n)+n(n-1)/2)$ $+2n^2-1$
RCTVMF	$(2C-1)(n(n^2-n)+n(n-1)/2)$ $+2n(n^2-n)+n^3-n+2C-1$	$C(n(n^2-n)+n(n-1)/2)$ $+2n^2+C$
VSD-ROM	$(2C-1)(n(n^2-n-1)+n(n-1)/2)$ $+2n(n^2-n-1)+n(n^2-2)+C(n^2-1)/2$	$C(n(n^2-n-1)+n(n-1)/2)$ $+(n^2-1)\log_2(n^2-1)+(C+1)(n^2-1)/2$

Table 4.1(b) Computational complexities of various vector median filters in terms of number of operations for a 3×3 window

Filters	**Additions/Subtractions**	**Comparisons**
VMF_1	165	71
RCVMF $(N_T = 5)$	165	80
RCTVMF $(N_T = 5)$	170	84
VSD-ROM filter	153	94

4.9 Experimental Results

The proposed algorithms were tested on a number of 24-bit, 512×512 RGB colour images selected from the test image database. The test images considered were the *Lena, Mandrill, Lake, Goldhill, Hut, Tree, Tulip, Cat, Miramar, Point Loma, Golden Gate, Everest, Terrain, Texmos1, Texmos3, Woodland Hill, Brain* and *Stomach.* Fixed-valued and random-valued impulses were artificially injected into these images at various noise ratios according to the colour impulse noise model given in Equation 2.10. The fixed-valued impulse took the values of 0 and 255 with an equal probability of 0.5, while for random-valued impulse, the noise values were uniformly distributed in the range of [0, 255]. The L_1-norm was used for computation of the distances in all experiments except in the cases otherwise stated.

The performance was evaluated using the criteria described in the earlier chapter. The *PSNR* value for the colour images is expressed as follows:

$$PSNR = 10 \log_{10} \left(\frac{I_{\text{MAX}}^2}{MSE} \right) \tag{4.19}$$

where I_{MAX} is the maximum of the three components of all the vector pixels over the original image and the MSE represents the mean square error between the filtered image and the original image. The *MSE* is given by

$$MSE = \frac{1}{M_1 M_2 C} \sum_{m=1}^{M_1} \sum_{n=1}^{M_2} \left| \boldsymbol{x}(m,n) - \hat{\boldsymbol{x}}(m,n) \right|^2 \tag{4.20}$$

where $M_1 \times M_2$ is the size of the image, C is the number of channels of the image and $\boldsymbol{x}(m,n)$ and $\hat{\boldsymbol{x}}(m,n)$ are the original and output vector pixels respectively at location (m,n). Following experiments were carried out.

Experiment 4.9.1: Selection of the Trimming set for the RCVMF

The first experiment was performed to decide the value of the trimming set parameter N_T for the RCVMF. As pointed out earlier, a lower value of N_T will lead to filtering more

number of vector pixels resulting in blurring while a large N_T will leave noisy pixels unfiltered. Figure 4.1 shows the *PSNR* performance of the RCVMF and RCTVMF at different fixed-valued and random-valued impulse noise ratios on the Lena and Mandrill images. As expected, higher values of N_T gave better *PSNR* performance at low noise ratios while for higher noise ratios, the best *PSNR* performance was observed near $N_T = 5$. N_T is observed to be independent of the type of impulse noise.

Experiment 4.9.2: Selection of the Threshold for the RCTVMF

Having selected the rank of the vector trimming set, we have to select a proper value for the threshold θ. The dependence of the *PSNR* value on the threshold in case of the Lena and Mandrill images is shown in Tables 4.2 and 4.3. Figure 4.2 graphically represents this dependence. It is observed that in the case of the Lena image the most appropriate values of θ are 40 and 30 for fixed-valued and random-valued impulse noise respectively. Similarly, the most appropriate thresholds are 90 and 80 for fixed-valued and random-valued impulse noise respectively in the case of the Mandrill image. Experimental results on different images suggest that the appropriate threshold lies between 30 and 100 for fixed-valued impulse and between 20 and 90 for random-valued impulse.

Experiment 4.9.3: Improvement of Performance over the VMF_1 by Rank-conditioning and Thresholding

This experiment investigates the improvement of performance over the VMF_1 by the inclusion of the detection step based on rank-conditioning and thresholding. Table 4.4 shows the relative *PSNR* performance of the VMF_1, RCVMF and RCTVMF on the Lena and Mandrill images. The value of N_T was fixed at 5 for the RCVMF and RCTVMF. Thresholds were set by experimentation at 40 and 30 in the case of the Lena image and at 90 and 80 in the case of the Mandrill image for fixed-valued and random-valued impulse respectively. The significant improvement in *PSNR* performance by inclusion of the detection step based on rank-conditioning and thresolding is obvious from Table 4.4.

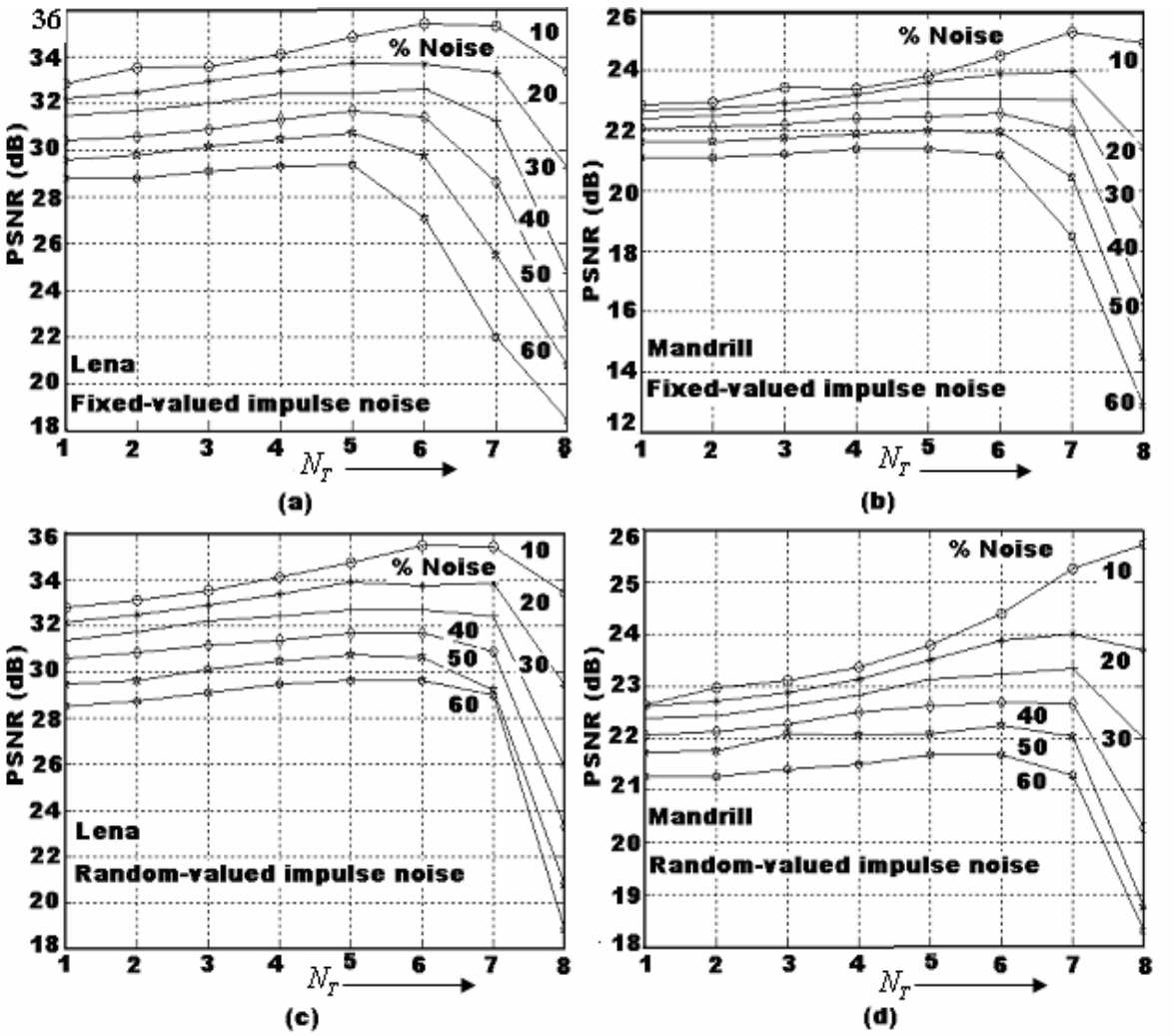


Figure 4.1a N_T – *PSNR* plots for the RCVMF on the Lena and Mandrill images at different impulse noise ratios: (a) and (b) for fixed-valued impulse noise; (c) and (d) for random-valued impulse noise.

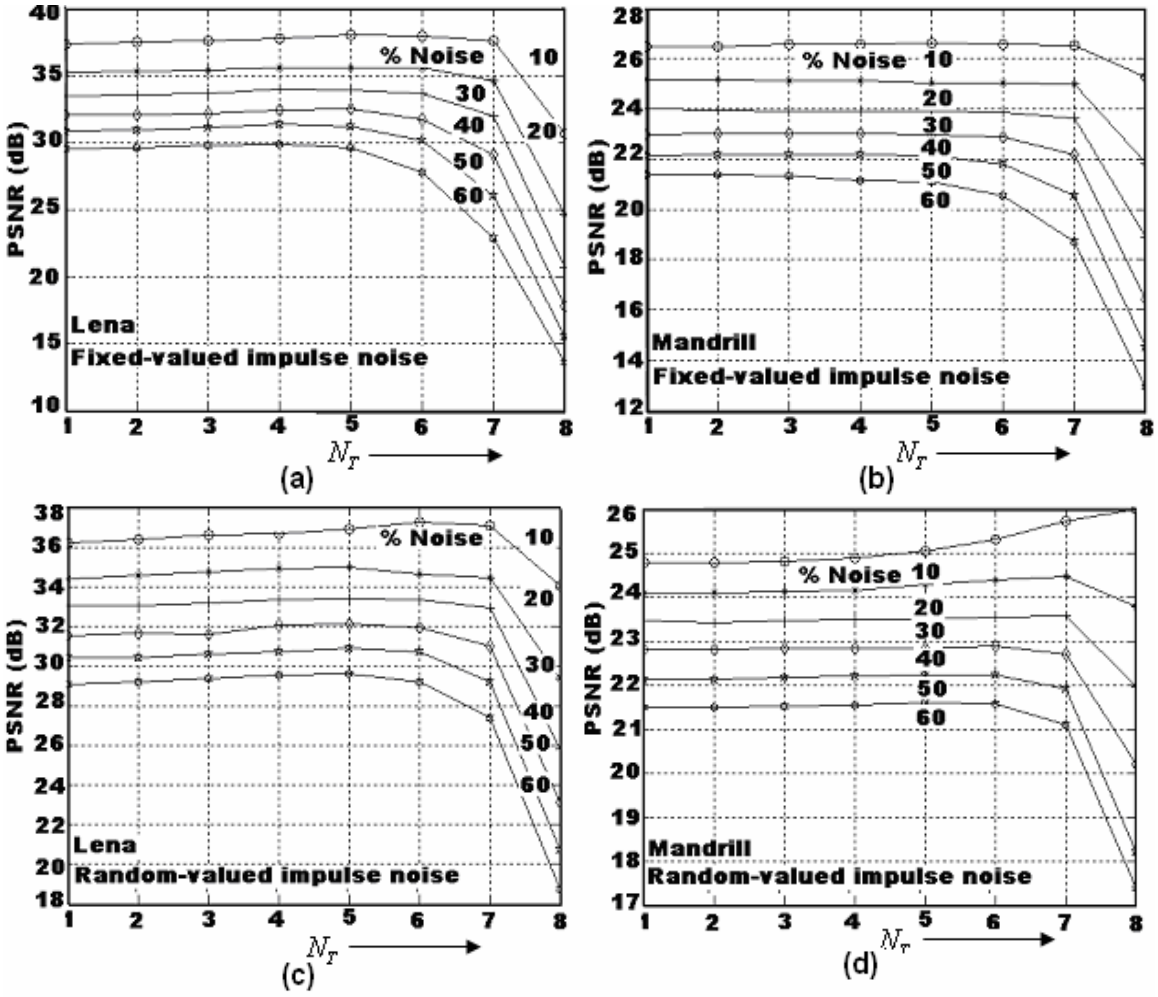


Figure 4.1b N_T – *PSNR* plots for the RCTVMF on the Lena and Mandrill images at different impulse noise ratios: (a) and (b) for fixed-valued impulse noise; (c) and (d) for random-valued impulse noise.

Table 4.2 Dependence of *PSNR* of the RCTVMF output on threshold - θ in the case of the Lena image

% Noise	Threshold θ for fixed-valued impulse noise						
	10	**20**	**30**	**40**	**50**	**60**	**70**
10	35.40	36.62	37.50	38.07	38.27	38.00	36.74
20	34.10	34.77	35.14	35.48	35.43	34.92	33.74
30	32.85	33.52	33.78	34.03	33.76	33.12	31.85
40	31.80	32.31	32.38	32.47	32.30	31.57	30.47
50	30.88	31.19	31.28	31.30	30.87	30.23	29.11
60	29.67	29.60	29.82	29.61	29.34	28.85	27.70
	Random-valued impulse noise						
10	35.29	36.40	37.04	36.98	36.36	35.43	34.09
20	34.10	34.87	34.96	34.53	33.73	32.57	31.13
30	32.72	33.30	33.33	33.06	32.02	30.74	29.35
40	31.87	32.20	32.10	31.65	30.67	29.40	27.99
50	30.93	31.11	30.85	30.29	29.34	28.22	26.14
60	29.73	29.88	29.65	29.10	28.25	27.02	25.74

Table 4.3 Dependence of *PSNR* of the RCTVMF output on threshold - θ in the case of the Mandrill image

% Noise	Threshold θ for fixed-valued impulse noise							
	30	**40**	**50**	**60**	**70**	**80**	**90**	**100**
10	24.11	24.40	24.79	25.17	25.75	26.21	26.58	26.89
20	23.82	24.00	24.25	24.52	24.82	25.01	25.07	25.01
30	23.20	23.33	23.50	23.75	23.86	23.98	23.96	23.72
40	22.66	22.80	22.98	23.14	23.19	23.19	22.98	22.66
50	22.12	22.27	22.41	22.50	22.48	22.34	22.12	21.68
60	21.61	21.64	21.73	21.74	21.66	21.46	21.16	20.67
	Random-valued impulse noise							
10	23.43	23.45	23.58	23.74	24.05	24.57	25.34	25.82
20	23.13	23.15	23.25	23.42	23.69	24.04	24.12	23.68
30	22.76	22.82	22.90	23.07	23.23	23.25	23.43	21.99
40	22.39	22.42	22.50	22.68	22.68	22.81	22.74	20.24
50	21.94	22.00	22.08	22.16	22.22	22.28	22.07	18.72
60	21.47	21.46	21.51	21.61	21.76	21.73	21.28	17.30

Experiment 4.9.4: Selection of the Colour Noise Model

We have used the colour impulse noise model given in Equation 2.10 that considers the corruption of single, double and triple channels of the vector pixel. The colour impulse model in Equation 2.9 used by Viero *et al.* [38] considers corruption of single and triple channels. A set of experiments was performed for comparison of filtering performance under both colour noise models. For the sake of convenience, equal inter-channel noise factors $p_1 = p_2 = p_3$ with $p_1 = 0.166,\ 0.2,\ 0.233$ and equal double-channel noise factors $p_4 = p_5 = p_6 = 0.1$ were assumed. Tables 4.5, 4.6, 4.7 and 4.8 give the restoration results of the VMF_1 and the proposed RCTVMF in removal of fixed-valued and random-valued impulse noise on the Lena image corrupted with 10% to 60% impulse noise ratios.

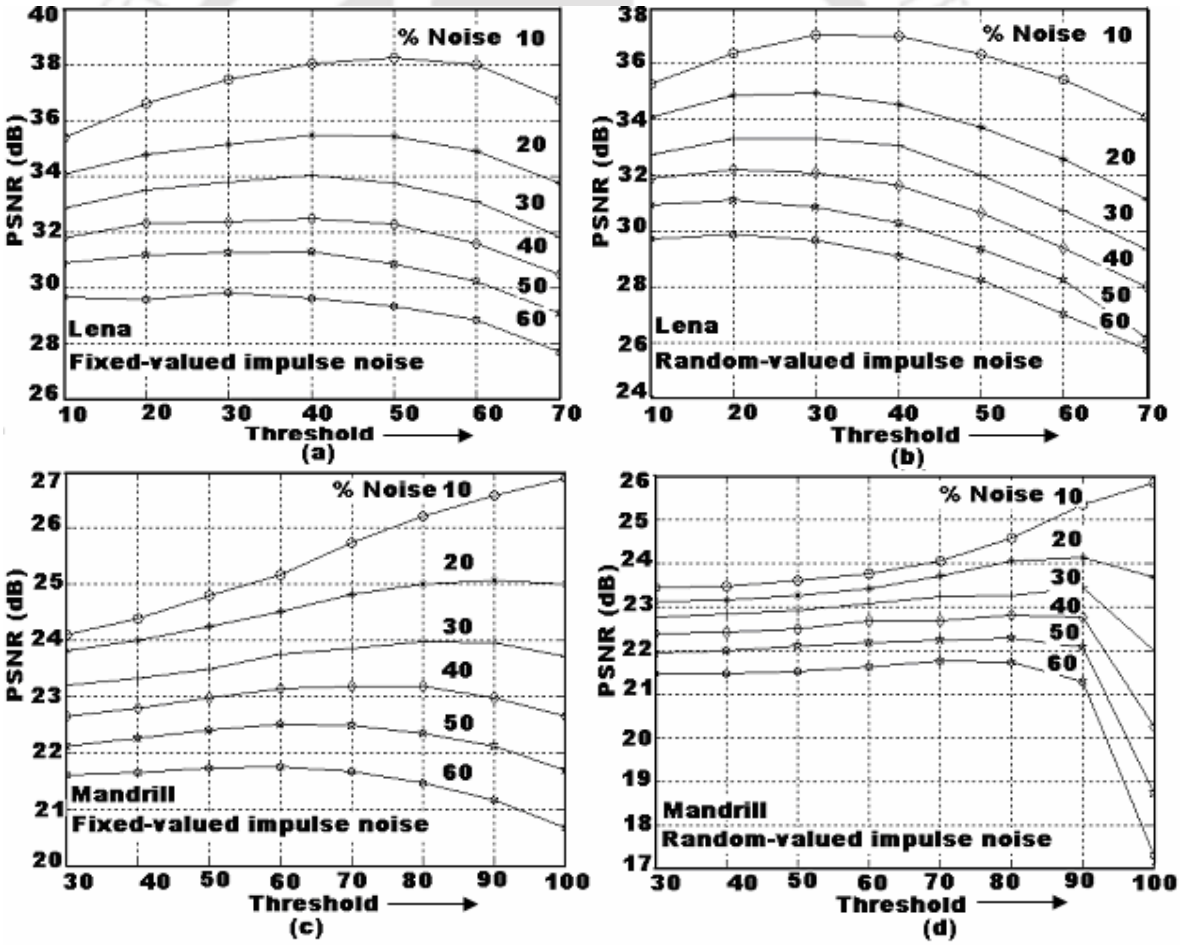


Figure 4.2 Threshold - *PSNR* curve (a) and (c) for fixed-valued impulse noise, and (b) and (d) for random-valued impulse noise, both on the Lena and Mandrill images.

Direct comparison between two models is not possible, because the model used by Viero *et al*. considers single and triple channel corruptions only and the proposed colour noise model considers single, double and triple channel corruptions. The *PSNR* is higher for the same noise ratio with the larger value of single channel noise factors in both models, because the total number of corrupted vector pixel components is less. For the same single channel noise factor, slightly lower *PSNR* results are observed in the case of the image corrupted according to colour noise model of Equation 2.9 than that of the image corrupted by the colour noise model of Equation 2.10. This is because the number of corrupted components of vector pixels is more in the case of corruption by the former model.

Table 4.4 Comparison of the *PSNR* performance of the VMF_1, RCVMF and RCTVMF on the Lena and Mandrill images

(a) Lena image						
% Noise	**Fixed-value impulse noise**			**Random-valued impulse noise**		
	VMF_1	**RCVMF**	**RCTVMF**	**VMF_1**	**RCVMF**	**RCTVMF**
10	32.74	34.75	38.07	32.68	34.70	37.04
20	32.14	33.69	35.48	32.11	33.97	34.96
30	31.36	32.47	34.03	31.43	32.58	33.33
40	30.38	31.57	32.47	30.48	31.68	32.10
50	29.54	30.38	31.30	29.36	30.58	30.85
60	28.67	29.26	29.61	28.41	29.64	29.65
(b) Mandrill image						
% Noise	**Fixed-value impulse noise**			**Random-valued impulse noise**		
	VMF_1	**RCVMF**	**RCTVMF**	**VMF_1**	**RCVMF**	**RCTVMF**
10	22.87	24.52	26.58	22.84	24.41	24.57
20	22.67	23.57	25.07	22.61	23.51	24.04
30	22.42	23.01	23.96	22.33	23.21	23.25
40	22.04	22.16	22.98	22.02	22.30	22.81
50	21.64	21.12	22.12	21.66	21.17	22.28
60	20.98	19.54	21.16	21.22	19.98	21.73

Experiment 4.9.5: Comparison with other Vector filters at different Noise Ratios

The next set of experiments was performed for comparison of the proposed RCVMF and RCTVMF with some of the standard vector filters. The following filters were used for comparison:

(1) Vector median filter (VMF_1) based on L_1 - norm
(2) Vector median filter (VMF_2) based on L_2 - norm
(3) Marginal median filter (MMF)
(4) Basic vector directional filter (BVDF)
(5) Directional distance filter (DDF)
(6) Centre-weighted vector median filter (CWVMF) and
(7) Vector signal dependent rank-ordered mean (VSD-ROM) filter

In all the cases, a window of size 3×3 was used. The optimal set of thresholds for the VSD-ROM filter was selected by trial and error. The value of the centre weight for the CWVMF was 2 throughout the experiments.

Table 4.5 *PSNR* performance of the VMF_1 on the Lena image corrupted with fixed-valued impulse noise according to the colour impulse noise models of Equation 2.9 and Equation 2.10

% Noise	**Model used by Viero *et al*** $p_1 = p_2 = p_3$			**Proposed Model** $p_1 = p_2 = p_3$ $p_4 = p_5 = p_6 = 0.1$		
	$p_1 = 0.166$	$p_1 = 0.2$	$p_1 = 0.233$	$p_1 = 0.166$	$p_1 = 0.2$	$p_1 = 0.233$
10	32.68	32.73	32.79	32.74	32.81	32.88
20	31.98	32.14	32.23	32.14	32.35	32.45
30	30.79	31.29	31.50	31.36	31.64	31.79
40	29.91	30.16	30.49	30.38	30.79	30.83
50	28.89	29.38	29.80	29.54	29.92	30.38
60	27.24	28.24	28.76	28.67	28.24	29.59

Table 4.6 *PSNR* performance of the VMF_1 on the Lena image corrupted with random-valued impulse noise according to the colour impulse noise models of Equation 2.9 and Equation 2.10

% Noise	**Colour Impulse Noise Model of Viero *et al*** $p_1 = p_2 = p_3$			**Proposed Colour Impulse Noise Model** $p_1 = p_2 = p_3$ and $p_4 = p_5 = p_6 = 0.1$		
	$p_1 = 0.166$	$p_1 = 0.2$	$p_1 = 0.233$	$p_1 = 0.166$	$p_1 = 0.2$	$p_1 = 0.233$
10	32.57	32.68	32.75	32.68	32.80	32.88
20	31.86	32.05	32.22	32.11	32.24	32.43
30	30.96	31.23	31.39	31.43	31.66	31.90
40	29.80	30.18	30.72	30.48	30.93	31.24
50	28.74	28.97	29.67	29.36	29.86	30.48
60	27.41	27.60	28.56	28.41	29.04	29.68

Table 4.7 *PSNR* performance of the RCTVMF on the Lena image corrupted with fixed-valued impulse noise according to the colour impulse noise models of Equation 2.9 and Equation 2.10

% Noise	**Colour Impulse Noise Model used of Viero *et al*** $p_1 = p_2 = p_3$			**Proposed Colour Impulse Noise Model** $p_1 = p_2 = p_3$ and $p_4 = p_5 = p_6 = 0.1$		
	$p_1 = 0.166$	$p_1 = 0.2$	$p_1 = 0.233$	$p_1 = 0.166$	$p_1 = 0.2$	$p_1 = 0.233$
10	37.91	37.82	37.82	38.07	37.89	38.01
20	35.28	35.36	35.45	35.48	35.43	35.58
30	33.71	33.71	33.73	34.03	33.93	34.02
40	32.16	32.29	32.53	32.47	32.58	32.71
50	30.60	30.71	31.02	31.30	31.46	31.46
60	28.81	29.13	29.50	29.61	29.89	30.17

Table 4.8 *PSNR* performance of the RCTVMF on the Lena image corrupted with random-valued impulse noise according to the colour impulse noise models of Equation 2.9 and Equation 2.10

% Noise	**Colour Impulse Noise Model by Viero *et al*** $p_1 = p_2 = p_3$			**Proposed Colour Impulse Noise Model** $p_1 = p_2 = p_3$ and $p_4 = p_5 = p_6 = 0.1$		
	$p_1 = 0.166$	$p_1 = 0.2$	$p_1 = 0.233$	$p_1 = 0.166$	$p_1 = 0.2$	$p_1 = 0.233$
10	36.93	36.94	36.96	37.04	37.02	36.40
20	34.82	34.91	37.40	34.96	35.11	34.90
30	33.19	33.19	33.18	33.33	33.52	33.41
40	31.76	31.79	31.83	32.10	32.35	32.37
50	30.48	30.49	30.46	30.85	31.18	31.46
60	29.06	28.92	29.01	29.65	30.02	30.47

Tables 4.9 and 4.10 list the restoration results of various filters in removal of fixed-valued and random-valued impulse noise from the Lena image, corrupted with 10% to 60% impulse noise ratios for inter-channel noise factors set at $p_1 = p_2 = p_3 = 0.166$ and $p_4 = p_5 = p_6 = 0.10$. At these values of noise factors, 50% of the noisy vector pixels are corrupted only in single channel, 30% are double channels and 20% are corrupted in triple channels. Data from Tables 4.9 and 4.10 were plotted graphically in Figure 4.3. It has been seen from these tables and the figure that the performance of the BVDF is inferior to all filters mentioned here. This is because of the insensitivity of the BVDF to identify luminance outliers. The BVDF is followed by the MMF in *PSNR* performance. Component wise processing by MMF gives inferior results compared to the VMF_1. The performance of the VMF_2 is slightly inferior to that of the VMF_1. It shows that vector median filter based on the L_1-metric outperforms one based on the Euclidean distance in removal of impulse noise from colour images.

The performance of the CWVMF was better than that of the VMF_1, VMF_2, MMF, BVDF and DDF at low impulse noise ratios, but it became inferior at high noise ratios. It

is due to the fact that the impulse noise dominates at high impulse ratio when more weight is given to the centre vector pixel. The performance of the RCVMF is observed to be better than that of the VMF_1, VMF_2, MMF, BVDF, DDF and CWVMF at different noise ratios. This is one of the remarkable results. It has been found experimentally that the most appropriate value of the rank N_T of the healthy vector pixel for the RCVMF was 5. The *PSNR* performance of the RCVMF on the Lena image at different noise ratios was found to be inferior to that of the VSD-ROM filter.Thresholds for the VSD-ROM filter were set at 30, 60, 100 and 140 for random-valued impulse noise and at 40, 70, 120 and 160 for fixed-valued impulse noise. The *PSNR*-performance RCTVMF was found to be superior to that of VMF_1, VMF_2, MMF, BVDF, DDF, CWVMF, RCVMF and VSD-ROM filter.

Figures 4.4 and 4.5 show the outputs of the VMF_1, VMF_2, MMF, CWVMF, RCVMF, VSD-ROM filter and RCTVMF in removal of fixed-valued and random-valued impulse noise respectively from the Lena image corrupted with impulse noise at a noise ratio p = 0.60 ($p_1 = p_2 = p_3 = 0.166$ and $p_4 = p_5 = p_6 = 0.10$). From the visual observation, it has been observed that the RCTVMF gives better result in comparison with other filters under consideration. The RCTVMF preserves the multidirectional and curved edges, smooth regions with fine textures (hat-ribbon), and the high-frequency irregular detailed structure (feather) in the Lena image. These regions are marked with arrows in the filtered outputs shown in Figures 4.4a(i) and 4.5a(i). The RCTVMF gives less blurring effect and removes impulse noise better than other filters. It is obvious from the magnitudes of differences between the original Lena image and outputs of the VMF_1, VMF_2, MMF, CWVMF, RCVMF, VSD-ROM filter and RCTVMF that are shown in Figures 4.4b and 4.5b. Here the differences were magnified two times for clear visibility. The impulse remnants and blurring are less in the case of the RCTVMF. The enlarged face parts of the VMF_1, VSD-ROM filter and RCTVMF outputs are shown in Figures 4.4c and 4.5c. The distortion of edges that could not be seen clearly in the original filtered outputs is seen in these figures. There is more blurring in the VMF_1 output. Such effects are less in the case of the RCTVMF.

Table 4.9 Comparative performance of various filters on the Lena image at different fixed-valued impulse noise ratios

% Noise	VMF_1	VMF_2	MMF	BVDF	DDF	CWV MF	RCV MF	VSD-ROM	RCT VMF
10	32.74	32.53	32.37	31.10	32.46	33.89	34.75	36.77	38.07
20	32.14	31.63	31.68	30.14	31.80	32.87	33.69	34.70	35.48
30	31.36	30.54	30.71	29.01	31.04	31.65	32.47	32.90	34.03
40	30.38	29.44	29.68	27.55	29.73	30.46	31.57	31.41	32.47
50	29.54	28.21	28.71	26.32	28.96	28.75	30.38	30.08	31.30
60	28.67	27.00	27.85	24.44	27.96	26.92	29.26	28.75	29.61

Table 4.10 Comparative performance of various filters on the Lena image at different random-valued impulse noise ratios

% Noise	VMF_1	VMF_2	MMF	BVDF	DDF	CWV MF	RCV MF	VSD-ROM	RCT VMF
10	32.68	32.50	32.41	31.07	32.41	33.98	34.70	36.21	37.04
20	32.11	31.66	31.65	30.12	31.77	33.17	33.97	34.03	34.96
30	31.43	30.72	30.71	28.96	30.97	32.25	32.58	32.40	33.33
40	30.48	29.61	29.72	27.88	30.02	31.08	31.68	30.75	32.10
50	29.36	28.30	28.75	26.24	28.97	30.02	30.58	29.52	30.85
60	28.41	27.03	27.52	24.40	27.80	28.75	29.64	28.22	29.65

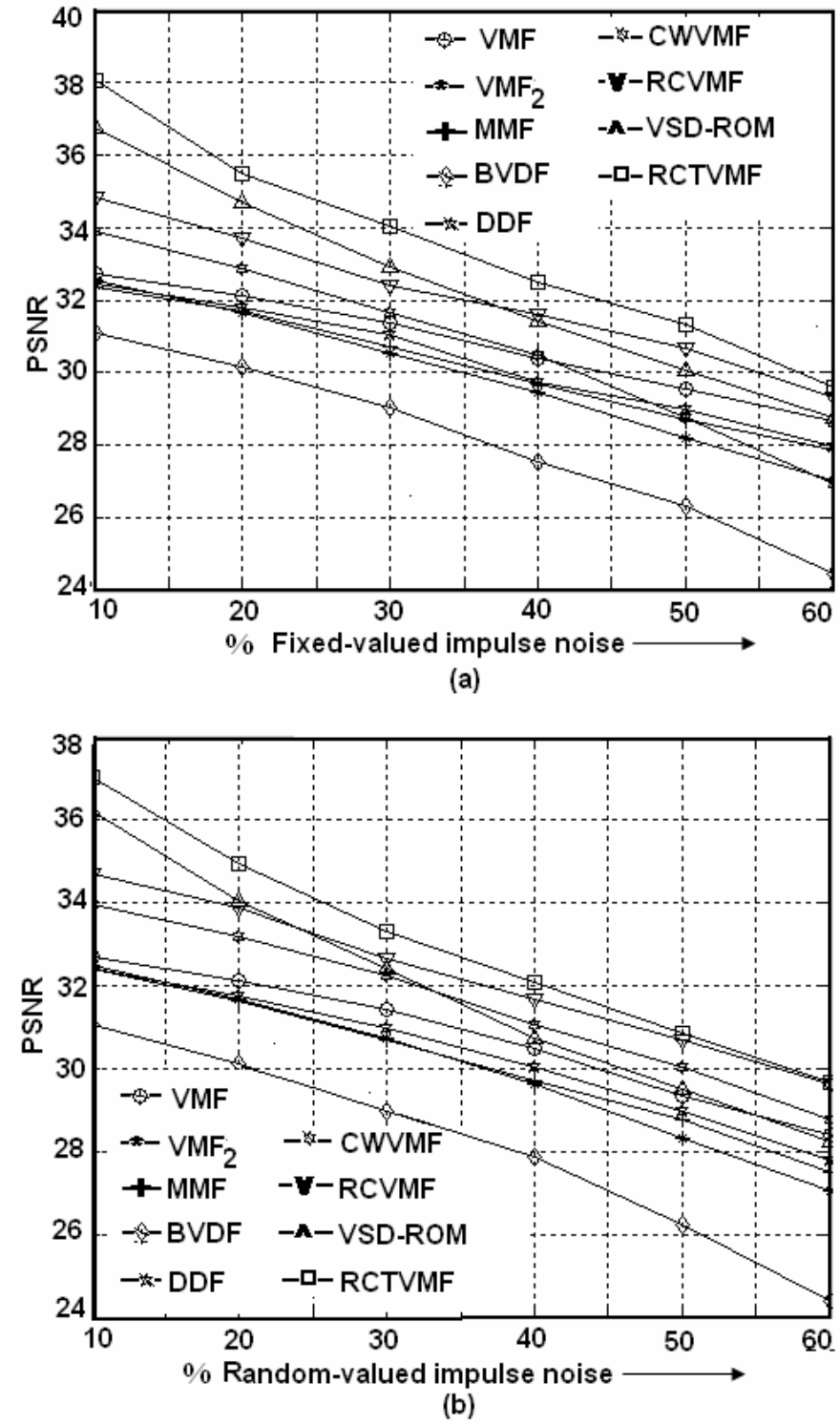


Figure 4.3 Noise ratio – *PSNR* curve for different vector filters on the Lena image: (a) removal of fixed-valued impulse noise and (b) removal of random-valued impulse noise.

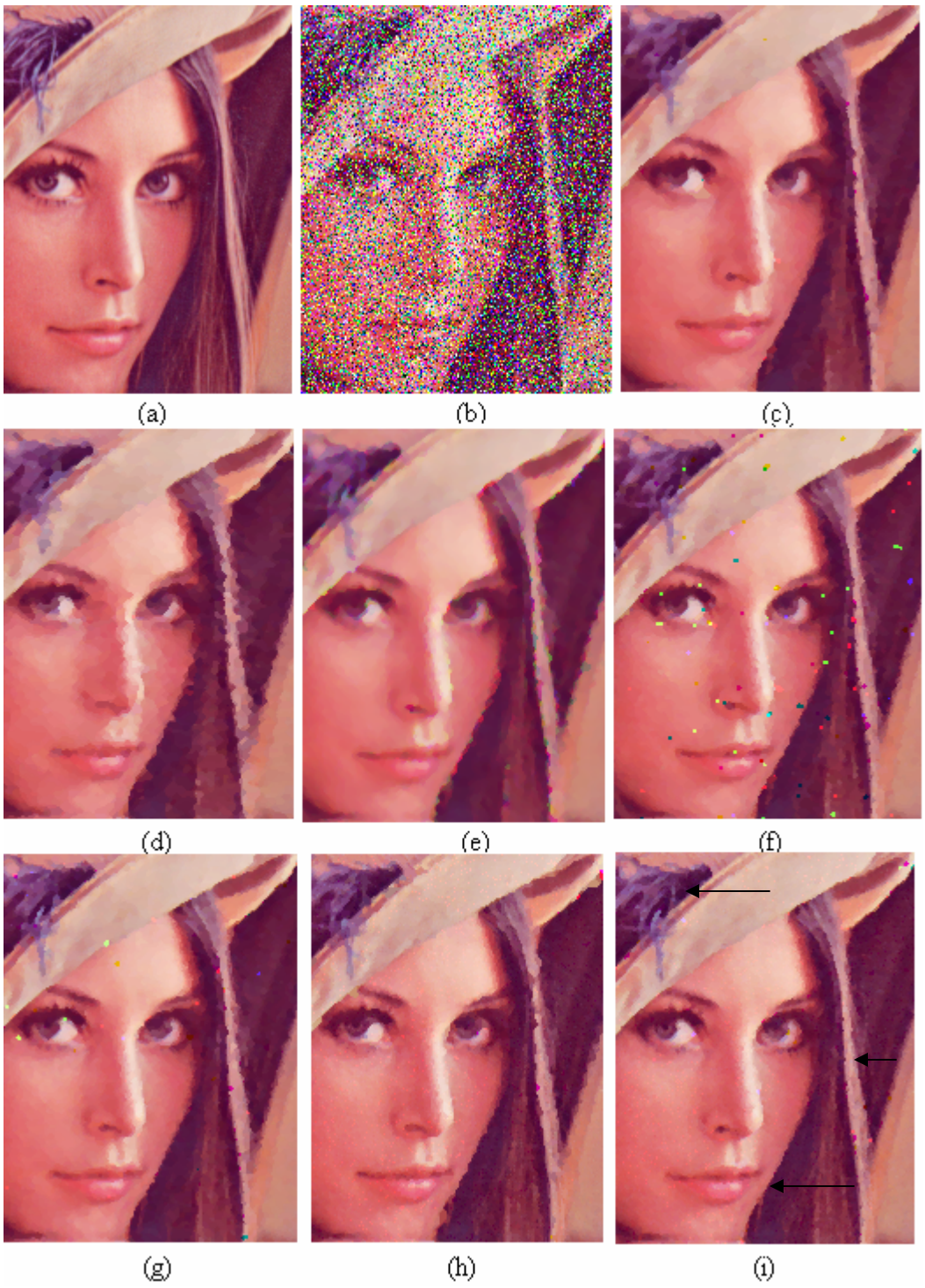


Figure 4.4a Visual performance of various vector filters on the Lena image: (a) original image (face portion), (b) image with 60% fixed-valued impulse noise, (c), (d), (e), (f), (g), (h) and (i) -- outputs of the VMF_1, VMF_2, MMF, CWVMF, RCVMF, VSD-ROM filter and RCTVMF.

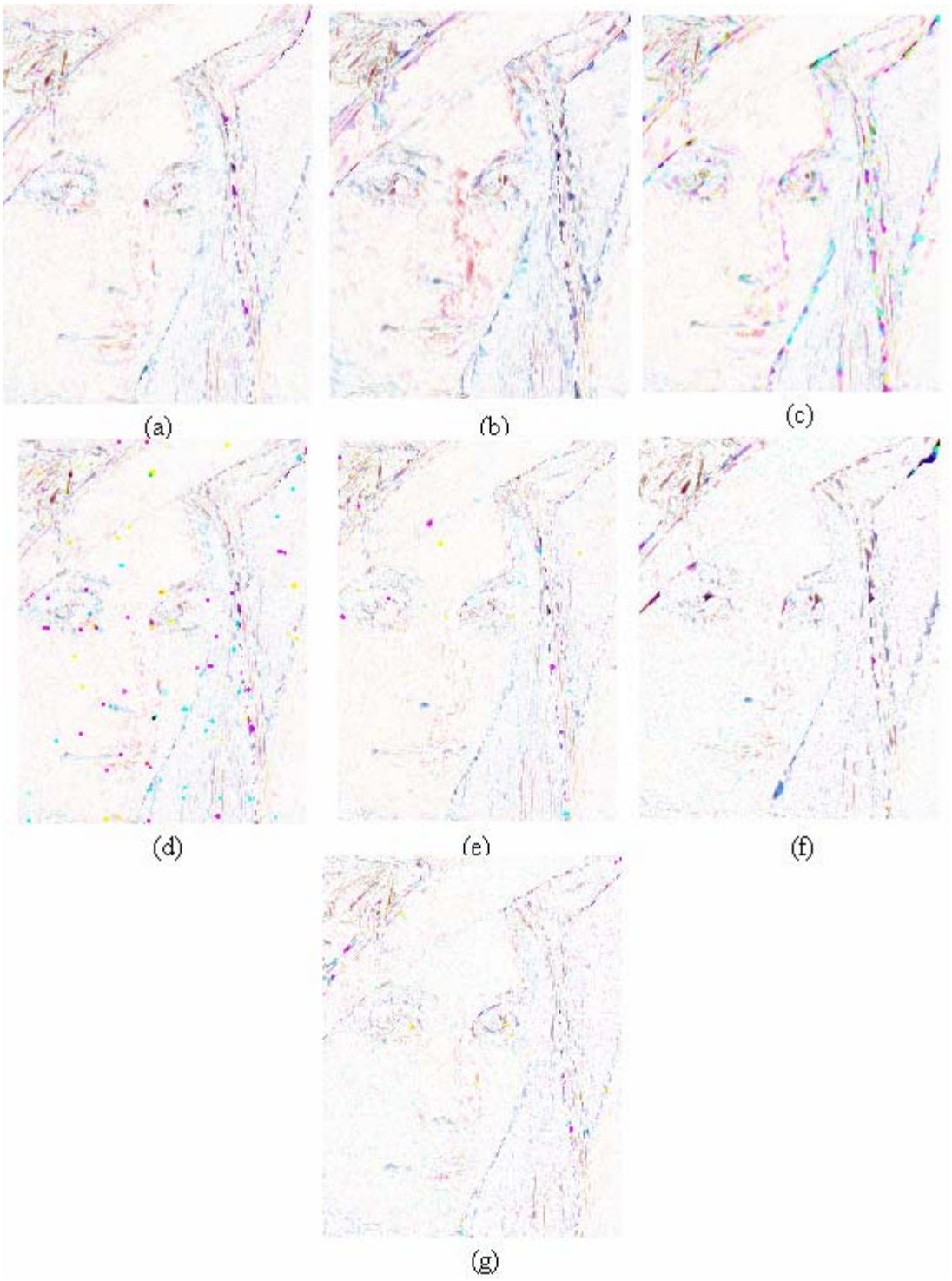


Figure 4.4b Magnitudes of differences between the original Lena image and outputs of filters for the image corrupted with 60% fixed-valued impulse noise: (a), (b), (c), (d), (e), (f) and (g) -- magnitudes of differences of the VMF_1, VMF_2, MMF, CWVMF, RCVMF, VSD-ROM filter and RCTVMF.

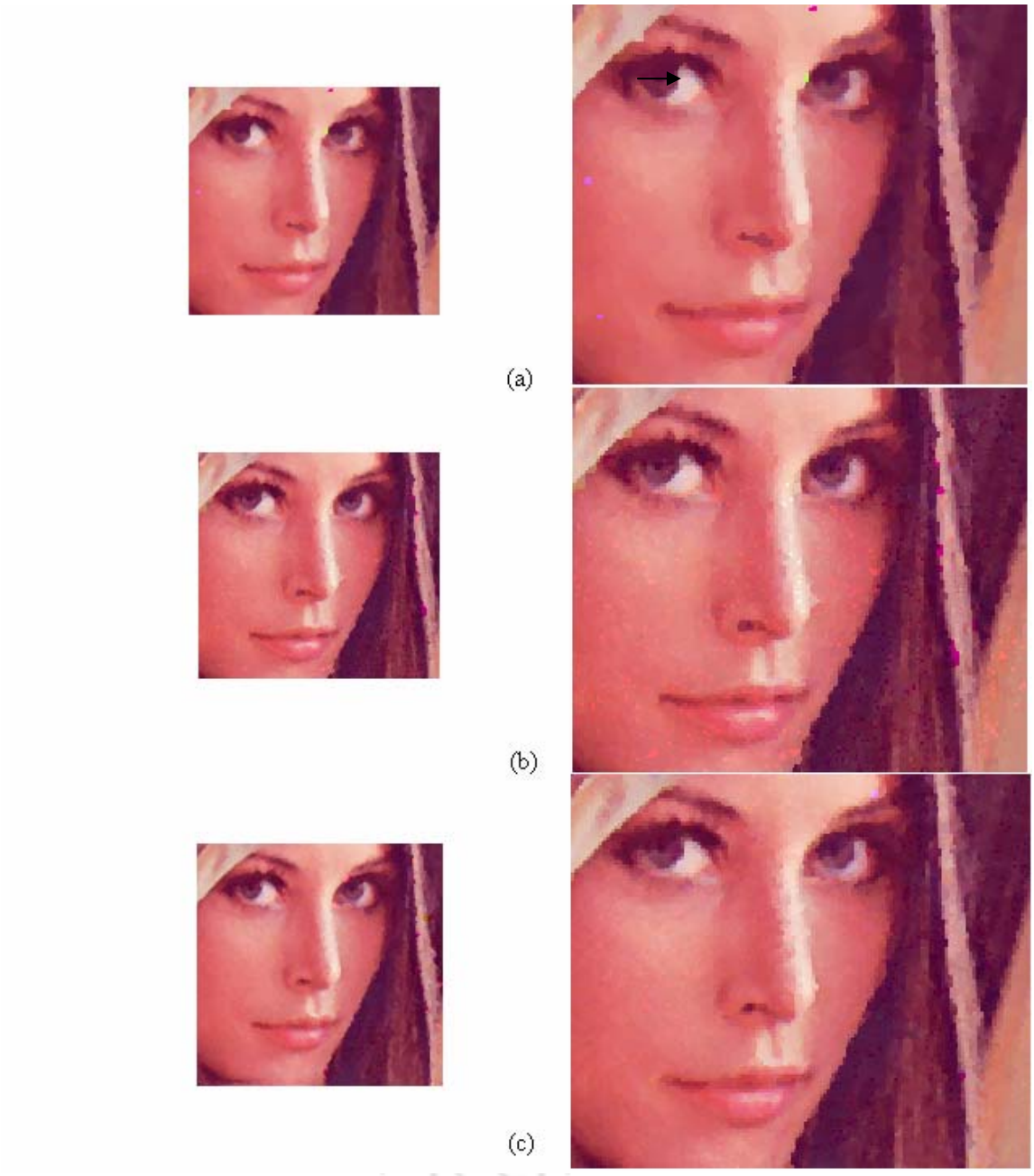


Figure 4.4c Enlarged outputs of the Lena image (face part portion) corrupted with 60% fixed-valued impulse noise: (a) VMF_1, (b) VSD-ROM filter and (c) RCTVMF.

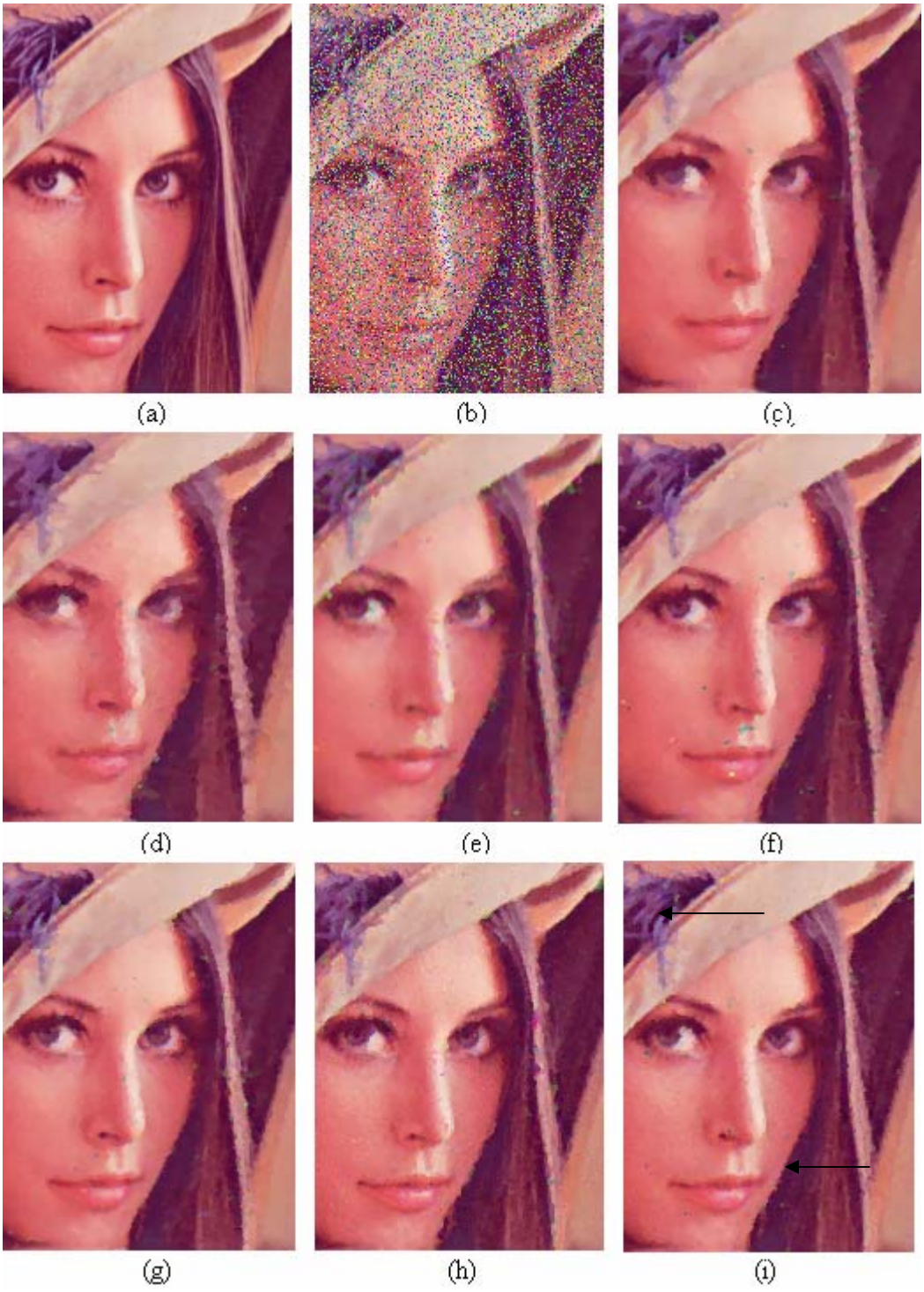


Figure 4.5a Visual performance of various vector filters on the Lena image: (a) original image (face portion), (b) image with 60% random-valued impulse noise, (c), (d), (e), (f), (g), (h) and (i) -- outputs of the VMF_1, VMF_2, MMF, CWVMF, RCVMF, VSD-ROM filter and RCTVMF.

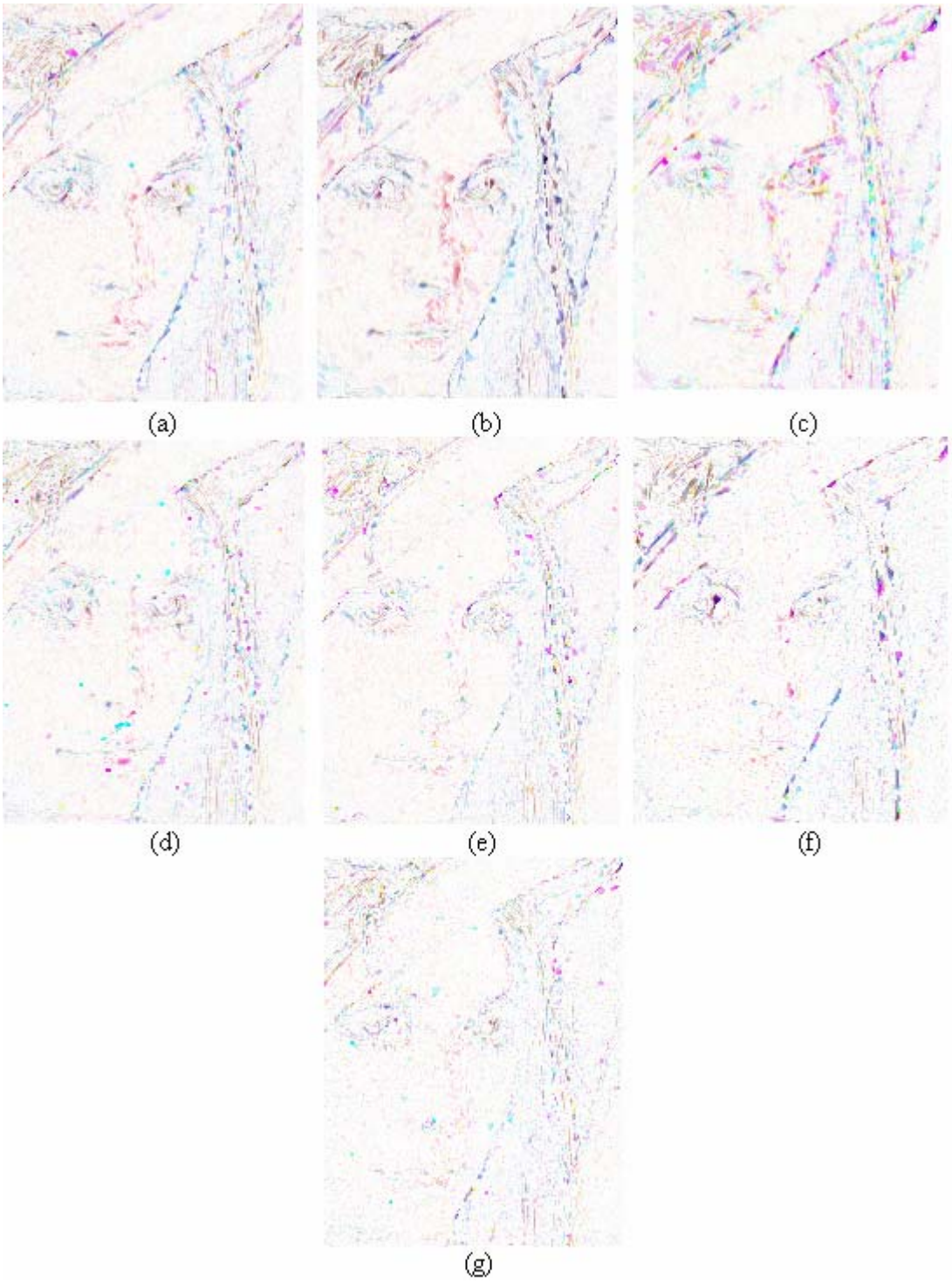


Figure 4.5b Magnitudes of differences between the original Lena image and outputs of filters for the image corrupted with 60% random-valued impulse noise: (a), (b), (c), (d), (e), (f) and (g) -- magnitudes of differences of the VMF_1, VMF_2, MMF, CWVMF, RCVMF, VSD-ROM filter and RCTVMF.

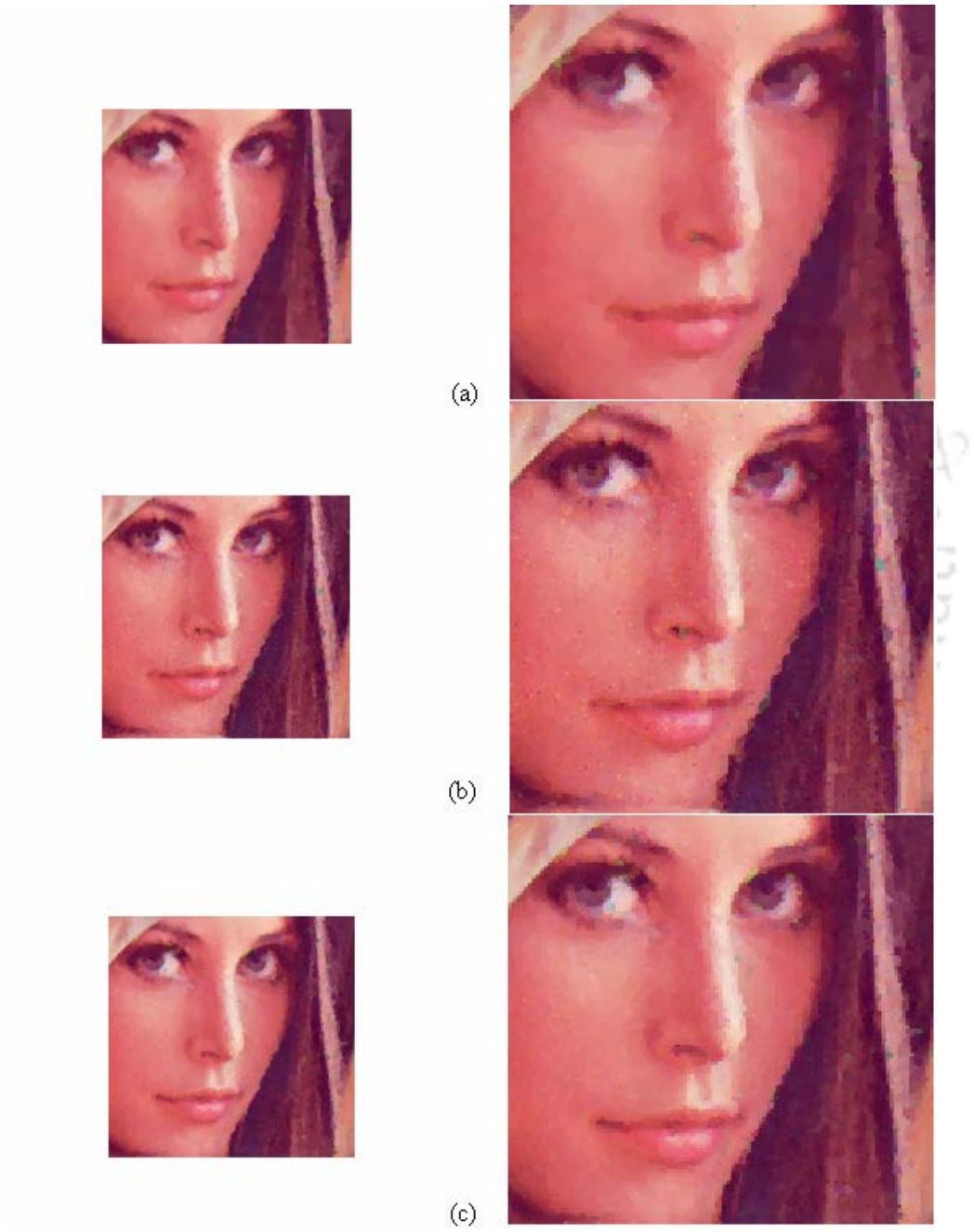


Figure 4.5c Enlarged outputs of the Lena image (face part portion) corrupted with 60% random-valued impulse noise: (a) VMF_1, (b) VSD-ROM filter and (c) RCTVMF.

Experiment 4.9.6: Performace of the Rank-conditioned and Threshold based Impulse Detection

This experiment was performed to study the ability of the proposed RCTVMF in detection of impulse noise. The *true hit*, *false hit*, *miss hit* and *percentage hit ratio* for the detection scheme were computed at different noise ratios for both types of impulse noise. Tables 4.11 and 4.12 show these quantities for the Lena image. The equivalent quantities for the VSD-ROM filter are also included. From these tables, it has been seen that the true hit is almost same for both the filters, while the false hit of the RCTVMF is lower than that of the VSD-ROM filter. This results demonstrate the ability of the proposed detection scheme to distinguish between an impulse and a healthy vector pixel.

Experiment 4.9.7: Comparison with the VSD-ROM Filter at Different Noise Ratios

This experiment was performed for the comparison between the VSD-ROM filter and RCTVMF in removal of the fixed-valued and random-valued impulse noise from the Mandrill, Golden Gate, Point Loma and Texmos3 images. The images were corrupted with fixed-valued and random-valued impulse noise at noise ratios ranging from 10% to 60%. Figures 4.6 and 4.7 show the comparison for fixed-valued and random-valued impulse noise respectively. In the case of fixed-valued impulse noise, the RCTVMF gives better results than the VSD-ROM filter at all different noise ratios except for the Point Loma image below 30% noise ratios. An increase in N_T may give *PSNR* value at par with that of the VSD-ROM filter for this case also. In the case of random-valued impulse noise, the RCTVMF performs uniformly better.

Experiment 4.9.8: Comparison with Other Vector Filters on Different Images

The aim of this experiment was to study the comparative performance of the filters under study on a wide variety of colour images. All images were degraded with p = 20% impulse noise (p_1 = 0.166, p_2 = 0.166, p_3 = 0.166 = p_4 = 0.10, p_5 = 0.10, p_6 = 0.10). Tables 4.13 and 4.14 show the results. It has been observed from these tables that the performance of the RCTVMF is superior to that of the other filters with exception to Point Loma image in removal of fixed-valued impulse noise, for which the VSD-ROM

filter gave better results at 20% impulse noise. Thus the proposed filters gave very stable performance on different types of images.

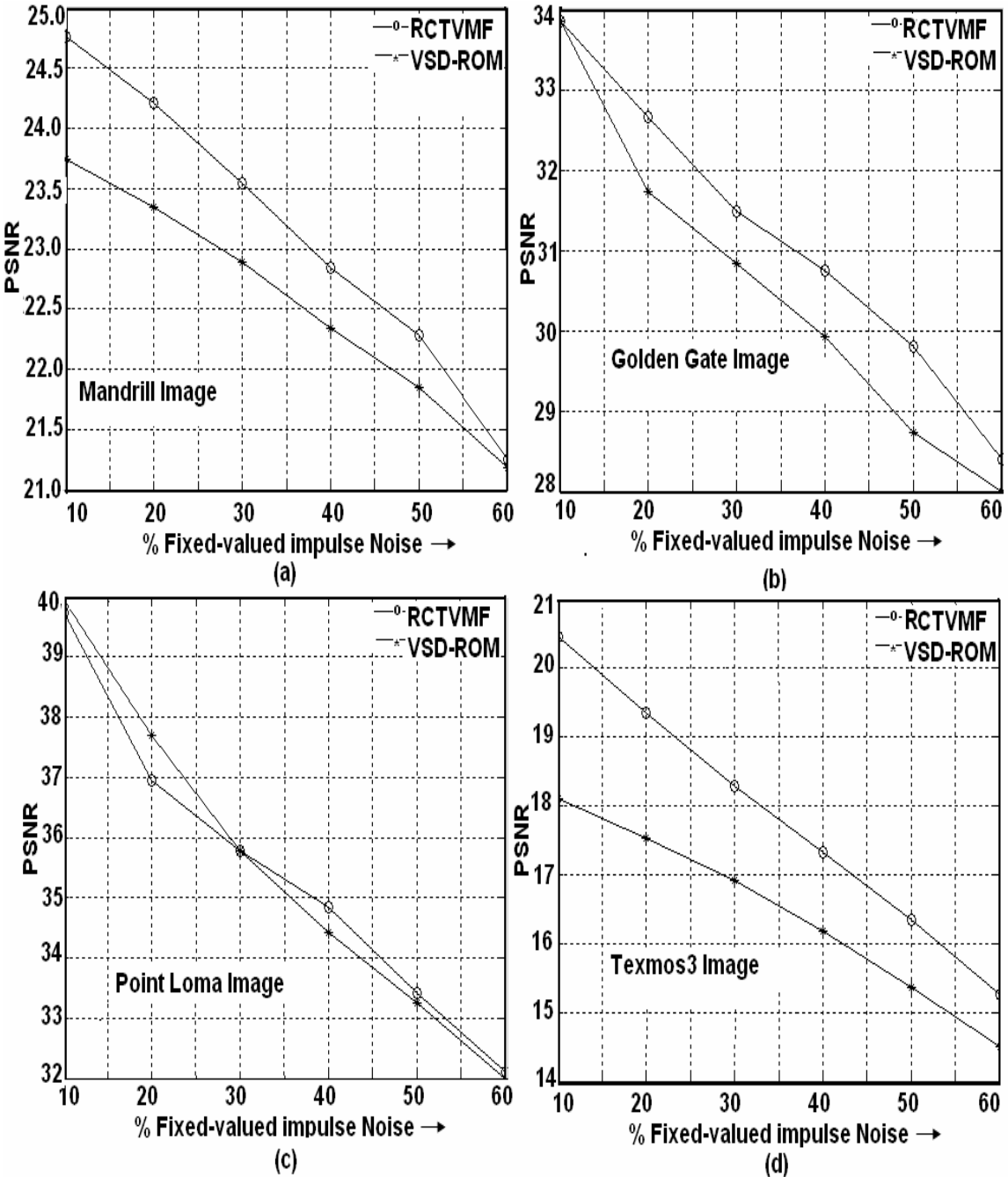


Figure 4.6 Comparison of performance of the RCTVMF and VSD-ROM filter in removal of fixed-valued impulse noise from (a) Mandrill, (b) Golden Gate, (c) Point Loma and (d) Texmos3 images corrupted with 10% to 60% impulse noise ratios.

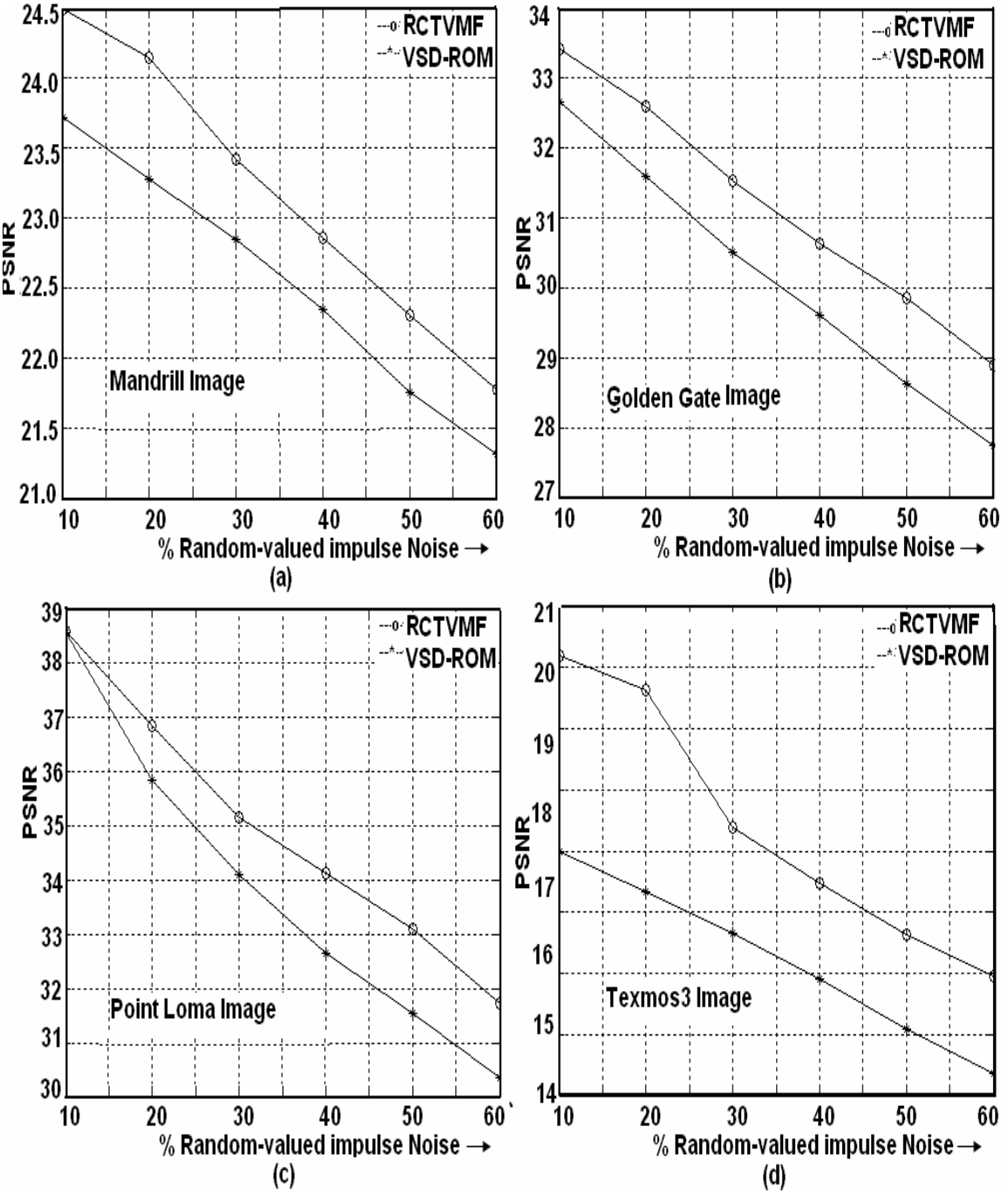


Figure 4.7 Comparison of performance of the RCTVMF and VSD-ROM filter in removal of random-valued impulse noise from (a) Mandrill, (b) Golden Gate, (c) Point Loma and (d) Texmos3 images corrupted with 10% to 60% of impulse noise ratios.

Table 4.11 Number of true hit, false hit and miss hit pixels of the RCTVMF and the VSD-ROM filter in detection of fixed-valued impulse noise on the Lena image

% Impulse noise	**10**	**20**	**30**	**40**	**50**	**60**
Number of corrupted pixels	26214	52429	78643	104860	131070	156744
RCTVMF						
True hit	25454	50552	75560	100530	124630	148702
False hit	987	1014	1059	1108	1247	1403
Miss hit	760	1877	3083	4330	6440	8042
% True hit	97.10	96.52	96.08	95.87	95.09	94.88
VSD-ROM						
True hit	25297	50352	75438	100874	126286	151161
False hit	1507	1883	1921	2868	3165	3647
Miss hit	917	2077	3205	3986	4784	5583
% True hit	96.50	96.04	95.92	96.20	96.35	96.44

Table 4.12 Number of true hit, false hit and miss hit pixels of the RCTVMF and the VSD-ROM filter in detection of random-valued impulse noise on the Lena image

% Impulse noise	**10**	**20**	**30**	**40**	**50**	**60**
Number of corrupted pixels	26214	52429	78643	104860	131070	156744
RCTVMF						
True hit	21933	43664	65486	87517	108810	130402
False hit	1876	1846	1798	1781	1828	1952
Miss hit	4281	8755	13157	17343	22260	26342
% True hit	83.28	83.05	83.08	83.16	82.87	82.84
VSD-ROM						
True hit	21523	43144	64663	86126	108664	130460
False hit	1858	1902	2095	2876	3220	3886
Miss hit	4691	9285	13980	18734	22406	26284
% True hit	82.10	82.29	82.22	82.13	82.91	83.23

Table 4.13 Comparative *PSNR* performance of various filters in removal fixed-valued impulse noise from different images corrupted with 20% noise ratio

Images	Filters								
	VMF_1	**VMF_2**	**MMF**	**BVDF**	**DDF**	**CWVMF**	**RCVMF**	**VSD-ROM**	**RCTVMF**
Lena	32.14	31.63	31.68	30.14	31.80	32.87	33.69	34.70	35.48
Mandrill	22.67	22.51	22.22	21.19	22.64	23.61	23.57	23.35	25.07
Lake	27.12	26.85	26.91	24.06	26.79	28.28	28.93	28.66	30.00
Goldhill	29.52	29.11	29.03	25.70	29.20	30.48	30.93	31.79	32.16
Hut	29.01	28.54	28.00	26.89	28.89	30.22	30.73	29.46	31.31
Tree	28.67	28.24	27.62	27.32	28.60	29.84	30.30	29.65	31.20
Tulip	32.52	31.49	31.34	28.29	32.19	33.30	34.25	33.68	34.79
Cat	29.11	28.86	28.44	24.56	29.05	30.20	30.62	30.13	31.13
Miramar	25.95	25.62	25.44	25.00	25.79	26.96	27.15	27.78	28.28
Point Loma	33.17	32.86	32.89	32.19	33.02	33.79	34.44	37.70	36.95
Golden Gate	30.41	30.27	29.87	32.01	30.36	31.39	31.50	31.69	32.79
Everest	22.80	22.73	22.48	21.84	22.78	23.69	23.96	23.14	23.96
Terrain	20.72	20.47	20.71	18.22	20.38	21.52	22.19	21.16	22.30
Texmos1	17.94	17.50	16.86	8.92	14.61	18.94	19.01	17.43	19.41
Texmos3	17.90	17.58	16.97	9.01	14.91	18.98	18.95	17.54	19.35
Woodland Hill	25.88	25.57	25.39	22.75	25.65	27.16	27.55	27.93	28.68
Brain	27.71	27.20	26.69	23.31	27.39	28.86	28.78	28.30	29.35
Stomach	33.31	33.08	32.76	31.57	33.30	33.71	34.13	36.30	36.31

Table 4.14 Comparative *PSNR* performance of various filters in removal random-valued impulse noise from different images corrupted with 20% noise ratio

Images	Filters								
	VMF_1	**VMF_2**	**MMF**	**BVDF**	**DDF**	**CWVMF**	**RCVMF**	**VSD-ROM**	**RCTVMF**
Lena	32.11	31.66	31.65	30.12	31.77	33.17	33.97	34.03	34.96
Mandrill	22.61	22.51	22.29	21.12	22.57	23.62	23.51	23.28	24.04
Lake	27.09	27.01	26.91	24.46	26.82	28.25	28.99	28.58	29.58
Goldhill	29.44	29.13	29.05	25.74	29.08	30.49	30.87	29.47	31.78
Hut	28.92	28.12	27.89	26.60	28.47	30.27	30.50	31.53	31.15
Tree	28.64	26.43	27.58	26.64	27.62	29.92	30.20	30.91	30.89
Tulip	32.12	31.65	31.30	28.26	32.06	33.64	34.28	33.38	34.84
Cat	27.90	27.73	28.00	24.97	28.67	30.26	30.59	30.08	31.14
Miramar	25.92	25.70	25.41	25.02	25.82	26.99	27.14	27.62	28.22
Point Loma	33.07	32.81	32.83	32.67	33.07	33.97	34.52	35.83	36.83
Golden Gate	30.44	28.48	29.95	27.38	28.71	31.47	31.64	31.59	32.59
Everest	22.83	22.22	22.46	25.39	22.40	23.76	23.48	23.11	23.98
Terrain	20.68	20.52	20.68	18.31	20.36	21.56	22.12	21.16	22.31
Texmos1	17.66	17.37	16.76	9.32	14.72	18.79	18.80	17.24	19.22
Texmos3	17.78	17.19	16.88	9.56	14.97	18.82	18.74	17.34	19.18
Woodland Hill	25.81	25.35	25.36	22.77	25.44	27.10	27.54	27.72	28.53
Brain	27.54	26.82	26.63	23.81	27.46	28.29	28.76	28.01	29.13
Stomach	33.23	32.92	32.70	31.98	33.16	34.05	34.16	36.63	36.56

4.10 Conclusions

This chapter described the development of the rank-conditioned vector median filter (RCVMF) and rank-conditioned and threshold vector median filter (RCTVMF) from the basic vector median filter (VMF). The proposed filtering algorithms detect impulse noise prior to filtering operation and the corrupted vector pixel is replaced by the vector median. The impulse noise detection scheme in the RCVMF declares the current vector pixel as corrupted if its rank is outside a trimming range. The RCTVMF is based on the detection of impulse noise by simultaneous application of rank-conditioning and thresholding the distance of the centre vector pixel from the vector median.

As in the scalar case, simultaneous rank-conditioning and thresholding enhances impulse detection. We compared the performance of the proposed RCVMF and the RCTVMF with that of seven different vector median filters, namely, the VMF_1, the VMF_2, the MMF, the BVDF, the DDF, the CWVMF and the VSD-ROM filter on different test images. Experimental results show that the proposed RCTVMF outperforms the VMF_1, the VMF_2, the MMF, the BVDF, the DDF, the CWVMF, the VSD-ROM filter and the RCVMF in terms of the *PSNR* and visual impression in removal of fixed-valued and random-valued impulse noise while preserving image features. The performance of the RCVMF is inferior to that of the RCTVMF. Both RCVMF and RCTVMF give very stable results for different noise ratios on different types of images. The proposed filters are computationally more complex than the VMF_1 though the asymptotic complexity is of the same order with that of the VMF_1.

The RCTVMF significantly improves the performance of the vector median filter and has the potential to be the better alternative to reduce the impulse noise from a colour image.

Chapter 5

Noncausal Linear Prediction Error Based Scalar and Vector Median Filters

5.1 Introduction

Single channel linear prediction has been extensively used for modelling, estimating and coding of one-dimensional random signals, notably in speech coding and understanding [71,72], geophysics and biomedical signal processing applications [73-76]. In speech processing, the linear prediction coding (*LPC*) model is widely used for short-term modelling of voiced speech. The model is used for compression, synthesis and understanding of the speech signal [77-83] and also to clean impulse noise [84] in received data samples. The linear prediction model (*LPM*) has been extended to two-dimensional cases [85-89]. It has been used in low bit rate image compression, image segmentation and classification and spectrum estimation [90-92]. Strong correlation that exists among image pixels over two-dimensional neighbourhood makes the linear prediction model very natural for estimating pixels at any location of the image. The scalar linear prediction is restricted to use in one-dimensional signals and single channel multidimensional signals only. Wiggin and Robinson have used the modified the Levinson recursion algorithm for solving multi-channel linear predictive system of equations [93].

Majority of applications use the causal linear prediction that estimates the current value as a weighted linear combination of past values. The causal linear prediction is sometimes used on a block-by-block basis. In this case, all the data samples in an entire block are available for analysis. Hsue and Yagle have opined that a better estimate of the pixel would be expected if the linear prediction is based on both the past and future samples in the neighbourhood of the current sample [52-55].

In linear prediction, the current sample of a stationary signal is predicted from the neighbouring samples. If an impulse occurs in the data, it cannot be linearly predicted from the neighbouring data and gets reflected in the linear prediction error (*LPE*). The *LPE* should therefore contain information about the impulse noise. Thus it is worthwhile to investigate the scope of *LPE* in impulse noise detection. Such a detection scheme will deviate from the traditional ranking-based impulse-noise detection schemes.

The new contribution here is to use the causal and noncausal linear prediction error based filters for the detection of impulse noise in blocks of single and multichannel images. Computationally efficient scalar and vector median filters are proposed based on this detection mechanism.

5.2 Linear Prediction Error based Median Filtering

Consider the problem of one dimensional (causal) linear prediction. Given a sequence of observation $x(n-1), x(n-2), \ldots, x(n-M)$, the linear prediction $\hat{x}(n)$ for $x(n)$ is given by

$$\hat{x}(n) = \sum_{i=1}^{M} a(i)x(n-i)$$

where $a(i), i = 1, 2, ..., M$ are the *prediction coefficients*. $\hat{x}(n)$ is the part of $x(n)$ correlated with the neighbouring samples. The prediction error $e(n)$ is the part uncorrelated with the neighbouring data and given by

$$e(n) = x(n) - \sum_{i=1}^{M} a(i)x(n-i)$$

Any non-stationary component in the data appears in the prediction error. It has the ability to detect impulses [94], as an impulse is uncorrelated with neighbouring data. For a noise free region in the data, the prediction error should be of smaller magnitude. At the location of the impulse, this error is expected to be of higher magnitude. The straight-forward method is to threshold the prediction error to decide the presence of an impulse. However, the following problems are foreseen in the above method:

(1) The impulse at one location contributes to the prediction error at a neighbouring location and relatively large errors will appear in the samples following the impulse.

(2) If multiple impulses are present in the neighbourhood of the corrupted pixel, they will take part in linear prediction and affect the prediction error at the location of the corrupted pixel.

(3) The linear prediction coefficients are estimated from the image data. The estimated coefficients from highly corrupted data will be erroneous thus affecting the quality of linear prediction.

We propose to apply a pre-filtering stage prior to linear prediction to overcome these difficulties. The principle of the proposed linear-prediction-based median filter is illustrated in the block diagram in Figure 5.1. The pre-filter is a lowpass-filter that outputs an impulse-free image to the block-based linear predictor. The image is a locally wide-sense stationary signal and therefore linear prediction is applied block-wise. The impulse noise detector first compares the difference of the centre pixel and the predicted one. The centre pixel is declared corrupted if their difference is bigger than a pre-defined threshold. The corrupted pixel is replaced by the median filter output. The theory of the linear prediction of image is discussed in the following section.

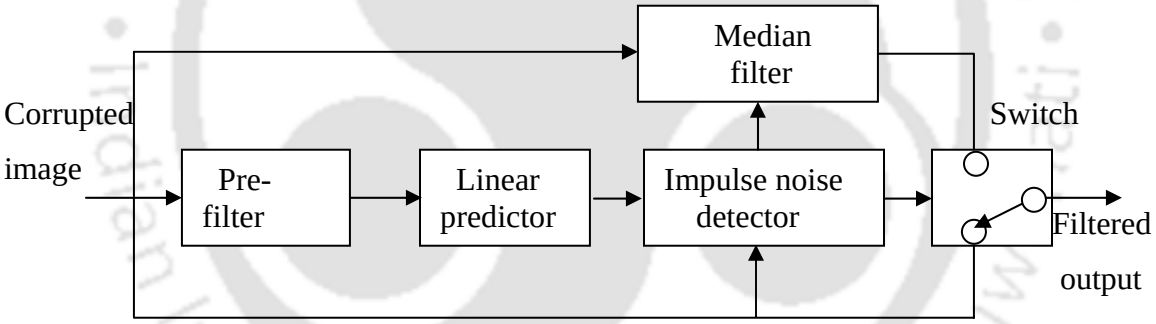


Figure 5.1 Block diagram of the linear prediction error based median filter.

5.2.1 Linear Prediction of Image

The basic linear prediction model of an image is based on the assumption that an image consists of highly correlated data and any sample in the image may be predicted as a linear combination of a few samples surrounding it. The image is modelled as a wide-sense stationary random field and the prediction coefficients are obtained from its autocorrelation functions. Both the causal and noncausal linear prediction models have been exploited for the prediction of a one-dimensional signal. However, the notion of causality cannot be applied naturally to two or higher dimensions. Line-by-line

processing technique cannot be used in the 2-D signals because, the interline dependence of the signals exists [85,89]. Two types of linear prediction models, namely the causal and the noncausal ones, are available. These models are shown in Figure 5.2.

5.2.1.1 Causal Linear Prediction

A linear predictor is said to be causal if the predicted value is a function of the only samples that arrive before it. The region of support for the 2-D causal system includes all samples except itself as shown in Figure 5.2(a). In practice, only a finite neighbourhood, called a *prediction window* is used for the prediction process.

Consider a gray-scale image. The predicted pixel $\hat{x}(m,n)$ at the location (m,n) is given by

$$\hat{x}(m,n) = \sum_{(i,j)\in W_1} \sum a(i,j)x(m-i,n-j) \tag{5.1}$$

where W_1 represents the causal prediction window that excludes $(0,0)$ and $a(i,j)$ is the prediction coefficient. Equation 5.1 is also called the nonsymmetric half-plane (*NSHP*) model. The *single-quadrant causal predictor* is a special case of this model.

5.2.1.2 Noncausal Linear Prediction

A noncausal predictor is based on the past and future samples surrounding the sample to be predicted. Thus it includes all samples in the surrounding except itself as shown in Figure 5.2(b). The predicted pixel $\hat{x}(m,n)$ at the location (m,n) is given by

$$\hat{x}(m,n) = \sum_{(i,j)\in W_2} \sum a(i,j)x(m-i,n-j) \tag{5.2}$$

where W_2 represents the noncausal prediction window that excludes $(0,0)$ and $a(i,j)$ is the noncausal prediction coefficient.

The *order* of the 2-D linear predictor is defined as illustrated in Figure 5.3 [54,95,96]. The noncausal neighbourhood points of order up to fifth are illustrated in the figure. The first-order predictor involves pixels marked as '1', the second-order set involves the

pixels marked as '1' and '2' and so on. A formal definition of the order of prediction is given in Appendix B.

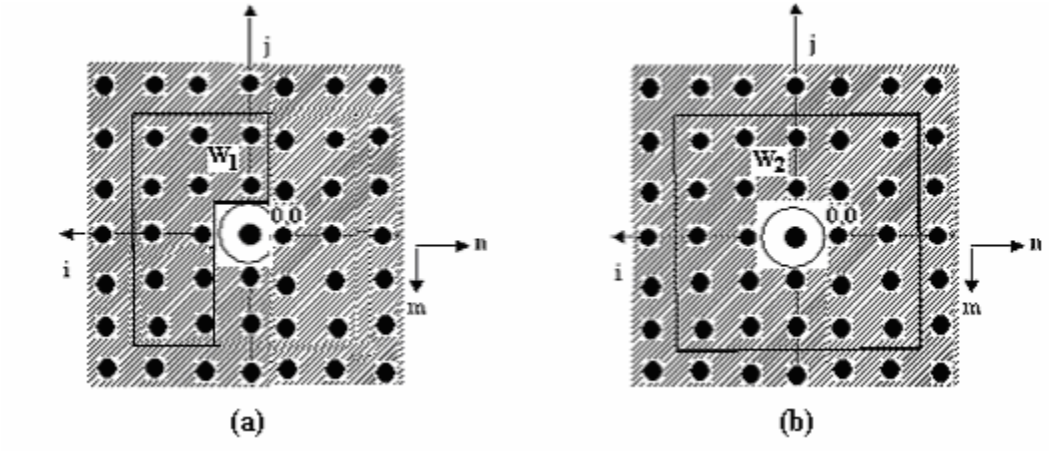


Figure 5.2 Prediction regions – (a) causal and (b) noncausal.

(5) $x(m-2,n-2)$	(4) $x(m-2,n-1)$	(3) $x(m-2,n)$	(4) $x(m-2,n+1)$	(5) $x(m-2,n+2)$
(4) $x(m-1,n-2)$	(2) $x(m-1,n-1)$	(1) $x(m-1,n)$	(2) $x(m-1,n+1)$	(4) $x(m-1,n+2)$
(3) $x(m,n-2)$	(1) $x(m,n-1)$	$x(m,n)$	(1) $x(m,n+1)$	(3) $x(m,n+2)$
(4) $x(m+1,n-2)$	(2) $x(m+1,n-1)$	(1) $x(m+1,n)$	(2) $x(m+1,n-2)$	(4) $x(m+1,n+2)$
(5) $x(m+2,n-2)$	(4) $x(m+2,n-1)$	(3) $x(m+2,n)$	(4) $x(m+2,n+1)$	(5) $x(m+2,n+2)$

Figure 5.3 Noncausal region of support for a fifth-order linear prediction.

5.2.1.3 Estimating the Prediction Coefficients

Rewriting Equation 5.2, we have

$$\begin{aligned}\hat{x}(m,n) &= \sum_{(i,j)\in W_2}\sum a(i,j)x(m-i,n-j) \\ &= \boldsymbol{x}_\eta \boldsymbol{a}_\eta\end{aligned} \tag{5.3}$$

where $\boldsymbol{x}_\eta$ is a row vector of pixels used for prediction and $\boldsymbol{a}_\eta$ is the column vector formed by the corresponding prediction coefficients. The linear prediction error (*LPE*) is given by

$$\begin{aligned}e(m,n) &= x(m,n) - \hat{x}(m,n) \\ &= x(m,n) - \boldsymbol{x}_\eta \boldsymbol{a}_\eta\end{aligned} \tag{5.4}$$

The corresponding *mean-square error* $Ee^2(m,n)$ is given by

$$\begin{aligned}Ee^2(m,n) &= E(x(m,n) - \boldsymbol{x}_\eta \boldsymbol{a}_\eta)^2 \\ &= Ex^2m,n) - 2\boldsymbol{a}_\eta^{'}E(\boldsymbol{x}_\eta^{'}x(m,n)) + \boldsymbol{a}_\eta^{'}E(\boldsymbol{x}_\eta^{'}\boldsymbol{x}_\eta)\boldsymbol{a}_\eta \\ &= R_x(0,0) - 2\boldsymbol{a}_\eta^{'}\boldsymbol{r}_{x_\eta} + \boldsymbol{a}_\eta^{'}\boldsymbol{R}_{x_\eta}\boldsymbol{a}_\eta\end{aligned} \tag{5.5}$$

where $\boldsymbol{R}_{x_\eta} = E(\boldsymbol{x}_\eta^{'}\boldsymbol{x}_\eta)$ is the autocorrelation matrix, $\boldsymbol{r}_{x_\eta} = E(x(m,n)\boldsymbol{x}_\eta^{'})$ is its cross-correlation vector with the pixel to be predicted and $R_x(0,0) = E(x^2(m,n))$.

The optimal prediction coefficient vector is obtained by minimizing $Ee^2(m,n)$ with respect to $\boldsymbol{a}_\eta$ and found by the condition

$$\nabla Ee^2(m,n) = 0 \tag{5.6}$$

where ∇ is the gradient operator with respect to the vector $\boldsymbol{a}_\eta$.

This yields in the following matrix form of *normal equations* or *Wiener filter* equations [85]:

$$\boldsymbol{R}_{\boldsymbol{x}_\eta}\boldsymbol{a}_\eta = \boldsymbol{r}_{\boldsymbol{x}_\eta} \tag{5.7}$$

The solution of the above matrix equation gives the prediction coefficient vector $\boldsymbol{a}_\eta$. Using the symmetry property $R_x(i, j) = R_x(-i, -j)$ of the autocorrelation function of wide-sense stationary image random field, it can be shown that [85]

$$a(-i, -j) = a(i, j) \tag{5.8}$$

The symmetry property simplifies the computation of the prediction coefficients. For example, the predicted value of the centre pixel $\hat{x}(m,n)$ for a second-order noncausal linear prediction filter is given by

$$\hat{x}(m,n) = \boldsymbol{x}_2\boldsymbol{a}_2 = \begin{bmatrix} x(m,n-1)+x(m,n+1) \\ x(m-1,n)+x(m+1,n) \\ x(m-1,n-1)+x(m+1,n+1) \\ x(m-1,n+1)+x(m+1,n-1) \end{bmatrix}' \begin{bmatrix} a(0,1) \\ a(1,0) \\ a(1,1) \\ a(1,-1) \end{bmatrix} \tag{5.9}$$

and the prediction coefficients obtained from Equation 5.7

$$\boldsymbol{R}_{\boldsymbol{x}_2}\boldsymbol{a}_2 = \boldsymbol{r}_{\boldsymbol{x}_2} \tag{5.10}$$

where

$$\boldsymbol{R}_{\boldsymbol{x}_2} = \begin{bmatrix} R_x(0,0)+R_x(0,2) & R_x(1,-1)+R_x(1,1) & R_x(1,0)+R_x(1,2) & R_x(1,-2)+R_x(1,0) \\ R_x(1,-1)+R_x(1,1) & R_x(0,0)+R_x(2,0) & R_x(0,1)+R_x(2,1) & R_x(0,1)+R_x(2,-1) \\ R_x(1,0)+R_x(1,2) & R_x(0,1)+R_x(2,1) & R_x(0,0)+R_x(2,2) & R_x(0,2)+R_x(2,0) \\ R_x(1,-2)+R_x(1,0) & R_x(0,1)+R_x(2,-1) & R_x(0,2)+R_x(2,0) & R_x(0,0)+R_x(2,-2) \end{bmatrix}$$

and

$$\boldsymbol{r}_{\boldsymbol{x}_2} = \begin{bmatrix} R_x(0,1) \\ R_x(1,0) \\ R_x(1,1) \\ R_x(1,-1) \end{bmatrix}$$

For numerical implementation, the autocorrelation functions in Equation 5.7 are replaced by the estimated autocorrelation functions in so called autocorrelation method [53]. The estimated autocorrelation function $\hat{R}_x(m,n)$, estimated from the pixels in a block of $P \times Q$ size is given by

$$\hat{R}_x(m,n) = \frac{1}{PQ} \sum_{i=1}^{P-|m|} \sum_{j=1}^{Q-|n|} x(i,j)x(i+|m|, j+|n|)$$

5.3 Multichannel Linear Prediction

The causal and noncausal models for two-dimensional linear prediction have been extended to multichannel images. For multichannel images, the linear predictor can be classified into three types: *intrachannel predictor, constrained intrachannel predictor and interchannel predictor* [97]. In the *intrachannel predictor*, a pixel in one channel is predicted based on the other pixels in the same channel. A special case of the intrachannel predictor is the *constrained intrachannel predictor* which uses the same prediction coefficients for different channels. Both methods do not exploit the interchannel correlation. The *interchannel predictor* is the only true vector linear predictor. It exploits the correlation among the vector pixels.

Consider an analysis block consisting of $P \times Q$ vector pixels for a C-channel image and let $\boldsymbol{x}(m,n) = [x_1(m,n)\ x_2(m,n)\ ...\ x_C(m,n)]'$ be the vector pixel at the centre (m,n) of window. Let W_2 be the noncausal region of support for the linear predictor. The prediction signal using the interchannel predictor given by

$$\hat{\boldsymbol{x}}(m,n) = \sum_{(i,j)\in W_2} \sum \mathbf{a}(i,j)\boldsymbol{x}(m-i,n-j) \qquad (5.11)$$

where $\mathbf{a}(i,j)$ is a $C \times C$ matrix. The elements of $\mathbf{a}(i,j)$ can be obtained minimizing $\boldsymbol{E}\left\|x(m,n) - \hat{\boldsymbol{x}}(m,n)\right\|^2$ and solving the corresponding matrix Wiener filter equation

involving autocorrelation between pixels of the same channel and cross-correlation between pixels in different channels [97].

If the interchannel correlation is not considered and a pixel in a channel is predicted on the basis of pixels in the same channel, we have the simpler *intrachannel predictor*. The intrachannel predictor is given by

$$\hat{x}_k(m,n) = \sum_{(i,j)\in W_2} \sum a_k(i,j) x_k(m-i,n-j) \quad k = 1,2...,C. \tag{5.12}$$

The prediction coefficients for the intrachannel predictor can be obtained by considering the scalar linear prediction for each channel separately.

The prediction model is further simplified when the same predictor is used for all the channels. The resulting *constrained intrachannel predictor* can be written as

$$\begin{aligned} \hat{\boldsymbol{x}}(m,n) &= \sum_{(i,j)\in W_2} \sum a(i,j)\boldsymbol{x}(m-i,n-j) \\ &= \boldsymbol{X}_\eta \boldsymbol{a}_\eta \end{aligned} \tag{5.13}$$

where $\boldsymbol{a}_\eta$ is the vector formed by the prediction coefficients, $\boldsymbol{X}_\eta$ is a matrix of C rows and M columns, M is the number of vector pixels used for prediction and η is the order of prediction.

Consider an RGB colour image with $\boldsymbol{x}(m,n) = [x_R(m,n)\ x_G(m,n)\ x_B(m,n)]'$ as an example. For a second-order prediction, $\boldsymbol{X}_2$ and $\boldsymbol{a}_2$ are given by

$$\boldsymbol{X}_2 = \begin{bmatrix} x_R(m,n-1) & x_G(m,n-1) & x_B(m,n-1) \\ x_R(m-1,n) & x_G(m-1,n) & x_B(m-1,n) \\ x_R(m-1,n-1) & x_G(m-1,n-1) & x_B(m-1,n-1) \\ x_R(m-1,n+1) & x_G(m-1,n+1) & x_B(m-1,n+1) \\ x_R(m,n+1) & x_G(m,n+1) & x_B(m,n+1) \\ x_R(m+1,n) & x_G(m+1,n) & x_B(m+1,n) \\ x_R(m+1,n+1) & x_G(m+1,n+1) & x_B(m+1,n+1) \\ x_R(m+1,n-1) & x_G(m+1,n-1) & x_B(m+1,n-1) \end{bmatrix}'$$

and $\boldsymbol{a}_2 = \left[a(0,1)\ \ a(1,0)\ a(1,1)\ a(1,-1)\ a(0,-1)\ a(-1,0)\ \ a(-1,-1)\ a(-1,1)\right]'$

The interchannel predictor is computationally more complex compared to the intrachannel counterpart. It has been shown in [98] that the interchannel predictor does not provide much prediction gain over the two varieties of intrachannel predictors in the case of the causal prediction. Considering the computational simplicity and without loss of generality, *the thesis uses the constrained intrachannel prediction model for both causal and noncausal cases*.

5.3.1 Constrained Intrachannel Noncausal Linear Prediction

Consider the constrained intrachannel predictor in Equation 5.13.

The error between the vector pixel $\boldsymbol{x}(m,n)$ and the linear prediction $\hat{\boldsymbol{x}}(m,n)$ can be written as

$$\begin{aligned} \boldsymbol{e}(m,n) &= \boldsymbol{x}(m,n) - \hat{\boldsymbol{x}}(m,n) \\ &= \boldsymbol{x}(m,n) - \boldsymbol{X}_\eta \boldsymbol{a}_\eta \end{aligned} \tag{5.14}$$

For a given vector pixel $\boldsymbol{x}(m,n)$ and the predicted vector pixel $\hat{\boldsymbol{x}}(m,n)$, the prediction error vector $\boldsymbol{e}(m,n)$ depends on the filter coefficient vector $\boldsymbol{a}_\eta$. The optimum two-dimensional linear prediction coefficients can be obtained by minimizing the mean-

square Euclidean norm of the error vector. The *mean-square Euclidean norm of the error vector* is given by

$$\begin{aligned} S &= E \,||\, \boldsymbol{e}(m,n) \,||^2 \\ &= E \,||\, \boldsymbol{x}(m,n) - \hat{\boldsymbol{x}}(m,n) \,||^2 \\ &= E \,||\, \boldsymbol{x}(m,n) - \boldsymbol{X}_\eta \boldsymbol{a}_\eta \,||^2 \\ &= E\boldsymbol{x}^{'}(m,n)\boldsymbol{x}(m,n) - 2\boldsymbol{a}_\eta^{'} E(\boldsymbol{X}_\eta^{'}\boldsymbol{x}(m,n)) + \boldsymbol{a}_\eta^{'} E(\boldsymbol{X}_\eta^{'}\boldsymbol{X}_\eta)\boldsymbol{a}_\eta \\ &= R_X(0,0) - 2\boldsymbol{a}_\eta^{'}\boldsymbol{r}_{X_\eta} + \boldsymbol{a}_\eta^{'}\boldsymbol{R}_{X_\eta}\boldsymbol{a}_\eta \end{aligned} \tag{5.15}$$

where $\boldsymbol{R}_{X_\eta} = E(\boldsymbol{X}_\eta^{'}\boldsymbol{X}_\eta)$ is the autocorrelation matrix of the input vectors, $\boldsymbol{r}_{X_\eta} = E(\boldsymbol{X}_\eta^{'}\boldsymbol{x}(m,n))$ is the cross-correlation of the input vectors and the vector pixel to be predicted and $R_X(0,0) = E(\boldsymbol{x}^{'}(m,n)\boldsymbol{x}(m,n))$.

Similar to the scalar case, the optimal prediction coefficient vector is found by the condition

$$\nabla E \,||\, \boldsymbol{e}(m,n) \,||^2 = 0 \tag{5.16}$$

The corresponding matrix form of the Wiener filter equations is given by

$$\boldsymbol{R}_{X_\eta}\boldsymbol{a}_\eta = \boldsymbol{r}_{X_\eta} \tag{5.17}$$

For an RGB colour image, suppose $\boldsymbol{x}(m,n) = [x_R(m,n)\; x_G(m,n)\; x_B(m,n)]'$ and $\boldsymbol{X}_\eta$ is partitioned as

$$\boldsymbol{X}_\eta = \begin{bmatrix} \boldsymbol{X}_{R_\eta} \\ ---- \\ \boldsymbol{X}_{G_\eta} \\ ---- \\ \boldsymbol{X}_{B_\eta} \end{bmatrix}$$

Then,

$$\boldsymbol{R}_{X_\eta} = \boldsymbol{R}_{R_\eta} + \boldsymbol{R}_{G_\eta} + \boldsymbol{R}_{B_\eta}$$
$$= E(\boldsymbol{X}_{R\eta}^{'}\boldsymbol{X}_{R\eta}) + E(\boldsymbol{X}_{G\eta}^{'}\boldsymbol{X}_{G\eta}) + E(\boldsymbol{X}_{B\eta}^{'}\boldsymbol{X}_{B\eta})$$

and

$$\boldsymbol{r}_{X_\eta} = \boldsymbol{r}_{R_\eta} + \boldsymbol{r}_{G_\eta} + \boldsymbol{r}_{B_\eta} = E(\boldsymbol{X}_{R\eta}^{'}x_R(m,n)) + E(\boldsymbol{X}_{G\eta}^{'}x_G(m,n)) + E(\boldsymbol{X}_{B\eta}^{'}x_B(m,n)).$$

Equation 5.17 is to be solved to find $\boldsymbol{a}_\eta$. We consider large blocks for linear prediction for the following reasons:

(1) Larger block-size means less number of predictors and involves less computation for finding the prediction coefficients.

(2) An autocorrelation function estimated from a smaller block of data is more erroneous as they involve less data.

As in the case of 2-D scalar noncausal linear prediction [85], it can be shown that $a(i,j) = a(-i,-j)$ for the constrained intrachannel noncausal predictor. For example, the predicted vector pixel $\hat{\boldsymbol{x}}(m,n)$ at the location (m,n) for a second-order noncausal linear prediction error based filter for the RGB colour image is given by

$$\hat{\boldsymbol{x}}(m,n) = \boldsymbol{X}_2\boldsymbol{a}_2 \tag{5.18}$$

where

$$\boldsymbol{X}_2 = \begin{bmatrix} x_R(m,n-1)+x_R(m,n+1) & x_G(m,n-1)+x_G(m,n+1) & x_B(m,n-1)+x_B(m,n+1) \\ x_R(m-1,n)+x_R(m+1,n) & x_G(m-1,n)+x_G(m+1,n) & x_B(m-1,n)+x_B(m+1,n) \\ x_R(m-1,n-1)+x_R(m+1,n+1) & x_G(m-1,n-1)+x_R(m+1,n+1) & x_B(m-1,n-1)+x_B(m+1,n+1) \\ x_R(m-1,n+1)+x_R(m+1,n-1) & x_G(m-1,n+1)+x_G(m+1,n-1) & x_B(m-1,n+1)+x_B(m+1,n-1) \end{bmatrix}^{'}$$

and

$$\boldsymbol{a}_2 = \begin{bmatrix} a(0,1) \\ a(1,0) \\ a(1,1) \\ a(1,-1) \end{bmatrix}$$

The prediction coefficients are given by

$$\boldsymbol{R}_{X_2}\boldsymbol{a}_2 = \boldsymbol{r}_{X_2} \tag{5.19}$$

where

$$\boldsymbol{R}_{X_2} = \begin{bmatrix} R_X(0,0)+R_X(0,2) & R_X(1,-1)+R_X(1,1) & R_X(1,0)+R_X(1,2) & R_X(1,-2)+R_X(1,0) \\ R_X(1,-1)+R_X(1,1) & R_X(0,0)+R_X(2,0) & R_X(0,1)+R_X(2,1) & R_X(0,1)+R_X(2,-1) \\ R_X(1,0)+R_X(1,2) & R_X(0,1)+R_X(2,1) & R_X(0,0)+R_X(2,2) & R_X(0,2)+R_X(2,0) \\ R_X(1,-2)+R_X(1,0) & R_X(0,1)+R_X(2,-1) & R_X(0,2)+R_X(2,0) & R_X(0,0)+R_X(2,-2) \end{bmatrix}$$

$$\boldsymbol{r}_{X_2} = \begin{bmatrix} R_X(0,1) \\ R_X(1,0) \\ R_X(1,1) \\ R_X(1,-1) \end{bmatrix} \text{ and}$$

$$R_X(m,n) = R_{X_R}(m,n) + R_{X_G}(m,n) + R_{X_B}(m,n).$$

Similar formulations can be derived for causal linear prediction. The derivation of the normal equations for first-order, second-order causal and noncausal linear prediction of gray-scale and colour images is shown in Appendix B.

5.4 Noncausal Linear Prediction Error based Median Filter for Gray-scale Images

Once the prediction has been made, the prediction error is used to decide whether a pixel is corrupted or not. The centre pixel $x(m,n)$ is replaced by the median value of all the pixels inside the window if the absolute difference between the original value and predicted one is bigger than a pre-defined threshold θ. We will use second-order noncausal linear prediction for illustration of the algorithm. The *second-order noncausal linear prediction error based standard median filter* (2-NCLPF$_{SM}$) that replaces the corrupted centre pixel by the median is given by

$$x_{2-\text{NCLPF}_{\text{SM}}}(m,n) = \begin{cases} x_{\text{SM}} & \text{if } |x(m,n)-\hat{x}(m,n)| > \theta \\ x(m,n) & \text{otherwise} \end{cases} \tag{5.20}$$

where $x_{SM} = \text{median}\{ x(m,n),\ x(m,n-1),\ x(m,n+1),\ x(m-1,n),\ x(m+1,n),\ x(m-1,n-1),\ x(m+1,n+1),\ x(m-1,n+1),\ x(m+1,n-1) \}$.

For the computational efficiency, the noncausal linear prediction error based median filter for gray-scale images can be implemented by the following simplified scheme, shown in Figure 5.4. In this scheme, the image is pre-filtered by a median filter and the median filter output is used for the linear prediction. If the prediction error is not greater than a pre-defined threshold, the median output is retained, else it is replaced by the original pixel value.

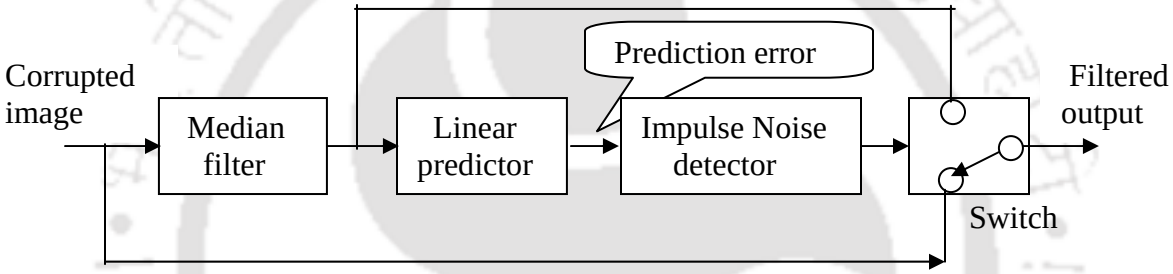


Figure 5.4 Block diagram of the simplified linear prediction error based median filter for gray-scale images.

The standard median filter used in the filtering stage of the linear prediction error based median filtering can be replaced by other suitable filters like the rank ordered mean (ROM) filter. The subscript *ROM* is used to denote ROM-based filtering.

5.5 Noncausal Linear Prediction Error based Vector Median Filter for Colour Images

The centre vector pixel inside the window is decided to be impulse noise and replaced by the vector median $\boldsymbol{x}_{VMF}(m,n)$, if the norm of the prediction error vector is bigger than a pre-defined threshold θ. We will again use the second-order noncausal linear prediction for illustration of the algorithm. Thus the output of the *second-order noncausal linear prediction error based vector median filter* (2-NCLPVMF) is given by

$$\boldsymbol{x}_{2-\text{NCLPVMF}}(m,n)=\begin{cases}\boldsymbol{x}_{\text{VMF}} & \text{if } ||e(m,n)||>\theta \\ \boldsymbol{x}(m,n) & \text{otherwise}\end{cases} \tag{5.21}$$

For determining the norm of the prediction-error vector, we choose the L_∞ norm so that

$$||e(m,n)||=max(|x_R(m,n)-\hat{x}_R(m,n)|,|x_G(m,n)-\hat{x}_G(m,n)|,|x_B(m,n)-\hat{x}_B(m,n)|).$$

The noncausal linear prediction error based vector median filter for colour images can implemented by the following simplified scheme, shown in Figure 5.5. The marginal median filter (MMF) is used for pre-filtering for computational efficiency. The VMF is applied only on the corrupted pixels.

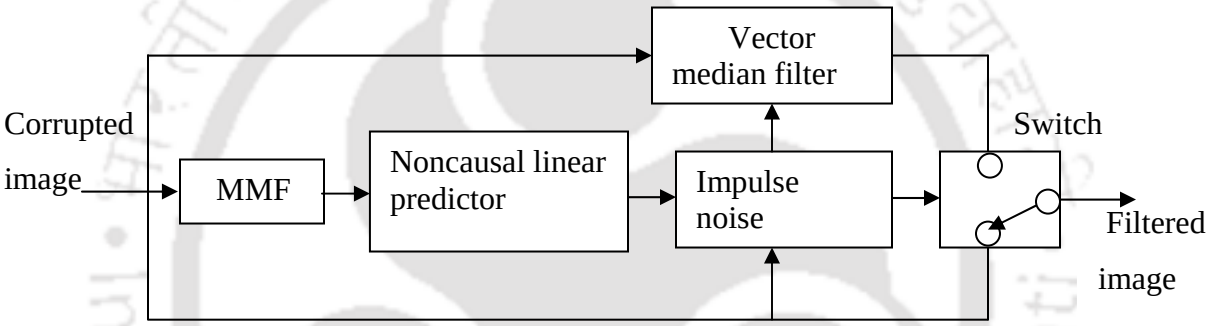


Figure 5.5 Block diagram of the noncausal linear prediction error based vector median filter for colour images.

5.6 Computational Complexity Analysis

It is interesting to study the computational efficiency of the proposed algorithms. The number of operations required in different types of linear-prediction based median filters is obtained in the following subsections. The number of operations obtained by this analysis is not optimal in the sense that it does not consider the optimal cost in different steps.

5.6.1 Computational Complexity of the Linear Prediction Error based Scalar Median Filter

We consider the second-order noncausal linear-prediction based median filter. We exclude the pre-filtering stage in this calculation. The detection stage involves the following computation:

(1) As obvious from Equation 5.10, the 4×4 matrix $\boldsymbol{R}_{x_2}$ includes 13 distinct autocorrelation functions. The computation of autocorrelation functions from the pixels in the block of $P \times Q$ size requires $13PQ - 15P - 15Q + 5$ additions and $13PQ - 15P - 15Q + 18$ multiplications. Because of the large block size, the number of blocks used in the linear prediction error based filter is comparatively very less as compared to the total number of pixels in the image.

(2) 10 additions are required to obtain the elements of $\boldsymbol{R}_{x_2}$ from the computed autocorrelation functions.

(3) The computation of $\boldsymbol{a}_2 = \boldsymbol{R}_{x_2}^{-1} \boldsymbol{r}_{x_2}$ requires 26 additions/subtractions and 36 multiplications/divisions by the Gaussian elimination method.

(4) The calculation of the second-order noncausal predicted pixel $\hat{x}(m,n)$ requires $7PQ$ addition/subtraction, $4PQ$ multiplication operations.

(5) The computation of $|e(m,n)| > \theta$ requires $2PQ$ comparison and PQ addition/subtraction operations.

Therefore, detection of impulse noise in an image block of $P \times Q$ size by the second-order noncausal linear predictor involves $21PQ - 15P - 15Q + 41$ additions, $17PQ - 15P - 15Q + 54$ multiplications and $2PQ$ comparisons. The number of operations involved in 2-CLPF can be similarly calculated. Table 5.1 gives the total number of operations used in detection of corrupted pixels in a block and computational complexity. For a $n \times n$ filtering window, the median filtering stage involves O(n) order of comparisons per pixel assuming the fast window histogram based algorithm [65]. Including the operations involved in impulse detection, the asymptotic complexity of the linear prediction error based median filter is summarised in Table 5.1 (c).

Table 5.1(a) Total number of operations for the detection of impulse noise in gray-scale images for a $P \times Q$ block size

Filters	Additions/Subtractions	Multiplications	Comparisons
2-NCLPF	$21PQ - 15P - 15Q + 41$	$17PQ - 15P - 15Q + 54$	$2PQ$
2-CLPF	$11PQ - 7P - 4Q + 44$	$11PQ - 7P - 4Q + 59$	$2PQ$

Table 5.1(b) Total number of operations per pixel for the detection of impulse noise for a 32×32 block size

Filters	Additions/Subtractions	Multiplications	Comparisons
2-NCLPF	20.10	16.12	2
2-CLPF	10.70	10.71	2

Table 5.1(c) Computational complexity of linear prediction error based median filter

Filters	Additions	Multiplications	Comparisons
2-NCLPF	O(1)	O(1)	O(n)

5.6.2 Computational Complexity of the Linear Prediction Error based Vector Median Filter

We consider the second-order noncausal linear-prediction based vector median filter. Excluding the MMF computation, the detection stage involves the following computation:

(1) The 4×4 autocorrelation matrix $\boldsymbol{R}_{X_2}$ in Equation 5.19 includes 13 distinct autocorrelation functions. Each of these autocorrelation functions is a sum of the corresponding autocorrelation functions in 3 channels. The computation of these elements from the pixels in the block of $P \times Q$ size requires $3 \times (13PQ - 15P - 15Q + 5)$ additions and $3 \times (13PQ - 15P - 15Q + 18)$ multiplications.

(2) It requires 10 additions to form $\boldsymbol{R}_{X_2}$ like in the scalar case.

(4) The computation of $\boldsymbol{a}_2 = \boldsymbol{R}_{X_2}^{-1} \boldsymbol{r}_{X_2}$ requires 26 additions/subtractions and 36 multiplications/divisions by the Gaussian elimination method.

(5) The calculation of the second-order noncausal predicted vector pixel $\hat{\boldsymbol{x}}(m,n)$ requires $3 \times 7PQ$ addition/subtraction and $3 \times 4PQ$ multiplication operations.

(6) The computation of $\| e(m,n) \| > \theta$ requires $3 \times 2PQ$ comparison and $3 \times PQ$ addition/subtraction operations.

Therefore, detection of impulse noise in a colour image block of $P \times Q$ size by the second-order noncausal linear predictor involves $63PQ - 45P - 45Q + 51$ additions, $51PQ - 45P - 45Q + 90$ multiplications and $6PQ$ comparisons. Similar calculation can be done for impulse detection by the 2-CLPVMF. Table 5.2 lists the number of operations involved in detection of impulse in a block of $P \times Q$ size. Assuming perfect detection, the replacement of the corrupted vector pixel requires $p(5(n(n^2-n)+n(n-1)/2)+2n(n^2-n)+n^3-n)$ additions/subtractions and $p(3(n(n^2-n)+n(n-1)/2))+n^2-1)$ comparisons per vector pixel, where p is the impulse noise ratio, using a fast running algorithm [94]. For a $n \times n$ filtering window, the MMF involves $3\mathrm{O}(n)$ comparisons. The MMF pre-filtering steps will not change the order of the complexity of compare operations. While additional operations are required to compute the prediction coefficients and the prediction error at each pixel location, they do not add to the order of asymptotic complexity of the vector median filter. Table 5.2(b) summarises the total number of operations per vector pixel for the detection of impulse noise in an RGB colour images using a 32×32 block size for block-based linear prediction. The asymptotic complexities of the proposed vector median filter is summarised in Table 5.2(c).

Table 5.2(a) Total number of operations for the detection of impulse noise in colour images for a $P \times Q$ block size for 3 channels

Filters	**Number of operations in a $P \times Q$ block for 3 channels**		
	Additions/Subtractions	**Multiplications**	**Comparisons**
2-NCLPVMF	$63PQ - 45P - 45Q + 91$	$51PQ - 45P - 45Q + 90$	$6PQ$
2-CLPVMF	$33PQ - 21P - 12Q + 64$	$33PQ - 21P - 12Q + 67$	$6PQ$

Table 5.2(b) Summary of total number of operations per vector pixel for the detection of impulse noise from colour images in a 32×32 block for 3 channels

Filters	**Number of operations/vector pixel in a 32×32 block for 3 channels**		
	Additions/Subtractions	**Multiplications**	**Comparisons**
2-NCLPVMF	60.27	48.28	6
2-CLPVMF	32.03	32.03	6

Table 5.2(c) Computational complexity of linear prediction error based VMF_1

Filters	**Additions**	**Multiplications**	**Comparisons**
2-NCLPVMF	$O(n^3)$	O(1)	$O(n^3)$
2-CLPVMF	$O(n^3)$	O(1)	$O(n^3)$

5.7 Experimental Results

5.7.1 Results of Noncausal Linear Prediction Error based Median Filters for Gray-scale Images

The 8-bit gray-scale images of the Lena, Mandrill, Airplane, Lake, Bridge, Goldhill, Miramar, Point Loma, Golden Gate, Terrain, Texmos1 and Texmos3 were considered for the experimentation. All images were of 512×512 size. Fixed-valued and random-valued impulses at different noise ratios were artificially injected in these images. Fixed-valued impulse noise was generated by two-sided impulse noise model and random-valued impulse noise was generated by Bernoulli-uniform impulse noise model.

A block size of 32 × 32 of the image was used for estimating the autocorrelation functions. In the case of pixels lying near the boundary between two adjacent blocks, a prediction window overlapping the adjacent blocks is used. In the case of pixels at the image boundary, samples are symmetrically appended for applying causal and noncausal prediction. Following experiments were performed to study the performance of the proposed linear- prediction based median filters.

Experiment 5.7.1.1: Selection of the Order of Predictors

The first set of experiments was performed to decide the order of linear prediction that gives the best results in the term of *PSNR* values. Experimental results on the Lena and Mandrill images corrupted with impulse noise at 10% to 40% ratios are reported in Table 5.3. Following linear-prediction based filters were used for comparison.

(1) First-order causal linear-prediction error based standard median filter (1-$CLPF_{SM}$)

(2) First-order noncausal linear-prediction error based standard median filter (1-$NCLPF_{SM}$)

(3) First-order causal linear-prediction error based ROM filter (1-$CLPF_{ROM}$)

(4) First-order noncausal linear-prediction error based ROM filter (1-$NCLPF_{ROM}$)

(5) Second-order causal linear-prediction error based standard median filter (2-$CLPF_{SM}$)

(6) Second-order noncausal linear-prediction error based standard median filter (2-$NCLPF_{SM}$)

(7) Second-order causal linear-prediction error based ROM filter (2-$CLPF_{ROM}$)

(8) Second-order noncausal linear-prediction error based ROM filter (2-$NCLPF_{ROM}$)

(9) Third-order causal linear-prediction error based standard median filter (3-$CLPF_{SM}$)

(10) Third-order noncausal linear-prediction error based standard median filter (3-$NCLPF_{SM}$)

(11) Third -order causal linear-prediction error based ROM filter (3-$CLPF_{ROM}$)

(12) Third-order noncausal linear-prediction error based ROM filter (3-$NCLPF_{ROM}$)

In case of the Lena image, the threshold for the prediction error was fixed at 35 for the fixed-valued impulse noise and at 20 for the random-valued impulse noise. Similarly, 65 and 50 were used as thresholds for the fixed-valued and random-valued impulse noise respectively in case of the Mandrill image. These thresholds were determined by trial and error.

Table 5.3(a) Comparison of *PSNR* of the Lena image restored by linear prediction error based median filters

Type of filters	Percentage of fixed-valued impulse noise				Percentage of random-valued impulse noise			
	10	20	30	40	10	20	30	40
1-CLPF$_{SM}$	38.41	34.59	31.93	29.99	34.93	32.03	31.14	28.01
1-NCLPF$_{SM}$	38.95	35.39	32.80	30.33	35.86	33.30	30.80	28.74
1-CLPF$_{ROM}$	37.43	34.19	31.74	29.53	34.24	31.61	29.99	27.94
1-NCLPF$_{ROM}$	38.82	35.70	33.13	30.02	35.42	33.07	30.92	28.61
2-CLPF$_{SM}$	38.36	34.54	32.13	29.72	35.51	32.66	30.60	28.44
2-NCLPF$_{SM}$	39.05	35.55	32.64	29.95	36.37	33.54	30.94	28.77
2-CLPF$_{ROM}$	37.04	34.64	32.38	29.90	36.32	33.52	31.51	29.57
2-NCLPF$_{ROM}$	39.62	36.03	32.88	29.71	37.64	34.65	32.47	30.17
3-CLPF$_{SM}$	38.04	34.29	31.12	28.88	36.62	33.75	31.21	28.76
3-NCLPF$_{SM}$	39.25	35.39	32.27	29.00	37.91	34.68	32.07	29.34
3-CLPF$_{ROM}$	37.31	34.50	31.54	29.34	36.65	33.07	30.80	29.05
3-NCLPF$_{ROM}$	39.29	36.07	32.68	29.39	37.78	34.66	32.28	29.41

Table 5.3(b) Comparison of *PSNR* of the Mandrill image restored by linear-prediction error based median filters

Type of filter	Percentage of fixed-valued impulse noise				Percentage of random-valued impulse noise			
	10	20	30	40	10	20	30	40
1-CLPF$_{SM}$	25.72	23.52	22.02	19.79	23.98	22.54	21.44	20.36
1-NCLPF$_{SM}$	26.06	23.99	22.11	20.13	24.20	22.75	21.65	20.57
1-CLPF$_{ROM}$	25.56	23.60	22.13	19.88	23.67	22.34	21.31	20.30
1-NCLPF$_{ROM}$	25.97	24.05	22.47	20.05	23.92	22.65	21.56	20.57
2-CLPF$_{SM}$	25.48	23.71	22.23	20.42	24.12	22.59	21.57	20.40
2-NCLPF$_{SM}$	25.98	23.97	22.16	20.03	24.45	23.01	21.83	20.58
2-CLPF$_{ROM}$	25.49	23.78	22.29	20.47	23.85	22.40	21.42	20.35
2-NCLPF$_{ROM}$	25.86	24.15	22.50	20.48	24.52	23.11	21.90	20.60
3-CLPF$_{SM}$	24.84	23.03	21.68	20.63	22.55	20.58	17.88	12.89
3-NCLPF$_{SM}$	25.95	23.74	22.11	19.74	24.55	21.76	17.20	11.73
3-CLPF$_{ROM}$	24.32	23.07	21.56	20.41	23.85	21.13	17.44	11.85
3-NCLPF$_{ROM}$	25.94	24.05	22.09	19.81	24.33	21.74	17.19	11.87

Following observations were made from these two tables:

(1) The most appropriate order of linear prediction is 2. With third-order prediction, the *PSNR* performance reduced more in the case of the random-valued impulse noise.

(2) Noncausal linear prediction error based filters gave slightly better *PSNR* results in comparison with filter based on causal prediction.

(3) For replacing the corrupted pixels, the rank-ordered mean gave better alternative to the median filter.

On the basis of these observations, the second-order noncausal linear predictor was selected for the rest of the experiments.

Experiment 5.7.1.2: Selection of Thresholds for the Prediction Error

The next set of experiment was conducted to find out an appropriate threshold of the prediction error to decide about the impulse. Experiments conducted on the Lena and Mandrill images at impulse noise ratios from 5% to 40% for both types of impulse noise are reported in Figure 5.6. It is seen that the optimum threshold reduces with increased percentage of noise. The figure suggests that the appropriate values of thresholds are near 35 and 20 for fixed-valued and random-valued impulse noise respectively in case of the Lena image. Similarly, the appropriate thresholds are near 65 and 50 for fixed-valued and random-valued impulse noise respectively for the Mandrill image. Note that a smaller value of the threshold results in filtering of more number of pixels and hence more blurring in the filter outputs. On the other hand, a bigger value means that many impulses are left undetected. Experimental results on different images suggest that the appropriate value of the threshold lies between 30 and 75 for fixed-valued impulse noise and between 15 and 60 for random-valued impulse noise. These ranges are wide, but nearly independent of the order of prediction.

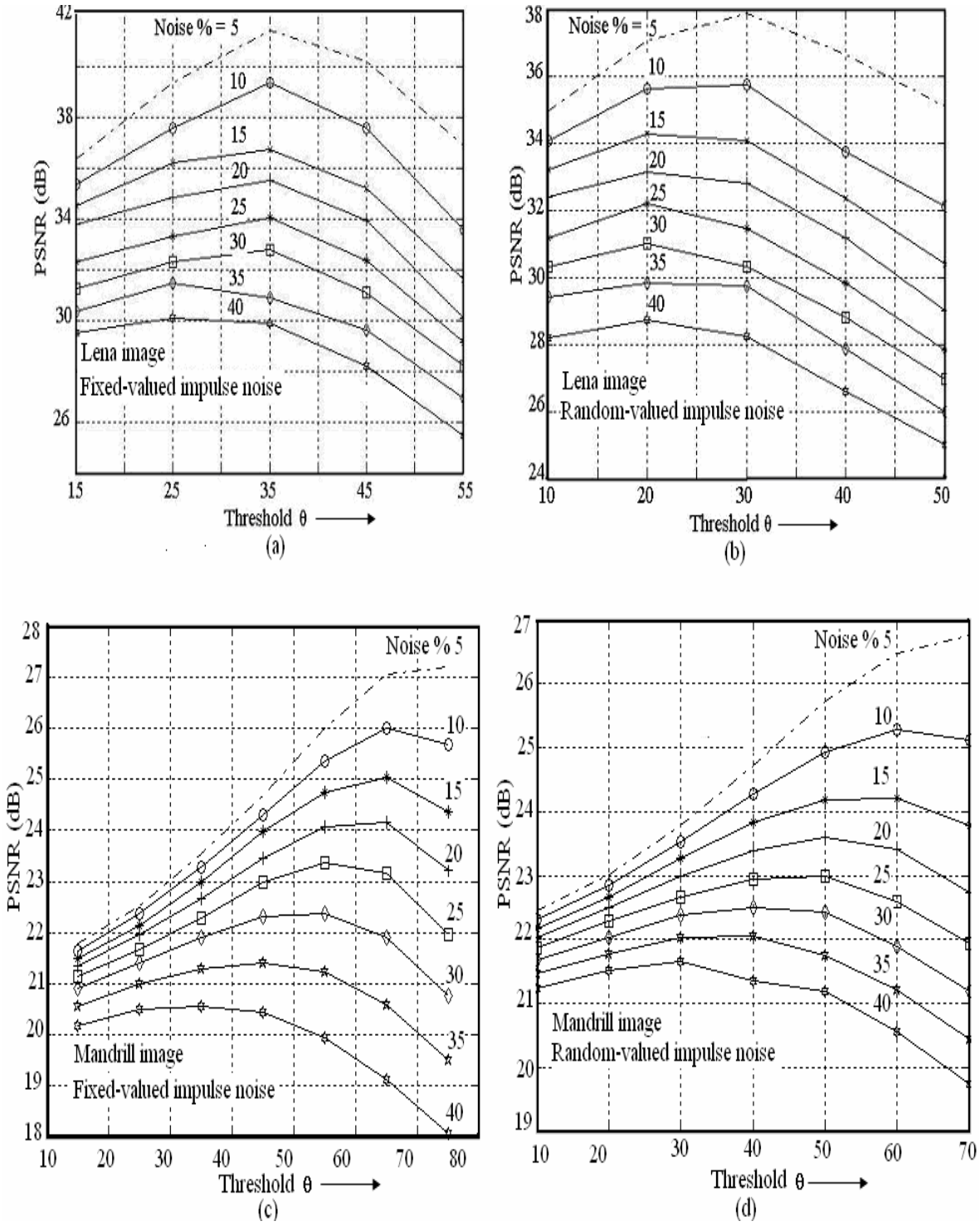


Figure 5.6 Threshold - *PSNR* curves at different percentages of impulse noise: (a) and (b) for the Lena image; (c) and (d) for the Mandrill image.

Experiment 5.7.1.3: Comparison with other Median Filters at Different Noise Ratios

The next set of experiments was performed for the comparison of the proposed filters with some of the important median filters. In all cases, a window of 3×3 size was used. The following filters were used for comparison with the proposed 2-NCLPF$_{SM}$ and 2-NCLPF$_{ROM}$ filters:

(1) Standard median (SM) filter
(2) Centre-weighted median (CWM) filter
(3) Rank-conditioned median (RCM) filter
(4) Tri-state median (TSM) filter
(5) Signal-dependent rank-ordered mean (SD-ROM) filter
(6) Chen and Wu's adaptive centre-weighted (Chen and Wu) median filter

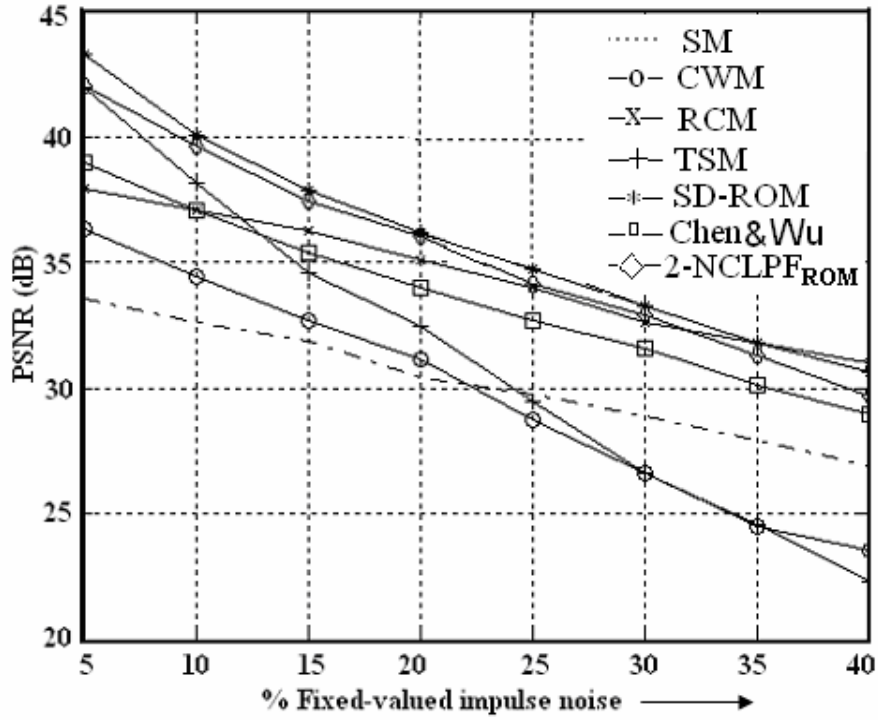
Results of experiments on the Lena image are tabulated in Table 5.4 (a) and Table 5.4 (b) for fixed-valued and random-valued impulse noise respectively, with the noise ratios ranging from 5% to 40%. The same thresholds of 20 for random-valued impulse noise and 35 for fixed-valued impulse noise were used in all observations. It has been observed from Table 5.4 (a) that the performance of the second-order noncausal linear prediction error based on rank-ordered mean (2-NCLPF$_{ROM}$) filter was better than those of the SM, CWM, RCM, TSM and Chen & Wu's adaptive centre-weighted median filters, and slightly inferior to the SD-ROM filter in removal of the fixed-valued impulse noise, but it outperformed all filters under consideration in removal of the random-valued impulse noise. It is also observed that the performance of the 2-NCLPF$_{ROM}$ without the pre-filter is inferior to the one with the pre-filter. Data from these tables were plotted graphically in Figure 5.7. The better performance of the proposed 2-NCLPF$_{ROM}$ in removal of the random-valued impulse noise has been observed from the plot.

Table 5.4(a) Comparison of *PSNR* of the Lena image corrupted with 5% to 40% of fixed-valued impulse noise and restored by different filters

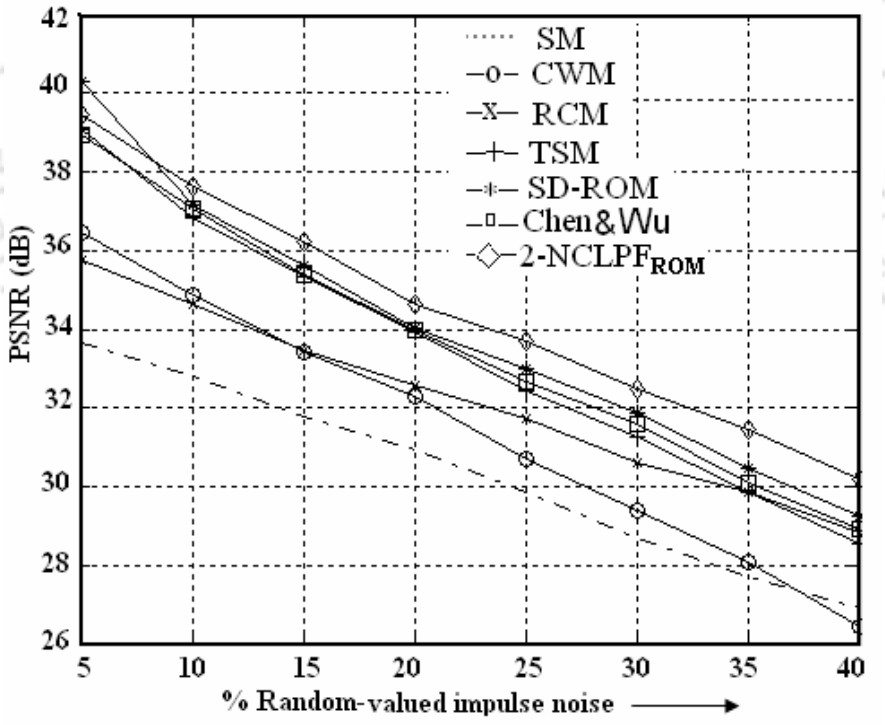
Filters	**% Fixed-valued impulse noise**							
	5	**10**	**15**	**20**	**25**	**30**	**35**	**40**
SM	33.58	32.61	31.85	30.39	29.76	28.91	27.90	26.90
CWM	36.37	34.47	32.71	31.12	28.72	26.65	24.46	23.52
RCM	37.92	37.07	36.28	35.09	34.01	32.62	31.82	31.00
TSM	41.97	38.15	34.60	32.44	29.47	26.63	24.59	22.31
SD-ROM	43.35	40.66	37.84	36.20	34.75	33.03	31.57	30.04
Chen and Wu	40.91	38.93	36.34	35.52	34.17	32.00	31.15	29.51
1-CLPF$_{SM}$	40.15	38.41	36.15	34.59	33.25	31.93	31.17	29.99
1-NCLPF$_{SM}$	41.42	38.95	36.95	35.39	34.14	32.80	31.51	30.33
2-NCLPF$_{SM}$	41.85	39.05	36.92	35.55	33.95	32.64	31.24	29.95
2-NCLPF$_{ROM}$	42.07	39.62	37.41	36.03	34.11	32.88	31.29	29.71
2-NCLPF$_{ROM}$ without pre-filter	39.22	36.36	34.83	33.08	31.92	31.00	29.92	28.95

Table 5.4(b) Comparison of *PSNR* of the Lena image corrupted with 5% to 40% of random-valued impulse noise and restored by different filters

Filters	**% Random-valued impulse noise**							
	5	**10**	**15**	**20**	**25**	**30**	**35**	**40**
SM	33.64	32.82	31.78	30.93	29.88	28.69	27.69	26.91
CWM	36.47	34.88	33.40	32.29	30.68	29.39	28.10	26.44
RCM	35.77	34.61	33.46	32.57	31.74	30.62	29.87	28.81
TSM	39.06	36.82	35.33	33.95	32.45	31.25	29.88	28.56
SD-ROM	39.60	38.03	36.05	34.53	33.20	32.18	30.80	29.72
Chen and Wu	38.95	37.08	35.38	34.00	32.66	31.61	30.11	28.93
1-CLPF$_{SM}$	36.97	34.93	33.07	32.03	31.20	31.14	29.09	28.01
1-NCLPF$_{SM}$	37.78	35.86	34.51	33.30	31.97	30.80	29.62	28.74
2-NCLPF$_{SM}$	38.50	36.37	34.89	33.54	32.03	30.94	29.99	28.77
2-NCLPF$_{ROM}$	39.45	37.64	36.20	34.65	33.71	32.47	31.46	30.17
2-NCLPF$_{ROM}$ without pre-filter	36.24	34.31	32.84	31.91	30.81	30.10	29.24	28.56



(a)



(b)

Figure 5.7 Comparison of the *PSNR* performance of various filters on the Lena image at different noise ratios in removal of (a) fixed-valued impulse noise and (b) random-valued impulse noise.

Figure 5.8a Outputs of various filters: (a) original Lena image (face portion), (b) the image with 20% fixed-valued impulse noise, (c) and (d) the outputs of the SM filter and 2-NCLPF$_{ROM}$, (e) and (f) the magnitudes of differences between the original image and (c) and (d) respectively.

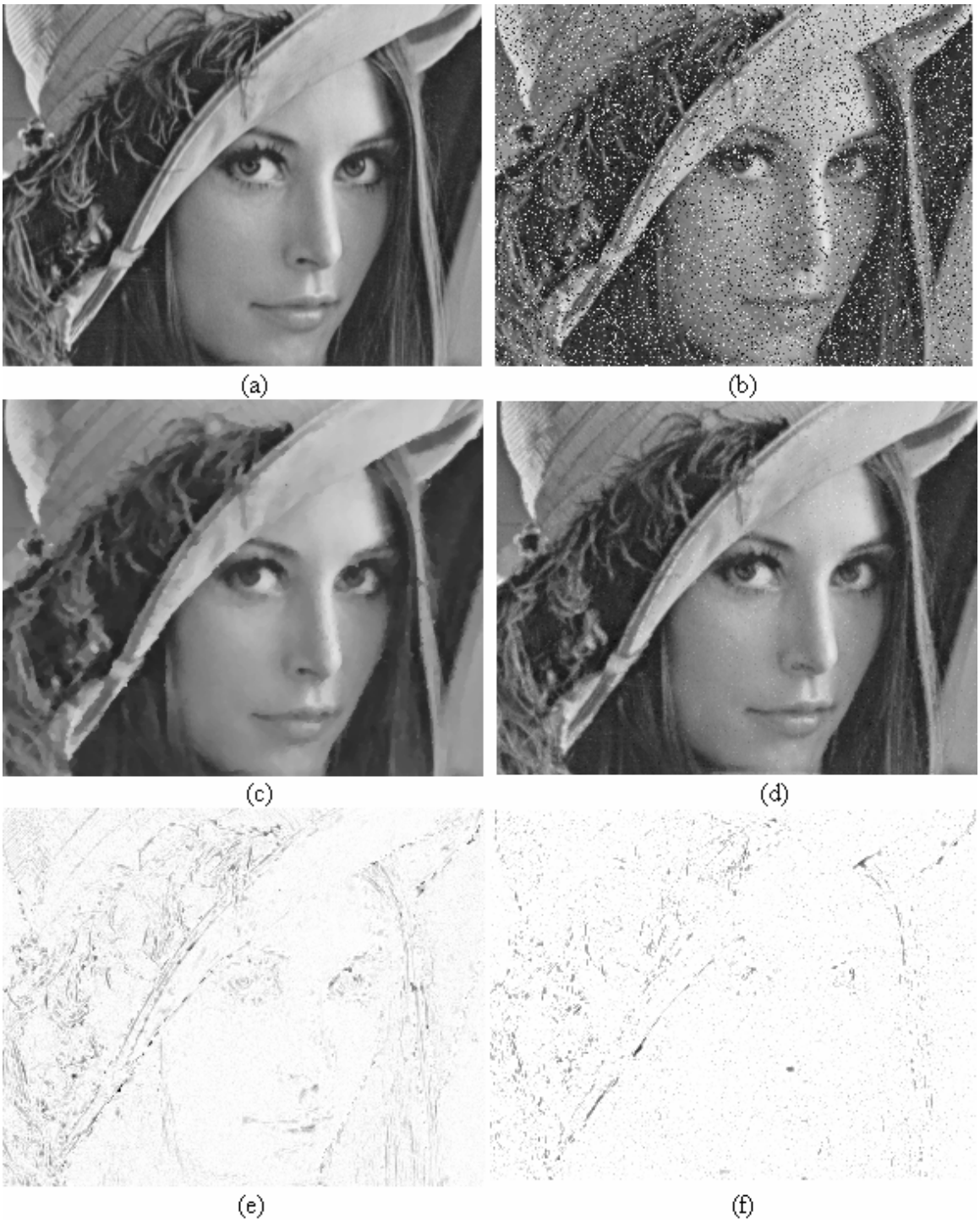


Figure 5.8b Outputs of various filters: (a) original Lena Image (face portion), (b) with 20% random-valued impulse noise, (c) and (d) the outputs of the SM filter and 2-$NCLPF_{ROM}$, (e) and (f) the magnitudes of differences between the original image and (c) and (d) respectively.

Figure 5.8a shows the filtered outputs of the SM filter and 2-NCLPF$_{ROM}$ on the face portion of the Lena image corrupted with 20% fixed-valued impulse noise, and remnant impulses left in filtered outputs (difference between the original image and filter outputs). It has been clearly seen from the figure that many impulses were left unfiltered in the case of the SM filter, whereas such leftovers were minimum in the case of the 2-NLCPF$_{ROM}$. The same was true in the case of random-valued impulse noise and the comparison with the SM filter is shown in Figure 5.8b.

Experiment 5.7.1.4: Comparison with Other Median Filters on Different Images at the same Noise Ratio

In this experiment, the comparative performance of the filters under considerations on different images at the same noise ratio was assessed. Experimental results on images corrupted with 20% of fixed- and random-valued impulse noise are listed in Table 5.5(a) and Table 5.5 (b) respectively. Thresholds were set at 35 and 25 for the linear prediction error based and TSM filters for filtering fixed-valued impulse noise from the Lena, Airplane, Lake, Bridge, Goldhill, Miramar, Point Loma, and Golden Gate images respectively, while these values were 30 and 10 for filtering random-valued impulse noise from the same images. Similarly, thresholds were set at 55 and 40 for the linear prediction error based and TSM filters for filtering fixed-valued impulse noise from the Mandrill, Terrain, Texmos1, and Texmos3 images respectively, while these values were 50 and 25 for filtering random-valued impulse noise from the same images. Different values of multiple thresholds were chosen for the Chen and Wu's adaptive centre-weighted median and SD-ROM filters according to the suggestions in the original papers by Chen and Wu [17] and Mitra *et al*. [3,15,16] respectively.

The noncausal second-order linear-prediction error based filters gave good performance in most of the cases. Thus the proposed filters gave very stable outputs on a large number of images.

Experiment 5.7.1.5: Performance of Linear Prediction Error based Detection

A set of experiments was performed to study the performance of the proposed impulse-noise detection scheme based on 2-D noncausal linear-prediction. The *true hit*, *false hit*,

miss hit and *percentage hit ratio* for the detection scheme were computed at different noise ratios for both types of impulse noise. Results for the Lena image are shown in Table 5.6 and Table 5.7. It has been seen from both the tables that the percentage true hit is uniformly high and close to 99% at the noise ratios considered for both types of noise. However, the number of false hit pixels is higher in case of the random-valued impulse noise and increases with the increase in the noise percentage for both types of noise. The miss hit is uniformly low in the case of random-valued impulse noise. The detection performance of the SD-ROM filter is also included for comparison. Miss hit is low for the SD-ROM filter for the removal of fixed-valued impulse noise, while it is high for the removal of random-valued impulse noise.

Table 5.5(a) Comparative performance of different filters in filtering images corrupted with 20% fixed-valued impulse noise

Images	**SM**	**CWM**	**RCM**	**TSM**	**SD-ROM**	**Chen & Wu**	**2-NCLPF$_{SM}$**	**2-NCLPF$_{ROM}$**
Lena	30.39	31.12	35.09	32.44	36.20	35.52	35.55	36.03
Mandrill	22.22	23.09	24.88	23.33	26.12	25.48	23.71	24.15
Airplane	26.86	29.46	33.11	30.88	31.99	31.99	28.39	28.45
Lake	26.65	27.91	31.05	29.48	30.40	30.38	28.59	28.40
Bridge	25.51	25.49	27.60	26.63	27.65	27.54	27.72	27.30
Goldhill	27.94	30.83	32.64	30.86	33.26	32.90	31.79	30.65
Miramar	24.59	25.69	27.89	26.97	28.42	28.53	27.38	27.80
Point Loma	31.71	31.62	36.06	33.51	37.19	38.64	37.18	37.54
Golden Gate	28.80	29.78	32.85	31.09	33.19	33.86	31.98	31.81
Terrain	22.87	22.97	24.71	23.71	24.29	24.75	26.93	25.00
Texmos1	17.63	17.62	19.73	17.46	16.37	17.10	17.97	18.57
Texmos3	17.71	17.85	19.66	17.52	16.58	15.41	18.41	18.51

Table 5.5(b) Comparative *PSNR* performance of different filters in filtering images corrupted with 20% random-valued impulse noise

Images	**SM**	**CWM**	**RCM**	**TSM**	**SD-ROM**	**Chen & Wu**	**2-NCLPF$_{SM}$**	**2-NCLPF$_{ROM}$**
Lena	30.93	32.29	32.57	33.95	34.53	34.00	33.54	34.65
Mandrill	22.62	23.54	22.83	23.29	23.74	24.15	23.01	23.11
Airplane	27.51	30.33	30.77	31.80	30.51	31.16	27.99	28.17
Lake	27.20	28.77	28.73	29.93	28.63	29.52	27.79	27.65
Bridge	23.94	24.02	23.48	24.50	24.00	24.47	27.32	26.34
Goldhill	28.93	30.11	30.34	31.99	31.30	31.85	30.83	29.53
Miramar	25.26	26.22	25.74	26.91	25.84	26.62	26.66	27.42
Point Loma	32.71	33.57	33.61	35.75	35.93	36.22	35.76	36.90
Golden Gate	30.07	31.29	30.78	31.74	31.23	31.76	31.10	30.82
Terrain	23.13	23.30	22.58	23.55	22.39	23.18	24.63	25.00
Texmos1	17.65	17.67	17.18	16.43	15.38	16.34	17.72	18.20
Texmos3	17.03	17.12	16.63	16.37	15.12	15.84	18.20	18.41

Table 5.6 Performance of the 2-D noncausal linear-prediction error based and SD-ROM filter based impulse detection in terms of the number of true hit, false hit and miss hit and percentage true hit on the Lena image at different ratios for fixed-valued impulse noise

% Impulse noise	**5**	**10**	**15**	**20**	**25**	**30**	**35**	**40**
Total number of Corrupted pixels	13107	26214	39321	52429	65535	78643	91749	104860
2-NCLPF$_{ROM}$								
True hit	12888	25525	39053	51617	64993	77666	90524	102839
False hit	332	474	998	836	3975	6203	10560	15774
Miss hit	219	689	268	812	942	977	1225	2021
% True hit	98	97	99	98	99	99	99	98
SD-ROM								
True hit	13107	26014	38996	52388	65365	78254	90980	103565
False hit	163	219	280	343	410	540	671	764
Miss hit	0	200	325	41	170	389	769	1295
% True hit	100	99.67	99.85	99.71	99.61	99.34	99.13	98.67

Table 5.7 Performance of the 2-D noncausal linear-prediction error based and SD-ROM filter based impulse detection in terms of the number of true hit, false hit and miss hit and percentage true hit on the Lena image at different ratios for random-valued impulse noise

% Impulse noise	**5**	**10**	**15**	**20**	**25**	**30**	**35**	**40**
Total number of corrupted pixels	13107	26214	39321	52429	65535	78643	91749	104860
2-NCLPF$_{ROM}$								
True hit	13020	25766	38968	51966	64583	78050	90491	104085
False hit	1309	1652	3045	524	7054	10755	15594	23588
Miss hit	97	448	353	463	952	593	1258	775
% True hit	99	98	99	99	99	99	99	99
SD-ROM								
True hit	11378	22603	33536	44698	54632	65096	75322	84482
False hit	906	950	934	993	1099	1139	1285	1417
Miss hit	1729	3611	5785	7731	10903	13547	16427	20378
% True hit	86.99	86.62	85.57	84.92	83.71	82.92	82.12	80.93

5.7.2 Experimental Results of Noncausal Linear-prediction Error based Filters for Colour Images

The test images considered were 24-bit colour images of the Lena, Mandrill, Lake, Goldhill, Hut, Tree, Tulip, Cat, Miramar, Point Loma, Golden Gate, Everest, Terrain, Texmos1, Texmos3, Woodland Hill, Brain and Stomach. All images are of size 512×512. Impulse noise was generated by the colour impulse noise model given in Equation 2.10. Both fixed-valued and random-valued impulses were artificially injected to these images at various noise ratios.

A block size of 32×32 was used for the block-based linear prediction. A window of size 3×3 and the City-block distance were used to find out the vector medians. Special care was taken in filtering the vector pixels lying near the block boundary. In the case of vector pixels lying near the boundary between two adjacent blocks, a prediction window overlapping with the adjacent blocks was used. In the case of vector pixels at the image boundary, samples are symmetrically appended for applying causal and noncausal predictions.

Experiment 5.7.2.1: Selection of the Order of Linear Prediction

The first experiment was performed to decide the order of the linear predictor that gives the best results in the term of *PSNR* values. Following prediction filters were used in this report.

(1) First-order causal linear-prediction error based vector median filter (1-CLPVMF)
(2) First-order noncausal linear prediction error based vector median filter (1-NCLPVMF)
(3) Second-order causal linear- prediction error based vector median filter (2-CLPVMF)
(4) Second-order noncausal linear- prediction error based vector median filter (2-NCLPVMF)
(5) Third-order causal linear- prediction error based vector median filter (3-CLPVMF)
(6) Third-order noncausal linear-prediction error based vector median filter (3-NCLPVMF)

Table 5.8(a) Comparison of *PSNR* of the Lena image restored by linear-prediction error based vector median filters for fixed-valued impulse noise

Filters	**% Ffixed-valued impulse noise**					
	10	**20**	**30**	**40**	**50**	**60**
1-CLPVMF	38.10	35.45	33.76	32.49	31.00	29.95
1-NCLPVMF	38.09	35.91	34.20	32.85	31.41	30.01
2-CLPVMF	38.44	35.73	34.00	32.46	31.12	29.56
2-NCLPVMF	38.45	36.10	34.27	32.65	31.45	30.13
3-CLPVMF	38.55	35.61	33.81	32.14	30.75	29.45
3-NCLPVMF	39.49	36.19	34.19	32.65	31.13	29.50

Table 5.8(b) Comparison of *PSNR* of the Lena image restored by linear-prediction error based vector median filters for random-valued impulse noise

Filters	**% Random-valued impulse noise**					
	10	**20**	**30**	**40**	**50**	**60**
1-CLPVMF	36.75	34.47	33.15	31.93	30.73	29.51
1-NCLPVMF	36.55	34.73	33.38	32.18	30.93	29.47
2-CLPVMF	36.74	34.46	32.92	31.60	30.23	29.05
2-NCLPVMF	36.96	35.09	33.48	32.08	30.78	29.78
3-CLPVMF	36.05	34.50	32.88	31.53	30.27	29.27
3-NCLPVMF	37.03	35.03	33.62	32.15	30.77	29.43

Table 5.9(a) Comparison of *PSNR* of the Mandrill image restored by linear-prediction error based vector median filters for fixed- valued impulse noise

Filters	**% Fixed-valued impulse noise**					
	10	**20**	**30**	**40**	**50**	**60**
1-CLPVMF	27.11	25.80	24.62	23.64	22.78	21.80
1-NCLPVMF	27.30	25.98	24.85	23.85	22.89	21.85
2-CLPVMF	27.07	25.79	24.50	23.65	22.75	21.90
2-NCLPVMF	27.37	25.99	24.87	23.89	22.97	22.03
3-CLPVMF	26.85	25.37	24.11	23.08	22.31	21.55
3-NCLPVMF	27.49	25.85	24.61	23.82	22.54	21.69

Table 5.9(b) Comparison of *PSNR* of the Mandrill image restored by linear-prediction error based vector median filters for random- valued impulse noise

Filters	**Random-valued impulse noise**					
	10	**20**	**30**	**40**	**50**	**60**
1-CLPVMF	25.98	25.16	23.10	22.77	22.18	21.47
1-NCLPVMF	26.08	25.20	23.87	23.31	22.47	21.74
2-CLPVMF	26.21	25.17	23.43	22.85	22.17	21.66
2-NCLPVMF	26.32	25.22	24.21	23.44	22.64	21.89
3-CLPVMF	25.86	24.67	23.58	22.76	21.39	21.22
3-NCLPVMF	27.00	25.30	24.09	23.35	22.56	21.77

Experimental results on the Lena and Mandrill images for two types of impulse noise are reported in Table 5.8 and Table 5.9. It has been observed from the tables that no significant improvement in *PSNR* performance is obtained by selecting the order of linear prediction beyond 2 and noncausal predictors give slightly better results than their causal counter-parts. We use the two-dimensional noncausal linear-prediction based vector median filter (2-NCLPVMF) keeping uniformity with the two-dimensional noncausal linear-prediction based median filter(2-NCLPMF).

Experiment 5.7.2.2: Selection of the Threshold for the Linear-prediction Errors for the 2-NCLPVMF

This set of experiment was conducted to find out an appropriate threshold for the prediction error to decide about the impulse in the colour image. Experiments conducted on the Lena and Mandrill images at impulse noise ratios from 5% to 40% for both types of impulse noise are presented in Figure 5.9. It has been observed that the most appropriate thresholds on the Lena image are 50 and 30 for fixed-valued and random-valued impulse noise respectively. Similarly, the most appropriate thresholds for the Mandrill image are 70 and 60 respectively. Experimental results on other images suggest that the appropriate value of the threshold lies between 50 and 80 for fixed-valued impulse noise and between 30 to 70 for random-valued impulse noise.

Experiment 5.7.2.3: Selection of the Colour Noise Model

We have so far used the colour impulse noise model given in Equation 2.10 that considers the corruption of single, double and triple channels. The colour impulse model in Equation 2.9 used by Viero *et al*. [38] considers the corruption of single and triple channels. A set of experiments was performed for comparison of filtering performance under both colour noise models. For the sake of convenience, equal interchannel noise factors $p_1 = p_2 = p_3$ with $p_1 = 0.166,\ 0.2$ and 0.233 and equal double-channel noise factors $p_4 = p_5 = p_6 = 0.1$ are assumed. Table 5.10 gives the restoration results of the proposed 2-NCLPVMF for removal of fixed-valued and random-valued impulse noise from the Lena image corrupted with 10% to 60% impulse noise ratios.

Direct comparison between two models is not possible. For the same single channel noise factor, slightly lower *PSNR* results are observed in the case of the image corrupted according to colour noise model of Equation 2.9 than that of the image corrupted by the colour noise model of Equation 2.10. This is because the number of corrupted components of vector pixels is more in the case of corruption by the former model.

Experiment 5.7.2.4: Comparison with other Vector Filters at different Noise Ratios

This set of experiments was performed for comparison of the proposed filter with some of the standard vector filters. The following filters were used for comparison with the proposed second-order noncausal linear prediction error based vector median filter (2-NCLPVMF):

(1) Vector median filter (VMF_1)
(2) Marginal median filter (MMF)
(3) Centre-weighted vector median filter (CWVMF)
(4) Vector signal dependent rank-ordered mean (VSD-ROM) filter

The performance of the 2-NCLPVMF without the pre-filtering stage is also included to study the importance of this step.

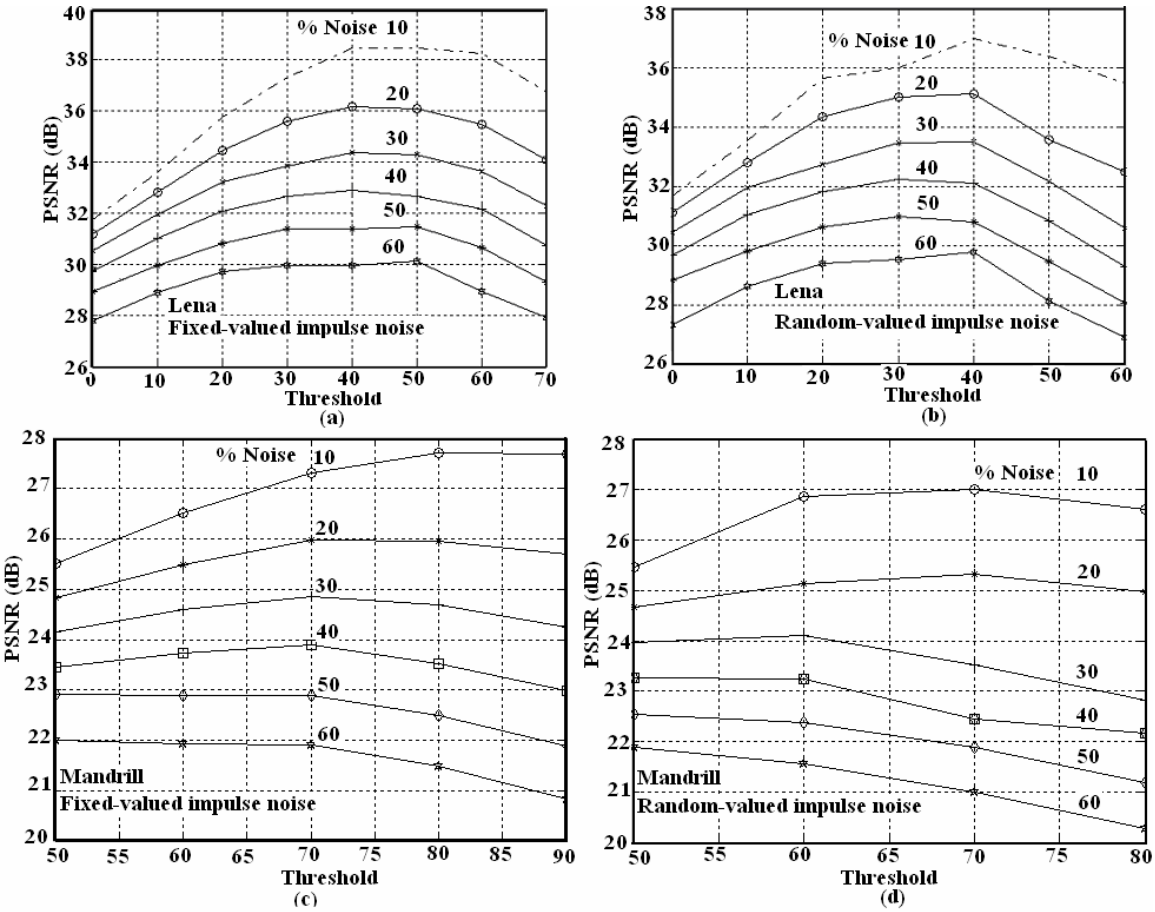


Figure 5.9 Threshold - *PSNR* curve at different percentages: (a), (c) for fixed-valued impulse noise and (b), (d) for random-valued impulse noise.

Table 5.10(a) *PSNR* performance of the 2-NCLPVMF in removal of fixed-valued impulse noise from the Lena image corrupted according to the colour impulse noise models of Equation 2.9 and Equation 2.10

% Noise	**Colour impulse noise model used by Viero *et al.***			**Proposed colour impulse noise model**		
	$p_1 = 0.166$	$p_1 = 0.2$	$p_1 = 0.233$	$p_1 = 0.166$	$p_1 = 0.2$	$p_1 = 0.233$
10	38.84	38.66	38.49	38.45	38.79	38.63
20	35.86	35.54	35.82	36.10	36.13	36.10
30	33.92	33.83	33.89	34.27	34.18	34.26
40	32.98	32.28	32.37	32.65	32.74	32.79
50	30.38	30.71	30.92	31.45	31.36	31.46
60	28.41	29.03	29.49	30.13	29.94	30.29

Table 5.10(b) *PSNR* performance of the 2-NCLPVMF in removal of random-valued impulse noise from the Lena image corrupted according to the colour impulse noise models of Equation 2.9 and Equation 2.10

% Noise	**Colour impulse noise model used by Viero *et al*.**			**Proposed colour impulse noise model**		
	$p_1 = 0.166$	$p_1 = 0.2$	$p_1 = 0.233$	$p_1 = 0.166$	$p_1 = 0.2$	$p_1 = 0.233$
10	37.58	37.48	37.49	36.96	37.42	38.39
20	34.99	35.02	35.07	35.09	35.12	35.02
30	33.15	33.22	33.26	33.48	33.28	33.36
40	31.64	31.77	31.82	32.06	31.96	32.02
50	29.97	30.29	30.51	30.78	30.69	30.86
60	28.28	28.78	29.28	29.78	29.38	29.84

Table 5.11(a) Comparative *PSNR* performance of various filters in restoring the Lena image corrupted with fixed-valued impulse noise at different noise ratios

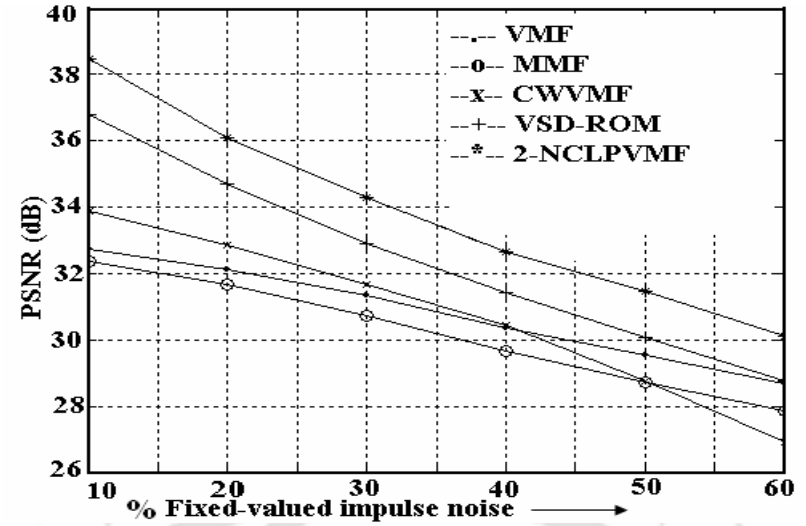
Filters	**% Fixed-valued impulse noise**					
	10	**20**	**30**	**40**	**50**	**60**
VMF	32.74	32.14	31.36	30.38	29.54	28.67
MMF	32.37	31.68	30.71	29.68	28.71	27.85
CWVMF	33.89	32.87	31.65	30.46	28.75	26.92
VSD-ROM	36.77	34.70	32.90	31.41	30.08	28.75
2-NCLPVMF	38.45	36.10	34.27	32.65	31.45	30.13
2-NCLPVMF without pre-filter	37.66	35.19	33.48	32.08	30.75	29.30

Table 5.11(b) Comparative *PSNR* performance of various filters in restoring the Lena image corrupted with random-valued impulse noise at different noise ratios

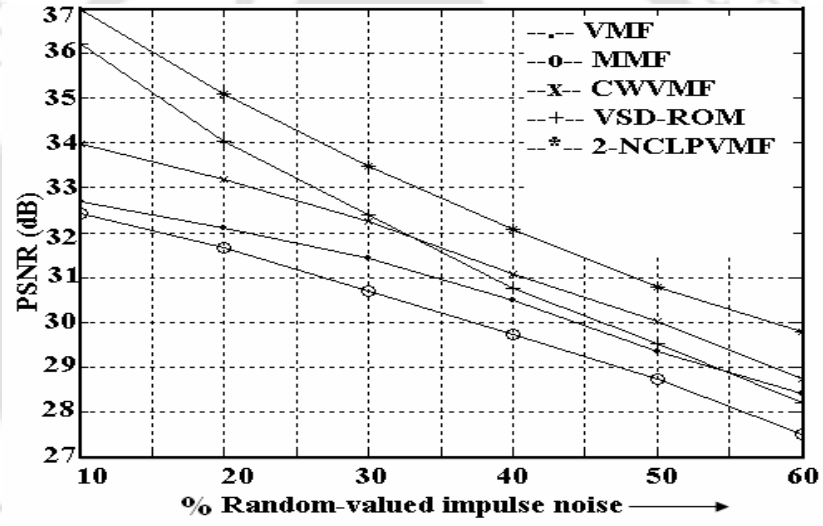
Filters	% Random-valued impulse noise					
	10	**20**	**30**	**40**	**50**	**60**
VMF	32.68	32.11	31.43	30.48	29.36	28.41
MMF	32.41	31.65	30.71	29.72	28.75	27.52
CWVMF	33.98	33.17	32.25	31.08	30.02	28.75
VSD-ROM	36.21	34.03	32.40	30.75	29.52	28.22
2-NCLPVMF	36.96	35.09	33.48	32.08	30.78	29.78
2-NCLPVMF without pre-filter	36.48	34.47	32.67	31.22	29.88	28.55

Results shown in Table 5.11 were taken on the Lena image for fixed- and random-valued impulse noise ratios ranging from 10% to 60%. Here p_1, p_2 and p_3 were kept fixed at 0.166 while p_4, p_5 and p_6 were assigned a fixed value of 0.10. The threshold for the *LPE* was set at 30 for random-valued impulse noise and at 50 for fixed-valued impulse noise. Thresholds for the VSD-ROM filter were set at 30, 60, 100 and 140 for random-valued impulse noise and at 40, 70, 120 and 160 for fixed-valued impulse noise. It is observed from the tables that the 2-NCLPVMF gives better performance in comparison with the VMF, the MMF, the CWVMF and the VSD-ROM filter. The 2-NCLPVMF without the pre-filter gives inferior results. Data from Table 5.11 are plotted graphically in Figure 5.10. Note the significant improvement in *PSNR*s achieved by the proposed method.

Figures 5.11 and 5.12 show the superiority of the 2-NCLPVMF over the VMF when applied on the Lena image corrupted with 60% impulse noise. Figures 5.11(e), 5.11(f), 5.13(e) and 5.13(f) show the difference between these filter outputs and the original image. It has been observed from these images that there are more impulses leftover in the VMF outputs. Enlarged parts of face are shown in Figures 5.12(a), 5.12(b), 5.14(a) and 5.14(b) respectively. Notice the distortion in the VMF output.



(a)



(b)

Figure 5.10 Impulse noise ratio – *PSNR* curve for different vector filters on the Lena image: (a) fixed-valued impulse noise and (b) random-valued impulse noise.

Experiment 5.7.2.5: Comparison with Other Vector Filters on Different Images

The aim of this experiment was to study the comparative performance of the filters under study on a wide variety of colour images. The results listed in Table 5.12 are at 20% impulse noise ratio (p_1 = 0.166, p_2 =0 .166, p_3 = 0.166, p_4 = 0.10, p_5 = 0.10 and p_6 = 0.10) on the Mandrill, Lake, Goldhill, Miramar, Point Loma, Golden Gate, Terrain, Texmos1, Texmos3, Kodak Girl, Hut, Tree, Tulip, Cat, Earth, Woodland Hill, Everest,

Brain and Stomach images. Here again, 2-NCLPVMF outperformed other filters for both types of impulse noise, except for the Lake image for which the VSD-ROM filter gives slightly better result for fixed- and random-valued impulse noise and the Hut image for which the VSD-ROM filter gives better results for random-valued impulse noise. However, the performance of the proposed filter on these images also was better than the VSD-ROM filter for the noise ratios above 20%. For example, *PSNR* values for the 2-NCLPVMF and VSD-ROM filter for 40% fixed-valued impulse noise on the Lake image are 26.31 dB and 26.17 dB respectively. Similarly, the *PSNR* values for the 2-NCLPVMF and VSD-ROM filter output at 40% fixed-valued impulse noise on the Hut image are 27.69 dB and 27.20 dB respectively. It has been observed from these results that the 2-NCLPVMF gives very stable outputs on different types of images.

Experiment 5.7.2.6: Performance of Constrained Intrachannel Linear Prediction Error based Impulse Detection

This set of experiments was carried out to study the performance of the proposed impulse-noise detection scheme based on 2-D noncausal constrained intrachannel linear-prediction. The *true hit*, *false hit*, *miss hit* and *percentage true-ratio* for the detection scheme were computed at different noise ratios for both types of impulse noise. Results for the Lena image are shown in Table 5.13 (a) and Table 5.13 (b). It has been seen from both the tables that the percentage true hit is uniformly high and close to 94% for fixed-valued impulse noise and 82% for random-valued impulse noise and is slightly lower than the corresponding values for the VSD-ROM filter. The number of miss hit pixels is comparatively higher in the case of random-valued impulse noise. The false hit is lower than that of the VSD-ROM filter except for the percentage noise of 60. The improved PSNR performance of the 2-NCLPVMF may be attributed to lower false hit in the detection stage.

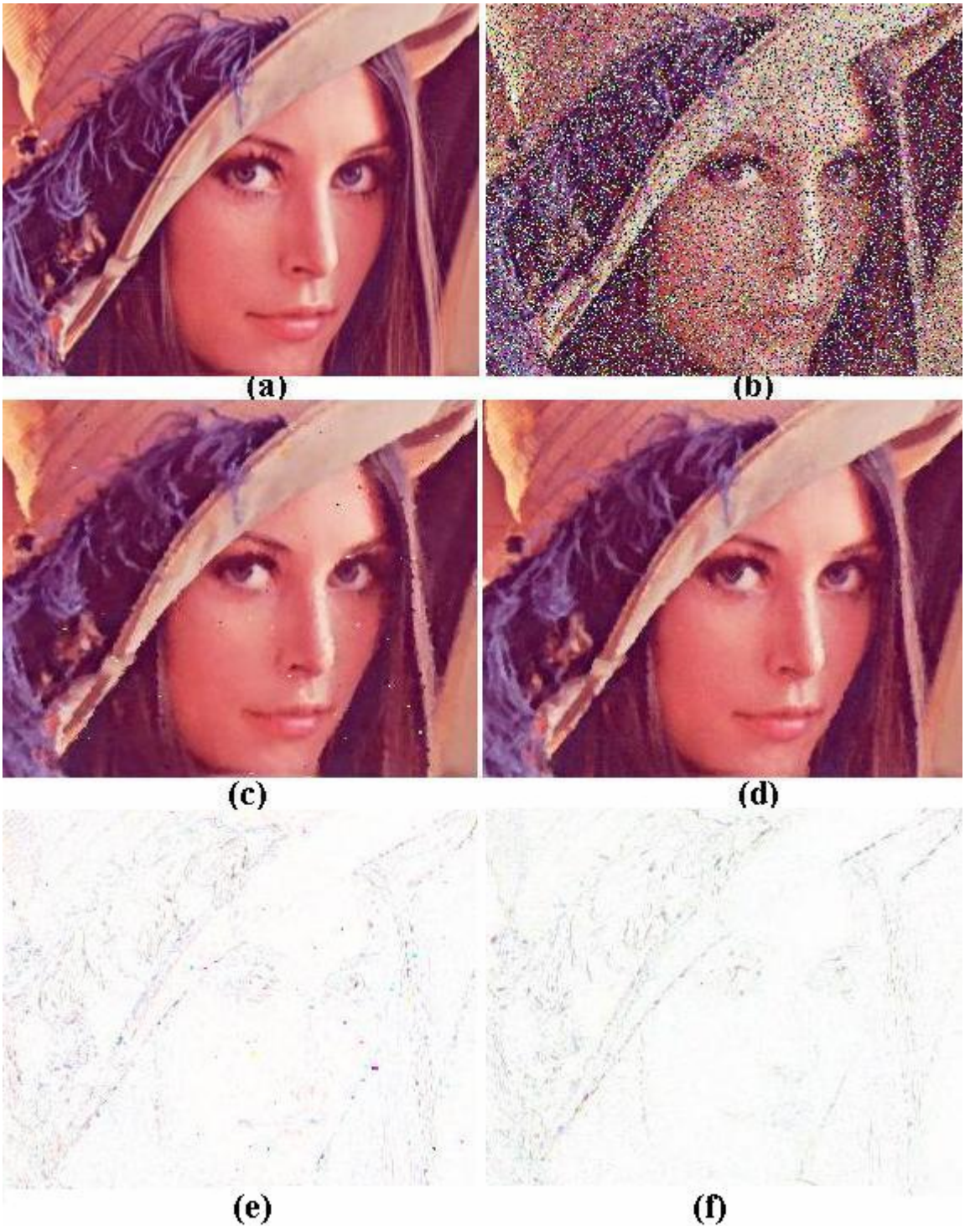


Figure 5.11 Comparison of the VMF_1 and 2-NCLPVMF: (a) original Lena Image (face portion), (b) Lena image corrupted with 60% fixed-valued impulse noise, (c) and (d) the outputs of VMF_1 and 2-NCLPVMF respectively, (e) and (f) the magnitudes of differences between the original and, (c) and (d) respectively.

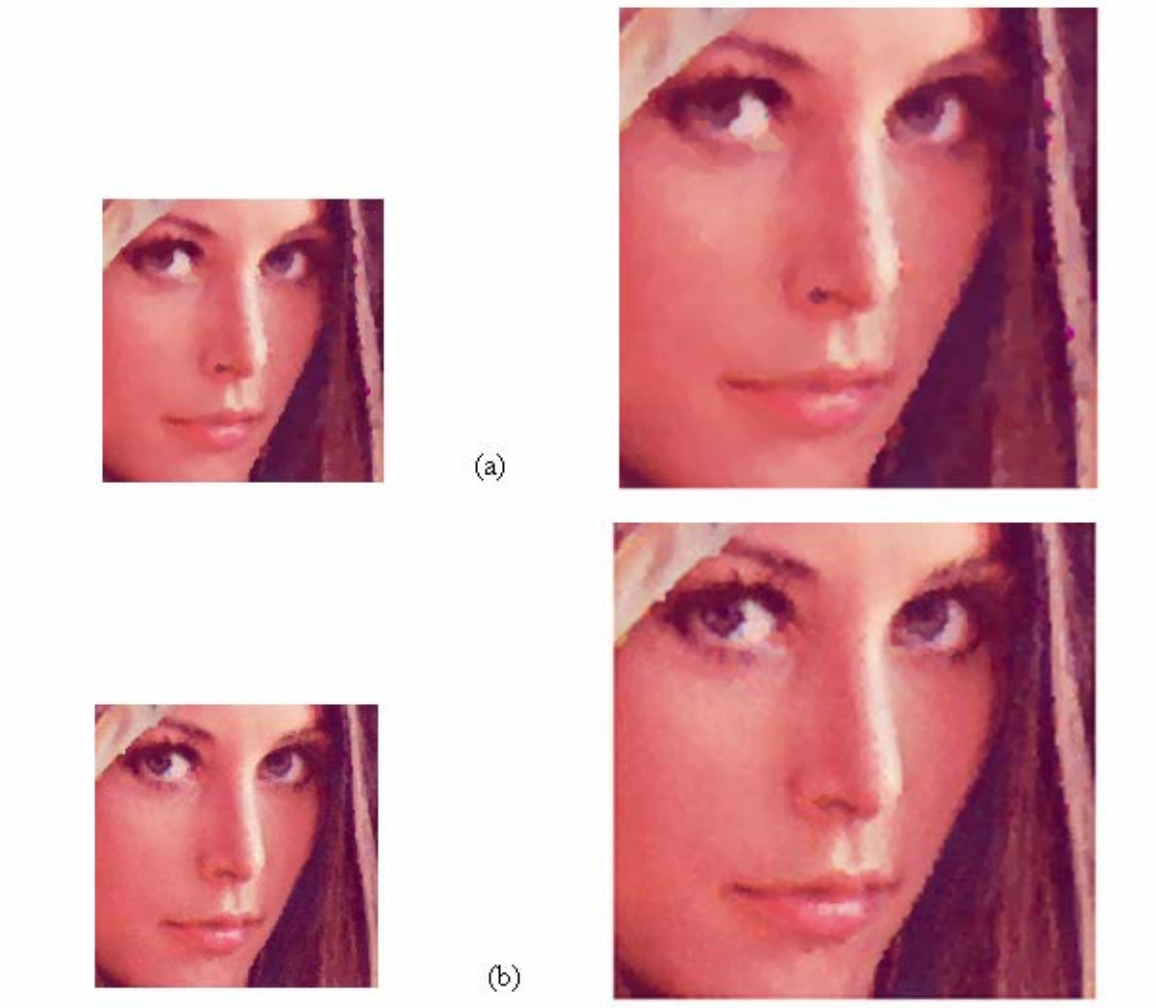


Figure 5.12 Comparison of the VMF_1 and 2-NCLPVMF for fixed-valued impulse noise on the Lena image (face part): VMF_1 output and its enlargement and (b) 2-NCLPVMF output and its enlargement.

Figure 5.13 Comparison of the VMF_1 and 2-NCLPVMF: (a) original Lena Image (face portion), (b) Lena image corrupted with 60% random-valued impulse noise, (c) and (d) the outputs of VMF_1 and 2-NCLPVMF respectively, (e) and (f) the magnitudes of differences between the original, and (c) and (d) respectively.

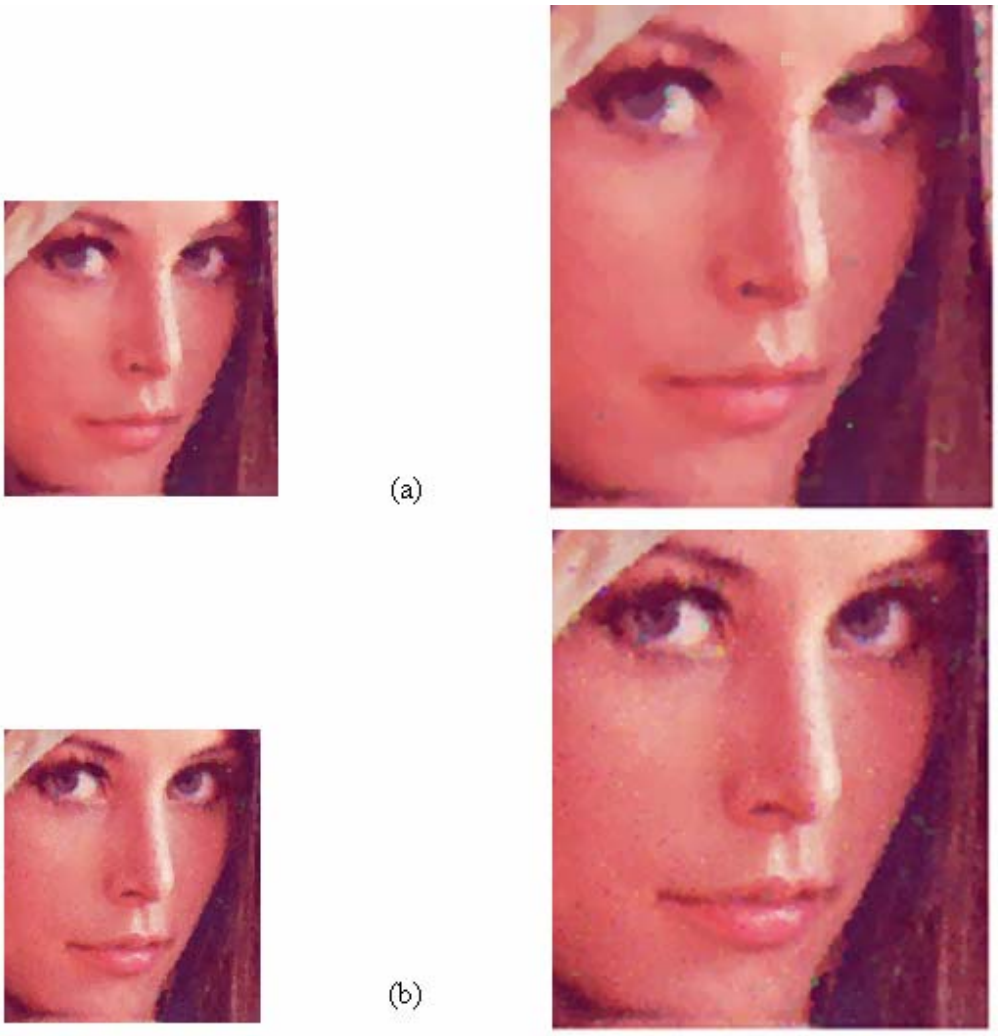


Figure 5.14 Comparison of the VMF_1 and 2-NCLPVMF for random-valued impulse noise on the Lena image (face part): (a) VMF_1 output and its enlargement and (b) 2-NCLPVMF output and its enlargement.

Table 5.12(a) Comparative restoration results of various filters in removal of fixed-valued impulse noise at 20% noise from different images

Images	Filters							
	VMF_1	**MMF**	**CWVMF**	**VSD-ROM**	**1-CLPVMF**	**1-NCLPVMF**	**2-CLPVMF**	**2-NCLPVMF**
Lena	32.14	31.68	32.87	34.70	35.45	35.91	35.73	36.10
Mandrill	22.67	22.22	23.61	23.35	25.80	25.98	25.79	25.99
Lake	27.12	26.91	28.28	28.66	28.01	28.26	28.09	28.20
Goldhill	29.52	29.03	30.48	31.79	31.86	32.06	32.92	32.15
Hut	29.01	28.00	30.22	29.46	30.98	30.81	31.78	31.24
Tree	28.67	27.62	29.84	29.65	31.15	31.89	31.56	31.80
Tulip	32.52	31.34	33.30	33.68	33.66	34.23	33.74	34.71
Cat	29.11	28.44	30.20	30.13	30.78	31.62	30.74	31.59
Miramar	25.95	25.44	26.96	27.78	28.43	30.33	30.03	30.37
Point Loma	33.17	32.89	33.79	37.70	38.80	37.66	36.19	38.00
Golden Gate	30.41	29.87	31.39	31.69	32.00	32.38	32.74	32.64
Everest	22.80	22.48	23.69	23.14	23.53	24.03	25.52	25.72
Terrain	20.72	20.71	21.52	21.16	25.49	24.70	24.31	24.67
Texmos1	17.94	16.86	18.94	17.43	20.54	21.03	20.71	21.24
Texmos3	17.90	16.97	18.98	17.54	20.68	21.11	20.86	21.23
Woodland Hill	25.88	25.39	27.16	27.93	26.76	28.07	28.14	28.13
Brain	27.71	26.69	28.86	28.30	29.22	29.52	29.77	30.02
Stomach	33.31	32.76	33.71	36.30	36.89	35.43	36.94	37.92

Table 5.12(b) Comparative restoration results of various filters in removal of random-valued impulse noise at 20% noise from different images

Images	Filters							
	VMF_1	**MMF**	**CWVMF**	**VSD-ROM**	**1-CLPVMF**	**1-NCLPVMF**	**2-CLPVMF**	**2-NCLPVMF**
Lena	32.11	31.65	33.17	34.03	34.47	34.73	34.46	35.09
Mandrill	22.61	22.29	23.62	23.28	25.16	25.20	25.17	25.22
Lake	27.09	26.91	28.25	28.58	27.77	28.11	27.91	28.14
Goldhill	29.44	29.05	30.49	29.47	31.51	31.79	31.72	31.85
Hut	28.92	27.89	30.27	31.53	30.17	30.84	30.51	31.16
Tree	28.64	27.58	29.92	30.91	30.56	31.39	31.06	31.60
Tulip	32.12	31.30	33.64	33.38	32.96	33.39	33.74	34.68
Cat	27.90	28.00	30.26	30.08	30.51	30.04	30.81	30.96
Miramar	25.92	25.41	26.99	27.62	28.04	28.65	28.59	29.07
Point Loma	33.07	32.83	33.97	35.83	36.23	35.72	35.58	36.73
Golden Gate	30.44	29.95	31.47	31.59	31.40	31.90	32.50	32.71
Everest	22.83	22.46	23.76	23.11	23.55	23.92	24.68	24.93
Terrain	20.68	20.68	21.56	21.16	23.75	24.09	23.58	23.82
Texmos1	17.66	16.76	18.79	17.24	19.82	20.27	20.12	20.51
Texmos3	17.78	16.88	18.82	17.34	20.00	20.38	18.99	20.62
Woodland Hill	25.81	25.36	27.10	27.72	27.56	26.98	27.51	28.06
Brain	27.54	26.63	28.29	28.01	28.62	28.92	28.79	28.97
Stomach	33.23	32.70	34.05	36.63	37.96	36.70	38.02	37.77

Table 5.13(a) Performance of the 2-NCLPVMF and VSD-ROM filter based impulse detection in terms of the number of true hit, false hit and miss hit and percentage true hit on the Lena image at different ratios for fixed-valued impulse noise

% Impulse noise	**10**	**20**	**30**	**40**	**50**	**60**
Total number of corrupted pixels	26214	52429	78643	104860	131070	156744
2-NCLPVMF						
True hit	24386	48959	73215	98801	122738	147442
False hit	346	395	704	1191	2014	3743
Miss hit	1828	3470	5428	6059	8332	9302
% True hit	93.03	93.38	93.10	94.23	93.64	94.74
VSD-ROM						
True hit	24777	50352	75438	100874	126286	151161
False hit	1868	1883	1921	2868	3165	3647
Miss hit	1237	2077	3205	3983	4786	6125
% True hit	94.52	96.04	95.93	96.20	96.35	96.11

Table 5.13(b) Performance of the 2-NCLPVMF and VSD-ROM filter based impulse detection in terms of the number of true hit, false hit and miss hit and percentage true hit on the Lena image at different ratios for random-valued impulse noise

% Impulse noise	**10**	**20**	**30**	**40**	**50**	**60**
Total number of corrupted pixels	26214	52429	78643	104860	131070	156744
2-NCLPVMF						
True hit	21227	41941	64159	85872	108031	129450
False hit	1022	1060	1880	2406	3051	5036
Miss hit	4987	10488	11484	18988	23039	27294
% True hit	80.98	80.00	81.58	81.90	82.42	82.30
VSD-ROM						
True hit	21523	43144	64663	86126	108664	130460
False hit	1858	1902	2095	2876	3220	3886
Miss hit	4096	9284	13980	18730	22408	26820
% True hit	82.10	82.29	82.22	82.14	82.91	82.95

5.8 Conclusions

This chapter described causal and noncausal linear prediction error based filters for removal of impulse noise from single and multichannel images. Block-based linear

prediction was considered and the prediction coefficients were obtained from the estimated autocorrelation functions in the block. The centre pixel inside the sliding window is predicted in terms of the neighbouring pixels. The centre pixel is replaced by the median or the ROM value if the difference between the centre pixel and estimated one is bigger than a pre-defined threshold. In the case of the multichannel image, the constrained intrachannel linear prediction was used for detection of the impulse and the corrupted vector pixels were replaced by the VMF_1 output. It has been seen from experimental results that the performance of noncausal predictors is comparatively better than that of causal predictors. It has also been seen that the performance gain beyond second-order predictors is not appreciable.

The 2-$NCLPF_{ROM}$ filter is the best among prediction error based filters for gray-scale images. Outputs of the prediction based filters were stable for different types of images at different impulse noise ratios. Six different filters namely the SM, CWM, RCM, TSM, SD-ROM, and Chen and Wu's adaptive median filters were used to compare with the proposed linear prediction error based filters. The performance of the 2-$NCLPF_{ROM}$ filter is better than that of the SM filter and comparable to that of the SD-ROM filter on most of the images under experiment.

The proposed 2-NCLPVMF was compared with the vector median filter (VMF_1), the marginal median filter (MMF), the centre-weighted vector median filter (CWVMF) and the vector signal dependent rank-ordered mean (VSD-ROM) filter. It outperformed other filters under considerations in removal of both types of impulse noise from eighteen different images in the terms of *PSNR* values. It preserves edges and image details while removing impulse noise from corrupted images. The overall computational cost of the proposed linear prediction error based vector median filters is less in comparison with that of the VMF_1 as the operations of the VMF_1 are performed only on the corrupted pixels and pre-filtering is done with the MMF.

Chapter 6

Conclusions

As a roundup of the present thesis, we will briefly discuss the achievements, compare the proposed algorithms and suggest possible extensions. The work concerned with developing detection-based filtering algorithms for removing impulse noise from an image. Two broad classes of detection-based impulse noise filters have been proposed. The filters are:

- Rank-conditioned and threshold median filters
- Filters using linear prediction error based impulse-noise detection

The filters have been appropriately extended for removing impulse noise from colour images. The strength of these filters lies on their ability to detect impulses from images prior to filtering operations and they discriminate between noise-free pixels and corrupted ones.

The rank-conditioned and threshold (RCT) median filters are based on impulse noise detection by *simultaneous application two detection conditions* with an aim to reduce the probability of detecting a healthy pixel as an impulse and the probability of detecting a noisy pixel as healthy. The two conditions are *rank conditioning* and *thresholding* the absolute differences between the centre pixel and healthy pixels in a window. A filtering window of size $n \times n$ containing $N = n^2$ pixels was used and the following RCT filters were developed.

- RCT- I – Rank-conditioning and thresholding the absolute difference between the centre pixel and the median
- RCT- II - Rank-conditioning and adaptive thresholding of the absolute differences between the centre pixel and the CWM filter outputs for $\frac{N-1}{2}$ centre weights
- RCT- III - Rank-conditioning and thresholding of the absolute differences between the centre pixel and $\frac{N-1}{2}$ rank-ordered pixels

The corrupted pixel is replaced by the rank-ordered mean (ROM) in all cases.

In the RCT-I filter, the centre pixel inside the sliding window is decided corrupted if its rank does not lie in the trimming range and if its difference from the median is bigger than a pre-defined threshold. An appropriate threshold is used to distinguish between an impulse and a feature pixel. It was seen from both subjective and objective observations that the performance of the RCT-I filter is better than that of the standard median (SM), centre-weighted median (CWM), rank-conditioned median (RCM), tri-state median (TSM) and Chen and Wu's adaptive centre-weighted median filters at different impulse noise ratios for fixed-valued and random-valued impulse noise.

The RCT-II filter employs the impulse detection schemes of the rank-conditioned median filter and the Chen and Wu's adaptive centre-weighted median filter. The Chen and Wu's adaptive centre-weighted median filter is based on thresholding the differences

between the centre pixel and the outputs of centre-weighted median (CWM) filters with varied centre weights. It was seen that the PSNR performance of the RCT-II filter is distinctly better than that of the standard median (SM), centre-weighted median (CWM), rank-conditioned median (RCM), tri-state median (TSM), SD-ROM, Chen and Wu's adaptive centre-weighted median and the RCT-I filters at various impulse noise ratios for fixed-valued and random-valued impulse noise. However, its computational cost is higher than that of the RCT-I filter.

The RCT-III filter uses the impulse detection schemes of the rank-conditioned median filter and the signal-dependent rank-ordered mean filter. The rank-ordered differences are thresholded. It was seen that the *PSNR* performance of the RCT-III filter is better than that of the standard median (SM), centre-weighted median (CWM), rank-conditioned median (RCM), tri-state median (TSM), SD-ROM and Chen and Wu's adaptive centre-weighted median filters at various impulse noise ratios for fixed-valued and random-valued impulse noise. Both *PSNR* performance and computational cost are intermediate between those of the RCT-I and RCT-II filters. The RCT filters have high true-hit and low false hit as evidenced by the experiments. The filters have very stable performance on different types of images.

Two detection-based vector median filters, namely, the rank-conditioned vector median filter (RCVMF) and the rank-conditioned and threshold vector median filter (RCTVMF) were developed to enhance the performance of the vector median filter (VMF). Vector pixels in the window are ranked according to the *sum of distances* of vector pixels from all other vector pixels in the window and a trimming range is defined. The vector pixel that has the least sum of distances is the vector median. In the RCVMF, the centre vector pixel will not be replaced by vector median so long as its rank lies in the trimming range. The RCTVMF replaces the centre vector pixel by the vector median if the rank of the centre vector pixel is bigger than that of a pre-defined rank and its distance from a healthy vector pixel is big. The vector median is considered as the healthy pixel. It was found that the proposed RCVMF outperforms the VMF in terms of *PSNR* values and visual perception. The RCTVMF outperforms the vector median filter based

on $L_1 - norm$ (VMF_1), the vector median filter based on $L_2 - norm$ (VMF_2), the basic vector directional filter (BVDF), the directional distance filter (DDF), the marginal median filter (MMF) filter, the centre-weighted vector median filter (CWVMF), the rank-conditioned vector median filter (RCVMF) and the vector signal dependent rank-order mean (VSD-ROM) filter for both types of noise in terms of the *PSNR* value and visual quality. However, its computational complexity is slightly higher than that of the VMF. True-hit is high and false hit is low for the proposed RCTVMF. Both RCTVMF and RCVMF have stable performance on various types of images.

Another novel scheme for detecting impulse noise using causal and noncausal linear prediction was developed for both gray-scale and colour images. The image is pre-filtered using a median filter in the case of gray-scale image and a marginal median filter in the case of a colour image. Block-based linear prediction is applied and the impulse is detected by thresholding magnitude of the prediction error in the case of the gray-scale image and the $L_\infty - norm$ of the prediction error vector in the case of the colour image. The performance of the second-order noncausal linear prediction error based median (2-$NCLPF_{SM}$) filter was found to be better than that of the corresponding causal one and that of the SM, CWM and RCM filters at low noise ratio. The computational cost of proposed filters is low. These filters gave very stable outputs on different types of images.

The second-order noncausal *LPE* based vector median filter (2-NCLPVMF) promises to be a better alternative to the VMF. It outperformed the VMF_1, VMF_2, MMF, BVDF, DDF, CWVMF, VSD-ROM filter, RCVMF and RCTVMF in removal of impulse noise from various images for both types of impulse noise in terms of *PSNR* value and visual perception. The computational cost of the 2-NCLPVMF under consideration is comparatively lower than the VMF as discussed earlier. It has stable performance on different types of images.

6.1 Overall Performance of Different Filters for Gray-scale Images

To assess the overall filtering performance of the proposed filters, the average *PSNR* over a number of images filtered by each (filter) is determined. For gray-scale images, the average *PSNR* performance on different images is considered. The average *PSNR* performance of the SM, CWM, RCM, TSM, SD-ROM and Chen and Wu's adaptive centre-weighted median filters is used for comparison. The filter with the maximum value of the average *PSNR* is graded as one. The other filters are similarly graded. Table 6.1 shows the overall performance of all filters under consideration on gray-scale images corrupted with both types of impulse noise at 20% noise ratio. The best filters based on this criterion are the RCT-II filter for fixed-valued impulse noise and the RCT-III filter for random-valued impulse noise. The average performance of the three RCT filters is ahead of other filters and the RCM filter follows these three in the case of fixed valued noise. In the case of random-valued noise, the TSM filter performs slightly better than the RCT–I filter and the RCM filter, as expected, performs very poorly. The performance of the *LPE* based filters is poorer than that of the RCT filters in the case of the gray-scale image.

The computational costs per pixel of the proposed *RCT* and *LPE* based filters for filtering with a 3×3 window at 20% impulse noise are summed up in Table 6.2. A block size of 32×32 is assumed for estimating the linear prediction parameters. The corresponding costs of the SM, the CWM, the RCM, the SD-ROM and Chen and Wu's adaptive centre-weighted median filters are also included. The tabulated costs are not optimal, but only indicative of the computational overhead of the proposed methods. Note the increased number of comparison operations RCT-II filters. The *LPE* based filters involve additional multiplication and addition operations. The 2-CLPF$_{SM}$ and 2-NCLPF$_{SM}$ involve less number of operations compared to each of the CWM, the SD-ROM, Chen and Wu's adaptive centre-weighted median and the other proposed filters except the RCT-I filter in the table.

From the presentations in the earlier chapters and the above discussion, it is clear that simultaneous application of two detection conditions improves the efficiency of impulse noise detection. The RCT filters have very good filtering performance under different impulse noise conditions. The *LPE* filters having better performance than the median filter compare poorly with the RCT filters.

Table 6.1(a) Ranking of filters based on average *PSNR* performance on gray-scale images with 20% fixed–valued impulse noise

Rank	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
Filters	**RCT - II**	**RCT - III**	**RCT - I**	**RCM**	**Chen and Wu**	**SD-ROM**	**2-NCLPF$_{SM}$**	**2-NCLPF$_{ROM}$**	**TSM**	**CWM**	**SM**
Average PSNR	30.70	30.69	30.33	29.39	28.99	28.98	27.97	27.85	27.48	26.49	25.45

Table 6.1(b) Ranking of filters based on average *PSNR* performance on gray-scale images with 20% random –valued impulse noise

Rank	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
Filters	**RCT - III**	**RCT - II**	**TSM**	**RCT - I**	**Chen and Wu**	**2-NCLPF$_{SM}$**	**SD-ROM**	**2-NCLPF$_{ROM}$**	**RCM**	**CWM**	**SM**
Average PSNR	28.30	28.17	27.85	27.74	27.48	27.18	27.16	27.05	27.04	26.86	25.83

Table 6.2 Computational cost per pixel for filtering with a 3×3 window

Filters	Number of Operations			
	Additions/Subtractions	Multiplications	Comparisons	Insertions
SM	---	---	16	---
CWM	---	---	36.53	2
RCM	---	---	30.53	---
SD-ROM	5	1	29	---
Chen and Wu	17	1	73.06	6
RCT - I	2	1	32.53	---
RCT - II	18	2	75.06	6
RCT - III	5	1	38	1
2-CLPF$_{SM}$	10.70	10.71	21.20	----
2-NCLPF$_{SM}$	20.10	16.12	21.20	----
2-CLPF$_{ROM}$	11.70	11.71	21.20	----
2-NCLPF$_{ROM}$	21.10	17.12	21.20	---

6.2 Overall Performance of Different Filters for Colour Images

To assess the comparative filtering performance of the proposed filters for colour images, the average *PSNR* values of the outputs of thirteen different filters on eighteen different colour images are considered. The filter with the maximum average *PSNR* value has rank "one" and so on for the remaining filters. Table 6.3 shows the overall performance of all filters under consideration on colour images corrupted with both types of impulse noise at 20% noise ratio. All the proposed vector filters perform ahead of existing vector filters. The best filter based on this criterion is the 2-NCLPVMF. In the case of fixed-valued impulse noise, first-order *LPE* based vector median filters perform significantly better than the VMF. In the case of random-valued impulse noise, the performance of the RCTVMF is comparable with that of the 2-CLPVMF

The computational costs per vector pixel for a 3×3 filtering window for the VMF_1, MMF, RCVMF, RCTVMF, VSD-ROM filter, 2-CLPVMF and 2-NCLPVMF (at $p = 0.20$ and 32×32 block size for the linear prediction error based filters) are summed up in Table 6.4. The tabulated costs are not optimal, but only indicative of the computational overhead of the proposed methods. It has been seen that the proposed 2-NCLPVMF requires more operations for the detection of the corrupted vector pixels. The overall computational time of the proposed linear prediction error based filters is less as compared with that of the VMF, CWVMF, VSD-ROM filter, RCVMF and RCTVMF.

Table 6.3(a) Ranking of filters based on average *PSNR* performance on colour images with 20% fixed–valued impulse noise

Rank	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Filters	**2-NCLPVMF**	**2-CLPVMF**	**1-NCLPVMF**	**1-CLPVMF**	**RCTVMF**	**RCVMF**	**VSD-ROM**	**CWVMF**	**VMF_1**	**VMF_2**	**DDF**	**MMF**	**BVDF**
Average PSNR	30.10	29.75	29.72	29.45	29.36	28.37	28.35	27.99	27.03	26.67	26.51	26.41	24.05

Table 6.3(b) Ranking of filters based on average *PSNR* performance on colour images with 20% random–valued impulse noise

Rank	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Filters	**2-NCLPVMF**	**RCTVMF**	**2-CLPVMF**	**1-NCLPVMF**	**1-CLPVMF**	**RCVMF**	**VSD-ROM**	**CWVMF**	**VMF_1**	**MMF**	**VMF_2**	**DDF**	**BVDF**
Average PSNR	29.55	29.16	29.10	29.06	28.89	28.32	28.20	28.02	26.88	26.35	26.31	26.29	24.12

Table 6.4 Computational cost per vector pixel using 3×3 window

Filters	**Number of Operations**		
	Additions/Subtraction	**Multiplications**	**Comparisons**
VMF	165	------	71
MMF	------	------	48
VSD-ROM	153	------	94
RCVMF	165	------	80
RCTVMF	170	------	84
2-CLPVMF	65.03	32.03	68.20
2-NCLPVMF	93.27	48.28	68.20

The presentations in the earlier chapters and the discussion above show the superiority of the proposed vector filters. The RCTVMF and the *LPE* based vector median filters have significant improvement in *PSNR* performance over the existing vector median filters at the cost of additional computational operations. These results also suggest that there is still room for improvement in the performance of the detection based VMF.

6.3 Scope for Further Developments

The thesis concerned developing median filters by incorporating better detection schemes. More stress was given to detection of impulse noise than to filtering. Several issues come to mind while summarising the proposed techniques:

(1) Our detection schemes, particularly the one based on rank-conditioning and thresholding, ensure that image features are not detected as impulses. However, the filtering stage still involves the noisy and feature pixels. It will be interesting to develop filtering schemes that preserve edges and other details.

(2) Like other median-based techniques, the proposed techniques work well for impulse noise ratio upto 50% for the gray-scale images. Some of the recent developments address the issues of filtering images with more than 50% of impulse noise ratio. It will be interesting to extend the proposed algorithm in the light of the recent developments for filtering images with impulse noise ratio more above 50%.

(3) The filtering window in the proposed algorithms was kept fixed at 3×3. It will be particularly worthwhile to investigate the improvement of filtering performance by increasing window size in case of rank-conditioning and thresholding. In [12] it is shown that a bigger window size has an advantage over the smaller window size in terms of the mean absolute error between the original and the RCM-filtered image.

(4) The linear prediction error based detection mechanism assumes local stationarity of the image. Impulse noise detection depends on the quality of the linear prediction. The prediction parameters are to be estimated accurately from the image data. As we are not concerned with synthesis through the inverse of the prediction error filter, stability is not an issue. However, *selection of a proper block size, windowing the data in the block etc.* may be addressed for better estimation of the autocorrelation functions. It will be also interesting to determine *the threshold on the prediction error by local statistical analysis of the image*. Such a procedure may improve the impulse-noise detection performance of the LPE-based method. Linear prediction may be implemented *adaptively* to take into account the nonstationarity of the image signal.

(5) It will be interesting to apply *LPE* based detection simultaneously with other detection schemes like the one based on rank-conditioning.

Bibliography

[1] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principle and Applications*, Boston Kluwer, 1990.

[2] J. Astola and P. Kousmanen, *Fundamental of Nonlinear Filtering*, CRC Press, 1997.

[3] S. K. Mitra and G. L. Sicuranza, *Nonlinear Image Processing*, S.K. Mitra and G.L. Sicuranza (Eds.), Academic Press, 2000.

[4] J. W. Tukey, *Exploratory Data Analysis*, Addison-Wesley, Mento Park, 1977.

[5] W. K. Pratt, *Digital Image Processing*, John Wiley and Sons, 1978.

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

[7] O. Yli-Harja, J. Astola and Y. Neuvo, "Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation", *IEEE Trans. on Signal Processing*, vol. SP-39, no. 2, pp. 395-410, Feb. 1991.

[8] T. Loupos, W. N. McDicken and P. L. Allan, "An adaptive weighted median filter for speckle suppression in medical ultrasonic images", *IEEE Trans. on Circuits Syst*., II , vol. 36, pp. 396-399, Jan. 1989.

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

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

[11] H. Hwang and R. A. Haddad, "Adaptive median filters: new algorithms and results," *IEEE Trans. Image Process*.,vol.4, no. 4, pp. 499-503, April 1995.

[12] L. Alparone, S. Baronti and R. Carla, "Two-dimensional rank-conditioned median filter", *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 42, no. 2, pp. 130-132, Feb., 1995.

[13] R. C. Hardie and K. E. Barner, "Rank-conditioned rank selection filters for signal restoration", *IEEE Trans. on Image Processing*, vol. 2, no. 2, pp. 192-206, Mar. 1994.

[14] A. Flaig, K. E. Barner and G. R. Arce, "Fuzzy ranking: theory and applications", *Elsevier Signal Processing*, vol. 80, issue 6, pp. 1017-1036, 2000.

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

[16] M. A. Moore and S, K. Mitra, "Performance analysis of the two-state signal-dependent rank order mean filter", *SPIE Proceedings - Nonlinear Image Processing X*, Edward R. Dougherty (Ed.), vol. 3646, pp. 56 – 66, 1999.

[17] T. Chen and H. R. Wu, "Adaptive impulse detection using center-weighted median filter", *IEEE Signal Processing Letters*, vol. 8, no. 1, pp. 1-3, Jan. 2001.

[18] Kh. Manglem Singh and Prabin K. Bora, "Multi-stage detection based filtering for removal of impulse noise from images", *IETE Journal of Research*, vol..5, no.2, pp. 125-131, March-April, 2004.

[19] Kh. Manglem Singh and Prabin K. Bora, "Improved rank conditioned median filter for removal of impulse noise from images", *IEEE Proc. Tencon International Conference*, pp. 557 – 560, Beijing, Oct. 2002.

[20] Kh. Manglem Singh, Prabin K. Bora and S. Birendra Singh, "Rank-ordered mean filter for removal of impulse noise from images", *IEEE Proc. ICIT*, Bangkok, pp. 980-985, 11-14 Dec. 2002.

[21] D. A. Florencio and R. W. Schafer, "Decision-based median filter using local signal statistics", *Proc. SPIE Symp. Visual Comm. Image Processing*, vol. 2038, pp. 268-275, Sep. 1994.

[22] T. Chen, K.-K. Ma and L. -H. Chen, "Tri-state median filter for image denoising", *IEEE Trans. on Image Processing*, vol. 8, pp. 1834-1838, Dec. 1999.

[23] F. Russo and G. Ramponi, "A fuzzy filter for images corrupted by impulse noise", *IEEE Signal Processing Letters*, vol. 3, no. 6, pp. 1996-1998, June 1996.

[24] H. –L. Eng and K. –K. Ma, "Noise adaptive soft-switching median filter", *IEEE Trans. Image Process*, vol. 48, no. 8, pp. 784-789, Aug. 2001.

[25] G. Pok, J. –C. Liu and A. S. Nai, "Selective removal of impulse noise based on homogeneity level information", *IEEE Trans. Image Process*., vol. 12, no.1, pp. 85-92, Jan. 2003.

[26] Z. Wang and D. Zhang, "Progressive switching median filter for the removal of impulse noise from highly corrupted images", *IEEE Trans. on Circuits and Systems - II Analog and Digital Signal Processing*, vol. 46, no. 1, pp. 78-80, Jan. 1999.

[27] S. Zhang and M. A. Karim, "A new impulse detector for switching median filter", *IEEE Signal Processing Letters*, vol. 9, no. 11, pp. 360-363, Nov. 2002.

[28] T. Sun and Y. Neuvo, "Detail-preserving median filters in image processing", *Pattern Recognit. Lett*. vol. 15, pp. 341-347, 1994.

[29] I. Pitas, and A. N. Venetsanopoulos, "Order statistics in digital image processing", *IEEE Proc.*, no. 80, pp.1893-1919, 1992.

[30] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters", *Proc. IEEE*, *v*ol. 78, pp. 678-689, 1990.

[31] V. Barnet, "The Ordering of Multivariate Data (with Discussion)", *J. Royal Statistical Society*, vol. 139, pp. 318-354, 1976.

[32] I. F.L. Cheikh, R. Hamila, M. Gabbouj and J. Astola, "Impulse noise removal in highly corrupted color images", *IEEE Proc. ICIP*, vol. 1, pp. 997-1000, 1996.

[33] M. A. Moore, M. Gabbouj and S. K. Mitra, "Vector SD-ROM filter for removal of impulse noise from color images", *Proc. Eurasip Conference, ECMCS*, 1999.

[34] J. Gangyi, Y. Mei and Y. Bokang, "A new method for adaptive color image filtering", *Chinese Science Bulletin*, vol. 45, no. 13, pp. 1230-1235, 2000.

[35] K.N. Plataniotis and A.N. Venetsanopoulos, Color Image Processing and Applications, *Springer-Verlag*, 2000.

[36] P. Trahanias and A. Venetsanopoulos, "Vector directional filters – A new class of multichannel image process filters", *IEEE Trans. on Image Processing*, vol. 2,

no. 4 pp. 528-534, 1993.

[37] D. Karakos and P. Trahanias, "Combining vector median and vector directiona filters: the directional distance filters", *IEEE Proc. ICIP*, vol. 1, pp.171-174, 1995.

[38] T. Viero, K. Oistamo and Y. Neuvo, "Three-dimensional median-related filters for color image sequence filtering", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, no. 2, pp.129-142, 1994.

[39] R. Lusac and B. Smolka, "Application of the center-weighted vector median framework for the enhancement of cDNA microarray images", *AMCS Int. Jour. Appl. Math. Sc.*, vol. 13, no. 3, pp. 369-383, 2003.

[40] L. Alparone, M. Barni, F. Bartolini and V. Cappellini, "Adaptive weighted vector median filters for motion-fields smoothing", *Proc. IEEE ICASSP*, pp. 771-776, 1996.

[41] Kh. Manglem Singh, Prabin K. Bora and S. Birendra Singh, "Vector median filter for removal of impulse noise from color images", *WSEAS Trans. Computers*, vol. 3, no. 1, ISSN 1109 – 2750, pp. 209-217, Jan. 2004.

[42] Kh. Manglem Singh, Prabin K. Bora and S. Birendra Singh, "Vector median filter for removal of impulse noise from color images", CD version, *Proc. WSEAS*, Singapore 2003.

[43] Kh. Manglem Singh, Prabin K. Bora and S. Birendra Singh, " Enhanced vector SD-ROM filter for removal of impulse noise from color images", *Proc. International Conference on HPC*, Bangalore, Dec. 2002.

[44] Kh. Manglem Singh and Prabin K. Bora, "Adaptive vector median filter for removal impulses noise from color images", *Proc. IEEE ISCAS*, pp. 396-399, vol. 2, Bangkok, May 2003.

[45] I. Pitas and A. N. Venetsanopulos, " Order statistics of digital image processing", *Proc. IEEE*, vol. 80, issue 12, pp. 1893-1921, Dec. 1992.

[46] I. Pitas and P. Tsakalides, "Multivariate ordering in color image filtering", *IEEE Trans. on Circuits and Systems for Video Technology*, vol.1, no. 3, pp. 247-259, 1991.

[47] M. Szczepanski, B. Smolka, K. N. Plataniotis and A.N. Venetsanopoulos "On distance function approach to color image enhancement", *Elsevier*, Discrete Applied Mathematics, pp. 283-305, 139(2004).

[48] K. N. Plataniotis, D. Androutsos and A.N. Venetsanopoulos, "Adaptive fuzzy systems for multichannel signal processing", *IEEE Proc*., vol. 87, no. 9, pp. 1601-1622, 1999.

[49] G. Louverdis, I. Andreadis and P. Tsalides, "New fuzzy model for morphological colour image processing", *IEE Proc.-Vis Image Signal Process*., vol. 149, no. 3, pp. 129-139, 2002.

[50] K. N. Plataniotis and A.N. Venetsanopoulos, "Vector filtering", in the colour image processing handbook, S. J. Sangwine and R. E. N. Horne Eds., pp. 188-209, *Chapman and Hall*, London U. K, 1998.

[51] T. Viero, K. Oistama and Y. Neuvo, "Three-dimensional median-related filters for color image sequence filtering", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, no. 2, pp. 129-142, April 1994.

[52] S. David and B. Ramamurthi, "Two-sided filters for frame-based prediction", *IEEE Trans. on Signal Processing*, vol. 39, pp. 789-794, 1991.

[53] J. –J. Hsue and A. E. Yagle, "Fast algorithms for close-to-Toeplitz plus Hankel systems and two-sided linear prediction", *IEEE Trans. on Signal Processing*, vol. 41, no. 7, pp. 2349-2361, Jul., 1993.

[54] N. Balram and M. F. Moura, "Noncausal predictive image codec", *IEEE Trans. on Image Processing*, vol. 5, no. 8, pp. 1229-1242, Aug., 1996.

[55] J. –J Hsue and A. E. Yagle, "Blind deconvolution of symmetric noncausal impulse response using two-sided linear prediction", *IEEE Trans. on Signal Processing*, vol. 42, no. 6, pp. 1509-1518, Jun., 1994.

[56] B. I. Justusson, "Median filtering: statistical properties", *Two-dimensional Digital Signal Processing*, T. S Huang (Ed.), pp. 161-196, Springer Verlag, 1981.

[57] R. C. Gonzalez and R. E. *Woods, Digital Image Processing*, Pearson Education Asia, 2nd edition, 2000.

[58] E. R. Dougherty, Random Processes for image and Signal Processing, *Prentice Hall of India*, 2003.

[59] N. C. Gallagher, Jr and G. L. Wise, "A theoretical analysis of the properties of median filters", *IEEE Trans. on Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 1136-1141, Dec. 1981.

[60] G. R. Arce and N. C. Gallagher, Jr, "State description of the root signal set of median filters", *IEEE Trans. on Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 894- 902, Oct. 1982.

[61] F. Kuhlmann and G. L. Wise, "On spectral characteristics of median filtered independent data", *IEEE Trans. on Commun.*, vol. COM-29, no. 9, pp. 1374-1380, Sept. 1981.

[62] E. Ataman, V. K. Aatre and K. M. Wong, "Some statistical properties of median filters", *IEEE Trans. on Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 1073-1075, Oct. 1981.

[63] T. A. Nodes and N. C. Gallagher, Jr, "The output distribution of median type filters", *IEEE Trans. on Commun*., vol. COM-32, pp. 532-541, May 1984.

[64] G-Y. Liao, T. A. Nodes and N. C. Gallagher, Jr, "Output distributions of two-dimensional median filters", *IEEE Trans. on Accoust., Speech and Signal Processing*, vol. ASSP-33, no. 4, pp. 1280-1295, 1985.

[65] T. S. Huang, G. J. Tang and G.Y. Tang, "A fast two-dimensional median filtering algorithm", *IEEE trans. On Accoustic, Speech and Signal Processing,* vol. ASSP 27, no. 1, pp. 13-18, Feb., 1979.

[66] A. Koschan and A. Abidi, "A comparison of median filter techniques for noise removal in color images", *Proc. 7th German Workshop on Color Image Processing*, pp. 69-79, Oct. 2001.

[67] I. Pitas, "Marginal order statistics in color image filtering," Optical Engineering, vol. 29, no. 95, pp. 493-503, 1990.

[68] M. Barni, "A fast algorithm for 1-norm vector median filtering", *IEEE Trans. on Image Processing*, vol. 6, no. 10, pp.1452-1455, Oct., 1997.

[69] M. Barni, F. Bartolini, F. Buti and V. Cappellini, "Optimum linear approximation of the Euclidean norm to speed up vector median filtering", *IEEE Proc. Image Processing*, pp. 362-365, Oct., 1995.

[70] M. Barni and V. Cappellini, "On the computational complexities of multivariate median filters", *Elsevier Signal Processing*, vol. 71, issue 1, pp. 45-54, Nov., 1998.

[71] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Englewood Cliffs, *N.J.*, *Prentice Hall*, 1984.

[72] A. Gersho, "Advances in speech and audio compression", *IEEE Proc.*, vol. 82, pp. 900-918, Jun., 1994.

[73] J. Makhoul, "Linear prediction: a tutorial review", *IEEE Proc.*, vol. 63, pp. 561-580, 1975.

[74] L .R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Englewood Cliffs, Prentice Hall, 1978.

[75] J. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications*, Englewood Cliffs, Prentice Hall, 1996.

[76] M. R. Schroeder and B. S. Atal, "Code-excited linear prediction (CELP): High quality speech at very low bit rates", *IEEE Proc. ICASSP*, pp. 937-940, 1985.

[77] J. D. Markel and A. H. Gray Jr., *Linear Prediction of Speech*, Springer-Verlag, 1976.

[78] J. Makhoul and J. Wolf, "Linear prediction and spectral analysis of speech", *BBN Report*, no. 2304, Aug., 1972.

[79] B. S. Atal and S .L. Hanauer, "Speech analysis and synthesis by linear prediction of speech wave", *Journal of Acoustical Society of America*, vol. 50, pp. 637-655, 1971.

[80] H. Harmansky, "Perceptual linear predictive (PLP) analysis of speech", *Journal of Acoustical Society of America*, vol. 4, pp. 1738-1752, 1990.

[81] J. Makhoul, "Spectral linear prediction: properties and applications", *IEEE Trans. on Acoust., Speech and Signal Processing*, ASSP-92, vol. 1, pp. 637-639,1975.

[82] B. S. Atal, "Linear prediction of speaker identification", *Journal of Acoustical Society of America*, vol. 55, no. 6, pp. 1354-1311, Jun., 1974.

[83] E . M. Hofstetter, "An introduction to the mathematics of linear prediction filtering as applied to speech analysis and synthesis", *Technical Note* 1973-36, MIT Lincoln Lab., Jul., 1973.

[84] S. R. Kim and A. Afron, "Adaptive robust impulse noise filtering", *IEEE Trans. on Signal Processing*, vol. 43, no. 8, pp. 1855-1866, Aug., 1995.

[85] A. K. Jain, *Fundamental of Digital Image Processing*, Prentice Hall, Information and System Science Series, 1989.

[86] A. K. Jain, "Image coding via a nearest neighbor image model", *IEEE Trans. on Communication*, vol. COM-23, no. 3, pp. 318-331, Mar., 1975.

[87] A. K. Jain, "Image data compression: a tutorial review", *IEEE Proc.*, vol. 69, no. 3, pp. 349-389, Mar., 1981.

[88] A. K. Jain, "Advances in mathematical models of image processing", *IEEE Proc.*, vol. 69, no. 5, pp. 502-527, May 1981.

[89] S. Raghunath and A. K. Jain, "Two-dimensional linear prediction models, part I: Spectral factorization and realization", *IEEE Trans. on Acoust., Speech, Signal Processing*, vol. 33, no. 1, pp. 280-299, Feb., 1985.

[90] W. Fang and A. E. Yagle, "Two-dimensional linear prediction and spectral estimation in polar raster", *IEEE Trans. on Signal Processing*," vol. 42, no. 3, pp. 628-641, May 1994.

[91] P. A. Maragos, R. W. Schafer and R. M. Mersereau, "Two-dimensional linear prediction and its application to adaptive prediction coding", *IEEE Trans. on Acoust., Speech and Signal Processing*, vol. 22, pp. 1213-1229, Dec., 1984.

[92] D. G. Ghost and L. Torres, "Efficient coding of homogeneous textures using stochastic vector quantization and linear prediction", *Proc. IEE -Vision, Image and Signal Processing*, vol. 146, no. 3, pp. 151-158, Jan., 1992.

[93] R. A. Wiggins and E. A. Robinson, "Recursive solution to multichannel filtering problem", *Journ. Geophysics Res.*, vol. 70, pp. 1885-1891, Apr., 1965.

[94] S. V. Vaseghi, *Advanced Digital signal Processing and Noise Reduction*, John Wiley and Sons, 2000.

[95] A. Asif and J. M. F. Moura, "Image codec by noncausal prediction residual mean removal, and cascaded VQ", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, no. 1, pp. 42-55, Feb., 1996.

[96] A. Asif, "Fast Rauch-Tung-Striebel smoother-based image restoration for noncausal images", *IEEE Signal Processing Letters***,** vol. 11 **,** no. 3 , pp. 371 – 374, Mar. 2004.

[97] J. –H. Hu, Y Wang and P. T. Cahill, "Multichannel code excited linear prediction coding and its application in magnetic resonance images", *IEEE Trans*. *on Image Processing*, vol. 6, no. 11, pp. 1155-15566, Nov, 1997.

[98] J. Hu, Y. Wang and P. Cahill, "Segmentation adaptive DPCM for lossy compression of multispectral MR images", *J. Vis. Commun. Image Represent*., vol. 8, pp. 69-82, Mar. 1997.

[99] Kh. Manglem Singh and Prabin K. Bora, "2-D linear prediction based median filtering", Proc. *of National Conference on Comm*., IIT, pp. 1-5, Bombay, Dec. 2001.

[100] Kh. Manglem Singh and Prabin K. Bora, "Noncausal vector linear prediction filters", *Proc. WSEAS ISTASC*, Rhodes Island, CD version, November 2003.

[101] Kh. Manglem Singh and Prabin K. Bora, "Noncausal vector linear prediction filter", *WSEAS Trans. on Circuits and Systems*, vol. 2, no 4, ISSN 1109-2734, pp. 1211-1219, Oct. 2003.

Appendix A

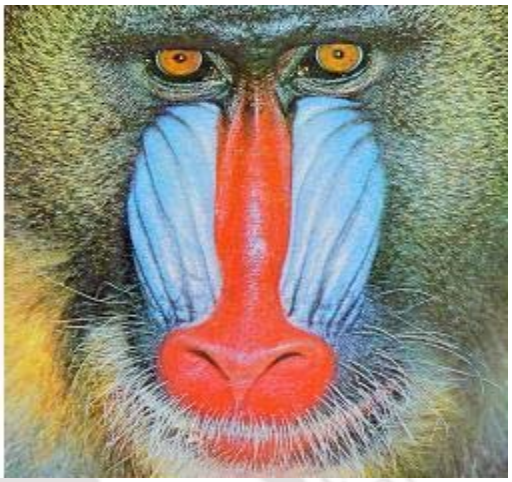
Test Image Database

A.1 Image Description

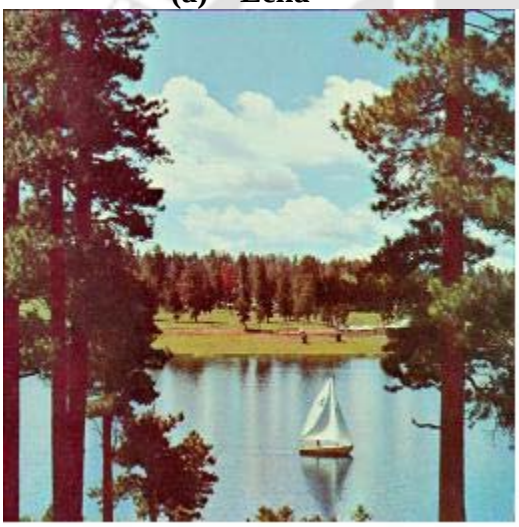
Twenty two test images were used for experimentation in our thesis and thirteen test images (Lena, Mandrill, Lake, Pepper, House, Bridge, Miramar, House, Point Loma, Woodland Hill, Airplane, Texmos1 and Texmos3) were collected from the image database of *the Signal and Image Processing Institute, the University of Southern California.* The images taken from all four categories specified in the database: (I) Miscellaneous image, (II) Aerial image, (III) Texture image and (IV) Medical image. The images are shown below.

A.2 Images

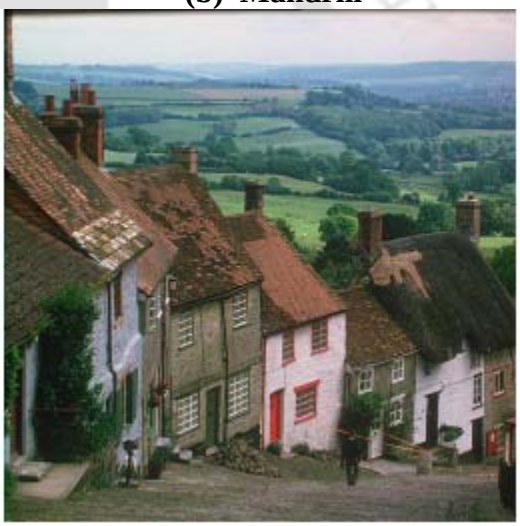
(a) Lena



(b) Mandrill



(c)Lake



(d) Gold Hill

(e) Brain

(f) Golden Gate

(g) Tree

(h) Tulip

(i) Cat

(j) Pepper

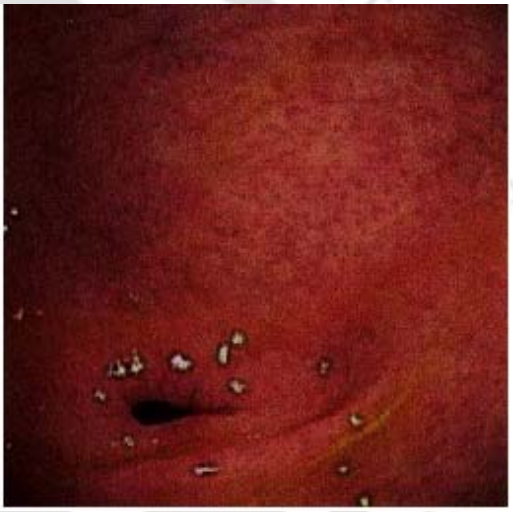
(k) House

(l) Bridge

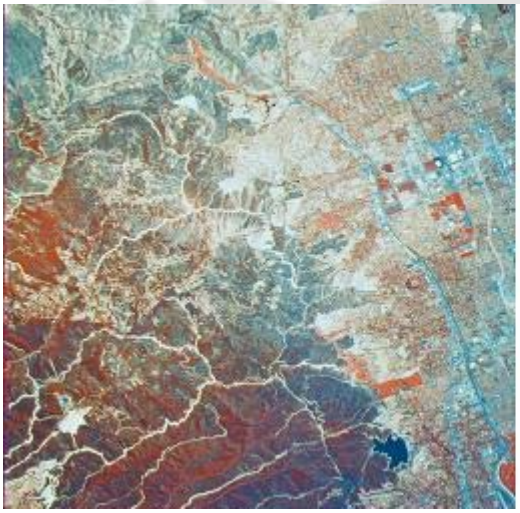
(m) Miramar

(n) Point Loma

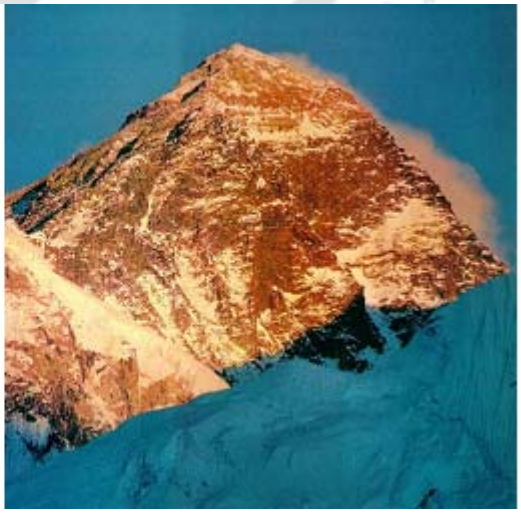
(o) Hut



(p) Stomach

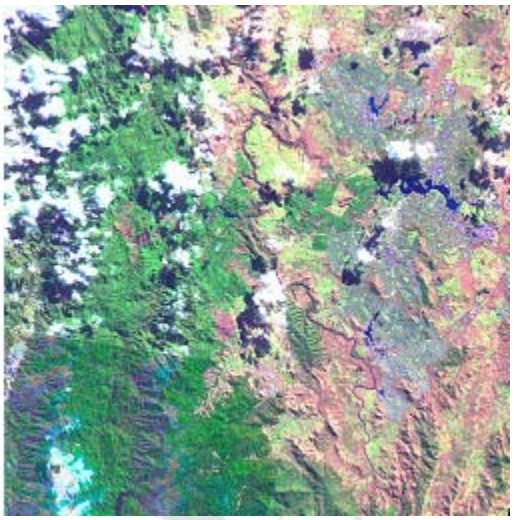


(q) Woodland Hill

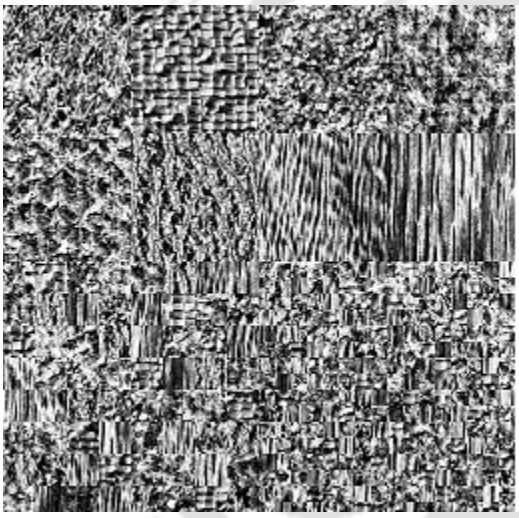


(r) Everest

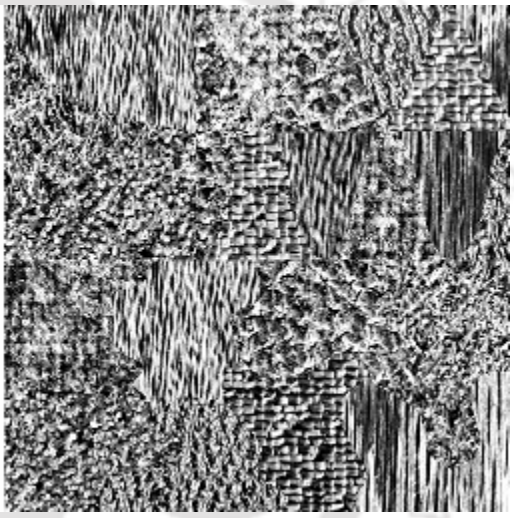
(s) Airplane



(t) Terrain



(u) Texmos1



(v) Texmos3

Figure A.1 Images

Appendix - B

Linear Prediction Coefficients

B.1 Order of Prediction

The image grid can be defined as a rectangular lattice. Consider a rectangular lattice $G = \{(i, j) : 0 < i \leq X, 0 < j \leq Y\}$ over which the digital image is defined. A *neighbourhood set* can be defined at a site $(m, n) \in G$ with the points lying within a specified radius as its elements. Using the Euclidian distance for defining the radius,

a ηth order neighbourhood set of site (m,n) is defined as

$$S^{\eta}_{(m,n)} = \left\{(i,j) : 0 < (m-i)^2 + (n-j)^2 \le D_{\eta}\right\} \text{ with } (i,j) \neq (m,n), \quad \text{(B.1)}$$

where D_{η} is an increasing integer function of η. Consider the following examples.

1) 1st order neighbourhood set: $\eta = 1; D_{\eta} = 1;$

$S^{1}_{(m,n)} = \{(m-1,n),(m+1,n),(m,n-1),(m,n+1)\}.$

2) 2nd order neighbourhood set: $\eta = 2; D_{\eta} = 2;$

$S^{2}_{(m,n)} = \{(m-1,n),(m+1,n),(m,n-1),(m,n+1),(m-1,n-1),(m+1,n+1),(m-1,n+1),(m+1,n-1)\}.$

Similarly, the 3rd order and 4th order neighbourhood sets are defined with $\eta = 3, D_{\eta} = 4$ and $\eta = 4, D_{\eta} = 5$ respectively.

These neighbourhood sets can be used for defining the neighbouring pixels taking part in two-dimensional linear prediction. The neighbourhood sets up to order 9 for noncausal linear prediction is shown in Figure B.1. The pixel to be predicted is labeled '0'. For first-order prediction, the neighbourhood set consists of pixels marked '1', for second-order prediction, the neighbourhood set involves pixels marked as '1' and '2' and so on.

9	8	7	6	7	8	9
8	5	4	3	4	5	8
7	4	2	1	2	4	7
6	3	1	0	1	3	6
7	4	2	1	2	4	7
8	5	4	3	4	5	8
9	8	7	6	7	8	9

Figure B.1: The neighbourhood sets for first-order to 9th-order noncausal linear prediction

B.2 Linear Predictor for Gray-scale Images

The first-order and the secod-order linear prediction for gray-scale images are discussed below.

B.2.1 First-order Causal Linear Predictor for Gray-scale Images

The first-order causal linear prediction $\hat{x}(m.n)$ of the centre pixel $x(m,n)$ is given by

$$\begin{aligned}\hat{x}(m.n) &= a_1(0,1)x(m,n-1) + a_1(1,0)x(m-1,n) \\ &= \boldsymbol{x}_1\boldsymbol{a}_1\end{aligned} \tag{B.2}$$

where $\boldsymbol{x}_1 = [x(m,n-1)\ \ x(m-1,n)]$ represents gray-scale pixels in the prediction window and $\boldsymbol{a}_1 = [a_1(0,1)\ \ a_1(1,0)]'$. The prediction coefficients $a_1(0,1)$ and $a_1(1,0)$ are related with the autocorrelation functions by the following normal equation in the matrix form:

$$\boldsymbol{R}_{\boldsymbol{x}_1}\boldsymbol{a}_1 = \boldsymbol{r}_{\boldsymbol{x}_1} \tag{B.3}$$

where

$$\boldsymbol{R}_{\boldsymbol{x}_1} = \begin{bmatrix} R_x(0,0) & R_x(1,-1) \\ R_x(1,-1) & R_x(0,0) \end{bmatrix}$$

$$\boldsymbol{r}_{x_1} = \begin{bmatrix} R_x(0,1) \\ R_x(1,0) \end{bmatrix}$$

B.2.2 First-order Noncausal Linear Predictor for Gray-scale Images

The first-order noncausal linear prediction of predicted value of the centre pixel $x(m,n)$ is given by

$$\begin{aligned}\hat{x}(m.n) &= a_1(0,1)\{x(m,n-1) + x(m,n+1)\} + a_1(1,0)\{x(m-1,n) + x(m+1,n)\} \\ &= \boldsymbol{x}_1\boldsymbol{a}_1\end{aligned} \tag{B.4}$$

where $\boldsymbol{x}_1 = [x(m,n-1)+x(m,n+1)\;\; x(m-1,n)+x(m+1,n)]$ represents the gray-scale pixels in the prediction window and $\boldsymbol{a}_1 = [a_1(0,1)\;\; a_1(1,0)]'$ is the vector formed by prediction coefficients. The normal equation in the matrix form is given by

$$\boldsymbol{R}_{\boldsymbol{x}_1}\boldsymbol{a}_1 = \boldsymbol{r}_{\boldsymbol{x}_1} \tag{B.5}$$

where

$$\boldsymbol{R}_{\boldsymbol{x}_1} = \begin{bmatrix} R_X(0,0)+R_X(0,2) & R_X(1,-1)+R_X(1,1) \\ R_X(1,-1)+R_X(1,1) & R_X(0,0)+R_X(2,0) \end{bmatrix}$$

$$\boldsymbol{r}_{x_1} = \begin{bmatrix} R_X(0,1) \\ R_X(1,0) \end{bmatrix} \tag{B.5}$$

B.2.3 Second-order Causal Linear Predictor for Gray-scale Images

The second-order causal linear prediction of the centre pixel $x(m,n)$ is given by

$$\begin{aligned}\hat{x}(m.n) &= a_2(0,1)x(m,n-1)+a_2(1,0)x(m-1,n)+a_2(1,1)x(m-1,n-1)+a_2(1,-1)x(m-1,n+1)\\ &= \boldsymbol{x}_2\boldsymbol{a}_2\end{aligned} \tag{B.6}$$

where $\boldsymbol{x}_2 = \begin{bmatrix} x(m,n-1) & x(m-1,n) & x(m-1,n-1) & x(m-1,n+1) \end{bmatrix}$ represents the gray-scale pixels in the prediction window and $\boldsymbol{a}_2 = \begin{bmatrix} a_2(0,1) & a_2(1,0) & a_2(1,1) & a_2(1,-1) \end{bmatrix}'$ is the vector formed by the prediction coefficients in a second-order causal linear prediction. The corresponding normal equation is given by

$$\boldsymbol{R}_{\boldsymbol{x}_2}\boldsymbol{a}_2 = \boldsymbol{r}_{\boldsymbol{x}_2} \tag{B.7}$$

where

$$R_{\boldsymbol{x}_2} = \begin{bmatrix} R_X(0,0) & R_X(1,-1) & R_X(1,0) & R_X(1,-2) \\ R_X(1,-1) & R_X(0,0) & R_X(0,1) & R_X(0,1) \\ R_X(1,0) & R_X(0,1) & R_X(0,0) & R_X(0,2) \\ R_X(1,-2) & R_X(0,1) & R_X(0,2) & R_X(0,0) \end{bmatrix}$$

$$r_{x_2} = \begin{bmatrix} R_x(0,1) \\ R_x(1,0) \\ R_x(1,1) \\ R_x(1,-1) \end{bmatrix}$$

B.3 Linear Predictor for colour Images

The first-order and the secod-order linear prediction for colour images are discussed below. Suppose $\boldsymbol{x}(m,n) = [x_R(m,n)\ x_G(m,n)\ x_B(m,n)]'$ is the centre vector pixel.

B.3.1 First-order Causal Linear Predictor for Colour Images

The causal first-order linear prediction $\hat{\boldsymbol{x}}(m,n)$ at the location (m,n) in the prediction window using the constrained intrachannel model is given by

$$\begin{aligned} \hat{\boldsymbol{x}}(m,n) &= a_1(0,1)\boldsymbol{x}(m,n-1) + a_1(1,0)\boldsymbol{x}(m-1,n) \\ &= \boldsymbol{X}_1\boldsymbol{a}_1 \end{aligned} \tag{B.8}$$

where $\boldsymbol{X}_1 = [\boldsymbol{x}(m,n-1)\ \ \boldsymbol{x}(m-1,n)]$ represents vector pixels in the prediction window and $\boldsymbol{a}_1 = [a_1(0,1)\ \ a_1(1,0)]'$ is the vector of prediction coefficients. The corresponding matrix normal equation is given by

$$\boldsymbol{R}_{X_1}\boldsymbol{a}_1 = \boldsymbol{r}_{X_1} \tag{B.9}$$

where

$$\boldsymbol{R}_{X_1} = \begin{bmatrix} R_X(0,0) & R_X(1,-1) \\ R_X(1,-1) & R_X(0,0) \end{bmatrix},$$

$$\boldsymbol{r}_{X_1} = \begin{bmatrix} R_X(0,1) \\ R_X(1,0) \end{bmatrix}$$

B.3.2 First-order Noncausal Linear Predictor for Colour Images

The predicted value $\hat{\boldsymbol{x}}(m,n)$ at the location (m,n) in the window for a first-order noncausal linear prediction filter for the RGB colour image is given by

$$\hat{\boldsymbol{x}}(m,n) = a_1(0,1)\{\boldsymbol{x}(m,n-1)+\boldsymbol{x}(m,n+1)\}+a_1(1,0)\{\boldsymbol{x}(m-1,n)+\boldsymbol{x}(m+1,n)\} \\ = \boldsymbol{X}_1\boldsymbol{a}_1 \tag{B.10}$$

where $\boldsymbol{X}_1 = [\boldsymbol{x}(m,n-1)+\boldsymbol{x}(m,n+1)\ \ \boldsymbol{x}(m-1,n)+\boldsymbol{x}(m+1,n)]$ represents vector pixels in the window and $\boldsymbol{a}_1 = [a_1(0,1)\ \ a_1(1,0)]'$ is the vector of the prediction coefficients. The normal equation is given by

$$\boldsymbol{R}_{X_1}\boldsymbol{a}_1 = \boldsymbol{r}_{X_1} \tag{B.11}$$

where

$$\boldsymbol{R}_{X_1} = \begin{bmatrix} R_X(0,0)+R_X(0,2) & R_X(1,-1)+R_X(1,1) \\ R_X(1,-1)+R_X(1,1) & R_X(0,0)+R_X(2,0) \end{bmatrix},$$

$$\boldsymbol{r}_{X_1} = \begin{bmatrix} R_X(0,1) \\ R_X(1,0) \end{bmatrix}$$

and

$$R_X(m,n) = R_{X_R}(m,n)+R_{X_G}(m,n)+R_{X_B}(m,n).$$

B.3.3 Second-order Causal Linear Predictor for Colour Images

The predicted value $\hat{\boldsymbol{x}}(m,n)$ at the location (m,n) in the window for second-order causal linear prediction for the RGB colour image is given by

$$\hat{\boldsymbol{x}}(m,n) = a_2(0,1)\boldsymbol{x}(m,n-1)+a_2(1,0)\boldsymbol{x}(m-1,n)+a_2(1,1)\boldsymbol{x}(m-1,n-1)+a_2(1,-1)\boldsymbol{x}(m-1,n+1) \\ = \boldsymbol{X}_2\boldsymbol{a}_2 \tag{B.12}$$

where $\boldsymbol{X}_2 = \begin{bmatrix} \boldsymbol{x}(m,n-1) & \boldsymbol{x}(m-1,n) & \boldsymbol{x}(m-1,n-1) & \boldsymbol{x}(m-1,n+1) \end{bmatrix}$ represents vector pixels in the window and $\boldsymbol{a}_2 = \begin{bmatrix} a_2(0,1) & a_2(1,0) & a_2(1,1) & a_2(1,-1) \end{bmatrix}'$ is the vector of prediction coefficients. The normal equations are given by

$$\boldsymbol{R}_{X_2}\boldsymbol{a}_2 = \boldsymbol{r}_{X_2} \tag{B.13}$$

where

$$\boldsymbol{R}_{X_2} = \begin{bmatrix} R_X(0,0) & R_X(1,-1) & R_X(1,0) & R_X(1,-2) \\ R_X(1,-1) & R_X(0,0) & R_X(0,1) & R_X(0,1) \\ R_X(1,0) & R_X(0,1) & R_X(0,0) & R_X(0,2) \\ R_X(1,-2) & R_X(0,1) & R_X(0,2) & R_X(0,0) \end{bmatrix}$$

and

$R_X(m,n) = R_{X_R}(m,n) + R_{X_G}(m,n) + R_{X_B}(m,n)$ as before.

New Developments in Median Filters

A PhD Thesis Synopsis

Submitted by

Khumanthem Manglem Singh

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINNERING

INDIAN INSTITUTE OF TECHNOLOGY, GUWAHATI

GUWAHATI – 781039, INDIA

July 2006

Among the numerous filters existing for reducing noise from images, the *median filter* is considered to be useful particularly for suppressing *impulse noise* while preserving edges and image details. Its performance is limited by the fact that it treats all observation samples equally regardless of their location within the observation window. Detection-based median filters address the issue of detecting impulse prior to the median filtering of the corrupted pixel. The objective of the present thesis is to develop new methods for detecting impulse noise in gray-scale and colour images. The thesis will comprise of six chapters outlined below.

Chapter 1: Introduction

The Chapter 1 of the thesis introduces the problem of impulse noise in images and median filtering to reduce this noise. A brief literature review of the modifications of median filtering for gray-level and colour images are presented.

The median filter is a nonlinear filter that takes the median of the data inside a moving window of pre-determined length. It performs well for heavy-tailed noise distributions and is appropriate to reduce impulse noise. The median based filter has been subject to growing interest since the discovery of the standard median (SM) filter in 1974 by Tukey [1] who applied it to the smoothing of time-series data. The median filter results in blurring of the image and destroys the fine image details such as thin lines and corners while reducing the noise. One way to overcome this drawback is the *weighted median (WM) filte*r [2], which gives more weight to some pixels within the window than others. The special case of the WM filter is the centre-weighted median (CWM) filter [3], which gives more weight only to the centre pixel of the window.

The median filter and its modifications are generally implemented to all pixels in an image. They tend to alter pixels undisturbed by noise and destroy image details in the cases where the noise ratio is high. As a result, their effectiveness in noise suppression is often at the expense of blurred and distorted image features. A better way to circumvent this drawback is to incorporate some decision-making processes in the filtering framework. At each pixel location, it is first determined whether the current pixel is contaminated. Filtering is applied to the pixel if it is detected as an impulse. A simple impulse detection filter is the *rank-conditioned median* (RCM) filter [4] in which pixels in the filtering window are ranked according to their magnitudes and the centre pixel is considered to be corrupted if it does not belong to the set of pixels obtained by trimming the extreme ranks. The corrupted pixels are replaced by the median values, while the noise-free pixels are left unaltered. Since not every pixel is filtered, undue distortion can be avoided. In the *signal-dependent rank-ordered mean* (SD-ROM) filter [5], all but the centre pixel

in a filtering window are considered for rank-ordering. An impulse is detected by thresholding the differences between the centre pixel and value of each of one half of the ranks. The detected impulse is replaced by the *rank-ordered mean*. The SD-ROM filter has been shown to work well in removing fixed-valued impulse noise and random-valued impulse noise. There are other decision based median filters and some of them are available in [6-9].

The simplest approach to the multichannel image filtering is to process each component independently by a scalar median filter and such type of the filter is known as the *marginal median filter* (MMF) [10]. Since each component is filtered independently, the filter output can combine to produce vector pixels not present in the original image. A more logical method is the *vector median filte*r (VMF) [11]. The principle is to map each vector into a scalar, rank the scalars and define the rank of the vectors according to the ranks of their corresponding scalars. The vector pixel with the lowest rank is the vector median. The sum of the angles between each vector pixel and other vector pixels in the window, rather than the sum of the distances, may also be considered for ranking. The *basic vector directional filte*r (BVDF) [12] works on this principle.

The vector median filter tends to alter pixels, undisturbed by noise as it is generally implemented on all vector pixels in an image. It modifies edges and image details in the cases where the noise ratio is high and results blurred and distorted image features. A simple but effective impulse detection filter is the *centre-weighted vector median filter* (CWVMF) [13], which emphasizes the centre vector pixel. It is an improvement over the *weighted vector median filter* [14]. In the *vector signal-dependent rank-ordered mean* (VSD-ROM) filter [15], all but the centre vector pixel in a filtering window are considered for rank-ordering. An impulse is detected by thresholding the differences between the centre vector pixel and each of the one half ranks. The detected impulse is replaced by the vector median.

Chapter 2: Impulse Noise and Median Filters

Chapter 2 presents the probabilistic models of the impulse noise and the mathematical formulation of the median filtering problem. Some important modifications of median filters are also described as a background to the algorithms developed in the following chapters.

An impulse noise may have fixed or variable amplitudes. The first type of impulse noise appears as black and/or white spot in gray-scale images and mainly arises owing to the saturation of the imaging sensors. Extreme values are digitised as minimum and maximum allowed values. Such fixed-valued impulse noise is called *salt-and-pepper noise*. The second type of impulse

noise, that has varying amplitudes, is more realistic and known as random-valued impulse noise. There are a number of models to represent impulse noise in images [16, 17]. The state representing occurrence of the impulse noise at a location is generally modelled by a Bernoulli process and the noise-corrupted pixel is modelled as one-sided impulse with a fixed value, two-sided impulse taking either of two fixed values with a fixed probability or a random variable following a particular probability distribution.

Modelling of impulse noise in a colour image presents a difficult task. Each channel of the colour image cannot be considered independently. An alternate model that considers the interdependence between the channels is available in [18]. We have proposed a colour impulse noise model that takes into account for the single, double and triple channel noise factors for a 3-channel image.

The performance of the median filter is analysed both from deterministic and probabilistic points of view. The probabilistic analysis can be done by considering probability distribution function of the median. In the deterministic analysis of median filters, considerable importance is given to a class of signals called the *root* signals that are invariant under median filtering. Non-roots of median filters that include ends of lines, corners etc. demand for modification of the median filtering algorithm.

Chapter 3: Rank-conditioned and Threshold Median Filters for Gray-scale Images

We propose to use detection schemes simultaneously to reduce the probability of detecting a healthy pixel as noisy. Three filters are developed by simultaneous application of two detection schemes. Each of these filters involves rank-conditioning and one of the thresholding conditions discussed later. In all three cases, the corrupted pixel is replaced by the rank-ordered mean (ROM).The filter-class based on these three detection schemes will be referred to as rank-conditioning and thresholding (RCT) filters.

Let $\boldsymbol{X}_{\mathrm{W}} = \{x_1, x_2, ..., x_N\}$ be a set of gray-scale pixels of N elements according to their position within an observation window of size $n \times n$ where $N = n^2$. The order statistics of the observed pixels form the set of rank-ordered pixels $\boldsymbol{X}_{\mathrm{R}} = \{x_{(1)}, x_{(2)}, ..., x_{(N)}\}$, where elements are arranged in ascending order**.** The median is the middle value in the set of rank-ordered pixels. The impulses, edges and image details usually occupy the left-most or right-most ranks, because their

values are either very small or very big. The order statistics $x_{(i)}$ and $x_{(N+1-i)}$ with $1 < i \leq \frac{N+1}{2}$ serve as the *trimming statistics* [19].

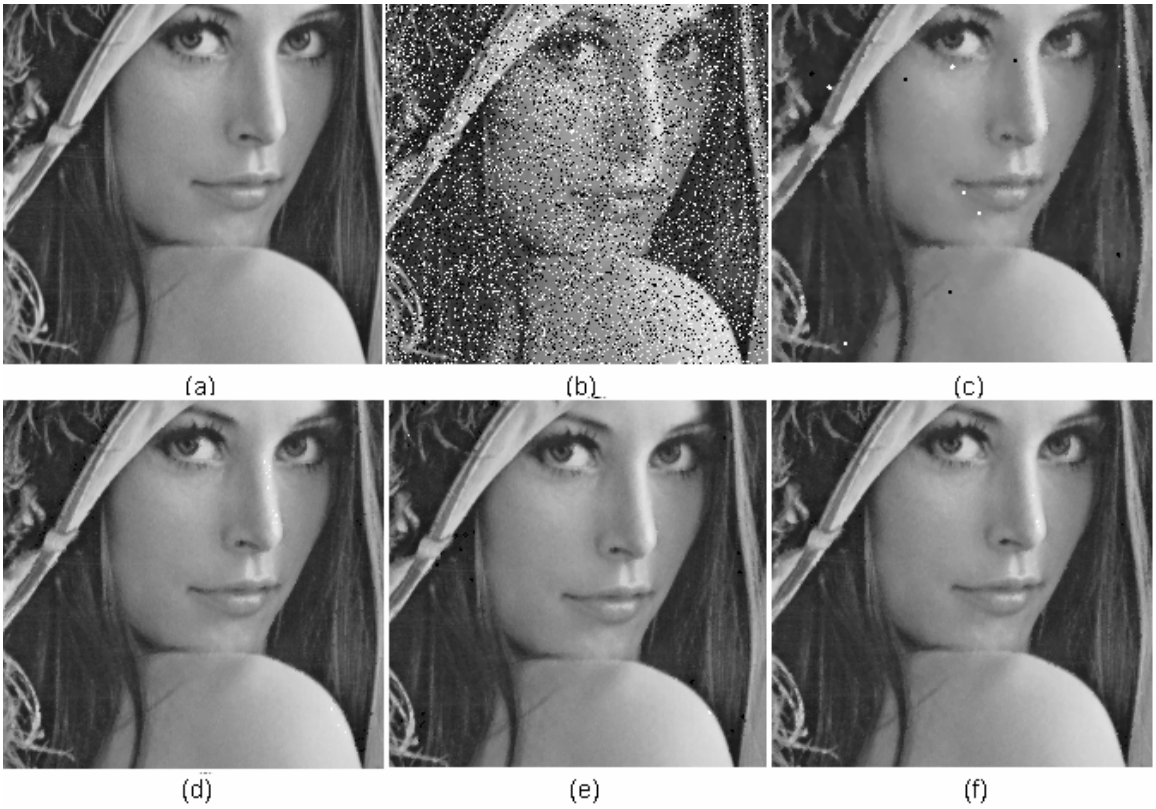


Figure 1 (a) Original image, (b) corrupted by 20% fixed-valued impulse noise, (c), (d), (e) and (f) the outputs of the SM filter, RCT-I, RCT-II and RCT-III respectively.

The *RCT-I filter* decides the centre pixel as corrupted by the impulse noise *if its rank does not lie inside the trimming range and the absolute difference between the centre pixel and the scalar median is bigger than a pre-defined threshold.* The impulse detection of the *RCT-II filter* is based on the rank-conditioned median filter [4] and the a*daptive centre weighted median filter* developed by Chen and Wu. [20]. It uses the differences $D_k = |x^{2k-1} - x|$, $k = 0,1,...,\frac{N-1}{2}$, where x^{2k-1} is the output of the centre-weighted median filter with a weight of $2k-1$. The centre pixel is declared corrupted *if its rank does not lie in the trimming range and* $D_k > T_k$ for any $k = 1,2,3,4$ *in a* 3×3 *window.* The multiple thresholds are obtained from the median absolute deviations (*MAD*) between pixels in the window and the scalar median.

The *RCT-III filter* is based on the rank-conditioned median filter [4] and signal-dependent rank-ordered mean (SD-ROM) filter [5]. The pixels in the window excluding the centre pixel are ranked and the differences between the centre pixel and each of one half of the ranks in the set of rank-ordered pixels are found. The centre pixel is corrupted *if its rank does not lie in the trimming range and any one of the differences is bigger than its corresponding pre-defined threshold.*

The filters have been tested on different images. The proposed the RCT-I, RCT-II and RCT-III filters outperformed the SM, the CWM, the RCM, the tri-state median, Chen and Wu's adaptive centre-weighted median and the SD-ROM filters in terms of PSNR values. The visual performance of the filters on one test image is shown in Figure 1.

Chapter 4: Rank-conditioned and Threshold Vector Median Filters for Colour Images

We have proposed a new filter called the *Rank-conditioned and Threshold Vector Median Filter (*RCTVMF*)* for removing impulse noise from colour images. The filter arranges the vector pixels in a window into a set of rank-ordered vector pixels on the basis of the sum of the distances of the vector pixel from all other neighbouring vector pixels in the window. The lowest-ranked vector pixel is the vector median [11] and the highest-ranked vector pixels may correspond to impulses, edges or image details. A trimming range for the vector pixels is defined by including ranks lower than a pre-defined value. The filter also computes the vector distance between the centre vector pixel and a pre-defined healthy vector pixel from the window, usually the vector median. *The centre vector pixel is corrupted by impulse noise if its rank does not lie in the trimming range and its difference from the vector median is bigger than a pre-defined threshold.* The corrupted vector pixel is replaced by the vector median. Excluding the thresholding condition, a weaker version called *the rank-conditioned vector median filter* (RCVMF) is also proposed. The RCTVMF has been tested on different colour images. The performance of the filter on one test image is shown in Figure 2. It outperformed the vector median filter based on $L_1 - \text{norm}$ (VMF_1), the vector median filter based on $L_2 - \text{norm}$ (VMF_2), the BVDF, the directional distance filter (DDF), the MMF, the CWVMF, the RCVMF and the VSD-ROM filter and the comparative *PSNR* performance (in dB) is shown in Figure 3.

Chapter 5: Noncausal Linear Prediction Error based Scalar and Vector Median Filters

Chapter 5 describes new algorithms for the detection of impulse noise from gray-scale and colour images based on the two-dimensional noncausal linear prediction [21] of the image signal. The colour image is divided into a number of analysis blocks of equal size and a vector pixel inside an analysis block is considered for prediction. The linear prediction $\hat{\boldsymbol{x}}(m,n)$ of the centre vector pixel $\boldsymbol{x}(m,n)$ is given by

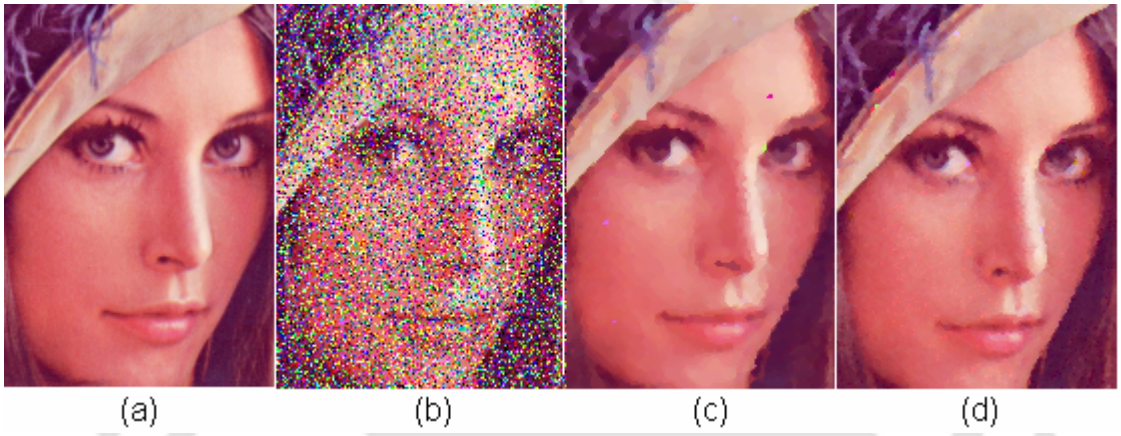


Figure 2 (a) Original image, (b) image corrupted with 60% fixed-valued impulse noise, (c) output of the VMF_1 and (d) output of the RCTVMF.

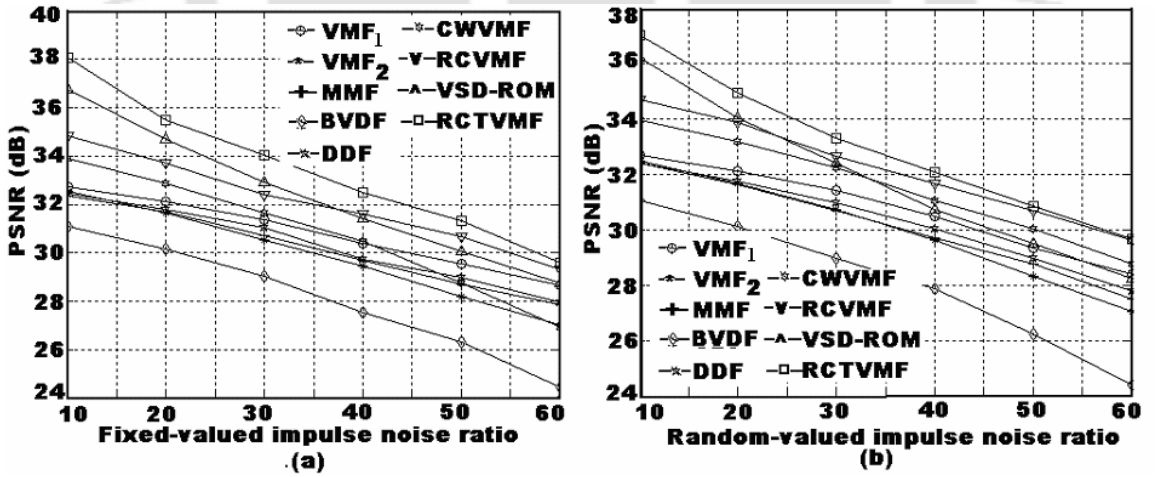


Figure 3 Noise ratio – *PSNR* curve for different filters: (a) removal of fixed-valued impulse noise and (b) removal of random-valued impulse noise.

$\hat{\boldsymbol{x}}(m,n) = \boldsymbol{X}_\eta \boldsymbol{a}_\eta$

where $\boldsymbol{X}_\eta$ is the $C \times M$ matrix of the components of the vector pixels used in prediction, C is the number of components of the vector pixel, M is the number of vector pixels used for prediction and $\boldsymbol{a}_\eta$ is the M-dimensional vector of prediction coefficients. We use the *constrained intrachannel predictor* [22] so that the same prediction coefficients are used for different channels. The vector $\boldsymbol{a}_\eta$ is chosen so as to minimize the mean-square error (*MSE*) $\boldsymbol{E}(\boldsymbol{x}(m,n) - \hat{\boldsymbol{x}}(m,n))'(\boldsymbol{x}(m,n) - \hat{\boldsymbol{x}}(m,n))$, where $\boldsymbol{E}$ is the expectation operato and $'$ denotes the transpose operation. The vector $\boldsymbol{a}_\eta$ is obtained form the normal equations in the matrix form given by

$$\boldsymbol{R}_\eta \boldsymbol{a}_\eta = \boldsymbol{r}_\eta$$

where $\boldsymbol{R}_\eta = E(\boldsymbol{X}_\eta' \boldsymbol{X}_\eta)$ is the autocorrelation matrix of the input vectors and $\boldsymbol{r}_\eta = E(\boldsymbol{X}_\eta' \boldsymbol{x}(m,n))$ is the cross-correlation of the input vectors and the vector pixel to be predicted.

The coefficient vector $\boldsymbol{a}_\eta$ is estimated from the estimates of the autocorrelation matrix $\boldsymbol{R}_\eta$ and the cross-correlation matrix $\boldsymbol{r}_\eta$. For the second-order noncausal prediction of an RGB colour images, only four prediction coefficients are required for eight input vector pixels, because $a(i, j) = a(-i, -j)$ due to symmetry property of the two-dimensional autocorrelation functions used in noncausal linear prediction.

The centre vector pixel is decided to be corrupted by impulse noise *if the* $L_\infty - norm$ *of the difference between the original centre vector and predicted one is bigger than a pre-defined threshold.* The corrupted vector pixel is replaced by the vector median. The filtering scheme is shown in Figure 3. The vector image is pre-filtered by the marginal median filter (MMF) for better prediction of the healthy vector pixel. The gray-scale image can be considered as a special case of the multichannel image with $C = 1$. In this case, the corrupted pixel is replaced by the median.

The proposed algorithms for colour and gray-scale images have been tested on different images. The performance of the algorithm on one test image is shown in Figure 4 and Figure 5. The proposed algorithms for gray-scale images outperformed the SM, the CWM, the RCM and Chen and Wu's adaptive centre weighted median filters. However its performance is inferior to that of the RCT-I, the RCT-II, the RCT-III, the SD-ROM and the TSM filters. The proposed second-order noncausal linear prediction error based vector median filter (2-NCLPVMF) for

colour images outperformed the VMF_1, VMF_1, MMF, BVDF, DDF, CWVMF, RCVMF, RCTVMF and VSD-ROM filter and the comparative *PSNR* performance on different colour images is shown in Table 1 and Table 2.

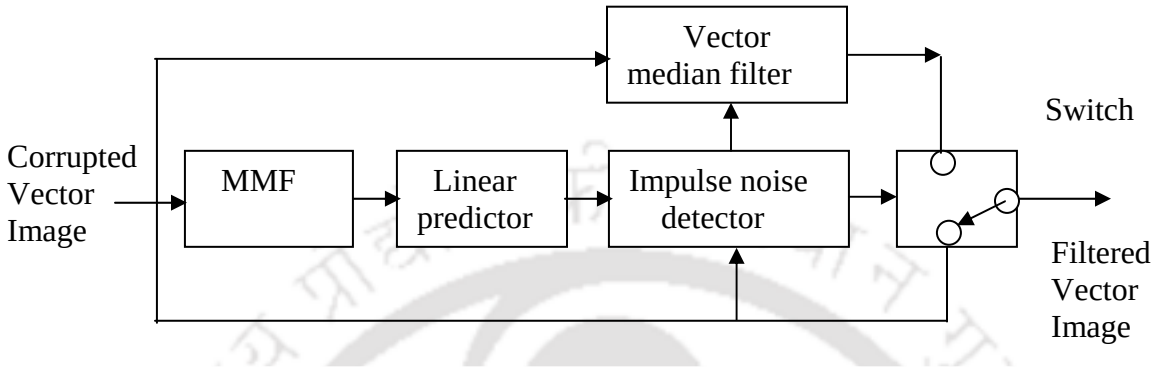


Figure 3 Block diagram of the non-causal linear prediction error based vector median filter.

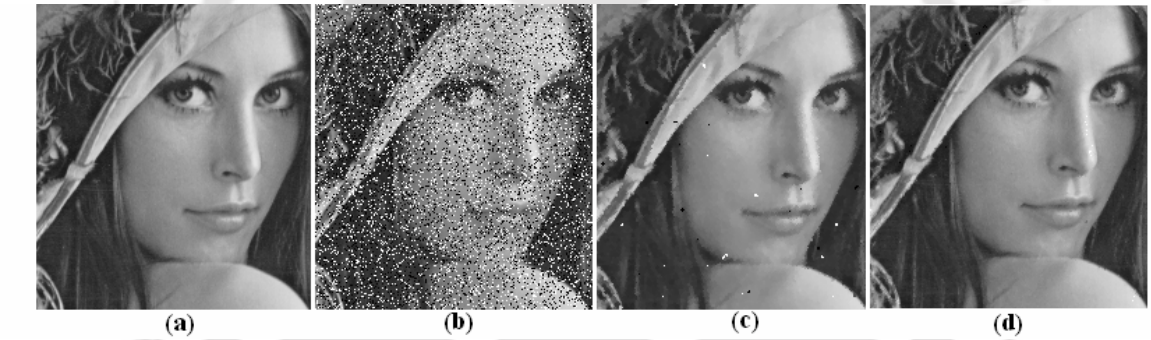


Figure 4 (a) Original image, (b) corrupted with 20% fixed-valued impulse noise, (c) output of the SM filter and (d) output of the second-order noncausal linear prediction error based median filter.

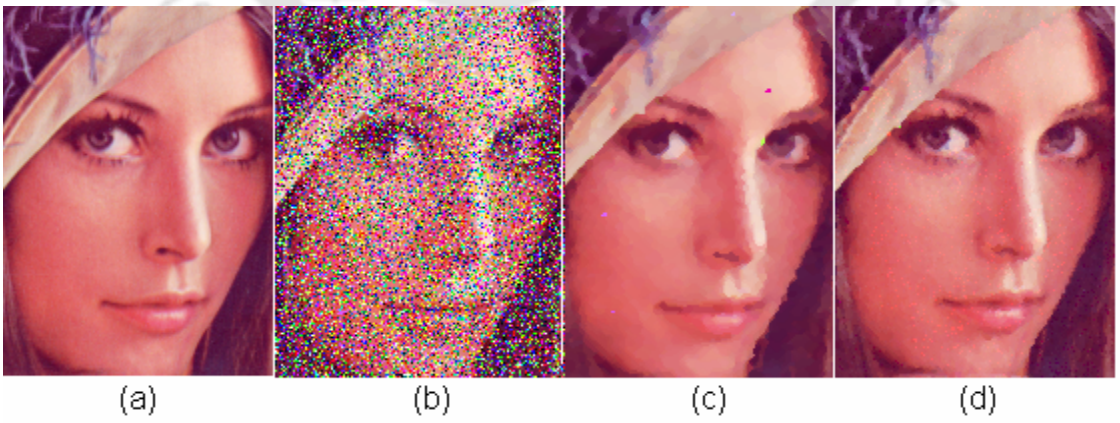


Figure 5 (a) Original image, (b) corrupted by 60% fixed-valued impulse noise, (c) output of the VMF_1 and (d) output of the 2-NCLPVMF.

Conclusions

Two classes of median filters with inbuilt impulse detection schemes have been proposed. They are

- Rank-conditioned and threshold scalar and vector median filters
- Two-dimensional noncausal linear prediction error based scalar and vector median filters

The RCT-II and RCT-III under the *Rank-conditioned and threshold median filters* outperformed a number of detection-based median filters under comparison. Similarly, the *rank-conditioned and threshold vector median filter* (RCTVMF) shows better performance than the vector median filter, the marginal median filter, the centre-weight vector median filter, the rank-conditioned vector median filter and the vector signal-dependent rank-ordered mean filter. The performance of the two-dimensional noncausal linear prediction error based vector median filters is found to be better than other filters for colour images including the rank-conditioned and threshold vector median filter.

Table 1 Comparison of performance (*PSNR*) of different filters in removal of impulse noise from colour images corrupted with 20% fixed-valued impulse noise

Images	VMF_1	MMF	VMF_2	BVDF	DDF	CW VMF	RC VMF	VSD-ROM	RCT VMF	2-NCLP VMF
Lena	32.14	31.68	31.63	30.14	31.80	32.87	33.69	34.70	35.48	36.10
Mandrill	22.67	22.22	22.51	21.19	22.64	23.61	23.57	23.35	25.07	25.99
Miramar	25.95	25.44	25.62	25.00	25.79	26.96	27.15	27.78	28.28	30.37
Texmos1	17.94	16.86	17.50	8.92	14.61	18.94	19.01	17.43	19.41	21.24
Brain	27.71	26.69	27.20	23.31	27.39	28.86	28.78	28.30	29.35	30.02

Table 2 Comparison of performance (*PSNR*) of different filters in removal of impulse noise from images corrupted with 20% random-valued impulse noise

Images	VMF_1	MMF	VMF_2	BVDF	DDF	CW VMF	RC VMF	VSD-ROM	RCT VMF	2-NCLP VMF
Lena	32.11	31.65	31.66	30.12	31.77	33.17	33.97	34.03	34.96	35.09
Mandrill	22.61	22.29	22.51	21.12	22.57	23.62	23.51	23.28	24.04	25.22
Miramar	25.92	25.41	25.70	25.02	25.82	26.99	27.14	27.62	28.22	29.07
Texmos1	17.66	16.76	17.37	9.32	14.72	18.79	18.80	17.24	19.22	20.51
Brain	27.54	26.63	26.82	23.81	27.46	28.29	28.76	28.01	29.13	28.97

References

[1] J. W. Tukey, *Exploratory Data Analysis*, Addison-Wesley, Mento Park, 1977.

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

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

[4] R. C. Hardie and K. E. Barner, "Rank-conditioned rank selection filters for signal restoration," *IEEE Trans. Image Processing*, vol. 2, no. 2, pp. 192-206, Mar. 1994.

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

[6] T. Chen, K.-K. Ma and L. -H. Chen, "Tri-state median filter for image denoising," *IEEE Trans. Image Processing*, vol. 8, pp. 1834-1838, Dec. 1999.

[7] F. Russo and G. Ramponi, "A fuzzy filter for images corrupted by impulse noise," *IEEE Signal Processing Letters*, vol. 3, no. 6, June 1996.

[8] Z. Wang and D. Zhang, "Progressive switching median filter for the removal of impulse noise from highly corrupted images," *IEEE Trans. Circuits and Systems - II Analog and Digital Signal Processing*, vol. 46, no. 1, Jan. 1999.

[9] S. Zhang and M. A. Karim, "A new impulse detector for switching median filter," *IEEE Signal Processing Letters*, vol. 9, no. 11, Nov. 2002.

[10] I. Pitas, "Marginal order statistics in color image filtering," Optical Engineering, vol. 29, no. 5, pp. 493-503, 1990.

[11] J. Astola, P. H. L. Yin, and Y. Neuvo, "Vector median filters," *Proc. IEEE* 78, April 1990.

[12] T. Viero, K. Oistamo and Y. Neuvo, "Three-dimensional median-related filters for color image sequence filtering", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, no. 2, pp.129-142, 1994.

[13] Alparone, M. Barni, F. Bartolini and V. Cappellini, "Adaptive weighted vector median filters for motion-fields smoothing," *Proc. IEEE ICASSP*, pp. 771-776, 1996.

[14] T. Viero, K. Oistamo and Y. Neuvo, "Three-dimensional median-related filters for color image sequence filtering", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, no. 2, pp.129-142, 1994.

[15] M. Moore, M. Gabbouj and S. K. Mitra, "Vector SD-ROM filter for removal of impulse noise from color images," *Proc. Eurasip Conference, ECMCS*, 1999.

[16] J. Astola and P. Kousmanen, *Fundamental of Nonlinear Filtering*, CRC Press, 1997.

[17] B. I. Justusson, "Median filtering: statistical properties," *Two-dimensional Digital Signal Processing*, T. S Huang (Ed.), Springer Verlag, 1981.

[18] K. Tang, J. Astola and Y. Neuvo, "Nonlinear multivariate image filtering techniques," *IEEE Trans. Image Processing*, vol. 4, no. 6, pp. 788-797, 1995.

[19] A. Flaig, K. E. Barner and G. R. Arce, "Fuzzy ranking: theory and applications," *Elsevier Signal Processing*, vol. 80, issue 6, pp. 1017-1036, 2000.

[20] T. Chen and H. R. Wu, "Adaptive impulse detection using center-weighted median filter," *IEEE Signal Processing Letters*, vol. 8, no. 1, Jan. 2001.

[21] A. K. Jain, *Fundamental of Digital Image Processing*, Prentice Hall, Information and System Science Series, 1989.

[22] J. –H. Hu, Y Wang and P. T. Cahill, "Multichannel code excited linear prediction coding and its application in magnetic resonance images," *IEEE Trans. on Image Processing*, vol. 6, no. 11, Nov., 1997.

Published research papers related to this work

[1] Kh. Manglem Singh and Prabin K. Bora, "Multi-stage detection based filtering for removal of impulse noise from images," *IETE Journal of Research*, vol. 50, no. 2, pp. 125-131, Mar.-Apr., 2004.

[2] Kh. Manglem Singh, Prabin K. Bora and S. Birendra Singh, "Vector median filter for removal of impulse noise from color images," *WSEAS Trans. Computers*, vol. 3, no. 1, pp. 209-217, Jan. 2004.

[3] Kh. Manglem Singh and Prabin K. Bora, "Noncausal vector linear prediction filter," *WSEAS Trans. on Circuits and Systems*, vol. 2, no 4, pp. 836-844, Oct. 2003

[4] Kh. Manglem Singh and Prabin K. Bora, "Fuzzy based multilevel median filter," *International Conference on Multimedia and Systems*, IIT Madras, August 2000.

[5] Kh. Manglem Singh, Prabin .K. Bora and Anil Mahanta, "Feature preserving filters using fuzzy Kohonen clustering network in detection of impulse noise," *IEEE Tencon International Conference*, vol. 1, pp. 420-423, Sept. 2001,

Singapore.

[6] Kh. Manglem Singh,Prabin K. Bora and S. Birendra Singh, "Texture image classification in the JPEG domain," *SPIE International Symposium,* vol. 4554, pp. 147-152, Oct. 2001, Wuhan, China.

[7] Kh. Manglem Singh,Prabin K. Bora and S. Birendra Singh, "Fuzzy marginal median filter for removal of impulse noise from color images," *International Conference on Intelligent Technologies*, Bangkok, Nov. 2001.

[8] Kh. Manglem Singh and Prabin K. Bora, "2-D linear prediction based median filtering," *National Conference on Comm.*, IIT, Bombay, Dec. 2001.

[9] Kh. Manglem Singh and Prabin K. Bora, "Improved rank conditioned median filter for removal of impulse noise from images," *IEEE Tencon International Conference*, vol. 1, pp. 557 – 560, Beijing, Oct. 2002.

[10] Kh. Manglem Singh, Prabin K. Bora and S. Birendra Singh, " Enhanced vector SD-ROM filter for removal of impulse noise from color images," *International Conference on HPC*, Bangalore, Dec. 2002.

[11] Kh. Manglem Singh, Prabin K. Bora and S. Birendra Singh, Rank-ordered mean filter for removal of impulse noise from images," *IEEE ICIT*, Bangkok, vol. 2, pp. 980-985, 11-14 Dec. 2002.

[12] Kh. Manglem Singh, Prabin K. Bora and S. Birendra Singh, "Impulse noise removal using fuzzy rank selection technique," *ICECE*, Dhaka, pp. 140-143, Dec. 2002.

[13] Kh. Manglem Singh and Prabin K. Bora, "Adaptive rank-ordered mean filter for removal of impulse noise from images," *ICVGIP* 2002, Ahmedabad, Dec 2002 .

[14] Kh. Manglem Singh and Prabin K. Bora, Adaptive vector median filter for removal impulses noise from color images, *IEEE ISCAS*, vol. 2, pp. 396-399, Bangkok, May 2003.

[15] Kh. Manglem Singh, Prabin K. Bora and S. Birendra Singh, "Vector median filter for removal of impulse noise from color images," *Proc. WSEAS*, Singapore 2003.

[16] Kh. Manglem Singh and Prabin K. Bora, "Noncausal vector linear prediction filters," *Proc on WSEAS ISTASC,* Rhodes Island, November 2003.

2014

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी

Indian Institute of Technology Guwahati

Guwahati 781 039

JANUARY

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

FEBRUARY

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	

MARCH

S	M	T	W	T	F	S
30	31					1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

APRIL

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

MAY

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

JUNE

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

JULY

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

AUGUST

S	M	T	W	T	F	S
31					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

SEPTEMBER

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

OCTOBER

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

NOVEMBER

S	M	T	W	T	F	S
30						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

DECEMBER

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

List of Holidays

14 January	–	Prophet Mohammad's Birthday
15 January	–	Magh Bihu/Pongal
26 January	–	Republic Day
17 March	–	Holi
13 April	–	Mahavir Jayanti
14 April	–	Vaisakhi/Vishu
15 April	–	Bohag Bihu
18 April	–	Good Friday
14 May	–	Buddha Purnima
29 July	–	Idu'l Fitr
15 August	–	Independence Day
2 October	–	Mahatma Gandhi's Birthday
3 October	–	Dussehra (Vijay Dashmi)
6 October	–	Idu'l Zuha
23 October	–	Diwali (Deepawali)
24 October	–	Diwali
4 November	–	Muharram
6 November	–	Guru Nanak's Birthday
25 December	–	Christmas Day

Contact Details

Phone: +91-361-2583000

Fax: +91-361-2690762

Web: www.iitg.ernet.in

Public Relations • pro@iitg.ernet.in

SYLLABUS FOR CSIR NET/JRF EXAMINATION IN PHYSICAL SCIENCES
PART-A (CORE)

I. Mathematical Methods of Physics

Dimensional analysis. Vector algebra and vector calculus. Linear algebra, matrices, Cayley-Hamilton Theorem. Eigenvalues and eigenvectors. Linear ordinary differential equations of first & second order, Special functions (Hermite, Bessel, Laguerre and Legendre functions). Fourier series, Fourier and Laplace transforms. Elements of complex analysis, analytic functions; Taylor & Laurent series; poles, residues and evaluation of integrals. Elementary probability theory, random variables, binomial, Poisson and normal distributions. Central limit theorem.

II. Classical Mechanics

Newtons laws. Dynamical systems, Phase space dynamics, stability analysis. Central force motions. Two body Collisions -scattering in laboratory and Centre of mass frames. Rigid body dynamics- moment of inertia tensor. Non-inertial frames and pseudoforces. Variational principle. Generalized coordinates. Lagrangian and Hamiltonian formalism and equations of motion. Conservation laws and cyclic coordinates. Periodic motion: small oscillations, normal modes. Special theory of relativity- Lorentz transformations, relativistic kinematics and mass energy equivalence.

III. Electromagnetic Theory

Electrostatics: Gauss law and its applications, Laplace and Poisson equations, boundary value problems. Magnetostatics: Biot-Savart law, Ampere's theorem. Electromagnetic induction. Maxwell's equations in free space and linear isotropic media; boundary conditions on the fields at interfaces. Scalar and vector potentials, gauge invariance. Electromagnetic waves in free space. Dielectrics and conductors. Reflection and refraction, polarization, Fresnels law, interference, coherence, and diffraction. Dynamics of charged particles in static and uniform electromagnetic fields.

IV. Quantum Mechanics

Wave-particle duality. Schrödinger equation (time-dependent and time-independent). Eigenvalue problems (particle in a box, harmonic oscillator, etc.). Tunneling through a barrier. Wave-function in coordinate and momentum representations. Commutators and Heisenberg uncertainty principle. Dirac notation for state vectors. Motion in a central potential: orbital angular momentum, angular momentum algebra, spin, addition of angular momenta; Hydrogen atom. Stern-Gerlach experiment. Time-independent perturbation theory and applications. Variational method. Time dependent perturbation theory and Fermi's golden rule, selection rules. Identical particles, Pauli exclusion principle, spin-statistics connection.

V. Thermodynamic and Statistical Physics

Laws of thermodynamics and their consequences. Thermodynamic potentials, Maxwell relations, chemical potential, phase equilibria. Phase space, micro- and macro-states. Micro-canonical, canonical and grand-canonical ensembles and partition functions. Free energy and its connection with thermodynamic quantities. Classical and quantum statistics. Ideal Bose and Fermi gases. Principle of detailed balance. Blackbody radiation and Planck's distribution law.

VI. Electronics and Experimental Methods

Semiconductor devices (diodes, junctions, transistors, field effect devices, homo- and hetero-junction devices), device structure, device characteristics, frequency dependence and applications. Opto-electronic devices (solar cells, photo-detectors, LEDs). Operational amplifiers and their applications. Digital techniques and applications (registers, counters, comparators and similar circuits). A/D and D/A converters. Microprocessor and microcontroller basics.

Data interpretation and analysis. Precision and accuracy. Error analysis, propagation of errors. Least squares fitting.

PART- B (ADVANCED)

I. Mathematical Methods of Physics

Greens function. Partial differential equations (Laplace, wave and heat equations in two and three dimensions). Elements of computational techniques: root of functions, interpolation, extrapolation, integration by trapezoid and Simpsons rule, Solution of first order differential equation using Runge-Kutta method. Finite difference methods. Tensors. Introductory group theory: SU(2), O(3).

II. Classical Mechanics

Dynamical systems, Phase space dynamics, stability analysis. Poisson brackets and canonical transformations. Symmetry, invariance and Noethers theorem. Hamilton-Jacobi theory.

III. Electromagnetic Theory

Dispersion relations in plasma. Lorentz invariance of Maxwells equation. Transmission lines and wave guides. Radiation- from moving charges and dipoles and retarded potentials.

IV. Quantum Mechanics

Spin-orbit coupling, fine structure. WKB approximation. Elementary theory of scattering: phase shifts, partial waves, Born approximation. Relativistic quantum mechanics: Klein-Gordon and Dirac equations. Semi-classical theory of radiation.

V. Thermodynamic and Statistical Physics

First- and second-order phase transitions. Diamagnetism, paramagnetism, and ferromagnetism. Ising model. Bose-Einstein condensation. Diffusion equation. Random walk and Brownian motion. Introduction to nonequilibrium processes.

VI. Electronics and Experimental Methods

Linear and nonlinear curve fitting, chi-square test. Transducers (temperature, pressure/vacuum, magnetic fields, vibration, optical, and particle detectors). Measurement and control. Signal conditioning and recovery. Impedance matching, amplification (Op-amp based, instrumentation amp, feedback), filtering and noise reduction, shielding and grounding. Fourier transforms, lock-in detector, box-car integrator, modulation techniques. High frequency devices (including generators and detectors).

VII. Atomic & Molecular Physics

Quantum states of an electron in an atom. Electron spin. Spectrum of helium and alkali atom. Relativistic corrections for energy levels of hydrogen atom, hyperfine structure and isotopic shift, width of spectrum lines, LS & JJ couplings. Zeeman, Paschen-Bach & Stark effects. Electron spin resonance. Nuclear magnetic resonance, chemical shift. Frank-Condon principle. Born-Oppenheimer approximation. Electronic, rotational, vibrational and Raman spectra of diatomic molecules, selection rules. Lasers: spontaneous and stimulated emission, Einstein A & B coefficients. Optical pumping, population inversion, rate equation. Modes of resonators and coherence length.

VIII. Condensed Matter Physics

Bravais lattices. Reciprocal lattice. Diffraction and the structure factor. Bonding of solids. Elastic properties, phonons, lattice specific heat. Free electron theory and electronic specific heat. Response and relaxation phenomena. Drude model of electrical and thermal conductivity. Hall effect and thermoelectric power. Electron motion in a periodic potential, band theory of solids: metals, insulators and semiconductors. Superconductivity: type-I and type-II superconductors. Josephson junctions. Superfluidity. Defects and dislocations. Ordered phases of matter: translational and orientational order, kinds of liquid crystalline order. Quasi crystals.

IX. Nuclear and Particle Physics

Basic nuclear properties: size, shape and charge distribution, spin and parity. Binding energy, semi-empirical mass formula, liquid drop model. Nature of the nuclear force, form of nucleon-nucleon potential, charge-independence and charge-symmetry of nuclear forces. Deuteron problem. Evidence of shell structure, single-particle shell model, its validity and limitations. Rotational spectra. Elementary ideas of alpha, beta and gamma decays and their selection rules. Fission and fusion. Nuclear reactions, reaction mechanism, compound nuclei and direct reactions.

Classification of fundamental forces. Elementary particles and their quantum numbers (charge, spin, parity, isospin, strangeness, etc.). Gellmann-Nishijima formula. Quark model, baryons and mesons. C, P, and T invariance. Application of symmetry arguments to particle reactions. Parity non-conservation in weak interaction. Relativistic kinematics.

RENEWAL REMINDER

OXFORD
UNIVERSITY PRESS

Renewal No: 01090516
Date: 16/06/2015
Account No: 000081852286

Billing Address:

Indian Institute of Technology Guwahati
Library
Doul Govinda Road
Amingaon, Guwahati
Assam
781039
INDIA

Great Clarendon Street
Oxford OX2 6DP
England
44 (0) 1865 353907 *telephone*
44 (0) 1865 353485 *fax*
jnls.cust.serv@oup.com
www.oxfordjournals.org

VAT No. OUP GB 125 50 67 30

Product/Service Description	Qty	Price	Disc%	Net	Tax%	Tax	Line Total
Journal of Consumer Research Print and Online QWC from Jan 2015 to Dec 2015 PO: ISA/I167/2014	1	256.00	0.00	256.00	0.00	0.00	256.00

Total Excl. Tax:	256.00
Total Tax:	0.00
Total Due:	256.00
Currency:	Sterling

To prevent delays in processing please return the Payment Advice section of this form with your payment or quote the Renewal No: shown above. Payments for OUP books MUST NOT be included in the same payment.

Please disregard this notice if you have sent payment within the past 21 days.

Payment is required with all orders.

This is not a VAT invoice

PAYMENT ADVICE - RENEWAL

OXFORD
UNIVERSITY PRESS

Renewal No: 01090516
Date: 16/06/2015
Total Due: 256.00
Currency: Sterling
Account No: 000081852286

METHODS OF PAYMENT:

1) By **bank transfer** to Barclays Bank Plc, Oxford Group Office, Oxford.
GB £ Sterling to a/c no: 70299332. Sort code: 20-65-18. IBAN: GB89 BARC 206518 70299332. SWIFT BIC: BARC GB22
Please include your renewal number on the transfer documents.
Please return this portion of the form to the appropriate address listed on the back of this notice.

2) By **American Express, Mastercard or Visa**. Please see important information on the second page. Due to excessive charges, subscription agents paying by credit card will forfeit their agency discount.

3) By **cheque** in sterling drawn on a UK bank, euro € drawn on a European bank or in US$ drawn on a US bank. Cheques should be made payable to Oxford Journals and sent in with this form to the appropriate address listed on the back of this notice.

If you are amending your order and your payment is not for the full amount due please make a copy of your renewal notice, mark any changes clearly and return the copy with your payment to Oxford Journals.

OXFORD
UNIVERSITY PRESS

Global Academic Publishing
Managing Director: Tim Barton

Great Clarendon Street
Oxford OX2 6DP
United Kingdom

+44 (0) 1865 556767 *telephone*
+44 (0) 1865 556646 *fax*
www.oup.com

IF YOU HAVE SENT PAYMENT TO US WITHIN THE PAST 14 DAYS PLEASE IGNORE THIS REMINDER

Dear subscriber,

Subscription Renewal Notice

IMPORTANT CHANGES TO YOUR SUBSCRIPTION TO THE *JOURNAL OF CONSUMER RESEARCH*

As a current subscriber to the *Journal of Consumer Research,* we would like to let you know that Oxford Journals will be publishing this title from 1st June 2015 (volume 42). To continue receiving future volumes of the journal, you will need to renew your subscription with us.

HOW WILL THIS AFFECT YOU?
We want to make the transfer of your subscription to us as straightforward as possible. All you need to do if you wish to continue receiving the *Journal of Consumer Research* is to renew your subscription with us. You will find full details on the cost of your subscription, how to pay, and your Oxford Journals subscriber number on the enclosed renewal notice.

You will see that we have detailed what it will cost to continue your existing subscription. As part of the transition we have modified the pricing structure to reflect regional currencies, and renamed the pricing tiers. Customers based in Europe will now pay in Euros (€); customers based in the US or Canada will continue to pay in US Dollars ($); and customers based in the UK or the rest of the world will now pay in Pounds Sterling (£). Customers who pay in Euros and Sterling may notice a slight price difference from the rate paid last year.

Please be assured that the actual price tier you will pay has not changed, only the description of the rate. This is to ensure that the range of choices for new subscribers is simplified, and the basis for price differential between the tiers is transparent and easy to validate.

- The rates previously described as Higher Education – Very Small, Higher Education – Small, Community College, Museums, Public Library, Govt/NonProfit, Secondary Schools I, Secondary Schools II, Secondary Schools III, are now consolidated into a single rate: Institutional FTEs under 6,000
- The Higher Education – Medium rate has become: *Institutional FTEs 6,000 to 16,999*
- The Higher Education – Large rate has become: *Institutional FTEs 17,000 to 35,999*
- The Higher Education – Very Large rate has become: *Institutional FTEs 36,000 and above*

If you wish to change your subscription package please just make it clear on your renewal notice which of the options you'd like and pay the relevant amount. Full pricing options can be found at: http://jcr.oxfordjournals.org/subscribe

Please send your payment to the nearest office detailed on your renewal notice. If you need any additional information on our pricing policies, you'll find full details at: www.oxfordjournals.org/access_purchase/

We look forward to receiving your renewal instructions in the near future.

Yours sincerely

Journals Customer Services
Oxford Journals | Oxford University Press

Oxford Journals – 2015 Renewal Information

We are sending your 2015 renewal information by mail as we do not hold a valid email address for you. So we can send future renewal communications to you by email, please provide your email address by filling in the form on our website here:

http://www.oxfordjournals.org/help/change_address.html

EU TAX CHANGES

From 1 January 2015 the VAT rules in Europe change. This affects customers for online products in the EU not in the UK who have not provided a VAT number. For payments received by 31 December 2014, UK VAT will apply to the online element of subscriptions (100% of an online subscription, 35% of a combined subscription). For payments received after that, the local rate applicable in each of the 27 EU countries will apply.

SUBSCRIPTION RENEWAL

OXFORD
UNIVERSITY PRESS

Renewal No: 01089398
Date: 11/05/2015
Account No: 000081852286

Billing Address:
Indian Institute of Technology Guwahati
Library
Doul Govinda Road
Amingaon, Guwahati
Assam
781039
INDIA

Great Clarendon Street
Oxford OX2 6DP
England
44 (0) 1865 353907 *telephone*
44 (0) 1865 353485 *fax*
jnls.cust.serv@oup.com
www.oxfordjournals.org

VAT No. OUP GB 125 50 67 30

Product/Service Description	Qty	Price	Disc%	Net	Tax%	Tax	Line Total
Journal of Consumer Research Print and Online QWC from Jan 2015 to Dec 2015 PO: ISA/I167/2014	1	256.00	0.00	256.00	0.00	0.00	256.00

To prevent delays in processing please return the Payment Advice section of this form with your payment or quote the Renewal No: shown above. Payments for OUP books MUST NOT be included in the same payment.

Total Excl. Tax:	256.00
Total Tax:	0.00
Total Due:	256.00
Currency:	Sterling

Payment is required with all orders.

This is not a VAT invoice

✂.....

PAYMENT ADVICE - RENEWAL

OXFORD
UNIVERSITY PRESS

Renewal No: 01089398
Date: 11/05/2015
Total Due: 256.00
Currency: Sterling
Account No: 000081852286

METHODS OF PAYMENT:

1) By **bank transfer** to Barclays Bank Plc, Oxford Group Office, Oxford.
GB £ Sterling to a/c no: 70299332. Sort code: 20-65-18. IBAN: GB89 BARC 206518 70299332. SWIFT BIC: BARC GB22
Please include your renewal number on the transfer documents.
Please return this portion of the form to the appropriate address listed on the back of this notice.

2) By **American Express, Mastercard or Visa**. Please see important information on the second page. Due to excessive charges, subscription agents paying by credit card will forfeit their agency discount.

3) By **cheque** in sterling drawn on a UK bank, euro € drawn on a European bank or in US$ drawn on a US bank. Cheques should be made payable to Oxford Journals and sent in with this form to the appropriate address listed on the back of this notice.

If you are amending your order and your payment is not for the full amount due please make a copy of your renewal notice, mark any changes clearly and return the copy with your payment to Oxford Journals.

OXFORD
UNIVERSITY PRESS

Global Academic Publishing
Managing Director: Tim Barton

Great Clarendon Street
Oxford OX2 6DP
United Kingdom

+44 (0) 1865 556767 *telephone*
+44 (0) 1865 556646 *fax*
www.oup.com

01 May 2015

Indian Institute of Technology Guwahati
Library
Doul Govinda Road
Amingaon, Guwahati
Assam 781039
INDIA

Subscription number: 81852286

Dear subscriber,

We are pleased to inform you that Oxford Journals has commenced publishing the *Journal of Consumer Research* from 1st April 2015 (volume 42). As a current subscriber, your details have been passed to us by the University of Chicago Press and entered into our database.

As part of the transition we have modified the pricing structure to reflect regional currencies, and renamed the pricing tiers. Customers based in Europe will now pay in Euros (€); customers based in the US or Canada will continue to pay in US Dollars ($); and customers based in the UK or the rest of the world will now pay in Pounds Sterling (£). Customers who pay in Euros and Sterling may notice a slight price difference from the rate paid last year.

Please be assured that the actual price tier you will pay has not changed, only the description of the rate. This is to ensure that the range of choices for new subscribers is simplified, and the basis for price differential between the tiers is transparent and easy to validate.

- The rates previously described as Higher Education – Very Small, Higher Education – Small, Community College, Museums, Public Library, Govt/NonProfit, Secondary Schools I, Secondary Schools II, Secondary Schools III, are now consolidated into a single rate: *Institutional FTEs under 6,000*
- The Higher Education – Medium rate has become: *Institutional FTEs 6,000 to 16,999*
- The Higher Education – Large rate has become: *Institutional FTEs 17,000 to 35,999*
- The Higher Education – Very Large rate has become: *Institutional FTEs 36,000 and above*

If your subscription has expired, or is due to expire soon we will shortly be sending you or your agent a renewal notice. Please arrange for the renewal to be processed as soon as possible to ensure uninterrupted supply. Upon our receipt of your renewal instructions and payment, you will be sent an order confirmation letter or email, together with full details on how to access content if you have purchased an online or combined subscription.

If your subscription is still active and includes online access, you may need to register an account with us to access online content. Please see our website for further instructions. Please note that content already published will unfortunately not be available on the Oxford Journals platform until 1st June 2015. To access previously published content before this date, please continue to use the previous publisher's website.

If you require further information regarding the transition or our services in general, please contact us using one of the following:

Subscribers in the USA and Canada
+ 1-800-852-7323 (toll-free)
jnlorders@oup.com

Subscribers in the UK and the rest of the world
+44 (0)1865 353907
jnls.cust.serv@oup.com

Yours sincerely

Journals Customer Services
Oxford Journals | Oxford University Press